Grounded Theory in Software Engineering Research: A Critical Review and Guidelines

Klaas-Jan Stol
Lero—the Irish Software Research Centre, University of Limerick
Ireland
klaas-jan.stol@lero.ie

Paul Ralph
Department of Computer Science
University of Auckland
New Zealand
paul@paulralph.name

Brian Fitzgerald
Lero—the Irish Software Research Centre, University of Limerick
Ireland
bf@lero.ie

ABSTRACT

Grounded Theory (GT) has proved an extremely useful research approach in several fields including medical sociology, nursing, education and management theory. However, GT is a complex method based on an inductive paradigm that is fundamentally different from the traditional hypothetico-deductive research model. As there are at least three variants of GT, some ostensibly GT research suffers from *method slurring*, where researchers adopt an arbitrary subset of GT practices that are not recognizable as GT. In this paper, we describe the variants of GT and identify the core set of GT practices. We then analyze the use of grounded theory in software engineering. We carefully and systematically selected 98 articles that mention GT, of which 52 explicitly claim to use GT, with the other 46 using GT techniques only. Only 16 articles provide detailed accounts of their research procedures. We offer guidelines to improve the quality of both conducting and reporting GT studies. The latter is an important extension since current GT guidelines in software engineering do not cover the reporting process, despite good reporting being necessary for evaluating a study and informing subsequent research.

CCS Concepts

• General and reference → Empirical studies

Keywords

Grounded theory, software engineering, review, guidelines

1. INTRODUCTION

> *"And make-it-up-as-you-go-along may be OK, but then you have to say, 'I'm making-it-up-as-I-go-along, guys.' "*
>
> *In:* "Dialogue: More on Muddling Methods" [57]

There is growing awareness that Software Engineering (SE) research must consider social, cultural and human aspects of software construction [6, 26, 62]. Scholars have consequently adopted diverse research methods from the social sciences. Qualitative research methods are increasingly employed in SE research as evidenced by journal special issues on their use in 2007 [23] and 2011 [26]. One method that is attracting particular attention is grounded theory (GT) [2, 47]. A quick search in the Scopus database indicates the number of grounded theory studies in computer science has been growing for the last decade (Fig. 1). Early examples of the use of GT in software engineering are by Carver [13] and Coleman and O'Connor [18].

Grounded theory is a method originally described by Glaser and Strauss in their seminal book *The Discovery of Grounded Theory* [38]. The goal of GT is to *generate* theory rather than *test* or *validate* existing theory. GT is suitable for investigating questions such as *what's going on here?* [2].

As a relatively young discipline, SE has yet to establish and validate abundant formal theories. Given the unique and novel aspects of the underlying technology in SE, theories from other disciplines may not be easy to borrow and adapt for SE. Inductive approaches such as GT are therefore useful to construct a relevant conceptual and theoretical foundation for the field.

Since its inception, GT has provided an extremely useful methodological approach in numerous disciplines—notably medical sociology [15], nursing [4], education [58] and management [52]. However, researchers have been criticized for using GT too casually, without clarifying that they have appreciated the intricacies of grounded theory, which is not only quite complex but also based on an inductive paradigm that is entirely different from the traditional hypothetico-deductive model [72]:

> *"'Grounded theory' is often used as rhetorical sleight of hand by authors who are unfamiliar with qualitative research and who wish to avoid close description or illumination of their methods. More disturbing, perhaps, is that it becomes apparent, when one pushes them to describe their methods, that many authors hold some serious misconceptions about grounded theory."*

It is therefore crucial that researchers appropriately design and accurately report studies using inductive methods including grounded theory. However, some software engineering articles that claim to use GT manifest considerable discrepancies between their description of what they have actually done and seminal GT

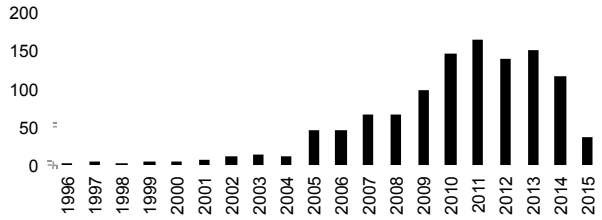

Figure 1. Rise of grounded theory studies in computer science
Source: Scopus (Aug 2015); search string: TITLE-ABS-KEY ("grounded theory"), limited to "computer science"

guidance. To assess the scale of this problem in the software engineering literature, we posed the following research question:

Research Question: *What is the state of practice of grounded theory research in software engineering?*

Several SE researchers have recently reported their experiences using GT and these provide useful guidance for prospective GT researchers [1, 17, 44]. However, this guidance does not extend to *reporting* GT studies. Reporting is important because this produces the persistent record that supports extension and contributes to the field's cumulative body of knowledge. Furthermore, this paper presents a review of almost 100 articles through which we identify a number of key issues with GT studies in SE.

We emphasize that our purpose is *not* to pedantically analyze and criticize the papers included in our study, nor to criticize the authors of those studies in any way. Instead, we draw attention to prevalent misunderstandings of grounded theory as an approach, and contend that only research that embodies GT's core principles (Sec. 2.1) should claim to be a grounded theory study. Based on the results below, we explore numerous considerations for conducting and reporting grounded theory and uncover challenges peculiar to software contexts. Our contribution is consequently fourfold—we provide (1) an in-depth comparison of the three main variants of GT; (2) a critical analysis of the state of practice of the use of grounded theory in the software engineering literature; (3) a set of considerations for conducting and reporting GT studies in SE; (4) three significant challenges for applying GT to software engineering phenomena.

This paper proceeds as follows. Section 2 presents a brief history of grounded theory (including its terminology and philosophical foundations) and a comparison of the different GT versions. Section 3 presents the research design that we employed. Section 4 presents the analysis and results of our study. Section 5 discusses the results and offers a checklist for future GT studies. Section 6 concludes the paper.

2. GROUNDED THEORY

2.1 Key Components of Grounded Theory

Grounded Theory refers to a method of inductively generating theory from data [38]. GT studies often focus on unstructured text (e.g. interview transcripts, documents, field notes); however, they may also include structured text, diagrams and images, and even quantitative data [35].

For the presentation of our analysis in Sec. 4, it is imperative to establish the key components of grounded theory. While GT has several variants (discussed in Sec. 2.3) they share many core features, including the following:

- **Limit exposure to literature**. Rather than beginning with a comprehensive literature review, grounded theory proponents (e.g. [19, 35]) recommend limiting exposure to existing literature and theories to promote open-mindedness and preempt confirmation bias (see Sec. 2.3 for different positions regarding the literature). A major reason to limit study of the literature is to prevent the researcher from testing existing theories, or thinking in terms of established concepts.
- **Treat everything as data.** When Glaser says, "all is data," he means *all*—qualitative data, quantitative data, semi-structured data, pictures, diagrams, videos and even existing theories and literature [36, 69].
- **Immediate and continuous data analysis**. The researcher begins analyzing data immediately and does not finish collecting data before beginning analysis—data collection and analysis are simultaneous [16], and subsequent data collection is driven by *theoretical sampling,* discussed next.
- **Theoretical sampling**. The researcher identifies further data sources based on gaps in the emerging theory or to further explore unsaturated concepts. Theoretical sampling is indeterministic, as opposed to conventional sampling techniques [16] (see *theoretical saturation* below).
- **Theoretical sensitivity**, which refers to the researcher's ability to conceptualize, and to establish relationships between concepts, lies at the heart of developing grounded theory. Both Glaser [35] (Ch. 5) and Strauss and Corbin [68] highlight the role of creativity in this process.
- **Coding**. The researcher uses inductive and abductive logic to construct analytical codes and infer theoretical categories from the data by labeling 'incidents' and their properties. The researcher does *not* classify data into a preconceived coding scheme, or infer categories from logically deduced hypotheses [16]. Glaser and Strauss [38] did not use the term abduction but emphasized induction to distance themselves from the deductive theorizing that was prevalent at the time of their publication. Both Glaser and Strauss later admitted a role for deduction in GT [36, 70].
- **Memoing**. The researcher writes memos (e.g. notes, diagrams, sketches) to elaborate categories as they emerge, describe preliminary properties and relationships between categories, and identify gaps [16]. These memos play such an important role in theory generation that Glaser baldly stated that, "*if the researcher skips this stage, he is* ***not*** *doing grounded theory*" [34] (Ch. 5, emphasis original).
- **Constant comparison**. From the start of the study, the researcher constantly compares data, memos, codes and categories [8]. Both categories and data interpretations evolve and saturate until they 'fit' the data [34].
- **Memo sorting,** also called **theoretical sorting** is the continuous process of oscillating between the memos and the emerging theory outline to find a suitable fit for all categories that resulted from the coding [34, 70]. Like memoing, Glaser argues that sorting cannot be skipped [35].
- **Cohesive theory**. The researcher attempts to move beyond superficial categories and develop a cohesive theory of the studied phenomenon.
- **Theoretical saturation**. The researcher stops collecting and analyzing data when theoretical saturation is reached. *Theoretical saturation* refers to the point at which a theory's components are well supported and new data is no longer triggering revisions or reinterpretations of the theory [34].

While this list of core features is by no means a complete description of grounded theory (both Glaser and Strauss have written numerous books to explain GT [34-36, 71]), it does highlight some distinctions from more traditional, deductive research methods. The above, however, largely ignores the differences between the various versions of GT, discussed in Section 2.3.

2.2 Philosophical Foundations

Research approaches are commonly (and simplistically) classified into two broad groups based on the epistemological positions of positivism and interpretivism [40]. GT can be confusing because it does not fit cleanly into either group. We briefly discuss ontology and epistemology, and then focus on how GT resists the classification of positivism and interpretivism.

The positivist approach has long been applied in the physical sciences, and has led to tremendous growth of knowledge in the area. It comprises five pillars [43]:

- **Unity of the scientific method**: the same approach to knowledge acquisition applies to all forms of enquiry.
- **Search for causal relationships**: science aims to find regularity and causal relationships among studied elements.
- **Belief in empiricism**: sense-experience is the only source of knowledge but subjective perception is not acceptable.
- **Science (and its process) is value-free**: science has no intrinsic values or perspectives; science is independent of politics, ideology, morality, society and culture.
- **Science is founded upon logic and mathematics**: causal relationships are demonstrated quantitatively, using the universal language of math and the formal basis of logic.

Positivism assumes that: (1) the universe behaves according to inalterable, discoverable laws; (2) systems are merely the sum of their components (reductionism); (3) science should be reproducible, reliable, rigorous and objective. Different scientists observing the same phenomenon should therefore reach equivalent conclusions.

Interpretivism makes the opposite assumptions (cf. [41]): (1) no universal truth or reality exists, rather, *"the important reality is what people imagine it to be"* [9]; (2) systems exhibit emergent behaviors not reducible to their component parts [33, 48]; (3) social science, which aims to understand and to interpret human behavior, is fundamentally different from natural science, and natural science methods including quantitative measurement, statistical significance and hypothesis testing are insufficient for understanding social phenomena [76]. Therefore, formulating hypotheses is not relevant to an interpretivist study. Understanding and explaining the social world requires emotion and empathy, which preclude pure objectivity [76]. Interpretivists have attacked positivism for promoting the myth of objectivity [56] and Berger and Kellner point out that *"direct access to facts and laws ... is never possible, no matter what one's standpoint ... there is no magic trick by which one can bypass the act of interpretation"* [5]. Interpretivists prefer qualitative methods, including interviews, case studies, ethnography and action research, arguing that these keep the researcher grounded in *"the first-order, primary, lived concepts of everyday life"* [21].

While positivism and interpretivism can be cast as polar opposites [31], many studies do not sit neatly in either paradigm. We have experiments where the dependent variable is 'measured' by combining the subjective ratings of expert judges [55], case studies with upfront hypotheses [63], interview studies where text is analyzed quantitatively [61] and mixed-method inquiries that combine questionnaires with case studies [60]. *"All qualitative data can be coded quantitatively"* by counting words and categorizing statements; meanwhile *"all quantitative data is based on qualitative judgment"* because we have to make assumptions to interpret the numbers [73]. More fundamentally, these groups involve several interconnected philosophical positions that cannot be reduced to a single spectrum, let alone a Boolean variable.

It is easy to mistake GT as a qualitative or interpretivist method because many GT studies focus on collecting and analyzing unstructured text. However, GT was developed in the 1960s, during the ontological and epistemological shift from positivism and objectivism to social constructionism and postmodernism. GT stems from a dissatisfaction with the way research was done, whereby new researchers were trained as "theoretical serfs" who tested the theories of "theoretical capitalists" [34] (p. 9), which could lack relevance to the real world. GT was developed due to a desire to build theories more rigorously and dispassionately by grounding them in objective reality.

2.3 Versions of Grounded Theory

Although Glaser and Strauss never explicated their epistemological position in *Discovery* (and Glaser later argued that GT is paradigm-agnostic), their terminology reflects an objectivist stance. The title of their seminal book is 'The *Discovery* of Grounded Theory' [38] rather than for example *Sensemaking* with grounded theory—the term *discovery* suggests that an objective reality exists *out there* waiting to be discovered. Glaser speaks of an indicator-concept model, analysis of a core variable, and aims for parsimony in the developed theory, reflecting a position that aligns with objectivism. Both Glaser and Strauss and Corbin also used objectivist terminology in their definitions of theory as a set of concepts and relationships among them that together offer explanations and predictions (i.e. causality) [34, 68]. While Glaser maintains that GT is independent from any philosophical stance, Corbin has gradually shifted towards interpretivism [19]. Meanwhile, Charmaz (a student of Glaser), developed 'constructivist' grounded theory by reinterpreting GT from a constructivist's stance [16] that is closely connected to interpretivism.

Due to extensive discussions on what constitutes grounded theory, it has been labeled a 'contested concept' [11]. Since Glaser and Strauss's seminal book, GT has seen considerable evolution resulting in the emergence of different versions. Denzin lists no less than seven different versions [22], although he does not specify the differences between all of them. It is now widely acknowledged that there are at least three main streams of GT [1]: Glaser's GT (*classic* or *Glaserian* GT); Strauss and Corbin's GT (*Straussian* GT); and Charmaz's *constructivist* GT.

Glaser's perspective is well reflected in the fact that he refers to his version of grounded theory as "classic" GT. He strongly disagrees with Strauss and Corbin's version of GT [35] and has argued that Strauss and Corbin's method is *not* grounded theory, but refers to it as "full conceptual description" [35]. Furthermore, Glaser has called 'constructivist' grounded theory a "misnomer" [37]. In this paper, we accept any version of grounded theory as 'grounded theory'—although we will argue below that consistency with a particular version is important. Table 1 summarizes some of the key differences between the three main strands of GT. An additional difficulty in comparing GT versions is that Straussian GT is still evolving, as briefly mentioned above.

Of the three main versions of GT, the difference between classic and Straussian GT has been discussed most extensively [10, 42, 51]. Classic GT can be characterized as having a strong focus on *emergence* (of research questions, of codes, of theory), whereas Straussian GT meticulously suggests a set of 'mini-steps.' This difference in focus on emergence is captured succinctly by Stern: *"Strauss, as he examines the data**, stops at each word to ask**, '**What if**?' Glaser keeps his attention **focused on the data** and asks, 'what do we have **here**?'"* [64] (our emphasis). Glaser requires *any* concept to be *grounded in the data*, whereas Strauss and Corbin *go beyond the data* by asking various questions on *what might be* to develop the emerging theory [35] (Ch. 8). Strauss's approach has been described as *"more free-wheeling flights of imagination,"* which contrasts strongly with Glaser's faithfulness to the data.

There is little agreement on what constitutes theory. In classic GT, theory consists of concepts that are related to one another, offering explanation and prediction. Constructivist GT emphasizes understanding and acknowledges that data, interpretations, and resulting theory depend on the researcher's view. In practice, however, such ontological and epistemological differences are rarely apparent in generated theories.

Table 1. Some of the key differences between the three main strands of grounded theory

Element	Classic / Glaserian grounded theory	Straussian grounded theory	Constructivist grounded theory
Research question	Should not be defined a priori, but emerge from the research—this makes the RQ *relevant* to the field. The researcher starts with an 'area of interest.' Literature in other areas may be consulted to increase the researcher's "theoretical sensitivity." Defining a RQ a priori is considered 'forcing' [35].	Research question may be defined upfront, derived from the literature or suggested by a colleague; RQ is often broad and open-ended.	Research begins with *"initial research questions,"* which evolve throughout the study [16].
Role of the literature	An extensive literature review should be delayed until after the theory is emerging to prevent the influence of existing concepts on the emerging theory. Until the researcher has defined the RQ, it is not clear *which* literature should be consulted. Existing concepts such as gender and age should not be included a priori, but must 'earn' their way into the emerging theory.	The literature may be consulted throughout the process, as concepts from the literature may be used if applicable; to enhance theoretical sensitivity, as a secondary data source; to formulate questions for data collection or stimulate questions during analysis; to suggest areas for theoretical sampling [70] (p. 49).	Acknowledges not only Glaser's reasons for delaying the literature review but also the impracticality of this strategy. Charmaz highlights the need to tailor a literature review to fit the purpose of the GT study [16] (p. 306).
Coding procedures	***Open coding***: 'fracturing' of the data; line by line coding is recommended to achieve full theoretical coverage, but does not reject coding sentences or paragraphs, or whole documents [35]. ***Selective coding***: delimiting coding to only those variables that relate to one (or in some cases, several) core variables to establish a parsimonious theory. The core variable guides further data collection. ***Theoretical coding***: establishing conceptual relations between substantive codes, resulting in the development of hypotheses. Glaser proposes several 'coding families,' which are theoretical codes that can be used by researchers, though these must 'earn' their way into the emerging theory (e.g. the Six C family in Fig. 4).	***Open coding:*** generation of 'categories' and how they vary dimensionally. Coding can be done line by line or by sentence or paragraph, or even the whole document [70]. ***Axial coding:*** putting back data in new ways after open coding by identifying relationships between categories; this is effectively Glaser's *theoretical coding*. Use of the 'paradigm model' or 'conditional matrix' (an analytical tool in Straussian GT [70], Ch. 12) to identify context, conditions, action / interaction strategies and consequences. ***Selective coding***: deciding on the central category that all major categories can link to [70].	***Initial coding:*** examining data word-by-word, line-by-line or incident-by-incident to make sense of the text without injecting the researcher's assumptions, biases, motivations. Similar to Glaser's open coding. Charmaz recommends "coding with gerunds." ***Focused coding:*** selecting categories from the most frequent or important codes, and using them to categorize the data; does not require a single core category or variable. ***Theoretical coding***: specifying the relationship between categories to integrate them into a cohesive theory.
Questions asked during analysis	• *What is this data a study of?* [34] • What category or what property of what category does this incident indicate? • What is actually happening in the data?	Asking questions about whom, when, where, how, with what consequences, and under what conditions phenomena occur, helps to 'discover' important ideas for the theory [69]. *'Free-wheeling flights of imagination'* [16]	• *What is this data a study of?* [16] • What do the data suggest? Pronounce? Leave unsaid? • From whose point of view? • What theoretical category does this specific datum indicate? [16]
Philosophical influences	**Objectivism**: There exists a single, correct description of reality; the researcher therefore *discovers* grounded theory from data [11].	**Pragmatism** and **symbolic interactionism**: actors engage in a world that requires reflexive interaction; reality is *constructed* through interaction and relies on language and communication [14].	**Social constructionism**: social reality is *constructed* by our individual and collective action. GT emerges from "shared experiences and relationships with participants"; Observers are not neutral [16].
Evaluation criteria	The generated categories must **fit** the data, the theory should **work** (it must be able to explain or predict what will happen); the theory must have **relevance** to the action of the area, and the theory must be **modifiable** as new data appear [34] (p. 4-5).	Seven criteria for the research process e.g. information on sample selection, major categories, derived hypotheses and discrepancies. Eight criteria regarding the empirical grounding, e.g. "are concepts generated?" "is variation built into the theory?" [70].	**Credibility** (e.g. is there *sufficient* data to merit claims?), **originality** (do your categories offer *new* insights?), **resonance** (does the GT make *sense* to participants), **usefulness** (does the GT offer *useful* interpretations?) [16] (p. 337).

While the 1998 edition of Strauss and Corbin's book specifies open, axial and selective coding, the 2008 edition (authored by Corbin alone after Strauss' death in 1996, making the term *Straussian* GT a misnomer and *Corbinian* more appropriate) no longer defines open and axial coding as separate activities [74]. This paper focuses on the 1998 version since it is very prevalent (in particular *axial coding*). As Table 1 shows, the three variants differ in their position with respect to key elements such as the role of the literature, but also in terminology and order of practices (e.g. coding procedures). For example, Strauss and Corbin interpret *selective coding* differently from Glaser. Furthermore, Strauss increasingly saw GT as a *verificational* method [16], a position that Glaser strongly rejects [35].

3. RESEARCH DESIGN

3.1 Study Identification and Selection

To investigate the state of practice of GT research in SE, we reviewed a selection of articles reporting GT studies. We adopted an automated search strategy; that is, we collected our sample by searching specific online databases using specific search strings (see below). We chose this over manually browsing selected publication outlets because it is more efficient and replicable. We pilot tested several search strings. For example, we conducted a search on "grounded theory," but this resulted in thousands of papers from other disciplines. We also tried limiting the search to the title, abstract and keywords, but some GT studies appear not to use the term 'grounded theory' in any of these fields. Based on this pilot test, we adopted the following query.

Search String: "grounded theory" AND "software engineering"

We searched Scopus, IEEE Xplore, the ACM Digital Library and ScienceDirect. We excluded Wiley Online and SpringerLink, as these are subsumed by Scopus. We adapted the search string to the specific characteristics of each database. Further constraints were introduced case-by-case to eliminate obviously irrelevant papers. Combining the search results and removing duplicates produced an initial dataset of 1,763 papers (Table 2). As this dataset is too large for manual analysis, we focused on articles published in well-known, peer-reviewed SE journals (Table 3).

We did not consider conference contributions because journal papers tend to have endured greater review, be more polished and have more liberal page limits. We also did not consider articles from peer-reviewed magazines including *Communications of the ACM* and *IEEE Software* because they tend to have briefer methodological descriptions, given their practitioner-oriented focus. In the interests of representativeness, we further excluded specialist journals such as *Requirements Engineering* and the *International Journal of Open Source Software and Processes*.

The selected journals coincide with those used in previous reviews (e.g. [39], except magazines as stated). We further added the *Software Quality Journal* and the journals that descended from the *Journal on Software Maintenance: Research and Practice*.

Table 2. Searched databases and search constraints

Database	Search constraints	No.
Scopus	N/A (full text)	1,668
ScienceDirect	Computer Science only (full text)	249
IEEE Xplore	Search on metadata only	73
ACM DL	Title, Abstract, Keywords only	13
	Subtotal	2,003
	Duplicates	240
	Total	**1,763**

Table 3. Selected journals and number of papers included

Journal	Articles
Information and Software Technology	42
Journal of Systems and Software	16
IEEE Transactions on Software Engineering	11
Empirical Software Engineering	10
Software Process: Improvement and Practice [a]	8
Journal of Software: Evolution and Process [b]	4
Software Quality Journal	3
ACM Trans. Software Engineering and Methodology	3
Journal of Software Maintenance and Evolution: Research and Practice [c]	1

[a] *Merged with J Software: Evolution and Process in 2012*
[b] *Successor of J Softw Maint Evol Research & Practice since vol. 24, 2012*
[c] *Vol. 1-12 published as J Software Maintenance: Research and Practice*

We removed editorials, secondary studies (systematic reviews), and articles that present methodological reflections on the use of GT, rather than a specific GT study (e.g. [1, 12, 18, 46, 59]), resulting in a final set of 98 papers (available in an appendix [67]). Fig. 2 shows the articles' distribution of publication year.

3.2 Data Extraction

We read all 98 papers to investigate the following questions.

- What is claimed concerning the use of grounded theory? (e.g. "we used grounded theory," "we took a grounded theory approach," "the data were coding using GT techniques");
- To what extent are different versions of grounded theory discussed and used? To what extent do papers state their epistemological stance?
- Is grounded theory mentioned in the title, keywords, abstract, or research question (or objective / topic / purpose)?
- What specific GT techniques and practices are used? (e.g. open coding, constant comparison, memoing);
- How is data collected and analyzed?
- What do GT studies produce and how do they present it? (e.g. as a diagram);
- Was the literature review (if any) conducted before, during or after the study; was the resulting theory (if any) integrated back into the literature?

All information was recorded in a spreadsheet. We also took extensive notes concerning interesting findings that did not fit in our predefined questions. In several studies, for example, we noted clear deviations from GT principles, such as the use of (preconceived) 'seed categories' to guide initial analysis which is viewed as inappropriate in GT. The data extraction and coding was done by the primary author, which was reviewed by the remaining authors.

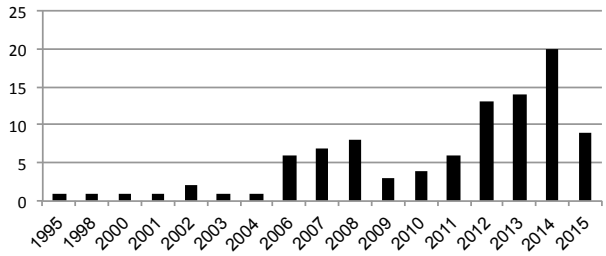

Figure 2. Distribution of publication year of selected articles

Note: Search conducted in Spring 2015, hence the drop in 2015.

4. ANALYSIS AND RESULTS

In this section we address the use of GT, the level of detail presented, variants of GT and the type of output of studies.

4.1 Grounded Theory "Use" is Ambiguous

We analyzed all 98 articles to investigate their claim of using grounded theory, and found that many claims are quite ambiguous. Fig. 3 (Box 1) shows that almost half (*n*=46) of the surveyed articles (*n*=98) merely borrow from grounded theory; for example:

- "Using concepts of grounded theory […]"
- "data analysis was carried out using a modified version of Grounded Theory"

Fifteen articles (Box 1.1 in Fig. 3) state that they use an approach that *resembles*, *adapts*, or *is inspired by* grounded theory, but do not in actual fact present a grounded theory study. An example of such a claim is: *"In a method similar to the first step in grounded theory (Glaser and Strauss 1967) […] we identified a set of categories."* Such studies are clearly not grounded theory studies.

Eighteen articles (Box 1.2 in Fig. 3) do not use the term 'grounded theory' in the main text at all (but only in its bibliography). Rather, they mention specific techniques such as 'coding' or 'theoretical saturation' and cite seminal works on grounded theory, such as Glaser and Strauss's *Discovery* book [38]. Thirteen other articles (Box 1.3) state that they use grounded theory 'techniques' or 'procedures,' and in most cases refer to coding and constant comparison. One example of such a statement is: *"The 'Open Coding' and 'Theoretical Coding' techniques of Glaser (1978) have been applied iteratively to identify different categories and their properties."* Such statements do not claim that GT was used, merely GT techniques. In several cases, authors explicitly acknowledge that their study is not a GT study.

This borrowing rhetoric is unusual in research methodology. We do not recall ever reading about studies that *"use randomized controlled trial techniques,"* were *"inspired by survey methodology,"* or *"adopted a modified questionnaire approach."* Claiming to *"use grounded theory techniques"* rather than GT wholly suggests that authors are aware that GT is a comprehensive research method from which they are borrowing certain elements.

The remaining 52 papers (Box 2 in Fig. 3) explicitly claim to use GT. Typical examples of such claims include (e.g. [30]):

- "Using a grounded theory approach […]"
- "We used grounded theory to […]"
- "We generated a grounded theory"

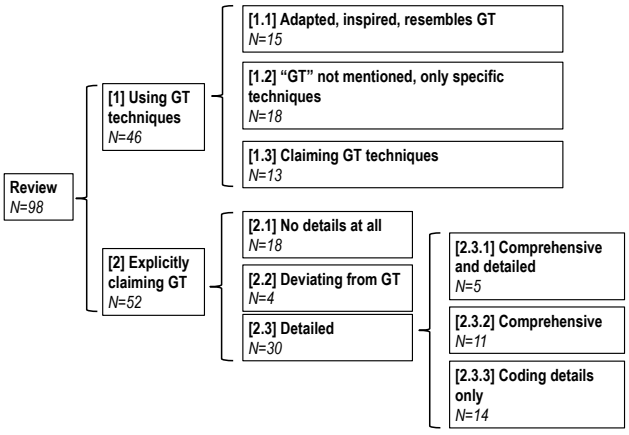

Figure 3. Breakdown of the articles included in our review

However, deciding whether or not a study uses grounded theory is far from trivial. While some articles clearly claim to use grounded theory, the phrasing of these claims varies substantially and some are ambiguous. For example, some studies use a "grounded theory approach." In the absence of further clarification, we assume this means GT was used, however, it could be interpreted as an approach based on GT. This made it more difficult to decide whether or not the authors were actually claiming to use grounded theory. This is simultaneously a potential threat to the validity of our findings and a surprising finding itself. While our exact count (52 studies making claims to use GT) should be interpreted with caution, the fact that this is ambiguous, and any large proportion of studies *borrowing* from a method rather than *using* it, is unusual and potentially problematic for a sound evaluation of such a study.

Of the 52 studies making a claim to have used GT, four studies (Box 2.1) deviated so sharply from GT that they have not used grounded theory at all. In three cases, the authors developed a set of preliminary categories, which were then combined with a "grounded theory approach"—starting with a classification from the literature is highly suspect, even when considering Strauss and Corbin's quite liberal use of the literature (see Table 1).

Of the 98 articles included in our review, six used the term 'grounded theory' in the title and 14 specified 'grounded theory' as a keyword. This suggests that grounded theory was essential to these studies rather than an afterthought. While clearly no conclusion should be drawn based on the presence of GT as a keyword, given the limitations of some journals on the number of keywords (as low as three), it might suggest that these authors more consciously wished to signal the role of GT in their study.

4.2 Many Studies Present Little to No Detail

Of the 52 articles claiming to have done a grounded theory study, 18 (Box 2.1 in Fig. 3) present no details *at all* beyond claims such as: *"[we] used a grounded theory approach for data gathering and data analysis."* In some cases, a brief and usually incomplete summary of grounded theory is provided, for example, by stating that grounded theory consists of three coding phases. Besides being incomplete, it also suggests coding happens in three distinct phases, which is not what Glaser or Strauss had in mind. Many of these articles state that the conceptualization presented in those articles were developed using grounded theory, without shedding any light on the process through which this was done.

We further inspected the 30 articles (Box 2.3) that present significant methodological details, to investigate the extent to which different GT practices are mentioned and used (Table 4). While GT is not reducible to a set of independent practices, one still expects GT studies to report details on key practices associated with GT (cf. Sec. 2.1).

However, many authors use GT techniques *à la carte*. Fewer than half of the 30 articles describe or confirm the use of key practices, such as simultaneous data collection and analysis (*n*=13), memoing (*n*=12), memo sorting (*n*=4), constant comparison (*n*=13), or theoretical sampling (*n*=12). Fifteen articles confirm that data collection continued until theoretical saturation was reached. All but one article discuss data sources, elucidate data collection and describe coding practices. Details varied from a brief paragraph to an extensive presentation. We also found misinterpretations of key practices. One article claimed theoretical sampling, but instead of collecting additional data to further investigate as of yet unsaturated concepts in the emerging theory, a number of case companies were selected seemingly a priori based on their experience in the area that the researchers were investigating.

Table 4. Grounded theory practices used GT (*n*=52)

Practice	Papers reporting	
GT Practice details reported		30
Simultaneous data collection and analysis	13	
Data sources and collection	29	
Theoretical sampling	12	
Coding	29	
Memoing	12	
Memo sorting	4	
Constant comparison	13	
Theoretical saturation	15	

Sixteen articles (Boxes 2.3.1, 2.3.2) provide a comprehensive presentation of their research method, of which five articles present extensive documentation about the GT research process [2, 18, 45, 47, 49]. Fourteen other articles provided details on the coding process only (Box 2.3.3).

4.3 Many Studies Ignore GT Variants

As discussed in Sec. 2.3, GT has several variants with significant differences with regards to the use of the literature, specific coding practices, and reflections on the role of the researcher in the research process. Of the 52 articles that claim to use grounded theory, 39 did *not* acknowledge the existence of different variants.

To investigate which sources authors might have consulted in their study design, we looked at the citations to seminal GT works. Of the 39 articles that do not claim a specific GT variant, 10 cited works on classic GT (Glaser, Glaser & Strauss), 13 cited works on Straussian GT (Strauss, Strauss & Corbin), and none cited constructivist GT. Thirteen articles cite conflicting seminal works on GT (e.g. [16, 19, 35]) without acknowledging any differences or indicating whose guidance they are following. Two articles cite works on all three variants of GT. One interpretation of this is that authors are now aware of the differences, and, in seeking to confer legitimacy on their research, provide copious references to several seminal works. However, we would argue that, had the authors actually read all three works, the existence of different variants would have been likely acknowledged.

In several cases we found inconsistent usage of the claimed variant of grounded theory. Two articles claim or cite classic GT but use *axial coding*, a Straussian practice (Sec. 2.3). Another article claims to use Straussian GT, but uses one of the *coding families* offered by Glaser for increasing theoretical sensitivity [34] (p. 74).

Table 5. Grounded theory variants acknowledged (*n*=52)

Grounded theory variant claimed	Articles	
Acknowledgment of different GT variants		13
Explicit claim classic GT	5	
Explicit claim Straussian GT	8	
Explicit claim constructivist GT	0	
Variants not acknowledged		39
Citing classic (Glaser / Glaser & Strauss)	10	
Citing Straussian (Strauss / Strauss & Corbin)	13	
Citing constructivist (Charmaz)	0	
Citing a combination of the above	13	
Citing others	3	
Epistemology acknowledged		5
Interpretivist or constructivist	5	
Other	0	

Three articles do not refer to any of the seminal texts on GT but refer to other sources. These may be innocent mistakes or benevolent simplifications. Alternatively, and more worryingly, they may indicate researchers who are presenting their research under the guise of techniques they have heard of, but not investigated.

Thirteen papers, however, do acknowledge the distinction between classic and Straussian GT—some in more detail than others. For example, one article stated that it incorporated *"a Strauss and Corbin grounded theory approach to data gathering and analysis,"* whereas other articles laid out the differences between the variants in detail. Of these, five claim to use classic GT, the other eight Straussian GT. None of the articles in our sample explicitly claim to use Charmaz's constructivist GT. Finally, only five articles state an epistemological position; in all cases the authors claim their study to be an interpretivist one. In four of those cases, reference was made to seminal works by Glaser, and Strauss and Corbin, which align more closely with positivism, as outlined in Sec. 2.

4.4 Few "GT" Studies Generate Theory

Since grounded theory is a method of generating theory, we investigated the extent to which the 52 studies claiming to have used GT developed theories. While it depends on one's definition of theory, few of the studies appear (or claim) to develop a theory, even though "a lack of existing theories" in a particular area is often given as a motivation to conduct a GT study.

Eight articles presented contributions that were clearly cohesive theories consisting of constructs and relationships, while a ninth article presented a set of hypotheses that could be considered a theory. Some of the topics theorized by these studies include:

- How is the software development process managed?
- How do software processes form and evolve?
- How do self-organizing agile teams self-organize?

Some articles present 'theory' in alternative forms instead of a set of concepts and relationships. For example, Hoda et al. [47] presented six roles that members of agile teams assume. Together these roles provide an explanation for the "social" process of self-organization in agile teams, and as such they go beyond a mere taxonomy of roles. Therefore, we argue such a coherent set of findings can be considered a theory.

In most cases, articles presented a graphical representation of the theory, usually simple boxes-and-arrows diagrams, to illustrate theoretical concepts and relationships. Three articles use Glaser's 'Six C' coding family [34] (p. 74) for visualization (e.g. Fig. 4). Other articles synthesized their results into various other types of contributions, including:

- **Conceptual frameworks**, such as a framework of factors that influence Software Process Improvement initiatives [29];
- **Conceptual models**, such as a model of the process for managing collaborations in open source [7];
- **A set of factors**, such as success factors for globally-distributed XP projects [53];
- **A set of themes or categories,** such as a set of categories representing the characteristics of product managers [54].

Such contributions are useful as they offer new foundations for empirical studies, but often they do not form a 'theory' that, in Glaser's words, *"account for a pattern of behavior."* We observe that studies that produce a 'set of themes' rather than a theory tend only to borrow discrete practices from GT—what we call grounded theory *à la carte*.

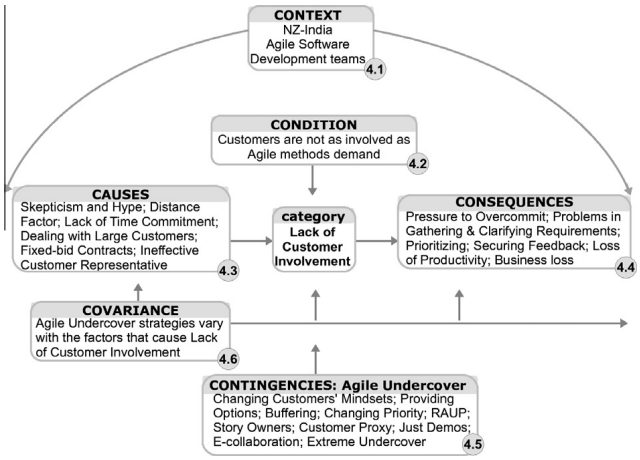

Figure 4. Example of the Six C coding family (from: [45])

Finally, ten articles present mere description. In many cases, the study's results are structured based on a set of research questions, which are answered in detail using quotes from participants. This type of output is quite common for those studies that only used coding techniques, but do not make a theoretical contribution.

5. DISCUSSION

A significant number of articles in our sample did not provide sufficient details for reviewers or readers to evaluate their methodological rigor. Several factors may contribute to a lack of methodological detail, including space constraints and simply not knowing what details to report. Since we only analyzed journal articles (rather than conference papers), space constraints are less valid as an excuse. Most seminal works on GT focus on how to collect and analyze data rather than what details to give in the methodology section of a paper. We therefore provide some general advice for reporting grounded theory studies followed by a list of specific details to include.

5.1 Method Slurring

Several SE articles claim to use grounded theory, yet do not appear to embrace its core characteristics (Sec. 2.2). If a study does not involve simultaneous data collection and analysis, constant comparison, coding, memoing and theory development, it is not a grounded theory study. If researchers collect most or all of their data before beginning analysis, collect or categorize data according to existing theory, base analysis on seed categories or preconceived analytical frameworks, they are not using grounded theory.

Claiming to use a research method without actually following its guidelines is referred to as "method slurring" [3]. Based on other authors' and our own observations, we suggest researchers might commit method slurring for at least five reasons:

1. **To confer legitimacy**. Grounded theory is more structured (and is therefore often perceived as more scientific) than other methods of building theories from primarily unstructured data. Charmaz lamented that *"Numerous researchers have invoked grounded theory as a methodological rationale to justify conducting qualitative research rather than adopting its guidelines to inform their studies"* [16].
2. **To avoid detailed and exhaustive literature review and initial conceptualization**: Many researchers may readily embrace the grounded theory maxim of avoiding becoming too familiar with the relevant literature, to excuse skipping necessary background work [72].
3. **For simplicity**. It is easier to state *"we used grounded theory"* than to thoroughly explain how a researcher converted a large amount of unstructured text into a cohesive theory. However, given that grounded theory is not widely understood (misunderstood, even) or known amongst SE researchers, we argue that such claim does not suffice.
4. **Because they simply do not understand GT or its relationship to other research methods.** Suddaby notes that *"researchers claim to have performed grounded theory research, support their claims with a cursory citation to Glaser and Strauss (1967),"* while offering little description of the applied method. When authors are invited to elaborate, Suddaby continues, *"to reveal how the data were collected and analyzed, it becomes clear that the term 'grounded theory' was interpreted to mean 'anything goes' "* [72].
5. **Per referee's suggestion.** We know of cases where referees have suggested to authors that the method they used "looks like grounded theory." Such authors may post-hoc present their research as grounded theory where such a claim is not valid, simply to satisfy reviewers.

Method slurring undermines grounded theory. Authors in the management literature have observed an *"overly generic use of the term 'grounded theory' "* [72]. Researchers in Information Systems (which has considerable overlap with software engineering) have lamented that *"the term 'grounded theory' itself has almost become a blanket term for a way of coding data"* [75]. Others have referred to the *"erosion of GT as a research method"* [25, 64]. Using the term 'grounded theory' to denote any kind of theory building or qualitative data analysis undermines the legitimacy of GT, which prescribes a highly structured analytical approach. This engenders undue suspicion of GT studies, possibly hindering publication.

Similarly, it undermines other qualitative methods. Grounded Theory is not the only valid method of either analyzing predominately qualitative data or generating theories. Numerous other qualitative methods exist [20]. Recasting interpretive interview studies, positivist case studies and ethnographies as grounded theory implicitly disparages and devalues these legitimate research approaches. There is nothing wrong with conducting an ethnography, for example, and researchers should not be hesitant to label their research as such. Theories can also be developed based on intuition and experience, or by extending and synthesizing existing research.

Furthermore, method slurring misrepresents the current research. A key principle of science communication is accurately describing how data was collected and analyzed [32]. This allows reviewers and readers to evaluate the quality of a study. If a study claims to have used GT while actually doing something different, it violates this principle.

Because so many GT articles lack methodological detail, it is difficult for readers to assess whether studies actually use grounded theory or simply reference grounded theory *"as a methodological rationale"* [16].

5.2 Considerations for Conducting and Reporting Grounded Theory

Individual researchers will have their own styles and preferences for conducting and reporting their studies. However, to avoid method slurring (among other problems) we offer four broad recommendations.

Firstly, it is important to study grounded theory before starting. As several authors have noted, grounded theory may suffer from its 'apparent simplicity' [31]. Superficially, GT appears to involve simply reading and categorizing some text. However, a key challenge in GT is that of theoretical sensitivity: a researcher's capability to develop useful and interesting concepts that contribute to a theory. Furthermore, GT is a complicated research method with multiple variants and conflicting guidance. Many overviews and guiding literature for SE researchers do not even include grounded theory (cf. [27, 77]). Anyone considering a GT study should read several books before even deciding whether GT is the right method, let alone beginning data collection. Good introductions are available for classic GT [34, 38], Straussian GT [19], and constructivist GT [16]. Our review contains numerous exemplars (e.g. [2, 18, 45, 47, 49]), which may be consulted. GT should be considered from the conception of a study as it differs in quite significant ways from traditional studies as outlined in Sec. 2. Research cannot be reconstructed as GT at write-up.

Secondly, researchers should describe *their implementation* of GT, not GT in principle. Some studies in our sample provided quite reasonable summaries of GT, but did not explain *their* practices, deviations or precisely what *they* did. Because GT is relatively new to SE, and to avoid method slurring, it is crucial to explain exactly what was done in the study at hand. In particular, we recommend explicitly describing how key practices (e.g. simultaneous data collection and analysis, constant comparison, memoing) were used. We also recommend explicitly describing deviations from GT guidelines.

Thirdly, researchers should avoid 'borrowing' rhetoric. If techniques have been borrowed from the grounded theory literature, researchers should simply state that those techniques have been used without discussing GT. Practices including coding, memoing and constant comparison are all part of the contemporary qualitative data analyst's toolbox. They can exist on their own, independent of their proponents or any particular research method. Bringing in GT clouds the issue.

Finally, and related to the previous point, researchers should not claim to have used grounded theory when they have not. Researchers should describe how *they* analyzed data or generated theory. If using another method it should be named. If a researcher has developed his/her own method, it should be explained. If a researcher has proceeded ad hoc, such a "pragmatic, agile approach" should be explained rather than dressing it up as grounded theory. To be clear, we accept any variant of GT as grounded theory, in contrast to Glaser who only recognizes 'classic' GT (as described in the 'Discovery' book) and considers Straussian GT not to be GT [35] (p. 123).

We further provide an extensive list of considerations for grounded theory in software engineering (Fig. 5), which include a variety of potentially relevant issues for consideration when conducting or evaluating a GT study. The items in Fig. 5 may be especially useful for novices writing their first GT study, experts who need to jog their memories for methodological dimensions to address, or anyone who struggles to explain how they collected and analyzed predominately qualitative data. The items in Fig. 5 are synthesized from existing methodological guidance for GT and predominately qualitative studies (cf. [16, 24, 28, 78]), as well as our own experience in conducting qualitative studies. No single article can or should include all of these items. Instead, we offer them as a reminder of "questions to ask oneself" before and

General Grounded Theory Issues

- What variant of grounded theory have you adopted? What published guidance did you follow?
- How and why have you adapted, or deviated from, this variant and guidance?
- State the research area or research question—either your initial question, the question that emerged during your study, or preferably both.
- State your epistemological and ontological positions (e.g. interpretivism, critical realism).
- State the duration of the study.

Site Selection and Description

- What organization, team, dataset, etc. did you study?
- Why did you study this data?
- Describe the context of the study (e.g. the kind of organization, who is involved, what kind of software is being developed).

Role of the Literature in the Grounded Theory Study

- Did you begin data collection with a clean theoretical slate?
- What topic areas did you review before and during the study?
- How does the literature inform, support or refute your analysis and results?

Presenting and Evaluating Grounded Theory

- Is the theoretical contribution clearly stated?
- Is the generated theory integrated back into the literature?
- Is the theory evaluated? If so, using which criteria?
- How might your own biases, preconceptions, background and beliefs affect your analysis?

Grounded Theory Data Collection and Analysis

- What data was collected (e.g. field notes, documents, emails, video of meetings), how and when?
- Who collected and analyzed the data? Was it an individual researcher or research team? If a team, who did what? How was this coordinated?
- Describe the pacing of analyzing data, and how it continued throughout the project.
- Describe your coding, memoing and sorting with examples.
- Describe the emergence of your core category, and how this affected your analysis.
- If using classic GT, did you use any of Glaser's coding families? If so, which, and did the theoretical codes earn their way into the theory?
- If using Straussian GT: state how you used the conditional matrix.
- How and where was your data stored? How did you manage the volume and heterogeneity of data?
- Describe your theoretical sampling with examples.
- Confirm that you employed constant comparison.
- When did you stop collecting data? Describe how theoretical saturation became apparent.
- Describe how the selected GT variant affected data collection and analysis.
- Did you conduct a reliability check; i.e., have your analysis reviewed by someone else. If so, who, how, what did they find and what changes resulted? Describe their expertise.

Figure 5. Specific considerations for conducting and reporting grounded theory

during a study and write-up. Simply confirming that a study follows the various core GT guidelines (e.g. simultaneous data collection and analysis, constant comparison, theoretical saturation) should be unnecessary. However, because GT is still relatively new to software engineering, and our study demonstrates some confusion about how GT works, clearly describing what was done and enumerating adherence to core guidelines will benefit readers, reviewers and editors.

5.3 Challenges in Doing Grounded Theory Research in Software Engineering

Software development contexts present several unusual challenges for grounded theory research. Most of the GT research we have read relies primarily on interviews and documents. However, software contexts provide diverse data sources including: source code, test suites, code commit logs, task and effort data from project management software, design diagrams (e.g. wireframes, class diagrams), design documents, domain models (e.g. scenarios, personas, user stories, use cases), project management documents (e.g. backlogs, burn-down charts), performance data, issue tracker data, photos of temporary diagrams (e.g. on white boards), online discussions (e.g. on IRC or Slack), contracts and financial statements. Combining these with the usual data (i.e. audio/video recordings of interviews and meetings, documents, email, field notes) exacerbates at least three challenges:

1. **Managing large amounts of heterogeneous data**. Version control, project management, team communication systems and other technical affordances make it easy to get access to an enormous, unreadable dataset. Capturing, storing, indexing and managing all this data is practically challenging. Systems appropriate for storing some data types (e.g. NVivo for audio, video, transcripts and documents) may be unsuitable for storing other data types (e.g. code). Determining what to read when you have more text than you can read in a lifetime is even more challenging. The implications of data magnitude for theoretical sampling remain unclear. However, one strategy is to choose an explicit primary data source (e.g. interviews) and theoretically sample from the remaining data based on leads arising from the primary data source.
2. **Coding unconventional texts**. While they may apply more broadly, the coding approaches associated with GT (e.g. open and theoretical coding) were developed primarily for analyzing unstructured text. It is not clear how to apply open coding to design diagrams, structured text (e.g. use cases) or source code. One approach is to open-code unstructured text and move directly to memoing for more structured data. Another is to adopt completely different analytical techniques; for instance, static code analysis.
3. **Cross-referencing participant statements with records**. Participants' post-hoc reconstructions of how and why they performed certain actions are less reliable than, for example, their accounts of their current frustrations or enduring values. Source code, commit logs, project management data and direct observation allow the researcher to triangulate many interviewee claims. This presents myriad challenges regarding not only how to triangulate but also how to resolve conflicting evidence.

6. CONCLUSION

Grounded theory is an increasingly popular research method in software engineering (see Fig. 2). However, grounded theory is complex and demanding, with several variants and conflicting guidance, and software engineering researchers may not be cognizant of its historical development or appreciate the differences across its three main variants. This paper aims to draw attention to this issue and to report on the use of Grounded Theory in SE. The contributions of this paper are fourfold:

1. We provide a detailed comparison of the three main variants of grounded theory, which may help aspiring grounded theory researchers in software engineering to select the variant that suits them best (Sec. 2);
2. Based on an analysis of almost 100 articles in nine prominent SE journals, we found that many SE articles do not generate a theory, do not clearly indicate which variant of grounded theory is used and do not provide sufficient methodological detail for rigorous evaluation (Sec. 4);
3. We offer integrated guidance for conducting and reporting grounded theory research in software engineering, including a set of suggestions for explaining the study's data collection and analysis procedures (Fig. 5);
4. We enumerate substantial challenges peculiar to conducting GT research in software engineering, including the proliferation of heterogeneous unstructured, semi-structured and structured data (Sec. 5).

These contributions should be interpreted in light of several limitations. We limited our study to those articles published in nine well-known software engineering journals. While we believe these journals are a reasonable surrogate for the broader SE literature, the field has many more, including journals which are focused on specific research areas (e.g. *Requirements Engineering*). We also excluded conference papers, reasoning that page limits would force authors to include less methodological detail. Further sampling bias could come from articles missed due to our specific search string and search strategy, or due to publication bias. Furthermore, we can only analyze the way each study is reported rather than how it was done. A few missing methodological details clearly does not mean that the research is poor or that the authors are unskilled. Our review simply reveals that more methodological detail is needed and suggests potential details to include in future articles.

We believe that grounded theory offers a highly suitable methodology to address social, cultural and human aspects in software engineering—several GT studies in SE have contributed novel and rich insights. As described above, software engineering presents non-trivial challenges for grounded theory research. However, grounded theory remains one of the most rigorous methods to generate new theories. This is a significant issue as the establishment of a strong theory base has been identified as an important challenge for the software engineering discipline [50, 65, 66]. We believe well conducted GT studies can make significant contributions to our field and help to develop rich theories to inform future empirical studies in SE.

7. ACKNOWLEDGMENTS

We thank Lutz Prechelt and the anonymous reviewers for constructive feedback. This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094 to Lero—the Irish Software Research Centre (www.lero.ie); the Irish Research Council New Foundations Scheme 2014; Enterprise Ireland grant IR/2013/0021 to ITEA2-SCALARE (www.scalare.org); and the Royal Irish Academy under the Charlemont Award Programme.

8. REFERENCES

[1] Adolph, S., Hall, W. and Kruchten, P. 2011. Using grounded theory to study the experience of software development. *Empirical Software Engineering*, 16, 4, 487-513.

[2] Adolph, S., Kruchten, P. and Hall., W. 2012. Reconciling perspectives: A grounded theory of how people manage the process of software development. *J Sys Softw*, 85, 1269-1286.

[3] Baker, C., Wuest, J. and Stern, P.N. 1992. Method slurring: the grounded theory/phenomenology example. *Journal of Advanced Nursing*, 17, 1355-1360.

[4] Benoliel, J.Q. 1996. Grounded theory and nursing knowledge. *Qualitative Health Research*, 6, 3, 406-428.

[5] Berger, P. and Kellner, H. 1981. *Sociology Reinterpreted: An Essay on Method and Vocation*. Penguin, Harmondsworth.

[6] Bertelsen, O. 1997. Toward a unified field of se research and practice. *IEEE Softw.*, 14, 6, 87-88.

[7] Bettenburg, N., Hassan, A.E., Adams, B. and German, D.M. 2013. Management of community contributions: A case study on the Android and Linux software ecosystems. *Empirical Software Engineering*, 20, 1, 252-289.

[8] Birks, M. and Mills, J. 2011. *Grounded Theory: A Practical Guide*. Sage.

[9] Bogdan, R. and Taylor, S. 1975. *Introduction to Qualitative Reseach Methods*. Wiley & Sons, New York.

[10] Boychuk Duchscher, J.E. 2004. Grounded Theory: Reflections on the emergence vs. forcing debate. *Journal of Advanced Nursing*, 48, 6, 605-612.

[11] Bryant, A. and Charmaz, K. 2007. *The SAGE Handbook of Grounded Theory*. Sage.

[12] Carvalho, L., Scott, L. and Jeffery, R. 2005. An exploratory study into the use of qualitative research methods in descriptive process modelling. *Information and Software Technology*, 47, 2, 113-127.

[13] Carver, J. 2004. The Impact of Background and Experience on Software Inspections. *Empir Software Eng*, 9, 259-262.

[14] Chamberlain-Salaun, J., Mills, J. and Usher, K. 2013. Linking symbolic interactionism and grounded theory methods in a research design: From Corbin and Strauss' Assumptions to Action. *SAGE Open*, 3, 3, 1-10.

[15] Charmaz, K. 1990. "Discovering" illness: using grounded theory. *Social Science and Medicine*, 30, 11, 1161-1172.

[16] Charmaz, K. 2014. *Constructing Grounded Theory*. Sage, 2nd Ed.

[17] Coleman, G. and O'Connor, R. 2008. Investigating software process in practice: A grounded theory perspective. *Journal of Systems and Software*, 81, 772-784.

[18] Coleman, G. 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.

[19] Corbin, J. and Strauss, A. 2015. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage, 4th Ed.

[20] Creswell, J.W. 2013. *Qualitative Inquiry & Research Design: Choosing Among Five Approaches*. Sage, 3rd Ed.

[21] Denzin, N. 1983. Interpretive interactionism. In: G. Morgan (Ed.) Beyond Method. Sage, California.

[22] Denzin, N. 2007. *Grounded Theory and the Politics of Interpretation*. Sage.

[23] Dittrich, Y., John, M., Singer, J. and Tessem, B. 2007. Editorial: For the Special issue on Qualitative Software Engineering Research. *Inform Soft Technol*, 49, 6, 531-539.

[24] Dube, L. and Pare, G. 2003. Rigor in information systems positivist case research: Current practices, trends and recommendations. *MIS Quart.*, 27, 4, 597-635.

[25] Duchscher, J.E.B. and Morgan, D. 2004. Grounded theory: reflections on the emergence vs. forcing debate. *Journal of Advanced Nursing*, 48, 6.

[26] Dybå, T., Prikladnicki, R., Rönkkö, K., Seaman, C. and Sillito, J. 2011. Special issue on qualitative research methods in software engineering. *Empir Software Eng*, 16, 2.

[27] Easterbrook, S., Singer, J., Storey, M.-A. and Damian, D. 2008. Selecting empirical methods for software engineering research. In: F. Shull, J. Singer and D. I. K. Sjøberg (Eds.) Guide to Advanced Software Engineering. Springer.

[28] Eisenhardt, K.M. 1989. Building theories from case study research. *Academy of Management Review*, 14, 4, 532-550.

[29] Espinosa-Curiel, I.E., Rodríguez-Jacobo, J. and Fernández-Zepeda, J.A. 2013. A framework for evaluation and control of the factors that influence the software process improvement in small organizations. *Journal of software: Evolution and Process*, 25, 4, 393-406.

[30] Fagerholm, F., Ikonen, M., Kettunen, P., Münch, J., Roto, V. and Abrahamsson, P. 2015. Performance Alignment Work: How software developers experience the continuous adaptation of team performance in Lean and Agile environments. *Inform Soft Technol*.

[31] Fitzgerald, B. and Howcroft, D. 1998. Towards dissolution of the IS research debate: from polarization to polarity. *Journal of Information Technology*, 13, 4, 313-326.

[32] Garvey, W.D. and Griffith, B.C. 1971. Scientific communication: Its role in the conduct of research and creation of knowledge. *American Psychologist*, 26, 4.

[33] Gell-Mann, M. 1999. Complex adaptive systems. In: Complexity: Metaphors, models and reality Westview Press.

[34] Glaser, B.G. 1978. *Theoretical Sensitivity*. Sociology Press.

[35] Glaser, B.G. 1992. *Basics of Grounded Theory Analysis: Emergence vs Forcing*. Sociology Press.

[36] Glaser, B.G. 1998. *Doing Grounded Theory: Issues and Discussions*. Sociology Press.

[37] Glaser, B.G. 2002. Constructivist Grounded Theory? *Forum: Qualitative Social Research*, 3, 3, Art. 12.

[38] Glaser, B.G. and Strauss, A.L. 1967. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine de Gruyter, New York.

[39] Glass, R.L., Vessey, I. and Ramesh, V. 2002. Research in software engineering: an analysis of the literature. *Inf Softw Technol*, 44, 8, 491-506.

[40] Goulding, C. 2002. *Grounded Theory: A Practical Guide for Management, Business and Market Researchers*. Sage.

[41] Guba, E. and Lincoln, Y. 1994. Competing paradigms in qualitative research. In: N. Denzin and Y. Lincoln (Eds.) The Handbook of Qualitative Research. Sage.

[42] Heath, H. and Cowley, S. 2004. Developing a grounded theory approach: a comparison of Glaser and Strauss. *International Journal of Nursing Studies*, 41, 141-150.

[43] Hirschheim, R. 1985. Information systems epistemology: an historical perspective. In: E. Mumford, R. Hirschheim, G. Fitzgerald and A. Wood-Harper (Eds.) Research Methods in Information Systems. Elsevier.

[44] Hoda, R., Noble, J. and Marshall, S. 2011. Grounded theory for geeks. In *Proc. 18th Conference on Pattern Languages of Programs*.

[45] Hoda, R., Noble, J. and Marshall, S. 2011. The impact of inadequate customer collaboration on self-organizing Agile teams. *Information and Software Technology*, 53, 5, 521-534.

[46] Hoda, R., Noble, J. and Marshall, S. 2012. Developing a grounded theory to explain the practices of self-organizing Agile teams. *Empir Software Eng*, 17, 6, 609-639.

[47] Hoda, R., Noble, J. and Marshall, S. 2013. Self-organizing roles on agile software development teams. *IEEE Trans Softw Eng*, 39, 3, 422-444.

[48] Holland, J.H. 1992. Complex Adaptive Systems. *Daedalus*, 121, 1, 17-30.

[49] Jantunen, S. and Gause, D.C. 2014. Using a grounded theory approach for exploring software product management challenges. *Journal of Systems and Software*, 95, 32-51.

[50] Johnson, P., Ekstedt, M. and Jacobson, I. 2012. Where's the Theory for Software Engineering? *IEEE Softw.*, 29, 5, 94-96.

[51] Kelle, U. 2005. "Emergence" vs. "Forcing" of Empirical Data? A crucial problem of "Grounded Theory" Reconsidered. *Forum: Qualitative Social Research*, 6, 2.

[52] Kenealy, G. 2008. Management Research and Grounded Theory: A review of grounded theory building approach in organisational and management research. *The Grounded Theory Review*, 7, 2, 95-117.

[53] Layman, L., Williams, L., Damian, D. and Bures, H. 2006. Essential communication practices for Extreme Programming in a global software development team. *Information and Software Technology*, 48, 9, 781-794.

[54] Maglyas, A., Nikula, U. and Smolander, K. 2013. What are the roles of software product managers? An empirical investigation. *Journal of Systems and Software*, 86, 12, 3071-3090.

[55] Mohanani, R., Ralph, P. and Shreeve, B. 2014. Requirements Fixation. In *Proc. International Conference on Software Engineering*. ACM.

[56] Morgan, G. (Ed.). 1983. Beyond Method. Sage, CA, USA.

[57] Morse, J.M. (Ed.). 1994. Critical Issues in Qualitative Research Methods. Sage.

[58] Opie, C. 2004. Research Approaches. In: C. Opie (Ed.) Doing educational research. Sage, London.

[59] Prechelt, L. and Oezbek, C. 2011. The search for a research method for studying OSS process innovation. *Empirical Software Engineering*, 16, 4, 514-537.

[60] Ralph, P. 2015. Software engineering process theory: A multi-method comparison of Sensemaking-Coevolution-Implementation Theory and Function-Behavior-Structure Theory. *Inform Soft Technol*, 70, 232-250.

[61] Ralph, P. and Kelly, P. 2014. The Dimensions of Software Engineering Success. In: International Conference on Software Engineering. ACM, Hyderabad, India.

[62] Seaman, C.B. 1999. Qualitative methods in empirical studies of software engineering. *IEEE Trans Softw Eng*, 25, 4.

[63] Seo, H., Sadowski, C., Elbaum, S., Aftandilian, E. and Bowdidge, R. 2014. Programmers' build errors: a case study (at google). In *Proc. International Conference on Software Engineering*. ACM.

[64] Stern, P.N. 1994. Eroding grounded theory. In: J. M. Morse (Ed.) Critical Issues in Qualitative Research Methods. Sage.

[65] Stol, K.J. and Fitzgerald, B. 2015. Theory-Oriented Software Engineering. *Science of Computer Programming*, 101, 79-98.

[66] Stol, K.J., Goedicke, M. and Jacobson, I. 2016. Introduction to the special section—General Theories of Software Engineering: New advances and implications for research. *Inf Softw Technol*, 70, 176-180.

[67] Stol, K.J., Ralph, P. and Fitzgerald, B. 2016. Appendix to "Grounded Theory Research in Software Engineering". University of Limerick.

[68] Strauss, A. and Corbin, J. 1991. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage.

[69] Strauss, A. and Corbin, J. 1994. Grounded Theory Methodology: An Overview. In: N. Denzin and Y. Lincoln (Eds.) Handbook of Qualitative Research. Sage.

[70] Strauss, A. and Corbin, J. 1998. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage, 2nd Ed.

[71] Strauss, A.L. 1987. *Qualitative analysis for social scientists*. Cambridge University Press.

[72] Suddaby, R. 2006. From the editors: What grounded theory is not. *Academy of Management Journal*, 49, 4, 633-642.

[73] Trochim, W. 2001. *Research Methods Knowledge Base*. Atomic Dog Publishing, Cincinnati, OH, USA.

[74] Urquhart, C. 2013. *Grounded Theory for Qualitative Research: A Practical Guide*. Sage.

[75] Urquhart, C., Lehmann, H. and Myers, M.D. 2010. Putting the 'theory' back into grounded theory: guidelines for grounded theory studies in information systems. *Information Systems Journal*, 20, 357-381.

[76] Walker, R. 1988. *Applied Qualitative Research*. Gower, Hampshire.

[77] Wohlin, C., Höst, M. and Henningsson, K. 2003. Empirical research methods in software engineering. In: ESERNET, volume LNCS 2765. Springer.

[78] Yin, R.K. 2008. *Case study research: Design and methods*. Sage, CA, USA, 4th Ed.