OPEN EDUCATIONAL RESOURCE

Qualitative Data Coding: Inductive, Deductive and Hybrid Coding

Derek Jansen, MBA
Grad Coach

Reviewed by Eunice Rautenbach, DTech · November 2026

Abstract

Coding is the step in most qualitative analyses that turns a large body of text, such as interview transcripts, field notes or documents, into material that can be compared and interpreted. It is also a step that methods sections often describe vaguely, and in which a stated approach and the actual practice can diverge. This article defines a code and distinguishes codes from categories and themes. It then sets out the three overarching approaches to coding: inductive coding, in which codes are developed from the data; deductive coding, in which codes are taken from existing theory and recorded in a codebook; and hybrid coding, which combines the two. A single interview extract is coded both ways to show what each approach captures and misses, and the approaches are compared side by side. The article describes coding as a process in first and second cycles, from initial codes to categories and themes, and summarizes common coding techniques, grouped by where their codes come from, among them descriptive, in vivo, process, values and structural coding. It closes with practices that keep coding consistent and transparent. It is intended for students and early-career researchers coding qualitative data, and for the educators who teach qualitative research methods.

Keywords: qualitative coding; qualitative data coding; inductive vs deductive coding; inductive coding; deductive coding; types of qualitative coding; examples of codes in qualitative research; open educational resource

Suggested citation: Jansen, D. (2026). *Qualitative data coding: Inductive, deductive and hybrid coding*. Grad Coach. https://doi.org/10.5281/zenodo.24198178

About this article: This is the formal edition of two articles first published by Grad Coach: gradcoach.com/qualitative-data-coding-101/ and gradcoach.com/inductive-deductive-coding/, where the tables can also be downloaded for reuse. Further open educational resources on research methods are listed at gradcoach.com/open-educational-resources/.

© 2026 Grad Coach. Free to share and adapt under CC BY 4.0 with credit to Grad Coach and a link to gradcoach.com/qualitative-data-coding-101/. Quoted passages remain their authors'.

1. Introduction

Qualitative data are bulky. Interview transcripts, field notes and documents quickly amount to many pages, and the findings must be drawn from all of them, not only from the passages a researcher remembers. Coding is the step that makes this possible. Each relevant passage is given a short label, so that passages with similar meanings can be found, compared and grouped, and so that a reader can see how the findings were built from the data (Linneberg & Korsgaard, 2019).

Coding is also a step that is easy to describe loosely. A methods section may state that the codes emerged from the data when they were in fact taken from a framework, or may name a coding technique that the coded transcripts do not show. Elliott (2018) argues that coding is best understood as a series of decisions: where codes come from, how large a passage to code and whether everything should be coded. Each depends on the study's research questions and design. A methods section that records those decisions is easier to defend than one that describes coding in general terms.

This article is written for students and researchers coding qualitative data and for those who teach qualitative methods. Section 2 defines a code and distinguishes codes from categories and themes. Section 3 sets out the three overarching approaches, inductive, deductive and hybrid coding, with a worked comparison. Section 4 describes coding as a process in two cycles. Section 5 summarizes common coding techniques, and Section 6 covers the practices that keep coding consistent and transparent.

2. Codes, Categories and Themes

A **code** is a label that describes what a passage of data is about. Saldaña (2009) gives a widely cited definition: a code is "most often a word or short phrase that symbolically assigns a summative, salient, essence-capturing, and/or evocative attribute for a portion of language-based or visual data" (p. 3). As he notes, the passage coded can be a single word, a sentence, a page of text or a stretch of video (p. 3). **Coding** is the process of developing codes and applying them to the data.

Coding involves judgment. Two researchers reading the same passage with different research questions, or from different theoretical positions, may reasonably label it differently; Saldaña (2009) describes coding as "not a precise science; it's primarily an interpretive act" (p. 4). That is one reason why a study needs to record what each code means and how it was applied (Section 6).

Methodologists also differ on whether coding is itself analysis. Saldaña (2009, p. 7) sets two views side by side. In one, coding is analysis (Miles & Huberman, 1994, as cited in Saldaña, 2009). In the other, "coding and analysis are not synonymous, though coding is a crucial aspect of analysis" (Basit, 2003, p. 145, as cited in Saldaña, 2009, p. 7). Linneberg and Korsgaard (2019) treat coding as the first stage of analysis, but stress that coding and interpretation are interrelated and that deriving an overall interpretation from codes is not mechanical. In practice, coding organizes the data and prepares the analysis; the interpretation that produces the findings is built on that work.

A short example illustrates the idea. Box 1 shows a hypothetical extract from an interview about doctoral supervision, and Table 1 shows how a researcher might code it on a first reading.

© 2026 Grad Coach · CC BY 4.0 · gradcoach.com

Box 1. A hypothetical interview extract on doctoral supervision, written for this article.

> **Interviewer:** How did you find the first few months of your project?
>
> **Participant:** Honestly, I felt quite lost. My supervisor was on sabbatical, so I worked from the proposal document and hoped I was heading in the right direction.
>
> **Interviewer:** And once she was back?
>
> **Participant:** Better, but the feedback arrived in big batches. I'd wait three weeks and then get four pages of comments all at once.

Table 1. Initial codes applied to the extract in Box 1.

Passage	Initial code
I felt quite lost … hoped I was heading in the right direction	Uncertainty
My supervisor was on sabbatical	Supervisor availability
I worked from the proposal document	Working from documents
the feedback arrived in big batches … four pages of comments all at once	Feedback timing

The codes in Table 1 are deliberately broad. They describe the content of each passage and say little yet about what it means; that comes later, as codes are compared and grouped.

Codes are the first of three levels. A **category** groups codes that share a characteristic, and a **theme** is a statement of meaning developed from the categories (Saldaña, 2009, pp. 8–13). In the example, the codes "supervisor availability" and "feedback timing" might be grouped under the category *supervision*. A theme goes further and makes an interpretive claim, such as that students describe supervision as unpredictable rather than absent. Braun and Clarke (2006) treat a theme as a pattern of meaning across the dataset that matters for the research question. A theme need not appear in any fixed proportion of the data. Whether something counts as a theme depends on the researcher's judgment of its importance, not simply on a count of passages (Braun & Clarke, 2006).

Not every author separates categories from themes. Where the two are separated, a category can be treated as describing the manifest content (what the text says on the surface), and a theme as expressing its latent content, the meaning beneath the surface (Graneheim & Lundman, 2004, as cited in Vaismoradi et al., 2013, p. 402). Elliott (2018) notes that the terms are used inconsistently across textbooks, but that there is general agreement on two levels: codes, applied to the data, and categories or themes, which are built from the codes rather than from the data directly (p. 2852). Saldaña (2009) makes the same point about themes: "A theme is an outcome of coding, categorization, and analytic reflection, not something that is, in itself, coded" (p. 13).

© 2026 Grad Coach · CC BY 4.0 · gradcoach.com

3. Where Codes Come From: Inductive, Deductive and Hybrid Coding

The most consequential coding decision is where the codes come from. Codes can be developed from the data, taken from existing theory before the data are read, or both. These three options are known as inductive, deductive and hybrid coding. Elliott (2018) notes that the same distinction appears under other names, such as emergent and a priori (predefined) codes, and that the choice depends on the research question and the epistemology of the design, that is, its assumptions about how knowledge is produced (p. 2855).

3.1 Inductive coding

In **inductive coding**, the researcher begins without a predefined list of codes and develops codes from the data while reading it. The codes stay close to what participants said, often using their own words, rather than the theoretical vocabulary of the researcher (Linneberg & Korsgaard, 2019). New codes are added as new ideas appear, and earlier transcripts are revisited as the list grows. Inductive coding is not, however, free of assumptions: Braun and Clarke (2006) point out that even data-driven coding is shaped by the researcher's theoretical and epistemological commitments.

Inductive coding suits exploratory studies and topics for which no theoretical concepts are readily available (Linneberg & Korsgaard, 2019). A study of how older patients with limited mobility experience virtual reality therapy, for example, might have little existing theory to draw on, so its codes would need to come from the participants' accounts. The approach is closely associated with grounded theory (Linneberg & Korsgaard, 2019), a methodology that aims to generate theory grounded in the data (Chun Tie et al., 2019).

The main risk is the volume of codes. Inductive coding tends to generate many narrow codes, and the process can become complicated and unfocused, especially for a novice (Linneberg & Korsgaard, 2019). Because the code list keeps changing, transcripts coded early may also need recoding once later codes exist (Elliott, 2018, p. 2855).

3.2 Deductive coding

In **deductive coding**, the codes are defined before coding begins, usually from a theory, a framework or earlier research, and are then applied to the data. The predefined list is recorded in a coding frame or **codebook**. This approach focuses the coding on issues already known to matter in the literature and is often used for theory testing or refinement (Linneberg & Korsgaard, 2019). A theory-driven study can convert its theoretical framework directly into a coding framework.

Table 2 shows a short codebook for a hypothetical study of whether self-determination theory explains how postgraduate students stay motivated. The theory proposes three innate psychological needs (competence, autonomy and relatedness), whose satisfaction supports motivation (Ryan & Deci, 2000, p. 68), so the codebook has one code for each need. In the theory, autonomy is the sense that one's actions are self-determined rather than externally controlled, and choice and opportunities for self-direction strengthen it (Ryan & Deci, 2000, pp. 69–70). Table 2 states when each of the three codes applies.

© 2026 Grad Coach · CC BY 4.0 · gradcoach.com

The strength of deductive coding is focus and consistency: every transcript is read against the same short list, so that results can be compared across cases and related directly to the theory. Its weakness is the reverse of that strength. Anything the theory did not anticipate has no code, and in a strictly deductive design it can be overlooked. Creswell notes that a priori codes "serve to limit the analysis to the 'prefigured' codes" and encourages researchers "to be open to additional codes emerging during the analysis" (Creswell, 2013, p. 185, as cited in Elliott, 2018, p. 2855). Linneberg and Korsgaard (2019) likewise note that a coding frame can be adjusted during coding when something new and interesting appears. Gale et al. (2013) consider it generally helpful, even when the approach is largely deductive, to code at least a few transcripts inductively first, without a codebook, so that nothing important in the data is overlooked.

Table 2. An illustrative deductive codebook based on self-determination theory.

Code	Apply when the participant mentions	Example extract (invented)
Autonomy	Acting by their own choice rather than under outside pressure	I could pick which chapter to draft first
Competence	Feeling capable at the task, or not	By week three I finally knew what I was doing
Relatedness	Connection to peers, supervisors or a community	The writing group was the only reason I kept going

3.3 Hybrid coding

Hybrid coding combines the two approaches: coding starts with a set of codes taken from theory, and new codes are added from the data where the starting set does not capture what participants describe. Fereday and Muir-Cochrane (2006) describe such a process in a doctoral study of performance feedback in nursing. A codebook was developed in advance from the research questions and the theoretical framework, and data-driven codes were added during coding, either as new codes or as expansions of codes already in the codebook. Linneberg and Korsgaard (2019) observe that, in practice, combining the two approaches is the most common choice.

A workable hybrid procedure runs as follows. The starting codebook is built from the theory, as for deductive coding. The first few transcripts are coded with it, and anything relevant that has no code is noted on a separate list. Once that list stops growing, recurring items are defined as codes and added to the codebook. The transcripts already coded are then recoded so that every case has been through the same frame. Gale et al. (2013) add a safeguard for the framework method: an "other" code under each category of the framework, so that passages it does not cover are still captured. Roberts et al. (2019) report a similar iterative process. Their preliminary codebook drew on the research question, the literature, an earlier survey and a first scan of the interviews, and it was applied to the data repeatedly and extended until no new codes emerged. Elliott (2018) makes the same point about emergent codes in general. They need to be codified so that the researcher can say that all the data have been considered in light of all the codes, and this means returning to the portions coded early on (p. 2855).

Hybrid coding needs one discipline in particular: **recording where each code came from**. The methods section of a hybrid study should state which codes were defined in advance and which were developed from the data, a distinction that is difficult to reconstruct after coding is com-

© 2026 Grad Coach · CC BY 4.0 · gradcoach.com

plete. A column in the codebook recording each code's source (theory or data) solves the problem with little effort.

3.4 Hybrid and abductive coding

Hybrid coding is sometimes called **abductive** coding (Linneberg & Korsgaard, 2019). The two terms can, however, be distinguished. Hybrid describes where the codes come from: some from theory and some from the data. Abduction describes a way of reasoning, in which the researcher moves back and forth between data and theory, remains open to surprises in the data and takes a particular interest in rethinking existing theory in light of them (Linneberg & Korsgaard, 2019). A study that combines a priori and emergent codes is hybrid in procedure. A study that is described as abductive should also be able to identify the surprising findings it reasoned from and say how they led it to rethink, extend or qualify the theory.

3.5 One extract, coded both ways

The difference between the approaches is clearest when the same passage is coded both ways. Box 2 takes a single hypothetical interview sentence and codes it first inductively and then deductively, using the codebook in Table 2.

Box 2. A hypothetical extract coded inductively and deductively, written for this article.

> **Extract:** I found online classes challenging because I struggled with time management and staying motivated.
>
> **Coded inductively** (no codebook): *time management challenges*; *motivation difficulties*. Both codes name what the participant describes.
>
> **Coded deductively** (Table 2): *Competence* applies, since struggling with time management reads as not feeling capable at the task. *Autonomy* and *Relatedness* do not appear, so neither is coded. Time management and motivation, as topics, have no code of their own in Table 2.

Each approach captures something the other misses. The inductive pass records time management and motivation difficulties. Neither has a code in the self-determination codebook, which codes the three needs rather than motivation itself, so a strictly deductive pass would leave both uncoded. The deductive pass links the sentence to a construct in the theory, and does so in the same way for every transcript, which the inductive pass does not attempt. Neither pass is more rigorous than the other. They answer different questions, which is why the choice between them follows from the research question.

3.6 Choosing an approach

Table 3 compares the three approaches. In general, inductive coding fits exploratory aims and topics with little existing theory, deductive coding fits the testing or application of an established framework, and hybrid coding fits studies whose aims are partly exploratory and partly confirmatory.

© 2026 Grad Coach · CC BY 4.0 · gradcoach.com

Whichever approach is chosen, it should be stated plainly and applied consistently. A study that starts from a framework and then adds codes is hybrid, and is better described that way than as purely inductive or purely deductive. If the approach changes partway through, the transcripts coded under the earlier approach should be recoded (Sandelowski, 1995, as cited in Nowell et al., 2017, p. 6), so that the findings do not reflect the order in which the data happened to be read.

Table 3. Inductive, deductive and hybrid coding compared.

	Inductive	Deductive	Hybrid
Where codes come from	The data	Existing theory or an earlier codebook	A starting codebook, extended from the data
Suits	Exploratory aims; little existing theory	Testing or applying an established framework	Aims that are partly exploratory, partly confirmatory
Main risk	Too many codes; early transcripts need recoding	Missing what the theory did not anticipate	Losing track of which codes came from where

4. The Coding Process: First and Second Cycles

Coding rarely happens in a single pass. Saldaña (2009) divides coding methods into **first-cycle** and **second-cycle** methods, and describes the process as cyclical rather than linear: data are compared with data, codes with codes, categories with categories and categories back with data (p. 45). Linneberg and Korsgaard (2019) describe the same two cycles.

Grounded theory uses its own names for the stages of coding, and its three main versions name them differently: a traditional version associated with Glaser, an evolved version associated with Strauss, Corbin and Clarke, and a constructivist version associated with Charmaz (Chun Tie et al., 2019). The traditional version calls them open, selective and theoretical coding; the evolved version calls them open, axial and selective coding; the constructivist version calls them initial, focused and theoretical coding (Chun Tie et al., 2019). Birks and Mills's generic terms for the three stages are initial, intermediate and advanced coding (Birks & Mills, 2015, as cited in Chun Tie et al., 2019).

4.1 Before coding

Three things should be in place before coding begins: a research question and design, some reading of the relevant literature, and at least some of the data, documented in a form that can be analyzed systematically (Linneberg & Korsgaard, 2019). The research question decides what is relevant. For deductive or hybrid coding, the literature also supplies the starting codebook.

A codebook records each code and what it means. Saldaña (2009) describes it as "a compilation of the codes, their content descriptions, and a brief data example for reference" (p. 21), and notes that it becomes especially important when several people code the same data. For team projects, MacQueen et al. recommend that each codebook entry should contain "the code, a brief definition, a full definition, guidelines for when to use the code, guidelines for when not to use the code, and examples" (MacQueen et al., 2008, p. 121, as cited in Saldaña, 2009, p. 21). An inductive study builds its codebook during coding; a deductive study begins with one.

© 2026 Grad Coach · CC BY 4.0 · gradcoach.com

4.2 First-cycle coding

First-cycle methods are those used in the initial coding of the data, and most of them are fairly simple and direct (Saldaña, 2009, p. 45). The aim is to read the data closely, attach codes to relevant passages, and build an initial code list (inductive) or apply the codebook (deductive). Table 1 is an example of a first-cycle pass. The techniques available for this cycle are summarized in Section 5.

Two decisions shape first-cycle coding. The first is **how large a passage to code**. Saldaña (2009, pp. 19–20) contrasts "lumper" coding, which gives a large excerpt a single code, with the approach of the "splitter," who divides the data into smaller units. Lumping is quicker and still allows detailed subcoding later, whereas splitting produces a more nuanced analysis from the start but can overwhelm the analyst when the codes must be categorized. Grounded theory's line-by-line coding is an example of splitting, used in inductive coding to capture the complexity and diversity of the data (Linneberg & Korsgaard, 2019).

The second decision is **whether to code everything**. Saldaña (2009, p. 15) notes that methodologists disagree: some treat every recorded detail as worth coding, while others, perhaps most, examine only the most salient portions. He advises novices to code everything at first. Elliott (2018) finds that most of the literature treats coding as a way of reducing the data rather than covering all of it (p. 2854). Elliott also notes that even passages left uncoded must be read closely in order to decide that they need no code (p. 2854).

4.3 Second-cycle coding

Second-cycle methods reorganize the first-cycle codes. They call for "such analytic skills as classifying, prioritizing, integrating, synthesizing, abstracting, conceptualizing, and theory building" (Saldaña, 2009, p. 45). Codes are compared, merged, renamed or dropped, and related codes are grouped into categories. One common second-cycle method is **pattern coding**, in which a more inferential "meta-code" brings together several descriptive codes (Punch, 2014, p. 174, as cited in Elliott, 2018, p. 2852).

In the supervision example, a second cycle might group "supervisor availability," "feedback timing" and similar codes from other transcripts under the category *supervision*, and "uncertainty" and "working from documents" under a category such as *working without guidance*. Comparing the categories across participants is what allows a theme, such as the unpredictability of supervision, to be stated and supported with extracts. Saldaña (2009) presents this movement from codes to categories to themes or concepts, and then to theory, as a streamlined model, and cautions that actually reaching theory is more complex and messy than the model suggests (p. 11). Not every study aims for theory; many end with a set of themes.

Throughout both cycles, researchers write **analytic memos**: notes on the codes, the participants, the phenomenon and the connections between them. Memos record ideas while they are fresh and help to bridge coding, analysis and the written findings (Linneberg & Korsgaard, 2019). A theme noticed during first-cycle coding belongs in a memo rather than in the code list, because themes are developed from codes rather than coded directly (Saldaña, 2009, p. 13).

© 2026 Grad Coach · CC BY 4.0 · gradcoach.com

5. Coding Techniques

Within each cycle, researchers can choose among many named coding techniques; Saldaña (2009, p. 46) profiles 23 first-cycle and six second-cycle methods. Each technique labels a different aspect of the data, such as its topic, the participants' own words, actions or values. Table 4 summarizes ten techniques that students commonly meet and groups them by where their codes usually come from. The grouping is this article's own. Two of its groups draw on Saldaña (2009, p. 49). He notes that some methodologists advise fixing a provisional list of codes in advance so that the analysis answers the research questions, and he points to structural, provisional and hypothesis coding as methods of that kind. He also links in vivo, process and initial coding with grounded theory and the development of new theory. Descriptive coding is placed in the middle group because Linneberg and Korsgaard (2019) note that it can be inductive or deductive. This article chose the groups for values, emotion and attribute coding; Saldaña (2009) files attribute coding among grammatical methods (p. 46) and does not list it among the methods whose codes are determined beforehand (p. 49).

In vivo coding uses the participant's own words as the code, rather than words chosen by the researcher (Saldaña, 2009, p. 48). In vivo codes are often verbatim quotations used to stand for a broader concept or process in the data (Chun Tie et al., 2019). Elliott (2018) summarizes the debate around them: some authors regard them as the best code labels because they keep the analysis close to participants' voices, while others caution against confusing the analytic phase with the presentation of findings (p. 2856). An in vivo code is still an interpretive choice, since the researcher decides which phrase stands for the passage.

Process coding captures action. Process codes use gerunds ("-ing" words) "exclusively to connote observable and conceptual action in the data" (Miles et al., 2014, p. 75, as cited in Elliott, 2018, p. 2855); *negotiating*, *avoiding* and *coping* would be typical labels. Grounded theory places particular weight on coding for action: Charmaz (2006, as cited in Chun Tie et al., 2019) advocates embedding actions in codes, and grounded theory codes identify social and psychological processes rather than themes (Chun Tie et al., 2019).

Initial coding is the constructivist name for the first stage of grounded theory coding, which the traditional and evolved versions call open coding (Chun Tie et al., 2019). At this stage the researcher works inductively, generating as many codes from the early data as possible and starting to compare them (Chun Tie et al., 2019). Outside grounded theory, "initial coding" and "open coding" are also used loosely for any first, exploratory pass through the data.

Descriptive coding labels a passage by what it is about, and leads to an inventory of the topics in the data (Linneberg & Korsgaard, 2019). A descriptive code is a basic label, often a noun (Miles et al., 2014, as cited in Elliott, 2018, p. 2855). The codes in Table 1 are descriptive. Saldaña (2009, p. 48) recommends descriptive coding for field notes, documents and artifacts, and it can be applied inductively or deductively, depending on whether the list of topics was agreed in advance (Linneberg & Korsgaard, 2019).

Values coding and **emotion coding** label subjective aspects of an account. Values coding labels participants' subjective perspectives (Saldaña, 2009, p. 7). Emotion coding labels emotions, such as emotional responses to particular events (Linneberg & Korsgaard, 2019). Both can draw their

© 2026 Grad Coach · CC BY 4.0 · gradcoach.com

codes from the data or from a framework of values or emotions chosen in advance.

Table 4. Ten coding techniques, grouped by where their codes usually come from.

Technique	What the code is
Usually developed from the data	
In vivo coding	The participant's own words, used as the code
Process coding	An action, named with a gerund (an "-ing" word)
Initial (open) coding	A provisional, open-ended label, refined in later passes
Either way	
Descriptive coding	The topic of a passage, in a word or short phrase
Values coding	A participant's values or subjective perspective
Emotion coding	An emotion, such as a response to an event
Usually defined in advance	
Structural coding	The research or interview question a passage relates to
Provisional coding	A code from a start list built from earlier research
Hypothesis coding	A code written to test a stated hypothesis
Attribute coding	Basic information about a case, such as role, site or age

Structural coding labels each passage by the research or interview question it relates to, so that all the material relevant to one question can be retrieved together rather than read chronologically. Richards and Richards suggest the term *indexing* for this kind of question-by-question tagging, to distinguish it from more analytical coding (Richards & Richards, 1994, as cited in Elliott, 2018, p. 2851). **Provisional coding** begins with a start list of codes prepared before coding, and **hypothesis coding** begins with codes written to test a stated hypothesis; Saldaña (2009, p. 49) groups both with structural coding among methods whose codes are determined beforehand. **Attribute coding** records basic information about each case, such as a participant's age or experience, or an organization's industry or size (Linneberg & Korsgaard, 2019). In comparative studies, Linneberg and Korsgaard note, it fixes an early structure for the dataset and makes differences between cases easier to explore.

Two cautions apply to Table 4. First, the grouping follows where the codes come from, not the technique's name, so several techniques can be used inductively or deductively. Second, studies rarely use a single technique. Indexing interviews with structural codes and then coding each segment for processes is a common combination. Saldaña (2009, p. 48) suggests a starting sequence for a first cycle: attribute coding; then structural or holistic coding (a single code for a large excerpt, as in the "lumper" coding of Section 4.2); then descriptive coding; then in vivo, initial or values coding. Pattern or focused coding follows in the second cycle. He also warns against using too many methods in one study or combining incompatible ones (p. 47). A methods section should name the techniques used, in the order used, and show a coded example of each.

© 2026 Grad Coach · CC BY 4.0 · gradcoach.com

6. Consistency, Transparency and Software

Coding is useful only to the extent that it is consistent. A code applied one way to the first transcript and another way to the twentieth will retrieve passages that do not belong together. Clear definitions are the first safeguard: Miles et al. describe “clear operational definitions” as “indispensable” (Miles et al., 2014, p. 84, as cited in Elliott, 2018, p. 2859). Each code should have one name and one meaning, so near-synonyms for the same idea should be merged, and the definitions should be reread and revised as coding proceeds.

Elliott (2018) distinguishes two kinds of consistency. Consistency over time is checked by recoding a clean copy of a document coded earlier and comparing the two. Consistency between researchers, often called intercoder reliability, is checked by having two coders code the same document independently (p. 2858). Agreement can be calculated statistically, usually on a subset of the data. O’Connor and Joffe (2020) observe that there is little agreement on how large that subset should be, but that 10%–25% of data units is typical, depending on the size of the dataset (p. 5). A data unit is a segment of the data to which codes are applied, such as a line, a paragraph, a response to a question or a whole interview (p. 6). They also report that methodologists widely reject simple percentage agreement, because part of any agreement between two coders arises by chance. Measures that correct for chance, such as Cohen’s kappa and Krippendorff’s alpha, are used instead (p. 8).

A qualitative comparison of which codes were least agreed on, and why, may nonetheless be more useful for sharpening definitions (Elliott, 2018, p. 2858). O’Connor and Joffe (2020) note that what coders disagree about can be at least as valuable as the final level of agreement (p. 4). In a team project, every coder needs to work from an agreed codebook and be trained in its use. Roberts et al. (2019) found that building the codebook and training coders took a great deal of time, and that even then full intercoder reliability was unlikely to be achieved.

Not every qualitative tradition treats agreement between coders as a goal. Some researchers value two coders precisely because they read the data differently (Linneberg & Korsgaard, 2019; Richards, 2015, as cited in Elliott, 2018, p. 2859). In grounded theory, where analysis repeatedly returns to earlier stages and builds up gradually, some researchers see a reliability check between coders as out of place (Oktay, 2012, as cited in O’Connor & Joffe, 2020, p. 2).

Transparency means that a reader can follow how the findings were developed from the data. Coding supports this by leaving a trail from data to codes, categories and themes (Linneberg & Korsgaard, 2019). Elliott (2018) endorses Richards’s advice to record every change to the codes and the coding method (Richards, 2015, p. 121, as cited in Elliott, 2018, p. 2859). Nowell et al. (2017) describe the practice in a team thematic analysis: changes in how the team analyzed the data were recorded in the codebook, and minutes of team meetings were kept, as part of an audit trail (p. 8). An audit trail lets readers see what the researchers decided at each stage of the study, and why (Koch, 1994, as cited in Nowell et al., 2017, p. 3). A coding log that records when codes were added, merged or redefined, and why, makes the methods section much easier to write.

Coding can be done on paper, in a word processor or spreadsheet, or with qualitative data analysis software such as NVivo, ATLAS.ti or MAXQDA. Software makes it easier to retrieve coded passages, organize codes in layers and count them (Elliott, 2018, pp. 2857–2858). Saldaña (2009) re-

© 2026 Grad Coach · CC BY 4.0 · gradcoach.com

gards it as indispensable for studies with many interviews or extensive field notes (p. 22). For a first or small-scale study, however, he recommends coding on paper printouts before moving to software (p. 22). Software has drawbacks as well: the ease of creating codes can lead to too many of them, and coding can become mechanical and lose analytical depth (Linneberg & Korsgaard, 2019). The software does not code the data; the researcher does (Saldaña, 2009, p. 22).

Finally, coding has its critics. Breaking data into coded segments can lose the context and the holistic understanding of each account (Linneberg & Korsgaard, 2019). A count of coded instances, Linneberg and Korsgaard add, is no substitute for detailed description, specific examples and attention to contradictory evidence. Rereading full transcripts alongside the coded segments, and looking for evidence that runs against the emerging categories, helps to keep the coding tied to the data it summarizes.

7. Conclusion

Qualitative coding labels passages of data so that they can be compared, grouped and interpreted. Codes describe passages; categories group related codes; themes are interpretive statements developed from the categories, not codes applied to the data. The most important coding decision is where the codes come from: from the data (inductive coding), from existing theory recorded in a codebook (deductive coding), or from both (hybrid coding). Each captures something the others miss, and the choice should follow from the research question. Coding proceeds in cycles. A first cycle labels the data, using techniques such as descriptive, in vivo, process, values or structural coding. A second cycle groups the codes into categories and themes. Clear code definitions, consistency checks, analytic memos and a record of changes make the process transparent. A methods section that states the approach, names the techniques, shows a coded example and records where each code came from allows a reader to see how the findings were built from the data.

© 2026 Grad Coach · CC BY 4.0 · gradcoach.com

References

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Chun Tie, Y., Birks, M., & Francis, K. (2019). Grounded theory research: A design framework for novice researchers. *SAGE Open Medicine*, 7, Article 2050312118822927. https://doi.org/10.1177/2050312118822927

Elliott, V. (2018). Thinking about the coding process in qualitative data analysis. *The Qualitative Report*, 23(11), 2850–2861. https://doi.org/10.46743/2160-3715/2018.3560

Fereday, J., & Muir-Cochrane, E. (2006). Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International Journal of Qualitative Methods*, 5(1), 80–92. https://doi.org/10.1177/160940690600500107

Gale, N. K., Heath, G., Cameron, E., Rashid, S., & Redwood, S. (2013). Using the framework method for the analysis of qualitative data in multi-disciplinary health research. *BMC Medical Research Methodology*, 13, Article 117. https://doi.org/10.1186/1471-2288-13-117

Linneberg, M. S., & Korsgaard, S. (2019). Coding qualitative data: A synthesis guiding the novice. *Qualitative Research Journal*, 19(3), 259–270. https://doi.org/10.1108/QRJ-12-2018-0012

Nowell, L. S., Norris, J. M., White, D. E., & Moules, N. J. (2017). Thematic analysis: Striving to meet the trustworthiness criteria. *International Journal of Qualitative Methods*, 16(1), 1–13. https://doi.org/10.1177/1609406917733847

O'Connor, C., & Joffe, H. (2020). Intercoder reliability in qualitative research: Debates and practical guidelines. *International Journal of Qualitative Methods*, 19, 1–13. https://doi.org/10.1177/1609406919899220

Roberts, K., Dowell, A., & Nie, J.-B. (2019). Attempting rigour and replicability in thematic analysis of qualitative research data; a case study of codebook development. *BMC Medical Research Methodology*, 19, Article 66. https://doi.org/10.1186/s12874-019-0707-y

Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1), 68–78. https://doi.org/10.1037/0003-066X.55.1.68

Saldaña, J. (2009). *The coding manual for qualitative researchers*. SAGE.

Vaismoradi, M., Turunen, H., & Bondas, T. (2013). Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study. *Nursing & Health Sciences*, 15(3), 398–405. https://doi.org/10.1111/nhs.12048

© 2026 Grad Coach · CC BY 4.0 · gradcoach.com