

DokuMet QDA: Softwarevermitteltes Forschen, Lehren und Lernen mit der Dokumentarischen Methode*

Burkhard Schäffer, Denise Klinge, Franz Krämer

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. U.a. werden folgende Fragen 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 damit auch besser lehrbar gemacht werden kann.

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

DokuMet QDA: Software-mediated Research, Teaching and Learning with the Documentary Method Abstract

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 methodological 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, Reconstructive Social Research, teaching of 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 virtuelle Forschungswerkstätten angeboten oder methodische Module werden auf Lernplattformen zum Selbststudium bereitgestellt (vgl. Bennecke, Berith-Schrader, Pfaff, Tervoren im Band). 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 angefallen sind. Die Rekonstruktion kann als exemplarisch für den durch die digitale Entwicklung ausgelösten Wandel von Wissensformen in Lehr-Lernkontexten angesehen werden. Denn auch im

* Preprint. Eingereicht bei der ZQF, Zeitschrift für qualitative Sozialforschung.

Bereich von Methodenvermittlung haben wir es häufig mit der Rekontextualisierung, Veränderung und manchmal auch mit Ersetzung analoger Handlungspraxen durch digitale Handlungsträgerschaft (vgl. Latour 2007) zu tun.

2. Die analoge Vermittlung und Aneignung von „Methodenwissen“: aus Büchern, in Seminaren und Forschungswerkstätten

„Methodenwissen“ in einem weiten Sinn erstreckt sich über Methoden, Methodologien, Grundlagen- und Gegenstandstheorien (vgl. Dörner & Schäffer, 2012) gleichermaßen. Jemand kann also beurteilen, wie eigene gegenstandsbezogene Fragen grundlagentheoretisch gerahmt sind und den Einsatz von Methoden methodologisch begründen. In einem engeren Sinn, geht es vor allem aber um Interpretationskompetenz: Alle gute theoretische Rahmung, perfekte Samplebildung, Erhebung und Forschungsorganisation etc. bringt gar nichts, wenn es an Interpretationskompetenz mangelt. Zur Interpretationskompetenz von „elaborierten“ Methoden (die Reichertz 2007, S. 197-198 von „ad-hoc-Methoden“ abgrenzt) trägt entscheidend die Einsicht bei, dass jegliches Interpretieren durch geeignete Verfahren zunächst 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“ (ebd). Ohne eine so verstandene Interpretationskompetenz verdoppeln Interpretierende, so die Kritik, die auch von Seiten der Dokumentarischen Methode vorgetragen wird, nur die „common sense“-Modelle der von ihnen befragten Akteure (vgl. Bohnsack 2013, S. 242-243). Ein wichtiges Ziel jeder qualitativen Methodenausbildung sollte es insofern sein, Interpretationskompetenz zu fördern, also den Einsatz von Methoden nicht instrumentell zu verstehen, sondern als reflexiven Prozess, in dem sich der Gegenstand und auch der/die Interpretierende verändert, also durchaus als einen komplexen (Bildungs-)Prozess der Auseinandersetzung mit Welt und Selbst.

Mit Lehrbüchern zu qualitativen Methoden ist Interpretationskompetenz nur sehr eingeschränkt vermittelbar. So stehen alle Einführungswerke und Handbücher zu (einzelnen) qualitativen Methoden oder Methodologien (u.a. bspw. Bohnsack 2007, Reichertz 2016, Nohl 2013, Przyborski/Wohlraab-Sahr 2008; Schäffer/ Dörner 2012 etc.) vor der Schwierigkeit, dass es sich bei Methodenwissen um eine komplexe Wissensart handelt, die theoretische Wissensbestände mit ganz konkreten interpretativen Fähigkeiten, eben der Interpretationskompetenz verbindet. Bücher haben ohne Zweifel ihre Stärken in der Vermittlung der theoretischen Grundlagen, bei der Vermittlung handlungspraktischer Arbeitsschritte sind ihre Defizite jedoch offensichtlich. Die Wahrscheinlichkeit, dass mittels der Lektüre von Methodenliteratur handlungsleitende Interpretationsroutrinen erlernt werden, ist nach unserer Erfahrung relativ gering. Gleiches gilt auch für die „Vermittlung“ von Methodenwissen in Form von Überblicksvorlesungen oder einführenden Seminaren: Sie sind ohne Zweifel notwendig und wichtig, um eine theoretische und deskriptive Wissensbasis aufzubauen, warum und wie man diese oder jene Methode in sein Forschungsdesign aufnimmt. Aber Methodenwissen im oben definierten Sinne des interpretativen „Könnens“ wird hierbei sicherlich nicht erworben.

Aufgrund der genannten Defizite von Buch und herkömmlichem Seminar bzw. Vorlesung 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 nicht mit (fertigen) Beispielinterpretationen, sondern mit uninterpretierten empirischen Materialien (Gruppendiskussionen, Interviews, Bilder, Videos etc.) „gearbeitet“. Wenngleich die Ursprünge hierin nicht verankert sind, lässt sich eine Forschungswerkstatt durchaus als eine Form des „Situational Learning“ (Lave & Wenger 1991) beschreiben, bei der in Gruppen aus Anfänger*innen und Fortgeschrittenen 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 durch die Besprechung leitet, „wissenschaftliche Orientierung“ gibt und ein „Arbeitsklima“ (S. 210) herstellt. Forschungswerkstätten haben jedoch einen Nachteil: Sie sind sehr personal- und ressourcenintensiv. Insofern eignen sie sich eher für die Ausbildung in ausgewählten Spezialisierungen im Masterbereich und in postgradualen Studiengängen, nicht aber in grundständigen Veranstaltungen.

Neben der Vermittlung von Methodenwissen durch Bücher, Seminare und Forschungswerkstätten lassen sich auch Softwareprogramme im Hinblick auf vermittelnde Dimensionen untersuchen. Unter den Stichwörtern Qualitative Data Analysis (QDA) bzw. Computer Assisted Qualitative Data Analysis, (CAQDAS) auf den Markt gekommene Programme unterstützen den Prozess qualitativer Forschung auf unterschiedliche Art und Weise (vgl. Zhao u.a. 2016; Humble 2015; Silver & Lewins 2014; Richards 2015; Rädiker & Kuckartz 2019). In diese „Un-

terstützung‘ sind mehr oder weniger deutliche pädagogische Implikationen eingeschrieben. Diese unterliegen jedoch alle bestimmten Bedingungen und Zwängen von Technologie- und im Besonderen 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, Methodologie und pädagogischer 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 Nutzungspraxis 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 (siehe auch Klinge, Krämer & Schäffer 2019): 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 technische Bedingung des Funktionierens von Software: Das Wissen aller beteiligten Akteure muss ‚technisiert‘ 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 von Methodenwissen, -vermittlung und -aneignung 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 hier, die Interpretationsschritte qualitativ-rekonstruktiver Forschung auf kleinteilige Art und Weise auf ihre Formalisierbarkeit und digitale Repräsentierbarkeit hin auszuleuchten und das Ergebnis in Programmcode umzusetzen oder umsetzen zu lassen. Dabei trifft die Formalisierung und Übersetzung des Wissens und der Praxis 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 vereindeutigt und in die Logik dessen, was innerhalb des Programmierens von Maschinen und der Gestaltung von Interfaces möglich ist, ‚übersetzt‘. Neben den Strukturierungszwängen bietet QDA-Software in ihrem fertigen Zustand aber auch Strukturierungspotential. Wir argumentieren nachfolgend, dass QDA-Programme weder als Werkzeug, noch als Methode aufgefasst werden können, sondern neuartige ‚Aufschreibesysteme‘ im Sinne Kittlers (2003) darstellen und damit auch neue Formen der Vermittlung generieren.¹

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 sozialkonstruktivistisch-technikdeterministischen Dichotomie von ‚Werkzeug versus Methode‘ nicht herauskommt. Es läuft auf ein ‚Entweder-Oder‘ hinaus: Entweder hat man die Software im Griff oder sie macht mit einem, was sie will. 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 „Grammophon, 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 jetzt plötzlich Software und die davon ausgelöste Digitalisierung als etwas grundstürzend Neues, als das Andere schlechthin 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 jeglichen menschlichen Erkenntnisprozesses*. Aus der hier favorisierten Perspektive ist es also nicht die Frage ‚ob‘, sondern allein ‚wie‘ und ‚eingebunden in welche anderen geistigen Strömungen‘ Software qualitative Sozialforschung verändert (genauso, wie die Einführung von Textverarbeitungsprogrammen oder die Erfindung von Tonband und Kassettenrecorder Einfluss auf den Forschungsprozess genommen haben (vgl. hierzu Schäffer 2020)).

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 User*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 Kodes zu diesen Textstellen.

Die spannende Frage ist, ob auch rekonstruktives Methodenwissen bzw. ein genuin hermeneutisches Vorgehen 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 ist. Welche Schritte der „didaktischen Reduktion und Rekonstruktion“ (vgl. Weinberg, 1991) der Wissensform ‚qualitativ-rekonstruktiven Methodenwissens‘ 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 (4.1). Die Vereindeutigungs- und Technisierungsprozesse waren aller-

dings 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. (4.2)

4.1 Prozesse der Formalisierung und Vereindeutigung

4.1.1 Vorgangsketten

Die 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 in Form von Aufsätzen (Schäffer 2020) ausformuliert, also mit Hilfe medialer Formate, die uns als Mittel des Nachdenkens über die Dokumentarische Methode dienen. Zu Beginn des Entwicklungsprozesses nutzten wir als Denkmedium Vorgangskettendiagramme, die wir in der Präsentationssoftware PowerPoint erstellten. 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.

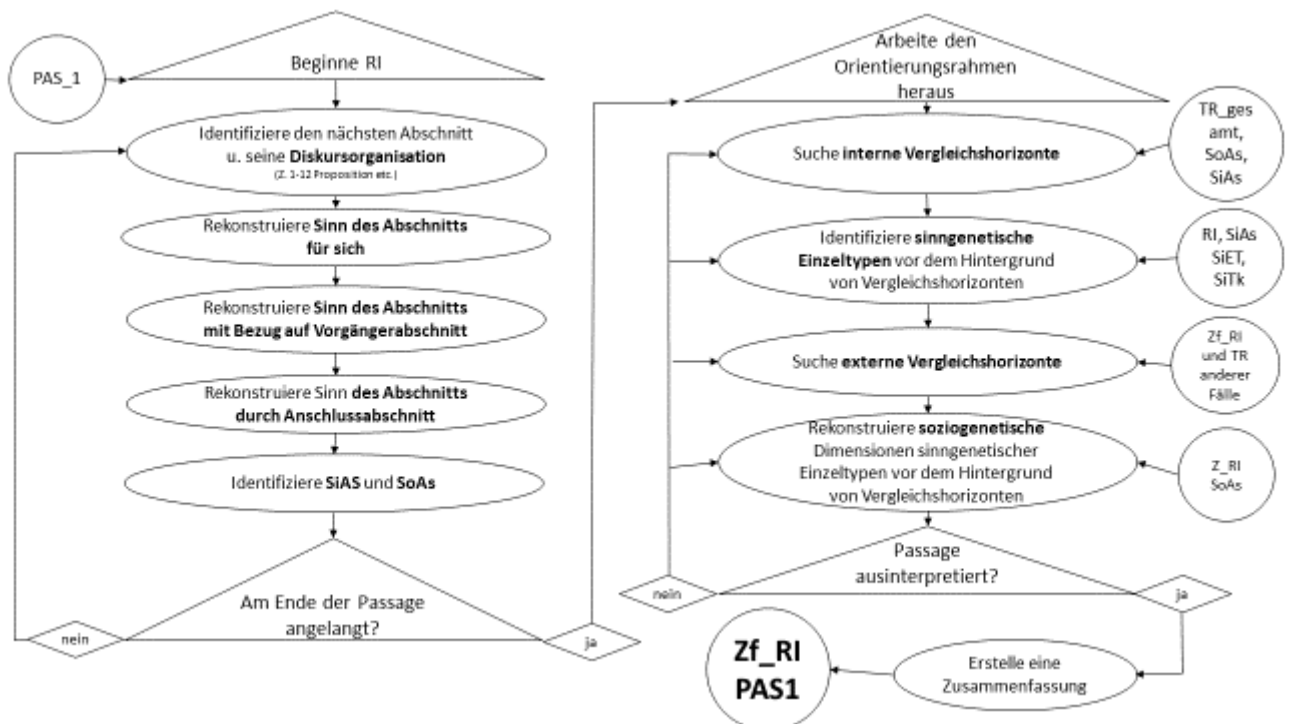


Abbildung 1: Vorgangskette „Reflektierende Interpretation“. Dreiecke zeigen den Beginn oder das Ende einer Tätigkeit an, Ovale die konkrete Tätigkeit und Kreise das erstellte Produkt bzw. das Ausgangsprodukt. Abkürzungen: PAS_1: Passage 1; RI: Reflektierende Interpretation; SiAs: Sinngenetischer Aspekt; SiET: Sinngenetischer Einzeltypus; SiTk: Sinngenetische Typik; SoAs: Soziogenetischer Aspekt; TR_gesamt: gesamtes Transkript; Zf_RI: Zusammenfassende Reflektierende Interpretation (vgl. zur Terminologie Schäffer 2020).

Für den Prozess der Interpretation konnten wir idealtypische Schritte herausarbeiten, die dem sequenzanalytischen Gebot Rechnung tragen, nicht vorzugreifen bei der Interpretation. Allerdings entspricht die Trennung der Arbeits-

schritte 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 Interpretationspraxis, bei der Hypothesen über ‚Typisches‘ im Sinne „abduktiver Blitze“ (Reichertz 1993) während der Beschäftigung mit einem Abschnitt eines Transkripts auftauchen. Genau hier ist die Abweichung zwischen herkömmlicher Interpretationspraxis mit paper und pencil und ihrer stark formalisierten Form mit einem Programm anzusiedeln. Über die Nutzung von Verfremdungstechniken wie den „Vorgangsketten“ (Scheer, 1992, S. 4) wurden konkrete Vorgehensweisen deutlicher, was einen hohen heuristischen Ertrag erzeugt (und später u.a. auch Konsequenzen bei der Konzeptualisierung des Schrittes der Typenbildung hatte, s.u).

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, was zur Identifizierung „blinder Flecken“ der Dokumentarischen Methode (vgl. hierzu Dörner/Schäfer et al. 2019) beitrug. Diese Leerstellen füllten dokumentarisch Forschende bislang offenbar 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 anhanddessen softwarelogisch jedoch nicht modellieren und musste im Explikationszwang der Softwarekonstruktion über die ganz konkreten Interpretationsschritte gefasst werden.

4.1.2 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 und der Diskursorganisation in Gruppendiskussionen mit bestimmten Bezeichnungen. In einem weiteren Formalisierungsschritt überführten wir die 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. Wir haben damit Teile der Textsortenanalyse und der Reflektierenden Interpretation syntaktisch reformuliert, und damit im Sinne Degeles (2000) die Objekte der Interpretation aus dem Zusammenhang genommen und in einer softwarelogischen Lesart und Repräsentationsmöglichkeit neu verknüpfbar zusammengefügt. Mit Blick auf die Methodenvermittlung sind die Begriffe im Programm mit anwählbaren Erklärungen hinterlegt, sodass sich im Interpretationsprozess ihr Sinn rekapitulieren lässt.

4.1.3 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 dienten 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.

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 eigentlich hinter bestimmten Arten der Mediatisierung der Typenbildung – Kreuztabellen, Netzwerkansichten, rhizomartige Darstellungsformen etc. – steckt, 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 kreierte hatten, methodologisch zu reflektieren.

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. 2) 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 ist. Dem/der Interpretierenden werden die Fälle und die entsprechenden Textstellen angezeigt, auf deren Basis er/sie dann entsprechende Verdichtungen als „sinngenetischer Einzelftypus“ (SiET) vornehmen kann.

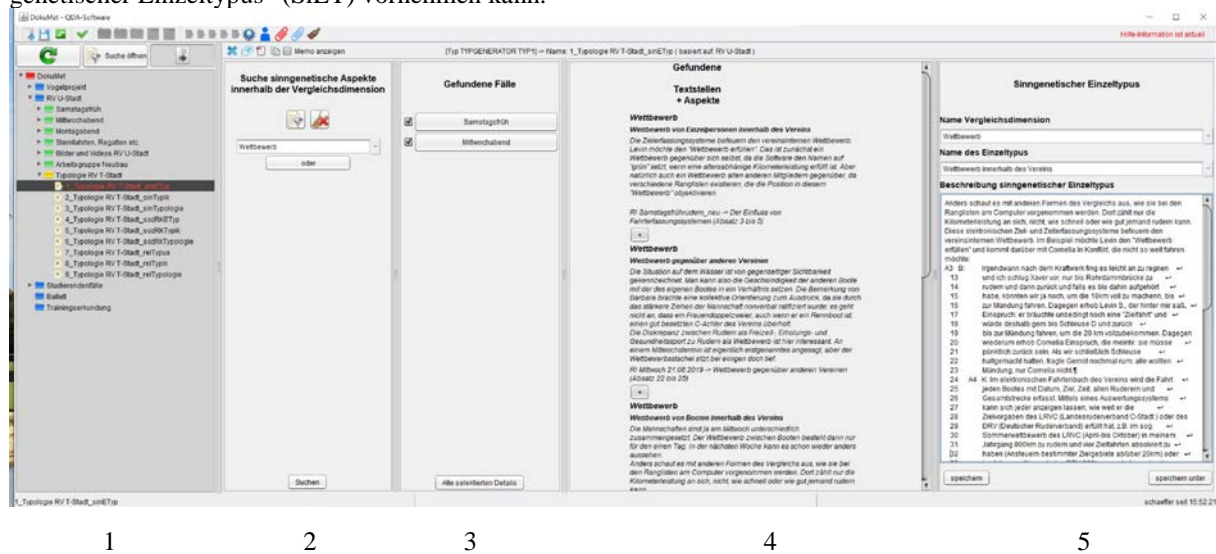


Abb. 2: Screenshot Interface Typenbildung DokuMet QDA

Links im Baum unten ist die neunstufige Ausdifferenzierung des „Typengenerators“ zu erkennen: 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. In der zweiten Spalte lässt sich nach sinngenetischen Aspekten suchen, deren Fälle man in der dritten Spalte zur Auswahl angeboten bekommt. In der vierten Spalte werden einem die Interpretamente und bei Bedarf auch die Textstellen im Originalmaterial angezeigt und in der fünften Spalte kann man schließlich aus den angebotenen Interpretamenten einen ersten Typen rekonstruieren. An diesem Beispiel zeigt sich, dass die Softwareentwicklung auch mit einer Ausdifferenzierung methodischer Schritte einhergeht und insofern ein Anrengungs-, Reflexions- und Veränderungspotenzial entfaltet, 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 2020). Im Vergleich zur Literatur zur Typenbildung der Dokumentarischen Methode (jüngst Bohnsack et al. 2018, Nohl 2019) ist die Typenbildung bei DokuMet QDA hierdurch nach unserem Dafürhalten transparenter und besser intersubjektiv vergleichbar geworden (vgl. hierzu Schäffer 2020).

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.

4.2 Softwareherstellung als intermediärer Erfahrungsraum

Nachdem ein erster konkreter Prototyp der Software herausgegeben wurde, bildete dieser quasi den „intermediären Erfahrungsraum“ (Schachtner 1993) für weitere Entwicklungsschritte. Wir kommunizierten fortan mit den Informatiker*innen nur noch in der Logik von Programmwürfen und berichteten über unsere Erfahrungen mit den Prototypen jeweils in Form von sogenannten ‚Buglists‘, 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 – wiederum formalisierendes – Kommunikationstool namens Jira³ zur Verfügung gestellt, in das wir einzelne Aufgaben („tasks“) 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 stellen, zu verstehen und in die Software zu implementieren. Es vollzog sich somit ein wiederum z.T. 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): Es wurde die Möglichkeit geschaffen, beinahe 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 (vgl. zu diesem Prozess Schäffer 2020a).

Den Vorschlag zur Typenbildenden Interpretation interpretierten die Programmierer*innen als zusätzliche Pflicht der Programmierung einer 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.

5. 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 anderen 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.

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 u.a. 2019) der Dokumentarischen Methode aufzudecken – die sonst in der Interpretationspraxis habituell gefüllt werden, mit dem Nachteil potentieller Exklusions- und Schließungsprozesse – und für das Methodentraining durch kleinteiligere Schritte zu konkretisieren. In Bezug auf die in Kapitel 1 stark gemachte Interpretationskompetenz bedeutet dies, dass Novizen durch die Explikation der Schritte in der Software schneller ins Tun kommen. Denn wichtig für die Aneignung von Interpretationskompetenz 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. Die Formalisierung des Methodenwissens und die Konstruktion (vermeintlich) eindeutiger Interpretationsschritte hat ein gewisses „Entstrukturierungspotential“

(Degele, 2000, S. 68) hinsichtlich bisheriger Begrenztheiten. So ist die Methodenvermittlung 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 ist es bspw. in QDA-Software möglich, Texte zu lesen, diese farblich zu manipulieren, zu sortieren, Notizen zu machen und ggf. die dazugehörigen Interviews als Sounddateien abzuspielen etc. Die erste Generation der Entwickler*innen interpretierte u.a. wegen dieser ‚Tausendsassa-Qualität‘ Software lerntheoretisch als Unterstützung für Denkprozesse und Kreativität (vgl. ebd. S. 83-98).

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 NutzerIn 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 & 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) aufsitze, sei dahingestellt. Wir denken dennoch, 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 Interpretationskompetenz 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 der 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 gewissermaßen im 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 die Schritte vorgibt etc. Die Vermittlung der Methodologie sehen wir dabei allerdings als eingeschränkt, dafür bedarf es weiterhin andere Medien.

Nichtsdestotrotz sind unsere Interpretation der Methode, unsere Erfahrungen mit der Vermittlung dieser und unsere Interpretationspraxis und die Entwicklungspraxis einer bestimmten Gruppe der 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 unsere Umsetzung auf den Prüfstein anderer Praxen zu stellen.

Endnoten

- 1 Dabei behaupten wir jedoch nicht, dass es bislang keinerlei mediale Unterstützung des Forschungsprozesses gegeben hätte: „Aufschreibesysteme“ unterschiedlichster Art, wie 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 genutzt und auch in der Vermittlung von Methodenwissen 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 2020).
- 2 Mittlerweile hat er diese These relativiert: Computer seien einfach nicht mehr mit Werkzeugen herkömmlicher Natur zu vergleichen, sondern „komplexe Algorithmenmaschinen“ (ebd.), „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).

Literatur

- Ahrens, S. (2017): Das Zettelkasten-Prinzip. Erfolgreich wissenschaftlich Schreiben und Studieren mit effektiven Notizen. Norderstedt: Books on Demand.
- Bohnsack, R. (2005). Standards nicht-standardisierter Forschung in den Erziehungs- und Sozialwissenschaften. *Zeitschrift für Erziehungswissenschaft*, 8(4), pp. 63–81.
- Bohnsack, R. (2007). *Rekonstruktive Sozialforschung. Einführung in qualitative Methoden*. Opladen: Barbara Budrich.
- Bohnsack, R. (2013): Typenbildung, Generalisierung und komparative Analyse: Grundprinzipien der Dokumentarischen Methode. In: Bohnsack, R./ Nentwig-Gesemann, I./ Nohl, A.-M. (Eds.): *Die Dokumentarische Methode und ihre Forschungspraxis. Grundlagen qualitativer Sozialforschung (3rd Edition)*. Wiesbaden, pp. 241-270.
- Bourdieu, P. (2015): *Entwurf einer Theorie der Praxis*. Frankfurt am Main: Suhrkamp.
- Degele, N. (2000). *Informiertes Wissen: Eine Wissenssoziologie der computerisierten Gesellschaft*. Frankfurt/Main, New York: Campus Verlag.
- Degele, N. (2002). *Einführung in die Techniksoziologie*. München: Fink.
- Dörner, Olaf/ Loos, Peter/ Schäffer, Burkhard/ Schondelmayer, Anne (2019): *Dokumentarische Methode: Triangulation und blinde Flecken*. Opladen: Barbara Budrich.
- Feyerabend, Paul K. (1995): *Against Method. Outline of an anarchistic theory of knowledge (deutsch: Wider den Methodenzwang. Skizze einer anarchistischen Erkenntnistheorie)*, Frankfurt am Main: Suhrkamp.
- Furger, F., & Heintz, B. (1997). Technologische Paradigmen und lokaler Kontext.: Das Beispiel der EREMETH. *Schweizerische Zeitschrift für Soziologie*, 23(3), pp. 533–566.
- Hall, S. (1980): Cultural Studies. Two paradigms. *Media, Culture and Society*, 1980(2), pp. 57-72.
- Haraway, D. (1988): Situated Knowledges. The Science Question in Feminism and the Privilege of Partial Perspective. *Feminist Studies*, 14(3), pp. 575-599.
- 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: *ZQF, Zeitschrift für Qualitative Sozialforschung*, 10(2)/2010, pp. 197-217.
- Humble, Á. M. (2015): Guidance in the World of Computer-Assisted Qualitative Data Analysis Software (CAQDAS) Programs. Review Essay. In: *FQS, Forum: Qualitative Social Research*, 16(2), Art. 22.
- Kittler, F. A. (2003)[1985]: *Aufschreibesysteme 1800-1900*. München: Wilhelm Fink Verlag.
- Kittler, F. A. (1986): *Grammophon, Film, Typewriter*. Berlin: Brinkmann und Bose.
- Klinge, D. (2018): Die (implizite) Pädagogik von Self-Tracking: Handlungspraxis und Vermittlungsweisen der EntwicklerInnen im Spannungsfeld von Entrepreneurship, Technik und Design. In D. Houben & B. Prietl (Eds.), *Digitale Gesellschaft: Vol. 17. Datengesellschaft. Einsichten in die Datafizierung des Sozialen*. Bielefeld: Transcript, S. 133–153.
- Klinge, D./ Krämer, F./ Schäffer, B. (2019): DIY-Science zwischen Markt, Wissenschaft und Zivilgesellschaft. Über die protoorganisationale Verfasstheit von „Quantified Self“-Initiativen. In: Schröer, Andreas/ Engel, Nicolaus/Fahrenwald, Claudia/ Göhlich, Michael/ Schröder, C./Weber, S.M. (Eds.): *Organisation und Zivilgesellschaft*. Wiesbaden: Springer Verlag für Sozialwissenschaften.
- Krämer, S. (2016): *Figuration, Anschauung, Erkenntnis. Grundlinien einer Diagrammatologie*. Frankfurt a.M.: Suhrkamp.
- Knorr-Cetina, K. (2002). *Wissenskulturen: Ein Vergleich naturwissenschaftlicher Wissensformen*. Frankfurt am Main: Suhrkamp.
- Koch, L. (2014). Lernen und Wissen. In M. Göhlich, C. Wulf, & J. Zirfas (Eds.), *Pädagogische Theorien des Lernens*. 2. Aufl. Weinheim: Beltz Juventa, pp. 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(3), pp. 1-13.
- Kuckartz, Udo (2010): Einführung in die computergestützte Analyse qualitativer Daten. Wiesbaden: VS Verlag, dritte, aktualisierte Auflage.
- Latour, B., & Woolgar, S. (1986). *Laboratory Life: The Construction of scientific facts*. Princeton: Princeton University Press.
- Latour, B. (2007). *Eine neue Soziologie für eine neue Gesellschaft: Einführung in die Akteur-Netzwerk-Theorie* (1st Edition). Frankfurt am Main: Suhrkamp.
- Latour, B. (1986). *Visualisation and Cognition: Drawing Things Together*. In H. Kuklick & E. Long (Eds.), *Knowledge and society. Studies in the sociology of culture past and present* (Vol. 6). Greenwich, Conn., London: Jai Press, pp. 1-40.
- Lave, J. & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. New York: Cambridge University Press.
- Luhmann, N. (1995): *Die Kunst der Gesellschaft*. Frankfurt a. M.: Suhrkamp.
- Mannheim, K.(1985 [1929]): *Ideologie und Utopie*. Bonn: Verlag Friedrich von Cohen.
- Manovich, L. (2013). *Software Takes Command*. New York: Bloomsbury.
- Mayer, R. (2009). Schnittstelle - Leerstell - Vermittlung: Reflexionen zum Verhältnis Medien, Technik, Bildung oder: Von der Tücke des Mediums und der Tücke des Subjekts. In M. Wimmer, R. Reichenbach, & L. Pongratz (Eds.), *Schriftenreihe der Kommission Bildungs- und Erziehungsphilosophie in der Deutschen Gesellschaft für Erziehungswissenschaft. Medien, Technik und Bildung*. Paderborn: F. Schöningh, pp. 13-31.
- Meyer, T. (2002): *Interfaces, Medien, Bildung. Paradigmen einer pädagogischen Medientheorie*. Bielefeld: transcript.
- Meyer-Drawe, Käte (2000): *Illusionen von Autonomie. Diesseits von Ohnmacht und Allmacht des Ich*. München: P. Kirchheim.
- Nohl, Arnd-Michael (2013): *Relationale Typenbildung und Mehrebenenvergleich: Neue Wege der dokumentarischen Methode*. Wiesbaden: VS-Verlag.
- Peirce, Charles S. (1979): *Collected papers of Charles Sanders Peirce*. 3. print. 8 Bände. Hartshorne, Charles; Weiss, Paul (Hrsg.). Cambridge, Mass.: Belknap Press of Harvard Univ. Press.
- Przyborski, Aglaja/Wohlrab-Sahr, Monika (2008): *Qualitative Sozialforschung. Ein Arbeitsbuch*. München: Oldenbourg.
- Rädiker, St./ Kuckartz, U. (2019): *Analyse qualitativer Daten mit MAXQDA*. Wiesbaden: Springer VS.
- Reichertz, Jo (2007): *Qualitative Sozialforschung – Ansprüche, Prämissen, Probleme*. In: *Erwägen, Wissen, Ethik*. Heft 2, 2007, S. 195-208.
- Reichertz, J. (2016): *Qualitative und interpretative Sozialforschung: Eine Einladung*. Wiesbaden: Springer VS.
- Richards, L. (2015): *Handling qualitative data: A practical guide* (3rd ed.). Los Angeles, CA: Sage.
- 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, Newsletter No. 8, July, of the International Sociological Association, Research Committee 38*, pp. 54–70
- Riemann, G. (2018): *Forschungswerkstatt*. In: Bohnsack, Ralf/ Geimer, Alexander/ Meuser, Michael (Hrsg.): *Hauptbegriffe qualitativer Forschung*, 4., vollst. Überarbeitete und erweiterte Auflage. Opladen: Verlag Barbara Budrich.
- Schachtner, C. (1993). *Geistmaschine: Faszination und Provokation am Computer* (1. Aufl). Suhrkamp-Taschenbuch Wissenschaft: Vol. 1072. Frankfurt am Main: Suhrkamp.
- Schäffer, Burkhard (2010): *Gruppendiskussionen lehren und lernen. Aspekte einer rekonstruktiven Didaktik qualitativer Forschung*. In: Bohnsack, Ralf/ Aglaja Przyborski/ Burkhard Schäffer: *Das Gruppendiskussionsverfahren in der Forschungspraxis*. Verlag Barbara Budrich, zweite vollständig überarbeitete und aktualisierte Auflage, S. 285-299.
- Schäffer, Burkhard (2020a): *Typenbildende Interpretation. Ein Beitrag zur methodischen Systematisierung der Typenbildung der Dokumentarischen Methode*. In: Ecarius, Jutta/ Schäffer, Burkhard Hrsg.)(2020): *Typenbildung und Theoriegenerierung. Methoden und Methodologien qualitativer Bildungs- und Biographieforschung*. Opladen: Verlag Barbara Budrich, 2te mit zusätzlichen Beiträgen ergänzte Auflage (i.E.).

- Schäffer, Burkhard (2020b): Zur Technikgeschichte qualitativer Methoden. In: Fuchs, xxxx
- Schäffer, Burkhard/ Olaf Dörner (Hrsg.) (2012): Handbuch qualitative Erwachsenenbildungs- und Weiterbildungsforschung. Opladen: Verlag Barbara Budrich.
- Schäffer, Burkhard/ Dörner, Olaf (2012): Zum Verhältnis von Gegenstands- und Grundlagentheorien zu Methodologien und Methoden in der Erwachsenen- und Weiterbildungsforschung. In: Schäffer, Burkhard/ Olaf Dörner (Hrsg.): Handbuch qualitative Erwachsenenbildungs- und Weiterbildungsforschung. Opladen: Verlag Barbara Budrich, S. 11-22.
- Scheer, August-Wilhelm (1992): Architektur integrierter Informationssysteme. Grundlagen der Unternehmensmodellierung, zweite, verbesserte Auflage, Berlin u.a.: Springer Verlag.
- Schelhowe, H. (2016). <Through the Interface> – Medienbildung in der digitalisierten Kultur. MedienPädagogik: Zeitschrift für Theorie und Praxis der Medienbildung. (25), pp. 41–58.
- Silver, Ch./ Lewins, A. (2014): Using Software in Qualitative Research: A Step-by-Step Guide (2nd ed.). Thousand Oaks, CA: Sage.
- Siemens, G. (2005). Connectivism: A Learning Theory for the Digital Age. International Journal of Instructional Technology and Distance Learning, 2(1), 3–9.
- Strauss, A. (1991): Grundlagen qualitativer Sozialforschung. München: Fink.
- Weinberg, J. (1991): Didaktische Reduktion und Rekonstruktion. In: Tietgens, Hans (Eds.). Didaktische Dimensionen der Erwachsenenbildung. Frankfurt: Pädagogische Arbeitsstelle, Deutscher Volkshochschulverband, pp. 130-150.
- Woolf, N., H./ Silver, Ch. (2018): Qualitative Analysis Using ATLAS.ti. The Five Level QDA Method. New York: Routledge.
- 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 / Forum: Qualitative Social Research, 2(17), Art. 16.

Die Autor*innen

Prof. Dr. Burkhard Schäffer ist Professor für Erwachsenenbildung/Weiterbildung am Institut für Bildungswissenschaft der Fakultät für Humanwissenschaften der Universität der Bundeswehr München.
 Forschungsschwerpunkte: Methoden und Methodologien qualitativer Erwachsenenbildungsforschung; Quantifizierung des Humanen; Alter(n)sbilder, lebenslanges Lernen und Weiterbildungsorientierungen; Generationsspezifische Medienpraxiskulturen und intergenerationelle Bildungsprozesse.
E-Mail: burkhard.schaeffer@unibw.de.

Dr. phil. Denise Klinge ist wissenschaftliche Mitarbeiterin am Institut für Bildungswissenschaft der Fakultät für Humanwissenschaften der Universität der Bundeswehr München.
 Forschungsschwerpunkte: Praktiken der Technikentwicklung, pädagogische Softwareforschung, dokumentarische Methode und elterliche Schulentscheidungen im Kontext von Übergangsforschung und Ungleichheit.
E-Mail: denise.klinge@unibw.de.

Dipl.-Päd. Franz Krämer ist Doktorand am Institut für Bildungswissenschaft der Fakultät für Humanwissenschaften der Universität der Bundeswehr München.
 Forschungsschwerpunkte: Selbstvermessungspraktiken in pädagogischer Perspektive, Bildungsprozesse Erwachsener, Digitalisierung qualitativer Method(ologi)en sowie Technik, Technologie und Technisierung in pädagogischer Perspektive.
E-Mail: franz.kraemer@unibw.de