

# **INTERVIEW QUESTION GENERATION & QUALITATIVE DATA ANALYSIS SOFTWARE A MODULARIZED CURRICULUM**

Christian Schmieder

UW-Madison, School of Education

UW-Cooperative Extension

[www.squaremethodology.com](http://www.squaremethodology.com)

**The development and public availability of these curricular materials were made possible through the Replicable Instructional Technology Grant, awarded by MERIT Library at the School of Education, University of Wisconsin-Madison.**

This guide is licensed under Creative Commons [\(CC BY-NC-SA 4.0\)](https://creativecommons.org/licenses/by-nc-sa/4.0/)  
[Attribution-NonCommercial-ShareAlike 4.0 International \(CC BY-NC-SA 4.0\)](https://creativecommons.org/licenses/by-nc-sa/4.0/)

## Contents

About the curriculum .....	3
Notes on Method and Origin .....	3
Using the Curriculum .....	5
Using different software .....	5
Using no QDA software.....	5
Let me know what you think! .....	5
Scenario 1: Every Participant has their own project.....	6
Using the modules as a full workshop .....	6
Scenario 2: Each group works on the same project.....	6
Using the modules as a full workshop .....	6
Prep Notes .....	7
Workshop Modules.....	8
Appendix .....	18
A: Notes on Mini-Lecture .....	18
Questions & Quality of Questions.....	18
Question Generation Strategy .....	18
Guide Layout .....	19
B: Intro MAXQDA (45) .....	21
General Function & Layout (5).....	22
Import (10) .....	22
Memos & Edit Mode (5).....	22
Coding (15) .....	23
Queries (5).....	23
Questions & Wrap-up (5) .....	23
Participant Example .....	24
Brainstorming for Questions.....	24
Annotation .....	25
Sorting.....	26
Code Line.....	26
Code Relations Browser .....	27
Set Intersections .....	28

## About the curriculum

With this curriculum, participants can generate a question guide for open and semi-structured interviews while actively using basic functions of a qualitative data analysis software. The idea behind this curriculum is to facilitate the meaningful, critical, and task-centered integration of qualitative data analysis software into methodology teaching. This approach is grounded in Carvajal's (2002) critique of issues connected to qualitative data analysis software teaching, and it aligns with di Gregorios (2014) call for "a new pedagogic approach to teaching CAQDAS".

As much as this curriculum is about generating questions: throughout the process, participants will *use QDA software in order to pursue analytic tasks*. They will explore features of the software; they will discover things they need the software to do; they will interact with their questions through the software; and they will bump against the limits of the software.

The setting of the use of software is a workshop focusing on generating questions – in my experience this is perceived by participants as a less high-stakes and high-pressure setting, compared to "analysis workshops". And yet: participants actually *do* analysis. They create and debate systems of classification, they sort, re-assemble, annotate, complicate, simplify, categorize and synthesize data consisting of their interview questions. This makes this curriculum a useful primer for workshop series/classes that focus on the analysis of qualitative data. This curriculum features the use of MAXQDA, but the curriculum can be easily adapted to any qualitative data analysis software.

**The development and public availability of these curricular materials were made possible through the Replicable Instructional Technology Grant, awarded by MERIT Library at the School of Education, University of Wisconsin-Madison. I also want to thank the UW School of Education Qualitative Research Methods Minor Committee for its ongoing support. Many thanks go also to the all the participants in the test workshops held at UW-Madison.**

## Notes on Method and Origin

The model of interview question generation that I use in this curriculum is based on the work of two German methodologists from the Freiburg circle of qualitative methodology (<http://www.iqs-forschung.de/>). Prof. Dr. Cornelia Helfferich (2005) has created this work flow at the Protestant University for Applied Sciences Freiburg, and it has been further developed by my methods mentor, the late Dr. Jan Kruse (2014) at the Institute of Sociology at Albert-Ludwigs University Freiburg.

This method of question generation is deeply grounded in the notion of openness as the main quality criterion in qualitative research. The process is designed to help researchers understand their own understanding of the research topic, and to make presuppositions and preconceptions explicit. The detailed and time-consuming refinement of questions throughout the process reflects a deep footing in language: In order to understand *what* is said, we need to analyze *how* something is said (cf. Garfinkel 1967). It is thus necessary to consciously reflect upon the language of the tool that generates our data in the first place.

## Literature

Carvajal, D. (2002). The Artisan' s Tools . Critical Issues When Teaching and Learning CAQDAS. *Forum Qualitative Social Research*, 3(2). Retrieved from <http://www.qualitative-research.net/index.php/fqs/article/view/853>

Di Gregorio (2014): Time for a new pedagogic approach to teaching CAQDAS. Slide Deck from CAQDAS 2014 Conference, Surrey (England). Retrieved from [http://www.surrey.ac.uk/sociology/research/researchcentres/caqdas/files/di\\_Gregorio\\_CAQ14\\_New\\_pedagogic\\_approach.pdf](http://www.surrey.ac.uk/sociology/research/researchcentres/caqdas/files/di_Gregorio_CAQ14_New_pedagogic_approach.pdf)

Garfinkel, H (1967). *Studies in Ethnomethodology*. Englewood Cliffs, NJ: Prentice-Hall.

Helfferich, C. (2005). *Qualität qualitativer Daten. Manual zur Durchführung qualitativer Einzelinterviews*. (2<sup>nd</sup> ed). Wiesbaden: VS-Verlag.

Kruse, J. (2014). *Einführung in rekonstruktive Interviewforschung*. Weinheim: Juventa.

## Using the Curriculum

The modules of this curriculum can flexibly be used in classes or research team meetings. Participants will walk away with a finished product, or at least a product that is in a very advanced alpha stage. This is a time-consuming process. If finishing the guide within the course of a workshop not your main goal (e.g. if you just want to illustrate the process), it is less consequential if modules are omitted or shortened.

Some modules are also good stand-alone sessions: For example, the “Software Intro MAXQDA” can be used as a lecture highlighting the functions & operational logic of QDA software; the module “Brainstorming for Questions” can be used in research teams in order to map out how different team members understand the research project.

Module Title		Time allotted for groups with three different research projects	Time allotted for groups with one research project
Sub-Modules	<ul style="list-style-type: none"><li>• Activities</li></ul>		
Sub-Modules	<ul style="list-style-type: none"><li>• Activities</li></ul>		
Instructor Notes			

## Using different software

This workflow can be easily adapted with other qualitative research software packages than MAXQDA.

## Using no QDA software

QDA Software (or any digital technology for that matter) is not crucial to this workflow; this question generation strategy has been developed as a non-digital work flow by Helfferich and Kruse. If you choose to not use QDA software, I recommend to have participants write each question on a single note card. The sorting/annotation steps can then be done using large tables or whiteboards with spread-out note cards.

## Let me know what you think!

I appreciate feedback and comments on this guide, and I am always interested in a conversation on integrating QDA software into methods teaching. Please feel free to contact me if you have questions, comments, suggestions, or if you want to share additions or adaptations.

## Scenario 1: Every Participant has their own project

The modules are designed for intense group work, and I have tailored the workshop for classes/groups that can be split in sub-groups consisting of three participants. If you choose to increase the group size per working group, the workshop time will increase greatly, because each group will have to work with one additional set of questions. In my experience, a group size of three is ideal: Participants will benefit from working with more than one other person, but the group is not becoming too big (or will split in two sub-groups). In general, I do not recommend exceeding a total class size of 15 participants. **Time durations for this scenario are indicated in blue in the modules.**

### Using the modules as a full workshop

The workshop modules constitute a full 2-day workshop. I suggest to end the first day right before or after “Annotating questions II”, depending on the energy level of the group on the first day.

## Scenario 2: Each group works on the same project

If every group of participants works on the same project, the activity can be conducted in a dramatically smaller amount of time. This will give instructors more flexibility concerning sub-group size: Since all participants in a group work on the same questions, sub-group sizes can vary. I recommend to not exceed a group size of four in workshop/class settings. In research team settings, larger groups are easier to manage, but I do still recommend splitting up large groups (7+), especially when it comes to the brainstorming of questions. If you use the brainstorming phase in a research project, I do recommend brainstorming for at least one hour. **Time durations for this scenario are indicated in green.**

### Using the modules as a full workshop

If every group of participants works on one project, the modules constitute a full 1-day workshop. Due to the somewhat strenuous process of brainstorming and annotating, I find it more effective to split the workshop into two half-day sessions, if possible.

## Prep Notes

- **Real Projects.** Creating questions for interviews is hard. It takes time, stamina, and the participants need to be willing to become aware of and challenge their preconceptions. This workshop and its modules work much better if participants have *real projects* for which they want to create *real questions*. Especially if you choose using the modules as full-day workshops, it is crucial that participants produce guides for real research projects; this will ensure they remain engaged and motivated throughout this demanding process.
- **Bring a research/project question.** If participants are generating questions for their own projects, ask them to bring a brief abstract to the workshop that includes their main research question. I recommend having participants bring abstracts that are up to 400 words long. This will make it easier for their groups to wrap their heads around the projects, and to generate relevant questions. By requiring an abstract, you will also minimize the number of participants who come ill-prepared.
- **Software is installed.** It helps if participants have already the software installed when they come to the workshop. Every participant should bring their own laptop. Demos for MAXQDA can be downloaded from their website at no cost (<http://www.maxqda.com/downloads>). Virtually all qualitative data analysis software packages offer free demonstrations that are fully functional for a number of days. Thus, it is not vital that participants purchase the software, and the workshop does not have to be conducted in a computer lab – by design, labs are often ill-equipped for workshops that revolve around collaboration in groups due to their classic ‘classroom’ setup.

MAXQDA demos are fully functional for 30 days. Thus, you should let participants know that they should install the software *just prior to the workshop*. Please note that the screenshots in this guide are from the Windows version of MAXQDA. The Mac version is very similar to the Windows version, but there are differences in detail. If you expect Mac-users in your group, familiarize yourself with both versions.

- **Timing is important.** Bring a stop watch to the workshop. The activities in this guide are timed, and it is essential that the instructor keeps the groups on track throughout the process.

## Workshop Modules

Intro		45	45
<b>Introduction &amp; Setup</b>	<ul style="list-style-type: none"><li>• Allow time to install the software &amp; resolve tech issues</li><li>• Use assorted candy or playing cards for creating random groups of three participants</li></ul>	15	15
<b>Mini-Lecture</b>	<ul style="list-style-type: none"><li>• Questions &amp; Quality of Questions</li><li>• Introduce question generating strategy</li><li>• Introduce question guide design</li></ul>	30	30
See appendix A for instructor notes.			



Software Intro MAXQDA		45	45
<b>Guided tour through basic functions</b>	<ul style="list-style-type: none"><li>• Import of texts</li><li>• Editing text</li><li>• Coding text</li><li>• Memos (in-text and code memos)</li></ul>	45	45
See appendix B for instructor notes around software.			

Brainstorming for Questions		100	20
<b>Project introduction</b>	Each participant is given 10 minutes to introduce their project to the others in their group.	30	n/a
At this point, it is very helpful if participants have brought a short outline of a proposal (400 words) including their research question. This allows the participants to use the limited time effectively.			
<b>Brainstorming questions</b>	<p>The groups brainstorm and write down questions for each participant in the group for 20 minutes. They write their questions directly into a newly created document in MAXQDA.</p> <ul style="list-style-type: none"> <li>• After each 20 minute session, there will be a 5 minute break.</li> <li>• Within each 20 minute phase, the instructor indicates when 7 and 14 minutes have passed. At these points, the person who is typing up the questions changes. This way the burden of typing is distributed (the typist is limited in thinking about questions while the rest of the team is 'firing' questions).</li> </ul>	70	20
<p>Make sure to repeatedly point out that participants are only to generate questions. Intervene if participants talk <i>about</i> questions. Also, encourage participants to write down every question separately, even if just one word differs. Even if one word is different: it is a different question, and a breadth of questions a good source for working on wording options later, and to discuss how small details in questions can have big impacts on what the questions evoke.</p> <p>Point out that participants should not overthink their questions, or self-censor. There are no bad, or stupid questions. This is a process of introspection, and it is important to write down every question. The refinement of the questions, and the sorting of the questions is reserved for later parts of the process.</p> <p>In this phase, participants are not allowed to talk about questions, or about the content of the research. The process is designed to <i>communicate about the research in terms of calling out interview questions</i>. This makes this activity a great tool for general reflection, even if no interview questions are necessary in a project. The activity can also be used in order to explore the different understandings of a more general research question amongst team members. The generation of questions serves as a tool for the de-indexicalization of the study question; it is a tool that helps to make explicit the implicit assumptions of the team members.</p> <p>If groups do not come up with more questions, encourage them to find more; it is normal to not come up with more questions at some point; this is a plateau that is important to face and to overcome, because it encourages participants to stretch their horizon, and to depart from their every-day understanding of the topic.</p> <p>For a full-fledged research project, 20 minutes are not enough for this. In a 'real' research scenario, it is advisable to do this activity for at least one hour, better for two hours. In more complex research projects it is also advisable to run several brainstorming phases for different subtopics/research sites/study participant types.</p>			

Annotating Questions (I)		90	30
<b>Review of core software functions</b>	<ul style="list-style-type: none"> <li>• Briefly review the coding functions</li> <li>• Point out how to delete codings</li> <li>• Point out how to move codes in the code browser in order to create sub-codes.</li> </ul>	5	5
<b>Annotation in groups</b>	Participants annotate the lists of questions in groups, for 25 minutes per person. Time a 5-minute break in between the 25 minute chunks. Participants use the coding functions to code their questions according to themes. They use the memoing function in order to take notes about the themes.	85	25
<p>In this phase, the participants work together to find themes in their questions. Participants should think of the themes as the topics that a particular question might evoke. For example, the question “What is volunteering for you?” could evoke themes such as “Reasons for volunteering / History of volunteering / Expectations / Personal rewards? / Stepping stones”. When thinking about themes, think about: What does this question evoke? What could an interviewee possibly answer here? What are topics an interviewee might bring up?</p> <p>Because participants use the coding functions, they can quickly see which questions can evoke more themes.</p> <p>By further breaking down their questions into themes, participants further de-indexicalize their understanding of the topic, and they unveil aspects of their pre-suppositions. The themes can later indicate which topics the researchers expect or presuppose; by looking at the different themes, researchers can also identify gaps in their questions, and think about themes that are not yet covered by any questions.</p>			

<b>Annotating Questions (II)</b>		<b>60</b>	<b>60</b>
<b>Review of core software functions</b>	<ul style="list-style-type: none"> <li>• Show weighting &amp; commenting functions</li> <li>• Show how simple queries can be used to check coding for consistency</li> <li>• Show how to move coded segments from one code to another (code merging)</li> </ul>	20	20
<b>Continue annotation as solo activity</b>	Each person works by themselves on their annotation for 40 additional minutes.	40	40
<p>This module should be extended if participants have created large amounts of questions. Ideally, participants will have annotated all questions before proceeding to the next steps. Thus, this activity is also a good homework task.</p>			

Sorting Questions (I)		60	60
<b>Review of core software functions</b>	<ul style="list-style-type: none"> <li>• Show crosstab functions &amp; intersection searches</li> <li>• Show how to use the Code Relations Browser for checking code overlaps</li> <li>• Show Commenting &amp; Weighting functions</li> </ul>	20	20
<b>Solo Sorting</b>	Each participant uses query functions to identify a set of questions that they could use in their guide. Participants run simple queries for each of their themes in order to compare questions, and identify good questions. They can use coding comments and code weights to sort questions further by openness or other quality criteria.	40	40
<p>Prompt participants to search for questions that cover the most ground (for their intro/lead questions) – i.e. questions with the most themes attached to them. In this phase, participants should also think about which expected themes they deem the most important for their overall research questions.</p> <p>At the end of this module, participants should have made a list of the main themes that are most important to their research question/project.</p>			

Sorting Questions (II)		70	20
<b>Peer discussion &amp; continuation of sorting</b>	Each participant introduces their main themes to their group, and groups discuss & modify the themes. (20 minutes per person, with 5 minutes break in between participants)	70	20
<p>Here, it is specifically helpful to discuss how the themes relate to the research/project question, and which themes might be more important, and less important.</p> <p>Participants can take notes on their team members' input, for example using code memos. This is another chance to speak about the memo functions.</p> <p><b>Alternative/additional steps:</b></p> <p>Participants give examples of their questions which the instructor writes on a white board. All participants engage in a critique of the questions, along the lines of:</p> <ul style="list-style-type: none"> <li>- What does this questions well?</li> <li>- Where/how is this question problematic?</li> <li>- Which presuppositions can we find in the question?</li> <li>- What may be effects on the narrative/answer the question stimulates?</li> <li>- How could we rephrase this question? To what effects?</li> </ul> <p>This activity can be inserted (and repeated) at any time after the participants have brainstormed questions. If your groups contain mainly less experienced researchers, it is advisable to add this activity, so that participants gain a deeper understanding on how to think about/evaluate their interview questions.</p> <p>Keep in mind that there are problematic aspects in every question – for example, every question contains presuppositions. The point is not to avoid problematic features – this would be impossible. The idea behind the method lies in consciously, and purposefully engaging with problematic, often paradox issues, phenomena, and situations. The main task lies in weighing the issues against each other – in order to build a useful instrument.</p>			

<b>First Iteration of Guide</b>		<b>120</b>	<b>60</b>
<b>Adding questions into the question guide</b>	Each participant decides on the 'top' questions for each section/topic of their guide, and arranges them within the question guide scheme. Participants should also include expected themes here.	30	30
This is a good homework assignment		Break (5)	
<b>Refining questions</b>	The teams work on refining their questions that they added into their question guides. (25 minutes each, with 5-minute breaks in between participants).	85	25
In this phase, groups should focus on question wording, on getting rid of redundant questions, and on finding alternative entry/lead/stimulus questions.			

<b>Second Iteration of Guide</b>		<b>120</b>	<b>60</b>
<b>Refining questions - solo</b>	Each participant refines their questions further, adding and excluding questions as necessary.	30	30
This is a good homework assignment		Break (5)	
<b>Refining questions – new groups</b>	Participants form new groups of three in order to review the new iterations of their question guides (25 minutes each, with 5-minute breaks in between participants).	85	25
<p><b>Alternative/additional steps (I):</b></p> <p>Participants form dyads, and exchange question guides. They comment on each other’s questions (20 minutes), and then discuss their suggestions (10 minutes each).</p> <p><b>Alternative/additional steps (II):</b></p> <p>Participants give examples of their questions which the instructor writes on a white board. All participants engage in a critique of the questions, along the lines of:</p> <ul style="list-style-type: none"> <li>- What does this questions well?</li> <li>- Where/how is this question problematic?</li> <li>- Which presuppositions can we find in the question?</li> <li>- What may be effects on the narrative/answer the question stimulates?</li> <li>- How could we rephrase this question? To what effects?</li> </ul> <p>This activity can be inserted (and repeated) at any time after the participants have brainstormed questions. If your groups contain mainly less experienced researchers, it is advisable to add this activity, so that participants gain a deeper understanding on how to think about/evaluate their interview questions.</p> <p>Keep in mind that there are problematic aspects in every question – for example, every question contains presuppositions. The point is not to avoid problematic features – this would be impossible. The idea behind the method lies in consciously, and purposefully engaging with problematic, often paradox issues, phenomena, and situations. The main task lies in weighing the issues against each other – in order to build a useful instrument.</p>			



Third Iteration of Guide & Outro		60	30
<b>Finalization of Question guides</b>	Each participant works on finishing the product.	30	n/a
<b>Evaluation &amp; Feedback</b>	Participants form new groups of three in order to review the new iterations of their question guides (25 minutes each, with 5-minute breaks in between participants).	30	30

## Appendix

### A: Notes on Mini-Lecture

#### Questions & Quality of Questions

Researchers, especially when less experienced, can feel a desire to ask too many questions, and questions that are too closed. While the openness of a question depends on the instrument design (e.g. along the line of the axis open interview – semi-structured interview – structured interview), I believe that it is always valuable to ‘push’ against the desire to ask too specific questions, and the desire to ask too many questions. I introduce participants to a series of aspects of “good” questions, while pointing out that every question has problematic features. In general, I like to point out the following aspects:

- Ask for processes rather than justifications
- Brief questions are better than long, cascading questions
- Interview questions, as instruments, are typically spoken. They should be designed in spoken language
- Try to elicit narrations and descriptions
- Try to point your questions at experiences, or concrete issues (instead of hypotheticals)
- Prefer conceptually open questions over suggestive questions

#### Question Generation Strategy

Here, I briefly refer to the 5-step workflow for question guide design based on Kruse (2014).

##### 1. Brainstorming

- Write down every question.
- There are no silly, inappropriate questions!
- DO NOT review or discuss questions! If you have a modification, or something to add, formulate it as a question.
- DO NOT discuss what kind of questions one should ask. Instead, formulate a question, and share it.
- It’s OK if there is silence. Keep thinking, there are always more questions!

##### 2. Sort questions by topics & themes

- Write down the topics each question could cover or evoke. What might a person say as an answer to this question? Which topics or themes might this person cover in their answer?
- Arrange the questions by main topics.

##### 3. Review/Sort your questions

- Are there questions that are more open and more closed?
- Are there questions that cover more topics than others?
- Are there duplicates?
- Are there problematic questions?
- How could questions be improved?

##### 4. Identify ‘top’ questions

- Omit inappropriate questions and duplicates, identify top questions

### 5. Put questions into guide

- Write your questions into the question guide
- Go back to step (2) or (3) to review and refine your questions

### Guide Layout

This guide layout is based on Helfferich’s (2005) suggestions in her work on the quality of qualitative data. Each guide consists of several rows. Each row contains one main question/stimulus. A guide for a semi-structured interview should be boiled down to 3-5 stimulus questions. Each stimulus is supplemented with an alternative question. This alternative question is basically a re-phrased stimulus that can be used if the first stimulus does not ‘catch’ (e.g. if participants ask “I’m not sure what you mean”). Using an alternative question prevents researchers from directly asking very focused, conceptually more loaded questions.

Each stimulus is supplemented with a list of expected themes – i.e. themes that the interviewer expects to be covered by this question. The themes are developed in the process of developing the questions.

Each expected theme corresponds with a follow-up or probe (a probe/follow-up can also cover more than one theme). This way the interviewer can ensure that he/she has a question ready for each of the important expected themes. In the last column, interviewers note questions that can extend narrations. Experienced interviewers may not need this last column, but it is a helpful reminder for novices to have these questions in hand in order to pace the interview, and in order to maintain a stance of openness in a potentially stressful situation.

The question guide layout allows for non-linear, open interviewing while giving researchers a clear framework that they can access during the interview. For example, if expected themes from stimulus question 2 were already covered in a narration evoked by stimulus 1, then the follow-up for that theme is not necessary. During the interview, researchers can use pens to discretely make dots next to expected themes that were covered. Thus, they can keep track of what follow-ups they still need to address in order to get consistent data that touches on all the points they are interested in.

Stimulus & backup	Expected Themes	Follow-ups & Probes	Stimuli for extended narrations
<b>Can you tell me a little bit about what volunteering is for you?</b>	- Reasons for volunteering - History of volunteering	Can you tell me a bit about your motivation to volunteer?	What else?  Can you tell me more about this?
<b>Based on your experience – what is volunteering?</b>	- Expectations - Personal rewards? - Stepping stones	What did you think about volunteering when you first started?  Can you explain to me what you meant with “personal rewards”?	What else comes to mind when you think about your volunteering?

In general, the guide is designed towards starting with open questions (on the left) towards more closed questions (as you go from left to right). The stimuli should also be ordered in terms of their openness (from more open to more closed stimuli going from the top to the bottom of the guide). In this context, I

suggest to ask for demographic data at the very end of the interview, and not in the beginning. Asking such closed questions, which are also associated with the quantitative research paradigm, may inhibit participants in responding to conceptually open, narrative-evoking questions.

Stimulus & backup	Expected Themes	Follow-ups & Probes	Stimuli for extended narrations
Can you tell me a little bit about your volunteering experience?	- Reasons for	Can you tell me a bit about	What else?
Based on your experience, what is it that motivates you to volunteer?	- Expectations - Personal rewards? - Stepping stones	What did you think about volunteering when you first started?  Can you explain to me what you meant with "personal rewards"?	Can you tell me more about this?  What else comes to mind when you think about your volunteering?

**Funneling from open questions to closed questions**

Using the guide is like using a set of golf clubs: You bring a club for the various situations you expect, but you do not know beforehand if you will even use all golf clubs. You use your stimulus as the driver – from there you see where the ball lands. Sometimes all you need is a driver and a small follow-up (the putter), and sometimes you need to utilize more questions when navigating an interview (the equivalent of ending up in a bunker after a messy drive). In some interviews, one or two stimuli can be enough to stimulate a narration that covers a majority, or even all expected themes.

## B: Intro MAXQDA (45)

In general, I like to start software workshops by having participants import, and annotate food recipes; this brief intro for the Question Generation Workshop is no exception to this. Using recipes in software intro courses has several advantages:

- An intro is designed to give a functional overview, and since recipes are naturally structured, they lend themselves to doing some simple organizational coding & memoing work, and to doing simple work with document variables.
- Recipes are a familiar type of data for virtually all participants; food is something everybody can relate to, and comment on.
- Recipes can be downloaded from different sites and often require formatting work. This is a great opportunity to illustrate that (and how) using software requires data clean-up.
- Recipes can be downloaded by users during workshops, thus they can create a sample database with things they curate, which increases ownership and motivation.
- Talking about recipes helps avoiding a workshop atmosphere in which individuals are intimidated by other participants' display or performance of methodological proficiency.
- Recipes are perceived by most participants as "fun", and help creating a friendly, communicative, informal workshop atmosphere.

This guide gives you a brief outline on what to cover during the intro MAXQDA session. This guide assumes that the instructor has some experience in using MAXQDA and/or other QDA software. Participants can download a demo of MAXQDA from the MAXQDA website (<http://www.maxqda.com/downloads/demo>).

Please also note that this 45-minute intro is a mere taste of the software. The full curriculum is designed to gradually have participants explore more functions as they go, and to gradually, and flexibly introduce more functions throughout the process.

This guide assumes that you will bring three word documents with written recipes to the workshop.

In addition to this intro, or as a primer, you can provide participants with intro videos on the logic and functions of QDA software:

<https://www.youtube.com/watch?v=tLKfaCiHvic>

<http://squaremethodology.com/blog/2015/5/26/three-intro-videos-on-qualitative-data-analysis-software>

### General Function & Layout (5)

- Explain the concept of project files in MAXQDA
- Open a project & explain what users generally do & see in the four windows
  - Document System -> Dataset is organized here
  - Document Browser -> Here you look at data
  - Code System -> Here you organize your codes
  - Retrieved Segments -> Here you find your query results
- Have participants start a new project.
- Explain how to manage the windows in MAXQDA. (I like the 'classic' view, with Document System & Browser on top, and Code System & Retrieved Segments on the bottom. I think it is the most space-effective display on laptop screens (which most participants are likely to experience the software through unless you teach in a lab).
- Point out that it is crucial to work on a big screen and/or on two screens in order to maximize the ease-of-use and efficiency of working with any kind of QDA software.

### Import (10)

- Together with participants...
  - import one recipe
  - rename the recipe
  - delete the recipe
- Have participants import three recipes themselves, and rename them.
- Point out that documents can only be imported if they are not used by other programs at the same time.
- Explain that imported files 'live' in the project file. They are not merely linked.
- Have participants view every recipe in the Document Browser
- This is a good opportunity to talk about the importance of pre-formatting data, and of being conscious of the data format when deciding on a QDA software package.

### Memos & Edit Mode (5)

- Together with participants, create a new document in MAXQDA
  - Show participants how to turn on/off the edit mode
  - Point out that the edit mode is a useful tool, but not a word processor
  - Point out the importance of cleaning up data *before* importing in QDA software
- Show participants how to create a memo in the Document Browser
  - Point out the color tags for memos
  - Point out the calendar stamps in memos
  - Mention that memos can also be attached to whole documents, and to codes, and that all memos in MAXQDA look the same.

### Coding (15)

- Together with participants...
  - create a new code: “Recipe Title”
  - code the recipes’ titles with that code
  - change the color of the code
- Have participants create two other codes: “ingredients” and “instructions”
- Together with participants...
  - create the code “Recipe Structure”
  - move the existing codes into the code “Recipe Structure”
- Have participants code the recipe data in the three recipes. (5)
- Show participants how to delete codings, how to re-code selections, and how to change the length of codings.

### Queries (5)

- Explain the logic of simple queries in MAXQDA via “activation”
- Together with participants, do simple queries based on the work so far.
- Point out how to export retrieved segments
- Point out how to re-contextualize findings via clicking on the yellow data card next to a query finding.

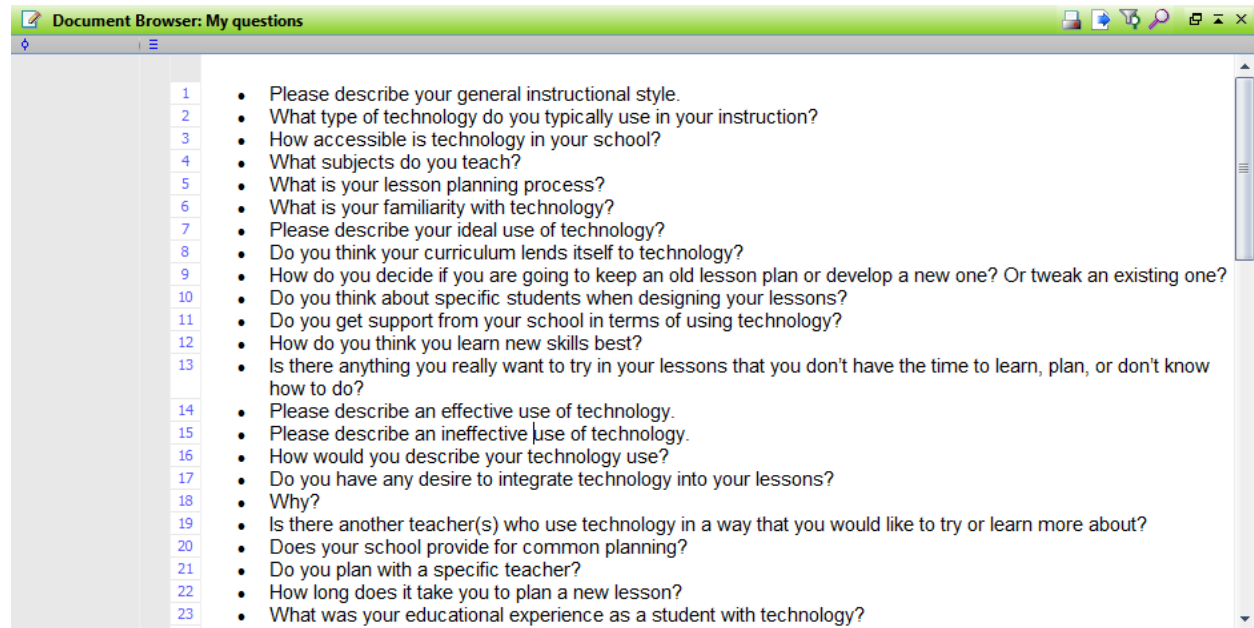
### Questions & Wrap-up (5)

This time can be used for questions, or as buffer.

## Participant Example

This section contains a series of screenshots from a researcher's project file that was generated in one of my workshops. It is meant to illustrate how specific tools in the software were used during the process of generating questions. This is just one way of using functions in the software; users will explore different tools that may be helpful to them in the process of the workshop, and they will use identical software features for very different purposes. I believe that playfully exploring the software plays an integral role when fostering creative, empowering, surprising uses of software in the context of research.

## Brainstorming for Questions



This is a list of questions generated in a word processor, and imported into MAXQDA. Note that each question has its own paragraph number, and that the bullet points were part of the imported data from the word processor.



## Annotation

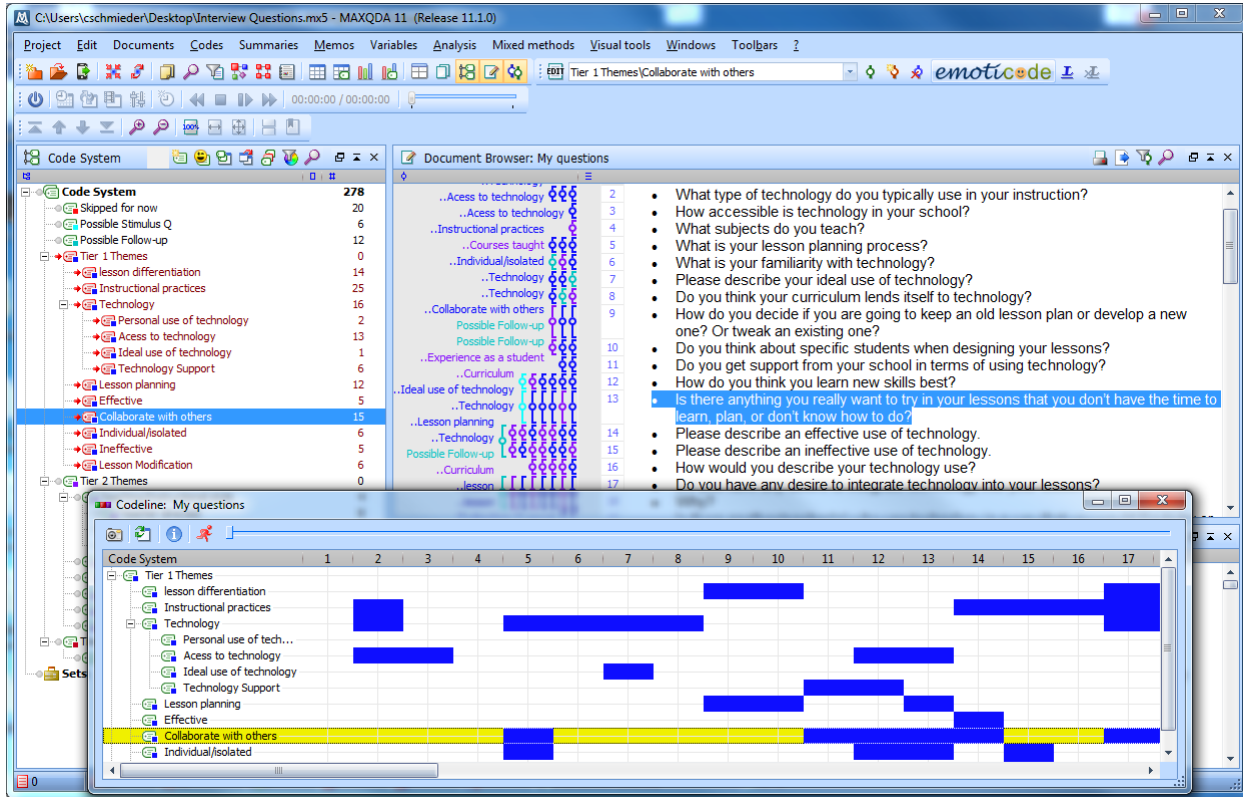
The screenshot displays a software interface for qualitative data analysis. On the left, a 'Code System' tree is visible, showing a hierarchical structure of codes. The 'Tier 2 Themes' section is expanded, listing various themes such as 'Teaching/Instructional style', 'Teacher directed', 'Mixed between TD and SC', 'Student centered', 'Teacher Education/training', 'Learning mathematics', 'Courses taught', 'Curriculum', 'Experience as a student', and 'Learning Space'. Each theme has a numerical count next to it. The main window, titled 'Document Browser: My questions', shows a list of interview questions with corresponding code labels and counts. The questions are numbered 2 through 30. The code labels include terms like 'Access to technology', 'Instructional practices', 'Collaborate with others', 'Lesson planning', 'Curriculum', 'Technology Support', 'Effective', 'Individual/isolated', 'Teacher Education/training', 'Collaborate with others', 'Learning mathematics', and 'Possible Stimulus Q'. The questions are listed on the right side of the interface, with some questions having multiple bullet points.

The question themes were captured in codes. In this case, the themes themselves were grouped in tiers. The researcher used the tiers in order to sort the themes: Tier 1 were themes that were crucial to cover in the interviews, Tier 2 was possible to covered, Tier 3 was not to be covered. Codes were also used to gradually indicate which questions could be used as stimuli, and as follow-ups. A code “skipped for now” was created in order to capture the questions that could not be initially labeled with a theme. This code was reviewed later, and segments coded were allocated to existing codes, or new codes were created.

## Sorting

Based on the annotation/indexing of data via codes, a series of queries can be made in order to identify promising questions, and in order to sorting through the question database.

## Code Line



The code line visualization tool (*Visual Tools-> Code Line*) can be a good tool to find questions that have a high density of themes (i.e. questions that cover potentially 'more ground', open questions). Clicking on a paragraph number in the code line visualization triggers the *Document Browser* to focus and highlight the corresponding question.

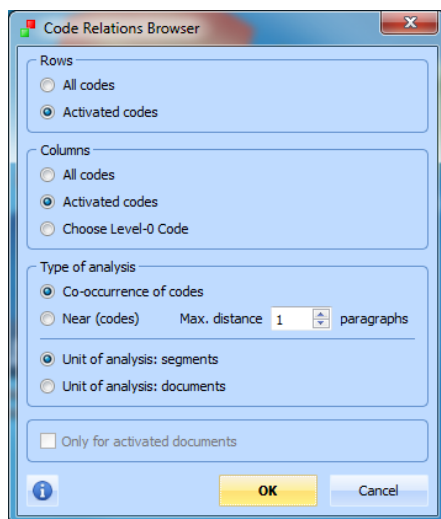
## Code Relations Browser

Code System	lesso...	Instru...	Tech...	Perso...	Acces...	Ideal ...	Tech...	Lesso...	Effective	Colla...	Indivi...	Ineffe...	Lesso...
lesson differentiation		10	5		2		1	7	1	6		1	5
Instructional practices	10		7		6		3	4	3	7	1	3	3
Technology	5	7		1	3	1		2	2	5	1	2	2
Personal use of tech...			1		1								
Access to technology	2	6	3	1			4	3	1	4	2	2	
Ideal use of technology			1										
Technology Support	1	3			4			2		3	1		
Lesson planning	7	4	2		3		2			5	2		2
Effective	1	3	2		1					2		3	
Collaborate with others	6	7	5		4		3	5	2		5	1	2
Individual/isolated		1	1		2		1	2		5		1	
Ineffective	1	3	2		2				3	1	1		
Lesson Modification	5	3	2					2		2			

The Code Relations Browser (*Visual Tools*-> *Code Relations Browser*) allows users to cross-tab the themes they have developed. It shows the intersections of codes. Users can either cross-tabulate all existing codes (above), or selected codes. The numbers show how many intersections the program found based on the user's coding. Double-clicking on one of them triggers a display of the respective questions in the *Retrieved Segments* window.

To only cross-tab selected codes, activate the codes you want to cross-tab, then select the *Code Relations Browser* in the *Visual Tools* menu. In the options menu popping up, users can decide whether the activated codes should be displayed as the columns or as the rows in the table.

In the example below, the researcher activated the codes "Possible Stimulus Q", and "Possible Follow-up". The researcher then selected the *Code Relations Browser* in the *Visual Tools* menu, and chose to have the activated codes be displayed in the rows. Now, questions can be compared: For example, to review all questions that cover "lesson differentiation" that could be possible follow-up questions, you simply need to click on the number that is displayed on the intersection of the respective columns & rows. The questions will then appear in the list of retrieved segments.



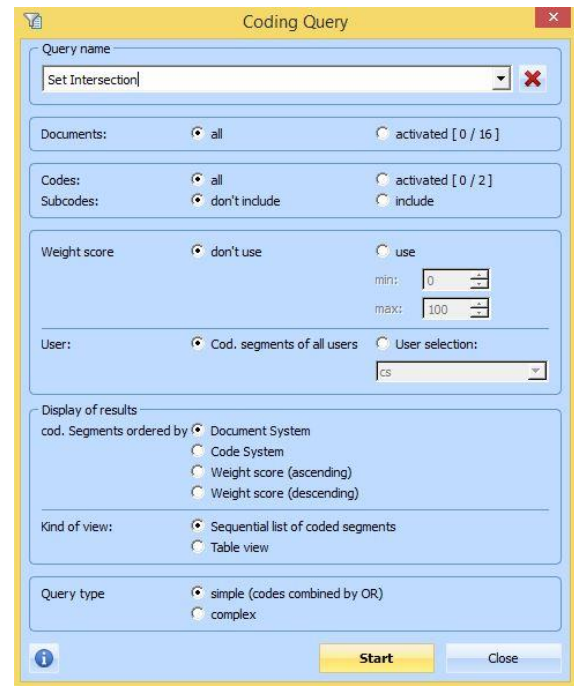
Code System	Possible Stimulus Q	Possible Follow-up
Tier 1 Themes		
lesson differentiation	2	6
Instructional practices	2	8
Technology	1	9
Personal use of tech...		
Access to technology	2	2
Ideal use of technology		1
Technology Support	1	1
Lesson planning	1	3
Effective	1	3
Collaborate with others	3	4
Individual/isolated	2	1
Ineffective		3
Lesson Modification	1	4

## Set Intersections

Another useful tool when reviewing questions is the *Set Intersection*. This advanced query tool allows users to select a series of codes, and to search for intersections. In this case, users can define how many of the selected codes should be overlapping (in contrast, the code relations browser only looks at intersections of two codes). This can be helpful when searching for open questions/questions that promise to cover a broad set of important topics.

First, users need to activate the codes they want to intersect. The complex query can then be accessed via the menu (*Analysis-> Coding Query*).

In the dialog on the left, users select a name for their query (e.g. Set Intersection). Now, they change the *Query Type* on the very bottom to *complex*, and click on *Start*.



In the following window, users can select the option “Intersection (Set)”. They select to include *All activated codes*, and set how many of the codes in the list they require to intersect. The number under *Result* is a preview of how many query results users will see. *Execute* starts the query, and results are displayed in the *Retrieved Segments* window.

