

From guided-instruction to facilitation of learning: the development of *Five-level QDA* as a CAQDAS pedagogy that explicates the practices of expert users

Christina Silver^{a*} and Nicholas H. Woolf^b

^aQualitative Data Analysis Services, Dorset, UK; ^bWoolf Consulting Inc, Santa Barbara, CA, USA

(Received 10 November 2014; accepted 4 April 2015)

This paper introduces *Five-level QDA* (Qualitative Data Analysis) as a pedagogy for the teaching and learning of CAQDAS (Computer Assisted Qualitative Data Analysis) that spans methodologies, software packages and teaching modes. Based on the authors' personal trajectories of using, teaching and researching CAQDAS since the late-1990s, the paper illustrates the need for a CAQDAS pedagogy by describing the challenges of learners in powerfully harnessing CAQDAS packages. The principles behind *Five-level QDA* are outlined, which focus on the contrast between the strategies and tactics of conducting QDA with software, and the need to translate between these. The implementation of *Five-level QDA* as an adaptable method of instruction is illustrated through the use of *Analytic Planning Worksheets* in the *Recurring Hourglass* design.

Keywords: CAQDAS; CAQDAS pedagogy; CAQDAS workshops; Five-level QDA

Introduction

Computer Assisted Qualitative Data Analysis (CAQDAS) packages have been available since the 1980s, yet despite extensive use and the work of several authors that address the challenges of teaching CAQDAS (di Gregorio & Davidson, 2008; Lewins & Silver, 2007; Silver & Lewins, 2014), no general pedagogy has been developed that is applicable across methodologies, software packages, and teaching modes. The absence of widespread embedding of CAQDAS instruction into university curricular (Silver & Rivers, 2014) reflects the broader marginalised position occupied by the teaching of research techniques and methodologies (Kilburn, Nind, & Wiles, 2014). However, the plethora of digital tools available to support research work (Paulus, Lester, & Dempster, 2014) means researchers increasingly expect to employ customized software to undertake analysis. Teachers are therefore responsible for developing appropriate and effective pedagogies that take account of variety in research methodologies and analytic techniques and that equip learners with the skills required to undertake robust computer-assisted analyses.

Learning CAQDAS happens in many ways. Workshop-based training for groups is the most common, offered in many formats by different providers. Various online resources are also provided by software developers, university teachers and

*Corresponding author. Email: christina.silver@qdaservices.co.uk

independent consultants, including video tutorials, user manuals, analytically-oriented support materials, online courses, remote coaching options (di Gregorio, 2014; Kaczynski & Kelly, 2004), and a small but increasing number of modules at Higher Education Institutions (Davis & Krayner, 2014; Gibbs, 2014). The range of learning options is testament to both the variety in individual learning styles (Honey & Mumford, 1982) and the ongoing demand for CAQDAS teaching. Rapid CAQDAS software development and increasing use across sectors, disciplines and methodologies compounds the need to develop a CAQDAS pedagogy that transcends the specificities of products, methodological applications and modes of teaching.

This paper presents a CAQDAS pedagogy that meets these criteria and that is incorporated into our teaching practices. We first outline how our individual trajectories of using and teaching CAQDAS packages led us to identify the nature of researchers' struggles to employ CAQDAS packages powerfully, and thus the requirements of a general CAQDAS pedagogy. Using CAQDAS packages powerfully means using a program from the start to finish of a project while remaining true throughout to the emergent ethos of qualitative data analysis (QDA). In each of our trajectories we moved from an initial focus on guided-instruction to an increasing emphasis on facilitated learning in order to overcome the challenges of learning to use CAQDAS programs powerfully. However, whilst independently arriving at common principles, our emphasis was different. Silver focused on challenges caused by diversity amongst CAQDAS packages, varieties in their users and uses, and curriculum design of workshops as her predominant mode of instruction. Her key outcome was the *Recurring Hourglass* workshop design which enables learners to acquire abstract knowledge and practical experience by alternating learning activities and instructional methods. Woolf focused on learners' challenges in moving beyond straightforward to sophisticated uses of CAQDAS, and the need to unpack the 'black-box' of expert use that develops over many years of practice. His key outcome was a model of the CAQDAS process that expert users may undertake unconsciously, focusing on the contrasts between the strategies and tactics of a QDA when conducted with software, and the need to translate between them (Woolf, 2014). Since meeting in 2013, we have collaborated to develop and implement an adaptable CAQDAS pedagogy based on two innovations, referred to as *Five-level QDA*, which is designed to facilitate the learning of sophisticated uses of CAQDAS packages. In this paper we illustrate the implementation of *Five-level QDA* in the context of intensive two-day CAQDAS workshops using Analytic Planning Worksheets embedded within the *Recurring Hourglass* design.

Silver's personal trajectory and the evolution of the *Recurring Hourglass* design

Phase 1: guided-instruction (ca. 1997–2002)

My experience during postgraduate research as a concurrent learner-teacher, with frequent delivery of intensive workshop-based training via my role at the CAQDAS Networking Project (CNP),¹ served as a gradual enculturation into a community of practice of CAQDAS teachers (Lave & Wenger, 1991). CAQDAS use was not yet widespread and experienced researchers were primarily interested in learning what these new technologies 'could do', seeking streamlined ways of undertaking already established analytic practices. Many had limited experience with complex computer packages and required step-by-step instruction in software operation. These factors informed workshop delivery via *guided-instruction* in the operation of the software.

Hand-outs containing detailed step-by-step instructions in software operation were refined and expanded as new software versions became available and were increasingly embedded as both an in-workshop learning tool and post-workshop resource.

The pedagogic aims of this whole-group hands-on experience were to familiarize learners with CAQDAS functionality by performing analytic tasks using sample data, and to encourage them to reflect on its potential role for their own projects. Such experiential learning involving concrete experience, observation and experimentation (Kolb, 1984) is a feature of my training provision throughout my trajectory.

During this phase, the standard one-day workshops followed a structured plan designed to impart key facts and to encourage learners' to employ features creatively for their own needs. Although the workshops succeeded in imparting the necessary information to operate software and were positively evaluated, it became clear that the focus and delivery were insufficiently flexible to accommodate a growing range of learner needs. As interest in and uptake of CAQDAS increased so did learner heterogeneity, and the need became increasingly pressing to deliver workshops that were more responsive to participants' varied needs. Additionally, teaching many CAQDAS packages highlighted their differences and the recognition that there is no 'best' CAQDAS package or ideal way of using the programs. Rather, they need to be chosen and used according to individual project characteristics (Fielding & Lee, 1991). This led to a transition to instruction-led facilitation in which the importance of planning for software use was prioritized.

Phase 2: instruction-led facilitation (ca. 2002–2008)

New CAQDAS packages and an increase in demand for training and on-going support led to more bespoke in-house workshops and personal coaching for researchers and project teams. To better meet learners' increasingly diverse needs, intensive one-day introductory workshops were redesigned to emphasize different uses of software. *Guided-instruction* using sample data was augmented with project-based illustrations from my own research and the use of learners' projects as instructional examples. Technical instruction concerning the logic and functioning of software remained core, but emphasis shifted towards facilitating the use of software features for the analytic needs of learners' own projects, and reflecting on the implications arising from illustration of my own projects. This allowed time for learners to experiment with software independently. Additionally, advanced workshops and user seminars were provided by the CNP to meet the needs of cohorts of existing CAQDAS users. Learners brought their work-in-progress and time was allotted to sharing and reflecting on the successes and challenges of using software for specific research projects.

An outcome of this phase was the first edition of *Using Software in Qualitative Research: A Step-by-step Guide* (Lewins & Silver, 2007), a textbook which included a model of common QDA processes: *integrating data*, *exploring data*, *identifying and organizing data*, *asking questions* and *interpretation*. Framing workshop instruction around these processes using real-world examples allowed the potential role and flexibility of CAQDAS packages to be more concretely illustrated. This transition to *instruction-led facilitation* was positively evaluated, and this emphasis on the agency of learners developed further during phase three.

The model of common QDA processes for planning software use in the context of specific project needs (Lewins & Silver, 2007) was a step forward in communicating the potential role of CAQDAS packages. Other textbooks and training providers have also related software features to analytic processes with their own standard models of underlying QDA processes (e.g. di Gregorio & Davidson, 2008; Friese, 2014). However, this approach is problematic because it can be misconstrued as a ‘one-size-fits-all’ model, which risks propagating the perception that software is a method rather than a tool. An outcome of increasing learners’ agency in workshops during the second phase was feedback that learners would benefit from illustrations that were more explicitly methodological. These observations led to further developments in textbook content and workshop delivery.

Phase 3: facilitation-led instruction (ca. 2008–2013)

In order to address these issues the CNP undertook qualitative longitudinal research tracking researchers’ learning and use of CAQDAS packages over 12 months (Silver, 2010). Results illustrated that successful use was related to methodological awareness, adeptness in the techniques of analysis, and technological proficiency (Silver & Rivers, 2014). *Methodological awareness* refers to familiarity with the variety in QDA philosophies and methodological approaches that underpin the choices researchers make in undertaking analysis. *Analytic adeptness* concerns learners’ experience in undertaking QDA, specifically the skills in designing analytic tasks in the context of underlying methodology and philosophy. *Technological proficiency* refers to competency in operating software and comfort with the idea of experimentation without fear of making mistakes. These findings informed the second edition of the textbook to incorporate a case-study approach (Silver & Lewins, 2014), and the re-focusing of workshop delivery to become more specific to the varied methodological contexts and analytic activities of each learner. The pedagogy that evolved constituted an intentional shift towards *facilitation-led instruction* by further emphasizing the responsibility of learners to plan in detail for their own software use in the context of their own methodologies, and culminated in the development of the *Recurring Hourglass* workshop design (Figure 1).

It was not feasible to provide examples of software use for the full range of qualitative methodologies. Instead case-studies were developed to illustrate the potential of CAQDAS in supporting each aspect of the research process, from problem formulation and literature review through data analysis to final write-up, without prescribing a ‘one-size-fits-all’ method for any methodological approach or analytic task. This was achieved by contrasting the *commonalities* in analytic activities across methodologies, and the *differences* in how they manifest in the selection and use of software features for specific analytic needs. The model from the 2007 textbook was thus adapted to focus on the analytic activities of *integration, organisation, reflection, exploration* and *interrogation* (2014, p. 45), and discussed both in terms of the case-studies and learners’ own projects, to facilitate them in designing appropriate ways of using CAQDAS features.

Additional instructional strategies were required to implement this expanded workshop instruction, culminating in the *Recurring Hourglass* design (see Figure 1). This is an adaptation of the elaboration theory of instruction (Reigeluth, 1979), and is designed to be applicable to face-to-face workshops of varying lengths and formats.

Focus ===== Level of learning	Instructional design
DAY 1	
Narrow ===== Whole-group	<p><u>OBJECTIVE: INTRODUCTIONS</u></p> <ul style="list-style-type: none"> • Set the tone of the workshop and establish a conducive learning environment via teacher and learner introductions. • Identify similarities and differences in participants’ projects to foster connections and prompt discussion. <p><u>INSTRUCTIONAL STRATEGIES</u></p> <ul style="list-style-type: none"> • Describe research experience and background, current project, needs from and expectations of training.
Broad ===== Whole-group	<p><u>OBJECTIVE: CONTEXTUAL DISCUSSIONS</u></p> <ul style="list-style-type: none"> • Ensure learning is grounded in appropriate contexts • Emphasize the flexibility and adaptability of software, thereby encouraging learners to reflect on their specific needs and to carefully plan for their use of software. • Establish the agency of learners in workshop-based learning and in appropriately manipulating software features <p><u>INSTRUCTIONAL STRATEGIES</u></p> <ul style="list-style-type: none"> • Slide-illustrated lecture discussing relevant underlying contexts: developmental; methodological; analytic; and practical and analytic activities: integration, organization, exploration, reflection and interrogation (drawn from Silver & Lewins, 2014). • Introduce <i>Five-level QDA</i> as a means of distinguishing between analytic strategies and tactics
Broad ===== Whole-group	<p><u>OBJECTIVE: SOFTWARE OVERVIEW</u></p> <ul style="list-style-type: none"> • Introduce software in its entirety to raise awareness of the breadth of its application and potential affordances to learners’ work <p><u>INSTRUCTIONAL STRATEGIES</u></p> <ul style="list-style-type: none"> • High-level description and illustration of software architecture and features via guided walk-through using a sample project
Narrow = Individual ====	<p><u>OBJECTIVE: ANALYTIC PLANNING #1</u></p> <ul style="list-style-type: none"> • Introduce Analytic Planning Worksheets (APWs) (Figure 3) as a tool for effectively designing software-assisted analysis <p><u>INSTRUCTIONAL STRATEGIES</u></p> <ul style="list-style-type: none"> • Illustration of partially completed Analytic Planning Worksheet and discussion about its logic in relation to project characteristics • Provide learners’ with time to begin completing an APW for Levels 1 & 2 (objectives and general analytic plan)
Narrow ===== Whole-group	<p><u>OBJECTIVE: GUIDED-INSTRUCTION</u></p> <ul style="list-style-type: none"> • Provide opportunity for hands-on experimentation of straightforward uses of software features in the context of a real-world project <p><u>INSTRUCTIONAL STRATEGIES</u></p> <ul style="list-style-type: none"> • Step-by-step guided set-up of a software project following the APW illustrated in Analytic Planning #1 workshop phase
Broad ===== Whole-group	<p><u>OBJECTIVE: ILLUSTRATIVE DISCUSSIONS</u></p> <ul style="list-style-type: none"> • Illustrate range of applications of software features • Ensure examples are relevant to learners’ contexts • Encourage reflection on the affordances of software features for learners’ particular needs <p><u>INSTRUCTIONAL STRATEGIES</u></p> <ul style="list-style-type: none"> • Demonstration and discussion of several contrasting software projects to illustrate the influence of project characteristics on use of software • Illustrations related to contextual discussions and learners’ projects

Figure 1. The *Recurring Hourglass* design: example of a two-day model with embedded *Five-level QDA*.

It is intended to build knowledge cumulatively such that each stage of instruction serves to elaborate the previous stage as well as serving as an overview of the next stage of elaboration.

Narrow ===== Individual	<u>OBJECTIVE: INDEPENDENT APPLICATION</u> <ul style="list-style-type: none"> • Provide learners with opportunity to begin applying learning in the context of their own project needs <u>INSTRUCTIONAL STRATEGIES</u> <ul style="list-style-type: none"> • Learners create new software project and set it up for their specific needs • Learners are supported as required with specifics of software operation
DAY 2	
Narrow ===== Whole-group	<u>OBJECTIVE: LEARNER SUMMARIES</u> <ul style="list-style-type: none"> • Encourage reflection on the learning from day one, focus attention on objectives for day two and re-establish focus of workshop <u>INSTRUCTIONAL STRATEGIES</u> <ul style="list-style-type: none"> • Describe ways in which it is envisaged software will be used. Outline intentions for the day
Narrow ===== Individual	<u>OBJECTIVE: ANALYTIC PLANNING #2</u> <ul style="list-style-type: none"> • Revisit APW and with key learning objectives for the day in mind fill out Levels 1, 2a and 2b in as much detail as possible <u>INSTRUCTIONAL STRATEGIES</u> <ul style="list-style-type: none"> • Short review of partially completed APW serves as a reminder of the logic of the Worksheet and its role in planning for powerful software use. • Facilitator supports individual learners fill out APW as required
Narrow ===== Individual	<u>OBJECTIVE: INDEPENDENT APPLICATION</u> <ul style="list-style-type: none"> • Learners continue with own work, following their APW <u>INSTRUCTIONAL STRATEGIES</u> <ul style="list-style-type: none"> • Facilitator spends time with each learner providing individual coaching as required • If several learners are struggling with operating software for a similar analytic task, individual work is interrupted to instruct the group as a whole
Broad ===== Whole-group	<u>OBJECTIVE: ILLUSTRATIVE DISCUSSIONS</u> <ul style="list-style-type: none"> • Illustrate a selection of sophisticated uses of software tools in real-world research projects • Enable learners to reflect on the implications of contextual discussions, refine analytic planning and translate abstract learning to their own needs <u>INSTRUCTIONAL STRATEGIES</u> <ul style="list-style-type: none"> • Demonstration and discussion of several contrasting software projects to illustrate the influence of project characteristics on use of software • Illustrations related to contextual discussions and participants' projects
Narrow ===== Individual	<u>OBJECTIVE: INDEPENDENT APPLICATION</u> <ul style="list-style-type: none"> • Participants continue with own work, following their APW <u>INSTRUCTIONAL STRATEGIES</u> <ul style="list-style-type: none"> • Facilitator spends time with each participant providing individual coaching as required • If several participants are struggling with operating software for a similar analytic task, individual work is interrupted to instruct the group as a whole
Narrow ===== Individual	<u>OBJECTIVE: ANALYTIC PLANNING #3</u> <ul style="list-style-type: none"> • Revisit APW, focusing on Levels 3, 4 and 5 <ul style="list-style-type: none"> • Introduce Translation Worksheets and show how to fill one out <u>INSTRUCTIONAL STRATEGIES</u> <ul style="list-style-type: none"> • Description of the process of translation and the difference between selecting and constructing tools • Illustrate that different tools may be used to fulfil the same set of analytic tasks and that appropriate choices are determined by methodology and principles guiding QDA and the general analytic plan
Broad ===== Whole-group	<u>OBJECTIVE: ILLUSTRATIVE DISCUSSIONS</u> <ul style="list-style-type: none"> • Illustrate a selection of sophisticated uses of software tools in real-world research projects • Enable participants to reflect on the implications of contextual discussions, refine analytic planning and translate abstract learning to their own needs <u>INSTRUCTIONAL STRATEGIES</u> <ul style="list-style-type: none"> • Demonstration and discussion of several contrasting software projects to illustrate the influence of project characteristics on use of software • Illustrations related to contextual discussions and participants' projects
Broad	<u>OBJECTIVE: CONTEXTUAL DISCUSSIONS</u>

Figure 1. (Continued).

===== Whole- group	<ul style="list-style-type: none"> • Remind participants of contexts within which software is used and range of analytic activities that can be undertaken using software • Provide opportunity for participants to discuss their work <p><u>INSTRUCTIONAL STRATEGIES</u></p> <ul style="list-style-type: none"> • Revisit contextual discussion slides used on day one to frame discussion • Q&A in relation to participant projects
Broad ===== Whole- group	<p><u>OBJECTIVE: PREPARATION FOR POST-WORKSHOP</u></p> <ul style="list-style-type: none"> • Prepare participants for post-workshop software use <p><u>INSTRUCTIONAL STRATEGIES</u></p> <ul style="list-style-type: none"> • Review range of resources available • Demonstrate how to access online materials • List sources of individual support
Narrow ===== Whole- group	<p><u>OBJECTIVE: PARTICIPANT REFLECTIONS</u></p> <ul style="list-style-type: none"> • Encourage participants to reflect on learning and plan for continued software use. <p><u>INSTRUCTIONAL STRATEGIES</u></p> <ul style="list-style-type: none"> • Summarise individual key learning outcomes and planned next steps with analysis

Figure 1. (Continued).

The design alternates activities with a broad, whole-group focus and a narrow, individual focus, hence the term *Recurring Hourglass* for this repeated oscillation between foci, and is based on an important lesson of the prior two phases: learning how to effectively use CAQDAS features requires a comprehensive understanding of the role, functioning and potential of CAQDAS packages *before* learning how to operate the software. This preliminary understanding is referred to by Reigeluth (1979) as the top-level *epitome*, or conceptual orientation to the coming material that is presented at an application level without requiring learning pre-requisites. In CAQDAS workshops this includes an overview of the software as a whole, illustration of its longer-term potential in later phases of a data analysis, and the appropriate use of software features in different phases of a data analysis. The complexity of CAQDAS packages and the diversity of learners' prior experience and current needs requires careful management in order to balance the acquisition of abstract knowledge and practical hands-on experience.

Alternating workshop activities occur in different sequences and frequencies depending on workshop format, learner needs, and the momentum and progress of individual workshops, enabling methodologically relevant facilitation of learning, which involves illustrating uses of software tools in a variety of analytic contexts. Figure 1 illustrates in detail the instructional strategies in a typical two-day workshop using the design.

The generally accepted criteria for successful instructional design is that students find instruction efficient, appealing, and effective (Smith & Ragan, 2005). The pedagogy in this phase was more successful in the first two criteria than the third. Participant feedback indicated that the use of real-world examples and a variety of instructional strategies were indeed efficient and appealing ways to make the teaching of CAQDAS packages more relevant to their needs. However, while the focus on analytic activities, the case study approach, and the *Recurring Hourglass* design emphasized user agency in using software features appropriately; research (Silver & Rivers, 2014), observation of learners, and participant feedback suggest that learners still struggle to employ CAQDAS powerfully in their own work after the workshop. Enabling learners to translate workshop illustrations into their own practice is the

core issue of the collaboration between both authors to implement *Five-level QDA* as a CAQDAS pedagogy applicable across methodologies, software packages and teaching modes.

Woolf's personal trajectory and the evolution of the framework for *Five-level QDA*

Phase 1: guided-instruction (ca. 1997–2003)

While conducting postgraduate research in the early 1990's I concluded that to develop sufficient CAQDAS expertise I needed to select a single package that could be adapted to the widest range of methodologies. I chose ATLAS.ti, and after using the program in a number of projects and coaching other students and instructors in the software, I taught a post-graduate class at the University of Iowa in CAQDAS, devoting one quarter of each class to instruction in operating ATLAS.ti. I began conducting two-day workshops across the USA with the same focus on teaching the operation of the software features in a logical order that matched the kinds of research projects I was involved in, typically using grounded theory approaches and content analysis.

Many participants requested post-workshop coaching, an experience that highlighted two limitations of this *guided-instruction* pedagogy. First, many learners were not conducting grounded theory or content analysis, and had either set up their projects as if they were or were using the software only for data management or initial coding. Others had a different problem: instruction in software operation out of the context of their own studies led to inert knowledge, knowledge gained out of the context of a real application and that is not later used in problem solving even though relevant (Whitehead, 1929). During follow-up consultations learners had to be instructed afresh in the selection and use of software features appropriate for their data analyses. These observations led me to adopt an anchored-instruction approach in the second phase, in which the software could be perceived as a tool for problem-solving as an expert would use it, rather than an arbitrary procedure (Bransford, Sherwood, Hasselbring, Kinzer, & Williams, 1990). Rather than teaching software features independent of a specific use, I adopted an *instruction-led facilitation* approach in order to enable learners' to problem-solve in their own projects.

Phase 2: instruction-led facilitation (ca. 2003–2010)

To avoid the problem of inert knowledge described above, I identified three key learning objectives for gaining transferable CAQDAS skills in workshop contexts: (i) to understand the software architecture as a whole through a conceptual overview of the entire program, to serve as an extended *advance organizer* (see below); (ii) to learn to conceptually harness the software for the particular purposes of a variety of studies; and finally (iii) to learn to operate the software features and put the first two objectives into practice.

Initially I alternated instruction amongst these three learning tasks in small chunks of one to two hours each. Over time the critical importance of the *advance organizer*, and the lowest priority of learning to operate the software features, became more and more apparent, based on learners' feedback during the workshops and their ability to later employ the software successfully. An advance organizer

is the ‘introduction of relevant subsuming concepts’ prior to learning unfamiliar material (Ausubel & Robinson, 1969, p. 29). Mayer (1979) reported that in the case of ‘expository advance organizers’, such as my extended conceptual overviews of the architecture of the software, the advance organizer has its strongest effect on the later transfer of knowledge to new domains (p. 382). In accordance with these principles, the extended conceptual overview was presented in complete form at the outset, and often filled the entire first day of the workshop without participants opening their laptops until the second day. This workshop design was evaluated more positively by participants.

Despite the advantages of this workshop design it was still apparent that participants experienced difficulty in transferring their learning to using the software powerfully in their own projects. Silver, who experienced the same phenomenon at approximately the same time in the UK, addressed this by developing the *Recurring Hourglass* design. I approached the issue differently by addressing the need to further unpack the process of powerfully harnessing software that develops in CAQDAS experts over time.

Phase 3: facilitation-led instruction (ca. 2010–2013)

The third phase was characterized by unpacking CAQDAS experts’ processes for harnessing CAQDAS packages, such that learners could more quickly employ software powerfully. Observation of my own research practices, the practices of other long-time researchers, and the challenges of novice researchers, led to the recognition that CAQDAS expertise lies in effectively reconciling the underlying contradiction between the strategies and tactics of a QDA when conducted with software. The strategies – *what you plan to do* – are to varying degrees iterative and emergent, while the tactics – *how you plan to do it* – comprise the use of cut-and-dried, pre-determined computer software tools. Novices often approach the process by first learning the software operations and looking for ways of using them in the data analysis, not recognizing the underlying contradiction or the need to reconcile it. Experts learn over time that powerful CAQDAS use begins by first specifying analytic tasks without regard to the available program features, and then harnessing appropriate software tools to accomplish the tasks.

The second learning objective – learning to conceptually harness the software for particular purposes in each learners’ study – was therefore reoriented to distinguish strategies from tactics in a QDA and demonstrate the process of translating between them. However, while learners appreciated the value of this process in the context of specific illustrations, they could not reliably transfer the learning to their own, novel situations after the workshop. It became clear that the translation process was not sufficiently explicated as a systematic process. I therefore searched the literature for a model of strategy and tactics that could inform the CAQDAS process.

Military studies is amongst few disciplines that investigate this relationship between strategies and tactics. The military strategist Edward Luttwak (2001) identified five levels of strategy as a descriptive model that does not prescribe specific courses of action, but provides the principles applicable to any context of conflict. At first glance a military metaphor does not appear apposite to qualitative research. Yet Luttwak’s five levels bear an uncanny resemblance to the general processes applicable to the use of CAQDAS packages regardless of the specifics of the

methodology or the content of a project, just as Luttwak's model describes the general processes of military strategy regardless of the type of conflict. Luttwak (2001)'s seminal contribution, now incorporated into army training manuals, is the identification of a middle level of operational coordination between strategies and tactics that is itself neither a strategy nor a tactic, but serves as the fulcrum between them. The characteristics of this middle level of operational coordination in the military context correspond closely to the middle level of translation between the strategies and tactics of a QDA that is undertaken unconsciously by expert CAQDAS users. Luttwak's (2001) exposition of military strategy informed the formalization of the expert process of CAQDAS as *Five-level QDA* (Woolf, 2014).

Learners' struggles to harness CAQDAS powerfully and the principles underlying *Five-level QDA*

Our separate experiences of teaching thousands of researchers and students, observing and investigating how they learn and use CAQDAS packages, and undertaking and consulting on numerous software-assisted analyses, has resulted in our continual reflection about the most effective ways to equip users with the skills required to harness CAQDAS packages powerfully. We independently reached the same conclusion: that learning how to use CAQDAS effectively requires a comprehensive overview of software architecture, a sense of its longer-term potential and the appropriate use of individual features at different stages of a project; *before* learning to operate any aspect of the software. We each gradually modified our workshop designs to give primacy to learning to harness rather than simply operate software, and to alternate these learning tasks. Despite our different backgrounds and teaching contexts, our observations of learners' struggles and our approaches to enabling them to overcome these are similar. Three principle causes of learners' struggles to use software powerfully underlie our refinement and implementation of *Five-level QDA*.

First is the misconception that there is an ideal or right way to use CAQDAS packages. The one-size-fits-all models often presented in textbooks are inappropriate because the objectives, methodologies, and analytic procedures of individual projects are so varied. It is neither possible nor appropriate to teach CAQDAS as if it were a uniform method.

Second is the skill of harnessing software features for sophisticated uses. Learning CAQDAS packages in a mechanical way, by focussing on operating the software out of the context of accomplishing particular analytic tasks, may be sufficient to allow researchers to use the software effectively for straightforward purposes. But CAQDAS users must often go beyond straightforward uses by harnessing software features in combined or unusual ways for more sophisticated uses that may not have been anticipated by the software developer, and may not be apparent from the presentation of the features on the computer screen.

Finally, the methods literature speaks almost exclusively of iterative and emergent analytic strategies, without distinguishing the contrasting characteristics of the more cut-and-dried tactics that will be used to execute those strategies, whether the tactics are coloured markers and whiteboards or CAQDAS packages. In the case of CAQDAS packages, thinking of analytic strategies and software tactics as a single activity can increase the challenge of going beyond straightforward uses of the software. This is compounded when the analytic strategy is not sufficiently clear, and

the software features are used as a kind of substitute for steps of analysis, making the analysis less iterative and emergent than is appropriate.

These principles frame our work in combining the most effective elements of each of our practices into an explicit method of teaching and using CAQDAS packages: *Five-level QDA*, a process that enables learners to harness software features to the needs of a specific analytic task as the last step of the analytic process, as opposed to first instructing them in mechanical software operations which then drive the choice and design of the analytic tasks that are undertaken.

Five-level QDA and translation

When CAQDAS packages are used in a mechanical way, use is often restricted to early project phases, or features are used in a more mechanical manner than is called for by the methodology. Powerful use refers to using CAQDAS packages for every stage of a project, from the very start to the very end, while remaining true throughout to the iterative and emergent spirit of qualitative research. This involves harnessing software features rather than simply operating them, and is achieved by distinguishing analytic strategies (which are to varying degrees iterative and emergent) from analytic tactics (which in the case of using computer software are more cut-and-dried) and consciously translating between them. This process, which we refer to as *Five-level QDA* (Woolf, 2014; Woolf & Silver, in press), is not proposed as a new or different method of undertaking computer-assisted analysis, but rather as an exposition of the processes that expert CAQDAS users undertake unconsciously (see Figure 2).

The objective of *Five-level QDA* is to resolve the contrast between the emergent nature of the analytic strategies and the more cut-and-dried nature of the software tactics in a conscious and effective manner. The focus is on the process of translation between the units of analysis that comprise each individual analytic task, and the affordances of the software features (Gaver, 1991). For analytic tasks in which the units are *formally equivalent*, analogous to word-by-word translation from one language to another (Nida & Taber, 1969), the translation process is straightforward, and a software feature can be selected and used for the task in the obvious manner. For example, using Microsoft Word as an analogy, the cut and paste buttons can be

<i>FIVE-LEVEL QDA</i>				
<i>two levels of strategy >>>>> translated to >>>>> two levels of tactics</i>				
<i>Level 1</i>	<i>Level 2</i>	<i>Level 3</i>	<i>Level 4</i>	<i>Level 5</i>
<u>Objectives:</u>	<u>Analytic plan:</u>	<u>Translation:</u>	<u>Selected tools:</u>	<u>Constructed tools:</u>
The purpose and context of a project, usually expressed as a research question and methodology	The conceptual framework and resulting analytic tasks	Translating from analytic tasks to software tools, and translating the results back again	Individual software operations used in a straightforward way	Combinations of software operations, or software operations used in a custom way

Figure 2. *Five-level QDA*.

used in the obvious way to cut and then paste text to a different location in a document. This is referred to in *Five-level QDA* as using a *selected tool*. However, as a qualitative data analysis proceeds the units of analytic tasks may not be formally equivalent to any obvious software feature. Continuing the analogy with Microsoft Word, the formatting of the text to be cut may be different from the formatting of the location to which it will be pasted, and without knowledge of the several alternative ways to paste the formatting as well as the characters of text, the most straightforward use of the cut and paste buttons may lead to undesired formatting in the pasted location. Returning to CAQDAS packages, it is in analogous situations that many users move the analysis off the CAQDAS package and continue in Microsoft Word or Excel or use pen and paper, or alternatively use the CAQDAS package in ways that to some extent suppresses the more complex or emergent aspects of the analytic tasks. To use CAQDAS packages powerfully the translation process needs to become more sophisticated analogous to *dynamic equivalence* in language translation in which word-by-word translation is inadequate, and high quality translation requires finding ‘just the right words’ in the second language to produce the same effect in the reader as the words in the first language (Nida & Taber, 1969). In *Five-level QDA* this is referred to as using a *constructed tool*, which simply means using software features in combination or in unusual ways that may not be obvious from their presentation on the screen. Expert CAQDAS users have typically learned through long practice to unconsciously undertake either straightforward or sophisticated translation of analytic tasks into *selected* or *constructed tools* as necessary. The purpose of *Five-level QDA* is to accelerate the gaining of this expertise by making the translation process conscious and explicit, thereby prioritizing the agency of learners in harnessing software. In workshop contexts this means the role of the teacher as facilitator rather than instructor is emphasized.

Implementing Five-level QDA

We have developed the *Analytic Planning Worksheet* to explain and illustrate translation in the context of learners’ own projects, and to enable learners to become proficient in the process. *Analytic Planning Worksheets* enable learners to differentiate between the two levels of strategy (*objectives* and *analytic plan*) and the two levels of tactics (*selected tools* and *constructed tools*), and to develop skills in *translating* analytic tasks into software operations.

Figure 3 illustrates part of an *Analytic Planning Worksheet*, showing one phase of action of a critical discourse analysis study. We document whole projects in this way, with one such worksheet for each phase of action, broken down into its constituent analytic tasks (Figure 3 illustrates three).

If a software feature exists that directly maps onto the analytic task, the translation process is straightforward and a tool can be selected to fulfil the task (as with the first analytic task in Figure 3). Where the analytic task is more complex and/or where there is no individual software feature available to fulfil it, a tool has to be constructed out of the use of more than one feature (as with the second and third analytic tasks in Figure 3). The purpose of the translation column in Figure 3 is to unpack this process. Where translation is straightforward, no additional work is required. However, where it is sophisticated because it is not immediately clear how to harness software features to fulfil an analytic task, an additional *Translation Worksheet* is used to explicate the analytic units, to evaluate software affordances, to

<p>Project: “Theory-testing Critical Discourse Analysis” of Tony Blair’s speeches post office, using MAXQDA v11 ANALYTIC PLANNING WORKSHEET: Phase 4</p>	
<p>Level 1: OBJECTIVES</p>	<p>RESEARCH QUESTIONS/OTHER OBJECTIVES: To what extent are elements of Blair’s rhetorical style, as identified by Fairclough in <i>New Labour, New Language, 2000</i>, identifiable in speeches he made during the 4 years after leaving office?</p>
<p>Level 2a: GENERAL ANALYTIC PLAN</p>	<p>METHODOLOGY/PRINCIPLES GUIDING ODA: “Theory-testing Critical Discourse Analysis” of Fairclough’s (2000) <i>interpretation of Blair’s rhetorical styles as a distinctive repertoire</i>. Process informed by Rosalind Gill’s Discourse Analysis process (in Bauer & Gaskell, 2000) but adapted to specifically test the applicability of Fairclough’s (2000) theory of New Labour, New Language.</p> <p>SUMMARY OF CONCEPTUAL FRAMEWORK See Phase 3 Worksheet and graphic model developed in MAXQDA software project : (Phase 3) Conceptual Framework#1 CONCEPTS: 5 of 6 aspects of Fairclough’s “Blair’s Distinctive Repertoire” (Blair’s Poetic, Conviction Politician, Getting Tough, Consensus Politician, the Normal Person) VARIABLES: Time since leaving office, Speech Topic/Genre, Speech Purpose.</p> <p>PRIOR PHASES OF ACTION COMPLETED: (Phase 1) Critical appraisal of Fairclough’s interpretations, (Phase 1a:Action Arising) Refinement of research question, (Phase 2) Develop conceptual framework, (Phase 3) Construct dataset</p> <p>CURRENT PHASE OF ACTION: (Phase 4) Categorize speeches to relevant variables</p> <p>NEXT PHASE OF ACTION ANTICIPATED: First critical reading of speeches</p>
<p>Level 2b: CURRENT SET OF ANALYTIC TASKS</p>	<p>Level 3: TRANSLATION ST: no translation worksheet required CT: see Translation Worksheet #4a CT: see Translation Worksheet #4b</p>
<p><i>(First analytic task of current phase of action)</i> Incorporate speeches into MAXQDA project (name speeches using consistent protocol that includes speech number, date of delivery and short descriptive title)</p>	<p>Levels 4/5: SELECTED OR CONSTRUCTED SOFTWARE TOOLS (ST or CT) <i>Selected Tool:</i> Create DOCUMENT GROUPS for each year (to compare over time) and import speeches. (add number of speeches per year to DOCUMENT GROUP name)</p>
<p><i>(Second analytic task of current phase of action)</i> Assign relevant factual variables to speeches (date, topic, genre, location, total running words, range of running words)</p>	<p><i>Constructed Tool:</i> (two-step process) 1) Export DOCUMENT VARIABLES to Excel, remove system variables, add variables 2) Import DOCUMENT VARIABLES</p>
<p><i>(Third analytic task of current phase of action)</i> Create analytically meaningful sub-sets of data for later interrogation (based on Speech Topics and Running Words Ranges)</p>	<p><i>Constructed Tool:</i> (series of three-step processes) 1) Identify most frequent speech topics using STATISTIC OF DOCUMENT VARIABLES 2) ACTIVATE DOCUMENTS BY DOCUMENT VARIABLES 3) Create and rename DOCUMENT SETS - For most frequent Topic groups : Faith (N=7), Climate Change (N=8), Global Relations - comprising Europe OR Middle East OR Northern Ireland (N=6). Create control group of remaining speeches. - Repeat for each Running Word Range (in groups of 500 from >500 to 4000-4999)</p>
<p>ACTION ARISING / SUMMARY OF OR REFLECTIONS ABOUT THIS COMPLETED PHASE OF ACTION: Speech Topic DOCUMENT SETS to constitute the main groups for comparison. Need to work out how to account for differences in length of speech when quantifying coding later. Next task of action, as planned: to critically read and summarize speeches, but will do this one Speech Topic at a time rather than in historic order of speech delivery.</p>	

Figure 3. Five-level QDA Analytic Planning Worksheet, example of one phase of action from a project using MAXQDA version 11.

select the software components to act upon, and finally to specify the sequence of software operations to undertake.²

Repeating the framing elements in each iteration of an *Analytic Planning Worksheet* (i.e. Level 1 and Level 2a) reflects the iterative nature of QDA, so that refinements following each phase of analysis is systematically documented. The *Analytic Planning Worksheet* for Phase 4 of the data analysis (see Figure 3) includes the refined research question. In practice, the *Analytic Planning Worksheets* for all phases would appear one after the other in a single continuous file to serve as a detailed audit-trail of the data analysis, allowing easy review of each iteration of the process.

Workshops modelled on the *Recurring Hourglass* design introduce the principles of *Five-level QDA* early during the introductory discussions about analytic contexts (Figure 1), and are revisited throughout. Learners begin filling out an *Analytic Planning Worksheet* after the first guided walk-through, once the architecture of the software has been illustrated in high-level overview. First they list Level 1 Objectives to identify what they plan to do in general terms in order to answer the research question. Later in the workshop they choose a phase of action to focus on, and transform the list-based analytic plan into a discrete set of tasks on their *Analytic Planning Worksheet*. This serves to focus attention on designing analytic tasks to be effected in software with regard to the strategies (the project objectives), rather than the tactics (the features of the software).

In two-day workshops participants revisit their *Analytic Planning Worksheet* several times, making alterations as required, and as the workshop progresses they begin filling out the software operations column in light of what they are learning about the program capabilities. The focus of day-two is hands-on work to put their plans into practice.

The adaptability of Five-level QDA

This paper has focused on illustrating the implementation of *Five-level QDA* within two-day intensive CAQDAS workshops. However, *Five-level QDA* can be adapted to different learning contexts. It has been successfully implemented in face-to-face workshops with groups varying from 5 to 14 participants, and over one or two days, and in remote coaching sessions delivered via Skype with both teams and individual researchers. It is also being written in textbook format for researchers already familiar with the basic operations of their chosen software who wish to accelerate the transition from basic to advanced use of the program (Woolf & Silver, *in press*). As well as using the textbook to augment our own teaching, *Five-level QDA* can be adopted by other CAQDAS teachers and learners.

Conclusion

Our combined trajectories of almost 40 years of CAQDAS teaching highlight the pedagogical challenges of developing learners' expertise in the powerful use of these software packages. The fast pace of software development as well as the rise of mixed methods, secondary analysis, and longitudinal qualitative research, have increased researchers' expectations of CAQDAS (Paulus et al., 2014; Silver & Lewins, 2014). Additionally, a new generation of digital-native students expect computer-assisted

solutions in all fields, including qualitative data analysis. Yet CAQDAS teaching is not widely embedded in higher education, as demonstrated by the high proportion of postgraduate students attending workshops: 68% of CNP workshop attendees in the 2013–2014 academic year (Silver & Rivers, 2014).

Increased demand for CAQDAS teaching has been accompanied by two trends that created distinct pedagogical challenges: increasing complexity of CAQDAS packages and increasing diversity amongst users and uses to which software is put. Complexity brings benefits of broader applicability, extension of analytic possibilities, mixing of techniques and collaborative working. The challenge for learners is in uncovering appropriate pathways to undertaking systematic and rigorous analysis. Because there are now so many alternative ways to execute particular analytic tasks, confusion is common (Silver & Rivers, 2014). The pedagogical challenge is to enable learners to traverse software complexities and employ their features appropriately.

The growing diversity of user needs and uses of CAQDAS has resulted in an increasingly heterogeneous population of learners seeking instruction in open-registration workshops, the mainstay of CAQDAS teaching. Certainly there are pedagogical advantages to this heterogeneity: diversity in project characteristics exposes learners' to a range of research contexts and illustrates the importance of harnessing CAQDAS features specifically to the requirements of each analytic task, and novice researchers benefit from hearing about the practices and experiences of more experienced researchers. However, diversity increases the challenge of ensuring each learners' needs are met.

It is common to address diversity by developing a 'one-size-fits-all' method for using CAQDAS that presents a generic set of computer-assisted QDA procedures applicable to all learners, methodologies and projects.³ However, our experience convinces us that this is neither feasible nor appropriate (Lewins & Silver, 2007) and to varying extents suppresses the iterative and emergent nature of QDA (Woolf, 2014). A 'one-size-fits-all' approach implies that the use of software itself constitutes a method of analysis, but just as there is no one method of QDA, there is no one method of using software. Our trajectory of pedagogic development has moved in the opposite direction, by increasingly *facilitating* learners to harness software appropriately for their specific requirements, rather than *instructing* them in a method of use. However, this is no easy matter given the complex task of applying cut-and-dried computer software to the emergent activity of QDA. We are encouraged that *Five-level QDA* is proving successful in our own practices, by continuing the transition from instruction to facilitation through emphasizing learner agency in harnessing CAQDAS packages appropriately for the chosen methodology and type of research project.

We hope that our experiences will support teachers in meeting the increasingly demanding needs of CAQDAS learners. *Five-level QDA* is our response to the need for an effective CAQDAS pedagogy, and is being developed as a full curriculum that can be adopted across methodologies, software packages and teaching modes (Woolf & Silver, *in press*). We will continue to critically reflect on this pedagogy and test its implementation in a variety of contexts in the years to come.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes

1. Established in 1994 the CAQDAS Networking Project provides independent information, advice, training and on-going support in the range of CAQDAS packages. <http://www.surrey.ac.uk/sociology/research/researchcentres/caqdas/>.
2. It is outside the scope of this article to illustrate *Translation Worksheets* but these are illustrated in full in Woolf and Silver (in press).
3. We refer to this as *Three-level QDA*, but it is beyond the scope of this article to go further into this approach to teaching CAQDAS (see Woolf, 2014; Woolf & Silver, in press).

Notes on contributors

Christina Silver, PhD, is the manager of the CAQDAS Networking Project (CNP) and co-director of the Day Courses in Social Research programme, both at the University of Surrey. Her research interests include the pedagogy of advanced methods teaching, uses and applications of CAQDAS technology, qualitative and mixed methods analysis, and the relationship between technology and methodology. She had a lead role in research conducted under the National Centre for Research Methods (NCRM) funded project Qualitative Innovations in CAQDAS (QUIC), and has contributed chapters on the use of software in qualitative data analysis to *The Oxford Handbook of Qualitative Research Methods* (Oxford, Oxford University Press, 2014); the *International Encyclopedia of Education* (Oxford, Elsevier, 2010); and the *Handbook of Qualitative Research in Psychology* (London, Sage, 2008). She is a co-author of *Using Software in Qualitative Research: A Step-by-Step Guide* (2nd Ed, London, Sage, 2014) and is a co-author of an upcoming textbook, *Five-level QDA: A method for learning to use QDA software powerfully*. She is a co-founder of Qualitative Data Analysis Services (QDAS), a consultancy providing services in qualitative and mixed-methods analysis.

Nicholas H. Woolf, PhD, is an independent qualitative research educator and a consultant in Santa Barbara, California, and most recently a visiting academic fellow at the University of Reading, UK. As a qualitative research consultant, he has conducted research and published in numerous fields including public health, management, education, and performance evaluation. His own research interests include the pedagogy of advanced qualitative methods teaching and the cognitive aspects of qualitative data analysis, and he is a co-author of an upcoming textbook, *Five-level QDA: A method for learning to use QDA software powerfully*.

References

- Ausubel, D. P., & Robinson, F. G. (1969). *School learning – An introduction to educational psychology*. New York, NY: Holt, Rinehart & Winston.
- Bransford, J. D., Sherwood, R. D., Hasselbring, T. S., Kinzer, C. K., & Williams, S. M. (1990). Anchored instruction: Why we need it and how technology can help. In D. Nix & R. Spiro (Eds.), *Cognition, education, and multimedia: Exploring ideas in high technology* (pp. 115–141). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Davis, H., & Krayner, A. (2014, July). *Embedding NVivo in postgraduate research training*. Paper presented at the 6th National Centre for Research Methods (NCRM) Research Methods Festival. University of Oxford, Oxford.
- di Gregorio, S. (2014, May). *Time for a new pedagogic approach to teaching CAQDAS*. Paper presented at CAQDAS 2014 Conference: Past, Present and Future: 25 years of CAQDAS. University of Surrey, Guildford.
- di Gregorio, S., & Davidson, J. (2008). *Qualitative research design for software users*. Maidenhead: Open University Press.
- Fielding, N., & Lee, R. M. (1991). *Using computers in qualitative research*. Newbury Park, CA: Sage.
- Friese, S. (2014). *Qualitative data analysis with ATLAS.ti*. London: Sage.
- Gaver, W. W. (1991, March). Technology affordances. In *Proceedings of the SIGCHI Conference on Human Factors In Computing Systems* (pp. 79–84). New York, NY: ACM.

- Gibbs, G. (2014). *Count: Developing STEM skills in qualitative research methods teaching and learning*. York: Higher Education Academy. Retrieved from <http://eprints.hud.ac.uk/22751/>
- Honey, P., & Mumford, A. (1982). *Manual of learning styles*. London: P Honey.
- Kaczynski, D., & Kelly, M. (2004, November). *Curriculum development for teaching qualitative data analysis online*. Paper presented at QualiIT2004: International Conference on Qualitative Research in IT & IT in Qualitative Research. Brisbane, Australia.
- Kilburn, D., Nind, M., & Wiles, R. (2014). Learning as researchers and teachers: The development of a pedagogical culture for social science research methods? *British Journal of Educational Studies*, 62, 191–207.
- Kolb, D. (1984). *Experiential learning: Experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice-Hall.
- Lave, J., & Wenger, E. (1991). *Situated learning*. Cambridge: Cambridge University Press.
- Lewins, A., & Silver, C. (2007). *Using software in qualitative research*. London: Sage.
- Luttwak, E. N. (2001). *Strategy: The logic of peace and war* (2nd ed.). Cambridge, MA: Harvard University Press.
- Mayer, R. E. (1979). Can advance organizers influence meaningful learning? *Review of Educational Research*, 49, 371–383.
- Nida, E. A., & Taber, C. R. (1969). *The theory and practice of translation*. Leiden: E. J. Brill.
- Paulus, T. M., Lester, J. N., & Dempster, P. G. (2014). *Digital tools for qualitative research*. Thousand Oaks, CA: Sage.
- Reigeluth, C. M. (1979). In search of a better way to organize instruction: The elaboration theory. *Journal of Instructional Development*, 2, 8–15.
- Silver, C. (2010). *Research design for longitudinal case-study project: Tracking the use of software in real projects using different methodologies*. Guildford: QUIC Briefing Paper, Qualitative Innovations in CAQDAS (QUIC), The CAQDAS Networking Project, University of Surrey, UK.
- Silver, C., & Lewins, A. (2014). *Using software in qualitative research: A step-by-step guide* (2nd ed.). London: Sage.
- Silver, C., & Rivers, C. (2014). *Learning from the learners: The role of technology acceptance and adoption theories in understanding researchers' early experiences with CAQDAS packages*. ATLAS.ti User Conference 2013: Fostering Dialog on Qualitative Methods, Technische Universität Berlin. Berlin: Universitätsverlag der TU Berlin. Retrieved from <http://nbn-resolving.de/urn:nbn:de:kobv:83-opus4-44300>
- Silver, C., & Rivers, C. B. (2014). The CAQDAS postgraduate learning model: An interplay between methodological awareness, analytic adeptness and technological proficiency. *International Journal of Social Research Methodology*.
- Smith, P. L., & Ragan, T. J. (2005). *Instructional design*. Hoboken, NJ: Wiley.
- Whitehead, A. N. (1929). *The aims of education and other essays*. New York, NY: The Free Press.
- Woolf, N. (2014). *Analytic strategies and analytic tactics*. Paper presented at the ATLAS.ti User Conference 2013: Fostering Dialog on Qualitative Methods, Technische Universität Berlin. Berlin: Universitätsverlag der TU Berlin. Retrieved from <http://nbn-resolving.de/urn:nbn:de:kobv:83-opus4-44159>
- Woolf, N. H., & Silver, C. (in press). *Five-level QDA: A method for learning to use QDA software powerfully*. Santa Barbara, CA: Woolf Publishing.

Copyright of International Journal of Social Research Methodology is the property of Routledge and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.