

A Methodological Leg to Stand On: Lessons Learned Using Grounded Theory to Study Software Development

Steve Adolph

Electrical and Computer
Engineering
University of
British Columbia

Wendy Hall

School of Nursing
University of
British Columbia

Philippe Kruchten

Electrical and Computer
Engineering
University of
British Columbia

Abstract

We are engaged in a qualitative research project to understand how people manage the process of software development. This study uses grounded theory as its method of inquiry and we have learned much about what is and what is not a grounded theory. We, like many researchers have claimed to follow grounded theory methods and even to have produced a grounded theory. In reality, we often only borrow a few grounded theory practices to categorize our data. This paper presents lessons learned about using grounded theory so that both researchers and reviewers can critically evaluate investigators' claims to be producing grounded theory.

1 Introduction

The rise in interest in agile software development with mantras such as “people trump process” [1] and a manifesto declaring “individual and interactions over process and tools” [2] means software engineering research and specifically software process engineering must take a greater interest in behavioral research. However, software engineering does not have a strong tradition of empirical software engineering research [3, 4, 5],

let alone experience with the qualitative research methods necessary for understanding the social and behavioral aspects of software development [6]. Only a limited number of qualitative studies are performed in the software engineering research community [7, 8].

Yet, if we are to truly understand people problems we must delve into the complexity of a phenomena [6] and qualitative research methods reveal the richness of a phenomena. Of the five qualitative methods elaborated by Creswell [9], grounded theory [10, 11, 12] is a research method employed in a number of software engineering studies. Grounded theory has been used to study software phenomena such as CASE tool adoption [13], information sharing in personal learning spaces [14], information systems development [15], software process improvement in the Irish software industry [16], short cycle time development [17], the social nature of agile teams [18] and automated testing [19].

We are currently engaged in a qualitative research project to understand how people manage the process of software development. We are using grounded theory as our method of inquiry and we have learned much about what is and what is not grounded theory. Like many other researchers who have claimed to follow grounded theory methods and even produce a grounded theory, many of us have only borrow a few grounded theory practices and have not followed grounded theory as a comprehensive method. At best, this means the legitimacy of good research can be easily

easily challenged, and at worst we are giving theory status to hearsay and anecdotal stories. This has serious consequences for our industry if the purpose of this research is to inform software development policy. This is not unexpected because as engineers we are not trained in social science research methodology. Yet, if we believe people factors are significant software project cost drivers and determinants to software team performance, then this is exactly the type of research methodologies we must be expert in if we are to understand software engineering phenomena.

This paper presents our lessons learned using grounded theory with the hope of better informing researchers and reviewers what differentiates grounded theory from what may sometimes only be called journalism. While there are numerous grounded theory texts, our own experience and the experience of others suggests we do not have a good understanding of grounded theory.

We will first present an overview of our research project and then describe grounded theory. Throughout our description of grounded theory we will attempt to illustrate good practices and poor practices from our own research, and other papers claiming to follow grounded theory. It is not our purpose to criticize what in many cases is excellent research conducted by our colleagues, but to demonstrate how easy it is to challenge the research because of the lack of rigor in the research method. However, we are concerned that if we do not identify these issues then we risk repudiation of the very methods we need to understand the influence behavior has on software engineering phenomena.

2 Our Study

We are interested in understanding how software methods are used (or not used) in software development projects. Methodologists argue the benefits of following a software development methodology (SDM) and studies demonstrate a positive correlation between SDM adoption and team effectiveness in terms of product quality and productivity [20, 21, 22]. However, industry data demonstrate SDM effect on productivity is limited [23]. Furthermore, methodology adoption and the contribution of software methods to team effectiveness are frequently questioned [1, 24, 25].

What is going on here? Low rates of software methods adoption are a strong indicator that software practitioners do not believe their needs are addressed by software methods. While we all have our anecdotal stories about methodology usage or lack of usage, we do not have a substantive theory which explains how people manage the software development process. The goal of this study is to create a substantive theory that will better inform the design of software methods. We are therefore interested in answering the question how do people manage the process of software development?

Our study is occurring in three phases, a First Report[26], a Pilot Study¹, and a Main Study. In our First Report we opportunistically took advantage of data collected by Scott Duncan during an Agile 2006 workshop. This was our first use of grounded theory and enabled us to experiment with coding, tool usage, and narrative writing. The First Report was followed by a Pilot Study where we took lessons learned from our First Report and experimented with the conduct of field work. During the Pilot Study we developed an interview guide, learned how to navigate the ethics review board, and conduct interviews. For the Pilot Study we interviewed four staff members at a company who had all worked on the same project. The first author of this paper is a former member of the project team. This paper describes some of the key lessons we learned about conducting a grounded theory study during our First Report and Pilot Study and how we will apply these lessons learned to our Main Study.

In addition to critically reviewing our own research we will also examine other software engineering research papers that applied grounded theory practices. The examples used in this paper were not chosen because they are either specifically good or poor examples of grounded theory. Rather they were chosen because the authors followed the best of scientific traditions and took care to clearly explicate their methodology enabling critical review of their work. Unfortunately many other papers from our literature search simply stated they used grounded theory and provide little or no description of their method.

¹ Unpublished study.

3 What is Grounded Theory?

Grounded theory is a qualitative research method that discovers theory from data [10] and is useful for discovering behavioral patterns that shape social processes as people interact together in groups. The intent is to develop an account of a phenomenon that identifies the major categories, their relationships, and the context and process, thus providing a theory of the phenomenon that is much more than a descriptive account [27]. The goal of this grounded theory is to understand the action in a substantive area from the point of view of the actors involved [28 pp 115].

Co-discoverers Barney Glaser and Anselm Strauss, called the method “grounded” because a theory is systematically generated from a broad array of data through a rigorous process of constant comparison. Grounded theory is different from the dominant logico-deductive methods of inquiry because rather than develop a theory and then systematically seek out evidence to verify it, grounded theory researchers gather data and systematically generate a mid-level substantive theory derived directly from the data [10 pp 4].

For many of us in the software engineering field, the term theory is a source of confusion with respect to grounded theory. Our strong physical sciences and mathematics backgrounds tend to make us regard a theory as a universal truth like Einstein’s theory of relativity. Theories do not have to be universal truths and can vary in their coverage. Grounded theory generates a mid level or substantive theory describing processes in social movements, organizations, or communities. A further difficulty for those more comfortable with an objective view of reality is that when studying people we cannot ignore personal values because people interpret reality based on their personal values.

3.1 Why Grounded Theory?

If people are important to the software development process, then their personal values will influence that process. Grounded theory is useful when we want to learn how people manage their lives in the context of a problematic situation and is useful for learning the process of how people understand and deal with what is happening to them through time and changing circumstances.

Grounded theory is useful for research in areas that have not been previously studied or where a new perspective might be beneficial [29].

Grounded theory is not an appropriate inquiry method for answering an objective question such as “are pair programming teams more effective than individual programmers” However, grounded theory is appropriate if we are interested in understanding the experience of pair programming by discovering “How do individuals manage the process of pair programming” The answer to such a question may help inform the creation of policy for pair programming.

We initially chose grounded theory for our First Report and Pilot Study without much thought because it seemed like the right thing to do. Grounded theory appeared more “analytical” and less ad hoc than other qualitative research methods, and seems to have some acceptance as a research method in the empirical software engineering research community. Unfortunately, we were effectively using grounded theory as a data management strategy to categorize data rather than as a method of inquiry.

As we progressed into our research and learned more about grounded theory we became aware of the investigative opportunity grounded theory opens for us and have re-shaped our research agenda to take advantage. Grounded theory gives us the capability to answer the question “what is going on here when it comes to developer’s apparent disdain for software methods?” The simplest way to answer this question is to go out and ask software developers what they see as their main concern and how do they resolve it rather than imposing our own pre-conceived ideas and theory of what the problem is. We may not get the answer we are looking for, but we will certainly get an answer to the problem the developers in the field think is important.

4 Which Version of Grounded Theory?

What confuses most novice grounded theory researchers is there is not one, but at least three similar, yet distinct research methods that claim the name grounded theory. A split between Glaser and Strauss began to reveal itself over the

development of grounded theory when Glaser published without Strauss *Theoretical Sensitivity* [30] which focused on developing researcher's creativity. In contrast, Strauss was interested in looking at ways for formalizing and proceduralising grounded theory and published *Basics of Qualitative Research* with Corbin in 1990 [11]. Differences of opinion between the grounded theory co-discoverers burst into full view with Barney Glaser's publication of *Emergence versus Forcing: Basics of Grounded Theory Analysis* [31] decried Straus and Corbin's approach. Kathy Charmaz attempted to clarify some of grounded theory's ontological and epistemological ambiguities by publishing *Constructing Grounded Theory* [12].

The more prescriptive approach of the Straussian form of grounded theory apparently appeals to software engineering researchers because more papers appear to follow Strauss and Corbin rather than Glaser. As software engineering researchers conducting a qualitative study, the philosophical arguments behind the different grounded theory methods should not concern us other than to be aware there are multiple flavours of grounded theory. What is important for us is to be clear which method we chose, why, and not mix methods.

This is where we made one of our first mistakes when in our First Report we cited Strauss and Corbin and yet we patched together different practices from Glaser and from Charmaz. We were simply unaware there were a variety of "competing" methods and in our methodology section simply stated like many others that we were following grounded theory.

Other studies also cite numerous grounded theory references, but do not explicitly describe the method they are following. For example, Whitworth and Biddle's study of agile teams and Hansen and Kautz's study of information systems development [15, 18] cite numerous grounded theory sources in their description of methodology. However they never explicitly state the specific method they are following, although explicit references to open coding, axial coding and selective coding infer they are following a Strauss and Corbin like method. Orlikowski's study of CASE tool adoption [13] also cites numerous grounded theory sources in her description of grounded the-

ory, and yet does not explicitly state which method she is following. It is only during the description of her study that one discovers she followed the inductive theory generation method described by Eisenhardt [32].

One exception is Coleman and O'Connor [16 pp 31] who clearly explain why they chose to follow Strauss and Corbin's approach to grounded theory. Clearly specifying which method a study follows is necessary if we are to have the correct standard by which to judge the quality of the study.

For our Main Study we have decided to follow the Glaserian approach for one very pragmatic reason; there are more Glaserian resources available to us in terms of mentors, seminars, and websites.

5 The Process of Grounded Theory

The process of grounded theory is like agile software development, deceptively simple conceptually, yet rigorous and disciplined in practice. Data is collected and incidents are coded into categories. The theoretical properties of the category are developed by comparing incidents in newly collected data with previous incident in the same category. This develops the properties and dimensions of those categories. As the study progress, the focus changes from comparing incidents with one another to comparing incidents with the properties of the categories.

5.1 The Research Problem

Grounded theory is a method for discovering the real problem that exists for the participants in a substantive area rather than what professional researchers may believe is their problem. Therefore, in a grounded theory study the researcher works with a general area of interest rather than with a specific problem [33]. This does not mean there is no specific problem, but as Glaser writes "This problem and its processing will emerge in the initial stages of the research. And it will emerge if it is not derailed by what the researcher thinks is relevant beforehand and forces it on the study" [28 pp 116].

Our First Report did not have a specific research question. Unfortunately, this resulted from a lack

thoughtful design rather than intent. In the First Report we opportunistically took advantage of data collected and just coded it. In contrast, our Pilot Study had a strong pre-conception that coordination is the real problem in software development and we used this problem to guide our research design. In retrospect, this preconception led us away from the problems that were more relevant to the participants. In commenting on problem emergence in his own study ‘awareness into dying’ [34] Glaser states, “we had no idea that awareness was a problem that the medical team had to handle constantly and that they did so by generating awareness contexts” [28 pp 119]

For our Main Study we are working with a very broadly defined problem statement of “how do people manage the process of software development?” While we have some preconceived ideas about potential problems, (e.g. coordination, communication, adaptation), we are leaving our problem statement open so that we can discover if any of our pre-conceived problems matter to people engaged in software development.

Whitworth and Biddle also defined a very broad problem statement for their study into the social nature of agile teams, asking the question “what is the experience of being in an agile software development team” [35 pp 8]. Coleman and O’Connor [16 pp 28] had more narrowly defined question for their study asking “why are software companies not using best practice SPI models?”

Many of the studies we reviewed had much more precisely defined problem statements, such as the following from Razavi and Iverson’s study of information sharing in personal learning spaces [14]:

What factors affect privacy of information from a user’s perspective in a personal learning space?

- *Is privacy management in a personal learning space considered important, and why?*
- *What are users’ challenges in managing selective disclosure?*
- *What are users’ strategies in achieving privacy?*
- *Are there any commonalities in the way users arrange their information with regard to sharing?*
- *Is it possible to drive a set of default privacy settings for different categories of*

information in this environment, which users can easily modify later in context?

When problems are constrained by the nature of the research questions from the outset (e.g., specific concepts and approaches) data collection will be constrained by how the researcher has defined the problem and the nature of the specific questions which reduced sensitivity and openness to emerging theory that may be meaningful to software developers [27].

What is also missing from this set of questions is a sense of process. A distinguishing characteristic of grounded theory is a focus on process – the process of how people resolve their main concern, which often results in identifiable stages and phases [29 pp 2]. An alternative to the above set of research questions might have been “how do users manage privacy in personal learning spaces?”

5.2 Constant Comparison

The heart of grounded theory is the constant comparison method of data analysis where the analyst starts by coding the data for incidents that explain what is happening in the data. As codes are developed they are compared with previous codes from within the same interview and from other interviews. Codes are progressively clustered into categories. As the coding continues, the constant comparative units change from the comparison of codes representing incidents to comparison with the properties of the categories, which resulted from the initial comparison of incidents. [36 pp 440].

Constant comparison is facilitated by the concurrent and iterative collection of data and analysis. This differentiates grounded theory from waterfall-like methods of inquiry where all data is collected during a data collection phase and then analyzed during an analysis phase. The constant comparison method enables researchers to adjust their questions as the ongoing analysis begins to reveal the key concern and problem the participants are resolving. The developing theory guides the inquiry. Without constant comparison there is no grounded theory.

Our First Report did not follow the constant comparison method because we opportunistically ana-

lyzed previously gathered data by simply categorizing the data without comparing the incident represented by the data with previously coded instances in the same category. In our Pilot Study, we iteratively and concurrently collected and analyzed data, but we did not follow the constant comparative process. Like our First Report, we were simply categorizing data rather than comparing incidents, the only difference being this time we were doing it iteratively. The experience gained in our Pilot Study is helping to inform the design of the Main Study and not to assume constant comparison is merely iterative collection and analysis of data.

About half of the studies we reviewed for this paper make reference to the constant comparison process and usually describe an iterative data collection and analysis process. Razavi and Iversen write how , “data collection and data analysis occurred simultaneously in this study” [14 pp 462]. Coleman and O’Connor also describes an iterative process where “Any memos, or propositions, that emerged through the coding process were recorded for further analysis and inclusion as questions in subsequent interviews” [16 pp 32]. This iterative approach enabled those researchers to adjust their inquiry as their ongoing analysis began to reveal the key concern and problem the participants are resolving. The developing theory guides the inquiry. However, it was not clear if like us in our Pilot Study if constant comparative analysis was only seen as iterative collection and analysis of data.

Orlikowski describes a very coarse grain approach to constant comparison, “To facilitate this iteration and comparison, two field sites were studied and analyzed in turn....The initial concepts thus emerged in one organizational context and were then contrasted, elaborated, and qualified in the other” [13 pp 312].

In the remaining studies we reviewed, all data was either collected during a primarily data collection phase and then later analyzed in a data analysis phase, or it is never clearly stated how the data was collected. Hansen and Kautz’s study of how systems developers use development methodologies [15] describes taking a month to collect and transcribe interviews before coding the data in the subsequent analysis phase.

Collecting the majority or all data during a data collection phase and then analyzing the data in a subsequent data analysis phase eliminates the opportunity to use the emerging theory to guide the data collection. This means the researcher cannot employ theoretical sampling and is not properly sampling the problem space.

5.3 Data Collection

We collected data in our Pilot Study using a semi structured interview. This is the method of data collection employed by most of grounded theory studies we have reviewed. Unfortunately some research methodologists are critical of relying exclusively or mainly on interview data [37] because interview data will only provide us with a superficial and recollective description of a phenomena. Field research means going into the field and immersing oneself in the environment. Returning to grounded theory’s symbolic interactionist roots, Blumer insisted social life must be studied through first hand observation [38 pp 38]. The researcher also has to honestly articulate his or her inescapable bias. Also, in some situations (studying sub-cultures) the participants will all use the researcher for their own purposes. Participant observation therefore becomes an important mainstream data collection method rather than a mere supplement to interviews.

In our Pilot Study we initially relied completely on interview data until we realized the importance of other data collection techniques. We tried to compensate for this deficiency by capturing our recollection of participants. We had spent many hours at the subject site and therefore were able to collect field notes from memory. However, these field notes are recollections and are not the same quality as field notes captured in real-time. For the design of our Main Study we plan to spend much more time collecting data through participant observation and preparing a participant observation guide.

In addition to participant observation we will also incorporate document analysis into our data collection protocol. This data triangulation approach to our data collection provides greater confidence that we’re discovering the real phenomena and not just what is recollected by the interview participants. There is a risk different kinds of data

may capture different things and it may be difficult to triangulate apples and oranges[39].

The majority of studies we examined made exclusive use of semi-structured interviews to collect data. Some however, collected more diverse data such as de Souza et. al. [40] who describe a rich data collection process using non-participant observation, semi-structured interviews, collection of documents such as meeting invitations, software change requests, and access to a corporate discussion data base. Hansen and Kautz [15] used organizational documents to supplement interview data. Orlikowski also describes a rich data collection process using unstructured and semi structured interviews, documentation review and observation [13 pp 312].

For the Pilot Study we used semi-structured interviews and prepared an interview guide with a set of “guiding questions” However, most of the questions became irrelevant as we began to pay more attention to the concerns of the subject rather than our specific interview questions. In retrospect, some of our early attempts to bring the participant back to the interview question resulted in us ignoring a potentially interesting line of investigation had we chosen to pursue it. While it is important to remain disciplined and focused in our data collection we must remember that grounded theory is an exploratory method that discovers how the participant handles a problem that is important to them. Our interview protocol must be suitably exploratory as well.

Morse [41] is critical of the use of semi-structured interviews and recommends an unstructured interview protocol because data from semi-structured interviews rarely contain the continuous in-depth stories that the retrospective accounts from unstructured interviews do. Grounded theory is about discovering a process and the snippets of data that are obtained from semi-structured interviews may not readily reveal the underlying process and its structure. According to Morse “the narrative form with events told as they unfold is best suited for grounded theory data” [41 pp 4]. For the design of our Main Study we taking Morse’s advice and designing a much less structured interview protocol.

5.4 Analysis

Data analysis in qualitative research discovers meaning from words and coding is one way of exploring meaning in the data by looking for similarities and differences within the data to categorize and label the data. While the various grounded theory flavours have developed their own specific practices and nomenclature for analysis, they follow the same basic approach of fracturing and integration. Line by line coding fractures the data to explore all possible aspects of the issues and ideas in the data and to develop descriptive codes as labels for the meanings of these issues and ideas [42 pp 253]. Categories and their properties emerge and these categories are integrated into relationships from which theory emerges.

The coding process is tightly integrated into the constant comparative approach. Glaser has defined coding as “conceptualizing data by constant comparison of incident with incident, and incident with concept” [31 pp 38]. All generated ideas are therefore “verified by all data and categories are constantly refitted” [30]. Each concept must earn its way into the theory by repeatedly being present in the data. Through this ongoing comparison of incidents in old and new data, the researcher is continuously challenging categories with fresh data, verifying their ideas and alerted to the danger of their biases forcing the data. [43]. This process also allows the identification of gaps in the understanding and leads the researcher to sample theoretically for participants or experiences to fill those gaps [44 pp 186]

With both our First Report and our Pilot Study we blissfully deceived ourselves that by simply discovering reasonable in-vivo codes and then reducing and integrating those codes into categories that we were following good grounded theory practices. What we failed to do was to follow constant comparative analysis and compare incidents in the new data to the existing categories. We did not provide ourselves with the opportunity to challenge ourselves that we may be forcing the data into pre-conceived concepts.

The most serious transgression we observed in grounded theories is the failure to select a core category or clearly describe the core category within the theory. The core category is that pat-

tern of behavior which is most related to all other categories and their properties in the theory which explains how the participants resolve their main concern (Glaser 1998, pg 117). Selecting a core category and building theory around the core category is necessary for the generation of parsimonious theory that is relevant and feasible, and hence it conveys functions of “integration, density, saturation, completeness and delimiting focus” [31 pp 75] .

In our First Report, we developed a “barrier” category as our core category which represented the obstacles that inhibit feedback. There is little substantive data to support our selection of this core category other than it fit our expectations of what an interesting theory about agile software development may look like. This is a blatant example of our pre-conceived ideas force fitting a concept into a pre-conceived theory. Our Pilot Study has not progressed far enough for a definitive core category to emerge yet, but one candidate is what we have called “Illuminated Reflection” that is the issues and phenomena a software method draws attention to that may otherwise go unnoticed or misunderstood by the team. This category so far appears to be related to many of our emerging categories and reoccurs frequently in our data [45].

Coleman’s and O’connor’s study of the Irish software industry identified *Cost of Process* as a core category [16 pp 34]. Coleman and O’Connor describe the core category as “the hub for all other identified categories, it must be central in that all other categories must relate to it and it must appear frequently in the data” [16 pp 31]. *Cost of Process* is offered as this hub, explaining why software companies are not using ‘best practice’ SPI models.

In their study of how practitioners use APIs in their daily work, de Souza et. al. hint at a possible core category when stating “we eventually realized the fundamental role of APIs in the management of the interdependencies” [40 pp 223].

None of the remaining papers we reviewed describe a single core category or a theory that explains how participants resolve their main concern as represented by the core category. Some researchers find themselves in the position of Whitworth and Biddle who explicitly declared they did not discover a core category due to time

limitations [35 pp 61]. Unfortunately, resource constraints, and time especially, is a very real problem for many researcher and clearly stating this is an acceptable approach to dealing with this problem.

Other researchers reported discovering multiple core categories. Razavi and Iverson [14] describe a theory of information sharing behaviour built around two core categories. Hansen and Kautz reported multiple “themes” because their data indicated more than one significant theme [15 pp 6]. Multiple core categories can detract from the claim the core category accounts for most of the variation in the theory and integrates and relates to the other categories.

5.5 Theoretical Sampling

In any research project sampling strategies and the adequacy of the selected sample impact the scientific validity and reliability of the study. For a grounded theory study, the population under study is the set of concepts that constitute the phenomena and not the set of individuals experiencing the phenomena. Therefore the sampling strategy must discover sufficient concepts to develop a conceptually dense substantive theory. Grounded theory employs theoretical sampling where the analysts “jointly collects, codes, and analyzes his data and decides what data to collect next and where to find them in order to develop his theory as it emerges” [30 pp 36].

Theoretical sampling is different from selective sampling which uses a predetermined selection process. Theoretical sampling is driven by the emerging categories and hypotheses, and is therefore an ongoing process which cannot be predetermined. It is a critical practice supporting the development of an emerging theory that is tightly intertwined and supportive of iterative and concurrent collection of data. “By responding to the need for theoretical completeness, theoretical sampling directs the researcher to new data sources which constantly generate the parsimony and scope of the theory as it accounts for how the main concern is constantly resolved” [28 pp 158]. According to Glaser “taking a random sample of a set number of people does not make sense in grounded theory” [28 pp 159]. By sampling individuals a researcher may repeat samples of the same concepts, but may not be adequately sam-

pling the concepts in the problem space to produce a grounded theory.

Theoretical sampling means it is difficult to articulate in a grounded theory proposal when it will be done because there is no “N”, just sampling for saturation and completeness [28]. Data collection continues until all categories are saturated, that is, when further collection of data indicates no new categories or the need for the expansion of existing categories and all data fits into the existing categories.

Our First Report did not employ a sampling strategy, we simply re-used data that had been acquired for another study and coded it. Re-using data acquired from another study is common in many studies. Orlikowski [13], collected her data as part of a larger study examining the social process of developing, implementing, and using process technology in a software production environment. Orlikowski’s approach to theoretical sampling was to preselect two organizations for their similarities as well as their differences [13 pp 312].

Many studies were like Whitworth and Biddle[18] where it is difficult to determine if interview candidates were pre-selected to provide a wide range of role perspectives or if they were selected as categories emerged during analysis because they were likely to possess knowledge useful for expanding the emerging category. Whitworth and Biddle do describe how their “interview guide evolved as the study progressed based on coding, memoing, theoretical category development surrounding the questions highlighted by the theoretical framework, and general impressions between interviews” [35 pp 56].

Razavi and Iverson [14] describe purposefully selecting nine participants for their study because they were seeking “expert participants with rich, extensive prior experience with the phenomenon in order to be able to provide the researcher with a valid account of their experience” [14 pp 461]. This purposeful sampling was subsequently followed up by “*theoretical sampling*, a procedure through which we consciously selected participants according to their potential for developing new insights or refining the insights that had already been gained”

For our Pilot Study, we initially started with purposeful sampling and selected three participants to interview. These participants were chosen because of their willingness to participate in our study and also because in our opinion they had a good sense of what was happening on their project and were able to articulate this. In our Pilot Study we can make a weak claim to theoretical sampling because during our analysis of the collected data we noticed the emergence of a category “management expectations”. We decided to expand this category by inviting the Chief Operating Officer to participate in our Pilot Study. The data we were able to collect was both enlightening and helped us expand the category.

6 Is it a Grounded Theory?

Our First Report and Pilot Study and most of the papers that explicated their research method committed what may be labeled methodological transgressions [46], the researchers did not follow the fundamental practices of grounded theory, constant comparison and theoretical sampling. Other common grounded theory pitfalls include:

No core category	The core category is the basis of grounded theory and there is no theory without it.
No Process	The theory did not describe a process of how the participants resolve their main concern.
Forcing	The researcher attempts to force the data into a pre-conceived theory rather than let the problem and theory emerge from the data.
Premature Closure	causes the theory to be incomplete (categories were not saturated) to lack density and to inadequately cover the behavioural variations

Many of the papers applied grounded theory mechanically revealing what Suddaby called “a neurotic overemphasis on coding” [47]. This unfortunately tends to lead to a belief that

grounded theory is merely a coding practice which risks creation of results that are merely products of content analysis and not theory. The problem is that content analysis has a significantly different goal than grounded theory and different criteria for reliability and validity.

A grounded theory is not an inventory of concepts annotated with quotes from research participants. A grounded theory is an integrated set of categories that explain the process of how the research participants resolve what is of concern to them. Examples of grounded theory are Hall's studies into the experience of parents after child birth [48, 49] and McCallin's study of health professionals working in interdisciplinary teams[50].

7 Is a Grounded Theory Necessary?

Constant comparison analysis makes grounded theory a time consuming tedious process. Some of the studies we reviewed explicitly stated they had used grounded theory practices and did not claim to create a grounded theory. This is perfectly reasonable as long as we do not claim that we are creating a grounded theory because in doing so we are claiming to have discovered the main problem of the participants and how they resolve it.

There are many alternative approaches for analyzing qualitative data and generating theory besides grounded theory. Reporting research in a descriptive study is a valuable and often overlooked option [51]. The use of grounded theory practices can help a researcher make sense of the large volumes of data collected during a qualitative study. Our First Report and most of the reviewed papers demonstrate the value of content analysis because while they do not present a grounded theory, they offer a rich description of a phenomenon. Numerous papers have pointed out the benefits of combining grounded theory techniques with other methods of inquiry such as ethnography [52].

Finally there are also other theory building methods such as Eisenhardt's method for inductive theory development from case studies [32]. Orlikowski's investigation into CASE tool adoption is an example of Eisenhardt's method.

8 Lessons Learned: It Isn't Easy

Length restrictions prevent us from fully characterizing other essential grounded theory practices (e.g. memoing, theoretical sensitivity, theoretical pacing). However, we have learned several lessons from our investigation into grounded theory that should guide how we design and report grounded theory research:

- There are multiple variants of grounded theory and we must be clear about which variant we are following.
- The researcher begins with a general area of interest rather than with a specific problem.
- The constant comparative method is one of the key practices of grounded theory and is much more than iterative collection and analysis of data.
- Exclusive use of semi-structured interviews does not do justice to grounded theory. Including interviews, document analysis, and participant observation is more desirable for data collection.
- There is no theory without a core category that accounts for a "pattern of behavior that is relevant and problematic to those involved" [30 pp 93].
- Theoretical sampling means the emerging theory guides the sampling process and we can only plan in advance initial purposive sampling.
- A grounded theory explains how people resolve their main concern. The core category is often a process.
- It is useful to employ grounded theory practices to manage, categorize, and describe data. It is not acceptable to claim that all description constitutes theory.

Grounded theory is not easy and our discipline is not alone in misunderstanding grounded theory. In a review of some 146 nursing papers which purported developing a grounded theory, only 33 papers qualified as a grounded theory [53]. Some of the early grounded theory writings do not help illuminate what grounded theory really is often describing grounded theory in near mystical terms. This makes it difficult for the grounded theory novice to fully understand what it really is all about. According to Glaser the most effective way to learn grounded theory is to learn from

someone who has already done it. Unfortunately, most of us do not have a mentor we can turn to help guide us through this land-mine filled maze. Glaser referred to those of us in this position as minus mentorees.

Some of us are fortunate to be associated with institutions that have research oriented Nursing Schools which have a long and respected tradition of qualitative research and grounded theory. This is how we began to improve our understanding of grounded theory by stepping out of our comfort zone into another discipline to learn the research methods we require to conduct this study. Our strongest recommendation is if this resource is available to you, then take advantage of it. The grounded theory institute (www.groundedtheory.org) is another resource.

If people truly trump process then we need research methods that help us understand how people interpret their situation and grounded theory is an excellent method for this. However if we are not rigorous in our application of grounded theory or do not raise our expectations of what is claimed as a grounded theory then we be offering journalism as research. Worse, we may be missing the real problems and their management that are the real concern to the developer in the cubicle.

Acknowledgements

The authors would like to thank the Agile Alliance for their generous support of this research.

About the Authors

Steve Adolph is a PhD candidate in Electrical and Computer Engineering at the University of British Columbia. His research interests include software process engineering, requirements management, and software architecture. His Internet address is steve@wsaconsulting.com

Wendy Hall is a nursing professor in the School of Nursing at the University of British Columbia. Her research interests include research methods and parent child development. Her Internet address is Wendy.Hall@nursing.ubc.ca

Philippe Kruchten is a professor of software engineering at the University of British Columbia. He

is the author of *Rational Unified Process: An Introduction*. His Internet address is pbk@ece.ubc.ca

References

- [1] A. Cockburn, "People and Methodologies in Software Development," in *Faculty of Mathematics and Natural Sciences*. vol. Doctor Philosophiae Oslo: University of Oslo, 2003.
- [2] AgileAlliance, "Manifesto for Agile Software Development," 2001.
- [3] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. El Emam, and J. Rosenberg, "Preliminary guidelines for empirical research in software engineering," *Software Engineering, IEEE Transactions on*, vol. 28, pp. 721-734, 2002.
- [4] Z. Carmen, M. Grigori, and M. Frank, "On the success of empirical studies in the international conference on software engineering," in *Proceeding of the 28th international conference on Software engineering* Shanghai, China: ACM, 2006.
- [5] E. P. Dewayne, A. P. Adam, and G. V. Lawrence, "Empirical studies of software engineering: a roadmap," in *Proceedings of the Conference on The Future of Software Engineering* Limerick, Ireland: ACM, 2000.
- [6] C. B. Seaman, "Qualitative Methods in Empirical Studies of Software Engineering Research," *IEEE Transactions on Software Engineering*, vol. 25, pp. 557-572, July August 1999 1999.
- [7] M. Shaw, "Writing Good Software Engineering Research Papers: Minitutorial," *IEEE Transactions on Software Engineering*, 2003 2003.
- [8] R. L. Glass, I. Vessey, and V. Ramesh, "Research in software engineering: an analysis of the literature," *Information and Software Technology*, vol. 44, pp. 491-506, 2002.
- [9] J. W. Creswell, *Qualitative Inquiry and Research Design: Choosing Among the Five Traditions*. London: Sage Publications, 1997.
- [10] B. G. Glaser and A. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chicago, Illinois: Aldine, 1967.
- [11] A. Strauss and J. Corbin, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Second Edition*: Sage Publications 1998.

- [12] K. Charmaz, *Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis* Thousand Oaks: Sage, 2006.
- [13] W. J. Orlikowski, "CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development," *MIS Quarterly*, vol. 17, pp. 309-340, 09 1993.
- [14] M. Razavi, N. and L. Iverson, "A grounded theory of information sharing behavior in a personal learning space," in *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work Banff*, Alberta, Canada: ACM, 2006.
- [15] B. H. Hansen and K. Kautz, "Grounded Theory Applied - Studying Information Systems Development Methodologies in Practice," 2005, pp. 264b-264b.
- [16] G. Coleman and R. O'Conner, *Software Process In Practice: A Grounded Theory of the Irish Software Industry* vol. 4257/2006: Springer Berlin, 2006.
- [17] R. Baskerville and J. Pries-Heje, "Short cycle time systems development," *Information Systems Journal*, vol. 14, pp. 237-264, 2004.
- [18] E. Whitworth and R. Biddle, "The Social Nature of Agile Teams," in *AGILE 2007*, 2007, pp. 26-36.
- [19] G. Melnik, "Empirical Analysis of Executable Test Driven Development," in *Computer Science*. vol. Doctor of Philosophy Calgary: University of Calgary, 2007.
- [20] D. E. Harter, M. S. Krishnan, and S. A. Slaughter, "Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development," *Management Science*, vol. 46, pp. 451-466, 2000.
- [21] M. S. Krishnan, C. H. Kriebel, S. Kekre, and T. Mukhopadhyay, "An Empirical Analysis of Productivity and Quality in Software Products," *Management Science*, vol. 46, pp. 745-759, 2000.
- [22] M. Diaz and J. Sligo, "How software process improvement helped Motorola," *Software, IEEE*, vol. 14, pp. 75-81, 1997.
- [23] C. Jones, *Software Assessments, Benchmarks, and Best Practices*. Boston: Addison-Wesley, 2000.
- [24] B. Fitzgerald, "An empirical investigation into the adoption of systems development methodologies," *Information & Management*, vol. 34, pp. 317-328, 1998.
- [25] T. Dyba, N. B. Moe, and E. Arisholm, "Measuring software methodology usage: challenges of conceptualization and operationalization," 2005, p. 11 pp.
- [26] Y. Dubinsky and P. Abrahamsson, "Inaugural International Research-in-Progress Workshop on Agile Software Engineering (RWASE '07)," 2007.
- [27] P. H. Becker, "Common Pitfalls in Published Grounded Theory Research," *Qual Health Res*, vol. 3, pp. 254-260, May 1, 1993 1993.
- [28] B. G. Glaser, *Doing Grounded Theory: Issues and Discussions*. Mill Valley, California: Sociology Press, 1998.
- [29] R. S. Schreiber and P. N. Stern, *Using Grounded Theory in Nursing*. New York: Springer Publishing Company, 2001.
- [30] B. G. Glaser, *Theoretical Sensitivity*. Mill Valley, California: Sociology Press, 1978.
- [31] B. G. Glaser, *Emergence vs Forcing: Basics of Grounded Theory Analysis*. Mill Valley, California: Sociology Press, 1992.
- [32] K. M. Eisenhardt, "Building Theories from Case Study Research," *The Academy of Management Review*, vol. 14, pp. 532-550, 10 1989.
- [33] A. M. McCallin, "Designing a grounded theory study: some practicalities," *Nursing in Critical Care*, vol. 8, pp. 203-208, 2003.
- [34] B. G. Glaser and A. Strauss, *Awareness of Dying*. Chicago: Adline 1965.
- [35] E. Whitworth, "Agile Experience: Communication and Collaboration in Agile Software Development Teams," in *Psychology*. vol. Master of Arts Ottawa: Carleton University, 2006.
- [36] B. G. Glaser, "The Constant Comparative Method of Qualitative Analysis," *Social Problems*, vol. 12, pp. 436-445, Spring 1965.
- [37] D. Silverman, "The Quality of Qualitative Health Research: The Open-ended Interview and its Alternatives," *Social Sciences in Health*, vol. 4, pp. 104-118, 1998.
- [38] H. Blumer, *Symbolic Interactionism: Perspective and Method*. University of California Press, 1969.
- [39] M. Q. Patton, *Qualitative Evaluation and Research Methods* 2nd ed. Newbury Park, CA: Sage Publications, 1990.
- [40] C. de Souza, R. B., D. Redmiles, L.-T. Cheng, D. Millen, and J. Patterson, "How a good software practice thwarts collaboration: the multiple roles of APIs in software development," *SIGSOFT Softw. Eng. Notes*, vol. 29, pp. 221-230, 2004.

- [41] J. Morse, "Situating Grounded Theory Within Qualitative Inquiry," in *Using Grounded Theory in Nursing*, R. S. Schreiber and P. N. Stern, Eds. New York: Springer Publishing Company, 2001.
- [42] Y.-H. Jeon, "The application of grounded theory and symbolic interactionism," *Scandinavian Journal of Caring Sciences*, vol. 18, pp. 249-256, 2004.
- [43] J. Corbin and A. Strauss, "Grounded Theory Research: Procedures, Canons, and Evaluative Criteria," *Qualitative Sociology*, vol. 13, p. 19, 1990.
- [44] J. Milliken and R. S. Schreiber, "Can You 'Do' Grounded Theory Without Symbolic Interactionism?," in *Using Grounded Theory in Nursing*, R. S. Schreiber and P. N. Stern, Eds. New York: Springer Publishing Company, 2001.
- [45] B. G. Glaser and J. Holton, "Remodeling Grounded Theory," in *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research in Nursing & Health*. vol. 5, 2004.
- [46] H. S. Wilson and S. A. Hutchinson, "Methodologic Mistakes in Grounded Theory," *Nursing Research*, vol. 45, pp. 122-124, 1996.
- [47] R. Suddaby, "From the Editors: What Grounded Theory Is Not," *Academy of Management Journal*, vol. 49, pp. 633-642, 2006.
- [48] W. A. Hall, "The experience of women returning to work following the birth of their first child," *Midwifery*, vol. 3, pp. 187-195, 1987.
- [49] W. A. Hall and P. Callery, "Enhancing the Rigor of Grounded Theory: Incorporating Reflexivity and Relationality," *Qual Health Res*, vol. 11, pp. 257-272, March 1, 2001 2001.
- [50] A. M. McCallin, "Pluralistic Dialoging: A Theory of Interdisciplinary Teamworking," in *Grounded Theory Review*. vol. 4, 2004.
- [51] M. Sandelowski, "Whatever happened to qualitative description?," *Research in Nursing & Health*, vol. 23, pp. 334-340, 2000.
- [52] M. T. K. Tan and W. Hall, "Beyond Theoretical and Methodological Pluralism in Interpretive IS Research: The Example of Symbolic Interactionist Ethnography," *Communications of the Association for Information Systems*, vol. 19, pp. 589-610, 2007.
- [53] J. Q. Benoliel, "Grounded Theory and Nursing Knowledge," *Qual Health Res*, vol. 6, pp. 406-428, August 1, 1996 1996.