Stefan Rädiker

Doing Grounded Theory with MAXQDA

Guidance and Tips for Your Practice

Guide

MAXQDA PRESS

Recommended citation:

Rädiker, Stefan (2023). Doing grounded theory with MAXQDA. Guidance und tips for your practice. Berlin: MAXQDA Press. https://doi.org/10.36192/978-3-948768164

ISBN 978-3-948768-16-4
https://doi.org/10.36192/978-3-948768164

MAXQDA Press is part of VERBI Software. Consult. Social Research. GmbH
Invalidenstr. 74, 10557 Berlin, info@maxqda-press.com
www.maxqda-press.com

Contents

1 What to expect from this guide?

Many research projects apply the approach of Grounded Theory, which was introduced by Barney Glaser and Anselm Strauss in their book *The Discovery of Grounded Theory* (1967). Grounded theory projects aim to develop middle-range theories on the basis of empirical data. Concepts and categories are developed from the data, aspects of which are not only described but related to one another, and often but not necessarily condensed to a central, explanatory key category (Corbin & Strauss, 2015, p. 13).

The MAXQDA software supports grounded theory studies through numerous functions and options, which are described in this guide:

- You can save the entire data material in a single MAXQDA project file.
- You can select and code any part of the data. Special coding techniques such as open coding and the creation of in vivo codes are supported.
- Your coding work is visualized directly next to and in the data material and you can change and comment on it in place.
- You can organize, rename, merge, and differentiate codes and categories using a hierarchical list and using a whiteboard-like environment.
- You can compile selections of the coded pieces of the data according to numerous criteria.
- You can write, modify, search, and merge Memos with analytical notes. Different memo types can be distinguished.
- You can link data pieces to each other and to memos.
- You can search for interesting words across all cases at any time.
- You can create interactive diagrams.

Since Glaser and Strauss laid the foundation for a long methodological tradition more than 50 years ago, the approach has been continuously developed. Both Glaser (1978) and Strauss (1987) have independently provided their own descriptions with different emphases. Kathy Charmaz (2006) has developed a constructivist-oriented version, and Adele Clarke (2005) a postmodernist variant under the name “situational analysis”. For an overview of the history and different streams of the grounded theory approach see, for example, Flick (2018b) and Birk and Mills (2015).

Rather than focusing on the differences between the different approaches, this guide focuses on the commonalities and introduces helpful features of MAXQDA for implementing grounded theory studies.

The focus is on practical implementation, while also addressing pitfalls I have encountered while consulting on research projects.

Even though this guide addresses important basic principles of the grounded theory approach, it cannot replace relevant methodological literature. Quite the contrary: from my experience, for the meaningful use of analysis software such as MAXQDA, it is important to have a basic knowledge of research methods and to know the procedures and principles of the chosen approach and make them your maxim. MAXQDA is a computer program that can assist you very effectively in conducting grounded theory studies. However, you will not find a "Grounded Theory" button for step-by-step instructions (nor will you find one for any other method), nor will MAXQDA take decisions for you in the research process. Rather, MAXQDA provides you with an environment for working systematically with your data, memos, codings, diagrams, and linkages.

To follow the guide, it helps if you are familiar with the basic functions of MAXQDA. It is sufficient to watch a 10-minute introductory video on YouTube, join a free webinar at www.maxqda.com, or read the getting started guide, also available at the MAXQDA website.

Grounded theory projects are diverse and there are many ways to implement them with MAXQDA. Therefore, I welcome your feedback and suggestions for improvement of this guide by sending me an email to raediker@methoden-expertise.de. I am also happy to receive references to grounded theory studies that were conducted with the help of MAXQDA.

All the best for your grounded theory studies with MAXQDA.
Stefan Rädiker

2 How does a grounded theory study progress?

While it is not possible to formulate a small-scale step-by-step guide for grounded theory studies because the approach is characterized by a flexible and circular procedure, the flow process can be illustrated as in the following figure:

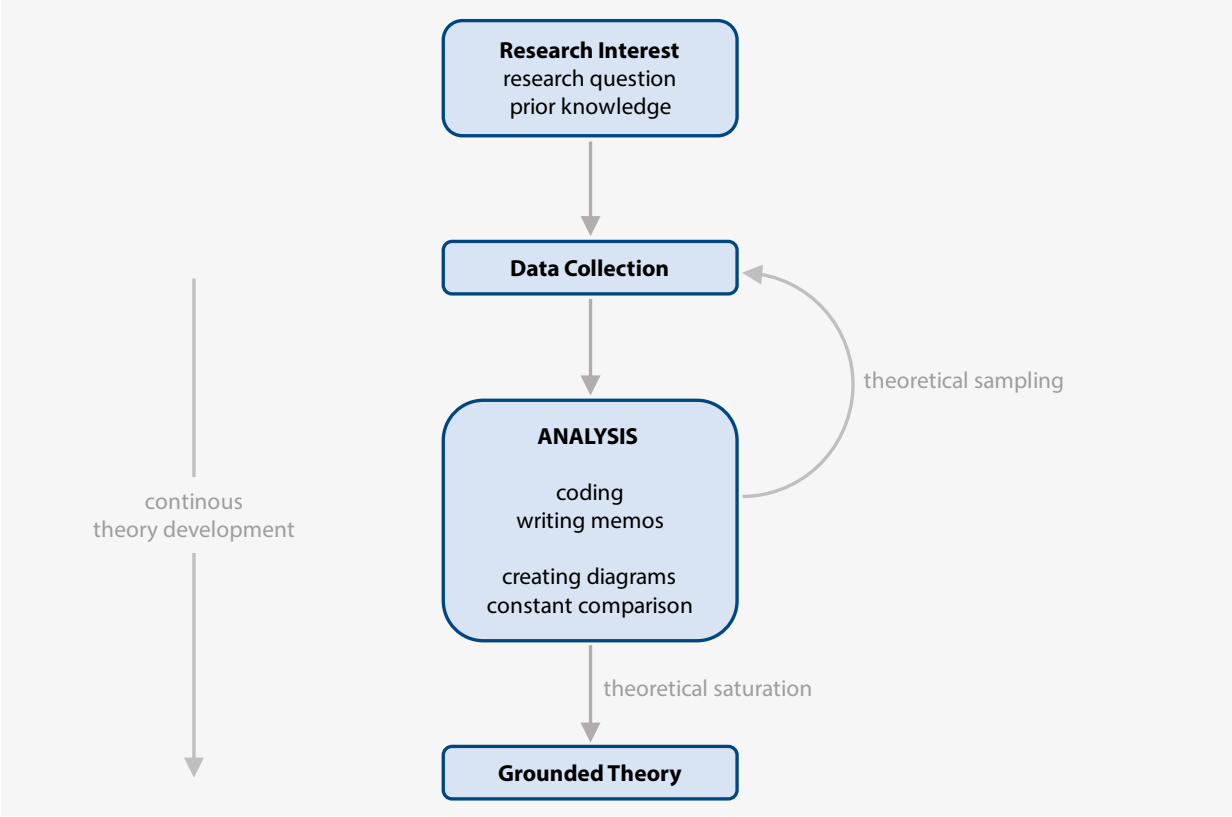


Fig. 1: Typical process of a grounded theory study

Researchers conducting a grounded theory study follow some basic principles (regardless of particular approaches) and apply recurring techniques and procedures, which can be outlined in terms of the process in Fig. 1 as follows (see, e.g., Bryant & Charmaz, 2007, p. 12):

The research process is iterative, with alternating data collection and analysis.

At the beginning there is a research interest, which is expressed in openly formulated research questions. Prior knowledge is made explicit and reflected upon to enter the field as openly and consciously as possible. The first data collection often takes place with the help of a purposive sampling of interview partners, texts, observation situations, etc. The analysis phase starts at the same time rather than, as with other methods, only after the data collection. The insights, gained through the progressive analysis of the data, guide further data collection, which is followed by further analysis and, if necessary, further cycles of data collection and analysis.

A central analysis technique is the coding of data. The resulting codes and categories become increasingly abstract and conceptual.

Immediately after their collection, the data and field notes are coded, and here different coding techniques can be distinguished. The first coding phase serves to "break up the data" (Strauss & Corbin, 1990) and works very closely and often line-by-line with the data; subsequent coding remains close to the data material but works with more abstract concepts and categories and often involves longer passages. The developed categories form the basic framework for the developing theory, and they are elaborated on and related to each other as the analysis progresses.

Writing, sorting, and integrating memos is part of the overall analysis process. Memos also tend to become more abstract and conceptual over time.

From the start of each new project, researchers write memos in which they record their thoughts and findings. Memos can refer to the research process, to data collection, to identified concepts and categories, and many more aspects of the analysis. In analytical memos, as in coding, the level of abstraction increases as the analysis progresses. Particularly, analytical memos provide a good basis for formulating the theory in the research report.

Another analytical strategy is constant comparison.

Constant comparison helps the researcher to move away from pure description and think analytically and conceptually about the data in order to "find consistencies and differences, with the aim of continually refining concepts and theoretically relevant categories" (Tie et al., 2019, p. 4). Comparison can take place

at different levels and be applied to different elements: phenomena described in the data, initial codes, concepts and categories developed from them, and also whole cases, people, and institutions.

The collection of further data follows principle of theoretical sampling and ends after theoretical saturation has been reached.

The insights gained by analyzing the data already collected are incorporated into the next data collection cycle. Thus, the collection of further data is not only guided by general preliminary considerations, but explicitly serves to test, refine and improve the current state of the categories through further, possibly challenging cases. Theoretical saturation occurs when further data no longer contribute to further development of the categories, their relationships, or the emerging theory.

Theory development is a continuous process which is supported by the preparation of concept maps.

The development of the theory does not take place at the end of the analytical process but is attempted from the very first minute. The degree of abstraction of the analysis increases continuously, that is, the memos, the categories and their relations become increasingly abstract and complex. One possibility is to single out a central category that has great explanatory power. The process of continuous theory development can be aided by the creation of concept maps showing how the discovered concepts relate to each other. Concept maps can also increase in complexity as the analysis progresses. Finally, the developed grounded theory can be formulated and presented in different ways, for example, in the form of statements, which are supported by a concept map.

The following chapters show how the process and principles described above can be implemented with MAXQDA.

3 Create project, import and organize data

Create MAXQDA project

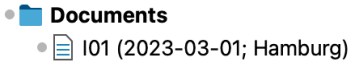
It is a good idea to set up a MAXQDA project even before the first data has been collected to be able to record all the important information about the planned study as early as possible and to establish MAXQDA as a central workspace during the implementation. For example, the logbook (available in the ***Home*** menu) can be used to record important decisions in the research process. The project memo (***Memos > Project Memo)*** appears at the top of the "Document System" and can be titled "Data collection and theoretical sampling" to record key steps in the data gathering process (please see Chapter 5 for more ways to use memos).

Import data

As soon as the first data are available, they are immediately imported into the MAXQDA project. The data are often interviews for which the audio recordings still need to be transcribed. This can be done directly in MAXQDA or, for example, with the help of services for automatic transcription. Typical for grounded theory projects is the inclusion of different types of data, which means that field notes and documents, reports and gray literature, websites and blog articles, videos and music, and even the results of a quantitative survey, can be added as data sources.

In principle, all data for a grounded theory study should be imported into a single MAXQDA project so that they can be viewed, searched, and used at any time and you do not have to switch between different software programs. You should import field notes as documents (rather than document memos) so that you can code them and add memos to them later. If necessary, you can also create field notes directly in the MAXQDA project by clicking on the ***New text document*** icon in the "Document System".

No matter whether it is an interview, field note or audio file: all the data imported into a MAXQDA project are called "documents" here and are available in the "Document System". To ensure that the documents can be easily recognized and distinguished, the document names should follow a uniform scheme and be as informative as possible. Depending on requirements, they can provide information about the respective case, document type, and collection time and place, such as these:



"I" stands for interview, "01" for the first case and the time and place of the data gathering are noted in brackets. Instead of the case numbers, pseudonyms can also be chosen and instead of the exact date, one can also note the number of the cycle. In any case, the document name of each type of data should begin with a different letter so that they can be easily distinguished and searched.

The additional information for each document can also be stored in document variables to allow easy access and grouping the documents later and to be able to output the information in different places:

1. In the ***Variables > List of Document Variables*** window, click on the icon to create new variables with type "Text", for example, "Document Type", "Location", "Date". For numerical information, the variable types "Integer" and "Decimal" are available.
2. In the ***Variables > Data Editor for Document Variables*** window, the data for each imported document is entered.

Organize data into document groups

In order to organize the data in the project systematically, folders should be created in the "Document System", which are called document groups in MAXQDA.

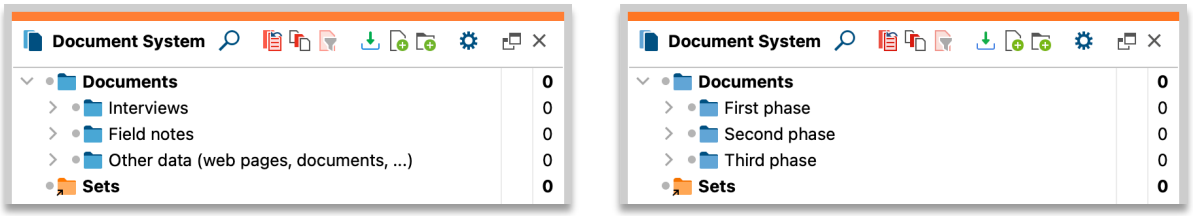


Fig. 2: Document groups in the "Document System" (left: sorted by document types, right: sorted by time)

The document groups can be created according to different aspects:

- By document types
- By cases (where several documents relate to each case)
- By date of data collection
- By place of data collection
- According to content-related criteria

Since MAXQDA supports up to two levels for document groups, the structure can also combine several of these characteristics, for example by differentiating within document types by date of collection. If, in the later course of analysis, the original structure needs to be adapted to newly emerging distinguishing characteristics, the document groups can be rearranged and renamed, and documents can be moved back and forth between individual document groups with the mouse.

Key points summarized:

- Create a MAXQDA project as early as possible.
- Always import all relevant data into the project.
- Use informative document names.
- Record additional information about each document (e.g., location and time) in document variables.
- Import field notes as documents, not memos.
- Organize the documents using document groups.

4 Coding data

General aspects of coding in grounded theory studies

Coding the data is of particular importance in grounded theory studies because coding "is the central process by which theories are built from data" (Strauss & Corbin, 1990, p. 57). The results of a grounded theory study go beyond a descriptive description of phenomena; instead, the goal is to draw out relationships, explanations, conditions, causalities, etc. To move away from pure descriptiveness, the adage "think conceptually" is important throughout the analysis process even during coding (Juliet Corbin in conversation with Cisneros-Puebla, 2004). Accordingly, there are numerous suggestions in the literature that support conceptual thinking and coding toward a theory:

- When coding, *questions are applied to the text*, both to individual passages and to an entire case: What is happening here? Who is involved? How does it happen? When and for how long? Where? How much and how strongly? Why? What for? By what? (Corbin & Strauss, 2015; Flick, 2018a).
- The *constant comparison* method is a central principle of analysis that is also used in coding: "while coding an incident for a category, compare it with the previous incidents in the same and different groups coded in the same category" (Glaser & Strauss, 1967, p. 106). Each new code assignment is compared with the existing ones to identify differences and similarities as well as to elaborate the theoretical properties of a category (ibid.).
- *Writing memos* helps the researcher to think about coding, to theorize, and to put this in writing. Memos are discussed in detail in Chapter 5.
- Great attention should be paid to *the naming of codes and categories.* For example, it is recommended by Charmaz (2014), referring to Glaser (1978), to formulate codes in the gerund in order to address processes and/or actions, which are often the focus of grounded theory studies. This can be implemented very well in English with the common "...ing" form, for example, codes such as "coaching", "supporting", "challenging", "stopping", "encouraging" can be developed when analyzing caregivers' actions.

It is important to note that different grounded theory approaches postulate and emphasize different aspects and ways of coding. Tab. 1 shows which terms the authors of different textbooks have chosen for different phases of coding (for a very instructive overview of the differences in coding, see Flick, 2018b).

Tab. 1: Terms and sequences of coding strategies in different grounded theory approaches

Glaser & Strauss (1967)	1) Comparing incidents applicable to each category 2) Integrating categories and their properties 3) Delimiting the theory
Glaser (1978)	1) Substantive coding (open coding followed by selective coding) 2) Theoretical coding (for example, using the coding families)
Strauss & Corbin (1990, 1998[2])	1) Open coding 2) Axial coding (for example, using the coding paradigm) 3) Selective coding
Charmaz (2006, 2014[2])	1) Initial coding 2) Focused coding 3) Theoretical coding

Despite the differences in the terms used and in the actual coding process, commonalities can be identified: coding usually starts very close to the material, often line-by-line, and becomes progressively more abstract and more large-scale. There are steps of integration, in which the previously formed codes are summarized under more abstract headings, and steps in which the relationships of concepts are elaborated. As the analysis progresses, coding focuses on selected concepts that have high relevance to many cases and to the structure of the categories that have been created; these are aspects of high importance to the developing theory. The different coding phases are not strictly separated from each other, but flow into each other and there is also a back and forth between the phases.

There is no consistent use of the terms “code”, “category”, and “concept” both across textbooks and within individual textbooks. Among other things, this is intended to express different degrees of abstraction, which is why I use the terms predominantly with the following meaning in this guide:

- *Codes* are formulated close to the data and are created at the beginning of the coding process.
- *Categories* are more abstract, they are usually developed from the codes and can summarize codes, but they can also summarize categories from a lower level of abstraction. They are the building blocks of the developing theory.
- *Concept* is a transversal term, in that codes can be understood as lower-level concepts and categories as higher-level concepts.

MAXQDA does not distinguish between codes, categories, or concepts; it is simply called “Code” on the interface and in all functions. Even the window is called “Code System” in MAXQDA, the categories are also organized and managed there.

The following sections present numerous MAXQDA functions that support the coding of data and the development of conceptual categories. The presentation begins with functions that are particularly suitable for the early stages of coding, followed by functions that support increasing levels of abstraction and the systematization of initial codes into elaborated categories.

The MAXQDA mode “Open Coding”

Coding often begins with the development of fine-grained codes on the data, regardless of whether this is referred to as “open coding” (Glaser, 1978; Strauss & Corbin, 1990, 1998) or “initial coding” (Charmaz, 2006, 2014). MAXQDA’s “Open Coding” mode, which is activated at the top of the “Document Browser” by clicking on the icon  , is ideal for this purpose. In this mode, as soon as you highlight a text passage, a window for creating a new code appears (Fig. 3). Here, you can enter a name and select a color for the code, such as red for conceptually promising codes so that they can be easily identified later. By clicking ***OK,*** the new code is inserted at the very top of the “Code System” and is also assigned to the selected text passage.

In addition to the code name, a code memo can also be entered in the input dialog, in which you can record your thoughts and notes about the respective code, for example connections that you would like to check later using further material. At the very bottom of the input dialog, you can write a comment on the code assignment. While the code memo refers to the code in general, the comment is limited to the code assignment to this specific coded segment.

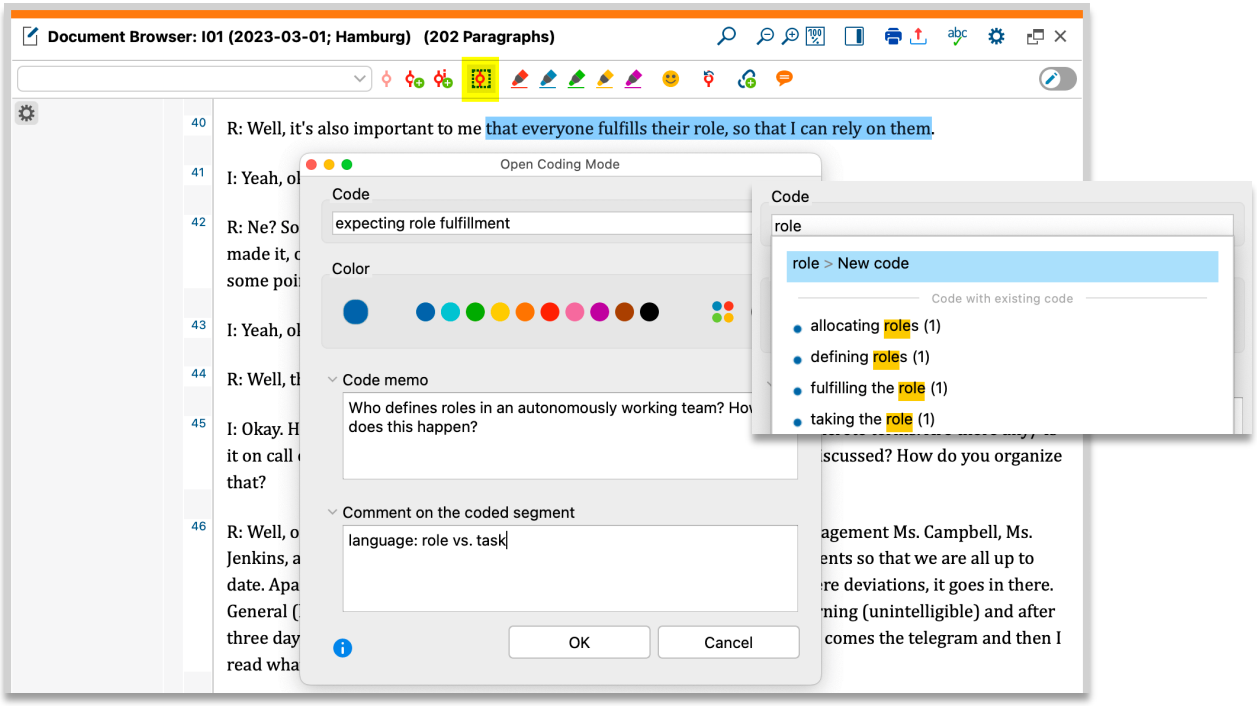


Fig. 3: Switch on "Open coding" mode (highlighted in yellow in the toolbar). List of already existing codes (right)

When entering a code name in the input dialog, MAXQDA lists all existing codes in which the name occurs (Fig. 3). As an alternative to a new code, one of these already existing codes can be assigned. This avoids an inflation of identical or very similarly formulated codes.

Tip: The input dialog from Fig. 3 can also be called up without switching on the "Open Coding" mode. To do this, select a text section and click on the ***Code with a new code*** icon at the top of the "Document Browser". Alternatively, you can call the function via right-click on the selection or by using the keyboard shortcut ***Alt+W*** (Windows) or ***cmd+option+W*** (Mac).

In vivo coding

In vivo codes are codes whose names are taken directly from the data material, which has the advantage that the initial concepts are close to it. To create an in vivo code, a usually quite short and concise text passage is marked and then the icon with the small "I" at the top of the "Document Browser" is clicked. Alternatively, you can use the keyboard shortcuts ***Alt+I*** (Windows) and ***cmd+option+I*** (Mac) or the context menu.

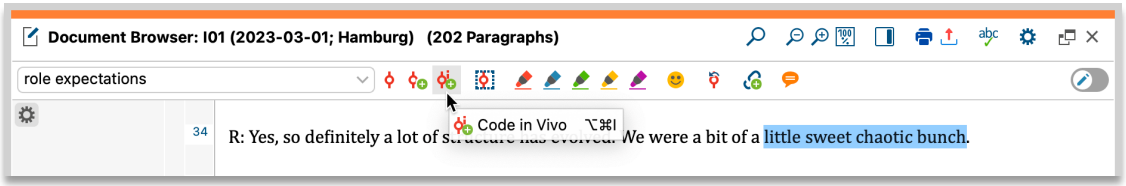


Fig. 4: Creating an in vivo code

New in vivo codes are inserted at the top of the "Code System" and can be moved to another position in the code system if necessary. Of course, other text passages can also be assigned to this code later, which becomes necessary when the initial in vivo code has established itself as a category.

You can mark in vivo codes as such for easy recognition, for example, by a specific color, by adding quotation marks to the code name, or by a specific code memo symbol. However, the usefulness of labeling is questionable, since it is ultimately the meaning of the content that counts. A simple but very effective form for "labeling" in vivo codes is described by Jacques (2021) and shown in Fig. 5: all in vivo codes were initially collected under a separate top code.

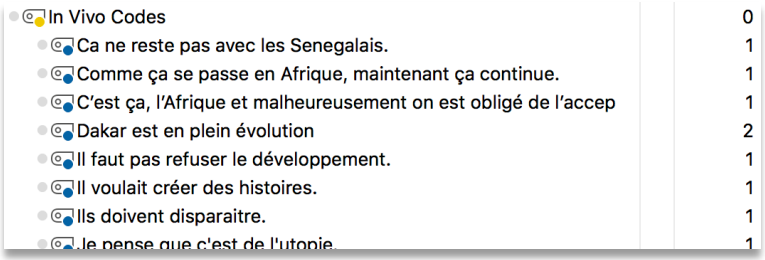


Fig. 5: In vivo codes sorted under a parent code for organizational purposes (source: Jacques, 2021, p. 26)

Color coding

With the help of color coding, individual text passages can be colored as if they were marked with a highlighter. Five different colors are available at the top of the “Document Browser”. As soon as a color code has been used for the first time, it is inserted as a code at the top of the “Code System”. The color codes can be used like other codes, you can rename and move them around in the hierarchy as desired.

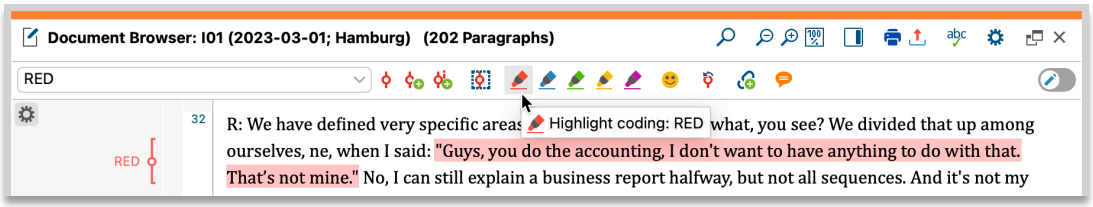


Fig. 6: A color coded text passage

The text passages are permanently highlighted with the color in the “Document Browser”. The markers can only be hidden by turning off the coding stripes with this color in the “Document Browser” (by clicking on the gear icon in the upper left corner of the codes margin). Thus, you should think carefully about what you code with it. Here are a few possibilities of information that could be color coded in a grounded theory study:

- Actors being talked about.
- Passages of interest noticed in a first read-through or that should be subjected to a more in-depth analysis.
- Passages that contradict the current state of the developed theory.

Creative Coding: Organize codes visually as on a whiteboard

“Creative Coding” is not another way of coding in grounded theory approaches, but the name of a function in MAXQDA that can be used to visually arrange and systematize codes, like moving post-its around on a whiteboard. After calling up the function via ***Codes > Creative Coding***, a new window opens in which all codes from the “Code System” are displayed on the left. All codes that you want to sort and group can be dragged onto the workspace area and roughly pre-sorted in this process (Fig. 7). The systematization process is started by clicking the ***Start Organizing Codes*** icon (at the very top left of the window).

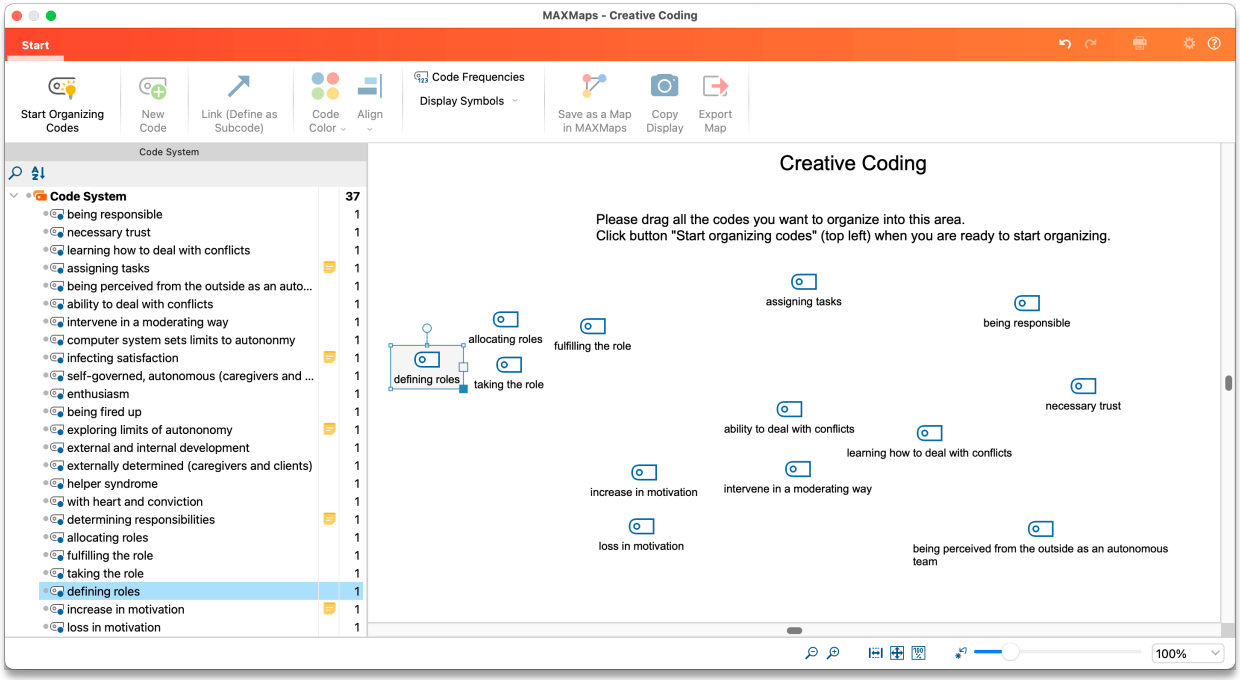


Fig. 7: Select codes for Creative Coding

The following options are available for organizing and grouping:

- Codes can be moved anywhere on the whiteboard with the mouse.
- Two codes can be merged into one code by dragging one onto the other.
- Codes with a higher level of abstraction (or categories in grounded theory terminology) can be inserted by clicking the ***New Code*** icon in the menu.
- Codes can be defined as a subcodes of another code (or category) by turning on the ***Link (Define as Subcode)*** mode in the menu. By clicking-and-dragging with the mouse, an arrow can then be created between two codes, indicating their relationship.
- One or more codes can be selected to assign a new color to them.

The changes to the code system, that is, the groupings and mergers of codes as well as newly created codes, are transferred to the “Code System” window when the Creative Coding window is closed.

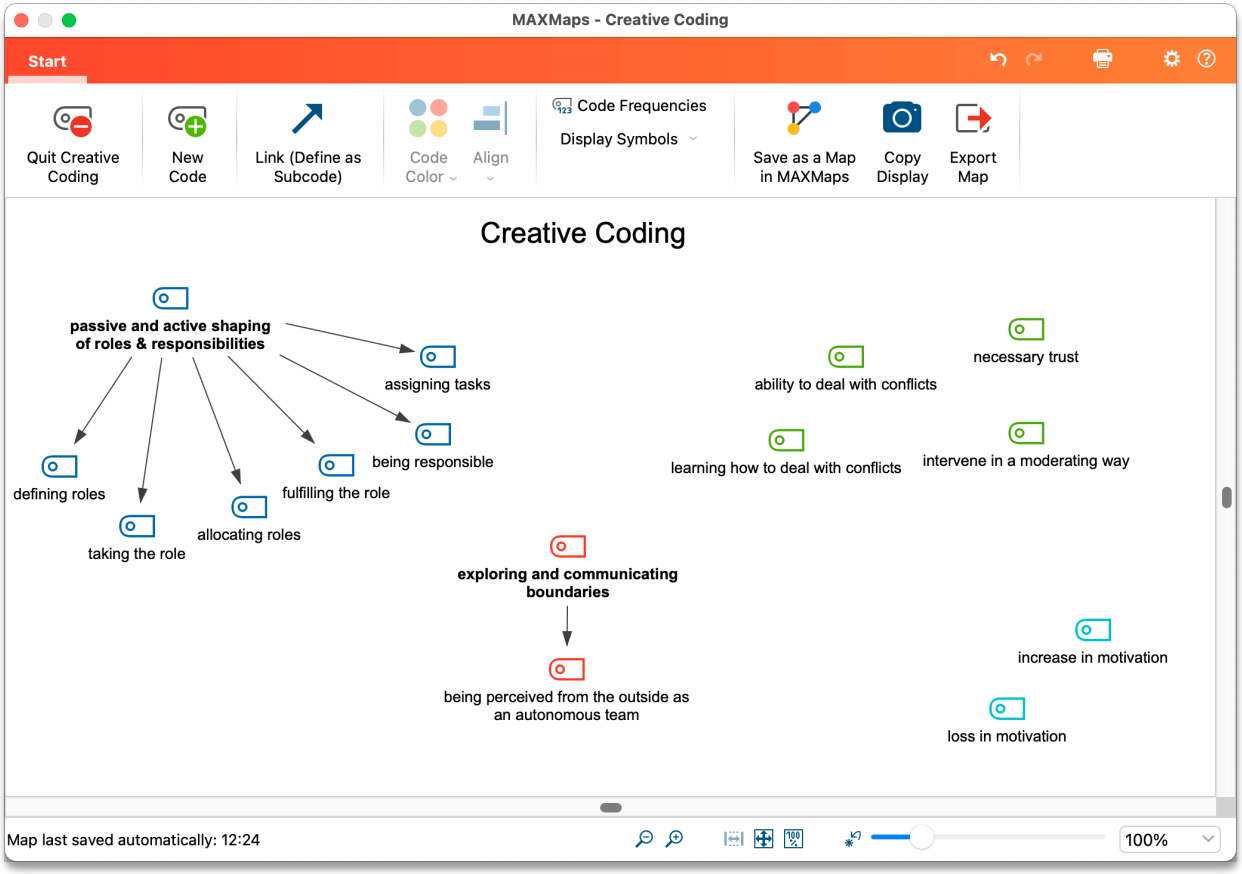


Fig. 8: Group codes on the Creative Coding "whiteboard"

In a grounded theory project, Creative Coding can be useful during open/initial coding to group first-level codes and to develop substantive categories in the sense of Strauss and Corbin for the formulation of the theory. However, Creative Coding can also be used later in the course of analysis to cluster intermediate-level categories into even more abstract categories or to merge similar categories. Creative coding can thus be used several times and at different stages in a grounded theory project.

Tips:

- If you have a lot of initial codes, you can use Creative Coding to group only some of the codes first and then include more codes in another cycle.
- For documentation purposes, you can use the icons at the top right of the Creative Coding window menu at any time to export the current view as an image or save it as a concept map in MAXMaps in the MAXQDA project (for more information on MAXMaps, see Chapter 6 “Creating diagrams”).
- To save the “Code System” as an image file before working with Creative Coding, you can use ***Codes > Export Code System > PNG Image***. To make that image of the code system available in the project, you can copy the exported file into a new free memo or import it as an image file into the “Document System”.

Organize codes in the “Code System”

The “Code System” is a central place for a Grounded Theory project. Here, from the beginning of coding, the codes are listed, and are gradually developed into more elaborate categories, for example, with the help of the *Creative Coding* function. The “Code System” should always be kept in mind and examined regularly in its entirety. It is usually little support for the analysis if the list is allowed to grow and grow (in consultations, I have seen 2,000 open codes at the top level – coupled with a certain despair about how to deal with the sheer mass). It is important to keep looking at the list with an analytical and conceptual eye and to have the courage to merge and develop abstract codes.

For efficient work with the “Code System”, it is best if you close the “Document System” above it, so that as much of the code system as possible is visible. This view is also recommended for coding a document. Various technical and methodological procedures are available for working in the “Code System”, which can be used as an alternative, supplement, or preparation to *Creative Coding*:

Sort codes

If you deal with many initially formed codes, you can first sort them alphabetically to identify codes which start with the same or similar wording. To sort the top-level codes, right-click the root in the code system and select ***Sort*** from the context menu. To sort lower-level subcodes, right-click their parent code and select the ***Subcodes...*** item.

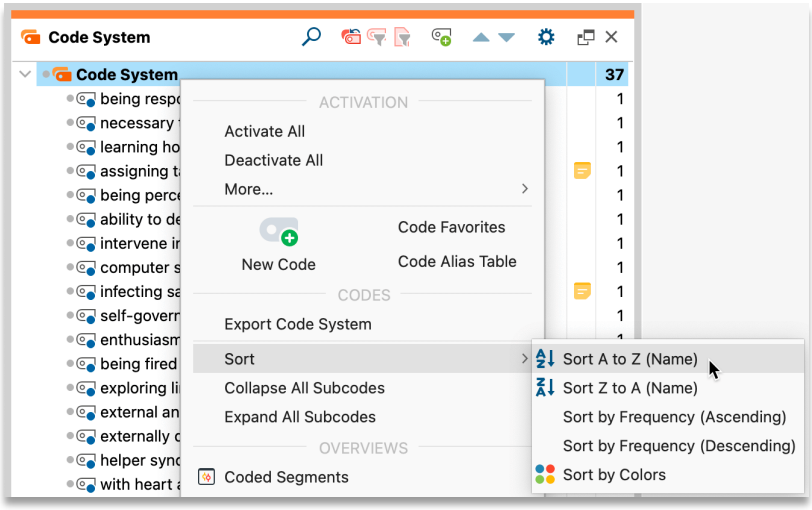


Fig. 9: Sort codes by names or by color

Tip: The codes can not only be sorted by name, but also by color. This is useful if you first go through the codes and assign them a color depending on the aspect addressed, so that the subsequent sorting then brings together aspects that belong together. This only makes sense if just a few aspects are distinguished. For more complex systems, you should group codes under parent codes.

Move codes

Single or even several selected codes can easily be moved with the mouse, for example to group similar codes. However, as soon as the list of open codes contains very many codes on one level and thus long paths have to be covered in the “Code System” window, the following procedure is more efficient:

1. Activate all codes to be placed or grouped together, for example, by clicking the gray circle to the left of the code symbol.
2. Right-click the root in the code system or an appropriate parent code and select ***Move Activated Codes Here*** from the context menu.

This will insert all currently activated codes directly under the selected code. The hierarchy of sublevels is preserved if the subcodes themselves are not activated, that is, only the parent code has been activated

(Fig. 10). This function is very useful when you need to identify all the initially formed codes that share a selected aspect, in order to then gather the associated codes in one place and process them further.

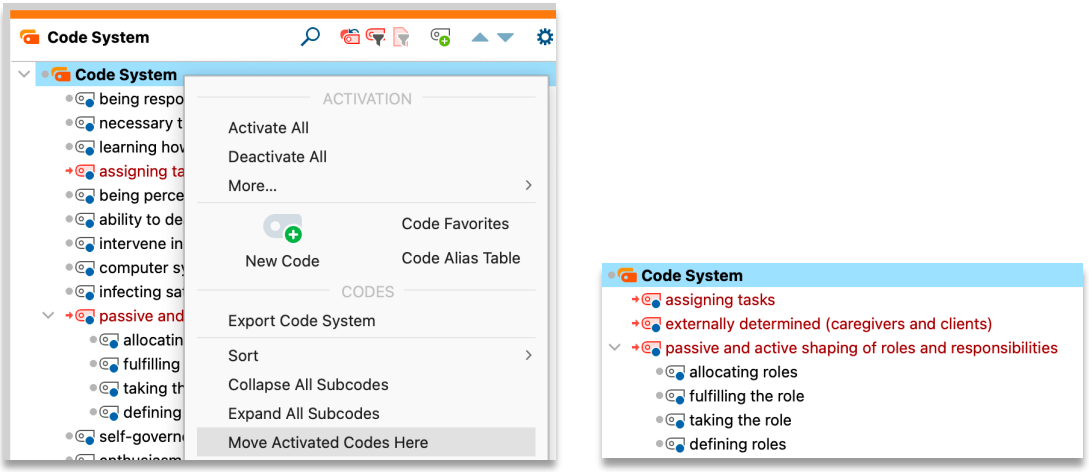


Fig. 10: Move activated codes. On the left you can see the activated codes, on the right the codes that have been moved to the top

Merge codes

Two similar codes are merged by dragging one code onto the other. The mouse button must be released when the mouse pointer touches the “Merge” field (Fig. 11). Multiple codes can be merged by selecting them while holding down the ***Alt*** key (Windows) or ***option*** key (Mac) and then calling the ***Merge Codes*** function from the context menu.

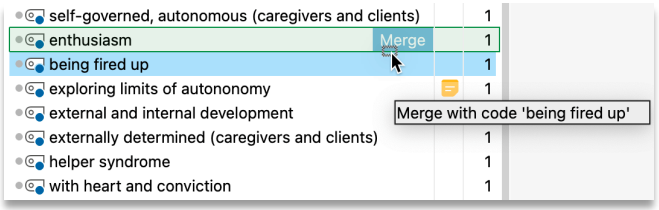


Fig. 11: Merge codes

When merging, MAXQDA adds a plus sign at the end of the remaining code name to mark the code as a merged code. The plus sign can usually be removed, since by default the procedure is also recorded in the code memo of the target code when other codes were merged with it. When merging, any existing memos of the codes are also added into that for the remaining code.

Elaborate properties of categories and their respective dimensions

In order to elaborate a category theoretically, Strauss and Corbin (1990) suggest working out the properties and dimensional characteristics of categories without waiting to complete open coding. They give the example of the category "Observe", which can be differentiated according to the following properties (with corresponding dimensional characteristics in parentheses):

- Frequency (often – never)
- Extent (much – little)
- Intensity (high – low)
- Duration (long – short)

The properties of this example have a universal character and can also be transferred to other categories, for example, to "autonomy of caregivers" or "pain experience". Although the material is used as a starting point for their development, not every property and dimension has to occur in the data material but can be deduced by theorizing. How can this entire process be implemented with MAXQDA? One possibility is to record properties and dimensions in the code memo supplemented by a detailed description, for example, in what way the extent and intensity of an observation can differ from one another. Another possibility is to create corresponding subcodes in the "Code System" similar to Fig. 12.

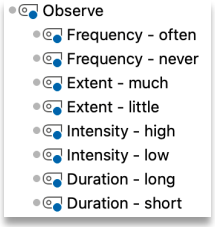


Fig. 12: Properties of categories and their dimensions as subcodes in the "Code System"

Advanced coding: Increasing the level of abstraction

As emphasized earlier, one task of the researcher in a grounded theory study is continuously to drive forward with the development of the categories and increase the level of abstraction in the analysis. As listed previously in Tab. 1, different authors have suggested different coding phases after the initial coding, for increasing the level of abstraction and elaborating the theory:

- Strauss and Corbin (1990) describe *axial coding*, in which, for example, with the help of the "coding paradigm", the relationships among the categories are elaborated and one or two key categories are identified. In *selective coding,* the analysis then focuses on these key categories, that is, "a large part of the material is recoded in order to clarify the relationships of the various subject-related concepts to the core categories and to bring about theoretical coherence" (Strübing, 2021, p. 17).
- Glaser (1978) describes *theoretical coding*, for which he provides a list of general codes grouped into "coding families". These can help to systematize the initial codes and determine the direction of further analysis (Flick, 2018b). The goal of theoretical coding is to identify a central category.
- Charmaz (2006) postulates *focused coding*, in which the focus is directed to selected analytically valuable codes that are conceptually elaborated and interrelated. Subsequent *theoretical coding* also integrates existing theories and research findings where appropriate (Flick, 2018b).

The coding paradigm of Strauss and Corbin

To elaborate the relationships between categories (and their subcategories) developed in open coding, Strauss and Corbin (1990) introduced what they call the coding paradigm. The paradigm is often visualized similarly to Fig. 13 and includes five domains:

- *Phenomenon:* What is the phenomenon under study?
- *Causes:* What led to the phenomenon under study?
- *Context and intervening conditions:* What is the context of the phenomenon under study (e.g., place, time) and what conditions influence it (e.g., important experiences, sociodemographic characteristics of the actors)?
- *Strategies:* What strategies do actors use to deal with the phenomenon?
- *Consequences:* What results from the phenomenon and the way actors deal with it?

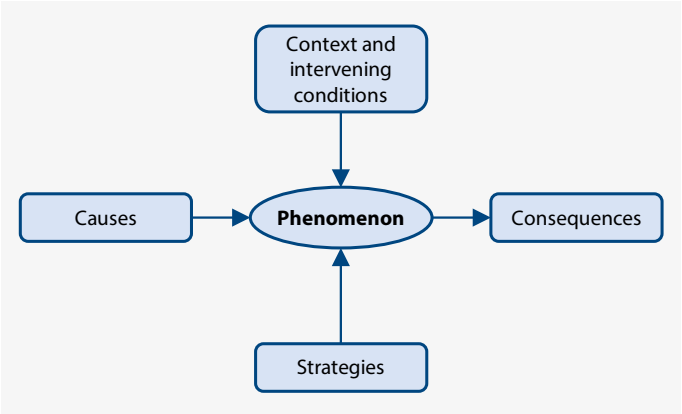


Fig. 13: The coding paradigm according to Strauss and Corbin (1990)

To use the coding paradigm in MAXQDA, MAXMaps is a good choice. MAXMaps is a workspace for creating concept maps and is started in the ***Visual Tools*** menu. After a new map has been created, the five domains of the paradigm can be placed and connected with arrows. Important categories are then dragged onto the area from the "Code System" window and placed in the appropriate areas. More important categories can be placed closer to a domain than less important ones.

In addition or alternatively, the coding paradigm can also be applied to the "Code System". For this purpose, the domains of the paradigm are created as header codes, so that important categories belonging to this domain can be placed as subcodes, and further sub-sub-levels can also be used (Fig. 15).

Tip: The coding paradigm can be applied to one or two cases initially. The resulting categories and relationships can then successively be refined as further cases are analyzed.

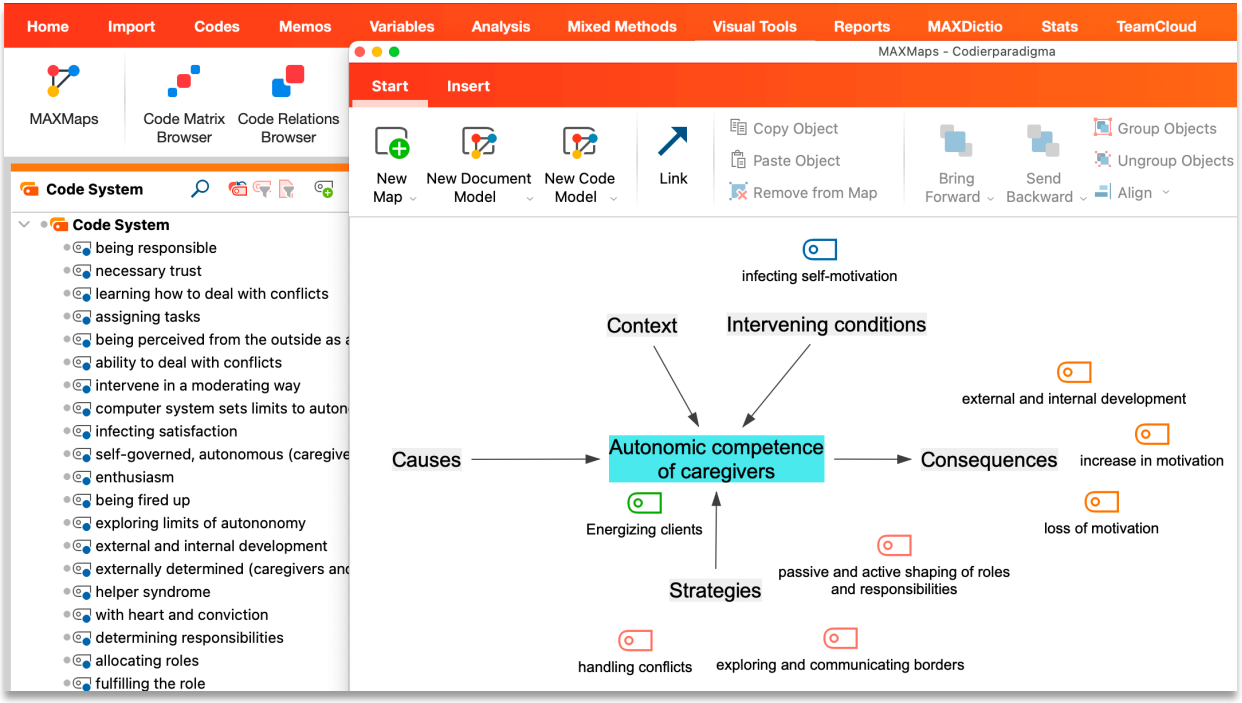


Fig. 14: Creating a concept map in MAXMaps using the domains of Strauss and Corbin's coding paradigm

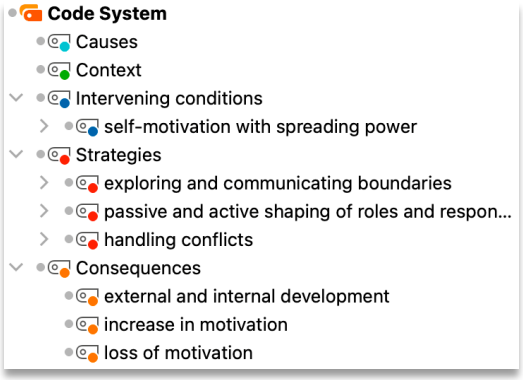


Fig. 15: "Code System" with elements of the coding paradigm as parent codes

The coding families of Glaser

Glaser (1978) formulated more than ten "coding families", each with multiple terms, including the following, which I present here with recourse to Thornberg and Charmaz (2014) and Flick (2018b):

- 6 Cs: causes, contexts, contingencies, consequences, conditions
- Process: stages, phases, phasing, transitions, passages, careers, chains sequences, etc.
- Degree: extent, level, intensity, range, amount, continuum, etc.
- Consensus: contracts, agreements, conformity, homogeneity-heterogeneity, etc.

The terms have a universal character and are intended to promote conceptual thinking about the identified categories. In MAXQDA, these families can be used in various ways. In all grounded theory projects, they should be copied into a free memo from where they can be accessed easily without a laborious effort to look them up. In the "Code System", individual families can be used to structure the categories, as with the coding paradigm. And when writing code memos, the coding families help to identify the different aspects covered by a category.

Key points summarized:

- Think conceptually and build codes and categories that are analytical in nature (and not just thematic codes).
- Note that there are different approaches to coding in the different grounded theory styles.
- For the formation of initial/open codes in the first coding phase, you can use the "Open coding" mode.
- To create "in-vivo codes", you can use the icon in the "Document Browser", the keyboard shortcuts ***Alt+I*** (Windows) and ***cmd+option+I*** (Mac), or the context menu.
- To group, abstract, and combine initial codes into more abstract categories, you can use the ***Codes > Creative Coding*** workspace.
- Codes in the "Code System" can be merged by clicking and dragging with the mouse.
- Properties and dimensions of categories can be recorded in code memos or by using subcodes.

- ❖ Use the coding paradigm as a graphic in MAXMaps or as parent codes in the "Code System" to elaborate relationships between categories and to identify central categories.
- ❖ Note the coding families in a free memo so that you can always access them.

5 Writing memos

What are memos?

In memo texts, researchers record their thoughts, findings, decisions, and notes. Memo writing is a central component of grounded theory projects, intended to support the analytical process and theory development (Strübing, 2021, p. 35). "Code for 30 minutes, write a memo for 30 minutes" is a motto postulated by Günter Mey in a workshop to illustrate the importance of taking time to reflect on the codes and categories generated and to note the results in memos. The numerous memos written in the course of a grounded theory study serve purposes of reflection, interpretation, and deepening the analysis, and they also form an important basis for the theory to be formulated.

Distinguish memos

There are several places in MAXQDA where memos can be pinned (Fig. 16):

- ❖ "Document System": a memo can be written for each document, document group, and document set. In addition, the *project memo is* available at the top entry of the list.
- ❖ "Code System": a *code memo* can be written for each code.
- ❖ "Document Browser" and "Multimedia Browser": any number of *in-document memos* can be attached to individual text passages or video segments.

MAXQDA also offers *free memos*, which are not directly assigned to any element, but are independent of data and categories. They can be called up in the ***Memos*** tab.

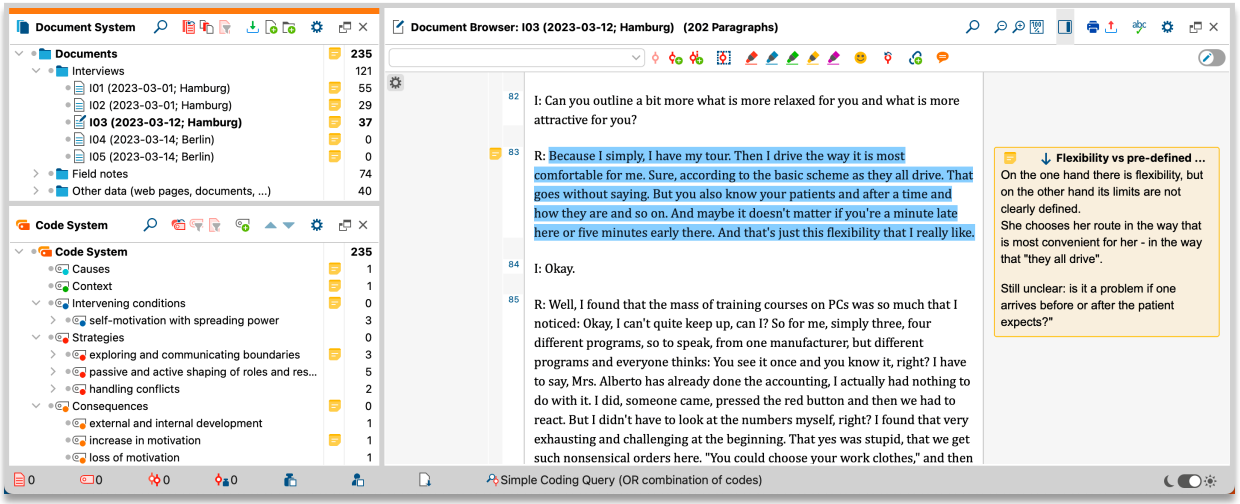


Fig. 16: Memos on the MAXQDA interface, symbolized by yellow sticky note icons

The different places where a memo can be pinned make it easy to distinguish between different types of memos in grounded theory projects. The following list presents some possibilities for using the different memos:

Project memo (top entry in the “Document System”)

- Research interest, focus of the study, and research questions.
- Important decisions in the research process and progress of the study. Alternatively, a free memo or the ***Home > Logbook*** function can be used for this purpose.
- Information on theoretical sampling with explanations of selection strategies and decisions made.

Document group and document set memos

- Background information on a sampling cycle or data type.
- Results of case comparisons.

Document Memos

- Background information on individual cases and situations of data collection.

- Summaries of key case features and case content.
- Analytically relevant features of a case, especially in comparison to other cases.

In-document and in-media memos (on a text passage or video clip)

- Notes that relate directly to the data, such as contradictions within a case.
- Suspected connections and vague ideas (in the style of "something similar was found in another case, so it may be worthwhile to look for any commonality and connection").
- Results of comparing a specific text passage with other text passages from the same or a different case.
- Summaries of the data, for example, summary in English for relevant sections of a Spanish transcript.

Code memos

- Analytical thinking about codes/categories/concepts, for example, their properties, dimensions, and relations to other categories.
- Decisions to merge codes.
- Source references for in vivo codes.

Free memos

- Pre-assumptions, pre-concepts, and "sensitizing concepts" (Bowen, 2006; Charmaz, 2014), with which the researcher entered the field.
- Thoughts on an overarching concept or key category, recurring patterns, and similarities and differences between cases.
- Integrative theory memos that summarize key findings from multiple memos.

Compose memos

All memos in MAXQDA consist of a title and a memo text (Fig. 17). For document and code memos, MAXQDA automatically adopts the code or document name as the title, which should not be changed to be able to distinguish the memos easily. For free memos and for in-document memos, a meaningful title should be assigned immediately upon creation to be able to facilitate finding the memos again later.

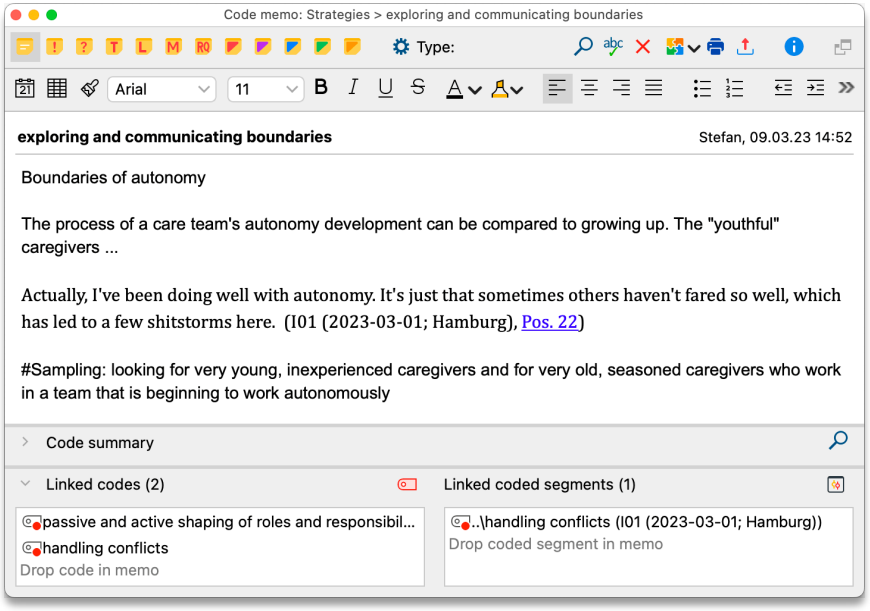
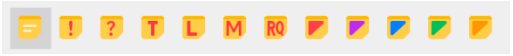


Fig. 17: Opened code memo. The title of the window is the name of the code to which the memo was assigned

Depending on the content or meaning of a memo, it can be assigned one of several icons available for selection, which will appear on the interface even after the memo is closed:



For example, the symbol “T” can be used for advanced memos with great importance for the developed theory, and the symbol with exclamation mark “!” can refer to places where you still want to be active. With the help of the colors red, purple, blue, green, orange the processing status of documents can be recorded. You can define, change, and view the meanings you have assigned to the various memo symbols in a specific MAXQDA project by clicking on the gear icon in the memo editor (Fig. 18). The type assigned to a memo is output in many places in MAXQDA, for example, in the info box that appears when the mouse pointer is held over a memo symbol.

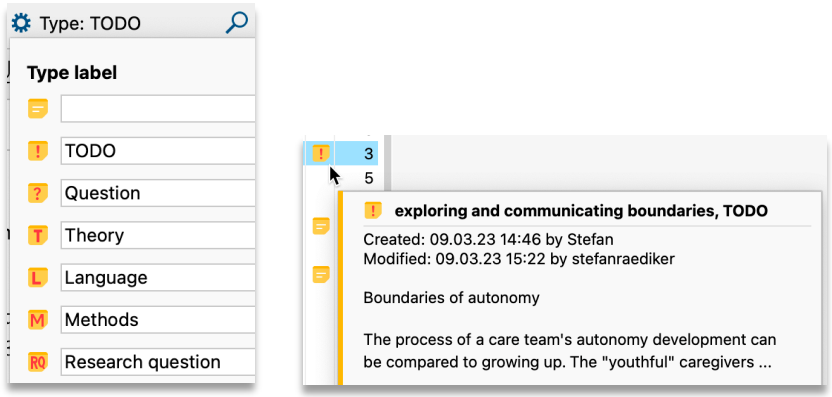


Fig. 18: Define labels for the different memo symbols (= memo types) (left). Display of the label in the title of the info box for a memo symbol (right)

Tips for writing memos:

- Include hashtags in the memo text to facilitate searching for notes and thoughts later, for example #context or #sampling.
- MAXQDA shows who created and last edited a memo. If you want to date individual entries within the text, you can click on the calendar page icon.
- It is also possible to create tables, for example, for properties and dimensions of a category.

Link memos

It is typical in grounded theory studies to have the desire to make connections between sections in the data, memos, codes, and categories. For example, in integrative theory memos the original memos should be linked, or the code memo of a category could be linked to other categories, their respective code memo, and selected coded segments. MAXQDA provides various types of links for this purpose.

Internal links can be used to create a connection between two points in the data, for example, between two contradictory statements by one person (Fig. 19). To add an internal link, a text passage is highlighted, the option ***Internal Link*** is called up from the context menu, and the same process is repeated at the target of the link. Internal links are highlighted in blue, and one click leads from the start to the target

and vice versa. In the same way, a text passage in the data can be linked to a passage in a memo, and two text passages from different memos can also be linked in this way.

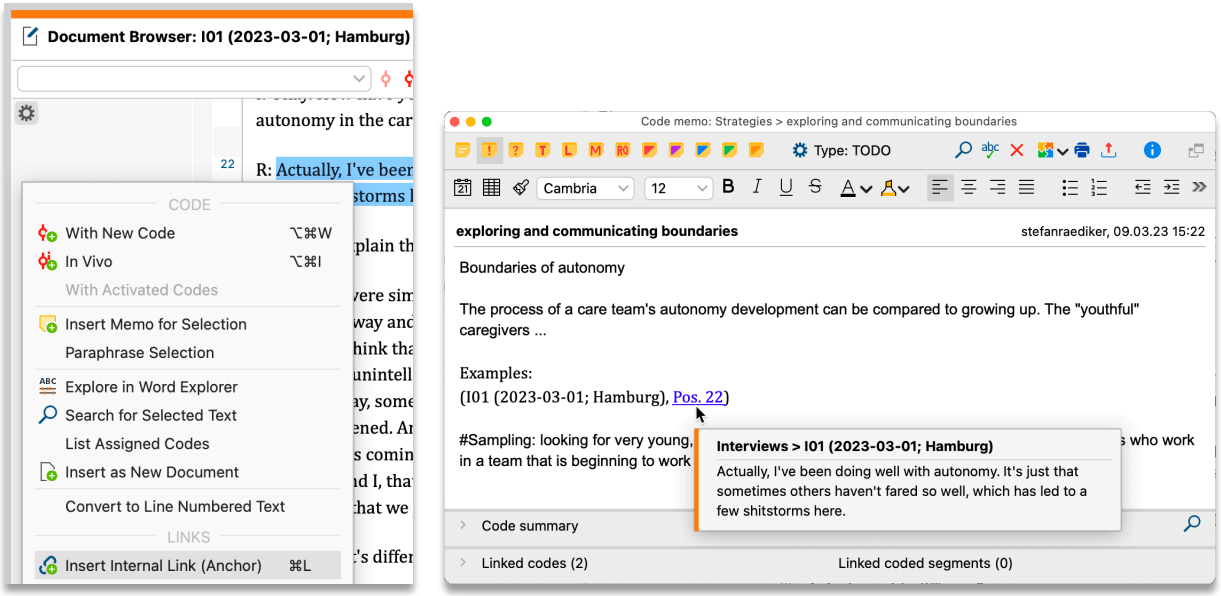


Fig. 19: Creating an internal link using the context menu of a highlighted text passage (left). Created link in a memo with preview of the linked destination (right).

Any number of codes can be assigned to each memo. In this way it is possible not only to explain the relationships between different codes or categories in the memo text, but also to create clickable links. To assign a code to a memo, you can drag the code name from the “Code System” directly into the opened memo window. The code name will then appear at the bottom left in the “Linked Codes” section as shown in Fig. 21. Clicking on the code selects it in the “Code System” window. Whether a code is linked to one or more memos can be seen in the context menu: an option ***Linked Memos*** appears, with which all linked memos are displayed in MAXQDA’s Memo Manager.

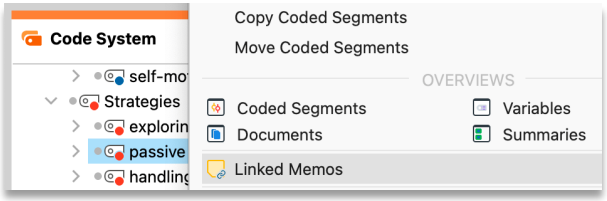


Fig. 20: Access all memos linked to a code

In a similar way, a coded segment can also be assigned to any opened memo, for example, to assign a segment with a particularly succinct interview statement: from the “Document Browser”, the code name or the coding stripe displayed next to the text is dragged into the lower part of the memo (Fig. 21).

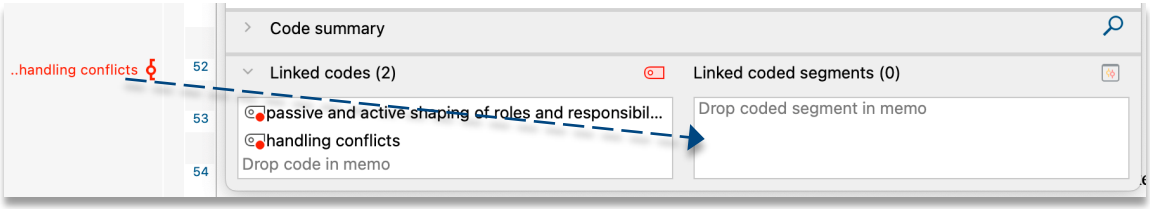


Fig. 21: Assign coded segment to memo by dragging the code name or vertical coding stripe from the “Document Browser”

Organize, edit, and search memos in the Memo Manager

It is not unusual for a grounded theory project to have many memos, sometimes more than 100. To keep track of the memos and their contents, the Memo Manager helps, where selected or all memos of a project can be displayed, edited, and searched. The Memo Manager is called up in the ***Memos*** menu tab, either with all memos by clicking on the icon of the same name or with a preselection of specific memos by clicking on ***Free Memos***, ***Code Memos***, ***In-Document Memos*** etc.

The Memo Manager presents all memos in a tree structure and by clicking on the icons at the top you can show or hide individual types of memos. Below each icon you can see how many memos of the respective type are present in the project.

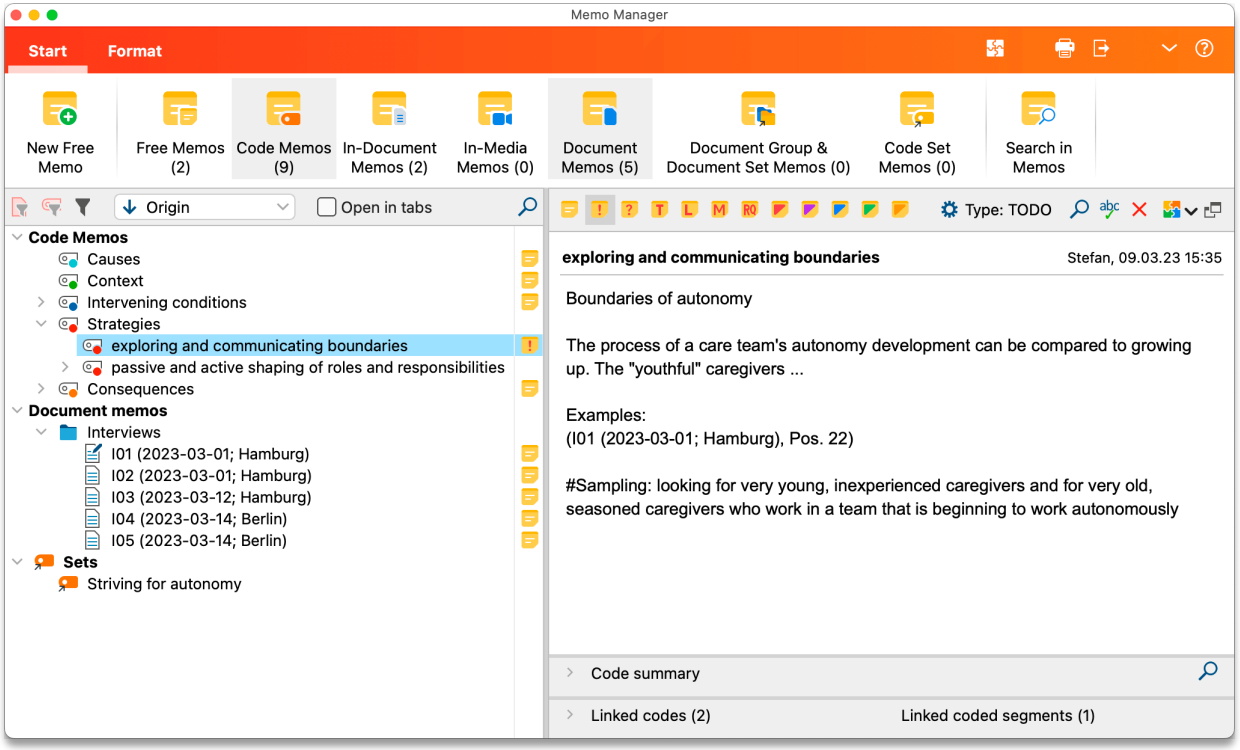
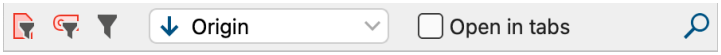


Fig. 22: The Memo Manager displays selected or all memos of a project

How does the Memo Manager support a grounded theory project?

- The memos can be conveniently sorted and filtered by last edit date, title, author, or memo symbol – various options are available above the memo tree for this purpose:



- From the ***Start*** menu of the Memo Manager, you can access a search function, with which all memos of the project can be searched for a key word.
- Memos can be grouped into sets to create thematically related memos. Right-click on the word “Set” at the bottom of the memo tree to select ***New Set*** and drag memos from the tree into the new set.

Tip: You can create new free memos directly in the Memo Manager by clicking on the icon of the same name in the menu tab. Since free memos are not assigned to any element in MAXQDA, you can only open them in the Memo Manager.

Integrate memos

Corbin and Strauss (2015) provide some impressive examples of how writing integrative memos can steadily increase their level of abstraction. In so-called "summary memos" (ibid., p. 122), the findings found so far are regularly compiled and described in condensed form. MAXQDA offers a free memo for writing such an integrative memo. To have easy access to several memos whose contents are to be integrated, the following options are available in the Memo Manager:

- The best way to open the free memo in a floating window is to select ***Open Memo in New Window*** from the context menu directly after creating the memo (Fig. 23).
- Existing memos can also be opened in this way and then placed on a second screen, if available.
- The ***Open in tabs*** option displays each memo in a new tab just like in an Internet browser, so it is easy to switch between tabs.
- The thematically related memos can be grouped together in a set.

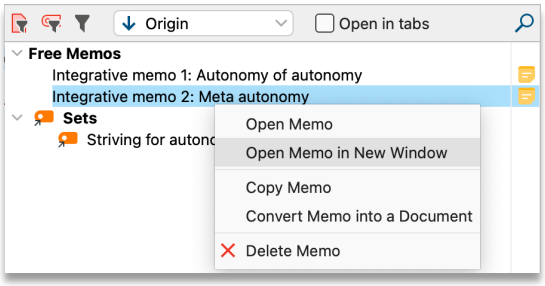


Fig. 23: Open memo in new window

Tip: You can convert a memo into a text document at any time by right-clicking the memo icon and selecting the corresponding function. This makes it possible to code the memo like a document, annotate it with further memos, etc. Please note that the interactivity of any links with codes and coded segments will be lost.

Key points summarized:

- Think about what information should be stored in which memos.
- Use different memo icons to distinguish the contents of memos easily.
- Assign meaningful titles to free memos and in-document memos.
- Different memos are accessed via the ***Memos*** menu tab – the memos are compiled in the Memo Manager, where they can be organized, edited, and integrated.
- To connect memos with other elements of the project, you can use internal links and drag codes and coded segments to the memo window.

6 Creating diagrams

At any time during a grounded theory study, diagrams can be created, for example, to visualize the connections between individual categories in a concept map and to discover or highlight a key category in this process. Creating a diagram supports analysis and theory development, because it forces researchers to relate individual elements of the theory using graphical elements on a two-dimensional surface and to model the developing theory.

In MAXQDA, diagrams are created using the ***Visual Tools > MAXMaps*** workspace, which was introduced in Chapter 4 for the visualization of Strauss and Corbin's coding paradigm. MAXMaps represents a very flexible working environment in which project elements such as documents, codes, coded segments, and memos as well as graphical elements such as arrows, boxes, and texts can be placed and formatted. The advantage of MAXQDA over a graphics program is that any project element can be inserted into a map by clicking and dragging, and the memos, documents, codes, and coded segments are interactively linked to their counterpart in the project. For example, double-clicking on a memo symbol opens the memo directly and allows you to edit it.

Create free concept maps

By clicking on the ***New Map*** icon, an empty map is created, which can then be flexibly designed and used. For example, you can add shapes like circles and triangles in the map, and after dragging two codes from the "Code System" into the map, a connecting line between the two codes or other elements can be drawn using ***Start > Link*** (Fig. 24). This can be changed to a directional arrow using the ***Line Format*** menu tab.

By right-clicking on a line, it can be labeled (by using ***Add Label*** from the context menu) in order to name the type of relation, if required. For the naming of category relations, you will find references to the semantic relations according to Spradley (1980) in the grounded theory literature, where cross-references are also made to Glaser's coding families (see Flick, 2018b; Urquhart, 2023). Here is an overview of the relations in abbreviated form:

- is a form of
- is part of
- is one way to
- is stage from
- is cause of
- is result of
- is place for
- is property of
- is used for

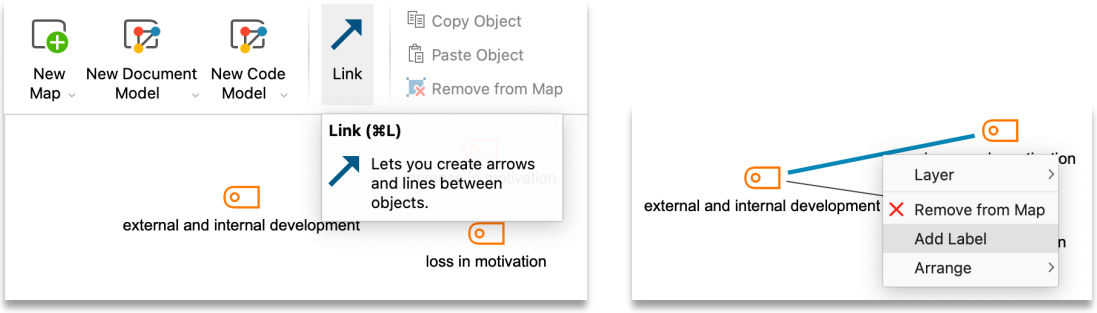


Fig. 24: Drawing a connecting line between two elements in MAXMaps (left) and assigning a label (right)

Tipp: MAXMaps is also suitable for creating a "conditional/consequential matrix" according to Strauss and Corbin or a "situational Map" according to Clarke.

Use model templates

MAXMaps offers templates for case-oriented and category-oriented representations. After selecting a model via ***Start > New Document Model*** or ***New Code Model***, elements can be selected from the "Document System" or the "Code System" and options can be set for the model. From these, a concept map is created automatically and this can then be customized as required.

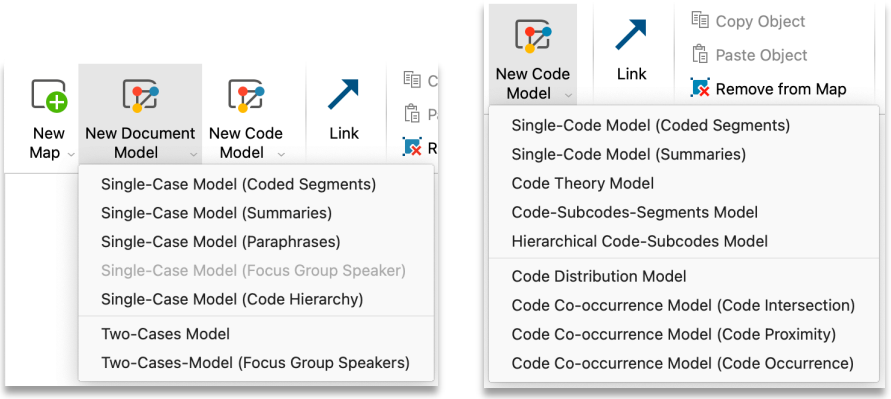


Fig. 25: Selecting model templates in the "Start" menu of MAXMaps

Many of the model templates are suitable for grounded theory projects, such as the following:

- Using a *Single Case Model,* the coded text segments of an interview or the coded excerpts of photos can be placed around a case sorted by the codes and categories.
- The *Code Co-occurrence Model* uses connecting lines to visualize how often two codes co-occur in a text section or somewhere in a document.
- The *Code Theory Model* represents the code memo and all linked memos for a selected code. The subcodes and their associated memos can also be integrated into the representation to increase complexity.

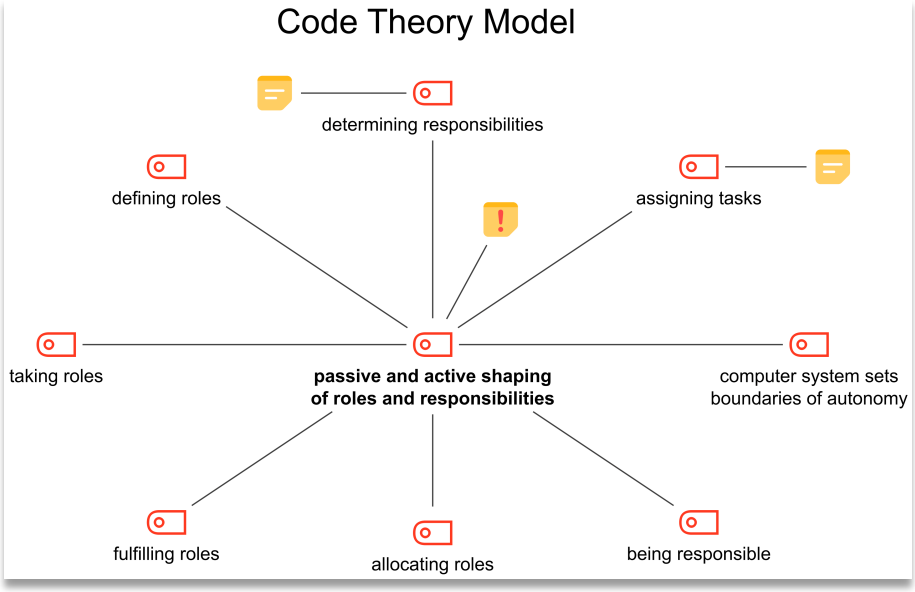


Fig. 26: Code Theory Model in MAXMaps including first level subcategories

Key points summarized:

- Creating concept maps will help you draw out connections between categories, identify a key category, and model a theory.
- Concept maps are created and edited via ***Visual Tools > MAXMaps***.
- Take advantage of MAXMaps' interactivity by, for example, double-clicking on code or memo symbols.
- Model templates are available for case- and category-oriented concept maps.

7 Finalizing and writing-up the theory

Using more MAXQDA functions

The development of a grounded theory is a continuous process that begins with the initial coding of the data. Further data, collected in the sense of theoretical sampling, are added to the "Document System". Using the new data, the current state of the theory can be repeatedly checked, questioned, optimized, and finalized. In addition to the features already presented, numerous other MAXQDA functions are available for the entire analysis process, such as the following:

- Using ***Analysis > Text Search & Autocode***, you can quickly search for the occurrence of specific terms in all texts.
- Word-based functions such as ***Visual Tools > Word Cloud*** and numerous functions in the ***MAXDictio*** menu tab make it possible to systematically examine the use of terms, both for purposes of exploration and for checking conjectures.
- The ***Visual Tools*** menu tab provides various functions for visualizing coded data. For example, the ***Code Matrix Browser*** can be used to analyze the distribution of codes among individual cases in order to find out which topic plays a major role among the actors in relation to other topics. However, the visualization of codes only makes sense in the advanced coding stage, after the initial open codes have been grouped into categories, and events and actions from different documents have been assigned to them.

Compiling aspects of the theory in the QTT workspace

The results of a grounded theory study can take a variety of forms and be presented as a list of hypotheses, a diagram, a typology, or a mixture of these and other components. As a general rule, it should not be just an unlinked collection of themes:

> "The results of a GT study are communicated as a set of concepts, related to each other in an interrelated whole, and expressed in the production of a substantive theory. A substantive theory is a theoretical interpretation or explanation of a studied phenomenon" (Tie et al., 2019, p. 7)

In MAXQDA, the "QTT" function can be used to compile the findings (***Analysis > QTT: Questions - Themes - Theories***). The QTT is a workspace that can act as a bridge between the MAXQDA project and the research

report. To use the QTT, you must first create a worksheet in which the research interest is entered in the form of research questions. Then, as the analysis proceeds, codes, categories, coded segments, memos, concept maps, and other elements of analysis that play a particular role in the theory can be collected in the worksheet. Insights can be formulated for each element, and these are automatically compiled in an area called “Integration of Insights” (Fig. 27).

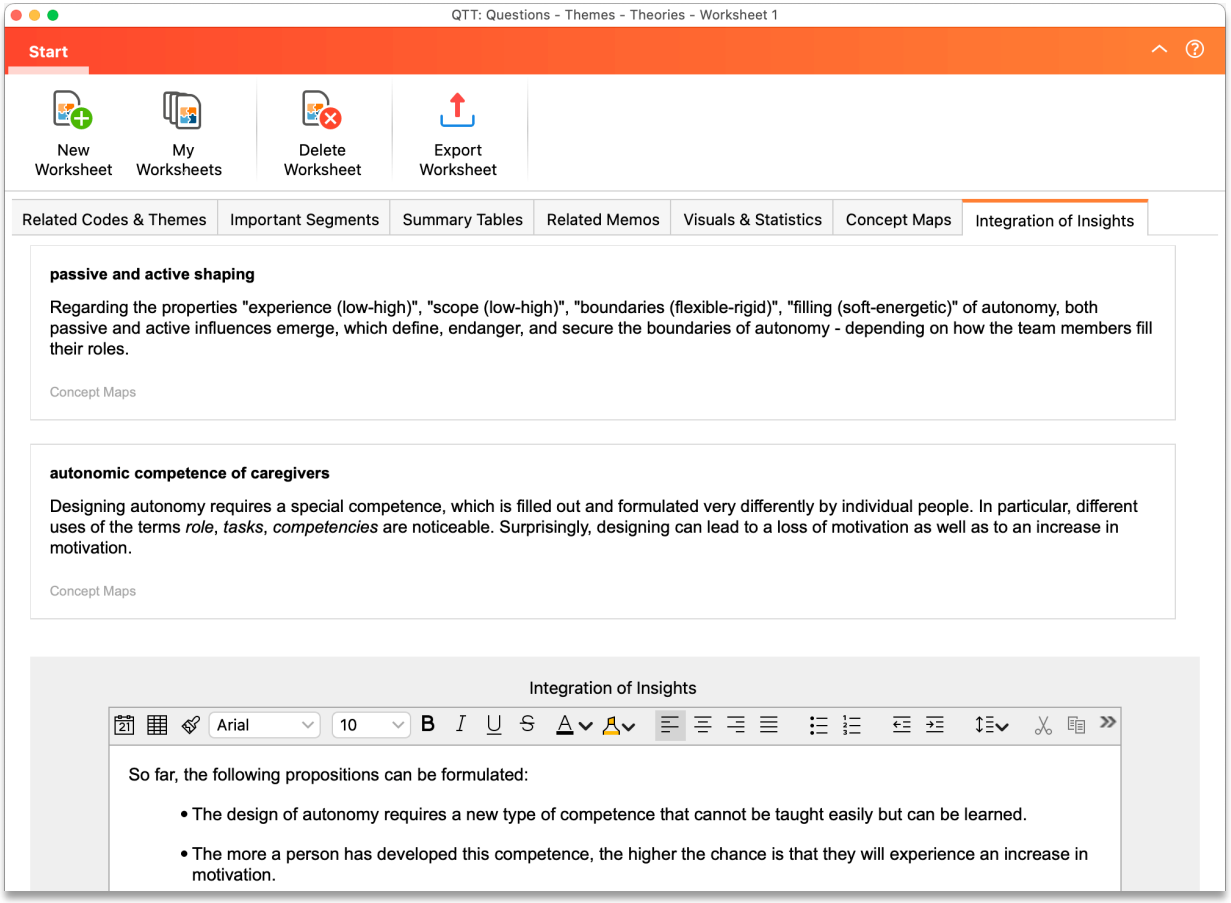


Fig. 27: Section “Integration of Insights” in a QTT worksheet

Key points summarized:

- You can search the entire project using the ***Analysis > Text Search & Autocode*** function.
- Word-based functions such as ***Visual Tools > Word Cloud*** enrich the analysis.
- The ***Analysis > QTT: Questions - Themes - Theories*** workspace is well suited for continuously collecting important aspects of the theory.

8 To conclude: A few general notes

Documentation of the procedure

In order to save different analysis states and thus easily be able to present the course of the analysis in the later research report, copies of the MAXQDA project should be created regularly in addition to technical backups. It may be helpful to save the current analysis state, for example, before inserting further data or before a major revision of the code system (you can use ***Home > Save Project As*** for this task). Images of the current code system can be created regularly via ***Codes > Export Code System > PNG image***.

Memos documenting key decisions and theoretical sampling are an important source for writing-up the methods chapter of the report.

Grounded theory or qualitative content analysis?

In my observation, many studies labeled as grounded theory projects tend to follow the principles of qualitative content analysis (Kuckartz & Rädiker, 2023; Mayring, 2021; Schreier, 2012) more than the principles of grounded theory. Some typical indications are listed here:

- Numerous concepts are defined and fixed in advance and are created as codes in the “Code System” before the analysis starts.
- Far from being a theory, the result consists only of a collection of categories that may have a content-structuring character, but not an analytical one.
- The code memos in MAXQDA contain only category definitions and notes on applying the category.
- Great importance is given to the intercoder agreement within a team of analysts.

- Only one type of data was analyzed, and no additional data was added to the MAXQDA project during the course of the project.
- The research interest focuses on opinions and knowledge of persons, rather than actions, processes, or chronological sequences of developments.

If some of these indications are true, this does not necessarily mean that no grounded theory study was conducted. However, it should have become clear in this guide that inductively forming categories from the data is not sufficient to classify a project as a grounded theory study. Inductive creation of categories is also possible in qualitative content analysis (Kuckartz & Rädiker, 2023) and thematic analysis (Boyatzis, 1998; Braun & Clarke, 2021; Guest et al., 2012); however, the objective and the type of categories formed differ: in a grounded theory project, the focus is less on forming descriptive and thematically oriented categories and much more on conceptually and analytically oriented ones. The motto in grounded theory is: "coding for theory, not for themes" (see Charmaz, 2014).

The added value of MAXQDA when conducting grounded theory studies

The computer-assisted implementation of a grounded theory study with MAXQDA has many advantages:

- There is access to all relevant data with just a few mouse clicks.
- The analysis procedure is transparent and traceable.
- Constant comparison is supported at all levels.
- MAXQDA also makes it easy to combine different styles of grounded theory on a pragmatic level, so using the coding paradigm of Strauss and Corbin does not preclude using Glaser's coding families as well.

As ever, the better you get to know the functions of MAXQDA, the better you will be able to use them for the particular purposes of your grounded theory project.

References

Birks, M., & Mills, J. (2015). *Grounded theory: A practical guide* (2nd ed.). SAGE.

Boyatzis, R. E. (1998). *Transforming qualitative information: Thematic analysis and code development*. SAGE.

Braun, V., & Clarke, V. (2021). *Thematic analysis: A practical guide to understanding and doing*. SAGE.

Bryant, A., & Charmaz, K. (2007). Introduction: Grounded theory research: Methods and practices. In A. Bryant & K. Charmaz (Eds.), *The SAGE handbook of grounded theory* (pp. 1–28). SAGE.

Charmaz, K. (2006). *Constructing grounded theory*. SAGE.

Charmaz, K. (2014). *Constructing grounded theory* (2nd ed.). SAGE.

Cisneros-Puebla, C. A. (2004). "To learn to think conceptually." Juliet Corbin in conversation with Cesar A. Cisneros-Puebla [53 paragraphs]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research, 5*(3), Art. 32.

Clarke, A. E. (2005). *Situational analysis: Grounded theory after the postmodern turn*. SAGE.

Corbin, J. M., & Strauss, A. L. (2015). *Basics of qualitative research: Techniques and procedures for developing grounded theory* (4th ed.). SAGE.

Flick, U. (2018a). *An introduction to qualitative research* (6th edition). SAGE.

Flick, U. (2018b). *Doing grounded theory*. SAGE.

Glaser, B. G. (1978). *Theoretical sensitivity: Advances in the methodology of grounded theory*. Sociology Press.

Glaser, B. G., & Strauss, A. L. (1967). *The discovery of grounded theory*. Aldine.

Guest, G., MacQueen, K. M., & Namey, E. E. (2012). *Applied thematic analysis*. SAGE.

Jacques, D. (2021). Using MAXQDA in ethnographic research: An example with coding, analyzing, and writing. In M. C. Gizzi & S. Rädiker (Eds.), *The practice of qualitative data analysis: Research examples using MAXQDA* (pp. 17–33). MAXQDA Press. https://doi.org/10.36192/978-3-948768058

Kuckartz, U., & Rädiker, S. (2023). *Qualitative content analysis: A guide to methods, practice and using software* (2nd ed.). SAGE.

Mayring, P. (2021). *Qualitative content analysis: A step-by-step guide*. SAGE.

Schreier, M. (2012). *Qualitative content analysis in practice*. SAGE.

Spradley, J. P. (1980). *Participant observation*. Holt, Rinehart and Winston.

Strauss, A. L. (1987). *Qualitative analysis for social scientists*. Cambridge University Press.

Strauss, A. L., & Corbin, J. M. (1990). *Basics of qualitative research: Grounded theory procedures and techniques*. SAGE.

Strauss, A. L., & Corbin, J. M. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory* (2nd ed.). SAGE.

Strübing, J. (2021). *Grounded Theory: Zur sozialtheoretischen und epistemologischen Fundierung eines pragmatistischen Forschungsstils*. Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-24425-5

Thornberg, R., & Charmaz, K. (2014). Grounded Theory and Theoretical Coding. In U. Flick, *The SAGE Handbook of Qualitative Data Analysis* (pp. 153–169). SAGE.

Tie, Y. C., Birks, M., & Francis, K. (2019). Grounded theory research: A design framework for novice researchers. *SAGE Open Medicine, 7*, 1–8. https://doi.org/10.1177/2050312118822927

Urquhart, C. (2023). *Grounded theory for qualitative research: A practical guide* (2nd ed.). SAGE.

After a brief overview of basic principles of grounded theory studies, this guide will show you how to conduct grounded theory projects with MAXQDA:

- Organizing data
- Coding data
- Writing memos
- Creating diagrams
- Finalizing and writing-up the theory

Each chapter ends with a brief summary of the key points.

The guide offers orientation as well as practical advice on how to conduct a study using a grounded theory approach. It is a valuable companion for novices and experienced MAXQDA users.

Dr. Stefan Rädiker is a consultant and trainer for research methods and evaluation. He holds a doctorate in educational sciences and his research focuses on computer-assisted analysis of qualitative and mixed methods data (www.research-coaching.de).

MAXQDA PRESS
www.maxqda-press.com

ISBN 978-3-948768-16-4
9 783948 768164