

Softwarevermitteltes Forschen, Lehren und Lernen mit der Dokumentarischen Methode

Schäffer, Burkhard; Klinge, Denise; Krämer, Franz

Veröffentlichungsversion / Published Version

Zeitschriftenartikel / journal article

Zur Verfügung gestellt in Kooperation mit / provided in cooperation with:

Verlag Barbara Budrich

Empfohlene Zitierung / Suggested Citation:

Schäffer, B., Klinge, D., & Krämer, F. (2020). Softwarevermitteltes Forschen, Lehren und Lernen mit der Dokumentarischen Methode. *Zeitschrift für Qualitative Forschung*, 21(2), 163-183. <https://doi.org/10.3224/zqf.v21i2.02>

Nutzungsbedingungen:

Dieser Text wird unter einer CC BY-SA Lizenz (Namensnennung-Weitergabe unter gleichen Bedingungen) zur Verfügung gestellt. Nähere Auskünfte zu den CC-Lizenzen finden Sie hier: <https://creativecommons.org/licenses/by-sa/4.0/deed.de>

Terms of use:

This document is made available under a CC BY-SA Licence (Attribution-ShareAlike). For more information see: <https://creativecommons.org/licenses/by-sa/4.0>

Burkhard Schäffer, Denise Klinge und Franz Krämer

Softwarevermitteltes Forschen, Lehren und Lernen mit der Dokumentarischen Methode

Software-mediated Research, Teaching and Learning with the Documentary Method

Zusammenfassung

Am Beispiel der Entwicklung von DokuMet QDA, einer neuen Software für die Auswertung mit der Dokumentarischen Methode wird in dem Beitrag empirisch und theoretisch rekonstruiert, wie methodisches und methodologisches Wissen in Software ‚eingeschrieben‘ wird. Die hierbei auftretenden ‚Übersetzungsprozesse‘ zwischen Methodenexpert*innen auf der einen und Programmierer*innen auf der anderen Seite werden empirisch anhand der Analyse unterschiedlicher Dokumente rekonstruiert, die bei dem Entwicklungsprozess anfallen. Folgende Fragen werden behandelt: Wie lässt sich (implizites) Methodenwissen in Software ‚übersetzen‘? Was geht dabei verloren und was kommt hinzu? Wie lässt sich also eine Methode ‚technisieren‘ und welche Folgen hat dies für Kontexte von Methodenausbildung? In den programmatischen Schlussfolgerungen wird argumentiert, dass der Forschungsprozess mittels Software intersubjektiv überprüfbarer, damit transparenter und so damit auch besser lehrbar gemacht werden kann.

Schlagerwörter: QDA-Software, Dokumentarische Methode, Rekonstruktive Sozialforschung, Methodenlehre, Methodenwissen, Medientheorie, Technikentwicklung

Abstract

Using the example of the development of DokuMet QDA, a new software for evaluation with the documentary method, the article empirically and theoretically reconstructs how methodological and methodical knowledge is ‘inscribed’ in software. The resulting ‘translation processes’ between method experts on the one hand and programmers on the other hand are reconstructed empirically on the basis of the analysis of different documents that arise during the development process. Among others, the following questions will be dealt with: How can (implicit) methodological knowledge be ‘translated’ into software? What is lost and what is added? So how can a method be ‘engineered’ and what consequences does this have for the contexts of method training? In the programmatic conclusions it is argued that the research process can be checked intersubjectively by means of software, thus making it more transparent and thus better teachable.

Keywords: QDA Software, Documentary Method, Qualitative Research, Intermediation of Research Methods, Interpretation Knowledge, Media Theory, Technology Development

1 Einleitung

Die Lehre von Methodenwissen und -können ist ein wichtiger Bestandteil universitärer Ausbildung. Im Bereich qualitativer Forschung werden in jüngerer Zeit zunehmend neue didaktische Formate wie z.B. virtuelle Forschungswerkstätten angeboten oder methodische Module auf Lernplattformen zum Selbststudium bereitgestellt (vgl. Brennecke, Schrader, Pfaff, Tervooren in diesem Themenschwerpunkt). Parallel hierzu hat sich spezialisierte Software für „qualitative data analysis“ (QDA) immer mehr als Forschungsunterstützung etabliert (etwa die Programme MaxQDA, Atlas.TI oder NVivo). Insgesamt scheint sich hier ein noch nicht genau abschätzbarer Wandel in den Praxen qualitativer Forschung und ihrer Vermittlung anzubahnen, den wir in diesem Artikel am Beispiel der eigenen Entwicklung einer neuen QDA-Software namens DokuMet QDA beschreiben wollen. Wir wollen aufzeigen, wie methodisches und methodologisches Wissen, das auf unseren Erfahrungen mit der Auswertungspraxis mit der Dokumentarischen Methode beruht, in die Software ‚eingeschrieben‘ wird und wie sich hierdurch auch die Methode verändert. Die ‚Übersetzungsprozesse‘ zwischen Methodenexpert*innen einerseits und den Programmierer*innen andererseits wollen wir empirisch anhand der Analyse unterschiedlicher Dokumente rekonstruieren, die bei dem Entwicklungsprozess entstanden sind. Die Rekonstruktion kann als exemplarisch für den durch die digitale Entwicklung ausgelösten Wandel von Wissensformen in Lehr-Lernkontexten angesehen werden. Denn auch im Bereich von Methodenvermittlung haben wir es häufig mit der Rekontextualisierung, Veränderung und manchmal auch mit dem Ersetzen analoger Handlungspraxen durch digitale Handlungsträgerschaft (vgl. Latour 2007) zu tun.

2 Die analoge Vermittlung und Aneignung von „konjunktivem“ und „kommunikativ-generalisiertem Methodenwissen“: aus Büchern, in Seminaren und Forschungswerkstätten

„Methodenwissen“ in einem allgemeinen Sinn erstreckt sich über Methoden, Methodologien, Grundlagen- und Gegenstandstheorien (vgl. Dörner/Schäffer 2012b) gleichermaßen. Jemand, der oder die allgemeines Methodenwissen besitzt, kann also beurteilen, wie eigene gegenstandsbezogene Fragen grundlagentheoretisch gerahmt sind und den Einsatz von Methoden methodologisch begründen. Dieses Wissen bewegt sich in den Begriffen der Dokumentarischen Methode auf einer kommunikativ-generalisierten Ebene (vgl. Bohnsack 2017, S. 84ff.), weshalb man hier auch von einem *kommunikativ-generalisierten Dimension des Methodenwissens* sprechen kann. Es handelt sich um ein theoretisches Wissen über etwas, nicht um ein Wissen, das in konkreten Handlungsvollzügen erworben wurde. Hiervon wollen wir eine Dimension des Methodenwissens abgrenzen, bei der es vor allem um ein handlungspraktisches Wissen beim Interpretieren empirischen Materials geht, also um ein in konkrete Praxisvollzüge eingebundenes *Methoden-*

können. In den Begriffen der Dokumentarischen Methode kann man hier auch von einem „konjunktivem Methodenwissen“ sprechen¹.

Mit Lehrbüchern zu qualitativen Methoden ist konjunktives Methodenwissen nur sehr eingeschränkt vermittelbar. So stehen alle Einführungswerke und Handbücher zu (einzelnen) qualitativen Methoden oder Methodologien (u.a. bspw. Bohnsack 2014; Reichertz 2016; Nohl 2013; Przyborski/Wohlraab-Sahr 2008; Schäffer/Dörner 2012a etc.) vor der Schwierigkeit, konjunktives Methodenwissen zu vermitteln. Bücher haben ohne Zweifel ihre Stärken in der Vermittlung der theoretischen Grundlagen, also dem kommunikativ-generalisierten Methodenwissen, bei der Vermittlung handlungspraktischer Arbeitsschritte sind ihre Defizite jedoch offensichtlich. Die Wahrscheinlichkeit, dass mittels der Lektüre von Methodenliteratur handlungsleitende Interpretationsroutinen erlernt werden, ist nach unserer Erfahrung relativ gering. Gleiches gilt auch für die Vermittlung in Form von Überblicksvorlesungen oder einführenden Seminaren: Sie sind ohne Zweifel notwendig und wichtig, um eine theoretische und generalisierte Wissensbasis aufzubauen, warum und wie man diese oder jene Methode in das eigene Forschungsdesign aufnimmt. Aber die Umsetzung in eine eigene Interpretationspraxis fällt Vielen schwer, so zumindest unsere Erfahrung aus einschlägigen Methodenseminaren.

Aufgrund der genannten Defizite von Büchern und herkömmlichem Seminaren bzw. Vorlesungen werden im Bereich der Vermittlung und Aneignung qualitativ-rekonstruktiver Methoden bislang „Forschungswerkstätten“ als Königsweg favorisiert (u.a. vgl. Bohnsack 2005, S. 76; Riemann/Schütze 1987; Riemann 2018; Schäffer 2010). Sie haben ihre Wurzeln in der Soziologie der Chicagoer Schule und sind in Deutschland vor allem durch Anselm Strauss (1991) bekannt gemacht und von Fritz Schütze und Gerhard Riemann (Riemann/Schütze 1987) etabliert worden. Wie der Werkstattbegriff nahelegt, wird in einer Forschungswerkstatt mit empirischen Materialien in unterschiedlichen Interpretationsstadien „gearbeitet“. Je nach Ausrichtung der Forschungswerkstatt werden entweder die „Rohdaten“ wie Transkripte von Gruppendiskussionen und Interviews oder Bilder und Videos etc. in der Gruppe interpretiert oder es wird auf der Basis von Interpretationsvorlagen von Mitgliedern der Forschungswerkstatt an dem Material gearbeitet, wie z.B. bei der Dokumentarischen Methode (vgl. Schäffer 2010). Eine Forschungswerkstatt lässt sich durchaus als eine Form des „situated learning“ (Lave/Wenger 1991) beschreiben, bei der die Novizen zunächst einige Sitzungen lang zuschauen können, wie es gemacht wird. Das dabei entstehende „Arbeitsbündnis“ (Hoffmann/Pokladek 2012) wird als eines beschrieben, das von einem „bescheidenen Meister“ geleitet wird, der „wissenschaftliche Orientierung“ gibt und ein „Arbeitsklima“ (S. 210) herstellt. Aus Sicht der hier vertretenen Perspektive auf Methodenwissen als konjunktives Wissen vom Interpretieren ist bei Forschungswerkstätten der mimetische Aspekt zentral: Novizen werden durch das Vormachen der „bescheidenden Meister*innen“ und der fortgeschrittenen Teilnehmenden in einem ‚Kollektiv der Interpretierenden‘ auf habituellem Wege in eine Forschungspraxis einsozialisiert.

Ein übergreifendes, multimodales Methodenwissen, das sowohl kommunikativ-generalisierte als auch konjunktive Dimensionen des Methodenwissens beinhaltet, also theoretische Wissensbestände mit ganz konkreten interpretativen Fähigkeiten verbindet, wird vermutlich nur in einer Kombination der drei geschilderten Dimensionen erworben: Das Lesen von Methodenliteratur, die diskursive Aufarbeitung in Vorlesungen und Seminaren und die handlungspraktische Ein-

übung in Forschungswerkstätten. Ein wichtiges Ziel hierbei ist, den Einsatz von Methoden nicht instrumentell zu verstehen, sondern als reflexiven Prozess, in dem konjunktive und kommunikative Wissensformen aufeinander bezogen werden. Hierbei kann sich sowohl der Forschungsgegenstand, als auch der/die Interpretierende verändern. Bei einer so verstandenen Beschäftigung mit Methodenwissen kommt es zu einem komplexen Prozess der Auseinandersetzung mit Welt und Selbst, der durchaus in Termini der (transformativen) Bildungstheorie verstanden werden kann (vgl. hierzu Nohl 2006). Allerdings wird ein solch anspruchsvolles Programm aufgrund curricularer Engführungen qualitativer Methodenausbildung in den wenigsten Fällen praktisch durchführbar sein.

Neben der Vermittlung von kommunikativ-generalisierbarem Methodenwissen durch Bücher, Seminare und Vorlesungen sowie von konjunktivem Interpretationswissen durch Forschungswerkstätten lassen sich auch QDA-Softwareprogramme (QDA = Qualitative Data Analysis) im Hinblick auf vermittelnde Dimensionen untersuchen. Es stellt sich die Frage, welche Wissensformen hierbei sowohl im Hinblick auf die kommunikativ-generalisierte Dimension des Methodenwissens im weiten, als auch auf die konjunktive Dimension des Interpretationswissens im engeren Sinne vermittelt werden. Stellt eine Interpretationspraxis ‚zusammen mit‘ QDA-Programmen, so die übergeordnete Frage, die wir hier nur anreißen, aber (noch) nicht beantworten können, eine eigene (neue?) Form des Interpretationshandelns dar, die neue Formen der Habitualisierung des Interpretationshandelns hervorbringen und damit auch neue Formen, dieses Wissen zu vermitteln?

Herkömmliche QDA bzw. CAQDAS (Computer Assisted Qualitative Data Analysis) Programme unterstützen den Prozess qualitativer Forschung auf unterschiedliche Art und Weise (vgl. Zhao et al. 2016; Humble 2015; Silver/Lewins 2014; Richards 2015; Rädiker/Kuckartz 2019). In diese ‚Unterstützung‘ sind mehr oder weniger deutliche pädagogische Implikationen eingeschrieben, die eine Praxis sui generis nahelegen und so zu neuen Formen habitualisierten Interpretationshandelns führen können. Diese unterliegen jedoch alle bestimmten Bedingungen von Technologie- und im Besonderen von Softwareentwicklung. Diese wollen wir im Folgenden näher untersuchen.

3 Die Entwicklung von (QDA-)Software und die Technisierung von Methodenwissen

Strukturiert wird jede Software zunächst einmal durch die Praktiken der Felder der Technikentwicklung und das dort sedimentierte Wissen. So üben Marktlogiken, Programmiersprachen, Designpraktiken und nicht zuletzt Nutzungsgewohnheiten wie z.B. „copy & paste“ oder die „Wischgeste“ Einfluss auf die Konstruktionsprozesse aus, die mit dem als Software zu formierenden Gegenstand, in unserem Fall: mit Methoden- und Interpretationswissen sowie deren Vermittlung zuerst einmal wenig zu tun haben (vgl. hierzu Klinge 2018). Auf dem Markt existierende QDA-Programme und ihre Funktionalitäten bieten darüber hinaus für weitere QDA-Entwicklungen den Referenzrahmen: Sie zeigen an, welche Funktionen technisch möglich sind und welche sich bisher bewährt haben. Dadurch,

dass Software allerdings nie abgeschlossen ist und in einer immerwährenden ‚Beta-Versionsförmigkeit‘ schwebt, in welcher Funktionen in der Praxis getestet und verändert werden, fließen auch kontinuierlich (sich ändernde) Annahmen und Modelle über Nutzer*innen und Gesellschaft als auch ‚Trends‘ ein (Manovich 2013, S. 29).

Aber nicht nur auf der Metaebene von Markt, Design, bisheriger Technologie und gesellschaftlichen Diskursen, sondern auch auf der Mikroebene der Entwicklungs- und Gestaltungsakteure wird das, was Software kann und soll, wesentlich bestimmt. Dabei sind wissensbezogen heterogene Gruppen in die QDA-Softwareentwicklung involviert (vgl. Klinge/Krämer/Schäffer 2020): Forscher*innen und Lehrende aus dem universitären Bereich, Informatiker*innen und Designer*innen aus dem Bereich der Produktentwicklung und nicht zuletzt auch Stakeholder, die für die Finanzierung sorgen. Sie müssen „eine gemeinsame Sprache“ (Furger/Heintz 1997, S. 536) finden, damit die Entwicklung gelingen kann.

Eine Form, der sich jedoch alle involvierten Akteure ‚beugen‘ müssen, ist Eindeutigkeit als technisch-funktionale Bedingung von Software: Das Wissen aller beteiligten Akteure muss in diesem Sinne „technisiert“, also vereinheitlicht werden (Degele 2002, S. 170-171). Die „Programmierbarkeit“ von Maschinen macht es erst möglich, dass eingegebene Daten verarbeitet und im Medium selbst prozessiert werden und dadurch Interaktivität zwischen Mensch und Programm bzw. Computer entstehen kann (Schelhowe 2016, S. 41).

Für die Konstruktion von QDA-Software bedeutet das zum einen, bisherige Praktiken der Vermittlung und Aneignung von kommunikativ-generalisiertem Methoden- und konjunktiven Interpretationswissen zu technisieren, aber auch (zukünftige) Nutzungs- und Umgangsweisen im Vollzug zu imaginieren bzw. zu antizipieren. Nicht ohne Grund spricht Schachtner (1993) von Softwareherstellung als „intermediäre[m] Erfahrungsraum“ (S. 30). Dabei muss die Komplexität von Welt auf die Strukturen formaler Computerlogik reduziert werden. Reduzieren meint buchstäblich ein „Rückführen“ der Interpretationspraxen qualitativ-rekonstruktiver Forschung auf die kleinsten Handlungseinheiten und diese auf ihre Formalisierbarkeit und digitale Rekonstituierbarkeit hin auszuleuchten und das Ergebnis in Programmcodes umzusetzen oder umsetzen zu lassen. Dabei trifft die Formalisierung und Übersetzung des Interpretationswissens im engeren und des Methodenwissens im weiteren Sinne qualitativ rekonstruktiver Forschung zwei Ebenen: Im Hintergrund prozessiert der Programmcode Daten im Sinne einer informatorischen Verarbeitung und im Vordergrund der Benutzeroberfläche der Nutzer*innen wird diese Verarbeitung grafisch dargestellt (vgl. Mayer 2009).

Als Zwischenfazit lässt sich festhalten: In QDA-Softwareentwicklungsprozessen wird Methodenwissen im weiten- und Interpretationswissen in engeren Sinne in die Logik dessen ‚übersetzt‘, was innerhalb des Programmierens von Maschinen und der Gestaltung von grafischen Interfaces möglich ist. Neben den Strukturierungsnotwendigkeiten bietet QDA-Software in ihrem fertigen Zustand aber auch ein Strukturierungspotenzial für neuartige Praxen des Interpretierens, die mit der herkömmlich verwendeten Metapher ‚Software als Werkzeug‘ nur unzureichend gefasst werden. Wir argumentieren nachfolgend, dass QDA-Programme insofern weder als neutrales Werkzeug, noch als eigenständige Methode aufgefasst werden können, sondern Teile eines neuartigen „Aufschreibesystems“ im Sinne Kittlers (2003) darstellen, in dem sich auch neue Formen der Vermittlung entwickeln.

4 QDA-Software als Werkzeug, Methode oder Aufschreibesystem?

Ob es sich bei QDA-Software um ein neutrales „Werkzeug“ („Tool“) handelt, wie z.B. Kuckartz als maßgeblicher Entwickler der Software MaxQDA lange Zeit behauptet (Kuckartz 2010, S. 13) hat², oder ob die Software auch den Forschungs- und damit den Erkenntnisprozess selbst verändert, ist eine Frage mit vielfältigen, u.a. medien- und erkenntnistheoretischen Implikationen, die ihrerseits Relevanz für methodologische Überlegungen entfalten. Die dichotome Unterscheidung von Software als Werkzeug oder als Methode findet sich implizit auch bei Woolf und Silver (2018), die zwischen „Analytischen Strategien“ und „Software Taktiken“ (S. 13-14) unterscheiden: Analytische Strategien betreffen die Entwicklung einer Fragestellung, die methodologische Begründung, die Methoden der Datengewinnung und die herangezogenen analytischen Methoden. Dagegen sind Software-Taktiken auf die konkrete Umsetzung bezogen, also auf das informatisch strukturierte Wie der Durchführung durch QDA-Software. Das Verhältnis zwischen Analyse-Strategien und Software-Taktiken beschreiben sie als widersprüchlich und differenzieren zwischen verschiedenen Formen, mit dem Widerspruch umzugehen. Als Königsweg erachten sie die „Transzendenz“ der Widersprüche zwischen analytischen Strategien und softwarebezogenen Taktiken: Es gehe darum, diese zueinander in Beziehung zu setzen mit dem Ziel der „Übersetzung“ bzw. der Transformation der Analyse-Strategien in Software-Taktiken. „Experten“, so die Autor*innen weiter, würden dies unbewusst die ganze Zeit tun, auch für die Lehre müsse man diese Übersetzungsprozesse aus der impliziten Ebene des Expertenwissens wieder herausholen.

Das Problem bei den vorgestellten Konzepten ist, dass man aus der Dichotomie von ‚Werkzeug versus Methode‘ nicht herauskommt. Konzipiert man dagegen Software von vornherein als „Aufschreibesystem“ im Sinne Friedrich Kittlers (2003), entgeht man dieser wenig fruchtbaren Dichotomie. Aus Kittlers medienmaterialistischer Perspektive generiert *jedes* „Aufschreibesystem“ grundsätzlich neue Formen des Denkens und der Bezugnahme auf Selbst und Welt: So hat das „Aufschreibesystem 1800“ (Kittler 2003) mit dem Buch als Zentralmedium und der klassischen Philosophie im Bunde ganzheitliche Vorstellungen über die *conditio humana* erzeugt (Subjekt- und Bildungsbegriff, Konzepte von „Aufklärung“, „Mündigkeit“ etc.), die dann vom Aufschreibesystem 1900 mit „Grammophonen, Film und Typewriter“ (1986) im Zusammenspiel mit Psychophysik und Psychoanalyse schonungslos dekonstruiert wurden. Bezogen nun auf die Frage, ob der Einsatz von QDA-Software im Prozess qualitativen Forschens etwas ändert, kann man mit Kittler antworten: Natürlich! Das „Aufschreibesystem 2000“ mit dem Computer als Zentralmedium, der im Hintergrund alles in Maschinensprache umwandelt, die mit Schrift im herkömmlichen Sinne nichts mehr zu tun hat, hat, wie alle Aufschreibesysteme vor ihm, einen massiven Einfluss auf das, was Menschen (denkend) schreiben und (schreibend) denken. In Konzepten, die Software und die davon ausgelöste Digitalisierung als etwas grundlegend Neues darstellen, dessen „Taktiken“ man mit humanen „Strategien“ entgegenzutreten habe, dokumentiert sich aus einer Kittler’schen Sicht (1986, 2003) nur eine gewisse Blindheit gegenüber der Medienabhängigkeit *jedigen* menschlichen Erkenntnisprozesses. „Aufschreibesysteme“ unterschiedlichster Art, wie Gänsefedern, Füllfederhalter,

Kugelschreiber, Bleistift und Papier, Bücher, Schreibmaschinen, Kameras, Tonband- und Kassettengeräte und später Powerpointdarstellungen, Schreibprogramme wie Word, Tabellentools zum Ordnen von Material u.ä. wurden von qualitativ Forschenden seit ihren jeweiligen Anfängen zum schreibenden Denken und denkendem Schreiben genutzt und auch in der Vermittlung von Methoden- und Interpretationswissen eingesetzt. Seit den 2000er und insbesondere den 2010er Jahren kommt es zu einer sprunghaften, mit der Digitalisierung der gesamten „Wertschöpfungskette“ qualitativer Forschung (von der Aufnahme, über die Interpretation bis zur Veröffentlichung) einhergehenden Entwicklung von Soft- und Hardware unterschiedlichster Provenienz (vgl. Schäffer 2021). Aus der hier favorisierten Perspektive ist es also nicht die Frage ‚ob‘, sondern allein ‚wie‘ und ‚eingebunden in welche anderen geistigen, sozialen und medientechnologischen Strömungen‘ Software qualitative Sozialforschung verändert.

Die Software-als-Aufschreibesystem-Perspektive hat auch Konsequenzen für die Frage nach den Möglichkeiten, Software beim Erlernen von Methoden- und Interpretationswissen einzusetzen. Nach unserem Dafürhalten agieren die meisten bislang verfügbaren QDA-Programme als Aufschreibesysteme mit recht einfachen Grundannahmen, die von Nutzer*innen beim Arbeiten mit der Software übernommen werden (müssen?). Im Gegensatz zu den Herangehensweisen hermeneutischer bzw. rekonstruierender Verfahren kommen bei der bisher auf dem Markt verfügbaren QDA-Software kodierende und damit im Kern inhaltsanalytisch vorgehende, das Material dekontextuierende Vorgehensweisen zum Einsatz;³ es geht um die Markierung von Textstellen und der (z.T. bereits automatisierten) Zuordnung von Codes zu diesen Textstellen.

Die spannende Frage ist, inwiefern auch rekonstruktives Methoden- und Interpretationswissen in Softwareprogrammen theoretisiert und umgesetzt werden und dann genauso wie derzeit vorfindliche Software die Vermittlung strukturieren kann – diesmal jedoch nicht am Paradigma des Kodierens, sondern am sequenziellen Interpretieren zusammenhängender Materialien orientiert. Welche Schritte der „didaktischen Reduktion und Rekonstruktion“ (vgl. Weinberg 1991) der Wissensform qualitativ-rekonstruktiven Methodenwissens im weiteren und Interpretationswissens im engeren Sinne sind hierfür notwendig?

5 Ergebnisse: Rekonstruktion der Entwicklungspraxis

Ein wesentlicher Schritt bei der Herstellung einer Software ist der Entwurf einer Struktur für die Organisation der Daten, die später einmal von der Software verarbeitet werden sollen. Es handelt sich um einen informatisch-technischen Ordnungsvorgang, den Entwickler*innen leisten müssen, und der voraussetzt, dass Klarheit darüber herrscht, was genau in dieser Datenstruktur abgebildet werden soll. Dieses Was betraf in unserem Fall einerseits die Darstellungsweisen bestimmter Interpretationsschritte der Dokumentarischen Methode und andererseits die Semantik dieser Interpretationsschritte.

Im Folgenden rekonstruieren wir unsere Entwicklungspraxis entlang der Frage, wie implizites Wissen von und explizites Wissen über die Dokumentarische Methode im Rahmen von Prozessen der Vereindeutigung in die Softwareanwendung eingeschrieben wurde (5.1). Die Vereindeutigungs- und Technisierungspro-

zesse waren allerdings nicht nur von methodisch-methodologischen Überlegungen geleitet, sie waren auch eingebettet in intermediäre Praktiken mit Entwickler*innen, Studierenden und Expert*innen der Dokumentarischen Methode. In einem zweiten Schritt fragen wir deshalb, wie sich diese unterschiedlichen Handlungslogiken in der Konstruktion der Software aufeinander bezogen haben. (5.2)

5.1 Prozesse der Formalisierung und Vereindeutigung

5.1.1 Vorgangsketten

Das Erfordernis einer konsistenten und formalisierten Semantik für die Herstellung von DokuMet QDA umfasst sowohl formale Abläufe der Interpretation mit der Dokumentarischen Methode als auch konkrete Inhalte einzelner Ablaufschritte. Die Formalisierung und Vereindeutigung von Inhalten vollzog sich über den gesamten Entwicklungsprozess hindurch mittels einer teils handschriftlichen, teils softwaregestützten Produktion von Diagrammen und Begriffsinventaren und wurde auch theoretisch ausformuliert (z.B. Schäffer 2020a), also mit Hilfe medialer Formate, die uns als Mittel des Nachdenkens über die Dokumentarische Methode dienten. Zu Beginn des Entwicklungsprozesses nutzten wir Vorgangskettendiagramme, die wir in der Präsentationssoftware PowerPoint erstellten und die im Entwicklungsprozess als Zwischenprodukte den Status von Denkmedien besaßen. Um hier Missverständnissen vorzubeugen: Eine solche Vorgangskette repräsentiert *nicht* den Interpretationsprozess als solches, sondern sie stellt eine ‚Leiter der Erkenntnis‘ dar, die man wegwerfen kann, sobald man sein Erkenntnisziel, die bessere Durchdringung des Prozesses, erreicht hat.

Bei den „Vorgangsketten“ handelt es sich um ein spezifisches Visualisierungsprinzip, das aus der Betriebswirtschaft stammt, und ursprünglich dazu gedacht war, Prozesse in Organisationen zu modellieren (vgl. Scheer 1992, S. 4). Es wurden Vorgangsketten für alle Verfahrensschritte des Arbeitens mit der Dokumentarischen Methode angelegt (u.a. für Transkription, Thematischer Verlauf, Formulierende und Reflektierende Interpretation, Fallzusammenfassung, Typenbildung). Abbildung 1 zeigt, wie wir die Reflektierende Interpretation (RI) einer Gruppendiskussion in eine Kette von diskreten Vorgängen aufgeschlüsselt haben.

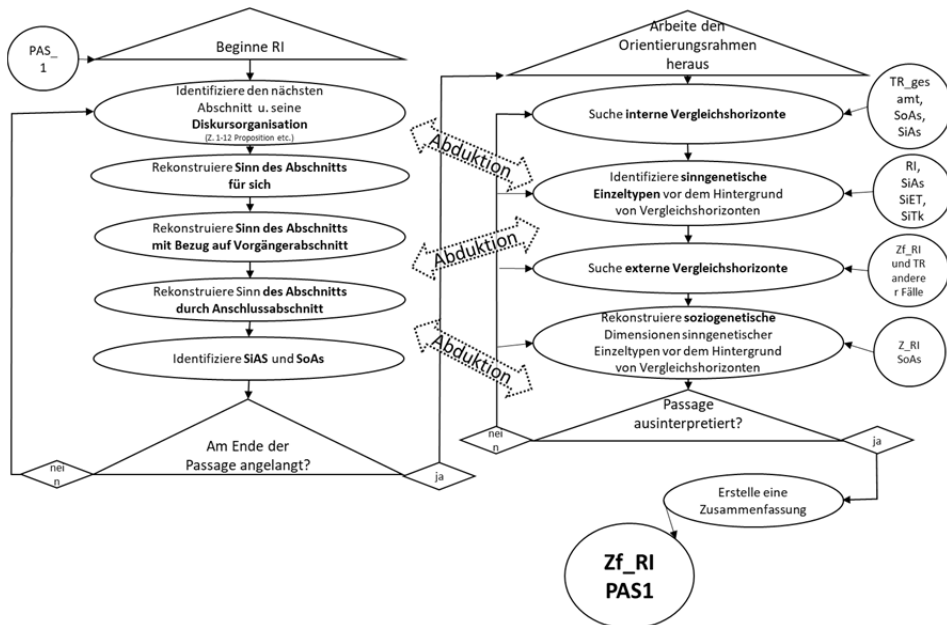


Abb. 1: Vorgangskette „Reflektierende Interpretation“

Dreiecke zeigen den Beginn oder das Ende einer Tätigkeit an, Ovale die konkrete Tätigkeit und Kreise das erstellte Produkt bzw. das Ausgangsprodukt. Abkürzungen: PAS_1: Passage 1; RI: Reflektierende Interpretation; SIAs: Sinngenetischer Aspekt; SiET: Sinngenetischer Einzeltypus; SiTk: Sinngenetische Typik; SoAs: Soziogenetischer Aspekt; TR_gesamt: gesamtes Transkript; Zf_RI: Zusammenfassende Reflektierende Interpretation (vgl. zur Terminologie Schäffer 2019).

Um uns den Prozess der Interpretation für die Softwarekonzeption zu vergegenwärtigen, haben wir zunächst idealtypisch die Schritte herausgearbeitet, die dem sequenzanalytischen Gebot Rechnung tragen, nicht vorzugreifen bei der Interpretation (linke Seite der Grafik). Allerdings entspricht die Trennung der Arbeitsschritte des schrittweisen Durchgehens durch ein Transkript (auf der linken Seite der Grafik) und der Herausarbeitung eines Orientierungsrahmens vor dem Hintergrund unterschiedlicher Vergleichshorizonte (rechts) nicht der gängigen, zu meist zirkulär zwischen Teilen (der Passage) und Ganzem (dem gesamten Text sowie anderen „Fällen“ als Vergleichshorizonten) oszillierenden Interpretationspraxis. Das handlungspraktische Interpretationswissen (s.o.) erfahrener Forscher*innen verweist vielmehr auf einen weitaus weniger formalisierten und formalisierbaren Prozess, bei dem Hypothesen über ‚Typisches‘ im Sinne ‚abduktiver Blitze‘ (Reichertz 1993) während der Beschäftigung mit einem Abschnitt eines Transkripts ‚aufscheinen‘ können, bevor man am Ende angeht. Dem haben wir (zu einem späteren Zeitpunkt) formal durch die Einfügung der Doppelpfeile mit der Aufschrift ‚Abduktion‘ Rechnung getragen. Prinzipiell sollte es beim Arbeiten mit der Software also möglich sein, so die Schlussfolgerung, gerade bei der Reflektierenden Interpretation auch schon Ideen und Schlussfolgerungen über ‚Typisches‘ aufzunehmen, die weit über die eine Textstelle hinausgehen. Hieraus entstand das Konzept der „sinn- und soziogenetischen Aspekte“, die schon wäh-

rend einer laufenden sequenziellen Interpretation einer Passage mit ausgewählten Textabschnitten verknüpft werden können (vgl. Schäffer 2020a, S. 76f.).

In dieser Phase der Softwareentwicklung erfolgte also zunächst einmal die Reformulierung, Graphisierung und Diagrammatisierung (vgl. Krämer 2016) der Methode in den Begriffen und Darstellungselementen der Vorgangsketten. Indem wir solche Formalisierungen mittels Verfremdungstechniken vornahmen, veranlassten wir die Explikation sowohl von Praxen mit der Dokumentarischen Methode als auch von Nicht-Wissen über sie: So wurde anhand von Rekonstruktionen eigener Interpretationspraxen zwar deutlicher, wie sich die herkömmliche (oft habitualisierte) Interpretationspraxis mit paper und pencil vollzieht. Gleichzeitig offenbarten sich allerdings auch „blinde Flecken“ in der methodischen Umsetzung der Dokumentarischen Methode, hier insbesondere, was den Umgang mit Abduktionen während der Reflektierenden Interpretation anbelangt (zu blinden Flecken der Dokumentarischen Methode in einem übergreifenden Sinne vgl. Dörner et al. 2019; zu Abduktionen: Schäffer 2020a, S. 72f.). Diese Leerstellen füllten dokumentarisch Forschende bislang u.E. mit Hilfe ihres über die Zeit erworbenen impliziten Wissens. Die methodologisch häufig als „Abduktion“ (Peirce 1979) repräsentierte Strategie, die zwar womöglich in Forschungswerkstätten inkorporiert wird und sich im Rahmen eigener Praxis zu einer eigenen Forschungshaltung verfestigen lässt, ließ sich anhand dessen softwarelogisch in Ansätzen über die Vergabe der sinn- und soziogenetischen Aspekte modellieren.

5.1.2 Begriffsarbeit

Weitere Vereindeutigungsprozesse vollzogen sich daneben im Medium von Begriffsarbeit. Die Dokumentarische Methode arbeitet insbesondere im Schritt der Reflektierenden Interpretation und dort in der Formalstrukturanalyse der Textsorten in narrativen Interviews sowie der Diskursorganisation in Gruppendiskussionen mit bestimmten Bezeichnungen (Proposition, Elaboration, Konklusion etc. vgl. Przyborski 2004). In einem weiteren Formalisierungsschritt überführten wir diese Bezeichnungen in ein aufgeschlüsseltes eindeutiges und somit programmierbares Begriffsinventar. Dabei haben wir für die Formalstrukturanalyse eine Art Satzbaukasten der Begriffe systematisiert, durch welchen sich einzelne Begriffe beliebig miteinander kombinieren lassen und sich somit ein sehr großes Spektrum von Bezeichnungen abbilden lässt (z.B. ‚Elaboration der Proposition von Cf in Interaktion mit Dm und Validierung durch Am‘). Wir haben damit Teile der Textsortenanalyse und der Reflektierenden Interpretation so vereindeutigt, dass sie im Sinne Degeles (2000) zum einen technisierbar, aber in der Software selbst auch neu kombinierbar sind, was auch bedeutet, dass die Elemente hier komplett von der Interpretationspraxis selbst zunächst einmal abstrahiert sind.

Mit Blick auf die Methodenvermittlung sind die Begriffe zur Diskursbezeichnung bei Gruppendiskussionen und zur Textsortentrennung bei narrativen Interviews im Programm mit anwählbaren Erklärungen hinterlegt, sodass sich im Interpretationsprozess ihr Sinn rekapitulieren lässt. Wenn ein*e Interpret*in z.B. in einer Reflektierenden Interpretation sich nicht sicher ist, wie eine „Proposition“ definiert ist, kann sie*er die F2 Taste drücken und bekommt die Information in einer scrollbaren Infobox angezeigt (vgl. den Kreis um die Infobox, Abb. 2):

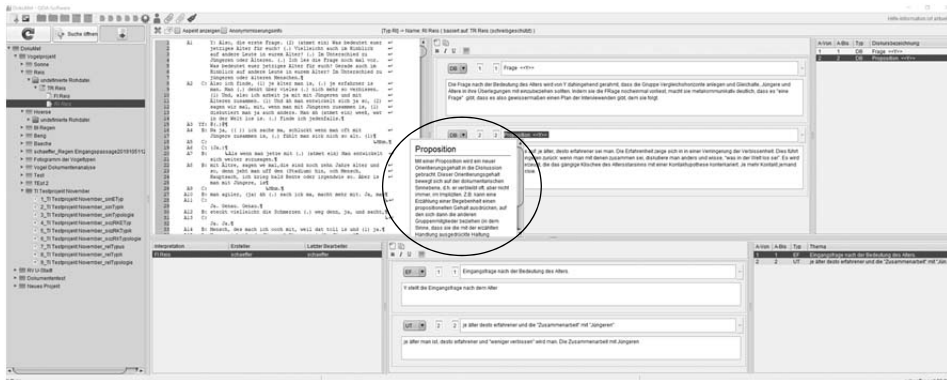


Abb. 2: Darstellung der scrollbaren Infoboxen

5.1.3 Nutzerinterfacegestaltung

Die weitere Formalisierungs- und Vereindeutigungsarbeit vollzog sich dann parallel zu ersten Überlegungen hinsichtlich des Nutzerinterfaces. Beides erfolgte im Medium der exzessiven Produktion von sogenannten Wireframes, einer häufig genutzten Methode des Interfacedesigns, bei der Funktion und Gestalt des Interfaces graphisch modelliert werden. Unsere teils handschriftlich, teils mit PowerPoint erstellten Wireframes dienen allerdings nicht nur als Medien der Reflektion, sondern auch als Kommunikationsmedien sowohl für die interne Kommunikation im Team der Konzeptentwickler*innen als auch für die Kommunikation mit Programmierer*innen über die Funktionen der Software. Die Visualisierungen haben uns im Sinne von Latours' Inskriptionen (1986) geholfen, Komplexität zu verdichten, eine Sichtbarkeit von Entitäten wie ‚Typenbildung‘ zu konstruieren und somit kommunizierbar zu machen. Hier eines der ersten, handschriftlich verfassten Frames (vgl. Abb.: 2).

Gerade zur Typenbildung entstanden verhältnismäßig viele Dokumente. Die Notwendigkeit, etwas auf eine bestimmte Weise darzustellen, rief nicht nur Abstimmungen über gegenstandsangemessene softwareinterne Darstellungsweisen von ‚semantischen Objekten‘ wie der Typenbildung hervor. Es gab ganz wesentlich auch Diskussionen darüber, welche Semantik hinter bestimmten Arten der Mediatisierung der Typenbildung – Kreuztabellen, Netzwerksansichten, rhizomartige Darstellungsformen etc. – steht, welche Bedeutung ein bestimmter methodischer Schritt also in methodologischer Hinsicht erlangt, wenn er mit Hilfe der Software auf eine ganz bestimmte (neue) Art und Weise ausgeführt wird. Während bei dem Interface „das Über sehen (sic) des Zeichens [...] notwendige Bedingung dafür [ist], dass Kommunikation möglich wird“ (Meyer 2002, S. 141), kam es für uns zunächst darauf an, das Zeichenhafte von Entwürfen und Ideen vor dem Hintergrund der bereits ausgearbeiteten Method(ologie) der Dokumentarischen Methode zu deuten und das Wie des Zeigevorgangs innerhalb des Interfaces, das wir kreiert hatten, methodologisch zu reflektieren.

Datei Ansicht Bearbeiten etc.

Name		tabellarische Übersicht		Gruppe Holz v. 20.10.13		Gruppe Holz		S.I.E.T: Nie wieder Kreis!	
SPI		Name	Datum	Typ (GD, Int, TB)	Typologie zugehörig	Z-RI, Passage Grotke		Gruppen Holz, Metall, Röhre	
TV			20.10.13	GD	a	Sinngebende Aspekte		Des Typens mit Kreis Kreis ist von Gruppe höhere Aktes gelag.	
TR		Metall	5.7.14	GD	a	Soziogenetische Aspekte		Des Gruppe Holz kommt das passiert auf die Punkt	
FI		Balken	3.5.15	GD	a				
RI		Brett	4.7.16	GD	c				
Z-RI		Stein	5.8.14	GD	b				
SIET		Hr. Wald	7.2.18	Int	b				
SIET		Fr. Wies	8.9.13	Int	b				
SITY		Hr. Blum	4.10.15	Int	b				
SITY		Fr. Offer	8.5.14	Int	a				
SoDi		TB Feste	7.3.15	TB	c				
SoTy		TB Feste	5.7.13	TB	d				
SoTy		TB feste	8.9.16	TB	a				
SoTy		TB feste	10.11.17	TB	e				

Info über Teil: Zugang + alle andere ...

Bislang erhobene Daten und Interpretationen

alle aktiv. Bei Praktiken offene Sichtweise 3 oder 4. Gibt es mehrere TBs, FIs, etc. oft sind hell davor neu

Lit + Links die sich auf Teil beziehen

Abb. 3: Beispiel Frühes Wireframe zur Typenbildung

Ein Statement, welches wir als Arbeitsergebnis dieser Prozesse festgehalten haben, war: „Nicht Komplexität reduzieren, sondern den Überblick über die Komplexität behalten“. Dies kann als eine Konsequenz gelesen werden, die wir aus dem unmöglichen Versuch gezogen haben, den Typenbildungsprozess in der QDA-Software auf wenige Schritte zu reduzieren, was dazu führte, die Typenbildung nicht in einem, sondern mehreren aufeinander aufbauenden Schritten (mittlerweile neun) in der Software auszuarbeiten. Folgende Abbildung (Abb. 4) zeigt einen Screenshot des Interfaces für die erste Dimension der sinngenetischen Typenbildung. In diesem zeigt die Software nach einer Suche nach „sinngenetischen Aspekten“ (SIAs) jene Fälle innerhalb einer Dimension an, bei denen während der RI Hinweise auf Typisierbares festgehalten worden sind. Dem/der Interpretierenden werden die Fälle und die entsprechenden Textstellen angezeigt, auf deren Basis er/sie dann entsprechende Verdichtungen als „sinngenetischer Einzeltypus“ (SiET) vornehmen kann.

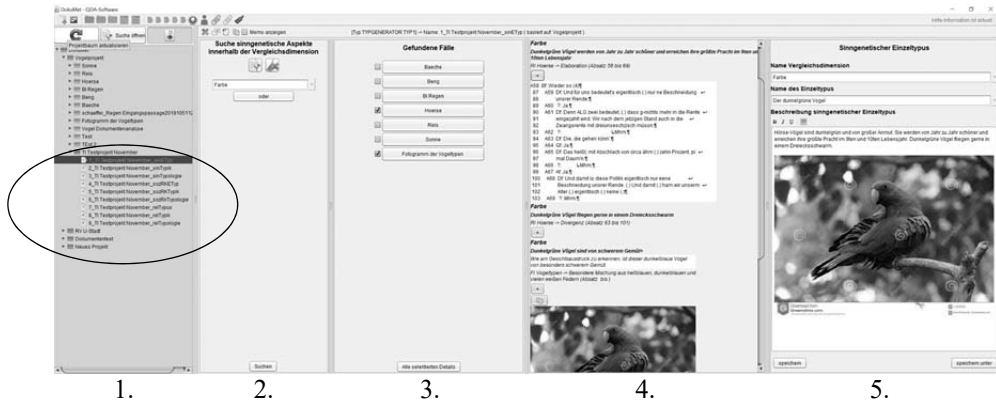


Abb. 4: Screenshot Interface Typenbildung DokuMet QDA

Links im Baum in der Spalte 1 ist die neunstufige Ausdifferenzierung des „Typengenerators“ zu erkennen (markiert durch umkreisten Bereich): je drei für die sinngenetische Typenbildung, drei für die soziogenetische Rekonstruktion dieser sinngenetischen Typen und drei für die relationale Typenbildung nach Nohl (2013). Alle drei sind intern in Typen, Typiken und Typologien ausdifferenziert (siehe hierzu Schäffer 2020a, S. 68f.). In der zweiten Spalte wird eine Suche innerhalb von Dimensionen ermöglicht, die man bei sinn- oder soziogenetischen Aspekten als Themen identifiziert hat, in der dritten Spalte werden Fälle zur Auswahl angeboten, in denen diese themenbezogenen Dimensionen vorkommen, in der vierten Spalte werden die Aspekte der dann ausgewählten Fälle dargestellt und es können wahlweise (durch den Plus-Button) die interpretierten Transkriptausschnitte oder die interpretierten Bilder, aber auch Dokumentenausschnitte (z.B. von Akten) angezeigt und dann in der letzten Spalte 5 in diesem ersten von neun „Typengeneratoren“ zu einem ersten „sinngenetischen Einzeltypus“, d.h. im Grunde einer ersten Hypothese über Typisches weiterverarbeitet werden.

Exkurs Typenbildende Interpretation

An diesem Beispiel zeigt sich, dass die Softwareentwicklung auch mit einer Ausdifferenzierung methodischer Schritte einherging und insofern ein Anregungs-, Reflexions- und Veränderungspotenzial entfaltetete, das wir so nicht erwartet hatten. Letztlich entstand ein Arbeitsschritt, den wir in Anlehnung an die Formulierende und Reflektierende Interpretation „Typenbildende Interpretation“ genannt haben (Schäffer 2020a, S. 71ff.): Hierunter sind Forschungshandlungen zu verstehen, „die mit dem abduktiven Prozess der Typenentdeckung, dem deduktiven Prozess der Typenbildung und dem induktiven Prozess der empirischen Absicherung sowie mit der theoretischen Verortung von Typen, Typiken und Typologien zu tun haben“ (a.a.O. S. 71). Man kann hier die Position vertreten, dass dies alles in der Reflektierenden Interpretation bereits geleistet wird. Allerdings entzieht sich dieser Prozess u.E. der methodischen Kontrolle. In der Literatur zur Typenbildung der Dokumentarischen Methode wird vor allem das beschrieben, was ty-

pisiert werden soll: „Orientierungsrahmen“ (Bohnsack et al. 2019) bzw. „Handlungsorientierungen“ (Nohl 2019). Dem wird hier auch uneingeschränkt zugestimmt. Aber der Weg dorthin, wie man von der sequenziellen Feinschrittigkeit einer Reflektierenden Interpretation zu Aussagen über Typisches kommt, bleibt tendenziell unexpliziert. Es wird zwar auf „Abduktion“ im Sinne Peirce³ Bezug genommen, allerdings ohne näher zu beschreiben, wie die Abduktionen so ‚verwertet‘ werden können, dass sie den Kern für eine Typisierung im sinn- oder soziogenetischen Sinne abgeben (Schäffer 2020a, S. 72f.). Die Idee, einen explizierbaren Arbeitsschritt der Typenbildenden Interpretation zu etablieren, entstand aufgrund dieser Leerstelle beim Prozedere der Dokumentarischen Methode, die erst bei der softwaretechnischen Rekonstruktion der Methodologie auffiel.

Nach unserem Dafürhalten wird die Typenbildung der Dokumentarischen Methode durch den Schritt der Typenbildenden Interpretation transparenter und besser intersubjektiv nachvollziehbar. Dabei werden erste typisierende Interpretationen, welche bislang in der habitualisierten Interpretationspraxis in der komparativen Analyse mehr oder weniger mitliefen, unter dem Vereindeutigungsdruck seitens der Softwarelogik systematisch expliziert. Die „Typenbildende Interpretation“ ist so gesehen kein neuer Interpretationsschritt, aber ein Teil bisheriger Interpretationspraxis, der sich bislang im Besonderen auf der Ebene des impliziten Interpretationswissens vollzog.

Insgesamt machen die aufgeführten Beispiele deutlich, dass unterschiedliche Logiken im Entwicklungsprozess letztlich zu einem Programm geführt haben, in dem die Logik der Praxis (Bourdieu 2015) der Dokumentarischen Methode aufgeht in einem „Hybridwesen“ im Sinne Latours (2007), bei dem der Wechsel des „Aufschreibesystems“ im Sinne Kittlers (2003) neue Möglichkeiten des (schreibenden) Denkens und (denkenden) Schreibens eröffnet, die letztendlich auch zu Readjustierungen in der Beschreibung des Gesamtprozesses führen.

5.2 Softwareherstellung als intermediärer Erfahrungsraum

Nachdem ein erster konkreter Prototyp der Software entstanden war, bildete dieser den „intermediären Erfahrungsraum“ (Schachtner 1993) für weitere Entwicklungsschritte. Wir kommunizierten fortan mit den Informatiker*innen in der Logik von Programmwürfen und berichteten über unsere Erfahrungen mit den Prototypen jeweils in Form von sogenannten ‚Buglists‘ (Fehlerlisten), was wiederum entsprechende Schleifen der Prototypenmodifikation vonseiten der Programmierer*innen nach sich zog. Die Reflexion des Wie der Zeigevorgänge im Interface und des datenbankgestützten Ordens von Komplexität vollzog sich in diesem Kontext auf einer Ebene technischen Funktionierens. Von Entwickler*innenseite wurde für Kommunikationszwecke im Laufe der Entwicklungsarbeit ein Kommunikationstool namens Jira⁴ zur Verfügung gestellt, in das wir einzelne Aufgaben (tasks⁵) einstellen konnten, die dann von den Informatiker*innen abgearbeitet wurden. Auf der anderen Seite versuchten die Entwickler*innen aber auch die Dokumentarische-Methode-Logik anhand von empirischem Material (Transkripte und Interpretationen), welche wir aus unserer Praxis zur Verfügung stellten, zu verstehen und in die Software zu implementieren. Es vollzog sich somit ein wiederum softwaregestützter Versuch der wechselseitigen Perspektivenverschränkung.

Es zeigten sich jedoch auch an einigen Stellen, u.a. anhand der Funktion des „Typengenerators“, also dem Arbeitsschritt der Typenbildenden Interpretation in der Software, die Schwierigkeit, eine gemeinsame Sprache zu finden: Unstimmigkeiten bestanden zum einen darin, dass Schritte der Typenbildenden Interpretation schon in dem vorhergehenden Analyseschritt implizit mitlaufen. Dieses in unserer Interpretationspraxis Implizite muss aber in der Softwarelogik explizit gemacht werden, um überhaupt verdichtetes ‚Material‘ für den Schritt der Typenbildung zu erzeugen. Die Lösung dessen gestaltete sich für uns wiederum in Formalisierungsschritten, hier zunächst in einer Segmentierung der Reflektierenden Interpretation (RI): Wie oben bereits im Hinblick auf die Vorgangsketten dargelegt (vgl. 5.5.1) wurde die Möglichkeit geschaffen, jederzeit sinn- und soziogenetische Aspekte (SiAs und SoAs) sowie allgemein theoretische Memos zu erzeugen (im Sinne „abduktiver Blitze“, vgl. Reichertz 1993, vgl. auch Abb. 1) die dann das Rohmaterial für den ‚Typengenerator‘ bildeten.

Den Vorschlag zur Typenbildenden Interpretation interpretierten die Programmierer*innen als zusätzliche nötige Funktion der ‚softwaregestützten Findung von Auswertungskategorien‘, die sie eigentlich den menschlichen Interpretierenden und nicht der Software zugerechnet hatten. Dieses unterschiedliche Verstehen der interpretierenden Akteure im und mit dem Programm musste dann in einem gemeinsamen Treffen ausgehandelt werden. Nicht nur zwischen uns als inhaltlichen Entwickler*innen und den Informatiker*innen musste eine gemeinsame Sprache gefunden werden, sondern es musste auch die in das Programm eingeschriebene Sprache mit der Community der Dokumentarischen Methode abgestimmt werden. Dies zeigte sich bspw. am Begriff des ‚Typengenerators‘, der zunächst, wie einige andere Begriffe, die für die Interpretationspraxis zu ‚technisch‘ anmuteten, verworfen wurde, aber sich aufgrund seiner Prägnanz (es ist der Programmteil, in dem die Typenbildende Interpretation ausgeführt wird) in der internen Kommunikation gehalten hat.

6 Rekonstruktion und Formalisierung des Methodenwissens in und durch Softwarekonstruktion – Von der Vereindeutigung zur Didaktisierung?

Die Rekonstruktion der Erfahrungen mit der QDA-Softwareentwicklung und insbesondere der Formalisierungsprozesse haben gezeigt, dass die Technisierung einer rekonstruktiven qualitativen Methode anderer Denkräume bedarf. So haben wir gleich zu Anfang gemerkt, dass wir uns der Praktiken anderer Disziplinen, wie des Designs oder der Wirtschaftsinformatik, bedienen müssen, um unsere teilweise impliziten Interpretationspraktiken abbildbar zu machen. Angelpunkte der Übersetzung unserer Erfahrungen in eine Technisierungsmöglichkeit waren häufig visuelle Konstruktionen: So konnten wir durch die „Vorgangsketten“ die Zeitlichkeit der Interpretationsschritte darstellen. Die Möglichkeit der Darstellung veranlasste uns dazu, darüber nachzudenken, inwiefern Zustände des Interpretationsprozesses definiert werden können und wann ein Zustand in einen an-

deren übergehen kann. Solche Zustände lassen sich etwa dadurch beschreiben, dass bestimmte Interpretationstätigkeiten im Gang sind oder abgeschlossen wurden, bspw. die Reflektierende Interpretation eines regelgeleitet ausgewählten Textabschnitts, und dass bestimmte Randbedingungen vorliegen, bspw. dass das Ende einer zuvor im Interpretationsprozess thematisch identifizierten Passage erreicht wurde oder dass bereits früher im Prozess bestimmte Interpretamente angefertigt wurden. Durch die Darstellung in „Wireframes“ konnten wir die Nutzung und Interaktion der Software imaginieren und die Visualisierung der typenbildenden Prozesse half uns dabei zu verstehen, welches implizite Wissen der Interpret*innen eigentlich eine Rolle spielt und was eben nicht expliziert ist. Wir konnten mit unserer Analyse auch zeigen, dass es sich bei der Formalisierung nicht nur um einen Übersetzungsprozess von Praxis in Software handelt, sondern von Praxis in das Intermedium der Visualisierung, welche dann über verschiedene Schritte in die Software gelangt. Dieses Intermedium der Visualisierung war für uns einerseits wesentliche Bedingung, um unsere Interpretationspraxis als diskrete Zustände eines Prozesses reformulieren und so auch blinde Flecken der Methodologie der Dokumentarischen Methode in diesem Kontext identifizieren zu können, und andererseits die Resultate dessen an die Informatiker*innen zu kommunizieren. Die ersten Prototypen der Software bildeten dann (neben den Visualisierungen) einen weiteren „intermediäre[n] Erfahrungsraum“ (Schachtner 1993), in dem sich weitere Entwicklungsschritte vollzogen. Angemerkt werden sollte bzgl. der Übersetzung bzw. Externalisierung des impliziten Interpretationswissens, dass es sich um *unser* Interpretationswissen handelt, welches wir alle auf ähnliche Art und Weise habitualisiert haben und welches sich nun in der Software verfestigt hat.

Für uns war die Entwicklung der Software insofern eine neue Art des Denkens über die Methode, als dass die Technisierung als höchst voraussetzungsvoller Prozess geholfen hat, „blinde“ oder zumindest nicht explizierte „Flecken“ (Dörner et al. 2019) der Dokumentarischen Methode aufzudecken – die sonst in der Interpretationspraxis habituell gefüllt werden, mit dem Nachteil potenzieller Exklusions- und Schließungsprozesse derjenigen, die nicht über diese Habitus verfügen – und für das Methodentraining durch kleinteiligere Schritte zu konkretisieren. In Bezug auf das in Kapitel 1 stark gemachte Interpretationswissen im engeren Sinne bedeutet dies, dass Novizen durch die Explikation der Schritte in der Software schneller ins Tun kommen. Denn wichtig für die Aneignung dieses Interpretationswissens ist der „pragmatische“ Vollzug; man muss zum Lernen etwas tun (Koch 2014, S. 42). Dabei ist es hilfreich, wenn der Interpretationsprozess nicht nur in Form einer nachzuvollziehenden und nachzuahmenden Praxis (wie in Forschungswerkstätten) oder in Form von Praxis-Rekonstruktionen (wie in Lehrbüchern) sichtbar wird, sondern als Abfolge von explizierten Zuständen, die die lernende Person zusammen mit der Software *selbst herstellen und ineinander übergehen lassen* kann.

Zum Umgang mit neuen Medien betont der Ansatz des Konnektivismus (Siemens 2005), dass Lernen in Netzwerken die Chancen des selbstgesteuerten und auch des selbstbestimmten Lernens und vor allem das Verstehen durch selbstständige Verknüpfungsmöglichkeiten ermöglicht. Auch könnte die Netzwerkstruktur des Interpretationswissens in Software weitere abduktive Blitze ermöglichen. Die Formalisierung des Methodenwissens und die Konstruktion (vermeintlich) eindeutiger Interpretationsschritte hat ein gewisses „Entstrukturierungspotential“ (Degele 2000, S. 68) hinsichtlich bisheriger Begrenztheiten. So ist die Me-

thodenvermittlung in und durch Software bspw. nicht an Zeit und Ort gebunden wie die Forschungswerkstatt und sie ist unmittelbar und potenziell auch individuell responsiv (etwa durch eingebaute Rückmeldungen an die Nutzenden). In Software ist es zudem möglich, alle bisherigen Aufschreibesysteme, wie Bücher, Vorlesungen, Radio etc. zu simulieren und in dieser Simulation neue Nutzungs- und Kombinationsweisen zu schaffen (vgl. Manovich 2013, S. 29-39). So können bspw. in QDA-Software Texte gelesen, farblich markiert, sortiert und Notizen dazu gemacht und ggf. die dazugehörigen Interviews als Sounddateien abgespielt werden etc. Die erste Generation der Entwickler*innen interpretierte u.a. deswegen Software lerntheoretisch als Unterstützung für Denkprozesse und Kreativität (vgl. ebd. S. 83-98). Daneben muss aber auch softwareförmiges Wissen eindeutig sein, was andere nicht vereindeutigbaren Formen des Wissens, wie Körperwissen, Intuition oder ästhetisch-sinnliches Erfahren außen vor lässt.

Weiterhin ist Software die Eigenschaften der „searchability“ (Manovich 2013, S. 114) und dafür notwendigerweise auch der „findability“ (ebd. S. 115) eingeschrieben, welche zum einen Datenstrukturen visualisierbar machen und zum anderen den/die Nutzer*in ermächtigen, mit gewaltigen Datenmengen kompetent umgehen zu können. Diese Eigenschaft der Software erinnert an den Zettelkasten von Niklas Luhmann, welcher durchaus als Denkmedium gerahmt werden kann, mit dem es möglich wird, zwischen großen Datenmengen Verknüpfungen herzustellen und in diesen Verknüpfungen das Denken (durchaus im Derrida'schen Sinne) zu externalisieren (Ahrens 2017).

Kop und Hill (2008) verweisen dabei vor allem die auf sich verändernde Rolle des Tutors, „where control is shifting from the tutor to an increasingly more autonomous learner“ (S. 11). Ob man hier nicht einer weiteren „Illusion von Autonomie“ (Meyer-Drawe 2000) aufsitzt, sei dahingestellt. Vermutlich werden mehr und mehr *hybride Strukturen der Vermittlung* in die Lehre Einzug halten: zwischen der face to face Vermittlung im Zwiegespräch, mimetischen Prozessen des Lernens durch Teilhabe an Forschungswerkstätten, dem Ausprobieren und Üben mit der Software und nicht zuletzt durch die Lektüre einschlägiger ein- und weiterführender Literatur. Aus erziehungstheoretischer Perspektive auf Lern- und Aneignungsprozesse ist eine nachhaltige Vermittlung nicht ohne eine „Zumutung von Handlungs- oder Lebensorientierungen“ (Nohl 2018, S. 122) und somit nicht gänzlich auf Autonomie basierend zu vollziehen.

Wir denken allerdings, dass die Software DokuMet-QDA auf jeden Fall eine wertvolle Ergänzung zu den eingangs dargestellten Vermittlungsmedien Buch, Seminar und Forschungswerkstatt darstellt. Dies vor allem deshalb, weil sie nach unseren ersten Erfahrungen mit Studierenden den Aufbau von Interpretationswissen im engeren Sinne fördert, da mit ihr das Interpretieren auf allen Ebenen kontinuierlich, methodisch kontrolliert und intersubjektiv nachvollziehbar geübt werden kann. Die Didaktisierung der Software sehen wir vor allem mit Hinblick auf die Rekonstruktion der Entwicklungsprozesse im Zwang der Vereindeutigung unseres Wissens als Lehrende und Forschende mit der Dokumentarischen Methode. Durch die Übersetzung unseres Methoden- und Lehrwissens in formal ausführbare Prozesse mussten wir unser implizites Interpretations- und Methodenwissen rekonstruieren und in kleinteilige aufeinanderfolgende Schritte explizieren, die in der Software aufgezeigt und simuliert werden und den Interpretationsvollzug unterstützen. Diese Schritte bieten nun Anschlussstellen für weitere didaktische Tools, welche in der Interaktivität des Mediums nutzbar gemacht werden können, wie Erklärungs-Fenster, die bei bestimmten Schritten aktiviert

werden, eine Assistenz, welche bestimmte Schritte vorschlägt etc. Die Vermittlung der Methodologie sehen wir dabei allerdings als eingeschränkt, dafür bedarf es weiterhin anderer Medien, allen voran das direkte Lehrgespräch zwischen Lernenden und Lehrenden.

Nichtsdestotrotz sind unsere Interpretation der Methode, unsere Erfahrungen mit der Vermittlung und unsere Interpretationspraxis sowie die Entwicklungspraxis einer bestimmten Gruppe von Informatiker*innen in die Software eingeschrieben. Wir haben gewissermaßen selbstreferentiell unser Wissen als „Wissenskulturen“ (Knorr-Cetina 2002) generiert und in der Software validiert. Es gilt, diese jetzt weiteren Expert*innen der Methode zur Verfügung zu stellen, um – nicht zuletzt in ausgiebigen „Betatests“ mit größeren Nutzer*innengruppen – unsere Umsetzung auf den Prüfstein anderer Praxen zu stellen.

Anmerkungen

- 1 Zum Methodenwissen „elaborierter“ Methoden (Reichertz 2007, S. 197-198) trägt zunächst die Einsicht bei, dass jegliches Interpretieren durch geeignete Verfahren die eigenen „impliziten alltäglichen Deutungsroutrinen“ (Soeffner 1998 zitiert nach Reichertz 2007, S. 198) reflektieren muss. Der Glaube, qualitative Methoden als Technik beliebig „anwenden“ zu können und die eigene „Standortgebundenheit“ bzw. „Seinsverbundenheit“ (Mannheim 1985, S. 227) nicht mit reflektieren zu müssen, ist kennzeichnend für sog. „ad-hoc-Methoden“ (Reichertz 2007, S. 197). Interpretierende verdoppeln, so die Kritik, die von Seiten der Dokumentarischen Methode vorgetragen wird, nur die „common sense“-Modelle der von ihnen befragten Akteure (vgl. Bohnsack 2013, S. 242-243).
- 2 Mittlerweile hat er diese These relativiert: Computer seien einfach nicht mehr mit Werkzeugen herkömmlicher Natur zu vergleichen, sondern „komplexe Algorithmenmaschinen“, „Software“ komme durchaus der Status einer „Methode“ zu (Rädiker/Kuckartz 2019, S. 10). Dieses Argument ist u.E. nur oberflächlich betrachtet richtig, denn Computer waren auch schon zu Zeiten der Entwicklung der ersten Programmgenerationen von MaxQDA im Jahr 1989 „komplexe Algorithmenmaschinen“.
- 3 So grenzt Kuckartz die Rolle von QDA-Software klar ein. Sie sei „nicht für solche Formen der Textanalyse konzipiert (...), die primär auf Textexegese ausgerichtet sind. Mikroskopische Analysen, die mit einer akribischen und zeitaufwändigen Zeile-für-Zeile Vorgehensweise arbeiten und Stunden oder Tage für die sorgfältige Analyse weniger Zeilen Text aufwenden, werden naturgemäß von Computerunterstützung nur wenig profitieren können“ (2010, S. 19).
- 4 <https://de.atlassian.com/software/jira>

Literatur

- Ahrens, S. (2017): Das Zettelkasten-Prinzip. Erfolgreich wissenschaftlich Schreiben und Studieren mit effektiven Notizen. Norderstedt.
- Bohnsack, R. (2005): Standards nicht-standardisierter Forschung in den Erziehungs- und Sozialwissenschaften. *Zeitschrift für Erziehungswissenschaft*, 8. Jg., H. 4, S. 63–81. https://doi.org/10.1007/978-3-322-80769-4_5
- Bohnsack, R. (2013): Typenbildung, Generalisierung und komparative Analyse: Grundprinzipien der Dokumentarischen Methode. In: Bohnsack, R./Nentwig-Gesemann, I./Nohl, A.-M. (Hrsg.): *Die Dokumentarische Methode und ihre Forschungspraxis. Grundlagen*

- qualitativer Sozialforschung. 3. Auflage Wiesbaden, S. 241–270.
https://doi.org/10.1007/978-3-531-19895-8_11
- Bohnsack, R. (2014): Rekonstruktive Sozialforschung. Einführung in qualitative Methoden. 9. Auflage Opladen.
- Bohnsack, R. (2017): Praxeologische Wissenssoziologie. Opladen/Toronto.
- Bohnsack, R./Hoffmann, N. F./Nentwig-Gesemann, I. (Hrsg.) (2019): Typenbildung und Dokumentarische Methode. Forschungspraxis und methodologische Grundlagen. Opladen.
<https://doi.org/10.2307/j.ctvdf047g>
- Bourdieu, P. (2015): Entwurf einer Theorie der Praxis. Frankfurt a.M.
<https://doi.org/10.3726/14247631X15269853499045>
- Degele, N. (2000): Informiertes Wissen: Eine Wissenssoziologie der computerisierten Gesellschaft. Frankfurt a.M./New York.
- Degele, N. (2002): Einführung in die Techniksoziologie. München.
- Dörner, O./Loos, P./Schäffer, B./Schondelmayer, A. (2019): Dokumentarische Methode: Triangulation und blinde Flecken. Opladen. <https://doi.org/10.3224/84742074>
- Furger, F./Heintz, B. (1997): Technologische Paradigmen und lokaler Kontext: Das Beispiel der EREMETH. In: Schweizerische Zeitschrift für Soziologie, 23. Jg., H. 3, S. 533–566.
- Hoffmann, B./Pokladek, G. (2012): Das spezielle Arbeitsbündnis in qualitativen Forschungswerkstätten. Merkmale und Schwierigkeiten aus der Perspektive der TeilnehmerInnen (The special work alliance in qualitative research workshops. Characteristics and difficulties from the perspective of the participants). In: Zeitschrift für Qualitative Sozialforschung, 10. Jg., H. 2, S. 197–217.
- Humble, Á. M. (2015): Guidance in the World of Computer-Assisted Qualitative Data Analysis Software (CAQDAS) Programs. Review Essay. In: Forum Qualitative Sozialforschung, 16. Jg., H. 2, Art. 22.
- Kittler, F. A. (1986): Grammophon, Film, Typewriter. Berlin. <https://doi.org/10.2307/778332>
- Kittler, F. A. (2003) [1985]: Aufschreibesysteme 1800-1900. München.
- Klinge, D. (2018): Die (implizite) Pädagogik von Self-Tracking: Handlungspraxis und Vermittlungsweisen der EntwicklerInnen im Spannungsfeld von Entrepreneurship, Technik und Design. In: Houben, D./Prietl, B. (Hrsg.): Digitale Gesellschaft. Datengesellschaft. Einsichten in die Datafizierung des Sozialen. 17. Auflage Bielefeld, S. 133–153.
<https://doi.org/10.14361/9783839439579-006>
- Klinge, D./Krämer, F./Schäffer, B. (2020): DIY-Science zwischen Markt, Wissenschaft und Zivilgesellschaft. Über die protoorganisationale Verfasstheit von „Quantified Self“-Initiativen. In: Schröer, A./Engel, N./Fahrenwald, C./Göhlich, M./Schröder, C./Weber, S. M. (Hrsg.): Organisation und Zivilgesellschaft. Wiesbaden, S. 167–177.
https://doi.org/10.1007/978-3-658-18005-8_16
- Knorr-Cetina, K. (2002): Wissenskulturen: Ein Vergleich naturwissenschaftlicher Wissensformen. Frankfurt a.M.
- Koch, L. (2014): Lernen und Wissen. In: Göhlich, M./Wulf, C./Zirfas, J. (Hrsg.): Pädagogische Theorien des Lernens. 2. Auflage Weinheim, S. 42–51.
- Kop, R./Hill, A. (2008): Connectivism: Learning theory of the future or vestige of the past? The International Review of Research in Open and Distance Learning, 9. Jg., H. 3, S. 1–13. <https://doi.org/10.19173/irrodl.v9i3.523>
- Krämer, S. (2016): Figuration, Anschauung, Erkenntnis. Grundlinien einer Diagrammatologie. Frankfurt a.M.
- Kuckartz, U. (2010): Einführung in die computergestützte Analyse qualitativer Daten. 3. Auflage Wiesbaden. <https://doi.org/10.1007/978-3-531-92126-6>
- Latour, B. (2007): Eine neue Soziologie für eine neue Gesellschaft: Einführung in die Akteur-Netzwerk-Theorie. Frankfurt a.M.
- Latour, B. (1986): Visualisation and Cognition: Drawing Things Together. In: Kuklick, H./Long, E. (Hrsg.): Knowledge and society. Studies in the sociology of culture past and present. 6. Auflage Greenwich/London, S. 1–40.
- Lave, J./Wenger, E. (1991): Situated learning: Legitimate peripheral participation. New York. <https://doi.org/10.1017/CBO9780511815355>

- Luhmann, N. (1995): *Die Kunst der Gesellschaft*. Frankfurt a.M.
- Mannheim, K. (1985) [1929]: *Ideologie und Utopie*. Bonn.
- Manovich, L. (2013): *Software Takes Command*. New York.
- Mayer, R. (2009): *Schnittstelle - Leerstell - Vermittlung: Reflexionen zum Verhältnis Medien, Technik, Bildung oder: Von der Tücke des Mediums und der Tücke des Subjekts*. In: Wimmer, M./Reichenbach, R./Pongratz, L. (Hrsg.): *Schriftenreihe der Kommission Bildungs- und Erziehungsphilosophie in der Deutschen Gesellschaft für Erziehungswissenschaft. Medien, Technik und Bildung*. Paderborn, S. 13–31.
https://doi.org/10.30965/9783657767410_003
- Meyer, T. (2002): *Interfaces, Medien, Bildung. Paradigmen einer pädagogischen Medientheorie*. Bielefeld.
- Meyer-Drawe, K. (2000): *Illusionen von Autonomie. Diesseits von Ohnmacht und Allmacht des Ich*. München.
- Nohl, A.-M. (2006): *Bildung und Spontaneität. Phasen biographischer Wandlungsprozesse in drei Lebensaltern – Empirische Rekonstruktionen und pragmatische Reflexionen*. Opladen. <https://doi.org/10.1007/978-3-658-01292-2>
- Nohl, A.-M. (2013): *Relationale Typenbildung und Mehrebenenvergleich: Neue Wege der dokumentarischen Methode*. Wiesbaden.
- Nohl, A.-M. (2018): *Zur intentionalen Struktur des Erziehens: Eine praxeologische Perspektive*. In: *Zeitschrift für Pädagogik*, 64. Jg., H. 1, S. 121–138.
- Nohl, A.-M. (2019): *Zur Bedeutung der relationalen Typenbildung für die Dokumentarische Methode*. In: Amling, S./Geimer, A./Schondelmayer, A.-C./Stützel, K./Thomsen, S. (Hrsg.): *Jahrbuch Dokumentarische Methode. Centrum für qualitative evaluations- und sozialforschung e.V. (ces)*, H. 1, S. 51–64.
- Peirce, C. S. (1979): *Collected papers of Charles Sanders Peirce*. In: Hartshorne, C./Weiss, P. (Hrsg.): *Principles of Philosophy and Elements of Logic*. Cambridge.
- Przyborski, A. (2004): *Gesprächsanalyse und dokumentarische Methode: Qualitative Auswertung von Gesprächen, Gruppendiskussionen und anderen Diskursen*. Wiesbaden. <https://doi.org/10.1007/978-3-531-90347-7>
- Przyborski, A./Wohlrab-Sahr, M. (2008): *Qualitative Sozialforschung. Ein Arbeitsbuch*. München.
- Rädiker, S./Kuckartz, U. (2019): *Analyse qualitativer Daten mit MAXQDA*. Wiesbaden. <https://doi.org/10.1007/978-3-658-22095-2>
- Reichertz, J. (2007): *Qualitative Sozialforschung – Ansprüche, Prämissen, Probleme*. In: *Erwägen, Wissen, Ethik*. 18. Jg., H. 2, S. 195–208.
- Reichertz, J. (2016): *Qualitative und interpretative Sozialforschung: Eine Einladung*. Wiesbaden. <https://doi.org/10.1007/978-3-658-13462-4>
- Richards, L. (2015): *Handling qualitative data: A practical guide*. 3. Auflage Los Angeles.
- Riemann, G. (2018): *Forschungswerkstatt*. In: Bohnsack, R./Geimer, A./Meuser, M. (Hrsg.): *Hauptbegriffe qualitativer Forschung*. 4. Auflage Opladen.
- Riemann, G./Schütze, F. (1987): *Some Notes on a Student Research Workshop on “Biography Analysis, Interaction Analysis, and Analysis of Social Worlds”*. In: *Biography and Society, of the International Sociological Association, Research Committee 38*, H. 8, S. 54–70.
- Schachtner, C. (1993): *Geistmaschine: Faszination und Provokation am Computer*. Frankfurt a.M.
- Schäffer, B. (2010): *Gruppendiskussionen lehren und lernen. Aspekte einer rekonstruktiven Didaktik qualitativer Forschung*. In: Bohnsack, R./Przyborski, A./Schäffer, B. (Hrsg.): *Das Gruppendiskussionsverfahren in der Forschungspraxis*. 2. Auflage Opladen, S. 285–299. <https://doi.org/10.2307/j.ctvddzp5w.20>
- Schäffer, B. (2020a): *Typenbildende Interpretation. Ein Beitrag zur methodischen Systematisierung der Typenbildung der Dokumentarischen Methode*. In: Ecarius, J./Schäffer, B. (Hrsg.): *Typenbildung und Theoriegenierung. Methoden und Methodologien qualitativer Bildungs- und Biographieforschung*. 2. Auflage Opladen, S. 65–88.
<https://doi.org/10.2307/j.ctvtxw2zx.6>

- Schäffer, B. (2021, i.E.): „Das Medium ist die Methode“. Zur Technikgeschichte qualitativer Methoden. In: Fuchs, T./Demmer, E./Wiezorek, C. (Hrsg.): Aufbrüche, Umbrüche, Abbrüche. Qualitative Bildungs- und Biographieforschung zwischen 1978 und 2018. Opladen.
- Schäffer, B./Dörner, O. (Hrsg.) (2012a): Handbuch qualitative Erwachsenenbildungs- und Weiterbildungsforschung. Opladen. <https://doi.org/10.2307/j.ctvddzk9g>
- Schäffer, B./Dörner, O. (2012b): Zum Verhältnis von Gegenstands- und Grundlagentheorien zu Methodologien und Methoden in der Erwachsenen- und Weiterbildungsforschung. In: Schäffer, B./Dörner, O. (Hrsg.): Handbuch qualitative Erwachsenenbildungs- und Weiterbildungsforschung. Opladen, S. 11-22. <https://doi.org/10.2307/j.ctvddzk9g.4>
- Scheer, A.-W. (1992): Architektur integrierter Informationssysteme. Grundlagen der Unternehmensmodellierung. 2. Auflage Berlin. <https://doi.org/10.1007/978-3-642-97403-8>
- Schelhowe, H. (2016): «Through the Interface» – Medienbildung in der digitalisierten Kultur. In: MedienPädagogik: Zeitschrift für Theorie und Praxis der Medienbildung, H. 25, S. 41–58. <https://doi.org/10.21240/mpaed/25/2016.10.27.X>
- Siemens, G. (2005): Connectivism: A Learning Theory for the Digital Age. International Journal of Instructional Technology and Distance Learning, 2. Jg., H. 1, S. 3–9.
- Silver, C./Lewins, A. (2014): Using Software in Qualitative Research: A Step-by-Step Guide. 2. Auflage Thousand Oaks. <https://doi.org/10.4135/9781473906907>
- Strauss, A. (1991): Grundlagen qualitativer Sozialforschung. München.
- Weinberg, J. (1991): Didaktische Reduktion und Rekonstruktion. In: Tietgens, H. (Hrsg.): Didaktische Dimensionen der Erwachsenenbildung. Pädagogische Arbeitsstelle, Deutscher Volkshochschulverband. Frankfurt a.M., S. 130–150.
- Woolf, N. H./Silver, C. (2018): Qualitative Analysis Using ATLAS.ti. The Five Level QDA Method. New York. <https://doi.org/10.4324/9781315181660>
- Zhao, P./Li, P./Ross, K./Dennis, B. (2016): Methodological tool or methodology? Beyond instrumentality and efficiency with qualitative data analysis software [49 paragraphs]. Forum Qualitative Sozialforschung, 2. Jg., H. 17, Art. 16.