

2. Untersuchungsgegenstand und Gliederung

Wie bereits die ersten Seiten zeigen, bewegt sich die vorliegende Untersuchung zwischen allgemeinen, abstrakteren Begriffen und der konkreten Empirie der Energiewirtschaft. Es geht um beides. Zum einen um eine allgemeine Problemstellung, die alle Industrien betrifft: industriespezifische Softwaregestaltung und welche Konzepte nützlich sind, um diese zu analysieren. Zum anderen um konkrete, industriespezifische Anwendungsbereiche von Software in der Energiewirtschaft, wie die Firmen der Branche Software gestalten und was die Besonderheiten der Branche sind, die sich in der Softwaregestaltung niederschlagen. Sowohl allgemeine als auch spezifisch energiewirtschaftliche Softwaregestaltung sind Forschungslücken und Gegenstand der Untersuchung.

2.1. Forschungsgegenstand und -fragen: Formen und Folgen von Softwaregestaltung

Ganz allgemein ist eine zentrale Eigenschaft von Computern, dass sie als universale Maschine programmiert werden können (und müssen). Alles Mögliche kann als Quellcode entwickelt werden: kleine Apps für Smartphones, Standardpakete zur Textverarbeitung, Programme für den automatisierten Börsenhandel oder Steuerungsprogramme für Atomkraftwerke. Die Folgen der Anwendung von Software auf Arbeit sind vielfältig untersucht (meist in Verbindung mit einer jeweils spezifischen Hardware): ob zu den Softwarelösungen der Gig-Ökonomie (vgl. Wood et al. 2019, Wu et al. 2019), Crowdworking (vgl. Gerber/Krzywdzinski 2019), ERP-Systemen (vgl. Hohlmann 2007, Walker 2016, Howcroft/Richardson 2012) oder in Call-Centern (vgl. Longen 2015). Auch zur Organisation von Softwareentwicklung bzw. der Arbeit der Programmierenden gibt es viel Forschung (vgl. Friedman/Cornford 1993, Barrett 2005, Upadhy 2009, Feuerstein 2013).

Beide Forschungsbereiche der Digitalisierung sind in Abbildung 1 schematisch dargestellt: der Arbeitsprozess der Anwendung einer Software (A) und jener ihrer Programmierung (C). Die vorliegende Arbeit untersucht (B), **die Arbeit zwischen Anwendung und Programmierung und wie sie organisiert ist**. Der Arbeitsprozess heißt **Softwaregestaltung**. Wie bereits in der Einleitung angesprochen, erfahren hier die Programmie-

renden, was zu programmieren ist. Diese drei in Abbildung 1 dargestellten Perspektiven auf die Digitalisierung sollten – so die These dieser Arbeit – voneinander unterschieden werden.

Abbildung 1: Drei Arbeitsprozesse digitaler Arbeit



Softwaregestaltung ist die Phase der Konzeption (bzw. Spezifikation oder Anforderungsaufnahme) der Softwareentwicklung. Abhängig vom Fall spielt dabei auch die Phase der Tests eine Rolle, weil in dieser die Anwendenden bzw. Testenden durch ihre Rückmeldungen Input für die Gestaltung geben. Fragen der Bedienoberfläche (Design) oder der Architektur, die Teil der Gestaltung von Software sein können, gehören hier nicht zum Kern des Arbeitsprozesses. Die Softwarearchitektur ist meist eine bereits entschiedene Ausgangsbedingung: z.B. welche Schnittstellen vorhanden sind, welche Einstellungsmöglichkeiten eine Software hat oder welche Erweiterungsmöglichkeiten eine Standardsoftware bietet. Das Software-Design in Sinne von z.B. anwendungsfreundlichen Oberflächen ist in den untersuchten Fallstudien kein gesonderter Arbeitsschritt mit den entsprechenden Spezialist:innen dafür.

Damit steht für die hier vorliegende Untersuchung im Mittelpunkt, dass Software eine gestaltungs offene Technologie ist, die je nach Umsetzung bspw. zu Formalisierung, Standardisierung, Überwachung oder Automatisierung führt. Das heißt, das Rationalisierungsziel des Softwareeinsatzes ist nicht per se die Formalisierung oder die Sammlung möglichst vieler Daten. Vielmehr geht es darum, für einen Anwendungsbereich die Entwicklungsmöglichkeiten der Technologie auszureizen.

Eine besondere Relevanz hat der Arbeitsprozess zwischen Anwendung und Programmierung bei industriespezifischer Software. Anders als bspw. bei Standardsoftwarepaketen wie LibreOffice oder MS Office ist der Wissenstransfer zwischen Industriefachleuten entscheidend. Organisationen überprüfen regelmäßig Entscheidungen über Make-or-Buy und müssen die Softwarelösungen kontinuierlich anpassen, weil sich die Regulierungen durch den Staat ändern oder um sich im Wettbewerb zu behaupten. Die Forschungsarbeit zeigt, dass der Ablauf industriespezifischer Softwaregestaltung ein eigenständiger, zentraler Arbeitsprozess der Digitalisierung ist und die Arbeit von Beschäftigten der jeweiligen Industrie (hier: der Energiewirtschaft) prägt. Die These ist, dass

für das Verständnis von heutiger Erwerbsarbeit das Verständnis der Softwaregestaltung neben anderen Einflüssen relevant ist: ob Finanzmärkte und der Shareholder Value (vgl. Windolf 2005, Vitols 2002, Dörre 2001, Höpner, 2003), der Position in der Wertschöpfungskette (vgl. Gereffi et al. 2005, Flecker/Meil, 2010, Mezihorak 2018), bestimmte Management- oder Organisationspraktiken (vgl. Gerst 2006, Boes et al. 2018) oder Produktionsmodelle (vgl. Kern/Schumann 1984, Herrigel 2010).

Die Softwareentwicklung als Ganzes ist Teil jeder Branche geworden. Jedes Unternehmen muss sich fragen, auf welcher Seite der Wissensgrenze zwischen Anwendung und Entwicklung es steht und an welchen Phasen des Entwicklungsprozesses – bestehend aus Spezifikation/Konzeption, Programmierung, Test, Support und Betrieb – es beteiligt sein will. Gleichzeitig ist für einen großen Teil der Nicht-Softwareunternehmen und ihrer industriespezifischen Fachabteilungen die Softwareentwicklung mit ihren Werkzeugen und Methoden zunächst einmal etwas Fremdes. Wie die Fallstudien zeigen werden, bedeutet branchenspezifische Software zu entwickeln eine Abkehr von der Fixierung auf eine nach Abteilungen gegliederte hierarchische Organisation und rein marktbasierte Beziehungen zu IT-Zulieferern wie IT-Dienstleistungsunternehmen (IT-DL) oder Softwarefirmen. Intern müssen Teams und Abteilungen (bspw. durch IT-Projekte) und extern unterschiedliche Organisationen kooperieren (bspw. bei der Zusammenarbeit mit Start-ups). Wenn gefragt wird: »Welche Ebene ist bestimmend für die Arbeit unter Bedingungen der Informatisierung: der Betrieb, das Unternehmen, das Projekt, das Netzwerk?« (Baukrowitz 2006: 82), so ist hier die Antwort: keine Ebene alleine, sondern die Softwaregestaltung, die sich über mehrere Ebenen erstrecken kann.

Es geht um die Konstellationen, in denen angewendete Software entsteht und welche Folgen das für die Anwendung hat: ob sie eine Softwarefirma entwickelt, ein eigenes Team an Programmierenden, eine Kooperation mehrerer Firmen; ob ein EVU eine Standardsoftware einsetzt, anpasst oder eine individuelle Lösung gestaltet. Es stehen zwei Kernfragen im Mittelpunkt der Analyse industriespezifischer Softwaregestaltung:

1. Kontrolle der Softwaregestaltung

- Wie kontrollieren Organisationen in unterschiedlichen Konstellationen den Arbeitsprozess der Softwaregestaltung? (Formen)
- Welche Auswirkungen hat die jeweilige Konstellation und Form der Softwaregestaltung für die Arbeit der Softwaregestaltenden? (Folgen)

2. Verhältnis zur Softwareanwendung

- Welche Auswirkungen hat die jeweilige Konstellation und Form der Softwaregestaltung für die Arbeitsgestaltung der Softwareanwendung? (Folgen)

Analytisch gesehen müssen, um die beiden Fragen nach den Formen und den Folgen untersuchen zu können, **a) Technikgestaltung, b) Arbeitsgestaltung und c) der Kontext beider gemeinsam untersucht werden**. Die reine Anwendung der Software ist nur insofern relevant, als sie diese Zusammenhänge erhellt. Das Methodenkapitel stellt unter 3.2.2 die zu den Kernfragen gehörenden Leitfragen für die Expert:inneninterviews vor.

Softwaregestaltung ist dabei weder primär eine Form von Innovation noch Teil einer Reorganisation. Auch wenn Softwaregestaltung innovativ im Sinn von »etwas Neues machen« ist, so liegt doch der Fokus auf den Gemeinsamkeiten mit anderen Arbeitsprozessen. In der Praxis nehmen Beschäftigte den Bereich nicht als gesonderten Innovationsbereich wahr: Programmierende, Anforderungsmanagende oder IT-Projektleitende sind in den untersuchten Fällen nicht allesamt in einer Innovations- oder F&E-Abteilung angesiedelt (höchstens einzelne Beteiligte). Auch wenn Softwaregestaltung mit organisatorischen Veränderungen einhergehen kann, ist dies in den untersuchten Fällen nicht der primäre Zweck der Software. Genauso wenig wie Innovationen nicht das primäre Ziel sind und in den untersuchten Fällen die Softwaregestaltung nicht per se als »innovativ« gilt – eher die verwendeten Methoden wie z.B. Scrum. Software ist eine schlichte Notwendigkeit, um das Tagesgeschäft (auch in Zukunft) effizient zu bewältigen.

Die analytische Perspektive auf den Arbeitsprozess fokussiert die Frage, womit Firmen und ihre Beschäftigten konfrontiert sind, wenn sie Software gestalten. Einerseits will das Management die Effizienz steigern, andererseits müssen sich Beschäftigte auf neue Methoden wie Scrum einlassen und unterschiedliche Abteilungen und Organisationen miteinander kooperieren. Untersuchungsgegenstand sind dabei einzelne Gestaltungsprozesse von Software(paketen) und nicht eine gesamte IT-Landschaft (aus einer Vielzahl an Softwarepaketen). Zudem geht es beim Arbeitsprozess um die kleinen, iterativen Schritte, zu denen regelmäßige Treffen gehören, um Anforderungen aufzunehmen, oder auch darum, den Anwendenden regelmäßige Updates der Software zur Verfügung zu stellen. Das Ziel ist, die Unterschiede und Gemeinsamkeiten der Formen der Softwaregestaltung in den unterschiedlichen Fallstudien inklusive der jeweiligen Folgen herauszuarbeiten.

2.2. Vorgehen: empirische, qualitativ-explorative Untersuchung der Energiewirtschaft

Die oben genannten Fragen werden **qualitativ-explorativ** erforscht (Näheres dazu im Methodenkapitel). Als empirischer Gegenstand fungiert die Energiewirtschaft und die dort betriebene industriespezifische Softwaregestaltung. Diese Branche ist nicht nur deshalb von Interesse, weil sie in der Soziologie kaum erforscht ist, sondern auch wegen ihrer jüngeren Liberalisierungsgeschichte, ihrer regulierten und unregulierten Bereiche: Die Fallstudien zeigen, dass das Folgen für die Softwaregestaltung hat, denn z.B. ist es im regulierten Netzbereich für die Unternehmen einfacher, sich auf einen Standard zu einigen. Im unregulierten Bereich der Energielieferung, in dem Wettbewerb herrscht, gestalten EVU häufiger individuelle Software, z.B. um ihrer Kundschaft besondere Angebote machen zu können. Durch die vielen und vielfältigen privaten, öffentlichen, großen und kleinen Unternehmen in der Energiewirtschaft gibt es einerseits einen großen Markt für Standardsoftware und andererseits Möglichkeiten der Kooperation zwischen den EVU. Da der Energiesektor insgesamt nicht zu den Vorreitern der Digitalisierung gehört, prallen *digital natives* und *digital immigrants* besonders heftig und ersichtlich aufeinander. Software innerhalb der EVU zu gestalten, ist in so einer Konstellation eine be-

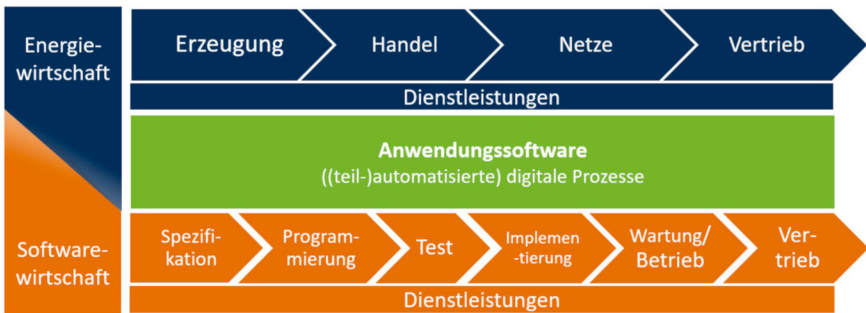
sondere Herausforderung. Zuletzt dient die Energiewirtschaft als Untersuchungsfeld, weil der Verfasser selbst als IT-Berater in der Branche tätig war.

Die vielfältigen und komplexen Anwendungsbereiche von Software in dieser Branche stellen eine Herausforderung für die Softwaregestaltung dar. Umfangreiche Wissensbestände müssen in Software übersetzt werden.

- Sei es technisches Wissen über Kraftwerke oder Netze: um sie zu steuern, zu überwachen oder zu warten. Will ein EVU bspw. seine Netz-Monteur:innen mit mobilen Apps und Drohnen ausstatten, ist ein intensiver Wissensaustausch mit ihnen über deren Arbeitsweise und die technischen Spezifika der Netzwartung notwendig, damit die Programmierenden wissen, was sie zu tun haben.
- Sei es regulatorisches Wissen über Gesetze und Verordnungen, welche den automatisierten Datenaustausch zwischen den Marktteilnehmenden regeln, damit bspw. die Stromkundschaft einfach per Internet das stromanbietende Unternehmen wechseln kann. Jede:r kann einen Blick auf die eigene Stromrechnung werfen, um zu sehen, was dort alles an Daten stehen muss: Abgaben, Umlagen, Steuern, technische Angaben, der Strommix etc. Oder jene Gesetze und Verordnung, welche regeln, wie Erneuerbare-Energie-Anlagen abgerechnet werden. Da kann es sein, dass Jurist:innen bei der Softwaregestaltung mitarbeiten, damit die Software auch das Richtige macht und nichts falsch abrechnet.
- Sei es die Regulierung zu neuen Themen wie dem Emissionshandel, der in Software übersetzt werden muss, damit ihn die Branche effizient betreiben kann.
- Sei es letztendlich Wissen über Marketing, Kund:innenservice oder Rechnungsstellung, die, auch wenn sie in anderen Branchen existieren, ihre Industriespezifika haben und für die deswegen eigene Softwarelösungen entwickelt oder zumindest Standardsoftwarepakete angepasst werden müssen.

Empirisch ist der Arbeitsprozess der Softwaregestaltung ein Teil der **Konvergenz zweier Industrien**. Es treffen zwei Wertschöpfungsketten aufeinander (Software- und Energiewirtschaft), die sich verschränken und meist auf verschiedene Organisationen verteilt sind. In Abbildung 2 sind die beiden Wertschöpfungsketten dargestellt. Jene der Energiewirtschaft setzt sich aus Energieerzeugung, Handel, Verteilung via (Strom-/Gas-)Netze und Vertrieb zusammen. In der Softwarewirtschaft besteht sie aus der Spezifikation, der Programmierung, den Tests, der Implementierung, Wartung, dem Betrieb bis zum Vertrieb von Software. Beide Wertschöpfungsketten arbeiten an bzw. mit der Anwendungssoftware. Aus Sicht der Energiewirtschaft ist Software zentral für die Datenhaltung, -verarbeitung und die Leistungserbringung. Dazu gehören Transaktionen zwischen Firmen und Kundschaft (B2C, Business to Customer) wie auch zwischen Firmen (B2B, Business to Business). In der Energiewirtschaft ist das zu einem großen Teil mit einem hohen Grad an Automatisierung verbunden, weil die vielen unterschiedlichen Marktteilnehmenden viele Massendaten (Strommengen, Geldbeträge etc.) untereinander austauschen müssen. Kapitel 7 geht näher auf die industriespezifischen Aspekte der Softwaregestaltung ein. Es reflektiert die Bedeutung der Industriestrukturen für die Softwaregestaltung und gibt Hintergrundwissen zum besseren Verständnis der Fallstudien.

Abbildung 2: Konvergenz zweier Branchen?



Die Fallstudien beleuchten verschiedene Konstellationen der Softwaregestaltung und vergleichen sie anschließend (kooperativ-firmenübergreifend, intern, durch Softwarefirmen, in einem Start-up etc.). Die Firmen sind jeweils auf unterschiedliche Weise Teil der Wertschöpfung der Softwarewirtschaft und tragen jeweils unterschiedlich zur Softwaregestaltung bei.

2.3. Zusammenhang von Fragestellung, Technologie, Praxis und Theorie

In dem qualitativen Forschungsprozess haben sich aus einem iterativen Prozess der Reflexion von Theorie und Empirie erst im Laufe der Zeit die Kernfragen, zentralen Probleme der Praxis, konzeptionelle Ausrichtung und theoretischen Einsichten ergeben. Ausgehend von der Fragestellung zu den Formen und Folgen des Arbeitsprozesses der industriespezifischen Softwaregestaltung ergeben sich die relevanten technologischen Grundlagen: der Quelltext, die unterschiedlichen technischen Schichten und sprachlichen Strukturierungen, mit denen die Beteiligten auf Software blicken, worauf sie sich in ihrer Arbeit beziehen und welche sie für die Kommunikation nutzen (Quelltext, Bedienoberfläche, Einstellungsmöglichkeiten, Schnittstellen, Modelle, Algorithmen etc.). Fragestellung und Technologie werfen in der Empirie bzw. den Fallstudien den Blick auf zentrale Mittel der Arbeitspraxis der Softwaregestaltung und typische Probleme.

Sowohl die Software selbst als auch ihr Anwendungsbereich kann sehr komplex sein und die Beteiligten müssen sich trotzdem auf eine Gestaltung einigen, weswegen Wissen und Kommunikation untereinander die zentralen Arbeitsmittel sind.

Als typische praktische Kernprobleme für den Arbeitsprozess der Softwaregestaltung hat die empirische Analyse die **softwaretechnische Interdisziplinarität** und die **softwaretechnischen Gestaltungsmöglichkeiten** identifiziert. Die Fallstudien sind unterschiedliche Beispiele dafür, wie Organisationen diese beiden Probleme lösen. Allgemein stellen sich beide Probleme unabhängig vom Kontext – egal ob ein Start-up oder ein seit Jahrzehnten existierender Industriekonzern Software gestaltet. Für die softwaretechnische Interdisziplinarität müssen die Agierenden das Wissen über die Potenziale der Softwaregestaltung mit dem Wissen über die fachlichen Bedarfe zusammenbringen. Erst der Dialog der Wissensdomänen (IT und Energiewirtschaft) schöpft die Möglichkeiten der Softwareentwicklung aus und überwindet Wissensgrenzen. Rol-

len wie IT-Beratende oder IT-Projektleitende stehen exemplarisch dafür, zwischen IT und (energiewirtschaftlichen) Fachbereichen zu vermitteln. Vor allem bei industriespezifischer Software können Organisationen die Möglichkeiten ihrer Gestaltung erst in Bezug auf den Arbeits-, Organisations- und Branchenkontext ausreizen. Der Begriff der softwaretechnischen Gestaltungsmöglichkeiten betrifft die Gestaltung von Software und von Organisation: Zum einen zeichnet Software aus, dass einerseits Synergien durch die Entwicklung eines Standardsoftwarebausteins, den viele Firmen nutzen, entstehen können. Andererseits kann eine individuelle Software Wettbewerbsvorteile garantieren, indem organisationsspezifisches Wissen einfließt. Zum anderen gibt es die Möglichkeit, die Organisation an einer Standardlösung auszurichten oder selbst Software zu gestalten. Eine ausführliche Beschreibung beider Kernprobleme ist unter 4.2 zu finden.

Mit **soziotechnisch** ist gemeint, dass sich in der Softwaregestaltung Technisches und Soziales »vermengen« (Conrad 2017): ob soziale Organisation und Softwarepaket, Organisation der Anwendung und Organisation der Entwicklung, energiewirtschaftliche Arbeitsaufgaben und technische Softwareanwendung, das Wissen über die Energiewirtschaft und das Wissen über Software(entwicklung), Rollen für IT-Arbeit und Rollen für Anwendungsarbeit. Um eine soziotechnische Arbeitsgestaltung handelt es sich, weil der Arbeitsprozess der Softwaregestaltung nicht nur Software verändert, sondern via Software auch Arbeit gestaltet. Je nach Verhältnis der Arbeitsprozesse von Softwaregestaltung und -anwendung kann dies eine umfassende Reorganisation für die anwendende Organisation bedeuten – vor allem dann, wenn bspw. ein Energieversorgungsunternehmen anfängt, selbst Software zu gestalten, dafür die eigene Organisation ändert und die Anwendungsbereiche (Abteilungen, Teams) reorganisiert.

Eine Kernaussage der vorliegenden Untersuchung ist, dass die Arbeit der Softwaregestaltung die Digitalisierung vorantreibt. Dabei konstituiert sich aus den wissensarbeitenden Softwaregestaltenden und den softwaretechnischen und sozialen Strukturen eine Kontrollform von Arbeit, die Kommunikation und Wissensaustausch ermöglicht. Die analytische Perspektive auf die Softwaregestaltung und wie sie die Softwareanwendung verändert, ist daher jene der **Kontrolle von Arbeit**. Unternehmen kontrollieren Arbeit, weil sie das Transformationsproblem von Arbeitskraft lösen müssen: Anders als bei anderen Produktionsfaktoren ist die Realisierung des Arbeitsvermögens nicht gesichert. Der Betrieb hat aber das Ziel, das Arbeitsvermögen möglichst optimal zu nutzen und an den Zielen der Organisation auszurichten (vgl. Minssen 2017: 301). Wie können Unternehmen sicherstellen, dass Beschäftigte Unbestimmtheitslücken im Arbeitsvertrag nicht in ihrem Interesse ausnutzen und bspw. »Dienst nach Vorschrift« machen? Was im Arbeitsvertrag steht, reicht dafür nicht aus (vgl. Marrs 2010: 331f.).

(Legitime) Herrschaft oder Macht, wie sie z.B. Weber versteht (vgl. Weber 1980: 27), ist dabei nur ein Mittel neben anderen. Die Forschung fasst Kontrolle theoretisch unterschiedlich auf: So kann Kontrolle als die Leitung, Anweisung, Überwachung, Disziplinierung und Belohnung von Arbeit verstanden werden, wozu Bürokratien oder Fließbänder gleichermaßen dienen können (vgl. Edwards 1981). Andere Autoren unterscheiden zwischen Input-, Verhaltens-, Ergebnis-, Clan- und Selbstkontrolle (vgl. Wiener et al. 2016). Wieder andere unterscheiden zwischen persönlicher Kontrolle durch Führungskraft und Kollegenschaft einerseits und unpersönlicher Kontrolle durch standardisierte Arbeits-

prozesse (Bürokratie), Qualifikationen (Professionen) oder Ergebnisse andererseits (vgl. Apitzsch 2010: 89). Wie sich Kontrolle im Fall der Softwaregestaltung am besten konzeptionell fassen lässt, arbeitet die vorliegende Forschungsarbeit heraus.

Dafür wurde nach konzeptionellen Bezügen gesucht, welche die konkreten praktischen Probleme der Softwaregestaltung analysieren helfen und die vorliegende Untersuchung in der Forschungslandschaft verorten. Vor allem die Forschung rund um organisationale, interpersonale und informationstechnische Netzwerke, zu Wissensarbeit und zur Rolle von Software bei der Kontrolle von Arbeit hat sich dafür als hilfreich erwiesen. Der Forschungsstand hat Studien aus der Informatik, der Information Science/ Wirtschaftsinformatik und unterschiedlichen Bereichen der Soziologie (Organisation, Technik, Arbeit) berücksichtigt. Insgesamt ist es der Versuch, die Rezeptionssperre der Sozialwissenschaften zu beenden, da diese die IT bisher weitgehend als Blackbox betrachten (vgl. Lenk 2016: 352).

Aus Theorie und Empirie sind konzeptionelle Ergebnisse entstanden. Zentral sind dabei der Analyserahmen und die beiden Konzepte der soziotechnischen Netzwerkarbeit und soziotechnischen Arbeitsgestaltung. Das Konzept der technikentwicklungsbezogenen Rationalisierung unterstreicht die Bedeutung der Softwaregestaltung für Organisationen.

Das erste konzeptionelle Ergebnis der Arbeit ist der **Analyserahmen der Softwaregestaltung**. Er basiert primär auf der Auswertung der empirischen Daten. Er dient dazu, die Fallstudien zu strukturieren und vergleichen zu können. In Abbildung 3 sind die drei zentralen Elemente des Rahmens dargestellt:

- die soziotechnische Konstellation aus Software (Anwendungsbereich und Architektur) und Organisation von Anwendung und Entwicklung (Gestaltung und Programmierung),
- der soziotechnische Arbeitsprozess der Softwaregestaltung als Kommunikation zwischen Anwendung und Programmierung,
- die Folgen für die Arbeit der Softwaregestaltenden und die soziotechnische Arbeitsgestaltung der Softwareanwendung in den Firmen der Energiewirtschaft.

Alle Elemente des Analyserahmens berücksichtigen, dass es sich um eine soziotechnische Arbeit handelt, wie es für die Arbeit mit Software typisch ist.

Für die Frage nach der Kontrolle der Softwaregestaltung stellt die **soziotechnische Netzwerkarbeit** den konzeptionellen Rahmen dar. Es geht um eine spezifische, netzwerkcharakteristische Form der Kontrolle von Arbeit. Sie ist abhängig von der soziotechnischen Konstellation, in der sie stattfindet. Sie besteht aus dem Arbeitsprozess der Softwaregestaltung und der Arbeit der Softwaregestaltenden. Für beide arbeitet das Empirie-Kapitel typische Unterschiede heraus (siehe 8.6.1.1). So gibt es z. B. Fälle, bei denen sich mehrere Organisationen in einem **zentralen** Arbeitsprozess der Softwaregestaltung abstimmen und die Softwaregestaltenden in einer **Matrixorganisation** arbeiten müssen. Das bedeutet für die soziotechnische Netzwerkarbeit zum einen viel Koordinationsarbeit. Zum anderen besteht die Möglichkeit, viele Perspektiven zu berücksichtigen und einen konsensbasierten Standard zu gestalten. In anderen Fällen gestaltet eine Organisation **dezentral** eine Software für sich selbst. Hier konzentriert sich die Arbeit im Netz-

werk auf das direkte Ausarbeiten der Anforderungen und der Koordinationsaufwand ist gering. Die Softwaregestaltenden arbeiten in einer **reinen Netzwerkorganisation** – ohne formale Hierarchien und Marktbeziehungen. Unabhängig von diesen Unterschieden zeichnet die soziotechnische Netzwerkarbeit aus, dass sie für Feedbackmöglichkeiten zwischen Anwendung und Programmierung sorgt, kooperative Beziehungen auch über Organisations- und Abteilungsgrenzen hinweg schafft und erhält, Softwarewerkzeuge für die verteilte Eingabe von Anforderungen nutzt und Softwaregestaltende erwartungsgeleitet kontrolliert (und z.B. nicht durch die Vorgabe konkreter Arbeitsschritte/-anweisungen).

Abbildung 3: Analyserahmen für die Darstellung und den Vergleich der Fallstudien



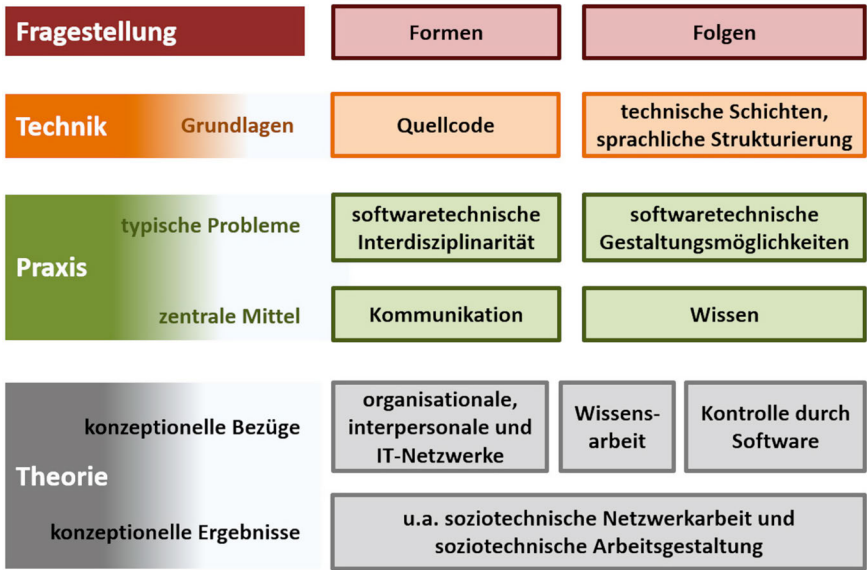
Für die Frage nach dem Verhältnis der Softwaregestaltung zur Softwareanwendung nutzt die vorliegende Untersuchung das Konzept der **soziotechnischen Arbeitsgestaltung**. Dabei unterscheiden sich die Fallstudien, ob sie **unabhängig** oder **abhängig** in der soziotechnischen Arbeitsgestaltung sind (siehe 8.6.1.3). Unabhängig sind sie z.B., wenn die EVU eigenständig eine individuelle Software gestalten. Dann können sie mögliche Konflikte zwischen Anwendung und Gestaltung selbst intern lösen und die Anwendung hat direkt Einfluss auf die Softwaregestaltung sowie die Softwaregestaltung auf die Softwareanwendung. Abhängig sind die EVU, wenn sie eine industriespezifische Standardsoftware verwenden, die eine Softwarefirma gestaltet. Dann müssen sie mit einer externen Firma mögliche Konflikte bei der Softwaregestaltung verhandeln und der Einfluss beschränkt sich primär darauf, wie sie eine Standardsoftware einsetzen, und nicht auf deren Gestaltung.

Mit dem Typ der **technikentwicklungsbezogenen Rationalisierung** grenzt die Untersuchung die Rationalisierung durch Softwaregestaltung von anderen Rationalisierungsformen ab (siehe 6.5.1 und 8.6.3). Es steht z.B. nicht die Arbeitsteilung zwischen Kopf- und Handarbeit im Mittelpunkt wie im Taylorismus, sondern zwischen Softwareanwendung, -gestaltung und -programmierung. Softwaregestaltung kann zudem ein

Mittel für unterschiedliche Zwecke sein (z.B. Automatisierung oder Prozessintegration). Sie rationalisiert die Softwareanwendung nicht nur, indem sie eine Software zur Verfügung stellt. Sie verändert auch die anwendende Organisation, weil diese entweder von einer Standardsoftware abhängt und sich auf die Softwareanwendung konzentriert oder selbst Software gestaltet. Indem Software immer zentraler für die Leistungserbringung von Unternehmen wird, rückt sie in den Mittelpunkt der Arbeit und der Organisation von Unternehmen. Der konkurrenzfähige Betrieb vieler Geschäftsbereiche ist ohne Software gar nicht vorstellbar. Viele Beschäftigte sind über Software in die Wertschöpfung eingebunden. Letztlich lassen sich viele branchenspezifische Prozesse nicht mehr getrennt von der Software(gestaltung) rationalisieren. Dabei muss die Rationalisierung durch Softwaregestaltung mit den Spannungen aus wissensintensiver Kommunikation und Effizienz, aus Innovation und Kostendruck umgehen – vor allem, wenn bestehende Strukturen nicht auf einen interdisziplinären Wissensaustausch ausgelegt sind. Denn auch für die Softwaregestaltung zählen wirtschaftliche Indikatoren: Effizienz, Erhalt und Gewinnung von Marktanteilen und Erzielung von Renditen. Die Unternehmen prüfen regelmäßig, ob es nicht eine bessere, billigere und innovativere Softwarezulieferfirma gibt oder ob sie die Gestaltung und auch Programmierung selbst übernehmen sollten.

Abbildung 4 stellt Fragestellung, Technik, Praxis und Theorie schematisch dar:

Abbildung 4: Übersicht über Kernbestandteile der Untersuchung



Aus der Untersuchung ergeben sich drei Kernthesen, die einen Beitrag zur Arbeits-, Organisations- und Techniksoziologie leisten sollen:

1. Ein zentraler Träger der Digitalisierung ist der Arbeitsprozess der Softwaregestaltung und die Softwaregestaltenden. Die Kontrolle ihrer Wissensarbeit lässt sich als **soziotechnische Netzwerkarbeit** konzeptionieren: erwartungsgeleitet, prozessförmig, beziehungsgebunden, softwarebezogen, netzwerkförmig, wissens- und kommunikationsbasiert und zwischen Anwendung und Programmierung vermittelnd.
2. Der Arbeitsprozess der Softwaregestaltung und sein Verhältnis zur Softwareanwendung stellt eine **soziotechnische Arbeitsgestaltung** dar: Es macht einen Unterschied für anwendende Organisationen wie EVU, ob sie eine individuelle oder eine Standardsoftware einsetzen, ob sie unabhängig gestalten können oder abhängig sind (z.B. von einer Softwarefirma).
3. Softwaregestaltung als **entwicklungsbezogene Rationalisierung** von Technik zielt darauf ab, die Möglichkeiten der Softwareentwicklung (Mittel) für den jeweiligen industriespezifischen Anwendungsbereich (Zweck) effizient zu nutzen.

2.4. Überblick über Kapitel und Argumentation

Die grundlegende Argumentationsstruktur ist wie folgt aufgebaut: 1. Die Softwaregestaltung ist ein eigenständiger und relevanter Arbeitsprozess mit spezifischen Problemen und Mitteln, zu dem weder ausreichend Konzepte noch Empirie existieren (Problem/Forschungsstand). 2. Der Arbeitsprozess der Softwaregestaltung lässt sich als soziotechnische Netzwerkarbeit und in ihren Folgen als soziotechnische Arbeitsgestaltung konzeptionieren (Theorie/Forschungsstand). 3. Die Branchenstruktur der Energiewirtschaft hat Folgen für die Softwaregestaltung und die Branche hat spezifische Anwendungsbereiche für Software (Forschungsfeld). 4. In den Fallstudien zeigen sich unterschiedliche Formen und Folgen der Softwaregestaltung. Der Analyserahmen dient dazu, sie darzustellen und zu analysieren (empirische Analyse). 5. Die Ergebnisse zeigen, dass Softwaregestaltung ein eigenständiger Arbeitsprozess ist, wie ihn Firmen kontrollieren und dass er ein Faktor ist, der erklärt, wie EVU organisiert sind, wie sie arbeiten und wie ein Teil der digitalen Transformation vonstattengeht (Bewertung der Ergebnisse).

Die Argumentation gliedert sich in die folgenden Kapitel: Das 3. Kapitel stellt das Forschungsdesign und die Forschungsmethode vor. Kapitel 4 verweist allgemein auf die veränderte Rolle der Softwareentwicklung in den Organisationen. Die zentralen Probleme in der Praxis der Softwaregestaltung werden dabei genauer ausgeführt. Aus diesen ergeben sich zusammen mit den Besonderheiten der Softwaretechnik (Kapitel 5), der Fragestellung und dem Forschungsdesign die konzeptionellen Bezüge. Diese werden im 6. Kapitel dargestellt und zeigen, dass die Formen und Folgen von Softwaregestaltung an bestehende Forschung anknüpfen und gleichzeitig Probleme darstellen, für deren Verständnis ein Zusammendenken von Software- und Arbeitsgestaltung notwendig ist. Als analytische Grundlage dient dafür der Begriff der soziotechnischen Netzwerkarbeit, bei dem Software in mehrfacher Hinsicht für Kontrolle sorgt und Softwaregestaltende und kooperative Beziehungen auf organisationaler und interpersoneller Ebene eine tragende Rolle spielen. Das Kapitel führt auch den Begriff der soziotechnischen Arbeitsgestaltung ein. Kapitel 7 führt in die Arbeitsweise und Strukturen der Energiewirtschaft ein. Es stellt industriespezifische Anwendungsgebiete von Software und einen Teil der Softwarezulie-

ferindustrie vor. Es beleuchtet das Verhältnis von Software(gestaltung) und Branchenstrukturen. Das 8. Kapitel präsentiert als ein Ergebnis der Untersuchung den Analyse-rahmen, stellt mit ihm die Fallstudien dar und vergleicht sie. Die Zusammenfassung des Kapitels stellt Bezüge zum Theorieteil her (6. Kapitel), bildet Idealtypen und diskutiert einige Hypothesen. Der Schluss (9. Kapitel) fasst die Untersuchung zusammen, stellt den Beitrag zu zwei arbeits- und industriesoziologischen Debatten dar und zeigt weiterführende Forschungsfragen auf.

Die vorliegende Untersuchung versucht so weit wie möglich auf technische Begriffe zu verzichten. Wo dies unumgänglich ist, sind sie erklärt. Wichtiges empirisches Vorwissen über SAP (S. 22), Scrum (S. 33) und Holokratie (S. 93) liefern kurze Exkurse. Fachbegriffe sind direkt in Fußnoten erläutert.

Da der Text einige Wörter sehr häufig verwendet, sind deren Abkürzungen zu Beginn der Untersuchung aufgelistet. Wichtig ist: Wenn Softwareentwicklung geschrieben steht, dann ist damit immer der gesamte Prozess inklusive Gestaltung und Programmierung gemeint.