Developing and Using a Codebook for the Analysis of Interview Data: An Example from a Professional Developme....

Article *in* Field Methods · May 2011

DOI: 10.1177/1525822X10388468

CITATIONS
194

READS
21,119

3 authors:

Jessica T. DeCuir-Gunby
North Carolina State University
30 PUBLICATIONS **799** CITATIONS
SEE PROFILE

Patricia L. Marshall
North Carolina State University
20 PUBLICATIONS **306** CITATIONS
SEE PROFILE

Allison W. Mcculloch
North Carolina State University
9 PUBLICATIONS **227** CITATIONS
SEE PROFILE

Field Methods
23(2) 136-155

DOI: 10.1177/1525822X10388468
http://fm.sagepub.com

Developing and Using a Codebook for the Analysis of Interview Data: An Example from a Professional Development Research Project

Jessica T. DeCuir-Gunby[1], Patricia L. Marshall[1], and Allison W. McCulloch[2]

Abstract

This article gives specific steps on how to create a codebook for coding interview data. The authors examine the development of theory- and data-driven codes through the discussion of a professional development (PD) research project. They also discuss how to train others to code using the codebook, including how to establish reliability. The authors end with practical suggestions from their experiences in creating a codebook.

[1] Department of Curriculum, Instruction, and Counselor Education, NC State University, Raleigh, NC, USA
[2] Department of Mathematics, Science, and Technology Education, NC State University, Raleigh, NC, USA

Corresponding Author:
Jessica T. DeCuir-Gunby, Department of Curriculum, Instruction, and Counselor Education, NC State University, Campus Box 7801, Raleigh, NC 27695, USA
Email: jessica_decuir@ncsu.edu

Keywords
codebook, coding, interviews, team-based research

Analyzing interview data is a multistep "sense-making" endeavor. To make sense of interviews, researchers must engage in the process of coding data. Although coding interviews is widely recognized as a common step in the interview analysis process, many researchers do not fully explicate how this is done. In addition, experts in qualitative methodology have not established a universally agreed on set of coding procedures that can be easily replicated (Coffey and Atkinson 1996). Because of this, many novice researchers are not certain of what procedures to use in coding interview data or how to begin using such procedures. Qualitative researchers often discuss the use of a codebook as one of the initial, and arguably the most critical, steps in the interview analysis process (Fereday and Muir-Cochrane 2006). Many articles and book chapters describe and demonstrate the different steps involved in the codebook development process (e.g., MacQueen et al. 1998; Ryan and Bernard 2000; Franklin and Ballan 2001; Fonteyn et al. 2008; MacQueen et al. 2008; Laditka et al. 2009; Bernard and Ryan 2010).

The goal of this article is to continue this conversation by showing how to create and use a codebook as a means of analyzing interview data, using real-world education data. We begin with a basic discussion of codes, codebooks, and coding, followed by a description of our professional development (PD) research project. Using our research project as a real-life example, we demonstrate how to create a codebook by discussing the development of both theory- and data-driven codes. Additionally, we address training others to use the codebook and establishing interrater reliability. We conclude with practical suggestions about the process of creating a codebook.

Codes, Codebooks, and Coding

Codes are defined as "tags or labels for assigning units of meaning to the descriptive or inferential information compiled during a study" (Miles and Huberman 1994: 56), and their development[1] is the initial step in analyzing interview data. To ensure meaningful labels, codes are assigned to chunks of data, usually phrases, sentences, or paragraphs that are connected to a specific context or setting (Miles and Huberman 1994). Codes can be developed a priori from existing theory or concepts (theory-driven); they can emerge from the raw data (data-driven); or they can grow from a specific

project's research goals and questions (structural), with most codes being theory- or data-driven (Ryan and Bernard 2003). The development of theory-driven codes typically requires constant revisiting of theory, whereas data-driven and structural codes necessitate repeated examination of the raw data. Thus, code development is an iterative process.

A codebook is a set of codes, definitions, and examples used as a guide to help analyze interview data. Codebooks are essential to analyzing qualitative research because they provide a formalized operationalization of the codes (MacQueen et al. 1998; Crabtree and Miller 1999; Fereday and Muir-Cochrane 2006; Fonteyn et al. 2008). Even so, like codes, codebooks are developed through an iterative process that may necessitate revising definitions as the researchers gain clearer insights about the interview data. The more specificity in a codebook, the easier it is for coders to distinguish between codes and to determine examples from nonexamples of individual codes. In addition, the more detailed the codebook, the more consistency there will be among coders when using it to code interviews. Thus, Macqueen et al. (1998) suggest that the structure of codebooks should consist of six components, including the code name/label, brief definition, full definition, inclusion criteria, exclusion criteria, and examples. However, in this case, we have chosen to structure our codebook using three components: code name/label, full definition (an extensive definition that collapses inclusion and exclusion criteria), and an example.

The actual process of coding is an integral part of the interview data analysis process. Coding is the assigning of codes (that have been previously defined or operationalized in a codebook) to raw data. This allows researchers to engage in data reduction and simplification. It also allows for data expansion (making new connections between concepts), transformation (converting data into meaningful units), and reconceptualization (rethinking theoretical associations; Coffey and Atkinson 1996). Further, through coding, researchers make connections between ideas and concepts. Applying codes to raw data enables the researcher to begin examining how their data supports or contradicts the theory that is guiding their research as well as enhances the current research literature. Coding is, in essence, a circular process in that the researcher may then revisit the raw data based upon theoretical findings and the current research literature. See Figure 1 for a visualization of the coding process.

According to Corbin and Strauss (2008), there are two major levels of coding—open coding and axial coding. When beginning to code interview data, the first step is to engage in the process of open coding or "breaking data apart and delineating concepts to stand for blocks of raw data" (Corbin and Strauss 2008:195). Open coding allows for exploration of the ideas and

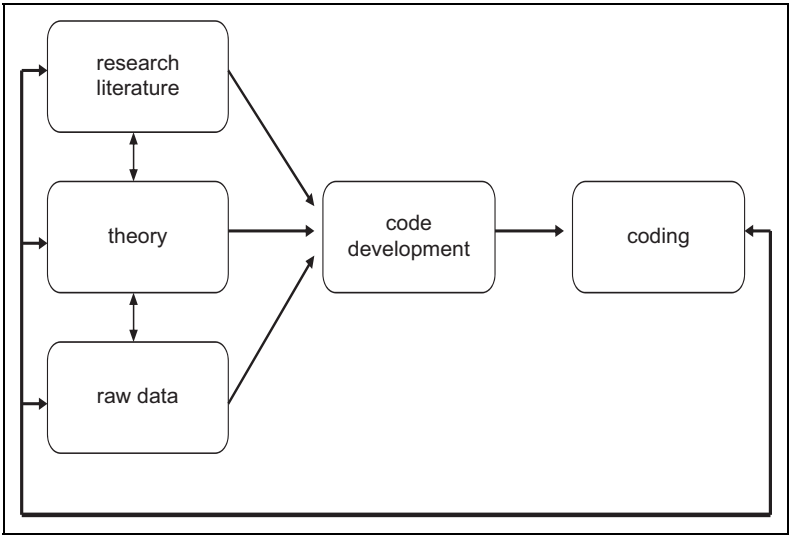

Figure 1. Circular process of coding.

meaning that are contained in raw data. While engaging in open coding, the researcher creates codes or concepts. Once codes have been created using open coding, it is necessary to analyze them through the process of axial coding. This higher level of coding enables researchers to identify any connections that may exist between codes.

When beginning the analysis process, inexperienced qualitative researchers are likely to have many questions, including the central question: "How do I create a codebook?" However, another question they *should* ask is, "What role does theory play in the creation of a codebook?" Similarly, once a codebook has been created, they may discover they need to ask, "How do I train others to use a codebook?" Questions such as these can frustrate and stymie the efforts of beginning researchers. Therefore, for the remainder of this article, we respond to these questions and describe how we created a codebook for analyzing interview data as part of our multiyear funded research project.

Nurturing Mathematics Dreamkeepers: A PD Research Project

Nurturing Mathematics Dreamkeepers, also known as NMD, was a multiyear, PD research project that involved the study of 65 kindergarten, first,

and second grade teachers and their students. Briefly, this project explored how teachers understand and adopt standards-based teaching practices (National Council of Teachers of Mathematics 2000) that promote young children's conceptual understanding in early mathematics. Enrollment in the project occurred at the start of each of three consecutive academic years (2005–2008), and teachers were required to attend our extensive PD intervention that was organized as four 2-day retreats (approximately 90 hours) spread over the course of each year the individual teacher participated in the project.[2]

In addition to participation in the PD retreats, teachers were required to complete the project instruments (e.g., Teacher Dispositions Questionnaire), be videotaped (by a project research assistant [RA]) teaching eight different mathematics lessons occurring at preselected intervals throughout the academic year, observe a same grade peer (also participating in the project) teaching mathematics lessons, and participate with that same grade peer in [post-teaching] reflection sessions led by an RA. Finally, each project teacher was required to participate in a one-on-one interview at the start and conclusion of each project year. Teachers were paid a $1,000 stipend for each year they participated in the study.

The PD component of NMD was designed to facilitate teachers' critical understandings of the impact of culture on the teaching–learning process. To this end, we attempted to promote understanding of cultural relevance (Ladson-Billings 1994) as a pedagogical orientation as well as incorporation of its broad tenets into mathematics teaching and post-instructional reflections. The goal was for teachers to adopt culturally relevant pedagogy as part of their professional identities, which we theorized would, in turn, impact their orientations toward the issue of equity not only in their mathematics instruction but in their approach to the teaching–learning process in general. In addition, a goal was to promote deep mathematical understanding by analyzing how students outside of school experiences with mathematics impacted their formal conceptions. This part of our study relates to the notion of *conception-based perspective* in mathematics teaching. A conception-based perspective characterizes teachers who operate from the assumption that a student's mathematical reality is not independent of that student's ways of knowing and acting, that what a student sees, understands, and learns is constrained and afforded by what that student already knows, and that mathematical learning is a process of transformation of one's knowing and ways of acting (Simon et al. 2000).

Although the corpus of data for our project was generated from a variety of sources, this discussion focuses exclusively on the various steps

we employed in preparing to analyze the teacher interviews. These semistructured interviews (Rubin and Rubin 2005) were used to help us gain a better understanding of individual teachers' dispositions regarding K–2 mathematics and the role or place of culture therein. All interviews were 30 minutes to 1 hour in length and were conducted at free periods before, during, or after school at each teacher's school campus. A total of 145 interviews were conducted, with teachers having participted in 2–6 interviews, depending on their cohort membership. The next section describes how the theoretical base of our study and themes that emerged from the interviews were utilized to create an interview codebook.

Creating a Codebook

As previously mentioned, codes are created from three major areas including theory (theory-driven), data (data-driven), and research goals (structural). In the case of NMD, only theory- and data-driven codes were created to assist in the coding of interviews. Boyatzis (1998) indicates that there are separate procedures for creating theory- and data-driven codes. Developing theory-driven codes involve three steps: (1) generate the code; (2) review and revise the code in context of the data; and (3) determine the reliability of coders and the code. Data-driven codes, on the other hand, involve five steps to inductively create codes for a codebook: (1) reduce raw information; (2) identify subsample themes; (3) compare themes across subsamples; (4) create codes; and (5) determine reliability of codes. We will use Boyatzis's framework to demonstrate the steps we used to create theory- and data-driven codes and codebook definitions. (See Figure 2 for a visual of the steps for creating a codebook.)

How Do You Develop Theory-Driven Codes?

The first step in developing theory-driven codes is to create codes. Codes are generated from the theories that guide the research. In the case of NMD, our theory-driven codes were developed from culturally relevant pedagogy and a conception-based perspective. In determining the potential codes, the three principal investigators (PIs) had a series of discussions regarding the theoretical frameworks that guided the study. We met for 3 hours, once a week, for 3 months, for a total of 36 hours. The PI team itself included diverse scholarly expertise and this diversity resulted in divergent interpretations. We constantly challenged each other's perspectives and interpretations, which demanded that we offer clear, logical, and rational

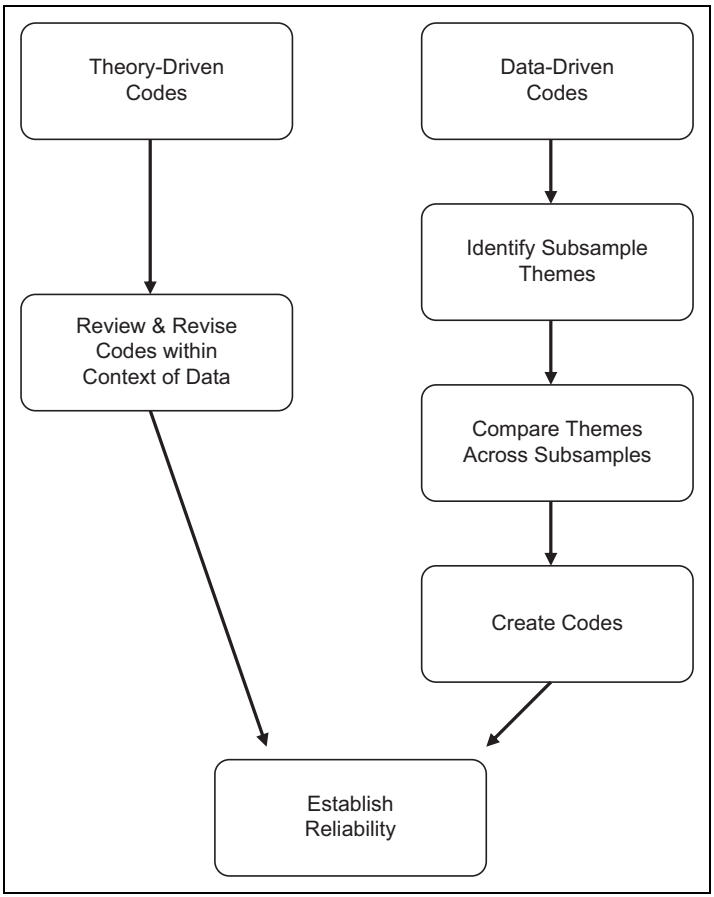

Figure 2. Steps for developing a codebook.

explanations and responses to the various questions that arose about the different aspects of the process. This lessened the possibility of groupthink in that we discussed each topic until we had a consensus.

Our conversations explored the relationships between culturally relevant pedagogy and a conception-based perspective and how these relationships can be captured through codes. It was difficult to reduce intricate theories such as culturally relevant pedagogy and a conception-based perspective to a few words because, in many cases, we had to think about how to best operationalize abstract concepts. Thus, it must be stated that creating

succinct theoretical codes is an arduous task that takes a considerable amount of time. Once the key concepts and their relationships were determined, the PIs began suggesting possible codes and definitions for the codebook.

The second step in developing theory-driven codes is reviewing and revising the codes in context. This necessitated discussing the appropriateness of the code labels and how they were to be applied to the data. Our goal was to create code labels, as suggested by Boyatzis (1998), which were conceptually meaningful, clear and concise, and close to the data. Doing so required revising several of the code labels. For example, we created the code *perception-based orientation*, a subconcept based on our conception-based framework. Initially, we interpreted this code as reflecting a particular instructional orientation or belief about mathematics teaching. However, once we attempted to assign this code to the actual interview text, we realized that the teachers' statements were examples of perception-based thoughts and not indications that their pedagogical approach to teaching mathematics was from a perception-based perspective. Thus, we changed the code label to *perception-based referencing*, because the teachers were discussing perception-based ideas rather than explicitly describing their beliefs about how children learn mathematics. We could not assume that making a statement consistent with a perception-based orientation meant that a teacher used a perception-based pedagogy in the classroom. Such a claim could be made only through comprehensive analyses and triangulation of interviews, reflection sessions, and classroom observations.

In addition to focusing on our code labels, we had to make sure that our definitions were specific, yet encompassing of the constructs we were trying to capture. To create comprehensive definitions, we engaged in several iterations of our interview definitions. For example, we originally defined *perception-based referencing* as:

> Characterizes teachers who realize that students must actively participate in the learning process, but believe that relationships exist "out there" and students just need to "see" them as the teacher "sees" them. For example, a teacher may use base-10 blocks to represent operations in a place-value and base-10 number system as if the abstract mathematical relationships exist 'out there'. Yet, the teacher uses the blocks because to understand, learners must actively "see" the mathematical relationships for themselves. It implies using first hand activities that can best *reveal* to student the mathematics *as the teacher sees it*.

After attempting to code the data utilizing this definition, we recognized that the definition contained too much extraneous information that could potentially cause interpretation problems for future coders. Thus, we reduced the definition to capture the essential elements of the code, resulting in the following:

> Teacher states or alludes to a belief that using activities in which students are directly engaged can best *reveal* to students the mathematics *as the teacher sees it*.

Once we agreed on code labels and definitions, we selected example quotes within the data that best illustrated each code. For instance, much of the data labeled as *perception-based referencing* alluded to teaching in a "hands-on" fashion and/or the necessity of using "manipulatives" for students to learn and understand mathematical concepts. Therefore, we chose an example that mentioned that K–2 mathematics should have hands-on elements:

> Basically, I think it [teaching] has to be *hands-on*. I think especially at this low level [first grade], they've got to have a concrete understanding of what a number looks like. You know, how much is seven? What's the difference between seven and two? When I compare the two, can I visually see what they look like?

The last step in developing theory-driven codes is determining reliability, including discussing utility and implementation. (An in-depth discussion on training others to code that includes establishing reliability is provided later in this article.) The PIs individually practiced coding interviews with the theory-driven codes and met to discuss individual findings. We discovered that coding individually resulted in multiple interpretations for the data and revealed inconsistencies in the coding protocol. We changed to coding as a group, which allowed us to move forward by sharing our reasons for utilizing codes in particular ways. In addition, coding as a group also afforded us the opportunity to explore examples and non-examples of the codes. These in-depth conversations were very enriching, allowing us to reach consensus on the coding procedure protocol. See Table 1 for a sample of final theory-driven codes, definitions, and examples.

How Do You Develop Data-Driven Codes?

The first step in developing data-driven codes is to determine how to reduce raw information into smaller units, such as categories or themes. We

Table 1. Sample Theory-Driven Codes, Definitions, and Examples

Code	Description	Example
Conception-based reference	Teacher states or alludes to a belief that *learners must construct meanings* for mathematical ideas *on the basis of the learner's existing conceptions* that may be quite different from those of the teacher	"Because that then takes—allows the children to use all different strategies to apply and figure out an answer. And there's not that one right answer, which I think is important for me to get kids away from. There's more than one way. It's okay you and I didn't do it the same way."
Cultural referencing	Teacher makes direct/indirect reference to specific elements of students' culture/background (e.g., race, socioeconomic status, language, other outside of school experiences, etc.) that may impact the teaching–learning process	"I think you can see it in a child who—for lack of a better term, is street-wise. You know, they understand what a concept of a number. Like if you have $5 and you know that you can buy X, Y, and Z with $5. Then they know five is more than two."
Procedural understanding description	Teacher describes or gives examples of what she believes characterizes procedural understanding.	"So I think procedure is just a rote kind of thing but you don't [know how] it works but you just do it. That's all you know."

discussed the possibility of coding line by line, on the sentence level, on the paragraph level, or by what we labeled the "level of meaning." After reading several interviews, we realized that coding line by line and on the sentence level were often not meaningful. The paragraph level, on the other hand, often featured a variety of themes, making it impossible to label with only one code. Based on this, we decided to focus on the level of meaning. From this perspective, the "lumping" and "splitting" of text could occur at different locations, enabling a code to be made up of a line, sentence, or paragraph, as long as the essence is the same (MacQueen et al. 2008). However, we agreed a separate code was warranted when the unit of analysis could "stand on its own" and convey meaning outside of the larger context of the interview. This same rule applied to the implementation of theory-driven codes.

The process we followed in developing data-driven codes involved identifying themes within subsamples. This meant identifying themes from various interviews.[3] As we read several interviews, we looked at the major themes that emerged per interview that had not been captured by the theory-driven codes. We then began to complete the codes per teacher interview. For example, we noticed how a teacher discussed her difficulty in teaching certain mathematics topics. We were not certain if her description fit under the code *teaching strategies*, which we defined as the following: "Teacher explains how/why she teaches in a certain way, her choice of lesson activities, her use of Differentiated Instruction." However, we felt these difficulties were important to capture.

As a result, we looked for themes across teacher interviews and saw that various teachers discussed not understanding or having difficulty with or experiencing general struggles associated with teaching. This was a consistent theme across teacher interviews. Several teachers made comments such as the following:

> I have a hard time figuring out what is developmentally appropriate. Where do I take them this early in the year? Where should they be by the end of the year? Okay, so they should know numbers 1 to 99. But what do I do to teach them that? And teach place value. And I'm not—I mean, I can teach it, but I'm just struggling with when I teach it. When is it appropriate to teach it? When do I introduce things?

Then we used the information gained through identifying and comparing teacher interviews to establish a way to capture their difficulty with teaching mathematics. This required reexamining the theory-driven code, *teaching strategies*, to determine if that code needed to be expanded or whether a new code had to be created. We agreed to create a new code and considered using an "in vivo" label or a label created from the actual words of a participant (Glaser and Strauss 1967). One participant used the phrase, "my misconceptions" to describe her difficulties teaching. Although we liked the *idea* of using that phrase as an in vivo label, it was neither comprehensive nor descriptive enough to capture the perspectives of others. Therefore, we agreed to create the code *pedagogical struggles* and we defined it accordingly: "Teacher expresses uncertainty, lack of clarity, and/or concern, about some aspect of the 'how, what, or when' of classroom practice."

The final step used in developing data-driven codes was to determine the utility/reliability of the codes using them to begin the analysis process. (Again, a full discussion of reliability will follow.) We followed the same

Table 2. Sample Data-Driven Codes, Definitions, and Examples

Code	Description	Example
Other influences on teachers	Teacher refers to influences on her practice and/or thinking (e.g., former professors, colleagues, students, other professional development experiences, etc.) excluding NMD	"You know, one of my professors at Meredith had the saying—and I've kind of forgotten it except the last part that said children can't understand math' til they hold it in their hand. And that has kind of been my guiding force the, you know, the years I've been teaching."
Curricular references	Teacher makes direct/indirect or general/specific references to curriculum (e.g., Standard Course of Study, pacing guides, Trailblazers, Every Day Math, etc.)	"At my grade level, I think I know the curriculum"
Pedagogical struggles	Teacher expresses uncertainty, lack of clarity, and/or concern, about some aspect of the "how, what, or when" of classroom practice	"But my concern is, you know, two years down the road, is there going to be some stepping-stone that we've missed, that's going to put that concrete fact . . . the child doesn't have. So that's probably my biggest concern."

procedures that were used for the theory-driven codes, including practicing coding individually and as PIs to synchronize our orientations to the process. In addition, we discussed examples and non-examples of the codes. Finally, we reexamined the data-driven codes in relation to the theory-driven codes to identify and eliminate any overlap. See Table 2 for a sample of data-driven codes.

How Do You Train Others to Code?

Because of the size of our project, after we created the codebook we had to train six RAs how to use it to code data. We provided our RAs with an

overview of the process, informing them that their input would be unique, and therefore truly valued. Further, we informed them that everyone's (including the PIs) interpretations of codes, as well as everyone's application of codes to any given data, could potentially be questioned and thereby subjected to critical analysis by any other member of the research team. As co-PIs, we modeled this process of questioning each other's interpretations by making the RAs privy to our initial code development process, including disagreements that emerged among us as project leaders. Also, we encouraged the RAs to question each other's interpretations and applications of the codes. The RAs practiced and honed their coding skills using interviews of various lengths. Similar to the process used by the PIs, the RAs coded the interviews individually as well as collectively and shared their thinking behind their coding as a group.

This entire process involved 2-hour weekly meetings over the course of 3 months, for a total of 24 hours. During this time, we specifically focused on code names, definitions, examples, and non-examples. Code definitions were written in simple, straightforward language. When we used project vernacular, we invited RAs to pose questions about meaning and, where appropriate, we substituted terminology for simpler, easier language. We also discussed the coding process, including how to determine when a code begins, when it ends, and the possibility of multiple coding, applying two or more codes to the same text (Bogdan and Biklen 2003). For our project, RAs were informed there would be situations when two or more codes were applicable to a specific segment of text; however, they were cautioned to use multiple coding sparingly and that certain codes would more likely to be used for multiple coding. For example, the code *teaching strategies* often accompanied the code *perception-based referencing* because for teachers to describe their general beliefs regarding how students learn (perception based), they often provided examples of what they did in class (teaching strategies).

In addition to learning how to code using the codebook, we also focused on the use of Computer Assisted Qualitative Data Analysis Software (CAQDAS). The entire research team attended structured sessions on how to store, code, and retrieve data using Atlas.ti, a type of CAQDAS. A colleague with proficiency in using Atlas.ti conducted two training sessions. The first session focused on the fundamentals of using Atlas.ti; the second session concentrated on coding using the project data. At these sessions, each PI posed questions and raised issues that could potentially emerge as the RAs were assigned to use the software in actual interview coding.

How Do You Establish Reliability?

Using multiple coders to analyze interview data necessitates establishing interrater reliability or the consistency in scoring between multiple raters (see Morse et al. 2002; Saldaña 2009). Although there are several statistical techniques for measuring interrater reliability (see Krippendorff 2004), there are three major approaches used in content analysis. The first approach is to calculate a basic proportion of agreement. Miles and Huberman (1994) suggest calculating reliability as the number of agreements divided by the total number of agreements + disagreements. A reliability of 90% or better is necessary for maximum consistency of coding. The second approach is to use the Pearson product–moment correlation coefficient (using continuous variables) or Spearman's rank correlation coefficient (using ordinal variables). Pearson's and Spearman's statistics measure pairwise correlations among raters. The closer the score is to 1, the stronger the correlation or interrater reliability. The third and most popular approach to calculating reliability is to use Cohen's kappa coefficient (Cohen 1960) or Fleiss's kappa statistic (Fleiss 1971) for use with nominal variables. The goal of Cohen's (to be used with two raters) and Fleiss's kappa (to be used with three or more raters) is to determine the consistency in ranking items or classifying items into mutually exclusive categories.

Both Cohen's and Fleiss's kappas are calculated by determining the amount of actual agreement divided by the amount of agreement expected by chance; they are scored between 0 and 1. Although there is no universally agreed on metric, Landis and Koch (1977) suggest that scores less than 0 represent poor agreement, scores of 0.0–0.20 are slight agreement, scores of 0.41–0.60 represent moderate agreement, scores of 0.61–0.80 are substantial agreement, and scores of 0.81–100 signify almost perfect agreement. However, it must be added that there are more versatile measures of reliability including measures that can be used with nominal, ordinal, and interval variables such as Krippendorff's alpha (see Krippendorff 2004).

For our project, we were interested in determining reliability in terms of *which* codes the RAs used as well as *how* the codes were used. This was especially difficult, given the large size of our research team. In light of this focus, we realized that each of the major approaches were problematic in a variety of ways. For example, calculating a basic proportion as suggested by Miles and Huberman (1994) does not illustrate how the various codes are being applied. In addition, Pearson's and Spearman's statistics do not take the magnitude of the differences between raters into account; consequently,

raters can have little to no agreement on particular items but still obtain a relatively high correlation. Also, with Cohen's/Fleiss's kappa coefficients, the percentage of agreement decreases with both the number of coders and categories added (Krippendorff 2004). Given the size of our research team and the number of categories (codes), the probability that we would get a high percentage agreement was very low.

Because we wanted to focus on both the quantity of codes (what) and the quality of codes (how), we extended the Miles and Huberman's (1994) approach and created our own process, which focused more on group consensus (Harry et al. 2005). This required us to code hard copies and calculate reliability by hand. (We didn't begin using the CAQDAS to code until the codebook was complete.) In doing so, we acknowledge that utilizing a statistical calculation such as Cohen/Fleiss's kappa or Krippendorff's alpha would have allowed for a more robust calculation of reliability, thus adding credibility to our findings.

In engaging in our process, we first decided to consider the consistency of labeling text with *each* code. The RAs coded several pages of an interview at a time, followed by a discussion of when and how specific codes had been applied. Codes that were applied by all RAs with no variations were considered to be 100% agreement among the RAs. After we determined the codes that were more easily and consistently identifiable (e.g., *NMD referencing*, *NMD Buddy/Reflection*, and *Curricular referencing*), we then honed in on the codes that were being applied less consistently. For example, the code *teacher identity* proved to be problematic. Teacher identity was used to capture an individual teacher's description of how she sees herself professionally, culturally, and/or mathematically, including references to experiences that she acknowledges as having influenced her sense of self in either of these realms. After careful deliberation, we decided the *teacher identity* code was developed to answer the central question, "Who am I?" rather than a general awareness. This led us to redefine the *teacher identity* code. Similarly, we discussed and redefined all other problematic codes. Next, we practiced coding using the new definitions of the codes until the RAs had 100% agreement. Finally, we engaged in our process of checking reliability at the beginning and at least one time during the data analysis process to make sure that coding remained consistent.

Conclusion

Developing a codebook is a challenging process, and for our team, the entire process, including code creation and coder training, took over a semester to

accomplish. Our final codebook consisted of 18 codes, including 10 theory-driven and 8 data-driven codes. Based on our experience, we have several suggestions to offer other researchers who may embark on such a task.

1. *Creating a codebook should be a team effort*. The process of creating a codebook is complex and tedious, and, because of all the various components, it can easily become an overwhelming challenge if undertaken by one person. To lessen the challenge, we highly recommend forming a codebook creation team, the members of which bring divergent viewpoints and (if possible) varying degrees of familiarity with the actual research project. Moreover, we recommend that the team leaders make a deliberate effort to create an atmosphere between and among the members that encourages and values critical questioning and constructive criticism. Researchers should be careful to formulate a team that strikes the most useful balance between divergent viewpoints and efficient task completion. Ultimately, researchers should remember that the more people involved in the process, the more divergent viewpoints will emerge. The more viewpoints, the greater the need for reconciliation and the longer the process.
2. *Developing a codebook is time intensive.* Many steps are necessary to create a codebook and to teach others how to use the codebook, all of which are time consuming. To reiterate, the PIs engaged in 36 hours to create the codebook and it took 24 hours to train the RAs on how to use the codebook, for a total of 60 hours of codebook development and training. Developing a codebook often requires revisiting codes and reexamining data. Because of this, researchers have to become comfortable with uncertainty and with the iterativeness of the process.

   In addition, the actual coding of text was time intensive. Because of the complexity of mathematics concepts being addressed, the lengths of the interviews varied; kindergarten teachers often had shorter interviews (approximately 30 minutes), whereas the first and second grade teachers had longer interviews (approximately 40–50 minutes). Coding often took 1½–2 times the length of the interview. With a total of 145 interviews on average lasting 40 minutes each, it is estimated that the coding of all of the interviews took around 145–193 hours. As illustrated by the previously discussed time commitments, it is important to keep the notion of *time* in mind when planning your research time line.

3. *Theory should play a critical role in the creation of a codebook.* Creating codes and subsequently coding interview data is about meshing all of the theoretical underpinnings of a study with the data that has been generated by the study. Thus, a critical feature of codebook creation is team engagement in conversations about the theory. Ironically, "theoretical conversations" happen to constitute a feature of the larger codebook development process that, for some researchers, can seem annoyingly protracted because of the speculative nature of theory itself. Recognizing this, we urge researchers to prepare for and to resist discomforts that may emerge in conjunction with theoretical conversations. This is because the core of crafting a codebook is about highlighting the theory that impacts the study and creating a firmer foundation for the research. It is also about determining how to operationalize theory and how to turn the abstract into the tangible.
4. *Training the RAs to use the codebook should be a systematic and structured process* The codebook should be clear to read, simple to follow, and easy to implement. In addition, there should be clear steps to help RAs to understand the codes, how to apply them while coding, and how to use qualitative software. To maintain a straightforward and structured process, we recommend that researchers check frequently to make sure all procedures are understood. They should also intermittently offer sincere compliments to individual members on their efforts and assure all members that when working as a team to create a useful codebook, no question is too insignificant to ask.

It is important to remember that creating a codebook is just the beginning of the coding process. The next steps are data analysis and interpretations (for suggestions, see Wolcott 1994; Coffey and Atkinson 1996; Saldaña 2009). We have shared here how we went about the multistep process of creating a codebook, which is often presented as relatively simple when discussed in the literature. We hope that we have shown that it is a complicated and complex process that is necessarily very tightly tied to theory and data.

Authors' Note

Any opinions, findings, and conclusions or recommendations reported here are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Declaration of Conflicting Interests

The author(s) declared no conflicts of interest with respect to the authorship and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research and/or authorship of this article: The research reported in this article was supported by a grant from the National Science Foundation (award #0353412).

Notes

1. In the case of grounded theory, codes are also referred to as concepts. For more information on grounded theory, see Glaser and Strauss (1967) and Corbin and Strauss (2008).
2. The intervention for each cohort was different each year of participation, with each cohort experiencing the same first-year intervention, and Cohorts I and II experiencing the same second-year experience. The third-year experience for Cohort I teachers involved collaboration with the researchers in delivering the intervention to teachers in Cohorts II and III.
3. We utilized the interviews of former project participants to practice identifying and confirming our codes. We are aware that not all projects will have this option. If this is the case, practice interviews will have to be recoded using the final codebook.

References

Bernard, H. R., and G. W. Ryan. 2010. *Analyzing qualitative data: Systematic approaches.* Thousand Oaks, CA: SAGE.

Bogdan, R., and S. K. Biklen. 2003. *Qualitative research for education: An introduction to theory and methods.* 4th ed. Boston: Allyn and Bacon.

Boyatzis, R. 1998. *Transforming qualitative information: Thematic analysis and code development.* Thousand Oaks, CA: SAGE.

Coffey, A. J., and P. A. Atkinson. 1996. *Making sense of qualitative data: Complementary research strategies.* Thousand Oaks, CA: SAGE.

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20:37–46.

Corbin, J., and A. L. Strauss. 2008. *Basics of qualitative research.* 3rd ed. Thousand Oaks, CA: SAGE.

Crabtree, B. F., and W. L. Miller. 1999. *Doing qualitative research.* Thousand Oaks, CA: SAGE.

Fereday, J., and E. Muir-Cochrane. 2006. Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International Journal of Qualitative Methods* 5:1–11.

Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76:378–82.

Fonteyn, M. E., M. Vettese, D. R. Lancaster, and S. Bauer-Wu. 2008. Developing a codebook to guide content analysis of expressive writing transcripts. *Applied Nursing Research* 21:165–68.

Franklin, C. S., and M. Ballan. 2001. Reliability and validity in qualitative research. In *Handbook of social work research methods,* ed. B. A. Thyer, 273–92. Thousand Oaks, CA: SAGE.

Glaser, B. G., and A. L. Strauss. 1967. *Discovery of grounded theory: Strategies for qualitative research.* Hawthorne, NY: Aldine De Gruyter.

Harry, B., K. Sturges, and J. Klinger. 2005. Mapping the process: An exemplar of process and challenge in grounded theory analysis. *Educational Researcher* 34:3–13.

Krippendorff, K. 2004. Reliability in content analysis: Some common misconception and recommendations. *Human Communication Research* 30:411–33.

Laditka, S. B., S. J. Corwin, J. N. Laditka, R. Liu, D. B. Friedman, A. E. Mathews, and S. Wilcox. 2009. Methods and management of the healthy brain study: A large multisite qualitative research project. *The Gerontologist* 49:S18–S22.

Ladson-Billings, G. 1994. *The dreamkeepers: Successful teachers of African American children.* San Francisco, CA: Jossey-Bass.

Landis, J. R., and G. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics* 33:159–74.

MacQueen, K., E. McLellan, K. Kay, and B. Milstein. 1998. Codebook development for team-based qualitative analysis. *Cultural Anthropology Methods* 10:31–36.

MacQueen, K., E. McLellan-Lemal, K. Bartholow, and B. Milstein. 2008. Team-based codebook development: Structure, process, and agreement. In *Handbook for team-based qualitative research,* eds. G. Guest and K. M. MacQueen, 119–35. Lanham, MD: AltaMira.

Miles, M. B., and A. M. Huberman. 1994. *Qualitative data analysis: An expanded sourcebook.* 2nd ed. Thousand Oaks, CA: SAGE.

Morse, J. M., M. Barrett, M. Mayan, K. Olson, and J. Spiers. 2002. Verification strategies for establishing reliability and validity in qualitative research. *International Journal of Qualitative Methods* 1:1–19.

National Council of Teachers of Mathematics. 2000. *Principles and standards for school mathematics.* Reston, VA: NCTM.

Rubin, H. J., and L. S. Rubin. 2005. *Qualitative interviewing: The art of hearing data.* 2nd ed. Thousand Oaks, CA: SAGE.

Ryan, G. W., and H. R. Bernard. 2000. Data management and analysis methods. In *The handbook of qualitative research*, 2nd ed., eds. N. K. Denzin and Y. S. Lincoln, 769–802. Thousand Oaks, CA: SAGE.

Ryan, G. W., and H. R. Bernard. 2003. Techniques to identify themes. *Field Methods* 15:85–109.

Saldaña, J. 2009. *The coding manual for qualitative researchers.* Thousand Oaks, CA: SAGE.

Simon, M. A., R. Tzur, K. Heinz, M. Kinzel, and M. S. Smith. 2000. Characterizing a perspective underlying the practice of mathematics teachers in transition. *Journal for Research in Mathematics Education* 31:579–601.

Wolcott, H. F. 1994. *Transforming qualitative data: Description, analysis, and interpretation.* Thousand Oaks, CA: SAGE.