Grounded Theory for Geeks

Rashina Hoda, James Noble, Stuart Marshall

School of Engineering and Computer Science
Victoria University of Wellington,   Wellington, New Zealand
{rashina,kjx,stuart}@ecs.vuw.ac.nz

ABSTRACT

Grounded Theory (GT) is gaining popularity as research method in Software Engineering; however, it is still not widely understood. We present some patterns based on the use of GT to study Software Engineering contexts including our own experiences of applying classic GT in an attempt to make it more accessible for Software Engineering researchers (aka 'geeks') .

INTRODUCTION

Glaser and Strauss, the developers of Grounded Theory, famously said: *"It doesn't take a genius to develop a useful grounded theory"* [9] and we believe *even a geek can do it!*

Grounded Theory (GT) is the systematic generation of theory from data analyzed by a rigorous research method [8-11]. GT is a complete research method in that it provides concrete procedures that cover all stages of research including sampling participants, data collection, data analysis, use of literature, and write-up. The emphasis is on uncovering the main concerns of a majority of participants and generation of a theory to explain how they go about resolving this main concern. The main concern could be any aspect of the field that the researcher is interested in exploring that is particularly important (and even problematic) for those involved. For example, securing sufficient customer involvement in software development projects could be a recurring concern for a number of practitioners. The role of the researcher is to uncover this main concern via data collection and analysis and present the theory that explains how people go about resolving it.

GT is appropriate as a research method to explore the human and social aspects of Software Engineering due to several reasons. Firstly, GT, used as a qualitative method, allows the study of social interactions and behaviour. Secondly, GT is useful when studying a relatively little explored areas and there has been limited research on the human and social aspects of Software Engineering. Thirdly, GT allows generation of concepts and categories which have immense "grab" for practitioners.

Recently, more researchers have started to use GT as a research method to study the social side of Software Engineering [1,5,6,7,13,14,19]. However, the method is still not widely understood and accepted in the field of Software Engineering. In this paper, we present some GT patterns based on the use of GT in Software Engineering studies including our own application of the method to study human and social aspects of Software Engineering. The aim of the paper is to describe the GT procedures in a patterns format in order to make it more accessible to fellow geeks.

THE CONTEXT

The GT patterns described below can help describe GT for geeks in a way that makes it more accessible to them. These GT patterns will be of particular interest to researchers exploring the social side of Software Engineering and their supervisors.
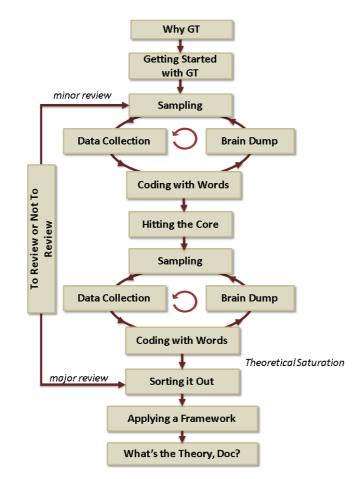
QUICK OVERVIEW

- **Choosing GT** explains how to go about exploring research methods and being sure if GT is the one to use.
- **Getting Started with GT** explains how to go about starting a GT research project.
- **To Review or Not to Review** explains the solution to the dilemma faced by the researchers on when, how, and how much 'Literature Review' to conduct.
- **Sampling** explains how to go about finding and selecting participants (via 'Theoretical Sampling' in GT)
- **Data Collection** explains how to collect data from participants.
- **Coding with Words** explains how to perform data analysis (accomplished by the 'Constant Comparison Method' in GT and divided into 'Open Coding' and 'Selective Coding' procedures in GT.)
- **Brain Dump** explains how to capture tacit information and explore relationship between categories (called 'Memoing' in GT)
- **Hitting the Core** explains how to go about discerning the main concern of the participants or central theme the analysis (called the 'Core category' in GT)
- **Sorting it Out** explains the process of forming a theoretical outline (called 'Sorting' in GT)
- **What's the Theory, Doc?** explains how to go about presenting a grounded theory in a thesis.

Table 1. GT Patterns: Problem-Solution Summaries

Pattern	Problem	Solution
Why GT	How do you know GT is the right research method for your research project?	Read about different research methods. Eliminate some methods on grounds of whether your planned research is qualitative or quantitative, whether you are excited at the prospect of generating theory or would rather test an existing one, etc. Evaluate how well your research falls into the GT sweet-spot. Finally, an important criteria for choosing between a few equally well-applicable methods (such as Ethnography) is the availability of a supervisor able to 'mentor' you in the use of that method.
Getting Started with GT	How do you get started on a GT research project?	Read a few GT books by Glaser and Strauss and examples of GT studies (preferably in the Software Engineering context) to get a sense of what it's all about. Read the rest of the patterns. Apply for Human Ethics approval for collecting data from human participants.
To Review or Not to Review	When, how, and how much literature review to conduct in GT?	Conduct a minor literature review upfront to learn some basic terminology in the research area. Conduct a major literature review towards the end in order to tie your findings with previous literature.
Sampling	How do you go about finding and selecting participants?	Contact practitioner communities or online user groups/mailing lists in your research area with a request for participation. Attend events and conferences in your research area to find practitioners.
Data Collection	How do you collect data from participants and when do you stop?	Conduct semi-structured interviews with open-ended questions. Use an audio recorder if possible. Evolve questions iteratively. Supplement interviews with observations of workplace, practices, and artefacts. Interview people in a range of roles to get multiple perspectives. Apply for travel grants. Combine data collection with conference travel to make your research funds go further.
Coding with Words	How do you perform data analysis?	Apply Open Coding procedure to analyze your data. The Constant Comparison Method allows patterns across different data sets to be captured. Once the main theme of the research is clear, move to Selective Coding.

Brain Dump	How do you capture tacit information and explore relationships between categories?	Write theoretical notes (or memos) throughout the GT process. It allows tacit knowledge such as the relationships between different patterns in the data and any further questions to be explored and recorded explicitly.
Hitting the Core	How to go about discerning the main concern of the participants or the 'Core category' from the analysis?	The category that is central, reoccurs frequently, is related to the other main categories, and accounts for most variations in data is the Core.
Sorting it Out	How do you form a theoretical outline?	Sorting involves arranging the memos on a conceptual level once the data collection is saturated. Sorting the memos results in an outline of the theory describing how the different categories relate to the Core category.
What's the Theory, Doc?	How do you go about presenting a grounded theory?	Familiarize yourself with the required thesis format at your department. The results section of the thesis will present the grounded theory following the theoretical outline generated as a result of Sorting. Expand each of the categories of the theory into chapters or sections. Use participant quotes and textual field notes to highlight emerging points of interest. Explain your research method well – remember others will look up to this as an example of a GT study.

RELATED PATTERNS

Figure 1. The Grounded Theory Patterns

THE PATTERNS FORMAT

All patterns presented in this paper follow a consistent format   which starts with a   **Pattern Name**, the **Context** of use, an **Example** of a geek researcher as introduced in the following section,   the   **Problem** that the pattern addresses, the   **Forces**   at play, the   **Solution**, the   **Consequences**   of the solution that resolve the   forces (shown with a positive symbol) and those that fail to   resolve them (shown with a negative symbol), the **Example   Resolved**, the **Known Uses**   from our own application of GT [14-18] and other examples of using GT in SE contexts [1, 12,19], and the **Related Patterns**.

SCENARIO

Sabrina has recently enrolled into a PhD degree at the *Very-Technical University of Technology.* The supervisors and other PhD candidates in her department have research interests varying from Subject-Oriented Paradigms, Informal Methods to Robot-Computer Interaction, Fandroid Applications, and Multi-Touch Chairs. Sabrina's is interested in studying software teams aka People!

Dr. Who: "*So what's up*?"
Sabrina: "*I think I want to explore Agile software development teams.*"
Dr. Who: "Great! That's an emerging area"
Sabrina: "Yeah…I think I could, you know, talk to some practitioners down-town…"
Dr. Who: "Sounds like a plan."
  [few cokes later]
"Do you have a research method in mind?"
Sabrina [puzzled look]: *"Huhh?"*

PATTERN 1: Why GT

Context: A researcher starting off on their research

Example: Having chosen a research area, Sabrina now needs to select a research *method*.

Sarbina: [It took me forever to come up with a research are already!!]
" what's the deal with research methods?"
Dr. Who: "*The answer is: there are many research methods to choose from…."*

And so Sabrina goes off to read more about research methods. She doesn't know very many details of her project yet but she knows that Agile software development is a relatively new area and has not been researched extensively yet. She's also pretty keen on conducting her research in the industry – collecting data from 'the field' – and will need to talk to people.

Problem: How do you know if GT is the right research method for your research project?

Forces:
- Researcher may not be aware of the different kinds of research methods out there
- Researcher may lack an understanding of whether a particular method (e.g. GT) applies
- Researcher may zero in onto a particular approach (e.g. GT) too early without having sufficient understanding
- Researcher needs to support why they chose a particular method in their research proposal
- Insufficient understanding may lead to researcher being unable to explain their choice or worse, fail their proposal.

Solution: Read about different research methods. Eliminate some methods on grounds of whether your planned research is qualitative or quantitative, whether you are excited at the prospect of generating theory or would rather test an existing one. Evaluate how well your research falls into the GT sweet-spot (explained below). Finally, an important criteria for choosing between a few equally well-applicable methods (e.g. Ethnography) is the availability of a supervisor to 'mentor' you in the use of the method.

The researcher should, either through reading up on their own or with advice from their supervisors, attempt to understand the different research methods available to them. Most universities have a standard course that goes over various research methods and their applicability to different research projects (Note: such a course may well be offered outside the Computer Science/Engineering faculty). Taking such a course will prove useful. At the very least, the researcher should read up on different research methods and given the limited details of their research, attempt to decide what applies best [20, 26, 27]. It may be easier to start by eliminating some research methods based on whether your planned research is qualitative or quantitative, whether you are excited at the prospect of generating theory or would rather test a self-developed hypothesis or an existing theory.

Grounded Theory is highly applicable when wanting to conduct a qualitative research to generate a theory in a relatively new research area (without extensive existing literature) via collecting data from interviews with people and observations 'on the field'. This is the GT sweet-spot.

Finally, an important criterion for choosing between a few equally well-applicable methods (such as Ethnography) is the availability of a supervisor able to 'mentor' you in the use of that method.

Consequences:

+   Researcher gets a better sense of what research methods are available
+   Based on the details of their planned research (qualitative/quantitative/generating theory/testing existing or hypothesized theory), they may be able to eliminate some choices.
+   If your research falls in the GT sweet spot you'll be – as we say in New Zealand – "sweet as!"
+   Being able to defend your choice of research method will make you more confident in answering questions related to methodology, increasing your likelihood of surviving the proposal phase.
[+ or -] If your research is an antithesis of the GT sweet spot (claims of quantitative GT studies notwithstanding) – look for a different research method.
[+ or -] Your choice may be helped by the availability of an expert mentor in a particular methodology you were considering – this may or may not be GT.

Example Resolved: Sabrina's narrowed her choices down to Ethnography and Grounded Theory. She likes the idea of generating a new theory and is not so excited about extensive periods of bird-watching (oops, I mean, observing - at length - participants go about their normal chores). Talking to Dr. Who makes the choice easier – he's an expert in GT.

Known Uses: We chose Grounded Theory as a research method for several reasons: strong institutional support and a successful history of using Grounded Theory for exploring human and social aspects of Agile teams [19] within the department were the primary reasons for selecting Grounded Theory over other applicable methods, such as Ethnography. Other reasons include the following: firstly, Agile methods focus on people and interactions and Grounded Theory, used as a qualitative research method, allows the study of social interactions and behaviour. Secondly, Grounded Theory focuses on theory generation, rather than extending or verifying existing theories---an interesting and exciting prospect. Thirdly, Grounded Theory is useful when studying relatively new areas or when trying to gain a fresh perspective on a well-known area

and there has been limited research on the human and social aspects of Agile software development. Finally, Grounded Theory has been used successfully, and continues to gain popularity, as a research method to study Agile software development teams around the world [1,5-7,13,14,19].

Related Pattern: Getting started with GT (next pattern)

PATTERN 2: GETTING STARTING WITH GT

Context: A geek starting off on their GT study

Example: Dr. Who recommends the books by Glaser and Strauss and one by Kathy Charmaz.
[Three weeks and seven attempts at reading these books later...]
Dr. Who: "*How's it going?*"
Sabrina [with a thumping migraine]: "*It's complicated. I can see some interesting bits but it's not easy to decode these books (for us geeks anyway)…I can only find examples of use in Nursing on the web but no examples of use in Software Engineering you know…like, where do I even begin?*"

Problem: How do you get started on a GT research project?

Forces:
- Reading ALL the GT books in chronological order may leave the researcher confused.
- The classic GT books by themselves may not be enough
- Researcher may have a research question/hypothesis in mind
- It may not be obvious that talking to human beings needs to be pre-approved

Solution: Read a few GT books by Glaser and Strauss and examples of GT studies (preferably in the Software Engineering context) to get a sense of what it's all about. Read the rest of the patterns in this paper.
Apply for Human Ethics approval for collecting data from human participants.

Researcher should read these GT books to begin with: *Discovery of Grounded Theory* (1967) and *Theoretical Sensitivity* (1978). Reading studies that used GT in a Software Engineering context is extremely useful in getting a sense of the process and the final product. Some examples include [1,5-7,13,14,19]. The rest of this paper aims to explain the basic GT procedures and can serve as a good starting point.

Some other classic GT books may be read throughout the research for better understanding in the following order: *Doing Grounded Theory: Issues and Discussions* (1998), *Basics of Grounded Theory Analysis* (1992), *The Grounded Theory Perspective III: Theoretical Coding* (2005).

The researcher should have a research area in mind but should NOT have a defined research question or hypothesis because the idea is to *generate new theory* not test existing theories. It may come as a surprise to some researchers that data collection from human participants requires Human Ethics approval. Applying and receiving this approval well in advance saves time. The application should clarify the use of semi-structured, open-ended questions so that the research can evolve the questions over the period of the research.

Consequences:

+ The researcher reads a few GT books by Glaser and Strauss to get an understanding of the GT method
+ Reading these books in the recommended order will foster easier understanding on the GT procedures
+ Reading the books alone is not enough and supplementing them with real examples of GT

studies (preferably in the SE context) will help.

+ Researcher gains proper Human Ethics approval

− Researcher may NOT find enough examples of use of GT in a Software Engineering context

− The researcher may need to abandon their hypothesis if they had one. In other words, if the researcher had a predefined hypothesis – a hunch about the main concern of the participants – they'll need to keep that aside and let the true concern of the participants emerge through the research.

− Alternatively, researcher may decide to abandon GT instead!

− Receiving Human Ethics approval requires time (depending on individual committees)

Example Resolved: Sabrina gives a second shot at reading the GT books – this time in the recommended order. She finds *Theoretical Sensitivity* is the single most valuable of all the GT books so far. She applies for and receives Human Ethics approval for data collection while she has been wondering whether she is should be conducting a proper literature review to know more about the area: Agile software development.

Known Uses: We read some of the GT books by Glaser and Strauss and soon figured out the most useful order in which to read them (as described In Solution above). When we started our research in 2006 there weren't very many examples of GT studies in Software Engineering that explained how they went about conducting GT. This was difficult. Over the years, more examples of use of GT in Software Engineering are available [1,5-7,13,14,19].

Reading examples of GT studies from Nursing helped get a feel for what the method and its final product – a grounded theory – is meant to look like. We had chosen Agile software development as our area of research.

Related Patterns: To Review or Not to Review (next pattern)

PATTERN 3: TO REVIEW OR NOT TO REVIEW

Context: A researcher starting off on their research using GT

Example: *Sabrina* knows some basic terminology and concepts of Agile software development courtesy some courses she took at university. She's now wondering whether she should conduct a proper literature review on Agile software development. She is also aware that reading too much before-hand can bias her study by introducing pre-conceived ideas in her head.

Problem: When, how, and how much literature review to conduct in GT?

Forces:
• Most research methods require the candidate to undertaken extensive literature review within their area of research
• University and department procedures may require candidate to produce an extensive literature review upfront
• Conference/Journal reviewers may expect to find a clear literature review section in submissions.
• Extensive literature review in the same area of research may influence researcher with preconceived ideas hindering their ability to generate new theory.
• No prior knowledge of the research area whatsoever may prove to be a problem when collecting data from participants.
• SE subject areas are full of jargon, which may take forever to learn.

Solution: Conduct a minor literature review upfront to learn some basic terminology in the research area and discern if any well-established theories pre-exist in the area. Conduct a major literature review towards the end in order to tie your findings with previous literature.

Literature review is not forbidden in GT, however the trick is to know when and how much of it to use. Conducting a minor review upfront will help discover any pre-existing, well-known theories in the research area so the researcher can avoid 're-inventing the wheel'. Conducting a full-blown review upfront, on the other hand, can fill the researcher's mind with too many pre-conceived ideas and inhibit theory generation.

Knowing enough terminology initially to carry out a conversation with the participants is essential so that you don't waste the participants' time in clarifying basic terminology in the field. Reading literature in other areas of research will help you familiar with GT studies in general. Conducting the major review in your area of research towards the end helps consolidate your own findings with previous literature. It also works faster because your review is limited to the main themes in your findings.

Consequences:

+ Researcher knows enough to carry out a conversation with their participants.
+ Researcher doesn't come across as an idiot.
+ Researcher doesn't waste precious interview/observation time capturing/clarifying basic terminology.
− Researcher may fail to fulfil any particular requirements of their university/department around extensive literature reviews.
− Researcher needs to resist the temptation to conduct extensive literature review upfront.

Example Resolved: Sabrina realizes that literature review in GT can be conducted in two phases. She starts off by reading basic Agile values and principles and the basic structure and terminology of some popular Agile methods such as Scrum and XP. Sabrina knows what planning poker, story wall, and demos mean. She is feels confident enough to talk to Agile practitioners.

Known Uses: The issue of literature review in GT has been a topic of debate [23,24]. As geeks exploring topics within SE, we soon realized that Agile, like other SE areas, is full of jargon. We read enough to know what most of the basic terminology refers to, such as names of different Agile methods (Scrum, XP, etc), values and principles of the Agile Manifesto, concept of iterations, story walls, user stories, planning poker, team velocity etc. This allowed us to conduct interviews and observations with greater ease and without wasting too much time learning these terms and their meanings from participants. Towards the later stages when the findings from the research have clearly emerged, we conducted extensive literature reviews for each of the major categories to see how well they fit with (or differed from) prior literature. Examples of other SE studies and their use of literature review can be found in [1,19].

Basic information about the research area – Agile software development – was gained through reading the Manifesto values and principles and introductory Scrum and XP books. We avoided reading research literature on Agile to begin with. Towards the end of the research, once our own findings were well grounded, we conducted a major review and incorporated related works with our results.

Related Patterns: Sampling (next pattern), **Sorting it Out**

PATTERN 4: SAMPLING

Context: A researcher having done a minor literature review

Example: Sabrina is now familiar with basic Agile terminology and is looking to find some participants.

Problem: How do you go about finding and selecting participants?

Forces:

- Researcher needs to find participants: practitioners in their researcher area
- Practitioners are busy people
- Practitioners may not be willing to participate

Solution: Contact practitioner communities or online user groups/mailing lists in your research area with a request for participation. Attend events and conferences in your research area to find practitioners.

Contacting practitioner communities and online user groups/mailing lists in your research area with a request for participation can get you some initial participants. These participants can provide access to other practitioners in their companies. Attending events and conferences in your area of research provides a platform to connect with many practitioners and opportunities to request them to participate in your research. Clearly stating the objectives of your research and a basic idea of your research method will help participants feel comfortable about what they are getting into. Confirming Human Ethics committee approval and complete confidentiality (if that's how you do it) can provide added comfort to prospective participants. As you progress in your research, you may want to sample participants that help expand your participant base or gain information on particular themes emerging from your analysis. This is called "Theoretical Sampling" in GT. Once you've got some publications from your research, you can offer these publications or presentations to your participants in return for their participation. Practitioners appreciate access to research that addresses their concerns.

Consequences:

- + Participants can be found through practitioner communities/user groups/mailing lists
- + Participants can be found at events/conferences in the research area
- + Having Human Ethics approval increases your chances of finding participants
- + Over time researcher can choose consciously concentrate or diversify their participant base.
- − Not all practitioners you want to include may be able or ready to participate despite all your effort. This is a limitation of your research.

Example Resolved: Sabrina has found several "real-life" Agile practitioners at the local Agile Club. One contact seems to lead to another and often she gets to talk to multiple people on a single Agile project.

Known Uses: We searched for and found the Agile Software Community of India (ASCI) user group which had a number of Agile practitioners as its members. Emailing the group found us some participants. Meeting them in person led to more contacts. Attending the Agile BarCamp in Wellington allowed us to connect with local Agile practitioners. Some of these offered to participate. Later with the establishment of the Agile Professional Network in Wellington it was easier to find participants. A few practitioners we approached did not oblige due to various reasons. Our earlier participants were from relatively new Agile teams. Using theoretical sampling, we concentration on more mature teams towards the later stages of the research. Similarly, we sampled practitioners in a range of different organizational roles such as developers, testers, business analysts, senior management, customers etc in order to get a rounded perspective.

Related Patterns: Data Collection (next pattern)

PATTERN 5: DATA COLLECTION

Context: A geek researcher having found some participants

Example: Sabrina can't wait to have her first interview and observation session. She takes along a notepad and pen and some open-ended questions to get things started. Once at the participant's office, she briefly explains the research aims (grim voice: "*to uncover the main challenges of practitioners performing Agile software development*") gets them to sign an agreement for providing the information and begins. This is so exciting! All sorts of interesting information pour forth. She's struggling to keep pace with the narration and starts to wonder if she should have got an audio-recorder along…

Problem: How do you collect data from participants and when do you stop?

Forces:
- Practitioners may be in another city or country
- Travelling to find participants is expensive.
- Interviewing only particular groups (ex. Managers) may introduce bias in data.
- Relying exclusively on interview data may not be enough.
- A survey-type fixed set of questions for all participants inhibits "real" issues to emerge
- Not using an audio-recorder means researcher spends far more time madly scribbling notes than carrying out a useful conversation with the participants.
- Researcher (especially geeks) may not be comfortable or skilled in the art of interviewing/observing.
- Research needs to be based on a substantial but finite number of participants.

Solution: Conduct semi-structured interviews with open-ended questions. Use an audio recorder if possible. Evolve questions iteratively. Supplement interviews with observations of workplace, practices, and artefacts. Interview people in a range of roles to get multiple perspectives. Apply for travel grants. Combine data collection with conference travel to make your research funds go further.

Conducting semi-structured interviews with open-ended questions allows the participants to describe their concerns and how they go about addressing them. Using audio-recording allows researcher to concentrate more the conversation than on jotting down all details. Audio-recordings can be shared with supervisors later for discussion. Allow questions to evolve and focus on emerging themes from the analysis. Information gained from interviews should be supplemented with observations of workplaces, practices,, and project artefacts as much as possible. Depending on your topic, an appropriate balance between interviews and observations can be achieved. Interviewing people fulfilling different organizational roles provides a rounded perspective of the issues and avoids biases. Collecting data may involve travel if the participants are distanced. Most universities have contestable research funding for such purposes. Combining data collection with conference travel can help maximum the utilization of your research funds. Using Skype can be an alternative if travel is highly constrained. Researcher can stop collecting data when more data does not add any new information or insight into the existing categories – this is called Theoretical Saturation in GT.

Consequences:

+ Using semi-structured interviews with open-ended questions allows emergence of "real" issues.
+ Using an audio-recorder allows researcher to better focus on conversation and to share these with their supervisors afterwards.
+ Interviewing people in a wide variety of different organizational roles fosters understanding issues from a rounded perspective.
+ Supplementing interview data with observations encourages better insight and validation.

- Iterative data collection and analysis allows researchers to sample future participants and evolve interview questions to focus on emerging patterns in data.
- Combining conference travel with data collection allows maximum utilization of research funds.
- Researcher can stop collecting more data once they reach theoretical saturation.
- Transcribing recorded interviews can be a pain (if done on your own) or expensive (if done for you)!
- Acquiring interviewing/observing skills comes with time and practice.
- Collecting data requires travelling – not always trivial/inexpensive.
- Predicting how many participants will be involved upfront is not possible.

Example Resolved: Sabrina is starting to enjoy this whole interviewing business. She's finding it much easier to concentrate on the conversation with her participants since she started using an audio-recorder. She transcribes the recorded interviews once home. Those research grants from her department and the Agile Alliance really helped as she got to go abroad to gather more data.

Known Uses: We applied for and received funding to travel for more data collection. We conducted semi-structured interviews with open-ended questions about an hour long focusing on the participants' major challenges of performing Agile. We observed several Agile practices and workplaces first hand. In order to ensure consistency, all data was collected and analyzed by same person – primary researcher [14-18]. The last few interviews and observations did not add any new information or insight into the existing categories   which was an indication of theoretical saturation. We stopped collecting data at this point. In all we interviewed 58 Agile practitioners from 23 different organizations till we reached saturation. Examples of other researchers conducting data collection can be found in [1,19]

Related Patterns: Sampling (previous pattern), Coding with Words (next pattern)

PATTERN 6: CODING WITH WORDS

Context: A researcher facing mountains of raw data to make sense of.

Example: Sabrina has painstakingly transcribed her first few interviews and field-notes of observations.   How is she supposed to find patterns in this mountain of data?

Problem: How do you perform data analysis?

Forces:

- Amount of raw data from even a few interviews can be daunting.
- Coding in Java may be the only type of coding a geek knows!
- Researcher may not be sure what they are meant to be looking for in the data to begin with.
- Keeping the analysis it too vague may run researcher out of patience.
- Keeping the analysis it too focused on one or two patterns may lead researcher to miss other important patterns.

Solution: Apply Open Coding procedure to analyze your data. The Constant Comparison Method allows patterns across different data sets to be captured. Once the main theme of the research is clear, move to Selective Coding.

Open coding begins by collating key points from every interview transcript. Then a code – a phrase summarizing the key point in 2 or 3 words – is assigned to each key point. The researcher should try to keep their minds free

of ay preconceived ideas while coding – hence the term 'open' coding. The codes arising out of each interview are constantly compared against the codes from the same interview, and those from other interviews and observations. This is GT's constant comparison method *10,14,15+. The constant comparison method is used again to group these codes to produce units of a higher level of abstraction, called concepts in GT. Finally the constant comparison method is repeated on concepts to produce a third level of abstraction called categories [14,15]

Once the main theme of the research is discovered (explained in Pattern #: hitting the core) the researcher can move into Selective coding. Selective coding is the same coding procedures but applied on a limited set of categories – the core category and those closely related to the core. Selective coding typically progresses faster due to a constrained focus.

Consequences:

+ Mountains of data starts to get condensed to several 'codes', some concepts, and a handful of categories.

+ Patterns in the data emerge in form of codes, concepts, and categories

+ Emerging codes, concepts, and categories help provide focus to future interviews/observations.

+ Researcher is able to capture the main concerns of a majority of participants

+ Constantly comparing data across participants is rigorous

− Constantly comparing data across participants is tedious

− Important ideas that are mentioned by one or a few participants may not bubble to higher levels.

− Tacit information may not show up through Coding.

Example Resolved: Sabrina likely keeping her head open and listening to the data 'speak'. She has very little preconceived ideas of the challenges of using Agile methods and starts to find patterns among different participants' data using Open coding. She finds the Constant Comparison method both rigorous and tedious.

Known Uses: We began by conducting Opening coding. Examples of other studies using the Open Coding procedures can be found in [1, 19, 25] Using the constant comparison method, we were able to find codes, concepts, and categories that spanned across the participant base. Examples of the use of the constant comparison method can be found in [1,14. 19, 25]. Later in the research, we moved into Selective coding where we coded only for the core category and other main categories related to the core.

Related Patterns: Data Collection (previous pattern)**, Brain Dump** (next pattern)

PATTERN 7: BRAIN DUMP

Context: A researcher full of reflections about the data derived from participants and observations

Example: Sabrina often finds herself thinking about certain comments by her participants and has a distinct feeling that there are certain relationships between some concepts. Perhaps she should be recording these somewhere.

Problem: How do you capture tacit information and explore relationships between categories?

Forces:

- Researcher needs to explore relationship between different emerging codes, concepts, and categories.

- Researcher may experience the need to jot-down their ideas and perceptions.

- Researcher may find further questions emerging from data analysis that has yet not been answered.

- Tacit knowledge is not captured via coding.

Solution: Write theoretical notes (or memos) throughout the GT process. It allows tacit knowledge such as the relationships between different patterns in the data and any further questions to be explored and recorded explicitly.

Writing down tacit knowledge accumulated by the researcher while conducting interviews, observations, and analysis in the form of memos is called Memoing. Memoing allows the researcher to dump all their ideas, perceptions, thoughts, and further questions of the data on to explicit notes. These memos allow the researcher to explore the relationships between the different concepts and categories which is an important step towards ultimately building a theory. Memoing can be done electronically in order to allow better storage, editing, access, and searching of memos. Saving memos by naming them after the main idea being captured by the memo makes it easier to sort them later (see pattern Sorting it Out).

Consequences:

+ Allows the researcher to explicitly record tacit knowledge (not captured by coding)
+ Allows researcher to explore relationships between different codes, concepts, and categories.
+ Relationships discovered via analysis and memoing helps build a theory eventually.
− Researcher (especially geeks) may not be used to expressing thoughts in words.
− Not done properly, can become a trivial exercise in tracing where the codes originated [25].

Example Resolved: Sabrina starts recording her perceptions, ideas, and further questions about the data and findings as quick 'memos'. Over time, she has many of these catalogued by topics. Sabrina can feel a 'memo' coming a long distance away these days. She stops all she's doing and gets herself to jot-down a memo anytime, anywhere. She finds these memos have considerably improved her insight into different concepts and the relationships between them.

Known Uses: We found Memoing to be a powerful way to allow the relationships between different concepts and later, between different categories, to emerge, as the similarities or differences between

each, or how one effected the other were noted down. We also found it useful to record memos electronically so they can be stored, searched, retrieved, and edited with greater ease [14-18]. Examples of other studies mentioning use of Memoing can be found here [1,19,25].

Related Patterns: **Hitting the Core** (next pattern), **Sorting it Out**

PATTERN 8: HITTING THE CORE

Context: A researcher well into the study, having identified a few main categories in the data.

Example: Sabrina has identified 5 main categories but most of them seem to be centred around or related to one category. She's starting to wonder if she's hit the Core…

Problem: How to go about discerning the main concern of the participants or the 'Core category' from the analysis?

Forces:
- Researcher may get exhausted not knowing what the main research problem is.
- Researcher may be desperate to discover the Core category
- Researcher may be urged to pick a random category!!
- Discovering the Core means reaching the peak of the study in a sense and will mark a descent towards completion (yay!)

Solution: The category that is central, reoccurs frequently, is related to the other main categories, and accounts for most variations in data is the Core. Discerning the core category from a few main categories can be difficult. A category that satisfies these conditions is the Core category: central, reoccurs frequently, is related to the other main categories, and accounts for most variations in data. Some categories may appear to be the core but such categories do not stand the test of the criteria for the core category.

Consequences:

- +   Applying the criteria helps uncover the core category
- −   Some categories may appear to be the core
- −   Discerning the true Core can require heroic amounts of patience

Example Resolved: Using the set of criteria, Sabrina has found her Core category. She is elated! She runs around the department shouting "Eureka!" while bewildered on-lookers half-smile and half-frown.

Known Uses: Discovering the Core in our case was neither quick nor trivial. For about half way through the research, the category lack of customer involvement was one of the most common concerns of the participants and looked promising to be the core. The solution to expose red-herrings (a near-core category appearing to be the core category) is to return to the list of criteria governing the core category. In checking the category lack of customer involvement against the core criteria list, it did not meet all the criteria, in particular it didn't account for most variations in data. It became apparent that lack of customer involvement was not the core category, rather one of the challenges faced by Agile teams in resolving their main concern, the core category: self-organizing Agile teams [14-18].

Related Patterns: Keeping it Open, Brain Dump, Data Abstraction, Getting Choosy (next pattern)

PATTERN 9: SORTING IT OUT

Context: A researcher having reached saturation.

Example: Sabrina's computer has become considerable slower over her PhD years primarily because of all the interview recordings, transcripts, codes, and memos that have *saturated* its memory. The last few interviews she conducted sounded like déjà-vu –which is good news because it means she has reached saturation. Now she can get on with sorting it all out.

Problem: How do you form a theoretical outline?

Forces:

- Sorting involves sorting memos.
- Sorting is done on a conceptual level
- Sorting requires a good collection of memos.

Solution: Sorting involves arranging the memos on a conceptual level once the data collection is saturated. Sorting the memos results in an outline of the theory describing how the different categories relate to the Core category.

Print out all your memos. Lay them out on a large area, sorting them together on a conceptual level. Group memos relating to similar concepts together. Order the groups of memos in an order that best describes the conceptual relationships between them. Capturing these relationships by drawing them out on white boards or papers with lines between related categories helps explain how the categories relate to the core category. This forms a theoretical outline.

Consequences:

+ Sorting is easy with a rich collection of memos which capture most aspects of the findings.
− Sorting in chronological order of when you wrote the memos will not yield a theoretical outline.
− Sorting requires memos to be written in the first place!

Example Resolved: Sabrina prints out all her memos and is busy sorting them out on a conceptual level. With some shuffling around, she's finally got what looks like an outline of her results chapter.

Known Uses: We sorted all our memos by ordering them by concepts they represented. Several memos grouped together under one main heading. Using the white board to diagrammatically capture the relationships suggested by Sorting yields a first look of the theory.

Related Patterns: Brain Dump, What's the Theory, Doc? (next pattern)

PATTERN 10: WHAT'S THE THEORY, DOC?

Context: All that remains is writing it all down!

Example: Sabrina can't believe she's done with all the data collection and analysis. She has a good number of publications in last few years based on her research findings. Writing the thesis, however, is a different ball game as she soon realizes.

Problem: How do you go about presenting a grounded theory?

Forces:
- The final step in GT is writing up the theory.
- The findings of the GT need to be presented in a thesis structure.
- The thesis requires more than just the grounded theory (results).
- The thesis structure and format varies from university to university.
- The typical GT monograph may not be appropriate for a Software Engineering thesis.
- Standard thesis format may not be an accurate representation of how the actual GT research was conducted.

Solution: Familiarize yourself with the required thesis format at your department. The results section of the thesis will present the grounded theory following the theoretical outline generated as a result of Sorting. Expand each of the categories of the theory into chapters or sections. Use participant quotes and textual field notes to highlight emerging points of interest. Explain your research method well – remember others will look up to this as an example of a GT study.

Depending on the university/department guidelines, the structure and format of the thesis may vary. Familiarize yourself with the required format. Most theses will require chapters such as an Introduction, Literature Review, Research Design, Results, Conclusion, References, and Appendices. The Literature Review section can contain the result of the minor review conducted towards the beginning of the research. The major literature review material is better presented as a Discussion section at the end of presenting each of the main categories in the Results chapter. This allows the relevant literature to be discussed in light of the research findings, weaving the two together. Such a structure of presenting Literature Review as minor and major provides a more accurate representation of how the GT study was conducted. Explaining the research method – GT – in detail helps explain the rationale behind some of your presentation/structural decisions and serves as an example of a GT study in Software Engineering.

Consequences:
+ The results chapter of the thesis needs to follow the theoretical outline generated as a result of Sorting.
+ The resulting theory can be summarized in a few sentences
+ The theory can also be represented in a diagrammatic format using the core and related categories.
+ Other chapters can be designed to conform to the required thesis structure and format at your department.
+ The thesis can present the Literature Review in two parts (minor and major) to provide a more accurate representation of how the GT study was conducted.
+ Conference/Journal publications resulting from the research can be weaved into the thesis.
+ Using participant quotes and textual field notes adds depth to the discussions and provides a better sense of grounding of data.
− Using quotations in place of a rich discussion doesn't cut it.

- No or little publications means you have to do all the writing in one go!
- Writer's block

Example Resolved: Sabrina puts all her publications back to back in a file and stands back happily to find a surprising thick document! If only she could present it like this. She downloads the thesis format standard at her department. Looking at other theses in her department suggests a common structure. She tries to conform as much as possible to the standard structure and format while trying to keep the presentation as close to the order in which we went about the research. She spends considerable effort in explaining GT and her application of it using examples. She wants this to be an exemplar GT thesis in the Software Engineering – what she was desperate to find in the beginning of the research herself.

Known Uses: Examples of our presentation of the theory can be found in [14]. Other examples of write of GT studies in Software Engineering context can be found in [1, 5, 7, 19].

Related Patterns: Sorting it Out (previous pattern)

Acknowledgement
We thank our shepherd Micheal Weiss for his insightful comments and feedback.

REFERENCES
1. Adolph et al., *Using Grounded Theory to Study the Experience of Software Development*, Empirical Software Engineering, pp.1-27, 2011.

2. Allan, G. *A critique of using grounded theory as a research method*. EJBRM 2, 1 (2003).

3. Benoliel, J. Q. *Grounded theory and nursing knowledge*. Qualitative Health Research 6, 3 (1996), 406–428.

4. Charmaz, K. *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis (Introducing Qualitative Methods series)*, 1 ed. Sage Publications Ltd, 2006.

5. Coleman, G., and O'Connor, R. *Using grounded theory to understand software process improvement: A study of Irish software product companies.* Information Software Technol. 49, 6 (2007), 654–667.

6. Crabtree, C. A., Seaman, C. B., and Norcio, A. F. *Exploring language in software process elicitation: A grounded theory approach*. In ESEM '09: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement (Washington, DC, USA, 2009), IEEE Computer Society, pp. 324–335.

7. Dagenais, B., Ossher, H., Bellamy, R. K. E., Robillard, M. P., and de Vries, J. P. *Moving into a new software project landscape*. In ICSE '10: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (2010), ACM, pp. 275–284.

8. Elliot, N., and Lazenbatt, A. *How to recognize a 'quality' grounded theory research study*. Australian Journal of Advanced Nursing 22, 3 (2005), 48–52.

9. Glaser, B. and Strauss, A.L., *The Discovery of Grounded Theory*. Adline, Chicago, 1967.

10. Glaser, B. *Theoretical Sensitivity: Advances in the Methodology of Grounded Theory*. Sociology Press, Mill Valley, CA, 1978.

11. Glaser, B. *Doing Grounded Theory: Issues and Discussions*. Sociology Press, Mill Valley, CA, 1998.

12. Glaser, B. *The Grounded Theory Perspective III: Theoretical Coding*. Sociology Press, Mill Valley, CA, 2005.

13. Adolph et al., *Using Grounded Theory to Study the Experience of Software Development*, Empirical Software Engineering, 1-27, 2011.

14. Hoda, Rashina. *Self-Organizing Agile Teams: A Grounded Theory*, PhD Thesis, Victoria University of Wellington, New Zealand, 2011.

15. Hoda et al., *Developing a Grounded Theory to Explain the Practices of Self-Organizing Agile Teams*, Empirical Software Engineering, pp.1-31, 2011.

16. Hoda et al., *The Impact of Inadequate Customer Collaboration on Self-Organizing Agile Teams*, Information and Software Technology, Volume 53, Issue 5, May, 2011

17. Hoda, R., Noble, J., and Marshall, S. *Using grounded theory to study the human aspects of software engineering*. In Human Aspects of Software Engineering (HAoSE) workshop at SPLASH2010 (Reno/Nevada, USA, 2010), ACM, pp. 5:1–5:2.

18. Hoda et al., *Organizing Self-Organizing Teams*, International Conference on Software Engineering, Cape Town, 2010.

19. Martin, Angela, *The Role of Customers in Extreme Programming Projects*, PhD Thesis, Victoria University of Wellington, NZ, 2009.

20. Myers, M. D. Qualitative research in information systems. MIS Quarterly 21, 2 (1997), 241–242.

21. Nathaniel, K. A. *A Grounded Theory Of Moral Reckoning* In Nursing. PhD thesis, West Virginia University, 2003.

22. Parry, K. *Grounded theory and social process: A new direction for leadership research*. Leadership Quarterly 9, 1 (1998), 85–105.

23. Stern, P. N. *Eroding grounded theory*. Critical Issues in Qualitative Research Methods (1994), 210–223.

24. Suddaby, R. *From the editors: What grounded theory is not*. Academy of Management Journal 49, 4 (2006), 633–642.

25. Georgieva, S., and Allan, G. *Best practices in project management through a grounded theory lens*. Electronic Journal of Business Research Methods 6, 1 (2008), 43–52.

26. Creswell, J.W. *Research Design: qualitative, quantitative, and mixed methods and approaches* (2nd edition). Sage Publications (2003)

27. Oates, B.J. *Researching Information Systems and Computing*. Sage Publications (2006)