


thinking about and using
be an essential companion
researcher. Using a number
methods. Software provides
practical in-depth research
Qualitative Research provides
a step-by-step guide.

Assisted Qualitative Data
(CAQDAS). Christina Silver
& Ann Lewins. An essential
practice and principles of
to the second edition. 

Using Software in Qualitative Research

Using Software in Qualitative Research a step-by-step guide

Christina Silver
& Ann Lewins

second edition



Los Angeles | London | New Delhi
Singapore | Washington DC



Los Angeles | London | New Delhi
Singapore | Washington DC

SAGE Publications Ltd
1 Oliver's Yard
55 City Road
London EC1Y 1SP

SAGE Publications Inc.
2455 Teller Road
Thousand Oaks, California 91320

SAGE Publications India Pvt Ltd
B 1/I 1 Mohan Cooperative Industrial Area
Mathura Road
New Delhi 110 044

SAGE Publications Asia-Pacific Pte Ltd
3 Church Street
#10-04 Samsung Hub
Singapore 049483

Editor: Katie Metzler
Assistant editor: Lily Mehrbod
Production editor: Ian Antcliff
Copyeditor: Richard Leigh
Proofreader: Clare Weaver
Indexer: David Rudeforth
Marketing manager: Sally Ransom
Cover design: Francis Kenney
Typeset by: C&M Digital (P) Ltd, Chennai, India
Printed and bound by CPI Group (UK) Ltd,
Croydon, CR0 4YY



© Christina Silver and Ann Lewins 2014

First published 2007. Reprinted 2007, 2008, 2009 and 2010

This edition 2014

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act, 1988, this publication may be reproduced, stored or transmitted in any form, or by any means, only with the prior permission in writing of the publishers, or in the case of reprographic reproduction, in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

Library of Congress Control Number: 2013954792

British Library Cataloguing in Publication data

A catalogue record for this book is available from
the British Library

ISBN 978-1-4462-4972-7

ISBN 978-1-4462-4973-4 (pbk)

Contents

<i>List of Figures</i>	xii
<i>List of Tables</i>	xv
<i>List of Boxes</i>	xvii
<i>Companion Website</i>	xxi
<i>Acknowledgements</i>	xxii
<i>About the Authors</i>	xxiii
Introduction	1
Some personal history	1
Our thinking	3
Why a second edition?	4
Chapter overview	5
Chapter exercises and the companion website	7
Our ultimate aim	8
Chapter 1: Qualitative Data Analysis and CAQDAS	9
Qualitative research and data analysis	10
The practicalities of research in the software context	12
Managing and referencing literature	13
Formulating the research problem and defining the research questions	13
Representing theoretical frameworks	14
Incorporating research materials	14
Defining factual features	14
Developing analytical areas of interest	15
Some basic principles and distinctions	16
Analytic <i>processes</i>	16
Levels and directions of work	17
Code-based and non-code-based approaches	18
Cuts through data	19
The rise of qualitative software	20
What types of software do we categorise as CAQDAS?	21
Which is the 'best' CAQDAS package?	22
Analytic strategies in the context of software use	22
Analysis of discourse	24
Narrative inquiry	25
Framework analysis	26
Grounded theory	27
Thematic analysis	29

Mixed methods research	30
Visual analysis	32
Concluding remarks: a critical yet flexible approach	33
Chapter 2: The Nature of Software Support for Research Projects	35
The project management potential of CAQDAS packages	35
Starting points	36
Familiarisation	36
The software project as a container for your work	37
Case-study examples	37
Case study A: Young People's Perceptions	38
Case study B: The Financial Downturn	43
Case study C: Coca-Cola Commercials	44
Qualitative activities and software tools	44
Integration of sources and analyses	46
Exploration of content and structure	48
Organising materials and ideas	51
Grouping	51
Coding	52
Hyperlinking	53
Reflecting upon data, interpretations, processes and results	53
Retrieve, review and rethink data and ideas about them	54
Memo, summarise, track, output	55
Connecting and visualising interpretations	55
Interrogating to identify, compare and test	56
Identifying patterns, relationships and anomalies	56
Comparing subsets and cases	57
Testing theories and assessing quality	58
The right tools for the job	59
Concluding remarks: flexibility in the sequencing of tasks	59
The bits in between	59
Chapter 3: Software Summaries	61
ATLAS.ti	61
Dedoose	64
HyperRESEARCH	66
MAXQDA	68
NVivo	70
QDA Miner	72
Transana	74
Resources	76
Chapter 4: Data and their Preparation for CAQDAS Packages	79
Data types	79
File formats	81
Textual formats	82

Multimedia formats	83
Quantitative formats	83
Textual data preparation	84
Data structures	84
Units of recognisable context	86
Transcription guidelines for textual data	88
Are special formatting considerations really necessary?	92
Structural coding without auto-coding (no special formatting)	94
Formal transcription conventions	95
Multimedia data preparation	96
Social media	97
Direct or indirect handling	97
Assistance for transcribing – and developing synchronised transcripts	99
Mixed data	100
Descriptive or quantitative data import	101
Pre-coding – survey data import and auto-processing	102
Concluding remarks: laying the groundwork	104
Exercises: data and their preparation	104
Chapter 5: Early Steps in Software: Practical Tasks and Familiarisation	106
The way work can happen	107
Gain familiarity with software by setting up a project	107
Creating the project	107
Transparency	108
Naming and backing-up routines	109
Incorporating research materials	111
Getting the software project and the interface shipshape	113
Project design	115
Early organisational structures for data	115
The virtue of empty places for thinking and growing	119
Creating a framework of memos	121
The first memo	122
The dispersal of notes around the project	123
Ideas for naming memos effectively	123
Overt reflections and reflexivity: thinking out loud; telling the story	124
Memos attached to other entities	125
Standalone memos – as project management devices	126
Scoping the topic area and critiquing the literature	126
Exports from customised literature management tools	127
Optimising tools for literature management	129
Concluding remarks: groundwork for efficient analysis	131
Exercises: getting started	131
Chapter 6: Exploration and Data-level Work	134
Early exploration of data	135
Familiarisation during early handling	136

Marking data for relevance and significance	137
Simple data reduction devices and workarounds	138
Annotation tools – their universal utility	139
Multimedia data: annotations and data reduction	142
Annotating data – to aid continuity, reflexivity and openness	145
Quick content searching tools	146
Word frequency tools in CAQDAS packages	147
Text or lexical searching – the practicalities	148
Text-mining tools and complex pattern searching	150
Hyperlinking	152
Practical aspects of hyperlinking	154
Concluding remarks: appropriate use of data-level tools	155
Exercises: exploration and data-level work	156
 Chapter 7: Qualitative Coding in Software: Principles and Processes	 158
What is qualitative coding?	158
How coding works in qualitative software	158
Approaches to coding	160
Induction, deduction, abduction: logics of reaching explanations	160
Coding terminology	161
Inductive approaches to coding	162
Deductive approaches to coding	166
Theoretical coding	168
Question-based coding	168
Combining approaches: the practice of abductive coding	
strategies using software	170
The flexibility of combining approaches	174
Coding visual data: ‘indirect’ and ‘direct’ approaches	175
Coding visual data ‘indirectly’ via synchronised transcripts	175
Coding visual data ‘directly’, without an associated transcript	177
Coding in software, whatever the approach	180
Bases for generating codes	181
Concluding remarks: using software to support <i>your</i> approach to coding	182
Chapter exercises	182
 Chapter 8: Basic Retrieval of Coded Data	 186
Principles of basic retrieval	187
Purposes of basic retrieval	188
Aiding continuity: where did I get to last time?	188
Aiding continuity: generating snapshots of coding status	188
Moving the analysis on: identifying areas for further consideration	189
Moving the analysis on: recoding	191
Moving the analysis on: comparing coding	192
Types of basic retrieval	193
Quantitative overviews	193
Horizontal cuts	195

Vertical cuts	198
Simple filtering devices for early comparative interrogations	200
Generating output	200
Concluding remarks: reflexivity and rigour	202
Chapter exercises	203
Chapter 9: Working with Coding Schemes	205
Breaking down data, building them back together	206
Structures of coding schemes in software	207
Functioning and implications of hierarchy	208
The behaviour of hierarchical coding schemes	208
Non-hierarchical systems	210
What type of coding scheme will suit the way you work?	210
Creating coding schemes	211
Project-related factors influencing the development of coding schemes	212
Escaping the confines of coding scheme structures	214
Separating areas of the coding scheme for pragmatic or theoretical reasons	214
The relationship between the coding scheme and the theoretical framework	215
Better ways to express and collate theory – mapping and short-cut groupings	219
Coding scheme maintenance – routine actions	221
All codes, all data	223
Concluding remarks: manipulating coding schemes for your needs	225
Exercises: managing and manipulating coding schema structures	226
Chapter 10: Managing Processes and Interpretations by Writing	229
The importance of writing in analysis	230
Writing as a continuous analytic process	230
Forms, purposes and spaces for writing	231
Appraisals	231
Field notes	232
Transcriptions	233
Annotations	233
Definitions	235
Analytic memos	236
Process memos	238
Summaries	240
Final write-ups	240
Considerations when writing in software	241
Managing your writing	244
Creating, naming and dating	244
Grouping memos	246
Structuring writing	248

Integrating your writing with the rest of your work	248
Linking writing	249
Visualising memos	249
Coding your own writing	251
Searching the content of your notes	251
Outputting writing	252
Concluding remarks: integrating writing	253
Exercises: managing processes and interpretations	254
Chapter 11: Mapping Ideas and Linking Concepts	257
Mapping traditions and other software	258
Other types of 'mapping'	259
Purposes of mapping in CAQDAS packages	259
Mapping to express theoretical connections	261
General mapping functionality in CAQDAS packages	262
Software-specific functions and specialities	264
Remembered vs. scribbled links	264
Working at the data level within maps (ATLAS.ti and MAXQDA)	266
Creating, hiding and revealing layers in maps (MAXQDA and NVivo)	267
Visualising co-occurring codes in maps (ATLAS.ti and MAXQDA)	270
Creating codes (and other project items) in a map (ATLAS.ti and MAXQDA)	271
'Intelligent' links and functional relationships in maps	273
Concluding remarks: extensive possibilities for mapping	275
Exercises: mapping ideas and linking concepts	276
Chapter 12: Organising Data by Known Characteristics	278
The importance of good organisation in reflecting project design	279
The earliest basics of organisation – and the limits	279
Timing: when to put more complex organisational structures in place	282
Illustrating the potential at the interrogation stage	283
What does a data file consist of?	284
Circumstances, conditions, contexts, cases	286
The evolution of data organisation	286
Imperfect categories	288
Case studies	289
Organising whole documents in software	290
Organising at document level – step-by-step advice	292
Organising at document level – by importing a spreadsheet (or survey)	293
Starting a table off in the right format?	294
Organising within the document (parts of documents)	294
Coding in step-by-step ways	297

Auto-coding	297
The implications of coding cases, respondents and parts of files in terms of their further organisation	298
Concluding remarks: potentials and cautions	299
Exercises: organising data by known characteristics	300
Chapter 13: Interrogating the Dataset	303
The role of interrogation in moving on	303
The incremental, iterative and repeatable nature of querying	304
Combining different dimensions of data	305
Test theories and expectations (hunches)	306
Creating signposts for and from queries	307
Identify patterns and relationships	309
Compare subsets, cases and interpretations	311
Quality control	314
Quality: queries improving interpretive processes	314
Quality: flag up problems and check work	315
Software tools for interrogating the database	315
Searching content and/or structure	316
Simple forms of retrieval	317
Readily available information about codes (without building complex queries)	319
Coding queries	320
Qualitative cross-tabulations	322
Visualising results	323
Tables and matrices	323
Charts and graphs	326
Concluding remarks: interrogation functionality in CAQDAS packages	327
Chapter exercises	328
Chapter 14: Convergence, Closeness, Choice	333
Planning for the use of software	333
Convergence of tasks and tools: software as a container for your work	334
Closeness to data: inside software and outside it	335
Changing techniques of data analysis	335
Automation, quantitisation and mixing methods	336
Visual and social media analysis	337
Focused effective use of software	338
<i>References</i>	339
<i>Index</i>	346

List of Figures

1.1	Graphic overview of qualitative research types (Tesch, 1990: 72–73)	23
2.1	Analytic activities and CAQDAS tools	45
4.1	Auto-coding for structured sections (NVivo and ATLAS.ti)	87
4.2	Section retrieval enabling auto-coding (QDA Miner)	95
4.3	Multiple synchronised transcripts for one media file (Transana)	99
4.4	Importing survey data from an MS Excel file (MAXQDA)	102
5.1	Importing research materials and starting projects (Dedoose, QDA Miner, NVivo)	112
5.2	Folders for storing project data (MAXQDA and NVivo)	116
5.3	Optimising the user interface (ATLAS.ti and MAXQDA)	119
5.4	The Report Manager for gathering materials (QDA Miner)	121
5.5	Interactivity of memos/annotations at data level (ATLAS.ti and MAXQDA)	125
5.6	Integrating literature (ATLAS.ti and NVivo)	128
5.7	The potential of codes as collection devices for tracking ideas from literature through a study	129
6.1	Data reduction: list of ‘quotations’ created for later attention (ATLAS.ti)	137
6.2	Textual annotation of video (Transana and NVivo)	142
6.3	Snapshotting and visual annotation (Transana)	144
6.4	Content searching (MAXQDA and HyperRESEARCH)	148
6.5	Contextual visualisations of text search finds (NVivo and QDA Miner)	149
6.6	The Phrase Finder (QDA Miner/WordStat)	151
6.7	Pattern Recognition Tool: Query by Example (QDA Miner)	153
7.1	Principles of coding processes and code margins (MAXQDA and NVivo)	159
7.2	Early code creation processes (Qualrus and Transana)	162
7.3	Inductively recoding descriptive or broad-brush codes in context (ATLAS.ti)	171
7.4	Synchronised playback of a single transcript (ATLAS.ti)	175
7.5	Direct coding of video (ATLAS.ti)	178
7.6	Visual annotation and coding of still images (Transana)	179
8.1	Code frequency information (QDA Miner and NVivo)	194
8.2	Codes by document and descriptor (variables) (Dedoose)	194
8.3	Horizontal retrieval of coded segments – two interactive views (MAXQDA)	196

8.4	Keyword sequence map – sequential comparison of coding (Transana)	199
8.5	Various reports: getting retrieval information out of the software (ATLAS.ti, NVivo and HyperRESEARCH)	201
8.6	Output of coded segments of still images (MAXQDA and QDA Miner)	202
9.1	Differing hierarchical coding schemes in three Case Studies (MAXQDA, NVivo and Transana)	208
9.2	‘Apparently hierarchical’ and ‘apparently non-hierarchical’ coding schemes (Dedoose and ATLAS.ti)	209
9.3	Simple indexing of news media (QDA Miner)	212
9.4	Families (sets) of codes for grouping ‘sensitising concepts’ or ‘analytic directions’ (ATLAS.ti)	219
9.5	Overview of the whole coding scheme and/or the whole project (ATLAS.ti and MAXQDA)	224
9.6	Early rationalisation of inductively generated codes using grouping and linking devices (HyperRESEARCH and Qualrus)	225
9.7	A streamlined codes-to-theory model for qualitative inquiry (Saldaña, 2013: 13)	226
10.1	Integrating memos with other work via linking (ATLAS.ti)	236
10.2	Logbook, Memo Manager and HTML output (MAXQDA)	239
10.3	Framework Matrices (NVivo) and Summary Grids (MAXQDA)	243
10.4	The Smart Publisher for building professional reports (MAXQDA)	250
10.5	Notes Browser and database view (Transana)	245
10.6	Memo Manager – linking and networking (ATLAS.ti)	247
10.7	Use of sets to gather evidence (NVivo)	247
10.8	Output options for memos (Dedoose)	253
11.1	Model showing members of a set with mixed items – codes, memos, documents (NVivo)	258
11.2	Hyperlinking to track a set of narrative associations (ATLAS.ti)	265
11.3	A map divided into layers on the basis of thematic aspects (MAXQDA)	268
11.4	Filtering codes in a model, by colour (NVivo)	269
11.5	Relationship nodes illustrated in a model (NVivo)	274
11.6	‘Intelligent’ links impacting on coding and selections elsewhere (HyperRESEARCH and Qualrus)	274
11.7	Map-based interrogation for co-occurrence (ATLAS.ti and MAXQDA)	275
12.1	Assigning variables to enable different types of analysis in a longitudinal project	285

12.2	Factual organisation at the document level (ATLAS.ti, NVivo and MAXQDA)	288
12.3	The case card (HyperRESEARCH)	293
12.4	Organising at the level of parts of a document (MAXQDA)	295
13.1	Quantification of keywords (QDA Miner with WordStat)	306
13.2	Incremental and multiple stage queries (HyperRESEARCH and ATLAS.ti)	308
13.3	Querying data for code sequences (QDA Miner)	312
13.4	Activating and filtering to short-cut groupings (MAXQDA and ATLAS.ti)	318
13.5	Easy access interrogation using the Chart Selector (Dedoose)	319
13.6	Examples of query operators and bases of data segment retrieval (MAXQDA)	321
13.7	Coding matrix query, tables, charts and qualitative data (NVivo)	323
13.8	Interactive tables for multiple aspects of interrogation (MAXQDA)	324
13.9	Portraits and comparative profiles (MAXQDA)	327

List of Tables

1.1	Common tasks of analysis supported by CAQDAS packages	9
2.1	Suggested processes for case-study example projects: An Overview	39
3.1	General resources for qualitative methodology and technology	76
3.2	Additional software applications for handling qualitative data	77
4.1	Types of data	81
4.2	Data and their preparation for CAQDAS packages: some suggestions in the context of case study A, Young People's Perceptions	83
4.3	Data and their preparation for CAQDAS packages: some suggestions in the context of case study B, The Financial Downturn	88
4.4	Data and their preparation for CAQDAS packages: some suggestions in the context of case study C, Coca-Cola Commercials	94
4.5	Text formats, auto-codeable structures and pre-coding facilities	100
5.1	Early steps in software: some suggestions in the context of case study A, Young People's Perceptions	111
5.2	Early steps in software: some suggestions in the context of case study B, The Financial Downturn	114
5.3	Early steps in software: some suggestions in the context of case study C, Coca-Cola Commercials	122
6.1	Exploration and data-level work: some suggestions in the context of case study A, Young People's Perceptions	135
6.2	Exploration and data-level work: some suggestions in the context of case study B, The Financial Downturn	141
6.3	Exploration and data-level work: some suggestions in the context of case study C, Coca-Cola Commercials	145
7.1	Coding in software: some suggestions in the context of case study A, Young People's Perceptions	163
7.2	Coding in software: some suggestions in the context of case study B, The Financial Downturn	169
7.3	Code development in a 'theory-informed' abductive approach (Case Study B, The Financial Downturn)	173
7.4	Coding in software: some suggestions in the context of case study C, Coca-Cola Commercials	177
8.1	Basic forms of retrieval: some suggestions in the context of case study A, Young People's Perceptions	191
8.2	Basic forms of retrieval: some suggestions in the context of case study B, The Financial Downturn	197

8.3	Basic forms of retrieval: some suggestions in the context of case study C, Coca-Cola Commercials	200
9.1	Coding scheme tasks	206
9.2	Working with coding schemes: some suggestions in the context of case study A, Young People's Perceptions	210
9.3	Working with coding schemes: some suggestions in the context of case study B, The Financial Downturn	213
9.4	Working with coding schemes: some suggestions in the context of case study C, Coca-Cola Commercials	223
10.1	Types of, spaces for and levels of writing in software	232
10.2	Managing processes and interpretations: some suggestions in the context of case study A, Young People's Perceptions	234
10.3	Managing processes and interpretations: some suggestions in the context of case study B, The Financial Downturn	243
10.4	Managing processes and interpretations: some suggestions in the context of case study C, Coca-Cola Commercials	250
11.1	Mapping ideas and linking concepts: some suggestions in the context of case study A, Young People's Perceptions	260
11.2	Mapping ideas and linking concepts: some suggestions in the context of case study B, The Financial Downturn	264
11.3	Mapping ideas and linking concepts: some suggestions in the context of case study C, Coca-Cola Commercials	270
12.1	Organising data by known characteristics: some suggestions in the context of case study A, Young People's Perceptions	280
12.2	Devices for organising factual features of data and respondents	281
12.3	Organising data by known characteristics: some suggestions in the context of case study B, The Financial Downturn	284
12.4	Organising data by known characteristics: some suggestions in the context of case study C, Coca-Cola Commercials	289
12.5	The way software usually handles the formalised 'case' element of organisation in case-based research (HyperRESEARCH excepted)	290
13.1	Interrogating the dataset: some suggestions in the context of case study A, Young People's Perceptions	305
13.2	Interrogating the dataset: some suggestions in the context of case study B, The Financial Downturn	309
13.3	Interrogating the dataset: some suggestions in the context of case study C, Coca-Cola Commercials	313

List of Boxes

Analytic notes boxes contain reflections on broad aspects relating to the nature of analysis. *Case notes* boxes provide case-study examples. *Functionality notes* boxes contain comments relating to specific analytic tasks enabled by software.

1.1	ANALYTIC NOTES: Research design and software project set-up	12
1.2	FUNCTIONALITY NOTES: Software tools for language-oriented approaches	25
1.3	FUNCTIONALITY NOTES: Software tools for Framework analysis	27
1.4	ANALYTIC NOTES: Features of grounded theory (1967)	28
1.5	FUNCTIONALITY NOTES: Software tools for theory-building approaches	29
1.6	FUNCTIONALITY NOTES: Use of software for mixed methods	31
1.7	FUNCTIONALITY NOTES: Software tools for visual analysis	33
2.1	Case Study A, Young People's Perceptions: Broad research questions	42
2.2	Case Study B, The Financial Downturn: Broad research questions	43
2.3	Case Study C, Coca-Cola Commercials: Broad research questions	44
2.4	CASE NOTES: Integration of sources and analyses	47
2.5	CASE NOTES: Exploring data	50
2.6	CASE NOTES: Organising data	52
2.7	CASE NOTES: Reflecting upon data	56
2.8	CASE NOTES: Interrogating data	57
4.1	FUNCTIONALITY NOTES: Options when working with PDF	82
4.2	FUNCTIONALITY NOTES: Minimal but useful formatting	85
4.3	CASE NOTES: Narrative-driven auto-coding of one respondent's contributions from a focus group transcript (Case Study B, The Financial Downturn)	90
4.4	FUNCTIONALITY NOTES: Auto-coding possibilities relevant to data formatting	92
4.5	CASE NOTES: Planning the use of transcripts (Case Study C, Coca-Cola Commercials)	97
4.6	ANALYTIC NOTES: Questions to ask yourself when preparing data	104
5.1	FUNCTIONALITY NOTES: Planning for software use	108
5.2	FUNCTIONALITY NOTES: Alternative ways of starting off software projects	113
5.3	CASE NOTES: Data storage according to data type (Case Study A, Young People's Perceptions)	116

5.4	CASE NOTES: Theoretical, analytical and practical uses for short-cut groupings in the case-study examples	120
5.5	ANALYTIC NOTES: Flexibility once work is going on in the software	123
5.6	CASE NOTES: Tracking the processes of analysis in a project process journal (Case Study A, Young People's Perceptions)	124
5.7	FUNCTIONALITY NOTES: Ways of integrating literature with your work	127
5.8	FUNCTIONALITY NOTES: Benefits and associated pitfalls of working with literature in CAQDAS packages	129
6.1	CASE NOTES: Literature familiarisation as means of focusing subsequent analytic work (Case Study A, Young People's Perceptions)	138
6.2	FUNCTIONALITY NOTES: Software architecture and flexible working at data level	139
6.3	ANALYTIC NOTES: Debating the need for transcripts of video data	143
6.4	CASE NOTES: Content searching for generic and contextual familiarisation (Case Study B, The Financial Downturn)	146
6.5	FUNCTIONALITY NOTES: Computational support for exploration and coding: pattern matching in QDA Miner and Qualrus	151
6.6	ANALYTIC NOTES: questions to ask yourself when exploring data	155
7.1	ANALYTIC NOTES: Considerations when coding inductively	163
7.2	FUNCTIONALITY NOTES: Software tasks supporting inductive approaches to coding	165
7.3	ANALYTIC NOTES: Questions to ask yourself when coding inductively	166
7.4	CASE NOTES: Considerations when coding deductively (Case Study A, Young People's Perceptions)	167
7.5	FUNCTIONALITY NOTES: Software tasks supporting deductive approaches to coding	168
7.6	ANALYTIC NOTES: Questions to ask yourself when coding visual materials	180
7.7	ANALYTIC NOTES: Limits and cautions when using software coding tools	181
8.1	ANALYTIC NOTES: Purposes and considerations for types of basic retrieval	187
8.2	CASE NOTES: The importance of snapshotting stages of work (Case Study C, Coca-Cola Commercials)	189
8.3	CASE NOTES: Revisiting data to identify areas of further consideration (Case Study B, The Financial Downturn)	190
8.4	ANALYTIC NOTES: Questions to ask yourself when retrieving to help you move forwards	190
8.5	ANALYTIC NOTES: Considerations when developing themes	193
8.6	ANALYTIC NOTES: Questions to ask yourself when retrieving data	195
9.1	ANALYTIC NOTES: Economy when developing codes	216

9.2	FUNCTIONALITY NOTES: Rules of thumb when working with hierarchies in software	216
9.3	ANALYTIC NOTES: Questions to ask yourself when working with a theoretical backdrop	217
9.4	CASE NOTES: Horizontal coding then coding-on to add sub-codes (Case Study A, Young People's Perceptions)	218
9.5	CASE NOTES: Use of sets to think about codes differently (Case Study B, The Financial Downturn)	221
9.6	FUNCTIONALITY NOTES: Tasks helping to refine the coding scheme and move on analytically	222
10.1	FUNCTIONALITY NOTES: Rationalising the use of writing spaces	237
10.2	CASE NOTES: The relationship between memo-writing and visualising ideas in maps (Case Study A, Young People's Perceptions)	238
10.3	ANALYTIC NOTES: Ideas for recording research processes, to aid transparency	239
10.4	CASE NOTES: Organising writing from the early stages of analytic work (Case Study C, Coca-Cola Commercials)	244
10.5	CASE NOTES: The various purposes of short-cut groupings in context (Case Study A, Young People's Perceptions)	246
10.6	ANALYTIC NOTES: Considerations when writing up the results of visual analysis	252
10.7	ANALYTIC NOTES: Questions to ask yourself when writing	253
11.1	CASE NOTES: Mapping grand theories and data-specific interpretations (Case Study A, Young People's Perceptions)	261
11.2	FUNCTIONALITY NOTES: General mapping functions common to ATLAS.ti, MAXQDA and NVivo	262
11.3	ANALYTIC NOTES: Suggestions for working at the data level within ATLAS.ti networks	266
11.4	ANALYTIC NOTES: Bases for retrieving co-occurring codes in maps and networks	271
11.5	CASE NOTES: Using maps to step back and take stock (Case Study B, The Financial Downturn)	272
11.6	ANALYTIC NOTES: Questions to ask yourself when creating maps	275
12.1	CASE NOTES: Organising individual speakers contributing to different forms of data (Case Study B, The Financial Downturn)	283
12.2	FUNCTIONALITY NOTES: Differentiating between organisational features and 'conceptual' codes	286
12.3	CASE NOTES: Evolving factual organisation: splitting variables to handle additional nuances (Case Study B, The Financial Downturn)	287
12.4	CASE NOTES: Using variable-type functionality for recording 'analytic facts' (Case Study A, Young People's Perceptions)	291

12.5	ANALYTIC NOTES: Questions to ask yourself when considering the factual organisation of your project	297
13.1	ANALYTIC NOTES: The value of querying in contributing to 'quality'	304
13.2	CASE NOTES: Being aware of the need for comparison queries (Case Study A, Young People's Perceptions)	307
13.3	CASE NOTES: Being aware of your expectations and the need to test hunches (Case Study C, Coca-Cola Commercials)	307
13.4	CASE NOTES: Incremental or multi-part queries to discover patterns (Case Study B, The Financial Downturn)	310
13.5	CASE NOTES: The quality of interpretive work (Case Study A, Young People's Perceptions)	314
13.6	ANALYTIC NOTES: Combining auto-coding of structures in data with conceptual coding	316
13.7	ANALYTIC NOTES: The importance of framing the questions properly	320
13.8	CASE NOTES: Exploring difference (Case Study A, Young People's Perceptions)	325

Companion Website

Using Software in Qualitative Research: A Step-by-Step Guide second edition is supported by a companion website. Visit <http://www.uk.sagepub.com/silverlewins2e> to take advantage of the learning resources for students and researchers, including:

- Full colour illustrations from the book
- Sample data to accompany case studies in the book
- Step-by-step instructions from software developers for the latest versions of the following packages:
 - ATLAS.ti
 - Dedoose
 - HyperRESEARCH
 - MAXQDA
 - NVivo
 - QDA Miner
 - Transana



Student Resources

1. Illustrations from the Book >
2. Case Studies >
3. Step-by-Step Software Guides >

Using Software in Qualitative Research

A Step-by-Step Guide

Christina Silver and Ann Lewins

A Welcome from the Authors

Welcome to the resources pages for *Using Software in Qualitative Research*, 2nd edition. The book is designed to help researchers get the most out of their use of software, whatever package they are using. The chapters provide an introduction to the principles of using CAQDAS packages and suggestions about procedures for using software in different analytic contexts.

The book will help you to choose the most appropriate package for your needs and get the most out of the software once you are using it. It considers a wide range of tasks and processes in managing and analysing data, showing how software can be used at different stages.

The book very much reflects the way that we think about and teach CAQDAS packages. These webpages supplement the book and our training courses, providing step-by-step instruction for the suggested exercises presented in the book and covered in training. If you need face-to-face training, check out the CAQDAS Networking Project website for scheduled events or contact us via QDA Services for tailored training and/or project-specific support or coaching.

The book presents three case studies with different forms of data (text, video and mixed data) showing how analytic activities process for each project could be supported by software. To find out more about each case-study and download sample data to experiment with click here. The step-by-step instructions for using software have in most cases been written in the context of one of the case study examples.



...be an essential companion researcher. Using a number of methods, Software provides practical in-depth research. Qualitative Research, provides a step-by-step guide. Assisted Qualitative Data (CAQDAS), Christina Silver & Ann Lewins. An essential practice and principles of to the second edition.

Author: Christina Silver and Ann Lewins
Pub Date: May 2014

Buy the book

Order Review Copy

Acknowledgements

We would like to thank the thousands of researchers and students we have met during our work with Computer Assisted Qualitative Data Analysis (CAQDAS) who have helped us to understand what works well in software, what they find easy to use, and what adapts well to their complex range of needs.

We are indebted to the early pioneers of the field and the many friends and colleagues who have contributed to our thinking, including Duncan Branley, Alan Bryman, Sarah Bulloch, Jeanine Evers, Graham Gibbs, Udo Kelle, Matthew Miles, Lyn Richards, Tom Richards, Christine Rivers, John Siedel, Renata Tesch, Eben Weitzman, Nick Woolf, and others mentioned below.

Historically, we have much to thank Ray Lee and Nigel Fielding for, particularly their inspirational creation of the CAQDAS Networking Project, but also for continually enabling us to draw on their experience and methodological expertise.

We thank Katie Meltzer at Sage Publications for her encouragement and feedback, and all who have contributed to the production of this book: for reading and commenting on early drafts, Jennifer Patashnick, whose scrutiny helped us flesh out ideas and avoid errors or omissions but was always provided in the best humour; Merete Watt Boolsen, whose theoretical eye has been invaluable; and Virginia Phillips, whose suggestions on writing style and feedback from a non-expert perspective helped us see the wood from the trees.

We particularly thank Jason Teal and Michael Strong for their patience, red wine, love and common sense. Also our wider families, especially parents (Christopher and Nelleke), children (Nathanael, Magdalena, Gregory and Emma) and grandchildren (Effie, Nell and Fred) with whom we are looking forward to spending more time now this book is finished.

Last, but not least, we would like to thank the ever responsive software developers and their support teams, without whom the companion website would not have been developed. Thomas Muhr, Susanne Frieze and Scientific Software (ATLAS.ti), Eli Lieber and SCRC (Dedoose), Anne Dupuis and Researchware (HyperRESEARCH), Udo Kuckartz, Anne Kuckartz, Stefan Rädiker, Graham Hughes and Verbi (MAXQDA), Normand Peladeau and Provalis Research (QDA Miner), David Woods, Joseph Woods and the University of Wisconsin-Madison Center for Education Research (Transana).

Finally, we owe a debt of gratitude to our fabulous little laptops and ... the comfy chair.

Christina Silver and Ann Lewins
October 2013

About the authors

Christina Silver became engaged with qualitative software from 1997 whilst studying for her MSc in Social Research Methods and shortly afterwards began working with Ann Lewins at the CAQDAS Networking Project. Christina has trained thousands of participants in the theory and practice of qualitative software, taught methods courses at universities across Europe and been involved in numerous qualitative and mixed methods research projects. In 2002 she co-founded with Ann Lewins Qualitative Data Analysis Services (QDAS), which provides bespoke consultancy, coaching and analysis for a range of clients. Since 2010 Christina has managed the CAQDAS Networking Project, leading its capacity-building and training activities and she also co-directs Day Courses in Social Research, a programme of methods courses based in the Department of Sociology at the University of Surrey. She is particularly interested in the relationship between methodology and technology, software-supported visual analysis and the application of qualitative software outside of the academic social sciences.

Ann Lewins was a founding member of the ESRC (UK) funded CAQDAS Networking Project, which she managed until 2010. The project was created to provide a forum for debate around qualitative technology and to support those embarking on the use of qualitative software throughout the wider research community. Ann designed and led a vigorous programme of seminars and training events supporting a range of software applications and gained a unique level of knowledge in the field. Her work helped to establish the standing of the CAQDAS Networking Project as the leading international authority on qualitative software, a reputation that continues to this day.

Between 1994 and 2010 Ann trained thousands of participants at events held at the University of Surrey and many hundreds more at universities and research institutions within UK, Ireland and Europe. Recently she has specialised in supporting teams in their use of CAQDAS programs and has written detailed protocols to guide some of the planning and preparation of such research that are available from the website. She has advised many individual and collaborative research projects concerning both the preparation of complex datasets and the building of creative solutions to the challenges of working with software.

Introduction

The use of technology in qualitative research is not a new idea. Documentary film-making, the use of photographs and other visual documentary evidence are early examples. Subsequently, first analogue and now digital recording devices have been used to capture interactions and discussions. Bespoke technological support for analytic procedures, however, has a less lengthy and more controversial history. In many senses this is curious, given the role of technology in everyday life. The past 10 years, however, has seen a rise in published academic articles mentioning the use of Computer Assisted Qualitative Data Analysis (CAQDAS), a rise in demand for instruction in their use, and an increase in interest in them from those outside the academic social science disciplines within which they originally developed.

This book is the result of a collaboration that started in 1997 when Christina was studying for her MSc and Ann was running the CAQDAS Networking Project (CNP), which began in 1994. Our work together since then has shaped how we think about, use and teach qualitative software, and this book results from those experiences.

In this introduction, we summarise our working relationship, in order to contextualise our thinking about the role of qualitative software, explain why this book is needed, and frame the way that we have written it.

Some personal history

We first met during an overview about qualitative software that Ann gave to social research methods MSc students at the University of Surrey. Ann's engagement with qualitative research and software started during a period of struggle with a part-time history of politics research degree. She was collecting pilot interviews from Labour Party activists when she started her post with the CNP. She felt she had dropped from a great height, simultaneously into the world of social research methodology, technology and the demands of getting a new project off the ground in an innovative methodological area.

The CNP was funded by the UK Economic and Social Research Council (ESRC) but even so survived from the beginning on a shoestring budget which went up and down (mostly down) on seven successive hard-fought-for grants. Ann managed the CNP from 1994 to 2010. During that time she helped shape the project's role whilst great changes were occurring in researchers' expectations and needs of qualitative software and support in its use. The project was the brainchild of Nigel Fielding

(University of Surrey) and Ray Lee (Royal Holloway University of London) who, together with Ann, established its international reputation. Some of the packages, mostly begun as academic exercises amongst social researchers, gradually became commercial enterprises. Ann raised awareness about less well-known packages and delivered training for those establishing a significant place in the research world. The project was committed to balancing its main priorities – absolute absence of any commercial connection to any one piece of software, answering an increasing demand for training, and creating a forum for debate and a space where methodologists and researchers could meet developers face-to-face.

Christina had conducted qualitative analysis for her undergraduate dissertation using manual methods: cutting up photocopied printouts of interviews, marking them using coloured highlighter pens and sticking bits of cut-up transcripts on the bedroom wall in a matrix of themes (represented as columns) by respondents (as rows). Following Ann's overview, Christina, intrigued by the idea that there might be an easier way to manage quantities of data, asked Ann for a free copy of software to use for her MSc dissertation. Ann said no! At that time the department only had licences for software in the computer lab where training courses were run. Four packages were becoming prominent in the UK at that time – ATLAS.ti, WinMax (later MAXQDA), NUD*IST (later NVivo) and The Ethnograph. The picture was slightly different elsewhere – for example, in the Netherlands the locally developed Kwalitan was well used. Already in 1997 the CNP had become a national resource centre for qualitative software, with a website, created and maintained by Ann, providing information about software developments and forums for discussion between users and developers.

Ann's recommendation to the poverty-stricken student was to download a free trial version. NUD*IST fitted the bill as the trial version was fully functional (the only restriction being that you could not save your work). Christina duly transcribed her interviews and over a bank holiday weekend, set about importing them and began coding. Work started off very broadly, printing out bodies of initially coded data and working on those printouts with highlighter pens and scissors. The dissertation was on a knife edge! At any moment there could have been a power cut, a computer or software crash.

Ann was helpful and encouraging, showing Christina how to get as much done as possible within time constraints and concerns that her computer might overheat if left on for days on end. After working on exported hard-copy code reports and giving the computer a rest, Christina reimported them as new data files, recoded and interrogated them. It was thus a quasi-computer-assisted approach, but it got the job done and the degree was obtained. Later that year Christina started a PhD, and from 1998 began assisting Ann with software training courses, as the interest in qualitative software and thus participant numbers were rapidly increasing. That was 17 years ago. Christina is now running the CNP, since Ann retired from the university in 2010.

Via the CNP and our consultancy, QDA Services, we have trained more than 12,000 researchers in the use of qualitative software. We have taught on postgraduate methods programmes, consulted on the technical and methodological aspects of academic and applied research projects, conducted analysis on qualitative and mixed methods datasets, and troubleshooted for hundreds of researchers struggling with

technical and conceptual aspects of using software. We have used different products in our own research. We have discussed, deliberated and disagreed amongst ourselves and with methodologists, students, researchers and software developers. We have, above all else, learnt from each other and those who have attended our training events and seminars.

Our thinking

When we teach, no two sessions are the same because participants arrive with different expectations and needs. Some are very methodologically oriented or experienced in specific manual analytic procedures. Such participants tend to know exactly what they want from software and either quickly succeed in finding it and leave satisfied, or feel frustrated because the software does not appear to live up to their expectations in some way. Others are much more practical in their thinking. Occasionally this manifests itself in a desire for short-cuts to speed up analysis. These participants often leave disappointed. More usually, participants are looking for systematic ways of managing large or complex sets of data. The virtues of CAQDAS packages in this respect come through quickly. The point for us has always been that software can be manipulated to suit a range of needs, that these needs vary from project to project, researcher to researcher, and change as projects progress. Contexts such as sector, discipline, methodology and computer 'savviness' all play their part in how individual researchers get on with CAQDAS packages. Many can teach themselves how to operate software, but what our workshops additionally do is to present packages in their entirety. Illustrating how to put the elements of software together so that you can exploit their potential fully is where we hope we add real value. Being methodologically aware adds an additional dimension when working in academic contexts. In applied contexts (e.g. public consultations, service evaluations), being clear about the practicality of required outcomes is fundamental to successful software use. PhD students fare better when they embark on software use with clearly specified analytic needs, have supportive and technologically knowledgeable supervisors and when they keep detailed notes about their software use so they can adequately justify their processes later on.

Using software is a creative process since qualitative software packages – in essence and design – are inherently flexible. There might well be an efficient way of proceeding, but there are multiple pathways to reaching a particular end-point and multiple purposes for doing so. Though key things need to be achieved in a workshop, the real driver for how it flows and the precise route we take are the participants. In the years we spent working and teaching closely together, we acted as a test-bed for ideas and applications for software use. There are challenges in reacting to the individual needs of participants. Finding solutions to a range of problems expanded our own understanding of the potential of software. We got to know developers and, together with many other software users, influenced the ways in which packages improved. Of course we cannot take too much credit for that! Although experienced users' feedback is critical, developers do know a thing or two about the potential of technology and how to make it work for analytic purposes!

The CNP is predicated upon the idea that there is no one 'best' software package. Each has a range of tools. There are general similarities, but each has different emphases and qualities. There are also tools in individual packages that are quite distinctive. We are impressed by some of these special aspects, and we hope this book will illustrate them.

Why a second edition?

The first edition of this book, published in 2007, resulted from a clear need for specific step-by-step support in the use of software. It was Ann's conceptualisation; Christina muscled in on it. The orientation was explicitly and intentionally practical. Many people do not read software manuals and find help menus confusing. Ann wanted to demystify the area in a practical way, and the first edition was firmly framed around that aim. Christina's current focus is to develop the teaching of technology within methodological frameworks. As software and research develop, the dialectic between them changes. So, in general terms, this second edition has been updated to account for some of the major developments that have occurred in the field since 2007. This has resulted in broadening the packages we continually illustrate and refer to. They do not all have the same focus, emphasising and excelling in different aspects of functionality. Some of the special elements within particular packages will appeal to different types of researcher.

The main issue with the first edition, as with most books about software, is that the specific technical information contained within it was quickly overtaken by software developments. We have therefore taken out most of the software-specific and step-by-step instruction from this edition, placing it on a **companion website** that we can keep up to date as new versions are released (www.uk.sagepub.com/silverlewins2e). Some of the software developers have been glad to contribute to the creation of these pages and we thank them for that.

The continued demand for CNP training in the full range of packages since the publication of our first edition is testament to a relative – and continued – failure of academic institutions to embed instruction in the technical aspects of CAQDAS packages into undergraduate and postgraduate curricula. More significantly, the questions still routinely asked at our training sessions indicate there to be a related lack of awareness of the methodological implications of using software to manage data and analyses. Therefore we have expanded this edition to offer more methodological contextualisation, including more detailed attention to visual analysis and mixed methods approaches. This is in response to increasing interest from outside academic social science disciplines and increasing requests for descriptions of how to undertake certain types of analysis using particular software packages. Both these trends speak to the breadth in potential for the use of CAQDAS packages, and the continued need for specific advice about how to use them.

To a certain extent this book is about highlighting excellence, both in the important routine tasks and in special or unique functions. We felt there was a need to point to particular strengths of software for particular research needs. Although it

was not feasible in this book to do so systematically for all methodologies or practical eventualities, we provide suggestions about phases and tasks of analysis and information concerning the general and special provision of tools in these contexts.

Chapter overview

Chapter 1 provides the practical and analytic framework for the book and the way we think about and teach software. Broadly, it introduces qualitative data analysis in the context of CAQDAS use, discussing some of the main types of analysis that can be supported by them. It defines the term ‘CAQDAS’ and the way we understand and use it; discusses the practicalities of research in the software context; provides some historical contextualisation about the rise of qualitative software; and discusses selected analytic strategies in the context of software use. The chapter emphasises a critical yet flexible approach to the use of CAQDAS packages, considering analytic processes, levels of work and cuts through data, which resonate through the remainder of the book.

Chapter 2 outlines the key ways in which CAQDAS packages can support research projects. It illustrates their project management potential and discusses the implications of this for the ways in which tools can be used effectively and efficiently. The focus is a discussion of analytic activities that are common to many approaches (Figure 2.1; p. 45), describing our model for how these may intersect with software tools and the sense in which software projects can reflect research design. The chapter describes the three case studies that the illustrated processes contained throughout the book are based on: Case A, Young People’s Perceptions; Case B, The Financial Downturn; and Case C, Coca-Cola Commercials. The chapter concludes by stressing that choosing the ‘right tools for the job’ depends on the nature of the project, and that software is inherently flexible, such that it always remains the researchers’ decision what, when, why and how to use tools.

Chapter 3 summarises the CAQDAS packages featured in this book: ATLAS.ti, Dedoose, HyperRESEARCH, MAXQDA, NVivo, QDA Miner, and Transana. The descriptions provide a general overview of functionality, focusing on what is distinctive about each. Representatives of most of the software companies have contributed to this chapter, therefore summaries reflect the developmental thinking of each software as well as our assessment of their distinctiveness and particular benefits. In the chapters that follow we make reference to these software packages and their tools where we believe functionality supports an analytic need particularly well. This book is therefore not systematically comparative in its treatment of the packages. The purpose is to provide an overview of what these packages can do generally, but also to regularly focus attention on aspects of individual software that work particularly well for certain analytic tasks. We therefore do not cover every function in every software package. The fact that we have not illustrated a tool in a particular package does not mean that it is not present. You should refer to the product websites for a comprehensive list of software functionality.

Chapter 4 discusses data and their preparation for use in CAQDAS packages. This includes file formats and preparation for textual, multimedia and mixed data. The chapter is rather detailed and much of its content might not be relevant to your particular project, because it covers a whole range of possible data preparation considerations. Therefore, you may not need to read this chapter unless and until you are at the point of needing to prepare data for analysis. With respect to textual data, we include minimal transcription guidelines and make references to optimal, i.e. software specific, guidelines. These often concern inherent structures and units of recognisable context. For multimedia data we discuss early considerations, including distinctions between working ‘directly’ or ‘indirectly’, the use of social media content and assistance for transcribing. In considering mixed data we outline processes for descriptive and quantitative data import, pre-coding and auto-processing functions.

Chapter 5 outlines useful early steps in software, focusing on practical tasks for getting started and tools which are important to become familiar with early on. This includes setting up software projects and putting early organisational structures in place which reflect initial research design, yet maintaining the flexibility to grow ideas and alter placeholders as projects evolve. We introduce memos and other forms of writing and the importance of being transparent and reflexive in all you do.

Chapter 6 discusses exploration and data-level work, focusing on the key tools involved in becoming familiar with data once they are incorporated within a software project. This includes discussion of the universal utility of annotation tools, quick content-searching tools (including text-mining functionality), hyperlinking, and editing and marking textual data for emphasis. The focus is on experimentation such that if they are relevant to your analytic needs, you are able to develop a strategy for using them in the context of your own preferred ways of working, the nature of your data and the needs of your project.

Chapter 7 discusses principles and processes for qualitative coding in software. This involves a broad discussion of different approaches to coding (inductive, deductive, abductive) and possible strategies for using software when adopting code-based approaches. There is a separate section on coding visual data, and the chapter concludes with discussion concerning limits and cautions when using software coding tools.

Chapter 8 covers basic retrieval options for coded data. It discusses principles, purposes and types of simple retrieval, illustrating that a lot can be achieved without recourse to complex query tools. Although we discuss these tasks in a separate chapter, retrieval usually occurs in tandem with coding, as part of an iterative process. We present them separately only because of the constraints of the linear book format. The chapter ends by discussing reflexivity and rigour in the context of code and retrieve functionality.

Chapter 9 is the third chapter specifically covering coding tasks, discussing ways of working with coding schemes. We discuss the processes of breaking down data and building them back together; the differing structures of coding schemes in individual packages and their implications; how to escape the confines of the structures available and how to handle coding schemes that become too ‘large’.

Chapter 10 discusses ways CAQDAS packages facilitate managing processes and interpretations through writing. This is one of the most important chapters in the book. We discuss the importance of writing in analysis, the forms, purposes and spaces for writing in software and considerations when doing so. This includes becoming aware of how you can integrate your writing with other aspects of work, and the benefits of doing so in eventually developing an evidenced and transparent account. We emphasise the flexibility provided by customised qualitative software in this respect.

Chapter 11 focuses on mapping tools where they are available in CAQDAS packages. It follows on from the discussion of managing processes and interpretations through writing as these are inherently related activities. We discuss and illustrate many different starting points and uses for maps, illustrating how they can be manipulated to suit various needs.

Chapter 12 discusses tools for organising data by known characteristics. This includes discussion of socio-demographic characteristics and other factual features in data, including those identified through the processes of analysis. We outline the importance of organisation and how this reflects research design. We discuss ways of organising whole and parts of documents, and illustrate the uses of working in each way.

Chapter 13 reiterates that interrogation happens throughout analysis, sometimes using simple tools in optimal ways (as discussed in Chapter 8). We build on this to outline more complex query tools designed to help you interrogate the dataset in a range of deeper ways. We discuss the role of interrogation in moving on with analysis, prioritising the incremental, iterative and repeatable nature of querying. We discuss ways to test theories, identify patterns and relationships, compare subsets and other interrogative options.

Chapter 14 concludes the book, presenting a brief round-up of our thinking. Entitled 'Convergence, Closeness, Choice', it discusses ways in which software functionality is coming together, but also how individual tools maintain their distinctiveness. We discuss the contentious issue of whether software acts as a barrier between researchers and their tactile relationship with data, or brings them closer to materials; we argue for the latter position. Above all, this chapter reinforces the threads common throughout: namely that you, as the researcher, remain in control of your methodological needs, your analytic strategies and your use of software. You have the choice of whether to use one of these customised tools in the first place; if you do, then which one; and once you have settled on a product, which tools within it to make use of, how, why and when.

CHAPTER EXERCISES AND THE COMPANION WEBSITE

All the task-based chapters (Chapters 4–13) include illustrations from our three case-study examples (described in Chapter 2). At the end of each chapter are exercises which can be followed using data from one of the case-study examples. Sample datasets can be downloaded from the companion website (www.uk.sagepub.com/silverlewins2e). The website includes step-by-step instructions for these exercises for most of the packages that we discuss in detail in the book. In

most cases, these have been developed in partnership with the respective developers, for which we are very grateful. These are updated regularly to take account of functionality changes in new releases. The website also contains a section on teamworking in the context of CAQDAS use, which is not discussed within the book itself.

Our ultimate aim

In illustrating examples of software use in different contexts through three distinct case-study examples we hope to paint a picture of some common aspects of analysis in the context of software tools so that you can draw out ideas about what might be useful in your own particular research. We understand that your choice of software may be limited within the constraints of local provision, but our purpose is to enable ambitious yet secure use of *any* CAQDAS package and the moulding of its functions to your needs, while also adding to your awareness of what other tools work well for particular contexts. We believe that a broad understanding of software packages other than the one you happen to be using will open up your thinking about your own work.

Above all, we see ourselves as ‘facilitators’ rather than ‘instructors’. The way we teach is informed by the belief that you are the expert about your project and your needs. We can show you tools, illustrate their benefits and caution against their potential limitations. We can make suggestions about their suitability (or not) for different approaches to data analysis. But you need to decide whether to use software at all – and if so, then which package. If you decide not to use software then you need to be able to justify this. If you decide to use software, you need to design a strategy for doing so within the parameters of your broader methodological context, specific analytic needs and any practical constraints within which you are working.

We hope this book will provide you with the context you need to frame your thinking about software, to give you insights into the way particular tools might be useful at various moments, and to heighten your reflection about the relationship between technology and methodology. More than anything else, we hope this book will inspire you to explore your data to greater depths, to experiment with software tools and to develop systematic and creative ways of conducting robust and well-evidenced analysis.

1

Qualitative Data Analysis and CAQDAS

This chapter introduces the eclectic field of Computer Assisted Qualitative Data Analysis (CAQDAS) in the context of qualitative research methodology and the techniques of analysis generally. We discuss the practicalities of research in the software context, outline some basic principles and distinctions which resonate throughout the book; discuss software developments, debates and functionality; and discuss selected qualitative approaches. The remaining chapters build from here, describing some core tasks you might undertake using CAQDAS packages, illustrated via three case-study examples (Chapter 2). Our overall emphasis is on the inherent fluidity between the processes involved in analysis and how customised CAQDAS packages reflect and reinforce them.

We discuss analysis in the context of technological possibilities. Table 1.1 lists common analytic tasks enabled by CAQDAS, but software itself does not dictate their sequencing, or whether certain tasks are undertaken or tools are used. These decisions rest entirely with you, informed by the interplay between methodology, analytic strategy, technology and practicality.

Table 1.1 Common tasks of analysis supported by CAQDAS packages

Task	Analytic rationale
Planning and managing your project	Keep together the different aspects of your work. Aid continuity, and build an audit trail. Later, illustrate your process and your rigour through transparent writing.
Writing analytic memos	Manage your developing interpretations by keeping track of ideas as they occur, and building on them as you progress.
Reading, marking and commenting on data	Discover and mark interesting aspects in the data as you see them. Note insights as they strike you, linked to the data that prompted them – enabling retrieval of thoughts together with data.

(Continued)

Table 1.1 (Continued)

Task	Analytic rationale
Searching (for strings, words, phrases etc.)	Explore data according to their content. Discover how content differs across data and considering how familiarising with content helps you understand what is 'going on'.
Developing a coding schema	Manage your ideas about your data by creating and applying codes (that represent themes, concepts, categories etc.). The structure and function of a coding scheme depends on methodology, analytic strategy and style of working.
Coding	Capture what is going on in your data. Bring together similar data according to themes, concepts etc. Generate codes from the data level (inductively) or according to existing ideas (deductively) as necessary; define the meaning and application of codes.
Retrieval of coded segments	Revisit coded data to assess similarity and difference, to consider how coding is helping your analysis, and prioritising 'where to go next'.
Recoding	Recode into broader or narrower themes or categories if appropriate and necessary. Perhaps bring data back together and think about them differently.
Organisation of data	Organise data according to known facts and descriptive features to allow consideration of how these aspects play a role in your understanding.
Hyperlinking	Link data to other data segments and/or to other files to track process, contradiction, association etc.
Searching the database and the coding schema	Test ideas, interrogate subsets for similarity and difference, identify anomalies, or generate another level of coding.
Mapping	Manage analytic processes by visualising connections, relationships, patterns, processes, ideas.
Generating output	Report on different aspects of your progress and the project at any stage. Save as files to capture status at an analytic stage, or to work in other applications. Print off to get away from the computer and think and work in more 'traditional' ways.

Qualitative research and data analysis

Qualitative research is a broad field that crosses disciplinary, methodological and sector-based boundaries, and it is important to acknowledge the variety contained within it. Different philosophical, theoretical and methodological traditions underpin the way researchers think about and do analysis. Much work has been done elsewhere to make sense of these – often competing and sometimes complementary – scientific principles. If you are new to the area we point you in the direction of the following in particular:

Bryman and Burgess (1994), Creswell (1998), Mason (2002), Bernard and Ryan (2010), Silverman (2010, 2011), Bazeley (2013) and Saldaña (2013). Neither the scientific and philosophical principles nor the disciplinary and methodological subtleties within approaches to qualitative research and analysis are the focus of discussion in this book. However, reflecting on your ontological and epistemological standpoints (i.e. how you understand the world to work and how you believe it can be investigated) is important in locating and justifying your research. In reading the literature you will come across many different terms used to define the context and manner of inquiry, including *perspective*, *framework*, *approach*, *strategy*, *methodology*, and *method*. There are no clear boundaries between or hierarchical structure to these terms; they overlap and are used differently in particular contexts. Categorisation of qualitative research in terms of data collection techniques has a long history, but detailed discussion concerning the *processes* and *procedures* involved in analysis (i.e. what we actually do) has only occurred more recently (Bryman and Burgess, 1994). This book discusses processes and procedures of analysis specifically in the context of customised software use.

The extent of diversity in the field is well illustrated by comparing the work of three authors, all of whom wrote during the 1990s yet conceptualised qualitative research rather differently. Tesch (1990) distinguished 27 *forms* of qualitative research (see Figure 1.1; p. 23). Woolcott (1994) differentiated qualitative research *strategies* according to six styles of collecting data (archival strategies, interview strategies, non-participant observation strategies, participant observation strategies, field study, ethnography). Miles and Huberman (1994: 7) argue that while a 'core' of recurring features exist across qualitative research, they are 'configured and used differently in any particular research tradition'. They distinguish between three traditions: *interpretivism* (including phenomenology, social interactionism, semiotics, deconstructionism, ethnomethodology and hermeneutics); *social anthropology* (including ethnography, life history, grounded theory, ecological psychology, narrative studies and case-study analysis; and *collaborative social research* (action research).

The range of ways used to describe qualitative research and analysis illustrates the difficulty of adequately reflecting the diversity in how general principles intersect to result in specific strategies. Most authors concede there to be much overlap between the distinctions they draw; there is often even blurring between understandings amongst different authors using the same terms. Researchers combine data collection methods in qualitative research design and borrow elements from various approaches in developing specific strategies for investigating new social problems or for using different forms of data.

Problems in categorisation systems are illustrated particularly clearly in contemporary writings about mixed methods. As more authors enter the debate, a tendency to generate increasingly specific categorical systems to reflect diversity ensues. Increasingly subtle differentiations complexify to such a degree that the area can become more difficult to access for novice researchers. Nevertheless, broad overviews and summaries are important in gaining entry to any field of scientific inquiry.

The use of customised software is not *required* in order to conduct robust analysis. But its use enables us to be more transparent in how we go about analysis because

the tasks we engage in, their sequence, role and documentation can be more easily illustrated than when working manually.

The practicalities of research in the software context

The availability of customised qualitative software occurred within a diverse methodological field, which has only become more varied with digital technology, big data and the rise of applied, commercial and citizen-research. In addition there is increasing discussion of mixed methods approaches to research and analysis and the use of visual methods. Reflection about of the rise of qualitative software and the implications of its use must be done in the context of the practicalities of research, in which analysis is understood as a core activity throughout an iterative process.

Whatever the characteristics of a particular study, there are certain core elements involved in doing research. Planning is paramount (Box 1.1). Authors usually discuss several aspects in planning and conducting research. Mason (2002), for example, discusses ‘questions of strategy’, ‘generating qualitative data’, and ‘analysing qualitative data’; Boolsen (2006) distinguishes between ‘problem formulation’, ‘research design’, ‘data collection’ and ‘analysis’. In our experience researchers often plan data collection carefully, but neglect to put the same degree of effort into planning the analysis.

In the context of the use of software, much less has been written about research design than in relation to qualitative (and, increasingly, mixed methods) approaches more generally. Di Gregorio and Davidson (2008) wrote the first comprehensive discussion of research design in the specific context of software use that transcends individual products. In further opening up discussion about the role of software in designing and conducting research, we identify six key tasks in setting up a software project to reflect initial research design (Silver and Lewins, 2014). These tasks reflect the sense in which CAQDAS packages are essentially project management tools which can be used from the earliest moments of conceiving a research idea, through all the phases of planning and implementation of analysis to the tasks of writing up an account for publication, preparing for a conference presentation or organising a thesis (Chapter 2).

BOX 1.1

ANALYTIC NOTES

Research design and software project set-up

Designing a research project is all about planning how you intend to carry out the research. What methods of data collection or generation will you employ? Why? And what will be the implications of doing so? What restrictions are there on the way you will proceed, arising

from the circumstances in which you will work? What are the likely consequences of your design choices? What is your analytic focus likely to be? How will you handle changes in focus? These sorts of questions should guide the way you set up a software project. Although it is common to plan research, often this is done primarily in relation to data collection. It is just as important, however, to plan the analysis. Using software from the outset will help with integrating all types of planning into your work.

These tasks are: (i) managing and referencing literature; (ii) defining research topic and questions; (iii) representing theoretical frameworks; (iv) incorporating research materials; (v) defining factual features; (vi) developing analytical areas of interest. They are inherently interrelated, occurring in tandem rather than as discrete stages. One of the main benefits of using qualitative software is that flexibility can be built into analytic designs to reflect changes as projects evolve. This is a common thread through this book. This way of thinking about setting up a software project emphasises the importance of making explicit what you plan to do and how you plan to do it. Woolf (2014a) describes these essential elements as the *strategies* and *tactics* of analysis.

Managing and referencing literature

Reviewing existing literature concerning your broad topic is a fundamental early task. Technological developments mean that this process is changing rapidly and significantly. Many journals have electronic versions providing free or easy access to full-text articles. Bibliographic software has developed to the point that it is quick and easy to transfer reference lists and online material directly into libraries, along with associated metadata. CAQDAS packages have also developed significantly in this area, with several now enabling the direct importation of PDF files and references from bibliographic software. Conducting a literature review within qualitative software is not only feasible, but also incredibly useful. Chapter 5 distinguishes between *direct* and *indirect* handling of literature, via annotating and coding full-text articles and/or developing critical appraisals about and linking within and between them. However you chose to proceed, integrating literature with the rest of your work through a CAQDAS package enables you to later systematically compare existing literature with your analysis.

Formulating the research problem and defining the research questions

Formulating the research problem is more than just deciding on the topic. It is informed by your ontological and epistemological standpoint and your familiarisation with and critiquing of the literature, both of which help you rationalise why

the area you are interested in requires further research. If you use software to facilitate the literature review, it makes sense also to write up your formulation of the research problem and define the initial research questions within the software project (Chapter 5). You can thus be explicit about your interest, assumptions, expectations and prejudices and link your writing to the literature that contributes to your problem formulation.

Representing theoretical frameworks

Whether your intention is to work within a clear theoretical framework, perhaps through applying existing theory or testing hypotheses on new bodies of data or areas of conceptual interest, or to develop theory from empirical data, you will never be working within a theoretical vacuum. Contrasting ways of working can broadly be distinguished according to the *direction* you are working in; whether *top-down* (deductive) or *bottom-up* (inductive). These approaches, and their combination (abduction), are discussed in the context of coding in Chapter 7. Whatever its role, it is important to relate your conceptualisation of the research problem to existing theory, to represent that within the software project at the outset and reflect how it evolves during the project. That might happen via memo-writing (Chapter 10) and/or the visualisation of theoretical contexts in visual maps (Chapter 11). You will be able to refer back to these ideas at later stages and compare initial assumptions and expectations with the analysis as you proceed.

Incorporating research materials

Data collection is all about constructing the best possible dataset in order to investigate the research problem. Under ideal circumstances, what data are required to answer the research questions? What data are available? Could you construct a suitable dataset from existing sources and conduct secondary analysis, or do you need to collect new data? What instruments will you use to generate new data if required? How will you ensure data are of sufficient quality? In the context of setting up a software project, you can create locations for storing data and other research materials early on (Chapter 5). You do not yet need to have data ready to incorporate. You may even change your mind and work with different materials later on, but thinking about data, how they are related to one another and how they will be handled as soon as possible is an important part of research design and software project set-up.

Defining factual features

Factual features are known characteristics about data and respondents (Chapter 12). Depending on your design you may sample on this basis; for example, if conducting a comparative case study in which you are focusing on two or more