

Die Generierung von Information wird für Unternehmen immer wichtiger. Das zeigt sich beispielsweise an ökonomischen Faktoren: Unternehmen, die mit Daten handeln, sind seit einigen Jahren wertvoller als klassische Industrieunternehmen. Aber auch unternehmensintern haben Informationen einen steigenden Wert. Bei einem sinnvollen Informationseinsatz können Abläufe im Unternehmen optimiert werden, was den Produktionsoutput erhöhen kann. Produzierende Unternehmen indes haben mit der Datennutzung noch immer Schwierigkeiten. Das liegt unter anderem in der hohen, schwer beherrschbaren Komplexität der Informationssystem-Architektur begründet, die zu Datenredundanz und schlechter Informationsqualität führt.

Um dieser Herausforderung entgegenzuwirken, wurde in Aachen das *Internet of Production* entwickelt. Dieses enthält einen Vorschlag für eine Infrastruktur, mit welcher Unternehmen den benötigten ubiquitären und echtzeitfähigen Datenzugriff realisieren können. Durch die Verknüpfung der vorhandenen Informationssysteme über eine *Middleware+* und die Datenverarbeitung in der *Smart-Data-Ebene* können Unternehmen darüber hinaus innovative Algorithmen zur Generierung von Informationen nutzen. Zur Umsetzung des *Internet of Production* wird eine software-definierte Plattform benötigt, welche die Funktionalitäten der beiden vorgenannten Ebenen abdeckt.

Um eine solche software-definierte Plattform sinnvoll in die bestehende Informationssystem-Architektur zu integrieren, müssen Unternehmen diese anpassen. Ziel der Arbeit ist es, Unternehmen den Weg dahin aufzuzeigen. Dies wird im Wesentlichen durch drei Teilzielsetzungen erreicht: Durch die Systematisierung von software-definierten Plattformen erhalten Unternehmen einen Überblick von vorhandenen Funktionalitäten und vorhandener Software. Zudem können Integrationsanforderungen solcher Plattformen abgeleitet werden. Das zweite Ziel betrifft die Systematisierung der in Unternehmen bestehenden Informationssystem-Architektur und die Entwicklung von Visualisierungsformen, welche die Handlungsfelder für Integration aufzeigen. Durch das Aufzeigen einer Ziel-Architektur werden als drittes die notwendigen Entscheidungen für Unternehmen dargelegt.

Entscheider in Unternehmen erhalten ein konkretes Vorgehen und eine Handlungsempfehlung zur Weiterentwicklung ihrer Informationssystem-Architektur aus den Blickwinkeln Technik, Organisation, Prozesse und Daten. Damit können sie den ersten Schritt zur Umsetzung des *Internet of Production* gehen.



Jörg Hoffmann

Informationssystem- Architekturen produzierender Unternehmen bei software- definierten Plattformen



Herausgeber:
Univ.-Prof. Dr.-Ing. Dipl.-Wirt. Ing. Günther Schuh

Informationssystem-Architekturen produzierender Unternehmen bei software-definierten Plattformen

Information System Architectures of Manufacturing Companies for Software Defined Platforms

Von der Fakultät für Maschinenwesen
der Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
genehmigte Dissertation

vorgelegt von

Jörg Hoffmann

Berichter:

Univ.-Prof. Dr.-Ing. Dipl.-Wirt. Ing. Günther Schuh
Außerplanmäßiger Professor Dr.-Ing. Volker Stich

Tag der mündlichen Prüfung: 31. Oktober 2018

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

SCHRIFTENREIHE RATIONALISIERUNG

Jörg Hoffmann

Informationssystem-Architekturen produzierender
Unternehmen bei software-definierten
Plattformen

Herausgeber:

Prof. Dr.-Ing. Dipl.-Wirt. Ing. G. Schuh

Band 158



Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <https://portal.dnb.de> abrufbar.

Jörg Hoffmann:

Informationssystem-Architekturen produzierender Unternehmen bei
software-definierten Plattformen

1. Auflage, 2018

Gedruckt auf holz- und säurefreiem Papier, 100% chlorfrei gebleicht.

Apprimus Verlag, Aachen, 2018
Wissenschaftsverlag des Instituts für Industriekommunikation und Fachmedien
an der RWTH Aachen
Steinbachstr. 25, 52074 Aachen
Internet: www.apprimus-verlag.de, E-Mail: info@apprimus-verlag.de

Printed in Germany

ISBN 978-3-86359-675-0

D 82 (Diss. RWTH Aachen University, 2018)

Per aspera ad astra.

(KLEIST 1822, S. 46)

Danksagung

Durch das Mühsame zu den Sternen – das beschreibt ganz gut den Weg, den ein jeder Promovend auf sich nehmen muss, wenn er eine Dissertation abfassen möchte. Wie auf jedem Weg hilft es auch in diesem Fall, wenn man ihn nicht alleine gehen muss. Daher möchte ich zuerst meinen FIR-Kollegen aus der #stardiss-Gruppe danken, die mit mir über anderthalb Jahre fast jedes Wochenende gleichzeitig an ihren Arbeiten geschrieben haben. Die gemeinsame Motivation und der gemeinsame Kaffee haben einen großen Anteil am Gelingen dieser Arbeit. Insbesondere gilt der Dank hierbei Martin Bleider, Dr. Steffen Nienke, Anne Bernardy und Felix Jordan.

Auf dem Weg haben mich auch meine Kollegen und Mitarbeiter Norman Schaffer, Marie Gruber, Eliane Hartard, Erik Pflugradt, Martin Bremer, Pit Heimes und Christian Hocken intensiv begleitet und standen jederzeit mit Rat und Tat zur Seite. Meinen Vorgesetzten, allen voran Dr. Violett Zeller und Dr. Matthias Deindl, danke ich für ihr Vertrauen, den Support und die Ideen, mit der sie zu dieser Arbeit beigetragen haben, genauso wie Ellen, Manuel und dem Rest meiner Familie für die wertvollen sprachlichen Hinweise.

Den Weg ermöglicht haben mir mein Doktorvater und Direktor des FIR, Prof. Dr. Günther Schuh und unser Geschäftsführer Prof. Dr. Volker Stich, die uns ein inspirierendes, lebendiges und industrienahes Umfeld geschaffen haben, wie man in Deutschland auf diesem Gebiet wohl kein zweites findet. Viele der grundlegenden Ideen, auf denen diese Arbeit basiert, konnten nur in diesem Umfeld entstehen.

Nicht zuletzt danke ich meiner Frau Elenya, die an vielen Abenden und Wochenenden auf mich verzichten musste und mir trotzdem jederzeit auf dem Weg zur Seite stand, mich motiviert und gestützt hat.

Zuletzt jedoch gilt mein Dank dem, der Welt, Zeit und meinen Weg in seiner Hand hält. *Soli deo gloria.*

Aachen, im November 2018

Jörg Hoffmann

Vorbemerkungen

Alle verwendeten Markennamen wurden aus Gründen der einfachen Lesbarkeit ohne Copyright- oder Trademark-Zeichen dargestellt. Die Rechte liegen bei den jeweiligen Eigentümern. Der Autor steht in keiner Beziehung zu den aufgezählten Unternehmen und deren Produkten, sie werden jeweils nur als Beispiele bekannter Marken dargestellt.

Ebenfalls wurde aus Gründen der Lesbarkeit auf eine gendergerechte Schreibweise von Personengruppen verzichtet. Dies darf in keinster Weise diskriminierend verstanden werden, vielmehr sind mit Begriffen wie *Mitarbeiter*, *Kunde* oder *Vorgesetzter* sowohl die männliche als auch die weibliche Form gemeint.

Informationssystem-Architekturen produzierender Unternehmen bei software-definierten Plattformen

Die Nutzung von Daten und die Generierung von Information werden für Unternehmen immer wichtiger. Das zeigt sich beispielsweise an ökonomischen Faktoren: Unternehmen, die mit Daten handeln, sind seit einigen Jahren wertvoller als klassische Industrieunternehmen. Aber auch unternehmensintern haben Informationen einen steigenden Wert. Bei einem sinnvollen Informationseinsatz können Abläufe im Unternehmen optimiert und teilweise auch automatisiert werden, was beispielsweise den Produktionsoutput erhöhen kann. Produzierende Unternehmen in Deutschland indes haben mit der Datennutzung noch immer Schwierigkeiten. Das liegt unter anderem in der hohen, schwer beherrschbaren Komplexität der Informationssystem-Architektur begründet, die zu Datenredundanz und schlechter Informationsqualität führt.

Um dieser Herausforderung entgegenzuwirken, wurde in Aachen das *Internet of Production* entwickelt. Dieses enthält einen Vorschlag für eine Infrastruktur, mit welcher Unternehmen den benötigten ubiquitären und echtzeitfähigen Datenzugriff realisieren können. Durch die Verknüpfung der vorhandenen Informationssysteme über eine *Middleware+* und die Datenanalyse und -verarbeitung in der *Smart-Data-Ebene* können Unternehmen darüber hinaus innovative Algorithmen zur Generierung von Informationen nutzen. Zur Umsetzung des *Internet of Production* wird eine software-definierte Plattform benötigt, welche die Funktionalitäten der beiden vorgenannten Ebenen abdeckt.

Um eine solche software-definierte Plattform sinnvoll in die bestehende Informationssystem-Architektur zu integrieren, müssen Unternehmen diese anpassen. Ziel der vorliegenden Arbeit ist es, Unternehmen den Weg dahin aufzuzeigen. Dies wird im Wesentlichen durch drei Teilzielsetzungen erreicht: Durch die Systematisierung von software-definierten Plattformen erhalten Unternehmen einen Überblick von vorhandenen Funktionalitäten und vorhandener Software. Zudem können Integrationsanforderungen solcher Plattformen abgeleitet werden. Das zweite Ziel betrifft die Systematisierung der in Unternehmen bestehenden Informationssystem-Architektur und die Entwicklung von Visualisierungsformen, welche die Handlungsfelder für Integration aufzeigen. Durch das Aufzeigen einer Ziel-Architektur werden als drittes die notwendigen Entscheidungen für Unternehmen dargelegt.

Entscheider in Unternehmen erhalten mit der vorliegenden Arbeit ein konkretes Vorgehen und eine Handlungsempfehlung zur Weiterentwicklung ihrer Informationssystem-Architektur aus den Blickwinkeln Technik, Organisation, Prozesse und Daten. Damit können sie den ersten Schritt zur Umsetzung des *Internet of Production* gehen.

Information System Architectures of Manufacturing Companies for Software Defined Platforms

The use of data and the generation of information is becoming increasingly important for companies. This can be seen, for example, in economic factors: companies with a business model based on the trade with data are more valuable than classic industrial companies. But information also has an increasing importance within companies. With a meaningful use of information, processes in a company can be optimized and partially automated, which, for example, can increase the production output. Manufacturing companies in Germany, however, still have difficulties with the use of data. One of the reasons for this is the high complexity of the information system architecture, which is difficult to control and leads to data redundancy and poor information quality.

To counteract this challenge, the *Internet of Production* was developed in Aachen. This contains a proposal for an infrastructure with which companies can obtain the required ubiquitous and real-time capable data access. By linking existing information systems via a *Middleware+* and data analysis and processing in the *smart data* layer, companies can also use innovative algorithms to generate information. To implement the *Internet of Production*, a software defined platform is required that covers the functionalities of the two levels mentioned before.

In order to integrate such a software defined platform into an existing information system architecture, companies must adapt the architecture. The aim of the present work is to show companies the way to this goal. This is essentially achieved through three partial objectives: By systematizing software defined platforms, companies obtain an overview of existing functionalities and existing software. In addition, integration requirements of such platforms can be derived. The second goal concerns the systematization of the existing information system architecture in companies and the development of visualization forms, which point out the fields of action for integration. By developing a target architecture, thirdly the necessary decisions for companies are presented.

Decision-makers in companies receive a concrete procedure and recommendations for further developing their information system architecture from the perspectives of technology, organization, processes and data. With this, they can take the first step towards implementing the *Internet of Production*.

Inhaltsübersicht

1	Einleitung	1
2	Grundlagen und Abgrenzung der Untersuchung	13
3	Stand der Erkenntnisse.....	45
4	Herleitung des Konzeptansatzes.....	73
5	Beschreibung software-definierter Plattformen.....	89
6	Beschreibung von Informationssystem-Architekturen	129
7	Ist- und Ziel-Zustand der Informationssystem-Architektur.....	189
8	Gestaltung des Integrationsprozesses software-definierter Plattformen...	213
9	Evaluation	225
10	Zusammenfassung und Ausblick.....	261
11	Literaturverzeichnis.....	265
	Anhang	301

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ausgangssituation und Problemstellung	1
1.2	Zielsetzung und Forschungsfrage	3
1.3	Wissenschaftlicher Bezugsrahmen und Forschungskonzeption	6
1.4	Aufbau der Arbeit	10
2	Grundlagen und Abgrenzung der Untersuchung	13
2.1	Informationssysteme und -architekturen	13
2.1.1	Informationen im Kontext von Daten und Wissen	13
2.1.2	Informationssysteme	16
2.1.3	Schichtenmodelle	19
2.1.4	Integration und Schnittstellen	22
2.1.5	Informationssystem-Architekturen	27
2.1.6	Informationssysteme in produzierenden Unternehmen	28
2.2	Management von Informationssystem-Architekturen	33
2.2.1	Informationsmanagement	33
2.2.2	IT-Komplexität	35
2.2.3	IT-Strategie	36
2.2.4	Rolle der IT-Abteilung	37
2.3	Die Vision vom Plattformeinsatz im Kontext der vernetzten Digitalisierung	39
2.4	Abgrenzung und Präzisierung des Untersuchungsbereichs	42
3	Stand der Erkenntnisse	45
3.1	Beschreibung von software-definierten Plattformen	45
3.2	Plattform-Integrationskonzepte	50
3.3	Beschreibung von Informationssystem-Architekturen	53
3.4	Konzepte von Informationssystem-Architekturen produzierender Unternehmen	58
3.5	Methoden zum IT-Komplexitätsmanagement	63
3.6	Zwischenfazit und Ableitung des Forschungsbedarfs	70
4	Herleitung des Konzeptansatzes	73
4.1	Anforderungen an die zu entwickelnde Methode	73
4.2	Methodische Grundlagen	75
4.2.1	Systemtheorie	75
4.2.2	Modellbildung	79
4.2.3	Morphologie und Typisierung	81
4.3	Konkretisierung der Vorgehensweise	84
5	Beschreibung software-definierter Plattformen	89
5.1	Definition des Begriffs <i>Software-definierte Plattform</i>	90
5.1.1	Herleitung der Definition	90

5.1.2	Erweiterte eigene Definition und Einordnung in die Infrastruktur des <i>Internet of Production</i>	96
5.2	Herleitung der Merkmalsdimensionen	97
5.3	Merkmale zur Strukturierung software-definierter Plattformen	98
5.3.1	Schnittstellen	98
5.3.2	Datenmanagement	99
5.3.3	Datenbankkonzept	101
5.3.4	Stufen der Datenverarbeitung	103
5.3.5	IT-Sicherheitsfunktionen	105
5.3.6	Informationsbereitstellung	106
5.3.7	Entwicklungsumgebung	108
5.4	Zusammenfassung der beschreibenden Merkmale	110
5.5	Anwendung der Morphologie zur Bildung von Typen software-definierter Plattformen	111
5.5.1	Plattformtyp I: vollumfängliches System	114
5.5.2	Plattformtyp II: eingeschränkter Funktionsumfang	116
5.5.3	Plattformtyp III: reine Vernetzung	119
5.6	Integrationsanforderungen software-definierter Plattformen	121
5.6.1	Herleitung der Integrationsanforderungen	122
5.6.2	Technische Integrationsanforderungen	124
5.6.3	Organisatorische Integrationsanforderungen	125
5.6.4	Prozessuale Integrationsanforderungen	126
5.6.5	Anforderungen der Datenintegration	127
5.7	Zusammenfassung Kapitel 5	128
6	Beschreibung von Informationssystem-Architekturen	129
6.1	Herleitung der Merkmalsdimensionen von Informationssystemen	130
6.2	Technische Merkmale	132
6.2.1	Schnittstellen	132
6.2.2	Outsourcinggrad	134
6.2.3	Echtzeitfähigkeit	137
6.2.4	Status des Softwarelebenszyklus	138
6.3	Organisatorische Merkmale	140
6.3.1	Systemursprung	140
6.3.2	Systemverantwortung	142
6.3.3	Organisationseinheit	144
6.3.4	Nutzerkreis	145
6.4	Prozessuale Merkmale	147
6.4.1	Systemkonzept	147
6.4.2	Kernprozess	151
6.5	Daten betreffende Merkmale	154
6.5.1	Betroffene Ebenen	154
6.5.2	Informationsqualität	157
6.5.3	Datenstandardisierung	160

6.6	Zusammenfassung der beschreibenden Merkmale	161
6.7	Darstellung von Informationssystem-Architekturen	163
6.7.1	Darstellungen mit Fokus auf technische Merkmale	165
6.7.2	Darstellungen mit Fokus auf organisatorische Merkmale.....	167
6.7.3	Darstellungen mit Fokus auf prozessuale Merkmale.....	168
6.7.4	Darstellungen mit Fokus auf Daten betreffende Merkmale	172
6.8	Typen von Informationssystem-Architekturen produzierender Unternehmen	174
6.8.1	Architekturtyp I: Dezentrales Architekturparadigma und IT- Organisation als Technologiepartner	177
6.8.2	Architekturtyp II: Zentrales Architekturparadigma und IT- Organisation als Technologiepartner	179
6.8.3	Architekturtyp III: Dezentrales Architekturparadigma und IT- Organisation als Prozess- oder Geschäftspartner.....	181
6.8.4	Architekturtyp IV: Zentrales Architekturparadigma und IT- Organisation als Prozess- oder Geschäftspartner.....	184
6.8.5	Architekturtyp V: IT-Organisation als Erfolgspartner	186
6.9	Zusammenfassung Kapitel 6.....	188
7	Ist- und Ziel-Zustand der Informationssystem-Architektur.....	189
7.1	Zielbeschreibung der Informationssystem-Architektur	190
7.1.1	Zielinformationssystem-Architektur in der Dimension <i>Technik</i>	192
7.1.2	Zielinformationssystem-Architektur in der Dimension <i>Organisation</i>	195
7.1.3	Zielinformationssystem-Architektur in der Dimension <i>Prozesse</i> ...	199
7.1.4	Zielinformationssystem-Architektur in der Dimension <i>Daten</i>	202
7.2	Zusammenhänge zwischen Ist-Architekturtypen und Zielbeschreibung....	205
7.3	Abgeleitete Entscheidungskriterien	209
7.4	Zusammenfassung Kapitel 7.....	212
8	Gestaltung des Integrationsprozesses software-definierter Plattformen...213	
8.1	Anwendung der Modelle im Problemlösungszyklus	214
8.1.1	Zielformulierung mit dem Beschreibungsmodell für software- definierte Plattformen	214
8.1.2	Situationsanalyse mit dem Beschreibungsmodell für Informationssystem-Architekturen.....	217
8.1.3	Synthese und Lösungsanalyse mit dem Erklärungsmodell	219
8.2	Hinweise zum Auswahl- und Einführungsprozess für software-definierte Plattformen.....	220
8.2.1	Definition einer Vorgehensweise	220
8.2.2	Auswahl einer software-definierten Plattform	221
8.3	Zusammenfassung Kapitel 8.....	222
9	Evaluation	225

9.1	Auswahl der Fallstudien	225
9.2	Fallstudie 1: e.GO Mobile AG.....	227
9.2.1	Das Unternehmen e.GO Mobile AG	227
9.2.2	Ausgangssituation bei e.GO.....	228
9.2.3	Ziele und Anforderungen software-definierter Plattformen bei e.GO	230
9.2.4	Die Informationssystem-Architektur von e.GO	241
9.2.5	Empfehlungen zur Entwicklung der Informationssystem-Architektur bei e.GO.....	246
9.3	Fallstudie 2: PTC.....	247
9.3.1	Das Unternehmen PTC	247
9.3.2	Der Softwareauswahlprozess software-definierter Plattformen aus Sicht eines Softwareherstellers	248
9.3.3	Anwendungsfall Kunde A	250
9.3.4	Anwendungsfall Kunde B	252
9.3.5	Handlungsempfehlungen für PTC	256
9.4	Zusammenfassende Bewertung der Evaluationsergebnisse.....	257
10	Zusammenfassung und Ausblick.....	261
10.1	Zusammenfassung.....	261
10.2	Ausblick.....	263
11	Literaturverzeichnis.....	265
Anhang		301

Abbildungsverzeichnis

Abbildung 1-1: Status quo der vernetzten Digitalisierung	1
Abbildung 1-2: Einführung einer software-definierten Plattform in der bestehenden Informationssystem-Architektur	4
Abbildung 1-3: Wissenschaftssystematik	6
Abbildung 1-4: Forschungskonzeption	8
Abbildung 1-5: Phasen der angewandten Forschung nach ULRICH	10
Abbildung 2-1: Ebenen der Semiotik	14
Abbildung 2-2: Begriffshierarchie von Zeichen, Daten, Informationen und Wissen ..	15
Abbildung 2-3: Smart-Data-Ansatz zur Generierung von Information.....	16
Abbildung 2-4: Zusammenhang von Systemen, Unter- und Umsystemen	17
Abbildung 2-5: Einordnung von Informationssystemen	18
Abbildung 2-6: Informationssysteme als soziotechnische Systeme.....	19
Abbildung 2-7: Schichteneinteilung bei ISO/OSI und TCP/IP	20
Abbildung 2-8: Referenzarchitekturmodell Industrie 4.0	21
Abbildung 2-9: Modulare Architektur des IIC	22
Abbildung 2-10: Dimensionen des Integrationsmanagements nach DERN	23
Abbildung 2-11: Direkte und indirekte Schnittstellen	24
Abbildung 2-12: Standardisierungsgrad von Schnittstellen	24
Abbildung 2-13: Teilarchitekturen von Informationssystem-Architekturen	28
Abbildung 2-14: Typische Informationssystem-Architektur eines produzierenden Unternehmens.....	30
Abbildung 2-15: Auflösung der Automatisierungspyramide	32
Abbildung 2-16: Informationsmanagement zwischen Ermöglichen der Unternehmensstrategie und Angleichen an die Unternehmensstrategie	33
Abbildung 2-17: IT-Komplexität und Geschäftskomplexität	35
Abbildung 2-18: Vorgehen zur Gestaltung einer IT-Strategie	36
Abbildung 2-19: Entwicklung der Rolle der IT-Abteilung.....	38
Abbildung 2-20: Idealtypen der transaktionszentrierten und der datenzentrierten Plattform	39
Abbildung 2-21: Schichtenmodell Digitale Infrastrukturen	40
Abbildung 2-22: Infrastruktur des <i>Internet of Production</i>	41

Abbildung 3-1: Leistungsbündel, Architekturmuster und generische Enabler innerhalb einer software-definierten Plattform	46
Abbildung 3-2: Architektur der <i>Product Cloud</i> innerhalb des <i>New Technology Stack</i>	47
Abbildung 3-3: Technologieelemente digitaler Plattformen	48
Abbildung 3-4: Funktionen einer IoT-Plattform	48
Abbildung 3-5: IoT-Referenzarchitektur	49
Abbildung 3-6: Referenzmodell für IoT-Plattformen	49
Abbildung 3-7: ERP-basierte Integration	50
Abbildung 3-8: Einordnung von FREYS Plattformkonzept in die Architektur der Smart-Service-Welt	52
Abbildung 3-9: Ganzheitliches Modell der Informationssystem-Architektur	53
Abbildung 3-10: <i>Zachman Framework</i> in generischer Form	54
Abbildung 3-11: Ordnungsrahmen für das <i>Enterprise Architecture Management</i>	55
Abbildung 3-12: ARIS-Haus	56
Abbildung 3-13: Architektur zur Mehrwerterzeugung aus Daten	57
Abbildung 3-14: Best-Practice-Unternehmensarchitektur	59
Abbildung 3-15: Vorgeschlagene Informationssystem-Architektur zur Interaktion und Kollaboration mit Anwendern des ‚ <i>Industrie 4.0 Maturity Index</i> ‘	60
Abbildung 3-16: Aufgabensicht als Grundstruktur des Aachener PPS-Modells	61
Abbildung 3-17: PLM-Funktionsmodell des WZL	62
Abbildung 3-18: Kategorien von Literaturquellen zu IT-Komplexität	63
Abbildung 3-19: Complexity-Cross	66
Abbildung 3-20: SIP-Prozess	69
Abbildung 4-1: Abbildung der Wirklichkeit durch Systeme	76
Abbildung 4-2: Übersicht der Grundprinzipien des <i>Systems Engineering</i>	78
Abbildung 4-3: Original, Modell und Verwender	79
Abbildung 4-4: Einordnung der Arbeit in das vorliegende Modellverständnis und Kapitelstruktur	81
Abbildung 4-5: Generelle Vorgehensweise der vorliegenden Arbeit	85
Abbildung 5-1: Übersicht und Zusammenhang der Inhalte des Kapitels 5	89
Abbildung 5-2: Vorgehen zur Begriffsdefinition der software-definierten Plattform	90
Abbildung 5-3: Inhalte der Definition einer software-definierten Plattform	92

Abbildung 5-4: Einordnung der Definition in die Infrastruktur des <i>Internet of Production</i>	96
Abbildung 5-5: Übersicht und grobe Ordnung der Merkmale software-definierter Plattformen	97
Abbildung 5-6: Vergleich relationaler und NoSQL-Datenbanken.....	101
Abbildung 5-7: Vierstufiges <i>Data Analytics Model</i>	104
Abbildung 5-8: Beispiel einer auf einer software-definierten Plattform erzeugten App	107
Abbildung 5-9: Screenshot einer Low-Code-Plattform aus einem Projekt des FIR. 109	
Abbildung 5-10: Schematische Übersicht der Typen software-definierter Plattformen	113
Abbildung 5-11: Dimensionen des Integrationsmanagements software-definierter Plattformen	122
Abbildung 5-12: Herleitung der Integrationsanforderungen aus den Plattfortmtypen je Dimension des Integrationsgegenstands.....	123
Abbildung 6-1: Übersicht und Zusammenhang der Inhalte des Kapitels 6	129
Abbildung 6-2: Ordnung der Merkmalsdimensionen für Informationssysteme.....	131
Abbildung 6-3: Schnittstellen zwischen vorhandenen Informationssystemen und software-defnierten Plattformen.....	133
Abbildung 6-4: Einordnung des Outsourcinggrads eines Informationssystems	135
Abbildung 6-5: Einordnung von harter und weicher Echtzeit in Abhängigkeit der Konsequenz.....	137
Abbildung 6-6: Nutzerzahl und Aktualität einer Software im Softwarelebenszyklus	139
Abbildung 6-7: Darstellung des Systemursprungs von Informationssystemen	141
Abbildung 6-8: Funktionale Organisationsstruktur von Unternehmen und Übertragung auf produzierende Unternehmen	144
Abbildung 6-9: Zusammenhang zwischen der Nutzerzahl und dem organisatorischen Integrationsaufwand	146
Abbildung 6-10: Einordnung der Ebenen in bestehende Schichtenmodelle	155
Abbildung 6-11: Dimensionen der Informationsqualität und deren Nutzung für die vorliegende Arbeit.....	158
Abbildung 6-12: Möglichkeiten der Standardisierung auf den Ebenen <i>Zeichen, Daten und Informationen</i>	161
Abbildung 6-13: Übersicht der Darstellungsvarianten von Informationssystem-Architekturen	164

Abbildung 6-14: Ausschnitt einer Informationssystem-Architektur (Lebenszyklusgrafik) mit Fokus auf den Softwarelebenszyklus	165
Abbildung 6-15: Ausschnitt einer Informationssystem-Architektur mit Fokus auf Schnittstellen und Echtzeitfähigkeit	166
Abbildung 6-16: Ausschnitt einer Informationssystem-Architektur mit Fokus auf die organisatorische Einordnung	168
Abbildung 6-17: Ausschnitt einer Informationssystem-Architektur mit Einordnung der Informationssysteme gemäß Kernprozessunterstützung und Organisationseinheit, zusätzliche Hervorhebung des Systemursprungs.....	169
Abbildung 6-18: Detaillierte Aufnahme von Informationssystem-Architekturen mit BPMN 2.0 am vereinfachten Beispiel eines Produktionsprozesses.....	170
Abbildung 6-19: Ausschnitt einer Informationssystem-Architektur mit Einordnung der Informationssysteme gemäß Kernprozessunterstützung und hauptsächlicher Ebenenzuordnung	171
Abbildung 6-20: Vereinfachter Ausschnitt aus der Datensicht einer Informationssystem-Architektur gemäß UML-Klassennotation	172
Abbildung 6-21: Beispielhafte Darstellung der Informationsqualität und Datenstandardisierung in einer Informationssystem-Architektur.....	173
Abbildung 6-22: Einordnung der Informationssystem-Architekturtypen	175
Abbildung 6-23: Verhältnis der Mitarbeiter zu IT-Mitarbeitern (eigen und fremd) und Einordnung der Beispielunternehmen	176
Abbildung 6-24: Beispiel für eine Informationssystem-Architektur nach Typ I eines Kleinserienproduzenten „Unternehmen I“	177
Abbildung 6-25: Beispiel für eine Informationssystem-Architektur nach Typ II eines Sondermaschinenbauers „Unternehmen II“.....	180
Abbildung 6-26: Beispiel für eine Informationssystem-Architektur nach Typ III eines Sondermaschinenbauers „Unternehmen III“.....	182
Abbildung 6-27: Beispiel für eine Informationssystem-Architektur nach Typ IV eines Nahrungsmittelproduzenten „Unternehmen IV“	184
Abbildung 6-28: Beispiel für eine Informationssystem-Architektur nach Typ V eines Metallproduzenten „Unternehmen V“	186
Abbildung 7-1: Übersicht und Zusammenhang der Inhalte des Kapitels 7	189
Abbildung 7-2: Eigenschaften der Zielinformationssystem-Architektur in der Dimension <i>Technik</i>	192
Abbildung 7-3: Beispielhafte Darstellung einer Zielinformationssystem-Architektur für Unternehmen I mit Fokus auf Schnittstellen	194

Abbildung 7-4: Eigenschaften der Zielinformationssystem-Architektur in der Dimension <i>Organisation</i>	195
Abbildung 7-5: Beispiel einer Organisationsstruktur einer IT-Abteilung in der Zielarchitektur	197
Abbildung 7-6: Beispielhafte Darstellung einer Zielinformationssystem-Architektur für Unternehmen I mit Fokus auf organisatorische Merkmale.....	198
Abbildung 7-7: Eigenschaften der Zielinformationssystem-Architektur in der Dimension <i>Prozesse</i>	199
Abbildung 7-8: Beispielhafte Darstellung einer Zielinformationssystem-Architektur für Unternehmen I mit Fokus auf prozessuale Merkmale	201
Abbildung 7-9: Eigenschaften der Zielinformationssystem-Architektur in der Dimension <i>Daten</i>	203
Abbildung 7-10: Vorgehen zur Herleitung der Zusammenhänge zwischen Ist- und Ziel-Zustand von Informationssystem-Architekturen.....	206
Abbildung 7-11: Zuordnung der Ist-Architekturtypen zur Zielbeschreibung	207
Abbildung 7-12: Vorgehen zur Herleitung der Hierarchisierung der Entscheidungen zur Ziel-Architektur in Abhängigkeit vom Plattformtyp	210
Abbildung 8-1: Einordnung der entwickelten Modelle in den Problemlösungszyklus nach HABERFELLNER ET AL.	213
Abbildung 8-2: Vorgehen zur Erhebung der Informationssysteme	218
Abbildung 8-3: Beispiele von Sichten auf Informationssystem-Architekturen mit markierten Herausforderungen.....	219
Abbildung 8-4: Auswahl und Einführung einer software-definierten Plattform	221
Abbildung 9-1: Übersicht der Fallstudien und deren Abdeckung des Gestaltungsmodells	226
Abbildung 9-2: Produktionsstätte von e.GO in Aachen.....	227
Abbildung 9-3: Historie der Informationssystem-Einführung bei e.GO	228
Abbildung 9-4: Organigramm der IT-Organisation bei e.GO	230
Abbildung 9-5: Prototypische Umsetzung des <i>Engineering Dashboards</i>	231
Abbildung 9-6: Prototypische Umsetzung des End-to-End-Monitorings in einer Prototypenfabrik	232
Abbildung 9-7: Prototypische Umsetzung des digitalen Zwillings	233
Abbildung 9-8: Model-based Systems-Engineering im e.GO Life	233
Abbildung 9-9: Übersicht Sonderbestellungsprozess e.GO Life	234
Abbildung 9-10: Entstehung der Fahrzeugdatenstruktur des e.GO Life	236

Abbildung 9-11: Ausschnitt der Informationssystem-Architektur bei e.GO mit Einordnung der Informationssysteme gemäß Kernprozessunterstützung und Organisationseinheit, zusätzliche Hervorhebung des Systemursprungs	242
Abbildung 9-12: Ausschnitt der Informationssystem-Architektur bei e.GO mit Fokus auf die organisatorische Einordnung	243
Abbildung 9-13: Ausschnitt der Informationssystem-Architektur bei e.GO mit Fokus auf Schnittstellen und Echtzeitfähigkeit	244
Abbildung 9-14: Von e.GO genutzte Darstellungen der Informationssystem-Architektur	245
Abbildung 9-15: Übersicht des Softwareauswahlprozesses aus Sicht von PTC.....	248
Abbildung 9-16: Ausschnitt aus der Informationssystem-Architektur von „PTC-Kunde A“	252
Abbildung 9-17: Ausschnitt aus der Informationssystem-Architektur von „PTC-Kunde B“	255
Abbildung 9-18: Aggregierte Darstellung des neu mit ThingWorx umgesetzten Change-Workflows bei "PTC-Kunde B"	256
Abbildung A-1: Übersicht über die zehn Quellen mit den meisten Zitierungen und deren Einfluss auf die einzelnen Kapitel der Arbeit	305
Abbildung A-2: Veränderung des Interesses an den Suchanfragen zu ERP, CRM, PLM und IoT bei Google	307
Abbildung A-3: Veränderung des Interesses an den Suchanfragen zu SCM, MES, CPS, ECM und BI bei Google.....	307
Abbildung A-4: Vorschlag einer erweiterten Referenzarchitektur mit möglichen Eigenschaften software-definierter Plattformen	310

Tabellenverzeichnis

Tabelle 2-1: Verschiedene Bedeutung von Informationen	15
Tabelle 2-2: Architekturtypen und Schichten von Informationssystemen in der betrieblichen Anwendung	20
Tabelle 2-3: Aufgaben des Informationsmanagements	34
Tabelle 2-4: Präzisierung des Untersuchungsbereichs	43
Tabelle 3-1: Übersicht untersuchter Quellen zum Stand der Erkenntnisse	71
Tabelle 4-1: Aufbau eines morphologischen Kastens zur Typisierung von Fahrzeugen am Beispiel des Typs „e.GO Life“, einem Elektrofahrzeug	82
Tabelle 5-1: Aufgaben einer software-definierten Plattform.....	94
Tabelle 5-2: Ausprägungen des Merkmals <i>Schnittstellen</i>	98
Tabelle 5-3: Ausprägungen des Merkmals <i>Datenmanagement</i>	99
Tabelle 5-4: Ausprägungen des Merkmals <i>Datenbankkonzept</i>	101
Tabelle 5-5: Ausprägungen des Merkmals <i>Stufen der Datenverarbeitung</i>	103
Tabelle 5-6: Ausprägungen des Merkmals <i>IT-Sicherheitsfunktionen</i>	105
Tabelle 5-7: Ausprägungen des Merkmals <i>Informationsbereitstellung</i>	106
Tabelle 5-8: Ausprägungen des Merkmals <i>Entwicklungsumgebung</i>	108
Tabelle 5-9: Merkmale software-definierter Plattformen	110
Tabelle 5-10: Software-definierte Plattformen zum Einsatz bei produzierenden Unternehmen.....	112
Tabelle 5-11: Typ I Vollumfängliches System.....	116
Tabelle 5-12: Typ II Eingeschränkter Funktionsumfang	118
Tabelle 5-13: Typ III Reine Vernetzung	120
Tabelle 6-1: Ausprägungen des Merkmals <i>Schnittstellen</i>	132
Tabelle 6-2: Ausprägungen des Merkmals <i>Outsourcinggrad</i>	134
Tabelle 6-3: Ausprägungen des Merkmals <i>Echtzeitfähigkeit</i>	137
Tabelle 6-4: Ausprägung des Merkmals <i>Status des Softwarelebenszyklus</i>	138
Tabelle 6-5: Ausprägungen des Merkmals <i>Systemursprung</i>	140
Tabelle 6-6: Ausprägungen des Merkmals <i>Systemverantwortlicher</i>	142
Tabelle 6-7: Ausprägungen des Merkmals <i>Organisationseinheit</i>	144
Tabelle 6-8: Ausprägungen des Merkmals <i>Nutzerkreis</i>	145
Tabelle 6-9: Ausprägungen des Merkmals <i>Systemkonzept</i>	147
Tabelle 6-10: Ausprägungen des Merkmals <i>Kernprozess</i>	151

Tabelle 6-11: Ausprägungen des Merkmals <i>Betroffene Ebenen</i>	154
Tabelle 6-12: Ausprägungen des Merkmals <i>Informationsqualität</i>	157
Tabelle 6-13: Ausprägungen des Merkmals <i>Datenstandardisierung</i>	160
Tabelle 6-14: Merkmale von Informationssystemen	162
Tabelle 6-15: Ausschnitt einer Liste an Informationssystemen eines produzierenden Unternehmens	163
Tabelle 7-1: Zieldimensionen der Informationssystem-Architektur in Abhängigkeit des gewählten Plattfortmtyps	211
Tabelle 8-1: Aufgaben einer software-definierten Plattform in Abhängigkeit von konkreten Anwendungsfällen.....	215
Tabelle 8-2: Exemplarische Morphologie der Eigenschaften software-definierter Plattformen als Auswahlhilfe zur Softwareauswahl	216
Tabelle 8-3: Integrationsanforderungen und beispielhafte Konkretisierungen für ein Unternehmen.....	217
Tabelle 9-1: Aufgaben der software-definierten Plattform bei e.GO	237
Tabelle 9-2: Eigenschaften der software-definierten Plattform <i>PTC ThingWorx</i> bei e.GO	238
Tabelle 9-3: Integrationsanforderungen und Konkretisierung bei e.GO	239
Tabelle 9-4: Notwendige Ausprägungen der software-definierten Plattform für „PTC-Kunde A“	251
Tabelle 9-5: Notwendige Ausprägungen der software-definierten Plattform für „PTC-Kunde B“	254
Tabelle A-1: Übersicht über die zentralen Literaturquellen	302
Tabelle A-2: Experteninterviews	306
Tabelle A-3: Typisierung software-definierter Plattformen	308
Tabelle A-4: Detaillierter Vergleich der Schichtenmodelle	311
Tabelle A-5: Liste der Informationssysteme bei e.GO	314

Abkürzungsverzeichnis

5G	<u>5</u> th <u>G</u> eneration of Mobile Networks
A.	<u>A</u> usprägung
acatech	Deutsche <u>A</u> kademie der <u>T</u> echnikwissenschaften e. V.
AES	<u>A</u> dvanced <u>E</u> ncryption <u>S</u> tandard
ALM	<u>A</u> pplication- <u>L</u> ifecycle- <u>M</u> anagement
ANSI	<u>A</u> merican <u>N</u> ational <u>S</u> tandards <u>I</u> nstitute
Anw.	<u>A</u> nwendung
API	<u>A</u> pplication- <u>P</u> rogramming- <u>I</u> nterface
Applik.	<u>A</u> pplik <u>a</u> tion
ARIS	<u>A</u> rchitektur integrierter <u>I</u> nformationss <u>s</u> ysteme
ASaP	<u>A</u> gentenbasierte <u>S</u> mart-Service- <u>P</u> lattform
ASP	<u>A</u> pplication- <u>S</u> ervice- <u>P</u> rovisioning
BCG	<u>B</u> oston <u>C</u> onsulting <u>G</u> roup, <i>eine Managementberatung</i>
BI	<u>B</u> usiness- <u>I</u> ntelligence
BOM	<u>B</u> ill <u>o</u> f <u>M</u> aterial
BPMN	<u>B</u> usiness- <u>P</u> rocess- <u>M</u> odelling- <u>N</u> otation
BWL	<u>B</u> etriebs <u>w</u> irtschafts <u>l</u> ehre
CAD	<u>C</u> omputer- <u>a</u> ided <u>D</u> esign
CAQ	<u>C</u> omputer- <u>a</u> ided <u>Q</u> uality
CD	<u>C</u> ompact <u>D</u> isk
CEO	<u>C</u> hief <u>E</u> xecutive <u>O</u> fficer
CFO	<u>C</u> hief <u>F</u> inancial <u>O</u> fficer
CIM	<u>C</u> omputer- <u>i</u> ntegrated <u>M</u> anufacturing
CIO	<u>C</u> hief <u>I</u> nformation <u>O</u> fficer
CLASP	<u>C</u> omprehensive- <u>L</u> ightweight- <u>A</u> pplication- <u>S</u> ecurity- <u>P</u> rocess
CMDB	<u>C</u> onfiguration- <u>M</u> anagement- <u>D</u> atabase
CNC	<u>C</u> omputerized <u>N</u> umerical <u>C</u> ontrol
CPS	<u>C</u> yber- <u>p</u> hysical <u>S</u> ystems, <i>cyberphysische Systeme</i>
CRM	<u>C</u> ustomer- <u>R</u> elationship- <u>M</u> anagement
CSO	<u>C</u> hief <u>S</u> trategy <u>O</u> fficer

CSV	<u>C</u> omma- <u>s</u> eparated <u>V</u> alues
CTO	<u>C</u> hief <u>T</u> echnology <u>O</u> fficer
DIN	<u>D</u> eutsches <u>I</u> nstitut für <u>N</u> ormung e. V.
DMS	<u>D</u> okumenten <u>m</u> anagement <u>s</u> ystem
DSGVO	<u>D</u> atens <u>s</u> chutz- <u>G</u> rund <u>v</u> er <u>o</u> rdnung
DV	<u>D</u> aten <u>v</u> erarbeitung
E.	<u>E</u> ntwicklungsumgebung
e.GO	<u>e</u> . <u>G</u> O Mobile AG, <i>ein Elektroautomobilhersteller aus Aachen</i>
E4TC	<u>E</u> uropean <u>4</u> . <u>0</u> <u>T</u> ransformation <u>C</u> enter GmbH, <i>ein Center am Cluster Smart Logistik auf dem RWTH Aachen Campus</i>
EAI	<u>E</u> nterprise- <u>A</u> rchitecture- <u>I</u> ntegration
EAM	<u>E</u> nterprise- <u>A</u> rchitecture- <u>M</u> anagement
eBOM	<u>e</u> ngineering <u>B</u> OM
eCAD	<u>e</u> lektrisches <u>C</u> AD
ECM	<u>E</u> nterprise- <u>C</u> ontent- <u>M</u> anagement
EDI	<u>E</u> lectronic <u>D</u> ata- <u>I</u> nterchange
EDIFACT	<u>E</u> lectronic <u>D</u> ata- <u>I</u> nterchange for <u>A</u> dministration, <u>C</u> ommerce and <u>T</u> ransport
Entw.	<u>E</u> ntwicklung
EPK	<u>e</u> reignisgesteuerte <u>P</u> rozess <u>k</u> etten
ERM	<u>E</u> ntity- <u>R</u> elationship- <u>M</u> odel
ERP	<u>E</u> nterprise- <u>R</u> esource- <u>P</u> lanning
ESB	<u>E</u> nterprise- <u>S</u> ervice- <u>B</u> us
EU	<u>E</u> uropäische <u>U</u> nion
FDDI	<u>F</u> iber- <u>d</u> istributed <u>D</u> ata- <u>I</u> nterface
FDIS	<u>F</u> inal- <u>D</u> raft- <u>I</u> nternational- <u>S</u> tandard
FIR	<u>F</u> orschungs <u>i</u> nstitut für <u>R</u> ationalisierung e. V. an der RWTH Aachen
FMEA	<u>F</u> ehler <u>m</u> öglichkeiten- und - <u>e</u> influss <u>a</u> nalyse
FTE	<u>F</u> ull- <u>T</u> ime- <u>E</u> quivalent
FTP	<u>F</u> ile- <u>T</u> ransfer- <u>P</u> rotocol
GE	<u>G</u> eneral <u>E</u> lectric Company, <i>ein amerikanischer Mischkonzern</i>
HTTP	<u>H</u> yper <u>t</u> ext- <u>T</u> ransfer- <u>P</u> rotocol

HW	<u>H</u> ard <u>w</u> are
IaaS	<u>I</u> nfrastructure <u>a</u> s <u>a</u> <u>S</u> ervice
IBAN	<u>I</u> nternational <u>B</u> ank- <u>A</u> ccount- <u>N</u> umber
IBM	<u>I</u> nternational <u>B</u> usiness <u>M</u> achines <u>C</u> orporation, <i>ein amerikanisches IT-Unternehmen</i>
IEC	<u>I</u> nternational <u>E</u> lectrotechnical <u>C</u> ommission
IEEE	<u>I</u> nstitute of <u>E</u> lectrical and <u>E</u> lectronics <u>E</u> ngineers
IIC	<u>I</u> ndustrial- <u>I</u> nternet- <u>C</u> onsortium
IIoT	<u>I</u> ndustrial <u>I</u> nternet <u>o</u> f <u>T</u> hings
IIRA	<u>I</u> ndustrial- <u>I</u> nternet- <u>R</u> eference- <u>A</u> rchitecture
IIoT	<u>I</u> ndustrial <u>I</u> nternet <u>o</u> f <u>T</u> hings
IKT	<u>I</u> nformations- und <u>K</u> ommunikationstechnologie
IM	<u>I</u> nformations <u>m</u> anagement
IoT	<u>I</u> nternet <u>o</u> f <u>T</u> hings
IP	<u>I</u> nternet- <u>P</u> rotocol
IPA	Fraunhofer- <u>I</u> nstitut für <u>P</u> roduktionstechnik und <u>A</u> utomatisierung
iPaaS	<u>I</u> ntegration- <u>P</u> latform <u>a</u> s <u>a</u> <u>S</u> ervices
IQ	<u>I</u> nformationsqualität
IRK	<u>I</u> IoT- <u>R</u> eferenzarchitektur für die <u>K</u> ommunikation
IRM	' <u>I</u> nternet of Things'- <u>R</u> eference- <u>M</u> odel
IS	<u>I</u> nformationss <u>s</u> ystem
ISO	<u>I</u> nternational <u>O</u> rganization for <u>S</u> tandardization, <i>die internationale Organisation für Normung</i>
IT	<u>I</u> nformationstechnologie
ITIL	<u>I</u> T- <u>I</u> nfrastructure- <u>L</u> ibrary
JSON	<u>J</u> ava <u>S</u> cript- <u>O</u> bject- <u>N</u> otation
KMU	<u>k</u> leine und <u>m</u> ittlere <u>U</u> nternehmen
KPI	<u>K</u> ey- <u>P</u> erformance- <u>I</u> ndicator
LDAP	<u>L</u> ightweight- <u>D</u> irectory- <u>A</u> ccess- <u>P</u> rotocol
mBOM	<u>m</u> anufacturing <u>B</u> OM
MBSE	<u>M</u> odel- <u>b</u> ased <u>S</u> ystems- <u>E</u> ngineering
mCAD	<u>m</u> echanisches <u>C</u> AD

MES	<u>M</u> anufacturing- <u>E</u> xecution- <u>S</u> ystem
MQ	<u>M</u> essage- <u>Q</u> ueue
MSB	<u>M</u> anufacturing- <u>S</u> ervice- <u>B</u> us
NC	<u>N</u> umeric <u>C</u> ontrol
NoSQL	<u>n</u> ot <u>o</u> nly <u>S</u> QL
OBD	<u>O</u> n- <u>B</u> oard <u>D</u> iagnose (auch: OBD II)
OPC-UA	<u>O</u> pen- <u>P</u> latform- <u>C</u> ommunications - <u>U</u> nified <u>A</u> rchitecture
OSI	<u>O</u> pen- <u>S</u> ystems- <u>I</u> nterconnection Model
OT	<u>O</u> perational <u>T</u> echnology
OWASP	<u>O</u> pen- <u>W</u> eb- <u>A</u> pplication- <u>S</u> ecurity- <u>P</u> roject
PaaS	<u>P</u> latform <u>a</u> s <u>a</u> <u>S</u> ervice
PDF	<u>P</u> ortable- <u>D</u> ocument- <u>F</u> ormat
PDM	<u>P</u> rodukt <u>d</u> aten <u>m</u> anagement
PLM	<u>P</u> roduct- <u>L</u> ifecycle- <u>M</u> anagement
PPD	, <u>P</u> räsentation, <u>P</u> rozess und <u>D</u> atenhaltung'-Modell
PPS	<u>P</u> roduktions <u>p</u> lanung und - <u>s</u> teuerung
Prod.	<u>P</u> roduktion
PTC	PTC Inc., <i>ein amerikanischer Softwarehersteller</i>
QMS	<u>Q</u> ualitätsmanagementsystem
RAMI4.0	<u>R</u> eferenz <u>a</u> rchitektur <u>m</u> odell <u>I</u> ndustrie <u>4.0</u>
REST	<u>R</u> epresentational <u>S</u> tate- <u>T</u> ransfer
RSA	, <u>R</u> ivest, <u>S</u> hamir und <u>A</u> dleman'-Kryptosystem
RWTH	<u>R</u> heinisch- <u>W</u> estfälische <u>T</u> echnische <u>H</u> ochschule
SaaS	<u>S</u> oftware <u>a</u> s <u>a</u> <u>S</u> ervice
SAP	SAP SE, <i>ein deutscher Softwarehersteller</i>
sBOM	<u>S</u> ervice <u>B</u> OM
SCADA	<u>S</u> upervisory- <u>C</u> ontrol <u>a</u> nd <u>D</u> ata- <u>A</u> cquisition
SCM	<u>S</u> upply- <u>C</u> hain- <u>M</u> anagement
SCOR	<u>S</u> upply- <u>C</u> hain- <u>O</u> perations- <u>R</u> eference-Model
SDK	<u>S</u> oftware- <u>D</u> evelopment- <u>K</u> its
SDL	<u>S</u> ecure- <u>D</u> evelopment- <u>L</u> ifecycle
SIMILAR	<u>S</u> tate, <u>I</u> nvestigate, <u>M</u> odel, <u>I</u> ntegrate, <u>L</u> aunch, <u>A</u> ssess, and <u>R</u> e-evaluate

SIP	<u>S</u> imple- <u>I</u> terative- <u>P</u> artitions
SLM	<u>S</u> ervice- <u>L</u> ifecycle- <u>M</u> anagement
SMTP	<u>S</u> imple- <u>M</u> ail- <u>T</u> ransfer- <u>P</u> rotocol
SOA	<u>s</u> ervice <u>o</u> rientierte <u>A</u> rchitektur
SOAP	<u>S</u> imple- <u>O</u> bject- <u>A</u> ccess- <u>P</u> rotocol
SOM	<u>s</u> emantisches <u>O</u> bjektmodell
SPS	<u>s</u> peicher <u>p</u> rogrammierbare <u>S</u> teuerung
SQL	<u>S</u> tructured <u>Q</u> uery- <u>L</u> anguage
SRM	<u>S</u> upplier- <u>R</u> elationship- <u>M</u> anagement
SSoT	<u>S</u> ingle <u>S</u> ource of <u>T</u> ruth
SW	<u>S</u> oft <u>w</u> are
SWIFT	<u>S</u> ociety for <u>W</u> orldwide <u>I</u> nterbank <u>F</u> inancial <u>T</u> elecommunication
SysML	<u>S</u> ystems- <u>M</u> odeling- <u>L</u> anguage
TCP	<u>T</u> ransmission- <u>C</u> ontrol- <u>P</u> rotocol
TOGAF	<u>T</u> he- <u>O</u> pen- <u>G</u> roup-Enterprise- <u>A</u> rchitecture- <u>F</u> ramework
TPM	<u>T</u> rusted- <u>P</u> latform- <u>M</u> odules
UDP	<u>U</u> ser- <u>D</u> atagram- <u>P</u> rotocol
UML	<u>U</u> nified- <u>M</u> odeling- <u>L</u> anguage
USB	<u>U</u> niversal- <u>S</u> erial- <u>B</u> us
VDA	<u>V</u> erband <u>d</u> er <u>A</u> utomobilindustrie e. V.
VDI	<u>V</u> erein <u>D</u> eutscher <u>I</u> ngenieure e. V.
VDMA	<u>V</u> erband <u>D</u> eutscher <u>M</u> aschinen- und <u>A</u> nlagenbau e. V.
Vertr.	<u>V</u> er <u>t</u> rieb
VIN	<u>V</u> ehicle- <u>I</u> dentification- <u>N</u> umber
VWL	<u>V</u> olkswirtschafts <u>l</u> ehre
WGP	<u>W</u> issenschaftliche <u>G</u> esellschaft für <u>P</u> roduktionstechnik
WYSIWYG	" <u>W</u> hat <u>Y</u> ou <u>S</u> ee <u>I</u> s <u>W</u> hat <u>Y</u> ou <u>G</u> et"
WZL	<u>W</u> erk <u>z</u> eugmaschinen <u>l</u> abor der RWTH Aachen
XML	<u>E</u> xtensible- <u>M</u> arkup- <u>L</u> anguage
ZF	ZF Friedrichshafen AG, <i>ein Automobilzulieferer aus Friedrichshafen</i>
ZVEI	<u>Z</u> entral <u>v</u> erband <u>E</u> lektrotechnik- und Elektronik <u>i</u> ndustrie e. V.

1 Einleitung

1.1 Ausgangssituation und Problemstellung

Im Zuge der aktuell zunehmenden und immer stärker vernetzten Digitalisierung entstehen **neue Geschäftsmodelle** (s. BAUERNHANS� ET AL. 2016, S. 3), welche erst **durch den Einsatz von Informationstechnologie ermöglicht** werden (s. LEIMEISTER 2015, S. 3-4). Den Erfolg dieser Geschäftsmodelle zeigen Statistiken wie der Markenwert von Unternehmen, bei denen Marken, die mit digitalen Technologien assoziiert werden, neun der zehn wertvollsten Marken ausmachen (s. SCHEPT 2017, S. 20-21; Abbildung 1-1 links). Den Kern von deren unternehmerischer Tätigkeit bilden Plattformen, über welche die Geschäftsmodelle der Unternehmen skalieren. Der Einsatz vernetzter Geräte, welche die Grundlage für solche Geschäftsmodelle bilden, wird hierbei sowohl im privaten als auch im betrieblichen Bereich noch zunehmen (s. GARTNER 2017c; Abbildung 1-1 rechts). 55 Prozent der **Industrieunternehmen in Deutschland haben diesen Trend inzwischen erkannt** und sehen für sich Wachstumspotenzial (s. ATKEARNEY U. BDI, 2017, S. 14).

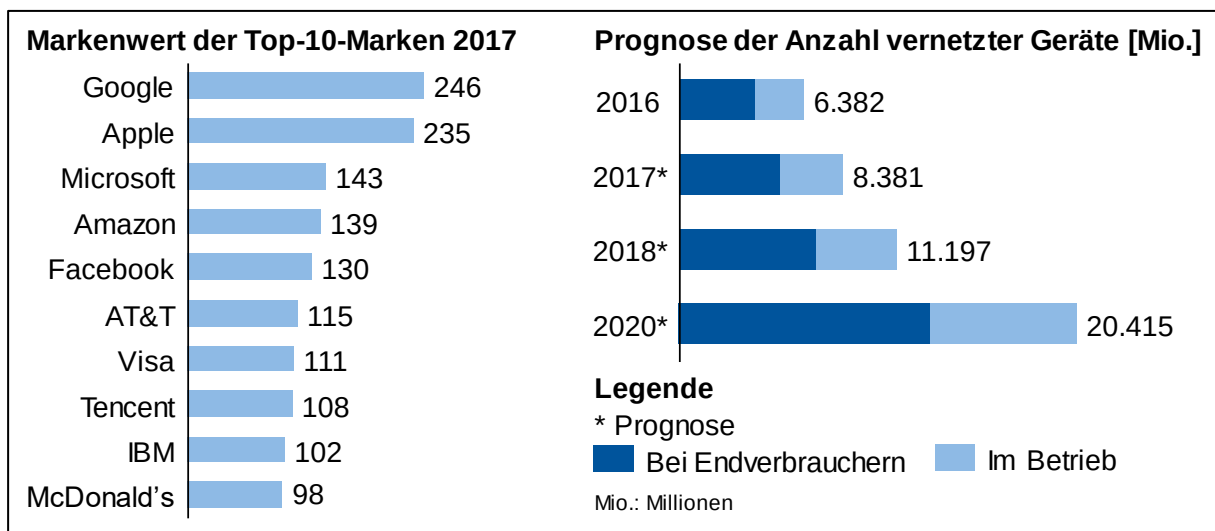


Abbildung 1-1: Status quo der vernetzten Digitalisierung (eigene Darstellung i. A. a. SCHEPT 2017, S. 20-21; GARTNER 2017c)

Unternehmensintern führen die beschriebenen Entwicklungen zu einer fortschreitenden Zunahme der Datenbestände, die in unterschiedlichsten Informationssystemen (IS) gehalten und bearbeitet werden (s. PLATTNER U. KRÜGER 2015, S. 445f). **Diese Daten nutzen viele Unternehmen jedoch zu wenig, um zum Beispiel ihre Produktionsabläufe zu optimieren oder eine verbesserte Übersicht über ihre Kunden zu bekommen** (s. u. a. in OTTO U. ÖSTERLE 2016, S. 18-20). Unternehmen erkennen den

Wert dieser Daten jedoch immer stärker und fordern von ihren IT-Abteilungen schnell neue Lösungen (s. ERWIN ET AL., S. 7; GREBE ET AL. 2016).¹

IT-Leiter und CIOs stehen vor großen Herausforderungen im Umgang mit den aus den beschriebenen Entwicklungen resultierenden Aufgaben und Anforderungen, da eine deutlich **stärkere horizontale und vertikale Integration der Informationssysteme benötigt** wird. Sie müssen ihre bisherige Informationssystem-Architektur daher komplett hinterfragen, weil die bisherigen heterogenen Informationssysteme an technische Grenzen stoßen (s. WEINREICH 2016, S. 102). Die Wichtigkeit von Informationssystemen in Unternehmen verschiebt sich gleichzeitig: Während bisher ERP-Systeme eine zentrale Rolle in der Informationssystem-Architektur innehatten, nimmt deren Bedeutung in den letzten Jahren ab. Stattdessen steigt die Relevanz des *Internet of Things* (IoT), wie unter anderem Google-Suchstatistiken zeigen (s. Anhang A.4). In Gesprächen mit IT-Leitern verschiedener Unternehmen im Zuge der Vorbereitungen auf das in der vorliegenden Arbeit behandelte Thema (s. Anhang A.3) zeigten sich vor allem zwei Aspekte: Einerseits wurde das **wachsende Aufgabenspektrum der IT-Abteilung** benannt, gleichzeitig verbunden mit dem Ziel, **durchgängige Lösungen für die verschiedenen Unternehmensbereiche** anzubieten. Andererseits wurde die **über die Jahre gewachsene interne Informationssystem-Architektur** beklagt, die zunehmend schwerer zu beherrschen sei. Betrachtet man diese beiden Hauptaussagen, so zeigt sich, dass der von extern gestellte Anspruch mit der internen Wirklichkeit in der Informationssystem-Architektur nicht übereinstimmt. IT-Leiter benötigen eine Möglichkeit, agil neue Anforderungen bezogen auf Daten und Funktionen umsetzen zu können, während sie gleichzeitig einen stabilen IT-Betrieb sicherstellen müssen.

Die Informationssystem-Architektur erfährt dadurch einen tiefgreifenden Wandel, der durch zusätzliche Umfänge und inhaltliche wie technische Anforderungen getrieben ist. Die Digitalisierung erfordert eine Vielzahl neuer Schnittstellen innerhalb und außerhalb der Unternehmen. Neue Partner müssen angebunden und Ansprüche an Datenzugriffe über den kompletten Lebenszyklus in komplexen Produktionsnetzwerken abgebildet und in bestehende Unternehmensprozesse integriert werden. Dabei umfasst die Informationssystem-Architektur nicht mehr nur die Business-Support-IT, sondern auch die Produktions-IT (s. BLEIDER ET AL. 2016, S. 11). Die Informationssystem-Architektur muss in der Lage sein, benötigte Informationen an der richtigen Stelle und in der benötigten Qualität bereitzustellen und gleichzeitig die Sicherheit der Informationen zu gewährleisten. Die hierzu verfügbare große Vielzahl an am Markt angebotenen Informationssystemen, die vom Unternehmen in Betracht gezogen werden müssen, stellt hierbei eine weitere Herausforderung dar. Diese in eine beste-

¹ Der IT-Begriff wird (nicht nur) im deutschen Sprachraum inflationär verwendet. Er wird in der vorliegenden Arbeit als Informationstechnologie verstanden und nur dort genutzt, wo sich seine Anwendung in Wortverbindungen etabliert hat (z. B. IT-Abteilung, IT-Leiter etc.) – s. auch Kapitel 2.2.1.

hende Umgebung einzubinden, stellt sich häufig als problematisch dar; meist sind Änderungen oder teilweise Abschaltungen an Alt-Systemen zur Umsetzung notwendig (S. GREBE ET AL. 2016).

Aufgrund der Vielzahl an zu betrachtenden Elementen und deren schneller Änderung **erhöht sich maßgeblich die Komplexität** (nach Definition von SCHUH 2005, S. 5-6) **der Informationssystem-Architektur**, im folgenden **IT-Komplexität** genannt (S. SCHUH ET AL. 2017e, S. 61). Die steigende IT-Komplexität erschwert die Steuerung der Informationssystem-Architektur, führt dadurch zu erhöhten Kosten und verhindert neue digitale Services (S. DERN 2011, S. 87; GREBE ET AL. 2016; LEMKE ET AL. 2017, S. 501-502).

Um die genannten Herausforderungen anzugehen, wurden schon verschiedentlich Konzepte entwickelt, mit denen die Datennutzung in Unternehmen verbessert, die IT-Komplexität reduziert und die Technologieintegration vereinfacht werden können. Allein die **Umsetzung der Konzepte ist bis heute nicht flächendeckend erfolgt**. Grundlagen wurden bei der Entwicklung des Computer-Integrated-Manufacturing-Konzepts (CIM) vor 30 Jahren gelegt, die seinerzeit jedoch theoretischer Natur blieben (vgl. FIEDLER U. REGENHARD 1991). Ein nächster Schwerpunkt entstammt dem *Enterprise-Architecture-Integration*-Konzept (EAI), das vor etwa 15 Jahren entwickelt wurde (vgl. KAIB 2002). Beide Konzepte betonen die Notwendigkeit, Insellösungen zugunsten einer integrierten Informationssystem-Architektur abzuschaffen. Aktuell wird das **Konzept des digitalen Schattens** diskutiert. Der Begriff *Digitaler Schatten* beschreibt das relevante Abbild der Realität in Echtzeit (S. SCHUH ET AL. 2017h, S. 120-121). Im selben Zuge adressieren verschiedene Autoren auch die *Single Source of Truth* (SSoT), welchen den Ansatz einer zentralen Zugriffsmöglichkeit auf alle Unternehmensdaten fordert (S. KRUMM ET AL. 2015, S. 13-16). Bei einer zersplitterten Informationssystem-Architektur mit nicht-integrierten Informationssystemen, die nicht über echtzeitfähige Schnittstellen verfügen, ist die Umsetzung des digitalen Schattens und der SSoT allerdings nur schwer möglich.

Eine **Lösungsmöglichkeit** ist der **Einsatz von software-definierten Plattformen**. Solche Plattformen können als Intermediär den Zugriff auf heterogene Datenbestände der im Unternehmen vorhandenen Informationssysteme ermöglichen. Allerdings bedarf es dazu einer Anpassung der Informationssystem-Architektur, um den vollen Nutzen der Plattform zu ermöglichen. Dazu **fehlt produzierenden Unternehmen allerdings ein konkretes Vorgehensmodell**, sowohl zur Vorbereitung der Informationssystem-Architektur als auch der Auswahl und Einführung einer solchen Plattform.

1.2 Zielsetzung und Forschungsfrage

In der vorliegenden Arbeit werden die beschriebenen Problemstellungen der IT-Komplexität und der fehlenden Möglichkeit der zentralen Nutzung von Daten durch die Entwicklung eines strukturierten Ansatzes angegangen. Dabei soll die Arbeit einen Beitrag zu den folgenden **übergeordneten Zielen** des Informationsmanagements leisten: Primär soll die **zu hohe Komplexität von Informationssystem-Architekturen in**

Unternehmen – also die IT-Komplexität – besser beherrschbar gestaltet werden. Hierzu werden die drei Dimensionen der IT-Komplexität angegangen. Interdependente Vielfalt von Systemen, Daten und Schnittstellen werden durch Einführung einer SSoT reduziert und verbessert geordnet. Die verunsichernde Dynamik wird kanalisiert, indem sich schnell ändernde Anforderungen sowohl von Kunden als auch von Fachbereichen, nicht in den zentralen Informationssystemen wie Enterprise-Resource-Planning-Systemen (ERP-Systeme) oder Product-Lifecycle-Management-Systemen (PLM-Systeme) umgesetzt werden, sondern als Apps mit Zugriff auf die SSoT. Damit soll mit dieser Arbeit ein **Baustein zur Umsetzung der Industrie 4.0 in Unternehmen** bereitgestellt werden. Elementarer Bestandteil der Entwicklung zur Industrie 4.0 ist die Nutzung von Daten, um Sichtbarkeit, Transparenz und Prognosefähigkeit in verschiedenen Bereichen des Unternehmens zu erreichen. Diese Daten müssen nahezu in Echtzeit zentral bereitgestellt werden. Technisch ist dies heute möglich, Unternehmen müssen jedoch auch organisatorisch und prozessual bereit dazu sein.

Um einen Beitrag zur Erreichung der vorgestellten übergeordneten Ziele zu leisten, hat diese Arbeit zum **zentralen Ziel, aufzuzeigen, wie produzierende Unternehmen in ihre bestehende Informationssystem-Architektur eine zentrale software-definierte Plattform integrieren können** (s. Abbildung 1-2).

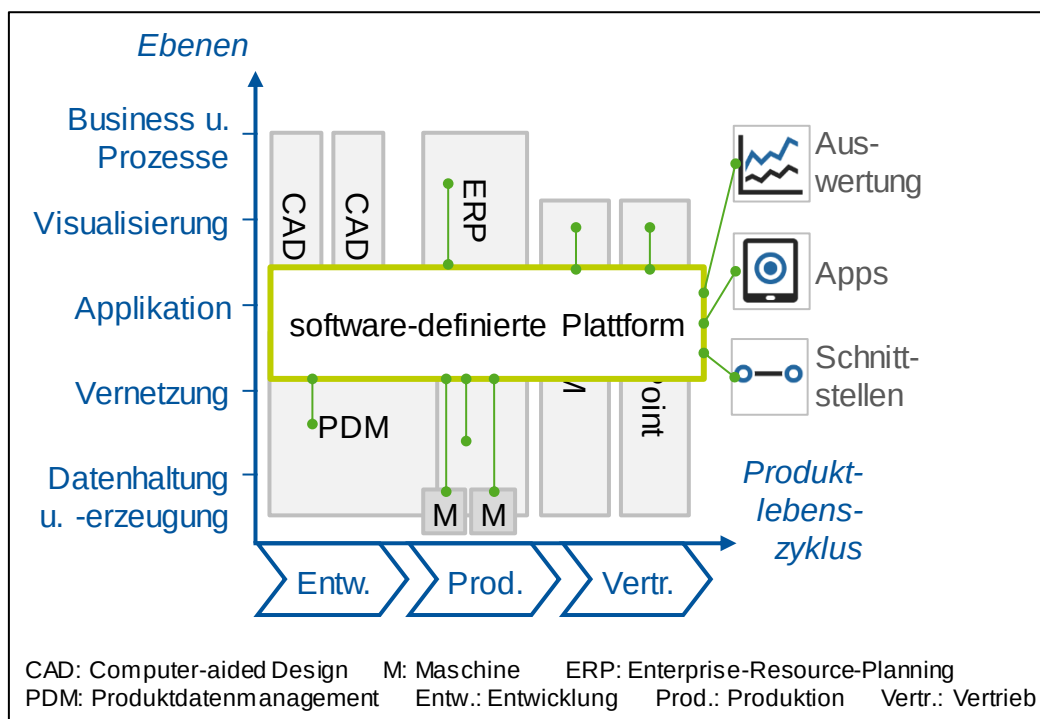


Abbildung 1-2: Einführung einer software-definierten Plattform in der bestehenden Informationssystem-Architektur (i. A. a. SCHUH ET AL. 2017e, S. 64)

Mit der Arbeit werden dementsprechend die folgenden **Teilziele** verfolgt:

(1) In den vergangenen Jahren der Entwicklung von Informationssystemen sowie der Informations- und Kommunikationstechnik im Ganzen haben sich unterschiedliche Arten von Plattformkonzepten entwickelt. Diese Arbeit möchte **eine strukturierte Über-**

sicht über vorhandene Plattformkonzepte geben, die für den Einsatz in produzierenden Unternehmen relevant sind. Unternehmen wird es damit ermöglicht, relevante Plattformkonzepte für sich zu identifizieren. Es wird dargestellt, wie diese in bestehende Informationssystem-Architekturen integriert werden können. Hiermit wird das Ziel beschrieben, welches von Unternehmen zu erreichen ist.

(2) Das dementsprechend zweite Ziel ist eine systematische Beschreibung des Ist-Stands, die Entwicklung einer einfachen und übergreifenden **systematischen Beschreibung von Informationssystem-Architekturen** sowie darauf aufbauend die Möglichkeit der Ordnung der Architekturen produzierender Unternehmen nach Standardtypen. Diese wird sowohl eine Darstellung für das Management ermöglichen als auch eine Verknüpfung dieser Visualisierung mit etablierten Darstellungsmethoden von Informationssystem-Architekturen aus dem Enterprise-Architecture-Management (EAM) anstreben. Ein weiteres Ziel ist es hierbei, eine einheitliche Sprache innerhalb von IT-Abteilungen einzuführen, um eine effiziente Kommunikation zwischen verschiedenen Bereichen zu ermöglichen.

(3) Als letztes Ziel möchte diese Arbeit die **Entscheidung von Unternehmen zur Einführung einer Plattform vorbereiten**, indem die Deltas zwischen Ist- und Ziel-Zustand aufgezeigt, notwendige Entscheidungen abgeleitet und ein konkretes Vorgehen vorgeschlagen werden. Damit werden neben einer wissenschaftlichen Ordnung der Ist- und Zielzustände pragmatische Empfehlungen für die Wirtschaft entwickelt. Das Dissertationsverfahren wird dabei ein Teil des Aachener Gesamtvorhabens der Entwicklung des *Internet of Production* (vgl. SCHUH ET AL. 2017h) bilden.

Um diese Ziele zu erreichen, werden in der vorliegenden Arbeit Informationssystem-Architekturen und software-definierte Plattformen beschrieben, deren Zusammenhänge erklärt und die konkrete Umsetzung der Ergebnisse in der Wirtschaft vorbereitet. Um die Anwendbarkeit dieser Ergebnisse sicherzustellen, werden sie gemäß den Prinzipien der Aktionsforschung (vgl. MCNIFF U. WHITEHEAD 2011; MCNIFF 2017) bei Industriepartnern genutzt und validiert. Ausgehend von dieser Zielsetzung ergibt sich folgende Forschungsfrage:

Wie können produzierende Unternehmen in ihre bestehende Informationssystem-Architektur eine zentrale software-definierte Plattform integrieren?

Daraus lassen sich die folgenden Sekundärfragen ableiten:

- Wie können zentrale software-definierte Plattformen und deren Integrationsanforderungen systematisch beschrieben werden?
- Wie können bestehende Informationssystem-Architekturen produzierender Unternehmen systematisch beschrieben werden?
- Wie kann anhand der Integrationsanforderungen der Unterschied zwischen Ist- und Ziel-Zustand für Unternehmen identifiziert werden?

- Wie kann der Integrationsprozess zur Einführung einer zentralen software-definierten Plattform in die Informationssystem-Architektur systematisch gestaltet werden?

1.3 Wissenschaftlicher Bezugsrahmen und Forschungskonzeption

Basierend auf der zuvor beschriebenen Zielsetzung soll nun zur grundsätzlichen Einordnung der vorliegenden Arbeit im Folgenden deren wissenschaftlicher Bezugsrahmen hergeleitet werden. Dazu ist ein **Blick in die Wissenschaftstheorie**, die sogenannte „Wissenschaftswissenschaft“ nötig (ULRICH U. HILL 1976a, S. 305). Eine einheitliche, allgemeinverbindliche Strukturierung der Wissenschaften, die für den Kontext der vorliegenden Arbeit relevant sind, existiert dabei nicht (s. ZELEWSKI 2008, S. 3), es hat sich jedoch eine Aufteilung anhand ihrer Erkenntnisobjekte in Formal-, Struktur- und Realwissenschaften durchgesetzt (vgl. dazu ULRICH U. HILL 1976a, S. 305; ZELEWSKI 2008, S. 4; DEINDL 2013b, S. 5; JUNG 2016, S. 22-23).² Abbildung 1-3 zeigt diese Aufteilung der Wissenschaften mit ihren jeweiligen Zielen und gibt Beispiele dazu.

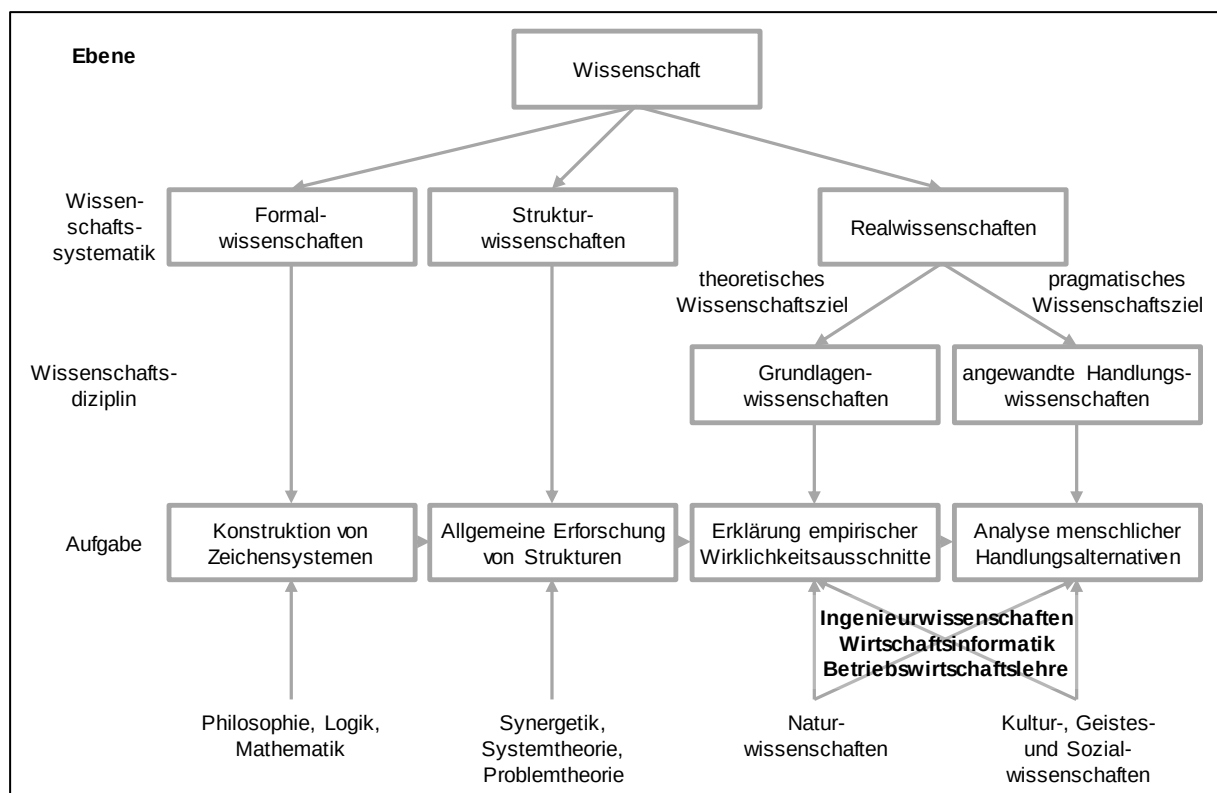


Abbildung 1-3: Wissenschaftssystematik (i. A. a. ULRICH U. HILL 1976a, S. 305; WILDE U. HESS 2007, S. 280; ZELEWSKI 2008; DEINDL 2013b, S. 6)

Die **Formalwissenschaften** beschäftigen sich mit der Konstruktion von wissenschaftlichen Sprachsystemen, von Sprachen und Zeichen, wie zum Beispiel die Mathematik,

² Formal- und Strukturwissenschaften werden teilweise auch als Idealwissenschaften zusammengefasst (s. JUNG 2016, S. 22).

und verfolgen dabei eine logische Vorgehensweise (s. ULRICH U. HILL 1976a, S. 305). **Strukturwissenschaften** befassen sich mit Strukturen von realen und formalen Objekten wie beispielsweise die Systemtheorie oder die Synergetik (s. ZELEWSKI 2008, S. 3). Die **Realwissenschaften** untersuchen dagegen real existierende Objekte und Vorkommen, wie es zum Beispiel die Naturwissenschaften, aber auch Geistes- und Sozialwissenschaften sowie die Wirtschaftswissenschaften tun (s. JUNG 2016, S. 22).

Die Realwissenschaften können weiter in **Grundlagenwissenschaften und angewandte Handlungswissenschaften** unterteilt werden. Bei den Grundlagenwissenschaften steht das Ziel der Erklärung von real existierenden Gesetzmäßigkeiten im Vordergrund. Hier wird grundsätzlich mithilfe der nomographischen Methode versucht, allgemeingültige Hypothesen zu identifizieren und diese empirisch gegen die Realität zu prüfen (s. ZELEWSKI 2008, S. 5). Dagegen nutzen die angewandten Handlungswissenschaften die ideographische Methode, mit deren Hilfe sie das Wesen des Untersuchungsobjekts und damit menschliche Handlungsalternativen untersuchen, verstehen und prognostizieren (s. ZELEWSKI 2008, S. 5; ULRICH U. HILL 1976a, S. 305; HILL ET AL. 1994, S. 34). Dabei verfolgen sie ein **eher pragmatisches Wissenschaftsziel, die Anwendbarkeit der Modelle steht vor der Allgemeingültigkeit von Theorien** (s. ULRICH 1981, S. 7; 1984, S. 174; LEHNER ET AL. 2008, S. 20; ÖSTERLE ET AL. 2010, S. 665). Sie beschäftigen sich daher mit der Gestaltung künftiger Realitäten, während sich die **Grundlagenwissenschaften mit der Erklärung existierender Realitäten befassen** (s. ULRICH 1984, S. 179). Eine strikte Unterscheidung zwischen den Letzgenannten ist jedoch in der Praxis nicht zielführend und auch nur bedingt möglich. So bedient sich zum Beispiel die Betriebswirtschaftslehre, die den angewandten Handlungswissenschaften zuzuordnen ist, sowohl nomographischer als auch ideographischer Methoden (s. ZELEWSKI 2008, S. 5).

Die vorliegende **Arbeit befasst sich mit Fragestellungen der Ingenieurwissenschaften, der Betriebswirtschaftslehre und der Wirtschaftsinformatik**. Ingenieurwissenschaften sind wegen der Untersuchung technisch orientierter Objekte genauso den angewandten Handlungswissenschaften zuzuordnen wie die Betriebswirtschaftslehre, die Modelle bildet, um anwendungsorientierte Handlungsempfehlungen zu geben (s. ULRICH U. HILL 1976a, S. 305; WÖHE ET AL. 2016, S. 12). Die Wirtschaftsinformatik ist eine Schnittstellendisziplin aus Betriebswirtschaftslehre und Informatik, die sich aber inzwischen als selbständige Disziplin entwickelt hat (s. LEIMEISTER 2015, S. VII). Letztere hat zwar einen strukturwissenschaftlichen Charakter, kann aber wegen ihres Anwendungsbezugs auf ein sächliches Objektsystem auch als Ingenieurwissenschaft eingeordnet werden (s. FINK ET AL. 2005, S. 256). Die in der vorliegenden Arbeit behandelten Themen der IT-Komplexität und der Implementierung von software-definierten Plattformen sind aus der Praxis hergeleitet und es wird die künftige Gestaltung von Informationssystem-Architekturen erforscht. Damit und mit der Einordnung der grundsätzlichen Fragestellungen kann die vorliegende **Arbeit den angewandten Handlungswissenschaften zugeordnet** werden.

Im Allgemeinen sind bei Realwissenschaften die **Grundprobleme der Subjektivität und der Kommunikation** zu lösen. Ersteres beschreibt Probleme der persönlichen

Wahrnehmung des Autors durch eigene Erfahrung und gewohnte Interpretation, letzteres Probleme der präzisen Sprache, die bei einer abstrakten, aber doch praxisnahen Arbeit einen hohen Einfluss auf die Forschungsergebnisse haben können (s. ULRICH U. HILL 1976a, S. 306). Um diesen Problemen zu begegnen, kann auf Basis dieser Einordnung der vorliegenden Arbeit der dafür passende Forschungsprozess abgeleitet werden (s. Abbildung 1-4).

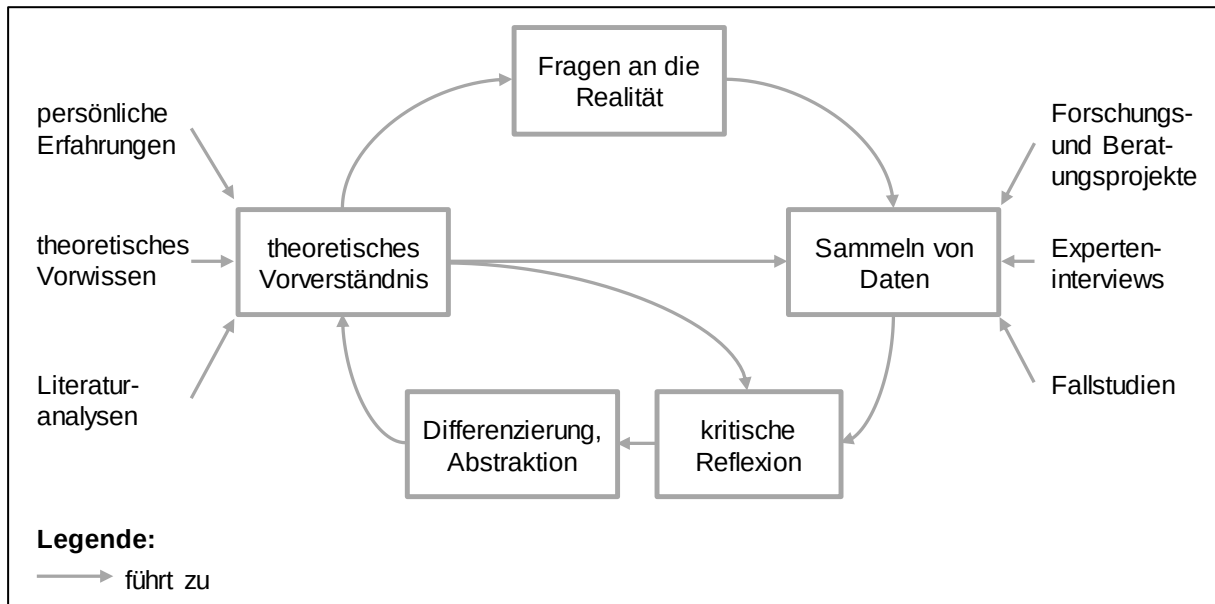


Abbildung 1-4: Forschungskonzeption (i. A. a. RHENSIUS 2010, S. 9; FRIEDLI 2000, S. 10; KUBICEK 1977, S. 14-15)

Hinter dem Forschungsprozess liegt das **Prinzip des wachsenden Verständnisses der Realität von KUBICEK**. Dabei werden im Sinne eines explorativen Vorgehens auf Basis des vorhandenen Vorverständnisses im Forschungsfeld in Verbindung mit Erfahrungen von Experten und Praktikern zielgerichtet auf Basis von eigenen Hypothesen Fragen gestellt und neue Erkenntnisse generiert (s. EISEND U. KUß 2017, S. 12-13; KUBICEK 1977, S. 14-15). Der Umweg der Prüfung beliebiger Hypothesen wird dadurch vermieden. Zur Bearbeitung komplexer Probleme bei geringem aktuellem Erkenntnisstand, für die zukünftige Lösungen gesucht werden, ist dieses Vorgehen nach KUBICEK die einzige Möglichkeit. Es entspricht den Prinzipien des **kritischen Rationalismus nach POPPER**. Dieser lehnt die Begründung von Hypothesen durch vollständige Induktion ab und besagt, dass Falsifikation die einzige Möglichkeit zur Widerlegung von Hypothesen sei (s. POPPER 1935, S. 1-4; ULRICH U. HILL 1976b, S. 345). Wichtiger als das Induktionskriterium ist für eine praxis- und gestaltungsorientierte Arbeit ohnehin das **Kriterium der Realisierbarkeit**. Dieses wird in dieser Arbeit durch Umsetzung der Ansätze der Aktionsforschung (engl. *Action Research*) erreicht, indem die entwickelten Modelle in realen Anwendungsfällen überprüft und weiterentwickelt werden (s. MCNIFF U. WHITEHEAD 2011, S. 9; MCNIFF 2017, S. 5-7). Dieses Verfahren ist heute unter anderem im Bereich der Wirtschaftsinformatik anerkannt und seine Anwendung weitet sich auch auf andere Wissenschaften aus (s. FRANK ET AL. 1998, S. 71ff; MOMMSEN U. PORTMANN 2017, S. 14-16). Insofern entspricht das Vorgehen, wie in Abbildung 1-4

dargestellt, einem iterativen, anwendungsorientierten Forschungsprozess (vgl. HESS ET AL. 2014, S. 130-131).

Um dem **Problem der Subjektivität** innerhalb des beschriebenen Forschungsprozesses entgegenzutreten, ist gemäß der Empfehlung von MAYRING das **Vorverständnis des Autors** entscheidend (s. MAYRING 2002, S. 25). Dieses ist maßgeblich durch dessen berufliche Laufbahn geprägt. Durch Mitarbeit und Leitung von industriellen Beratungsprojekten bei einer IT-Beratungsgesellschaft im Automotive-Umfeld sowie am Forschungsinstitut für Rationalisierung e. V. an der RWTH Aachen (FIR) konnte der Autor umfangreiche Kenntnisse über Praxisprobleme im praktischen Informationsmanagement von sowohl kleinen und mittleren (KMU) als auch Großunternehmen sammeln. Am FIR wurde diese Arbeit durch Forschungsprojekte im Bereich des Informationsmanagements, speziell im Kontext des IT-Komplexitätsmanagements, der IT-Serviceprozesse und der IT-Strategie, flankiert und ergänzt. Es ist damit sichergestellt, dass der Autor eine ausreichende Nähe zum Gegenstand seiner Forschung hat (s. MAYRING 2002, S. 50). Weiterhin wird die Intersubjektivität durch **vielfache Diskussionen der praktischen Herausforderungen, entwickelten Methoden und Lösungsansätze** im Rahmen von Forschungsprojekten, Treffen mit Verbänden, wie dem VDMA, Vorträgen und Diskussionen auf verschiedenen Foren wie der CeBIT, sowie im Zuge der durchgeführten formalen Evaluierung (s. Kapitel 9) sichergestellt (s. MAYRING 2002, S. 147).

Das genannte **Problem der Kommunikation** wird im Rahmen dieser Arbeit primär durch die im folgenden zweiten Kapitel vorgenommenen **umfangreichen Definitionen** angegangen. Präzise Definitionen helfen, die Wahrheitsnähe von theoretischen Aussagen sicherzustellen (s. EISEND U. KUß 2017, S. 14). Hier wird, soweit wie möglich, auf etablierte Definitionen im Kontext des Informationsmanagements zurückgegriffen, und diese, wo nötig, um praxisnahe Begriffe ergänzt.

Die Aufgaben, die sich durch den skizzierten Forschungsprozess ergeben, lassen sich nach ULRICH U. HILL in terminologisch-deskriptive, empirisch-induktive und analytisch-deduktive Aufgaben unterteilen und folgen damit einer operationsanalytischen Forschungskonzeption (s. ULRICH U. HILL 1976b, S. 347; HILL ET AL. 1994, S. 38). Dabei bedient sich das Vorgehen an Methoden aus angrenzenden Wissenschaftsdisziplinen im Sinne der in Abbildung 1-3 vorgestellten Abfolge. Im ersten Aufgabenbereich sind als die **terminologisch-deskriptiven** Aufgaben solche zu fassen, die Begriffe und deren Anwendung definieren, sowie solche, die Eingrenzungen des Forschungsbereichs auf relevante Dimensionen und Bildung von Typen vornehmen. Aufgaben im **empirisch-induktiven** Bereich sind die Ableitung von Hypothesen aus beobachteten Zusammenhängen und die empirische Überprüfung dieser Hypothesen. Eine zentrale Stellung nehmen sodann die **analytisch-deduktiven** Aufgaben ein, unter die logische Aufgaben wie die Modellbildung und die Ableitung von situationsspezifischen Handlungsalternativen fallen (s. ULRICH U. HILL 1976b, S. 348).

In die Forschungskonzeption, die dieser Arbeit zugrunde liegt, werden zusätzlich zu den genannten Methoden, wie von DÜNNEBACKE vorgeschlagen, auch Methoden der

Sozialforschung ergänzt. Dies sind Desk-Research, also die strukturierte Literaturanalyse vorhandener Quellen zum Themengebiet dieser Arbeit, sowie Interviews, Gespräche und Diskussionen mit Experten zur ständigen Überprüfung und Entwicklung des Vorgehens (s. DÜNNEBACKE 2016, S. 13).

In der Dissertationsschrift wird der Forschungsprozess quasi eingefroren. Sie ist in zehn Kapitel eingeteilt, folgt den Phasen der angewandten Forschung nach ULRICH und ist in Abbildung 1-5 dargestellt (vgl. ULRICH 1981, S. 20).

1.4 Aufbau der Arbeit

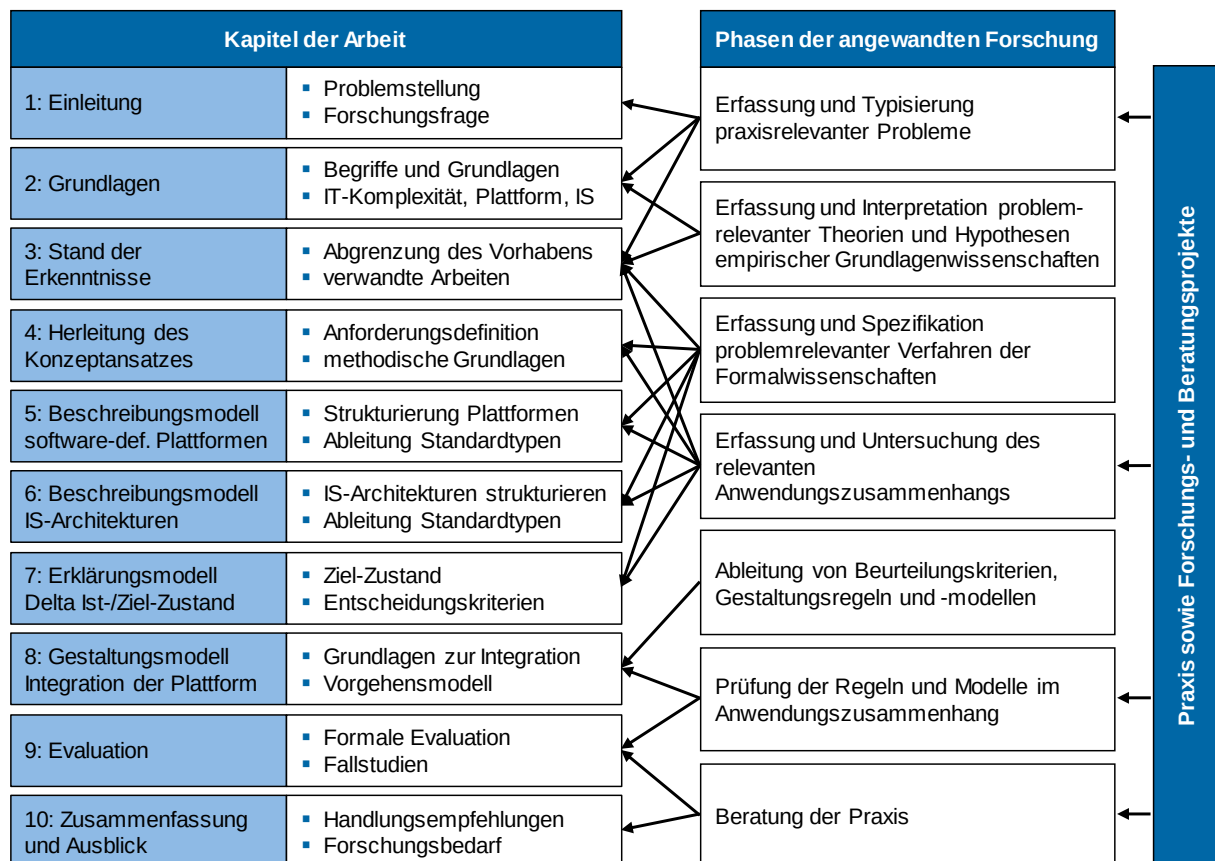


Abbildung 1-5: Phasen der angewandten Forschung nach ULRICH (eigene Darstellung i. A. a. SCHMIDT 2008, S. 7)

Nach der vorliegenden Einleitung beginnt die Arbeit mit der Beschreibung relevanter Grundlagen zum Verständnis für den Leser und die Erfassung des Untersuchungsbereichs in **Kapitel 2**. Dazu gehört die Definition der zentralen Begriffe sowie die Abgrenzung der Arbeit. **Kapitel 3** beschreibt den Forschungsbedarf empirisch-induktiv abgeleitet aus der Untersuchung des aktuellen Stands der Erkenntnisse in deutschsprachiger und internationaler Literatur sowie der Praxis. Es zeigt außerdem Arbeiten anderer Autoren auf, die in der vorliegenden Arbeit genutzt und weiterentwickelt werden. Im darauffolgenden **Kapitel 4** wird die methodische Grundlage für die entwickelten Modelle gelegt. Es werden Anforderungen an die Arbeit definiert und das methodische Vorgehen erläutert. Kern der vorliegenden Arbeit sind zwei Beschreibungsmodelle, ein

Erklärungs- und ein Gestaltungsmodell, die unter Nutzung von empirisch-induktiver und analytisch-deduktiver Schritte entwickelt werden.

Das Beschreibungsmodell in **Kapitel 5** präzisiert zuerst den Begriff der software-definierten Plattformen und leitet dann Merkmale solcher Plattformen ab. Die Herleitung erfolgt durch Abstraktion von betrachteten Plattformen in der Praxis. Es erfolgt eine Typisierung der software-definierten Plattformen auf Basis ihres Anwendungsumfangs. Zur Integration der einzelnen Plattfortmtypen werden deduktiv Integrationsanforderungen abgeleitet. Ein weiteres Beschreibungsmodell in **Kapitel 6** dient der Beschreibung von Informationssystem-Architekturen. Dort wird zuerst eine morphologische Beschreibung von Informationssystemen anhand von integrationsrelevanten Beschreibungsmerkmalen vorgestellt und darauf aufbauend werden Informationssystem-Architekturen und deren Visualisierung dargestellt. Unter Nutzung empirisch-induktiver Untersuchungen werden sodann Typen von Informationssystem-Architekturen entwickelt und beschrieben. Diese werden im Erklärungsmodell in **Kapitel 7** genutzt, um den Unterschied zwischen einem dort definierten Ziel-Zustand und den Architekturtypen zu erklären und Entscheidungskriterien für Unternehmen vorzustellen. **Kapitel 8** beschreibt die Anwendung der Modelle zur Gestaltung der Informationssystem-Architekturen produzierender Unternehmen zur Integration von software-definierten Plattformen. In **Kapitel 9** wird dann eine eingeschränkt empirisch-induktive Verifikation der Ergebnisse anhand einer exemplarischen Anwendung der Modelle in Fallstudien vorgestellt.

Zuletzt schließt **Kapitel 10** die Arbeit mit einer Zusammenfassung der Ergebnisse und einem Ausblick auf künftig relevante Forschungsthemen.

2 Grundlagen und Abgrenzung der Untersuchung

Zur Einordnung der Arbeit und zur Nachvollziehbarkeit des Verständnisses des Autors bedarf es an dieser Stelle einer Einordnung und Definition der grundlegenden Begriffe, die im Rahmen dieser Arbeit relevant sind.

2.1 Informationssysteme und -architekturen

Das Management von Informationssystemen ist zentraler Aspekt dieser Arbeit. Daher soll im Folgenden zuerst auf den Begriff der **Informationen** und **Daten** eingegangen werden (s. Kapitel 2.1.1), welche von **Informationssystemen** gehalten, bearbeitet, verteilt und gelöscht werden (s. Kapitel 2.1.2). Diese Informationssysteme haben **Schnittstellen** zu anderen Systemen (s. Kapitel 2.1.4) und decken unterschiedliche technische Ebenen ab. Daher sind diese **Schichten** (s. Kapitel 2.1.3) zu beschreiben, in die Informationssysteme eingeordnet werden können. Die Summe der so geordneten Informationssysteme in Unternehmen bilden die **Informationssystem-Architektur** (s. Kapitel 2.1.5), die in der vorliegenden Arbeit im Speziellen bei **produzierenden Unternehmen** untersucht wird (s. Kapitel 2.1.6).

2.1.1 Informationen im Kontext von Daten und Wissen

Für die Begriffe Information und Daten besteht eine Vielzahl von Definitionen und Begriffen, die teilweise synonym verwendet werden. Eine Unterscheidung von Daten und Informationen ist jedoch nötig, um die Diskussion zu strukturieren. MALIK beschreibt diese Notwendigkeit im betrieblichen Kontext folgendermaßen:

„An Daten fehlt es heute kaum in einem Unternehmen. Wir haben eher zu viel davon. Information hingegen ist noch immer Mangelware, und man kann sich nicht darauf verlassen, dass alle Manager wissen, wie man von Daten zu Informationen kommt.“ (MALIK 2006, S. 337)

Umgangssprachlich wird unter Information Auskunft, Aufklärung oder Belehrung verstanden (s. HEINRICH ET AL. 2014, S. 14). Im wissenschaftlichen Kontext unterscheiden sich Begriffe und Definitionen vor allem hinsichtlich ihrer Herkunft und der Wissenschaftsdisziplin, in der sie Verwendung finden. Die **Nachrichtentheorie** (vgl. SHANNON U. WEAVER 1976) versteht unter Informationen die Wahrscheinlichkeit des Auftretens von Zeichen in Nachrichten. **ISO/IEC 2382** definiert Information als Wissen, das Unsicherheit über das Auftreten eines bestimmten Ereignisses aus einer Reihe möglicher Ereignisse reduziert oder beseitigt (s. ISO/IEC 2382). Beide Definitionen vermeiden eine klare Erklärung des Begriffs; im Zusammenhang dieser Arbeit sind diese daher als nachrangig zu betrachten.

Grundlegend ist dagegen die **Semiotik**, die als Zeichen- und Bedeutungslehre vier Ebenen zur Differenzierung des Informationsbegriffs nutzt (s. Abbildung 2-1). Die Syntaktik beschäftigt sich mit den Beziehungen zwischen Zeichen als Informationsträger eines Sprachsystems und legt Regeln für ebendiese fest. Abbildungsfunktionen von

Gegenständen durch die Verwendung von Zeichen wird durch die Sigmatik untersucht. Die Semantik betrachtet Beziehungen zwischen Zeichen und ihrer inhaltlichen Bedeutung, wohingegen die Pragmatik die Verbindung zwischen Zeichen und Zeichenverwendung untersucht.

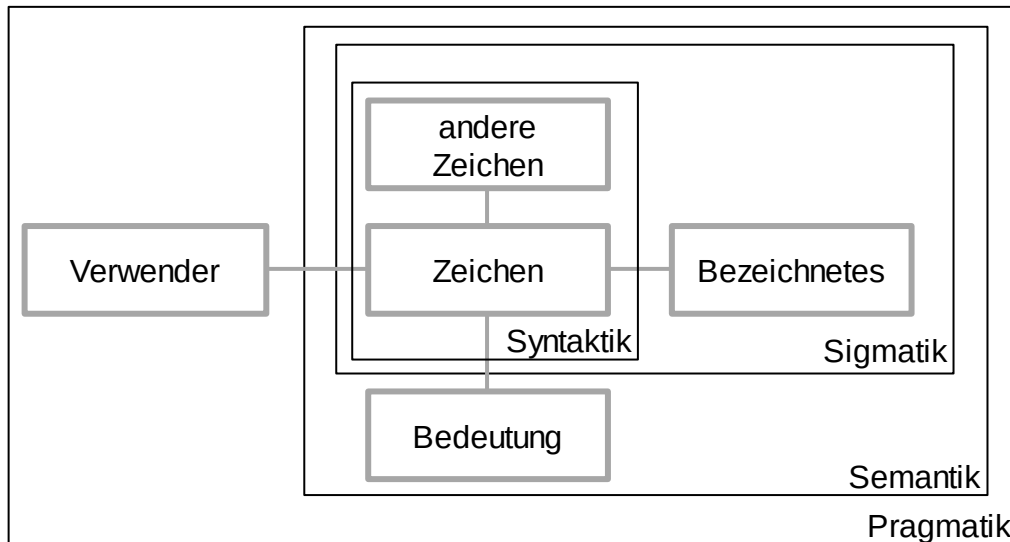


Abbildung 2-1: Ebenen der Semiotik (i. A. a. BERTHEL 1975, S. 1869)

Aus **betriebswirtschaftlicher Perspektive** sind Informationen als zweckorientiertes Wissen zur Grundlage für die Vorbereitung betrieblicher Entscheidungen zu verstehen (s. WITTMANN 1959, S. 14; SCHULTE-ZURHAUSEN 2014, S. 70). In diesem Kontext haben sich Informationen auch als eigener Produktionsfaktor neben den klassischen Elementarfaktoren und dispositiven Faktoren nach GUTENBERG (vgl. GUTENBERG 1979; WÖHE ET AL. 2016, S. 16) entwickelt. Sie unterscheiden sich jedoch von den klassischen Wirtschaftsgütern dadurch, dass sie mehrfach genutzt und dabei nicht verbraucht werden. Zudem hängt deren Wert sehr stark von deren Nutzung ab, aber auch von Qualität, Aggregation und Umfang. Zum Verständnis von Information bei Sender und Empfänger sind Standards notwendig, nach denen diese kodiert beziehungsweise dekodiert werden (s. SCHULTE-ZURHAUSEN 2014, S. 70-71; KRCMAR 2015, S. 16). In Unternehmen bergen Informationen also einen hohen Wert; ihre Nutzung hat in den vergangenen Jahren kontinuierlich zugenommen (s. ERWIN ET AL., S. 20).

In der **Wirtschaftsinformatik** versteht man Informationen als „handlungsbestimmendes Wissen über vergangene, gegenwärtige und zukünftige Zustände der Wirklichkeit und Vorgänge in der Wirklichkeit“ (HEINRICH ET AL. 2011, S. 12). Informationen werden damit als zweckbezogenes Wissen verstanden, welches den Kenntnisstand des Informationsnutzers verbessert und zur Vorbereitung von Handlungen genutzt wird. (s. KRCMAR 2015, S. 15)

Die folgende Tabelle fasst zusammen, dass in den verschiedenen Perspektiven und Disziplinen also sehr unterschiedliche Bedeutungen von Information bestehen. Teilweise stehen diese im Kontext von Daten und Zeichen, teilweise auch von Wissen.

Tabelle 2-1: Verschiedene Bedeutung von Informationen (s. KRCMAR 2015, S. 14; i. A. a. RAUTERBERG 1989, S. 220)

1. Information als Nachricht (Syntax)
2. Information als Bedeutung der Nachricht (Semantik)
3. Information als Wirkung der Nachricht (Pragmatik)
4. Information als Prozess
5. Information als Wissen
6. Information als Bestandteil der Weltstruktur

Zur **Beschreibung des Zusammenhangs zwischen Zeichen, Daten, Informationen und Wissen** hat sich im Kontext des Informationsmanagements die Definition nach KRCMAR und REHÄUSER U. KRCMAR sowie LEIMEISTER durchgesetzt. Demnach besteht die unterste Ebene aus einem Vorrat verschiedener Zeichen. Eine regelbasierte Kombination dieser Zeichen (Syntax) lässt Daten entstehen. Erfolgt eine **Interpretation dieser Daten**, also eine Anreicherung der Daten mit Kontext, werden diese zu **Informationen**. Entgegen der oben aufgeführten, betriebswirtschaftlichen beziehungsweise wirtschaftsinformatischen Definition des Informationsbegriffs entsteht gemäß dieser Logik Wissen erst durch eine Vernetzung und einen Kontextbezug der Informationen (s. REHÄUSER U. KRCMAR 1996, S. 5; LEIMEISTER 2015, S. 24). Dieses Verständnis ist in Abbildung 2-2 zusammengefasst und soll anhand eines Beispiels erläutert werden.

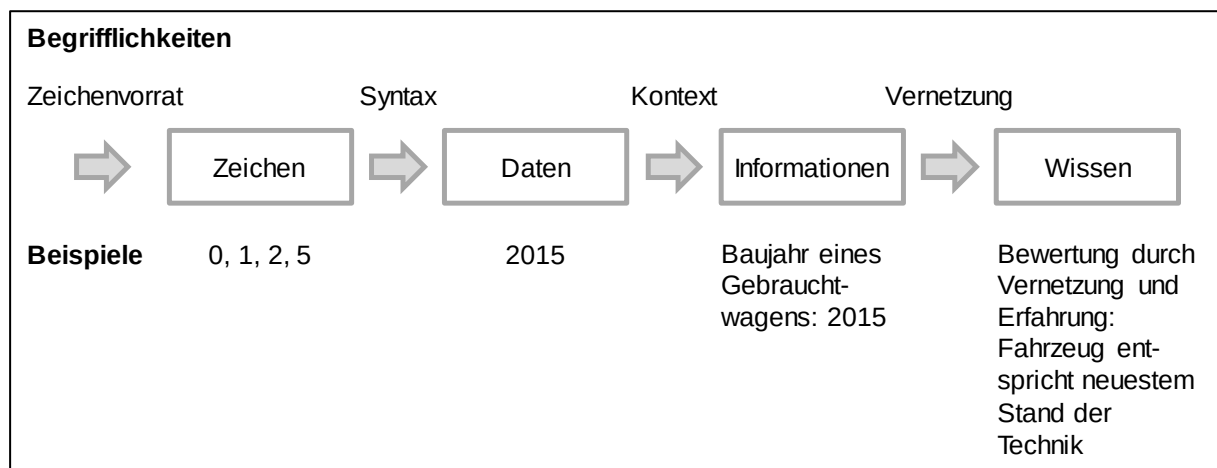


Abbildung 2-2: Begriffshierarchie von Zeichen, Daten, Informationen und Wissen (i. A. a. KRCMAR 2015; S. 12, ACKOFF 1989)

Die erste Ebene beschreibt einen großen Vorrat verschiedener Zeichen und dient als Basis der weiter oben angesiedelten Begriffe. Das Beispiel aus Abbildung 2-2 bietet als Zeichen „0“, „1“, „2“ und „5“ an. Durch das Ordnen der Zeichen in einem regelbasierten Zusammenhang (Syntax) entstehen Daten, die Zahl 2015. Informationen als nächst-höhere Ebene entstehen durch eine Anreicherung dieser Daten mit zusätzlichem Kontext, den Daten wird also eine Bedeutung verliehen. Im Beispiel wird durch die Zahl 2015 das Baujahr eines Gebrauchtwagens ausgedrückt. Durch Vernetzung

von Informationen mit weiteren Informationen sowie Erfahrungen entsteht auf der letzten Ebene der Hierarchie Wissen. Im aufgeführten Beispiel wird durch die Vernetzung der Information „Baujahr 2015“ Wissen über Eigenschaften des Gebrauchtwagens erlangt, nämlich, dass dieser aufgrund seines jungen Alters dem neuesten Stand der Technik entspricht.

Entsprechend HEINRICH ET AL. ist es nicht Aufgabe dieser Dissertation, neue Begriffskonstrukte in der Wirtschaftsinformatik zu erzwingen, weshalb die **gängigen Begriffe verwendet werden** sollen und das Verständnis aus Abbildung 2-2 in dieser Arbeit zugrunde gelegt werden soll (s. HEINRICH ET AL. 2014, S. 15; KRCMAR 2015, S. 16).

Im eben definierten Verständnis soll auch der im Folgenden relevante Begriff der „smarten“ Daten (engl. *Smart Data*) definiert werden. **Smarte Daten** sind solche, die eine sinnvolle semantische Beschreibung beinhalten. Diese entsteht, weil sie vor der Speicherung einem Prozess unterzogen werden, in dem sie um Metadaten angereichert werden, damit kategorisiert und somit um Fachwissen ergänzt sind (s. FREY 2015, S. 30-31). Dies hat den Vorteil, dass die Menge an zu speichernden Datenpaketen verringert und gleichzeitig die Datenanalyse beziehungsweise -auswertung zur Generierung von Informationen beschleunigt werden kann (s. Abbildung 2-3). In diesem Zusammenhang werden Daten zukünftig auch ein entscheidendes Rohmaterial darstellen (s. SCHUH ET AL. 2017g, S. 11)



Abbildung 2-3: Smart-Data-Ansatz zur Generierung von Information (i. A. a. NIENKE U. FUHS 2017, S. 23)

2.1.2 Informationssysteme

Abgeleitet vom Begriff der Informationen im vorherigen Kapitel soll hier der **Begriff des Informationssystems** festgelegt werden. Wie bei der Information selbst gibt es auch hier eine Vielzahl von Definitionen und synonym verwendeten Begriffen. So wird **in der Praxis, teilweise auch in der Wissenschaft, oft vom IT-System gesprochen**. Da der Begriff „IT“ jedoch, entgegen seiner ursprünglich technologischen Bedeutung der reinen Informationstechnologie, eine Vielzahl von neuen Verwendungen erfahren hat, die auch personelle („IT-Leiter“) oder organisatorische („IT-Abteilung“) Komponenten enthalten, soll von seiner Verwendung in dieser Arbeit möglichst abgesehen werden (s. LEHNER ET AL. 2008, S. 42; HEINRICH ET AL. 2014, S. 14-15). Der stattdessen **in der Wirtschaftsinformatik etablierte Begriff des Informationssystems** besteht aus zwei Komponenten: der *Information*, wie sie im vorherigen Kapitel beschrieben wurde, und dem *System*.

Ein **System** wird generell als ganzheitlicher Zusammenhang von Objekten verstanden, die zueinander in Abhängigkeit stehen, von der Umwelt definitorisch oder real abgegrenzt und über Beziehungen beziehungsweise Schnittstellen über die Systemgrenze

hinweg mit anderen Systemen verbunden sind (s. HEINRICH ET AL. 2014, S. 14; HABERFELLNER ET AL. 2015, S. 31ff). Gemäß Abbildung 2-4 bestehen Systeme aus Elementen, die als Untersysteme wiederum Systeme sein können. Außerhalb der Systemgrenze haben Systeme Schnittstellen und Beziehungen zu Umsystemen, die sich im Umfeld des Systems befinden. Mit dieser Logik können Systeme in einer optimalen Welt strukturiert und hierarchisiert werden. In der realen Welt müssen Systeme jedoch **oft auch als sogenannte Blackbox oder Greybox betrachtet** werden, da zwar Systemgrenzen bekannt, die enthaltenen Elemente jedoch (teilweise) unbekannt sind. Ebenfalls relevant ist der *Aspekt*, unter dem ein System modelliert und dargestellt ist: Je nach Modellierungszweck werden unterschiedliche Elemente und Beziehungen in die Modellierung aufgenommen (s. HABERFELLNER ET AL. 2015, S. 32-39). Diese sich unterscheidenden Aspekte können unter anderem zu einer diffusen Wahrnehmung der tatsächlichen Situation führen (s. dazu Kapitel 2.2.2).

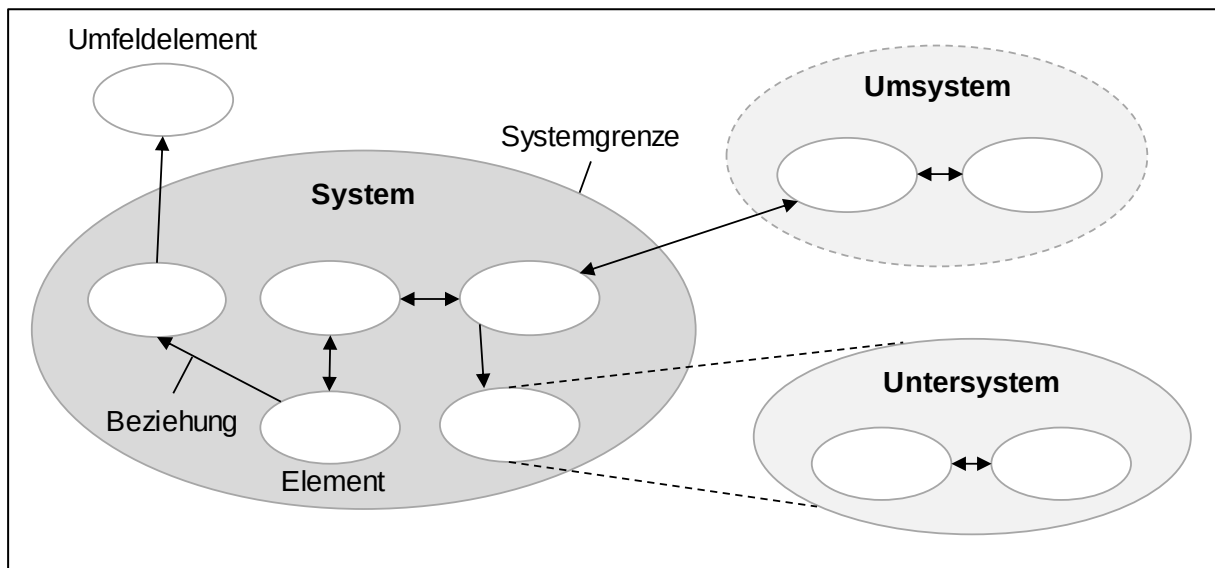


Abbildung 2-4: Zusammenhang von Systemen, Unter- und Umsystemen (i. A. a. HABERFELLNER ET AL. 2015, S. 32-34)

Die vorgenommene Systemdefinition kann auf Informationssysteme angewandt werden. Prinzipiell ist ein **Informationssystem Teil des umfassenden Systems Informationssystem-Architektur**, das wiederum Teil des Umsystems *Unternehmen* ist; gleichzeitig besteht es auch aus Teilsystemen. Dies widerspricht der Ansicht in der Organisationstheorie, in der klassisch von nur *einem* Informationssystem im Unternehmen ausgegangen wird. In der vorliegenden Arbeit soll vom Ansatz der Wirtschaftsinformatik ausgegangen werden, in der es verschiedene Informationssysteme je nach Aufgabe und Inhalt gibt (s. KRCMAR 2015, S. 23). Dazu können, abhängig vom Modellierungsaspekt, unterschiedliche **Grenzen** gezogen werden. In Abbildung 2-5 sind diese Aspekte zusammengefasst: Zuerst kann entsprechend der Beziehungsart zwischen den Elementen unterschieden werden. Informationssysteme stehen über die Beziehungsart „Information“ miteinander in Beziehung, alle anderen Elemente im Übersystem *Unternehmen* werden als Basissystem bezeichnet und sind über physische Beziehungsarten wie Material- oder Energieflüsse miteinander verbunden. Wei-

terhin kann über die Art der Aufgabenbearbeitung in automatisierte und nicht-automatisierte Systeme unterschieden werden. Zuletzt sind Informationssysteme in Lenkungssysteme, die zur Planung, Steuerung und Kontrolle von Um- oder Teilsystemen dienen, und Leistungssysteme, die Leistungen erstellen, zu unterscheiden. (S. LEIMEISTER 2015, S. 28)

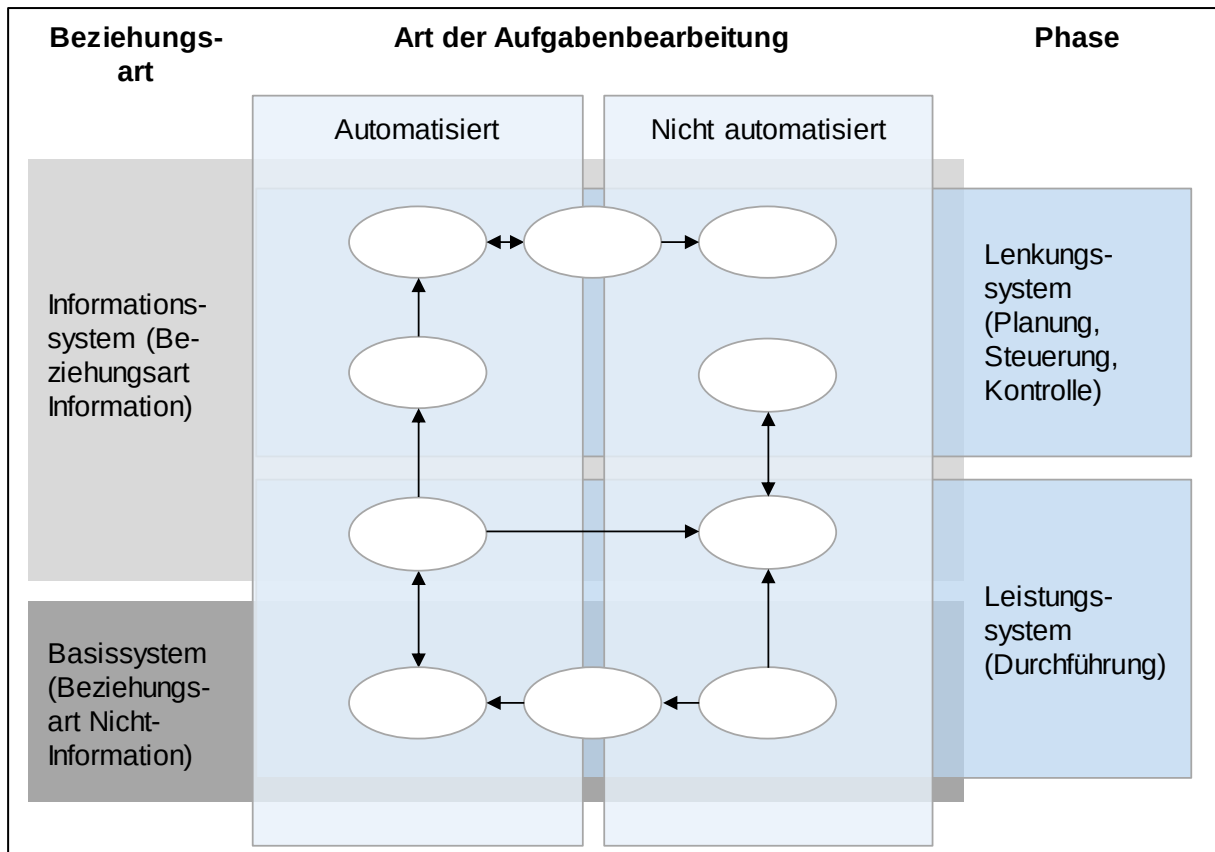


Abbildung 2-5: Einordnung von Informationssystemen (i. A. a. LEHNER ET AL. 2008, S. 42; LEIMEISTER 2015, S. 28)

In der Normung werden die Teilsysteme, aus denen Informationssysteme bestehen, aufgeschlüsselt in das informationsverarbeitende System sowie menschliche, technische und finanzielle Ressourcen, die als Teilsysteme die Informationen nutzen und verteilen (s. ISO/IEC 2382). Dementsprechend kann man Informationssysteme also als **soziotechnische Systeme, die menschliche und maschinelle Teilsysteme umfassen und zur Bereitstellung von Information und Kommunikation nach wirtschaftlichen Kriterien eingesetzt werden**, verstehen (s. KRCMAR 2015, S. 22). Sie bestehen aus dem Menschen, der Nutzer, aber auch Ersteller von Informationssystemen ist. Sie bestehen weiterhin aus der Maschine, die sich aus Hardware- und Softwarekomponenten zusammensetzen lässt. Die Softwarekomponenten wiederum können unterschieden werden in die Betriebsumgebung,³ mit welcher der Nutzer keine

³ Entgegen der Definition von KRCMAR wird in Anlehnung an ALPAR ET AL. in der vorliegenden Arbeit die Betriebsumgebung als separater Bestandteil von Informationssystemen gesehen (vgl. ALPAR ET AL. 2016, S. 423).

Interaktion durchführt, und die Anwendung selbst, mit welcher der Mensch oder ein anderes System arbeitet und in der Daten abgelegt sind sowie Prozesse umgesetzt werden (s. Abbildung 2-6). Aufbauend auf dieser Definition werden in Kapitel 6 die Merkmale von Informationssystemen vertieft.

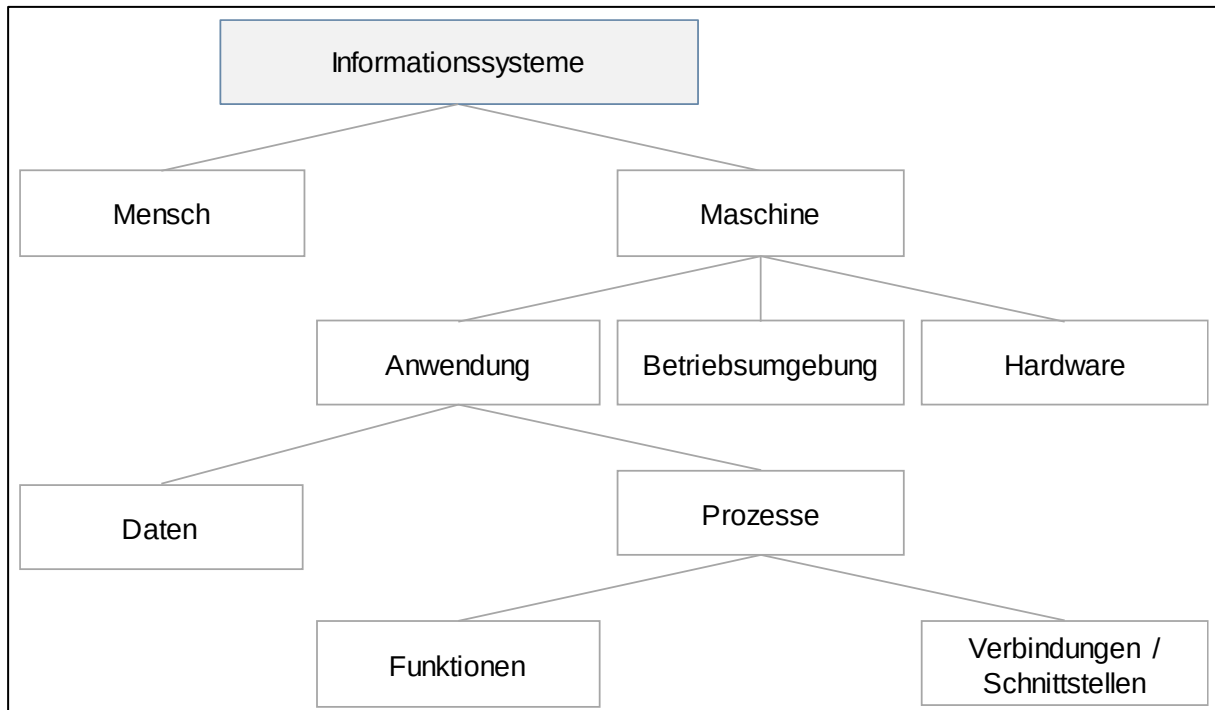


Abbildung 2-6: Informationssysteme als soziotechnische Systeme (eigene Darstellung i. A. a. KRCMAR 2015, S. 22)

Im Zuge dieser Definition werden **Anwendungssysteme nicht als Synonym für Informationssysteme** angesehen (vgl. LEHNER ET AL. 2008), sondern sind ein Subsystem von Informationssystemen (s. GABRIEL 2018). Dies ist durch den Ast „Anwendung“ in Abbildung 2-6 dargestellt.

2.1.3 Schichtenmodelle

Die **Elemente von Informationssystemen** aus Abbildung 2-6 können auf unterschiedliche technische und organisatorische **Ebenen verteilt** werden. Für diese Aufteilung gibt es mehrere etablierte Schichtenmodelle, die für die vorliegende Arbeit relevant sind. Diese Modelle beschreiben und legen fest, wie Kommunikation zwischen Elementen auf den jeweiligen Ebenen funktioniert, und bilden daher die Grundlage für Überlegungen zur Integration und zu Schnittstellen (s. Kapitel 2.1.4).

Das wichtigste Modell ist das **ISO-OSI-Referenzmodell**, das 1984 in der ISO/IEC 7498 (heute ISO/IEC 7498-1) genormt wurde. Es legt grundsätzlich den Aufbau von Kommunikation zwischen Informations- und Kommunikationstechnik fest. Bei der Entwicklung sind grundlegende Erfahrungen mit den am weitesten verbreiteten Übertragungsprotokollen eingeflossen, der Internetprotokollfamilie TCP/IP (*Transmission Control Protocol/Internet Protocol*). Abbildung 2-7 gibt einen Überblick über beide Modelle und deren Zusammenhang.

ISO/OSI-Schicht	TCP/IP-Protokollfamilie		
	Schicht	Protokollbeispiele	
7 Anwendungsschicht	Prozess / Applikation	FTP	(Dateiübertragung)
6 Darstellungsschicht		SMTP	(E-Mail)
5 Kommunikationssteuerungsschicht		HTTP	(World Wide Web)
4 Transportschicht	Host to Host	TCP	paketbasierte Datenübertragung
3 Vermittlungsschicht	Internet	IP	Internetprotokoll
2 Sicherungsschicht	Lokales Netz oder Netzzugriff	Ethernet, Token Ring, FDDI	
1 physische Schicht			

FDDI: Fiber Distributed Data Interface FTP: File Transfer Protocol HTTP: Hypertext Transfer Protocol
SMTP: Simple Mail Transfer Protocol

Abbildung 2-7: Schichteneinteilung bei ISO/OSI und TCP/IP (i. A. a. LEIMEISTER 2015, S. 102)

Die Schichten im ISO/OSI-Referenzmodell lassen sich zusammenfassen in die Transportschichten in den Schichten eins bis vier sowie Anwendungsschichten in den darüber liegenden Schichten fünf bis sieben. In dieser Ebene befindet sich die fünfte Schicht (Kommunikationssteuerung) auf Ebene des Betriebssystems in Form von Sitzungen (engl. *Sessions*), die sechste dient zur Decodierung von Daten in Sprache oder Grafik, während erst die letzte Schicht Anwendungsklassen definiert, mit denen der Endnutzer in Kontakt kommt. Die tatsächliche Datenübertragung findet dabei ausschließlich auf der ersten, der physischen, Schicht statt (s. ISO/IEC 7498-1; LEIMEISTER 2015, S. 100-102).

Tabelle 2-2: Architekturtypen und Schichten von Informationssystemen in der betrieblichen Anwendung (i. A. a. ALT u. PUSCHMANN 2016, S. 153)

Architekturtyp	1-Tier	2-Tier	3-Tier	n-Tier
Anzahl Schichten	Eine	Zwei	Drei	größer drei
Beispiel	Host-zentrierte Systeme	2-Tier Client-/Server	3-Tier Client-/Server	4-Tier Integrationsarchitektur
Programmelemente	Präsentation, Anwendungsfunktionalität, Daten sind in einem Informationssystem integriert	zusammengefasste Präsentations- und Anwendungsfunktionalität, separate Daten-Server-Ebene	Anwendungsfunktionalität, Daten und Präsentation sind als eigen Schicht umgesetzt	Anwendungsfunktionalität, Daten und Präsentation sind auf viele Schichten verteilt

Für Informationssysteme im Ganzen haben sich mehrschichtige Auftrennungen der Systeme nach bestimmten Prinzipien durchgesetzt. Während in der Vergangenheit noch hauptsächlich Host-zentrierte Systeme, welche Präsentation, Anwendung und Daten in einem monolithischen System zusammenfassten, im Einsatz waren, wird heute in zwei oder mehr Ebenen getrennt. In einer gängigen „3-Tier“-Architektur werden Systeme nach den Schichten *Präsentation*, *Anwendungsfunktionalität* und *Daten* aufgeteilt. Wenn übergreifend zwischen verschiedenen Informationssystemen gearbeitet wird, geht diese Aufteilung noch weiter und es werden Zwischenschichten eingeschoben, die als Middleware- oder EAI-Systeme bezeichnet werden (s. ALPAR ET AL. 2016, S. 170-172).

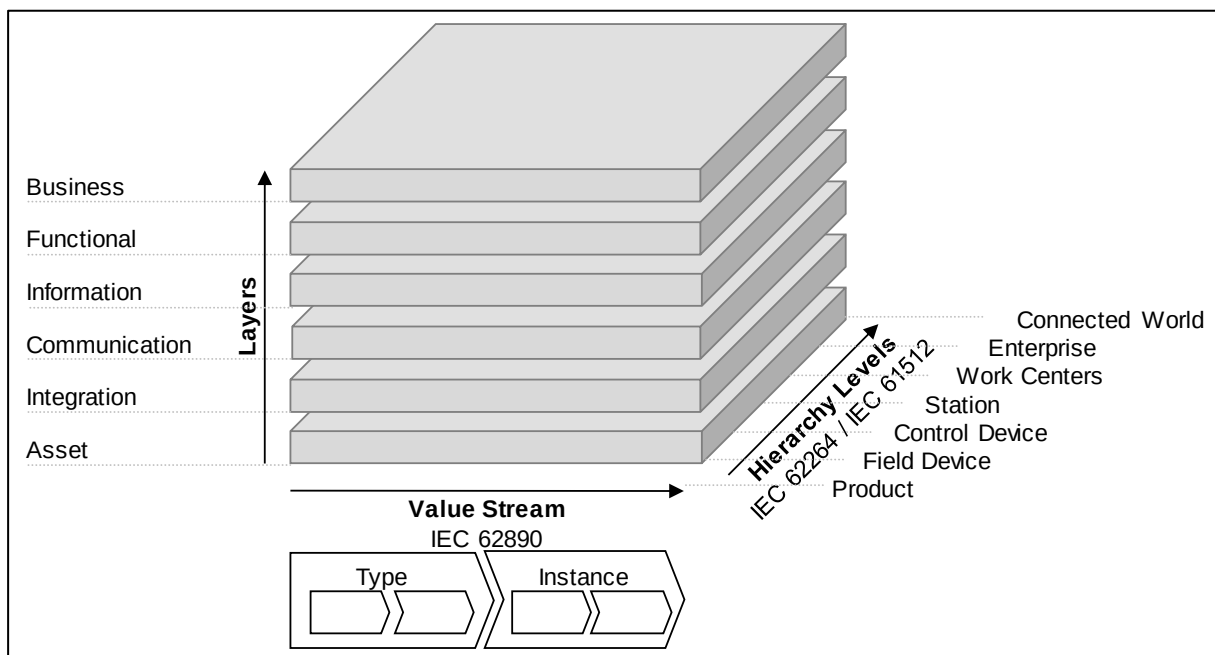


Abbildung 2-8: Referenzarchitekturmodell Industrie 4.0 (i. A. a. DIN SPEC 91345, S. 19)

Das **Referenzarchitekturmodell Industrie 4.0** (RAMI4.0) ist ein von der Plattform Industrie 4.0 zusammen mit dem Verband ZVEI entwickeltes, dreidimensionales Schichtenmodell zur besseren Handhabung der industriellen Komplexität (s. Abbildung 2-8). Inzwischen ist auch dieses Modell der Normung zugeführt worden und in der DIN SPEC 91345 spezifiziert. Es werden hierbei die Achsen „Layers“, „Hierarchy Levels“ sowie „Value Stream“ unterschieden (s. HANKEL 2015, S. 1-2; ADOLPHS ET AL., 2015, S. 8). Für die Informationssystem-Architektur sind vor allem die „Layers“ von besonderem Interesse. Hier werden die Informationsebenen „Asset“, „Integration“, „Communication“, „Information“, „Functional“ bis hin zum Layer „Business“ abgebildet (s. PLATTFORM INDUSTRIE 4.0, 2016, S. 10; BLEIDER ET AL. 2017, S. 27-31).

Mit der **Industrial Internet Reference Architecture** (IIRA) stellt das *Industrial Internet Consortium* (IIC) fast zeitgleich zur Plattform Industrie 4.0 eine Informationsarchitektur bereit, die durch unterschiedliche Sichtweisen versucht, eine Struktur zu etablieren, um vorhandene und zukünftige Komplexität zu reduzieren. Die IIRA unterscheidet dabei in die Sichtweisen „Business Viewpoint“, „Usage Viewpoint“, „Functional Viewpoint“

und „Implementation Viewpoint“ (s. HAAG 2015, S. 2-8). Aufbauend auf dem „Wie“ (*Implementation Viewpoint*) ergeben sich die Ebenen des „Was“ (*Usage & Functional Viewpoint*), woran sich weiterführend die „Warum“-Ebene anschließt. Es gelingt der IIRA, unterschiedliche Schwerpunkte auf Basis verschiedener Nutzer zu etablieren. Wie in Abbildung 2-9 gezeigt ist, wird ein modulares Modell mit fünf aufeinander aufbauenden funktionalen Bereichen vorgeschlagen, welche mit den physischen Systemen im Austausch stehen.

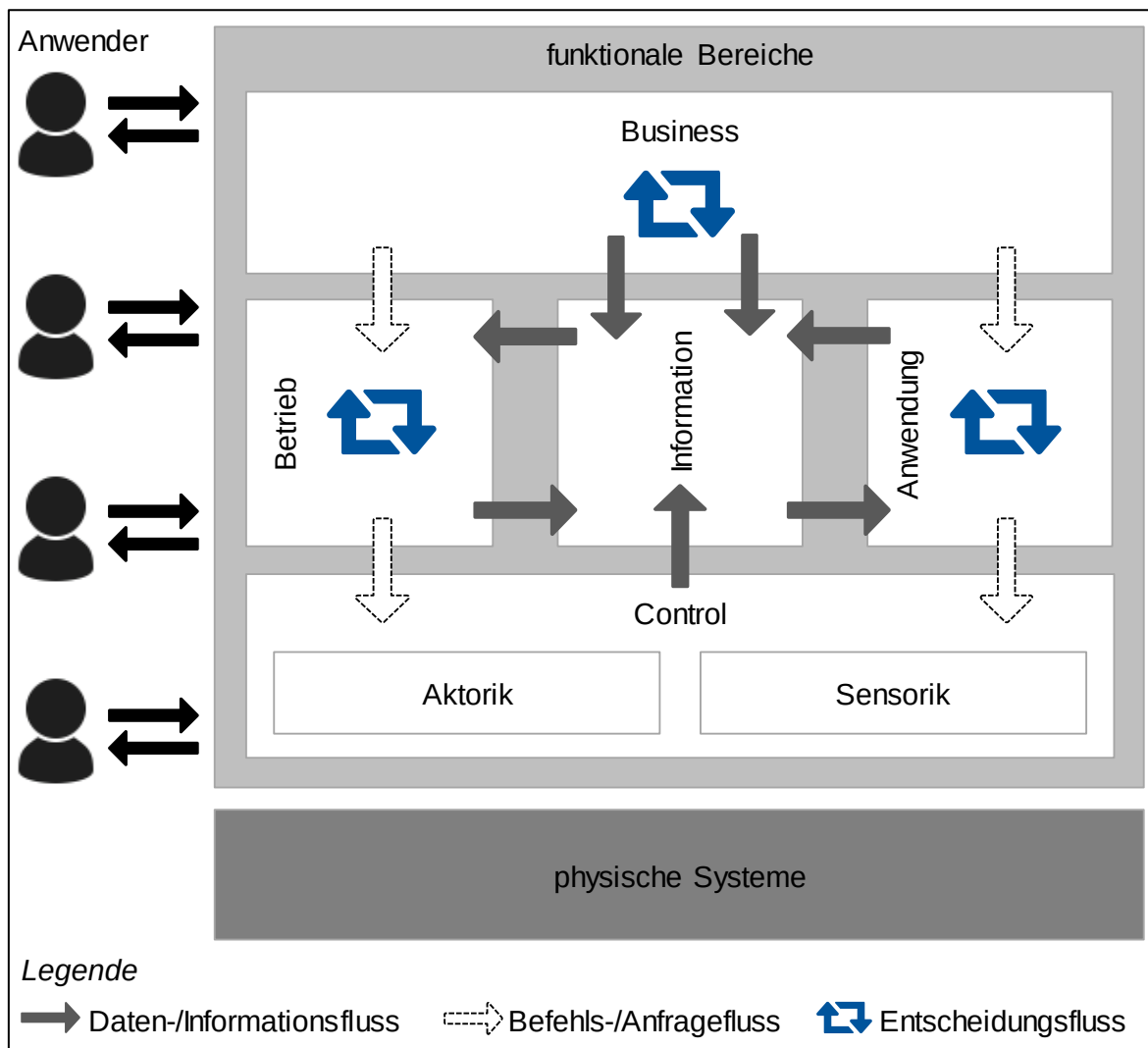


Abbildung 2-9: Modulare Architektur des IIC (i. A. a. INDUSTRIAL INTERNET CONSORTIUM 2017, S. 27)

Alle angesprochenen Modelle versuchen, sinnvolle Aggregationen der verschiedenen Schichten vorzunehmen. Je nach Betrachtung der Systeme können sehr unterschiedliche Elemente zu Schichten zusammengefasst werden. Deshalb wurde für diese Arbeit in Kapitel 6.5.1 ein eigenes Schichtenmodell entwickelt.

2.1.4 Integration und Schnittstellen

Nachdem **Informationssysteme in den 1960er Jahren** zuerst zur Automatisierung bestimmter Spezialfälle als **Insellösungen** eingeführt wurden, hat sich mit zunehmen-

dem Einzug von Informationstechnologie in Unternehmen die Notwendigkeit zur Integration der vorhandenen Systeme eingestellt (s. FERSTL U. SINZ 2013, S. 239). Für die Umsetzung der Industrie 4.0 ist die Integration von Informationssystemen zur **gemeinsamen Datennutzung entlang der Wertschöpfungskette eine wichtige Grundlage** (s. SCHUH ET AL. 2017a, S. 28). Unter Integration wird hierbei das Zusammenfügen von Einzelteilen zu einer Einheit unter Anwendung einer bestimmten Methode verstanden (s. KAIB 2002, S. 10); im hier relevanten Kontext also die Einheit von Informationssystemen gemäß der Definition in Kapitel 2.1.2. Dies kann in unterschiedlichen Formen wie der Daten-, Datenfluss-, Funktions- oder Prozessintegration stattfinden. Entsprechend reichen die technischen Konzepte hierzu von zentralen Datenbanken (Datenintegration), standardisierten Schnittstellen (Datenflussintegration) bis zu übergreifenden Anwendungen mit entsprechender Workflowsteuerung (s. LEHNER ET AL. 2008, S. 44-47; FERSTL U. SINZ 2013, S. 243-251). Ausführlicher unterscheidet DERN zwischen den folgenden vier Dimensionen (s. Abbildung 2-10): Die **Integrationsperspektive**, bei KAIB auch Integrationsrichtung genannt, klärt die grundsätzliche Sicht, die auf die Integration genommen werden muss. Hier wird auch zwischen horizontaler (über verschiedene Unternehmensbereiche, aber auf einer Ebene) und vertikaler Integration (innerhalb eines Bereichs wie zum Beispiel dem Vertrieb, aber über verschiedene Ebenen) unterschieden. Die Dimension der **Integrationsreichweite** definiert, ob inner- oder zwischenbetriebliche Integration betrachtet wird, und der **Integrationsgegenstand** legt die Elemente innerhalb eines Systems fest, die es zu integrieren gilt. Zuletzt beschreibt die Dimension des **Automatisierungsgrades**, inwieweit der Mensch bei der Integration eingreifen muss (s. KAIB 2002, S. 15; DERN 2011, S. 41).

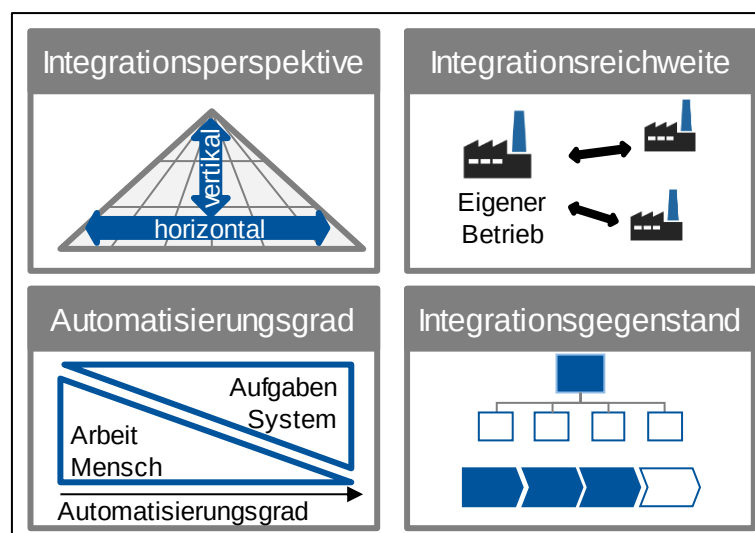


Abbildung 2-10: Dimensionen des Integrationsmanagements nach DERN (eigene Darstellung)

In diesem Kontext ist auch der Begriff des EAI, der sich in der Fachliteratur findet, einzuführen. EAI ist ein Konzept, mit dem die Integrationsziele der Redundanz, Datenverknüpfung, Konsistenz und Zielorientierung gleichermaßen verfolgt werden. Umgesetzt wird das Konzept über die Einführung von sogenannten Middleware-Systemen,

welche zwischen den verknüpften heterogenen Hard- und Softwarekomponenten vermitteln und übersetzen (s. LEHNER ET AL. 2008, S. 194f; FERSTL U. SINZ 2013, 260-264; vgl. KAIB 2002).

Für die Umsetzung eines EAI-Konzepts, aber auch für die Umsetzung der Industrie 4.0, sind Schnittstellen zwischen den Systemen von zentraler Bedeutung. Sie ermöglichen den direkten, medienbruchfreien vertikalen und horizontalen Datenaustausch zwischen Informationssystemen (s. DANILUK ET AL. 2016, S. 205; SCHUH ET AL. 2017a, S. 28). Die Literatur unterscheidet zwischen verschiedenen Typen von Schnittstellen, die sich auch im Unternehmensalltag finden. Abbildung 2-11 fasst diese zusammen: **Indirekte Schnittstellen** nutzen eine Dateiablage als Zwischenschritt (in der Praxis oft CSV-Dateien⁴), in dem die übertragenen Daten konvertiert werden. **Direkte Schnittstellen** ermöglichen dagegen eine direkte, medienbruchfreie Übertragung zwischen Informationssystemen.



Abbildung 2-11: Direkte und indirekte Schnittstellen (i. A. a. SINDERMAN 2014, S. 328f)

Weiterhin unterscheiden sich Schnittstellen durch ihren **Standardisierungsgrad** (s. Abbildung 2-12). Nicht-standardisierte Schnittstellen, die spezifisch von einem Softwarehersteller entwickelt sind und oft von anderen Systemen nicht direkt gelesen werden können, werden als proprietär bezeichnet. Standardisierte beziehungsweise neutrale Schnittstellen, die ein gemeinsames Datenmodell nutzen, bedürfen der Einigung von Softwareherstellern und werden oft der Normierung zugeführt.

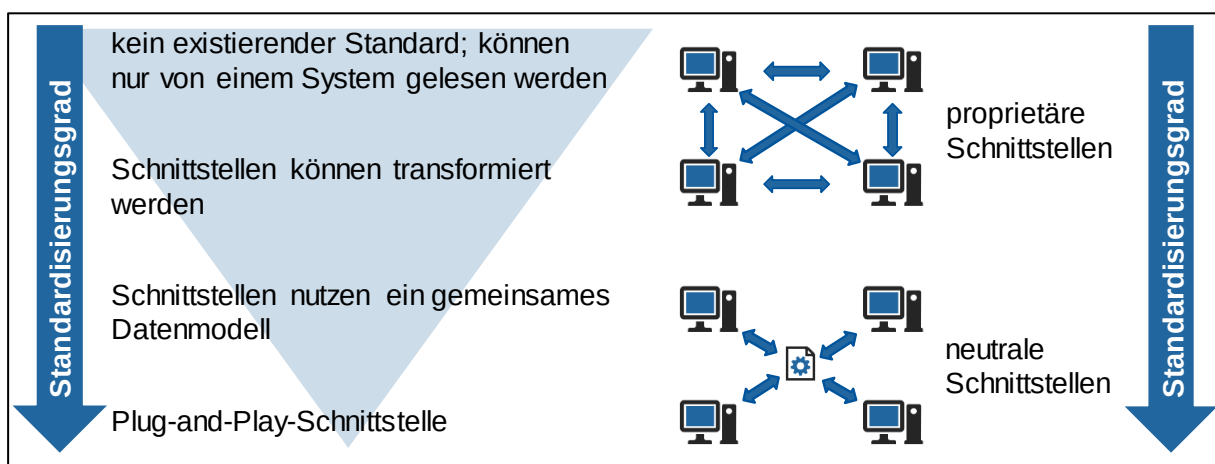


Abbildung 2-12: Standardisierungsgrad von Schnittstellen (eigene Darstellung i. A. a. NEUMANN 2007, S. 4; SINDERMAN 2014, S. 329)

⁴ Comma Separated Values (CSV) sind kommagetrennte Werte in einer einfachen Textdatei.

Im Optimalfall entstehen sogenannte „Plug and Play“-Schnittstellen, die eine direkte Nutzung nach der Kopplung ermöglichen. Prominentestes Beispiel hierzu ist der USB- (*Universal Serial Bus*)-Standard, welcher die einfache Nutzung von Peripheriegeräten an Rechnern ermöglicht (s. NEUMANN 2007, S. 4; SINDERMAN 2014, S. 328-330; LEIMEISTER 2015, S. 100).

Standardisierte Schnittstellen zwischen Informationssystemen gibt es in großer Zahl, je nach Anwendungsgebiet und Ebene im Schichtenmodell. Im Folgenden sollen daher nur die für das Anwendungsgebiet der vorliegenden Arbeit relevantesten vorgestellt werden.

Die **technische Verbindung** zwischen zwei Informationssystemen ermöglichen etablierte Standards wie das im vorigen Kapitel vorgestellte verbindungsorientierte TCP oder das verbindungslose UDP (*User Datagram Protocol*) (s. ALPAR ET AL. 2016, S. 472). Über diese Protokolle wird sichergestellt, dass sich die betroffenen Hardwarekomponenten über ein Netzwerk finden, verstehen und die Kommunikation sicher abläuft. Auch innerhalb von Informationssystemen, zum Beispiel bei Trennung nach dem oben vorgestellten 3-Tier-Modell, muss die Kommunikation zwischen den Komponenten sichergestellt werden. Zentral ist dabei vor allem die Aufgabenverteilung zwischen den einzelnen Komponenten (*Tiers*). Bei strenger Aufgabenteilung zwischen Präsentation, Prozess und Datenhaltung wird vom PPD-Modell gesprochen (s. LEIMEISTER 2015, S. 108). Oberhalb der technischen Ebene erfolgt die Kommunikation zwischen den Einzelkomponenten eines Informationssystems oft über proprietäre Schnittstellen, vor allem, wenn alle Komponenten des PPD von einem Softwarehersteller stammen. Dabei erfolgt die Kommunikation zwischen den Schichten *Prozess* und *Datenbank* oft über einen vom Hersteller der Datenbanklösung (z. B. *Oracle*) bereitgestellten, integrierten Client.

Für die **Kommunikation auf Anwendungsebene** zwischen verschiedenen Informationssystemen haben sich, oft spezifische für bestimmte Anwendungsfälle, **normierte Standards** für Schnittstellen durchgesetzt. Eher ausgelegt für die Kommunikation zwischen Systemen unterschiedlicher Unternehmen sind Standards, die unter den Oberbegriffen EDI (*Electronic Data Interchange*) und MQ (*Message Queue*) entwickelt wurden. Im Kontext EDI, das Standards zum Austausch von strukturierten Daten wie Rechnungen oder Bestellungen zwischen Unternehmen zusammenfasst, ist vor allem der EDIFACT-Standard zu nennen, der in der DIN ISO 9735 normiert ist (s. DIN ISO 9735). Durch dessen Einsatz müssen Unternehmen keine Datenkonvertierung vornehmen, wenn sie Daten von anderen Unternehmen empfangen (s. LEIMEISTER 2015, S. 171). Des Weiteren gibt es viele **Standards, die spezifisch für eine Branche oder Region entwickelt** wurden, zum Beispiel VDA im Automobilbereich und SWIFT im Bankensektor oder ANSI X.12 im amerikanischen Raum (s. DATA INTERCHANGE PLC 2012, S. 3). Es existieren jedoch auch Lösungen, welche die Konvertierung von EDI-Nachrichten vornehmen, wie das der *MyOpenFactory*, die beispielsweise verschiedene ERP-Systeme integriert und somit einen automatisierten Austausch ermöglicht (s. SCHMIDT ET AL. 2014, S. 401; FRECHT 2013, S. 55). *MyOpenFactory* gibt somit

auch KMU eine Möglichkeit, Bestellabwicklungsprozesse mit Partnern mit Systemen zu unterstützen beziehungsweise vollständig automatisiert abzuwickeln (s. SCHUH 2008, S. 29). MQ, meist in Message-Oriented-Middleware umgesetzt, basiert auf einem Publish-Subscribe-Modell zum asynchronen Nachrichtenaustausch. Es wird vom sendenden System eine Nachrichtenschlange (engl. *Queue*) erstellt, die vom empfangenden System abgerufen wird. Auch hier gibt es unterschiedlichste Umsetzungen, wie beispielsweise das in der Automobilindustrie häufig genutzte *Websphere MQ* von IBM, aber auch eine Reihe von Open-Source-Lösungen, zum Beispiel auf Java-Basis (s. AHUJA U. MUPPARAJU 2014).

Für den Datenaustausch zwischen Informationssystemen innerhalb eines Unternehmensnetzwerks wird vor allem das **API-Konzept** genutzt. API steht allgemein für ein *Application Programming Interface*, also eine von Anwendungen im Programmcode direkt angesteuerte Schnittstelle. APIs erlauben die Übergabe von Daten und Befehlen von einem System an ein anderes mit entsprechender Rückmeldung des zweiten Systems an das erste. Für die Übertragung der Daten werden Standards wie *XML (Extensible Markup Language)* oder *JSON (JavaScript Object Notation)* eingesetzt. Welche Daten und Befehle übertragen werden können, müssen in der jeweiligen Dokumentation der Systeme hinterlegt werden, oder sind in Softwarebibliotheken hinterlegt und so für andere Systeme nutzbar (s. NIJIM U. PAGANO 2014, S. 4; ANUFF 2014, S. 3ff; PUSCHMANN U. AUSSERHOFER 2017, S. 148; STAHL U. STAAB 2017, S. 32).

Das am stärksten wachsende API-Protokoll ist das sogenannte **REST-Protokoll** (*Representational State Transfer*), das inzwischen den größten Anteil an veröffentlichten Webschnittstellen einnimmt (s. NIJIM U. PAGANO 2014, S. 38; PUSCHMANN U. AUSSERHOFER 2017, S. 147).⁵ Ursprünglich von FIELDING definiert, unterstützt es die folgenden Eigenschaften: das Client-Server-Prinzip, eine statuslose Verbindung, um den Client unabhängig vom Server zu gestalten, die Möglichkeit des Cachings zur Unterstützung großer Netzwerke mit Latenzen, eine einheitlich definierte Schnittstelle auf Client- und Serverseite, mehrschichtige Systeme sowie die Möglichkeit zur Übergabe von Programmcode auf Abruf vom Server an den Client (s. FIELDING 2000, S. 76ff). Parallel nimmt auch **SOAP** (*Simple Object Access Protocol*) einen wichtigen Anteil der heute genutzten Schnittstellen-Protokolle ein. Auch SOAP definiert die Kommunikation zwischen zwei Systemen und nutzt dazu XML-Nachrichten, sogenannte *Envelopes*. Meist wird SOAP zusammen mit HTTP (*Hypertext Transfer Protocol*) genutzt, kann aber auch über andere Protokolle wie FTP (*File Transfer Protocol*) abgebildet werden

⁵ Neben den etablierten Schnittstellen ist derzeit das 2016 von Facebook veröffentlichte GraphQL verstärkt in der Diskussion, REST abzulösen. Außerhalb von Facebook Inc. ist die Anwendung dieser Technologie zum Zeitpunkt der Abfassung dieser Arbeit aber nur vereinzelt umgesetzt (vgl. BUNA 2016; HARTIG U. PÉREZ 2017).

(s. GUDGIN ET AL. 2007). So standardisierte Schnittstellen werden von Softwarehersteller vermehrt in ihre Software integriert, um den Datenaustausch zwischen ihrem und anderen Systemen nahtlos zu gestalten.

Auch in der Diskussion zur **Umsetzung der Industrie 4.0** nehmen Schnittstellen mit dem ultimativen Ziel einer Plug-and-Play-Lösung einen wichtigen Stellenwert ein (s. NYHUIS ET AL. 2017, S. 34). Der dort aktuell am meisten diskutierte Standard ist **OPC-UA** (*Open Platform Communications - Unified Architecture*), das durch das IEC international genormt ist (s. SCHLEIPEN 2016; HOPPE 2016, S. 374) und sich langsam im Produktionsumfeld etabliert (s. DREMEL U. HERTERICH 2016, S. 654). OPC-UA wurde primär für die vertikale Kommunikation und Integration, also vom Sensor auf dem Shopfloor bis hin zum ERP-System, entwickelt, um maximale Interoperabilität zu leisten. Es basiert auf dem TCP-Protokoll, worauf aufbauend generische Dienste beschrieben sind, die betriebssystemunabhängig implementiert werden können. Aufbauend auf den generischen Diensten können Daten, Alarme und Events übertragen werden, welche generische Informationsmodelle nutzen (s. OPC FOUNDATION 2013, S. 12-13; HOPPE 2016, S. 371ff).

2.1.5 Informationssystem-Architekturen

In den vorherigen Kapiteln wurden die Bausteine beschrieben, aus denen sich eine Informationssystem-Architektur zusammensetzt. Generell wird unter einer Architektur sowohl die Einordnung als auch die Verbindung von Systemelementen verstanden (s. ISO 15704, S. 1; FRIEDLI 2000, S. 79). Entsprechend beschreibt die **Informationssystem-Architektur die Anordnung der grundlegenden Elemente**, also der Informationssysteme mit den darin enthaltenen Informationen, entlang von definierten Dimensionen und deren Verknüpfung über Schnittstellen untereinander und zur Umwelt (s. WINTER U. AIER 2016; HABERFELLNER ET AL. 2015, S. 182; HANSCHKE 2016, S. 36). Informationssystem-Architekturen haben grundsätzlich drei Funktionen im Unternehmen (s. HEINRICH ET AL. 2014, S. 87): Zuerst haben sie eine **beschreibende Funktion**, indem sie eine **Sicht auf den Ist-Stand** liefern. Im zweidimensionalen Raum kann diese Sicht immer nur Teile der Information bieten. Deshalb werden zur Darstellung von Informationssystem-Architekturen auch verschiedene Teilarchitekturen genutzt. Dies sind beispielsweise die Technologiearchitektur, in der im Einsatz befindliche Elemente der Infrastruktur dargestellt werden, die Datenarchitektur, in welcher Daten und Informationen in Informationssystemen dargestellt werden, oder die IT-Applikationsarchitektur, welche die Anwendungen aus Endanwendersicht aufzeigt (s. TIEMEYER 2016, S. 45; 2017a, S. 105-106). In Kapitel 6.7 werden verschiedene Teilarchitekturen in der konkreten Anwendung vorgestellt.

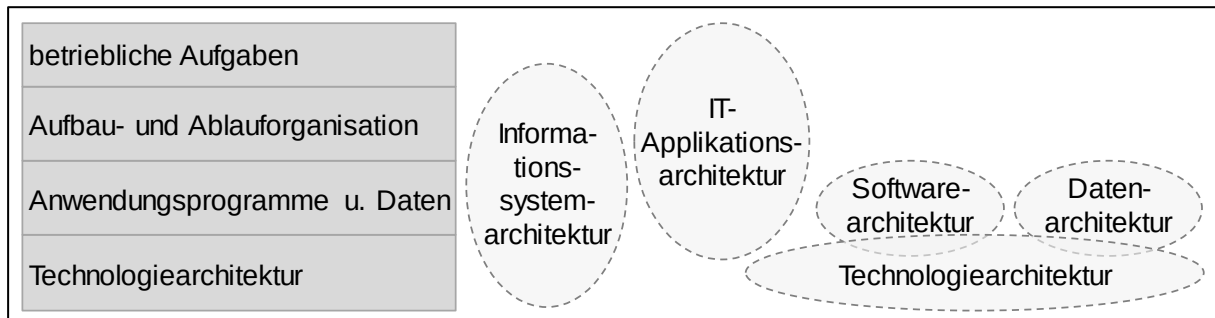


Abbildung 2-13: Teilarchitekturen von Informationssystem-Architekturen (i. A. a. HEINRICH ET AL. 2014, S. 88)

Abbildung 2-13 gibt einen Überblick über die Einordnung verschiedener Teilarchitekturen. Sie zeigt auf, dass eine Informationssystem-Architektur gewissermaßen das Bindeglied zwischen der Geschäftsarchitektur (betriebliche Aufgaben) und der technischen Infrastruktur (Technologiearchitektur) bildet (s. HANSCHKE 2016, S. 36). In der Abbildung zeigen die Rechtecke, welche Inhalte in den Teilarchitekturen enthalten sind, während die Ovale die Teilarchitekturen darstellen. Die Abgrenzung letzterer ist nicht immer eindeutig, weshalb die Ovale mit gestrichelten Umrissen dargestellt sind (s. HEINRICH ET AL. 2014, S. 88).

Weiterhin hat die Informationssystem-Architektur eine **Kommunikationsfunktion**. Fachbereiche, IT-Abteilung und Geschäftsführung haben meist eine unterschiedliche Sicht auf dieselben Sachverhalte innerhalb von Informationssystem-Architekturen. Die Architekturbeschreibung mit ihren verschiedenen Teilaspekten dient dazu, in Projekten, bei Entscheidungen oder im Controlling Kommunikationsdefizite auszugleichen.

Zuletzt hat Informationssystem-Architektur eine **Gestaltungsfunktion**, da vom Ist-Stand aus **Soll-Zustände für die künftige Entwicklung** abgeleitet werden können. Dazu werden zum Beispiel Soll-Bebauungspläne genutzt, welche im Zuge von IT-Strategie-Maßnahmen über Projekte umgesetzt werden (s. Kapitel 2.2.3). In der Gestaltung soll die Informationsinfrastruktur koordiniert und weiterentwickelt werden mit den Zielen der Vereinfachung und Redundanzvermeidung sowie möglichst guter Integration und Automatisierung der Informationssysteme (s. HEINRICH ET AL. 2014, S. 86-87; TIEMEYER 2017a, S. 99-100).

Zum Management von Informationssystem-Architekturen haben sich eine Reihe von Standardframeworks etabliert. Diese werden in Kapitel 3.3 vorgestellt, darauf aufbauend wird in Kapitel 6 ein Beschreibungsmodell von Informationssystem-Architekturen erarbeitet.

2.1.6 Informationssysteme in produzierenden Unternehmen

Bevor Spezifika von Informationssystemen in produzierenden Unternehmen vorgestellt werden, soll hier zuerst beschrieben werden, was in dieser Arbeit unter einem produzierenden Unternehmen verstanden wird.

Produzierende Unternehmen

Lehrbücher unterschiedlicher Disziplinen schlagen verschiedene Unterscheidungen der Wirtschaftszweige vor. Die Volkswirtschaftslehre nach FISHER und CLARK teilt sie in Primär-, Sekundär- und Tertiärsektoren. Der Sekundärsektor, auch industrieller Sektor genannt, verarbeitet Güter aus dem Primärsektor (Rohstoffe) zu Produkten weiter (vgl. FISHER 1935; CLARK 1941). In der Betriebswirtschaftslehre gibt es nach WÖHE eine Unterteilung der Wirtschaftszweige in Sachleistungs- und Dienstleistungsebene (s. WÖHE ET AL. 2016, S. 31). Im Zuge der europäischen Harmonisierung hat sich jedoch das Verständnis der EUROPÄISCHEN UNION (EU) durchgesetzt.

Die EU nimmt eine Unterscheidung der Wirtschaftszweige gemäß der statistischen Systematik der Wirtschaftszweige in der europäischen Gemeinschaft vor (s. EUROPÄISCHE KOMMISSION 2008a). Demnach ordnet sie folgende **Wirtschaftszweige dem produzierenden Gewerbe** zu: Bergbau und Gewinnung von Steinen und Erden, verarbeitendes Gewerbe, Energie und Wasserversorgung und das Baugewerbe (s. EWALD 1996, S. 34). Dem verarbeitenden Gewerbe lassen sich somit insbesondere Unternehmen, die Metallerzeugnisse produzieren oder direkt an der Metallerzeugung und -bearbeitung beteiligt sind, sowie der Maschinen- und Anlagenbau zuordnen (s. EUROPÄISCHE KOMMISSION 2008a; 2008b, S. 47).

Das **verarbeitende Gewerbe stellt Produkte durch eine substanzielle Veränderung der eingesetzten Materialien her oder veredelt Erzeugnisse**. Dabei stellen Unternehmen des verarbeitenden Gewerbes durch mechanische, physikalische oder chemische Umwandlung von Ausgangsmaterialien durch ihre wirtschaftlichen Tätigkeiten neue Waren her, ohne dass es sich dabei um sogenannte „handelsübliche Manipulationen handelt“ (BUNDESAMT FÜR WIRTSCHAFT (Hrsg.) 2013, S. 6). In Anlehnung an das verarbeitende Gewerbe liegt der Hauptfokus produzierender Unternehmen demnach auf der Herstellung und Bereitstellung von Produkten; alle freien Kapazitäten werden in die Verbesserung der Produkte sowie in die zur Herstellung notwendigen Prozesse investiert. In Deutschland betrug der Anteil des produzierenden Gewerbes an der Entstehung des Bruttoinlandsprodukts 2017 25,6 Prozent beziehungsweise 835,4 Mrd. Euro (s. STATISTISCHES BUNDESAMT 2018, S. 11).

In dieser Arbeit werden unter **produzierenden Unternehmen solche des verarbeitenden Gewerbes** nach Definition der Europäischen Kommission verstanden. Darin werden alle Typen von fertigenden Unternehmen mit einbezogen, sowohl Massenproduzenten in Chargen- oder Serienproduktion, als auch Einzelproduzenten im Make-to-Order-Verfahren wie beispielsweise Sondermaschinenbauer (s. WEBER ET AL. 2018, S. 205-206; DYCKHOFF U. SPENGLER 2010, S. 232).

Informationssysteme in produzierenden Unternehmen

Informationssysteme in produzierenden Unternehmen stehen seit deren Einführung in den 1950er Jahren in einem ständigen, **langsam fortschreitenden Wandel**. Während zu Beginn die programmierbare Informationsverarbeitung im Vordergrund stand, entwickelten sich Systeme entlang der Themen Konnektivität, Automatisierung und IT-

Industrialisierung weiter (s. ZELLER ET AL. 2016, S. 203). Wie in Abbildung 2-5 gezeigt, können diese Systeme nach verschiedenen Kriterien strukturiert werden. Informationssysteme im betrieblichen Einsatz decken unterschiedlich viele Funktionen und Verwendungszwecke ab und werden dementsprechend von einer variierenden Anzahl an Personen innerhalb und außerhalb des Unternehmens genutzt. Somit gibt es in den meisten Unternehmen alle Variationen, von der Individual-Software, die nur von einem einzigen Anwender genutzt wird, bis zur Standardsoftware wie einem ERP-System, das einen Großteil der Unternehmensprozesse unterstützt und von den meisten Mitarbeitern des Unternehmens bedient wird (s. LEMKE U. BRENNER 2015, S. 149).

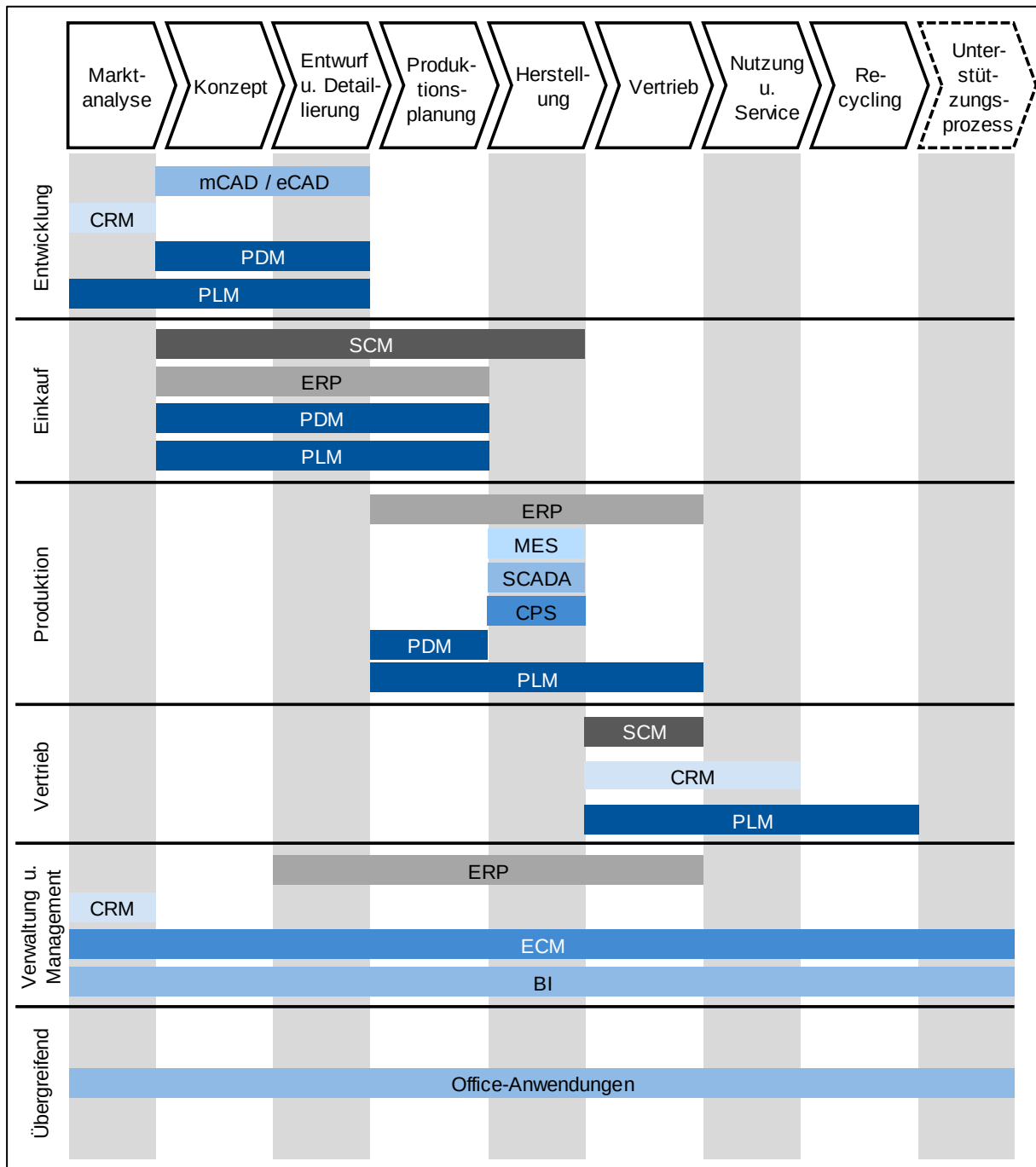


Abbildung 2-14: Typische Informationssystem-Architektur eines produzierenden Unternehmens (eigene Darstellung)

In jüngerer Zeit haben sich eine Reihe von **betrieblichen Standardsoftwaresystemen** herausgebildet, die heute in produzierenden Unternehmen im Einsatz sind (S. KAMPKER ET AL. 2011, S. 180; SCHUH ET AL. 2017h, S. 119). Zwar sind die tatsächlich auf dem Markt befindlichen Softwareprodukte in ihren Funktionen nicht komplett disjunkt, trotzdem können die Informationssysteme so eingeteilt werden. Abbildung 2-14 ordnet diese Systeme entsprechend der Nutzung im Unternehmenskernprozess entsprechend VDI 2216 (s. VDI 2219) und den Organisationseinheiten eines produzierenden Unternehmens (s. SCHUH U. KAMPKER 2011, S. 144-145) ein (vgl. TIEMEYER 2017a, S. 126; LEMKE U. BRENNER 2015, S. 144-147). Die gewählte Darstellungsform wird in Kapitel 6.7.3 erklärt.

Entlang des Kernprozesses eines produzierenden Unternehmens werden dabei die folgenden Systeme genutzt. Eine genauere Beschreibung findet sich in Kapitel 6.4.1:

- **Customer Relationship Management (CRM)**: Beinhaltet die sogenannten *Customer Journey*, bei der die Interaktion mit Kunden hinterlegt werden.
- **PLM**: Ist als Konzept zur ganzheitlichen und unternehmensweiten Steuerung von Produktdaten entlang des Produktlebenszyklus zu verstehen. Dies umfasst auch das Produktdatenmanagement (PDM) in der Entwicklung.
- **Computer Aided Design (CAD)**: CAD-Systeme unterstützen die mechanische (mCAD) und elektrische Entwicklung (eCAD).
- **ERP**: Ist fokussiert auf die fristgemäße Planung und Steuerung aller Ressourcen im Unternehmen wie Kapital, Personal, Betriebsmittel und Material.
- **Manufacturing Execution System (MES)**: ME-Systeme dienen dem Fertigungsmanagement basierend auf einer Echtzeitplanung zur konkreten Planung, Lenkung, Kontrolle und Auswertung von Aktionen in der Fertigung.
- **Supply Chain Management (SCM)**: Bildet die softwaretechnische Unterstützung für die Koordination des überbetrieblichen Material- und Informationsflusses über die gesamte Wertschöpfungskette.
- **Enterprise Content Management (ECM)**: Bezieht sich nicht auf einen bestimmten Unternehmensprozess, sondern hat die Aufgabe, den Umgang und die Präsentation von Inhalten zu planen, steuern, verwalten und zu koordinieren.
- **Business Intelligence (BI)**: Liefert Werkzeuge zur Entscheidungsunterstützung auf Basis von Daten.
- **Office- und Kollaborationssysteme**: Dienen der täglichen Arbeitsunterstützung als Editoren und zur Kommunikation für alle Fälle, in denen kein spezifisches Informationssystem vorhanden ist.
- **Operational Technology (OT)** und **cyber-physische Systeme (CPS)**: OT ist Hardware und Software, die eine Änderung durch die direkte Überwachung o-

der Kontrolle von physischen Geräten, Prozessen und Ereignissen im Unternehmen erkennen oder verursachen. Im Gegensatz zur klassischen OT sind CPS offene Systeme, die Verbindungen zwischen Vorgängen der physischen Realität und den heute verfügbaren Netzinfrastrukturen bieten. Sie bestehen aus Sensorik, Aktorik, Komponenten zur Datenanalyse und definierten Schnittstellen zu anderen Informationssystemen sowie zum menschlichen Bediener.

Eine **klassische Darstellung** zur Strukturierung der in produzierenden Unternehmen eingesetzten Informationssysteme ist die sogenannte **Automatisierungspyramide**. Insbesondere durch die Entwicklung von CPS werden die benötigten Daten nicht mehr zwangsweise auf den verschiedenen Ebene der Automatisierungspyramide beziehungsweise der entsprechenden Systeme abgelegt, auch die Trennung zwischen Informationstechnologie (IT) und OT verschwimmt. Vielmehr erfolgt eine **dezentrale Datennutzung** an dem Ort, an dem die Daten für eine flexible und effiziente Produktion benötigt werden. Die klassische **Automatisierungspyramide löst sich damit auf**, ein Netz aus verschiedenen Elementen ermöglicht eine nicht-hierarchische, dezentrale Bereitstellung der relevanten Daten und Funktionen. Abbildung 2-15 stellt diese Entwicklung vereinfacht dar (s. BAUERNHANSL ET AL. 2016, S. 11-12; VDI/VDE-GESELLSCHAFT, 2013, S. 4).

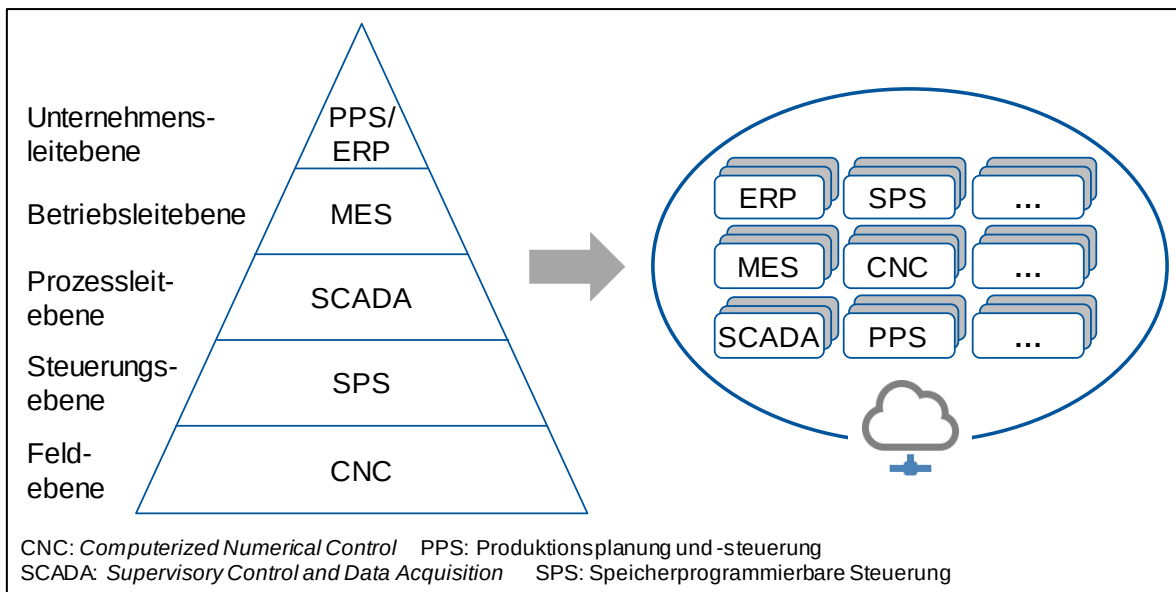


Abbildung 2-15: Auflösung der Automatisierungspyramide (eigene Darstellung i. A. a. VDI/VDE-GESELLSCHAFT, 2013, S. 4; BAUERNHANSL ET AL. 2016, S. 11)

Getrieben durch diese Entwicklung verschmelzen IT und OT zunehmend. Dies hat zur Entstehung des IoT geführt; der Begriff wurde erstmals von KEVIN ASHTON im Jahr 1999 verwendet (s. ZITNIK ET AL. 2016, S. 857; KRAUSE ET AL. 2017, S. 6). Unter IoT wird die Verknüpfung eindeutig identifizierbarer physischer Objekte mit einer virtuellen Repräsentation und digitalen Modellierung im Internet verstanden (s. FREY 2015, S. xxvi). Die grundsätzliche Idee ist dabei ähnlich zu CPS, agiert dabei aber branchenübergreifend. Durch das IoT wird eine Vernetzung zwischen einer Vielzahl von Objekten mit dem Internet angestrebt. Im betrieblichen Umfeld sind es ebenfalls Sensorik, Aktorik und Informationssysteme, die vernetzt werden. Zusätzlich werden mit dem IoT aber

auch weitere, heterogene Geräte, beispielsweise mobile Endgeräte (Smartwatch), beziehungsweise allgemein Objekte, die über eine Schnittstelle zur Vernetzung verfügen, angebunden. Das IoT kann demnach auch als eine dynamische, selbstkonfigurierende und globale Netzwerkinfrastruktur angesehen werden, die eine ganzheitliche Vernetzung anstrebt (s. GRONAU ET AL. 2016, S. 1-3).

2.2 Management von Informationssystem-Architekturen

2.2.1 Informationsmanagement

Informationsmanagement hat die Aufgabe, die vorgenannten Informationssysteme und -architekturen zu regeln. Dabei geht es weniger um das Management von Informationstechnologie, weshalb hier auch vom Begriff IT-Management (vgl. z. B. TIEMEYER 2017b, S. 21) abgesehen werden soll, entsprechend der Diskussion um CARRs Artikel „IT Doesn't Matter“ (s. CARR 2003, S. 24-38), die VARIAN sehr passend zusammenfasst:

*„It is not information technology itself that matters, but how you use it.“
(VARIAN 2004)⁶*

Vielmehr liegt der Fokus des Informationsmanagements auf dem richtigen Umgang mit der Informationstechnologie und deren Nutzen für das Unternehmen. Damit befindet sich das Informationsmanagement im Dilemma zwischen Ermöglichung der Unternehmensstrategie mit Informationssystemen und der Angleichung der Informationssysteme an die Unternehmensstrategie, wie Abbildung 2-16 zeigt.

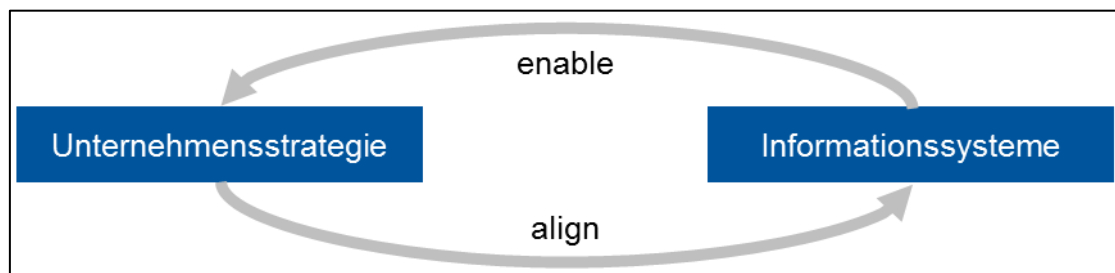


Abbildung 2-16: Informationsmanagement zwischen Ermöglichen der Unternehmensstrategie und Angleichen an die Unternehmensstrategie (i. A. a. KRCMAR 2015, S. 93)

Als wichtige Führungsaufgabe im Rahmen des Managements produzierender Unternehmen nimmt das **Informationsmanagement** heute eine **tragende Rolle im Unternehmen** ein. Unter Management wird in diesem Kontext Planung, Organisation, Führung und Kontrolle im Unternehmen verstanden. Es wird dabei zwischen dem institutionellen, funktionalen und individuellen Management unterschieden. Ersteres befasst sich mit den Personengruppen, die unter einer Legitimation ihre Führungsaufgabe

⁶ Deutsch etwa: *Nicht die Informationstechnologie selbst ist entscheidend, sondern der Umgang damit.* Antwort von VARIAN auf CARRs Artikel „IT Doesn't Matter“ (vgl. CARR 2003) beziehungsweise dessen Buch „Does IT Matter?“ (vgl. CARR 2004).

wahrnehmen, das Zweite mit den Aufgaben, welche von der führenden Person übernommen werden und Letzteres mit den einzelnen handelnden Akteuren (s. BOOS ET AL. 2011, S. 3-5; KRCMAR 2015, S. 24ff; RÜEGG-STÜRM U. GRAND 2017, S. 33-34).

Im **funktionalen Management** gibt es wiederum eine Reihe von unterschiedlichen Modellen, welche Aufgaben des Informationsmanagements darstellen. So unterscheiden HEINRICH ET AL. zwischen strategischen, administrativen und operativen Aufgaben (s. HEINRICH ET AL. 2014, S. 32-34), während KRCMAR nach Führungsaufgaben sowie dem Management der Informationswirtschaft, der Informationssysteme und der Informations- und Kommunikationstechnik einteilt (s. KRCMAR 2015, S. 107). In einer umfangreichen Literaturanalyse hat DEINDL konkrete Aufgaben des Informationsmanagements zusammengefasst, die für produzierende Unternehmen von Relevanz sind. Diese sind in der folgenden Tabelle zusammengefasst. Die Zuordnung der Aufgaben nach KRCMAR (s. Tabelle 2-3 Mitte) soll für diese Dissertationsschrift übernommen werden.

Tabelle 2-3: Aufgaben des Informationsmanagements (eigene Darstellung, s. DEINDL 2013a und vgl. FERSTL U. SINZ 2013; KRCMAR 2015; HEINRICH ET AL. 2014; LEHNER ET AL. 2008; BUCHTA ET AL. 2010; ZARNEKOW U. BRENNER 2004; BIETHAHN ET AL. 2004)

Aufgabe nach DEINDL	Zuordnung nach KRCMAR	Zuordnung nach HEINRICH ET AL.
IT-Planung	Führungsaufgaben des Informationsmanagements	Strategisches Management
IT-Früherkennung	Führungsaufgaben des Informationsmanagements	Strategisches Management
Strategisches IT-Controlling	Führungsaufgaben des Informationsmanagements	Strategisches Management
Operativ-taktisches IT-Controlling	Führungsaufgaben des Informationsmanagements	Administratives Management
Anforderungsmanagement	Management der Informationswirtschaft	Administratives Management
IT-Projektmanagement	Management der Informationswirtschaft	Administratives Management
Informationssystem-Konzeption	Management der Informationssysteme	Administratives Management
Informationssystem-Umsetzung	Management der Informationssysteme	Administratives Management
Informationssystem-Einführung	Management der Informationssysteme	Administratives Management
Informationsinfrastrukturbetrieb	Management der Informations- und Kommunikationsinfrastruktur (IKT)	Operatives Management
Informationsfunktionsbereitstellung	Management der IKT	Operatives Management

2.2.2 IT-Komplexität

Eine zentrale Herausforderung des Informationsmanagements ist das Management der in den vergangenen Jahren zunehmenden IT-Komplexität (s. Kapitel 1.1). Dieses noch relativ junge Forschungsfeld beschäftigt sich seit gut zehn Jahren mit der Kausalität der IT-Komplexität (s. BEETZ 2014, S. 2). Die Ursprünge gehen allerdings zurück in die Mitte der 1950er Jahre und basieren auf den Strukturen der Systemtheorie und der allgemeinen Komplexitätstheorie. Prägend war hier das **Gesetz von ASHBY**, das besagt, dass „*only variety [...] can destroy variety*“ (ASHBY 1957, S. 207). Er beschreibt damit, dass die Varietät der Steuerung mindestens so groß sein muss wie die Varietät der Störung. Übertragen auf die Komplexität bedeutet dies, dass ein System mit gegebener Komplexität nur mit einem mindestens ebenso komplexen System beherrscht werden kann. Dies liefert auch die Begründung für den Zusammenhang zwischen Komplexität auf Seiten des Unternehmens und der IT-Komplexität, der von Beratungsgesellschaften, wie beispielsweise BCG, identifiziert wurde (s. Abbildung 2-17). Die Aufgabe im IT-Komplexitätsmanagement ist es daher, **nichtwertschöpfende IT-Komplexität zu erkennen und möglichst zu verhindern**.

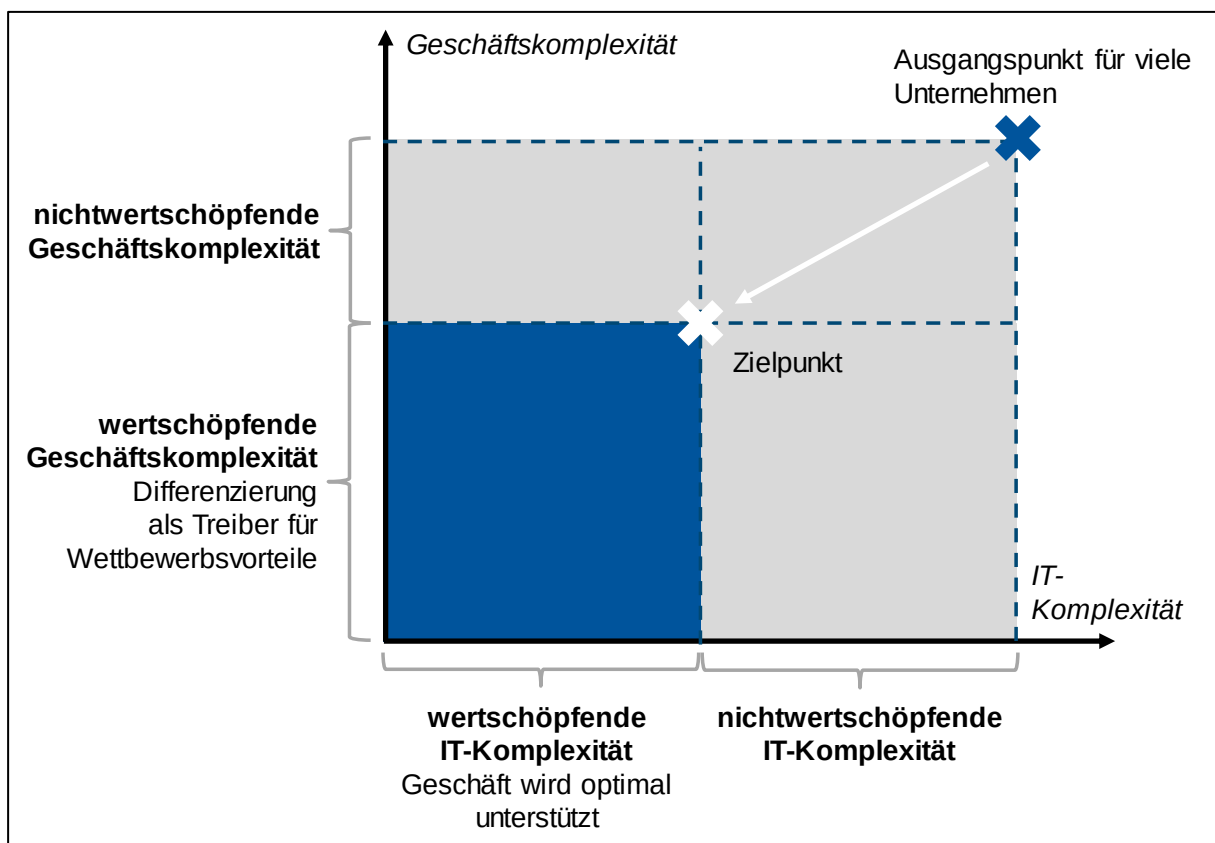


Abbildung 2-17: IT-Komplexität und Geschäftskomplexität (i. A. a. GREBE U. DANKE 2013, S. 3)

Wichtige Ergänzungen zum Komplexitätsbegriff gab LUHMANN in seinen Ausführungen im Kontext der Organisationstheorie. Er beschreibt den Begriff der Komplexität als mehrdimensional und definiert dafür drei Dimensionen: (1) die Zahl der Elemente, (2) die Zahl der im System möglichen Beziehungen zwischen den Elementen und (3) die

Verschiedenartigkeit dieser Beziehungen (s. LUHMANN 1980, S. 1064-1065). Diese Dimensionen wurden von verschiedenen Autoren weiterentwickelt (vgl. WINZER 2016, S. 10-11) und von SCHUH ET AL. definiert mit den **drei Komplexitätstreibern** (1) *Interdependente Vielfalt*, (2) *Verunsichernde Dynamik* und (3) *Wahrnehmung* (s. SCHUH ET AL. 2013, S. 40; SCHUH U. RIESENER 2018, S. 12-13). Diese Komplexitätstreiber **treffen auch auf Informationssystem-Architekturen zu** und können übertragen werden. In Kapitel 7.1 schlägt der Autor darauf aufbauend eine eigene umfangreiche Definition von IT-Komplexität vor.

2.2.3 IT-Strategie

Zum Management der im vorigen Kapitel beschriebenen IT-Komplexität ist ein strukturiertes Vorgehen unumgänglich. Deshalb ist eine **Aufgabe des Informationsmanagements**, die **IT-Planung** regelmäßig durchzuführen, wodurch eine IT-Strategie festgelegt wird (s. Kapitel 2.2.1). Solche IT-Strategien sind in vielen produzierenden Unternehmen, vor allem in KMU, oft noch unüblich (s. LEIMEISTER 2015, S. 189), mit zunehmender Bedeutung des Informationsmanagements jedoch unumgänglich (s. TIEMEYER 2017b, S. 7-9). Neben der Festlegung der **Rahmenbedingungen zur Entwicklung der Informationssysteme** ist vor allem die Optimierung der Zusammenarbeit zwischen Fachbereichen und IT-Abteilung ein wichtiger Aspekt bei der Gestaltung der IT-Strategie (s. ZELLER ET AL. 2016, S. 201). Dieses sogenannte **Business-IT-Alignment** schafft nicht nur eine bessere Unterstützung der Geschäftsprozesse durch Informationstechnologie, sondern ermöglicht auch neue Innovationen für die Fachbereiche. Ein Zusammenhang zwischen guter Unterstützung von Geschäftsprozessen durch Informationssysteme und dem Geschäftserfolg konnte durch den *Business Performance Index* nachgewiesen werden (s. GROß 2016, S. 2).

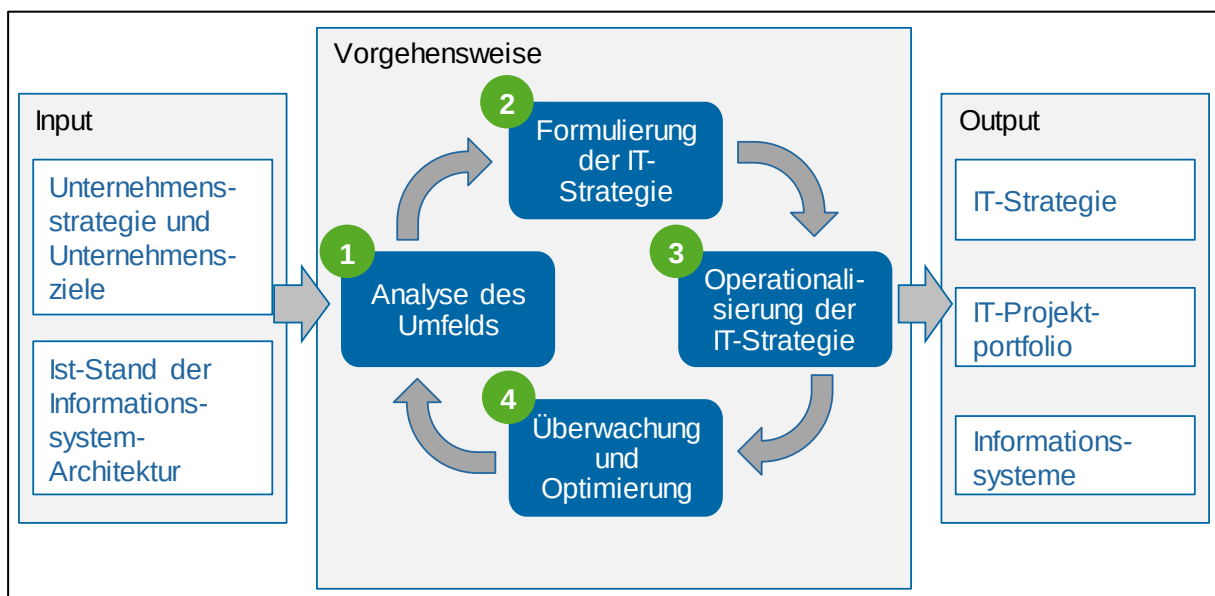


Abbildung 2-18: Vorgehen zur Gestaltung einer IT-Strategie (eigene Darstellung)

Um das Business-IT-Alignment umzusetzen, leitet sich eine IT-Strategie aus zwei Quellen ab: Der erste **Inputgeber** ist die **Unternehmensstrategie**, welche die Wei-

terentwicklung des Unternehmens zum Beispiel in neue Märkte oder mit neuen Produkten darstellt (s. KRCMAR 2015, S. 399). Als zweiter Inputgeber dient der **aktuelle Stand der Informationssystem-Architektur** und des Informationsmanagements, inklusive der IT-Organisation, der IT-Methoden und der IT-Kultur. Das Vorgehen zur Entwicklung einer IT-Strategie besteht, wie in Abbildung 2-18 gezeigt, aus vier iterativ durchzuführenden Schritten. Der Beginn ist die **Analyse** des Umfelds, in dem die Strategie entwickelt wird. Dazu zählt sowohl die Beachtung der relevanten Anspruchsgruppen, wie auch der Unternehmensstrategie und des Ist-Stands. Aus dieser Analyse ergeben sich Handlungsfelder, die über die **Formulierung** der Strategie im zweiten Schritt angegangen werden. Dabei werden die Vision des Informationsmanagements sowie Zielbilder für einzelne Bereiche, wie zum Beispiel eine Zielinformationssystem-Architektur, entwickelt. Die Zielbilder werden im Schritt der **Operationalisierung** in ein konkretes Projektportfolio überführt, aus dem letztendlich neue oder veränderte Informationssysteme entstehen. Der vierte Schritt des IT-Strategievorgehens beschreibt die kontinuierliche **Überwachung** und **Optimierung**, um Fehler schnell zu identifizieren und bei geänderten Rahmenbedingungen schnell nachsteuern zu können (s. LEIMEISTER 2015, S. 191-193).

Die IT-Strategie alleine liefert allerdings keinen Geschäftserfolg. Auch die **stetige Anpassung von ineffizienten Geschäftsprozessen und Überarbeitung der Unternehmensstrategie** durch wechselnde Marktumfelder sind notwendig, um die Effizienz und Effektivität der Fachbereiche zu steigern (s. BUCHTA ET AL. 2010, S. 24-26).

2.2.4 Rolle der IT-Abteilung

Die im Zuge der bisherigen Diskussion beschriebenen **Aufgaben des Informationsmanagements** werden in produzierenden Unternehmen teilweise **durch die IT-Abteilung wahrgenommen**. Wie groß der Anteil ist, hängt sehr vom Unternehmen, dessen Historie und Strategie ab. In der Literatur, bestätigt durch eine Reihe von Expertengesprächen, haben sich **vier Organisationstypen** von IT-Abteilungen herausgebildet, die in Abbildung 2-19 dargestellt sind (vgl. MCFARLAN ET AL. 1983; HANSCHKE 2016). Dabei können IT-Abteilungen zentral oder dezentral organisiert sein; es haben sich außerdem gemischte Modelle herausgebildet. Diese Unterscheidung ist für die vorliegende Arbeit jedoch nicht prioritär zu betrachten.

Die Typen zeigen gleichzeitig eine **Transformation, in der sich moderne IT-Abteilungen befinden**. Während sie zu Beginn der Entwicklung von Informations- und Kommunikationstechnologie als reiner Technologiepartner fungierten, werden sie heute meist als Typ C geführt. In diesem Fall haben Unternehmen den Strategiebeitrag erkannt, den die IT-Abteilung liefern kann (s. ZELLER ET AL. 2016, S. 102).

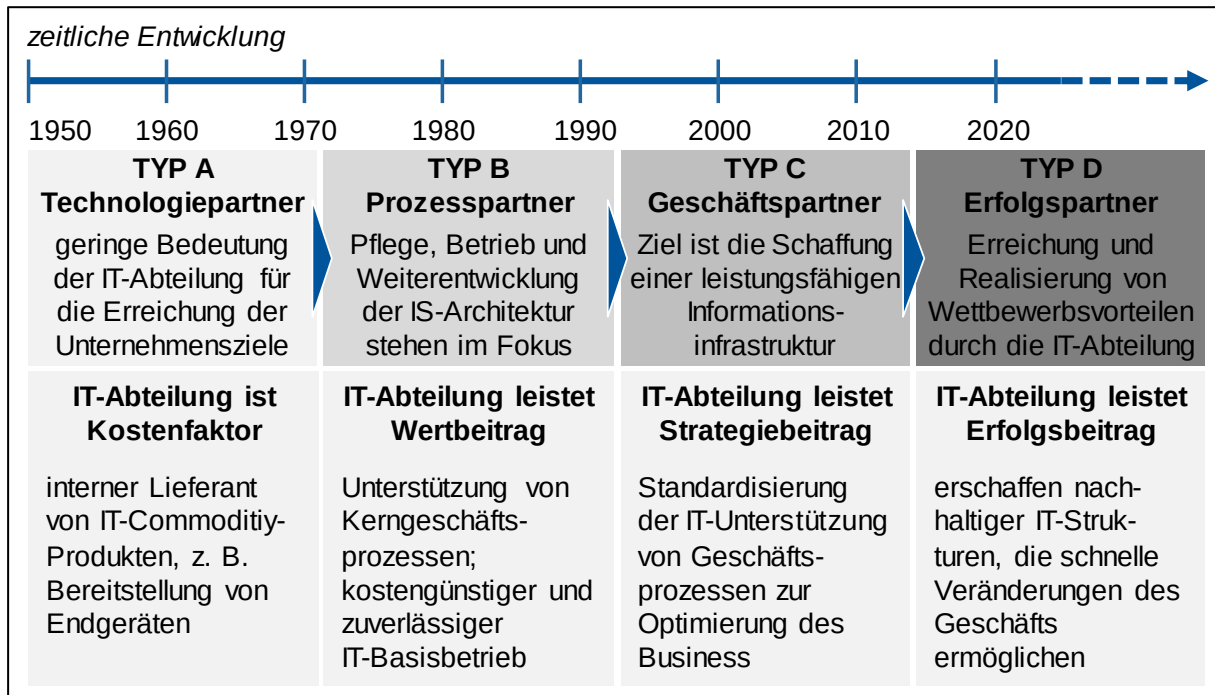


Abbildung 2-19: Entwicklung der Rolle der IT-Abteilung (eigene Darstellung i. A. a. MCFARLAN ET AL. 1983, S. 150-152; ZELLER ET AL. 2016, S. 102; HANSCHKE 2016, S. 184)

Entwickelt sich die IT-Abteilung nicht in dem Maße, in dem das Unternehmen und dessen Weiterentwicklung sie benötigt, entstehen in vielen Fällen Parallelgebilde. Diese, auch als **Schatten-IT** bezeichneten, Strukturen werden von Fachbereichen vorangetrieben, denen eine ausreichende IT-Unterstützung fehlt (s. ZIMMERMAN U. RENTROP 2012, S. 60-61). Die Folge dessen ist, dass der IT-Abteilung der Überblick fehlt und das genaue Ausmaß der Schatten-IT unklar ist (s. KOPPER ET AL. 2017, S. 98). Infolgedessen werden von der IT-Abteilung nicht-wertschöpfende Entscheidungen getroffen und die IT-Komplexität steigt (s. Kapitel 2.2.2). Ein Teil der IT-Entscheidungen wird darüber hinaus in dezentralen Fachbereichen getroffen, wodurch die IT-Abteilung weiter die Kontrolle über IT-Investitionen und Informationssysteme verliert und die IT-Komplexität wiederum steigt (s. KOPPER ET AL. 2017, S. 100-101). Um dies zu verhindern, muss die IT-Abteilung nicht alle Aufgaben des Informationsmanagements selbst erfüllen, es sollte jedoch klar definiert sein, wer im Unternehmen welche der Aufgaben übernimmt, damit eine klare Kommunikation sichergestellt werden kann. Aufgabe der IT-Abteilung ist demnach die Erhöhung des Reifegrads des Business-IT-Alignments. Gleichzeitig kann ein gewisses Maß an Schatten-IT auch positive Aspekte beinhalten, da sie nicht gestillte Nutzerbedürfnisse erfüllt und zur Identifikation von Verbesserungsmaßnahmen für die eigene Organisation und die eigenen Services herangezogen werden kann (s. KOPPER ET AL. 2017, S. 102).

2.3 Die Vision vom Platfformeinsatz im Kontext der vernetzten Digitalisierung

Über den Begriff der Plattform, beziehungsweise der Plattformökonomie, ist im Zuge der Digitalisierung viel geschrieben worden. Fachspezifische Förderprogramme des Bundesministeriums für Wirtschaft und Energie (BMWi), wie beispielsweise *Smart Service Welt I* und *Smart Service Welt II* oder *PAiCE*,⁷ zeigen, wie stark digitale Plattformen in den Fokus von Wirtschaft und Wissenschaft gerückt sind. Grundsätzlich **nimmt durch die Digitalisierung die Bedeutung von Intermediären zu**, die als zentrales Bindeglied mehrere Gruppen und Systeme miteinander verbinden und durch automatisierte Abläufe und Datenaustausch Transaktionskosten verringern, wodurch große Skaleneffekte erzielt werden können. Plattformen können als solche Intermediäre genutzt werden. Vor allem im Business-to-Business-Bereich (B2B) konnten sie dieses Potenzial schon eindrucksvoll entfalten (s. DREWEL ET AL. 2017, S. 55; LEMKE ET AL. 2017, S. 424-425). Insbesondere in produzierenden Unternehmen werden digitale Plattformen eine wichtige Rolle für die Veränderung bestehender Kunden-Anbieter-Konstellationen und der Erschließung neuer Geschäftsmodelle einnehmen (s. WANGLER ET AL., 2017, S. III). Allerdings sind die Definitionen dieser Plattformen vielfältig und nur unscharf voneinander abgegrenzt.

Digitale Plattformen werden grundsätzlich definiert als „Produkte, Dienstleistungen oder Technologien, die als Basis für eine Vielzahl von Firmen dienen, um komplementäre Produkte, Dienste und Technologien anzubieten“ (BAUMS 2015, S. 15). Im Zuge dieser Arbeit soll diese Definition für Plattformen genutzt werden, wobei innerhalb der Domäne der digitalen Plattformen weiterhin eine Vielzahl von Begrifflichkeiten und Charakterisierung besteht:

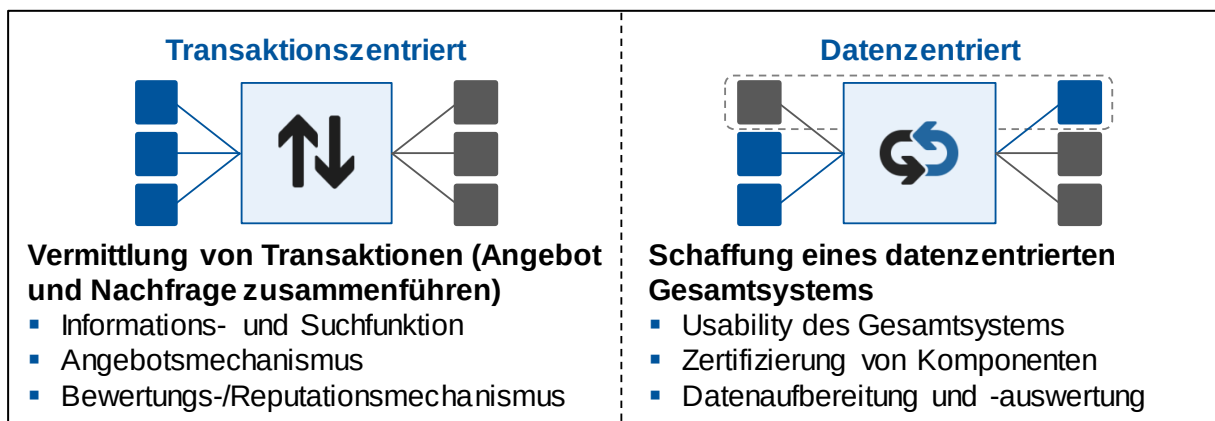


Abbildung 2-20: Idealtypen der transaktionszentrierten und der datenzentrierten Plattform (eigene Darstellung i. A. a. WANGLER ET AL., 2017, S. III-V)

⁷ Für weitere Informationen s. <http://www.digitale-technologien.de/DT/Navigation/DE/Foerderprogramme/foerderprogramme.html>.

Ein pragmatischer Ansatz sieht eine **Unterscheidung zwischen überbetrieblichen und innerbetrieblichen digitalen Plattformen** vor. WANGLER ET AL. führen eine Unterscheidung digitaler Plattformen in transaktionszentrierte und datenzentrierte Plattformen ein. Bei transaktionszentrierten Plattformen steht eine Vermittler- beziehungsweise Marktplatzfunktion im Vordergrund, während datenzentrierte Plattformen vornehmlich eine datenbasierte Vernetzung beziehungsweise die Schaffung eines datenzentrierten Gesamtsystems verfolgen (s. WANGLER ET AL., 2017, S. III-V). Innerbetriebliche digitale Plattformen sind meist datenzentriert umgesetzt.

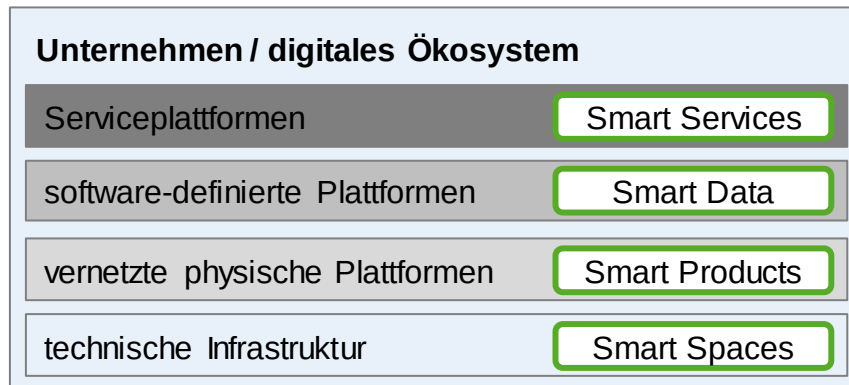


Abbildung 2-21: Schichtenmodell Digitale Infrastrukturen (i. A. a. ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2015, S. 17)

Die Deutsche Akademie der Technikwissenschaften (ACATECH) hat im *Arbeitskreis Smart Service Welt* ein **Schichtenmodell für digitale Infrastrukturen** entwickelt (s. Abbildung 2-21), welches vier verschiedene Schichten von Plattfortmtypen vorschlägt. Dabei wird der Begriff der digitalen Plattform noch um die Beschreibung einer Kollaborationsumgebung, die zum Beispiel ein Unternehmen sein kann, erweitert (vgl. ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2015).

Die unterste Schicht, als Grundlage digitaler Kollaborationsplattformen, bildet eine leistungsfähige und intelligente **technische Infrastruktur**. Darauf aufbauend verbinden **vernetzte physische Plattformen** intelligente Fabriken, Maschinen und Anlagen (sogenannte *Smart Products*) und deren digitale Abbilder mit- und untereinander. Die dritte Schicht der **software-definierten Plattform** bildet die technologische Integrationsschicht für heterogene Informationssysteme. Auf dieser werden die durch intelligente Maschinen erzeugten Daten zusammengeführt und verarbeitet. Mithilfe von Algorithmen und Analysen entsteht Smart Data (vgl. Kapitel 2.1.1). Über das Schlüsselprinzip der Virtualisierung wird eine geplante oder Ad-hoc-Zusammenarbeit smarter Produkte und Dienste ermöglicht. Die vierte Schicht „**Serviceplattformen** für digitale Ökosysteme“ bildet die betriebswirtschaftliche Integrationsschicht. Auf dieser werden den Plattformnutzern datenbasierte Smart Services angeboten. (s. ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2015, S. 16-17; FREY 2015, S. 32-36; BULLINGER ET AL. 2017, S. 100-103)

Im weiteren Verlauf dieser Arbeit soll das Schichtenmodell nach der ACATECH, das auch in einer im Jahr 2017 durchgeführten Studie der ACATECH bestätigt wurde (s. ENGELS ET AL. 2017, S. 21-22), zur Einordnung von Plattformen genutzt werden.

Von besonderer Relevanz ist die Schicht der software-definierten Plattform: **Die Vision des zentralen Zugriffs auf alle Unternehmensdaten kann über diesen Plattform-Typ realisiert werden.** Definiert werden diese als „[...] die technologische Basis für neuartige, internetbasierte Dienstleistungen. Sie können durch Virtualisierung die Zusammenarbeit unterschiedlicher Vernetzter physischer Plattformen sicherstellen und die Konnektivität einzelner Smart Products bereitstellen.“ (FREY 2015, S. 34)

Auf Basis des Schichtenmodells nach der ACATECH, dem Konzept von *MindSphere* - *Siemens Cloud for Industry* sowie Anforderungen an agile Unternehmen wurde der Aachener Ansatz der Infrastruktur,⁸ das sogenannte „*Internet of Production*“ für agile Unternehmen, entwickelt (s. Abbildung 2-22).

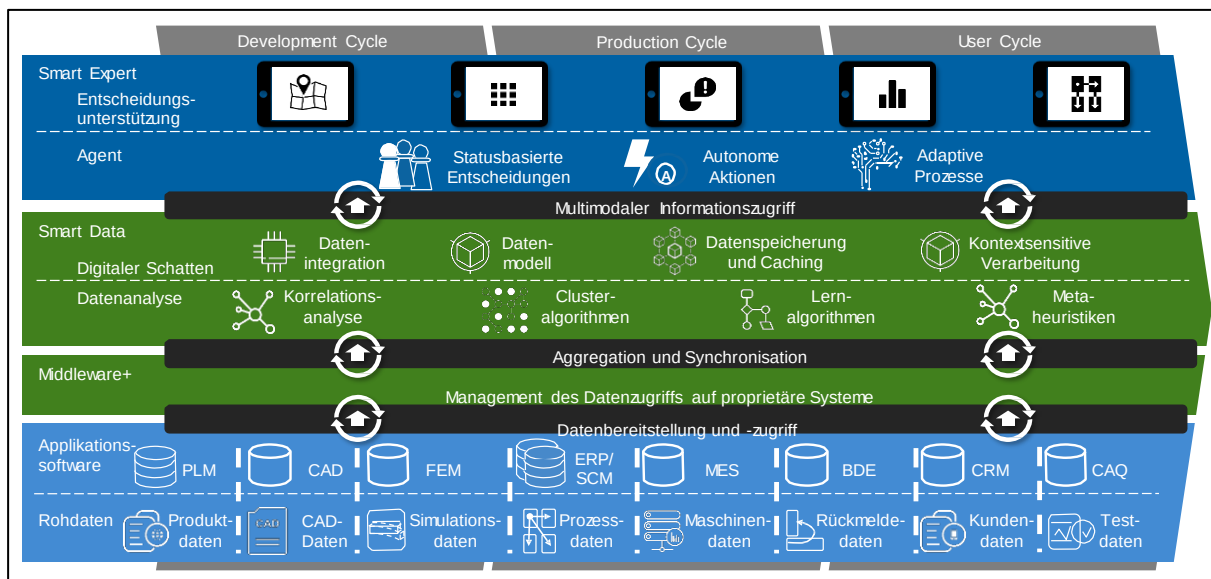


Abbildung 2-22: Infrastruktur des *Internet of Production* (entnommen aus SCHUH ET AL. 2017h, S. 121)

Das *Internet of Production* befähigt Unternehmen zur schnellen Initiierung fundierter Entscheidungen und Änderungsvorgänge (s. KLOCKE ET AL. 2017a, S. 289). Von links nach rechts werden in Abbildung 2-22 aggregiert drei Produktlebenszyklusphasen, von der Prozessentwicklung bis zum Einsatz des Produkts beim Kunden, dargestellt. Die Ebenen von unten nach oben stellen die erhobenen Daten, auf diese Daten angewandte Algorithmen und die eingesetzten Assistenzsysteme und Apps dar. Die **aus unterschiedlichen Quellen entstehenden Rohdaten** halten die digitalisierten Prozesszustände fest und dienen als Basis für weitere Auswertungen und Analysen. Die **Middleware+** stellt diese Rohdaten für die *Applikationssoftware* bereit, durch Analysealgorithmen entsteht **Smart Data** (s. Kapitel 2.1.1). Die für den Anwendungsfall relevante Daten aus den unterschiedlichen, proprietären Applikationen und die Rohdaten werden verknüpft und bilden die **Datengrundlage für digitale Schatten** (vgl. STICH ET AL. 2018). Der Begriff *Digitaler Schatten* beschreibt das relevante Abbild der Realität

⁸ Für weitere Informationen zu Siemens MindSphere s. <https://www.siemens.com/global/en/home/products/software/mindsphere.html>.

in Echtzeit. Als Ergebnis der Datenintegration entstehen Datenmodelle, die als digitaler Schatten gespeichert und kontextabhängig den *Smart Experts* bereitgestellt werden. Über Apps werden die relevanten Informationen zur Entscheidungsfindung bereitgestellt oder über *Agenten* die Entscheidungen automatisch abgewickelt. Als **App** werden in der vorliegenden Arbeit Informationssysteme verstanden, die einen klar umrissenen Funktionsumfang haben und für einen bestimmten Anwendungszweck geschaffen sind. Sie zeichnen sich durch intuitive Benutzerführung und Variabilität der Nutzung auf verschiedensten Endgeräten aus (s. LEMKE ET AL. 2017, S. 374). Eine App benötigt immer eine bestimmte Betriebssystemgebung oder Plattform, welche ihr als Datenlieferant und Laufzeitumgebung zur Verfügung steht. **Getroffene Entscheidungen werden in die Ebene der Smart Data zurückgespielt**, die wiederum durch eine Synchronisation mit den Rohdaten zurück in die *Applikationssoftware* gespeist werden. Diese kontinuierliche Synchronisation schafft eine einheitliche Datenbasis und verhindert das Entstehen von zueinander in Konflikt stehenden Datensätzen (s. SCHUH ET AL. 2017h, S. 120-121; KLOCKE ET AL. 2017b, S. 266-268; KLOCKE ET AL. 2017a, S. 289).

Die Ebenen *Smart Data* und *Middleware+* stellen eine software-definierte Plattform nach der ACATECH dar, die die Vision des zentralen Zugriffs auf alle Unternehmensdaten durch eine Vernetzung der im Unternehmen genutzten Informationssysteme (s. Kapitel 2.1.6) realisiert. Eine darauf aufbauende eigene Definition des Begriffs *Software-definierte Plattform* wird in Kapitel 5.1.2 vorgestellt.

2.4 Abgrenzung und Präzisierung des Untersuchungsbereichs

Eine Forschungsarbeit muss, um ein Thema fokussieren zu können, Teile der Realität ausgrenzen (s. MALONE U. CROWSTON 1994, S. 100). Dies entspricht auch dem Ansatz der Systemtheorie, auf dem diese Arbeit basiert (s. Kapitel 1.4 sowie 4.2.1). Deshalb dient dieses Kapitel dazu, auf Basis der in den vorherigen Unterkapiteln festgelegten Begriffe und Themenbereiche den Untersuchungsbereich der vorliegenden Arbeit einzugrenzen und zu präzisieren. In Tabelle 2-4 wird diese Abgrenzung in einer morphologischen Darstellung zusammengefasst, indem die für diese Arbeit relevanten Bereiche hervorgehoben sind.

Der **Modellierungszweck**, der in der vorliegenden Arbeit verfolgt wird, ist die Beschreibung von software-definierten Plattformen sowie der Informationssystem-Architektur produzierender Unternehmen; es werden Zusammenhänge zwischen diesen beiden erklärt sowie ein Gestaltungsmodell für die Einführung von software-definierten Plattformen entwickelt (vgl. dazu Kapitel 4.2.2 und LEHNER ET AL. 2008, S. 29ff.). Das zu entwickelnde Modell wird bestimmte **Schichten** der Informationssystem-Architektur betrachten (s. Kapitel 2.1.3). Nimmt man dazu die „Layers“ des RAMI4.0 als Vorlage, sind dies die Schichten *Functional* und *Information*, sowie teilweise die Schichten *Communication* und *Integration*, soweit hier Schnittstellen beschrieben werden (s. DIN SPEC 91345, S. 20-22). Ziel der Dissertationsschrift ist es dann, die software-definierte **Plattform** in die Informationssystem-Architektur einzuführen. Diese grenzt sich

nach Definition des ARBEITSKREISES SMART SERVICE WELT nach oben zur *Serviceplattform*, die zur Vernetzung von Anbietern zur Bereitstellung von Dienstleistungen genutzt wird, sowie nach unten zur *vernetzten physischen Plattform* ab (s. ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2015, S. 16-17). Die entwickelten Modelle werden entsprechend der Forschungskonzeption (s. Kapitel 1.3) in zwei Fallstudien sowie durch die reale Nutzung bei einem Unternehmen auf dem RWTH Aachen Campus (vgl. STICH ET AL. 2018) **evaluiert**, um dem Anspruch an die Qualität der Modelle, wie sie von FETTKE U. LOOS formuliert werden, gerecht zu werden (s. FETTKE U. LOOS 2004, S. 1-2).

Tabelle 2-4: Präzisierung des Untersuchungsbereichs (eigene Darstellung)

	Merkmal	Ausprägungen					
Modellentwicklung	Modellierungszweck (s. LEHNER ET AL. 2008)	Beschreibung	Erklärung	Gestaltung	Meta-Modelle	sonstige Modelle	
	Schichten (s. DIN SPEC 91345)	Business	Functional	Information	Communication	Integration	Asset
	Plattformen (s. ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2015)	Service-Plattform	software-definierte Plattform	vernetzte physische Plattform	technische Infrastruktur		
	Evaluation (s. FETTKE U. LOOS 2004)	formal		Fallstudien		reale Nutzung	
Anwendung	Anwendungsdomäne (s. FETTKE U. LOOS 2004)	Produzierende Unternehmen		Dienstleistungen		sonstige	
	Informationsmanagement (IM)-Aufgaben (s. KRCMAR 2015)	Führungsaufgaben des IM	Management der Informationswirtschaft		Management der Informationssysteme		Management der IKT
	Unternehmensgröße (s. EUROPÄISCHE KOMMISSION)	Mikrounternehmen		KMU		Großunternehmen	

Legende

behandelt

teilweise behandelt

nicht behandelt

Die **Anwendung** der Ergebnisse wird primär in der Domäne der produzierenden Unternehmen sein (s. Kapitel 2.1.6), wenngleich die Herausforderungen des Informationsmanagements in anderen Industrien ähnlich und daher die Modelle zumindest teilweise übertragbar sind. Von den beschriebenen **Aufgaben des Informationsmanagements** können, in der Definition von KRCMAR, vor allem die *Führungsaufgaben* und

Aufgaben des *Managements der Informationstechnik* von den entwickelten Ergebnissen profitieren (s. dazu Kapitel 2.2.1 u. vgl. KRCMAR 2015, S. 107). Die **Unternehmensgröße** ist dabei prinzipiell irrelevant, allerdings sind die Ergebnisse in Mikrounternehmen, die keine eigenen Verantwortlichen für das Informationsmanagement haben, nur bedingt übertragbar. Dasselbe gilt für Großunternehmen, die aufgrund hoher Spezialisierung der Fachkräfte in den IT-Abteilungen noch weitergehende Herausforderungen betrachten müssen.

3 Stand der Erkenntnisse

Das zentrale Ziel der vorliegenden Arbeit wird in Kapitel 1.2 vorgestellt: Methoden zu entwickeln, um zentrale software-definierte Plattformen in bestehende Informationssystem-Architekturen zu integrieren. Um dies zu erreichen, verfolgt der Autor die **Lösungshypothese**, dass **durch Anpassung der Informationssystem-Architektur erst der volle Funktionsumfang von software-definierten Plattformen erreicht werden kann**. Für die notwendigen Veränderungen der Informationssystem-Architektur **gibt das Forschungsgebiet des IT-Komplexitätsmanagements Handlungsempfehlungen**. Deshalb wurde im Rahmen der Vorbereitungen für die vorliegende Arbeit eine umfangreiche Literaturanalyse zum Stand der Forschung über IT-Komplexität durchgeführt. Diese zeigt auf, dass kein gut für Unternehmen anwendbares und gleichzeitig umfangreiches Modell zum Umgang mit IT-Komplexität besteht, jedoch einige Methoden sinnvoll verwendet werden können (s. Kapitel 7.1). Die Ergebnisse der Literaturanalyse wurden vom Autor mit anderen Autoren in einer Vorveröffentlichung publiziert (vgl. SCHUH ET AL. 2017e, S. 61-65).

Um gegenüber dem beschriebenen Ziel den aktuellen Stand der Erkenntnisse vorzustellen, werden im vorliegenden Kapitel zuerst vorhandene Modelle zur **Beschreibung von software-definierten Plattformen** vorgestellt (s. Kapitel 3.1), gefolgt von **Integrationskonzepten** für solche Plattformen (s. Kapitel 3.2). Demgegenüber muss die bestehende Informationssystem-Architektur produzierender Unternehmen beschrieben werden, weshalb in Kapitel 3.3 vorhandene **Beschreibungsmodelle für Informationssystem-Architekturen** dargestellt werden. Neben der reinen Beschreibung wird außerdem dargelegt, wie andere Autoren eine **zielgerichtete Informationssystem-Architektur** sehen, die Grundvoraussetzung zur effizienten Nutzung einer software-definierten Plattform ist (s. Kapitel 3.4). Das Kapitel schließt mit Auszügen der angesprochenen **Literaturanalyse zum IT-Komplexitätsmanagement** (s. Kapitel 3.5), welche Bausteine für das in Kapitel 7 vorgestellte Erklärungsmodell liefert. Eine Zusammenfassung in Kapitel 3.6 gibt einen grafischen Überblick über die analysierte Literatur.

3.1 Beschreibung von software-definierten Plattformen

In Wissenschaft und Praxis gibt es verschiedene Plattform-Konzepte, die sich den digitalen Plattformen im Allgemeinen sowie software-definierten Plattformen im Besonderen zuordnen lassen. Diese werden im Folgenden kurz vorgestellt.

Sehr allgemein führt IBM den Begriff *Software Defined Environment* (dt. software-definierte Umgebung) ein. Er bezeichnet eine **Abstraktion von der Hardwareeschicht durch ein Verschieben technologischer Funktionen in eine virtualisierte Infrastruktur**. Software-definierte Umgebungen führen software-definierte Rechenoperationen, Netzwerke und Speicherung zusammen und vereinen die Kontrolle und Steuerung für individuelle software-definierte Komponenten (s. LI ET AL. 2014, S. 2; QUINTERO ET AL. 2015, S. 3-5).

Eine Ausprägung einer software-definierten Umgebung ist der durch die ACATECH eingeführte Begriff der **software-definierten Plattform** im Rahmen des Schichtenmodells digitaler Infrastruktur (s. Abbildung 2-21). Zwar herrscht weder in Wissenschaft noch Praxis ein Konsens über die Definitionen und Abgrenzungen zwischen verschiedenen digitalen Plattformen (s. ENGELS ET AL. 2017, S. 13), allerdings wird das Schichtenmodell der ACATECH auch von anderen Veröffentlichungen aufgegriffen. In im Rahmen der Begleitforschung zur *Smart Service Welt* veröffentlichten Werken (vgl. ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2016) wird das Modell genutzt. FREY, CASTILLO ARAUJO, WEIBER ET AL. und BULLINGER ET AL. nutzen ebenfalls das Verständnis der ACATECH und stellen gleichzeitig die einzigen Werke außerhalb der Smart-Service-Welt dar, die ebenfalls den Begriff der software-definierten Plattform nutzen (vgl. FREY 2015; CASTILLO ARAUJO 2015; WEIBER ET AL. 2017; BULLINGER ET AL. 2017).⁹

Software-definierte Plattformen erlangen über **generische Enabler** Daten aus der Schicht der vernetzten physischen Plattform (s. Abbildung 3-1). Beispielhafte Bausteine können eine ubiquäre Benutzermodellierung, Big-Data-Mining und Streaming-Analytics oder aktive digitale Objektgedächtnisse sein. Über Architekturmuster werden auf der Plattform **Leistungsbündel entwickelt**, die mittels einer Serviceplattform bereitgestellt werden.

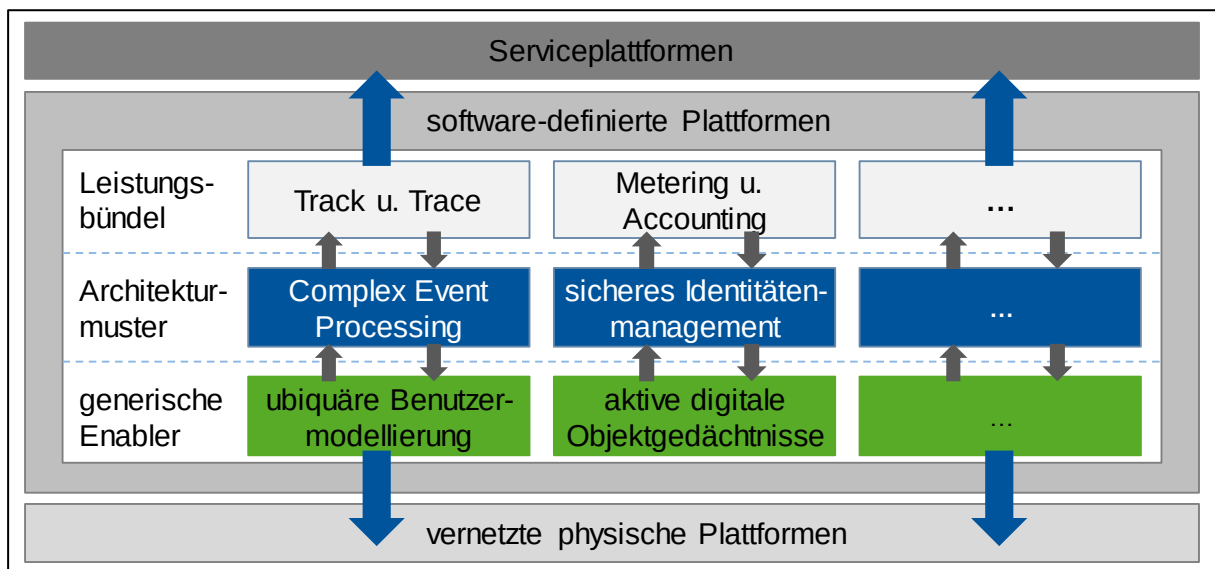


Abbildung 3-1: Leistungsbündel, Architekturmuster und generische Enabler innerhalb einer software-definierten Plattform (eigene Darstellung i. A. a. ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2015, S. 72)

PORTER U. HEPPELMANN beschreiben in ihrem vielzitierten Artikel „How Smart, Connected Products Are Transforming Competition“ den **New Technology Stack**. Die Auto-

⁹ Eine Literaturrecherche nach den Begriffen „software-definierte Plattform“ beziehungsweise „software defined platform“ ergibt, neben eigenen Werken des Autors und Werken zu Netzwerk-Plattformen, zum Stand 20. April 2018, nur diese wissenschaftlichen Werke als Ergebnis einer Suche auf *IEEE Xplore* und *Google Scholar*.

ren führen auf, dass vernetzte Produkte eine vollständig neue technologische Infrastruktur (*technology stack*) benötigen. Dafür beschreiben sie die grundsätzlich notwendige Architektur. Zentraler Baustein ist darin die *Product Cloud* (s. Abbildung 3-2). Innerhalb dieser werden eine Produktdaten-Datenbank, eine Plattform zur Applikationsentwicklung, eine *Rules Engine* und Analyseplattform sowie produktbezogene, nicht direkt in das Produkt integrierte, Applikationen abgebildet. Die *Product Cloud* stellt ebenfalls eine Software-definierte Plattform dar (s. PORTER U. HEPPELMANN 2014, S. 68-69).

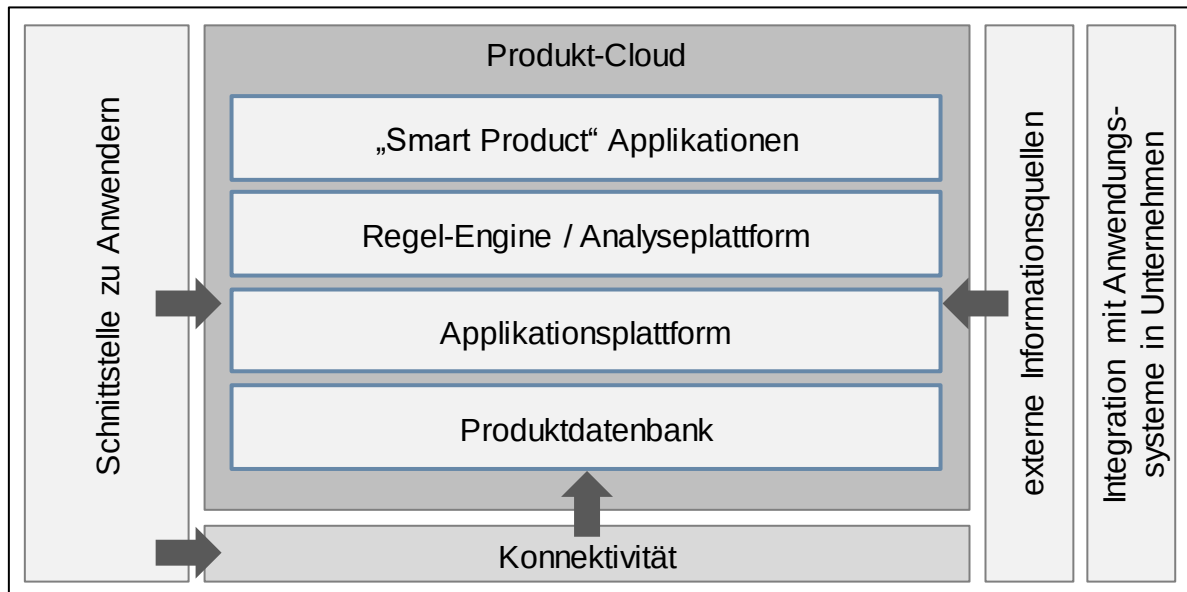


Abbildung 3-2: Architektur der *Product Cloud* innerhalb des *New Technology Stack* (eigene Darstellung u. Übersetzung i. A. a. PORTER U. HEPPELMANN 2014, S. 69)

Das **Kompodium Industrie 4.0** leistet einen Beitrag, um Industrie 4.0 und den damit einhergehenden Transformationsprozess zu beschreiben und zu verstehen, und gibt Politik und Wirtschaft Hilfestellungen an die Hand. Unter anderem werden dabei die **zentralen Elemente einer digitalen Plattform** beschrieben: Dateninput, Datenaggregation und Datenverwertung (s. Abbildung 3-3). In der Ebene des Dateninputs werden Daten aus heterogenen Quellen auf einer zentralen Plattform gesammelt. Die Daten kommen beispielsweise aus unternehmensinternen Informationssystemen (z. B. ERP-, CRM-System), aus installierter Sensorik, von weiteren Endgeräten oder auch aus externen Quellen, beispielsweise Wetterdaten. Auf der Ebene der Datenaggregation werden die heterogenen Daten gesammelt und zu Informationen veredelt. Hierfür sind bestimmte Regeln und Bausteine, beispielsweise durch APIs, zur Datenverwertung notwendig. Die Ebene der Datenverwertung stellt die gewonnenen Erkenntnisse als Services bereit oder ermöglicht eine Vermarktung ebendieser (s. BAUMS 2015, S. 16).

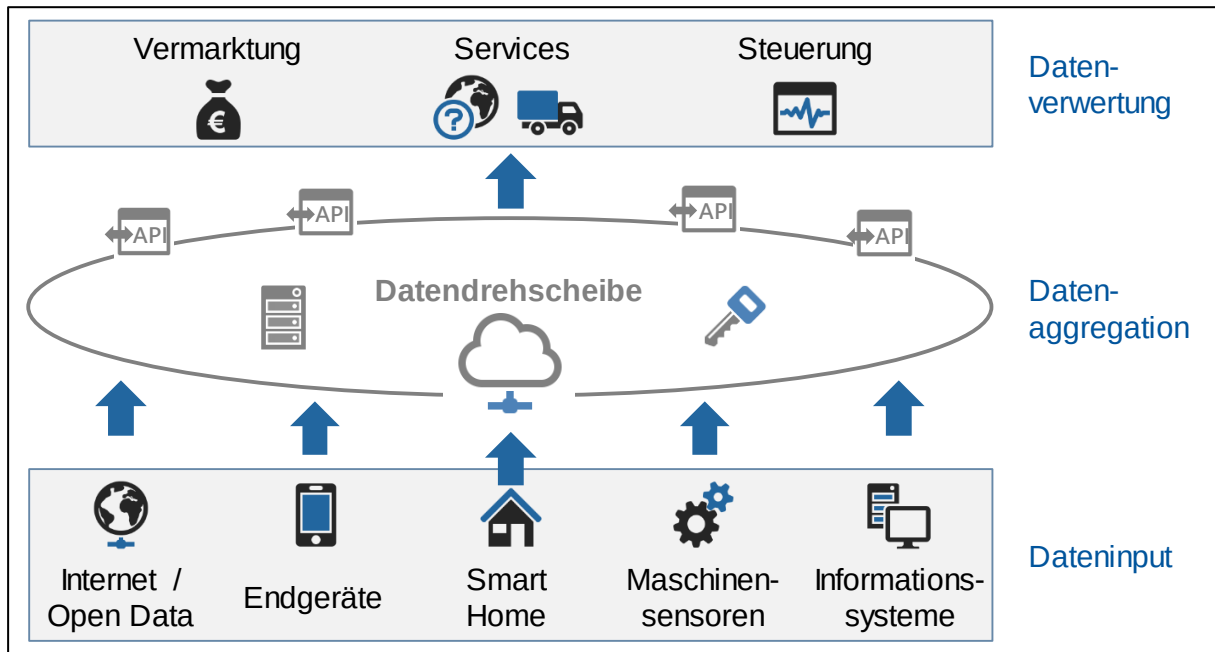


Abbildung 3-3: Technologieelemente digitaler Plattformen (i. A. a. BAUMS 2015, S. 17)

Ein weiteres Konzept, das sich den software-definierten Plattformen zuordnen lässt, ist das der sogenannten **IoT-Plattformen** beziehungsweise IoT-Softwareplattformen. Grundsätzlich vereinfachen IoT-Plattformen die Vernetzung von Produkten und Prozessen, indem Geräte und Infrastruktur sowie Anwendungssysteme in Unternehmen über die Plattform miteinander vernetzt werden (s. Abbildung 3-4). Die durch IoT-Plattformen umgesetzten Funktionen lassen sich in die fünf Kategorien *Verbinden*, *Sichern*, *Verwalten*, *Analysieren* und *Erstellen* gruppieren. *Analysieren* bedeutet dabei ein Umwandeln von Daten in zeitnahe, relevante Erkenntnisse und Maßnahmen, wohingegen *Erstellen* das Erstellen von Anwendungen und eine Integration in Anwendungssysteme in Unternehmen beschreibt (s. PELINO U. HEWITT 2016, S. 3-5). Eine Liste von IoT-Plattformen findet sich bei MINERAUD ET AL. (s. MINERAUD ET AL. 2015), oder für den deutschen Markt bei KRAUSE ET AL. (s. KRAUSE ET AL. 2017); im Folgenden werden einige hierzu relevante Konzepte beschrieben.

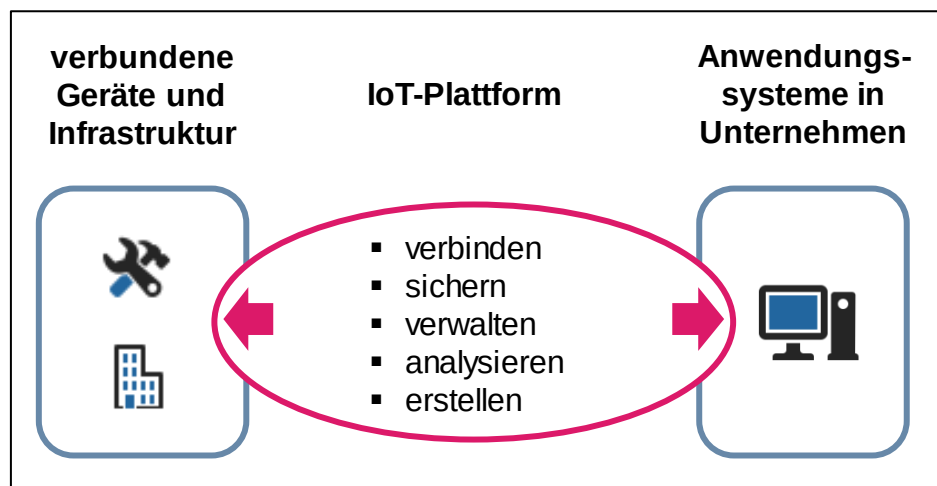


Abbildung 3-4: Funktionen einer IoT-Plattform (i. A. a. PELINO U. HEWITT 2016, S. 5)

GUTH ET AL. entwickeln auf der Basis verschiedener State-of-the-Art IoT-Plattformen eine **Referenzarchitektur für das IoT**. Diese ist der eingeführten Beschreibung durch BAUMS grundsätzlich ähnlich und enthält ähnliche Elemente wie die Beschreibung von Plattformen der ACATECH. Es gibt vielzählige weitere Referenzarchitekturen für das IoT (vgl. bspw. NETTSTRÄTER 2012, GREEN 2014, KUBACH 2016 oder SCULLY 2016). Allerdings sind die enthaltenen Bausteine zu großen Teilen dieselben. Die Architektur nach GUTH ET AL. spiegelt die wichtigsten Module und relevanten Funktionen wider (s. Abbildung 3-5).

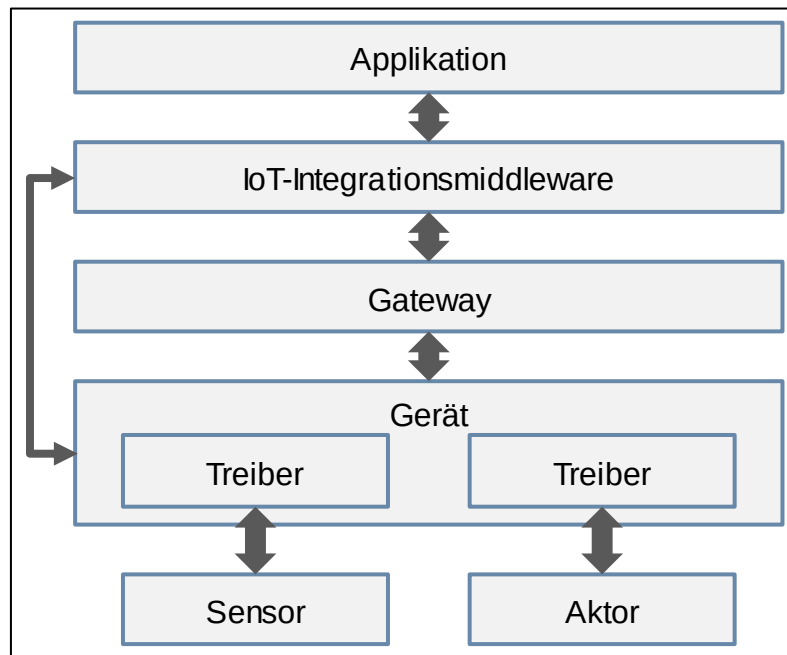


Abbildung 3-5: IoT-Referenzarchitektur (i. A. a. GUTH ET AL. 2016, S. 2)

KRAUSE ET AL. analysieren in einer Marktstudie aus dem Jahr 2017 eine Vielzahl deutschsprachiger IoT-Plattformen. Um einen standardisierten Vergleich zu ermöglichen, entwickeln die Autoren ebenfalls ein **Referenzmodell für IoT-Plattformen** (s. Abbildung 3-6). Das Referenzmodell setzt sich insgesamt aus zehn Bereichen mit unterschiedlichen Teilfunktionen und -aufgaben zusammen und beinhaltet im Vergleich zu den bisher genannten Modellen zusätzliche Querschnittsbereiche für Geschäftsprozesse und Geschäftsmodelle (s. KRAUSE ET AL. 2017, S. 12).

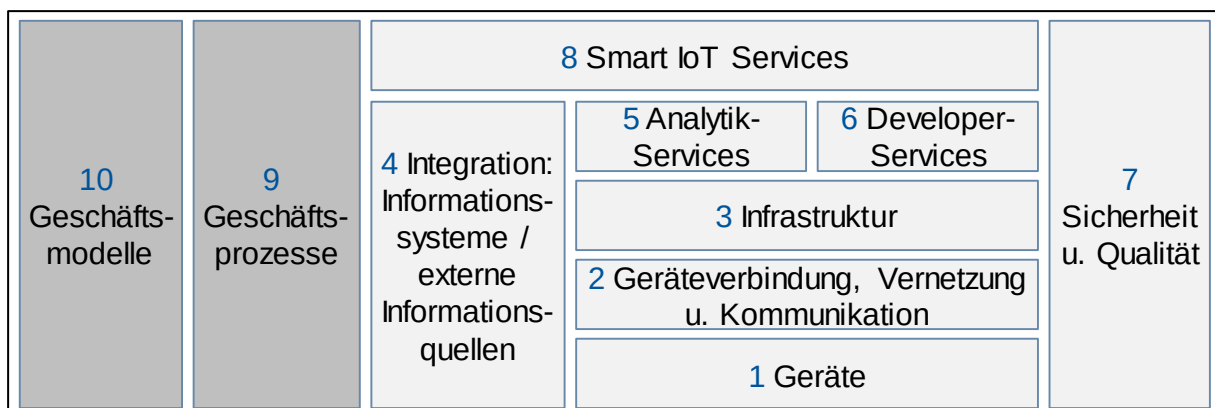


Abbildung 3-6: Referenzmodell für IoT-Plattformen (i. A. a. KRAUSE ET AL. 2017, S. 12)

Zusammenfassend lässt sich festhalten, dass nur wenige Werke den Begriff der software-definierten Plattform aufgreifen. In der vorliegenden Arbeit wird daher in Kapitel 5.1 eine eigene Definition des Begriffs hergeleitet. Die Architektur software-definierter Plattformen findet sich in der Beschreibung weiterer Plattformen, insbesondere bei IoT-Plattformen, wieder. Diese Plattformen werden daher in Kapitel 5 einer vertieften Betrachtung unterzogen.

3.2 Plattform-Integrationskonzepte

Innerhalb betrieblicher Anwendungssysteme lässt sich eine starke Entwicklung in Richtung modularisierter Architekturkonzepte beobachten. Allerdings ist die bilaterale Kopplung von Modulen gerade in großen Unternehmen mit mehreren hundert oder sogar tausend Anwendungssystemen mit einem erheblichen Integrationsaufwand verbunden. Daher hat sich eine Integrationsschicht herausgebildet, die eigene Anwendungssysteme zur zentralisierten Verwaltung von Schnittstellen, zur Transaktionsabwicklung über mehrere verteilte (operative) Anwendungssysteme hinweg sowie zur Bereitstellung zentralisierter Dienste (z. B. Verzeichnisse) abbildet.

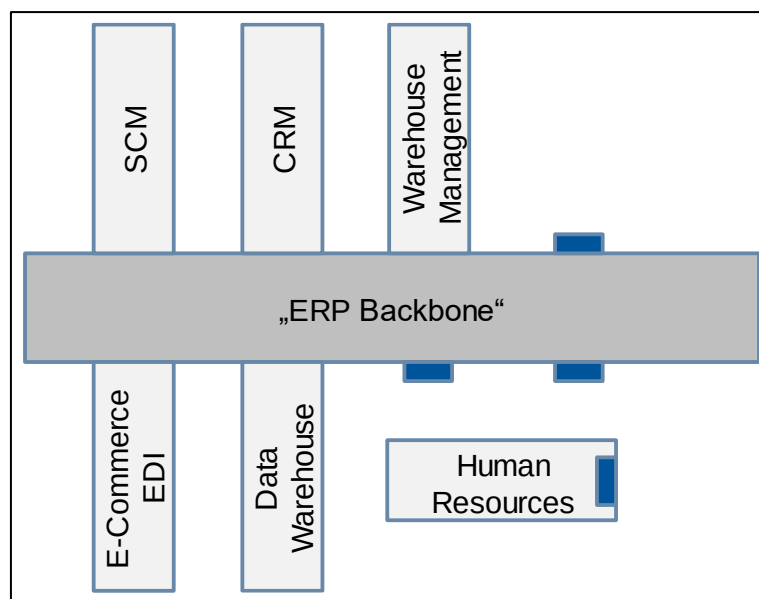


Abbildung 3-7: ERP-basierte Integration (i. A. a. KAIB 2002, S. 70)

Vor allem durch die Einführung von ERP-Software in produzierenden Unternehmen ist eine Art der Plattform-Integration entstanden (s. Abbildung 3-7). Dabei dient das ERP-System als „backbone“ beziehungsweise zentrale Integrationsplattform, auf dem weitere Einzelanwendungen (CRM-System, SCM-System etc.) für den spezifischen Bedarf über Standardschnittstellen aufgesetzt werden (s. KAIB 2002, S. 70).

Am bekanntesten sind in diesem Zusammenhang **Middleware- und EAI-Systeme**, die auf den Prinzipien einer serviceorientierten Architektur (SOA) aufbauen. Diese folgt dem Prinzip, wonach sich Komponenten mit festgelegter Funktionalität und Schnittstelle entkoppeln und wiederverwenden lassen (ALPAR ET AL. 2016, S. 169-171). Bei einer **Middleware-basierten Integration** wird eine vermittelnde Softwareschicht – die Middleware – zwischen mehrere Systeme geschaltet, sodass Konnektivität zwischen

verschiedenen Anwendungen geschaffen wird und die angebundenen Anwendungen plattformunabhängig Daten austauschen können (s. KAIB 2002, S. 72-73). EAI baut ebenfalls auf einer Middleware-basierten Integration auf, allerdings lassen sich mithilfe von EAI nur sehr umständlich Plattformen in die Informationssystem-Architektur produzierender Unternehmen integrieren. Denn bei EAI werden alle Unternehmenssysteme und -anwendungen gleichberechtigt über eine zentrale EAI-Komponente, eine Art „**EAI-Informationsdrehschreibe**“, verbunden (s. KAIB 2002, S. 81-82). Dies würde eine spezifische Anpassung der Applikationen an die jeweiligen Integrationsschnittstellen und insgesamt eine schwerfällige Architektur bedingen (s. BOSSERT U. LAARTZ 2017, S. 30). Demgegenüber soll durch den Einsatz von Plattformen eine Ebene der zentralen Datennutzung geschaffen werden, die als Grundlage zur Entscheidungsunterstützung und -automatisierung dient (s. STICH ET AL. 2018, S. 27-28).

Ebenfalls auf einer SOA basiert der **Manufacturing-Service-Bus** (MSB) als eine Ausprägung des sogenannten Enterprise-Service-Bus (ESB). Dabei handelt es sich um eine Integrationsplattform, die eine flexible Verknüpfung unterschiedlichster produktionsbasierter IT-Services erlaubt (s. BAUERNHANSLE ET AL. 2016, S. 21). Neben dem Prinzip der SOA wird das Prinzip der Anwendungsintegration genutzt. Die Verbindung zwischen Produktionsanwendungen wird über den MSB realisiert, der eine **Kommunikationsinfrastruktur zum diensteübergreifenden Nachrichtenaustausch** bildet (s. MINGUEZ 2013, S. 276-279). Somit können Interoperabilitäts- und Schnittstellenkompatibilitätsprobleme gelöst werden. Der MSB kommt beispielsweise bei der Plattform *Virtual Fort Knox* (s. Kapitel 3.4) zum Einsatz. Allerdings ist eine kontinuierliche Adaption und Anpassung des MSB-Integrationsprozesses notwendig; mit zunehmender Größe werden Bussysteme daher schwerfällig (s. BOSSERT U. LAARTZ 2017, S. 31).

Die Dissertationsschrift von DERN liefert ein Konzept zum **ganzheitlichen Integrationsmanagement** in der Unternehmens-IT. Zu diesem Zweck gibt der Autor vor allem für größere Unternehmen Empfehlungen zur Gestaltung der Unternehmens-IT. Neben den Gestaltungsprinzipien und Empfehlungen besteht das Konzept aus sogenannten eingebetteten Artefakten, die Definitionen, Modelle und Methoden umfassen. Diese basieren auf einem eigens entwickelten Grundmodell zum Integrationsmanagement. In diesem Zusammenhang definiert der Autor Integrationsleistungen als durch die IT-Organisation erbrachte Leistung, um aus Leistungsanforderungen an die Unternehmens-IT abgeleitete Integrationsanforderungen zu erfüllen (s. DERN 2011, S. 56). Das Grundmodell betrachtet ein Unternehmen ganzheitlich und liefert einen Ansatz zur Überführung von internen und externen Anforderungen in Integrationsleistungen innerhalb einer Unternehmens-IT. Innerhalb des Konzepts wird insbesondere der Integrationsgegenstand, also die zu integrierenden Elemente, definiert. Jener ist daher grundsätzlich unbeschränkt (s. DERN 2011, S. 50-73).

FREY entwickelt in seiner Dissertationsschrift ein **Integrationskonzept für Smart Services** innerhalb intelligenter Umgebungen am Beispiel von Smart-Homes. Dieses basiert auf der Referenzarchitektur der Smart-Service-Welt und nutzt Ansätze und Konzepte des Industriestandards *Universal Remote Console* sowie des SOA-Ansatzes.

Mithilfe des Konzepts entwickelt FREY eine auf einem **Multiagentensystem aufbauende Middleware-Plattform** für intelligente Umgebungen zur dienstübergreifenden Integration von Smart Services, eine sogenannte agentenbasierte Smart-Service-Plattform (ASaP). Diese Plattform ordnet er auch in die Architektur der Smart-Service-Welt ein, wie Abbildung 3-8 zeigt. (s. FREY 2015, S. 121-137)

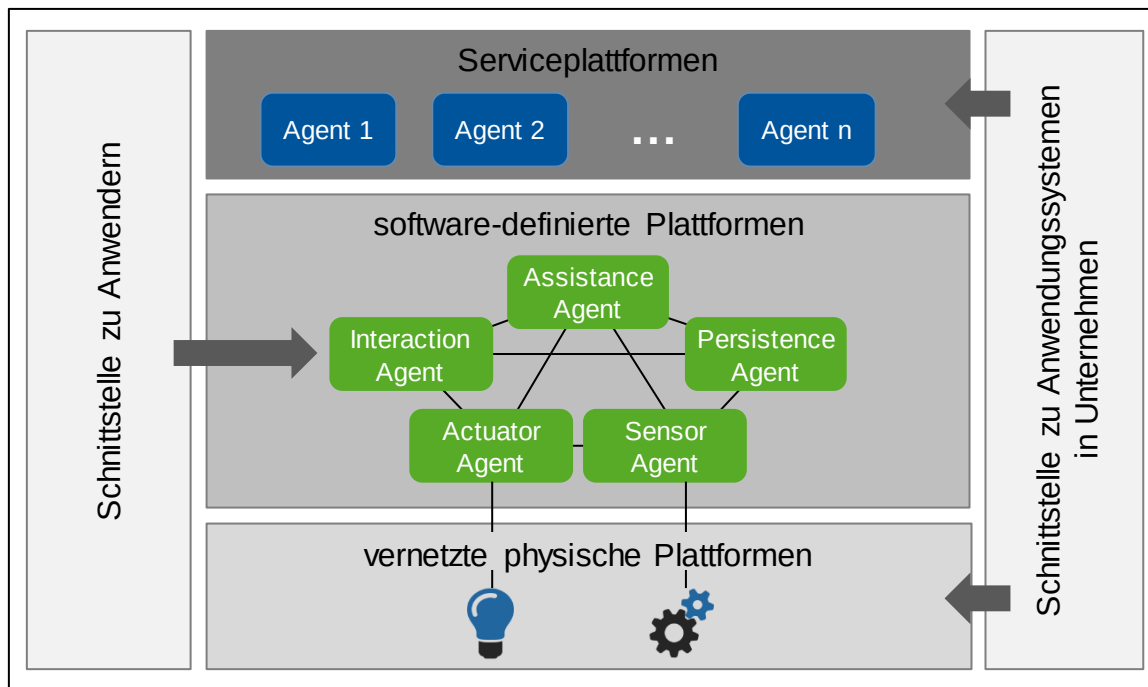


Abbildung 3-8: Einordnung von FREYS Plattformkonzept in die Architektur der Smart-Service-Welt (eigene Darstellung i. A. a. FREY 2015, S. 135)

Durch die zunehmende Entwicklung von Cloud-Lösungen sind erste Konzepte zur Integration veralteter Informationssysteme mit Cloud-Anwendungen entstanden. Das Prinzip der **Integration Platform as a Service (iPaaS)** vereint Anwendungs- und Datenintegration und liefert eine Cloud-Integrationsplattform. Entsprechende Plattformen ermöglichen die Entwicklung, Ausführung und Steuerung von Integrationsworkflows für lokale (On-Premises) und cloudbasierte Anwendungen. Zu diesem Zweck liefern iPaaS-Lösungen typischerweise vorgefertigte Konnektoren und anwendergeeignete Werkzeuge für die Entwicklung und Konfiguration. Der Fokus liegt insbesondere auf der Integration von Cloud-Lösungen in eine bestehende Infrastruktur (bzw. vice versa). Außerdem gibt es häufig API-Frameworks, um Schnittstellen festzulegen, zu erstellen und zu nutzen. Auf diese Weise können sogenannte „**Integrationflows**“ zur **Verbindung einer beliebigen Kombination von lokalen und cloudbasierten Prozessen**, Services, Applikationen und Daten innerhalb oder zwischen Organisationseinheiten realisiert werden (s. RICHERT 2015; INFORMATICA 2017). Auf diese Weise kann eine Entkopplung von Diensten und Funktionen innerhalb der Unternehmensarchitektur realisiert und auf einer zentralen Plattform umgesetzt werden (s. BOSSERT U. LAARTZ 2017, S. 29-31).

Neben den bestehenden Konzepten zur Plattform-Integration gibt es verschiedene **Frameworks für einzelne Disziplinen der Integration**. Diese unterstützen vor allem

Entwickler in der Definition der Integrationsanforderungen sowie der Durchführung der Integration der jeweiligen Teildisziplinen. Bestehende Frameworks liefern beispielsweise Richtlinien zur Integration von Geräten oder Informationssystemen (s. KUBER ET AL. 2016, S. 28).

Die Herausforderungen bei der Umsetzung der einzelnen Plattform-Integrationskonzepte sind, wie oben beschrieben, vor allem auf deren geringe Flexibilität zurückzuführen. Trotzdem können die **bestehenden Konzepte in der vorliegenden Arbeit als Grundlage verwendet** werden, um darauf aufbauend in Kapitel 8 darzustellen, wie eine Plattformintegration in produzierenden Unternehmen umgesetzt werden kann.

3.3 Beschreibung von Informationssystem-Architekturen

Neben den in Abschnitt 2.1.5 beschriebenen Funktionen von Informationssystem-Architekturen können Architekturmodelle unter anderem als Entscheidungsgrundlage genutzt werden, um den Informationsbedarf verschiedener Anspruchsgruppen zu bestimmen beziehungsweise zu erfüllen (s. BUCHER ET AL. 2006). So hilft die Informationssystem-Architektur bei der Konstruktion und Evolution der IT-Unterstützung von fachlichen Anforderungen. Somit können, ausgehend von einer Beschreibung der Informationssystem-Architektur, notwendige Schnittstellen für eine zentrale Plattform identifiziert werden. Im Folgenden werden, aufbauend auf den in Kapitel 2.1.5 definierten Grundlagen, **relevante Ansätze zur Beschreibung von Informationssystem-Architekturen** vorgestellt, die im weiteren Verlauf dieser Arbeit Anwendung finden.

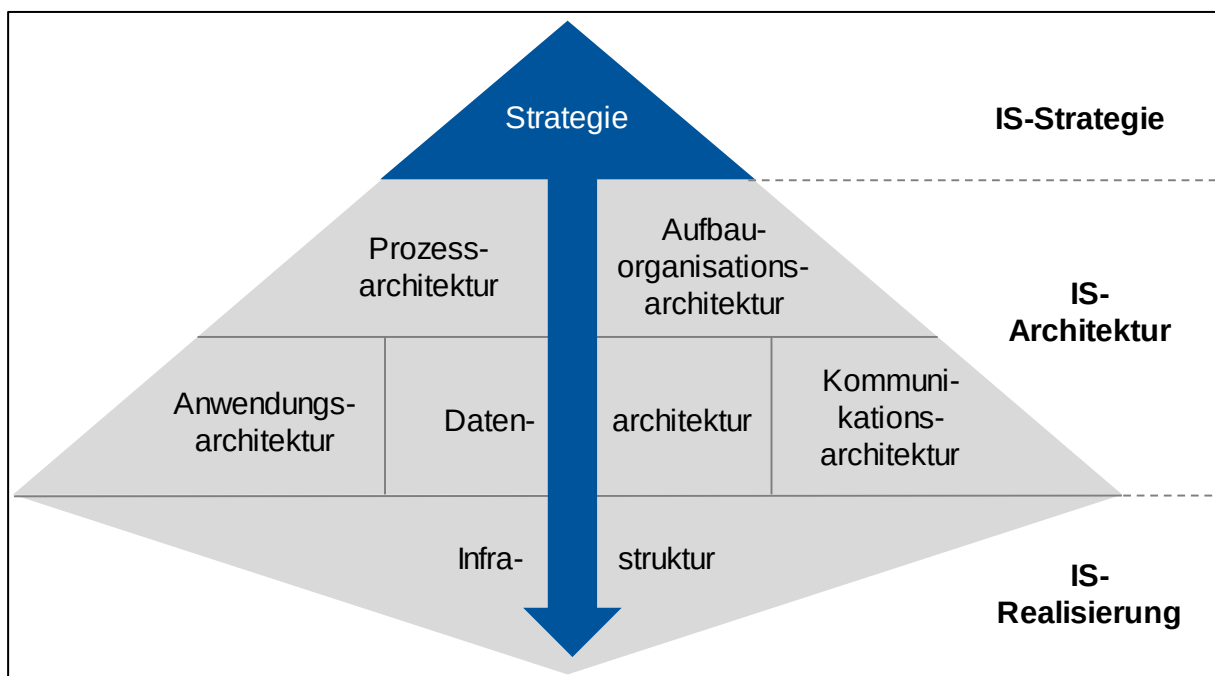


Abbildung 3-9: Ganzheitliches Modell der Informationssystem-Architektur (i. A. a. KRCMAR 1990, S. 399)

KRCMAR schlägt 1990 das sogenannte **ganzheitliche Modell der Informationssystem-Architektur** vor, dessen ganzheitlicher Charakter sich bis heute in verschiedenen Beschreibungsmodellen für Informationssysteme und -architekturen wiederfindet.

Demnach beinhaltet die Informationssystem-Architektur die Strategieebene, die Architekturebene und die Infrastruktur- beziehungsweise Realisierungsebene.

Die oberste Schicht der Informationssystem-Strategie enthält Elemente der Geschäftsstrategie, deren Vision sich durch das ganze Unternehmen und durch alle Systeme zieht, dargestellt als durchgängiger Pfeil (s. Abbildung 3-9). Die zweite Schicht beinhaltet die Aufbau- und Ablauforganisationsarchitektur (Prozessarchitektur). Die dritte Schicht enthält die Architekturen der Anwendungen, der Daten und der Kommunikation. Die Anwendungsarchitektur beschreibt Geschäftsprozesse und deren Unterstützung, wohingegen die Datenarchitektur den statischen Zusammenhang zwischen Daten beschreibt, wodurch Datenmodelle entwickelt werden können. Die Kommunikationsarchitektur hält die Informationsflüsse zwischen Anwendungen und Daten. Die unterste Schicht der Infrastruktur beschreibt den Einsatz von IKT im Unternehmen (s. KRCMAR 1990, S. 395-402).

	Was (Daten)	Wie (Funktion)	Wo (Netzwerk)	Wer (Personen)	Wann (Zeit)	Warum (Motivation)
Planer						
Besitzer						
Designer						
Builder						
Programmierer						
Nutzer						

Abbildung 3-10: Zachman Framework in generischer Form (i. A. a. ZACHMAN 1987)

Das **Zachman Enterprise Framework** nach ZACHMAN ist ein Ordnungsrahmen zur Entwicklung von Informationssystemen. Das ursprüngliche Konzept wurde 1987 eingeführt und beschreibt aus sechs verschiedenen Perspektiven (Zeilen in der Matrix) sechs verschiedene Dimensionen (Spalten in der Matrix) zur Modellierung der IS-Architektur (s. Abbildung 3-10). Demnach fokussiert das Framework die beteiligten Rollen und gibt eine Strukturierung für die jeweils zu verfolgende Sicht vor (s. ZACHMAN 1987). So betrachtet beispielsweise die Perspektive des Planers, welche Ressourcen, also beispielsweise welche Daten, notwendig sind (*Was*), welche Methoden, beispielsweise grafische Darstellungsmethoden, genutzt werden sollen (*Wie*) oder wer die Anwender bestimmter Funktionen, beispielsweise Operational Manager oder Mitarbeiter auf dem Shopfloor, innerhalb des Informationssystems sind (*Wer*). Obwohl das Framework sehr generisch ist, verdeutlicht es sehr gut die Komplexität einer Informationssystem-Architektur. Allerdings dient es vielmehr als Basis zur Entwicklung einer Architektur, kann aber kaum genutzt werden, um diese vollständig zu beschreiben.

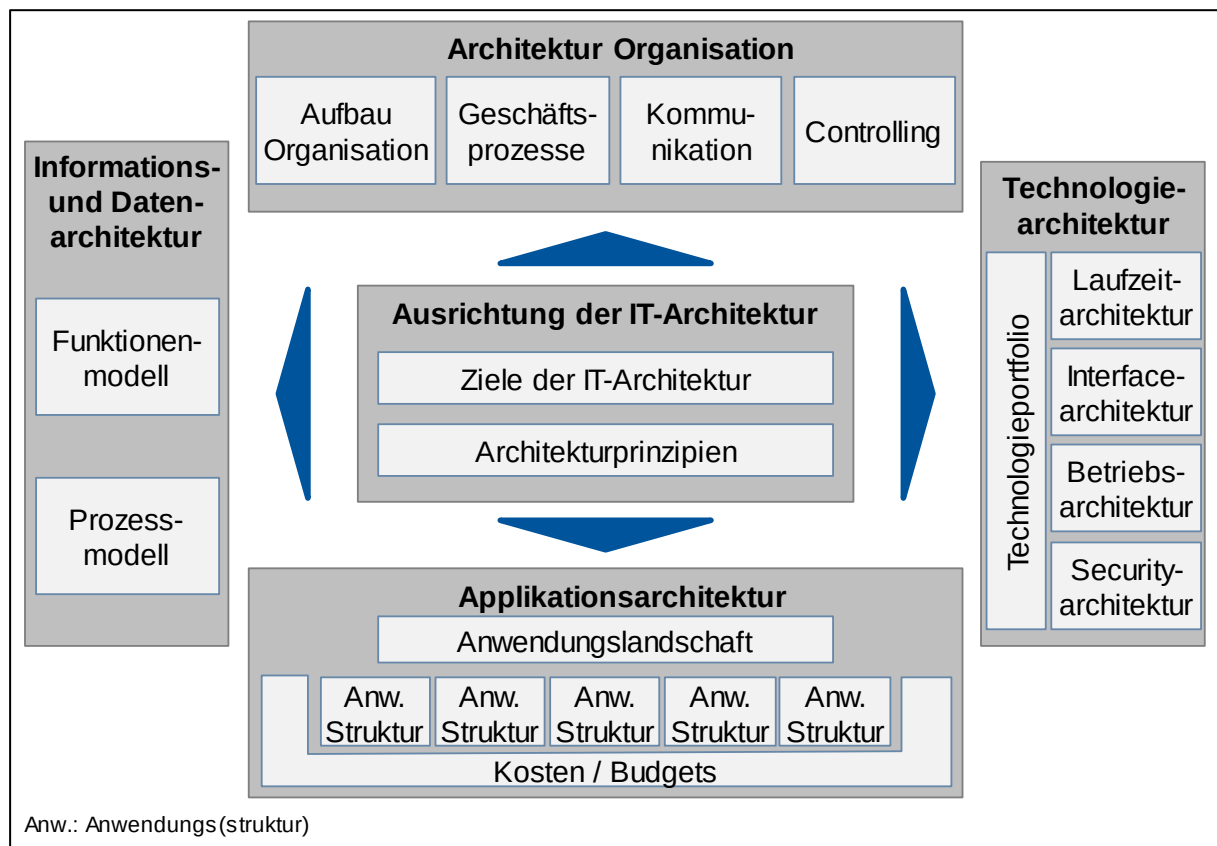


Abbildung 3-11: Ordnungsrahmen für das Enterprise Architecture Management (i. A. a. TIEMEYER 2017a, S. 105)

Der Ansatz des EAMs ist grundsätzlich dem durch KRCMAR eingeführten ganzheitlichen Modell der Informationssystem-Architektur sehr ähnlich. Durch einen ganzheitlichen Blick auf die Unternehmensarchitektur soll diese transparenter gemacht werden und sich so die Planbarkeit und Steuerbarkeit des Unternehmens und dessen Produkte und Services verbessern (s. TIEMEYER 2017a, S. 102).

Die Gesamtunternehmensarchitektur wird in den Teilebenen Strategie, Geschäftsarchitektur, Anwendungsarchitektur, Informationsarchitektur und technologische Architektur dargestellt und beinhaltet die relevanten fachlichen und technischen Strukturen eines Unternehmens (s. Abbildung 3-11). EAM bietet somit eine Grundlage für eine strategische Ausprägung der gesamten Geschäftsprozess-IT-Landschaft und die Grundlage für Optimierungen ebendieser und liefert die dafür erforderlichen Prinzipien, Methoden und Werkzeuge. (s. WEBER 2011, S. 16; HANSCHKE 2016, S. 7-26)

Das **The Open Group Enterprise Architecture Framework (TOGAF)** bietet ein werkzeugunabhängiges Vorgehensmodell zur Planung, Weiterentwicklung und Wartung von Unternehmensarchitekturen, welches ebenfalls die Prinzipien von EAM nutzt (s. MATTHES 2011, S. 188ff). TOGAF ist das aktuell am meisten genutzte Framework des Architekturmanagements (s. TIEMEYER 2017a, S. 149-150). Dabei wird die Unternehmensarchitektur in vier Bereiche aufgeteilt: Geschäftsarchitektur, Datenarchitektur, Anwendungsarchitektur und Technologiearchitektur. Ausgehend von den Anforderungen an die zu entwickelnde Architektur wird in neun Phasen beschrieben, wie aus der Architekturvision eine implementierte und überwachte Architektur entstehen kann

(s. JOSEY 2011, S. 28ff; THE OPEN GROUP 2014). In der konkreten Anwendung ist TOGAF mitunter aufwendig (s. FRIEDRICHSEN 2008) und für kleine Unternehmen schwer handhabbar. In Bezug auf EAM können Elemente der von TOGAF entwickelten Verfahren als Ansatzpunkt für die entwickelten Modelle in den folgenden Kapiteln verwendet werden.

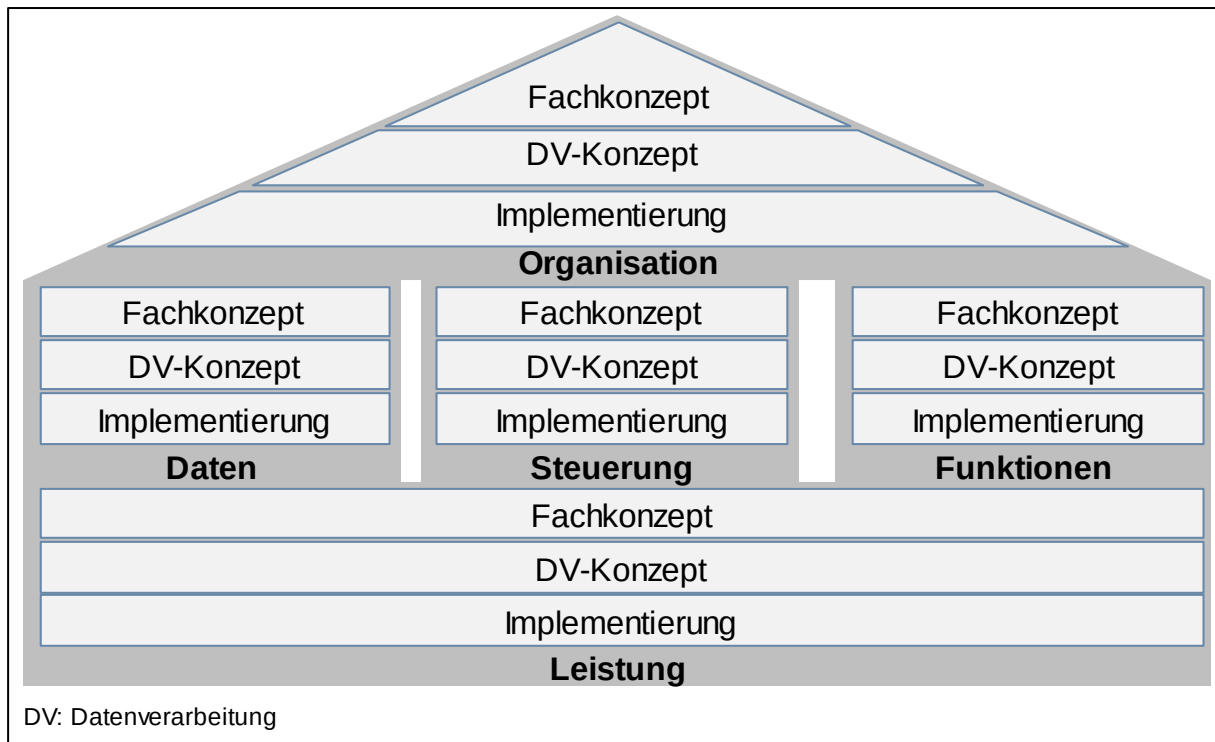


Abbildung 3-12: ARIS-Haus (i. A. a. SCHEER 2001, S. 41)

Das von SCHEER entwickelte Konzept der **Architektur integrierter Informationssysteme** (ARIS) wurde als Metamodell entwickelt, um die Komplexität des betrieblichen Geschehens systematisch erfassen und gestalten zu können. ARIS beziehungsweise die ARIS-Plattform ist in Deutschland eines der am meisten verbreiteten Konzepte des Prozessmanagements (s. SCHEER 2001, S. 1ff; FETTKKE 2012). Zur Beschreibung der Informationssystem-Architektur werden bei ARIS verschiedene Ebenen wie die Datensicht, die Funktionssicht, die Organisationssicht und die Steuerungssicht, „ausgehend von der Ebene des Problems in der Fachabteilung (Fachkonzept), immer weiter im Hinblick auf die DV-technische Umsetzung (DV-Konzept) bis hin zur Implementierung konkretisiert“ (LACKES 2014). Die grundlegende Sichten-Logik dieses Ansatzes kann zur Entwicklung der Modelle innerhalb der vorliegenden Arbeit angewendet werden.

Das IT-Service-Management-Modell **Information Technology Infrastructure Library** (ITIL) stellt derzeit den De-facto-Standard für IT-Service-Management in der Wirtschaft dar (s. SCHÄFER 2015, S. 2; BELMS 2016, S. 111; ZELLER 2018, S. 30).¹⁰ Dabei werden für zentrale Prozesse der Entwicklung und Bereitstellung von IT-Services

¹⁰ Zum Zeitpunkt der Abfassung dieser Arbeit war die im Jahr 2011 zuletzt aktualisierte Version 3 dieses Rahmenwerks der aktuelle Stand.

Praxiserfahrungen definiert und beschrieben, sodass dieses Regelwerk eine praktische Erfahrungssammlung liefert (s. KRCMAR 2015, S. 608). Zahlreiche Unternehmen haben ihre IT-Organisation gemäß den ITIL-Grundsätzen organisiert beziehungsweise Teile des ITIL-Frameworks für sich adaptiert (s. BELMS 2016, S. 112; s. KRESSE U. BAUSE 2012, S. 12-13). Zur Erfassung der Unternehmens-IT schlägt ITIL die sogenannte *Configuration-Management-Database* (CMDB) als zentrale Datenbasis für das IT-Service-Management vor. Es werden darin die Komponenten (Hardware- und Softwarekomponenten) und deren Beziehungen festgehalten, die von der IT-Organisation betreut werden. ZARNEKOW erkennt in ITIL vor allem die Stärke der Anwendungsunterstützung, jedoch fokussiert ITIL aus seiner Sicht stark die Identifikation von Ansatzpunkten zur Prozess- und Serviceverbesserung und liefert keine Lösungsansätze, um diese auch umzusetzen (s. ZARNEKOW 2007, S. 5; BEIMS U. ZIEGENBEIN 2014, S. 12-16).

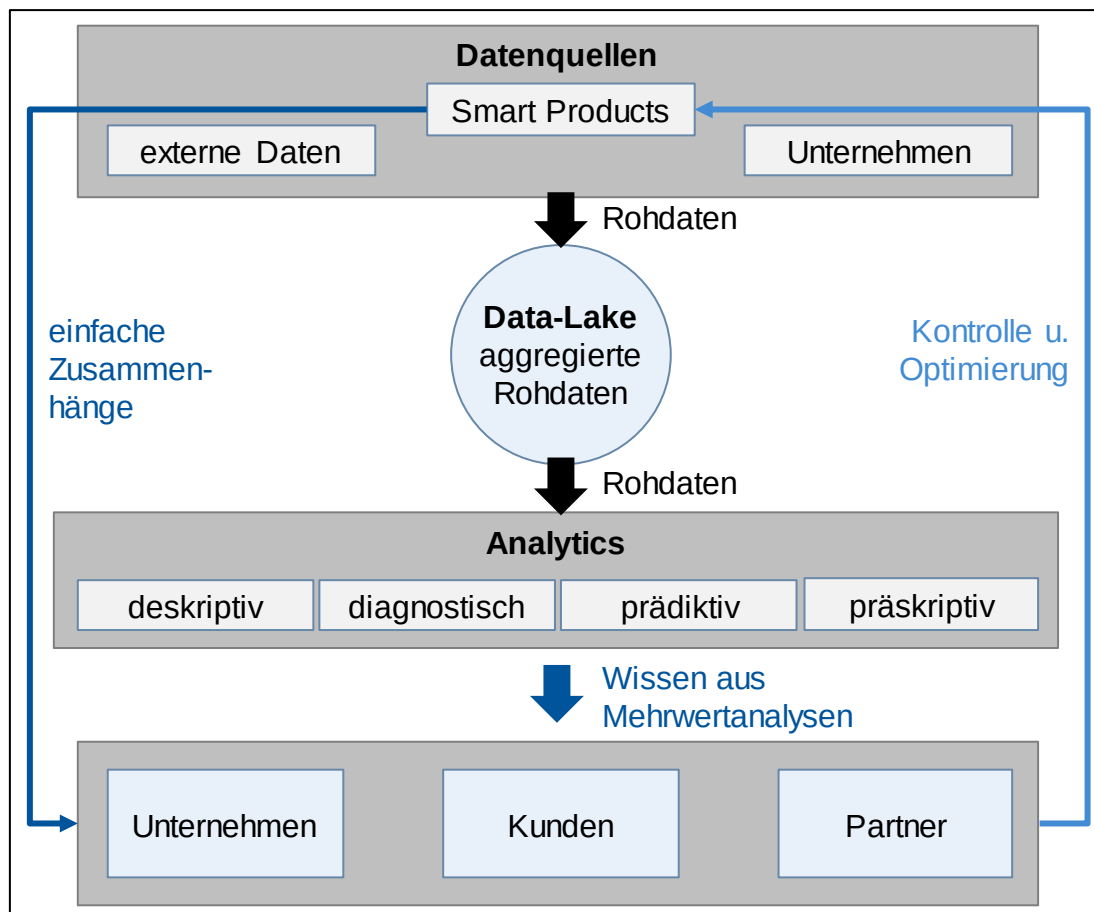


Abbildung 3-13: Architektur zur Mehrwerterzeugung aus Daten (i. A. a. PORTER U. HEPPELMANN 2015, S. 109)

PORTER U. HEPPELMANN beschreiben generisch, wie **eine Informationssystem-Architektur für Smart Products** und damit verbundene Mehrwertanalysen aussehen (s. Abbildung 3-13). Die von *Smart Products* generierten Daten können von Unternehmen, Kunden und Partnern zur Gewinnung einfacher Zusammenhänge, wie beispielsweise Nutzungsmuster, direkt verwendet werden. Um tiefere Einblicke zu gewinnen, sind Mehrwertanalysen notwendig. Dafür werden die Nutzungsdaten des Produkts ge-

meinsam mit externen Datenquellen und Daten aus den Anwendungssystemen in Unternehmen in *Data Lakes* gesammelt. Anschließend werden mithilfe verschiedener Analysewerkzeuge und -algorithmen Mehrwertanalysen durchgeführt, um neue Informationen zu gewinnen. Diese werden wiederum zur Verfügung gestellt (s. PORTER U. HEPPELMANN 2015). Dieses Modell zeigt entgegen der klassischen Architekturmodelle eine mögliche Architektur mit zentralem Datenzugriff auf. In der vorliegenden Arbeit wird es in Kapitel 7.1 genutzt, um einige Elemente der Ziel-Architektur abzuleiten.

Die beschriebenen Informationssystem-Architekturmodelle bilden den Denkraum dieser Dissertationsschrift. Es wurde jeweils darauf hingewiesen, auf welche Bestandteile dieser anerkannten Modelle in den folgenden Kapiteln aufgebaut wird. Dem Hauptkritikpunkt, dass die Modelle sehr generisch beschrieben sind, was deren Anwendbarkeit in produzierenden Unternehmen erschwert, wird durch die umfangreiche Beschreibung und praxisnahe Darstellung von Informationssystem-Architekturen in Kapitel 6.7 der vorliegenden Arbeit entgegnet.

3.4 Konzepte von Informationssystem-Architekturen produzierender Unternehmen

Zur Gestaltung einer Informationssystem-Architektur müssen stets die IT-Strategie und die übergeordnete Unternehmensstrategie beachtet werden (s. Kapitel 2.2.3), da die Strategie einen Einfluss auf die Informationssystem-Architektur und die Architektur einen Einfluss auf die Strategie hat (s. KAMPKER ET AL. 2011, S. 178; TIEMEYER 2017a). Die Gestaltung der Informationssystem-Architektur lässt sich demnach als eine strategische Aufgabe einordnen. Im Folgenden werden bestehende Methoden vorgestellt, die sich dieser gestaltenden Aufgabe annehmen.

In den vergangenen Jahren lässt sich eine starke Entwicklung in Richtung modularisierter IT-Architekturkonzepte (insbesondere SOA) für Teilbereiche von produzierenden Unternehmen beobachten (s. ALPAR ET AL. 2016, S. 170; HOFFMANN U. HEIMES 2018, S. 989). Das vom Fraunhofer IPA vorgeschlagene **Smart Manufacturing Network** ist eine **mögliche SOA im Produktionsumfeld**, das auf die Cloud-Lösung „*Virtual Fort Knox*“ zurückgreift. Dabei werden sowohl die verschiedenen Unternehmenssysteme als auch Mitarbeiter und Maschinen, einfach angebunden. Die angeschlossenen Computer können die entsprechenden Software-Tools als Service ausführen, Algorithmik und Software werden über das Netzwerk bereitgestellt (s. KLOCKE ET AL. 2017b, S. 274-275). In dem **Forschungsprojekt pICASSO** (Industrielle CloudbAsierte Steuerungsplattform für eine Produktion mit Cyber-physischen Systemen) kommen ebenfalls Prinzipien einer SOA zum Einsatz. Das Ziel des Projekts ist eine flexible Bereitstellung von Steuerungstechnik für CPS. Zu diesem Zweck wird die vorhandene, monolithisch gekapselte Steuerungstechnik aufgebrochen, modularisiert und durch verschiedene Mechanismen des Cloud-Computings, wie einer zentralen Datenverarbeitung und erwähnter SOA, erweitert (s. BAUERNHANSL ET AL. 2016, S. 19; KRÜGER ET AL. 2017, S. 93).

Obwohl sich **SOA** in Branchen wie im Banken- und Versicherungssektor (s. Kapitel 3.5) als Informationssystem-Architektur durchgesetzt hat (s. GRUBER ET AL. 2017, S. 1608-1609; ALT U. PUSCHMANN 2016, S. 181-188), stellt dies **für produzierende Unternehmen nur einen Teil der Lösung** dar. Die Stärke einer SOA liegt im modularen Aufbau und den definierten Schnittstellen zwischen verschiedenen Modulen. Wenn produzierende Unternehmen ihre komplette Informationssystem-Architektur auf SOA umstellen würden, hätte dies zur Folge, dass sie diese Module vollständig neu aufbauen müssten (s. NISSEN ET AL. 2012, S. 32).

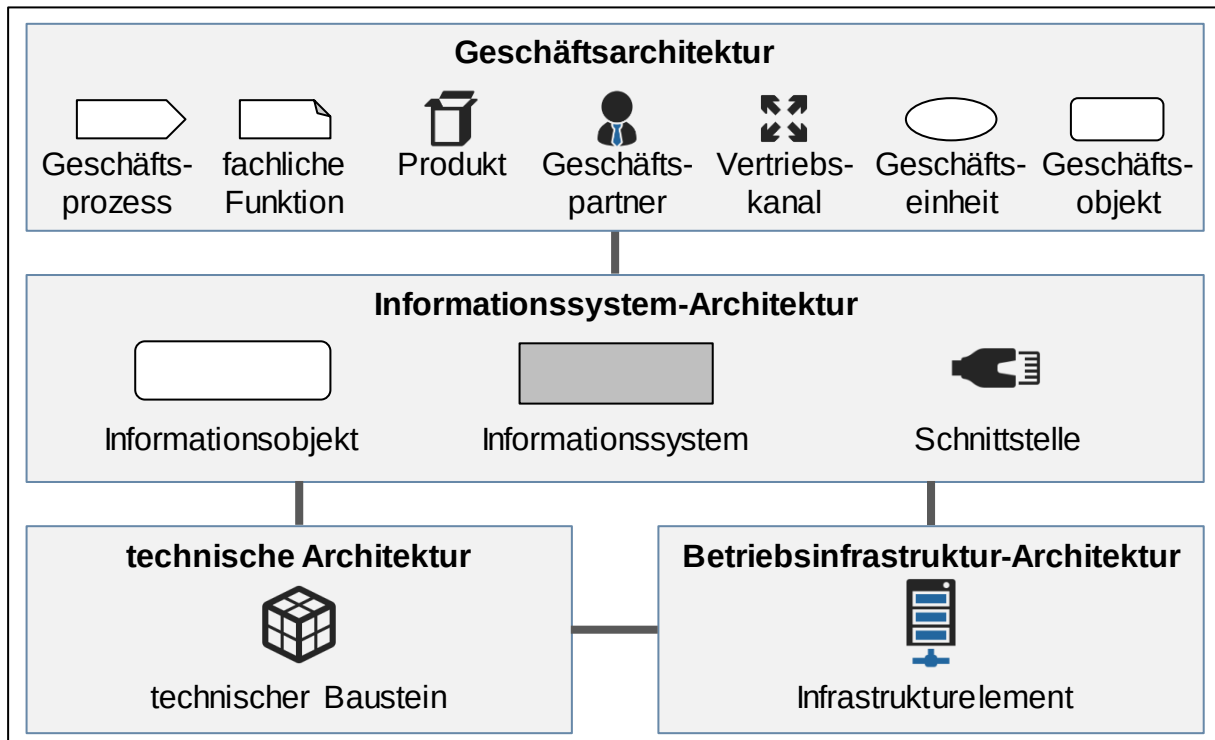


Abbildung 3-14: Best-Practice-Unternehmensarchitektur (i. A. a. HANSCHKE 2016, S. 36)

In einem Metamodell liefert HANSCHKE eine mögliche Gestaltung der Informationssystem-Architektur auf Basis von EAM (s. Kapitel 3.3). Die sogenannte **Best-Practice-EAM-Methode** wurde auf der Grundlage von TOGAF entwickelt und hilft bei einem schnellen und erfolgreichen Aufbau sowie einer kontinuierlichen Weiterentwicklung der Informationssystem-Architektur. Dabei folgt die Methode dem Leitmotiv „Einfach und effektiv“ und versucht, eine Lösung mit möglichst geringer Komplexität, die gleichzeitig ausreichend zielführend ist, anzubieten. Best-Practice-EAM besteht aus verschiedenen Bereichen, wie der Best-Practice-Unternehmensarchitektur, einer Sammlung von Katalogen, Leitfäden, Templates und Mustern, dem EA-Governance-Haus sowie einer etablierten Standardvorgehensweise für die nutzenorientierte agile Einführung und den Ausbau von EAM (s. Abbildung 3-14). Die Best-Practice-Unternehmensarchitektur bildet die Grundlage des Metamodells und beinhaltet alle wesentlichen fachlichen und IT-Strukturen sowie deren Verknüpfung und liefert einen Ansatz zum Aufbau und der Steuerung der Informationssystem-Architektur (s. HANSCHKE 2016, S. 33-57).

Der durch die ACATECH herausgegebene und am FIR entwickelte **„Industrie 4.0 Maturity Index“** zeigt grob eine ideale Architektur für produzierende Unternehmen. Bei dem Werk handelt es sich um einen Leitfaden für produzierende Unternehmen, der zur Erreichung des Ziels eines lernenden, agilen Unternehmens, das sich einer wandelnden Umwelt ständig anpassen kann, unterstützt. Im Vordergrund stehen Agilität und Wandlungsfähigkeit produzierender Unternehmen, die neben einer passenden Organisationsstruktur über eine geeignete Informationssystem-Architektur verfügen sollten. Die Informationssysteme müssen vertikal und horizontal integriert werden, um eine durchgängige Datenverarbeitung zu realisieren. (s. SCHUH ET AL. 2017a, S. 21-37)

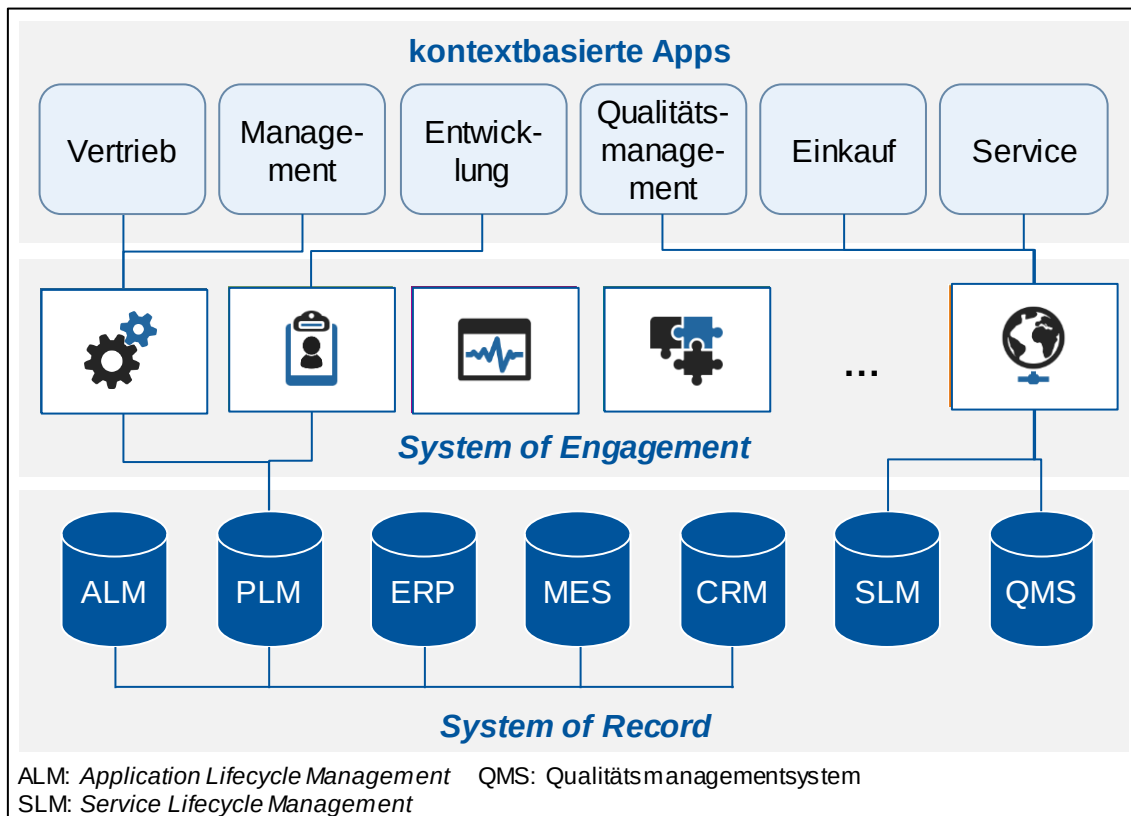


Abbildung 3-15: Vorgeschlagene Informationssystem-Architektur zur Interaktion und Kollaboration mit Anwendern des „Industrie 4.0 Maturity Index“ (eigene Darstellung i. A. a. SCHUH ET AL. 2017a, S. 27)

Die ideale Informationssystem-Architektur **verfügt demnach über eine zentrale Plattform**, die bestehende Informationssysteme untereinander und mit den Ressourcen vernetzt. Ein führendes Informationssystem beinhaltet alle Daten zur gemeinsamen Nutzung (SSoT). Zur Umsetzung dieser Architektur werden zwei Gestaltungsprinzipien vorgestellt. Das Prinzip der selbstlernenden Informationsverarbeitung beinhaltet automatische Datenanalysen, eine kontextbasierte Bereitstellung von Informationen, anwendungsgerechte Benutzerschnittstellen und den Aufbau einer resilienten IT-Infrastruktur. Über Apps können ausschließlich relevante, aufbereitete Informationen den richtigen Entitäten verfügbar gemacht werden. Diese Apps beziehen die benötigten Informationen aus den darunterliegenden Informationssystemen. Das zweite Gestaltungsprinzip beschreibt die **Integration der Informationssysteme**. Diese müssen zur

gemeinsamen Datennutzung vertikal und horizontal integriert werden. Eine hinreichende Datenqualität ist notwendig zur Integration, entsprechende Date-Governance-Richtlinien sollen umgesetzt werden. Außerdem müssen entsprechende Vorgehensweisen zur Umsetzung einer ausreichenden IT-Sicherheit beachtet werden. Abbildung 3-15 zeigt die beschriebene Informationssystem-Architektur (s. SCHUH ET AL. 2017a, S. 21-37). Eine Umsetzung dieser vorgeschlagenen Informationssystem-Architektur bietet die Architektur des *Internet of Production*, die in Kapitel 2.3 beschrieben wurde.

Für **einzelne Unternehmensbereiche** existieren **spezielle Modelle zur Gestaltung der Informationssystem-Architektur**. Für Produktionsprozesse und angrenzende Bereiche gibt es beispielsweise das **Aachener PPS-Modell** und die Automatisierungspyramide (s. Kapitel 2.1.6).

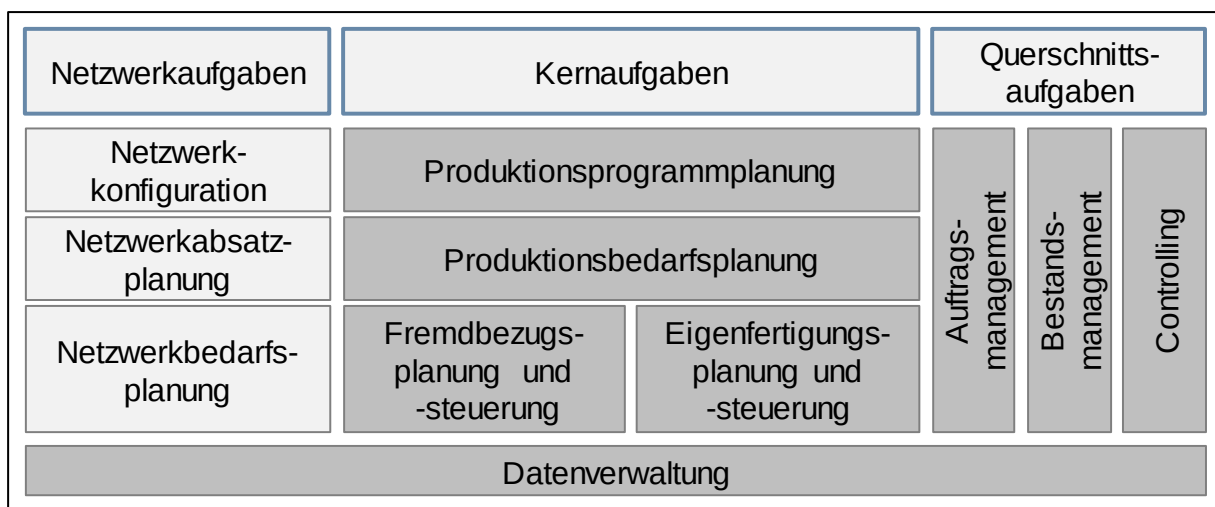


Abbildung 3-16: Aufgabensicht als Grundstruktur des Aachener PPS-Modells (i. A. a. SCHUH U. GIERTH 2007a, S. 21)

Die Produktionsplanung und -steuerung (PPS) ist der wesentliche Baustein von Produktionssystemen. Das Aachener PPS-Modell gibt eine abstrakte, ganzheitliche Betrachtung der Zusammenhänge in PPS, das mithilfe einzelner Teilmodelle und Referenzsichten produzierende Unternehmen unterstützt, PPS-Konzepte zu entwickeln und PPS-Prozesse zu harmonisieren, um so im Produktionsbereich eine strukturierte Informationssystem-Architektur aufzubauen. Das Modell besteht insgesamt aus vier unterschiedlichen Referenzsichten: Aufgabensicht, Prozessarchitektursicht, Funktionssicht und Prozesssicht. Die einzelnen Sichten sind für den Einsatz in unterschiedlichen PPS-Projekten vorgesehen. Die Grundstruktur des Modells wird durch die Aufgabensicht gebildet. Diese unterscheidet zwischen überbetrieblichen Netzwerkaufgaben und innerbetrieblichen Kern- und Querschnittsaufgaben. Abbildung 3-16 stellt diese Sicht des Aachener PPS-Modells dar (s. SCHUH U. GIERTH 2007a, S. 11, 12, 18).

Ein weiteres Teilmodell für eine Informationssystem-Architektur produzierender Unternehmen kann im **Bereich PLM** gefunden werden. PLM beschreibt einen Ansatz zur ganzheitlichen, unternehmensweiten Verwaltung und Steuerung aller Produktdaten und Prozesse (s. RUDOLF U. SCHREY 2016b, S. 90).



Abbildung 3-17: PLM-Funktionsmodell des WZL (i. A. a. RUDOLF U. SCHREY 2016b, S. 92)

Wie bereits in Abschnitt 2.1.6 dargelegt, werden unterschiedliche Systeme zur Datenverarbeitung über verschiedene Unternehmensfunktionen genutzt. Diese Strukturen müssen optimal miteinander verknüpft werden, sodass ein durchgängiger Informationsfluss ermöglicht wird. Das PLM-Funktionsmodell des Werkzeugmaschinenlabors (WZL) der RWTH Aachen liefert einen möglichen Ansatz, um die gezeichnete Herausforderung mit dem Ziel des PLM zu meistern. Insbesondere wird eine Schnittstelle der Produktdaten zu weiteren Datenmanagementsystemen, vor allem ERP- und SCM-Systemen, realisiert. Insgesamt kann durch PLM eine **umfassende Vernetzung der Produktdaten aller Unternehmensfunktionen** erreicht werden. Abbildung 3-17 zeigt das PLM-Funktionsmodell. Im Zentrum steht das Stücklistenmanagement, aus dessen Daten sich die anderen Funktionen bedienen (s. RUDOLF U. SCHREY 2016b, S. 90-94).

Während über den Aufbau einer Informationssystem-Architektur für produzierende Unternehmen nur Teilmodelle und Übersichtsbilder bestehen, geben FOTH sowie URBACH U. AHLEMANN **allgemeingültige Handlungsempfehlungen zu Gestaltung und Aufbau** einer ebensolchen. Eingeleitet werden sollte der Aufbau einer Informationssystem-Architektur mit einer Bestandsaufnahme der aktuell vorhandenen Systeme. Unter Anpassung an die im Unternehmen bestehenden Komponenten wie Unternehmensstrategie, Geschäftsprozessmodell, verfügbare Technologien, vorhandene Informationssysteme und IT-Infrastruktur kann die Definition der Anforderungen an eine ideale Architektur erfolgen. Aus dem Vergleich zwischen Ist- und Soll-Anforderungen lässt sich der erforderliche Handlungsbedarf ableiten (s. FOTH 2016, S. 40). Die zu erreichenden Eigenschaften einer Informationssystem-Architektur belaufen sich auf fünf übergeordnete Anforderungen. Elementar zur Reduktion der Komplexität bei bestehenden Weiterentwicklungsmöglichkeiten für hinzukommende Komponenten ist die

Modularität der Informationssystem-Architektur. Zur Sicherung der Interoperabilität verschiedener Anwendungen und dynamischen Anpassung an die Nachfrage sollten eine hohe *Flexibilität* und *Elastizität* der Architektur gegeben sein. Schließlich sollten auch Informationssystem-Architekturen aktuellen übergreifenden Unternehmensanforderungen gerecht werden und *ubiquitär*, also weltweit und auf jedem Endgerät verfügbar, *kostengünstig* und *sicher* sein (s. URBACH U. AHLEMANN 2016, S. 131-133).

In der vorliegenden Arbeit werden die Grundgedanken der im vorliegenden Kapitel vorgestellten Modelle aufgegriffen. So bildet das EAM den Grundstein für die Architekturdarstellungen in Kapitel 6.8, das Architekturmodell des ‚*Industrie 4.0 Maturity Index*‘ eine Vorlage für die Ordnung der Informationssysteme und die Empfehlungen von FOTH sowie URBACH U. AHLEMANN fließen in die Gestaltung der Zielinformationssystem-Architektur ein. Die PPS- und PLM-Architektur bilden hingegen nur ein Teilmodell der Informationssystem-Architektur produzierender Unternehmen ab und sind daher nicht ausreichend für die in der vorliegenden Arbeit vorgenommen Betrachtung des Gesamtunternehmens.

3.5 Methoden zum IT-Komplexitätsmanagement

Der Fokus wissenschaftlicher Aktivitäten im Bereich des IT-Komplexitätsmanagements liegt entweder auf der IT-Komplexität als mathematisches Problem (vgl. WIDJAJA ET AL. 2012) oder im Bereich der Banken und Versicherungen mit sehr speziellen Informationssystem-Architekturen (vgl. BEETZ 2014).

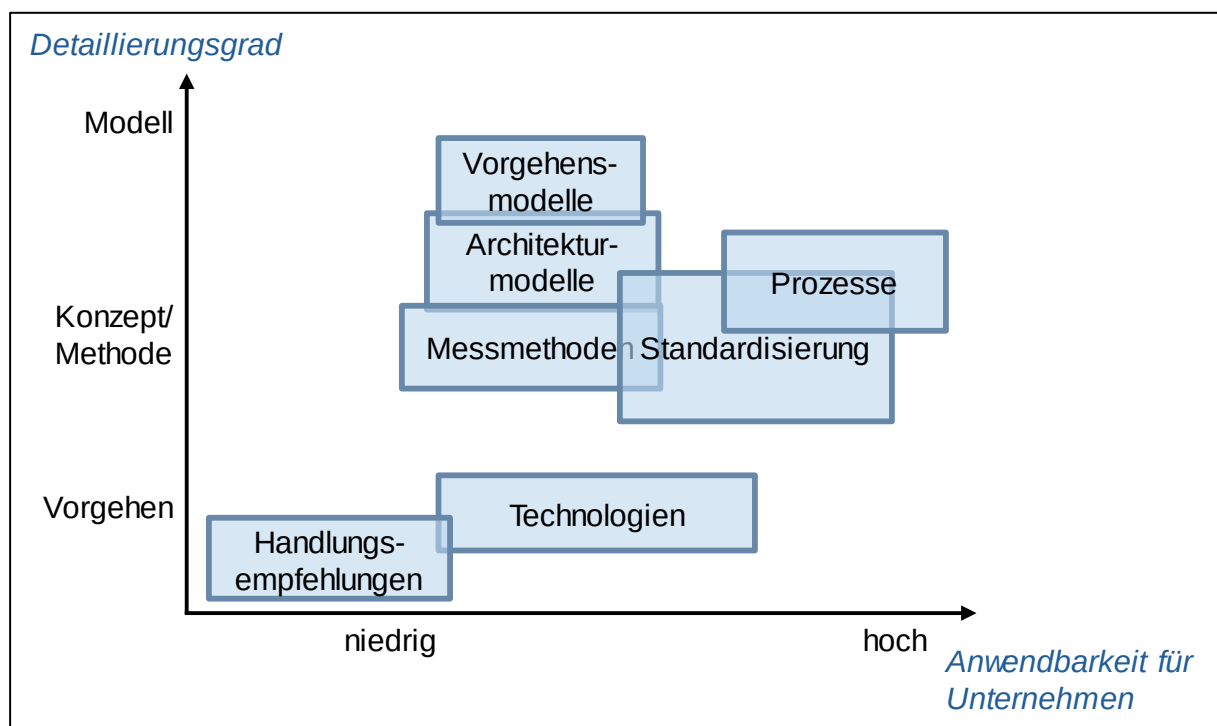


Abbildung 3-18: Kategorien von Literaturquellen zu IT-Komplexität (entnommen aus GRUBER ET AL. 2017, S. 1610; i. A. a. SCHUH ET AL. 2017f, S. 64)

Trotzdem bieten die einzelnen Quellen **wertvolle Hinweise zur Gestaltung von Informationssystem-Architekturen**, welche in Kapitel 7.1 der vorliegenden Arbeit aufgegriffen werden. Einen Überblick über existierende Modelle, Methoden und Vorgehensbeschreibungen, der in einem umfangreichen Literaturreview entstanden ist (vgl. SCHUH ET AL. 2017f, S. 62-66), gibt Abbildung 3-18, die im Folgenden näher detailliert werden.

Prozesse

Die Kategorie der *Prozesse* beschreibt alle Quellen, die die Modellierung, Optimierung, Integration oder Implementierung von Prozessen im Hinblick auf eine Reduzierung der IT-Komplexität thematisieren (s. SCHUH ET AL. 2017f, S. 63).

BECKER U. ROSEMANN erkannten die Wichtigkeit der **Prozessmodellierung im IT-Komplexitätsmanagement**. Die Modellierung erhöht die Transparenz, vereinfacht dadurch das Eliminieren von Elementen und Beziehungen und fördert somit ein aktives Schnittstellenmanagement (s. BECKER U. ROSEMANN 1998, S. 111-124). Rund acht Jahre später beschreiben ROSS ET AL. den **Einfluss von Prozessstandardisierungen auf den Erfolg des Unternehmens**. Die Einführung von Standardisierungen soll nach Ansicht der Autoren an die Geschäftsziele angepasst werden. Überdies stellen die Autoren einen Zusammenhang zwischen dem *Operating Model* und der Unternehmensarchitektur her, die es aufeinander abzustimmen gilt (s. ROSS ET AL. 2006). Weniger detailliert beschreiben RICHTER ET AL. den Ansatz **von Prozesstransparenz und Standardisierung zur Komplexitätsreduzierung** mithilfe eines „ganzheitliche[n] Prozess- und IT-Architekturansatz[es]“ (RICHTER ET AL. 2008, S. 21). Um die Prozesstransparenz herbeizuführen, sprechen die Autoren davon, die Strukturen zu „entwirren“, führen diese Idee aber nicht weiter aus.

ARMAS sagt in ihrem Artikel, dass die Entwicklung eines Informationssystems ein komplexes Problem darstellt und führt dies auf die Verlässlichkeit von Informationen und die interne sowie externe Heterogenität des Unternehmens zurück. Nach Einschätzung der Autorin kann ein Unternehmen durch seine Kernprozesse dargestellt werden. Deshalb analysiert sie die IT-Komplexität aus einer prozessorientierten Sicht, indem sie die **Abhängigkeiten von Prozessen und Informationen** darstellt. (s. ARMAS 2011, S. 117-127)

Handlungsempfehlungen

Handlungsempfehlungen thematisieren allgemein die Problematik der steigenden IT-Komplexität, stellen aber keine konkreten Methoden oder Konzepte vor (s. SCHUH ET AL. 2017f, S. 63). BECKER U. ROSEMANN benennen dabei das **Schnittstellenmanagement zur Komplexitätsreduzierung**, weil viele Schnittstellen die Flexibilität verringern und den Austausch von Teilsystemen erschweren (s. BECKER U. ROSEMANN 1998, S. 111-124). FROHMÜLLER U. KIEFER weisen darauf hin, dass neue und **alte Informationssysteme gut integriert** werden müssen und die Informationssysteme in erster Linie die Kernprozesse unterstützen und abdecken sollen (s. FROHMÜLLER U. KIEFER 1999, S. 832-837).

NOTH beschreibt in einem Artikel über die Finanzbranche, dass **Informationssysteme zentralisiert** werden sollen und Standardisierungen der Heterogenität und den hohen IT-Kosten entgegenwirken sollen (s. NOTH 2005, S. 12-14). Diese Ansicht, ebenfalls in Bezug auf die Finanzbranche, vertritt auch BRÜHWILER in seinem Artikel. Er stellt die Vorteile von standardisierten Gesamtbankensystemen heraus (s. BRÜHWILER 2012, S. 47-48).

In seinem dreiteiligen Artikel stellt MURCH einige Lösungsmöglichkeiten für die Reduzierung der IT-Komplexität vor. Als Voraussetzung für die Umsetzung von Reduzierungslösungen nennt MURCH die **Zerlegung von Prozessen**, weil kleinere Strukturen einfacher zu managen sind, sowie die Einführung von Best-of-Breed-Lösungen. Auch die **Konsolidierung von Soft- und Hardware** sowie eine Umgestaltung der Informationssystem-Architektur und IT-Outsourcing werden vom Autor als Lösungsmöglichkeiten aufgeführt (s. MURCH 2004). Der Beitrag von FEHLMANN fokussiert die Abhängigkeit von Informationssystemen und umgesetzten Prozessen. Hiernach sollen **Prozesse mithilfe von IT möglichst vollständig automatisiert** und integriert sowie manuelle Schnittstellen eliminiert werden (s. FEHLMANN 2006, S. 36-37). MARIOTTI hebt die **Relevanz der Anzahl von Informationssystemen**, ebenso die Dokumentation, die Größe der Datenbanken und die Prozesslandschaft hervor (s. MARIOTTI 2008).

LEMBERGER U. MOREL stellen vier wesentliche Konzepte vor: (1) Bei *Simplicity by Reduction* geht es darum, unnötige Funktionen oder Informationssysteme abzustellen, die meistens bei der Migration eines neuen Systems in ein Altsystem entstanden sind. (2) *Simplicity by Hiding* verfolgt die Idee, Komplexität zu verstecken, indem man diese mittels Software-as-a-Service auslagert. (3) *Simplicity through organization (of the IT system)* bedeutet, die Konfigurationen auf ein Minimum zu reduzieren, um Unordnung im System zu vermeiden. (4) *Simplicity through Learning* baut darauf auf, dass Wissen IT-Komplexität reduzieren kann, weil größeres Wissen komplexe Zusammenhänge einfacher aussehen lässt (s. LEMBERGER U. MOREL 2012).

In einem Beitrag der *Boston Consulting Group* (BCG) werden Empfehlungen mit Zahlen aus Studien hinterlegt und anhand von Beispielen konkretisiert. Neben der **Konsolidierung von Applikationen**, um Redundanzen und Schnittstellen zu reduzieren, empfehlen GREBE U. DANKE auch den Einsatz von Shared-Service-Centers. Diese Zentralisierung von Prozessen führt zu einer effizienteren Nutzung und senkt Kosten. Außerdem sprechen die Autoren von einer **Verschlinkung der Strukturen** in der Hard- und Software und bei Middleware-Elementen (s. GREBE U. DANKE 2013).

Standardisierungen

Unter *Standardisierungen* werden in der Literatur häufig EAM und SOA beschrieben. Transparenz und Strukturierung sind häufig genannte Gründe für die Umsetzung (s. SCHUH ET AL. 2017f, S. 63; GRUBER ET AL. 2017, S. 1608).

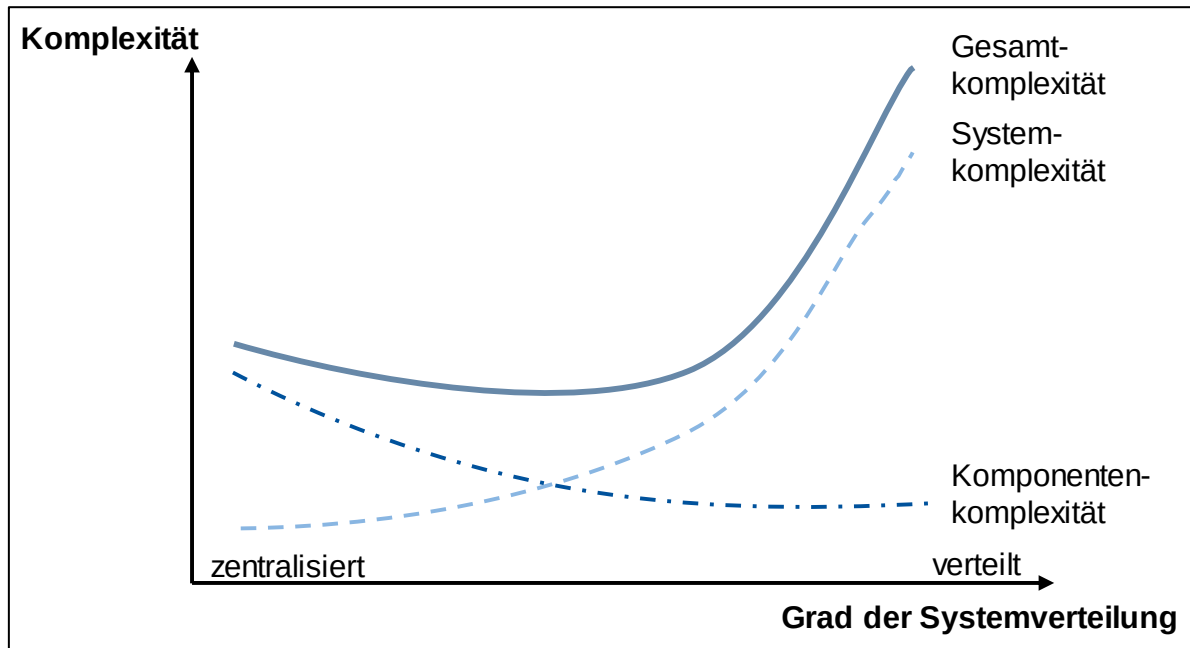


Abbildung 3-19: Complexity-Cross (i. A. a. SCHNEBERGER U. MCLEAN 2003, S. 219)

SCHNEBERGER U. MCLEAN beschreiben in ihrem **Complexity-Cross** die Abhängigkeit von steigender Systemkomplexität und sinkender Komponentenkomplexität sowie den Anstieg der gesamten Komplexität, wenn die beiden erstgenannten zu weit auseinandergehen. Diese Entwicklung ist nach Aussage der Autoren auf den Wechsel von zentralisierten Großrechnern, also komplexen Komponenten in einfachen Systemen, auf verteilte Systeme, also einfache Komponenten in komplexen Systemen, zurückzuführen. Um die IT-Komplexität zu reduzieren, müssen demnach beide Komplexitätskurven in ihrem Verlauf gebremst werden, sodass an der Kreuzung der Kurven ein optimales Komplexitätsniveau entsteht. Im Bereich der Systemkomplexität empfehlen die Autoren umfangreiche Standardisierungen *“to the maximum extent possible”* (SCHNEBERGER U. MCLEAN 2003, S. 216-225).

SOA hat sich insbesondere in der Finanzbranche durchsetzen können und wird auch in dem Artikel von PLOOM ET AL. in diesem Zusammenhang vorgestellt. Da die Umstellung der Informationssystemlandschaft auf eine SOA bei einer Bank zwar eine Reduzierung der Komplexität herbeigeführt hat, die Komplexität allerdings trotzdem noch unüberschaubar ist, stellen die Autoren eine Kombination aus einer *Model Driven Architecture* und einer SOA vor. Ergebnis dieser theoretischen Analyse ist, dass eine Kombination der beiden Ansätze in der Praxis eine Komplexitätsreduzierung zur Folge hätte, aber mit sehr hohem Implementierungsaufwand verbunden wäre (s. PLOOM ET AL. 2007, S. 59-68).

KEUNTJE U. BARKOW wie auch SCHNEIDER U. MATTHES betonen die **Einführung von EAM** (s. Kapitel 3.3) als Mittel zur Komplexitätsreduzierung in historisch gewachsenen Informationssystemlandschaften (s. KEUNTJE U. BARKOW 2010; SCHNEIDER U. MATTHES 2014, S. 694-699).

BENTE **kritisiert das klassische EAM** aufgrund geringer Erfolgsaussichten und einer unflexiblen Herangehensweise. Um das Feld des EAMs agiler zu gestalten, beschreibt

er den Ansatz der *kollaborativen EA* mit drei Leitlinien: (1) Schlanker Satz von IT-Prozessen und Regeln, (2) evolutionärer Prozess für die IT-Planung und Problemlösung und (3) Partizipation vieler Beteiligten an IT-Entscheidungen (s. BENTE 2014, S. 187-223; GRUBER ET AL. 2017, S. 1608).

ALBAYRAK U. GADATSCH empfehlen den Einsatz von **Standards in den Bereichen Hard- und Software** sowie bei der IT-Organisation und den IT-Prozessen. Sie weisen aber andererseits auch auf die Einbußen bei der Flexibilität hin (s. ALBAYRAK U. GADATSCH 2014, S. 680-685).

Technologische Lösungen

Technologische Lösungen wurden insbesondere durch IBM und die Idee der sich selbst kontrollierenden Systeme vorangetrieben (s. SCHUH ET AL. 2017f, S. 63). Die Idee des **autonomic computing** wird von GANEK ET AL. vorgestellt. Es geht dabei um Systeme, welche die Prinzipien der Selbstkonfiguration, -heilung, -optimierung und des Selbstschutzes vereinen (s. GANEK ET AL. 2004, S. 151-157). Der *autonomic manager* hat die Fähigkeiten, Daten des *managed element* zu überwachen, zu analysieren, zu planen und auszuführen. Welche Funktion genutzt wird, ist jeweils abhängig von dem Ergebnis der Analyse. Während dieser Analyse sammelt der Manager Wissen, das für die zukünftige Entscheidungsfindung genutzt und permanent erweitert wird.

MOAWED beschäftigt sich mit neuen Technologien zur Reduzierung von IT-Komplexität. Die Grundlage dazu bildet eine Studie von CIOInsight und SAP, bei der 339 Unternehmen zu IT-Komplexität befragt wurden. Die am wichtigsten bewertete Technologie zur Reduktion von IT-Komplexität stellt demnach eine **In-Memory-Datenbank** dar (s. MOAWED 2015). Solche Ansätze lösen allerdings nicht die Ursache von Komplexität, sondern lindern nur deren Symptome; sie werden daher in der vorliegenden Arbeit nicht weiterverfolgt.

Messung

Die Ansätze bei der *Messung* von IT-Komplexität sind vielseitig (s. SCHUH ET AL. 2017f, S. 63-64). Ein übergeordneter Ansatz wurde 2004 von ARTETA U. GIACHETTI veröffentlicht. Sie berechnen die **Komplexität von Geschäftsprozessen mithilfe der Entropiekennzahl**. Ein Geschäftsprozess wird durch „a set of activities connected by information and/or material flows“ (ARTETA U. GIACHETTI 2004, S. 500) charakterisiert. Sie beweisen mittels eines Vergleichs von nicht optimierten gegen optimierte Prozesse, dass weniger komplexe Systeme einfacher zu ändern sind. Dies erreichen sie durch die Berechnung des sogenannten *ease of change* der beiden verglichenen Prozesse (s. ARTETA U. GIACHETTI 2004, S. 495-503; GRUBER ET AL. 2017, S. 1606-1607).

BOOCH beschreibt einen weiteren Ansatz zur **Messung der Komplexität** der Informationssystem-Architektur. Er empfiehlt als Basis das 4+1-Schichtenmodell von KRUCHTEN (vgl. KRUCHTEN 1995). Zu **jeder Schicht** – *Logical, Development, Process, Physical View* und *Scenarios* – sollen die Klassen und Objekte, Anzahl der Prozesse, Threads und Komponenten sowie die Knoten **gezählt werden**. Diese Zahlen geben

einen Überblick über die Größe des Systems. Werden dann die Beziehungen betrachtet, wird zusätzlich die Dynamik deutlich und zuletzt wird die Anzahl der verwendeten Entwurfsmuster identifiziert. Der Autor gibt keinerlei Hinweise über ein Vorgehen oder eine Interpretation der Zahlen, diese sollen nur über die Zeit hinsichtlich Veränderungen beobachtet werden (s. BOOCH 2008, S. 14-15).

Um das Gebiet der IT-Komplexität besser verstehen zu können, hat MOCKER in seinem Artikel auf Basis einer Literaturanalyse verschiedene Typen von Komplexität in Bezug auf Anwendungen identifiziert. Basis sind **Kennzahlen zur Messung von Komplexität** (s. GRUBER ET AL. 2017, S. 1606-1607). DERN U. JUNG greifen dies auf und stellen folgende Kennzahlen vor: Die Anzahl der Informationssysteme innerhalb der Architektur bestimmt die *Komponentenanzahl* (n). Überdies sollen Abhängigkeiten identifiziert werden. Als *Abhängigkeit* (a) wird eine Informationsbeziehung zwischen IT-Systemen beschrieben. Die tatsächliche Anzahl der Abhängigkeiten soll ins Verhältnis zu der maximalen Anzahl von Abhängigkeiten in einem IT-System gesetzt werden.

Als dritte Kennzahl wird die *Homogenität* (h) gemessen. Diese bezieht sich auf die Gleichartigkeit der verwendeten Informationssysteme. Die Autoren unterscheiden zwischen vollständiger Homogenität (Wert 1) und vollständiger Heterogenität (Wert 2), „wenn alle Komponenten paarweise voneinander verschieden sind“ (DERN U. JUNG 2009, S. 671). Merkmale zur Bestimmung werden nicht vorgegeben und Werte zwischen 1 und 2 werden geschätzt.

Wie ARTETA U. GIACHETTI haben auch SCHMIDT ET AL., SCHÜTZ ET AL., SCHÜTZ und WIDJAJA ET AL. ein Modell entwickelt, das auf dem **Entropiemaß** beruht. Die Grundidee ist die folgende: „das Entropie-Maß steigt mit der Anzahl unterschiedlicher Charakteristika und mit zunehmender Gleichverteilung der Elemente“ (SCHMIDT ET AL. 2013, S. 1265). Ziel ist es demnach, **das Maß zu verringern und eine homogenere IT-Architektur anzustreben**. Anhand eines praxisnahen Beispiels werden IT-Komplexitätsaspekte identifiziert und jeweils dem Ist-Wert ein Ziel-Wert zugeordnet. Außerdem wird anhand des Beispiels erklärt, welche Aktivitäten unternommen werden müssen, um den Zielwert zu erreichen (s. ARTETA U. GIACHETTI 2004, S. 495-503; WIDJAJA ET AL. 2012, S. 1-14; SCHMIDT ET AL. 2013, S. 1261-1275; SCHÜTZ ET AL. 2013, S. 1-12; SCHÜTZ 2017, S. 86).

In der Dissertationsschrift von BEETZ werden viele der zuvor genannten messungs-basierten Ansätze beschrieben, analysiert und zu einem **Gesamtmodell der IT-Architekturkomplexität** vereint. Die Dimensionen beruhen zum größten Teil auf dem zuvor vorgestellten Artikel von MOCKER und wurden in ihrer Messbarkeit von BEETZ erweitert. Das Modell verfolgt das Ziel, die IT-Architekturkomplexität von großen Unternehmen abzubilden. Der Autor identifiziert auf Basis der IT-Architektur sieben verschiedene Komplexitätsarten: *Größe der IT-Landschaft, Interdependenzen, Anpassungsgrad, Redundanz, Diversität, Abweichung von Standards und Alter*. Außerdem unterscheidet er jeweils zwischen IT-Applikationen und der IT-Infrastruktur. Zur jeweiligen Komplexitätsart wird angegeben, wie diese erfasst werden soll. Um diese Größen zu errechnen, liefert der Autor Hilfestellungen (s. BEETZ 2014, S. 84).

Architekturmodelle

In der Kategorie der *Architekturmodelle* finden sich verschiedene Herangehensweisen, die Informationssystemlandschaften eines Unternehmens zu gestalten. Es geht sowohl um die optimale Strukturierung als auch um Schnittstellen und Abhängigkeiten (s. SCHUH ET AL. 2017f, S. 63). Zuerst stellten LANGLINIS U. FORBES eine Lösung vor, bei der Methoden eines Autoherstellers auf den Finanzsektor übertragen werden. Die Idee beruht auf einer **modularen Plattform**. Diese soll nach verschiedenen Geschäftsbereichen aufgeteilt werden und verringert dadurch die Redundanzen von Prozessen und Ressourcen (s. LANGLINIS U. FORBES 2006, S. 8A).

Ein weiteres Modell stellt SESSIONS vor. Er beschreibt eine Methode, wie die Unternehmensarchitektur sinnvoll aufgebaut oder nachträglich optimiert werden kann. Sein Vorgehen beinhaltet drei Schritte und ist mathematischer Natur. Es ist außerdem mit einem Algorithmus (Äquivalenzrelation) hinterlegt, der darauf beruht, dass die Komplexität eines Systems durch die Funktion seiner möglichen Zustände definiert ist. Weiterhin beschreibt der Autor den Prozess **Simple Iterative Partitions** (SIP), um die Komplexität der Unternehmensarchitektur zu kontrollieren und die Informationssystem-Architektur an die Geschäftsprozesse anzupassen, damit diese effektiv zusammenarbeiten. Die zuvor genannten drei Schritte sind in diesem Modell integriert. Das Vorgehen von SIP wird in der Abbildung 3-20 verdeutlicht.

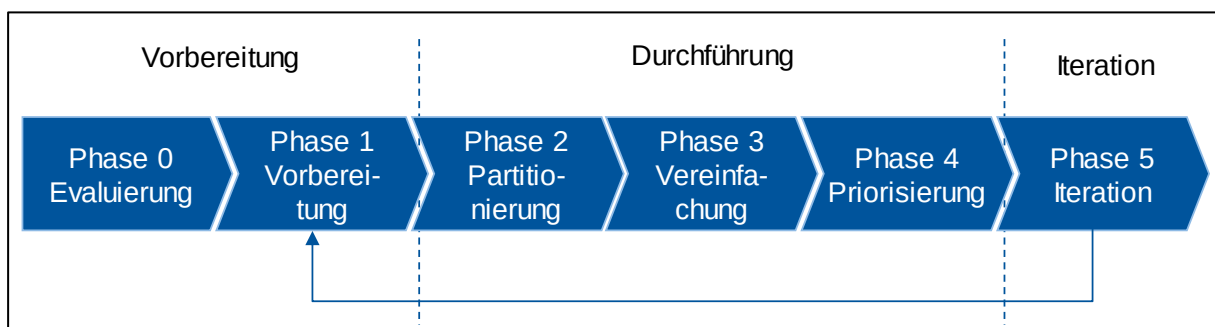


Abbildung 3-20: SIP-Prozess (i. A. a. SESSIONS 2008, S. 108)

SOLIC ET AL. beschreiben in ihrem Artikel die von SESSIONS vorgestellte Methode und ergänzen diese mit kleinen Beispielen (vgl. SOLIC ET AL. 2011, S. 1528-1531).

Vorgehensmodelle

In der Kategorie der *Vorgehensmodelle* ist der **IT-Complexity-Check** von WIENKER zu erwähnen. Dieses Vorgehen stellt einen methodischen Ansatz zur Reduktion der IT-Komplexität im Unternehmen dar und beinhaltet fünf aufeinanderfolgende Schritte. Der Autor ist der Ansicht, dass die Kernprozesse eines Unternehmens einen wesentlich höheren wertschöpfenden Beitrag leisten und deshalb „die Grundlage für die IT-Strategie bilden“ (WIENKER 2006, S. 39).

DURST geht auf die Probleme der Unternehmen mit der wachsenden IT-Komplexität ein und führt dies auf eine Integration von Informationssystemen zurück, die sich nicht an der Unternehmensstrategie orientiert und dadurch keinen Wertbeitrag liefert. Er

stellt ein Modell vor, das die **Architekturkonformität** bewertet. Auf Basis der Konformität kann beispielsweise über eine geplante Implementierung eines Informationssystems entschieden werden. Ein Kennzahlenmodell zum Soll-Ist-Vergleich von Informationssystem-Architekturen unterstützt diese Entscheidung. Die Zielerreichung wird prozentual ermittelt. Die allgemeinen strategischen Ziele unterteilen sich in Strategieobjekte. Jedes dieser Objekte verfolgt Teilziele, nach dessen Erfüllungsgrad entschieden wird, ob ein Informationssystem mit diesen konform ist (s. DURST 2007).

3.6 Zwischenfazit und Ableitung des Forschungsbedarfs

Ausgangspunkt der Untersuchung ist die Forschungslücke im Bereich der IT-Komplexität und die Lösungshypothese, hierzu eine an den Integrationsanforderungen von software-definierte Plattformen ausgerichtete Informationssystem-Architektur zu gestalten (s. Kapitel 1.2). Insgesamt zeigt die in Tabelle 3-1 zusammengefasste Untersuchung von Ansätzen aus der Literatur und Praxis, dass es zu vielen Bereichen **Teilansätze** gibt, ein **generischer, zusammenhängender Lösungsansatz** für produzierende Unternehmen jedoch **fehlt**. Im Einzelnen bedeutet dies:

- Der Begriff der **software-definierten Plattformen** ist zwar durch die ACATECH im Jahr 2015 eingeführt worden, eine weitere Vertiefung, im Speziellen für die praktische Anwendung für Unternehmen, steht jedoch aus. In der **Praxis** können **IoT-Plattformen** als software-definierte Plattformen eingesetzt werden, jedoch sind diese hauptsächlich für die Nutzung von smarten Geräten beschrieben, nicht für den Einsatz in der Informationssystem-Architektur.
- **Integrationskonzepte** existieren für andere Informationssysteme, sie fehlen jedoch für software-definierte Plattformen und greifen nicht die Herausforderungen der IT-Komplexität auf.
- In den letzten 30 Jahren wurden viele allgemeingültige **Beschreibungsmodelle für Informationssystem-Architekturen** entwickelt. Diese sind jedoch oft allgemein gehalten und lassen sich nicht direkt auf produzierende Unternehmen anwenden. Auf diese Modelle, im Speziellen auf **EAM**, kann jedoch aufgebaut werden.
- Gänzlich fehlt ein Ansatz, der eine **Informationssystem-Architektur** für produzierende Unternehmen beschreibt. Es gibt zwar Teilmodelle wie das Aachener PPS-Modell im ERP-Kontext, jedoch keine aktuell gültige Beschreibung, welche Informationssysteme eingesetzt werden sollten und wie sie zu integrieren sind.
- Der **Forschungsbereich der IT-Komplexität** ist noch immer ein neues Forschungsgebiet. In jüngerer Zeit wurde hierzu einiges publiziert, jedoch bezieht sich die Forschung noch immer hauptsächlich auf die Branche der Banken und Versicherungen. Trotzdem können die getroffenen Empfehlungen genutzt werden, um ein einfach **anwendbares Modell für Unternehmen für die Industrie** abzuleiten.

Im folgenden Kapitel 4 werden, um die beschriebenen Lücken zu schließen, daher die Ziele der Arbeit vorgestellt und das passende methodische Vorgehen erläutert, mit dem die Ergebnisse nachvollziehbar erarbeitet werden.

Tabelle 3-1: Übersicht untersuchter Quellen zum Stand der Erkenntnisse (eigene Darstellung)

Differenzierungsmerkmale	Modellierungszweck				betrachtete Schichten						Plattformen			Evaluation			Anwendungsdomäne			Informationsmanagement-Aufgaben				Unternehmensgröße		
untersuchte Ansätze [AUTOR (Konzept)]	Beschreibung	Erklärung	Gestaltung	sonstige Modelle	Business	Functional	Information	Communication	Integration	Asset	Service-Plattform	software-definierte Plattform	vernetzte physische Plattformen	Formal	Fallstudien	reale Nutzung	produzierende Unternehmen	Dienstleistungen	Sonstige	Führungsaufgaben des IM	Management d. Informationswirt.	Management d. IS	Management d. IKT	Mikrounternehmen	KMU	Großunternehmen
1. Beschreibung von software-definierten Plattformen																										
ACATECH 2015 (Smart Service Welt)	●	●	●		●	●	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●
PORTER U. HEPPELMANN 2014	●	●			●	●	●	●	●	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●
BAUMS 2015	●	●	●			●	●	●	●	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●
PELINO U. HEWITT 2016	●	●					●	●	●	●		●	●				●	●	●	●	●	●	●	●	●	●
GUTH ET AL. 2016	●	●					●	●	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●
KRAUSE ET AL. 2017	●	●			●	●	●	●	●	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●
2. Plattform-Integrationskonzepte																										
KAIB 2002 (EAI)	●		●	●	●	●	●	●	●	●			●	●	●		●	●	●	●		●		●	●	●
MINGUEZ 2013 (MSB)	●	●	●	●	●	●	●	●	●	●		●	●	●			●	●	●	●		●	●	●	●	●
DERN 2011	●	●	●	●	●	●	●	●	●	●				●	●		●	●	●	●	●	●	●	●	●	●
FREY 2015	●	●	●	●		●	●	●	●	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●
INFORMATICA 2017 (iPaaS)			●	●			●	●	●	●		●	●		●	●	●	●	●	●	●	●	●	●	●	●
3. Beschreibung von Informationssystem-Architekturen																										
KRCMAR 1990 (ISA)	●	●	●		●	●	●	●	●	●				●			●	●	●	●	●	●	●	●	●	●
ZACHMAN 1987	●		●		●	●	●	●	●	●				●	●		●	●	●	●	●	●	●	●	●	●
TIEMEYER 2017a (EAM)	●	●	●		●	●	●	●	●	●					●		●	●	●	●	●	●	●	●	●	●
THE OPEN GROUP 2014 (TOGAF)	●	●	●		●	●	●	●	●	●				●	●		●	●	●	●	●	●	●	●	●	●
SCHEER 2001 (ARIS)	●	●	●		●	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●
KRESSE U. BAUSE 2012 (ITIL)	●	●	●		●	●	●	●	●	●		●			●		●	●	●	●	●	●	●	●	●	●
PORTER U. HEPPELMANN 2015	●	●	●			●	●	●	●	●		●			●		●	●	●	●	●	●	●	●	●	●
4. Informationssystem-Architekturen produzierender Unternehmen																										
KLOCKE ET AL. 2017b (SOA i. d. Prod.)	●	●	●		●	●	●	●	●	●		●	●		●	●	●	●	●	●	●	●	●	●	●	●
HANSCHKE 2016 (Best Practice EAM)	●	●	●	●	●	●	●	●	●	●					●		●	●	●	●	●	●	●	●	●	●
SCHUH ET AL. 2017a (Acatech Maturity Ind.)	●	●	●		●	●	●	●	●	●	●	●	●		●		●	●	●	●	●	●	●	●	●	●
SCHUH U. GIERTH 2007a (Aachener PPS)	●	●	●		●	●	●							●	●		●			●		●		●	●	●
RUDOLF U. SCHREY 2016b (PLM Schema)	●	●	●		●	●	●	●	●			●			●		●	●		●		●		●	●	●

4 Herleitung des Konzeptansatzes

Wissenschaft basiert auf Fakten, jedoch sind Wahrnehmung und Interpretation dieser Fakten subjektiv (s. CHALMERS 2013, S. 16-17). Die in den vorherigen Kapiteln beschriebenen Problemstellungen und der Forschungsbedarf sollen jedoch sowohl intersubjektiv als auch nachvollziehbar angegangen werden. Daher ist ein **wissenschaftlich fundiertes Vorgehen als Grundlage dieser Arbeit** elementar (s. SCHNEEWEIß 1992, S. 254; RAFFÉE 1995, S. 11; NACHREINER 1997, S. 87). Im folgenden Kapitel werden deshalb die **formalen und problembezogenen Anforderungen** an das Ergebnis dieser Arbeit definiert. Um einen **methodischen Ansatz** zur Integration von software-definierten Plattformen in Informationssystem-Architekturen produzierender Unternehmen zu entwickeln, wird auf den in Kapitel 1.3 definierten generellen wissenschaftlichen Bezugsrahmen zurückgegriffen und daraus methodische Grundlagen abgeleitet (s. Kapitel 4.2). Diese werden im letzten Teil zum Vorgehen aggregiert, was die Grundlage für die Ergebnisse dieser Arbeit bildet.

4.1 Anforderungen an die zu entwickelnde Methode

Zur zielorientierten Anwendung des in Kapitel 1.3 skizzierten Forschungsprozesses ist es notwendig, die Anforderungen an die zu entwickelnden Konzepte klar zu definieren. Diese Konzepte, zusammengefasst als Theorie zu bezeichnen (s. KERLINGER U. LEE 2000, S. 11), beschreiben den für diese Arbeit relevanten Teilbereich der Realität in produzierenden Unternehmen: die Informationssystem-Architekturen. Nach POPPER ist eine Theorie wie folgt zu verstehen:

„Die Theorie ist das Netz, das wir auswerfen, um ‚die Welt‘ einzufangen, – sie zu rationalisieren, zu erklären und zu beherrschen. Wir arbeiten daran, die Maschen des Netzes immer enger zu machen.“ (POPPER 1935, S. 25)

Die an Theorien gestellte Anforderungen können in inhaltliche und formale Anforderungen eingeteilt werden (s. SCHMIDT 2008, S. 43). Die **inhaltlichen Anforderungen** an diese Arbeit, die sich aus den in Kapitel 1.2 formulierten Problemstellungen und Zielen ableiten, können wie folgt zusammengefasst werden:

- Erstellen einer **Übersicht über software-definierte Plattform-Konzepte** mit den Eigenschaften domänenübergreifender Datenzugriffsmöglichkeit, Echtzeit-Datenauswertemöglichkeit und Datensatzverwaltung
- **Systematische Beschreibung von software-definierten Plattformen** mit den vorgenannten Eigenschaften
- **Beschreibung von Integrationsanforderungen** von software-definierten Plattformen für die Integration in die Informationssystem-Architektur
- **Beschreibung von Informationssystem-Architekturen** produzierender Unternehmen im Hinblick auf die Integration einer software-definierten Plattform

- **Erklärung der Unterschiede zwischen Ist- und Ziel-Zustand** einer Informationssystem-Architektur anhand der Integrationsanforderungen von software-definierten Plattformen
- **Gestaltung der Einführung von software-definierten Plattformen** in die Informationssystem-Architektur produzierender Unternehmen
- **Anwendernahe Unterstützung** durch praxisnahe Empfehlungen, um IT-Komplexität besser zu beherrschen

Die **formalen Anforderungen** beziehen sich auf die wissenschaftliche Konzeption und die Güte des Forschungsprozesses. Es finden sich dazu zahlreiche Kriterien in der Wissenschaftsliteratur. Gerade im Kontext der noch relativ jungen Wissenschaft der Wirtschaftsinformatik, in deren Forschungsfeld diese Arbeit zum großen Teil fällt (s. Kapitel 1.3), finden in der jüngeren Literatur lebhafte Diskussionen darüber statt (vgl. ÖSTERLE ET AL. 2010). Fasst man diese zusammen, so lassen sich **vier Überbegriffe für formale Anforderungen** formulieren: Originalität, Validität, Reliabilität und Utilität (s. POPPER 1935, S. 40-51; NACHREINER 1997, S. 87-92; SCHÜTTE 1998, S. 111-116; ÖSTERLE ET AL. 2010, S. 662-679; GRAUMANN 2015, S. 531; EISEND U. KUß 2017, S. 113-129).

Die **Forderung nach Originalität** der Theorie und Ergebnisse der Forschungsarbeit steht sicherlich bei der Abfassung einer Dissertation als erstes Kriterium vor allen anderen. Eine wissenschaftliche Arbeit muss einen **innovativen Beitrag** leisten und damit eine Neuerung gegenüber dem Status quo hervorbringen (s. ÖSTERLE ET AL. 2010, S. 668; GRAUMANN 2015, S. 531; EISEND U. KUß 2017, S. 116). Solche Neuerungen sind sicherlich nicht immer revolutionär, wie beispielsweise die eines COPERNICUS, um ein historisches Beispiel heranzuziehen (vgl. COPERNICUS 1543), trotzdem sollten sie eine Weiterentwicklung, Ergänzung oder Veränderung des aktuellen Stands der Wissenschaft darlegen.

Validität, die zweite Forderung, bezieht sich auf **logische Korrektheit** einer Theorie (s. EISEND U. KUß 2017, S. 114). Die wissenschaftliche Arbeit sollte in Vorgehen und Ergebnis widerspruchsfrei und damit konsistent sein (s. POPPER 1935, S. 40-51; EISEND U. KUß 2017, S. 114). Gleichzeitig beinhaltet die Validität auch die Forderung nach Gültigkeit der Repräsentanz der Wirklichkeit, die sogenannte **Konstruktionsadäquanz** (s. NACHREINER 1997, S. 90; SCHÜTTE 1998, S. 113-114). Darin inbegriffen ist eine gewisse Abstraktion von der Wirklichkeit, die für einen definierten Gültigkeitsbereich Allgemeingültigkeit beansprucht (s. ÖSTERLE ET AL. 2010, S. 668; EISEND U. KUß 2017, S. 114-115). Zuletzt muss eine valide Theorie dem POPPER'schen Kriterium der **Falsifizierbarkeit** folgen, die Theorie also so aufgebaut sein, dass sie durch entsprechende Beobachtungen der Wirklichkeit widerlegt werden kann (s. POPPER 1935, S. 50-51; EISEND U. KUß 2017, S. 114).

Die **Reliabilität** bezieht sich zuerst auf die **Nachweisbarkeit, Wiederholbarkeit und Zuverlässigkeit** der Theorie (s. SCHMIDT 2008, S. 44). Werden die Ergebnisse dieser Arbeit also, bei gleichen Rahmenbedingungen, wiederholt angewendet, so sollten die

gleichen Ergebnisse zu erwarten sein (s. FRIEDRICHS 1981, S. 102). Hiermit inbegriffen ist die Forderung nach **Präzision** von Sprache, Nachvollziehbarkeit von Begründungen und exakter Definition von Begriffen, ohne die eine Theorie nicht nachvollziehbar wiederholt werden kann (s. ÖSTERLE ET AL. 2010, S. 668; EISEND U. KUß 2017, S. 115). Die Forderung, die Reliabilität nachzuweisen, ist jedoch aufgrund der Verschiedenheit von produzierenden Unternehmen in der Praxis nur schwer zu erfüllen (s. SCHMIDT 2008, S. 44; DEINDL 2013b, S. 54).

Im Kontext der in Aachen besonders hervorgehobenen **anwendungsorientierten Forschung** ist die letzte Forderung der **Utilität** von zentraler Bedeutung. ÖSTERLE ET AL. sprechen hierbei vom Nutzen des Ergebnisses, um Mehrwert für seine Anspruchsgruppen zu erzeugen (s. ÖSTERLE ET AL. 2010, S. 669). Dies schließt die Forderung nach **Einfachheit und Klarheit** mit ein, um nicht durch zu hohe Komplexität die Anwendbarkeit und Verständlichkeit zu erschweren (s. SCHÜTTE 1998, S. 115; EISEND U. KUß 2017, S. 115). Eine nutzenstiftende wissenschaftliche Arbeit sollte zudem dem **Wirtschaftlichkeitskriterium** genügen, dem jede ökonomisch arbeitende Unternehmung aufgrund der naturgegebenen Knappheit an Ressourcen folgen muss (s. SCHÜTTE 1998, S. 114; WÖHE ET AL. 2016, S. 4). Die Erfüllung des Kriteriums der Utilität kann im Forschungsprozess vor allem durch die enge Rückkopplung mit Experten aus der Wirtschaft sichergestellt werden.

4.2 Methodische Grundlagen

Forschungsmethoden gibt es vielfältige, je nach Wissenschaftsgebiet und Anwendungsfall. Manche Methoden, wie die der Empirie, werden seit langer Zeit genutzt, andere, wie die Aktionsforschung, werden international seit einigen Jahren interdisziplinär angewendet (s. MCNIFF U. WHITEHEAD 2011, S. 8), haben in Deutschland im Kontext der Wirtschaftsinformatik aber eine noch geringe Verbreitung. Wichtig ist für die **methodischen Grundlagen** dieser Arbeit, dass sie **auf die jeweilige Fragestellung abgestimmt** werden und nicht umgekehrt (s. NACHREINER 1997, S. 92).

Da es Kernziel der vorliegenden Arbeit ist, systematische Beschreibungen und Zusammenhänge im Kontext von Informationssystem-Architekturen zu entwickeln, sind *Systemtheorie* und *Modellbildung* die grundlegenden Methoden für diese Arbeit. In der Vorbereitung dieser Arbeit wurde zudem eine grundlegende und vollständige Literaturanalyse anhand des Vorgehens von BROCKE ET AL. durchgeführt (s. BROCKE ET AL. 2009), die methodisch und inhaltlich bei SCHUH ET AL. (vgl. SCHUH ET AL. 2017f) beschrieben ist.

4.2.1 Systemtheorie

Die Systemtheorie beschäftigt sich mit der Lösung komplexer Aufgaben mit dem Ziel, enthaltene Einzelaufgaben zielsicher zu lösen. Dazu zeigt sie Methoden zur Abstraktion der Wirklichkeit in Systemen auf Basis von Systemdenken auf und hilft, Details zu betrachten, ohne dabei das große Ganze aus dem Blick zu verlieren (s. WINZER 2016, S. 1). Der Ansatz der Systemtheorie, zuerst von BERTALANFFY vorgeschlagen (s. u. a.

BERTALANFFY 1949, S. 157) und von H. ULRICH in die Betriebswirtschaftslehre eingeführt (vgl. ULRICH 1970), wird inzwischen wegen seines generalistischen, strukturierten und vorurteilsfreien Ansatzes interdisziplinär verwendet und ist **im Gebiet der Wirtschaftsinformatik eine der meistgenutzten Methoden** der letzten Jahre (s. ULRICH U. HILL 1976a, S. 308-309; HOUY ET AL. 2014, S. 20). Obschon die Systemtheorie in jüngerer Zeit intensiv diskutiert und hinterfragt wird, ob sie in einer sich schnell ändernden Welt, die schnelle Entscheidungen bei Unsicherheit benötigt, noch zweckmäßig ist (s. COLLOPY 2015, S. 536; WINZER 2016, S. 10; GAYNOR 2016, S. 7), so scheint sie für die Strukturierung des in einer Dissertation von einer einzelnen Person bearbeiteten Problemfeldes noch immer zweckmäßig.

Grundsätzlich befasst sich die Systemtheorie mit Systemen, deren Elementen und den Beziehungen zwischen Teil- und Gesamtsystemen (s. ACKERMANN 2007, S. 19). Die Übertragung dieses Ansatzes auf die Gestaltung von Systemen wird als **Systemtechnik** beziehungsweise **Systems Engineering** bezeichnet, welche Teil der allgemeinen Systemtheorie ist (s. WINZER 2016, S. 2; ROPOHL 2012, S. 34). Im *Systems Engineering* wird zur Anwendung des Systemdenkens ein strukturiertes Vorgehen vorgestellt, mit dem die Komplexität bei Planung, Entwurf und Realisierung von umfangreichen Systemen reduziert wird (s. HABERFELLNER ET AL. 2015, S. 55).

Um dieses Verfahren anzuwenden, ist zuerst die Klärung des grundsätzlichen Begriffs **System** notwendig. Dieser Begriff wurde schon ausführlich in Kapitel 2.1.2 vorgestellt; zentral ist dort die Abbildung 2-4, in der die Zusammenhänge von Systemen, Unter- und Umsystemen dargestellt sind. Wichtig im Kontext des *Systems Engineering* ist, dass ein System nur Teile der Wirklichkeit als Modell der realen Ganzheit abbilden kann (s. ROPOHL 2012, S. 54). Ein Systemmodell bildet entsprechend der Intention des Systemmodellierers nur die relevanten Teilaspekte ab und unterstützt damit das Verständnis der Wirklichkeit (s. Abbildung 4-1; GRAUMANN 2015, S. 531). Daher kann es zu *einer* Wirklichkeit auch *mehrere* Systemmodelle geben. So kann beispielsweise ein Logistiksystem aus Sicht des Informationsflusses wie auch aus der Sicht des Materialflusses abgebildet werden (s. BÖHM U. FUCHS 2002, S. 20-24).

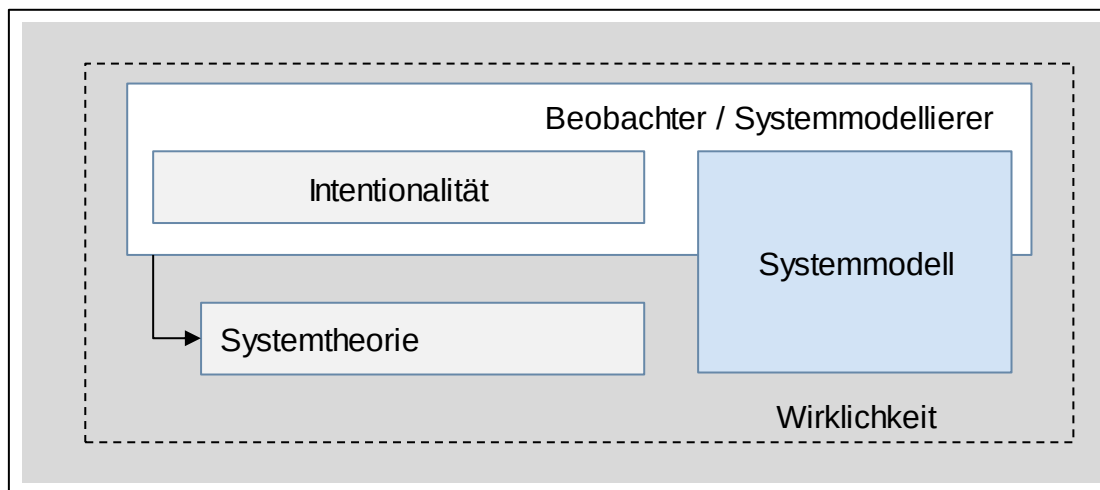


Abbildung 4-1: Abbildung der Wirklichkeit durch Systeme (i. A. a. ROPOHL 2012, S. 54)

Zur Anwendung des *Systems Engineering* schlagen HABERFELLNER ET AL. ein Vorgehensmodell mit vier Grundgedanken vor.¹¹ Der erste Grundgedanke beschreibt das Vorgehen **vom Groben zum Detail** (*top down*). Ziel dabei ist es, das Betrachtungsfeld nicht im Vorhinein schon zu stark einzuengen und damit gegebenenfalls sinnvolle Lösungen auszuschließen. Außerdem soll vermieden werden, Details zu intensiv zu betrachten, wenn das Gesamtbild noch unklar ist. Mithilfe des Systemdenkens sollen also zuerst die Systeme, deren Umwelt und ihre Beziehungen zueinander strukturiert werden, um das Problemfeld zu erkennen und generelle Ziele festzulegen. Erst danach sind die einzelnen Systeme, die bisher als Blackbox betrachtet wurden, zu präzisieren (S. HABERFELLNER ET AL. 2015, S. 56-59; WINZER 2016, S. 16).

Als zweites Grundprinzip wird die **Variantenbildung** vorgeschlagen. Deren Ziel ist es, nicht vorschnell die erste gefundene Lösung als einzige weiterzuverfolgen. Da es **für jede komplexe Problemstellung immer mehrere Lösungsalternativen** gibt und zu Beginn nicht bewertet werden kann, welche Alternative die beste in Bezug auf die gegebenen Zielgrößen ist, sind verschiedene Varianten zu entwickeln. Dieses Vorgehen wird auch als Kreativitätstechnik verwendet, da bei der Diskussion von Vor- und Nachteilen einzelner Lösungen oft neue Ideen entstehen. Auch hierbei ist der oben vorgestellte Verfeinerungsansatz zu nutzen; bei jeder Verfeinerungsebene sind Zwischenentscheidungen zu treffen, damit nicht jede Einzelvariante bis ins tiefste Detail diskutiert werden muss (S. HABERFELLNER ET AL. 2015, S. 59-62; WINZER 2016, S. 19).

Die **Gliederung in Projektphasen** beschreibt das dritte Grundprinzip. Dieses Prinzip erweitert die beiden zuvor vorgestellten Prinzipien in Richtung eines generischen Projektplans mit vordefinierten Entscheidungspunkten. Dieser ist als „Makro-Logik“, also als **übergeordnete Vorgehensweise zur Problemlösung**, zu verstehen. Wie in Abbildung 4-2 dargestellt, beginnt der Projektplan mit dem Anstoß. Diese unstrukturierte Phase hat die Entscheidung zum Ergebnis, ob eine Vorstudie gestartet werden soll und entsprechend Ressourcen dazu zur Verfügung gestellt werden. In den anschließenden Vor-, Haupt- und Detailstudienphasen wird das behandelte Problem schrittweise entsprechend dem Prinzip „*vom Groben zum Detail*“ präzisiert. Nach jeder Phase gibt es einen Entscheidungspunkt, der entweder zum Projektabbruch führt oder die Varianten festlegt, die einer detaillierteren Betrachtung unterzogen werden. Erst danach beginnt der tatsächliche Systemaufbau und -test sowie die Systemeinführung, woran sich die Projektabschlussarbeiten anschließen (S. HABERFELLNER ET AL. 2015, S. 63-71).

Vorschläge für den **Problemlösungszyklus**, dem vierten Grundprinzip des *Systems Engineering*, gibt es in der Literatur vielzählige (vgl. WINZER 2016, S. 23 ff). BAHILL U. GISSING haben aus einem Vergleich einer Vielzahl an verfügbaren Systems-Engineering-Ansätzen einen allgemeinen Problemlösungszyklus geschaffen, den SIMILAR-

¹¹ Eine Sammlung weiterer Grundprinzipien hat WINZER zusammengestellt (vgl. WINZER 2016, S. 16-20).

Prozess.¹² Dieser beschreibt ein Vorgehen von der Problembeschreibung, Variantenentwicklung über die Erprobung und Bewertung bis zur Verbesserung (s. BAHILL U. GISSING 1998, S. 516-518). Teile dieses Prozesses haben HABERFELLNER ET AL. in ihren Ansatz integriert: Der **Problemlösungszyklus** beschreibt dort als „**Mikro-Ebene**“ das Vorgehen für jede der vorgenannten Phasen (s. Abbildung 4-2). Vereinfacht besteht er aus drei Lösungsbestandteilen: In der Zielsuche sollen problemorientiert der Ist- und der Ziel-Zustand beschrieben werden. Die Lösungssuche dient der Auswahl und Untersuchung von möglichen Lösungen. Danach folgt die Auswahl, in der die beste Lösung für die entsprechende Phase bewertet und ausgewählt wird.

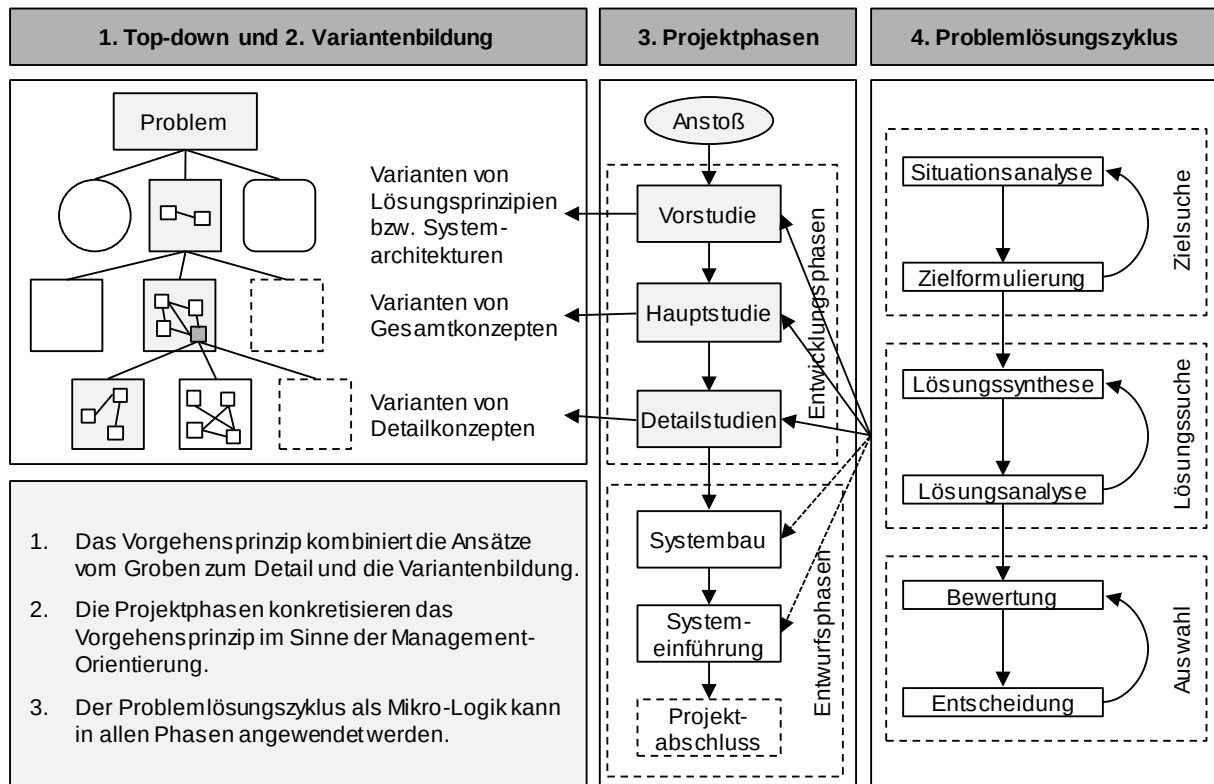


Abbildung 4-2: Übersicht der Grundprinzipien des Systems Engineering (i. A. a. HABERFELLNER ET AL. 2015, S. 83; DEINDL 2013b, S. 57)

Anwendung der Systemtheorie für die vorliegende Arbeit

Für die vorliegende Arbeit bildet der **Problemlösungszyklus** die **Grundlage der Modellstruktur**. Er bildet in einem sollzustandsorientierten Vorgehen die Modelle ab, die in der Arbeit entwickelt werden (s. HABERFELLNER ET AL. 2015, S. 81; DEINDL 2013b, S. 138). Die Zielformulierung findet in Kapitel 5 anhand des Beschreibungsmodells zukünftig eingesetzter software-definierter Plattformen statt. Die Situationsanalyse wird anhand des Beschreibungsmodells aktuell vorhandener Informationssystem-Architekturen in produzierenden Unternehmen durchgeführt (s. Kapitel 6). Für die Lösungssynthese wird das Erklärungsmodell zur Erklärung des Unterschieds zwischen Ist- und

¹² SIMILAR steht als Akronym für: State, Investigate, Model, Integrate, Launch, Assess, and Re-evaluate (s. BAHILL U. GISSING 1998, S. 516).

Ziel-Zustand entwickelt (s. Kapitel 7). Für den Schritt der Bewertung und Entscheidung gibt das Gestaltungsmodell Hinweise, hier liegt jedoch nicht der Fokus der vorliegenden Arbeit (s. Kapitel 8).

4.2.2 Modellbildung

Der **Modellbegriff** hängt eng mit der oben beschriebenen Systemtheorie zusammen (s. SCHÜTZ 2017, S. 17), da Modelle grundsätzlich die Grundlage zur Abbildung von Systemen bilden (s. ROPOHL 2012, S. 54). Damit ist der Modellbegriff grundlegend in der Wissenschaft (s. LEHNER ET AL. 2008, S. 29), es existieren aber unterschiedliche Definitionen für Modelle. In dieser Arbeit soll die gängige Definition nach STACHOWIAK gelten (s. DEINDL 2013b, S. 58; SCHÜTZ 2017, S. 17). Demnach gilt, wie in Abbildung 4-3 dargestellt: „X ist Modell des Originals Y für den Verwender k in der Zeitspanne t bezüglich der Intention Z“ (STACHOWIAK 1994, S. 219).

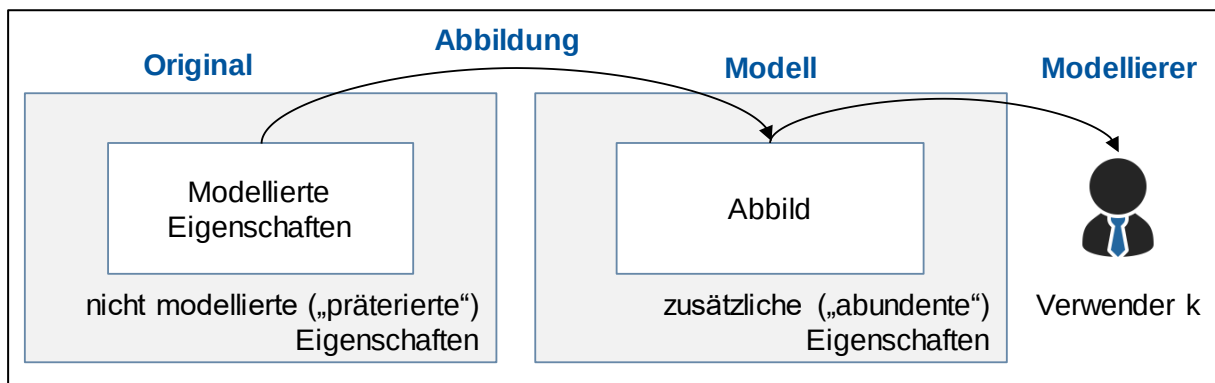


Abbildung 4-3: Original, Modell und Verwender (i. A. a. STACHOWIAK 1994, S. 219; HESSE U. MAYR 2008, S. 381)

Damit bilden Modelle die Realität ab und geben dem Verwender ein Hilfsmittel zum Denken (s. LEHNER ET AL. 2008, S. 29-30; JUNG 2016, S. 42). Bei dieser Abbildung kann es jedoch aufgrund der gegebenenfalls zu hohen Abstraktion zu Problemen der Ungenauigkeit kommen (s. COLLOPY 2015, S. 538). Bei jeder Modellbildung muss daher klar sein, dass **Modelle unter Annahmen** (Prämissen oder präferierte Eigenschaften) **des Verwenders entstehen** und nur einen bestimmten Teil der Wirklichkeit abbilden können (s. JUNG 2016, S. 42). Modelle müssen insofern mindestens strukturelle, funktionelle oder Verhaltensähnlichkeit mit der modellierten Wirklichkeit besitzen, um wissenschaftlichen Kriterien standzuhalten (s. STACHOWIAK 1989, S. 154-158 zit. nach LEHNER ET AL. 2008, S. 29).

Die bisher beschriebene **Abbildungsfunktion** eines Modells kann sehr gut mit dem Beispiel einer Landkarte verglichen werden. Landkarten bilden einen zweidimensionalen Ausschnitt eines gewissen Gebiets auf der Erde ab. Dabei werden auch zusätzliche Informationen wie Aussichtspunkte oder der Maßstab ergänzt, die so nicht in der Wirklichkeit vorkommen (s. ROPOHL 2012, S. 53-54). Nach ROPOHL können daraus die allgemeinen Modellmerkmale von STACHOWIAK abgeleitet werden. Das **Abbildungsmerkmal** beschreibt aufbauend auf der Mengenlehre der Mathematik, dass Modelle Abbildungen von Originalen oder anderen Modellen sind, die künstlicher oder realer

Natur sein können. Das **Verkürzungsmerkmal** besagt, dass Modelle nicht alle Eigenschaften ihres jeweiligen Originals aufzeigen, sondern nur die jeweils relevanten. Zuletzt beschreibt das **pragmatische Merkmal**, dass Modelle nicht nur Abbildung von einer bestimmten Wirklichkeit unter einer bestimmten Verkürzung sind, sondern auch für einen Zweck, eine Anspruchsgruppe und für einen bestimmten Gültigkeitszeitraum geschaffen sind (s. STACHOWIAK 1973, S. 131-133).

In der Wissenschaftspraxis hat sich, je nach Art des Untersuchungszwecks, eine Reihe von Modellen etabliert, die im Folgenden kurz vorgestellt werden sollen:

- **Beschreibungs- und Erfassungsmodelle** dienen der möglichst präzisen Abbildung von empirischen Beobachtungen. Dabei haben sie eine rein deskriptive Funktion, stellen also Sachverhalte nur dar und dienen nicht der Analyse. Die Gestaltung von Beschreibungsmodellen hängt von deren jeweiliger Aufgabe ab, auch wenn ihre tatsächliche Nutzung von der ursprünglich geplanten Aufgabe abweichen kann. Die im obigen Beispiel genannte Landkarte ist ein Beispiel für ein Beschreibungsmodell (s. BÖHM U. FUCHS 2002, S. 26; LEHNER ET AL. 2008, S. 30-31; JUNG 2016, S. 43).
- Auf Basis von Beschreibungsmodellen lassen sich **Erklärungsmodelle** aufstellen, um Zusammenhänge, Gesetzmäßigkeiten und Ursachen zu untersuchen. Dabei sind keine Einzelsituationen zu betrachten, sondern generelle Eigenschaften über mehrere Einzelsituationen hinweg. Erklärungsmodelle haben den Zweck, Beschreibungsmodelle zu verbessern, zukünftiges Verhalten zu prognostizieren oder ein Verhalten zu erklären. Daher sind Erklärungsmodelle meist hypothesengetrieben und bei einem empirischen Ansatz mit entsprechenden Datenerhebungen zu validieren (s. LEHNER ET AL. 2008, S. 31; JUNG 2016, S. 44).
- **Gestaltungsmodelle** bilden, aufbauend auf Beschreibungs- oder Erklärungsmodellen, die höchste Klasse an Modellen und unterstützen beim Erreichen von Zielen. Sie zeigen Gestaltungsalternativen auf und erweitern ein System damit um einen Sinn, einen Zweck oder ein Ziel. Ergänzt man diese Modelle um die Ziele eines Entscheidungsträgers, werden sie auch **Entscheidungsmodelle** genannt (s. LEHNER ET AL. 2008, S. 31-32; ZELEWSKI 2008, S. 46). Solche Modelle weisen in der Anwendung Grenzen auf, weil soziale Zusammenhänge und menschliches Verhalten, das bei Entscheidungen in Unternehmen immer eine große Rolle spielt, nicht vollständig in Modellen abgebildet werden können (s. JUNG 2016, S. 45-46).
- Modelle, die ein übergeordnetes Ziel verfolgen, sind als **Meta- oder Referenzmodelle** zu bezeichnen. Sie dienen als Blaupause für die Entwicklung von Modellen und fassen als „Modell von Modellen“ verschiedene andere Modelle zusammen (s. LEHNER ET AL. 2008, S. 32).

Veranschaulichung eines Untersuchungsbereichs hilft (s. ZELEWSKI 2008, S. 35). Speziell die morphologische Methode kann dabei für alle Arten von Problemen angewendet werden (s. ZWICKY 1966, S. 114).

Morphologie

Die **morphologische Methode** wird in der Wissenschaft schon lange eingesetzt. Schon GOETHE nutzte sie 1817 zur Beschreibung von verschiedenen Naturphänomenen, die er durch Erfahrungen und Betrachtungen kennengelernt hatte (s. GOETHE 1817). In neuerer Zeit hat sie über viele Jahre ZWICKY zuerst in der Astronomie eingesetzt und daraufhin als Kreativitätstechnik etabliert und beworben (s. ZWICKY 1948; 1957; 1966; 1972). In der aktuellen Literatur wird sie weiterhin als diskursive Kreativitätstechnik dargestellt, um systematisch ein Problem in Bestandteile zu zerlegen und mögliche Lösungskomponenten zu identifizieren. Sie wird dabei mit dem Ziel genutzt, **möglichst viele Lösungen auf systematischem Weg zu entdecken** und diese vorbehaltlos zu betrachten (s. SCHULTE-ZURHAUSEN 2014, S. 618; HABERFELLNER ET AL. 2015, S. 386; SONNTAG 2015, S. 1). Zugleich wird die morphologische Methode in etwas abgewandelter Form aber auch dazu genutzt, **neue Themenbereiche zu strukturieren und zu standardisieren** (s. KEES 2015, S. 40-41; SCHUH U. GIERTH 2007b, S. 6).

Die morphologische Methode nutzt als Kernelement den **morphologischen Kasten**, wie er beispielhaft in Tabelle 4-1 gezeigt ist. Er besteht aus Merkmalen, denen jeweils mindestens zwei Ausprägungen zugeordnet sind. Dabei bilden die **Merkmale** voneinander unabhängige Einflussgrößen. Die **Ausprägungen** sollten alle, gegebenenfalls auch nur theoretisch denkbaren möglichen Lösungen enthalten. Um trennscharfe Ausprägungen zu erhalten, müssen diese unter Umständen auf höherer Ebene zusammengefasst werden. Insgesamt sollte die Matrix nicht zu groß werden, um den Bewertungsaufwand gering zu halten (s. SCHULTE-ZURHAUSEN 2014, S. 618).

Tabelle 4-1: Aufbau eines morphologischen Kastens zur Typisierung von Fahrzeugen am Beispiel des Typs „e.GO Life“, einem Elektrofahrzeug (eigene Darstellung)

Merkmal	Ausprägung 1	Ausprägung 2	Ausprägung 3	Ausprägung 4
Antrieb	Elektro	Diesel	Benzin	Gas
Farbe	einfarbig	mehrfarbig		
Sitze	2	3	4	5
Anhängerkupplung	ja	nein		
...	...	...	...	...

Zur **Durchführung der morphologischen Methode** wird ein Vorgehen in fünf, bei manchen Autoren auch sechs Schritten empfohlen. Es beginnt mit der **Definition der Fragestellung**. Diese sollte so konkret und genau wie möglich gehalten werden und bildet damit die Grundlage für die Bearbeitung der konkreten Aufgabe. Im zweiten Schritt werden die **Merkmale bestimmt** und beschrieben. Damit werden alle Lösungen ermittelt, die Einfluss auf die Fragestellung haben könnten. Auf dieser Basis können im dritten Schritt **Ausprägungen gesammelt** und damit der morphologische Kasten aufgestellt werden. Unter Nutzung des morphologischen Kastens können nun **Varianten gebildet** werden, indem vorurteilsfrei sinnvolle Kombinationen der Ausprägungen je Merkmal gewählt werden. Je Merkmal wird nach HABERFELLNER ET AL. genau eine Ausprägung gewählt, in einigen Werken wie beispielsweise bei SCHUH U. GIERTH wird auch die Auswahl mehrerer Ausprägungen je Merkmal erlaubt. Zuletzt sind die Varianten zu bewerten und für die Weiterentwicklung geeignete Varianten auszuwählen (s. ZWICKY 1966, S. 116-117; SCHUH U. GIERTH 2007a, S. 6; HABERFELLNER ET AL. 2015, S. 386; SONNTAG 2015, S. 1-3).

Anwendung von Morphologien in der vorliegenden Arbeit

Morphologien sind ein **zentrales Werkzeug anwendungsorientierter Forschung** und werden auch in der vorliegenden Arbeit für die Strukturierung von Realitätsausschnitten genutzt. Die Eigenschaften von software-definierten Plattformen (s. Kapitel 5.4), die in der Form noch nie beschrieben wurden, genauso wie die Eigenschaften von Informationssystemen, werden mithilfe von Morphologien beschrieben (s. Kapitel 6.6). Außerdem hilft die Morphologie bei der Strukturierung der Zusammenhänge zwischen Ist- und Ziel-Zustand der Informationssystem-Architektur (s. Kapitel 7.2).

Typisierung

Wie von KNOBLICH definiert, kann ein Typus als Muster verstanden werden, bei dem nur das Wesentliche der Erscheinung Ausdruck findet (s. KNOBLICH 1969, S. 25). Dem entsprechend wird die Typisierung dazu genutzt, Muster zu erkennen und deren **wesentliche Erscheinungsformen** bei einer Menge von Objekten **zu identifizieren**. Damit dient die Typisierung dazu, die Komplexität eines Untersuchungsbereichs zu reduzieren (s. WELTER 2006, S. 113). Aufgrund ihrer generalistischen Einsatzmöglichkeit ist sie als deskriptive Forschungsmethode unter anderem in der Betriebswirtschaftslehre (BWL) und der Soziologie schon seit Jahrzehnten anerkannt und wird auch im Informationsmanagement erfolgreich genutzt (s. FLEIß 2010, S. 4; DEINDL 2013a, S. 60-63; DÜNNEBACKE 2016, S. 69-73).

Die Methode der Typisierung unterscheidet sich von der morphologischen Methode vor allem darin, dass sie **faktisch wahre Lösungen** hervorbringt, indem sie die nur theoretisch denkbaren Alternativen eliminiert. Sie unterscheidet sich außerdem von der Klassifizierung, die nur *ein* klassenbildendes Merkmal benötigt, dadurch, dass *mehrere* Merkmale zu Typenbildung benötigt werden (s. KNOBLICH 1969, S. 27; WELTER 2006, S. 114).

Ergebnis einer Typisierung ist eine **Typologie**, die formalen und inhaltlichen Kriterien zu folgen hat. Formal hat eine Typologie den Kriterien der Widerspruchs- und Überschneidungsfreiheit zu entsprechen, außerdem müssen sich die Typen nur über die Kombination von Merkmalen bilden lassen. Allgemeine inhaltliche Qualitätskriterien lassen sich nur schwer bilden, daher müssen Typologien vor allem nachvollziehbar sein. Die Begründungen, warum genau welche Merkmale gewählt wurden, sind dazu darzulegen, um die Gedanken des Forschers, welche er bei der Bildung der Typen hatte, nachvollziehen zu können. Damit wird eine Rekonstruierbarkeit der Typologie ermöglicht (s. FLEIß 2010, S. 5-7).

Das **Vorgehen der Typisierung** kann dabei intuitiv-konstruktiv sein. Hierbei werden Typen durch Überlegung und Vergleich oder durch deduktive Konstruktion gewonnen. Sie können aber genauso durch eine analytische Zerlegung oder synthetische Zusammenfügung von Merkmalen gebildet werden (s. KNOBLICH 1969, S. 30-31). Das Vorgehen zur Bildung des morphologischen Kastens (s. o.) aufgreifend, werden die Typen im vierten Schritt als solche gebildet, die **praktisch brauchbar** sind. Es folgt danach noch ein Schritt der **grafischen Darstellung** der Typen durch entsprechende Visualisierung (s. WELTER 2006, S. 115-116).

Anwendung der Typisierung für die vorliegende Arbeit

Mit dem Vorgehen von WELTER werden in der vorliegenden Arbeit – anhand der erhobenen software-definierten Plattformen, auf Basis der Morphologie von deren Eigenschaften – Typen software-definierter Plattformen abgeleitet (s. Kapitel 5.5). Diese Typen dienen der grundsätzlichen Einordnung existierender Plattformen und helfen dabei, die in Kapitel 7.3 und 8.1 vorgestellten Entscheidungskriterien und Gestaltungshinweise allgemeingültig diesen Plattfortmtypen zuzuordnen.

Eine weitere Typisierung wird für Informationssystem-Architekturen vorgenommen. Diese wird indirekt von der Morphologie der Eigenschaften von Informationssystemen abgeleitet, indem aus der Vielzahl an Informationssystemen Informationssystem-Architekturen zusammengestellt und diese anhand zweier bestimmter Eigenschaften typisiert werden (s. Kapitel 6.8).

4.3 Konkretisierung der Vorgehensweise

In den obigen Kapiteln werden Ziele an die Ergebnisse der vorliegenden Arbeit sowie die eingesetzten Methoden beschrieben. Im Folgenden soll dargestellt werden, wie die Methoden konkret dazu eingesetzt werden, die inhaltlichen und formalen Anforderungen zu erfüllen, um Informationssystem-Architekturen bei software-definierten Plattformen zu gestalten. Dabei fordern die formalen Ziele, insbesondere das Ziel der Validität und Nachvollziehbarkeit, ein prinzipiell systematisches Vorgehen bei der Modellbildung. Die Arbeit orientiert sich dazu an den **Grundsätzen ordnungsgemäßer Modellierung von SCHÜTTE**, die fünf Schritte zur Modellbildung vorsehen (s. SCHÜTTE 1998,

S. 184-188). Dabei werden die Prinzipien der Modellbildung eingesetzt und die Systemtheorie bildet den Referenzmodellrahmen (s. Kapitel 4.2.1 u. 4.2.2). Das Vorgehen ist in Abbildung 4-5 zusammengefasst.

Die entwickelten **Modelle haben dabei eine Abbildungs- und Vorbildfunktion** für die Realität. Die Abbildungsfunktion ist aufgrund der Modelleigenschaft inhärent gegeben (s. HESSE U. MAYR 2008, S. 381). Die Vorbildfunktion ist dadurch gegeben, dass die Modelle, insbesondere das Erklärungsmodell in Kapitel 7.1, einen Ziel-Zustand abbilden, den produzierende Unternehmen anstreben sollten (s. ALTENDORFER-KAISER 2016, S. 169).

Methoden	Vorgehen nach SCHÜTTE	Kapitel der Arbeit
Expertengespräche, Vorwissen, Literaturanalysen	1. Problemdefinition - Definition der Ziele und des Nutzens - Abgrenzung des Untersuchungsbereichs	Ausgangssituation (1) Grundlagen und Abgrenzung (2) Stand der Erkenntnisse (3)
Literaturanalysen, Modellierung, Systemtheorie, Experteninterviews, eigene Erfahrung (Aktionsforschung)	2. Konstruktion des Modellrahmens - Anforderungsidentifikation und Vorgehensdefinition - Entwicklung eines Bezugsrahmens	Konkretisierung der Vorgehensweise (4)
	3. Konstruktion der Modellstruktur - Entwicklung von konfigurierbaren Partialmodellen - Validierung	software-definierte Plattformen (5) Informationssystem-architekturen (6) Ist- und Ziel-Zustand (7)
	4. Komplettierung des Modells - Entwicklung eines integrierten Modells - Intra- und Intermodelkonsistenz	Gestaltung des Integrationsprozesses (8)
formale Überprüfung, Fallstudien	5. Anwendung des Modells formale Evaluation und empirische Realisierungsversuche mittels Fallstudien	fallstudienbasierte Evaluation (9)

Abbildung 4-5: Generelle Vorgehensweise der vorliegenden Arbeit (i. A. a. SCHÜTTE 1998, S. 184-188; FLUHR 2014, S. 79)

Konkret beginnt die Vorgehensweise mit der **Problemdefinition** (1). Dazu wurde zu Beginn dieser Arbeit die Ausgangssituation (s. Kapitel 1) vorgestellt und anhand des Stands der Erkenntnisse die wissenschaftliche Lücke identifiziert. Letzteres wurde anhand von umfangreichen Literaturanalysen in den Themenbereichen IT-Komplexität, Informationssystem-Architekturen, software-definierte Plattformen und Plattformintegration durchgeführt (s. Kapitel 3). Die Abgrenzung des Untersuchungsbereichs wurde auf Basis von grundlegenden Definitionen in Kapitel 2 vorgenommen und mit Expertengesprächen in der Praxis validiert (s. Anhang A.3).

Die **Konstruktion des Modellrahmens** (2) beginnt mit der Identifikation von Anforderungen an die Vorgehensweise. Dazu wurden sowohl inhaltliche als auch formale Anforderungen an die Methodik vorgestellt sowie die methodischen Grundlagen beschrieben, mit denen ein nachvollziehbares Forschungsergebnis erzielt werden kann (s. Kapitel 4.1 u. 4.2). SCHÜTTE empfiehlt in diesem Vorgehensschritt die Nutzung eines Referenzmodellrahmens (s. SCHÜTTE 1998, S. 187). Für die vorliegende Arbeit kann die Systemtheorie nach HABERFELLNER ET AL., im Speziellen der dort beschriebene Problemlösungszyklus, als ein solcher Referenzmodellrahmen angesehen werden. Die einzelnen Modelle der Arbeit bilden nicht nur die Bestandteile des Problemlösungszyklus ab, sondern dieser bildet auch die Grundlage für das Gestaltungsmodell (s. Kapitel 8).

In den auf das vorliegende Kapitel folgenden Kapiteln 5 und 6 wird **die Konstruktion der Modellstruktur** (3) beschrieben. Dazu werden Beschreibungsmodelle für software-definierte Plattformen und deren Integrationsanforderungen sowie für Informationssystem-Architekturen produzierender Unternehmen entwickelt.

Das **Beschreibungsmodell für software-definierte Plattformen** basiert dabei einerseits auf der Untersuchung von in der Praxis verfügbaren Plattformen (s. Anhang A.5) andererseits auf eigenen Erfahrungen des Autors in der Anwendung der Plattformen (s. STICH ET AL. 2018, S. 39). Dies entspricht den Ansätzen der Aktionsforschung, bei dem der Wissenschaftler und die von ihm entwickelten Modelle selbst Teil der praktischen Anwendung sind (s. MCNIFF U. WHITEHEAD 2011, S. 9; MCNIFF 2017, S. 6). Da der Begriff *software-definierte Plattformen* im Kontext der Anwendung im *Internet of Production* bisher nicht ausreichend definiert war, wird zuerst eine eigene Definition vorgenommen (s. Kapitel 5.1). Daraufhin werden Merkmale dieser Plattformen mithilfe der morphologischen Methode (s. Kapitel 4.2.3) erarbeitet. Die Ableitung der Merkmalsausprägungen erfolgt sachlogisch auf Basis der untersuchten Plattformen, ebenso die Ableitung der Plattfortmtypen (s. Kapitel 5.4 und 5.5). Die Plattfortmtypen wiederum bilden die Basis für eine strukturierte Deduktion der Integrationsanforderungen (s. Kapitel 5.6). Als Ergebnisvorschau wurden die zentralen Inhalte dieses Modells vom Autor zusammen mit anderen Autoren in STICH ET AL. 2018 veröffentlicht.

Auch das **Beschreibungsmodell der Informationssystem-Architekturen** produzierender Unternehmen basiert neben bestehenden Modellen in der Literatur (s. Kapitel 3.3) auf Experteninterviews und Erfahrungen des Autors. Eine eigene Definition für Informationssysteme muss in dem Fall nicht vorgenommen werden, stattdessen stützt sich das Modell auf die Definition von KRCMAR (s. Kapitel 6). Die Merkmale von Informationssystemen werden darauf basierend mit dem Fokus auf die Integrationsfähigkeit von Informationssystemen abgeleitet (s. Kapitel 6.1). Die Ausprägungen der Merkmale basieren auf Literaturquellen und der sachlogischen Ableitung (s. Kapitel 6.6) anhand von Projekterfahrungen, die in den Kapiteln 6.8 und 9.2 vorgestellt werden. Informationssystem-Architekturen werden daraufhin als multidimensionale Räume, in denen die Informationssysteme durch die Merkmalsausprägungen eingeordnet sind, beschrieben. Unter Nutzung von EAM-Methoden werden Darstellungsformen von Informationssystem-Architekturen vorgestellt und mithilfe von realen Beispielen typisiert (s. Kapitel 6.7 u. 6.8).

Die **Komplettierung des Modells** (4) erfolgt durch das **Erklärungsmodell** in Kapitel 7. Es integriert die beiden vorgenannten Modelle und erklärt die Zusammenhänge zu einem Ziel-Zustand, der in diesem Kapitel vorgestellt wird. Dazu wird die Situationsanalysemethode aus dem Problemlösungszyklus von HABERFELLNER ET AL. auf die Herausforderung der Informationssystem-Architekturgestaltung angewendet (S. HABERFELLNER ET AL. 2015). Hierbei erfolgt auch die Inter- und Intramodellkonsistenzprüfung, wie von SCHÜTTE gefordert (S. SCHÜTTE 1998, S. 188). Als Antwort auf die Integrationsanforderungen wird dazu eine Zielarchitektur, basierend auf allgemeinen Anforderungen an Informationssystem-Architekturen aus der Literatur (s. Kapitel 3.4), erstellt (s. Kapitel 7.1). Wiederum unter Nutzung der morphologischen Methode wird die Vergleichbarkeit der einzelnen Informationssystem-Architekturtypen zur Zielarchitektur hergestellt und beschrieben sowie spezifische Entscheidungskriterien erläutert (s. Kapitel 7.2 u. 7.3). Als Ergebnisvorschau wurden auch zentrale Elemente der Modelle aus den Kapiteln 6 und 7 vom Autor zusammen mit einem Co-Autor in HOFFMANN U. HEIMES 2018 veröffentlicht.

Die Modellvervollständigung endet mit dem **Modell zur Gestaltung des Integrationsprozesses software-definierter Plattformen**. Dieses beschreibt ein Vorgehen für die Anwendung der vorgenannten Modelle in der industriellen Praxis und damit einen vollständigen Mikrozyklus im Problemlösungszyklus (s. Kapitel 8). Es leitet über in die konkrete **Anwendung der Methodik** (5) in empirischen Untersuchungen durch Fallstudien. Dabei werden die entwickelten Modelle exemplarisch eingesetzt und damit verifiziert (s. Kapitel 9).

5 Beschreibung software-definierter Plattformen

Wie in Kapitel 4.3 dargestellt, liegt der erste Schritt zur Entwicklung einer Informationssystem-Architektur produzierender Unternehmen bei software-definierten Plattformen in der Detaillierung des Ziel-Zustands. Die Vision des Plattformeinsatzes aus Kapitel 2.3 wird dabei in diesem Kapitel aufgegriffen und vertieft. Das Feld an möglichen Softwaretypen, welche potentiell als *software-definierte Plattform* angesehen werden könnten ist weit. So können unter anderem Middleware- und EAI-Systeme, BI-Systeme, sowie ECM- oder PLM-Systeme mit Unternehmensdaten umgehen und diese zentral verwalten. Um dieses Feld auf die relevanten Systeme einzuschränken, wird in diesem Kapitel zuerst eine ausführliche Definition von software-definierten Plattformen für produzierende Unternehmen im Kontext des Aachener *Internet of Production* (vgl. Kapitel 2.3) vorgenommen (s. Kapitel 5.1). Auf dieser Basis wird ein Beschreibungsmodell entwickelt, mit dem software-definierte Plattformen strukturiert werden können. Dazu werden Merkmalsdimensionen hergeleitet (s. Kapitel 5.2) und die beschreibenden Merkmale dargestellt (s. Kapitel 5.3) und zusammengefasst (s. Kapitel 5.4). Anhand der Beschreibung werden verschiedene Einsatzkonzepte als Typen definiert und die Integrationsanforderungen dafür dargestellt (s. Kapitel 5.6).

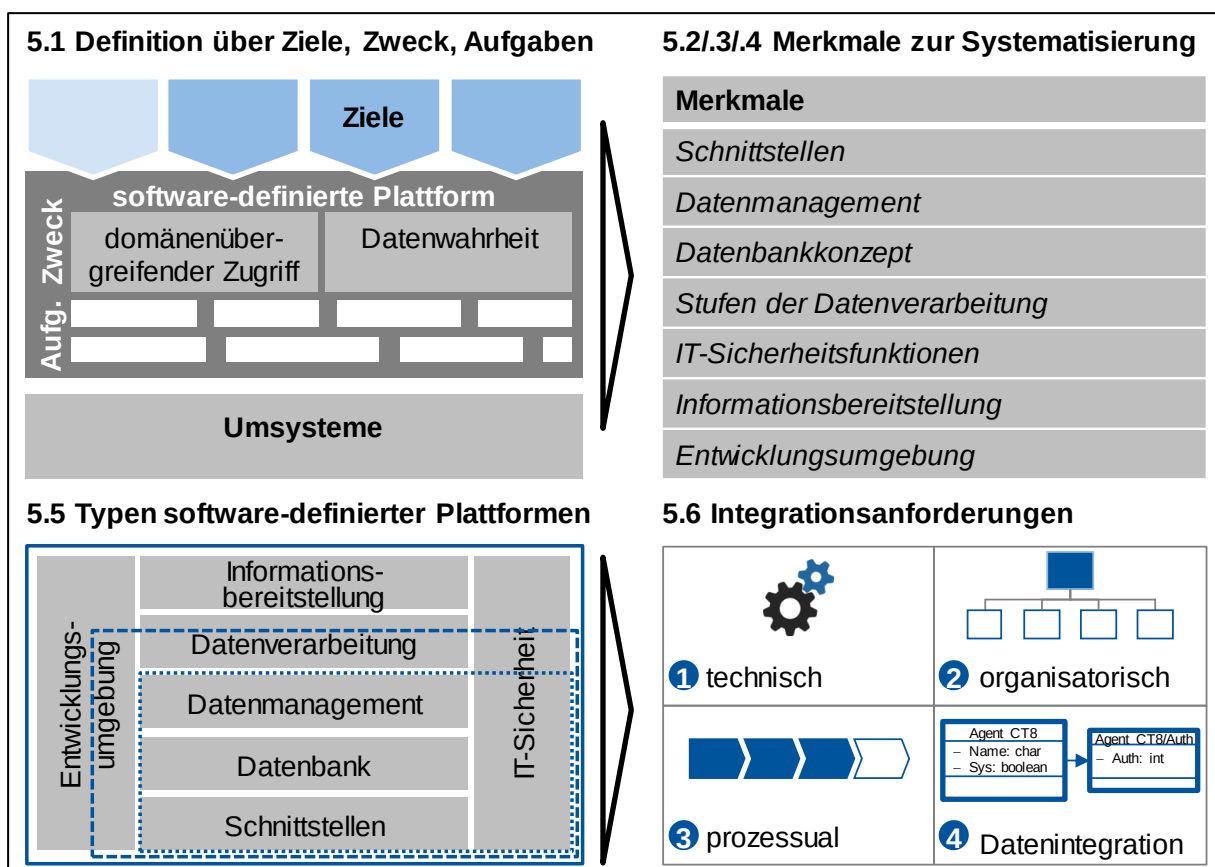


Abbildung 5-1: Übersicht und Zusammenhang der Inhalte des Kapitels 5 (eigene Darstellung)

Dieses Kapitel beantwortet damit die Forschungsunterfrage: **Wie können zentrale software-definierte Plattformen und deren Integrationsanforderungen systematisch beschrieben werden?**

5.1 Definition des Begriffs *Software-definierte Plattform*

5.1.1 Herleitung der Definition

Die Begriffsdefinition in Kapitel 2.3 sowie der Stand der Erkenntnisse zur Beschreibung von software-definierten Plattformen in Kapitel 3.1 haben gezeigt, dass der Begriff „Software-definierte Plattform“ bisher nur grob definiert ist. Der Begriff selbst, in seiner aktuellen Deutung, wurde vom ARBEITSKREIS SMART SERVICE WELT sowie der ACATECH eingeführt und dort spezifisch auf die Anwendung auf produktorientierte, smarte Dienstleistungen ausgelegt. Andere Autoren haben ihn danach wieder aufgegriffen, aber nicht weiter vertieft (vgl. FREY 2015; CASTILLO ARAUJO 2015; ENGELS ET AL. 2017). Software-definierte Plattformen können jedoch sowohl für vernetzte Produkte als auch zur Optimierung von Prozessen und Abläufen eingesetzt werden (s. RICHARDSON U. RYMER 2016, S. 3). Um eine solche Vertiefung des Begriffs und letztendlich eine Definition herzuleiten, wird das Vorgehen wie in Abbildung 5-2 dargestellt angewendet: Der Begriff wird zuerst in die Begriffswelt des Oberbegriffs eingeordnet, dann erklärt und darauf aufbauend weiter für produzierende Unternehmen konkretisiert.

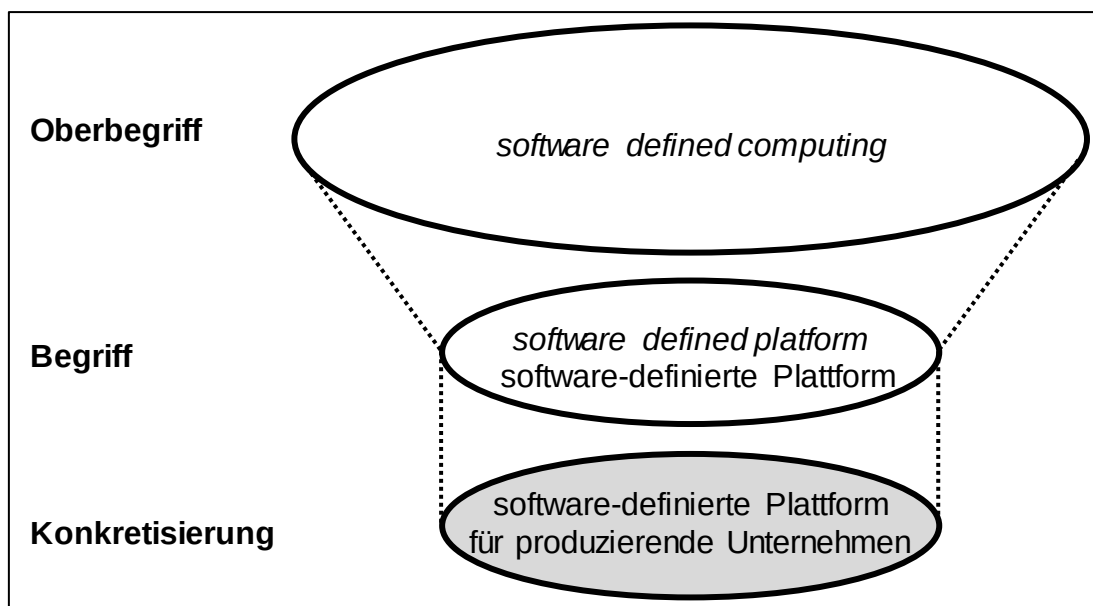


Abbildung 5-2: Vorgehen zur Begriffsdefinition der software-definierten Plattform (eigene Darstellung)

Die *software-definierte Plattform*, beziehungsweise in der angelsächsischen Literatur die *software defined platform*, gliedert sich ein in den **Oberbegriff** des *software defined computing*. Darunter wird eine Systeminfrastruktur verstanden, die durch dynamische Konfiguration und Programmierung automatisierbar betrieben werden kann. Die Systeminfrastruktur bezieht sich dabei auf Rechenkapazitäten, Speicherkapazitäten und das Netzwerk (s. LI ET AL. 2014, S. 1). Seinen Ursprung hat das *software defined computing* in der Virtualisierung der Systeminfrastruktur, die, wenn sie als Dienstleistung über das Internet angeboten wird, als *cloud computing* bezeichnet wird (s. PRODAN U. OSTERMANN 2009, S. 17). Ziel dabei ist es, jeweils das Gerätemanagement von der Hardware zu trennen. So wird beispielsweise in einem *software defined network* das

Management von Switches und Routern über einen Server durchgeführt, statt einer Gerätekonfiguration direkt über die Programmierschnittstellen der Hardware. Dadurch wird ein optimierter und agiler Netzbetrieb möglich, da Anpassungen in viel niedrigerer Granularität möglich sind als bei einer manuellen Konfiguration (s. LI U. LIAO 2013, S. 113).

Einen Schritt weiter gehen aktuelle Diskussionen, die von einem *software defined environment* oder *software defined everything* sprechen. Hierbei werden nicht nur einzelne Komponenten virtualisiert und über Software konfiguriert, sondern eine darüberliegende Ebene koordiniert und optimiert die darunterliegenden virtualisierten, software-definierten Komponenten (s. LI ET AL. 2014, S. 2; CASTILLO ARAUJO 2015, S. 5; IIDA 2016, S. 6). Dieser Ansatz wird nach Meinung von FUJITSU künftig dabei unterstützen können, die Verbindung zwischen einzelnen Cloud-Computing-Anwendungen herzustellen und diese zu managen (s. IIDA 2016, S. 1-2).

Damit kann die Eigenschaft der „Software-Definiertheit“ des **Begriffs** „Software-definierte Plattform“ verstanden werden. Innerhalb eines Unternehmens hat eine software-definierte Plattform nicht zur Aufgabe, selbst eine Virtualisierung von Hardwarekomponenten durchzuführen, sondern setzt auf vernetzten Produkten und auf vorhandenen Informationssystemen auf. Wie beim *software defined computing* bilden software-definierte Plattformen damit einen Abstraktionslayer, der bei der Koordination und gemeinsamen Nutzung der darunterliegenden Systeme hilft. Der zweite Begriffsbestandteil der „Plattform“ soll, wie in Kapitel 2.3 dargestellt, entsprechend der Definition von BAUMS verstanden werden. Eine (digitale) Plattform dient dementsprechend hauptsächlich dazu, verschiedenste Systeme miteinander zu verknüpfen, um aus deren Einzeltechnologien, -dienstleistungen und -daten komplementäre Dienste anbieten zu können (s. BAUMS 2015, S. 15).

Technisch betrachtet beschreibt der Begriff einer software-definierten Plattform also ein Softwaresystem, das zur Verknüpfung verschiedenster Systeme in ein Unternehmen eingeführt wird, um aus der Zusammenschaltung der Systeme neue, von diesen Systemen unabhängige Dienste nutzen zu können. Weiterhin hat die software-definierte Plattform die Aufgabe der Orchestrierung und Konfiguration der mit ihr verbundenen Systeme.

Die Definition der ACATECH hebt „die Virtualisierung als Schlüsselprinzip“ hervor (ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2016, S. 71). Dies muss entsprechend dem eben vorgestellten *software defined computing* verstanden werden und darf nicht mit einer bestimmten Form der Virtualisierung der Hardware, auf der eine software-definierte Plattform installiert ist, verwechselt werden. Zwar muss eine software-definierte Plattform die Ansprüche an ein performantes und skalierbares Informationssystem erfüllen, ob dies jedoch auf einer *On-Premises* Installation, in einer *private* oder *public Cloud* oder einer anderen Form einer *X-as-a-Service*-Leistung geschieht, ist für diese Definition irrelevant.

Zur **Konkretisierung** der technischen Definition sollen im Folgenden Ziel, Zweck und Aufgaben einer software-definierten Plattform in produzierenden Unternehmen, abgeleitet aus der in Kapitel 2.3 vorgestellten Vision des Plattformeinsatzes, dargestellt werden. Abbildung 5-3 gibt eine Übersicht über diese Definitionsbestandteile.

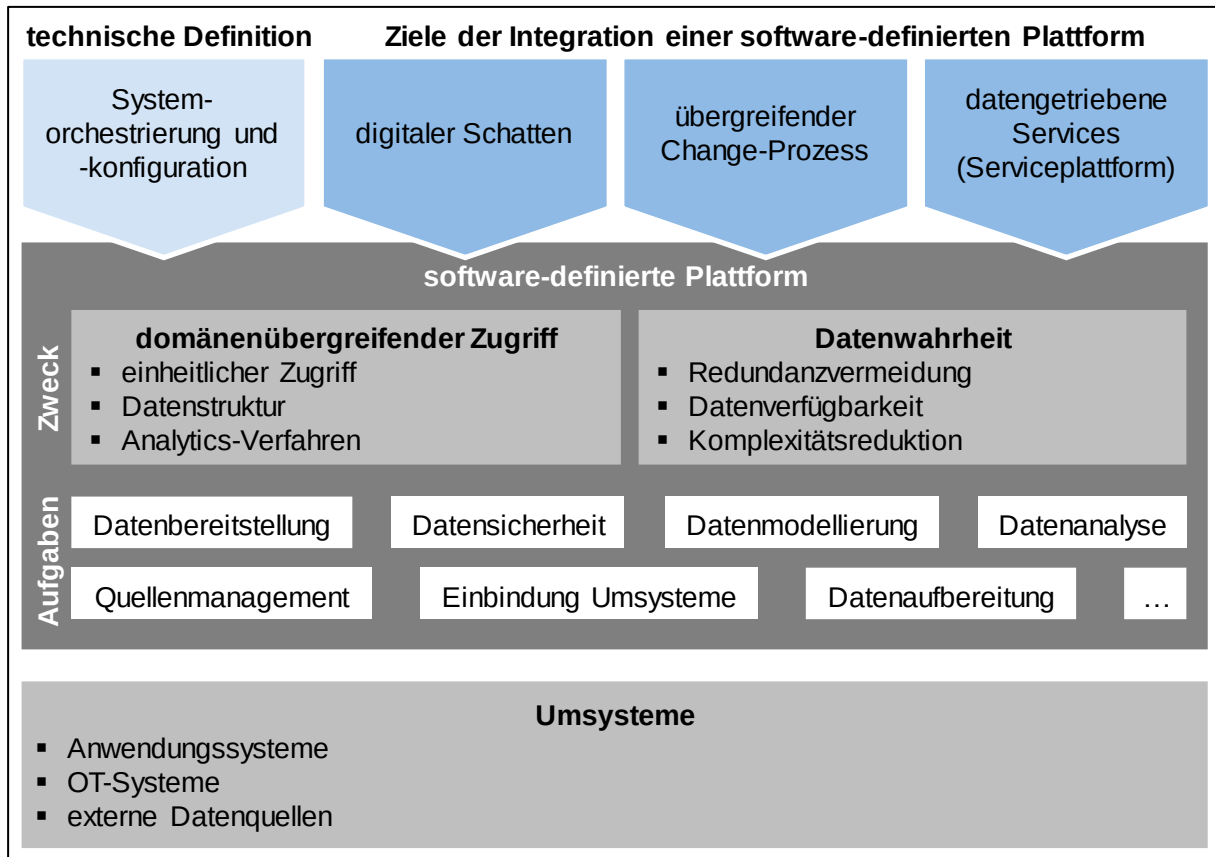


Abbildung 5-3: Inhalte der Definition einer software-definierten Plattform (eigene Darstellung)

Als genereller Ermöglicher von Industrie 4.0 (s. SCHUH ET AL. 2016b, S. 11) ist das erste **Ziel** der Umsetzung software-definierter Plattformen die **Ermöglichung von digitalen Schatten**. Der Begriff steht hierbei nach der WGP-Definition von SCHUH ET AL. für „ein hinreichend genaues, digitales Abbild der Prozesse, Information und Daten eines Unternehmens“ (SCHUH ET AL. 2016b, S. 11). Je nach Anwendungsbereich kann es dabei nicht nur einen, sondern mehrere digitale Schatten im Unternehmen geben, beispielsweise den des Produktionsprozesses, eines Kundenakquise-Prozesses (auch: *Customer Journey*) oder auch des Energieverbrauchs. Ein digitaler Schatten hat die Aufgabe, die relevanten Daten in die virtuelle Welt zu überführen. Dazu ist ein relationales Datenmodell nötig, auf dem Data-Analytics-Verfahren eingesetzt werden können, um die Daten auswählen und auswerten zu können. Um ein solches Datenmodell zu erzeugen, ist die Kenntnis der darunterliegenden Systeme, Prozesse und Daten elementar (s. SCHUH U. BLUM 2016, S. 2164-2166; SCHUH ET AL. 2016a, S. 48-51). Ein digitaler Schatten wird also eingesetzt, um Transparenz über aktuelle und vergangene Ereignisse erzielen zu können.

Ein zweites Ziel, das durch die Einführung einer software-definierten Plattform erreicht werden soll, ist die Möglichkeit, einen **übergreifenden Change-Prozess umsetzen**

zu können (s. dazu auch Kapitel 9.3.4). Da Änderungen von Produkten immer häufiger auftreten und auch gewünscht sind, um den Kundennutzen zu erhöhen, sind schnelle und effektive Engineering-Changes notwendig. Einerseits müssen Unternehmen dazu agil und lernfähig werden, andererseits wird die entsprechende Informationssystem-Architektur benötigt. Da ein Change nahezu alle Bereiche eines Unternehmens betrifft, sind dazu bereichsübergreifende Workflows elementar. Um im Workflow die richtigen Entscheidungen treffen zu können, bedarf es außerdem aktueller und direkt verfügbarer entscheidungsrelevanter Daten. Diese müssen unternehmensübergreifend in abgestimmten Daten- und Produktdatenstrukturen bereitgestellt werden und greifen damit auf den digitalen Schatten zurück (s. SCHUH ET AL. 2017h, S. 113-126; SCHUH ET AL. 2017d, S. 342-344).

Zuletzt reiht sich eine software-definierte Plattform ein in das Schichtenmodell der digitalen Infrastrukturen der *Smart Service Welt* (s. Kapitel 2.3). Service-Plattformen greifen dabei auf die Funktionen von software-definierten Plattformen zurück, um Kunden **datengetriebene Services bereitstellen** zu können.

Eine software-definierte Plattform, welche die vorgenannten Ziele erfüllen soll, wird also zu folgenden **Zwecken** eingeführt: Hauptsächlich soll der *domänenübergreifende Zugriff* auf aktuelle Daten ermöglicht werden. Dies beinhaltet den Zugriff auf alle relevanten Anwendungs- und OT-Systeme im Unternehmen sowie eine entsprechende Datenstruktur. Der Datenzugriff muss multimodal ermöglicht werden, also unabhängig von den Gerätetypen, den Benutzern und den Systemen, welche die angeforderten Daten bereitstellen oder weiternutzen. Gleichzeitig muss die software-definierte Plattform Auswertungsmöglichkeiten nahezu in Echtzeit schaffen und dazu entsprechende Analytics-Verfahren beherrschen.

Durch den domänenübergreifenden Zugriff wird die **software-definierte Plattform zu einer Quelle für Datenwahrheit**. Ist eine solche Datenwahrheit im Unternehmen erstmal verfügbar, so ändert sich der Umgang mit Daten insgesamt (s. SCHUH ET AL. 2017g, S. 8-20). Daten müssen dann nicht mehr redundant in verschiedenen Systemen vorgehalten werden, sondern können immer aus der einen Quelle genutzt werden. Ein weiterer Zweck ist also, Datenredundanz zu verringern und die mit der Datenredundanz verbundenen Probleme, die zu Fehlentscheidungen führen können, zu mindern. Letztendlich wird dadurch auch der Zweck der IT-Komplexitätsreduktion verfolgt, da unnötige Schnittstellen und Schatten-IT vermieden werden (s. SCHUH ET AL. 2017e, S. 63-64; SCHUH ET AL. 2017g, S. 20).

Tabelle 5-1: Aufgaben einer software-definierten Plattform (eigene Darstellung i. A. a. PORTER U. HEPPELMANN 2014, S. 71; ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2016, S. 71-80; SCHUH ET AL. 2017e, S. 63-64; SCHUH ET AL. 2017h, S. 118-121; KLOCKE ET AL. 2017b, S. 266-268; BULLINGER ET AL. 2017, S. 100-102)

Aufgabe	Beispiele	
Datenbereitstellung	Sicherstellung Performance	Echtzeitverfügbarkeit
Einbindung Umsysteme	Anwendungssysteme	OT-Systeme
Quellenmanagement	Plug and Automate	Synchronisation mit der Quelle
Datenmodellierung	Aggregation der Daten	Integration von Daten
Datenanalyse	Machine-Learning	Big Data
Datenaufbereitung	schnelle Erstellung von „Business-Apps“	lernende Entscheidungsunterstützung
Datensicherheit	Schutz der Übertragung	Schutz der Datenablage

Damit können zuletzt die **Aufgaben** festgelegt werden, die eine software-definierte Plattform erfüllen muss (s. Tabelle 5-1 und Abbildung 5-3). Die erste Aufgabe ist die **Datenbereitstellung** der von der software-definierten Plattform verwalteten Daten für verschiedene Anwendungszwecke. Dazu muss sie über hochperformante Speicher- und Lesemechanismen verfügen sowie die intelligente Datenverteilung für die notwendige verteilte Speicherung der Daten beherrschen (s. ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2015, S. 80; BULLINGER ET AL. 2017, S. 101). Zudem muss die Architektur auch über die Möglichkeit verfügen, **hocheffiziente Übertragungstechniken**, wie den neuen Mobilfunkstandard 5G, anbinden zu können. Dieser ermöglicht beispielsweise eine direkte Kommunikation zwischen zwei Systemen (engl. *Device-to-Device Communication*) und sich selbst organisierende Netzwerke (s. SHARMA U. SINGH 2016, S. 56-57), die direkt von der Plattform angesteuert werden sollen.

Es ist weiterhin Aufgabe einer software-definierten Plattform, **Umsysteme einzubinden**. Die wichtigsten Schnittstellen sind sicherlich die zu den Anwendungssystemen in der Informationssystem-Architektur des Unternehmens. Daten aus zum Beispiel ERP- und PLM-Systemen müssen über die software-definierte Plattform zugreifbar und auswertbar sein (s. SCHUH ET AL. 2017h, S. 120). Weiterhin müssen Schnittstellen zu Systemen auf dem Shopfloor des Unternehmens, also OT-Systemen, bereitgestellt werden, um direkten Echtzeitzugriff auf die Daten der Produktion zu erhalten. Hierzu können Standards wie OPC-UA genutzt werden (s. Kapitel 2.1.4 und SCHUH ET AL. 2017e, S. 64). Auch externe Quellen, welche beispielsweise Energieverbrauchs-, Handels-, Börsen- oder Wetterdaten bereitstellen, sollten über Schnittstellen eingebunden werden können (s. PORTER U. HEPPELMANN 2014, S. 71). Zuletzt sind Schnittstellen zu Serviceplattformen, wenn es solche im Unternehmen gibt, notwendig (s. ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2015, S. 72).

Dem **Management der Quellen**, die über Schnittstellen verbunden sind, kommt ebenfalls eine wichtige Bedeutung zu. Neben dem oben angesprochenen Prinzip der Systemkonfiguration und -orchestrierung schlägt der ARBEITSKREIS SMART SERVICE WELT hierzu den Enabler des „*Plug and Automate*“ vor, das dazu dient, Schnittstellen automatisch zu konfigurieren und Funktionen bereitzustellen (s. ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2015, S. 74). Eine weitere Aufgabe ist die Synchronisation von Daten, die in der software-definierten Plattform aggregiert oder per Nutzereingabe erstellt wurden, zurück in die Quellsysteme. Hierzu müssen intelligente Regeln implementiert sein, die verhindern, dass Daten unzulässigerweise überschrieben werden und es damit zu Inkonsistenzen kommt (s. SCHUH ET AL. 2017h, S. 120).

Software-definierte Plattformen haben sodann die **Aufgabe der Datenmodellierung**. Jene müssen es möglich machen, Daten zu normalisieren, indem zum Beispiel Messwerte auf dieselbe Einheit umgerechnet werden. Weiterhin muss die Datenmodellstruktur dergestalt sein, dass sowohl historische Daten gesichert als auch aktuelle Daten in Echtzeit verarbeitet werden können (s. PORTER U. HEPPELMANN 2014, S. 71). Um dies umzusetzen, müssen Datenpunkte im Datenmodell sowohl aggregiert werden als auch aus verschiedenen Quellen angereichert werden, um zu Smart Data zu werden (s. KLOCKE ET AL. 2017b, S. 267; Kapitel 2.1.1).

Unter Nutzung verschiedenster mathematischer Verfahren kommen einer software-definierten Plattform die Aufgaben der **Datenanalyse** und **Datenaufbereitung** zu. Sie muss dazu über intelligente Algorithmen verfügen, wie beispielsweise Machine-Learning, Data-Mining oder die Auswertung großer Datenmengen (*Big Data*). Da die Daten letztendlich genutzt werden, um Entscheidungen zu treffen, sollten auch Algorithmen zur selbstlernenden Entscheidungsunterstützung Teil der software-definierten Plattform sein (s. ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2015, S. 73; BULLINGER ET AL. 2017, S. 102; SCHUH ET AL. 2017h, S. 120-121). PORTER U. HEPPELMANN ergänzen, dass die Möglichkeit, schnell und einfach Apps zu erzeugen, ebenfalls durch die software-definierte Plattform unterstützt werden muss. Dazu muss sie einfachen Datenzugriff ermöglichen und die Vorbereitung für einfache Visualisierungswerkzeuge schaffen, indem beispielsweise gewisse Auswertungen berechnet oder Views auf Datenbestände vorgehalten werden (s. PORTER U. HEPPELMANN 2014, S. 71).

Eine Stelle, an der alle relevanten Daten eines Unternehmens zentral verfügbar sind, ist ein lukratives Angriffsziel. Daher ist es nicht zuletzt Aufgabe einer software-definierten Plattform, für den **Schutz und die Sicherheit der Daten** und Übertragungswege zu sorgen. Gleichzeitig darf darunter die Nutzbarkeit des Systems nicht leiden. Weitere Sicherheitsfunktionen, welche die Schnittstellen vor unberechtigtem Zugriff schützen sowie die unerlaubte Veränderung von Daten überwachen, sind ebenfalls Aufgabe einer software-definierten Plattform (s. PORTER U. HEPPELMANN 2014, S. 71; ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2015, S. 74).

5.1.2 Erweiterte eigene Definition und Einordnung in die Infrastruktur des *Internet of Production*

In Erweiterung der Definition der *Smart Service Welt* (s. ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH 2015, S. 17) und entsprechend der Herleitung in Kapitel 5.1.1 soll die folgende Definition einer software-definierten Plattform für diese Arbeit genutzt werden, die der Autor zusammen mit anderen Autoren im Vorfeld der Arbeit veröffentlicht hat:

Software-definierte Plattformen sind selbst Informationssysteme und bilden die technologische Integrationsschicht für Umsysteme wie weitere heterogene Informationssysteme und Datenquellen. Sie ermöglichen damit einen domänenübergreifenden, einheitlichen Datenzugriff auf eine Quelle der Datenwahrheit. Dazu stellen sie Schnittstellen zu Umsystemen bereit und konfigurieren diese aktiv; sie strukturieren Daten anhand generischer und spezifischer Datenmodelle und analysieren historische und Echtzeitdaten mit modernen Data-Analytics-Verfahren. Sie bereiten Daten auf und stellen diese Entscheidungssystemen bereit. Dabei stellen sie Funktionen bereit, um höchste Ansprüche an die Datensicherheit zu wahren. (STICH ET AL. 2018, S. 30)

Diese Definition lässt sich leicht in die in Kapitel 2.3 vorgestellte Infrastruktur des *Internet of Production* einordnen: Die software-definierte Plattform entspricht den in grün dargestellten Ebenen „*Middleware+*“ und „*Smart Data*“. Die dort enthaltenen Funktionen wurden in der Definition wieder aufgegriffen. Die in der Definition als „Umsysteme“ bezeichneten Systeme befinden sich in den beiden blauen Ebenen „*Smart Expert*“ und „*Applikationssoftware*“; letztere sind in dieser Arbeit als Anwendungssysteme definiert. Abbildung 5-4 veranschaulicht diese Zusammenhänge.

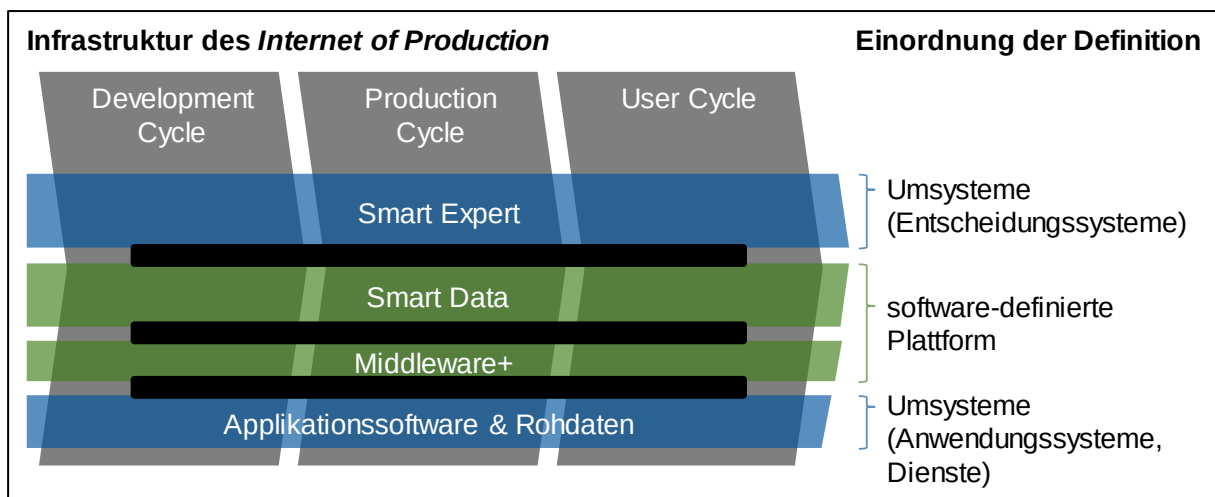


Abbildung 5-4: Einordnung der Definition in die Infrastruktur des *Internet of Production* (eigene Darstellung auf Basis SCHUH ET AL. 2017h, S. 121)

5.2 Herleitung der Merkmalsdimensionen

Nachdem der Begriff der software-definierten Plattform anhand der Ziele und Aufgaben hergeleitet und definiert ist, können darauf aufbauend **Merkmale zur Strukturierung von software-definierten Plattformen** herausgearbeitet werden. Wie in Kapitel 4.3 beschrieben, erfolgt diese Ausarbeitung mithilfe der morphologischen Methode. Dazu müssen, nach Festlegung der Fragestellung, voneinander unabhängige Merkmale definiert werden, welche die Fragestellung beeinflussen. Zur Erhebung dieser Merkmale wurden deshalb einerseits am Markt existierende Plattformen untersucht (s. Anhang A.5), andererseits die Merkmale auf Basis der Aufgaben der software-definierten Plattform, die in Kapitel 5.1.1 aufgezeigt wurden, zu beschreibenden Merkmalen aggregiert. Ergänzt wurde dies um Erfahrungen bei der Einführung einer solchen Plattform in der Informationssystem-Architektur der Demonstrationsfabrik Aachen (s. STICH ET AL. 2018, S. 39). Die **beschreibenden Merkmale sind technischer Natur**. Die sozio-technische Komponente des Informationssystems „software-definierte Plattform“ wird in Kapitel 5.6 bei der Vorstellung der Integrationsanforderungen näher beleuchtet. Entsprechend einem Pfad von unten nach oben in Abbildung 5-5 können die Merkmale folgendermaßen sortiert werden: Zuerst werden **Schnittstellen** zu den Umsystemen benötigt, darauf aufbauend werden mithilfe der **Datenbankkonzepte** im **Datenmanagement** die Daten in der software-definierten Plattform vorgehalten. Diese werden dann über verschiedene **Stufen der Datenverarbeitung** zu Informationen gewandelt. Die **Informationsbereitstellung** bildet die Grenze zu den Entscheidungssystemen nach oben. Gleichzeitig werden alle verarbeiteten Daten über **IT-Sicherheitsfunktionen** geschützt. Außerhalb des Pfades dienen **Entwicklungsumgebungen** der agilen Konfiguration der Plattform.

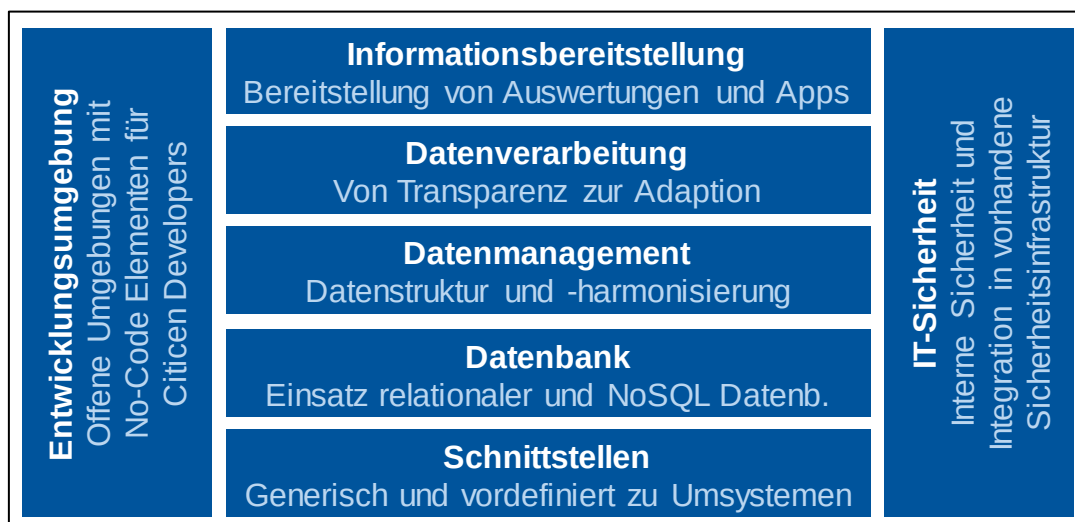


Abbildung 5-5: Übersicht und grobe Ordnung der Merkmale software-definierter Plattformen (eigene Darstellung)

Der dritte Schritt in der morphologischen Methode ist die Beschreibung von Ausprägungen je Merkmal. Solche Ausprägungen lassen sich mannigfaltig finden, was schon allein darin begründet ist, dass der Markt an verfügbaren software-definierten Plattfor-

men, beispielsweise von IoT-Plattformen, ständig größer wird und schon über 100 Anbieter umfasst (s. POSTSCAPES.COM 2017). Teilweise haben die im Folgenden genannten Ausprägungen mehrere darunterliegende Ebenen der Detaillierung.¹³ Entsprechend den Anforderungen, dass Ausprägungen auf möglichst wenige Elemente – im Optimalfall nur zwei – zu beschränken sind und dass sich die Merkmale auf derselben inhaltlichen Ebene befinden sollten, wurden die **Ausprägungen teilweise stark zusammengefasst**. Dies entspricht auch dem Ziel dieser Arbeit, einen *Überblick* über software-definierte Plattformen zu verschaffen – statt einer vertiefenden Beschreibung dieser Plattformen (s. Kapitel 4.1).

5.3 Merkmale zur Strukturierung software-definierter Plattformen

5.3.1 Schnittstellen

Tabelle 5-2: Ausprägungen des Merkmals *Schnittstellen* (eigene Darstellung)

<i>Schnittstellen</i>	generische Schnittstellen	vordefinierte Schnittstellen

Die Reihe der Merkmale beginnt mit den Schnittstellen. Der Begriff der Schnittstelle wurde in Kapitel 2.1.4 vorgestellt, genauso wie die gängigen Typen von Schnittstellen. Schnittstellen sind im vorliegenden Fall als bidirektionale Schnittstellen zu verstehen. Es muss also die Möglichkeit geben, nicht nur Daten aus Umsystemen herauszulesen, sondern von der software-definierten Plattform verarbeitete Daten auch wieder zurückzuschreiben (s. PERRY 2016, S. 11; SCHUH ET AL. 2017h, S. 120-121).

Software-definierte Plattformen unterstützen neutrale Schnittstellen, welche die API-Protokolle REST, SOAP oder Publish-Subscribe nutzen, sowie die entsprechenden Formate wie XML und JSON, aber auch Standards wie OPC-UA (s. Kapitel 2.1.4). Neben diesen direkten Schnittstellen bieten software-definierte Plattformen auch indirekte Schnittstellen, die beispielsweise Textdateien mit kommagetrennten Werten (CSV) einlesen können. Damit stellen sie die Kompatibilität zu Legacysystemen, die keine andere Datenausgabe ermöglichen, sicher. Die vorgenannten Schnittstellen werden in der Ausprägung **generische Schnittstellen** zusammengefasst.

Neben generischen Schnittstellen besitzen software-definierte Plattformen Bibliotheken **vordefinierter Schnittstellen**. Diese sind deutlich umfangreicher, als es generische Schnittstellen sein können. Der größte Unterschied liegt darin, dass Datenmodelle, Zeichenformate und Funktionsaufrufe spezifisch hinterlegt sind, die von der Plattform aus genutzt werden können. Die vordefinierten Schnittstellen betreffen sowohl die Verbindung zu Anwendungssystemen als auch zu OT-Systemen und intelligenten Objekten. **Schnittstellen zu Anwendungssystemen** sind beispielsweise APIs

¹³ Beim Merkmal *Schnittstellen* beispielsweise gibt es die Ausprägung *generische Schnittstellen*. Hierunter zählen Schnittstellen nach dem *API-Konzept*, wovon eine Ausprägung wiederum die *REST-Schnittstelle* ist, die über verschiedene Datenformate wie z. B. JSON kommuniziert.

zu Anwendungen der großen Softwarehersteller von betrieblicher Anwendungssoftware wie Microsoft, Oracle, IBM oder SAP. Sie dienen dazu, die Plattform nahtlos in die vorhandene Informationssystemlandschaft, aber auch in die Prozesswelt der Unternehmen, integrieren zu können (s. PELINO U. VOCE 2017, S. 2). Daher ist auch die bidirektionale Übergabe von Workflows über entsprechend definierte Funktionsaufrufe wichtiger Bestandteil solcher Schnittstellen. Zudem sollten die Schnittstellen vom jeweiligen Hersteller des Anwendungssystems zertifiziert sein, damit das Zusammenspiel der Systeme sicher und reibungslos funktioniert.

Um auf Quelldaten, beispielsweise von Sensoren aus der Produktion oder Maschinen im Einsatz beim Kunden, direkt zugreifen zu können, sind auch **vordefinierte Schnittstellen zu solchen intelligenten Objekten** nötig (s. DEINDL 2013b, S. 71-72). Bibliotheken zum Zugriff auf diese Objekte werden oft vom Hersteller des Objekts geliefert und können in die Plattform mit aufgenommen werden. So hat in einem ähnlichen Ansatz beispielsweise ein Hersteller von Software zum Schaltanlagenbau eine umfangreiche Bibliothek von elektrischen Teilen zur digitalen Montage der Schaltanlagen aufgebaut, die von den Herstellern selbst gepflegt wird.¹⁴ Entsprechend der Definition von software-definierten Plattformen in Kapitel 5.1.2 ist es zudem Aufgabe der Plattform, die Konfiguration und Verwaltung der Objekte zu übernehmen, da dies mit steigender Zahl verbundener Geräte manuell nicht mehr handhabbar ist. Wie es heute beispielsweise auch im Heimautomationsbereich üblich ist,¹⁵ gehört dazu das Pflegen der Konfiguration der Objekte, das Sicherstellen der Verbindung sowie die Überprüfung und Installation neuer Firmware-Stände (s. ENGELHARDT 2014, S. 45-49).

5.3.2 Datenmanagement

Tabelle 5-3: Ausprägungen des Merkmals *Datenmanagement* (eigene Darstellung)

<i>Datenmanagement</i>	feste Datenstruktur	Datenharmonisierung und -bereinigung	Datenanreicherung	frei konfigurierbare Datenobjekte

Nachdem Schnittstellen zu anderen Systemen definiert sind, ist die **Datenhaltung und damit das Management der erhaltenen Daten** ein wichtiges Merkmal software-definierter Plattformen. Die in den Kapiteln 1.1 und 4.1 beschriebenen Herausforderungen der IT-Komplexität aufgrund gewachsener IT-Strukturen erschweren diese Aufgabe jedoch, da die vorhandenen Datenbestände oft inkonsistent sind (s. LIEBER ET AL. 2013, S. 390-391). Dies betrifft sowohl Stammdaten, die oft in verschiedenen Anwendungssystemen gespeichert sind, als auch alle Arten von Bewegungsdaten, wie

¹⁴ Das *eplan data portal* findet sich unter <http://eplandata.de/portal/portal.php> und enthält alle relevanten Eigenschaften der Teile für den Schaltschrankbau wie Ausmaße, Anschlüsse und elektronische Eigenschaften.

¹⁵ Die freie Plattform *FHEM* beispielsweise unterstützt neben einer Vielzahl an Protokollen auch die Konfiguration und das Firmware-Update von intelligenten Sensoren und Aktoren in der Heimautomation (<https://fhem.de/>).

auch unstrukturierte Daten aus Informationssystemen und von intelligenten Objekten. Ohne systemische Unterstützung machen die notwendigen Schritte der Datenharmonisierung und -transformation oft bis zu 80 Prozent des Aufwands bei einer Systemintegration aus (s. GABRIEL ET AL. 2009, S. 129). Software-definierte Plattformen, die **solche Verfahren nicht unterstützen** und über ein festes, unflexibles Datenmodell verfügen, wird die Ausprägung „**feste Datenstrukturen**“ zugewiesen.

Der erste Schritt zum Umgang mit strukturierten Daten liegt daher in der **Datenbereinigung und -harmonisierung**. Die Datenbereinigung umfasst dabei die Behebung syntaktischer und inhaltlicher Datendefekte, die zum Beispiel aus Übertragungsfehlern oder Systemnichtverfügbarkeiten stammen (s. KEMPER U. FINGER 2016, S. 131). Es gehören ebenfalls das Entfernen von Doubletten und unvollständigen Datensätzen, beispielsweise der Stammdaten, dazu (s. DEUSE ET AL. 2014, S. 388). Danach kann eine Datenharmonisierung erfolgen. Diese beinhaltet die Angleichung der Datenformate, Zeichensätze und Einheiten. Die Datenharmonisierung kann teilweise automatisch erfolgen, benötigt jedoch eine vorgelagerte Erstellung spezifischer Regeln durch Mitarbeiter des Unternehmens (s. DREMEL U. HERTERICH 2016, S. 656; KEMPER U. FINGER 2016, S. 137-139). Um Speicherplatz zu sparen, sollte die Plattform außerdem die differenzielle Speicherung von Daten unterstützen und sie entsprechend komprimieren.

Die nächste Stufe des Datenmanagements lässt sich mit „**Datenanreicherung**“ zusammenfassen. Bei unstrukturierten Daten wie Bildern oder Videoaufzeichnungen kann dies **beispielsweise die Ergänzung um Zeitstempel** sein. Strukturierte Daten lassen sich nach bestimmten Dimensionen wie *Zeit* oder *Summe von Teilen* verdichten und besonders herausstechende Werte können hervorgehoben werden.

Auf der höchsten Stufe ermöglicht eine software-definierte Plattform, auf Basis der bisher genannten Ausprägungen des Datenmanagements, **frei definierbare Objekte**. In diesen Objekten werden dynamisch alle relevanten Daten zusammengeführt, die für einen bestimmten Zweck notwendig sind. Das Konzept entspricht dem Datalake-Konzept, das für Big-Data-Analysen entwickelt wurde; die Auswertungen können auch grob mit einer View in SQL (*Structured Query Language*) verglichen werden, die einer dynamischen Suchabfrage über fest definierte Tabellen entspricht. In einem solchen Konzept können **Daten in ihrer Rohform abgelegt** werden, da zum Zeitpunkt der Datenmodellierung die Anwendungszwecke noch unbekannt sind (s. THIELE ET AL. 2017, S. 58).¹⁶ Zur späteren Nutzung der Daten sind jedoch **Metadaten erforderlich**, um die Daten in den richtigen Kontext setzen zu können (s. dazu auch das Smart-Data-Konzept, Kapitel 2.1.1). Hiervon betroffen sind sowohl historische Daten, die in einer Datenbank abgelegt werden, als auch Datenströme, die erst zu einem späteren Zeitpunkt nach der Plattformintegration in Echtzeit ausgewertet werden sollen. Der große Vorteil

¹⁶ Dies wird auch mit dem Fachbegriff „*Schema-on-Read*“ bezeichnet, bei dem das Datenmodell (*Schema*) erst bei der Datenauswertung (*Read*), statt vor der Datenspeicherung (dort: „*Schema-on-Write*“) festgelegt wird (s. THIELE ET AL. 2017, S. 58).

eines solchen Konzepts ist, dass die Objekte untereinander kombinierbar sind, um flexibel auf neue Anforderungen reagieren zu können.

5.3.3 Datenbankkonzept

Tabelle 5-4: Ausprägungen des Merkmals **Datenbankkonzept** (eigene Darstellung)

Datenbank-konzept	relationale Datenbank	NoSQL-Datenbank	sonstige Datenbankkonzepte

Dem Nutzer verborgen, jedoch für die optimale Verarbeitung der Daten unumgänglich, sind die **Datenbankkonzepte, auf denen eine software-definierte Plattform aufbaut**. Diese Konzepte sind teilweise vom Hersteller der Plattform vorgegeben, teilweise jedoch auch frei wählbar. Je nachdem, wie welche Typen von Daten abgelegt sind, kann deren Auswertung und Visualisierung mit unterschiedlich hohem technischen Ressourcenaufwand erfolgen. Beim Aufbau von Datenmodellen in einer software-definierten Plattform muss daher auch auf die Datentypen und die zu erwartenden Auswertungen geachtet und die dafür passenden Datenbankkonzepte ausgewählt werden (s. PERRY 2016, S. 5). Hierfür muss zwischen strukturierten und unstrukturierten Daten, sowie anhand der Datenmenge, die in einer Operation ausgewertet werden soll, unterschieden werden.

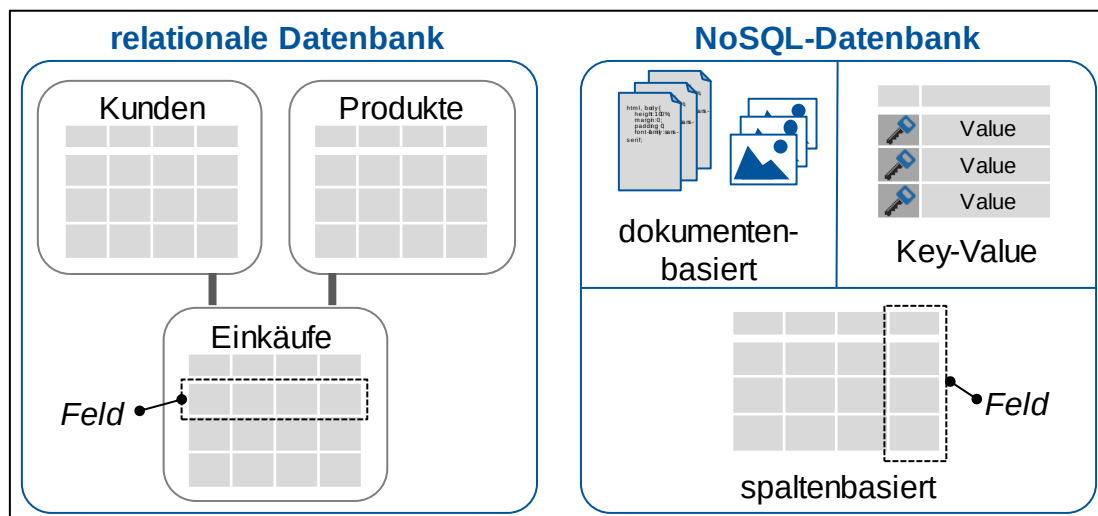


Abbildung 5-6: Vergleich relationaler und NoSQL-Datenbanken (eigene Darstellung auf Basis von WODEHOUSE 2016)

Das klassische und meistgenutzte Datenbankkonzept der letzten Jahre ist das der **relationalen Datenbanken**. Relationale Datenbanken sind für **strukturierte Daten in einem fest definierten Schema** von Codd entwickelt worden, um die Nutzung der Daten von deren Repräsentanz im Datenbanksystem zu entkoppeln (s. Codd 1970,

S. 377). Sie sind prinzipiell auch in der Lage, unstrukturierte Daten aufzunehmen.¹⁷ Das Datenbankmodell basiert auf der mathematischen Mengentheorie und besteht, vereinfacht gesagt, aus Tabellen mit definierten Spalten und beliebiger Anzahl an Zeilen. Über Relationen (*Verweise*) lassen sich einzelne Entitäten (*Zeilen*) miteinander verbinden. Zur Definition einer relationalen Datenbank werden daher auch Entity-Relationship-Modelle genutzt, teilweise auch Unified-Modeling-Language-Klassendiagramme (UML-Diagramme) (s. LEIMEISTER 2015, S. 88-89; LEMKE ET AL. 2017, S. 348-349).

Die Zunahme an unstrukturierten Daten, die Notwendigkeit die Schnelligkeit der Auswertung über die absolute Datenkonsistenz zu setzen sowie die Notwendigkeit der verteilten Datenablage hat in der jüngeren Vergangenheit zur Entwicklung neuer, verteilter Datenbankkonzepte geführt (s. SATTLER 2016, S. 183; SCHICKER 2017, S. 307-308). Diese Datenbankkonzepte fassen die meisten Autoren unter dem Begriff *NoSQL*, eine Abkürzung für *Not only SQL*, zusammen (s. KUMAR ET AL. 2017, S. 71-73). **NoSQL-Datenbanken erfassen unstrukturierte und strukturierte Daten** auf Basis eines flexiblen Datenmodells, sie nutzen verteilte Ablagen und stellen Konsistenz teilweise erst mit Zeitversatz her.¹⁸ Es haben sich aktuell drei für software-definierte Plattformen relevante NoSQL-Datenbankkonzepte etabliert:¹⁹ *Key-Value-Datenbanken*, die Datensätze zusammen mit einem Schlüssel abspeichern und sehr einfach skalieren, *dokumentenbasierte Datenbanken*, die sich zur Ablage aller Arten von Dokumenten vor allem im Web-Umfeld eignen, sowie *spaltenorientierte Datenbanken*, welche Daten statt in Zeilen (s. relationale Datenbanken) in Spalten abspeichern, um beispielsweise Summenbildungen zu erleichtern (s. KUMAR ET AL. 2017, S. 71; SCHICKER 2017, S. 315). NoSQL-Datenbanken werden oft auch mithilfe der **In-Memory-Technik** (bzw. teilweise auch Main-Memory-Datenbanken genannt) genutzt. Sie basiert darauf, dass die Kosten für Hauptspeicher in den letzten Jahren stark gesunken sind und es daher wirtschaftlich sinnvoll ist, Datenbanken komplett im Hauptspeicher zu betreiben. Dies hat den großen Vorteil der Echtzeitfähigkeit von Auswertungen. Für In-Memory-Datenbanken sind einige spezifische Designanforderungen zu beachten, vor allem der Umgang mit der Flüchtigkeit des Hauptspeichers. Da Datenbanken auch bei Stromausfall oder Systemneustart konsistent bleiben müssen, sind entsprechende Sicherheitsmechanismen zur Ablage einer Kopie auf der Festplatte zu implementieren.

¹⁷ Unstrukturierte Daten können von einer relationalen Datenbank gespeichert werden, indem entweder die entsprechenden Daten, wie z. B. ein Bild, über den Datentyp BLOB in einer Zelle gespeichert werden oder indem ein Verweis auf eine Datei in einem Dateiverzeichnis hinterlegt wird.

¹⁸ Dies führt beispielsweise dazu, dass auf der Plattform eBay neu eingestellte Artikel erst nach einiger Zeit auf allen eBay-Seiten angezeigt werden und nicht sofort verfügbar sind. Würde eBay eine relationale Datenbank nutzen, würde durch das Zwei-Phasen-Commit sofort nach der Bestätigung des erfolgreichen Starts einer Auktion diese auch überall zu sehen sein. Dies hätte jedoch negative Auswirkungen auf die Webseitenperformance.

¹⁹ Als ein weiteres Konzept gilt das der Graphen-Datenbanken. Dieses ist jedoch primär für Geodatenbanken oder Navigationsgeräte im Einsatz (s. SCHICKER 2017, S. 316). Zudem findet es zunehmend Verbreitung in sozialen Netzwerken (s. auch Fußnote zu Kapitel 2.1.4).

Weitere Ansätze beziehen sich vor allem auf die sich unterscheidende Art der Hardwareauslegung im Hauptspeicher, die neue Ansätze der Datenablage, Indizierung und Abfragemethoden erfordert, die bei einer Implementierung zu beachten sind (s. SHERMAN 2014, S. 162; LARSON U. LEVANDOSKI 2016, S. 1609-1610). In der Praxis finden sich oft Mischformen der vorgenannten Ansätze, um deren Vorteile miteinander zu kombinieren.

Potentiell könnten software-definierte Plattformen auch **sonstige Datenbankkonzepte**, wie beispielsweise objektorientierte Datenbanken, unterstützen. Diese Ausprägung kommt jedoch in der Praxis selten vor, da sich diese Datenbankkonzepte nur in sehr spezifischen Anwendungen durchgesetzt haben (s. SCHICKER 2017, S. 319).

5.3.4 Stufen der Datenverarbeitung

Tabelle 5-5: Ausprägungen des Merkmals *Stufen der Datenverarbeitung* (eigene Darstellung)

<i>Stufen der Datenverarbeitung</i>	Erzeugen von Sichtbarkeit	Transparenz und Diagnose	Prognose	Adaption und Entscheidungsautomatisierung

Hinsichtlich der Nutzung von in der Plattform gespeicherten Daten nimmt die Datenverarbeitung eine wichtige Rolle ein. Ausgehend von Diskussionen um Daten in BI-Systemen haben zuerst KART ET AL. von GARTNER in ihrem **vierstufigen „Data Analytics Maturity Model“** die **verschiedenen Stufen der Datenverarbeitung** aufgezeigt (s. KART ET AL. 2013, S. 2-4; Abbildung 5-7), die dann später von verschiedenen Autoren übernommen wurden (s. u. a. in SHERMAN 2014, S. 16) und sich als wissenschaftlicher Standard etabliert haben (s. WENKING ET AL. 2017, S. 34). Sie wurden auch auf die Anwendung des digitalen Schattens in der Produktion (s. SCHUH ET AL. 2016a) sowie auf die Industrie-4.0-Reife eines Unternehmens in Summe (s. SCHUH ET AL. 2017a) übertragen. Diese Stufen dienen als Ausprägungen des Merkmals „Stufen der Datenverarbeitung“. Software-definierte Plattformen bieten oft Bibliotheken verschiedener Verfahren zur Datenauswertung, die bei der Entwicklung eingebunden werden können (s. z. B. GE DIGITAL 2016, S. 24; PTC 2017).

Zwar bieten software-definierte Plattformen die Möglichkeit, Datenauswertungen entsprechend den nachfolgend beschriebenen Stufen umzusetzen; aktuell wird diese Möglichkeit allerdings nur unzureichend genutzt. Eine im Jahr 2017 mit produzierenden Unternehmen durchgeführte Studie zeigt, dass nur sechs Prozent der Unternehmen ihre Produktionsdaten nutzen, um beispielsweise Verbesserungen der Produktqualität oder der Produktionsprozesse zu erzielen (s. WENKING ET AL. 2017, S. 33-37). Um die im Folgenden beschriebenen Möglichkeiten zu nutzen, müssen Unternehmen ihre Bestrebungen zur Datennutzung deutlich verstärken und außerdem die richtigen Voraussetzungen schaffen (s. dazu Kapitel 5.3.2).

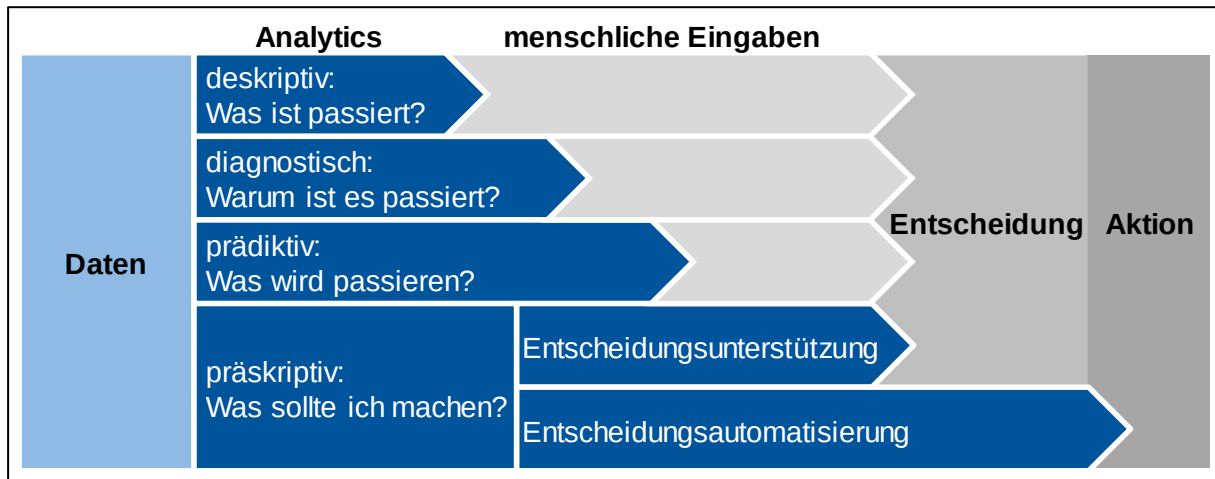


Abbildung 5-7: Vierstufiges Data Analytics Model (i. A. a. KART ET AL. 2013, S. 2)

Die **erste Stufe** beschreibt die der **Sichtbarkeit**. Diese Stufe entspricht den in Kapitel 5.3.2 beschriebenen Ausprägungen, die dazu dienen, historische Daten aus verschiedenen Systemen bereitzustellen und damit digitale Schatten zu erzeugen. Ist dies geschehen, können die Daten von Anwendern dazu genutzt werden, Prozessabläufe nachzuvollziehen. Diese Stufe dient als Grundlage, um beispielsweise ungewollte Ereignisse überhaupt zu erkennen.

Die **zweite Stufe**, in der Zusammenhänge erfasst und diese richtig interpretiert werden, wird mit **Transparenz** oder **Diagnose** beschrieben. Hierzu müssen die erhobenen Daten kontextsensitiv analysiert werden. Die software-definierte Plattform setzt dazu auf Datenanalyseverfahren aus dem Bereich der Big-Data-Analytics. Diese umfassen einerseits klassische Auswertungen mit bekannten Hypothesen. Sie benötigen einen Datenanalysten, der die Hypothesen festlegt, die dann auf einem abgeschlossenen Datensatz geprüft werden. Andererseits geht es bei explorativen Verfahren wie Machine-Learning oder Künstlicher Intelligenz darum, neue Zusammenhänge zu entdecken. Solche Verfahren können durch systematisches Anlernen der Algorithmen durch den Menschen zusätzlich verfeinert werden (s. SCHUH ET AL. 2017a, S. 12; THIELE ET AL. 2017, S. 59).

Die **dritte Stufe** unterstützt auf Basis der Auswertung von historischen Daten die **Prognose** zukünftiger Daten. Auch hierzu werden mathematische Verfahren wie Regressionsanalysen eingesetzt, um künftige Ereignisse wie beispielsweise Lieferprobleme oder eine Überlastung der Fertigung vorherzusagen. Der digitale Schatten wird zu diesem Zweck in die Zukunft projiziert und unterschiedliche Szenarien gebildet, die anhand ihrer Eintrittswahrscheinlichkeit bewertet werden.

Zuletzt bildet die Stufe der **Adaption und Entscheidungsautomatisierung** die **höchste Optimierungsmöglichkeit**, die auf Basis von Daten umgesetzt werden kann. Mithilfe des digitalen Schattens können Entscheidungen mit den größten positiven Auswirkungen autonom und ohne menschliches Zutun in kürzester Zeit getroffen und die daraus resultierenden Maßnahmen umgesetzt werden (s. SCHUH ET AL. 2017a, S. 18). Der Grad der Autonomie hängt von der Komplexität von Entscheidungen und

dem Kosten-Nutzen-Verhältnis ab. Oftmals ist es sinnvoll, nur einzelne Prozesse autonom zu gestalten. Im vorherigen Beispiel „Überlastung der Fertigung“ würde in dieser Stufe automatisch eine Gegenmaßnahme durchgeführt werden, wie zum Beispiel eine Umplanung von Fertigungsaufträgen.

5.3.5 IT-Sicherheitsfunktionen

Tabelle 5-6: Ausprägungen des Merkmals *IT-Sicherheitsfunktionen* (eigene Darstellung)

<i>IT-Sicherheitsfunktionen</i>	nur interne Sicherheitsfunktionen	Sicherheit für Plattform und angebundene Systeme	Vollintegration in IT-Sicherheitsinfrastruktur des Unternehmens

IT-Sicherheitsfunktionen bilden ein ebenfalls elementares Merkmal software-definierter Plattformen. Eine Plattform, von der aus **alle unternehmensrelevanten Daten erreichbar sind, ist ein hochpotentes Angriffsziel**. Gleichzeitig arbeiten viele Menschen mit den Daten einer solchen Plattform, die selbst unbeabsichtigt potentielle Sicherheitsrisiken darstellen – sowohl bei der Integration, Administration als auch bei der Nutzung der Plattform (s. HOLZ ET AL. 2016, S. 5-6). Es müssen also Sicherheitsfunktionen sowohl zum Schutz gegen unberechtigte externe Zugriffe verfügbar sein, als auch inhärente Sicherheitsfunktionen, mit denen sich die Plattform selbst schützt.

Zu den **internen Sicherheitsfunktionen** gehört zuerst die Möglichkeit zur verschlüsselten Datenablage von sensiblen Daten. Die Plattform verarbeitet solche Daten zudem unter Nutzung von vorhandenen Hardware-Features wie beispielsweise *Trusted Platform Modules* (TPM) oder spezifische Sicherheitschips, um Angriffe über die Hardware zu erschweren. Weiterhin verfügt sie über umfangreiche Überwachungs- und Protokollierungsfunktionalitäten, um im Falle eines unberechtigten Zugriffs den Zugriffsweg blockieren oder die Ursache beheben zu können (s. KUBER ET AL. 2016, S. 40-41). Zu den internen Sicherheitsfunktionen gehört zudem die Möglichkeit, umfangreiche und hierarchische Lese- und Schreibberechtigungen vergeben zu können. So kann sichergestellt werden, dass notwendige Zugänge von außen nur eine beschränkte Eingriffsmöglichkeit eröffnen. Auch die Geschäftsführung benötigt keinen Vollzugriff auf das System, das Lesen von Auswertungen kann hier oft ausreichen (s. BLECK 2016, S. 4). Um bei der Entwicklung eigener Funktionen oder Apps ebenfalls ein hohes Sicherheitsniveau zu erreichen, sind zudem entsprechende Frameworks für sicheres Entwickeln Teil der Entwicklungsumgebung der Plattform (s. Kapitel 5.3.7). Diese sehen die Nutzung von Vorgehensmodellen zur sicheren Entwicklung wie Microsofts *SDL* (Secure Development Lifecycle) oder den *CLASP* (Comprehensive Lightweight Application Security Process) der Organisation *OWASP* vor (s. ALPAR ET AL. 2016, S. 348-350).

Neben der internen Sicherheit können software-definierte Plattformen darüber hinaus auch über **Sicherheitsmechanismen für angebundene Systeme** verfügen. Auch

hier ist die Verschlüsselung ein zentrales Element. Die Schnittstellen der Plattform verschlüsseln den ein- und ausgehenden Datenverkehr mit gängigen Verfahren wie dem *Advanced Encryption Standard* (AES) oder dem *Rivest, Shamir und Adleman-Kryptosystem* (RSA). Dazu wird beispielsweise AES 256 für symmetrische Schlüssel oder RSA 2048 für asymmetrische, öffentliche Schlüssel eingesetzt. Solche Verfahren setzen voraus, dass die Gegenseite ebenfalls die Verschlüsselung unterstützt und über ausreichend Prozessorleistung dazu verfügt. Sie sind aber unumgänglich, wenn Datenmanipulation vermieden werden soll. Um sicherzustellen, dass eine Verbindung tatsächlich von dem System stammt, von dem sie erwartet wird, empfiehlt es sich außerdem, feste Kommunikationspunkte zu definieren und auch Systemen selbst über Authentifizierungsverfahren via User und Passwort beziehungsweise über Zertifikate Zugriff zu gestatten (s. BLECK 2016, S. 3).

In der höchsten Ausprägung lässt eine software-definierte Plattform die **Vollintegration in die Sicherheitsinfrastruktur des Unternehmens** zu. Dazu gehört beispielsweise die Einbindung des unternehmenseigenen Verzeichnisdienstes (auch *Lightweight Directory Access Protocol* (LDAP)) oder der Zertifizierungsstellen des Unternehmens. Genauso ist eine Integration in die Namensdienste und in die Firewallkonfiguration unerlässlich (s. BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK 2013). Besonders mit Blick auf die Situation in Deutschland ist relevant, dass die Plattform auf dieser Stufe auch Bestrebungen zur Wahrung des Datenschutzes unterstützt. Dies geschieht, indem beispielsweise die Möglichkeit bereitgestellt wird, die Verarbeitung von Anwenderdaten jederzeit nachvollziehen und bestimmte Verarbeitungen unterbinden zu können, wie es die europäische Datenschutzverordnung (DSGVO) vorsieht (s. KUBER ET AL. 2016, S. 35-36).

5.3.6 Informationsbereitstellung

Tabelle 5-7: Ausprägungen des Merkmals *Informationsbereitstellung* (eigene Darstellung)

<i>Informationsbereitstellung</i>	keine eigene Auswertungsfunktion	Bereitstellung von Auswertungen und Visualisierungen	Generierung eigener Apps/Widgets

Als Intermediär ist es Aufgabe software-definierter Plattformen, erhobene Daten und generierte Informationen Anwendern und Folgeprozessen bereitzustellen. Dazu stehen ihnen unterschiedliche Technologien zur Verfügung, die in den folgenden Ausprägungen zusammengefasst werden. Sollte die Plattform über keine Funktionen zur Auswertung von Daten verfügen, so entspricht sie der Ausprägung „**keine eigene Auswertungsfunktionalität**“. In diesem Fall werden Daten über Schnittstellen an angebundene Informationssysteme, wie beispielsweise ein BI-System, übergeben, welche diese Aufgaben übernehmen.

Ziel der Einführung einer software-definierten Plattform ist es jedoch, systemübergreifende **Auswertungen und Visualisierungen** von sowohl historischen als auch aktuellen Daten bereitzustellen. In der Produktion beispielsweise gilt dies auf verschiedenen Ebenen: von der Visualisierung von Verschleißdaten einer einzelnen Maschine bis hin zum Vergleich von Kennzahlen verschiedener Fertigungslinien, die beim Line-Balancing unterstützen (s. JINUSHI ET AL. 2016, S. 77-78). Zur Umsetzung der Visualisierung können Cockpits oder Dashboards verwendet werden. Viele Informationssysteme liefern Bausteine für solche Cockpits, die dann vom Anwender selbst konfiguriert werden können. Sie dienen zur Darstellung von Key-Performance-Indikatoren (KPI), zum Beispiel im Rahmen einer *Balanced Scorecard*. Weiterhin sind Reports, die in regelmäßigen Abständen gewisse Auswertungen zusammenfassen und bereitstellen, Teil dieser Ausprägung (s. BANGE 2016, S. 112-118).

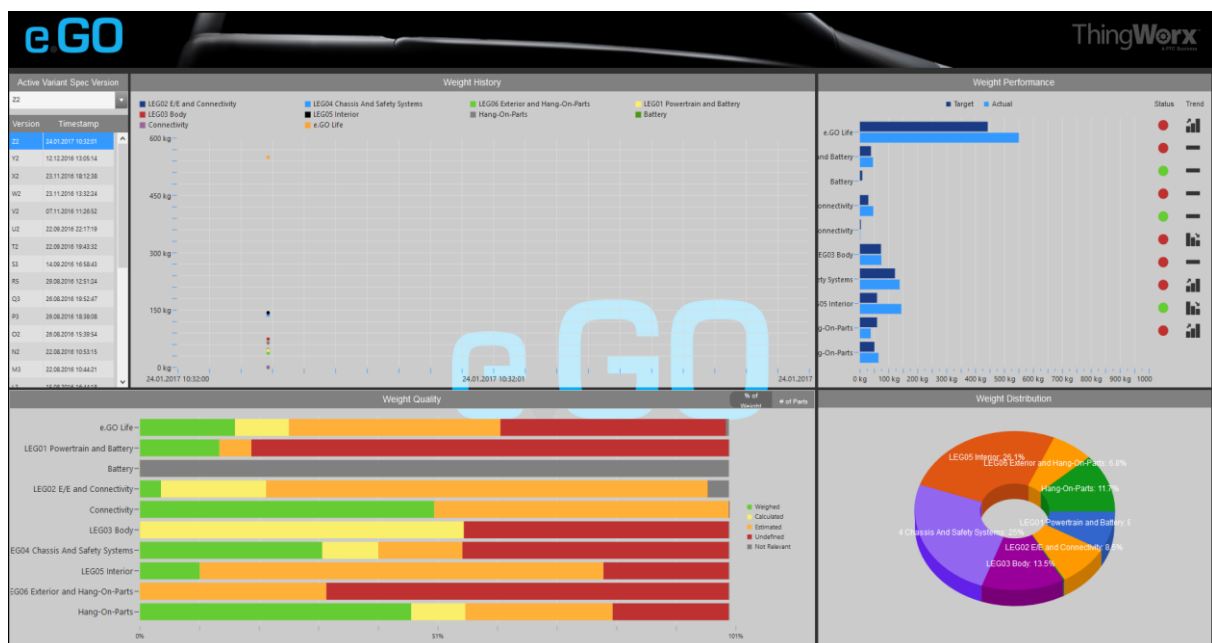


Abbildung 5-8: Beispiel einer auf einer software-definierten Plattform erzeugten App (Screenshot von e.GO)

In der höchsten Ausprägungsstufe erlaubt die software-definierte Plattform die **Erzeugung von eigenen Apps oder Widgets** (s. Beispiel in Abbildung 5-8). Viele Plattformen unterstützen diese Funktion, die über ein reines Anzeigen von Daten hinausgeht (s. MINERAUD ET AL. 2015, S. 1). Solche Apps erlauben, wie auch vom Smartphone bekannt, Interaktionen mit den darunterliegenden Systemen. Entscheidungen von Nutzern, die sie auf Basis der bereitgestellten Information in der App getroffen haben, müssen von der Plattform verarbeitet und an die relevanten angebotenen Systeme verteilt werden. Zu den Funktionen der Bereitstellung von Apps zählt auch die Unterstützung verschiedener Anzeigegeräte. Dies betrifft klassische PCs und Laptops sowie mobile Anzeigegeräte, aber auch die Möglichkeit zur Anbindung von *Augmented-Reality-Technologien*, wie sie beispielsweise in der Systemarchitektur von ALAM ET AL. vorgeschlagen wird (s. ALAM ET AL. 2017, S. 111). Wie die Spieleindustrie anhand des inzwischen wieder abgeklungenen Hypes um das Spiel *Pokémon Go* (s. GÖTZ ET

AL. 2017, S. 251), aber auch die Vielzahl an inzwischen verfügbaren industriellen Anwendungsfällen zeigen, hilft *Augmented Reality*, um für Menschen einfach und verständlich Informationen im Anwendungskontext bereitzustellen.²⁰ Solche Anwendungsfälle finden sich sowohl in der Entwicklungsabteilung, der Produktion und Logistik als auch im Vertrieb und ermöglichen sowohl Darstellung von als auch Interaktion mit Informationen (s. PORTER U. HEPPELMANN 2017, S. 46-51). Dabei sollten nicht nur eigene Technologien des Anbieters der Plattform unterstützt werden, sondern entsprechend offene Frameworks integriert sein.

5.3.7 Entwicklungsumgebung

Tabelle 5-8: Ausprägungen des Merkmals *Entwicklungsumgebung* (eigene Darstellung)

<i>Entwicklungsumgebung (E.)</i>	keine eigene E.	eigene, proprietäre E.	offene E.

In produzierenden Unternehmen sah ein Softwareimplementierungsprozess in der Vergangenheit oft folgendermaßen aus: Das aktuelle ERP-System war in die Jahre gekommen, es war oft schon bis zu zehn Jahre im Einsatz (s. ABTS U. MÜLDER 2017, S. 515). Daher erfolgte eine Ausschreibung und Neuauswahl eines Systems. Dieses wurde dann in einem oft Jahre dauernden Prozess auf die zu diesem Zeitpunkt relevanten, spezifischen Unternehmensprozesse angepasst und die Altdaten migriert. Nach der Einführung des Systems wurden zumeist nur noch kleinere Anpassungen vorgenommen, vor allem um Fehler zu beseitigen. Danach wurde es meist über viele Jahre nur marginal angepasst, oft nur aufgrund regulatorischer Zwänge. Für diese Anpassung gab es wenige Spezialisten, welche die Programmierung übernahmen.

Hier findet derzeit ein Paradigmenwechsel statt, welcher von der Entwicklungsumgebung einer software-definierten Plattform unterstützt werden muss. Die Einbindung neuer Datenquellen, Auswertung der Daten und Entwicklung von Apps kann in agilen Unternehmen nicht durch wenige Experten allein durchgeführt werden. In der englischsprachigen Literatur werden Personen, die solche Entwicklungen künftig umsetzen sollen, als „*Citizen Developers*“, also bürgerliche Entwickler, bezeichnet (s. GARTNER 2017a). Dies sind Anwender, die **ohne Programmierkenntnisse eigene Anwendungen erstellen** können, die den Qualitätsanforderungen entsprechen, welche an Anwendungssysteme in Unternehmen gestellt werden. Solche Entwickler benötigen spezifische Entwicklungsumgebungen, die als „*Low Code Platform*“ bezeichnet werden (s. Abbildung 5-9). Diese Umgebungen benötigen einen minimalen Aufwand für Installation und Training der Mitarbeiter (s. KOPPER ET AL. 2017, S. 105; RICHARDSON U. RYMER 2016, S. 2). Sie finden ihren Ursprung in der Webentwicklung, wo schon früh für die Pflege von Webseiten über in Content-Management-Systeme

²⁰ Die Anwendungsfälle für produzierende Unternehmen reichen von Kommissionierlösungen wie „*Pick-by-Vision*“ über Montageanleitungen zum Anlernen von Montagearbeitern bis hin zu Lösungen im Service, die dem Servicetechniker oder Kunden Instandhaltungsanweisungen geben.

eingebaute WYSIWYG-Editoren²¹ gearbeitet wurde und so die Contentpflege von Webentwicklern an einfache Anwender übergeben werden konnte.

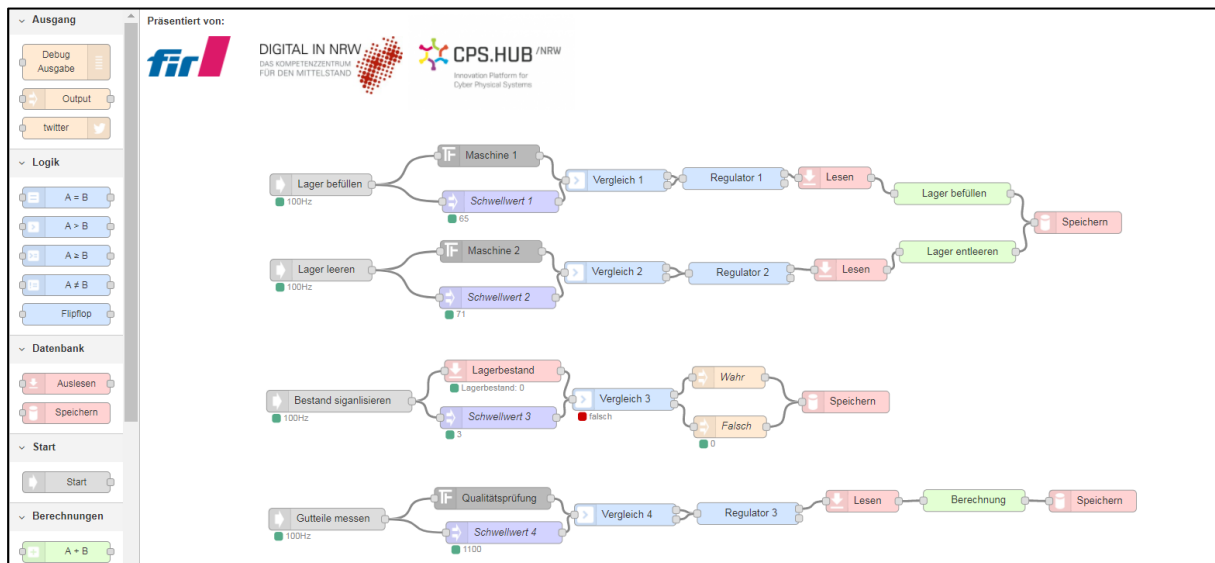


Abbildung 5-9: Screenshot einer Low-Code-Plattform aus einem Projekt des FIR (Screenshot des FIR)

Entwicklungsplattformen von software-definierten Plattformen können die folgenden Ausprägungen besitzen: Im einfachsten Fall besitzen sie **keine eigene Entwicklungsplattform**, sondern setzen auf anderen allgemeinen Plattformen wie zum Beispiel *Eclipse* auf. Wichtig in diesem Fall ist, dass durch den Softwarehersteller entsprechende Bibliotheken bereitgestellt werden, damit die Plattformfunktionen optimal bei der Entwicklung genutzt und integriert werden können.

Viele Anbieter stellen jedoch eigene, **proprietäre Entwicklungsumgebungen** bereit. Solche Umgebungen bieten teilweise einen verringerten Komfort für Entwickler, weil ihnen beispielsweise Schnittstellen zu Versionsverwaltungssystemen wie GitHub fehlen (s. SHAHIN ET AL. 2017, S. 3922-3923). Sie sollten jedoch mindestens wichtige *Software Development-Kits* (SDKs) unterstützen. Weiterhin sollen sie, im Sinne des zuvor dargestellten Low-Code-Plattform-Konzepts, beispielsweise visuelle Möglichkeiten zur Softwareentwicklung (auch *Visual Scripting* genannt) bereitstellen.

Als letzte Ausprägung verbindet der hybride Ansatz einer **offenen Entwicklungsumgebung** die Vorteile beider Welten. Eine solche Umgebung bietet die Möglichkeit, verschiedene Tools wie beispielsweise zur Codeversionierung, der Codeanalyse oder zur Testautomatisierung als Add-on zu nutzen, kommt aber auch ohne diese aus und bietet entsprechende Basisfunktionalitäten. Zusätzlich bieten sie die Möglichkeit, über Marktplätze Funktionsbausteine anderer Anbieter zu erwerben und in die eigene Lösung zu integrieren.

²¹ Das „What You See Is What You Get“-Prinzip wird beispielsweise auch von Textanwendungen wie Microsoft Word umgesetzt: Die in der Bearbeitungssicht erstellten Dokumente sehen auf dem Ausdruck beziehungsweise als PDF-Dokument genau gleich aus.

5.4 Zusammenfassung der beschreibenden Merkmale

Die Beschreibungen in den obigen Kapiteln bilden eine Übersicht über die Eigenschaften software-definierter Plattformen und helfen, diese systematisch zu beschreiben. Sie werden im Folgenden dazu genutzt, Plattformen zu typisieren. Eine zusammenfassende Darstellung gibt, entsprechend der Forderung von WELTER (s. WELTER 2006, S. 115-116), Tabelle 5-9. Die Ausprägungen der Merkmale sind, mit Ausnahme der Merkmale *Schnittstellen* und *Datenbankkonzepte*, aufsteigend ordinalskaliert.

Tabelle 5-9: Merkmale software-definierter Plattformen (eigene Darstellung)

Merkmals	Ausprägungen			
<i>Schnittstellen</i>	generische Schnittstellen	vordefinierte Schnittstellen		
<i>Datenmanagement</i>	feste Datenstruktur	Datenharmonisierung und -bereinigung	Datenanreicherung	frei konfigurierbare Datenobjekte
<i>Datenbankkonzept</i>	relationale Datenbank	NoSQL-Datenbank	sonstige Datenbankkonzepte	
<i>Stufen der Datenverarbeitung</i>	Erzeugen von Sichtbarkeit	Transparenz und Diagnose	Prognose	Adaption und Entscheidungsautomatisierung
<i>IT-Sicherheitsfunktionen</i>	nur interne Sicherheitsfunktionen	Sicherheit für Plattform und angebundene Systeme	Vollintegration in IT-Sicherheitsinfrastruktur des Unternehmens	
<i>Informationsbereitstellung</i>	keine eigene Auswertungsfunktion	Bereitstellung von Auswertungen und Visualisierungen	Generierung eigener Apps/Widgets	
<i>Entwicklungsumgebung (E.)</i>	keine eigene E.	eigene, proprietäre E.	offene E.	

Eine software-definierte Plattform, deren Ausprägungen sich weiter rechts einordnen lassen, hat zwar einen höheren Funktionsumfang, trotzdem ist sie nicht zwingend besser für das Unternehmen, das sie einsetzt. Plattformen mit zu großem Funktionsumfang sind unter Umständen schwerer zu bedienen und komplizierter in der Anwendung. Die Merkmale selbst sind voneinander unabhängig, obschon Softwarehersteller

die Module ihrer Plattformen trotzdem mit Abhängigkeiten vertreiben, was jedoch Vertriebsgründen geschuldet ist.

Sollten die beschriebenen Merkmale zur Auswahl einer bestimmten Plattform für ein Unternehmen genutzt werden, so sind in einem **zweiten Schritt weitere Kriterien zu beachten**, die nicht direkt messbar sind und aus den Softwarespezifikationen erhoben werden können. Für den erfolgreichen Einsatz der Software und eine hohe Akzeptanz beim Anwender sind sie dennoch elementar: Da der Datenbestand und das Einsatzgebiet einer software-definierten Plattform stark anwachsen können, ist die **Skalierbarkeit** ein wichtiges Kriterium. Dies hängt eng zusammen mit dem Kriterium der **Agilität**, also der Möglichkeit schnell und einfach Änderungen in Prozessen, Datenmodellen oder Visualisierungen vornehmen zu können (s. PERRY 2016, S. 13). Dabei unterstützt eine gute **Usability** der Oberflächen der Plattform den Anwender. Usability-Tests können am einfachsten im Rahmen eines Softwareauswahl-Prozesses, beispielsweise bei einer gut vorbereiteten Systempräsentation, durchgeführt werden. Zuletzt zählt auch die **Zuverlässigkeit** der Plattform zu den nicht im Voraus messbaren Kriterien (s. VASISHT ET AL. 2017, S. 516). Da eine software-definierte Plattform zu den Kernanwendungen eines Unternehmens zählen kann, sollten vor dem Go-live beispielsweise umfangreiche Lasttests durchgeführt werden. Zusätzlich sind technische Voraussetzungen, wie die Bereitstellung von ausfallsicheren Betriebsumgebungen, zu treffen, um einen zuverlässigen Betrieb zu ermöglichen.

5.5 Anwendung der Morphologie zur Bildung von Typen software-definierter Plattformen

Um Typen software-definierter Plattformen herzuleiten, muss die **Landschaft der Softwarehersteller genauer betrachtet** werden. Für die vorliegende Arbeit wurden deshalb verfügbare Plattformen untersucht, die den vorgenannten Eigenschaften mindestens teilweise entsprechen (siehe Anhang A.5 sowie in HOFFMANN ET AL. 2018). Die software-definierten Plattformen aus dieser Untersuchung, die für produzierende Unternehmen eine hohe Relevanz aufgrund deren Marktdurchdringung, Einsatzgebiet und Nennung in Studien haben, zeigt Tabelle 5-10. Es hat sich dabei gezeigt, dass sich ein Teil der als „IoT-Plattform“ vertriebenen Systeme für diesen Zweck nutzen lassen (s. Kapitel 3.1). Dies ist unter anderem durch die hohe Überlappung der Merkmale software-definierter Plattformen (s. Kapitel 5.3) mit denen von IoT-Plattformen begründet, wie sie beispielsweise von KRAUSE ET AL. aufgeführt werden (s. KRAUSE ET AL. 2017, S. 12-21). Zudem hat dies der Einsatz von IoT-Plattformen für konkrete Anwendungsfälle in der Aachener Demonstrationsfabrik durch den Autor gezeigt (s. STICH ET AL. 2018, S. 39-40). Es gibt jedoch auch andere, den IoT-Plattformen ähnliche Systeme, wie beispielsweise die von MPDV vorgestellte *Manufacturing Integration Platform*, die ERP-, PLM-, und weitere Informationssysteme in der Fertigung integriert (s. MPDV MIKROLAB 2017).

Tabelle 5-10: Software-definierte Plattformen zum Einsatz bei produzierenden Unternehmen (eigene Darstellung basierend auf PELINO U. HEWITT 2016; CROOK U. MACGILLIVRAY 2017)

Logo	Plattform	Software-hersteller	Plattformtyp
	Agile IoT Platform	Ayla Networks	Typ I
	Amazon Web Services (AWS) IoT Platform	Amazon	Typ I (unter Nutzung anderer AWS)
	Azure IoT Suite	Microsoft (MS)	Typ I (über Integration in MS-Systemwelt)
	Bosch IoT Suite	Bosch Software Innovations	Typ II
SAP Cloud Platform	Cloud Platform	SAP	Typ I (über Integration in SAP-Systemwelt)
	Fujitsu Cloud Service K5 IoT Platform	Fujitsu	Typ II
	Jasper Control Center	Cisco	Typ II
	Murano IoT Platform	Exosite	Typ II
	Oracle IoT Cloud Service	Oracle	Typ II
	Predix	GE	Typ I
	ThingWorx	PTC	Typ I
	Watson IoT Platform	IBM	Typ II
	Xively	LogMeIn	Typ II

Der Markt an solchen Plattformen ist aktuell in einem starken Wandel und Wachstum begriffen. Es sind sowohl Konsolidierungsbestrebungen zu sehen als auch das gleichzeitige Entstehen immer neuer Systeme (s. PELINO U. HEWITT 2016, S. 6; KRAUSE ET AL. 2017, S. 8). Dies liegt darin begründet, dass für unterschiedliche Anwendungsfälle spezifische Plattformen entstehen und es wenige generell einsetzbare Systeme gibt (s. CROOK U. MACGILLIVRAY 2017, S. 2). Verschiedene Initiativen versuchen regelmäßig, diesen Markt zu strukturieren; beispielhaft sei hier die *IoT Landscape*

von TURCK erwähnt (s. TURCK 2016). Parallel entstehen auch Meta-Plattformen, die wiederum zur Integration anderer software-definierter Plattformen genutzt werden können. Ein entsprechendes Architekturmodell dazu schlagen CARREZ ET AL. vor, mit dem Ziel, interoperable Daten aus verschiedensten Quellen einheitlich bereitzustellen (s. CARREZ ET AL. 2017, S. 1).

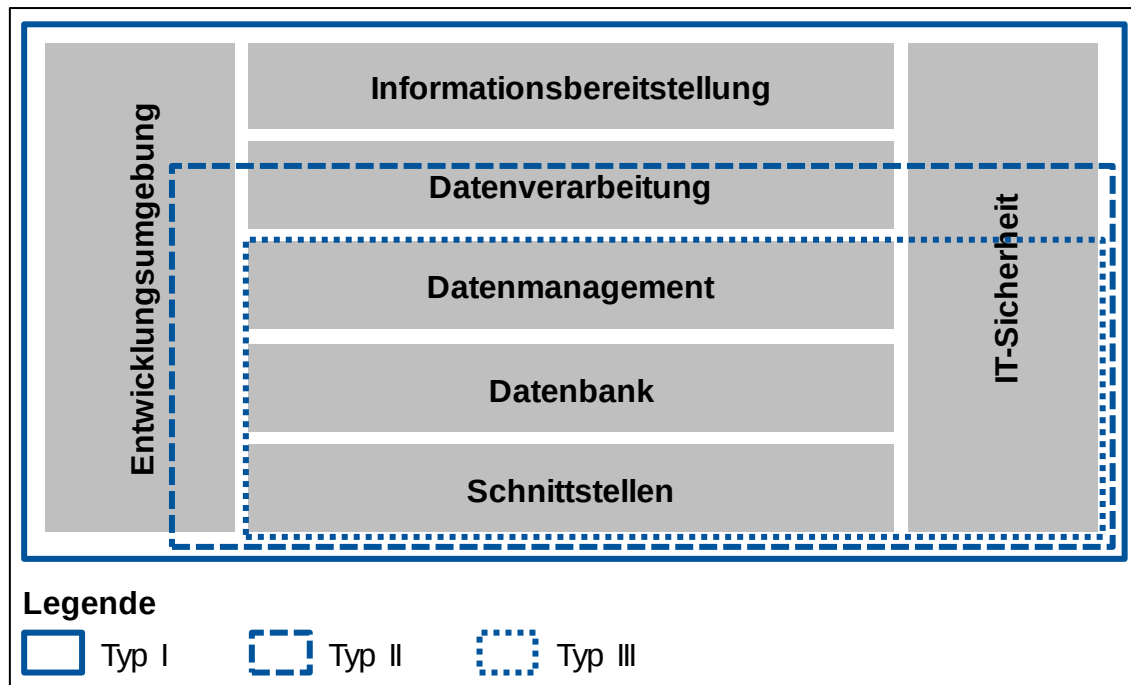


Abbildung 5-10: Schematische Übersicht der Typen software-definierter Plattformen (eigene Darstellung)

Am Markt vorhandene **Plattformen lassen sich nach verschiedenen Kriterien unterscheiden**. Neben den Merkmalen und Funktionen software-definierter Plattformen differieren auch die Kosten und Lizenzmodelle, die Internationalisierung (im Speziellen die Verfügbarkeit in Deutschland) sowie die Zukunftsperspektiven des Anbieters (s. KRAUSE ET AL. 2017, S. 7). In der vorliegenden Arbeit, die die Integration solcher Plattformen in vorhandene Informationssystem-Architekturen produzierender Unternehmen fokussiert, sollen jedoch **allein die Funktionen und Merkmale der vorhandenen Plattformen betrachtet** werden. Daher wird die in Kapitel 5.4 erstellte Morphologie entsprechend dem in Kapitel 4.2.3 beschriebenen Vorgehen genutzt, um daraus gemäß der Empfehlung von KNOBLICH intuitiv-konstruktiv, grundsätzlich basierend auf empirischen Daten, Typen abzuleiten (s. KNOBLICH 1969, S. 30-31). Auf Basis der Untersuchung von am Markt existierenden Plattformen (s. Anhang A.5 u. Tabelle 5-10) lassen sich drei für den Zweck dieser Arbeit relevante Typen ableiten. Der Funktionsumfang der damit typisierten Plattformen baut dabei aufeinander auf: Während in Typ III eine reine Vernetzung von Informationssystemen umgesetzt ist und Typ II Plattformen mit eingeschränktem Funktionsumfang zusammenfasst, entspricht Typ I dem vollumfänglichen Funktionsumfang (s. Abbildung 5-10).

5.5.1 Plattformtyp I: vollumfängliches System

Typ I beschreibt all diejenigen software-definierten Plattformen, welche die oben beschriebenen Merkmale software-definierter Plattformen voll erfüllen – also solche mit maximalem Funktionsumfang. Das bedeutet für alle ordinalskalierten Merkmale, dass sie eine Ausprägung ganz rechts im morphologischen Kasten haben, beziehungsweise für alle anderen Merkmale, dass alle Ausprägungen erfüllt sind. Diese Aussage muss in einem Punkt eingeschränkt werden: Zum Zeitpunkt der Ausarbeitung der vorliegenden Arbeit waren keine Plattformen verfügbar, die im Merkmal „Stufen der Datenverarbeitung“ die höchste Ausprägung „Adaption und Entscheidungsautomatisierung“ erfüllten.

Anbieter software-definierter Plattformen bieten ihre Software teilweise komplett an, teilweise werden einzelne Module auch von Drittanbietern ergänzt. **Beispiele für Plattformen vom Typ I** sind *ThingWorx* von *PTC*, *Predix* von *GE* und *Watson IoT* von *IBM*. Diese werden im Markt auch unter der Bezeichnung IoT-Plattform angeboten.

ThingWorx von *PTC* wird in vielen Marktstudien als eine der führenden IoT-Plattformen beschrieben (s. VOGT 2016, S. 7; CROOK U. MACGILLIVRAY 2017, S. 1; PELINO U. VOCE 2017, S. 8). Die Plattform hilft Industrieunternehmen dabei, intelligente, vernetzte Abläufe, Produkte und Lösungen zu erstellen, zu betreiben und zu warten. Die verschiedenen Komponenten von *ThingWorx* ermöglichen die Systemintegration und Verwaltung von Geräten, Echtzeitanalysen, Zukunftsprognosen sowie Leistungsverbesserungen (s. PTC 2018). Neben einer umfangreichen Protokollkompatibilität werden vielfältige Anwendungspakete geboten. Funktionen wie Machine-Learning, Text- und Bildanalysen, bidirektionale Konnektivität zwischen Sensoren, Geräten und Servern sowie erweiterte Realität gehören ebenso in das Lösungsangebot. Besonders im Bereich der erweiterten Realität nimmt PTC eine Führungsposition ein (s. PELINO U. HEWITT 2016, S. 12). Der Einsatz von *ThingWorx* wird in Kapitel 9.3 näher beleuchtet.

Auch *Predix* von *GE* erreicht vordere Plätze in Marktstudien. Auf Grundlage einer verteilten Anwendungs- und Servicearchitektur (genannt „*Edge to Cloud*“) bietet die Plattform verschiedene Funktionen für Konnektivität, Geräteverwaltung, Datenmanagement, Datenanalysen, IT-Sicherheit und verschiedene Betriebsservices (s. GE DIGITAL 2017). Neben verschiedenen Nutzeroberflächen und Frameworks stellt *GE* zudem Analysemodelle bereit, auf deren Basis eigene Apps entwickelt werden können. Mit *Predix Mobile* werden zusätzliche plattformübergreifende SDKs und ein mobiler Dienst zur Erstellung von Applikationen angeboten (s. CROOK U. MACGILLIVRAY 2017, S. 9-12). *Predix* hebt sich besonders durch die Funktion zur Erstellung digitaler Zwillinge hervor. Hierbei wird es Anwendern ermöglicht, dynamische digitale Modelle von physischen Systemen, wie beispielsweise Windkraftanlagen oder Flugzeugtriebwerken, zu erstellen und zu testen. Diese digitalen Zwillinge unterstützen Anwender dabei, Erkenntnisse über Produktleistungen zu erlangen und mögliche Vorhersagen zukünftiger Operationen zu treffen (s. PELINO U. HEWITT 2016, S. 12).

Die Funktionen von *Watson IoT* von *IBM* können in vier Kernbereiche gegliedert werden: *Konnektivität*, *Risikomanagement*, *Informationsmanagement* und *Analyse*.²² Fokussiert wird hier eine sichere Verbindung zwischen Geräten, Sicherheitsanalysen für einen proaktiven Schutz, Datenspeicherung, -integration und -transformation sowie prädiktive, kognitive, echtzeitfähige Edge-Analysen (s. CROOK U. MACGILLIVRAY 2017, S. 13-14). Dabei zeichnet sich die Plattform besonders durch kognitive Fähigkeiten, eine integrierte Blockchain- und Zahlungsverarbeitungsfunktion, erweiterte Realität, Bild- und Textanalysen oder auch maschinelles Lernen, aus (s. PELINO U. HEWITT 2016, S. 12).

Die genannten Plattformen können im Speziellen zur Erfüllung des in Kapitel 5.1.1 vorgestellten **Ziels „Umsetzung des digitalen Schattens“** eingesetzt werden und bringen damit Informationstransparenz in Unternehmen. Durch die Möglichkeit, Daten verschiedenster Umsysteme zu integrieren, zu speichern und zu verarbeiten sowie als smarte Daten bereitzustellen, eignen sie sich sehr gut zur Erfüllung der Anforderungen beispielsweise des digitalen Schattens der Auftragsabwicklung (vgl. SCHUH ET AL. 2016a). Ein Einsatzbereich kann auch im Aftersales des Unternehmens ausgemacht werden. Dort können beispielsweise bei einem Fahrzeughersteller in einer software-definierten Plattform Kundendaten aggregiert und ausgewertet werden, die aus Informationssystemen wie dem ERP- und CRM-System sowie einem internetbasierten Fahrzeugkonfigurator stammen. Ergänzt werden diese um Echtzeitdaten der einzelnen Fahrzeuge, welche mit einer Sendeeinheit verbunden sind. Neben der Auswertung solcher Daten zur Entscheidungsunterstützung, beispielsweise des Vertriebs, kann die software-definierte Plattform durch die Bereitstellung einer App diese auch dem Fahrzeugnutzer zur Verfügung stellen. Der Nutzer bekommt damit eine Übersicht über seine Fahrzeugverwendung, aber auch Optimierungsvorschläge zur Nutzung seines Fahrzeugs auf Basis der Daten anderer Nutzer können somit an den Nutzer übergeben werden.

Tabelle 5-11 zeigt die Ausprägungen der **einzelnen Merkmale software-definierter Plattformen**, die gemeinsam den **Typ „volumfängliches System“** bilden. Solche Plattformen haben sowohl vordefinierte als auch generische Schnittstellen, um alle nötigen Informationssysteme anzubinden. Da sich die angebundenen Informationssysteme schnell ändern können, unterstützt das Datenmanagement frei konfigurierbare Objekte. Von den Datenbankkonzepten werden prinzipiell alle nötigen, im Speziellen solche für große Datenmengen, unterstützt und die Datenverarbeitung ermöglicht neben Sichtbarkeit und Diagnose auch die Prognose künftiger Zustände. Um die Daten einerseits gut schützen zu können, andererseits aber auch den Zugriff für Mitarbeiter des Unternehmens einfach zu arrangieren, ermöglichen sie die vollständige Integration in die IT-Sicherheitsinfrastruktur des Unternehmens. Wie im oben genannten Bei-

²² Für weitere Informationen s.: <https://www.ibm.com/internet-of-things/spotlight/watson-iot-platform>.

spiel vorgestellt, können solche Plattformen nicht nur einfache Auswertungen und Reports erzeugen, sondern auch Apps und Widgets. Dazu sind entsprechende Entwicklungsumgebungen, bei den meisten Plattformen als offene Entwicklungsumgebung umgesetzt, im Einsatz.

Tabelle 5-11: Typ I Vollumfängliches System (eigene Darstellung)

Merkmal	Ausprägungen			
<i>Schnittstellen</i>	generische Schnittstellen	vordefinierte Schnittstellen		
<i>Datenmanagement</i>	feste Datenstruktur	Datenharmonisierung und -bereinigung	Datenanreicherung	frei konfigurierbare Datenobjekte
<i>Datenbankkonzept</i>	relationale Datenbank	NoSQL-Datenbank	sonstige Datenbankkonzepte	
<i>Stufen der Datenverarbeitung</i>	Erzeugen von Sichtbarkeit	Transparenz und Diagnose	Prognose	Adaption und Entscheidungsautomatisierung
<i>IT-Sicherheitsfunktionen</i>	nur interne Sicherheitsfunktionen	Sicherheit für Plattform und angebundene Systeme	Vollintegration in IT-Sicherheitsinfrastruktur des Unternehmens	
<i>Informationsbereitstellung</i>	keine eigene Auswertungsfunktion	Bereitstellung von Auswertungen und Visualisierungen	Generierung eigener Apps/Widgets	
<i>Entwicklungsumgebung (E.)</i>	keine eigene E.	eigene, proprietäre E.	offene E.	

Legende:



teilweise
erfüllt



vollständig
erfüllt

5.5.2 Plattfortmtyt II: eingeschränkter Funktionsumfang

Mit Typ II werden solche Plattformen zusammengefasst, die für einen spezifischen Anwendungsfall für produzierende Unternehmen entwickelt wurden oder nur für einen solchen eingesetzt werden können. Entgegen der in Typ I vorgestellten Plattformen ist

ihr Funktionsumfang eingeschränkt. Da **in der Praxis vielfältige software-definierte Plattformen** mit Anwendungsfeldern in unterschiedlichen Industriezweigen und auch im privaten Bereich existieren, wird dieser Typ explizit auf die **folgenden Anwendungsfälle eingeschränkt**. Einen Einsatzzweck bildet das in Kapitel 5.1.1 vorgestellte Ziel des **durchgängigen Change-Prozesses**. Eine software-definierte Plattform muss dafür beispielsweise nur mit Anwendungssystemen verbunden werden und nicht mit OT-Systemen aus der Fertigung. Dahingegen muss sie mit den spezifischen Datenmodellen, wie der einer Stückliste (BOM, engl. *Bill of Material*), umgehen können. Spezifische Plattformen können auch die Schnittstelle zu einer Serviceplattform (s. Definition der Smart-Service-Welt in Kapitel 2.3) bilden, auf der datenbasierte Services für den Endkunden angeboten werden. Einen weiteren Anwendungsfall bildet die Umsetzung eines Model-Based-Systems-Engineering-Ansatzes. Dabei werden Systemmodelle als Grundlage des Entwicklungsprozesses verwendet und iterativ verfeinert (s. FRIEDENTHAL ET AL. 2015, S. 15-17). Die Verfeinerung und Validierung wird in modernen Ansätzen unterstützt durch Echtzeiten aus der Anwendung von Produkten des Unternehmens, die mit Simulationsdaten der Modelle verknüpft werden (s. FRIEDENTHAL ET AL. 2015, S. 22-23). Auch hierzu ist, ähnlich wie im ersten Beispiel, die Notwendigkeit zur spezifischen Umsetzung dieses Prozesses gegeben. **Beispiele für solche software-definierte Plattformen** sind *Pulse* von *Agheera* und die *AXOOM-Plattform*.

Die Plattform *Pulse* von *Agheera* unterstützt den spezifischen Anwendungsfall der Datenaggregation und -bereitstellung in der Containerlogistik. Zur Datenerhebung werden smarte Devices sowohl des Unternehmens als auch von Fremdanbietern genutzt. Diese werden zusammen mit externen Daten von Telematik-Service-Providern in der Plattform zusammengeführt.²³ Ein Vorteil der Plattform ist die grafische Visualisierung von Geopositionsdaten und aggregierten Containereigenschaften sowie die Möglichkeit eigener Auswertungen durch den Anwender. Durch den Betrieb in der Microsoft-Cloud ist sie weltweit Verfügbar (s. KRAUSE ET AL. 2017, S. 202-210).

Die modular aufgebaute Plattform *AXOOM* ermöglicht es Anwendern, durch drei verschiedene Konnektoren den Anwendungsfall Maschinenvernetzung umzusetzen. Es können dabei die interne Steuerung der Maschine angesprochen, Standardschnittstellen benutzt, oder auch externe Sensoren angebracht werden. Die Plattform ermöglicht damit die Verwaltung von Maschinen in der Produktion. Maschinendaten können zu jeder Zeit abgerufen und analysiert werden. Störungen, Maschinenfehler und Stillstandszeiten können damit frühzeitig erkannt und beseitigt oder vermieden werden. Die *AXOOM*-Plattform bereitet gesammelte Daten analytisch auf und visualisiert diese auf einem Dashboard. Mit der App-SDK wird zudem ein Werkzeug bereitgestellt, mit dem Kunden selbständig Applikationen entwickeln können.²⁴

²³ Für weitere Informationen s.: <http://www.agheera.com/en/products/agheera-telematics>.

²⁴ Für weitere Informationen s.: <https://axoom.com/de/produkte/iot/>.

Tabelle 5-12: Typ II Eingeschränkter Funktionsumfang (eigene Darstellung)

Merkmal	Ausprägungen			
<i>Schnittstellen</i>	generische Schnittstellen	vordefinierte Schnittstellen		
<i>Datenmanagement</i>	feste Datenstruktur	Datenharmonisierung und -bereinigung	Datenanreicherung	frei konfigurierbare Datenobjekte
<i>Datenbankkonzept</i>	relationale Datenbank	NoSQL-Datenbank	sonstige Datenbankkonzepte	
<i>Stufen der Datenverarbeitung</i>	Erzeugen von Sichtbarkeit	Transparenz und Diagnose	Prognose	Adaption und Entscheidungsautomatisierung
<i>IT-Sicherheitsfunktionen</i>	nur interne Sicherheitsfunktionen	Sicherheit für Plattform und angebundene Systeme	Vollintegration in IT-Sicherheitsinfrastruktur des Unternehmens	
<i>Informationsbereitstellung</i>	keine eigene Auswertungsfunktion	Bereitstellung von Auswertungen und Visualisierungen	Generierung eigener Apps/Widgets	
<i>Entwicklungsumgebung (E.)</i>	keine eigene E.	eigene, proprietäre E.	offene E.	

Legende:

	teilweise erfüllt		vollständig erfüllt

Dargestellt im morphologischen Kasten in Tabelle 5-12 zeigen sich die **Ausprägungen der ordinalskalierten Merkmale** von solchen software-definierten Plattformen eher in der Mitte. Im Detail bedeutet dies: Es sind auch bei diesem Plattfortmtyyp sowohl generische als auch vordefinierte Schnittstellen notwendig, um die verschiedenen relevanten Informationssysteme einfach anbinden zu können. Das Datenmanagement reduziert sich auf die Datenharmonisierung und -bereinigung, da die Erzeugung von smarten Daten (s. Kapitel 2.1.1) für die genannten Anwendungsfälle nicht benötigt wird. Weiterhin nutzen solche Plattformen hauptsächlich relationale Datenbanken, da der Umgang mit unstrukturierten Daten und damit der Einsatz von NoSQL-Datenbanken für die beschriebenen Anwendungsfälle nicht benötigt wird.

Die Datenverarbeitung ist bei den untersuchten Plattformen weniger stark ausgeprägt, für die Stufen „Prognose“ und „Entscheidungsautomatisierung“ wird oft zusätzliche Software von Drittanbietern genutzt. Dagegen ist die Integration in die Sicherheitsinfrastruktur von Unternehmen bei allen Plattformen sehr hoch und wird meist auch explizit als Modul vermerkt. Im Bereich der Informationsbereitstellung haben die im *Typ II* zusammengefassten Plattformen Funktionen zur einfachen Darstellung und Auswertung der enthaltenen Daten oder übergeben die Daten anhand von Schnittstellen an Umsysteme. Entwicklungsumgebungen werden meist nicht mit angeboten, stattdessen stehen Konfiguratoren für die spezifischen Funktionen bereit. Teilweise existieren auch proprietäre Entwicklungsumgebungen.

5.5.3 Plattfortmtyt III: reine Vernetzung

In einem dritten Typ, genannt „reine Vernetzung“, sind software-definierte Plattformen zusammengefasst, die ihren Fokus auf die Vernetzung von Informationssystemen im Bereich der OT legen. Viele heute verfügbare Plattformen aus Typ I haben ihren Ursprung in einer solchen Konfiguration, wurden aber auf dieser Basis weiterentwickelt. Solche Plattformen haben das **Ziel**, die in Kapitel 5.1.1 beschriebene **Software-Definiertheit umzusetzen, also als Abstraktionsschicht oberhalb von physischen Systemen** genutzt zu werden, und die Konfiguration dieser Systeme zu übernehmen. Dies entspricht der Aufgabe der „*Middleware+*“ aus der Infrastruktur des *Internet of Production* (s. Kapitel 5.1.2). Betrachtet man diese Ziele, handelt es sich bei solchen Systemen um eine moderne Umsetzung von Konzepten wie dem ESB oder von EAI, wie sie in Kapitel 3.2 vorgestellt wurden. Daher haben solche software-definierten Plattformen des Typs III auch keine direkte Schnittstelle zum Anwender, sondern dienen streng nach der Ebenenordnung der Smart-Service-Welt als interne Vermittlerschicht, die ihre Daten anderen Informationssystemen bereitstellt. In der konkreten Umsetzung gibt es Unternehmen, in denen beispielsweise schon eine ausgeprägte BI-Systemlandschaft besteht. Diese kann dann mit einer zusätzlichen Datenquelle ergänzt werden. Andere Unternehmen haben Dashboards in Systemen wie Microsoft SharePoint oder als Eigenentwicklung zur Darstellung der Produktionsperformance oder von Finanzkennzahlen entwickelt, auch hier kann eine software-definierte Plattform als zentraler Datenlieferant genutzt werden. **Beispiele solcher Plattformen sind die Connectivity-Plattformen von Vodafone und Ericsson.**

Die „*Managed IoT Connectivity*“-Plattform von *Vodafone* ist eine Verbindungsplattform für die Kommunikation zwischen Systemen. Sie beschränkt sich auf das reine Vernetzen smarter Produkte und erfüllt damit die Eigenschaften einer Typ-III-Plattform. *Vodafone* bietet mit seinem eigenen Mobilfunknetzwerk die Möglichkeit, Geräte weltweit zu verwalten, Daten zu erfassen und Unregelmäßigkeiten hervorzuheben. Dabei kann die Plattform auf ungewöhnliche Aktivitäten hinweisen und Berichte über Leistungs- und Datennutzung erstellen, welche über ein Dashboard dargestellt werden können (s. VODAFONE GROUP 2015).

Ericsson bietet mit dem *IoT Accelerator*²⁵ ebenfalls eine cloudbasierte Verbindungsplattform mit der eine große Anzahl an Geräten vernetzt und verwaltet werden kann. Weitere Kernbereiche der Plattform sind die echtzeitfähige Kostenkontrolle, sowie Verwaltung und Kontrolle von verbundenen Geräten sowie Gerätediagnosen. Dabei verfügt *Ericsson* ebenso wie *Vodafone* über ein flächendeckendes Glasfasernetzwerk, über das die stabile Kommunikation sichergestellt werden kann (s. ERICSSON 2017).

Tabelle 5-13: Typ III Reine Vernetzung (eigene Darstellung)

Merkmal	Ausprägungen			
<i>Schnittstellen</i>	generische Schnittstellen	vordefinierte Schnittstellen		
<i>Datenmanagement</i>	feste Datenstruktur	Datenharmonisierung und -bereinigung	Datenanreicherung	frei konfigurierbare Datenobjekte
<i>Datenbankkonzept</i>	relationale Datenbank	NoSQL-Datenbank	sonstige Datenbankkonzepte	
<i>Stufen der Datenverarbeitung</i>	Erzeugen von Sichtbarkeit	Transparenz und Diagnose	Prognose	Adaption und Entscheidungsautomatisierung
<i>IT-Sicherheitsfunktionen</i>	nur interne Sicherheitsfunktionen	Sicherheit für Plattform und angebundene Systeme	Vollintegration in IT-Sicherheitsinfrastruktur des Unternehmens	
<i>Informationsbereitstellung</i>	keine eigene Auswertungsfunktion	Bereitstellung von Auswertungen und Visualisierungen	Generierung eigener Apps/Widgets	
<i>Entwicklungsumgebung (E.)</i>	keine eigene E.	eigene, proprietäre E.	offene E.	

Legende:



teilweise
erfüllt



vollständig
erfüllt

²⁵ Für weitere Informationen s. <https://www.ericsson.com/en/internet-of-things/solutions/iot-platform>.

Wie die Darstellung des morphologischen Kastens in Tabelle 5-13 zeigt, sind die **ordinalskalierten Merkmale für Typ-III-Plattformen** geringer ausgeprägt als bei den Typen I und II, was mit deren geringerem Funktionsumfang begründet ist. Plattformen dieses Typs verfügen nur über generische Schnittstellen, über die eine Vielzahl von unterschiedlichen OT-Systemen angebunden werden können. Im Kontext des Datenmanagements kann eine Datenharmonisierung und -bereinigung erfolgen, um entsprechend „gesäuberte“ Daten an Umsysteme weiterzugeben. Da es sich bei den Daten hauptsächlich um Werte über Zeitreihen handelt, kommen in solchen Plattformen relationale Datenbanken zum Einsatz, obschon die Nutzung anderer Konzepte meist ebenfalls möglich ist. Eine intelligente Datenverarbeitung ist in software-definierten Plattformen des Typs III nicht vorgesehen, daher können die Daten nur zur Erzeugung von Sichtbarkeit genutzt werden. In besonderem Maße stehen in diesen Plattformen jedoch Sicherheitsfunktionen für die Plattform selbst, sowie aufgrund der „Software-Definiertheit“ auch für die angebundenen heterogenen Systeme, zur Verfügung. Die generierten Informationen werden an andere Systeme weitergegeben, sodass keine Auswertungsfunktionen wie auch keine Entwicklungsumgebungen in diesen Plattformen zur Verfügung stehen.

5.6 Integrationsanforderungen software-definierter Plattformen

Wie bereits die Vorstellung der Merkmale im vorangegangenen Kapitel gezeigt hat, kann eine software-definierte Plattform nicht einfach auf einem Server des Unternehmens installiert und unmittelbar wertschöpfend genutzt werden. Vielmehr stellen sich **Integrationsanforderungen auf unterschiedlichen Ebenen**, die im Folgenden erläutert werden.

Integration von Informationssystemen ist ein seit Jahren durch die Wissenschaft und Praxis stark bearbeitetes Themenfeld (siehe dazu Kapitel 1.1 u. 2.1.4). Rund um die beschriebenen Konzepte des EAI werden heute weitere Konzepte zur Integration entwickelt, sowohl für die *Interunternehmensintegration* als auch für *Intraunternehmensintegration*. Studien zeigen, dass Unternehmen, die ihre Informationssysteme besser integriert haben, Wettbewerbsvorteile erzielen (s. HE u. XU 2014, S. 35). Diese Entwicklung wird künftig noch zunehmen, da Entscheidungen immer mehr von Daten abhängen und mithilfe von Funktionalitäten wie Machine-Learning verbessert werden (s. ZILIS u. CHAM 2016). Gleichwohl ist Integrationsmanagement in Unternehmen wachsende Aufgabe, weil stetig neue Informationstechnologien in Unternehmen Einzug erhalten.²⁶ Es bedarf einer **hohen Qualität der Integration** und auch einer Weiterentwicklung der zugehörigen Methoden und Funktionen, um beispielsweise Fehler-toleranzen zu optimieren (s. HE u. XU 2014, S. 40). Für software-definierte Plattformen

²⁶ So wird beispielsweise der Bereich der kabellosen Technologie mit neuen Entwicklungen wie 5G aufgrund neuer Anwendungsfälle zu weiteren Integrationsnotwendigkeiten führen (s. BERNARDY 2017, S. 12).

sind daher Informationsanforderungen abzuleiten, die im Folgenden strukturiert werden.

5.6.1 Herleitung der Integrationsanforderungen

Als komplexes Problem muss das Integrationsmanagement nun aus verschiedenen Sichten betrachtet werden (s. HABERFELLNER ET AL. 2015, S. 197), die im Kapitel 2.1.4 erläutert wurden. Für die vorliegende Arbeit soll die Integrationsperspektive sowohl vertikal als auch horizontal eingenommen werden; die Integrationsreichweite beschränkt sich auf innerbetriebliche Themen und der Automatisierungsgrad ist abhängig von den Eigenschaften der software-definierten Plattform (s. Kapitel 5.3). **Elementares Strukturierungselement für die folgenden Kapitel ist jedoch der Integrationsgegenstand.** Nach LENT kann der Integrationsgegenstand in die Bereiche der **technischen Integration**, der **Prozessintegration** und der **Integration der Organisation**, beziehungsweise der Personen, eingeteilt werden (s. LENT 2013, S. 263). Da Daten eine zentrale Rolle spielen und im Kontext des Datenmanagements in Kapitel 5.3.2 bereits aufgezeigt wurde, dass die Integration von Daten eine besondere Herausforderung darstellt, soll die **Datenintegration** als vierte Dimension mit betrachtet werden. Abbildung 5-11 visualisiert die vorgenannten Dimensionen und deren Zusammenhänge.

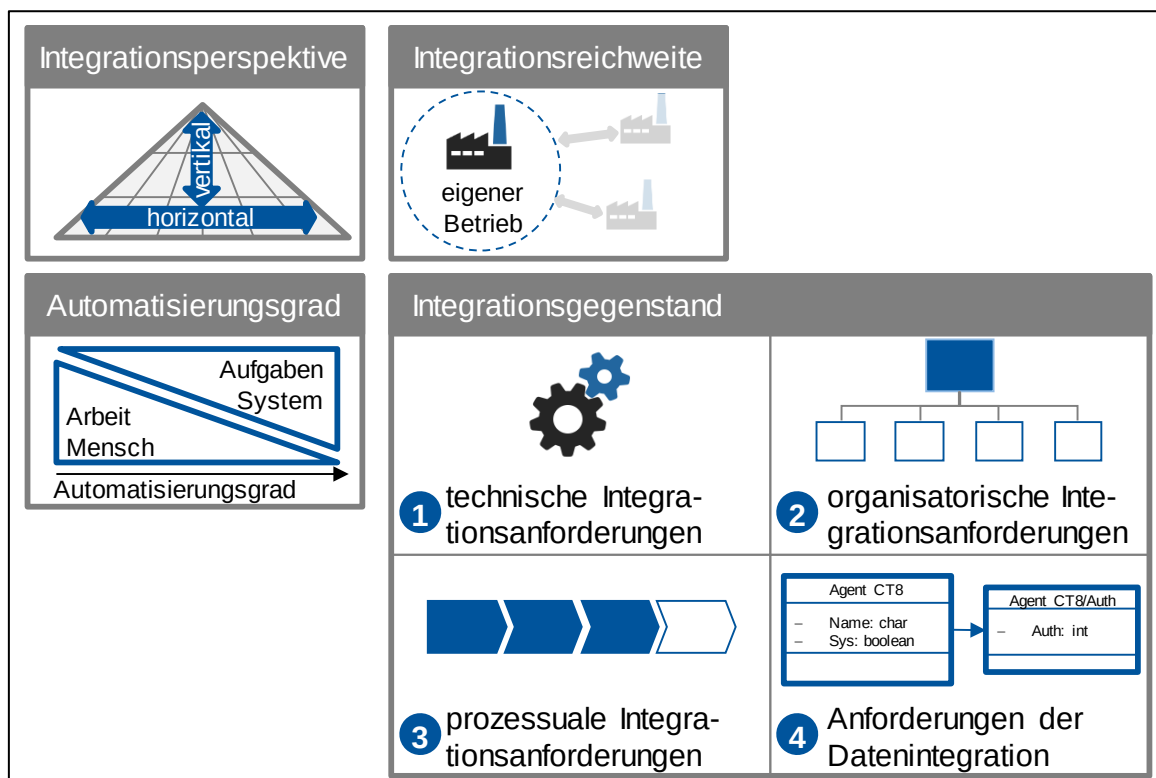


Abbildung 5-11: Dimensionen des Integrationsmanagements software-definierter Plattformen (eigene Darstellung)

Die Integrationsanforderungen software-definierter Plattformen werden im Folgenden nach den vier Dimensionen des Integrationsgegenstands strukturiert. Die Herleitung der einzelnen Anforderungen erfolgt einerseits aus den Plattfortmtypen und den dahinterliegenden Anwendungsfällen, die in Kapitel 5.5 beschrieben wurden, andererseits

aus den Merkmalsbeschreibungen aus Kapitel 5.3. Dabei zeigt sich, dass, entsprechend der Herleitung der Plattfortmtypen, auch deren Integrationsanforderungen aufeinander aufbauen: Anforderungen, die aus Typ III abgeleitet sind, gelten ebenfalls für Typ II und Typ I. Fasst man die **Integrationsanforderungen** zusammen, zeigen sich drei Kernthemen: Die erste Anforderung ist die **Transparenz** über den Ist-Stand der Informationssystem-Architektur – eine Grundbedingung, um Integration sinnvoll zu ermöglichen. Passend dazu ist die zweite Anforderung, **Standards** beispielsweise von Daten oder Schnittstellen umzusetzen. Hinsichtlich der künftigen Nutzung und für die Ableitung von spezifischen Integrationsanforderungen fällt außerdem das Thema der **Anwendungsfälle** auf, an dem sich die Integration eines Informationssystems, wie eine software-definierte Plattform, ausrichtet. Abbildung 5-12 fasst die Integrationsanforderungen für die erhobenen Plattfortmtypen zusammen und teilt sie außerdem nach Anforderungen an *Integrationsergebnis* und sowie an das *Integrationsprojekt* auf.

Integrationsanforderungen für ...	technische Anforderungen		organisatorische Anforderungen		prozessuale Anforderungen		Datenintegrationsanforderungen	
	Ergebnis	Projekt	Ergebnis	Projekt	Ergebnis	Projekt	Ergebnis	Projekt
Typ III	rekonfigurierbare Anbindung passendes Betriebskonzept Konfigurationsmöglichkeit für Umsysteme technisches Anbindungskonzept		geregeltete Betriebsverantwortung etabliertes Sicherheitskonzept Einbezug relevanter Mitarbeiter		in Anwendungsfälle eingebettet		definiertes lokales Datenmodell hohe Informationsqualität Transparenz über vorhandene Daten	
Typ II	Kenntnis der Informationssystem-Architektur		geregeltete Inhaltsverantwortung integriertes Sicherheitskonzept Transparenz über Rollen und Rechte		entwickelte initiale Anwendungsfälle Standardisierte Prozesse Ist-Prozesse bekannt Soll-Prozesse grob entwickelt		Datenstandardisierung ermöglicht	
Typ I	echtzeitfähige Anbindung		angepasste Organisation für schnelle Änderungen umfangreiches Change-management				einheitliches Kerndatenmodell	

Abbildung 5-12: Herleitung der Integrationsanforderungen aus den Plattfortmtypen je Dimension des Integrationsgegenstands (eigene Darstellung)

5.6.2 Technische Integrationsanforderungen

Die Dimension der technischen Integrationsanforderungen beschreibt Anforderungen, die bei der Implementierung einer software-definierten Plattform auf der Ebene von Hardware, Schnittstellen und anderen technischen Ebenen innerhalb des Unternehmens gestellt werden. Konzeptionell zu beachten ist hier, dass die Integration einer software-definierten Plattform eine „Ex-Post“-Integration darstellt, da sie auf eine vorhandene Systemlandschaft aufbaut, die nicht unbedingt auf die Kommunikation mit der Plattform ausgelegt ist.

Im Optimalfall hat die Plattform dedizierte Schnittstellen zu vorhandenen Informationssystemen oder die Systeme verfügen selbst über generische APIs, welche von der Plattform angesprochen werden können. In jedem Fall ist eine transparente Dokumentation der vorhandenen Informationssysteme und Schnittstellen notwendig, um die jeweiligen Endpunkte schnell an die Plattform anschließen zu können. Auf Basis der Dokumentation muss ein **Anbindungskonzept** erstellt werden, in dem die Anforderungen an die Schnittstellen und Umsetzungsmöglichkeiten dargestellt werden. Die Schnittstellen sollten als **rekonfigurierbare Anbindung** umgesetzt werden, durch die keine gegenseitigen Abhängigkeiten zwischen den Systemen entstehen. Da sich die Systemlandschaft schnell ändern kann, sollte der Austausch eines bestimmten Systems keine Auswirkung auf den Betrieb der software-definierten Plattform haben. Über die Kopplung sollte auch die **Konfigurationsmöglichkeit** der Schnittstelle und der angebundenen Informationssysteme bestehen. So sollte beispielsweise bei Änderung des Datenmodells die Möglichkeit zur Anpassung desselben in den Umsystemen durch die software-definierte Plattform automatisiert ablaufen können.

Weitere technische Integrationsanforderungen sind, wie bei jedem anderen Anwendungssystem, die Hardware- und Middlewareumgebung betreffend, für die jeder Softwarehersteller andere Kompatibilitäten angibt. Auch klassische Integrationsthemen des Betriebs wie die Integration in das LDAP oder die Einbindung von Reports der Plattform in vorhandene Reporting-Tools hängen von der jeweiligen Plattform ab. Dazu ist ein passendes **Betriebskonzept** zu entwickeln.

Zu den das Integrationsprojekt betreffenden Anforderungen gehört die **Transparenz über die Informationssystem-Architektur**, in welche die Plattform integriert werden soll. Eine solche Transparenz besteht entweder in Form von Dokumentation über vorhandene Systeme und deren Schnittstellen oder sie muss sich im Zuge des Integrationsprojekts erarbeitet werden. Daraus muss ein entsprechendes technisches Anbindungskonzept entwickelt werden, das als Grundlage für die Umsetzung der Integration genutzt werden kann.

Für Typ-I-Plattformen gilt zudem die Anforderung nach **Echtzeitfähigkeit**, die im Kontext der Big-Data-Analysen entsteht, welche auf der software-definierten Plattform umgesetzt werden sollen. Die Datenmengen, die in einem Unternehmen anfallen, sind zu groß, als dass alle historischen Daten in Verbindung mit Livedaten in Echtzeit ausge-

wertet werden könnten. Daher muss festgelegt werden, auf welche Daten Echtzeitzugriffe und -auswertungen notwendig sind (s. MARZ U. WARREN 2015, S. 14-20). Diese Daten sollten dann möglichst direkt an der Datenquelle mit der Plattform verbunden werden, um nicht beispielsweise zuerst von einem ME-System und einem ERP-System verarbeitet werden zu müssen. Auf technischer Ebene sind außerdem eine schnelle Netzwerkanbindung der entsprechenden Systeme sicherzustellen, genauso wie ausreichend Rechenkraft und Hauptspeicher zur Durchführung der gewünschten Analysen. Ob diese Rechenkraft von einem Cloud-Anbieter bereitgestellt wird oder von Rechnern auf dem Firmengelände, sollte für die Integration einer software-definierten Plattform indifferent sein.

5.6.3 Organisatorische Integrationsanforderungen

Informationssysteme sind soziotechnische Systeme, die stets einen humanen Anteil enthalten. Somit entstehen bei der Integration einer software-definierten Plattform ebenfalls **Anforderungen, die diejenigen Menschen betreffen**, die mit ihr zu tun haben. Hier sind verschiedene Gruppen zu beachten: Diejenigen, die das Integrationsprojekt vorantreiben, diejenigen, die später den Betrieb übernehmen und die, welche Nutzer des künftigen Systems sind, aber auch diejenigen Personen, die aktuell für die Daten verantwortlich sind, die in die Plattform integriert werden sollen. Bei allen Gruppen gibt es soziologische Herausforderungen, die dem Silodenken entstammen, also dem Denken in Gruppen oder Abteilungen ohne den Blick auf die Gesamtunternehmung. Silodenken führt zu Abwehrmechanismen, wenn beispielsweise die „Alleinherrschaft“ über Daten aufgegeben werden muss (s. STAHL U. STAAB 2017, S. 29-30). Hierzu sind vor der Plattformintegration entsprechende organisatorische Maßnahmen zu treffen, um diese Personen in das Projekt einzubeziehen.

Eine Integration kann nur gelingen, wenn ausreichend Wissen über Prozesse, Datenmodelle und Datenquellen im Projektteam vorhanden ist (s. ALPAR ET AL. 2016, S. 415). Es sollte daher darauf hingewirkt werden, dass im Unternehmen eine **offene Kultur** herrscht, in dem die Bereiche ihre Daten mit anderen Bereichen teilen wollen. Sollten Vorbehalte beispielsweise gegen die Arbeitsweise eines anderen Bereichs des Unternehmens bestehen, so sind diese entweder auszuräumen oder bei der Implementierung des Projekts zu beachten.

Die anderen Personengruppen betreffend ist eine Integrationsanforderung für Typ-I- und -II-Plattformen, dass im Rahmen eines Integrationsprojekts festgelegt werden muss, wie und **von wem die Plattform zukünftig betrieben** wird, wer die **Verantwortung für die enthaltenen Daten** und deren **Sicherheit** erhält und wie eine stetige Weiterentwicklung sichergestellt werden kann. Auch der Datenschutzbeauftragte muss rechtzeitig mit in das Implementierungsprojekt aufgenommen werden. Ein entsprechendes Management-Commitment ist dazu unumgänglich.

Damit die Integration nachhaltig gelingen kann, muss im Zuge der Plattformintegration eine **Organisation** entwickelt werden, die eine **schnelle Anpassung** der Prozesse, Datenbestände und Dienste auf der Plattform **ermöglicht**. Eine solche Organisation

muss die Anforderungen der Anwender verstehen und schnell in Lösungen umsetzen können. Sie benötigt daher sowohl eine enge Anbindung an die Fachbereiche als auch ein umfangreiches technisches Verständnis der Informationssysteme des Unternehmens. Ob die Organisation nur aus vom Unternehmen angestellten Mitarbeitern besteht oder auch aus extern beauftragten Firmen, ist in dem Fall irrelevant.

Da sich Prozesse und Vorgänge im Unternehmen durch die Einführung einer software-definierten Plattform ändern werden, sollte die Einführung, wie jedes Softwareeinführungsprojekt, durch **Change-Management-Maßnahmen** begleitet werden. Wenn diese Anforderung erfüllt wird, können die Mitarbeiter selbst die Veränderung mitgestalten und werden so in den Veränderungsprozess einbezogen.

5.6.4 Prozessuale Integrationsanforderungen

Wie bei jeder Softwareeinführung gilt auch für software-definierte Plattformen, dass zu Beginn festgelegt wird, wie die Software im Unternehmen eingesetzt werden soll. Es empfiehlt sich daher, **Anwendungsfälle** zu definieren, die mit der Plattform umgesetzt werden sollen. Solche spezifischen Anwendungsfälle dienen sowohl der Zielfokussierung als auch der einfachen Erfolgskontrolle bei der Implementierung der Plattform. Sie können außerdem dazu genutzt werden, im Zuge des Entscheidungsprozesses für einen bestimmten Softwarehersteller Vergleichbarkeit herzustellen sowie dem Management einfach aufzeigen zu können, welche neuen Möglichkeiten durch die Plattformintegration entstehen. Zur Definition von Anwendungsfällen kann auf bewährte Methoden der Wirtschaftsinformatik zurückgegriffen werden, wie beispielsweise UML-Use-Case-Diagramme (s. ALPAR ET AL. 2016, S. 379-384; LEMKE ET AL. 2017, S. 313-321).

Bei einer tieferen Integration, wie sie beispielsweise bei Typ-II-Plattformen angestrebt wird, sollten zudem die Soll-Prozesse, welche durch die Plattform künftig unterstützt werden sollen, vor der Einführung definiert werden. Hierzu kann auf das vielfach bewährte Drei-Phasen-Konzept zur Softwareeinführung von BROSZE ET AL. zurückgegriffen werden, das in der ersten Phase die **Organisationsanalyse** beschreibt. In dieser werden zuerst Ist-Prozesse und -Strukturen aufgenommen und aus den vorhandenen Schwachstellen sowie neuen Möglichkeiten, welche die neue Software bietet, **Soll-Prozesse** und gegebenenfalls auch **Soll-Organisationsstrukturen** entwickelt (s. BROSZE ET AL. 2009, S. 54-56). Sollten bei der Integration agile Methoden genutzt werden, so darf dies keinesfalls dazu verleiten, kein klares Zielbild zu formulieren. Vielmehr ist das Zielbild feingranular zu entwickeln, um dem agilen Implementierungsteam die notwendigen Bausteine für *Minimal Viable Products* bereitzustellen.²⁷ Agile Methoden unterscheiden sich hierbei nur dadurch, dass sich die Anforderungen über die Zeit

²⁷ *Minimal Viable Products* sind Produkte, die über eine bestimmte Eigenschaft verfügen und mit dieser Eigenschaft bereits nutzbar sind. Sie dienen als Zwischenprodukte in einem agilen Entwicklungsprodukt mit dem Vorteil, direkt eingesetzt und vom Kunden getestet werden zu können. Sie bieten dem Kunden einen minimalen Mehrwert.

und auf Basis der Erfahrungen mit schon bereitgestellten Bestandteilen einer Lösung verändern können.

Wichtig in Bezug auf die prozessualen Integrationsanforderungen ist, dass bei der Definition von Prozessen und deren Aufteilung auf Systeme zu beachten ist, dass in der software-definierten Plattform **nicht die Logik der angebundenen Systeme nachgebaut** werden soll. Die Plattform soll beispielsweise kein ERP-System mit den darin enthaltenen, gegebenenfalls branchenspezifischen Logiken ersetzen, sondern stattdessen auf dessen Funktionen aufbauen.

5.6.5 Anforderungen der Datenintegration

Der letzte Integrationsgegenstand sind die Daten, die wegen ihrer zentralen Rolle hier separat betrachtet werden sollen. Dass ein integrierter und damit nutzbarer Datenbestand für die Weiterverarbeitung und Auswertung von hoher Relevanz ist, hat die Informatik schon lange erkannt. Unter dem Stichwort „*Garbage In, Garbage Out*“ wurde dies sehr treffend in einer Vielzahl an Artikeln, die den Big-Data-Einsatz untersuchen, dargestellt (s. u. a. in RAJPUROHIT 2013, S. 30; NAJAFABADI ET AL. 2015, S. 7). „Schlechte“ Daten können hierbei sowohl durch menschliche Faktoren, wie Schreibfehler bei der manuellen Eingabe, als auch systemisch aufgrund von Fehlkonfigurationen oder defekter Hardware entstehen. Die Anforderung an die Datenintegration ist es daher, für **qualitativ hochwertige Daten** zu sorgen.

Obschon eine software-definierte Plattform die Datenintegration in gewissem Maße unterstützt (s. Kapitel 5.3.2), so ist die Datenintegration dennoch eine umfangreiche Aufgabe, da viel implizites **Wissen über die Zusammenhänge von Daten** im Unternehmen benötigt wird, die in der Plattform hinterlegt werden müssen. Am Beispiel des Stammdatums „*Teilenummer*“ lässt sich dies gut aufzeigen: Die Nummer eines Teils kann in sehr vielen verschiedenen Systemen vorkommen. Ein Konstruktionsteil wird vom Konstrukteur im CAD-System angelegt und im PLM-System gespeichert. Der Einkauf und später die Produktion benötigen es im ERP-System und im Servicefall ist eine Kundenbestellung eines bestimmten Teils unter Umständen auch im CRM-System hinterlegt. Diese nicht abschließende Liste zeigt, dass ein einziges Teil in einem produzierenden Unternehmen durchaus von vier oder mehr Personen in ebenso vielen Anwendungssystemen angelegt werden kann. Wenn nun in einer software-definierten Plattform aufgrund eines Change-Falls eine Auswertung über ein bestimmtes Teil und dessen Vorkommen in verschiedenen Unternehmensprozessen gemacht werden soll, so ist dies nur möglich, wenn es entweder in jedem System dieselbe Teilenummer hat oder in der Plattform eine Übersetzung der verschiedenen Teilenummern durchgeführt wird.

Die für ein einzelnes Teil beschriebene Herausforderung gilt auch auf übergeordneter Ebene für die Produktdatenstruktur. Um Module durch verschiedene Prozesse zu verfolgen, benötigen Unternehmen eine **unternehmenseinheitliche Produktdatenstruktur**. Jedes Unternehmen muss, abhängig von der Komplexität seiner Produkte,

entscheiden, bis auf welche Ebene eine solche Struktur zu definieren ist. Diese Integrationsherausforderung greift auch in die Organisationsstruktur des Unternehmens ein und muss insofern ganzheitlich für das Unternehmen betrachtet werden.

Um die Datenintegration umzusetzen, benötigt es daher zumindest ein grobes **Datenmodell des Gesamtunternehmens**, beziehungsweise des Ausschnitts des Gesamtunternehmens, das durch die Plattform abgedeckt werden soll. In diesem Modell muss auch Transparenz über vorhandene Datentöpfe hergestellt werden (s. TIEMEYER 2017a, S. 117-118).

Zur Umsetzung der Integration kann beispielsweise das Drei-Stufen-Modell nach STAHL U. STAAB angewendet werden. Dieses beginnt mit der logischen Zentralisierung der Daten, die sich dann entweder physisch oder virtuell verknüpft an einer Stelle befinden. Darauf folgt die einheitliche Datenmodellierung, wofür unter anderem einheitliche Konzepte und Attribute benötigt werden. Zuletzt müssen die Daten semantisch angeglichen werden, sodass die Werte einer bestimmten Variablen aus verschiedenen Systemen miteinander verglichen werden können (s. STAHL U. STAAB 2017, S. 23-27).

5.7 Zusammenfassung Kapitel 5

Das vorliegende Kapitel hatte zum Ziel, **software-definierte Plattformen zu beschreiben** und deren **Integrationsanforderungen generisch abzuleiten**. Um dies zu erreichen, wurde eine auf der bisherigen Literatur aufbauende, vertiefende **Definition des Begriffs** „software-definierte Plattform“ entwickelt, beschrieben und im Aachener Modell des *Internet of Production* integriert (s. Kapitel 5.1). Der Fokus des Kapitels liegt auf der Herleitung und Beschreibung von **Merkmale zur Strukturierung** von software-definierten Plattformen. Dazu wurden, basierend auf einer Untersuchung existierender Plattformen, unter anderem aus der Welt der IoT-Plattformen und strukturierter Überlegungen, sieben beschreibende Merkmale und ihre Ausprägungen ermittelt (s. Kapitel 5.2, 5.3 und 5.4). Aus der Untersuchung der Plattformen konnten **außerdem drei generische Typen** software-definierter Plattformen abgeleitet werden (s. Kapitel 5.5). Diese bilden den Ausgangspunkt für weitere Untersuchungen der nachfolgenden Kapitel.

Ein zweiter Schwerpunkt des Kapitels lag in der Darstellung **der Integrationsanforderungen** software-definierter Plattformen. Hierzu wurden eine vierdimensionale Struktur für Integrationsanforderungen entwickelt und auf Basis der drei Plattfortmtypen praxisorientierte Anforderungen an ein Integrationsprojekt und dessen Ergebnis herausgearbeitet (s. Kapitel 5.6). Diese werden im Folgenden in Kapitel 7 verwendet, um ein Zielbild einer Informationssystem-Architektur zu entwickeln und zu erklären.

6 Beschreibung von Informationssystem-Architekturen

Entsprechend dem in Kapitel 4.3 vorgestellten Vorgehen folgt auf die Beschreibung von software-definierten Plattformen und deren Integrationsanforderungen die **Beschreibung von bestehenden Informationssystem-Architekturen** produzierender Unternehmen. Um diese systematisch beschreiben zu können, sind zuerst die darin **enthaltenen Informationssysteme** zu systematisieren. Die Informationssystem-Architektur bildet darauf aufbauend eine Zusammenfassung und Einordnung der Einzelsysteme.

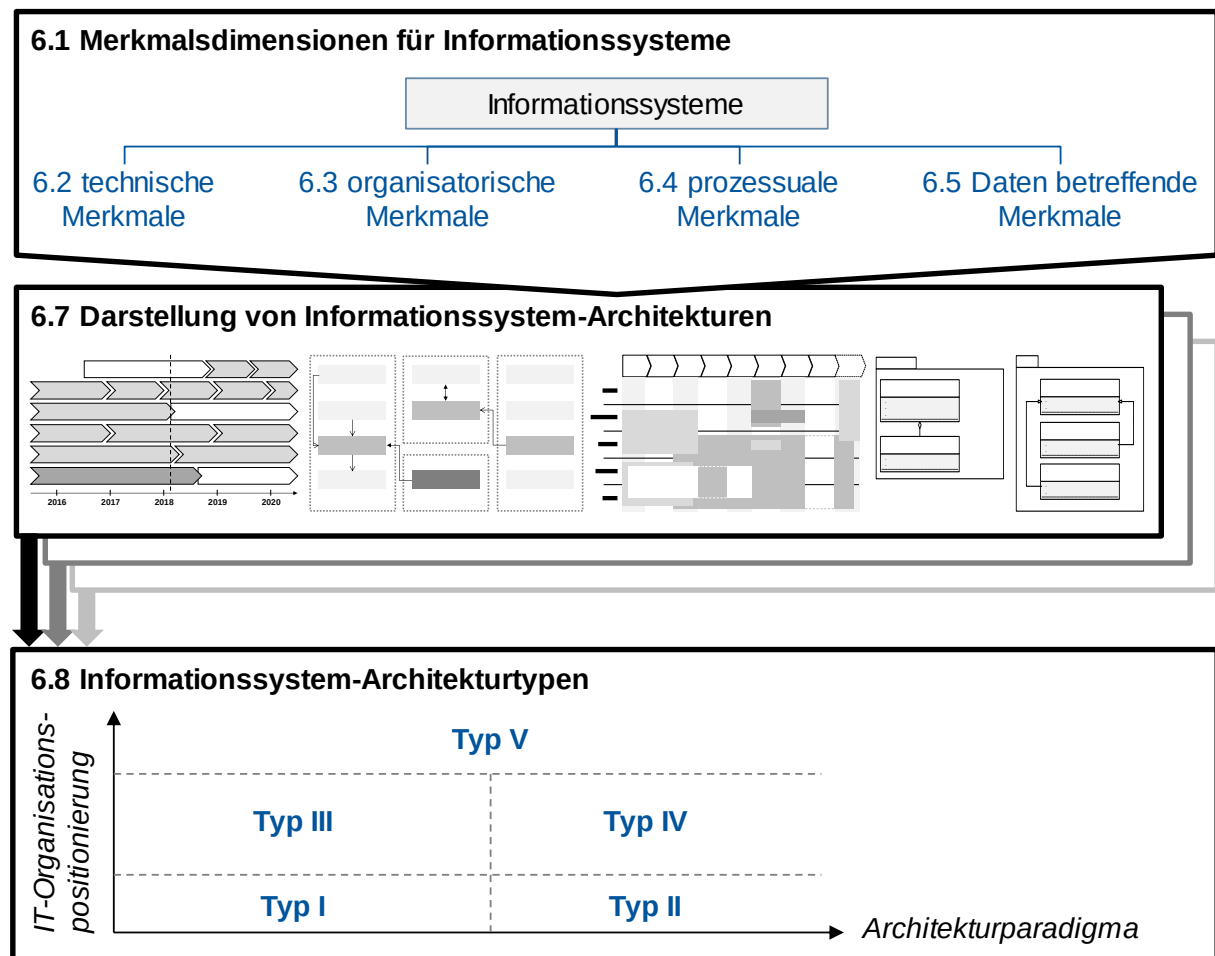


Abbildung 6-1: Übersicht und Zusammenhang der Inhalte des Kapitels 6 (eigene Darstellung)

Wie in Kapitel 2.1.2 dargestellt, werden Informationssysteme in der vorliegenden Arbeit **entsprechend der Definition von KRCMAR** als soziotechnische Systeme zur Bereitstellung von Information und Kommunikation verstanden (s. KRCMAR 2015, S. 22). Diese Definition bezieht sowohl den Menschen mit ein, der das Informationssystem bedient, als auch das Anwendungssystem, die enthaltenen Daten und die zugehörige Hardware. Im folgenden Kapitel werden nicht alle dieser Teilbereiche im gleichen Maße betrachtet: Die zu einem Informationssystem gehörige Hardware wird nur, inso-

fern sie direkte eine Auswirkung auf die Funktionalität der Anwendung hat, mit aufgenommen und ansonsten wegen des Anwendungsfokus dieser Arbeit nicht weiter beachtet. Die Notwendigkeit, Informationssystem-Architekturen nach verschiedenen Dimensionen zu strukturieren, zeigen verschiedene Modelle, die zur Beschreibung der Unternehmens-IT entwickelt wurden; die wichtigsten davon sind in Kapitel 3.3 aufgeführt. Solche Modelle unterscheiden sich jedoch je nach deren Darstellungsziel und fokussieren einzelne Aspekte (s. WOLTERS U. DÜNNEBACKE 2012, S. 36; DÜNNEBACKE 2016, S. 32-35). Dementsprechend wurde auch das **vorliegende Modell für den spezifischen Einsatzzweck** bei produzierenden Unternehmen **als Vorbereitung für die Integration einer software-definierten Plattform** entwickelt und kann dafür angewendet werden (s. Kapitel 9). Es dient dazu, Architekturtypen abzuleiten, für die in Kapitel 8 Gestaltungsempfehlungen zur Optimierung der IT-Komplexität gegeben werden.

Daher werden im Folgenden zuerst die relevanten Merkmalsdimensionen von Informationssystemen hergeleitet (s. Kapitel 6.1), im Einzelnen erklärt, mit ihren Ausprägungen dargestellt (s. Kapitel 6.2, 6.3, 6.4 u. 6.5) und als morphologischer Kasten zusammengefasst (s. Kapitel 6.6). Informationssystem-Architekturen werden daraufhin als ordnungsgebende Strukturen verstanden, die Informationssysteme im zweidimensionalen Raum darstellen. Ergänzt wird diese reine Darstellung um weitere Architekturinformationen, die sich nicht aus den Einzelsystemen ableiten lassen, wie beispielsweise die Informationsflussarchitektur (s. Kapitel 6.7). Durch die Untersuchung von Informationssystem-Architekturen verschiedener produzierender Unternehmen, die im Rahmen dieser Arbeit erforscht wurden, sind Typen dieser Architekturen abgeleitet worden, die in Kapitel 6.8 dargestellt werden. Das damit entwickelte Beschreibungsmodell für Informationssystem-Architekturen beantwortet damit die Forschungsfrage: **Wie können bestehende Informationssystem-Architekturen produzierender Unternehmen systematisch beschrieben werden?**

6.1 Herleitung der Merkmalsdimensionen von Informationssystemen

Wie oben aufgezeigt, werden Beschreibungsmodelle immer für einen bestimmten Zweck erstellt. Der Zweck des folgenden Modells für Informationssysteme ist es, die Integration von software-definierten Plattformen zu ermöglichen. Entsprechend werden Merkmale für Informationssysteme entwickelt, welche für eine solche Integration relevant sind. Die Merkmale werden daher nach den in Kapitel 5.6 entwickelten vier Dimensionen zur Plattformintegration aufgeteilt (s. Abbildung 6-2).

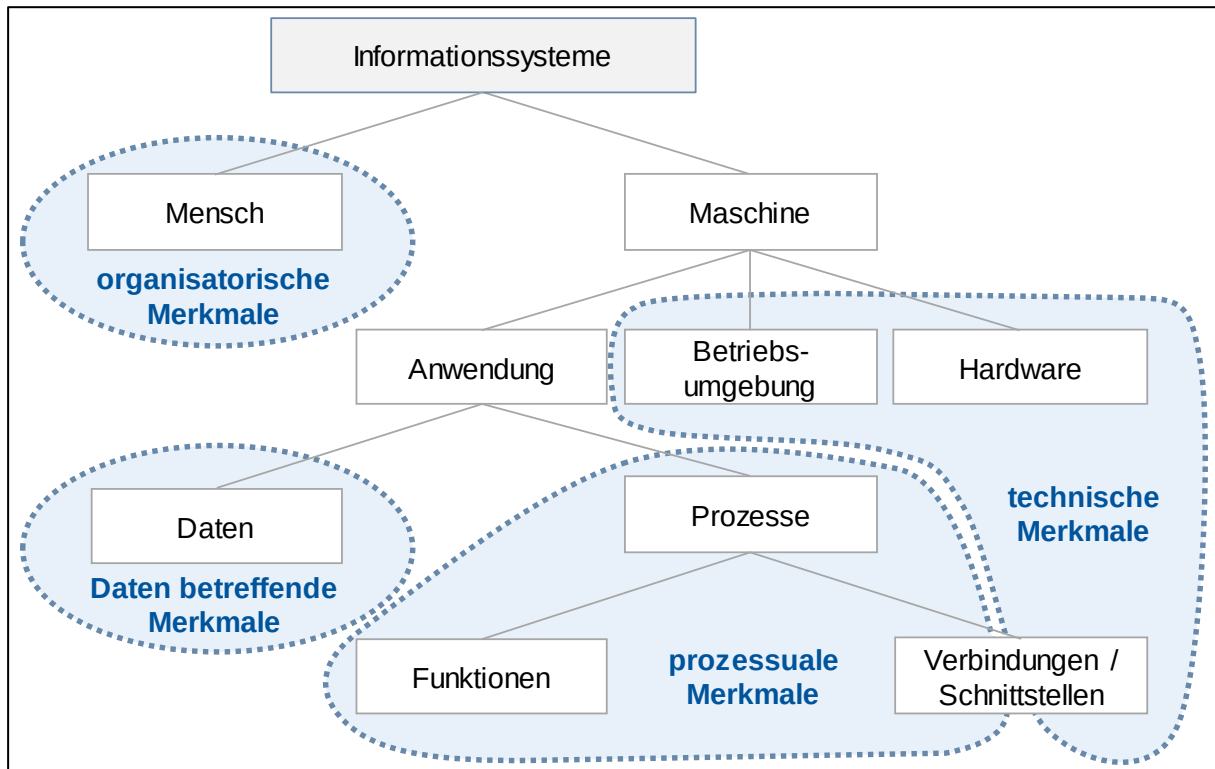


Abbildung 6-2: Ordnung der Merkmalsdimensionen für Informationssysteme (eigene Darstellung i. A. a. KRCMAR 2015, S. 22)

- Technische Merkmale** umfassen die Informations- und Kommunikationstechnik, im Speziellen die Elemente Hardware, Betriebsumgebung und die technische Umsetzung der Schnittstellen aus der Definition von Informationssystemen (s. Kapitel 2.1.2). Entsprechend wurde als ein Merkmal die Kompatibilität der **Schnittstellen** definiert, über die der Datenzugriff erfolgen kann. Auch die **Echtzeitfähigkeit** des Systems und der Schnittstellen wird als Merkmal betrachtet, um das Ziel der Entscheidungsunterstützung durch aktuelle Daten zu ermöglichen. Weiterhin ist der **Outsourcinggrad** ein Merkmal, das sich einerseits auf den Standardisierungsgrad auswirkt, wenn das Informationssystem beispielsweise komplett durch einen externen Dienstleister betrieben wird. Andererseits sinkt damit die Möglichkeit der Einflussnahme darauf, das System zu verändern. Legacy-Software erschwert tendenziell Integration, daher muss auch der **Status des Informationssystems im Softwarelebenszyklus** als Merkmal mit aufgenommen werden.
- Organisatorische Merkmale** betreffen die mit dem Informationssystem interagierenden Menschen in den verschiedenen Lebensphasen des Systems sowohl als Bereitsteller wie auch als Nutzer des Informationssystems. Diese Lebensphasen beginnen mit der ersten Bereitstellung des Informationssystems, bei dem anhand des Merkmals des **Systemursprungs** unterschieden werden muss, ob es sich um eine Standardentwicklung oder ein Individualsystem handelt. Während der Nutzungsphase ist als Merkmal die **Systemverantwortung** mit zu betrachten, um ein Verständnis darüber zu schaffen, welche Personen im Unternehmen bei der Integration zu konsultieren sind. Dazu gehört auch das

Merkmal der **zugeordneten Organisationseinheiten**, das bei der prinzipiellen Einordnung von Informationssystemen in die Unternehmensarchitektur benötigt wird. Wichtig für die Frage, ob ein System einfach integriert werden kann, ist auch das Merkmal des **Nutzerkreises**, das die Menge der mit dem System interagierenden Personen beschreibt.

- Als **prozessuale Merkmale** werden solche verstanden, die einen Bezug zu den Geschäftsprozessen des Unternehmens herstellen, welche das Informationssystem unterstützt. Darin enthalten sind auch die prozessualen, nicht-technischen Übergabepunkte zwischen Prozessschritten durch das Informationssystem. Die beschreibenden Merkmale dazu sind das **Systemkonzept**, das zur generellen Einordnung der Funktionen des Informationssystems dient, sowie der **Kernprozess**, welcher durch das Informationssystem unterstützt wird.
- Zuletzt bilden die **Daten betreffenden Merkmale** alle relevanten Informationen hinsichtlich der Qualität und Struktur der im Informationssystem gespeicherten und verarbeiteten Daten ab. Es wird dazu ein **Ebenenmodell** vorgestellt, um einzuordnen, ob das Informationssystem selbst Daten erzeugt, diese verarbeitet oder nur speichert. Als weiteres Merkmal wird das der **Informationsqualität** angesprochen, da es für die Integration relevant ist, ob den Informationen des Systems vertraut werden kann und ob sie fehlerfrei, vollständig und aktuell sind. Ebenfalls muss untersucht werden, ob das Informationssystem eine **Standardisierung der enthaltenen Daten ermöglicht** oder ob es nur proprietäre Datenformate vorsieht.

Die detaillierten Merkmale wurden sowohl empirisch-induktiv auf Basis der Unterlagen zahlreicher Beratungsprojekte des FIR bei produzierenden Unternehmen – viele davon wurden durch den Autor selbst durchgeführt – als auch analytisch-deduktiv unter Nutzung von bestehenden Modellen und Beschreibungen in der wissenschaftlichen Literatur entwickelt.

6.2 Technische Merkmale

6.2.1 Schnittstellen

Tabelle 6-1: Ausprägungen des Merkmals *Schnittstellen* (eigene Darstellung)

<i>Schnittstellen</i>	generische Schnittstellen	vordefinierte Schnittstellen

Entsprechend der Definition von Informationssystemen sind Schnittstellen eines von deren wichtigen Bestandteilen. Schnittstellen können, wie oben beschrieben, in einen technischen Anteil und einen nicht-technischen Anteil unterteilt werden. Im technischen Anteil werden die Verbindungen, also beispielsweise Protokolle und Austauschformate, welche durch die Schnittstelle unterstützt werden, sowie vor allem deren Erreichbarkeit und Interaktionsmöglichkeit beschrieben. Bei Schnittstellen auf IP-Basis

sind beispielsweise die IP-Adresse und der Port, über die das Informationssystem erreichbar ist, zentrale Informationen. Damit kann zum Beispiel ein Systemintegrator die Interaktion mehrerer Informationssysteme konfigurieren.

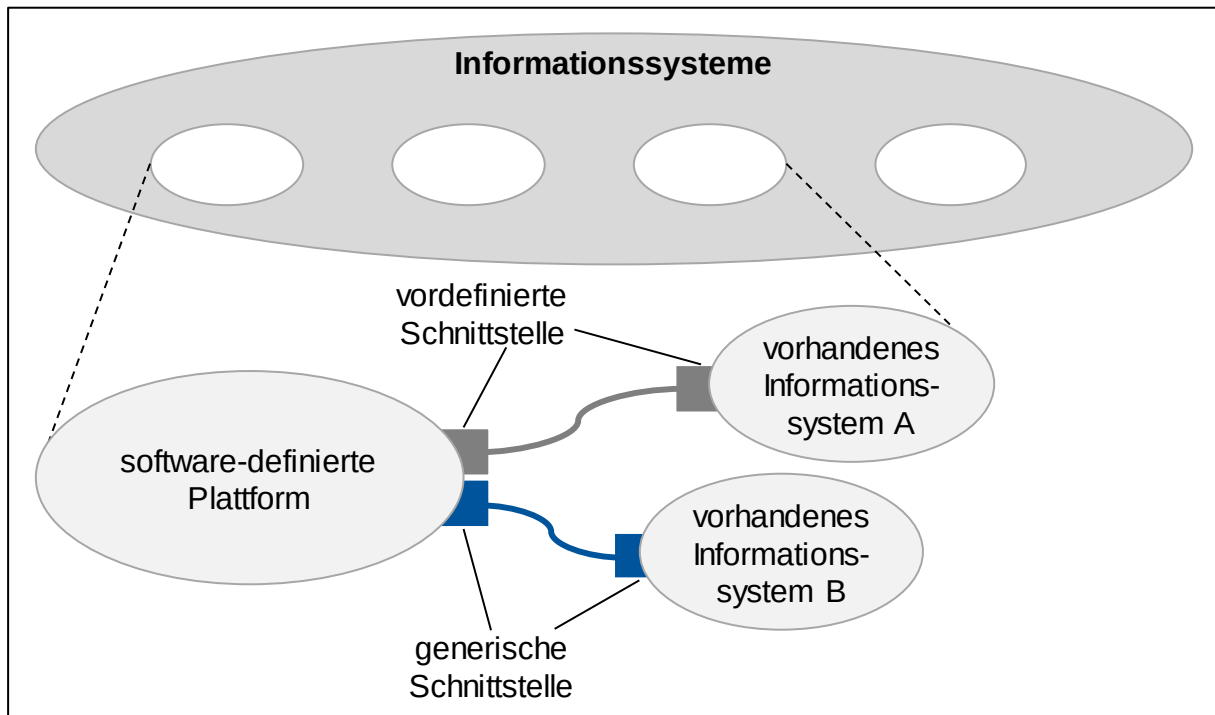


Abbildung 6-3: Schnittstellen zwischen vorhandenen Informationssystemen und software-definierten Plattformen (eigene Darstellung)

Nicht-technische Merkmale betreffen die Funktionen der Schnittstelle, die für jedes Informationssystem unterschiedlich sein können und immer in Verbindung mit dem Prozess stehen, in welchem das Informationssystem genutzt wird. Bei einer REST-Schnittstelle können beispielsweise bestimmte Funktionsaufrufe wie „Adresse ausgeben“ oder „Maschinenstatus schreiben“ zusammen mit mitgelieferten Parametern fest definiert werden. Solche Funktionen können im Rahmen einer morphologischen Beschreibung von Informationssystemen nicht allgemein beschrieben werden, es kann nur die Einordnung des Informationssystems in einen oder mehrere Prozesse vorgenommen werden, in denen es eingesetzt wird. Dies geschieht im Rahmen der prozessualen Merkmale in Kapitel 6.4.

Der technische Anteil der Schnittstelle soll im hier beschriebenen Merkmal „Schnittstelle“ genauso definiert werden, wie schon für software-definierte Plattformen in Kapitel 5.3.1; also mit den Ausprägungen **generische** und **vordefinierte Schnittstellen** (s. Abbildung 6-3). Dies hat zwei Gründe: Erstens bilden die technischen Schnittstellen gewissermaßen den Anknüpfungspunkt zwischen software-definierten Plattformen und bestehenden Informationssystemen des Unternehmens. Daher ist es sinnvoll, dieselbe Beschreibungslogik zu verwenden. Zweitens sind **software-definierte Plattformen** entsprechend der Definition in Kapitel 5.1.2 **selbst Informationssysteme**. Dementsprechend müssen die beschreibenden Merkmale, die sich in den Beschreibungsmodellen überdecken, auch gleich sein.

6.2.2 Outsourcinggrad

Tabelle 6-2: Ausprägungen des Merkmals *Outsourcinggrad* (eigene Darstellung)

<i>Outsourcinggrad</i>	interner Betrieb	Infrastruktur-Outsourcing	Anwendungssystem-Outsourcing	Businessprozess-Outsourcing

IT-Sourcing ist ein weites, weil schon seit über 30 Jahren bestehendes, Forschungsfeld der Wirtschaftsinformatik (s. WILLCOCKS ET AL. 2017, S. 3). Die Vielzahl an verschiedenen Sourcing-Konzepten darzustellen, die über die Zeit entstanden sind, würde den Rahmen dieser Arbeit sprengen. Es sollen an dieser Stelle daher kurz einige Begriffe und die wichtigsten Entwicklungen von IT-Sourcing im Kontext von Informationssystemen vorgestellt werden. Das hier vorgestellte Merkmal *Outsourcinggrad* betrachtet weniger konkrete IT-Sourcing-Varianten, sondern gibt vielmehr einen **Überblick, inwieweit das Informationssystem vom Unternehmen selbst oder von einem Dienstleister betrieben und verwaltet wird** (s. HOFFMANN U. HEIMES 2018, S. 992). Existierende IT-Sourcingkonzepte werden in die Ausprägungen einsortiert.

Unter IT-Outsourcing ist generell die Verlagerung von IT-Dienstleistungen und -Prozessen an IT-Dienstleistungsunternehmen zu verstehen. Die IT-Outsourcing-Entscheidung ist dabei eine Make-or-buy-Entscheidung, die aus individuellen Qualitäts- oder Kostengesichtspunkten getroffen wird (s. ABTS U. MÜLDER 2017, S. 583-584). In seinen Ursprüngen war IT-Outsourcing primär personengebunden und mit der Auslagerung von bestimmten Abteilungen oder Gruppen an Dienstleister verbunden. Bis zum „Platzen“ der Internetblase 2001 entwickelte sich zudem das *Application Service Provisioning* (ASP), bei dem Anwendungen über das Internet vor allem KMU standardisiert bereitgestellt wurden. Aus den zu der Zeit entwickelten Konzepten entwickelte sich ab 2008 das Cloud-Computing-Modell (s. WILLCOCKS ET AL. 2017, S. 6-7; REPSCHLÄGER ET AL. 2010, S. 6). Dort wird unabhängig von den beteiligten Personen ein bestimmter Service als standardisierte Leistung „aus der Cloud“ bereitgestellt (s. dazu auch Kapitel 5.1.1).

In der Informationssystem-Architektur produzierender Unternehmen werden oft viele der im vorherigen Absatz vorgestellten IT-Outsourcingkonzepte parallel eingesetzt, teilweise auch aufgrund von Vorgaben durch den Softwarehersteller.²⁸ Daher ist es wichtig, die vorhandenen IT-Sourcing-Ansätze entsprechend einzusortieren. Einen noch immer gültigen Ansatz haben dazu JOUANNE-DIEDRICH ET AL. mit ihrer IT-Sourcing-Map vorgestellt, die im Folgenden bei der Strukturierung hilft (s. JOUANNE-DIEDRICH ET AL. 2005, S. 23; JOUANNE-DIEDRICH 2007; GADATSCH 2016, S. 21-23). Die

²⁸ Anwendungssysteme der Firma Adobe beispielsweise können nur noch aus der „Creative Cloud“ bezogen werden und bedingen damit beispielsweise ein Arbeiten mit Cloud-Computing-Modellen. Auch Microsoft geht mit Produkten wie „Office 365“ einen ähnlichen Weg, erlaubt allerdings zum aktuellen Zeitpunkt noch die Nutzung von On-Premises-Modellen.

dort verwendete Dimension der Geschäftsorientierung gliedert auf, ob nur der Infrastrukturanteil von Informationssystemen an einen externen Dienstleister übergeben wurde, die Gesamtapplikation oder sogar der komplette Businessprozess. Je nachdem, wie weit das Outsourcing reicht, hat dies Auswirkungen auf die Eingriffsmöglichkeiten des Unternehmens auf das Informationssystem. Bei umfangreicherem IT-Outsourcing sinkt generell betrachtet diese Eingriffsmöglichkeit, was ein Integrationsprojekt unter Umständen erschwert. Gleichzeitig steigt notwendigerweise jedoch auch die Standardisierung des Informationssystems, da ein externer IT-Dienstleister Skaleneffekte nur dann erzeugen kann. Dies kann eine Integration positiv beeinflussen. Abbildung 6-4 stellt diese Zusammenhänge anhand des Ebenenmodells (s. Kapitel 6.5.1) dar.

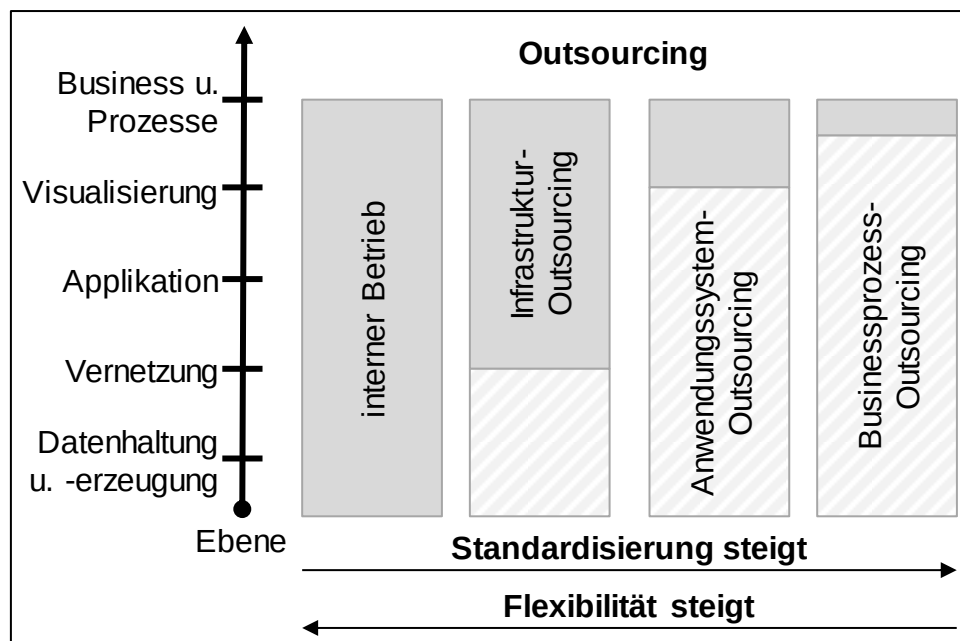


Abbildung 6-4: Einordnung des Outsourcinggrads eines Informationssystems (eigene Darstellung)

Die Ausprägung **interner Betrieb** fasst verschiedene Varianten der vollständigen Eigenverantwortung des Unternehmens für das Informationssystem zusammen. Dies kann der Betrieb durch die eigene IT-Abteilung sein, der gegebenenfalls auch durch externe Berater, die im Unternehmen arbeiten, unterstützt wird. Wie später im Merkmal „Systemverantwortlicher“ beschrieben (s. Kapitel 6.3.2), kann der Betrieb auch durch Nicht-IT-Mitarbeiter, beispielsweise aus dem Fachbereich, erfolgen. Der **Eigenbetrieb umfasst dabei alle technischen Bestandteile des Informationssystems** und bezeichnet damit eine Situation, in der ein Anwendungssystem auf einer unternehmens-eigenen Hardware installiert ist. Die durchgängige Verfügbarkeit des Systems wird dabei vom Unternehmen sichergestellt und Weiterentwicklungen sowie Updates werden vom Unternehmen selbstverantwortlich installiert. Der Standardisierungsgrad des Informationssystems mit seinen Schnittstellen hängt dabei vollständig vom Unternehmen selbst ab.

Die Ausprägung **Infrastruktur-Outsourcing** beschreibt Informationssysteme, bei denen der Infrastrukturbetrieb komplett durch einen externen Dienstleister durchgeführt

wird. Hierunter fallen die beiden Cloud-Computing-Servicemodelle „Infrastructure as a Service“ (IaaS) und „Platform as a Service“ (PaaS). Ersteres umfasst nur die reine Infrastruktur wie Rechner, Speicher und Netzwerke. Im Fall von PaaS sind zusätzlich Entwicklungs- und Laufzeitumgebungen sowie Datenmanagement und Zugriffskontrolle inbegriffen. Im Fall von Cloud-Computing **bedingt das Infrastruktur-Outsourcing eine vollständige Virtualisierung der Infrastruktur**, die bei den meisten produzierenden Unternehmen aufgrund der finanziellen Vorteile und der Flexibilitätsvorteile für ihre Informationssysteme umgesetzt ist. Zusammen mit der Virtualisierung erfolgt eine Standardisierung der Komponenten sowie eine gegebenenfalls notwendige Befähigung der Anwendungssystemkomponenten, eine virtualisierte Infrastruktur zu nutzen.²⁹ Infrastruktur-Outsourcing ist die am weitesten verbreitete Ausprägung des IT-Outsourcings (s. JOUANNE-DIEDRICH 2007; LEMKE U. BRENNER 2015, S. 128; ABTS U. MÜLDER 2017, S. 156).

Aufbauend auf dem Infrastruktur-Outsourcing erfolgt das **Anwendungssystem-Outsourcing**, das als Ausprägung im Folgenden beschrieben wird. Diese Ausprägung beinhaltet sowohl das oben beschriebene ASP-Konzept als auch das daraus weiterentwickelte „Software as a Service“- (SaaS)-Konzept. Hier wird der Betrieb sowohl des Hardware- als auch des **Anwendungsteils des Informationssystems an einen externen Dienstleister** abgegeben. Solche Lösungen verbreiten sich heute immer stärker; viele, auch kleinere, Softwarehersteller bieten ihre Lösung heute so an. Software wird in diesem Fall nicht mehr in Form von Lizenzen verkauft, sondern als Service bereitgestellt. Für Hersteller hat dies den Vorteil angepasster Geschäftsmodelle mit dauerhaften Geschäftsbeziehungen und regelmäßigen Zahlungsflüssen, während produzierende Unternehmen verringerte Investitionskosten stemmen müssen. Der technische Vorteil solcher Lösungen ist die hohe Flexibilität bei Wachstum und erhöhtem Bedarf beziehungsweise auch die Möglichkeit, den Service einzuschränken, wenn er nicht mehr benötigt wird. Allerdings sind Unternehmen, die ein Anwendungssystem in der Form beziehen, auf die Standards des Anbieters beschränkt und können das Informationssystem nur bedingt an ihre eigenen Bedürfnisse anpassen (s. LEMKE U. BRENNER 2015, S. 128; ABTS U. MÜLDER 2017, S. 157).

Die umfangreichste Ausprägung von IT-Outsourcing ist das **Businessprozess-Outsourcing**. Hier werden neben dem Anwendungsteil des Informationssystems auch der „Mensch-Anteil“ des soziotechnischen Systems an einen externen Dienstleister übergeben. Ein verbreitetes Beispiel bei produzierenden Unternehmen ist das Outsourcing von Entgeltabrechnungen oder des Rechnungswesens, beispielsweise an eine Steuerberatungsgesellschaft. Hier wird also nicht nur die technische, sondern auch die **fachliche Verantwortung komplett an einen Dienstleister** übertragen. Im Sinne ei-

²⁹ Darunter zählt beispielsweise die Trennung von Hardware und Software, die bei einigen älteren oder sehr spezifischen Systemen wie CAD-Lösungen noch nicht zum Standard zählt. Ohne eine solche Trennung ist die Umsetzung von Virtualisierungslösungen nicht möglich.

ner Integration von Informationssystemen, die der beschriebenen Ausprägung gehorchen, hat das Unternehmen die geringste Einflussmöglichkeit (s. JOUANNE-DIEDRICH 2007; ABTS U. MÜLDER 2017, S. 585-586).

6.2.3 Echtzeitfähigkeit

Tabelle 6-3: Ausprägungen des Merkmals *Echtzeitfähigkeit* (eigene Darstellung)

<i>Echtzeitfähigkeit</i>	keine Echtzeit	weiche Echtzeit	harte Echtzeit

Die Anforderung nach Echtzeitbereitstellung und -nutzung von Daten und Informationen nimmt in der Industrie-4.0-Diskussion einen hohen Stellenwert ein. Daten sollen beispielsweise genutzt werden, um ein virtuelles Abbild der Fabrik immer aktuell zu halten und damit durch einen Vergleich von Plan- und Istwerten eine Planoptimierung durchzuführen (s. z. B. BAUERNHANSL 2017, S. 12-15; KAGERMANN 2017, S. 236; SCHEER 2017, S. 53). Die **inflationäre Verwendung des Echtzeitbegriffs** geschieht in den meisten Werken jedoch ohne eine klare Definition dessen, was mit „Echtzeit“, teilweise auch „echtzeitnah“, gemeint ist. Dabei ist der Echtzeit-Begriff nicht neu, und vor allem im Kontext von Steuerungssystemen, zum Beispiel von Maschinen, schon seit Jahrzehnten im Einsatz. Nach ISO/IEC 2382 handelt es sich dabei um eine **Rechenoperation**, die mit einem **Zeitskalenfaktor von eins** durchgeführt wird (s. ISO/IEC 2382). In der realen technischen Anwendung von Echtzeit kann diese Zeitskala sowohl im Bereich von Millisekunden wie auch von Sekunden liegen (s. MARZ U. WARREN 2015, S. 209). Die tatsächlich notwendige Zeiteinheit für eine Echtzeitberechnung ist abhängig von den Anforderungen an das System, hängt aber vor allem auch mit dem Systemverhalten in Extremfällen zusammen. Nach BENRA U. HALANG muss ein echtzeitfähiges System den Anforderungen der *Rechtzeitigkeit* und *Gleichzeitigkeit* sowie auch der *Vorhersehbarkeit* und der *Verlässlichkeit* gehorchen. Vor allem bei sicherheitskritischen Systemen, aber auch bei Rechenoperationen, die bei Verzögerung unter Umständen hohen finanziellen Schaden hervorrufen können, sind diese vier Kriterien besonders wichtig (s. BENRA U. HALANG 2009, S. 3-5). Insofern ist die Echtzeitfähigkeit weniger von einer festen Zeitskala abhängig, sondern vielmehr vom verlässlichen Verhalten des Systems bei hoher Last.

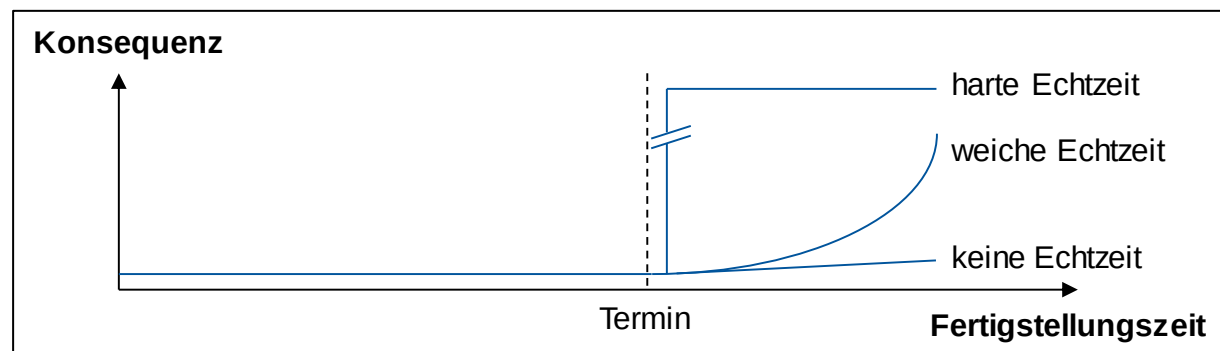


Abbildung 6-5: Einordnung von harter und weicher Echtzeit in Abhängigkeit der Konsequenz (eigene Darstellung i. A. a. BENRA U. HALANG 2009, S. 4)

Für die Beschreibung von Informationssystemen ist es notwendig, das Merkmal der Echtzeitfähigkeit mit zu betrachten, da die Daten von vorhandenen Informationssystemen bei deren Integration in software-definierte Plattformen für Entscheidungen genutzt werden. Dort können offensichtlich nur Echtzeitdaten angezeigt werden, wenn die datenliefernden Informationssysteme dies auch unterstützen. Wie bei eingebetteten Systemen üblich, kann dabei zwischen weicher und harter Echtzeit unterschieden werden (s. Abbildung 6-5).

Wie schon von DÜNNEBACKE vorgeschlagen, soll das Merkmal *Echtzeitfähigkeit* daher in die Ausprägungen der *harten* und *weichen* Echtzeit sowie das Merkmal *keine Echtzeit* eingeteilt werden (s. DÜNNEBACKE 2016, S. 123-124). Harte und weiche Echtzeit unterscheiden sich anhand der Konsequenzen, die eine Verletzung der Echtzeitanforderungen nach sich ziehen. Bei **weicher Echtzeit** sind leichte Verspätungen hinnehmbar, die Konsequenz steigt jedoch mit zunehmender Zeit exponentiell an. Der bei BAUERNHANS� genutzte Begriff „echtzeitnah“ kann hierzu synonym genutzt werden (s. BAUERNHANS� 2017, S. 12). Bei **harter Echtzeit** sind zeitliche Verzögerungen unter keinen Umständen hinnehmbar, weil das Ergebnis einer Berechnung dann nutzlos wird. Im Fall von **keine Echtzeit** nehmen die Konsequenzen bei einer Verzögerung trotzdem zu, allerdings in deutlich längeren Zeitabständen. Entsprechend können Informationssysteme eingeteilt werden in solche, die harte oder weiche Echtzeit unterstützen und solche, die keine Echtzeitfähigkeit beherrschen.

Wenn bei realen Informationssystemen aufgrund der Datenlage nicht geklärt werden kann, inwieweit das System den Echtzeitanforderungen gehorcht, kann dies mit Verifikationsverfahren wie beispielsweise mithilfe der diversitären Rückwärtsanalyse überprüft werden (s. ROTH 2012, S. 91-92).

6.2.4 Status des Softwarelebenszyklus

Tabelle 6-4: Ausprägung des Merkmals *Status des Softwarelebenszyklus* (eigene Darstellung)

<i>Status des Softwarelebenszyklus</i>	initiale Software	Evolution u. Servicing	Ausphasung

Informationssysteme unterliegen einem Lebenszyklus, der sich gemäß der Definition von Informationssystemen in einen Hardwareanteil und einen Softwareanteil unterteilen lässt. Wie oben beschrieben (s. Kapitel 6.1), soll die Hardware jedoch außer Acht gelassen werden, solange sie keine Auswirkung auf die Anwendung als solche hat. Sowohl Hardware für Server- als auch für Clientkomponenten eines Informationssystems werden heute über Betriebssysteme und Middlewarekomponenten soweit vom Anwendungsteil abstrahiert, dass sie nahezu unabhängig vom Lebenszyklus der Softwarekomponenten verändert werden können. Daher wird als Merkmal für Informationssysteme der *Status des Softwarelebenszyklus* herangezogen.

Über Softwarelebenszyklusstatus gibt es in der Literatur und Normung primär Quellen, die sich mit Softwareentwicklung befassen und deshalb die Entwicklungsphase fokussieren (vgl. u. a. ISO/IEC/IEEE FDIS 12207; LEIMEISTER 2015, S. 301-325; BROU U. KUHRMANN 2013, S. 61-83). Fokus dieser Arbeit sind jedoch **Informationssysteme, die bereits aktiv eingesetzter Teil der Informationssystem-Architektur** des Unternehmens sind. Deshalb soll hier in drei Ausprägungen, angelehnt an die Definition von MASAK, unterschieden werden (s. MASAK 2006, S. 33-38): Die erste Ausprägung beschreibt *initiale Software*, die neu im Unternehmen eingeführt wurde und daher dem aktuellsten technischen Stand entspricht.³⁰ Die folgende Ausprägung von *Evolution und Servicing* beschreibt den Status von zunehmend alternder, aber im aktiven Einsatz befindlicher Software. Zuletzt beschreibt die Ausprägung *Ausphasung* solche Software, die nicht mehr aktiv genutzt, aber beispielsweise aus Haftungsgründen noch weiter betrieben wird. Die **Phasen werden nicht immer sequentiell durchschritten**. Oft wird Software beispielsweise umfangreichen Überarbeitungen unterzogen und kann danach wieder der Ausprägung „initiale Software“ zugewiesen werden (s. MASAK 2006, S. 37-38; KÖNIGS 2017, S. 355-357). Abbildung 6-6 abstrahiert davon und zeigt den sequentiellen Verlauf von Aktualität und Nutzerzahl einer Software im Unternehmen, der im Folgenden näher beschrieben wird.

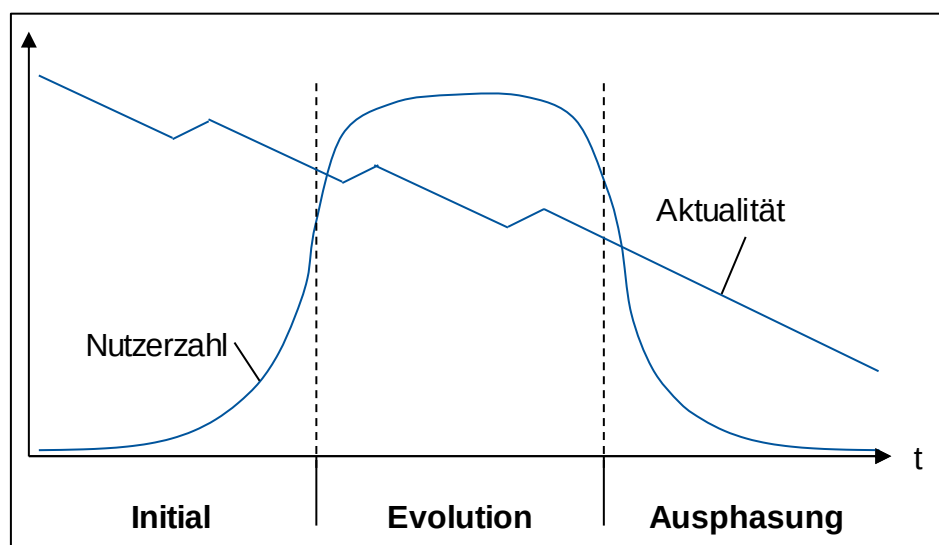


Abbildung 6-6: Nutzerzahl und Aktualität einer Software im Softwarelebenszyklus (eigene Darstellung)

Initiale Software kann entweder schon eingeführt sein, oder sich gerade in der Einführung befinden. In dieser Phase gibt es oft umfangreiche Anpassungen der Software, um sie den Prozessen des Unternehmens anzugleichen. Die Nutzung durch die Anwender ist in einer Phase der Ausbreitung. In den meisten Fällen befindet sich die Software auf dem aktuellsten Stand der Technik und sollte damit auch über aktuelle

³⁰ Da Softwareimplementierungsprozesse in Unternehmen im Einzelfall auch mehrere Jahre dauern können, ist diese Voraussetzung in der Praxis nicht automatisch gegeben, soll aber hier der Vereinfachung wegen angenommen werden.

Standards verfügen und aktuelle Softwarearchitekturen unterstützen, was deren Integration erleichtert. Aufgabe des IT-Betriebs in dieser Phase ist es, den Systembetrieb zu stabilisieren, die Verfügbarkeit zu optimieren und die Anwender optimal zu unterstützen (s. ISO/IEC/IEEE FDIS 12207, S. 91-94).

Die Ausprägung **Evolution und Servicing** beschreibt Software, die sich weiter in der Softwarewartung befindet, aber nicht mehr auf dem neuesten technischen Stand ist. Kleinere Systemoptimierungen werden in dieser Phase weiterhin durchgeführt. Dies kann Optimierungen von verschiedenen Bestandteilen des Informationssystems betreffen (s. LEIMEISTER 2015, S. 308; ISO/IEC/IEEE FDIS 12207, S. 95). Wenn die Software „viele Jahre“³¹ im Einsatz ist, so wird sie als Legacy-Software bezeichnet (s. LEIMEISTER 2015, S. 309; LEMKE ET AL. 2017, S. 327). Diese lässt sich mit zunehmendem Alter immer schwerer ablösen und auch integrieren, da sich die Technologie- und Softwarearchitektur verändert hat, gleichzeitig sinkt die Wartbarkeit und Wartungskosten steigen (s. MARTENS 2016, S. 33-35). Der IT-Betrieb muss in dieser Phase die Steuerung von Verbesserungen übernehmen sowie ständig überwachen, ob die Software in sich verändernden Technologieszenarien weiterhin verfügbar bleiben kann.

Die **Ausphasung**, als letzte Phase vor der kompletten Abschaltung, beschreibt Software, die sich nicht mehr im Service befindet und schrittweise aus der aktiven Nutzung herausgenommen wird. Oft ist Software in dieser Phase so alt, dass es nur noch wenige Personen gibt, welche sie anpassen können und detailliert verstehen. Da in vielen Fällen auch die Dokumentation solcher Systeme nicht mit dem realen Stand übereinstimmt, fällt deren Integration schwer. Wichtig in dieser Phase ist es, die Ausphasung möglichst konsequent zum Ende zu bringen und die Daten in andere Informationssysteme zu migrieren, um das System danach abzuschalten zu können, da sonst unnötige Aufwände und Kosten entstehen.

6.3 Organisatorische Merkmale

6.3.1 Systemursprung

Tabelle 6-5: Ausprägungen des Merkmals *Systemursprung* (eigene Darstellung)

<i>System- ursprung</i>	Individualsystem	angepasstes Standardsystem	Standardsystem

Zu Beginn des IT-Einsatzes in Unternehmen wurden Informationssysteme speziell für deren Einsatz bei einem einzelnen Unternehmen entwickelt. Solche *Individualsysteme* werden auch heute noch genutzt, deren Verbreitung sinkt aber. In den vergangenen etwa 20 Jahren haben stattdessen *Standardlösungen* vermehrt Einsatz bei Unternehmen gefunden (s. ALPAR ET AL. 2016, S. 409), also Informationssysteme, deren Bestandteile in sehr vielen Unternehmen im Einsatz sind. Einen Zwischenweg wählen

³¹ Wann genau der Zeitpunkt „viele Jahre“ eingetreten ist, wird in der Literatur nicht eindeutig festgelegt.

viele Unternehmen, indem sie *angepasste Standardlösungen* nutzen, die auf Standardlösungen aufbauen, jedoch durch individuelle Module ergänzt werden (S. HOFFMANN U. HEIMES 2018, S. 993).

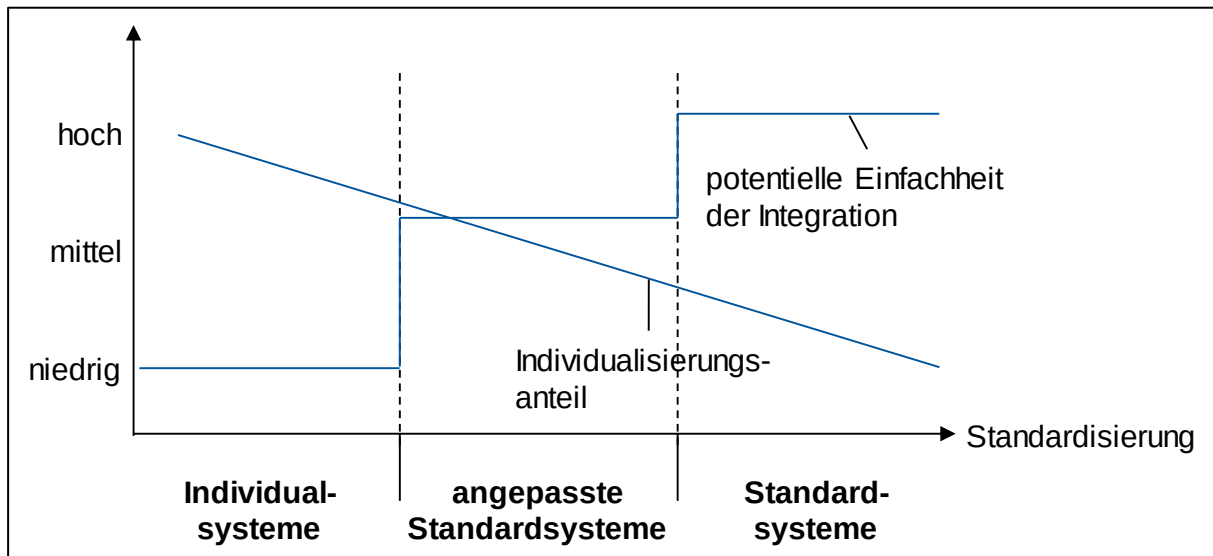


Abbildung 6-7: Darstellung des Systemursprungs von Informationssystemen (eigene Darstellung)

Informationssysteme können daher entsprechend ihres Systemursprungs in die genannten drei Ausprägungen einsortiert werden. Es soll hier nicht näher darauf eingegangen werden, welche Art von Lösung für welchen Einsatzzweck die beste ist, sondern auf entsprechende Standardliteratur verwiesen werden (vgl. u. a. Tabellen in LEHNER ET AL. 2008, S. 211; ABTS U. MÜLDER 2017, S. 59-60). Stattdessen wird im Folgenden detailliert, wie **Informationssysteme entsprechend deren Systemursprung hinsichtlich der Integration in eine software-definierte Plattform einzuschätzen** sind (s. Abbildung 6-7).

Individualsysteme zeichnen sich, neben der individuellen Anpassung auf die Prozesse des Unternehmens, durch ihre gute Integration in die Informationssystem-Architektur des Unternehmens aus, da auch dessen Schnittstellen auf andere vorhandene Informationssysteme angepasst sind. Durch oftmals vernachlässigte Dokumentation lässt sich eine Integration in ein zusätzliches Informationssystem, wie beispielsweise eine software-definierte Plattform, nur schwer bewerkstelligen (s. LEHNER ET AL. 2008, S. 210-211). Da nur die Funktionen implementiert sind, die zum Zeitpunkt der Einführung benötigt wurden, lassen Individualsysteme oft auch Standardschnittstellen vermissen, die für die Integration benötigt werden (s. Kapitel 6.2.1).

Standardsysteme zeichnen sich demgegenüber durch die meist auch vertraglich geregelte Verpflichtung des Systemherstellers aus, Updates und Systemaktualisierungen bereitzustellen. Bei großen Softwareherstellern kann zudem tendenziell davon ausgegangen werden, dass es auch künftig neue Versionen der Software geben und

eine gewisse Kompatibilität zwischen den Versionen sichergestellt wird.³² Standardsysteme beinhalten im Optimalfall die gesammelte Kenntnis des Softwareherstellers über die Best-Practice-Prozesse von ähnlichen Unternehmen. Daher unterstützen sie die Standardprozesse oft sehr gut oder geben Unternehmen sogar Hilfestellung bei der Standardisierung und Optimierung ihrer Prozesse. Die geringere Anpassung auf die Spezifika des Unternehmens kann aber auch zum Entstehen von Zusatzlösungen um das Standardsystem herum führen, was eine Integration erschwert. Standardsysteme haben jedoch den Vorteil, dass sie oft über Standardschnittstellen verfügen und Anbieter von software-definierten Plattformen solche Systeme, die über eine hohe Marktdurchdringung verfügen, oft schon über vordefinierte Schnittstellen eingebunden haben (s. Kapitel 5.3.1).

Den Mittelweg bilden **angepasste Standardsysteme**. Je nachdem, welche Komponente des Systems verändert wurde und wie stark das System vom Standard abweicht, gelten die oben genannten Vor- und Nachteile.

6.3.2 Systemverantwortung

Tabelle 6-6: Ausprägungen des Merkmals Systemverantwortlicher (eigene Darstellung)

Systemverantwortung	IT-Abteilung	Fachbereich	Produktions-IT	Extern

Die Systemverantwortung ist innerhalb der organisatorischen Merkmale ein zentrales Merkmal, da es beschreibt, wer im Unternehmen die Hand über die Entwicklung des Systems, die enthaltenen Daten und auch die Zugriffe hat. Damit gibt es auch Indikationen, mit welchem Personenkreis im Unternehmen eine Integration des Systems organisatorisch zu klären ist.

Die klassische Informationssystemverantwortung liegt bei der unternehmensinternen **IT-Abteilung**. Je nach Unternehmensgröße kann diese unterschiedlich organisiert sein und wird oft auch durch externe Dienstleister unterstützt (s. Kapitel 2.2.4). Als Querschnittsfunktion im Unternehmen hat sie das Ziel der optimalen Informationsbereitstellung und dabei meist keine eigene Datenhoheit über bestimmte Informationssysteme. Hat die IT-Abteilung die Kontrolle über ein IT-System, so ist davon auszugehen, dass es entsprechend den Vorgaben, die das Unternehmen festlegt, betrieben wird.³³ Dazu gehören beispielsweise regelmäßige (Sicherheits-)Updates, Unterstüt-

³² Dass es sich dabei nicht um einen Automatismus handelt, zeigt beispielsweise die im Oktober 2017 verkündete Abkündigung von *Windows Mobile*, dem Smartphonebetriebssystem von *Microsoft* (s. WILKENS 2017).

³³ Erfahrungen des Autors aus Beratungsprojekten am FIR zeigen, dass dieser Optimalzustand, abhängig von der Qualifikation der IT-Mitarbeiter sowie den zur Verfügung stehenden Ressourcen, in der Praxis nicht immer anzutreffen ist.

zung gewisser Standards, eine ausreichende Dokumentation des Informationssystems sowie die zunehmende Integration in andere Informationssysteme (S. BLEIDER ET AL. 2016, S. 20).

Liegt ein Informationssystem nicht in der Hand der IT-Abteilung, sondern eines **Fachbereichs**, wird, wie in Kapitel 2.2.4 beschrieben, von Schatten-IT gesprochen. In solchen Fällen wird das Informationssystem vom Fachbereich sowohl angeschafft als auch betrieben und genutzt. Schatten-IT bringt den Fachbereichen durchaus Vorteile, da ohne langandauernde Abstimmungsschleifen schnell Systeme angeschafft werden können. Solche Informationssysteme unterstützen aber unter Umständen nicht die IT-Vorgaben des Unternehmens, wodurch ihre Integration erschwert wird (S. KOPPER ET AL. 2017, S. 98-100). Bei zunehmender Verlagerung der Informationssysteme in die Cloud sinkt diese Herausforderung jedoch, da dort prinzipiell eine höhere Standardisierung vorherrscht (s. Kapitel 6.2.2).

Der Betrieb von Informationssystemen durch den Unternehmensbereich **Produktion** ist zwar auch ein Fall von Schatten-IT, muss aber aufgrund seiner gesonderten Historie separat betrachtet werden. Begründet liegt diese Historie in der zunehmenden Computerisierung und Vernetzung von Produktionsequipment. Um den informationstechnischen Betrieb solcher Maschinen sicherzustellen, haben sich Rollen und Verantwortlichkeiten innerhalb von Produktionsabteilungen entwickelt. Da die Maschinen anfangs nicht mit dem Unternehmensnetzwerk verbunden waren, war eine Einbindung der IT-Abteilung nicht nötig. Heute ist diese OT jedoch vergleichbar angebunden wie klassische Informationssysteme (s. Kapitel 2.1.6), trotzdem liegen sie oft noch in der Hand des Fachbereichs Produktion und der dortigen IT-Mitarbeiter. Diese Historie sowie die spezifischen Eigenschaften von Produktionsequipment wie beispielsweise dessen längere Lebenszyklen sind bei der Integration zu beachten (S. LARA ET AL. 2016, S. 124; HOFFMANN U. HEIMES 2018, S. 993).

Informationssysteme eines Unternehmens können auch komplett **extern** verantwortet werden, mit dann sehr geringer Einflussnahmemöglichkeit durch das Unternehmen. Ein Beispiel ist der Betrieb der Unternehmenswebseite, beauftragt durch eine externe Marketing-Agentur, wenn zum Beispiel der komplette Business-Prozess Marketing outgesourct ist (s. Kapitel 6.2.2). Oft sind die Daten in solchen Informationssystemen nur von bedingter Relevanz für die Entscheidungsfindung in Unternehmen. In Einzelfällen können sie aber auch von hohem Wert für das Unternehmen sein, wie beispielsweise die Webseitenzugriffe, die als Datenquelle für eine software-definierte Plattform dienen können. Für die Integration solcher Informationssysteme müssen im Einzelfall spezifische Regelungen mit den entsprechenden Dienstleistern getroffen werden.

6.3.3 Organisationseinheit

Tabelle 6-7: Ausprägungen des Merkmals *Organisationseinheit* (eigene Darstellung)

<i>Organisations-einheit</i>	Entwick-lung	Einkauf	Produktion	Vertrieb	Verwaltung u. Manage-ment	übergrei-fend

Unternehmen können als Funktionalorganisation, Spartenorganisation oder Regionalorganisation, beziehungsweise als eine Mischung der vorgenannten, organisiert sein. Da die **Funktionalorganisation**, als eine Gruppenbildung auf Basis von verrichtungsorientierter Spezialisierung, in Deutschland die **vorherrschende Organisationsform von Industrieunternehmen** darstellt, soll sie als Basis für die organisatorische Zuordnung von Informationssystemen zu Unternehmensorganisationseinheiten genutzt werden (s. FRESE ET AL. 2012, S. 422-424). Die Ausprägungen funktionaler Organisationseinheiten sind in der Praxis vielfältig, bedingt durch die Historie des Unternehmens und dessen Produktspektrum. Als immer noch gültiges Modell zur allgemeinen Strukturierung der Organisationseinheiten hat sich die von PORTER entwickelte *Generic Value Chain* durchgesetzt (s. PORTER 1985, S. 37; RÜEGG-STÜRM U. GRAND 2017, S. 128). Diese Struktur wurde von KAMPKER ET AL. leicht zusammengefasst und auf produzierende Unternehmen übertragen, wie Abbildung 6-8 zeigt (s. KAMPKER ET AL. 2011, S. 144). Die Organisationseinheiten lassen sich in Grundfunktionen und Querschnittsfunktionen unterteilen. Grundfunktionen sind Aufgabengebiete, die für das Bestehen des Unternehmens elementar sind. Dies ist vorwiegend der Einkauf, die Produktion, der Vertrieb und Teile des Managements (bspw. die Finanzierung). Die Querschnitts- oder Servicefunktionen haben einen direkten Bezug zu den Grundfunktionen und umfassen neben dem Management der Unternehmung die Verwaltung und das Personalwesen, die Logistik und Entwicklungstätigkeiten.

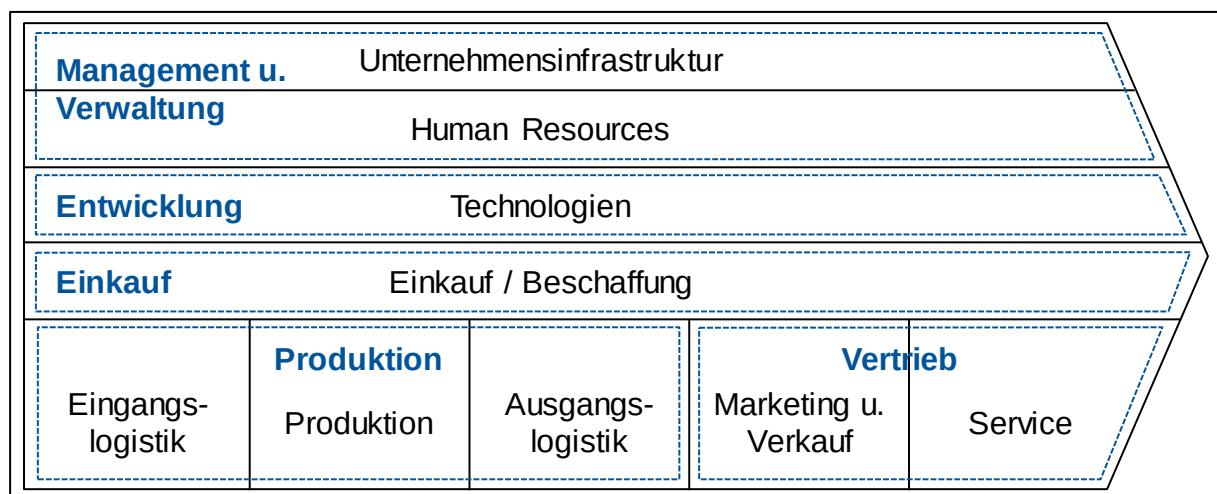


Abbildung 6-8: Funktionale Organisationsstruktur von Unternehmen und Übertragung auf produzierende Unternehmen (eigene Darstellung i. A. a. PORTER 1985, S. 37)

Wie LEHNER ET AL. zeigen, können Informationssysteme diesen Funktionen zugeordnet werden (s. LEHNER ET AL. 2008, S. 180). Ergänzt um die Organisationseinheit „*Verwaltung und Management*“ soll diese Struktur daher für die vorliegende Arbeit übernommen werden. Als weiteres Merkmal wird zusätzlich „*übergreifend*“ definiert, um die Informationssysteme klassifizieren zu können, die nicht spezifisch einer Organisationseinheit zuzuordnen sind, wie beispielsweise ein E-Mail-System.

Das **Management** ist mit der Führung der Unternehmung betraut und gilt nach außen als verantwortlich. Dies umfasst vor allem Aufgaben der Organisation und Planung beziehungsweise der Steuerung des Unternehmens. Die **Verwaltung** führt die administrativen Tätigkeiten aus. Vor allem bei kleineren Unternehmen ist häufig das Personalwesen, also die Beschaffung, Betreuung und Verwaltung des Personals, ebenfalls Teil der Verwaltung. Die **Entwicklung** ist verantwortlich für das Engineering neuer sowie Veränderung bestehender Produkte und Dienstleistungen. Der **Einkauf** kümmert sich um die Beschaffung von Roh-, Hilfs- und Betriebsstoffen sowie Fertigerzeugnissen und stellt die für die Fertigung erforderlichen Materialien zur Verfügung. Die **Produktion** beinhaltet die Erstellung von materiellen und immateriellen Gütern und Dienstleistungen durch Produktionsfaktoren wie Arbeit, Betriebsmittel und Information. Teilweise fällt auch die Logistik der Produktion zu. Der **Vertrieb** ist verantwortlich für den Verkauf der produzierten Güter und Dienstleistungen am Markt. Er kümmert sich außerdem bei kleineren Unternehmen auch um das Servicegeschäft, also die Instandhaltung und -setzung sowie Produkterweiterung beim Kunden.

6.3.4 Nutzerkreis

Tabelle 6-8: Ausprägungen des Merkmals *Nutzerkreis* (eigene Darstellung)

<i>Nutzerkreis</i>	Einzelpersonen	Teilmenge des Unternehmens	unternehmens- übergreifend

Für die Integration eines Informationssystems sowie die Komplexitätsreduktion bildet es einen entscheidenden Unterschied, wie groß der Nutzerkreis dieses Systems ist (s. Abbildung 6-9). Deshalb werden im Folgenden die Unterschiede der möglichen Ausprägungen *Einzelpersonen*, *Teilmenge des Unternehmens* und *unternehmens-übergreifend* herausgearbeitet (s. HOFFMANN U. HEIMES 2018, S. 993).

Hat ein Informationssystem nur **wenige einzelne Nutzer**, so ist es, externe Faktoren ausgenommen, einfacher dieses entweder komplett abzulösen oder auch kurzzeitig stillzulegen, um die Integration umzusetzen. Systeme mit Einzelnutzern sind, nach Erfahrungen des Autors aus der Praxis, meist sehr spezialisierte, oft autark nutzbare Informationssysteme für einen bestimmten Anwendungszweck, wie beispielsweise eine Software zur Unterstützung der Prozesse des Betriebsarztes oder des Fuhrparks eines Unternehmens. Sollten sie geändert werden, beschränkt sich beispielsweise der Schulungsaufwand auf nur diese Einzelnutzer. Auch ein Stillstand dieser Systeme ist zu verkraften, da sie meist keine Kernprozesse unterstützen.

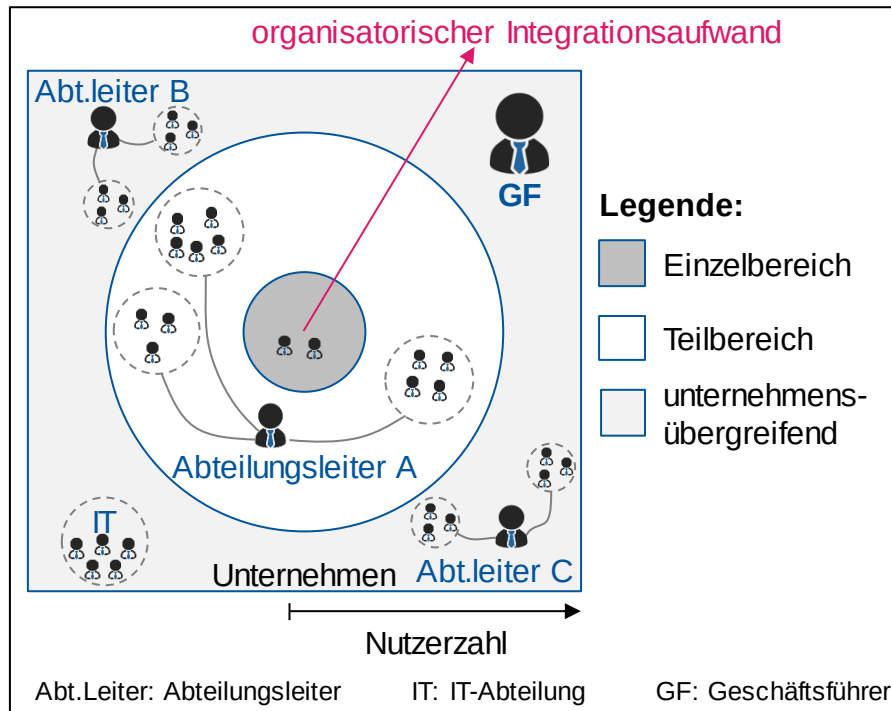


Abbildung 6-9: Zusammenhang zwischen der Nutzerzahl und dem organisatorischen Integrationsaufwand (eigene Darstellung)

Informationssysteme, die von einer definierten **Teilmenge des Unternehmens** genutzt werden, sind oft solche mit einem spezifischen Systemkonzept (s. folgendes Kapitel 6.4.1), die meist von einer oder mehreren spezifischen Organisationseinheiten (s. vorheriges Kapitel 6.3.3) genutzt werden. Organisatorisch kann eine Integration in diesem Fall über zentral Verantwortliche, wie beispielsweise Abteilungsleiter der betroffenen Organisationseinheit, gelenkt werden.

Für Informationssysteme, die **unternehmensübergreifend** genutzt werden, stellt sich die Integration am schwierigsten dar. Hier benötigt es eine zentrale Steuerung und Kenntnis des Systems, beispielsweise durch die IT-Abteilung (s. Kapitel 6.3.2). Gleichzeitig enthalten Informationssysteme, die von nahezu allen Anwendern des Unternehmens genutzt werden, auch die meisten für die Bildung eines digitalen Schattens relevanten Daten und Informationen. Sollten für die Integration solcher Systeme größere Änderungen nötig sein, so ist eine mit allen Beteiligten abgestimmte konkrete Planung erforderlich, um Produktivitätsausfälle zu vermeiden. Auch nach einer erfolgreichen Integration ist eine ständige Abstimmung der Systemverantwortlichen notwendig, um Systemveränderungen abzustimmen und Kompatibilitäten zu gewährleisten.

6.4 Prozessuale Merkmale

6.4.1 Systemkonzept

Tabelle 6-9: Ausprägungen des Merkmals *Systemkonzept* (eigene Darstellung)

<i>Systemkonzept</i>	PLM	ERP	ME	CPS / OT	SCM	CRM	BI	ECM	Sonstige

Um ein Informationssystem richtig zu systematisieren und prozessual einzuordnen, ist das Systemkonzept ein entscheidendes Merkmal. Die einzelnen Konzepte wurden in Kapitel 2.1.6 kurz vorgestellt und sollen im Folgenden als Ausprägungen des Merkmals detailliert werden. Zur Einordnung der Systeme sei auf die Abbildung 2-14 verwiesen; außerdem gibt Anhang A.4 eine Übersicht über die Veränderung der Relevanz der Systemkonzepte über die Zeit – gezeigt am Verlauf der Google-Suchanfragen.

Product Lifecycle Management (PLM)

PLM beschreibt zuerst das Konzept zur ganzheitlichen und unternehmensweiten Steuerung von Produktdaten entlang des Produktlebenszyklus. Der Produktlebenszyklus beginnt bei den Produkthanforderungen und geht über die Entwicklung, Verifikation, Produktion über Produktnutzung und Service bis hin zur Entsorgung. Ein PLM-System unterstützt als zentrale Datenbank die Strukturierung der im Laufe des Produktlebenszyklus entstehenden Daten. Dies umfasst nicht nur das PDM in der Entwicklung, sondern auch Daten, die bei der Produktion oder im Betrieb von Produkten anfallen.

Inputgeber für PLM-Systeme sind eine Reihe von Systemen zur softwaregestützten Entwicklung wie zum Beispiel Systeme zum Anforderungsmanagement, CAD-Systeme für mCAD und eCAD sowie Tools zur Entwicklung von Montageanleitungen oder Servicebeschreibungen (s. KAMPKER ET AL. 2011, S. 187-190).

Strukturiert werden die Daten entlang der Stückliste (BOM), die sich im Laufe des Produktlebenszyklus stetig weiterentwickelt. Ausgehend von der eBOM (Engineering-BOM) wird eine mBOM (Manufacturing-BOM) entwickelt, die weiterhin zu einer sBOM (Service-BOM) ergänzt wird. Hierüber wird auch der Zugriff von Zulieferern geregelt sowie Änderungen strukturiert und Projekte organisiert. Aufgrund seiner Nutzung in den verschiedensten Fachbereichen eines Unternehmens kann das PLM-System ein zentrales Rückgrat einer Informationssystem-Architektur bilden (s. RUDOLF U. SCHREY 2016b, S. 90-92).

Enterprise Resource Planning (ERP)

ERP ist ebenfalls zuerst einmal als Konzept zu sehen, das darauf ausgerichtet ist, die fristgemäße Planung und Steuerung aller Ressourcen im Unternehmen wie Kapital, Personal, Betriebsmittel und Material durchzuführen (s. BACH ET AL. 2017, S. 14). ERP-Systeme enthalten heute eine Vielzahl von Funktionen, welche das genannte Konzept umsetzen (s. LEMKE U. BRENNER 2015, S. 149). Neben den zentralen Funktionen zur Umsetzung des Auftragsmanagements wie Einkauf, Vertrieb, Lagerverwaltung, Produktionsplanung, Produktion und Versand werden auch Funktionen der Buchhaltung,

des Projektmanagements sowie von Service und Instandhaltung angeboten (s. LEIMEISTER 2015, S. 172). Entsprechend bildet ein ERP-System im Unternehmen auch unterschiedliche Tätigkeiten von Mitarbeitern ab, die darauf abzielen, Ressourcen optimal auszulasten und Engpässe zu vermeiden. Dies betrifft, je nach Konfiguration des ERP-Systems im Unternehmen, nicht nur Maschinen und Material, sondern auch Menschen und Kapital (s. KAMPKER ET AL. 2011, S. 184). Eine Studie hat gezeigt, dass heute fast jedes produzierende Unternehmen ein ERP-System einsetzt (s. SCHUH ET AL. 2015, S. 201). ERP-Systeme sind somit eines der am weitesten verbreiteten IT-Systeme zur Unterstützung von Geschäftsprozessen (s. NIEHUES ET AL. 2017, S. 137).

Technisch erfolgt die Datenhaltung in einer relationalen Datenbank, die aus einer Vielzahl von Tabellen besteht, welche über Systemfunktionen und Workflows sowie automatisch ablaufende Kalkulationsjobs verändert werden. Daten werden sowohl über das ERP-System selbst gepflegt als auch über Schnittstellen zu Kunden und Lieferanten sowie von angrenzenden Systemen wie dem SCM-System automatisch ergänzt. Innerhalb von Produktion und Logistik bestehen außerdem Möglichkeiten, über Sensorik aufgenommene Daten zu hinterlegen und damit Prozesse zu automatisieren (s. KAMPKER ET AL. 2011, S. 184-185).

Manufacturing Execution System (MES)

MES sind Systeme zum Fertigungsmanagement, die sich in der Automatisierungspyramide unterhalb des ERP-Systems eingliedern. Im Gegensatz zur Langfristplanung eines ERP-Systems sind MES auf einer Echtzeitplanungsebene zur konkreten Planung, Lenkung, Kontrolle und Auswertung von Aktionen in der Fertigung eingeordnet (s. VDI 5600, S. 1-7). MES umfassen dabei die Funktionsgruppen *Fertigung*, *Produktion* und *Qualitätssicherung*. Im Bereich Fertigung finden sich MES direkt auf dem Shopfloor und ermöglichen die direkte Erfassung von Maschinendaten sowie die Betriebsdatenerfassung über Rückmeldeterminals. Weiterhin wird die Steuerung der Fertigung über Fertigungsleitstände und direkter NC (*Numerical Control*) Steuerung von Maschinen ermöglicht. Ähnliche Funktionen werden auch in den anderen Funktionsgruppen ermöglicht (s. KLETTI 2015, S. 20-22).

Nicht überlappungsfrei ist die Umsetzung der tatsächlich von Softwareherstellern angebotenen Systeme. Dies gilt sowohl bei ERP-Systemen, da dort oft ebenfalls Planungsdaten und Datenerfassung möglich sind oder auch Add-on-Lösungen als MES-Bausteine verfügbar sind (s. NIEHUES ET AL. 2017, S. 137), als auch bei maschinennahen Systemen wie beispielsweise SCADA-Systemen (*Supervisory Control and Data Acquisition*), die zur technischen Steuerung von Prozessen eingesetzt werden. In jedem Fall beinhalten MES Schnittstellen zu den zuletzt genannten Systemen, da langfristige Planungsdaten aus dem ERP berücksichtigt und Ist-Daten wieder zurückgespielt werden, sowie Ausführungsanforderungen an SCADA oder direkt an SPS (Speicherprogrammierbare Steuerung) gesendet werden. Nach aktuellen Umfragen wird die Relevanz von MES in Zukunft noch zunehmen (s. MÜLLER, S. 2015; BOCK 2015).

Operational Technology und CPS

Bei OT handelt es sich gemäß GARTNER um Hardware und Software, die eine Änderung durch die direkte Überwachung oder Kontrolle von physischen Geräten, Prozessen und Ereignissen im Unternehmen erkennen oder verursachen (s. GARTNER 2017b). Die Verantwortung für OT-Systeme liegt oft in produktionsnahen Bereichen und die Planung und Steuerung erfolgt häufig weniger strategisch und mit einem kürzeren Zeithorizont (s. REINHART U. SCHINDLER 2010, S. 1-2; ANDRIOLE 2015, S. 52).

Teil der OT sind auch CPS. Dabei handelt es sich um eingebettete Systeme zur Überwachung und Steuerung physischer Vorgänge mittels Sensoren und Aktuatoren über Kommunikationseinrichtungen mit digitalen Netzen (s. BROY 2010, S. 17). Für den Begriff lassen sich in der aktuellen Literatur vielfältige Definitionen finden, von denen sich bisher keine langfristig etablieren konnte (s. SCHUH ET AL. 2017b, S. 151). Dennoch ist allen Definitionen gemein, dass sich diese durch bestimmte Charakteristika beschreiben lassen. Im Gegensatz zur klassischen OT sind CPS offene Systeme, die Verbindungen zwischen Vorgängen der physischen Realität und den heute verfügbaren Netzinfrastrukturen bieten. Sie bestehen aus Sensorik, Aktorik, Komponenten zur Datenanalyse und definierten Schnittstellen zu anderen Informationssystemen sowie zum menschlichen Bediener (s. SCHUH ET AL. 2017b, S. 152-153). Durch die Integration von CPS können mit einem geringen Kapitaleinsatz die bestehenden Arbeitssysteme der Produktion adaptiv mit Sensoren ausgestattet werden (s. NYHUIS ET AL. 2017, S. 34).

Supply Chain Management (SCM)

SCM-Systeme bilden die softwaretechnische Unterstützung für die Koordination des überbetrieblichen Material- und Informationsflusses über die gesamte Wertschöpfungskette (s. ALPAR ET AL. 2016, S. 297ff). Damit bieten sie Input für die interne Planung im Unternehmen und fokussieren zusätzlich das Management der Lieferketten im Unternehmen (s. LEMKE U. BRENNER 2015, S. 150). SCM basiert auf dem „*Supply Chain Operations Reference Model*“ (SCOR), das in der aktuellen Revision sechs Kernprozesse definiert: Planung, Beschaffung, Produktion, Distribution, Rückgabe und technische Ermöglichung (vgl. dazu SUPPLY CHAIN COUNCIL 2012). Ein wichtiger technischer Aspekt von SCM ist die elektronische Kommunikation zwischen den Partnern, die oft über EDI (s. Kapitel 2.1.4) abgewickelt wird (s. LEIMEISTER 2015, S. 175).

Customer Relationship Management (CRM)

Während sich die vorgenannten Systeme am Produktlebenszyklus beziehungsweise am Auftragsabwicklungsprozess orientiert haben, beinhaltet CRM die sogenannte *Customer Journey*, bei der die Interaktion mit Kunden vom ersten Kontakt, zum Beispiel über die Unternehmenswebseite oder einen Messebesuch, über den Kauf bis hin zum Kundendienst hinterlegt wird (s. LEIMEISTER 2015, S. 176). CRM umfasst dabei sowohl die kundenorientierte Unternehmensstrategie, als auch deren Unterstützung durch Informationssysteme (s. HIPPER ET AL. 2011, S. 18). Letzteres beschreibt die CRM-Systeme, die eine immer stärkere Rolle in Unternehmen spielen. Oft sind die

eingesetzten Systeme Teil von ERP-Systemen, erst etwa 30 Prozent der produzierenden Unternehmen setzen eigenständige Produkte ein (s. SONTOW ET AL. 2015, S. 8). Die Relevanz von CRM-Systemen in der Unternehmens-IT nimmt jedoch deutlich zu. Laut einer Statistik von GARTNER hat der weltweite Umsatz mit CRM-Software 2017 erstmals den Umsatz von ERP-Software überholt; CRM-Software ist somit zur umsatzstärksten Business-Software geworden (s. GARTNER 2017d, S. 1).

CRM-Systeme fokussieren den Kunden und optimieren die Gestaltung der Beziehung zu diesen, um eine bessere Auslastung, einen höheren Umsatz sowie eine bessere Kundenbindung und -zufriedenheit zu erreichen (s. LEMKE U. BRENNER 2015, S. 150). Operativ liefert ein CRM-System Informationen über die Kundeninteraktionen, mit denen über Analysen und Prognosen Auswertungen hinsichtlich eines erwarteten Kundenverhaltens bestimmt werden können. Weiterhin ist das CRM-System die zentrale Datenbank für die kanalübergreifende Bearbeitung von Kundenkontakten, in die neuerdings auch *Social Media* und *Location Based Services* eingebunden werden (s. ALPAR ET AL. 2016, S. 293ff).

Enterprise Content Management (ECM)

ECM bezieht sich, im Gegensatz zu den vorgenannten Systemen, nicht auf einen bestimmten Unternehmensprozess. Vielmehr hat ECM die Aufgabe, den Umgang und die Präsentation von Inhalten (engl. *Content*) zu planen, steuern, verwalten und zu koordinieren (s. BERCHTENBREITER 2004, S. 212). Die dazu nötigen Technologien, wie sichere elektronische Archivierung, Workflowsteuerung oder Dokumentenerfassung, stellen ECM-Systeme bereit (s. RIGGERT 2009, S. 5-6); von manchen Herstellern werden diese auch als Dokumentenmanagementsysteme (DMS) bezeichnet (s. TREUTLEIN ET AL. 2016, S. 8). Damit unterstützen sie eine Vielzahl von Geschäftsprozessen, oft auch informationssystemübergreifend. Klassisch sind die unterstützten Prozesse Verwaltungsprozesse wie der Eingangsrechnungsprozess oder die Urlaubsfreigabe; mit der zunehmenden Digitalisierung werden jedoch auch Funktionen zur unternehmensübergreifenden Zusammenarbeit, des mobilen Zugriffs auf Daten sowie Auswertungen großer Datenmengen unterstützt (s. DIESSNER ET AL. 2015, S. 6).

Business Intelligence (BI)

BI liefert Werkzeuge zur Entscheidungsunterstützung auf Basis von Daten. Es stellt dazu die Informationsversorgung betrieblicher Fach- und Führungskräfte zu Analyse-zwecken sicher, indem Datenpools aus verschiedenen Systemen zusammengestellt werden und daraus standardisierte und spezifische Auswertungen erstellt werden. Dies findet Anwendung in allen Unternehmensbereichen, in denen Planung und Budgetierung eine Rolle spielen sowie Daten visualisiert werden müssen (s. CHAMONI U. GLUCHOWSKI 2016, S. 7-8). Im Hintergrund solcher BI-Systeme liegt oft ein *Data Warehouse*, in das in regelmäßigen Abständen Daten aus internen und externen Quellen aggregiert werden (s. ALPAR ET AL. 2016, S. 246-248). Damit sind auch BI-Systeme keiner bestimmten Rolle im Unternehmensprozess zuzuordnen, sondern bilden, wie auch ECM-Systeme, eine Metaebene über vorhandene Informationssysteme. BI-

Systeme haben sich zur klassischen Datenauswertung beispielsweise von Finanzdaten etabliert, stehen in Bezug auf die Nutzung in der Produktion jedoch teilweise in der Kritik, weil dort die nötige Datenzielstruktur nicht unbedingt bekannt ist (s. THIELE ET AL. 2017, S. 58).

Sonstige Informationssysteme wie Office- und Kollaborationssysteme sowie Spezialsysteme

Zu den meistgenutzten Informationssystemen der Anwender in produzierenden Unternehmen gehören allerdings die Standard-Office-Systeme; hier stellt das Microsoft-Office-Paket einen Quasistandard dar, das in Deutschland einen Marktanteil von 92 Prozent der Bürosoftware hat (s. BRANDT 2015). Viele der Systemfunktionen der oben genannten Systeme werden in einigen Unternehmen auch durch Umsetzung in einem der Office-Anwendungen genutzt, statt sich einer speziell dafür geschaffenen Lösung zu bedienen (s. SCHUH ET AL. 2017h, S. 119).

Weiterhin sind an der Stelle Kollaborationssysteme zu nennen, über die Daten innerhalb eines Unternehmens und zwischen verschiedenen Unternehmen ausgetauscht werden. Solche Systeme reichen von einfachen Datentransfersystemen auf FTP-Basis bis hin zu umfangreichen Lösungen, die zum Beispiel die Projektplanung unterstützen oder eine Chatfunktionalität integrieren.

Zuletzt werden in dieser Ausprägung alle weiteren Systeme zusammengefasst, die keinem spezifischen vorgenannten Systemkonzept zugeordnet werden können. Dies können spezielle Berechnungssysteme, Spezialanwendungen für Sonderrollen wie Betriebsärzte, Qualitätsmanager oder Fuhrparkverantwortliche sein oder auch Informationssysteme, die nur von der IT-Abteilung zu deren eigener Dokumentation genutzt werden.

6.4.2 Kernprozess

Tabelle 6-10: Ausprägungen des Merkmals *Kernprozess* (eigene Darstellung)

<i>Kernprozess</i>	Markt-analyse	Konzept	Entwurf u. Detail-lierung	Produkti-onspla-nung	Herstel-lung	Vertrieb	Nutzung und Ser-vice	Recyc-ling	Unter-stüt-zungs-prozess

Die Anwendung von Informationssystemen im Unternehmen orientiert sich maßgeblich an den unternehmerischen Kernprozessen, welche in Kapitel 2.1.6. bereits kurz vorgestellt wurden. Im Folgenden sollen die Ausprägungen des Merkmals *Kernprozess* in produzierenden Unternehmen detailliert werden. Diese sind hauptsächlich abgeleitet aus der Richtlinie VDI 2219, die Prozesse *Entwurf* und *Detaillierung* werden jedoch zusammengefasst betrachtet (s. Tabelle 6-10). Für die Einordnung von Informationssystemen ist eine weitere Detaillierung an dieser Stelle nicht nötig (s. OEDEKOVEN 2011, S. 29).

Marktanalyse

Die Marktanalyse bietet eine empirische Grundlage betriebswirtschaftlicher und produktpolitischer Entscheidungen. Sie dient der Definition von Zielen, Strategie und Maßnahmen zur Steigerung der Wettbewerbsfähigkeit und Kundenorientierung. Inhaltlich kann sie eine Marktbeobachtung, eine umfassende Sammlung und Analyse von internen und externen Informationen, die Erfassung von Veränderungen im Unternehmensumfeld sowie die Reaktion auf Veränderungen umfassen (s. HERRMANN U. HUBER 2008, S. 31). Produktinnovationen auf Basis von Marktanalysen finden insbesondere im Bereich der Serienproduktion statt. (Sonder-)Anlagen und Einzelanfertigungen werden eher auf Basis spezifischer Angebote produziert (s. OEDEKOVEN 2011, S. 11). Tiefe und Umfang der Marktanalyse hängen stark vom jeweiligen Bedarf und den Kapazitäten des Unternehmens ab.

Konzept

Die Produktkonzeption steht am Anfang der Produktentwicklung und dient der Bewertung und Auswahl der grundsätzlich zu realisierenden Produktfunktionen. Zugrunde liegen die Funktionsstruktur, Wirkprinzipien und Auswirkungen des Produkts während des gesamten Lebenszyklus (s. O'SHEA 2002, S. 12). Das Vorgehen umfasst im Allgemeinen eine Definitionsphase, eine Konzeptphase sowie die Erarbeitung einer prinzipiellen technischen Lösung (s. EHRENSPIEL U. MEERKAMM 2013, S. 265).

Charakteristisch für die Phase der Produktkonzeption ist die hohe Unsicherheit im Planungsprozess, welche großer Datenungenauigkeit und hohem Verbesserungspotenzial auf dieser Stufe geschuldet sind. Gleichzeitig lastet auf der Konzeptphase eine hohe Kostenverantwortung (s. O'SHEA 2002, S. 13).

Entwurf und Detaillierung

An die Phase der Produktkonzeption schließen sich der Entwurf und die Detaillierung an. Inhalt dieses Kernprozesses ist die Übertragung von einem qualitativen Konzept in ein quantitativ funktionsfähiges, gegebenenfalls in Serie produzierbares Produkt. Dabei werden die wesentlichen Bauteile und -gruppen in einem oft iterativen Prozess synthetisiert und analysiert. In der Detaillierungsphase findet die endgültige Gestaltung und Festlegung aller Details statt (s. EHRENSPIEL U. MEERKAMM 2013, S. 267). Hierbei wird unterschieden in Prozessdetaillierung, also die Verfeinerung der Arbeitsschritte, und Objektdetaillierung; die Unterteilung des Objekts erfolgt in Teilobjekte, Subsysteme und Module (s. EHRENSPIEL U. MEERKAMM 2013, S. 340-341). CAD-Systeme sowie PLM-Systeme sind hierfür oft die wichtigsten Werkzeuge (s. EHRENSPIEL U. MEERKAMM 2013, S. 266).

Produktionsplanung

Ein Kernprozess, welcher insbesondere für produzierende Unternehmen hochrelevant ist, ist die Planung und Steuerung der Produktion, oder genauer: der Fertigungs- und Montageprozesse (s. SCHUH U. KAMPKER 2011, S. 180). Sie umfasst die qualitative, quantitative und zeitliche Organisation des Produktionsprozesses, die Planung der Kapazitäten und Aufträge, die Koordination und Steuerung des räumlichen und zeitlichen

Ablaufs in Produktion und Montage sowie die Planung des Produktionsprogramms (s. KISTNER U. STEVEN 2001, S. 3-4; FRANKEN 2004, S. 15).

Herstellung

Der Kern produzierender Unternehmen liegt in der Produktion, also der Transformation von Materialflüssen in Produkte (s. FRANKEN 2004, S. 15). Auch die Arbeitsvorbereitung für Fertigung und Montage mit Losgrößenrechnung, Feinterminierung, Ressourcenfeinplanung, Reihenfolgeplanung, Verfügbarkeitsprüfung sowie die Auftragsfreigabe sind Teil der Herstellung (s. SCHUH U. KAMPKER 2011, S. 182). Das Ziel der Erzeugnisfertigung ist die Realisierung des Wertschöpfungsprozesses für die vorgegebenen Produkte. Diese kann nach verschiedenen Verfahren erfolgen und nach Kriterien wie Prozessart, Technologie, Erzeugnismenge, Kontinuität oder Fertigungsablauf gegliedert sein (s. MAYER 2017, S. 384). Wichtige Instrumente zur Herstellung können Fertigungszellen und -systeme mit Werkzeugmaschinen und Materialhandhabungssystemen sein (s. SCHUH U. KAMPKER 2011, S. 192).

Vertrieb

Der Vertriebsprozess beinhaltet den Ablauf von der ersten Kontaktaufnahme zu Kunden über die Bearbeitung von Anfragen bis hin zur Lieferung und Fakturierung. Klassischerweise stellt der Vertrieb den Übergang von der Herstellung zur Nutzung dar. Dies gilt insbesondere in Unternehmen mit Serienproduktion oder Produktion auf Lager, in welchen der Vertrieb *nach* der Herstellung erfolgt. In Unternehmen mit auftragsorientierter Produktion, zum Beispiel im Maschinen- und Anlagenbau, kann der Vertrieb vorgelagert sein und zwischen Marketing und Produktion erfolgen (s. SCHUH U. KAMPKER 2011, S. 185). Der Vertrieb kann direkt, indirekt oder kooperativ erfolgen. Mit steigender Komplexität des Produkts und der Beschaffungsprozesse, zum Beispiel wie bei Maschinen und Anlagen, wird oft der direkte Vertriebsweg durch den Hersteller gewählt (s. KLEINALTENKAMP U. SAAB 2009, S. 101-104). Der moderne Vertriebsprozess gestaltet sich außerdem zunehmend aus umfassenden Geschäftsmodellen bei welchen die Durchführung ganzer Teilprozesse oder gar der komplette Betriebsprozess von industriellen Produkt-Service-System-Anbietern übernommen wird. Die Wertschöpfung findet bei jenen Geschäftsmodellen vorrangig in der Interaktion statt (s. MEIER U. UHLMANN 2017, S. 22).

Nutzung und Service

Wie der Vertrieb unterziehen sich auch Nutzung und Service neuen Geschäftsmodellen und Angebotsformen. Diese bestehen aus Produkt-Service-Kombinationen, also aus den Vertrieb ergänzende Betreuung und Begleitung des Kunden mit einem hohen Anteil interaktiver Zusammenarbeit zwischen Kunde und Anbieter (s. MEIER U. UHLMANN 2017, S. 19-20). Sogenannte After-Sales-Dienstleistungen leisten hiermit einen unverzichtbaren Beitrag zur Kundenintegration und -bindung. Hierbei kann die strategische Bedeutung dieser Dienstleistungen die des eigentlichen Produktge-

schäfts im Unternehmen übersteigen. Die Gewinnmargen im After-Sales-Service liegen heute bei über 15 Prozent, während sie im Produktgeschäft oft unter 10 Prozent betragen (s. PRANDINI ET AL. 2018, S. 7).

Recycling

Das sich an die Nutzung anschließende Produktrecycling gilt als Teilbereich der Entsorgung und umfasst die Wieder- und Weiterverwendung von Produkten. Hierbei wird unterschieden zwischen Produkt- und Materialrecycling, wobei letzteres die Aufspaltung der Produktgestalt mit sich führt und eine Verringerung des Produktwertes zur Folge hat. Um diesen Prozess zu vereinfachen, hilft eine recyclinggerechte Produktentwicklung. Komponenten dieser können eine demontagegerechte Gestaltung, Vermeidung umweltbelastender Materialien, Auswahl leicht entsorgbarer Werkstoffe sowie deren Kennzeichnung sein (s. SCHUH 2016, S. 5-6). Wenn eine Wiederverwendung der Produktkomponenten oder -materialien nicht möglich ist, beinhaltet das Recycling eine fachgerechte Produktentsorgung.

Unterstützungsprozesse

Unterstützungsprozesse sind jene Prozesse im Unternehmen, welche die wertschöpfenden Kernprozesse unterstützen, selbst jedoch in der Regel nicht an den Kunden und den Markt gerichtet und in der Wertschöpfungskette nicht ausschlaggebend sind. Unterstützungsprozesse sind in der Regel nur an Stakeholder des Unternehmens selbst gerichtet und daher stärker standardisierbar als Kernprozesse. Oft wird die Möglichkeit des Outsourcings wahrgenommen (s. MÜLLER U. NEIDHÖFER 2012, S. 10-11; Kapitel 6.2.2). Zu den typischen Unterstützungsprozessen gehören Buchhaltung und Controlling, Informationsmanagement und IT, Organisationsmanagement, Personalmanagement sowie Qualitätsmanagement. Weitere Unterstützungsprozesse wie Instandhaltungsmanagement oder Beanstandungsmanagement sind möglich (s. BASZENSKI 2008, S. 13).

6.5 Daten betreffende Merkmale

6.5.1 Betroffene Ebenen

Tabelle 6-11: Ausprägungen des Merkmals *Betroffene Ebenen* (eigene Darstellung)

<i>betroffene Ebenen</i>	Datenhaltung u. -erzeugung	Vernetzung	Applikation	Visualisierung	Business u. Prozesse

In Kapitel 2.1.3 wurde vorgestellt, dass es verschiedene Schichtenmodelle zur Einordnung von Informationssystemen, beziehungsweise deren Komponenten, in eine Informationssystem-Architektur gibt. Jedes dieser Modelle hat einen spezifischen Zweck

und wurde daraufhin detailliert. Auf Basis eines Vergleichs der vorhandenen Modelle wurde für diese Arbeit eine eigene Einordnung entwickelt.³⁴

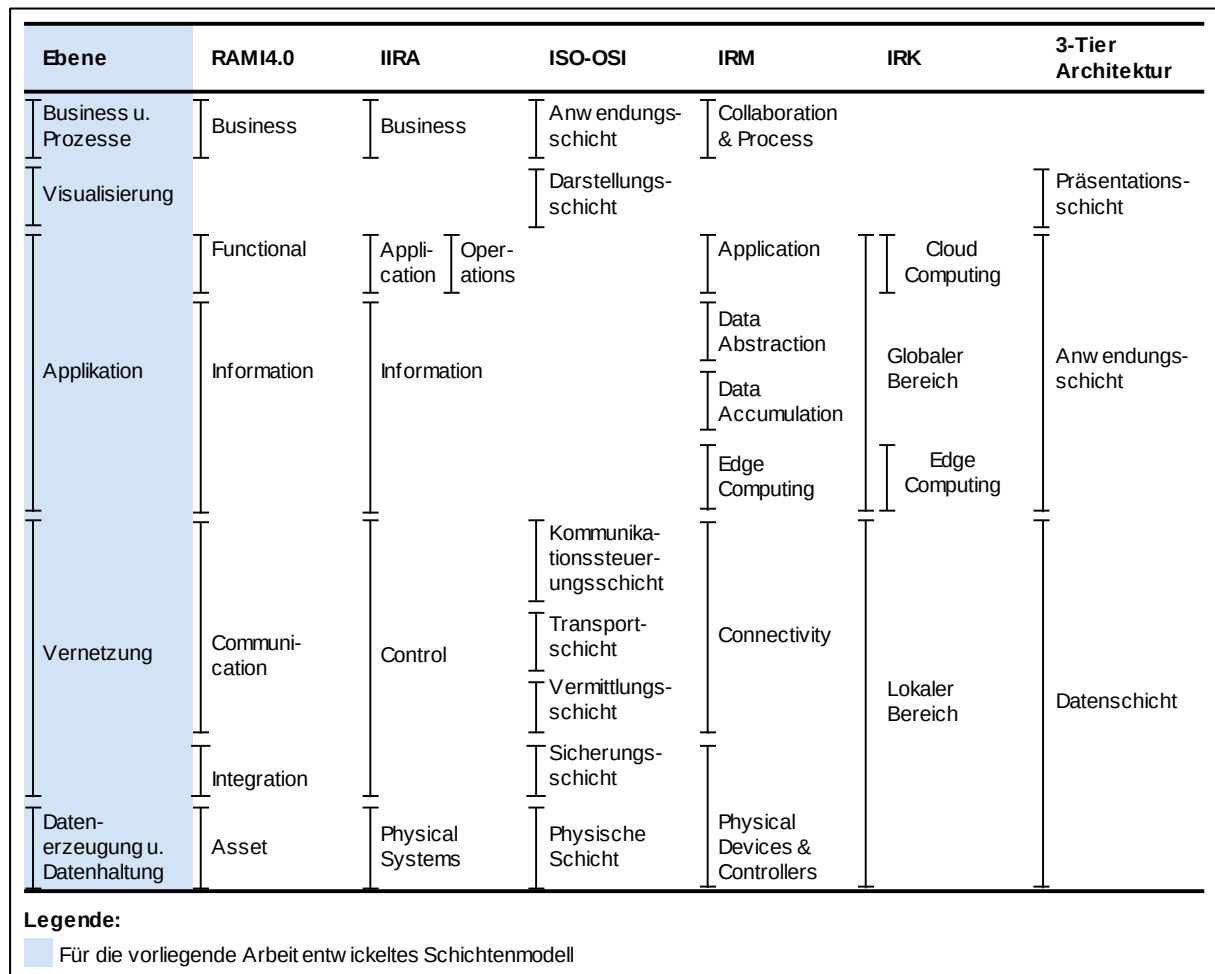


Abbildung 6-10: Einordnung der Ebenen in bestehende Schichtenmodelle (eigene Darstellung i. A. a. SCHUH ET AL. 2018, S. 329 auf Basis von DIN SPEC 91345, S. 19-22; INDUSTRIAL INTERNET CONSORTIUM 2017, S. 27-34; LEIMEISTER 2015, S. 101; GREEN 2014, S. 3-11; HEIDRICH U. JIJUN LUO 2016, S. 20; ALT U. PUSCHMANN 2016, S. 153)

Der Vergleich zeigt, dass sich alle Modelle nebeneinanderstellen lassen und die einzelnen Ausprägungen derselben Sortierung unterliegen (s. Abbildung 6-10): Diese beginnt immer beim physischen System auf unterster Ebene und endet auf oberster Ebene beim Business. Die Daten werden dabei von unten nach oben durch verschiedene Verarbeitungsstufen hochwertiger. Die Unterschiede zeigen sich im Detail. Während die *Layers* des RAMI4.0 sowie die *Functional Domains* der IIRA des IIC relativ ähnlich verteilt sind, hat das ISO-OSI-Modell einen hohen Detaillierungsgrad im Bereich Vernetzung. Das *Internet of Things Reference Model* (IRM) fokussiert dagegen die Datenerzeugung und Verarbeitung, genauso wie auch die *IloT Referenzarchitektur*

³⁴ Eine detailliertere Darstellung des Vergleichs der Schichtenmodelle findet sich im Anhang A.5. Das Schichtenmodell wurde vom Autor zusammen mit anderen Autoren im Voraus veröffentlicht (s. HOFFMANN U. HEIMES 2018, S. 994 sowie für die Herleitung SCHUH ET AL. 2018, S. 329)

für die Kommunikation (IRK), das hier für den „globalen Bereich“ zwischen Edge- und Cloud-Computing unterteilt.

Das für diese Arbeit entwickelte Ebenenmodell, in der Abbildung 6-10 ganz links dargestellt, soll dazu genutzt werden, **Informationssysteme so einzuordnen, dass der Schwerpunkt der mit ihnen durchgeführten Aufgaben dargestellt** wird. Würde diese Einschränkung nicht getroffen, müssten fast immer alle Ebenen markiert werden, da die Bestandteile eines Informationssystems (s. Kapitel 2.1.2) alle Ebenen betreffen. Die genannten Ebenen werden wie folgt als Ausprägungen des Merkmals „*Betroffene Ebenen*“ definiert:

Die unterste Ebene und damit die erste Ausprägung ist die der **Datenerzeugung und Datenhaltung**. Diese Ebene beschreibt einerseits, wie von allen anderen Modellen vorgeschlagen, den physischen Bereich, durch den Daten erzeugt werden. Dies können aus dem OT-Bereich beispielsweise Maschinen oder Sensoren sein. Zudem soll auf dieser untersten Ebene auch die Datenhaltung verordnet werden, wie es bei einer vielschichtigen (n-Tier-)Architektur von Informationssystemen, die in Kapitel 2.1.3 vorgestellt wurde, üblich ist (s. ALT U. PUSCHMANN 2016, S. 153). Dies sind also technische Datenbanken, Fileserver oder andere Systeme zur reinen Datenhaltung.

Aufbauend auf die Datenhaltung folgt die **Vernetzung**, die hier deutlich weniger feingliedrig dargestellt werden soll als in anderen Modellen. Dies liegt darin begründet, dass der Fokus des vorliegenden Beschreibungsmodells nicht auf der technischen Abbildung liegt, sondern Integrationsperspektiven aufzeigen möchte (s. Kapitel 6.1). Da beispielsweise die verschiedenen Transportschichten des ISO-OSI-Modells heute so virtualisiert sind, dass sie von Entwicklern eines Anwendungssystems nicht beachtet werden müssen, sind sie hier zusammengefasst. In diese Ausprägung fallen auch Systeme, die Schnittstellen zwischen Informationssystemen bilden, wie beispielsweise Middleware-systeme.

Die **Applikationsebene** beschreibt die Datenberechnung und -umwandlung, die Funktionen, die ein Informationssystem anbietet, und die Informationen, die dort bereitgestellt werden. Die meisten Informationssysteme haben hier ihren Fokus, da hier die Logik und damit die Kernkompetenz des Softwareherstellers liegt. Für das vorliegende Modell soll keine Unterscheidung getroffen werden, ob die Applikation lokal oder unternehmensextern ausgeführt wird. Der Anwendungsteil kann also sowohl Teil eines monolithischen Informationssystems des Unternehmens sein als auch ein Micro-service in der Cloud.

Die Ebene der **Visualisierung** wird nur vom ISO-OSI-Modell aufgegriffen, obschon für Anwender im Unternehmen die Darstellung der im Unternehmen vorhandenen Daten und Informationen einen entscheidenden Unterschied macht. Dies ist einerseits darin begründet, dass Informationssysteme so gestaltet sein müssen, dass sich ein Anwender gut darin zurechtfindet, damit er das System auch gerne nutzt. Andererseits ist die Informationsflut so hoch, dass es Systeme mit guter Informationsvisualisierung bedarf, um Entscheidungsunterstützung zu leisten. Daher beschreibt diese Ausprägung den Schwerpunkt von Informationsdarstellung, wie beispielsweise durch ein BI-System.

Nahezu alle betrachteten Modelle beschreiben auch die Ebene des **Business** beziehungsweise der **Prozesse** und der Zusammenarbeit. Diese Ausprägung bildet damit die direkte Schnittstelle zum Anwender und stellt die fachliche Funktionalität des Informationssystems dar. Insbesondere werden darin Geschäftsprozesse und deren Schnittstellen abgebildet. Hier wird auch zwischen den oben genannten Systemkonzepten unterschieden (s. Kapitel 6.4.1).

6.5.2 Informationsqualität

Tabelle 6-12: Ausprägungen des Merkmals *Informationsqualität* (eigene Darstellung)

<i>Informationsqualität</i>	vollständig	fehlerfrei	aktuell	zugänglich	keine Informationsqualität

Die Integration von Informationssystemen in eine software-definierte Plattform ist nur dann sinnvoll, wenn die enthaltenen Daten für die Bildung von digitalen Schatten nutzbar sind. Die Daten müssen also in ausreichender Qualität vorliegen. WANG U. STRONG haben daher Datenqualität nach dem in der allgemeinen Qualitätsforschung gültigen „Fit for use“-Prinzip definiert: Datenqualität ist dann gegeben, wenn die Daten aus Sicht der Anwender für den gedachten Einsatzzweck genutzt werden können (s. WANG U. STRONG 1996, S. 6). In der deutschsprachigen Literatur wird statt Daten- teilweise synonym von Informationsqualität gesprochen; die **Unterscheidung zwischen beiden Begriffen wird in der Literatur kontrovers diskutiert**, teilweise werden sie auch uneinheitlich verwendet (s. SLONE 2006, S. 9; ROWLEY 2007, S. 177-178; HILDEBRAND ET AL. 2015, S. IV-V). Für die vorliegende Arbeit soll, entsprechend der Definition von Daten und Informationen in Kapitel 2.1.1, **von Informationsqualität gesprochen werden**, da Daten ohne die Betrachtung des Kontextes nicht von anderen Informationssystemen zur Entscheidungsfindung genutzt werden können.

Aufbauend auf Arbeiten von WANG U. STRONG haben sich im deutschen Sprachraum die Dimensionen der Informationsqualität von ROHWEDER ET AL. durchgesetzt (s. SCHEIBMAYER 2014, S. 38-42). Dieses Konzept geht von einer ganzheitlichen Betrachtung der Informationsqualität aus. Für die vorliegende Arbeit sind jedoch **nur die Dimensionen relevant**, die sich auf die **Übertragung zwischen Informationssystemen beziehen**. So können beispielsweise die darstellungsbezogenen Informationsqualitätsdimensionen von ROHWEDER ET AL. hier außer Acht gelassen werden, da die Informationsdarstellung auch, nach einer möglichen Transformation der Informationen, innerhalb der software-definierten Plattform geschehen kann. Abbildung 6-11 gibt eine Übersicht über diese Dimensionen und ihre Nutzung für die vorliegende Arbeit.

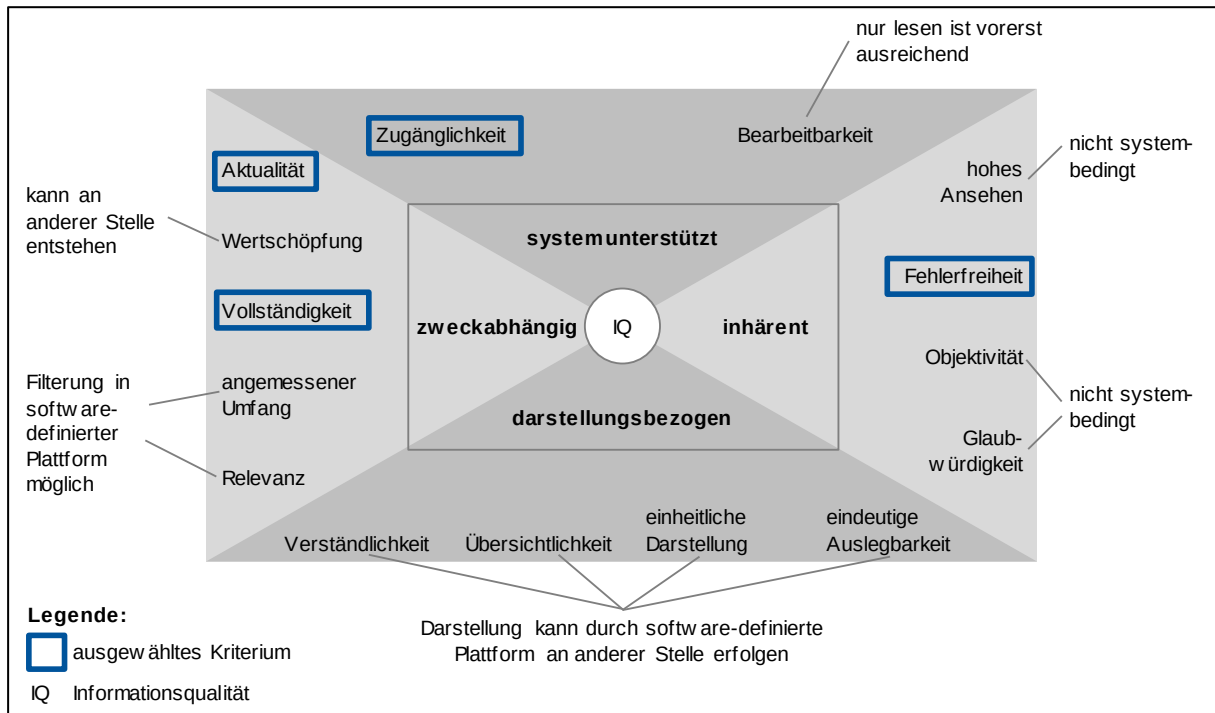


Abbildung 6-11: Dimensionen der Informationsqualität und deren Nutzung für die vorliegende Arbeit (i. A. a. ROHWEDER ET AL. 2015, S. 30)

Als erste Ausprägung von Informationsqualität sei die **Vollständigkeit** beschrieben. Vollständigkeit bezieht sich darauf, inwiefern die erwarteten und durch die Datenstruktur vorgesehenen Informationen im Informationssystem enthalten sind (s. ROHWEDER ET AL. 2015, S. 34; SCHEIBMAYER 2014, S. 114). Die Vollständigkeit beschreibt also die Lückenlosigkeit der Informationen (s. WENKING ET AL. 2017, S. 37). Zusätzlich kann auch die Vollständigkeit der Struktur der enthaltenen Datenpunkte als Bestandteil dieser Ausprägung angesehen werden. Entsprechend der Argumentation von SCHEIBMAYER wird auch hier auf die von ROHWEDER ET AL. postulierte kontextabhängige Definition, die den Aspekt der Pünktlichkeit mit aufnimmt, zugunsten von einer Interdependenzfreiheit der Ausprägungen verzichtet. Auf den zeitlichen Aspekt wird in der Ausprägung *Aktuell* eingegangen.

Ob mit unvollständigen Informationen weitergearbeitet werden kann, hängt von der Fehlermenge ab. Sind Informationen so codiert, dass ausreichend fehlende Bestandteile aus dem Kontext ergänzt werden können, ist dies zwar mit erhöhtem Rechenaufwand verbunden, führt jedoch nicht zu Fehlern.³⁵ Abhängig von den konkreten Anforderungen zur Nutzung der Information ist die Vollständigkeit von Informationen in Informationssystemen außerdem kein zwingend zu erfüllendes Kriterium. Wenn beispielsweise bei einer Ortung von Gegenständen wegen eines Verbindungsabbruchs

³⁵ Verwiesen sei hier auf das Prinzip der Fehlerkorrektur in linearen fehlerkorrigierenden Block-Codes, wie sie beispielsweise derzeit im aktuellen Entwurf des 5G-Standards vorgesehen sind. Der Code basiert auf Polar-Codes (vgl. ARIKAN 2009, S. 3051-3072) und erreicht das theoretische Kapazitätslimit der Fehlerkorrektur. Über geschickte Einfügung von Redundanz kann die Information geschützt werden und trotz Störungen während der Übertragung vollständig wiederhergestellt werden.

eine Ortungsposition fehlt, zu einem späteren Zeitpunkt jedoch wieder vermerkt wird, kann unter Umständen der zugehörige Prozess trotzdem ausgeführt werden.

Informationen in einem Informationssystem sind dann **fehlerfrei**, wenn sie mit der Realität übereinstimmen (s. ROHWEDER ET AL. 2015, S. 37-38; WANG U. STRONG 1996, S. 31). Die Fehlerfreiheit kann sich auch hier auf mehrere Ebenen beziehen: Die syntaktische Fehlerfreiheit bezieht sich beispielsweise auf Rechtschreibfehler oder die Verwendung einer anderen als der spezifizierten Maßeinheit oder Codierung. Syntaktische Fehler können damit sowohl aufgrund von Fehlern bei der Systemkonfiguration, bei der Datenübertragung, als auch bei der Nutzung durch den Anwender entstehen (s. SCHEIBMAYER 2014, S. 112). Einige solcher Fehler können über die Nutzung von Prüfziffern, wie sie aus der *International Bank Account Number* (IBAN)³⁶ bekannt sind, erkannt werden (s. MANZ 2017, S. 20). Nahezu unmöglich ist dagegen das Erkennen von inhaltlichen Fehlern. Um dies zu überprüfen, müssen organisatorische Maßnahmen, wie beispielsweise das Vorhandensein eines Vier-Augen-Prinzips bei kritischen Informationen, überprüft und anhand von Stichproben validiert werden.

Vor allem im Kontext der beschriebenen möglichen Echtzeitfähigkeit zur zeitnahen Entscheidungsfindung ist die Ausprägung der **Aktualität** von Informationen zu betrachten (s. Kapitel 5.3.3 u. 6.2.3). Informationen in einem Informationssystem sind dann aktuell, wenn sie eine zeitnahe Abbildung der von ihnen beschriebenen Objekte sicherstellen (s. ROHWEDER ET AL. 2015, S. 41). Auch hier hängt, entsprechend dem oben vorgestellten Fit-for-use-Prinzip, die Aktualität davon ab, was für die Anwendung der Information benötigt wird. So können beispielsweise auch tagesaktuelle Wetterwerte für eine Korrelation mit dem Einkaufsverhalten ausreichend sein (s. HOFFJAN U. GANTENBRINK 2012, S. 136), andererseits sind für Entscheidungen der Produktionssteuerung Aktualitäten von zumeist unter einer Sekunde notwendig. Die Prüfung, ob Informationen in einem Informationssystem der geforderten Aktualität entsprechen, kann theoretisch aufgrund der Systemarchitektur berechnet, oder anhand von Stichproben aufgenommen werden.

Eine ausschlaggebende Ausprägung für die Integration von Informationssystemen ist die **Zugänglichkeit** der darin enthaltenen Informationen. Eng mit dem Begriff der *Zugänglichkeit* hängt auch der Begriff der *Verfügbarkeit* zusammen, wobei ersterer sich primär auf eine einmalige Aktion bezieht (beispielsweise im Rahmen eines Integrationsprojekts), während letzteres einen andauernden Zustand beschreibt, der einen Zugriff möglichst ohne Wartezeit darstellt (s. SCHEIBMAYER 2014, S. 109). Beide Aspekte sollen für die Beschreibung der Ausprägung Zugänglichkeit in der vorliegenden Arbeit betrachtet werden. Das Merkmal der Informationsqualität, hier einsortiert in die Daten

³⁶ Die IBAN ist nach folgendem Prinzip aufgebaut: 2-stelliger Ländercode, 2-stellige Prüfziffer, maximal 30-stellige Kontoidentifikation (s. MANZ 2017, S. 20). Dabei können maximal zwei Fehler erkannt und ein Fehler korrigiert werden.

betreffenden Merkmale, hat zugleich auch technische und organisatorische Bestandteile (s. STAHL U. STAAB 2017, S. 4-5). Die Zugänglichkeit von Informationen kann durch organisatorisch Verantwortliche für das Informationssystem durch die Vergabe von Zugriffsbeschränkungen oder durch technische Beschränkungen für einen bestimmten Zeitraum eingeschränkt sein. Außerdem hängt die Zugänglichkeit auch mit Informationssicherheitsmaßnahmen zusammen; Informationen sollten nur für einen jeweils bestimmten Nutzerkreis zugänglich sein. Wenn ein Informationssystem jedoch in eine software-definierte Plattform für einen bestimmten Anwendungsfall integriert werden muss, so muss auch die Zugänglichkeit der entsprechenden Informationen, unter Nutzung der entsprechenden Sicherheitsmechanismen, gegeben sein.

Ein Informationssystem, dessen enthaltene Daten keiner der oben genannten Ausprägungen entsprechen, wird die Ausprägung **keine Datenqualität** zugeordnet. Solche Systeme sollten für eine Integration nicht beachtet werden.

6.5.3 Datenstandardisierung

Tabelle 6-13: Ausprägungen des Merkmals *Datenstandardisierung* (eigene Darstellung)

<i>Datenstandardisierung</i>	möglich	nicht möglich

Die Notwendigkeit der Datenstandardisierung wurde in den Kapiteln 5.3.2 und 5.6.5 dargestellt. Für eine solche Standardisierung ist eine Strategie nötig, welche die Verantwortung für die Daten, deren Pflege sowie Abstimmung zwischen verschiedenen Informationssystemen festlegt (s. KRCMAR 2015, S. 179). Weitere Maßnahmen wie die Einführung eines übergeordneten Ordnungssystems oder einer einheitlichen Klassifizierung von Daten werden empfohlen (s. STAHL U. STAAB 2017, S. 45-48; Kapitel 7.1.4).

Einzelne Informationssysteme müssen entsprechende Maßnahmen unterstützen, damit eine Datenintegration gelingen kann. Informationssysteme, die eine **Datenstandardisierung ermöglichen**, haben Funktionen, die eine Standardisierung auf den Ebenen der Zeichen, Daten und Information gestatten, da auf all diesen Ebenen ansonsten Übertragungsfehler entstehen können, welche die Datenqualität mindern. Solche Funktionen können beispielsweise der Zugriff auf Referenzdatensysteme oder Ähnlichkeitssuchen zur Stammdatenbereinigung sein (s. BLEIHOLDER U. SCHMID 2015, S. 138-139). Auch müssen Informationssysteme die Transformation operativer Daten ermöglichen, indem beispielsweise Maßeinheiten angeglichen, oder – auf Ebene der Zeichen – Zeichensätze standardisiert werden können (s. KEMPER U. FINGER 2016, S. 129-132).³⁷ Eine Möglichkeit einer solchen Standardisierung kann beispielsweise

³⁷ Über Herausforderungen, die bei nicht möglicher Standardisierung von Zeichensätzen entstehen, schreibt der Autor zusammen mit anderen Autoren an anderer Stelle (s. STICH ET AL. 2018, S. 39).

auch die Angleichung von Artikelnummern in verschiedenen Kernsystemen des Unternehmens sein. So hat die e.GO Mobile AG (s. Kapitel 9.2) beispielsweise eine globale Identifikationsnummer für Artikel in ihrem ERP- und PLM-System eingeführt, das auch für nachfolgende Prozesse wie dem Service genutzt wird.

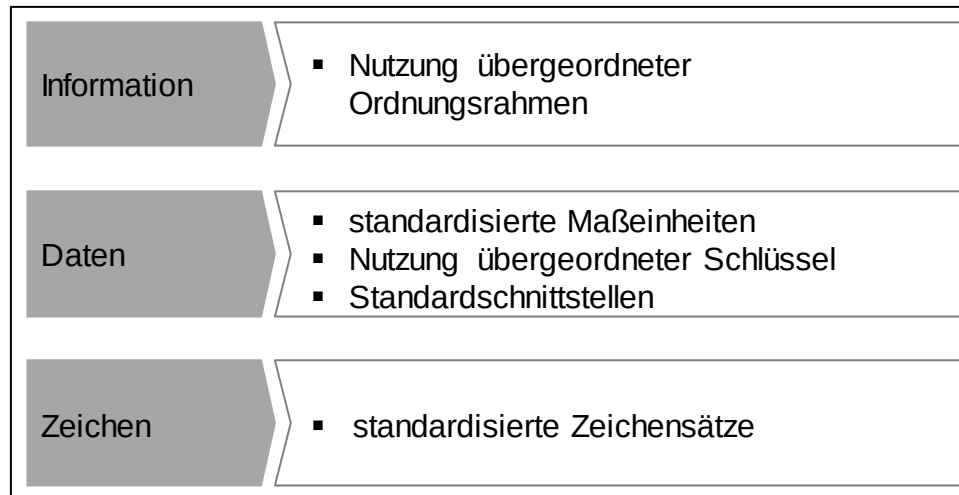


Abbildung 6-12: Möglichkeiten der Standardisierung auf den Ebenen Zeichen, Daten und Informationen (eigene Darstellung)

Ist eine solche Datenstandardisierung in einem Informationssystem **nicht möglich**, so müssen Daten mit erhöhtem Aufwand an anderer Stelle überarbeitet werden. Dies kann unter Umständen nicht ohne weiteres möglich sein,³⁸ weshalb eine Integration in dem Fall deutlich aufwendiger wird.

6.6 Zusammenfassung der beschreibenden Merkmale

Die in den vorhergehenden Kapiteln 6.2 bis 6.5 genannten Merkmale dienen dazu, Informationssysteme hinsichtlich ihrer Einordnung in Informationssystem-Architekturen beschreibbar zu machen und damit zu systematisieren. Sie sind außerdem spezifisch derart beschrieben, dass Integrationsherausforderungen dargestellt werden. Tabelle 6-14 gibt gemäß der Forderung von WELTER zur morphologischen Methode (s. WELTER 2006, S. 115-116) einen Überblick über die ausgewählten Merkmale und ihre Ausprägungen.

³⁸ Ein Beispiel eines produzierenden Unternehmens: Zur Finanzkonsolidierung muss das als Gruppe geführte Unternehmen die Finanzkennzahlen seiner Tochterunternehmen zusammenfassen. Da diese teilweise unterschiedliche Kontenrahmen und unterschiedliche ERP-Systeme benutzen, die nicht alle den Gruppenstandard unterstützen, ist die Datenintegration in dem Fall nur mit hohem manuellem Aufwand möglich.

Tabelle 6-14: Merkmale von Informationssystemen (entnommen aus HOFFMANN U. HEIMES 2018, S. 991)

Merkmal		Ausprägungen								
Technische Merkmale	<i>Schnittstellen</i>	generisch	vordefiniert							
	<i>Outsourcinggrad</i>	interner Betrieb	Infrastruktur-Outsourcing	Anwendungssystem-Outsourcing	Businessprozess-Outsourcing					
	<i>Echtzeitfähigkeit</i>	keine Echtzeit	weiche Echtzeit	harte Echtzeit						
	<i>Status des Softwarelebenszyklus</i>	initiale Software	Evolution u. Servicing	Ausphasung						
Organisatorische Merkmale	<i>Systemursprung</i>	Individu-alsystem	angepasstes Standardsystem	Standardsystem						
	<i>Systemverantwortung</i>	IT-Abteilung	Fachbereich	Produktions-IT	Extern					
	<i>Organisationseinheit</i>	Entwicklung	Einkauf	Produktion	Vertrieb	Verwaltung u. Management	übergreifend			
	<i>Nutzerkreis</i>	Einzelpersonen	Teilmenge des Unternehmens	unternehmensübergreifend						
Prozessuale Merkmale	<i>Systemkonzept</i>	PLM	ERP	MES	CPS / OT	SCM	CRM	BI	ECM	Sonstige
	<i>Kernprozess</i>	Marktanalyse	Konzept	Entwurf u. Detaillierung	Produktionsplanung	Herstellung	Vertrieb	Nutzung u. Service	Recycling	Unterstützungsprozess
Daten betreffende Merkmale	<i>betroffene Ebenen</i>	Datenhaltung u. -erzeugung	Vernetzung	Applikation	Visualisierung	Business u. Prozesse				
	<i>Informationsqualität</i>	vollständig	fehlerfrei	aktuell	zugänglich	keine Informationsqualität				
	<i>Datenstandardisierung</i>	möglich	nicht möglich							

Die Ausprägungen der Merkmale sind, soweit möglich, ordinalskaliert. Teilweise, wenn eine solche Skalierung fachlich nicht sinnvoll war, wurden die Ausprägungen anhand des Software- beziehungsweise des Produktlebenszyklus sortiert (Merkmale *Status des Softwarelebenszyklus*, *Organisationseinheit*, *Systemkonzept* und *Kernprozess*). Allein bei den Merkmalen *Schnittstellen* und *Informationsqualität* konnten die Ausprägungen nicht anhand einer Skala sortiert werden. Im Folgenden werden die erarbeiteten Merkmale genutzt, um Informationssystem-Architekturen darzustellen.

6.7 Darstellung von Informationssystem-Architekturen

Der Architekturbegriff und seine Anwendung auf Informationssysteme wurden in Kapitel 2.1.5 definiert und vorgestellt. Demnach bildet die Anordnung und die Verbindung von Informationssystemen die Informationssystem-Architektur. Um eine **solche Anordnung verständlich zu machen, muss sie visualisiert werden**. Wollte man Informationssysteme anhand aller im vorherigen Kapitel herausgearbeiteten Merkmale in einer Darstellung einordnen, entstünde dabei eine Darstellung in einem 13-dimensionalen Raum. Da ein solcher nicht sinnvoll zu visualisieren ist, muss die Informationssystem-Architektur in mehreren **unterschiedlichen Diagrammen mit jeweils anderen Dimensionen** gezeigt werden. Ein solches Vorgehen wird von verschiedenen Autoren empfohlen. Entsprechend dem Ziel, das mit einer Darstellung verfolgt wird, sowie den Fragestellungen, die damit beantwortet werden sollen, sollen die Visualisierungen gestaltet werden (s. TIEMEYER 2017a, S. 111; HANSCHKE 2016, S. 58). Unübersichtliche Darstellungen sind zu vermeiden, obschon für die reine Informationssammlung Listen (s. Tabelle 6-15) und textuelle Beschreibungen unverzichtbar sind (s. KRCMAR 2015, S. 101; TIEMEYER 2017a, S. 111).

Tabelle 6-15: Ausschnitt einer Liste an Informationssystemen eines produzierenden Unternehmens (eigene Darstellung, Vorlage für nachfolgende durchgängige Beispiele)

Informationssystem	Schnittstellen	Outsourcinggrad	Echtzeitfähigkeit	Status des Softwarelebenszyklus	Systemursprung	Systemverantwortung	Organisationseinheit	Nutzerkreis	Systemkonzept	Kernprozess	Betroffene Ebenen	Informationsqualität	Datenstandardisierung
DATEV	vordefiniert	Anwendungssystem Outsourcing	keine Echtzeit	Evolution u. Servicing	Standardsystem	Fachbereich	Verwaltung u. Management	Teilmenge des Unternehmens	ERP / Sonstige	Unterstützungsprozess	Business u. Prozesse / Visualisierung / Applikation	zugänglich	nicht möglich
E-Mail (Exchange)	generisch	interner Betrieb	keine Echtzeit	Evolution u. Servicing	Standardsystem	IT-Abteilung	übergreifend	Unternehmensübergreifend	Sonstige	[Alle]	Business u. Prozesse / Visualisierung / Applikation / Vernetzung	aktuell / zugänglich	nicht möglich
Git	generisch	interner Betrieb	weiche Echtzeit	Evolution u. Servicing	Standardsystem	Fachbereich	Entwicklung	Teilmenge des Unternehmens	Sonstige	Konzept / Entwurf u. Detaillierung / Herstellung	Datenhaltung u. -erzeugung / Vernetzung	zugänglich	möglich
orgaMAX	vordefiniert	interner Betrieb	keine Echtzeit	Ausphasung	angepasstes Standardsystem	Fachbereich	Verwaltung u. Management	Einzelpersonen	ERP	Vertrieb	Applikation	keine Informationsqualität	nicht möglich
Echtzeitautomatisierungssystem	vordefiniert	interner Betrieb	Harte Echtzeit	Evolution u. Servicing	Individualsystem	Produktions-IT	Produktion	Teilmenge des Unternehmens	MES	Herstellung	Datenhaltung u. -erzeugung / Vernetzung / Applikation	vollständig/ aktuell / zugänglich	möglich
Produktionsstatus-DB (Access)	vordefiniert	interner Betrieb	keine Echtzeit	Evolution u. Servicing	Individualsystem	Produktions-IT	Produktion	Einzelpersonen	ERP	Unterstützungsprozess / Herstellung	Datenhaltung u. -erzeugung / Vernetzung / Applikation	vollständig	nicht möglich
Sage ERP	vordefiniert	interner Betrieb	weiche Echtzeit	Ausphasung	angepasstes Standardsystem	Fachbereich	Übergreifend	Unternehmensübergreifend	ERP	[Alle außer Marktanalyse, Konzept, Recycling]	Datenhaltung u. -erzeugung / Vernetzung / Applikation	keine Informationsqualität	nicht möglich
SharePoint	generisch	interner Betrieb	keine Echtzeit	Ausphasung	Standardsystem	IT-Abteilung	Vertrieb	Teilmenge des Unternehmens	Sonstige	Vertrieb	Business u. Prozesse / Visualisierung	keine Informationsqualität	möglich
SolidWorks	generisch	interner Betrieb	keine Echtzeit	Evolution u. Servicing	Standardsystem	Fachbereich	Entwicklung	Teilmenge des Unternehmens	PLM	Konzept / Entwurf u. Detaillierung	Applikation / Visualisierung	vollständig/ aktuell / zugänglich	möglich
SolidWorks PDM	generisch	interner Betrieb	weiche Echtzeit	Evolution u. Servicing	Standardsystem	Fachbereich	Entwicklung	Teilmenge des Unternehmens	PLM	Konzept / Entwurf u. Detaillierung	Datenhaltung u. -erzeugung / Vernetzung	vollständig/ zugänglich	möglich
Webseite	vordefiniert	Anwendungssystem Outsourcing	keine Echtzeit	Evolution u. Servicing	angepasstes Standardsystem	Extern	Vertrieb	Einzelpersonen	Sonstige	Vertrieb / Nutzung und Service	Visualisierung	zugänglich/ aktuell	nicht möglich

Die folgenden **Abbildungen sind generisch nutzbar**. Sie werden allerdings für die vorliegende Arbeit **anhand eines Beispiels instanziiert**, damit sie praxisnäher und

einfacher zu verstehen sind. Die in Tabelle 6-15 aufgelisteten und in den Abbildungen gezeigten Informationssysteme sind jedoch durch jede andere Software eines anderen Herstellers austauschbar. Tabelle 6-15 zeigt anonymisiert einen Ausschnitt einer Informationssystem-Architektur eines produzierenden Unternehmens. Es handelt sich dabei um ein Unternehmen des Typs I (s. Kapitel 6.8.1).

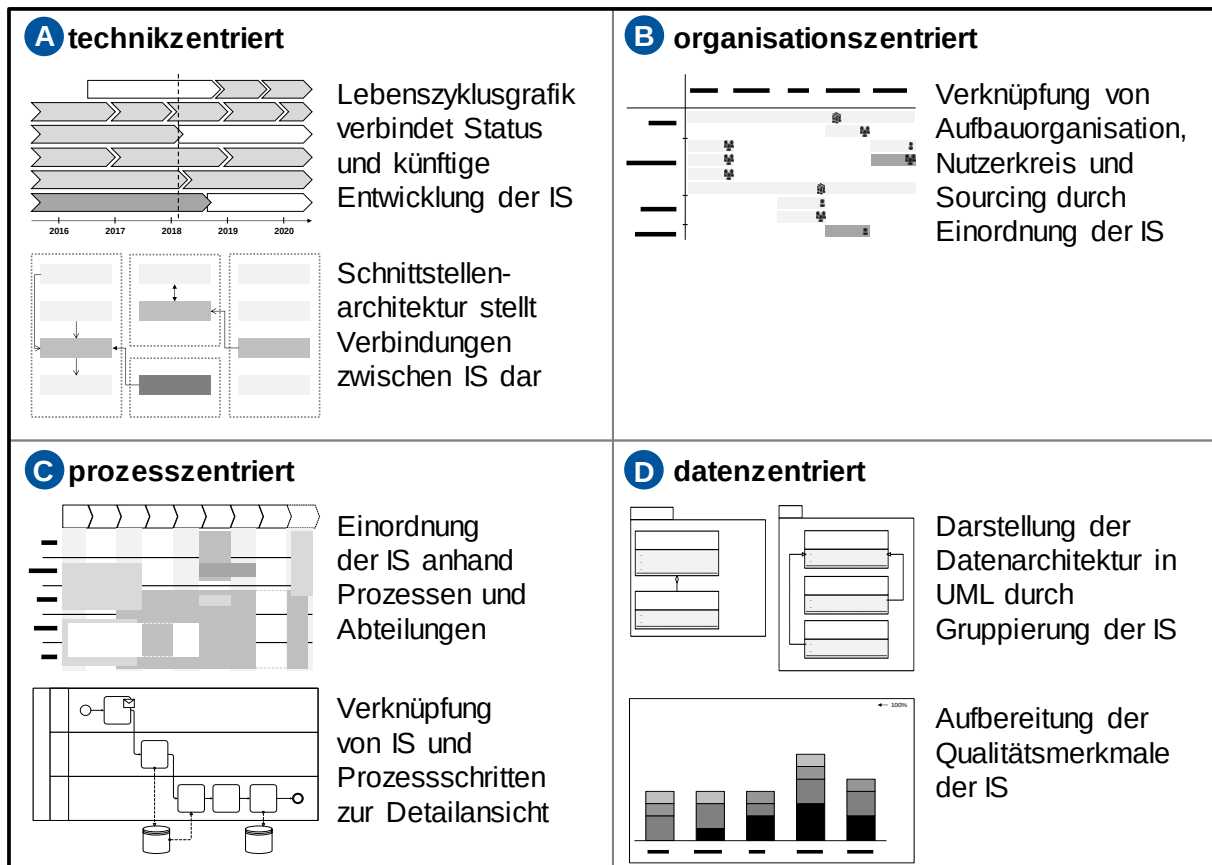


Abbildung 6-13: Übersicht der Darstellungsvarianten von Informationssystem-Architekturen (entnommen aus HOFFMANN U. HEIMES 2018, S. 995)

Die Literatur gibt eine Vielzahl von Vorschlägen und Modellen zu Informationssystem-Architekturen und deren Visualisierung, vor allem im Kontext von EAM. In den einführenden Kapiteln 2.1.5, 2.1.6 und 3.3 wurden die zentralen Konzepte im Hinblick auf produzierende Unternehmen vorgestellt. Davon abgeleitet wurden für die vorliegende Arbeit eigene **Darstellungsformen mit dem Ziel entwickelt, die verschiedenen Integrationsfragestellungen software-definierter Plattformen beantworten zu können**. Weiterhin verfolgen die Darstellungsformen allgemeine, in der Literatur vorgeschlagene Ziele: Entsprechend dem „*Mile-Wide-Inch-Deep-Prinzip*“ von KEUNTJE sollen nicht unbedingt alle Details über Informationssysteme in einer Darstellung aufgenommen werden, dafür aber die ganze Breite an Informationssystemen dargestellt werden (s. KEUNTJE 2010, S. 176; DERN 2009, S. 49). Informationssystem-Architekturen sollen den Gesamtzusammenhang darstellen, um langfristige Ausrichtungen zu entscheiden (s. ALPAR ET AL. 2016, S. 172). Weiterhin sollen die Darstellungen nach HANSCHKE spezifisch auf die Anforderungen von verschiedenen Stakeholdern wie Unternehmensleitung oder die operative IT-Abteilung ausgerichtet sein, um damit adäquat und zielgruppengerecht kommunizieren zu können (s. HANSCHKE 2016, S. 58).

Aus eigener Erfahrung des Autors müssen sie zudem so gewählt werden, dass sie eine *konkrete* Aussage hinsichtlich aktueller Herausforderungen und künftiger Entwicklungen treffen lassen.

Die für die vorliegende Arbeit entwickelten Darstellungsformen werden im Folgenden, geordnet nach ihrem primären Darstellungszweck, entlang der vier Dimensionen, in denen die Merkmale von Informationssystemen einsortiert sind, vorgestellt (s. Abbildung 6-13). Dabei werden nicht nur Merkmale *einer* Dimension miteinander kombiniert, sondern teilweise auch übergreifend eingesetzt. Weiterhin sind die Darstellungen als Beispiele zu verstehen; weitere Kombinationen von Merkmalen sind möglich und sollten entsprechend der oben genannten Ziele in der praktischen Anwendung eingesetzt werden. Die Darstellungen sind so gewählt, dass jede Dimension mindestens in eine Darstellung einfließt.

6.7.1 Darstellungen mit Fokus auf technische Merkmale

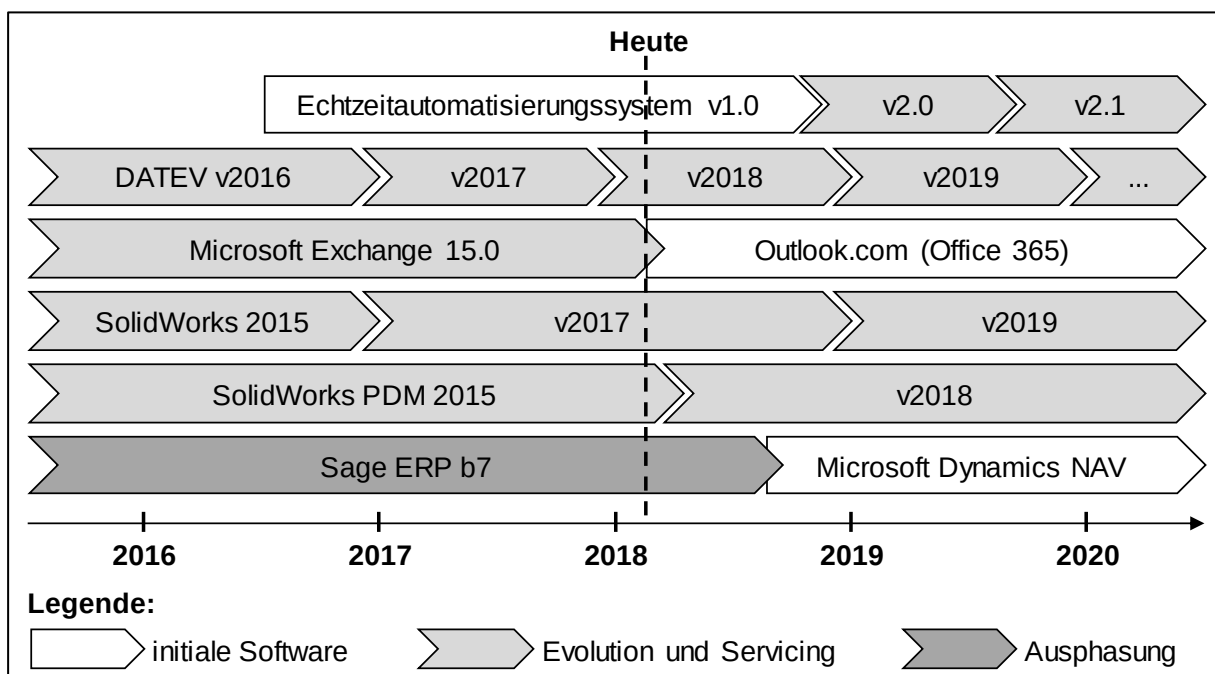


Abbildung 6-14: Ausschnitt einer Informationssystem-Architektur (Lebenszyklusgrafik) mit Fokus auf den Softwarelebenszyklus (eigene Darstellung)

Die erste Darstellungsform mit Fokus auf technische Merkmale ist die **Lebenszyklusgrafik**, wie sie, angewendet auf die Informationssysteme der Tabelle 6-15, beispielhaft in Abbildung 6-14 gezeigt ist. Die Informationssysteme sind dabei mit den jeweils aktuellen Versionsständen sowie der Planung der Einführung zukünftiger Versionsstände **auf einer Zeitachse verortet**. Zusätzlich sind sie entsprechend ihrem Status im Softwarelebenszyklus (s. Kapitel 6.2.4) farblich hinterlegt. Solche Darstellungen sind primär für die Planung der Weiterentwicklung der Informationssystem-Architektur auf einer technischen Ebene relevant. Sollten sich Abhängigkeiten zwischen einzelnen Informationssystemen, beziehungsweise von deren Versionsständen, ergeben, so können sie hier visualisiert werden. Wenn solche Darstellungen noch weiter detailliert

werden und beispielsweise auch Middlewarekomponenten oder Betriebssysteme enthalten, so können damit auch Migrationsbedarfe hergeleitet werden (s. HANSCHKE 2016, S. 87).³⁹ Weiterhin gibt eine solche Darstellung eine Gesamtübersicht über den technischen Zustand der Informationssystem-Architektur; es lassen sich somit erste Schlüsse über die eingesetzten Technologien und deren Integrationsfähigkeit ziehen (s. DERN 2009, S. 73).

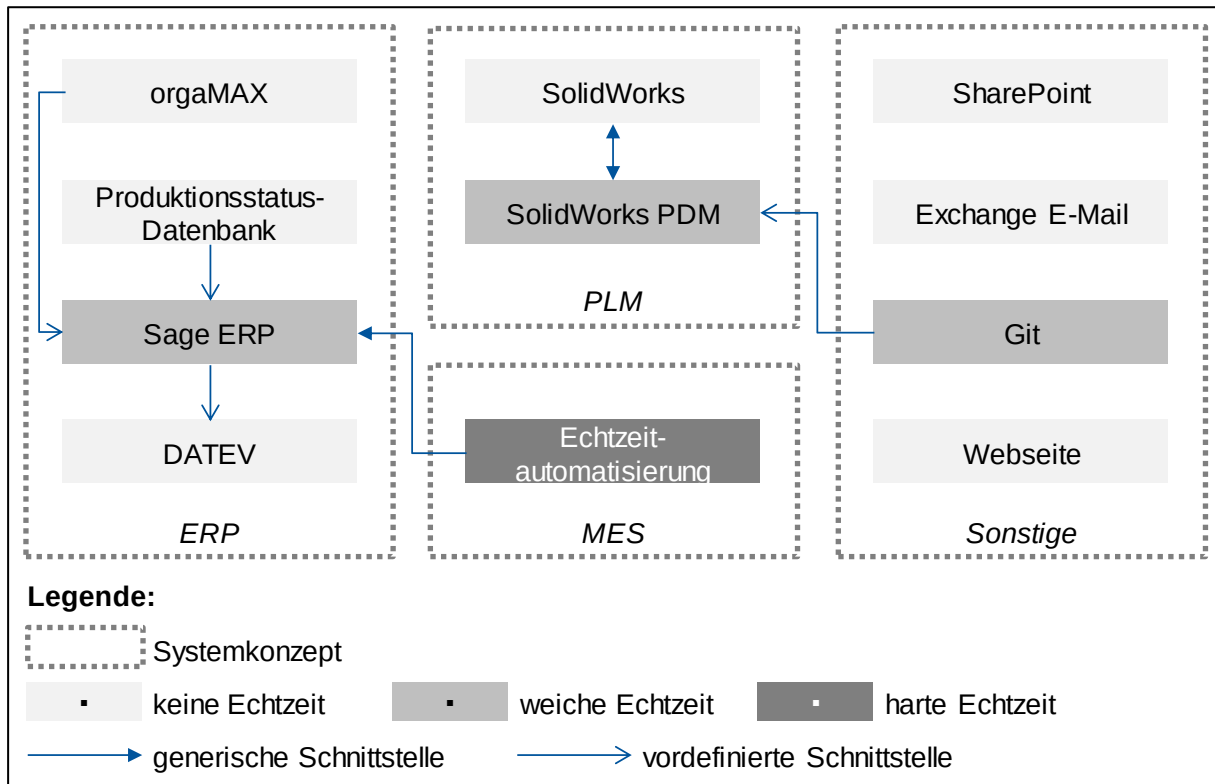


Abbildung 6-15: Ausschnitt einer Informationssystem-Architektur mit Fokus auf Schnittstellen und Echtzeitfähigkeit (eigene Darstellung)

Die Darstellung von **Schnittstellenarchitekturen**, wie sie beispielhaft Abbildung 6-15 zeigt, dient dazu, die **Kommunikation zwischen einzelnen Informationssystemen** transparent zu machen. Diese auch als Anwendungslandkarten bezeichneten Diagramme (s. TIEMEYER 2017a, S. 112) zeigen außerdem **Interdependenzen**, die bei einer möglichen Ablösung eines Informationssystems zu beachten sind. Eine solche Darstellung zielt außerdem darauf ab, eine Übersicht zu geben, welche Systeme in welchem Grad Daten in Echtzeit liefern können. Sie geben damit einen ersten Überblick darüber, welche Informationssysteme direkt mit einer software-definierten Plattform verbunden werden müssen, und wo gegebenenfalls noch Änderungsbedarf von

³⁹ Beispielsweise könnte in einem Unternehmen die Migration der Client-Betriebssysteme von *Windows 7* auf *Windows 10* geplant sein. Die aktuell eingesetzte Version der CAD-Anwendung unterstützt allerdings *Windows 10* nicht. Um in diesem Kontext eine Entscheidung zu treffen, ist unter anderem der Softwarelebenszyklus des CAD-Systems relevant. Befindet er sich in der Phase der „Ausphasung“, kann beispielsweise bis zur kompletten Außerbetriebnahme ein PC mit *Windows 7* behalten werden. Soll das CAD-System weiter genutzt werden, ist ein Upgrade zu planen.

Systemen oder Schnittstellen besteht, deren Daten in Echtzeit in der Plattform verarbeitet werden sollten (s. Kapitel 6.2.3). In der gezeigten Beispielabbildung wurden die Informationssysteme außerdem noch anhand des Merkmals *Systemkonzept* geordnet (s. Kapitel 6.4.1). Eine solche Vorgehensweise gibt im Speziellen bei umfangreichen Architekturen Struktur, hilft aber auch dabei, zu verstehen, inwieweit ähnliche Informationssysteme, die prinzipiell auch ähnliche Datenbestände enthalten, miteinander vernetzt sind. Zudem lassen sich so auch Datenquellen und Datensinken transparent machen und fehlende Datenübertragungen aufzeigen. Schnittstellenarchitekturen sind eng mit Datenarchitekturen verwandt (s. TIEMEYER 2017a, S. 117-118); letztere zielen jedoch auf eine schematische Darstellung und abstrahieren dabei von den tatsächlichen Informationssystemen (s. Kapitel 6.7.4).

Zur Erstellung einer solchen Darstellung ist, neben der Erhebung der Einzelsysteme, auch eine übergreifende Betrachtung notwendig, in welcher der Weg *zwischen* verschiedenen Informationssystemen analysiert wird. Direkte Schnittstellen (s. Kapitel 6.2.1) können dabei oft relativ leicht aufgespürt werden, indirekte Schnittstellen benötigen oft intensive Recherche. Dies liegt darin begründet, dass es sich dabei um manuell angesteuerte Datenübergaben handelt, die erst durch Befragung der richtigen Anwender erkannt werden können.

Schnittstellenarchitektur-Grafiken können als Informationsflussgrafiken noch detaillierter dargestellt werden (s. NIENKE ET AL., S. 242-243). Dazu werden erstens die Daten- und Informationsobjekte, die über eine Schnittstelle ausgetauscht werden, an die gerichteten Verbindungspfeile geschrieben. Zweitens wird die Grafik insofern verfeinert, als dass einzelne Module oder Dienste von Informationssystemen separat dargestellt werden, um Informationsflüsse noch detaillierter nachverfolgen zu können (s. HANSCHKE 2016, S. 84-85).

6.7.2 Darstellungen mit Fokus auf organisatorische Merkmale

Zur reinen Darstellung von Organisationsformen, auch innerhalb von IT-Abteilungen, existiert vielfältige Fachliteratur, auf die an dieser Stelle verwiesen sei (vgl. bspw. LEHNER ET AL. 2008, S. 241-246; HEINRICH ET AL. 2014, S. 296-306). Im vorliegenden Abschnitt soll vielmehr die **Einordnung von Informationssystemen anhand organisatorischer Merkmale** diskutiert werden. Abbildung 6-16 gibt ein Beispiel der Darstellung einer solchen Einordnung. In Form einer sogenannten **Zuordnungstabelle** (s. HANSCHKE 2016, S. 86) werden Informationssysteme hier anhand insgesamt vier Dimensionen dargestellt. Zentral ist dabei die Einordnung der Informationssysteme in eine Matrix mit den Dimensionen *Organisationseinheit* und *Systemverantwortung*. Im dargestellten Beispiel zeigt sich beispielsweise, dass die IT-Abteilung die übergreifend genutzten Office-Systeme verantwortet, während alle anderen Informationssysteme in der Hand derjenigen Fachbereiche sind, welche sie hauptsächlich nutzen. Aus einer solchen Darstellung lässt sich herauslesen, wie stark die Rolle einer IT-Abteilung im Unternehmen ist, und damit auch, welche Möglichkeiten die IT-Abteilung hinsichtlich der Standardisierung von Informationssystemen hat (vgl. dazu Kapitel 2.2.4).

Als zusätzliche Informationen werden hier angegeben, inwieweit das jeweilige Informationssystem intern oder durch einen Dienstleister betrieben wird (s. Kapitel 6.2.2) sowie die Größe des Nutzerkreises (s. Kapitel 6.3.4).

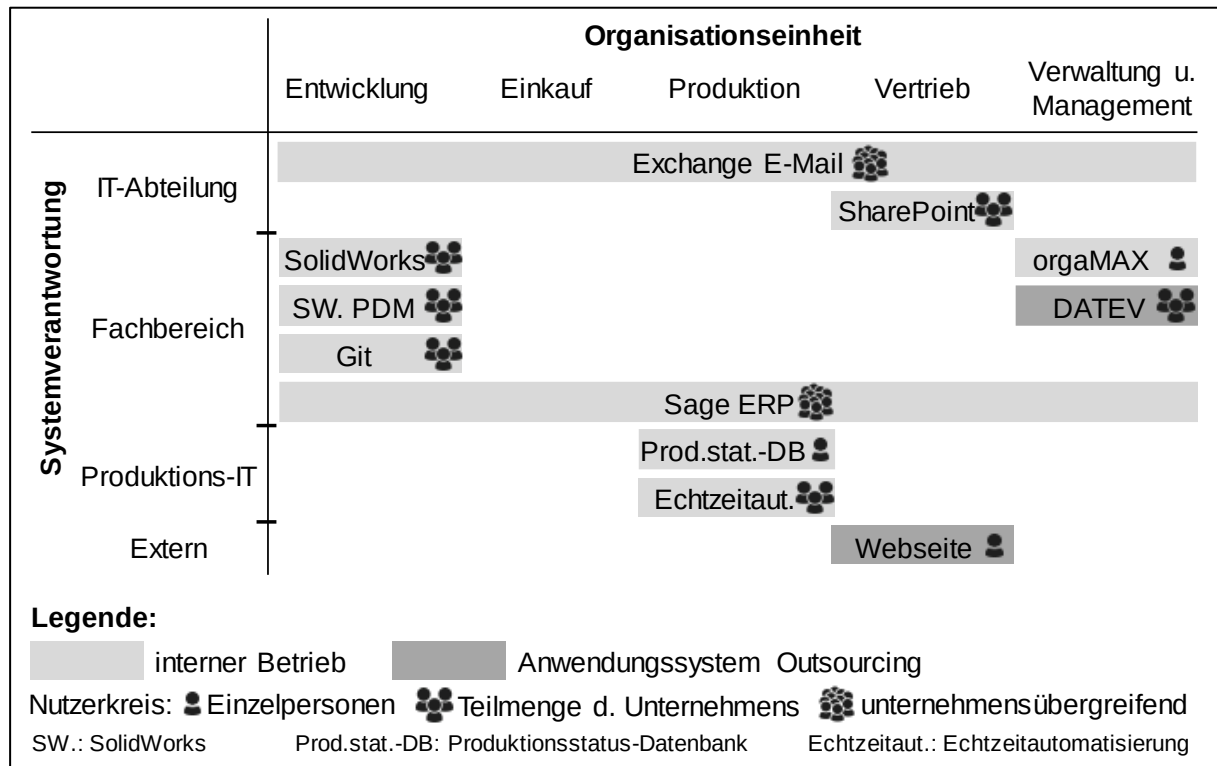


Abbildung 6-16: Ausschnitt einer Informationssystem-Architektur mit Fokus auf die organisatorische Einordnung (eigene Darstellung)

Wie hier gezeigt, können Zuordnungstabellen für eine Vielzahl weiterer Gegenüberstellungen genutzt werden, um Auswertungen der Informationssystem-Architektur zu visualisieren. Die im vorliegenden Abschnitt genannten vier Dimensionen lassen beispielsweise in nahezu jeder Konstellation sinnvolle Aussagen zu:

- Die Einordnung von Informationssystemen in die Achsen *Nutzerkreis* und *Organisationseinheit* zeigt an, wie stark Abteilungen auf gemeinsame Systeme setzen.
- Ein Diagramm mit den Achsen *Sourcing* und *Systemverantwortung* gibt Hinweise darauf, wie stark beispielsweise Fachbereiche Informationssysteme komplett ohne die Einbindung einer IT-Abteilung betreiben.

6.7.3 Darstellungen mit Fokus auf prozessuale Merkmale

Die Darstellung, die sowohl in der Literatur die meisten Nennungen erfährt, wie auch in den Projekten des FIR am häufigsten eingesetzt wird, ist die **Bebauungsplangrafik**, eine Form des Matrix-Diagramms (s. HANSCHKE 2016, S. 75-77; DURST 2007, S. 39; DERN 2009, S. 40; KEUNTJE 2010, S. 166-167; TIEMEYER 2017a, S. 112).

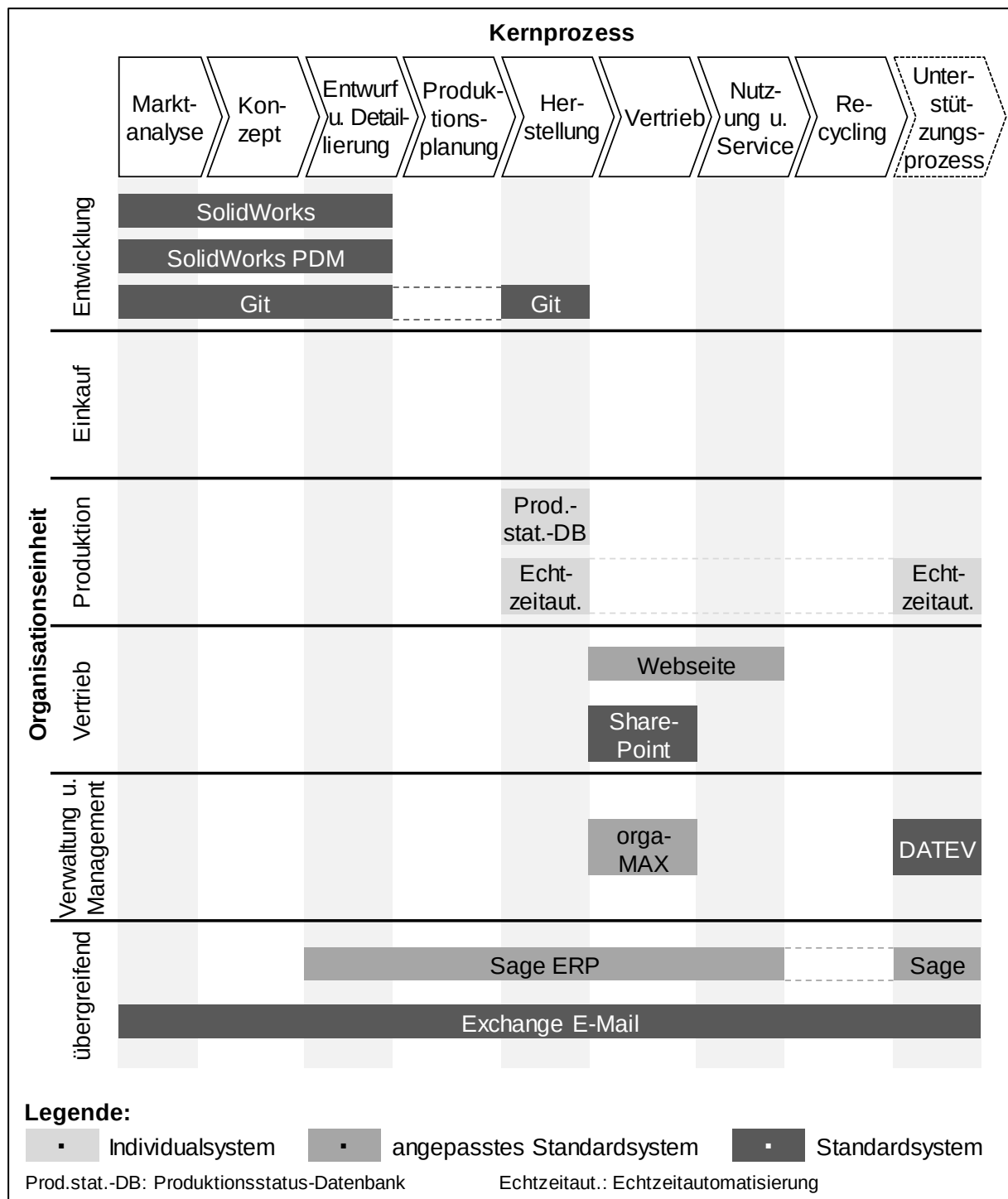


Abbildung 6-17: Ausschnitt einer Informationssystem-Architektur mit Einordnung der Informationssysteme gemäß Kernprozessunterstützung und Organisationseinheit, zusätzliche Hervorhebung des Systemursprungs (eigene Darstellung)

In der Bebauungsplangrafik in Abbildung 6-17 werden Informationssysteme entsprechend ihrer Nutzung in einem Kernprozess durch Mitarbeiter einer bestimmten Organisationseinheit verortet. Damit werden die Fähigkeiten der Informationssysteme, einen bestimmten Prozess zu unterstützen, in Verbindung mit der tatsächlichen Nutzung dieser Fähigkeiten gezeigt (s. KEUNTJE 2010, S. 166). Zusätzlich wird durch die Hervorhebung dargestellt, ob es sich um ein Individual- oder ein Standardsystem handelt.

Auch in dieser Form der Darstellung sind verschiedene Kombinationen und Hervorhebungen denkbar; beispielsweise kann durch eine Hervorhebung der Informationssysteme, die nicht von der IT-Abteilung betreut werden, die Nutzung von Schatten-IT visualisiert werden (s. dazu Kapitel 2.2.4).

Das Ziel von Bebauungsplangrafiken ist es, **die IT-Unterstützung von Unternehmensprozessen** (oder gegebenenfalls auch Produkten, Kundengruppen etc.) **darzustellen**. Dabei wird sowohl gezeigt, wo Redundanzen in der Unterstützung vorhanden sind, als auch, wo dort Lücken auftreten, um daraus Handlungsbedarfe abzuleiten (s. KEUNTJE 2010, S. 168). Damit ist eine Bebauungsplangrafik ein wichtiges Managementinstrument, das zudem die Kommunikation zu verschiedenen Stakeholdern vereinfacht. Neben der Darstellung des Ist-Zustands werden Bebauungsplangrafiken oft auch zur Darstellung eines Ziel-Zustands verwendet und dienen dann als Zielbild, das handlungsleitend für die beteiligten Akteure ist (s. KEUNTJE 2010, S. 170; HANSCHKE 2016, S. 79).

Je nach Anwendungsbereich und Umfang der zu visualisierenden Daten ist es zweckmäßig, Bebauungspläne in verschiedenen Detaillierungsstufen zu erstellen. Ein übergeordneter Plan könnte beispielsweise nur übergreifend genutzte Informationssysteme in einer Einordnung von Kernprozess und Geschäftsbereichen aufzeigen, während auf höchster Detaillierungsstufe nur die Informationssysteme einer einzelnen Abteilung dargestellt werden.

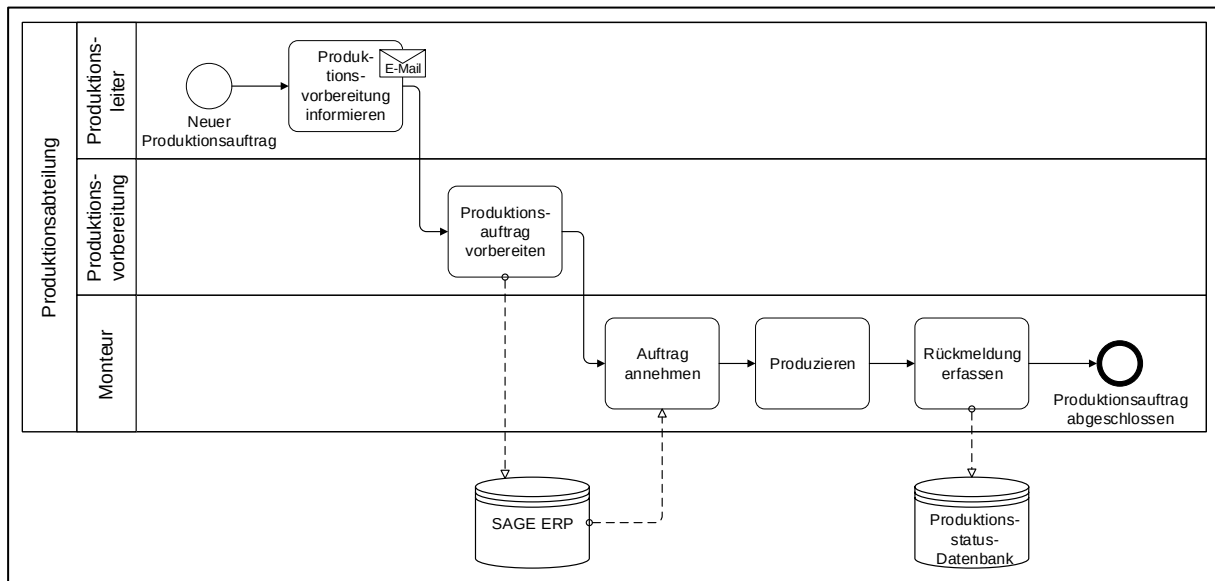


Abbildung 6-18: Detaillierte Aufnahme von Informationssystem-Architekturen mit BPMN 2.0 am vereinfachten Beispiel eines Produktionsprozesses (eigene Darstellung)

Zur **Detaillierung** können außerdem **Prozessdarstellungen** genutzt werden, wie sie in Abbildung 6-18 gezeigt sind. Zu deren Notation existieren verschiedene Standards wie die in der Abbildung verwendete *Business Process Modelling Language* (BPMN) oder *Ereignisgesteuerte Prozessketten* (EPK) (s. REISS U. REISS 2016, S. 196). Zu deren Anwendung sei auch hier auf entsprechende Fachliteratur verwiesen (vgl. REISS U. REISS 2016, S. 193-204; SCHEER 1998; ALLWEYER 2015). Wichtig hierbei ist, dass neben der Darstellung von Aufgaben und Entscheidungen auch Informationsobjekte

und Informationssysteme modelliert werden. So können Systemeinsatz und Systemnutzung detailliert nachvollzogen werden. Vor allem bei der Einführung von neuen Informationssystemen sollten für die Kernprozesse des Unternehmens solche Prozessaufnahmen durchgeführt werden. Andernfalls werden möglicherweise wichtige Teile der täglichen Aufgaben der Mitarbeiter nicht unterstützt, oder es kommt zu umfangreichen Nacharbeiten nach erfolgter Systemeinführung.

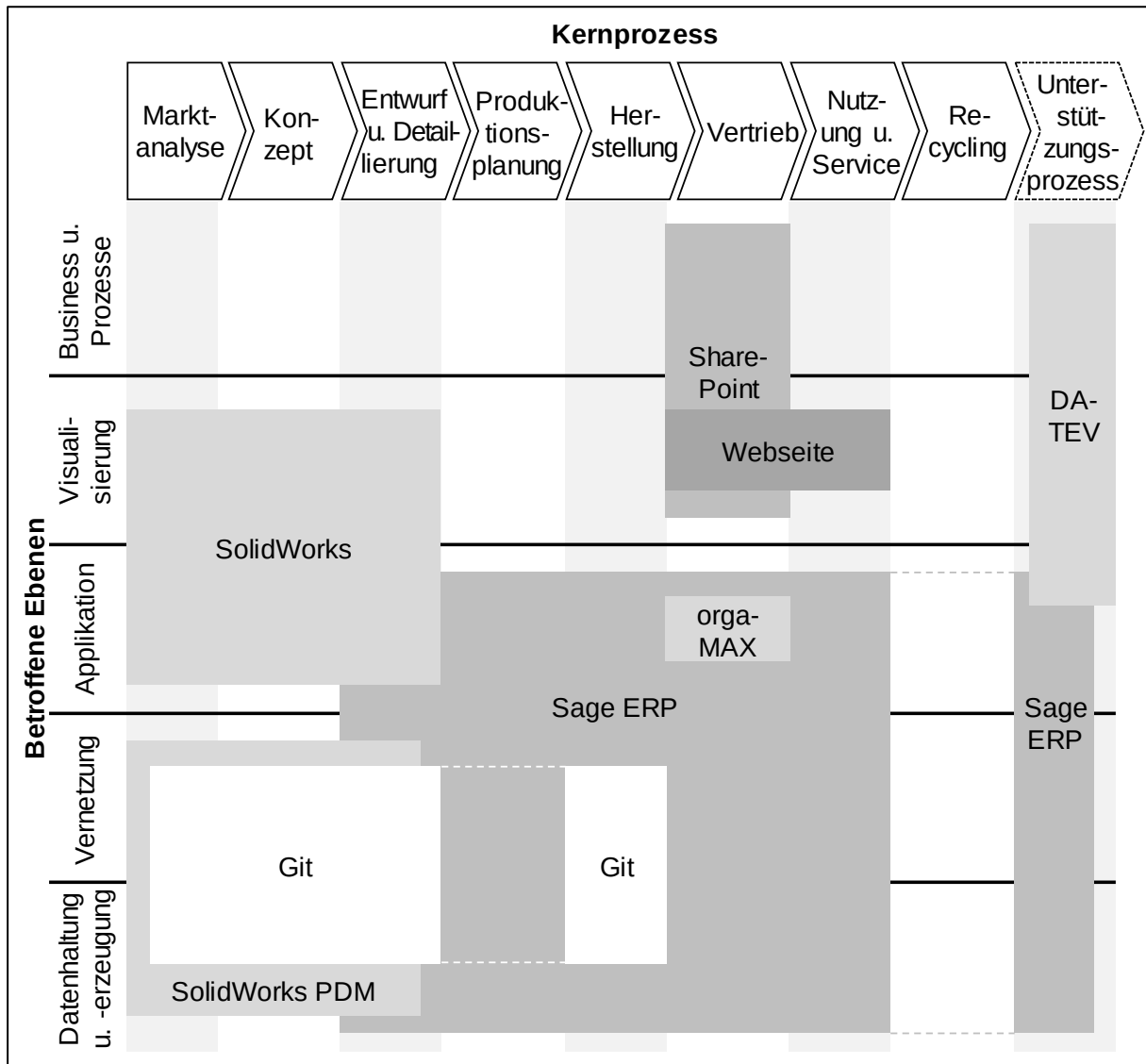


Abbildung 6-19: Ausschnitt einer Informationssystem-Architektur mit Einordnung der Informationssysteme gemäß Kernprozessunterstützung und hauptsächlicher Ebenenzuordnung (eigene Darstellung)

Eine bereits in den Einführungskapiteln dieser Arbeit verwendete Darstellungsform wird in Abbildung 6-19 gezeigt (s. Kapitel 1.2). Dort werden **Informationssysteme entsprechend den betroffenen Ebenen und dem unterstützten Kernprozess eingeordnet**. Für die Einführung von einer software-definierten Plattform in eine vorhandene Informationssystem-Architektur ist eine solche Darstellung deshalb von zentraler Bedeutung, weil die Informationssysteme hervorgehoben werden, die selbst Daten

halten oder in besonderer Weise verarbeiten. Nur solche Informationssysteme müssen, entsprechend den in Kapitel 5.1.1 definierten Zielen des Plattformeinsatzes, unter anderem die verbesserte Datentransparenz, an die Plattform angebunden werden.

6.7.4 Darstellungen mit Fokus auf Daten betreffende Merkmale

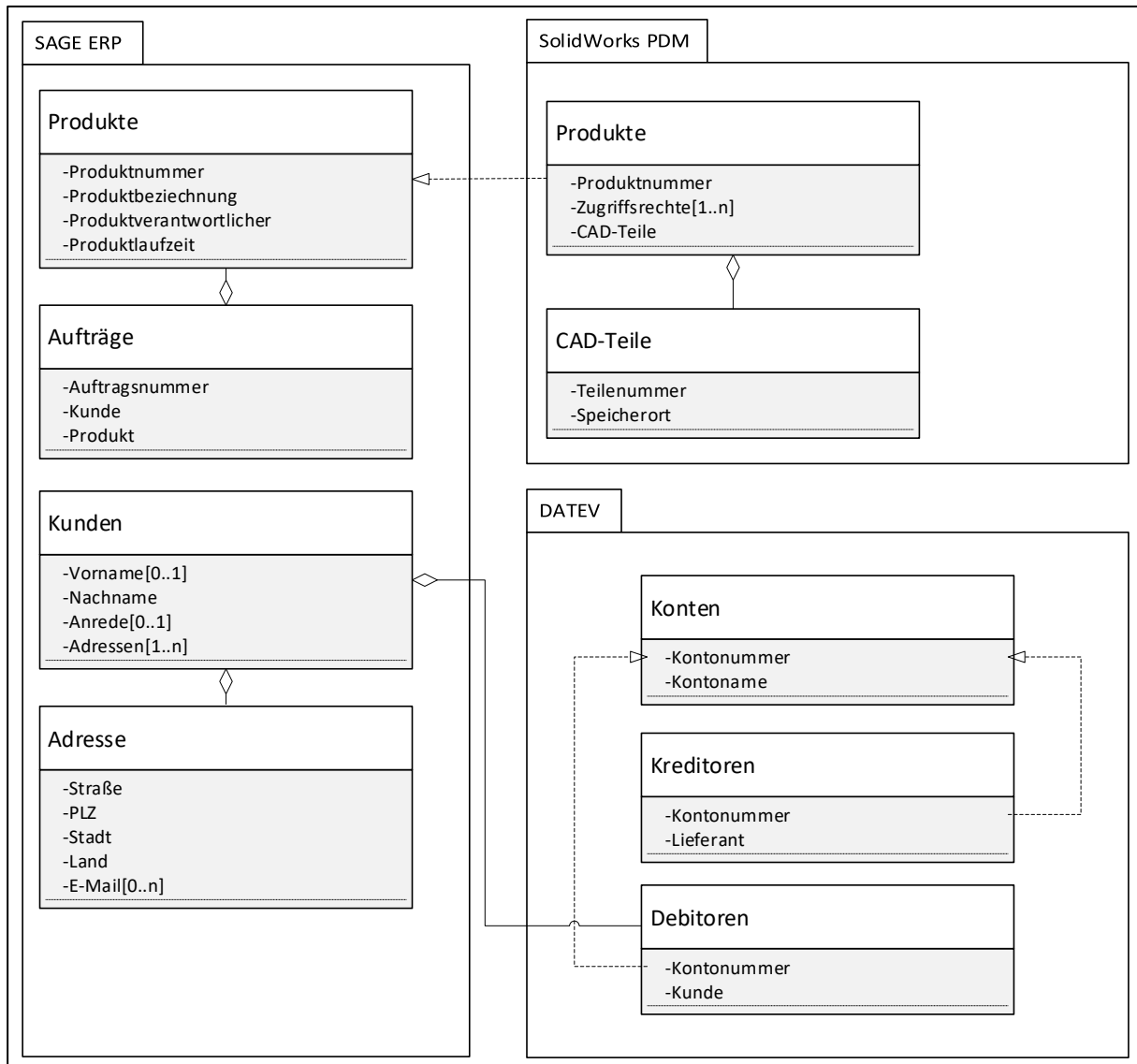


Abbildung 6-20: Vereinfachter Ausschnitt aus der Datensicht einer Informationssystem-Architektur gemäß UML-Klassennotation (eigene Darstellung)

Neben den Architekturdarstellungen auf Anwendungsebene sind vor allem **Daten- beziehungsweise Informationsarchitekturen** in der Literatur weit verbreitet. Oft beschränken sie sich dabei auf die Darstellung von Architekturen *innerhalb* einzelner Informationssysteme (vgl. z. B. ALPAR ET AL. 2016, S. 175-177). Es werden dazu Techniken wie *Entity-Relationship-Model* (ERM) oder UML-Klassendiagramme genutzt. In der vorliegenden Arbeit sollen diese Darstellungen jedoch auf die Gesamtarchitektur angewendet werden.

Abbildung 6-20 zeigt einen Ausschnitt aus einer Datenarchitektur in UML-Klassennotation, geordnet nach einzelnen Informationssystemen. In einer solchen

systematischen Darstellung werden Informationsobjekte mit zugehörigen Datenpunkten als Tabelle gezeigt und durch Pfeile logische Zusammenhänge visualisiert. Um die Darstellung nicht zu umfangreich zu gestalten und die Lesbarkeit zu erhalten, müssen auch hier entweder Ausschnitte der Gesamtarchitektur fokussiert oder es muss eine grobe Darstellung nur der unternehmensweit bedeutsamen Daten vorgenommen werden (s. TIEMEYER 2017a, S. 117).

Anhand entsprechender Darstellungen kann die **Mehrfachhaltung von Daten identifiziert** werden. Sie zeigt außerdem, ob diese Daten auf ein Master-Objekt referenzieren oder ob sie jeweils voneinander unabhängig gehalten werden. Somit ist die Datenarchitektur ein wichtiges Werkzeug zur Stammdatenharmonisierung, als deren erster Schritt von NIENKE U. FUHS die Datenanalyse vorgeschlagen wird (s. NIENKE U. FUHS 2017, S. 19). Auch daraus abgeleitete Ziel-Zustände und Darstellung der künftigen Datenstruktur können mit der gezeigten Darstellungsform visualisiert werden.

In Bezug auf die Integration einer software-definierten Plattform zeigt eine Datenarchitektur die Eingriffspunkte auf, an denen Schnittstellen einer solchen Plattform angeschlossen werden müssen. Aus vorhandenen Datenmodellen, beispielsweise für den digitalen Schatten in der Auftragsabwicklung (s. SCHUH U. BLUM 2016, S. 2166), können die notwendigen Integrationsbedarfe abgeleitet werden.

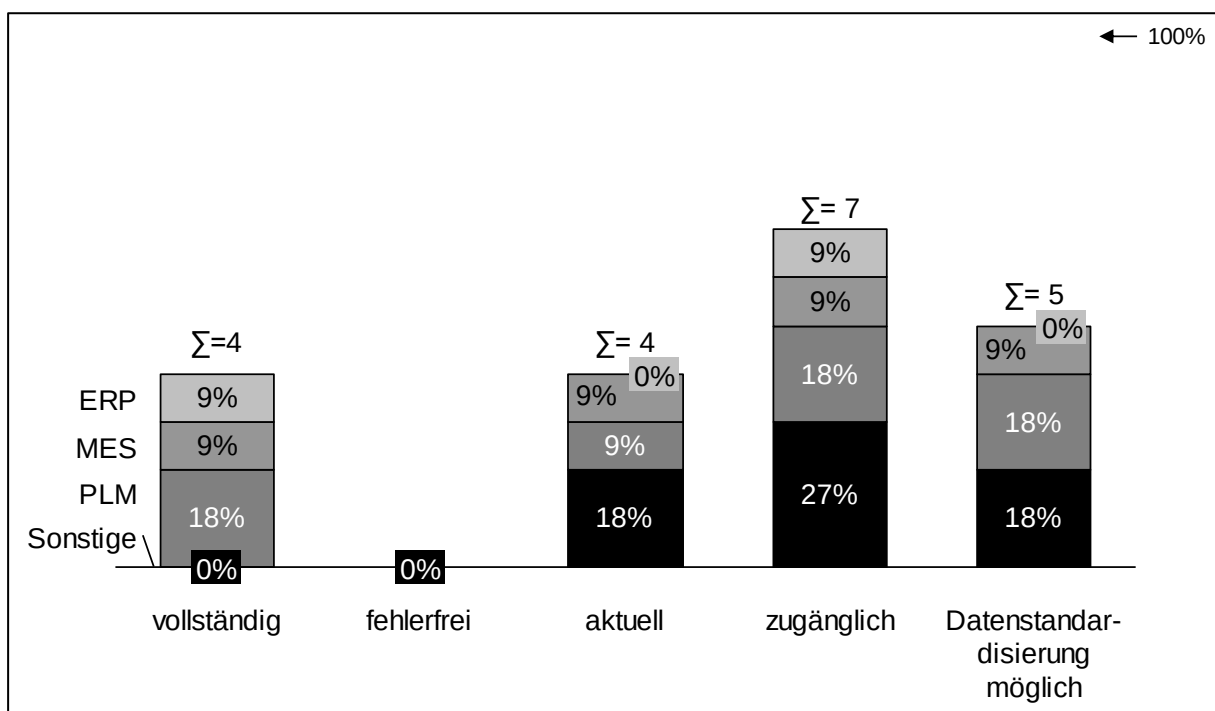


Abbildung 6-21: Beispielhafte Darstellung der Informationsqualität und Datenstandardisierung in einer Informationssystem-Architektur (eigene Darstellung)

Eine **weitere Form der Darstellung von Datenarchitekturen sind Auswertungen**, wie beispielhaft in Abbildung 6-21 gezeigt. Dort ist zusammengefasst, inwieweit verschiedene Informationssysteme, geordnet nach Systemkonzept, den Kriterien der Datenqualität (s. Kapitel 6.5.2) entsprechen, und inwieweit die Datenstandardisierung (s. Kapitel 6.5.3) in den Systemen ermöglicht wird. Solche Darstellungen geben eine gute Übersicht darüber, wie hoch der **Aufwand zur Konsolidierung in Vorbereitung**

einer **Plattformintegration** ist. Ebenfalls kann damit auch die Entscheidung unterstützt werden, bei welcher Systemwelt ein Umsetzungsprojekt begonnen werden kann (s. Kapitel 8).

6.8 Typen von Informationssystem-Architekturen produzierender Unternehmen

Nachdem das Wesen von Informationssystem-Architekturen in verschiedenen Darstellungsdimensionen vorgestellt wurde, werden nachfolgend Typen von Informationssystem-Architekturen identifiziert. Zur Entwicklung der Typen wurden dabei die Systematisierung von Informationssystemen aus Kapitel 6.6 sowie die Architekturdefinition des vorherigen Kapitels genutzt.

Die **Ableitung von allgemeingültigen Typen** ist dabei, auch wenn sie auf produzierende Unternehmen eingegrenzt ist, nur grob und auf Basis konzeptioneller Architekturtreiber möglich. Diese Treiber – die **IT-Organisationspositionierung** und das **Architekturparadigma** – wurden einerseits auf Basis existierender Literatur, andererseits auf Basis von Untersuchungen realer Architekturen abgeleitet.

Die theoretische Fundierung bilden die vier Positionierungstypen von Informationssystemen, die von MCFARLAN ET AL. entwickelt und von HANSCHKE sowie KRCMAR auf Organisationstypen von IT-Abteilungen übertragen wurden (s. MCFARLAN ET AL. 1983, S. 150-153; KRCMAR 2015, S. 404-406; HANSCHKE 2016, S. 184-185) – eine detaillierte Vorstellung findet sich in Kapitel 2.2.4. Diese vier Organisationstypen, in dieser Arbeit bezeichnet als *Technologiepartner*, *Prozesspartner*, *Geschäftspartner* und *Erfolgspartner*, wirken direkt und indirekt auf die Gestaltung von Informationssystem-Architekturen in Unternehmen. Tritt die IT-Abteilung als **Erfolgspartner** auf, unterstützt sie nicht nur die Weiterentwicklung der Geschäftsmodelle, sondern auch die Geschäftsprozesse der Fachbereiche optimal. Dazu sind **durchgängige, durch Informationssysteme unterstützte Prozesse** nötig, was wiederum nur durch eine geordnete Architektur mit geringer Funktions- und Datenredundanz möglich ist. In gleicher Weise führt eine IT-Abteilung, die als **Technologiepartner** auftritt, dazu, dass Prozesse und zugehörige Informationssysteme nicht zentral gesteuert werden und stattdessen **lokale Optima durch die IT-Entscheidungen einzelner Fachbereiche** entstehen (s. WINTER 2014, S. 361). Um die Anzahl an Informationssystem-Architekturtypen handhabbar zu gestalten, wurden für diese Arbeit die beiden MCFARLAN'schen Typen *Prozesspartner* und *Geschäftspartner* zusammengefasst, da in beiden Fällen die Prozessunterstützung im Vordergrund der für die IT-Abteilung handlungsleitenden IT-Strategie steht.

Eine weitere Fundierung der Informationssystem-Architekturtypen kommt aus der Praxis. Dort haben sich zwei Architekturparadigmen gezeigt, die in der Literatur als klassisches Begriffspaar in verschiedensten Wissenschaften wie der BWL, Volkswirtschaftslehre (VWL), Organisationstheorie oder auch der Soziologie Verwendung findet: Zentralität und Dezentralität (s. FRESE ET AL. 2012, S. 214-215). Viele produzie-

rende Unternehmen setzen auf ein **zentrales ERP-System**, in vielen Fällen vom Hersteller SAP, als zentrales, stamm- und bewegungsdatenhaltendes Informationssystem, mit dem andere Informationssysteme über Schnittstellen verbunden sind. Eine Informationssystem-Architektur, die dem **dezentralen Architekturparadigma** unterliegt, findet sich ebenfalls in vielen produzierenden Unternehmen. Dort werden mehrere, auf bestimmte Anwendungsfälle beziehungsweise Abteilungen spezialisierte Informationssysteme eingesetzt, die jeweils einen Teil des Stamm- und Bewegungsdatenbestands des Unternehmens halten. Beide Architekturparadigmen haben ihre Vor- und Nachteile, die hier jedoch nicht näher erläutert werden sollen.⁴⁰

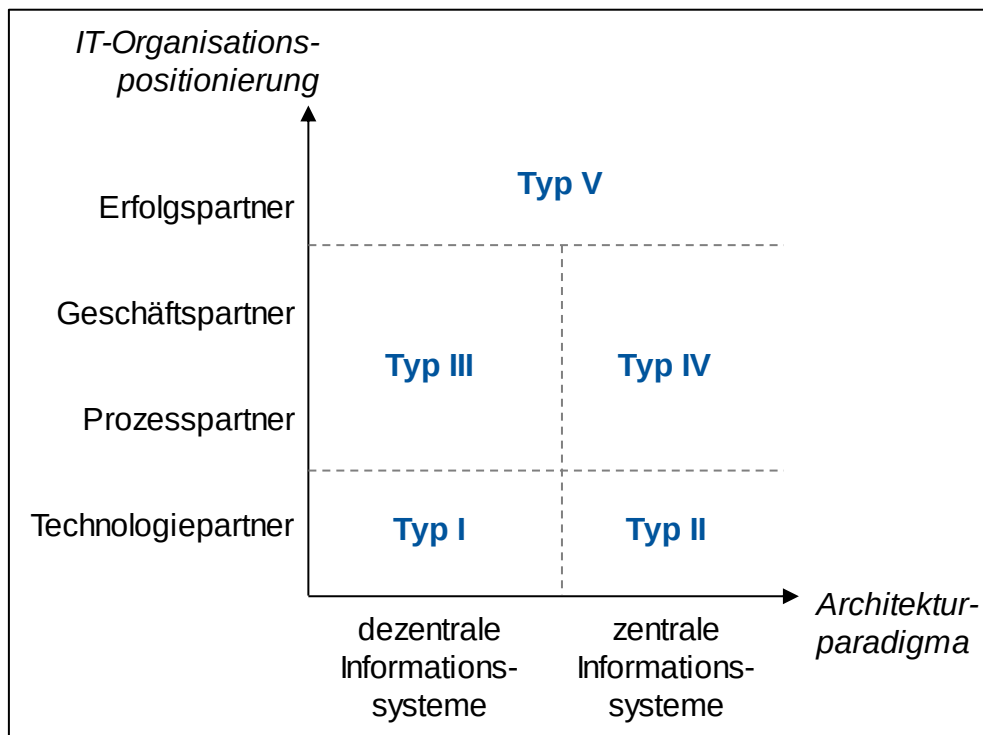


Abbildung 6-22: Einordnung der Informationssystem-Architekturtypen (eigene Darstellung)

Aus der Kombination der beiden Treiber *IT-Organisationspositionierung* und *Architekturparadigma* können **nun fünf Informationssystem-Architekturtypen abgeleitet** werden, die in Abbildung 6-22 veranschaulicht sind. Auf weitere Dimensionen, wie beispielsweise die Typenbildung auf Basis von Komplexitätskennzahlen, die auf ein Entropiemaß und damit im Endeffekt auf eine Zählung von Informationssystemen und Schnittstellen basieren, wie sie DERN U. JUNG oder SCHUETZ ET AL. vorschlagen (vgl. DERN U. JUNG 2009, S. 671; SCHUETZ ET AL. 2013, S. 5; SCHÜTZ 2017,

⁴⁰ Die Hauptvorteile einer zentralen Informationssystem-Architektur liegen in der Redundanzfreiheit und in Kompatibilitätsvorteilen, da hauptsächlich in einer Systemwelt eines Anbieters gearbeitet wird. Gleichzeitig ist eine solche Architektur schwerfällig und die Abhängigkeit von einem Anbieter ist groß. Das Konträre gilt für eine dezentrale Architektur: Dort werden die Vorteile der Spezialisierung der einzelnen Systeme genutzt, es herrscht prinzipiell eine höhere Flexibilität, die jedoch mit erhöhten Aufwänden in der Sicherstellung von Kompatibilitäten der inhaltlichen und technischen Schnittstellen zwischen den einzelnen Informationssystemen einhergeht.

S. 130-132), soll hier verzichtet werden. Solche Kennzahlen geben allenfalls einen Hinweis auf Herausforderungen in der Architektur, können aber aufgrund der hohen Informationsverdichtung nicht zur Ableitung von Handlungsempfehlungen genutzt werden (s. Kapitel 3.5).

Die Architekturtypen bilden sich aus den **Schnittmengen der beschriebenen Ausprägungen** der IT-Organisationspositionierung – bei Zusammenfassung von Geschäfts- und Prozesspartner – sowie den beiden Architekturparadigmen. *Typ V* umfasst dabei sowohl dezentrale als auch zentrale Informationssystem-Architekturen, da im Falle einer IT-Abteilung, die als Erfolgspartner agiert, diese Unterscheidung keine besondere Relevanz hat (s. Kapitel 6.8.5). Dies ist darin begründet, dass solche IT-Abteilungen den Fachbereichen eine Architektur bereitstellen, bei der die Mitarbeiter direkten Zugriff auf die richtigen Informationen haben. Die technische Realisierung ist in einem solchen Fall zweitrangig.

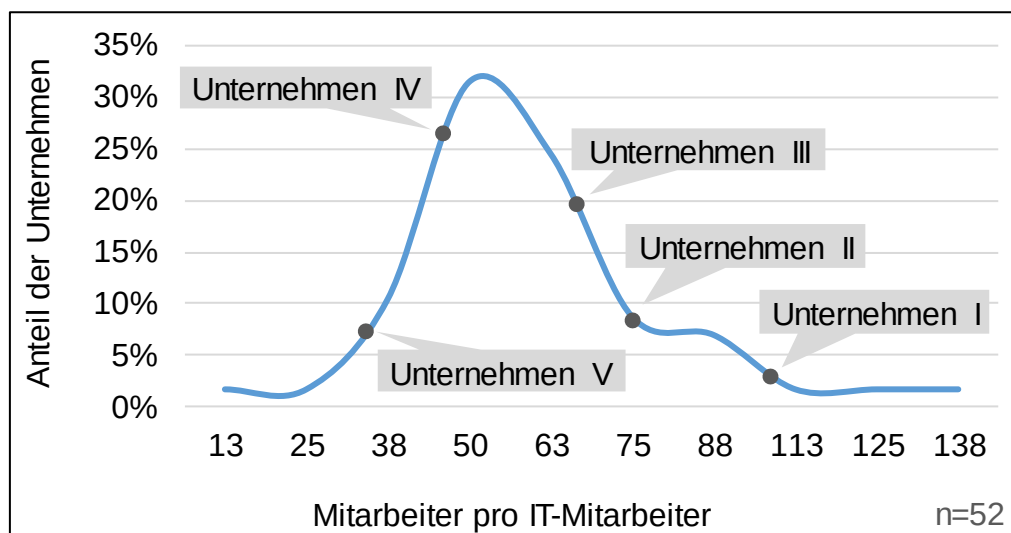


Abbildung 6-23: Verhältnis der Mitarbeiter zu IT-Mitarbeitern (eigen und fremd) und Einordnung der Beispielunternehmen (eigene Darstellung auf Basis von REIMANN 2017, S. 7)

Alle fünf Typen werden im Folgenden detailliert **anhand jeweils eines anonymisierten Beispiels aus der Praxis vorgestellt** und die Eigenschaften der Typen hervorgehoben.⁴¹ Auch hier **dienen die Praxisbeispiele der einfacheren Verständlichkeit**. Jedes im Beispiel eingesetzte Informationssystem kann prinzipiell auch durch ein System eines anderen Softwareherstellers ersetzt werden. Zur Einordnung der jeweiligen IT-Organisationen der vorgestellten Unternehmen sind diese in der Abbildung 6-23, welche das **Betreuungsverhältnis** der Mitarbeiter des Unternehmens durch IT-Mitarbeiter darstellt (s. REIMANN 2017, S. 7), als „Unternehmen I“ bis „Unternehmen V“ einsortiert. Sie zeigt, jedenfalls für die untersuchten Unternehmen, einen Zusammenhang zwischen einem geringen Betreuungsverhältnis und der Positionierung der IT-

⁴¹ Die Datengrundlage aller Beispiele entstammen Projekten des FIR aus den Jahren 2015 bis 2017. Sie wurden anonymisiert und teilweise vereinfacht, um eine sinnvolle Darstellung zu ermöglichen.

Abteilung. Dies ist darin begründet, dass eine IT-Abteilung als *Erfolgspartner* deutlich mehr Aufgaben wahrnimmt als eine IT-Abteilung als *Technologiepartner*.

6.8.1 Architekturtyp I: Dezentrales Architekturparadigma und IT-Organisation als Technologiepartner

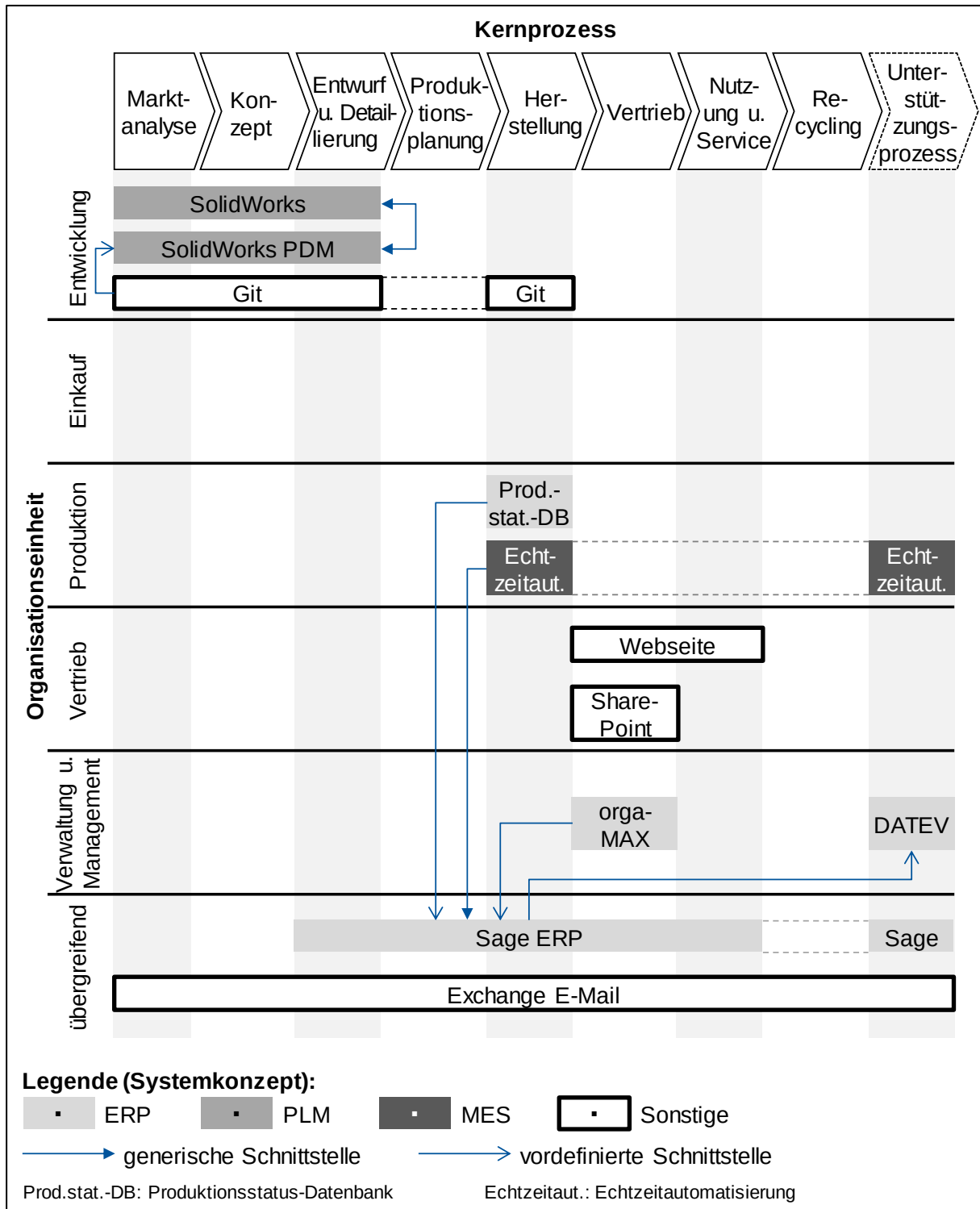


Abbildung 6-24: Beispiel für eine Informationssystem-Architektur nach Typ I eines Kleinserienproduzenten „Unternehmen I“ (eigene Darstellung gem. Kapitel 6.7)

Als typisches Beispiel für eine Informationssystem-Architektur nach *Typ I* wurde ein mittelständisches Unternehmen – genannt *Unternehmen I* – gewählt, das in Kleinserie

mechatronische Anlagen entwickelt und produziert.⁴² Einen Ausschnitt aus dessen Informationssystem-Architektur zeigt Abbildung 6-24, mit Fokus auf der Einordnung der Informationssysteme im Kernprozess des Unternehmens, der Systemkonzepte sowie der Schnittstellen zwischen den Systemen. Die Darstellung zeigt das dezentrale Architekturparadigma des Unternehmens. Da die IT-Abteilung sich als Technologiepartner versteht, ist die Dezentralität der Architektur nicht zentral geplant, sondern historisch gewachsen. Die wichtigsten Informationssysteme sind ein ERP- und ein PDM-System, die jeweils durch zusätzliche, teils selbst programmierte Informationssysteme ergänzt werden und durch Schnittstellen verbunden sind. Das ERP-System in diesem Unternehmen wird in mehreren Mandanten – für unterschiedliche Abteilungen – geführt, was einen Informationsaustausch zwischen Mitarbeitern zusätzlich erschwert.

Strukturiert nach den vier in der vorliegenden Arbeit genutzten Integrationsdimensionen zeigen sich, unabhängig vom oben gezeigten Beispiel, folgende allgemeingültigen Eigenschaften von Informationssystem-Architekturen vom *Typ I*. In der **technischen Sicht** werden viele Informationssysteme eingesetzt, die im Softwarelebenszyklus schon weit fortgeschritten sind und sich teilweise sogar bereits außerhalb der Wartung des Softwareherstellers befinden. Sie werden nur aktualisiert, wenn der Fachbereich, der die Verantwortung für diese Informationssysteme trägt, eine neuere Version anfordert. Dies ist beispielsweise häufig bei CAD-Systemen der Fall, da nur in aktuellen Versionen die volle Kompatibilität beim Datenaustausch mit Lieferanten und Kunden gegeben ist. Outsourcingkonzepte im Applikationsbereich werden nicht strukturiert umgesetzt, sondern es werden in Einzelfällen Sourcing-Entscheidungen getroffen, was zu einer „zerklüfteten“ Informationssystem-Architektur führt. Da die Hauptverantwortung über die Informationssysteme, zumindest über deren Anwendungsteil, bei den Fachbereichen liegt, werden Schnittstellen oft nur marginal umgesetzt, was zu viel manuellem Datenaustausch führt.⁴³ Insgesamt ist die Aufgabenstruktur des Betriebs der Informationssystem-Architektur, bezogen auf *alle* ihre Informationssysteme, oft nicht klar geregelt.

Wie auch im vorliegenden Beispielfall haben Unternehmen des *Typs I* oft eine vergleichsweise geringe Anzahl an IT-Mitarbeitern (s. Abbildung 6-23). Da sie sich als Technologiepartner verstehen, **konzentrieren sich ihre Tätigkeiten auf die Pflege und Wartung der Backend-IT-Infrastruktur und der Client-Systeme**, wie PCs und Laptops beziehungsweise Drucker. In Softwareentscheidungen wird die IT-Abteilung meist nur beratend für IT-Infrastruktur-Themen eingebunden, hat damit also wenig Möglichkeit, auf eine Standardisierung der Anwendungslandschaft hinzuwirken. Auf-

⁴² Dasselbe Unternehmen hat schon als Beispiel in Kapitel 6.7 gedient.

⁴³ Beispiel: Im vorliegenden Fall des *Unternehmens I* mussten bis vor kurzem Rechnungen, die in einem ERP-System erzeugt wurden, per Papier an die Buchhaltung übergeben werden, wo sie in einem manuellen, fehleranfälligen Prozess wieder in die Buchhaltungssysteme übernommen wurden.

grund der lokalen Optimierung, welche die Fachbereiche vornehmen, kommen oft Individualsysteme zum Einsatz, die dann vom Fachbereich selbst betrieben werden. Besonders in der Produktionsabteilung ist dies vermehrt anzutreffen (s. Kapitel 6.3.2).

Die nur lokale Optimierung führt in Unternehmen des *Typs I* oft dazu, dass **keine durchgängigen Prozesse** möglich sind, unter anderem deshalb, weil Informationssysteme eines Systemkonzepts mehrfach vorhanden sind und unterschiedlich genutzt werden. Oft fehlt in solchen Unternehmen auch der Überblick über die von Informationssystemen unterstützte Prozesslandschaft. Dies ist vermehrt der Fall, wenn das Unternehmen unterschiedliche, räumlich getrennte Standorte besitzt. Dann finden sich, trotz gleichen oder ähnlichen Geschäftsmodells, unterschiedliche Prozessumsetzungen unter Nutzung anderer Informationssysteme. Aufgrund manuell vernetzter Systeme sind Prozessanpassungen oft nur mit hohem Aufwand möglich, was die Flexibilität von Geschäftsprozessänderungen erschwert.

Zuletzt führt ein solches Szenario dazu, dass eine **Steuerung des Unternehmens aufgrund von Daten nur eingeschränkt** möglich ist. Im Speziellen ist dies dadurch begründet, dass Stamm-, Bewegungs- und Produktdaten in verschiedenen Systemen oft mehrfach vorliegen und nicht harmonisiert sind. Die Informationsqualität hinsichtlich Vollständigkeit, Aktualität und Zugänglichkeit ist damit im Gesamten nicht gegeben.

6.8.2 Architekturtyp II: Zentrales Architekturparadigma und IT-Organisation als Technologiepartner

Unternehmen II, das beispielhaft für den *Informationssystem-Architekturtyp II* steht, ist ein produzierendes Unternehmen des Sondermaschinenbaus. Der Kernprozess in der Abbildung 6-25, welche dessen Informationssystem-Architektur darstellt, ist daher umsortiert: Der Vertrieb erfolgt *vor* der Produktkonzeption und -detaillierung. In diesem Unternehmen soll das ERP-System als zentrales Informationssystem genutzt werden, um dort alle relevanten Informationen des Produkts, des Produktionsprozesses und der Logistik-, Service- und Buchhaltungsabwicklung zu hinterlegen. Da das Unternehmen jedoch über eine geringe Betreuungsquote durch IT-Mitarbeiter verfügt (s. Abbildung 6-23) und auch keine andere Person die Systemnutzung forciert, ist eine Durchsetzung der entsprechenden Prozesse nur schwer machbar und aktuell nicht umgesetzt. Die IT-Mitarbeiter würden zwar über die Fähigkeiten dazu verfügen, sind jedoch mit reinen Betriebsaufgaben überlastet und treten daher als Technologiepartner auf.

Bezogen auf die **technische Dimension** zeichnen sich Informationssystem-Architekturen des *Typs II* allgemein dadurch aus, dass sie über einen hohen Anteil an intern betriebenen Informationssystemen verfügen. Das ist darin begründet, dass solche produzierenden Unternehmen konservativ ausgerichtet und neuen Technologien gegenüber (Stichwort „Angst vor der Cloud“) eher abwehrend eingestellt sind; es führt aber auch dazu, dass die IT-Abteilung sehr viele Betriebsaufgaben selbst wahrnehmen muss. Aufgrund geringer Investitionsbereitschaft hinsichtlich Informationstechnologien finden sich, wie in *Unternehmen II*, oft selbst entwickelte, nicht fehlerfrei funktionie-

rende Schnittstellen zwischen dem zentralen ERP-System und anderen Informationssystemen. Die Systemdurchdringung von Prozessen ist, ebenfalls bedingt durch ein fehlendes Korrektiv und fehlende Kapazitäten für die Mitarbeiterschulung, teilweise gering. Das bedeutet, dass Prozessschritte manuell, teils papierbasiert und nicht systemunterstützt durchgeführt werden – ebenfalls teilweise begründet durch das Fehlen funktionaler Schnittstellen.

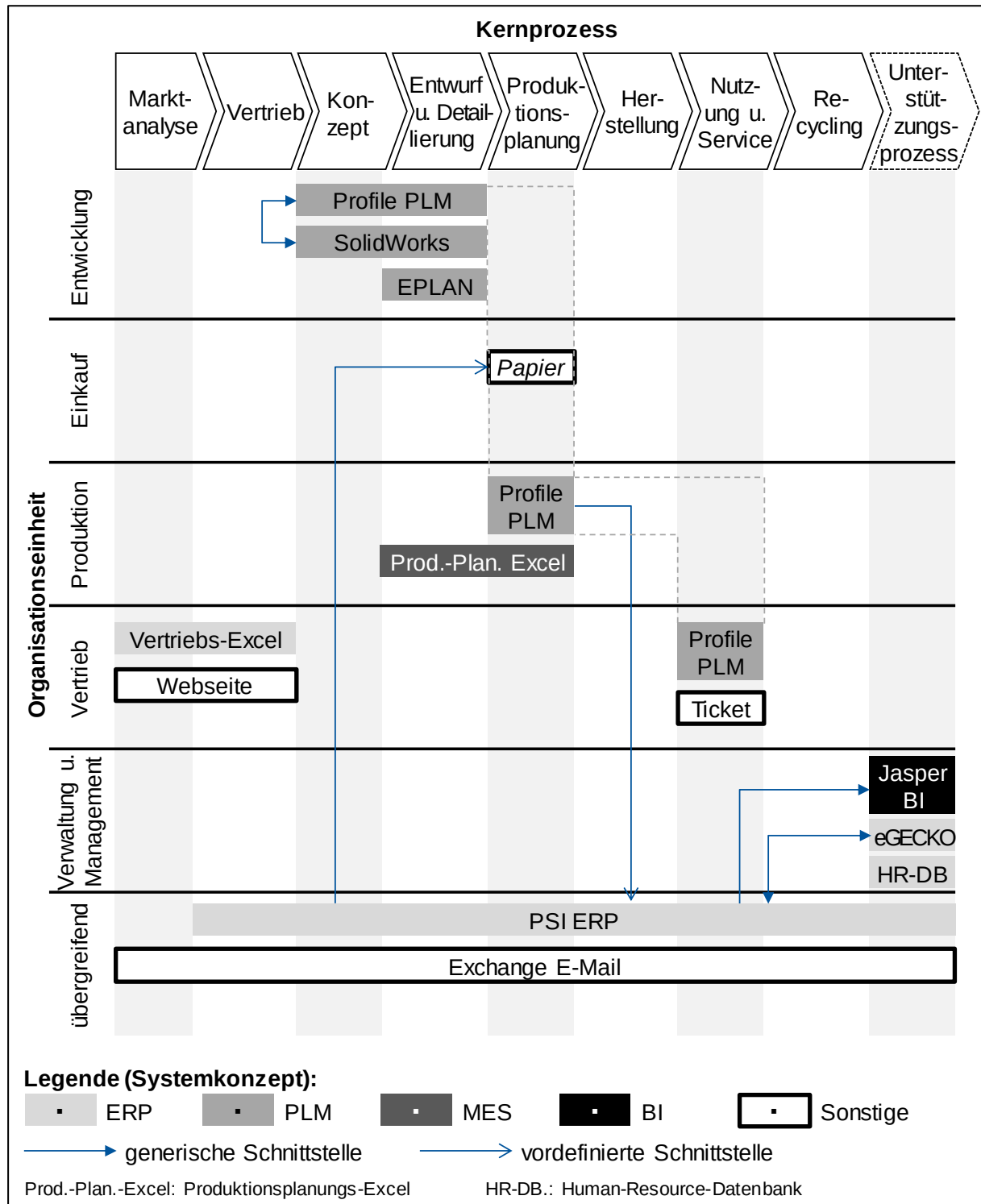


Abbildung 6-25: Beispiel für eine Informationssystem-Architektur nach Typ II eines Sondermaschinenbauers „Unternehmen II“ (eigene Darstellung)

Bezogen auf **organisatorische Aspekte** der Informationssystem-Architektur zeigt sich in *Typ II* nicht nur die geringe Zahl an IT-Mitarbeitern und ein geringer Outsourcinganteil, sondern auch ein hoher Anteil an individualprogrammierten Informationssystemen. Diese entstehen, weil die Funktionalität des zentralen ERP-Systems nicht vollständig genutzt wird oder weil aufgrund von Unkenntnis oder fehlender Investitionsbereitschaft vordergründig einfachere Informationssysteme durch die Fachbereiche etabliert wurden. Diese Lösungen sind meist nicht, oder nur in geringem Maß, mit dem ERP-System über Schnittstellen verbunden, was zu Informationsasynchronität im Unternehmen führt. Die Einführung von vielen Informationssystemen durch Fachbereiche führt außerdem zu fehlender Transparenz über die Informationssystem-Architektur des Gesamtunternehmens, weil eine zentrale Stelle, welche die Informationen sammelt, fehlt.

Die Beschränkung auf ein zentrales ERP-System führt dazu, dass zwar nicht mehrere Informationssysteme eines Systemkonzepts vorliegen, dafür jedoch eine Reihe von Excel- oder Access-Lösungen etabliert werden. **Durchgängige**, gegebenenfalls sogar echtzeitfähige **Prozesse sind damit nur schwer möglich**. Gleichzeitig fehlt in solchen Unternehmen die Übersicht über die Prozesslandschaft des Unternehmens, an der die Informationssysteme ausgerichtet werden könnten.

Ebenfalls sind die **Daten**, die der Geschäftsführung auf Basis der vorhandenen Informationssysteme vorliegen, **unvollständig**. Ein gegebenenfalls vorhandenes BI-System kann nur auf diese Daten zugreifen; zentrale Auswertungen benötigen zusätzliche Anreicherungen durch Excel-Listen, die in verschiedenen Abteilungen vorhanden sind. Trotz der Existenz eines zentralen ERP-Systems ist somit die Informationsqualität bezogen auf das Gesamtunternehmen gering, weil das System nicht konsequent in seiner Rolle umgesetzt wird.

6.8.3 Architekturtyp III: Dezentrales Architekturparadigma und IT-Organisation als Prozess- oder Geschäftspartner

Das *Unternehmen III* ist, genauso wie *Unternehmen II*, ein mittelständischer Sondermaschinenbauer, der hier als Beispiel für den *Architekturtyp III* gezeigt wird. Die Informationssystem-Architektur von *Unternehmen III* ist ausschnittsweise in Abbildung 6-26 dargestellt. Sie zeigt einen Dreiklang aus PDM-, ERP- und CRM-System als zentrale, führende Informationssysteme für unterschiedliche Kernprozesse. Ergänzt werden diese um weitere Informationssysteme mit spezifischen Aufgaben, wie ein CAD-System in der Entwicklungsabteilung oder ein Ticketsystem (bei Unternehmen III: „OTRS“) für den Service in der Vertriebsabteilung. Weiterhin sind auch hier einige Excel-Lösungen im Einsatz, die von den Fachbereichen in Absprache mit der IT-Abteilung gepflegt werden. In der IT-Abteilung, die personell im Mittelmaß ausgestattet ist (s. Abbildung 6-23), gibt es sowohl Mitarbeiter, die sich um die IT-Infrastruktur kümmern, als auch solche, welche die Beratung der IT-Anwender, insbesondere in der Nutzung des ERP-Systems, im Fokus ihrer Arbeit haben.

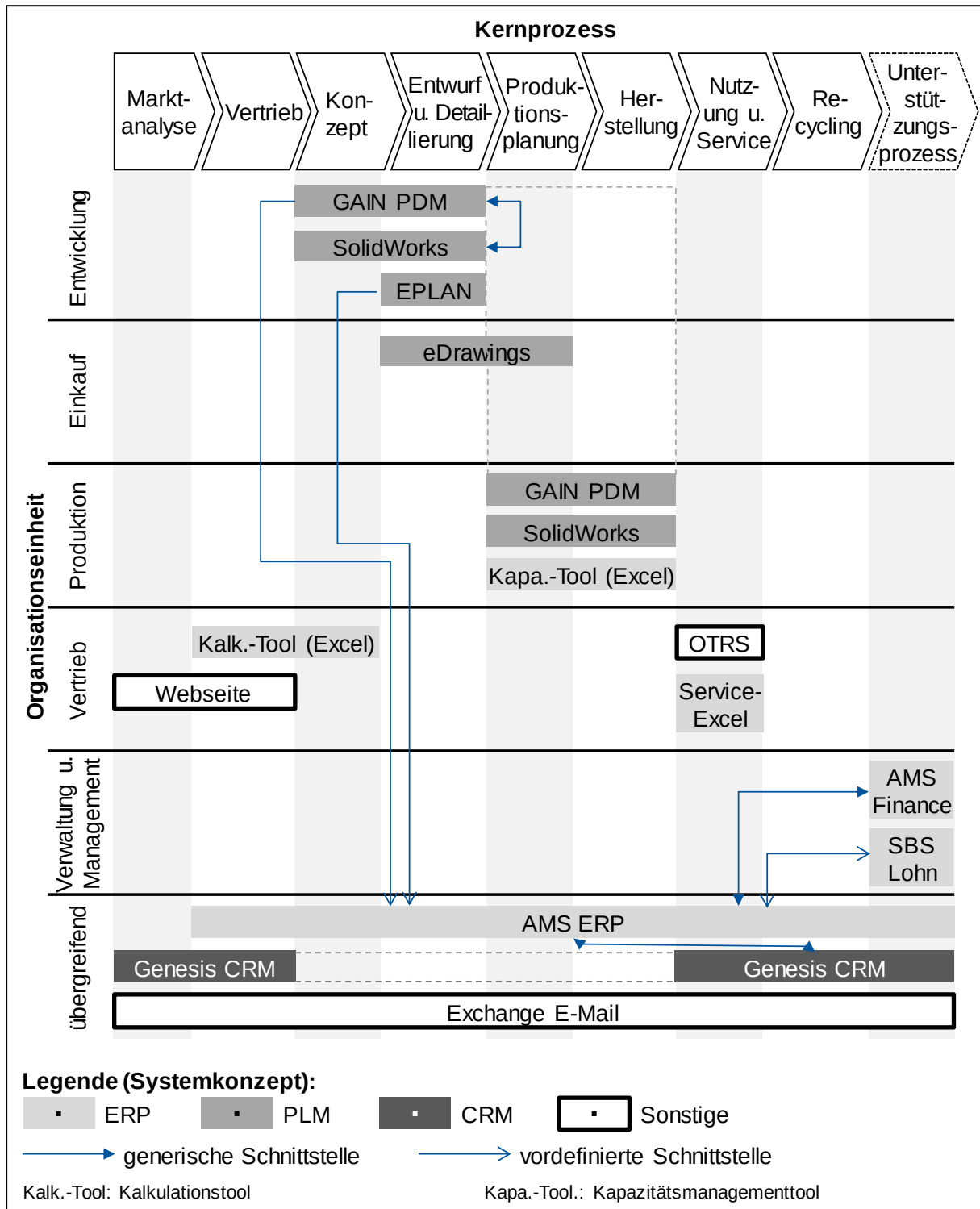


Abbildung 6-26: Beispiel für eine Informationssystem-Architektur nach Typ III eines Sondermaschinenbauers „Unternehmen III“ (eigene Darstellung)

Allgemein betrachtet, haben Unternehmen des *Architekturtyps III* generell einen **höheren Outsourcinggrad der verwendeten Informationstechnologie**, als es die bisher besprochenen *Typen I* und *II* haben. Das resultiert daraus, dass Entscheidungsträger in Bezug auf Informationstechnologie einen geringeren Fokus auf die Technologie selbst legen, sondern stärker die Prozesse und geschäftlichen Abläufe des Unternehmens fokussieren. Gerade das Infrastrukturoutsourcing sowie das Outsourcing des Usersupports sind daher gängig. Das im Beispiel betrachtete *Unternehmen III* hat

solche Leistungen bisher noch in nur sehr geringem Umfang outgesourct, plant jedoch aktuell eine Erweiterung. Informationssystem-Architekturen des *Typs III* zeichnen sich außerdem dadurch aus, dass die Informationssysteme in regelmäßigen Abständen aktualisiert werden, was jedoch zu erhöhtem Aufwand bei der Schnittstellenanpassung führt, wenn eines der Kernsysteme angepasst wird. Die Notwendigkeit, Datendurchgängigkeit herzustellen, ist der IT-Abteilung bewusst, jedoch sind nicht alle Schnittstellen vollautomatisiert, sodass manche Informationssysteme nicht über echtzeitfähige Vernetzung verfügen.

Hinsichtlich der **organisatorischen Themen** zeichnet sich die Informationssystem-Architektur nach *Typ III* dadurch aus, dass der Anteil an Standardsystemen aufgrund von Standardisierungsanstrengungen der IT-Abteilung höher ist als bei Architekturen des *Typs I*. Weiterhin liegt der größte Teil der Systemverantwortung nicht bei den Fachbereichen, sondern bei der IT-Abteilung. Außerdem nutzen IT-Abteilungen solcher Unternehmen zur internen Optimierung oft De-facto-Standards wie ITIL (s. Kapitel 3.3), um sich nach Best-Practice-Ansätzen serviceorientiert aufzustellen. Wegen des nur mittelmäßigen Outsourcinggrades und der oft auch geringen Ausstattung an IT-Mitarbeitern sind die IT-Abteilungen, die Informationssystem-Architekturen des *Typs III* betreuen, jedoch oft stark ausgelastet bis überlastet, weil die Anforderungen der Fachabteilungen aufgrund des höheren Dienstleistungsumfangs der IT-Abteilung auch entsprechend höher sind.

Prozessual liegt der Fokus bei *Typ-III-Informationssystem-Architekturen* auf einer Unterstützung der Kernprozesse jeweils durch ein passendes Kernsystem, wie beispielsweise ein CRM-System im Marketing und Vertrieb, oder ein PDM-System in allen Entwicklungsbereichen sowie dem Service. Im Optimalfall existieren nur ein System jeder Systemwelt im Unternehmen sowie fachlich klar definierte Übergabepunkte, die in Form technischer Schnittstellen umgesetzt sind. Damit ist ein problemloser Datenaustausch, beispielsweise zwischen verschiedenen Standorten, möglich. In dem im Beispiel vorgestellten *Unternehmen III* ist das für die meisten, aber noch nicht für alle Standorte der Fall.

Aufgrund höherer Standardisierung, höherer Informationssystemnutzung und stärkerer Vernetzung ist die **Gesamtinformationsqualität einer solchen Architektur** ebenfalls höher, jedoch erfüllt sie noch nicht alle Eigenschaften der Informationsqualität (s. Kapitel 6.5.2). Weiterhin fehlt auch bei diesem Typ ein zentrales Datenmodell, das in allen Informationssystemen umgesetzt ist. Eine Nutzung von Daten aus mehreren Informationssystemen wird daher erschwert, unter anderem auch bedingt dadurch, dass die Schnittstellen zwischen den Informationssystemen nur auf bestimmte Datentypen, die zum Entwicklungszeitpunkt festgelegt wurden, beschränkt sind.

6.8.4 Architekturtyp IV: Zentrales Architekturparadigma und IT-Organisation als Prozess- oder Geschäftspartner

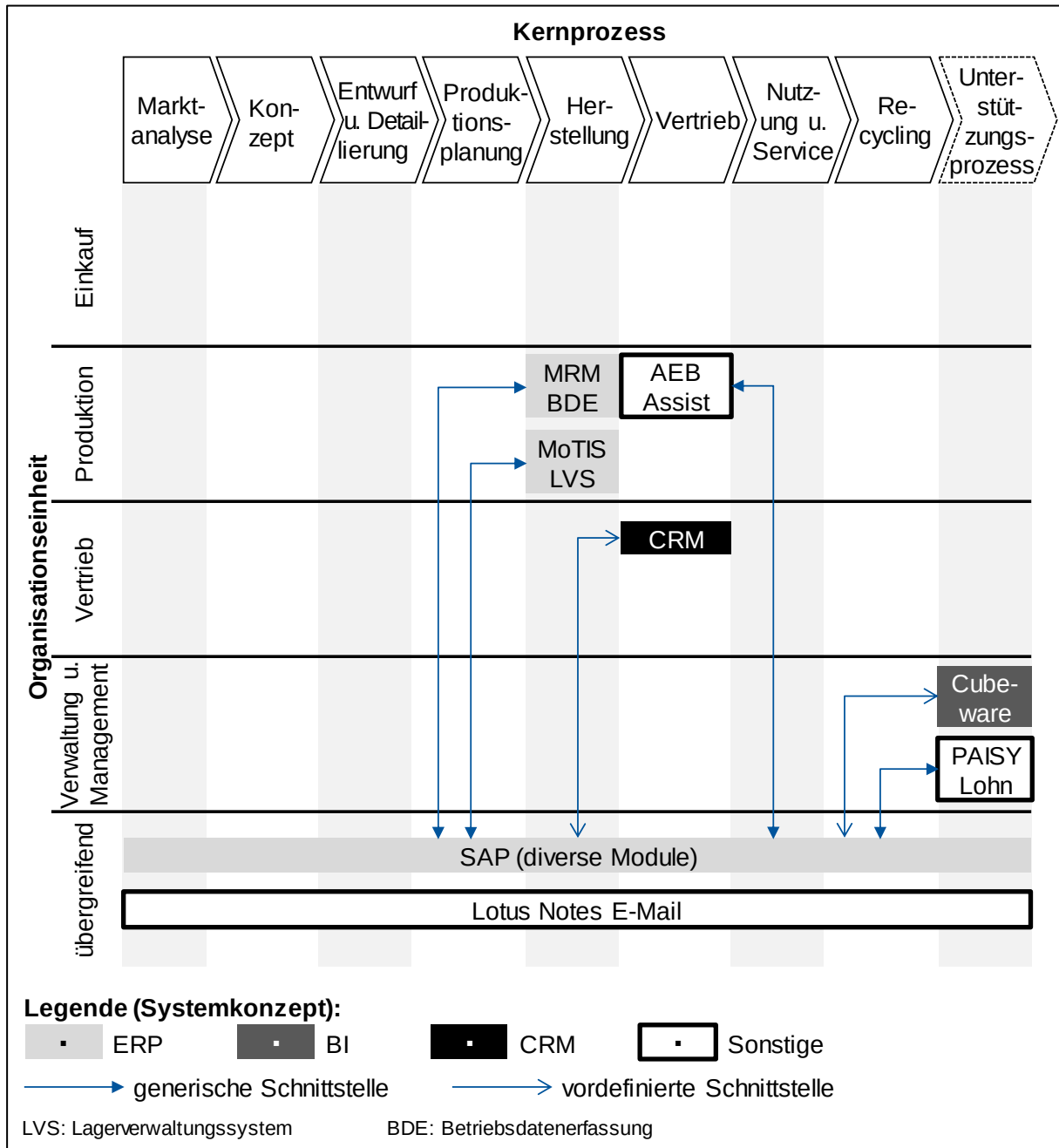


Abbildung 6-27: Beispiel für eine Informationssystem-Architektur nach Typ IV eines Nahrungsmittelproduzenten „Unternehmen IV“ (eigene Darstellung)

Das *Beispielunternehmen IV* steht stellvertretend für ein Unternehmen mit einem zentralen Architekturparadigma und einer fortschrittlichen IT-Organisation, die sich als Geschäftspartner sieht. Als zentrales Informationssystem wurde dort kürzlich SAP mit verschiedenen Modulen neu eingeführt und so etabliert, dass 95 Prozent der Mitarbeiter dieses Unternehmens nun mit diesem System arbeiten. Neben dem zentralen ERP-System werden einige weitere Informationssysteme eingesetzt, die jedoch unterstützend dazu arbeiten und beispielsweise die Erzeugung von Exportdokumenten (bei Unternehmen IV: „AEB Assist“) oder das Management des Lagers (bei Unternehmen IV:

„MoTIS LVS“) ermöglichen. In Abbildung 6-27, welche einen Ausschnitt der Informationssystem-Architektur von *Unternehmen IV* zeigt, fehlt der Bereich „Entwicklung“, da es sich um ein Unternehmen aus der Nahrungsmittelindustrie handelt, in dem keine klassische Produktentwicklung stattfindet. Im *Beispielunternehmen IV* sind außerdem alle Schnittstellen zum zentralen SAP-System automatisiert, weil sie durch den zurückliegenden SAP-Rollout ebenfalls aktualisiert wurden. Die IT-Abteilung sieht sich als Geschäftspartner, konnte jedoch, aufgrund intensiver Arbeiten zum Umbau der Informationssystem-Architektur, noch nicht als Erfolgspartner in Erscheinung treten, welcher die Geschäftsmodelle weiterentwickelt.

In der übergreifenden Betrachtung des *Typs IV*, für den *Unternehmen IV* ein Beispiel bildet, zeigt sich, dass Informationssystem-Architekturen des *Typs IV* in den **technischen Architekturmerkmalen** denen des *Typs III* ähneln: Aufgrund existierender Sourcingstrategien sind einige IT-Leistungen extern vergeben, da sich die IT-Abteilung stärker um die Betreuung der Fachbereichsanforderungen und weniger um einen stabilen Infrastrukturbetrieb kümmert; die eingesetzte Software ist im Softwarelebenszyklus hauptsächlich im Bereich der *Evolution* und des *Servicings*, weil aktuelle Updates regelmäßig installiert werden. Aufgrund der zentralen Architektur und der damit verbundenen zentralen Datenhaltung sind die meisten im Unternehmen vorhandenen Daten echtzeitfähig.

Entsprechend dem zentralen ERP-System trägt eine ebenfalls **zentrale IT-Abteilung die Verantwortung für alle Informationssysteme**. Da solche IT-Abteilungen durch Einbindung der Fachbereiche in verschiedenen Projekten oft eine hohe Kundenzufriedenheit erreichen, existieren auch nur in geringem Umfang prozesskritische, selbstentwickelte Excel-Lösungen. Stattdessen werden veränderte Anforderungen der Fachbereiche zusammen mit der IT-Abteilung erarbeitet und beispielsweise unter Nutzung agiler Methoden gemeinsam implementiert.

Das zentrale ERP-System trägt außerdem dazu bei, dass **Prozesse** innerhalb der Architektur **medienbruchfrei** ablaufen können. Andere Informationssysteme müssen sich dem zentralen System unterordnen und entsprechend konfiguriert werden. Ein guter Prozessüberblick wird allenfalls durch eine unter Umständen unzureichende Dokumentation der Prozesse innerhalb des ERP-Systems erschwert, da die Aufgabe der Dokumentation auch in fortschrittlichen IT-Abteilungen oft nicht prioritär verfolgt wird.

Ebenfalls begründet durch das zentrale ERP-System, existiert in Informationssystem-Architekturen des *Typs IV* ein **übergreifendes Datenmodell**, das zumindest alle Daten aus dem Office-IT-Bereich beinhaltet. Daten aus OT-Systemen sind nicht unbedingt enthalten. Der hohe Anteil an Mitarbeitern, die im zentralen System arbeiten, führt außerdem zu einer relativ großen Vollständigkeit und leichten Zugänglichkeit der Informationen und damit zu einer insgesamt hohen Gesamtinformationsqualität innerhalb der Architektur.

6.8.5 Architekturtyp V: IT-Organisation als Erfolgspartner

Als Erfolgspartner hat eine IT-Organisation in einer Informationssystem-Architektur des *Typs V* das Ziel, nicht nur durchgängige Prozesse entsprechend den Fachbereichsanforderungen zu ermöglichen, sondern auch zur Weiterentwicklung von Geschäftsmodellen und Innovationen beizutragen. Eine solche Position kann eine IT-Abteilung nur erreichen, wenn sie als gleichberechtigter Partner des Business wahrgenommen wird. Dazu muss sie eine anforderungsgerechte Informationssystem-Architektur bereitstellen, durch welche die Anwender jederzeit die für ihre Arbeit notwendigen Informationen erhalten. Welche Informationssysteme dazu im Einsatz sind, sollte für den Anwender im Optimalfall zweitrangig sein.

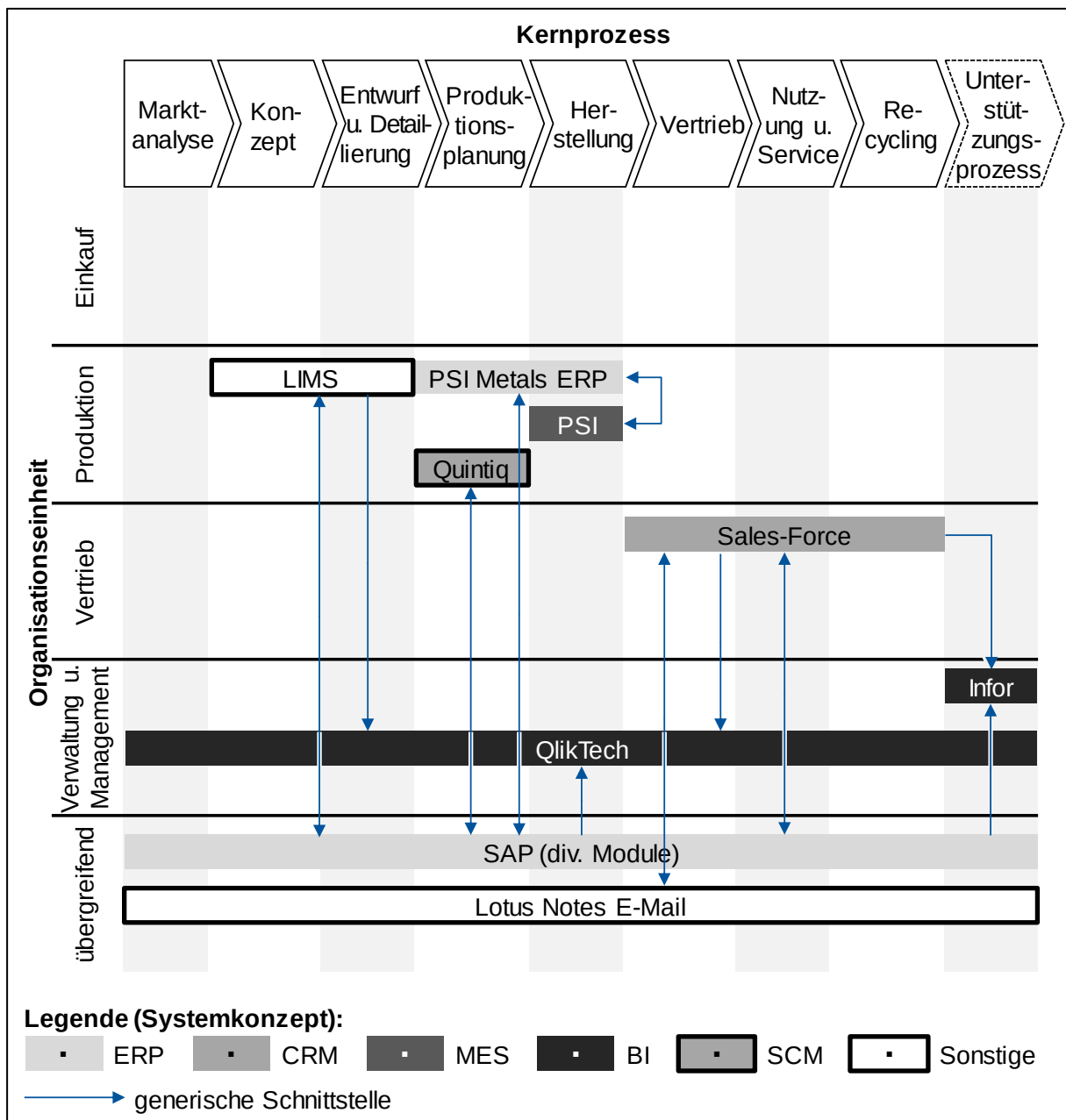


Abbildung 6-28: Beispiel für eine Informationssystem-Architektur nach Typ V eines Metallproduzenten „Unternehmen V“ (eigene Darstellung)

Das *Unternehmen V*, das für den *Architekturtyp V* als Beispiel dient, hat dies durch die Einführung eines übergreifenden BI-Systems umgesetzt. Abbildung 6-28 zeigt dessen Informationssystem-Architektur. Das BI-System bildet dort den Entkopplungspunkt zwischen den datenliefernden Systemen und der Datennutzung. Eine ähnliche Vorstellung verfolgt das Infrastrukturkonzept des *Internet of Production* (s. Kapitel 2.3), wobei dort deutlich mehr Funktionalität hinsichtlich des Datenrückflusses vorgesehen ist, als bei einem BI-System möglich. Weiterhin zeichnet sich die Informationssystem-Architektur des *Unternehmens V* durch ein zentrales ERP-System aus, das aber für die spezifischen Anforderungen eines metallproduzierenden Unternehmens im Produktionsbereich durch ein weiteres ERP-System („PSI Metals“), und im Bereich von Marketing und Vertrieb durch ein CRM-System ergänzt wird.

Bezogen auf die **technischen Merkmale** zeichnet sich eine Informationssystem-Architektur nach *Typ V* allgemein betrachtet vor allem durch eine hohe Vernetzung über generische Schnittstellen aus. Eine „Cloud-first-Strategie“ führt zu einem hohen Outsourcinganteil, um der IT-Abteilung mehr Flexibilität und eine schnellere Umsetzungsgeschwindigkeit zu ermöglichen. Durch den hohen Outsourcinganteil wird überdies auch eine hohe Aktualität der Informationssysteme im Softwarelebenszyklus sichergestellt. Da sich die Systemarchitekturen in modernen Webanwendungen von denen klassischer „On-Premises“-Installationen unterscheiden, ist deren Updatefähigkeit deutlich erhöht.⁴⁴

Im Gegensatz zum zentralen Organisationsansatz von *Architekturtyp IV* zeichnet sich *Architekturtyp V* durch eine **Mischorganisation** aus. Die Verantwortung für einige Informationssysteme liegt hier wieder beim Fachbereich beziehungsweise bei IT-Mitarbeitern, die organisatorisch direkt dem Fachbereich zuzuordnen sind (s. KOPPER ET AL. 2017, S. 103-105). Dies dient dem verbesserten Business-IT-Alignment, weil so die spezifischen Fachbereichsanforderungen besser verstanden und schnell umgesetzt werden können. Gleichzeitig existiert für die Standardisierung, die Umsetzung größerer Implementierungen und die Steuerung des IT-Betriebs eine zweite IT-Organisation, die nach klassischen Implementierungsprozessen arbeitet; man spricht von einer „**Bimodalen IT**“ (s. BERESFORD U. FENN 2016; LEMKE ET AL. 2017, S. 237). Somit existieren sowohl angepasste Standardsysteme an den Stellen, wo der Fachbereich sie benötigt, als auch reine Standardsysteme, die das Rückgrat der Architektur bilden.

Prozessual betrachtet, gibt es in Informationssystem-Architekturen vom *Typ V* zwar Dopplungen im Systemkonzept, diese wurden jedoch, im Gegensatz zu *Typ I*, spezifisch ausgewählt, gut integriert und lassen trotzdem durchgängige Geschäftsprozesse

⁴⁴ Durch die Anwendung von Ansätzen des *Continuous Deployment* zeigen beispielsweise Facebook.com und andere große Internetfirmen, dass dort bis zu 1.000 Updates von deren Webseite am Tag möglich sind (s. SAVOR ET AL. 2016, S. 21).

zu. Die Veränderbarkeit der Geschäftsprozesse innerhalb der Informationssystem-Architektur ist bedingt durch die Architekturen der eingesetzten Informationssysteme. Tendenziell ermöglichen aktuelle Systeme solche Änderungen (s. Kapitel 6.2.4).

Informationssystem-Architekturen vom *Typ V* ermöglichen einen durchgängigen und vollständigen Zugriff auf alle Unternehmensdaten. Im *Beispielunternehmen V* ist dies durch eine BI-Lösung umgesetzt. In jedem Fall werden dazu jedoch ein **einheitliches Datenmodell und standardisierte Daten** in den Systemen benötigt, um beispielsweise eine übergreifende Datenauswertung erst möglich zu machen. Somit verfügen solche Unternehmen über eine übergreifend hohe Datenqualität.

6.9 Zusammenfassung Kapitel 6

Auf Basis der Informationssystemdefinition nach KRCMAR werden im vorliegenden Kapitel zuerst **Merkmale von Informationssystemen produzierender Unternehmen** erarbeitet und vorgestellt (s. Kapitel 6.1 bis 6.6). Die Merkmale sind **spezifisch ausgewählt**, um die Informationssysteme auf deren Integrationsfähigkeit hin zu untersuchen. Unter Nutzung dieser morphologisch definierten Merkmale können daraufhin **Informationssystem-Architekturen** definiert werden (s. Kapitel 6.7). Dazu werden verschiedene **Darstellungsformen** entwickelt, mit denen die Informationssysteme anhand ihrer Merkmale geordnet werden können. Auf diese Weise setzen sich Informationssystem-Architekturen aus mehreren Ausschnitten des multidimensionalen Architekturraums zusammen. Anhand einer Untersuchung verschiedener Informationssystem-Architekturen produzierender Unternehmen aus der praktischen Erfahrung des Autors können daraufhin **Typen von Informationssystem-Architekturen abgeleitet** werden (s. Kapitel 6.8). Diese werden im Folgenden (s. Kapitel 7.2) benötigt, um den Zusammenhang zwischen dem Ist-Zustand der Informationssystem-Architektur und dem Ziel-Zustand bei Einsatz einer software-definierten Plattform zu erklären.

7 Ist- und Ziel-Zustand der Informationssystem-Architektur

Das Ziel der vorliegenden Arbeit ist es, Informationssystem-Architekturen produzieren der Unternehmen bei software-definierten Plattformen zu entwickeln. Das bedeutet, dass der **Ziel-Zustand der Informationssystem-Architektur darzustellen** ist. Unternehmen müssen ihre Informationssystem-Architektur in Richtung dieses Ziel-Zustands transformieren, um eine software-definierte Plattform mit den in Kapitel 5.1 definierten Zielen optimal nutzen zu können.

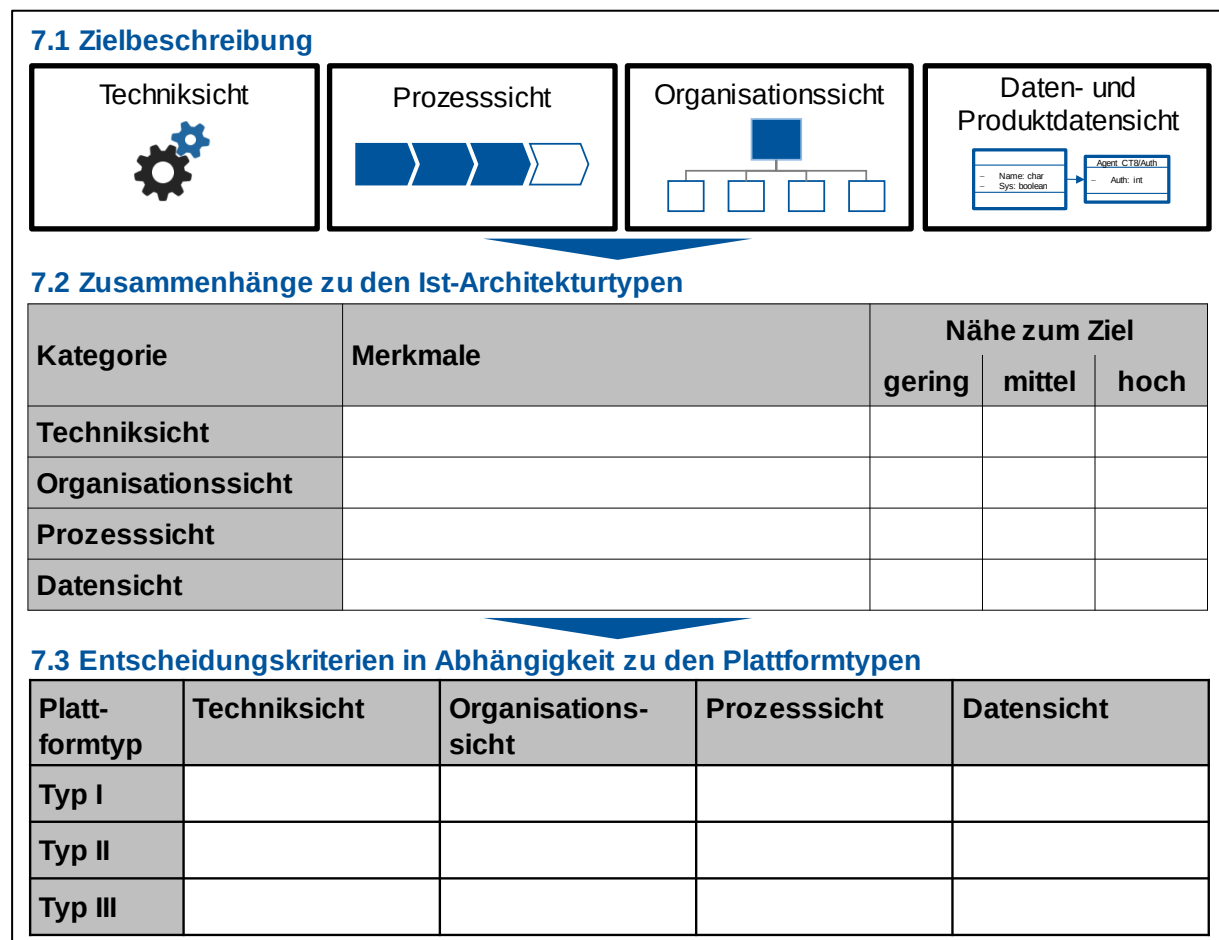


Abbildung 7-1: Übersicht und Zusammenhang der Inhalte des Kapitels 7 (eigene Darstellung)

Um diesen Ziel-Zustand zu definieren, werden in der vorliegenden Arbeit die zuvor definierten Beschreibungsmodelle für software-definierte Plattformen und für Informationssystem-Architekturen (s. Kapitel 5 u. 6) sowie Quellen aus dem Stand der Erkenntnisse (s. Kapitel 3) genutzt. Insbesondere können aus den Integrationsanforderungen software-definierter Plattformen (s. Kapitel 5.6) sowie den relevanten Methoden des IT-Komplexitätsmanagements (s. Kapitel 3.5) Hinweise an die Architekturgestaltung abgeleitet werden.

Das vorliegende Kapitel zeigt auf (s. Abbildung 7-1), wie der Ziel-Zustand der Informationssystem-Architektur aussieht (s. Kapitel 7.1) und wie die Zusammenhänge zwischen den fünf Typen des Ist-Zustands der Informationssystem-Architektur zum Ziel-Zustand zu erklären sind (s. Kapitel 7.2). Für Unternehmen werden daraus konkrete Entscheidungskriterien und -notwendigkeiten abgeleitet (s. Kapitel 7.3). Im darauffolgenden Kapitel 8 wird aufgezeigt, wie die Transformation der Informationssystem-Architekturen konkret stattfinden kann.

Damit beantwortet das vorliegende Kapitel die Forschungsfrage: **Wie kann anhand der Integrationsanforderungen der Unterschied zwischen Ist- und Ziel-Zustand der Informationssystem-Architektur für Unternehmen identifiziert werden?**

7.1 Zielbeschreibung der Informationssystem-Architektur

Bei der Entwicklung eines Ziel-Zustands einer Informationssystem-Architektur sind die Ziele einer Architekturgestaltung zu beachten. So ist eine gute Architektur eine, „welche die an sie gestellten Ziele erfüllt“ (HABERFELLNER ET AL. 2015, S. 186). Für moderne Informationssystem-Architekturen können in diesem Zusammenhang in der aktuellen Literatur allgemeine Ziele an solche Architekturen abgeleitet werden.

Eine **handhabbare IT-Komplexität** zu erreichen, ist das wichtigste und übergeordnete Ziel, das an eine Informationssystem-Architektur gestellt werden muss (s. FOTH 2016, S. 29; URBACH U. AHLEMAN 2016, S. 127-128; SCHUH ET AL. 2017e, S. 61; SCHÜTZ 2017, S. 1-2). Die Herausforderung von Komplexität entsteht durch die beiden Dimensionen der interdependenten Vielfalt und der hohen Dynamik und wird verstärkt durch die diffuse Wahrnehmung der Situation (s. SCHUH ET AL. 2013, S. 38-52; SCHUH U. RIESENER 2018, S. 12-13). Der Autor hat diese Definition auf die Informationssystem-Architektur übertragen und zusammen mit anderen Autoren wie folgt veröffentlicht:

IT-Komplexität beschreibt die Komplexität der Informationssystem-Architektur von Unternehmen, getrieben durch die **interdependente Vielfalt der Elemente** (z. B. Applikationen, Daten, Schnittstellen) und die **hohe Dynamik** von technologischen Entwicklungen und Anforderungen (z. B. von Fachbereichen, Kunden, Regulatoren). Durch hohe Vielfalt und Dynamik entsteht **diffuse Wahrnehmung** über die Informationssystemlandschaft, auf deren Basis Entscheidungen getroffen werden, die wiederum zu einer Erhöhung der IT-Komplexität führen können. (SCHUH ET AL. 2017e, S. 61)

Aus dieser Komplexitätsdefinition leitet sich das **Ziel der Redundanzfreiheit** der Architekturelemente ab, um die Vielfalt der Elemente besser zu beherrschen (s. LEMKE ET AL. 2017, S. 330). Redundanzfreiheit kann sich dabei auf verschiedene Ebenen der Architektur beziehen. Auf Ebene der Informationssysteme sollte pro Systemkonzept möglichst nur ein System vorhanden sein (s. Kapitel 6.4.1). Auf Ebene der Daten sollte klar geregelt sein, welches Informationssystem das führende System für jedes Datum

ist. Gleichzeitig wird auf technischer Ebene Redundanz benötigt, um die Ausfallsicherheit der technischen Elemente zu erhöhen. Solche bewusst erzeugte Redundanz ist vom Ziel der Redundanzfreiheit ausgenommen.

Weiterhin leitet sich aus der IT-Komplexitätsdefinition das **Ziel der Standardisierung** ab, das hilft, auf die hohe Dynamik zu reagieren (s. URBACH U. AHLEMANN 2016, S. 128-131). Auch Standardisierung kann in verschiedenen Dimensionen erreicht werden, beispielsweise in der technischen Dimension, indem Hardwarekomponenten im Unternehmen standardisiert werden, um den Serviceaufwand gering zu halten. In der Prozessdimension kann eine Standardisierung beispielsweise durch Standardisierung der Businessprozesse und deren IT-Unterstützung vorgenommen werden. Standardisierung ist jedoch nur bis zu einem gewissen Grad sinnvoll, wie das „Complexity Cross“ von SCHNEBERGER U. MCLEAN (s. Kapitel 3.5) zeigt: Aus zu hoher Standardisierung folgt eine Zentralisierung von Funktionalitäten in Standard-Informationssysteme und damit eine Komplexitätssteigerung bei den Einzelkomponenten, wie sie bei früher üblichen Mainframe-Systemen der Fall war (s. SCHNEBERGER U. MCLEAN 2003, S. 219).

Ferner hat die Informationssystem-Architektur dem Zweck zu dienen, das Geschäftsmodell und damit die **Fachbereiche optimal zu unterstützen**. Die Architektur muss sich daher an die Kernprozesse des Unternehmens anpassen (s. RICHTER ET AL. 2008, S. 21; FOTH 2016, S. 11) und sich an der Strategie des Unternehmens ausrichten (s. FOTH 2016, S. 39). Nur so kann ein optimales Business-IT-Alignment erreicht werden.

Auch bei Lastspitzen, die vor allem bei allen zyklischen Geschäftsmodellen entstehen, die auch bei produzierenden Unternehmen anzutreffen sind, muss die Informationssystem-Architektur funktionsfähig bleiben. Um sich diesem und den sich ändernden Ansprüchen der Fachbereiche gut anpassen zu können, muss eine Informationssystem-Architektur zudem **skalierbar beziehungsweise elastisch** sein (s. HABERFELLNER ET AL. 2015, S. 187; URBACH U. AHLEMANN 2016, S. 133).

Zuletzt gilt als allgemeines Ziel, anhand dem Informationssystem-Architekturen angepasst werden müssen, auch, eine **sichere, kostengünstige, digitale Arbeitswelt für die Anwender** des Unternehmens bereitzustellen. Letzteres wird bei dem steigenden Wert digitaler Informationen weiter wichtiger und daher künftig auch gesetzlich gefordert (s. URBACH U. AHLEMANN 2016, S. 133). Das Ziel, kostengünstig zu sein, ist nach wie vor auch im Jahr 2017 in den Top-Drei der wichtigsten Ziele von IT-Abteilungen enthalten (s. HEIMANN ET AL. 2017, S. 13).

Diese allgemeinen Ziele, welche von einer Informationssystem-Architektur erfüllt werden müssen, werden in den vier Sichten *Technik, Organisation, Prozesse* und *Daten* in den folgenden Abschnitten konkretisiert.

Um eine Architektur zu gestalten, empfehlen HABERFELLNER ET AL. das Vorgehen „vom Groben zum Detail“ (s. Kapitel 4.2.2; HABERFELLNER ET AL. 2015, S. 185). Diesem Vorgehen folgt auch die vorliegende Arbeit. Da deren Umfang begrenzt ist, muss der **De-**

tailgrad der folgenden Architekturbeschreibungen auf einem hohen Niveau verbleiben, es wird jedoch jeweils auf vertiefende Literatur verwiesen. Die Architekturgestaltung umfasst zudem die Aufgabe der Dokumentation (s. LEMKE ET AL. 2017, S. 324). Die Nutzung von entsprechenden Darstellungsformen, wie sie beispielsweise in Kapitel 6.7 vorgestellt wurden, ist daher von großer Wichtigkeit für die Kommunikation der Ergebnisse solcher Strategieprozesse.

7.1.1 Zielinformationssystem-Architektur in der Dimension *Technik*

In der Dimension *Technik* werden die Eigenschaften der Zielinformationssystem-Architektur anhand des in Kapitel 6.5.1 definierten Ebenenmodells strukturiert (s. Abbildung 7-2). Es sei an dieser Stelle darauf hingewiesen, dass hier nicht auf die *Informationstechnologiearchitektur* im Speziellen eingegangen wird. Für Themen wie die Gestaltung von Netzwerkarchitekturen (vgl. OSTERHAGE 2017, S. 103-152), Storage-Konzepten (vgl. MÜLLER, K.-R. 2015, S. 390-402) oder virtualisierten Serverarchitekturen (vgl. BENDEL ET AL. 2015, S. 411-427) sei auf entsprechende Fachliteratur verwiesen. Es wird vielmehr die Informationssystemansicht eingenommen, wie sie in Kapitel 6.1 vorgestellt wurde.

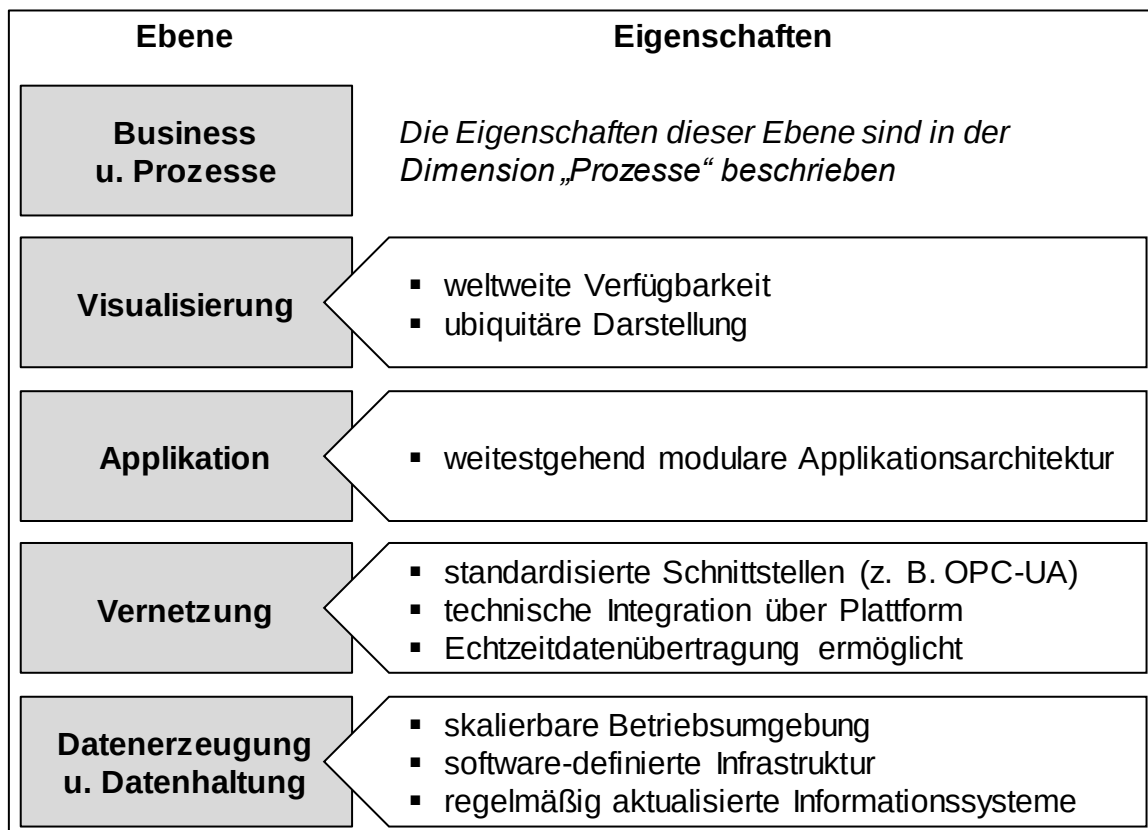


Abbildung 7-2: Eigenschaften der Zielinformationssystem-Architektur in der Dimension *Technik* (eigene Darstellung)

Auf der untersten Ebene der **Datenerzeugung** und Datenhaltung ist in einer Zielinformationssystem-Architektur eine **skalierbare Betriebsumgebung** umgesetzt. Diese ermöglicht Flexibilität bei der Speichernutzung sowie von Rechenkapazität, Lizenzen, Nutzerzahlen und Übertragungsbandbreiten. Mit einer solchen Architektur ist ein Wachsen und Schrumpfen mit den sich ändernden Anforderungen des Unternehmens

möglich. Die technische Umsetzung erfolgt über einen hohen Anteil an Cloud-Infrastruktur mit entsprechenden Dienstleistern und Verträgen, die diese Flexibilität ermöglichen (s. URBACH U. AHLEMANN 2016, S. 134).

Der Teil der **Infrastruktur**, der nicht an Cloud-Dienstleister ausgelagert ist, **ist software-definiert** (s. Kapitel 5.1.1). Dies ermöglicht nicht nur eine vereinfachte Konfiguration der Hardwarekomponenten, sondern man benötigt auch geringere Aufwände bei Konfigurationsänderungen. Somit sind aufwandsarm schnell Anpassungen möglich, wenn diese beispielsweise aufgrund von technischen Bedrohungslagen oder Organisationsanpassungen nötig sind.

Um mit den aktuellen technischen Entwicklungen mitzuhalten, sind **regelmäßige Aktualisierungen der Informationssysteme** notwendig. Wird nicht regelmäßig aktualisiert, sind die technischen Sprünge zwischen dem genutzten Stand und dem des neuen Updates sehr hoch. Dadurch entstehen nicht nur technische Risiken, sondern auch für die Anwender ein hoher Aufwand zum Umlernen. Bei betrieblichen Informationssystemen wird in absehbarer Zeit das Ziel täglicher Aktualisierungen noch nicht erreicht werden (s. Kapitel 6.8.5), trotzdem ist der evolutionäre Ansatz insgesamt dem revolutionären vorzuziehen.

Der evolutionäre Änderungsansatz führt allerdings auch zu ständigen Aufwänden der Abstimmung der Datenübertragung zwischen verschiedenen Informationssystemen. Um dies zu vereinfachen, verfügen Informationssysteme in der Zielarchitektur auf Ebene der **Vernetzung** daher über **standardisierte Schnittstellen** wie OPC-UA (s. Kapitel 2.1.4). Darüber ist ein einfacher und flexibler Datenverkehr sichergestellt.

Die in der Zielarchitektur eingeführte software-definierte Plattform dient aus technischer Sicht auch als Bussystem. In dieser Funktion übernimmt sie die **Datenübertragung zwischen einem großen Teil der Informationssysteme** – vor allem bei solchen, die zuvor nicht oder nur schlecht integriert waren. Existierende Schnittstellen zwischen zentralen Informationssystemen, wie beispielsweise zwischen dem ERP- und dem CRM-System, können jedoch bestehen bleiben, da diese oft hochintegriert sind. Ein Nachbau solcher Integrationslösungen in der software-definierten Plattform würde nur zusätzlichen Aufwand, dem Unternehmen aber keine Vorteile bringen. Neue Informationssysteme sollten jedoch möglichst über die Plattform vernetzt werden, um den Lock-in-Effekt möglichst klein zu halten und eine flexible Architektur zu erhalten.

Insgesamt verfügt die Zielarchitektur über ein **abgestimmtes Schnittstellenkonzept**. Darin ist beschrieben, wie die Grundsätze der Vernetzung des Unternehmens definiert sind. Abgeleitet ist dies vom Datenmodell des Unternehmens (s. Kapitel 7.1.4). Ziel dieses Schnittstellenkonzepts ist ein möglichst geringer Schnittstellenumfang, um der Forderung nach Komplexitätsreduktion nachzukommen (s. BECKER U. ROSEMAN 1998, S. 118-120).

Das Schnittstellenkonzept beantwortet auch die Frage nach **Echtzeitzugriffsmöglichkeiten**. Dort, wo Anwendungsfälle der software-definierten Plattform es benötigen

und wo es betriebswirtschaftlich sinnvoll ist, werden Schnittstellen echtzeitfähig umgesetzt. Ein **Beispiel einer solchen Architektur** zeigt, beziehend auf das in Kapitel 6.7 eingeführte *Beispielunternehmen I*, Abbildung 7-3. Die dort mit einem Sternchen gekennzeichneten Systeme sind, im Gegensatz zur Abbildung 6-15, neu eingeführt; fast alle Schnittstellen sind nun direkte Schnittstellen generischer Art.

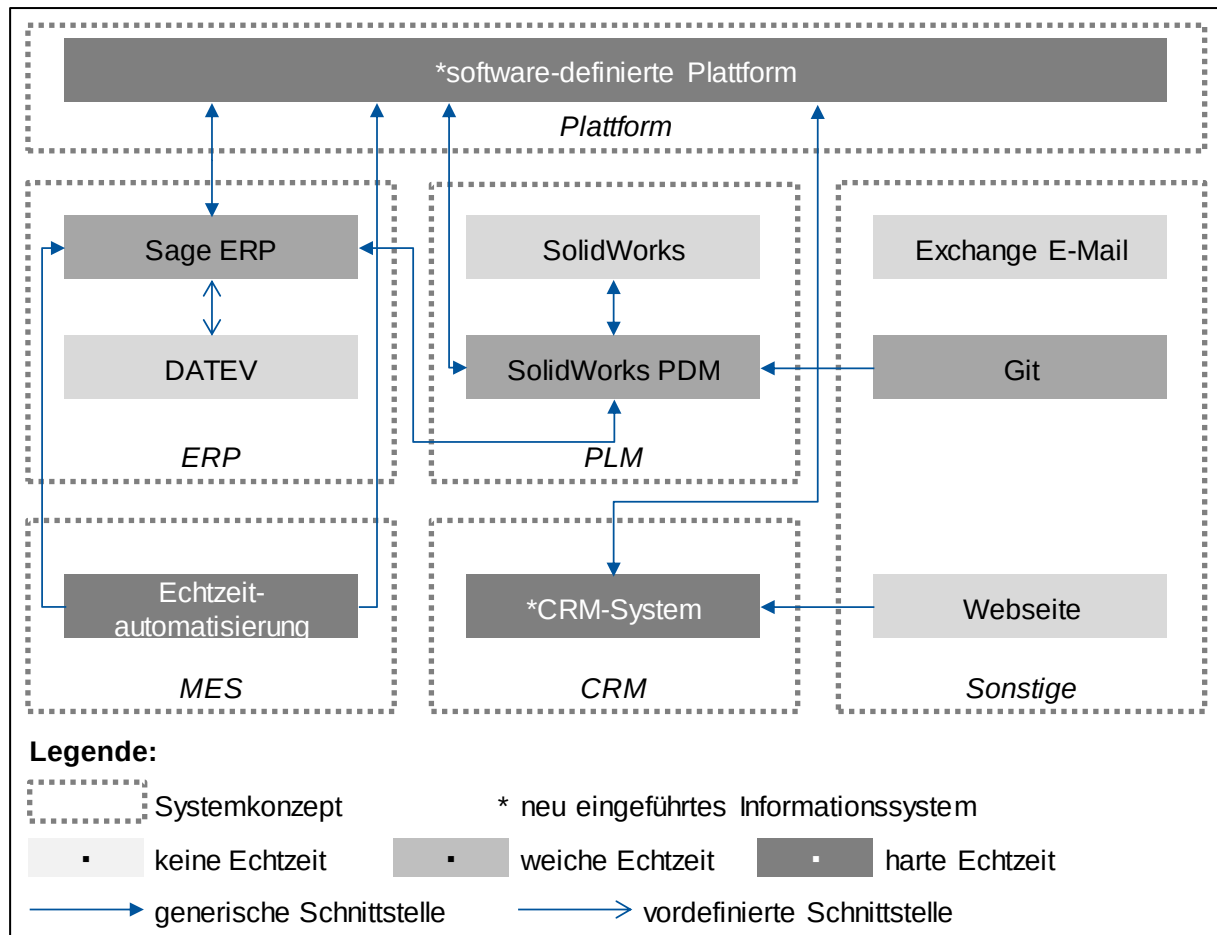


Abbildung 7-3: Beispielhafte Darstellung einer Zielinformationssystem-Architektur für Unternehmen I mit Fokus auf Schnittstellen (eigene Darstellung)

Aus gesamtarchitektonischer Sicht auf Ebene der **Applikation** wäre die Umsetzung der Informationssystem-Architektur als SOA sinnvoll (s. LANGLINAIS U. FORBES 2006, S. 8A; URBACH U. AHLEMANN 2016, S. 131). Eine solche komplett modularisierte Architektur ist jedoch für Unternehmen – im Speziellen auch produzierende – nur mit hohem Aufwand umsetzbar (s. PARKER ET AL. 2016, S. 57). Trotzdem ist die Bildung von **abgeschlossenen Modulen im Sinne der Komplexitätsreduktion** zielführend. So sind in der Zielarchitektur beispielsweise Informationssysteme *komplett* in die Cloud ausgelagert und verbleiben nicht teilweise in der Betriebsverantwortung der IT-Abteilung des Unternehmens.

Auf der Ebene der **Visualisierung** ist die Forderung von **Ubiquität in der Zielarchitektur** erfüllt (s. URBACH U. AHLEMANN 2016, S. 132). Dies bezieht sich zuerst auf die **weltweite Verfügbarkeit** der Informationssysteme und der darin enthaltenen Datenbestände. Über entsprechende technische Lösungen, wie jederzeit nutzbare VPN-Verbindungen und Endgeräte mit Mobilfunkmodul für die Mitarbeiter, ist der gesicherte

Datenzugriff jederzeit möglich. Als zweites bezieht sich der ubiquitäre Zugriff auch auf die **Endgeräteunabhängigkeit**. Alle Informationssysteme in der Zielarchitektur können entweder über Apps oder über Webbrowser aufgerufen werden, weil sie beispielsweise über in HTML 5 geschriebene, responsive Oberflächen verfügen (vgl. PREIM U. DACHSELT 2015, S. 94-96; HOFMANN 2016). Eine Abhängigkeit von Notebooks, als einzig möglichem mobilen Endgerät, ist nicht mehr vorhanden.

Die Ebene **Business und Prozesse** ist hier ausgelassen, da sie sich nicht auf die technische Dimension bezieht, sondern auf die Unternehmensprozesse, die in Kapitel 7.1.3 beschrieben sind.

7.1.2 Zielinformationssystem-Architektur in der Dimension *Organisation*

In der Zielarchitektur sind die **Rollen**, die damit verbundenen Aufgaben und das Zusammenarbeitsmodell der IT-Abteilung, der Fachbereiche und der Anwender im Gegensatz zum klassischen Verständnis **neu definiert**. Dahinter liegen Denkmodelle wie das *Architectural Thinking* von WINTER oder das *St. Galler Management-Modell* von RÜEGG-STÜRM U. GRAND (s. WINTER 2014, S. 361-364; RÜEGG-STÜRM U. GRAND 2017, S. 149-164). Abbildung 7-4 gibt einen Überblick dieser veränderten Rollendefinitionen anhand der organisatorischen Eigenschaften der Zielinformationssystem-Architektur.

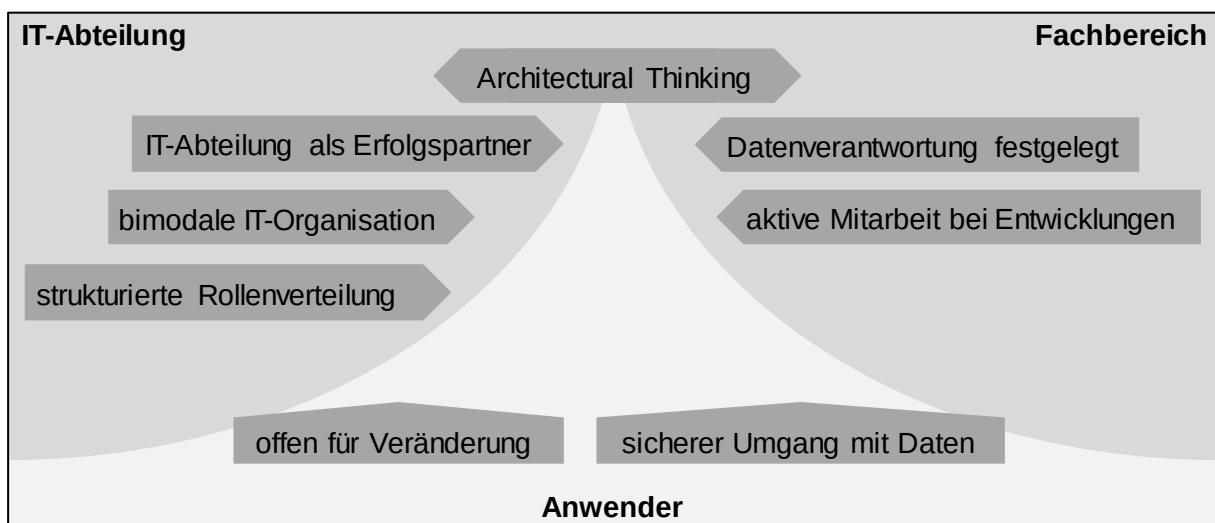


Abbildung 7-4: Eigenschaften der Zielinformationssystem-Architektur in der Dimension *Organisation* (eigene Darstellung)

Im *Architectural Thinking* wird vorgeschlagen, dass IT-Abteilung und Fachbereiche **Informationssystem-Architekturentscheidungen gemeinsam treffen**. Dies wird erreicht, indem bei den Fachbereichen ein Verständnis für die Ziele des Architekturmanagements wie Redundanzarmut und Anpassbarkeit eingeworben wird (s. WINTER 2014, S. 363; LEMKE ET AL. 2017, S. 325). Ein auf IT-Governance basierendes Entscheidungssystem, bei dem die IT-Abteilung als restriktive Einheit auftritt, wird so nicht mehr benötigt. Vielmehr können Entscheidungen sowohl von Seiten der IT-

Abteilung wie auch von einem Fachbereich initiiert und die Architektur so auf Basis gemeinsamer Ziele weiterentwickelt werden.⁴⁵

In der Zielarchitektur kann die **IT-Abteilung** damit **als Erfolgspartner** (s. *Architektur-typ V* in Kapitel 6.8.5) agieren. Sie hat dabei das Ziel, das Geschäftsmodell des Unternehmens durch Bereitstellung der richtigen Informationstechnologie weiterzuentwickeln. Das kann beispielsweise dadurch geschehen, dass sie neue Technologien, wie beispielsweise ortsabhängige Inhalte für Apps, für das Unternehmen evaluiert, testet und umsetzt und damit dem Unternehmen neue Verkaufskanäle bereitstellt.

Um so auftreten zu können, befasst sich die IT-Abteilung nur noch in geringem Umfang mit Commodity-Themen wie Hardwareanschaffung, Netzwerkkonfiguration oder dem Serverbasisbetrieb (s. FOTH 2016, S. 31). Solche Themen sind heute so standardisiert, dass sie entweder an spezialisierte Dienstleister abgegeben oder komplett in die Cloud verlagert werden können (s. Kapitel 7.1.1). Stattdessen treibt die IT-Abteilung Digitalisierungsprojekte voran und entwickelt so die Informationssystem-Architektur weiter.

Für diese Weiterentwicklung ist die **IT-Abteilung bimodal aufgestellt** (s. BERESFORD U. FENN 2016). In einer solchen Organisation, auch als *IT der zwei Geschwindigkeiten* bezeichnet, befasst sich ein Teil mit der Weiterentwicklung der Standardsysteme, die oft einen längeren Entwicklungszyklus benötigen. Der andere Teil kümmert sich um schnelle Anpassungen, beispielsweise von Apps und Datenauswertungen, die Teil der software-definierten Plattform sind (s. Kapitel 5.6.3). Dazu setzt das Unternehmen auf agile Entwicklungs- und Einführungsmethoden, welche eine enge Einbeziehung des Fachbereichs erfordern (vgl. HANSCHKE 2017).

⁴⁵ Hierzu kann EAM-Software genutzt werden. Die Innogy SE verwendet beispielsweise die Software „LeanIX“ und konnte eine deutliche Verbesserung der Zusammenarbeit mit Fachbereichen feststellen.

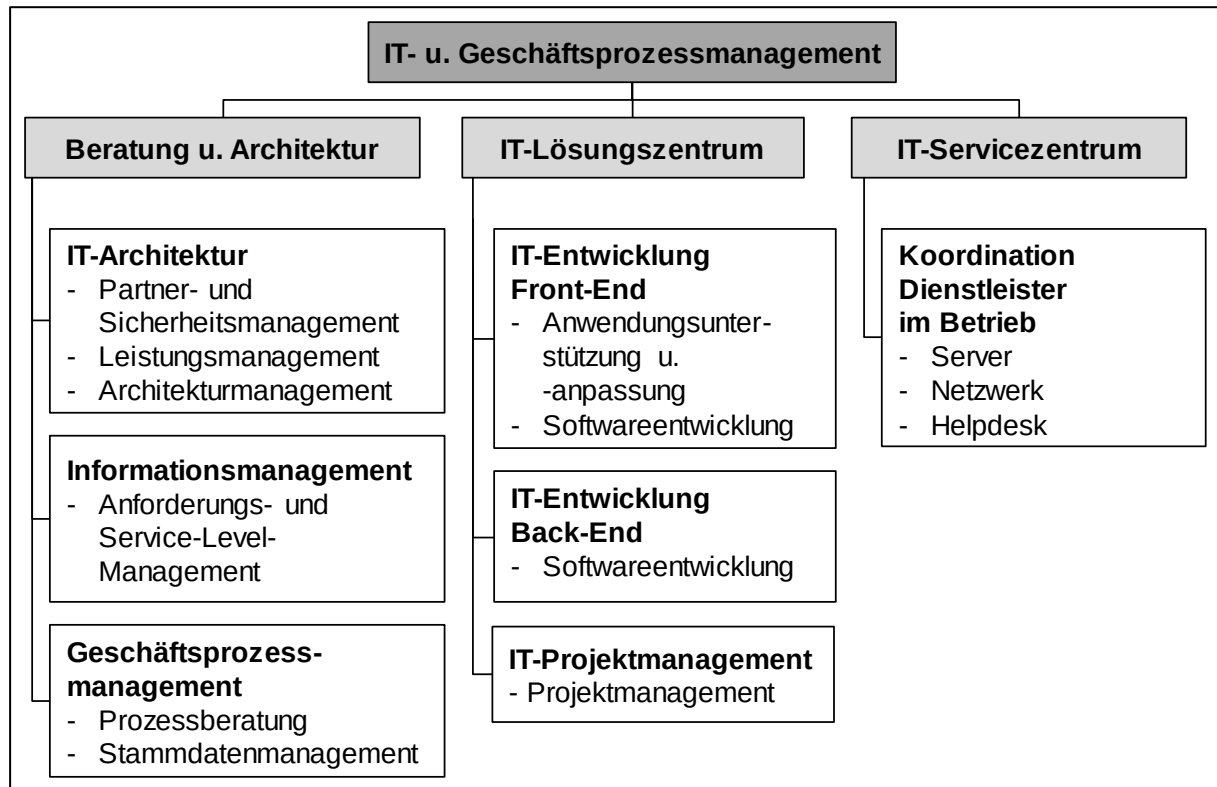


Abbildung 7-5: Beispiel einer Organisationsstruktur einer IT-Abteilung in der Zielarchitektur (eigene Darstellung i. A. a. FOTH 2016, S. 18)

Eine **organisatorische Aufteilung der IT-Abteilung** in der Zielarchitektur wird anhand von Rollen der Beratung und der strategischen Architekturentwicklung, der Entwicklung von Apps und technischen Lösungen sowie des IT-Service zur Anwenderunterstützung vorgenommen. Ein mögliches Organigramm zeigt Abbildung 7-5. Bei großen Unternehmen mit vielen Standorten sollte eine solche Organisationsstruktur als gute Mischung zwischen zentraler und dezentraler Organisation umgesetzt werden. So wurden beispielsweise im oben vorgestellten *Unternehmen III* lokale IT-Koordinatoren mit eigener Entscheidungskompetenz eingeführt, um die Interessen der dezentralen Standorte zu vertreten.

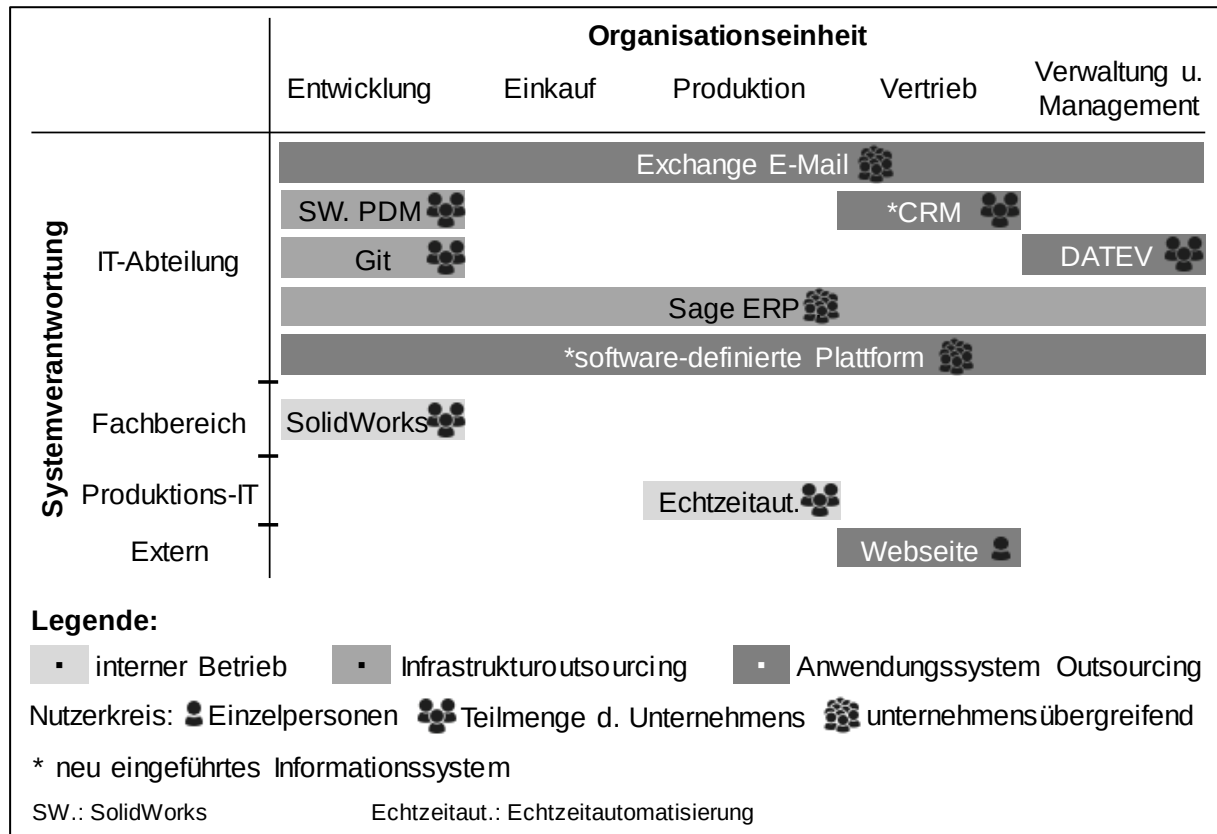


Abbildung 7-6: Beispielhafte Darstellung einer Zielinformationssystem-Architektur für Unternehmen I mit Fokus auf organisatorische Merkmale (eigene Darstellung)

Auch außerhalb der IT-Abteilung sind Rollen und Verantwortungen für den Fachbereich betreffende Teile von Informationssystemen in der Zielarchitektur definiert: So ist klar geregelt, wer die **Verantwortung** für welche **Unternehmensdaten** hat, zudem ist eine unternehmensübergreifende Rolle, welche sich um die Harmonisierung und Standardisierung von Stammdaten, Produktdaten und -datenstrukturen kümmert, etabliert (s. Kapitel 5.6.3). Neben den Datenstrukturen ist ebenfalls eine Rolle, welche die **Prozessstandardisierung** im Unternehmen vorantreibt, in der Struktur verankert. Diese Rolle kümmert sich darum, die notwendige regelmäßige Weiterentwicklung von Wertschöpfungsprozessen des Unternehmens voranzutreiben (s. RÜEGG-STÜRM U. GRAND 2017, S. 118) und deren Abbildung in Informationssysteme sicherzustellen (s. Kapitel 7.1.3). Die Verantwortung über einzelne Informationssysteme ist dabei transparent geregelt. Ein großer Teil der Informationssysteme ist in der Hand der IT-Abteilung. Sehr spezifische Anwendungen mit wenigen Anwendern werden von den Fachbereichen selbst, unter Rücksprache mit der IT-Abteilung, betrieben. Beispielhaft ist diese Verantwortungsteilung in der Abbildung 7-6 dargestellt, welche auf Basis des Ist-Zustands aus Kapitel 6.7.2 eine mögliche Zielarchitektur darstellt.

Die Anwender der Informationssysteme sind in der Zielarchitektur **offen für Veränderungen**, weil diese früh und transparent von den Verantwortlichen kommuniziert werden. Außerdem können die Anwender selbst ihre Anliegen zu Veränderungen kundtun (s. Kapitel 5.6.3); diese Anliegen werden bearbeitet und schnell durch die IT-Abteilung

und ihre Dienstleister umgesetzt. So entsteht eine erhöhte Zufriedenheit der Mitarbeiter des Unternehmens mit dessen Informationssystem-Architektur.

Weiterhin sind die Mitarbeiter im **Umgang mit Daten geschult**. Dadurch können sie den Wert, der in den Daten steckt, besser einschätzen und diese für das Unternehmen gewinnbringend einsetzen. So können sie beispielsweise unter Nutzung der software-definierten Plattform die für ihren Prozess notwendigen Daten selbst konfigurieren. Die software-definierte Plattform unterstützt die Mitarbeiter dabei, indem sie die Datenwünsche der Mitarbeiter erlernt und so auch andere relevante Daten vorschlagen kann.

7.1.3 Zielinformationssystem-Architektur in der Dimension Prozesse

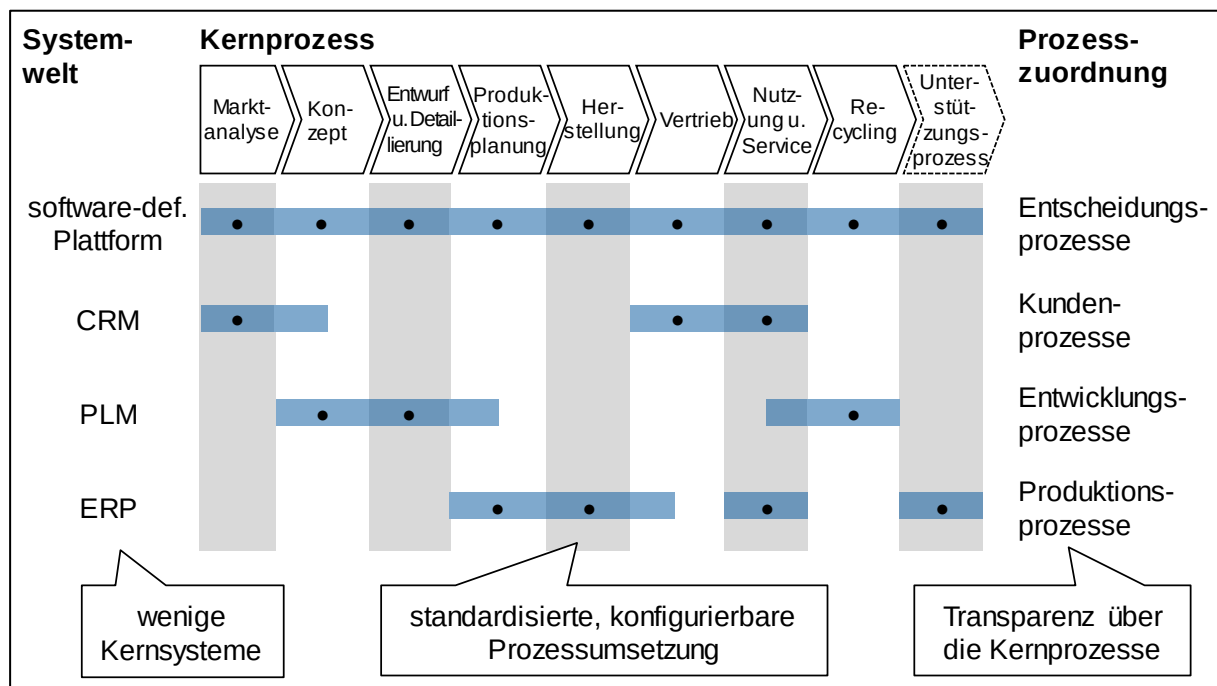


Abbildung 7-7: Eigenschaften der Zielinformationssystem-Architektur in der Dimension Prozesse (entnommen aus HOFFMANN U. HEIMES 2018, S. 1000)

In der Dimension *Prozesse* werden Eigenschaften der Zielinformationssystem-Architektur beschrieben, die sich auf die Nutzung von Informationssystemen in den Wertschöpfungsprozessen eines Unternehmens beziehen. **Grundsätzliche Ziele** hierbei sind die Erhöhung der **Architekturkonsistenz** (S. LEMKE ET AL. 2017, S. 330) sowie die einfachere **Rekonfigurierbarkeit** der Einzelprozesse, wie sie im Rahmen der Industrie 4.0 von verschiedenen Autoren, speziell im Bereich der Produktion, gefordert wird (S. DEUSE ET AL. 2015, S. 103; HEES 2017, S. 4-5). In Abbildung 7-7 ist die Zielarchitektur in einem Übersichtsbild zusammengefasst. Sie zeigt die im Folgenden beschriebenen Eigenschaften der *Prozesstransparenz*, *geringen Informationssystemvielfalt* und *Prozessagilität*.

Als Voraussetzung der oben genannten grundsätzlichen Ziele dient der Zielarchitektur die **Transparenz** über Anwendungsfälle und Prozesse. Mit Anwendungsfällen sind die Nutzungsszenarien von Informationssystemen gemeint; sie beschreiben, welche

Funktionen durch die Informationssysteme hauptsächlich unterstützt werden. Folgende Themen sind beispielsweise darin beschrieben:

- Der aktuelle Status der Maschinenverfügbarkeit in der Produktion kann über die software-definierte Plattform abgerufen werden.
- Alle für die Vertriebsabteilung relevanten Informationen über Kunden finden sich im CRM-System.
- Im PLM-System werden von allen Produkten und deren Baugruppen die Versionsstände der Entwicklung nachgehalten. (s. HOFFMANN U. HEIMES 2018, S. 999)

Daneben ist die **Prozesstransparenz** ein entscheidendes Merkmal der Zielinformationssystem-Architektur. Die Relevanz dessen wird im Kontext einer komplexitätsarmen Informationssystem-Architektur in der Literatur verschiedentlich beschrieben (s. Kapitel 3.5). ROSS ET AL. heben die Wichtigkeit der Modellierung aller Unternehmensprozesse hervor, ARMAS ergänzt den Zusammenhang zu Informationen, mit denen Mitarbeiter arbeiten (s. ROSS ET AL. 2006; ARMAS 2011, S. 117-127). RICHTER ET AL. betonen die Verbindung zwischen Unternehmensprozessen und der Informationssystem-Architektur (s. RICHTER ET AL. 2008, S. 21). Ein wichtiger Aspekt der Transparenz der Zielarchitektur ist die Kongruenz zwischen definierten und in Informationssystemen umgesetzten Prozessen. Prozess- und Qualitätsmanagement, beispielsweise gemäß der Norm ISO 9001 (vgl. DIN EN ISO 9001), wird damit Realität in produzierenden Unternehmen, weil die definierten Prozesse den gelebten Prozessen entsprechen.

Zur Umsetzung der Prozesse setzen Unternehmen in der Zielarchitektur nur **wenige zentrale Informationssysteme** ein. Diese sind so weit wie möglich als Standardanwendungen etabliert und nur dort angepasst, wo die Wettbewerbsdifferenzierung dies notwendig macht. Um die geringe Systemvielfalt zu bewahren, wird für spezifische Anforderungen der Fachbereiche nicht zwingend das zentrale Informationssystem angepasst, sondern entsprechend der Anforderung entschieden: Bei systemübergreifender Datennutzung oder wenig umfangreichen Anforderungen kann eine App auf der software-definierten Plattform bereitgestellt werden. So wurde beispielsweise bei der e.GO Mobile AG in deren software-definierter Plattform eine App zum Management und zur Nachverfolgung der Fahrzeuggewichte während der Entwicklungsphase eingeführt (s. Kapitel 9.2.3). Diese App greift über die Plattform auf das angeschlossene PLM-System zu und wertet Gewichtsinformationen daraus aus (s. Kapitel 5.3.6). Bei umfangreichen Anforderungen wird geprüft, ob diese in einem spezifischen Informationssystem umgesetzt werden können. Zentrales Entscheidungskriterium neben den fachlichen Anforderungen ist dabei die Integrationsfähigkeit dieser Anwendung in die bestehenden zentralen Informationssysteme. Im *Beispielunternehmen V* wurde beispielsweise *PSI metals* in Ergänzung zum zentralen *SAP-System* eingeführt, um die spezifischen Anforderungen der Metallproduktion abbilden zu können. Das *PSI metals* verfügt jedoch über eine zertifizierte Schnittstelle zu *SAP*, sodass dort eine nahtlose

Integration erreicht werden konnte (s. Kapitel 6.8.5). Handelt es sich bei der Anforderung um eine einfache Konfigurationsänderung, so wird diese im zentralen Informationssystem vorgenommen oder bei dessen nächsten Releasestand umgesetzt.

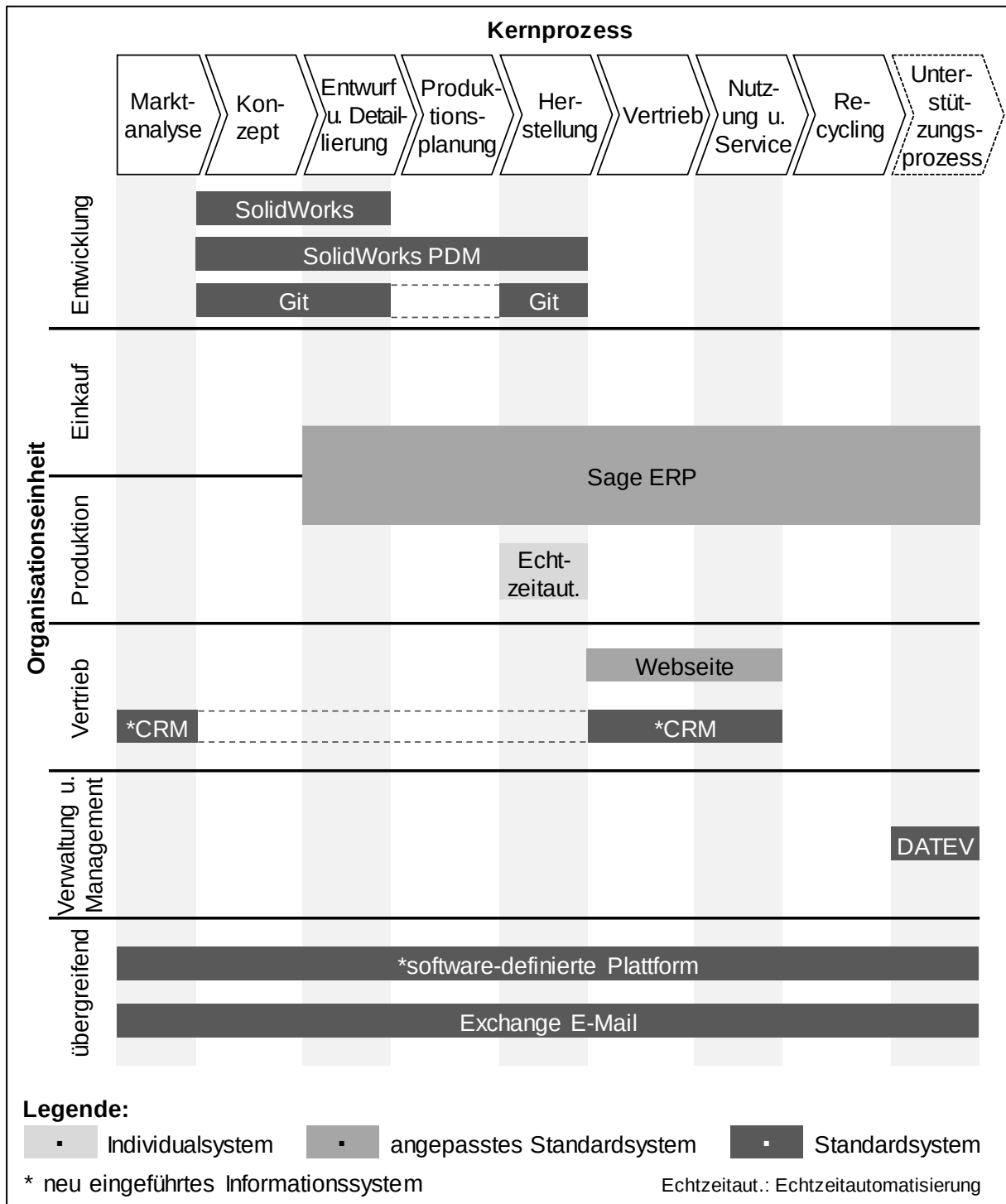


Abbildung 7-8: Beispielhafte Darstellung einer Zielinformationssystem-Architektur für Unternehmen I mit Fokus auf prozessuale Merkmale (eigene Darstellung)

Die zentralen und datenführenden Informationssysteme, die bei produzierenden Unternehmen in der Zielinformationssystem-Architektur eingesetzt werden, sind in der Abbildung 7-7 zusammengefasst. Abbildung 7-8 zeigt eine beispielhafte Übertragung dieser Architekturprinzipien auf das *Beispielunternehmen I*. Im Einzelnen sind dies:

- Eine software-definierte Plattform für die in den vorherigen Kapiteln beschriebenen datengetriebenen Anwendungsfälle.
- Ein CRM-System zur Ablage der Kundendaten und Umsetzung der kunden-zentrierten Prozesse.
- Ein PLM-System zur Strukturierung und Versionierung der Produktdaten.
- Ein ERP-System für alle Prozesse, Stamm- und Bewegungsdaten der Produktherstellung und Logistik.

Je nach eingesetzten Softwareprodukten werden zusätzlich benötigt:

- Ein separates Human-Resource-System als zentrales Informationssystem über Mitarbeiterdaten wie Personalakten, Gehälter, Überstunden und ähnliches, falls dies nicht als Modul des ERP-Systems geliefert wird.
- Ein separates Buchhaltungssystem, falls dies kein Modul des ERP-Systems ist.
- Wenn viel mit Dokumenten gearbeitet wird: ein ECM-System, welches das Dokumentenhandling im Hintergrund steuert.

Die genannten zentralen und datenhaltenden Informationssysteme sind dabei in die einzelnen **Unternehmensprozesse optimal integriert**. So weit wie möglich werden die Prozessschritte in den Informationssystemen durchgeführt, sodass keine Umwege über Excel-Exporte oder Papierdokumente notwendig sind. Im Speziellen ist die software-definierte Plattform für alle Anwender zugreifbar, nutzbar und in deren Entscheidungsprozesse integriert. Wird auf Basis der dort verfügbaren Informationen eine Entscheidung getroffen – beispielsweise die Anpassung eines Produktionsplans wegen prognostizierter Engpässe – werden diese Anpassungen direkt in die Umsysteme zurückgespielt. Die Prozesse, besonders die in der Plattform umgesetzten, sind einfach konfigurierbar und können damit sich verändernden Gegebenheiten angepasst werden.

Gemäß dem EAM-Gedanken geht die IT-Komplexität von den Unternehmensprozessen aus (S. SCHNEIDER U. MATTHES 2014, S. 696). Daher unterstützt eine möglichst **hohe Prozessstandardisierung** die Komplexitätsreduktion der Informationssystem-Architektur: Einheitliche Unternehmensprozesse ermöglichen einheitliche Informationssysteme, und Prozessvarianz wird in der Zielarchitektur nur dort zugelassen, wo das vom Kunden gewünscht ist und Wettbewerbsdifferenzierung ermöglicht (S. ALBAYRAK U. GADATSCH 2014, S. 682-683).

7.1.4 Zielinformationssystem-Architektur in der Dimension *Daten*

Mit Blick auf die Daten zeichnet sich die Zielinformationssystem-Architektur durch vereinheitlichte, strukturierte und hochwertige Datenstrukturen aus. Dies bezieht sich sowohl auf die Daten selbst (**Datenebene**) und die grundlegende **Datenstruktur**, als auch auf die darüberliegende, informationssystemimmanente Datenstruktur auf **Anwendungsebene**. Bei produzierenden Unternehmen ist bei letzterem vor allem auf die

Produktdatenstruktur zu achten. Abbildung 7-9 fasst diese Eigenschaften zusammen, die im Folgenden detailliert werden.

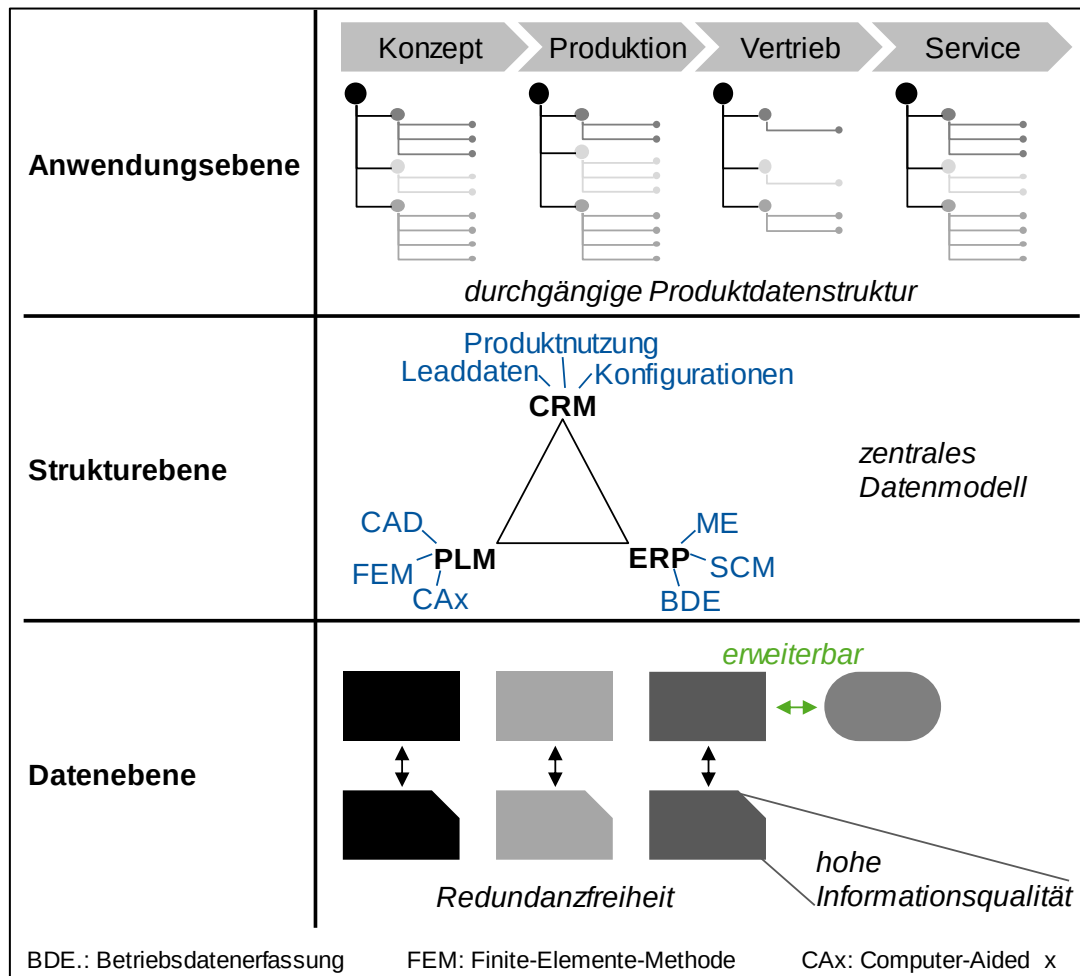


Abbildung 7-9: Eigenschaften der Zielinformationssystem-Architektur in der Dimension Daten (entnommen aus HOFFMANN U. HEIMES 2018, S. 1001)

Klarheit und Strukturiertheit der Daten beginnt bei der Qualität einzelner Datenklassen. In der Zielarchitektur sind daher die Datenbestände des Unternehmens einheitlich modelliert, angeglichen und semantisch harmonisiert (s. STAHL U. STAAB 2017, S. 24). Es ist also geklärt, in welchen Formaten, Detailgraden und Einheiten die Datenklassen vorliegen sollen, um sie gemeinsam nutzen zu können. Wo möglich, wird dabei auf existierende Normen wie den eCI@ss-Standard zurückgegriffen (vgl. CWA 16525). Die **Datenbestände** haben sodann eine **hohe Qualität** in Bezug auf die Dimensionen Vollständigkeit, Fehlerfreiheit, Aktualität und Zugänglichkeit (s. Kapitel 6.5.2). Diese hohe Qualität rührt einerseits daher, dass die Mitarbeiter des Unternehmens im Umgang mit Daten geschult sind und täglich mit den Daten arbeiten, andererseits daher, dass ein zentrales Datenmanagement als Unternehmensfunktion eingeführt wurde (s. Kapitel 7.1.2; FOTH 2016, S. 82-83).

Ein wichtiger qualitätsunterstützender Faktor ist die **Redundanzfreiheit der Stamm- und Bewegungsdaten**. Für jedes Datum ist eindeutig festgelegt, in welchem Informationssystem dessen Original gespeichert ist und wo nur Kopien vorhanden sind. Eine komplett datenkopie-freie Architektur ist jedoch technisch nicht sinnvoll umsetzbar, da

bei fehlenden Caching-Mechanismen die Netzwerke überlastet wären und damit Performanceeinbußen entstünden. Eine zentrale Rolle spielt in dieser Architektur die software-definierte Plattform, da dort die Zusammenhänge zwischen den Datenstrukturfragmenten der einzelnen Informationssysteme hergestellt werden. Hierdurch erhalten die Anwender Transparenz darüber, welche Daten in welchem Informationssystem zu finden sind. Sie können damit eigene Auswertungen erstellen, die sie für ihre Entscheidungsfindung benötigen. Die Regeln des Datenschutzes und der Datenschutzgrundverordnung werden, durch die software-definierte Plattform unterstützt, eingehalten (s. Kapitel 5.3.5),⁴⁶ indem beispielsweise nutzerspezifische Daten aus der Produktion anonymisiert oder pseudonymisiert werden.

Gleichzeitig ist die Datenstruktur derart gestaltet, dass weitere Datenpunkte außerhalb der zentralen Informationssysteme, beispielsweise aus dem Internet, von Maschinen oder von Produkten, einfach integrierbar sind. Diese **Erweiterbarkeit** ist und bleibt notwendig, da sich sowohl Datenerfordernisse als auch Datenangebot im Unternehmen ständig verändern.

Auf der Strukturebene ist ein **zentrales, unternehmensübergreifendes Datenmodell** in allen Informationssystemen umgesetzt (s. BECKER U. ROSEMAN 1998, S. 118). Abbildung 7-9 zeigt einen Ausschnitt eines solchen Datenmodells. Es ist in diesem Fall organisiert nach den drei Kernelementen *Produkt*, *Kunde* und *Auftrag*. Produktrelevante Daten, also solche, die das PLM-Konzept betreffen, sind beispielsweise Teilenummern, Stücklisten, CAD-Dateien oder Versionshistorien. Kundendaten entsprechend dem CRM-Konzept sind Leaddaten und die Abbildung der *Customer Journey*, Produktnutzungsdaten einzelner Kunden oder auch Konfigurationsdaten aus der Kundenbestellung. Zuletzt umfassen auftragsbezogene Daten alle Daten des ERP-Konzepts. Als zentrale Daten können daher Produkt- beziehungsweise Teilenummern, Kundennummern und Auftragsnummern angesehen werden. Mindestens eine dieser Nummern wird in jedem Kernprozess des Unternehmens genutzt, weswegen sich alle anderen Daten dazu in Bezug setzen lassen können. Ausgehend von diesen zentralen Elementen ist in der Zielarchitektur das Datenmodell netzartig aufgebaut. Es ist dabei auch geregelt, welche Wege Daten nehmen, indem Informationslogistikkonzepte für die Kernanwendungsfälle des Unternehmens ausgearbeitet sind (s. NIENKE ET AL., S. 242).

Aufbauend auf den oben genannten grundlegenden Strukturierungen für die Daten des Unternehmens sind in der Zielinformationssystem-Architektur die **Produktdatenstrukturen** entwickelt. Sie beachten dabei den gesamten Lebenszyklus des Produkts

⁴⁶ Der Einsatz des hier beschriebenen Modells unterstützt auch bei der Einhaltung aktueller Datenschutzvorschriften: Die DSGVO schreibt unter anderem vor, dass Personen das Recht haben, jederzeit Auskunft über die Daten zu erhalten, die über sie beispielsweise in Informationssystemen eines Unternehmens gespeichert sind (s. Artikel 15 Absatz 3 DSGVO). Solche Daten können in der Zielarchitektur anhand des einheitlichen Datenmodells über die software-definierte Plattform abgerufen werden.

und umfassen damit auch verschiedene Sichten unterschiedlicher Fachbereiche. Klassischerweise wird die Produktstruktur und davon abgeleitet die Produktdatenstruktur in der Entwicklungsabteilung erstellt und davon die Stücklisten für die Produktion, den Vertrieb und den Service abgeleitet, was oft zu Inkonsistenzen führt. Es existieren Ansätze, die Produktstruktur auch beispielsweise montage- oder instandhaltungsgerecht zu gestalten (s. MÜLLER 2013, S. 712-713; PAHL ET AL. 2013, S. 726); daran orientiert sich das Produktdatenmodell in der Zielarchitektur. Es folgt dabei der Empfehlung von MEYER ET AL., bei der Produktmodellierung zuerst ein ganzheitliches und übergreifendes Modell zu implementieren (s. MEYER ET AL. 2014, S. 413).

Dazu existiert in der Zielarchitektur ein *generisches* Modell als **Produktdatenreferenzstruktur**, das auf Ebene der Baugruppen eine Struktur definiert, die im Unternehmen standardisiert genutzt wird. Diese Struktur wird auf das jeweilige Produkt des Unternehmens, und spezifisch auf die unterschiedlichen Sichten des Vertriebs oder des Service-Bereichs, angepasst (s. RUDOLF U. SCHREY 2016a, S. 97). Wie in Abbildung 7-9 schematisch dargestellt, bleiben dabei jedoch die Beziehungen zwischen den Baugruppen erhalten. Weiterhin stellt die eindeutige Identifikation der Teile sicher, dass in den verschiedenen Stücklistensichten eine Referenz zu einzelnen Teilen bestehen bleibt. Um die Spezifika, beispielsweise einer Vertriebsstückliste, umzusetzen, ist die Stückliste dort auf Modulebene anders strukturiert als die Entwicklungsstückliste. Die oberste Ebene der Baugruppen und die unterste Ebene der Teile bleiben jedoch erhalten. Werden die Stücklisten instanziiert, entstehen Eins-zu-n-Beziehungen. Zu einem entwickelten Produkt kann es mehrere Varianten geben, die wiederum jeweils zu unterschiedlichen Zeiten gefertigt wurden und an unterschiedliche Kunden verkauft wurden. Es entsteht somit ein Baum an Stücklisten mit dem entwickelten Produkt als Wurzel. Zentral bei der Umsetzung eines solchen Modells ist die gemeinsame **Abstimmung aller Fachbereiche** zur Struktur und die Umsetzung der Struktur in den zentralen Informationssystemen.

7.2 Zusammenhänge zwischen Ist-Architekturtypen und Zielbeschreibung

Der in Kapitel 7.1 beschriebene Ziel-Zustand der Informationssystem-Architektur bei software-definierten Plattformen kann in einer Morphologie zusammengefasst werden. Dabei bilden die vier Sichten jeweils eine Kategorie und die beschriebenen Eigenschaften der Zielarchitektur die Merkmale. Um die Zusammenhänge der Ist-Architekturtypen aus Kapitel 6.8 in Verbindung mit dem Ziel-Zustand zu bringen, besitzt jedes Merkmal der Morphologie die generischen Ausprägungen *gering*, *mittel* und *hoch*.

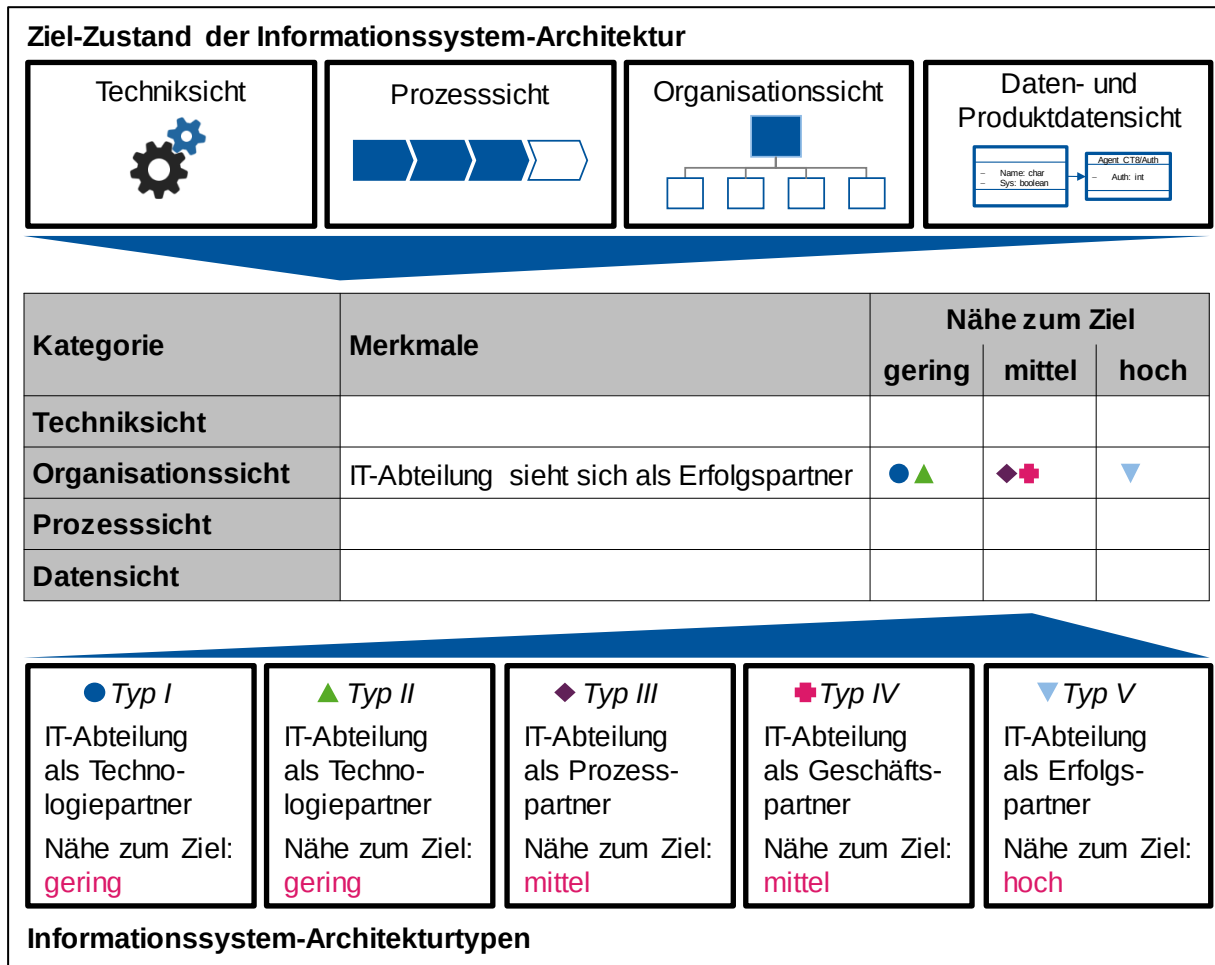


Abbildung 7-10: Vorgehen zur Herleitung der Zusammenhänge zwischen Ist- und Ziel-Zustand von Informationssystem-Architekturen (eigene Darstellung)

Auf Basis der Beschreibung der Architekturtypen kann damit **für jeden Architekturtyp dessen Abstand zur Zielarchitektur zusammengefasst dargestellt** werden. Abbildung 7-10 zeigt dieses Vorgehen anhand eines Beispiels auf: Das Merkmal der Zielarchitektur „IT-Abteilung sieht sich als Erfolgspartner“. Die Nähe zum Ziel wird hierbei formal-sachlogisch hergeleitet, indem jeweils der Aufwand abgeschätzt wird, der benötigt wird, um das geforderte Ziel zu erreichen. Die Abschätzung ist hierbei rein qualitativ auf Basis von Erfahrungen des Autors in Rücksprache mit Experten erfolgt (s. Anhang A.3).

In Bezug auf das oben gezeigte Beispiel ist die Nähe zum Zielzustand beim *Informationssystem-Architekturtyp I* als *gering* eingeschätzt. Dies liegt darin begründet, dass sich die IT-Abteilung bei diesem Architekturtypen als Technologiepartner sieht (s. Kapitel 6.8.1); eine Weiterentwicklung zum Erfolgspartner ist mit großem Aufwand verbunden, weil sie einen komplett veränderten Aufgabenumfang der IT-Mitarbeiter und daher beispielsweise entsprechende Schulungsmaßnahmen, veränderte Prozesse und erweiterte Personalressourcen nach sich zieht.

Entsprechend diesem Vorgehen wurden die fünf Informationssystem-Architekturtypen gegen alle einzelnen Merkmale der Zielinformationssystem-Architektur geprüft und die Ergebnisse der Prüfung in einer Tabelle festgehalten. Die Zusammenfassung dieser

Prüfung zeigt Abbildung 7-11, aggregiert in einer Darstellung für alle Architekturtypen. Die zentralen Aspekte dieser Abbildung werden im Folgenden erläutert.

Kategorie	Merkmal	Nähe zum Ziel		
		gering	mittel	hoch
Techniksicht	regelmäßig aktualisierte Informationssysteme	● ▲	◆	✚ ▼
	software-definierte Infrastruktur	●	▲	◆ ✚ ▼
	skalierbare Betriebsumgebung	●	▲	◆ ✚ ▼
	Echtzeitdatenübertragung ermöglicht	▲	● ◆ ✚ ▼	
	technische Integration über Plattform	● ▲ ◆	✚ ▼	
	standardisierte Schnittstellen	● ▲	◆	✚ ▼
	weitestgehend modulare Applikationsarchitektur	● ▲	◆ ✚	▼
	ubiquitäre Darstellung	● ▲ ◆ ✚	▼	
	weltweite Verfügbarkeit		● ▲ ◆ ✚ ▼	
Organisations-sicht	Architectural Thinking	● ▲	◆ ✚	▼
	IT-Abteilung sieht sich als Erfolgspartner	● ▲	◆ ✚	▼
	bimodale IT-Organisation	● ▲	◆ ✚	▼
	strukturierte Rollenverteilung	● ▲	◆ ✚	▼
	Datenverantwortung beim Fachbereich		● ▲ ◆ ✚ ▼	
	Mitarbeiter im Umgang mit Daten geschult	● ▲	◆ ✚ ▼	
Prozesssicht	Transparenz über Anwendungsfälle der IS	● ▲	◆	✚ ▼
	Kernprozestransparenz		● ▲ ◆ ✚	▼
	konfigurierbare Prozesse	● ▲ ◆	✚	▼
	wenige zentrale Informationssysteme	● ◆	▲	✚ ▼
	hohe Prozessstandardisierung	● ▲	◆ ✚	▼
Datensicht	hohe Informationsqualität	● ▲	◆ ✚ ▼	
	redundanzfreie Datenbestände	● ▲ ◆	✚	▼
	erweiterbare Datenmodelle	● ▲	◆ ✚ ▼	
	zentrales Unternehmensdatenmodell	● ▲ ◆	✚	▼
	durchgängige Produktdatenstruktur	●	▲ ◆ ✚ ▼	
Legende: ● Architekturtyp I ▲ Architekturtyp II ◆ Architekturtyp III ✚ Architekturtyp IV ▼ Architekturtyp V				

Abbildung 7-11: Zuordnung der Ist-Architekturtypen zur Zielbeschreibung (eigene Darstellung)

In der Kategorie der **Techniksicht** ist zuerst hervorzuheben, dass die **IT-Infrastruktur** bei vielen Unternehmen schon sehr nah am Ziel entwickelt ist. Dies ist nicht verwunderlich, weil ein großer Fokus der technischen Entwicklung der vergangenen Jahre auf diesem Thema lag. Durch die Einführung von Virtualisierungs- und Cloud-Techniken

und Vereinfachung der Konfiguration durch software-definierte Netzwerke und Speicher konnten Unternehmen große Fortschritte in diesem Bereich erzielen. Dagegen stellt die **Echtzeitdatenübertragung** Unternehmen noch vor Herausforderungen. In einzelnen, speziell dafür designten Systemen ist dies schon möglich; ein Echtzeitzugriff auf einen größeren Teil der Datenbestände ist jedoch aufgrund des hohen technischen Aufwands noch nicht umgesetzt. Dementsprechend konnte bei keinem der Architekturtypen eine große Nähe zum Ziel festgestellt werden.

Da software-definierte Plattformen in der hier beschriebenen Form noch in sehr wenigen Unternehmen im Einsatz sind, ist die **technische Integration über die Plattform** noch nicht umgesetzt. Eine durchdachte Schnittstellenarchitektur und aktuelle Informationssysteme unterstützen jedoch bei einer künftigen solchen Integration, weshalb die *Architekturtypen IV* und *V* eine mittlere Nähe zum Ziel-Zustand aufweisen. Dagegen ist eine **ubiquitäre Darstellungsmöglichkeit** der Inhalte bei den meisten Unternehmen nicht, oder nur für einzelne Datenbestände, möglich. Auch dies wird erst durch den Einsatz einer software-definierten Plattform gelöst, genauso wie auch die **weltweite Verfügbarkeit** von Informationen. Letzteres wird heute größtenteils über VPN-Lösungen umgesetzt, über die Mitarbeiter sicher Zugriff auf beispielsweise das ERP-System bekommen. „Unternehmens-Apps“, wie sie für einzelne Anwendungsfälle wie beispielsweise Reisekosten, schon im Einsatz sind, können den Zugriff jedoch noch vereinfachen.

Aus **Organisationssicht** hat die Organisation der IT-Abteilung einen hohen Anteil an der Entfernung zum Ziel-Zustand. Daher sind die *Architekturtypen I* und *II*, die auf einer IT-Organisationspositionierung als Technologiepartner basieren, bei vielen Merkmalen dieser Kategorie der Ausprägung *gering* zugeordnet, während der *Architekturtyp V* der Ausprägung *hoch* zugeordnet ist. Das liegt in den Kulturaspekten begründet, die mit der Selbst- und Fremdwahrnehmung der IT-Abteilung als ein bestimmter Typ verbunden sind. Umso mehr die **IT-Abteilung als Erfolgspartner** wahrgenommen wird, umso stärker wird sie auch in Architekturentscheidungen mit einbezogen. Eine solche IT-Abteilung kennt die im **Informationsmanagement wahrzunehmenden Aufgaben** (s. Kapitel 2.2.1) und verteilt sie entsprechend im Unternehmen – und strukturiert sich auch intern so, dass sie Kundenanforderungen schnell umsetzen kann.

Die Merkmale, die den Anwender in Betracht nehmen, sind von der Organisation der IT-Abteilung unabhängig zu sehen. So nehmen in allen fünf Architekturtypen die **Fachbereiche** nur teilweise **Verantwortung über ihre Daten** wahr. Dabei fehlt es beispielsweise am Verständnis für den Wert von Daten und dem zu deren Schutz notwendigen IT-Sicherheitslevel (s. ALPAR ET AL. 2016, S. 87-88). Auch der Blick über die eigenen Daten hinaus hin zu einer unternehmensweiten Datenorganisation wird in produzierenden Unternehmen in der Art selten umgesetzt. Wegen der noch niedrigen Datennutzung sind auch die **Mitarbeiter im Umgang mit Daten** nicht ausreichend **geschult**. Dazu fehlt sicherlich ebenfalls das Werkzeug, das erst in Form der software-definierten Plattform bereitgestellt wird.

In der Kategorie der Prozesse fällt auf, dass Unternehmen eine recht **hohe Transparenz über ihre Kernprozesse** haben. Besonders in Deutschland hat die Zertifizierung nach der Qualitätsmanagementnorm ISO 9001 (vgl. DIN EN ISO 9001), die eine Vielzahl von Unternehmen umgesetzt haben, einen Anteil daran. Zwar stimmen die ISO-Prozesse nach praktischer Erfahrung des Autors bei vielen Unternehmen nicht mit den gelebten Prozessen überein, sie bilden aber trotzdem einen guten Ausgangspunkt für eine tatsächliche Prozesstransparenz. Auf der Basis können dann über die software-definierte Plattform auch **konfigurierbare Prozesse** umgesetzt werden. Die Anforderung nach wenigen zentralen Informationssystemen ist nur bei den *Architekturtypen IV* und *V* voll ausgeprägt, was daran liegt, dass hier zentrale ERP-Systeme den Kern der Informationssystem-Architektur bilden. Auf einer solchen Basis ist es aus Prozesssicht einfacher möglich, schnell die Vorteile einer software-definierten Plattform im Unternehmen nutzen zu können.

Mit Blick auf die Daten und die für datenbasierte Entscheidungen notwendige **Informationsqualität** ist festzustellen, dass eine hohe Informationsqualität nach der Definition in Kapitel 6.5.2 bei allen Architekturtypen nicht durchgängig erreicht ist. Wenn Mitarbeiter künftig jedoch noch stärker auf die Informationen, die ein Informationssystem liefert, vertrauen müssen, ist eine hohe Informationsqualität jedoch unabdingbar. Dies zu erreichen, ist gleichzeitig eine große Herausforderung, die nur mit den zunehmend besseren technischen Möglichkeiten, automatisiert Daten aufzunehmen, erreichbar ist. Auch auf der Anwendungsebene der Daten ist eine **durchgängige Produktdatenstruktur**, welche den Informationsfluss bei produzierenden Unternehmen steuert, nicht umgesetzt. Dies ist sicherlich auch bedingt durch die fehlende technische Umsetzung entsprechender Schnittstellen zwischen den oft unterschiedlichen Informationssystemen, die zur Erzeugung und Nutzung der Produktdatenstruktur in verschiedenen Prozessschritten bei Unternehmen genutzt werden.

Das Ziel, **erweiterbare Datenmodelle** umzusetzen, wird noch in keinem der Architekturtypen erreicht. Dies ist erst möglich, wenn nicht nur eine Datenstruktur und ein unternehmensweites Datenmodell aufgesetzt wurden, sondern auch die entsprechenden Prozesse und Organisationen, wie solche Erweiterungen implementiert werden können, im Unternehmen etabliert sind.

7.3 Abgeleitete Entscheidungskriterien

Die im Kapitel 7.1 beschriebenen Eigenschaften einer Zielinformationssystem-Architektur ist für viele produzierende Unternehmen nur mit sehr hohen Aufwänden erreichbar, vor allem für solche, deren aktuelle Informationssystem-Architektur den *Architekturtypen I* und *II* zugeordnet werden kann. Für eine **effiziente Nutzung einer software-definierten Plattform ist jedoch gerade für diese Unternehmen eine Transformation der Informationssystem-Architektur unumgänglich**. Um die Aufwände dazu richtig zu priorisieren und in einem wirtschaftlich sinnvollen Rahmen zu halten, sind hierzu Nutzen-Aufwand-Entscheidungen zu treffen, da nicht jedes beschriebene

Ziel für die konkreten Herausforderungen des Unternehmens denselben Nutzen bringt. Dabei muss das Unternehmen den Aufwand fallabhängig bewerten.

Der Nutzen für das Unternehmen lässt sich aus der Unternehmensstrategie und dem Geschäftsmodell des Unternehmens ableiten, aber auch aus dem Abstand zum gewünschten Ziel-Zustand der Informationssystem-Architektur (s. FOTH 2016, S. 41). Dieser Abstand kann in der obigen Abbildung 7-11 bei den einzelnen Zielen für jeden Architekturtyp abgelesen werden. Generell empfehlen HABERFELLNER ET AL., zuerst die Themen anzugehen, die den meisten Aufwand erfordern (s. HABERFELLNER ET AL. 2015, S. 238). Ähnlich lautet auch die Empfehlung von SCHUH ET AL. für den Aufbau von Fähigkeiten eines Unternehmens zur Umsetzung von Industrie 4.0, die benötigten Fähigkeiten gleichmäßig über alle Dimensionen aufzubauen – also beispielsweise nicht Ziele der organisatorischen Dimension den anderen zu bevorzugen (s. SCHUH ET AL. 2017a, S. 48).

Es empfiehlt sich also, die **Entscheidungen zur Änderung der Informationssystem-Architektur vom Abstand zum Ziel-Zustand abhängig** zu machen, und dabei darauf zu achten, dass die **vier Dimensionen Technik, Organisation, Prozess und Daten angeglichen** werden.

Plattformtyp	technische Anforderungen	organisatorische Anforderungen	prozessuale Anforderungen	Datenintegrationsanforderungen
Typ III				
Typ II				
Typ I				
Plattformtyp	Techniksicht	Organisations-sicht	Prozess-sicht	Datensicht
Typ I - III				
Typ I u. II				
Typ I				

Abbildung 7-12: Vorgehen zur Herleitung der Hierarchisierung der Entscheidungen zur Ziel-Architektur in Abhängigkeit vom Plattformtyp (eigene Darstellung)

Anhand der Einsatzzwecke und den Zielen, die das Unternehmen mit der Plattformnutzung verfolgt, kann das Unternehmen entscheiden, welche software-definierte Plattform welchen Plattfortmtyps es ausgewählt. In Abhängigkeit des Plattfortmtyps können die Anforderungen an die Zielarchitektur zusätzlich hierarchisiert werden. In der in Abbildung 5-12 dargestellten Herleitung der Integrationsanforderungen software-definierter Plattformen (s. Kapitel 5.6.1) wurde gezeigt, dass die Integrationsanforderungen von *Plattfortmtyp III* bis *I* aufeinander aufbauen. Daraus abgeleitet können auch die Anforderungen an eine Zielinformationssystem-Architektur aufgeteilt werden in Basisanforderungen, die für *Plattfortmtyp III* erfüllt sein müssen, und darauf aufbauenden Anforderungen für die *Plattfortmtypen II* und *I*. Abbildung 7-12 veranschaulicht diese Zusammenhänge: So können beispielsweise aus den technischen Integrationsanforderungen für Plattfortmtyp III die primär zu verfolgenden Ziele der Zielarchitektur aus technischer Sicht abgeleitet werden. Tabelle 7-1 zeigt im Ergebnis dieser Ableitung die Zuordnung der einzelnen Anforderungen, wie sie in Kapitel 7.1 beschrieben wurden.

Tabelle 7-1: Zieldimensionen der Informationssystem-Architektur in Abhängigkeit des gewählten Plattfortmtyps (eigene Darstellung)

Plattfortmtyp	Techniksicht	Organisations-sicht	Prozesssicht	Datensicht
Typ I - III	skalierbare Betriebsumgebung standardisierte Schnittstellen technische Integration über Plattform software-definierte Infrastruktur weltweite Verfügbarkeit	strukturierte Rollenverteilung Datenverantwortung beim Fachbereich Mitarbeiter im Umgang mit Daten geschult	Transparenz über Anwendungsfälle der IS hohe Prozessstandardisierung	hohe Informationsqualität
Typ I u. II	regelmäßig aktualisierte Informationssysteme weitestgehend modulare Applikationsarchitektur	IT-Abteilung sieht sich als Erfolgspartner Architectural Thinking	Kernprozessstransparenz konfigurierbare Prozesse wenige zentrale Informationssysteme	redundanzfreie Datenbestände erweiterbare Datenmodelle
Typ I	Echtzeitdatenübertragung ermöglicht ubiquitäre Darstellung	bimodale IT-Organisation		zentrales Unternehmensdatenmodell durchgängige Produktdatenstruktur

Ein Vorgehen zur Entscheidungsfindung im Unternehmen zur besseren Beherrschung der IT-Komplexität und letztendlich zur Einführung einer software-definierte Plattform zeigt das folgende Kapitel 8 auf.

7.4 Zusammenfassung Kapitel 7

Im vorliegenden Kapitel 7 wird ein **Ziel-Zustand der Informationssystem-Architektur** produzierender Unternehmen vorgestellt. Jener ist aus den Integrationsanforderungen software-definierter Plattformen sowie Empfehlungen zum Umgang mit IT-Komplexität abgeleitet. Hierzu werden zuerst **allgemeine Anforderungen an diesen Ziel-Zustand definiert** und dann die **Merkmale der Zielinformationssystem-Architektur** in vier Dimensionen ausgearbeitet (s. Kapitel 7.1). Anhand des definierten Ziel-Zustands kann der **Unterschied** zwischen den Informationssystem-Architekturtypen produzierender Unternehmen und dem Ziel-Zustand **erklärt** werden, indem für jedes einzelne Zielmerkmal dargestellt ist, wie weit dies pro Architekturtyp vom Ziel-Zustand entfernt ist (s. Kapitel 7.2). Zuletzt können auf dieser Basis **Entscheidungskriterien für die Transformation** der Informationssystem-Architektur produzierender Unternehmen abgeleitet werden. Unter anderem hängen diese vom Plattfortmtyyp ab, den das Unternehmen einführen möchte, weshalb die Zielarchitekturmerkmale den einzelnen Plattfortmtypten zugeordnet werden (s. Kapitel 7.3).

8 Gestaltung des Integrationsprozesses software-definierter Plattformen

Das im Folgenden beschriebene Gestaltungsmodell des Integrationsprozesses software-definierter Plattformen für produzierende Unternehmen bringt die in den Kapiteln 5, 6 und 7 beschriebenen Modelle in die Anwendung. Anhand der Struktur des von HABERFELLNER ET AL. beschriebenen **Problemlösungszyklus** werden die Schritte im Sinne eines sollzustandsorientierten Vorgehens von der Zielformulierung bis zur Entscheidung aufgezeigt (s. Abbildung 8-1 u. HABERFELLNER ET AL. 2015, S. 81). Dieser ist als Mikrozyklus in einem Projekt zu verstehen, der, je nach Projektansatz, ein- bis n-mal durchlaufenden werden kann (s. Kapitel 4.2.1).

In der Anwendung unterstützt dieses Vorgehen bei der **Initiierung des IT-Strategie-Zyklus** (s. Kapitel 2.2.3). Die Informationssystem-Architekturgestaltung ist ein zentrales Element innerhalb der IT-Strategie. In der ersten Phase der Untersuchung des Umfelds, der Ziele des Unternehmens und des Ist-Zustands hilft das Beschreibungsmodell der Informationssystem-Architektur (s. Kapitel 6). Bei der darauffolgenden Formulierung des Ziel-Zustands unterstützt das Beschreibungsmodell der software-definierten Plattformen (s. Kapitel 5). Bei der Operationalisierung zur Ableitung konkreter Projekte kann das Erklärungsmodell (s. Kapitel 7) genutzt werden.

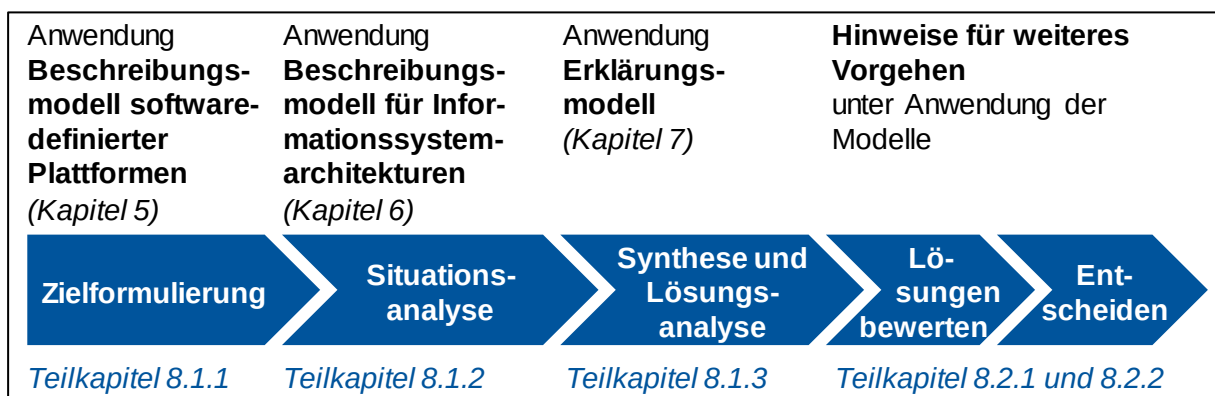


Abbildung 8-1: Einordnung der entwickelten Modelle in den Problemlösungszyklus nach HABERFELLNER ET AL. (eigene Darstellung)

Konkret wird das Beschreibungsmodell für software-definierte Plattformen dazu genutzt, **Anforderungen an das Ziel** – eine Informationssystem-Architektur mit eingeführter software-definierter Plattform – zu definieren (s. Kapitel 8.1.1). Gegen dieses Ziel kann die **aktuelle Situation der Informationssystem-Architektur** des Unternehmens analysiert werden. Dazu kann das Beschreibungsmodell für Informationssysteme genutzt werden, um einzelne Informationssysteme mit ihren relevanten Eigenschaften zu beschreiben. Die Typisierung von Informationssystem-Architekturen dient als Ausgangspunkt für eine eigene Detaillierung durch das Unternehmen (s. Kapitel 8.1.2). Anhand des Erklärungsmodells können die **Synthese und Lösungsanalyse** erfolgen. Durch die Darstellung des Zusammenhangs zwischen Ist- und Ziel-Zustand kann ein Unternehmen eigene Entscheidungsbedarfe ableiten (s. Kapitel 8.1.3).

Somit ist der Rahmen für den Integrationsprozess software-definierter Plattformen geschaffen. Zur Gestaltung des Prozesses gibt das vorliegende Kapitel sodann Hinweise zum konkreten Verfahren. Dazu ist zuerst eine **Vorgehensweise** zu definieren, die beispielsweise bei einem bestimmten Unternehmensbereich beginnt (s. Kapitel 8.2.1). Daraufhin ist ein konkretes Softwareprodukt mit den Funktionen einer **software-definierten Plattform auszuwählen** (s. Kapitel 8.2.2).

Damit beantwortet das vorliegende Kapitel die folgende Forschungsfrage: **Wie kann der Integrationsprozess zur Einführung einer zentralen software-definierten Plattform in die Informationssystem-Architektur produzierender Unternehmen systematisch gestaltet werden?**

8.1 Anwendung der Modelle im Problemlösungszyklus

8.1.1 Zielformulierung mit dem Beschreibungsmodell für software-definierte Plattformen

Eine zentrale Anforderung an die in der vorliegenden Arbeit vorgestellten Methoden ist die **anwendungsnahe Unterstützung von Unternehmen** bei der Beherrschung von IT-Komplexität (s. Kapitel 4.1). Als ersten Schritt zur Erfüllung dieser Anforderung werden im Folgenden Hinweise zur Formulierung von Gestaltungszielen der Informationssystem-Architektur gegeben.

Zuerst muss ein Unternehmen die Ziele festlegen, die es mit seiner Informationssystem-Architektur und deren Änderung verfolgt. Die vorliegende Arbeit schlägt in diesem Kontext vor, die Architektur mit dem Ziel zu transformieren, eine software-definierte Plattform als zentrale Datenquelle für das Unternehmen einzuführen. Die **Ziele zur Transformation der Informationssystem-Architektur sind also abhängig von den Zielen, die ein Unternehmen durch die Einführung der Plattform erreichen möchte**. In Kapitel 5.1.1 wurden hierzu die Aufgaben vorgestellt, die mit einer software-definierten Plattform erfüllt werden können. Dies sind die Datensicherheit, Datenaufbereitung und -analyse, die Datenmodellierung und das Quellenmanagement zusammen mit der Einbindung der Umsysteme und zuletzt die Datenbereitstellung. Für seine **konkreten Anwendungsfälle** muss das Unternehmen nun entscheiden, **inwieweit es die genannten Aufgaben künftig durch die software-definierte Plattform umsetzen möchte**. Dazu sind eben diese Anwendungsfälle zu definieren. Sie hängen wiederum ab von der Unternehmensstrategie und können unter Nutzung bewährter Methoden wie SWOT-Analysen⁴⁷ (s. FLUHR 2014, S. 137) entwickelt und in Usecase-Diagrammen dargestellt werden (s. Kapitel 5.6.4).

⁴⁷ SWOT steht für *Strengths, Weaknesses, Opportunities and Threats*. Eine SWOT-Analyse wird genutzt, um strategische Handlungsempfehlungen für eine bestimmte Problemstellung abzuleiten (s. KERTH ET AL. 2015, S. 174-178).

Aus den Anwendungsfällen kann das Unternehmen ableiten, welche Aufgaben hauptsächlich mit der software-definierten Plattform umgesetzt werden. Sollen beispielsweise künftig anhand eines digitalen Schattens der Auftragsabwicklung die Datentransparenz und die Datennutzung erhöht werden, muss die software-definierte Plattform nahezu alle der genannten Aufgaben erfüllen. Soll die software-definierte Plattform dagegen zur Verknüpfung von OT-Systemen genutzt werden, ist vor allem das Quellenmanagement und die Einbindung der Umsysteme zu beachten; die anderen Aufgaben sind für diesen Anwendungsfall weniger relevant. Tabelle 8-1 zeigt die mögliche Anwendung dieser Aufgabenzuordnung.

Tabelle 8-1: Aufgaben einer software-definierten Plattform in Abhängigkeit von konkreten Anwendungsfällen (eigene Darstellung)

Aufgabe	Anwendungsfall			Summe (beispielhaft)
	1: digitaler Schatten der Auftragsabwicklung	2: Verknüpfung von OT-Systemen	<i>n</i>	
Datenbereitstellung	x		...	5 62,5 %
Einbindung Umsysteme	x	x	...	8 100 %
Quellenmanagement		x	...	1 12,5 %
Datenmodellierung	x		...	3 37,5 %
Datenanalyse	x		...	3 37,5 %
Datenaufbereitung	x		...	4 50 %
Datensicherheit	x		...	6 75 %

Auf dieser Basis können im nächsten Schritt die **relevanten Merkmale einer software-definierten Plattform bestimmt** werden. Dies kann als Vorbereitung für die Erstellung eines Lastenheftes für die Softwareauswahl genutzt werden (s. Kapitel 8.2.2). Entsprechend dem Vorgehen zur Softwareauswahl in produzierenden Unternehmen (s. BROSZE ET AL. 2009, S. 56; LEYH 2015, S. 425), das eine von den Anwendungsfällen und Soll-Prozessen ausgehende, **iterative Befüllung eines Lastenheftes** vorsieht, kann die Morphologie der Eigenschaften software-definierter Plattformen (s. Kapitel 5.4) dazu verwendet werden. Dabei hilft die Gesamtsicht, welche die Morphologie

über die Eigenschaften software-definierter Plattformen bietet. Damit können Unternehmen die Merkmale auswählen, die sie für die Erfüllung ihrer Anforderungen benötigen (s. Tabelle 8-2 mit hervorgehobenen, beispielhaft ausgewählten Merkmalsausprägungen). Weitere spezifische Anforderungen des Unternehmens sollten entsprechend ergänzt werden.

Tabelle 8-2: Exemplarische Morphologie der Eigenschaften software-definierter Plattformen als Auswahlhilfe zur Softwareauswahl (eigene Darstellung)

Merkmal	Ausprägungen			
<i>Schnittstellen</i>	generische Schnittstellen	vordefinierte Schnittstellen		
<i>Datenmanagement</i>	feste Datenstruktur	Datenharmonisierung und -bereinigung	Datenanreicherung	frei konfigurierbare Datenobjekte
<i>Datenbankkonzept</i>	relationale Datenbank	NoSQL-Datenbank	sonstige Datenbankkonzepte	
<i>Stufen der Datenverarbeitung</i>	Erzeugen von Sichtbarkeit	Transparenz und Diagnose	Prognose	Adaption und Entscheidungsautomatisierung
<i>IT-Sicherheitsfunktionen</i>	nur interne Sicherheitsfunktionen	Sicherheit für Plattform und angebundene Systeme	Vollintegration in IT-Sicherheitsinfrastruktur des Unternehmens	
<i>Informationsbereitstellung</i>	keine eigene Auswertungsfunktion	Bereitstellung von Auswertungen und Visualisierungen	Generierung eigener Apps/Widgets	
<i>Entwicklungsumgebung (E.)</i>	keine eigene E.	eigene, proprietäre E.	offene E.	

Darauffolgend können, ausgehend vom Typ der einzuführenden Plattform, **Anforderungen an die Transformation der Informationssystem-Architektur** gestellt werden. Dazu können die typabhängigen Integrationsanforderungen, die in Kapitel 5.6 der vorliegenden Arbeit vorgestellt wurden, genutzt werden. Das Unternehmen muss dabei zuerst die Integrationsperspektive, die Integrationsreichweite und den angestrebten Automatisierungsgrad definieren. Daraufhin sind die vier Integrationsdimensionen

Technik, Organisation, Prozesse und Daten zu betrachten. Zur Nutzung dieser allgemeingültigen Integrationsanforderungen sind diese **im Unternehmen mit konkreten Technologien, Personen, Prozessen und Datenbeständen des Unternehmens zu detaillieren**. Tabelle 8-3 zeigt ein Beispiel für eine solche Konkretisierung.

Tabelle 8-3: Integrationsanforderungen und beispielhafte Konkretisierungen für ein Unternehmen (eigene Darstellung)

Dimension	Integrationsanforderung	Konkretisierung
<i>Technik</i>	einheitliche Schnittstellen	Prüfung der Schnittstellen des ERP-Anbieters, Rücksprache mit Softwarehersteller
<i>Organisation</i>	geordnete Betriebsverantwortung	Outsourcing-Entscheidung durch Frau Maier vorbereiten, Betriebskonzept durch IT-Abteilung prüfen
<i>Organisation</i>	Organisation an schnelle Änderungen angepasst	Prüfung der Kompetenz von Mitarbeiter Müller und Schulz im Thema ‚Agile Entwicklung‘
<i>Daten</i>	einheitliches Kerndatenmodell	Prüfung des aktuellen Datenmodells auf Vollständigkeit durch Mitarbeiter aus den Abteilungen Vertrieb, Produktion und Entwicklung
...	...	...

8.1.2 Situationsanalyse mit dem Beschreibungsmodell für Informationssystem-Architekturen

Im nächsten Schritt zur Umsetzung einer Informationssystem-Architektur bei software-definierten Plattformen muss eine **Situationsanalyse der aktuellen Informationssystem-Architektur** durchgeführt werden. Dies ist elementar, um ein gutes Verständnis und einen Überblick über die Architektur zu erhalten. Oft bestehen in Unternehmen schon Dokumentationen über Teile der Architektur, die hierzu als Quelle genutzt werden können. Für den Zweck der Situationsanalyse wird jedoch ein vollständiges Bild der Architektur benötigt, gemäß dem „*Mile-Wide-Inch-Deep-Prinzip*“ von KEUNTJE (S. KEUNTJE 2010, S. 176). Das bedeutet, dass **nicht alle Informationen** über die im Unternehmen genutzten Informationssysteme benötigt werden, aber **alle Informationssysteme** des Unternehmens aufgenommen werden müssen.

Zur Aufnahme der relevanten Informationen über Informationssysteme kann die **Morphologie für Informationssysteme** aus Kapitel 6.6 dieser Arbeit **genutzt werden**. Diese enthält alle für die Einführung einer software-definierten Plattform relevanten Eigenschaften von Informationssystemen. Falls eine entsprechende Dokumentation im Unternehmen besteht, kann diese Aufnahme relativ schnell entstehen. Falls nicht,

kann die notwendige Aufnahme hinterher zum initialen Aufsetzen einer Assetdatenbank dienen, in welcher die Informations- und Kommunikationssysteme eines Unternehmens hinterlegt sind. Werkzeuge zur Datenerhebung können klassische Interviews mit Verantwortlichen aus der IT-Abteilung und den Fachbereichen sein, aber auch technische Hilfsmittel wie Netzwerkscanner zur Identifikation von Serversystemen im Unternehmensnetzwerk (s. Abbildung 8-2).

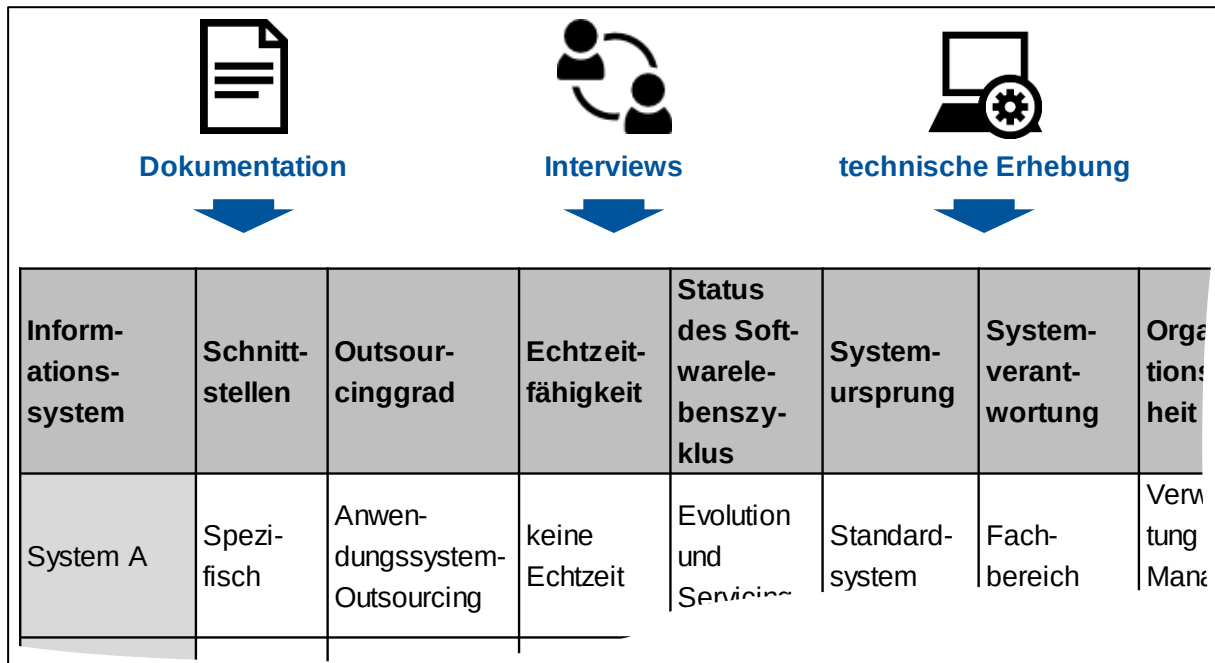


Abbildung 8-2: Vorgehen zur Erhebung der Informationssysteme (eigene Darstellung)

In Kapitel 6.7 wurden verschiedene Darstellungsformen vorgestellt, die zur Visualisierung der Informationssystem-Architektur genutzt werden können. Aus den Visualisierungen können **Schlüsse über Redundanzen und Lücken bei der Funktionsüberdeckung** von Anwendungssystemen und der Datenarchitektur, genauso auch über **organisatorische Verantwortungsverteilung** und **Systemunterstützung der Kernprozesse**, gewonnen werden. Das Unternehmen kann schon daraus Schlüsse ziehen, wenn Teile der Architektur den selbst gesetzten Zielen widersprechen, und so anzugehende Herausforderungen auflisten (s. Abbildung 8-3). Anhand der aufgenommenen Eigenschaften der Informationssysteme sind auch weitere Visualisierungen möglich. Sie sollten im Speziellen dazu erstellt werden, um vermeintliche Eigenschaften der Informationssystem-Architektur mit Fakten zu hinterlegen.

Auf Basis der Architekturdarstellungen und der in Kapitel 6.8 vorgestellten Beispiele kann ein Unternehmen seine Informationssystem-Architektur sodann in einen der fünf Architekturtypen einordnen. Zentrale Entscheidungspunkte dazu sind das generelle Architekturparadigma des Unternehmens sowie die IT-Organisationspositionierung. Die Einordnung in einen der Typen dient dazu, im folgenden Schritt die Zusammenhänge zum Ziel-Zustand zu erklären.

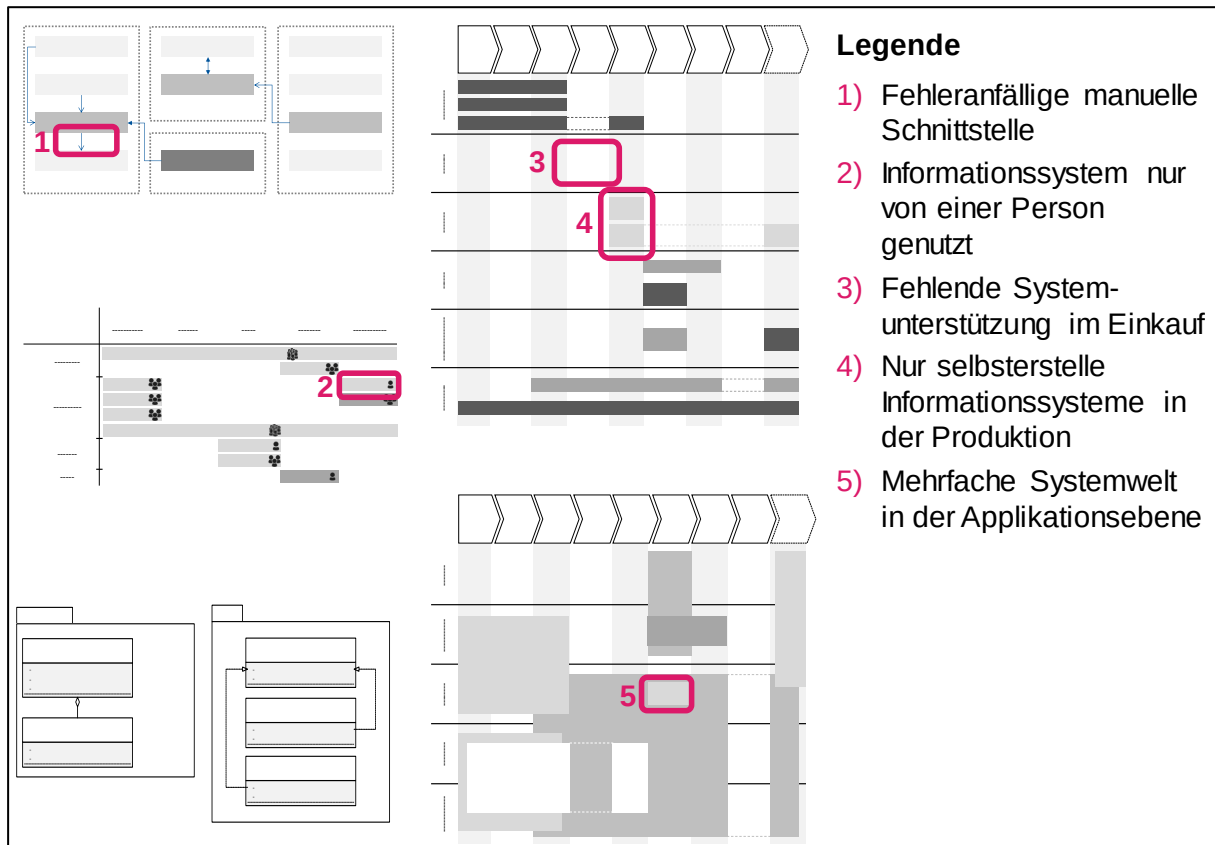


Abbildung 8-3: Beispiele von Sichten auf Informationssystem-Architekturen mit markierten Herausforderungen (eigene Darstellung)

8.1.3 Synthese und Lösungsanalyse mit dem Erklärungsmodell

Nachdem die Anwendungsfälle und Integrationsanforderungen der software-definierten Plattform festgelegt wurden und demgegenüber der Ist-Zustand der Informationssystem-Architektur betrachtet wurde, folgt im Problemlösungszyklus nun der Schritt der Synthese und Lösungsanalyse. Hierzu kann das in Kapitel 7 vorgestellte Erklärungsmodell genutzt werden. Der dort definierte **Ziel-Zustand der Informationssystem-Architektur dient dem anwendenden Unternehmen zum Abgleich mit eigenen Zielen** und als Leitlinie für notwendige Transformation der Architektur. Dazu gibt das Modell zuerst allgemeine Hinweise zur Reduzierung von IT-Komplexität und bringt diese mit Integrationsanforderungen software-definierter Plattformen zusammen.

Unter Nutzung der in der vorliegenden Arbeit vorgestellten Typisierung von Informationssystem-Architekturen kann ein Unternehmen anhand seines Typs eine Einordnung in die in Kapitel 7.2 vorgestellte Erklärung des Zusammenhangs zwischen Ist- und Ziel-Zustand vornehmen. In Verbindung mit gegebenenfalls ergänzten unternehmensspezifischen Herausforderungen kann daraus abgelesen werden, in welchen Bereichen große und in welchen kleine Abstände zum Ziel-Zustand bestehen.

Eine **Priorisierung der notwendigen Umsetzungen kann zuerst anhand des Abstands zum Ziel-Zustand vorgenommen** werden. Zusätzlich sollte in Ergänzung dazu das in Kapitel 7.3 vorgestellte Modell genutzt werden, das die Anforderungen an

die Zielarchitektur in Verbindung mit dem vom Unternehmen ausgewählten Plattformtyp bringt. Damit werden dem anwendenden Unternehmen der vorgestellten Modelle also die **Handlungsfelder aufgezeigt, die es zur Einführung einer software-definierten Plattform angehen** muss, sowie eine Priorisierung der Handlungsfelder vorgeschlagen, um im nächsten Schritt konkrete Umsetzungsprojekte daraus abzuleiten.

8.2 Hinweise zum Auswahl- und Einführungsprozess für software-definierte Plattformen

8.2.1 Definition einer Vorgehensweise

Die Vorbereitung der Informationssystem-Architektur, die im vorherigen Kapitel beschrieben wurde, ist die Vorbedingung, die erfüllt werden muss, wenn eine software-definierte Plattform in ein Unternehmen eingeführt wird. Hinsichtlich der Einführung selbst ist ebenfalls eine Vorgehensweise zu planen.

Da vielen Unternehmen die Erfahrung im Umgang mit zentralen Informationsplattformen fehlt, hilft es, eine software-definierte Plattform zuerst nur für einen **bestimmten Bereich des Unternehmens einzuführen** und damit Erfahrungen zu sammeln. Um einen solchen Bereich auszuwählen, kann das Unternehmen sich an dem vorgestellten zentralen Unternehmensdatenmodell orientieren (s. Kapitel 7.1.4). Es sollte also einer der drei Bereiche *Produkt*, *Auftrag* oder *Kunde* gewählt werden. Wird beispielsweise der Bereich *Auftrag* gewählt, sind alle auftragsrelevanten Daten betroffen, die Mitarbeiter des Unternehmens, welche sich mit der Produktion und Logistik beschäftigen, die Prozessbereiche *Produktionsplanung* und *Herstellung* und die zu den jeweiligen Bereichen gehörigen technischen Bestandteile der genutzten Informationssysteme.

Zur **transparenten Kommunikation des Projekts** sollten ein Projektplan und ein Projektteam aufgestellt werden. Die Organisation muss einen Projektsponsor vorsehen, der umfangreiche Entscheidungsbefugnisse im Unternehmen hat, da die Plattformeinführung nahezu alle Bereiche des Unternehmens betrifft (s. Kapitel 5.6.3). Weiterhin sind die bekannten Regeln für Softwareeinführungsprojekte in Unternehmen umzusetzen (vgl. bspw. LEYH 2015, S. 418-432; ALPAR ET AL. 2016, S. 409-416).

Bei der Projektplanung im Kontext software-definierter Plattformen sind besonders zwei Grundprinzipien zu beachten: das **Minimal-Viable-Product-** und das **Standard-Prinzip**. Ersteres beschreibt einen Ansatz, bei dem schnell nutzbare Ergebnisse für den Kunden bereitgestellt werden. Im ersten Schritt ist dabei bewusst noch nicht der volle geplante Leistungsumfang implementiert, dafür können die Anwender bereits Erfahrung mit den Teilergebnissen sammeln. Zentrales Element dieses Vorgehens ist die Festlegung von Anwendungsfällen (s. Kapitel 8.1.1), die schrittweise umgesetzt werden sollen (s. HANSCHKE 2017, S. 40-43). Dieses Prinzip ist deshalb wichtig, weil ein **neues, von der Art her im Unternehmen bisher unbekanntes, Informationssystem eingeführt** wird. Da es dabei nur schwer möglich ist, alle Funktionen im Voraus zu spezifizieren, hilft es dabei, schrittweise vorzugehen. Gleichzeitig können so

auch Vorbehalte im Unternehmen abgebaut werden. Wie in Kapitel 5.6.3 beschrieben, bilden sich in Unternehmen oft „Fürstentümer“, welche die Hoheit über „ihre“ Informationen beanspruchen. Die Einführung einer software-definierten Plattform löst dies auf, da im Ziel-Zustand alle relevanten Informationen für alle Unternehmensbereiche verfügbar sind. Wenn schnell mit ersten Funktionen gearbeitet werden kann, hilft dies dabei, dass die Mitarbeiter schnell den Nutzen dieses Vorgehens erkennen können.

Das **Standard-Prinzip** empfiehlt den Ansatz, möglichst wenig von den vom Softwarehersteller vordefinierten Funktionen der eingesetzten Informationssysteme abzuweichen. In der Vergangenheit tendierten gerade auch produzierende Unternehmen dazu, bei der Softwareeinführung zuerst **umfangreiche Anpassungsprogrammierungen** vorzunehmen, bevor die Software eingeführt werden konnte. Dies ist im Zuge einer schnellen Einführung **nicht zielführend**; außerdem hat sich nach Erfahrung des Autors gezeigt, dass die heute verfügbaren systemgestützten Standardprozesse und -anwendungsfälle für fast alle Anwendungsfälle von Unternehmen ausreichend sind.

8.2.2 Auswahl einer software-definierten Plattform

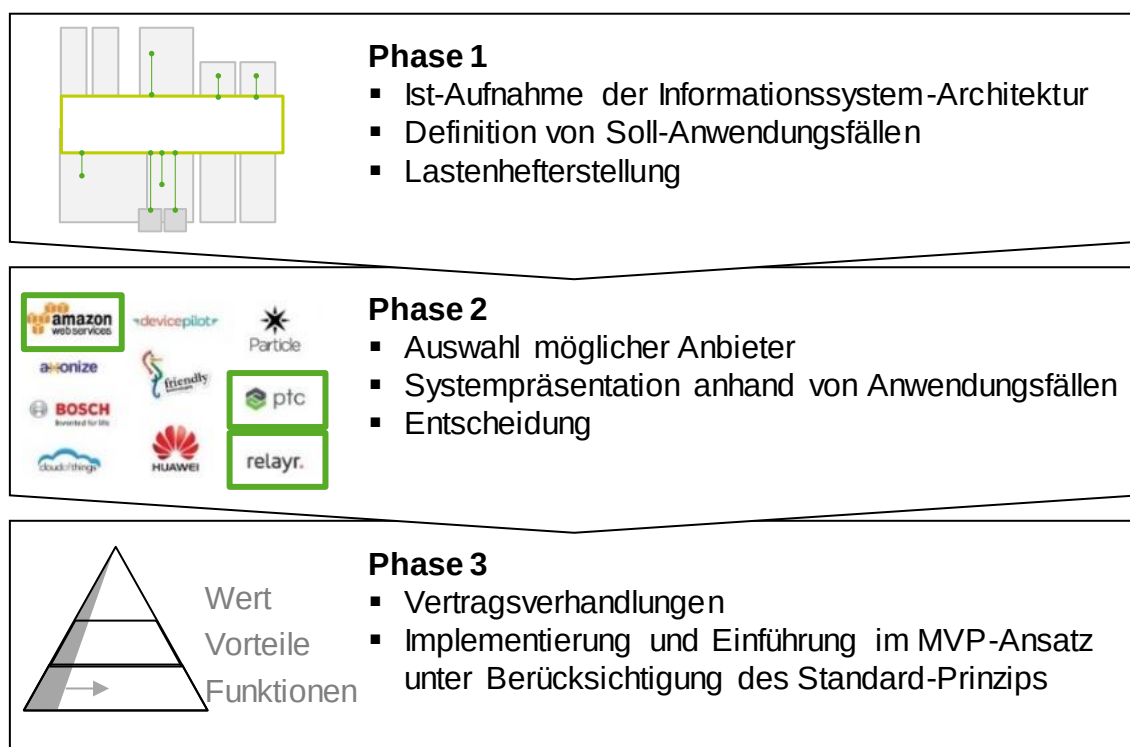


Abbildung 8-4: Auswahl und Einführung einer software-definierten Plattform (eigene Darstellung)

Zur Auswahl von Software besteht am FIR ein seit Jahren erprobtes Vorgehen: das schon in anderen Kapiteln referenzierte **Drei-Phasen-Konzept** (vgl. BROSZE ET AL. 2009). Dieses basiert einerseits auf umfangreichen Maximallastenheften, die alle möglichen Anforderungen an beispielsweise ein ERP-System beinhalten. Andererseits gibt es Listen von Softwareherstellern und Implementierungspartnern, beispielsweise für ERP-Systeme, welche die Funktionen ihrer Systeme in derselben Struktur wie das Lastenheft bereitgestellt haben. Zum Zeitpunkt der Abfassung der vorliegenden Arbeit gab es diese beiden Komponenten nicht für software-definierte Plattformen. Das Drei-

Phasen-Konzept kann daher für software-definierte Plattformen **nur mit Anpassungen eingesetzt** werden (s. Abbildung 8-4), die im Folgenden vorgestellt werden.

Wie im vorherigen Kapitel beschrieben, ist das *Minimal-Viable-Product*-Prinzip zentral bei der Implementierung einer software-definierten Plattform (s. Kapitel 8.2.1). Ein wichtiges Werkzeug dabei ist die Definition von Anwendungsfällen. Im Gegensatz zum Vorgehen beim Drei-Phasen-Konzept, das Soll-Prozesse nutzt, um den Ziel-Zustand zu beschreiben, sollten bei der Softwareauswahl einer software-definierten Plattform jene Anwendungsfälle diese Rolle einnehmen. Das Unternehmen sollte, geordnet nach den Anwendungsbereichen *Produkt*, *Auftrag* und *Kunde*, diese **Anwendungsfälle auflisten** und sich damit ein **Bild vom Ziel-Zustand machen**. Um ein Lastenheft zu erstellen, müssen die Anforderungen mit möglichen Funktionen software-definierter Plattformen abgeglichen werden. Dazu kann der **Funktionskatalog** im Anhang A.6 dieser Arbeit genutzt werden. Er gibt einen detaillierten Überblick möglicher Funktionen aktuell vorhandener Plattformen. Anhand eines solchen Lastenheftes kann dann ein Abgleich mit vorhandenen Plattformen vorgenommen werden. Eine erste, nicht vollumfassende **Liste möglicher einsetzbarer Softwareprodukte** liegt als Anhang A.5 dieser Arbeit bei.

Zur konkreten Anbietersauswahl sollten zudem Anforderungen an den Softwarepartner selbst, wie beispielsweise räumliche Nähe oder Branchenkenntnis, ergänzt werden. Wie beim Drei-Phasen-Konzept ist es in der finalen Phase der Softwareauswahl wichtig, *vergleichbare* Kriterien zur Auswahl zu bekommen. Dies ist einerseits durch einen Vergleich der Erfüllung der Anwendungsfälle und allgemeinen Anforderungen gegeben, aber auch durch die **Vergleichbarkeit einer Softwarepräsentation**. Dazu sollten den zur Auswahl stehenden Anbietern dieselben zentralen Anwendungsfälle zur Verfügung gestellt werden, an denen sie ihre Präsentation auszurichten haben.

Nach der Entscheidung für einen Anbieter können die definierten Anwendungsfälle gut als Test- und Abnahmekriterien gegenüber dem Softwarehersteller genutzt werden.

8.3 Zusammenfassung Kapitel 8

Kapitel 8 schließt den Methodenteil der vorliegenden Arbeit ab. Hierzu wurden alle bisher vorgestellten Modelle aufgegriffen und in ein anwendbares Gestaltungsmodell integriert. Das Gestaltungsmodell orientiert sich am Problemlösungszyklus von HABERFELLNER ET AL. und zeigt diesen als *einen* Mikrozyklus auf (s. Kapitel 4.2.1). Dieser beginnt mit der **Zielformulierung** anhand des Beschreibungsmodells für software-definierte Plattformen. Es hilft beim Verständnis des zu erreichenden Ziels (s. Kapitel 8.1.1). Die **Situationsanalyse** wird mit dem Beschreibungsmodell für Informationssystem-Architekturen produzierender Unternehmen durchgeführt. Damit können einzelne Informationssysteme aufgenommen und die Architektur einer ersten Bewertung unterzogen werden (s. Kapitel 8.1.2). Der Schritt der **Synthese und Lösungsanalyse** wird unter Zuhilfenahme des Erklärungsmodells zwischen Ist- und Ziel-Zustand vorgenommen. Ergebnis dessen sind priorisierte Handlungsfelder für die Transformation der Informationssystem-Architektur mit dem Ziel einer besser beherrschbaren IT-

Komplexität und zur Vorbereitung der Einführung einer software-definierten Plattform (s. Kapitel 8.1.3).

Zu dieser **Softwareeinführung** gibt der letzte Abschnitt Hinweise zum **Projektaufsatz** und zur **Softwareauswahl** (s. Kapitel 8.2.1). Hier wird eine Ergänzung des bewährten Drei-Phasen-Konzepts des FIR vorgeschlagen und auf entwickelte Hilfsmittel im Anhang verwiesen (s. Kapitel 8.2.2).

9 Evaluation

Das vorliegende Kapitel dient entsprechend dem in Kapitel 1.3 vorgestellten Forschungsprozess dazu, die **Validierung der erarbeiteten Ergebnisse vorzustellen**. Dabei ist diese Evaluation keineswegs als einmaliges Ereignis nach Entwicklung aller Modelle zu verstehen. Vielmehr sind entsprechend des iterativen Problemlösungszyklus von HABERFELLNER ET AL. die Modelle wiederkehrend in der Praxis validiert, mit Experten diskutiert und daraufhin auch angepasst worden (s. Anhang A.3). Dabei konnte, angelehnt an die Methode der Aktionsforschung, der Autor selbst die Ergebnisse mitgestalten. Wo es zur praxisnahen Darstellung der Modelle sinnvoll erschien, wurden diese **Ergebnisse in Teilen schon in den vorherigen Kapiteln** (s. z. B. Kapitel 5.5, 6.7 u. 6.8) **vorgestellt**. Im vorliegenden Kapitel sollen die Ergebnisse im **Gesamtzusammenhang anhand von zwei Fallstudien validiert** werden. Dies entspricht einem qualitativ-empirischen Vorgehen, bei dem einige Merkmalsträger besonders intensiv betrachtet werden (s. WILDE U. HESS 2007, S. 282; SCHREINER ET AL. 2015, S. 4). Die Fallstudien dienen dazu, in dem verfolgten anwendungsorientierten Forschungsprozess die direkte Anwendbarkeit der neu entwickelten Modelle und deren Nützlichkeit in der Praxis zu evaluieren und kritisch zu reflektieren (s. HESS ET AL. 2014, S. 132; ULRICH 1984, S. 193).

Wie die Fallstudien dazu ausgewählt wurden und wie sie **strukturiert** sind, wird in Kapitel 9.1 erklärt. Die beiden Fallstudien werden dann in den Kapiteln 9.2 und 9.3 **vorgestellt**, indem zuerst die Ausgangssituation erläutert wird und daraufhin die angewendeten Modelle gezeigt werden. In Kapitel 9.4 werden die Fallstudien zusammengefasst und es wird **kritisch reflektiert**, ob die an die vorliegende Arbeit gestellten Ziele (s. Kapitel 4.1) erreicht wurden.

Durch die in diesem Kapitel durchgeführte Evaluation wird die **Forschungsfrage** „*Wie können produzierende Unternehmen in ihre bestehende Informationssystem-Architektur eine zentrale software-definierte Plattform integrieren?*“ **überprüft**.

9.1 Auswahl der Fallstudien

Für die vorliegende Arbeit wurden zwei Fallstudien unterschiedlicher Ausprägung gewählt. Die erste Fallstudie untersucht das Aachener **Elektromobile-Start-up „e.GO Mobile AG“** (im Folgenden kurz: „e.GO“), das nahezu einem Greenfield-Unternehmen entspricht.⁴⁸ Der Autor hatte selbst Gelegenheit, das Unternehmen über fast drei Jahre zu begleiten und die Informationssystem-Architektur mitzugestalten. e.GO hat in dieser Zeit nicht nur eine Informationssystem-Architektur komplett neu aufgebaut, son-

⁴⁸ Wie weiter unten beschrieben, wurde e.GO als neues Unternehmen gegründet und daher auch die Informationssystem-Architektur komplett neu aufgebaut. Allerdings wurden zu Beginn viele Systeme und Prozesse von der Firma Streetscooter übernommen, aus der heraus e.GO entstanden ist.

dern auch direkt mit dem Ziel gestaltet, einen übergreifenden Datenzugriff zu ermöglichen. Die zweite Fallstudie wurde zusammen mit dem **Plattformanbieter PTC** anhand von ausgewählten Anwendungsfällen von PTC-Kunden durchgeführt. Somit werden sowohl die Anwendersicht als auch die Anbietersicht reflektiert. Wie oben beschrieben, werden mit Fallstudien keine Querschnittsevaluationen durchgeführt, sondern das entwickelte Gestaltungsmodell (s. Kapitel 8) anhand von ausgewählten Fällen intensiv beleuchtet (s. DEINDL 2013b, S. 144). Die beiden Fallstudien beleuchten dabei die Bestandteile des Gestaltungsmodells unterschiedlich stark (s. Abbildung 9-1).

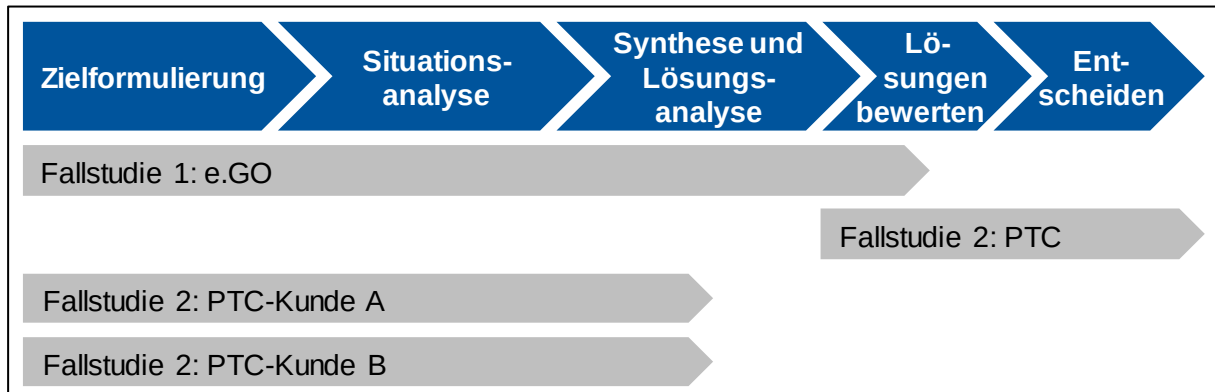


Abbildung 9-1: Übersicht der Fallstudien und deren Abdeckung des Gestaltungsmodells (eigene Darstellung)

Dazu werden die Fallstudien wie folgt aufgebaut: Zuerst wird die **Ausgangssituation** vor Anwendung der Modelle beschrieben. Im zweiten Schritt folgt die Vorstellung von **Anwendungsfällen** und Zielen, die in der Fallstudie mit dem Platfformeinsatz verfolgt wurden. Darauf folgt unter Nutzung der Morphologie der Merkmale software-definierter Plattformen (s. Kapitel 5.4) die Ableitung von **technischen Auswahlkriterien** für solche Plattformen. Es wird überprüft, inwieweit die morphologische Beschreibung die tatsächlichen Kriterien der Fallstudienpartner erfüllt. Anhand dieser erhobenen Merkmale kann der Plattfortmtyt identifiziert werden, dem die ausgewählten Merkmale zugeordnet werden können. Unternehmen können damit die **Integrationsanforderungen konkretisieren**, die in der vorliegenden Arbeit plattformtypabhängig herausgearbeitet wurden (s. Kapitel 5.6). Im nächsten Schritt werden die Informationssysteme, die in dem Fallstudienunternehmen im Einsatz sind, unter Nutzung der Morphologie zur Beschreibung von Informationssystemen (s. Kapitel 6.6) gesammelt. Relevante **Architekturdarstellungen** werden abgeleitet und abgebildet. Diese werden mit in den untersuchten Unternehmen vorhandenen Darstellungen verglichen und mit Unternehmensmitarbeitern auf Anwendbarkeit überprüft. Gemeinsam mit den Unternehmen werden **Herausforderungen in der Informationssystem-Architektur** abgeleitet sowie die Architektur der Unternehmen zu den fünf **Architekturtypen** zugeordnet. Dies wird im folgenden Schritt genutzt, um die im Erklärungsmodell vorgestellten **Handlungsfelder zur Umsetzung der Zielarchitektur** zu verifizieren. Im Gespräch mit den Unternehmen kann geprüft werden, wie die Umsetzung der Handlungsempfehlungen priorisiert werden müssen und inwieweit dies mit den in der vorliegenden Arbeit vorgestellten Entscheidungskriterien harmonisiert.

Die **Hinweise zur Auswahl und Einführung** einer software-definierten Plattform werden den anwendenden Unternehmen als Hilfsmittel für weitere Einführungen an die Hand gegeben, jedoch **primär mit dem Softwarehersteller** PTC validiert. In der Fallstudie wird überprüft, inwieweit die vorgeschlagenen Vorgehensweisen die Auswahl und Einführung tatsächlich unterstützen können.

9.2 Fallstudie 1: e.GO Mobile AG

9.2.1 Das Unternehmen e.GO Mobile AG

Die e.GO Mobile AG ist ein im März 2015 gegründeter **Elektrofahrzeughersteller mit Sitz in Aachen**. Das Unternehmen hat mit drei festangestellten Mitarbeitern in einem kleinen Besprechungsraum auf dem RWTH Aachen Campus begonnen. Seitdem ist es sowohl räumlich als auch personell stark gewachsen: zum Ende des Jahres 2017 auf bereits **über 120 Mitarbeiter**, die auf zwei Standorte in Aachen verteilt sind. e.GO hat sich zum Ziel gesetzt, **bezahlbare und praktische Elektromobilität zu schaffen, „die Spaß macht“** (E.GO MOBILE 2018). Dazu entwickelt e.GO nicht nur den **e.GO Life**, ein Stadtauto, das voraussichtlich ab Mitte 2018 produziert wird, sondern auch ein Pe-delec mit vier Rädern, das **e.GO Kart**, sowie einen autonom fahrenden kleinen Stadtbus, den **e.GO Mover**. Weitere Fahrzeuge und Fahrzeugderivate sind geplant. Für die Entwicklung und Produktion wurden verschiedene Kooperationen und Konsortien mit großen Zulieferern der Automobilindustrie wie Bosch und ZF geschlossen, um deren Know-how zu nutzen und gemeinsam an neuen Komponenten für die Elektromobilität zu forschen.



Abbildung 9-2: Produktionsstätte von e.GO in Aachen (entnommen aus E.GO MOBILE 2017)

e.GO verfolgt eine **intensive Wachstumsstrategie**. So sollen im Jahr 2018 knapp 2.000 Fahrzeuge hergestellt werden, für 2019 sind bereits 15.000 Fahrzeuge geplant. Insgesamt plant e.GO mit seinen Produkten zu den fünf größten deutschen Herstellern im E-Mobilitätsmarkt zu gehören (s. MÜLLER 2017). Um dies zu erreichen, sind neben den Aachener Produktions- und Prototypenwerkstätten weitere **Werke im Ausland geplant**. Dort sollen sie möglichst als Franchisefabriken durch lokale Unternehmen

betrieben werden und eine Kopie der Aachener Fabrik darstellen. Diese wird als „Industrie-4.0-Vorzeigefabrik“ (E.GO MOBILE 2017) aufgebaut und unter anderem über eine vollständige Teileverfolgung und Transparenz über die aktuellen Aufträge verfügen, aber auch flexibel auf veränderte Fahrzeugkonzepte anpassbar sein. Die dort umgesetzten Konzepte werden zuvor in den Prototypenwerkstätten des Unternehmens getestet. Dazu nutzt e.GO seine Tochterfirmen *e.GO Digital* und das *European 4.0 Transformation Center* (E4TC), welche die Entwicklung von Apps sowie die Nutzung der unternehmensinternen Daten und der Produktdaten in Partnerschaft von Software- und Hardwarelösungsanbietern vorantreiben.

9.2.2 Ausgangssituation bei e.GO

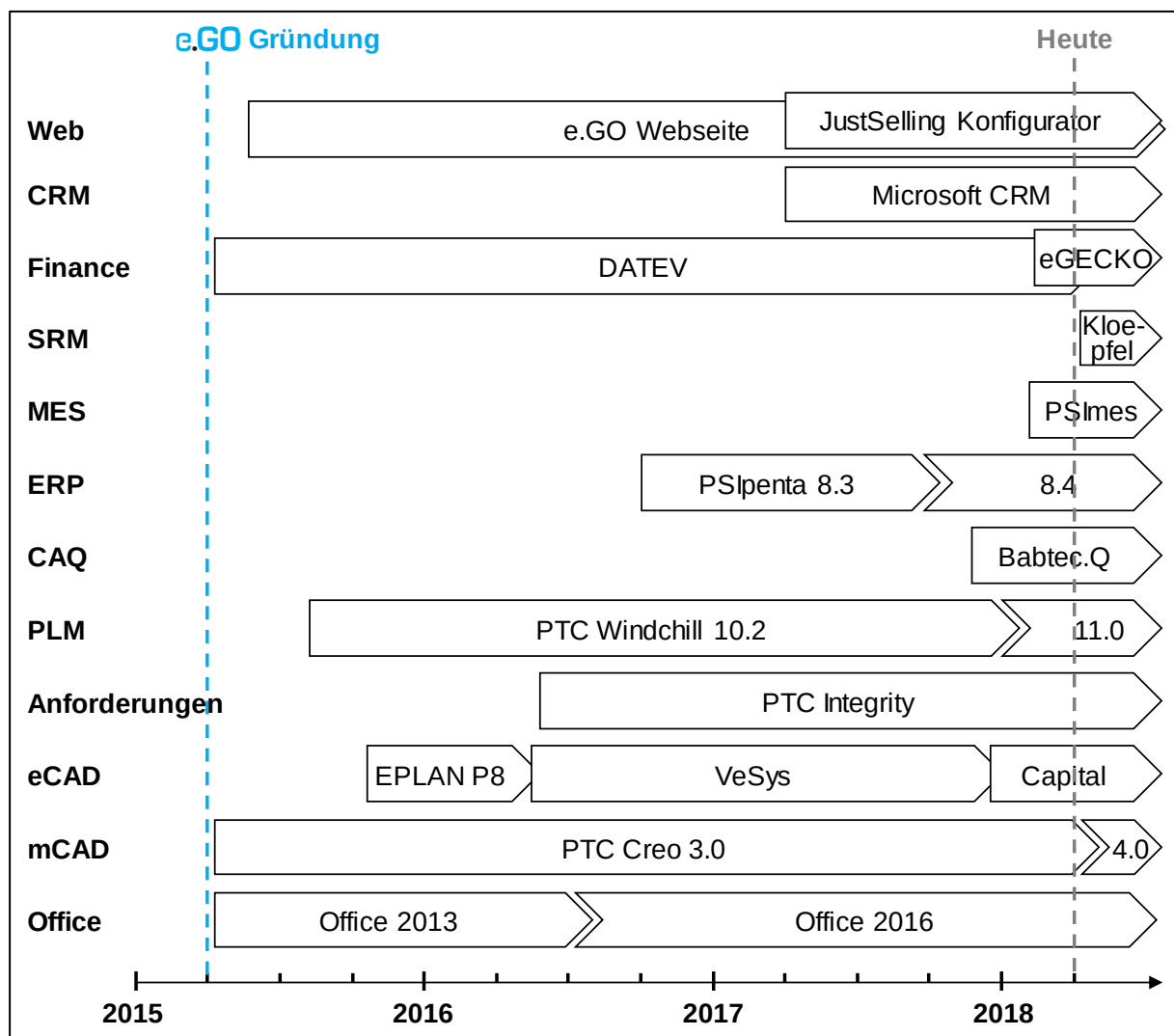


Abbildung 9-3: Historie der Informationssystem-Einführung bei e.GO (eigene Darstellung zum Stand März 2018)

e.GO hat sich **entlang des Produktlebenszyklus entwickelt**. Zwar war von Beginn der Unternehmung an für jede Phase des Produktlebenszyklus eine verantwortliche Person bestimmt, es wurde dann jedoch zuerst in der Entwicklung Personal aufgebaut, gefolgt vom Einkauf, Prototypenbau und der Produktionsplanung. Ebenso wurden, neben E-Mail-, Office und Buchhaltungsanwendungen, zuerst das CAD- und danach das

PLM-System zur Unterstützung der Entwicklungsprozesse bei e.GO eingeführt. Zu der Zeit haben die Bereiche Einkauf und Produktionsplanung noch mit einfachen Excel-Listen gearbeitet, bis im September 2016 das ERP-System eingeführt wurde. Auch Kontaktdaten potentieller Kunden und sonstiger Leads wurden über Monate in Excel-Listen geführt und nach Einführung der CRM-Systems im Februar 2017 in dieses übernommen. In Vorbereitung auf den Produktionshochlauf wurden in jüngerer Zeit weitere Systeme wie ein Supplier-Relationship-Management-System (SRM-System), ein MES und ein Computer-aided-Quality-System (CAQ-System) eingeführt (s. Abbildung 9-3). Somit sind zum Stand März 2018 die **zentralen Informationssysteme des Unternehmens eingeführt**. Auch eine Reihe an weiteren Anwendungssystemen, wie ein Anforderungsmanagementtool oder Anwendungen für statistische Analysen, wurden ausgewählt und installiert. Allerdings sind die Systeme **hauptsächlich als nicht-integrierte Systeme** implementiert. Dies liegt ebenso in der Historie begründet: Die Informationssysteme mussten zuerst aufgebaut, die Prozesse und Datenstrukturen innerhalb der Systeme etabliert und von den Mitarbeitern erlernt werden. Erst mit gefestigten Strukturen kann nun in einem weiteren Schritt eine verstärkte Integration der Systeme angegangen werden.

Die grundlegenden Strukturen einer systemübergreifenden Datennutzung waren bei e.GO schon von Beginn an geplant. So wurde das ERP-System so konfiguriert, dass es dieselben Teilenummern nutzt wie das PLM-System. Damit wurde auch die Übergabe der mBOM, die im PLM-System aus der eBOM heraus abgeleitet wird, über eine indirekte Schnittstelle zum ERP-System ermöglicht. Die Umsetzung von vordefinierten Schnittstellen zwischen den Systemen ist jedoch bei e.GO bisher nur zwischen den Systemen der *PSI Automotive&Industry*, wie *PSIpenta* und *PSIwms*, sowie PSI-Partnerprodukten, wie dem *Babtec.Q* CAQ-System, umgesetzt. Auch ein **systemübergreifender Datenzugriff ist noch nicht möglich**, allerdings wurde in einigen Anwendungsfällen bereits der Einsatz einer software-definierten Plattform getestet; weitere Anwendungsfälle sind zum Zeitpunkt der Abfassung der vorliegenden Arbeit in Umsetzung.

Zur Bereitstellung der beschriebenen Lösungen arbeiten inzwischen 13 Mitarbeiter und betreuen damit etwa 300 Anwender des Unternehmens, was einem überdurchschnittlichen Betreuungsverhältnis von 23 Anwendern pro Mitarbeiter entspricht (s. Abbildung 6-23 in Kapitel 6.8). Hinsichtlich der IT-Organisation hat sich e.GO für ein Modell entschieden, bei dem der **Infrastrukturbetrieb komplett extern vergeben** ist. So können sich die **e.GO-eigenen Mitarbeiter um die strategische Weiterentwicklung der Informationssystem-Architektur** sowie die Einführung und Konfiguration der Informationssysteme, in Zusammenarbeit mit dem Fachbereich, kümmern. Die Mitarbeiter sind nach Systemkonzepten aufgeteilt organisiert, unterstützt durch eine Fachkraft, welche die Datendurchgängigkeit im Kontext des *Model Based Systems Engineering* (MBSE) vorantreibt. Weiterhin ist der Bereich Produktdigitalisierung („Connected Service“), wie auch das E4TC, als vorgelagertes Demonstrationsumfeld, Teil der IT-Organisation (s. Abbildung 9-4). Die schnelle Entwicklung von Apps auf Basis einer software-definierten Plattform ist unter Nutzung der Tochtergesellschaft

e.GO Digital geplant, aber aufgrund der noch nicht funktionalen zentralen Plattform (s. Kapitel 9.2.4) aktuell noch nicht umgesetzt.

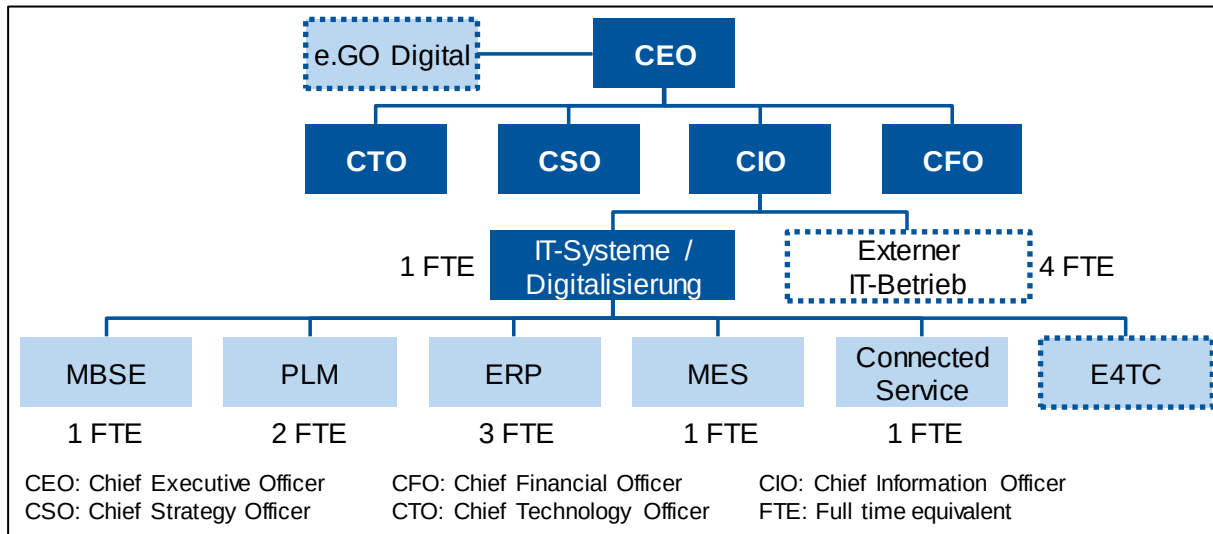


Abbildung 9-4: Organigramm der IT-Organisation bei e.GO (eigene Darstellung zum Stand Februar 2018)

Aus technischer Sicht ist der **größtenteils interne Systembetrieb** zu nennen. Mit Ausnahme des CRM-Systems und einiger kleinerer Tools werden alle Informationssysteme durch den IT-Dienstleister auf internen Servern von e.GO gehostet. Die Migration zu einem Cloud-Anbieter befindet sich am Anfang der Planungsphase. Ebenfalls mit Ausnahme des CRM-Systems **kann auf Unternehmensdaten aktuell nicht ubiquitär zugegriffen werden**, sondern rein unter Nutzung einer VPN-Lösung.

Eine vertiefende Sicht der Informationssystem-Architektur bei e.GO im Ausgangszustand liefert die Beschreibung in Kapitel 9.2.4.

9.2.3 Ziele und Anforderungen software-definierter Plattformen bei e.GO

Um die zentrale Nutzung der software-definierten Plattform voranzutreiben geht e.GO den Weg, schrittweise Anwendungsfälle umzusetzen. Zum Stand März 2018 sind einige Anwendungsfälle definiert, wovon ein Teil bereits prototypisch umgesetzt ist. **Die prototypischen Umsetzungen sind bisher noch isolierte Lösungen**, die keine Verbindung untereinander haben. Die Anwendungsfälle werden im Folgenden kurz vorgestellt. Sie dienen dazu, wie in Kapitel 8.1.1 beschrieben, die Aufgaben software-definierter Plattformen abzuleiten. Im Fall von e.GO dient dies dazu, um zu überprüfen, ob die gewählte Plattform den Anforderungen entspricht.

Anwendungsfall 1: *Engineering Dashboard*

Ziel des Anwendungsfalls „*Engineering Dashboard*“ ist es, **Transparenz über den Status des agilen Entwicklungsprozesses für ein einzelnes Produkt** zu erhalten. Das Dashboard soll dazu alle relevanten Daten über den Entwicklungsstand des Produkts auf einer Oberfläche visualisieren und damit dem Management eine einfache Entscheidungs- und Eingriffsmöglichkeit geben, ohne dass es sich die Einzeldaten mühsam zusammensuchen muss (s. Abbildung 9-5). Bei e.GO ist die prototypische

Umsetzung so erfolgt, dass aufgeteilt nach den sogenannten „*Lead Engineering Groups*“ (LEGs) der Status der Detaillierung der Bauteile sowie das Gewicht der Bauteile dargestellt wurde (s. SCHUH ET AL. 2018, S. 331). Das Gewicht hat bei einem Kleinwagen, wie dem e.GO Life, eine besondere Bedeutung, vor allem auch deshalb, weil ein höheres Gewicht die Reichweite des Fahrzeugs verringert (s. RAHIMZEI 2015, S. 3). Weiterhin wird der Entwicklungsstand auch grafisch als dreidimensionale Visualisierung dargestellt.



Abbildung 9-5: Prototypische Umsetzung des *Engineering Dashboards* (Screenshot von e.GO)

Um dieses Dashboard umzusetzen, sind **Daten aus den Informationssystemen aus dem Bereich der Entwicklung notwendig**. Anforderungsdaten, wie das Zielgewicht einer Baugruppe, sind im Anforderungsmanagementtool (bei e.GO: *PTC Integrity*) hinterlegt. Die tatsächlichen Gewichtsangaben und der Status der Entwicklung werden im PLM-System (bei e.GO: *PTC Windchill*) gespeichert (s. SCHUH ET AL. 2018, S. 331). Dort werden auch die CAD-Daten aus den eCAD- und mCAD-Tools *PTC Creo* und *Mentor Capital* aggregiert. Um das Dashboard umzusetzen, mussten diese Systeme mit einem Mandanten der bei e.GO eingesetzten software-definierten Plattform *PTC ThingWorx* über Schnittstellen verbunden werden. Da alle Systeme vom Softwarehersteller PTC stammen und über vordefinierte Schnittstellen verfügen, war dies im betreffenden Fall relativ einfach möglich.

Anwendungsfall 2: End-to-End-Monitoring in der Fabrik

Das End-to-End-Monitoring soll es bei e.GO ermöglichen, jederzeit Transparenz über den Zustand und Ort der Assets in der Produktion sowie den Status von Aufträgen zu bekommen. Dies beinhaltet **detaillierte Informationen über die Maschinen in der Produktion, die Anbindung von verfügbarer Sensorik wie auch eine entspre-**

chende Visualisierung. Damit wird der digitale Schatten, wie in Kapitel 5.1.1 beschrieben, verwirklicht. Die Darstellung gibt Mitarbeitern die einfache Möglichkeit, Daten aus der Produktion zu nutzen und notwendige Entscheidungen daraus abzuleiten. Eine prototypische Umsetzung erfolgte in der Prototypenfabrik von e.GO (s. Abbildung 9-6). Sie zeigt eine Übersicht mit farblichen Hervorhebungen für einzelne Assets sowie die Möglichkeit zum Hineinzoomen in einzelne Objekte. Im unten dargestellten Screenshot werden beispielsweise Lagerbestände eines Kleinteilelagers angezeigt. Ist der Mindestbestand unterschritten, wird eine rote Markierung dargestellt.



Abbildung 9-6: Prototypische Umsetzung des End-to-End-Monitorings in einer Prototypenfabrik (Screenshot von e.GO / E4TC)

Zur Umsetzung dieses Anwendungsfalls müssen die Schnittstellen der verschiedenen anzubindenden OT- und Anwendungssysteme bekannt sein. Schon bei der prototypischen Umsetzung zeigten sich große Herausforderungen. So verfügte nahezu **jedes Objekt, welches als Datenquelle in die Plattform integriert werden sollte, über eigene, hochspezialisierte Schnittstellen**. Die Schnittstellen konnten teilweise genutzt werden, zum Teil mussten Wrapper-Funktionen zur Datenstandardisierung geschrieben werden (s. STICH ET AL. 2018, S. 39). Genauso gab es auch organisatorische und die Daten betreffende Integrationsherausforderungen, die bei e.GO für die Umsetzung des Anwendungsfalls im produktiven Werk besonders beachtet werden müssen.

Anwendungsfall 3: Digitaler Zwilling und Simulation

Im Gegensatz zum digitalen Schatten, der das aggregierte digitale Abbild eines Teils des Unternehmens abbildet, ist der **digitale Zwilling definiert als sein digitales Abbild eines physischen Produkts** (s. KUHN 2017, S. 440). e.GO möchte für alle seine Produkte im Markt solche digitalen Zwillinge erstellen, um die vollständigen Daten der

Produkte im Feld nutzen zu können und damit auch Kunden Mehrwerte, wie beispielsweise eine verbesserte Fahrzeugverfügbarkeit, bereitstellen zu können.

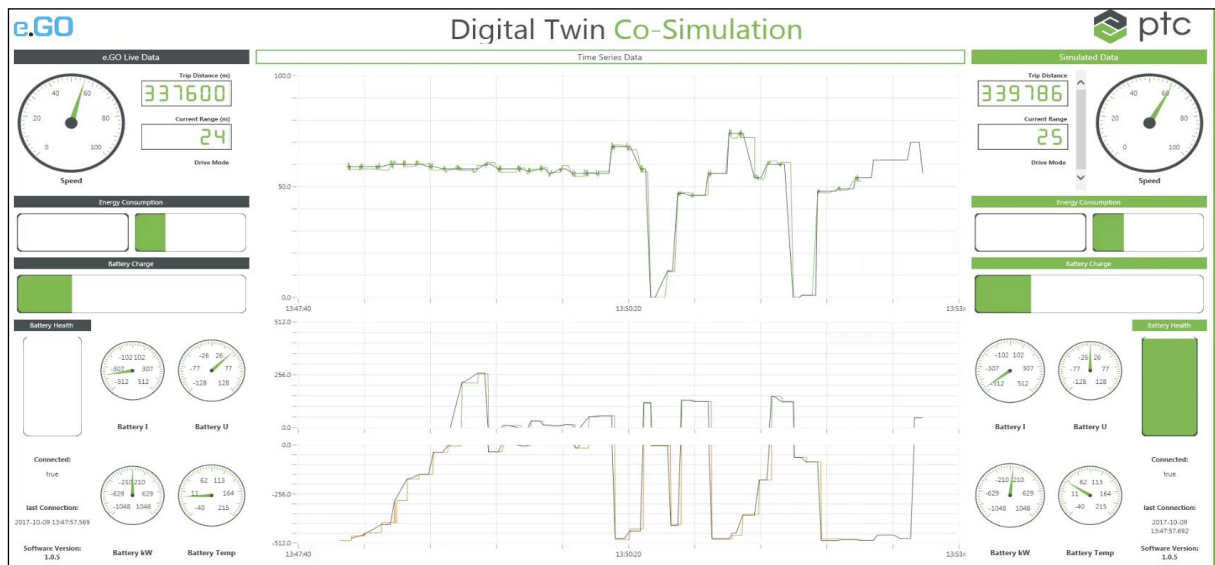


Abbildung 9-7: Prototypische Umsetzung des digitalen Zwillings (Screenshot von e.GO)

Aber auch in der Prototypenphase, in der sich der e.GO Life zum Zeitpunkt der Abfassung der vorliegenden Arbeit befindet, möchte e.GO den digitalen Zwilling des Fahrzeugs nutzen, um es zu verbessern. Hierzu wird, dem MBSE-Ansatz folgend, ein **SysML-Modell des e.GO Life** genutzt. Dieses wird erstellt, um **mit entsprechenden Simulationsmodellen Simulationen des Fahrzeugverhaltens durchzuführen** (s. Abbildung 9-8). Durch den Vergleich von Simulationsdaten mit den tatsächlichen Daten aus dem digitalen Zwilling können die Simulationsmodelle auf ihre Validität überprüft werden. Mit validen Modellen wiederum kann eine Vielzahl von weiteren Testfällen virtuell durchgeführt werden, was die Notwendigkeit physischer Tests reduziert.

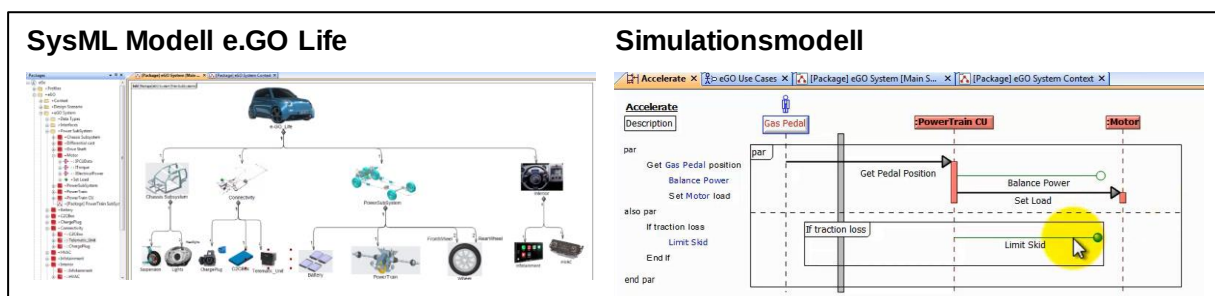


Abbildung 9-8: Model-based Systems-Engineering im e.GO Life (Screenshots von e.GO)

Auch dies ist ein Anwendungsfall der software-definierten Plattform. Die Darstellung und Zusammenfassung der Daten, wie in Abbildung 9-7 prototypisch gezeigt, erfolgt dort auf Basis von Daten aus dem PLM-System (bei e.GO: *PTC Windchill*) und dem Anforderungsmanagementtool (bei e.GO: *PTC Integrity*), sowie aus Daten, die aus einer über OBD II angeschlossenen sogenannten Car-to-Cloud-Box im Fahrzeug gewonnen werden. Die Herausforderung liegt hierbei vor allem an der korrekten Zuordnung der verbauten Fahrzeugteile, der gesendeten Daten und der Simulationsmodelle zum digitalen Zwilling.

Anwendungsfall 4: Übergreifende Qualitätsprüfung

Bei der laufenden Neueinführung des *Babtec.Q-CAQ*-Systems stellt sich bei e.GO die Frage, wie der **Qualitätsingenieur die Teilequalität möglichst aufwandsarm prüfen** kann. Für seine Arbeit benötigt er neben dem CAQ-System, das qualitätsrelevante Daten dokumentiert, analysiert und archiviert (s. BABTEC 2018), auch Daten aus dem ERP- und dem PLM-System sowie Visualisierungen aus dem CAD-System. Damit der Mitarbeiter nicht in vier Informationssystemen gleichzeitig arbeiten muss, wurde nach einer Lösung gesucht, die einen übergreifenden Zugriff ermöglicht.

Zur Umsetzung wurden zwei Varianten geprüft: Die erste Variante ist die Implementierung von Schnittstellen zwischen dem CAQ- sowie dem ERP- und dem PLM-System. Als zweite Variante ist eine Lösung unter Nutzung der software-definierten Plattform, welche in einem **Dashboard anhand der Artikel-ID alle für den Qualitätsingenieur relevanten Informationen anzeigen** kann, angedacht. Für die Umsetzung wird letztere Lösung favorisiert, weil sie einen geringeren Verkopplungsgrad der Systeme und damit eine höhere Flexibilität ermöglicht. Diese Lösung ist zum Zeitpunkt der Abfassung der vorliegenden Arbeit noch nicht umgesetzt.

Anwendungsfall 5: Prozess für Sonderbestellung

Das Fahrzeug e.GO Life wird in einem agilen Entwicklungsprozess in vergleichbar kurzer Zeit entwickelt (s. SCHUH ET AL. 2017c, S. 40-41). Dies führt dazu, dass kurz vor Produktionsbeginn noch nicht alle Fahrzeugteile final fertiggestellt und freigegeben sind. Um trotzdem die ersten Fahrzeuge frühzeitig produzieren zu können, müssen **Langläuferteile in einem Sonderbestellprozess freigegeben und daraufhin bestellt werden**. Dazu wird ein Prozess benötigt, der, zeitlich begrenzt, vom definierten Standardbestellprozess abweicht und trotzdem hohen Qualitätsansprüchen genügt. Abbildung 9-9 gibt einen Überblick über diesen Prozess.

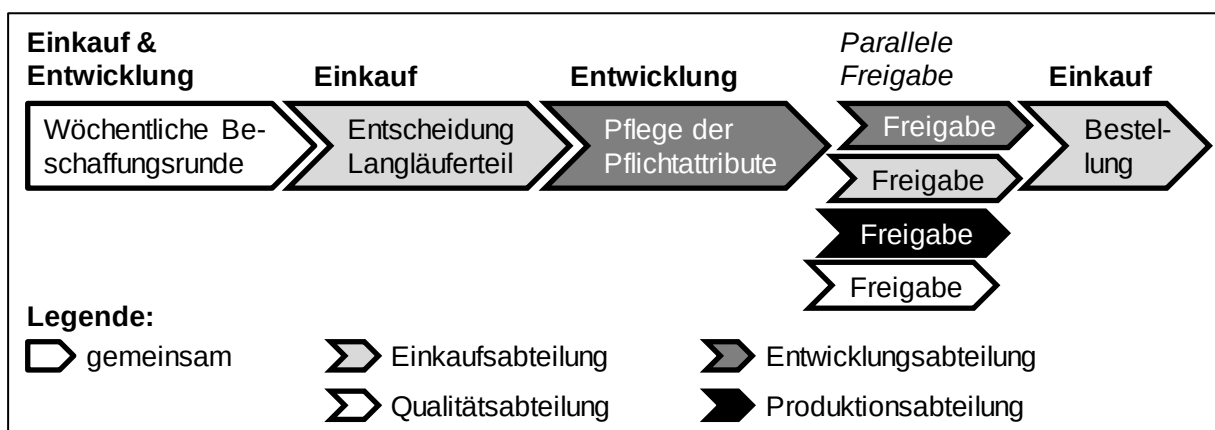


Abbildung 9-9: Übersicht Sonderbestellungsprozess e.GO Life (eigene Darstellung)

Es sind dabei die Einkaufs- und die Entwicklungsleitung (die LEG-Leiter) beteiligt, die sich in einer wöchentlichen Beschaffungsrunde speziell der Teile annehmen, die zur rechtzeitigen Lieferung bestellt werden müssen. Außerdem müssen die Teile auch von der Produktions- und der Qualitätsabteilung freigegeben werden. Dazu werden Daten

aus dem ERP- und dem PLM-System benötigt, um während der Besprechung Teileinformationen wie Material oder Größe zu bestimmen. Weiterhin wird ein Workflow benötigt, welcher die Pflege der Pflichtattribute und die Freigabe des Teils durch den jeweiligen LEG-Leiter sowie die anderen beteiligten Abteilungen bis hin zur Übergabe an den Einkauf systemübergreifend steuert.

Um dies zu ermöglichen, plant e.GO die Umsetzung einer Freigabe-App über seine software-definierte Plattform (bei e.GO: *PTC ThingWorx*). Die notwendigen Teiledaten aus dem ERP- und PLM-System werden ausgelesen und in der App angezeigt. Damit bekommen **die Führungskräfte einen einfachen Überblick über die zu besprechenden Teile**. Die App verfolgt dann den Freigabestatus und auch die Informationsqualität der gepflegten Attribute der Teile. Nach der Entscheidung durch den LEG-Leiter übergibt die App ein Token an das ERP-System, damit die Einkaufsabteilung die Bestellung starten kann. Durch entsprechende Attribute (Releasestatus „Sonderbestellung“) können die so bestellten Teile eindeutig identifiziert und später wieder in den Standardprozess im ERP-System übernommen werden.

Anwendungsfall 6: Fahrzeugdatenstruktur

Die Fahrzeugdaten eines e.GO Life sind mehrdimensional und bauen sich erst im Laufe des Produktlebenszyklus auf. Zusammengefasst kann von den folgenden Dimensionen gesprochen werden:

- Das *Produkt* mit seiner Produktstruktur, die im Engineering als 150-Prozent-Stückliste entworfen wurde;
- die *Produktvariante*, welche eine Teilmenge der Produktstruktur darstellt, angereichert um die variantenspezifischen Informationen;
- die *Kundenbestellung*, welche sich auf eine Variante bezieht, der jedoch eine spezifische Fahrzeugidentifikationsnummer (VIN) zugeordnet wird;
- die *Produktionsinformationen*, welche die tatsächlichen Arbeitsmittel, eingebauten Teile etc. enthalten;
- *Nutzungs- und Serviceinformationen* über das Fahrzeug, die in dessen Lebenszeit fortlaufend entstehen.

Abbildung 9-10 gibt eine Übersicht über die zum Zeitpunkt der Abfassung der vorliegenden Arbeit bei e.GO betrachteten Daten, die über die zentrale software-definierte Plattform zusammengefasst werden sollen. Weitere Daten aus der Fahrzeugnutzung werden folgen. Damit wird die grundlegende Datenstruktur gebildet, die für die Umsetzung des digitalen Zwillings benötigt wird (s. o.).

Zentrales Element in der Fahrzeugdatenstruktur ist die VIN. Diese wird generiert,⁴⁹ sobald ein Kunde ein Fahrzeug bestellt. Die VIN bezieht sich auf eine bestimmte Fahrzeugvariante, für welche die notwendigen Arbeitsgänge im ERP ausgeleitet und dem MES übergeben wurden. Die VIN wird bei der Montage des Fahrzeugs auf den Fahrzeugrahmen graviert. Um später, beispielsweise bei einem Unfall, genau feststellen zu können, welche Arbeitsmittel zur Montage des Fahrzeugs genutzt wurden, wird die Fahrzeugakte im MES angelegt und über die software-definierte Plattform alle **Produktionsinformationen mit der VIN verknüpft**.

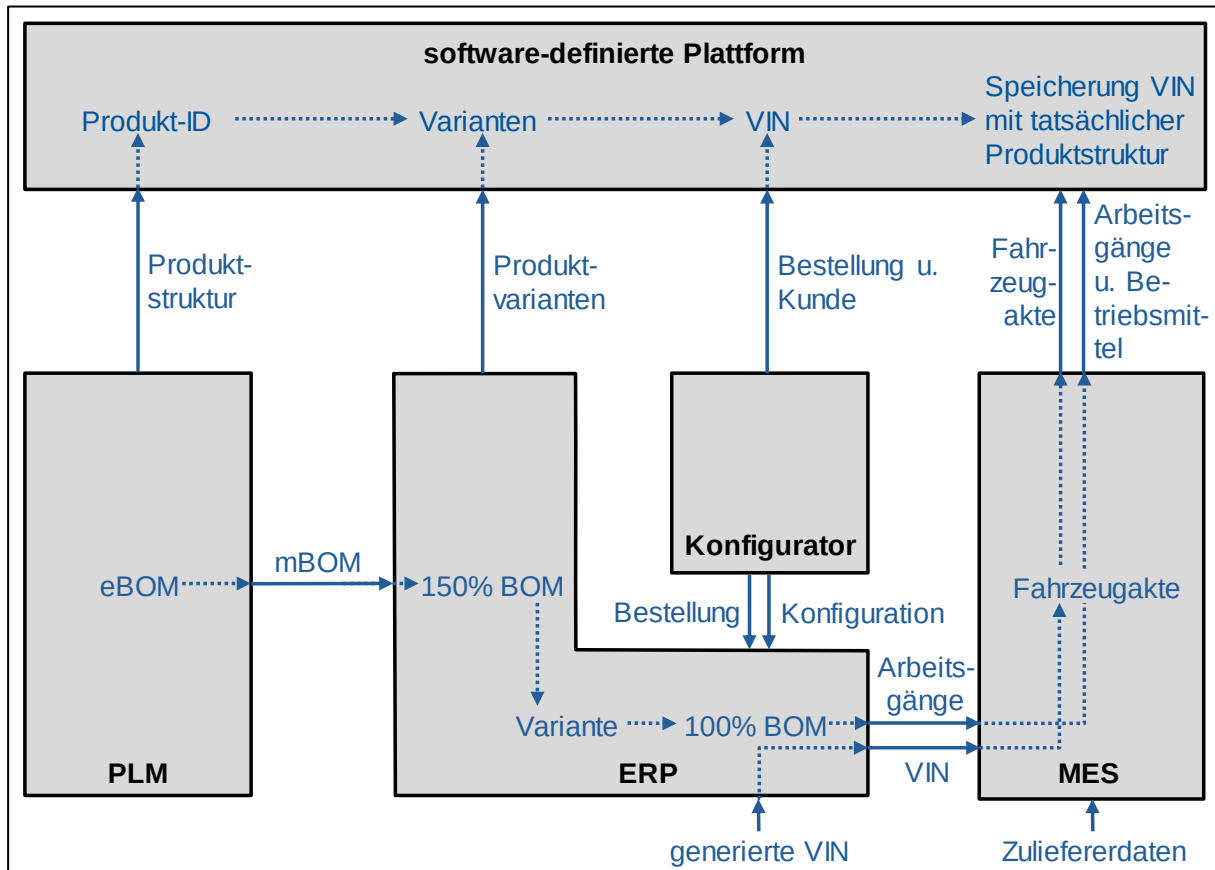


Abbildung 9-10: Entstehung der Fahrzeugdatenstruktur des e.GO Life (eigene Darstellung)

Da keines der zentralen Informationssysteme bei e.GO alle relevanten Informationsdimensionen erfasst, hat e.GO entschieden, die Datenverknüpfung in der software-definierten Plattform vorzunehmen. Dort werden **für jedes Fahrzeug die benötigten Informationen im Laufe des Lebenszyklus ergänzt**. Dazu werden ein entsprechendes Datenmodell sowie eine Anbindung des PLM- und ERP-Systems, des MES und des Kundenkonfigurators benötigt.

⁴⁹ Der Prozess der VIN-Generierung ist an sich komplizierter, da alle VINs durch das Kraftfahrtbundesamt zertifiziert werden müssen. Dieser Schritt ist für den vorliegenden Anwendungsfall aber nicht relevant, weshalb davon abstrahiert wurde.

Übersicht der Anwendungsfälle

Tabelle 9-1 fasst die Aufgaben der software-definierten Plattform für die genannten sechs Anwendungsfälle zusammen. Diese Zusammenfassung dient, wie in Kapitel 8.1.1 beschrieben, der ersten Einschätzung darüber, welcher Typ von software-definierter Plattform bei e.GO benötigt wird. **Die bisher geplanten Anwendungsfälle decken** dabei – wenn auch in unterschiedlich hohem Grad – schon **alle Aufgaben einer software-definierten Plattform ab**. Ein besonderer Fokus liegt dabei auf der Einbindung von Umsystemen, der Datenmodellierung und der Datensicherheit. All dies passt zu dem Ziel von e.GO, eine datengetriebene Unternehmung zu werden.

Tabelle 9-1: Aufgaben der software-definierten Plattform bei e.GO (eigene Darstellung)

Aufgabe	Anwendungsfall						Summe
	1	2	3	4	5	6	
Daten-bereitstellung	x	x	x			x	4 67 %
Einbindung Umsysteme	x	x	x	x	x	x	6 100 %
Quellen-management		x	x		x		3 50 %
Daten-modellierung	x	x	x	x	x	x	6 100 %
Datenanalyse		x	x				2 33 %
Daten-aufbereitung	x			x	x		3 50 %
Datensicherheit	x	x	x	x	x	x	6 100 %

Aus den genannten Aufgaben lassen sich die **Eigenschaften der einzusetzenden software-definierten Plattform** ableiten. e.GO benötigt eine Typ-I-Plattform, also eine Plattform, die den kompletten Funktionsumfang software-definierter Plattformen bereitstellen kann. e.GO hat sich für die Plattform *ThingWorx* von PTC entschieden, da diese über eine hohe Kompatibilität mit den weiteren eingesetzten Systemen, wie dem PLM- und mCAD-System, sowie dem Anforderungsmanagementtool verfügt. Die folgende Tabelle 9-2 gibt eine Übersicht über die Eigenschaften der eingesetzten Plattform. Diese deckt die Typ-I-Kriterien fast vollständig ab, allein beim Datenbankkonzept gibt es insofern Einschnitte, als dass eine NoSQL-Datenbank nur über Konnektoren als zusätzliche Datenquelle verbunden, aber nicht nativ von der Plattform zur Datenspeicherung genutzt werden kann. **e.GO hat also eine software-definierte Plattform**

ausgewählt, die zu den Anforderungen aus den beschriebenen Anwendungsfällen passt.

Tabelle 9-2: Eigenschaften der software-definierten Plattform *PTC ThingWorx* bei e.GO (eigene Darstellung)

Merkmal	Ausprägungen			
<i>Schnittstellen</i>	generische Schnittstellen	vordefinierte Schnittstellen		
<i>Datenmanagement</i>	feste Datenstruktur	Datenharmonisierung und -bereinigung	Datenanreicherung	frei konfigurierbare Datenobjekte
<i>Datenbankkonzept</i>	relationale Datenbank	NoSQL-Datenbank	sonstige Datenbankkonzepte	
<i>Stufen der Datenverarbeitung</i>	Erzeugen von Sichtbarkeit	Transparenz und Diagnose	Prognose	Adaption und Entscheidungsautomatisierung
<i>IT-Sicherheitsfunktionen</i>	nur interne Sicherheitsfunktionen	Sicherheit für Plattform und angebundene Systeme	Vollintegration in IT-Sicherheitsinfrastruktur des Unternehmens	
<i>Informationsbereitstellung</i>	keine eigene Auswertungsfunktion	Bereitstellung von Auswertungen und Visualisierungen	Generierung eigener Apps/Widgets	
<i>Entwicklungsumgebung (E.)</i>	keine eigene E.	eigene, proprietäre E.	offene E.	

Legende:

	teilweise erfüllt		vollständig erfüllt

Integrationsanforderungen der software-definierten Plattform

Da, wie oben beschrieben, die software-definierte Plattform bei e.GO bisher nur in Einzelfällen prototypisch genutzt wird, steht die vollständige Integration zum Zeitpunkt der Abfassung der vorliegenden Arbeit noch aus. Es existiert eine Reihe von konkreten Integrationsanforderungen, die im Folgenden zusammengefasst werden (s. Tabelle 9-3).

Tabelle 9-3: Integrationsanforderungen und Konkretisierung bei e.GO (eigene Darstellung)

Dimension	Integrationsanforderung	Konkretisierung e.GO
<i>Technik</i>	passendes Betriebskonzept	franchisefähiges, internationalisierungstaugliches Betriebskonzept ist auszuarbeiten
<i>Technik</i>	rekonfigurierbare Anbindung	Verringerung Point-to-Point Schnittstellen umsetzen
<i>Technik</i>	technisches Anbindungskonzept	Schnittstellen möglichst über ThingWorx umsetzen
<i>Technik</i>	echtzeitfähige Anbindung	Umsetzung von 5G in der Produktion vorantreiben
<i>Organisation</i>	integriertes Sicherheitskonzept	Informationssicherheitskonzept muss entwickelt werden
<i>Organisation</i>	geregelte Inhaltsverantwortung	Festlegung der Verantwortung über konkretes Datenmodell
<i>Prozessual</i>	Entwicklung initialer Anwendungsfälle	Anwendungsfälle für Production- und User-Cycle ausbauen
<i>Prozessual</i>	Soll-Prozesse entwickelt	Überarbeitung von Prozessen wie dem Änderungsprozess auf App-Basis
<i>Daten</i>	hohe Informationsqualität	Informationsqualitätsmanagement als Führungsaufgabe etablieren
<i>Daten</i>	Transparenz über vorhandene Daten	Komplettaufnahme des Datenmodells umsetzen
<i>Daten</i>	einheitliches Kerndatenmodell	Anpassungen an Informationssystemen auf Basis des aufgenommenen Datenmodells

Die Integrationsanforderungen bei e.GO lassen sich anhand der vier Integrationsdimensionen (s. Kapitel 5.6.1) strukturieren. In der obigen Tabelle ist eine für e.GO relevante Auswahl der allgemeinen Integrationsanforderungen, die in Kapitel 5.6 der vorliegenden Arbeit erarbeitet wurden (s. Abbildung 5-12), konkretisiert. Im Bereich der **technischen Integrationsanforderungen** sind vor allem das Betriebskonzept und die Schnittstellenarchitektur hervorzuheben. Ein **Betriebskonzept** ist nach Aussage des IT-Leiters bisher nicht erarbeitet worden. Vor dem Hintergrund, dass schon wenige Monate nach Abfassung der vorliegenden Arbeit ein IT-Betriebskonzept für ein Werk in Asien fertiggestellt sein soll, gleichzeitig ein Konzept für das deutsche Werk aber

noch im Aufbau ist, liegen dort noch große Herausforderungen. Da die e.GO-Strategie den Aufbau von Franchise-Fabriken vorsieht, muss ein für die deutschen und ausländischen Standorte passendes Hosting-Konzept entwickelt werden, welches die Grundlage für den reibungslosen Betrieb der software-definierten Plattform bildet. Insbesondere muss die geforderte Plug-and-Play-Fähigkeit der Informationssystem-Architektur so entwickelt werden, dass sie einerseits einfach umsetzbar ist und andererseits auch unberechtigte Datenzugriffe unterbindet.⁵⁰

Auch die **Schnittstellenarchitektur** muss entwickelt werden. Für die technische Umsetzung gilt dabei, dass Datennetze verfügbar sein müssen, welche die schnelle Datenübertragung in der Fertigung ermöglichen. Es wird zwar keine harte Echtzeitübertragung benötigt, jedoch eine Datenübertragung von hoher Qualität mit möglichst hohem Datendurchsatz. Eine veränderbare Informationssystem-Architektur erfordert weiterhin, dass Schnittstellen über den Backbone, also die software-definierte Plattform, umgesetzt werden. Somit kann die Kompliziertheit des Schnittstellennetzwerks geringgehalten werden.

Aus **organisatorischer Sicht** muss im Bereich der **Informationssicherheit** ein Konzept entwickelt werden. Neben technischen Vorgaben für die IT-Sicherheit sind hier vor allem Regelwerke für die Mitarbeiter bezüglich des Datenschutzes sowie klar definierte Rollen für die Umsetzung der Informationssicherheit festzulegen. Nicht nur in Bezug auf den Sicherheitsaspekt, sondern auch aus Sicht der **Qualität der verarbeiteten Informationen** müssen klare Verantwortungen definiert werden. Daher sollten bei e.GO, anhand des geplanten zentralen Datenmodells, auch Informationsverantwortliche aus den Fachbereichen definiert werden.

Die im Zuge der Integrationsanforderungen definierte Vorgehensweise, anhand von **Anwendungsfällen** die Einbindung der software-definierten Plattform in die Informationssystem-Architektur festzulegen, wurde bei e.GO für entwicklungsnahe Bereiche schon umgesetzt. Für eine übergreifende Nutzung sollte dies jedoch auch für weitere Bereiche, wie die Produktion und Produktnutzung, geschehen. Darüber hinaus fordert die Integrationsanforderung der Prozessintegration, dass klare **Soll-Prozesse** entwickelt werden, welche die Einbindung der verschiedenen Informationssysteme darstellen. e.GO hat hier erste Ansätze unternommen, indem Unternehmensprozesse in einer Prozessplattform zusammengefasst und für die Mitarbeiter bereitgestellt werden.

In Bezug auf die Datenintegration sind neben der schon angesprochenen „Informationsqualität als Führungsaufgabe“ die Anforderungen an ein zentrales Datenmodell sowie eine hohe Informationsqualität anzusprechen. Um Transparenz über den aktuellen Zustand zu bekommen, sollte e.GO hier zuerst die schon begonnenen Anstrengungen zur Aufnahme der aktuellen Datenstruktur fortsetzen. Auf der Basis sollte dann ein

⁵⁰ Wie viele international tätige Unternehmen plant auch e.GO den Einsatz sogenannter „Chinese Walls“, die eine Zusammenarbeit mit (ggf. Tochter-) Unternehmen in China ermöglichen, den Zugriff auf nicht direkt notwendige Daten jedoch einschränken.

abgestimmtes **zentrales Datenmodell** entwickelt werden, das zum Ziel hat, Datenredundanz zu minimieren, Datenquellen zu harmonisieren und den ubiquitären Zugriff über die Plattform zu ermöglichen. Auch Maßnahmen zur **Steigerung der Informationsqualität** sollten weiterverfolgt werden, wie beispielsweise die Bereitstellung von Apps zur Prüfung von Daten. Damit wird Führungskräften eine einfache Möglichkeit an die Hand gegeben, ihrer Verantwortung für die Informationsqualität gerecht zu werden.

9.2.4 Die Informationssystem-Architektur von e.GO

Zum Verständnis des **Ist-Zustands bei e.GO** wird im Folgenden die Informationssystem-Architektur von e.GO vorgestellt. Die Sammlung der notwendigen Daten dazu hat sich als größere Herausforderung dargestellt als erwartet. Trotz des jungen Unternehmensalters nutzt e.GO inzwischen eine **Vielzahl von Informationssystemen**, über die nur in der Buchhaltung ein vollständiger Überblick besteht. Die IT-Abteilung besitzt nur ausschnittshafte Darstellungen der Architektur. Dies ist darin begründet, dass der Fokus der IT-Abteilung bisher im Systemaufbau bestand und deshalb umfangreiche Dokumentationen nicht notwendig waren. Aus Gesprächen mit den verschiedenen Verantwortlichen (s. Anhang A.3) sowie eigenen Erfahrungen des Autors konnten letztendlich die im Folgenden dargestellten Informationen gewonnen werden.⁵¹

Die nachfolgend dargestellte **Informationssystem-Architektur stellt eine Momentaufnahme dar und wird sich weiter verändern**. So arbeitet e.GO beispielsweise zum Zeitpunkt der Fallstudie an einer verbesserten Datendurchgängigkeit, welche Änderungen der Informationssystem-Architektur zur Folge haben wird. Ziel ist eine verstärkte Nutzung der software-definierten Plattform *ThingWorx* und deren Integration mit den vorhandenen Informationssystemen – dies ist jedoch bisher noch nicht umgesetzt (s. Kapitel 9.2.3).

Abbildung 9-11 zeigt die komplette Informationssystem-Architektur von e.GO mit Ausnahme der nur für das e.GO Kart genutzten Systeme *Siemens NX*, *Siemens Teamcenter* und *SAP ERP*. Diese bilden eine geschlossene, forschungsähnliche Umgebung, die für die weitere Betrachtung nicht relevant ist.

Die zahlenmäßig meisten Informationssysteme finden sich im **Bereich der Entwicklung**. Außer dem eCAD-System *Captial* und den Simulationssystemen stammen alle Informationssysteme in diesem Bereich vom Softwarehersteller *PTC*. Auch die Systeme, welche die **Herstellungsprozesse** unterstützen, sind hauptsächlich von einem Hersteller: *PSI Automotive & Industry*. Informationssysteme für die **Produktnutzung** und den **Service** sind aktuell geplant, aber noch nicht implementiert. **Die Architektur folgt dem Motto: „Standard, wo möglich“**. Die eingeführten Informationssysteme wurden zwar konfiguriert, aber nicht weiter angepasst, sodass fast ausschließlich Standardsysteme im Einsatz sind. Mit Ausnahme der oben genannten Systeme für das

⁵¹ Die komplette Liste der Informationssysteme von e.GO ist dem Anhang beigelegt (s. Anhang A.8).

e.GO Kart gibt es keine mehrfachen Systemwelten. Zentrale Informationssysteme sind das PLM- und das ERP-System.

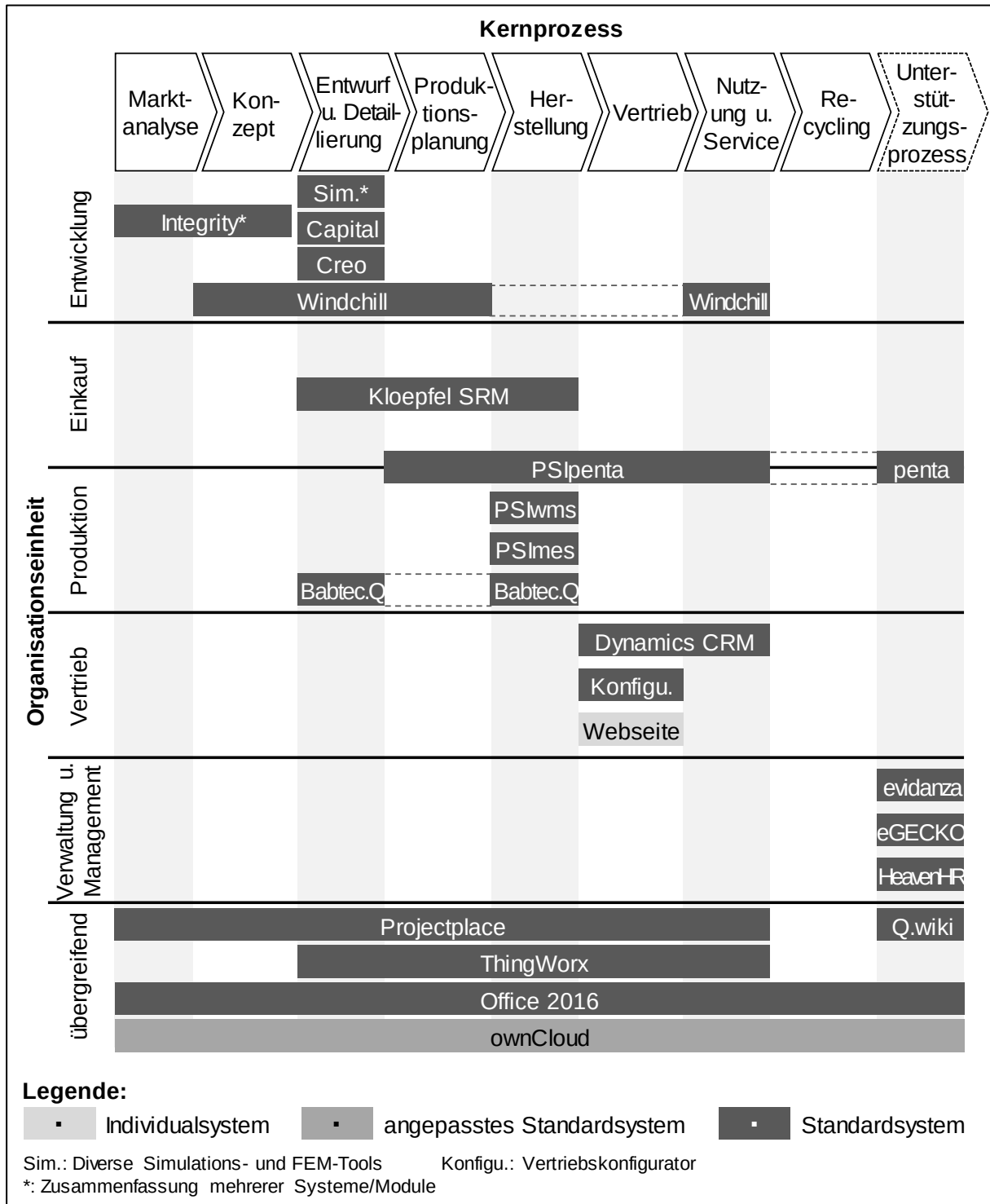


Abbildung 9-11: Ausschnitt der Informationssystem-Architektur bei e.GO mit Einordnung der Informationssysteme gemäß Kernprozessunterstützung und Organisations-einheit, zusätzliche Hervorhebung des Systemursprungs (eigene Darstellung zum Stand März 2018)

Die **Verantwortung für die Informationssysteme** liegt größtenteils bei der **IT-Abteilung**, die, wie oben beschrieben, fachbereichsnah aufgestellt ist. Im Fall der produktionsnahen Informationssysteme wird die Verantwortung zwischen der IT-

Abteilung und der Produktions-IT, also den spezifischen IT-Fachkräften in der Produktionsabteilung, geteilt. Informationssysteme mit wenigen Nutzern, wie die Simulationswerkzeuge, werden von den Fachbereichen selbst verantwortet. In letzterem Fall werden die System auch rein intern betrieben. Ansonsten sind alle Server des Unternehmens an einen externen Infrastrukturdienstleister outgesourct. Bei einigen Informationssystemen fiel die Entscheidung zugunsten eines Cloud-Anbieters, sodass hier ein Anwendungsoutsourcing betrieben wird. Dies betrifft entweder Systeme, die nicht anders zu beziehen sind, wie das *Q.wiki*, oder Systeme, die eine einfache mobile Verfügbarkeit benötigen, wie das *Dynamics CRM*. **Zwei Informationssysteme werden komplett extern betrieben** und der zugehörige Businessprozess teilweise outgesourct, weil damit Skalenvorteile erzielt werden konnten. Abbildung 9-12 visualisiert diese Zusammenhänge.

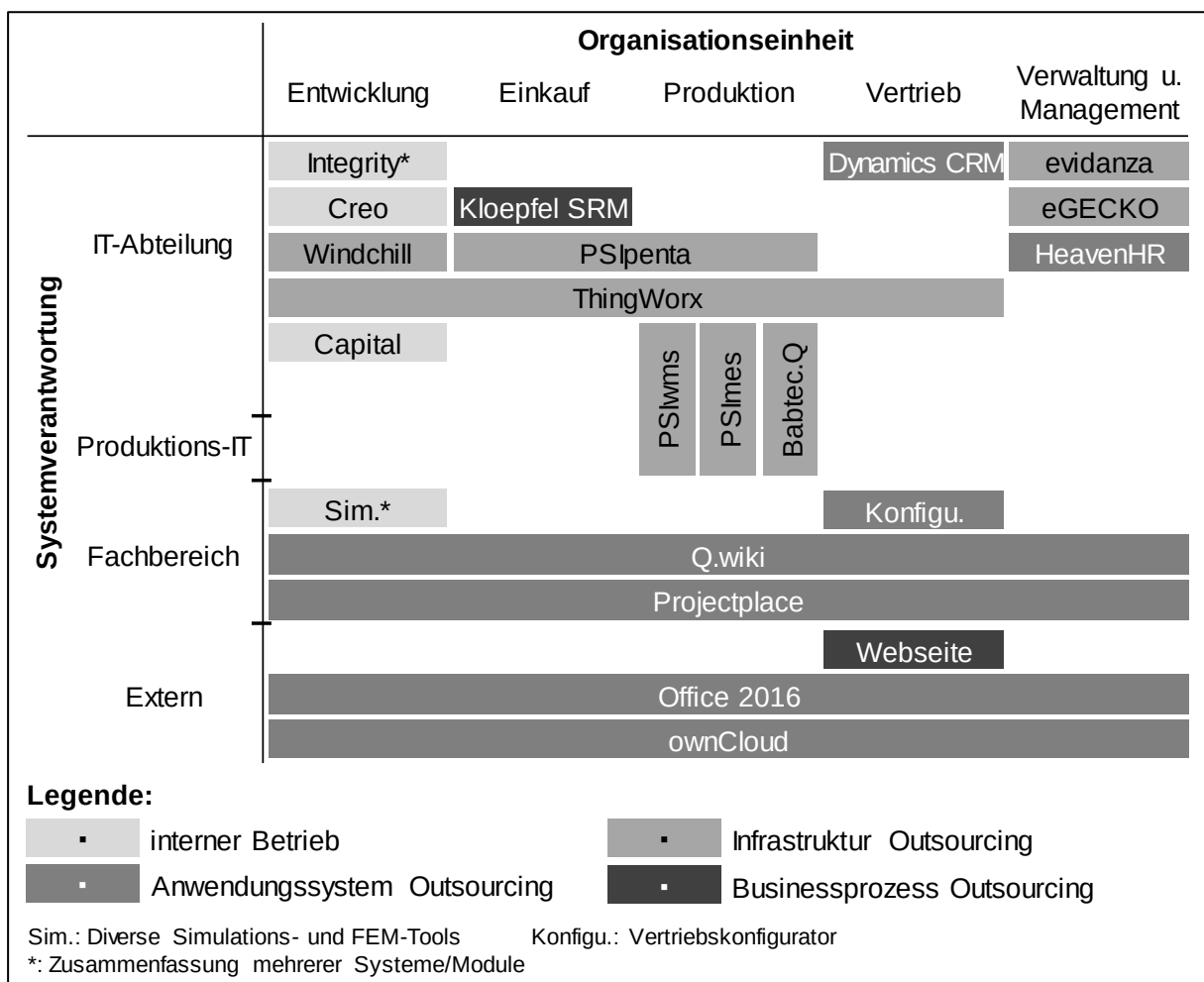


Abbildung 9-12: Ausschnitt der Informationssystem-Architektur bei e.GO mit Fokus auf die organisatorische Einordnung (eigene Darstellung zum Stand März 2018)

Die folgende Abbildung 9-13 gibt einen groben Überblick über die **Schnittstellenarchitektur** bei e.GO. Sie zeigt, dass aktuell das **ERP-System PSlpenta** die **zentrale Datendrehscheibe** ist. Im Bereich der Entwicklung stellt zusätzlich das PLM-System *Windchill* ein zentrales Informationssystem dar. Die software-definierte Plattform

ThingWorx verfügt bereits über Schnittstellen zu den beiden vorgenannten Informationssystemen, sodass die Umsetzung erster systemübergreifender Anwendungsfälle technisch möglich wäre.

e.GO nutzt vor allem Informationssysteme von den zwei Softwareherstellern PTC und PSI Automotive&Industry beziehungsweise aus deren Partnernetzwerk. Zwischen den Systemen eines Herstellers und seines Partnernetzwerks bestehen teilweise vordefinierte Schnittstellen (s. Kapitel 5.3.1), die eine umfangreiche Integration einfach ermöglichen. Ansonsten werden generische Schnittstellen zur Anbindung der Informationssysteme genutzt.

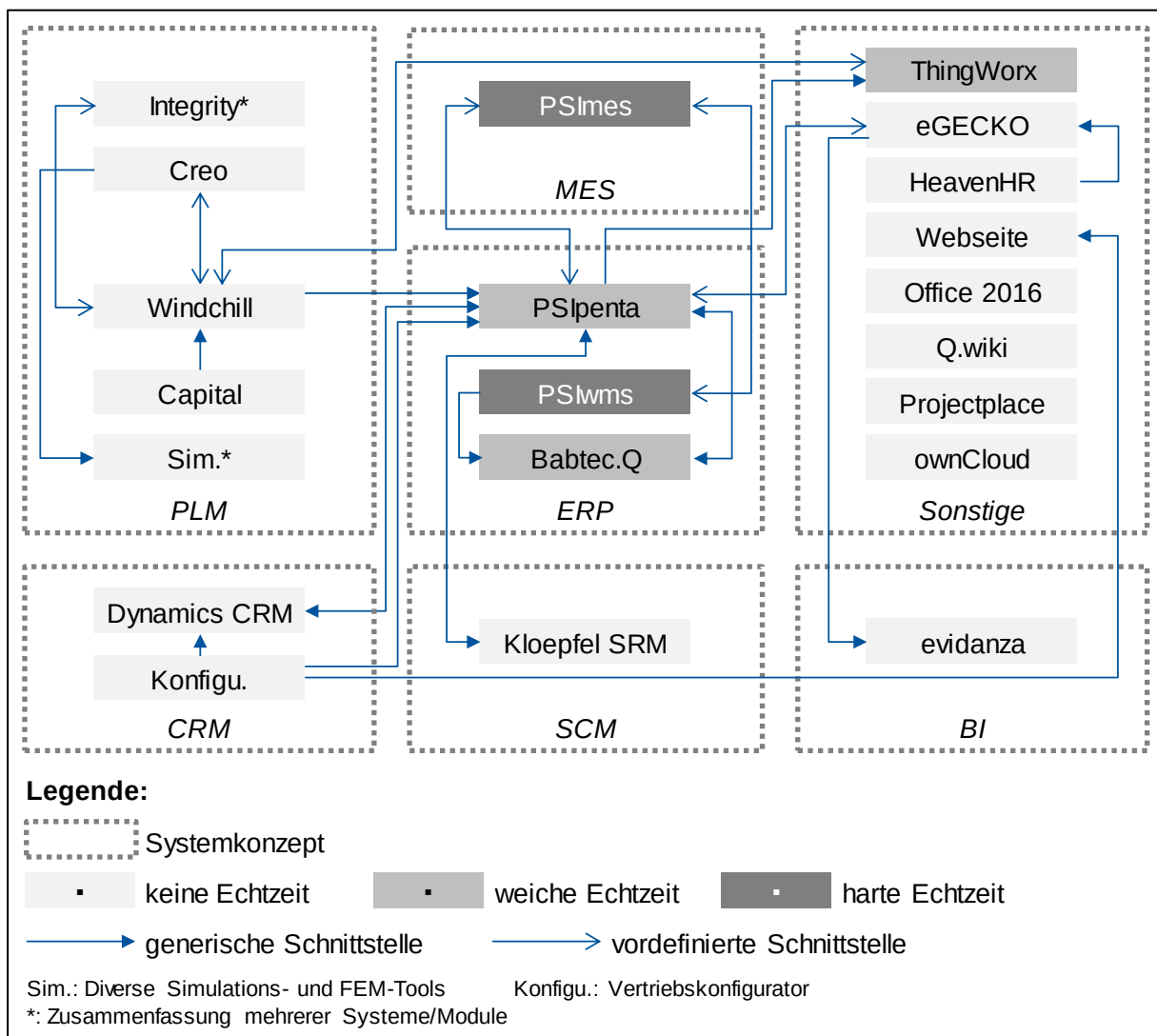


Abbildung 9-13: Ausschnitt der Informationssystem-Architektur bei e.GO mit Fokus auf Schnittstellen und Echtzeitfähigkeit (eigene Darstellung zum Stand März 2018)

Mit den obigen Darstellungen der Informationssystem-Architektur erhält e.GO erstmals eine **komplette, einheitliche Übersicht über seine Informationssystem-Architektur**. Bisher bestanden nur ausschnittsweise und lückenhafte Darstellungen, die in der nachfolgenden Abbildung 9-14 zusammengefasst sind. Nach Aussage des IT-Leiters hilft diese Übersicht bei der Weiterentwicklung der Architektur entsprechend den im Zuge der Fallstudie erarbeiteten Empfehlungen (s. Kapitel 9.2.5).

Auf Basis des aufgenommenen Ist-Zustands plant e.GO weitere Entwicklungen der Informationssystem-Architektur, vor allem in Richtung eines stabilen, echtzeitfähigen Betriebs. Dazu werden Konzepte des Cloud-Computings evaluiert wie auch Technologien untersucht, welche den Echtzeitbetrieb in den Montagebereichen ermöglichen.

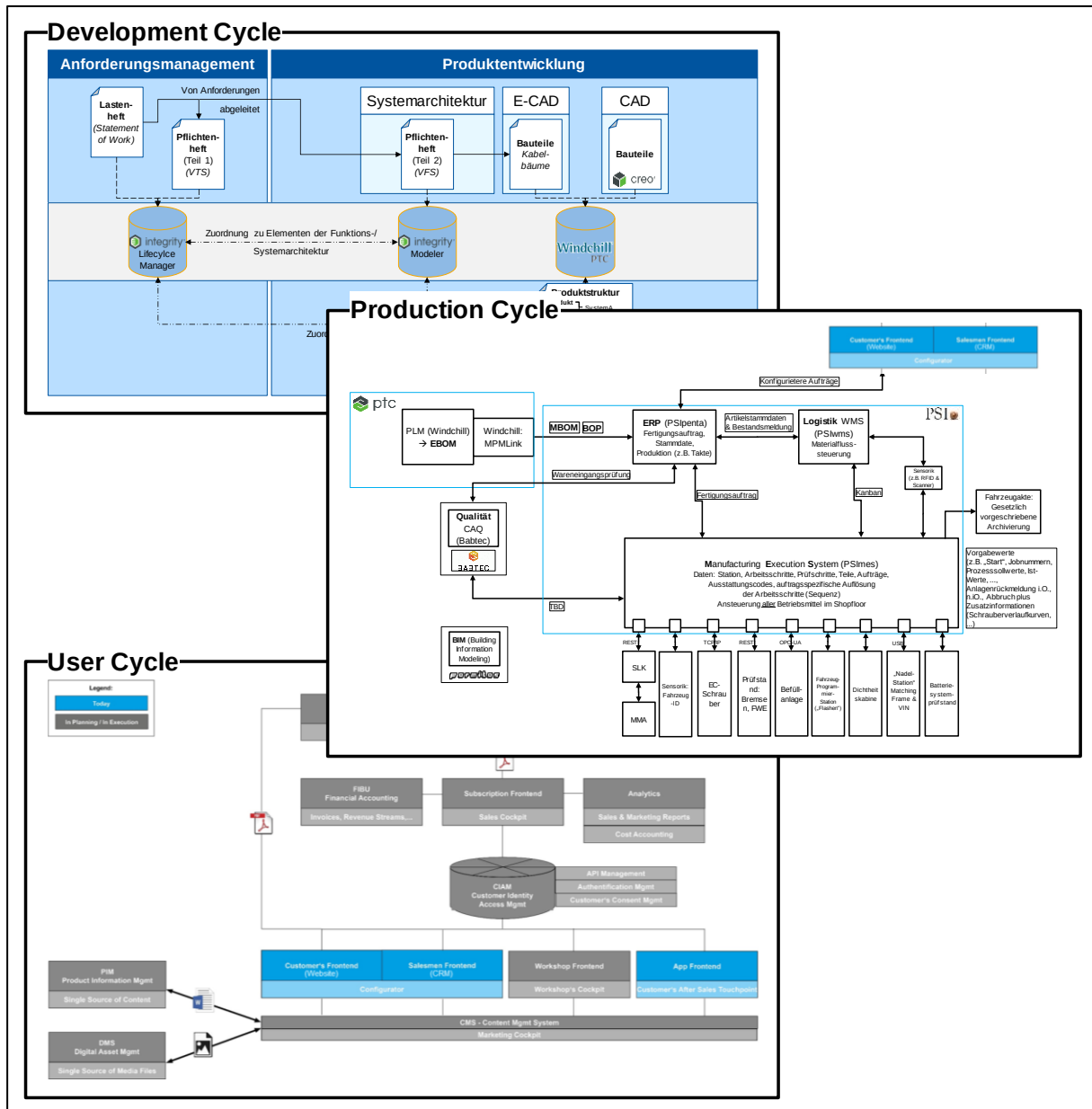


Abbildung 9-14: Von e.GO genutzte Darstellungen der Informationssystem-Architektur (eigene Darstellung unter Nutzung von Präsentationen von e.GO zum Stand März 2018)

Aus der Untersuchung des Ist-Zustands der Informationssystem-Architektur lässt sich der **Architekturtyp** bestimmen. Die **Informationssystem-Architektur von e.GO lässt sich dem Architekturtyp III zuordnen** (s. Kapitel 6.8.3). Dies liegt einerseits begründet in der Dezentralität der Architektur: Neben dem ERP-System existieren zwei weitere zentrale Informationssysteme (PLM und Plattform), welche wichtige Konzentrationspunkte der Architektur darstellen. Andererseits kann die IT-Organisation noch nicht als Erfolgspartner, sondern als Geschäftspartner, eingeordnet werden. Begründen lässt sich das damit, dass die IT-Abteilung zum Zeitpunkt der Abfassung der

vorliegenden Arbeit zwar sehr eng mit den Fachbereichen zusammenarbeitet und die Informationssystem-Architektur weiterentwickelt, aber noch nicht zu neuen Geschäftsmodellen oder datenbasierten Dienstleistungen beiträgt. Eine Entwicklung in diese Richtung ist jedoch geplant.

9.2.5 Empfehlungen zur Entwicklung der Informationssystem-Architektur bei e.GO

e.GO ist auf dem Weg, ein **datengetriebenes Unternehmen** zu werden. Auf diesem Weg muss zuerst Basisarbeit geleistet werden – in Bezug auf das Informationsmanagement heißt das: Es müssen Informationssysteme aufgebaut und integriert werden. Diese Basisarbeit ist inzwischen nahezu abgeschlossen, wie die oben gezeigte Informationssystem-Architektur zeigt. Ein zentrales Element für die durchgängige Datennutzung ist der Einsatz einer software-definierten Plattform. Um diese einzusetzen, hat e.GO bereits einige Anwendungsfälle geplant und sich für einen Softwarehersteller entschieden.

Entsprechend dem in der vorliegenden Arbeit vorgeschlagenen Vorgehen muss e.GO nun die **Integration dieser Plattform vorantreiben**, um eine zukunftsfähige, komplexitätsarme Informationssystem-Architektur zu erhalten. Im Speziellen muss e.GO dazu die Transparenz über die Informationssysteme in allen Integrationsdimensionen erhöhen. Jene muss auch dann erhalten bleiben, wenn weitere Informationssysteme aufgebaut werden, um alle notwendigen Anforderungen der Fachbereiche zu erfüllen.

Abgeleitet aus dem in Kapitel 7.1 vorgestellten Ziel-Zustand und in Verbindung mit dem Architekturtyp der Informationssystem-Architektur von e.GO sowie dem Typ der eingesetzten software-definierten Plattform, lassen sich die folgenden Umsetzungsempfehlungen ableiten:

Aus **technischer Sicht** muss eine Architektur entwickelt werden, welche den Anforderungen nach **Skalierbarkeit, Verfügbarkeit und Ubiquität** entspricht. Dazu bedarf es nicht nur eines Betriebskonzepts, sondern auch entsprechende Serverarchitekturen. Weiterhin muss die Umsetzung der Architektur in der Cloud geprüft und vorangetrieben werden, um das angestrebte Franchise-Konzept sowie weiteres dynamisches Mitarbeiterwachstum technisch zu ermöglichen. Hierzu muss e.GO klare Vorgaben für mögliche Dienstleister entwickeln.

Die Entwicklung der internen Informationssystem-Architektur und die Umsetzung von datenbasierten Dienstleistungen zu Kunden müssen integral verfolgt werden. Deshalb ist, bezogen auf die organisatorischen Integrationsanforderungen, eine **Weiterentwicklung der IT-Organisation** in Richtung des **Erfolgspartners** (s. Kapitel 2.2.4) notwendig. Weiterhin muss die Datenverantwortung organisiert werden, um eine durchgängig hohe Informationsqualität sicherzustellen. Dazu müssen für alle Daten des zentralen Datenmodells Verantwortliche benannt werden.

In Bezug auf prozessuale Integrationsanforderungen hat e.GO schon eine Reihe von Anforderungen erfüllt, indem Anwendungsfälle für die software-definierte Plattform de-

finiert wurden (s. Kapitel 9.2.3). Weitere Entwicklungen sollten in Richtung der **Prozesstransparenz** erfolgen. Wenn alle relevanten Informationssysteme in die software-definierte Plattform integriert sind, können zudem **konfigurierbare Prozesse** ermöglicht werden, die der Agilität von e.GO Rechnung tragen.

Die Qualität der von Unternehmen angebotenen Informationen, also die **Informationsqualität**, wird künftig ein entscheidender Erfolgsfaktor sein. Daher sollte e.GO seine Anstrengungen in der Integrationsdimension *Daten* erhöhen, um die Qualität stetig zu steigern. Dazu gehört nicht nur ein **redundanzfreies und in den Informationssystemen implementiertes Kerndatenmodell**, sondern auch Mitarbeiterschulungen und die Bereitstellung von Apps, welche es den **Verantwortlichen** vereinfachen, die Informationsqualität zu überprüfen.

9.3 Fallstudie 2: PTC

9.3.1 Das Unternehmen PTC

PTC ist ein in den USA ansässiger Softwarehersteller, der 1985 gegründet wurde. Das Unternehmen ist weltweit tätig und beschreibt sich selbst als **Marktführer im Bereich industrieller Softwareplattformen**. Der Fokus der Betätigung liegt auf dem amerikanischen Kontinent, allerdings wird signifikanter Umsatz auch in Europa und Asien generiert. Dazu kann PTC auf ein weltweites Partnernetzwerk aus über 1.100 Partnern zurückgreifen. Im gesamten PTC-Ökosystem arbeiten über 300.000 Entwickler an der Implementierung von PTC-Software in Unternehmen.⁵²

Das Portfolio an Softwareprodukten hat PTC in den vergangenen Jahren stetig erweitert. Begonnen hat PTC mit der CAD-Software *Pro/ENGINEER*, das heute *Creo* heißt, im Jahr 1988. Dem folgte das PLM-System *Windchill* im Jahr 1998. Seit 2014 gehört die software-definierte Plattform *ThingWorx* zum PTC-Produktspektrum, ergänzt um AR-Software und OT-Konnektivitätslösungen (s. BRUNELLI 2014). Heute setzt sich das **Produktportfolio zusammen aus CAD-, PLM- und SLM-Systemen, sowie Lösungen für das IoT und intelligente Fertigungen**. Dabei sind die Systeme untereinander sehr gut integriert, beispielsweise kann ein in *Creo* entwickeltes Modell in *Windchill*, *ThingWorx* und der Augmented-Reality-Lösung *Vuforia* visualisiert werden. Das verbindende Element zum komplexen Datenaustausch zwischen den Systemen bildet *ThingWorx* (s. auch Kapitel 9.3.3 u. 9.3.4).

PTC sieht sich selbst als **Meinungsführer im Bereich der digital-physischen Konvergenz**. Dies zeigt sich unter anderem an den von PTC veröffentlichten, vielzitierten Harvard-Business-Review-Artikeln von PORTER u. HEPPELMANN (vgl. PORTER u. HEPPELMANN 2014; 2015; 2017) sowie durch die PTC-Beteiligung am *Industrie 4.0 Maturity Index* (vgl. SCHUH ET AL. 2017a). Auch externe Institutionen belegen die Vorreiterrolle, so wird PTC beispielsweise von den IT-Marktforschungsinstituten *Forrester*

⁵² Weitere Informationen unter <https://www.ptc.com/de/about>.

und IDC als Marktführer im Bereich von IoT-Plattformen angesehen (s. PELINO U. HEWITT 2016, S. 9; CROOK U. MACGILLIVRAY 2017, S. 1).

9.3.2 Der Softwareauswahlprozess software-definierter Plattformen aus Sicht eines Softwareherstellers

Aus der Sicht eines Softwareherstellers können Softwareauswahlprozesse verschiedener Unternehmen verglichen werden. Durch Gespräche mit verschiedenen PTC-Experten (s. Anhang A.3) konnte in der vorliegenden Fallstudie **die typische Abfolge der üblichen Schritte zusammengefasst** werden. Abbildung 9-15 gibt einen Überblick über diese Schritte, die im Folgenden näher erläutert werden. Dies dient dazu, die in Kapitel 8.2 der vorliegenden Arbeit vorgestellten Hinweise zum Auswahl- und Einführungsprozess von software-definierten Plattformen zu validieren. Der Prozess wird bei PTC von verschiedenen Einheiten betreut.⁵³ Die Einheit *Presales* deckt den größten Teil ab: Sie nimmt initial den Kontakt zum Kunden auf und begleitet diesen bis zur Finalisierung von Prototypen. Kommerzielle Aspekte werden von der ‚PTC Sales‘-Einheit übernommen. Die Umsetzung des Echt-systems sowie die darauffolgenden Wartungs- und Pflegeaktivitäten übernimmt die Einheit *PTC Cooperate*.

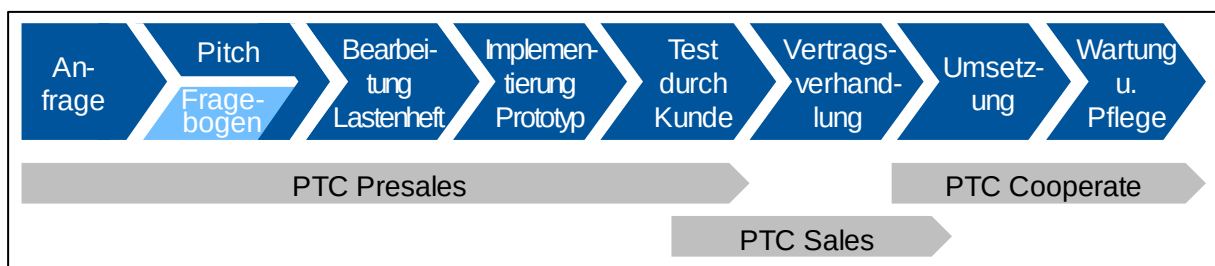


Abbildung 9-15: Übersicht des Softwareauswahlprozesses aus Sicht von PTC (eigene Darstellung)

Der Softwareauswahlprozess für software-definierte Plattformen beginnt, wie jeder andere Softwareauswahlprozess, mit einer **Anfrage des Kunden** (s. ALPAR ET AL. 2016, S. 410). Diese kann konkret oder unspezifisch sein; meist befindet sich der Kunde in einer Phase, in der er verschiedene Softwareanbieter anspricht, um deren Lösungen kennenzulernen. Einige Kunden versenden in diesem Schritt mehr oder weniger umfangreiche **Fragebögen**, um Informationen über die Softwarelösung zu sammeln und vergleichen zu können. Danach – oder gegebenenfalls auch direkt – folgt die Einladung, die Software **in einem Pitch kurz vorzustellen**. An dieser Stelle nutzt PTC Standardbeispiele, welche die Funktionalität der Software gut darstellen können.

In der nächsten Phase beginnt der unternehmensspezifische Teil des Softwareauswahlprozesses. Kunden stellen in dieser Phase oft **Lastenhefte** bereit, in der sie – in unterschiedlichen Detaillierungsgraden – ihre Ist-Situation, die konkreten Herausforderungen und mögliche Lösungsansätze beschreiben. Die Ist-Situation umfasst dabei

⁵³ Es wird hier der Prozess für PTC Deutschland besprochen. Dieser unterscheidet sich unter Umständen in anderen PTC-Einheiten vom hier vorgestellten.

eine Unternehmensbeschreibung, Prozessbeschreibungen sowie Ausschnitte aus der vorhandenen Informationssystem-Architektur. Die Herausforderung beinhaltet zumindest einen Anwendungsfall. Dieser befasst sich im Kontext software-definierter Plattformen mit einer fehlenden systemübergreifenden Transparenz oder Prozesssteuerung in einem bestimmten Teil des Unternehmens. Dies kann, wie in den unten gezeigten Kundenbeispielen (s. Kapitel 9.3.3 u. 9.3.4), der Change-Prozess oder eine FMEA-Optimierung sein, oder auch der Produktionsprozess und die Maschinenwartungsoptimierung. In den Lastenheften definieren Kunden eigene Lösungsansätze. Diese können neben angepassten Prozessen ausgearbeitete Schnittstellenkonzepte und graphische Oberflächen enthalten. Wenn die Lösungsideen weniger ausgeprägt, oder mit PTC Software in der gewünschten Form nicht umsetzbar sind, hat PTC entsprechend höhere Aufwände bei der Ermittlung der Anforderungen vom Kunden.

Bei der folgenden **Implementierung eines Prototyps** ist die Mitarbeit des Kunden von zentraler Bedeutung – einerseits, damit der Kunde die Software näher kennenlernt, andererseits, um als Informationslieferant für die Implementierung durch PTC zu dienen. Für die Implementierung werden tiefgehende Informationen über die Datenstrukturen in den vorhandenen Informationssystemen, über vorhandene Schnittstellen und vorhandene Prozesse benötigt – diese müssen von Mitarbeitern des Unternehmens bereitgestellt werden. Die Prototypen werden von PTC in Testsystemumgebungen umgesetzt, um die Livesysteme nicht zu beeinträchtigen. PTC setzt in dieser Phase nicht den kompletten Anforderungsumfang des Kunden um, sondern beschränkt sich beispielsweise auf die Konnektierung von zwei zentralen Informationssystemen. Der Umfang der Implementierung hängt dabei von der Größe des Kunden und vom Status von dessen Entscheidungsfindung ab.

Nach Umsetzung des Prototyps folgt eine Phase, in welcher der **Kunde umfangreiche Tests mit der Lösung** vollziehen kann, bevor er eine Entscheidung trifft. Danach werden die **Vertragsverhandlungen** durchgeführt und, bei erfolgreichem Vertragsabschluss, die **Umsetzung der Echt-Lösung** begonnen. Auch hier hängt es vom jeweiligen Kunden ab, ob die software-definierte Plattform zuerst für einen Anwendungsfall eingeführt wird oder ob sie schrittweise aufgebaut, getestet und dann komplett eingeführt wird. Allen Ansätzen gemein ist, dass PTC ein schrittweises Entwicklungsvorgehen durchführt, bei dem ein Anwendungsfall nach dem anderen vom Kunde abgenommen wird.

Die beschriebene Vorgehensweise, die PTC bei der Auswahl von software-definierten Plattformen durch seine Kunden erfährt, entspricht dem Vorgehensvorschlag aus Kapitel 8.2. Durch Umsetzung einzelner Anwendungsfälle wird dem MVP-Konzept gefolgt. PTC bietet darüber hinaus sogenannte „*Industrial Apps*“ an,⁵⁴ welche Standardanwendungsfälle bei produzierenden Unternehmen abdecken. Somit folgt das Vorgehen teilweise auch dem Standard-Prinzip. Es muss jedoch bemerkt werden, dass viele

⁵⁴ Weitere Informationen unter <https://www.ptc.com/de/thingworx-applications/manufacturing>.

Lösungen aufgrund von Kundenwünschen einen großen Anteil an Individualentwicklungen enthalten, weil gerade bei großen Unternehmen eine Anpassung an eine existierenden Informationssystem- und Prozessarchitektur vorgenommen werden muss.

9.3.3 Anwendungsfall Kunde A

Bei PTC-Kunde A handelt es sich um ein Unternehmen aus dem Automotive-Umfeld. Da die Beziehung zwischen den Kunden A und B zu PTC durch eine Vertraulichkeitsvereinbarung geschützt ist, werden beide Anwendungsfälle entsprechend anonymisiert dargestellt. Weiterhin ist bei den Kunden nur ein Ausschnitt des Unternehmens bekannt – nämlich der Ausschnitt, auf den sich die Anforderungsbeschreibung des Kunden bezieht. Insofern kann auch nur der Ausschnitt der Anforderungen und der Informationssystem-Architektur bewertet werden.

PTC hat folgende **Ausgangssituation** beim Kunden vorgefunden: Das Unternehmen hat eine gewachsene Informationssystem-Architektur mit einer Vielzahl an Informationssystemen, die jedoch nur teilweise miteinander vernetzt sind. Im Speziellen führt die Nutzung des Informationssystems zur Fehlermöglichkeits- und -einflussanalyse (FMEA) zu einem hohen manuellen Aufwand durch die Nutzer. Dies liegt daran, dass keine Verbindung zu den Fehlerkatalogen im vorhandenen PDM-System und auch keine Rückkopplung mit Testfällen aus dem Produkteinsatz vorgenommen werden konnte. Weiterhin waren in dem betrachteten Umfeld bisher keine PTC-Tools im Einsatz, sondern ein PLM-System und ein Anforderungsmanagementtool von Drittanbietern. Allerdings plant Kunde A, neben der software-definierten Plattform *ThingWorx* auch das PLM-System von PTC, *Windchill*, einzuführen.

Der **Anwendungsfall**, der unter Nutzung von ThingWorx umgesetzt werden sollte, war also die Verbindung von Anforderungen und Bauteilen aus dem Anforderungsmanagementtool und dem PDM-System sowie von Systemelementen aus dem neu einzuführenden *Integrity Modeler* zur Umsetzung von MBSE-Zyklen. Die so hergestellte Relation soll dann dem vorhandenen FMEA-Tool per Schnittstelle übergeben werden. Damit soll die Qualitätsabsicherung optimiert werden, indem zusätzliche Daten aus der Realität die digitalen Abbilder der Teile und die Fehlerabschätzungen genauer machen.

Die **Auswahl einer software-definierten Plattform für einen solchen Anwendungsfall** ist der richtige Weg, da verschiedene vorhandene Informationssysteme verbunden und die darin enthaltenen Daten ausgewertet und aggregiert werden müssen. In Tabelle 9-4 werden die notwendigen Ausprägungen der Merkmale software-definierter Plattformen zur Umsetzung des genannten Anwendungsfalls zusammengefasst. Die Ausprägungen entsprechen der einer Typ-II-Plattform (s. Kapitel 5.5.2). Sind weitere Anwendungsfälle bei Kunde A geplant, kann der Einsatz von *ThingWorx*, das über mehr als die geforderte Funktionalität verfügt, jedoch ebenfalls sinnvoll sein.

Tabelle 9-4: Notwendige Ausprägungen der software-definierten Plattform für „PTC-Kunde A“ (eigene Darstellung)

Merkmal	Ausprägungen			
<i>Schnittstellen</i>	generische Schnittstellen	vordefinierte Schnittstellen		
<i>Datenmanagement</i>	feste Datenstruktur	Datenharmonisierung und -bereinigung	Datenanreicherung	frei konfigurierbare Datenobjekte
<i>Datenbankkonzept</i>	relationale Datenbank	NoSQL-Datenbank	sonstige Datenbankkonzepte	
<i>Stufen der Datenverarbeitung</i>	Erzeugen von Sichtbarkeit	Transparenz und Diagnose	Prognose	Adaption und Entscheidungsautomatisierung
<i>IT-Sicherheitsfunktionen</i>	nur interne Sicherheitsfunktionen	Sicherheit für Plattform und angebundene Systeme	Vollintegration in IT-Sicherheitsinfrastruktur des Unternehmens	
<i>Informationsbereitstellung</i>	keine eigene Auswertungsfunktion	Bereitstellung von Auswertungen und Visualisierungen	Generierung eigener Apps/Widgets	
<i>Entwicklungsumgebung (E.)</i>	keine eigene E.	eigene, proprietäre E.	offene E.	

Legende:

	teilweise erfüllt		vollständig erfüllt

Im **Ist-Zustand der Informationssystem-Architektur** ist neben dem Drittanbieter-PDM-System und dem anzubindenden FMEA-Tool auch das Anforderungsmanagementtool *Doors* von IBM zu betrachten. Im Ziel-Zustand soll noch der *Integrity Lifecycle Manager*, das PTC-Tool zum Anforderungsmanagement, als Übersetzungsfunktion zu *Doors* eingeführt werden. Selbiges gilt auch für *PTC Windchill*, das als Übersetzer für das vorhandene PDM-System eingeführt werden soll. Der *PTC Integrity Modeler* wird, wie auch *PTC ThingWorx*, neu eingeführt. Als eine zusätzliche Herausforderung hat sich eine politische Dimension bei der Systemeinführung aufgetan: Die ebenfalls ge-

plante Ablösung des aktuell genutzten PDM-Systems sollte im Unternehmen vor einigen Mitarbeitern noch geheim gehalten werden, weshalb die Implementierung von Schnittstellen zwischen diesem System und dem neu eingeführten *Windchill* nur über Umwege möglich war. Abbildung 9-16 zeigt den beschriebenen Ausschnitt der Informationssystem-Architektur inklusive einer Darstellung der Schnittstellenarchitektur.

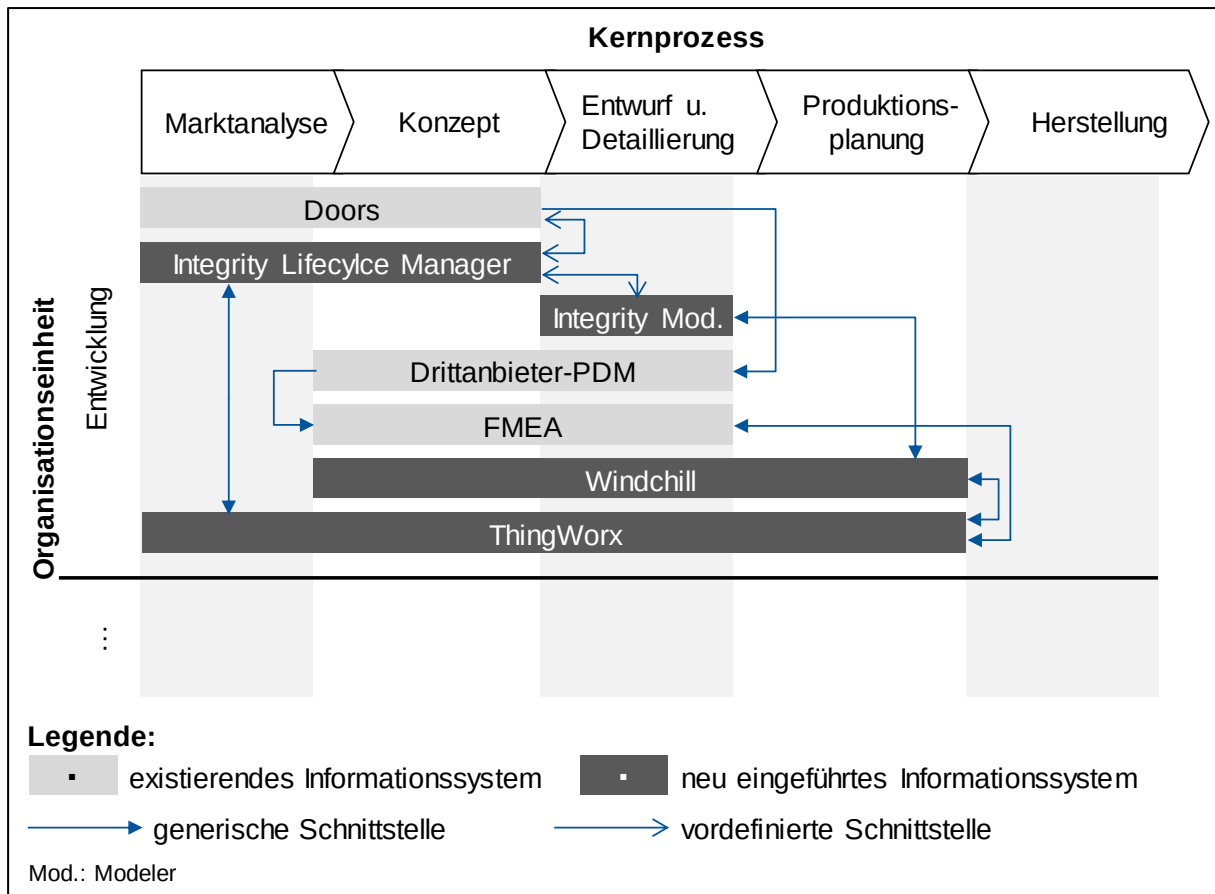


Abbildung 9-16: Ausschnitt aus der Informationssystem-Architektur von „PTC-Kunde A“ (eigene Darstellung)

Aufgrund der fehlenden Informationen kann der **Architekturtyp** von Kunde A nicht genau bestimmt werden. Die Dezentralität und die noch nicht vollständige Unterstützung der Geschäftsprozesse lässt jedoch die Vermutung zu, dass es sich um eine Typ-III-Architektur (s. Kapitel 6.8.3) handelt.

9.3.4 Anwendungsfall Kunde B

Bei PTC-Kunde B handelt es sich ebenfalls um ein **Unternehmen aus dem Automotive-Umfeld**. Auch bei diesem Unternehmen ist nur ein Ausschnitt der Informationssystem-Architektur bekannt, da das Unternehmen keine weiteren Informationen zur Verfügung gestellt hat.

Die **Ausgangssituation** bei Kunde B stellt sich wie folgt dar: Ebenso wie bei Kunde A hat auch Kunde B eine gewachsene Informationssystem-Architektur. Besonders hervorzuheben ist, dass viele Informationssysteme mehrfach im Unternehmen vorhanden sind: So gibt es beispielsweise über zehn Installationen von *PTC Integrity*. Im Gegensatz zu Kunde A hat Kunde B schon seit längerer Zeit PTC-Software im Einsatz. Das

erleichtert die Integration von *ThingWorx*. Die spezifische Herausforderung, wegen der PTC um die Präsentation seiner software-definierten Plattform gefragt wurde, ist die Asynchronität der Change-Prozesse in den vorhandenen Informationssystemen. Ein Produkt-Change kann sowohl im ERP- wie auch im PLM- oder Projektplanungstool initiiert werden. Eine Verknüpfung der einzelnen Change-Workflows fand bis dato jedoch nicht automatisiert statt. Genauso wurden bisher auch keine systemübergreifenden Baselines gezogen.⁵⁵

Der **Anwendungsfall** für *ThingWorx* ist daher die Verknüpfung der Change-Workflows, die Erzeugung von systemübergreifenden Baselines sowie die Bereitstellung von einfachen Apps zur Initiierung, Bearbeitung und Freigabe von Changes. Mögliche Changes aus den vorhandenen Informationssystemen können sein:

- die Initiierung eines Engineering-Changes aus dem PLM-System *Windchill*,
- die Veränderung einer Kundenanforderung, die dann im Anforderungsmanagementtool *Integrity* zu Veränderungen führt,
- eine Materialveränderung durch einen Zuliefererwechsel oder eine Anpassung des Fertigungsprozesses, der zu einem Change-Prozess im ERP-System *SAP* führt, oder
- eine Änderung des Projektplans aus dem Projektplanungstool *Planisware*.

Kunde B hat das Ziel, dass die genannten **Changes direkt aus einer einfachen App über *ThingWorx* gestartet** und dann mit den vorhandenen Informationssystemen synchronisiert werden können. Zugleich soll es möglich sein, auch wie bisher aus den schon vorhandenen, umfangreichen Change-Prozessen in den Informationssystemen wie *SAP* einen Change initiieren zu können. Dieser muss dann über *ThingWorx* mit den anderen Informationssystemen synchronisiert werden. Durch diesen Einsatz von *ThingWorx* sollen die vorhandenen Change-Prozesse zusätzlich vereinfacht werden, da die bisher manuelle Synchronisation wegfällt. Da den vorhandenen Prozessen und Informationssystemen auch umfangreiche Zugriffsrechte hinterliegen, soll der systemübergreifende Change-Prozess die vorhandenen Prozesse nutzen und nicht etwa ersetzen.

Um die genannten Anforderungen zu erfüllen, benötigt Kunde B eine software-definierte Plattform, die vordefinierte Schnittstellen zu den Umsystemen ermöglicht, die Datenanreicherung unterstützt und eigene, individuell konfigurierbare Apps bereitstellen kann. Tabelle 9-5 fasst diese **Ausprägungen der Merkmale von software-defi-**

⁵⁵ Bei einer Baseline handelt es sich um das Einfrieren eines Datenstands zu einem bestimmten Zeitpunkt. Im Fall von Changes bedeutet dies, dass zu Beginn des Changes eine Baseline gezogen wird, um Veränderungen durch den Change darstellen und gegebenenfalls zum alten Zustand zurückspringen zu können.

nierten Plattformen zusammen. Die notwendigen Ausprägungen übersteigen die einer Typ-II-Plattform, sodass die Auswahl einer Typ-I-Plattform wie *ThingWorx* gerechtfertigt erscheint.

Tabelle 9-5: Notwendige Ausprägungen der software-definierten Plattform für „PTC-Kunde B“ (eigene Darstellung)

Merkmal	Ausprägungen			
<i>Schnittstellen</i>	generische Schnittstellen	vordefinierte Schnittstellen		
<i>Datenmanagement</i>	feste Datenstruktur	Datenharmonisierung und -bereinigung	Datenanreicherung	frei konfigurierbare Datenobjekte
<i>Datenbankkonzept</i>	relationale Datenbank	NoSQL-Datenbank	sonstige Datenbankkonzepte	
<i>Stufen der Datenverarbeitung</i>	Erzeugen von Sichtbarkeit	Transparenz und Diagnose	Prognose	Adaption und Entscheidungsautomatisierung
<i>IT-Sicherheitsfunktionen</i>	nur interne Sicherheitsfunktionen	Sicherheit für Plattform und angebundene Systeme	Vollintegration in IT-Sicherheitsinfrastruktur des Unternehmens	
<i>Informationsbereitstellung</i>	keine eigene Auswertungsfunktion	Bereitstellung von Auswertungen und Visualisierungen	Generierung eigener Apps/Widgets	
<i>Entwicklungsumgebung (E.)</i>	keine eigene E.	eigene, proprietäre E.	offene E.	

Legende

	teilweise erfüllt		vollständig erfüllt

Die aktuelle Informationssystem-Architektur von Kunde B besteht, wie oben schon erwähnt, aus einer Vielzahl von teilweise mehrfach vorhandenen Informationssystemen. Für den vorliegenden Anwendungsfall sind die vorhandenen Informationssysteme *Integrity*, *Windchill*, *SAP* und *Planisware* relevant. Diese sind teilweise über Schnittstel-

len verbunden, die Integration ist jedoch nicht ausreichend tief. Das neu von PTC eingeführte *ThingWorx* ist mit allen diesen Informationssystemen über Schnittstellen verbunden, die eine Übergabe von Change-Prozessen ermöglichen. Abbildung 9-17 zeigt den relevanten Ausschnitt aus der Informationssystem-Architektur.

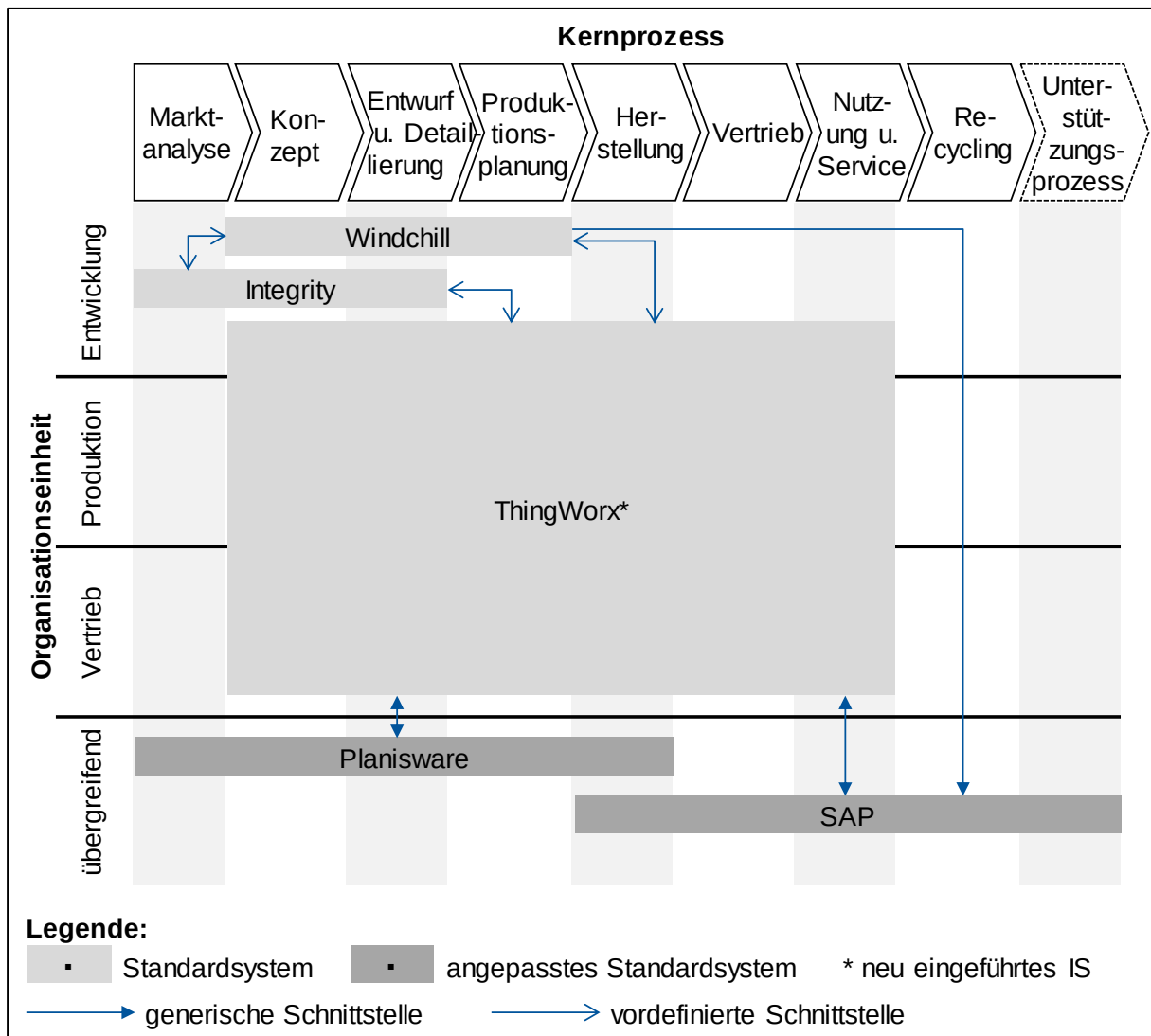


Abbildung 9-17: Ausschnitt aus der Informationssystem-Architektur von „PTC-Kunde B“ (eigene Darstellung)

Die **Herausforderungen** in der aktuellen Architektur liegen auf verschiedenen Ebenen. **Technisch** besteht, neben nicht ausreichenden Schnittstellen, die Herausforderung darin, dass das SAP-System die gewünschte Baseline-Funktion technisch nicht unterstützt. Um eine systemübergreifende Baseline zu ermöglichen, musste zuerst ein Workaround über *ThingWorx* bereitgestellt werden. Auf **organisatorischer Ebene** war die im Projektteam nicht ausreichend vorhandene Entscheidungsbefugnis ebenfalls ein Problem bei der Umsetzung. Weil das Team nur aus Mitarbeitern aus der Engineering-IT bestand, das SAP-System jedoch von der Corporate-IT betrieben wird, konnten die zur Entwicklung der *ThingWorx*-Schnittstelle notwendigen Informationen vom Kunde nicht in ausreichendem Maß zur Verfügung gestellt werden. Ebenso eine organisatorische Herausforderung war, dass Teile der Informationssysteme stark angepasste Standardlösungen sind, die keine generischen Schnittstellen besitzen. So

musste der Kunde eine Schnittstelle zwischen *Planisware* und *ThingWorx* erst selbst entwickeln, um den Zugriff darauf zu ermöglichen.

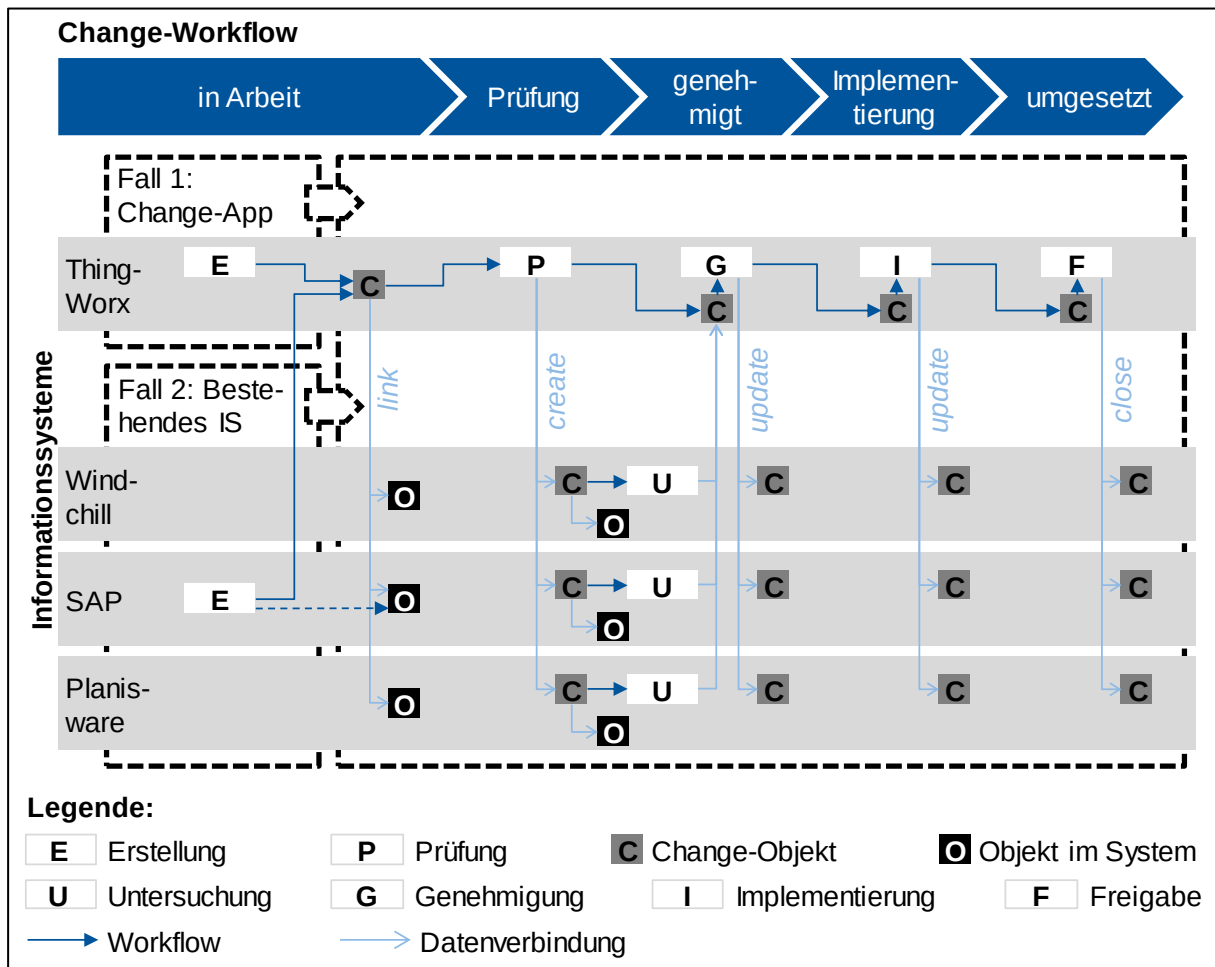


Abbildung 9-18: Aggregierte Darstellung des neu mit ThingWorx umgesetzten Change-Workflows bei "PTC-Kunde B" (eigene Darstellung)

Trotzdem konnte der gewünschte Anwendungsfall von PTC umgesetzt werden. Abbildung 9-18 zeigt den **typischen Ablauf eines Changes**, der nun sowohl aus *ThingWorx* (Fall 1) als auch aus einem bestehenden Informationssystem heraus (Fall 2) gestartet werden kann. *ThingWorx* ermöglicht die Verknüpfung der vom Change betroffenen Objekte in den einzelnen Informationssystemen und sorgt von da an für die Synchronisation der unabhängig voneinander ablaufenden Change-Prozesse in den Einzelsystemen. Für Führungskräfte, die für die Freigaben im Prozess verantwortlich sind, wurden entsprechende Apps entwickelt, die einen systemunabhängigen Zugriff auf alle change-relevanten Informationen ermöglicht.

9.3.5 Handlungsempfehlungen für PTC

Die Methoden, die PTC zur Einführung einer software-definierten Plattform nutzt, entsprechen schon zum großen Teil dem in der vorliegenden Arbeit vorgestellten Vorgehensmodell. Die von PTC bereitgestellte **software-definierte Plattform ThingWorx erfüllt alle Anforderungen an eine Typ-I-Plattform** und liefert somit den vollen Funktionsumfang. Auf technischer Ebene beherrscht PTC vor allem die Implementierung

von Schnittstellen zu verschiedenen Informationssystemen aufgrund entsprechender Erfahrung aus vorangegangenen Projekten. Im Bereich organisatorischer Integrationsanforderungen sollte PTC einen **größeren Fokus auf die Personen legen, die von der Einführung der Plattform betroffen** sind. Hierin liegt – neben der Erfüllung funktionaler Anforderungen – nach Erfahrung des Autors ein Schlüssel zum Erfolg von solchen Einführungsprojekten (s. Kapitel 7.1.2). Auch die **höhere Transparenz anhand von Datenmodellen** sollte in den Kundenprojekten vorangetrieben werden, damit den Kunden gegebenenfalls über die angeforderten Implementierungen hinaus datenbasierte Mehrwerte aufgezeigt werden können (s. Kapitel 7.1.4).

Insgesamt ist das Vorgehen über die **Implementierung einzelner Anwendungsfälle sinnvoll**, jedoch sollte auch die in der Arbeit vorgeschlagene vorherige **Transformation der Informationssystem-Architektur** in den Blick genommen werden. Dies liegt zwar oft nicht in der Hand eines Softwareherstellers, jedoch sollte PTC seine Kunden dazu motivieren, damit diese in Richtung einer komplexitätsärmeren Architektur arbeiten, statt mit einem PTC-System nur ein weiteres Informationssystem hinzu zu bekommen.

9.4 Zusammenfassende Bewertung der Evaluationsergebnisse

Durch die beiden Fallstudien mit e.GO und PTC sowie die exemplarische Anwendung der Methoden in den Kapiteln 5.5, 6.7, 6.8 und 7.1 kann gezeigt werden, dass die in der vorliegenden Arbeit entwickelten Modelle den Anforderungen entsprechen, die in Kapitel 4.1 definiert sind. Die erste **inhaltliche Anforderung** verlangt, **eine Übersicht über software-definierte Plattformen** zu generieren. Dies ist erreicht, indem zuerst das Verständnis für den Begriff „software-definierte Plattform“ geschärft wurde (s. Kapitel 5.1.2). Auf dieser Basis können real verfügbare software-definierte Plattformen untersucht (s. Anhang A.5), deren Eigenschaften systematisch beschrieben und die Plattformen zu Typen aggregiert werden. Diese Plattfortmtypen können für die Zusammenfassung von Anforderungen im Zuge einer Softwareauswahl genutzt werden, wie die Anforderung nach einer Typ-I-Plattform bei e.GO und PTC-Kunde B sowie nach Typ II bei PTC-Kunde A zeigt.

Die **Beschreibung der Integrationsanforderungen von software-definierten Plattformen** erfolgte allgemeingültig, strukturiert nach den vier Dimensionen des Integrationsgegenstands: *technische, organisatorische und prozessuale Integrationsanforderungen* sowie *Anforderungen an die Datenintegration*. Auch diese Anforderungen konnten sich in den beiden Fallstudien, insbesondere bei e.GO und bei PTC-Kunde B, wiederfinden lassen.

Das Ziel, **Informationssystem-Architekturen produzierender Unternehmen systematisch zu beschreiben**, wird mit dem Beschreibungsmodell in Kapitel 6 erreicht. Dabei werden zuerst Informationssysteme strukturiert und darauf aufbauend Informationssystem-Architekturen abgeleitet. Anhand der Ist-Architekturen können Unternehmen spezifische Handlungsfelder zur Reduktion der IT-Komplexität ihrer Architektur und zur Vorbereitung der Integration von software-definierten Plattformen evaluieren.

Die Anwendung dieses Modells ist, sowohl in den praxisnahen Beschreibungen in Kapitel 6.8 als auch durch die Anwendung bei e.GO und den PTC-Kunden, validiert.

Um den **Unterschied zwischen Ist- und Ziel-Zustand der Informationssystem-Architektur** zu erklären, wird zuerst eine Ziel-Architektur entwickelt. Diese basiert auf den Integrationsanforderungen software-definierter Plattformen sowie den Empfehlungen aus der Literatur zum verbesserten IT-Komplexitätsmanagement. Die Erklärung wird dann über die Zusammenhänge zwischen den Ist-Architekturtypen und der Ziel-Architektur in vier Betrachtungsdimensionen hergeleitet (s. Kapitel 7.2). Diese Erklärung trifft auch auf die Unternehmen der Fallstudien zu – insbesondere bei e.GO konnten die daraus abgeleiteten Handlungsempfehlungen durch den IT-Leiter bestätigt werden (s. Kapitel 9.2.5). Unter Nutzung dieser Erklärung kann mit dem Gestaltungsmodell aus Kapitel 8 die **Einführung von software-definierten Plattformen gestaltet** werden. Dieser Einführungsprozess für Unternehmen konnte aus der Anbietersicht durch Gespräche mit dem Softwarehersteller PTC validiert werden; PTC erfährt bei seinen Kunden im Optimalfall ähnliche Vorgehensweise (s. Kapitel 9.3.2).

Als letztes inhaltliches Ziel ist eine **anwendernahe Unterstützung durch praxisnahe Empfehlungen** gefordert. Dies ist im Rahmen der vorherigen Kapitel durch eine Vielzahl von praktischen Beispielen, eine praxisnahe Sprache, umfangreiche Darstellungen sowie zuletzt durch die konkrete Anwendung der Modelle bei produzierenden Unternehmen im Rahmen der Fallstudien umgesetzt.

Auch die **formalen Anforderungen** an die vorliegende Arbeit sind durch die Fallstudien überprüft. Die **Originalität**, also der innovative Beitrag, ist schon durch das Thema selbst gegeben: Die Infrastruktur des *Internet of Production* ist noch in keinem dem Autor bekannten Unternehmen vollständig umgesetzt. Die vorliegende Arbeit gibt in diesem Kontext Vorgehensvorschläge zur Implementierung des *Internets of Production* aus Sicht des Informationsmanagements. Auch aus Sicht des Softwareherstellers PTC wurde in den Expertengesprächen bestätigt, dass mit dieser Arbeit ein Baustein entwickelt wurde, der PTC bisher bei der Umsetzung von Projekten zur Einführung von *ThingWorx* noch fehlt. Weitere innovative Beiträge der Arbeit sind die vollständigen Definitionen von IT-Komplexität und software-definierten Plattformen sowie die konkrete Zielinformationssystem-Architektur für produzierende Unternehmen.

Die **Validität**, also die logische Korrektheit der Modelle, ist erfüllt, wenn die Modelle in der Praxis in verschiedenen Fällen nachvollziehbar eingesetzt werden können (s. DEINDL 2013b, S. 166). Dies wird durch die Fallstudien bestätigt. Die Validität kann jedoch nie definitiv sichergestellt werden, weil sie dem Falsifizierbarkeitskriterium von POPPER unterliegt. Weitere Anwendungen in der Praxis können ihre Richtigkeit jedoch untermauern.

Zur Validität gehört auch das Kriterium der **Reliabilität**, also der Nachweisbarkeit und Wiederholbarkeit. In einer realen Welt, in der kein Unternehmen dem anderen exakt gleicht, kann nie final geprüft werden, ob die entwickelten Modelle auf jedes Unterneh-

men anwendbar sind. Soweit es im Rahmen dieser Arbeit möglich war, wurde die Zuverlässigkeit der Modelle jedoch anhand von Gesprächen mit Experten und der Anwendung in den Fallstudien sichergestellt.

Zuletzt zielt das formale Kriterium der **Utilität** in die gleiche Richtung wie das inhaltliche Kriterium, praxisnahe Hilfestellungen zu geben. Da die Methoden aus der Praxis im Rahmen der Aktionsforschung entstanden sind, ist ihre Anwendbarkeit sichergestellt. Auf die Einfachheit und Klarheit der Modelle und deren Beschreibung wurde besonderer Wert gelegt. Einen wirtschaftlichen Nutzen stiftet die Anwendung der Modelle durch eine Verringerung der IT-Komplexität und der damit verbundenen Komplexitätskosten, auf die in der vorliegenden Arbeit jedoch nicht genauer eingegangen wird.

10 Zusammenfassung und Ausblick

10.1 Zusammenfassung

„Data is the new oil“ ist der vielzitierte Ausspruch von CLIVE HUMBY (HUMBY 2006), der unser aktuelles Industriezeitalter kurz und knapp beschreibt. Dass er mit dieser Aussage bis heute Recht behält, zeigt eine Vielzahl von Fakten: Unternehmen, die datenbasierte Dienstleistungen anbieten, sind inzwischen mehr wert als klassische Industrieunternehmen (s. GIERBERG). Die Anzahl an Geräten, die Daten bereitstellen, wächst ständig (s. GARTNER 2017c). Experten im Kontext digitaler Daten gehören heute zu den gefragtesten Arbeitskräften (s. SCHMIDT 2017). **Produzierende Unternehmen in Deutschland haben indes Schwierigkeiten mit den nötigen Veränderungen.** Zwar haben sie die Potenziale der vernetzten Digitalisierung erkannt (s. HEIMANN ET AL. 2018, S. 17), allerdings führen Herausforderungen bei der Umsetzung dazu, dass vorhandene Daten nicht zur Entscheidungsfindung genutzt werden. (STICH ET AL. 2018, S. 26)

Die Gründe dessen sind vielfältig: Gerade in kleineren Unternehmen fehlen oft Mitarbeiter, die sich mit dem Thema Digitalisierung beschäftigen. Vielmehr verbleibt die IT-Abteilung im Stadium des Technologie- oder Prozesspartners. Weiterhin ist, bedingt durch das Wachstum an IT-Lösungen im Unternehmen, eine **nicht beherrschbare Komplexität der Informationssystem-Architektur** entstanden. Dies führt dazu, dass Daten mehrfach gehalten und teilweise manuell übertragen werden, und somit die Datenintegrität nicht sichergestellt werden kann. Somit kann der Auftragsabwicklungsprozess, der die Arbeit produzierender Unternehmen strukturiert, nicht durchgängig in Informationssystemen umgesetzt werden. Vielmehr wird oft eine Vielzahl an Lösungen genutzt, die nur einzelne Prozessschritte unterstützen. (STICH ET AL. 2018, S. 26-27)

Vor diesem Hintergrund hatte die vorliegende Arbeit zum **Ziel**, einen **Gestaltungsvorschlag für Informationssystem-Architekturen produzierender Unternehmen** zu entwickeln, der die **Integration von software-definierten Plattformen ermöglicht** und damit die Umsetzung des *Internet of Production* unterstützt. Die Basis hierfür wurde in der **Definition** relevanter Begriffe im Kontext von Informationssystemen und Informationssystem-Architekturen gelegt. Zusätzlich wurden **Grundlagen** des Managements von Informationssystem-Architekturen und die Vision vom Plattformeinsatz zur Ermöglichung der vernetzten Digitalisierung vorgestellt.

Aufbauend auf der Abgrenzung des relevanten Untersuchungsbereichs wurden **bestehende Ansätze** aus der Wissenschaft und Praxis bewertet. Die untersuchten Ansätze tragen alle nur teilweise zum Ziel der Arbeit bei. Ein ganzheitlicher und praxisnaher Ansatz zur Transformation von Informationssystem-Architekturen produzierender Unternehmen, inklusive der dazu notwendigen Beschreibungs- und Vorgehensmodelle zur Integration software-definierter Plattformen, fehlt bisher.

Für den Begriff der software-definierten Plattform bestand bisher keine umfangreiche Definition, die den Einsatz einer solchen Plattform für produzierende Unternehmen berücksichtigt. Deshalb wurde für die vorliegende Arbeit eine **eigene Definition** hergeleitet, welche die *technologische Abstraktionsschicht*, den *ubiquitären Datenzugriff*, die *Datenverarbeitung* und *-veredelung* sowie die *Datensicherheit* als zentrale Elemente beschreibt und diese in die Infrastruktur des *Internet of Production* einordnet: Eine software-definierte Plattform kann die Kernelemente dessen – die *Middleware+* sowie den Bereich *Smart Data* – funktional abdecken. Um den Bereich der **software-definierten Plattformen** weiter zu **systematisieren**, wurden deren Merkmale morphologisch hergeleitet. Dabei wurden sowohl Aus- und Eingang der Plattform über *Schnittstellen* und *Informationsbereitstellung* beschrieben als auch interne Merkmale des *Datenmanagements*, der zugrundeliegenden *Datenbankkonzepte* und die möglichen *Stufen der Datenverarbeitung* dargelegt. Unterstützende Merkmale von *IT-Sicherheitsfunktionen* und *Entwicklungsumgebungen* vervollständigen die Merkmalsliste. Mithilfe der Merkmale und anhand der Untersuchung von software-definierten Plattformen in der Praxis konnten sodann drei aufeinander aufbauende **Plattformtypen** identifiziert werden. Den geringsten Funktionsumfang nehmen dabei Plattformen ein, die hauptsächlich die intelligente Vernetzung von Informationssystemen ermöglichen. Darüber hinaus gibt es Plattformen, die einen spezifischen Anwendungsfall im Unternehmen unterstützen, wie beispielsweise die Datenerhebung auf dem Shopfloor. Den vollen Funktionsumfang bietet ein dritter Plattfortmty, mit dem die Umsetzung von digitalen Schatten möglich wird. Um Anforderungen an die notwendige Transformation der Informationssystem-Architektur zu stellen, wurden mithilfe der Plattfortmtyten **Integrationsanforderungen** aus den Merkmalen abgeleitet und anhand der vier Dimensionen des Integrationsgegenstands strukturiert. Diese vier Dimensionen – *Technik*, *Organisation*, *Prozesse* und *Daten* – sind strukturgebend für die weiteren, im Rahmen dieser Arbeit entwickelten Modelle.

Im deutschen Sprachraum hat sich die Informationssystem-Definition von KRCMAR durchgesetzt, welche diese als soziotechnische Systeme beschreibt. Diese wurde für die vorliegende Arbeit übernommen und auf dieser Grundlage integrationsrelevante Merkmale von Informationssystemen, strukturiert nach den vier Dimensionen des Integrationsgegenstands, abgeleitet. Diese **Merkmale beschreiben Informationssysteme morphologisch**, beispielsweise anhand des Outsourcinggrads, des Systemkonzepts oder der Informationsqualität. Um die Beschreibung nicht nur von einzelnen Informationssystemen, sondern auch von Informationssystem-Architekturen zu ermöglichen, wurden einige praxistaugliche **Visualisierungen von Informationssystem-Architekturen** vorgestellt. Diese nutzen jeweils eine Auswahl der Merkmale der Informationssysteme und stellen diese in bis zu vier Dimensionen dar. Außerdem enthalten solche Visualisierungen zusätzliche Informationen, wie beispielsweise die Schnittstellenarchitektur, die nicht anhand eines einzelnen Informationssystems dargestellt werden können. Anhand der vier IT-Organisationstypen von McFARLAN ET AL. und der beiden Architekturparadigmen *zentral* und *dezentral* wurden **fünf Informationssystem-Architekturtypen hergeleitet** und anhand von Beispielen aus der Praxis vorgestellt.

Anhand der Integrationsanforderungen von software-definierten Plattformen und unter Nutzung von allgemeingültigen Anforderungen an komplexitätsarme Informationssystem-Architekturen wurde dann eine **Zielinformationssystem-Architektur produzierender Unternehmen bei software-definierten Plattformen** entwickelt. Auch diese wurde in den vier Dimensionen *Technik*, *Organisation*, *Prozesse* und *Daten* vorgestellt. Im Zusammenhang mit den fünf Informationssystem-Architekturtypen wurden dann die **Zusammenhänge zur Zielarchitektur erklärt** und **Entscheidungskriterien** für Unternehmen abgeleitet. Letztere wurden in Verbindung mit den Plattfortmtypen gesetzt, um Unternehmen eine Priorisierung der Umsetzungsziele, abhängig von der Plattform, die sie implementieren möchten, zu ermöglichen.

Daran anschließend wurde eine **Vorgehensweise** zur systematischen Gestaltung von Informationssystem-Architekturen beschrieben. Diese basiert auf dem Problemlösungszyklus von HABERFELLNER ET AL. und ermöglicht die Zielformulierung, Situationsanalyse und Synthese sowie Lösungsanalyse mit den beschriebenen Modellen. Um konkret eine software-definierte Plattform auszuwählen und zu implementieren, wurden danach **Hinweise zum Auswahl- und Einführungsprozess** gegeben.

Zum Abschluss der Arbeit erfolgte die **Überprüfung des entwickelten Vorgehens** durch die Evaluation in zwei Fallstudien mit den Firmen e.GO und PTC. Aus zwei unterschiedlichen Sichten – der des Anwenders und der des Anbieters – konnte die Praxistauglichkeit der Ergebnisse erfolgreich überprüft werden. Es konnte gezeigt werden, dass die inhaltlichen und formalen Anforderungen an die vorliegende Arbeit erfüllt wurden.

Damit liefert die vorliegende Arbeit einen **wissenschaftlichen Beitrag** zur Strukturierung der, von der ACATECH vorgeschlagenen, aber noch wenig beforschten, software-definierten Plattformen. Weiterhin wurde die Forschungslücke von konkret anwendbaren Methoden zum Management von IT-Komplexität bei produzierenden Unternehmen geschlossen, indem verfügbare Vorschläge zu einem Gesamtmodell aggregiert wurden. Die Arbeit leistet damit insgesamt einen Beitrag zur Detaillierung des Aachener Modells des *Internet of Production* und damit auch zur weiteren Umsetzung der Industrie 4.0 in Deutschland.

Auf den **praktischen Nutzen** wurde bei der Entwicklung der vorliegenden Modelle großer Wert gelegt. So liefert die Arbeit nicht nur einen aktuellen Überblick der technischen Entwicklung im Bereich der software-definierten Plattformen und der Informationssysteme in produzierenden Unternehmen, sondern gibt auch konkrete Gestaltungsvorschläge zur Transformation von Informationssystem-Architekturen. Durch viele grafische Darstellungen wurde zudem zum besseren Verständnis der teilweise komplexen Sachverhalte beigetragen.

10.2 Ausblick

An verschiedenen Stellen wurde in den obigen Kapiteln dieser Arbeit darauf hingewiesen, dass diese Arbeit **nur einen Überblick** über das Thema geben kann und von einer weiteren Vertiefung abgesehen wurde, um den Umfang der Arbeit handhabbar

zu gestalten. So wurden beispielsweise nur drei Typen software-definierter Plattformen untersucht, obwohl die große Vielfalt an verfügbaren Plattformen eine deutlich höhere Zahl an Typen ermöglichen würde. Dies liegt auch darin begründet, dass sich der Softwaremarkt noch etabliert und stark verändert und es nicht Zielsetzung der vorliegenden Arbeit ist, hier tiefgreifende Analysen durchzuführen. Hier besteht also noch **weiterer Forschungsbedarf**. Genauso wurde erstmals eine detaillierte Zielinformationssystem-Architektur für produzierende Unternehmen entwickelt, die jedoch nicht in Bezug auf verschiedene Arten produzierender Unternehmen diversifiziert wurde, obwohl beispielsweise bei Sondermaschinenbauern teilweise andere Prozesse vorliegen als bei Schüttgutproduzenten. Die Details der Zielarchitektur, beispielsweise eine optimale Produktdatenstruktur, sollten ebenfalls weiter verfeinert werden. Auch der Auswahl- und Implementierungsprozess software-definierter Plattformen wurde nur skizziert. Es ist eine eigene Arbeit wert, diesen Prozess ausführlich zu untersuchen und weiterzuentwickeln.

Die **Entwicklung digitaler Plattformen aller Formen wird in den nächsten Jahren rasant weitergehen**. Auf Ebene der Serviceplattformen wird die Plattformökonomie in noch mehr Branchen Einzug halten – auch in solche, die bisher noch klassische Geschäftsmodelle verfolgen. Software-definierte Plattformen sind das Fundament für eine noch stärker nach außen gerichtete Informationsindustrie (s. PARKER ET AL. 2016, S. 11). In der Ebene der vernetzten, physischen Plattformen werden smarte Geräte noch mehr Bereiche des privaten und beruflichen Lebens durchdringen. In dieser dynamischen Welt ist der Forschungsbedarf zur Strukturierung, Bewertung und Rationalisierung nicht exakt zu bemessen, aber in jedem Fall vorhanden.

11 Literaturverzeichnis

- ABTS, D.; MÜLDER, W.: Grundkurs Wirtschaftsinformatik. Eine kompakte und praxisorientierte Einführung. 9., erw. u. akt. Auflage. Springer, Wiesbaden [u. a.] 2017.
- ACKERMANN, J.: Modellierung, Planung und Gestaltung der Logistikstrukturen kompetenzzellenbasierter Netze. Inst. für Betriebswiss. und Fabrikssysteme, Chemnitz 2007. - Zugl.: Chemnitz, Techn. Univ., Diss.
- ADOLPHS, P.; BEDENBENDER, H.; DIRZUS, D.; EHLICH, M.; EPPLE, U.; HANKEL, M.; HEIDEL, R.; HOFFMEISTER, M.; HUHLE, H.; KÄRCHER, B.; KOZIOLEK, H.; PICHLER, R.; POLLMEIER, S.; SCHEWE, F.; WALTER, A.; WASER, B.; WOLLSCHLAEGER, M.: Statusreport Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0). Düsseldorf, April 2015. https://www.vdi.de/fileadmin/user_upload/VDI-GMA_Statusreport_Referenzarchitekturmodell-Industrie40.pdf (zuletzt geprüft am: 29.03.2018).
- AHUJA, S. P.; MUPPARAJU, N.: Performance Evaluation and Comparison of Distributed Messaging Using Message Oriented Middleware. In: Computer and Information Science 7(2014)4, S. 9-20. <http://www.ccsenet.org/journal/index.php/cis/article/view/36416> (zuletzt geprüft am: 29.03.2018).
- ALAM, M. F.; KATSIKAS, S.; BELTRAMELLO, O.; HADJIEFTHYMIADES, S.: Augmented and virtual reality based monitoring and safety system: A prototype IoT platform. In: Journal of Network and Computer Applications 89(2017), S. 109-119.
- ALBAYRAK, C.D.; GADATSCH, A.: Komplexitätsmanagement als Instrument des IT-Controllings. Ansätze im Kontext Standardisierung versus Flexibilität bei der Komplexitätssteuerung. In: Controlling 26(2014)12, S. 680-685.
- ALLWEYER, T.: BPMN 2.0 - Business Process Model and Notation. Einführung in den Standard für die Geschäftsprozessmodellierung. 3., aktualis. u. erw. Auflage. Books on Demand, Norderstedt 2015.
- ALPAR, P.; ALT, R.; BENSBURG, F.; GROB, H. L.; WEIMANN, P.; WINTER, R.: Anwendungsorientierte Wirtschaftsinformatik. Strategische Planung, Entwicklung und Nutzung von Informationssystemen. 8., überarb. Auflage. Springer Vieweg, Wiesbaden 2016.
- ALT, R.; PUSCHMANN, T.: Digitalisierung der Finanzindustrie. Grundlagen der Fintech-Evolution. Springer Gabler, Heidelberg [u. a.] 2016.
- ALTENDORFER-KAISER, S.: Referenzmodellierung - Eine Möglichkeit zur Strukturgebung innerhalb der Informationslogistik? In: Industrial Engineering und Management. Beiträge des Techno-Ökonomie-Forums der TU Austria. Hrsg.: H. Biedermann. Springer, Wiesbaden [u. a.] 2016, S. 165-176.
- ANDRIOLE, S. J.: Who owns IT? In: Communications of the ACM 58(2015)3, S. 50-57.

- ANUFF, E.: APIs are Different than Integration. Internet and enterprise technologies unite in a new enterprise foundation. Hrsg.: APIGEE. <https://pages.apigee.com/rs/apigee/images/APIs-not-integration-ebook-05-2014.pdf> (zuletzt geprüft am: 29.03.2018).
- ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH (Hrsg.): [Abschlussbericht] Smart Service Welt - Umsetzungsempfehlungen für das Zukunftsprojekt 'Internetbasierte Dienste für die Wirtschaft'. Berlin März 2015. http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Projekte/Laufende_Projekte/Smart_Service_Welt/Smart_Service_Welt_2015/BerichtSmartService2015_D_lang_bf.pdf (zuletzt geprüft am: 29.03.2018).
- ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH (Hrsg.): Smart Service Welt. Digitale Serviceplattformen - Praxiserfahrungen aus der IndustrieBerlin April 2016. http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Publikationen/Kooperationspublikationen/BerichtSmartService2016_DE_barrierefrei.pdf (zuletzt geprüft am: 29.03.2018).
- ARIKAN, E.: Channel Polarization. A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. In: IEEE Transactions on Information Theory 55(2009)7, S. 3051-3073.
- ARMAS, I.: The process oriented complexity of enterprise's information system. In: Hyperion International Journal of Econophysics 4(2011)1, S. 117-127.
- ARTETA, B. M.; GIACHETTI, R. E.: A measure of agility as the complexity of the enterprise system. In: Robotics and Computer-Integrated Manufacturing 20(2004)6, S. 495-503.
- ASHBY, W. R.: An introduction to cybernetics. Chapman & Hall, London 1957.
- ATKEARNEY; BDI (Hrsg.): Wie mutig ist der Mittelstand? Das Mittelstandspanel 2017 im Auftrag von A.T.Kearney und BDI. Hamburg, September 2017. <http://www.atkearney.de/documents/856314/14365786/BDI-Mittelstandspanel.pdf/e4c92157-63e1-1a6f-c4c8-fe9b9db20a83> (zuletzt geprüft am: 29.03.2018).
- BABTEC (Hrsg.): CAQ-Software von Babtec: Für erfolgreiches Qualitätsmanagement. Hrsg.: Babtec. <https://www.babtec.de/caq-software.html> (zuletzt geprüft am: 18.03.2018).
- BACH, T.; FREY, D.; KREBS, U.; RESCHKE, J.; SCHIEMANN, D.; SONTOW, K.; TREUTLEIN, P.; WETZCHEWALD, P.; SCHRÖER, T.: Marktspiegel Business Software ERP/PPS 2017/2018. Anbieter - Systeme - Projekte. 9. Auflage, revidierte Ausgabe. Aachen 2017.
- BAHILL, A. T.; GISSING, B.: Re-evaluating systems engineering concepts using systems thinking. In: IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) 28(1998)4, S. 516-527.
- BANGE, C.: Werkzeuge für analytische Informationssysteme. In: Analytische Informationssysteme. Business Intelligence-Technologien und -Anwendungen. Hrsg.: P.

- Gluchowski; P. Chamoni. 5., vollst. überarb. Auflage. Springer, Berlin [u. a.] 2016, S. 97-126.
- BASZENSKI, N.: Methodensammlung zur Unternehmensprozessoptimierung. Köln 2008. <http://www.peter-michael-kurz.de/dogmacms/assets/shop/33.pdf> (zuletzt geprüft am: 29.03.2018).
- BAUERNHANSL, T.: Die Vierte Industrielle Revolution – Der Weg in ein wertschaffendes Produktionsparadigma. In: Handbuch Industrie 4.0; Bd. 4: Allgemeine Grundlagen. Hrsg.: B. Vogel-Heuser; T. Bauernhansl; M. ten Hompel. 2. Auflage. Springer, Berlin [u. a.] 2017, S. 1-31.
- BAUERNHANSL, T.; KRÜGER, J.; REINHART, G.; SCHUH, G.: WGP-Standpunkt Industrie 4.0. Darmstadt 2016. https://www.ipa.fraunhofer.de/content/dam/ipa/de/documents/Presse/Presseinformationen/2016/Juni/WGP_Standpunkt_Industrie_40.pdf (zuletzt geprüft am: 29.03.2018).
- BAUMS, A.: Analyse. Was sind digitale Plattformen? In: Wie digitale Plattformen unsere Wirtschaft verändert - und wie die Politik gestalten kann. Kompendium Digitale Standortpolitik. Hrsg.: A. Baums; M. Schössler; B. Scott. Berlin Oktober 2015, S. 13-24. <http://plattform-maerkte.de/wp-content/uploads/2015/11/Kompendium-High.pdf> (zuletzt geprüft am: 29.03.2018).
- BECKER, J.; ROSEMAN, M.: Informationsmanagement - ein Beitrag zur Beherrschung von Komplexität? In: Komplexitätsmanagement. Hrsg.: D. Adam. Schriften zur Unternehmensführung; Bd. 61. Gabler, Wiesbaden 1998, S. 111-124.
- BEETZ, R. K.: Wirkung von IT-Governance auf IT-Komplexität in Unternehmen. Beeinflussung der IT-Redundanz durch Verantwortungsteilung im IT-Projektportfoliomanagement. Springer Gabler, Wiesbaden 2014.
- BEIMS, M.; ZIEGENBEIN, M.: IT-Service-Management in der Praxis mit ITIL. Der Einsatz von ITIL Edition 2011, ISO/IEC 20000:2011, COBIT 5 und PRINCE2. 4., überarb. und erw. Auflage. Hanser, München [u. a.] 2014.
- BELMS, M.: Frameworks und Methoden für das IT-System- und IT-Service-Management. In: Handbuch IT-Systemmanagement. Handlungsfelder, Prozesse, Managementinstrumente, Good-Practices. Hrsg.: E. Tiemeyer. Hanser, München [u. a.] 2016, S. 109-161.
- BENGEL, G.; BAUN, C.; KUNZE, M.; STUCKY, K.-U.: Masterkurs Parallele und Verteilte Systeme. Grundlagen und Programmierung von Multicore-Prozessoren, Multiprozessoren, Cluster, Grid und Cloud. 2., erw. u. aktualis. Auflage. Springer, Wiesbaden [u. a.] 2015.
- BENRA, J. T.; HALANG, W. A.: Einführung. In: Software-Entwicklung für Echtzeitsysteme. Hrsg.: J. T. Benra; W. A. Halang. Springer, Berlin [u. a.] 2009, S. 1-8.

- BENTE, S.: Kollaborative Enterprise-Architektur. Managementwerkzeug für komplexe IT-Systeme. In: Komplexitätsmanagement in Unternehmen. Herausforderungen im Umgang mit Dynamik, Unsicherheit und Komplexität meistern. Hrsg.: K.-P. Schoeneberg. Springer Gabler, Wiesbaden 2014, S. 187-223.
- BERCHTENBREITER, R.: Grundlagen von Content-Management-Systemen und Ansätze ihrer Bedeutung für das CRM. In: IT-Systeme im CRM. Aufbau und Potenziale. Hrsg.: H. Hippner. Gabler, Wiesbaden 2004, S. 209-240.
- BERESFORD, J.; FENN, J.: CIO Futures: The IT Organization in 2030. <https://www.gartner.com/doc/3480817?ref=unauthreader&srcId=1-3478922254#1141951085> (zuletzt geprüft am: 29.03.2018).
- BERNARDY, A.: General classification of mobile communication technologies like 5G for the production and logistic. In: Logistisches Produktionsmanagement. Hrsg.: H. E. Zsifkovits; S. Altendorfer-Kaiser. epubli, Berlin 2017, S. 11-20.
- BERTALANFFY, L. von: Problems of Organic Growth. In: Nature 163(1949)4135, S. 156-158.
- BERTHEL, J.: Information. In: Handwörterbuch der Betriebswirtschaft. Hrsg.: W. Wittmann. Schäffer-Poeschel, Stuttgart 1975, S. 1860-1880.
- BIETHAHN, J.; MUCKSCH, H.; RUF, W.: Ganzheitliches Informationsmanagement. 6., vollst. überarb. und neu gefasste Auflage. Oldenbourg, München [u. a.] 2004.
- BLECK, R.: Protecting smart devices and applications. Guidance on security throughout the IoT ecosystem. 2016. <https://www.ptc.com/en/internet-of-things/protecting-smart-devices-and-applications/> (zuletzt geprüft am: 29.03.2018).
- BLEIDER, M.; HORSTMANN, G.; KELKAR, O.; SCHMID, F.; WOLTER, B.: [Leitfaden] Industrie 4.0 – Die neue Rolle der IT. Hrsg.: Bitkom e. V. Berlin 2016. <https://www.bitkom.org/Publikationen/2016/Leitfaden/Industrie-40-Die-neue-Rolle-der-IT/160421-LF-Industrie-40-Die-neue-Rolle-der-IT.pdf> (zuletzt geprüft am: 29.03.2018).
- BLEIDER, M.; ZELLER, V.; SERVOS, M.; KÖNIGS, M.; BELINA, J.; CHAZLI, S.: [Schlussbericht] DigiKMU- Entwicklung einer Methode zur Unterstützung von KMU im Maschinen- und Anlagenbau bei der Umsetzung von Industrie 4.0 im Bereich der CAD-CAM-NC-Verfahrenskette. Aachen 28.07.2017. https://www.fir.rwth-aachen.de/sites/default/files/projekt/digikmu_schlussbericht_20171004.pdf (zuletzt geprüft am: 29.03.2018).
- BLEIHOLDER, J.; SCHMID, J.: Datenintegration und Deduplizierung. In: Daten- und Informationsqualität. Auf dem Weg zur Information Excellence. Hrsg.: K. Hildebrand; M. Gebauer; H. Hinrichs; M. Mielke. 3., erw. Auflage. Springer, Wiesbaden [u. a.] 2015, S. 121-140.
- BOCK, A.: [Pressemitteilung] Manufacturing Execution Systeme (MES) sind die Basis für Industrie 4.0. Studie zur Digitalisierung der Produktion belegt: Mit einem

- durchgängig eingesetzten MES als Teil der strategischen IT-Landschaft in der Produktion stellen Unternehmen eine wichtige Weiche für die digitale Zukunft-Frankfurt 01.09.2015. <https://www.bearingpoint.com/de-de/ueber-uns/pressemitteilungen-und-medienberichte/pressemitteilungen/manufacturing-execution-systeme-mes-sind-die-basis-fuer-industrie-40/> (zuletzt geprüft am: 29.03.2018).
- BÖHM, R.; FUCHS, E.: System-Entwicklung in der Wirtschaftsinformatik. vdf, Zürich 2002.
- BOOCH, G.: Measuring Architectural Complexity. In: IEEE Software 25(2008)4, S. 14-15.
- BOOS, W.; VÖLKER, M.; SCHUH, G.: Grundlagen des Managements produzierender Unternehmen. In: Strategie und Management produzierender Unternehmen. Hrsg.: G. Schuh; A. Kampker. Handbuch Produktion und Management; Bd. 1. Springer, Berlin [u. a.] 2011, S. 1-62.
- BOSSERT, O.; LAARTZ, J.: Neue Kriterien für die Softwareauswahl digitaler Plattformen. In: ERP-Management 13(2017)2, S. 29-40.
- BRANDT, M.: Office-Software dominiert deutsche Büros. 02.04.2015. <https://de.statista.com/infografik/3367/nutzung-von-office-software-in-deutschland/> (zuletzt geprüft am: 29.03.2018).
- BRECHER, C.; KLOCKE, F.; SCHMITT, R.; SCHUH, G. (Hrsg.): AWK Aachener Werkzeugmaschinen-Kolloquium 2017: Internet of Production für agile Unternehmen. Ap-primus, Aachen 2017.
- BROCKE, J.; SIMONS, A.; NIEHAVES, B.; REIMER, K.; PLATTFAUT, R.; CLEVEN, A.: Reconstructing the Giant: On the impact of rigour in documenting the literature search process. In: ECIS 2009 Proceedings(2009)161, o. S. https://www.iwi.uni-hannover.de/fileadmin/wirtschaftsinformatik/lehre/RECONSTRUCTING_THE_GIANT__ON_THE_IMPORTANCE_OF_RIGOUR_IN_DOCUMENT.pdf (zuletzt geprüft am: 29.03.2018).
- BROSZE, T.; NOVOSZEL, T.; SCHMIDT, C.: Das 3PhasenKonzept zur Auswahl von ERP-/PPS-Systemen Bewährte Werkzeuge zur Reorganisation, Potenzialanalyse und Bewertung des Systemeinsatzes. In: UdZ - Unternehmen der Zukunft 10(2009)1, S. 54-59.
- BROY, M. (Hrsg.): Cyber-Physical Systems. Innovation durch Softwareintensive eingebettete SystemeSpringer. Berlin [u. a.] 2010.
- BROY, M.; KUHRMANN, M.: Projektorganisation und Management im Software Engineering. Springer Vieweg, Berlin [u. a.] 2013.
- BRÜHWILER, J.: Gesamtbankensysteme Publi-Forum. Je standardisierter, je günstiger? In: Schweizer Bank(2012)11, S. 47-48.

- BRUNELLI, M.: A Quick History of PTC and PTC Creo 14.08.2014. <https://www.ptc.com/en/cad-software-blog/a-quick-history-of-ptc-and-ptc-creo> (zuletzt geprüft am: 14.04.2018).
- BUCHER, T.; FISCHER, R.; KURPJUWEIT, S.; WINTER, R.: Analysis and Application Scenarios of Enterprise Architecture. An Exploratory Study. In: Proceedings / 10th IEEE International Enterprise Distributed Object Computing Conference. Hong Kong, China, October 16 - 20, 2006. Hrsg.: IEEE Computer Society. IEEE, Los Alamitos, Calif. 2006, 28ff.
- BUCHTA, D.; EUL, M.; SCHULTE-CROONENBERG, H.: Strategic IT Management. Increase value, control performance, reduce costs. 3. Auflage. Gabler, Wiesbaden 2010.
- BULLINGER, H.-J.; GANZ, W.; NEUHÜTTLER, J.: Smart Services - Chancen und Herausforderungen digitalisierter Dienstleistungssysteme für Unternehmen. In: Dienstleistungen 4.0; Bd. 1: Konzepte - Methoden - Instrumente. Hrsg.: M. Bruhn; K. Hadwich. Springer, Wiesbaden [u. a.] 2017, S. 97-120.
- BUNA, S.: Learning GraphQL and Relay. Build data-driven React applications with ease using GraphQL and Relay. Packt Publishing, Birmingham, UK 2016.
- BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (Hrsg.): IT-Grundschutz. M 4.237 Sichere Grundkonfiguration eines IT-Systems. Hrsg.: Bundesamt für Sicherheit in der Informationstechnik. https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_content/m/m04/m04237.html?nn=6604938 (zuletzt geprüft am: 29.03.2018).
- BUNDESAMT FÜR WIRTSCHAFT (Hrsg.): Merkblatt für Unternehmen des produzierenden Gewerbes zu den gesetzlichen Regelungen nach §§ 40 ff. Erneuerbare-Energien-Gesetz 2012. Eschborn 2013. https://www.energieeffizienz-ihk.de/wp-content/uploads/2013/05/130507_EEG_Ausgleichsregelung_Merkblatt_fuer_Unternehmen.pdf (zuletzt geprüft am: 29.03.2018).
- CARR, N. G.: Does It Matter? Information Technology and the Corrosion of Competitive Advantage. Harvard Business Review Press, Cambridge (MA) 2004.
- CARR, N. G.: IT Doesn't Matter. In: Harvard Business Review 38(2003)5, S. 24-38.
- CARREZ, F.; ELSALEH, T.; GOMEZ, D.; SANCHEZ, L.; LANZA, J.; GRACE, P.: A Reference Architecture for federating IoT infrastructures supporting semantic interoperability. In: 5G - European roadmap, global impact. EuCNC - European Conference on Networks and Communications. Hrsg.: P. Pirinen. IEEE, Piscataway (NJ) 2017, S. 1-6.
- CASTILLO ARAUJO, V. M.: Big Data y el camino a la inteligencia artificial autonómica. Big Data and the road towards and autonomic artificial intelligence. In: VIII Reunión Internacional de Gestión y Desarrollo. Las Metas del Milenio al 2015. Hrsg.:

- AVEGID-AIGID. UCV, AVEGID 2015. <http://avegid-aigid.com/assets/victor-castillo-ponencia.pdf> (zuletzt geprüft am: 30.03.2018).
- CHALMERS, A. F.: What is this thing called science? Fourth edition. Open University Press, Maidenhead, Berkshire 2013.
- CHAMONI, P.; GLUCHOWSKI, P.: Analytische Informationssysteme. Einordnung und Überblick. In: Analytische Informationssysteme. Business Intelligence-Technologien und -Anwendungen. Hrsg.: P. Gluchowski; P. Chamoni. 5., vollst. überarb. Auflage. Springer, Berlin [u. a.] 2016, S. 3-12.
- CLARK, C.: The conditions of economic progress. In: The Economic Journal 51(1941)201, S. 120-124.
- CODD, E. F.: A relational model of data for large shared data banks. In: Communications of the ACM 13(1970)6, S. 377-387.
- COLLOPY, P. D.: Systems engineering theory. What needs to be done. In: 2015 Annual IEEE Systems Conference (SysCon) Proceedings // 9th Annual IEEE International Systems Conference (SysCon), 2015. 13. - 16. April 2015, Vancouver (BC); proceedings. Hrsg.: B. Rassa. IEEE, Piscataway (NJ) 2015, S. 536-541.
- COPERNICUS, N.: De Revolutionibus Orbium Coelestium. Digitalisat der 1. Auflage. Petreium, Nürnberg 1543. Harvard University, Cambridge (MA) o. J. http://ads.harvard.edu/books/1543droc.book/1543droc_front.pdf (zuletzt geprüft: 18.04.2018).
- CROOK, S.; MACGILLIVRAY, C.: IDC MarketScape: Worldwide IoT Platforms (Software Vendors). Vendor Assessment, 2017. https://www.ge.com/de/sites/www.ge.com/de/files/IDC%20MarketScape_Worldwide%20IoT%20Platforms_Software%20Vendors_US42033517%5B1%5D.pdf (zuletzt geprüft am: 30.03.2018).
- CWA 16525: Multilingual electronic cataloguing and classification in eBusiness - Classification Mapping for open and standardized product classification usage in eBusiness. EUROPEAN COMMITTEE FOR STANDARDIZATION (Hrsg.); ICS 01.140.20; 35.240.63. CEN, Brussels, December 2012.
- DANILUK, D.; WOLF, O.; TEN HOMPEL, M.: Industrie-4.0-fähige Software-Dienste auf Basis von Cloud Computing. In: Handbuch Industrie 4.0; Bd. 3: Logistik. Hrsg.: B. Vogel-Heuser; T. Bauernhansl; M. ten Hompel. 2. Auflage. Springer Vieweg, Berlin 2016, S. 201-218.
- DATA INTERCHANGE PLC (Hrsg.): EDI Core Standards. Peterborough 2012. <http://www.datainterchange.com/Downloads/Support/Manuals/EPIC/VM-0001-11%20EDI%20Core%20Standards.pdf> (zuletzt geprüft am: 30.03.2018).
- DEINDL, M.: Anwendungsorientiertes Informationsmanagement. Aufgaben und Methoden der Informationsmanagementforschung im RWTH Aachen Campus-Cluster Logistik. [Vortrag im Rahmen der mündlichen Doktorprüfung], FIR e. V. an der RWTH Aachen, Aachen 16.09.2013. [öffentlich verfügbar über die Bibliothek des FIR]. [=2013a]

- DEINDL, M.: Gestaltung des Einsatzes von intelligenten Objekten in Produktion und Logistik. Schriftenreihe Rationalisierung; Bd. 118. Hrsg.: G. Schuh. Apprimus, Aachen 2013. - Zugl.: Aachen, Techn. Hochsch., Diss., 2013. [=2013b]
- DERN, G.: Integrationsmanagement in der Unternehmens-IT. Systemtheoretisch fundierte Empfehlungen zur Gestaltung von IT-Landschaft und IT-Organisation. Vieweg+Teubner, Wiesbaden 2011.
- DERN, G.; JUNG, R.: IT-Architektur-Governance auf Basis von Kennzahlen zur Komplexitätsmessung. In: Controlling 21(2009)12, S. 669-672.
- DERN, G.: Management von IT-Architekturen. 3., durchges. Auflage. Springer, Wiesbaden [u. a.] 2009.
- DEUSE, J.; EROHIN, O.; LIEBER, D.: Wissensentdeckung in vernetzten, industriellen Datenbeständen. In: Industrie 4.0. Wie intelligente Vernetzung und kognitive Systeme unsere Arbeit verändern. Hrsg.: W. Kersten; H. Koller; H. Lödding. GITO, Berlin 2014, S. 373-395.
- DEUSE, J.; WEISNER, K.; HENGSTEBECK, A.; BUSCH, F.: Gestaltung von Produktionssystemen im Kontext von Industrie 4.0. In: Zukunft der Arbeit in Industrie 4.0. Hrsg.: A. Botthof; E. A. Hartmann. Springer, Berlin [u. a.] 2015, S. 99-109.
- DIESSNER, H.; FRÜH, F.; GANTNER, M.; HENRICHSEN, F.; KAISER, S.; ROOS, W.; SONTOW, K.; WOLF, G.; ZÖLLER, B.: ECM Trend-Check 2015. Enterprise Content Management im Einfluss globaler Megatrends Berlin September 2015. <https://www.bitkom.org/noindex/Publikationen/2015/Leitfaden/ECM-Trendcheck-2015/ECM-TrendCheck2015-150921.pdf> (zuletzt geprüft am: 30.03.2018).
- DIN EN ISO 9001: Qualitätsmanagementsysteme - Anforderungen. DIN (Hrsg.); ICS 03.100.70; 03.120.10. Beuth Verlag, Berlin, November 2015.
- DIN ISO 9735: Elektronischer Datenaustausch für Verwaltung, Wirtschaft und Transport (EDIFACT). DIN (Hrsg.); ICS 35.240.60. Beuth Verlag, Berlin, November 2002.
- DIN SPEC 91345: Referenzarchitekturmodell Industrie 4.0 (RAMI4.0). DIN (Hrsg.); ICS 03.100.01; 25.040.01; 35.240.50. Beuth Verlag, Berlin, April 2016.
- DREMEL, C.; HERTERICH, M.: Digitale Cloud-Plattformen als Enabler zur analytischen Nutzung von operativen Produktdaten im Maschinen- und Anlagenbau. In: HMD - (HMD Praxis der Wirtschaftsinformatik) 53(2016)5, S. 646-661. <https://doi.org/10.1365/s40702-016-0250-9>.
- DREWEL, M.; GAUSEMEIER, J.; KLUGE, A.; PIERENKEMPER, C.: Erfolgsgarant digitale Plattform – Vorreiter Landwirtschaft. In: Wissenschaftsforum Intelligente Technische Systeme (WInTeSys) 2017. Band 369. Hrsg.: E. Bodden; F. Dressler; R. Dumitrescu; G. Gausemeier; Meyer auf der Hiede, F.; C. Scheytt; A. Trächtler. Hans Gieselmann Druck und Medienhaus, Bielefeld 2017, S. 53-66.

- DÜNNEBACKE, D.: Beschreibung und Typisierung der IT-Unterstützung im Maschinen- und Anlagenbau. Schriftenreihe Rationalisierung; Bd. 136. Apprimus, Aachen 2016. - Zugl.: Aachen, Techn. Hochsch., Diss., 2015.
- DURST, M.: Wertorientiertes Management von IT-Architekturen. Dt. Univ.-Verl., Wiesbaden 2007. - Zugl.: Erlangen, Nürnberg, Univ., Diss., 2007.
- DYCKHOFF, H.; SPENGLER, T. S.: Produktionswirtschaft. Springer, Berlin [u. a.] 2010.
- E.GO MOBILE: [Pressemitteilung] Der e.GO Life wird in Aachen in Serie gebaut. Wirtschaftsminister Garrelt Duin gibt die Standortentscheidung der e.GO Mobile AG bekannt und übergibt Zuwendungsbescheid Aachen 25.04.2017. <http://www.e-go-mobile.com/de/newspool/pressemitteilung-der-e.go-life-wird-in-aachen-in-serie-gebaut/> (zuletzt geprüft am: 06.03.2018).
- E.GO MOBILE: Webseite. <http://www.e-go-mobile.com/> (zuletzt geprüft am: 06.03.2018).
- EHRENSPIEL, K.; MEERKAMM, H.: Integrierte Produktentwicklung. Denkabläufe, Methodeneinsatz, Zusammenarbeit. 5. überarb. u. erw. Auflage. Hanser, München [u. a.] 2013.
- EISEND, M.; KUß, A.: Grundlagen empirischer Forschung. Springer, Wiesbaden [u. a.] 2017.
- ENGELHARDT, E. F.: Hausautomation mit Raspberry Pi. Alarmanlage, Lampen, Heizung, Smart Home, W-LAN & Co: 25 Projekte, die Ihr Leben leichter machen. 3. Auflage. Franzis, Haar bei München 2014.
- ENGELS, G.; PLASS, C.; RAMMIG, F.-J. (Hrsg.): IT-Plattformen für die Smart Service Welt. Verständnis und Handlungsfelder. Juni 2017. http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Publikationen/acatech_diskutiert/acatech_DISKUSSION_IT-Plattformen_WEB.pdf (zuletzt geprüft am: 30.03.2018).
- ERICSSON (Hrsg.): Ericsson Device Connection Platform. Hrsg.: Ericsson. https://www.ericsson.com/ru/ourportfolio/industries-solutions/device-connection-platform?nav=solutioncategory041%7Cf gb_101_0841 (zuletzt geprüft am: 30.03.2018).
- ERWIN, T.; HEIDKAMP, P.; POLS, A.; GRIMM, F.: Mit Daten Werte schaffen. Report 2016. <https://cdn2.hubspot.net/hubfs/571339/LandingPages-PDF/kpmg-mdws-201-sec.pdf> (zuletzt geprüft am: 30.03.2018).
- ERWIN, T.; HEIDKAMP, P.; POLS, A.: Mit Daten Werte schaffen. Report 2015. <https://www.bitkom.org/noindex//2015/Studien/Bitkom-KPMG-Bitkom-Research-Studie/KPMG-Bitkom-Research-Studie-MDWS-final-2.pdf> (zuletzt geprüft am: 30.03.2018).

- EU: Datenschutz-Grundverordnung (DSGVO). In: Amtsblatt EU L 59 (2016) 119 27.04.2016, S. 1-88. <http://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32016R0679&rid=1> (zuletzt geprüft am: 20.01.2018).
- EUROPÄISCHE KOMMISSION (Hrsg.): Empfehlung der Kommission vom 6. Mai 2003 betreffend die Definition der Kleinstunternehmen sowie der kleinen und mittleren Unternehmen (Text von Bedeutung für den EWR) (Bekannt gegeben unter Aktenzeichen K(2003) 1422). <http://eur-lex.europa.eu/legal-content/DE/TXT/?uri=celex:32003H0361> (zuletzt geprüft am: 30.03.2018).
- EUROPÄISCHE KOMMISSION (Hrsg.): Metadata: Statistische Systematik der Wirtschaftszweige in der Europäischen Gemeinschaft, Rev. 2. Luxemburg 2008a. http://ec.europa.eu/eurostat/ramon/nomenclatures/index.cfm?TargetUrl=LST_NOM_DTL&StrNom=NACE_REV2&StrLanguageCode=DE (zuletzt geprüft am: 04.05.2017).
- EUROPÄISCHE KOMMISSION (Hrsg.): NACE Rev. 2. Statistische Systematik der Wirtschaftszweige in der Europäischen Gemeinschaft Luxemburg 2008b. <http://ec.europa.eu/eurostat/documents/3859598/5902453/KS-RA-07-015-DE.PDF/680c5819-8a93-4c18-bea6-2e802379df86?version=1.0> (zuletzt geprüft am: 30.03.2018).
- EWALD, D.: Ergebnisse der Umstellung der Klassifikation im produzierenden Gewerbe. In: Statistisches Monatsheft(1996)10, S. 34-40.
- FEHLMANN, R.: Wie managen wir die steigende IT-Komplexität? In: Organisator(2006)4, S. 36-37.
- FERSTL, O. K.; SINZ, E. J.: Grundlagen der Wirtschaftsinformatik. 7. aktualis. Auflage. Oldenbourg, München [u. a.] 2013.
- FETTKE, P.: Architektur integrierter Informationssysteme. Enzyklopädie der Wirtschaftsinformatik. Online Lexikon. <http://www.encyklopaedie-der-wirtschaftsinformatik.de/wi-encyklopaedie/lexikon/daten-wissen/Informationsmanagement/Business-Engineering/-Business-Engineering--Ansätze-des/Architektur-integrierter-Informationssysteme-/index.html/?searchterm=aris> (zuletzt geprüft am: 30.03.2018).
- FETTKE, P.; LOOS, P.: Entwicklung eines Bezugsrahmens zur Evaluierung von Referenzmodellen - Langfassung eines Beitrages. Working papers of the Research Group Information Systems & Management; Paper 20. Johannes Gutenberg-Universität Mainz August 2004. http://www.uni-saarland.de/fileadmin/user_upload/Professoren/fr13_ProfLoos/isym_paper_020.pdf (zuletzt geprüft am: 26.04.2017).
- FIEDLER, A.; REGENHARD, U.: Mit CIM in die Fabrik der Zukunft? Probleme und Erfahrungen. Sozialverträgliche Technikgestaltung/ Materialien und Berichte; Bd. 17. Westdt. Verl., Opladen 1991.

- FIELDING, R. T.: Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine, University of California, Diss. 2000. - Zugl.: <http://jpkc.fudan.edu.cn/picture/article/216/35/4b/22598d594e3d93239700ce79bce1/7ed3ec2a-03c2-49cb-8bf8-5a90ea42f523.pdf> (zuletzt geprüft am: 23.03.2018).
- FINK, A.; SCHNEIDERREIT, G.; VOß, S.: Grundlagen der Wirtschaftsinformatik. 2., überarb. Auflage. Physica, Heidelberg 2005.
- FISHER, A.G.B.: The clash of progress and security. Macmillan, London 1935.
- FLEIß, J.: Paul Lazarsfelds typologische Methode und die Grounded Theory. Generierung und Qualität von Typologien. In: Österreichische Zeitschrift für Soziologie 35(2010)3, S. 3-18.
- FLUHR, J.: Beschreibung und Gestaltung eines Informationssystems zur Integration von Elektrofahrzeugen in das Stromversorgungssystem. Schriftenreihe Rationalisierung; Bd. 131. Hrsg.: G. Schuh. Apprimus, Aachen 2014. - Zugl.: Aachen, Techn. Hochsch., Diss., 2014.
- FOTH, E.: Erfolgsfaktoren für eine digitale Zukunft. IT-Management in Zeiten der Digitalisierung und Industrie 4.0. Springer, Berlin [u.a.] 2016.
- FRANK, U.; KLEIN, S.; KRUMHOLTZ, H.; TEUBNER, A.: Aktionsforschung in der WI-Einsatzpotentiale und Einsatzprobleme. In: Wirtschaftsinformatik und Wirtschaftstheorie. Grundpositionen und Theoriekerne. Arbeitsberichte des Instituts für Produktion und Industrielles Informationsmanagement; Nr. 4. Hrsg.: R. Schütte; J. Siedentopf; S. Zelewski. Institut für Produktion und Industrielles Informationsmanagement, Universität Essen, Essen 1998, S. 71-90.
- FRANKEN, M.: Produktionsplanung und -steuerung in strategischen Netzen. Ein logistikorientierter Koordinationsansatz. Dt. Univ.-Verl., Wiesbaden 2004. - Zugl.: Duisburg, Essen, Univ., Diss., 2003.
- FRECHT, N.: IT und Logistik: Alles im (Info)-Fluss. In: VDMA-Nachrichten(2013)2, S. 52-58.
- FRESE, E.; GRAUMANN, M.; THEUVSEN, L.: Grundlagen der Organisation. Entscheidungsorientiertes Konzept der Organisationsgestaltung. 10. überarb. u. erw. Auflage. Gabler, Wiesbaden 2012.
- FREY, J.: ASaP - Integrationsplattform für Smart Services in Intelligenten Umgebungen. Saarbrücken 2015. - Zugl.: Saarbrücken, Univ., Diss., 2015.
- FRIEDENTHAL, S.; MOORE, A.; STEINER, R.: A practical guide to SysML. The systems modeling language. Third edition. Morgan Kaufmann, Waltham (MA) 2015.
- FRIEDLI, T.: Die Architektur von Kooperationen. Dissertation; Nr. 2407. Difo-Druck, Bamberg 2000. - Zugl.: St. Gallen, Univ., Diss.
- FRIEDRICHS, J.: Methoden empirischer Sozialforschung. 9. Auflage. WV-Studium; Bd. 28: Sozialwiss. Westdt. Verl., Opladen 1981.

- FRIEDRICHSEN, U.: TOGAF in der Praxis. 26.12.2008. codecentric-Blog. <https://blog.codecentric.de/2008/12/togaf-in-der-praxistogaf-applied/> (zuletzt geprüft am: 30.03.2018).
- FROHMÜLLER, K.-P.; KIEFER, T.: IT-Komplexität: Ursachen und Beherrschungsmechanismen. In: Die Bank(1999)12, S. 832-837.
- GABRIEL, R.: Anwendungssystem. In: Enzyklopädie der Wirtschaftsinformatik - Online-Lexikon. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/uebergreifendes/Kontext-und-Grundlagen/Anwendungssystem/index.html?searchterm=anwendungssystem> (zuletzt geprüft am: 30.03.2018).
- GABRIEL, R.; GLUCHOWSKI, P.; PASTWA, A.: Datawarehouse & Data Mining. W3L, Herdecke [u. a.] 2009.
- GADATSCH, A.: IT-Controlling für Einsteiger. Praxiserprobte Methoden und Werkzeuge. Springer, Wiesbaden [u. a.] 2016.
- GANEK, A. G.; HILKNER, C. P.; SWEITZER, J. W.; MILLER, B.; HELLERSTEIN, J. L.: The response to IT complexity: autonomic computing. In: Proceedings of the Third IEEE International Symposium on Network Computing and Applications (NCA'04). Hrsg.: IEEE Computer Society. o. A. 2004, S. 151-157.
- GARTNER (Hrsg.): [Definition] Citizen Developer. Gartner IT Glossary 2017. <http://www.gartner.com/it-glossary/citizen-developer/> (zuletzt geprüft am: 05.04.2018). [=2017a]
- GARTNER (Hrsg.): [Definition] Operational Technology (OT). Gartner IT Glossary 2017b. <http://www.gartner.com/it-glossary/operational-technology-ot/> (zuletzt geprüft am: 06.04.2017). [=2017b]
- GARTNER (Hrsg.): [Pressemitteilung] Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016. Egham (UK) 07.02.2017. <http://www.gartner.com/newsroom/id/3598917> (zuletzt geprüft am: 30.03.2018). [=2017c]
- GARTNER (Hrsg.): Umsatz ausgewählter Segmente im Bereich Business-Software weltweit von 2010 bis 2012 und Prognose bis 2017 (in Milliarden US-Dollar) 2017. <https://de.statista.com/statistik/daten/studie/262351/umfrage/umsatz-ausgewaehlter-segmente-im-bereich-business-software-weltweit/>. [=2017d]
- GAYNOR, G. H.: Solving Problems. In: IEEE Engineering Management Review 44(2016)4, S. 6-8.
- GE DIGITAL (Hrsg.): GE Predix Architecture and Services 28.11.2016. <http://go.digital.ge.com/Predix-Architecture-WP.html> (zuletzt geprüft am: 26.04.2018).
- GE DIGITAL (Hrsg.): Predix Platform. Combining assets, data, analytics, and applications to transform industrial operations. <https://www.ge.com/digital/predix/platform> (zuletzt geprüft am: 30.03.2018).

- GIERSBERG, G.: Jack MA ist der große Gewinner. In: faz.net, 05.07.2017. <http://www.faz.net/aktuell/wirtschaft/unternehmen/apple-google-und-alibaba-sind-besonders-wertvoll-15091321.html> (zuletzt geprüft am: 30.03.2018).
- GOETHE, J. W. v.: Zur Naturwissenschaft überhaupt, besonders zur Morphologie. Erfahrung, Betrachtung, Folgerung, durch Lebensereignisse verbunden. Zur Morphologie; Bd. 2. Stuttgart [u. a.] 1817. Digitalisat über Google Books, Original von der Bayerischen Staatsbibliothek, digitalisiert am 20. Mai 2014. <https://books.google.de/books?id=0Zx0U-IsqCcC> (zuletzt geprüft: 17.04.2018).
- GÖTZ, M.; MENDEL, C.; BULLA, C.: "I always wanted to be a Pokémon master". Die Faszination der App Pokémon Go. In: Communicatio Socialis 50(2017)2, S. 251-263.
- GRAUMANN, M.: Was ist eine Theorie? In: WiSt - Wirtschaftswissenschaftliches Studium 44(2015)9, S. 528-532.
- GREBE, M.; DANKE, E.: Simplify IT. Six Ways to reduce Complexity. Hrsg.: BCG. März 2013. http://image-src.bcg.com/Images/BCG_Simplify_IT_Mar_2013_tcm9-96858.pdf (zuletzt geprüft am: 30.03.2018).
- GREBE, M.; DÖSCHL, S.; SCHMID, C.; KOOPMANS, J.: Simplifying IT to Accelerate Digital Transformation. April 2016. <https://www.bcgperspectives.com/content/articles/technology-digital-simplifying-it-accelerate-digital-transformation/> (zuletzt geprüft am: 18.05.2017).
- GREEN, J.: [Whitepaper] The Internet of Things Reference Model. Hrsg.: CiscoChicago 04.06.2014. http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf (zuletzt geprüft am: 30.03.2018).
- GRONAU, N.; THIM, C.; FOHRHOLZ, C.: B2B-Studie 2017: Wettbewerbsfaktor Analytics im Internet der Dinge. Eine Studie des Lehrstuhls für Wirtschaftsinformatik insb. Prozesse und Systeme der Universität Potsdam in Zusammenarbeit mit SAS Institute GmbH. Potsdam 2016. http://docs.media.bitpipe.com/io_13x/io_134635/item_1448895/B2B-Studie%202017%20-%20Wettbewerbsfaktor%20Analytics%20im%20Internet%20der%20Dinge.pdf (zuletzt geprüft am: 30.03.2018).
- GROß, H.: BPI Fertigung 2015 D/A/CH. Fachbereichsstudie IT-OrganisationKassel 2016. <https://www.business-performance-index.de/file/get/wE8DyXjLiNd8hREt> (zuletzt geprüft am: 25.04.2017).
- GRUBER, M.; HOFFMANN, J.; KARLA, J.: Methoden zum Management der IT-Komplexität. Eine strukturierte Literaturanalyse. In: Lecture Notes in Informatics (LNI). Proceedings. Hrsg.: M. Eibl; M. Gaedke. Ges. für Informatik, Bonn 2017, S. 1601-1615.
- GUDGIN, M.; HADLEY, M.; MENDELSON, N.; MOREAU, J.-J.; NIELSEN, H. F.; KARMARKAR, A.; LAFON, Y.: SOAP Version 1.2 Part 1. Messaging Framework. Hrsg.: W3C. <https://www.w3.org/TR/soap12/> (zuletzt geprüft am: 30.03.2018).
- GUTENBERG, E.: Grundlagen der Betriebswirtschaftslehre. 2. Band: Der Absatz. 16. Auflage. Springer, Berlin [u.a.] 1979.

- GUTH, J.; BREITENBUCHER, U.; FALKENTHAL, M.; LEYMAN, F.; REINFURT, L.: Comparison of IoT platform architectures: A field study based on a reference architecture. In: 2016 Cloudification of the Internet of Things (CIoT). Hrsg.: DNAC. IEEE, 2016, S. 1-6.
- HAAG, S.: Industrial Internet Reference Architecture, Technische Universität Darmstadt, Fachgebiet Datenverarbeitung in der Konstruktion, Darmstadt 05.10.2015.
- HABERFELLNER, R.; DE WECK, O.; FRICKE, E.; VÖSSNER, S. (Hrsg.): Systems Engineering. Grundlagen und Anwendung. 13., aktualisierte Auflage. Orell Füssli, Zürich 2015.
- HANKEL, M.: Das Referenzarchitekturmodell Industrie 4.0 (RAMI4.0). Hrsg.: ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie e.V. Frankfurt am Main 2015. http://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/zvei-faktenblatt-rami.pdf?__blob=publicationFile&v=3 (zuletzt geprüft am: 30.03.2018).
- HANSCHKE, I.: Agile in der Unternehmenspraxis. Fallstricke erkennen und vermeiden, Potenziale heben. Springer Vieweg, Wiesbaden 2017.
- HANSCHKE, I.: Enterprise Architecture Management - einfach und effektiv. Ein praktischer Leitfaden für die Einführung von EAM. 2., überarb. Auflage. Hanser, München [u. a.] 2016.
- HARTIG, O.; PÉREZ, J.: An Initial Analysis of Facebook's GraphQL Language. In: Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web. Hrsg.: J. Reutter; D. Srivastava. CEUR Workshop Proceedings; 1912. Sun SITE, Aachen 2017, S. 11. <http://ceur-ws.org/Vol-1912/paper11.pdf>.
- HE, W.; XU, L. D.: Integration of Distributed Enterprise Applications. A Survey. In: IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS 10(2014)1, S. 35-42.
- HEES, A. F.: System zur Produktionsplanung für rekonfigurierbare Produktionssysteme. Utz, München 2017. - Zugl.: München, Techn., Univ., Diss., 2017.
- HEIDRICH, M.; JIJUN LUO, J.: [Whitepaper] Industrial Internet Of Things: Referenzarchitektur für die Kommunikation. Hrsg.: Fraunhofer-Institut für eingebettete Systeme und Kommunikation ESKMünchen 2016. https://www.esk.fraunhofer.de/content/dam/esk/dokumente/Whitepaper_IoT_dt_April16.pdf (zuletzt geprüft am: 30.03.2018).
- HEIMANN, T.; PRÄDEL, J.-M.; EMENAKO, D.: Studie IT-Trends 2017. Überfordert Digitalisierung etablierte Unternehmensstrukturen? Hrsg.: CapgeminiBerlin 2017. <https://www.capgemini.com/de-de/wp-content/uploads/sites/5/2017/02/it-trends-studie-2017> (zuletzt geprüft am: 29.03.2018).
- HEIMANN, T.; PRÄDEL, J.-M.; EMENAKO, D.: Studie IT-Trends 2018. Digitalisierung: Aus Ideen werden Ergebnisse. Hrsg.: CapgeminiBerlin 2018. <https://www.capgemini.com/de-de/wp-content/uploads/sites/5/2018/02/it-trends-studie-2018.pdf> (zuletzt geprüft am: 06.04.2018).

- HEINRICH, L. J.; RIEDL, R.; HEINZL, A.: Wirtschaftsinformatik. Einführung und Grundlegung. 4. überarb. u. erw. Auflage. Springer, Berlin [u.a.] 2011.
- HEINRICH, L. J.; STELZER, D.; RIEDL, R.: Informationsmanagement. Grundlagen, Aufgaben, Methoden. 11. vollst. überarb. Auflage. DE Gruyter Oldenbourg, Berlin [u. a.] 2014.
- HERRMANN, A.; HUBER, F.: Produktmanagement. Grundlagen - Methoden - Beispiele. 2., vollst. überarb. u. erw. Auflage. Gabler, Wiesbaden 2008.
- HESS, T.; MATT, C.; HILBERS, K.: Bekannte und weniger bekannte Wege zu praxisrelevanter Forschung in der Wirtschaftsinformatik. In: Wirtschaftsinformatik in Wissenschaft und Praxis. Festschrift für Hubert Österle. Hrsg.: W. Brenner; T. Hess. Springer-Gabler, Berlin [u. a.] 2014, S. 129-139.
- HESSE, W.; MAYR, H. C.: Modellierung in der Softwaretechnik. Eine Bestandsaufnahme. In: Informatik-Spektrum 31(2008)5, S. 377-393.
- HILDEBRAND, K.; GEBAUER, M.; HINRICHS, H.; MIELKE, M. (Hrsg.): Daten- und Informationsqualität. Auf dem Weg zur Information Excellence. 3., erw. Auflage. Springer, Wiesbaden [u. a.] 2015.
- HILL, W.; FEHLBAUM, R.; ULRICH, P.: Organisationslehre 1: Ziele, Instrumente und Bedingungen der Organisation sozialer Systeme. 5. Auflage. Uni-Taschenbücher; Haupt, Bern [u. a.] 1994.
- HIPPNER, H.; HUBRICH, B.; WILDE, K. D. (Hrsg.): Grundlagen des CRM. Strategie, Geschäftsprozesse und IT-Unterstützung. 3., vollst. überarb. und erw. Aufl. Gabler, Wiesbaden 2011.
- HOFFJAN, A.; GANTENBRINK, J.: Das Controlling wetterfest machen. In: Controlling 24(2012)3, S. 136-137.
- HOFFMANN, J.; HEIMES, P.: Informationssystem-Architekturen produzierender Unternehmen für die Digitalisierung gestalten. In: HMD Praxis der Wirtschaftsinformatik 55(2018)6, S. 984-1005. <https://link.springer.com/article/10.1365/s40702-018-0440-8>.
- HOFFMANN, J.; HEIMES, P.; SENEL, S.: IoT Platforms for the Internet of Production. In: IEEE Internet of Things Journal(2018), S. 1-8.
- HOFMANN, G. R.: Mobile vs. Ubiquitous vs. Portable Computing. Einige Bemerkungen zu den Begriffen - und zur Case-based Evidence einer Daten-Bank. In: Mobile Anwendungen in Unternehmen. Konzepte und betriebliche Einsatzszenarien. Hrsg.: T. Barton; C. Müller; C. Seel. Springer, Wiesbaden [u. a.] 2016, S. 9-26.
- HOLZ, T.; POHLMANN, N.; BODDEN, E.; SMITH, M.; HOFFMANN, J.: Human-Centered Systems Security. IT-Sicherheit von Menschen für Menschen. http://www.it-sicherheit-nrw.de/download/Strategiepapier_IT-Sicherheit-von-Menschen-fuer-Menschen.pdf (zuletzt geprüft am: 30.03.2018).

- HOPPE, S.: Standardisierte horizontale und vertikale Kommunikation. In: Handbuch Industrie 4.0; Bd. 2: Automatisierung. Hrsg.: B. Vogel-Heuser; T. Bauernhansl; M. ten Hompel. 2. Auflage; 2. Springer Vieweg, Berlin 2016, S. 371-390.
- HOUY, C.; FRANK, J.; NIESEN, T.; FETTKE, P.; LOOS, P.: Zur Verwendung von Theorien in der Wirtschaftsinformatik. Eine quantitative Literaturanalyse. In: Veröffentlichungen des Instituts für Wirtschaftsinformatik(2014)198, S. 1-82.
- HUMBY, C.: Data is the New Oil. http://ana.blogs.com/maestros/2006/11/data_is_the_new.html (zuletzt geprüft am: 10.08.2017).
- IIDA, I.: Hyperconnected Cloud to Embrace Megatrends in ICT. In: Fujitso Scientific & Technical Journal 52(2016)1, S. 3-8.
- INDUSTRIAL INTERNET CONSORTIUM (Hrsg.): Industrial Internet Reference Architecture. 04.06.2015. <http://www.iiconsortium.org/IIRA-1-7-ajs.pdf> (zuletzt geprüft am: 30.03.2018).
- INDUSTRIAL INTERNET CONSORTIUM (Hrsg.): The Industrial Internet of Things. Volume G1: Reference Architecture. 31.01.2017. http://www.iiconsortium.org/IIC_PUB_G1_V1.80_2017-01-31.pdf (zuletzt geprüft am: 30.03.2018).
- INFORMATICA (Hrsg.): Integration Platform-as-a-Service (iPaaS). Visuelle Tools für Entwickler und Architekten. Assistenten mit Self-Service für Business User. 2017. <https://www.informatica.com/de/products/integration-platform-as-a-service.html> (zuletzt geprüft am: 30.03.2018).
- ISO 15704: Industrial automation systems - Requirements for enterprise-reference architectures and methodologies. ISO (Hrsg.); ICS 25.040.01, 35.240.50. ISO, Genf, 15.08.2005.
- ISO/IEC 2382: Information technology - Vocabulary. ISO/IEC (Hrsg.); ICS 01.040.35, 35.020. Beuth, Mai 2015.
- ISO/IEC 7498-1: Information technology - Open Systems Interconnection - Basic reference model: The basic model. ISO/IEC (Hrsg.); ICS 35.100.01. Beuth Verlag, Berlin, 01.11.1994.
- ISO/IEC/IEEE FDIS 12207: Systems and software engineering - Software life cycle processes. ISO/IEC; IEEE COMPUTER SOCIETY (Hrsg.); ICS 35.080. IEEE, Genf, Juni 2017.
- JINUSHI, T.; CHISAKI, K.; KAWAKAMI, Y.: Optimization of Factory Production Activities by Utilizing IoT. In: Fujitso Scientific & Technical Journal 52(2016)4, S. 77-83.
- JOSEY, A.: TOGAFreg; Version 9.1 A Pocket Guide. Van Haren Publishing, Zaltbommel 2011.
- JOUANNE-DIEDRICH, H. v.: ephorie.de IT-Sourcing-Map. Die Orientierungshilfe im stetig wachsenden Dschungel der Outsourcing-Konzepte. <http://www.ephorie.de/it-sourcing-map.htm#Zitierung%20ephorie.de%20IT-Sourcing-Map> (zuletzt geprüft am: 30.03.2018).

- JOUANNE-DIEDRICH, H. V.; ZARNEKOW, R.; BRENNER, W.: Industrialisierung des IT-Sourcings. In: HMD Praxis der Wirtschaftsinformatik 42(2005)245, S. 18-27.
- JUNG, H.: Allgemeine Betriebswirtschaftslehre. De Gruyter, Berlin [u. a.] 2016.
- KAGERMANN, H.: Chancen von Industrie 4.0 nutzen. In: Handbuch Industrie 4.0; Bd. 4: Allgemeine Grundlagen. Hrsg.: B. Vogel-Heuser; T. Bauernhansl; M. ten Hompel. 2. Auflage. Springer, Berlin [u. a.] 2017, S. 235-246.
- KAIB, M.: Enterprise Application Integration: Grundlagen, Integrationsprodukte, Anwendungsbeispiele. Springer, Berlin [u. a.] 2002.
- KAMPKER, A.; SCHUH, G.; SCHITTNY, B.: Unternehmensstruktur. In: Strategie und Management produzierender Unternehmen. Hrsg.: G. Schuh; A. Kampker. Handbuch Produktion und Management; Bd. 1. Springer, Berlin [u. a.] 2011, S. 133-230.
- KART, L.; LINDEN, A.; SCHULTE, W. R.: Extend your portfolio of analytics capabilities. 23.09.2013. <https://www.gartner.com/doc/2594822/extend-portfolio-analytics-capabilities> (zuletzt geprüft am: 30.03.2018).
- KEES, A.: Open Source Enterprise Software. Grundlagen, Praxistauglichkeit und Marktübersicht quelloffener ERP-Systeme. Springer Vieweg, Wiesbaden 2015.
- KEMPER, H.-G.; FINGER, R.: Transformation operativer Daten. In: Analytische Informationssysteme. Business Intelligence-Technologien und -Anwendungen. Hrsg.: P. Gluchowski; P. Chamoni. 5., vollst. überarb. Auflage. Springer, Berlin [u. a.] 2016, S. 129-145.
- KERLINGER, F. N.; LEE, H. B.: Foundations of behavioral research. 4. Auflage. Harcourt College Publishers, Fort Worth 2000.
- KERTH, K.; ASUM, H.; STICH, V.: Die besten Strategietools in der Praxis. Welche Werkzeuge brauche ich wann? Wie wende ich sie an? Wo liegen die Grenzen? 6., erw. Auflage. Hanser, München 2015.
- KEUNTJE, J. H.; BARKOW, R. (Hrsg.): Enterprise Architecture Management in der Praxis. Wandel, Komplexität und IT-Kosten im Unternehmen beherrschen. Symposium, Düsseldorf 2010.
- KEUNTJE, J. H.: EAM Content. In: Enterprise Architecture Management in der Praxis. Wandel, Komplexität und IT-Kosten im Unternehmen beherrschen. Hrsg.: J. H. Keuntje; R. Barkow. Symposium, Düsseldorf 2010, S. 143-187.
- KISTNER, K.-P.; STEVEN, M.: Produktionsplanung. Integrierte Produktentwicklung. Denkabläufe, Methodeneinsatz, Zusammenarbeit. 3., vollst. überarb. Auflage. Physica-Verlag, Heidelberg 2001.
- KLEINALTENKAMP, M.; SAAB, S.: Technischer Vertrieb. Eine praxisorientierte Einführung in das Business-to-Business-Marketing. Springer, Berlin [u. a.] 2009.

- KLEIST, H. v.: Die Schlacht bei Fehrbellin. Schauspiel in fünf Akten. 1. Auflage. Wallishäuser; Reimer, Berlin 1822.
- KLETTI, J.: MES als Werkzeug für die perfekte Produktion. In: MES - Manufacturing Execution System. Moderne Informationstechnologie unterstützt die Wertschöpfung. Hrsg.: J. Kletti. 2. Auflage. Springer Vieweg, Berlin [u. a.] 2015, S. 19-30.
- KLOCKE, F.; BASSETT, E.; BÖNSCH, C.; GÄRTNER, R.; HOLSTEN, S.; JAMAL, R.; JURKE, B.; KAMPS, S.; KERZEL, U.; MATTFELD, P.; SHIROBOKOV, A.; STAUDER, J.; STAUTNER, M.; TRAUTH, D.: Assistenzsysteme in der Produktionstechnik. In: AWK Aachener Werkzeugmaschinen-Kolloquium 2017: Internet of Production für agile Unternehmen. Hrsg.: C. Brecher; F. Klocke; R. Schmitt; G. Schuh. Apprimus, Aachen 2017, S. 287-313.
- KLOCKE, F.; BERGS, T.; BOBEK, T.; HUWER, T.; LIU, G.; POTHEN, M.; STAASMEYER, J.-H.; WINANDS, K.: Vernetzte adaptive Produktion. In: AWK Aachener Werkzeugmaschinen-Kolloquium 2017: Internet of Production für agile Unternehmen. Hrsg.: C. Brecher; F. Klocke; R. Schmitt; G. Schuh. Apprimus, Aachen 2017, S. 263-285.
- KNOBLICH, H.: Betriebswirtschaftliche Warentypologie. Grundlagen u. Anwendungen. Westdt. Verl., Köln [u. a.] 1969.
- KÖNIGS, H.-P.: IT-Risikomanagement mit System. 5. überarb. u. erw. Auflage. Springer, Wiesbaden [u. a.] 2017.
- KOPPER, A.; WESTNER, M.; STRAHRINGER, S.: Kontrollierte Nutzung von Schatten-IT. In: HMD - Praxis der Wirtschaftsinformatik 54(2017)1, S. 97-110.
- KRAUSE, T.; STRAUß, O.; SCHEFFLER, G.; KETT, H.; LEHMANN, K.; RENNER, T.: IT-Plattform für das Internet der Dinge (IoT). Basis intelligenter Produkte und Services. Fraunhofer Verlag, Stuttgart 2017.
- KRCMAR, H.: Bedeutung und Ziele von Informationssystem-Architekturen. In: Wirtschaftsinformatik 32(1990)5, S. 395-402.
- KRCMAR, H.: Informationsmanagement. 6., überarb. Auflage. Springer, Berlin [u. a.] 2015.
- KRESSE, M.; BAUSE, M. (Hrsg.): ITIL, alles was man wissen muss. Edition 2011. 2. Auflage. Serview GmbH, Bad Homburg v. d. H. 2012.
- KRUCHTEN, P.: The 4+1 View Model of architecture. In: IEEE Software 12(1995)6, S. 42-50.
- KRÜGER, J.; VICK, A.; CHEMNITZ, M.; ROSENSTRAUCH, M.; HÜGLE, J.; FECHTELER, M.; BLANKENBUERG, M.: Daten, Information und Wissen in Industrie 4.0. In: Handbuch Industrie 4.0: Geschäftsmodelle, Prozesse, Technik. Hrsg.: G. Reinhart. Hanser, München [u. a.] 2017, S. 89-110.

- KRUMM, S.; SCHITTNY, S. U.; RUDOLF, S.; SCHREY, E.: Lean innovation (Teil 4). Schwerpunkt: Datenkonsistenz "Single source of truth". In: Complexity Management Journal(2015)2, S. 13-16.
- KUBACH, U.: Device Clouds: Cloud-Plattformen schlagen die Brücke zwischen Industrie 4.0 und dem Internet der Dinge. In: Handbuch Industrie 4.0; Bd. 3: Logistik. Hrsg.: B. Vogel-Heuser; T. Bauernhansl; M. ten Hompel. 2. Auflage. Springer Vieweg, Berlin 2016, S. 181-200.
- KUBER, P.; RUSSEL, B.; SUNDARAM, S.: Future-proofing the Connected World: 13 Steps to Developing Secure IoT Products. 2016. <https://downloads.cloudsecurityalliance.org/assets/research/internet-of-things/future-proofing-the-connected-world.pdf> (zuletzt geprüft am: 30.03.2018).
- KUBICEK, H.: Heuristische Bezugsrahmen und heuristisch angelegtes Forschungsdesign als Elemente einer Konstruktionsstrategie empirischer Forschung. In: Empirische und handlungstheoretische Forschungskonzeptionen in der Betriebswirtschaftslehre. Hrsg.: R. Köhler. Poeschel, Stuttgart 1977, S. 3-36.
- KUHN, T.: Digitaler Zwilling. In: Informatik-Spektrum 40(2017)5, S. 440-444.
- KUMAR, K. B. S.; Srividya; MOHANAVALI, S.: A performance comparison of document oriented NoSQL databases. In: Proceedings ICCSP 2017. Hrsg.: S. Nadar; R. Srinivasan; K. Vijayakumar; S. Salivahanan; T. Nagarajan. IEEE, Piscataway (NJ) 2017, S. 71-76.
- LACKES, R.: [Definition]Architektur integrierter Informationssysteme. <http://wirtschaftslexikon.gabler.de/Definition/architektur-integrierter-informationssysteme.html?referenceKeywordName=ARIS> (zuletzt geprüft am: 01.04.2018).
- LANGLINAIS, T. C.; FORBES, S.: Automakers Sketch Out a Road Map for Banks. In: American Banker 171(2006)20, 8A.
- LARA, P.; SÁNCHEZ, M.; VILLALOBOS, J.: Bridging the IT and OT Worlds Using an Extensible Modeling Language. In: Conceptual Modeling: 35th International Conference, ER 2016, Gifu, Japan, November 14-17, 2016, Proceedings. Hrsg.: I. Comyn-Wattiau; K. Tanaka; I.-Y. Song; S. Yamamoto; M. Saeki. Springer, Cham [u. a.] 2016, S. 122-129.
- LARSON, P.-Å.; LEVANDOSKI, J.: Modern main-memory database systems. In: Proceedings of the VLDB Endowment 9(2016)13, S. 1609-1610.
- LEHNER, F.; WILDNER, S.; SCHOLZ, M.: Wirtschaftsinformatik. Eine Einführung. 2. Auflage. Hanser, München [u. a.] 2008.
- LEIMEISTER, J. M.: Einführung in die Wirtschaftsinformatik. 12. vollst. neu überarb. Auflage. Springer Gabler, Wiesbaden [u. a.] 2015.
- LEMBERGER, P.; MOREL, M.: Managing complexity of information systems. The value of simplicity. ISTE; Wiley, London [u. a.] 2012.

- LEMKE, C.; BRENNER, W.; KIRCHNER, K.: Gestalten des digitalen Zeitalters. Springer Gabler, Berlin [u. a.] 2017.
- LEMKE, C.; BRENNER, W.: Verstehen des digitalen Zeitalters. Springer Gabler, Berlin [u. a.] 2015.
- ŁENT, B.: IT-Projektmanagement als kybernetisches System. Intelligente Entscheidungsfindung in der Projektführung durch Feedback. Springer, Wiesbaden [u. a.] 2013.
- LEYH, C.: Implementierung von ERP-Systemen in KMU – Ein Vorgehensmodell auf Basis von kritischen Erfolgsfaktoren. In: HMD - Praxis der Wirtschaftsinformatik 52(2015)3, S. 418-432.
- LI, C.-S.; BRECH, B. L.; CROWDER, S.; DIAS, D. M.; FRANKE, H.; HOGSTROM, M.; LINDQUIST, D.; PACIFICI, G.; PAPPE, S.; RAJARAMAN, B.; RAO, J.; RATNAPARKHI, R. P.; SMITH, R. A.; WILLIAMS, M. D.: Software defined environments. An introduction. In: IBM Journal of Research and Development 58(2014)2/3, 1:1-1:11.
- LI, C.-S.; LIAO, W.: Software defined networks [Guest Editorial]. In: IEEE Communications Magazine 51(2013)2, S. 113.
- LIEBER, D.; EROHIN, O.; DEUSE, J.: Wissensentdeckung im industriellen Kontext. Herausforderungen und Anwendungsbeispiele. In: ZWF - Zeitschrift für wirtschaftlichen Fabrikbetrieb 108(2013)6, S. 388-393.
- LUHMANN, N.: Komplexität. In: Enzyklopädie der Betriebswirtschaftslehre. Bd. 2: Handwörterbuch der Organisation. Hrsg.: E. Grochla. 2., völlig neu gest. Auflage; 2. Poeschel, Stuttgart 1980, S. 1064-1070.
- MALIK, F. F.: Führen. leisten. leben: Wirksames Management für eine neue Zeit. Campus, Frankfurt [u. a.] 2006.
- MALONE, T. W.; CROWSTON, K.: The interdisciplinary study of coordination. In: ACM Computing Surveys 26(1994)1, S. 87-119.
- MANZ, O.: Fehlerkorrigierende Codes. Springer Vieweg, Wiesbaden 2017.
- MARIOTTI, J. L.: The Complexity crisis: Why too many products, markets and customers are crippling your company - and what to do about it. Adams Media, Avon (MA) 2008.
- MARTENS, A.: Ablösung von Legacy-Systemen in Zeiten des digitalen Wandels. In: Wirtschaftsinformatik & Management 8(2016)6, S. 32-40.
- MARZ, N.; WARREN, J.: Big data. Principles and best practices of scalable real-time data systems. Manning, Shelter Island 2015.
- MASAK, D.: Legacysoftware. Das lange Leben der Altsysteme. Springer, Berlin [u. a.] 2006.
- MATTHES, D.: Enterprise Architecture Frameworks Kompendium. Über 50 Rahmenwerke für das IT-Management. Springer, Berlin [u. a.] 2011.

- MAYER, K.: Nachhaltigkeit. 111 Fragen und Antworten. Springer, Wiesbaden [u. a.] 2017.
- MAYRING, P.: Einführung in die qualitative Sozialforschung. Eine Anleitung zu qualitativem Denken. Beltz, Weinheim [u. a.] 2002.
- McFARLAN, F.W.; MCKENNEY, J. L.; PYBURN, P.: The information archipelago-plotting a course. In: Harvard Business Review 61(1983)1, S. 145-156.
- MCNIFF, J.: Action research. All you need to know. 1. Auflage. SAGE Publications, Thousand Oaks CA 2017.
- MCNIFF, J.; WHITEHEAD, J.: All you need to know about action research. 2. Auflage. SAGE, Los Angeles 2011.
- MEIER, H.; UHLMANN, E. (Hrsg.): Industrielle Produkt-Service Systeme. Entwicklung, Betrieb und Management. Springer, Berlin [u. a.] 2017.
- MEYER, A.; WÜNSCH, A.; WIESNER, M.: Anwendungs- und Informationsintegration. In: Integrated Design Engineering. Hrsg.: S. Vajna. Springer, Berlin [u. a.] 2014, S. 411-446.
- MINERAUD, J.; MAZHELIS, O.; SU, X.; TARKOMA, S.: Contemporary Internet of Things platforms. 28.01.2015. <https://arxiv.org/abs/1501.07438> (zuletzt geprüft am: 30.03.2018).
- MINGUEZ, J.: Der Manufacturing Service Bus. In: Digitale Produktion. Hrsg.: E. Westkämper; D. Spath; C. Constantinescu; J. Lentz. Springer, Berlin [u.a.] 2013, S. 271-289.
- MOAWED, M.: Managing IT Complexity in the Enterprise 2015. <https://de.scribd.com/document/282792341/CIOinsight-Managing-IT-Complexity-in-the-Enterprise-Full-Report-1036266>.
- MOCKER, M.: What Is Complex About 273 Applications? Untangling Application Architecture Complexity in a Case of European Investment Banking. In: 42nd Hawaii International Conference on System Sciences, 2009. HICSS '09 ; Waikoloa, Hawaii, 5 - 8 Jan. 2009. Hrsg.: R. H. Sprague. IEEE, Piscataway (NJ) 2009, S. 1-14.
- MOMMSEN, D.; PORTMANN, E.: Gestaltungs- und praxisorientierte Promotionsarbeiten im Spannungsfeld zwischen Anwendung und Forschung. In: Wirtschaftsinformatik in Theorie und Praxis. Festschrift zu Ehren von Prof. Dr. Andreas Meier. Hrsg.: E. Portmann. Springer, Wiesbaden [u. a.] 2017, S. 11-34.
- MPDV MIKROLAB (Hrsg.): Manufacturing Integration Platform: Plattform für die Fertigungs-IT. IT & Production online, 26.05.2017. <https://www.it-production.com/news/plattform-fuer-die-fertigungs-it/> (zuletzt geprüft am: 31.03.2018).
- MÜLLER, B.: RWTH Aachen - Heimat der Streetscooter. In: Süddeutsche Zeitung(2017). <http://www.sueddeutsche.de/wirtschaft/rwth-aachen-heimat-der-streetscooter-1.3748607> (zuletzt geprüft am: 06.03.2018).

- MÜLLER, K.-R.: Handbuch Unternehmenssicherheit. Umfassendes Sicherheits-, Kontinuitäts- und Risikomanagement mit System. 3., aktualis. u. erw. Auflage. Springer, Wiesbaden [u. a.] 2015.
- MÜLLER, K.-R.; NEIDHÖFER, G.: IT für Manager. Mit geschäftszentrierter IT zu Innovation, Transparenz und Effizienz. 1. Auflage. Vieweg & Teubner, Wiesbaden 2012.
- MÜLLER, R.: Montagegerecht. In: Pahl/Beitz Konstruktionslehre. Hrsg.: J. Feldhusen; K.-H. Grote. 8. vollst. überarb. Auflage. Springer, Berlin [u. a.] 2013, S. 702-725.
- MÜLLER, S.: Manufacturing Execution Systeme (MES): Status quo und Ausblick in Richtung Industrie 4.0. Books on Demand, Norderstedt 2015.
- MURCH, R.: Managing Complexity in IT, Part1: The problem. informIT - The trusted technology learning source. Hrsg.: Pearson. 01.10.2004. <http://www.informit.com/articles/article.aspx?p=336860> (zuletzt geprüft am: 31.03.2018).
- NACHREINER, F.: Methoden empirischer Forschung in der Arbeitswissenschaft. In: Handbuch Arbeitswissenschaft. Hrsg.: H. Luczak; W. Volpert. Schäffer-Poeschel, Stuttgart 1997, S. 87-124.
- NAJAFABADI, M. M.; VILLANUSTRE, F.; KHOSHGOFTAAR, T. M.; SELIYA, N.; WALD, R.; MUHAREMAGIC, E.: Deep learning applications and challenges in big data analytics. In: Journal of Big Data 2(2015)1, S. 886.
- NETTSTRÄTER, A.: Internet-of-Things Architecture. IoT-A 07/2012. http://cocoa.ethz.ch/downloads/2014/01/1524_D1.3_Architectural_Reference_Model_update.pdf (zuletzt geprüft am: 16.04.2018).
- NEUMANN, S.: Position Paper. In: Proceedings Summary. Hrsg.: S. Widergren. Pacific Northwest National Laboratory, Richland (WA) 2007, S. 4-8.
- NIEHUES, M.; SCHUH, G.; REINHART, G.; SCHMITT, R.; BRAMBRING, F.; ELLERICH, M.; ELSE, H.; FRANK, D.; GROGGERT, S.; GÜTZLAFF, A.; HEINRICHS, V.; HEMPEL, T.; KOSTYSZYN, K.; NGO, H.; NIENDORF, L.; PERMIN, E.; PROTE, J.; REUTER, C.; TÜRDMANN, R.: Organisation, Qualität und IT-Systeme für Planung und Betrieb. In: Handbuch Industrie 4.0: Geschäftsmodelle, Prozesse, Technik. Hrsg.: G. Reinhart. Hanser, München [u. a.] 2017, S. 137-168.
- NIENKE, S.; FUHS, G.: [Whitepaper] Stammdatenmanagement. 2., neu bearb. Auflage. FIR e.V. an der RWTH Aachen, Aachen 2017.
- NIENKE, S.; ZELLER, V.; ZÖLLNER, O.; SCHUH, G.: Introducing an Information Logistics Approach to Support the Development of Future Energy Grid Management. In: Proceedings of the 20th International Conference on Enterprise Information Systems - (Volume 2). March 21 - 24, 2018. Hrsg.: J. Filipe; M. Smialek; O. Camp; S. Hammoudi. SciTePress, Funchal, S. 239-246. <http://www.scitepress.org/PublicationsDetail.aspx?ID=J/6+9Ya+vZo=&t=1> (zuletzt geprüft am: 31.03.2018).
- NIJIM, S.; PAGANO, B.: APIs For Dummies. John Wiley & Sons, New York 2014.

- NISSEN, V.; TERMER, F.; RENNENKAMPFF, A. von: Agile IT-Anwendungslandschaften als strategische Unternehmensressource. In: HMD - Praxis der Wirtschaftsinformatik 49(2012)2, S. 24-33.
- NOTH, T.: Stärkung der Wettbewerbsfähigkeit des Kooperationsverbundes. In: Zeitschrift für das gesamte Kreditwesen 58(2005)03, S. 12-14.
- NYHUIS, P.; HÜBNER, M.; QUIRICO, M.; SCHÄFER, P.; SCHMIDT, M.: Veränderung in der Produktionsplanung und -steuerung. In: Handbuch Industrie 4.0: Geschäftsmodelle, Prozesse, Technik. Hrsg.: G. Reinhart. Hanser, München [u. a.] 2017, S. 31-50.
- OEDEKOVEN, D.: Nutzenpotenziale harmonisierter Stammdaten in den Prozessen der Auftragsabwicklung von Auftragsfertigern. Schriftenreihe Rationalisierung; Bd. 110. RHrsg.: G. Schuh. Apprimus, Aachen 2011. - Zugl.: Aachen, Techn. Hochsch., Diss., 2011.
- OPC FOUNDATION (Hrsg.): OPC unified architecture. Wegbereiter der 4. industriellen (R)Evolution. <https://www.iosb.fraunhofer.de/servlet/is/21752/OPC-UA-Wegbereiter-der-I40.pdf?command=downloadContent&filename=OPC-UA-Wegbereiter-der-I40.pdf> (zuletzt geprüft am: 31.03.2018).
- O'SHEA, M.: Planungsverfahren für die Produktkonzeption. Ein systematisches Vorgehenskonzept unter Berücksichtigung des Lebenszyklus-Ansatzes. Betriebswirtschaftliche Forschung zur Unternehmensführung; Bd. 41. Dt. Univ.-Verl., Wiesbaden 2002. - Zugl.: Hamburg, Univ., Diss., 2001.
- OSTERHAGE, W. W.: IT-Kompendium. Die effiziente Gestaltung von Anwendungsplattformen. Springer, Berlin [u. a.] 2017.
- ÖSTERLE, H.; BECKER, J.; FRANK, U.; HESS, T.; KARAGIANNIS, D.; KRCMAR, H.; LOOS, P.; MERTENS, P.; OBERWEIS, A.; SINZ, E. J.: Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. In: Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung 62(2010)6, S. 664-672.
- OTTO, B.; ÖSTERLE, H.: Corporate Data Quality. Voraussetzung erfolgreicher Geschäftsmodelle. Springer, Berlin [u. a.] 2016.
- PAHL, G.; BEITZ, W.; FELDHUSEN, J.; GROTE, K.-H.: Instandhaltungsgerecht. In: Pahl/Beitz Konstruktionslehre. Hrsg.: J. Feldhusen; K.-H. Grote. 8. vollst. überarb. Auflage. Springer, Berlin [u. a.] 2013, S. 726-729.
- PARKER, G.; VAN ALSTYNE, M.; CHOUDARY, S. P.: Platform revolution. How networked markets are transforming the economy and how to make them work for you. W. W. Norton & Company, New York 2016.
- PELINO, M.; HEWITT, A.: The Forrester Wave: IoT Software Platforms, Q4 2016. The 11 Providers That Matter Most And How They Stack Up. 18.07.2016. <https://slidex.tips/download/res136087-de> (zuletzt geprüft am: 31.03.2018).

- PELINO, M.; VOCE, C.: Vendor Landscape: IoT Software Platforms. 5.10.2017. <https://reprints.forrester.com/#!/assets/2/57/RES136557/reports> (zuletzt geprüft am: 09.04.2018).
- PERRY, M. J.: Evaluating and choosing an IoT platform. First edition. O'Reilly Media, Sebastopol (CA) 2016.
- PLATTFORM INDUSTRIE 4.0 (Hrsg.): Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0). Eine Einführung Berlin, April 2016. <http://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/rami40-eine-einfuehrung.html> (zuletzt geprüft am: 31.03.2018).
- PLATTNER, H.; KRÜGER, J.: In-memory data and process management. In: Handbook on Business Process Management; Bd.1. Hrsg.: J. Vom Brocke; M. Rosemann. 2. Auflage. Springer, Berlin [u. a.] 2015, S. 445-461.
- PLOOM, T.; KURMANN, A.; HAGEN, C.: Kombination der MDA und SOA als Mittel zur IT-Komplexitätsreduktion bei Credit Suisse. In: MDD, SOA und IT-Management (MSI 2007). Workshop, Oldenburg, April 2007. Hrsg.: U. Steffens; J. S. Addicks; N. Streekmann. GITO, Berlin 2007, S. 59-68.
- POPPER, K. R.: Logik der Forschung. Zur Erkenntnistheorie der Modernen Naturwissenschaft. Schriften zur Wissenschaftlichen Weltauffassung; Bd. 9. Springer, Wien [u. a.] 1935.
- PORTER, M. E.: Competitive advantage. Creating and sustaining superior performance. Free Press, New York 1985.
- PORTER, M. E.; HEPPELMANN, J. E.: How Smart, Connected Products Are Transforming Companies. In: Harvard Business Review 93(2015)10, S. 96-114.
- PORTER, M. E.; HEPPELMANN, J. E.: How Smart, Connected Products Are Transforming Competition. In: Harvard Business Review 92(2014)11, S. 64-88.
- PORTER, M. E.; HEPPELMANN, J. E.: Why Every Organization Needs an Augmented Reality Strategy. In: Harvard Business Review 95(2017)6, S. 46-62.
- POSTSCAPES.COM (Hrsg.): IoT Cloud Platform Landscape. 2017 Vendor List. <https://www.postscapes.com/internet-of-things-platforms/#iot-cloud-platforms-provider-list-comparison> (zuletzt geprüft am: 31.03.2018).
- PRANDINI, M.; LEHMANN, R.; BLUMER, H.; KELLER, J.: Industrielle After Sales Services in China. Rahmenbedingungen, Geschäftsmodelle, Analysen, Empfehlungen. Springer, Wiesbaden [u. a.] 2018.
- PREIM, B.; DACHSELT, R.: Interaktive Systeme, Bd. 2: User Interface Engineering, 3D-Interaktion, Natural User Interfaces. 2. Auflage. Springer, Berlin [u. a.] 2015.
- PRODAN, R.; OSTERMANN, S.: A survey and taxonomy of infrastructure as a service and web hosting cloud providers. In: 2009 10th IEEE/ACM International Conference on Grid Computing. Hrsg.: IEEE Computer Society. IEEE, 2009, S. 17-25.

- PTC (Hrsg.): Machine Learning and IoT Analytics Platform. <https://www.thingworx.com/platforms/thingworx-analytics/> (zuletzt geprüft am: 02.09.2017).
- PTC (Hrsg.): ThingWorx Industrial Innovation Platform. <https://www.ptc.com/en/products/iot/thingworx-platform> (zuletzt geprüft am: 31.03.2018).
- PUSCHMANN, C.; AUSSERHOFER, J.: Social Data APIs. In: The Datafied Society. Studying Culture through Data. Hrsg.: M. T. Schäfer; K. van Es. Amsterdam University Press, Amsterdam 2017, S. 147-154.
- QUINTERO, D.; GENOVESE, W. M.; KIM, K.; LI, M. M.J.; MARTINS, F.; NAINWAL, A.; SMOLEJ, D.; TABINOWSKI, M.; TIWARY, A.: IBM Software Defined Environment. Vervante, Springville (UT) 2015.
- RAFFÉE, H.: Grundprobleme der Betriebswirtschaftslehre. Betriebswirtschaftslehre im Grundstudium der Wirtschaftswissenschaft; Bd. 1. - 9., unveränd. Nachdruck der 1. Auflage. Vandenhoeck & Ruprecht, Göttingen 1995.
- RAHIMZEI, E.: Fragen rund um das Elektrofahrzeug. Wie kommen die Angaben über den Stromverbrauch und die Reichweite von Elektrofahrzeugen zustande? April 2015. http://schaufenster-elektromobilitaet.org/de/content/dokumente/dokumente_1/dokument_details_16708.html (zuletzt geprüft am: 08.03.2018).
- RAJPUROHIT, A.: Big Data for Business Managers. Bridging the gap between Potential and Value. In: IEEE International Conference on Big Data, 2013. 6 - 9 Oct. 2013, Silicon Valley (CA), USA; proceedings. Hrsg.: X. Hu; A. Rajpurohit. IEEE, Piscataway (NJ) 2013, S. 29-31.
- RAUTERBERG, M.: Über das Phänomen "Information". In: Zur Terminologie in der Kognitionsforschung. Hrsg.: B. Becker. GMD, St. Augustin 1989, S. 219-241.
- RAY, P. P.: A survey of IoT cloud platforms. In: Future Computing and Informatics Journal 1(2016)1-2, S. 35-46.
- REHÄUSER, J.; KRCMAR, H.: Wissensmanagement im Unternehmen. In: Wissensmanagement. Managementforschung; Bd. 6. Hrsg.: G. Schreyögg; P. Conrad. De Gruyter, Berlin [u. a.] 1996, S. 1-40.
- REIMANN, G.: VDMA IT-Benchmark Ergebnisse 2017. Ausgewählte IT-Kennzahlen für den Maschinen- und Anlagenbau, VDMA, Frankfurt 20.12.2017.
- REINHART, G.; SCHINDLER, S.: A Strategic Evaluation Approach for Defining the Maturity of Manufacturing Technologies. November 2010. <https://waset.org/publications/11195/a-strategic-evaluation-approach-for-defining-the-maturity-of-manufacturing-technologies> (zuletzt geprüft am: 31.03.2018).
- REISS, M.; REISS, G.: Praxisbuch IT-Dokumentation. Vom Betriebshandbuch bis zum Dokumentationsmanagement – die Dokumentation im Griff. 2., aktualis. u. erw. Auflage. Hanser, München [u. a.] 2016.

- REPSCHLÄGER, J.; PANNICKE, D.; ZARNEKOW, R.: Cloud Computing. Definitionen, Geschäftsmodelle und Entwicklungspotenziale. In: HMD - Praxis der Wirtschaftsinformatik 47(2010)5, S. 6-15.
- RHENSUS, T.: Typisierung von RFID-Anwendungen. Schriftenreihe Rationalisierung und Humanisierung; Bd.102. Hrsg.: G. Schuh. Apprimus, Aachen 2010. - Zugl.: Aachen, Techn. Hochsch., Diss., 2010.
- RICHARDSON, C.; RYMER, J. R.: The Forrester Wave™: Low-Code Development Platforms, Q2 2016. The 14 Providers That Matter Most And How They Stack Up 14.4.2016. <https://agilepoint.com/wp-content/uploads/Q2-2016-Forrester-Low-Code.pdf> (zuletzt geprüft am: 31.03.2018).
- RICHERT, A.: Was Integration Platform as a Service (iPaaS) einen Plattform-Status verleiht. <http://www.searchenterprisesoftware.de/tipp/Was-Integration-Platform-as-a-Service-iPaaS-einen-Plattform-Status-verleiht> (zuletzt geprüft am: 31.03.2018).
- RICHTER, G.; LEHMEYER, O.; SCHOTT, S.: IT-Business-Alignment erfolgreich umsetzen. In: Information Management & Consulting 23(2008)1, S. 17-21.
- RIGGERT, W.: ECM – Enterprise Content Management. Konzepte und Techniken rund um Dokumente. Vieweg+Teubner, Wiesbaden 2009.
- ROHWEDER, J. P.; KASTEN, G.; MALZAHN, D.; PIRO, A.; SCHMID, J.: Informationsqualität – Definitionen, Dimensionen und Begriffe. In: Daten- und Informationsqualität. Auf dem Weg zur Information Excellence. Hrsg.: K. Hildebrand; M. Gebauer; H. Hinrichs; M. Mielke. 3., erw. Auflage. Springer, Wiesbaden [u. a.] 2015, S. 25-46.
- ROPOHL, G.: Allgemeine Systemtheorie. Einführung in transdisziplinäres Denken. Ed. Sigma, Berlin 2012.
- ROSS, J. W.; WEILL, P.; ROBERTSON, D.: Enterprise architecture as strategy. Creating a foundation for business execution. Harvard Business Review Press, Boston (MA) 2006.
- ROTH, M.: Einsatz von Echtzeitstrategien in der MES-Automatisierung. In: Herausforderungen durch Echtzeitbetrieb. Echtzeit 2011. Hrsg.: W. A. Halang. Springer, Berlin [u. a.] 2012, S. 91-100.
- ROWLEY, J.: The wisdom hierarchy. Representations of the DIKW hierarchy. In: Journal of Information Science 33(2007)2, S. 163-180.
- RUDOLF, S.; SCHREY, E.: Datendurchgängigkeit – der Befähiger zur Kollaboration. In: Chefsache Komplexität. Eine Themenreise. Hrsg.: Complexity Management Academy Schuh&Co. Aachen September 2016, S. 95-98. http://www.schuh-group.com/site/assets/files/1533/broschuere_chefsache_komplexitaet_2017-10-25_esa_online.pdf (zuletzt geprüft am: 31.03.2018).

- RUDOLF, S.; SCHREY, E.: Product Lifecycle Management. Herausforderungen einer ganzheitlichen IT-Unterstützung der Geschäftsprozesse. In: Chefsache Komplexität. Eine Themenreise. Hrsg.: Complexity Management Academy Schuh&Co. Aachen September 2016, S. 90-98. http://www.schuh-group.com/site/assets/files/1533/broschuere_chefsache_komplexitaet_2017-10-25_esa_online.pdf (zuletzt geprüft am: 31.03.2018).
- RÜEGG-STÜRM, J.; GRAND, S.: Das St. Galler Management-Modell. 3., überarb. u. weiterentw. Auflage. Haupt, Bern [u. a.] 2017.
- SATTLER, K.-U.: DFG-Schwerpunktprogramm „Skalierbares Datenmanagement für zukünftige Hardware“. In: Datenbank-Spektrum 16(2016)2, S. 183-185.
- SAVOR, T.; DOUGLAS, M.; GENTILI, M.; WILLIAMS, L.; BECK, K.; STUMM, M.: Continuous deployment at Facebook and OANDA. In: Proceedings of the 38th International Conference on Software Engineering Companion - ICSE '16. Hrsg.: L. Dillon; W. Visser; L. Williams. ACM Press, New York 2016, S. 21-30.
- SCHÄFER, D.: Referenzmodelle des IT-Management. In: Lean-Informationstechnik im Finanzdienstleistungssektor. Hrsg.: D. Schäfer. Springer, Wiesbaden [u.a.] 2015, S. 19-34.
- SCHEER, A.-W.: ARIS - Modellierungsmethoden, Metamodelle, Anwendungen. 3., völlig neubearbeitete und erweiterte Auflage. Springer, Berlin [u. a.] 1998.
- SCHEER, A.-W.: ARIS - Modellierungsmethoden, Metamodelle, Anwendungen. 4. Auflage. Springer, Berlin [u. a.] 2001.
- SCHEER, A.-W.: The Big Change. Auswirkungen der neuen Technologien von Industrie 4.0. In: Handel 4.0. Die Digitalisierung des Handels: Strategien, Technologien, Transformation. Hrsg.: R. Gläß; B. Leukert. Springer Gabler, Berlin [u. a.] 2017, S. 51-58.
- SCHEIBMAYER, M.: Bestimmung von Zielgrößen für die Produktstammdatenqualität in produzierenden Unternehmen. Schriftenreihe Rationalisierung; Bd. 130. RHRsg.: G. Schuh. Apprimus, Aachen 2014. - Zugl.: Aachen, Techn. Hochsch., Diss., 2014.
- SCHEPT, K.: BrandZ - Top 100 Most valuable global brands 2017, 2017. <http://www.milwardbrown.com/brandz/top-global-brands/2017> (zuletzt geprüft am: 31.03.2018).
- SCHICKER, E.: Datenbanken und SQL. Eine praxisorientierte Einführung mit Anwendungen in Oracle, SQL Server und MySQL. 5., aktualis. u. erw. Auflage. Springer, Wiesbaden 2017.
- SCHLEIPEN, M.: OPC UA und AutomationML in der Industrie 4.0-Begriffswelt. Informatik Aktuell, 06.04.2016. <https://www.informatik-aktuell.de/betrieb/netzwerke/opc-ua-und-automationml-in-der-industrie-40-begriffswelt.html> (zuletzt geprüft am: 31.03.2018).

- SCHMIDT, C.: Konfiguration überbetrieblicher Koordinationsprozesse in der Auftragsabwicklung des Maschinen- und Anlagenbaus. Schriftenreihe Rationalisierung und Humanisierung; Bd. 87. Hrsg.: G. Schuh. Shaker, Aachen 2008. - Zugl.: Aachen, Techn. Hochsch., Diss., 2008.
- SCHMIDT, C.; KREBS, U.; SCHIEMANN, D.: Produktivität steigern durch den Einsatz von EDI. In: ZWF - Zeitschrift für wirtschaftlichen Fabrikbetrieb 109(2014)6, S. 400-402.
- SCHMIDT, C.; WIDJAJA, T.; SCHÜTZ, A.: Messung der Komplexität von IT-Landschaften auf der Basis von Architektur-Metamodellen: Ein generischer Ansatz und dessen Anwendung im Rahmen der Architektur-Transformation. 2013. <http://subs.emis.de/LNI/Proceedings/Proceedings220/1261.pdf> (zuletzt geprüft am: 31.03.2018).
- SCHMIDT, H.: Experten für künstliche Intelligenz gesucht! Handelsblatt online, 28.09.2017. <http://www.handelsblatt.com/my/unternehmen/beruf-und-buero/digitaler-jobindex/digitaler-job-monitor-experten-fuer-kuenstliche-intelligenz-gesucht/20373114.html?ticket=ST-1858557-GgLwJxBH9eFySJdX4nH-ap1> (zuletzt geprüft am: 31.03.2018).
- SCHNEBERGER, S. L.; MCLEAN, E. R.: The complexity cross. In: Communications of the ACM 46(2003)9, S. 216-225.
- SCHNEEWEIß, C.: Planung; Bd. 2: Konzepte der Prozeß- und Modellgestaltung. Springer, Berlin [u. a.] 1992.
- SCHNEIDER, A. W.; MATTHES, F.: Unternehmensarchitekturgestütztes Controlling zur Beherrschung der IT-Komplexität. In: Controlling 26(2014)12, S. 694-699.
- SCHREINER, M.; HESS, T.; BENLIAN, A.: Gestaltungsorientierter Kern oder Tendenz zur Empirie? Zur neueren methodischen Entwicklung der Wirtschaftsinformatik. - Arbeitsbericht; Nr. 01/2015, Ludwig-Maximilians-Universität, München 2015. http://www.wim.bwl.uni-muenchen.de/download/epub/ab_2015_01.pdf (zuletzt geprüft am: 01.04.2018).
- SCHUETZ, A.; WIDJAJA, T.; KAISER, J.: Complexity In Enterprise Architectures. Conceptualization And Introduction Of A Measure From A System Theoretic Perspective. In: ECIS 2013 Completed Research(2013)Paper 202, S. 1-13.
- SCHUH, G.; ANDERL, R.; GAUSEMEIER, J.; HOMPEL, M. ten; WAHLSTER, W. (Hrsg.): [Studie] Industrie 4.0 Maturity Index. Die digitale Transformation von Unternehmen gestalten. Utz, München 2017. [=2017a]
- SCHUH, G.; BERNARDY, A.; ZELLER, V.; STICH, V.: New Requirement Analysis Approach for Cyber-Physical Systems in an Intralogistics Use Case. In: Collaboration in a Data-Rich World. 18th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2017, Vicenza, Italy, September 18-20, 2017, Proceedings. Hrsg.: L. M. Camarinha-Matos; H. Afsarmanesh; R. Fornasiero. IFIP Advances in Information

- and Communication Technology; 506. Springer, Cham [u. a.] 2017, S. 149-156. [=2017b]
- SCHUH, G.; BLUM, M.: Design of a data structure for the order processing as a basis for data analytics methods. In: 2016 Portland International Conference on Management of Engineering and Technology (PICMET). Hrsg.: IEEE Computer Society. IEEE, 2016, S. 2164-2169.
- SCHUH, G.; BLUM, M.; RESCHKE, J.; BIRKMEIER, M.: Der Digitale Schatten in der Auftragsabwicklung. In: ZWF - Zeitschrift für wirtschaftlichen Fabrikbetrieb 111(2016)1-2/2016, S. 48-51. [=2016a]
- SCHUH, G.; DIELS, F.; ORTLIEB, C.; RIESENER, M.; SCHRÖDER, S.: Agile Produktentwicklung. In: AWK Aachener Werkzeugmaschinen-Kolloquium 2017: Internet of Production für agile Unternehmen. Hrsg.: C. Brecher; F. Klocke; R. Schmitt; G. Schuh. Apprimus, Aachen 2017, S. 29-51. [=2017c]
- SCHUH, G.; GARTZEN, T.; SOUCY-BOUCHARD, S.; BASSE, F.: Enabling Agility in Product Development through an Adaptive Engineering Change Management. In: Procedia CIRP 63(2017)6, S. 342-347. [=2017d]
- SCHUH, G.; GIERTH, A.: Aachener PPS-Modell. In: Produktionsplanung und -steuerung. Grundlagen, Gestaltung und Konzepte. Hrsg.: G. Schuh. Springer, Dordrecht [u. a.] 2007, S. 11-27.
- SCHUH, G.; GIERTH, A.: Einführung. In: Produktionsplanung und -steuerung. Grundlagen, Gestaltung und Konzepte. Hrsg.: G. Schuh. Springer, Dordrecht [u. a.] 2007, S. 5-10.
- SCHUH, G.; HOFFMANN, J.; BLEIDER, M.; ZELLER, V.: Assessment of IS Integration Efforts to Implement the Internet of Production Reference Architecture. In: Collaborative Networks of Cognitive Systems. PRO-VE 2018. Hrsg.: L. Camarinha-Matos; H. Afsarmanesh; Y. Rezgui. IFIP Advances in Information and Communication Technology; 534. Springer, Cham 2018, S. 325-333.
- SCHUH, G.; HOFFMANN, J.; GRUBER, M.; ZELLER, V.: Managing IT Complexity in the Manufacturing Industry. An Agenda for Action. In: Journal on Systemics, Cybernetics and Informatics 15(2017)2, S. 61-65. [=2017e]
- SCHUH, G.; HOFFMANN, J.; GRUBER, M.; ZELLER, V.: Managing IT Complexity in the Manufacturing Industry. An Agenda for Action. In: Proceedings of the 8th International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2017). Hrsg.: N. Callos; J. Horne; B. Sánchez; M. Savoie; A. Tremante. IIS, Winter Garden (Florida) 2017, S. 62-66. [=2017f]
- SCHUH, G. (Hrsg.): Effiziente Auftragsabwicklung mit myOpenFactory. Hanser, München [u. a.] 2008.
- SCHUH, G.; KAMPKER, A. (Hrsg.): Strategie und Management produzierender Unternehmen. Handbuch Produktion und Management; Bd. 1. Springer, Berlin [u. a.] 2011.

- SCHUH, G.; KRUMM, S.; AMANN, W.: Chefsache Komplexität. Springer, Wiesbaden [u. a.] 2013.
- SCHUH, G.: Lebenszyklusorientierte Produktentwicklung. In: Handbuch Unternehmensorganisation. Strategien, Planung, Umsetzung. Hrsg.: D. Spath; E. Westkämper. Living reference work, continuously updated edition. Springer, Berlin [u. a.] 2016, S. 1-16.
- SCHUH, G.; NYHUIS, P.; REUTER, C.; HAUPTVOGEL, A.: Produktionsdaten als Enabler für Industrie 4.0. Gemeinsame Studie der produktionstechnischen Institute IFA, IPMT, Fraunhofer IWU und WZL. In: Werkstatttechnik online 105(2015)4, S. 200-203.
- SCHUH, G.: Produktkomplexität managen. Strategien – Methoden – Tools. 2. überarb. u. erw. Auflage. Hanser, München [u. a.] 2005.
- SCHUH, G.; RIESENER, M.: Produktkomplexität managen. Strategien - Methoden - Tools. 3., vollst. überarb. Auflage. Hanser, München 2018.
- SCHUH, G.; SALMEN, M.; JUSSEN, P.; RIESENER, M.; ZELLER, V.; HENSEN, T.; BEGOVIC, A.; BIRKMEIER, M.; HOCKEN, C.; JORDAN, F.; KANTELBERG, J.; KELZENBERG, C.; KOLZ, D.; MAASEM, C.; SIEGERS, J.; STARK, M.; TÖNNES, C.: Geschäftsmodell-Innovation. In: Handbuch Industrie 4.0: Geschäftsmodelle, Prozesse, Technik. Hrsg.: G. Reinhart. Hanser, München [u. a.] 2017, S. 3-30. [=2017g]
- SCHUH, G.; STICH, V.; BASSE, F.; FRANZKOCH, B.; HARZENETTER, F.; LUCKERT, M.; PROTE, J.; RESCHKE, J.; SCHMITZ, S.; TÜCKS, G.; WEIßKOPF, J.: Change Request im Produktionsbetrieb. In: AWK Aachener Werkzeugmaschinen-Kolloquium 2017: Internet of Production für agile Unternehmen. Hrsg.: C. Brecher; F. Klocke; R. Schmitt; G. Schuh. Apprimus, Aachen 2017, S. 109-131. [=2017h]
- SCHUH, G.; WALENDZIK, P.; LUCKERT, M.; BIRKMEIER, M.; WEBER, A.; BLUM, M.: Keine Industrie 4.0 ohne den Digitalen Schatten. In: ZWF - Zeitschrift für wirtschaftlichen Fabrikbetrieb 111(2016)11, S. 745-748. [=2016b]
- SCHULTE-ZURHAUSEN, M.: Organisation. 6., überarb. u. aktualis. Auflage. Vahlen, München 2014.
- SCHÜTTE, R.: Grundsätze ordnungsmäßiger Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle. Neue betriebswirtschaftliche Forschung; Bd. 233. Gabler, Wiesbaden 1998. - Zugl.: Münster, Univ., Diss., 1997.
- SCHÜTZ, A.: Komplexität von IT-Architekturen. Konzeptualisierung, Quantifizierung, Planung und Kontrolle. Springer Gabler, Wiesbaden 2017.
- SCHÜTZ, A.; WIDJAJA, T.; KAISER, J.: Complexity In Enterprise Architectures - Conceptualization And Introduction Of A Measure From A System Theoretic Perspective. In: ECIS 2013 Completed Research Paper(2013)202, S. 1-12.

- SCULLY, P.: 5 Things To Know About The IoT Platform Ecosystem. Hrsg.: IoT Analytics GmbH. <https://iot-analytics.com/5-things-know-about-iot-platform/> (zuletzt geprüft am: 01.04.2018).
- SESSIONS, R.: Simple architectures for complex enterprises. Microsoft Press, Redmond (WA) 2008.
- SHAHIN, M.; ALI BABAR, M.; ZHU, L.: Continuous Integration, Delivery and Deployment. A Systematic Review on Approaches, Tools, Challenges and Practices. In: IEEE Access 5(2017), S. 3909-3943.
- SHANNON, C. E.; WEAVER, W.: Mathematische Grundlagen der Informationstheorie. Oldenbourg, München [u. a.] 1976.
- SHARMA, S.; SINGH, B.: 5G networks: The next gen evolution. In: 2016 International Conference on Signal Processing and Communication (ICSC). Hrsg.: IEEE Computer Society. IEEE, 2016, S. 55-60.
- SHERMAN, R.: Business intelligence guidebook. From data integration to analytics. Morgan Kaufmann, Waltham 2014.
- SINDERMANN, S.: Schnittstellen und Datenaustauschformate. In: Modellbasierte virtuelle Produktentwicklung. Hrsg.: M. Eigner; D. Roubanov; R. Zafirov. Springer Vieweg, Berlin [u. a.] 2014, S. 327-347.
- SLONE, J. P.: Information quality strategy: An empirical investigation of the relationship between information quality improvements and organizational outcomes. - Unpublished doctoral dissertation, Capella University, Minneapolis, USA 2006. <http://web.mit.edu/smadnick/www/IQ%20Dissertations/Skip.Slone.Dissertation.pdf> (zuletzt geprüft am: 20.10.2017).
- SMIK, B.: The web-based platforms for IoT (Internet of Things) management. - Bachelorthesis, Masaryk University, Brno 2016. <https://is.muni.cz/th/x1tjt/bc-thesis-smik.pdf> (zuletzt geprüft am: 13.04.2018).
- SOLIC, K.; SEBO, D.; JOVIC, F.: Methodology for complexity reduction of IT system. (adjustment of the sessions' methodology). In: Proceedings of the 34th International Convention on Information and Communication technology, Electronics and Microelectronics. Hrsg.: MIPRO. Opatija 2011, S. 1528-1531.
- SONNTAG, A.: Instrument Morphologische Analyse. Hamburg November 2015. <https://www.inf.uni-hamburg.de/de/inst/ab/itmc/research/completed/promidis/instrumente/morphologische-analyse> (zuletzt geprüft am: 01.04.2018).
- SONTOW, R.; SCHWETZ, W.; SIEGERS, J.: CRM in der Praxis - Anwenderzufriedenheit, Nutzen & Perspektiven 2015/2016. Trovarit, Aachen 2015. https://www.crm-finder.ch/fileadmin/Daten/PDF/studien/CRM_in_der_Praxis.pdf (zuletzt geprüft am: 01.04.2018).
- STACHOWIAK, H.: Allgemeine Modelltheorie. Springer, Wien [u. a.] 1973.

- STACHOWIAK, H.: Information(stheorie). In: Handlexikon der Wissenschaftstheorie. Hrsg.: H. Seiffert; G. Radnitzky. 2. Auflage. Deutscher Taschenbuch-Verlag, München 1989, S. 154-158.
- STACHOWIAK, H.: Modell. In: Handlexikon der Wissenschaftstheorie. Hrsg.: H. Seiffert. 2. Auflage. Deutscher Taschenbuch-Verlag, München 1994, S. 219-222.
- STAHL, R.; STAAB, P.: Die Vermessung des Datenuniversums. Datenintegration mithilfe des Statistikstandards SDMX. Springer, Berlin [u. a.] 2017.
- STATISTISCHES BUNDESAMT (Hrsg.): Bruttoinlands-Produkt 2017 für Deutschland. Wiesbaden 11.01.2018. https://www.destatis.de/DE/PresseService/Presse/Pressekonferenzen/2018/BIP2017/Pressebrochure_BIP2017.pdf?__blob=publicationFile (zuletzt geprüft am: 08.04.2018).
- STICH, V.; HOFFMANN, J.; HEIMES, P.: Software-definierte Plattformen: Eigenschaften, Integrationsanforderungen und Praxiserfahrungen in produzierenden Unternehmen. In: HMD - Praxis der Wirtschaftsinformatik 55(2018)1, S. 25-43.
- SUPPLY CHAIN COUNCIL, I.: SCOR: The Supply Chain Reference. 11.0. The Supply Chain Council, Inc, United States of America 2012.
- THE OPEN GROUP (Hrsg.): TOGAF Version 9.1, An Open Group Standard August 2014. <https://publications.opengroup.org/q091> (zuletzt geprüft am: 01.04.2018).
- THIELE, T.; HOFFMANN, M.; MEISEN, T.: Industrial Big Data: Tieferes Prozessverständnis durch Daten. Modernes Informationsmanagement in der Produktion. In: Industrie-4.0-Management 33(2017)4, S. 57-60.
- TIEMEYER, E.: Dokumentation der IT-Systemlandschaft. Architektur-, Asset- und Konfigurationsmanagement. In: Handbuch IT-Systemmanagement. Handlungsfelder, Prozesse, Managementinstrumente, Good-Practices. Hrsg.: E. Tiemeyer. Hanser, München [u. a.] 2016, S. 42-70.
- TIEMEYER, E.: Enterprise Architecture Management (EAM). IT-Architekturen planen und steuern. In: Handbuch IT-Management. Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis. Hrsg.: E. Tiemeyer. 6., überarb. u. erw. Auflage. Hanser, München [u. a.] 2017, S. 97-154.
- TIEMEYER, E. (Hrsg.): Handbuch IT-Management. Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis. 6., überarb. u. erw. Auflage. Hanser, München [u. a.] 2017.
- TIEMEYER, E.: IT-Management. Einordnung, Handlungsfelder, Rollenkonzepte. In: Handbuch IT-Management. Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis. Hrsg.: E. Tiemeyer. 6., überarb. u. erw. Auflage. Hanser, München [u. a.] 2017, S. 1-54.
- TREUTLEIN, P.; ZELLER, V.; MÜLLER, M.: Marktspiegel Business Software: ECM/DMS 2016/2017. Enterprise Content Management/Dokumenten Management Systeme. 3. Auflage, revidierte Ausgabe. Trovarit, Aachen 2016.

- TURCK, M.: Internet of Things: Are We There Yet? The 2016 IoT Landscape 28.03.2016. <http://mattturck.com/2016-iot-landscape/> (zuletzt geprüft am: 01.04.2018).
- ULRICH, H.: Die Betriebswirtschaftslehre als anwendungsorientierte Sozialwissenschaft. In: Die Führung des Betriebes. Hrsg.: M. Geist; R. Köhler. Poeschel, Stuttgart 1981, S. 1-25.
- ULRICH, H.: Die Betriebswirtschaftslehre als anwendungsorientierte Sozialwissenschaft. In: Management. Hrsg.: H. Ulrich. Haupt, Bern [u. a.] 1984, S. 168-199.
- ULRICH, H.: Die Unternehmung als produktives soziales System. Grundlagen der allgemeinen Unternehmungslehre. P. Haupt, Bern [u.a.] 1970.
- ULRICH, P.; HILL, W.: Wissenschaftstheoretische Grundlagen der Betriebswirtschaftslehre (Teil I). In: WiSt - Wirtschaftswissenschaftliches Studium 5(1976)7, S. 304-309. [=1976a]
- ULRICH, P.; HILL, W.: Wissenschaftstheoretische Grundlagen der Betriebswirtschaftslehre (Teil II). In: WiSt - Wirtschaftswissenschaftliches Studium 5(1976)7, S. 345-350. [=1976b]
- URBACH, N.; AHLEMANN, F.: IT-Management im Zeitalter der Digitalisierung. Auf dem Weg zur IT-Organisation der Zukunft. Springer, Berlin [u. a.] 2016.
- VARIAN, H. R.: IT may or may not matter; it depends on how you use IT. 06.05.2004. http://www.nytimes.com/2004/05/06/business/economic-scene-it-may-or-may-not-matter-it-depends-on-how-you-use-it.html?_r=0 (zuletzt geprüft am: 01.04.2018).
- VASISHT, D.; KAPETANOVIC, Z.; WON, J.; JIN, X.; CHANDRA, R.; SINHA, S. N.; KAPOOR, A.; SUDARSHAN, M.; STRATMAN, S.: FarmBeats. An IoT Platform for Data-Driven Agriculture. In: Proceedings of NSDI '17. Hrsg.: A. Akella; J. Howell. USENIX Association, Berkeley (CA) 2017, S. 515-529.
- VDI 2219: Informationsverarbeitung in der Produktentwicklung. VDI (Hrsg.); ICS 03.100.40, 35.240.10, 35.240.50. Beuth Verlag, Berlin, September 2016.
- VDI 5600: Fertigungsmanagementsysteme. VDI (Hrsg.); ICS 35.240.50. Beuth Verlag, Berlin, 10/2016.
- VDI/VDE-GESELLSCHAFT (Hrsg.): Cyber-Physical Systems. Chancen und Nutzen aus Sicht der Automation Düsseldorf [u. a.], April 2013. https://www.vdi.de/uploads/media/Stellungnahme_Cyber-Physical_Systems.pdf (zuletzt geprüft am: 01.04.2018).
- VODAFONE GROUP (Hrsg.): Managing your global M2M estate. Vodafone Global M2M Platform 2015. <http://www.vodafone.com/business/iot/managed-iot-connectivity-platform> (zuletzt geprüft am: 01.04.2018).

- VOGT, A.: Industrie 4.0/IoT Vendor Benchmark 2017. München Oktober 2016. http://www.itac.de/content/cms/publis-hed/3ee1daa87e725976f86f880df1483856/files_ExpertonGroup_Industrie4.0_I-oTBenchmark2017_LightVersion_ger.pdf (zuletzt geprüft am: 01.04.2018).
- WANG, R. Y.; STRONG, D. M.: Beyond Accuracy: What Data Quality Means to Data Consumers. In: Journal of Management Information Systems 12(1996)4, S. 5-33.
- WANGLER, L.; ENGELHARDT, S. von; WISCHMANN, S.: Eigenschaften und Erfolgsfaktoren digitaler Plattformen. Eine Studie im Rahmen der Begleitforschung zum Technologieprogramm AUTONOMIK für Industrie 4.0 des Bundesministeriums für Wirtschaft und Energie Berlin, März 2017. https://www.digitale-technologien.de/DT/Redaktion/DE/Downloads/Publikation/autonomik-studie-digitale-plattformen.pdf?__blob=publicationFile&v=10 (zuletzt geprüft am: 01.04.2018).
- WEBER, M.: Enterprise Architecture Management – neue Disziplin für die ganzheitliche Unternehmensentwicklung. Berlin 2011. <https://www.bitkom.org/Publikationen/2011/Leitfaden/Leitfaden-EAM-Enterprise-Architecture-Management/EAM-Enterprise-Architecture-Management-BITKOM-Leitfaden.pdf> (zuletzt geprüft am: 01.04.2018).
- WEBER, W.; KABST, R.; BAUM, M.: Einführung in die Betriebswirtschaftslehre. Springer, Wiesbaden [u. a.] 2018.
- WEIBER, R.; WEIBER, T.; MOHR, L.: Butler-Services als Dienstleistung 4.0 zur Entlastung von Konsumenten in ihren Alltagsprozessen. In: Dienstleistungen 4.0; Bd. 1: Konzepte - Methoden - Instrumente. Hrsg.: M. Bruhn; K. Hadwich. Springer, Wiesbaden [u. a.] 2017, S. 61-96.
- WEINREICH, U.: Lean Digitization. Digitale Transformation durch agiles Management. Springer, Berlin [u. a.] 2016.
- WELTER, M.: Die Forschungsmethode der Typisierung. Charakteristika, Einsatzbereiche und praktische Anwendung. In: WiSt - Wirtschaftswissenschaftliches Studium 35(2006)2, S. 113-116.
- WENKING, M.; BENNINGHAUS, C.; GROGGERT, S.: Die Zukunft von Manufacturing Data Analytics. Implikationen für eine erfolgreiche Datennutzung im produzierenden Umfeld. In: Industrie 4.0 Management 33(2017)4, S. 33-37.
- WIDJAJA, T.; KAISER, J.; TEPEL, D.; BUXMANN, P.: Heterogeneity in IT Landscapes and Monopoly Power of Firms: A Model to Quantify Heterogeneity. In: 33. International Conference on Information Systems. (ICIS 2012). Hrsg.: ICIS. Curran, Orlando (FL) 2012, S. 1-14. <http://aisel.aisnet.org/icis2012/proceedings/Break-throughIdeas/3> (zuletzt geprüft am: 18.05.2018).
- WIENKER, R.: Komplexität der Unternehmens-IT beherrschbar machen. In: ERP-Management 2(2006)4, S. 38-41.

- WILDE, T.; HESS, T.: Forschungsmethoden der Wirtschaftsinformatik. Eine empirische Untersuchung. In: Wirtschaftsinformatik 49(2007)4, S. 280-287.
- WILKENS, A.: Microsoft-Manager: Keine neuen Funktionen mehr für Windows-Phones. Heise online, News 10/2017. <https://www.heise.de/newsticker/meldung/Microsoft-Manager-Keine-neuen-Funktionen-mehr-fuer-Windows-Phones-3852862.html> (zuletzt geprüft am: 01.04.2018).
- WILLCOCKS, L. P.; LACITY, M. C.; SAUER, C.: Introduction. In: Outsourcing and Offshoring Business Services. Hrsg.: L. Willcocks; C. Sauer; M. Lacity. Springer, Cham [u. a.] 2017, S. 1-22.
- WINTER, R.; AIER, S.: [Definition] Informationssystem-Architektur. In: Enzyklopädie der Wirtschaftsinformatik. - Online-Lexikon 2016. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/daten-wissen/Informationsmanagement/Information-/Informationssystem-Architektur/index.html> (zuletzt geprüft am: 01.04.2018).
- WINTER, R.: Architectural Thinking. In: Business & Information Systems Engineering 6(2014)6, S. 361-364.
- WINZER, P.: Generic Systems Engineering. Springer, Berlin [u. a.] 2016.
- WITTMANN, W.: Unternehmung und unvollkommene Information. Westdt. Verl., Köln [u. a.] 1959.
- WODEHOUSE, C.: SQL vs. NoSQL Databases: What's the Difference? <https://www.upwork.com/hiring/data/sql-vs-nosql-databases-whats-the-difference/> (zuletzt geprüft am: 01.04.2018).
- WÖHE, G.; DÖRING, U.; BRÖSEL, G.: Einführung in die allgemeine Betriebswirtschaftslehre. 26., überarb. u. aktualis. Auflage. Vahlen, München 2016.
- WOLTERS, P.; DÜNNEBACKE, D.: Wertbeitrag der IT. Messen des Wertbeitrages der Unternehmens-IT. Schlussbericht zum Forschungsprojekt 16105 N. 2012. http://www.ifu.rwth-aachen.de/fileadmin/user_upload/IfU/Forschung/Projekt-Archiv/WertIT/WertIT_Abschlussbericht.pdf (zuletzt geprüft am: 01.04.2018).
- ZACHMAN, J. A.: A framework for information systems architecture. In: IBM systems journal 26(1987)3, S. 276-292.
- ZARNEKOW, R.; BRENNER, W.: Integriertes Informationsmanagement. Vom Plan, Build, Run zum Source, Make, Deliver. In: Informationsmanagement. Konzepte und Strategien für die Praxis. Hrsg.: R. Zarnekow. dpunkt-Verl., Heidelberg 2004, S. 3-24.
- ZARNEKOW, R.: Produktionsmanagement von IT-Dienstleistungen. Grundlagen, Aufgaben und Prozesse. Springer, Berlin [u. a.] 2007.
- ZDRAVKOVIĆ, M.; TRAJANOVIĆ, M.; SARRAIPA, J.; JARDIM-GONÇALVES, R.; LEZOCHÉ, M.; AUBRY, A.; PANETTO, H.: Survey of Internet-of-Things platforms. In: ICIST 2016. Proceedings. Hrsg.: M. Zdravković; M. Trajanović; Z. Konjović. Society for Information Systems and Computer Networks, Belgrad 2016, S. 216-220.

- ZELEWSKI, S.: Grundlagen. In: Betriebswirtschaftslehre; Bd. 1. Hrsg.: H. Corsten; M. Reiß. 4. vollst. überarb. u. wesentl. erw. Auflage. Oldenbourg, München [u. a.] 2008, S. 1-98.
- ZELLER, V.; HOFFMANN, J.; GRUBER, M.; SCHUH, G.: GradelT – Prozessqualität bei kleinen und mittleren IT-Service-Providern. In: HMD - Praxis der Wirtschaftsinformatik 53(2016)2, S. 200-212.
- ZELLER, V.: Systematische Verbesserung von Prozessen für IT-Dienstleistungen. Schriftenreihe Rationalisierung; Bd. 154. RHrsg.: G. Schuh. Apprimus, Aachen 2018. - Zugl.: Aachen, Techn. Hochsch., Diss., 2018.
- ZILIS, S.; CHAM, J.: The current state of machine intelligence 3.0. - Watching the appeal and applications of machine intelligence expand. 07.11.2016. <https://www.oreilly.com/ideas/the-current-state-of-machine-intelligence-3-0> (zuletzt geprüft am: 01.04.2018).
- ZIMMERMAN, S.; RENTROP, C.: Schatten-IT. In: HMD - Praxis der Wirtschaftsinformatik 49(2012)288, S. 60-68.
- ZITNIK, S.; JANKOVIC, M.; PETROVIC, K.; BAJEC, M.: Architecture of standard-based, interoperable and extensible IoT platform. In: 2016 24th Telecommunications Forum (TELFOR). Belgrade, Serbia, November, 22-23, 2016 : proceedings of papers. Hrsg.: Đ. Paunović; Z. Jovanović; A. Nešković; L. Milić; I. Reljin; V. Marković. IEEE, Piscataway (NJ) 2016, S. 857-860.
- ZWICKY, F.: Entdecken, Erfinden, Forschen im Morphologischen Weltbild. Droemer-Knaur, München 1966.
- ZWICKY, F.: Morphological Astronomy. In: The Observatory 68(1948)08, S. 121-143.
- ZWICKY, F.: Morphological Astronomy. Springer, Berlin [u. a.] 1957.
- ZWICKY, F.: Morphologisches Denken und Vorgehen. In: Die neuen Methoden der Entscheidungsfindung. Hrsg.: G. W. Tumm; H. Bielowski. Verlag Moderne Industrie, Landsberg 1972, S. 130-143.

Anhang

Verzeichnis der Anlagen

A.1 Zentrale wissenschaftliche Quellen der Arbeit.....	302
A.2 Übersicht der Literatureinflüsse	305
A.3 Experteninterviews	306
A.4 Veränderung der Suchanfragen zu zentralen Informationssystemen	307
A.5 Übersicht software-definierter Plattformen	308
A.6 Eigenschaften software-definierter Plattformen	310
A.7 Vergleich von Schichtenmodellen.....	311
A.8 Informationssysteme bei e.GO	314

A.1 Zentrale wissenschaftliche Quellen der Arbeit

Frü eine höhere Transparenz der genutzten Quellen werden anbei alle Literaturquellen aufgelistet, die zehn oder mehr Mal in der vorliegenden Arbeit zitiert wurden und damit den größten Einfluss auf die Ergebnisse haben (Abbildung A-1 visualisiert diesen Einfluss für die zehn wichtigsten Quellen).

Tabelle A-1: Übersicht über die zentralen Literaturquellen (eigene Darstellung)

Quelle	Anzahl Zitationen
HABERFELLNER, R.; DE WECK, O.; FRICKE, E.; VÖSSNER, S. (Hrsg.): Systems Engineering. Grundlagen und Anwendung. 13., aktualisierte Auflage. Orell Füssli, Zürich 2015	31
KRCMAR, H.: Informationsmanagement. 6., überarb. Auflage. Springer, Berlin [u. a.] 2015	29
LEIMEISTER, J. M.: Einführung in die Wirtschaftsinformatik. 12. vollst. neu überarb. Auflage. Springer Gabler, Wiesbaden [u. a.] 2015	22
LEHNER, F.; WILDNER, S.; SCHOLZ, M.: Wirtschaftsinformatik. Eine Einführung. 2. Auflage. Hanser, München [u. a.] 2008	22
HANSCHKE, I.: Enterprise Architecture Management - einfach und effektiv. Ein praktischer Leitfaden für die Einführung von EAM. 2., überarb. Auflage. Hanser, München [u. a.] 2016	19
ALPAR, P.; ALT, R.; BENSBERG, F.; GROB, H. L.; WEIMANN, P.; WINTER, R.: Anwendungsorientierte Wirtschaftsinformatik. Strategische Planung, Entwicklung und Nutzung von Informationssystemen. 8., überarb. Auflage. Springer Vieweg, Wiesbaden 2016	17
TIEMEYER, E. (Hrsg.): Handbuch IT-Management. Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis. 6., überarb. u. erw. Auflage. Hanser, München [u. a.] 2017	15
HEINRICH, L. J.; STELZER, D.; RIEDL, R.: Informationsmanagement. Grundlagen, Aufgaben, Methoden. 11. vollst. überarb. Auflage. DE Gruyter Oldenbourg, Berlin [u. a.] 2014	14
ARBEITSKREIS SMART SERVICE WELT SOWIE ACATECH (Hrsg.): [Abschlussbericht] Smart Service Welt - Umsetzungsempfehlungen für das Zukunftsprojekt 'Internetbasierte Dienste für die Wirtschaft'. Berlin März 2015. http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Projekte/Laufende_Projekte/Smart_Service_Welt/Smart_Service_Welt_2015/BerichtSmartService2015_D_lang_bf.pdf (zuletzt geprüft am: 29.03.2018)	13

Quelle	Anzahl Zitationen
BRECHER, C.; KLOCKE, F.; SCHMITT, R.; SCHUH, G. (Hrsg.): AWK Aachener Werkzeugmaschinen-Kolloquium 2017: Internet of Production für agile Unternehmen. Apprimus, Aachen 2017	13
HOFFMANN, J.; HEIMES, P.: Informationssystem-Architekturen produzierender Unternehmen für die Digitalisierung gestalten. In: HMD Praxis der Wirtschaftsinformatik 55(2018)6, S. 984-1005. https://link.springer.com/article/10.1365/s40702-018-0440-8	12
KRAUSE, T.; STRAUß, O.; SCHEFFLER, G.; KETT, H.; LEHMANN, K.; RENNER, T.: IT-Plattform für das Internet der Dinge (IoT). Basis intelligenter Produkte und Services. Fraunhofer Verlag, Stuttgart 2017	12
SCHUH, G.; ANDERL, R.; GAUSEMEIER, J.; HOMPEL, M. ten; WAHLSTER, W. (Hrsg.): [Studie] Industrie 4.0 Maturity Index. Die digitale Transformation von Unternehmen gestalten. Utz, München 2017a	12
STICH, V.; HOFFMANN, J.; HEIMES, P.: Software-definierte Plattformen: Eigenschaften, Integrationsanforderungen und Praxiserfahrungen in produzierenden Unternehmen. In: HMD - Praxis der Wirtschaftsinformatik 55(2018)1, S. 25-43	12
FOTH, E.: Erfolgsfaktoren für eine digitale Zukunft. IT-Management in Zeiten der Digitalisierung und Industrie 4.0. Springer, Berlin [u.a.] 2016	11
FREY, J.: ASaP - Integrationsplattform für Smart Services in Intelligenten Umgebungen. Saarbrücken 2015. - Zugl.: Saarbrücken, Univ., Diss., 2015	11
LEMKE, C.; BRENNER, W.; KIRCHNER, K.: Gestalten des digitalen Zeitalters. Springer Gabler, Berlin [u. a.] 2017	11
SCHÜTTE, R.: Grundsätze ordnungsmäßiger Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle. Neue betriebswirtschaftliche Forschung; Bd. 233. Gabler, Wiesbaden 1998. - Zugl.: Münster, Univ., Diss., 1997	11
URBACH, N.; AHLEMANN, F.: IT-Management im Zeitalter der Digitalisierung. Auf dem Weg zur IT-Organisation der Zukunft. Springer, Berlin [u. a.] 2016	11
EISEND, M.; KUß, A.: Grundlagen empirischer Forschung. Springer, Wiesbaden [u. a.] 2017	10
KAIB, M.: Enterprise Application Integration: Grundlagen, Integrationsprodukte, Anwendungsbeispiele. Springer, Berlin [u. a.] 2002	10

Quelle	Anzahl Zitationen
WELTER, M.: Die Forschungsmethode der Typisierung. Charakteristika, Einsatzbereiche und praktische Anwendung. In: WiSt - Wirtschaftswissenschaftliches Studium 35(2006)2, S. 113-116	10

A.2 Übersicht der Literatureinflüsse

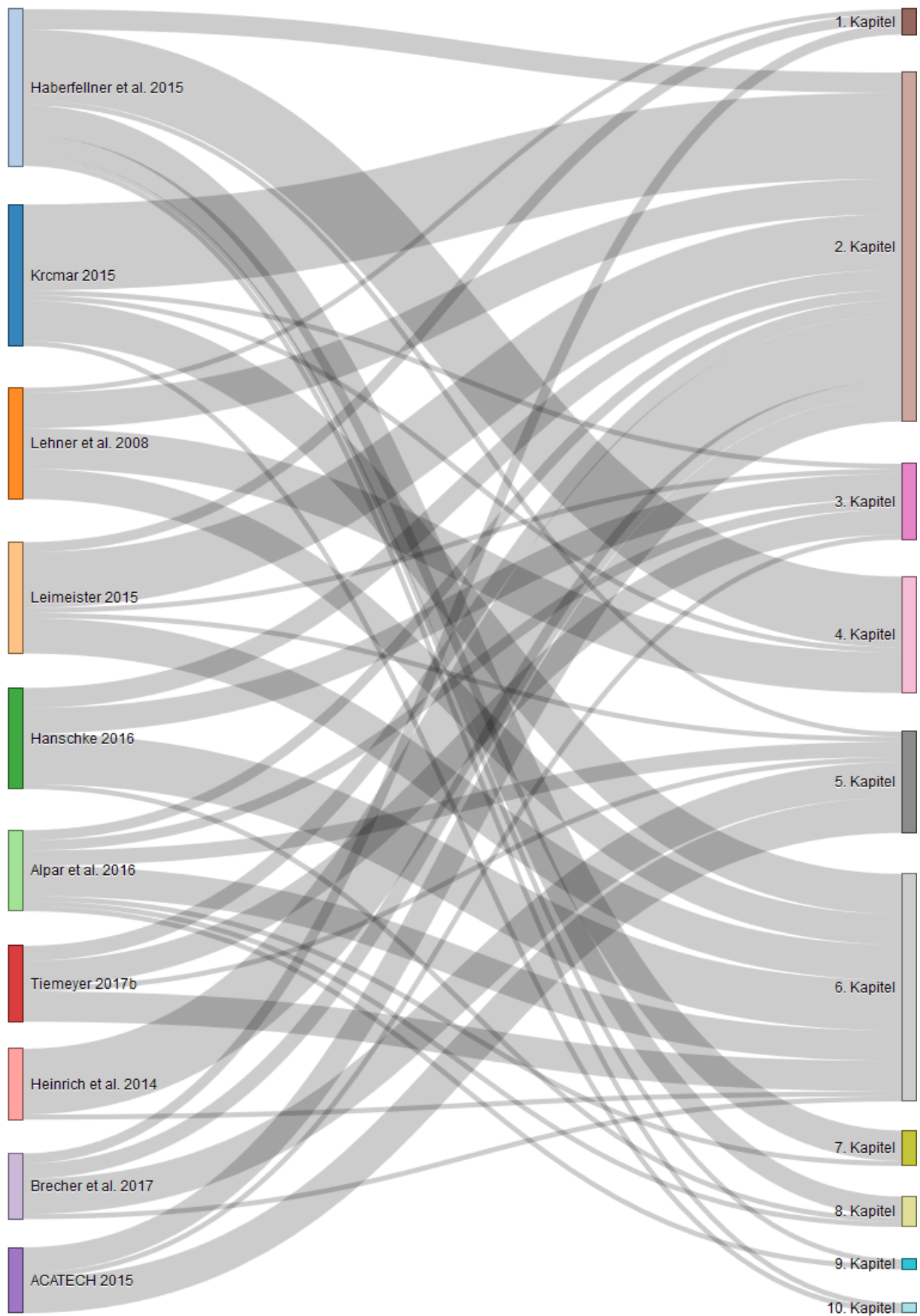


Abbildung A-1: Übersicht über die zehn Quellen mit den meisten Zitierungen und deren Einfluss auf die einzelnen Kapitel der Arbeit (eigene Darstellung)

A.3 Experteninterviews

Tabelle A-2: Experteninterviews

Name	Unternehmen	Funktion	Datum
Jürgen Boshold	Ortlinghaus	IT-Leiter	Jun. 2015
Dr. Rolf Rettlinger	Lenze	IT-Leiter / Geschäftsführer	Jun. 2015
Stefan Orbach	Heidelberger Druckmaschinen	CIO	Jul. 2015
Dietrich Baumann	PTC	Chief Architect	20.10.2016
Dr. Ruperte Deger	e.GO Mobile AG	CIO	Dez. 2016; diverse Gespräche in 2017
Pit Heimes	M2Expert	Projektleiter IoT	01.08.2017
Anika Wübben	e.GO Mobile AG	Senior Manager Connected Service	09.08.2017
Nicklas Radbruch	e.GO Mobile AG	Head of PLM	06.03.2018
Stefan Derissen	GPS GmbH	IT-Leiter	07.03.2018
Thies Bach	e.GO Mobile AG	ERP-Experte	07.03.2018
Ernest Debets	e.GO Mobile AG	VP IT & Digitaliza- tion	16.03.2018
Jon Maessen	PTC	Senior IoT Consul- tant	22.03.2018
Martin Bremer	e.GO Mobile AG	MES-Experte	div. Gespräche im April 2018
Christoph Certain	PTC	Technical Account Director	18.04.2018
Dr. Thomas Holzapfel	Innogy SE	Leiter Information Management Cor- porate Functions	20.07.2018

A.4 Veränderung der Suchanfragen zu zentralen Informationssystemen

Die Veränderung der in Unternehmen genutzten Informationssysteme lässt sich auch anhand der Suchanfragen bei Google erkennen. Google ermöglicht über Google Trends⁵⁶ die Untersuchung von Suchanfragen über den zeitlichen Verlauf. Für die zentralen Informationssysteme, die in der vorliegenden Arbeit vorgestellt wurden, zeigen die folgenden beiden Abbildungen die entsprechenden Suchverläufe.⁵⁷

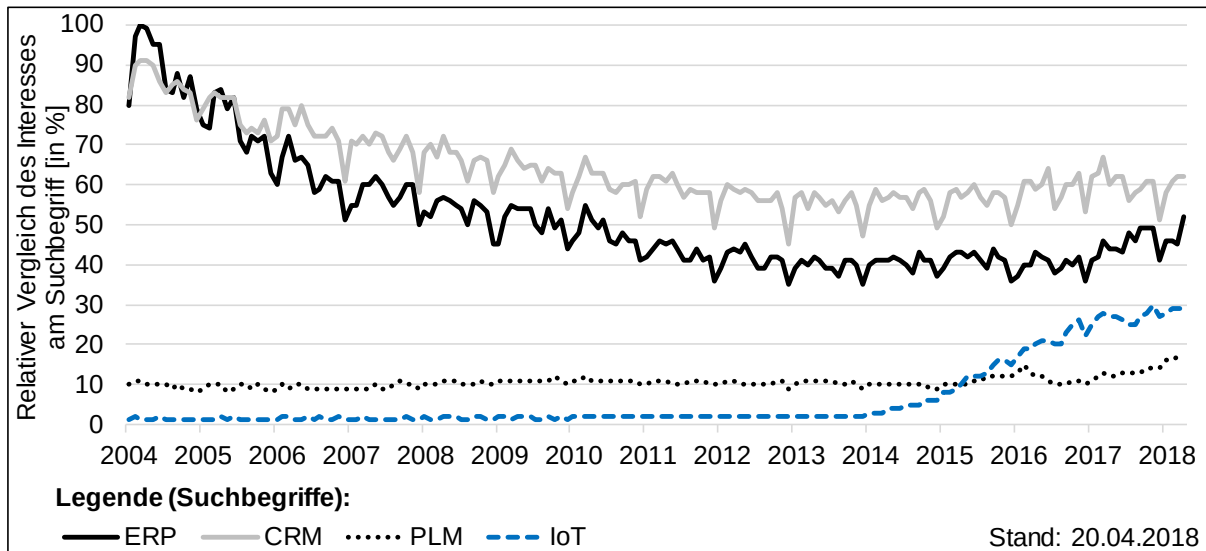


Abbildung A-2: Veränderung des Interesses an den Suchanfragen zu ERP, CRM, PLM und IoT bei Google (eigene Darstellung mit Daten von Google Trends)

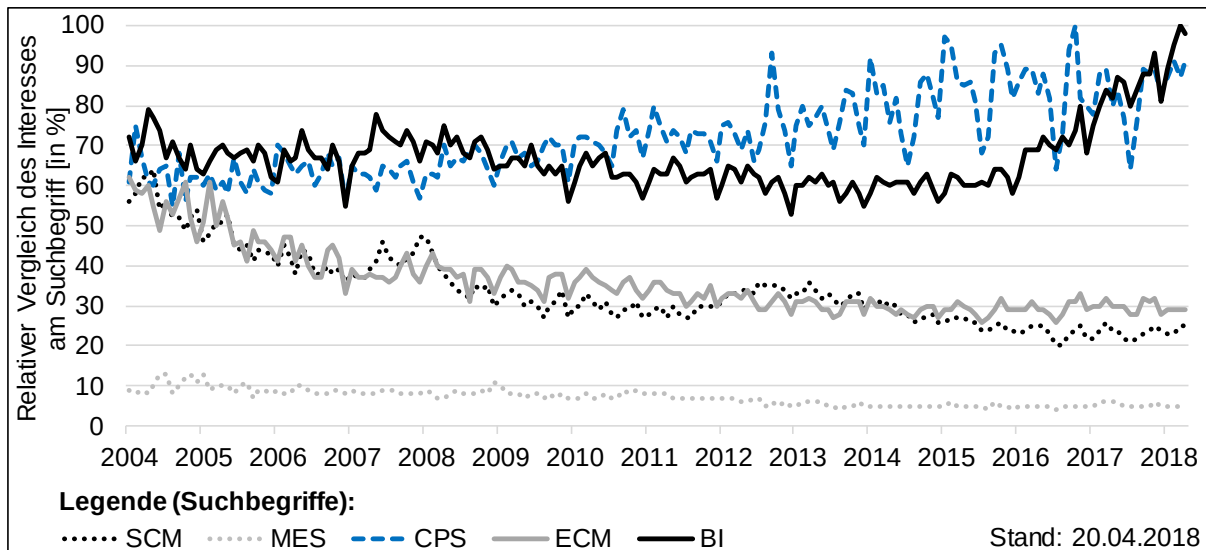


Abbildung A-3: Veränderung des Interesses an den Suchanfragen zu SCM, MES, CPS, ECM und BI bei Google (eigene Darstellung mit Daten von Google Trends)

⁵⁶ Erreichbar unter <https://trends.google.de/trends/explore>.

⁵⁷ Anmerkung: Bei „BI“ wurde aufgrund der Mehrdeutigkeit des Suchbegriffs stattdessen nach dem Thema „Business Intelligence“ gesucht.

A.5 Übersicht software-definierter Plattformen

Tabelle A-3: Typisierung software-definierter Plattformen (eigene Darstellung)

Plattformname	Hersteller	Einsatzgebiet „Produzierende Unternehmen“ laut Hersteller	Plattformtyp
Software-definierte Plattformen mit hoher Relevanz⁵⁸ (basierend auf PELINO U. HEWITT 2016; CROOK U. MACGILLIVRAY 2017)			
Agile IoT Platform	Ayla Networks	X	Typ I
Amazon Web Services (AWS) IoT Platform	Amazon	X	Typ I (unter Nutzung anderer AWS)
Azure IoT Suite	Microsoft	X	Typ I (über Integration in Microsoft-Systemwelt)
Bosch IoT Suite	Bosch Software Innovations	X	Typ II
Cloud Platform	SAP	X	Typ I (über Integration in SAP-Systemwelt)
Fujitsu Cloud Service K5 IoT Platform	Fujitsu	X	Typ II
Jasper Control Center	Cisco	X	Typ II
Murano IoT Platform	Exosite	X	Typ II
Oracle IoT Cloud Service	Oracle	X	Typ II
Predix	GE	X	Typ I
ThingWorx	PTC	X	Typ I
Watson IoT Platform	IBM	X	Typ II
Xively	LogMeIn	X	Typ II
Software-definierte Plattformen mit mittlerer Relevanz (basierend auf RAY 2016; ZDRAVKOVIĆ ET AL. 2016; SMIK 2016; TURCK 2016)			
Aeris IoT Services	Aeris Communications		Typ III
AirVantage	Sierra Wireless	X	Typ II
Arrayent	Prodea		Typ III
Axoom	Axoom (Trumpf)	X	Typ II
Carriots	Altair	X	Typ III
Chordant	InterDigital		Typ II (Erweiterbar über Marketplace)
Covisint	OpenText	X	Typ II
Cumulocity	Software AG	X	Typ I
Edge	MachineShop	X	Typ III

⁵⁸ Die Relevanz ist für diese Plattformen mit „hoch“ gekennzeichnet, da sie von vielzitierten Studien referenziert werden, die eine hohe Marktdurchdringung der Plattformen beschreiben, und weil deren Funktionsumfang gut zur Infrastruktur des *Internet of Production* passt.

Plattformname	Hersteller	Einsatzgebiet „Produzierende Unternehmen“ laut Hersteller	Plattformtyp
Everyware Device Cloud	Eurotech	X	Typ I
Evrythng	Evrythng		Typ II
Fusion Connect	Autodesk	X	Typ III
IoT Accelerator	Ericsson		Typ III
IoT Security Platform	Mocana	X	Typ III
IoT Suite	People Power		Typ III
Kaa	KaaProject	X	Typ III
Managed IoT Connectivity	Vodafone		Typ III
MindSphere	Siemens	X	Typ II (allerdings Integration mit anderen Plattformen)
Motionloft	Motionloft		Typ II
nxFast	Numerex		Typ III
Particle	Particle		Typ III
Placemeter	Netgear		Typ II
Pulse	Agheera		Typ II
RuBAN	Davra Networks		Typ II
SecuriThings	SecuriThings		Typ III
SensorCloud	LORD Sensing	X	Typ III
Sumo Logic	Sumo Logic	X	Typ II
Telit IoT Platform	Telit Communications	X	Typ II
Temboo	Temboo	X	Typ II
Thethings	The Things Network	X	Typ II
ThingSpeak	MathWorks		Typ II
Uptake	Uptake	X	Typ II
WiSilica	WiSilica		Typ III
Zebra Savanna	Zebra Technologies		Typ II

A.6 Eigenschaften software-definierter Plattformen

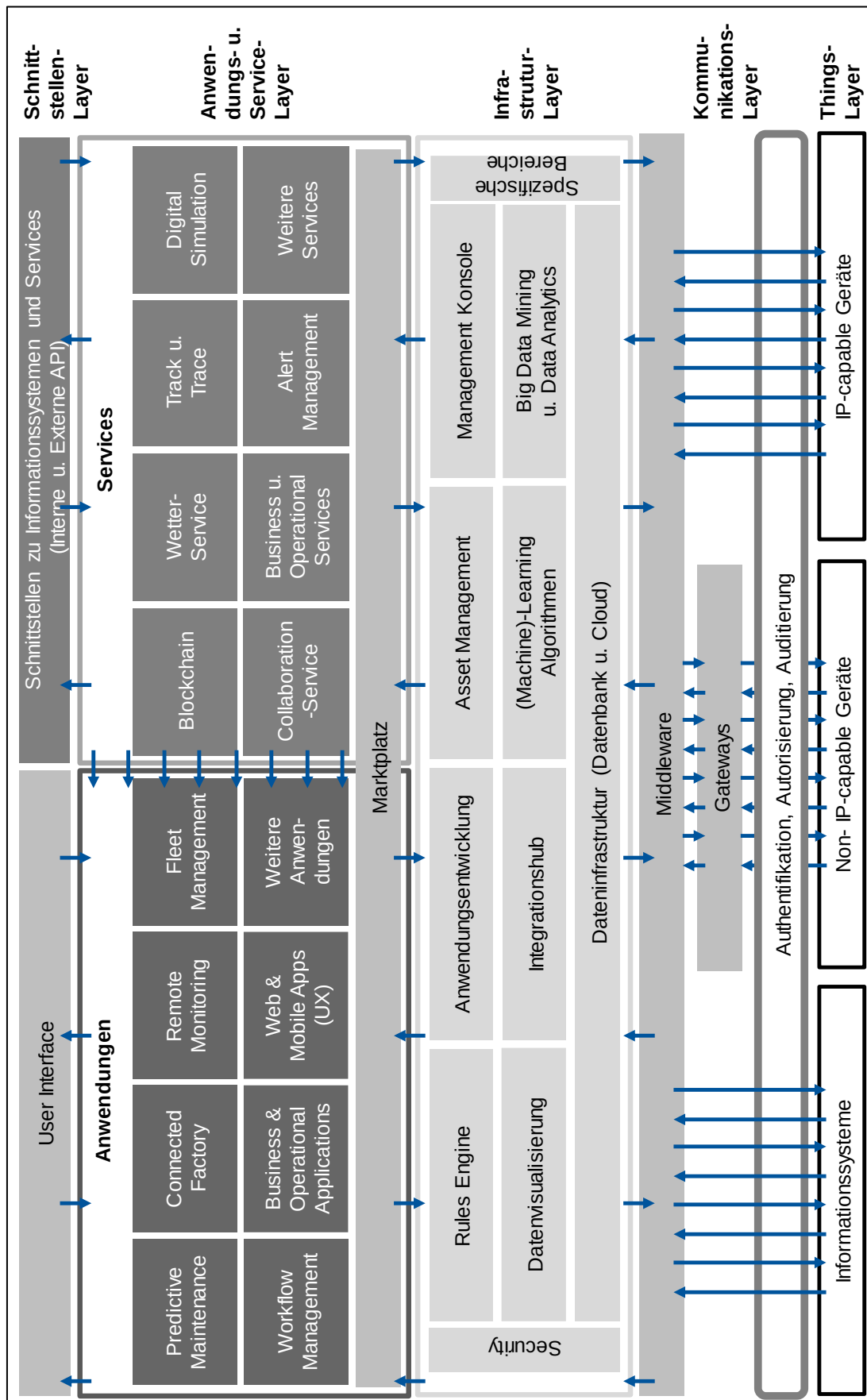


Abbildung A-4: Vorschlag einer erweiterten Referenzarchitektur mit möglichen Eigenschaften software-definierter Plattformen (i. A. a. HOFFMANN ET AL. 2018, S. 7)

A.7 Vergleich von Schichtenmodellen

Tabelle A-4: Detaillierter Vergleich der Schichtenmodelle

Modell	RAMI 4.0	IIRA	ISO-OSI	Internet of Things Reference Model (IRM)	IIoT Referenzarchitektur für die Kommunikation (IRK)
Quelle	DIN SPEC 91345	INDUSTRIAL INTERNET CONSORTIUM 2015	LEIMEISTER 2015, S. 101	GREEN 2014	HEIDRICH U. JIJUN LUO 2016
Ebene					
Datenhaltung und Erzeugung	1 Asset: Physische Elemente (Blech, Dokumente); Mensch	Physical Systems: Ebene, die durch die Control-Einheit gesteuert wird	1 Physische Schicht, Bit-Übertragungsschicht: Umwandlung Bits in systempassendes Signal		
Vernetzung	2 Integration: Generierung eines elektronischen Ereignisses aus Asset, Bereitstellung Hardware, Software	Control: Auslesen von Sensordaten, Übergabe von Befehlen über Aktoren. Sicherstellen der Kommunikationswege, Modellierung und Asset-Management	2 Sicherungsschicht: Sicherstellung Übertragung von Daten, Fehlererkennung, zielgerichtete Datenweiterleitung	Physical Devices & Controllers "Edge": Datensammlung von Sensoren, Geräten, Maschinen	"lokaler Bereich": Vernetzung zwischen physischem Gerät und erster logischer Informationsebene (Edge-Computing)

Modell	RAMI 4.0	IIRA	ISO-OSI	Internet of Things Reference Model (IRM)	IIoT Referenzarchitektur für die Kommunikation (IRK)
Quelle Ebene	DIN SPEC 91345	INDUSTRIAL INTERNET CONSORTIUM 2015	LEIMEISTER 2015, S. 101	GREEN 2014	HEIDRICH U. JIJUN LUO 2016
	<u>3 Communication:</u> Vereinheitlichung der Kommunikation durch Bereitstellung eines einheitlichen Datenformates		<u>3 Vermittlungsschicht:</u> Routing von Daten, Adressierung an Endgeräte, unabhängig vom Übertragungsmedium <u>4 Transportschicht:</u> Zuordnung der Datenpakete zu einer Anwendung <u>5 Kommunikationssteuerschicht:</u> Steuerung der Verbindungen und des Datenaustauschs, Kontrolle der Datenströme	<u>Connectivity:</u> Zusammenfassung zu Kommunikations- und Prozesseinheiten	
Applikation	<u>4 Information:</u> Ausführung von einfachen Regeln, Erzeugung von Ereignissen, Gewinnung höherwertiger Daten, Entgegennahme von Ereignissen und Weiterverarbeitung	<u>Information:</u> Sammeln von Daten aus verschiedenen Quellen, Berechnung, Umformung, Modellierung und Übertragung der Daten.		<u>Edge Computing:</u> Datentransformation auf richtiges Format sowie einfache Analyse <u>Data Accumulation:</u> Speicherung von Daten; <u>Data Abstraction:</u> Zugangsverwaltung und	<u>"globaler Bereich":</u> Vernetzung zwischen Diensten/Anwendungen vom Edge-Computing und

Modell	RAMI 4.0	IIRA	ISO-OSI	Internet of Things Reference Model (IRM)	IIoT Referenzarchitektur für die Kommunikation (IRK)
Quelle Ebene	DIN SPEC 91345	INDUSTRIAL INTERNET CONSORTIUM 2015	LEIMEISTER 2015, S. 101	GREEN 2014	HEIDRICH U. JIJUN LUO 2016
				Zusammenfassung/Aggregation von Daten	Dienst-/Anwendungen des Cloud-Computing
	5 Functional: Logische Funktionen und fachliche Funktionalität	<u>Application / Operations</u> <u>Operations:</u> Management von Assets, u. a. auch Prognosewerkzeuge und Optimierung <u>Application:</u> Logik und Regelwerke, API und Anwenderoberfläche		<u>Application:</u> Controlling, Analysen, Reporting	
Visualisierung	<u>6 Business:</u> Abbildung Geschäftsprozess, Modellierung von Regeln für System, Verbindungselement zwischen verschiedenen Ebenen		<u>6 Darstellungsschicht:</u> Umwandlung von systemabhängigen Daten in unabhängiges Format		
Business		<u>Business:</u> Ende-zu-Ende-Unterstützung von Business-Prozessen	<u>7 Anwendungsschicht:</u> Bereitstellung von Funktionen für die Anwendung, Verbindung zu unteren Schichten herstellen, Eingabe von Zielwerten	<u>Collaboration & Processes:</u> Einbeziehung von Führungskräften und Geschäftsprozessen	

A.8 Informationssysteme bei e.GO

Tabelle A-5: Liste der Informationssysteme bei e.GO (eigene Darstellung zum Stand März 2018)

Informations-system	Her-steller	Schnitt-stellen	Out-sourcing-grad	Echt-zeitfä-higkeit	Status des Soft-warele-benszyk-lus	Sys-temur-sprung	System-verant-wortung	Organi-sations-einheit	Nutzer-kreis	Sys-tem-kon-zept	Kernpro-zess	betroffene Ebenen	Informa-tions-qualität	Daten-standi-sie-rung
Wind-chill	PTC	gene-risch / vorde-finiert	Infrastruk-tur Out-sourcing	keine Echt-zeit	Evolution / Servi-cing	Stand-ardsys-tem	IT-Abtei-lung	Entwick-lung	Teilmenge des Unter-nehmens	PLM	Konzept / Entwurf u. Detaillierung / Produkti-onsplanung / Nutzung u. Service	[Alle]	zugäng-lich	mög-lich
Creo	PTC	gene-risch / vorde-finiert	interner Betrieb	keine Echt-zeit	Evolution / Servi-cing	Stand-ardsys-tem	IT-Abtei-lung	Entwick-lung	Teilmenge des Unter-nehmens	PLM	Entwurf u. Detaillierung	Visualisie-rung / Ap-plikation	-	mög-lich
Integrity	PTC	vorde-finiert	interner Betrieb	keine Echt-zeit	Evolution / Servi-cing	Stand-ardsys-tem	IT-Abtei-lung	Entwick-lung	Einzelper-sonen	PLM	Marktana-lyse / Kon-zept	Visualisie-rung / Ap-plikation	-	mög-lich
Thing-Worx	PTC	gene-risch / vorde-finiert	Infrastruk-tur Out-sourcing	weiche Echt-zeit	Initiale Software	Stand-ardsys-tem	IT-Abtei-lung	Entwick-lung / Einkauf / Produk-tion / Vertrieb	Teilmenge des Unter-nehmens	Sons-tige	Entwurf u. Detaillierung / Produkti-onsplanung / Herstellung / Vertrieb u. Nutzung u. Service	[Alle]	aktuell / zugäng-lich	mög-lich
Capital	Mentor	gene-risch	interner Betrieb	keine Echt-zeit	Evolution / Servi-cing	Stand-ardsys-tem	IT-Abtei-lung	Entwick-lung	Einzelper-sonen	PLM	Entwurf u. Detaillierung	Visualisie-rung / Ap-plikation	-	nicht mög-lich
Matlab	Math Works	gene-risch	interner Betrieb	keine Echt-zeit	Evolution / Servi-cing	Stand-ardsys-tem	Fachbe-reich	Entwick-lung	Einzelper-sonen	PLM	Entwurf u. Detaillierung	Visualisie-rung / Ap-plikation	-	mög-lich
Inca	ETAS	gene-risch	interner Betrieb	keine Echt-zeit	Evolution / Servi-cing	Stand-ardsys-tem	Fachbe-reich	Entwick-lung	Einzelper-sonen	PLM	Entwurf u. Detaillierung	Visualisie-rung / Ap-plikation	-	nicht mög-lich
GT Suite	Gama Technologies	gene-risch	interner Betrieb	keine Echt-zeit	Evolution / Servi-cing	Stand-ardsys-tem	Fachbe-reich	Entwick-lung	Einzelper-sonen	PLM	Entwurf u. Detaillierung	Visualisie-rung / Ap-plikation	-	nicht mög-lich
Star-CCM Plus	Siemens	gene-risch	interner Betrieb	keine Echt-zeit	Evolution / Servi-cing	Stand-ardsys-tem	Fachbe-reich	Entwick-lung	Einzelper-sonen	PLM	Entwurf u. Detaillierung	Visualisie-rung / Ap-plikation	-	nicht mög-lich
d-SPACE		gene-risch	interner Betrieb	keine Echt-zeit	Evolution / Servi-cing	Stand-ardsys-tem	Fachbe-reich	Entwick-lung	Einzelper-sonen	PLM	Entwurf u. Detaillierung	Visualisie-rung / Ap-plikation	-	nicht mög-lich
Ansys	Ansys	gene-risch	interner Betrieb	keine Echt-zeit	Evolution / Servi-cing	Stand-ardsys-tem	Fachbe-reich	Entwick-lung	Einzelper-sonen	PLM	Entwurf u. Detaillierung	Visualisie-rung / Ap-plikation	-	mög-lich
PSI penta	PSI A&I	gene-risch / vorde-finiert	Infrastruk-tur Out-sourcing	weiche Echt-zeit	Evolution / Servi-cing	Stand-ardsys-tem	IT-Abtei-lung	Einkauf / Produk-tion	Teilmenge des Unter-nehmens	ERP	Produktions-planung / Herstellung / Vertrieb u. Nutzung u. Service / Unterstüt-zungspro-zess	Applikation / Vernet-zung / Datenhaltung u. -erzeugung	zugäng-lich / vollständ-ig	mög-lich
PSI wms	PSI A&I	vorde-finiert	Infrastruk-tur Out-sourcing	harte Echt-zeit	Initiale Software	Stand-ardsys-tem	IT-Abtei-lung	Produk-tion	Teilmenge des Unter-nehmens	ERP	Herstellung	Applikation / Vernet-zung / Datenhaltung u. -erzeugung	zugäng-lich / vollständ-ig	nicht mög-lich
PSI mes	PSI A&I	vorde-finiert	Infrastruk-tur Out-sourcing	harte Echt-zeit	Evolution / Servi-cing	Stand-ardsys-tem	IT-Abtei-lung	Produk-tion	Teilmenge des Unter-nehmens	MES	Herstellung	Applikation / Vernet-zung / Datenhaltung u. -erzeugung	aktuell / zugäng-lich / vollständ-ig	nicht mög-lich
Bab-tec.Q	Bab-tec	gene-risch	Infrastruk-tur Out-sourcing	weiche Echt-zeit	Initiale Software	Stand-ardsys-tem	IT-Abtei-lung	Produk-tion	Einzelper-sonen	ERP	Konzept / Entwurf u. Detaillierung / Herstellung	Visualisie-rung / Ap-plikation	zugäng-lich / vollständ-ig	mög-lich

Informations-system	Hersteller	Schnittstellen	Outsourcinggrad	Echtzeitfähigkeit	Status des Softwarelebenszyklus	Systemum-sprung	System-verant-wortung	Organi-sations-einheit	Nutzer-kreis	Sys-tem-kon-zept	Kernpro-zess	Betroffene Ebenen	Informa-tions-qualität	Daten-standi-sie-rung
dyna-mics CRM	Micro-soft	gene-risch	Anwen-dungssys-tem Out-sourcing	keine Echt-zeit	Evolution / Servi-cing	Stan-dardsys-tem	IT-Abtei-lung	Vertrieb	Teilmenge des Unter-nehmens	CRM	Vertrieb / Nutzung u. Service	Applikation / Vernet-zung / Da-tenhaltung u. -erzeu-gung	aktuell / zugäng-lich	mög-lich
Ver-triebs-konfigu-rator	just-Sel-ling	gene-risch	Anwen-dungssys-tem Out-sourcing	keine Echt-zeit	Evolution / Servi-cing	Stan-dardsys-tem	Fachbe-reich	Vertrieb	Teilmenge des Unter-nehmens	CRM	Vertrieb	Visualisie-rung	-	nicht mög-lich
Kloepfel SRM	Kloepfel	gene-risch	Business-prozess Outsor-cing	keine Echt-zeit	Initiale Software	Stan-dardsys-tem	IT-Abtei-lung	Einkauf	Teilmenge des Unter-nehmens	SCM	Entwurf u. Detaillierung / Produkti-onsplanung / Herstellung	Business u. Prozesse	-	nicht mög-lich
evi-danza		gene-risch	Infrastruk-tur Out-sourcing	keine Echt-zeit	Initiale Software	Stan-dardsys-tem	IT-Abtei-lung	Verwal-tung u. Manage-ment	Teilmenge des Unter-nehmens	BI	Unterstüt-zungspro-zess	Visualisie-rung / Ap-plikation	-	mög-lich
eGecko	CSS	vorde-finiiert	Infrastruk-tur Out-sourcing	keine Echt-zeit	Evolution / Servi-cing	Stan-dardsys-tem	IT-Abtei-lung	Verwal-tung u. Manage-ment	Teilmenge des Unter-nehmens	Sons-tige	Unterstüt-zungspro-zess	Applikation	-	nicht mög-lich
Heaven HR	HeavenHR	gene-risch	Anwen-dungssys-tem Out-sourcing	keine Echt-zeit	Initiale Software	Stan-dardsys-tem	IT-Abtei-lung	Verwal-tung u. Manage-ment	unterneh-mens-übergrei-fend	Sons-tige	Unterstüt-zungspro-zess	Applikation / Vernet-zung / Da-tenhaltung u. -erzeu-gung	-	mög-lich
Office 2016	Micro-soft	gene-risch	Anwen-dungssys-tem Out-sourcing	keine Echt-zeit	Evolution / Servi-cing	Stan-dardsys-tem	Extern	[Alle]	unterneh-mens-übergrei-fend	Sons-tige	[Alle]	Visualisie-rung / Ap-plikation	-	mög-lich
Project-Place	Plan-view	gene-risch	Anwen-dungssys-tem Out-sourcing	keine Echt-zeit	Evolution / Servi-cing	Stan-dardsys-tem	Fachbe-reich	[Alle]	unterneh-mens-übergrei-fend	Sons-tige	[Alle außer Recycling / Unterstüt-zungspro-zess]	Applikation / Vernet-zung / Da-tenhaltung u. -erzeu-gung	zugäng-lich	nicht mög-lich
Qwiki	Mo-dell Aache-n	gene-risch	Anwen-dungssys-tem Out-sourcing	keine Echt-zeit	Evolution / Servi-cing	Stan-dardsys-tem	Fachbe-reich	[Alle]	unterneh-mens-übergrei-fend	Sons-tige	Unterstüt-zungspro-zess	Visualisie-rung	-	nicht mög-lich
Web-seite	GPS	gene-risch	Business-prozess Outsor-cing	keine Echt-zeit	Evolution / Servi-cing	Individu-alsys-tem	Extern	Vertrieb	Einzelper-sonen	Sons-tige	Vertrieb	Visualisie-rung	-	nicht mög-lich
Own-cloud	Open Sourc-e	gene-risch	Anwen-dungssys-tem Out-sourcing	keine Echt-zeit	Evolution / Servi-cing	Ange-passtes Stan-dardsys-tem	Extern	[Alle]	unterneh-mens-übergrei-fend	Sons-tige	[Alle]	Vernetzung / Datenhal-tung u. -er-zeugung	aktuell / zugäng-lich	nicht mög-lich

