

8. Formen und Folgen der Softwaregestaltung – die Empirie

Darstellung und Vergleich der Fallstudien

8.1. Einführung: Vorgehen und Kurzvorstellung der sieben Fallstudien

Die Forschungsarbeit beleuchtet, was zwischen Anwendung und Programmierung passieren muss, damit Firmen die Möglichkeiten industriespezifischer Softwareentwicklung für sich nutzen können, und welche Folgen dies hat. In den hier vorgestellten sieben Fallstudien lösen die Organisationen die beiden Kernprobleme der Softwaregestaltung (jene der softwaretechnischen Gestaltungsmöglichkeiten und der Interdisziplinarität) jeweils unterschiedlich, um mal gemeinsam einen Standard zu gestalten oder für sich eine individuelle Software herzustellen. Beides sind Wege, auf denen die EVU die Möglichkeiten der Softwaregestaltung für sich nutzen und ihre Effizienz steigern. Beides sind Wege, welche die Fallstudien nicht immer in ihrer Reinform darstellen. In einigen Fällen liegt eine Mischung zwischen Individual- und Standardsoftwaregestaltung vor. Darüber hinaus reorganisieren sich die EVU in unterschiedlichem Maße, aber nie vollständig, um die Möglichkeiten der Softwaregestaltung zu nutzen. Nur in einem Fall ist eine Organisation von Anfang an auf die Softwaregestaltung ausgerichtet und schöpft ihre Möglichkeiten voll aus: STARTUP gestaltet für sich eine individuelle Software und bietet diese als (Standard-)Produkt anderen Organisationen zur Anwendung an. Bei ihr gilt der Primat der Softwareentwicklung (4.1) und die Anwendung ist nicht nur von einer Software, sondern auch von einer kontinuierlichen Softwaregestaltung abhängig und den Bedarfen der Softwaregestaltung untergeordnet, damit diese ihre Möglichkeiten voll ausschöpfen kann.

Dank der Fallstudien konnte die Forschungsarbeit für die beiden Konzepte der soziotechnischen Netzwerkarbeit (Kontrolle der Softwaregestaltung) und soziotechnischen Arbeitsgestaltung (Verhältnis von Softwaregestaltung zur Softwareanwendung) einen **kategorienbasierten Analyserahmen** ausarbeiten. Zudem hat die Analyse der Fallstudien ergeben, dass es wichtig ist, zwischen Konstellation, Softwaregestaltung und Folgen für Softwaregestaltende und Softwareanwendung zu unterscheiden.

Die **soziotechnische Netzwerkarbeit** als bestehend aus vier Ebenen zu konzeptionieren, war ein erster, vereinfachender Schritt (siehe 6. Kapitel). Nun kommen für die Softwaregestaltung spezifische Kategorien hinzu und die Differenzierung zwischen Konstellation, Arbeitsprozess und Arbeit der Softwaregestaltenden, um die Zusammenhänge zwischen diesen zu untersuchen. Dabei setzt sich die soziotechnische Netzwerkarbeit aus dem Arbeitsprozess der Softwaregestaltung und der Arbeit der Softwaregestaltenden zusammen.

Für die **soziotechnische Arbeitsgestaltung** geht es um die Zusammenhänge zwischen Konstellation und Verhältnis von Softwaregestaltung zur Softwareanwendung und welchen Einfluss Gestaltung und Anwendung aufeinander haben und welche Konflikte zwischen ihnen bestehen.

Der Analyserahmen besteht aus vier Teilen:

1. Der Teil der **soziotechnischen Konstellation** beschreibt die Bedingungen, unter denen im jeweiligen Fall die Organisationen die Möglichkeiten der Softwaregestaltung nutzen. Sie ist deshalb ein eigenständiger Teil des Analyserahmens, weil die Softwaregestaltung auszeichnet, dass sie in sehr unterschiedlichen Kontexten stattfindet. Die soziotechnische Konstellation hat Folgen dafür, wie die Softwaregestaltung stattfindet, die Softwaregestaltenden arbeiten und welche und wie weitgehend Firmen die Möglichkeiten der Softwaregestaltung nutzen können.
2. Der **Arbeitsprozess der Softwaregestaltung** selbst findet unter diesen Bedingungen der soziotechnischen Konstellation statt und zeigt, wie die Organisationen diesen Arbeitsprozess kontrollieren. Was müssen die Organisationen und Arbeitenden können, um eine individuelle oder eine Standardsoftware zu gestalten? Er findet entweder dezentral in den softwareanwendenden Organisationen statt oder zentralisiert in einer Organisation, die für mehrere anwendende Organisationen eine Software gestaltet.
3. Für die **Arbeitsbedingungen der Softwaregestaltenden** sind soziotechnische Konstellation und Arbeitsprozess der Softwaregestaltung mehr oder weniger förderlich. Entweder arbeiten sie in einer Matrixorganisation, d.h., die Softwaregestaltung ist Teil einer Hierarchie oder einer Marktbeziehung zwischen Organisationen. Oder sie arbeiten in einem reinen organisationalen Netzwerk, in dem weder formale Hierarchie noch Markt die Softwaregestaltung beeinflussen. Unabhängig davon, ob es sich um eine Matrixorganisation oder ein reines Netzwerk handelt, stellen die Softwaregestaltenden in allen Fallstudien eine eigenständige Beschäftigengruppe dar, die hinsichtlich des Beschäftigungssystems, der Kontrolle und der Wissensverteilung ihre Besonderheiten aufweist.
4. Zuletzt hat es für die **soziotechnische Arbeitsgestaltung der Softwareanwendung** Folgen, wie die Fallstudien die Möglichkeiten der Softwaregestaltung nutzen. Sie hat Folgen vor allem dahingehend, ob die softwareanwendenden Organisationen die Software unabhängig gestalten können oder abhängig sind von einer Softwarefirma, einem IT-Dienstleistungsunternehmen (IT-DL) oder anderen EVU. Es geht darum, wie die Organisationen die Möglichkeiten der soziotechnischen Arbeitsgestaltung durch die Softwaregestaltung nutzen. Dabei geht es um das Verhältnis der beiden Arbeitsprozesse: Welchen Einfluss haben die Arbeitsprozesse von Softwaregestaltung

und Softwareanwendung aufeinander und welche Konflikte bestehen zwischen beiden?

Als Ergebnis veranschaulicht der Schluss des Kapitels die Unterschiede der Fallstudien in puncto soziotechnischer Netzwerkarbeit und soziotechnischer Arbeitsgestaltung jeweils anhand von vier Idealtypen. Die vier Typen der soziotechnischen Netzwerkarbeiten machen deutlich, dass vor allem der Koordinationsaufwand, aber auch die Möglichkeiten, einen Standard basierend auf einem breiten Konsens zu etablieren, sich je nach Netzwerk unterschieden. Jene vier Typen der Arbeitsgestaltung verdeutlichen dabei noch einmal, dass Organisationen die Möglichkeiten der Softwaregestaltung dann voll ausschöpfen, wenn sie individuell für sich eine industriespezifische Software gestalten und diese gleichzeitig als Standardsoftware anderen Firmen anbieten. Vor allem zeigt sich, dass es um soziotechnische Möglichkeiten geht: nicht nur um individuelle Softwaregestaltung, sondern auch um eine individuelle Gestaltung einer gesamten Organisation, bei der die gestaltete und angewandte Software im Zentrum steht.

Zudem stellt der Schluss einen neuen Rationalisierungstyp vor, der sich aus den Gemeinsamkeiten der Fallstudien ergibt. Kapitel 6 hat ihn bereits theoretisch von anderen abgegrenzt und als Typ der technikentwicklungsbezogenen Rationalisierung bezeichnet. Er basiert auf der Arbeitsteilung von Softwareanwendung, -gestaltung und -programmierung, ist rollen- und softwarebasiert und hat die Kommunikation und den Wissensaustausch zwischen Anwendung und Programmierung als Gegenstand. Mit ihm wird noch einmal das Argument unterstrichen, dass, indem der Arbeitsprozess der Softwaregestaltung Arbeit und Organisation in EVU gestaltet, er Anwendungsbe reiche rationalisiert. Unabhängig davon, welche Form die Softwaregestaltung in den einzelnen Fallstudien annimmt: Es zeigen sich doch immer die gleichen Elemente einer technikentwicklungsbezogenen Rationalisierung.

Zuerst führt das Kapitel kurz in die sieben Fallstudien ein: Wie lösen sie jeweils die Kernprobleme der Softwaregestaltung von softwaretechnischen Gestaltungsmöglichkeiten und Interdisziplinarität? Inwiefern gestalten sie einen Standard oder eine individuelle Software? Zuletzt stellt die Einleitung des Kapitels den Analyserahmen vor und zeigt, wie er mit den Konzepten der soziotechnischen Netzwerkarbeit und der soziotechnischen Arbeitsgestaltung (beide aus Kapitel 6) in Verbindung steht. Die nächsten vier Abschnitte behandeln dann je einen Teil des Analyserahmens, für den sie jeweils die sieben Fälle detaillierter darstellen. Zudem fassen die vier Abschnitte Gemeinsamkeiten und Unterschiede zwischen den Fallstudien zusammen. Der letzte der vier Abschnitte bzw. Teil des Analyserahmens zur soziotechnischen Arbeitsgestaltung diskutiert darüber hinaus drei Thesen: Inwiefern handelt es sich um einen intervenierenden Betriebsrat? Inwieweit kann von einer direkten Partizipation der Anwendenden gesprochen werden? Inwieweit bewegen sich EVU auf einen neuen Typus von Prozessorganisation zu? Das Fazit dieses Kapitels fasst die Ergebnisse zusammen, stellt Bezüge zu bestehenden Forschungsarbeiten aus Kapitel 6 her, geht auf den Typ der technikentwicklungsbezogenen Rationalisierung und auf zwei Punkte gesondert ein: inwiefern die Softwaregestaltung eine Konkurrenz zum Management darstellt und ob es eine industriespezifische Softwaregestaltung gibt.

Um die Fallstudien besser zu verstehen, haben bereits vorhergehende Kapitel in Scrum (5.2.4) und die Entwicklungsplattformen von SAP (4.3) eingeführt. Am Ende der Kurzvorstellung der sieben Fallstudien steht eine kurze Erläuterung zur Holokratie, die bei der Fallstudie STARTUP eine Rolle spielt. Sonstige Fachbegriffe sind in Fußnoten erklärt.

8.1.1. Kurzvorstellung der Fallstudien: Wie sie die Kernprobleme der softwaretechnischen Gestaltungsmöglichkeiten und Interdisziplinarität lösen

Zur Einführung stellt der nächste Abschnitt die sieben Fallstudien kurz vor. Dabei steht im Fokus, wie jeder Fall die Kernprobleme der softwaretechnischen Gestaltungsmöglichkeiten und Interdisziplinarität löst und ob sie eher die Möglichkeiten einer individuellen oder einer Standardsoftware nutzen. Zudem nennt er die Eckdaten der Fälle: Um welche Unternehmen handelt es sich, welche Software entwickeln sie, welchen Anwendungsbereich hat die Software, wer sind die Anwendenden, Programmierenden und Softwaregestaltenden? Die unterschiedlichen Fälle werden dann die nächsten Abschnitte differenzierter und detaillierter mithilfe der vier Teile des Analyserahmens und ihrer jeweiligen Kategorien analysieren.

8.1.1.1. INTERN1: erweitertes Scrum für die mobile Auftragssteuerung der Netz-Instandhaltung

Die Softwaregestaltung findet in dem Fall innerhalb eines EVU statt. Wie löst das EVU das Problem der softwaretechnischen Gestaltungsmöglichkeiten? Es ist softwaretechnisch sowohl auf die Anwendung als auch auf die Entwicklung von Software ausgerichtet. Weil die Softwaregestaltung parallel zur bestehenden Organisation des EVU besteht, existiert eine Matrixorganisation. Iterativ übergeben Rollen wie Anforderungsmanagende und Product Owner:innen im Scrum-Ablauf die Anforderungen aus dem anwendenden Fachbereich an die programmierende IT-Abteilung. Der softwaretechnische Zuschnitt ist individuell, weil das EVU zwar die ERP-Software von SAP erweitert, es die mobile Lösung für die Instandhaltung aber allein für den Fachbereich gestaltet. Die Interdisziplinarität besteht in diesem Fall aus einem Netzwerk der Softwaregestaltung zwischen IT-Abteilung und dem Fachbereich Instandhaltung.

Damit gelingt dem EVU etwas, was anderen misslingt: Es kann intern einen funktionierenden, netzwerkförmigen, kommunikativen Arbeitsprozess für die Softwaregestaltung quer zu bestehenden Abteilungen und Teams etablieren. Es fehlt weder an softwaretechnischem Know-how, noch vereiteln Kommunikationshürden wie Abteilungssilos, hierarchische Konflikte oder Widerstände in den Fachabteilungen die Gestaltung der Software. Einzig die Frage ist offen, wer zukünftig die Hoheit über die Softwaregestaltung hat: der Fachbereich oder die IT-Abteilung.

Das betreffende EVU gehört zu den größeren in Deutschland. Es deckt die gesamte Wertschöpfung der Branche mit unterschiedlichen Tochterfirmen ab. Neben einer zentralen IT gibt es für die Geschäftsfelder Vertrieb und Verteilnetz eigene IT-Abteilungen. Bei dem Unternehmen handelt es sich um einen Verteilnetzbetreiber (VNB) für Strom und Gas, das Teil eines Konzerns ist und mehrere Tausend Mitarbeiter hat. Der Anwen-

dungsbereich der mobilen Lösung ist die Netz-Instandhaltung mit 600 Monteur:innen und 100 Dispatchenden. Das EVU engagiert externe Programmierende. Für den Arbeitsprozess der Softwaregestaltung existiert ein angepasstes Scrum mit Scrum Master und mehreren Product Owner:innen aus der IT-Abteilung und mehreren Anforderungsmanagenden und Key User:innen in den Fachbereichen.

Tabelle 8: Steckbrief Fallstudie INTERN1

Allgemeine Eckdaten	Unternehmen	VNB Strom und Gas (ca. 4000 MA), Teil eines Konzerns
	Anwendungsbe-reich	Instandhaltung Netze
	Software	Mobile Lösung für Monteure und Auftragsverarbeitung für Dispatcher (beides integriert in ERP-System)
	Anwendende	600 Monteure, 100 Dispatcher, Strom und Gas, alle intern
	Programmierende	8 für die mobile Lösung (Backend ¹ , Frontend, Middleware), 2 für die ERP-Integration der Auftragsverarbeitungssoftware; Teil der IT-Abteilung, 4 davon sind Externe
Software-gestaltende	Key User:innen	52 für Monteure, 4 für Dispatcher
	Product Owner:innen	4 (einer davon führend), Teil des Innovationsbereichs der IT-Abteilung
	Anforderungs-mgmt.	5 bis 6 aus dem Fachbereich
	Scrum Master	1 (Teamleiter)
	Ausgewählte An-wendende	für Tests, Resonanzgruppen, Workshops und andere Treffen

8.1.1.2. INTERN2: zentrale Anforderungs runde mehrerer Fachbereiche für die Auftragsverarbeitung

Auch bei INTERN2 findet die Softwaregestaltung innerhalb eines EVU statt. Auch in diesem Fall löst das EVU das Problem der softwaretechnischen Gestaltungsmöglichkeiten damit, dass es sich softwaretechnisch nicht komplett reorganisiert, sondern ergänzend zu bestehenden Strukturen der Softwareanwendung den Arbeitsprozess der Softwaregestaltung etabliert. In dieser Matrixorganisation gestaltet das EVU individuell eine Software. Aber in diesem Fall tut es dies abteilungsübergreifend und anders als bei INTERN1 für mehrere Fachbereiche, die sich regelmäßig in einer Anforderungs runde treffen. Dabei erweitert das EVU die SAP-ERP-Software nicht um eine mobile App, sondern passt ein von SAP geliefertes Modul an. Hier besteht das interdisziplinäre Netzwerk der Softwaregestaltung aus der IT-Abteilung und den dortigen Programmierenden, mehreren Fachbereichen und Abteilungen, von denen einzelne eigene, kleine IT-Teams haben.

¹ Datenzugriffsschicht einer Software (Backend), im Gegensatz zur Präsentationsschicht (Frontend) als Quellcode zur Darstellung der Daten. Eine Middleware ist eine Software, die den Datenaustausch zwischen zwei ansonsten nicht verbundenen Softwarekomponenten ermöglicht.

Der Fall zeigt, wie es einem größeren EVU gelingt, einen Kommunikationsprozess zu etablieren, der eine ganzheitliche, abteilungsübergreifende Betrachtungsweise eines Prozesses erlaubt. Für diese abteilungsübergreifende Softwaregestaltung ist die Kooperation mehrerer Firmenbereiche notwendig, die an einem digitalen Prozess arbeiten und sich die dafür zuständigen Programmierenden teilen.

Das betreffende EVU ist eines der größten in Deutschland. Es deckt die gesamte Wertschöpfung der Branche mit unterschiedlichen Tochterfirmen ab. Neben einer zentralen IT hat der Netzbereich seine eigene IT-Abteilung. Die Tochterfirma für den Netzbetrieb Strom und Gas hat mehrere Tausend MA. Der Anwendungsbereich betrifft die Auftragsverarbeitung bzw. das Work-Management, wie das SAP-Modul heißt. Mehrere Fachbereiche nutzen es. Für die Softwaregestaltung existieren Scrum Master, Key User:innen und Product Owner:innen in den unterschiedlichen Fachbereichen. Dort gibt es auch eigene kleine IT-Teams. Die Abstimmungsrunde bearbeitet ca. 10 Tickets pro Woche. Die Programmierenden sind größtenteils externe Freelancer, welche die IT-Abteilung steuert.

Tabelle 9: Steckbrief Fallstudie INTERN2

Allgemeine Eckdaten	Unternehmen	VNB für Strom und Gas (ca. 4000 MA), Teil eines Konzerns
	Anwendungsbereich	Work-Management (Auftragsverarbeitung): betrifft mehrere Fachbereiche (u.a. Netzzanschluss, Zählerwesen, Abrechnung)
	Software	Teil der ERP-Software von SAP
	Anwendende	keine genauen Zahlen vorhanden, aber auf jeden Fall mehr als 300 Anwendende für das Work-Management
	Programmierende	Programmierende des Work-Managements: 2 Interne, 7 Externe
Arbeitsprozess Softwaregestaltung	Abstimmungsrunde	bestehend aus den betroffenen Fachbereichen (u.a. Netzzanschluss, Zählerwesen, Abrechnung), der IT des Zentralbereichs, Programmierende und Scrum Master
	Product Owner:innen/Anforderungsmgmt.	mehrere in den Fachbereichen und der IT der Abteilung Zentralbereich
	Scrum Master	1
	Key User:innen	20–30 im Fachbereich Netzzanschluss (für die anderen FB liegen keine Zahlen vor)
	Tickets	pro Woche min. 10 Tickets in der Abstimmrunde, ca. 300 im Backlog

8.1.1.3. KOOP1: kooperativ verhandelte, industriespezifische Erweiterung und Anpassung einer Standard-ERP-Software

In dem Fall überlassen die EVU den größten Teil der SAP-Erweiterung und -Anpassung einem IT-DL. Wobei sie im Zuge des vom IT-DL organisierten Anforderungsmanag-

ments kooperativ über den Zuschnitt der Software verhandeln: was sie in einen gemeinsamen Standard aufnehmen und was einzelne EVU individuell gestalten. Die interdisziplinären Netzwerke der Softwaregestaltung erstrecken sich zwischen IT-DL und mehreren anwendenden EVU, wobei die größeren EVU auch noch eigene IT-Abteilungen haben und selbst Software gestalten.

In dem Fall gelingt es den beteiligten Organisationen, die widersprüchlichen Ziele von gemeinsamen Synergien und individueller Gestaltung zu verhandeln. Dabei setzt das IT-DL einen Mediator ein:

»Ich glaube, eines der größten Learnings ist, dass die größten Herausforderungen, einer der großen Kostenpunkte, die Abstimmung überhaupt und den Austausch über den Kunden (unv.). Also dieses ganze Anforderungsmanagement und alles was nichts mit Technik zu tun hat, [...] sondern wirklich miteinander reden und abstimmen und irgendwie sich auf einen gemeinsamen Nenner zu einigen. Dass das eigentlich die größten Hürden sind und das über Jahre hinweg irgendwie so am Laufen zu halten. Dafür haben wir eine Lösung gefunden, auch sogar irgendwann extern moderiert und auch immer noch. Das tut auch gut, wenn ein neutraler Externer dabei ist, der zwischen Kunden und auch uns als Dienstleister irgendwie vermittelt und irgendwie schaut, dass einem da gerecht wird.« (Digitalisierungsmanager IT-DL)

Tabelle 10: Steckbrief Fallstudie KOOP1

Allgemeine Eckdaten	Unternehmen	mehrere EVU (vor allem Netz, Vertrieb, Strom, Gas) und IT-DL
	Anwendungsbereich	sämtliche energiewirtschaftlichen Kernprozesse (Marktkommunikation, Abrechnung, Geräteverwaltung etc.)
	Software	SAP-ERP-Standardsoftware, industriespezifisch erweitert
	Anwendende	mehrere Tausend in unterschiedlichen EVU
	Programmierende	15–20 für den industriespezifischen Teil des ERP beim IT-DL, vereinzelt Programmierende in EVU (z.B. 3 bei EVU2)
Software-gestaltende	Key User:innen	verteilt auf die EVU
	Key Account Managende	mehrere des IT-DL, zuständig für einzelne EVU
	Anforderungsmgmt.	mehrere im IT-DL und den EVU
	IT-Projektmanagende	7 beim IT-DL (2 auf Tests spezialisiert), vereinzelt bei EVU
	IT-Beratende	28–30 für den industriespezifischen Teil des ERP beim IT-DL
	Prozessmanagende	bei zwei der befragten EVU
	IT-Koordinierende	bei einzelnen EVU
	Digitalisierungsmanager	bei einem befragten EVU
	Applikationsbetreuer	bei einem befragten EVU

Die Gesellschaftenden des IT-DL sind mehrere EVU. Die durch das IT-DL betriebene und weiterentwickelte ERP-Software bildet sämtliche Kernprozesse der EVU ab (Marktkommunikation, Abrechnung, Geräteverwaltung etc.). Die Anwendenden und Key User:innen arbeiten in den EVU. Dort findet vereinzelt auch Softwaregestaltung statt (z.B. durch Applikationsbetreuende oder IT-Koordinierende). Programmierende, IT-Beratende, IT-Projektmanagement und Key Account Managende sind beim IT-DL angestellt.

8.1.1.4. KOOP2: prekäre Kooperation für eine industriespezifische Erweiterung und Anpassung einer Standard-ERP-Software

Wie auch bei KOOP1 lösen das Problem der softwaretechnischen Gestaltungsmöglichkeiten mehrere Organisationen. Nur gibt es nicht das eine institutionalisierte Treffen, in dem sich EVU darüber verständigen, welche anstehenden Anforderungen an die IT sie gemeinsam umsetzen und welche nicht. In diesem Fall überlassen die EVU nur einen kleinen Teil der SAP-Erweiterung und -Anpassung einem IT-DL. Den anderen Teil übernehmen die EVU (vor allem die größeren) selber. Das geht so weit, dass sie selbst Softwaregestaltungsprojekte zusammen mit anderen EVU machen. Über den Zuschnitt der Software der EVU entscheidet damit nicht zentral eine Organisation: Es gibt Standard-Erweiterungen und -Anpassungen der SAP-Software durch das IT-DL, durch einzelne EVU und auch Softwaregestaltung mit anderen Softwarefirmen. Die interdisziplinären Netzwerke der Softwaregestaltung erstrecken sich zwischen IT-DL und mehreren EVU. Die größeren EVU haben in diesem Fall eigene IT-Abteilungen.

Der Fall zeigt, dass sich die beteiligten EVU immer mehr aus einer Kooperation lösen, wenn die zentrale Steuerung der Softwaregestaltung nicht gelingt. Im Zeitverlauf wandern koordinative und operative Aufgaben zwischen EVU und IT-DL hin und her. Die uneinheitliche Lage drückt sich darin aus, dass es schwierig war, sich durch die Interviews einen Überblick zu verschaffen: welches EVU nun was selber macht (ob Anwendung, Programmierung oder Softwaregestaltung) oder inwieweit sich die industriespezifische Standardsoftware von EVU zu EVU unterscheidet.

Das IT-DL ist im Eigentum einiger EVU und bietet sowohl Softwareentwicklung als auch -anwendung (inkl. Business Process Outsourcing) an. Dabei soll es nicht nur für die gesellschaftenden Unternehmen Aufgaben übernehmen, sondern mit der Betreuung anderer EVU Geld verdienen. Kernanwendungsbereich der Software ist die Datenverarbeitung für die energiewirtschaftlichen Geschäftsprozesse (Marktkommunikation, Kund:innenservice, Energiemengenbilanzierung etc.). IT-Beratende und Programmierende arbeiten für das IT-DL. Größere EVU haben ihre IT-Projektleitenden, IT-Koordinierenden, Key User:innen und Prozessmanagenden und arbeiten mit unterschiedlichen Softwarefirmen direkt zusammen.

Tabelle 11: Steckbrief Fallstudie KOOP2

Allgemeine Eckdaten	Unternehmen	IT-DL, EVU, diverse Softwarefirmen
	Anwendungsbereich	Datenverarbeitung Geschäftsprozesse EVU
	Software	SAP-ERP-Software, Softwarelösungen anderer Firmen: Marktkommunikation, Kund:innenservice und Netzleitstelle
	Anwendende	verteilt auf mehrere EVU und beim IT-DL
	Programmierende	vereinzelt in EVU, beim IT-DL, Freelancer, Softwarefirma
Software-gestaltende	Key User:innen, Prozessmanagende	EVU2: ERP, Marktkommunikation, Kund:innenservice Netz
	Teamleitung Marktkommunikation	EVU2 Netzbetrieb
	IT-Projektleitung	EVU2 Netzbetrieb
	IT-Koordinierende Fachbereich	EVU2: zuständig für ca. 60 Mitarbeitende Netzbetrieb
	Manager Digitalisierung	EVU3: Vertrieb
	IT-Beratende	IT-DL

8.1.1.5. PAKET: industriespezifische ERP-Standardsoftware entwickelt durch eine Softwarefirma

Die EVU der Fallstudie lösen das Problem der softwaretechnischen Gestaltungsmöglichkeiten damit, dass sie ihre Organisation auf die Anwendung einer Standardsoftware ausrichten und den Standardsoftware-Zuschnitt der Softwarefirma überlassen. Ausgewählte EVU und ihre Expert:innen nehmen an Arbeitskreisen und Projekten mit der Softwarefirma teil. Diese interdisziplinären Netzwerke ergänzen die in der Softwarefirma vorhandene Interdisziplinarität, die notwendig ist, damit sie die industriespezifische Standardsoftware entwickeln kann.

Damit ist, anders als in den anderen Fallstudien zuvor, die entwickelte Software eine Standardware, bei der die Softwarefirma auf Skalierung setzt und die EVU vorwiegend auf Kosteneffizienz und Auslagerung der Softwareentwicklung. Dennoch ist die Softwarefirma auf das Feedback und den Input der EVU zur Qualitätssicherung und Weiterentwicklung des Standards angewiesen.

In dem Fall geht es um eine Softwarefirma und mehrere EVU. Die Software bildet die energiewirtschaftlichen Kernprozesse wie Marktkommunikation, Abrechnung oder Energiedatenmanagement ab. An der Softwaregestaltung sind im wesentlichen Key User:innen, Fachexpert:innen (via Arbeitskreise), Anwendungsbetreuende, Prozessmanagende oder Führungskräfte der EVU sowie Fachexpert:innen, IT-Projektleitende, Führungskräfte und IT-Beratende der Softwarefirma beteiligt. Die Programmierenden arbeiten allesamt in der Softwarefirma. Die Anwendenden sitzen in den EVU, wobei die Softwarefirma auch BPO (Business Process Outsourcing) anbietet und dafür eigene Anwendende beschäftigt.

Tabelle 12: Steckbrief Fallstudie PAKET

Allgemeine Eckdaten	Unternehmen	eine Softwarefirma, mehrere EVU
	Anwendungsbereich	energiewirtschaftliche Kernprozesse wie Marktkommunikation, Abrechnung, Energiedatenmanagement
	Software	ERP-Lösung inklusive industriespezifischer Prozesse
	Anwendende	verteilt auf mehrere EVU, bei EVU6 z.B. ca. 380 Anwendende
	Programmierende	konzentriert auf eine Softwarefirma, genaue Anzahl unbekannt
Software-gestaltende	Key User:innen	verteilt auf diverse EVU
	Fachexpert:innen	verteilt auf diverse EVU und innerhalb Softwarefirma
	IT-Projektleitende	in EVU oder Softwarefirma
	IT-Beratende	mehrere der Softwarefirma
	Anwendungsbetreuung	in manchen EVU
	Prozessmanagende	in manchen EVU
	Führungskräfte	in manchen EVU an Softwaregestaltung beteiligt

8.1.1.6. KOOP3: Ko-Produktion einer IoT-Software für stadtwerksnahe Anwendungen

Wie auch bei PAKET richten sich in dem Fall die EVU drauf aus, die Kernmodule der IoT-Software anzuwenden. Sie überlassen den softwaretechnischen Zuschnitt des Standards der Softwarefirma. Das Problem der softwaretechnischen Interdisziplinarität in Bezug auf die energiewirtschaftlichen Anwendungsfälle löst KOOP3 dadurch, dass das IT-DL die Rolle des Vermittlers in die Energiewirtschaft übernimmt. Es führt die Implementierungsprojekte in den EVU durch und gibt bei deren Durchführung aufkommende Anforderungen an die Softwarefirma weiter.

Im Gegensatz zu den Fällen von KOOP1, KOOP2 und PAKET hat die Kooperation stark informellen Charakter (rudimentäre Verträge, flache Hierarchien, Konflikte werden persönlich gelöst). Weder Marktmechanismen noch hierarchische Befehlsketten oder Abteilungsgrenzen stellen Hemmnisse dar, um das notwendige Wissen aus dem Anwendungsbereich mit jenem der Programmierung zusammenzubringen.

Die beteiligten Unternehmen sind die IoT-Softwarefirma, das IT-DL und die EVU. Schwerpunkt der Anwendungsfälle ist die Installation von Sensoren. Die IoT-Lösung liest diese über LoRaWan² aus, stellt ihre Daten über Schnittstellen zur Verfügung, so dass verschiedene Anwendungen der EVU die Daten weiterverarbeiten bzw. darstellen können. Die Programmierenden arbeiten in der Softwarefirma und zum Teil beim IT-DL oder in EVU, wenn diese die Standardsoftware um eigene Module erweitern. Für die Softwaregestaltung gibt es IT-Projektmanagende in der Softwarefirma, dem IT-DL

² Form der kabellosen Datenübertragung in einem Netzwerk, die sich durch geringen Energiebedarf auszeichnet.

und den EVU. Für die Zusammenarbeit mit der Softwarefirma hat das IT-DL einen extra Product Owner.

Tabelle 13: Steckbrief Fallstudie KOOP3

Allgemeine Eckdaten	Unternehmen	IoT-Softwarefirma, IT-DL, EVU
	Anwendungsbereich	IoT, LoRaWan
	Software	Datenplattform: Daten diverser Sensoren sammeln; bietet Schnittstellen zu anderen Systemen
	Anwendende	keine konkreten Anwendenden, weil diverse Anwendungen die gesammelten Daten verwenden können
	Programmierende	zentral in der Softwarefirma (ca. 15); IT-DL und manche EVU programmieren kleinere Lösungen, die sie über Schnittstellen mit der IoT-Software verbinden
Software-gestaltende	IT-Projektmanagement	Softwarefirma, IT-DL, EVU
	Account Management	Softwarefirma
	Product Owner	IT-DL
	Teamleiter	EVU1

8.1.1.7. STARTUP: Primat der Softwareentwicklung für den digitalen Emissionshandel (E-Mobilität)

Das ist der einzige Fall, in dem die Organisation klar auf Softwareentwicklung ausgerichtet ist und ihre eigene Software herstellt. Damit löst STARTUP das Problem der softwaretechnischen Gestaltungsmöglichkeiten in diesem Fall mit dem Primat der Softwareentwicklung (siehe dazu 4.1). Die Anwendung der Software beschränkt sich allerdings nicht auf STARTUP selbst. Die entwickelte Software bietet die Firma anderen Unternehmen als Anwendungsplattform in Form einer White-Label-Software³ an (z.B. Firmen der Automobilbranche). Die Interdisziplinarität besteht intern zwischen unterschiedlichen Rollen und es gibt keine Grenzen durch Abteilungen oder abgeschottete Teams.

Das Start-up organisiert den Quoten- bzw. Emissionshandel von E-Autos für einzelne Autobesitzer:innen und für Ladesäulen. Um die Emissionen im Verkehrssektor zu reduzieren, hat der Staat die Treibhausgasminderungsquote (kurz: THG-Quote) eingeführt. Unternehmen mit hohen Emissionen, wie z.B. die Mineralölindustrie, können ihre Quoten erfüllen, indem sie über den Quotenhandel Zertifikate von Unternehmen erwerben, die emissionsarme Kraftstoffe für den Verkehr herstellen, wie z.B. Stadtwerke, die Strom z.B. über PV-Anlagen produzieren und über Ladesäulen anbieten. Die vom

³ Wie bereits unter 7.2.1.3 ausgeführt, handelt es sich bei einer White-Label-Software darum: »Eine Organisation entwickelt eine Software, stellt sie zur Verfügung (ob via Cloud oder On-Premises) und versieht sie in der Darstellung nach außen mit der entsprechenden Aufmachung des anwendenden EVU (bspw. Logo und Name Stadtwerk).«

Start-up entwickelte Software hat im Wesentlichen zwei Teile, die auch den zwei angebotenen Dienstleistungen entsprechen: einen für die Anmeldung von E-Autos (B2C) und einen für den Quotenhandel (B2B).

Die Programmierenden arbeiten alle für das Start-up. An der Softwaregestaltung nehmen vor allem all jene teil, die an den zwei Kreisen für Handel und Anmeldung zum Handel mitarbeiten, wozu die Unternehmensleitung, Product Owner, Solution Architect, aber auch Programmierende und andere Beschäftigte gehören.

Tabelle 14: Steckbrief Fallstudie STARTUP

Allgemeine Eckdaten	Unternehmen	organisiert Quotenhandel für E-Mobilität
	Anwendungsbereich	Quotenhandel: Anmeldung von E-Autos (B2C) und Handel mit diesen Zertifikaten, Handel mit Mengen aus Ladesäulen (B2B)
	Software	a) Webseite und App, um E-Autos anzumelden, b) Software für Quotenhandel
	Mitarbeitende insgesamt	ca. 20–25, ca. 13 Vollzeitäquivalent (schwankt im Befragungszeitraum)
	Anwendende	E-Auto-Besitzende, interne Mitarbeitende
	Programmierende	2 für das Modul Anmeldung, 3 für das Modul Zertifikate-Handel
Software-gestaltung	Geschäftsführung	2
	Product Owner:innen	2
	Kreise	2 (je Produkt: Handel und Anmeldung)
	Solution Architect	1

Exkurs: Holokratie

In der Fallstudie STARTUP organisieren u.a. Kreise die Zusammenarbeit. Woher kommt die Idee von Kreisen? Der Verfasser führt dies auf das Organisationskonzept der Holokratie zurück. Es stammt wie das agile Manifest aus der Softwareentwicklung. Bei der Holokratie sind Rollen zentral und es soll eine hierarchiefreie Organisation geschaffen werden. Das Organisationskonzept soll hier zumindest erwähnt werden und der Exkurs auf die Kernelemente hinweisen, um den Fall STARTUP besser zu verstehen und zu zeigen, aus welchen unterschiedlichen Ansätzen sich Organisationen bedienen, um Softwaregestaltung zu organisieren.

Auf der Webseite www.holacracy.org sind auf der Seite Verfassung (<https://www.holacracy.org/constitution/5>) als Organisationsstrukturen aufgeführt:

- Role Definition
- Responsibility of Role Leads

- Circles
- Circle Leads

Dabei werden Kreise wie folgt definiert:

»A ›Circle‹ is a container for organizing Roles and Policies around a common Purpose. The Roles and Policies within a Circle make up its acting ›Governance‹.« (HolacracyOne 2023)

Die Befragten sprachen in den Interviews nie über »Policies«. Doch ist unverkennbar, dass in der Fallstudie STARTUP Rollen und Kreise zentral sind. Die Verantwortlichkeiten der Rollen verhandeln die Beteiligten untereinander. Diese Treffen werden in der Holacracy »governance meetings« genannt:

»Governance meetings help define how we will work together – they facilitate uncovering and assigning the roles needed to reach the circle's aim.« (Robertson 2007)

Kreise haben konkrete Zwecke (z.B. ein Modul der Software zu gestalten) und es gibt Regeln, nach denen sie arbeiten (z.B. wer Treffen organisiert). Obwohl die Anleihen aus dem Organisationskonzept der Holokratie unverkennbar sind, konnten die Befragten nicht bestätigen, dass ihre Organisation davon inspiriert ist. Manchmal sprechen sie von Teams, manchmal von Kreisen.

8.1.2. Unterschiedliche Möglichkeiten der Softwaregestaltung: zwischen Standard- oder Individualsoftware und Überblick über die Fallstudien

Letztlich lassen sich die Unterschiede zwischen den Fallstudien in Bezug auf die Nutzung der Möglichkeiten der Softwaregestaltung auf zwei diametral entgegengesetzte Typen reduzieren: Gestalten sie eine individuelle Software oder einen Standard? Welche Form der Softwaregestaltung am effizientesten für ein EVU ist, lässt sich nicht so einfach beantworten. Denn sowohl eine individuell auf die jeweiligen Bedürfnisse zugeschnittenen Software kann effizient sein als auch eine Software, die viele EVU einsetzen und die einen Standard darstellt. Andererseits kann es unnötig hohe Kosten verursachen, etwas Individuelles zu gestalten, oder es kann ineffizient sein, sich auf eine Standardsoftware auszurichten, wenn eine Organisation dadurch vorher effiziente und optimierte interne Prozesse ändern muss.

In den Fallstudien liegt, was die gesamte IT-Landschaft der EVU betrifft, meistens eine Mischung aus industriespezifischen Standardlösungen oder -softwarebausteinen und individuell gestalteter Software vor. Für den Arbeitsprozess der Softwaregestaltung, der den Kern der jeweiligen Fallstudie bildet, lässt sich jedoch sagen, dass bei INTERN1, INTERN2 und STARTUP Unternehmen eine individuelle Software gestalten. Bei PAKET und KOOP3 gestalten Softwarefirmen eine Standardlösung. Bei KOOP1 ist ein kooperativer Standard das Ziel, wenn es auch individuelle Abweichungen gibt. Bei KOOP2 ist es gar nicht so einfach zu sagen, welches EVU einem gemeinsamen Standard folgt und

welches nicht, weil kein zentralisiertes Entscheidungsgremium dafür existiert. Das restliche Kapitel macht die Unterschiede zwischen den Fallstudien deutlicher und vertieft sie.

Die obige Kurzvorstellung der Fallstudien und wie sie die beiden Kernprobleme der Softwaregestaltung der softwaretechnischen Gestaltungsmöglichkeiten und der Interdisziplinarität lösen, sind in der untenstehenden Tabelle zusammengefasst. Wie ist die Tabelle zu lesen? Zum Beispiel ist bei INTERN1 das EVU durch eine Matrixorganisation aus Softwareanwendung (SA) und Softwaregestaltung (SG) auch auf die Softwaregestaltung ausgerichtet. Dabei besteht der Arbeitsprozess der Softwaregestaltung im Kern aus einem um einige Methoden erweiterten Scrum-Arbeitsprozess. Die Software hat einen individuellen Zuschnitt für den Fachbereich Instandhaltung und die softwaretechnische Interdisziplinarität ist in einem EVU integriert, weil beteiligte Softwaregestaltende, -programmierende und -anwendende Mitarbeitende eines EVU sind. Bei KOOP1 hingegen gibt es sowohl EVU, die nur anwenden, als auch diejenigen, die mitgestalten und dafür den entsprechenden Arbeitsprozess der Softwaregestaltung haben. Den Zuschnitt der Software verhandeln die EVU untereinander und die softwaretechnische Interdisziplinarität ist nicht in einer Organisation integriert, sondern verteilt sich auf EVU und IT-DL, auch wenn einige EVU über softwaretechnisches Wissen verfügen und das IT-DL tiefergehendes energiewirtschaftliches Wissen hat.

Tabelle 15: Überblick über die Fallstudien – softwaretechnische Gestaltungsmöglichkeiten und Interdisziplinarität

Fall	Softwaretechnische Gestaltungsmöglichkeiten		Software. Interdisziplinarität
	softwaretechnische Ausrichtung anwendende Organisation (EVU)	softwaretechnischer Zuschnitt	organisatorische und interpers. Netzwerke der Softwaregestaltung
INTERN1	SG: Matrix Fachbereich – IT-Abteilung, erweitertes Scrum	individuell für einen Fachbereich	integriert: IT-Abteilung und ein Fachbereich
INTERN2	SG: Matrix Fachbereiche – IT-Abt., Anforderungsgrunde	individuell für mehrere Fachbereiche	integriert: IT-Abteilung, mehrere Fachbereiche inkl. deren IT-Teams
KOOP1	SA/SG: organisationsübergr. Matrix EVU – IT-DL, zentrales Anforderungsmanagement	kooperativ verhandelt: Standard – individuell für mehrere EVU	desintegriert: IT-DL und mehrere EVU; Fachbereiche und IT-Abteilung innerhalb der EVU
KOOP2	SA/SG: organisationsübergr. Matrix EVU – IT-DL, verteiltes Anforderungsmanagement	verteilte Entscheidungshoheit: Standard – individuell	desintegriert: IT-DL und mehrere EVU; Fachbereiche und IT-Abteilung innerhalb der EVU

Fall	Softwaretechnische Gestaltungsmöglichkeiten		Software. Interdisziplinarität
	softwaretechnische Ausrichtung anwendende Organisation (EVU)	softwaretechnischer Zuschnitt	organisatorische und interpers. Netzwerke der Softwaregestaltung
PAKET	SA: Anwendung Standard-software inkl. Einstellungen vornehmen	Standard für viele EVU	desintegriert: Arbeitskreise und IT-Projekte zw. Softwarefirma und EVU, intern in Softwarefirma
KOOP3	SA: Anwendung IoT-Software	Standard -Kernmodul IoT für viele EVU	desintegriert: IT-DL, Softwarefirma, EVU
START-UP	SG: Primat der Softwareentwicklung	individuell für die Organisation und als Standard für andere	integriert: intern, rollenbasiert (keine Abteilungsgrenzen), Kreise

Der Analyserahmen hilft, die soziotechnische Netzwerkarbeit der Softwaregestaltung in ihren Zusammenhängen und wesentliche Elemente zu untersuchen.

8.1.3. Der Analyserahmen

8.1.3.1. Die vier Teile und ihre Kategorien

Die vier Teile des Analyserahmens wurden bereits kurz in der Einleitung vorgestellt. Der folgende Abschnitt geht ausführlicher auf sie ein, indem er die jeweiligen Kategorien der Teile vorstellt.

1. Ausgangsbedingungen für die Softwaregestaltung (soziotechnische Konstellation)

Die soziotechnische Konstellation von Anwendung und Entwicklung bestimmt den Spielraum und die Grenzen, welche Möglichkeiten der Softwaregestaltung Organisationen überhaupt in einem Anwendungsbereich verwirklichen können. Indem die Analyse der Fallstudien die Ausgangsbedingungen der Softwaregestaltung berücksichtigt, kann sie zeigen, wie in verschiedenen Kontexten die Beteiligten Wissen austauschen und kommunizieren und welche Folgen der Kontext für den Arbeitsprozess der Softwaregestaltung, die Arbeit der Softwaregestaltenden und die soziotechnische Arbeitsgestaltung der Softwareanwendung hat. Die empirische Untersuchung hat vier zentrale Kategorien der soziotechnischen Konstellation ausgemacht: Anwendungsbereich der Software, Softwarearchitektur, Arbeitsteilung und die Grundkoordination zwischen Anwendung und Entwicklung (Gestaltung und Programmierung). Vom Anwendungsbereich hängt der Digitalisierungsbeitrag ab: Inwieweit kann Software zur Arbeit beitragen oder diese ersetzen? Welches energiewirtschaftliche Domänenwissen ist relevant und wie ist es verteilt? Die Softwarearchitektur prägt die Arbeitsteilung zwischen Anwendung und Programmierung: Welche technischen Gestaltungsmöglichkeiten gibt es außerhalb einer Softwarefirma (Schnittstellen, Anpassungsmöglichkeiten)? Wie ist die Software technisch aufgeteilt und wer gestaltet welchen Teil? Die Arbeitsteilung beschreibt, wie sich Arbeit und Wissen von Anwendung und Entwicklung verteilen und wie die Wis-

sensgrenzen verlaufen. Sind die Programmierenden Teil der IT-Abteilung oder einer Softwarefirma? Sind die Anwendenden Teil eines Fachbereichs eines EVU oder eines IT-DL? Von der Grundkoordination ausgehend müssen Organisationen die kooperative Zusammenarbeit zwischen Anwendung und Entwicklung etablieren. Arbeiten die beteiligten Teams, Abteilungen oder Organisationen primär via Markt (z.B. zwischen Softwarefirma und EVU), Hierarchie (z.B. innerhalb eines EVU) oder Netzwerk (z.B. projektförmig) zusammen?

2. Folgen der Softwaregestaltung für die Arbeit der Softwaregestaltenden

Wie die Softwaregestaltenden als Beschäftigtengruppe in einer Organisation arbeiten, hängt mit der soziotechnischen Konstellation und dem Arbeitsprozess der Softwaregestaltung zusammen. Dabei sind Beschäftigungssystem, Kontrolle und die Wissensverteilung Gegenstand der Analyse und wie sie sich unterscheiden in Fällen von Matrix- oder reinen Netzwerkorganisationen. Mit Beschäftigungssystem ist in erster Linie die Allokation von Arbeitskräften inklusive ihrer Karrierewege gemeint. Den Vergleich mit Anwendenden und Programmierenden nutzt die Zusammenfassung des Abschnitts, um die Eigenarten der Arbeit der Softwaregestaltenden zu verdeutlichen.

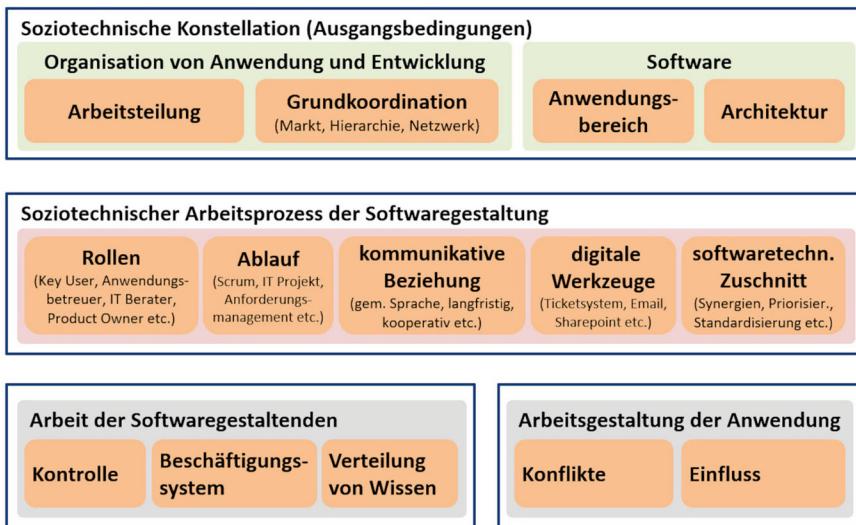
3. Folgen der Softwaregestaltung für die soziotechnische Arbeitsgestaltung der Softwareanwendung

Die Folgen für die soziotechnische Arbeitsgestaltung der Softwareanwendung ergeben sich im Verhältnis von Softwaregestaltung und Softwareanwendung. Erst in diesem Verhältnis zeigen sich Einfluss und Konflikte zwischen beiden. Zentrale Elemente dabei sind, inwiefern Anwendende an der Softwaregestaltung partizipieren, welche Ziele die EVU verfolgen (bspw. die Entwicklung einer Softwarefirma überlassen), ob die Softwaregestaltung die Softwareanwendung kontrollieren kann und ob eine Reorganisation stattfindet (bspw. damit ein EVU selbst Software entwickeln kann). Die soziotechnische Arbeitsgestaltung der Softwareanwendung durch die Softwaregestaltung ist damit nur ein Teil der Arbeitsgestaltung in EVU – allerdings kein unwichtiger. Denn die gestaltete Software prägt je nach Anwendungsbereich weitgehend Arbeitsinhalte und Arbeitsprozesse und vom Arbeitsprozess der Softwaregestaltung hängt die Organisation der anwendenden Organisation ab. Die Fallstudien unterscheiden sich dahingehend, wie unabhängig die EVU Software und damit die Arbeit der Anwendenden gestalten können.

Der Analyserahmen teilt die vier Ebenen der soziotechnischen Netzwerkarbeit von Ablauf, Beziehungen, Software und Softwaregestaltenden aus dem 6. Kapitel auf die drei Teile soziotechnische Konstellation, Arbeitsprozess der Softwaregestaltung und Arbeit der Softwaregestaltenden auf, um die Zusammenhänge zwischen ihnen zu untersuchen. Der Arbeitsprozess der Softwaregestaltung besteht aus den Kategorien Rollen, Ablauf, kommunikative Beziehungen, digitale Werkzeuge und softwaretechnischer Zuschnitt.

Der Rahmen verdeutlicht die soziotechnischen Strukturen der Fallstudien: Softwarearchitektur und Anwendungsbereich der Software als Teil der soziotechnischen Konstellation, softwarebasierte Werkzeuge und softwaretechnischer Zuschnitt als Elemente des Arbeitsprozesses der Softwaregestaltung.

Abbildung 11: Analyserahmen für die Formen und Folgen der Softwaregestaltung – soziotechnische Netzwerkarbeit und soziotechnische Arbeitsgestaltung



8.1.4. Was sind große, mittlere und kleine EVU?

In der Untersuchung steht zwar nicht im Fokus, welche Auswirkung die Organisationsgröße auf die Softwaregestaltung hat. Jedoch gibt es Auffälligkeiten, die mit der Größe der Firmen zu tun haben. Um diese einzuordnen, unterscheidet die Untersuchung der Fallstudien drei Größen:

- kleine EVU: weniger als 800 Mitarbeitende
- mittlere EVU: zwischen 800 und 5000 Mitarbeitende
- große EVU: über 5000 Mitarbeitende

Die Einteilung geht darauf zurück, welche EVU in den Fallstudien welche Auffälligkeiten zeigen (z.B. selbst Software zu gestalten, mitzugestalten oder nur Software anzuwenden). Das ist eine ganz andere Einteilung als jene des Statistischen Bundesamtes, das sämtliche EVU mit mehr als 250 Mitarbeitenden in eine Kategorie steckt (siehe 7.2.1.1). Eine andere Unterscheidung zwischen EVU wäre jene der BNetzA, die EVU mit weniger als 100.000 Kund:innen von einigen regulatorischen Vorgaben befreit (siehe 7.1.2.2.). Der Frage, welche Folgen diese Grenze für die Softwaregestaltung hat, konnte die Arbeit nicht nachgehen. Es scheint aber, dass jene mit weniger als 100.000 Kund:innen tendenziell auf andere Standardlösungen vertrauen (siehe 7.2.1.1).

8.2. Soziotechnische Konstellation als Ausgangssituation der Softwaregestaltung

Die beiden Kernprobleme der softwaretechnischen Gestaltungsmöglichkeiten und Interdisziplinarität lösen die EVU je Fallstudie unterschiedlich. Die folgenden Abschnitte vertiefen mithilfe des Analyserahmens und seiner Kategorien das Verständnis darüber, wie dies geschieht.

Der in 8.2 behandelte erste Teil des Analyserahmens nimmt die soziotechnische Konstellation (Anwendungsbereich, Arbeitsteilung, Grundkoordination, Softwarearchitektur, s.o.) in den Blick. Sie stellt die Ausgangsbedingungen und Grenzen des Arbeitsprozesses der Softwaregestaltung dar, in denen sich die Möglichkeiten der Softwaretechnik verwirklichen lassen. Sie legt bereits im Wesentlichen fest, ob Organisationen eine individuelle Software gestalten oder einen Standard, ob innerhalb eines EVU oder zentralisiert in einer Softwarefirma oder einem IT-DL. Nach der Falldarstellung stellt die Zusammenfassung noch einmal die Unterschiede und Gemeinsamkeiten der Fallstudien heraus und stellt kurz die Zusammenhänge mit den anderen Teilen des Analyserahmens dar. Auf die Folgen der dargestellten Ausgangsbedingungen gehen erst die darauffolgenden Punkte 8.3.ff. ein, die sich den anderen Teilen des Analyserahmens widmen.

8.2.1. Darstellung der Fallstudien

8.2.1.1. INTERN1: Erweiterung ERP-Software durch IT- und Fachabteilung zur Steuerung der Instandhaltung, hierarchisch

Welche Folgen haben in diesem Fall die Ausgangsbedingungen der soziotechnischen Konstellation von Arbeitsteilung, Grundkoordination, Softwarearchitektur und Anwendungsbereich? Die Arbeitsteilung trennt Anwendung und Programmierung klar voneinander. Die so bestehenden Wissensgrenzen zwischen den Abteilungen muss der Arbeitsprozess für die individuelle Softwaregestaltung überwinden, mehrere Programmierenden-Teams einbinden und viele Anwendende mit unterschiedlichen Spezialisierungen berücksichtigen. Die zehn Programmierenden gehören zur IT-Abteilung. Die Anwendenden der Software umfassen ca. 600 Monteur:innen und 100 Dispatcher:innen und sind selbst noch einmal fachlich spezialisiert für Hoch-/Nieder-/Mittelspannung und Strom-, Gas- und Wassernetze.

Aufgrund der Grundkoordination muss der Arbeitsprozess der internen Softwaregestaltung im EVU quer zu den existierenden hierarchischen Strukturen und Abteilungsgrenzen stattfinden. Dies zeigt sich neben den Hierarchieebenen von IT- und Instandhaltungsabteilung an den hierarchisch vorgegebenen Zielen für die Softwaregestaltung: Kostenreduzierung (weniger Personal bzw. mehr Arbeit mit gleichem Personal) und Qualitätssicherung in der Instandhaltung.

Dabei bringt die Softwarearchitektur eine Spezialisierung und damit zusätzliche Wissensgrenzen zwischen den Programmierenden mit sich, die der Arbeitsprozess der Softwaregestaltung berücksichtigen muss. Denn die Software besteht aus vier Teilen, die allesamt für den Arbeitsprozess der Instandhaltung notwendig sind: die Software auf den mobilen Endgeräten, die dazugehörige Software für die Anbindung an das ERP-

System (Middleware⁴), die ERP-Software selbst und eine Software für die Auftragsverarbeitung (Dispatching). Diese dezentrale Softwaregestaltung im EVU, unabhängig vom anbietenden Unternehmen der ERP-Software SAP, ermöglicht die SAP-Architektur, die vielfältige Erweiterungsmöglichkeiten z.B. durch diverse Schnittstellen bietet.

Vom Anwendungsbereich hängt ab, für was das EVU die von ihm gestaltete Software einsetzen kann. In dem Fall ist nicht der gesamte Arbeitsablauf der Netz-Instandhaltung in der Software abgebildet, sondern nur die Steuerung der Monteur:innen. Ihre Steuerung soll möglichst automatisiert ablaufen. Früher haben Meister:innen den Monteur:innen Zettel mitgegeben, auf denen stand, was zu tun ist. Jetzt nimmt ein Team von Auftragsverarbeitenden zentral Aufträge auf (z.B. von Baufirmen) und gibt sie digital an Monteur:innen weiter, die von ihrem mobilen Gerät über Aufträge informiert werden. Sie geben in das mobile Gerät Daten wie Arbeitsstunden oder Zählerstände ein. Wie weit die Automatisierung der Auftragsverteilung geht, ist noch offen. Aktuell gibt es intern im Büro noch sogenannte Dispatchende. Sie stehen für Rückfragen der Monteur:innen zu Aufträgen zur Verfügung und verteilen Aufträge, welche die Software nicht automatisch zuordnen kann.

8.2.1.2. INTERN2: Anpassung ERP-Software durch mehrere Fachbereiche und die IT-Abteilung zur Auftragsdatenverarbeitung, hierarchisch

In diesem Fall führt die soziotechnische Konstellation dazu, dass der Arbeitsprozess der individuellen Softwaregestaltung im Gegensatz zu INTERN1 Wissensgrenzen nicht nur zwischen IT und einem Fachbereich, sondern zwischen mehreren Fachbereichen und IT-Teams überwinden muss. Die mehreren Hundert Anwendenden für das SAP-Modul Work-Management, welches zur Auftragsverarbeitung im Netzbereich dient, sind auf einige Fachbereiche verteilt. Im Unterschied zu INTERN1 sind mehrere IT-Teams auch außerhalb der IT-Abteilung involviert: Zum einen ist eine Abteilung (»Zentralabteilung«) für das gesamte Modul des Work-Managements zuständig. Sie kümmert sich um Weiterentwicklung, Fehlerbehebung und Datenbereinigung. Zum anderen gibt es dezentrale IT-Teams in den Fachbereichen wie Netzzanschluss, die automatisieren und digitalisieren sollen. Dann gibt es noch die zwei internen und sieben externen Programmierenden, die im zuständigen IT-Team der zentralen IT-Abteilung sitzen.

Der interne Arbeitsprozess der Softwaregestaltung im EVU muss althergebrachte interne Hierarchien, deren jeweilige Ziele und Vorgaben und, im Unterschied zu INTERN1, mehrere Abteilungsgrenzen berücksichtigen. Dies zeigt sich neben den Hierarchieebenen auch in den hierarchisch vorgegebenen IT-Budgetzielen für die Softwaregestaltung: Das Management gibt jedem Fachbereich einzeln ein IT-Budget. Im Fachbereich Netzzanschluss gibt das Management Ziele vor: Orientierung an der Kundschaft, Effizienz und Ausrichtung am Standard des SAP-Softwarepaketes.

Dabei ermöglicht die Softwarearchitektur wie bei INTERN1 eine dezentrale Softwaregestaltung unabhängig vom anbietenden Unternehmen der ERP-Software SAP. Wobei es in dem Fall vor allem um die Anpassung des SAP-Standards durch Customi-

4 Eine Middleware ist eine Software, die den Datenaustausch zwischen zwei ansonsten nicht verbundenen Softwarekomponenten ermöglicht.

zing⁵ und Programmierung geht sowie Schnittstellen zu anderen Softwarelösungen herzustellen und weniger darum, eine eigenständige Erweiterung zu gestalten wie bei INTERN1.⁶

Neben Arbeitsteilung, Grundkoordination und Softwarearchitektur stellt der Anwendungsbereich eine Ausgangsbedingung für die Softwaregestaltung dar. Anders als die Instandhaltung bei INTERN1 lässt sich die gesamte Auftragsverarbeitung in Software abbilden. Die EVU können mit dem Modul Work-Management der ERP-Software von SAP anstehende Arbeiten wie den Einbau von Stromzählern oder das Verlegen eines Netzzanschlusses koordinieren und dann entsprechende Rechnungen stellen. Es geht um den Datenverarbeitungsprozess und die Prozessintegration mehrerer, auf Teams verteilter Arbeitsschritte: vom ersten Kontakt mit der Kundschaft über das Verlegen des Anschlusses bis zur Rechnungsstellung. In diesem Prozess erfasst der Fachbereich Netzzanschluss dann z.B. die Dokumentation zu den Anschlässen, Strom-/Gas-/Wasser-Verteilern und Häusern im ERP-System, und der Fachbereich Abrechnung schickt die Rechnung an die Kund:innen.

8.2.1.3. KOOP1: Erweiterung und Anpassung ERP-Software durch IT-DL und EVU zur Datenverarbeitung, marktisiert

Die soziotechnische Konstellation hat in dem Fall zur Folge, dass mehrere EVU darüber kooperativ verhandeln, was sie als gemeinsamen Standard und was sie individuell gestalten. Dabei muss aufgrund der Arbeitsteilung der Arbeitsprozess für die Verhandlung zwischen Individual- und Standardsoftwaregestaltung mehrere Wissensgrenzen überwinden: nicht nur zwischen dem IT-DL und den EVU, sondern auch innerhalb einiger EVU und innerhalb des IT-DL selbst. Denn es besteht in diesem Fall zwar eine starke Zentralisierung der Softwareprogrammierung und -gestaltung aufgrund der Auslagerung vieler IT-Tätigkeiten an das IT-DL. Einige wenige EVU haben jedoch eigene Programmierende, Dienstleistungsunternehmen oder Softwarefirmen, mit denen sie zusammenarbeiten. Bezogen auf die umfangreichen Einstellungsmöglichkeiten an der SAP-Standardsoftware haben neben den Programmierenden auch andere Mitarbeitende die Möglichkeit, über vorgegebene Einstellungsmöglichkeiten Änderungen an der Software vorzunehmen.

Anders als in vorhergehenden Fällen muss sich der Arbeitsprozess der Softwaregestaltung mit einer Marktbeziehung als Grundkoordination zwischen den EVU und dem IT-DL arrangieren. Die Marktbeziehung zeigt sich an Verträgen, Kostenkalkulation und Monitoring durch monatliche Berichte. Zwei der befragten EVU sagen explizit, dass das IT-DL mit anderen IT-DL konkurriert und sie Aufträge auch an andere vergeben. Sie ha-

5 Unternehmensspezifische Einstellung einer Standardsoftware.

6 Zugleich bleibt die Softwaregestaltung auf einen Teil der von Softwareanwendenden verwendeten Softwarelösungen beschränkt. Der befragte Anwendende aus dem Fachbereich Netzzchluss setzt neben dem Modul Work-Management von SAP noch mehrere weitere Systeme ein, weil die Standardsoftware nicht alle notwendigen Funktionalitäten abbildet. Dadurch beschränkt sich die individuelle Softwaregestaltung auf einen Teil der verwendeten Software. Für die anderen Softwarepakete muss der Fachbereich mit anderen Teams der IT-Abteilung zusammenarbeiten.

ben intern wieder Know-how aufgebaut, um Aufwandsschätzungen für Anforderungen des IT-DL hinterfragen zu können. Die Marktbeziehung geht mit Misstrauen einher.

»Also, Beraterfirmen sind auch Vertriebler. Vertriebler verkaufen ihre Seele. Das ist einfach so. [...] Das ist immer alles toll. Die können auch immer alles, was fehlt, noch weiterentwickeln, klar.« (Teamleiter Fachbereich EVU2)

Doch ist das Besondere, dass die beteiligten Organisationen die Marktbeziehungen durch kooperative Beziehungen ergänzen. Es gibt Treffen unterschiedlicher Hierarchieebenen von EVU und IT-DL für operative und strategische Abstimmungen. Es gibt langjährige, persönliche Beziehungen zwischen Beschäftigten der Organisationen. Vor allem die größeren EVU und das IT-DL arbeiten intern über Hierarchien und Abteilungen hinweg zusammen, z.B. in Projekten, wie es für Matrixorganisationen typisch ist. Ob zwischen oder innerhalb der Organisationen: In diesem Fall besteht die Herausforderung darin, die Zusammenarbeit sowohl für einzelne Teile der Software, den gemeinsamen Industriestandard, individuelle Anforderungen als auch für langfristige strategische Projekte zu organisieren. Somit muss die zentralisierte Softwaregestaltung des kooperativen Standards durch das IT-DL trotz Marktbeziehungen und quer zu den hierarchischen Strukturen und Abteilungsgrenzen innerhalb und zwischen den Organisationen stattfinden.

Die Softwarearchitektur des kooperativen Standards sieht drei Stufen der Standardisierung vor: Der Kern besteht aus einem harten Standard, den jedes der beteiligten EVU hat. Dann kommt der Template-Bereich, dessen modulare Funktionalitäten einzelne der EVU nutzen können, aber nicht müssen. Auf der äußersten Schale finden sich freie, individuelle Anpassungen, die nur einzelne EVU auf ihren Systemen haben. Laut Befragten sind derzeit ca. 80 % der Anpassungen und Erweiterungen der ERP-Standardsoftware harmonisiert, d.h. für alle EVU, die an der Kooperation teilnehmen, gleich. Die Softwarearchitektur ermöglicht eine Softwaregestaltung sowohl dezentral in den EVU als auch zentral im IT-DL und zudem noch – wie in den Fallstudien INTERN1 und INTERN2 auch – unabhängig vom anbietenden Unternehmen der ERP-Software SAP.

Neben Arbeitsteilung, Grundkoordination und Softwarearchitektur bestimmt der Anwendungsbereich darüber, wie die Organisationen die Softwaregestaltung einsetzen. In dem Fall ist der Anwendungsbereich der gestalteten Software die industriespezifische Datenverarbeitung: Sie reicht von der Abrechnung der Kundschaft über die regulierungsbedingte Marktkommunikation bis hin zur Energiemengenbilanzierung. Entsprechend gibt es umfangreiche Möglichkeiten der Automatisierung. Eine befragte Sachbearbeiterin (EVU2) ist vor allem mit Fehlerklärungsarbeiten, Vervollständigungsarbeiten (z.B. fehlende Daten ergänzen, Daten aus E-Mails oder Telefonaten mit der Kundschaft eingeben, einzelne Schritte im Abrechnungsprozess manuell durchführen) oder Tests von Softwareänderungen beschäftigt. Aus ihrer Sicht sind 90–95 % der Abrechnung automatisiert und in der Marktkommunikation seien die Zahlen noch höher.

8.2.1.4. KOOP2: Erweiterung und Anpassung ERP-Software durch IT-DL und EVU zur Datenverarbeitung, marktbasierter

Wie bei KOOP1 schafft die soziotechnische Konstellation die Möglichkeit, dass sich EVU mithilfe eines gemeinsamen IT-DL auf einen gemeinsamen Standard einigen – nur, dass es in diesem Fall nicht gelingt. Anders als bei KOOP1 ist die Arbeitsteilung in dem Fall weder für die Anwendung noch die Programmierung oder Gestaltung weitestgehend in den EVU oder beim IT-DL zentralisiert. Nachdem ursprünglich das IT-DL die Softwaregestaltung verantwortet hat, haben einzelne EVU wieder verstärkt die Koordination der Softwaregestaltung übernommen. Programmierung für die Erweiterung der ERP-Software findet sowohl beim IT-DL als auch teilweise innerhalb der EVU oder in den von EVU geleiteten Projekten statt (dann u.U. mit externen Programmierenden wie Freelancern). Das IT-DL übernimmt u.a. Entwicklungsaufgaben für den (halb)jährlichen Formatwechsel, der von der Regulierung vorgegeben ist, oder entwickelt kleinere Software-Lösungen. Es nimmt zudem Customizing an der ERP-Software vor. Die uneinheitliche Arbeitsteilung zeigt sich auch an der Anwendung der Software. Diese findet zum einen in den EVU und, weil einige sie ausgelagert haben, beim IT-DL statt. Ein EVU aus dem Organisationsnetzwerk hat laut einem Befragten gar keine operativen Anwendungen mehr für industriespezifische Arbeiten mit dem ERP-System. Wie bei KOOP1 muss der Arbeitsprozess für die individuelle oder Standardsoftwaregestaltung aufgrund der Arbeitsteilung Wissensgrenzen zwischen Fach- und IT-Wissen nicht nur zwischen dem IT-DL und den EVU, sondern auch innerhalb einiger EVU und innerhalb des IT-DL selbst überwinden.

Wie auch bei KOOP1 muss sich der Arbeitsprozess der Softwaregestaltung aufgrund der Grundkoordination mit der Marktbeziehung zwischen den EVU und dem IT-DL arrangieren. Wie bei KOOP1 reichen Verträge für die Kooperation nicht aus. Beziehungs-pflege gehört zur Koordination zwischen den Firmen dazu. Wobei der Fall zeigt, wie brüchig eine Kooperation sein kann. Die Befragten nennen einzelne Ereignisse, die Auslöser oder Manifestationen von fehlendem Vertrauen zwischen beiden waren, was dann zur Rückverlagerung von IT-Arbeit in die EVU geführt hat.

»Warum ist man seit Jahren unzufrieden und warum ist das eskaliert? Manchmal ist es relativ einfach. Die Erwartungshaltung ist zu groß – von Anfang an. Man hat sich nie verstanden oder nie so richtig ausgetauscht. [...] [W]as möchte der Kunde und was kann das Unternehmen überhaupt leisten? [...] Und wir sind nie das Kernproblem angegangen. Nie. ›Komm, wir reden mal offen darüber.‹ Das ist dann sukzessive eskaliert und irgendwann hatten die Mitarbeiter alle keine Lust mehr.« (Teamleiter IT-DL)

Ein anderer Grund ist, dass ein EVU Entscheidungen über die Softwareanwendung und deren Gestaltung autonom treffen will:

»Der Grund, warum wir zurückgeholt wurden, war, dass das Know-how wieder im Haus ist, weil die Markt-Kommunikation sehr komplex geworden ist und man nicht mehr vom Dienstleister abhängig sein wollte. [...] Ja, die sind diejenigen, die die Prozesse ausführen. Aber sie sind nicht Prozesseigentümer, sollte auch kein Dienstleister sein. Aber wenn ich als Auftraggeber Entscheidungen treffe, muss ich das fachliche

Know-how haben hinten dran ... und deswegen wurde dann damals entschieden, uns wieder zurückzuholen.« (Teamleiterin Marktkommunikation EVU2)

Im Unterschied zu KOOP2 führen die EVU teilweise selbst kooperative Projekte mit anderen EVU durch und das IT-DL stellt Mitarbeitende nur zur Umsetzung einzelner Anforderungen zur Verfügung. Zusätzlich existiert innerhalb der EVU eine Matrixorganisation, d.h., die Mitarbeitenden arbeiten über Hierarchien und Abteilungen z.B. in IT-Projekten zusammen. Die Softwaregestaltung im IT-DL und jene dezentral in den EVU müssen damit basierend auf diesen Marktbeziehungen und quer zu den hierarchischen Strukturen und Abteilungsgrenzen innerhalb der EVU stattfinden.

Anders als in KOOP1 ließen die Interviews keine klare Systematik erkennen, welche EVU inwieweit einen gemeinsamen Standard einsetzen und was sie individuell noch angepasst haben. Die Organisationen erweitern das gleiche SAP-Standard-ERP-System wie in den vorhergehenden Fällen (INTERN1, INTERN2, KOOP2) und somit prägt die gleiche Softwarearchitektur die Arbeitsteilung zwischen Anwendung und Entwicklung: Das ERP-System ermöglicht es, über Schnittstellen Drittsysteme anzubinden, und erlaubt weitreichende Erweiterungs- und Anpassungsmöglichkeiten am Standard. Die Softwarearchitektur ermöglicht sowohl eine dezentrale Softwaregestaltung in den EVU als auch zentral im IT-DL unabhängig von SAP, das die ERP-Software anbietet.

Neben Arbeitsteilung, Grundkoordination und Softwarearchitektur ist der Anwendungsbereich eine Ausgangsbedingung. Er entscheidet darüber, für was die Organisationen Software gestalten und wie weitgehend sie Arbeit durch Software ersetzen können. In dem Fall ist der Anwendungsbereich der gestalteten Software wie bei KOOP1 vorwiegend die industriespezifische Datenverarbeitung: u.a. Abrechnung, Marktkommunikation und Energiemengenbilanzierung.

8.2.1.5. PAKET: Entwicklung industriespezifischer Standard-ERP-Software durch Softwarefirma zur Datenverarbeitung, marktisiert

Als Teil der soziotechnischen Konstellation führt in diesem Fall die Arbeitsteilung dazu, dass der Arbeitsprozess der Softwaregestaltung einer Standardsoftware zentral in der Softwarefirma stattfindet und deshalb die Wissensgrenzen geringer sind. Denn die Softwarefirma macht federführend die Konzeption, die Tests und den Support für die Software in interdisziplinären Teams, die auf einzelne Funktionalitäten oder Module spezialisiert sind. Sie besitzt für die industriespezifische Softwareentwicklung tiefergehendes Wissen zur Energiewirtschaft und speziell zur Regulierung und den grundlegenden Geschäftsprozessen, d.h. all das, was alle EVU leisten müssen. Trotzdem ist die Zusammenarbeit mit ausgewählten EVU zur Gestaltung des Standards notwendig, wobei die Softwarefirma selbst über Branchenwissen verfügt, weswegen nicht so viel Wissen über die Marktbeziehung hinweg ausgetauscht werden muss. Dabei geht es vor allem um Feedback der EVU zu fertigen Konzepten und Umsetzungen. Die operativ Anwendenden sind auf einen fachlichen Bereich spezialisiert und damit auch auf den entsprechenden Teil der Software (Kund:innenservice, Energiedatenmanagement, Abrechnung etc.). Sie sitzen in den EVU, in der Softwarefirma selbst (die als Dienstleistung für EVU Business Process Outsourcing (BPO) anbietet) und bei anderen IT-DL, die für die EVU Geschäftsprozesse abwickeln.

Aufgrund der Grundkoordination innerhalb der Softwarefirma muss sich der Arbeitsprozess der Softwaregestaltung zwar mit den internen Hierarchien arrangieren. Jedoch gibt es spezialisierte, interdisziplinäre Teams für einzelne, fachlich abgegrenzte Teile der Software, in denen ein hierarchieunabhängiger Austausch möglich ist. Weil es sich zwischen Softwarefirma und EVU um eine Lieferbeziehung mit einer klar definierten Leistungserbringung handelt, sind Verträge zentral für die Beziehung. Sie legen fest, welche Funktionen die Software erfüllt. Die Softwarefirma garantiert z.B. die Umsetzung der Regulierung zum gesetzlichen Stichtag, bildet die energiewirtschaftlichen Kernprozesse ab und garantiert Zuverlässigkeit in puncto Qualität und fristgerechte Fertigstellung von Softwareupdates. Die EVU sind davon abhängig, weil sie viele ihrer Aufgaben nur mit der Software erledigen können. Für einzelne, ausgewählte EVU besteht jedoch keine reine Marktbeziehung. Diese bindet die Softwarefirma bei strategischen Entscheidungen über ein Gremium ein. Sie machen bei Arbeitsgruppen und Entwicklungsprojekten mit. Hier muss sich der Arbeitsprozess der Softwaregestaltung mit der Marktbeziehung zwischen EVU und Softwarefirma arrangieren, wenn auch nur für bestimmte Projekte, umfangreichere Konzepte oder kleinere Anforderungen.

Anders als in den anderen Fallstudien, die SAP verwendet haben, sieht die Softwarearchitektur von PAKET keine Programmierungen durch die EVU selbst vor. Der Standard an branchenspezifischen Funktionalitäten ist umfassend und soll ausreichen, um die notwendigen Geschäftsprozesse abzuwickeln. Die Anpassungsmöglichkeiten durch die EVU beschränken sich auf vordefinierte Einstellungsmöglichkeiten. Diese müssen jedoch vorgenommen werden, weshalb trotz des Standards nicht von einer Plug-and-Play-Anwendung gesprochen werden kann. Darüber hinaus stellt PAKET Schnittstellen zur Verfügung, dank derer die EVU Softwarelösungen anderer Softwarehersteller anschließen können.

Neben Arbeitsteilung, Grundkoordination und Softwarearchitektur beeinflusst der Anwendungsbereich die Softwaregestaltung, indem u.a. von ihm abhängt, welche Arbeiten Software erledigen oder steuern soll. In dem Fall ist der Anwendungsbereich zugleich der Kern der energiewirtschaftlichen ERP-Software: die industriespezifische Datenverarbeitung – wie bei KOOP1 und KOOP2. Sie ist das Zentrum der Leistungserbringung in Bereichen wie Abrechnung, Marktkommunikation oder Bilanzierung von Energiemengen. Wie auch in den vorhergehenden Fallstudien ist aufgrund des hohen Anteils an Datenverarbeitung eine hohe Automatisierung möglich. Bei hoher Automatisierung kümmern sich die Anwendenden der Software vor allem um Restfälle, d.h. bspw. einzelne Kund:innen oder Datensätze, welche die Software nicht automatisiert verarbeiten kann. In einem befragten EVU ist die Automatisierung der Datenverarbeitung seit längerer Zeit ein Thema:

Der »automatisierte [...] Rechnungseingang: Seit 15 Jahren arbeiten wir da dran in verschiedenen Stufen.« (IT-Leiter EVU2)

8.2.1.6. KOOP3: Ko-Produktion einer IoT-Anwendung zwischen Softwarefirma und IT-DL zur Überwachung, netzwerkförmig

Wie bei PAKET hat die Arbeitsteilung zwischen Softwarefirma, IT-DL und EVU zur Folge, dass die Softwarefirma die Software zentralisiert programmiert und das IT-DL nur

mitgestaltet, indem es Anforderungen einbringt. Doch anders als bei PAKET spielt die Arbeitsteilung zwischen den Organisationen als Wissensgrenze eine geringe Rolle. Ersstens ist das Wissen gleichmäßig verteilt: Mit dem IT-DL gibt es eine Organisation, die die IoT-Software des Softwareunternehmens bei den EVU implementiert, das Softwareunternehmen bei der Weiterentwicklung der Software unterstützt, selbst Erweiterungsmodulen entwickelt und auch Schulungen für EVU anbietet, damit diese eigenständig Module entwickeln können. Zweitens ist das Wissen über IoT-Anwendungsfälle in der Energiewirtschaft weniger kompliziert und sowohl die Softwarefirma als auch das IT-DL verfügen über Wissen über IoT und die IoT-Standardsoftware. Weil die Wissensgrenzen geringer sind, sich die Möglichkeiten von individueller und Standardsoftwaregestaltung auf verschiedene Organisationen verteilen und sie dabei auch noch kooperieren, zählt der Fall zu den kooperativen Formen der Softwaregestaltung.

Zwar bestehen Marktbeziehungen zwischen Softwarefirma und IT-DL, doch spielen diese bei der Zusammenarbeit eine geringe Rolle. Vielmehr ist die Grundkoordination in diesem Fall das Netzwerk. Warum? Weil Verträge und hierarchische Vorgaben oder Strukturen in diesem Fall keine Hindernisse für die Softwaregestaltung darstellen. Verträge wurden von den Befragten nicht als zentral genannt, genauso wenig wie das IT-Budget. Die Kosten für Sensoren und IoT-Software sind gering. Zudem hat das IT-DL keine große Erwartung, was die Einnahmen durch IoT-Projekte und die EVU was z.B. die Kostensparnis durch IoT-Projekte anbelangt. Es geht primär darum, den Einsatz von IoT voranzutreiben und damit um eine kooperative Softwaregestaltung bzw. Ko-Produktion einer IoT-Software.

Wissensgrenzen spielen auch wegen der Softwarearchitektur eine untergeordnete Rolle. Zum liegt das an der Modularität der Software. Die zentrale Datenplattform, von der Softwarefirma entwickelt, hat Schnittstellen, welche das IT-DL oder die EVU nutzen, um selbst Erweiterungsmodulen zu programmieren. Die Softwarefirma hat ein Interesse daran, dass sich die Arbeit nicht in der Softwarefirma zentralisiert. Das zeigt sich daran, dass die Datenplattform auch ohne größere IT-Kenntnisse bedienbar sein soll u.a. weil z.B. Kommunen wenig Geld und Personal haben. Dafür existiert eine nach dem Prinzip »low code«⁷ gestaltete Software, die es erlaubt, ohne große Programmierkenntnisse Sensoren zu verbauen und in den Datenfluss zu integrieren.

Der Anwendungsbereich grenzt die Möglichkeiten der Softwaregestaltung auf eine Software ein, die Daten erfasst. In diesem Fall setzen die EVU die IoT-Software zur Überwachung ein: ob von Straßen, Stromnetzen oder Abwasserkanälen. Denn unter IoT versteht der untersuchte Fall, dass Sensoren ausgebracht werden und deren Daten in einer IoT-Datenplattform zusammenfließen. Letztere soll Basis für Smart-City-Lösungen sein. Da es sich um eine Infrastruktur für Daten handelt, geht es bei der Implementierung weniger um Anwendende als um Anwendungsfälle. Die Daten fließen meist in andere Anwendungen ein. Bei einem befragten EVU fließen Daten von Sensoren über die Datenplattform in ein bestehendes System des Netzbetriebs zur Netzkontrolle ein.

7 Ansatz in der Softwareentwicklung, der versucht, auf Programmierung zu verzichten, und stattdessen auf grafische Designwerkzeuge oder stark vereinfachte Programmiersprachen setzt.

8.2.1.7. STARTUP: Primat der Softwareentwicklung in einer Organisation zur Datenverarbeitung, netzwerkförmig (rollenbasiert)

Die Arbeitsteilung ist in dem Fall so organisiert, dass die Wissensgrenzen gering sind und bleiben. Dank der soziotechnischen Konstellation muss der Arbeitsprozess für die individuelle Softwaregestaltung keine Organisations-, Team- oder Abteilungsgrenzen überwinden. Die Arbeitsteilung zwischen Anwendung und Entwicklung ist vielmehr so organisiert, dass die Programmierenden und softwaregestaltenden energiewirtschaftlichen Fachleute des Start-ups fest und regelmäßig in Kreisen zusammenarbeiten. Es gibt je einen Kreis für die zwei Teile der Software: für den Zertifikate-Handel und für die Anmeldung von E-Autos. Die Anwendenden sind nicht Teil dieser Kreis-Organisation. Sie sind aber auch nicht jene mit dem tiefergehenden Fachwissen über den Zertifikate-Handel, das die Softwaregestaltung benötigt.

Anders als in den anderen Fällen gestaltet STARTUP die Software primär nur für sich selbst. Es ist von Anfang an auf die Softwaregestaltung ausgerichtet. Es gilt der Primat der Softwareentwicklung (siehe 4.1). Die Grundkoordination ist in diesem Fall das Netzwerk – weder Hierarchie noch Markt stören bei der Softwaregestaltung. Vielmehr ist die Organisation rollenbasiert. Die Verantwortung für Aufgaben je Mitarbeitenden sind als Rollen definiert (»Programmierende«, »Product Owner:in«, »Social Media Management« etc.). Über diese Rolle sind die Mitarbeitenden Kreisen zugeordnet, wobei eine Person mehrere Rollen haben und eine Rolle mehreren Kreisen angehören kann – temporär oder langfristig. Der Netzwerkcharakter zeigt sich auch daran, dass es für den Einzelnen eine Flexibilität gibt, was die Arbeitsaufgaben anbelangt: Es gibt regelmäßige Treffen, wo alle sagen können, ob sie eine weitere Rolle übernehmen oder abgeben wollen.

»Also, wir haben da halt auch keine Hierarchie dahinter in dem Sinne. Sondern, das ist eben ein rein rollenbasiertes System. [...] [I]m Endeffekt basiert alles bei uns auf Kompetenz und Verantwortung und nicht in dem Sinne auf: ›Ja, die Person hat halt das und das Sagen, sondern immer nur inhaltlich auf die entsprechenden Kompetenzbereiche bezogen.« (Programmierer1)

Zuletzt zeigt sich der Netzwerkcharakter daran, dass Mitarbeitende spontan ohne Absprache mit einem Vorgesetzten via direkter Kommunikation Themen bearbeiten, z.B. durch informelle Treffen oder Online-Chats.

Dabei ermöglicht die Softwarearchitektur eine getrennte Softwaregestaltung der beiden Teile (Anmeldung der E-Autos und Quotenhandel) unabhängig voneinander in den Kreisen. Es sind nur wenige Absprachen notwendig. Die beiden Teile verfügen jeweils über Schnittstellen, um andere Softwarelösungen anzuschließen, welche die Firma nicht selbst entwickelt hat (z.B. für die Rechnungslegung).

Neben Arbeitsteilung, Grundkoordination und Softwarearchitektur ist der Anwendungsbereich eine Ausgangsbedingung für die Softwaregestaltung. Er legt fest, für welchen Zweck die gestaltete Software eingesetzt wird. Im Anwendungsbereich der Software geht es um die Anmeldung von E-Autos für den Zertifikate-Handel. Da es sich um eine reine Datenverarbeitung handelt, sind die Möglichkeiten der Softwaregestaltung weitgehend und eine Automatisierung naheliegend. Die Grenzen liegen in dem Fall mehr

in den fehlenden digitalen Schnittstellen zu Behörden und notwendigen Prüfschritten, weil z.B. beim Lesen der Fahrzeugscheine noch Fehler auftreten können oder das STARTUP ausgezahlte Beträge noch einmal manuell prüfen will. Für die Prüfung der Fahrzeugscheine gibt es zwei bis drei 450-Euro-Kräfte als Anwendende. Weil STARTUP von Anfang an und auch weiterhin auf die Möglichkeiten der Softwareentwicklung setzt, machen Anwendende nur das, was (noch) nicht die Software erledigt.

8.2.2. Zusammenfassung

8.2.2.1. Überblick über zentrale Unterschiede und ihre Folgen

Die Fallstudien haben gezeigt, dass die Softwaregestaltung unter ganz unterschiedlichen soziotechnischen Konstellationen stattfindet. Zusammenfassend gibt dieser Teil einen Überblick über die Unterschiede zwischen den Fallstudien, ihre Folgen für den Arbeitsprozess der Softwaregestaltung, die Arbeit der Softwaregestaltenden und die soziotechnische Arbeitsgestaltung in den EVU.

In der untenstehenden Tabelle sind die Ausprägungen der vier Kategorien der soziotechnischen Konstellation für jede Fallstudie zusammengefasst. Wie die Tabelle zeigt, besteht die Arbeitsteilung bei INTERN1 und INTERN2 zwischen verschiedenen Abteilungen (IT- und Fachbereiche innerhalb der EVU), bei KOOP1 und KOOP2 zwischen mehreren EVU und einem IT-DL, bei PAKET zwischen mehreren EVU und einer Softwarefirma, bei KOOP3 zwischen mehreren EVU, dem IT-DL und der Softwarefirma und bei STARTUP innerhalb einer Organisation ohne Abteilungsgrenzen. Der Anwendungsbereich unterscheidet sich zwischen Datenverarbeitung (DV), Steuerung von Beschäftigten und Überwachung. Die Grundkoordination unterscheidet zwischen Markt, Hierarchie und Netzwerk. Bei KOOP1, KOOP2 und PAKET besteht zwischen EVU und IT-DL bzw. Softwarefirma eine Marktbeziehung, innerhalb der Organisationen jedoch eine Hierarchie, die bei der Softwaregestaltung eine Rolle spielt. Die Softwarearchitektur ist bei den ersten vier Fallstudien durch das SAP-ERP-System geprägt. In diesen Fallstudien wird die SAP-Software erweitert und angepasst. Die Ausprägung je Fall ist in der folgenden Tabelle aufgelistet:

Tabelle 16: Überblick soziotechnische Konstellation je Fall

Fall	Schwerpunkt Anwendungsbereich	Arbeitsteilung	Grundkoordination	Softwarearchitektur
INTERN1	Auftragssteuerung Instandhaltung (Steuerung von Beschäftigten)	integriert: ein Fachbereich und IT-Abteilung	Hierarchie	individuell: Erweiterung und Anpassung ERP, mobile App
INTERN2	Arbeitsauftragsverarbeitung Netze (DV)	integriert: etliche Fachbereiche und IT-Abteilung	Hierarchie	individuell: Anpassung ERP

Fall	Schwerpunkt Anwendungsbereich	Arbeitsteilung	Grundkoordination	Softwarearchitektur
KOOP1	Energiewirtschaftliche Geschäftsprozesse (DV)	desintegriert: EVU und IT-DL	Markt/ Hierarchie	kooperativer Standard: Erweiterung und Anpassung ERP
KOOP2	Energiewirtschaftliche Geschäftsprozesse (DV)	desintegriert: EVU und IT-DL	Markt/ Hierarchie	prekärer Standard: Erweiterung und Anpassung ERP
PAKET	Energiewirtschaftliche Geschäftsprozesse (DV)	desintegriert: EVU und Softwarefirma	Markt/ Hierarchie	Standard-ERP
KOOP3	Überwachung mittels Sensoren	desintegriert: EVU, IT-DL, Softwarefirma	Netzwerk	Standard-IoT-Software, individuelle Erweiterung
STARTUP	Anmeldung und Handel CO ₂ -Zertifikate E-Autos (DV)	integriert: etliche Rollen und Kreise	Netzwerk	individuell: eigenständige Module Anmeldung und Handel

Die Unterschiede der soziotechnischen Konstellation haben Folgen für den 2. Teil des Analyserahmens: den Arbeitsprozess der Softwaregestaltung. Soziotechnische Konstellation und Arbeitsprozess der Softwaregestaltung haben wiederum Folgen für die Teile der Arbeitsgestaltung und die Arbeit der Softwaregestaltenden. Die wesentlichen typischen Zusammenhänge seien kurz aufgeführt und anhand von diametralen Typen zur Unterscheidung der Fallstudien verdeutlicht.

1. Die Kategorien der soziotechnischen Konstellation von Arbeitsteilung, Grundkonstellation und Architektur wirken sich auf den **Arbeitsprozess der Softwaregestaltung** aus. Sie bestimmen, ob er **dezentral in EVU oder zentralisiert** Software gestaltet.
2. Für die **Beschäftigtengruppe der Softwaregestaltenden** hängt von der soziotechnischen Konstellation ab, inwiefern sie in einer **reinen Netzwerk- oder in einer Matrixorganisation** arbeiten.
3. Auf die **soziotechnische Arbeitsgestaltung** wirkt sich die soziotechnische Konstellation dahingehend aus, ob in den Fällen ein EVU **unabhängig oder nur abhängig** von anderen Organisationen Software und damit die eigene Arbeit gestalten kann.

Die weiter unten folgenden Abschnitte zu den jeweiligen Teilen des Analyserahmens und die Zusammenfassung des Kapitels gehen ausführlicher auf die diametralen Unterscheidungstypen wie zentral – dezentral, Matrix – reines Netzwerk und unabhängig – abhängig und die Zusammenhänge ein.

8.2.2.2. Ergebnisse des Fallvergleichs je Kategorie

Für jede Kategorie der soziotechnischen Konstellation zeigen die Ergebnisse des Fallvergleichs, welche allgemeinen Aussagen sich treffen lassen.

Arbeitsteilung – wie die Wissensgrenzen verlaufen

Aus Sicht der Softwaregestaltung wird die Arbeitsteilung zwischen Anwendung und Programmierung zur Wissensgrenze, die sie überwinden muss. Sie kann innerhalb oder zwischen Firmen bestehen. Es gibt Fälle, in denen Anwendung und Programmierung in einer Organisation, aber durch Bereichsgrenzen getrennt sind (INTERN1, INTERN2). Es gibt Fälle, bei denen einzelne Organisationen interdisziplinär arbeiten und die Wissensgrenzen nur noch zwischen einzelnen Beschäftigten bestehen – im STARTUP, innerhalb von interdisziplinären Teams in der Softwarefirma von PAKET oder den IT-DL von KOOP1/KOOP2. Was auffällt, ist, dass in den Fällen weniger die Anwendenden Programmierungswissen haben, als vielmehr die Programmierenden über Fachwissen verfügen.

Die untenstehende Tabelle zeigt je Fallstudie die Arbeitsteilung zwischen Anwendung und Entwicklung (Programmierung und Gestaltung).

Tabelle 17: Arbeitsteilung zwischen Anwendung und Entwicklung

Fall	Arbeitsteilung Anwendung – Gestaltung – Programmierung
INTERN1	eine Firma (Fachbereich und IT-Abteilung getrennt), einige Programmierende extern
INTERN2	eine Firma (Fachbereiche und IT-Abteilung getrennt), einige Programmierende extern
KOOP1	verteilt auf IT-DL (Programmierung, Gestaltung, BPO) und EVU (Anwendung, Gestaltung), wenige EVU mit eigenen Programmierenden
KOOP2	innerhalb IT-DL (Programmierung, Gestaltung, BPO) und EVU (Anwendung, Gestaltung), wenige EVU mit eigenen Programmierenden
PAKET	Softwarefirma programmiert, gestaltet und wendet an (BPO), EVU wenden an und gestalten (nehmen Einstellungen am Standard vor)
KOOP3	Softwarefirma programmiert Kern und IT-DL gestaltet mit, IT-DL und EVU programmieren und gestalten Erweiterungen; Anwendung durch EVU
STARTUP	eine Firma, interdisziplinäre Kreise

Grundkoordination – Ausgangspunkt von Kommunikation und Kooperation

Von der Grundkoordination ausgehend (egal welche vorliegt) müssen die Organisationen der Fallstudien den Arbeitsprozess der Softwaregestaltung etablieren. Als Formen der Grundkoordinationsen zwischen Anwendung und Programmierung zeigen sich in den Fällen Markt, Hierarchie oder Netzwerk. Wobei Letzteres besser zur Softwaregestaltung passt, weil Wissensgrenzen geringer sind und es weniger Hürden bei der Zusammenarbeit gibt. Kontrollelemente wie IT-Budget, Service Level Agreements (SLA) zwischen EVU und IT-DL, Projektverträge innerhalb eines EVU oder andere Zielvorgaben können die Folgen von Hierarchien oder Märkten auf die Softwaregestaltung verstärken

oder, wie im Fall von KOOP3, so eine geringe Rolle spielen, dass trotz Marktbeziehung der Typ der Netzwerk-Grundkoordination vorliegt.

Neben diesen Kontrollelementen kann innerhalb der EVU die IT-Abteilung der EVU die Zusammenarbeit zwischen Anwendung und Programmierung prägen. Sie spielen in den Fallstudien zwar eine Rolle. Insgesamt sind in den Fällen die IT-Abteilungen aber keine Hürde bei der Softwaregestaltung. Sie sind für den Betrieb der IT-Systeme zuständig, stellen Programmierende zur Verfügung oder koordinieren die als Externe eingebundenen Programmierenden. Vor allem bei größeren EVU koordinieren sie die Softwaregestaltung.

Tabelle 18: Grundkoordination je Fall und die Rolle der IT-Abteilung

Fall	Grundkoordination	IT-Abteilung EVU
INTERN1	Hierarchie: interne Matrixorganisation	stellt Programmierende und IT-Infrastruktur zur Verfügung
INTERN2	Hierarchie: interne Matrixorganisation	stellt Programmierende und IT-Infrastruktur zur Verfügung, verteilte IT-Teams für Softwaregestaltung in den Fachbereichen
KOOP1	Markt: zwischen EVU und IT-DL	wenn vorhanden (größere EVU): organisieren Anforderungsaufnahme, manche machen eigene IT-Projekte
	Hierarchie: Matrixorganisation innerhalb EVU und IT-DL	
KOOP2	Markt: zwischen EVU und IT-DL	wenn vorhanden (größere EVU): in einem EVU Anforderungsaufnahme organisieren; in anderem EVU initiiert sie IT-Projekte in Fachbereichen
	Hierarchie: Matrixorganisation innerhalb EVU und IT-DL	
PAKET	Markt: Softwarefirma – EVU	FB der EVU arbeiten direkt mit SF zusammen; in einem EVU entscheidet FB unabhängig von IT-Abteilung über Softwareauswahl
	Hierarchie: innerhalb Softwarefirma und EVU	
KOOP3	Netzwerk: Ko-Produktion IoT	Ansprechpersonen bei Implementierung IoT für Integration IoT-Software in bestehende IT-Landschaft
STARTUP	Netzwerk: rollenbasierte Kreise	keine

Softwarearchitektur – Grundstruktur der Arbeitsteilung in Organisationen und Organisationsnetzwerken

Die Softwarearchitektur hat sich in allen Fällen als zentral dafür erwiesen, wie sich die Arbeit in den Gestaltungsnetzwerken verteilt. Erstens ist sie zentral wegen der Aufteilung der Software in z.B. Module, mit der eine Spezialisierung der Programmierenden einhergeht und die darüber entscheidet, was ein EVU außerhalb von Softwarefirmen wie SAP, von PAKET oder der IoT-Softwarefirma von KOOP3 gestalten kann. Dabei ist die Aufteilung der Module, Softwareteile oder Schnittstellen bereits im Vorhinein gegeben. Die gegebene Architektur führt die softwaregestaltenden Organisationen fort. Zweitens schafft die Architektur Abhängigkeiten, wenn sich mehrere Teams, Abteilungen oder Organisationen auf einen Standard einigen müssen. In einigen Fallstudien ist es Teil der

Softwaregestaltung zu verhandeln, welche Anforderung in einen Standard einfließt und welche individuell umgesetzt wird.

Die Architektur prägt erstens, wo und wer Software gestalten und programmieren kann und wie sich die Programmierenden spezialisieren. Alle Fälle zeigen, dass die jeweiligen Programmierenden(teams) auf bestimmte Softwareteile spezialisiert sind. Bei KOOP3 will die Softwarefirma die technischen Voraussetzungen schaffen, damit es möglichst einfach ist, durch Schnittstellen weitere Module anzuschließen. Solche individuellen Module kann dann z.B. das IT-DL unabhängig von der IoT-Softwarefirma programmieren. Bei INTERN1 sind die Programmierenden entweder auf die mobile Lösung oder das Instandhaltungsmodul der ERP-Software spezialisiert, bei INTERN2 auf das Modul für die Auftragsverwaltung. Im Fall STARTUP entwickeln unterschiedliche Programmierende jeweils die Module für Anmeldung oder Handel von CO₂-Zertifikaten.

Bei den Organisationen, die größere ERP-Pakete (mit)gestalten, wie die IT-DL von KOOP1 und KOOP2 und die Softwarefirma von PAKET, verteilen sich die Programmierenden auf fachlich aufgeteilte Softwareteile für energiewirtschaftliche Anwendungsbereiche wie Marktkommunikation, Energiemengenbilanzierung, Abrechnung oder Instandhaltung. In diesen Fällen existieren auf diese Bereiche spezialisierte, interdisziplinäre Teams. Dort programmiert und gestaltet ein Team aus Fach- und Softwarespezialist:innen ein Modul eigenständig.

Die ersten vier Fälle haben das gleiche ERP-System von SAP (Näheres zur Entwicklungsplattform unter 4.3). Für alle EVU, die dessen individuelle Erweiterungsmöglichkeiten nutzen, bedeutet das, dass zur betrieblichen Realität nicht einfach nur die Anwendung von Software gehört, sondern auch eine Test- und Entwicklungsumgebung existiert. So hat die Architektur, die individuelle Erweiterungen zulässt, Folgen für die Anwendenden: Sie können und/oder müssen die kontinuierlichen Softwareänderungen testen.

Die Softwarearchitektur prägt zweitens die Softwaregestaltung dadurch, dass sie vorgibt, was individuell gestaltbar ist und was nicht. Gestalten Organisationen gemeinsam einen Standard, müssen sie sich im Arbeitsprozess der Softwaregestaltung darüber verständigen, was Teil des Standards wird. Dann spiegeln sich Kommunikationsstrukturen und Softwarearchitektur nicht. Ein Team programmiert den gemeinsamen Standard, und die Softwaregestaltung sorgt dafür, dass die Beteiligten über Architekturgrenzen hinweg kommunizieren (das können auch mehrere Organisationen sein). Wenn eine Organisation wie eine Softwarefirma den Standard vorgibt oder ein EVU eine Software individuell gestaltet, ist eine solche Abstimmung zwischen Organisationen nicht Teil der Softwaregestaltung.

In der Mehrzahl der EVU ist die energiewirtschaftliche Arbeitsteilung nach wie vor mehr an fachlichen Themen als an der Software-Architektur orientiert. Alle FB arbeiten mit den für sie relevanten Teilen der ERP-Software, ohne sich mit anderen abstimmen zu müssen. Es gibt aber EVU in einigen Fallstudien, die Software über Architekturgrenzen hinweg gestalten. Ob bei INTERN2 oder KOOP1: Durch fachbereichs- oder EVU-übergreifende Anforderungsrunden entscheiden verteilte fachliche Expert:innen über einen zentral durch ein Team an Programmierenden entwickelten Softwareteil. Die Kommunikationswege sind damit auf ein Netzwerk von Organisationen und Teams verteilt, die allesamt zu einem zentralen, spezialisierten Programmierendenteam führen. Geringer

ist der Kommunikationsaufwand, wenn wie bei PAKET eine Softwarefirma einen Standard gestaltet. Das jeweilige zuständige Team der Softwarefirma gestaltet und programmiert einen Teil der Software, und damit konzentriert sich dort die Kommunikation.

Die Ergebnisse fasst der Schluss des Kapitels noch einmal zusammen und stellt Bezug zur Literatur aus dem 6. Kapitel her.

Anwendungsbereich – Ausgangspunkt und Grenze für den Beitrag von Software zur Arbeit

Welche allgemeinen Aussagen lassen sich über den Anwendungsbereich fällen? Je Fallstudie ist es vom Anwendungsbereich abhängig, wie nützlich Software für diesen sein kann. Das betrifft den Anteil der Datenverarbeitung im Anwendungsbereich und ob die Softwaregestaltung einen gesamten Prozess gestaltet oder nur Teile davon. Von beidem hängt ab, was der Kern der Rationalisierungsmöglichkeit ist und wie komplex die abzubildenden Prozesse und Wissensdomänen sind.

Letztendlich lassen sich die Anwendungsbereiche der Fallstudien unterscheiden, inwieweit sie sich in Software abbilden lassen. Wenn sich der gesamte Anwendungsbereich in Software übersetzen lässt, wie dies bei der Datenverarbeitung der Fall ist, kann die Softwaregestaltung mehr leisten, als wenn nur ein Teil des Anwendungsbereichs in Software übersetzt werden kann, wie dies z.B. in der Instandhaltung der Fall ist. Die Arbeit der Monteur:innen kann Software zwar steuern. Die Arbeit an den Netzen kann die Software aber nicht erledigen. Somit unterscheiden sich die Fallstudien darin, ob Software a) Arbeitende steuert und informiert oder b) Arbeitende ersetzt (Automatisierung). Fälle von c) Steuerung von Maschinen kommen in keiner der Fallstudien vor. Vielmehr geht es in den meisten Fallstudien um die Verarbeitung von Daten, z.B. Buchhaltung, Datenaustausch mit anderen EUVs oder Rechnungsstellung. Von dem Beitrag der Software zur Arbeit, d.h. wie sehr die Möglichkeiten der Softwareentwicklung genutzt werden können, hängt der primäre Fokus der Rationalisierung ab. Wenn eine hohe Automatisierung mithilfe von Software möglich ist, hat das für die Anwendung zur Folge, dass die Anwendenden meist nur nicht-automatisierbare Restfälle bearbeiten müssen. Geht es in erster Linie um die Steuerung der Arbeitenden, bedeutet dies für die Anwendung, dass die Anwendenden Teil eines digitalen Prozesses werden und dieser ihre Arbeit steuert und/oder sie für diesen Input liefern müssen.

Tabelle 19: Anwendungsbereich: Anteil der Datenverarbeitung, Fokus Rationalisierung, spezifische Folgen für Anwendung

Fall	Bereich	Anteil Datenverarbeitung	Primärer Fokus Rationalisierung	Spezifische Folgen für Anwendung
Alle	energiewirtschaftliche (Geschäfts-)Prozesse	hoch	Automatisierung	Restfallbearbeitung
INTERN1	Instandhaltung	gering	Steuerung	Fernsteuerung, Prozessintegration
KOOP2	Kund:innenservice, IoT	gering	Information	Prozessintegration

Der Anwendungsbereich bestimmt nicht nur, wie weit die Möglichkeiten der Softwaregestaltung gehen und was der Fokus der Rationalisierung ist. Er legt zusätzlich die Prozesstiefe fest und damit, wie lang die Prozesskette ist, welche die Softwaregestaltung betreffen kann. In den Fallstudien geht mit der Prozesstiefe immer auch eine zunehmende Vielfalt der Wissensdomänen einher. Vor allem bei INTERN2 ist das ein Thema, weil mehrere Fachbereiche zusammenarbeiten müssen und ein höherer Koordinationsaufwand besteht. Dieser ist bei STARTUP deutlich geringer. KOOP3 bildet keinen Arbeitsprozess ab.

Tabelle 20: Prozesstiefe und Wissensdomänen je Fall

Fall	Prozesstiefe	Energiewirtschaftliche Wissensdomänen
INTERN1	2 Prozessteile (Instandhaltung und Auftragsverarbeitung)	verschiedene Energiesparten und Netzgebiete
INTERN2	1 Prozessteil (Softwaremodul »Work-Management«)	verschiedene Fachbereiche, ein Thema: Verwaltung von Arbeitsaufgaben für den Netzanschluss
KOOP1	viele Prozessteile	viele und verschiedene Fachbereiche auf mehrere EVU verteilt
KOOP2	viele Prozessteile	viele und verschiedene Fachbereiche auf mehrere EVU verteilt
PAKET	viele Prozessteile	viele und verschiedene Fachbereiche auf viele EVU verteilt
KOOP3	keine prozessabbildende Software	diverse Anwendungsfälle
STARTUP	2 Prozessteile (Anmeldung und Übergabe an Händler)	THG-Quote (Emissionshandel E-Mobilität)

8.3. Formen des soziotechnischen Arbeitsprozesses der Softwaregestaltung

Der soziotechnische Arbeitsprozess der Softwaregestaltung ist jener Teil des Analyserahmens, der die Umsetzung der Softwaregestaltung in den Blick nimmt. Die Analyse der Interviews ergab fünf Kategorien, die zentral für diesen Arbeitsprozess zwischen Anwendung und Programmierung sind. Mit ihrer Hilfe zeigt dieser Abschnitt, wie Organisationen oder Organisationsnetzwerke im Arbeitsprozess die Arbeitskraft transformieren, um die Möglichkeiten der Softwaregestaltung zu nutzen – ob für eine Individual- oder für eine Standardsoftware.

8.3.1. Arbeitsprozess der Softwaregestaltung: zwischen zentral und dezentral

Beim Arbeitsprozess der Softwaregestaltung lassen sich die Fallstudien danach unterscheiden, ob die Softwaregestaltung eher dezentral (innerhalb der EVU: ob als Projekte, in Abteilungen oder Teams) oder zentralisiert (in einer Softwarefirma oder einem IT-DI)

passiert. Beide Extreme hängen mit der Arbeitsteilung, der Grundkoordination und der Architektur der soziotechnischen Konstellation zusammen.

Bei **dezentraler** Softwaregestaltung ist aufgrund einer Arbeitsteilung, bei der die Softwaregestaltung Teil der anwendenden Organisation (EVU) ist, der Arbeitsprozess darauf ausgerichtet, Anforderungen aufzunehmen, diese zu sammeln und aufzubereiten und direkt zwischen Anwendenden, Gestaltenden und Programmierenden zu kommunizieren. Es können sich langfristige Beziehungen, eine gemeinsame Sprache und eine gemeinsame Wissensbasis etablieren. Entsprechend helfen die digitalen Werkzeuge dabei, direkten Input der Beteiligten zu organisieren. Der softwaretechnische Zuschnitt ist individuell.

Bei einer **zentralisierten** Softwaregestaltung ist aufgrund einer Arbeitsteilung, bei der sich die EVU auf die Softwareanwendung konzentrieren, der Arbeitsprozess darauf ausgerichtet, Anforderungen verschiedener Organisationseinheiten zu koordinieren, sie in Gremien zu verhandeln und über stärker formalisierte Abläufe aufzunehmen. Dabei stehen Anforderungen im Fokus, welche für die gesamte Branche oder zumindest für mehrere EVU relevant sind. Es ist schwieriger, kooperative Beziehungen zu etablieren, weil Spannungen bestehen: zwischen verschiedenen Marktpersonen mit unterschiedlichen Interessen oder weil Hierarchien oder Führungskräfte mit unterschiedlichen Interessen und Kompetenzbereichen beteiligt sind. Erwartungen sind abzugleichen und Konflikte zu lösen. Die digitalen Werkzeuge helfen u.a. durch die Schaffung von Transparenz bei der Koordination von Handelnden, die über mehrere Organisationseinheiten verteilt sind. Der softwaretechnische Zuschnitt ist ein Standard.

Tabelle 21: Idealtypen zentraler und dezentraler Arbeitsprozess der Softwaregestaltung

Typ	Typische Erwartungen an die Rollen	Typisch für den Ablauf	Typischer Nutzen Werkzeuge	Typische kommunikative Beziehungen	Software-technischer Zuschnitt
Zentral	Koordination	Gremien für Verhandlungen, Erwartungsabgleich, Konfliktlösung	Transparenz, Abstimmungen	Spannungen ausgleichen, bürokratisch	Standard
Dezentral	Anforderungsaufnahme	Anforderungen direkt sammeln und ausarbeiten	direkter Input	kooperativ, direkt, gemeinsame Sprache	individuell

Inwiefern eine Zuordnung zu einem aus dem empirischen Material der Interviews entwickelten Idealtyp so einfach möglich ist, zeigen die Fallstudien selbst und diskutiert die Zusammenfassung am Schluss dieses Abschnitts zum Arbeitsprozess der Softwaregestaltung.

8.3.2. Darstellung der Fallstudien

Die folgenden sieben Fallstudien sind anhand der Kategorien Rollen, Ablauf, kommunikative Beziehungen, digitale Werkzeuge und softwaretechnischer Zuschnitt dargestellt. Sie vertiefen und ergänzen die im obigen Überblick aufgelisteten Unterschiede und wie die Organisationen die Möglichkeiten zwischen Individual- und Standardsoftwaregestaltung nutzen.

8.3.2.1. INTERN1: erweitertes Scrum, Gestaltungsnetzwerke, langfristige Beziehungen, dezentral

In dem Fall liegt ein dezentraler Arbeitsprozess der Softwaregestaltung für eine individuelle Software innerhalb eines EVU vor. Um einen solchen zu ermöglichen und um Anforderungen aufzunehmen, setzt das EVU Methoden wie Scrum, Resonanzgruppen und Workshops ein. Indem die kommunikativen Beziehungen offen, direkt, langfristig und tief verankert im Fachbereich bzw. nah an den Monteur:innen dran sind, erleichtern sie den Austausch quer zu den Abteilungsgrenzen und Hierarchien. Neben einem Ticketsystem wird ein Chat-Kanal verwendet, der den dezentralen Input von Anwendenden für Anforderungen ohne bürokratische Hindernisse erlaubt.

Rollen: Product Owner:innen, Anforderungsmanagende, Key User:innen

Wie für einen dezentralen Arbeitsprozess der Softwaregestaltung typisch, besteht an die Rollen vor allem die Erwartung, Anforderungen aufzunehmen, und zwar sehr nah an und direkt mit den Anwendenden im Fachbereich. Dafür gibt es in dem Fall gleich mehrere Rollen: So bearbeitet selbst der Haupt-Product-Owner (neben seiner Leitungsfunktion) Anforderungen. Daneben gibt es noch drei weitere Product Owner:innen, die für einzelne Funktionalitäten zuständig sind (wie z.B. für die mobile App der Monteur:innen), und noch fünf bis sechs spezialisierte Anforderungsmanagende aus dem Fachbereich (Hochspannung, Nieder-/Mittelspannung, Gas, Wasser, Strom). Eine befragte Anforderungsmanagerin hat viele Jour fixes mit verschiedenen Bereichen (ob IT-, Fachabteilungen oder Teams). Sie beschreibt ihre Arbeit so:

»Ich schaue mir die Prozesse an und schaue mir an: Wo haben wir Verbesserungspotenzial, wo wir mit Digitalisierung irgendwas verbessern können? Ich nehme aber auch Anforderungen auf von Anwendern, wo die einfach sagen: Hier habe ich ein Stück Papier, das will ich künftig nicht mehr ausfüllen können. Könnt ihr das digitalisieren? Oder: Hier haben wir einen Prozess, da muss ich zehnmal telefonieren, das passt mir nicht. Und dann schaue ich mir das an. Schaue noch nach rechts und links, wie es andere machen und dass wir da irgendwie Synergieeffekte vielleicht gewinnen können. Und wir überlegen uns dann Lösungen. Ich schreib dann eine Story⁸, tausche mich mit dem PO⁹ [...] oder mit den Ansprechpartnern dort gezielt aus – je nachdem ob backend¹⁰ oder frontend. Und dann wird es in eine Story gegossen, priorisiert und

⁸ Anderes Wort für Anforderung, meist im Scrum-Setting so bezeichnet.

⁹ Product Owner

¹⁰ Datenzugriffsschicht einer Software, im Gegensatz zu Präsentationsschicht (Frontend) mit dem Quellcode zur Darstellung der Daten.

umgesetzt und von mir dann getestet und kommuniziert nach draußen: Ab jetzt haben wir den und den Prozess. Das mache ich dann auch noch. [...] Ich telefoniere sehr viel.« (Anforderungsmanagerin)

Als letzte Stufe der Softwaregestaltung zwischen Anwendung und Programmierung existieren Monteur:innen, welche zusätzlich zu ihrer Montagearbeit als Key User:innen agieren. Sie sind Sprecher:innen, Multiplikator:innen und technischer Support. Sie installieren z.B. mobile Apps.

Nicht nur die vielen unterschiedlichen Rollen innerhalb des Fachbereichs der Instandhaltung zeigen, dass die Organisation die Ausgestaltung der Rollen für eine dezentrale Softwaregestaltung situativ anpasst. Die situative Anpassung der Rollen zeigt sich in diesem Fall auch an der Rolle des Scrum Masters. Diese übernimmt, anders als in der Scrum-Methode vorgesehen, eine Führungskraft (ein Teamleiter). Er koordiniert die Aufgaben und kümmert sich um Schwierigkeiten im Ablauf.

Ablauf: Scrum, Resonanzgruppen, Workshops

Der Ablauf besteht primär, wie für einen dezentralen Arbeitsprozess der Softwaregestaltung typisch, im Ausarbeiten und Sammeln von Anforderungen, wobei das in diesem Fall besonders feedback-intensiv passiert. Grundlage dafür ist Scrum. Das EVU setzt die Methode vollumfänglich um, wozu ein iteratives Vorgehen gehört. Neben den typischen Rollen und Artefakten (siehe 5.2.4) ist der Ablauf darauf ausgelegt, immer wieder zwischen den Beteiligten dezentral im EVU ein gegenseitiges Verständnis zu sichern, Erwartungen abzulegen und Wissen auszutauschen, um die Anforderungen aufzunehmen. Resonanzgruppen waren vor allem zu Beginn des Projektes und bei größeren Umsetzungsideen Teil des Ablaufs. Dort sagen z.B. Programmierende, wie sie die Anforderung verstanden haben. Nach ein paar Wochen stellen sie die Änderungen den Anwendenden vor und fragen, ob sie den Erwartungen entsprechen. Der Austausch hat sich mit der Zeit immer mehr zu den Key User:innen hin verlagert. Unter anderem in Workshops erarbeiten die Anforderungsmanagenden mit den Key User:innen Anforderungen.

Kommunikative Beziehungen: zugänglich, offen, interdisziplinär und gut vernetzt

Der Fall zeigt die für einen dezentralen Arbeitsprozess der Softwaregestaltungen typischen direkten und langfristigen Beziehungen und eine gemeinsame Sprache. Anders als in anderen Organisationen stellen für Mitarbeitende des EVU die typischen Kommunikationshemmnisse interner Gestaltungsprojekte wie Team-, Abteilungssilos oder Hierarchien keine Hindernisse dar. So existieren zusätzlich zu den Resonanzgruppen oder Workshops persönliche Beziehungen vom IT-Team zum FB und es entsteht eine gemeinsame Sprache. Die Programmierenden reden direkt mit Anwendenden, vor allem, wenn sie feststellen, dass sie über eine Story zu wenig wissen.

»Aber ja, wir kommen gut an Informationen ran. Wir müssen nicht irgendwie ewig über irgendwelche Chefs gehen und sagen: Dürfen wir mit dem so oder so sprechen. Das ist sehr angenehm, muss ich sagen.« (Programmierer)

Der befragte Architekt beschreibt eine Offenheit im Austausch:

»Da ist, wie gesagt, eine große Offenheit der Menschen. Insofern kann das Wissen frei fließen. [...] Ich hatte den Eindruck, dass wenn man fragt, man immer Antworten kriegt. Halt so gut, wie der Gegenüber das weiß, sowohl auf Fachbereichs- als auch auf IT-Seite, als auch bei schwierigen Themen, die man vielleicht erst mal überhaupt fassen muss, damit man darüber sprechen kann.« (Architekt)

Die Anforderungsmanagerin hat ein internes Netzwerk, das ihr direkte Gespräche unabhängig von den hierarchischen Strukturen ermöglicht. Sie fragt nach, wenn sie etwas nicht versteht. Sie bekommt mit, was andere Abteilungen machen, und kann Beschäftigte zusammenbringen, bei denen sie feststellt, dass sie am gleichen Thema arbeiten. Die richtige Sprache zu sprechen, d.h. die richtige Terminologie zu verwenden, ist eine Qualifikation, die von der Anforderungsmanagerin mit der Zeit erworben wurde:

»Ich musste IT-Deutsch lernen. Wenn ich Storys schreibe, dann muss ich dies so schreiben, dass die Softwareentwicklung versteht, obwohl ich aus der Anwender-Ecke gekommen. Ja, ich muss mich da einfach in so eine IT-Denke reinschrauben.« (Anforderungsmanagerin)

Der Fachbereich hat aus Sicht des befragten Programmierers dazugelernt, kann besser formulieren, was er will. Auch die Fehlermeldungen, die bei Tests aufgenommen werden, sind qualifizierter und unterstützen so die Fehleranalyse. Es wird nicht nur geschrieben »es tut nicht« (Teamleiter IT).

Digitale Werkzeuge: dezentraler Input dank Ticketsystem und Chat-Kanal

In dem Fall geht das EVU über die für digitale Werkzeuge typische Verwendung in dezentralen Arbeitsprozessen der Softwaregestaltung hinaus. Es existiert nicht nur ein Ticketsystem, das die auf verschiedene Abteilungen verteilten Mitarbeitenden vernetzt und ihnen ermöglicht, direkt und dezentral Anforderungen aufnehmen zu können. Es existiert zusätzlich ein Microsoft-Teams-Kanal, den Key User:innen und Anforderungsmanager:innen des Fachbereichs verwenden. Er hat ca. 110 Mitglieder. Dort können die Teilnehmenden Informationen austauschen, Anforderungen und Probleme besprechen. Dadurch ist eine offene, hierarchieunabhängige und direkte Kommunikation möglich. Die verwendeten Werkzeuge dienen damit primär als Infrastruktur für die Zusammenarbeit, um die Kommunikation zu ermöglichen, zu dokumentieren und zu koordinieren, und weniger zur Standardisierung oder Formalisierung.

Softwaretechnischer Zuschnitt: individuelle Anforderungen der Anwendenden

Der für einen dezentralen Arbeitsprozess der Softwaregestaltung typische individuelle softwaretechnische Zuschnitt zeigt sich nicht nur darin, dass ein EVU eine Software allein für sich gestaltet. Er zeigt sich auch darin, dass die Softwaregestaltung die individuelle Sicht der Anwendenden des EVU auf zweifache Weise berücksichtigt.

Erstens verhandelt der Arbeitsprozess die Priorisierung von Anforderungen des Managements und der Anwendenden. Das Management trägt seine Anforderungen – einige Befragte sprechen von Unternehmensanforderungen – an den Haupt-Product-Owner heran. Weil die Anforderungsmanager:innen näher an den Monteur:innen dran sind,

bringen sie entsprechend mehr Anforderungen der Anwendenden ein und votieren für deren Umsetzung. Auch wenn letztendlich der Haupt-Product-Owner über die Priorität von Unternehmens- oder Anwendenden-Anforderungen entscheidet, fließen individuelle Anforderungen der Anwendenden ein.

Zweitens berücksichtigt die Softwaregestaltung die Begrifflichkeiten aus dem Arbeitsalltag der Monteur:innen. Die unterschiedlichen Netzbereiche verhandeln einen Konsens, welche Begriffe auf der Softwareoberfläche auftauchen. Innerhalb des Quellcodes der Software verwenden Programmierende Begrifflichkeiten aus dem Arbeitsfeld der Monteur:innen des EVU, um sprachliche Hürden zu verringern¹¹.

8.3.2.2. INTERN2: gemeinsame Anforderungsrunde mehrerer Fachbereiche, dezentral

In dem Fall liegt ein dezentraler Arbeitsprozess der Softwaregestaltung für eine individuelle Software innerhalb eines EVU vor. Die Besonderheit liegt darin, dass hier, anders als bei INTERN1, mehrere Fachbereiche an der Softwaregestaltung beteiligt sind. Repräsentant:innen aus den Fachbereichen verständigen sich zentralisiert in einer Anforderungsrunde darüber, wie sie die Software gestalten. Aufgrund der fachbereichsübergreifenden Softwaregestaltung sind die kommunikativen Beziehungen durch hierarchische Hürden gehemmt, auf kooperative Fachbereiche angewiesen und wegen der zentralen Anforderungsrunde mit bürokratischen Abläufen konfrontiert. Die beteiligten Fachbereiche verwenden unterschiedliche Ticketsysteme, die aber teilweise dezentralen Input ermöglichen. Über den softwaretechnischen Zuschnitt entscheidet die Anforderungsrunde, so dass sich trotz individueller Softwaregestaltung im EVU die individuellen Wünsche einzelner Fachbereiche nicht immer durchsetzen.

Rollen: Scrum Master, Product Owner, Anforderungsmanager

Wie für einen dezentralen Arbeitsprozess der Softwaregestaltung typisch, besteht an die Rollen vor allem die Erwartung, Anforderungen aufzunehmen, und zwar sehr nah an und direkt mit den Anwendenden in den Fachbereichen.

In diesem Fall verteilen sich die Rollen auf verschiedene Fachbereiche. So ist z.B. der befragte Product Owner zu 100 % in dem dezentralen IT-Team des Fachbereichs Netzanschluss. Seine Aufgabe ist es, die Anwendungsprozesse inkl. Software und sein IT-Team weiterzuentwickeln, Anforderungen aufzunehmen und an der Anforderungsrunde teilzunehmen. Dann gibt es noch die Anforderungsmanagerin aus der Abteilung Zentralbereich. Sie arbeitet zusammen mit der IT-Abteilung Fachkonzepte zu einzelnen Tickets aus und testet die Umsetzungen. Wie auch bei vielen anderen Fallstudien gibt es die Rolle Key User:in, die den Anwendenden am nächsten ist. Wobei in diesem Fall die Möglichkeit besteht, dass alle Anwendenden sich einbringen. Der befragte Product Owner aus der dezentralen IT im Fachbereich Netzanschluss schätzt, dass für das reguläre Anfor-

11 Das nennt man Domain-Driven Design: Die Programmierung berücksichtigt die Wissensdomäne des Anwendungsbereichs, z.B. indem Fachbegriffe aus dem Anwendungsbereich in den Quellcode einfließen. Dadurch wird die Kommunikation zwischen Programmierenden und Anwendenden einfacher, weil sie die gleichen Begriffe verwenden.

derungsmanagement 80 % der 300 Anwendenden mindestens einmal an einem Jour fixe teilnehmen.

Der Scrum Master hat die Verantwortung für die Abstimmungsrunde der Fachbereiche: dass und wie sie abläuft und optimiert werden kann. Dabei nimmt er manchmal die Rolle des Mediators ein, wie es für die Rolle Scrum Master:in üblich ist.

»[N]atürlich gibt es auch Interessenskonflikte und da muss man natürlich schauen, dass man auf den größten gemeinsamen Nenner kommt oder im Idealfall vielleicht sogar auf eine geniale Idee, die dann alle Interessen unter einen Hut bringt. Aber der Input kommt von allen Seiten, genau. Die Moderation oder auch Mediation, falls erforderlich, die liegt dann schon bei mir.« (Scrum Master)

Dabei fällt stärker als bei INTERN1 auf, was allgemein für die Softwaregestaltung typisch ist: Die Ausgestaltung der Rollen erfolgt situativ angepasst, ändert sich im Zeitverlauf und anders als bspw. in Scrum vorgesehen, sind sie nicht immer klar spezialisiert. So koordiniert der Scrum Master zusätzlich die Beauftragungen und die Auslastung der externen Programmierenden und optimiert die Arbeit des programmierenden Teams. Letztendlich ist er sich gar nicht sicher, was er ist: IT-Koordinator, Scrum Master, agiler Coach oder Projektleiter, weil er sich nicht in einem reinen Scrum-Setting bewegt und seine Rolle nicht klar geschnitten ist. Auch die befragte Anforderungsmanagerin ist nicht nur mit Anforderungen beschäftigt, sondern nimmt auch Einstellungen an der ERP-Software vor, macht Datenbereinigungen und -auswertungen.

Im Unterschied zu INTERN1 übernehmen aufgrund der zentralen Anforderungsrunde der Product Owner oder die Anforderungsmanagerin auch koordinative Aufgaben. Sie schreiben Anforderungen nicht nur selbst, sondern sammeln sie auch und bereiten sie auf.

Ablauf: Anforderungsrunde, Refinement-Termine, IT-Teams in Fachbereichen, Tests

Der Ablauf besteht primär, wie für einen dezentralen Arbeitsprozess der Softwaregestaltung typisch, im Ausarbeiten und Sammeln von Anforderungen der Anwendenden. Wobei in diesem Fall aufgrund der zentralen Anforderungsrunde auch Verhandlungen Teil des Ablaufs sind. Aus Sicht des Scrum Masters liegt kein reines Scrum vor, sondern sie verwenden Bestandteile aus unterschiedlichen Ansätzen.

Wie dezentral die Softwaregestaltung ist, zeigt sich daran, dass der Arbeitsprozess Konzepte iterativ mit Anwendenden erarbeitet. Programmierende, Gestaltende und Anwendende arbeiten eng zusammen, was sich an kontinuierlichen Feedbackschleifen zeigt. So bringen Fachbereiche Vorkonzepte in die Anforderungsrunde ein, über die sich der Fachbereich und die Programmierenden bereits direkt ausgetauscht haben. Dadurch ist die eingebaute Anforderung näher an der späteren Umsetzung und es gibt kein Hin und Her mehr zwischen Anwendung und Programmierung, weil die Programmierenden nicht erst in der Umsetzungs- bzw. Sprintphase mit einem Konzept konfrontiert sind, sondern vorher bereits nachfragen können. Größere Fachbereiche haben dafür ein- oder zweimal die Woche feste Refinement-Termine. Der Scrum Master sitzt meistens auch in den Refinement-Terminen, leitet sie und bekommt so mit, was dort für Themen auftauchen. Kleinere Fachbereiche können in vorgegebenen Zeitfen-

tern Termine mit den Programmierenden vereinbaren. Bei kleineren Anforderungen ist ein direkter Austausch zwischen Fachbereich und Programmierenden möglich, um nachzufragen, ob ein Konzept so in Ordnung ist. Die Schleifen zwischen Fachbereich und Programmierung gehen so lange, bis der Reifegrad passt. Dann wird die Anforderung Teil eines Sprints. Neben diesen Feedbackschleifen bei der Konzeptionierung gibt es eine zweite Feedbackmöglichkeit durch Tests. Wenn ein Fachbereich Fehler entdeckt oder mit einer Umsetzung unzufrieden ist, geht das entsprechende Ticket zurück an die Programmierung.

Die dezentrale Softwaregestaltung zeigt sich zudem durch einen Ablauf, der aktiv und direkt auf Anwendende in den Fachbereichen zugeht. So nutzt das IT-Team des Fachbereichs Netzanschluss seine dezentrale Position, um Interviews durchzuführen. Dazu werden Mitarbeitende aus jedem Team eingeladen, die einzelnen Prozesse durchgesprochen und Änderungswünsche gesammelt. Die so erarbeiteten Konzepte kommen von den Anwendenden selbst und sie nehmen deren Umsetzungen gut an, so eine befragte Person. Nach der Umsetzung holt die Programmierung noch Feedback ein, ob die Umsetzung den Erwartungen entspricht.

Dass trotz der zentralen Anforderungs runde die dezentralen, individuellen Anforderungen der Fachbereiche eine Rolle spielen, zeigt sich daran, dass die Beteiligten gleichberechtigt sind. Zur Zeit der Befragung nahmen an der Anforderungs runde zwei Vertreter aus der IT (inkl. Programmierende), zwei aus der IT des Zentralbereichs und vier Vertretende aus den Fachbereichen¹² teil. Für eine Sitzung nimmt jede beteiligte Person ihre Anforderungen mit, die in der Woche angefallen sind. Die Anforderungen stellen die Vertretenden der Fachbereiche kurz inhaltlich vor und agieren damit in der Rolle von Product Owner:innen. Die einzelnen Fachbereiche sind dort gleichgestellt, verfügen über ein gemeinsames Budget und sollen sich demokratisch einigen.

»Ja, wenn man so will, sind ja die Vertreter der Fachbereiche die einzelnen Product Owner. Dafür haben wir natürlich keinen ›Ober Product Owner‹, sondern wir setzen im Moment darauf, dass sich die [...] [Abstimmungs runde] schon einigen wird. Und das hat bisher auch funktioniert, so eher demokratisch.« (IT-Manager Fachbereich)

So bereiten Konflikte keine größeren Probleme. Die Teilnehmenden haben manchmal unterschiedliche Vorstellungen und Interessen, was Anforderungen an die Software angeht. Dort können sie sich darüber austauschen und gemeinsame Lösungen finden.

Kommunikative Beziehungen: Kooperationsbereitschaft und direkte Kommunikation

Der Fall zeigt die für einen dezentralen Arbeitsprozess der Softwaregestaltungen typischen direkten und langfristigen Beziehungen intern in einem EVU, welche die oben genannten Rollen und Abläufe ergänzen. Allerdings existieren anders als bei INTERN1 Spannungen in den Beziehungen, weil mehrere Fachbereiche beteiligt sind. Die veränderten Erwartungen an die Beschäftigten im Zuge der Softwaregestaltung spiegeln sich

¹² Netzanschluss, IT-Zentralbereich (Work-Management und Messstellenbetrieb), Zählerwesen, Abrechnung.

darin nieder, dass es für manche Mitarbeitende schwer war, Kommunikation als Teil ihrer Arbeit zu begreifen.

Das EVU konnte den Arbeitsprozess der Softwaregestaltung erst etablieren, als die fehlende Kooperationsbereitschaft sowohl zwischen den Fachbereichen als auch zwischen IT-Abteilung und Fachbereichen beseitigt war. Ein Befragter bezeichnet es als »revolutionär«, dass, als sich der Fachbereich neu organisieren wollte, dieser zwei Vertretende der IT zu einigen Terminen dazu geladen hat, aus denen dann die Abstimmungsrunde hervorging. Dass es so gut geklappt hat, lag daran, dass die Beteiligten keine vorbelasteten Beziehungen hatten:

»[W]eil sie die Historie nicht mit sich herumgeschleppt haben, wer da mal vor zehn Jahren wem irgendwie ans Schienbein getreten ist oder irgendwie was weggenommen oder irgendwas Blödes gemacht hat.« (IT-Manager Fachbereich)

Allgemein gibt es persönliche Netzwerke zwischen den Fachbereichen und der IT-Abteilung. Als die IT noch ausgelagert war, hat das Management versucht, dass Fachbereiche mit ihr nur noch über formal festgelegte Wege kommunizieren. Jetzt ist es wieder in Ordnung, sich persönlich zu kennen und anzusprechen. Der Scrum Master beschreibt die direkte Kommunikation als effizienter, weil es kein Hin und Her zwischen den Beteiligten gibt:

»Aber wir haben halt versucht, gerade halt möglichst mehr auf eine direkte Kommunikation umzusteigen, weil gerade in der Vergangenheit gab es dann in den Tickets häufig das Phänomen, dass dann so ein Frage-Antwort-Pingpongspiel hin und her ging. Und der Durchschnitt war da tatsächlich, dass wir zu einem Ticket so 30 Kommentare hatten.« (Scrum Master)

Die Anforderungsmanagerin aus der Zentralbereichs-IT meint, dass sie bestimmte Themen mit den Programmierenden direkt bespricht. Entweder macht sie das per E-Mail, per Telefon oder, wenn das nicht ausreicht, persönlich.

Um sich direkt auszutauschen, müssen Mitarbeitende dazu bereit sein. Einige Befragte weisen darauf hin, dass dies nicht selbstverständlich ist. So haben sich ältere Programmierende anfangs nicht gern in agilen Teams ausgetauscht. Mit Scrum verbinden sie viele Termine, Kaffeekränzchen und dass man nicht zum Arbeiten kommt. Einige Mitarbeitende wollen lieber vor sich hinarbeiten, als sich regelmäßig mit anderen abzustimmen. Im Gegenteil dazu sieht die Anforderungsmanagerin Kommunikation als selbstverständlichen Teil ihrer Arbeit an.

Digitale Werkzeuge: mehrere Softwarelösungen, um Anforderungen zu sammeln

In dem Fall zeigt sich in dem EVU die für digitale Werkzeuge typische Verwendung in dezentralen Arbeitsprozessen der Softwaregestaltung in Extremform: Es geht primär darum, Anforderungen zu sammeln, und weniger um Kontrolle oder Koordination der gesamten Softwaregestaltung. Denn es gibt unterschiedliche Ticketsysteme, Web-Formulare und Exceldateien, mit denen die Fachbereiche Anforderungen für die Anforderungsrunde sammeln. Wegen dieser disparaten Landschaft an Werkzeugen existiert kein in-

tegriertes digitales Gesamtsystem, weswegen die verwendeten digitalen Softwaretools nicht zur zentralen Kontrolle taugen. Vielmehr wenden die Beteiligten pragmatisch dezentral an, was sie kennen, können und ihnen nützlich erscheint – Hauptsache, sie können Anforderungen einfach sammeln. Allerdings gibt es zentral in der Anforderungsrunde eine Liste, damit den Teilnehmenden transparent ist, welche Anforderungen wie priorisiert sind. In Zukunft sollen aber alle das gleiche Ticketsystem verwenden.

Softwaretechnischer Zuschnitt: größter Nutzen für das Unternehmen

Der für einen dezentralen Arbeitsprozess der Softwaregestaltung typische individuelle softwaretechnische Zuschnitt zeigt sich auch in diesem Fall. Die zentrale Anforderungsrunde übt jedoch einen Konsens- und Standardisierungsdruck auf die einzelnen Fachbereiche aus. Deren individuellen Vorstellungen sind dadurch Grenzen gesetzt.

Mehr noch als bei INTERN1 muss sich die individuelle Sicht der Anwendenden des EVU gegen die Unternehmenssicht durchsetzen, weil zentral mehrere Repräsentant:innen der Fachbereiche über Anforderungen und deren Priorisierung verhandeln. Maßgeblich kann z.B. sein, wie viele Anwendende eine Anforderung betrifft oder wie teuer eine Umsetzung wird. Wie wichtig die Priorisierung ist, zeigt sich an den ca. 100 Tickets im Backlog der Anforderungsrunde, die noch umzusetzen sind.

»Also, ich sag mal, in der Abstimmungsrunde sitzen eben die Leute drinnen, die wissen, wir müssen als Gesamtunternehmen effizient sein. Ich kann nicht jedes Wünsch-Dir-Was-Thema einsteuern. Wir sind aber auch keine Anwender. Wenn ich bei mir im Fachbereich priorisiere, da sitzt ein persönlicher Leidensdruck oft dahinter, dass er sagt: Mich stört das täglich in meiner Arbeit. Betrifft aber vielleicht nur fünf Mitarbeiter von den 300. Manche Themen die betreffen alle und das in Waage zu halten, das ist, ja, eine Herausforderung.« (Product Owner IT-Team Netzanschluss)

Die dezentrale Gestaltung einer individuellen Software in einem EVU zeigt sich zudem daran, dass laut einem Befragten das genutzte System stark vom SAP-Standard abweicht. Ein Teil des Arbeitsprozesses ist es, zu verhandeln, ob ein Fachbereich eine Standardausprägung von SAP nutzt oder diesen Standard anpasst. Ein Grund für eine individuelle Anpassung kann sein, dass es mehr kosten kann, wenn das EVU beim Standard bleibt. Das ist dann der Fall, wenn dadurch das EVU mehr Personal braucht, weil die Software nicht auf die firmenindividuellen Prozesse angepasst ist.

Wie auch bei INTERN1 zeigt sich die individuelle Softwaregestaltung daran, dass der dezentrale Arbeitsprozess auf EVU-eigene Begriffe Rücksicht nimmt und einen Konsens über sie verhandelt. Die von der Softwaregestaltung betroffenen Anwendenden im Netzanschluss sind zahlreich und in unterschiedlichen Netzgebieten tätig (und damit regional weit verteilt), weswegen die Mitarbeitenden teilweise die gleichen Dinge mit anderen Namen bezeichnen.

8.3.2.3. KOOP1: zentralisiertes Anforderungsmanagement, um Standard zu verhandeln

Anders als bei INTERN1 und INTERN2 geht es in dieser Fallstudie in erster Linie darum, dass mehrere EVU die Möglichkeit der Softwaregestaltung für einen gemeinsamen

Standard nutzen wollen. Dafür gibt es einen zentralen Arbeitsprozess der Softwaregestaltung in Form eines Anforderungsmanagements, welches das IT-DL organisiert und bei dem mehrere EVU mitmachen. Jedoch ist das Besondere an dem Fall, dass die EVU verhandeln, was zum Standard gehört und was sie individuell gestalten. Dementsprechend erfolgt auch die Softwaregestaltung dezentral in und für einzelne EVU und, wie für diese Form der Softwaregestaltung typisch, die Anforderungsaufnahme direkt mit den Anwendenden. Aufgrund der Marktbeziehung zwischen IT-DL und EVU herrscht eine Spannung in den kommunikativen Beziehungen: einerseits aus partnerschaftlich-kooperativem Vorgehen für den gemeinsamen Standard und andererseits nur so lange zu kooperieren, wie es dem einzelnen EVU nützt, und z.B. Vorteile aus individuellen Alleingängen zu ziehen. Die eingesetzten digitalen Werkzeuge erledigen sowohl die typischen Zwecke einer dezentralen als auch zentralisierten Softwaregestaltung. Sie dienen dazu, die Verhandlungen zwischen den Organisationen transparent zu halten, zu koordinieren, die Umsetzung des IT-DL zu kontrollieren und dezentral Input in den EVU zu ermöglichen.

Rollen: Anforderungsmanagende, IT-Beratende, Key User:innen, Anwendungsbetreuende & Co.
– zwischen und innerhalb der Organisationen

Wie für einen zentralen Arbeitsprozess der Softwaregestaltung typisch, besteht an die Rollen vor allem die Erwartung, die Anforderungserarbeitung über Organisationsgrenzen hinweg zu koordinieren. Nur so ist eine stetige Verhandlung darüber möglich, was zentral das IT-DL oder dezentral die EVU gestalten. Doch gibt es Unterschiede zwischen den EVU, inwiefern sie sich a) rein auf die kooperativen, zentralisierten Anforderungsprozesse des IT-DL ausrichten oder b) selbst Rollen haben, die eine eigenständige, dezentrale Softwaregestaltung erlauben.

Für die Koordination zwischen IT-DL und EVU gibt es sowohl innerhalb des IT-DL als auch in den EVU Rollen. Innerhalb des IT-DL gibt es mehrere Rollen, welche die zentralisierte Softwaregestaltung koordinieren. So gibt es ein neunköpfiges Team für kooperative Projekte, wovon zwei für das Anforderungsmanagement zuständig sind. Daneben gibt es extra Rollen, um die Beziehungen zu den EVU zu pflegen: die Key Account Managenden. Über sie läuft alles Kaufmännische wie z.B., wenn das Budget für eine Anforderung nicht ausreicht und EVU und IT-DL sich auf das weitere Vorgehen einigen müssen. Für die Übersetzung zwischen IT- und Energiewirtschaft existieren beim IT-DL IT-Beratende, die spezialisiert sind auf einzelne Fachgebiete wie Stammdaten, Abrechnung, Ablesung oder Instandhaltung. Sie schreiben Konzepte, unterstützen Programmierende mit fachlichem Wissen oder beraten die EVU. Jede beratende Person hat zudem als Aufgabe, bei Anforderungen zu prüfen, welche EVU sie betreffen bzw. für welche sie interessant sein könnten.

Innerhalb der EVU gibt es für das vom IT-DL koordinierte Anforderungsmanagement zuständige Mitarbeitende. Ihre Arbeit zeigt, dass die zentralisierte Standardgestaltung in diesem Fall auf dezentrale Anforderungen der Anwendenden Rücksicht nimmt und sie in den Arbeitsprozess der Softwaregestaltung einbindet. Der Anforderungsmanager von EVU2 kümmert sich um die kaufmännische Abwicklung und

sammelt Anforderungen – sowohl für den Netz- als auch für den Vertriebsbereich seines EVU.

Ebenso innerhalb der EVU sind die Key User:innen für das zentrale Anforderungsmanagement aktiv. Sie sind noch näher an den Anwendenden dran, auch selbst Anwendende der Software und sie speisen Anforderungen in das Anforderungsmanagement des IT-DL ein. Ein befragter Teamleiter (EVU2) hat in seinem Team neben Key User:innen und sechs Sachbearbeitenden noch eine zusätzliche Rolle, die an der Softwaregestaltung teilnimmt: die Fachkoordinator:innen. Sie arbeiten übergreifender als Key User:innen, stärker koordinativ, strategische und fachliche Diskussionen und Umsetzungen begleitend. In EVU2 macht ein befragter Teamleiter am Anforderungsmanagement mit. Manche EVU haben keine Key User:innen und stattdessen übernimmt das IT-DL deren Aufgaben.

Die EVU verfügen zudem über Rollen, mit denen sie dezentral und unabhängig vom zentralisierten Anforderungsmanagement im IT-DL selbst Software gestalten. So gibt es im EVU2 einen Anwendungsbetreuer, der die von ihm betreuten Softwareanwendungen unabhängig vom IT-DL gestaltet: Er leitet Projekte, macht Prozessanalysen, macht Ausschreibungen und nimmt Anforderungen auf. EVU2 ist allgemein ein Beispiel dafür, dass die Verhandlung von Individual- und Standardsoftwaregestaltung auch innerhalb der EVU stattfindet. Der befragte Anwendungsbetreuer aus EVU2 meint, dass im EVU aufgrund von Sparprogrammen vor einigen Jahren IT-Querschnittsfunktionen wie seine abgebaut wurden, weil sie kein Geld bringen würden. Nun entstehen sie wieder, weil man deren Notwendigkeit erkannt hat. Laut dem Digitalisierungsmanager von EVU2 agieren die Fachbereiche unterschiedlich: Die einen binden die IT stärker ein und arbeiten mehr mit ihr zusammen. Andere stellen selber Key User:innen ein und bauen ERP-Know-how auf, um eigenständig Einstellungen an der Software vornehmen zu können. Aus seiner Sicht gibt es einen Wellenverlauf zwischen De- und Zentralisierung.

Der Prozessmanager aus dem Vertrieb von EVU3 koordiniert dezentral die Gestaltung des CRM-Moduls (Customer-Relationship-Management¹³). Wie für dezentrale Arbeitsprozesse der Softwaregestaltung typisch, fließen die individuellen Anforderungen von Anwendenden ein:

»Also, zum großen Teil kommen diese Anforderungen von den Anwendern. Weil letztlich sie ja in dem System arbeiten und dort Probleme und Anforderungen, um den Prozess zu optimieren, besser einsehen können.« (Prozessmanager EVU3)

Er prüft deren Anforderungen und den Ist-Zustand der Software, spricht mit dem IT-DL und internen Mitarbeitenden und schreibt Anforderungen für das Anforderungsmanagement, die dann bei den Programmierenden landen. Zudem informiert er Anwendende, was geändert wurde, oder organisiert bei größeren Veränderungen Schulungen.

Ein letztes Beispiel für dezentrale Softwaregestaltung ist die Prozessmanagerin aus EVU1. Sie arbeitet eigenständig mit den IT-Beratenden des IT-DL zusammen, tauscht

¹³ Software zur Erfassung und Verwaltung von Daten der Kundschaft vor allem zu vertrieblichen Zwecken (bspw. um Marketingaktionen durchzuführen).

sich mit Programmierenden aus und spricht Softwarekonzepte mit den involvierten Anwendenden durch, damit diese die Veränderungen verstehen und sinnvoll finden. Zugleich nimmt sie selber Einstellungen an der Software vor.

Ablauf: zentral organisiertes Anforderungsmanagement und dezentrale Softwaregestaltung

Der Ablauf besteht primär daraus – wie für einen zentralisierten Arbeitsprozess der Softwaregestaltung typisch –, über Anforderungen zu verhandeln, Konflikte zu lösen und Feedback von verschiedenen EVU einzuholen. Wobei in diesem Fall trotzdem auch dezentral die EVU von den Anwendenden Anforderungen aufnehmen. So gibt es aufgrund der Mischung aus zentraler und dezentraler Softwaregestaltung Feedbacks zwischen Programmierenden, Anwendenden und Softwaregestaltenden in einem weitverzweigten Gestaltungsnetzwerk.

Die für eine zentralisierte Softwaregestaltung typischen Konflikte zwischen den EVU moderiert das IT-DL im Anforderungsmanagement auf operativer und strategischer Ebene. Auf der operativen übernehmen das die Anforderungsmanagenden des IT-DL. Dort bringen die EVU ihre jeweiligen Erfahrungen und Sichtweisen in die Diskussion über die Anforderungen ein. Dann teilt jedes EVU mit, welche Anforderungen es haben will und welche nicht – je mehr EVU eine Anforderung gemeinsam umsetzen, umso günstiger wird es. Alle zwei Wochen treffen sich die EVU dafür in einer Telefonkonferenz. Auf der strategischen Ebene des Anforderungsmanagements ist ein professioneller Mediator aktiv. Der Mediator kann durch seine Neutralität zwischen den verschiedenen Interessen des Dienstleistungsunternehmens und der EVU vermitteln. Die EVU müssen sich auf gemeinsame Ziele einigen, wenn sie einen Standard gestalten wollen. Die Treffen behandeln u.a. größere Themen oder Projekte wie z.B. eine Umstellung auf eine neue Version des ERP-Systems.

Das Besondere an dem Fall ist nun, dass nicht einfach Fachleute zentral einen Standard gestalten, sondern dezentral die EVU von den Anwendenden Anforderungen sammeln. Dafür müssen die EVU intern für sich herausfinden, wie sie die Möglichkeiten der Softwareentwicklung nutzen wollen: Welche Anforderungen nur für sie und welche für das zentrale Anforderungsmanagement relevant sind und damit in den kooperativen Standard einfließen. Die EVU müssen für sich herausfinden, wie sie sich intern organisieren, wenn sie für die Softwaregestaltung mit mehreren Organisationen zusammenarbeiten. Das EVU2 macht jede Woche Anforderungsmanagementrunden mit den Anforderungsmanagenden aus den unterschiedlichen Bereichen. Dort tauschen sich die beteiligten Personen über Anforderungen aus, die für das zentrale Anforderungsmanagement relevant sind. Daneben gibt es ca. einmal im Monat Key-User-Runden, in denen sich die Key User:innen der Fachbereiche treffen, um z.B. abzustimmen, was sie gemeinsam gestalten.

Gleichzeitig gibt es für Projekte ohne das IT-DL dezentral eigenständige Abläufe zur Softwaregestaltung. Der verantwortliche Applikationsbetreuer von EVU2, der für die Software zur Instandhaltung zuständig ist, führt als Projektleiter Workshops durch, um Anforderungen aufzunehmen. Er sieht agile Anleihen in seiner Art des IT-Projektmanagements: Die Arbeitspakete ergeben sich iterativ, es gibt vierwöchige Sprints inklusive Reviews und Retrospektive, wie es in Scrum üblich ist. Das heißt, er arbeitet

direkt Anforderungen für Programmierende aus und stimmt sich mit diesen ab. So existiert dezentral ein direkter Feedbackmechanismus zwischen Anwendung und Programmierung. EVU3 koordiniert intern für sich die Gestaltung einer Softwarelösung zur Betreuung seiner Kundschaft, die sowohl das IT-DL als auch zwei Softwarefirmen entwickeln. Das heißt, intern muss das EVU nicht nur Anforderungen aufnehmen, sondern auch die unterschiedlichen Organisationen koordinieren, welche die Programmierung übernehmen. Weil mehrere Fachbereiche des EVU betroffen sind, hat das EVU intern ein Anforderungsmanagement etabliert, damit nicht alle betroffenen Mitarbeitenden direkt ein Ticket an das IT-DL schreiben und sich die verschiedenen Teams stattdessen absprechen, was die Anforderungen anbelangt. Der befragte Prozessmanager spricht von einer »stadtwerkemodifizierten Scrum-Lösung«.

Der Grund für die dezentrale Softwaregestaltung innerhalb der EVU liegt auch darin, dass der zentralisierte Arbeitsprozess zu bürokratisch ist:

»Aber wir machen eigentlich alles, was prozessual selber durch uns zu heben ist, selber. Weil erstens schneller, günstiger und ja, runder irgendwie. Sonst muss man immer Tickets schreiben, Ressourcen anfordern, feststellen, dass man die Ressource erst in Wochen bekommt, dann testen und irgendwann freigeben. Da wo es halt geht, machen wir es halt möglichst agil selber.« (Teamleiter EVU1)

Typisch für eine zentralisierte Softwaregestaltung und Teil der Feedbackschleife zwischen Anwendung und Programmierung sind die vom IT-DL zentral koordinierten Tests. Sie sind ein wichtiger Teil der Softwaregestaltung, weil sie Rückmeldung darüber geben, ob eine Umsetzung den Vorstellungen der EVU entspricht. Neben dem gemeinsamen Anforderungsmanagement testen die EVU und das IT-DL gemeinsam. Das IT-DL hat extra Rollen für das Testmanagement, um die Tests zu koordinieren. Dezentral hat z.B. EVU2 selber noch eine Person, die Testfallkataloge schreibt und sie an die Fachbereiche verteilt.

Kommunikative Beziehungen: partnerschaftlich und bei Bedarf direkt

In dem Fall existieren die für einen zentralen Arbeitsprozess der Softwaregestaltungen typischen bürokratischen und spannungsgeladenen marktförmigen Beziehungen, welche die Kommunikation erschweren. Doch sorgen neben Mediator, kooperativen Rollen und Abläufen kommunikative Beziehungen dafür, dass der Arbeitsprozess der Softwaregestaltung trotzdem gelingt. So gibt es langfristige Beziehungen auf Augenhöhe und direkte Kommunikation trotz formal-bürokratischer Abläufe. Trotz Marktbeziehungen zwischen EVU und IT-DL sprechen Beteiligte von einer Familie und von einem Geben und Nehmen.

Über das IT-DL Software zu gestalten, bringt einen langwierigen, formalen Prozess mit sich, an den sich die beteiligten Rollen ausrichten müssen:

»Ticket, Angebot schreiben, Change im System, Alphatest, Kundentest, Go-Live, Stabilisierung. Ich habe ja jedes Mal eine Destabilisierung des Systems« (Key Account Manager).

Solche Hürden der Softwaregestaltung ergeben sich auch durch interne Prozesse in den EVU. Der befragte Applikationsbetreuer aus EVU2 meint, dass es dauert, bis Anforderungen durch die interne Hierarchie gehen, weil die Befehlskette für Abstimmungen zu berücksichtigen ist.

Kommunikative Beziehungen wie direkte Kontakte verringern diese Kommunikationshürden. Ein befragter Programmierer arbeitet zwar viel via Ticketsystem und ohne direkten Kontakt zu den EVU. Gleichzeitig ist er im Gegensatz zu früher wieder telefonisch erreichbar. Er ruft manchmal direkt bei einem EVU an und diese wiederum können ihn direkt auf seinem Diensthandy erreichen. Das erleichtert seiner Meinung nach die Abstimmung.

»Mittlerweile ist mir das aber lieber, wenn ich auch direkten Kundenkontakt habe. Ich rufe dann auch manchmal direkt an. Ich habe ein Diensthandy. Die Leute sehen dann auch meine Mobilfunk-Nummer und können mich bei Nachfragen oder auch direkt erreichen. Das erleichtert meiner Meinung nach die Tätigkeit, die Abstimmung.« (Programmierer)

Der andere befragte Programmierer2 wird bei bestimmten Themen (für die er bekannt ist) von den EVU direkt angesprochen und danach wird eine Anforderung aufgenommen. Ein Prozessmanager aus einem EVU meint:

»Man hat einfach diesen Stille-Post-Effekt. Wenn ich eine Anforderung aufgebe, [...] auf der [IT-DL]-Seite einen Berater gibt, der diese Anforderung aufnimmt und diese dann mit dem Entwickler bespricht, dann ist es halt manchmal so, je nachdem, wie weit schon der andere den anderen kennt oder halt eben auch nicht, dass manchmal nicht das Richtige dabei rauskommt. Und ich glaube schon, dass es einfacher wäre, wenn wir im Haus intern eine Entwicklungsabteilung hätten und man dann direkt mit den Entwicklern spricht, denen die Anforderung geben könnte. Dann hätte man halt eine Schnittstelle weniger.« (Prozessmanager EVU1)

In den Interviews werden auch verschiedene Elemente der Kommunikation genannt, die für die Zusammenarbeit hilfreich sind, wie Offenheit und professionelle Kommunikation:

»Das wichtigste Konzept in unserem Bereich ist wirklich die Offenheit, da wirklich auch Leute zu integrieren, mitzunehmen, abzuholen, die immer wieder auch... Es gibt ja immer diese schöne Bringschuld. Das wird bei uns nicht funktionieren, wenn nicht jeder auch wüsste, es kann wichtig sein für die Kollegen, die oder die Veränderung zu wissen. [...] Wie gesagt, bei uns läuft alles über Offenheit.« (Programmierer)

Ein befragter Betriebsrat betont die Erwartungen an die Softwaregestaltenden, nicht nur das IT-System zu betreuen, sondern den Anwendenden zuzuhören, zu verstehen, zu übersetzen, Lösungen anzubieten, die Sprache der Anwendenden zu sprechen. Das hätte sich seit Mitte/Ende der 2010er Jahre verändert.

»Dieses Verstehen, das einander verstehen und zuhören. Und das ist etwas, was meine Erfahrung ist: Also, bei uns hat das was mit Menschen zu tun. Wir haben die richtigen Leute eingestellt in den letzten Jahren und die sind extra..., die sind Kommunikationsprofis, auch IT-Profis« (Betriebsrat)

Neben dem direkten Austausch ist die langfristige Zusammenarbeit eigentlich charakteristisch für eine dezentrale Softwaregestaltung. Sie spielt aber auch in diesem Fall eine Rolle. Ein EVU hat mal das IT-DL gewechselt, ist aber wiedergekommen. Es hat erst beim neuen Dienstleistungsunternehmen gemerkt, was es am alten und dessen Wissen über das eigene EVU hatte. Es geht auch darum, ein gemeinsames Verständnis entwickelt zu haben:

»Naja, je besser man mit den Personen sprechen kann und je besser man sich kennt, in der Regel ist es auch, umso besser werden dann auch die Anforderungen umgesetzt. [...] Weil man einfach eine andere Sprache spricht, wenn man die gleichen Leute und wenn man weiß, wie die sozusagen ticken und die auch wissen, wie man selber so tickt. Ich glaube schon, je länger man miteinander zusammenarbeitet, dass, umso besser wird das Ergebnis. Vorausgesetzt, dass man gut miteinander zusammenarbeitet und nicht gegeneinander.« (Prozessmanager EVU1)

Das spielt auch bei individueller Softwaregestaltung dezentral für ein EVU eine Rolle. So kennt durch die lange Zusammenarbeit das IT-DL die individuellen Anpassungen im Bereich CRM für das EVU3 sehr gut. Das IT-DL war bereits bei der Einführung dabei. Zudem sind die IT-Beratenden fest den EVU zugeteilt und zugleich programmieren sie auch. Laut dem befragten Prozessmanager (EVU3) ist deswegen keine detaillierte Beschreibung bei Anforderungen notwendig. Er hat mit dem CRM-Team beim IT-DL ein kollegiales Verhältnis.

Die EVU und das IT-DL arbeiten trotz Marktbeziehungen kooperativ zusammen und sprechen von einer Art Familie. Denn trotz Konkurrenz und Kostenkalkül arbeiten sie partnerschaftlich zusammen und kennen sich untereinander. Die Kooperation ist partnerschaftlich, weil es ein Geben und Nehmen ist. Das kooperative Verhalten zeigt sich auch daran, dass die EVU die in den SLA festgelegten Möglichkeiten nicht ausreizen, Strafen einzutreiben.

»Das ist ein sehr schönes Arbeiten, muss ich sagen, mit dem Kunden. Ich habe vielleicht auch das große Glück, dass ich auf der anderen Seite auch, sagen wir mal, Menschen habe, die sehr angenehm sind. Es ist aber auch bei anderen Kunden, wo es manchmal vielleicht etwas hitziger zugeht, wie ich so bei den Kollegen höre, ist es aber immer die Situation und der Alltag getrieben eigentlich von der Einstellung: Es muss ein Geben und Nehmen sein. Selbst bei den SLA: Es gibt natürlich SLA-Verstöße. Es sitzt bei keinem Kunden einer, der nur darauf wartet und sich freut und dann sofort die Rechnung und den Taschenrechner raus und sagt: ›Hier drei SLA-Verstöße [...] Ich hätte gerne 15500 Euro für den Monat.‹ Diese Dinge werden immer erst besprochen, die Gesamtgemengelage, Ursachen, vielleicht schwierige Rahmenbedingungen mit besprochen, Herstellerqualität, Corona: Das kommt alles jetzt, auch wenn es jetzt ak-

tuell, das kommt alles zusammen und das ist ein Geben und Nehmen.« (Key Account Manager)

Es kommt vor, dass sich EVU unabhängig vom IT-DL untereinander absprechen und Änderungen vornehmen:

»... wir auch teilweise Kontakt zu den anderen Häusern aufnehmen, Ideen aufzugreifen, weiterzuverfolgen oder abzustimmen. Die Stadtwerke so und so hat schon eine Lösung dafür. Dann machen wir einfach einen Termin mit denen. Ich nehme meine Leute dazu, wie die das gebaut haben und überlegen uns noch, ob wir das auch wollen.« (Teamleiter EVU1)

Digitale Werkzeuge: mehrere Ticketsysteme und ERP-Entwicklungsumgebung

So wie der Fall zwar um eine zentralisierte Softwaregestaltung kreist, aber die dezentrale Softwaregestaltung in den EVU zulässt, so ist auch die Verwendung der digitalen Werkzeuge eine Mischung aus beiden Typen. Die digitalen Werkzeuge verwenden die EVU sowohl typisch für zentrale Arbeitsprozesse, um Abstimmungen und Transparenz gewährleisten, als auch für direkten Input durch Anwendende, was typisch für eine dezentrale Softwaregestaltung ist. Die kooperativen Beziehungen zwischen IT-DL und EVU drücken sich darin aus, dass direkte E-Mails an einzelne Mitarbeitende eine große Rolle spielen.

So ist im vom IT-DL bereitgestellten MS Sharepoint für sämtliche EVU einsehbar, welche Anforderungen im Anforderungsmanagement für den gemeinsamen Standard vorliegen und was die Umsetzung jeweils kostet. Das MS Sharepoint dient zur Abstimmung darüber, wer welche Anforderungen haben will und bezahlt. Die Ergebnisse der Abstimmungen inklusive der gesammelten Anforderungen sind dort ebenso dokumentiert.

Zusätzlich betreibt das IT-DL ein eigenes Ticketsystem für die gesamte Kommunikation und Dokumentation von Tickets. Das betrifft kleinere Anforderungen und Störungen bezüglich der Software. Die EVU können online Tickets aufgeben. Wenn die EVU für ein Ticket die betroffene Software und das fachliche Thema auswählen, dann ordnet das Ticketsystem das Ticket automatisch dem verantwortlichen Team beim IT-DL zu. Über das Ticketsystem können die EVU aber auch das IT-DL kontrollieren, z.B. wie lange eine Umsetzung gedauert hat.

Der Prozessmanager der dezentralen Softwaregestaltung im EVU3 nutzt nicht das Ticketsystem des IT-DL. Wie für dezentrale Arbeitsprozesse der Softwaregestaltung typisch, haben die digitalen Werkzeuge primär die Funktion, dass Anwendende mitgestalten können. Dafür setzt der Prozessmanager die MS-Office-365-Lösung ein, welche die Mitarbeitenden gut kennen. Dort gibt es einen MS-Teams-Kanal, der direkten Austausch und Input ermöglicht.

Soziotechnischer Zuschnitt: zentrale Synergien durch gemeinsamen Standard und dezentrale Abweichungen

Weil in dem Fall die EVU für die gesamte industriespezifische Softwarelandschaft verhandeln, was sie als Standard gestalten und was individuell, zeigt sich hier besonders,

wie der softwaretechnische Zuschnitt zustande kommt. So prägen a) institutionelle Unterschiede der Geschäftsbereiche Vertrieb und Netz die Möglichkeiten eines Standards. Es bestehen aber auch b) unabhängig davon Unterschiede zwischen den Firmen, inwiefern sie individuell gestalten. Zudem müssen die EVU c) intern die Kompetenzen haben, um überhaupt Synergien erkennen zu können.

Zu a): Allein durch die institutionelle Begebenheit, dass Prozesse im Netzbereich stärker durch die Regulierung bestimmt sind, gibt es dort in den EVU weniger individuelle Ausprägungen in der Softwaregestaltung. Im Vertriebsbereich ist ein Konsens bei Anforderungen schwieriger, weil individuelle Vorstellungen bei den einzelnen EVU vorliegen. Laut einem Anforderungsmanager (EVU2) geht auf der Vertriebssseite immer mehr zurück, auf was man sich in der Kooperation einigen kann. Wenn der Vertrieb sagt, er braucht das und das für den Markt, dann ist es schwer, den Standard zu halten, vor allem wenn dafür Geld in den EVU da ist.

»Und das sind ja Vertriebe und das können Sie sich sicherlich vorstellen, die Vertriebe sind ja teilweise völlig unterschiedlich ausgerichtet.« (Anforderungsmanager EVU2)

Ein Beispiel dafür sind die verschiedenen Strom- und Gastarife, die man der Energiekundschaft anbietet. Auf der Netzseite ist der Gestaltungsspielraum durch die Regulierung eingeschränkt, weshalb es dafür in der Kooperation standardisierte Tarife gibt. Im Vertrieb sind die Produkte individueller, weswegen einen Standard auszuprägen schwieriger ist. Ein Befragter aus dem EVU1 nimmt eigenständig Einstellungen an der Software vor, um der Kundschaft des EVU unterschiedliche Produkte anbieten zu können.

Zu b): Neben den Unterschieden zwischen den Geschäftsbereichen gibt es auch noch firmenbedingte. Ein befragter Betriebsrat (EVU4) meint, dass früher das EVU sehr viel individuell entwickelt hat und man jetzt trotz einzelner individueller Programmierungen auf den Standard durch das IT-DL setzt.

»Man hat das bei uns in diesem Projekt immer genannt: Hier gibt es blau, grün, gelb karierte Maiglöckchen für jeden Anwender. Also, es wurde alles versucht, in SAP zu programmieren. Selbst der kleinste Fall. [...] Das hat uns immer sehr viel Geld gekostet. [...] Jetzt haben wir natürlich sehr viel auf Standard gesetzt.« (Betriebsrat EVU4)

Keines der befragten EVU sagt, dass es rein auf den Standard setzt. Ein Befragter meint, dass bei der Einführung versucht wurde, am Standard zu bleiben, und die Fachabteilungen argumentieren mussten, warum eine individuelle Anpassung notwendig sei. Beispiele dafür, dass EVU vom Standard abweichen, sind eine unabhängig ausgewählte und angepasste Instandhaltungssoftware (EVU2) oder unterschiedliche Lösungen für das Energiedatenmanagement (EVU1). Bei EVU3 hat die starke Orientierung an der Kundschaft dazu geführt, dass es ein individuelles Portal für seine Kundschaft entwickelt hat:

»Wir haben versucht, es immer unseren Kunden recht zu machen. Das führt dazu, warum der Standard sehr umgebaut worden ist zu einem sehr individuellen Produkt.« (Prozessmanager, EVU3)

EVU3 nutzt das Modul von SAP für das Customer-Relationship-Management (CRM) dafür ganz anders als die anderen EVU der Kooperation. EVU3 will es für Privat-, Geschäftskundschaft und den ÖPNV nutzen und regionale Besonderheiten berücksichtigen. Es gestaltet lieber dezentral selbst, weil die Abstimmung mit anderen EVU zu aufwendig ist. Anders als andere EVU setzt EVU3 seit längerem auf das CRM-Modul von SAP.

»Entweder müssten alle anderen Häuser diese Entwicklungsschritte gehen, die wir gehen, oder wir müssten die Entwicklungen, die wir haben, irgendwie erst einmal beiseite stellen und sie in dieser Kooperation mit den anderen Partnern entwickeln, obwohl wir sie eigentlich schon haben (lacht). Und da ist halt die Frage, inwieweit da tatsächlich dann ein Mehrwert ist oder ob es da nicht kostengünstiger in Anführungsstrichen ist, wenn wir für uns selbst eine Plattform aufstellen.« (Prozessmanager EVU3)

Zu c): Unabhängig von solchen Unterschieden war es und ist es ein langwieriger Prozess, die Fähigkeiten zu entwickeln, um Synergien zu erkennen. Im EVU2 wurde die Rolle Prozessberatung 2020 geschaffen, um bereichsübergreifende Aufgaben zu erledigen. Sie hat nicht nur die Aufgabe, Gestaltungsprojekte durchzuführen. Sie soll auch mögliche Synergien feststellen und Entscheidungsvorlagen für übergreifende Optimierungen erarbeiten. Die Rolle soll dafür sorgen, dass Fachbereiche nicht unabhängig voneinander Anforderungen an das IT-DL stellen, sondern der Gesamtzusammenhang berücksichtigt wird. Durch den Blick auf die Organisation von den digitalen Prozessen und nicht allein von der energiewirtschaftlichen Fachlichkeit her hat das EVU eine neue Perspektive gewonnen. Erst dadurch ist aufgefallen, dass Anforderungen aus unterschiedlichen Fachbereichen zusammenhängen und z.B. Synergien möglich sind.

8.3.2.4. KOOP2: EVU und IT-DL zwischen dezentraler und zentraler Softwaregestaltung

Wie bei KOOP1 nutzen die Organisationen in diesem Fall sowohl die Möglichkeiten der individuellen als auch der Standardsoftwaregestaltung. Doch es gibt einen Unterschied: Es gibt keine zentralisierte Verhandlung darüber beim IT-DL und nur einen kleinen Teil der Softwaregestaltung erledigt das IT-DL zentral für die EVU in Form eines gemeinsamen Standards. Der Grund für die fehlende Zentralisierung liegt darin, dass IT-DL und EVU die Spannungen aufgrund der Marktbeziehungen nicht immer beseitigen konnten. Deshalb etablieren einzelne EVU wie EVU2 verstärkt selbst dezentrale Arbeitsprozesse der Softwaregestaltung. Das geht bei dem EVU so weit, dass es gemeinsame Projekte mit anderen EVU initiiert, wofür es die entsprechenden, typischen koordinierenden Rollen einer zentralisierten Softwaregestaltung hat. So muss in dem Fall von prekär-kooperativen kommunikativen Beziehungen gesprochen werden, und zwar auch für die dezentralen Arbeitsprozesse innerhalb der EVU: weil sie, nachdem das IT-DL die Koordination der Softwaregestaltung nicht mehr übernimmt, erst intern lernen müssen, abteilungsübergreifend für Softwareprojekte Beziehungen aufzubauen und zu kommunizieren. Weil es keine eingespielte, langfristige zentralisierte Abstimmung zwischen IT-DL

und EVU gibt, gibt es auch keine klare Systematik des softwaretechnischen Zuschnitts und damit, welches EVU welchem Standard folgt und was EVU individuell gestalten.

Rollen: verteilte Rollen für zentralisierte und dezentrale Softwaregestaltung (IT-koordinierender Fachbereich, Key User:innen, IT-Projektleitende & Co.)

Da es in diesem Fall sowohl zentrale als auch dezentrale Arbeitsprozesse der Softwaregestaltung gibt, existieren auch die typischen Rollen für beide. Aber anders als KOOP1 zeigt der Fall noch einmal besonders, was passiert, wenn EVU anfangen, dezentral Software zu gestalten, und sie dafür intern die entsprechenden Rollen brauchen. Die EVU brauchen dann noch mehr koordinierende Rollen als bei KOOP1, weil sie neben der Anforderungsaufnahme auch noch die Umsetzung selbst koordinieren (z.B. externe Freelancer oder das IT-DL beauftragen).

EVU2 koordiniert die Softwaregestaltung wieder verstärkt selbst. Es verändert seine internen Strukturen nur dahingehend, dass sie[es?] diese um einzelne Rollen für die Softwaregestaltung ergänzt. Diese agieren dann fachbereichsübergreifend wie auch beschränkt auf einzelne Fachbereiche und arbeiten teilweise eigenständig mit externen Softwareunternehmen zusammen. Zusätzlich gibt es extra Rollen wie eine Projekt-Coaching, um die Bedingungen für die Softwaregestaltung im EVU zu verbessern.

Zuerst zu den fachbereichsübergreifenden Arbeitsprozessen der Softwaregestaltung: Hier zeigt sich an den Rollen, dass das EVU sich auf Softwaregestaltung einstellt, dafür die althergebrachten Strukturen aber nicht radikal ändert. Das hat zur Folge, dass nicht die direkte Anforderungsaufnahme im Mittelpunkt steht, sondern die Koordination der Softwaregestaltung quer zu den althergebrachten Strukturen aus Hierarchien und Abteilungsgrenzen. So kann zwar der befragte IT-Projektmanager des EVU2 dank seiner Position in der Stabstelle Netzbereich, die hierarchisch über den Abteilungen steht, ein zentralisiertes, fachbereichsübergreifendes Softwaregestaltungsprojekt steuern. Er kann als Bindeglied zwischen kaufmännischen und technischen Fachbereichen fungieren. Dabei übernimmt er aber vor allem eine koordinierende Rolle: innerhalb des EVU und zu Externen wie IT-DL oder anderen EVU. Das Projekt, das er leitet, findet zusammen mit anderen EVU statt und es betrifft nicht nur unterschiedliche Fachbereiche, sondern auch unterschiedliche Systeme.

»Netzbetrieb bedeutet das, das, das und wir brauchen dafür die und die Systeme und die hängen so und so zusammen. Und danach kommt dann Abrechnung, Bilanzierung, was auch immer zusammenhängt. Was ich brauche... Ja, fachbereichsübergreifende Kompetenz über Prozesse, wie hängt etwas zusammen, und ich kann dann erst die Systeme auch verorten und damit die Schnittstellen.« (IT-Projektleiter EVU2)

Unabhängig von einzelnen Projekten hat EVU2 die Aufnahme von Anforderungen und die Koordination der Softwaregestaltung in den einzelnen Fachbereichen langfristig bürokratisch gelöst, indem es die bestehenden hierarchischen Strukturen ergänzt hat. So hat es die Rollen der IT-Koordinierenden sowohl auf IT- als auch auf Fachbereichsseite für die Softwaregestaltung für jeden Fachbereich geschaffen. Die technischen IT-Koordinierenden aus der IT-Abteilung kümmern sich z.B. um die Abrechnung mit IT-DL und kontrollieren das Budget. Die fachlichen IT-Koordinierenden aus den Fachberei-

chen sind u.a. dafür zuständig, innerhalb des Fachbereichs für einen durchgängigen digitalen Prozess zu sorgen und Anforderungen aufzunehmen. Sie informieren die technischen IT-Koordinierenden u.a. über Anforderungen oder Aufträge an einen IT-DL.

Somit hat das EVU die althergebrachte Organisation auf die Softwaregestaltung eingestellt, indem es IT-Projekte durchführt und zentrale und dezentrale Koordinationsrollen in den Fachbereichen und der IT-Abteilung etabliert. Weil das EVU bemerkt hat, dass diese Strukturen noch nicht reichen, hat es zusätzlich eine Projekt-Coachin eingestellt, die die Projektarbeit durch verschiedene Maßnahmen verbessern soll. Sie führt Schulungen zur besseren Kommunikation unter Projektbeteiligten und in Workshops durch.

Im gleichen EVU2 gibt es aber auch ein Beispiel für einen dezentralen Arbeitsprozess der Softwaregestaltung unabhängig von den IT-Koordinierenden, der sich auf die Anforderungsaufnahme und -umsetzung konzentriert und weniger koordinative Aufwände betreiben muss. In dem Fachbereich für Marktkommunikation kümmert sich die Teamleiterin um die Weiterentwicklung der Marktkommunikationssoftware, die zwar an das SAP-ERP-System angeschlossen, aber nicht von SAP ist. Dafür arbeitet sie direkt mit einem Prozessmanager der IT-Abteilung, einem IT-Berater der Softwarefirma der Marktkommunikationslösung und den Key User:innen ihres Teams zusammen. Die Key User:innen ihres Teams sammeln die Anforderungen von den anderen Anwendenden ein und sind in mehrfacher Hinsicht Übersetzende:

»Im Prinzip sind meine Leute nicht nur Key User, sie sind Prozess Manager, sie sind Problemlöser, sie sind die Schnittstelle zwischen Sachbearbeiter und der Entwicklung. Also, die müssen es übersetzen, dass es der Entwickler versteht und umgekehrt auch der Sachbearbeiter.« (Teamleiterin Mako EVU2)

Solche Rollen für eine dezentrale, von der internen Hierarchie unabhängige Softwaregestaltung gibt es auch in anderen EVU. Im EVU1 hat der befragte Sachbearbeiter, obwohl er kein Key User ist, zusätzlich die Rolle, regelmäßig bei Treffen zu Anforderungen für eine Software zur Abrechnung von Anlagen erneuerbarer Energien dabei zu sein.

Dezentrale Softwaregestaltung zu ermöglichen und zuzulassen, ist auch ein Thema bei EVU3. Es hat vor zweieinhalb Jahren entschieden, die Kompetenzen zum Thema Digitalisierung zu bündeln. Vorher haben die Fachbereiche selbstständig bei der Anpassung von SAP agiert. Jetzt wird zentral koordiniert, um eine Gesamtsicht über die Digitalisierung zu haben (auch zu Themen wie IT-Sicherheit, Datenschutz, Softwarearchitektur). Die Rolle des befragten Manager Digitalisierung ist Teil dieser Strategie. Er kümmert sich darum, eine zentrale, digitale Infrastruktur für Softwareentwicklungen zu schaffen. Diese soll es u.a. erleichtern, externe Programmierende einzubinden. Er initiiert Softwareentwicklungsprojekte, die dann auf dieser zentralen Infrastruktur stattfinden.

Weil die EVU intern mehr Know-how bei der Softwaregestaltung aufbauen und koordinieren, haben auch typische Rollen wie IT-Beratende beim IT-DL andere Aufgaben: Anders als in anderen Fällen programmiert der befragte IT-Berater des IT-DL hauptsächlich, hat kaum eine vermittelnde oder fachlich beratende Funktion und wird flexibel eingesetzt. Die EVU kaufen nur seine Leistung als Programmierer. Anders als die EVU setzt das IT-DL zwar Scrum ein. Wie in den anderen Fallstudien lebt es die Rollen aber

anders. Der Scrum Master ist gleichzeitig Projektleiter, Product Owner und macht bei Softwaretests mit.

»Ich sage mal, die Rollen sind ein bisschen anders, nicht ganz so scharf geschnitten. Der Scrum Master ist eigentlich auch Projektleiter oder Product Owner. Das verschwimmt dann häufig ein bisschen.« (IT-Berater)

Ablauf: dezentral und zentralisiert in den EVU

Da die EVU zum Teil selbst Software entwickeln – mit mehreren Fachabteilungen und in Zusammenarbeit mit anderen EVU –, finden sich in diesen EVU typische Abläufe für zentrale Arbeitsprozesse, um Anforderungen zu verhandeln und Konflikte zu lösen. Gleichzeitig zeigen die Beispiele dezentraler Arbeitsprozesse der Softwaregestaltung in den EVU den typischen Fokus auf das Ausarbeiten und Sammeln von Anforderungen, mit den dazugehörigen dezentralen Gestaltungsnetzwerken und Feedbackschleifen zwischen Anwendung und Programmierung. Das IT-DL ist dann nur noch in die von den EVU koordinierte Softwaregestaltung eingebunden, weil es Programmierende für sie abstellt oder einzelne Anforderungen umsetzt.

Wie die Rollen bereits gezeigt haben, ist EVU2 ein Beispiel dafür, wie ein EVU mehr Verantwortung bei der Softwaregestaltung übernimmt, ohne bestehende Strukturen zu verändern, sondern diese vielmehr ergänzt. Bei einem von EVU2 koordinierten großen IT-Projekt¹⁴ schlagen sich die internen Strukturen in der Koordination des Ablaufs nieder, der auf Gremienarbeit basiert, wie es für eine zentralisierte Softwaregestaltung typisch ist: Es gibt einen dreiköpfigen Lenkungskreis und fünf Teil-Projektleitende nach Bereichen aufgeteilt, um eine bereichsübergreifende Kooperation horizontal hinzubekommen (u.a. die Bereiche Netzbau, Netz-Service, Netz-Management und kaufmännischer Service). An täglichen Jour fixes des Projekts, in denen Aktivitätenlisten durchgegangen werden und was noch an Arbeitspaketen offen ist, nehmen immer auch die verschiedenen Teilprojekte teil. Wie in Scrum wird iterativ vorgegangen. Doch reduziert sich das regelmäßige Feedback der Anwendenden auf Tests, mit denen sie entscheiden, ob sie einer Entwicklung grünes Licht geben. Wie für eine zentralisierte Softwaregestaltung typisch, waren es vor allem Fachleute und Führungskräfte, die an den Workshops für die Anforderungsaufnahme teilgenommen und sich mit den Programmierenden ausgetauscht haben.

Ein solcher zentraler, top-down-orientierter Softwaregestaltungsprozess röhrt aber nicht allein aus der Organisation des EVU her. Das hängt davon ab, ob es sich um einen zentralisierten – im Fall von EVU2 fachbereichsübergreifenden – Softwaregestaltungsprozess in einer Hierarchie und mit mehreren Abteilungen handelt oder einen dezentralen. Bei einem anderen Projekt von EVU2, bei dem es um die Betreuung der Kundenschaft im Netzbereich geht, laufen die Anpassungen dezentral über die betroffenen Teams. Treffen finden direkt zwischen Anwendenden und Programmierenden stattfinden. Die Hierarchie schlägt sich dann aber in einem bürokratischen Ablauf bei

14 Es ging um die Umsetzung von Redespatch 2.0, einer Regulierung für die Verteilnetzbetriebe (VNB).

der Umsetzung nieder: Der IT-Koordinator des Fachbereichs prüft und bewertet Anforderungen, die ihm die Anwendenden geben. Er übergibt die Anforderungen an den für den Fachbereich zuständigen technischen IT-Koordinator aus der IT-Abteilung, der dann die Umsetzung organisiert und dazu auf das IT-DL oder andere Programmierende bspw. Freelancer zugreift.

Stärker noch dezentral und auf Sammeln und Ausarbeiten von Anforderungen konzentriert ist die Softwaregestaltung im Fachbereich für Marktkommunikation von EVU2. Der Fachbereich nimmt individuelle Anpassungen an der Software direkt mit dem Softwareunternehmen vor. Die Anforderungen sammelt die Teamleitung mithilfe von Key User:innen bei den Anwendenden ein und gibt sie in Form von Tickets an die Softwarefirma weiter. Alle zwei Wochen hat die Teamleitung einen Termin mit einem Berater der Softwarefirma, in dem offene Punkte und Probleme besprochen werden. Doch auch hier muss die Teamleitung existierende Hierarchien berücksichtigen. Weil Änderungen an der Schnittstelle zur ERP-Software vorzunehmen sind, muss die IT-Abteilung die entsprechenden Programmierenden beauftragen. Die Teamleitung macht regelmäßig Termine mit den jeweiligen für das SAP-Softwarepaket verantwortlichen Prozessmanagenden, um Softwareänderungen abzustimmen.

Eine andere Art, wie ein EVU selbst Abläufe schafft, um individuell Software zu gestalten, zeigt EVU3. Ein neu geschaffener, zentraler Bereich für IT-Projekte treibt für mehrere Fachbereiche die Softwaregestaltung voran. So hat das EVU über kleinere IT-Projekte Erfahrungen in der App-Entwicklung gesammelt und dezentral in den Fachbereichen ausprobiert.

»Das war so die erste Applikation, die in eine Cloud migriert ist von On-Prem¹⁵, und das war so ein bisschen so ein Versuchsballon für alles Mögliche. Und da haben wir sehr viele Verfahrensweisen mittlerweile auch für andere Applikationen übernommen. Aus diesen Teams raus haben sich dann noch andere Teams gebildet, die dann zum Beispiel für den Kundenservice eine App entwickelt haben, wo man dann auch irgendwo so ein bisschen Selbstvertrauen gewonnen hat und man gesagt hat: Was wir hier gemacht haben, können wir dann eigentlich für einen anderen Bereich auch machen. [...] Ob wir uns jetzt eine teure Lösung einkaufen oder hier was Kleines, Schnelles selber entwickeln. Lass uns mal die eigene Entwicklung probieren. Und das haben wir an vielen Ecken und Enden mittlerweile gemacht. Und funktioniert ganz gut. Funktioniert nicht alles, muss man dazu sagen. Aber viele dieser Projekte haben wirklich Mehrwert gebracht.« (Manager Digitalisierung)

Das IT-DL hat in dieser Fallstudie deutlich weniger koordinativen Aufwand als bei KOOP1. Es nimmt vor allem Anforderungen entgegen oder stellt Programmierende für IT-Projekte zur Verfügung und leitet nur noch wenige Projekte selbst. Die Umsetzung innerhalb des IT-DL ist zum Teil nach einem angepassten Scrum organisiert. Sonst herrschen laut einem Befragten im Alltag des IT-DL verschiedene Mischformen vor – aus Scrum, Wasserfall oder anderen Methoden.

¹⁵ Kurzform für On-Premises und vor der Zeit von Cloud-Computing das gängige Modell, um Software zu betreiben: auf einem lokalen Server mit der entsprechenden Lizenz.

Kommunikative Beziehungen: prekäre Beziehungen, direkte Kommunikation und hinderliche Hierarchien

Der Fall zeigt, dass mangelhafte kommunikative Beziehungen dazu führen können, dass kein zentralisierter Arbeitsprozess der Standardsoftwaregestaltung für mehrere EVU in einem IT-DL möglich ist. Die Marktbeziehung versucht das IT-DL nun durch Beziehungspflege zu verbessern, profitiert aber bereits vom direkten Kontakt zwischen Mitarbeitenden des IT-DL und den EVU. Jene EVU, die nun die Koordination der Softwaregestaltung übernehmen, merken, dass es eine Herausforderung darstellt, kommunikative Beziehungen innerhalb ihrer Matrixorganisation herzustellen.

Das IT-DL ist noch dabei, die Beziehungspflege zu den EVU zu verbessern. Ein befragter Teamleiter des IT-DL betont die Wichtigkeit der Beziehungspflege zu den EVU. Es sei Teil seiner Arbeit. Er musste seine Führungskraft aber erst überzeugen, dass durch eine gute Beziehung die EVU z.B. eher Fehler verzeihen würden. Dazu gehört es, öfters mal anzurufen, als direkter Ansprechpartner da zu sein und auch ehrlich und offen zu sagen, was man nicht kann, um dadurch Vertrauen zu schaffen.

Doch obwohl die kooperative Softwaregestaltung der EVU mit dem IT-DL als koordinierende Organisation nicht so gut funktioniert, ist ein direkter Austausch der Beschäftigten zwischen IT-DL und EVU möglich. Für das IT-DL ist es von Vorteil, wenn Ansprechpersonen und die internen Wege in den EVU bekannt sind. Weil das IT-DL durch Outsourcing aus den EVU entstanden ist, haben einzelne Mitarbeitende in den EVU noch alte Kontakte. Der befragte IT-Koordinator eines Fachbereichs von EVU2 spricht trotz eigentlich vorhandener und vorgegebener formaler Wege über die IT-Abteilung direkt mit dem IT-DL. Er betont die Vorteile direkter Kommunikation:

»Nichtsdestotrotz halte ich es für klug, wenn der Anforderer direkt mit dem Entwickler spricht, wenn es eben um das Customizing geht oder halt das Thema an sich, dass das wirklich auch der Entwickler im O-Ton hört, was will der Anforderer eigentlich haben. [...] Bevor ich das dann hier auf drei DINA4 Seiten runter beschrieben habe, das macht dann keinen Sinn, dann spricht man lieber.« (IT-Koordinator EVU2)

Da nun EVU2 selbst anfängt, Softwaregestaltung für fachbereichs- und organisationsübergreifende Projekte zu organisieren, muss es erst noch lernen, mit den Spannungen aufgrund von Hierarchien und Abteilungsgrenzen umzugehen, und eine gemeinsame Sprache finden. Der Projektleiter aus EVU2 beschreibt ausführlich, wie er initial zwischen den verschiedenen Bereichen horizontal interdisziplinäre Kontakte geknüpft hat. Er hat Termine gemacht, damit die verschiedenen im Projekt involvierten Fachbereiche die Perspektive der anderen kennengelernt haben. Das folgende Zitat zeigt, dass bestehende Strukturen erhalten und informelle Kontakte aus dem Projekt heraus bestehen bleiben, eine gemeinsame Wissensbasis entsteht und ein Austausch unabhängig von der Projektleitung stattfinden muss:

»Ich persönlich bin hergegangen und habe die Kolleginnen und Kollegen quasi zusammengenommen und habe versucht, jeweils die Sichtweisen der anderen darzustellen. Das heißt, die kaufmännischen Kollegen haben was mitbekommen von den technischen Kollegen, was deren ihre Hauptaufgabe ist – auch andersherum. Und,

dass jeder ein Gefühl bekommt für die Tätigkeit der anderen, dass man einen Blick hat, dass man versteht, was wird benötigt vom anderen und was kann der überhaupt liefern. Was ist der in der Lage...? Um erst mal ein Gefühl für den Gesamtprozess zu bekommen. Und entsprechende Anpassungen wurden dann auch gemeinsam diskutiert. Das hat anfangs ein bisschen länger gedauert, weil natürlich die Sichtweise nicht da war. Aber gerade jetzt, am Ende des Projektes, zeigt sich, dass das wirklich auch nicht die schlechteste Entscheidung war, Ressourcen dafür zu verwenden, weil dann auch im wirklichen Tagesgeschäft das Know-how vorhanden ist. Da brauchen wir es. Und die Kontakte sind geknüpft. Das heißt, nicht als Projektleiter die zentrale Funktion zu sein und zentrale Kommunikation, sondern die Kolleginnen und Kollegen sollten direkt miteinander auch kommunizieren, sich austauschen, ein Gefühl für die Arbeit der anderen bekommen.« (IT-Projektleiter EVU2)

Laut dem befragten IT-Projektleiter gibt es Kommunikationshürden durch das Hierarchiedenken z.B., wenn er die IT-Koordinatoren einbeziehen muss, diese aber keine fachbereichsübergreifenden Entscheidungen fällen können. Zudem würde der IT-Abteilung der Einblick in die tägliche Arbeit der Fachbereiche fehlen. Sie habe keine Vorstellung über die Probleme der Fachbereiche und aus seiner Sicht arbeitet sie nicht lösungsorientiert.

Auch aus Sicht des IT-Koordinators aus dem Fachbereich sind Hierarchien das Problem. Die operative Zusammenarbeit unter Mitarbeitenden unterschiedlicher Bereiche funktioniert dagegen gut:

»Also, Ingenieure untereinander, die sind sich einig und zwar, da ist es völlig wurscht, ob die aus drei unterschiedlichen Bereichen kommen. Und es ist eben auch die Erfahrung, die wir bei uns gemacht haben und auch immer noch machen: Auf der operativen Basis funktioniert alles wunderbar, ja. Die Probleme fangen in den Führungsebenen an, weil da andere Befindlichkeiten ins Spiel kommen, die wir meistens auch gar nicht kennen.« (IT-Koordinator Fachbereich EVU2)

Bei den Fachbereichen setzt, wenn es um IT-Budget oder um zu erreichende Ziele geht, bei den Führungskräften ein Silodenken ein: Wer muss bezahlen? Wer bekommt die Anerkennung? Dies ist ein weiteres Beispiel dafür, wie innerhalb der EVU das Hierarchiedenken die kommunikativen Beziehungen behindert und bei Softwaregestaltung innerhalb von EVU der direkte Austausch nicht unbedingt einfacher ist.

Digitale Werkzeuge: Ticketsysteme, MS Excel und E-Mails

In dem Fall zeigt sich die typische Verwendung von digitalen Werkzeugen für die Softwaregestaltung. Allerdings ist sie weniger eindeutig auf die Typen von zentralisierten oder dezentralen Arbeitsprozessen zurückzuführen. Auch hier liegt es daran, dass, anders als bei KOOP1, die Softwaregestaltung wieder mehr die EVU übernehmen, auch was deren Koordination anbelangt.

Zwar hat das IT-DL ein Ticketsystem. Dies nutzen die EVU aber weniger, um die von ihnen koordinierte Softwaregestaltung zu organisieren, als vielmehr dem IT-DL Anforderungen zur Umsetzung zu übergeben. Wie bei KOOP1 können die EVU mithilfe dieses

Ticketsystems die Arbeit des IT-DL kontrollieren. Dazu gehört die Kontrolle vereinbarter SLA zwischen IT-DL und EVU oder die Tickets mit einer Priorität zu versehen.

Zusätzlich dazu verwendet EVU2 ein Ticketsystem der ERP-Software von SAP für die Umsetzung von Anforderungen. Für die Erarbeitung und den Austausch über Anforderungen setzt es MS Excel und E-Mails ein. Weil es die Koordination der Umsetzung der Softwaregestaltung selbst übernimmt, nutzt das EVU zudem noch die Ticketsysteme anderer Softwarefirmen und IT-DL, mit denen es zusammenarbeitet. So nutzt es die digitalen Werkzeuge sowohl für direkten, dezentralen Input als auch für Transparenz bei zentralisierter Softwaregestaltung. Es gibt nicht das eine Werkzeug, um die Kommunikation zentral zu kontrollieren.

Softwaretechnischer Zuschnitt: Synergien über gemeinsame Projekte und Release des IT-DL
 So wie auch der Arbeitsprozess der Softwaregestaltung nicht klar zentralisiert ist, zeigt sich in dem Fall auch nicht der entsprechend typische softwaretechnische Zuschnitt eines Standards. Es zeigt sich ein Flickenteppich aus Standard von SAP, durch das IT-DL gestaltetem Standard, durch EVU gestaltetem Standard und individuellen Anpassungen und Erweiterungen. So reizen die EVU in diesem Fall die Möglichkeiten eines gemeinsamen IT-DL nicht aus, zusammen Synergien zu heben.

Der Release an Softwareänderungen des IT-DL für einen gemeinsamen Standard konzentriert sich vor allem auf einzelne Regulierungen, die sämtliche EVU umsetzen müssen. Doch auch dieser gemeinsame Release bzw. der durch das IT-DL für mehrere EVU entwickelte Softwareteil war für einige Jahre ausgesetzt. Ein anderer Weg zu Synergien sind gemeinsame Projekte, bei denen die EVU unabhängig vom IT-DL gemeinsam einen Standard gestalten, wie dies in dem oben geschilderten Projekt von EVU2 passiert ist.

Anders als bei KOOP1 gibt es keine abgestimmte Systematik zwischen den EVU, was sie als gemeinsamen Standard und was sie individuell gestalten. Für die Untersuchung wäre es daher ein nicht abzuschätzender Aufwand gewesen, genau herauszufinden, welches EVU was individuell gestaltet und wo es einem Standard folgt.

8.3.2.5. PAKET: zentrale Softwaregestaltung durch eine Softwarefirma

In dem Fall gestaltet eine Softwarefirma eine industriespezifische Standard-ERP-Software, bei der die EVU nur individuelle Einstellungen vornehmen können. Entsprechend existiert ein zentralisierter Arbeitsprozess der Softwaregestaltung der Softwarefirma mit den typischen Rollen und Abläufen zur Koordination, die Verhandlungen zwischen Softwarefirma und EVU über den Standard zulassen. Doch da innerhalb der Softwarefirma interdisziplinäre Teams existieren, ist sie nicht ausschließlich auf den Input der EVU angewiesen. Zudem ist allgemeines Branchenwissen wichtiger für die Gestaltung der Branchen-Standardsoftware als Wissen über spezifische Arbeitsabläufe in einem EVU. Deswegen spielen die Anwendenden bei der Softwaregestaltung nahezu keine Rolle. Jedoch: Da die EVU dezentral Einstellungen an der Software vornehmen können, gibt es intern in den EVU noch eine dezentrale Softwaregestaltung, die sich darum kümmert – mit den entsprechenden Rollen und Abläufen. In dem Fall spielen die organisations- und abteilungsübergreifenden kommunikativen Beziehungen eine geringere Rolle als

in den anderen Fällen. Dies ist zum einen so, weil die Softwarefirma (SF) spezialisiert auf die Softwareentwicklung ist und dort interdisziplinäre Teams existieren, die sich direkt austauschen können. Zum anderen ist zwar der Austausch zwischen EVU und SF wegen der Marktbeziehung stark formalisiert und die Spannungen aufgrund der unterschiedlichen Interessen von EVU und SF fallen auf. Allerdings ist die Kommunikation zwischen EVU und SF weniger wichtig für die Softwaregestaltung der ERP-Software, weil diese hauptsächlich von der Softwarefirma ausgeht. Nur mit einzelnen, ausgewählten EVU gibt es eine engere, kooperative Zusammenarbeit für die Softwaregestaltung. In dem Fall dient das Ticketsystem der Softwarefirma entsprechend vor allem dazu, die Kommunikation und die Umsetzung von Anforderungen und Fehlern (die den EVU in der Anwendung auffallen) zu kontrollieren. Dadurch, dass die Softwarefirma für viele EVU einen Standard entwickelt, schöpft sie die Möglichkeiten an Synergien in der Softwaregestaltung stark aus. Gleichzeitig gibt es durch die Einstellungsmöglichkeiten an der Software begrenzten Spielraum für individuelle Softwaregestaltung durch die EVU.

Rollen: verteilte Fachexpert:innen zwischen allgemeinem Branchen- und firmenspezifischen Anwendungswissen

Wie für einen zentralen Arbeitsprozess der Softwaregestaltung typisch, bestehen an die Rollen vor allem die Erwartungen, die Softwaregestaltung zu koordinieren und den Standard zu gestalten. Zusätzlich existieren innerhalb der EVU noch Rollen, um individuelle Einstellungen an der ERP-Software vorzunehmen.

Die Softwarefirma koordiniert die Softwaregestaltung: ob in ihren interdisziplinären Teams, durch Projekte mit EVU oder durch Arbeitskreise, zu denen Fachleute bzw. fachliche Repräsentant:innen aus den EVU eingeladen sind. Eine Rolle ist der Lösungsarchitekt. Er koordiniert Erweiterungen der Standardsoftware. Er prüft, was in der Branche an neuen Anforderungen entsteht, und entscheidet dann, was die Softwarefirma selbst umsetzt und was sie anderen Softwarefirmen überlässt. Zum Beispiel deckt die Softwarefirma beim Thema Abrechnung von Strom aus E-Auto-Ladesäulen mit ihrer Software nicht das Auslesen der Strommenge aus der Ladesäule ab, weil es dafür Angebote anderer Unternehmen gibt. Daneben hat die Softwarefirma IT-Beratende. Sie nehmen Anforderungen von den EVU auf und Einstellungen an der Standardsoftware vor und unterstützen darüber hinaus die EVU bei der Anwendung des Standardprodukts.

»Also, der reguläre Weg ist, wir haben den Consultant, der nimmt dann sozusagen unsere Anforderungen auf und dann geht das [Softwarefirma]-intern weiter.« (Teamleiterin Abrechnung EVU5)

Außerhalb der Softwarefirma bringen fachliche Repräsentant:innen der Branche sich in Arbeitskreisen bei der Gestaltung des Standards ein. So sind von den vielen EVU, welche die Software nutzen, nur wenige an der Gestaltung beteiligt.

In den EVU gibt es mal mehr und mal weniger interne Beschäftigte, die sich um einen optimalen Einsatz der Software kümmern können, dafür Einstellungen an dieser vornehmen und einen hohen Grad an Automatisierung garantieren. Wenn EVU solche internen Rollen haben, handelt es sich meist um Key User:innen oder Anwendungsbetreu-

ende. Key User:innen nehmen Anforderungen auf, nehmen Einstellungen an der Software vor, geben Tickets an die Softwarefirma auf oder informieren die Anwendenden über Änderungen an der Software.

»Ja, also, die [Anwender] bringen sich ein. Das sind ja auch die, die immer am lebenden Objekt... Die wissen ja noch mehr, was sie brauchen, damit sie diesen Prozess bearbeiten können. [...] Also, der Key User sammelt das alles ein.« (Teamleiter Abrechnung EVU5)

Anwendungsbetreuende haben meist noch eine übergreifende Perspektive auf das System und nehmen Prozessoptimierungen vor.

»Weil es natürlich auch oft so ist, dass man schauen muss, wie man eben seine eigenen Prozesse in die neue Software implementiert und dann erst mal schauen muss, wie funktioniert es eigentlich? Was kommt da hinten raus?« (Anwendungsbetreuung EVU3)

In dem Fall zeigt sich die für Rollen in der Softwaregestaltung typische situative Ausgestaltung, wie die, dass es z.B. mal nur Key User:innen gibt und mal auch noch Anwendungsbetreuende. Zudem gibt es mehr Führungskräfte als in den anderen Fallstudien, welche die Software gestalten. Es gibt Team-/Gruppenleitende, die durch ihre langjährige Erfahrung mit der Software Systemwissen aufgebaut haben und wegen ihres fachlichen Know-hows eine vermittelnde Rolle zwischen Software und Anwendung spielen. Mal arbeiten sie am Schreiben von Anforderungen mit (Teamleiter EVU5), mal nehmen sie in geringem Umfang selbst Einstellungen an der Software vor (Gruppenleiter EVU4).

Ablauf: innerhalb und zwischen Organisationen – etablierte Kommunikationswege in der Softwarefirma und Einbindung von Branchenexpert:innen

Das Sammeln und Ausarbeiten von Anforderungen erfolgt in diesem Fall maßgeblich durch die Softwarefirma. Das heißt, zentral schreibt die Softwarefirma Konzepte und holt sich die Meinungen der EVU über verschiedene Wege ein, z.B. in gemeinsamen Treffen wie Arbeitskreisen, um über den Standard zu verhandeln. Nur in Ausnahmefällen schreiben die EVU selbst die Konzepte. Weil die Softwarefirma die Hoheit hat, was in den Standard kommt, muss der Ablauf selten tiefergehende inhaltliche Konflikte zwischen den EVU lösen. Weil es um einen Branchenstandard geht, sind die Gestaltungsnetzwerke vor allem auf Branchenexpert:innen begrenzt und schließen weniger individuelle Anwendende ein. Die Feedbacks zwischen EVU und SF reduzieren sich dann meistens auf Fehler und Tests und finden, was die Softwaregestaltung betrifft, mehr innerhalb der SF statt. Dafür setzt sie verschiedene Methoden zentral in der Softwarefirma (agile wie Scrum, Wasserfallmodell, Teamarbeit, Projekte) und zentralisiert für die Softwaregestaltung (Projekte, Arbeitskreise, Anwendendengruppen, Entwicklungskooperationen) ein.

In den (Fach-)Arbeitskreisen treffen sich mehrere EVU und einige Vertreter:innen der SF mindestens zweimal im Jahr, um über Details einzelner, dringender Themen zu sprechen. 10–15 Personen im Schnitt sprechen über aktuelle Umsetzungen und Anfor-

derungen. Die Mitglieder sind von der SF ausgesuchte Mitarbeitende aus EVU, die für ihre Expertise bekannt sind. Die Softwarefirma stellt erste Entwurfsfassungen von Konzepten vor und die EVU können eigene Vorschläge einbringen. Es werden Prozesse und Eingabemasken besprochen und ggf. gezeigt. Teilweise erstellen Mitarbeitende der EVU einzelne Kapitel der Konzepte.

»Es gibt sogenannte Facharbeitskreise und da wirken wir immer mit und es gibt quasi gewisse... Es gibt dort die Fachabteilung Abrechnung. Es gibt den technischen Netzbetrieb, es gibt Marktkommunikation. Und dort sind jeweils Kollegen von uns, die dann natürlich aufmerksam auch die Gesetze lesen und die Veränderungen und geben dann der [Softwarefirma] Hinweise. Oder: Wir hätten das gerne so und so und könnt ihr das nicht so bauen?« (Gruppenleiter EVU3)

Die Herausforderung bei zentralisierter Softwaregestaltung, nicht nah an Anwendenden dran zu sein, löst PAKET durch Prototyping, z.B. indem es laut Befragten ca. zwei von drei Anforderungen agil umsetzt, d.h. erst ein Grundgerüst programmiert und dann auf Basis von Feedback durch die EVU kontinuierlich weiterentwickelt. In anderen Fällen stellt die Softwarefirma umgesetzte Features in einer Web-Konferenz ausgewählten Anwendenden vor, um Feedback zu erhalten.

Dabei kann der interdisziplinäre Austausch innerhalb der Softwarefirma sehr intensiv und kontinuierlich erfolgen, was ein Vorteil der Zentralisierung ist:

»Es wird meistens ein zumindest grobes fachliches Konzept vorgegeben und alles weitere, die konkrete fachliche Ausprägung und so, das wird dann mit den entsprechenden Spezialisten für die Prozesse dann geklärt und dokumentiert.« (Programmierer)

In den Teams der Softwarefirma, die jeweils für einen Teil der ERP-Software zuständig sind, gibt es tägliche Besprechungen wegen neuer Anforderungen vor allem wegen neuer Regulierung (typischerweise ohne Anwendende oder EVU). Im Team des befragten Programmierers gibt es drei, vier Jour fixes die Woche und dann zwei, drei abhängig von Ereignissen, Problemen etc., über die man reden muss.

Die Ausnahme stellt EVU5 dar, das erst vor kurzem die ERP-Software eingeführt hat. Weil das EVU einige Eigenheiten hat, die bisher die Standardsoftware nicht berücksichtigt, gibt es eine engere Zusammenarbeit – eine Art Entwicklungskooperation. Es gibt regelmäßigen Austausch zwischen dem EVU und der Softwarefirma. Die befragte Gruppenleitung aus EVU5 hat zusammen mit Key User:innen einmal in der Woche einen Termin mit der Softwarefirma und deren Programmierenden.

Neben der Teilnahme an der Konzeptionierung für den Branchenstandard leisten die EVU eine wichtige Qualitätskontrolle/-verbesserung, indem sie Tests durchführen und Fehlermeldungen zu neuen Versionen der Software aufnehmen. Wie auch bei der Konzeptionierung machen die EVU dabei in unterschiedlichem Ausmaß mit (tendenziell eher die größeren EVU).

Kommunikative Beziehungen: zwischen reiner Zulieferbeziehung, Formalisierung und partnerschaftlicher, direkter Kommunikation

Die für einen zentralen Arbeitsprozess der Softwaregestaltungen typischen spannungs-geladenen und bürokratischen Beziehungen existieren in dem Fall. Wobei einige der EVU gut mit der reinen Lieferbeziehung leben können. Andere wünschen sich einen partnerschaftlichen Umgang, um das komplizierte Produkt besser einsetzen zu können. So oder so geht es in diesem Fall zwischen EVU und Softwarefirma weniger um die Gestaltung der Software als um die Qualität eines Standardprodukts und welche Aufgaben die Softwarefirma als Zulieferunternehmen darüber hinaus hat. Nur wenige EVU arbeiten bei der Gestaltung der Software enger mit der Softwarefirma zusammen.

Wie für einen zentralisierten Arbeitsprozess der Softwaregestaltung typisch, wollen die EVU in erster Linie eine funktionierende Standardsoftware. Deshalb existieren zwischen Softwarefirma und EVU nicht immer kommunikative Beziehungen. Ein Befragter spricht von »reine[r] Dienstleistungsbeziehung« (Teamleiter EVU2). Für einen anderen Befragten ist die Beziehung erst einmal nicht wichtig, weil die Software einfach funktionieren soll. Er erwartete, dass die Softwarefirma Fehler oder Anforderungen termingerecht behebt bzw. umsetzt. Teilweise kommt es zu Eskalationen aufgrund von zu langsaamer Umsetzung. Der Teamleiter aus EVU2 ist misstrauisch und hat den Verdacht, dass größere EVU schneller Rückmeldung bekommen. Er ist davon genervt, dass er immer nachfragen muss, wenn es länger dauert.

Doch ist die industriespezifische ERP-Software ein zu kompliziertes Produkt, als dass immer eine bloße Auslieferung ausreicht. Einige EVU haben deshalb die Erwartung, dass die Beziehung darüber hinausgeht. Für einige befragte EVU hat die Softwarefirma, obwohl nicht vertraglich fixiert oder Teil des Produkts, eine Bringschuld. Sie kommt nicht auf die EVU zu, um auf Verbesserungsmöglichkeiten, Einstellungs-möglichkeiten oder (neue) Funktionalitäten hinzuweisen. Ein Befragter merkt manchmal nur durch Austausch mit anderen EVU, was noch mit der Software möglich wäre. Ein EVU hat erst durch Umstieg auf ein neues Lizenzmodell gemerkt, was es noch alles für Module gibt. Aus der Sicht eines anderen Befragten wäre es partnerschaftlich, wenn mehr Austausch als dringend notwendig stattfinden würde. In Einzelfällen gibt es das und dann ist es ein Geben und Nehmen: Laut Befragten existieren bei Tickets, die schwer zu beschreiben sind und deshalb ausführlicher ausfallen, engagierte Mitarbeitende der Softwarefirma. Diese rufen an und geben Tipps und notwendige Hintergrundinfor-mationen, was das Problem sein könnte und wie es zu lösen ist.

Das Gegenteil zur reinen Zulieferbeziehung sind die engeren Beziehungen der Softwarefirma mit EVU, mit denen sie gemeinsam Innovationsprojekte macht oder langfristig bei der Gestaltung des Standards zusammenarbeitet. Mit diesen EVU ist die Zusam-menarbeit »partnerschaftlich« (Lösungsarchitekt Softwarefirma).

In diesem Fall verwenden viele EVU und ihre unzähligen Anwendenden eine Standardsoftware und es steht ihnen nur das Ticketsystem und andere formale Wege der Kommunikation mit der Softwarefirma offen. Weil die Softwarefirma eine große Kund-schaft hat, ist eine stärker formalisierte Kommunikation notwendig. Dies hat Nachteile und Befragte ziehen die direkte Kommunikation vor:

»Also, wenn man einen direkten Draht hat, dann ist es gut. Dann läuft es. Aber die Kommunikation rein über das Ticketsystem ist... Meiner Meinung nach verhindert das eine schnelle Lösung.« (Gruppenleiter Abrechnung EVU5)

»Da ist es in der Kommunikation meistens wesentlich einfacher, wenn man direkt mit dem Kunden dann spricht und sagt: ›Hier, wie siehts aus?‹ Mit dem Bearbeiter halt wirklich, weniger dann mit den Entscheidungsebenen, sondern mit dem Benutzer wirklich der Software.« (Programmierer)

Unabhängig, ob Ticketsystem, E-Mail oder Telefon: Einige befragte Personen betonen auch bei der zentralisierten Softwaregestaltung den Vorteil, wenn es langfristige Ansprechpersonen und Netzwerke in der Softwarefirma gibt und man die gleiche Sprache spricht. Im EVU5 gibt es einen Mitarbeiter, der Programmierer in der Softwarefirma war und nun Berater ist und der beide Seiten versteht: Er kann die Anforderungen und Probleme der Anwendenden verstehen und weiß zugleich, was mit der Software möglich ist. Eine Befragte hat die Erfahrung gemacht, dass eine höfliche, sehr klar formulierte und gradlinige Kommunikation hilft, schnellere Antworten zu bekommen. Kolleg:innen, die das nicht machen, sind schon »auf die Nase gefallen« (Anwendungsbetreuerin EVU3).

Digitale Werkzeuge: Ticketsystem – digitale Vernetzung und Detaillierung

In dem Fall setzt die Softwarefirma digitale Werkzeuge für eine zentralisierte Softwaregestaltung auf typische Weise ein: um die Softwaregestaltung zu koordinieren und zu kontrollieren und weniger, damit Anwendende direkt Anforderungen aufnehmen können. Es geht darum, die Kommunikation und den Fluss an Fehlern und Anforderungen der umfangreichen Kundschaft der Softwarefirma zu kontrollieren. Zwei Befragte meinten, dass 90 % der Zusammenarbeit mit der Softwarefirma über Tickets läuft. Der Rest sind E-Mails an konkrete Ansprechpersonen. Ein anderer Interviewter meint, dass alles über Tickets läuft. EVU3 gibt ca. 400 Tickets im Jahr auf. Für die Sortierung der Tickets können die EVU eine Priorisierung hinterlegen und haben drei Typen zur Auswahl: Anforderungen, Fehler oder Unterstützung. In der Softwarefirma gibt es Ticketmanagernde, deren Aufgabe es ist, ein Auge auf sämtliche Tickets zu haben, und wenn z.B. etwas eskaliert, leiten sie es an das mittlere Management oder eine Teamleitung weiter.

Softwaretechnischer Zuschnitt: zentrale Standardisierung und Priorisierung durch die Softwarefirma
Der für einen zentralen Arbeitsprozess der Softwaregestaltung typische softwaretechnische Zuschnitt für einen Standard zeigt sich in diesem Fall sehr klar. Dabei ist die Konsensfindung über den Standard in der Softwarefirma zentralisiert. Sie hat letztendlich die Entscheidungshoheit. Größere EVU haben mehr Einfluss u.a. durch die Teilnahme an Arbeitskreisen. Bei kleineren EVU geht es vor allem um eine günstige Software. Sie folgen dem Standard, weil das für sie billiger ist.

»Bei kleineren ist es wirklich ein reines Kostending. [...] Bei größeren Werken wollen auch zwar die Kosten senken, möchte aber gerne immer ganz viel individuelles Zeugs beibehalten.« (Lösungsarchitekt Softwarefirma)

Laut drei Befragten haben große EVU mehr Einfluss auf die Software. Dies liegt daran, dass ihr Kundenstamm größer ist und ein Fehler in der Software dementsprechend kostspielig und folgenschwer sein kann. Zwischen Kosten und Anpassungen muss jedes EVU für sich eine Lösung finden.

»[ERP-Software] ist ein Standard und das Stadtwerk [Name] oder was auch immer, hat aber kein Interesse, weil der hat fünf Fehler davon und wir haben vielleicht 5000 Fehler davon. Dann haben wir natürlich ein viel höheres Interesse daran, dass es [...] die Software das vollumfänglich abarbeitet.« (Gruppenleiter EVU3)

Es ist noch einfacher als bei KOOP1, im zentralisierten Softwaregestaltungsprozess regulatorisch notwendige Umsetzungen oder solche, die sämtliche EVU betreffen, in den Standard aufzunehmen. Wenn einzelne EVU sich nicht dem Standard unterordnen, dann vor allem bei wettbewerbsdifferenzierenden Prozessen, z.B. wenn EVU Anschreiben oder Rechnungen an ihre Kundschaft senden:

»Ich habe schon das Gefühl, wir werden so ein bisschen in die Standardlösung geprésst. Wir haben immer noch sehr individualisierte Schreiben und so weiter. So viele Sachen, die sind für uns grundsätzlich unumstößlich. Wir werden jetzt keine Standardschreiben rausschicken wie 90 Prozent der [Software]-Anwender. Die müssen schon individualisierbar sein.« (Anwendungsbetreuung EVU3)

8.3.2.6. KOOP3: zentrale Softwaregestaltung durch Softwarefirma in Ko-Produktion mit IT-DL

In dem Fall nutzen die beteiligten Organisationen beide Möglichkeiten der Softwaregestaltung: Die Softwarefirma gestaltet eine Standard-IoT-Software und das IT-DL und einige EVU erweitern diese individuell. Das heißt, für den Kern der Software existiert ein zentraler Arbeitsprozess der Softwaregestaltung. Das Besondere ist aber nun im Unterschied zu den vorhergehenden Fällen, dass Hierarchien und Marktbeziehungen, auch wenn sie zwischen und innerhalb der Organisationen vorhanden sind, kein Hindernis für die Softwaregestaltung darstellen. Denn die kommunikativen Beziehungen zwischen IT-DL und SF sind sehr kooperativ, direkt und durch eine interpersonale Beziehungsfähigkeit gekennzeichnet, wodurch die Mitarbeitenden Konflikte informell lösen. Zusätzlich dazu sind nur wenige an der Softwaregestaltung beteiligt und die interdisziplinären Wissensgrenzen sind gering, wodurch sich letztendlich Anforderungen unkompliziert aufnehmen, ausarbeiten und umsetzen lassen. So kombiniert der Fall in den Rollen und Abläufen Eigenschaften einer zentralisierten und dezentralen Softwaregestaltung. Letztendlich kann von einer Ko-Produktion gesprochen werden, weil IT-DL und Softwarefirma beide ein Interesse daran haben, dass das IoT-Kernmodul der Softwarefirma weiterentwickelt wird, alle ihren Teil kooperativ beitragen und das IT-DL eigene Module an das Kernmodell andockt. Anders als bei PAKET existiert zwischen IT-DL und Softwarefirma keine reine Lieferbeziehung. Was in den Standard des IoT-Kernmoduls kommt, darüber entscheidet aber letztendlich die Softwarefirma.

Rollen: Projektmanager, Account Manager, Product Owner

Das Besondere an dem Fall ist, dass zwar die für einen zentralisierten Arbeitsprozess einer Standardsoftware typischeren koordinierenden Rollen existieren. Diese verteilen sich aber auf Softwarefirma, IT-DL und einzelne EVU. Anders als für eine zentralisierte Softwaregestaltung typisch, sind die Rollen weniger spezialisiert, weil die Implementierungsprojekte, welche die Hauptquelle von neuen Anforderungen sind, nur wenige Beteiligte haben. So mischen sich in den Rollen die Aufgaben des Koordinierens, der Aufnahme und der Ausarbeitung von Anforderungen.

Die besondere Aufgabe des IT-DL ist es in diesem Fall, in die Energiewirtschaft zu vermitteln. Er hat Rollen, die zwischen EVU und Softwarefirma koordinieren und zugleich Anforderungen sammeln und an die Softwarefirma übergeben, damit diese in die Standard-IoT-Software einfließen. Der Product Owner des IT-DL beschäftigt sich unabhängig von einzelnen IoT-Implementierungsprojekten kontinuierlich mit der Weiterentwicklung der IoT-Standard-Datenplattform. Darüber hinaus betreut er die vom IT-DL selbst entwickelten individuellen Module.

Einzelne EVU setzen sich intensiver mit dem Thema IoT auseinander und schaffen interne Positionen dafür. Ein befragtes EVU hat einen eigenen Product Owner für IoT etabliert. Er spricht sich intern mit Projektleitenden ab, welche die Implementierungsprojekte durchführen. Er ist Ansprechpartner für die eigene Kundschaft (das EVU betreut andere EVU).

In der Softwarefirma selbst gibt es Rollen, welche die Zusammenarbeit mit dem IT-DL und den EVU koordinieren, wobei sie auch Anforderungen sammeln und ausarbeiten. Wenn die Softwarefirma selbst Kontakt zu den EVU hat, dann vor allem über Projektmanagende. Das ist aber selten, weil die Zusammenarbeit mit EVU meist über das IT-DL läuft. Sie nehmen Anforderungen auf und geben sie an die Programmierenden weiter. Der Account Manager der Softwarefirma ist für den ersten Kontakt mit interessierten Unternehmen zuständig, bevor es zur technischen Umsetzung kommt. Er und das Projektmanagement haben, wie es für eine Standardlösung typisch ist, auch noch die Aufgabe, die Kundschaft zu beraten und ihre Erwartungen mit den Möglichkeiten der Software in Einklang zu bringen. Es geht darum, die Kundschaft darauf einzustellen, welche Gestaltungsmöglichkeiten es gibt und wann eine Anforderung Teil des Standards wird.

Ablauf: Anforderungstreffen, Projekte, Scrum

Auch wenn es sich in dem Fall um eine zentralisierte Softwaregestaltung handelt, Softwareanwendung, -gestaltung und Programmierung nicht in einer Organisation stattfinden und unterschiedliche Treffen Anforderungen für den Standard zum Thema haben: Sammeln und Ausarbeiten bilden jeweils gleichwertige Schwerpunkte des Ablaufs, weil aufgrund der wenigen Beteiligten die Koordination weniger kompliziert als in anderen Fällen ist. Zudem sind in diesem Fall trotz der zentralen Softwaregestaltenden Verhandlungen und Konfliktlösungen kein wesentlicher Bestandteil des Ablaufs. Er ist unbürokratisch.

Um zentral über energiewirtschaftliche Anforderungen zu sprechen, gibt es monatliche Treffen zwischen dem IT-DL und der IoT-Softwarefirma. Dort übergibt das IT-DL

Anforderungen an die Softwarefirma und beide diskutieren über sie. Die Anforderungen stammen aus IT-Projekten mit EVU und den dabei stattfindenden (wöchentlichen) IoT Jour fixes. Dort haben IT-DL und EVU Anforderungen gesammelt und über sie diskutiert, um sie dann alle ein, zwei Monate in dem Termin mit der Softwarefirma zu besprechen.

Auch die IoT-Softwarefirma macht Implementierungsprojekte, in denen sie Anforderungen sammelt. Doch betreffen solche Projekte meist Kundschaft außerhalb der Energiewirtschaft.

Intern nutzt die Softwarefirma für die Umsetzung und Ausarbeitung der Anforderungen Scrum. Diese iterative Entwicklungsmethode bietet sich an, weil auch die Anforderungen iterativ entstehen: peu à peu durch die Implementierungsprojekte oder neuen Ideen der Softwarefirma, des IT-DL oder der EVU.

Kommunikative Beziehungen: Reziprozität und persönliche Beziehungspflege

Die für einen zentralen Arbeitsprozess der Softwaregestaltungen typischen bürokratischen und spannungsgeladenen Beziehungen gibt es nicht. Das liegt daran, weil die Marktbeziehungen zwischen Softwarefirma, IT-DL und EVU und die Hierarchien innerhalb der Organisationen die Kommunikation nicht behindern. Es gibt direkte Kommunikation unabhängig von Hierarchien oder Verträgen, basierend auf Reziprozität, gemeinsamem Mehrwert bzw. gemeinsamer Ko-Produktion, längerfristigen Beziehungen und einer gemeinsamen Wissensbasis.

Das IT-DL kann durch längerfristige Zusammenarbeit und eigenes Know-how zu IoT der Softwarefirma genau sagen, was zu tun ist und wo das Problem liegt. Es hat sich zwischen beiden bereits eine gemeinsame Sprache etabliert. In diesem Zusammenhang kommt dem IT-DL das umfassende Know-how zugute, das es im Laufe der Zeit zur IoT-Software und deren Anwendung in der Energiewirtschaft erworben hat. Wenn Anforderungen direkt von den EVU kommen, ist das nicht immer der Fall. Dann ist es aufwendiger, zu einem gemeinsamen Verständnis zu kommen, damit letztendlich das umgesetzt wird, was das EVU erwartet und aus Sicht der Softwarefirma möglich ist.

Konflikte sind kein Hemmnis für die Kommunikation, z.B. wenn das befragte EVU1 andere Erwartungen an einen Umsetzungszeitraum hat. IT-DL und EVU1 klären Konflikte informell z.B. via Telefon. Es findet ein Erwartungsabgleich statt. Die unerfüllten Erwartungen und die divergierenden Vorstellungen werden angesprochen, was zur Konfliktlösung beiträgt.

»Ja, also Konflikte gab es auch mal. Ich sage mal, es passiert glaube ich relativ automatisch mal. [...] Aber ich glaube... Kein Konflikt der schriftlich erfolgt, sagen wir es mal so. Ganz normal am Telefon oder persönlich. Man sagt, dass man das Thema unprofessionell findet und eine andere Erwartungshaltung hat. So lief es bei uns auf jeden Fall bis jetzt. [...] Aber wir machen es immer rein auf der persönlichen Ebene.« (Teamleiter IT EVU1)

Die Zusammenarbeit der Firmen ist geprägt durch Vertrauen und die Reziprozität der Ko-Produktion. Der direkt erlebte und vorführbare Mehrwert in Form der Software, d.h. die Daten aus den Sensoren in die bestehende IT-Landschaft der EVU zu integrieren,

ist für die Beteiligten grundlegend für eine erfolgreiche Zusammenarbeit. Sowohl beim IT-DL als auch beim befragten EVU1 ist es möglich, kleinere Piloten für IoT ohne größeren bürokratischen Aufwand durchzuführen. Stellenweise werden Projekte ganz ohne Verträge gemacht, wie EVU1 berichtet:

»Also das sind so, ich will nicht sagen, bisschen Pilotcharakter oder, ich will nicht sagen, wie kann man das sagen: Einstieg zu einem ›proof of concept‹, zu einem neuen Produkt, der aber noch keine vertragliche Relevanz hat. Also vertraglich irgendwie zugesichert ist. Sondern beide geben einen gewissen Aufwand rein, ohne ihn vorher zu klassifizieren oder zu monetarisieren.

I: Aber es gibt Verträge bei solchen Projekten oder ist das komplett ohne Verträge?

B: Sehr oft ohne Vertrag.« (Teamleiter IT EVU1)

Das IT-DL hat den Vorteil gegenüber der Softwarefirma, dass es bereits länger in der Energiewirtschaft aktiv ist und es die EVU kennen. Manche EVU sind Gesellschaftende vom IT-DL. Das heißt, nicht nur die Beziehung zwischen IT-DL und Softwarefirma ist eng, sondern auch jene zwischen IT-DL und EVU. Weder Markt noch Hierarchie prägen allein die Zusammenarbeit. Verträge, partnerschaftliche Beziehungen und Projektkoordination vermengen sich.

»Wenn du den Kunden seit 20 Jahren kennst, dann sagst du: ›Du, pass auf, ich habe das intern [im EVU] geklärt. Wir machen das jetzt, wir können das entwickeln. Aber ich brauche noch drei Wochen, bis das durch den Einkauf geht.‹ Weil solche Prozesse halt mal sehr langsam sein können. Dann sagen wir: ›Ja, okay, wir kennen uns seit 20 Jahren, wir gehen in die Entwicklung.‹« (Product Owner IT-DL)

Digitale Werkzeuge: Ticketsystem, Test-Accounts

In dem Fall zeigt sich in dem EVU die für digitale Werkzeuge typische Verwendung in zentralen Arbeitsprozessen der Softwaregestaltung in abgeschwächter Form: Denn es gibt zwar ein Ticketsystem der IoT-Softwarefirma, das vor allem dazu dient, die Anforderungen oder Fehler der diversen Kundschaft kanalisiieren zu können. Ca. 100 Tickets bearbeitet die Softwarefirma im Monat (Stand 2021). Doch wurde 2021 probeweise dem IT-DL Zugriff auf das Ticketsystem gegeben. Er kann dadurch Tickets für die Softwarefirma nicht nur anlegen, sondern auch bearbeiten. Das sollen in Zukunft auch die EVU können. Das hat den Vorteil, dass das IT-DL Anforderungen sieht und direkt mit dem EVU reden kann, dass das jeweilige Ticket aufgenommen hat. So unterstützt das Ticketsystem die Ko-Produktion, bei der zentral die Softwarefirma den Standard und IT-DL und EVU dezentral individuelle Module programmieren. Sonst nutzt das IT-DL für die Übergabe von Anforderungen aus den EVU an die Softwarefirma pragmatisch E-Mails, Excel-Dateien oder andere Softwarelösungen. Wie für dezentrale Softwaregestaltung typisch, dienen sie vor allem dazu, dezentral Anforderungen direkt aufzunehmen und zu dokumentieren.

Softwaretechnischer Zuschnitt: zentral durch IoT-Softwarefirma

In dem Fall zeigt sich der für einen zentralisierten Softwaregestaltungsprozess klare Standard-Zuschnitt. Wobei zu diesem Schnittstellen gehören, so dass IT-DL oder EVU dezentral individuelle Erweiterungen anlegen können.

Grundsätzlich legt die IoT-Softwarefirma die Priorisierung fest und teilt mit, bis wann sie was umsetzt. Sie entscheidet darüber, was Teil des Standards wird. Nur bei etwas Grundlegendem oder wenn mehrere EVU des IT-DL etwas wollen (laut einem Befragten so ab drei bis vier EVU), konzeptionieren und priorisieren IT-DL und Softwarefirma gemeinsam.

8.3.2.7. STARTUP: dezentral in Kreisen – eklektische Mischung aus Holokratie und Scrum

In dem Fall liegt ein dezentraler Arbeitsprozess der Softwaregestaltung für eine individuelle Software innerhalb von STARTUP vor. Von Anfang an hat STARTUP darauf gesetzt, die Softwaregestaltung selbst in die Hand zu nehmen. Der Fall ist ein Beispiel für den Primat der Softwareentwicklung einer Organisation. Deshalb gibt es innerhalb der Organisation für den Arbeitsprozess keine Hürden durch Hierarchien oder Abteilungsgrenzen der Softwareanwendung. Es existiert vielmehr eine rollenbasierte Organisation, die stark darauf konzentriert ist, in einer Mischung aus Scrum und Holokratie Anforderungen auszuarbeiten und umzusetzen. Wie für einen dezentralen Arbeitsprozess der Softwaregestaltung typisch, sind die kommunikativen Beziehungen offen, direkt, flexibel und werden durch ein gemeinsames Ziel gestützt. Neben einem Ticketsystem nutzt STARTUP Chat-Kanäle, die den dezentralen Input von Gestaltenden, Programmierenden und Anwendenden für Anforderungen ohne bürokratische Hindernisse erlauben. Die individuell für die eigene Organisation gestaltete Software bietet STARTUP anderen Organisationen wie Automobilfirmen als Standard-White-Label-Lösung an.

Rollen: Product Owner, Gründer, Solution Architect

Wie für einen dezentralen Arbeitsprozess einer individuellen Softwaregestaltung typisch, bestehen an die Rollen vor allem die Erwartungen, Anforderungen aufzunehmen und auszuarbeiten. So ist die befragte Product Ownerin sehr wenig mit Koordination beschäftigt. Sie prüft die rechtlichen Voraussetzungen, z.B. wer bei Firmenwagen die monetäre Pauschale bekommt, stimmt sich mit Behörden ab, entwickelt Roadmaps für die Produktentwicklung und lässt Feedback des eigenen Support-Teams der mobilen Anmeldungsapp in die Software einfließen. Zudem optimiert sie die Anmeldungsprozesse und beantwortet besonders schwierige Support-Anfragen. Ganz ohne Koordinierung geht es aber nicht.

»Ich muss alles im Blick haben und bin die Brücke zwischen Produkt-, Tech-Team und Kommunikation. Ich koordiniere und kommuniziere intern (Meetings, Tickets, Pläne, Dokumentation...).« (Product Ownerin Anmeldung E-Autos)

Einer der Gründer macht als Geschäftsführer bei der Softwaregestaltung mit. Er hat Prozesswissen zum Emissionshandel, übernimmt Aufgaben im Vertrieb, im Business Development und ist auch an der Produktentwicklung beteiligt.

»[Der] Geschäftsführer so Wirtschaftsinformatiker ist, der beide Seiten versteht. Und das ist sehr, sehr fruchtbar, weil er versteht beide Sprachen: einmal die Wirtschaftler und einmal die Techies. Und dann ist er ein sehr guter Mediator und setzt dann auch angemessene Prioritäten und Deadlines und so in Absprache auch mit uns.« (Programmierer2)

Auch in dem Bereich, der sich um die Anmelde-Software kümmert, ist er als Moderator tätig und wacht darüber, dass die Termine für die Teamtreffen stattfinden. Er agiert damit ähnlich wie ein Scrum Master. Neben diesen Kernrollen, die Anforderungen für die Programmierenden schreiben, haben in der rollenbasierten Organisation auch noch andere als Teil ihrer Rolle, bei Bedarf an Konzepten mitzuschreiben: z.B. die Programmierenden selbst, der Solution Architect oder Personen aus dem Kommunikationsbereich von STARTUP.

Der Fall zeigt, wie andere Fälle auch, dass es den Arbeitsprozess der Softwaregestaltung auszeichnet, dass Beteiligte je nach Konstellation auch mal mehrere Rollen ausfüllen – situativ angepasst und wechselnd. Wobei es hier so weit geht, dass alle gemeinsam regelmäßig in dafür vorgesehenen Treffen darüber verhandeln, welche Rollen jemand zusätzlich zu seiner Stammrolle übernimmt oder ob er gar seine Stammrolle wechselt.

Ablauf: Kreise, Scrum

Der Ablauf besteht primär, wie für einen dezentralen Arbeitsprozess der Softwaregestaltung typisch, im Ausarbeiten und Sammeln von Anforderungen. Die Organisation ist von Anfang an auf die Softwaregestaltung ausgerichtet. Es besteht ein enger und pragmatischer Austausch. Es stören keine Hierarchien oder althergebrachte, auf die Softwareanwendung ausgerichtete Organisationsstrukturen dabei, die Möglichkeiten der Softwaregestaltung auszuschöpfen – anders als bei einer Softwaregestaltung innerhalb von EVU.

Der Ablauf der Konzeptionierung zeichnet sich durch direkte Feedbacks und den direkten Austausch zwischen Anwendung, Gestaltung und Programmierung aus: Erst trifft man sich in einem Kreis, dann werden die Anforderungen in Scrum-Arbeitsweise abgearbeitet. Die offenen Anforderungen stehen im Ticketsystem Jira: im Backlog jene, die priorisiert sind, und auf dem Kanban-Board¹⁶ jene, die aktuell die Programmierenden bearbeiten. An den Kreisen nehmen Product Owner:innen, Geschäftsführer und Programmierende teil, um Aufgaben zu planen und Anforderungen für die Softwaregestaltung zu erarbeiten.

Der Produkt-Kreis für die Anmeldesoftware findet wöchentlich statt. Die Product Ownerin lädt dazu Programmierende, bei Bedarf auch jemanden aus der Kommunikationsabteilung und, wenn auch selten, Kundschaft ein. Mit einem Kunden (einem Automobilkonzern, der die App zur E-Auto-Anmeldung nutzt) hat STARTUP eng zusammen-gearbeitet, um dessen Bedürfnisse bei der Entwicklung zu berücksichtigen. Der Kreis

¹⁶ Darunter ist erst einmal nur die Visualisierung der Arbeitsschritte in eine Software gemeint und nicht die Kanban-Methode. Meist ist die Darstellung ganz einfach und unterteilt die Aufgaben in solche, die noch offen sind, gerade bearbeitet werden und abgeschlossen sind.

bespricht Konzepte und nimmt Ideen aus dem Support auf, worüber auch Anforderungen der Anwendenden einfließen. Den Backlog für die Programmierenden pflegt im Bereich der Anmelde-Software »hauptsächlich« (Product Ownerin) die Product Ownerin.

Scrum lebt STARTUP im Bereich der App für die Anmeldung von E-Autos in folgender Version, was die enge und unkomplizierte Zusammenarbeit zwischen Gestaltenden und Programmierenden verdeutlicht:

»Wir machen meistens einen Termin, in dem wir besprechen, was die Vorstellungen sind und wie das dann technisch umsetzbar ist. Das halte ich fest und wir machen erste Aufgabenstellungen daraus. Das wird dann bei den Scrum-Meetings gemeinsam nochmal durchgegangen. Für [Teil des Start-ups] nehme ich dann Ideen und Entwicklungen ab. In dem Scrum [Entwicklerteam eines Teils des Start-ups] sind 5 Personen, aber niemand in Vollzeit. Wir machen keine täglichen Stand-ups, sondern nur wöchentlich. Ansonsten machen wir Planning und Review inkl. Retro. Hinzu kommen dann bei größeren Aufgaben auch das Epic-Planning. Es gibt feste Sprints.« (Product Owner)

Auch diese Fallstudie zeigt, wie die Firmen für die Softwaregestaltung Methoden wie Scrum anpassen. STARTUP setzt kein reines Scrum um. Zum Beispiel gibt es keine täglichen Treffen, die sogenannten *Daylies*. Größere Themen teilen die Beteiligten in mehrere Aufgaben auf und deren Bearbeitung ist dann wieder in der Rollen- und Kreis-Struktur organisiert.

Was STARTUP auszeichnet und typisch für dezentrale Softwaregestaltung ist: Neben Product Owners als übliche Anforderungsmanagende kann jede Person informell Anforderungen an die Programmierenden stellen:

»Bei uns läuft da bei weitem nicht alles unbedingt die definierten Wege, sondern oft viel pragmatischer. Wenn dann jemand von der Kommunikation direkt zu mir kommt und sagt: ›Hey, ich hatte folgende spannende Idee und was hältst du davon?‹« (Programmierer1)

Feedback ist zum einen nach dem zweiwöchigen Sprint dadurch möglich, dass die Programmierenden »eigentlich immer« (Product Owner) Prototypen vorstellen. Zum anderen gibt es, wie in der Softwareentwicklung üblich, Tests. Die Tests laufen, typisch für diesen Fall, unbürokratisch ab – »wie es grade passt« (Product Ownerin).

Für das Software-Modul zum Handel mit Emissionszertifikaten treffen sich die Beteiligten dreimal wöchentlich. Dort ist zusätzlich noch ein Solution Architect im Kreis mit dabei. Zum Zeitpunkt der Befragung wurde der Sprint-Zyklus ausgesetzt, weil es einige dringende Themen abzuarbeiten galt, was die Flexibilität des dezentralen Ablaufs verdeutlicht.

Kommunikative Beziehungen: flexibler, direkter und offener Austausch

In dem Fall existieren die für einen dezentralen Arbeitsprozess der Softwaregestaltungen typischen direkten und langfristigen Beziehungen und eine gemeinsame Sprache. Die Befragten betonen den Vorteil direkter Kommunikation und wie offen und unkompliziert die Beziehungen die Kommunikation machen. Hemmnisse durch Hierarchien,

Abteilungs- oder Teamgrenzen oder Marktbeziehungen gibt es nicht. Als Hürde für direkte Kommunikation und schnellen Austausch ohne Termin sehen Befragte die überwiegende Home-Office-Arbeit.

Ein befragter Programmierer führt den offenen Austausch darauf zurück, dass Mitarbeitende nicht untereinander konkurrieren:

»Das kenn ich normalerweise ganz anders bei den anderen Unternehmen. Meiner Erfahrung nach ist das immer so, dass Leute immer ein bisschen ihre Projekte oder sehr darauf geachtet haben, was halt andere sehen, was umgesetzt wird. Und so weiter. Also so ein bisschen diese Konkurrenzsituation und das nehme ich bei uns gar nicht wahr, sondern ganz im Gegenteil. Man kümmert sich untereinander.« (Programmierer1)

Da einige Mitarbeitende Teil mehrerer Kreise sind (bspw. Product Ownerin oder Geschäftsführung), tauschen sich diese nicht nur innerhalb eines Teams aus, wie das in anderen Organisationsformen der Fall wäre.

Die Kommunikation ist kontinuierlich, flexibel und immer direkt möglich – ob durch Treffen oder Chat. Die Product Ownerin sieht, was direkte Ansprache betrifft, intern keine Hürden. Laut einem befragten Programmierer werden Anforderung auch außerhalb der Kreis-Treffen aufgenommen:

»Das ist alles nicht festgefahren, sondern man weiß genau, wenn sich jetzt jemand meldet und sagt: ›Du, ich habe folgendes Problem‹, und das kurz beschreibt, dann kann man also, wenn es notwendig ist, auch alle Planungen irgendwie umwerfen. Und auch mal, ja und sich halt um das kümmern, ganz pragmatisch, was halt gerade notwendig ist. Und da gibt es dann keine Leute, die irgendwie blöd kucken oder sagen: ›Ja puh, aber wir wollten doch dies, das oder so.‹ Sondern da ziehen halt alle am gleichen Strang, ohne viele persönliche Befindlichkeiten, die dahinterstecken, ohne irgendwelche Ego-Geschichten.« (Programmierer1)

Programmierer2, der den Handels-Software-Teil programmiert, meint, dass er auch selbst etwas in das Ticketsystem Jira einpflegt, wenn er einen Fehler an der Software feststellt. Zudem schreiben interne Anwendende und Kund:innen per E-Mail oder Chat Anforderungen oder zumindest Ideen für solche.

Der regelmäßige Austausch in den Kreisen verringert mit der Zeit die sprachlichen Hürden zwischen den Beteiligten. Nur durch die getrennten Kreise besteht noch eine Wissensgrenze, weswegen die Product Ownerin zwischen diesen vermitteln muss. Dem Kommunikations-Kreis fehlt manchmal der Überblick über Prozesse wie »Anmeldung E-Autos« oder »Handel mit Emissionszertifikaten«. Der Produkt-Kreis wiederum bekommt die Anforderungen der Kommunikation nicht automatisch mit. Diese Grenzen sind für die Softwaregestaltung allerdings sekundär, weil alle Kreise unabhängig an einem Teil der Software arbeiten können.

Digitale Werkzeuge: vor allem für direkten Input – Ticketsystem, Chats, E-Mails

In dem Fall zeigt sich in dem STARTUP die für digitale Werkzeuge typische Verwendung in dezentralen Arbeitsprozessen der Softwaregestaltung. Das Ticketsystem, in das die

Product Owner:innen den Backlog pflegen und dessen Kanban-Board die Programmierenden für die Sprints nutzen, dient vor allem dazu, Anforderungen aufzunehmen, und weniger zur Überwachung oder Standardisierung. STARTUP nutzt die ganze Bandbreite an Kommunikationskanälen, um sich direkt abzusprechen und direkt Input für Anforderungen zu sammeln und Anforderungen auszuarbeiten: E-Mails, Telefon und Microsoft Teams (Videokonferenzen, Chat-Kanäle). Durch den Chat (MS Teams) können Beschäftigte direkt eine Person aus der Kollegenschaft anschreiben und eine Chat-Gruppe aufmachen, um eine Anforderung oder ein Thema zu bearbeiten.

Softwaretechnischer Zuschnitt: individuell und eigenständig durch Start-up

Über den für einen dezentralen Arbeitsprozess der Softwaregestaltung typischen individuellen softwaretechnischen Zuschnitt entscheidet das STARTUP eigenständig. Wobei die Teilnehmenden innerhalb der Kreise gemeinsam darüber entscheiden, was die Software können soll. Die Product Ownerin priorisiert, aber diskutiert dann mit anderen, was am wichtigsten ist. Nur wenn es nicht ganz klar ist, entscheidet sie. Dabei kann die Priorisierung flexibel verändert werden, d.h., es gibt keinen eindeutigen formal-bürokratischen Prozess dafür. Anforderungen von Nutzenden der Anmelde-App fließen ein, wenn sie als sinnvoll erachtet werden und es die Software nicht zu kompliziert macht.

Individuelle Anpassungen für Firmen, welche die Software zur Anmeldung von E-Autos als White-Label-Lösung einsetzen, gibt es nicht.

8.3.3. Zusammenfassung

Das Resümee fasst die Unterschiede von zentralisierter und dezentraler Softwaregestaltung je Kategorie des Arbeitsprozesses zusammen und stellt allgemeine Aussagen zu den Kategorien auf.

8.3.3.1. Unterschiede zwischen dezentralen und zentralisierten Arbeitsprozessen der Softwaregestaltung

Die Fallstudien INTERN1, INTERN2 und STARTUP können dem Typ dezentral und KOOP1, KOOP2, KOOP3 und PAKET dem Typ zentralisiert zugeordnet werden. Doch zeigen die Falldarstellungen, dass sie nicht immer klar einem der diametralen Typen entsprechen und sich teilweise dezentrale und zentralisierte Formen mischen. Hier seien zum Abschluss des Abschnitts die Einordnungen der Fälle kurz begründet.

Ob dezentrale oder zentralisierte Softwaregestaltung: Egal welche Arbeitsteilung, Grundkoordination oder Architektur vorliegt, findet in den Fallstudien durch den entsprechenden AP die Softwaregestaltung statt. Für Fälle wie STARTUP, bei denen die Organisation auf die Softwaregestaltung ausgerichtet ist, ist aber der Koordinationsaufwand deutlich geringer. In Fällen wie KOOP1 ist der Koordinationsaufwand größer, dafür bestehen aber auch erhebliche Möglichkeiten, Synergien durch einen Standard zu heben. Bei PAKET müssen sich die EVU dem von der Softwarefirma gestalteten Standard unterordnen.

In jedem Fall übernehmen die **Rollen** Aufgaben der Koordination und der Anforderungsaufnahme oder -ausarbeitung. Die Schwerpunkte sind jedoch unterschiedlich. Bei der **zentralisierten** Softwaregestaltung der Fallstudien KOOP1, KOOP2, PAKET und

KOOP₃ sind extra Rollen mit der Koordination beschäftigt: Anforderungsmanagement, Key Account Managende, IT-Projektleitung, IT-Koordination, Prozessmanagement, Lösungsarchitekt oder Anwendungsbetreuung. Das liegt schlicht an den zusätzlichen Koordinationsaufgaben aufgrund der Arbeitsteilung zwischen Anwendung und Entwicklung, die sich auf mehrere Organisationen erstreckt und wenn mehrere Organisationen eine Software gestalten wollen. Bei KOOP₃ ist aufgrund der netzwerkförmigen Grundkoordination der Koordinationsaufwand geringer, weil die Softwaregestaltung keine Hierarchien und Marktbeziehungen überwinden muss. Bei **dezentralen** Softwaregestaltungen wie INTERN₁, INTERN₂ und STARTUP liegt der Fokus darauf, Anforderungen aufzunehmen und an die Programmierenden zu übergeben. Koordinative Aufgaben fallen zwar auch an, aber in einem geringeren Umfang. Wobei bei INTERN₂ auch mehr Koordinationsaufwand besteht, den einzelne Mitarbeitende erledigen müssen, weil mehrere Fachbereiche bei der Softwaregestaltung zusammenarbeiten. Bei STARTUP ist der Koordinationsaufwand geringer: Es existiert eine Netzwerkorganisation, d.h., die Beschäftigten müssen keine Hierarchien, Abteilungs- oder Organisationsgrenzen bei der Softwaregestaltung berücksichtigen.

Die **Abläufe** sind in den **zentralisierten** Fällen aufgrund der Arbeitsteilung darauf ausgerichtet, immer wieder Erwartungen zwischen den beteiligten Organisationen abzugleichen, mit Eskalationen umzugehen und Konflikte zu lösen. Bei KOOP₁, KOOP₂ und PAKET finden Abstimmungen auf mehreren Ebenen zwischen den Organisationen statt. Strategische Themen verhandeln Führungskräfte bzw. Manager:innen in entsprechenden Gremien oder Terminen. Für die Abstimmung über einzelne Anforderungen gibt es separate Abläufe, wie das Anforderungsmanagement bei KOOP₁ oder Arbeitskreise bei PAKET. KOOP₁ setzt einen Mediator und Key Account Managende ein, um die Beziehungen zu pflegen und mit Konflikten umzugehen. In dem Fall dienen die Abläufe für Verhandlungen darüber, was zentral das IT-DL und was dezentral die EVU gestalten. Bei KOOP₃ gibt es regelmäßig Treffen zwischen dem IT-DL und der Softwarefirma und es entstehen aus Projekten mit EVU neue Anforderungen an die IoT-Software. Allerdings ist aufgrund der netzwerkförmigen Grundkoordination und der kooperativen Beziehungen der Koordinationsaufwand gering. Bei den **dezentralen** Fällen konzentrieren sich die Abläufe auf die Aufnahme und Ausarbeitung von Anforderungen. INTERN₁ nutzt vielfältige Methoden, um die Anwendenden direkt einzubeziehen. Bei INTERN₁, INTERN₂ und STARTUP arbeiten Programmierende und Gestaltende wie Product Owner sehr kontinuierlich und langfristig zusammen. Wobei bei STARTUP aufgrund der Grundkoordination die Abläufe weder Hierarchien noch Märkte überwinden müssen.

Die **kommunikativen Beziehungen** müssen bei den **zentralisierten** Fällen aufgrund der Arbeitsteilung die Spannungen ausgleichen: zwischen der notwendigen Kooperation für die Softwaregestaltung einerseits und hierarchischen und marktförmigen Beziehungen zu anderen Organisationen andererseits. Zudem sind bei den Fällen KOOP₁, KOOP₂ und PAKET bürokratischere Anforderungsprozesse vorhanden. Zum Beispiel müssen Führungskräfte Anforderungen absegnen oder es existiert ein formalisiertes Vorgehen. Bei KOOP₃ ist das aufgrund der netzwerkförmigen Grundkoordination und den kooperativen Beziehungen nicht notwendig. Bei den **dezentralen** Fällen liegen kooperative und direkte Kommunikationsbeziehungen vor. Die unkomplizierte und offene Zusammenarbeit betonen besonders Befragte von STARTUP und INTERN₁. Bei

INTERN1 und vor allem INTERN2 kommen bürokratische und formalisierte Elemente hinzu. In ersterem Fall existieren Hierarchien zwischen den Product Ownern (z.B. wer über Priorisierung entscheidet) und bei INTERN2 entscheidet die Anforderungsrounde über Anforderungen.

Die direkte Kommunikation schlägt sich in den **dezentralen** Fällen INTERN1 und STARTUP in den verwendeten **digitalen Werkzeugen** nieder. In beiden Fällen können Beteiligte der Softwaregestaltung direkt via Chat Anforderungen aufnehmen und sich absprechen, d.h. sich ohne formale Hürden austauschen. Bei den **zentralisierten** Fällen KOOP1, KOOP2 und PAKET dienen die digitalen Werkzeuge dazu (vor allem die Ticketsysteme), um für Transparenz zwischen den beteiligten Organisationen zu sorgen. Vor allem für die EVU-Kundschaft der IT-DL von KOOP1 und KOOP2 oder der Softwarefirma von PAKET ist das Ticketsystem wichtig, um vertragliche Vereinbarungen zu kontrollieren und durchzusetzen (z.B. durch in SLA festgelegte Reaktionszeiten auf Tickets). Aufgrund der netzwerkförmigen Grundkoordination und der kooperativen Beziehungen spielt beim Fall KOOP3 das Ticketsystem der Softwarefirma primär eine koordinierende Rolle. In Zukunft soll es dazu dienen, dass das IT-DL selbst Tickets bearbeiten kann, was den Charakter der Ko-Produktion in dem Fall noch verstärkt.

Bei den **dezentralen** Fällen INTERN1, INTERN2 und STARTUP ist der **softwaretechnische Zuschnitt** individuell. Bei KOOP1 existiert ein ausgefeilter Ablauf, mit dem sich die EVU darüber verständigen, was in den gemeinsamen Standard einfließt und was sie individuell selber gestalten. Bei KOOP2 existiert ein solcher Ablauf nur für bestimmte Teile der Software. In beiden Fällen gestalten einzelne EVU selbst dezentral Software unabhängig von einem gemeinsamen Standard. PAKET entwickelt **zentral** eine Standardsoftware, die allerdings individuelle Einstellungen durch die EVU zulässt. Die Möglichkeiten individueller Einstellungen nutzen einige (vor allem größere) EVU ausgiebig und haben dafür extra Rollen (z.B. Anwendungsbetreuende). Bei KOOP3 entwickelt die Softwarefirma ebenfalls eine Standardlösung, allerdings ohne große Einstellungsmöglichkeiten und dank Schnittstellen erweiterbar durch individuelle Module.

8.3.3.2. Ergebnisse des Fallvergleichs je Kategorie

Die Fallstudien zeigen, dass die Organisationen durch Rollen, Abläufe, kommunikative Beziehungen und softwarebasierte Werkzeuge die Softwaregestaltung kontrollieren und wie der softwaretechnische Zuschnitt der Software zustande kommt. Im Folgenden werden allgemeine Aussagen zu den Kategorien aufgestellt, die sich aus dem Fallvergleich ergeben.

Rollen – situativ, teils mehrere und wechselnde Rollen übernehmen

Im Konzept der soziotechnischen Netzwerkarbeit erklärt der Rollenbegriff, wie Einzelne ihr Handeln auf die Softwaregestaltung ausrichten, ohne dass ihnen genau gesagt wird, was sie in der jeweiligen Situation konkret en détail zu tun haben. Die Fallstudien zeigen, dass das Konzept der Rolle den empirischen Umständen gerecht wird, dass 1. viele der Befragten nicht mehr nur eine Rolle haben, sondern mehrere und wechselnde. Zudem 2. sind die Befragten weniger auf vordefinierte Handgriffe spezialisiert, sondern vielmehr Teil von mehreren Arbeitsprozessen wie Softwaregestaltung und -anwendung und die

Organisation und die oder der Einzelne muss die Rollen entsprechend ausgestalten und kontrollieren: situativ, verhandelt, als Teil eines Arbeitsprozesses zwischen Anwendung und Programmierung. Außerdem findet 3. Rollen-Handeln innerhalb von Hierarchien statt, auch wenn Methoden wie Scrum keine formalen Hierarchien vorsehen.

Tabelle 22: Rollen: reine Softwaregestaltungsrollen oder gemischt mit anderen und wer Schulungen zur Rolle erhalten hat

Fall	Rolle nur für Softwaregestaltung (Anforderungen, Koordination)	Gemischte Rollen (Anwendung, Programmierung, Customizing, (IT) Management, Support für SA)	Schulung Rollen-Kompetenz (in Klammern welche Rolle Schulung absolviert hat)
INTERN1	Product Owner:in (PO)	Key User:innen (KU), führender PO, Anforderungsmanager, Teamleiter als Scrum Master (SM)	PO Schulung (POs), Scrum Schulung und Story Writing (Anforderungsmanager)
INTERN2		KU, Ansprechperson in Fachbereichen, SM, Anforderungsmanager, PO	PO Schulung (Anforderungsmanager, PO), SM Schulung (SM)
KOOP1	Anforderungsmanager (IT-DL), IT-PL, Anwendungsbetreuer	IT-Berater, Prozessmanager, IT-Manager, Anforderungsmanager (EVU)	SM, PO, PM (Anforderungsmanager und PL IT-DL), Schulung Requirements Engineering (Anwendungsbetreuer)
KOOP2	IT-Koordinator Fachbereich, IT-PL	IT-Koordinator IT-Abteilung, KU, Manager Digitalisierung, Teamleiterin, IT-Berater, Change Manager	Scrum (IT-Berater, Change Manager), PM (Change Manager, PL), Change Manager schult intern zu Projektthemen
PAKET	EW-Fachleute in Arbeitskreisen, Prozessbetreuer	Teamleitung Entwicklung, IT-Berater, Lösungsarchitekt, KU, Anwendungsbetreuer	PM (Anwendungsbetreuung)
KOOP3	PO (EVU, IT-DL)	Projektmanager, Account Manager	unbekannt
STARTUP	PO	Solution Architect, Programmierer, Geschäftsführung	PO und Programmierer keine

Zu 1.: In allen Fallstudien gibt es Rollen, die für die Softwaregestaltung zuständig sind. Sie koordinieren oder schreiben selbst Anforderungen. Meistens haben diese Rollenträger noch weitere Verantwortungsbereiche. Wie viele Rollen jemand übernimmt und in wie vielen Arbeitsprozessen, ergibt sich im Wesentlichen aus: erstens der Überschneidung der Softwaregestaltung mit anderen Arbeitsprozessen. Manche Softwaregestaltenden sind in mehreren Projekten aktiv und/oder sind zusätzlich Anwendende und Programmierende der Software. Zweitens ist die Spezialisierung bei kleinen EVU gerin-

ger oder sie verzichten ganz auf eine gestaltende Rolle (z.B. haben einzelne EVU keine eigenen Key User:innen oder internen Support für die Standardsoftware).

Die Überblickstabelle (Tabelle 22) fasst noch einmal zusammen, dass die meisten Softwaregestaltenden mehrere Rollen innehaben. Zudem zeigt sich, dass in vielen Fällen die Rollenträger Schulungen besucht haben, worauf weiter unten näher eingegangen wird.

Zu 2.: Die Fallstudien zeigen nicht nur, dass die Angestellten oftmals mehrere Rollen übernehmen. Zudem erlauben es die Rollen, situativ und unabhängig von einer hierarchischen Position erwartungsgemäß zu agieren. Zwar zeigt die Tabelle 22 oben, dass viele der Befragten für ihre Rolle Schulungen besucht haben. In den Fallstudien sind die Schulungen und Zertifikate für Methoden bzw. Rollen (wie z.B. Product Owner:in oder Projektmanagement) aber nur der Ausgangspunkt dafür, die Rollen in einer Organisation zu etablieren.

Im Arbeitsprozess der Softwaregestaltung reihen sich die Rollen zwischen Anwendung und Programmierung wie an einer Kette aneinander. Den Ablauf der Softwaregestaltung und die Tätigkeiten der einzelnen Rollen genau zu planen, ist jedoch schwierig, weil die Kompetenzen und Wissensbestände der einzelnen Beteiligten, die verbale Ausdrucksfähigkeit und der Mitteilungswille, die soziotechnische Konstellation sowie die Abläufe der Anforderungserarbeitung unterschiedlich sind. Vielmehr findet ein Erwartungsabgleich innerhalb des Netzwerks für Softwaregestaltung statt. Wie situativ das Gestaltungsnetzwerk die Rollen prägt, zeigt sich daran, dass Softwaregestaltende mal mehr, mal weniger energiewirtschaftliches Fachwissen haben und brauchen. Somit ist der Arbeitsprozess eine eigenständige Sozialisationsinstanz, wenn auch in vielen Fällen Schulungen in den jeweiligen Rollen die Grundlage liefern.

Zu 3.: Die Fallstudien zeigen, dass sich Rollen und hierarchische Positionen unterschiedlich zueinander verhalten. Es ist je Fall anders, wie unabhängig von der formalen Hierarchie und von der Position einzelne Rollen agieren können. Vor allem in den reinen Netzwerken von KOOP3 und STARTUP agieren die Rollen unabhängig von hierarchischen Vorgaben. In mehreren Fällen geht eine hierarchische Position mit einer Rolle in der Softwaregestaltung einher. Bis auf STARTUP treffen hierarchisch den Softwaregestaltenden höhergestellte Personen in allen Fallstudien Entscheidungen über Ressourcen wie IT-Budget oder Personal.

Fazit: Für die Beschäftigten bestehen im Arbeitsprozess der Softwaregestaltung zusätzliche Erwartungen, wenn auch in unterschiedlichem Ausmaß und abhängig von der jeweiligen Situation. Zu den Erwartungen gehören jene, zwischen mehreren Rollen zu wechseln, in unterschiedlichen Arbeitsprozessen mitzuarbeiten, sich einzufügen und entsprechend anzupassen. Dazu gehört, mit ihrem jeweiligen Wissen herauszufinden, was zu tun ist. Sie sollen mit nicht immer genau geregelten Verantwortlichkeiten und Erwartungs-/Rollenkonflikten umgehen und vorhandene Spielräume für die Softwaregestaltung nutzen. Zudem besteht die Erwartung, die Rolleneinhaltung mit Kolleg:innen abzustimmen und mit zu kontrollieren. Diese Erwartungen kommen zu jenen aus 6.4.3.2 hinzu: organisationale Grenzen zu überbrücken, kooperativ zu sein, selbstorganisiert zu arbeiten, sich auf Softwareobjekte einzulassen, mit Nicht-Wissen umzugehen oder sich im organisationalen Netzwerk zu bewegen und zu lernen.

Ablauf – kombiniert mit Gestaltungsnetzwerk, Primat der Kooperation, Feedbackschleifen und Lernprozesse

IT-Projekte, Scrum und Anforderungsrunden sind in allen Fällen die wesentlichen Methoden dafür, den Ablauf der Anforderungserarbeitung und deren Übergabe an die Programmierenden zu organisieren. Sie stellen auf unterschiedliche Weise Interdisziplinarität her und werden immer situativ angepasst angewendet, z.B. Scrum ohne Review, mit mehreren Product Owner:innen oder ohne Scrum Master:in. Die Fälle zeigen Folgendes: A) Sie verbinden für die Softwaregestaltung einen mal mehr und mal weniger formalisierten Ablauf mit einem Netzwerk an Beziehungen für Anforderungen. B) Der Ablauf muss vereinbar sein mit verschiedenen Handlungsorientierung, wobei der Primat der Kooperation gilt. C) Für die Softwaregestaltung sind Feedbackmöglichkeiten wichtig, die der Ablauf herstellt. D) Der Ablauf lässt sich an die jeweilige Konstellation und die vorhandenen Rollen anpassen lässt und er in sämtlichen Fallstudien durch Lernprozesse geprägt ist, was seine Ausgestaltung anbelangt.

A) Die Fallstudien zeigen, dass es nicht um den Ablauf (wie IT-Projekte oder Scrum) allein geht, sondern auch um die dazugehörigen Gestaltungsnetzwerke. Zusätzlich zu den Projekten oder zu Scrum existieren in den Fallstudien immer horizontale Netzwerke. Dabei gibt es sowohl klare, formale Anforderungswege als auch informelle Beiträge Einzelner. Formalisiert sind z.B. die regelmäßigen Treffen (wer teilnimmt, Turnus etc.), das Niederschreiben der Konzepte, welche softwarebasierten Werkzeuge verwendet werden und einzelne Rollen, um die Kommunikation unter den Beteiligten am Arbeitsprozess der Softwaregestaltung zu organisieren.

B) Der Ablauf muss die Kombination verschiedener Handlungsorientierungen zulassen (wie schon bei der Projektarbeit unter 6.3 ausgeführt), weil er in unterschiedlichen Grundkoordinationen funktionieren muss. Bei der internen Softwaregestaltung sind es die Hierarchien und bei der firmenübergreifenden Softwaregestaltung die Marktbeziehungen, mit denen die Softwaregestaltenden umgehen können müssen. Es wird erwartet, jenseits ökonomischer Kriterien oder Weisungsbefugnissen informelle Kooperationsmöglichkeiten herzustellen, z.B. trotz formaler Vorgaben von Führungskräften, die primär auf eine kostengünstige Umsetzung abzielen, oder Verträgen, die kooperatives Arbeiten erschweren, einen persönlichen Austausch und Vertrauen herzustellen.

C) Für die Softwaregestaltung ist das Feedback zentral, weil es das gegenseitige Verständnis sichert. Ein großer Unterschied zwischen Scrum, IT-Projekten und den Anforderungsrunden ist, inwiefern Feedbacks Teil der Methode sind. Bei Scrum sind Feedbacks fester Bestandteil (siehe die Grundlagen 5.2.4). INTERN1 und INTERN2 zeigen eine feedback-intensive Kommunikation zwischen Anwendenden, Gestaltern und Programmierern, die sogar über das in Scrum Vorgesehene hinausgeht. Bei INTERN1 gibt es noch Resonanzgruppen, in denen Anwendende Feedback zu einer Umsetzung geben. Bei INTERN2 führt der befragte Product Owner zusätzlich noch Interviews mit Anwendenden des Fachbereichs (für Anforderungen und nach der Umsetzung). Unabhängig von der Methode sind allgemein Tests eine Möglichkeit für Feedbacks in allen Fallstudien: Softwarefirma oder IT-DL stellen eine Version der Software IT-Beratenden, Fachexpert:innen oder Anwendenden auf einem Testsystem zur Verfügung. Diese geben dann Rückmeldung darüber, ob Fehler auftreten oder was noch verbessert werden könnte.

D) In vielen Fällen ist der Ablauf in einem Lernprozess entstanden und die jeweilige Organisation entwickelt ihn weiter. Im Scrum gibt es die Rolle Scrum Master:in, die für einen reibungslosen Ablauf verantwortlich ist und versucht, den Prozess zu verbessern. Im STARTUP gibt es einen agilen Coach, der kontinuierlich den Ablauf optimiert. Anforderungsrunden wie in KOOP1 oder INTERN2 haben sich in einem Lernprozess herausgebildet und sind nun fest etabliert. Der zuständige Scrum Master in INTERN2 sieht sich als agiler Coach. Letztendlich sind im Kontext der Softwaregestaltung die Abläufe Gegenstand von Optimierungen, die aber nicht alle Fälle gleich intensiv betreiben.

Kommunikative Beziehungen – trotz Markt oder Hierarchie

Beim Vergleich der Fallstudien fällt auf, dass Rollen und Abläufe allein nicht ausreichen, um Software zu gestalten. Kommunikative Beziehungen sind notwendig, wenn es darum geht, A) Märkte und Hierarchien zu überwinden in der Kommunikation. Dann ist ein B) bestimmtes Maß an persönlicher Beziehung wichtig, um sich auszutauschen. Zuletzt ist C) eine gemeinsame Sprache nützlich, damit Beteiligte sich untereinander auf Anforderungen an die Programmierenden verständigen können.¹⁷

A) Kommunikative Beziehungen trotz Markt und Hierarchie: Bei KOOP1 und KOOP2 gibt es zusätzlich zum Vertragsverhältnis zwischen IT-DL und EVU langfristige Beziehungen. Gleichzeitig sind jene zwischen den Firmen brüchiger. Mehrere Befragte drücken eine Abwägung zwischen Alleingängen und Kooperation aus. Im Gegensatz zu KOOP2 moderiert bei KOOP1 die Konflikte zwischen den Organisationen, was gemeinsame Strategie o. ä. betrifft, ein professioneller Mediator. Im Fall PAKET existieren kooperative, langfristige Beziehungen zu EVU. Allerdings nur zu einzelnen, mit denen die SF regelmäßig in IT-Projekten oder in Arbeitskreisen zusammenarbeitet. Bei den Fallstudien INTERN1 und INTERN2 existieren persönliche Beziehungen zwischen Softwaregestaltung, Programmierenden und anwendenden Fachbereichen jenseits formaler Hierarchien und Fachbereichsgrenzen: zum einen, weil die Product Owner:innen vorher in Fachbereichen gearbeitet haben, und zum anderen, weil die Anforderungsmanagerin aus INTERN1 schon länger mit dem FB zusammenarbeitet. Bei INTERN2 sieht sich der Scrum Master als Mediator zwischen den Fachbereichen, die an der Anforderungsrunde teilnehmen. Bei KOOP3 und beim STARTUP sind Beziehungen primär kooperativ und die Zusammenarbeit nicht durch Markt und Hierarchie geprägt.

In allen Fallstudien sehen Befragte den direkten Kontakt zwischen Anwendenden, Softwaregestaltenden und Programmierenden als vorteilhaft an. Dabei erwähnen einige aus den ersten fünf Fallstudien, dass versucht wird, die direkte Kommunikation zwischen Softwareanwendung, -gestaltung und -programmierung zu unterbinden, z.B. durch Führungskräfte, Anforderungsmanagende oder vertragliche Regelungen. Die Softwarefirma von PAKET hat viele EVU als Kundschaft. Durch eine Kommunikation beschränkt auf das Ticketsystem kann sie deren vielfältige Anfragen besser verwalten.

¹⁷ Um zwischen einer gemeinsamen Sprache, einem gemeinsamen Verständnis und einer gemeinsamen Wissensbasis empirisch unterscheiden zu können, hätten noch detailliertere Fragen gestellt werden müssen. Das ist aber nicht passiert, weswegen im Weiteren keine entsprechend differenzierte Darstellung erfolgt.

Die Fallstudien lassen nicht den Schluss zu, dass der direkte Austausch innerhalb einer Organisation immer einfacher oder schwieriger wäre als zwischen verschiedenen Organisationen. Auf jeden Fall ist es einfacher im STARTUP, wo weder Führungskräfte noch Kund:innen diese erschweren, z.B. durch ein Management, welches die Kommunikation untereinander auf ein Ticketsystem beschränkt, oder eine Kundschaft, die eine funktionierende Software, aber keine aufwendigen Abstimmungen erwartet.

B) Es zeigt sich im Vergleich der Fallstudien eine enge Verbindung zwischen persönlichen Beziehungen und Kommunikation. In den Fällen sind die Beteiligten der Softwaregestaltung nicht nur unpersönliche Informationsvehikel. Sie kommunizieren ausgehend von zwischenmenschlichen Beziehungen. Diese basieren auf Langfristigkeit, Vertrauen, individuellen Kompetenzen, was Beziehungen und Kommunikation anbelangt, und weniger auf Schulungen oder Methoden.

In den Fallstudien charakterisieren die Befragten ihre Beziehungen unterschiedlich. Die verwendeten Begriffe lassen sich drei Clustern zuordnen, die Abgrenzungen zu marktförmigen und hierarchischen Beziehungen sowie die Emotionalität der Beziehungen betonen:

- Zum einen wird auf die Emotionalität und die damit zusammenhängende Kompetenz verwiesen: »Familie«, »freundschaftlich«, »miteinander sprechen«, »Konflikte persönlich klären«, »sich kümmern«, »nicht mehr verfeindet sein«, »sympathisch«, »persönliche Bindung«, »menschliche Komponente«, »sich gut kennen«.
- Es wird ein Gegensatz zum Markt betont: »keine Konkurrenz«, »Vertrauen«, »partnerschaftlich«, »transparent«, »gemeinsam«, »miteinander«, »ohne Ego-Geschichten«.
- Ein nicht-hierarchisches Verhältnis wird ausgedrückt: »auf Augenhöhe«, »offen/Offenheit«, »Geben und Nehmen«.

Obwohl Befragte eine direkte Kommunikation und Beziehungen als wichtig ansehen, gibt es keine Trainings dazu. Weder wenden die Organisationen spezielle Methoden zum Aufbau von Beziehungen an, noch bieten sie Schulungen zum partnerschaftlichen Umgang miteinander an. Nur EVU2 (KOOP2) schult in der Methode »Working Out Loud«, die beim verbalen Teilen von Wissen helfen soll. Auch die Mitarbeitenden von STARTUP haben keine Schulungen im kooperativen Handeln besucht. Ein befragter Programmierer des Start-ups meint, er ist es bereits aus seinen vorhergehenden Jobs gewohnt, so zu arbeiten.

C) Bei der Sprache und dem gemeinsamen Verständnis zeigen die Fallstudien, dass nicht jede:r jede:n immer sofort verstehen muss. Aber Wissen über den Fachbereich und sprachliches Mitteilungsvermögen, langfristige Zusammenarbeit oder ein Rollen-Tausch helfen bei der Kommunikation.

Eine gemeinsame Sprache und ein gemeinsames Verständnis sind wichtig für die Softwaregestaltung. In den Fallstudien reicht es aber aus, dies mit der Zeit zu lernen. Softwaregestaltende sind jene, die »IT-Deutsch« (Anforderungsmanagerin INTERN1) lernen müssen. Dafür müssen Key User:innen, Anforderungsmanagende, IT-Projektleitende, Anwendungsbetreuende oder Product Owner:innen aber zumindest bereit sein, es zu lernen.

Dabei ist die Bedeutung von Wissen nicht eindeutig. Zum einen ist Fachwissen nicht immer entscheidend dafür, zu verstehen, was eine andere Person sagt oder schreibt: Softwaregestaltende zwischen Anwendung und Programmierung verfügen nicht immer über Fachwissen oder lernen es, wenn notwendig, erst im Austausch und über die Zeit. Der Product Owner bei INTERN2 meint, er braucht immer weniger davon. Ein Programmierender von STARTUP sagt, er braucht das fachliche Verständnis, weil es die Kommunikation vereinfacht. Es ist aber nicht unbedingt notwendig. Zum anderen ist ein bestimmtes Maß an Wissen über die Möglichkeiten von Softwaregestaltung – ob Standard oder individuell – notwendig. Genauso erforderlich ist ein rudimentäres Wissen über den Anwendungsbereich der Software. Einzelne Product Owner:innen wie von STARTUP oder KOOP3 haben interdisziplinäres Wissen in einem Ausmaß, das es ihnen erlaubt, selbst Anforderungen zu schreiben, mit denen Programmierende etwas anfangen können. Die Ausarbeitung einer Anforderung kann dann trotzdem noch Feedbackschleifen beinhalten, weil mehrere Teilnehmende ihr Wissen einbringen. 8.4 geht ausführlicher auf das Wissen der Softwaregestaltenden ein und zeigt, dass es auch stark von der Rolle im Arbeitsprozess abhängt, welches Wissen der oder die Einzelne benötigt oder lernt. Zudem ist grundsätzlich, wenn es um Softwaregestaltung geht, die Praxis des Arbeitsprozesses wichtiger. Das liegt daran, weil es ja um das Schreiben von Anforderungen für Programmierende geht und damit darum, Wissen in Software zu übersetzen, und nicht, Wissen für sich zu behalten.

Sprachliche Fertigkeiten sind neben persönlichen Beziehungen oder interdisziplinärem Wissen wichtig, damit sich zwei Personen überhaupt austauschen. Das zeigt sich z.B. bei INTERN1. Die Anforderungsmanagerin unterbindet die direkte Kommunikation zwischen Anwendenden und Programmierenden, und zwar deshalb, weil Letztere sich nicht für die Anwendenden verständlich ausdrücken können. Eine andere Möglichkeit, eine gemeinsame Sprache zu entwickeln, ist der Rollentausch. In einigen Fällen werden Beratende genannt, die vorher programmiert haben, oder Product Owner:innen, die vorher im Fachbereich waren. Sie haben den Vorteil, dass sie die fachliche und technische Welt kennen und dadurch besser über sie sprechen können. Zuletzt erweist sich in vielen Fällen die längere Zusammenarbeit über Wissensgrenzen hinweg als hilfreich (z.B. feste Ansprechpersonen zu haben), um sich immer besser zu verstehen und eine gemeinsame Basis an Wissen und Vokabular auszubilden.

Werkzeuge - Softwaregestaltung als (teil-)integrierter digitaler Prozess

In sämtlichen Fällen gehören Ticketsysteme zum digitalen Werkzeugkasten der Softwaregestaltung. Bei dezentralen Arbeitsprozessen der Softwaregestaltung geht es in erster Linie darum, Anforderungen aufzunehmen, bei zentralisierten um Transparenz und Abstimmungen. Koordinieren und Kommunizieren steht damit im Vordergrund und weniger die direkte, zentrale Kontrolle einzelner Arbeitsschritte von Mitarbeitenden. Das zeigt sich daran, dass in einigen Fällen unterschiedliche, nicht digital integrierte Softwarelösungen nebeneinander existieren, obwohl es technisch möglich wäre, nur eine (Ticket-)Software zu verwenden oder sie alle miteinander zu verbinden. Meist werden Ticketsysteme verwendet, die aus der Softwareentwicklung/-gestaltung stammen, wie Microsoft Azure DevOps, Jira oder von SAP.

Von einer technischen Kontrolle durch ein softwarebasiertes Werkzeug kann nur insofern gesprochen werden, weil Eingabefelder und Prozesswege (der Tickets bzw. Anforderungen) durch Software vorgegeben sind und der Zugriff auf die verschiedenen Softwaresysteme über Benutzer:innenrechte technisch beschränkt ist. Allerdings entscheiden unterschiedliche Mitarbeitenden und nicht nur das Management über die Priorisierungen von Tickets, Zugang zu den Systemen oder wann der Status welcher Anforderung von »Konzeptionierung« in »Programmierung« oder »Test« wechselt. Es gibt keinen festen, maschinellen Takt. Je nach Fallstudie bestimmen vielmehr Deadlines, Reaktionszeiten (festgelegt durch SLA) oder ad hoc zu erledigende Anforderungen wie Fehler oder Tests den Arbeitsrhythmus. Das spielt vor allem in den Beziehungen zwischen Softwarefirma und EVU oder IT-DL und EVU in den Fällen KOOP1, KOOP2 und PAKET eine Rolle. Durch SLA sind Strafzahlungen festgelegt, welche bei zu langsamer Reaktion auf bestimmte Tickets fällig werden. Oft ist es auch die staatliche Regulierung, die Termine für eine Umsetzung vorgibt, oder das IT-Budget setzt ein Limit, was die aufgewendete Zeit anbelangt. Das ist aber, wie gesagt, keine technisch festgelegte Zeiteinteilung.

Der durch die Transparenz sich einstellende Panoptikum-Effekt kann nicht ausgeschlossen werden. Denn die Ticketsysteme geben z.B. Auskunft darüber, welches Ticket welchen Status hat, wer es umsetzt, wie es umgesetzt wurde. Zum einen verändert allein das Wissen darum, dass alles digital dokumentiert ist, das Handeln der Mitarbeitenden. Zum anderen lassen die softwarebasierten Werkzeuge nachträgliche Auswertungen zu. Dabei entzieht sich eine direkte, persönliche Kommunikation zwischen Mitarbeitenden nicht dieser Transparenz. Denn die Organisation erwartet von Mitarbeitenden in den Fallstudien, dass sie die Konzepte als Tickets oder in anderer Form dokumentieren. Das heißt, die informelle Kommunikation wird nachträglich formalisiert. Aber auch andere digitale Werkzeuge wie E-Mail- oder Chatprogramme dokumentieren Kommunikation. Dass das Management diese unterschiedlichen Medien zur Bewertung von Mitarbeitenden direkt nutzt, erwähnt keiner der Befragten, ebenso wenig eine individualisierte Kontrolle. Es ist anzunehmen, dass die Betriebsräte nicht nur in der Softwareanwendung die individuelle Verhaltenskontrolle unterbinden, sondern auch für die Softwaregestaltung.

Ticketsysteme sind nicht die einzigen Werkzeuge, welche den Arbeitsalltag der Softwaregestaltung prägen. Für viele EVU ist es mittlerweile Realität, dass sie drei Systeme zur Verfügung haben: eines, auf dem die aktuelle Version der Software läuft (das Produktivsystem), ein Testsystem mit zu testenden Neuerungen der Software und ein Entwicklungssystem, auf dem sie programmieren. Bei SAP R/3 und damit den ersten vier Fallstudien ist die Entwicklungs- und Testumgebung integriert, d.h. die Arbeitsprozesse von Anwendung (wenn sie testen müssen), Programmierung und Gestaltung arbeiten mit einer Softwarelösung (Tests durchführen, Analyse der Software). Sie ist die Infrastruktur, die Softwaregestaltung in den EVU ermöglicht. Bei PAKET haben die EVU eine Testumgebung, bei KOOP3 nicht, wobei es in Planung ist. Da STARTUP die Software selbst entwickelt, ist die digitale Infrastruktur zum Testen und Programmieren entsprechend vorhanden.

Softwaretechnischer Zuschnitt – zwischen individuell und Standard

Der softwaretechnische Zuschnitt entscheidet darüber, ob z.B. eine Anforderung Teil eines Standards wird und welche Priorisierung sie hat. Dieser Teil des Arbeitsprozesses der Softwaregestaltung gehört zum Kernproblem der softwaretechnischen Gestaltungsmöglichkeiten (siehe 3.2). Zwischen den Fallstudien gibt es Unterschiede, wo und wer über den Zuschnitt entscheidet, wie dadurch Synergien gehoben und wie Prioritäten gesetzt werden. Es können sich verschiedene Gruppen bei der Entscheidung gegenüberstehen.

Der individuelle Zuschnitt zeigt sich daran, ob einzelne Anwendende Anforderungen einbringen können. Ob sie das tun können, hängt in den Fallstudien vor allem davon ab, wie viele Anwendende in wie vielen unterschiedlichen EVU es gibt. Zudem zeigt sich der individuelle Zuschnitt in den Fallstudien daran, welche Perspektive primär in die Software einfließt: jene des Managements, der Branche oder einzelner Anwendender. Dazu gehört, welche Priorität die zu diesen Perspektiven gehörenden Anforderungen haben. Zuletzt gibt es Fälle, in denen die Begrifflichkeiten eines individuellen Fachbereichs in die Software einfließen – ob in deren Oberfläche oder den Quellcode.

Die Gestaltung eines Standards braucht einen aufwendigeren Koordinierungsprozess, und entweder ein IT-DL, ein EVU oder eine Softwarefirma verantworten ihn. Dabei fällt auf, dass es einen institutionellen Unterschied gibt, was den Grad der Standardisierung anbelangt: Im regulierten Netzbereich ist es einfacher, sich auf einen gemeinsamen Standard zu einigen, als im wettbewerbsorientierten und auf die Wünsche der Kundschaft ausgerichteten Vertriebsbereich. Einige Fallstudien zeigen, dass es in den EVU einen bunten Flickenteppich geben kann und es schwer herauszufinden ist, wo und wer innerhalb der Organisation einen Standard oder eine individuell gestaltete Software anwendet. Einer Softwarefirma fällt es weniger schwer, zentral über einen (Branchen-)Standard zu entscheiden und diesen durchzusetzen, als wenn mehrere EVU darüber verhandeln. Bietet STARTUP seine App auch anderen Organisationen an, wird aus einer individuellen Software ein Standardprodukt.

Letztendlich zeigen die Fallstudien, dass eine Organisation entsprechend organisiert sein muss, um die Möglichkeiten der Softwaregestaltung für Synergien zu erkennen. Ob durch eine einzelne Person oder ein Gremium: Die Organisation muss fähig dazu sein, Synergien zu erkennen. Bei KOOP1 gibt es ein Anforderungsmanagement, in dem sich verschiedene EVU darüber austauschen. Bei INTERN2 kann die Anforderungslandschaft mehrere Fachbereiche Synergien erkennen. Im EVU3 von KOOP1 ist es Aufgabe des Prozessmanagers, dies für die zwei Bereiche Privat- und Geschäftskundschaft zu tun. Aber auch Einzelpersonen achten auf Synergiepotenziale: Bei INTERN1 achtet die Anforderungsmanagerin darauf und bei KOOP1 sind die IT-Beratenden dazu angehalten.

8.4. Folgen für die Arbeit der Beschäftigtengruppe der Softwaregestaltenden

Die soziotechnische Konstellation und der Arbeitsprozess der Softwaregestaltung prägen die Arbeit der Softwaregestaltenden in den Organisationen der Fallstudien. Dabei ist es für die Beschäftigtengruppe der Softwaregestaltenden von Vorteil, wenn sie in einer reinen Netzwerkorganisation arbeiten und nicht in einer Matrixorganisation. Das ist

unabhängig davon, ob sie eine Individual- oder eine Standardsoftware gestalten. Warum das so ist, zeigt der Abschnitt mithilfe dreier Kategorien:

- Beschäftigungssystem: Softwaregestaltende zeichnen sich weniger durch eine hierarchische Karriere aus, sondern eher durch eine Karriere, in der es darum geht, Kompetenzen, Erfahrungen und interessante Projekte zu sammeln. Außerdem sind Softwaregestaltende in der Regel Akademiker:innen und flexibler in ihrer Beschäftigung (häufigere Wechsel von Organisationen, Projekten usw.).
- Kontrolle: Softwaregestaltende arbeiten eigenständig, gemäß ihrer Rolle entlang eines softwarezentrierten Arbeitsprozesses. Wobei die Grundkoordination der soziotechnischen Konstellation sich darauf auswirkt, wie stark Hierarchie oder Märkte die Arbeit der Softwaregestaltenden mitkontrollieren.
- Wissensverteilung: Es existiert eine Praxisgemeinschaft der Softwaregestaltenden, der es nicht darum geht, möglichst viel Wissen anzusammeln und zu konzentrieren. Sie will Wissen in Quellcode übersetzen. Nicht das Wissen der Softwaregestaltenden allein nimmt zu, sondern vor allem das in Software materialisierte und in verschiedenen digitalen Quellen hinterlegte Wissen (z.B. Quellcode, Ticketsystem, Dokumentation). So zeichnen sich Softwaregestaltende primär dadurch aus, zu wissen, wie Softwaregestaltung funktioniert. Auch wenn Softwaregestaltende interdisziplinär arbeiten: Die Fallstudien zeigen nicht, dass energiewirtschaftliches oder softwaretechnisches Wissen eine notwendige Bedingung für ihre Arbeit ist. Es ist hilfreich. Aber ob es notwendig ist und wie viel interdisziplinäres Wissen sie brauchen, ergibt sich aus der jeweiligen Praxis im Arbeitsprozess der Softwaregestaltung.

Der Abschnitt geht zum Schluss in der Zusammenfassung auf die Unterschiede zu den anderen zwei Gruppen der Anwendenden und Programmierenden ein. Der Vergleich hilft, die Besonderheiten der Arbeit der Softwaregestaltenden herauszuarbeiten. Zuerst stellt der nächste Punkt die unterschiedlichen Typen von Bedingungen vor, unter denen die Softwaregestaltenden arbeiten, um danach die Fälle darzustellen.

8.4.1. Softwaregestaltende: zwischen Matrix- und reiner Netzwerkorganisation

Für die Arbeit der Softwaregestaltenden lassen sich die Fälle grundsätzlich darin unterscheiden, ob sie aufgrund der soziotechnischen Konstellation und des Arbeitsprozesses der Softwaregestaltung in einer reinen Netzwerkorganisation arbeiten oder in einer Matrixorganisation.

Bei einer **reinen Netzwerkorganisation** findet die Softwaregestaltung im Rahmen der Grundkoordination Netzwerk statt. Die Kontrolle zeichnet sich durch eine Peer- und Objekt-Kontrolle aus (siehe 6.4.2.1) mit dem Schwerpunkt darauf, sich abzustimmen. Weil die formalen Hierarchien flach sind und für die Softwaregestaltenden eine Karriere in der Hierarchie keine Priorität hat, gibt es Kompetenzkarrieren. Die Gruppen von Softwaregestaltenden, Softwareanwendenden und Programmierenden unterscheiden sich primär soziotechnisch aufgrund ihres Verhältnisses zur Software und weniger durch rein soziale Hierarchien. Das heißt, sie sind nicht nur organisational getrennt bspw. durch verschiedene Rollen oder Teams, sondern durch unterschiedliche Zugän-

ge und Aufgaben, was die Software anbelangt. Das Wissen der Softwaregestaltung für den Arbeitsprozess der Softwaregestaltung ist durch Märkte oder Hierarchien nicht getrennt. Die Softwaregestaltenden können sich auf das Übersetzen zwischen Anwendung und Programmieren konzentrieren: Sie können sich das interdisziplinäre Wissen in der Praxis allein oder im Gestaltungsnetzwerk aneignen, weil das Wissen zugänglich ist und sie nicht nur temporär Teil des Gestaltungsnetzwerks sind. Darüber hinaus besteht eine gemeinsame Wissensbasis, z.B. weil die Programmierenden selbst über ausreichend interdisziplinäres Wissen für die Zusammenarbeit verfügen.

In einer **Matrixorganisation** findet die Softwaregestaltung im Rahmen der Grundkoordinationen Markt oder Hierarchie statt. Es besteht eine Spannung einerseits aus der Kontrolle durch Führungskräfte in einer Hierarchie oder Kundschaft und Verträge in einem Markt und andererseits einer reinen Koordination bzw. einem offenen Wissensaustausch. Weil Hierarchien oder Märkte zusätzlich zu einem Netzwerk der Softwaregestaltung existieren (z.B. bei IT-Projekten), können die Softwaregestaltenden sowohl eine Karriere in der Hierarchie als auch im Netzwerk machen. Wobei es dann eine Herausforderung für die Organisationen darstellt, beidem gerecht zu werden. Die Gruppen von Softwaregestaltenden, Softwareanwendenden und Programmierenden sind durch Hierarchien und Märkte getrennt. Ebenso verteilt sich das für die Softwaregestaltung benötigte Wissen auf Märkte und Hierarchien. Durch die Matrixorganisation ist die interdisziplinäre Zusammenarbeit nicht immer eingespielt und oftmals nur temporär. Das hat zur Folge, dass die Softwaregestaltenden eine interdisziplinäre Wissensbasis mitbringen oder diese gemeinsam mit Anwendenden und Programmierenden erarbeiten müssen.

Tabelle 23: Idealtypen Matrix- und reine Netzwerkorganisation

Typ	Kontrolle	Beschäftigungssystem	Wissensverteilung
Reines NW	Peer- und Objekt-Kontrolle, Fokus auf Koordination	keine Karriereleiter, Aufgaben und Position abhängig von Stellung zur Software	Wissen rein horizontal im Gestaltungsnetzwerk verteilt
Matrix	Mischung mit Markt und Hierarchie	Teil von Märkten oder Hierarchien	Wissen getrennt durch Hierarchien oder Märkte

Sind nun die einzelnen Fälle Beispiele dafür, dass Softwaregestaltung in einer reinen Netzwerk- oder in einer Matrixorganisation stattfindet? Antwort darauf geben die Falldarstellungen und die Zusammenfassung am Schluss dieses Abschnitts.

8.4.2. Darstellung der Fallstudien

Die Falldarstellung der Arbeit der Softwaregestaltenden gliedert sich in die Kategorien Beschäftigungssystem, Kontrolle und Wissensverteilung.

8.4.2.1. INTERN1: Softwaregestaltende zwischen Fachbereich und IT-Abteilung, Matrix

In diesem Fall, in dem ein EVU dezentral eine individuelle Software gestaltet, finden die Softwaregestaltenden in der Matrixorganisation förderliche Bedingungen für ihre Arbeit vor. Die Softwaregestaltenden sind zwar in die Hierarchie der Softwareanwendung eingebunden und das Management gibt den Rahmen vor. Allerdings lässt es die Softwaregestaltenden weitestgehend eigenständig arbeiten. Ob sie eigene Karrierewege verfolgen können und immer zu ihrer Expertise passende Aufgaben bekommen, ist noch offen. Wie für eine Matrixorganisation typisch, arbeiten Softwaregestaltende temporär in Projekten und die Karrierewege entsprechen einem EVU, das primär auf die Softwareanwendung ausgerichtet ist. Dafür konnte das EVU trotz Matrixorganisation dafür sorgen, dass die Softwaregestaltenden sich auf die Übersetzungsarbeit von fachlichen Bedarfen und Anforderungen konzentrieren und sich in der Praxis interdisziplinäres Wissen aneignen, wobei sie die Abteilungsgrenzen dabei nicht behindern. Dafür ist hilfreich, dass die Softwaregestaltenden und Programmierenden selbst über energiewirtschaftliches Wissen verfügen und längerfristig zusammenarbeiten.

Das **Beschäftigungssystem** richtet sich insofern an die Softwaregestaltenden aus, als sie nicht in einer Abteilung feststecken. Sie sind nur temporär zu Softwaregestaltungsprojekten wie der mobilen App für Monteur:innen zugewiesen oder haben parallel noch andere Projekte. Der befragte Teamleiter aus der IT gesteht, dass sie überlegen, wie die Softwaregestaltenden zwischen verschiedenen Projekten wechseln können, damit den intrinsisch motivierten Leuten nicht langweilig wird. Dazu würde gehören, die Karrierewege anzupassen. Eine Karriere in der Hierarchie interessiert die befragte Anforderungsmanagerin nicht. Sie will keine Führungskraft werden, sondern ist inhaltlich motiviert.

»Wir arbeiten ja, wenn wir was Größeres haben, in einer Projektstruktur und dann kann man da für das Projekt halt Projektleiter werden, Projektverantwortliche. Sowas könnte ich mir eher vorstellen, weil es eine inhaltliche Arbeit ist. [...] Disziplinarisch, das nächste, was ich machen könnte, wäre Teamleiter. Hätte ich gar keine Lust drauf.« (Anforderungsmanagerin)

In der Matrixorganisation arbeiten die Softwaregestaltenden eigenständig. Bei der **Kontrolle** legt die Hierarchie bzw. das Management weder konkrete einzelne Arbeitsschritte fest noch die Arbeitsgeschwindigkeit. Das zeigt sich daran, dass sie die digitalen Werkzeuge vor allem zur Koordination und Kommunikation verwenden. Das Management entscheidet über die Ressourcen wie IT-Budget und Stellenvergabe und welche Softwaregestaltungsprojekte das EVU macht. Die Führungskräfte kontrollieren nur die Ergebnisse oder wenn etwas im Prozessablauf auffällt:

»Also, ich glaube, solange ich liefere, interessiert es keinen, wann ich was mach. Also, ich fühle mich nicht kontrolliert, nee.« (Anforderungsmanagerin)

Das führt dazu, dass die Softwaregestaltung als eigenständige Arbeit erlebt wird und einer persönlichen Identifikation mit der Arbeit Raum lässt:

»Also das, was ich mache, ist eigentlich Handwerk und das ist das Schöne daran. Ich habe ein Ergebnis.« (Anforderungsmanagerin)

Nur weil die Softwaregestaltung Teil der Hierarchie eines EVU ist, konzentriert sich das **Wissen** über sie nicht beim Management. Vielmehr verteilt sich das Wissen auf mehrere Abteilungen und auf jene, die (temporär) an ihr mitwirken. Dabei kann in dem Fall trotz der Matrixorganisation das EVU einen Arbeitsprozess der Softwaregestaltung etablieren, der es den Softwaregestaltenden erlaubt, sich auf die Übersetzung zwischen Anwendung und Entwicklung zu konzentrieren, d.h. darauf, Anforderungen zu sammeln und zu schreiben. Dabei kooperieren die Anwendenden, auch wenn die Softwaregestaltenden ihnen nicht disziplinarisch vorgesetzt sind, und die Softwaregestaltenden sammeln interdisziplinäres Wissen an:

»Also, ich kannte die agile Arbeitsweise und die ganzen Methoden nicht. Das habe ich alles gelernt. Ich musste sehr viel SAP lernen. [...] Also, ich könnt immer noch nichts entwickeln. Ich bin kein Entwickler. Aber wenn ich einen Fehler sehe, weiß ich, in welche Ecke ich den wahrscheinlich schieben muss. Wen ich brauche, welchen Entwickler: Frontend, Backend und welche Richtung dort.« (Anforderungsmanagerin)

Dazu gehört, dass die anderen Beteiligten auch über interdisziplinäres Wissen verfügen: Die vier Product Owner:innen waren vorher im Fachbereich und die Programmierenden arbeiten bereits länger für den Fachbereich.

8.4.2.2. INTERN2: Softwaregestaltende in mehreren Fachbereichen, Matrix

Wie bei INTERN1 sind die Arbeitsbedingungen für die Softwaregestaltenden im EVU in der Matrixorganisation insgesamt förderlich. Zwar gibt es einerseits in dem Fall bereits teilweise eigene Karrierewege für Softwaregestaltende. Andererseits nennen Befragte das Management explizit als hinderlich für die eigene Arbeit. Denn es verfolgt keine abteilungsbürgreifende Perspektive, was für die Softwaregestaltenden hilfreich wäre. Dafür können die Softwaregestaltenden ungehindert von Abteilungsgrenzen im horizontal verteilten Gestaltungsnetzwerk zwischen den Fachbereichen das notwendige Wissen sammeln und in Anforderungen übersetzen.

Was das **Beschäftigungssystem** anbelangt, fällt in dieser Fallstudie auf, dass es im Gegensatz zu INTERN1 teilweise neue Karrierewege gibt und gleichzeitig alte Muster fortbestehen. Denn das EVU bietet einzelnen Softwaregestaltenden bereits die Möglichkeit, eine für sie reizvolle Karriere zu verfolgen. Derzeit diskutiert das EVU, wie Scrum Master:innen oder Product Owner:innen aufsteigen können. In der Zentralbereichs-IT gibt es bereits die Möglichkeit, eine Karriere als Fachexpert:in ohne personelle Verantwortung zu machen und auf diesem Wege in eine höhere Vergütungsgruppe zu kommen. Andererseits hat der befragte Product Owner aus der Fachbereichs-IT seine Rolle schon länger inne und ist in sie hineingewachsen, ohne dass sich dies in der formalen Organisation niederschlägt (z.B. in einer neuen Stellenbezeichnung). Er hat nicht die althergebrachte Karriere gemacht vom Teammitglied zum Teamleiter. Der Scrum Master strebt selbst als nächsten Karriereschritt die Leitung eines internen Teams inklusive

Personalverantwortung an. Für ihn ist eine Karriere im Netzwerk, d.h. sich fachlich weiterzuentwickeln, nur zweite Wahl.

Obwohl wie bei INTERN1 die Softwaregestaltung Teil einer Hierarchie ist, sind die Softwaregestaltenden keiner direkten **Kontrolle** der Führungskräfte ausgesetzt und arbeiten selbstständig. Wie auch bei INTERN1 steht die Ergebniskontrolle im Vordergrund – ob beim Product Owner, Scrum Master oder der Anforderungsmanagerin. Inhaltlich spielt die Führungskraft des Scrum Masters keine Rolle. Er setzt aber Ziele fest.

»Und sorgt eigentlich dafür, dass es mir gut geht, sage ich mal so.« (Scrum Master)

Teilweise verhindern aber bestimmte Führungskräfte z.B. aus dem Fachbereich einen offenen Austausch. Übergreifende Entscheidungen innerhalb eines Fachbereichs oder zwischen den Fachbereichen fallen schwer. Sie müssen sich z.B. darüber einigen, wie die Anforderungsrede abläuft oder wie sich das IT-Budget verteilt.

Was das **Wissen** anbelangt, ist für die softwaretechnische Interdisziplinarität die Matrixorganisation ein Nachteil. Denn in diesem Fall verteilt sich das Wissen für Softwareanwendung, -gestaltung und -programmierung auf mehrere Fachbereiche und die Softwaregestaltenden müssen in einer Organisation agieren, die primär auf die Softwareanwendung ausgerichtet ist, und entsprechend haben u.a. die Führungskräfte kein Softwaregestaltungswissen. Doch wie auch bei INTERN1 können die Softwaregestaltenden durch den kontinuierlichen Austausch im Gestaltungsnetzwerk Wissen austauschen – fast wie in einem reinen Netzwerk. Weder Programmierende noch Anwendende halten ihr Wissen zurück. So lernen die Beteiligten voneinander und alle bauen mit der Zeit interdisziplinäres Wissen auf. Dabei sagen Scrum Master, Product Owner und Anforderungsmanagerin gleichermaßen, dass für ihre Arbeit das Wissen, wie Software gestaltet wird, wichtiger ist als Fach- oder ERP-Wissen. Die Anforderungsmanagerin meint allerdings, dass man ohne energiewirtschaftliches Wissen langsamer ist und vieles nicht sofort versteht. Das Wissen lernt sie jetzt mit der Zeit in der Praxis und durch einen Kollegen, der mehr Erfahrung hat.

Die Softwaregestaltenden kompensieren untereinander die Nachteile einer Organisation, die primär auf die Softwareanwendung ausgerichtet ist, indem sie sich EVU-weit vernetzen. Der Product Owner aus dem Fachbereich tauscht sich in einem Expert:innenkreis mit Mitarbeitenden anderer Netzbetriebe aus, die eine ähnliche Tätigkeit wie er haben. Im IT-Bereich gibt es einen firmenweiten, organisierten Austausch für Scrum Master:innen und sogenannte Gilden für Software-Architekt:innen.

8.4.2.3. KOOP1: IT-DL als Heimat der Softwaregestaltenden, Matrix

Im Vergleich zu INTERN1 und INTERN2 ist es für die Arbeit der Softwaregestaltenden von Vorteil, dass das IT-DL auf die Softwaregestaltung spezialisiert ist. Entsprechend bietet das IT-DL die gewünschten Karrierewege, kann für die Auslastung der Softwaregestaltenden sorgen und eine stetige softwaretechnische Interdisziplinarität garantieren. In dem Fall zeigt sich, dass für die Softwaregestaltenden ein Arbeitsmarkt existiert, den sie für eine fachliche Karriere unabhängig von internen Hierarchien nutzen können. Der Nachteil ist in diesem Fall die Marktbeziehung zwischen IT-DL und EVU. Das führt dazu, dass die Kundschaft (EVU) des IT-DL die Arbeit der Softwaregestaltenden mitkon-

trolliert. Zudem führt es zu einer Abhängigkeit, was Wissen anbelangt: Einerseits fehlt den meisten EVU das Wissen, um selbst die Möglichkeiten der Softwaregestaltung einschätzen zu können. Andererseits haben jene EVU, die selbst intern Softwaregestaltende beschäftigen und mit dem IT-DL langfristig zusammenarbeiten, eine Ansprechperson, mit der sie eine gemeinsame Wissensbasis teilen.

Was das **Beschäftigungssystem** anbelangt, zeigt sich in dem Fall statt einer fachlichen Karriere innerhalb einer Hierarchie, dass für Softwaregestaltende eine fachliche Karriere im Markt möglich ist. Das IT-DL ist Teil dieses Marktes für IT-Fachkräfte. Es gibt Mitarbeitende, für die ist das IT-DL nur eine Durchgangsstation in ihrem beruflichen Werdegang. Vor allem die Spezialisierung auf die ERP-Software von SAP bietet die Möglichkeit, sich für eine Vielzahl anderer Firmen zu qualifizieren. Für das IT-DL bedeutet das laut einem Befragten eine stetige Personalfluktuation, was aus seiner Sicht neue Impulse bringt und dafür sorgt, dass das IT-DL nicht überaltert.

Für die Softwaregestaltenden ist auch eine Karriere innerhalb des IT-DL möglich. Zudem ist für sie der Vorteil einer zentralisierten Softwaregestaltung in einem IT-DL, dass sie für mehrere EVU tätig werden können. Dadurch ist für sie in diesem Fall einfacher möglich, was sie sich auch schon bei INTERN1 und INTERN2 gewünscht haben: interessante Aufgaben und Projekte zu bearbeiten. Vor allem für junge Mitarbeitende sind innovative Projekte ein stärkerer Anreiz als der Aufstieg zur Teamleitung. Der Anforderungsmanager des IT-DL meinte, dass ihn die Aufgabe motiviert, nicht die Aussicht auf eine Karriere. Auch in den EVU haben die Softwaregestaltenden solche Wünsche. In EVU2 gibt es bereits zwei Karrierepfade: eine Führungskraft-Karriere und Fachkarriere. Im EVU3 spricht der Prozessmanager davon, dass er eine fachliche Karriere und keine disziplinarische machen will.

Inwiefern ist die Matrixorganisation für die **Kontrolle** der Softwaregestaltenden wichtig oder stört sie sogar bei ihrer Arbeit? Es überrascht, dass auch in diesem Fall, wie schon bei INTERN1 und INTERN2, trotz Markt und Hierarchien die Führungskräfte sowohl bei den EVU als auch dem IT-DL nur wenn notwendig direkt kontrollieren. Auch wenn die Befragten aus den EVU alle unterschiedliche Positionen haben und daher unterschiedliche Führungskräfte, berichten alle davon (ob Anwendungsbetreuer, Anforderungsmanager oder Prozessmanager), dass sie wenig bis nur in Ausnahmefällen Kontakt mit ihrer Führungskraft haben. Sie können weitestgehend selbstständig arbeiten. Die Selbständigkeit zeigt sich auch an der Arbeitsbelastung und den Zielen. Die Arbeitsbelastung ist individuell unterschiedlich und hängt von Vorgaben des Managements ab. Ein anderer Befragter meint, er kommt durch Selbstorganisation gut hin und hat keinen Zeitdruck.

»Es ist so bei uns, dass wir unsere Arbeitsbelastung gut selbst managen können.« (Applikationsbetreuer EVU2)

Von welchen anderen Umständen die Arbeitsbelastung abhängt, ist unklar. Beide befragten Prozessmanager (EVU3, EVU1) sprechen von einer hohen Arbeitsbelastung. Beim IT-DL müssen zwar die Softwaregestaltenden bei mehreren Projekten oder Tickets für mehrere EVU mitarbeiten. Jedoch hat diese keine direkten Folgen für die Arbeitsbelastung des einzelnen Beschäftigten. Das liegt z.B. daran, dass das Projektmanagement-

Team des IT-DL Projektarbeit selbstorganisiert verteilt. Jede:r muss sich selbst fragen, was er oder sie noch schafft.

Auch vorgegebene Ziele führen nicht automatisch dazu, dass Führungskräfte das Handeln der Beschäftigten einschränken. Der IT-Projektleiter des EVU sieht sich durch das IT-Budget wenig eingeschränkt. Auch in einem anderen EVU gibt es »relativ viel« (Prozessmanager EVU3) Budget und es wird zur Verfügung gestellt und spielt keine große Rolle. Der Applikationsbetreuer (EVU2) schließt die Zielvereinbarung jährlich mit seiner Führungskraft ab (woran auch sein Gehalt hängt). Auch der Prozessmanager (EVU1) hat individuelle Ziele, die aber seiner Meinung nach nicht so wichtig sind für sein Gehalt. Beim IT-DL erarbeiten Führungskräfte ergänzend zum Arbeitsvertrag mit den einzelnen Angestellten Ziele im Dialog. Das Unternehmen selbst hat Ziele, was Umsatz, Projekte und Qualität anbelangt. Diese spielen aber laut den Befragten bei der Arbeitsbelastung keine Rolle. Der Anforderungsmanager des IT-DL setzt sich die intrinsischen Ziele selbstbestimmt, wie z.B. defizitäre Projekte zu vermeiden oder besser zu werden. Die Ziele der Firma sind für ihn sekundär. Dazu kommt, dass der Bonus, den das IT-DL ausschüttet, für alle Mitarbeitenden gleich ist.

Was die **Wissensverteilung** anbelangt, können sich die Softwaregestaltenden trotz Hierarchien innerhalb der EVU austauschen und trotz Marktbeziehungen existieren längerfristige Beziehungen zwischen EVU und IT-DL, wodurch eine gemeinsame Wissensbasis nicht nur temporär besteht. Durch das vom IT-DL kooperativ organisierte Anforderungsmanagement bestehen Gestaltungsnetzwerke in die EVU hinein und damit eine gemeinsame Wissensbasis bei den Beteiligten. Allerdings ist die Beziehung nicht gesichert, denn das IT-DL steht im Wettbewerb mit anderen. Wenn die Wissensbasis fehlt, fällt das schnell auf, so z.B., wenn Mitarbeitende des IT-DL noch nicht so lange für die Energiewirtschaft tätig sind:

»[E]s gibt da teilweise Kollegen, die da neu sind bei dem Dienstleister und die machen da irgendwas, was nicht richtig ist, weil die aber einfach die Gegebenheiten, diese stadtwerkespezifischen Gegebenheiten nicht kennen« (Anforderungsmanager EVU2).

Manche EVU sind abhängig vom IT-DL, weil sie selbst kein Know-how in der Softwaregestaltung haben. Manche bauen erst wieder Kompetenzen auf oder sammeln durch eigene Softwaregestaltung interdisziplinäres Wissen. Es gibt in den EVU Mitarbeitende wie Prozessmanager, die sowohl fachliches als auch softwaretechnisches Wissen mitbringen. Die befragte Prozessmanagerin des EVU1 schätzt bei ihr das Verhältnis von IT- zu Fachwissen mit 30:70 ein. Der Vorteil ist in diesem Fall, dass innerhalb des IT-DL die Interdisziplinarität tägliche Praxis ist. Die befragten Programmierenden sehen sich weniger als Teil ihrer Programmierenden-Teams als vielmehr von interdisziplinären Teams, die sich für Projekte oder einzelne Anforderungen zu spezifischen Fachthemen zusammenfinden. Es gibt IT-Beratende und Programmierende, die nur für ein EVU arbeiten und sich dadurch besser mit dessen System auskennen. In einem Fall ist der Ansprechpartner des IT-DL Programmierer und Berater zugleich, der bei der Einführung der Lösung und an ihrer individuellen Anpassung beteiligt war. Er verfügt damit über ein umfassendes fachliches und softwaretechnisches Wissen.

Schulungen dienen als Einstieg. Die einzelnen Mitarbeitenden bauen ihr Wissen über die Jahre auf und verfügen über individuelle Wissensbestände und Spezialisierung – je nachdem, an welchen Projekten sie teilgenommen haben und in welchem Fachbereich sie tätig sind.

8.4.2.4. KOOP2: Softwaregestaltende in fremdem Umfeld der EVU, Matrix

Wie im Fall von KOOP1 hat hier das IT-DL den Vorteil, sich auf die Softwaregestaltung spezialisieren zu können. Entsprechend finden die Softwaregestaltenden dort andere Karrierewege, eine von der Softwareanwendung unabhängige Kontrolle und eine Spezialisierung auf interdisziplinäres Wissen vor. Jedoch haben einzelne EVU in diesem Fall die Softwaregestaltung wieder verstärkt selbst übernommen. Diese EVU können weder passende Karrierewege noch eine stetige Auslastung oder immer interessante Projekte für die Softwaregestaltenden gewährleisten. Stärker als in anderen Fällen zeigt sich das Problem von einer Wissensverteilung in Hierarchien und Marktbeziehungen. Die Gestaltungsnetzwerke der Softwaregestaltenden sind prekär und so fehlt manchmal die gemeinsame Wissensbasis. Dafür sind die Softwaregestaltenden in den EVU unabhängig vom IT-DL, was das notwendige Wissen zur Softwaregestaltung anbelangt.

Das Besondere an dem Fall ist, dass aufgrund der ursprünglichen Spezialisierung des IT-DL auf die Softwaregestaltung dieser sich hinsichtlich **Beschäftigungssystem** auf diese ausgerichtet hat. Ähnlich wie in der Fallstudie KOOP1 nimmt nach Aussage des befragten Bereichsleiters im IT-DL die Bedeutung von Karriereleitern ab, und die Mitarbeitenden wollen weniger Führungskraft werden als vielmehr interessante Projekte machen. Das Unternehmen versucht seit mehreren Jahren neben der Führungslaufbahn neue Karrieremodelle zu entwickeln. Die Idee einer Fachexpert:innen- und Projektmanagementlaufbahn hat das IT-DL aufgegeben. Es gibt nun Kompetenzlevel mit entsprechenden Gehaltsbändern, bei denen Mitarbeitende durch Erfüllen bestimmter Anforderungen fachlich und nicht hierarchisch aufsteigen können. Was den Wechsel zwischen Organisationen anbelangt: Wie auch bei KOOP1 wechseln IT-Fachkräfte wie IT-Beratende und Programmierende eher zu Wettbewerbern und nicht innerhalb des organisationalen Netzwerks aus EVU und IT-DL. Die fachliche Karriere im Markt – inklusive der Neuverhandlung des Gehalts – ist damit wie schon bei KOOP1 eine Alternative zu jener innerhalb einer Organisation.

Ein EVU2 der Fallstudie hat zwar wieder die Softwaregestaltung übernommen. Aber es zeigt sich, dass es dabei zu Konflikten mit den bestehenden Beschäftigungssystemen kommt. Denn dort dominiert die Karriere in der Hierarchie. Die Befragten – IT-Koordinator und Projekt Coach – haben das EVU2 einige Monate nach dem Interview verlassen. Das lag nicht daran, dass sie keine hierarchischen Karrieren machen können. Vielmehr gründet es in der fehlenden Perspektive, dass das EVU sich zu einer agileren Organisation mit flachen Hierarchien und interessanten Projekten wandeln würde.

Was die **Kontrolle** der Arbeit der Softwaregestaltenden anbelangt, gibt es Unterschiede zwischen IT-DL und EVU. Das Beispiel eines IT-Beratenden beim IT-DL zeigt exemplarisch, wie sich selbstständiges Arbeiten mit Markt und Hierarchien mischt. Der befragte Teamleiter als Führungskraft des IT-Beratenden kümmert sich um Budget, Verträge, neue Projekte, Entscheidungen zu fachübergreifenden Themen und Personalführung. Er fordert eigenständiges Arbeiten ein: stärkere Lernbereitschaft

und Eigeninitiative. Beides sieht er als festen Bestandteil der Arbeit an. Mitarbeitende sind gefragt, die bereit sind, sich auch über eigene Themengebiete hinaus jenseits von Schulungen selbstständig in neue Themen einzuarbeiten. Sie sollen Unternehmer im Unternehmer werden:

»Habe es aber geschafft, meine Philosophie ein bisschen durchzusetzen. Also, das, was ich bei [große Beratungsfirma] erfahren habe, gelernt habe: dass der Berater der Unternehmen ein Unternehmer im Unternehmen ist und für sich selbst verantwortlich ist und man ihm durchaus genügend Verantwortung und Kompetenz geben kann, auch beim Kunden vor Ort entsprechend seine Aufgaben zu übernehmen. Das hat in den letzten Jahren eigentlich ganz gut gefruchtet.« (Teamleiter IT-DL)

Der IT-Berater gibt an, wenig Kontakt zu seiner Führungskraft zu haben, und er arbeitet die Aufgaben ihrer Priorisierung gemäß in seinem eigenen Tempo der Reihe nach ab (ohne feste Vorgaben für die Anzahl der umgesetzten Anforderungen o. ä.). Jedoch spielen neben der Führungskraft für die Mitarbeitenden des IT-DL Markt-Kennzahlen eine Rolle. Umsatzziele fließen in die Faktura-Ziele der Beratenden ein, d.h., sie müssen eine bestimmte Anzahl an Tagen bei einem EVU sein und damit Umsatz für das IT-DL generieren.

Innerhalb der EVU nehmen einzelne Befragte ihre Führungskräfte als hinderlich oder gar überflüssig für die Softwaregestaltung wahr. Der befragte IT-Koordinator stellt seine Führungskraft in Frage, da er alles eigenständig erledigt. Andere Führungskräfte entscheiden über das Vorgehen bei der Softwareanpassung. Was an dem Fall auffällt: In einer Matrixorganisation hängen die Softwaregestaltenden davon ab, dass Führungskräfte sie unterstützen. Führungskräfte koordinieren Projekte, sprechen bei Problemen mit dem IT-DL. Sie fungieren als Netzwerkende, die Beteiligte eines Projekts kennen und helfen, sich innerhalb der Firma durchzusetzen. Sie werden bei Eskalationen aktiv. Wie auch schon in anderen Fallstudien haben Softwaregestaltende Freiheiten bei der Zielvereinbarung. Der IT-Koordinator aus EVU2 verhandelt mit seiner Führungskraft, was die Ziele sind:

»Ich sag auch meinem Chef ganz klar: ›Ja, kann ich noch machen.‹ Oder: ›Nee, das passt jetzt nicht, weil, ich möchte noch die anderen Themen auch sauber zu Ende bringen.‹ Und das ist auch etwas, was ich halt lernen habe müssen. Und ich glaube, das muss jeder irgendwann lernen, seine eigenen Grenzen und halt eben auch zu sagen: Was ist mir wichtig? Ist mir wichtig, nur Quantität rauszuballern? Und mir ist Qualität ganz wichtig.« (IT-Koordinator Fachbereich EVU2)

Die Matrixorganisation zeigt sich in der **Wissensverteilung** vor allem innerhalb der EVU. Dort besteht eine starke Trennung zur Softwareanwendung und -programmierung. Erst mit der Zeit, oftmals temporär und abhängig von Einzelpersonen oder einzelnen Fachbereichen, entstehen a) Gestaltungsnetzwerke zwischen Softwareanwendung, -gestaltung und -programmierung und b) entwickeln Führungskräfte zumindest ein grundlegendes Verständnis für die Softwareentwicklung.

Bei einer auf die Softwareanwendung ausgerichteten Organisation wie EVU2 ist erst einmal keine gemeinsame, interdisziplinäre Wissensbasis für die Softwaregestaltung vorhanden. Das zeigt sich an Führungskräften, die kein Wissen über die Softwaregestaltung haben, aber haben sollten, weil sie in Matrixorganisationen über den Wissensaustausch entscheiden können. Das zeigt sich an einem Softwaregestaltungsprojekt in EVU2, bei dem es an einer gemeinsamen, interdisziplinären Wissensbasis fehlt, z.B. bei der Zusammenarbeit zwischen IT-Abteilung und Fachbereichen. Es sind neue, für die Softwaregestaltung geschaffene Rollen wie die IT-Koordinierenden der Fachbereiche, die sich teamübergreifende Kompetenzen aneignen können. Sie sind für die Softwaregestaltung in mehreren Teams zuständig und können so interdisziplinäre Zusammenhänge verstehen lernen. Im Team Marktkommunikation des EVU2 ist die Zusammenarbeit mit dem zuliefernden Softwareunternehmen eingespielt und die Teamleitung und einzelne ihrer Mitarbeitenden haben das entsprechende Wissen zur Softwaregestaltung. Hier zeigt sich, dass es in einer Matrixorganisation von Vorteil ist, wenn die Führungskraft selbst mitgestaltet oder weiß, wie Softwaregestaltung funktioniert.

EVU3 hat eine andere Variante, um in einer Matrixorganisation für Softwaregestaltende eine Wissensbasis zu schaffen, damit sie mit Anwendenden und Programmierenden zusammenarbeiten können: Das EVU hat zentral ein Team etabliert, das selbst Wissen durch Softwaregestaltungsprojekte in unterschiedlichen Fachbereichen aufbaut. Zugleich sorgt es dafür, dass die Matrixorganisation interdisziplinäres Gestaltungswissen in und zwischen den Fachbereichen aufbaut.

Eine Basis interdisziplinären Wissens ist fest in der Hierarchie des IT-DL verankert. Der befragte Teamleiter war selber einmal IT-Berater und hat fachlich und softwaretechnisch tiefergehendes Wissen. Der IT-Berater kann programmieren und kennt sich in einzelnen Prozessen der Energiewirtschaft aus. Insgesamt hat das IT-DL sowohl softwaretechnisches als auch fachliches Wissen. Es kennt die ERP-Firma SAP und deren Software sowie die EVU sehr gut. Zudem hat es aufgrund der BPO-Dienstleistungen selbst Anwendende der Software im Haus.

8.4.2.5. PAKET: Softwaregestaltende einer Standardsoftware, Matrix

In diesem Fall, in dem eine Softwarefirma zentral eine Standardsoftware gestaltet, finden die Softwaregestaltenden nur bedingt förderliche Bedingungen für ihre Arbeit vor. Die Softwaregestaltenden innerhalb der Softwarefirma sind getrennt von den vielen Anwendenden in den EVU. Die Softwaregestaltenden in den EVU, welche Einstellungen am Standard vornehmen, sind vom Standard abhängig, den die Softwarefirma ausliefert. Aber ob innerhalb der Softwarefirma oder der EVU: Die Softwaregestaltenden sind größtenteils fest einem fachlich spezialisierten Bereich wie Energiedatenmanagement oder Marktkommunikation zugeordnet und arbeiten entsprechend kontinuierlich interdisziplinär. Anders als für SAP gibt es in diesem Fall keinen eigenen Arbeitsmarkt für IT-Beratende der industriespezifischen ERP-Software und damit weniger Karrieremöglichkeiten für die Softwaregestaltenden. Innerhalb der EVU sind Softwaregestaltende höhergestellt als die Anwendenden, aber Führungskräften untergeordnet. In dem Fall fällt stärker auf als bei KOOP1 und KOOP2, dass aufgrund der Marktbeziehung die Wissensverteilung zwischen EVU und IT-DL zu einer Abhängigkeit führt. Die EVU müssen sich eigenständig über Neuerungen an der Software nach Updates und über die vorhandenen

Einstellungsmöglichkeiten am Standard informieren. Der Wissensaustausch zwischen EVU und Softwarefirma ist weniger problematisch, wenn beide aus einer Kooperation in puncto Softwaregestaltung Vorteil ziehen. So geschieht dies z.B., wenn ausgewählte EVU für die Gestaltung des Standards Anforderungen liefern, bewerten oder testen und damit den Standard bereits kennen, bevor ihn die Softwarefirma ausliefer.

Das **Beschäftigungssystem** der Softwaregestaltenden ist davon geprägt, dass sie in hierarchischen Organisationen angestellt sind. In den EVU gibt es für sie keine eigenständigen Kompetenzkarrieren. Sie machen Karrieren in der Hierarchie, und je nach EVU oder Team kann die Softwaregestaltung Teil einer Führungsposition oder ganz ausgelagert sein, z.B. an IT-Beratende. Was auffällt, ist, dass in einigen EVU die Team- und Gruppenleitung Software gestaltet, d.h., sie nimmt Einstellungen an der Software vor oder gibt Fehler-Tickets auf. Selbstständiges Arbeiten und IT-Affinität sind von Vorteil, wenn man Karriere machen will.

»Aber im Wesentlichen ist mein Anreiz sozusagen: Ich interessiere mich für Prozesse. Ich interessiere mich für Prozessoptimierung und habe halt sozusagen auch eine hohe IT-Affinität. Und das ist sozusagen ein bisschen auch mein eigener Antrieb für diese Position, in der ich jetzt bin.« (Gruppenleiter Abrechnung EVU5)

Softwaregestaltende der Softwarefirma wie IT-Beratende oder Lösungsarchitekten können für unterschiedliche EVU arbeiten, aber auch Softwaregestaltungsprojekte innerhalb der Softwarefirma durchführen. Aber anders als bei den ersten vier Fallstudien, die SAP verwenden, sind hier die IT-Beratenden ausschließlich in der Softwarefirma angestellt. Sie haben als Arbeitsmarkt für eine mögliche weitere fachliche Karriere nur die EVU, welche das Standardpaket anwenden, und keine große Anzahl an kooperierenden Firmen wie bei SAP.

Bei der **Kontrolle** der Softwaregestaltenden ist wie auch in den anderen Fallstudien die Erfolgskontrolle wesentlich. Die Führungskräfte erwarten selbstständiges Arbeiten und eigenständiges Lernen. Zugleich sind Softwaregestaltende entweder in Hierarchien eingebunden oder in Marktbeziehungen mit den entsprechenden Folgen. Letzteres bedeutet, dass z.B. SLA deren Arbeit mitkontrollieren. Innerhalb der Hierarchien setzen Führungskräfte den Rahmen für die Arbeit. Das gilt für Key User:innen oder Anwendungsbetreuende innerhalb der EVU, Lösungsarchitekten oder IT-Beratende in der Softwarefirma.

Eigenständiges Arbeiten und reine Erfolgskontrolle durch Führungskräfte kann zu einer extremen Belastung werden. Das ist vor allem dann der Fall, wenn eine Person sowohl das Tagesgeschäft der Softwareanwendung als auch der Softwaregestaltung übernehmen muss, zusätzlich einem Termindruck ausgesetzt ist und keinen Einfluss auf die Softwarefirma und deren Softwaregestaltung hat. So kann sich ein befragter Gruppenleiter mit der Softwaregestaltung nur nebenher beschäftigen und es gibt keine Person in seiner Kollegenschaft, die ihm die Softwaregestaltung abnimmt und sich mit der Softwarefirma austauscht. Er muss sowohl Anwendungsarbeiten erledigen als auch intern die Rolle als Anwendungsbetreuer übernehmen. Weil über einen längeren Zeitraum eine Kombination von Termindruck durch die Regulierung und stetige Updates der Software hinzukam, ist er fast ein Jahr aufgrund von Burnout ausgefallen.

Was die **Wissensverteilung** anbelangt, fällt bei diesem Fall besonders auf, dass die Softwaregestaltenden der EVU, die Einstellungen an der Standardsoftware vornehmen, nicht Teil des Praxisnetzwerks sind, das die Standardsoftware gestaltet. Mit der Folge, dass sie erst an dem fertigen Softwareprodukt über Neuerungen etwas lernen können. Das Learning by Doing und damit der Aufbau von Erfahrungswissen geschieht erst in der Anwendung der Standardauslieferung:

»Wir müssen es aber dann im [...] täglichen Geschäft praktisch beginnen, die Prozesse im Kern zu verstehen, und deswegen ist da Learning by Doing ganz häufig so, dass man dann die Fälle im System sieht und autodidaktisch versucht, sich die Sachen so zurechtzulegen, wie sie denn sein sollen und dann darüber den nötigen Erkenntnisgewinn zu bekommen und Worst Case hat man es gerade verstanden und dann kommt das nächste Update und es ist schon wieder hinfällig. Also, da ist man eigentlich im permanenten Sprint hinterherzukommen.« (Leiter EVU2)

Wobei in manchen EVU die Softwaregestaltenden privilegierten Zugang zu dem Wissen haben und Marktbeziehungen und Hierarchien keine Hürden sind. Dann sind aber die Softwareanwendenden noch nicht über Änderungen informiert, was zeigt, wie Hierarchien den Wissensaustausch im Team bestimmen:

»Also, die Anwendungsbetreuer sind viel intensiver im Austausch mit dem Softwarehersteller und dadurch >persönlich betreut< in Anführungszeichen. Die Mitarbeitenden, die operativ tätig sind und die Prozesse vollziehen, sind diejenigen, die es sich selbst beibringen aktuell. Und das ist eigentlich eine Reihenfolge, die ist eigentlich falsch.« (Betriebsrat EVU6)

Weil die Softwareanwendenden nicht Teil eines Gestaltungsnetzwerkes sind, hängt es von ihrer Eigenmotivation ab, sich mit der Software auseinanderzusetzen. Doch fällt es der Führungskraft eines EVU schwer, die Mitarbeitenden dazu zu bringen, eigenständig zu lernen.

»Es ist schon so, dass die Kollegen ambitionierter sein müssten, was ihr Arbeitsumfeld angeht. Also, sie müssten schon nach links und rechts gucken, wie man das so schön sagt oder versuchen eigenständig sich Dokumentationen anzulesen. Es gibt von der [Softwarefirma] unzählige Dokumente, die man sich durchlesen kann, und dann versteht man natürlich das Programm auch anders. Und da ist natürlich der Kollege, der sagt: Nö, damit möchte ich mich nicht beschäftigen. Es muss mir jemand erklären. Und wenn mir das niemand erklärt, dann...« (Gruppenleiter Anwendungsbetreuung EVU3)

Aus Sicht des befragten Gruppenleiters wären noch mehr IT-affine Leute hilfreich, die Anwendungsbetreuung machen und Einstellungen an der Software vornehmen können.

Die Softwaregestaltenden innerhalb der Softwarefirma arbeiten kontinuierlich interdisziplinär. Wobei das Wissen über die Branche wichtiger ist als jenes über die Anwendung in einzelnen EVU. Dabei sind Führungskräfte anders als in anderen Fallstudien eng eingebunden. Es zeigt sich der Vorteil, wenn Führungskräfte auch Wissen über

Softwaregestaltung haben. Der Teamleiter eines befragten Programmierers hat tieferes fachliches Wissen über die Energiewirtschaft:

»[D]er fachliche Teamleiter, [...] der sich um die Prozesse kümmert, das Fachliche im Kopf hat und uns dann genauer sagen kann: Okay, an der Stelle musst du das und das machen. Oder wenn man halt mal Fragen hat, wenn man mal den Prozess nicht so 100 % kennt: Hier, wie ist das in den und den Spezialfällen?« (Programmierer)

Der befragte Programmierer selbst meint, bei ihm ist das Verhältnis 40 % Wissen über die Energiewirtschaft und 60 % über die Programmierung. Das Problem an der Marktbeziehung ist für ihn, dass Programmierende schwerer an die Bedarfe der Anwendenden in den EVU herankommen, weil sie sehr weit weg von ihnen sind.

8.4.2.6. KOOP3: Softwaregestaltung als neues Betätigungsfeld, reines Netzwerk

In dem Fall behindert die Verteilung der Softwaregestaltung auf unterschiedliche Organisationen nicht die Arbeit der Softwaregestaltenden, was typisch für eine reine Netzwerkorganisation ist. Für die Beschäftigten stehen weniger Karrieren in den Hierarchien im Mittelpunkt. Sie können ihrer intrinsischen Motivation nachgehen und das Thema IoT für ihre jeweilige Organisation voranbringen. Die Befragten arbeiten selbstständig, weil sie IoT auch als ihr eigenes Projekt begreifen und nicht nur als eines ihrer Vorgesetzten oder ihrer Kundenschaft. Dabei ist das interdisziplinäre Wissen über die Möglichkeiten der Softwaregestaltung und die Bedarfe der EVU anders als bei PAKET gleichmäßig verteilt, für die Softwaregestaltenden gut zugänglich und es besteht eine gemeinsame Wissensbasis.

Die Softwaregestaltenden arbeiten in einem **Beschäftigungssystem**, das es ihnen erlaubt, kontinuierlich an dem Thema IoT zu arbeiten. Sie arbeiten zwar flexibel an Projekten mit, jedoch immer zum gleichen Thema IoT und der gleichen IoT-Standardsoftware der Softwarefirma. Es gibt sowohl in den EVU als auch dem IT-DL neue Aufgaben und gar neue Teams für IoT. Die Beteiligten begreifen es als ihr eigenes Projekt. Die Softwaregestaltung ist in diesem Fall ein klar abgegrenztes Betätigungsfeld. Aus den Interviews geht nicht hervor, dass mit der Mitarbeit an der Softwaregestaltung – ob im EVU, beim IT-DL oder in der Softwarefirma – eine hierarchische Karriere verbunden wäre. Weniger ist der Aufstieg in der Hierarchie ein Anreiz als beim Thema IoT dazuzulernen, interessante Projekte zu machen und die Software weiterzuentwickeln.

Selbstständiges Arbeiten und damit eine indirekte **Kontrolle** sind vorherrschend. Der Fokus der Beteiligten liegt auf der Koordination untereinander, was typisch für reine Netzwerke ist. Die Beteiligten vor allem des IT-DL und der Softwarefirma sind intrinsisch motiviert und sie nutzen formelle Wege vor allem dazu, um die Arbeit zu koordinieren. Die Zusammenarbeit ist persönlich, partnerschaftlich und auf Augenhöhe. Die Führungskräfte fungieren bei den befragten EVU als Bindeglied in den Hierarchien, wie es in Matrixorganisationen für die Softwaregestaltung hilfreich ist.

Trotz Organisationsgrenzen und Marktbeziehungen ist das **Wissen** gleichmäßig im organisationalen Netzwerk verteilt. Die Softwaregestaltenden verfügen nach längerer Mitarbeit im organisationalen Netzwerk über eine gemeinsame Wissensbasis, um mitgestalten zu können. Das Besondere an dem Fall ist, dass sowohl die Softwarefirma als

auch das IT-DL und EVU Wissen über die Implementierung (deren Durchführung und über Funktionalitäten und Anwendungsfelder der Software) und Erweiterung (zusätzliche Module programmieren) der Software haben. Das Wissen, was in den Quellcode der IoT-Software einfließt, kommt aus Implementierungsprojekten unterschiedlicher Anwendungsfälle des IT-DL mit EVU. Wie bei den anderen Fallstudien liefert die Softwarefirma neben der Software Wissen über Änderungen aus: Die Softwarefirma stellt bei Updates Frequently Asked Questions (FAQ) als Dokumentation von Funktionalitäten und Neuerungen zur Verfügung. Das IT-DL schickt zu Updates einen Newsletter an die EVU. Das heißt, Softwarefirma und IT-DL erwarten, dass die Empfangenden, für die diese Informationen von Interesse sind, sie sich selbst aneignen. Somit spielt die Marktbeziehung doch noch insofern eine Rolle, als hier kein direkter Austausch erfolgt. Es bleibt den anwendenden Einzelnen nur, sich eigenständig einzulesen.

8.4.2.7. STARTUP: Softwaregestaltende im Kern der Organisation, reines Netzwerk

Der Fall zeigt typische Eigenschaften davon, wenn Softwaregestaltende in einem reinen Netzwerk arbeiten: Was das Beschäftigungssystem anbelangt, können Mitarbeitende in der rollenbasierten Organisation ihren Interessen nachgehen, indem sie Rollen wechseln, und es geht nicht darum, in einer Hierarchie aufzusteigen. Die Kontrolle passiert im Wesentlichen horizontal durch die Kollegenschaft und ermöglicht Eigenmotivation, individuelle Steuerung der Arbeitsbelastung und Selbstständigkeit. Durch den kontinuierlichen, iterativen Austausch im Netzwerk für die Softwaregestaltung ist eine gemeinsame Wissensbasis und ein unkomplizierter Zugang zum Wissen gesichert. Weil das Wissen nicht hierarchisch oder über einen Markt verteilt ist, bestehen keine Hindernisse für den Austausch.

Was das **Beschäftigungssystem** anbelangt, können die Beschäftigten nicht in einer formalen Hierarchie aufsteigen, sondern nur Verantwortung für zusätzliche oder andere Rollen übernehmen. Um sich darüber auszutauschen, wer welche Verantwortungsbereiche hat, gibt es regelmäßig Treffen. So geht es im STARTUP um das Arbeiten an einem gemeinsamen Ziel und den möglichen zu teilenden Mehrwert durch eine wachsende Organisation:

»Je mehr erfolgreiche Ideen wir haben, die wir umsetzen können, desto mehr Kuchen ist halt da, der aufzuteilen ist.« (Programmierer1)

Der Primat der Softwareentwicklung zeigt sich in dem Fall daran, dass die Softwaregestaltenden Teil des stabilen Kerns der Organisation sind. In den ersten vier Fallstudien sind die Softwareanwendenden fest in den EVU angestellt und langfristige Mitglieder ihrer Teams. Sie sind dort auch schon länger. Es sind die Softwaregestaltenden, die sich flexibel in einer Matrixorganisation bewegen. Hier ist es umgekehrt, weil in dem Fall die Softwareentwicklung der Ausgangspunkt der Organisation ist. Die Anwendenden sind flexible 450-Euro-Kräfte, die weniger verdienen, Teilzeit arbeiten und nicht studiert haben. Von den anderen Beschäftigten haben alle bis auf jemanden aus dem Kommunikationsbereich einen akademischen Abschluss.

Bei der **Kontrolle** spielen Führungskräfte keine Rolle. Die Arbeit ist ein persönliches Projekt in stetiger Auseinandersetzung mit der iterativ gestalteten Software, der Kun-

den- und Kollegenschaft. Die Product Ownerin lernt viel und meint, es macht ihr Spaß. Sie schreibt Arbeitszeiten auf, um nicht zu viel zu arbeiten. Kontrolle gäbe es gar nicht. Druck entsteht durch sich verantwortlich fühlen, durch die Kundschaft und die Erwartungen an die Software.

»Druck entsteht eher dadurch, dass ich mich für vieles verantwortlich fühle und (gerade in einem Start-up) oft das Gefühl entsteht, dass es um die Existenz geht.« (Product Ownerin)

Man erwartet Verantwortungsbereitschaft und Eigenverantwortung voneinander. Die Peer-Kontrolle ist Teil der rollenbasierten Organisation.

In dem Fall behindern weder Hierarchien noch Marktbeziehungen den Wissensaustausch der Softwaregestaltenden. Das notwendige **Wissen** zur Softwaregestaltung befindet sich innerhalb von STARTUP und ist gemeinsame Basis der Arbeit. Den Kern der Organisation bilden interdisziplinäre Kreise, in denen sich regelmäßig Fach- und IT-Wissen treffen und die Beteiligten Möglichkeiten der Softwaregestaltung mit jenen der energiewirtschaftlichen Bedarfe abgleichen. Die Softwaregestaltenden können als Kern der Organisation immer auf Programmierende oder Anwendende zugreifen. Die befragte Product Ownerin meint, sie lernt durch die Teilnahme an den Kreisen etwas über interne Abläufe und den Nutzen für die Kundschaft. Alle, die z.B. wie die Anwendenden nicht an Kreisen teilnehmen, nehmen nicht am Wissensaustausch teil. Das Wissen konzentriert sich damit bei den Mitarbeitenden, die dort mitmachen, und sie können interdisziplinäres Wissen akkumulieren unabhängig von Hierarchien und Märkten. Es ist also nicht die Hierarchie, die die Wissensverteilung bestimmt. Wenn, dann ist die Wissensverteilung die Ursache für informelle Hierarchien.

8.4.3. Zusammenfassung

8.4.3.1. Unterschiede: zwischen reiner Netzwerk- und Matrixorganisation

Die Falldarstellungen zeigen, dass sie nicht immer klar einem der diametralen Typen von Matrixorganisation und reiner Netzwerkorganisation entsprechen und sich teilweise einzelne Elemente der beiden Typen vermischen. Trotzdem vereinen sie entweder mehr Eigenschaften von dem einen oder dem anderen Typ. So gehören KOOP3 und STARTUP zum Typ reiner Netzwerkorganisationen, die auf die Softwaregestaltung ausgerichtet sind. Dort müssen die Softwaregestaltenden in ihrer Arbeit keine Kompromisse mit bestehenden Strukturen eingehen. Reine Netzwerkorganisationen stellen einen Vorteil für die wissensintensive Softwaregestaltung im Vergleich zu jenen vom Typ Matrixorganisation dar (INTERN1, INTERN2, KOOP1, KOOP2 und PAKET). Hier sei zum Abschluss des Abschnitts kurz dargelegt, warum welcher Fall welchem Typ zugeordnet werden kann.

Bei STARTUP und KOOP3 zeigt sich die Netzwerkorganisation an der Kontrolle der Softwaregestaltung, bei der die Peer- und Objekt-Kontrolle (im Sinne Rennstams, siehe 6.4.2.1) zentral sind, Softwaregestaltende unabhängig von Führungskräften agieren und der Fokus auf der Koordination der Zusammenarbeit liegt. Das Beschäftigungssystem zeichnet sich durch Kompetenzkarrieren statt durch solche in Hierarchien aus. Es

geht um Arbeit als persönliches Projekt und darum, etwas zu lernen. Statt rein sozial in Hierarchien eingeteilt und Führungskräften untergeordnet zu sein, unterscheiden sich Programmierende, Softwaregestaltende und Anwendende in ihrer Position soziotechnisch durch ihren jeweiligen Bezug zur Software. Die Softwaregestaltenden haben keine Schwierigkeiten, auf das für ihre Arbeit notwendige Wissen zuzugreifen, weil eine langfristige Praxisgemeinschaft der Softwaregestaltung mit interdisziplinärer Wissensbasis besteht und die diversen Methoden der Softwaregestaltung den Kommunikations- und Wissensaustausch sicherstellen. Entweder sind die Softwaregestaltenden durch ihre Rollen und ihre Teilnahme in Kreisen Teil des interdisziplinär arbeitenden Kerns der Organisation wie bei STARTUP. Oder Markt und Hierarchien spielen im organisationsübergreifenden Netzwerk eine so geringe Rolle, dass sie für ihre Arbeit keine Hürden beim Wissensaustausch darstellen wie bei KOOP3.

Bei den anderen Fallstudien liegen Matrixorganisationen vor. Ob in IT-Projekten, Arbeitskreisen oder Scrum: Immer sind die Softwaregestaltenden Teil von Märkten und/oder Hierarchien, deren Kontrolle sie unterliegen. Es besteht eine Spannung darin, die Mitarbeitenden zu kontrollieren (ob durch Führungskräfte in Hierarchien oder Kundschaft in Marktbeziehungen) und ihnen die notwendige Freiheit für den kommunikativen Austausch und das Schreiben von Anforderungen zu geben, damit sie dabei ihre Subjektivität einbringen können, z.B. in Form intrinsischer Motivation. Die Beschäftigungssysteme für Softwaregestaltende zeichnen sich dadurch aus, dass sie Teil von Märkten oder Hierarchien sind und dadurch entsprechende Karrierechancen haben: in einer internen Hierarchie aufzusteigen oder in andere Organisationen zu wechseln. Vor allem Ersteres hat ein begrenztes Potenzial, um sich rein fachlich weiterzuentwickeln und dafür die (finanzielle) Anerkennung zu bekommen. In den Matrix-Fallstudien ist das Wissen durch Hierarchien oder Märkte getrennt. Doch stellen die Beteiligten Interdisziplinarität nicht nur temporär her. Es ist noch mehr als sonst bei der Softwaregestaltung von Vorteil, wenn die oder der Einzelne viel weiß. Dann kann er nämlich unabhängig von Hierarchien und Marktbeziehungen agieren und eine fehlende interdisziplinäre Wissensbasis bei Anwendenden und Programmierenden ausgleichen. Dabei ist es hilfreich, wenn Führungskräfte der EVU sich nicht nur als Anwendende begreifen und entsprechend auch Wissen über Software und Softwaregestaltung haben. Sie nehmen in einer Matrixorganisation zentrale Positionen ein und können die Softwaregestaltung unterstützen, z.B. indem sie eine abteilungsübergreifende Zusammenarbeit fördern.

8.4.3.2. Ergebnisse: Fallvergleich je Kategorie und Vergleich mit Anwendenden und Programmierenden

Die Falldarstellungen haben gezeigt, dass Softwaregestaltende zwar immer in organisationalen Netzwerken arbeiten, die reine Netzwerkorganisation jedoch Vorteile gegenüber der Matrixorganisation hat. Durch den Vergleich mit den Beschäftigtengruppen der Anwendenden und Programmierenden arbeitet dieser Abschnitt noch einmal heraus, was das Beschäftigungssystem, die Kontrolle und die Wissensverteilung der Softwaregestaltenden ausmacht.

Beschäftigungssystem – Kompetenzkarriere, flexible Beschäftigte und Akademiker:innen-Dominanz
Insgesamt zeigen die Fallstudien, dass sich neben einer allgemeinen Akademisierung ein Karrieresystem unabhängig von Hierarchien und der energiewirtschaftlich-fachlichen Arbeitsteilung wie z.B. Netz, Vertrieb, Handel etabliert.

Es fällt auf, dass Programmierende und Softwaregestaltende allesamt studiert haben und es bei den Anwendenden mehr die Key User:innen, Prozessmanagende oder Anwendungsbetreuende sind, die einen akademischen Abschluss haben – also eben jene, die sich mit der Softwaregestaltung befassen. Eine befragte Sachbearbeiterin braucht ein abgeschlossenes Studium, um mehr Projektarbeit machen zu dürfen. Die Bedeutung von IT-Wissen nimmt zu und hochautomatisierte und digital-integrierte Prozesse machen ein tiefergehendes und breiteres fachliches Wissen für die (Fehler-)Fallbearbeitung notwendig.

Wie die Fallstudien zeigen, haben Softwaregestaltende im Unterschied zu Programmierenden und -anwendenden ganz eigene Karrieremöglichkeiten – und zwar unabhängig davon, ob sie innerhalb eines EVU angestellt sind oder für einen IT-DL arbeiten. Die meisten befragten Softwaregestaltenden wollen keine Führungskraft werden oder in der Hierarchie aufsteigen, sondern eine Karriere als Fachexpert:innen und damit z.B. interessante Projekte machen. Vor allem in den jüngeren Organisationen, wo es um IoT oder Emissionshandel geht, ist die Arbeit ein persönliches Projekt der Beschäftigten, mit dem sie sich identifizieren. Beim IT-DL von KOOP2 gibt es bereits Kompetenzkarrieren, bei denen man durch Klettern auf definierten Kompetenzprofilen die entsprechenden Gehaltsbänder nach oben steigt. In den bestehenden Hierarchien in den EVU sind in einigen Fällen Kompetenzen in der Softwaregestaltung hilfreich für eine Führungskräftekarriere oder der Aufstieg damit verbunden, Softwaregestaltung zu koordinieren.

Eine Gemeinsamkeit zwischen Programmierenden und Gestaltenden ist, dass sie sich auf die Vermarktung ihrer Arbeitskraft konzentrieren¹⁸, vor allem wenn sie für ein IT-DL arbeiten. Das tun sie zum einen für das IT-DL selbst, weil es Geld damit verdient, seine Fachkräfte zu verkaufen bzw. auszuleihen. Sie vermarkten ihre Arbeitskraft auch für die eigene weitere Karriere, um in anderen Organisationen mehr zu verdienen oder interessantere Projekte zu machen. Im Vergleich zu den Anwendenden haben Programmierende und Gestaltende eine bessere Bezahlung und bessere Marktchancen. Beide Gruppen zielen eher auf eine fachliche Karriere ab. Wenn sie in einer Matrixorganisation arbeiten, teilt sie die Softwaregestaltenden flexibel zu IT-Projekten zu, um sie stetig auszulasten.

In den Fallstudien gibt es für Anwendende im Vergleich zu den anderen Gruppen weniger Karrieremöglichkeiten. Vertiefte Softwarekenntnisse zu erwerben, kann für sie hilfreich sein, um in der Hierarchie aufzusteigen. So bedeutet der Aufstieg als

¹⁸ Sieht man die sozialen Netzwerke von Xing und LinkedIn als Wege, sich selbst zu vermarkten, unterstützt der Feldzugang (siehe Kapitel zu Forschungsdesign und Methode, siehe 3.2.1) die These, dass Softwaregestaltende und Programmierende sich stärker am externen Arbeitsmarkt orientieren: Auf den Webseiten von Xing und LinkedIn sind vor allem Programmierende, IT-Beratende, ERP-Fachleute, Product Owner:innen, Projektmanagende, Anforderungsmanagende oder Scrum Master:innen zu finden.

Key User:in einen Aufstieg über Softwaregestaltung. Einige der befragten Team- oder Gruppenleitenden waren vorher Sachbearbeitende und damit reine Anwendende.

Kontrolle - eigenständiges Arbeiten in softwarezentrierten Prozessen

Beim Vergleich der Fallstudien überrascht, dass Führungskräfte in der Softwaregestaltung eine ähnliche Rolle spielen wie in der Anwendung und Programmierung. Unabhängig vom Arbeitsprozess halten sich die Führungskräfte aus der operativen Arbeit weitestgehend heraus. Bei allen ist der Kern der Kontrolle ein softwareintegrierter Arbeitsprozess, kombiniert mit Führungskräften, die nur in Ausnahmesituationen intervenieren und, wenn überhaupt, die Ergebnisse kontrollieren. Man könnte auch sagen: Die Beschäftigten arbeiten weitestgehend selbstständig an und mit der Software, die sie in einen digitalen Prozess integriert. Wobei die Grundkoordination (der soziotechnischen Konstellation) zwischen Anwendung und Programmierung Folgen für die Kontrolle hat.

Kontrolleigenschaften unabhängig von Gruppenzugehörigkeit: selbstständiges Arbeiten, softwarezentrierter Arbeitsprozess, Auswirkungen der Grundkoordination

Zuerst zu jenen Dingen, welche die Kontrolle der Arbeit unabhängig davon prägen, ob sie Teil von Softwareanwendung, -gestaltung oder -programmierung sind: Erstens arbeiten die Befragten eigenständig. Zweitens arbeiten alle softwarezentriert. Drittens prägt bei allen die soziotechnische Konstellation die Arbeit.

Erstens geben in allen Fällen und für alle Gruppen die Befragten an, dass sie eigenständig arbeiten und die Führungskräfte nur in Ausnahmefällen eingreifen und die Arbeitsergebnisse kontrollieren. Sie sind disziplinarische Führungskräfte, geben aber nur selten oder gar keine Arbeitsanweisungen. Je nach Fall legen sie das Budget fest, stellen Mitarbeitende ein, initiieren Projekte oder klären Konflikte innerhalb (z.B. mit anderen Führungskräften oder Teams) oder zu anderen Organisationen (Softwarefirma, IT-DL, EVU). In manchen Fällen wirken sie bei der Softwaregestaltung mit (EVU2 von KOOP2, einige EVU bei PAKET). Einige Zitate sollen das nochmal verdeutlichen:

Anwendende:

»Wenn wir uns eine Woche lang nicht mehr zurückgemeldet haben, dann kommen schon Fragen: Warst du überhaupt beim Arbeiten oder hast du überhaupt die Arbeit gemacht? Wir arbeiten viel bei Kunden und wenn die etwas von uns wollen und da ist der Monteur nicht gekommen, der meldet sich natürlich auch dann, wie es denn aussieht, wenn er seinen Baustrom kriegt oder so.« (INTERN1 Monteur)

»Ich war sehr selbstorganisiert, tatsächlich auch schon in der Abrechnung. Und meine Chefin hat mir da eigentlich auch immer den Weg auch so gelassen, weil ich hatte ja meine Fälle zugewiesen. Um die habe ich mich gekümmert. [...] Natürlich hat jede Führungskraft drauf geschaut, wenn irgendwas länger liegt, wie es aussieht, warum es länger liegt. [...] Ich habe meine Arbeit immer schon, ich sag es einfach mal, gut gemacht, so dass ich nicht kontrolliert werden musste.« (KOOP2 EVU2 Teamleiterin Mako)

»Ja, ich bin recht frei in meiner Arbeit, tatsächlich. Ich darf sehr viel, Gott sei Dank. Ich werde nicht so viel kontrolliert.« (KOOP2 EVU2 Sachbearbeiterin)

Programmierende:

»Und war haben Reviews, Code Reviews: Da kontrollieren wir uns gegenseitig auf Augenhöhe. Und wir haben verschiedene Analyse-Tools, die uns selber helfen, uns selbst zu kontrollieren. Also, im Grunde ist eigentlich die Eigenmotivation sehr, sehr hoch und für einen selbst, aber auch aus dem Entwicklerteam heraus. Und da gibt es eine gewisse Kontrolle. Wobei das nicht als Kontrolle empfunden wird, sondern eher als Unterstützung, würde ich sagen. Es gibt aber keinen, der auf meine Arbeit draufguckt. Könnte auch glaube ich keiner.« (INTERN1 Programmierer)

Gestaltende:

»Es ist so bei uns, dass wir unsere Arbeitsbelastung gut selbst managen können.« (KOOP1 Applikationsbetreuer EVU2)

»Aber zum großen Teil sind wir da relativ alleine unterwegs oder alleinständig¹⁹ ... eigenständig, so wollte ich sagen« (KOOP1 Anforderungsmanager IT-DL)

Zweitens sind die Befragten alle Teil eines softwarezentrierten Arbeitsprozesses mit unterschiedlichen Schwerpunkten, welche Software sie für ihre tägliche Arbeit hauptsächlich verwenden (ERP-System, Ticketsystem, E-Mails, MS Excel etc.). Die Anwendenden werden von der Software gesteuert (Monteur:innen) oder arbeiten ausschließlich mit ihr (Sachbearbeitende, Kund:innenservice). Genauso die Programmierenden, die ihre Programmierung über die Entwicklungsumgebung abarbeiten. Für die Softwaregestaltenden sind Ticketsysteme zentral. So spielt bei allen Software als ein Wissensobjekt (nach Rennstam 2012, ausführlich unter 6.4.2.1) und damit eine Form von Objekt-Kontrolle durch Technik eine Rolle. In allen Fällen werden die Tätigkeiten digital geführt – ob durch Ticketsystem, Entwicklungsumgebung oder ERP-Software.

Drittens prägt die Grundkoordination die Kontrolle: Von ihr hängt der Schwerpunkt der Kontrolle durch die Software und durch eine Gruppe (Management, Kundschaft, Kollegenschaft) ab.

Bei KOOP3 und STARTUP ist die Software vor allem ein Bezugspunkt zur Koordination und die softwarebasierten Werkzeuge fungieren mehr als Koordinationsinfrastruktur denn zur Kontrolle. Die Softwaregestaltenden können sie von überall aus nutzen. In den genannten Fällen kontrollieren aufgrund der digitalen Transparenz durch die verwendeten Werkzeuge und Methoden die Kolleg:innen mit. Bei KOOP1, KOOP2 und PAKET erfüllen die Ticketsysteme eine Funktion für die Kundschaft. Sie machen für diese transparent, was die Softwarefirma bzw. das IT-DL macht, und sind die Basis für die Überprüfung von SLA (z.B. die vereinbarten Reaktionszeiten für Tickets zu kontrollieren). Bei den internen Fällen ist nicht ganz klar, inwiefern das Management das Ticketsystem nutzt. Keine befragte Person hat das erwähnt oder von individueller Verhaltenskontrolle gesprochen. Trotzdem besteht intern für alle anderen Beteiligten, die an den

¹⁹ Auch wenn die Interpretation einzelner Phrasen und Wörter nicht Teil der Forschungsmethode ist, so zeigt dieser Versprecher doch auf, wie der Arbeitsalltag für viele aussieht: Sie arbeiten viel allein und bekommen wenig bis gar keine Anweisungen durch die Führungskraft im Arbeitsalltag.

Tickets arbeiten, eine Transparenz über den Arbeitsfortschritt. Keine befragte Person berichtet davon, dass Arbeitsschritte und das Arbeitstempo durch die Software vorgegeben sind. Vielmehr liefert die Software für jeden (Management, Kundschaft, Kollegenschaft) eine Rückmeldung, wie viele Fälle noch offen sind, welchen Bearbeitungsstand ein Ticket hat oder wo das nächste Strom- oder Gasnetz zu warten ist.

Je nach Grundkoordination ist der Schwerpunkt unterschiedlich, welche Gruppen kontrollieren: Die Fallstudien unterscheiden sich, ob mehr das Management (Hierarchie), die Kundschaft (Markt) und das Gegenüber (Netzwerk) kontrolliert. Bei den internen Softwaregestaltungen (INTERN1, INTERN2) entscheidet das Management über die Ressourcen (IT-Budget, Stellenvergabe, Entscheidung über Projekte). Zusätzlich zum Management gibt es bei den firmenübergreifenden Gestaltungsprozessen (KOOP1, KOOP2, PAKET, KOOP3) noch die Kundschaft bzw. die EVU, die in Form von SLA und Preisen kontrollierend wirken. Bei KOOP1, KOOP2 gibt es festgelegte Reaktionszeiten für bestimmte Tickets, welche die EVU den IT-DL stellen. Bei STARTUP und KOOP3 ist es schwer zu entscheiden, ob die Kolleg:innen oder die Kund:innen wichtiger sind. Denn sowohl in der Kollegenschaft besteht ein stetiger Austausch als auch mit der Kundschaft, die die App zur Anmeldung von E-Autos bzw. die IoT-Software nutzt. So oder so ist in beiden Fällen die hierarchische Kontrolle schwach. In keinem der Fälle gibt es eine direkte Kontrolle durch die Führungskraft in Form von konkreten Arbeitspaketen und wie diese abzuarbeiten sind.

In der Tabelle sind die Schwerpunkte der Kontrolle durch Gruppe und Ticketsystem je Fall aufgelistet, die mit der Grundkoordination zusammenhängen:

Tabelle 24: Vergleich Schwerpunkt der Kontrolle in Abhängigkeit zur primären Grundkoordination je Fall

Fall	Primäre Grundkoordination	Schwerpunkt Kontrolle durch Gruppe	Besondere Verwendung Ticketsystem
INTERN1	Hierarchie	Management (IT-Budget, Ziele)	Management-Kontrolle
INTERN2	Hierarchie	Management (IT-Budget, Ziele)	Management-Kontrolle
KOOP1	Markt	Verträge (SLA), Preise	Kontrolle durch Kundschaft bei IT-DL
KOOP2	Markt	Verträge (SLA), Preise	Kontrolle durch Kundschaft bei IT-DL
PAKET	Markt	Verträge (SLA), Preise	Kontrolle durch Kundschaft bei SF
KOOP3	Netzwerk	Ko-Produktion	Koordination
STARTUP	Netzwerk	Kollegenschaft/Peers und Kundschaft	Koordination

Kontrolleigenschaften je Gruppe: Unterschiede in der qualitativen und quantitativen Intensivierung
Im Vergleich zur Gruppe der Anwendenden und Programmierenden sind die **Gestaltenden** zum einen Teil eines anderen Arbeitsprozesses und allein deshalb schon anders kontrolliert. Wie bereits in den vorhergehenden Kapiteln ausgeführt, spielt für sie in ih-

rer Arbeit Kommunikation und Kooperation eine größere Rolle. Ihre Arbeit ist weniger durch die Software vorgegeben (z.B. bearbeiten sie nicht nur Restfälle). Für sie ist Software mehr ein Hilfsmittel. Sie sind interdisziplinär zwischen Anwendung und Entwicklung tätig. Mehrere Befragte geben explizit an, dass sie ihre Arbeit gern machen. Ihre Arbeit ist unterschiedlich stark formalisiert (z.B. wie genau der Ablauf des Anforderungsmanagements vorgegeben ist). Sie sind einer *qualitativen Intensivierung* ausgesetzt. Erstens sind sie das, weil sie in vielen Fällen IT-, energiewirtschaftliches und Methodenwissen kombinieren (Key User:innen, IT-Beratende, Anwendungsbetreuende) oder zumindest mit den jeweiligen Expert:innen kommunizieren können müssen (Anforderungsmanagende, IT-Projektmanagende). Zweitens arbeiten sie öfters gleichzeitig in mehreren Projekten und müssen mehrere Prozessabschnitte der Datenverarbeitung überblicken und die gestaltete Software ist nicht ihr alleiniger Arbeitsgegenstand. Die *quantitative Intensivierung* ist bei den Gestaltenden schwer zu beurteilen. Mehrere sprechen davon, dass es phasenweise mal mehr und mal weniger zu tun gibt und sie selbst Grenzen setzen müssen (durch Priorisierung oder »Nein« sagen können). Die meisten sprechen von einer geregelten Arbeitszeit. Zudem sind die Kontexte sehr unterschiedlich: ob sie für mehrere Kund:innen und in mehreren Projekten tätig sind oder nicht.

Die Arbeit der **Programmierenden** wird durch die Tests anderer und/oder durch Code-Reviews von Kolleg:innen geprüft. Es ist durch die Ticketsysteme transparent, was sie abgearbeitet haben, und falls es regelmäßige Treffen wie Daily bei Scrum gibt, müssen sie den Teammitgliedern Rede und Antwort stehen. Ihre Spezialisierung und die meist schwierige Abschätzung des Arbeitsaufwandes geht mit einer fehlenden detaillierten Kontrolle von außen einher. Eine *qualitative Intensivierung* existiert bei den befragten Programmierenden, weil sie fachliches mit IT-Wissen kombinieren und in verschiedenen Projekten, Rollen oder für mehrere Kund:innen tätig sind. Sie spitzt sich bei Programmierenden wie im Fall eines Befragten von KOOP2 zu, der für mehrere, unterschiedliche energiewirtschaftliche Fachbereiche entwickeln muss und dadurch entsprechend mehr Wissen braucht. Zudem müssen aufgrund individueller Anpassungen für einzelne EVU bei KOOP1 und KOOP2 die Programmierenden den Überblick über diese individuellen Umsetzungen behalten. Einige Programmierende beschreiben es als positiv, wenn sie sich auch mal in neue Technologien einlernen müssen. Wie bei den Gestaltenden ist es bei den Programmierenden schwierig, pauschal von einer *quantitativen Intensivierung* zu sprechen. Sie arbeiten in den untersuchten Fällen meist Programmieraufgaben ab, für die Prioritäten vorgegeben sind. Das lässt Spielraum, um eigene Grenzen zu setzen:

»Ich könnte mich gerne überlasten, aber ich tu's nicht.« (Programmierer KOOP1)

Eine Intensivierung ist aber nicht ausgeschlossen, wenn z.B. eine fixe Deadline existiert: wenn zu einem Stichtag mehrere umfangreiche Umsetzungen, z.B. aufgrund einer neuen Regulierung, abgeschlossen werden müssen. Zwar arbeiten die Programmierenden in einigen Fällen für mehrere EVU gleichzeitig (bei KOOP1, KOOP2, KOOP3 und PAKET). Doch hat dies nicht zwangsläufig eine quantitative Intensivierung zur Folge. Eine Programmierung kann für viele EVU gleichzeitig gelten (bspw. bei einer Standardsoftware), d.h., eine Fehlerkorrektur kann gleich mehrere Anfragen von EVU zufriedenstellen.

Die **Anwendenden** sind hierarchisch und von der Bezahlung Programmierenden und Gestaltenden untergeordnet. Die Anwendenden sind meist passiv und reagierend: Sie müssen sich auf Änderungen einstellen oder erfahren erst nach Fertigstellung, was sich geändert hat. Ihre Arbeit ist das Objekt der Gestaltung und nur wenige gestalten mit. Vor allem wenn die Softwaregestaltung intern stattfindet, können Anwendende eigene Wünsche äußern und der Betriebsrat die Mitgestaltung einfordern. Ihre Arbeit ist in den meisten Fällen das, was die Software nicht erledigen kann bzw. noch nicht in dieser abgebildet ist. Das bedeutet auch, dass sie fehlerhafte oder nicht ausgereifte Software kompensieren müssen. Sie fungieren als Puffer für mangelhafte Softwareentwicklung. In einem Fall ist die Cloud langsam (so ein Befragter von KOOP2), im anderen sind in den Softwareupdates Fehler, welche die Arbeit der Anwendenden belasten (PA-KET), oder die Prozessintegration leistet der Anwendende manuell (INTERN2). Sie müssen mit den Reaktionszeiten leben, die sich durch die Zusammenarbeit mit einem IT-DL oder Softwareunternehmen ergeben. Diese Puffer-Position kann über Fragen der Softwaregestaltung hinausgehen und die Anwendenden können zu allgemeinen, operativen Puffern werden. Ein befragter Gruppenleiter, der auch operative Software anwendet, hatte Burnout. Diesen hatte er vor allem, weil zu wenig Personal vorhanden war, und wegen der ständigen regulatorischen Änderungen und des Termindrucks im Bereich der Energiemengenbilanzierung, in der er tätig ist. Bei der Energiemengenbilanzierung müssen zu einem Stichtag nicht nur Daten, sondern vor allem Geld fließen. Anderen Anwendenden wie der befragten Sachbearbeiterin war die Restfallbearbeitung zu viel Routinearbeit, weswegen sie die Stelle gewechselt hat. Zudem sind die Anwendenden direkt im Kontakt mit der Kundschaft. Der befragte Monteur meint, er bekommt die Aggressionen der Kundschaft ab, wenn er bspw. wegen einer Baustelle den Strom bei jemandem abstellen muss. Für einen Teil der Anwendenden findet eine *qualitative Intensivierung* statt, weil sie vermehrt in Prozesszusammenhängen denken müssen (Sachbearbeitende), mehrere Fachgebiete (Key User:innen) oder mehrere Netzgebiete und Sparten kennen müssen (Monteur:innen). Arbeiten Anwendende im BPO-Bereich, arbeiten sie für eine größere Kundschaft und es ist mit einer *quantitativen Intensivierung* zu rechnen. Wobei in einigen EVU diese auch unabhängig davon aufgrund von Personalmangel oder phasenweise durch IT-Projekte entsteht.

Wissensverteilung – Praxisgemeinschaft und in Software materialisiertes Wissen

Es lassen sich grundsätzlich drei Wissensgruppen unterscheiden: Programmierung, Anwendung und Gestaltung. Dabei fällt in den Fallstudien auf, dass es für die Softwaregestaltenden weniger darum geht, viel Wissen anzusammeln. Sie sind vielmehr dazu da, um Wissen in Software(quellcode) zu überführen. Primär geht es darum, an das notwendige interdisziplinäre Wissen heranzukommen und lernbereit zu sein. Wissen selbst zu haben ist zweitrangig, wenngleich nützlich und in den meisten Fällen vorhanden. Das Wissen von Softwaregestaltenden kann umfassend sein: Wissen über die Möglichkeiten der Softwaregestaltung und individuelle Bedarfe der EVU oder allgemeine der Branche zu haben. Methoden wie Scrum oder IT-Projektmanagement zu beherrschen. Manche setzen Softwaregestaltung selbst um, wenn sie programmieren oder Einstellungen an einer Standardsoftware vornehmen. Der Extremfall sind IT-Beratende, die sich energie-

wirtschaftlich auskennen, Anforderungen schreiben, selbst programmieren und Einstellungen an einer Standardsoftware vornehmen. Doch auch sie müssen sich mit Anwendenden oder anderen Beteiligten wie IT-DL oder Softwarefirma absprechen. Die Praxis ist deshalb so entscheidend, weil die Software sich stetig verändert, ebenso wie die Organisationen und die Branche. Es muss immer wieder neu verstanden werden, was energiewirtschaftlich notwendig und was softwaretechnisch möglich ist. Drei Thesen stehen im Mittelpunkt der Zusammenfassung zur Kategorie Wissensverteilung:

1. Die Softwaregestaltung ist eine Praxisgemeinschaft, in der die Softwaregestaltenden arbeiten.
2. Für das Verständnis der Wissensverteilung ist das in Software materialisierte Wissen entscheidend.
3. Das Management verfügt nur selten über Wissen zur Softwaregestaltung.

Erstens stellt sich der Arbeitsprozess der Softwaregestaltung in den Fallstudien als eine Praxisgemeinschaft dar. Sie verfügt über das Wissen zu Methoden der Softwaregestaltung, und wenn sie schon selbst kein tiefergehendes Wissen über energiewirtschaftliche Regulierung, Standardsoftwarelösungen oder Programmierung hat, schafft sie zumindest die Möglichkeit, sich darüber auszutauschen – ob längerfristig oder temporär in Projekten. Dass die Praxis wichtiger ist, als Wissen zu besitzen, zeigt sich an drei Aspekten: an a) dem kontinuierlichen Austausch zur Softwaregestaltung, der stattfindet und zu Anforderungen führt (von dem bestimmte Gruppen ausgeschlossen sind), dass b) unterschiedliche Praxis- und damit Lernbiografien entstehen und c) EVU feststellen, dass sie nur dann Software gestalten können, wenn sie Teil dieser Praxis sind.

Zu a) In der Softwaregestaltung arbeiten Beschäftigte zwischen Anwendung und Programmierung zusammen. Auch einzelne Anwendende wie Key User:innen sind Teil davon. Die Gestaltenden haben Kontakte, kennen die Organisation und einzelne Umsetzungen und Methoden wie Scrum. Es besteht ein kontinuierlicher Austausch zwischen einem Kern an Mitarbeitenden in den Fällen INTERN1, INTERN2, KOOP3, STARTUP, genauso wie bei PAKET intern in der Softwarefirma. Bei KOOP1 und KOOP2 sind es längere Beziehungen zwischen den Firmen und viele Treffen, welche eine gemeinsame Praxis herstellen. Aber auch innerhalb des IT-DL und teilweise in den EVU arbeiten die verschiedenen Gruppen kontinuierlich zusammen. Bestimmte Berufsgruppen wie IT-Beratende erfahren einen kontinuierlichen interdisziplinären Wissenserwerb. Sie machen nicht nur in verschiedenen Projekten mit, sondern sind sowohl fachlich nah an den Anwendungsbereichen als auch softwaretechnisch nah an der Software dran. Die Programmierenden verfügen in allen Fällen über energiewirtschaftliches Wissen, um die industriespezifischen Anforderungen umzusetzen.

Zu b) Dabei bestimmt die Praxis, was die oder der Einzelne weiß. Es entstehen durch individuelle, interdisziplinäre Praxisbiografien individuelle Lernbiografien, weil je nach Softwaregestaltung die Softwaregestaltenden an unterschiedlichen Softwarelösungen, an unterschiedlichen Anforderungen, Projekten, energiewirtschaftlichen Themengebieten und mit unterschiedlichen EVU zusammenarbeiten. Wer mitgemacht hat bei der Gestaltung, weiß mehr darüber. Die klassische Lernbiografie von Ausbildung, dann Sachbearbeiterertätigkeit, Wechsel in ein anderes Team oder Aufstieg zur Teamleitung gibt es

immer noch. Diese wird jedoch nun durch neue Möglichkeiten ergänzt. Vor allem die befragten Gestaltenden und Programmierenden haben mehr interdisziplinäres Wissen und Einkommen im Laufe der Zeit gewonnen, ohne dadurch hierarchisch aufgestiegen zu sein.

Zu c) Dass die Praxisgemeinschaft den Zugang zu Wissen sichert, zeigt sich daran, dass das Wissen, das über Anforderungen in der Software eingeflossen ist oder dokumentiert wurde, allein nicht ausreicht. Es macht einen Unterschied, ob man Teil der Praxis war bzw. ist oder nicht. Das zeigen vor allem Marktbeziehungen, die zu einer Abhängigkeit führen und Hürden für eine gemeinsame Praxis darstellen. Bei KOOP2 verlagert ein EVU die Anwendung der Software zurück in das Unternehmen, um intern Wissen zu haben und die Softwaregestaltung selbst machen zu können. Zudem bauen einige EVU intern IT-Projektmanagement-Kompetenz auf. Innerhalb des IT-DL von KOOP1 arbeiten die Programmierenden eng mit energiewirtschaftlichen Fachleuten zusammen, wodurch ein stetiger Wissensaustausch besteht. Ein Klient des IT-DL von KOOP1 hat das Wissen, was das IT-DL über die Jahre über das EVU gesammelt hat, unterschätzt. Er hat den Wechsel zu einem anderen IT-DL bereut. Bei PAKET sind kleinere EVU abhängig von externem Support der Softwarefirma (z.B. durch IT-Beratende), weil sie intern keine ERP-Fachleute haben. Bei KOOP3 schickt das IT-DL Newsletter über Softwareupdates, die sich die EVU durchlesen können. Die reine Anwendung, die nicht an der Softwaregestaltung teilnimmt, ist aus der Wissensgemeinschaft Softwaregestaltung ausgeschlossen. Das ist in allen Fallstudien so.

Zweitens prägt die Wissensverteilung die Arbeit der Softwaregestaltenden dahingehend, dass der zentrale Ort des Wissens die Software selbst ist: ob Quellcode der gestalteten Software oder die verwendeten Softwarewerkzeuge wie Ticketsysteme. Immer mehr Wissen, ob über energiewirtschaftliche Regulierung, Geschäftsprozesse, Arbeitsabläufe, Softwareänderung etc., wandert in Software. Dabei müssen die Softwaregestaltenden alles, was sie gestaltet haben, nicht mehr aktiv wissen – auch nicht, wie es die Programmierenden umgesetzt haben. Das zeigen alle Fallstudien: Peu à peu wächst durch Anforderungen die Software. In ERP-Systemen wie jenem aus der Fallstudie PAKET steckt alles, was zum operativen Betrieb der energiewirtschaftlichen Datenverarbeitung notwendig ist. Dabei findet die Softwaregestaltung in einem technischen Umfeld aus lesbaren Objekten statt. Softwaregestaltende dokumentieren in den softwarebasierten Werkzeugen wie Ticketsystem, Dokumentationssoftware²⁰, MS Sharepoint oder E-Mails (inkl. Newsletter über Updates), Programmierende im Quelltext der Software selbst.

Was alle Befragten egal welcher Gruppe sagen, ist, dass Learning by Doing für sie wichtig ist – ob aus Dokumenten oder mit der Software selbst. Auch wenn viel Wissen lesbar oder durch Ausprobieren und Mitmachen erlernbar ist, sieht ein Befragter Software als primäres Wissensquelle und -lager kritisch: Ein Betriebsrat findet, dass sein EVU zu sehr auf Learning by Doing im Softwaregebrauch setzt. Es sollten wieder mehr Schulungen stattfinden. Ein befragter Programmierer hält eine Art Führerschein für Anwendende für hilfreich, weil sie in der alltäglichen Anwendung nicht alles lernen (bspw. die Zusammenhänge und Einstellungsmöglichkeiten in der Software). Bei PAKET sagen

²⁰ Eine weit verbreitete Softwarelösung dafür, die auch STARTUP einsetzt, ist Confluence der Firma Atlassian.

mehrere befragte Personen, dass die Anwendenden eine funktionierende Software wollen, sich nicht eigenständig mit den möglichen Einstellungen befassen möchten und in ihrem Arbeitsalltag nicht genug dazu lernen.

Drittens verfügt das Management wie Team-, Abteilungs-, Bereichs- oder IT-Leitung nur in wenigen Fällen über Softwaregestaltungswissen. Größtenteils nimmt es nicht direkt an der Softwaregestaltung teil und interveniert teilweise nur bei der Auswahl der Anforderungen. Drei befragte Team- bzw. Gruppenleitende wirken bei der Softwaregestaltung mit. Wobei es dabei immer um eine Standardsoftware geht, die nur durch Einstellungen von den EVU verändert werden kann. Softwaregestaltungskompetenzen haben nur wenige FK (wie z.B. Schulungen in Scrum oder IT-Projektmanagement). Das Management setzt den Rahmen z.B. durch das IT-Budget oder kann in Matrixorganisationen ein wichtiger Sponsor für Softwaregestaltungsprojekte sein, indem es z.B. die Zusammenarbeit mit anderen Abteilungen durch die Führungskräfte dort vorantreibt. KOOP1, KOOP2 und PAKET nutzen Marktmechanismen und kaufen die Software(ge-staltung) ein. Der Nachteil ist, dass das Wissen dann nicht direkt im Zugriff des EVU-Managements ist. Wie auch allgemein für Anwendung und Entwicklung hat das alles zur Folge, dass das Management bzw. die Führungskräfte keine direkten, konkreten Anweisungen mehr geben können, weil ihnen das Wissen dazu fehlt. In dem Fall, in dem Arbeitsanweisungen gemacht werden, erstellen dies die Key User:innen (PAKET, EVU5). In anderen Anwendungsbereichen werden keine verwendet (Sachbearbeitende KOOP2 EVU1, Teamleitung EVU2 KOOP2, Anwendende KOOP1 EVU3, Anwendende EVU1 KOOP1). Die Sachbearbeitenden von EVU2, KOOP2 haben nur ein Schema, wie sie vorgehen sollen, weil die Fälle zu individuell sind.

Allgemein wird das Wissen immer umfangreicher und eine Organisation allein kann nicht mehr darüber verfügen. Zum Beispiel sind bei INTERN1 und INTERN2 einige programmierende Externe. Bei KOOP1, KOOP2, KOOP3 verteilt sich das Wissen auf mehrere Organisationen. Die fachliche Spezialisierung der Mitarbeitenden nimmt zu und damit das Wissen, was ihre Führungskräfte nicht haben: u.a. durch Automatisierung, komplexere Regulierung, gestiegene Bedeutung von IT, höhere Ansprüche der Kundschaft, Produktvielfalt.

8.5. Folgen für die soziotechnische Arbeitsgestaltung der Softwareanwendung in den EVU

Ein Teil der Arbeitsgestaltung der Softwareanwendung in den EVU hängt an der Softwaregestaltung. Der Abschnitt nimmt diesen Teil in den Blick, der sich aus dem Verhältnis der Arbeitsprozesse von Softwaregestaltung und Softwareanwendung ergibt. Er lässt sich mithilfe der Kategorien Einfluss und Konflikt analysieren. Anders als im letzten Punkt 8.4. geht es dabei nicht um einzelne Gruppen von Beschäftigten. Es geht um das Verhältnis zweier Arbeitsprozesse zueinander, über das nicht nur die operativ Software-gestaltenden oder -anwendenden entscheiden, sondern in den Fällen, wo Hierarchien vorliegen, das Management strategische Entscheidungen fällt, Ziele vorgibt, Ressourcen zur Verfügung stellt und eigene Anforderungen einbringt.

Der **Einfluss** beider Arbeitsprozesse aufeinander betrifft vier Aspekte:

- Kontrollverhältnis: Wird der Arbeitsprozess der Softwareanwendung jenem der Softwaregestaltung untergeordnet? Werden Softwareanwendung und -gestaltung z.B. getrennt in zwei verschiedenen Organisationen kontrolliert?
- Reorganisation: Wird der Arbeitsprozess der Softwareanwendung zum Zwecke der Softwaregestaltung reorganisiert?
- Ziele: Wie verhalten sich die jeweiligen Ziele von Softwareanwendung und -gestaltung zueinander und wer kann sie festlegen (z.B. individuell etwas für einen Anwendungsbereich zu entwickeln oder eine skalierbare Standardsoftware; die Anwendungsorganisation zu reorganisieren oder nicht)?
- Partizipation: Wer gestaltet wie mit? Wer aus der Softwareanwendung hat Zugang zur Softwaregestaltung? Welche Rollen nehmen Expert:innen ein und wie verhalten sich Anwendenden- und Unternehmens- bzw. Managementanforderungen zueinander?

Für zwei der genannten Aspekte des Einflusses der Softwaregestaltung auf die Softwareanwendung diskutiert der Abschnitt folgende Thesen:

1. Für die Reorganisation deutet sich in einigen EVU an, dass sie sich auf eine bestimmte Form der Organisation zubewegen: eine auf die Softwaregestaltung ausgerichtete softwaretechnische Prozessorganisation. Eine solche Organisation bedeutet nichts anderes, als dass die Softwaregestaltung die Software und die Organisationen eines End-to-end-Prozesses übergreifend und ohne die Behinderung durch Fachbereichsgrenzen ändern kann.
2. Hinsichtlich der Partizipation stellt sich die Frage, inwieweit die Anwendenden an der Gestaltung der Software beteiligt sind und ob sich die Fälle Partizipationstypen zuordnen lassen.

Die **Konflikte** zwischen Gestaltung und Anwendung sind entweder innerhalb (wie je-ne zwischen Management- und Anwendendenanforderungen) oder außerhalb der EVU (wie z.B. zwischen individuellen Alleingängen einzelner EVU und kooperativem Standard mehrerer EVU). Dabei zeigen die Fallstudien, dass Betriebsräte innerhalb der EVU bei der soziotechnischen Arbeitsgestaltung insofern mitbestimmen, indem sie intervenieren, aber nicht aktiv gestaltend sind. Sie spielen bei Fragen der Partizipation insofern eine Rolle, als sie bei Rahmenbedingen mitentscheiden und weniger bei der inhaltlichen Gestaltung der Software (z.B. wenn ein Betriebsrat einfordert, dass die Softwaregestaltung Anforderungen der Anwendenden berücksichtigt).

Dieser Abschnitt geht in den folgenden Punkten auf die Unterschiede zwischen den Fallstudien ein, stellt die Fälle mithilfe der Kategorien Einfluss und Konflikt einzeln dar und fasst die Ergebnisse zusammen.

8.5.1. Soziotechnische Arbeitsgestaltung: zwischen Abhängigkeit und Unabhängigkeit

Die Fälle lassen sich dahingehend unterscheiden, ob die EVU unabhängig oder abhängig von IT-DL oder Softwarefirmen die Arbeit der Softwareanwendung gestalten können.

Welcher Typ in einem Fall vorliegt, ergibt sich aus der soziotechnischen Konstellation und dem Arbeitsprozess der Softwaregestaltung.

Bei der **unabhängigen** soziotechnischen Arbeitsgestaltung ist aufgrund der Arbeitsteilung die Softwaregestaltung Teil der anwendenden Organisation (EVU) und der **Einfluss** auf die soziotechnische Arbeitsgestaltung durch die Softwaregestaltung umfasst nicht nur die Software, sondern auch den anzuwendenden Arbeitsprozess. Die **Konflikte** bei der Arbeitsgestaltung sind rein intern. Je nach Grundkoordination bestehen dann diese internen Konflikte in Hierarchien, Märkten oder Netzwerken.

Bei einer soziotechnischen Arbeitsgestaltung vom Idealtyp **abhängig** reduziert sich der **Einfluss** der Softwaregestaltung auf die angewendete Software und ist nur über Externe möglich (IT-DL, Softwarefirmen). Das liegt an der Arbeitsteilung, bei der sich die EVU ausschließlich auf die Softwareanwendung konzentrieren. Die **Konflikte** in der Arbeitsgestaltung durch Softwaregestaltung existieren entsprechend vor allem mit den externen Softwaregestaltenden. Je nach Grundkoordination bestehen dann diese Konflikte in Hierarchien, Märkten oder in Netzwerken.

Tabelle 25: Idealtypen unabhängige und abhängige soziotechnische Arbeitsgestaltung

Typ	Einfluss der Softwaregestaltung	Konflikte
unabhängig	intern: Software und Organisation	intern
abhängig	extern: nur auf Software	extern

Anhand von vier Unterkategorien lässt sich der Einfluss zwischen Softwareanwendung und -gestaltung näher beschreiben: Kontrollverhältnis, Reorganisation, Ziele und Partizipation.

Was das Verhältnis der Arbeitsprozesse der Softwareanwendung und Softwaregestaltung zueinander betrifft, ist bei einer **unabhängigen** Arbeitsgestaltung das **Kontrollverhältnis** beider so, dass die Softwaregestaltung die Softwareanwendung kontrollieren kann, was sich z.B. darin niederschlägt, dass Softwareanwendende oder deren Führungskräfte Veränderungen aufgrund der Softwaregestaltung nicht verhindern können. Zudem ist eine wechselseitige **Reorganisation** von Softwaregestaltung, Software und Softwareanwendung möglich. Bei den **Zielen** kann das EVU sowohl über jene der Softwareanwendung als auch der -gestaltung entscheiden. Ebenso kann bei der **Partizipation** das EVU entscheiden, wer an der Softwaregestaltung teilnimmt, und es ist ein direkter Einbezug der Anwendenden möglich.

Was das Verhältnis der Arbeitsprozesse der Softwareanwendung und Softwaregestaltung zueinander betrifft, ist bei einer **abhängig** Arbeitsgestaltung das **Kontrollverhältnis** beider so, dass sie sich nicht gegenseitig kontrollieren. Die Softwaregestaltung kann bei der -anwendung keine Veränderungen durchsetzen. Auch eine **Reorganisation** von Softwareanwendung und -gestaltung ist nur getrennt möglich. Die **Ziele** beider Arbeitsprozesse sind getrennt: Jene des Arbeitsprozesses Softwaregestaltung sind auf den Softwarereproduzenten (IT-DL oder Softwarefirma) ausgerichtet und die vom Arbeitspro-

zess der Softwareanwendung auf das EVU. Die **Partizipation** konzentriert sich auf Fach-expert:innen und nicht auf Anwendende bzw. entscheidet nicht das EVU darüber, wer mitgestaltet.

Tabelle 26: Idealtypen unabhängige und abhängige soziotechnische Arbeitsgestaltung – Unterkategorien

Typ	Kontrollverhältnis Softwareanwendung (SA) – Softwaregestaltung (SG)	Reorganisation SA – SG	Ziele SA – SG	Partizipation SA an SG
unabhängig	Kontrolle SA durch SG möglich	Software, SG und SA im Wechselspiel möglich	beides auf EVU-Nutzen ausgerichtet	direkter Einbezug Anwendende möglich
abhängig	Kontrolle SA – SG: nur getrennt möglich	nur getrennt möglich	Ziel SG entscheidet Softwarefirma oder IT-DL	(Branchen-)Fachleute, anwendende Organisation entscheidet nicht

Sind nun die einzelnen Fälle Beispiele für eine unabhängige oder abhängige soziotechnische Arbeitsgestaltung der Softwareanwendung? Antwort darauf geben die Falldarstellungen und die Zusammenfassung am Schluss dieses Abschnitts.

8.5.2. Darstellung der Fallstudien

Das Verhältnis der Arbeitsprozesse von Anwendung und Gestaltung stellt dieser Abschnitt für die sieben Fallstudien jeweils anhand der Kategorien Einfluss (inkl. ihrer vier Unterkategorien) und Konflikt dar.

8.5.2.1. INTERN1: iterativer soziotechnischer Wandel eines Fachbereichs, unabhängig

Es hat Folgen für die soziotechnische Arbeitsgestaltung des Anwendungsbereichs der Software, dass das EVU in der Fallstudie die Möglichkeiten der individuellen Softwaregestaltung dezentral für den Fachbereich Instandhaltung nutzt: Dadurch kann das EVU unabhängig Arbeit via Software gestalten, d.h. den Arbeitsprozess der Softwareanwendung verändern. Das zeigt sich erstens am Einfluss. Denn das EVU hat sowohl Einfluss auf die Gestaltung der Software als auch auf die anwendende Organisation. Zudem kann das EVU eigenständig über die Ziele sowohl von Softwaregestaltung als auch Softwareanwendung entscheiden. Der Einfluss erlaubt zuletzt eine direkte Partizipation der Anwendenden – wenn auch begrenzt. Zweitens verlaufen die Konflikte zwischen den beiden Arbeitsprozessen innerhalb des EVU (zwischen Fachbereich und IT-Abteilung), wodurch es diese eigenständig lösen kann. Allerdings muss sie diese Konflikte innerhalb bestehender Hierarchien lösen, in denen das Management das Sagen hat.

Die unabhängige Arbeitsgestaltung zeigt sich beim Einfluss auf vierfache Weise. Erstens zeigt sie sich am **Kontrollverhältnis**: Die Softwaregestaltung hat die Kontrolle über die energiewirtschaftlichen Prozesse, weil die Anwendenden zwar die Möglichkeit haben, an der Gestaltung mitzuwirken, aber nicht, diese aufzuhalten, und weil sie grundlegende Vorgaben (wie die Arbeitssteuerung über mobile Endgeräte) nicht ändern können. Das zeigt sich auch daran, dass die vier Product Owner:innen, die jeweils u.a. für unterschiedliche Funktionalitäten der Instandhaltungssoftware zuständig sind, die bestehenden Hierarchien aus Fachbereichen und IT nicht berücksichtigen müssen, was zudem den Entscheidungsaufwand verringert:

»Unsere POs, die können Dinge entscheiden. Man muss nicht wegen jedem blöden Knopf zu drei Chefs rennen und sagen: Ist der jetzt grün oder blau? Relativ umfangreiche Dinge entscheiden wir einfach selbst und machen das und das geht auch ganz gut.« (Programmierer)

Zweitens gestaltet das EVU nicht nur eine individuelle Software, sondern **reorganisiert** die Arbeit in der Softwareanwendung. Zum einen tut es das, weil das Gestaltungsnetzwerk aus Anforderungsmanagenden und Key User:innen Teil der Fachbereiche geworden ist. Zum anderen ändert der Arbeitsprozess der Softwaregestaltung iterativ den Arbeitsprozess der anwendenden Instandhalter:innen und Dispatcher:innen, indem er diesen nicht nur abbildet, sondern in Frage stellt und ändert. So gibt es z.B. die Position des Meisters nicht mehr. Wechselseitig entstehen eine individuelle Software und eine veränderte anwendende Organisation. Mittlerweile hat sich durch das iterative Vorgehen eine Sättigung an möglichen Anforderungen vor allem bei der zentralen App für die Arbeitssteuerung der Monteur:innen eingestellt, weswegen neue Bereiche wie die Arbeit der Dispatcher:innen in den Blick genommen werden.

Drittens hat sich das EVU zum Ziel gesetzt, die Software ausgehend vom eigenen, althergebrachten Status quo der Softwareanwendung zu gestalten. Das heißt, das EVU hat als Ziel, die Anwendung inkrementell zu verändern. Es dockt dabei zwar an das Standardpaket von SAP an, kann dabei aber seine eigenen Ziele verfolgen, weil es die individuelle Erweiterung unabhängig von SAP gestaltet.

Zuletzt zeigt sich die unabhängige soziotechnische Arbeitsgestaltung durch die Softwaregestaltung an der **Partizipation**. Denn das EVU kann selbst entscheiden, wer mitgestaltet. Ausschließlich eigene Mitarbeitende gestalten mit. Anwendende können moderiert über Anforderungsmanagende oder Product Owner:innen Anforderungen aufgeben. Das heißt nicht, dass die internen Hierarchien nicht mehr gelten. Die Managementziele bleiben unangetastet. Die Beteiligten weichen nicht von Zielen wie der Kostenoptimierung ab. Die Partizipation durch Anwendende und deren Anforderung sind vor allem nützlich und steigern die Akzeptanz, ohne die Ziele des Managements in Frage zu stellen. Die Partizipation war vom Betriebsrat gewünscht, aber auch Teil eines bewussten Change Managements, wodurch das Management Konflikte mit Beschäftigten vermeiden wollte und auch vermieden hat.

Der kontinuierliche Fortschritt bei der Entwicklung hängt davon ab, dass die Softwareanwendung und -gestaltung durch die bestehende Konstellation entstehende **Konflikte** löst. Diese sind in dem Fall rein intern und so kann das EVU sie eigenständig lösen.

Sie zeigen sich zum einen an der zwiespältigen Lage von Product Owner:innen und Anforderungsmanagenden, weil Erstere in der IT-Abteilung, Letztere im Fachbereich angesiedelt sind, Erstere Anforderungen des Managements aufnehmen und Letztere jene aus dem Fachbereich. Zum anderen zeigen sich die Konflikte in der Zusammenarbeit mit dem Betriebsrat.

Die Fachabteilung Instandhaltung ist nur Auftraggeber und nicht steuernd. Der Haupt-Product-Owner aus der IT-Abteilung entscheidet über Ressourcen wie Budget und Priorisierung. Weil nun das obere Management über den Haupt-Product-Owner Anforderungen einbringt und unterschiedliche Vorstellungen über die Priorisierung größerer Themen bestehen, kommt es zu Konflikten. Die Anforderungsmanagerin aus dem Fachbereich hat es schwer:

»Also, so wie wir heute die Hierarchien haben und leben, könnte ich das nie und nimmer durchsetzen, wenn ich jetzt da eine andere Vorstellung hätte.« (Anforderungsmanagerin)

Bei diesem Konflikt zwischen Unternehmens- und Anwendendensicht hat das letzte Wort der Haupt-Product-Owner. Nur aufgrund guter Kontakte kann der Fachbereich seine Vorstellungen einbringen, trotz bestehender Hierarchien zwischen den Product Owner:innen aus der IT-Abteilung und dem Fachbereich. Letztendlich hätte der Fachbereich gern die allgemeine Entscheidungshoheit (über Priorisierung, Budget, Aufgabenverteilung). Es ist aber noch offen, ob er diese in Zukunft bekommen wird.

Für den Betriebsrat bietet die unabhängige Arbeitsgestaltung durch die interne Softwaregestaltung mehrere Vorteile. Seine Forderung nach stärkerer Partizipation der Monteur:innen wurde umgesetzt. Er kann bereits im Lauf der Entwicklung darauf achten, dass keine individuelle Leistungskontrolle möglich wird. Einige der Befragten (Programmierer, Architekt) berücksichtigen deswegen bereits bei der Konzeption, dass sie auf die Interessen des BR eingehen. Zudem ist er gut informiert, weil er in den entsprechenden Treffen wie andere Stakeholder der Softwareentwicklung dabei ist. Andererseits gesteht er ein, dass er auf bestimmte Entscheidungen, die dort getroffen werden, keinen Einfluss hat. Der Spielraum des BR, was die Gestaltung anbelangt, ist auch bei dieser internen Softwaregestaltung beschränkt.

8.5.2.2. INTERN2: fachbereichsübergreifende Softwaregestaltung, unabhängig

Wie bei INTERN1 kann auch in diesem Fall das EVU unabhängig die Software und damit den eigenen Arbeitsprozess der Softwareanwendung gestalten. Jedoch ist der Einfluss der Softwaregestaltung auf die Softwareanwendung geringer. Zum einen liegt das daran, weil sie nur eine der Softwarelösungen, welche die Anwendenden verwendet, gestaltet. Zum anderen ist allein die Software im Fokus und weniger die Anwendungsorganisation insgesamt. Das liegt vor allem daran, dass die Softwaregestaltung die Organisation mehrerer Fachbereiche verändern müsste – anders als bei INTERN2. Es gibt aber die Idee, die Team- und Abteilungssilos aufzulösen und ein integriertes Prozessteam für die Softwaregestaltung zu schaffen (siehe dazu die Diskussion zur softwaregestaltenden Prozessorganisation 8.5.3.4). In diesem Fall verlaufen die Konflikte vor allem zwischen

den Fachbereichen. Allerdings kann das EVU diese, wie auch jene mit dem Betriebsrat, eigenständig lösen, weil es interne Konflikte sind.

Auch wenn die Softwaregestaltung insofern Einfluss auf die Anwendung hat, als dass sie iterativ die Software ändert, gibt es Unterschiede zu INTERN1. Prinzipiell ist der Spielraum geringer als bei INTERN2, weil das EVU keine eigenständige Erweiterung programmiert, sondern den SAP-Standard anpasst. Das **Kontrollverhältnis** ist anders als bei INTERN1, weil die Softwaregestaltung dem Arbeitsprozess der Softwareanwendung hierarchisch nicht übergeordnet ist, sondern mit den Hierarchien der Anwendung und Abteilungsgrenzen zu kämpfen hat. Das EVU **reorganisiert** die Softwareanwendung nur dahingehend, dass es dezentrale IT-Teams in den Fachbereichen und Gestaltungsnetzwerke für die Anforderungsrunden etabliert hat. Tiefergehende organisatorische Veränderungen, wie die IT- und Fachabteilung aufzulösen und interdisziplinäre Teams zu bilden, meidet das EVU. Das EVU setzt die **Ziele** der Softwaregestaltung nur teilweise selbst. Was den softwaretechnischen Zuschnitt anbelangt, ist sie von SAP abhängig, das einen Standard anstrebt. Was die organisatorische Ausrichtung betrifft, kann das EVU selbst entscheiden, vom Status quo aus zu gestalten und nicht die komplette Organisation in Frage zu stellen. Doch wie bei INTERN1 erlaubt der Einfluss eine direkte **Partizipation** der Anwendenden, auch wenn sich hier mehrere Fachbereiche einigen müssen und damit die Fachbereiche voneinander abhängig sind. Allerdings meint ein befragter Anwender, er würde in Entscheidungsprozesse über Anforderungen nicht eingebunden, vor allem bei solchen, bei denen es um die Usability geht und darum, die Anwendung zu vereinfachen. Für ihn sind Priorisierungen nicht nachvollziehbar und was aus welchem Bereich umgesetzt oder nicht umgesetzt wird.

Ein Beispiel für den mangelnden Einfluss der Softwaregestaltung, was Reorganisation und Kontrollverhältnis aufgrund einer fehlenden prozess- bzw. fachbereichsübergreifenden Sicht anbelangt, ist der Fachbereich Netzanschluss. Dort gibt es zwar eine dezentrale Fachbereichs-IT, die sich um die Digitalisierung des Fachbereichs kümmert. Sie ist aber hierarchisch den anderen (Nicht-IT-)Teams des Fachbereichs gleichgestellt und nicht direkt bei der Bereichsleitung angesiedelt. Sie ist im Team der Regionalleitung. Diese ist disziplinarisch verantwortlich, müsste Entscheidungen fällen und Aufträge vergeben, was die Softwaregestaltung anbelangt. Die Regionalleitung macht das jedoch nicht:

»[D]. h. Prozesse, die eigentlich waagerecht laufen, die werden dauernd durch irgendwelche Hierarchien getrennt und der Leiter sagt halt: >Nicht mein Bier. Ich kümmere mich nicht um die Schnittstelle, weil mir tut es ja nicht weh.< Also, es fehlt die übergeordnete Verantwortung für einen Prozess aus Kundensicht oder aus Dienstleistersicht. Wir kucken eigentlich immer nur nach: Wie sind wir strukturiert aus der Historie?« (Product Owner Netzanschluss)

Ein weiteres Beispiel für den geringen Einfluss der Softwaregestaltung auf die Softwareanwendung aufgrund der Abteilungsgrenzen sind die unterschiedlichen Grade der Automatisierung. Unter anderem stellen die Schnittstellen zwischen Teams, die mit dem Modul Work-Management bzw. Auftragsverarbeitung arbeiten, einen Bruch dar, der nicht digital überbrückt und damit automatisiert ist. Bereiche wie Zählerwesen oder Netz-

planung haben jeweils für sich eine eigene interne Organisation. Sie haben z.B. unterschiedlich viel Budget für IT. Das erschwert ein einheitliches Vorgehen. Jeder einzelne Bereich kann sich für sich selbst optimieren und denkt erst einmal an die Veränderungen bei sich. Um den gesamten End-to-end-Prozess der Auftragsverarbeitung zu optimieren, müssten alle Nachbarabteilungen mitmachen und Teamgrenzen überwunden werden. Das Silo-Denken verdeutlicht der Umstand, dass aktuell niemand den gesamten End-to-end-Prozess Netzanschluss auswerten kann, weil er sich auf mehrere Abteilungen und Systeme erstreckt.

Doch hat sich durch die gemeinsame Anforderungsrunde zumindest die Wahrnehmung auf die Organisation und die Gestaltungsmöglichkeiten verändert, indem die Beteiligten vom gemeinsamen (Software-)Prozess aus auf die Firma blicken:

»Ja, den Eindruck hatte ich auch, dass praktisch durch diese Software, diese alten Strukturen auf einmal feststellen: Wir haben ja was gemeinsam. Was vorher halt einfach nicht war, weil man fachlich getrennt war, weil es eben diese Fachexperten gab für Montage oder Hausanschluss oder was weiß ich, was es da alles gibt. Und jetzt haben die aber doch irgendwie so bestimmte Sachen gemeinsam oder müssen sich jetzt versuchen abzustimmen und das passt nicht mehr zu den alten Strukturen. Das ist zumindest den Eindruck, den ich habe.« (Anforderungsmanager)

Neben den organisatorischen Brüchen innerhalb und zwischen den Fachbereichen und Abteilungen beschränken die Vielzahl verschiedener Softwarelösungen den Einfluss des Arbeitsprozesses der Softwaregestaltung. Der befragte Anwendende aus dem Fachbereich Netzanschluss verwendet zwei Systeme und zwei Oberflächen, hinter denen zwei getrennte IT-Bereiche stecken, die jeweils getrennt Anpassungen vornehmen. Das heißt, der Arbeitsprozess der Softwaregestaltung in Form der Anforderungsrunde betrifft nur eine der angewendeten Softwarelösungen.

Es war eine Voraussetzung für die Etablierung des Softwaregestaltungsprozesses, dass zwischen den Fachbereichen und der IT-Abteilung keine grundsätzlichen **Konflikte** vorhanden waren (siehe die Entstehungsgeschichte der Abstimmungsrunde unter 8.3.2.2). Inhaltliche Interessengegensätze scheinen die Fachbereiche in der zentralen Abstimmungsrunde und dezentral in ihren Teams nun konstruktiv zu verhandeln.

Die Befragten sehen den **Betriebsrat** nicht als inhaltlich gestaltend, sondern intervierend. Bei der Anforderungsrunde macht er nicht mit. Der Betriebsrat prüft vielmehr allgemein neue Lösungen, ob sie eine individuelle Leistungskontrolle ermöglichen. Bei Prozessoptimierungen muss das Management ihn einbinden. Manche aus der IT-Abteilung sehen, so ein Befragter, den Betriebsrat als Behinderung an, weil man ihm die Lösungen vorschlagen und mit ihm diskutieren muss. Für einen anderen Befragten hat er bisher in der Softwaregestaltung noch keine Rolle gespielt.

8.5.2.3. KOOP1: zentraler Standard oder dezentral individuell – Wo endet die Kooperation der EVU?

Bei der soziotechnischen Arbeitsgestaltung sind die EVU in diesem Fall größtenteils abhängig. Das liegt zum einen daran, dass mehrere EVU gemeinsam vermittelt über ein IT-DL kontinuierlich und kooperativ verhandeln, was sie als Standard und was sie individuell

duell gestalten. Der Einfluss dieser zentralisierten Softwaregestaltung auf die Softwareanwendung beschränkt sich im Wesentlichen auf die Software selbst. Die EVU entscheiden für sich, wie sie sich organisieren. Das bedeutet in diesem Fall, von ihrem jeweiligen Status quo ausgehend und sich nicht radikal auf die Softwaregestaltung ausrichtend. Das gemeinsame Ziel, einen Standard zu gestalten, widerspricht in manchen Fällen den Zielen einzelner EVU, individuell ihre Softwareanwendung zu optimieren. Doch lässt die zentralisierte Verhandlung zu, dass EVU zumindest Teile ihrer Software unabhängig von anderen gestalten. Inhaltliche Konflikte verhandeln EVU und IT-DL zwar im gemeinsamen Anforderungsmanagement. Was die Softwaregestaltung allgemein und damit die soziotechnische Arbeitsgestaltung betrifft, sind einzelne EVU durch die Marktbeziehung abhängig vom IT-DL und den anderen EVU.

Bei KOOP1 besteht durch das **Kontrollverhältnis** keine Unterordnung der Anwendenden unter die Bedarfe der Softwaregestaltung. Sie sind vielmehr einem Softwarestandard untergeordnet: dem von SAP und dann noch dem industriespezifischen, den die EVU kooperativ gestalten. Die EVU kontrollieren die Softwareanwendung unabhängig von der Softwaregestaltung dieses Standards. Bei jenen EVU, die verstärkt dezentral selbst Software gestalten, gibt es Fälle, die den Arbeitsprozess der Softwareanwendung an die Bedürfnisse der Softwaregestaltung ausrichten.

Um zumindest Einfluss auf die Software zu haben, haben EVU im Zeitverlauf eigene Strukturen zur Softwaregestaltung ab- und wieder aufgebaut. In EVU3 und EVU2 wurden 2020 neu Stellen geschaffen, um intern Synergien zwischen mehreren Fachbereichen in der Gestaltung zu erkennen. Im EVU2 existiert ein eigenes Team, das sich mit dem Thema Digitalisierung beschäftigt (Strategien entwickeln, eigene Fachkräfte für IT-Projekte etc.). Das ist für den Digitalisierungsmanager der Firma Teil einer Wellenbewegung:

»Also, so das Thema Make-or-Buy oder zentral, dezentral. Also, was habe ich zentral und was habe ich dezentral? Das ist so kurvenartig. So alle zehn Jahre schwingt das um. Wir müssen wieder alles dezentralisieren hin zu zentralisieren. Wahrscheinlich ist in zehn Jahren wieder alles zentral bei uns. Und, ja, ich glaube, da gibt es keine feste Kurve, wo sich das hin entwickelt.« (Digitalisierungsmanager EVU2)

Auch andere EVU verzichten auf Synergien durch die Kooperation zugunsten individueller Gestaltungswege. So hat das EVU1 eine andere Software für das Energiedatenmanagement und die Instandhaltung, um diese alleine, unabhängig von den anderen EVU, die mit dem IT-DL zusammenarbeiten, zu gestalten.

Aufgrund der abhängigen Arbeitsgestaltung ist die **Reorganisation** nur getrennt für Softwareanwendung und -gestaltung möglich. Es gibt zwar individuelle Anpassungen bei KOOP1, jedoch führen sie nur vereinzelt zu Reorganisationen: Ein EVU von KOOP1 betreut die Bereiche für Geschäfts- und Privatkundschaft nicht mehr durch getrennte IT-Teams, sondern gemeinsam durch ein IT-Team, um die Prozesse beider Bereiche zu vereinheitlichen. Die Prozesse zur Kundschaft sollen von den Softwaregestaltenden end-to-end überblickt und optimiert werden können. Der Anstoß für die Reorganisation war, dass die verschiedenen Bereiche ähnliche Anforderungen an die Softwaregestaltung gestellt haben:

»Und da ist halt dann aufgefallen, dass es teilweise bzw. eigentlich nicht teilweise, sondern der übergreifende Teil immer der war, dass es die gleichen Anforderungen gibt, die aber letztendlich losgelöst voneinander umgesetzt worden sind oder genau der andere Fall, dass Team A irgendetwas umgesetzt hat, was bei Team B dazu führte, dass irgendwann nicht mehr funktioniert hat. Da ist man halt zum Entschluss gekommen, dass man eigentlich diese beiden Teams oder Funktionen zusammenbringen muss, um letztendlich dort Synergien zu entwickeln.« (Prozessmanager EVU3)

Hier haben Beschäftigte und Management intern die softwaretechnischen Gestaltungsmöglichkeiten erkannt und reagieren darauf mit einer Reorganisation.

Die Abhängigkeit bei der soziotechnischen Arbeitsgestaltung zeigt sich auch an den Zielen. Bei KOOP1 haben vor allem die kleinen EVU das Ziel, eine Standardware zu nutzen, die günstig sein soll. Allgemein behalten die EVU ihre fachlichen Abteilungen und hierarchisch gegliederten Strukturen bei. Wie auch schon bei den anderen Dimensionen des Einflusses der Softwaregestaltung auf die Softwareanwendung nutzen einzelne EVU die Möglichkeit, mehr individuell selbst zu gestalten und sowohl was die Software als auch die eigene anwendende Organisation betrifft, unabhängig zu entscheiden.

Die Partizipation erfolgt beim kooperativen Anforderungsmanagement bürokratisch reglementiert. Es ist ein formalisierter Prozess, in dem EVU im Anforderungsmanagement Anforderungen verhandeln und der Standard einen Konsens darstellt. In einem EVU prüft das Management intern noch jede Anforderung, die zum IT-DL in das Anforderungsmanagement wandert. Oder es legt z.B. das IT-Budget für Anforderungen fest und entscheidet, welche Anwendenden mitgestalten.

Der Fall ist ein Beispiel für eine abhängige soziotechnische Arbeitsgestaltung durch Softwaregestaltung, weil die **Konflikte** extern sind. Konflikte über die inhaltliche Gestaltung lösen die EVU zwar über das IT-DL im Ablauf des Anforderungsmanagements. Wenn kein Kompromiss gefunden wird, sieht die Softwarearchitektur und die Kooperationsvereinbarung vor, dass einzelne EVU unabhängig davon gestalten können. Das Kernproblem der Softwaregestaltung via Marktbeziehung (wenn auch kooperativ) bleibt bestehen: Zwischen Kooperation, Innovation und Kostenkalkül müssen EVU und IT-DL eine gemeinsame Basis finden. Die Folgen für die Gestaltung sind, dass die Kooperation beschränkt bleibt und teilweise von EVU in Frage gestellt wird.

Aus Sicht von möglichen Synergien in der Softwaregestaltung wäre es ein großer Vorteil, wenn die EVU besser kooperieren und ihre Unabhängigkeit dafür aufgeben würden. Doch ist es für das IT-DL schwierig, etwas kooperativ zu bewegen, wenn die EVU intern sich nicht über ihre IT-Strategie einig sind. Einen konkreten Plan oder Vorgehen, um eine gemeinsame Strategie zu entwickeln, gibt es aktuell nicht. Das IT-DL muss sich mit den EVU individuell zusammensetzen. Die EVU haben z.B. eigene Strategien, welche Softwarepakete sie einsetzen (auch von anderen IT-DL). So gibt es u.a. beim CRM-Einsatz keine Einigkeit im Verbund, weswegen EVU3 seinen eigenen Weg geht. Ein Mindestmaß an Kooperationsbereitschaft besteht jedoch. Dies zeigt sich im Anforderungsmanagement, wo sich die EVU zumindest in einigen strategischen Fragen, wie z.B. der Umstellung auf eine neue Version des ERP-Systems, einigen können.

Neben den Konflikten darüber, was Standard und was individuell ist, zeigt sich die Abhängigkeit der EVU in der Umsetzung der Softwaregestaltung. Das betrifft die Dau-

er, den Support und die Qualität der Umsetzung. Das hat auch Folgen für die Anwendenden, weil diese in ihrer Arbeit von einer funktionierenden Software abhängen. Für viele Befragte ist die Reaktionszeit und der Service des IT-DL nicht immer befriedigend. Das IT-DL verweist manche EVU bei Fragen an die Servicehotline oder sie müssen auf Rückrufe warten. Die EVU erleben sich als Teil einer größeren Kundschaft. Manche Anforderungen erledigt das IT-DL trotz existierender regulatorischer Fristen zu langsam. Es herrscht Misstrauen, was die Aufwandsschätzung anbelangt, weswegen einige EVU intern wieder Know-how haben, um sie prüfen zu können.

Für die **Betriebsräte** in den EVU ergeben sich durch die Zusammenarbeit mit einem IT-DL, die Projektarbeit und den stetigen Wandel der IT-Systeme verschiedene Einflussmöglichkeiten. Allerdings betrifft das weniger die inhaltliche Softwaregestaltung. Diese geht nicht darüber hinaus, eine Leistungskontrolle zu verhindern. Bei EVU4 ist der Betriebsrat dank einer Betriebsvereinbarung für IT-Projekte in diese eingebunden. Er sitzt in allen Lenkungsausschüssen und ist deshalb gut informiert. Er darf bei großen Projekten auch bei der Beraterauswahl mitentscheiden. Bei mehreren IT-Projekten hat er die Leistungs-/Verhaltenskontrolle geprüft und verhindert. In dem EVU ist der Betriebsrat auf Augenhöhe mit dem Management und fährt mit zu Führungskräfte tagungen, bei denen es u.a. um die Digitalisierungsstrategie geht. Aus seiner Sicht kann er bei der Softwaregestaltung nicht mitmachen, weil ihm das nötige Wissen fehlt und er nur einen eingeschränkten internen Machtbereich hat.

Der Betriebsrat aus EVU2 meint dazu:

»Da sind Leute, die sprechen eine Sprache. Ich kann mich mit denen nicht unterhalten. Also, ich kann mit denen ein Bier trinken. Aber fachlich... Ich kenn kein Wort von denen. Die Rede auch kein Deutsch. Das ist natürlich eine andere Welt für uns, völlig andere Welt.« (Betriebsrat EVU2)

In dem EVU sitzt der BR in einem Gremium, das sämtliche neuen IT-Systeme prüft. Laut dem Applikationsbetreuer von EVU2 muss für jedes neue IT-System eine Betriebsvereinbarung geschlossen werden. Zudem versucht der Betriebsrat von EVU2, das IT-DL zu kontrollieren. Einmal hat er mitgewirkt, einen externen Berater zu tauschen. Er sieht seine Firma in der Verantwortung, die Belegschaft vor negativem Einfluss von außen zu schützen. Gleichzeitig sieht er keinen Grund, nicht auf Wettbewerb zwischen den IT-DL zu setzen.

»Wir haben unseren ganzen IT-Bereich in mehrere Lose verteilt und die werden ausgeschrieben und entweder kriegen sie es oder kriegen es nicht [...] Deswegen: Wir nehmen den, der am besten ist, ganz deutlich.« (Betriebsrat EVU2)

Der Betriebsrat von EVU3 sagt klar, dass er die Software nicht mitgestaltet. Stattdessen hat er Obergrenzen durchgesetzt, was die Arbeitsstunden anbelangt, und auch schon Projekte ausgesetzt, weil die Belastung zu groß wurde. Aus seiner Sicht hat der Datenschützer mehr Mitspracherechte bei IT-Themen als er. Auch er konzentriert sich darauf, eine individuelle Leistungskontrolle zu verhindern.

8.5.2.4. KOOP2: Prekär-kooperativ – wieder unabhängiger vom IT-DL?

Im Gegensatz zu KOOP1 gestalten einige EVU in diesem Fall wieder unabhängig von anderen ihre Software und damit die Arbeit ihrer Anwendenden – auch wenn es, wie bei KOOP1, meist nur die Anpassung oder Erweiterung einer Standardsoftware ist. Statt zentral einen Industriestandard wie bei KOOP1 zu gestalten, gestalten manche EVU das bestehende ERP-System selbst individuell – durch eigene Programmierungen oder Softwarelösungen anderer Softwarefirmen. Doch reizen diese EVU die Möglichkeiten der Arbeitsgestaltung nicht aus, weil sie sich auf die Gestaltung der Software der Anwendenden konzentrieren und die anwendende Organisation unangetastet lassen. Wobei es wie bei KOOP1 Unterschiede gibt, wie stark sich einzelne EVU auf die Softwaregestaltung einlassen und damit auf eine individuelle und unabhängige soziotechnische Arbeitsgestaltung. Ein befragter Sachbearbeiter zeigt die für diesen Fall typischen vielfältigen Formen der Softwaregestaltungen in den EVU in einer Extremform: Er muss mit einer Cloud-Lösung von SAP arbeiten. Das hat zur Folge, dass er und sein EVU komplett abhängig sind, was die Gestaltung der Software anbelangt.

Der Einfluss auf die soziotechnische Arbeitsgestaltung ist in diesem Fall gering, obwohl einige EVU unabhängig Software gestalten. Das ist so in EVU2 (ob bei einem größeren IT-Projekt, einer dezentralen Softwaregestaltung durch den Fachbereich Marktkommunikation oder einer Softwaregestaltung in einzelnen Fachbereichen über deren IT-Koordinierenden) und ebenso in EVU3 der Fall (wo über ein zentrales IT-Team, das eine Entwicklungsplattform betreibt, dezentral Fachbereiche kleinere Apps gestalten). Warum ist der Einfluss gering? In all diesen Beispielen kontrolliert der Arbeitsprozess der Softwaregestaltung nicht jenen der Softwareanwendung. Durch die dezentrale und individuelle Softwaregestaltung wäre zwar ein anderes **Kontrollverhältnis** möglich. Die EVU nutzen es aber nicht. Sie sind intern auf die Softwareanwendung ausgerichtet. Vielmehr geht es erst einmal darum, was sich vor allem bei EVU2 zeigt, intern Strukturen wie eine Stabstelle für Projektmanagement aufzubauen, um eigenständig Software zu gestalten. Zwar sind die IT-Koordinatoren der Fachbereiche im EVU2 für größere Bereiche zuständig. Sie sind aber hierarchisch nicht so angesiedelt, dass sie fachübergreifende Themen vorantreiben oder gar Reorganisationen durchsetzen könnten. Stärker noch als bei KOOP1 zeigt der Fall, dass trotz der dezentralen Softwaregestaltung in den EVU keine grundlegende **Reorganisation** erfolgt. Die bestehende Organisation aus Abteilungen, Teams und Hierarchien bleibt unangetastet. Wie für eine Matrixorganisation typisch, betreiben die EVU temporär IT-Projektarbeit oder arbeiten kontinuierlich mit Standardsoftwarefirmen von Speziallösungen zusammen. Es geht primär darum, Software zu gestalten und weniger die Organisation der Anwendenden auf die Softwaregestaltung auszurichten. Wie auch bei INTERN1, INTERN2 und KOOP1 verwenden die EVU das Standard-ERP von SAP und dieses ergänzende Standardpakete. Sie sind damit einerseits von den **Zielen** der Softwarefirma abhängig, ein Standardprodukt profitträchtig zu verkaufen. Andererseits sind sie unabhängig, wenn es um Anpassungen und Erweiterungen dieser Standardsoftwarelösungen geht. Das nutzen sie für ihre Ziele, die eigene Organisation nicht zu stark ändern zu müssen und die Software selbst gestalten zu können. Die EVU, die wieder eigenständig Software gestalten, können eine **Partizipation** ihrer Mitarbeitenden ermöglichen, wie es für eine unabhängige Arbeitsgestaltung typisch ist. Dabei sind bei EVU2 die internen Hierarchien prägend. Die Anwendenden

können nicht unabhängig von diesen Hierarchien (und Abteilungsgrenzen) die Software und damit einen Teil ihrer Arbeit gestalten. So bespricht die Teamleitung der Marktkommunikation mit der Softwarefirma die Anforderungen, oder IT-Koordinator:innen sammeln Anforderungen aus den Fachbereichen ein.

Eine klare Abhängigkeit in der Arbeitsgestaltung erlebt der befragte Sachbearbeiter von EVU1. Er muss sich einer neuen, von SAP entwickelten Cloud-Lösung für Marktkommunikation unterwerfen. Weil es eines seiner zentralen Werkzeuge für die Arbeit ist, ist er komplett von der Softwarefirma SAP, die sie zur Verfügung stellt, abhängig. Mehr als Fehler aufnehmen und warten, bis sie die Softwarefirma behoben hat, kann er nicht.

In dem Fall verlaufen die **Konflikte** sowohl zwischen IT-DL und EVU als auch innerhalb der EVU, weil diese selbst Software gestalten. Die Konflikte zwischen IT-DL und EVU sind weniger inhaltlicher Natur als vielmehr Beziehungskonflikte, die auf unterschiedlichen Erwartungen beruhen. Die Konflikte zeigen sich vor allem daran, dass sich die beteiligten EVU und das IT-DL nicht auf eine zentrale Gestaltung einigen konnten. Diese Konflikte führten zum Ausstieg einzelner EVU aus der kooperativen Softwaregestaltung oder zur sukzessiven Rückverlagerung der Softwaregestaltung in die EVU.

Trotzdem sind die EVU noch vom IT-DL abhängig, aber weniger inhaltlich und mehr in der Umsetzung. Ein Thema ist die Priorisierung von Anforderungen beim IT-DL. Ein Befragter hat den Verdacht, dass seine Tickets vom IT-DL langsamer bearbeitet wurden, weil sein EVU eine Pauschale bezahlt und ein anderes EVU pro Ticket, wodurch der Anreiz für das IT-DL höher sei, Letzteren schneller zu bedienen. Ein anderer Befragter ist der Ansicht, dass das IT-DL den Personal-Wünschen seiner gesellschaftenden EVU²¹ entgegenkommt (z.B. bekommen diese EVU ihre Wunschkandidaten für Projekte). Ein Befragter des IT-DL sagt, er behandelt alle EVU gleich, egal ob gesellschaftendes EVU oder nicht. Andere Konfliktpunkte sind, dass die EVU meinen, dem IT-DL fehle die Sicht seiner Kundschaft, es sei nicht lösungsorientiert, würde Tickets zu langsam bearbeiten und hätte zu wenige Mitarbeitende.

Innerhalb der EVU fällt auf, dass sie zwar die Möglichkeit hätten, die Konflikte zwischen Gestaltung und Anwendung eigenständig zu lösen. Jedoch zeigt sich, dass der Einfluss der Fachbereiche auf die IT-Abteilung oder der IT-Koordinierenden auf die Fachbereiche gering ist. Dies ist womöglich auch ein Grund, warum keine organisatorischen Änderungen stattfinden und sich manche befragte Softwaregestaltende zwar unzufrieden mit den bestehenden Strukturen zeigen, mit diesen aber leben müssen oder das Unternehmen verlassen (wie das zwei Befragte einige Monate nach den Interviews getan haben).

Der **Betriebsrat** ist, wie in den vorhergehenden Fallstudien auch, intervenierend tätig und hat die Möglichkeit mitzugestalten, weil die Softwaregestaltung wieder stärker intern stattfindet. Der BR von EVU2 hat eine Betriebsvereinbarung IT gemacht. Diese regelt, dass IT-Projektanträge durch den BR genehmigt werden müssen. Zudem achtet der BR darauf, dass keine privaten Daten der Mitarbeitenden zugänglich werden und durch Datenauswertungen des IT-Systems nicht auf einzelne Mitarbeitende und deren Arbeit geschlossen werden kann. Aufgrund der immer vielfältigeren IT-Landschaft würde dem BR mehr IT-Wissen bei der Arbeit helfen.

²¹ EVUs, die Gesellschaftende, d.h. deren Eigentum die IT-DL sind.

8.5.2.5. PAKET: abhängig von der Standardsoftwarefirma

In dem Fall sind die EVU von der Softwarefirma abhängig, was die soziotechnische Arbeitsgestaltung anbelangt. Warum? Zum einen wenden die EVU eine industriespezifische Standard-ERP-Lösung an, die sie nicht ändern können (und wollen). Zum anderen bietet die Software zwar Einstellungsmöglichkeiten. Die gibt aber der Standard vor. Selbst die wenigen EVU, die beim industriespezifischen Standard über Arbeitskreise oder einzelne Projekte mitwirken, setzen einen Standard ein, den alle anderen EVU auch einsetzen. So können die EVU nur den Arbeitsprozess der Softwareanwendung selbst kontrollieren und reorganisieren. Andererseits zeigt der Fall, dass die Einstellungsmöglichkeiten an der Standard-ERP-Software sehr umfassend sind: Wenn die EVU diesen Teil der Softwaregestaltung nutzen, kann dies organisatorische Veränderungen in der Softwareanwendung mit sich bringen, und zwar weitergehender als bei einigen EVU von KOOP1, KOOP2 oder INTERN2 – nämlich auf eine Prozessorganisation hin. Die Abhängigkeit der EVU bei der soziotechnischen Arbeitsgestaltung zeigt sich auch an den Konflikten, bei denen es weniger um die inhaltliche Softwaregestaltung geht als um die Abhängigkeit vom gelieferten Standardprodukt.

Der Einfluss der Softwaregestaltung auf den Arbeitsprozess der Softwareanwendung ist, was das **Kontrollverhältnis** anbelangt, in diesem Fall gering. Die EVU können nur die Softwareanwendung kontrollieren und weniger den Gestaltungsprozess, der außerhalb der EVU stattfindet. Vielmehr prägt die fertige Standardsoftware die Anwendung der EVU. Mit dem geringeren Gestaltungsspielraum (vom Management bis zu den Anwendenden in den EVU) geht eine Unterordnung der Anwendung unter die Standardsoftware einher. Sie dient als Referenz und zur Disziplinierung von Veränderungswünschen seitens der EVU – ob der eigenen Anwendenden oder der Kundschaft. So nutzt EVU4 den Standard dafür, Wünsche seiner Kundschaft zu beschränken, z.B. indem es auf die Grenzen der Abrechenbarkeit komplexer Erzeugungsanlagen durch die Software verweist. Eine Extremform der Unterordnung der Anwendung unter die Standardsoftware ist, wenn die Softwarefirma oder EVU die ERP-Software als Anwendungsplattform einsetzen. Aufgrund des Funktionsumfangs und der hohen Automatisierung der Software steckt so viel Wissen in ihr und die Aufgaben, die sie erledigt, sind so umfangreich, dass sie eine Anwendungsplattform darstellt. Jeden, der die Software besitzt (wie bspw. die Softwarefirma), trennt nicht mehr viel von einem EVU:

»Also theoretisch können wir auch einfach unseren eigenen Stromversorger aufmachen.« (Programmierer Softwarefirma)

So liegt es nahe, dass die Softwarefirma Business Process Outsourcing (BPO) anbietet.

»Es gibt aber auch Unternehmen, gerade so im Bereich der kleineren Lieferanten, so Start-ups, wo das Unternehmen eigentlich aus einem Geschäftsführer, paar Vertrieb-lern und Marketingleuten besteht. Und die gesamte Abwicklung: Abrechnung, Markt-kommunikation, EDM²², Buchhaltung etc. pp. läuft dann oder geht dann über unsere

[BPO-Firmeneinheit]. Da spielen wir sozusagen Stadtwerk.« (Lösungsarchitekt Softwarefirma)

Manche EVU bieten anderen den Betrieb der industriespezifischen ERP-Software an. In einem Fall bedeutet das, dass die anwendenden EVU die Einstellungen, welche das betreibende EVU an der Software vorgenommen hat, 1:1 übernehmen. Das heißt, es richtet sich in der Anwendung nicht nur an der Standardsoftware aus, sondern auch an den Standardeinstellungen des betreibenden EVU.

Von den Extrempfälten abgesehen, haben die EVU Einfluss auf die Einstellungen am Standard und ob sie den Standard einsetzen. In einem Fall hat sich ein Fachbereich gegen ein Modul der Standardsoftware entschieden (auch wenn der befragte IT-Leiter dagegen war). Um Einstellungen vorzunehmen, brauchen EVU Zeit und das entsprechende, qualifizierte Personal. Nicht in allen EVU ist es möglich, sich intern optimal auf die Software auszurichten, weil die Softwarefirma die Updates zu kurzfristig ausliefert. Das heißt, auch hier sind die EVU abhängig von der Softwarefirma. Aus Sicht eines Teamleiters (EVU2) sind sie zu sehr mit den regelmäßigen Updates der Software beschäftigt und haben keine Zeit und kein Geld für Optimierungen. Er wird häufig von Kolleg:innen gefragt, ob er etwas an der Software optimieren könnte. Das würde aus seiner Sicht viel bringen in puncto Effizienz. Ein anderer Befragter, der Teamleiter und Sachbearbeiter zugleich ist, hat auch keine Zeit für Optimierungen, weil sich regelmäßig die Regulierung verändert. Manchmal kommt das Update der Softwarefirma für Änderungen erst einige Tage vor der Frist, in der eine Regulierung in der Software umzusetzen ist. Sie ist dann noch nicht komplett fehlerfrei und so werden Fehler auch noch Wochen nach der Frist behoben. Er hat keine Zeit für Tests und ist froh, wenn er die Arbeit schafft, obwohl sie mit jedem neuen Gesetz mehr wird.

Einige der befragten EVU, welche selbst Einstellungen an der Software vornehmen, konstatieren einen Mangel an Positionen, die übergreifende Veränderungen an der Software vornehmen können:

»Oh, sehr viele. Ja, erst mal... Also aktuell ist es so, dass wir immer noch den Wunsch haben nach einem übergeordneten Projektleiter, der eben genau solche Aufgaben übernimmt. Aktuell versuchen wir das so ein bisschen zu machen, so nebenbei noch, was schwierig ist, weil wir ja auch gar nicht weisungsbefugt sind.« (Anwendungsbelebung EVU3)

Um solche Einstellungen vornehmen zu können, gibt es bei mittelgroßen EVU in begrenztem Umfang **Reorganisationen** in Form von neuen Stellen und einer Prozessausrichtung. Es entstehen neue Aufgaben für Führungskräfte und teamübergreifende Positionen, um für teamübergreifende Prozesse Einstellungen an der Standardsoftware vorzunehmen. So hat EVU3 erst vor ein paar Jahren angefangen, Prozesse aufzunehmen, und die Projektarbeit professionalisiert, d.h. Stellen geschaffen für Projekt- und Prozessmanagement.

»Ich kann mich noch erinnern... vor ein paar Jahren angefangen habe, da war das so: Prozesse, was ist das? Das habe ich ja noch nie gehört. So was braucht doch kein

Mensch.« Und jetzt ist es unumgänglich [...] [H]eute machen wir alles in Prozessen. Und so ist das auch mit Projekten. Früher: »Projekte, was ist das?« (Gruppenleiter EVU3)

Auch EVU2 hat festgestellt, dass Projekte mittlerweile häufig abteilungsübergreifend sind. EVU5 hat eine größere Reorganisation in Richtung Prozessorganisation vorgenommen. Die Organisation teilt sich nicht mehr nach Abrechnungsarten auf (Tarifkundschaft, Sonderkundschaft, Einspeiser etc.), sondern nach Prozessschritten, die sämtliche Abrechnungsarten betreffen (Marktkommunikation, Abrechnung, Zahlungsverkehr). In Terminen der betroffenen Bereiche optimiert das EVU die Gesamtprozesse.

Jedoch steht das **Ziel** der Standardsoftwaregestaltung über den individuellen Wünschen einzelner EVU. Das ist für jene EVU in Ordnung, die vor allem für wenig Geld Software einsetzen wollen. Einerseits erwarten die EVU ein funktionierendes System, mit dem sie ihre Arbeit erledigen können und welches die Regulierung abdeckt:

»Dadurch, dass wir ein Lizenz-System haben, ist der Softwareanbieter dafür verantwortlich, die regulatorischen Bedingungen im System umzusetzen. Also, wenn man einen Software-Dienstleister darauf hinweisen muss, dass er eine regulatorische Umsetzung umsetzen muss, hat er seinen Job verfehlt. [...] [Das] System ist ein rein energetisch wirtschaftliches System.« (Gruppenleiter EDM EVU4)

Andererseits hat das Standardpaket Nachteile, weil die Softwarefirma Anforderungen nur umsetzt, wenn sie auch andere EVU betreffen. Eine individuelle Entwicklung wäre schneller in der Umsetzung:

»Also, es ist schon einfacher für uns, wenn wir die Entwicklungsveränderungen im Haus vornehmen können und das dann einfach umgesetzt wird, als wenn wir darauf angewiesen sind, dass es anderen auch noch so geht. Oder dass erst mal aufwendige Beschreibungen und Tests durchgeführt werden müssen, damit der Softwareanbieter erkennt, wo die To-dos liegen.« (Betriebsrat EVU6)

Einige EVU haben nicht nur das Ziel einer möglichst günstigen Software, sondern wollen auch individuelle Ideen umsetzen und möglichst günstige, individuelle Prozesse haben. Laut dem Gruppenleiter aus EVU3 würden individuelle Entwicklungen helfen, interne Prozesse zu optimieren. Ein Befragter aus EVU1 meint, sie haben gar nicht die Ressourcen, um von einem Standard abzuweichen, und hätten deshalb sämtliche individuellen Entwicklungen beseitigt. Das sieht auch der IT-Leiter aus EVU2 so.

In dem Fall **partizipieren** nur ausgewählte EVU und Fachleute u.a. über Arbeitskreise an der Softwaregestaltung. Branchenfachleute haben ein größeres Gewicht bei der Frage, wie Anforderungen zustande kommen, und weniger die Anwendenden mit ihren individuellen Anwendungskontexten. Partizipation ist nur bei den Einstellungen möglich. Die überlassen die Anwendenden aber lieber anderen:

»Die Anwender sind dann einfach nur die, die sagen: »Ich bin hier zum Abrechnen da und mich interessiert eigentlich nicht, wie das System funktioniert, wie es funktionieren sollte. Für mich muss es laufen und wenn es eine Neuerung gibt, dann muss es mir

jemand sagen.« Also, da muss quasi schon alles vorbereitet sein. Am besten, wenn es eine Neuerung gibt, gibt es eine vorbereitete Doku, die beschreibt, welchen Knopf ich wann drücken muss.« (Anwendungsbetreuung EVU3)

Die Abhängigkeit bei der soziotechnischen Arbeitsgestaltung durch Softwaregestaltung drückt sich in dem Fall vor allem für kleinere EVU darin aus, dass bei inhaltlichen **Konflikten** die Softwarefirma die Oberhand hat. Für viele stellt sich also weniger die Frage, was in den Standard aufgenommen wird. Wenn sie den Standard so nicht wollen, verwenden sie eine andere Software. Neben inhaltlichen Konflikten geht es vielmehr um Leistungen, die über die Softwarelizenz hinausgehen und die EVU erwarten. Wie schon bei KOOP1 und KOOP2 kritisieren EVU die Servicequalität bzw. erwarten einen bestimmten Service: Erstens nehmen die EVU die formale Kommunikation über Ticketsysteme als ineffizient wahr. Zweitens erwarten die EVU, dass die Softwarefirma mehr von sich aus über Möglichkeiten informiert, die die Software bietet. Sie haben intern kein Wissen über die Möglichkeiten der Softwaregestaltung und sind abhängig davon, dass die Softwarefirma darüber informiert.

Wie die EVU haben die **Betriebsräte** wenig Einfluss auf die Softwaregestaltung. Die Konflikte betreffen die Implementierung der Standardsoftware. In EVU6 gibt es eine Betriebsvereinbarung für Softwareanwendung und deren Einführung. Bei IT-Projekten stellt das Management dem Betriebsrat »meistens« (Betriebsrat EVU6) den Ablauf des Projekts und das Schulungskonzept für die Anwendenden vor. Bei der Umstellung auf eine andere ERP-Software (was gerade diskutiert wird) käme es aus Sicht des Betriebsrates zu einer Änderung an der Organisation und dieser ERP-Wechsel wäre damit mitbestimmungsrelevant.

»Und von daher alles, [...] was zu einem auch an Einführung Software bedingter Mittel oder der Digitalisierung der Projekte angeht, unterliegt auch der Mitbestimmung, so dass wir sowieso in letzter Konsequenz immer sagen müssen: ja oder nein. Und da haben wir es einfach für uns so eingeführt, dass der Betriebsrat von Anfang an bei Digitalisierung des Projektes mit dabei ist, damit ich nicht als Bremsklotz hinterher fungiere, sondern als Mitgestalter.« (Betriebsrat EVU6)

Wie weit diese Mitgestaltung geht, konnte das Interview nicht vertiefen.

8.5.2.6. KOOP3: IoT Ko-Produktion – abhängige Standardgestaltung, unabhängige Modulgestaltung

Der Fall ist ein Beispiel dafür, dass Softwaregestaltung nicht immer relevant sein muss für die soziotechnische Arbeitsgestaltung eines Anwendungsbereichs. Zwar liegt in dem Fall die zentralisierte Gestaltung einer Standard-IoT-Software vor. Jedoch hat die dadurch gegebene Abhängigkeit in der Softwaregestaltung geringe Folgen für die soziotechnische Arbeitsgestaltung. Denn es existiert kein Arbeitsprozess der Softwareanwendung. Es ist schlicht keine Anwendungssoftware des täglichen EVU-Geschäfts. Vielmehr stellen die IoT-Software und die mit ihr verbundenen Sensoren eine Erweiterung der bestehenden Dateninfrastruktur innerhalb der EVU dar, als dass sie komplizierte Geschäftsprozesse abbilden.

So spielt beim Einfluss weder das **Kontrollverhältnis** noch die **Reorganisation** der Softwareanwendung eine Rolle. Bei den **Zielen** besteht zwar eine Abhängigkeit zwischen IT-DL und EVU in Bezug auf die IoT-Software. Die Softwarefirma will vor allem ein skalierbares Produkt haben. Doch besteht ein gemeinsames Interesse mit dem IT-DL, neue Anwendungsfelder für IoT zu erschließen und auch außerhalb der Softwarefirma Module entwickeln zu können. Letztendlich lässt die Architektur eine dezentrale Entwicklung in Form von Modulen ebenso zu, wie die IoT-Standardlösung einfach nur ohne tiefergehende IT-Kenntnisse und nach erfolgreicher Implementierung anzuwenden.

Bei der **Partizipation** liefern Expert:innen, die bei der Implementierung der IoT-Lösung mitmachen, die Anforderungen und entwickeln dadurch neue Ideen für Erweiterungen. Das IT-DL gibt die Anforderungen aus Implementierungsprojekten an die Softwarefirma weiter.

Die **Konflikte** bestehen in dem Fall bei der abhängigen Softwaregestaltung zwar zwischen mehreren Organisationen. Doch lösen sie die Beteiligten durch eine kooperative Arbeitsweise. Zudem sind nicht viele Anwendende oder Teams betroffen, weswegen es weniger Konflikte bei der Gestaltung gibt. Nur ausgewählte Mitarbeitende in den EVU nehmen temporär an der Implementierung teil und nutzen dann die Daten aus den Sensoren in ihrer Arbeit. Es gibt jedoch durch die Konstellation angelegte, tiefergehende Konfliktpotenziale, die sich allerdings noch nicht manifestiert haben: Es besteht ein Wettbewerb zwischen IT-DL und Softwarefirma. Denn sowohl Softwarefirma als auch IT-DL führen Implementierungsprojekte durch und das IT-DL bietet den EVU die Einbindung der Daten in deren bestehende Softwarelandschaft an, programmiert selbst Module und verkauft sie.

Der **Betriebsrat** spielt nur beim Thema individuelle Leistungskontrolle und als Stakeholder, der mitgenommen werden sollte, eine Rolle. Das EVU1 hat den Betriebsrat eingebunden und hatte am Anfang der Implementierung einen »relativ intensiven Austausch« (Teamleiter IT). Es ging u.a. darum, dass Daten erhoben werden, ohne dass Mitarbeitende mehr an einen bestimmten Ort fahren müssen, d.h., ihre Tätigkeit hat sich verändert. Der Befragte von EVU2 betont, dass der Betriebsrat in die Projekte eingebunden wird, vor allem wenn die Software Mitarbeiterdaten sammelt. Das IT-DL erklärt im Normalfall dem Betriebsrat der EVU in einem ersten Termin, was gemacht wird. Das reicht meistens aus. Nur in einem Fall, in dem Monteur:innen dauerhaft Daten zur Netzvermessung erhoben haben, kam es zu tiefergehenden Gesprächen, die aber zu einer Lösung führten.

8.5.2.7. STARTUP: Pramat der Softwareentwicklung

Der Fall ist ein typisches Beispiel für den Pramat der Softwareentwicklung: für eine unabhängige, soziotechnische Arbeitsgestaltung durch Softwaregestaltung. STARTUP gestaltet eine individuelle Software, und zwar anders als in anderen Fallstudien nicht ausgehend von einer Struktur, die historisch bedingt rein auf die Softwareanwendung ausgerichtet ist. Stattdessen dient die gesamte Organisation dazu, eine Software für den Anwendungsbereich zu gestalten. Die Softwareanwendung ist all das, was die entwickelte Software nicht erledigen kann. Wobei sich iterativ mit der Software die Anwendung verändert. So hat allein STARTUP Einfluss auf Software und Organisation.

Der unabhängige Einfluss zeichnet sich durch ein **Kontrollverhältnis** aus, bei dem die Anwendung klar der Softwaregestaltung untergeordnet ist und keine Hindernisse z.B. durch Hierarchien bestehen, die Absprachen mit der Anwendung notwendig machen würden. Durch den Primat der Softwaregestaltung gab es vorher keine Anwenderorganisation ohne Software, sondern die Arbeit der Anwendenden ergibt sich seit Gründung der Firma daraus, was für eine Software existiert und wie sie die Organisation weiterentwickelt. Größere **Reorganisationen** sind aus Sicht der Softwaregestaltung nicht notwendig, weil die Organisation von Beginn an auf diese ausgerichtet ist. Kleinere Veränderungen im Wechselspiel von Software und Anwendung(-organisation) finden aber statt. Diese Möglichkeit eigenständig nutzen zu können, zeichnet ja die unabhängige, soziotechnische Arbeitsgestaltung aus. Was noch typisch für eine unabhängige soziotechnische Arbeitsgestaltung ist: STARTUP kann sowohl die **Ziele** der Softwaregestaltung als auch jene der Softwareanwendung festlegen. Primär kann STARTUP dadurch Software für die eigenen Zwecke gestalten und zugleich auch noch sekundäre Ziele verfolgen. Sekundär hat STARTUP eine Kommodifizierung der Software im Blick. Sie stellt einen Teil der Software als White-Label anderen Firmen zur Verfügung. An der Gestaltung **partizipieren** nur interne Mitarbeitende via Kreise oder Chats. Auch wenn Anwendende Tickets aufgeben können oder bei Tests mitmachen: Diese direkte Teilnahme wurde von keiner befragten Person als wesentlicher Beitrag genannt. Die Anwendenden sind aus der Softwaregestaltung größtenteils ausgeschlossen. Das liegt aber an der Entscheidung von STARTUP, dass sie deren Input für die Gestaltung nur in geringem Umfang als notwendig erachten.

Was die soziotechnische Arbeitsgestaltung durch Softwaregestaltung anbelangt, gibt es wenig **Konflikte**, weil die Organisation auf Interdisziplinarität, Softwareentwicklung und ein gemeinsames Ziel ausgerichtet ist. Die Anwendenden waren und sind von Anfang an klar hierarchisch untergeordnet und es gibt auch keinen Betriebsrat, der diese vertreten würde. Wenn es Konflikte gibt, sind alle intern, d.h. allein durch STARTUP lösbar und betreffen die Umsetzung. So gibt es z.B. einen Konflikt zwischen guten Ideen (d.h. offenen Anforderungen) einerseits und der Reduktion der Komplexität der Software, fehlenden Fachkräften und begrenztem Budget andererseits.

8.5.3. Zusammenfassung

Die Softwaregestaltung gestaltet die Software und einen Teil der anwendenden Organisation. Dabei unterscheiden sich die Fallstudien darin, ob die EVU unabhängig oder abhängig jenen Teil von Arbeit und Organisation ihrer Softwareanwendenden gestalten können, der auf die Softwaregestaltung zurückgeht. Die Zusammenfassung ordnet die einzelnen Fälle den Typen einer unabhängigen oder abhängigen soziotechnischen Arbeitsgestaltung zu. Sie fasst die Ergebnisse zusammen – auch zur Rolle des Betriebsrats – und diskutiert am Schluss zwei Thesen zum Einfluss der Softwaregestaltung auf die Softwareanwendung: Inwiefern können Anwendende direkt partizipieren? Wie würde eine reorganisierte Prozessorganisation aussehen, welche die Möglichkeiten der Softwaregestaltung für einen abteilungsübergreifenden Prozess ausschöpft?

8.5.3.1. Einordnung der Fälle zwischen abhängig und unabhängig

Die Fallstudien INTERN1, INTERN2 und STARTUP können dem Typ einer unabhängigen soziotechnischen Arbeitsgestaltung der Softwareanwendung und KOOP1, KOOP2, KOOP3 und PAKET jenem einer abhängigen zugeordnet werden. Doch zeigen die Falldarstellungen, dass sie nicht immer klar einem der diametral-gegensätzlichen Typen entsprechen und sich teilweise abhängige und unabhängige Formen mischen.

Weil STARTUP, INTERN1 und INTERN2 unabhängig in der soziotechnischen Arbeitsgestaltung durch die Softwaregestaltung sind, haben sie sowohl Einfluss auf die Software als auch die Organisation von Anwendung und Entwicklung. Doch schlägt sich diese Unabhängigkeit in den Fallstudien nicht in gleicher Weise auf das Verhältnis der Arbeitsprozesse von Softwaregestaltung und Softwareanwendung nieder:

- Die Organisationen können eigenständig die Kontrollverhältnisse zwischen den Arbeitsprozessen bestimmen. Doch ist nur bei STARTUP der Arbeitsprozess der Softwaregestaltung klar führend und die Softwaregestaltung entscheidet allein darüber, wie die Softwaregestaltung und wie die Anwendenden arbeiten. Bei INTERN2 hingegen haben die Anwendenden ihre eigenen Führungskräfte mit eigenem IT-Budget, über das sie verfügen. Bei INTERN1 treibt die Softwaregestaltung in kleinen Schritten Veränderungen im anwendenden Fachbereich auch gegen Widerstände voran.
- Es wäre eine wechselseitige Reorganisation von Softwaregestaltung und Softwareanwendung möglich. STARTUP ist bereits von Anfang an auf die Softwareentwicklung ausgerichtet, weswegen von Beginn an die Organisation den Arbeitsprozess der Softwareanwendung so verändert, wie es aus Sicht der Softwaregestaltung sinnvoll erscheint. Bei INTERN2 findet keine Reorganisation der direkten Anwendung statt. In den anwendenden Fachbereichen existieren zu den bestehenden Strukturen ergänzende Rollen wie Product Owner. Befragte der Fallstudie sehen in Zukunft eine Reorganisation aller beteiligten Fachbereiche, um die Möglichkeiten übergreifender Softwaregestaltung verwirklichen zu können. Wie das aussehen könnte, skizziert 8.5.3.4 weiter unten. INTERN1 hat sich reorganisiert, weil die mobilen Endgeräte die Position der Meister ersetzt haben, und auch dort gibt es im anwendenden Fachbereich extra ergänzende Stellen für die Softwaregestaltung.
- Die anwendenden Organisationen können sowohl die Ziele der Softwaregestaltung als auch der Anwendung festlegen. INTERN1 und INTERN2 entscheiden sich dafür, vom Status quo aus inkrementell Software und Softwareanwendung weiter zu gestalten. STARTUP hat sich für den Primat der Softwareentwicklung entschieden und dafür, zusätzlich einen Teil der Software anderen Organisationen als Standard zu verkaufen.
- Die Organisationen können selbst entscheiden, wer mitgestaltet. Jedoch bedeutet dies nicht, dass damit die Anwendenden alle mitgestalten dürfen. INTERN1 und INTERN2 beziehen ausgewählte Anwendende ein, wobei diese dann nicht das letzte Wort darüber haben, ob das EVU die Anforderung genau so umsetzt. Bei STARTUP liefern die Softwaregestaltenden maßgeblich den Input für Anforderungen und die Anwendenden geben Feedback zu fertigen Umsetzungen oder melden Fehler.

In den Fällen PAKET, KOOP₃, KOOP₂ und KOOP₁ sind die EVU abhängig in der soziotechnischen Arbeitsgestaltung der Softwareanwendung. Softwarefirmen und IT-DL gestalten die Software. Beim Einfluss auf die beiden Arbeitsprozesse zeigt sich die Abhängigkeit auf vierfache Weise:

- Wegen des Kontrollverhältnisses der beiden Arbeitsprozesse zueinander können die anwendenden EVU nur die Softwareanwendung kontrollieren. Das bedeutet bei KOOP₁, KOOP₂ und PAKET für die EVU eine Unterordnung unter eine Standardsoftware. Bis auf einzelne EVU von KOOP₁, KOOP₂ und KOOP₃ kontrollieren die Arbeitsprozesse der Softwaregestaltung IT-DL oder Softwarefirmen.
- Die EVU können nur die Softwareanwendung reorganisieren. Das nutzen EVU unterschiedlich. So berichten Befragte eines EVU von PAKET darüber, dass sie sich organisatorisch immer mehr an Prozessen ausrichten und weniger nach fachlicher Spezialisierung.
- Die EVU können nur die Ziele der Softwareanwendung festlegen, z.B. welche Rechnungen sie noch manuell prüfen bzw. wie hoch die Automatisierung ist. Die Softwarefirma entscheidet unabhängig z.B. darüber, was Teil des Standards wird. Doch können über die IT-DL die EVU von KOOP₁ und KOOP₂ über die Ziele der zentralen Softwaregestaltung mitentscheiden und es ist vor allem bei KOOP₁ von ihrer Kooperation abhängig, wie umfassend der Standard wird und als solcher auch noch anderen EVU angeboten werden kann.
- Bei PAKET und KOOP₃ entscheidet die Softwarefirma, welche EVU mitgestalten und welche Anforderungen in den Standard einfließen. Ideen und Ausarbeitungen von Anforderungen kommen zu einem großen Teil von Fachleuten. Die Abhängigkeit bei KOOP₁ und KOOP₂ ist insofern abgeschwächt, weil die EVU Gesellschaftende des IT-DL sind und auch strategische Fragen der Softwaregestaltung mitentscheiden. Einzelne Anwendende der EVU können aber nur vermittelt über formale Prozesse Anforderungen aufnehmen (z.B. über das Ticketsystem) und es gibt ein zentrales Anforderungsmanagement, das über Anforderungen abstimmt (bei KOOP₂ betrifft das nur einen kleineren Teil der Softwaregestaltung).

KOOP₁ und KOOP₂ gehören zwar grundsätzlich zum Typ abhängig. Jedoch gestalten einzelne EVU ihre Software unabhängig vom IT-DL oder der Softwarefirmen, Anwendende können direkt partizipieren und sie setzen ihre eigenen Ziele durch, indem sie Teile der Software individuell gestalten. Bei KOOP₁ gibt es einen Bereich in einem EVU, der sich reorganisiert, um besser Software gestalten zu können. Ein EVU von KOOP₂ führt eigenständig Softwaregestaltung durch, z.B. in Form von Projekten. Allerdings reorganisiert sich dieses EVU weniger, als dass es vielmehr die bestehende Organisation um zusätzliche Stellen ergänzt. Bei KOOP₃ ist die IoT-Software keine Anwendungssoftware, sondern liefert Daten, die andere Anwendungen verarbeiten. Daher betreffen die Folgen weniger die Gestaltung eines Anwendungsbereiches bzw. die Arbeit/Organisation der Softwareanwendung und mehr die Gestaltung einer Software.

Die Tabelle 18 gibt einen Überblick über Kontrollverhältnis, Reorganisation, Strategie und Partizipation in den einzelnen Fallstudien:

Tabelle 27: Einfluss von Softwaregestaltung auf Softwareanwendung

Fall	Typ	Kontrollverhältnis Softwareanwendung (SA) – Softwaregestaltung (SG)	Reorganisation SA wegen SG	Ziele SG für SA	Partizipation SA an SG
INTERN1	unabh.	Phasenweise Kontrolle SA durch SG	teilweise Reorg.	SG für Status quo	ausgewählte Anw., moderiert
INTERN2	unabh.	Kontrolle SA und SG getrennt	Teilgestaltung je Fachbereich	SG für Status quo	ausgewählte Anw., moderiert
KOOP1	abh./un-abh.	EVU: Unterordnung SA unter Standard, teilw. Kontrolle SG und SA	nur vereinzelt Reorg.	Standardware + koop. SG für Status quo	bürokratisch, moderiert, verhandelt zwischen EVU
KOOP2	abh./un-abh.	EVU: Unterordnung SA unter Standard, teilw. Kontrolle SG und SA	keine Reorg.,	Standardware + intern SG für Status quo	bürokratisch in EVU und zwischen IT-DL und EVU
PAKET	abh.	in EVU: Unterordnung SA unter Standard	ggf. Reorg. für übergreifende Einstellungen am Standard	Standardware + Status quo	ausgewählte EVU und Fachleute
KOOP3	abh./un-abh.	keine Anwendersoftware, Kooperation	keine Reorg.	Standardware, neue Anwendungsfelder	Fachleute vermittelt über IT-DL
STARTUP	unabh.	Unterordnung SA unter SG	Optimierung SG	Primag Softwareentwicklung + Standardware	verhandelt in Kreisen und (Chat-)Foren

Was die Konflikte zwischen Softwaregestaltung und Softwareanwendung anbelangt, sind sie bei den Fällen vom Typ unabhängig intern und damit durch die Organisation selbst kontrollierbar. Bei INTERN1 besteht der Konflikt vor allem darin, dass der Fachbereich gern selber die Hoheit über die Softwaregestaltung hätte. Aktuell hat das letzte Wort der Haupt-Product-Owner aus der IT-Abteilung. Bei INTERN2 mussten die Fachbereiche inkl. der IT-Abteilungen erst bereit zur Kooperation sein und nun verhandeln die Beteiligten viele Themen in der gemeinsamen Anforderungsrunde. Bei STARTUP haben die Befragten keine Konflikte genannt (leider fehlt die Perspektive der Anwendenden aus dem Fall, weil keine Interviews mit ihnen vorliegen).

In jenen Fällen, in denen die EVU in der soziotechnischen Arbeitsgestaltung der Softwareanwendung abhängig sind, bestehen die Konflikte vor allem zwischen Organisationen. Bei KOOP2 haben Konflikte dazu geführt, dass ein EVU die Kooperation verlassen hat und andere EVU die Softwaregestaltung nun wieder selbst in die Hand nehmen. Diese Aufkündigung der Kooperation hatte die Folge, dass z.B. bei einem der nun wieder

selbst gestaltenden EVU von KOOP2 die Konflikte nun intern anfallen. KOOP1 hat ein Set an Abläufen, Rollen und Beziehungen institutionalisiert, welche helfen, auf Konflikte zu reagieren und stetig Erwartungen abzugleichen. Sowohl bei KOOP1, KOOP2 und PAKET gibt es Konflikte, was die Servicequalität anbelangt, und z.B. bei PAKET finden anwendende Fachbereiche aus den EVU, dass die Softwarefirma Fehler zu langsam behebt. Bei KOOP3 bestehen weniger Konflikte zwischen Anwendung und Gestaltung und wenn überhaupt, dann innerhalb der Gestaltung, weil IT-DL und Softwarefirma miteinander konkurrieren.

8.5.3.2. Inwiefern handelt es sich um einen intervenierenden Betriebsrat?

In den Fallstudien nimmt der Betriebsrat eine intervenierende Rolle ein. Eine direkte Mitgestaltung im Sinne einer Mitwirkung an Konzepten findet nicht statt. Vielmehr verhindert er eine individuelle Verhaltenskontrolle, lässt sich über den Stand in IT-Projekten informieren, entscheidet bei der Auswahl des IT-DL mit oder setzt durch, dass die Beschäftigten bei der Softwaregestaltung einbezogen werden.

INTERN1 hat die Forderung des Betriebsrates nach einer stärkeren Partizipation der Monteur:innen umgesetzt. Weil das EVU Software intern entwickelt, hat der BR bereits im Prozess der Entwicklung die individuelle Leistungskontrolle verhindert und hat diese nicht wie bei Standardsoftwareprodukten erst nachträglich ausschalten lassen. Er wird durch die Teilnahme an den entsprechenden Sitzungen ebenso informiert wie die anderen an der Softwareentwicklung Beteiligten. Bei INTERN2 macht der Betriebsrat nicht bei der Anforderungsgrunde mit. Der Betriebsrat prüft vielmehr allgemein neue Lösungen, ob sie eine individuelle Leistungskontrolle ermöglichen. Die einen Befragten sehen ihn als Hilfe an, wenn er frühzeitig eingebunden, überzeugt und partnerschaftlich mit ihm umgegangen wird. Andere nehmen ihn als Hindernis wahr.

Bei KOOP1 hat im EVU4 der Betriebsrat eine Betriebsvereinbarung für IT-Projekte, die ihm garantiert, dass er in allen Projekt-Lenkungsausschüssen sitzt und gut informiert ist. Teilweise entscheidet er bei der Beraterauswahl mit. Wie in den anderen Fällen prüft er, ob eine Software eine individuelle Leistungskontrolle ermöglicht. Er sieht sich zwar auf Augenhöhe mit dem Management. Eine inhaltliche Gestaltung der Software ist für ihn aufgrund fehlenden Wissens und eingeschränkten internen Machtbereichs nicht möglich. Für den Betriebsrat aus EVU2 sprechen die ITler eine andere Sprache, die er nicht spricht. Auch im EVU3 gestaltet der BR nicht mit. Er hat dafür gesorgt, dass die Stunden bei Projekten begrenzt werden, damit die Mitarbeitenden nicht zu viel arbeiten, und verhindert eine Leistungskontrolle. Bei KOOP2 gibt es eine Betriebsvereinbarung für IT-Projekte. Der BR achtet darauf, dass das Management keine Datenauswertung auf Ebene der Mitarbeitenden machen kann. Für eine Mitgestaltung an der Software fehlt ihm das Wissen.

Der Betriebsrat eines EVU des Falls PAKET kritisiert, dass von den Mitarbeitenden zu viel Learning by Doing verlangt wird. Sein EVU sollte wieder mehr Schulungen anbieten. Auch er hat eine Betriebsvereinbarung für IT-Projekte und wird über deren Ablauf und die in deren Zuge stattfindenden Schulungen informiert. Er sieht sich als Mitgestalter. Doch müssten noch Nachfragen gestellt werden, ob er konkret über die Auswahl von Softwarepaketen mitentscheidet oder Softwareänderungen vorschlägt.

Bei KOOP3 verhindert der Betriebsrat in den EVU eine Leistungskontrolle. In einem Fall wurde er am Anfang eines IoT-Projektes eingebunden und es gab einen intensiven Austausch. Aus Sicht des IT-DL reicht es im Normalfall, wenn er den BR über das Projekt und dessen Stand informiert.

Im STARTUP gibt es keinen Betriebsrat.

8.5.3.3. Nehmen die Anwendenden an der Gestaltung teil?

Anstatt dass die Anwendenden direkt an der Gestaltung der Software beteiligt sind, sind es meist ihre Repräsentant:innen oder energiewirtschaftliche Expert:innen.

Zwar haben in manchen Fällen sämtliche Anwendende theoretisch in bestimmten Phasen die Möglichkeit, Anforderungen zu stellen oder Tickets aufzunehmen. Es ist auch die Rede davon, dass IT-Abteilungen, IT-DL oder Softwarefirmen Anforderungen umsetzen, die viele Anwendende aus den EVU benötigen (KOOP1, INTERN2, PAKET). Doch nehmen letztendlich nur die wenigsten Anwendenden teil und die finale Entscheidung liegt nicht bei ihnen, sondern u.a. in Gremien, Anforderungsrunden oder bei Product Owners. Bei STARTUP sind die Anwendenden aus den Kreisen, die über Anforderungen sprechen, ausgeschlossen. Bei INTERN1 und INTERN2 benennen Befragte den Konflikt zwischen Unternehmen und Anwendenden explizit, wenn es um Anforderungen geht. Wobei sich das Management letztendlich durchsetzt. Des Weiteren sind sie von den Softwarefachleuten abhängig, um überhaupt etwas über die Möglichkeiten der Softwaretechnik zu wissen und die Kosten abschätzen zu können. Es bleibt eine Ausnahme, dass einzelne Anwendende ihre spezifischen Anforderungen durchbekommen, wie z.B. bei INTERN1, wo es Befragte explizit erwähnt haben, oder bei INTERN2, wo der Product Owner Interviews mit Anwendenden führt, um deren Bedarfe zu ermitteln.

Inwiefern Anwendende mitgestalten, lässt sich anhand von drei Variablen strukturieren: dem Zugang zur Gestaltung (direkt oder über Repräsentant:innen), der Rolle der Expert:innen und wie sich Unternehmens- und Anwenderinteressen zueinander verhalten. Es lassen sich drei Typen bilden:

Bei der direkten Partizipation agieren die Anwendenden selbst als Expert:innen (z.B. als Teil eines interdisziplinären Teams), die Anforderungen stellen und mitgestalten. Sie haben das notwendige Wissen, um eigenständig Anforderungen formulieren zu können, und sind dafür nicht von Expert:innen abhängig. Der Typ geht von einer Produktionsgemeinschaft aus und damit von keinem Interessengegensatz zwischen Softwareanwendung und Softwareentwicklung.

Beim Typ der repräsentativen Technokratie werden Anwendende vertreten: ob durch Key User:innen, IT-Beratende oder andere Anwendende, die Anforderungen aufnehmen. Die Softwaregestaltung sucht einen Ausgleich zwischen den Interessen der Technokratie und den Anwendenden, indem sie bspw. ein Budget für die Umsetzung von Anforderungen der Anwendenden zur Verfügung stellt.

Bei der kapitalistischen Technokratie geben die Anwendenden maximal Feedback zu einer von Expert:innen entwickelten Software (z.B. durch Tests oder eingegebene Daten). In diesem Fall ist das Wissen (ob explizites oder implizites) der Anwendenden für Anforderungen nicht relevant und die Software bildet einen Best-Practice-Prozess ab. Der Klassenkonflikt zwischen Interessen von Unternehmen und Anwendenden schlägt sich in unterschiedlichen und unvereinbaren Anforderungen an die Software nieder.

Tabelle 28: Typen der Partizipation von Anwendenden an der Softwaregestaltung

Nr.	Partizipationszyp	Zugang der Anwendende	Rolle Expert:innen	Interessen Unternehmen vs. Anwendende
1	Direkte Partizipation	direkt	keine	kongruent
2	Repräsentative Tech-nokratie	indirekt	beratend, vermit-telnd	ausgeglichen
3	Kapitalistische Tech-nokratie	keinen	alleinig softwarege-staltend	Konflikt

Die analytisch gebildeten Typen sollen verdeutlichen, dass keine der Fallstudien einer reinen, direkten Partizipation entspricht. Es gibt immer noch Vermittelnde zwischen Anwendenden und Programmierenden, der Betriebsrat interveniert gegen eine individuelle Leistungskontrolle und bestimmte Gestaltungswünsche des Managements haben Priorität. Ansonsten sind die Fallstudien Mischformen zwischen Typ 2 und Typ 3. Bei Fallstudien wie KOOP1 und KOOP2 hängt es von den einzelnen EVU ab, inwieweit sie überhaupt noch mitgestalten. INTERN1 und INTERN2 suchen zwar einen Ausgleich zwischen den Anwendenden- und Unternehmensinteressen, einzelne Anwendende können direkt Anforderungen stellen und Befragte berichten von einer Zufriedenheit mit der Softwarelösung. Das Unternehmensinteresse übertrumpft aber letztendlich die Interessen der Anwendenden. Beim STARTUP sind die Anwendenden weitestgehend ausgeschlossen und Fachleute der Softwaregestaltung wie Product Owner:innen gestalten die Software. Bei PAKET und KOOP3 fließt die Expert:innenmeinung im Wesentlichen über Arbeitskreise ein und der einzelne Anwendende in einem EVU hat wenig direkte Mitsprache.

Letztendlich verteilt sich Partizipation wie z.B. bei KOOP1, wo verschiedene Ebenen entscheiden: ein strategisches Gremium des Managements für größere Projekte und grundlegende Entscheidungen und operative Gremien für einzelne Anforderungen, die aus unterschiedlichen EVU kommen.

8.5.3.4. Wie würde eine auf Softwaregestaltung ausgerichtete Prozessorganisation aussehen?

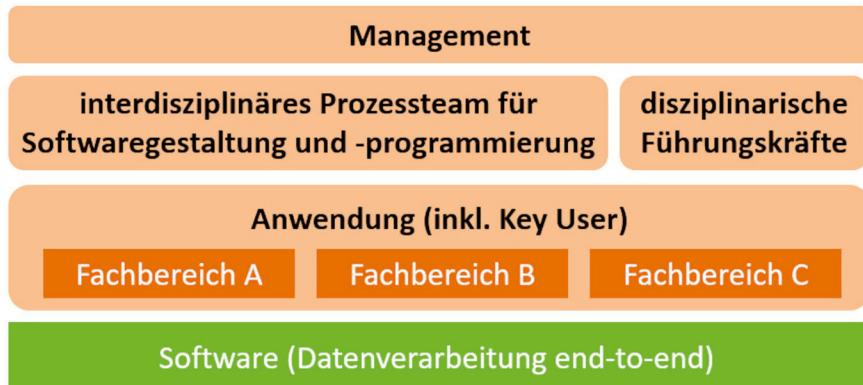
Die soziotechnische Transformation durch Softwaregestaltung stellt in einigen Fällen die bestehende Organisation in Frage. Bei INTERN2 werden die Fachbereiche (»Silos«) in Frage gestellt und ein befragter Manager will eine team- und fachbereichsübergreifende Prozessoptimierung. Dafür soll es interdisziplinäre Teams geben, bei denen Softwaregestaltende und -programmierende ohne Barrieren durch anwendende Teams oder Fachbereiche die Arbeit der Anwendenden gestalten.

Bei INTERN1 betrifft die Softwaregestaltung bereits zwei Prozessteile (Instandhaltung und Auftragsverarbeitung). Bei KOOP1 gibt es in einem EVU ein Team, das sich reorganisiert, um für Geschäfts- und Privatkundschaft zusammen die Software zu gestalten, weil im Zuge der Softwaregestaltung das EVU viele Gemeinsamkeiten zwischen

beiden festgestellt hat. Bei PAKET gibt es in einem EVU Prozessmanagende, die teamübergreifend die Software durch Einstellungen am Standard optimieren.

Zwar hat sich dieser neue Typ von Organisation in keinem der Fälle verwirklicht. Er sei hier aber kurz skizziert, um einige Kernthesen der Arbeit zu verdeutlichen: Es wäre eine soziotechnische Prozessorganisation, bei der interdisziplinäre Teams an und mit einer Software arbeiten. Programmierende und andere IT-Fachleute wären nicht mehr Teil einer IT-Abteilung, sondern Teil der Fachabteilung. Fachleute für die energiewirtschaftlichen Prozesse, für ERP-Systeme und Anwendende würden kontinuierlich eng zusammenarbeiten. Es wäre eine technikentwicklungsinduzierte Reorganisation, weil sich das EVU so aufstellen würde, dass es die Möglichkeiten der Softwaregestaltung optimal nutzen kann. Kern der Prozessorganisation wäre ein End-to-end-Prozess (bspw. vom Auslesen der Zählerdaten bis zum Erstellen der Rechnung), der in einer Software abgebildet ist. Statt mehrerer Teamleitender für einzelne Prozessabschnitte gibt es ein Prozessteam, das eine funktionsübergreifende Perspektive auf den ganzen Prozess einnimmt und die entsprechende Macht hat, Änderungen in jedem fachlichen Anwendungsbereich entlang des Prozesses umzusetzen. Der Ablauf wäre an Scrum ausgerichtet und mit einem Anforderungsmanagement verbunden, deren Teilnehmende in die Teams hinein vernetzt sind. Er würde zwischen Anwendung und Programmierung für Feedbackschleifen inkl. Tests und Prototyping sorgen. Eine direkte Kommunikation zwischen den Beteiligten wäre möglich, z.B. durch regelmäßige Sprechstunden oder Chats. Die Beziehungen wären partnerschaftlich, verbindlich, kooperativ. Alle Beschäftigten zeigen individuelle Verantwortungsbereitschaft gemäß ihrer Rolle. Die Anwendenden wären aufgeteilt in Key User:innen und reine Anwendende. Erstere machen aktiver an der Softwaregestaltung mit und unterstützen Letztere, bei denen z.B. die Abarbeitung nicht automatisierter Fälle im Zentrum steht. Es gibt disziplinarische Führungskräfte, die sich vom Prozessteam unterscheiden und die Rolle von Coaches einnehmen. Das Prozessmanagement überwacht die Kennzahlen.

Abbildung 12: Schema einer auf Softwaregestaltung ausgerichteten softwaretechnischen Prozessorganisation



So wäre die anwendende Organisation auf die Softwaregestaltung ausgerichtet. Dabei prüft das Prozessteam kontinuierlich, was sie individuell gestalten oder wo sie eine Standardlösung einsetzen und ob sie Technologien wie maschinelles Lernen nutzen. Dazu gehört, dass sich sowohl das Prozessteam als auch die Anwendung entsprechend den ausgeschöpften Möglichkeiten reorganisieren (z.B. aufgrund von fortschreitender Automatisierung). Weil Softwaregestaltung nicht immer im gleichen Umfang nötig ist, wären manche Gestaltende und Programmierende nur phasenweise dabei und z.B. als Externe engagiert oder noch in anderen IT-Projekten oder Prozessteams tätig.

Der Organisationstyp soll noch einmal verdeutlichen, dass nicht nur die Software, sondern auch die Softwaregestaltung zentral für die anwendende Organisation ist und dass die EVU in den Fallstudien im Wesentlichen noch so organisiert sind, wie es aus rein energiewirtschaftlicher Sicht naheliegt. Eine Reorganisation würde vor allem Führungsebenen wie Teamleitende betreffen. Sie würde klar zwischen Führungskräften unterscheiden, die nur disziplinarisch verantwortlich sind, und solchen, die die soziotechnische Arbeitsgestaltung durch Softwaregestaltung übernehmen. Wie sehr sich dabei die Interessen des Managements jenseits der Nutzung der Technologie für die betrieblichen Zwecke durchsetzen, würde von betrieblichen Bedingungen abhängen, bspw. ob ein Betriebsrat vorhanden ist oder nicht.

8.6. Synthese, Zusammenfassung und Diskussion des Fallvergleichs

Die sieben Fallstudien stellen Formen und Folgen der Softwaregestaltung in der Energiewirtschaft dar und werfen einen neuen Blick auf die Arbeit und die Organisationen der Branche. Sie zeigen die Softwaregestaltung als eigenständigen Arbeitsprozess, der in verschiedenen soziotechnischen Konstellationen stattfindet und der es den Organisationen erlaubt, die Möglichkeiten der Softwareentwicklung (Standard oder individuell) mit den Bedarfen der Energiewirtschaft zusammenzuführen. Sie zeigen, dass sich mithilfe des Analyserahmens die soziotechnische Konstellation, der Arbeitsprozess der Softwaregestaltung, die Folgen für die Arbeit der Softwaregestaltenden und die soziotechnische Arbeitsgestaltung der Softwareanwendung für die Fallstudien darstellen, vergleichen und Zusammenhänge zwischen ihnen untersuchen lassen. Damit haben die Fallstudien gezeigt, dass neben Methoden wie Scrum oder IT-Projekten auch noch andere Dinge zu einer adäquaten konzeptionellen Beschreibung von Softwaregestaltung und ihren Folgen gehören, wie z.B. organisationsübergreifende Abstimmungen, Karrierewege für Softwaregestaltende und die Reorganisation von Abteilungen oder zumindest die Ergänzung der anwendenden Organisation um Rollen und Abläufe für die Softwaregestaltung. Software erhält als wesentliche Entität einer Organisation einen zentralen Stellenwert dabei, Handeln in Strukturen zu erklären und zu verstehen. Zudem zeigen die Fallstudien die Situativität des Arbeitsprozesses: Rollen, Abläufe, digitale Werkzeuge und kommunikative Beziehung zeichnen Anpassung, Reflexivität, Phasenverläufe, Verhandlungen und Lernprozesse aus.

Aus der Empirie konnte das Kapitel allgemeine Unterschiede für den Arbeitsprozess der Softwaregestaltung (zentral oder dezentral), die Arbeit der Softwaregestaltenden (in einer Matrix- oder reinen Netzwerkorganisation) und die soziotechnische Ar-

beitsgestaltung der Anwendung (ob unabhängig oder abhängig) herausarbeiten. Wobei die letztere Unterscheidung zwischen unabhängiger und abhängiger Arbeitsgestaltung sich nicht auf die zwei grundsätzlich unterschiedlichen organisatorischen Ausrichtungen von In- und Outsourcing bzw. Make-or-Buy reduzieren lässt. Sie ist immer verbunden mit einem softwaretechnischen Zuschnitt (individuell/Standard) (siehe weiter unten, 8.6.1.3). Die Fallstudien zeigen die genannten drei idealtypischen Unterschiede nicht immer in Reinform. Um die Fälle zu sortieren und die Ergebnisse zu systematisieren, stellt das Empirie-Kapitel Idealtypen vor, welche die Unterschiede zuspitzen: jeweils vier für die soziotechnische Netzwerkarbeit und die soziotechnische Arbeitsgestaltung der Softwareanwendung. Sie lassen sich in 4-Felder-Matrizen einsortieren (siehe Abbildung 15 und Abbildung 16 unten), wodurch sie auf einen Blick veranschaulichen, welche unterschiedlichen Formen und Folgen der Softwaregestaltung es gibt.

Zudem fasst der abschließende Abschnitt des Kapitels die Ergebnisse der Fallstudien für die einzelnen Kategorien des Analyserahmens zusammen und gleicht sie mit der Literatur und den Konzepten aus dem 6. Kapitel ab. Er stellt ausführlich den technikentwicklungsbezogenen Rationalisierungstyp vor, der noch einmal untermauert, dass Softwaregestaltung Anwendungsbereiche rationalisiert und für das Verständnis der Rationalisierung der Branche wichtig ist. Abschließend diskutieren die letzten beiden Punkte 8.6.4 und 8.6.5 zwei Thesen: inwiefern die Softwaregestaltenden eine Konkurrenz zum Management in den EVU darstellen und inwiefern es eine industriespezifische Softwaregestaltung gibt.

8.6.1. Synthese: Typen soziotechnischer Netzwerkarbeit und soziotechnischer Arbeitsgestaltung

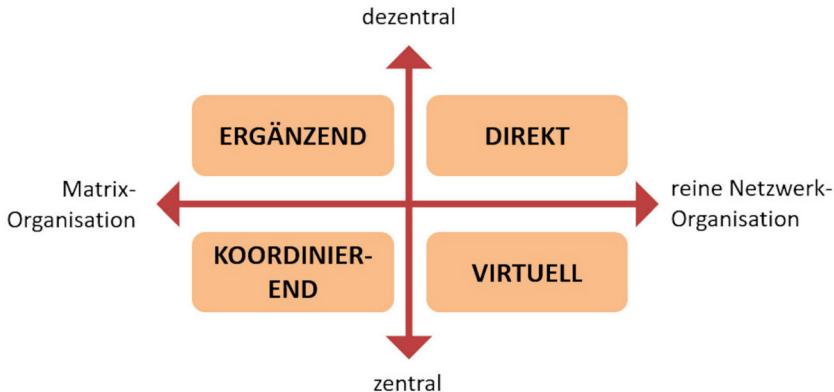
8.6.1.1. Die vier Grundtypen der soziotechnischen Netzwerkarbeit

Bei der Frage, wie Organisationen die Arbeit der Softwaregestaltenden kontrollieren, spielen sowohl der Arbeitsprozess als auch die Arbeitsbedingungen der Softwaregestaltenden eine Rolle. Aus beiden zusammen lässt sich eine 4-Felder-Matrix bilden, in die sich die unterschiedlichen Fallstudien einsortieren lassen: zwischen Matrix- und reiner Netzwerkorganisation für die Arbeit der Softwaregestaltenden einerseits und zwischen dezentralem und zentralem Arbeitsprozess der Softwaregestaltung andererseits. Wie die untenstehende Abbildung 15 darstellt, entsprechen den vier möglichen Kombinationen dieser Unterschiede vier Idealtypen: DIREKT, ERGÄNZEND, KOORDINIEREND und VIRTUELL. Die Eigenschaften der Idealtypen ergeben sich entsprechend aus den idealtypischen Unterscheidungen der Fallstudien, wie sie zu Beginn der jeweiligen Abschnitte zu den Teilen des Analyserahmens aufgeführt sind: unter 8.3.1 für den Arbeitsprozess und 8.4.1 für die Arbeit der Softwaregestaltenden.

Im ersten Quadranten der 4-Felder-Matrix befinden sich dezentrale Arbeitsprozesse der Softwaregestaltung, bei denen die Softwaregestaltenden in einer reinen Netzwerkorganisation arbeiten. Der Idealtyp der soziotechnischen Netzwerkarbeit für diesen Quadranten ist **DIREKT**. Bei DIREKT arbeiten die Beteiligten kontinuierlich interdisziplinär zusammen, und zwar dezentral in Abteilungen, Teams oder, wie im Fall von STARTUP, in Form von Kreisen. Die direkte Anforderungsaufnahme, -ausarbeitung und

-übergabe an die Programmierenden ist weder durch Markt oder Hierarchie getrennt, wie es z.B. für ein Scrum-Team typisch ist.

Abbildung 13: Matrix Arbeitsprozess (dezentral – zentral) und Arbeitsbedingungen der Softwaregestaltenden (Matrixorganisation – reine Netzwerkorganisation) und die vier Idealtypen der soziotechnischen Netzwerkarbeit



Im zweiten Quadranten der 4-Felder-Matrix befinden sich dezentrale Arbeitsprozesse der Softwaregestaltung, bei denen die Softwaregestaltenden in einer Matrixorganisation arbeiten. Der Idealtyp für diesen Quadranten ist **ERGÄNZEND**. INTERN1 und INTERN2 sind Beispiele für diesen Typ, bei denen es vor allem darum geht, trotz Abteilungsgrenzen direkt im Anwendungsbereich Anforderungen aufzunehmen, auszuarbeiten und von den Programmierenden der IT-Abteilung umsetzen zu lassen.

Im dritten Quadranten der 4-Felder-Matrix befinden sich zentrale Arbeitsprozesse der Softwaregestaltung, bei denen die Softwaregestaltenden in einer Matrixorganisation arbeiten. Der Idealtyp für diesen Quadranten ist **KOORDINIEREND**, weil sich mehrere Organisationen oder Abteilungen über den softwaretechnischen Zuschnitt abstimmen. Wie für eine zentrale Softwaregestaltung typisch, sorgt der Arbeitsprozess für die Koordination der Zusammenarbeit mehrerer Organisationen bzw. Organisationseinheiten. Die Organisationen, Abteilungen oder Teams verhandeln über Anforderungen und lösen die dabei entstehenden Konflikte. KOOP1, KOOP2 und KOOP2 sind nur zum Teil Beispiele dafür, weil in diesen Fallstudien neben dem zentralisierten Arbeitsprozess vor allem bei KOOP2 noch dezentrale Arbeitsprozesse innerhalb der EVU existieren. Allen ist jedoch gemein, dass die Softwaregestaltenden Organisation- und Abteilungsgrenzen überwinden müssen. Bei PAKET ist das der Fall, wenn die Softwarefirma in Arbeitskreisen oder in Entwicklungsprojekten mit EVU zusammenarbeitet. Allerdings existieren innerhalb der Softwarefirma sowohl interdisziplinäre Teams für einzelne Module der Standardlösung, die einer soziotechnischen Netzwerkarbeit vom Typ DIREKT nahekommen, als auch mehrere Teams betreffende interne Projekte für größere Anforderungen, die eher dem Typ ERGÄNZEND entsprechen.

Im vierten Quadranten der 4-Felder-Matrix befinden sich zentrale Arbeitsprozesse der Softwaregestaltung, bei denen die Softwaregestaltenden in einer reinen Netzwerkorganisation arbeiten. Der Idealtyp für diesen Quadranten ist **VIRTUELL**. Dabei gestalten Akteur:innen zentralisiert Software, z.B. auf einer frei zugänglichen Online-(Open-Source-)Entwicklungsplattform. Es handelt sich um ein reines Netzwerk, da die einzige Hürde zur Teilnahme darin bestehen kann, ein Benutzer:innenkonto für die Plattform zu haben. Wer eins hat, kann den Quellcode einsehen und sehen, wer gerade an welchem Teil der Software arbeitet. Der Austausch über branchenspezifische Anforderungen und softwaretechnische Möglichkeiten sowie die Lösung von Konflikten erfolgt digital. Die Beteiligten wählen selbst, wo sie mitarbeiten bzw. welchen Beitrag sie leisten. Keine der Fallstudien entspricht diesem Typ.

8.6.1.2. Zwei organisationale Kernstrategien der Arbeitsgestaltung in den EVU: Ausrichtung auf Softwareanwendung vs. auf Softwareentwicklung

Auch für die soziotechnische Arbeitsgestaltung durch die industriespezifische Softwaregestaltung lassen sich vier Grundtypen und eine 4-Felder-Matrix bilden. Was die soziotechnische Arbeitsgestaltung durch die Softwaregestaltung anbelangt, haben sich in der Empirie die Fallstudien dahingehend unterschieden, ob bei ihnen eine unabhängige oder abhängige Arbeitsgestaltung vorliegt. Dahinter verbirgt sich aber ein Verhältnis zwischen Softwaregestaltung und Softwareanwendung, das sowohl die Organisation als auch die Software der Softwareanwendung betrifft.

Doch zeigt bereits eine rein eindimensionale, rein organisatorische Sicht auf die Unterschiede zwischen den Fallstudien einige gravierende Folgen für die Arbeitsgestaltung. Dazu dient die Gegenüberstellung von Make-or-Buy bzw. ob sich ein EVU auf die Softwareanwendung oder die Softwareentwicklung ausrichtet. Nachteile der Fremdentwicklung bzw. organisatorische Abhängigkeiten treten hier deutlich hervor: Es bestehen Interessengegensätze und Machtungleichgewichte. Eine anwendende Organisation wie ein EVU kann nur die Softwareanwendung kontrollieren und entsprechend ist auch nur für die Softwareanwendung eine interne Rationalisierung möglich. Intern ist kein eigenes Wissen über die softwaretechnischen Möglichkeiten vorhanden. Im Wettbewerb muss die anwendende Organisation primär auf das Mittel der Spezialisierung setzen. Diese Nachteile existieren in den Fallstudien, in denen die EVU nicht selbst gestalten (vor allem bei PAKET).

Die Wettbewerbsstrategie des *Competing on Complexity* dank selbst entwickelter Software zeichnet laut Bessen (2022) die Marktführer verschiedenster Branchen aus (wie bereits unter 4.1 ausgeführt). Sie ist nur zu verwirklichen, wenn sich eine Organisation auf die Softwareentwicklung ausrichtet. Die Fallstudie STARTUP zeigt noch am ehesten diese Strategie, wobei die Organisation eine Nische bedient und, was die Umsätze anbelangt, mit EVU nicht konkurrieren kann. INTERN1 und INTERN2 zeigen, dass die Typen der soziotechnischen Arbeitsgestaltung auch für einzelne Teile einer Organisation gelten können. Das heißt, einzelne Fachbereiche können selbst anfangen, Software zu entwickeln.

Tabelle 29: Unterschiede zwischen integrierter und desintegrierter Softwareentwicklung aus EVU-Sicht

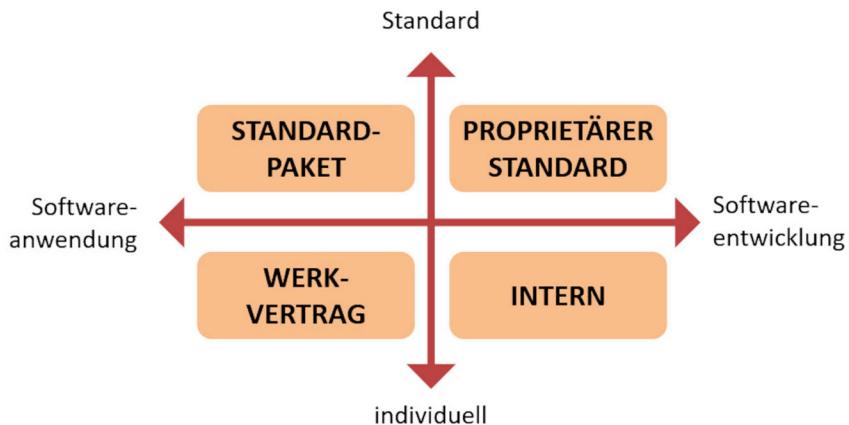
Dimension aus EVU-Sicht	Ausrichtung auf Softwareanwendung	Ausrichtung auf Softwareentwicklung
Arbeitsteilung Softwareanwendung und -gestaltung	desintegriert	integriert
Interessen	Gegensätze, Abhängigkeiten	Kongruenz, Möglichkeiten ausschöpfen
Kontrollverhältnis	Kontrolle Softwareanwendung durch (fremdgefertigte) Software	Kontrolle Softwareanwendung durch -gestaltung möglich: direkte Gestaltung Anwendungsbereich
Professionalisierung und Rationalisierung	Softwareanwendung, -gestaltung, -programmierung getrennt	gleichzeitig: Softwareanwendung, -gestaltung, -programmierung
Dynamik (branchen)fachliche und softwaretechnische Veränderungen	getrennt voneinander	integriert, Wechselspiel
Zentrale Wettbewerbsstrategie	Competing on Core Competency (Anwendung oder Entwicklung)	Competing on Complexity (Anwendung und Entwicklung)
Mögliche Kommodifizierung der Software durch	Softwarefirma	Anwenderfirma (deren Geschäftsfeld primär nicht der Verkauf von Software ist)

Die zwei Strategietypen berücksichtigen nur die Unterscheidung auf organisationaler Ebene zwischen der Ausrichtung auf Softwareanwendung oder Softwareentwicklung. Es fehlt die Unterscheidung zwischen Individual- und Standardsoftwaregestaltung. Dies geschieht in den vier Grundtypen der soziotechnischen Arbeitsgestaltung der Softwareanwendung, die über die klassische Unterscheidung von Make-or-Buy hinausgehen und soziotechnisch definiert sind.

8.6.1.3. Die vier Grundtypen der soziotechnischen Arbeitsgestaltung durch Softwaregestaltung

Die Unterschiede im Verhältnis der Arbeitsprozesse von Softwaregestaltung und Softwareanwendung treten besonders deutlich hervor, wenn sie auf vier Typen reduziert werden. Ausgehend von den softwaretechnischen Gestaltungsmöglichkeiten und den Fallstudien, lassen sich vier Idealtypen der soziotechnischen Arbeitsgestaltung bilden. Sie lassen sich auf den Achsen des soziotechnischen Zuschnitts zwischen individuell und Standard und der organisatorischen Ausrichtung zwischen Softwareanwendung und -entwicklung verorten.

Abbildung 14: Matrix softwaretechnischer Zuschnitt (Standard – individuell) und organisatorische Ausrichtung (Anwendung – Entwicklung) und die vier Idealtypen der soziotechnischen Arbeitsgestaltung (als Verhältnis von Softwaregestaltung zu -anwendung)



Beim Idealtyp **STANDARDPAKET** gestaltet wie in der Fallstudie PAKET eine Softwarefirma einen industriespezifischen Standard. Die Softwarefirma schlägt Profit aus der Skalierung des Standardprodukts. Dafür braucht sie (Branchen-)Fachwissen für den Standard und weniger Wissen aus individuellen Arbeitskontexten. Die anwendenden EVU richten sich auf diesen Standard aus. Sie spezialisieren sich auf die Softwareanwendung und zielen nicht darauf ab, sich durch eine individuelle Softwaregestaltung von Wettbewerbern zu differenzieren. Im Arbeitsprozess der Softwaregestaltung entscheiden die Beteiligten für jede Anforderung, ob sie Teil des Standards wird oder nicht. Zumindest was den kooperativ gestalteten Standard anbelangt, gehören KOOP1 und KOOP2 zu diesem Typ. Ebenso gehört KOOP3 dazu, da die IoT-Softwarefirma Standardmodule entwickelt.

Beim Typ **WERKVERTRAG** entwickelt eine Softwarefirma oder ein IT-Dienstleistungsunternehmen (IT-DL) eine industriespezifische Software im Auftrag eines EVU. Die anwendenden EVU richten sich auf diese für sie individuell gestaltete Software aus. Es besteht eine beidseitige, hohe Abhängigkeit. Dies liegt zum einen am fehlenden Wissen über Softwareentwicklung im EVU, was u.a. zu Schwierigkeiten dabei führt, dem finanziellen Aufwand der Umsetzungen zu schätzen oder etwas gegen eine zeitlich verzögerte oder mangelhafte Entwicklung zu tun. Zum anderen liegt es daran, weil der Softwarefirma oder dem programmierenden IT-DL das Wissen über den industriespezifischen Anwendungsbereich fehlt. Das ist vor allem dann problematisch, wenn stetig Änderungen notwendig sind und es sich um eine komplexe, branchenspezifische Wissensdomäne handelt. Keine Fallstudie entspricht diesem Typ. Die EVU wirken immer noch mit an der Softwaregestaltung, wenn auch nur durch Konzepte oder Tests. Bei KOOP1 und KOOP2 gibt es EVU, die das IT-DL mit einzelnen individuellen, kleineren Programmierungen beauftragen, wobei das IT-DL längerfristig mit den EVU zusammenarbeitet und über EVU- und Branchenwissen verfügt, weswegen die Abhängigkeiten geringer sind.

Beim Typ **INTERN** entwickeln EVU wie in den Fallstudien INTERN1 und INTERN2 eine individuelle industriespezifische Software. Theoretisch könnten sich die EVU so organisieren, dass es für den Arbeitsprozess der Softwaregestaltung nützlich ist, und z.B. Abteilungsgrenzen auflösen. Jedoch findet in den beiden Fallstudien keine umfassende Reorganisation für die Softwaregestaltung statt. Die EVU gestalten Software von althergebrachten Strukturen bzw. ihrem Status quo aus. Zudem erweitern und adaptieren die EVU ein Standard-ERP-System und die meisten Programmierenden sind Externe. Allerdings zeigt sich ein Lernprozess, der sich in den EVU durch unterschiedliche Grade an organisatorischen Ergänzungen auszeichnet: von einzelnen neuen Rollen und Methoden wie IT-Projekte oder Scrum bis hin zu umfassenden Arbeitsprozessen der Softwareentwicklung, die jenen einer darauf spezialisierten Softwarefirma ähneln (wenn auch z.B. über Abteilungsgrenzen hinweg organisiert).

Beim Typ **PROPRIETÄRER STANDARD** schöpft eine Organisation die Möglichkeiten der Softwaregestaltung maximal aus. Sie gestaltet individuell eine eigene Software und verkauft sie zudem auch noch anderen Unternehmen. Die Organisation ist auf die Softwareentwicklung ausgerichtet. Ausgehend vom Pramat der Softwareentwicklung besteht der Anwendungsbereich aus jener Arbeit, den Software nicht erledigen kann. Die Arbeit der Anwendenden reorganisiert die Organisation bei Bedarf beliebig und kontinuierlich im Wechselspiel mit der Weiterentwicklung der Software. Sowohl eine Differenzierung von Wettbewerbern durch individuelle Softwaregestaltung als auch eine Kommodifizierung für die Teile der Software, die anderen (auch Wettbewerbern) zur Nutzung angeboten werden, ist möglich. Zu diesem Typ gehört die Fallstudie STARTUP.

Wie für Idealtypen üblich, weichen die realen Fälle davon ab. PAKET ist zwar vom Typ STANDARDPAKET, jedoch ergänzen einzelne EVU die Softwarelösung noch um Systeme anderer Softwarefirmen und es sind umfangreichere Einstellungen an der Standardlösung möglich, die einige EVU eigenständig vornehmen. Der Fall STARTUP bietet nur einen Teil seiner Lösung zur Verwendung für andere an (Teil-Kommodifizierung). Die anderen Fallstudien passen immer einen bestehenden Standard an: Bei KOOP1 gelingt es dem IT-DL, einen eigenen Standard auszuprägen, stetig weiterzuentwickeln und zu kommodifizieren, indem die beteiligten EVU für diesen bezahlen. Manche EVU von KOOP1 gestalten zusätzlich noch selbst intern. Einige EVU, insbesondere kleinere, sind rein anwendend. Das heißt, bei KOOP1 gehören einzelne EVU auch zum Typ INTERN und andere EVU gehören nur zum Typ STANDARDPAKET. Bei KOOP2 ist dies ähnlich. Wobei sich dort einzelne EVU klar aus der Kooperation verabschiedet haben und unter den Typ INTERN fallen. KOOP3 trennt durch Modularisierung der Software den Standardteil von individuellen Entwicklungen. Einige EVU verwenden das IoT-Standardpaket, während ein befragtes EVU selbst individuelle Erweiterungsmodule entwickelt.

8.6.1.4. Zusammenhänge zwischen den einzelnen Teilen des Analyserahmens

Es zeigen sich zwei wesentliche Zusammenhänge zwischen den Teilen des Analyserahmens, wenn es um den Arbeitsprozess der Softwaregestaltung und die soziotechnische Arbeitsgestaltung der Softwareanwendung geht: (1.) Wie Organisationen die Möglichkeiten der Softwaregestaltung nutzen und den Arbeitsprozess kontrollieren können. (2.) Wie effizient Organisationen die Möglichkeiten der Softwaregestaltung zum Einsatz bringen.

Zuerst zu den typischen Zusammenhängen zwischen dem Arbeitsprozess der Softwaregestaltung und der soziotechnischen Arbeitsgestaltung der Softwareanwendung. Unter 8.3.1 ist der Idealtyp des dezentralen Arbeitsprozesses der Softwaregestaltung mit dem softwaretechnischen Zuschnitt individuell und jener des zentralen Arbeitsprozesses mit Standard definiert. Entsprechend zeigen sich die Zusammenhänge in den Fallstudien:

- In den Fallstudien INTERN1 und INTERN2 liegen eine soziotechnische Netzwerkarbeit vom Typ ERGÄNZEND und eine individuelle soziotechnische Arbeitsgestaltung der Anwendung vom Typ INTERN vor.
- In der Fallstudie STARTUP liegen eine soziotechnische Netzwerkarbeit vom Typ DIREKT und eine individuelle soziotechnische Arbeitsgestaltung der Anwendung vom Typ INTERN vor.
- In den Fallstudien KOOP1, KOOP2, KOOP3 und PAKET liegen eine soziotechnische Netzwerkarbeit vom Typ KOORDINIEREND und eine soziotechnische Arbeitsgestaltung der Anwendung vom Typ STANDARDPAKET vor.

Allerdings sind diese Zusammenhänge in den Fallstudien nicht eindeutig, weil in einigen Fallstudien mehrere Arbeitsprozesse der Softwaregestaltung existieren oder einzelne Arbeitsprozesse nicht den Idealtypen entsprechen.

- Bei INTERN2 arbeiten mehrere Fachabteilungen zusammen und stimmen sich ab. Sie müssen sich teilweise auf einen gemeinsamen Standard einigen.
- STARTUP bietet die individuell gestaltete Software anderen Firmen als Standard an, weshalb die Fallstudie noch mehr zum Typ PROPRIETÄRER STANDARD gehört als zum Typ INTERN.
- Bei KOOP1, KOOP2 und KOOP3 gibt es innerhalb der EVU Softwaregestaltung, weswegen eine soziotechnische Netzwerkarbeit des Typs ERGÄNZEND für eine soziotechnische Arbeitsgestaltung des Typs INTERN existiert. Zugleich gestalten die IT-DL für manche EVU individuelle Software, weswegen eine soziotechnische Netzwerkarbeit des Typs ERGÄNZEND in Kombination mit der soziotechnischen Arbeitsgestaltung vom Typ WERKVERTRAG existiert.
- PAKET gestaltet unabhängig von den EVU Teile des STANDARDPAKETs mit soziotechnischer Netzwerkarbeit vom Typ ERGÄNZEND oder DIREKT.

Somit besteht zwar ein enger Zusammenhang zwischen soziotechnischer Konstellation (Arbeitsteilung, Grundkonstellation und Architektur), soziotechnischer Netzwerkarbeit (Arbeitsprozess, Arbeit der Softwaregestaltenden) und soziotechnischer Arbeitsgestaltung. Abweichungen sind aber möglich und vorhanden.

Neben diesen Zusammenhängen, welche die bloße Umsetzung der Softwaregestaltung betreffen, zeigen die Fallstudien Zusammenhänge hinsichtlich der effizienten Umsetzung dieser Möglichkeiten der Softwaregestaltung.

Erstens bietet für die Softwaregestaltung die reine Netzwerkorganisation Vorteile. Dann müssen die Softwaregestaltenden keine Hierarchien oder Abteilungsgrenzen überwinden und nicht mit den Spannungen zwischen Organisationen aufgrund von

Marktbeziehungen zurechtkommen. Solche Vorteile zeigen sich in der soziotechnischen Netzwerkarbeit des Typs DIREKT wie bei STARTUP. Dort ist die gesamte Organisation auf die interdisziplinäre Zusammenarbeit für die Softwaregestaltung ausgerichtet und eingespielt. Die Nachteile zeigen sich in der Fallstudie KOOP2 im EVU2. Es muss erst noch die interne Zusammenarbeit in der Matrixorganisation lernen.

Zweitens, was die soziotechnische Arbeitsgestaltung anbelangt, nutzt eine anwendende Organisation die Möglichkeiten der Softwaregestaltung dann maximal, wenn sie selbst individuell eine Software für die eigene Anwendung gestaltet und zugleich diese Software auch noch anderen anbietet und damit Geld verdient wie beim Typ PRORITÄRER STANDARD. Auch diese Vorteile nutzt exemplarisch STARTUP. Es ist eine anwendende Softwarefirma, welche an zwei Wertschöpfungsketten teilnimmt: jener der Energiewirtschaft und jener der Software-Industrie.

Doch egal wie effizient die Organisationen die Möglichkeiten der Softwaregestaltung nutzen: Die soziotechnische Netzwerkarbeit der Softwaregestaltung müssen alle hinbekommen. Die folgenden Abschnitte fassen die Ergebnisse für jeden Teil und jede Kategorie des Analyserahmens zusammen und stellen Bezüge zur Forschungsliteratur aus dem 6. Kapitel her.

8.6.2. Zusammenfassung je Teil des Analyserahmens

8.6.2.1. Soziotechnische Konstellation

Die Fallstudien zeigen, dass die soziotechnische Konstellation die Ausgangsbedingungen für die Softwaregestaltung darstellt. Zugleich stellt sie auch deren Grenzen dar, in denen sich die Möglichkeiten der Softwaretechnik verwirklichen lassen: Vom Anwendungsbereich hängt der mögliche Digitalisierungsbeitrag der Softwaregestaltung ab, von der Softwarearchitektur die Arbeitsteilung zwischen Anwendung und Entwicklung, von der Arbeitsteilung wiederum die Wissensgrenzen und unabhängig davon, wie die Grundkoordination aussieht, müssen die Beteiligten von dieser ausgehend die Zusammenarbeit zwischen Anwendung und Entwicklung etablieren.

Je Fallstudie ist es vom **Anwendungsbereich** abhängig, welche Möglichkeiten für die Softwaregestaltung bestehen. Das betrifft den Anteil der Datenverarbeitung im Anwendungsbereich und ob sie einen gesamten Prozess gestalten kann oder nur Teile davon. Von beidem hängt erstens ab, was der Kern der Rationalisierungsmöglichkeit ist: z.B. entweder eine weitgehende Automatisierung der Datenverarbeitung (so dass keine Anwendenden mehr eingreifen müssen) oder die automatisierte Steuerung von Arbeitskräften z.B. mit mobilen Endgeräten. Zweitens hängt vom Anwendungsbereich ab, wie komplex die abzubildenden Prozesse und Wissensdomänen sind, z.B. ob die Software einen Prozess abbildet, der mehrere Abteilungen betrifft oder nur ein Team in einem EVU.

Aus Sicht der Softwaregestaltung wird die **Arbeitsteilung** zwischen Anwendung und Programmierung zur Wissensgrenze, die überwunden werden muss. Sie kann innerhalb oder zwischen Firmen bestehen. Es gibt Fälle, in denen Anwendung, Gestaltung und Programmierung Teil einer Organisation sind, aber sie Abteilungsgrenzen trennen (INTERN1, INTERN2). Es gibt Fälle, die interdisziplinär arbeiten und in denen die Wissensgrenzen nur noch zwischen einzelnen Beschäftigten bestehen – im STARTUP, in-

nerhalb von interdisziplinären Teams in der Softwarefirma der Fallstudien PAKET und KOOP3 oder den IT-DL von KOOP1, KOOP2 und KOOP3. Bei den zuletzt genannten fünf Fällen betrifft die engere interdisziplinäre Zusammenarbeit jedoch Gestaltung und Programmierung. Anwendende sitzen vorwiegend (außer jene der BPO-Bereiche von IT-DL und Softwarefirma) in den EVU (PAKET, KOOP1, KOOP2, KOOP3) oder sind aus den Kreisen ausgeschlossen, in denen schwerpunktmäßig die Softwaregestaltung stattfindet (STARTUP).

Von der **Grundkoordination** ausgehend etablieren die Beteiligten die Softwaregestaltung als soziotechnische Netzwerkarbeit. Als primäre Koordinationsformen zeigen sich in den Fällen zwischen Anwendung und Entwicklung Markt, Hierarchie oder Netzwerk. Egal welche Koordination vorliegt, die Beteiligten müssen immer einen Weg finden, um für die Softwaregestaltung zusammenzuarbeiten. Es reicht aus Sicht der vorliegenden Arbeit nicht aus, dass mehrere Organisationen oder Mitarbeitende einer Organisation quer zur Hierarchie und Abteilungen in einer Matrixorganisation zusammenarbeiten, um von einer Netzwerkkoordination zu sprechen. Denn zur Grundkoordination von Markt, Hierarchie oder Netzwerk gehören in der Empirie immer auch noch Kontrollelemente wie IT-Budget, Service Level Agreements (SLA), interne Projektverträge oder Quellcode-Reviews. Zum Beispiel sind die bei KOOP1 bestehenden Marktbeziehungen zwischen den Firmen durch SLA detailliert vertraglich fixiert. Bei KOOP3 hingegen bestehen zwar auch Marktbeziehungen zwischen IT-DL und Softwarefirma. Jedoch spielen Verträge eine untergeordnete Rolle, weswegen die Grundkoordination in diesem Fall unterm Strich das Netzwerk ist. So existieren in den Fallstudien gemischte Koordinationsformen, wie dies auch schon die Forschung zu IT-Projekten (6.3) oder Autoren wie Lamoreaux/Raft/Temin (2003), Bradach/Eccles (1989) oder Wiesenthal (2000) allgemein festgestellt haben. Was zeichnet nun eine Netzwerk-Grundkoordination aus? Sie zeichnet aus, dass wie bei STARTUP und KOOP3 ein Arbeitsprozess der Softwaregestaltung aus Rollen, Abläufen und interpersonalen Beziehungen existiert, der ungehindert von hierarchisch-formalen Wegen, vertraglichen Einschränkungen oder Marktkalkülen für die Zusammenarbeit zwischen Anwendung und Programmierung sorgt.

Die IT-Abteilung der softwareanwendenden Organisationen ist in den meisten Fallstudien Teil der Arbeitsteilung zwischen Anwendung und Entwicklung. Sie ist für den Betrieb der IT-Systeme zuständig, stellt die Programmierenden zur Verfügung, koordiniert die als Externe eingebundenen Programmierenden und in einigen Fällen sind die internen Softwaregestaltenden ihr zugeordnet. Auch wenn in einzelnen Fällen die Fachbereiche eine bessere Zusammenarbeit von ihr erwarten, sind in den Fallstudien die IT-Abteilungen insgesamt kooperativ, was die Softwaregestaltung anbelangt.

Anders als in manchen anderen Untersuchungen (vgl. Ortmann et al. 1990, Symon 2000, Silva 2005) zeigen die Fallstudien nur wenige Interessenkonflikte oder Machtkämpfe. Nur bei INTERN1 hätte der Fachbereich gern die Hoheit bei der Softwaregestaltung, bekommt sie jedoch von der IT-Abteilung nicht. Die Anforderungsmanagerin aus dem Fachbereich muss sich den Entscheidungen des in der IT-Abteilung angesiedelten Haupt-Product-Owners fügen. Interessenkonflikte oder Machtkämpfe existieren, wie dies auch einige Studien thematisieren (vgl. Peled 2001, Flecker/Holtgrewe 2008, Kaniadakis 2012, Mezihorak 2018), mehr zwischen IT-DL und EVU. Wobei es nur bei KOOP2 zum Bruch kommt und ein EVU gar nicht mehr mit dem IT-DL arbeitet. Bei PA-

KET thematisieren EVU zwar die Abhängigkeit von der Softwarefirma. Allerdings geht es dabei mehr um die Qualität und weniger um Kosten und es spricht keine der befragten Personen davon, die softwarezuliefernde Firma zu wechseln. Sonst zeigen die Fallstudien, dass die Organisationen auf unterschiedlichen Wegen die Kooperation zwischen IT-DL und EVU aufrechterhalten, z.B. durch regelmäßige Treffen oder Mediatoren. Bereits die Forschung zum IT-Alignment schlägt regelmäßigen Austausch zwischen IT-Abteilung und Fachbereichen, Co-Lokation, gemeinsame Planung, gemeinsame Projekte und Bildung von sozialem Kapital zur besseren Zusammenarbeit vor (vgl. Reich/Benbasat 2000, Chan/Reich 2007, Masak 2006, Schlosser et al. 2015, Valorinta 2011). Ein Befragter von EVU2 der Fallstudie KOOP2 thematisiert, dass die IT nicht die Probleme der Fachabteilungen kennen würde, und weist damit auf ein mangelndes IT-Alignment hin.

Die **Softwarearchitektur** hat sich in allen Fällen als prägend für die Softwaregestaltung erweisen: Erstens entscheidet ihre Aufteilung z.B. in Module darüber, wie sich die Arbeit innerhalb und zwischen den Organisationen verteilt. Zweitens bestimmt ihr softwaretechnischer Zuschnitt (Standard, individuell), welche Abhängigkeiten zwischen Organisationen oder Teilen einer Organisation bestehen.

Die Arbeitsteilung bestimmt die Softwarearchitektur durch ihren Aufbau vor allem für die Programmierenden. In den Softwarefirmen, den IT-DL und den EVU sind die Programmierenden spezialisiert und immer für bestimmte Teile der Software zuständig. Die dazugehörigen Fachleute mit dem branchenspezifischen Wissen sind ebenso meist spezialisiert auf einzelne Teile der Software und arbeiten enger mit den jeweiligen Programmierenden zusammen. Insofern ist einerseits Conways Law richtig (vgl. Conway 1968) und die Entwicklungsorganisation prägt die Architektur (Spiegelungshypothese). Wobei in den Fallstudien die Architektur eine Ausgangsbedingung ist, welche die Beschäftigten fortführen und nicht ändern (z.B. gibt sie SAP vor oder die IoT-Softwarefirma). Andererseits gilt die Spiegelung nicht immer. Bei INTERN1 unterbindet die Anforderungsmanagerin, dass Programmierende Insellösungen schreiben, d.h. mehrmals das Gleiche programmieren. Bei KOOP1, KOOP2 und KOOP3 tauschen sich mehrere Organisationen regelmäßig darüber aus, was sie individuell und was als Standard gestalten. Das heißt, hier findet Kommunikation über die Softwaregestaltung jenseits des für ein Modul zuständigen Teams aus Gestaltenden und Programmierenden statt. Wobei wir schon beim zweiten Punkt wären:

Die Softwarearchitektur prägt in der Mehrzahl der Fälle durch ihren Zuschnitt (individuell, Standard) und die daraus resultierenden Abhängigkeiten den Kommunikationsaufwand und welche Kommunikationswege zusätzlich zur Arbeitsteilung zwischen Anwendung und Entwicklung existieren müssen. Wenn z.B. mehrere EVU sich darüber abstimmen, wie sie einen gemeinsamen Teil der Standardsoftware gestalten, dann etablieren sie zusätzliche Strukturen wie in der Fallstudie KOOP1 ein Anforderungsmanagement. Diese Kommunikationswege spiegeln sich allerdings nicht in der Softwarearchitektur. Vielmehr gehen sie auf Abhängigkeiten zwischen unterschiedlichen Stakeholdern eines gemeinsam gestalteten Softwareteils zurück.

Letztendlich ist der Literatur zuzustimmen, dass Conways Law die Kommunikationswege nicht festlegt und umgangen werden kann, es aber trotzdem noch prägend ist (vgl. Colfer/Baldwin 2016, Hvatum/Kelly 2005). Die Kommunikationswege weichen

vor allem dann vom Aufbau der Softwarearchitektur ab, wenn Organisationen einen gemeinsamen Standard gestalten und sich deswegen abstimmen müssen oder auf mehrere Organisationen verteilte Anwendende an einer Software mitgestalten, z.B. via Chats oder Ticketsysteme. Wie auch in der Forschung schon festgestellt, können sich durch eine veränderte Architektur die Abhängigkeiten in einer Organisation verändern und Abstimmungen notwendig machen (vgl. Remer 2008). Das ist bei den Fallstudien wie KOOP1, KOOP2 und KOOP3 vor allem dann der Fall, wenn sich ein EVU entscheidet, eine Software individuell zu gestalten, und sich somit nicht mehr mit anderen abstimmen muss. Das unterstreicht die Bedeutung des softwaretechnischen Zuschnitts als ein Kernproblem der Softwaregestaltung und als eigenständige Kategorie, um den Arbeitsprozess der Softwaregestaltung zu beschreiben. In den Fallstudien verhandeln die Firmen kontinuierlich darüber, teilweise auch bezogen auf einzelne Anforderungen, ob sie diese individuell oder als Standard umsetzen.

Was Conways Law nicht berücksichtigt, aber sich für die Kommunikation als entscheidend in den Fallstudien herausgestellt hat, sind die zwischenmenschlichen Beziehungen. Ein befragter Architekt stellt einen Zusammenhang mit der Softwarearchitektur her:

»Unabhängig der vielleicht potenziell möglichen Kommunikationsformen ist der Habitus von Abteilungsleitern oder auch ganzen Abteilungen das Wesentliche dafür, ob man Conways Law sehen kann oder nicht. Ich glaube, dass das quasi unabhängig des Mediums ist. Wenn sich zwei Leute mögen, dann siehst du das quasi in der Software, weil die gut funktioniert und gut zusammenarbeiten. Und ich habe auch schon Momente gehabt, da konnte ich in der Software den Weg um die Abteilung herum sehen.« (Software-Architekt)

In der Fallstudie KOOP2 führen die schlechten Beziehungen zwischen dem IT-DL und einem EVU dazu, dass Letzteres sich nicht mehr von ihm betreuen lässt, wodurch auch ein gemeinsamer Standard scheitert. Insofern zeigen die Fallstudien, dass es von einer gelungenen Kooperation abhängt, gemeinsam eine Standardarchitektur hinzubekommen.

Wie der im nächsten Punkt behandelte Teil des Analyserahmens zeigt, sind kommunikative Beziehungen grundsätzlich Teil des Arbeitsprozesses der Softwaregestaltung.

8.6.2.2. Soziotechnischer Arbeitsprozess der Softwaregestaltung

Die Fallstudien vertiefen das Verständnis darüber, wie die im 6. Kapitel beschriebenen vier Ebenen der soziotechnischen Netzwerkarbeit die Arbeit der Softwaregestaltung kontrollieren und dadurch Arbeitskraft transformieren. Neben der soziotechnischen Konstellation aus Arbeitsteilung, Grundkoordination und Architektur basieren die vier Ebenen auf dem Arbeitsprozess der Softwaregestaltung: 1. Ein von Rollen und deren Erwartungen kontrolliertes Handeln der Softwaregestaltenden. 2. Ein mehr oder weniger formaler Ablauf mit Gestaltungsnetzwerk, Feedbackschleifen, situativen Anpassungen und Lernprozessen. 3. Interpersonale, kommunikative Beziehungen als Grundlage für Kommunikation und Kooperation. 4. Softwarewerkzeuge und die gestaltete Software selbst, die das Handeln ermöglichen und einschränken, als gemeinsame Bezugspunkte,

die die Arbeit sämtlicher Beteiligten koordinieren und den Wissensaustausch und die Kommunikation dokumentieren. Zudem zeigen die Fallstudien eine Flexibilität der Ebenen von Rollen, Ablauf, kommunikativen Beziehungen und digitalen Werkzeugen. So kompensieren in manchen Fällen z.B. engagierte Beschäftigte fehlende formalisierte Abläufe oder ersetzt direkte Kommunikation jene rein formale über ein Ticketsystem.

Der Fallvergleich legt für die Kategorie der **Rollen** dar, dass, statt en détail die Arbeit der Softwaregestaltenden in dezentralen oder zentralisierten Arbeitsprozessen festzulegen, es darum geht, dass Arbeitende Erwartungen erfüllen. Davon existieren mehrere: Softwaregestaltende sollen bei Bedarf mehrere Rollen übernehmen, Rollen wechseln, unterschiedliche Rollen in den unterschiedlichen Arbeitsprozessen von Softwaregestaltung, -anwendung und -programmierung einnehmen. Dazu gehört, dass sich die Softwaregestaltenden in den Arbeitsprozess der Softwaregestaltung, wie er jeweils ist, einführen, sich entsprechend anpassen, ggf. selbst herausfinden, was zu tun ist, die Rolleneinhaltung mit anderen verhandeln und sie gegenseitig kontrollieren – z.B. dadurch, dass Person A an Person B Erwartungen formuliert. Der Arbeitsprozess ist eine Sozialisationsinstanz für das Lernen von Rollen, wobei in den meisten, aber nicht in allen Schulungen Methoden wie Scrum oder IT-Projektmanagement Ausgangspunkte dafür sind. Eine weitere Erwartung ist, mit den Graubereichen der Verantwortlichkeiten und mit Erwartungs-/Rollenkonflikten umzugehen und die eigenen Spielräume zu nutzen. Das röhrt daher, dass in vielen Fällen eine Mischung aus hierarchie-/markt- und rollenbasierter Organisation vorliegt. Zum einen sind Softwaregestaltende zwar oftmals Teil von Hierarchien, in denen z.B. Führungskräfte Entscheidungen treffen. Zum anderen sind die Softwaregestaltenden selbst nicht hierarchisch organisiert, weswegen sie Entscheidungen im Dialog treffen oder Erwartungen abgleichen müssen, ohne auf Hierarchien zur Durchsetzung zurückgreifen zu können. Somit erlauben Rollen es, situativ und unabhängig von Hierarchien zu agieren. In Fallstudien wie KOOP3 oder STARTUP gibt es gar keine formalen Hierarchien innerhalb des Arbeitsprozesses der Softwaregestaltung.

Für den **Ablauf** der Softwaregestaltung zeigen die Fallstudien, dass dieser zwar formalisiert ist – aber unterschiedlich stark. Er ist mit einem Netzwerk an Beziehungen für die Anforderungsgewinnung verbunden. Er ist vereinbar mit verschiedenen Handlungsorientierungen, wobei der Primat der Kooperation gilt, um über Hierarchien und Marktbeziehungen hinweg zusammenzuarbeiten. Der Ablauf stellt in den Fallstudien Feedbackschleifen zwischen Softwareentwicklung, -anwendung und -gestaltung her und sicher: ob durch Treffen, E-Mails, Tests, persönliche Gespräche oder Methoden wie Prototyping oder Resonanzgruppen. Er fügt sich in die jeweilige soziotechnische Konstellation ein. Was seine Ausgestaltung anbelangt, ist er in sämtlichen Fallstudien durch Lernprozesse geprägt und kann temporär oder langfristig existieren. Die Informatilität des Ablaufs besteht mal in Ad-hoc-Absprachen via Chat, direkttem, persönlichem Austausch oder dass nicht formal festgelegt ist, wie Entscheidungen gefällt werden. Seien es Entscheidungen über den Einsatz von Methoden, wer an einem Termin teilnimmt, die Priorisierungen und Aufwandsschätzung von Tickets, wie viele Anforderungen die Programmierenden in einem bestimmten Zeitraum erledigen können, oder die konkreten Verantwortlichkeiten einer Rolle. Die in der Forschung untersuchten Lernprozesse bei der Steuerung von organisationsübergreifender Zusammenarbeit (vgl. Mola et al. 2017, van Fenema/Keers/Zijm 2014) zeigen sich vor allem bei KOOP1. Bei KOOP2 ist die Koope-

ration daran gescheitert, dass die Beteiligten keine Mittel fanden, um Konflikte zu lösen und Erwartungen abzugleichen.

Die Kategorie der **kommunikativen Beziehungen** zeigt, dass ein bestimmtes Maß an Beziehungsfähigkeit Teil des Arbeitsprozesses der Softwaregestaltung ist und Rollen, Abläufe oder digitale Werkzeuge nicht ausreichen. Damit ist Kommunikation kein abstrakter, rein kognitiver Informationsaustausch. Sie gründet auf zwischenmenschlichen Beziehungen, die Befragte mit Begriffen wie »partnerschaftlich«, »kooperativ«, »auf Augenhöhe«, »Familie«, »Offenheit« oder »Geben und Nehmen« beschreiben. Dazu gehören langfristige Beziehungen und direkte Kommunikation zwischen zwei Personen verschiedener Abteilungen oder Organisationen trotz Hierarchien oder Marktbeziehungen. Dazu gehört eine gemeinsame Sprache, um die interdisziplinären Wissensgrenzen zwischen IT- und Energiewirtschaft zu überwinden. Aber auch gemeinsame Begriffe, wenn mehrere Teams innerhalb eines EVU oder mehrere EVU zusammenarbeiten, gehören dazu, ebenso wie die Fähigkeit, sich verständlich auszudrücken. Wenn Beziehungen so eine wichtige Rolle dafür spielen, dass miteinander zum Zwecke der Softwaregestaltung geredet wird, ist es naheliegend, dass in einem Fall ein professioneller Mediator unterstützt (KOOP1). Erstaunlich ist, dass nur in einem Fall eine Befragte eine Schulung zur Förderung der Kommunikation erwähnt (EVU2, KOOP2²³). Damit sind die Ergebnisse zum einen anschlussfähig an die Forschung darüber, dass Projektstrukturen allein für die kooperative Zusammenarbeit nicht ausreichen und Beziehungsfaktoren wie Kooperationsbereitschaft, Sozialkompetenz (vgl. Rüegg-Stürm/Young 2001) oder Vertrauen (vgl. Powell 1990) notwendig sind und informelle Strukturen der Kooperationen die Organisationen gezielt durch verschiedene Methoden erst etablieren müssen (vgl. Bolte/Porschen 2007).

Was die **digitalen Werkzeuge** anbelangt, sind in sämtlichen Fällen Ticketsysteme und zudem oftmals Entwicklungs- und Testumgebung Teil der betrieblichen Realität. Manchmal werden sie kombiniert mit Chat-Gruppen, Excel-Dateien oder Microsoft Sharepoint und natürlich: E-Mails. Je nach Fall fungieren sie neben der Koordination zur Kontrolle der Arbeit durch Kundschaft oder Führungskraft, weil sie für diese transparent machen, welche Anforderungen es gibt, wie lang ihre Umsetzung dauert und welchen Status sie haben. Aber der Zeitrhythmus wird nicht durch einen digital-maschinellen Takt bestimmt, sondern durch Erwartungen von Kundschaft und Führungskraft, die Aufwandsschätzung durch einzelne Mitarbeitende wie Programmierende, IT-Beratende oder Product Owner, durch die Ad-hoc-Umsetzung einer dringenden Anforderung, Umsetzungsfristen durch die Regulierung, die Dauer eines Sprints in Scrum oder durch in Verträgen (SLA) festgelegte Reaktionszeiten für Tickets. Durch die Dokumentation der Softwaregestaltung im Ticketsystem oder anderen Softwarelösungen und dem Quellcode selbst entsteht Transparenz über sie. Auch wenn keine befragte Person von einer individuellen Leistungskontrolle berichtet, sind die Voraussetzungen dafür gegeben, dass der Panoptikum-Effekt (bzw. »electric panopticum« nach Zuboff 1988) wirkt. Wie im 6. Kapitel über die Rolle von Software bei der Arbeitskontrolle ausgeführt (6.4.2.1), zeigen die Fallstudien, dass nicht nur eine Software eine bestimmte Funktion

23 Die Befragte hat die Working-Out-Loud-Methode (WOL) genannt.

erfüllt: Vielmehr verwenden die Beteiligten im Arbeitsprozess mehrere Softwarelösungen gleichzeitig und permanent und die Software wirkt sowohl ermöglicht als auch einschränkend, wobei die Besonderheit in der Wissensarbeit die Objektkontrolle durch Software (vgl. Rennstam 2012) ist.

Über den **softwaretechnischen Zuschnitt** als Teil der softwaretechnischen Gestaltungsmöglichkeiten (siehe 3.2) entscheidet der Arbeitsprozess der Softwaregestaltung. Er ist ein kritischer Punkt im Arbeitsprozess, wenn es um Fragen von Konflikten, Effizienz und Partizipation geht. Es zeigen sich in den Fallstudien Unterschiede, wo und wer über den Zuschnitt entscheidet, wie dadurch Synergien gehoben und wie Prioritäten gesetzt werden. Es können sich verschiedene Gruppen bei der Entscheidung gegenüberstehen und verhandeln müssen. Die Entscheidungsprozesse können institutionalisiert sein oder an einzelnen Rollen hängen, die bspw. Synergien bei der Programmierung erkennen (oder eben nicht). An der Fallstudie KOOP1 zeigt sich, dass mehrere EVU dank des gemeinsamen, ausgefeilten Anforderungsmanagements über mehr als 10 Jahre hinweg kontinuierlich darüber verhandeln, wo sie Synergien sehen und was sie individuell machen wollen. Das heißt, mehrere EVU sind fähig, organisationsübergreifend Synergien zu erkennen. Bei INTERN2 gibt es zum Erkennen von Synergien die Anforderungsrunde mehrere Fachbereiche. Im EVU3 von KOOP1 ist es Aufgabe des Prozessmanagers, dies für die beiden getrennten Anwendungsbereiche von Privat- und Geschäftskundschaft in seiner Abteilung zu tun. Die Anforderungsmanagerin von INTERN1 versucht zu verhindern, dass die Programmierenden Insellösungen schreiben.

Die **Flexibilität der Kommunikation** in der Softwaregestaltung (6.4.4) zeigt sich auch in den Fallstudien. Es gibt ganz unterschiedliche Faktoren, die beeinflussen, wie die Kommunikation in den Fallstudien abläuft. Bei KOOP3 ist die IoT-Software modular, weswegen EVU1 und IT-DL eigenständig Erweiterungen programmieren können, was die Kommunikation mit der Softwarefirma reduziert. Bei INTERN1 kann der Fachbereich die Anforderungen mittlerweile so formulieren, dass die Programmierenden verstehen, was gemeint ist, wodurch weniger direkte Kommunikation notwendig ist. Bei INTERN2 gibt es Treffen vor der Anforderungs runde, um Anforderungen so auszuformulieren, dass der Austausch nicht nur via Ticketsystem stattfindet. Eine frühzeitige, direkte Kommunikation soll ein späteres Hin und Her via Ticketsystem vermeiden. Bei KOOP1 ist das Anforderungsmanagement formalisiert, eingespielt und es gibt spezialisierte Rollen dafür. Bei KOOP2 im EVU2 muss der Projektmanager erst noch die Abläufe etablieren und die Beteiligten des Projekts eine gemeinsame Wissensbasis aufbauen, um effektiv Anforderungen erarbeiten zu können. Damit zeigen sich auch die von Srikanth/Puranam (2014) genannten drei Elemente aus Common Ground, modularer Softwarearchitektur und direkter Kommunikation, die je nach Kontext unterschiedlich die Kommunikation bestimmen.

Wie in der Forschung bereits beschrieben, kommen vielfältige Kommunikationskanäle zum Einsatz (vgl. Heidenreich/Kirch/Mattes 2008) und die Beschäftigten kombinieren direkte Kommunikation mit digitaler (Chats, Ticketsysteme, geteilte Dokumente). Wobei alle Organisationen verbale Kommunikation, die für die Programmierung relevant wird, verschriftlichen – sei es im Ticketsystem, E-Mails oder anderen Softwarelösungen zur Dokumentation. Auffällig ist, dass viele Befragte die direkte Kommunikation in der Softwaregestaltung bevorzugen.

8.6.2.3. Folgen für die Arbeit der Softwaregestaltenden

Die Fallstudien zeigen, dass hinter der Softwaregestaltung eine eigenständige Gruppe von Beschäftigten steht, was Beschäftigungssystem, Kontrolle und Wissen anbelangt. So verändert der Arbeitsprozess der Softwaregestaltung die EVU und die Energiewirtschaft allein schon dadurch, dass ausgehend von ihm drei Gruppen zu unterscheiden sind: Anwendende, Gestaltende und Programmierende. Auch wenn die Arbeitsprozesse der Softwareanwendung und -programmierung nicht Gegenstand der vorliegenden Untersuchung sind, dienen die Aussagen aus den Interviews zur Arbeit der Anwendenden und Programmierenden dazu, den Blick für die Eigenheiten der Softwaregestaltung zu schärfen und zu prüfen, inwiefern sie sich voneinander unterscheiden. Doch zunächst etwas über die Arbeit der Softwaregestaltenden:

In den Fallstudien fällt auf, dass die Führungskräfte die Softwaregestaltenden vor allem anhand ihrer Ergebnisse und nur bei Auffälligkeiten direkt **kontrollieren**. Führungskräfte kümmern sich um den Rahmen: Sie legen je nach Fall das Budget fest, stellen Mitarbeitende ein, initiieren Projekte oder klären Konflikte innerhalb (z.B. mit anderen Führungskräften oder Teams) oder zu anderen Organisationen (Softwarefirma, IT-DL, EVU). Softwaregestaltende relativieren öfters ihre Arbeitsbelastung (z.B. weniger schlimm als in der Beratung) und stellen sie als verhandelbar dar. Sie sprechen öfters davon, dass sie intrinsisch motiviert sind. Sie sind hierarchisch höhergestellt als Anwendende. Dabei prägt die Grundkoordination die Kontrolle ihrer Arbeit: Von ihr hängt ab, ob mehr das Management (Hierarchie), die Kundschaft (Markt) oder die Kollegenschaft (Netzwerk) kontrolliert. Sie sind einer *qualitativen Intensivierung* ausgesetzt, weil sie in vielen Fällen IT-, energiewirtschaftliches und Methodenwissen kombinieren oder zumindest mit den jeweiligen Expert:innen kommunizieren müssen, öfters gleichzeitig oder wechselnd in unterschiedlichen Projekten und an verschiedenen Softwarelösungen arbeiten. Die *quantitative Intensivierung* ist bei den Gestaltenden nicht eindeutig. Die einen sprechen davon, dass sie phasenweise mal mehr und mal weniger zu tun haben, die anderen davon, dass sie selbst Grenzen setzen müssen und können, wieder andere davon, dass sie geregelte Arbeitszeiten haben.

In einigen Fallstudien sind die Softwaregestaltenden in puncto **Beschäftigungssystem** Teil eines IT-DL oder einer Softwarefirma und arbeiten mit mehreren EVU zusammen. Dadurch aber auf eine höhere Arbeitsbelastung zu schließen, lassen die Aussagen der Befragten nicht zu. Denn diese stellt sich für einzelne Befragte egal welcher Gruppe als sehr unterschiedlich dar und ist nicht einfach auf einen Faktor zurückzuführen, wie z.B. mit mehreren EVU zusammenzuarbeiten oder in einem kleinen EVU angestellt zu sein. Sonst zeichnen sich die Gestaltenden durch ein geringeres Interesse an einer disziplinarischen Karriere aus, mehr (Arbeits-)Marktmacht und eigenen Arbeitsmärkten. Die Organisationen setzen sie flexibel ein – ob innerhalb einer Matrixorganisation oder in anderen Organisationen. Vor allem tun die Organisationen das, wenn sie spezifisches Methodenwissen haben (z.B. Scrum) oder tiefere Kenntnisse einer auch in anderen Branchen genutzten Softwareumgebung (z.B. SAP). Das zeigt sich bei KOOP1 daran, dass die Softwaregestaltenden das IT-DL als Durchgangsstation in ihrer Karriere sehen. Wenn EVU anfangen, Software zu gestalten, gibt es teilweise neue Karrierewege für Softwaregestaltende (»Kompetenzkarriere«). Wobei die herkömmlichen Karrieremuster als Aufstieg auf einer Leiter in einer Hierarchie mit disziplinarischer Verantwortung weiter-

bestehen. Das IT-DL von KOOP2 hat schon die Kompetenzkarriere eingeführt, was dem EVU2 aus der gleichen Fallstudie schwerer fällt. Schafft ein EVU es nicht, entsprechende Karriereperspektiven für Softwaregestaltende zu schaffen, verlassen diese wie im Falle des EVU2 von KOOP2 die Organisation. Wenn ein IT-DL die Softwaregestaltung für mehrere EVU organisiert, dann kann es einfacher mehr Möglichkeiten für die Beschäftigten bieten (unterschiedliche Projekte, unterschiedliche EVU). Bei STARTUP gibt es keine formalen Hierarchien und die Segregation zwischen den Anwendenden und dem Rest fällt auf: Die Anwendenden sind 450-Euro-Kräfte und haben nicht studiert.

Der Wissensaustausch im Gestaltungnetzwerk entscheidet, welches Wissen in die Software einfließt. Das Lernen ergibt sich situativ in einer Praxis, in der Software(-Objekte) eine wichtige Rolle spielt(en). Die Softwaregestaltenden der unterschiedlichen Fallstudien verfügen über sehr unterschiedliche interdisziplinäre Wissensstände. Das Spektrum reicht von Softwaregestaltenden, die sich vor allem mit der Koordination der Softwaregestaltung beschäftigen und nur noch wenig inhaltliches Wissen benötigen, bis hin zu IT-Beratenden, die nicht nur Anforderungen aufnehmen, sondern auch gleich noch umsetzen. Bei der **Wissensverteilung** der Softwaregestaltung kann von einer interdisziplinären Praxisgemeinschaft gesprochen werden. Die Praxisgemeinschaft zeigt sich zum einen dadurch, dass jemand, der davon ausgeschlossen ist, nicht mehr Softwaregestaltung betreiben kann und von anderen abhängt, weil ihm das Wissen fehlt. Zum anderen zeigt sie sich dadurch, dass sich die beteiligten Personen in kontinuierlichem Austausch befinden, um Software zu gestalten. So entstehen unterschiedliche Praxisbiografien und damit unterschiedliche Lernbiografien für jeden einzelnen Beschäftigten. Hohmann hat von einem Gestaltungnetzwerk gesprochen, welches das interdisziplinäre Wissen hat, um die SAP-Standardsoftware über das Implementierungsprojekt hinaus zu gestalten. In Anlehnung an Wengers *Community of Practice* (1999) kann diese als *Community of Practice and Software Objects* bezeichnet werden. Denn für das, was die Softwaregestaltenden wissen, sind weniger Schulungen entscheidend und vielmehr die Praxis mit anderen und der Software. Teil der Wissensverteilung der Softwaregestaltung sind die verwendeten, softwarebasierten Werkzeuge und die Software selbst. In ihnen materialisiert sich das Wissen (z.B. Quellcode, Ticketsystem, E-Mails, Dokumentationen von Umsetzungen und über Funktionalitäten). Nicht nur Personen sind Träger des Wissens, sondern auch die Software mit dem entsprechend hinterlegten Wissen. Auffallend bei den Interviews war, dass sowohl Anwendende, Programmierende und Gestaltende sagen, dass Learning by Doing für sie wichtig ist – ob aus Dokumenten oder mit der Software selbst. In vielen Fallstudien sind Führungskräfte nicht Teil der Softwaregestaltung und haben entsprechend wenig Wissen über sie oder Softwareentwicklung im Allgemeinen.

Sonst gibt es Besonderheiten je Fall, die aber alle unterschiedliche Varianten der interdisziplinären Praxisgemeinschaft sind: Bei STARTUP ist die gesamte Organisation durch die Kreise darauf ausgerichtet, permanenten interdisziplinären Wissensaustausch zu ermöglichen und eine gemeinsame Wissensbasis zu schaffen. Die Anwendenden sind in diesem Fall weitestgehend ausgeschlossen. Was die Softwaregestaltung anbelangt, ist der Großteil der Anwendenden selbst in Fällen wie INTERN1 vom Wissensaustausch ausgeschlossen, wo sie aktiv in die Softwaregestaltung einbezogen werden. Ihr Wissen ziehen Softwaregestaltende und Programmierende nur temporär für die

Softwaregestaltung heran. Bei KOOP1 besteht die Interdisziplinarität zentral im IT-DL und durch langfristige Zusammenarbeit mit den EVU. Bei KOOP2 fällt bei der internen Softwaregestaltung in EVU2 in den IT-Projekten auf, dass interdisziplinäres Wissen zwischen IT und Fachbereichen fehlt. Das ist eine Folge davon, dass das EVU Teile der Softwaregestaltung erst seit einigen Jahren wieder vom IT-DL zurückverlagert hat. Die Softwarefirma von PAKET besteht aus Entwicklungsteams (Programmierenden, Tester, Teamleitung), die auf energiewirtschaftliche Fachbereiche und die dazugehörigen Softwaremodule der Standardsoftware spezialisiert sind. Bei KOOP3 hat das IT-DL und die IoT-Softwarefirma Wissen über IoT und Implementierungskompetenz für die Standardlösung, wobei das IT-DL das branchenspezifische Wissen aus den Implementierungsprojekten mit EVU in Form von Anforderungen an die IoT-Firma weitergibt.

Im Vergleich zu den Softwaregestaltenden fällt bei den Programmierenden und Anwendenden auf, dass sie auch eigenständig arbeiten, die Führungskräfte nur in Ausnahmefällen eingreifen und die Arbeitsergebnisse kontrollieren. Zudem sind sie wie die Softwaregestaltenden Teil eines softwarezentrierten Arbeitsprozesses mit anderen Schwerpunkten, welche Software sie für ihre tägliche Arbeit hauptsächlich verwenden (ERP-System im Kund:innenservice, mobile Endgeräte in der Instandhaltung, Entwicklungsumgebung und Ticketsystem in der Programmierung etc.). Auch für Programmierende und Anwendende prägt die Grundkoordination von Hierarchie, Markt und Netzwerk die Kontrolle. Die **Anwendenden** sind hierarchisch und von der Bezahlung Programmierenden und Gestaltenden untergeordnet. Die Anwendenden sind meist passiv und reagierend: Sie müssen sich auf Änderungen einstellen oder erfahren erst nach Fertigstellung, was sich geändert hat. Ihre Arbeit ist in den meisten Fällen das, was die Software nicht erledigen kann bzw. noch nicht in dieser abgebildet ist. Das bedeutet auch, dass sie fehlerhafte oder nicht ausgereifte Software kompensieren müssen. Sie fungieren als Puffer für mangelhafte Softwareentwicklung. Für einen Teil der Anwendenden findet eine *qualitative Intensivierung* statt, weil sie vermehrt in Prozesszusammenhängen denken müssen (Sachbearbeitende), mehrere Fachgebiete (Key User:innen) oder mehrere Netzgebiete und Sparten kennen müssen (Monteur:innen). Arbeiten Anwendende im BPO-Bereich für mehrere EVU, ist mit einer *quantitativen Intensivierung* zu rechnen. Wobei dies bei einigen EVU auch unabhängig davon aufgrund von Personalmangel oder phasenweise durch IT-Projekte entsteht. Die Arbeit der **Programmierenden** wird durch die Tests anderer und/oder durch Code-Reviews von Kolleg:innen geprüft. Es ist durch die Ticketsysteme transparent, was sie abgearbeitet haben, und falls es regelmäßige Treffen wie Daily bei Scrum gibt, müssen sie den Teammitgliedern Rede und Antwort stehen. Ihre Spezialisierung und die meist schwierige Abschätzung des Arbeitsaufwandes geht mit einer fehlenden detaillierten Kontrolle von außen einher. Eine *qualitative Intensivierung* existiert bei den befragten Programmierenden, weil sie fachliches mit IT-Wissen kombinieren und wenn sie in verschiedenen Projekten, Rollen oder für mehrere EVU tätig sind. Wie bei den Gestaltenden ist es bei den Programmierenden schwierig, pauschal von einer *quantitativen Intensivierung* zu sprechen. Sie arbeiten in den untersuchten Fällen meist Programmieraufgaben ab, für die Prioritäten vorgegeben sind. Das lässt Spielraum, eigene Grenzen zu setzen. Eine

Intensivierung ist aber nicht ausgeschlossen, wenn z.B. eine fixe Deadline für eine zu programmierende Funktionalität existiert.

8.6.2.4. Folgen für die Arbeitsgestaltung in den EVU

Die Fallstudien zeigen, dass eine Folge der Softwaregestaltung ist, dass sie die Arbeit der Softwareanwendung gestaltet. Sie ist damit Teil der Kontrolle der Softwareanwendung. Dies tut sie durch einen soziotechnischen Wandel von Software, Organisation und Arbeit. Dabei dienen die Kategorien von Einfluss und Konflikten dazu, das Verhältnis von Softwaregestaltung und -anwendung zu untersuchen. Der Fallvergleich konnte zusätzlich dazu vier Unterkategorien des Einflusses herausarbeiten, welche das Verhältnis der beiden Arbeitsprozesse von Softwaregestaltung und -anwendung beschreiben helfen: Kontrollverhältnis, Partizipation, Reorganisation und Ziele. Zur Partizipation der Anwendenden und einer möglichen Reorganisation der EVU hin zu einer softwaretechnischen Prozessorganisation hat das Kapitel jeweils allgemeine Thesen diskutiert, die hier noch einmal kurz zusammengefasst werden. Beim Konflikt zwischen Management und Angestellten nimmt der Betriebsrat in den Fällen eine intervenierende Funktion bei der Arbeitsgestaltung ein. Welche Rolle das Management bei der Arbeitsgestaltung durch Softwaregestaltung überhaupt noch spielt, diskutiert Abschnitt 8.6.4 weiter unten.

Was den **Einfluss** der Softwaregestaltung auf die Arbeitsgestaltung anbelangt, unterscheiden sich die Fallstudien dahingehend, ob sie für mehrere EVU und damit viele Anwendende Software gestalten oder nicht. Dabei ändert der Arbeitsprozess der Softwaregestaltung in nur wenigen Fällen nicht nur die Software, sondern auch die Organisation der Softwareanwendung (z.B. INTERN1). Bei den Fällen, in denen EVU intern Software gestalten, fallen die **Konflikte** innerhalb der Organisationen mehr auf (z.B. zwischen Unternehmens- bzw. Management- und Anwendenden-Anforderungen). Bei den Fällen, in denen mehrere Organisationen zusammenarbeiten, treten die Konflikte zwischen diesen besonders hervor. Wobei es dann vor allem darum geht, zwischen Kooperation und individuellen Alleingängen abzuwagen und die Erwartungen dementsprechend abzugleichen. Bei Fallstudien der Grundkoordination Netzwerk sind die Konflikte bezüglich der Arbeitsgestaltung schwach ausgeprägt, wie z.B. bei STARTUP. Weder unterschiedliche Marktagierende wie IT-DL, Softwarefirma und EVU noch unterschiedliche Abteilungen und deren Führungskräfte stehen sich gegenüber.

Je nach Fallstudie ist der **Betriebsrat** unterschiedlich eingebunden, fokussiert sich aber immer auf die Folgen für die Anwendenden und interveniert. Grundsätzlich ist er Teil des Konflikts in der Softwaregestaltung zwischen Beschäftigten und Management. Er kann Rahmenbedingungen setzen, kann Informationen einfordern (bei entsprechenden Betriebsvereinbarungen) und individuelle Leistungskontrolle verhindern. In den Fallstudien fehlt ihm das Wissen, Anforderungen für die Softwaregestaltung zu stellen, und er ist nicht in den Arbeitsprozess eingebunden. Findet die Softwaregestaltung intern statt, kann er dafür sorgen, dass Beschäftigte mitgestalten dürfen. Bei externer Softwaregestaltung gibt es Fälle, bei denen er bei der Auswahl von IT-DL interveniert.

Um den Einfluss der Softwaregestaltung auf die Softwareanwendung tiefergehend zu analysieren, hat das Kapitel die Unterkategorien Kontrollverhältnis, Partizipation, Reorganisation und Ziele verwendet. Erstens ist die soziotechnische Arbeitsgestaltung

des Anwendungsbereichs durch die Softwaregestaltung ein **Kontrollverhältnis**. In der Mehrzahl der Fallstudien kontrollieren die beteiligten Organisationen die Arbeitsprozesse von Softwaregestaltung und Softwareanwendung jeden für sich, z.B. als Teil von unterschiedlichen Teams oder Abteilungen. In solchen Fällen haben die Führungskräfte der Anwendung oftmals genauso viel oder gar noch mehr Macht als jene der Softwaregestaltung. Wenn die gesamte Organisation von Anfang an auf die Softwareentwicklung ausgerichtet ist, wird, wie schon unter 4.1 ausgeführt, vom Primat der Softwareentwicklung gesprochen. Das ist bei STARTUP so, wo die Fallstudie zeigt, wie Organisationen den Arbeitsprozess der Softwareanwendung zum Nutzen für die Softwaregestaltung organisieren, um die Möglichkeiten der Softwareentwicklung auszunutzen. Das bedeutet, dass die Arbeit der Anwendenden nicht nur durch die Software kontrolliert wird, sondern auch durch den Arbeitsprozess der Softwaregestaltung.

Eine direkte **Partizipation**, bei der die Anwendenden alle in die Softwaregestaltung einbezogen werden und auch genug Wissen haben, um mitzustalten, existiert in den Fallstudien nicht. Selbst dann existiert sie nicht, wenn die Anwendenden Anforderungen aufgeben können. Auch dann existiert sie nicht, wenn die EVU den Anwendungsbereich bzw. die Softwareanwendung nicht der Softwaregestaltung unterordnen (indem z.B. die Führungskraft der Softwaregestaltung hierarchisch mit jener der Softwareanwendung gleichgestellt ist). In den meisten Fällen haben Anwendende nur über Vermittler Zugang zur Softwaregestaltung oder es gibt formale Wege, Anforderungen zu stellen, bei denen letztendlich andere entscheiden, welche der Anforderungen wie umgesetzt werden. Oftmals gestalten Fachleute Software und in manchen Fällen suchen die Organisationen einen Ausgleich zwischen Anwendenden- und Unternehmensinteressen, wobei das Management das letzte Wort hat. Die drei analytischen Typen der Partizipation bei der Softwaregestaltung (siehe die Diskussion unter 8.5.3.3) untermauern diese These.

Die Arbeitsgestaltung unterscheidet sich im Fallvergleich dahingehend, welche **Ziele** in puncto Softwaregestaltung die EVU selbst festlegen können – hinsichtlich ihrer Organisation und der Kommodifizierung von Software. Bei Fragen der Organisation geht es z.B. darum, ob ein EVU von seinem Status quo aus anfängt, sich so zu organisieren, ob es selbst Software gestalten kann oder die Softwaregestaltung an einen IT-DL auslagert. Bei der Kommodifizierung geht es darum, ob eine Softwarefirma eine Standardsoftware programmiert, die viele EVU einsetzen und die damit skalierbar ist. Das können einzelne Teile einer Software, Applikationen oder wie bei STARTUP Softwareteile sein, welche eine Organisation selbst nutzt und gleichzeitig anderen anbietet. Auf jeden Fall zeigen die Fallstudien, dass Software strategisch einen Unterschied machen kann und keine Standardware wie Bürostühle oder Seife ist (siehe bereits 4.2.2). Strategisch macht Software vor allem dann einen Unterschied, wenn von ihr die individuelle Wettbewerbsfähigkeit abhängt. Das zeigt sich daran, dass es den EVU leichter fällt, einen Standard für den regulierten Netzbereich einzusetzen, der nicht dem Marktzwettbewerb ausgesetzt ist. Für den vertriebs- und wettbewerbsorientierten Lieferbereich fällt es EVU schwerer. Zuletzt betreffen die Ziele, inwiefern eine Organisation mit Software oder Softwaregestaltung Geld verdient (z.B. ob als IT-Dienstleistung oder mit einem fertigen Softwareprodukt). Die IT-DL von KOOP1, KOOP2 und KOOP3 gehören EVU, verdienen Geld mit Softwaregestaltung und Standardsoftware und die Softwarefirma von PAKET lebt allein vom Softwaregeschäft.

In den Fallstudien zeigt sich – viertens – der Einfluss der Softwaregestaltung auf die -anwendung daran, ob sich die EVU **reorganisieren**. Es gibt Fallstudien, in denen organisationale Hürden die Softwaregestaltung im jeweiligen Anwendungsbereich nicht mehr bremsen. Das bedeutet aber nicht in jedem Fall, dass EVU die Anwendung oder Entwicklung reorganisieren. Zum Beispiel verbleiben bei INTERN1 und INTERN2 die Programmierenden in der IT-Abteilung und die bestehende Arbeitsteilung zwischen Teams bleibt bestehen (inkl. getrennter IT-Budgets). In einzelnen Fallstudien ändert die Softwaregestaltung weder etwas an den Arbeitsabläufen noch den Arbeitsstellen, sondern nur etwas an der Software. Im Extremfall des Primats der Softwareentwicklung (Fall STARTUP) gab es keinen energiewirtschaftlichen Status quo und die Organisation baut sich um die Software herum auf und verändert sich mit ihr. Von Anfang an war die Organisation auf die Softwareentwicklung ausgerichtet. Um zu verdeutlichen, wie eine Reorganisation im Sinne der Softwaregestaltung innerhalb eines EVU aussehen könnte, hat die Untersuchung den Organisationstyp der auf Softwaregestaltung ausgerichteten, softwaretechnischen Prozessorganisation ausgearbeitet (siehe 8.5.3.3). Er untermauert die These, dass sich die untersuchten EVU (und wohl viele andere auch) noch gar nicht auf die Softwaregestaltung ausgerichtet haben. Sie denken zwar darüber nach, doch nutzen sie Rationalisierungspotenziale durch eine verstärkte organisationale Ausrichtung auf die Softwaregestaltung nicht – auch wenn das EVU von INTERN2 darüber nachdenkt.

8.6.3. Synthese: Rationalisierungstyp der technikentwicklungsbezogenen Rationalisierung

Die Fallstudien zeigen, dass die EVU auf unterschiedliche Weise Softwaretechnikgestaltung betreiben: mal einen Standard anpassen, mal von Grund auf etwas individuell Neues gestalten; mal durch einen IT-DL gestalten lassen, mal selbst gestalten. Eine andere Variante ist, dass sie wie bei der Fallstudie STARTUP von Beginn an Arbeit und Organisation auf eine individuelle Softwaregestaltung ausrichten. Aber egal, wie sie es machen: Indem sie die Möglichkeiten der Softwaregestaltung nutzen, rationalisieren sie dadurch die Anwendungsbereiche.

Um zu untermauern, dass Softwaregestaltung die Anwendungsbereiche rationalisiert, stellt die vorliegende Untersuchung die Hypothese auf, dass es sich bei den unterschiedlichen Ausprägungen der Softwaregestaltung in den Fallstudien um verschiedene Formen einer technikentwicklungsbezogenen Rationalisierungsform handelt. Dabei rationalisieren die Organisationen – je nach Fall EVU, IT-DL, Softwarefirmen – sowohl die soziotechnische Arbeitsgestaltung der Softwareanwendung durch die Softwaregestaltung als auch die soziotechnische Netzwerkarbeit der Softwaregestaltung selbst. Die Softwaregestaltung zu kontrollieren, ist die Voraussetzung dafür, dass die Organisationen deren Möglichkeiten der Rationalisierung zwischen Standardsynergien und individueller Differenzierung im Wettbewerb nutzen können. Damit ist nicht gesagt, dass es einen »one best way« der Rationalisierung gibt und alle Organisationen z.B. dem Prinzip der Softwareentwicklung folgen sollten. Technikentwicklungsbezogene Rationalisierung ist contingent und reflexiv und bedeutet, dass Organisationen einer Branche das Mittel der Softwaregestaltung und -programmierung für die eigenen Zwecke und Bedin-

gungen zweckorientiert im jeweiligen Kontext einsetzen. Weil sich viele Tätigkeiten und Prozesse durch Software effizienter oder überhaupt nur noch so wettbewerbsfähig erleidigen lassen, ist diese Form der Rationalisierung Teil vieler Organisationen. Sie zeichnet sich durch eine Dynamik zwischen Standard- und Individualsoftware, dezentraler und zentraler Softwaregestaltung aus.

Auch wenn Kommunikation eine zentrale Rolle bei der Softwaregestaltung spielt, unterscheidet sich die technikentwicklungsbezogene Rationalisierung von der kommunikativen Rationalisierung, wie sie von Rock/Ulrich/Witt (1990) beschrieben wurde. Letztere hat als Rationalisierungsgegenstand die sozialen Interaktionsprozesse selbst, unabhängig davon, ob es dabei um die Softwareentwicklung geht. Das ist bei der Softwaregestaltung anders, wo die Kommunikation Mittel zum Zweck der Softwaregestaltung ist. In Anlehnung an die Gegenüberstellung der drei Rationalisierungstypen (tayloristisch, systemisch, kommunikativ) (vgl. ebd.: 74) führt die unten stehende Tabelle 30 die Merkmale der technikentwicklungsbezogenen Rationalisierung auf. Diese Form der Rationalisierung stellt eine Synthese aus dem Konzept der soziotechnischen Netzwerkarbeit, der soziotechnischen Arbeitsgestaltung, dem Analyserahmen und dem Fallvergleich dar.

Ausgehend von diesem Rationalisierungstyp betrachtet man eine Branche und ihre Organisationen aus einer neuen Perspektive. Gegenstand der Rationalisierung ist die Technikgestaltung selbst und weniger die Techniknutzung. Es geht darum, die Möglichkeiten der Softwaregestaltung für einen Anwendungsbereich zu nutzen. Das bedeutet zum einen, dass den technikentwicklungsbezogenen Rationalisierungstyp eine stetige Dynamik aus individueller und Standardsoftwaregestaltung auszeichnet, die sowohl innerhalb der Organisationen als auch in der Branche zwischen Organisationen wie IT-DL, Softwarefirmen und EVU besteht. Für einen industriespezifischen Anwendungsbereich kann es sich im Zeitverlauf von Jahrzehnten mehrmals ändern, ob dort Unternehmen eine Standardlösung einsetzen oder eine individuelle Lösung. Darüber hinaus können Organisationen Software je nach Anwendungsbereich unterschiedlich nutzen. In Anwendungsbereichen mit einem hohen Anteil an Datenverarbeitung kann Software Arbeit ersetzen. Bei anderen steuert oder überwacht Software Arbeit, die nicht digitalisiert werden kann.

Anders als im Taylorismus steht deshalb nicht die Arbeitsteilung zwischen Hand- und Kopfarbeit im Zentrum der Rationalisierung. Es geht nicht darum, dass Kopfarbeitende die Arbeit der Handarbeitenden kontrollieren, so dass die Handarbeitenden nur noch möglichst wenig denken müssen. Vielmehr geht es unabhängig davon, ob die Anwendenden mehr ihre Hände oder mehr ihren Kopf benutzen, um die Trennung von Softwareanwendung, -gestaltung und -programmierung. Die einen machen die Software, welche die anderen in ihrer Arbeit anwenden, und das in vielen Fällen ihren gesamten Arbeitstag über und ausschließlich.

Dabei sind Grundlage der technikentwicklungsbezogenen Rationalisierung keine klar strukturierten Aufgaben. Es geht um Anforderungen, die Programmierende für die Software eines Anwendungsbereichs umsetzen müssen. Diese müssen erst Software-gestaltende erarbeiten und den Programmierenden übergeben, so dass dann auch die Software macht, was sie machen soll. Dabei sind die wesentlichen Mittel Wissen und Kommunikation inkl. Feedbackschleifen zwischen Anwendung und Programmierung. Dieser kommunikative Austausch und das Erarbeiten wissensbasierter Anforderungen

gelingt in organisationalen und interpersonalen Netzwerken besser als in hierarchischen Organisationen oder Marktbeziehungen. Die soziotechnische Netzwerkarbeit der Softwaregestaltung zeichnet aus, dass sie keiner Best Practice folgt und nicht detailliert vorgeplant werden kann. Sie ist vielmehr situativ und abhängig von der soziotechnischen Konstellation, zu der die Arbeitsteilung zwischen Anwendung und Entwicklung (z.B. zwischen IT-DL und EVU oder IT-Abteilung und Fachabteilungen innerhalb eines EUV) und die Softwarearchitektur gehört. Letztere legt fest, wer was gestalten kann (z.B. ob Schnittstellen vorhanden oder individuelle Einstellungen an einem Standard möglich sind) und welche Abhängigkeiten bestehen (z.B. wenn mehrere Organisationen sich auf einen Standard verständigen müssen). Das Organisationsprinzip der Situationsität zeigt sich an angepassten Rollen (z.B. Product Owner oder IT-Projektleitung) und Abläufen (z.B. Scrum), welche die zu gestaltende Software als Gegenstand haben und softwarebasierte Werkzeuge (z.B. Ticketsysteme) einsetzen.

Die Kernprobleme der soziotechnischen Interdisziplinarität und Gestaltungsmöglichkeiten löst der Arbeitsprozess der Softwaregestaltung entweder als Teil einer Matrix- oder reinen Netzwerkorganisation und zentral oder dezentral. Das hat Folgen für die Softwareanwendung: Die anwendende Organisation kann unabhängig oder abhängig sein und die eigene Organisation auf die Softwareentwicklung ausrichten und für sich entscheiden, ob Synergien durch einen Standard wichtiger sind als Wettbewerbsvorteile durch eine individuelle Software.

Die Dynamik aus Standard – individuell und dezentral – zentral zeigt sich auch bei den möglichen Strategien. Der technikentwicklungsbezogene Rationalisierungstyp zeichnet sich dadurch aus, dass es grundsätzlich zwei unterschiedliche Strategien gibt: 1. Spezialisierung auf Anwendung oder Entwicklung in getrennten Organisationen oder Abteilungen; 2. Anwendung und Entwicklung gemeinsam in einer Organisation in Form des Primats der Softwareentwicklung. Letzteres ermöglicht die Umsetzung der Strategie des Competing on Complexity, die laut Bessen (2022) erfolgreiche Unternehmen in softwareintensiven Industrien auszeichnet (Näheres siehe 4.1). Andererseits kann es für Organisationen am effizientesten sein, für bestimmte Anwendungsbereiche eine Standardsoftware einzusetzen und sich rein auf die Anwendung zu konzentrieren. Dass kann sich aber, wie oben bereits geschrieben, über die Zeit ändern.

Die Softwaregestaltung zeichnet sich dadurch aus, dass selbstständiges Lernen und Eigenmotivation wichtig sind (Subjektivität). Der Führungsstil ist nicht autoritär wie im Taylorismus und ergebnisorientiert und Führungskräfte sind vor allem aktiv, wenn Ausnahmen auftreten oder die Ergebnisse für sie nicht zufriedenstellend sind. Auch wenn es in den Fallstudien nicht vorkam, so wäre der Coaching-Ansatz für die Führung wohl der passende. Den Softwaregestaltenden geht es nicht darum, in Hierarchien aufzusteigen oder disziplinarische Verantwortung zu übernehmen. Sie wollen interessante Projekte machen und etwas dazulernen. Solche Kompetenzkarrieren schließen aber nicht aus, dass es Gehaltsstufen gibt, auf denen Softwaregestaltende aufsteigen können. Die eingesetzten Werkzeuge dienen in erster Linie dazu, den Softwaregestaltungsprozess von Anwendung bis Programmierung digital zu integrieren, die Kommunikation zu ermöglichen und Wissen zu speichern (u.a. Dokumentation von Anforderungen und deren Umsetzung).

Tabelle 30: Rationalisierungstyp der technikentwicklungsbezogenen Rationalisierung, angelehnt an Rock/Ulrich/Witt (1990: 74)

Merkmal	Ausprägung
Rationalisierungsgegenstand	Softwaregestaltung für industriespezifische Anwendungsbereiche
Arbeitsteilung	Anwendung und Entwicklung (inkl. Gestaltung)
Kernaufgaben	Anforderungen ²⁴
Organisationsziel	Kommunikation inklusive Feedbackschleifen (mündlich oder schriftlich)
Kernkontrollform	Netzwerk statt Markt oder Hierarchie
Organisationsprinzip	rollen-, ablauf- und softwarebasiert; situativ je soziotechnische Konstellation
Besondere Organisationsdeterminante	Softwarearchitektur inkl. softwaretechnischer Zuschnitt (Standard, individuell)
Kernprobleme	- softwaretechnische Interdisziplinarität - softwaretechnische Gestaltungsmöglichkeiten: organisatorische Ausrichtung (nur auf die Softwareanwendung oder sowohl auf Softwareanwendung wie auch -entwicklung), Zuschnitt (Standard/individuell)
Rationalisierungsbereiche	Arbeitsprozess der Softwaregestaltung: - zwischen Matrixorganisation und reiner Netzwerkorganisation - zwischen dezentral und zentral soziotechnische Arbeitsgestaltung der Softwareanwendung - zwischen unabhängig und abhängig
Strategien	zwischen <i>Competing on Core Competency</i> (Anwendung oder Entwicklung) und <i>Competing on Complexity</i> (Anwendung und Entwicklung)
Ausrichtung anwendende Organisation	zwischen Primat der Softwareentwicklung und Anwendung Standardsoftware
Personalentwicklung	selbstständiges Lernen in Bezug auf Software und Softwaregestaltung
Führungsstil	Management by Exception and Results; Coaching
Typische Karriere	fachliche Kompetenzkarriere ohne disziplinarische Verantwortung
Softwarebasierte Werkzeuge	prozessintegrierend von Anwendung bis Programmierung für Kommunikation und als Wissensspeicher

8.6.4. Neue Konkurrenz für das Management durch die Softwaregestaltenden?

Beim Arbeitsprozess der Softwaregestaltung fungieren Führungskräfte mehr als Rahmengeber und sind nur teilweise direkt involviert. Vielmehr sind Softwaregestaltende und Programmierende Träger der Arbeitsgestaltung. Welchen Anteil hat das Management noch an der Arbeitsgestaltung? Inwiefern gibt es Konflikte innerhalb des Managements? Können die Softwaregestaltenden als Teil des Managements betrachtet werden?

24 Andere Begriffe dafür sind Konzepte, Spezifikationen, Stories, Tickets; zu Anforderungen gehört auch mündliches oder schriftliches Feedback wie z.B. durch Tests.

Insgesamt ist das Management abhängig von den Softwaregestaltenden. Sie können nicht allein die optimale Nutzung der Softwaretechnik umsetzen. Bei den Arbeitskreisen von PAKET oder der Anforderungsrunde von INTERN2 sitzen ausgewählte Expert:innen. Bei INTERN1 ist das Management einer neben vielen anderen Anforderungsstellenden. Bei KOOP1 und KOOP2 übernehmen IT-DL oder intern in den EVU IT-Projekt-leitende, Applikationsbetreuende oder Prozessverantwortliche die Softwaregestaltung. Nur in wenigen Fällen machen Teamleitung oder Gruppenleitung bei der Softwarege-staltung operativ mit: EVU von PAKET, Teamleitung von KOOP2 oder die Geschäftsfüh-
rung von STARTUP. Dabei gestalten sie dann aber nur Teile einer Software mit.

Vielmehr setzt das Management die Rahmenbedingungen. Die Kontrolle über finan-zielle und personelle Ressourcen und strategische Entscheidungen liegt beim Manage-
ment. Es entscheidet über die betriebsinternen Karrieren – egal ob von Software an-wendendem, gestaltendem oder programmierendem Personal. Wenn die Softwareent-wicklung komplett ausgelagert ist, dann richtet sich der Fokus des Managements der EVU darauf, die IT-DL, IT-Beratenden oder Softwarefirmen zu kontrollieren. Bei IN-
TERN1 und INTER2 genehmigt das Management die Softwaregestaltung. Bei KOOP1 hat das Management über die Auslagerung der IT entschieden und das Anforderungsma-nagement eingeführt. In einem EVU kontrolliert der Digitalisierungsmanager Anfor-
derungen, bevor sie an das IT-DL weitergeleitet werden. Beim strategischen Anfor-
derungsmanagement von KOOP1 machen die leitenden Personen der IT-Abteilungen aus den EVU mit. Bei KOOP2 hat das Management eines EVU entschieden, wieder Teile der Softwaregestaltung vom IT-DL zurückzuverlagern. Bei KOOP3 hat das Management des IT-DL beschlossen, einen IoT-Bereich mit entsprechendem Personal zu gründen. Nur beim STARTUP gibt es keine formalen Hierarchien. Doch basiert der Primat der Soft-
wareentwicklung darauf, dass die Gründer über Wissen zur Softwareentwicklung ver-
fügen und die softwaretechnische Interdisziplinarität verkörpern.

Dann unterscheidet es sich von Fall zu Fall, ob das Management und bestimmte Führ-
ungsebenen ein Hindernis für die Softwaregestaltung sind oder diese selbst überneh-
men. In Matrixorganisationen sind sie oftmals wichtige Sponsoren z.B. für IT-Projek-te, um sie abteilungsübergreifend durchzusetzen. Bei INTERN1 und INTER2 erleben Befragte Führungskräfte auf Ebene der Fachbereichsleitung als hinderlich für die Soft-
waregestaltung und es gibt Kompetenzkonflikt[e(?)] zwischen Abteilungen (IT, Fachbe-reiche). Bei KOOP2 werden Hierarchien in Frage gestellt. In einem EVU hat das Manage-
ment die Hoheit über die Gestaltung vom IT-DL zurückgeholt (u.a. dadurch, dass das EVU wieder das Projektmanagement selbst macht). Bei PAKET gibt es ein EVU, in dem die Führungskraft eines Fachbereiches über die Wahl eines Softwareunternehmens ent-scheidet, ohne die IT-Abteilung zu fragen. Im STARTUP herrscht ein anderes Führungs-verständnis vor. Die Gründer sind höchstens informell führend aufgrund ihrer Kom-petenzen. Darüber, wer welche Rollen übernimmt, entscheiden die Mitarbeitenden ge-meinsam.

In den Fallstudien treiben die Softwaregestaltenden die Technikgestaltung in den Organisationen voran. Sie helfen damit, ein Kernproblem des Kapitals zu lösen, wodurch sie eine bestimmte Position für sich beanspruchen können (vgl. Armstrong 1985). So be-stimmt sich, wie schon bei Hohlmann beschrieben, ihre Position technikinduziert: Sie sind notwendig, wie z.B. die Rolle von Key User:innen, damit die Organisation die Tech-

nologie einsetzen kann (vgl. Hohlmann 2007 zum Einsatz der SAP ERP-Software). Einerseits sind Softwaregestaltende, indem sie Arbeit gestalten, Teil des Managements. Sie helfen, die Arbeit der Anwendenden zu kontrollieren. Andererseits: Wenn es um das Beschäftigungssystem geht, ist bei den hier untersuchten Fällen noch nicht zu erkennen, dass es zu einer direkten Konkurrenz zum existierenden Management kommt. Vielmehr entsteht eine eigenständige Gruppe, die bestehende Karrierewege in Frage stellt und teilweise eigene bekommt. Ob in Zukunft in den EVU vor allem Softwaregestaltende oder -programmierende aufsteigen: Dafür gibt es keine Indizien. In den Fallstudien sehen sie sich nicht als Teil des Managements. Programmierende und Gestaltende wollen größtenteils keine disziplinarische Verantwortung haben und Teil des Managements werden. Laut Pohlmann zeichnet das Management neben Funktionen wie Gestaltung, Kontrolle und Koordination auch eine hierarchische Position aus, mit der Führung und Weisungsbefugnisse einhergehen (vgl. Pohlmann 2017: 207f.). Das ist bei den meisten hier befragten Softwaregestaltenden nicht der Fall und das Management hat die Arbeitsgestaltung durch Softwaregestaltung delegiert oder ausgelagert. Wie STARTUP und die diskutierte softwaretechnische Prozessorganisation (siehe 8.5.3.3) zeigen, werden Teile des Managements überflüssig, wenn eine Organisation sich auf die Softwaregestaltung ausrichtet.

8.6.5. Facetten einer industriespezifischen Softwaregestaltung

Die Organisationen in den Fallstudien gestalten eine industriespezifische Software. Es zeigen sich mehrere Eigenheiten der Softwaregestaltung in der Energiewirtschaft: die industriespezifische Detailtiefe und Komplexität, die große Bedeutung der Regulierung, die Arbeitsteilung zwischen großen und kleinen EVU, die lokale Verankerung der EVU, der zögerliche Wandel und die noch offene Frage, welche Rolle Softwaregestaltung in Zukunft für die Wettbewerbsfähigkeit von Firmen in der Branche spielen wird.

Es gibt vielfältige Wissensdomänen und ein komplexes technisches und regulatorisches Umfeld, die eine Herausforderung für die Softwaregestaltung in der Energiewirtschaft darstellen. Im 7. Kapitel wurden einige der vielen Gesetze und Verordnungen genannt. Dazu kommen technisch komplexe Anlagen – ob Kraftwerke oder Netze mit ihren Zählern, Trafostationen und verschiedenen physikalischen Eigenheiten (Gas, Strom). Dazu kommt, dass trotz Entflechtung viele Stadtwerke noch Konzerne mit unterschiedlichen Geschäftsfeldern inklusive Schwimmbädern und öffentlichem Nahverkehr sind. Typisch für die Industrie ist, dass, obwohl in anderen Industrien ähnliche Anwendungsbereiche (z.B. Instandhaltung) existieren, diese trotzdem individuell sind. INTERN1 kann durch die interne Softwaregestaltung eine industriespezifische Instandhaltungssoftware entwickeln, z.B. mit Notruffunktion für Monteur:innen. Die individuelle Softwaregestaltung ist darauf ausgelegt (wie ein Befragter meint), Netzkonzession zu bekommen. Das heißt, die Softwaregestaltung richtet sich an institutionelle Begebenheiten, die darin bestehen, dass die netzbewirtschaftende Organisation (meist die Kommunen) den Netzbetrieb vergibt und spezifische Erwartungen hat.

Zudem ist die große Bedeutung der Regulierung für die Softwaregestaltung und die damit einhergehenden stetigen Softwareanpassungen etwas Industriespezifisches. Ein großer Teil der Softwaregestaltung wäre ohne den stetigen Wandel der Regulierung nicht

notwendig. Das STARTUP existiert nur aufgrund einer Regulierung (Emissionshandel) und ihre Dienstleistung ist es, diese digital umzusetzen. Für die IT-DL, Softwarefirmen und allgemein die Softwaregestaltenden ist die sich stetig ändernde Regulierung und die neuen Anforderungen durch die Energiewende Geschäftsgrundlage und Existenzberechtigung. Zudem stellt die Regulierung eine Grundlage für Synergien und Kooperationen bei der Softwaregestaltung dar: Sie ist für alle gleich, und so stellt sie einen Ausgangspunkt dar, um zu kooperieren und zumindest für die regulierten Bereiche eine industriespezifische Standardsoftware zu entwickeln. Das zeigt sich in der Softwaregestaltung vor allem bei KOOP1 und PAKET, wo ein IT-DL bzw. eine Softwarefirma für mehrere EVU einen Standard ausprägt. Die Unterscheidung zwischen regulierten und wettbewerblichen Bereichen wird in der Softwaregestaltung zur Frage, was EVU kooperativ und was sie individuell programmieren.

Das Besondere an der Energiewirtschaft in Deutschland sind die vielen Stadtwerke. Sie stellen einen großen Markt für Software dar. Vor allem die kleineren EVU zeigen sich in den Fallstudien als Trittbrettfahrer, wenn es um Softwaregestaltung geht. In den Fallstudien sind es eher die mittleren und großen EVU, die sich an der Entwicklung eines Standards beteiligen. Dies geschieht bei einem IT-DL wie bei KOOP1 oder KOOP2 oder indem EVU mit Softwarefirmen Projekte machen oder extra Personal haben, um Anforderungen zu stellen und zu testen wie bei PAKET. So profitieren die kleineren EVU davon, was an Software entwickelt wird. Ein EVU aus PAKET oder die IT-DL von KOOP1 und KOOP2 stellen die entwickelte Software anderen (meist kleineren) EVU zur Verfügung. Das kann so weit gehen, dass auf dem gleichen Server für mehrere EVU die Anwendungssoftware läuft und sie ein EVU oder ein IT-DL betreibt und gestaltet. Bei KOOP1 und PAKET sprechen Befragte explizit davon, dass IT-DL oder Softwareunternehmen Stadtwerke »spielen« können und dies auch tun. Es wird in den Fällen auch angeboten (wie auch in KOOP2), einzelne Prozesse via BPO für EVU zu übernehmen, d.h. nicht nur die Software zur Verfügung zu stellen, sondern auch die Anwendenden.

Speziell an der Branche ist zudem die Kultur, zu der Kooperationen, ein regionaler Bezug und zögerlicher Wandel innerhalb der EVU gehören. Einerseits erleichtert die Kultur die Kooperation wie bei KOOP1. Die kooperative Kultur stammt wohl aus einer Zeit, in der kein Wettbewerb herrschte und die Stadtwerke die gleichen Herausforderungen hatten, aber keinen gemeinsamen Markt. Immer noch gibt es eine getrennte, regionale Verantwortung und begrenzten Wettbewerb. Gleichzeitig zeigt KOOP2, dass Kooperationen scheitern können. Die Branche zeichnet sich zudem durch einen regionalen Bezug aus, der sich in spezifischen Netzen (INTERN1), regionaler Kundschaft (KOOP1, KOOP2) und regionalen Arbeitsmärkten niederschlägt (einzelne Befragte bei KOOP2 und PAKET). Für die Softwaregestaltung bedeutet das zum einen, die Anforderungen vor Ort aufzunehmen, und zum anderen den Vorteil, Fachkräfte aus mittelgroßen oder kleineren Städten gewinnen zu können, weil die großen Konzerne und IT-Unternehmen dort meist keine Niederlassungen haben. In der Branche zeigt sich ein zögerlicher Wandel wie bei INTERN1 und INTERN2 oder den EVU aus KOOP1 und KOOP2: ob bei der Durchsetzung gemeinsamer Werkzeuge für die Softwaregestaltung, der teamübergreifenden Zusammenarbeit oder der Einführung neuer Methoden wie Scrum. Die EVU versuchen, die Wettbewerbsfähigkeit ihrer althergebrachten Strukturen zu erhalten, und transformieren sie nur inkrementell. Bei KOOP3

funktioniert das IT-DL als Mittelsmann in die Branche für das IoT-Softwareunternehmen. Es kann als übersetzende Organisation zwischen der Start-up-Welt und der EVU-Welt gesehen werden. Der Fall zeigt, dass innovativere Technologien wie IoT ausgelagerte oder branchenfremde Organisationen gestalten und die EVU, wenn überhaupt, dann eine fertige Standardlösung erweitern (wie EVU1 von KOOP3). EVU2 von KOOP2 hat eine Person eingestellt, welche die Projektkultur fördern soll. An den althergebrachten energiewirtschaftlichen Strukturen hält das EVU aber fest: Die funktionale Aufteilung nach fachlichen Abteilungen und Teams bleibt bestehen (Abrechnung, Netzanschluss, Energiehandel, IT-Abteilung etc.) und für die Softwaregestaltung ergänzt das EVU die Organisation um Positionen für IT-Projektmanagement, IT-Koordination und Key User:innen.

Nimmt man die Matrix der vier Typen der soziotechnischen Arbeitsgestaltung (8.6.1.3), so ging es in keiner der Fallstudien um reine Auftragsarbeiten oder um softwarebasierte EVU, deren proprietäre Software zu einem Standard in der Branche wird und die sich disruptiv mithilfe von Softwareentwicklung durchsetzen können. START-UP entwickelt zwar eine proprietäre Software, die sie anderen anbietet, jedoch für eine sehr spezielle Funktionalität (Emissionshandel). In sämtlichen anderen Fällen geht es um Standardsoftware, die bis auf PAKET und einige EVU bei KOOP1, KOOP2, KOOP3 die EVU weiterentwickeln. Es ist und bleibt eine offene Frage, welcher der vier Typen der Arbeitsgestaltung aktuell und in Zukunft quantitativ dominiert. Von den untersuchten Fällen steht kein EVU kurz davor, in naher Zukunft aus dem Markt gedrängt zu werden. Somit bestätigen die Fallstudien, was unter 7. geschrieben wurde: Das Überleben in der Branche sichern unterschiedliche Formen der Softwaregestaltung. Wobei es variiert, wie umfangreich der verwendete Standard ist und welche individuellen Anpassungen die EVU vornehmen.