

Technische Universität München

Fakultät für Informatik
Lehrstuhl für Wirtschaftsinformatik (I 17)
Prof. Dr. Helmut Krcmar

Referenzarchitektur eines integrierten Informationssystems zur Unterstützung der Instandhaltung

Andreas Reidt

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigten Dissertation.

Vorsitzender: Prof. Dr. Martin Bichler
Prüfer der Dissertation: 1. Prof. Dr. Helmut Krcmar
2. Prof. Dr. Florian Matthes

Die Dissertation wurde am 08.06.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 29.01.2019 angenommen.

Zusammenfassung

Die fortschreitende Digitalisierung und Trends wie die Industrie 4.0 führen zu großen Umwälzungen im produzierenden Gewerbe. Unterstützende Prozesse wie die Instandhaltung trifft dieser Wandel besonders stark, da deren Komplexität und Bedeutung durch neuartige Technologien und Geschäftsmodelle sowie die zunehmende Serviceorientierung wächst. Höhere Anforderungen an das Personal durch kürzere Innovationszyklen, komplexere Tätigkeiten, steigende IT-Durchdringung und Kostendruck sind die Folge und stehen einem steigenden Alter der Beteiligten in Deutschland und einer hohen Fluktuation des Personals gegenüber. Um diese Herausforderungen zu meistern und zur Unterstützung des Personals bei der Instandhaltung können Informationssysteme entwickelt werden, die ein Wissensmanagement anbieten, Fernwartung und vorrausschauende Instandhaltung ermöglichen und kontextsensitive Informationen über neuartige Benutzerschnittstellen bereitstellen.

Entwicklung und Einsatz solcher Informationssysteme sind jedoch aus einer Vielzahl an Gründen bisher kaum in der Praxis angekommen: Die Entwicklung solcher Systeme erfordert ein tiefes Verständnis der Domäne und der technischen Möglichkeiten. Weiter sind die einzelnen sehr komplexen Elemente und Teilsysteme nur in Verbindung mit anderen Systemen sinnvoll nutzbar. Fehlende Standards sowie isolierte, proprietäre Entwicklungen führen zusammen mit der Unkenntnis über umfassenden und nicht nur anwendungsfallspezifischen Anforderungen zu Inkompatibilitäten, schwerer Erweiterbarkeit und eingeschränkter unternehmensübergreifender Zusammenarbeit. Daraus folgt, dass die Entwicklung solcher Systeme eine hohe Komplexität und Kostenintensität aufweist, wodurch es insbesondere für kleinere und mittlere Unternehmen nicht möglich ist an den Potentialen der Digitalisierung der Instandhaltung teilzuhaben.

Um die integrierte Entwicklung von Instandhaltungssystemen zu erleichtern, wird in dieser Arbeit eine Referenzarchitektur für ein integriertes Informationssystem zur Unterstützung der Instandhaltung entwickelt. Referenzarchitekturen sind in Forschung und Praxis ein probates Mittel, um Domänen- und Architekturwissen bereitzustellen und damit die Entwicklung zu fördern. Die entworfene Referenzarchitektur soll die Entwicklung durch die Bereitstellung von Domänenwissen über die Instandhaltung, generische und spezifische Anforderungen an Instandhaltungssysteme sowie eine abstrakte, auf Erweiterung ausgelegte Architektur des zu entwickelnden Systems mit vorab definierten Variationspunkten unterstützen. Die Informationen sollen so dargestellt werden, dass dadurch die domänenübergreifende Zusammenarbeit zwischen an dem Entwicklungsprozess beteiligten Stakeholdern gefördert wird.

Zur Bearbeitung dieser Fragestellung wird, das gestaltungsorientierte Forschungsparadigma als Basis gewählt. Anhand von Literaturanalysen wird der aktuelle Stand der Wissenschaft und Praxis bzgl. Referenzarchitekturen, Instandhaltung und deren Systemen und Anforderungen erhoben. Anschließend werden die Art der Instandhaltung bei verschiedenen Industrieunternehmen untersucht und Anforderungen an Instandhaltungssysteme ermittelt. Hieraus werden generische und optionale Elemente eines integrierten Instandhaltungssystems extrahiert. Basierend auf diesen Informationen wird die Referenzarchitektur zur Erleichterung

der Entwicklung erarbeitet. Um die Eignung der Referenzarchitektur dahingehend zu überprüfen, wird sie mittels verschiedener Methoden evaluiert und ihre Nützlichkeit damit attestiert

Inhaltsverzeichnis

ZUSAMMENFASSUNG	II
INHALTSVERZEICHNIS	IV
ABBILDUNGSVERZEICHNIS	IX
TABELLENVERZEICHNIS	XI
ABKÜRZUNGSVERZEICHNIS	XIII
1 EINLEITUNG	1
1.1 Motivation und Relevanz.....	1
1.2 Forschungsleitende Fragestellungen.....	5
1.3 Wissenstheoretische Grundlagen.....	8
1.3.1 Einordnung in die Wirtschaftsinformatik.....	8
1.3.2 Forschungsdesign	8
1.4 Aufbau der Arbeit.....	13
2 MERKMALE VON REFERENZARCHITEKTUREN	15
2.1 Einleitung.....	15
2.2 Begriffliche und theoretische Grundlagen.....	16
2.2.1 Softwarearchitektur	16
2.2.2 Produktlinienarchitektur	17
2.2.3 Referenzmodell	17
2.2.4 Definition Referenzarchitektur.....	18
2.3 Methodik zur Erstellung von Referenzarchitekturen.....	34
2.3.1 Methoden zur Erstellung von Referenzarchitekturen.....	34
2.3.2 Verwendete Methodik zur Erstellung von Referenzarchitekturen.....	37
2.4 Zusammenfassung	41
3 INSTANDHALTUNG IM KONTEXT DER INDUSTRIE 4.0	42
3.1 Einleitung.....	42
3.2 Definition und Begriffe der Instandhaltung.....	44
3.3 Instandhaltungskonzepte und -strategien.....	46
3.3.1 Instandhaltungskonzepte	47
3.3.2 Instandhaltungsstrategien	52
3.4 Verbundene Systeme für die Instandhaltung.....	54
3.4.1 Informationssysteme eines produzierenden Unternehmens.....	55
3.4.2 Informationssysteme zur expliziten Unterstützung der Instandhaltung	61

Inhaltsverzeichnis	V
3.4.2.3 Condition Monitoring System.....	62
3.4.3 Weitere verwandte Systembegriffe	65
3.4.4 Diskussion verbundene Systeme	66
3.5 Zusammenfassung	68
4 EXTRAKTION GENERISCHER ANFORDERUNGEN AN INTEGRIERTE INSTANDHALTUNGSSYSTEME	69
4.1 Einleitung.....	69
4.2 Literaturanalyse: Anforderungen an ein integriertes Instandhaltungssystem.....	70
4.2.1 Vorgehen Literaturanalyse	70
4.2.2 Diskussion der Ergebnisse	74
4.3 Praxisanforderungen an ein integriertes Instandhaltungssystem.....	93
4.3.1 Vorgehensweise der Anforderungserhebung	93
4.3.2 Die untersuchten Unternehmen	94
4.3.3 Anforderungen aus der Praxis	95
4.4 Diskussion der Ergebnisse.....	104
4.4.1 Diskussion Verbindung Literatur- und Praxisanforderung	104
4.4.2 Diskussion der Anforderungsbetrachtung	110
4.5 Zusammenfassung	113
5 DARSTELLUNGSMODELL DER RAI	114
5.1 Einleitung.....	114
5.2 Darstellung von Referenzarchitekturen	115
5.2.1 Logische Sicht	116
5.2.2 Entwicklungssicht	116
5.2.3 Prozesssicht	116
5.2.4 Physische Sicht.....	116
5.2.5 Szenarios	116
5.3 Angepasstes Sichtenmodell der RAI.....	118
5.3.1 Funktionale Sicht.....	119
5.3.2 Verteilungssicht.....	123
5.3.3 Use Case Sicht.....	124
5.3.4 Prozesssicht	124
5.3.5 Weitere Besonderheiten bei der Darstellung.....	126
5.4 Zusammenfassung	128
6 REFERENZARCHITEKTUR RAI	129

6.1	Einleitung.....	129
6.2	Funktionale Sicht.....	130
6.2.1	M.1. Handlungsleitfäden/Checklisten/Prüflisten	133
6.2.2	M.2. Dokumentenmanagement	135
6.2.3	M.3. Wiki	136
6.2.4	M.4. Anlageninformationen	138
6.2.5	M.5. Anlagenübersicht und -auswertung	140
6.2.6	M.6. Berechnung/Zugriff auf Produktionsplanung	142
6.2.7	M.7. Interne Navigation	143
6.2.8	M.8. Kataster für Hilfs- und Betriebsstoffe der Anlagen.....	144
6.2.9	M.9. Auftragsverwaltung/Priorisierung	145
6.2.10	M.10. Mitarbeitermanagement.....	150
6.2.11	M.11. Wartungsmanagement	152
6.2.12	M.12. Auftragsmanagement in Verbindung mit ERP	155
6.2.13	M.13. Schichtbuchfunktionalität/Synchronisation	156
6.2.14	M.14. Work and People Tracking	157
6.2.15	M.15. Arbeitszeiterfassung	158
6.2.16	M.16. Fehlererkennung & Condition Monitoring.....	160
6.2.17	M.17. Fehlerdatenbank.....	164
6.2.18	M.18. Fehlermeldung und -darstellung	167
6.2.19	M.19. Predictive Maintenance	169
6.2.20	M.20. Synchronisation Hersteller und Betreiber.....	172
6.2.21	M.21. Navigation.....	173
6.2.22	M.22. Reisekostenabrechnung	174
6.2.23	M.23. Serviceprotokolle	175
6.2.24	M.24. Vertragsmanagement	176
6.2.25	M.25. Signierfunktion	177
6.2.26	M.26. Kommunikation	178
6.2.27	M.27. Konnektivität/Intranetzugang	180
6.2.28	M.28. Ersatzteilmanagement.....	181
6.2.29	M.29. QR-CODE auslesen	182
6.2.30	M.30. Remote Zugriff/Fernwartung auf Anlagen/Maschinen	184
6.2.31	M.31. Technisches Nutzermanagement	185
6.2.32	M.32. Verarbeitung audiovisueller Medien	187

6.2.33	M.33. Funktionalitäten mobiles Endgerät	188
6.2.34	M.34. Synchronisation zw. Geräten und Systemen	190
6.2.35	M.35. Maschinenspezifika	191
6.2.36	M.36. Pluginintegration.....	192
6.2.37	M.37. Wetterdaten.....	193
6.2.38	M.38. Semantische Suche	195
6.2.39	M.39. Personalisierung des Systems	196
6.2.40	M.40. Telefonersatz.....	196
6.3	Verteilungssicht	198
6.4	Prozesssicht	200
6.4.1	Grundaktivität.....	200
6.4.2	Aktivität A1 – Inspektion/Wartung Total Productive Maintenance	204
6.4.3	Aktivität A2 – Inspektion Wartung Instandhalter	205
6.4.4	Aktivität A3 – Instandsetzung.....	206
6.4.5	Aktivität A4 – Vereisung der Windenergieanlage (WEA)	207
6.4.6	Aktivität A5 – Ereignisgesteuerte Instandsetzung	208
6.4.7	Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung.....	209
6.4.8	Aktivität A7 – Zustandsabfrage Wartung (Condition Monitoring)	210
6.4.9	Aktivität A8 – Verfügbarkeit (technisch/organisatorisch).....	212
6.4.10	Aktivität A9 – Auswertung Fehlerhistorien einer Anlage	213
6.4.11	Aktivität A10 – Synchronisation.....	214
6.5	Use Case Sicht	216
6.6	Diskussion Referenzarchitektur.....	219
6.7	Zusammenfassung	226
7	EVALUATION DER REFERENZARCHITEKTUR RAI	227
7.1	Einleitung.....	227
7.2	Ziele und Vorgehen der Evaluation.....	228
7.3	Anforderungen an die RAI	231
7.4	Analytische Evaluation der RAI	240
7.5	Evaluation der RAI anhand des Anwendungsfalles „Ressourcen-Cockpit“	246
7.5.1	Prototyp Ressourcen-Cockpit.....	246
7.5.2	Vorgehen Experteninterview.....	254
7.6	Zusammenfassung	264
8	FAZIT UND AUSBLICK	265

Inhaltsverzeichnis	VIII
8.1 Einleitung.....	265
8.2 Zusammenfassung und Ausblick.....	266
8.3 Implikationen für die Praxis und die Forschung	271
8.3.1 Theoretischer Beitrag und Forschungsergebnisse.....	271
8.3.2 Praktischer Beitrag der Forschungsergebnisse.....	273
8.4 Limitationen.....	276
8.5 Ausblick.....	277
REFERENZEN.....	279
AUFLISTUNG DER RELEVANTEN VORVERÖFFENTLICHUNGEN.....	296
ANHANG	297
8.6 Fragebogen Anforderungsaufnahme	297
8.7 Leitfaden Telefoninterview	302

Abbildungsverzeichnis

Abbildung 1: Design Science Rahmenkonzept mit drei Zyklen	10
Abbildung 2: Forschungsvorgehen der vorliegenden Arbeit (FF = Forschungsfragen)	11
Abbildung 3: Aufbau der Arbeit	13
Abbildung 4: Ziele und Treiber von Referenzarchitekturen	26
Abbildung 5: Abstraktionsniveau und Architekturhierarchie von Referenzarchitekturen.....	30
Abbildung 6: Abbau des Abnutzungsvorrates und seine Erstellung durch Maßnahmen der Instandhaltung	44
Abbildung 7: Unterscheidung zwischen Instandhaltungskonzepten, -strategien und -maßnahmen	46
Abbildung 8: Die acht Pfeiler des TPM nach dem Vorschlag der JIPM	50
Abbildung 9: Kategorisierung von Instandhaltungsarten.....	53
Abbildung 10: Übersicht der potentiellen Informationssysteme für Produktion und Instandhaltung	55
Abbildung 11: Häufigkeit von Systembezeichnungen und Techniken in den Ergebnissen der Literaturrecherche	78
Abbildung 12: Verteilung der gefundenen Artikel nach Wissenschaftsdomäne	79
Abbildung 13: 4+1 Sichten einer Softwarearchitektur.....	115
Abbildung 14: Darstellung des Sichtenmodells der RAII.....	118
Abbildung 15: Erklärung Aufbau der Module	121
Abbildung 16: Legende simplifizierte Moduldarstellung	121
Abbildung 17: Darstellung Prozesssicht – Veränderte Darstellungselemente.....	125
Abbildung 18: Übersicht über generische/optionale Module und Bausteine.....	131
Abbildung 19: Gesamtübersicht Module/Bausteine der RAII	132
Abbildung 20: Soll-Ist Vergleich auf Basis dynamischer Prozesseingriffsgrenzen	162
Abbildung 21: Verteilungssicht	198
Abbildung 22: Aktivität – Grundaktivität (graue Aktionen/Module sind optional)	204
Abbildung 23: Aktivität A1 – Use Case UC1- U1 – Inspektion/Wartung TPM.....	205
Abbildung 24: Aktivität A2 – Use Case UC2-U1 – Inspektion Wartung IH.....	206
Abbildung 25: Aktivität A3 – Use Case UC3-U1 – Instandsetzung.....	207
Abbildung 26: Aktivität A4 – Use Case UC4-U3 – Vereisung der Windenergieanlage (WEA)	208

Abbildung 27: Aktivität A5 - Use Case UC5-U2 – Ereignisgesteuerte Instandsetzung von Anlagen	209
Abbildung 28: Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung.....	210
Abbildung 29: Aktivität A7 – Use Case UC7-U2 – Zustandsabfrage Wartung (Condition Monitoring)	211
Abbildung 30: Aktivität A8 – Use Case UC8/UC9 – Auswertungen Verfügbarkeit (technisch/Organisatorisch).....	213
Abbildung 31: Aktivität A9 – Use Case UC9-U2: Auswertungen Fehlerhistorie einer Anlage	214
Abbildung 32: Aktivität A10 – Use Case Unteraktivität – Synchronisation	215
Abbildung 33: Unterteilung des Evaluationsvorgehen in drei Abschnitte.....	229
Abbildung 34: Fehlerkatalog und Fehlermeldung	249
Abbildung 35: Anzeige aktueller Fehler mit Status	249
Abbildung 36: Anzeige aktuelle Aufträge und Auftragshistorie	250
Abbildung 37: Ansicht Handlungsanweisungen zur Fehlerbehebung Ressourcen-Cockpit..	250
Abbildung 38: Dashboard eines Mitarbeiters mit Übersicht der aktuellen Aufträge und Mitteilungen	251
Abbildung 39: Anlageninformationen und Betriebsdaten	251
Abbildung 40: Wartungsmanagement.....	252
Abbildung 41: Ersatzteilmanagement des Ressourcen-Cockpits.....	252
Abbildung 42: Nachrichtenhub Ressourcen-Cockpit.....	253
Abbildung 43: Allgemeines inhaltsanalytisches Ablaufmodell nach Mayring.....	257
Abbildung 44: Auswertung Evaluation Experteninterview nach einzelnen Personen.....	262
Abbildung 45: Auswertung Evaluation Experteninterview Gesamt	263
Abbildung 46: Beitrag Forschung und Praxis	274

Tabellenverzeichnis

Tabelle 1: Übersicht über Definitionen des Begriffs Referenzarchitektur.....	19
Tabelle 2: Klassifikationsschema Referenzarchitekturen	31
Tabelle 3: Angewandtes Klassifikationsschema auf die RAI.....	32
Tabelle 4: Suchterme und (benutzte) Ergebnisse der Literaturrecherche	73
Tabelle 5: Klassifikation der Ergebnisse der Literaturanalyse von Instandhaltungssystemen	77
Tabelle 6: Anforderungen Instandhaltungssystem – Technischer Kundendienst und Teleservice	80
Tabelle 7: Anforderungen Instandhaltungssystem - Remote Maintenance	81
Tabelle 8: Anforderungen Instandhaltungssystem – Auftragsmanagement und Auftragsinformationen	82
Tabelle 9: Anforderungen Instandhaltungssystem – Wartungsmanagement.....	82
Tabelle 10: Anforderungen Instandhaltungssystem – Instandhaltungsplanung, -optimierung und Risikoklassifikation.....	83
Tabelle 11: Anforderungen Instandhaltungssystem – Anlagenübersicht und -informationen.	84
Tabelle 12: Anforderungen Instandhaltungssystem – Ersatzteilmanagement	85
Tabelle 13: Anforderungen Instandhaltungssystem – Mitarbeitermanagement	85
Tabelle 14: Anforderungen Instandhaltungssystem – Kommunikation.....	86
Tabelle 15: Anforderungen Instandhaltungssystem – Mobiles System.....	87
Tabelle 16: Anforderungen Instandhaltungssystem – (Historische) Auswertungen.....	87
Tabelle 17: Anforderungen Instandhaltungssystem – Fehlermanagement	88
Tabelle 18: Anforderungen Instandhaltungssystem – Condition Monitoring und Predictive Maintenance	88
Tabelle 19: Anforderungen Instandhaltungssystem – Dokumentenmanagement.....	89
Tabelle 20: Anforderungen Instandhaltungssystem – Wissensmanagement	89
Tabelle 21: Anforderungen Instandhaltungssystem – Handlungsempfehlungen.....	90
Tabelle 22: Anforderungen Instandhaltungssystem – Externe Dienste	90
Tabelle 23: Anforderungen Instandhaltungssystem – Anforderungen Systemadministration.	91
Tabelle 24: Anforderungen an ein Instandhaltungssystem – Übersicht der häufigsten Anforderungen (Angabe ab 10 Nennungen)	92
Tabelle 25: Aspekte der Instandhaltung in den untersuchten Unternehmen.....	95
Tabelle 26: Aufgenommene Anforderungen in den untersuchten Unternehmen	98
Tabelle 27: Generische Anforderungen	100

Tabellenverzeichnis	XII
Tabelle 28: Optionale Anforderungen.....	101
Tabelle 29: Priorisierung der generischen Anforderungen	103
Tabelle 30: Generische Anforderungen in Verbindung mit den Anforderungen der Literaturrecherche	106
Tabelle 31: Spezifische Anforderungen im Vergleich mit den Anforderungen aus der Literaturrecherche	109
Tabelle 32 : Tabellarischer Aufbau eines Moduls der RAII	123
Tabelle 33:Kategorisierte Übersicht von Evaluationsmethoden	228
Tabelle 34: Anforderungen an die RAII	239
Tabelle 35: Darstellung der Managementfunktionen des Ressourcen-Cockpits	248
Tabelle 36: Komponentenüberprüfung Ressourcen-Cockpit.....	254
Tabelle 37: Schritte 1-5 des allgemeinen Ablaufmodells nach Mayring angewandt auf die Evaluation der RAII	259
Tabelle 38:Übersicht Kodierung mit Ankerbeispiel und Kodierregeln	261
Tabelle 39: Elemente einer Designtheorie	273
Tabelle 40: Priorisierung von Anforderungen	297
Tabelle 41: Fragebogen Anforderungserhebung.....	301
Tabelle 42: Leitfaden Telefoninterview	303

Abkürzungsverzeichnis

AR	Augmented Reality
ASP	Active Server Pages
BCM	Business Centered Maintenance
CAD	Computer-Aided Design
CBM	Condition Based Maintenance
CBR	Case-Based Reasoning (fallbasiertes Schließen)
CMMS	Computerized Maintenance Management System
CMS	Condition Monitoring System
CPS	Cyber-physische Systeme
CRM	Customer Relationship Management
CRMS	Customer Relationship Managementsystem
CSS	Cascading Style Sheets
DMS	Dokumentenmanagementsystem
DSS	Decision Support System
EDM	Engineering Data Management
eETK	Elektronische Ersatzteilkataloge
E-MS	E-Maintenance System
ERP	Enterprise Resource Planning
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Mode and Effects and Criticality Analysis
HTML	Hypertext Markup Language
HW/SW	Hardware/Software
IIS	Integriertes Instandhaltungssystem (IIS)
IKT	Informations- und Kommunikationstechnik
IMS	Intelligent Maintenance System
IPS	Instandhaltungsplanungs- und -steuerungssysteme
JSON	JavaScript Object Notation
KMS	Knowledge Management System
LCC	Life Cycle Costing
LDAP	Lightweight Directory Access Protocol

MES	Manufacturing Execution System
MQTT	Message Queue Telemetry Transport
MSS	Mobile Support System
MTBF	Mean Time between Failures
MTTR	Mean Time to Repair
OEE	Overall Equipment Effectiveness
OPC UA	Open Platform Communications Unified Architecture
PDM	Produktdatenmanagementsystem
PDF	Portable Document Format
PIS	Performance Information System
PLC	Programmable Logic Controller
PLMS	Product Lifecycle Management System
PMS	Predictive Maintenance System
PSS	Product Service System
RAII	Referenzarchitektur eines integrierten Informationssystems zur Unterstützung der Instandhaltung
RBCM	Risk Based Centered Maintenance
RCM	Reliability Centered maintenance
RDF	Resource Description Framework
SCADA	Supervisory Control and Data Acquisition
SP	Serviceplattform
SPS	Speicherprogrammierbare Steuerung
TKD	Technischer Kundendienst
TPM	Total Productive Maintenance
VDM	Value Driven Maintenance
WMS	Wissensmanagementsystem
WYSIWYG	What You See Is What You Get
XML	Extensible Markup Language

1 Einleitung

1.1 Motivation und Relevanz

Die fortschreitende Digitalisierung führt nicht nur zu völlig neuen Unternehmen, sondern stellt auch traditionelle, etablierte Unternehmen vor große Herausforderungen, da auch sie einer fundamentalen Veränderung unterliegen (Horváth, 2017, S. 113). Besonders betroffen ist das in Deutschland stark vertretene produzierende Gewerbe. Als Treiber der Digitalisierung wirken hier verbundene Trends wie die digitale Fabrik (Himmler/Amberg, 2013), Cyber-physische Systeme (CPS) (Lee et al., 2015) das Internet der Dinge (Gubbi et al., 2013), Industrie 4.0 (Lachenmaier et al., 2015) oder dessen Pendant, das Industrial Internet (Lin et al., 2015). Dies hat einerseits zur Folge, dass bestehende Prozesse der produzierenden Industrie einem starken rein technologischen Wandel unterliegen, andererseits werden Änderungen am bisher betriebenen Geschäftsmodell durch die aktuellen Technologien nicht nur möglich, sondern auch nötig und zwingen Unternehmen, sich für einen disruptiven Wandel ihres Geschäftes vorzubereiten (Reidt et al., 2017a, S. 23). Im Bereich der produzierenden Unternehmen können diese Effekte darin erkannt werden, dass diese sich zunehmend von ursprünglich rein produzierenden Unternehmen zu produzierenden Serviceanbietern wandeln (Daeuble et al., 2015b, S. 1018). Damit verbunden sind insbesondere Auswirkungen auf unterstützende Prozesse wie der Instandhaltung, welche sich mit enorm erhöhter Komplexität, aber auch Bedeutung auseinandersetzen muss (Reidt et al., 2016a, S. 1). Die Abgrenzung von einst wertschöpfenden und unterstützenden Prozessen fällt im Zuge der Digitalisierung daher immer schwerer (Bley/Leyh, 2016, S. 1651). Unter Instandhaltung wird in dieser Arbeit nach der Definition der DIN die „Kombination aller technischen und administrativen Maßnahmen sowie Maßnahmen des Managements während des Lebenszyklus einer Betrachtungseinheit zur Erhaltung des funktionsfähigen Zustandes oder der Rückführung in diesen, so dass sie die geforderte Funktion erfüllen kann“ (DIN 31051:2012-09, S. 6) verstanden.

Die zunehmende Komplexität und Bedeutung resultiert dabei aus vornehmlich zwei Aspekten: Zum einen wird aufgrund der angeführten zunehmenden Integration von fortschrittlicher Informationstechnologie in Produktionsanlagen und Maschinen die Instandhaltung und damit der Anspruch an die handelnden Personen fortlaufend herausfordernder (Reidt/Krcmar, 2016, S. 377). Die Fehleranalyse und -behebung erfordert ein zunehmend tieferes Sachverständnis des Aufbaus von Maschinen und der dahinterliegenden Systeme. Dies trifft insbesondere dann zu, wenn einzelne Instandhalter diverse Maschinen von unterschiedlichen Herstellern betreuen. Hier kommt erschwerend hinzu, dass diese sich zum Teil stark bei den verwendeten Technologien unterscheiden. Zum anderen wird eine technisch, organisatorisch anspruchsvollere, vorrausschauende und nicht reaktive und damit komplexere Instandhaltung benötigt, die die Maschinenverfügbarkeit erhöht, um die Umsetzung von neuen, lukrativen Geschäftsmodellen zu ermöglichen (Huber/Kaiser, 2015, S. 684).

Neben den steigenden Anforderungen durch den technologischen Fortschritt ermöglicht dieser ebenfalls, dass die Instandhaltung durch effiziente Informationssysteme Unterstützung erhält. Neben häufig vorhandenen Steuerungs- und Planungssystemen wie dem Computerized

Maintenance Management System (CMMS)¹ (Bohoris et al., 1995; Gabbar et al., 2003), kann der Instandhalter u. a. bei der Fehlersuche durch ein mobiles Assistenzsystem oder auch Mobile Support Systems (MSS)² die nötigen Informationen zur Arbeitsbewältigung aggregiert dargestellt bekommen (Daeuble et al., 2015a; Fellmann et al., 2013). Diese Informationen können weiterhin durch Augmented Reality (AR) (Emmanouilidis et al., 2011) vereinfacht dargestellt werden. Wartungspläne können durch Condition Monitoring Systeme (CMS)³ in Abhängigkeit von der aktuellen Abnutzung bestimmter Komponenten berechnet (Abdennadher et al., 2010) oder bevorstehende Ausfälle durch die Anwendung von Data Mining Techniken im Rahmen von Predictive Maintenance⁴ erkannt werden (Lee et al., 2009).

Die Einbindung dieser Informationssysteme in unternehmensspezifische Produktionsprozesse und die Bündelung von Informationen aus mehreren Quellen und Informationssystemen ist jedoch oft unzureichend (Bienzeisler et al., 2014, S. 9). Sofern Informationssysteme für die Instandhaltung existieren, wie bspw. CMS und CMMS, so existieren oftmals isolierte Dateninseln der einzelnen Systeme, ohne dass diese miteinander kombiniert werden können (Galar, 2014, S. 113). Dies führt dazu, dass Daten zur Fehleranalyse und -vorhersage oft zerstreut zwischen unabhängigen Systemen vorzufinden sind, wodurch sie schwierig zu erreichen und zu verbinden bzw. korrelieren sind (Galar, 2014, S. 113). Erschwerend kommt in manchen Fällen hinzu, dass an der Instandhaltung mehrere Unternehmen beteiligt sein können, z. B. wenn der Hersteller der Maschinen die Instandhaltung übernimmt oder andere externe Dienstleister Instandhaltung als Service anbieten. Diese einzelnen Unternehmen haben wiederum verschiedene Informationssysteme im Einsatz und besitzen oftmals keinen Zugriff auf Informationssysteme der Kunden und umgekehrt. Als Folge sind relevante Daten von potentiell vorhandenen betrieblichen Informationssystemen und Maschinen für den Instandhalter nicht oder nur eingeschränkt verfügbar (Trommler et al., 2014, S. 51). In der Konsequenz heißt dies, dass die Instandhalter oftmals nicht adäquat durch Informationssysteme unterstützt werden.

In der in einem späteren Kapitel vorgestellten Analyse der Instandhaltung bei vier Unternehmen bestätigt sich dieser Sachverhalt: In den untersuchten Unternehmen müssen bspw. Daten aus einer Vielzahl an Informationssystemen oder analogen Quellen manuell gesucht und extrahiert werden. Dieser Umstand führt bei der Arbeit eines Instandhalters zu hohen Wartezeiten, erhöhter Fehleranfälligkeit und in letzter Konsequenz längeren Stillstandszeiten.

¹ Computerized Maintenance Management System (CMMS) werden im deutschen Sprachraum auch „Instandhaltungsplanungs- und -steuerungssysteme“ (IPS) (Reichel et al., 2009, S. 153) genannt. In dieser Arbeit wird aufgrund der größeren Verbreitung des englischen Begriffs dieser bzw. dessen Abkürzung CMMS verwendet.

² Mobile Support Systems (MSS) werden im deutschen Sprachraum auch „mobile Assistenzsysteme“ (Güntner et al., 2015, S. 42) genannt. In dieser Arbeit wird aufgrund der größeren Verbreitung des englischen Begriffs dieser bzw. dessen Abkürzung MSS verwendet.

³ Condition Monitoring Systeme (CMS) werden im deutschen Sprachraum auch „Zustandsüberwachungssysteme“ (Pawellek, 2016, S. 64) genannt. In dieser Arbeit wird aufgrund der größeren Verbreitung des englischen Begriffs dieser bzw. dessen Abkürzung CMS verwendet.

⁴ Techniken des Predictive Maintenance werden im deutschen Sprachraum mitunter als „vorausschauende Wartung“ (Fisseler/Reiners, 2015, S. 917), „vorausschauende Instandhaltung“ (Roth, 2016, S. 265). Predictive Maintenance wird in dieser Arbeit als eigenständiger Begriff angesehen und auch aufgrund der größeren Verbreitung des englischen Begriffs dieser verwendet.

Die Gründe hierfür sind vielfältig: Die Entwicklungskosten für zentrale Informationssysteme zur Unterstützung der Instandhaltung sind sehr hoch, da die Integration von Daten aus heterogenen, oft proprietären Systemen mit hohem manuellen Aufwand verbunden ist. CMS sind bspw. oft nur bei einigen Maschinenherstellern in teils einfacher Art und Weise vorhanden und hauptsächlich für die eigenen Maschinen verfügbar (Reidt/Krcmar, 2016, S. 378). Diese entwickeln die CMS mit jeweils unterschiedlichen Datenprotokollen, Übertragungsmechanismen und zusätzlich meist noch unterschiedlichen informationstechnischen Konzepten (Winter/Wollschlaeger, 2015, S. 1). Des Weiteren werden die Informationssysteme auch sehr kleinteilig entwickelt. Wechselwirkungen mit anderen Informationssystemen oder die potentielle Integration von anderen Funktionalitäten werden nicht betrachtet oder deren Anforderungen sind nicht bekannt.

Ein von Anfang an auf Erweiterbarkeit und Anpassbarkeit bedachtes integriertes Instandhaltungssystem (IIS), welches system- und endgeräteübergreifend modular genutzt werden kann, stellt eine Möglichkeit zur Lösung der angesprochenen Herausforderungen dar. Unter Instandhaltungssystem wird in dieser Arbeit ein Informationssystem zur Unterstützung der Instandhaltung verstanden. Durch die Verwendung aktueller Technologien zur Integration und Interpretation von Daten und Informationen aus einer Vielzahl an Informationssystemen können Informationen zielgerichtet gebündelt und ausgewertet werden. Durch die Ausnutzung und Integration der Möglichkeiten eines CPS von mobilen Endgeräten und eines darauf dynamisch reagierenden Backends können Inhalte kontext-sensitiv dezentral bereitgestellt werden. In Kombination mit neuen, sich verbreitenden Standards in der Produktion, wie OPC UA (Enste/Mahnke, 2011), ist es möglich, dezentral Informationen von Maschinen mit deutlich weniger Aufwand zu extrahieren. Dies ermöglicht Funktionen leichter über verschiedene Informationssysteme zu verteilen und diese ohne größeren Aufwand in bestehende Systeme der Produktion einzubinden. Aufgrund dieser dezentralen losen Kopplung einzelner Komponenten und der inhärenten Möglichkeit zur Erweiterbarkeit kann leichter auf zukünftige Entwicklungen reagiert werden.

Um jedoch ein solches Informationssystem entwickeln zu können bedarf es einer umfassenden Sicht auf die zu entwickelnde Architektur, da sie eine große Bandbreite an Funktionalitäten, Systemen und Schnittstellen unterstützen muss oder durch diese erweitert werden kann. Dabei besteht zudem die Herausforderung, dass die an der Entwicklung beteiligten Personen und späteren Nutzer aus unterschiedlichen Domänen stammen und unterschiedliche Rollen im Unternehmen begleiten (bspw. aus den Domänen der Anwendungsentwicklung, Produktion und Instandhaltung sowie aus den Rollen der Manager, Controller und Instandhalter). Dies führt zu einer unterschiedlichen Sichtweise auf die spätere Nutzung und den Nutzen des zu entwickelnden Informationssystems sowie zu stark individuellen Anforderungen, deren Erhebung oftmals Fachexperten der jeweiligen Domäne bedarf.

Die Identifizierung von individuellen Anforderungen an ein umfassendes Instandhaltungssystem und die Entwicklung der verbundenen Softwarearchitekturen selbst ist daher aufwändig und erfordert erhebliche Ressourcen. Insbesondere kleine und mittelständische Unternehmen können aufgrund dieses hohen Aufwandes kaum Informationssysteme entwickeln, welche deren individuellen Anforderungen genügen, noch können alle Anforderungen vorab erhoben werden. Dies liegt auch daran, dass mangelndes Know-how bei Beurteilung der technologischen Potentiale von Instandhaltungslösungen ein großes Hemmnis darstellt, die verfügbaren Technologien effizient einzusetzen (Bienzeisler et al., 2014, S. 7). Dies spielt insbesondere eine große Rolle, da ein Instandhaltungssystem für

die Ausnutzung neuer Technologien Zugriff auf alle verfügbaren Daten erlangen muss und somit Abhängigkeiten von nahezu allen vorhandenen Systemen und Sensoren der Produktion besitzt. Hierzu ist ein Verständnis von der möglichen dezentralen Intelligenzverteilung von Anlagensystemen, deren Standards bis hin zur Darstellung von kontextsensitiven Informationen auf mobilen Endgeräten nötig. Denn eine dafür nötige integrierte Sichtweise ist oftmals nur durch eine domänenübergreifende Sichtweise möglich. Weiterhin ist ein uneinheitliches Verständnis über die Funktionsweise und Schnittstellen eines solchen Instandhaltungssystem ein Faktor, der zu Kompatibilitätsproblemen mit anderen Informationssystemen führen kann und somit eine Weiterentwicklung hemmt.

Richtlinien, Blaupausen und wiederverwendbare Architekturmuster für die Erstellung würden die Entwicklung von solchen Informationssystemarchitekturen erleichtern. Ein probates Mittel, um diese Punkte zu adressieren, ist die Erstellung von Referenzarchitekturen (Muller, 2007, S. 2–3). Referenzarchitekturen stellen spezielle, abstrakte Arten von Softwarearchitekturen dar, welche allgemeinen Richtlinien zur Spezifikation von konkreten Softwarearchitekturen einer Klasse von Systemen setzen (Angelov et al., 2008, S. 226). Sie erleichtern dadurch den Softwareentwicklungsprozess (Angelov/Grefen, 2008, S. 1817; Nakagawa et al., 2014a, S. 143), besonders wenn verschiedene Anbieter, Lokationen und Unternehmen an diesen Systemen partizipieren müssen (Muller, 2008, S. 1). Den Prozess erleichtern sie dadurch, dass die daran Beteiligten sich an den Richtlinien orientieren können, Teile der Referenzarchitektur wiederverwendet werden und anhand abstrakter Artefakte der Referenzarchitektur zielgerichteter diskutiert werden können. Dies fördert insbesondere die domänenübergreifende Kommunikation mit potentiellen Stakeholdern (Angelov/Grefen, 2008, S. 1817) und die zunehmende Integration von Systemen (Muller, 2008, S. 1). Diese Gründe sind insbesondere dafür ausschlaggebend, warum Referenzarchitekturen so populär im Kontext von Industrie 4.0 und der Digitalisierung geworden sind und damit besonders für IIS eine Grundlage zur Erleichterung der Entwicklung darstellen.

Daher wird in diesem Beitrag eine Referenzarchitektur für ein integratives Informationssystem zur Unterstützung der Instandhaltung (RAII) präsentiert, um die Entwicklung für diese Klasse von Informationssystemen durch die Festlegung von allgemeinen Richtlinien zu erleichtern. Diese Richtlinien werden in dem vorliegenden Fall dadurch gesetzt, dass in abstrakter Weise dargestellt wird, welche Funktionen, Prozesse, Konzepte und damit verbunden, welche Intelligenzverteilung zwischen verschiedenen Systemen durch effiziente Ausnutzung der Fähigkeiten eines CPS und mobilen Endgeräten abgebildet werden können. Durch eine einfache und intuitive Darstellungsweise und eine einheitliche Domänensprache soll zudem die domänenübergreifende Zusammenarbeit gefördert werden.

1.2 Forschungsleitende Fragestellungen

In den letzten Jahren nimmt die Veröffentlichung von Referenzarchitekturen insbesondere im Bereich von Trends wie Industrie 4.0 und Digitalisierung zu (Adolphs et al., 2015; Barkmeyer/Wallace, 2016; Lin et al., 2015; Usländer/Epple, 2015). Dies liegt zum einen an einem immer größer werdenden Bedarf an Interoperabilität zwischen Komponenten und Systemen, da monolithische Architekturen zunehmend von cloudbasierten Micro-Service-Architekturen oder früher schon SOA-Ansätzen verdrängt werden. Zum anderen dringen immer komplexer werdende Informationssysteme im Kontext der Industrie 4.0 in Bereiche vor, die sich durch eine niedrige IT-Affinität auszeichnen. Dies stellt die Entwickler, aber auch die Nutzer, vor große Herausforderungen. Eine domänenübergreifende Zusammenarbeit bei der Erstellung solcher Informationssysteme ist daher unerlässlich und diese kann, wie auch die Interoperabilität, durch eine Referenzarchitektur gefördert werden.

Trotz dieser Entwicklung fehlt es an Referenzarchitekturen, die eine umfassende Entwicklung der Instandhaltung und den damit verbundenen Informationssystemen zur Unterstützung der Instandhaltung behandeln. Unternehmen mit dem Bedürfnis, die Instandhaltung durch Digitalisierung zu optimieren, würden Referenzarchitekturen darin unterstützen, sich mit den Anforderungen der Digitalisierung und den daraus entstehenden Möglichkeiten im Bereich der Instandhaltung zu beschäftigen. Referenzarchitekturen existieren nur für Teilbereiche wie CMS (Wollschlaeger et al., 2015) und besitzen somit keinen Fokus auf einer umfassenden Sichtweise für IIS. Da der Bedarf in der Praxis an solchen Systemen zweifellos vorhanden ist und in der Literatur die Problematik der Integration und die Verbindung zwischen Systemen im Kontext der Instandhaltung als besonders kritisch eingeschätzt wird, sind dies weitere Gründe für die Erstellung einer Referenzarchitektur zur Unterstützung der Entwicklung von integrierten Informationssystemen für die Instandhaltung.

Die Entwicklung einer solchen Referenzarchitektur, die an der Schnittstelle von Fachverantwortlichen und Entwicklern eingesetzt wird und deren Kommunikation erleichtern soll, umfasst mehrere zu erforschende Aspekte: Es bedarf einer Untersuchung hinsichtlich der Domäne der Referenzarchitekturen und der Instandhaltung, um die nötige inhaltliche Basis für den Entwurf der Referenzarchitektur zu erlangen. Dazu gehören ableitend Anforderungen an ein IIS, potentielle Entwicklungspfade dieses Informationssystems, mögliche Teilsysteme und bestehende Informationssysteme in der Produktion und Instandhaltung. Weiterhin müssen die nötigen Elemente und die Darstellung der Referenzarchitektur bestimmt und in eine abstrakte, technische Architektur übersetzt werden. Diese Architektur muss das grundlegende Ziel erfüllen, die Entwicklung von IIS zu erleichtern. Diese Aspekte lassen sich in die folgenden drei Forschungsfragen überführen:

Forschungsfrage 1: Welche Anforderungen an Informationssysteme zur Unterstützung der Instandhaltung existieren in Literatur und Praxis?

Zur Erstellung einer Referenzarchitektur, welche ein IIS umfassen soll, ist es vonnöten, möglichst allgemeingültige Anforderungen aus verschiedenen Perspektiven an diese aus Literatur und Praxis zu bestimmen. Dazu müssen aktuelle Informationssysteme, welche für die Instandhaltung eine Rolle spielen, gesammelt und hinsichtlich Zweck und Nutzen untersucht werden. Zwei Arten von Informationssystemen sind für diese Untersuchung relevant: Zum

einen Informationssysteme, aus denen Informationen für die Instandhaltung benötigt werden, jedoch nicht speziell für die Instandhaltung ausgerichtet sind. Zum anderen Informationssysteme, die direkt zur Unterstützung der Instandhaltung entwickelt worden sind.

Anforderungen und Funktionen aus den jeweiligen Informationssystemen sollen gesammelt und auf eine einheitliche Basis gesetzt werden, sodass ein umfassendes Bild über Anforderungen für Instandhaltungssysteme entsteht. Diese Informationen bilden die Basis für die spätere Erstellung der Referenzarchitektur, da diese die elementaren Anforderungen und Eigenschaften von bestehenden Informationssystemen adressiert.

Um diese Anforderungen aufzunehmen, wird eine Literaturanalyse hinsichtlich Anforderungen für verschiedene Instandhaltungssysteme durchgeführt und Anforderungen für IIS in der Praxis bei vier Unternehmen aufgenommen. Die Unternehmen werden so ausgewählt, dass sie sich in der Ausführung der Instandhaltung stark unterscheiden und somit eine große Breite an Anforderungen aufweisen. Als Ergebnis wird eine Auflistung relevanter Instandhaltungssysteme, deren Anforderungen sowie eine Auswertung zu Anforderungen an IIS erstellt.

Forschungsfrage 2: Welche funktionellen Module werden für ein integriertes Informationssystem benötigt, das effektiv die Instandhaltung unterstützt?

Die Referenzarchitektur muss möglichst verallgemeinerbare Grundelemente und -konzepte eines IIS abdecken. Diese bilden den generischen Kern der Referenzarchitektur ab, der bestimmt, welche Funktionen, Eigenschaften und Komponenten Teil der Softwarearchitektur eines IIS sein müssen. Mit Beantwortung dieser Frage soll daher geklärt werden, welche funktionalen Module und welche Entitäten (Mobiles Endgerät, Backend, Anlagen, Sensoren) die Referenzarchitektur umfassen muss, um die Instandhaltung effizient zu unterstützen. Darüber hinaus sollen optionale Elemente und die Erweiterung der Referenzarchitektur durch diese ermöglicht werden.

Um diese funktionalen Module zu bestimmen, werden die Ergebnisse aus der ersten Forschungsfrage genutzt, um aus ihnen generische und optionale Anforderungen an IIS abzuleiten. Im nächsten Schritt werden ähnliche Anforderungen gruppiert und gemäß technischer Umsetzbarkeit in funktionale Module überführt. Auf Basis dieser Anforderungen, Module und dem Wissen über aktuelle und bestehende Systeme werden die Entitäten festgelegt, auf denen ein IIS arbeiten muss. Dazu werden auch die Interaktion bzw. Interaktionsmuster zwischen den Modulen und den Entitäten untersucht, sodass eine effiziente Verteilung der funktionalen Komponenten zwischen verschiedenen Entitäten möglich ist.

Als Endergebnis werden generische und optionale Module beschrieben, welche mit den Anforderungen aus Forschungsfrage 1 verbunden sind und Entitäten zugeordnet werden können. Diese bilden die Grundlage zur Erschaffung und Darstellung einer Referenzarchitektur.

Forschungsfrage 3: Wie sieht eine Referenzarchitektur aus, die die Entwicklung eines integrierten Informationssystems zur Unterstützung der Instandhaltung erleichtert?

In dieser Forschungsfrage soll eine Referenzarchitektur basierend auf den Erkenntnissen aus Forschungsfrage 1 und 2 erstellt werden, welche die Entwicklung von umfassenden, IIS unterstützt. Zuerst wird dazu die aktuelle Forschung bzgl. Referenzarchitekturen untersucht, um die Grundlagen zur Erstellung ebenjener zu schaffen. Dies betrifft Eigenschaften, Klassifikationen sowie potentielle Methoden zur Erstellung sowie zur Darstellung, die der Zielerreichung dienlich sind. Anschließend wird die Referenzarchitektur auf Basis dieses Wissens und den Ergebnissen aus Forschungsfrage 1 und 2 so erstellt, dass sie die domänenübergreifende Zusammenarbeit fördert und ein klares Bild von dem zu entwickelnden System enthält. Abschließend wird das Artefakt der Referenzarchitektur dahingehend evaluiert, ob sie diese Aspekte und das Ziel der Erleichterung der Entwicklung von Informationssystemen zur Unterstützung der Instandhaltung erfüllt.

Die Referenzarchitektur wird nach dem Vorgehen von Reidt et al. (2017a) iterativ erstellt. Zur Evaluation der Referenzarchitektur werden drei Arten der Evaluation durchgeführt: eine analytische, merkmalsbasierte Evaluation, eine Evaluation anhand der Entwicklung eines Prototypens auf Basis der Referenzarchitektur und eine Evaluation anhand von Experteninterviews.

Das Ergebnis wird eine evaluierte, auf Basis des Feedbacks angepasste Referenzarchitektur sein, deren Nützlichkeit bei der Erleichterung der Entwicklung von IIS nachgewiesen wird.

1.3 Wissenstheoretische Grundlagen

Damit Ergebnisse einer wissenschaftlichen Forschungsarbeit intersubjektiv nachvollziehbar sind, haben sich in den verschiedenen wissenschaftlichen Disziplinen meist Forschungsdesigns und Forschungsmethoden entwickelt, an denen sich Forscher ausrichten können (Tschamler, 1996, S. 67). Diese unterscheiden sich jedoch nach der wissenstheoretischen Grundposition, die die jeweilige Wissenschaftsdisziplin vornehmlich einnimmt. So konzentriert sich die Naturwissenschaft vornehmlich darauf, einen Gegenstand zu erklären und damit auch eine Ursache-Wirkungsbeziehung zu erläutern. Wohingegen sich die Geisteswissenschaften mit dem Verstehen eines Gegenstandes befassen (Tschamler, 1996, S. 22). Dieser Hintergrund beeinflusst den Forscher demzufolge bei der Auswahl der Forschungsmethoden und somit die jeweiligen Ergebnisse der Arbeit. Daher wird die vorliegende Dissertation nachfolgend in die Wissenschaftsdomäne der Wirtschaftsinformatik eingeordnet und das verwendete Forschungsdesign vorgestellt.

1.3.1 Einordnung in die Wirtschaftsinformatik

Die Wirtschaftsinformatik beschäftigt sich im Kern mit der Untersuchung der Entwicklung, Nutzung und Wartung betrieblicher Informationssysteme (Becker et al., 2003, S. 3; Frank, 2007, S. 155). Informationssysteme dienen dabei nicht dem Selbstzweck, sondern als Mittel zur Erreichung von Unternehmenszielen (Frank, 2007, S. 155). Daher wird die Wirtschaftsinformatik als anwendungsorientierte Disziplin beschrieben, deren Ergebnisse in die Praxis nutzbringend transferiert werden sollten (Frank, 2007, S. 155). Ihre Ursprünge hat die Wirtschaftsinformatik in der Informatik und den Wirtschaftswissenschaften (Becker et al., 2003, S. 3) und fungiert als Subdisziplin thematisch auch als Schnittstelle zwischen diesen Disziplinen (Winter, 2009, S. 223). Dieser Ursprung beeinflusst auch die Forschungsmethoden, die in der Wirtschaftsinformatik Verwendung finden (Heinrich, 2005, S. 106).

Die Entwicklung der Referenzarchitektur eines integrierten Informationssystems zur Unterstützung der Instandhaltung ist nach dieser Beschreibung thematisch dem Bereich der Wirtschaftsinformatik zuzuordnen. Die Referenzarchitektur soll die Entwicklung von betrieblichen Informationssystemen (hier in Form von Informationssystemen zur Unterstützung der Instandhaltung) erleichtern und ist aufgrund der hohen Praxisrelevanz ein Problem der angewandten Forschung. Neben den technischen Aspekten bzgl. Softwarearchitektur, Instandhaltungsstandards und verfügbaren Informationssystemen, spielt die Dienstleistungs- und Geschäftsmodellforschung hinsichtlich der Entwicklung der Modelle der Instandhaltung eine große Rolle bei der Entwicklung der Referenzarchitektur und damit verbunden auch die Veränderung der dazu nötigen betrieblichen Informationssysteme. Zusätzlich ist der daraus folgende Aspekt der domänenübergreifenden Zusammenarbeit zwischen Personen aus den (betriebswirtschaftlichen) Fachbereichen, der IT und des Maschinenbaus von hoher Relevanz für die spätere Nützlichkeit der erstellten Lösungen. In diesem Spannungsfeld zwischen technischen und betrieblichen Fragestellungen ist die Verbindung der Arbeit mit der Wirtschaftsinformatik zusätzlich begründet.

1.3.2 Forschungsdesign

Im Bereich der Wirtschaftsinformatik kann nach Hevner et al. (2004, S. 75) grundlegend zwischen zwei Forschungsparadigmen unterschieden werden: Zwischen dem behavioristischen

Paradigma, das auch als verhaltenswissenschaftliches Paradigma (engl. „Behavioral Science“) bezeichnet wird, und dem gestaltungsorientierten Paradigma (engl. „Design Science“).

Der behavioristische, auch als Behaviorismus bezeichnete Forschungsansatz ist vor allem im angelsächsischen Raum und der dort vorherrschenden Information Systems Research verbreitet (Österle et al., 2010, S. 1). Die Arbeiten, die diesem Ansatz folgen, widmen sich vor allem der „Beobachtung von Eigenschaften von Informationssystemen und des Verhaltens von Benutzern“ (Österle et al., 2010, S. 1) und damit der (reaktiven) „Analyse von Ausgestaltung und Wirkung verfügbarer IT-Lösungen auf Unternehmen und Märkte“ (Wilde/Hess, 2007, S. 281). Ziel dieser Arbeiten ist die Entwicklung und Verifikation von Theorien, die menschliches und organisatorisches Verhalten in Bezug auf Informationssysteme erklären und vorhersagen (Hevner et al., 2004, S. 76).

Das gestaltungsorientierte Forschungsparadigma hingegen zielt darauf ab, die Grenzen von organisatorischen und menschlichen Fähigkeiten zu erweitern, indem neue und innovative IT-Artefakte erstellt werden (Hevner et al., 2004, S. 75). Diese Artefakte sollen wichtige bestehende und zukünftige praxisrelevante Probleme lösen und dahingehend evaluiert werden, ob sie Nutzen- bzw. Relevanz-, Qualitäts- und Effizienzkriterien erfüllen (Bichler, 2006, S. 134). Daneben muss das erzeugte Artefakt einen signifikanten, verifizierbaren Beitrag zur Forschung hinsichtlich Neuartigkeit, Verallgemeinerbarkeit und Bedeutung darstellen, wobei mindestens eine dieser Eigenschaften erfüllt sein muss (Hevner et al., 2004, S. 87). Damit ist neben der Entwicklung die Evaluation ein zentrales Element der gestaltungsorientierten Forschung (Hevner, 2007, S. 90f), welche idealerweise in mehreren Evaluationsphasen zyklisch durchlaufen wird, um die Relevanz des Artefakts hinsichtlich der Lösung des Problems nachzuweisen. Diese Überprüfung der geschaffenen Artefakte muss anhand in einem Forschungsplan festgelegter Methoden gegen vorab definierte Ziele erfolgen, um zusätzlich eine Rigorosität nachzuweisen (Becker, 2010, S. 16). Zusammenfassend ist für die gestaltungsorientierte Forschung ein stringentes Forschungsvorgehen auf Basis wissenschaftlicher Methoden und ein fundiertes Wissen aus der zu erforschenden Domäne nötig. Methoden und Wissen bilden dabei die nach Hevner sogenannte Wissensbasis (Hevner, 2007, S. 89). Ergebnistypen dieser Art von Forschungsarbeiten sind nach Österle et al. (2010, S. 4) „Konstrukte (Konzepte, Terminologien, Sprachen), Modelle, Methoden und Instanzen (Implementierung konkreter Lösungen als Prototypen oder produktive Informationssysteme)“.

Diese Zusammenhänge werden durch das Rahmenkonzept der gestaltungsorientierten Forschung von Hevner illustriert (Hevner et al., 2004, S. 80), welches durch Hevner (2007, S. 88) um drei inhärente Zyklen erweitert wurde und in Abbildung 1 dargestellt ist. Der Relevanzzyklus verbindet die Umwelt, bestehend aus der Anwendungsdomäne mitsamt den dazugehörigen Personen, organisationalen und technischen Systemen sowie den dort vorherrschenden Problemen und Möglichkeiten, mit den Aktivitäten der gestaltungsorientierten Forschung (Hevner/Chatterjee, 2010, S. 17ff). Dieser Zyklus liefert Anforderungen aus der Praxis an das Artefakt, durch deren Erfüllung der Nutzen des Artefakts sichergestellt werden kann und ermöglicht durch Feldtests den aktuellen Stand des Artefakts zu testen und Feedback einzuholen. Der Rigor Zyklus wiederum verbindet die Aktivitäten der gestaltungsorientierten Forschung mit der sogenannten Wissensbasis. Diese Wissensbasis beinhaltet die Grundlagen hinsichtlich bestehender wissenschaftlicher Theorien und Methoden, Erfahrung und Expertise hinsichtlich des State of the Art der zu untersuchenden Anwendungsdomäne sowie das Wissen über bereits bestehende Artefakte und Prozesse der Anwendungsdomäne (Hevner/Chatterjee, 2010, S. 17ff). Der innere Design Zyklus ist das Kernstück der Frameworks von Hevner und

verbindet die Artefakterstellung mit der Evaluation. Der Zyklus wird iterativ durchlaufen und währenddessen werden solange Designalternativen erstellt und evaluiert, bis eine die Anforderungen zufriedenstellend erfüllt. Input hinsichtlich Anforderungen und Wissensbasis wird kontinuierlich durch die zwei anderen Zyklen geliefert. Die Durchführung des Design Zyklus läuft davon abgesehen weitestgehend unabhängig ab.

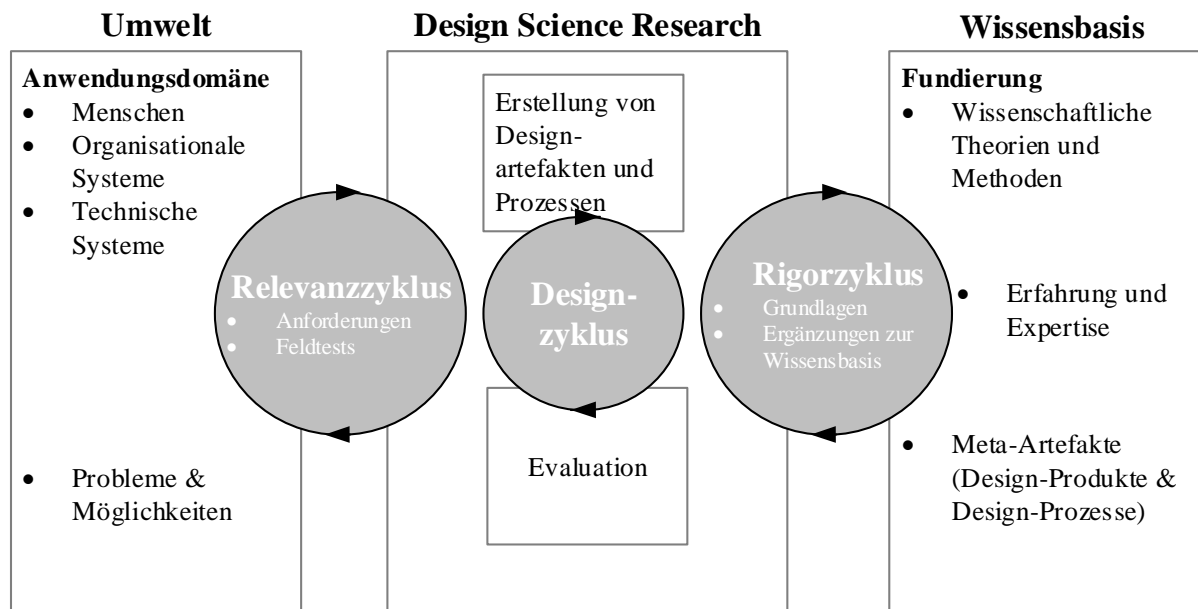


Abbildung 1: Design Science Rahmenkonzept mit drei Zyklen

Quelle: in Anlehnung an Hevner und Chatterjee (2010, S. 16)

Bezüglich der Forschungsparadigmen in der Wirtschaftsinformatik ist die Arbeit dem Bereich der gestaltungsorientierten Forschung zuzuordnen, da von einer konkreten Problemstellung (Erstellung von (integrierten) Instandhaltungssystemen) innovative Artefakte erstellt werden (Referenzarchitektur, generischen und optionale Anforderungen und Elemente der Architektur und domänenübergreifend verständliche Darstellung der Referenzarchitektur).

Die Referenzarchitektur wird anhand dieser iterativen Zyklen erstellt, als Rahmenkonzept zur Durchführung dieser Arbeit wird jedoch das Vorgehensmodell von Becker (2010) benutzt, welches die folgenden Phasen durchläuft:

- **Analysephase:** Diese Phase beschreibt bzw. formuliert Problemstellung und Forschungsziele. Die zu untersuchende Problemstellung muss dabei einem relevanten Problem zugeordnet werden können. Bei den Forschungszielen ist darauf zu achten, dass neben Gestaltungszielen häufig auch Beschreibungs- und Erklärungsziele zusätzlich verfolgt werden können. Weiterhin soll ein Forschungsplan mit den für die Entwicklung und Evaluation geeignetsten Forschungsmethoden erstellt werden.
- **Entwurfsphase:** In der Entwurfsphase wird das Artefakt anhand anerkannter Methoden entworfen, erstellt und gegen bekannte Lösungen abgegrenzt.
- **Evaluationsphase:** Um die Rigorosität zu erfüllen, wird das geschaffene Artefakt mit den in der Analysephase festgelegten Methoden gegen die auch dort festgelegten Ziele überprüft. Hier soll insbesondere der Nutzen, der durch das Artefakt hergestellt wird, dargestellt und begründet werden.
- **Diffusion:** In der Diffusionsphase werden die erzeugten Ergebnisse an die jeweiligen Stakeholder verbreitet. Dies sollte durch wissenschaftliche Publikationen oder

Praxisveröffentlichungen, Konferenzbeiträge, Dissertationen oder weiteren der Anspruchsgruppe gerechten Veröffentlichungen geschehen.

Die Ausprägung des Vorgehensmodells in dieser Arbeit und die verwendeten Methoden sind in Abbildung 2 dargestellt. Die Abbildung ist in drei Spalten unterteilt. Der linke Bereich stellt die Phasen nach Becker dar. In der mittleren Spalte werden die für jede Phase verwendeten Methoden genannt. Die rechte Spalte gibt an, welche Forschungsfrage durch die Ergebnisse welcher Phase beantwortet wird.

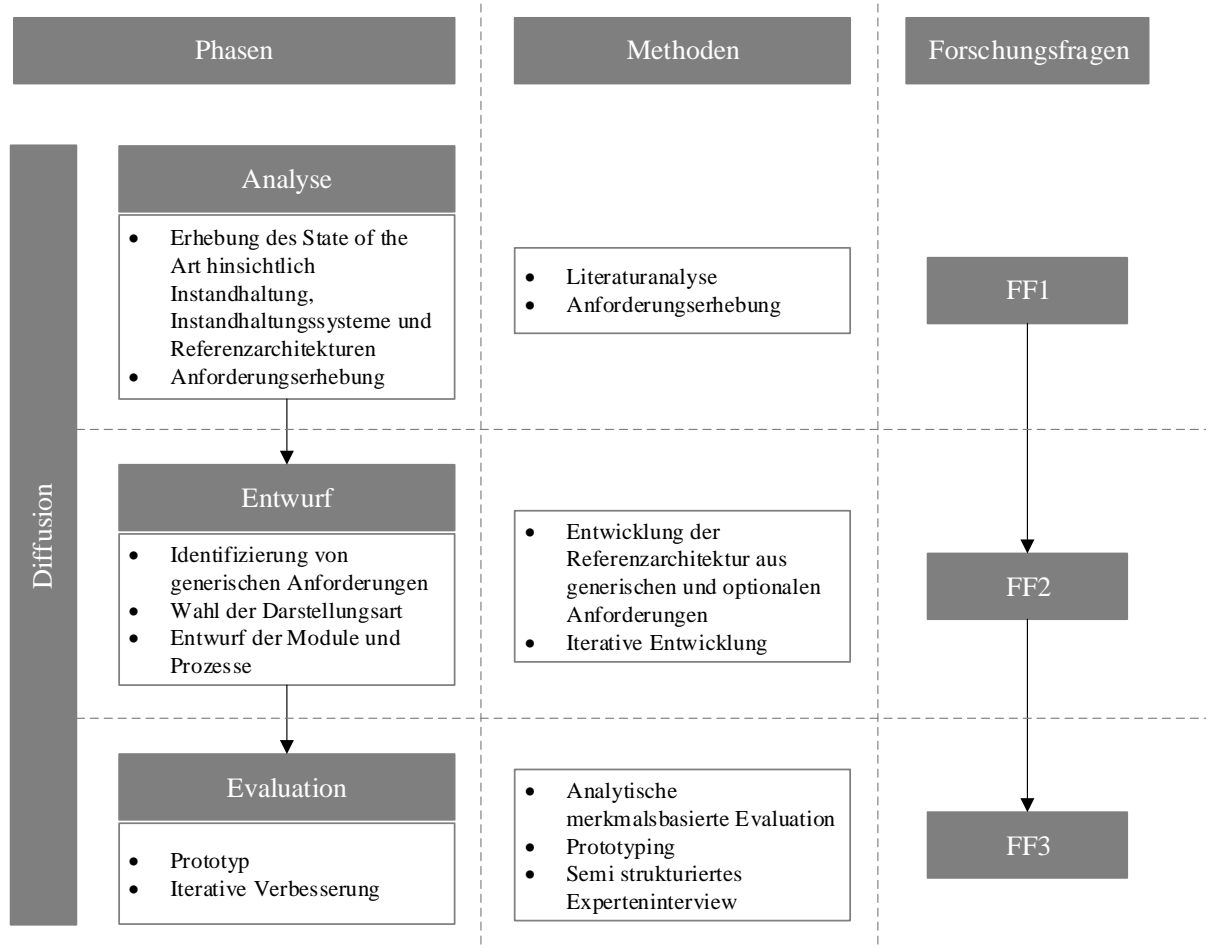


Abbildung 2: Forschungsvorgehen der vorliegenden Arbeit (FF = Forschungsfragen)

Quelle: Eigene Darstellung

Die Analysephase wird als erste durchlaufen. In dieser Phase wird die Erhebung des State of the Art hinsichtlich der Instandhaltung, Instandhaltungssysteme und Referenzarchitekturen durchgeführt. Daneben erfolgt eine Anforderungserhebung in Literatur und Praxis. Die verwendeten Methoden sind Literaturanalyse und eine Anforderungserhebung bei vier Unternehmen aus der Praxis. Mit dem erlangten Wissen kann die erste Forschungsfrage nach den Anforderungen an Instandhaltungssysteme in Literatur und Praxis beantwortet werden.

In der zweiten Entwurfsphase werden auf Basis der Ergebnisse von Forschungsfrage 1 die generischen und optionalen Anforderungen identifiziert, die ein IIS erfüllen muss bzw. kann. Die Darstellungsart und der Umfang der Referenzarchitektur wird festgelegt und die dazugehörigen Module und Prozesse definiert. Die Referenzarchitektur wird in dieser Phase erstellt. Die Entwicklung der Referenzarchitektur erfolgt iterativ mit kontinuierlichem Feedback. Mit den festgelegten Modulen und Prozessen wird die zweite Forschungsfrage

hinsichtlich der generischen und optionalen Module und deren Verbindung zueinander beantwortet.

In der dritten Phase wird die gesamte Referenzarchitektur dahingehend evaluiert, ob sie die Entwicklung von IIS erleichtert. Besonders hervorzuheben ist bei der Beantwortung dieser Frage, dass durch das Artefakt zum einen die domänenübergreifende Zusammenarbeit gefördert werden soll und zum anderen ein einfach zu kommunizierendes Bild des zu erstellenden Systems und dessen Architektur über den gesamten Entwicklungszyklus hinweg vermittelt werden soll. Die Evaluation erfolgt mit einem multimethodalen Ansatz: Zuerst wird eine analytische, merkmalsbasierte Evaluation anhand vorab definierter Anforderungen an die Referenzarchitektur als erste Evaluationsstufe durchlaufen. Als zweite Art der Evaluation wird anhand der Referenzarchitektur ein Prototyp erstellt, wodurch die Implementierungsfähigkeit der Referenzarchitektur nachgewiesen werden soll. Im dritten Teil der Evaluation werden semi-strukturierte Interviews mit Experten durchgeführt, bei denen diese über ihre Arbeit mit der Referenzarchitektur reflektieren und die Referenzarchitektur hinsichtlich ihrer Nützlichkeit für den Entwicklungsprozess bewerten.

Als Endergebnis soll einerseits eine anhand des Feedbacks verbesserte Referenzarchitektur für IIS stehen, die nachweislich die Entwicklung erleichtert. Andererseits soll auf Basis des State of the Art über Referenzarchitekturen aus Forschungsfrage 1 in Kombination mit der erschaffenen Referenzarchitektur Wissen über Definitionen, Methoden, Darstellungsformen und Klassifikationen für Referenzarchitekturen bereitgestellt werden, um die zukünftige Entwicklung von Referenzarchitekturen zu erleichtern. Mit diesen Ergebnissen wird die dritte Forschungsfrage beantwortet.

1.4 Aufbau der Arbeit

Ziel der vorliegenden Arbeit ist die Konzeption, Entwicklung und Evaluation der RAI. Die einzelnen Aspekte dieses Vorgehens und der jeweilige Inhalt werden in dieser Arbeit in acht Kapiteln strukturiert dargestellt. Der Aufbau der Arbeit ist in Abbildung 3 zusammengefasst dargestellt und untergliedert sich wie folgt:

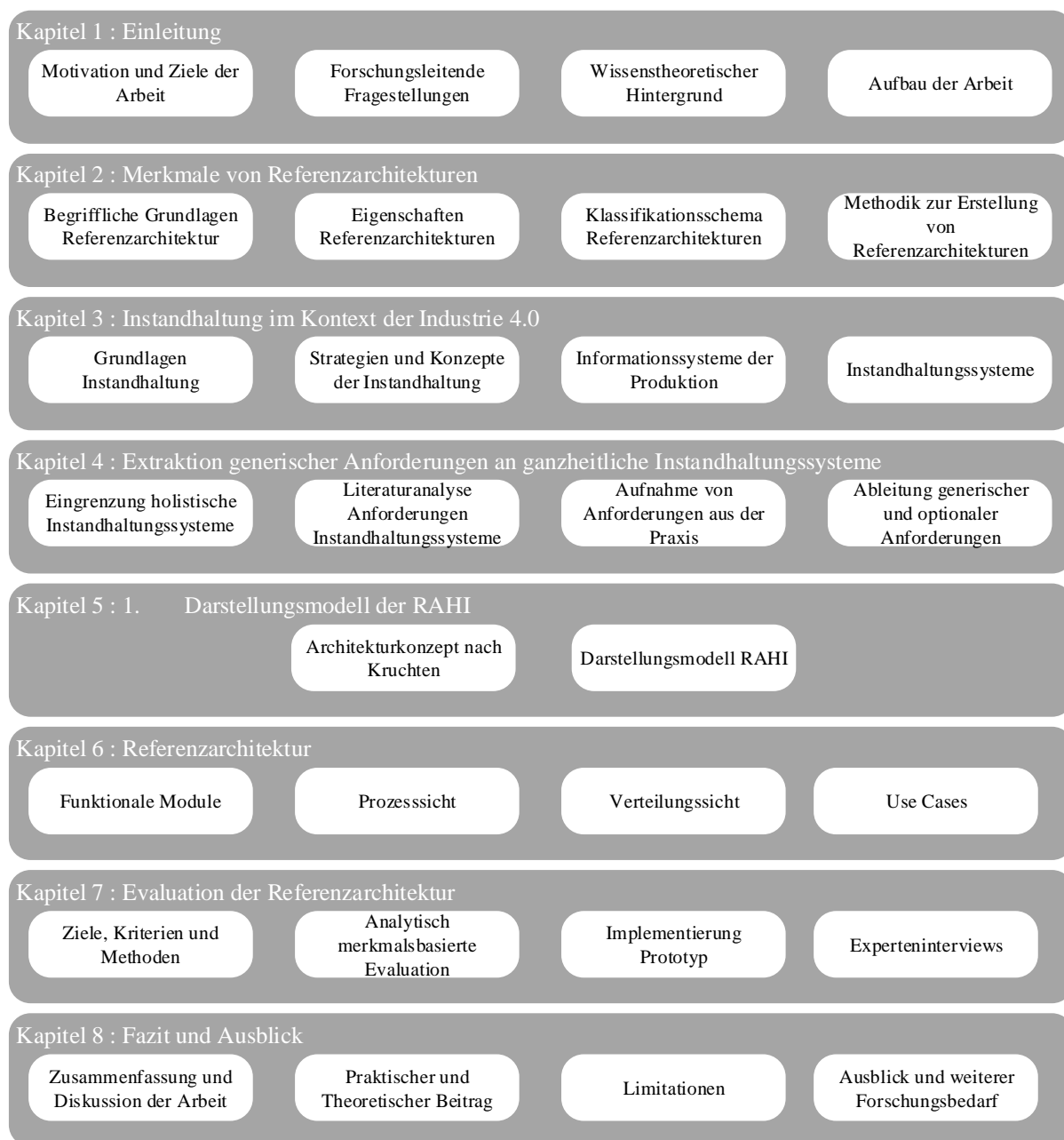


Abbildung 3: Aufbau der Arbeit

Quelle: Eigene Darstellung

Basierend auf der eingangs in **Kapitel 1** dargelegten Motivation, den forschungsleitenden Fragestellungen im Verbund mit dem wissenstheoretischen Hintergrund und Forschungsdesign der Arbeit, wird in **Kapitel 2** eine allgemeine Einführung in das Konzept der Referenzarchitekturen gegeben. Neben der Erläuterung der begrifflichen Grundlagen werden in diesem Kapitel die Eigenschaften, Ziele und Arten von Referenzarchitekturen beschrieben und anhand eines Klassifikationsschemas sichtbar gemacht. Auf Basis dieses Verständnisses

werden Methoden zur Erstellung von Referenzarchitekturen in Abhängigkeit von der Art der Referenzarchitektur vorgestellt und die Methode beschrieben, nach der die RAII erstellt wurde. Anhand dieser Methode wird der Betrachtungsrahmen der RAII definiert und die jeweiligen Ausprägungen der einzelnen Schritte diskutiert.

Aufbauend auf diesem Wissen wird die Domäne der Instandhaltung in **Kapitel 3** beleuchtet. Dieses Wissen bildet den inhaltlichen, fachspezifischen Grundkern der Referenzarchitektur und umreißt den Betrachtungsrahmen, in welchem sich diese bewegt. Dazu werden die Begrifflichkeiten, Konzepte und aktuellen Entwicklungen der Instandhaltung vorgestellt. Zusätzlich werden dort als Ausgangspunkt für die Anforderungsuntersuchung von Instandhaltungssystemen aktuellen Systeme der Produktion sowie aktuelle existierende Systeme zur Unterstützung der Instandhaltung beleuchtet.

Anhand dieses Hintergrundwissens wird in **Kapitel 4**, basierend auf dem in Kapitel 2 definierten Betrachtungsrahmen der Referenzarchitektur, das Vorgehen für eine dedizierte Anforderungsaufnahme in Literatur und Praxis definiert. Zuerst wird daher eine Literaturrecherche und -analyse hinsichtlich Instandhaltungssystemen und deren Anforderungen durchgeführt. Als nächstes werden Anforderungen aus der Praxis bei vier Unternehmen erhoben. Diese Anforderungen werden analysiert und in generische und optionale Anforderungen unterteilt, sodass diese als Grundlage zur Erstellung der Kernelemente der Referenzarchitektur genutzt werden können.

In **Kapitel 5** wird die Darstellungsform der Referenzarchitektur vorgestellt. Zuerst wird das 4+1 Sichtenkonzept nach Kruchten (1995) erläutert, auf dem viele Software- wie auch Referenzarchitekturen basieren. Anschließend wird auf die Anpassungen dieses Konzeptes zur Darstellung der RAII näher eingegangen und das resultierende Darstellungsmodell präsentiert. Bei dieser Vorstellung werden insbesondere Aspekte und Anforderungen zur Darstellung von Referenzarchitekturen diskutiert.

Anhand des aus diesen Diskussionen resultierenden 3+1 Sichtenkonzeptes zur Darstellung der RAII wird diese in **Kapitel 6** ausführlich erläutert, indem die Inhalte und Konzepte der jeweiligen Architektursichten explizit vorgestellt werden. Auf den Inhalt und die Konzepte der Referenzarchitektur folgend wird in **Kapitel 7** die Evaluierung der Referenzarchitektur dargestellt. Zuerst werden die Ziele und Kriterien sowie die Methoden der Evaluation beschrieben. Die Evaluation unterteilt sich anschließend in drei Blöcke: eine analytische, merkmalsbasierte Evaluation, eine prototypische Umsetzung der Referenzarchitektur sowie eine Evaluation anhand durchgeführter Experteninterviews zur Bewertung der Referenzarchitektur.

Abschließend werden die Erkenntnisse aus der Dissertation in **Kapitel 8** zusammengefasst, der praktische und theoretische Beitrag herausgestellt sowie die Limitationen und zukünftigen Forschungsrichtungen beleuchtet.

2 Merkmale von Referenzarchitekturen

2.1 Einleitung

Das Artefakt dieser Dissertation soll die Entwicklung von integrierten Informationssystemen zur Unterstützung der Instandhaltung erleichtern. Das Artefakt ist hierbei eine sogenannte Referenzarchitektur, welche die Erleichterung der Entwicklung u. a. dadurch erzielt, dass sie als Blaupause für die Entwicklung von konkreten Softwarearchitekturen dient und zur Verbesserung von Anforderungserhebung/-management dienen kann. Der Begriff Referenzarchitektur ist jedoch in Literatur und Praxis nicht eindeutig definiert (Angelov et al., 2009, S. 141; Fernandez et al., 2016, S. 227). Allerdings kann eine deutliche Zunahme von Referenzarchitekturen insbesondere im Industrie 4.0 Kontext beobachtet werden. Um die Erstellung der RAI zu systematisieren und das in dieser Arbeit verwendete Verständnis von Referenzarchitekturen zu erläutern, ist es notwendig, die Begrifflichkeiten und Eigenschaften sowie den aktuellen wissenschaftlichen Stand zu Referenzarchitekturen zu untersuchen. Hierdurch sollen eine bessere und schnellere Verständlichkeit des Artefakts und eine wissenschaftliche Vorgehensweise zur Erstellung und Darstellung dieser identifiziert werden. Weiterhin soll die erstellte Referenzarchitektur durch eine Klassifizierung leichter in den Kontext von bisherigen Veröffentlichungen gesetzt werden können, um den Nutzen und die Art besser abgrenzen und vergleichen zu können. Für diese Untersuchung und die Herleitung der Klassifikation werden daher in diesem Kapitel verschiedene Sichtweisen hinsichtlich des Begriffes der Referenzarchitektur vorgestellt und diskutiert. Anschließend wird die dieser Arbeit zugrundeliegende Definition und Methodik zur Erstellung von Referenzarchitekturen ausgearbeitet.

Der Aufbau des Kapitels ist wie folgt: In Kapitel 2.1 werden die grundlegenden Begriffsdefinitionen im Kontext des Begriffes Referenzarchitektur vorgestellt. Zuerst werden benachbarte Begriffe definiert, um eine Unterscheidung und Abgrenzung zu ermöglichen. Anschließend werden verschiedene Definitionen und Sichtweisen auf eine Referenzarchitektur vorgestellt und diskutiert. Auf Basis dieser Definitionen und aktuellen wissenschaftlichen Publikationen zu Referenzarchitekturen werden deren Eigenschaften und Ziele hergeleitet. Anschließend wird die in dieser Arbeit verwendete Definition herausgearbeitet und in Kontext zu den benachbarten Begriffen gesetzt. Im Anschluss wird ein Klassifikationsschema vorgestellt, in den die in diesem Beitrag entwickelte Referenzarchitektur eingeordnet wird.

Auf Basis dieser Informationen werden Methoden für die Erstellung bestimmter Arten von Referenzarchitekturen in Kapitel 2.3 vorgestellt. Insbesondere wird das Vorgehen zur Erstellung einer Referenzarchitektur anhand spezifischer Anforderungen von Reidt et al. (2017a) präsentiert, welches zur Erstellung der RAI benutzt wurde und die einzelnen Schritte am Beispiel der RAI erläutert.

2.2 Begriffliche und theoretische Grundlagen

Für das Verständnis des Artefakts der RAII bedarf es der Kenntnis grundlegender Begrifflichkeiten im Kontext der Entwicklung von Referenzarchitekturen. Da keine einheitliche Sichtweise auf den Begriff Referenzarchitektur existiert, können verschiedene Sichtweisen und vielfältige Arten von Referenzarchitekturen in Literatur und Praxis unterschieden werden. Diese unterschiedlichen Sichtweisen und die daraus resultierenden Aspekte haben jedoch einen großen Einfluss auf das Vorgehen zur Erstellung einer Referenzarchitektur, den mit ihr verfolgten Zielen und Anforderungen, die an sie gestellt werden. Um die RAII in diesen Kontext einzuordnen und die eigene Erstellung zu systematisieren, werden die verschiedenen Sichtweisen und der aktuelle Forschungsstand in diesem Abschnitt vorgestellt und diskutiert. Anschließend wird anhand der Diskussion das in dieser Arbeit benutzte Begriffsverständnis illustriert.

Nachfolgend werden zunächst die benachbarten Begriffe der Referenzarchitektur Software, Produktlinienarchitektur und Referenzmodell definiert, um die Grundlagen für das Verständnis der späteren Definition einer Referenzarchitektur zu schaffen. Anschließend werden verschiedene Definitionen des Begriffs Referenzarchitektur gegenübergestellt, diskutiert und bestimmte Eigenschaften einer Referenzarchitektur aus diesen Definitionen und aktuellen Publikationen abgeleitet. Darauf aufbauend wird das dieser Arbeit zugrundeliegende Begriffsverständnis für den Term Referenzarchitektur vorgestellt. Diese eigene Begriffsdefinition wird herausgearbeitet und im Verhältnis zu grundlegenden Begriffen abgegrenzt. Anschließend werden Ziele und allgemeine Faktoren, die zu einer Referenzarchitektur führen, vorgestellt und das gesammelte Wissen dazu genutzt, eine Klassifikation von Arten einer Referenzarchitektur zu erstellen. Anhand dieser Klassifikation wird die RAII kategorisiert.

2.2.1 Softwarearchitektur

Softwarearchitekturen sind für die Erstellung von Softwaresystemen essentiell, da sie die konkreten Architekturentscheidungen darstellen und so die Entwicklung anleiten bzw. das Ergebnis beschreiben. In dieser Arbeit wird daher eine Softwarearchitektur nach der Definition von Bruns und Dunkel (2010, S. 202) wie folgt beschrieben:

*„Eine **Softwarearchitektur** beschreibt die übergeordnete Struktur eines Softwaresystems und deren globale Kontrollstrukturen. Sie beschreibt die wesentlichen Softwarebausteine in Form von Komponenten und legt fest, wie diese interagieren und kooperieren“.*

Im Rahmen dieser Arbeit werden Softwarearchitekturen auch als **konkrete Softwarearchitekturen** bezeichnet, um diese klar von Referenzarchitekturen abzugrenzen und die Spezifität dieser Art von Architekturen auszudrücken.

2.2.2 Produktlinienarchitektur

Ein weiterer Begriff, der im Bereich der Softwarearchitektur oft genannt wird, ist der der **Produktlinienarchitektur**⁵. Diese ist nach Nakagawa et al. (2011b, S. 210) eine spezielle Art der Softwarearchitektur und wird dazu benutzt, Produktlinien zu erstellen. Sie beschreibt explizit die Gemeinsamkeiten und die Variabilität der einzelnen Produkte einer Produktlinie und bildet die Basis für alle Produkte der Linie. Als Produkte können hier bspw. einzelne Softwareprodukte einer gesamten Softwarelinie verstanden werden, die einen gemeinsamen Kern besitzen.

Sie wird von manchen Autoren als spezielle Art einer Referenzarchitektur angesehen (Vogel et al., 2009, S. 513), die weniger abstrakt als eine Referenzarchitektur ist und besonderen Wert auf die umfassende Darstellung der Variabilität der einzelnen Produkte legt (Nakagawa et al., 2011b, S. 210). Taylor et al. (2009, S. 594) sehen die entscheidenden Unterschiede zum einen darin, dass Produktlinienarchitekturen komplett sind. Das heißt, sie stellen die komplette Ausprägung der Architektur der einzelnen Produkte im Verbund dar, Referenzarchitekturen hingegen müssen nicht vollständig sein. Das bedeutet, dass Teile einer Referenzarchitektur offen oder sehr abstrakt sein können, sodass diese abstrakten Elemente erst beim Erstellen einer konkreten Softwarearchitektur ausdefiniert werden. Zum anderen werden Produktlinienarchitekturen nur für eine Organisation und eine feste Anzahl an vorab bekannten Produkten erstellt. Manche Autoren verwenden den Begriff Produktlinienarchitektur synonym mit Referenzarchitektur (Arboleda et al., 2014, S. 178; Eklund et al., 2005), andere bezeichnen ihn als austauschbar (Gallagher, 2000, S. 3). In dieser Arbeit werden Produktlinienarchitekturen als Spezialformen einer Referenzarchitektur angesehen.

2.2.3 Referenzmodell

Einige Autoren sehen eine Referenzarchitektur als eine softwaretechnische Weiterführung eines Referenzmodells (Bass et al., 2003), andere benutzen auch diesen Begriff synonym. In dieser Arbeit wird der Begriff wie folgt definiert:

*„Ein **Referenzmodell** (ausführlich: Referenz-Informationsmodell) ist ein Informationsmodell, das Menschen zur Unterstützung der Konstruktion von Anwendungsmodellen entwickeln oder nutzen, wobei die Beziehung zwischen Referenz- und Anwendungsmodell dadurch gekennzeichnet ist, dass Gegenstand oder Inhalt des Referenzmodells bei der Konstruktion des Gegenstands oder Inhalts des Anwendungsmodells wieder verwendet werden“ (vom Brocke, 2015, S. 34).*

Mit dieser Definition wird der Begriff des Referenzmodells nicht allein an spezifischen Qualitätsmerkmalen (bspw. der Allgemeingültigkeit und Empfehlungswürdigkeit) festgemacht, da je nach Anwendungsfall spezifische Merkmale für die Qualität eines Referenzmodells ausschlaggebend sein sollten (vom Brocke/Buddendick, 2004, S. 341). Vielmehr wird die Abhängigkeit eines Referenzmodells vom Anwendungsmodell hervorgehoben und damit die spezifische Positionierung von Referenzmodellen im Konstruktionsprozess von Informationsmodellen (vom Brocke, 2015, S. 34).

⁵ Nach Nakagawa et al. (2011b, S. 210) existieren für diese Art der Architekturen folgende Namen in der Literatur: Software Produktlinienarchitektur, domänenspezifische Softwarearchitektur, Domänenarchitektur und teils auch Referenzarchitektur.

2.2.4 Definition Referenzarchitektur

Der Begriff der Referenzarchitektur wird zunehmend in wissenschaftlichen wie auch in praxisorientierten Veröffentlichungen verwendet. Dies geschieht jedoch in den meisten Fällen ohne fundierte Definition des Begriffs, was dazu führt, dass der Begriff Referenzarchitektur unterschiedlichste Bedeutungen für verschiedene Personen besitzt (Cloutier et al., 2009, S. 16). Daneben herrscht auch in der wissenschaftlichen Literatur kein eindeutiges Begriffsverständnis. Mittlerweile lassen sich dort unterschiedliche Definitionen und Sichtweisen beobachten. Nachfolgend ist in Tabelle 1 eine Auswahl an aktuellen Definitionen aufgeführt, die im Anschluss diskutiert werden. Dabei stellen die nachfolgenden Punkte des Kapitels 2.2.4 eine erweiterte Fassung des Beitrages von Reidt et al. (2018a) dar.

Autor	Definition
(Bergey et al., 1999, S. 32f)	„A reference architecture is an architectural style and/or design pattern for a specific domain or product line. It is insufficient for building systems in that a style or pattern by its nature requires the binding of unresolved issues. In addition, an architectural style represents a set of constraints on architectures of individual products to be built as part of the product line, and therefore expresses the commonality among the architectures of the different products.“
(Bass et al., 2003, S. 33f)	“A reference architecture is a reference model mapped onto software elements (that cooperatively implement the functionality defined in the reference model) and the data flows between them. Whereas a reference model divides the functionality, a reference architecture is the mapping of that functionality onto a system decomposition.”
(Gallagher, 2000, S. 3)	“A reference architecture is the generalized architecture of several end systems that share one or more common domains. The reference architecture defines the infrastructure common to the end systems and the interfaces of components that will be included in the end systems. The reference architecture is then instantiated to create a software architecture of a specific system. The definition of the reference architecture facilitates deriving and extending new software architectures for classes of systems. A reference architecture, therefore, plays a dual role with regard to specific target software architectures. First, it generalizes and extracts common functions and configurations. Second, it provides a base for instantiating target systems that use that common base more reliably and cost effectively.”
(Pohl et al., 2005, S. 124)	“The reference architecture is a core architecture that captures the highlevel design for the applications of the software product line. The reference architecture includes the variation points and variants documented in the variability model. It provides limits for the architectures of the separate applications, i.e. the application architectures. It determines which components are reusable and thus have to be developed during domain realization”
(Cloutier et al., 2009, S. 17)	“Reference Architectures capture the essence of existing architectures, and the vision of future needs and evolution to provide guidance to assist in developing new system architectures.”
(Vogel et al., 2009, S. 254)	“Referenzarchitekturen kombinieren allgemeines Architektur-Wissen und allgemeine - Erfahrung mit spezifischen Anforderungen zu einer architektonischen Gesamtlösung für einen bestimmten Problembereich. Sie dokumentieren die Strukturen des Systems, die wesentlichen Systembausteine, deren Verantwortlichkeiten und deren Zusammenspiel.”
(OASIS Committee, 2012, S. 9)	“A reference architecture models the abstract architectural elements in the domain of interest independent of the technologies, protocols, and products that are used to implement a specific solution for the domain. It differs from a reference model in that a reference model describes the important concepts and relationships in the domain focusing on what distinguishes the elements of the domain; a reference architecture elaborates further on the model to show a more complete picture that includes showing what is involved in realizing

Autor	Definition
	the modeled entities, while staying independent of any particular solution but instead applies to a class of solutions.”
(VDI/VDE Fachausschuss Industrie 4.0, 2017, S. 17)	“Modell für eine Architekturbeschreibung (für Industrie 4.0), die allgemein genutzt wird und als zweckmäßig anerkannt ist (Referenzcharakter hat)”
(Lin et al., 2015, S. 10f)	“A reference architecture provides guidance for the development of system, solution and application architectures. It provides common and consistent definitions in the system of interest, its decompositions and design patterns, and a common vocabulary with which to discuss the specification of implementations so that options may be compared. A reference architecture provides a common framework around which more detailed discussions can center. By staying at a higher level of abstraction, it enables the identification and comprehension of the most important issues and patterns across its applications in many 80 different use cases. By avoiding specifics, a reference architecture allows subsequent designs to follow the reference architecture without the encumbrance of unnecessary and arbitrary restrictions.”

Tabelle 1: Übersicht über Definitionen des Begriffs Referenzarchitektur

Quelle: Eigene Darstellung

Die Definition von Bergey et al. (1999, S. 32f) sieht eine Referenzarchitektur als bestimmten Architekturstil oder Entwurfsmuster für eine Domäne oder Produktlinie an, die explizite Variationspunkte enthält und diese bei dem Entwurf einer konkreten Softwarearchitektur instanziiert werden können. Variationspunkte sind in einer Referenzarchitektur vorab definierte entsprechende Anpassungs- und Erweiterungspunkte die spätere Änderungen und Erweiterungen an bestimmten Stellen der Architektur möglichst einfach gestaltbar machen (Trefke, 2012, S. 10). Beim Entwurf einer konkreten Softwarearchitektur auf Basis der Referenzarchitektur bestimmt diese mithilfe von Randbedingungen den Entwicklungsprozess und die spätere Lösung. Spezifische Elemente der konkreten Softwarearchitektur müssen komplett definiert werden, da nach Bergey et al. (1999) Referenzarchitekturen nicht zwangsläufig als vollständig gelten. Das heißt bestimmte Elemente sind in der Referenzarchitektur nur abstrakt oder unvollständig beschrieben und sie kann bei der späteren Instanzierung einer konkreten Softwarearchitektur auch an Stellen erweitert werden, die vorab nicht vorgesehen sind.

Ein Referenzmodell ist bei Bass et al. (2003) eine Aufteilung von Funktionalitäten mitsamt Datenfluss zwischen diesen Elementen. Genauso offen wird eine Referenzarchitektur definiert: Sie ist ein Referenzmodell, welches auf Softwareelementen abgebildet wird. Diese Definition ist eine der am häufigsten verwendeten in der wissenschaftlichen Literatur (Bauer et al., 2013; Nakagawa et al., 2011a).

Gallagher (2000) sieht in der Referenzarchitektur eine generalisierte Architektur von mehreren Endsystemen, die eine oder mehrere Domänen teilen. Bei ihm hat eine Referenzarchitektur einen softwaretechnischen Infrastrukturcharakter, da die Referenzarchitektur diese von mehreren Systemen zusammen mit Interfaces definiert.

Pohl et al. (2005) sehen eine Referenzarchitektur als Kernarchitektur, die jedoch das übergeordnete Design für mehrere Produkte einer Softwareproduktlinie definiert. Dabei spielt

bei ihnen vor allem die Wiederverwendbarkeit von bestimmten Komponenten eine Rolle, welche bei den restlichen Definitionen oft nicht direkt genannt wird. Sie betonen ebenso wie Bergey et al. (1999), dass Referenzarchitekturen Randbedingungen festlegen, die bei der Ableitung konkreter Softwarearchitekturen berücksichtigt werden müssen, aber auch Hilfestellung bei der Ableitung geben können.

Cloutier et al. (2009) sehen Referenzarchitekturen als Essenz von bestehenden Architekturen, die Visionen von zukünftigen Entwicklungen, Bedürfnissen sowie Evolutionen umfassen. Dies dient vordergründig dazu die Entwicklung von neuen Systemen zu erleichtern. In ihren weiteren Ausführungen sehen sie eine Referenzarchitektur explizit nicht als eine rein technische Architektur, da sie ebenso eine Facharchitektur und einen Kundenkontext enthalten sollte. Ihrer Sicht nach sollte eine Referenzarchitektur ein gemeinschaftliches Lexikon und eine Taxonomie bereitstellen. Dies liegt u. a. daran, dass eine Referenzarchitektur für eine große Anzahl an Stakeholdern verständlich und nachvollziehbar sein muss, damit sie ein gemeinsames Verständnis zwischen den Beteiligten herstellen kann. Referenzarchitekturen konzentrieren sich bei Cloutier et al. (2009) darauf, die Gemeinsamkeiten von Systemen darzustellen und können viele mögliche Variationen bzw. Variationspunkte ignorieren. Sie bezeichnen ihre Sichtweise auf Referenzarchitekturen als Systems Engineering orientiert. Dadurch unterscheidet sie sich teilweise deutlich von den zuvor dargestellten Definitionen, die vor allem aus Sicht des Software Engineering entworfen wurden.

Vogel et al. (2009) heben den Wissensmanagementaspekt von Referenzarchitekturen hervor, da diese allgemeines Architekturwissen und allgemeine Erfahrung enthalten sollen. Weiter wird dieses Wissen mit spezifischen Anforderungen für die zu entwerfende Gesamtlösung eines bestimmten Problembereichs kombiniert. Daraus sollen Systembausteine, deren Verantwortlichkeiten und das Zusammenspiel in Form der Struktur des zu entwerfenden Systems resultieren. Die Sichtweise von Vogel et al. (2009) ist auch technisch geprägt, unterscheidet sich aber vor allem durch den Fokus auf das Wissensmanagement und der expliziten Nennung von spezifischen Anforderungen, wodurch auch nicht rein technische Aspekte in der Definition enthalten sind.

Die Definition des OASIS Committee (2012) stellt vor allem den abstrakten Charakter einer Referenzarchitektur heraus. Hier wird die Hauptaufgabe einer Referenzarchitektur darin gesehen, ein komplettes Bild einer Klasse von Lösungen darzustellen und zu vermitteln. Abstrakte Architekturelemente werden nach dieser Definition technologie-neutral definiert und zeigen die wichtigsten Konzepte und Verbindungen zueinander auf. Darüber hinaus werden nicht nur die Elemente selbst beschrieben, sondern auch wie diese realisiert werden können und wer und was daran beteiligt ist.

Die Definition, die insbesondere bei deutschen Referenzarchitekturen im Kontext der Industrie 4.0 benutzt wird, ist die des VDI/VDE Fachausschuss Industrie 4.0 (2017). Diese Definition betont vor allem den Referenzcharakter und damit die allgemeine Nutzung einer Referenzarchitektur. Die Definition ist sehr breit gehalten, da von einer Architekturbeschreibung gesprochen wird, die jedoch nicht weiter definiert wird. Von den hier vorgestellten Definitionen lässt diese den größten Interpretationsspielraum zu.

Die Definition von Lin et al. (2015), die in der Referenzarchitektur des Industrial Internet präsentiert wird, betont hingegen nicht nur den technischen Aspekt, sondern auch die Bereitstellung von allgemeinen Definitionen und eines einheitlichen Vokabulars, durch die die Entwicklung von Systemen und Architekturen unterstützt wird. Anhand einer

Referenzarchitektur soll aufgrund dieser Eigenschaften die Diskussion über das zu entwickelnde System und der Referenzarchitektur an sich ermöglicht werden. Sie betonen die übergreifende Wirkung von Referenzarchitekturen, die nicht nur für ein bestimmtes System oder einer Klasse von Systemen genutzt werden können. Diese Definition unterscheidet sich demnach im Bereich der Kommunikationsunterstützung und dem expliziten Hinweis auf die Use Case-übergreifende Wirkung von Referenzarchitekturen.

Anhand all dieser Definitionen wird ersichtlich, dass sehr unterschiedliche Sichtweisen auf den Begriff Referenzarchitekturen aus wissenschaftlicher und praktischer Perspektive bestehen. Da sich ein Großteil der in Literatur und Praxis veröffentlichten Referenzarchitekturen auf keine Definitionen beziehen und die Interpretationen der Definitionen einen großen Spielraum lassen, ist die Bandbreite der Arten von Referenzarchitekturen enorm. Um diese Arten von Referenzarchitekturen zu systematisieren, werden nachfolgend auf Basis der vorgestellten Definitionen Eigenschaften von Referenzarchitekturen und die Sichtweise auf diese in der vorliegenden Arbeit diskutiert. Diese dienen als Grundlage zur Herleitung der in dieser Arbeit benutzten Definition und der erstellten Klassifikation von Referenzarchitekturen.

2.2.4.1 Diskussion Eigenschaften einer Referenzarchitektur

Abstraktion: Referenzarchitekturen zeichnen sich in vielen Fällen durch eine deutlich höhere Abstraktion wie Softwarearchitekturen aus. Zum Beispiel werden individuelle Besonderheiten von konkreten Architekturen abstrahiert, um eine Allgemeingültigkeit der Referenzarchitektur zu erlangen (Trefke, 2012), individuelle Freiheiten bei der Ableitung von konkreten Architekturen zu ermöglichen oder um Zusammenhänge gezielter darzustellen. Der Grad der Abstraktion kann sich jedoch von Referenzarchitektur zu Referenzarchitektur sehr stark unterscheiden. Zudem existieren in der Literatur unterschiedliche Sichtweisen auf den Grad der Abstraktheit. Während einige Autoren in ihren Definitionen zwingend von einer hohen Abstraktion einer Referenzarchitektur ausgehen (vgl. Lin et al., 2015, S. 10f), beinhaltet eine Referenzarchitektur für andere eine detaillierte Architektur auf dem Level von Programmiercode und fertig implementierten Funktionen (vgl. Gallagher, 2000). Vogel et al. (2009, S. 257) stellen eine bestimmte Art von Referenzarchitekturen vor, die eine niedrige Art von Abstraktion vorsieht. Sie nennen diese Arten von Referenzarchitekturen „Plattformbezogene Referenzarchitekturen“. Diese haben laut ihnen einen Standardcharakter und müssen bzw. sollen so umgesetzt werden, wie sie beschrieben werden. Dazu sind konkrete Implementierungen in Form von Quelltext und lauffähige Bausteine als Teil der Referenzarchitektur enthalten.

Im Begriffsverständnis dieser Arbeit wird eine Referenzarchitektur als tendenziell abstrakt bezeichnet, jedoch existieren mehrere Arten von Referenzarchitekturen, die sich bezüglich des Abstraktionsgrades unterscheiden. Daher wird bei dieser Eigenschaft zwischen einer detaillierten Referenzarchitektur auf Codebasis, einer Mischform, die teils abstrakte Elemente und teils spezifische Elemente enthält, sowie einer abstrakten Art der Referenzarchitektur unterschieden. Eine Referenzarchitektur auf Codebasis kann konkrete Elemente implementieren, Interfaces spezifizieren und eine lauffähige Lösung beinhalten, wenn dies dem Ziel dient, die Entwicklung zu erleichtern oder eine Standardisierung herbeizuführen. Abstrakte Architekturen, die von der Codebasis abstrahieren, versuchen damit bspw. Gemeinsamkeiten zwischen Architekturen herauszustellen, Technologieneutralität zu gewähren (OASIS

Committee, 2012) oder Raum für Spezialisierung bzw. Anpassung zu lassen. Weiterhin sind auch Mischformen denkbar, bei denen einige Aspekte der Referenzarchitektur detailliert ausgearbeitet sind, andere hingegen offengelassen wurden. Die abstrakteren Elemente erlauben offene Erweiterungen und individuelle Freiheiten und bei den spezifischen Stellen wird eine Standardisierung herbeigeführt. Eine weitere Unterscheidung zwischen dem Grad der Abstraktheit insbesondere im Hinblick auf die Darstellung wäre prinzipiell möglich, jedoch lassen sich hier keine objektiven Kriterien aufstellen, da die verwendeten Techniken zur Erstellung und Darstellung einer Referenzarchitektur zu sehr variieren. Aus diesem Grunde wurde bewusst die breite Unterscheidung gewählt.

Technologieneutralität: Ein Teilaspekt der Abstraktion ist die Wahrung der Technologieneutralität einer Referenzarchitektur. Um keine Einschränkung bei der Implementierung vorzugeben wird bei Referenzarchitekturen mit höherer Abstraktion keine Technologie spezifiziert, sondern die gesamte Referenzarchitektur technologieneutral definiert (OASIS Committee, 2012). Dies kann den Fokus eher auf die logische Funktionsweise und die Anforderungen legen. Andere Ansätze implementieren schon komplette Teile der Applikation in einer Technologie oder Referenzarchitekturen sind explizit für bestimmte Technologien definiert (Hurley/Wang, 2014). Zu letzteren Kategorie gehören auch die „Plattformbezogenen Referenzarchitekturen“ von Vogel et al. (2009, S. 257). Auch hinsichtlich der Technologieneutralität existieren Mischformen, bei denen Referenzarchitekturen bei bestimmten Elementen des zu entwickelnden Systems Vorgaben oder Empfehlungen aussprechen (z. B. bei Kommunikationsprotokollen), der Kern der Architektur wird jedoch technologieneutral belassen. In dieser Arbeit werden daher jene drei Ausprägungen der Technologieneutralität als Eigenschaften von Referenzarchitekturen angesehen.

Referenzarchitekturen können für eine bestimmte Industrie bzw. Branche erschaffen werden oder branchenübergreifend gültig sein. Vogel et al. (2009) unterscheiden zwischen zwei Arten von Referenzarchitekturen hinsichtlich ihrer Industriebezogenheit. Die eine Art bezeichnet er als industriebezogene Referenzarchitekturen, die für eine spezifische Industrie geschaffen wurden (z. B. AUTOSAR). Die andere Art nennen sie industrieübergreifende Referenzarchitekturen, die mehrere Industriezweige betreffen. Dies können bspw. Referenzarchitekturen für SCM oder CRM sein, die mehrere Domänen betreffen. Insbesondere im Industrie 4.0 Kontext sind Referenzarchitekturen zu finden, die industrieübergreifend und produktunabhängig gestaltet sind und auch mehrere Arten von Systemen umfassen (Adolphs et al., 2015; Lin et al., 2015). In dieser Arbeit werden daher Referenzarchitekturen hinsichtlich ihres **Industriefokus** unterschieden.

Eine weitere Eigenschaft einer Referenzarchitektur betrifft den **Produktfokus**. Einige Definitionen gehen davon aus, dass Referenzarchitekturen auf ein Produkt oder eine Produktgruppe bzw. -linie fokussiert sind, wobei die Produkte vorab feststehen bzw. deren ungefähre Ausgestaltung fixiert ist (Arboleda et al., 2014; Bergey et al., 1999; Pohl et al., 2005). Dies ist vor allem bei Autoren anzutreffen, für die der Begriff Referenzarchitektur synonym mit dem Begriff Produktlinienarchitektur ist. Im Kontrast dazu existieren Referenzarchitekturen, die produkt- und produktklassenunabhängig sind. Diese beschreiben beispielweise Securityelemente, die in vielen unterschiedlichen Produkten aus unterschiedlichen Domänen genutzt werden können, oder sie beschreiben abstrakte Konzepte, die domänenübergreifend eingesetzt werden können. Beispiele sind Industrie 4.0/Industrial

Internet (Adolphs et al., 2015; Lin et al., 2015) sowie Security Referenzarchitekturen (Addo et al., 2014; Carroll et al., 2013; Fernandez et al., 2016). Bei dem Referenzarchitekturbegriff in dieser Arbeit wird zwischen diesen Ausprägungen unterschieden, da sie verschiedene Arten von Referenzarchitekturen darstellen und auf unterschiedliche Art und Weise die Entwicklung von Systemen zu erleichtern. Es wird zwischen einer Referenzarchitektur mit Fokus auf nur einem Produkt, einer Produktfamilie oder einem produktübergreifenden Fokus unterschieden. Referenzarchitekturen sind demnach nicht wie bei anderen Definitionen auf eine Ausprägung beschränkt.

Ein weiterer wichtiger Aspekt zur Einordnung von Referenzarchitekturen ist der **Unternehmensfokus**. Es existieren viele Referenzarchitekturen, die nur für ein Unternehmen genutzt werden (Martinez-Fernandez et al., 2013, S. 95) um dort bspw. Produkte zu harmonisieren und Wiederverwendbarkeit zu fördern. Diese sind meist nicht öffentlich und dienen dazu, interne Lösungen zu standardisieren und Wissen zu konservieren. Demgegenüber stehen unternehmensübergreifende Referenzarchitekturen, die prinzipiell offen und nicht auf ein Unternehmen zugeschnitten sind.

Der Referenzcharakter wird von manchen Autoren betont (VDI/VDE Fachauschuss Industrie 4.0, 2017), bis zum dem Punkt, dass Referenzarchitekturen **Allgemeingültigkeit** besitzen sollten (Trefke, 2012, S. 17). Dies kann zum einen normativ gelten, sodass Referenzarchitekturen als Standard akzeptiert werden. Als Standard werden Referenzarchitekturen bspw. akzeptiert, wenn sich die Überlegenheit des Konzeptes beweisen lässt, wegen hoher Akzeptanz der Lösung oder wegen einer Einigung auf Referenzarchitekturen in den nötigen Fachgremien und/oder marktdominierenden Unternehmen. Zum anderen kann eine Allgemeingültigkeit auch deskriptiv sein, da die Referenzarchitektur einen bestehenden Sachverhalt abstrahiert und die bestehenden Zusammenhänge allgemeingültig beschreibt (bspw. können Gemeinsamkeiten von bestimmten Web-Browser Architekturen im gewissen Rahmen allgemeingültig sein (vgl. Grosskurth/Godfrey, 2005)). Ähnlich der Argumentation seitens vom Brockes bzgl. des Begriffs des Referenzmodells wird das in dieser Arbeit verfolgte Begriffsverständnis der Begriff Referenz nicht automatisch mit einem Empfehlungscharakter oder einer Allgemeingültigkeit verbunden, da beide Aspekte schwer überprüfbar sind (vom Brocke, 2015, S. 32ff). Der Begriff Referenz wird hier als Bezugspunkt angesehen. Eine Referenzarchitektur dient dabei als Bezugspunkt in dem Sinne, dass sie sachdienlich eine inhaltsbezogene Unterstützung zur Ableitung von konkreten Architekturen ermöglicht. Sofern eine Referenzarchitektur instanziiert und/oder logisch hergeleitet werden kann, gilt diese Eigenschaft im Rahmen des verwendeten Begriffsverständnisses als erfüllt. Einen Empfehlungscharakter besitzt eine Referenzarchitektur hingegen, wenn sie nicht nur einen Bezugspunkt darstellt, sondern dieser Bezugspunkt auch eine Art Best Practice bzw. Good Practice darstellt. Durch das Befolgen dieses Best Practices soll davon ausgegangen werden können, dass durch die Anwendung ein Vorteil resultiert und das dieses Best Practice auf andere Fälle übertragbar ist. Die oben beschriebene deskriptive bzw. normative Allgemeingültigkeit gehen dabei über diese beiden Punkte hinaus. Einen Empfehlungscharakter und die Allgemeingültigkeit können nachgewiesen und als zusätzliches Qualitätskriterium herangezogen werden, dies ist jedoch vom Anwendungsfall abhängig.

Referenzarchitekturen können explizite **Variationspunkte** enthalten (Bergey et al., 1999; Pohl et al., 2005). Wenn Architekturen von Referenzarchitekturen abgeleitet werden, können diese

genutzt werden, um sich für bestimmte Varianten zu entscheiden und vorab Möglichkeiten der Variation an bestimmten Ankerpunkten zuzulassen. Die Verwendung von Variationspunkten ist im hier angewandten Begriffsverständnis eine fakultative Möglichkeit, definierte Wahlmöglichkeiten oder optionale Elemente in die Referenzarchitektur zu integrieren. Demzufolge wird zwischen Referenzarchitekturen mit Variationspunkten und Referenzarchitekturen ohne Variationspunkten unterschieden.

Technischer Fokus: Die Sichtweise mancher Autoren geht davon aus, dass Referenzarchitekturen (rein) technischer Natur sind (Bergey et al., 1999; Gallagher, 2000), andere Autoren betonen eine Notwendigkeit, dass der Fachkontext auch abgedeckt werden muss (Nakagawa/Oquendo, 2013). Im dieser Arbeit zugrundeliegenden Begriffsverständnis besitzt eine Referenzarchitektur prinzipiell auch einen technischen Fokus. Darunter wird verstanden, dass eine Referenzarchitektur technische bzw. softwaretechnische Aspekte einer Architektur adressiert. Dies kann jedoch technologiefrei und abstrakt geschehen. Neben dem technologischen Fokus ist die Einbindung des Kontextes eines technischen Systems ein zusätzliches Qualitätskriterium einer Referenzarchitektur.

Referenzarchitekturen können **unvollständig** sein (Bergey et al., 1999). Sie müssen nicht alle Teile bereithalten, die zur Ableitung einer konkreten Architektur nötig sind. Bestimmte Aspekte einer konkreten Architektur können der jeweiligen Implementierung überlassen werden und werden daher nicht in der Referenzarchitektur behandelt. Taylor et al. (2009, S. 589) unterscheiden zwischen drei Arten von Referenzarchitekturen hinsichtlich der Vollständigkeit:

- **Vollständige Referenzarchitekturen** für ein einzelnes Produkt. Diese bilden komplette System für eine spezielle Domäne ab und werden daher als vollständig bezeichnet.
- **Unvollständige invariante Referenzarchitekturen** zeichnen sich dadurch aus, dass eine Teilarchitektur dieser spezifiziert werden kann, die konstant und ohne Anpassung bei allen Produkten eingesetzt wird. Teile, die bei abgeleiteten Produkten unterschiedlich sind, werden nicht spezifiziert. Teilweise kann jedoch ein Leitfadenteil der Referenzarchitektur sein, der ein Vorgehen für die Implementierung der offenen, nicht spezifizierten Elemente beschreibt.
- **Invariante Referenzarchitekturen mit expliziten Variationspunkten** von Taylor et al. (2009, S. 589) besitzen ebenso wie unvollständige invariante Referenzarchitekturen Variationspunkte. Diese sind jedoch nicht unvollständig, sondern alle Variationsalternativen sind im Detail ausgearbeitet.

Der Aspekt der Unvollständigkeit findet sich in nahezu jeder Definition wieder und wird auch in dem Begriffsverständnis dieser Arbeit so interpretiert, dass Referenzarchitekturen unvollständig sein können. Nur mit dieser Eigenschaft können Ausführungen über Technologieunabhängigkeit und Abstraktion als nachvollziehbar erscheinen. Die drei Arten von Taylor et al. (2009, S. 589) werden zusammen mit der Unterteilung von Referenzarchitekturen nach vollständig und unvollständig und der Eigenschaft der Variationspunkte abgedeckt.

Referenzarchitekturen können nach Angelov et al. (2014a, S. 2) entweder **praxisgetrieben** sein und bestehendes Wissen einer Domäne zusammenfassen, sofern genug bewährtes Wissen zusammengetragen wurde. Oder sie sind **forschunggetrieben** und haben die Unterstützung

der Erstellung von neuartiger Architekturen zum Ziel, die zum Zeitpunkt der Definition der Referenzarchitektur als schwer zu erstellen wahrgenommen werden. Bei Galster (2015, S. 7) zielen forschungsgetriebene Referenzarchitekturen darauf ab, dass die Entwicklung von innovativen Produkten erleichtert wird. Dabei sind diese Referenzarchitekturen durch einen größeren Grad an Freiheit gekennzeichnet. Galster (2015, S. 7) bezeichnet diese Art von Referenzarchitekturen nicht explizit als forschungsgetrieben. Die Art von Referenzarchitekturen, die in dieser Arbeit als forschungsgetrieben bezeichnet werden, sind jedoch identisch mit der Art von Referenzarchitekturen, die Galster beschreibt.

2.2.4.2 Ziele einer Referenzarchitektur

Die Ausprägungen der jeweiligen Eigenschaften hängen stark von den verfolgten Zielen einer Referenzarchitektur ab. In der Literatur werden verschiedene Faktoren und übergeordnete Ziele für die Entwicklung von Referenzarchitekturen genannt:

Cloutier et al. (2009, S. 18) sehen als Treiber der Entwicklung von Referenzarchitektur die **steigende Dynamik** und **Integrationsanforderungen** von aktuellen **offenen Systemen** und der deutlich **steigenden Komplexität, Größe und Umfang** von Softwarelösungen. Muller (2007, S. 2) sieht in diesem Kontext insbesondere die Herausforderung, dass die **Anzahl der beteiligten Personen** an der Entwicklung zunimmt, diese sich an **verschiedenen Lokationen** befinden, **unterschiedlichen Unternehmen** und **Kulturen** angehören und es somit Unterstützung bei der Entwicklung dieser verteilten Systeme bedarf.

Trefke (2012, S. 10) nennt zwei Aspekte, die mit einer Referenzarchitektur ermöglicht werden sollen: Zum einen die **Erhöhung von Qualität** durch Nutzung **bewährter Lösungen**, da aus seiner Sicht Referenzarchitekturen allgemeingültige **Best Practices** darstellen. Zum anderen wird aufgrund der **Wiederverwendung** von Elementen aus der Referenzarchitektur die Entwicklung mit **geringerem Zeitaufwand** und **niedrigeren Kosten** im Vergleich zu einer Neuentwicklung durchgeführt. Weiterhin nennt Trefke (2012) die notwendige **interdisziplinäre Zusammenarbeit**, den **Umfang des Systems** und die **große Anzahl der daran beteiligten Akteure** als Treiber von Referenzarchitekturen (Trefke, 2012, S. 9).

Eine **Standardisierung** kann nach Gallagher (2000, S. 3) das primäre Ziel einer Referenzarchitektur sein. Dies soll durch standardisierte und fest beschriebene Interfaces zwischen Softwarekomponenten erreicht werden. Dadurch können die sekundären Ziele **Risikoreduzierung** und **Kostenreduzierung** durch eine **vereinfachte und beschleunigte Entwicklung** aufgrund von fixierten Standards und Standardkomponenten erfüllt werden.

Nakagawa et al. (2011b, S. 1) nennen auch die **Reduzierung des Time-to-Market** sowie die **Größe und Komplexität** als Treiber für Referenzarchitekturen. Martínez-Fernández et al. (2013a, S. 308) nennen in ihrer Untersuchung zu Vor- und Nachteilen von Referenzarchitekturen weitere übergeordnete Ziele wie z. B. die Reduzierung der Instandhaltungskosten durch Referenzarchitekturen. Dieses Ziel wird hier als Kombination der Reduzierung des Time-to-Market und Kosten- und Zeiteinsparungen angesehen⁶.

⁶ Auf die Nennungen weiterer genannten Vor- und Nachteile von Martínez-Fernández et al. (2013a), die auch als übergeordnete Ziele interpretiert werden kann, wird hier verzichtet, da diese Punkte meist Spezialisierungen der bereits angeführten Ziele darstellen.

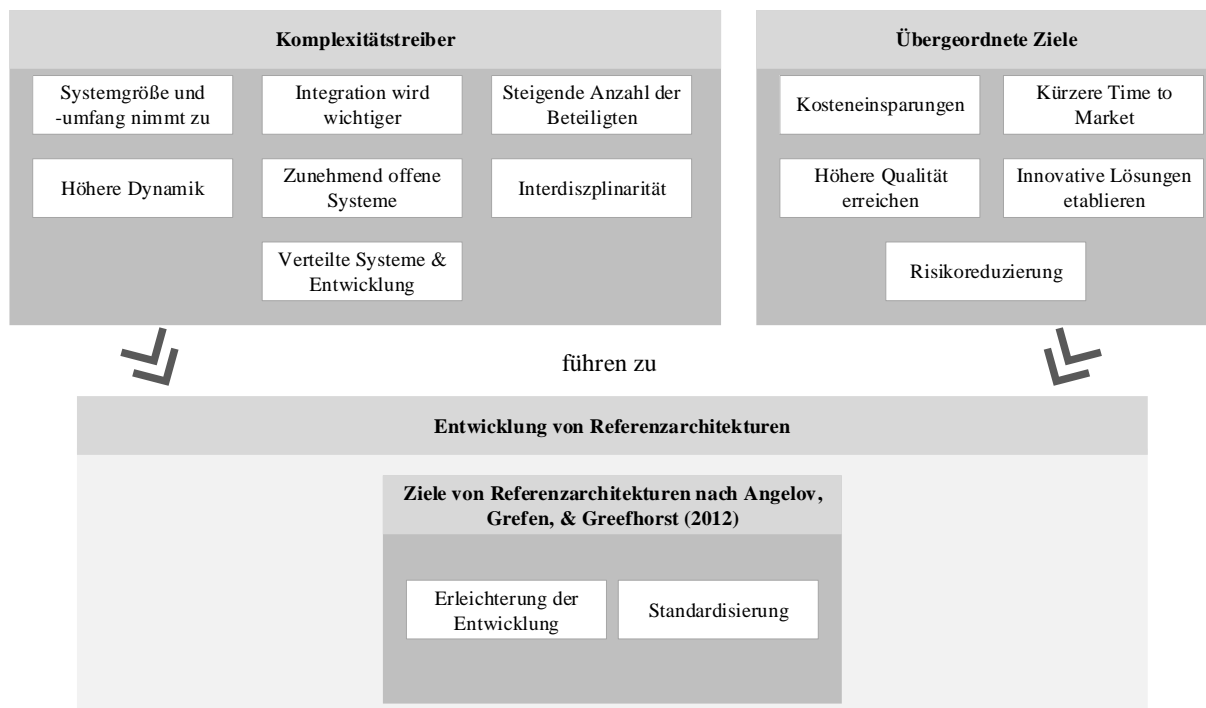


Abbildung 4: Ziele und Treiber von Referenzarchitekturen

Quelle: In Anlehnung an Reidt et al. (2018a)

In Abbildung 4 sind viele der genannten Aspekte, die zur Entwicklung einer Referenzarchitektur führen, als Komplexitätstreiber zusammengefasst. Diese umfassen bspw. die zunehmende Systemgröße, die zunehmenden Anforderungen durch verteilte offene Systeme und deren Entwicklung. Diese Faktoren führen zu dem Bedarf der Komplexitätsbeherrschung durch eine Referenzarchitektur. Daneben lassen sich übergeordnete Ziele zusammenfassen, die aus Organisationssicht mit einer Referenzarchitektur umgesetzt werden sollen. Dazu gehören angestrebte Kosteneinsparungen, Risikoreduzierung, höhere Qualität und die Etablierung innovativer Lösungen. Diese Aspekte führen zur Entwicklung von Referenzarchitekturen.

Referenzarchitekturen sollen daher zum einen die Komplexität beherrschbar machen, zum anderen die Erfüllung von übergeordneten Zielen einer Organisation bzw. mehrerer Organisationen ermöglichen. Dabei ist die Komplexitätsbeherrschung eine Grundbedingung zur Erfüllung der übergeordneten Ziele. Je nachdem wie Referenzarchitekturen die Komplexität beherrschbar machen sollen oder wie sie die übergeordneten Zielen erfüllen, können sie nach Angelov et al. (2012, S. 421) auch selbst nach ihren Zielen unterschieden werden:

- **Referenzarchitekturen mit Standardisierungsziel:** Das Ziel dieser Referenzarchitekturen ist es, eine Standardisierung der Architektur bzw. der entwickelten Systeme zu erreichen.
- **Referenzarchitekturen mit Erleichterungsziel:** Referenzarchitekturen mit diesem Ziel versuchen durch Richtlinien, Wissen oder Inspirationen die Entwicklung von Systemen zu erleichtern.

Insbesondere die Erleichterung der Entwicklung wird von vielen Referenzarchitekturen angestrebt, doch die Wege, wie dies geschieht, wird bei Referenzarchitekturen sehr unterschiedlich behandelt. Nachfolgend werden verschiedene Möglichkeiten aufgezählt, die zur Erreichung der Ziele führen.

- Bei Referenzarchitekturen können Interfaces und (abstrakte) Komponenten detailliert spezifiziert werden (Haft et al., 2005, S. 109). Hierdurch wird insbesondere eine **Standardisierung** herbeigeführt und folglich ein Element von Referenzarchitekturen mit Standardisierungsziel (Angelov et al., 2012). Dadurch wird auch die Interoperabilität von Lösungen und Elementen ermöglicht, die auf solch einer Referenzarchitektur basieren. Zusätzlich werden Richtlinien und Regeln für die Implementierung definiert. Dabei kann die Erleichterung der Entwicklung auch indirekt aus der Erfüllung des Standardisierungsziels folgen.
- **Erfassung von Wissen** - Referenzarchitekturen können Sammlungen von bestimmtem Wissen enthalten. Dieses Wissen soll die Entwicklung erleichtern. Häufig genannt werden folgende Arten:
 - Referenzarchitekturen erfassen **architektonisches Wissen** in Form von: Architekturmustern und bestimmten Architekturstilen (Muller/van de Laar, 2009, S. 3) oder durch Darstellung von Gemeinsamkeiten verschiedener Architekturen einzelner Produkte/Systeme (Bergey et al., 1999; Gallagher, 2000; Galster/Angelov, 2015) .
 - Referenzarchitekturen können Wissen in Form von **Best Practices** enthalten (Nakagawa et al., 2014a, S. 144). Diese werden in Form von bewährten Architekturen bzw. Lösungen der „besten“ Architekturen oder Richtlinien zur Entwicklung bereitgestellt. Dies ist nur in reifen Domänen möglich, wo Zugriff auf einen großen Erfahrungsschatz hinsichtlich bewährter Lösungen verfügbar ist.
 - Daneben wird von einigen Autoren die Bereitstellung von explizitem und implizitem **Domänenwissen** als ein sehr wichtiger Aspekt zur Bereitstellung von Wissen genannt (Martínez-Fernández et al., 2013b, S. 100; Nakagawa et al., 2011b, S. 210; Vogel et al., 2009, S. 254). Das Domänenwissen kann durch die
 - Bereitstellung einer einheitlichen, eindeutigen, breit verstandenen **Domänenterminologie** verfügbar gemacht werden (Martínez-Fernández et al., 2013b, S. 100; Nakagawa et al., 2014a, S. 144) oder
 - **Geschäftsregeln und -abläufen** (Martínez-Fernández et al., 2013b, S. 100; Nakagawa et al., 2014a, S. 144).
 - Daneben können Referenzarchitekturen **Softwareelemente** enthalten, welche die Wiederverwendbarkeit fördern und fordern, sodass hierdurch Arbeitersparnisse erzielt werden (Arboleda et al., 2014, S. 173).
 - Diese Softwareelemente können Infrastrukturkomponenten sein, die bei allen auf der Referenzarchitektur basierenden Systemen eingesetzt werden (Gallagher, 2000).
 - Neben ausgearbeiteten Softwareelementen können auch Funktionalitäten von Komponenten auf Systemebene beschrieben werden, um Wissen bereitzustellen. Die Beschreibung kann dabei auch abstrakt sein (Bass et al., 2003; Vogel et al., 2009).
 - **Randbedingungen** zur Ableitung von konkreten Architekturvorgaben können vorgegeben werden, sodass sich die Entwicklung innerhalb eines festen Rahmens bewegen kann (Bergey et al., 1999; Pohl et al., 2005).
- Daneben wird die Entwicklung durch Referenzarchitekturen nicht nur durch das Bereitstellen von Wissen erleichtert, sondern auch dadurch, dass die

Referenzarchitektur als **Kommunikationselement** eingesetzt werden kann: Dabei wird eine Referenzarchitektur vordergründig als Kommunikationselement benutzt (Lin et al., 2015, S. 10f), um durch Bereitstellung eines **kompletten Bildes** des zu entwickelnden Systems und einer einheitlichen Begriffswelt in Form von bspw. Taxonomien eine Zusammenarbeit verschiedener Stakeholder zu fördern (Martinez-Fernandez et al., 2015, S. 154).

2.2.4.3 Diskussion und Einordnung des Begriffs Referenzarchitektur

Wie die vorangegangenen Kapitel zeigen, zeichnet sich der Begriff einer Referenzarchitektur durch eine große Heterogenität aus. Im wissenschaftlichen Diskurs werden die in den vorherigen Kapiteln dargelegten Eigenschaften genannt, jedoch bei keinem der untersuchten Autoren in der Gänze aufgezählt oder auch zueinander in Verbindung gesetzt. Teils ergibt sich aus den Definitionen ein Bild, dass Referenzarchitektur sehr technisch sein können, bei manchen Autoren gar komplette technische Architekturen sind. Andere Autoren sehen vielmals eine Inkludierung nicht technischer Faktoren als besonders wichtig oder gar essentiell für Referenzarchitekturen an. Diese und andere Aspekte erzeugen eine aus wissenschaftlicher Sicht nicht eindeutige Bestimmtheit des Begriffs.

Von der wissenschaftlichen Literatur abgesehen wird der Begriff vor allem stark bei praktischen Beiträgen für Industrie 4.0, Internet of Things oder Digitalisierung benutzt. Die Beiträge haben jedoch oft wenig gemeinsam, weder bei der Darstellung, noch beim Inhalt. Es scheint oft nur um den Referenzcharakter zu gehen bzw. um den Versuch, einen Referenzcharakter herzustellen.

Die meisten Definitionen beinhalten aktuell keine umfassende Bestimmung, die alle validen wissenschaftlichen Aspekte umfasst. Falls doch, umreißen sie das Thema auf eine solch generische Art und Weise, dass sie zu großen Interpretationsraum ermöglichen. Die Entwicklung macht den Vergleich und die Interpretation von Referenzarchitektur enorm schwierig und es bedarf daher eine grundlegende Einordnung, welche die relevanten Aspekte und Ziele zusammenfasst.

Die Frage die sich an diesen Punkten anschließt ist daher, ob es überhaupt möglich ist, die aktuelle Forschung und die Entwicklung von Referenzarchitekturen einzugrenzen und ob eine klare Abgrenzung zu anderen Begriffen wie bspw. ein Referenzmodell möglich ist. Durch die identifizierten Aspekte in den vorangegangenen Kapiteln kann dies aus Sicht des Autors klar bejaht werden. Einige der Punkte finden sich nicht typischerweise bei anderen Artefakten, auch wenn die Trennung zu anderen Begriffen nicht komplett trennscharf definiert werden kann. Durch die Auswahl, Aufzählung und Aggregation ist es möglich, den Begriff exakter zu umreißen.

Basierend auf den vorgestellten Erkenntnissen wird in dieser Arbeit daher eine eigene Begriffsdefinition des Begriffs Referenzarchitektur herausgearbeitet und verwendet. Nachfolgend wird diese vorgestellt und Abgrenzung hinsichtlich der bereits vorgestellten benachbarten Begriffe vorgenommen.

*„Eine **Referenzarchitektur** ist eine abstrakte Architektur, die den Menschen die Entwicklung von Systemen, Lösungen und Applikationen erleichtern soll, indem sie Wissen bereitstellt und einen Rahmen zur Entwicklung vorgibt. Die Beziehung*

zwischen Referenzarchitektur und konkreter Architektur ist dadurch gekennzeichnet, dass Gegenstand oder Inhalt der Referenzarchitektur bei der Konstruktion der konkreten Architektur des jeweiligen zu entwickelnden Systems (wieder-)verwendet werden. Die Referenzarchitektur besitzt einen technischen Fokus, verbindet diesen jedoch mit dem dazugehörigen Fachwissen der jeweiligen Domäne. Sie bildet durch ihre Ausprägung und ihren Inhalt ein gemeinsames Rahmenwerk, um die detaillierten Diskussionen aller bei der Entwicklung beteiligten Stakeholder geführt werden können“ (Reidt et al., 2018a).

Durch diesen Begriff wird aufbauend auf dem Referenzmodellbegriff von vom Brocke (2015, S. 34) ein Referenzarchitekturbegriff abgeleitet, der eine Referenzarchitektur vor allem als Artefakt zur Unterstützung der Entwicklung von Systemen, Lösungen und Applikationen in Form von bereitgestelltem Wissen und dazugehörigen Rahmen sieht. Analog zum Referenzmodellbegriff bei vom Brocke (2015) wird der Referenzarchitekturbegriff in dieser Dissertation in Verbindung mit einer konkreten Architektur gesetzt. Weiter wird der technische und fachliche Inhalt der Referenzarchitektur hervorgehoben. Ihre Ausprägung ist zwar technisch orientiert, jedoch ergibt sie nur mit dem dazugehörigen Fachwissen bzw. Domänenwissen eine Referenzarchitektur in dem hier verstandenen Sinne. Dieser bedingt auch, dass die Referenzarchitektur nicht nur im technischen Sinne zur Unterstützung der Entwicklung herangezogen wird, sondern auch als Element zur Förderung der Kommunikation aller an der Entwicklung beteiligten Stakeholder. Hier sind bewusst Parallelen zu der Definitionen von Lin et al. (2015) gezogen worden. Ansonsten ist die Definition absichtlich breit gewählt worden, um die jeweiligen Ausprägungen nicht zu stark einzuschränken. Jedoch wird im Gegensatz zu noch offeneren Definitionen wie von Bass et al. (2003) und des VDI/VDE Fachausschuss Industrie 4.0 (2017), insbesondere mit der im späteren Verlauf des Kapitels vorgestellten Klassifikation von Eigenschaften von Referenzarchitektur, ein klarerer Rahmen für den Begriff gezogen. Dadurch wird eine zielgerichtetere Entwicklung ermöglicht.

Die Abgrenzung des Begriffs der Referenzarchitektur gegenüber den zu Beginn des Kapitels verwendeten Begriffen ist Abbildung 5 illustriert. Dort ist das Verhältnis der benutzten Begriffe hierarchisch von links nach rechts dargestellt. Dabei bestimmt die X-Achse den Abstraktionsgrad des jeweiligen Begriffs. Eine Referenzarchitektur ist ein Referenzmodell im Sinne von vom Brocke (2015, S. 34), die jedoch um einen technischen Architekturfokus erweitert wurde. Dabei muss ein Referenzmodell nicht zwangsläufig als Vorstufe der Referenzarchitektur bestehen. Die Referenzarchitektur kann u. a. Informationen aus bisherigen Architekturen, Architekturmustern, Erfahrungen, Best Practices und Domänenwissen enthalten. Weiter können Referenzmodelle, Teilimplementierungen sowie Variationspunkte Elemente einer Referenzarchitektur sein. Dabei spielen Domäneninformationen und aus den adressierten Punkten abgeleitete Architektur- und Domänenanforderungen eine besondere Rolle. In einer Referenzarchitektur soll demnach der Teil der Informationen vorhanden sein, der die Entwicklung von konkreten Softwarearchitekturen positiv beeinflussen kann und einen entsprechenden Rahmen zur Entwicklung zur Verfügung stellt. Die Unterscheidung zwischen Referenzmodell und Referenzarchitektur wird am technischeren Charakter der Referenzarchitektur gegenüber dem Referenzmodell festgemacht, da diese Architekturentscheidungen und ggfs. schon Softwareelemente beinhaltet.

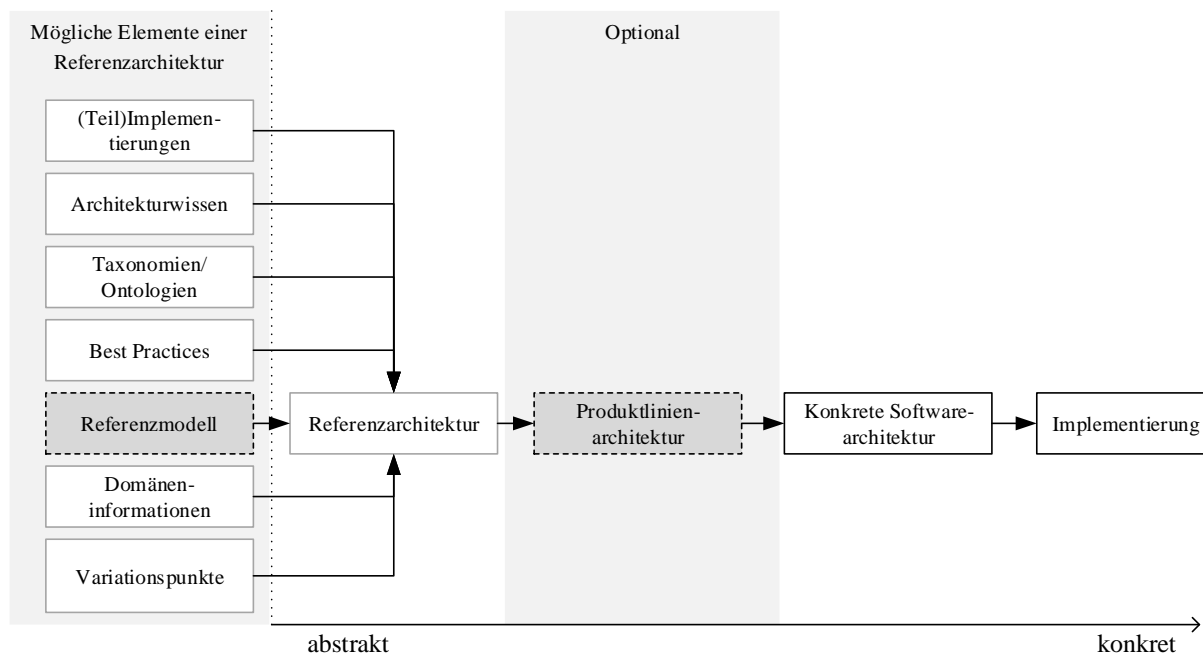


Abbildung 5: Abstraktionsniveau und Architekturhierarchie von Referenzarchitekturen

Quelle: In Anlehnung an Reidt et al. (2018a)

Eine Referenzarchitektur unterscheidet sich von einer Produktlinienarchitektur in dem Sinne, dass sie deutlich abstrakter (Nakagawa et al., 2011b, S. 210) und unvollständig ist (Taylor et al., 2009, S. 594). Unvollständig bedeutet hierbei, dass zwar Varianten und Variationspunkte von bestimmten abzuleitenden Architekturen genannt und enthalten sein können, dies jedoch kein Muss ist. Referenzarchitekturen können sich auch allein auf Informationen einer Domäne und/oder Applikation beziehen, die für die Ableitung der konkreten Architektur Rahmenbedingungen setzen oder die Ableitung anleiten. Ein weiterer Unterscheidungsaspekt ist, dass Produktlinienarchitekturen für eine spezifische Organisation und deren Produkte erschaffen werden (Galster et al., 2012, S. 430). Referenzarchitekturen können auch überorganisational, also unternehmensübergreifend und produktunabhängig, definiert sein. Produktlinienarchitekturen werden daher als spezielle Art einer Referenzarchitektur verstanden, die für eine fest definierte Menge an Produkten geschaffen werden und deren konkrete Softwarearchitekturen verbinden, indem ein Architekturkern mitsamt Variationen enthalten ist. Die Erstellung einer Produktlinienarchitektur ist damit optional und nicht zwingend, um eine konkrete Architektur von einer Referenzarchitektur abzuleiten.

Die konkrete Softwarearchitektur ist eine Instanziierung der Referenzarchitektur bzw. der Produktlinienarchitektur und gibt eine konkrete Architektur für eine spätere Implementierung vor. Die konkrete Architektur wird von der Referenzarchitektur abgeleitet und konkretisiert, was in der Referenzarchitektur bisher nur abstrakt beschrieben wurde. Dabei werden auf Basis von evtl. vorhandenen Variationspunkten die Varianten ausgewählt, die zur Umsetzung der der Architektur zugrundeliegenden Anwendungsfälle nötig sind. Die Implementierung bezeichnet hingegen die Umsetzung der konkreten Architektur. Im Kontext der Softwareentwicklung wäre die Implementierung das entstandene Softwaresystem, welches durch die Architektur beschrieben wird.

2.2.4.4 Klassifikation von Referenzarchitekturen

Basierend auf der vorgestellten Definition und den vorgestellten Eigenschaften und Zielen lässt sich folgende Klassifikation für Referenzarchitekturen erstellen.

Charakteristik	Ausprägungen		
Abstraktionsgrad	Detailliert (Codebasis)	Mischform	Abstrakt
Technologieneutralität	Ja	Mischform	Nein
Industriefokus	Industriespezifisch		Industrieübergreifend
Produktfokus	Fokus auf ein Produkt	Produktfamilie	Produktübergreifend
Unternehmensfokus	Unternehmensspezifisch		Unternehmensübergreifend
Referenzcharakter	Bezugspunkt	Allgemeingültig	Empfehlungscharakter
Variationspunkte	Enthalten		Nicht enthalten
Technisch Fokus	Rein technisch		Technisch mit Domäneninformationen
Vollständigkeit	Vollständig		Unvollständig
Praxis- oder Forschungsgetrieben	Praxis		Forschung
Ziel	Standardisierung		Erleichterung
Vorgehensweise	Induktiv	Kombination	Deduktiv
Enthaltenes Wissen	Architekturwissen	Softwareelemente	Richtlinien
	Weiteres Wissen:		

Tabelle 2: Klassifikationsschema Referenzarchitekturen

Quelle: Eigene Darstellung

Die letzten beiden Charakteristiken wurden zusätzlich zu den diskutierten Eigenschaften hinzugefügt. Die Charakteristik Vorgehensweise unterteilt eine Referenzarchitektur nach der Art der Erstellung. Entweder wird sie induktiv, deduktiv oder in Kombination der beiden Vorgehensweisen erstellt. Bei der induktiven Vorgehensweise wird aus dem Speziellen eine Verallgemeinerung extrahiert. Im Sinne einer Referenzarchitektur wäre dies bspw. der Fall, wenn aus vielen konkreten Architekturen auf eine Referenzarchitektur geschlossen wird. Die deduktive Vorgehensweise schließt vom Allgemeinen aufs Spezielle und würde im übertragenen Sinne daher von Normen, Vorgaben und übergreifenden Architekturen auf eine spezielle Referenzarchitektur schließen. Die nächste Charakteristik stellt die Art von Wissen dar, welches durch die Referenzarchitektur zu Verfügung gestellt wird. Die Ausprägungen umfassen Architekturwissen (unter das Architekturmuster, -techniken etc. fallen), Softwareelemente, Richtlinien und weitere zu nennende Punkte.

Anhand dieser Klassifikation können die verschiedenen Ausprägungen von Referenzarchitekturen eingeordnet und systematisch verglichen werden. Daneben dient die Klassifikation der Auswahl von Methoden zur Erstellung, die im nächsten Kapitel behandelt werden. Die Klassifikation der RAI nach dem vorgestellten Schema ist in Tabelle 3 dargestellt. Die auf die RAI zutreffenden Ausprägungen der Charakteristiken sind in Grau gekennzeichnet.

Charakteristik	Ausprägung		
Abstraktionsgrad	Detailliert (Codebasis)	Mischform	Abstrakt
Technologieneutralität	Ja	Teilweise	Nein
Industriefokus	Industriespezifisch		Industrieübergreifend
Produktfokus	Fokus auf ein Produkt	Produktfamilie	Produktübergreifend
Unternehmensfokus	Unternehmensspezifisch		Unternehmensübergreifend
Referenzcharakter	Bezugspunkt	Allgemeingültigkeit	Empfehlungscharakter
Variationspunkte	Enthalten		Nicht enthalten
Technisch Fokus	Rein technisch		Technisch mit Domäneninformationen
Vollständigkeit	Vollständig		Unvollständig
Praxis- oder Forschungsgetrieben	Praxis		Forschung
Ziel	Standardisierung		Erleichterung
Vorgehensweise	Induktiv	Kombination	Deduktiv
Enthaltenes Wissen	Architekturwissen	Softwareelemente	Richtlinien
	Weiteres Wissen:	Optionale und generischen Anforderungen	

Tabelle 3: Angewandtes Klassifikationsschema auf die RAI

Quelle: Eigene Darstellung

Die Referenzarchitektur eines Informationssystems zur Unterstützung der Instandhaltung hat einen hohen Abstraktionsgrad, sie ist technologieneutral, gibt aber teilweise Empfehlungen, mit welchen Technologien etwas umgesetzt werden könnte und nennt dazu bspw. verschiedene Maschinenprotokolle (z. B. OPC UA). Die Referenzarchitektur ist industrieübergreifend, da sie für jegliche Form der Instandhaltung von Produktionsanlagen eingesetzt werden kann. Es gibt keinen Fokus auf Automotive, Luftfahrt oder bestimmte andere Branchen. Der Unternehmensfokus ist nicht auf ein einzelnes Unternehmen gerichtet, da die Referenzarchitektur sich an ein umfassendes Publikum richtet und die Entwicklung von Instandhaltungssystemen im Allgemeinen erleichtert. Die Referenzarchitektur soll als Bezugspunkt mit Empfehlungscharakter dienen. Aus diesem Grunde wird die Referenzarchitektur auch instanziiert und von Benutzern hinsichtlich ihres Empfehlungscharakters evaluiert. Von einer Allgemeingültigkeit kann aufgrund des Vorgehens bzgl. der Anforderungen aus Literatur und Praxis in vier Fällen ausgegangen werden, jedoch ist dies nicht beweisbar. Daher wird auf dieses Merkmal verzichtet.

Die Referenzarchitektur besitzt Variationspunkte und offene Erweiterungsmöglichkeiten. Diese sind nicht ausformuliert und informell gehalten. Aus diesem Grund ist die Referenzarchitektur auch unvollständig. Sie beinhaltet neben technischen Aspekten auch Domäneninformationen und bietet ein Vokabular, um die stakeholderübergreifende Kommunikation zu erleichtern. Sie ist praxis- wie auch forschungsgetrieben, sodass zum einen bestehende Lösungen zusammengefasst werden und auf bewährte Teilaspekte zurückgegriffen wird. Zum anderen werden jedoch viele innovative Elemente und die zukünftige Entwicklung von der Referenzarchitektur beschrieben, sodass ein innovativer Charakter erhalten bleibt. Weiterhin basieren die Informationen nicht auf bestehenden Systemen, sondern auf Anforderungen auf Literatur und Praxis, was für eine Forschungsorientierung steht.

Das Ziel der Referenzarchitektur ist die Erleichterung der Entwicklung durch das bereitgestellte Wissen, die optionalen und generischen Anforderungen, den daraus entstandenen umfassenden Blick auf das System und die einfache Darstellung. Die Referenzarchitektur wird daher aus Kombination einer deduktiven und induktiven Vorgehensweise erstellt. Auf Basis aus der Literatur abgeleiteter Anforderungen an Instandhaltungssysteme werden Anforderungen an spezifische Lösungen mehrerer Unternehmen gesammelt und aus diesen einzelnen, speziellen Lösungen auf eine generische, allgemeine Lösung geschlossen. Daneben werden ausgehend von technologischen Möglichkeiten und Standards entsprechende Änderungen an der Referenzarchitektur Top-down vorgenommen.

2.3 Methodik zur Erstellung von Referenzarchitekturen

Trotz des starken Anstiegs der Publikationen von Referenzarchitekturen finden sich wenige, die sich zu verwendeten Methoden oder Vorgehen zur Erstellung äußern oder diese behandeln. Dies liegt im Allgemeinen daran, dass ein Großteil der Referenzarchitekturen ad hoc erstellt werden, das heißt ohne einen systematischen Prozess (Nakagawa et al., 2014b, S. 63). Dies führt jedoch zu zwei Nachteilen: Zum einen sind die Referenzarchitekturen und deren Erstellung nicht nachvollziehbar und können so schlechter überprüft werden. Zum anderen fehlen Anleitungen, Best Practices, Prozesse und Methoden, um Referenzarchitekturen zu bauen und zu evaluieren (Angelov et al., 2014b, S. 1). Dabei ist ein systematischeres Vorgehen von großem Interesse, wodurch auch effektivere Referenzarchitekturen erstellt werden könnten (Nakagawa et al., 2014b, S. 63). Nachfolgend werden daher aktuelle Methoden und Vorgehensweisen in der Literatur vorgestellt und auf Basis der Klassifikation der RAI diskutierte. Anschließend wird der Ansatz zur Erstellung der Referenzarchitektur dieser Arbeit vorgestellt.

2.3.1 Methoden zur Erstellung von Referenzarchitekturen

In der Literatur finden sich verschiedene Ansätze zur Erstellung von Referenzarchitekturen. Diese unterscheiden sich stark in ihrem Grad an Abstraktion und der Art der Referenzarchitektur, für den diese entworfen wurden.

Methoden wie PuLSE-DSSA (Bayer et al., 2004) orientieren sich an Referenzarchitekturen, die identisch mit Produktlinienarchitekturen sind. Die Basis für die iterative Entwicklung dieser Methode der Referenzarchitektur bilden generische Szenarien, diese werden aufgrund von Erfahrungen mit bestehenden Lösungen abgebildet. Aus bestehenden Architekturen werden durch Reverse Engineering Gemeinsamkeiten und Lösungsmöglichkeiten für die Szenarien gebildet und diese ausgestaltet. Damit wäre diese Methode nur für praxisgetriebene Referenzarchitekturen anwendbar. Spezifische Abänderungen für bestimmte Produkte einer Produktfamilie werden als Variationspunkte dargestellt. Ordnet man diese Art der Referenzarchitekturen nach der Klassifikation aus Tabelle 2, wäre der Abstraktionsgrad dieser Art von Referenzarchitekturen niedrig, da sie bis auf die Codebasis ausdetailliert sind, sie wären nicht technologieneutral, wären für eine Produktfamilie erstellt worden mit einem reinen technischen Fokus. Das Vorgehen orientiert sich aufgrund der Konkretheit und der kompletten Ausgestaltung am Vorgehen zur Entwicklung einer konkreten Softwarearchitektur und wäre für die Erstellung der RAI ungeeignet.

Ähnlich technisch sind die Ausführungen von Pohl et al. (2005), die jedoch eher Handlungsempfehlungen bzw. technische Aspekte bei der Erstellung diskutieren. Hier geht es vor allem um den Aspekt, die wiederverwendbaren technischen Artefakte festzulegen und Variationspunkte zu setzen. Konkrete Empfehlungen und Schritte werden jedoch nicht genannt.

Cloutier et al. (2009) beschreiben eine Vorgehensweise auf hoher Abstraktionsebene, die das Sammeln von Informationen für Referenzarchitekturen näher beleuchtet, die Evolution von Referenzarchitekturen behandelt und davon spricht, dass das Erstellen von Referenzarchitekturen hauptsächlich auf dem Extrahieren von Informationen aus bestehenden Architekturen basiert.

Galster und Avgeriou (2011) erkennen das Problem der fehlenden systematischen Erstellung von Referenzarchitekturen und stellen einen Ansatz vor, Referenzarchitekturen empirisch

fundiert zu erstellen. Dieser fußt auf den Ansätzen zur Erstellung von Referenzmodellen von Ahlemann und Gastl (2007) und Karow et al. (2008). Galster und Avgeriou (2011) leiten basierend auf diesen Methoden ein Vorgehen mit sechs Schritten zur Erstellung von Referenzarchitekturen ab. Diese sechs Schritte sind:

- **Schritt 1 – Festlegen des Typs der Referenzarchitektur:** Zuerst wird anhand des Klassifikationssystems von Angelov et al. (2009) der Typ der Referenzarchitektur festgelegt. Hier sind insbesondere zwei Ausprägungen der Charakteristiken von Referenzarchitekturen entscheidend für deren Erstellung. Zum einen das Ziel der Referenzarchitektur, welches entweder die Standardisierung und/oder die Erleichterung der Entwicklung ist. Zum anderen ob die Referenzarchitektur klassisch praxisorientiert ist und damit auf bewährten Architekturen und Lösungen basiert oder ob diese zukunftsorientiert ist und noch nicht existierende Lösungen beschreibt.
- **Schritt 2 – Festlegen der Designstrategie:** Abhängig vom Typ der Referenzarchitektur wird diese entweder aus Best Practices bzw. Projekterfahrung und bestehenden Architekturen gebildet oder von Grund auf neu gebildet.
- **Schritt 3 – Empirische Akquirierung von Daten:** In diesem Schritten werden Quellen und Personen identifiziert, von denen Informationen extrahiert werden können, die zur Erstellung notwendig sind. Die Art der Quellen und Informationen unterscheiden sich hinsichtlich des Typs der Referenzarchitektur. Praxisorientierte Referenzarchitekturen können bspw. Informationen und Architekturen beziehen, wohingegen forschungsgetriebene Referenzarchitekturen andere Arten von Informationen benötigen. Als nächstes werden die Informationen aufgenommen und dokumentiert. In diesem Schritt werden Stakeholder definiert, die zum einen die Referenzarchitektur zur Instanziierung von konkreten Architekturen benutzen und zum anderen die Stakeholder, welche die instanziierten konkreten Architekturen benutzen. Zusätzlich werden architekturrelevante Anforderungen bestimmt.
- **Schritt 4 – Konstruktion der Referenzarchitektur:** In diesem Schritt wird die Referenzarchitektur modelliert und in verschiedene Architektursichten unterteilt. Dabei wird zwischen gemeinsamen und spezifischen Elementen auf Basis der Informationen aus Schritt 3 unterschieden. Gemeinsame Elemente sind jene Elemente, die in jeder zugrundeliegenden Architektur enthalten sind und demnach einen Kernbestandteil der Referenzarchitektur darstellen müssen.
- **Schritt 5 – Ermöglichung der Variabilität:** In diesem Schritt wird die Variabilität in Form von spezifischen Elementen bspw. als Variationspunkte in die Referenzarchitektur eingefügt.
- **Schritt 6 – Evaluation der Referenzarchitektur:** Abhängig vom Typ und damit vom Ziel der Referenzarchitektur wird die Nützlichkeit oder die Unterstützung für die effiziente Adaption und Instanziierung im Vordergrund stehen.

Das Konzept von Galster und Avgeriou (2011) ist relativ generisch und abstrakt, insbesondere hinsichtlich der Auswahl der Informationen und der Konstruktion, umfasst jedoch eine klare Eingruppierung der nötigen Schritte in Abhängigkeit von bestimmten Referenzarchitekturarten. Den Fokus der Beschreibung bilden jedoch praktische Referenzarchitekturen, die auf bestehenden Architekturen und Best Practices aufbauen. Architekturen, die für neue Systeme ausgelegt sind und nicht auf bewährten Systemen aufbauen können werden erwähnt, jedoch werden kaum konkrete Schritte für die Erstellung dieser genannt. Dadurch ist dieser Ansatz für

die Erstellung der RAI nicht ausreichend, sondern kann nur einen Rahmen für die Vorgehensweise vorgeben.

Nakagawa et al. (2014a) stellen den Prozess ProSA-RA vor, der die Erstellung, Repräsentation und Evaluation von Referenzarchitekturen systematisieren soll. Bei diesem wird zuerst der Anwendungsbereich der Referenzarchitektur festgelegt, bspw. die Zieldomäne. Anschließend werden folgende vier Schritte durchgeführt:

- **Schritt 1 – Auswahl und Untersuchung von Informationsquellen:** Bei diesem Prozessschritt werden die relevanten Informationsquellen ausgewählt, sodass aus diesen die nötigen Informationen über Prozesse, Aktivitäten und Aufgaben, die durch das Softwaresystem der Zieldomäne unterstützt und abgedeckt werden sollen, extrahiert werden können. Im Vergleich zu „normalen“ konkreten Architekturen wird umfassenderes Domänenwissen für die Erstellung von Referenzarchitekturen benötigt, da diese die Grundlage für mehrere unterschiedliche Systeme darstellt. Als potentielle Informationsquelle werden Personen (Kunden, Benutzer, Entwickler, etc.), Softwaresysteme, Publikationen und Domänenontologien o. Ä. genannt.
- **Schritt 2 – Identifizierung von Architektur Anforderungen:** Ausgehend von den Informationen aus Schritt 1 müssen die relevanten Informationen der Applikationsdomäne identifiziert und als Resultat die Anforderung an die Referenzarchitektur aufgestellt werden. Dabei untergliedert sich diese Aufgabe in die Identifikation von Systemanforderungen (funktionale und nicht funktionale Anforderungen) durch Techniken der Anforderungserhebung und in die Etablierung von Referenzarchitektur Anforderungen, die auf den Systemanforderungen aufbauen und mehrere Systemanforderungen umfassen können. Sie besitzen daher eine höhere Abstraktionsstufe und können funktionale wie auch nicht funktionale Anforderungen umfassen. Anschließend werden Domänenkonzepte herausgearbeitet, die wiederum die Referenzarchitektur Anforderungen umfassen und demzufolge einen noch höheren Abstraktionsgrad besitzen. Diese Domänenkonzepte sind die Grundlage für das architekture Design der Referenzarchitektur.
- **Schritt 3 – Etablierung der architekturealen Beschreibung der Referenzarchitektur:** Die Repräsentation und damit die Modellierung der Referenzarchitektur wird in diesem Schritt so bestimmt, dass diese einer großen Anzahl von Stakeholdern verständlich ist. In den meisten Fällen ist die Modellierung daher informell. Auch bei diesem Vorgehen werden verschiedene Architektursichten empfohlen, die jedoch bis auf eine Konzeptsicht sehr technisch gehalten sind (technische Modulsicht, Laufzeitumgebungssicht, Verteilungssicht).
- **Schritt 4 – Evaluation der Referenzarchitektur:** Im letzten Schritt wird die Referenzarchitektur mit anzupassenden Methoden wie bspw. die Architectural Tradeoff Analysis, Software Architecture Analysis Method oder einer checklistbasierten Evaluation hinsichtlich ihrer Nützlichkeit untersucht. Hier spielen insbesondere die Verständlichkeit und die Darstellung der Referenzarchitekturen eine Rolle.

Ähnlich wie das Vorgehen von Galster und Avgeriou (2011) ist das Vorgehen von Nakagawa et al. (2014a) sehr generisch, geht dabei aber konkreter auf die Informationsquellen und die nötigen Stakeholder ein, wodurch eine eindeutigere Hilfestellung gegeben wird. Das Vorgehen von Nakagawa et al. (2014a) orientiert sich dabei auch an technischere Arten von Referenzarchitekturen im Sinne von Produktlinienarchitekturen. Eine Unterscheidung des Vorgehens nach Art der Referenzarchitektur findet nicht statt. Die Unterteilung von

Anforderungen hinsichtlich Systemanforderungen, Referenzarchitekturanforderungen und Domänenkonzepten kann als Grundlage zur Erstellung der RAI genutzt werden.

2.3.2 Verwendete Methodik zur Erstellung von Referenzarchitekturen

Die vorgestellten Referenzarchitekturen bieten sich als Rahmenwerk für die Erstellung einer Referenzarchitektur an, jedoch beinhalten sie zu wenig Hilfestellung für die Erstellung der Art der Referenzarchitektur, der die RAI zuzuordnen ist. Für eine forschungsorientierte, technologieneutrale Referenzarchitektur mit Domänenwissen, das nicht aus offenen, bestehenden Architekturen extrahiert werden kann, finden sich kaum konkretere Handlungsempfehlungen und Schritte bei den vorgestellten Methoden. Zwar werden bei Nakagawa et al. (2014a) die Anforderungen an eine Referenzarchitektur betont und Techniken der Anforderungsaufnahme verwendet, jedoch wird auch hier nicht der Erstellungsprozess anhand von Anforderungen, sondern nur eine Aggregation von Anforderungen beschrieben. Um diese Herausforderungen zu adressieren und eine forschungsorientierte Referenzarchitektur ohne explizit vorhandene Basisarchitekturen zu erstellen, wird von Reidt et al. (2017a) eine Methode beschrieben, wie eine Referenzarchitektur anhand von individuellen Unternehmensanforderungen erstellt werden kann. Die Methode ist eine Mischung aus induktiver Vorgehensweise, die von bestehenden Teilsystemen und Literatur auf die Gemeinsamkeiten eines übergreifenden Systems schließt und einer deduktiven Vorgehensweise, die von Standards und Zielsetzungen aus die Erstellung der Referenzarchitektur beeinflusst. Dabei müssen bei der induktiven Vorgehensweise keine bestehenden Architekturen vorhanden sein, es reichen Anforderungen an solche aus, um Gemeinsamkeiten zu destillieren.

Die RAI wird anhand dieses Vorgehens erstellt, welches sich in den in Kapitel 1 vorgestellten Forschungsrahmen einbettet. Dabei stellen die nachfolgend vorgestellten Schritte Teile der Analyse-, Entwurfs- und Evaluationsphase aus Abbildung 2 dar.

- **Schritt 1 – Festlegung des Ziels der Referenzarchitektur:** Im ersten Schritt erfolgt die Definition des Ziels der Referenzarchitektur. Damit verbunden ist die Festlegung des Betrachtungsrahmens. Der Betrachtungsrahmen sollte so festgelegt werden, dass die Ziele, die mit der Referenzarchitektur verfolgt werden mit dem Fokus dieser erreicht werden können. Ausgehend vom Ziel und dem Betrachtungsrahmen kann die Art der Referenzarchitektur auf Basis des in Tabelle 2 dargestellten Klassifikationsschemas gewählt werden. Verbunden mit diesen Aspekten ist die Durchführung einer Stakeholderanalyse, da die relevanten Personen bestimmt werden müssen, die durch die Referenzarchitektur betroffen sind und mit ihr oder dem System arbeiten müssen. Mit diesen gewählten Punkten sollte auch die Darstellungsart der Referenzarchitektur festgelegt und im weiteren Verlauf verfeinert werden, damit die Art der Darstellung das Erreichen der Ziele unterstützt. Es wird empfohlen in diesem Schritt das Sichtenkonzept von Kruchten (1995) mit angepassten Sichten in Abhängigkeit der verfolgten Zielen zu wählen. Dabei werden eine funktionale bzw. logische Sicht und eine Prozesssicht empfohlen.
 - Im Fall der RAI ist das Ziel die Erleichterung der Entwicklung durch die Bereitstellung eines Rahmenkonzeptes zur Erstellung von IIS und durch die Förderung der domänenübergreifenden Zusammenarbeit. Der Betrachtungsrahmen umfasst demzufolge den gesamten Bereich von Informationssystemen zur Unterstützung der Instandhaltung. Um diesen Bereich abzubilden sowie eine

langlebige und erweiterbare Lösung zu erreichen, wurde eine abstrakte, technologieneutrale und domänenübergreifende Darstellungsart gewählt. Um die Erweiterbarkeit und die Anpassbarkeit noch weiter zu fördern, wurden zudem Variationspunkte in die RAI integriert. Die Modellierung der RAI basiert dabei auf dem Sichtkonzept von Kruchten (1995), sodass mehrere unterschiedliche Sichten auf die RAI möglich sind.

Stakeholder der Referenzarchitektur bei der Instanziierung sind vornehmlich alle an der Entwicklung beteiligten Stakeholder des Systems, da diese die gewünschten Funktionalitäten für die konkrete Architektur ableiten müssen. Darüber hinaus muss mit dem System fachübergreifend gearbeitet werden. Die Referenzarchitektur wird aus diesem Grund die domänenübergreifende Zusammenarbeit fördern. Dabei sind neben der einfachen Darstellung auch sehr komplexe Wissens Elemente enthalten, die sich vornehmlich an die Entwickler und Architekten des Systems richten.

Stakeholder bei der Benutzung sind alle bei der Instandhaltung beteiligten Personen, seien es die Instandhalter selber, Maschinenführer, externe Dienstleister der Instandhaltung, Maschinenhersteller, die Instandhaltungskomponenten einbinden wollen oder Manager der Instandhaltung und Controller ebener. Diese Personen sollten auch bei der Instanziierung bei den für sie relevanten Teilen der Referenzarchitektur eingebunden werden.

- **Schritt 2 – Literaturrecherche:** Bevor empirisch Daten erhoben, Wissensquellen identifiziert und ausgewählt werden, ist ein grundlegendes Verständnis der Domäne und ein Aufbau einer Wissensbasis erforderlich. Diese Wissensbasis sollte den aktuellen Stand der Wissenschaft und Technik beinhalten, die zur Erstellung der Referenzarchitektur und den daraus resultierenden Systemen notwendig ist. Ein Weg dies zu erreichen ist es, eine wissenschaftliche Literaturrecherche durchzuführen, die sich an den nötigen Aspekten der zu entwickelnden Referenzarchitektur orientiert. Ein Vorteil dieser Vorgehensweise ist, dass durch eine Literaturanalyse Aspekte identifiziert werden können, welche über den Praxisfall hinausgehen.
 - Im Falle der RAI wurde der aktuelle Status quo zu Instandhaltungsthemen und -systemen im wissenschaftlichen Kontext ergründet, weitere Komponenten für Instandhaltungssysteme erfasst und möglichst verwandte Referenzarchitekturen identifiziert. Eine Darstellung dieser Untersuchung findet sich in diesem Kapitel sowie in den Kapiteln 3 und 4 wieder.
- **Schritt 3 – Situation und Anforderungen aus der Praxis:** Um die Praxisrelevanz einer Referenzarchitektur sicherzustellen und die wichtigsten Informationen zur Erstellung dieser zu sammeln, müssen umfassende und generalisierbare Anforderungen für spezifische Einzelarchitekturen aufgenommen werden. Dazu müssen Anforderungen an Systeme, für welche die Referenzarchitektur die Grundlage bilden soll, bei unterschiedlichen Fällen aus der Praxis aufgenommen werden. Anschließend müssen die untersuchten Fälle zusammen betrachtet einen umfassenden und nicht spezifischen Blick auf die Domäne bzw. das zu entwickelnde System ermöglichen. Dieser Schritt ähnelt dem Schritt 3 – empirische Akquirierung von Daten aus dem Ansatz von Galster und Avgeriou (2011), jedoch werden die zu extrahierenden Daten auf Anforderungen an das zu entwickelnde System eingeschränkt, ähnlich zu dem Ansatz von Nakagawa et al. (2014a), der auch die Ableitung von Anforderungen als Ziel besitzt. Der Unterschied zu Nakagawa et al. (2014a) liegt hierbei darin, dass in dem hier vorgestellten Vorgehen Anforderungen an nicht existierende Einzelsystem aufgenommen werden, um aus diesen Informationen für eine übergreifende

Lösung zu extrahieren. Mit den Schritten 2 und 3 wird das Domänenwissen, welches zur Erstellung der Referenzarchitektur nötig ist, erlangt, und von praktischer wie auch wissenschaftlicher Seite untersucht.

- **Schritt 4 – Erstellung einer Domänenbeschreibung und eine Vereinheitlichung durch Festlegung/Standardisierung von Begrifflichkeiten:** Die Domäne der Referenzarchitektur sollte eindeutig beschrieben und nötige Begriffe identifiziert und vereinheitlicht werden. Gegebenenfalls müssen hier schon Standards und Definitionen für einige der Begrifflichkeiten oder Aspekte der Domäne festgelegt werden. Dabei besteht ebenso die Möglichkeit, Domänenontologien und -taxonomien für bestimmte Domänenaspekte zu erstellen.
 - Bei der RAI sind dies vor allem Definitionen zu technischen Aspekten und Begrifflichkeiten der Referenzarchitektur sowie Benennungen von Funktionen und Anforderungen. Diese werden in Kapitel 6 ausführlich präsentiert.
- **Schritt 5 – Extraktion von generischen und optionalen Anforderungen an eine Referenzarchitektur:** Anhand der Ergebnisse aus Schritt 2 und 3 werden die Anforderungen aggregiert, auf eine einheitliche Basis gebracht und in generische und optionale Anforderungen unterteilt. Generisch sind Anforderungen, welche als allgemeingültig angesehen werden können und bei allen untersuchten Fällen der Praxis (und Literatur) vorkommen oder vorkommen sollten. Optionale Anforderungen sind spezifisch für einzelne Anwendungsfälle und nur bei diesen umzusetzen. Ergebnis dieses Schrittes sind Listen und Beschreibungen von generischen sowie optionalen Anforderungen. Nötige Unklarheiten sind mit den jeweiligen Ansprechpartnern der Anforderungen möglichst im Vorfeld zu klären. Neben der induktiven Vorgehensweise sollten in diesem Schritt aufbauend auf den Informationen aus 2 und 3 auch Vorgaben für die Architektur gemacht werden, die als essentiell aus Erstellersicht anzusehen sind. Damit können bestimmte Aspekte, die mit der Referenzarchitektur betont werden sollen, hervorgehoben oder die Einhaltung von Standards oder übergeordneten Vorgaben ermöglicht werden, die evtl. nicht aus Anforderungen extrahiert werden können.
 - Das genaue Vorgehen im Fall der RAI wird in Kapitel 4 beschrieben.
- **Schritt 6 – Überführung der gefundenen Anforderungen in n:1 Beziehungen in funktionale, logische Module:** Anhand der generischen und optionalen Anforderungen werden logische Module gebildet, sodass jede Anforderung durch ein Modul umgesetzt wird. Ein Modul kann hierbei beliebig viele Anforderungen umsetzen. Ziel ist es, dass ein Modul möglichst eine logisch in sich geschlossene Einheit bildet und für eine Funktion des Systems verantwortlich ist. Die Aufteilung und der Schnitt der Module sollte nach Aspekten der Kapselung und des Separations of Concerns (Laplante, 2007) getätigt werden. Eine einfache technische Umsetzbarkeit wird hierdurch gewährleistet. Lassen sich Anforderungen nicht auf ein Modul aufteilen, bedarf es einer Ausdifferenzierung bzw. Aufspaltung der Anforderungen. Die Module können wiederum generisch oder optional sein. Je nachdem ob hauptsächlich generische oder optionale Anforderungen in ihnen enthalten sind. Weiterhin ist in den Modulen die jeweilige Beschreibung und Ausarbeitung dieser in Hinsicht Umsetzbarkeit, Intelligenzverteilung und Identifizierung von Verbindungen zwischen Modulen enthalten. Weiterhin soll in diesem Schritt die Verbindung der Module zu den potentiell möglichen Sichten ausgearbeitet werden.
 - Die genaue Darstellung von Modulen, die jeweilige Aufteilung von Anforderungen und der Inhalt der Module wird in Kapitel 6.2 beschrieben. In der RAI wird zusätzlich eine Verteilungssicht eingeführt (vgl. Kapitel 6.3), welche die Verteilung

von Modulen auf verschiedenen Entitäten und Endgeräten aufzeigt. Dadurch soll die Diskussion um Intelligenzverteilung und Dezentralität von Architekturen erleichtert und diese Aspekte in die Planung von konkreten Architekturen einbezogen werden.

- **Schritt 7 – Erstellung von Referenzprozessen:** Unter Einbezug der logischen Module sowie Anforderungen der jeweiligen Stakeholder sollen in diesem Prozess Referenzprozesse erstellt werden. Diese Referenzprozesse sind möglichst mit den technischen Aspekten der Referenzarchitektur verbunden. Referenzprozesse bauen in ihrer Ausprägung auf den Kern von Referenzarchitekturdesign von Bayer et al. (2004, S. 9) mithilfe von generischen Szenarien auf, erweitern diese jedoch um den Referenzcharakter durch das Zusammenfassen von einzelnen Prozessen und der Verbindung mit den logischen Modulen aus Schritt 6. Die Prozesse können dabei aus spezifischen Anwendungsfällen, User Stories oder ähnlichen Szenarien gewonnen werden. Andere bereits erstellte Sichten sind in diesem Schritt in Verbindung zu der Prozesssicht und ihren Referenzprozessen zu setzen. Dadurch dient die Prozesssicht als Basis und Kontrollfunktion für alle weiteren Sichten.
 - Referenzprozesse wurden in der RAI durch das Zusammenfassen einer Vielzahl an Use Cases erzeugt. Diese werden mit der funktionalen Sicht, Use Case Sicht sowie der Verteilungssicht verbunden. Der Referenzprozess der RAI wird auch als Grundaktivität bezeichnet. Detailliert wird das Vorgehen in Kapitel 6.4 beschrieben.
- **Schritt 8 – Feedbackzyklen & Anwendung der Referenzarchitektur:** Zusätzlich müssen im Kontext der Anforderungsaufnahme und Entwicklung mehrere Feedbackgespräche mit Entwicklern und Stakeholdern bzgl. der Referenzarchitektur geführt werden, um die Anforderungen nach dem jeweiligen Feedback zu verbessern. Weiterhin ist der Entwurf der Module, der einzelnen Sichten und der Anforderungen ein iterativer Prozess, da diese sich im Designprozess mehrmalig ändern.
 - Die in dieser Arbeit behandelte Referenzarchitektur wurde für die Entwicklung verschiedener Prototypen im Rahmen des Forschungsprojektes S-CPS genutzt und diente dem generellen Anforderungsmanagement und der Entwicklung des in Kapitel 7.5 vorgestellten Prototypen. Die Schritte 3-8 wurden und in mehrmaligen Iteration samt Feedback durchlaufen, um so die Referenzarchitektur stetig zu verbessern.

Durch das Vorgehen kann sichergestellt werden, dass die erarbeitete Lösung hohe Praxisrelevanz besitzt, da viele Anforderungen direkt aus dem Bedarf der untersuchten Industrieunternehmen stammen. Die damit verbundene empirische Fundierung, die von Galster und Avgeriou (2011) in ihrer Vorgehensweise gefordert wird, wird hierdurch erreicht. Zusätzlich werden konkrete Handlungsempfehlungen im Fall dieser forschungsorientierten Referenzarchitektur aufgezeigt und genannt. Insbesondere das Vorgehen bei der Erstellung der Anforderungen und die durchgängige Nachverfolgbarkeit fördert die Konkretetheit dieses Ansatzes. Durch den agilen Verbesserungsprozess kann sichergestellt werden, dass die Darstellungsart und der Inhalt optimal verstanden und angewendet werden konnte. Weiterhin stellen diese acht Schritte einen konkreten Leitfaden dar, um technologieunabhängige forschungsorientierte Referenzarchitekturen zu konstruieren.

2.4 Zusammenfassung

In diesem Kapitel wurden die Grundlagen von Referenzarchitekturen dargelegt, die zum weiteren Verständnis der Arbeit, ihrer Artefakte und Ergebnisse im Allgemeinen notwendig sind. Zuerst wurden Begrifflichkeiten definiert, welche im Kontext von Referenzarchitekturen häufig genannt werden. Verschiedene Definitionen von Referenzarchitekturen, die in der Wissenschaft und Praxis benutzt werden, wurden anschließend vorgestellt und die zugrundeliegenden Sichtweisen diskutiert. Eine starke Heterogenität konnte bei dem Verständnis des Begriffs Referenzarchitektur konstatiert werden. Dies führt zu erheblichen Herausforderungen bei der Einordnung, Anwendung und Interpretation einzelner Artefakte, die sich als Referenzarchitektur bezeichnen. Obwohl eine trennscharfe Abgrenzung des Begriffs einer Referenzarchitektur zu anderen Architektur oder Modellbegriffen nicht immer möglich ist, kann jedoch eine Klassifikation des Begriffs Referenzarchitektur und eine grundlegende Abgrenzung das allgemeine Verständnis des Begriffs und die Benutzung erleichtern.

Daher wurden aus diesen Sichtweisen und Definitionen Eigenschaften und Ziele, die in der wissenschaftlichen Literatur zu Referenzarchitekturen diskutiert werden, herausgestellt und deren Ausprägungen vorgestellt. Anhand dieses Wissens wurde die in der vorliegenden Arbeit benutzte Definition von Referenzarchitekturen herausgearbeitet, welche zum einen den Bezug zu der in dieser Arbeit verwendeten Referenzmodelldefinition nach vom Brocke (2015, S. 34) verwendet und zum anderen den technischen Aspekt und die Unterstützung des Menschen bei der Entwicklung betont. Anhand der Definition wurde die Abgrenzung und die Beziehung zu anderen Architekturbegriffen dargelegt und ein Klassifikationsschema vorgestellt, welches auf die RAII angewandt wurde. Dieses Schema kann zum Vergleich und zur Auswahl von Methoden zur Erstellung von Referenzarchitekturen genutzt werden. Damit verbundene aktuelle Entwicklungen und Methoden zur Erstellung wurden vorgestellt. Ableitend daraus wurde die in der Arbeit benutzte Vorgehensweise zur Erstellung von Referenzarchitekturen und deren Abhängigkeit zur gewählten Klassifikation der RAII präsentiert.

Ein wichtiges Fundament dieses Vorgehen ist die Erlangung von Domänenwissen zur Erstellung der Referenzarchitektur. Für die RAII ist ein Teil dieses Domänenwissen das Wissen über die Instandhaltung, deren Arten, Konzepte und Systeme. Im nächsten Kapitel werden diese Aspekte vorgestellt.

3 Instandhaltung im Kontext der Industrie 4.0

3.1 Einleitung

Ein grundlegender Baustein um effektive und effiziente Informationssysteme zu entwickeln ist es, die dazugehörige Domäne zu verstehen, für die das System entwickelt wird. Mit diesem Verständnis gelingt es die relevanten Probleme zu identifizieren und deren Lösungen so bereitzustellen, dass sie zur Domäne und den jeweiligen dort arbeitenden Nutzergruppen passen. Dies gilt ebenso für die Entwicklung von Referenzarchitekturen. Aus diesem Grund und um den Hintergrund der Dissertation zu beleuchten, wird in diesem Kapitel die Instandhaltung im Kontext der Industrie 4.0 vorgestellt. Hierzu zählt es, Begrifflichkeiten zu definieren, verschiedene Konzepte, Arten und Strategien der Instandhaltung vorzustellen und aktuelle Entwicklungen der selbigen zu beleuchten. Damit wird das Domänenwissen bereitgestellt, welches zur Erstellung der Referenzarchitektur nötig ist und zusätzlich einen essentiellen Teil der Referenzarchitektur darstellt. Anforderungen an ein Informationssystem zur Unterstützung der Instandhaltung ergeben sich teilweise direkt aus den Strategien und den Arten der Instandhaltung und geben wiederum Rückschlüsse darauf, welche Funktionen später in einer Referenzarchitektur abgedeckt werden müssen.

Daneben wird durch die genaue Untersuchung der Domäne der Betrachtungsrahmen der Referenzarchitektur, der im vorherigen Kapitel umschrieben wurde, anhand konkreter Beispiele illustriert. Dazu ist es zusätzlich nötig, bisherige existierende Systeme zu untersuchen, die aktuell in der Produktion eingesetzt werden. Zum einen dient dies dazu, die in ein IIS zu integrierenden Altsysteme und damit potentiell nötige Schnittstellen aufzunehmen. Zum anderen sollen Informationssysteme vorgestellt werden, die aktuell die Instandhaltung unterstützen, um den Betrachtungsrahmen anhand feststehender Begriffe zu definieren und sich von diesen Systemen abzugrenzen. Diese Systeme zur Unterstützung der Instandhaltung umfassen etablierte und neuartige Systeme. Am Beispiel dieser wird die aktuelle Entwicklung der Instandhaltung bzw. der Instandhaltungsunterstützung aufgezeigt. Diese Entwicklungen zeigen auf, in welche Richtung sich die Instandhaltung bewegt bzw. bewegen kann und welche technologischen Potentiale eine Lösung zur Unterstützung der Instandhaltung nutzen kann. Rückschlüsse auf Anforderungen zur Erweiterbarkeit der Referenzarchitektur bzw. des Systems lassen sich hierzu gewinnen und in der Praxis überprüfen.

Weiterhin dient die Aufnahme der Systeme als Basis, um im nächsten Kapitel eine Literaturrecherche durchzuführen. Nachfolgend wird die Aufteilung des Kapitels vorgestellt, wodurch die angesprochenen Aspekte näher beleuchtet werden:

Zuerst werden in Kapitel 3.2 der Begriff der Instandhaltung und damit verbundene Unterbegriffe beleuchtet und definiert. Dieses Kapitel wurde in weiten Teilen der Veröffentlichung der Referenzarchitektur von Reidt et al. (2016a) übernommen. In Kapitel 3.3 werden unterschiedliche Instandhaltungskonzepte und damit verbundene Instandhaltungsstrategien vorgestellt, welche aktuell betrieben werden. Basierend auf diesem Wissen lassen sich die Anforderungen an die Instandhaltung und die unterstützenden Systeme ableiten. Nachdem die Basis der Instandhaltung, dazugehörige aktuelle Konzepte und Strategien beleuchtet wurden, werden in Kapitel 3.4 die Informationssysteme untersucht, welche für eine effektive Instandhaltung nötig sind. Dabei wird eine Differenzierung vorgenommen: Zuerst werden die typischen Informationssysteme der Produktion vorgestellt,

aus denen Daten für die Instandhaltung benötigt werden. Anschließend werden die Informationssysteme genannt, welche explizit die Instandhaltung unterstützen sollen. Die zu entwickelnde Referenzarchitektur muss als Grundlage die genannten Systeme integrieren oder bestimmte Aufgaben dieser übernehmen können. Die gewonnenen Erkenntnisse werden in Kapitel 3.5 zusammengefasst.

3.2 Definition und Begriffe der Instandhaltung

In dieser Dissertation wird die Definition der Instandhaltung nach der DIN 31051 (DIN 31051:2012-09, S. 5) befolgt. Diese stellt für den deutschsprachigen Raum die verbreitetste und wichtigste Definition dar. Für das weitere Verständnis ist es wichtig, die grundlegenden Begrifflichkeiten aus der DIN 31051 zu erläutern, welche in der RAII verwendet werden. Dadurch kann die Domäne der Instandhaltung aus einem einheitlichen Blickwinkel betrachtet werden. Die Instandhaltung selbst wird in der DIN 31051 als die „Kombination aller technischen und administrativen Maßnahmen sowie Maßnahmen des Managements während des Lebenszyklus einer Einheit, die dem Erhalt oder der Wiederherstellung ihres funktionsfähigen Zustands dient, sodass sie die geforderte Funktion erfüllen kann“ (DIN 31051:2012-09, S. 6), definiert.

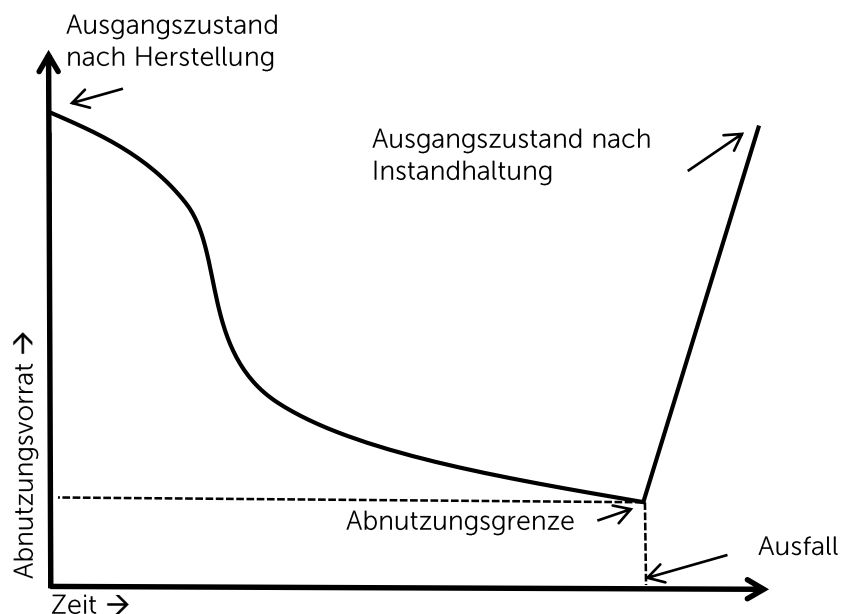


Abbildung 6: Abbau des Abnutzungsvorrates und seine Erstellung durch Maßnahmen der Instandhaltung

Quelle: In Anlehnung an DIN 31051:2012-09 (S. 8)

Die Einheit bezeichnet in der DIN 31051 ein „Teil, Bauelement, Gerät, Teilsystem, Funktionseinheit, Betriebsmittel oder System, das/die für sich allein beschrieben und betrachtet werden kann“ (DIN 31051:2012-09, S. 7). Eine Einheit besitzt einen sogenannten Abnutzungsvorrat, welcher dem „Vorrat der möglichen Funktionserfüllungen unter festgelegten Bedingungen“ (DIN 31051:2012-09, S. 8) entspricht. Ist dieser Abnutzungsvorrat aufgrund von physikalischen und/oder chemischen Vorgängen verbraucht, kommt es zu einem Ausfall, was in der „Beendigung der Fähigkeit einer Einheit [...], eine geforderte Funktion [...] zu erfüllen“ (DIN 31051:2012-09, S. 10) resultiert. Der Abbau des Abnutzungsvorrates und der Zusammenhang mit Maßnahmen der Instandhaltung kann mit einer Funktion beschrieben werden. Zur Illustration ist in Abbildung 6 ein exemplarischer Verlauf dargestellt.

In diesem Verlauf ist der Abnutzungsvorrat nach der Herstellung der Einheit sehr hoch und nimmt im Laufe der Zeit ab. Die Abnutzungsgrenze ist hierbei der vereinbarte oder festgelegte Mindestwert des Abnutzungsvorrates, bei dem eine Maßnahme der Instandhaltung durchgeführt werden muss. Diese Maßnahme erhöht den Abnutzungsvorrat bspw. durch Austausch verbrauchter Elemente.

Der erwähnte Verlauf des Abnutzungsvorrates skizziert einen typischen Anwendungsfall einer Instandhaltung. Instandhaltung ist hierbei ein notwendiger Bestandteil der betrieblichen Anlagenwirtschaft, orientiert sich typischerweise am Ist- und Sollzustand der Betrachtungseinheit, ist für die Werterhaltung und -steigerung sowie für die Nutzung des Anlagenvermögens eines Unternehmens nach VDI 2895:2012-12 (S. 3) verantwortlich.

Im Rahmen der Instandhaltung existieren verschiedene Maßnahmen, um die Ziele der Werterhaltung und -steigerung zu erreichen. Nach der DIN 31051:2012-09 (S. 4 ff.) werden diese Möglichkeiten in vier Grundmaßnahmen unterteilt, welche nachfolgend beschrieben werden:

- **Wartung** – Maßnahmen zur Verzögerung des Abbaus des vorhandenen Abnutzungsvorrats.
- **Inspektion** – Maßnahmen zur Feststellung und Beurteilung einer Einheit einschließlich der Bestimmung der Ursache der Abnutzung und dem Ableiten der notwendigen Konsequenzen für eine künftige Nutzung.
- **Instandsetzung** – physische Maßnahmen, die ausgeführt werden, um die Funktion einer fehlerhaften Einheit wiederherzustellen.
- **Verbesserung** – Kombination aller technischer Maßnahmen sowie Maßnahmen des Managements zur Steigerung der Zuverlässigkeit und/oder Instandhaltbarkeit und/oder Sicherheit einer Einheit, ohne ihre ursprüngliche Funktion zu ändern.

Zur Vereinfachung wird in dieser Arbeit unter dem Begriff des Ausfalls nur grob zwischen Störung, Fehler und Ausfall differenziert. Ein Fehler stellt eine unzulässige Abweichung des Sollwerts eines Merkmals einer Betrachtungseinheit dar, wobei prinzipiell die Funktionsfähigkeit erhalten bleiben kann. Eine Störung hingegen ist ein wahrnehmbarer Grad der Nichterfüllung der Funktion und damit eine direkte Beeinträchtigung der Funktion. Ein Ausfall ist die Nichterfüllung der Funktion. Weiter wird die Betrachtungseinheit in dieser Arbeit als Anlage, Maschine, Prozessgerät oder Teilsystem etc. angesehen. Auf Basis individueller Anforderungen wird zwar zwischen diesen Arten der Betrachtungseinheiten unterschieden, im Allgemeinen werden diese Begriffe in dieser Arbeit jedoch synonym verwendet.

Zur Durchführung dieser Maßnahmen und der Instandhaltung allgemein existieren verschiedene Instandhaltungskonzepte und -strategien, welche wiederum Informationen aus einer Vielzahl an Informationssystemen benötigen. Diese Strategien sollen helfen, die Instandhaltung nach vorher festgelegten Zielen mitsamt ihren Maßnahmen optimal durchzuführen und werden daher im nachfolgenden Kapitel behandelt.

3.3 Instandhaltungskonzepte und -strategien

In der Instandhaltungsliteratur wird neben den reinen Instandhaltungsmaßnahmen auch zwischen übergeordneten Instandhaltungsstrategien und -konzepten unterschieden. Instandhaltungskonzepte umfassen eine Menge von Instandhaltungsstrategien und -tätigkeiten sowie die Entscheidungsstruktur, nach welcher diese eingeplant und eingesetzt werden (Pintelon/Parodi-Herz, 2008, S. 27). Diese Instandhaltungskonzepte bestimmen die Geschäftsphilosophie, nach der sich die Instandhaltung ausrichtet und betrieben wird und sind nötig, um die Komplexität der Instandhaltung zu managen. Daher werden Instandhaltungskonzepte auch als Managementkonzepte bezeichnet, nach der die Instandhaltung betrieben wird (Pawellek, 2016, S. 5). Diese Konzepte besagen auch, wann welche Instandhaltungsstrategie angewendet werden sollte, um das nach diesem Konzept optimale Ergebnis zu erreichen. Diese Konzepte spielen für die Art der Durchführung der Instandhaltung eine große Rolle, da diese den Rahmen für die Ausrichtung vorgeben.

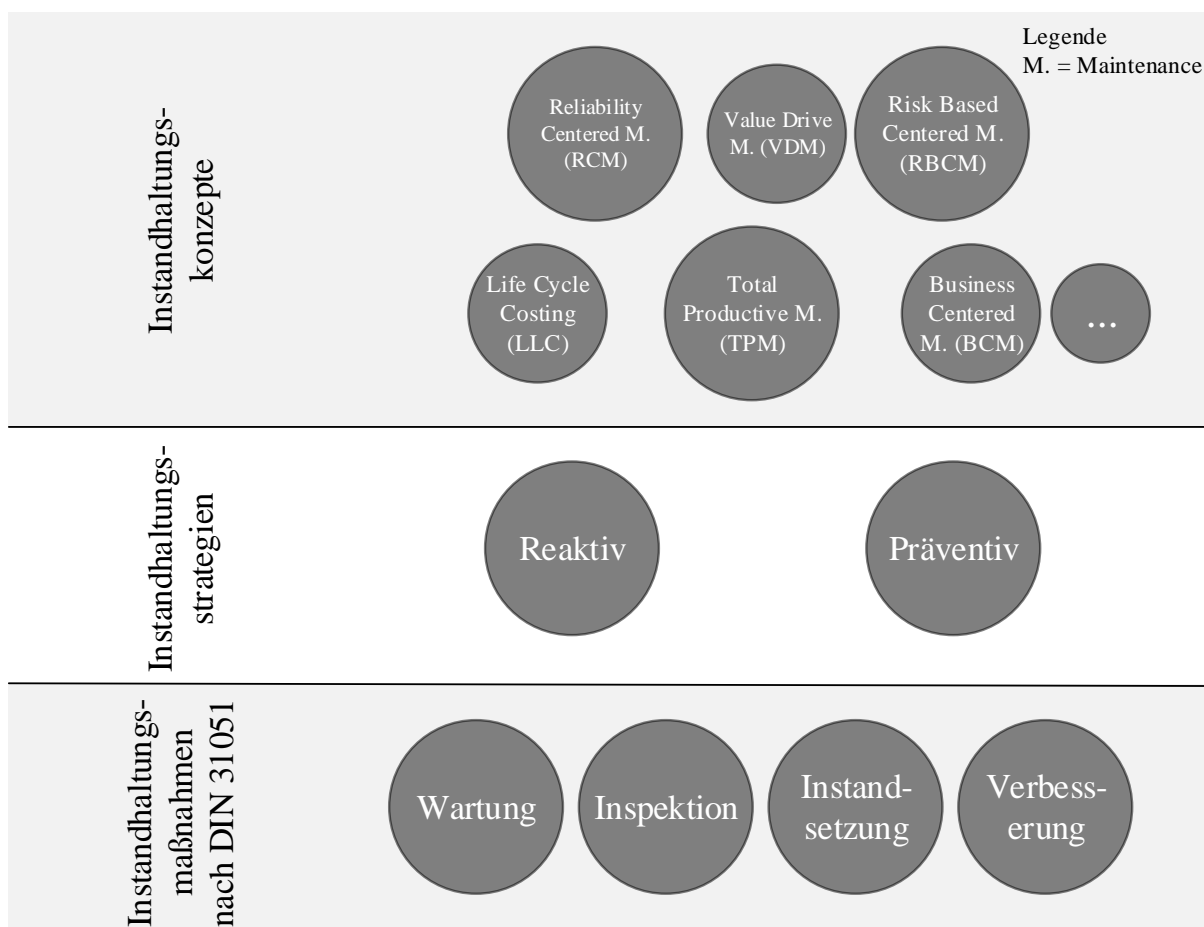


Abbildung 7: Unterscheidung zwischen Instandhaltungskonzepten, -strategien und -maßnahmen

Quelle: Eigene Darstellung

Nach Pawellek, 2016 (S. 171) existiert keine einheitliche Definition des Begriffes Instandhaltungsstrategie. In dieser Arbeit wird dem Begriffsverständnis von Matyas (2002, S. 13) gefolgt:

„Instandhaltungsstrategien sind Regeln, die angeben, zu welchen Zeitpunkten welche Aktionen an welchen Aggregaten bzw. Bauteilen vorgenommen werden sollen“.

Die Auswahl der richtigen Instandhaltungsstrategie befindet sich nach Matyas (2002, S. 13) im Spannungsfeld von Wirtschaftlichkeit, Sicherheit und Verfügbarkeit, um eine Kostenminimierung und eine Verfügbarkeitsmaximierung der Anlagen zu erreichen.

Der Zusammenhang zwischen Instandhaltungsstrategie, -konzepten und -maßnahmen wird auf der nachfolgenden Grafik erläutert. Dort sind pro Kategorie mehrere Ausprägungen dargestellt, die in nachfolgenden Unterkapiteln behandelt werden, beginnend mit den verschiedenen Ausprägungen der Instandhaltungskonzepte.

3.3.1 Instandhaltungskonzepte

3.3.1.1 Reliability Centered Maintenance

Reliability Centered Maintenance (RCM), in Deutsch auch als „zuverlässigkeitsorientierte Instandhaltung“ bezeichnet, stellt eine hochstrukturierte Methode für die Planung der Instandhaltung und der damit verbundenen Aktivitäten dar (Starr et al., 2010, S. 7).

Der Ursprung von RCM stammt aus der Luftfahrtindustrie der USA in den 1960er Jahren, bevor es vom amerikanischen Militär adaptiert wurde und ebenso bei hochriskanten Nuklearanlagen eingesetzt wird (Pintelon/Parodi-Herz, 2008, S. 37). Aktuell wird es industrieübergreifend angewandt, wenn Zuverlässigkeit und Instandhaltung bei Produkten und Anlagen eine große Rolle spielen. Bei der Verbreitung des Konzeptes sind insbesondere die Beiträge von Nowlan und Heap (1978), Anderson und Neri (1990) und Moubray (1997) entscheidend gewesen. Heutzutage existieren einige Varianten von RCM, eine der bekanntesten ist die sogenannte streamlined RCM (Pintelon/Parodi-Herz, 2008, S. 38).

Das Vorgehen von RCM berücksichtigt Instandhaltungsanforderungen für einzelne Komponenten eines untersuchten Systems in ihrem Einsatzumfeld, was auch Sicherheitsabwägungen, Instandhaltungskosten und die Kosten von Produktionsausfall inkludiert (Mikler, 2015, S. 224).

Ein maßgeblicher Bestandteil des RCM-Konzeptes sind die folgenden sieben Kernfragen des RCM, die für jede Ausrüstungskomponente und jedes System beantwortet werden sollten. Diese Fragen sind nach Moubray (1996, S. 20) wie folgt:

- „Welche Funktionen und damit verbundenen Leistungsnormen gibt es für die Anlagenkomponente unter Berücksichtigung der momentanen Betriebsbedingungen?
- Wie versagt eine Anlagenkomponente bei der Ausführung ihrer Funktionen?
- Welche Ursachen hat die Funktionsstörung?
- Was geschieht, wenn die Anlagenkomponente versagt?
- Auf welche Weise stört das Versagen?
- Was kann getan werden, um der Störung vorzubeugen?
- Was sollte unternommen werden, wenn keine annehmbare vorbeugende Lösung gefunden werden kann?“

Zur Beantwortung dieser sieben Fragen existieren klare Schritt-für-Schritt-Prozeduren und weitere Materialien. Kern des systematischen Vorgehens der RCM ist oftmals die Failure Mode and Effects Analysis (FMEA) (Ahuja/Khamba, 2008, S. 714), in Deutsch auch „Fehlermöglichkeits- und -einflussanalyse“ genannt, oder die Failure Mode and Effects and

Criticality Analysis (FMECA) (Rausand, 1998, S. 123), um die relevanten Komponenten und Fehlermodi zu identifizieren (Starr et al., 2010, S. 8). Damit hilft dieses Konzept bei der Wahl der optimalen Instandhaltungsstrategie zum Betrieb von bestimmten technischen Anlagen.

Falls RCM korrekt eingesetzt wird, reduziert sich nach (Moubray, 1997, S. 312) der Anteil von Routinearbeit von Instandhaltern um 40-70 %. Die Vorteile von RCM können gewöhnlicherweise in zwei Kategorien unterteilt werden: Risikoreduktion und Kostenreduzierung (Johnston, 2002, S. 513). Eine Kostenreduzierung entsteht meist dadurch, dass durch die Analyse Möglichkeiten aufgedeckt werden, die Instandhaltungskosten bei einem bestimmten Ausrüstungsbauteil zu verringern ohne dessen Zuverlässigkeit zu reduzieren. Risikoreduzierung entsteht durch Erhöhung der Zuverlässigkeit der instandzuhaltenden Komponenten durch Identifikation von optimaleren Instandhaltungsstrategien. Neben diesen Vorteilen hat RCM aber auch einige Nachteile. Nach Pintelon und Parodi-Herz (2008, S. 37) hat RCM aus konzeptioneller Sicht folgende Schwachstellen:

- RCM bietet keine Möglichkeiten, Aufgaben zu paketieren und automatisch einen durchführbaren Arbeitsplan von relevanten Instandhaltungsmaßnahmen abzuleiten.
- Die standardisierten Entscheidungsdiagramme in vielen RCM-Implementierungen sind hilfreich, jedoch weit entfernt davon perfekt zu sein.
- Besonders ist von akademischer Seite zu bemängeln, dass die wissenschaftliche Basis der RCM, die FMEA Analyse, oft nur auf eine ad-hoc Basis durchgeführt wird. Oft sind die statistischen Daten unzureichend oder nicht genau. Der Abnutzungsprozess der eingesetzten Komponenten (Fehlermechanismus) in der physischen Umgebung (z. B. staubige Umgebung) wird oftmals nicht verstanden. Dies führt dazu, dass die Balance zwischen wertvoller Erfahrung und objektivem statistischen Wissen nicht gewahrt werden kann.
- Im Allgemeinen ist die Einführung eines RCM ein sehr herausforderndes Projekt, welches eine ständige Beschäftigung erfordert, weswegen diese Projekte oft nicht korrekt durchgeführt werden, da diese sehr ressourcenintensiv sind.

Neben den Nachteilen, die von Pintelon und Parodi-Herz (2008) diskutiert werden, wenden auch Starr et al. (2010, S. 9) folgende Punkte gegen RCM ein:

- Ähnlich wie die statistischen Daten sind Fehlerdaten schwer zu erlangen, da Komponenten vor dem Auftreten von Fehlern üblicherweise ersetzt werden, um hohe folgende Kosten, insbesondere in der Prozess- und Chemieindustrie, zu vermeiden.
- Zuverlässigkeit ist eventuell nicht der Hauptfokus der Instandhaltung, Produktionsanlagen fokussieren hauptsächlich die Verfügbarkeit.
- Der RCM-Prozess ist oft nicht mit der Zustandsüberwachung gekoppelt, sodass von aktuellen Möglichkeiten des Condition Monitorings bzw. des Predictive Maintenance kein Gebrauch gemacht wird. Als Resultat wird die Entwicklung von potentiellen Fehlern nicht in das RCM Konzept einbezogen.

3.3.1.2 Total Productive Maintenance

Total Productive Maintenance (TPM) ist eine spezifische japanische Philosophie, die durch die Nippon Denso Company Co. Ltd, einem Zulieferer der Toyota Motor Company entwickelt wurde (Ahuja/Khamba, 2008, S. 715). Als eine Antwort auf die zunehmend wettbewerbsintensiveren Märkte, welche die Unternehmen dazu zwangen, die Produktion zu

verbessern (Rodrigues/Hatakeyama, 2006, S. 276), begann die Entwicklung des Konzeptes des TPM bei dem Zulieferer in den 1970er mit der Erkenntnis, dass es die Maschinenführer sind, die den Status Quo der Maschine und der Produktionsprozesse am besten kennen und so wissen, wie man einen Ausfall verhindert (Yamashina, 1995, S. 36). Damit verfolgt TPM eine Reintegration der Instandhaltung in den regulären Betrieb eines produzierenden Unternehmens (Zerbst, 2000, S. 2). Denn Zerbst (2000, S. 2) legt dar, dass die Instandhaltung zu Beginn der Industrialisierung einen Teil des Betriebs darstellte und nur aufgrund der zunehmenden Automation und damit verbundenen Arbeitsteilung von den restlichen Arbeitsbereichen abgetrennt wurde.

Diesem Aspekt der Dezentralität folgend soll als ein Ziel von TPM die Maximierung der Gesamtanlageneffektivität erreicht werden (Schröder, 2009, S. 149). Dazu muss die Maximierung des Outputs (Qualität, Kosten, Lieferzeit) bei minimalem Mitteleinsatz (Personal, Material, Maschine) und bei gleichzeitiger Vermeidung von Verschwendung erreicht werden (Schröder, 2009, S. 149). Als Maß für die Gesamtanlageneffektivität wird die Kennzahl Overall Equipment Effectiveness (OEE) verwendet, die als Standardkennzahl des TPM angesehen werden kann. OEE ist dabei ein Maß für die Wertschöpfung einer Anlage und kombiniert Verfügbarkeit, Leistungseffizienz und Qualität (Chand/Shirvani, 2000, S. 151).

Nach Schuh und Lorenz (2009, S. 78) ist TPM im Gegensatz zu den übrigen Ansätzen ein ganzheitliches und in sich geschlossenes Konzept und folgt – ähnlich den ganzheitlichen Produktionssystemen (z. B. Toyota-Produktionssystem) – dem Prinzip der Vermeidung jeglicher Verluste. Die Verluste werden bei TPM in sechs unterschiedliche Fehlerarten von insgesamt drei Kategorien unterschieden. Diese sind nach Al-Radhi und Heuer (1995, S. 17ff):

- Verluste durch Ausfallzeiten:
 - Dazu gehören Ausfallzeiten, die auf Anlagenstörungen und -ausfälle zurückzuführen sind sowie
 - Ausfallzeiten, welche durch Rüstzeiten verursacht werden.
- Geschwindigkeitsverluste durch
 - Fehlermeldungen von Sensoren, Werkstückblockierungen, die kurze Stopps verursachen oder Geschwindigkeitsverluste durch
 - gedrosselte Produktionsgeschwindigkeit, welche dann resultiert, wenn die geplante Geschwindigkeit nicht mit der realen Geschwindigkeit übereinstimmt.
- Fehler aus Defekten
 - Verluste, die aus Bearbeitungsfehlern im Prozess entstehen.
 - Ungünstige Anlagenanlaufphasen können zu verringertem Ertrag führen, da die Zeit zwischen Start und stabiler Produktion zu lange dauert.

Um diese Verluste zu vermeiden und schließlich die OEE zu erhöhen werden acht Konzepte im Rahmen des TPM verfolgt, die auch als die acht Säulen des TPM bezeichnet werden (Ahuja/Khamba, 2008, S. 721; Kłos/Patalas-Maliszewska, 2013, S. 18). Diese sind auf Abbildung 8 dargestellt und werden nachfolgend erläutert.

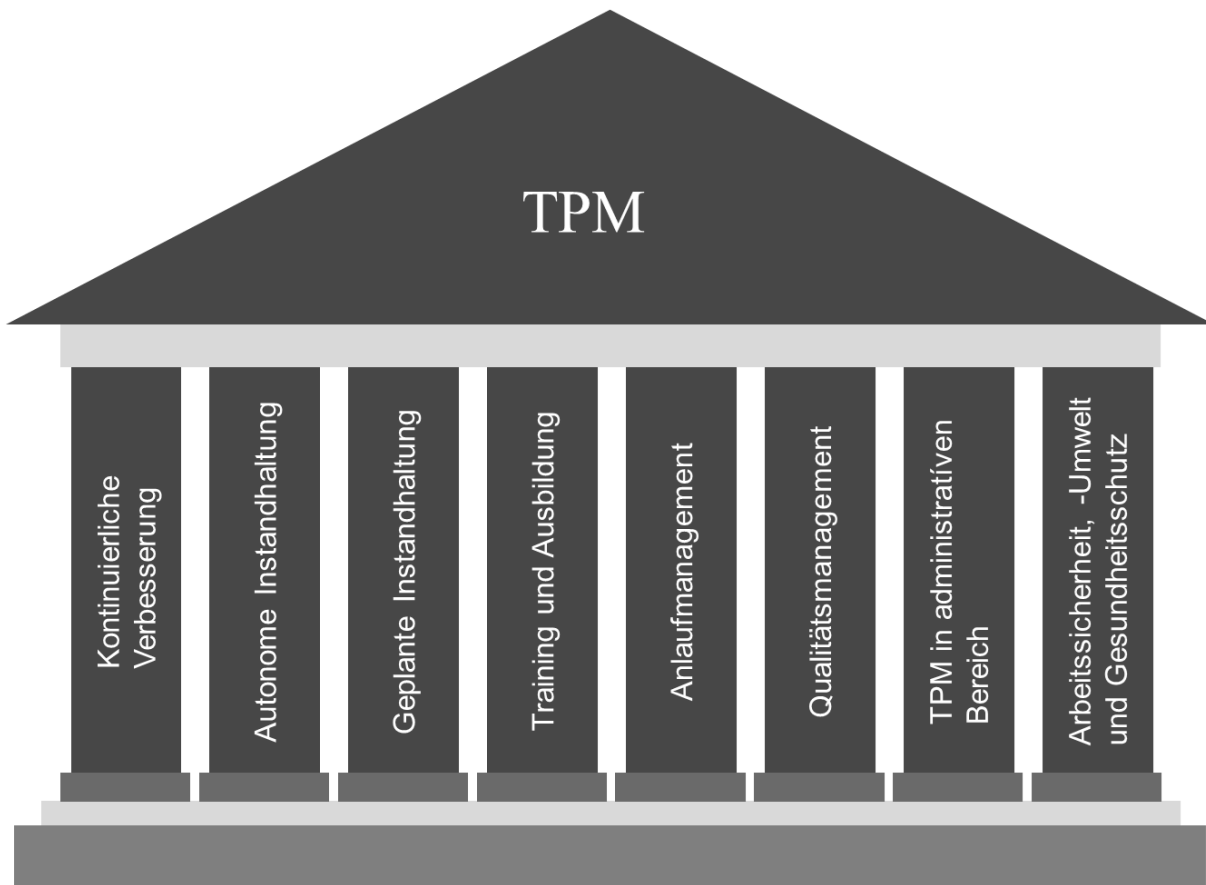


Abbildung 8: Die acht Pfeiler des TPM nach dem Vorschlag der JIPM

Quelle: In Anlehnung an Ahuja und Khamba (2008, S. 721)

- Autonome Instandhaltung:** Hintergrund dieses Pfeilers ist die Erkenntnis, dass ohne das Interesse und die Kooperation der Produktionsmitarbeiter keine ordentliche Instandhaltung etabliert werden kann (Yamashina, 1995, S. 36). Der Grundgedanke einer autonomen Instandhaltung ist daher, dass von Mitarbeitern der Produktion signifikante Instandhaltungsaufgaben durchgeführt werden müssen (Starr et al., 2010, S. 9), aber auch das Sammeln von nötigen Instandhaltungsdaten der Maschine obliegt (Lee/Lapira, 2011, S. 70f). Hierdurch wird ein schnelleres Handeln und eine Verantwortung für die „eigene“ Maschine erzeugt, sodass zukünftig vom Produktionspersonal Gegenmaßnahmen bei erkannten Fehlern getroffen werden können. Der Einführung der autonomen Instandhaltung liegen nach Tsuchiya (1992, S. 31ff) die folgenden fünf Prinzipien zugrunde: Seiri (Organisation, Aussortieren), Seiton (effiziente Ordnung), Seiso (Reinigung, Säuberung), Seiketsu (Sauberkeit, Standardisierung) und Shitsuke (Disziplin, Training).
- Kontinuierliche Verbesserung:** Dieses Grundprinzip des TPM beinhaltet die systematische Identifikation von Elimination von Verlusten durch Umsetzung einer Null-Verluste-Strategie. Dies resultiert darin, dass die TPM-Aktivitäten darauf ausgelegt sind, die sogenannten „six major losses“ zu eliminieren (Swanson, 2001, S. 239). Darüber hinaus ist das Prinzip der kontinuierlichen Verbesserung nicht nur auf Verluste und Fehler zu beziehen, sondern soll dazu führen, dass jegliche Tätigkeiten im Rahmen des TPM ständig hinterfragt und fortlaufend verbessert werden.
- Geplante Instandhaltung:** Ein weiteres Grundelement des TPM stellt das geplante Instandhaltungsprogramm dar (Schröder, 2009, S. 151). Dabei handelt es sich um

Inspektions-, Wartungs-, und Instandsetzungsmaßnahmen, die aufgrund spezieller Eigenschaften nur selten auf autonome Gruppen übertragbar sind und aus wirtschaftlichen Gründen dezentral vorgehalten werden (Al-Radhi/Heuer, 1995, S. 80ff). Grundlage sind nach Schröder (2009, S. 151) Standards zur effizienten Durchführung von Instandhaltungsprozessen wie bspw. Instandhaltungspläne, bedarfsgerechte Terminierung der Inspektion o. Ä.

- **Qualitätsmanagement:** Designfehler und die Qualitätsverbesserung von produzierten Gütern sollen durch geeignete Tools zur Qualitätsmessung erreicht werden.
- **Training und Ausbildung:** Als wesentliches Konzept zur Umsetzung von TPM gilt die kontinuierliche Schulung und das Training der Mitarbeiter, da der Mensch im Zentrum des TPM Konzeptes steht (Al-Radhi/Heuer, 1995, S. 99). Kontinuierliche Verbesserung und der Fokus auf Kleingruppentätigkeiten sowie Verantwortung sollen hiermit weiter gefördert werden, damit Wissenslücken, die der Zielerreichung des TPM entgegenstehen, geschlossen werden.
- **Arbeitssicherheit, Umwelt- und Gesundheitsschutz:** Allgemein soll die Arbeitsumgebung ein sicheres, umweltgerechtes Arbeiten ermöglichen und durch entsprechende Maßnahmen zur Sicherung Arbeitsunfälle verhindern.
- **TPM in administrativen Bereichen:** Die angeschlossenen Funktionen und Bürotätigkeiten sollen die Grundgedanken des TPMs und Konzepte übernehmen, da nur eine ganzheitliche Durchführung von TPM einen Erfolg verspricht.
- **Anlaufmanagement:** Durch das Anlaufmanagement soll eine nahezu senkrechte Anlaufkurve bei Neuanlauf von Maschinen und Anlagen gewährleistet werden. Dies soll durch eine bereichsübergreifende Planung von Produkten und Systemen sowie der Initiierung von Entwicklungsprozessen erreicht werden (Schuh/Lorenz, 2009, S. 83f).

Die Implementierung von TPM ist ein herausfordernder Prozess, da dieser in den meisten Fällen einen Wandel der Unternehmenskultur erfordert. Solch einen kulturellen Wandel in einer Fabrik herbeizuführen ist eine schwierige Aufgabe, die bei TPM das Arbeiten in kleinen Gruppen, eine starke Rolle des Maschinenführers innerhalb des Instandhaltungsprogramms und die Unterstützung der Instandhaltungsabteilung erfordert (Willmott/McCarthy, 2000). Neben den ganzen Erfolgsgeschichten von TPM existieren daher auch viele Fehlschläge, welche jedoch kaum bekannt sind (Ljungberg, 1998, S. 495). Dies resultiert aus der schieren Komplexität von TPM, das eher als Philosophie angesehen werden kann, statt nur als Konzept. TPM erwartet nach den Grundstrukturen und -elementen eine totale Partizipation auf allen Ebenen des Unternehmens (Pintelon/Parodi-Herz, 2008, S. 37).

3.3.1.3 Weitere Planungskonzepte

Neben den zwei ausführlich dargestellten Planungskonzepten TPM und RCM werden in der Instandhaltungsliteratur noch weitere Planungskonzepte behandelt. Diese sind jedoch nicht so stark verbreitet, wie die vorherigen und werden daher verkürzt dargestellt. Diese Konzepte umfassen zum einen alternative Konzepte wie das Life Cycle Costing (LCC). Bei der LCC, oder in Deutsch der "Lebenszykluskostenrechnung", ist der Gedanke, dass die gesamten Kosten eines Produktes über den Lebenszyklus hinweg betrachtet werden. Fokus der Optimierung nach der LCC sind daher nicht mehr die Instandhaltungskosten sondern die Lebenszykluskosten (Kuhn et al., 2006, S. 28). Instandhaltung soll nach dem Konzept des LCC also direkt bei dem Design des Produktes beachtet und eingeplant werden (Blanchard, 2004, S. 205). Dieses

Konzept kann auch mit den anderen Managementkonzepten verwendet werden, da es insbesondere eine Art der wirtschaftlichen Betrachtung vorgibt. Zum anderen Ansätze, die als Weiterentwicklungen von TPM und RCM angesehen werden können, wie Business Centered Maintenance (BCM), Risk Based Centered Maintenance (RBCM) oder Value Driven Maintenance (VDM), die den Aspekt des Risikos oder der Wertorientiertheit betonen. BCM wurde von Kelly (1997) als Konzept zur Erstellung eines detaillierten Instandhaltungsplan nach wirtschaftlichen Gesichtspunkten vorgestellt. Das Konzept bezieht insbesondere den administrativen Aufwand der Instandhaltung mit ein. Ausgehend von Geschäftszielen werden Instandhaltungsziele Top-down abgeleitet. Danach wird Bottom-up für jede Anlage ein Instandhaltungsplan entworfen, um anschließend diese beiden Planungen zu verbinden. RBCM hingegen kann als eine spezielle Art des RCM angesehen werden, bei der vor allem der statistische Hintergrund bei der Zuverlässigkeitsanalyse ausgeprägt ist (Pintelon/Parodi-Herz, 2008, S. 39). Die als Teil des RCM oft nur ad-hoc durchgeführte FMCA wird hier mit dem Risiko eines möglichen Fehlers ergänzt, der als Produkt des Fehlerpotentials und der Wahrscheinlichkeit des Auftretens verbunden ist (Jones, 1995). Teilweise sind die Ansätze von Jones in aktuelle Implementierungen des RCM übergegangen. VDM-Ansätze richten die Instandhaltung und ihre Maßnahmen nach dem hinzugefügten Shareholder Value aus. Hier ist insbesondere die Unterscheidung von anderen Ansätzen erkennbar, die meist eine reine Kostenbetrachtung der Instandhaltung beinhalten. Bei dem VDM-Ansatz wird die Instandhaltung hingegen hinsichtlich der potentiellen Wertsteigerung beurteilt. Ursprünglich geht das Konzept VDM auf Haarman und Delahay (2013) zurück, wobei ähnliche Planungskonzepte unter dem Namen Value Driven Maintenance Planning bei Rosqvist et al. (2009) zu finden sind.

3.3.2 Instandhaltungsstrategien

Instandhaltungsstrategien, nach denen Instandhaltung ausgeführt wird, werden nach (Niu et al., 2010) in zwei verschiedene Kategorien unterteilen, welche in Abbildung 9 dargestellt sind. Grundsätzlich kann zwischen der reaktiven und der präventiven Instandhaltung unterschieden werden. Die reaktive Instandhaltung bezeichnet die Fehlerbehebung nach dem Auftreten einer Störung, wohingegen die präventive Instandhaltung vor dem Eintreten einer Störung stattfindet, um diese zu verhindern. Die präventive Strategie lässt sich weiter in zwei Subkategorien unterteilen: Eine periodisch festgelegte und eine zustandsabhängige (engl.: Condition Based Maintenance (CBM)) Instandhaltung. Im Vergleich mit der DIN 31051 kann die präventive Instandhaltung mit der Maßnahme der Wartung und die reaktive Instandhaltung mit der Maßnahme der Instandsetzung gleichgesetzt werden. Die aktuelle zustandsorientierte Überprüfung im Kontext der präventiven Instandhaltung von Maschinen, Anlagen und Bauteilen wird allgemein als Condition Monitoring (deutsch: „Zustandsüberwachung“) bezeichnet. Darunter fällt nach DIN 13306 die „manuell oder automatisch ausgeführte Tätigkeit zur Messung der Merkmale und Parameter des Ist-Zustands einer Einheit in bestimmten Zeitabständen“ (DIN 13306:2015-09, S. 16).

Zusätzlich zu der Unterscheidung nach Niu et al. (2010) wird in dieser Arbeit noch die zustandsorientierte Instandhaltung in zwei Punkte unterteilt. Zum einen in das Herleiten von Instandhaltungsmaßnahmen in Abhängigkeit des aktuellen Zustandes, was in dieser Arbeit auch als Condition Monitoring bezeichnet wird. Condition Monitoring betrifft in dieser Arbeit nur das Überwachen des aktuellen Zustandes und nicht die Vorhersage des Zustandes. CBM hingegen umfasst Condition Monitoring und Predictive Maintenance. Zum anderen in

vorausschauende Instandhaltung, die anhand von Datenanalysen, statistischen Modellen oder Techniken des Data Mining zukünftige Fehler bzw. Zustandsprognosen berechnet und daraus Instandhaltungsmaßnahmen abgeleitet werden. Dieser Aspekt wird in dieser Arbeit auch als Predictive Maintenance bezeichnet. Der Übergang zwischen diesen beiden Aspekten ist zum Teil fließend, jedoch soll durch die Unterscheidung vor allem eine Differenzierung zwischen Technologien ermöglicht werden, da bei der Zustandsprognose bzw. dem Predictive Maintenance zunehmend hochkomplexe Data Mining Techniken eingesetzt werden, die sich doch erheblich von den traditionellen Condition Monitoring Techniken unterscheiden.

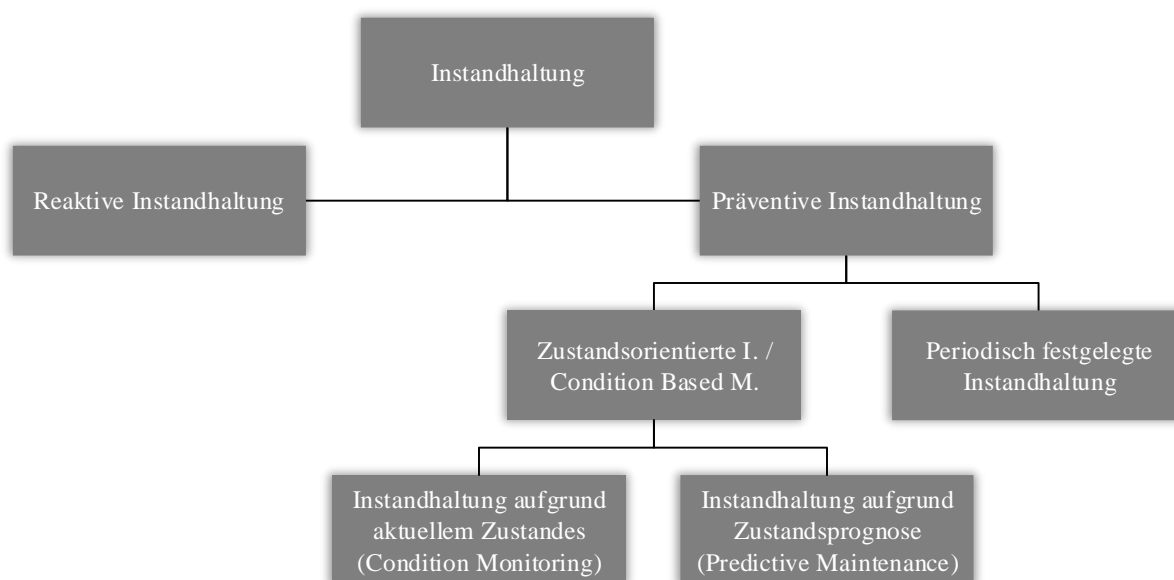


Abbildung 9: Kategorisierung von Instandhaltungsarten

Quelle: In Anlehnung an Niu et al. (2010, S. 787)

Diese Instandhaltungsarten werden in der Praxis parallel eingesetzt und besitzen jeweils eigene Anforderungen an unterstützende Informationssysteme. Die reaktive Instandhaltung benötigt Informationssysteme, die den Instandhalter bei der Fehleranalyse und -behebung unterstützen. Die präventive Instandhaltung wird hingegen hauptsächlich durch Systeme unterstützt, die einen wirtschaftlich optimalen Wartungsplan erstellen, der sich nach dem möglichst exakten Zustand der Maschinen richtet.

3.4 Verbundene Systeme für die Instandhaltung

In industriellen Anwendungsfällen werden in der Instandhaltung Informationen aus einer Vielzahl an Informationssystemen benötigt, um der Arbeit effizient nachzugehen. Sie bilden die Grundlage in vielen Unternehmen, um die Produktion und den Instandhalter mit Informationen zu versorgen und befinden sich auf verschiedenen Ebenen der Automatisierungspyramide (Bauernhansl et al., 2014). Zur Interpretation der RAI und der Identifikation eigener Informationssysteme ist das Wissen über die Vielzahl an potentiellen Systemen essentiell, um den individuellen Fall zu interpretieren und zu erkennen, welche Funktionen und Daten aus welchen Systemen zu beschaffen sind. Dies hat auf die Anwendung der RAI starken Einfluss, da diese darauf ausgelegt ist, dass bestehende Systeme eingebunden werden können. Dadurch werden Mehrfachimplementierungen verhindert und der Aspekt der Datenaggregation eines integrierten bzw. eines integrativen Instandhaltungssystems erleichtert. Darüber hinaus erleichtert die Auflistung der Systeme auch die Auflistung von Anforderungen im eigenen Unternehmen, da die bestehenden Systeme leichter untersucht werden können. Daneben kann auch die Auswahl von extern zu beschaffenden Systemen erleichtert werden, da diese in den Kontext des Instandhaltungssystems integriert bzw. in diese eingeplant werden können. Aus diesem Grund werden in diesem Kapitel die Informationssysteme der Produktion und Instandhaltung vorgestellt, welche in einem umfassenden Instandhaltungssystem integriert werden können bzw. Teile von diesem darstellen.

Die nachfolgend vorgestellten Systeme basieren auf den Arbeiten von Hänsch und Endig (2010, S. 271 ff) sowie Reidt und Krcmar (2016, S. 380) und werden um neuartige und noch nicht enthaltene Systeme ergänzt. Weiterhin werden die Systeme in drei Kategorien klassifiziert: **Klassische betriebliche Informationssysteme, deren Daten für die Instandhaltung von Relevanz sind.** Hierunter fallen typische Informationssysteme in einem produzierenden Unternehmen, die bspw. Teil der Automatisierungspyramide sind, aber auch sonstige betriebliche Informationssysteme, die in einem produzierenden Unternehmen vorhanden sind, jedoch keinen reinen Produktionsfokus besitzen. Diese werden in Kapitel 3.4.1 vorgestellt. Die nächste Kategorie, die in Kapitel 3.4.2 präsentiert wird, widmet sich den **Instandhaltungssystemen**, welche den alleinigen Fokus besitzen, die Instandhaltung zu unterstützen. Die dritte Kategorie wird in Kapitel 3.4.3 vorgestellt und beinhaltet **weitere Systembegriffe**, die in Verbindung mit den vorigen Systemen oder der Instandhaltung stehen, jedoch einen anderen Hauptfokus oder eine geringe Verbreitung besitzen. Diese Kategorie umfasst Systemnamen, die im Kontext der Instandhaltung in Literatur und Praxis zu finden sind, jedoch nicht direkt den ersten beiden Kategorien zugeordnet werden können, da sie meist nur bestimmte Techniken und Methoden bezeichnen, übergeordnete Konzepte darstellen oder Nischenerscheinungen sind oder waren. Abbildung 10 zeigt die identifizierten Systeme, die nachfolgend kurz beschrieben und deren einzelne Anwendungsfälle skizziert werden.

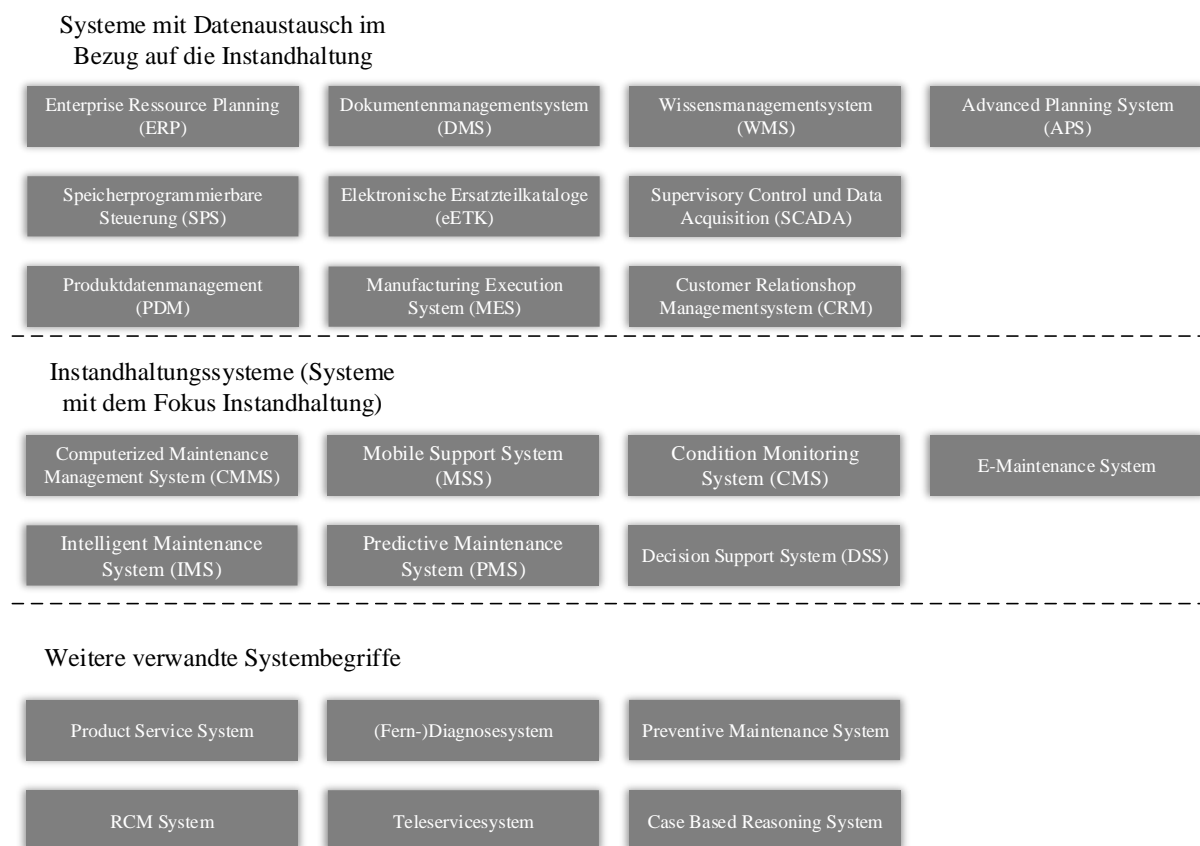


Abbildung 10: Übersicht der potentiellen Informationssysteme für Produktion und Instandhaltung

Quelle: In Anlehnung an Reidt et al. (2018b, S. 251)

3.4.1 Informationssysteme eines produzierenden Unternehmens

3.4.1.1 Produktdatenmanagementsystem

Das Produktdatenmanagementsystem implementiert das Produktdatenmanagement (engl. Product data management) (PDM) und hat damit die Aufgabe, alle bei der Produktentwicklung bzw. bei Änderungen von vorhandenen Abläufen anfallenden Daten strukturiert zu verwalten und in allen Phasen des Produktlebenszyklus verfügbar zu machen. Dabei werden produktdefinierende, -repräsentierende und -präsentierende Daten und Dokumente gespeichert und verwaltet. Synonym wird für das Product Data Management auch der Begriff Engineering Data Management (EDM) verwendet. PDM-Systeme sind im Kontext von Dokumentenmanagement (DMS), Computer-Aided Design (CAD) und ERP-Systemen aus der Problematik entstanden, die zunehmenden CAD-Dokumente parallel mit gescannten Papierdokumenten in einer dem herkömmlichen Zeichnungsarchiv entsprechenden Form zu verwalten (Eigner et al., 2014, S. 268). Diese Systeme können als spezialisierte Kombination von DMS-, CAD-Systemen und ERP-Systemen angesehen werden.

Weiterhin hat ein PDM-System nach Hänsch und Endig (2010, S. 275) folgende Ziele:

- Inhalte, Abhängigkeiten und Strukturen der Daten bereitstellen, welche ein Produkt beschreiben
- Auffinden, Weitergabe und Verwaltung von Daten

- Abbilden von Abläufen
- Integration bzw. Verbindung mit anderen Informationssystemen

Eine Weiterentwicklung von PDM-Systemen stellen Product Lifecycle Management Systeme (PLMS) dar. PLMS sind gekennzeichnet durch einen höheren Integrationsgrad in allen Phasen des Produktlebenszyklus. Dies beinhaltet eine stärkere Integration über verschiedene Disziplinen und Standorte sowie über die Prozesskette des Supply Chain Managements hinweg (Eigner et al., 2014, S. 269).

Für die Instandhaltung sind aus diesem System insbesondere 3D-Zeichnungen von Bedeutung. PLMS können darüber hinaus viele zusätzliche Daten für die Instandhaltung beinhalten, sodass abhängig von der konkreten Ausprägung z. B. Daten für Ersatzteile oder dem After-Sales vorhanden sein können.

3.4.1.2 Elektronische Ersatzteilkataloge

Elektronische Ersatzteilkataloge (eETK) sind digitalisierte oder digital erzeugte Kataloge, die Informationen, Daten und Dokumente über Ersatzteile in sich vereinen. Dazu gehören bildliche Darstellungen, 3D-Modelle wie auch Stücklisteninformationen. Damit ist es möglich, benötigte Ersatzteile zu identifizieren und diese zu bestellen.

Nach VDI 4500-3 (VDI 4500-3:2006-6) sind folgende Funktionalitäten durch einen eETK gegeben:

- Grafische Darstellungen von Ersatzteilen – Grafische (3D/2D) Darstellung der Ersatzteile samt korrekter Verbindung zu Stücklisten.
- Identifikation und Klassifikation der Stücklisteninhalte – Herstellerinformationen für die einzelnen Ersatzteile werden durch das System vorgehalten sowie interne Informationen bzgl. des Ersatzteils. Zusätzlich können Preisinformationen direkt abgerufen werden.
- Bestellfunktion – Direkte Bestellung von Ersatzteilen beim Hersteller und Händler sind durch das System möglich.
- Navigationsstrukturen – Der Katalog ist ähnlich wie das Produkt strukturiert und kann entsprechend durchsucht werden.
- Suchfunktionalitäten – Suchfunktionalitäten über das reine Suchen per Identifikationsnummer werden angeboten.
- Mehrsprachigkeit – verschiedene Sprachen werden bei Bedarf unterstützt.
- Publishing – der Ersatzteilkatalog kann in mehreren Formen publiziert werden.

Existiert dieses System bei Unternehmen, werden hierdurch auch die für die Instandhaltung nötigen Ersatzteile und dazugehörigen Informationen bereitgestellt. Weiterhin ist eine Verbindung zur Instandhaltungsmaßnahme und den dazu benötigten und im Anschluss verbrauchten Ersatzteilen möglich.

3.4.1.3 Wissensmanagementsystem

Ein Wissensmanagementsystem (WMS) (engl. Knowledge Management System (KMS)) ist ein Informationssystem im Sinne einer IKT-Plattform oder -Applikation, das Funktionen

bereitstellt und/oder kombiniert, die dazu dienen, explizites sowie implizites Wissen bzw. Informationen einer Organisation oder Teile davon kontextabhängig zur Verfügung zu stellen (Maier, 2007, S. 86). Diese Systeme bündeln daher das angesammelte Wissen in Unternehmen. Dies wird erreicht, indem Informationen gesammelt, geeignet modelliert und anschließend gespeichert werden. Im Kontext der Instandhaltung könnten diese Systeme Hilfestellung für die Arbeitsschritte geben. Jedoch ist eine adäquate, aufbereitete Darstellung der Informationen für den Instandhalter oft nicht gegeben. Es existieren jedoch bei WMS einige spezielle Implementierungen, die sich alleine der Instandhaltung widmen oder Techniken von WMS in Instandhaltungssysteme übertragen.

Die Art der Systeme, die unter den Begriff WMS fallen, sind sehr vielfältig: Es lassen sich Wikisysteme, Tasklisten, Notizsysteme, Suchsysteme, strukturierte Wissensdatenbanken und Groupwaresysteme unter diesen Begriff subsumieren.

3.4.1.4 Dokumentenmanagementsystem

Dokumentenmanagementsysteme (DMS) dienen der Verwaltung von in physischer bzw. digitaler Form auftretenden heterogenen Dokumenten innerhalb von Unternehmen. Grundsätzlich sollten DMS nach Götzer et al. (2004, S. 10) alle wichtigen Belangen unterstützen, die für Unternehmen hinsichtlich

- der Erzeugung,
- dem Empfang,
- der Ablage,
- dem Verwalten,
- dem Wiederauffinden und
- der Weiterverarbeitung eines Dokumentes von Relevanz sind.

Um dies zu erreichen, stellt ein DMS nach Hänsch und Endig (2010, S. 273) folgende Funktionen bereit:

- Dokumentimport – Import von einerseits digitalen, sich noch nicht im System befindlichen, wie auch physischen Dokumenten. Diese werden durch z. B. OCR-Programme digitalisiert und darauffolgend importiert und indiziert.
- Indizierung – Indizierung der jeweiligen Dokumente mithilfe von Stichworten oder Inhalten des Textes.
- Verwaltung – Hierzu zählen die Möglichkeiten Dokumente und ihre Metadaten sicher zu hinterlegen, Freigabe- und Änderungsmanagement anzubieten und in Verbund dazu verschiedene Versionen eines Dokumentes mitsamt einer Versionsverwaltung bereitzustellen.
- Archivierung – Darunter ist zum einen die sichere Langzeitverwahrung von Dokumenten bzw. Metadaten zu verstehen, zum anderen die Verwaltung des Archivs und eventueller Sicherheitsmechanismen.
- Recherche – Zum möglichst schnellen Auffinden von Dokumenten werden vielfältige Suchmechanismen durch das DMS zur Verfügung gestellt.
- Reproduktion – Die Möglichkeit im System vorhandene Dokumente anzuzeigen, sie wieder physisch greifbar zu machen bzw. sie zu vervielfältigen.

- Administration – Dies bezieht sich auf die Verwaltung des DMS selbst. Hierunter fallen Funktionen zur Festlegung von Zugriffsberechtigungen, Benutzereinstellungen, Auswertungen und das Sichern und Wiederherstellen von Dokumenten.

DMS enthalten oft wichtige Daten für die Instandhaltung, z. B. Informationen über Wartungen in Form von Wartungsberichten, Anleitungen für Maschinen oder Bauteile. Während der Instandhaltungstätigkeit selbst werden Anleitungen von Maschinen bzw. Wartungspläne benötigt, sofern diese im DMS gehalten und gepflegt werden.

3.4.1.5 Enterprise Resource Planning

Ein Enterprise Resource Planning (ERP)-System unterstützt sämtliche in einem Unternehmen ablaufende Geschäftsprozesse. Es enthält meist Module für die Bereiche Beschaffung, Produktion, Vertrieb, Anlagenwirtschaft, Personalwesen, Finanz- und Rechnungswesen usw., die über eine gemeinsame Datenbasis miteinander verbunden sind. Zusätzlich können mithilfe von Instandhaltungsmodulen Funktionen direkt für die Instandhaltung bereitgestellt werden. Integrierte Bestandteile eines ERP-Systems werden auch als Produktionsplanungs- und Steuerungssystem (PPS) bezeichnet bzw. auch separat vertrieben. In der vorliegenden Auflistung werden diese als integriert betrachtet.

Die Instandhaltung wird bei den ERP-Systemen meist als Bestandteil betrieblicher Geschäftsprozesse angesehen und als Weiterführung der geschäftlichen Wertschöpfung verstanden. Instandhaltungsmodule verfügen dabei über den Funktionsumfang traditioneller IPS-Systeme. Es existieren jedoch Verbindungen zu den Funktionalitäten des ERP-Systems, wie der Materialwirtschaft, dem Qualitätsmanagement, der Finanz-/Debitorenbuchhaltung, der Personalwirtschaft etc.

3.4.1.6 Manufacturing Execution System

Als Manufacturing Execution System (MES) wird ein am technischen Produktionsprozess operierendes Informationssystem bezeichnet. Es zeichnet sich gegenüber ähnlich wirksamen Systemen zur Produktionsplanung (ERP-Systeme) durch die direkte Anbindung an die verteilten Systeme des Prozessleitsystems aus und ermöglicht die Führung, Lenkung, Steuerung und Kontrolle der Produktion in Echtzeit.

Die Funktionalitäten hierbei sind nach VDI 5600-3 (VDI 5600-3:2013-7) u. a.:

- Feinplanung und -steuerung.
- Unterstützt die Erledigung des Arbeitsvorrats unter Beachtung vorhandener Einschränkungen und Zielvorgaben.
- Betriebsmittelmanagement.
- Sicherstellung der termin- und bedarfsgerechten Verfügbarkeit und technische Funktionsfähigkeit der Betriebsmittel.
- Materialmanagement.
- Termin- und bedarfsgerechte Versorgung der Fertigung mit Material und Führung von Umlaufbeständen.
- Personalmanagement.

- Termingerechte Zurverfügungstellung von Personal mit ausreichender Qualifikation für den Produktionsprozess.
- Qualitätsmanagement.
- Unterstützt die Sicherstellung der Produkt- und Prozessqualität.
- Informationsmanagement.
- Datenerfassung.
- Manuelle, halb- oder vollautomatische Erfassung von Daten aus dem Prozess inklusive Eingangs- und Vorverarbeitung der Informationen und Bereitstellung von Statusinformationen.
- Leistungsanalyse.
- Ermittlung und Bereitstellung von Performance-Indikatoren und Realisierung von kurzen und langen Regelkreisen zum Erkennen von Soll-Ist-Abweichungen und Optimierung der Produktion.

Viele der im MES enthaltenen Informationen sind für die Instandhaltung oder auch die Instandhaltungsplanung von entscheidender Bedeutung. Performance-Indikatoren sind u. U. auch für Instandhaltungssysteme essentiell, ebenso Daten für die Leistungsanalyse oder das Betriebsmittelmanagement können bei der Instandhaltung von Interesse sein.

3.4.1.7 Speicherprogrammierbare Steuerung

Eine Speicherprogrammierbare Steuerung (SPS) (engl. Programmable Logic Controller (PLC)) wird definiert als „eine rechnergestützte programmierte Steuerung, deren logischer Ablauf über eine Programmierereinrichtung, zum Beispiel ein Bedienfeld, einen Hilfsrechner oder ein tragbares Terminal, veränderbar ist“ (DIN IEC 60050-351:2014-09).

Der Aufbau ist zweigeteilt: Es wird ein Hardwaresystem benötigt, welches u. a. die Verbindung zum Prozess und den weiteren Informationssystemen der Automatisierung realisiert. Als zweites Bestandteil ist die Software zu nennen, die den Zugriff auf die Hardware umsetzt (Heinrich et al., 2015, S. 304). In der SPS werden die Überwachungs-, Regelungs- oder auch Steuerungsaufgaben realisiert (Heinrich et al., 2015, S. 28).

Für die Instandhaltung ist die Überprüfung der Programme ein wichtiger Bestandteil der Arbeit, da daraus Fehlerursachen hergeleitet werden können. Weiterhin sind die Daten, die zum Teil in einer SPS enthalten sind, auch für die Zustandsüberwachung relevant.

3.4.1.8 Supervisory Control and Data Acquisition (SCADA)

Computersysteme für das Sammeln und Analysieren von Echtzeitdaten werden als Supervisory Control and Data Acquisition (SCADA)-Systeme bezeichnet. Der Fokus bei diesen reinen Softwaresystemen liegt auf der Überwachung von verbundenen SPS (Daneels/Salter, 1999, S. 339). SCADA-Systeme sind als Netzleitsysteme für die Überwachung, Steuerung und Optimierung von Industrieanlagen im Einsatz. Sie werden u. a. in der Wasseraufbereitung, in energieerzeugenden und verteilenden Anlagen, den Smart Grids, in Chemieanlagen, in Anlagen für die Stahlerzeugung oder für die Pkw-Produktion eingesetzt.

SCADA-Systeme sind durch ihre Features im Bereich der Überwachung und Kontrolle schon zum Teil wichtige Systeme der Instandhaltung, da sie hierdurch Remote Maintenance ermöglichen.

3.4.1.9 Customer Relationship Management System

Der Begriff des Customer Relationship Managements (CRM) kam nach Payne und Frow (2013, S. 167) in der IT Verkaufs- und Praxis Community Mitte der 1990er Jahren auf und beschreibt einen strategischen Ansatz, der durch die Entwicklung von angemessenen Beziehungen mit Schlüsselkunden und Kundensegmenten den Shareholder Value erhöhen soll. Dieser Ansatz wird durch die Nutzung von Systemen zum Customer Relationship Management unterstützt (Chen/Popovich, 2003, S. 672). Dabei wird durch solche Systeme das Potential von Beziehungsmanagementstrategien und IT kombiniert, um profitable, langfristige Beziehungen mit Kunden und besonders wichtigen Stakeholder zu erreichen und zu managen. Die Aufgabe eines Customer Relationship Managementsystems (CRMS) ist nach Schenk et al. (2014, S. 750) die Kundenbeziehungsverwaltung, deren Dokumentation und die direkte Ausrichtung der Geschäftsprozesse auf den Kunden. Dazu können durch das CRMS bspw. Kundendaten analysiert, gefiltert und visualisiert werden, direkte Kommunikationsschnittstellen zum Kunden bereitgestellt oder eine Übersicht über alle Dialoge und Transaktionen mit dem Kunden geführt werden.

Insbesondere wenn die Instandhaltung bei (externen) Kunden ausgeführt wird, sind die damit verbundenen Daten im CRMS zu finden und von elementarer Bedeutung für die Instandhaltung. Auftragsmanagement, Service- und Kontrolltermine, Ansprechpartner und weitere Informationen sind meist im CRMS vorhanden oder müssen mit dem CRMS synchronisiert werden. Ein exemplarisches Beispiel, welche Rolle ein CRM-System bzw. die Kundendienstverwaltung für die Instandhaltung spielt, wird bei Nüttgens et al. (2014) ausführlich präsentiert.

3.4.1.10 Advanced Planning Systeme

Zur Erfüllung von Kundenaufträgen müssen heutzutage Materialverfügbarkeit, Personal, Maschinen und Werkzeuge über mehrere Produktionsanlagen, Länder und verschiedene Organisationen hinweg geplant werden. Die Planung von solchen Aspekten und deren Unterteilung in eine kurz-, mittel- und langfristige Planung wird als Advanced Planning bezeichnet. Softwaresysteme, die solch eine Planung durchführen, werden nach Stadtler (2005, S. 18) als Advanced Planning Systeme (APS) bezeichnet. APS können in einem Unternehmen u. a. für die Mitarbeiter-, Produktions- und Absatzplanung eingesetzt werden. Diese Daten sind für die Planung der Instandhaltung essentiell, da Instandhaltung möglichst in produktionsfreien Zeiten eingeplant werden sollte und die Mitarbeiterplanung der Instandhaltung von dieser abhängen kann. Weiterhin können zwingende Instandhaltungsmaßnahmen auch Auswirkung auf die Planung in der APS haben. Daher ist für eine effiziente Planung der Instandhaltung und der Produktion eine Verbindung beider Planungen vonnöten. Eine Übersicht zu typischen Softwaremodulen und Planungsaufgaben eines APS ist bei Meyr et al. (2005, S. 109ff) vorhanden.

3.4.2 Informationssysteme zur expliziten Unterstützung der Instandhaltung

Neben den typischen Systemen der Automatisierung und der IT-Unterstützung der Produktion existieren viele Arten von Informationssystemen, deren Hauptfokus die Unterstützung, Planung und Steuerung der Instandhaltung ist. In der Literatur und Praxis gibt es jedoch keine eindeutige Bezeichnung für diese Arten von Systemen. Vielmehr existieren je nach Alter und Hauptanwendungsfokus eigene Bezeichnungen. Aus diesem Grund werden nachfolgend verschiedene Systeme und deren Bezeichnungen vorgestellt, die in Literatur und Praxis häufig genannt werden.

3.4.2.1 Computerized Maintenance Management Systeme

Computerized Maintenance Management Systeme (CMMS), auf Deutsch "Instandhaltungsplanungs- und -steuerungssysteme (IPS)", gehören zu den ersten Informationssystemen, die den Fokus auf die Instandhaltung legten (Mouzoune/Taibi, 2013, S. 2). Erste wissenschaftliche Arbeiten hierzu sind vor allem Ende der 1980er und 1990er Jahre zu finden (Bohoris et al., 1995; Hora, 1987). Diese Systeme dienen oft zur Unterstützung von Instandhaltungskonzepten wie TPM (Hora, 1987) und RCM (Gabbar et al., 2003). CMMS sind daher im Allgemeinen Informationssysteme, welche die Instandhaltung bei Planung, Steuerung und Überwachung von Instandhaltungsmaßnahmen sowie der Arbeitsplanerstellung, Termin- und Kapazitätsplanung und Schwachstellenanalyse unterstützen (Reidt et al., 2016a, S. 14). Ziele sind die Erhöhung der Material- und Anlagenverfügbarkeit sowie der Effizienz der Instandhaltung als auch die Reduzierung der Materiallagerbestände und Instandhaltungskosten.

Die Funktionalitäten hierbei sind nach Hänsch und Endig (2010, S. 275) u. a:

- Objektverwaltung – Umfasst Informationen und Verwaltung von Maschinen und Anlagen, welche Instandhaltungsbedarf aufweisen. Dazu gehören Informationen über den Ort und die Historie von Anlagen und Maschinen.
- Auftragsplanung – Festlegung der nötigen Instandhaltungsaufträge und ihrer Art (Zustands-, Störungs-, Intervallbasierend).
- Auftragssteuerung – Übersicht über aktuell laufende Instandhaltungsmaßnahmen wie auch Möglichkeiten zum direkten Eingreifen in diese.
- Kostencontrolling – Umfasst Funktionen zur Kostenplanung, -steuerung und -kontrolle.
- Ersatzteil- und Bestellwesen – Ersatzteilbeschaffung und Überwachung.
- Auswertungen, Analysen und Berichte – Trends und Kennzahlen können anhand von Auswertungen, Analysen und Berichten erstellt werden.

3.4.2.2 Mobile Support Systems

Mobile Support Systems (MSS), auf Deutsch "mobile Assistenzsysteme", unterstützen den Instandhalter bei seiner Tätigkeit durch ein mobiles Endgerät, meist in Form einer App. Beispielsweise werden Abrechnungsdaten oder Herstellerunterlagen bereitgestellt und optisch ansprechend dargestellt. Beispiele sind bei Fellmann et al. (2013) im Rahmen des Projektes EMOTEC entwickelt worden oder auch in einer einfachen Form von Campos et al. (2009). MSS sind hauptsächlich als Datenaggregatoren im Einsatz und konzentrieren sich auf eine einfache, intuitive Darstellung und damit verbundene Bedienbarkeit. Diese Systeme sind noch

relativ neu und werden vor allem durch die Entwicklung im Bereich der mobilen Endgeräte vorangetrieben. Es entstehen immer kleinere und leistungsfähigere Endgeräte wie Smartphones, Tablets oder Smartglasses, welche die Nutzungsszenarien von Informationssystemen deutlich erweitern und verändern können. So ist es möglich, durch diese Systeme den Instandhalter direkt bei seiner Arbeit mit kontextsensitiven Informationen zu versorgen und sogar die eigene Sichtweise durch Techniken der AR (Aschenbrenner et al., 2016) anzureichern.

3.4.2.3 Condition Monitoring System

Condition Monitoring Systeme (CMS) haben die Aufgabe, einzelne Maschinen bzw. ganze Anlagen kontinuierlich zu überwachen. Wenn während des Betriebs einer Anlage Ist-Werte gemessen werden, die eine zu große Abweichung vom Sollzustand darstellen, erzeugt das CMS eine entsprechende Meldung. Darüber hinaus stellt das CMS eine stetig aktualisierte Datenbasis über den Zustand der Anlage dar. Diese Informationen helfen Fehler und Störungen frühzeitig zu erkennen und dadurch die Instandhaltungskosten und -dauer zu reduzieren.

Um dieses Aufgaben zu bewältigen, bieten CMS nach Hänsch und Endig (2010, S. 272) die folgenden Funktionalitäten an:

- Messwerverfassung,
- Messwertspeicherung und -verdichtung,
- Online-Überwachung und -Verknüpfung von Messwerten,
- Auswertung der Messdaten,
- Erfassung und Verarbeitung von Zusatzinformationen.

Nach Voigt et al. (2013, S. 10) kann ein CMS typischerweise in die Bereiche Software, Hardware und Remote Plattform unterteilt werden. Die Software umfasst die vorgestellten Funktionalitäten, die Hardware liefert hierzu die nötige Möglichkeit zur Datenaufnahme und -auswertung. Je nach Automatisierungsgrad werden die einzelnen Schritte softwaregestützt oder vollautomatisch durchgeführt. Die Remote Plattform ermöglicht den Zugriff aus der Ferne (aus internen oder auch ggfs. aus externen Netzwerken).

In der deutschsprachigen Literatur wird teilweise von CMS und (Fehler)-Diagnosesystemen gesprochen und eine Unterteilung vorgenommen (Nüttgens et al., 2014, S. 7), um die Komplexität der Aufgabe des Interpretierens der Condition Monitoring Daten hervorzuheben. Susto et al. (2012, S. 175) nennen solche Systeme Fault Detection and Classification Systeme. Diese haben ebenso ihren Schwerpunkt darin, den aktuellen Zustand und Fehler von Komponenten und Anlagen zu bewerten. Nach der hier verwendeten Definition von Hänsch und Endig (2010, S. 272), wonach die Auswertung der Daten auch Bestandteil des Systems ist wird es hier als Teil eines CMS angesehen. Diese vergleichbaren Systeme werden bei Bengtsson (2004, S. 1) als Condition Based Maintenance Systeme bezeichnet. Bengtsson (2004, S. 19) definiert diese Art von System als System, das die präventive Instandhaltung auf Basis von Leistungs- und/oder Parameterüberwachung und daraus folgende Aktionen nutzt, um Aktionen der vorausschauenden Instandhaltung automatisch oder im Austausch mit anderen Systemen und Menschen zu bestimmen und einzuplanen. Hierbei erweitert bzw. kombiniert er traditionelle CMS mit Fähigkeiten der Zustandsvorhersage und der Instandhaltungsplanung. Im

Kontext der Arbeit würden Teile des Systems von Bengtsson (2004), die die Zustandsprognose betreffen, den Predictive Maintenance Systemen (PMS) zugeordnet werden.

3.4.2.4 E-Maintenance System

E-Maintenance ist ein seit den frühen 2000ern auftretender Begriff (Iung et al., 2009, S. 220), der mit den Weiterentwicklungen der IKT im Allgemeinen und des Internets im Speziellen entstanden ist (Li et al., 2005, S. 132). E-Maintenance wird in der wissenschaftlichen Literatur als vier unterschiedliche Konzepte begriffen: Als Instandhaltungsstrategie, Instandhaltungsplan, Instandhaltungstyp und Instandhaltungsunterstützung (Muller et al., 2008a, S. 1166). Dementsprechend vielfältig sind die Definitionen zu diesem Begriff. Nach Levrat et al. (2008, S. 409) werden daher nur zwei Faktoren hervorgehoben, die das Entstehen von E-Maintenance auslösen und bei fast allen Definitionen angesprochen werden:

- Zum einen das Auftreten von Technologien, welche die Effizienz, Geschwindigkeit und generell die Optimierung der Instandhaltung ermöglichen.
- Zum anderen der Bedarf, dass die Instandhaltung andere Dienste des Unternehmens bzw. in diese integriert wird und mit diesen kollaboriert.

Ein E-Maintenance System (E-MS), auch oft Plattform genannt, wird von einigen Autoren als Software System, als ein Hardware/Software System oder gar als ein komplettes System bezeichnet, das physische Prozesse und bestehende Unternehmenssysteme integriert (Iung et al., 2009, S. 223). Aufgrund dieser Vielfältigkeit werden Systeme für die Intelligente Prognose oder CBM Systeme (Holmberg, 2005), Fernwartungssysteme (Iung, 2003), PMS (Koç et al., 2005) sowie auch um Webfähigkeit erweiterte CMMS Systeme (Hausladen/Bechheim, 2004) (teilweise e-CMMS genannt) als E-MS bezeichnet. Eine klarere Abgrenzung fällt bei der Vielfalt der Verwendung des Begriffes schwierig. Oftmals werden aber bestehende CMMS Systeme um neuartige Technologien erweitert, damit beantwortet werden kann, welche Anlagen, Bauteile etc. einer Instandhaltung bedürfen, wann diese geschehen soll und wie (Anleitung) und womit (vorhandene Ersatzteile) vorgegangen werden kann (Holmberg et al., 2010, S. 2). Levrat et al. (2008, S. 423) nennen nur den Begriff E-Maintenance Plattform, da jedoch kaum erkennbare Unterschiede zu E-MS zu erkennen sind, werden diese in dieser Arbeit einheitlich als Systeme bezeichnet.

Eine Kernherausforderung von E-MS und des E-Maintenance im Allgemeinen ist bei den meisten genannten Systeme nicht das eigentliche System, sondern verschiedenartige auf Instandhaltung bezogene Systeme in ein einzelnes Informationssystem zu integrieren (Mouzoune/Taibi, 2013, S. 2). Integrationsansätze zur Verbindung von Low-Level Daten wie Sensorwerte und High Level Daten, die vom Menschen erhoben wurden und diese automatisiert Fehlern zuordnen werden z. B. von Emmanouilidis et al. (2016) im Rahmen von Diagnosemodulen von E-MS vorgestellt. Diese stellen insbesondere Webtechnologien und semantische Integration in den Vordergrund bei E-MS.

3.4.2.5 Intelligent Maintenance System

Ungefähr zeitgleich mit dem Begriff des E-Maintenance kam der Begriff der Intelligent Maintenance Systeme (IMS) auf. Fokus sind hier insbesondere Arbeiten um Jay Lee (vgl. dazu

Lee et al., 2006), die insbesondere auch das Center of Intelligent Maintenance Systems betreiben (Webseite des Center for Intelligent Maintenance Systems).

Wang et al. (2006) beschreiben, dass das Kernparadigma von IMS darin besteht, dass anhand von Analysen und Vorhersagen von Leistungsminderung von Komponenten aufgrund von Sensordaten eine Fehlervorhersage ermöglicht wird. Daraus folgend ist bei dem Begriff unschwer eine starke Verbindung zur Predictive Maintenance und dem CMS zu erkennen. Darüber hinaus ist durch die hohe Integration mit dem Internet und neuartigen IKT-Technologien eine starke Ähnlichkeit mit E-Maintenance gegeben. Der Fokus der meisten IMS und der dazugehörigen Publikationen ist jedoch die Fehlerdiagnose bzw. -vorhersage (Lee et al., 2006; Zuccolotto et al., 2015). Dabei bauen sie meist CMS aus oder erweitern diese um Vorhersagealgorithmen und Internetfähigkeit bzw. Fernwartung. Weitere Arbeiten im Rahmen von IMS behandeln bspw. die Priorisierung von Fehlermeldungen (Moore/Starr, 2006) und Probleme des Scheduling von Instandhaltungsaufträgen (Wang et al., 2006).

Derselbe Autorenkreis verwendet die Bezeichnung Prognostics and Health Management System (PHMS) (Crabb, 2014; Lee et al., 2011; Monnin et al., 2011) für eine Kombination aus RCM, CBM sowie Preventive Maintenance. Innerhalb dieser Arbeit werden aufgrund der großen Überschneidungen die Begriffe PHMS und IMS synonym benutzt.

3.4.2.6 Predictive Maintenance System

PMS haben den Fokus ähnlich wie IMS noch stärker auf der Vorhersage von Fehlern bzw. der Vermeidung von Fehlern anhand von Algorithmen. Daher konzentrieren sie sich vor allem auf die konkrete Anwendung von Algorithmen und Data Mining Techniken. Nach Susto et al. (2013) können PMS statistisch den Zustand einer Anlage oder Komponente bewerten, wodurch Fehler vorab erkannt werden. Dies geschieht vor allem aufgrund historischer Daten, einem ad hoc definierten Zustandsfaktor oder allgemein statistischen Methoden. Beispiele für solche Systeme finden sich bei Abdennadher et al. (2010) und Bastos et al. (2012).

PMS sind teilweise sehr ähnlich zu CMS und IMS. Sie benötigen den aktuellen Zustand und historische Zustände einer Maschine oder Komponente, um Vorhersagen abzuleiten. CMS bieten diese Daten als Kern ihrer Funktionalität an, darüber hinaus werden mittlerweile auch bei den meisten CMS Prognosen über den zukünftigen Zustand von Maschinen und Komponenten erstellt. Der entscheidende Unterschied ist der ursprüngliche Kern der jeweiligen Systeme, der bei den CMS die Zustandsbewertung und bei den PMS die Vorhersage ist.

3.4.2.7 Decision Support System (for Maintenance)

Ein weiterer oft genutzter Term für Systeme zur Unterstützung der Instandhaltung ist der des Decision Support Systems (DSS) oder auch deutsch "Entscheidungsunterstützungssystem". Diese Systeme betonen den Aspekt der Datenanalyse und der Darstellung, um anhand moderner Algorithmen und den Entscheidungsträgern die Analyse der aktuellen Situation durch besonders zielgerichtete Informationen zu erleichtern. Dabei empfiehlt sich der Einsatz numerischer Modelle und quantitativer Methoden, die immer detailliertere Aussagen zulassen. Die Gesamtheit dieser computergestützten Methoden wird unter dem Begriff des DSS zusammengefasst (Mödinger, 2006, S. 21).

DSS erleichtern daher Management Entscheidungen zu treffen, indem sie Daten kombinieren, anspruchsvolle analytische Modelle und Werkzeuge bereitstellen und dies alles in einer möglichst nutzerfreundlichen Darstellung auf einem mächtigen System bereithalten. Mit diesem System wird das Treffen von semi-strukturierten oder unstrukturierten Entscheidungen unterstützt (Dey, 2004, S. 48). Zu finden sind solche Systeme bei Rasoyska et al. (2005) und teils bei Campos et al. (2009) und McArthur et al. (2004), jedoch meist in Kombination mit anderen Systemen zur Unterstützung der Instandhaltung. Huang, Gardoni, & Coulibaly (2009) entwerfen ein DSS für personalisierte Empfehlungen für Fehlerbehebungen und proaktive Wartung in Form von elektronischen Anleitungen für komplexe mechanische Produkte. Sie filtern dazu aktuelle Nutzer und Umgebungsvariablen, um Bedingungen für die Nutzung aufzunehmen und weisen bestimmten Nutzergruppen personalisierte Anleitungen zu. Es existieren auch DSS, die bspw. den Fokus besitzen Pipeline Fehler frühzeitig zu erkennen und Inspektionsstrategien abzuleiten (Dey, 2004). Damit ähneln sie teilweise CMS, jedoch ist der Fokus bei dem System die Ableitung der korrekten Inspektionsstrategie.

Der Term DSS beinhaltet eine enorme Definitionsweite und lässt sich auf fast alle bisherigen Systeme anwenden, da alle dazu genutzt werden, durch intuitive Darstellungen zu unterstützen und viele auch Analysemethoden zur Informationsgewinnung benutzen. Jedoch ist der Term nicht für Instandhaltungssysteme speziell festgelegt und wird für jedes Entscheidungsunterstützungssystem genutzt werden. Ein Aspekt, der bei diesen Systemen vernachlässigt wird, ist, dass die Datenintegration und die Herausforderungen zur Gewinnung und Extraktion aktueller Daten kaum eine Rolle spielen.

3.4.3 Weitere verwandte Systembegriffe

Neben diesen Systemen und den Produktionssystemen existieren in einigen Publikationen weitere Systeme, die zur Unterstützung der Instandhaltung eingesetzt werden. Jedoch ist der Fokus dieser Systeme meist nicht allein die Instandhaltung oder die Begriffe werden nur bei einigen wenigen Publikationen verwendet. Weiterhin sind manche der Systembegriffe nur eine Erweiterung eines bestehenden Systems mit einer bestimmten Technik oder Methode. Für die Vollständigkeit und Ableitung von Anforderungen in Kapitel 4 wird diese Gruppe der Systeme nachfolgend vorgestellt:

- **Computer aided RCM System:** Dieser Systembegriff wird von Cheng et al. (2010) benutzt, um ein System vorzustellen, welches das RCM Konzept unterstützt. Hierbei kann von einem speziellen CMMS für RCM ausgegangen werden.
- **Preventive Maintenance System:** Jin et al. (2009) und Fitouhi und Nourelfath (2012) nennen ihre vorgestellten Systeme Preventive Maintenance Systeme. Dies stellen unterstützende Systeme oder mathematische Modelle zur optimalen Planung von Maßnahmen der präventiven Instandhaltung dar. Diese Art von System können den CMMS und DSS zugerechnet werden.
- Qu (2006) und Hung et al. (2003) entwickeln in ihren jeweiligen Publikationen Systeme zur Ferndiagnose. Diese Systeme nennen sie **Remote Diagnostic System** (Qu, 2006) bzw. **e-diagnostics framework** (Hung et al., 2003). Weitere Autoren nutzen den Hinweis auf Fernwartung bzw. auf die Diagnose als Teilaktivität zur Beschreibung ihres Systems bzw. der verwandten Technik. Aufgrund der einfachen Zurechnung zu anderen Systemen wird der Begriff daher nicht als eigenständiges System bezeichnet und wird wie das reine Diagnosesystem als Teil eines CMS betrachtet.

- Der Begriff Teleservice, oder englisch Remote Service, wird in der Instandhaltungsliteratur verwendet, wenn der Fokus einer Instandhaltungslösung auf Ferndiagnose, -wartung und/oder -reparatur liegt, die Teil des Teleservices für die Instandhaltung ist (Renner, 2006, S. 17). Bamberger et al. (2010, S. 33) bezeichnen Teleservice als “eine mögliche Serviceform, Anfragen von der Service-Hotline weiter zu geben. Es wird versucht, über die Distanz das Problem des Kunden zu lösen. Je nach Ausgestaltung wird der Service mit unterschiedlichen Technologien unterstützt”. **Teleservicesysteme** sollen daher zum einen schnelle und effiziente Fehlerdiagnosen und zum anderen durch die Übermittlung von Zustandsdaten eine zustandsabhängige, vorbeugende Instandhaltung ermöglichen (Harms, 2007, S. 328). Die Funktionalität des Teleservices ist jedoch oft Teil von anderen Systemen wie MSS, E-MS, CMMS oder gar SCADA-Systemen und wird daher nicht als eigenständiger Begriff geführt.
- Ein **Product Service System (PSS)** ist eine integrierte Kombination von Produkten und Services, indem es die traditionelle Funktionalität eines Produktes mit zusätzlichen Services anreichert (Baines et al., 2007, S. 1543). Instandhaltung und Instandhaltungssysteme spielen als Teil eines PSS eine große Rolle, da sie das Produkt um Services, wie die Instandhaltung, erweitern können. Der Begriff zielt jedoch nicht alleine auf Instandhaltung ab und wird nicht im Speziellen für Instandhaltungssysteme oder Systeme genutzt, aus denen die Instandhaltung Daten zwingend benötigt.
- **Case-Based Reasoning Systeme/Systeme für Fallbasiertes Schließen** sind eine spezielle Form der Entscheidungsunterstützungssysteme bzw. der Assistenzsysteme. Solche Systeme benutzen die Methodik des Case-Based Reasonings (deutsch: „fallbasiertes Schließen“). Case-Based Reasoning (CBR) ist ein maschinelles Lernverfahren, das durch eine Basis von bereits gelösten Fällen, wie bspw. Reparaturfällen, ähnliche Fälle erkennt und eine Problemlösung bereitstellt. Diese Systeme werden in der Literatur meist jedoch anderen Systemen zugeschrieben und sind Teil von ihnen, z. B. den MSS (Nüttgens et al., 2014), E-MS (Rasoyska et al., 2005), RCM-Systemen (Cheng et al., 2010), Entscheidungsunterstützungssystemen (Rasovska et al., 2008) oder Wissensmanagement im weiteren Sinne (Potes Ruiz et al., 2013). Es wird daher nicht als einzelnes System betrachtet, sondern als einzelne Methodik.

3.4.4 Diskussion verbundene Systeme

Mit dieser Übersicht ist ersichtlich, dass viele Informationssysteme in einem produzierenden Unternehmen vorhanden sein können, aus denen Daten für die Instandhaltung benötigt werden können. Erkennbar ist aber auch, dass viele unterschiedliche Informationssysteme und damit verschiedene Ansätze existieren, die Instandhaltung zu unterstützen. Der Fokus der jeweiligen Systeme liegt bei den meisten Systemen auf einem ganz bestimmten Aspekt oder einer bestimmten Art, die Instandhaltung zu unterstützen.

Eine aktuelle Entwicklung, die bei der Weiterentwicklung der jeweiligen Systeme zu erkennen ist die, dass die ursprünglich für einen Einsatzzweck entworfenen Systeme zunehmend um die Fähigkeiten anderer Systeme erweitert werden, jedoch oft ihren ursprünglichen Namen behalten. Daher zeigt sich in der Praxis, dass die Grenzen der Systeme zunehmend verschwimmen. Es sind vermehrt Module in einzelnen Systemen verfügbar, welche die „typischen“ Aufgaben eines anderen Systems übernehmen. Darüber hinaus ist der Zweck der Informationssysteme meist nicht allein der Unterstützung der Instandhaltung zu dienen, sondern der Produktion, der Abrechnung oder der Bestellung von Ersatzteilen. Diese Punkte

können die Interpretation der vorhandenen Daten und Systeme eines Unternehmens erschweren. Dennoch sind die Anforderungen und Funktionen mancher Systeme stark unterschiedlich und erfordern eine komplett andere Konzipierung von Systemen.

Ein CMMS kann bspw. traditionell zentral aufgebaut werden, wobei ein CMS zwangsläufig dezentral sein muss, jedoch um zentrale Komponenten erweitert werden kann, um eine Fernwartung oder -überwachung zu gewährleisten. Ein PMS benötigt darüber hinaus Daten aus einer Vielzahl an dezentralen CMS und zur Berechnung zwangsläufig eine zentrale Instanz. Die Resultate einer solchen Analyse müssen weiterhin in ein CMMS übertragen werden, damit dieses System Wartungs- und Instandhaltungspläne erstellen kann.

Aus diesem Beispiel können zwei Schlüsse gezogen werden: Zum einen ist allein aus der Anzahl der genannten Systeme erkennbar, dass eine Vielzahl von potentiellen Schnittstellen notwendig ist, damit die unterschiedlichen Informationen sinnvoll genutzt werden können. Zum anderen ist es nötig, dass die einzelnen Aspekte der Systeme je nach Bedarf miteinander kombiniert oder zumindest Daten miteinander ausgetauscht werden können, um das volle Potential der IT-Unterstützung der Instandhaltung zu erreichen. Dazu muss die interne IT-Architektur eines Unternehmens es ermöglichen, diese Verbindungen zwischen den Systemen zu etablieren. Um eine solche Architektur zu ermitteln, müssen die essentiellen Funktionalitäten eines jeden Systems und insbesondere die Kernfunktionalitäten (Funktionalitäten, die zwingend erforderlich sind) bestimmt und in einen ganzheitlichen Kontext gesetzt werden.

Daher ist eine umfassende Sicht auf ein Informationssystem zur Unterstützung der Instandhaltung nötig, welches zum einen den Informationsbedarf und -austausch aus den typischen Informationssystemen für die Produktion erleichtert und zum anderen die Integration aller Aspekte der Systeme zur Unterstützung der Instandhaltung ermöglicht. Dieses Informationssystem wird daher nicht nach einem bestehenden System zur Unterstützung der Instandhaltung benannt, sondern als integriertes Informationssystem zur Unterstützung der Instandhaltung bezeichnet, da es die jeweiligen Informationen und Anforderungen der bestehenden Systeme integrieren kann und eine umfassende Sicht auf alle möglicherweise benötigten Ressourcen zulässt.

Zusammenfassend können daher folgende Aspekte für den Betrachtungsrahmen der RAI abgeleitet werden:

- Die Referenzarchitektur umfasst die essentiellen Funktionalitäten der untersuchten Informationssysteme zur Unterstützung der Instandhaltung.
- Die Referenzarchitektur umfasst eine integrierte Sicht über die Informationssysteme, sodass eine Verbindung aus Funktionalitäten aus mehreren System hergestellt werden kann.
- Die Referenzarchitektur muss Teilsysteme, die bestimmte Funktionalitäten abdecken, integrieren können.
- Die Referenzarchitektur muss die essentiellen Schnittstellen zu den in der Produktion vorhandenen Systemen unterstützen.

Zur Ergründung der essentiellen Funktionalitäten, Anforderungen und Bestandteile der RAI dienen die identifizierten Systeme.

3.5 Zusammenfassung

In diesem Kapitel wurde die Domäne der Instandhaltung vorgestellt. Begrifflichkeiten wurden definiert und erläutert. Dabei wurden Instandhaltungskonzepte und -strategien sowie damit verbundene Aspekte vorgestellt. Um alle wichtigen Arten und Strategien effizient umzusetzen, sind viele Informationen aus den typischen Informationssystemen eines produzierenden Unternehmens nötig. Diese spielen für eine RAI eine große Rolle, da Informationen aus diesen für eine effektive Instandhaltung notwendig sind. Diese Systeme wurden daher mitsamt ihrer Funktionalität in diesem Kapitel beschrieben. Weiterhin wurden aktuelle Systeme vorgestellt, die explizit das Ziel haben die Instandhaltung zu unterstützen.

Anhand dieser Auflistung und einem Vergleich wurde zum einen sichtbar, dass Abgrenzungen zwischen manchen Systemen zur Unterstützung der Instandhaltung trotz unterschiedlicher Fokusse oftmals schwierig sind. Die Beschreibungen und Definitionen der jeweiligen Systeme überschneiden sich in bestimmten Bereichen enorm. Dies führt auch dazu, dass die Systeme oft mit ähnlichen oder identischen Daten arbeiten. Dies ist insbesondere für die Zusammenführung dieser herausfordernd, wenn nicht in jedem System Datensilos entstehen sollen. Zum anderen wurde sichtbar, dass die Gesamtheit der Systeme mit ihren unterschiedlichen Fokussen zusammen betrachtet die gesamte Bandbreite der Unterstützung der Instandhaltung abdecken. Ein System, das die Instandhaltung effizient unterstützen will, muss daher einen systemübergreifenden Blick besitzen und die jeweiligen Aspekte zusammenführen.

Aus diesem Grund wurde der Begriff des IIS gewählt, da die zu entwickelnde Referenzarchitektur systemübergreifend funktionieren soll, keinen Aspekt der jeweiligen Systeme überbetonen will und sich auch nicht als reinen Ersatz, sondern auch als Architektur zur Bündelung vorhandener Systeme sieht. Dadurch sollen die auf dieser Referenzarchitektur zu entwickelnde Systeme integriert, d. h. mit dem umfassenden Blick auf alle Systeme zur Unterstützung der Instandhaltung entwickelt bzw. erweitert werden können. Mitsamt des in Kapitel 2 und 3 gewonnenem Domänenwissens wurde damit der Grundstein für die in Kapitel 4 durchzuführende Anforderungsaufnahme gelegt.

4 Extraktion generischer Anforderungen an integrierte Instandhaltungssysteme

4.1 Einleitung

In Kapitel 2 wurde die Methodik zur Erstellung einer Referenzarchitektur anhand spezifischer Anforderungen vorgestellt. Die Ergebnisse der ersten Schritte dieses Vorgehens werden in diesem Kapitel mit Fokus auf die spezifischen Anforderungen präsentiert, welche die Grundlage der RAI bilden. Aufbauend auf dem Domänenwissen aus Kapitel 3 wird zu Beginn eine Literaturanalyse bzgl. der Anforderungen an ein IIS durchgeführt und deren Ergebnisse präsentiert. Basierend auf diesen Literaturanforderungen werden zusätzlich Anforderung an ein IIS bei vier Praxisunternehmen aufgenommen. Aus diesen wissenschaftlichen und praktischen Anforderungen wird die Grundlage für die Referenzarchitektur gewonnen, indem generische und optionale Anforderungen an ein IIS extrahiert werden.

Das Kapitel gliedert sich in eine Vorstellung der in Kapitel 4.2 durchgeführten Literaturanalyse, in dem auch die Ergebnisse hinsichtlich der Häufigkeit und Verteilung der Anforderungen diskutiert werden. In Kapitel 4.3 wird die Anforderungsaufnahme bei vier Praxisunternehmen vorgestellt und deren Ergebnisse präsentiert. Dabei werden generische und optionale Anforderungen an ein IIS extrahiert. Abschließend werden die Ergebnisse in Kapitel 4.4 diskutiert und in Kapitel 4.5 erfolgt die Zusammenfassung des gesamten Kapitels.

4.2 Literaturanalyse: Anforderungen an ein integriertes Instandhaltungssystem

Mittlerweile sind für die effiziente Nutzung der meisten untersuchten Systeme Daten aus einer Vielzahl von Sensoren und Systemen zu sammeln und zu verarbeiten. Darüber hinaus ist eine Verbindung der einzelnen Systeme und damit eine integrierte Sicht notwendig, um die Synergien und Potentiale von neuen Technologien zu nutzen. Zur Erlangung dieser Sicht wird nachfolgend eine Literaturanalyse durchgeführt, welche die Anforderungen für jedes der in Kapitel 3 vorgestellten Systeme untersucht. Diese werden miteinander verglichen und systemübergreifend zusammengeführt. Daraus werden umfassende Anforderungen extrahiert, die als Basis für weitere Anforderungserhebungen und den Bau von integrierten Architekturen für Instandhaltungssystemen dienen können. Das nachfolgend dargestellte Vorgehen und die Ergebnisse basieren dabei auf der Publikation von Reidt et al. (2018b) und wurden für diese Arbeit umfassend erweitert und näher analysiert.

Ziel der Literatursuche ist es daher, die bekanntesten Artikel bzgl. Anforderungen für Instandhaltungssysteme aus unterschiedlichsten Domänen und für unterschiedliche Arten der Systeme abzuleiten. Um dies zu ermöglichen, wird möglichst breit und domänenübergreifend gesucht.

4.2.1 Vorgehen Literaturanalyse

Um die Nachvollziehbarkeit der Literatursuche zu gewährleisten und diese zu systematisieren, werden eindeutige Schlagwörter und daraus resultierende Kombinationen festgelegt, nach denen die jeweilige Suche in Literaturdatenbanken durchgeführt wird.

Die Schlagwörter speisen sich aus den Bezeichnungen für die in Kapitel 3.4 vorgestellten Systemen, den Begriffen der Instandhaltung und den jeweiligen Instandhaltungskonzepten und -strategien aus den Kapiteln 3.2 und 3.3 sowie bestimmten Begriffen, die in Verbindung mit der Instandhaltung und deren Tätigkeiten stehen. Bei letzterem werden insbesondere die Begriffe „industrial maintenance“, „maintenance management“, „maintenance control“ und „maintenance performance“ als allgemeine Begriffe benutzt und mit anderen kombiniert. Alle Begriffe werden mit den Bezeichnungen „system“ und/oder „requirements“ und/oder „maintenance“ angereichert, um Literatur zu finden, die sich den Informationssystemen und Anforderungen auseinandersetzt. Die Zusammenstellung der Schlagwörter für eine jeweilige Suche in der Literaturdatenbank wird im Folgenden als „Suchterm“ bezeichnet.

Die Suchterme werden sowohl in englischer als auch in deutscher Ausführung benutzt. Als Literaturdatenbank/Suchmaschine wurde Google Scholar benutzt, da dies eine domänenübergreifende Suche insbesondere bei sehr breiten Suchtermen ermöglicht. Die Ergebnisse von Google Scholar sind weitaus umfassender hinsichtlich nicht rein wissenschaftlicher Artikel. Diese Art von Artikeln spielt insbesondere für die Auswertung von Anforderungen an Instandhaltungssysteme eine relevante Rolle, da Prototypen und Instandhaltungssysteme häufig auch in populärwissenschaftlicher Literatur beschrieben werden. Die Nachteile der Benutzung von Google Scholar (vgl. dazu (Haddaway et al., 2015)) werden dahingehend reduziert, dass auf Basis einer bereits vorhandenen, breiten Literaturbasis gesucht wird und die Suchstrings so gewählt werden, dass sämtliche bereits vorhandene Literatur gefunden werden kann. Aufgrund der Komplexität und der Vielzahl an Ergebnissen

bei bestimmten Suchtermen werden nur die relevantesten Ergebnisse untersucht. Eine Vorwärts- und Rückwärtssuche wird durchgeführt und bei Bedarf werden neue Suchterme hinzugefügt, damit alle relevanten Beiträge auch gefunden werden können. Auf eine genauere Dokumentation der Vorwärts- und Rückwärtssuche wird jedoch verzichtet. Die Suchzeitraum für die Publikationen reicht von 2006 bis 2016, um möglichst alle Anforderungen zu finden, die zur technologischen Entwicklung passen. Patente und Zitationen werden ausgeklammert. Die Relevanz einer Publikation hinsichtlich des Ziels der Literaturanalyse wird in mehreren Schritten bewertet. Dazu werden die Ergebnisse eines Suchterms in drei Kategorien eingeteilt:

- Die erste Kategorie umfasst alle Treffer des Suchterms,
- die zweite Kategorie umfasst interessante Publikationen, deren Relevanz jedoch nur anhand des Titels und des Abstracts bewertet wird.
- Die dritte Kategorie umfasst alle benutzten Publikationen. Sofern diese als interessant angesehen werden, werden sie dahingehend untersucht, ob Anforderungen aus ihnen extrahiert werden können. Ist dies der Fall, werden sie in der Literaturanalyse verwendet und finden sich dementsprechend in dieser Kategorie wieder. Anforderungen können extrahiert werden, wenn Publikationen diese Anforderungen an Instandhaltungssysteme explizit nennen oder wenn sie die implizite Ableitung von Anforderungen an solche Systeme ermöglichen. Bei folgenden Fällen können diese implizit extrahiert werden:
 - Belegte und beschriebene Gründe für die Entwicklung eines Systems, aus denen sich Anforderungen ableiten lassen.
 - Erfahrungen und Beschreibungen von bestimmten Systemen, d. h. welche Erfahrungen können damit gemacht werden, in welchen Fällen sind sie erfolgreich? Darüber hinaus: welche Funktionen und Elemente enthalten sie, sodass aus diesen Anforderungen destilliert werden können?
 - Literaturstudien oder Praxisberichte über Probleme und erfolgreichen Einsatz bestimmter Systeme, aus denen sich Anforderungen ableiten lassen.
 - Wenn implizite Eigenschaften/Features von Systemen übernommen werden können,⁰ da diese genau erkennbar sind oder ausführlich beschrieben werden.

Nachfolgend werden in Tabelle 4 die Suchterme samt ihren Ergebnissen aufgelistet. Dabei basieren die einzelnen Spalten der Tabelle auf folgenden Begriffen:

- **Suchtermnummer (StNr.):** Die Nummer des Suchterms zur Kategorisierung.
- **Suchterm:** In dieser Spalte wird der in Google Scholar verwendete Suchterm genauso eingetragen, wie er in die Suchmaske eingegeben wurde. Terme in Anführungszeichen müssen genau in dieser Form in der untersuchten Literatur vorkommen, Terme ohne Anführungszeichen hingegen können eine andere Reihenfolge oder Zusammensetzung in der Literatur besitzen.
- **Anzahl der untersuchten Seiten (AuS.):** Bei einer sehr großen Anzahl an Treffern bei einem bestimmten Term werden nicht alle Ergebnisse untersucht, da die Treffer vom ersten Ergebnis ausgehend deutlich an Relevanz verlieren. Die Spalte gibt daher an, wie viele Treffer untersucht werden. Pro Suchterm wird entweder die Anzahl der durchsuchten Seiten angegeben (z. B. AnS. = Ende Seite (E. S., 11 bedeutet, dass bis zur Ende Seite 11 durchsucht wurde) oder falls die kompletten Ergebnisse durchsucht wurden wird dies per „k.“ angegeben. Eine Seite enthält 20 Ergebnisse.
- **Ergebnisse (Erg.):** Die Anzahl der Treffer der jeweiligen Suchanfrage, wie Google Scholar sie zurückgibt, ohne jegliche Bewertung.

- **Interessant (Int.):** In dieser Kategorie der interessanten Publikationen finden sich alle Treffer, die als interessant eingestuft werden. Diese Einstufung wird nur anhand des Titels und des Abstracts vorgenommen.
- **Benutzt (Ben.):** Hierunter fallen Paper, die zur Extraktion von Anforderungen an Instandhaltungssysteme genutzt werden können.
 - Beiträge, die sich vor allem der Leistungsmessung und Beurteilung sowie der Bildung von KPIs ohne Systembezug widmen, werden nicht in dieser Spalte inkludiert. Übersichten finden sich hierzu bei (Kumar et al., 2013; Stenström et al., 2013).

StNr.	Suchterm	AuS.	Erg.	Int.	Ben.
1	"Product-Service Systems" maintenance requirements	E. S. 11	4090	17	1
2	"produkt service system" instandhaltung anforderung	k.	28	0	0
3	"produkt service system" Wartung Anforderung	k.	42	0	0
4	Diagnosesystem Instandhaltung System	k.	302	9	2
5	"Diagnostic system" maintenance requirements	E. S. 4	11000	6	2
6	"fault diagnostic system" maintenance requirements	k.	525	12	4
7	Wartungsmanager instandhaltung system	k.	22	1	1
8	"Preventive maintenance requirements" system	k.	168	13	0
9	"Preventive maintenance system" requirements	E. S. 4	562	1	1
10	Predictive maintenance system requirements	E. S. 4	495	7	2
11	maintenance "decision support system" requirements	E. S. 20	17400	16	4
12	"E-maintenance" system	E. S. 20	4970	19	2
13	"e-Maintenance system" requirements	E. S. 4	341	14	8
14	"intelligent maintenance" system	E. S. 20	2800	8	4
15	cmms maintenance system	E. S. 20	5500	6	1
16	"Computerized Maintenance Management System" system	E. S. 20	2030	4	2
17	instandhaltungsplanungs- und -steuerungssystem	E. S. 4	507	3	0
18	"Instandhaltungsplanungs und -steuerungssystem" Anforderungen	k.	34	4	1
19	Zustandsüberwachung instandhaltung Anforderung	k.	226	2	0
20	Condition Monitoring System Requirements	E4	5510	6	1
21	"Mobiles Assistenzsystem"	k.	33	4	2
22	"mobile system" requirement "customer service"	E. S. 20	467	4	3
23	"mobile system" requirement "field service"	k.	71	3	2
24	"mobiles System" technischer Kundendienst	k.	93	2	2
25	"mobile support system" maintenance requirements	k.	109	9	3
26	teleservice instandhaltung system	E. S. 17	176	3	0
27	Teleservice maintenace system requirements	E. S. 20	1150	7	3
28	"remote service maintenance" requirements system	k.	5	1	1

StNr.	Suchterm	AuS.	Erg.	Int.	Ben.
29	Fernwartung System Instandhaltung Anforderungen	k.	188	0	0
30	requirements "industrial maintenance"	E. S. 20	2420	9	3
31	"industrial maintenance" "internet of things"	k.	83	4	1
32	"industrial maintenance" cloud	E. S.20	234	7	2
33	"industrial maintenance" Wearable Computing	k.	195	11	1
34	intelligent factory "industrial maintenance"	E. S. 20	319	12	3
35	"maintenance performance" requirements system	E. S. 20	5360	40	4
36	"maintenance control" requirements systems	E. S. 20	3230	28	3
37	maintenance management requirements system	E. S. 20	Ü10k	27	8
38	"Value driven maintenance" requirements system	k.	101	24	2
39	"Risk-based maintenance" requirements system	E. S. 20	1390	20	1
40	"reliability-centered maintenance" requirements system	E. S. 20	4280	23	4
41	"RCM System" requirements	E. S. 4	355	8	1
42	"TPM System" Requirements	E. S. 4	274	2	0
43	"Total Productive Maintenance" System Requirements	E. S. 4	7870	9	0
Summe		1164955		405	85
Insgesamt benutzte Paper					61
Insgesamt benutzte Paper ohne Duplikate von Demonstratoren/Prototypen					56

Tabelle 4: Suchterme und (benutzte) Ergebnisse der Literaturrecherche

Quelle: Eigene Darstellung

Es wurden insgesamt 43 unterschiedliche Suchterme benutzt, die aus 61 benutzten Publikationen resultierten. Einzelne Prototypen und Entwicklungen wurden in mehreren Beiträgen vorgefunden. Um zu verhindern, dass das Ergebnis durch Vielfachnennung einzelner Prototypen und deren Anforderungen verzerrt wird, wird einer der Beiträge als Referenz in die Literaturanalyse integriert, die anderen hingegen ausgeschlossen. Die Anforderungen aus den ausgeschlossen Beiträgen werden gesammelt und als eine Anforderung aus dieser Referenzpublikation gezählt. Mehrfachnennungen der Anforderungen werden ausgeschlossen und nur einmal gezählt. Nachfolgend sind die Beiträge genannt auf welche dies zutrifft und derjenige Beitrag gefettet markiert, der stellvertretend für die exkludierten Beiträge in die Analyse miteinbezogen wurde.

- Prototypen im Rahmen des Projektes DYNAMITE werden in folgenden Beiträgen behandelt:
 - Development of a maintenance system based on web and mobile technologies (Campos et al., 2007)
 - A Mobile Maintenance Decision Support System (Campos et al., 2008)
 - **A web and mobile device architecture for mobile e-maintenance** (Campos et al., 2009)
- Prototypen und Systeme, die im Rahmen eines Projektes für Saab bzw. das schwedische Militär entwickelt wurden, werden in den folgenden Beiträgen behandelt:
 - eMaintenance-Information logistics for maintenance support (Candell et al., 2009)

- A service-oriented approach to e-maintenance of complex technical systems (Karim, 2008)
- Development of ICT-based maintenance support services (Karim et al., 2009)
- **Development of Information System for e-Maintenance Solutions within the Aerospace Industry** (Candell et al., 2011)

4.2.2 Diskussion der Ergebnisse

Nach der Reduzierung der gewonnenen Ergebnisse werden für die nachfolgende Ergebnisvorstellung und -diskussion 56 Publikationen genutzt. Diese Beiträge lassen sich nach dem in Tabelle 5 vorgestellten Schema klassifizieren. Dort werden die Beiträge anhand der Autoren und dem Veröffentlichungsjahr genannt und folgende Kriterien dem Beitrag zugeteilt:

- **System (Sys.):** Die untersuchten Beiträge stellen ein selbstentwickeltes System zur Unterstützung der Instandhaltung vor und beschreiben dieses. Dabei handelt es sich meist um einen speziellen Prototyp oder ein spezielles System, das in der Industrie im Einsatz ist.
- **Vo./VE./FW.:** Diese Beiträge stellen eine Vorgehensweise, ein Verfahren oder ein Framework vor, das unmittelbar der Instandhaltung zugeordnet ist, jedoch nicht in einem Software-Prototyp umgesetzt wurde. Anhand dieser Beschreibung lassen sich (implizite) Anforderungen an ein Instandhaltungssystem ableiten, welches die elementaren Teile der Vorgehensweisen umsetzt.
- **Explizite Nennung von Anforderungen (Ex. Anf.):** Beiträge dieser Kategorie nennen explizite Anforderungen an ein bestimmtes Instandhaltungssystem, diskutieren diese und oftmals sind die Anforderungen selbst der Hauptfokus des Beitrags.
- **Literaturreview Anforderungen an ein bestimmtes System (Lr. AS):** Beiträge, die Literaturübersichten über Anforderungen an eine bestimmte Systemklasse, wie bspw. MSS (Ebke/Däuble, 2015) aggregieren.
- **(Literatur-)Review Trends, Funktionen, Systeme (Lr. TFS):** Hierunter fallen Beiträge, die Literaturübersichten oder technologische Übersichten beinhalten, die Trends, Funktionen und Systeme der Instandhaltung behandeln. Die herausgearbeiteten Eigenschaften und Verbindungen zu Instandhaltungssystemen lassen sich in Anforderungen überführen oder werden explizit genannt.
- **System-/Technologiefokus:** In dieser Spalte werden die technischen Systeme oder Technologien eingetragen, denen die Publikationen thematisch zugeordnet werden können. Beschreibt die Publikation Anforderungen oder Konzepte zur Anforderungserhebung bei einem CMMS, dann ist CMMS in dieser Spalte einzutragen. Ist ein Beitrag eindeutig dem Condition Monitoring zuzuordnen oder beinhaltet es ein CMS, so steht in der Spalte CMS. Werden Trends in der Instandhaltung bei der Publikation behandelt und lassen sich mehrere Anforderungen bzgl. Remote Maintenance und CMS ableiten, dann werden diese beiden Systeme eingetragen. Es muss sich demzufolge nicht um eine Publikation handeln, die nur ein bestimmtes System behandelt. Für die Systembeschreibung werden folgende Abkürzungen in der Tabelle verwendet:
 - WMS = Wissensmanagementsystem
 - PSS = Product Service System
 - CMMS = Computerized Maintenance Management System
 - E-MS = E-Maintenance System

- PMS = Predictive Maintenance System
- CMS = Condition Monitoring System
- DSS = Decision Support System
- SP = Serviceplattform
- RM = Remote Maintenance
- TKD = Technischer Kundendienst (System oder Fokus)
- SCADA = SCADA System
- OPT = Systemfokus mit Methoden der (mathematischen) Optimierung
- MSS = Mobile Support System
- PIS = Systemfokus mit System zur Performance Messung und Nachverfolgung
- DS = Diagnosesysteme
- IMS = Intelligent Maintenance System
- AR = Augmented Reality
- RCM = System mit Fokus auf das Reliability Centered Maintenance
- PER = Systeme mit Fokus Performancemessung

Publikation/Kriterium	Sys.	Vo./Ve. /FW.	Ex. Anf.	LR. AS.	Lr. TFS.	System- /Technologiefokus
(Zhu et al., 2012)	x					WMS, PSS
(Fellmann et al., 2011)	x		x			TKD
(Crespo Marquez/Gupta, 2006)		x			x	CMMS
(Candell et al., 2011, 2009; Karim, 2008; Karim et al., 2009)	x					E-MS
(Kans, 2008)			x			CMMS
(Swift et al., 2011)	x					PM, CMS, DSS
(Pistofidis et al., 2012)	x					E-MS
(Anaya-Lara et al., 2006)					x	RM, SCADA
(Muller et al., 2008a)					x	E-MS
(Potes Ruiz et al., 2014)		x				WMS
(Qingfeng et al., 2011)	x					PM, OPT
(Sharma et al., 2011)					x	OPT
(Campos et al., 2007, 2009, 2008)	x					MSS, E-MS, DSS
(Gómez Fernández/Crespo Márquez, 2009)		x				OPT
(Carnero/Novés, 2006)		x	x			CMMS
(Hwang et al., 2007)	x					CMMS, OPT, PER
(Potes Ruiz et al., 2013)	x					WMS
(Arnaiz et al., 2006)		x			x	MSS, E-MS, CMS
(Cannata et al., 2009)					x	E-MS
(Ebke/Däuble, 2015)			x	x	x	MSS, TKD
(Matijacic et al., 2013)			x	x	x	MSS, TKD

Publikation/Kriterium	Sys.	Vo./Ve./FW.	Ex. Anf.	LR. AS.	Lr. TFS.	System-/Technologiefokus
(Grundnig/Raschhofer, 2010)	x					RM, DS
(Schmidt/Hoof, 2013)	x		x			SP
(Persona et al., 2007)	x				x	E-MS
(Fonseca et al., 2008)s	x					PM
(Kłos/Patalas-Maliszewska, 2013)	x					CMMS, PER
(Marquez/Iung, 2008)					x	E-MS
(Goh et al., 2006)					x	CMS, PM
(Muller et al., 2008b)	x					CMS, PM
(Lee et al., 2006)	x					E-MS, CMS, PM, IMS
(Tsang et al., 2006)		x				CMS, OPT
(Lee et al., 2013)	x					DSS, CMS, IMS
(Lee/Akin, 2011)	x		x			MSS, AR
(Savioja et al., 2007)	x					MSS, AR
(Bahga/Madisetti, 2012)	x					CMS, PM, WMS, DS
(Pistofidis et al., 2016)	x					PM, WMS
(Wang et al., 2007)	x					MSS, CMS, RM
(Ni/Jin, 2012)	x					DSS, OPT
(Efthymiou et al., 2012)	x					CMS, WMS
(Ali et al., 2008)	x					MSS, CMS, RM
(Karray et al., 2011)	x					E-MS, WMS
(Levrat/Iung, 2007)	x					E-MS, CMS, RM
(Wang et al., 2006)	x					CMS, PM, RM, DSS, IMS, OPT
(Houy et al., 2012)	x		x			SP, TKD
(Hameed et al., 2010)	x		x			CMS
(Schlicker, 2014)	x					MSS, SP, TKD
(Syafar/Gao, 2013)		x	x			MSS
(Özcan et al., 2013)	x		x			MSS
(Ziegler et al., 2011)	x					MSS
(Daeuble et al., 2015b)			x	x		MSS, TKD
(Niemöller et al., 2016)			x	x		MSS, AR
(Fay, 2009)			x		x	RM
(Kühl/Fay, 2010)			x		x	RM, TKD
(Kamsu-Foguem/Noyes, 2013)		x				WMS
(Kamsu-Foguem/Mathieu, 2014)	x					RM
(Bae et al., 2009)	x					RCM

Publikation/Kriterium	Sys.	Vo./Ve. /FW.	Ex. Anf.	LR. AS.	Lr. TFS.	System- /Technologiefokus
Summe	35	8	15	4	13	
Prozentual von 56 Beiträgen	62,5%	14,3%	26,8%	7,1%	23,2%	

Tabelle 5: Klassifikation der Ergebnisse der Literaturanalyse von Instandhaltungssystemen

Quelle: Eigene Darstellung

Die Ergebnisse zeigen, dass ein Großteil der Beiträge (62,5 %) direkt entwickelte Systeme vorstellt und diskutiert. Diese Systeme haben in den meisten Fällen einen bestimmten Hauptfokus (z. B. CMS, Optimierung oder Predictive Maintenance) und ermöglichen einen umfassenden Blick auf die aktuellen Entwicklungen zu Instandhaltungssystemen. Von diesen Beiträgen, die ein speziell entwickeltes System behandeln, werden jedoch nur in sechs Publikationen (17 %) explizite Anforderungen genannt, beim Rest mussten die Anforderungen implizit extrahiert werden. 14,3 % aller Beiträge behandeln Verfahren, Vorgehensweisen und Frameworks, ohne ein eigenes Informationssystem zu entwickeln. Diese Beiträge ermöglichen es, Anforderungen zu extrahieren, da Informationssysteme zur Durchführung der Vorgehensweisen bestimmte Anforderungen erfüllen müssen oder diese explizit genannt werden. 23,2 % der Beiträge sind der Kategorie des Reviews von Systemen, Trends und Funktionen der Instandhaltung zuzuordnen. Nur vier Beiträge und damit 7,1 % sammeln per Literaturreview Anforderungen für ein bestimmtes Instandhaltungssystem. Diese Publikationen sind dabei einem bestimmten Institut bzw. Fachrichtung zuzuordnen und betrachten ausschließlich Anforderungen für MSS. Insgesamt werden explizite Anforderungen für ein Instandhaltungssystem bei 26,8 % der Beiträge beschrieben.

Die meisten Beiträge lassen sich nach verschiedenen Arten von Systemen oder Technologien kategorisieren. Mehrfachkategorisierungen sind dabei häufig anzufinden, da bei manchen Publikationen keine klare Grenze zwischen den betrachteten Systemen gezogen werden kann oder Systeme zusammen mit einer bestimmten Technologie verwendet werden.

Die am häufigsten behandelten Systeme bzw. Themengebiete sind in Abbildung 11 dargestellt. Dort ist die Anzahl der Nennungen pro System auf der X-Achse dargestellt. Die schwarze Linie zeigt die aggregierte Summe über alle Systeme hinweg in Prozent. Sie steigt von links beginnend bis zum Ende der X-Achse auf 100 % an. Hierbei ist erkennbar, dass die ersten fünf Systeme schon 50 % der Nennungen ausmachen.

Abnehmend nach der Anzahl sortiert ergibt sich Folgendes: MSS (14), CMS (13) und E-MS (11) haben jeweils über 10 Nennungen. Ebenso häufig werden Systeme im Bereich PMS, Remote Maintenance, TKD, WMS, CMMS und DSS genannt (jeweils über 5). IMS, Serviceplattformen, Diagnose-, Performance- RCM- sowie PS-Systeme sind hingegen seltener vertreten (unter 5). Techniken wie Optimierung (7) und AR (3) werden genannt. Letztere besonders häufig verbunden mit MSS.

Hier zeigt sich, wie bereits in Kapitel 3 erwähnt, dass die Übergänge zwischen den Systemen und den benutzten Technologien teilweise fließend sind, da zum einen bestimmte Anforderungen sehr ähnlich zwischen verschiedenen Systemklassen und bestimmte Anforderungen bei fast allen Systemklassen vorzufinden sind. Zum anderen kann ein Beitrag vielfach mehreren Systemen und Systemeigenschaften zugeordnet werden. Es gibt einige Systeme, die einen besonderen Fokus und teilweise spezielle Anforderungen haben, wie bspw.

MSS. Eine eindeutige Unterscheidung zwischen bestimmten Systemgruppen ist jedoch schwer zu treffen, was sich ebenso in den extrahierten Anforderungen zeigt. Hier wird wiederum der Bedarf nach einer integrierten Sichtweise sichtbar, um bestimmte Aspekte nicht mehrfach abzudecken.

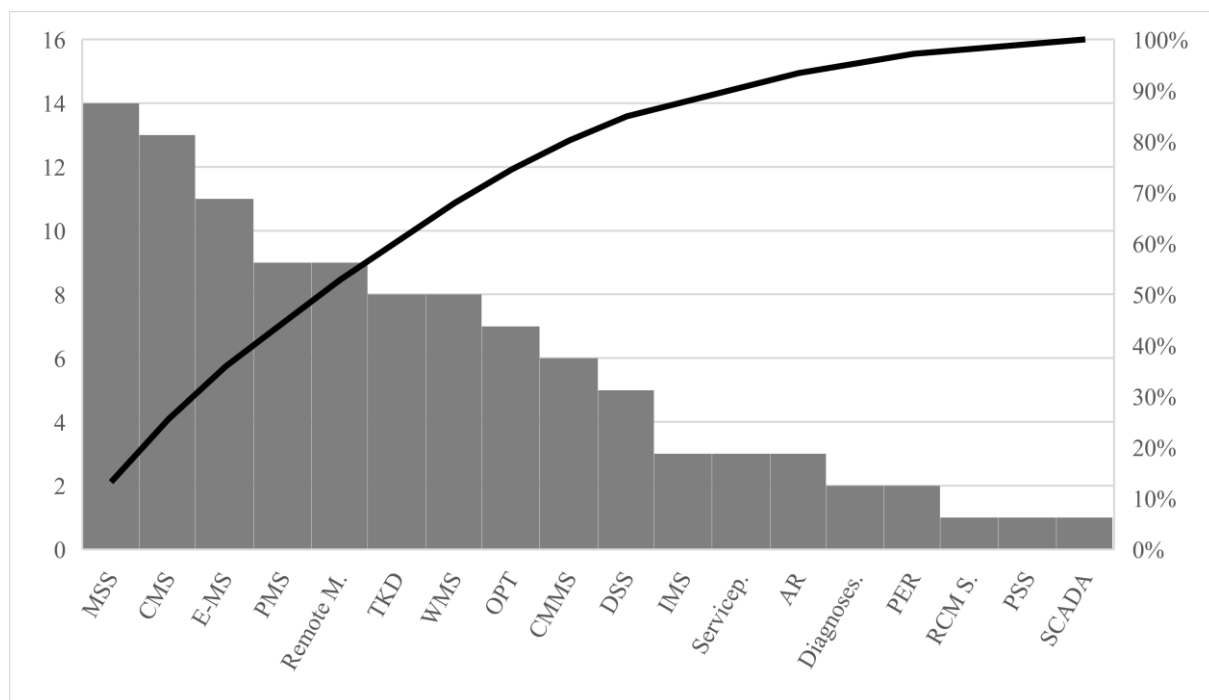


Abbildung 11: Häufigkeit von Systembezeichnungen und Techniken in den Ergebnissen der Literaturrecherche

Quelle: Eigene Darstellung

Durch diese große Bandbreite an untersuchten Systemen, die jeweils zwar unterschiedliche Schwerpunkte besitzen, jedoch ebenso viele Überschneidungen, kann davon ausgegangen werden, dass anhand der zu untersuchenden Anforderungen an die einzelnen Instandhaltungssysteme zusammengenommen eine integrierte Sicht auf Anforderungen an Instandhaltungssysteme abgeleitet werden kann.

Um eine umfassende Sicht auf die Anforderungen an ein Instandhaltungssystem abzudecken, bedarf es weiterhin, dass unterschiedliche Sichtweisen aus verschiedenen Domänen und Forschungsrichtungen in die Untersuchung integriert werden. Ansonsten könnten die Publikationen einer einzelnen Domäne nur Anforderungen abdecken, die für eine Domäne relevant wären oder diese zu stark gewichten, und stattdessen Anforderungen aus anderen Domänen nicht behandeln. Dass die untersuchten Publikationen eine umfassende domänenübergreifende Sicht auf Anforderungen an die Instandhaltungssysteme erlauben, spiegelt sich in der Betrachtung der Forschungsdomäne der 56 Publikationen wieder. Abbildung 12 zeigt die Verteilung der untersuchten Paper nach vorgenommener Einteilung nach Domänen.

Dabei wird aufgrund des Publikationsortes (Konferenz, Journal) eine Kategorisierung nach folgenden Domänen vorgenommen:

- **P/IA/MB:** Publikationen, die sich der Produktion(-sforschung), Industrieautomatisierung oder Maschinenbauthemen widmen, werden in diese Kategorie einsortiert. Aufgrund der schwierigen Abgrenzbarkeit zwischen den einzelnen Themen wird keine Unterscheidung zwischen diesen vorgenommen. Jedoch

kann bei den unterschiedlichen Themenbereichen von einer großen Bandbreite gesprochen werden.

- **Informatik:** Publikationen, die einen klaren Bezug zu reinen Informatikthemen besitzen, werden in diese Kategorie einsortiert. Hier spielen vor allem Big-Data und Softwarearchitekturen eine Rolle.
- **Wirtschaftsinformatik:** In diese Kategorie werden Beiträge eingeordnet, die klar dem Spektrum der Wirtschaftsinformatik zugeordnet werden können.
- **Instandhaltung:** Beiträge, die in reinen Instandhaltungspublikationen veröffentlicht werden oder klar der Instandhaltung zuzuordnen sind, werden in diese Kategorie einsortiert.
- **Divers:** Diese Kategorie sammelt alle Beiträge, die keiner Kategorie oder einer Kategorie, die jedoch keine weitere große Anzahl an Publikationen enthielt, zugeordnet werden können. Darunter befinden sich bspw. Beiträge aus dem Kontext der erneuerbaren Energien, Sicherheitsforschung, Wirtschaft und Serviceforschung.

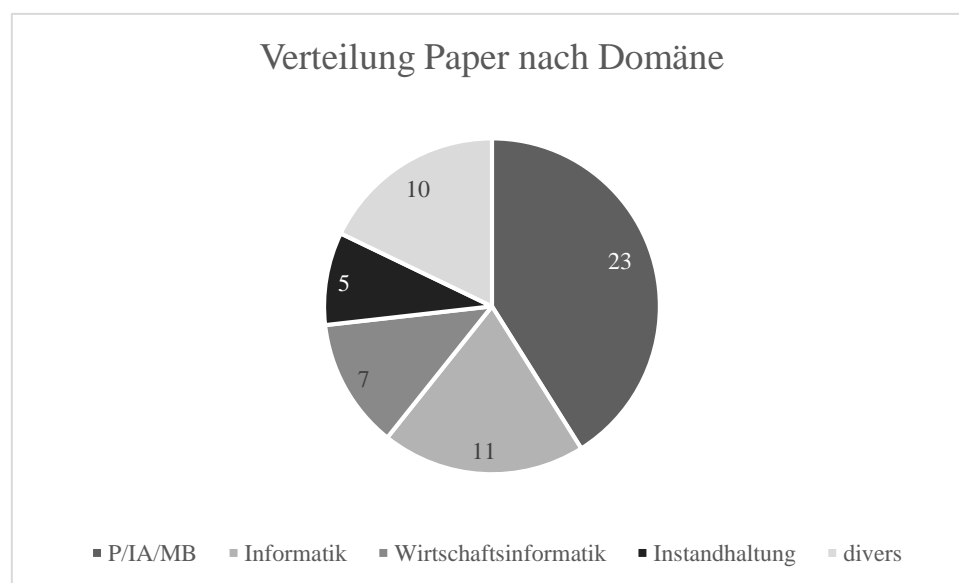


Abbildung 12: Verteilung der gefundenen Artikel nach Wissenschaftsdomäne

Quelle: Eigene Darstellung

Die Verteilung der Beiträge und die damit verbundenen Publikationsorte zeigen, dass das Thema der Instandhaltungssysteme ein domänenübergreifendes Thema ist. Dies zeigt sich auch dadurch, dass viele Konferenzen und Journale aus dem Bereich des P/IA/MB einen interdisziplinären Charakter betonen. Hier können 23 Publikationen dem mit Intention breit gewählten Bereich zugeordnet werden. Am zweithäufigsten sind die Beiträge, die der Informatik zuzurechnen sind. 10 Beiträge lassen sich in die Kategorie Divers zuordnen. Die Bereiche Wirtschaftsinformatik (7) und Instandhaltung (5) folgen als nächstes.

Die Ergebnisse illustrieren, dass durch die Literaturrecherche und die verwendeten Suchterme Beiträge gefunden wurden, die zum einen unterschiedliche Instandhaltungssysteme behandeln und zum anderen diese Systeme aus verschiedenen Wissenschaftsdomänen betrachten und untersuchen. Dadurch kann von einer umfassenden Sicht auf die Domäne der Instandhaltungssysteme und deren Anforderungen ausgegangen werden. Für eine umfassende Sicht auf die Instandhaltungssysteme und die daraus resultierenden Anforderungen können die Ergebnisse als Grundlage genutzt werden.

4.2.2.1 Auflistung der gefundenen Anforderungen an Instandhaltungssysteme

Aufbauend auf den Ergebnissen der Literaturrecherche werden aus diesen Anforderungen an die Instandhaltungssysteme extrahiert. Zur Extraktion der Anforderungen aus der Literatur werden folgende Schritte durchgeführt:

- Zuerst werden logisch nachvollziehbare Anforderungen aus allen Beiträgen einzeln in ihrer jeweiligen Detaillierungsstufe extrahiert und gesammelt.
- Im nächsten Schritt werden Anforderungen von Beiträgen, die bestimmte bereits aufgenommene Anforderungen voraussetzen, aber diese bisher nicht erwähnen, so behandelt, als ob diese auch die Anforderungen nennen.
- Anschließend werden gleiche und ähnliche Anforderungen zusammengeführt.
- Gruppen von ähnlichen Anforderungen werden übergreifenden Kategorien zugeordnet.
- Die Kategorisierung wird iterativ weiterentwickelt und Anforderungen werden zu allgemeineren Anforderungen aggregiert, um ein einheitliches Abstraktionsniveau von sehr ähnlichen Anforderungen zu erreichen.

Nachfolgend werden die identifizierten Kategorien mit den jeweiligen Anforderungen in mehreren Tabellen dargestellt und diskutiert. Die Anforderungen sind mit einer eindeutigen Nummerierung versehen, die in den Spalten mit der Bezeichnung **LNr.** eingetragen sind. Diese Abkürzung steht für Literaturnummer, welche als Bezeichnung für die Nummerierung für Literaturanforderungen benutzt wird, um diese von den Praxisanforderungen und deren Nummerierung zu unterscheiden. Daneben ist die Benennung der jeweiligen Anforderungen in der zweiten Spalte eingetragen und die jeweilige Anzahl (**Anz.**) der Nennung in der dritten Spalte. Anforderungen sind Fall aufgrund der Aggregation nicht zwangsläufig atomar.

Technischer Kundendienst und Teleservice		
LNr.	Anforderung	Anz.
1	Anzeige d. Information über Kundenbeziehung inklusive Vertragsdaten, Serviceinformationen und Maßnahmenverlauf-/Historie	8
2	Kundenverwaltung (Anlegen, Pflege d. Daten)	8
3	Möglichkeit Experten zur Lösung per Teleservice heranzuziehen/optional Bild und Audio	7
4	Interface für Kunden um Service beim Hersteller/Serviceanbieter anzumelden	3
5	Interface für Kunden mit Beschwerdefunktion	3
6	Einbindung ext. Dienstleister Instandhaltung (Informationsweitergabe, Kommunikation, Eskalation)	2
7	Automatische Erstellung von Leistungsberichten und Serviceprotokollen nach Beendigung der Tätigkeit beim Kunden	1
8	(Automatische) Abrechnungsmöglichkeit/Rechnungstellung nach Beendigung der Tätigkeit	1
9	Auswahlmöglichkeit potentieller externen Dienstleister bei Auftrag	1

Tabelle 6: Anforderungen Instandhaltungssystem – Technischer Kundendienst und Teleservice

Quelle: Eigene Darstellung

Die Anforderungen im Bereich des technischen Kundendienstes und des Teleservice richten sich an ein Instandhaltungssystem, das in der Lage sein muss, entweder aus Serviceanbietersicht Kunden extern einzubinden oder aus Servicenehmersicht Dienstleister für

die Instandhaltung in das interne System einzubinden und zu verwalten. Hierbei ist vor allem eine Kundenverwaltung oder Anbieterverwaltung als Grundlage wichtig, um die restlichen Anforderungen erfüllen zu können. Weiterhin sind Anforderungen im Bereich der Informationsweitergabe und (automatischen) Erstellung von Leistungsberichten und Rechnungen häufig. Diese Anforderungen spielen nur bei Publikationen eine Rolle, bei denen Betrieb und Instandhaltung von unterschiedlichen Unternehmen durchgeführt werden. Wohingegen die Anforderungen, einen Teleservice Experten durch das System hinzuzuziehen zu können (Anforderung LNr. 3), extern wie auch intern relevant ist.

Remote Maintenance		
LNr.	Anforderung	Anz.
10	Remote Maintenance/Fokus Fernüberwachung	22
11	Fernzugriff (Ändern der Einstellungen, Steuerung von Komponenten)	12
12	Audioübertragung der Maschine	1
13	Bildübertragung der Maschine	1

Tabelle 7: Anforderungen Instandhaltungssystem - Remote Maintenance

Quelle: Eigene Darstellung

Anforderungen aus der Kategorie Remote Maintenance beziehen sich auf den Fernzugriff auf Anlagen, Maschinen und Komponenten. Es wird zwischen dem Auslesen der Daten unterschieden und dem direkten Zugriff auf die jeweilige Anlage, Maschine oder Komponente und dem Ändern der jeweiligen Einstellungen. Beide Aspekte werden separiert, da die Implementierung des Fernzugriffs deutlich anspruchsvoller hinsichtlich Sicherheitsanforderungen ist. Weiterhin werden Übertragungsmöglichkeiten per Audio und Bild als Anforderungen genannt. Eng verbunden sind diese Aspekte mit dem Condition Monitoring, das jedoch nicht automatisch einen Fernzugriff ermöglicht.

Die Kategorie Auftragsmanagement (dargestellt in Tabelle 8) bündelt Anforderungen, die sich mit einer Auftragsverwaltung, Auftragshandhabung, der Darstellung von Aufträgen für Mitarbeiter sowie der Informationsanreicherung von Aufträgen befassen. Anhand der Anzahl der Nennungen wird ersichtlich, dass das Auftragsmanagement bzw. die Auftragsverwaltung den essentiellen Bestandteil eines Instandhaltungssystems darstellt. Hierunter fällt zum einen die schlichte Auftragshandhabung, sodass bspw. Aufträge verfügbar sind und durch das System verarbeitet werden können. Zum anderen die Verwaltung, wozu das Anlegen, Bearbeiten, Löschen, Anreichern mit Informationen etc. von Aufträgen, gehört.

Auftragsmanagement und -informationen		
LNr.	Anforderung	Anz.
14	Auftragsmanagement	27
15	Automatisiertes Erstellen von Aufträgen durch bestimmte Ereignisse (CM, Fehler, Wartung)	7
16	Priorisierung von Instandhaltungsaufgaben	5
17	Abfrage des Auftragsstatus	4
18	Management von ungeplanten Serviceaufträgen (kurzfristiges Einplanen von Aufträgen)	2

Auftragsmanagement und -informationen		
LNr.	Anforderung	Anz.
Auftragsmanagement/UI/Mitarbeiterfokus		
19	Zuweisen, Ablehnen und Bestätigen von Aufträgen möglich	5
20	Darstellung Aufträge pro Instandhalter	2
21	Erinnerungsfunktion über ausstehende Aufgaben	2
22	Unterbrechung von Aufträgen möglich	1
23	Eskalationsmanagement (Möglichkeit Aufträge auf höhere Stufe zu eskalieren, ggfs. mit angebundener Kommunikation und Zuordnung zu einer Person)	1
Auftragsinformationen		
24	Zeitvorhersagen für Aufträge und damit verbundenen Instandhaltungsaufgaben	2
25	Ausführliche Auftragsbeschreibung	2
26	Benötigte Skills, rechtliche Regularien für Auftrag enthalten	3
27	Bewertung von einzelnen Aufgaben und Aufträgen nach Schwierigkeitsgrad	2
28	Fotos an Auftragsinformationen anhängbar	1

Tabelle 8: Anforderungen Instandhaltungssystem – Auftragsmanagement und Auftragsinformationen

Quelle: Eigene Darstellung

Daneben wird das automatisierte Erstellen von Aufträgen durch Ereignisse wie Fehlervorhersage aus dem Condition Monitoring sowie die Priorisierung und die Abfrage der Auftragsstatus häufig genannt. Neben diesen Anforderungen werden zum einen Anforderungen identifiziert, die Auftragsmanagementfunktionen mit Mitarbeiterfokus besitzen, wie die Darstellung der Aufträge pro Instandhalter oder eine Erinnerungsfunktion für kommende Aufgaben. Zum anderen sind Anforderungen häufig zu finden, die sich mit der Informationsanreicherung von Aufträgen bzw. Auftragsdaten befassen, wie bspw. die Zeitvorhersage von Aufträgen, eine genaue Beschreibung und benötigte Skills zur Lösung des Auftrages sowie rechtliche Regularien etc.

Wartungsmanagement		
LNr.	Anforderung	Anz.
29	Wartungspläne für Anlagen bereitstellen	16
30	Manuelles Festlegen der Wartungsgrenzen	4
31	Wartungsdokumentation	2

Tabelle 9: Anforderungen Instandhaltungssystem – Wartungsmanagement

Quelle: Eigene Darstellung

Die Anforderungen des Wartungsmanagements stellen sicher, dass Wartungspläne für Anlagen bzw. Komponenten im System vorhanden sind (um ggfs. eine automatische Auftragserstellung auslösen zu können). Die Anforderungen manuelles Festlegen der Wartungsgrenzen ermöglicht neben der Wartungsplanerstellung, dass Grenzwerte festgelegt werden sowie ab wann Wartungen bei bestimmten Komponenten und Maschinen fällig sind. Dies kann beispielweise im Zusammenspiel mit den Wartungsplänen zeitlich oder nach der Abnutzung in Kombination

mit dem Condition Monitoring geschehen. Die jeweiligen Auslöser bzw. Intervalle sollen hiermit eingestellt werden. Schließlich wird die Anforderung erwähnt, dass eine (automatische) Wartungsdokumentation nach getätigter Wartung durch das System unterstützt wird.

Instandhaltungsplanung, -optimierung und Risikoklassifikation		
LNr.	Anforderung	Anz.
32	Optimierung der Instandhaltung hinsichtlich, Aufträgen, Inspektionen, Zeiten (Window of Opportunity) etc.	13
33	Intelligente Disposition von Mitarbeitern (Standorte, Fähigkeiten, Regularien, Prioritäten)	12
34	Möglichkeit eine Disposition manuell zu ändern	1
Risikoklassifikation und RCM		
35	Bewertung und Erfassung von kritischen Risiken, Prioritätszuweisung sowie Klassifikation von Anlagen/Komponenten	10
36	FMEA/FMCEA	7
37	RCM-Unterstützung	4
Kostenmanagement		
38	Kostenmanagement der Instandhaltung (Mitarbeiter, Materialien, Tools, Historie, Planung und Bewertung)	10

Tabelle 10: Anforderungen Instandhaltungssystem – Instandhaltungsplanung, -optimierung und Risikoklassifikation

Quelle: Eigene Darstellung

Anforderungen aus der Kategorie Instandhaltungsplanung, Optimierung und Risikoklassifikation sind übergreifende Funktionen, die bspw. für eine Optimierung der Aufträge, Inspektionen und Wartungen sorgen, sodass niedrige Kosten oder kurze Wege mit eingeplant werden können. Mit dem Begriff Window of Opportunity ist die Einplanung von Instandhaltungsmaßnahmen in produktionsfreie Zeiten oder das Zusammenfassen mehrerer Aufträge zu einem bestimmten Zeitpunkt gemeint. Durch letzteres können bspw. Aufträge für eine Anlage, die an verschiedenen Zeitpunkten durchzuführen sind, zusammengelegt werden, sodass sich Anfahrtszeiten und Stillstandszeiten verkürzen. In den meisten Fällen ist mit der Optimierung eine mathematische Optimierung gemeint, die Aufträge und Maßnahmen nach bestimmten Randbedingungen und Maximierungs- bzw. Minimierungszielen untersucht.

Speziell im Mitarbeitermanagement steht die Anforderung für das intelligente Disponieren von Mitarbeitern, sodass Aufträge entsprechend dem Standort, den Fähigkeiten und den Auftragsprioritäten der Mitarbeiter zugeordnet werden. Neben den Optimierungsfunktionen sind Anforderungen, die sich mit Risiken auseinandersetzen, häufig in der Literatur anzutreffen. Kritische Risiken von bestimmten Fehlern sollen eingetragen werden, wie auch Anlagen, deren Auswahl extrem schwerwiegend ist. Beides spielt für eine gezielte Optimierung eine große Rolle. Daneben wird die Unterstützung von FMEA bzw. FMCEA, sowie generelle RCM-Unterstützung als Anforderung genannt. Diese Anforderungen bzgl. der Risiken bzw. der Risikoklassifikation hängen dabei eng miteinander und der Optimierung zusammen, da sie alle zu einer Priorisierung von Aufträgen, Fehlern und Wartungsintervallen führen können.

Die letzte Unterkategorie betrifft das Kostenmanagement der Instandhaltung. Diese Anforderung enthält die finanzielle Bewertung von Instandhaltungsaufgaben hinsichtlich

Mitarbeiter-, Material- und Werkzeugeinsatz sowie weiteren Faktoren. In der Ausprägung umfasst es neben der historischen Auswertung die Vorhersage von Kosten der Instandhaltung wie auch eine Bewertung bestimmter Aufgaben.

Anlagenübersicht und -informationen		
LNr.	Anforderung	Anz.
39	Übersicht über Anlagen mitsamt Anlageninformation (Standort, Ersatzteile, Funktionalität,..)	24
40	Komponentenübersicht/-bäume der Anlage verfügbar	13
41	Technische Dokumente für Anlagen verfügbar	10
42	Zuweisung von Instandhaltungsunternehmen bzw. Instandhalter zu Anlage mitsamt Kontaktdaten	7
43	Anzeige von Serviceverträgen für Anlagen	4
44	GPS Koordinaten der Anlagen	4
45	Darstellung der GPS Informationen auf Karte	2
46	Übersicht über Fehler/Störungen/Aufträge/Maßnahmen zu einer Anlage	3
47	Bereitstellung von Informationen zur Selbstwartung/Self-Service für Kunden	2
48	CAD-Zeichnungen für Anlagen verfügbar	1
49	Optische Identifizierung Anlagen (AR)	1
50	Softwarestand von Komponenten/Anlagen anzeigen	1

Tabelle 11: Anforderungen Instandhaltungssystem – Anlagenübersicht und -informationen

Quelle: Eigene Darstellung

Die häufigsten Anforderungen zur Anlagenübersicht und mit Anlagen verbundenen Informationen sind das Bereitstellen einer Übersicht vorhandener Anlagen, die verbunden ist mit der Bereitstellung von Information über deren Ausrüstung, Lokation, benötigte Ersatzteile, deren Funktionalität und verbundene ökonomische Informationen. Das Sortieren und Filtern dieser Übersicht soll ermöglicht werden, wie auch das Anzeigen von Produktionsaufträgen und des aktuellen Status der Anlage. Daneben sollen für die jeweiligen Anlagen Komponentenübersichten sowie die dazugehörigen technischen Dokumente im System verfügbar sein. Hierbei wird nicht explizit eine Bereitstellung eines sogenannten digitalen Zwillings (Kassner et al., 2017, S. 1676) als Anforderung genannt, jedoch kann dies eine Form der Erfüllung der Anforderung darstellen.

Weitere Anforderungen umfassen die Möglichkeit, Ansprechpartner bzw. Instandhaltungsunternehmen zu einer Anlage mitsamt Kontaktdaten als Information hinzuzufügen sowie Serviceverträge für eine Anlage zu hinterlegen und anzuzeigen. Weiterhin soll der Ort in Form von GPS-Koordinaten der Anlage und die Darstellung auf einer Karte per GPS erfolgen, Fehler einer Anlage übersichtlich dargestellt werden, Informationen für den Self-Service bereitgestellt werden und Versionen bestimmter Softwareapplikationen einer Anlage abrufbar sein.

Ersatzteilmanagement		
Nr.	Anforderung	Anz.
51	Ersatzteilmanagement (Ansicht Lagerort, Verfügbarkeit, Kosten)	18
52	Ersatzteil-, Material- und ggfs. Werkzeuglisten pro Auftrag verfügbar	12
53	Ersatzteilbestellung	7
54	Forecast von benötigten Ersatzteilen und Vergleich Lager	5
55	Verfügbarkeit bei Zulieferern anzeigen	2
56	Materialverbrauch und Ersatzteilverbrauch erfassen nach Auftrag	1
57	Plausibilitätschecks bei Ersatzteilbestellung	1

Tabelle 12: Anforderungen Instandhaltungssystem – Ersatzteilmanagement

Quelle: Eigene Darstellung

Ersatzteilmanagement spielt bei Instandhaltungssystemen eine große Rolle, denn diese Anforderung wird bei unterschiedlichen Instandhaltungssystemen 18-mal genannt. Dies soll vordergründig Informationen über Ersatzteile, deren Lagerort, Verfügbarkeit und ggfs. deren Kosten bereitstellen. Die nächsthäufige Anforderung betrifft die Verbindung der Ersatzteile mit den Aufträgen, sodass vorab bekannt ist, welche Aufträge welche Ersatzteile erfordern und welche bei bestimmten Aufträgen verbraucht worden sind. Daneben gehen einige Autoren auch explizit auf die Bestellmöglichkeit von Ersatzteilen durch das System und den Forecast von benötigten Ersatzteilen gegenüber der Verfügbarkeit ein. Dabei kann zwischen der Verfügbarkeit im eigenen Lager oder bei einem Zulieferer unterschieden werden. Weiter werden Plausibilitätschecks bei Bestellungen sowie das Erfassen des Materialverbrauchs als Anforderung genannt und unter dem Begriff des Ersatzteilmanagements zusammengefasst.

Mitarbeitermanagement		
LNr.	Anforderung	Anz.
58	Anzeige einer Übersicht über interne und externe Kompetenzträger für die Instandhaltung mitsamt Personaldaten	12
59	Arbeitszeitmessung	4
60	Verfügbarkeit mit Kontakt anzeigen	3
61	Reisekostenerfassung	2
62	Skills des Personals anzeigen	2
63	Trainingsprogramme für Instandhalter (z. B. Lernen des Interpretierens von CM-Daten, Systemfunktionen)	2
64	Ort des Personals anzeigen (GPS)	2
65	Automatisches Erkennen des verfügbaren Personals mit entsprechenden Fähigkeiten für bestimmten Auftrag	1
66	Evaluationsmöglichkeit der Mitarbeiter (Berechnung von Leistungen, Kontrolle, Bewertungsschema)	1

Tabelle 13: Anforderungen Instandhaltungssystem – Mitarbeitermanagement

Quelle: Eigene Darstellung

Im Bereich des Mitarbeitermanagements ist die Übersicht von internen und externen Kompetenzträgern für die Instandhaltung bei 12 Artikeln eine wichtige Anforderung. Diese

Anforderung bildet die Bedingung für die meisten anderen Anforderungen in dieser Kategorie, da durch diese eine Mitarbeiterverwaltung eingeführt wird. In diese Kategorie fallen Anforderungen zur Arbeitsbewertung wie die Arbeitszeitmessung und die Evaluationsmöglichkeit der Mitarbeiter. Zusätzlich wird die Anzeige der Verfügbarkeit von Instandhaltern und die Anzeige des Ortes als Anforderungen genannt sowie auch die Reisekostenerfassung und das Bereitstellen von Trainingsprogrammen durch das System. Ein Beitrag nennt explizit das automatische Erkennen des verfügbaren Personals mit nötigen Fähigkeiten bei einem bestimmten Auftrag als Anforderung.

Kommunikation		
LNr.	Anforderung	Anz.
67	Kommunikation und Kontaktaufnahme zu anderen Mitarbeitern/Personen (E-Mail, SMS, Nachrichten, Telefon)	14
68	Nachrichtenservice	4
69	Videofunktionalität	3
70	Social Web 2.0 Fähigkeiten, Kommentieren, Bewerten, diskutieren	2
71	Feedback Funktion für Informationen und Verbesserungsvorschläge	2
72	Blog	1

Tabelle 14: Anforderungen Instandhaltungssystem – Kommunikation

Quelle: Eigene Darstellung

In die Kategorie Kommunikation fallen Anforderungen, welche die Kommunikation z. B. per E-Mail, SMS; Messaging und Telefon sicherstellen, aber auch Social Web Funktionalitäten wie das Kommentieren, Bloggen und Feedbackfunktionen bereitstellen. Zusätzlich fallen unter diese Kategorie ein allgemeiner Nachrichtenservice und eine Videofunktionalität, die das Aufnehmen und das Abspielen von digitalen Videos ermöglicht. Die mit Abstand am häufigsten genannte Anforderung betrifft dabei die allgemeine Kommunikation und die Kontaktaufnahme zu anderen Mitarbeitern und Personen durch das Instandhaltungssystem, die auch eine notwendige Bedingung für den Teleservice darstellt.

Mobiles System		
LNr.	Anforderung	Anz.
73	Barcode Unterstützung (Identifikation Maschinen/Material)	6
74	Mobiler Zugriff auf Instandhaltungsinformationen	5
75	Diagnosefunktion für Instandhaltungsobjekte/dezentrales Abgreifen von Daten/Wireless Zugriff auf Daten z. B. per Nahnetzwerk	5
76	Arbeiten im Offlinemodus	2
77	Fotografiefunktion	2
78	Signierfunktion	2
79	GPS Routenplanung	1
80	Benachrichtigung Kunde über Eintreffen Mitarbeiter	1
81	Broadcast Multicastfunktion	1

Mobiles System		
LNr.	Anforderung	Anz.
82	Direkt die Anzeige drucken	1
83	Proaktives Bereitstellen von Informationen zu Auftrag	1

Tabelle 15: Anforderungen Instandhaltungssystem – Mobiles System

Quelle: Eigene Darstellung

Die Anforderungen im Bereich mobiles System beschäftigen sich mit Anforderungen, die bei der Benutzung von mobilen Endgeräten im Instandhaltungskontext als wichtig eingeschätzt werden. Die häufigste Anforderung stellt die Barcode Unterstützung dar, z. B. verbunden mit direktem Zugriff auf die Informationen einer Anlage oder eines Bauteils. Weiter stellen der mobile Zugriff auf Instandhaltungsinformationen, die dezentrale Diagnosefunktion und das Arbeiten im Offlinemodus mehrfach genannte Anforderungen dar. Diese Anforderungen sind ausschließlich bei MSS vorzufinden.

(Historische) Auswertungen		
LNr.	Anforderung	Anz.
84	Fehlerhistorien	27
85	Informationen über Maßnahmen- und Werkzeughistorie von Anlagen	22
86	Zustandshistorie von Teilen/Anlagen	19
Auswertungen		
87	Performancebewertung und -überprüfung der Anlagen (OEE, Stopps etc.)	15
88	Fehlerauswertung	9
89	Fehlermeldung mit ausführlicher Beschreibung	8

Tabelle 16: Anforderungen Instandhaltungssystem – (Historische) Auswertungen

Quelle: Eigene Darstellung

Sehr häufig werden Anforderungen bzgl. der Bereitstellung und Archivierung von historischen Daten und deren Auswertung in der Literatur behandelt. Bei den historischen Daten wird zwischen Fehlerhistorien, anlagenspezifische Maßnahmen- und Werkzeughistorien und Zustandshistorien unterschieden. Auswertungen sollen möglichst bzgl. Fehlern, Performancemessung der Instandhaltung und der Anlagen möglich sein.

Fehlermanagement		
LNr.	Anforderung	Anz.
90	Fehlermeldung mit ausführlicher Beschreibung	12
91	Fehlermeldungen werden direkt auf die mobilen Endgeräte der zuständigen Personen gesendet	8
92	Fehlermeldungen mit Ursache-Wirkungsbeziehung	6
93	Möglichkeit Fehler manuell zu melden	4
94	Fehlerpriorisierung	4

Fehlermanagement		
LNr.	Anforderung	Anz.
95	Fehlerlokalisierung (grafisch)	1

Tabelle 17: Anforderungen Instandhaltungssystem – Fehlermanagement

Quelle: Eigene Darstellung

Fehlermanagement ist für ein Instandhaltungssystem ein vordergründiges Ziel. In der Literatur ist in diesem Kontext vor allem die Informationsgüte der Fehlermeldungen in vielen Beiträgen wichtig. Genaue Beschreibungen und Ursache-Wirkungsbeschreibungen, die mit dem jeweiligen Fehler verbunden sind, sollen helfen, dass der Instandhalter schneller die Fehler beheben kann und werden dementsprechend als Anforderungen genannt. Weiterhin sollen Fehlermeldungen direkt auf die mobilen Endgeräte der jeweilig zuständigen Personen gesendet werden und es sollte Möglichkeiten geben, Fehler manuell zu melden, diese zu priorisieren und grafisch zu lokalisieren.

Condition Monitoring und Predictive Maintenance		
LNr.	Anforderung	Anz.
96	Condition Monitoring	31
97	Predictive Maintenance	18
98	Darstellung Anlagenzustand in Dashboard	13
99	Zuordnung von Mustern zu Fehlern und Wartungen	2
100	Taggen von Fehlern und Beobachtungen (Data Mining)	1

Tabelle 18: Anforderungen Instandhaltungssystem – Condition Monitoring und Predictive Maintenance

Quelle: Eigene Darstellung

In mehr als der Hälfte der untersuchten Literatur wird Condition Monitoring als Anforderung genannt. Zusätzlich wird die visuelle Darstellung des Anlagenzustands und damit der Condition Monitoring Daten 13-Mal genannt. Predictive Maintenance wird hingegen 18-Mal als Anforderung erwähnt, jedoch wird hier oftmals nicht klar zwischen Condition Monitoring und Predictive Maintenance unterschieden, sodass die Übergänge teilweise fließend sind. 13 dieser Publikationen zu Predictive Maintenance enthalten als Teil des Predictive Maintenance die Berechnung einer Prognose zur Zustandsentwicklung von Komponenten (z. B. per Remaining Useful Life o. Ä). Nur wenige benutzen Techniken des Data Mining. Zur besseren Ausführung von Predictive Maintenance wird die Möglichkeit der Zuordnung von Mustern zu Fehlern und Wartungsfällen in zwei Publikationen als Anforderung oder dieses Feature im Kontext von CBR genannt, um ähnliche Fehler zu erkennen. Eine Publikation nannte als Anforderung die Möglichkeit, Fehler und Beobachtungen zu taggen, damit Algorithmen dadurch einfacher angelernt werden können.

Dokumentenmanagement		
LNr.	Anforderung	Anz.
101	Dokumentenmanagement (Speichern von Dokumenten, Freigabe/Teilen von Dokumenten, Bereitstellung von Dokumenten)	15
102	Teilautomatische Dokumentation abgeschlossener Aufträge/Fehler und Reports	11
103	Ausfüllassistent für Formulare und Reports	4
104	Traceability aller Interaktionen und Aktionen in System	1

Tabelle 19: Anforderungen Instandhaltungssystem – Dokumentenmanagement

Quelle: Eigene Darstellung

Dokumentenmanagement, Protokollierung von Interaktionen und Funktionen zur Erleichterung der Dokumentation werden in der Literatur im Kontext des allgemeinen Dokumentenmanagements genannt. Erleichterungen in diesen Bereichen können für die Instandhalter zu enormer Reduktion von oft als negativ angesehenen Arbeitsaspekten führen. Dabei spielt insbesondere die teilautomatische Dokumentation abgeschlossener Aufträge/Fehler und Reports als am häufigsten genannte Anforderung der Kategorie eine große Rolle, da sie als notwendige Bedingung für die Anforderungen der Wartungsdokumentation sowie der Anforderung der automatischen Erstellung von Serviceprotokollen gilt. Die beiden letzteren sind dabei Spezialisierungen der hier vorliegenden Anforderung.

Wissensmanagement		
LNr.	Anforderung	Anz.
105	Wissensmanagement	21
106	Informationen zum aktuellen Instandhaltungsfall automatisch verknüpfen (CBR, Wahrscheinlichkeiten etc.)	10
107	Ontologie/Taxonomie für semantische Wissensspeicherung	7
108	Suche und Aufrufen von strukturierten und unstrukturierten Daten	3
109	Bestätigung einer passenden Lösung fürs das spätere Lernen	2
110	Wiki	2
111	Informationsportal	1
112	Taggen von Wissen und Informationen (Verschlagwortung)	1

Tabelle 20: Anforderungen Instandhaltungssystem – Wissensmanagement

Quelle: Eigene Darstellung

In vielen der untersuchten Beiträge spielt Wissensmanagement als Teil eines Instandhaltungssystems eine sehr wichtige Rolle. Entsprechend wird diese Anforderung 21-mal genannt. Insbesondere sollen Informationen zu einem Auftrag oder Instandhaltungsfall verknüpft werden können. Dies kann manuell oder analytisch geschehen, indem ähnliche Fehler oder Lösungsansätze verglichen werden. Weiterhin werden Ontologien oder Taxonomien zur semantischen Wissensspeicherung als notwendig angesehen, um bspw. kontextsensitives Wissen zur Verfügung zu stellen und zu speichern. Das Auffinden von Daten wird so erleichtert, jedoch soll dies zusätzlich durch eine Suche bei drei Beiträgen, eines Wikis (zwei Beiträge) sowie eines Informationsportals (1 Beitrag) und dem Taggen und Bestätigen von

Wissen weiter vereinfacht werden. All diese Anforderungen können in einem Instandhaltungssystem einen Beitrag zur Wissensarchivierung und -bereitstellung leisten.

Handlungsempfehlungen		
LNr.	Anforderung	Anz.
113	Handlungsempfehlungen für Fehler	11
114	Handlungsempfehlungen für Fehlerdiagnose	7
115	Feedback für jeden Schritt der Handlungsempfehlung möglich	1
116	Sicherheitsinstruktionen	1
117	Feedback über Erfolg der Reparatur	1

Tabelle 21: Anforderungen Instandhaltungssystem – Handlungsempfehlungen

Quelle: Eigene Darstellung

Ein weiterer wichtiger Teilaspekt des Wissensmanagements betrifft Handlungsempfehlungen, die per System den Instandhaltern Schritte und Möglichkeiten aufzeigen, wie Fehlerbehebungen oder bestimmte Arbeitsvorgänge durchzuführen sind. Diese Handlungsempfehlungen werden bei Fehlern und bei Fehlerdiagnosen häufig in der Literatur erwähnt. Sie ermöglichen es, gezielt Wissen zu archivieren und dem Instandhalter in einer einfach abzuarbeitenden und zu protokollierenden Art bereitzustellen.

Externe Dienste		
LNr.	Anforderung	Anz.
118	Anbindung beliebiger externer Datenquellen	2
119	Strombörsen implementieren	2
120	Wetterprognosen einbinden	2
121	Terminmanagement und Koordination (mit externen Anbietern)	1
122	Feedbackfunktion für Anlagenhersteller zum Übermitteln von Maschinendaten	1
123	Ableiten von dispositiven Daten zur Verbesserung der Produktion	1
124	Erzeugung von Informationen für die Produktentwicklung	1

Tabelle 22: Anforderungen Instandhaltungssystem – Externe Dienste

Quelle: Eigene Darstellung

Sehr selten werden Anforderungen aus den Kategorien der externen Dienste genannt, die insbesondere zur Einbindung spezifischer externer Quellen dienen. Hierzu gehören Strombörsen, Wetterdienste oder Produktionssysteme. Daneben wird die Anforderung für die Bereitstellung von Daten für die Anlagenhersteller, für die Produktion und für die Produktentwicklung genannt.

Systemadministration		
LNr.	Anforderung	Anz.
125	Funktionen zur Systemadministration (Nutzermanagement, Rollenmodell)	7
126	Unterstützung von Webtechnologien/Zugang über Browser	5
127	Authentifizierung	3

Systemadministration		
LNr.	Anforderung	Anz.
128	Verschlüsselung der Daten zw. Systemen	2
129	Import Export von Stammdaten (Anlagendaten, Vertragsdaten etc.) und weiteren relevanten Daten	2
130	Unterstützung mehrerer Sprachen	2
131	Zusammenführung und Aufbereitung heterogener Daten- und (Produkt-)Informationsbestände	2
132	Unternehmensübergreifend verwendbar	1
133	Remote Softwareupdates	1
134	Berechnung von Daten in der Cloud – Massive Learning	1
135	Nutzung offener Standards des Systems	1

Tabelle 23: Anforderungen Instandhaltungssystem – Anforderungen Systemadministration

Quelle: Eigene Darstellung

Allgemeine Anforderungen an das Instandhaltungssystem, die vor allem den technischen Hintergrund betreffen, werden in dieser Kategorie aggregiert. Hierunter fallen allgemeine Funktionen zur Systemadministration, wie ein Nutzermanagement und Rollenmodell oder die Unterstützung von Webtechnologien, die bspw. den Zugang über den Browser ermöglichen. Sicherheitsanforderungen wie eine Authentifizierung und eine Verschlüsselung der Daten zw. den Systemen bzw. ihren Subsystemen oder Anforderungen, die das Einbinden von fremden Systemen und Daten betreffen, werden weiterhin genannt.

4.2.2.2 Zusammenfassung der häufigsten Anforderungen

Insgesamt können 28 Anforderungen identifiziert werden, welche bei mindestens 10 Beiträgen genannt werden. Dargestellt sind die Anforderungen in Tabelle 24. Dort sind die Anforderungen nach Häufigkeit sortiert dargestellt, jeweils mit ihrer Nummer, Bezeichnung und der jeweiligen numerischen Häufigkeit. Unter diesen 28 Anforderungen sind aus fast jeder vorgestellten Kategorie Anforderungen vertreten. Dabei dominieren Anforderungen, welche die Zustandsbewertung von Anlagen, den Fernzugriff, das Wissensmanagement sowie das Auftragsmanagement betreffen. Diese 28 Anforderungen decken insgesamt 450 Nennungen von Anforderungen ab. Über alle 135 Anforderungen existieren 751 Nennungen. Daraus folgt, dass diese Anforderungen fast 60 % aller Nennungen auf sich vereinen. Oftmals stellen die Anforderungen demgemäß Anforderungen dar, die bei den meisten Systemen erfüllt sein müssen.

Daher stellen die gewonnenen Erkenntnisse eine optimale Grundlage dar, um generische Anforderungen für eine Referenzarchitektur abzuleiten, da bei den großen Überschneidungen der Anforderungen über die jeweiligen Systeme hinweg von einem generischen Kern gesprochen werden kann. Dabei ist die Relevanz der Anforderungen abnehmend mit ihrer Nennung. Die gewonnenen Erkenntnisse können daher genutzt werden, um eine Anforderungsaufnahme bei vier Unternehmen durchzuführen und deren Ergebnisse zu vergleichen. So können die Anforderungen aus der Literatur und der Praxis miteinander verglichen werden.

LNr.	Anforderung	Anz.
96	Condition Monitoring	31
14	Auftragsmanagement	27
84	Fehlerhistorien	27
39	Übersicht über Anlagen mitsamt Anlageninformation (Standort, Ersatzteile, Funktionalität,..)	24
10	Remote Maintenance/Fokus Fernüberwachung	22
85	Informationen über Maßnahmen- und Werkzeughistorie von Anlagen	22
105	Wissensmanagement	21
86	Zustandshistorie von Teilen/Anlagen	19
51	Ersatzteilmanagement (Ansicht Lagerort, Verfügbarkeit, Kosten)	18
97	Predictive Maintenance	18
29	Wartungspläne für Anlagen bereitstellen	16
87	Performancebewertung und -überprüfung der Anlagen (OEE, Stopps etc.)	15
101	Dokumentenmanagement (Speichern von Dokumenten, Freigabe/Teilen von Dokumenten, Bereitstellung von Dokumenten)	15
67	Kommunikation und Kontaktaufnahme zu anderen Mitarbeitern/Personen (E-Mail, SMS, Nachrichten, Telefon)	14
32	Optimierung der Instandhaltung hinsichtlich Aufträgen, Inspektionen, Zeiten (Window of Opportunity) etc.	13
40	Komponentenübersicht/-bäume der Anlage verfügbar	13
98	Darstellung Anlagenzustand in Dashboard	13
11	Fernzugriff (Ändern der Einstellungen, Steuerung von Komponenten)	12
33	Intelligente Disposition von Mitarbeitern (Standorte, Fähigkeiten, Regularien, Prioritäten)	12
52	Ersatzteil-, Material- und ggfs. Werkzeuglisten pro Auftrag verfügbar	12
58	Anzeige einer Übersicht über interne und externe Kompetenzträger für die Instandhaltung mitsamt Personaldaten	12
90	Fehlermeldung mit ausführlicher Beschreibung	12
102	Teilautomatische Dokumentation abgeschlossener Aufträge/Fehler und Reports	11
113	Handlungsempfehlungen für Fehler	11
35	Bewertung und Erfassung von kritischen Risiken, Prioritätszuweisung sowie Klassifikation von Anlagen/Komponenten	10
38	Kostenmanagement der Instandhaltung (Mitarbeiter, Materialien, Tools, Historie, Planung und Bewertung)	10
41	Technische Dokumente für Anlagen verfügbar	10
106	Informationen zum aktuellen Instandhaltungsfall automatisch verknüpfen (CBR, Wahrscheinlichkeiten etc.)	10
Anzahl Anforderungen mit mindestens 10 Nennungen		28

Tabelle 24: Anforderungen an ein Instandhaltungssystem – Übersicht der häufigsten Anforderungen (Angabe ab 10 Nennungen)

Quelle: Eigene Darstellung

4.3 Praxisanforderungen an ein integriertes Instandhaltungssystem

Neben dem theoretischen Hintergrund der Literaturanalyse und den daraus extrahierten Anforderungen soll der Kern einer Referenzarchitektur aus Anforderungen bestehen, die direkt aus der Praxis extrahiert werden. Dies dient dazu, die Praxisrelevanz der Referenzarchitektur sicherzustellen. Um generalisierbare Anforderungen abzuleiten, bedarf es neben der Übersicht aus der Literatur auch einer relevanten Menge an Anforderungen, die für spezifische Einzelarchitekturen aufgenommen werden (Reidt et al., 2017a, S. 28). Die Unternehmen und deren Anforderungen, aus denen sich die spezifischen Einzelarchitekturen ableiten lassen, sollen zusammen betrachtet einen umfassenden und nicht spezifischen Blick auf die Domäne bzw. das zu entwickelnde integrierte System ermöglichen. Für die RAI werden daher vier Unternehmen aus unterschiedlichen Branchen und mit unterschiedlichen Betriebsmodellen bzgl. des Themas der Instandhaltung untersucht. Hierdurch wird es möglich, aus den einzelnen, spezifischen Anforderungen auf potentielle spezifische Architekturen der einzelnen Instandhaltungssysteme der Unternehmen zu schließen. Im Folgenden werden der Prozess der Anforderungsaufnahme und die untersuchten Unternehmen vorgestellt. Anschließend werden die Anforderungen dargelegt, auf denen die RAI fußt.

4.3.1 Vorgehensweise der Anforderungserhebung

Die systematische Anforderungserhebung wurde in Anlehnung an Aspekte der Anforderungserhebung von Pohl (2010) durchgeführt. Zusätzlich erstreckt sich die Anforderungserhebung auf die Laufzeit von ungefähr zwei Jahren, da diese iterativ im Rahmen des Projektes S-CPS durchgeführt wurde. Diese iterative Vorgehensweise wurde in folgenden Schritten durchlaufen.

- Zuerst wurde der aktuelle Stand der Instandhaltung bei den einzelnen Unternehmen aufgenommen inklusive Instandhaltungsprozessen, beteiligter Systeme und wie diese zur Unterstützung der Instandhaltung herangezogen werden.
- Im zweiten Schritt füllten die Unternehmen basierend auf dem aktuellen Stand der Instandhaltung ein Lastenheft für ein IIS aus, welches den Instandhalter mobil, dezentral und zentral unterstützt. In diesem Lastenheft wurden von den Unternehmen selbst Anforderungen und Use Cases nach Vorlage von Cockburn (2002) aufgenommen. Diese Anforderungen beinhalten zudem eine Priorisierung. Mit der Anforderungsspezifizierung durch die Unternehmen kann ein Einfluss durch vorgegebene Anforderungen aus der Literatur vermieden werden.
- Im Laufe der Entwicklung der Systeme wurden die Bedürfnisse der Unternehmen deutlich klarer. Daher wurden aufbauend auf den aufgenommenen Anforderungen aus dem Lastenheft und den Anforderungen aus der Literatur zusätzlich nach 1,5 Jahren iterativer Entwicklung die Anforderungen anhand eines Fragebogens persönlich abgefragt (der Fragebogen ist in Tabelle 41 im Anhang dargestellt). Die Priorisierung der dort abgefragten Anforderungen richtet sich nach der numerischen Priorisierung von Karlsson et al. (1998).
- Anschließend wurden die Anforderungen aggregiert und die Ergebnisse auf Basis des Lastenheftes und der Befragung zusammengeführt.

4.3.2 Die untersuchten Unternehmen

Damit die spezifischen Anforderungen der einzelnen Unternehmen zusammen genommen einen generischen Kern aufweisen, muss eine möglichst große Bandbreite an unterschiedlichen Betriebsformen der Instandhaltung abgedeckt werden. Die betrachteten Unternehmen unterscheiden sich daher grundlegend in einigen Aspekten der Instandhaltung und wurden von Reidt et al. (2017a, S. 7) wie folgt eingeordnet: „Instandhaltung wird bei diesen Unternehmen teils als eigener Service für andere Unternehmen erbracht, teils auch von Dritten getätigt oder vom eigenem Personal für fremdbezogene Maschinen betrieben. Auch sind die Anforderungen an ein System zur Unterstützung der Instandhaltung in einer eigenen Fabrik andere als bei Windrädern, die sich in einem spärlich besiedelten Raum mit großen Distanzen zueinander befinden. Dort stellen u. a. Konnektivität und damit die Verfügbarkeit von Informationen große Herausforderungen dar.“ Aufgrund der unterschiedlichen Branchenzugehörigkeit, Ausprägungen der Instandhaltung der untersuchten Unternehmen sowie umfassenden Anforderungserhebung an ein nicht auf einen bestimmten Bereich eingegrenztes Instandhaltungssystem kann daher ein umfangreiches Bild der Instandhaltung und der Bedürfnisse der Unternehmen gezeichnet werden.

Die angesprochenen Unternehmen und deren Auswahl werden in diesem Abschnitt detaillierter hinsichtlich der Instandhaltung vorgestellt. Dabei basieren die nachfolgenden Aspekte auf den Ausführungen von Reidt et al. (2016a). Ein wichtiges Kriterium für die Auswahl der Unternehmen ist die Verwendung unterschiedlicher Modelle zur Durchführung und Betrieb der Instandhaltung. Dies führt im Stadium der Anforderungsanalyse zu einem umfassenden Blick auf die Domäne der Instandhaltung. Die Unternehmen werden gemäß § 267 I-III Handelsgesetzbuch (HGB) in Größenklassen anhand der Bilanzsumme, den Umsatzerlösen und der Mitarbeiterzahl eingeordnet:

- Unternehmen U1 – stellt einen Automobilzulieferer dar, der in die Kategorie der großen Kapitalgesellschaft fällt.
- Unternehmen U2 – stellt einen Automobilhersteller dar, der in die Kategorie der großen Kapitalgesellschaft fällt.
- Unternehmen U3 – stellt einen Windparkbetreiber dar, der in die Kategorie der kleinen Kleinstkapitalgesellschaft fällt.
- Unternehmen U4 – stellt einen Maschinenhersteller dar, der in die Kategorie der mittelgroßen Kapitalgesellschaft fällt.

Bei der Rolle der Instandhaltung in den vier untersuchten Unternehmen können vier Fälle unterschieden werden. Im ersten Fall, bei Unternehmen U2, wird die Instandhaltung nahezu komplett autark vom produzierenden Unternehmen geleistet, während im zweiten Fall, bei Unternehmen U3, die Instandhaltung nur als Service bezogen wird. Im dritten Fall, bei Unternehmen U4, wird die Instandhaltung für die eigenen verkauften Maschinen als Service angeboten und ist Teil des Geschäftsmodells. Im Rahmen der RAI konzentriert man sich bei diesem Unternehmen auf die externe Serviceerbringung. Die Instandhaltung der eigenen internen Maschinen wird bei diesem Unternehmen nicht betrachtet. Abschließend ist eine Mischform von hauptsächlich eigenem Betrieb mit einer Kombination aus externem Service bei Unternehmen U1 anzutreffen. Zusammengefasst werden diese Aspekte in Tabelle 25. Zusätzlich werden in dieser Tabelle die Rolle und der betrachtete Funktionsbereich dargestellt. Neben den organisatorischen Aspekten enthält die Tabelle die technische Systemstruktur der Instandhaltung der untersuchten Unternehmen, den Ort der Durchführung der Instandhaltung

sowie den momentanen Hauptfokus der Optimierung der Instandhaltung zum Zeitpunkt der Untersuchung. Neben den unterschiedlichen Betriebsarten der Instandhaltung variieren die Orte der Durchführung stark und das Optimierungsziel hinsichtlich Instandhaltung durch die Einführung eines umfassenden Instandhaltungssystems ist ebenso unterschiedlich.

		Anwendungsunternehmen			
		U1	U2	U3	U4
Ist Zustand	Rolle bei der Instandhaltung	Betreiber u. Servicenehmer	Betreiber	Servicenehmer	Serviceanbieter
	Betrachteter Funktionsbereich	Interne und externe Instandhaltung	Interne Instandhaltung	Externe Instandhaltung als Service bezogen	Interne Instandhaltung als Service angeboten
	Systemstruktur	Internes Netzwerk mit SPS, MES, ERP sowie DBs	Internes Netzwerk mit SPS, MES sowie DBs	SPS, MES, DBs mit Daten der Windanlagen	SPS, Industrie-PCs, PDM
	Distanz bis zur Durchführung der Instandhaltung	Am gleichen Standort	Am gleichen Standort	Große Distanz zwischen Standorten	Unterschiedliche externe Standorte mit teils großer Distanz
	Ort der Instandhaltung	Am eigenen Standort	Am eigenen Standort	Unterschiedliche Windanlagen	Unterschiedliche externe Kunden
Soll	Fokus der Optimierung bei Instandhaltung	Wartung und Instandsetzung	Störungserkennung, -beseitigung, -vermeidung	Datenaustausch für Instandsetzung und Wartung	Serviceprozesszeiten und Servicequalität

Tabelle 25: Aspekte der Instandhaltung in den untersuchten Unternehmen
Quelle: Eigene Darstellung

4.3.3 Anforderungen aus der Praxis

4.3.3.1 Spezifische Anforderungen an ein Instandhaltungssystem

Das Ergebnis der gesamten Anforderungserhebung wird nachfolgend vorgestellt. Die spezifischen Anforderungen der jeweiligen Unternehmen sind in der folgenden Tabelle zusammengefasst. Dabei wird die jeweilige Kurzbeschreibung der Anforderungen der Unternehmen ohne Anpassung übernommen, außer einer evtl. nötigen Anonymisierung.

Nr.	Unternehmen U1	Unternehmen U2	Unternehmen U3	Unternehmen U4
1	E-Mail	Kurze prägnante Meldetexte bei Störfall + Fehlercode	Informationsaustausch mit Anlage	Betriebsprotokollierung der Anlage (Störung, Zustandsdaten)
2	Zustandsbezogene Infos der Anlage	Handlungsleitfäden zu Störungen/Editierbar	Mail	Remote Anlagen-/Komponentensteuerung durch Hersteller

3	Auftragsmanagement zusammen mit ERP	Anlagen-, Bauteil- und Maßnahmenhistorie	Internet	Remote Anlagenüberwachung (auch SPS)
4	Barcode scannen	Automat. Doku und Ausw. d. Maßnahmen der Störungsbehebung	Detaillierte Fehlerbeschreibung	Standardschnittstelle und -datenstruktur an Maschine
5	Fehlercodierung, eindeutige Fehlerbeschreibung	Zugriff Herstellerunterlagen	Anzeigen Anlagentemperatur	Optimierung der Auswertung der Dokumentation
6	Anleitung zur Fehlerbehebung	Fehlerdatenbank mit Suchfunktion (Historie)	Anzeige Windrichtung/-stärke	Abgleich von Betriebsdaten (z. B. Protokoll Störfälle & Fehlermeldungen) zwischen Anlagenbetreiber und -hersteller zur zentralen (Cloud-) Speicherung
7	Anzeige Maschinenpläne	Schichtbuch	Anzeige der Wetterbeschaffenheit	Erweiterung der Leitfäden zur Fehlerbehebung
8	Möglichkeit zur Ergänzung/Korrektur von Dokumentationen + Wikis	Automatischer Link zu Leitfäden bei Störung	Dokumentationsmöglichkeit	Unterstützung bei Kommunikation mit Kollegen und Kunden
9	Bild/Videoinformation	Zugriff Fachbereichsinformation	Fehlerkategorisierung	Anzeige Störungsmeldung
10	Anlagen-, Bauteil- und Maßnahmenhistorie	Mobiles Endgerät als Telefonersatz	GPS Genauigkeit für die Kartenversion (Zustand der Anlage)	Nachverfolgung offener Punkte
11	Condition Monitoring	Condition Monitoring	Betriebsprotokollierung	Telefonbücher
12	Wartungspläne und -historie verfügbar	Wartungspläne und -historie verfügbar	3D-Darstellung mit Fehlerlokalisierung	Anlagenplan
13	Anzeige offener und abgeschlossener Aufträge durch das System	Anzeige offener und abgeschlossener Aufträge durch das System	Anlagenplan	Fotografieren
14	OPC UA Schnittstelle zum Empfang bzw. Senden von Maschinendaten	OPC UA Schnittstelle zum Empfang bzw. Senden von Maschinendaten	Anzeige Bestandsituation des Herstellers	Kommunikationsmöglichkeiten
15	Bauteilinformationen für Anlagen verfügbar	Synchronisationsmechanismus zwischen Geräten und Systemen	People Tracking	Mailfunktion
16	Anlagenbeschreibung vom Hersteller	Anzeige Maschinenpläne	Ticket System	Internet

17	Synchronisationsmechanismus zwischen Geräten und Systemen	Störungsbenachrichtigungs-/Anlagenzuweisung zu Personengruppe	Wartungsberichte	Unterstützung Datentypen
18	Überwachung/Protokollierung relevanter Maschinendaten und Zuständen in Form eines Dashboards	Status Störung		Bereitstellung von 3D-Daten
19	Pluginkonzept zur Integration von diversen Schnittstellen und Maschinen	Annahme und Ablehnung von Aufträgen durch Instandhalter möglich		Instandhaltungsmaßnahmenhistorie
20	Anonymisierungsfunktion der Mitarbeiter für das Management	Ersatzteilmanagement		Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit
21	Annahme und Ablehnung von Aufträgen durch Instandhalter möglich	Mobiler Zugriff auf Prozessgeräte		Erfassung der Arbeitszeit der Instandhalter
22	Algorithmus präventive Instandhaltung	Mobiles Abarbeiten von Checklisten und Prüflisten		OPC UA Schnittstelle zum Empfang bzw. Senden von Maschinendaten
23	Priorisierung Instandhaltungsfälle	Anzeige Stördauer (mittel und aktuell)		Maschinenpläne im System verfügbar
24	3D-Darstellung mit Fehlerlokalisierung	Übersicht installierter Anlagen und deren Fehler/Status		Mobiles Arbeiten im Offlinemodus
25	Ersatzteilmanagement	Pluginkonzept zur Integration von diversen Schnittstellen und Maschinen		Erfassung von Reisekosten durch das System
26	Anzeige Softwarestände und Besonderheiten	Überwachung/Protokollierung relevanter Maschinendaten und Zuständen in Form eines Dashboards		Anzeige von aktuellen Vertragsdaten für Auftrag (Leistungen etc.)
27	Link zum Kataster von Hilfs- und Betriebsstoffen für die jeweilige Anlage	Predictive Maintenance (kontinuierlich lernendes, vorausschauendes System)		Signierfunktion durch Auftraggeber und Instandhalter
28	Anzeige Betriebstemperatur	Personalisierung		Annahme und Ablehnung von Aufträgen durch Instandhalter möglich
29	Produktionsplanung/Anlagenbelegung	Barcodescanner		Anzeige Wartungsbericht
30	Bestands-/Liefersituation	Synonymfunktion/Wörterbuch		Serviceprotokoll automatisch in digitaler Version erstellen lassen
31	Intranet/Internet Zugang	Bauteilinformationen		Condition Monitoring

32	Fehlerdatenbank mit Suchfunktion im System verfügbar	Intranetzugang		Fehlerdatenbank mit Suchfunktion im System verfügbar
33	Remotesteuerung Anlage	WhatsApp/Bild versenden/Foto		Predictive Maintenance (kontinuierlich lernendes, vorausschauendes System)
34	Auswertung und Anzeige der aktuellen und mittleren Laufzeiten aller Anlagen (Stördauer/Anzahl Störungen etc.)	Wiki/Forum		Kommunikation mit ERP-System
35	Mobiles Arbeiten im Offlinemodus	Push Benachrichtigung Updates bei Änderungen von HW/SW Komponenten d. Anlage		Pluginkonzept zur Integration von diversen Schnittstellen und Maschinen
36	Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit	Automatische Priorisierung Instandhaltungsfälle		Störmeldung mitsamt Fehlermeldung und Zustand Anlage
37	Eskalationsmöglichkeit mit Informationsweitergabe	"Hilfe-Button"		Störmeldungsformular wird automatisiert vollständig ausgefüllt und an Servicedienstleister versandt
38		Status Mitarbeiter		Bauteilinformationen verfügbar
39		Mobiler Zugriff auf SPS		Bereitstellung Handlungsleitfäden für Kunden
40		Installations-berechtigung Apps		Navigation zum Kunden
41		Spracheingabe		Barcodescanner
42		Laufkarte zur Anlage		Eskalationsmöglichkeit mit Informationsweitergabe
43		Outlook Zugang		
44		Internetzugang		
45		Videofunktion		
46		Eskalationsmöglichkeit mit Informationsweitergabe		

Tabelle 26: Aufgenommene Anforderungen in den untersuchten Unternehmen

Quelle: Eigene Darstellung

Die Anforderungen der einzelnen Unternehmen zeigen, dass deutlich weniger Anforderungen genannt wurden bzw. die Granularität in der Beantwortung im ersten Schritt von den

Anforderungen aus der Literatur abwich. Im Folgenden werden die praktischen Anforderungen miteinander verglichen und anschließend mit den Literaturerkenntnissen abgeglichen.

4.3.3.2 Generische Anforderungen

Die einzelnen Anforderungen wurden dahingehend untersucht, ob sie einen allgemeinen Charakter besitzen. Dazu wurden die vorgestellten spezifischen Anforderungen analysiert und zu generischen Anforderungen aggregiert: Anforderungen, die trotz teils unterschiedlicher Benennung die gleiche Bedeutung hatten und Anforderungen, die gruppiert werden konnten, nachdem sie auf eine gemeinsame Abstraktionsebene gebracht wurden, sind zu einer Anforderung zusammengelegt worden. Das Ergebnis dieses Prozesses ist in Tabelle 27 zu erkennen. Hier sind die zusammengefassten Anforderungen dargestellt, die bei mindestens drei Unternehmen genannt wurden. Das Vorkommen bei den einzelnen Unternehmen ist dabei gesondert gekennzeichnet. Die Nummerierung wird hier mit PNr. im Gegensatz zu LNr. abgekürzt, um eine eindeutige Nummerierung und Unterscheidung von Praxis- wie auch Literaturanforderung zu gewährleisten. Eine Besonderheit stellt dabei die Nummer 14 „Technisches Nutzermanagement inklusive Rollensystem“ dar. Diese Anforderung wurde nicht direkt von den Unternehmen genannt, jedoch wurde diese aufgrund technischer Notwendigkeit hinzugefügt. Kein System und die meisten Anforderungen benötigen diese Kernanforderung.

PNr.	Generische Anforderung	U1	U2	U3	U4
1	Automatische Fehlermeldung	x	x	x	x
2	Detaillierte Fehlermeldung und Störungsart	x	x	x	x
3	Kommunikation (Messenger, Telefon, Mail, Kontakte)	x	x	x	x
4	Condition Monitoring an der Maschine	x	x	x	x
5	Überwachung/Protokollierung relevanter Maschinendaten und Zustände in Form eines Dashboards	x	x	x	x
6	Auftragsverwaltung mit Anzeige Status der Instandhaltungsaufträge durch das System	x	x	x	x
7	(Mobiler) Zugriff auf relevante Maschinendaten	x	x	x	x
8	Zugang zu relevanten Webanwendungen über Internet (z. B. Intranet)	x	x	x	x
9	Dokumentenmanagement und -bearbeitung	x	x	x	x
10	Anzeige von Maschinenplänen (Steuerungspläne, SPS, etc.)	x	x	x	x
11	Handlungsleitfäden/Checklisten für Instandhaltungsaufgaben	x	x	x	x
12	Anlagen-, Bauteil-, und Maßnahmenhistorie	x	x	x	x
13	Synchronisationsmechanismus zwischen Geräten und Systemen	x	x	x	x
14	Technisches Nutzermanagement inklusive Rollensystem	x	x	x	x
15	OPC UA Schnittstelle zum Empfang und Senden von Maschinendaten	x	x		x
16	Bauteilübersicht der Anlagen verfügbar	x	x		x
17	Herstellerunterlagen der Anlage verfügbar	x	x		x
18	Annahme und Ablehnung von Aufträgen durch Instandhalter möglich	x	x		x
19	Ersatzteilmanagement (Informationen und Verfügbarkeit)	x	x	x	x

20	Mobiles Arbeiten im Offlinemodus	x	x	x	x
21	Wartungsplanmanagement & Historie (manuell und automatisch)	x	x	x	x
22	Priorisierung der abzuarbeitenden Tätigkeiten (automatische Priorisierung der Aufträge, Tätigkeiten, Fehler, Auftragsverteilung)	x	x	x	x
23	Remote Zugriff auf Anlage	x	x	x	x
24	Auftragsmanagement in Verbindung mit ERP	x	x	x	x
25	Anlagenübersicht	x	x	x	x
26	Möglichkeit zur Erweiterung der Handlungsanweisungen	x	x		x
27	Automatische Doku und Auswertung der Instandhaltungsmaßnahmen	x	x		x
28	Bereitstellung von 3D-Daten/3D-Darstellung der Anlage	x		x	x
29	Anzeige von Betriebstemperaturen	x		x	x
30	Scannen der Anlagencodierung (Barcodescanner)	x	x		x
31	Auswertung und Anzeige der aktuellen und mittleren Laufzeiten aller Anlagen (Stördauer/Anzahl Störungen etc.)	x	x	x	
32	Predictive Maintenance (kontinuierlich lernendes, vorausschauendes System)	x	x		x
33	Wiki	x	x	x	
34	Fehlerdatenbank mit Suchfunktion	x	x		x
35	Pluginkonzept zur Integration von diversen Schnittstellen und Maschinen	x	x		x
36	Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit	x	x		x
37	Mobiler Zugriff auf die SPS	x	x	x	x
38	Verarbeitung audiovisueller Medien (Erstellen, Speichern, Anzeigen von Foto, Video und Audio)	x	x		x
39	Eskalationsmöglichkeit mit Informationsweitergabe	x	x		x

Tabelle 27: Generische Anforderungen

Quelle: Eigene Darstellung

Des Weiteren sind alle verbliebenen Anforderungen, die nicht den generischen zugeordnet werden konnten, als optionale Anforderungen gruppiert und in Tabelle 28 zusammengefasst worden. Die Grundstruktur der Tabelle ist mit der vorherigen identisch.

PNr.	Optionale Anforderung	U1	U2	U3	U4
1	Anzeigen Softwarestände und Besonderheiten	x			
2	Link zum Kataster der Hilfs - und Betriebsstoffe	x			
3	3D-Darstellung mit Fehlerlokalisierung	x		x	
4	Informationen zur Produktionsplanung/Anlagenbelegung	x			
5	Anonymisierungsfunktion der Mitarbeiter für das Management	x	x		
6	Mobiles Endgerät als Telefonersatz		x		
7	Störungsbenedachrichtigungs-/Anlagenzuweisung zu Personengruppe		x		
8	Personalisierung des Systems		x		
9	Synonymfunktion/Wörterbuch		x		
10	Schichtbuch		x		

11	Push Benachrichtigung Updates bei Änderungen von HW/SW Komponenten d. Anlage		x		
12	Hilfe-Button für Anforderung eines Instandhalters		x		
13	Installationsberechtigung Apps auf mobilem Gerät		x		
14	Spracheingabe		x		
15	Laufkarte zur Anlage		x		
16	Anzeige Windrichtung/-stärke			x	
17	Anzeige der Wetterbeschaffenheit			x	
18	Tracking der Instandhalter			x	
19	GPS Genauigkeit für die Kartenversion (Zustand der Anlage)			x	
20	Anzeige Wartungsbericht			x	x
21	Signierfunktion durch Auftraggeber und Instandhalter				x
22	Anzeige von aktuellen Vertragsdaten für Auftrag (Leistungen etc.)				x
23	Erfassung von Reisekosten durch das System				x
24	Remote Anlagen-/Komponentensteuerung durch Hersteller				x
25	Standardschnittstelle und -datenstruktur an Maschine				x
26	Abgleich von Betriebsdaten (z. B. Protokoll Störfälle & Fehlermeldungen) zwischen Anlagenbetreiber und -hersteller zu zentralen (Cloud-) Speicherung				x
27	Störmeldungsformular wird automatisiert und vollständig ausgefüllt und an Servicedienstleister versandt				x
28	Erfassung der Arbeitszeit der Instandhalter				x
29	Serviceprotokoll automatisch in digitaler Version erstellen lassen				x
30	Bereitstellung Handlungsleitfäden für Kunden				x
31	Navigation zum Kunden				x

Tabelle 28: Optionale Anforderungen

Quelle: Eigene Darstellung

Insgesamt wurden 70 unterschiedliche Anforderungen bei den untersuchten Unternehmen aufgenommen. Die einzelnen Anforderungen wurden von jedem Unternehmen priorisiert. Zur Erreichung einer generischen Priorisierung wurden die ursprünglich von jedem Unternehmen vorgenommenen Priorisierungen übernommen, falls diese identisch waren. Ansonsten wurde der gerundete Mittelwert berechnet und als Priorität genommen. Bei Anforderungen, die sich aus mehreren nicht komplett gleichen Anforderungen zusammensetzen, wurde die Gewichtung anteilig berechnet. In Fällen, bei denen sich kein einheitliches Bild ergab, wurde eine erneute Priorisierung der generischen Anforderung seitens der Unternehmen durchgeführt. Bei dieser mussten sich die Unternehmen auf eine Priorisierung einigen. Diese Priorisierungen der generischen Anforderungen sind in Tabelle 29 dargestellt.

Bei den optionalen Anforderungen konnten die Anforderungen mitsamt der Priorisierung übernommen werden, jedoch spielen diese Anforderungen nur bei Spezialfällen eine Rolle.

PNr.	Priorisierung	Anforderung
1	1	Automatische Fehlermeldung
2	1	Detaillierte Fehlermeldung und Störungsart
3	1	Kommunikation (Messenger, Telefon, Mail, Kontakte)
4	1	Condition Monitoring an der Maschine
5	1	Überwachung/Protokollierung relevanter Maschinendaten und Zustände in Form eines Dashboards
6	1	Auftragsverwaltung mit Anzeige Status der Instandhaltungsaufträge durch das System
7	1	(Mobiler) Zugriff auf relevante Maschinendaten
8	1	Zugang zu relevanten Webanwendungen über Internet (z. B. Intranet)
9	1	Dokumentenmanagement und -bearbeitung
10	1	Anzeige von Maschinenplänen (Steuerungspläne, SPS, etc.)
11	1	Handlungsleitfäden/Checklisten für Instandhaltungsaufgaben
12	1	Anlagen-, Bauteil-, und Maßnahmenhistorie
13	1	Synchronisationsmechanismus zwischen Geräten und Systemen
14	1	Technisches Nutzermanagement inklusive Rollensystem
15	1	OPC UA Schnittstelle zum Empfang und Senden von Maschinendaten
16	1	Bauteilübersicht der Anlagen verfügbar
17	1	Herstellerunterlagen der Anlage verfügbar
18	1	Annahme und Ablehnung von Aufträgen durch Instandhalter möglich
19	2	Ersatzteilmanagement (Informationen und Verfügbarkeit)
20	2	Mobiles Arbeiten im Offlinemodus
21	2	Wartungsplanmanagement & Historie (manuell und automatisch)
22	2	Priorisierung der abzuarbeitenden Tätigkeiten (automatische Priorisierung der Aufträge, Tätigkeiten, Fehler, Auftragsverteilung)
23	2	Remote Zugriff auf Anlage
24	2	Auftragsmanagement in Verbindung mit ERP
25	2	Anlagenübersicht
26	2	Möglichkeit zur Erweiterung der Handlungsanweisungen
27	2	Automatische Doku und Auswertung der Instandhaltungsmaßnahmen
28	2	Bereitstellung von 3D-Daten/3D-Darstellung der Anlage
29	2	Anzeige von Betriebstemperaturen
30	2	Scannen der Anlagencodierung (Barcodescanner)
31	2	Auswertung und Anzeige der aktuellen und mittleren Laufzeiten aller Anlagen (Stördauer/Anzahl Störungen etc.)
32	2	Predictive Maintenance (kontinuierlich lernendes, vorausschauendes System)
33	2	Wiki
34	2	Fehlerdatenbank mit Suchfunktion
35	2	Pluginkonzept zur Integration von diversen Schnittstellen und Maschinen

36	2	Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit
37	3	Mobiler Zugriff auf die SPS
38	3	Verarbeitung audiovisueller Medien (Erstellen, Speichern, Anzeigen von Bildern, Video und Audio)
39	3	Eskalationsmöglichkeit mit Informationsweitergabe

Tabelle 29: Priorisierung der generischen Anforderungen

Quelle: Eigene Darstellung

4.4 Diskussion der Ergebnisse

Zur Diskussion der Ergebnisse sollen die in der Praxis aufgenommen Anforderungen den aus dem Literaturteil extrahierten gegenübergestellt werden. Bei diesem Vergleich der Anforderungen aus Literatur und Praxis ergeben sich gleich mehrere Herausforderungen:

Die jeweiligen Anforderungen befinden sich je nach Bedarf auf unterschiedlichem Aggregations- und Abstraktionsniveau. So kann es vorkommen, dass eine Anforderung im Literaturteil mehrere Anforderungen aus dem Auftragsmanagement aufdeckt und umgekehrt.

Ein weiterer Grund für die Mehrfachnennung ist, dass manche Anforderungen Spezialisierungen anderer Anforderungen sind und somit automatisch die grundlegendere Anforderung voraussetzen. Dies gilt für die Anforderungen aus dem Literatur- wie auch aus dem Praxisteil. Eine gesonderte Abhängigkeitsdarstellung wird aus Komplexitätsgründen in dieser Dissertation nicht verfolgt. Damit verbunden sind auch Anforderungen, welche die grundlegende Infrastruktur behandeln und daher grundlegend für die Funktion anderer sind, jedoch diese keine Spezialisierung im eigentlichen Sinne darstellen. Diese Anforderungen sind gesondert als generisch gekennzeichnet worden.

Trotz dieser Herausforderungen sind die Überschneidungen leicht erkennbar und eine Gegenüberstellung ermöglicht Rückschlüsse über die Validität der in der Praxis aufgenommenen Anforderungen. Daher werden nachfolgend in Kapitel 4.4.1 die generischen und spezifischen Anforderungen mit den jeweiligen vergleichbaren Anforderungen aus der Literaturanalyse gegenübergestellt. Anschließend werden in Kapitel 4.4.2 die Ergebnisse diskutiert.

4.4.1 Diskussion Verbindung Literatur- und Praxisanforderung

In Tabelle 30 werden die generischen und in Tabelle 31 die spezifischen Anforderungen den Literaturanforderungen gegenübergestellt den Literaturanforderungen gegenübergestellt. Beide Tabellen sind bis auf die Unterscheidung zwischen generischen und spezifischen Anforderungen identisch aufgebaut.

- In der ersten Spalte (PGNr.) steht die Nummer der generischen Praxisanforderung.
- In der zweiten Spalte (Anforderung) wird der Name der jeweiligen Praxisanforderung genannt.
- Die dritte Spalte (LitAnf. = Literaturanforderung) enthält die Literaturnummer (LNr.) der Anforderungen aus der Literaturanalyse, die mit den Anforderungen aus der Praxis vergleichbar sind. Es können dabei mehrere Anforderungen genannt werden, wenn die Praxisanforderung mehrere Anforderungen aus der Literatur abdeckt bzw. diesen ähnlich ist.
- Weiterhin sind in der vierten Spalte verbundene Anforderungen (Verb. Anf. = Verbundene Anforderung) aufgeführt, die implizit auch durch die Anforderung aus der Praxis abgedeckt werden müssen, jedoch nicht direkt genannt werden oder diese nur teilweise erfüllen.
- In der fünften Spalte (Topanf. = Topanforderung) ist hinterlegt, ob die Literaturanforderungen, die durch die Praxisanforderungen abgedeckt sind, eine der am häufigsten genannten Anforderungen aus Tabelle 24 enthalten. Dabei können auch mehrere dieser Anforderungen abgedeckt werden. Diese Anforderungen werden im folgenden Topanforderungen genannt, da sie besonders häufig genannt werden.

- In Spalte TopVerb. wird die gleiche Logik auf die verbundenen Anforderungen angewandt.

Am Ende der jeweiligen Tabellen befinden sich drei Summenreihen, die eine Übersicht über die Anzahl der vergleichbaren Anforderungen geben. In der ersten Reihe wird die Anzahl der Anforderungen eingetragen, die in den jeweiligen Spalten vorkommen. In der zweiten Reihe werden Duplikate der jeweiligen Spalten eliminiert und die Anzahl der insgesamt vorkommenden unterschiedlichen Anforderungen genannt. Die dritte Spalte bildet die Summe aller Anforderungen, wiederum werden die Duplikate entfernt und so ergibt sich eine Zahl der Anforderungen, die direkt oder indirekt durch die praktischen Anforderungen behandelt werden.

PGNr.	Anforderung	LitAnf.	Verb. Anf.	Topanf.	TopVerb.
1	Automatische Fehlermeldung	91; 15	14		x
2	Detaillierte Fehlermeldung und Störungsart	90; 92		x	
3	Kommunikation (Messenger, Telefon, Mail, Kontakte)	67		x	
4	Condition Monitoring an der Maschine	96		x	
5	Überwachung/Protokollierung relevanter Maschinendaten und Zustände in Form eines Dashboards	86; 98	96	x, x	x
6	Auftragsverwaltung mit Anzeige Status der Instandhaltungsaufträge durch das System	14; 17		x	
7	(Mobiler) Zugriff auf relevante Maschinendaten	74	10; 75		x
8	Zugang zu relevanten Webanwendungen über Internet (z. B. Intranet)		74		
9	Dokumentenmanagement und -bearbeitung	101		x	
10	Anzeige von Maschinenplänen (Steuerungspläne, SPS, etc.)		41		x
11	Handlungsleitfäden/Checklisten für Instandhaltungsaufgaben	113; 114	115; 116; 117, 106, 105	x	x
12	Anlagen-, Bauteil-, und Maßnahmenhistorie	85		x	
13	Synchronisationsmechanismus zwischen Geräten und Systemen				
14	Technisches Nutzermanagement inklusive Rollensystem	125	127		
15	OPC UA Schnittstelle zum Empfang und Senden von Maschinendaten				
16	Bauteilübersicht der Anlagen verfügbar	40		x	
17	Herstellerunterlagen der Anlage verfügbar	41		x	
18	Annahme und Ablehnung von Aufträgen durch Instandhalter möglich	19			
19	Ersatzteilmanagement (Informationen und Verfügbarkeit)	51; 52	53; 54; 55; 56; 57	x, x	
20	Mobiles Arbeiten im Offlinemodus	76			

PGNr.	Anforderung	LitAnf.	Verb. Anf.	Topanf.	TopVerb.
21	Wartungsplanmanagement & Historie (manuell und automatisch)	29; 30	31	x	
22	Priorisierung der abzuarbeitenden Tätigkeiten (automatische Priorisierung der Aufträge, Tätigkeiten, Fehler, Auftragsverteilung)	16; 32; 33; 94	14; 35; 38	x, x	x, x, x
23	Remote Zugriff auf Anlage	11	10	x	x
24	Auftragsmanagement in Verbindung mit ERP	118			
25	Anlagenübersicht	39	46	x	
26	Möglichkeit zur Erweiterung der Handlungsanweisungen		105; 101		x, x
27	Automatische Doku und Auswertung der Instandhaltungsmaßnahmen	102	109; 71; 7	x	
28	Bereitstellung von 3D-Daten/3D-Darstellung der Anlage	48			
29	Anzeige von Betriebstemperaturen	96		x	
30	Scannen der Anlagencodierung (Barcodescanner)	73			
31	Auswertung und Anzeige der aktuellen und mittleren Laufzeiten aller Anlagen (Stördauer/Anzahl Störungen etc.)	87		x	
32	Predictive Maintenance (kontinuierlich lernendes, vorausschauendes System)	97		x	
33	Wiki	110	105		
34	Fehlerdatenbank mit Suchfunktion	84; 108	89	x	
35	Pluginkonzept zur Integration von diversen Schnittstellen und Maschinen	118			
36	Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit	20; 58; 60		x	
37	Mobiler Zugriff auf die SPS	11		x	
38	Verarbeitung audiovisueller Medien (Erstellen, Speichern, Anzeigen von Foto, Video und Audio)	70; 78; 29	101	x	x
39	Eskalationsmöglichkeit mit Informationsweitergabe	3; 23	6		
Summe		50	32	26	12
Summe abzüglich Duplikate der Kategorie		46	27	23	8
Summe abzüglich Duplikate aggregiert		68		28	

Tabelle 30: Generische Anforderungen in Verbindung mit den Anforderungen der Literaturrecherche

Quelle: Eigene Darstellung

Die generischen praktischen Anforderungen behandeln insgesamt 50 Anforderungen aus der Literatur direkt, 32 sind indirekt verbunden, wobei sie 26 Topanforderungen direkt und 12 indirekt nennen. Nach Reduzierung der mehrmals genannten Anforderungen werden durch die 39 praktischen generischen Anforderungen insgesamt 46 Anforderungen aus der Literatur direkt abgedeckt, 27 sind hingegen indirekt verbunden. Die 46 abgedeckten Anforderungen enthalten 23 Topanforderungen, wodurch 82 % der Topanforderungen direkt durch die

generischen praktischen Anforderungen abgedeckt werden. Daher kann von einer großen Überschneidung des generischen Kerns der praktischen Anforderungen mit den Topanforderungen ausgegangen werden. Folgende Topanforderungen werden nicht direkt durch die praktischen Anforderungen abgedeckt:

- LNr. 10: Remote Maintenance/Fokus Fernüberwachung
- LNr. 35: Bewertung und Erfassung von kritischen Risiken, Prioritätszuweisung sowie Klassifikation von Anlagen/Komponenten
- LNr. 38: Kostenmanagement der Instandhaltung (Mitarbeiter, Materialien, Tools, Historie, Planung und Bewertung)
- LNr. 105: Wissensmanagement
- LNr. 106: Informationen zu aktuellem Instandhaltungsfall automatisch verknüpfen (CBR, Wahrscheinlichkeiten etc.)

Die genannten Anforderungen werden jedoch teilweise indirekt abgedeckt. Remote Maintenance/Fokus Fernüberwachung wird nicht direkt genannt, sondern nur von der Remote Zugriff auf die jeweilige Maschine. Dies stellt eine Spezialisierung der Fernüberwachung dar, womit diese zwangsläufig auch erfüllt werden muss.

Nr. 35 und Nr. 38 sind in der Genauigkeit nicht in den praktischen Anforderungen enthalten. Es wird weder ein dediziertes Kostenmanagement noch eine Bewertung von kritischen Risiken ähnlich dem RCM-Vorgehen bei der praktischen Anforderungserhebung genannt. Jedoch ist für die Erfüllung der PNr. 22 „Priorisierung der abzuarbeitenden Tätigkeiten (automatische Priorisierung der Aufträge, Tätigkeiten, Fehler, Auftragsverteilung)“ sinnvoll, dass eine Priorisierung der Aufträge hinsichtlich der Kritikalität sowie der Kosten durchgeführt wird. Werden diese Aspekte in der Praxisanforderung implementiert, so sind Teile der Literaturanforderungen erfüllt bzw. müssen erfüllt werden.

Daneben wird kein explizites Wissensmanagement (LNr. 105) in der Praxis gefordert, sondern es bestehen nur Anforderungen, die dieses in Spezialisierungsfällen umsetzen. Sei es durch PNr. 33 „Wiki“, durch PNr. 26 „Möglichkeit zur Erweiterung der Handlungsanweisungen“ sowie vor allem PNr. 11 „Handlungsleitfäden/Checklisten für Instandhaltungsaufgaben“. LNr. 106 „Informationen zu aktuellem Instandhaltungsfall automatisch verknüpfen (CBR, Wahrscheinlichkeiten etc.)“ wird ebenso durch die Handlungsempfehlungen teilweise abgedeckt, da diese automatisch zu ähnlichen Fällen zugeordnet werden müssen. Jedoch wird in der Praxis kein explizites CBR oder eine Ähnlichkeitsanalyse gefordert.

Drei der praktischen Anforderungen finden sich nicht direkt in den Literaturanforderungen wieder, zwei der praktischen Anforderungen PNr. 15 „OPC UA Schnittstelle zum Empfang und Senden von Maschinendaten“ und PNr. 13 „Synchronisationsmechanismus zwischen Geräten und Systemen“ lassen sich überhaupt nicht in der Literatur finden.

Zusammengefasst werden alle 28 Topanforderungen entweder direkt oder indirekt durch die generischen Anforderungen aus der Praxis abgedeckt. Insgesamt lassen sich mit den generischen Anforderungen aus der Praxis über 50 % der Literaturanforderungen in Verbindung setzen. Aus den 39 praktischen generischen Anforderungen lässt sich eine Verbindung zu 68 Literaturanforderungen ableiten, was einen Faktor von 1,74 von praktischer zur Literaturanforderungen ergibt. Der Faktor bestimmt, wie viele Anforderungen aus der Literatur durch eine Anforderung aus der Praxis abgedeckt werden. Liegt dieser höher als 1, lässt dies auf einen höheren Abstraktionsgrad und Generalisierbarkeit der Anforderungen aus

der Praxis schließen. Der Wert 1,74 ergibt sich daher durch die Division von 68 durch 39. Daraus lässt sich auf eine relativ breite Abdeckung und einen höheren Aggregations- bzw. Abstraktionsgrad der generischen Anforderungen aus der Praxis schließen.

In der nachfolgenden Tabelle ändert sich die Nummerierung der Anforderungen aufgrund des Wechsels von generischer zu optionaler Anforderungen von PGNr. zu PONr.

PONr.	Spezifische Anforderungen	LitAnf	Verb. Anf.	Topanf.	TopVerb.
1	Anzeigen Softwarestände und Besonderheiten	50			
2	Link zum Kataster der Hilfs - und Betriebsstoffe	52		x	
3	3D-Darstellung mit Fehlerlokalisierung	95	48		
4	Informationen zur Produktionsplanung/Anlagenbelegung				
5	Anonymisierungsfunktion der Mitarbeiter für das Management				
6	Mobiles Endgerät als Telefonersatz		67		x
7	Störungsbenedachrichtigungs-/Anlagenzuweisung zu Personengruppe	42			
8	Personalisierung des Systems				
9	Synonymfunktion/Wörterbuch				
10	Schichtbuch		118		
11	Push Benachrichtigung Updates bei Änderungen von HW/SW Komponenten d. Anlage				
12	Hilfe-Button für Anforderung eines Instandhalters	91			
13	Installationsberechtigung Apps auf mobilem Gerät				
14	Spracheingabe				
15	Laufkarte zur Anlage				
16	Anzeige Windrichtung/-stärke	120			
17	Anzeige der Wetterbeschaffenheit	120			
18	Tracking der Instandhalter	64			
19	GPS Genauigkeit für die Kartenversion (Zustand der Anlage)	44; 45			
20	Anzeige Wartungsbericht	31			
21	Signierfunktion durch Auftraggeber und Instandhalter	78			
22	Anzeige von aktuellen Vertragsdaten für Auftrag (Leistungen etc.)	1; 43			
23	Erfassung von Reisekosten durch das System	61			
24	Remote Anlagen-/Komponentensteuerung durch Hersteller	11	10	x	x
25	Standardschnittstelle und -datenstruktur an Maschine		135		
26	Abgleich von Betriebsdaten (z. B. Protokoll Störfälle & Fehlermeldungen) zwischen Anlagenbetreiber und -Hersteller zu zentralen (Cloud-) Speicherung		122		

PONr.	Spezifische Anforderungen	LitAnf	Verb. Anf.	Topanf.	TopVerb.
27	Störmeldungsformular wird automatisiert und vollständig ausgefüllt und an Servicedienstleister versandt	6			
28	Erfassung der Arbeitszeit der Instandhalter	59			
29	Serviceprotokoll automatisch in digitaler Version erstellen lassen	7	101		x
30	Bereitstellung Handlungsleitfäden für Kunden	47			
31	Navigation zum Kunden	79			
Summe		21	7	2	3
Summe abzüglich Duplikate der Kategorie		20	7	2	3
Summe abzüglich Duplikate aggregiert		27		5	

Tabelle 31: Spezifische Anforderungen im Vergleich mit den Anforderungen aus der Literaturrecherche

Quelle: Eigene Darstellung

Ein großer Kontrast zu den generischen praktischen Anforderungen ist bei den optionalen praktischen Anforderungen zu erkennen. Diese 31 Anforderungen stehen in Verbindung zu aggregierten 27 Anforderungen. Damit besitzen sie einen Faktor von 0,87 im Gegensatz zu den 1,74 der generischen Anforderungen. Ein weiterer Unterschied liegt darin, dass durch die optionalen praktischen Anforderungen kaum Topanforderungen abgedeckt werden. Direkt lediglich zwei, indirekt drei Stück. Bei der indirekten Verbindung handelt es sich vor allem um allgemeine Anforderungen wie

- LNr. 101 das Wissensmanagement,
- LNr. 10 Remote Maintenance/Fokus Fernüberwachung sowie
- LNr. 67 Kommunikation und Kontaktaufnahme zu anderen Mitarbeitern/Personen (E-Mail, SMS, Nachrichten, Telefon),

welche speziellere Anforderungen aus der Praxis indirekt ermöglichen. Das gleiche gilt für die direkte Verbindung zu den Topanforderungen Fernzugriff sowie dem Link zu dem Kataster, die ebenso Spezialfälle von Literaturanforderungen darstellen, jedoch auch einen Teil der Topanforderung implementieren und ebenso von den generischen praktischen Anforderungen erfüllt werden.

Vier optionale Anforderungen werden nur indirekt von Literaturanforderungen behandelt, sieben Stück werden komplett gar nicht in der Literatur erwähnt. Hier handelt es sich insbesondere um Anforderungen, die in Verbindung mit einem mobilen Endgerät stehen, aber auch Anonymisierungsfunktionen und Zugriff auf die Produktions- bzw. Anlagenplanung. Hier sind Lücken in der Literatur erkennbar. Weiterhin zeigen die Ergebnisse, dass sich die praktischen Erkenntnisse hinsichtlich der optionalen Anforderungen mit der Häufigkeit der Nennung in der Literatur decken.

Zusammengefasst lassen sich über 83 generische, wie auch optionale praktische Anforderungen aus der Literatur in Verbindung mit den 70 praktischen Anforderungen setzen. Damit decken die praktischen Anforderungen ca. 61 % aller Anforderungen in der Literatur ab, von den häufig genannten sogar 100 %. Dabei decken diese 61 % der Literaturanforderungen 628 Nennungen ab. Daraus ergibt sich, dass diese 61 % der Literaturanforderungen 84 % aller Nennungen auf

sich vereinen. Aufgrund dieser Überdeckung kann von den erhobenen generischen Anforderungen aus der Praxis auf eine herausragende Relevanz für umfassende Instandhaltungssysteme gesprochen werden.

Einige Nennungen von Anforderungen aus der Praxis unterscheiden sich stark von den Anforderungen aus der Literatur. Die generische Anforderung 1 – Automatische Fehlermeldung – ist bei der Anforderungserhebung in der Praxis die am höchsten priorisierte Anforderung. Sie umfasst die Literaturanforderungen LNr. 98 und LNr. 16, die beide jedoch nicht in den häufigsten Anforderungen bei den Literaturanforderungen zu finden sind. Instandhaltungssysteme ohne automatische Fehlermeldungen, die durch die Maschinen ausgelöst werden, sind jedoch undenkbar. Die Anforderungen PGNr. 13 und PGNr. 15 sind in der Form in der Literatur ebenso nicht zu finden. Die Wichtigkeit einer OPC UA Schnittstelle der Anlagen und Komponenten zum Senden von Daten ist dort nicht erkennbar, ebenso wie ein Synchronisationsmechanismus zwischen Geräten und Systemen. Aspekte der Dezentralität und die Schwierigkeit des Einbindens von Maschinen, Sensoren und externen Daten werden in der Literatur weit weniger mit konkreten Anforderungen behandelt. Dies liegt zum einen daran, dass die in der Literatur untersuchten Systeme meist keinen integrativen Ansatz haben, also meist nur ein CMS, MSS oder ein CMMS behandeln und nicht die Verbindung zwischen diesen in den Vordergrund stellen. Zum anderen schlagen sich Trends wie CPS und neuere technische Möglichkeiten noch zu wenig in den untersuchten Architekturen und Systemen nieder.

Ein weiterer Unterschied betrifft das Arbeiten im Offlinemodus mit einem mobilen Endgerät. Diese Anforderung (PGNr. 20) stellt ein wichtiges Anliegen der untersuchten Unternehmen dar. In der untersuchten Literatur wird es hingegen nur zweimal erwähnt. Ein zusätzlich herausstechender Punkt ist, dass in den Anforderungen in der Praxis der (Fern-)Zugriff auf die SPS und die Maschinenpläne der SPS von großer Bedeutung sind, in der Literatur hingegen nur unter den Begriffen Fernwartung, Remote Maintenance subsummiert werden oder nur von der Verfügbarkeit von relevanten technischen Dokumenten gesprochen wird. Weitere Unterschiede existieren bei den spezifischen Anforderungen. Hier finden sich viele Anforderungen, die keine Entsprechung in der Literaturanalyse finden. Eine Anonymisierungsfunktion für die Mitarbeiter aus Datenschutzgründen, eine Personalisierungsfunktion des Systems und weitere Aspekte werden nicht behandelt.

4.4.2 Diskussion der Anforderungsbetrachtung

Die Aufnahme der Anforderungen aus Literatur und Praxis und der Vergleich der Ergebnisse führt zu folgender Erkenntnis:

- Trotz vieler unterschiedlicher Instandhaltungssysteme existiert ein generischer Kern an Anforderungen an umfassende Instandhaltungssysteme.

Denn 22 aufgenommene Kernanforderungen der Literatur decken 52 % aller Anforderungsnennungen in der Literatur ab und werden als Topanforderungen bezeichnet. Die 28 häufigsten genannten Anforderungen decken gar knapp 60 % aller Nennungen ab. Durch den Vergleich der Literaturanforderungen mit den Praxisergebnissen im vorherigen Kapitel ist erkennbar, dass die generischen Anforderungen aus der Praxis 82 % dieser 28 Literaturanforderungen direkt abdecken. Zählt man die indirekten Nennungen dazu, sind es 100 %. Damit werden die Ergebnisse aus der Literatur durch die Praxisergebnisse validiert und bestätigt. Die generischen Anforderungen aus der Praxis stellen demnach mit hoher Wahrscheinlichkeit den generischen Kern eines IIS dar.

Weiter decken die Praxisanforderungen zwar nur 61 % aller genannten Anforderungen aus der Literatur ab, diese 61% enthalten jedoch alle besonders häufig genannten Anforderungen. Diese decken somit knapp 84 % aller Nennungen von Anforderungen aus der Literatur ab. Daher kann konstatiert werden:

- Die häufig genannten Anforderungen, die jeweils in Praxis und in der Literatur behandelt werden, sind von besonderer Bedeutung.
- Die häufig genannten Ergebnisse der Literaturrecherche werden auch in der Praxis bestätigt.

Daraus folgt auch, dass sich Praxis und Literaturanforderungen stark überschneiden und die nicht in der Praxis genannte Literaturanforderungen demzufolge auch tendenziell eine geringere Relevanz besitzen.

Weiter werden zwei generische Anforderungen in der Literatur überhaupt nicht genannt. Zum einen die essentielle Verbindung der Maschinen und des Instandhaltungssystems mit OPC UA und die Synchronisation zwischen den einzelnen Endgeräten. Dies liegt höchstwahrscheinlich daran, dass OPC UA ein relativ neuer Standard ist, der erst in den letzten Jahren besonders im europäischen Raum an Fahrt gewinnt. Es ist aktuell davon auszugehen, dass er in Zukunft einen Standard bei der Maschinenkommunikation darstellt. Bei der untersuchten Literatur war dies scheinbar nicht der Fall, da dies nicht vorhersehbar war oder die Anforderungen der Maschinenanbindung wenig Beachtung geschenkt wurde. Bei der Synchronisation lässt sich das Fehlen der Anforderungen so erklären, dass bisher kaum Systeme umfassend betrachtet wurden, also das Zusammenspiel von Backend zu mobilen Endgerät zu Anlage. Die meisten untersuchten Systeme und Prototypen in der Literatur konzentrieren sich auf genau ein Endgerät. Bei der aktuellen Entwicklung mit einer Vielzahl an unterschiedlichen, sich im Einsatz befindlichen Geräten, erscheint die Synchronisation als eine essentielle Anforderung, besonders wenn Konnektivitätsprobleme vorherrschen. Mit diesen Punkten und den in Kapitel gesondert dargelegten Unterschieden kann folgendes festgehalten werden:

- Die gefundenen Ergebnisse in der Praxis fügen sich weitestgehend in die Literaturergebnisse ein und ergänzen diese insbesondere hinsichtlich dezentraler Aspekte und der umfassenden Sicht, die endgeräteübergreifende Faktoren eines Systems betonen.

Mit diesem Punkt und der tendenziell höheren Aggregationsstufe der praktischen Anforderungen sowie der Bestätigung durch den Abgleich mit den Literaturanforderungen werden diese als Grundlage für den Schnitt der Referenzarchitektur genommen. Die Literaturanforderungen werden nur als Ergänzung herangezogen. Folglich kann daher folgendes festgehalten werden:

- Generische Praxisanforderungen in Kombination mit den Literaturanforderungen können als Grundlage zur Identifizierung des generischen Kerns eines IIS genutzt werden.

Damit ermöglichen die Ergebnisse der Anforderungsuntersuchung eine umfassende Betrachtung der Anforderungen an ein IIS. Weiter werden ein Vergleich und die Ableitung eigener Anforderungen bei der Entwicklung anhand dieser Ergebnisse ermöglicht. Die optionalen Anforderungen aus der Praxis und die geringe Überschneidung mit den häufig genannten Anforderungen aus der Literatur sprechen weiter für die Optionalität dieser

Anforderungen. Aufgrund des unterschiedlichen Abstraktionsgrades der Anforderungen und deren Form kann jedoch nicht von eindeutigen Anforderungen gesprochen werden.

Zusammenfassend kann daher konstatiert werden, dass die aufgenommenen Praxisanforderungen im folgenden Verlauf der Arbeit als Grundlage zur Erstellung der Referenzarchitektur genutzt werden. Die damit verbundenen Literaturanforderungen fließen nicht nur in die praktischen Anforderungen hinein, sie werden auch dazu genutzt, Erweiterungen der logischen funktionalen Module, die auf den generischen und optionalen Anforderungen aus der Praxis basieren, in Kapitel 6 auszuformulieren.

4.5 Zusammenfassung

Aufbauend auf den Informationen aus Kapitel 2 und 3 konnten die für die Erstellung der Referenzarchitektur nötigen Grundlagen zur Anforderungserhebung in Literatur und Praxis geschaffen werden. Durch die Eingrenzung einer Referenzarchitektur, die Herleitung eines Vorgehens zur Erstellung und den Anforderungen, die sich aus der Domäne der Instandhaltung und die für die Instandhaltung nötigen Informationssysteme resultieren, konnte der Rahmen für die Literaturanalyse zur Aufnahme der Anforderungen aus der Literatur gelegt werden. Die Literaturanalyse wurde in diesem Kapitel vorgestellt mit dem Ergebnis, dass einige der essentiellen Anforderungen an unterschiedliche Instandhaltungssystemen identifiziert werden konnten. Aufbauend auf den Anforderungen aus der Literatur wurde eine iterative Anforderungserhebung bei vier Unternehmen durchgeführt. Diese Unternehmen und das Vorgehen wurden beschrieben und die jeweiligen daraus extrahierten Anforderungen präsentiert. Aufgrund der großen Bandbreite der Anforderungen aus diesen Unternehmen und deren unterschiedliche Ausrichtung konnten aus den aufgenommenen Informationen generische und optionale Anforderungen für ein IIS gewonnen und priorisiert werden. Die Resultate dieser umfassenden Anforderungsermittlung wurden diskutiert und Unterschiede und Gemeinsamkeiten zwischen Literatur und Praxisuntersuchungen hervorgehoben. Als Ergebnis für die weitere Erstellung der Referenzarchitektur werden die generischen und optionalen Anforderungen aus der Praxis benutzt. Bevor die Integration dieser Anforderungen in die Referenzarchitektur in Kapitel 6 erläutert wird, soll zum besseren Verständnis die Darstellungsform der Referenzarchitektur im nachfolgenden Kapitel vorgestellt werden.

5 Darstellungsmodell der RAI

5.1 Einleitung

Die Art der Darstellung einer Referenzarchitektur ist entscheidend für deren Akzeptanz und beeinflusst stark die Art und Weise, wie mit dieser gearbeitet wird. Das mit einer Referenzarchitektur verbundene Ziel und die damit arbeitende Zielgruppe beeinflussen daher stark die Wahl der Darstellung. Für Guessi et al. (2014, S. 2) ist eine adäquate Darstellung essentiell für den Erfolg einer Referenzarchitektur, da durch diese Darstellung die Referenzarchitektur für eine große Bandbreite an Stakeholdern verständlich gemacht werden kann. In diesem Kapitel soll daher die für die RAI gewählte Darstellungsart einer Referenzarchitektur vorgestellt werden, um die Darstellung von Referenzarchitekturen im Allgemeinen und der RAI im Speziellen zu diskutieren und um die Grundlagen zum Verständnis der RAI bereitzustellen.

Das Kapitel widmet sich dazu dem Hintergrund bzgl. der Darstellung von Softwarearchitekturen und Referenzarchitekturen in Kapitel 5.2. Dort wird auch die Basis der Darstellung von Softwarearchitekturen nach Kruchten (1995) erläutert und basiert auf den Ausführungen von Reidt et al. (2016a). Anschließend werden die Anpassungen und die konkrete Darstellung der RAI in Kapitel 5.3 vorgestellt. Abschließend werden die Erkenntnisse in Kapitel 5.4 zusammengefasst.

5.2 Darstellung von Referenzarchitekturen

In der Literatur ist keine einheitliche Art der Darstellung für Referenzarchitekturen im Allgemeinen und in Abhängigkeit der Ziele einer Referenzarchitektur erkennbar. Eine Bevorzugung simpler Darstellungsarten, welche auf keinerlei festgelegten Konzepten basieren, kann jedoch festgestellt werden (Nakagawa et al., 2014a, S. 144). Weiter kann eine große Anzahl an Publikationen identifiziert werden, die Referenzarchitekturen anhand verschiedener Sichten darstellen (Galster/Avgeriou, 2011; Nakagawa et al., 2014b). Grundlegend basiert ein Großteil davon auf dem 4+1 Sichtenmodell für Softwarearchitekturen von Kruchten (1995). Reed, 2002 (S. 4) propagiert diese Art der Darstellung für Referenzarchitekturen als idealtypisch. Die Darstellung in verschiedene Sichten zu untergliedern hat den Vorteil, dass bestimmte Aspekte der Architektur von einer einzelnen Sicht hervorgehoben und zielgruppenspezifisch dargestellt und mit den restlichen Sichten verbunden werden können. Somit wird eine stakeholderzentrierte Darstellung ermöglicht, die jedoch einen gemeinsamen Kern aufweist.

Eine Kernanforderung an die RAI ist, dass durch sie eine domänenübergreifende Zusammenarbeit ermöglicht bzw. gefördert werden soll. Zur Förderung der Kommunikation werden bei der Modellrepräsentation von Referenzmodellen in der Regel Sprachen ohne formale Semantik und nur mit einer bedingt wohl-definierten Syntax verwendet, da diese Art von Sprachen zuweilen eine höhere Akzeptanz, Verständlichkeit und Verwendbarkeit zugeschrieben wird (Fettke/Loos, 2004b, S. 333–334). Dieser Aspekt ist auf die Referenzarchitekturmodellierung analog übertragbar, daher wird bei der RAI weitestgehend mit einer simplen semi-formalen Darstellungsart ohne hohe Notationskomplexität modelliert. Da bei Kruchten die Art der Notation und die konkrete Darstellung der einzelnen Sichten nicht fixiert ist, bietet dieser Aspekt eine gute Grundlage zur Darstellung von Referenzarchitekturen. Aus diesen Gründen wird die Darstellungsweise nach Kruchten als Basis für die Darstellung der RAI ausgewählt. Im Nachfolgenden wird der Abschnitt zu Kruchten aus Reidt et al. (2016a) wiedergegeben.

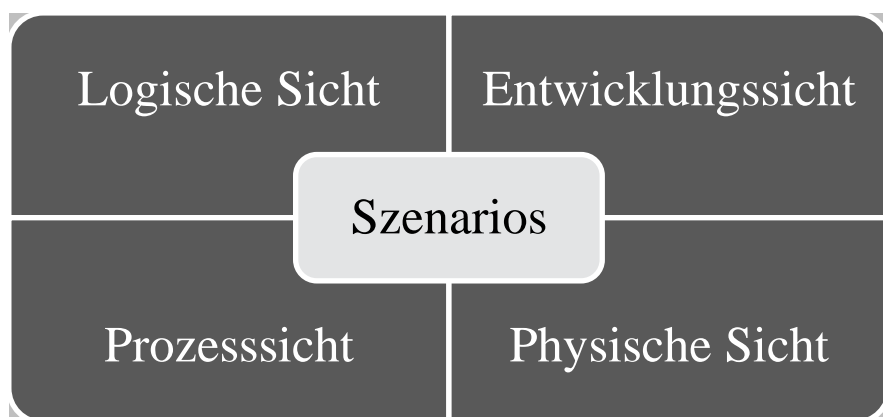


Abbildung 13: 4+1 Sichten einer Softwarearchitektur

Quelle: In Anlehnung an Kruchten (1995)

Das Modell von Kruchten unterscheidet ursprünglich zwischen 4+1-Sichten, die jeweils einen Blickwinkel eines bestimmten Stakeholders (z. B. Endnutzer, Entwickler oder Projektmanager) auf das zu entwickelnde System darstellen. Dies erlaubt für jeden Projektbeteiligten eine übersichtliche Darstellung der benötigten Informationen. Darüber hinaus können neue Sichten einfach eingefügt werden, falls zusätzliche Blickpunkte wünschenswert sind. Die Sichten selbst

sind in ihrer Notation und ihrem Aufbau nicht festgelegt und die Auswahl der adäquatesten Darstellungsform wird an den jeweiligen Anwendungsfall angepasst. Nachfolgend werden die auf Abbildung 13 dargestellten 4+1-Sichtenbeschrieben und anschließend wird auf Anpassungen hinsichtlich der Referenzarchitektur eingegangen.

5.2.1 Logische Sicht

Diese Sicht beschreibt die Funktionalitäten und Services, die dem Endnutzer zur Verfügung stehen. Die logische Sicht unterstützt vordergründig die funktionalen Anforderungen. Dem Endnutzer wird somit die logische Architektur dargestellt, womit er die zu erwartenden Services identifizieren kann. Um diese Ansicht zu erzeugen, wird das vorliegende System analysiert, in eine Menge von sogenannten „Schlüsselabstraktionen“ zerlegt und anschließend werden Gemeinsamkeiten und Designelemente zwischen diesen herausgearbeitet (Kruchten, 1995, S. 44). Schlüsselabstraktionen bei Kruchten können dabei als Objekte oder Klassen angesehen werden. Die Informationen dieser Sicht werden meist per UML-Klassendiagramm, -Aktivitätsdiagrammen oder -Sequenzdiagrammen dargestellt.

5.2.2 Entwicklungssicht

Die Entwicklungssicht beschreibt das System vom Standpunkt eines Entwicklers und beschäftigt sich mit dem Softwaremanagement. Die logische Sicht bildet hierbei die Grundlage. Aus den einzelnen Objekten und Klassen der logischen Sicht werden in der Entwicklungssicht Softwarekomponenten erzeugt. Deren Verteilung auf verschiedene Subsysteme und Schichten wird bspw. durch UML-Komponenten- oder -Paketdiagramme abgebildet. Zusätzlich werden diese Komponenten in eine Hierarchie einsortiert, wodurch sich eine für den Entwickler hierarchische Darstellung der zu implementierenden Funktionalitäten ergibt. Diese sind per klaren Interfaces voneinander getrennt. Hierdurch können Arbeitspakete einzelnen Entwicklern oder kleinen Teams zugewiesen werden.

5.2.3 Prozesssicht

In der Prozesssicht werden die dynamischen Aspekte des Systems verdeutlicht. Der Zusammenhang zwischen den Elementen aus der logischen Sicht, deren Zuordnung zu Kontrollflüssen, Kommunikationswegen und der notwendigen Synchronisation werden beschrieben. Dadurch wird das Laufzeitverhalten ersichtlich und zusätzlich nichtfunktionale Anforderungen wie Parallelität, Verteilung, Integration, Performance und Skalierbarkeit hervorgehoben.

5.2.4 Physische Sicht

Die „Physical View“ bzw. physische Sicht beschäftigt sich mit der Systemtopologie, der Verteilung und Kommunikation der verschiedenen Komponenten auf physischer Ebene. Es ist eine Sicht für Systemarchitekten, welche die Verteilung des zu entwickelnden Systems auf verschiedene Hardwarekomponenten und Netzwerkverbindungen planen.

5.2.5 Szenarios

Die der Architektur zugrundeliegenden Szenarien bilden die fünfte Sicht. Diese Szenarien stellen die wichtigsten Anwendungsfälle der Architektur bzw. der Anwendung dar. Die

Szenarios sind dabei teils redundant mit den vorhergehenden Sichten (daher das „+1“), jedoch helfen die dort enthaltenen Abläufe Architekturelemente zu identifizieren, zu veranschaulichen und die Architektur mit all ihren Sichten zu überprüfen. Die Szenarien können in Form von bspw. Use Cases grafisch oder textuell beschrieben werden. Die beschriebenen Use Cases gehen schlussendlich vollständig in den beschriebenen Sichten auf und dienen als Implementierungsgrundlage für Anwendungen. Sie verbinden damit alle anderen Sichten.

5.3 Angepasstes Sichtenmodell der RAI

Die vorgestellten Sichten bilden in den meisten Fällen eine Grundlage zur Darstellung von konkreten Softwarearchitekturen. Im Falle einer abstrakteren Referenzarchitektur wie der RAI werden jedoch andere Anforderungen an die Darstellung und die Architektur selbst gestellt. Sie muss als Vorlage benutzt werden können, um technische, konkrete Architekturen zu entwickeln, jedoch genügend Freiheitsgrade besitzen, damit die Referenzarchitektur technologieneutral und erweiterbar bleibt. Die ursprüngliche Darstellung von Kruchten ist auf eine spezifische IT-Architektur ausgerichtet. Abstraktion ist in diesem Modell nicht vorgesehen. Aus diesem Grund werden die Sichten von Kruchten dahingehend angepasst, dass die Modellierung einer abstrakten, technologieneutralen Referenzarchitektur möglich ist. Das Ergebnis der Anpassung wird aus Reidt et al. (2016a) entnommen und ist in Abbildung 14 zu erkennen. Dort sind die einzelnen Sichten und deren Verbindungen zueinander abgebildet. Besteht eine Verbindung zwischen zwei Sichten, so sagt dies aus, dass eine Sicht auf den Informationen der anderen aufbaut. Die Darstellung der einzelnen Sichten fokussieren einen jeweiligen Aspekt eines IIS, sind jedoch im Bezug der Darstellung und ihren Querverweisen so miteinander verbunden, dass von einer Integriertheit gesprochen werden kann, die sich an die Integriertheit von Referenzmodellen anlehnt (vgl. dazu vom Brocke, 2015, S. 108). Nachfolgend werden die Sichten und deren Auswahl beschrieben.

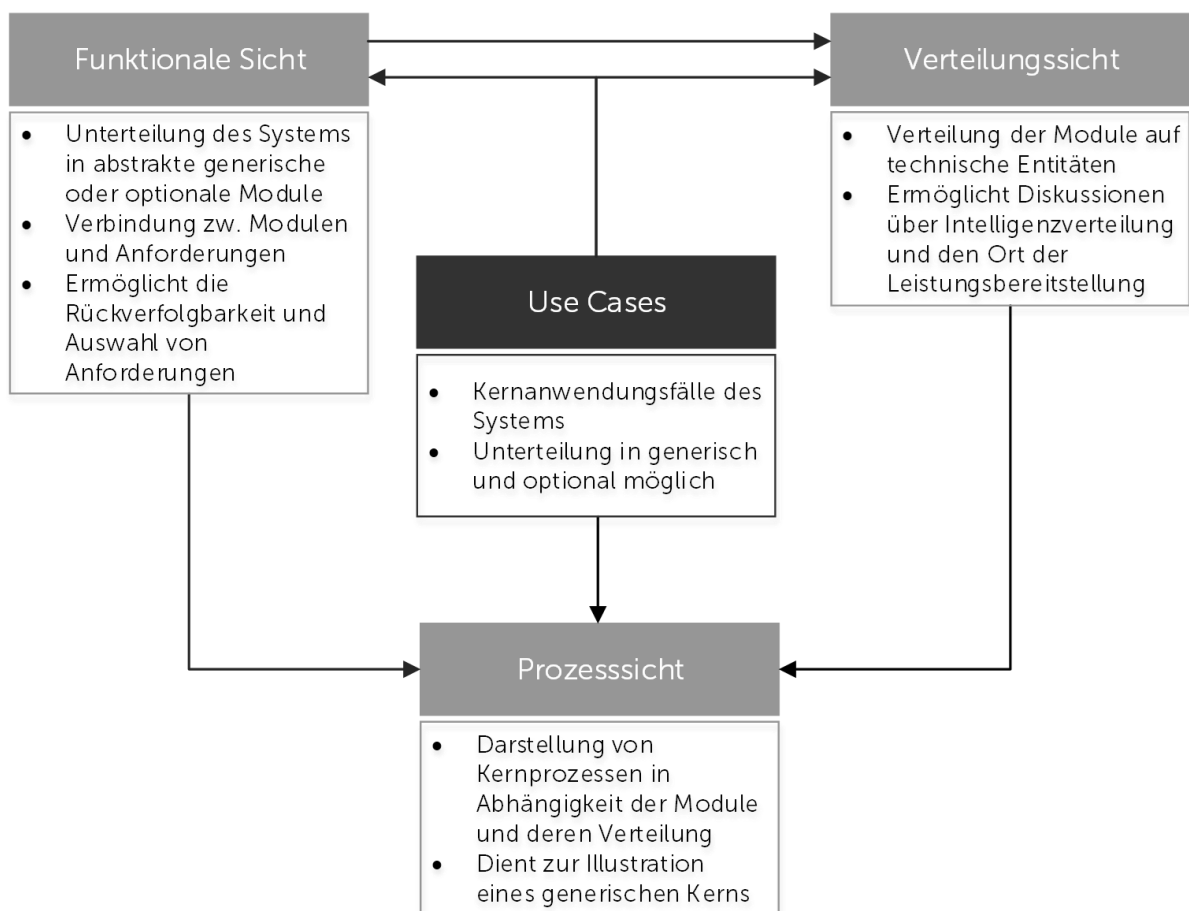


Abbildung 14: Darstellung des Sichtenmodells der RAI

Quelle: In Anlehnung an Reidt et al. (2017b, S. 49)

Wie im vorherigen Abschnitt dargelegt, ist die logische Sicht im ursprünglichen Modell von Kruchten sehr detailliert dargestellt. Dort werden die funktionellen Anforderungen in Klassen

oder Objekte heruntergebrochen und miteinander verbunden und/oder in Vererbungshierarchien gebracht. Die Entwicklungssicht teilt die Klassen in Module, Subsysteme und verschiedene Schichten ein, die über spezifische Schnittstellen miteinander kommunizieren müssen. Diese Darstellung ist für eine Referenzarchitektur zu spezifisch, da konkrete Klassen, Schnittstellen und Subsysteme gegenläufig zu einer hohen Abstraktion und einer generischen Sichtweise wären. Eine individuelle Adaptierung auf die eigenen technischen Gegebenheiten und Systeme wäre demzufolge kaum möglich.

5.3.1 Funktionale Sicht

Daher wird die ursprüngliche logische Sicht dahingehend angepasst, dass sie den logischen Aufbau und die Funktionsweise der RAI in abstrakter, möglichst technologieneutraler Weise beschreibt. Dazu wird die gesamte Architektur bzw. das komplette integrierte Informationssystem zur Unterstützung der Instandhaltung in einzelne Bausteine, Module und Anforderungen untergliedert. Der Kern dieser Sicht bezieht sich auf die Module, die Beschreibung der Funktionalität dieser Module und die Darstellung der Verbindungen der Module zueinander. Module sind in dieser Sicht nicht wie bei Kruchten Elemente, die verschiedene Klassen oder Objekte enthalten. Module stellen im Kontext der RAI eine logische/technische Funktionseinheit dar und entsprechen eher einem Modul im Sinne der ursprünglichen Entwicklungssicht. Jedoch sind sie auf einem deutlich höheren Abstraktionsniveau und enthalten daher keine konkreten Klassen, Verweise auf technische Bibliotheken oder Vererbungshierarchien. Sie stellen vielmehr eine abstrakte Funktionsbeschreibung einer logischen Einheit dar, die jeweils aus einer bestimmten Menge an unternehmensspezifischen, optionalen und generischen Anforderungen aus Kapitel 4 gebildet wird.

Diese Anforderungen stellen die Basis für den Rahmen und den Umfang der logischen Einheit dar. Dies ermöglicht eine einfache Darstellung von nötigen und optionalen Funktionen in Verbindung mit generischen und optionalen Anforderungen. Durch diese nachvollziehbare Verbindung von Modul und Anforderung wird die individuelle Auswahl der Module anhand eigener Anforderungen sowie eine transparente Rückverfolgbarkeit dieser bei Änderungen erreicht. Wird ein Modul umgesetzt, so werden nach Logik der RAI die jeweilig enthaltenen Anforderungen erfüllt bzw. können erfüllt werden, wenn diese optional sind.

Die Wiederverwendbarkeit und die Grundlage der Module als Nutzung als eine Art Blaupause eines bestimmten Teils des Instandhaltungssystems werden weiter mit dieser Art der Darstellung angestrebt. Der modulare Aufbau der RAI und dieser Sicht orientiert sich weiter an dem modularen Aufbau von Referenzmodellen nach Schwegmann (1999, S. 63–66). Der Aufbau soll die Referenzarchitektur, ähnlich wie beim Referenzmodellen, zum Zwecke der Komplexitätsreduktion in Teilmodelle zerlegen, hier daher auch Module genannt. Durch die Modularität bzw. den modularen Aufbau wird die Wiederverwendbarkeit erhöht und dem Grundsatz der Klarheit wird Rechnung getragen. Denn der Inhalt eines Moduls bzw. einzelner Teile eines Systems sind so leichter zu erfassen. Die Aspekte Anpassbarkeit und Erweiterbarkeit, die für Referenzmodelle von hoher Relevanz sind, werden hier analog durch Darstellung von Modulen und deren angestrebter hoher Kohäsion zu erreichen versucht. Zwar wird ähnlich wie bei Schwegmann eine Wohldefiniertheit eines Moduls angestrebt, d. h. ein Modul soll eine inhaltlich exakt abgegrenzte Aufgabe vollständig erfüllen. Jedoch wird dies bei der RAI teilweise dahingehend aufgebrochen, dass Module zusammen eine Aufgabe erfüllen

bzw. in Interaktion stehen. Dies liegt in Einzelfällen daran, dass Module eines Bausteins so stark modularisiert wurden, dass bestimmte optionale Funktionen bzw. einzelne komplexe Funktionalitäten ausgelagert wurden, die jedoch zwingend auf Daten anderer Module angewiesen sind. Die Wohldefiniertheit wird daher in bestimmten Aspekten nicht komplett erfüllt. Die Module der RAI sind zudem durch ihre Abstraktheit und ihre fehlenden Schnittstellenbeschreibungen näher zum Modulbegriff der Referenzmodellierung denn der Moduldefinition im Bereich des Software Engineerings einzuordnen (vgl. dazu Schwegmann, 1999, S. 63–67).

Weiterhin stellen die Module auch Muster im Sinne von Fowler dar, der ein Muster als eine Idee ansieht, die nützlich in einem praktischen Kontext ist und wahrscheinlich auch in einem anderen sein wird (Fowler, 2010). Dabei unterscheidet sich die hier gewählte Form des Moduls von den typischen Entwurfsmustern von Gamma et al. (1995), die auch als die „Gang of Four“ bezeichnet werden. Die Differenz entsteht u. a. dahingehend, dass die hier vorgestellten Module wiederverwendbare Teile eines Systems darstellen, jedoch nicht die Problemlösung eines konkreten Problems so adressieren, dass sie eine bestimmte Lösung vorschlagen. Die Module besitzen demnach einen anderen Fokus und eine deutlich abstraktere Art der Darstellung. Von daher wird im Kontext der Arbeit auch nicht von Mustern oder Entwurfsmustern von Instandhaltungssystem gesprochen, da die Module nur eine sehr weite Definition von Mustern im Sinne von Fowler erfüllen. Es wurde sich bei der tabellarischen Beschreibung der Module dennoch an die zwei Kernkomponenten von Mustern von Fowler (2003, S. 56) gehalten: Es wird teilweise auf abstrakte Weise beschrieben, wie die einzelnen Module implementiert werden könnten und erläutert, wann diese benutzt werden sollen.

Darstellung der Funktionalen Sicht

Es existieren zwei Darstellungsarten für die einzelnen Module: Zum einen eine vor allem visuelle, komprimierte Version des Moduls, zum anderen eine ausführlichere, die auf Details des Moduls eingeht. Durch diese Zweiteilung wird der Zugang zu den einzelnen Modulen erleichtert, da nur bei Bedarf komplexere Informationen bereitgestellt werden. In der visuellen Form wird der Aufbau eines solchen Moduls wie in Abbildung 15 dargestellt.

Das äußere Rechteck stellt einen übergeordneten Baustein dar. Dieser umfasst mehrere Module, falls deren Funktion bzw. deren Datenaustausch miteinander essentiell ist und/oder die Funktionen der Module in enger thematischer Nähe zueinanderstehen. Dabei sind die einzelnen Module in dem Baustein so eigenständig, dass sie eine eigene Funktion umfassen und daher nicht in andere Module übergehen. Die Zuordnung zu einem Baustein ist nicht bei allen Modulen vorhanden, da einige Module nahezu autark sind oder sich thematisch nicht einem einzelnen Baustein zuordnen lassen. In Hellgrau ist das eigentliche Modul mit einer eindeutigen Nummerierung dargestellt (hier M.29. QR-CODE auslesen). Innerhalb dieses Kastens befinden sich die Anforderungen, die durch das Modul erfüllt werden können. Dabei sind generische Anforderungen in grün gekennzeichnet und stellen aufgrund ihres Referenzcharakters die vordergründig zu erfüllenden Anforderungen dar. Des Weiteren werden die einzelnen Anforderungen der Unternehmen U1-U4 genannt und in unterschiedlichen Farben dargestellt. Diese sind optional, sofern sie nicht durch die jeweiligen generischen Anforderungen ebenso abgedeckt werden.

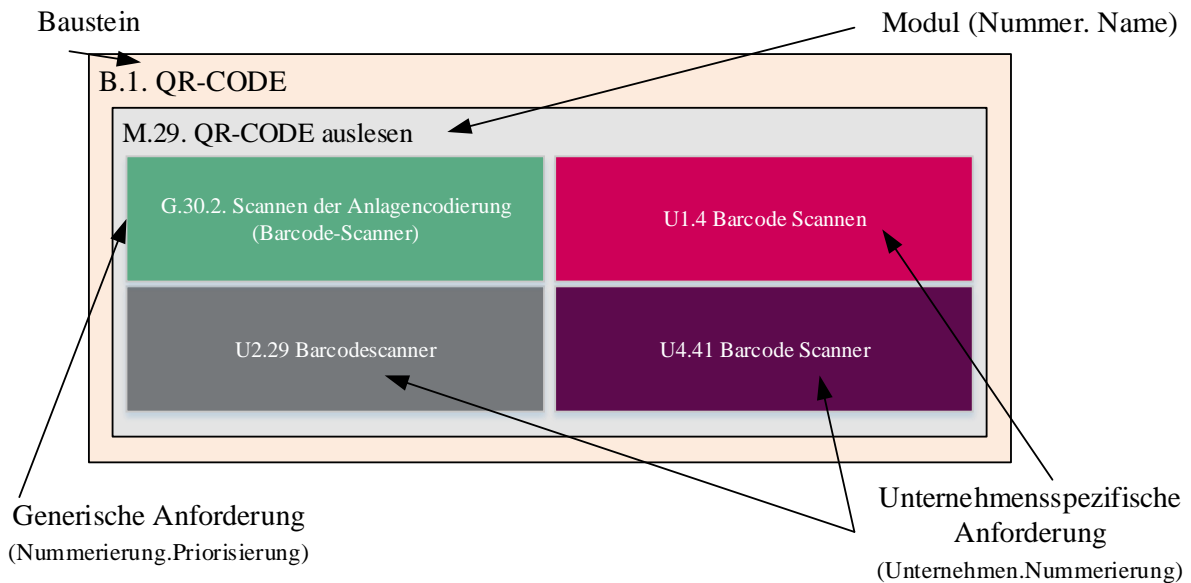


Abbildung 15: Erklärung Aufbau der Module

Quelle: In Anlehnung an Reidt et al. (2016a, S. 52)

Die gewählte einfache schematische Darstellung ermöglicht eine Verständlichkeit über alle Stakeholder eines solchen Projektes hinweg, da kein Expertenwissen nötig ist, um den Zusammenhang zwischen Modulen und Anforderungen zu erkennen und die Funktionsweise zu verstehen. Das Schema der Nummerierungen, Unternehmen und Farbgebung der einzelnen Elemente werden in der nachfolgenden Abbildung erklärt.






<u>Legende</u> <u>Farbschema</u> <u>Anforderung</u>	<u>Nummerierung und Priorisierung</u> <u>Anforderungen</u>	<u>Weitere Abkürzungen</u>
 U1	Generische Anforderung (z. B. G.1.1.) G = Generisch Erste Zahl = Nummer d. Anforderung Zweite Zahl = Priorisierung d. Anforderung	Modul M. Nummerierung des Moduls M.1 für Modul 1 Entität E. Nummerierung der Entität E.1 für Entität 1 Baustein B.
 U2		
 U3	Unternehmensspezifische Anforderung (U1.1.) UX = Unternehmensnummer Zweite Zahl Nummer d. Anforderung	Nummerierung des Bausteines B.1 für Baustein 1
 U4		
 Generisch		

Abbildung 16: Legende simplifizierte Moduldarstellung

Quelle: In Anlehnung an Reidt et al. (2016a, S. 53)

Die Anforderungen sind dort nach den jeweiligen Unternehmen U1-U4 und nach generischen Anforderungen aufgeteilt. Die Nummerierung und Priorisierung wird anhand der Bedeutung der einzelnen Zahlen erläutert. Hier wird zwischen der Nummerierung einer generischen Anforderung und einer unternehmensspezifischen bzw. optionalen Anforderung unterschieden. Letztere werden ohne Priorisierung dargestellt. Daneben werden die Abkürzungen für Modul, Entität (z. B. mobiles Endgerät, Backend oder Maschine) und Baustein dargelegt.

Die detaillierte Darstellung eines Moduls erfolgt tabellarisch, wobei die visuelle Darstellung zur besseren Übersicht als ein Element in dieser integriert ist. In Tabelle 32 werden diese Art der Darstellung und die einzelnen Kategorien der Tabelle erläutert. Die Beschreibung ist dabei

leicht angelehnt an die Beschreibung von Muster, die bei Gamma et al. (1995) zu finden ist. Weiter wurde sie während der Entwicklung mit den entsprechenden Praxispartnern abgestimmt, sodass es auch möglich ist, verschiedene Informationen aus anderen Sichten in diese tabellarische Darstellung zu integrieren.

Die nachfolgenden Zeilen enthalten daher den Namen, das Problem bzw. die Funktion und ggfs. eine abstrakte Lösung, die mit dem Modul angestrebt werden soll. Dabei ist die Lösung deutlich abstrakter als die bei Mustern benutzte Darstellung von Lösungen, da bei der RAI die Lösung möglichst offen und technologie-neutral gehalten werden soll. Weiter gibt die RAI auch kein Architekturkonzept vor oder schreibt bestimmte Lösungswege vor. Es wird nicht gesagt, ob die Architektur ereignisgesteuert oder monolithisch sein soll. Dies kann später frei und je nach Bedarf entschieden werden. Die Lösung wird daher abstrakt gemeinsam mit dem Problem durch die Zeilen der Tabelle Zweck, Funktionsweise sowie Unterfunktionen beschrieben. Die Art der Unterteilung wurde gewählt, um eine Kurzbeschreibung des Moduls anhand einer kurzen Erläuterung durch den Zweck zu besitzen und eine umfassendere durch die Funktionsbeschreibung, welche durch die Nennung von Unterfunktionen noch weiter unterteilt werden kann. Der Kontext und die Bestimmung, wann das Modul nötig ist, wird weiter durch die Spalte generisch/optional gegeben. Zusätzlich werden die Gruppierung und die Verbindung zu anderen Modulen angegeben, wobei die Verbindungen hier nicht näher modelliert werden, sondern nur beschrieben wird, dass eine Verbindung besteht. Weiter sind Spalten zu den Entitäten und den verbundenen Aktivitäten durch das Modul angegeben, wodurch die besagte Verbindung zu anderen Sichten zusätzlich gegeben ist. Diese Verbindungen lassen sich auch durch die anderen Sichten erschließen, sind aber aus Gründen der Übersichtlichkeit in der tabellarischen Form ebenso enthalten.

Die Beschreibung der jeweiligen Funktionen und Module erfolgt in natürlicher Sprache und besitzt keine starr festgelegte Form. Diese Art der Beschreibung erlaubt einen deutlich höheren Freiheitsgrad als formellere Darstellungsarten wie Modellierungssprachen oder Anforderungen in formaler Sprache. Aufgrund des angestrebten umfassenden Anspruchs der Referenzarchitektur und der jeweiligen dafür benötigten Abstraktion wurde sich für diese Art der Informationsdarstellung entschieden. Formellere Arten von Darstellungen hätten eine solch große Komplexität in die Darstellung eingeführt, dass eine Aggregation und die verfolgten Ziele, wie z. B. die Einfachheit in Darstellung und Verständnis, nicht erreicht hätten werden können. Auch ist die Informationsdichte zwischen einzelnen Modulen zum Teil sehr heterogen, wodurch eine einheitliche formelle Art der Darstellung kaum als sinnvoll erachtet wird.

Für den logischen Aufbau der einzelnen Module wird nach dem Prinzip des Separations of Concerns (Laplante, 2007) verfahren, sodass zusammen mit einer angestrebten losen Kopplung eine hohe Modularität der RAI erreicht wird. Das Befolgen dieser Prinzipien innerhalb der RAI führt dazu, dass sich ein einzelnes Modul genau einem „Concern“ bzw. einer Angelegenheit widmet und die einzelnen Module untereinander eine geringe Abhängigkeit besitzen. Weiterhin resultiert aus diesem Aufbau, dass die RAI durch neue Module leicht erweiterbar ist, ohne dass sich der Grundaufbau ändert. Hierdurch wird eine hohe Verständlichkeit, Anpassbarkeit und Zukunftssicherheit erreicht.

Komponente	In dieser Sektion wird der Name der Komponente eingetragen
Darstellung in Referenzarchitektur	In dieser Sektion wird der vergrößerte Ausschnitt, welcher das Modul beschreibt, aus Abbildung 19 übernommen. Dadurch ist klar erkennbar, welche Anforderungen dem Modul zugeordnet sind und wie sich dieses in die Gesamtarchitektur einbettet.
Zweck/Ziel	Hier wird das Ziel, das mit dem Modul verfolgt wird bzw. der Zweck des Moduls, beschrieben.
Generisch/optional	Dieses Feld beschreibt, ob das Modul als generisch oder als optional zu betrachten ist.
Funktionsbeschreibung	Die Funktionsbeschreibung des Moduls wird in dieser Sektion gegeben. Unter Funktionsbeschreibung wird hier das allgemeine Vorgehen zur Erfüllung des Ziels bzw. des Zweckes verstanden. Es können ebenso auf spezifische Herausforderungen während der Implementierung hingewiesen und gegebenenfalls deren Lösung skizziert werden.
Mögliche Unterfunktionen	Eine Aufteilungen des Moduls in Unterfunktionen wird hier anhand der Funktionsnennung aufgelistet.
Verbindungen zu anderen Modulen	Hier werden die Module genannt mit welchen das vorliegende Modul im Austausch steht bzw. zusammen einen Baustein bildet. Zuerst werden die Module im eigenen Baustein genannt und folgend die außerhalb. <i>Baustein in dem sich das Modul befindet:</i> <ul style="list-style-type: none"> • Module, die sich ebenso in diesem Baustein befinden, zu denen das vorliegende Modul Verbindungen besitzt. <i>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</i> <ul style="list-style-type: none"> • Module, die sich nicht in dem gleichen Baustein befinden und Verbindungen zu dem vorliegenden Modul besitzen.
Verbundene Aktivitäten	Hier werden die jeweiligen Aktivitäten genannt, in denen das Modul vorkommt.
Aufteilung nach Entität	Die Aufteilung des Moduls auf Entitäten wird hier erklärt. Eine komplette Darstellung mit allen Modulen und deren Aufteilung ist darüber hinaus in Kapitel 6.3 zu finden.

Tabelle 32 : Tabellarischer Aufbau eines Moduls der RAI

Quelle: In Anlehnung an Reidt et al. (2016a, S. 54f)

Die Module bilden in der RAI die Basis für eine einheitliche Domänensprache eines solchen Informationssystems und stellen in ihrer Ausprägung die Grundlage für die weitere Entwicklung dar. So bieten die Module ein Fundament für die spätere Implementierung, ohne jedoch entscheidende Freiheitsgrade bei dieser einzuschränken. Der Name dieser Sicht ist demgemäß "Funktionale Sicht", auf eine isolierte Entwicklungssicht wird komplett verzichtet.

5.3.2 Verteilungssicht

Weitergehend wird die physische Sicht stark angepasst. Sie wird ursprünglich genutzt, um den physischen Aufbau mitsamt Hardwareallokation so zu beschreiben, dass die Architektur nicht-funktionale Anforderungen wie Performance oder die Verfügbarkeit erfüllen kann. Da der

physische Aufbau von der Wahl der Module, deren Implementierung und der Größe der Applikationen im Allgemeinen abhängt, wäre eine solche Darstellung nicht zielführend. Sie wird im Kontext der Referenzarchitektur daher zu einer Verteilungssicht umgewandelt und zeigt auf, welche Module aus der Funktionalen Sicht auf welchen Entitäten (Backend, mobiles Endgerät, Maschine) verfügbar sein müssen. Der Fokus liegt also auf der abstrakten Verteilung von Komponenten und Funktionen. Dies hat den Vorteil, dass die Stakeholder der Entwicklung einfach erkennen können, welche Funktionen zentral, dezentral oder mobil verfügbar sind. Hierdurch lässt sich die Intelligenzverteilung zwischen Backend, mobilem Endgerät und Anlage klar strukturieren und diskutieren. Die Frage nach der Intelligenzverteilung stellt einen enorm wichtigen Punkt bei der Entwicklung von Industrie 4.0 Applikationen dar, welche die Fähigkeiten von CPS effizient ausnutzen wollen.

Die Darstellung der Sicht ist dabei stark simplifiziert: Die einzelnen Module aus der Funktionalen Sicht werden als Rechtecke mit der jeweiligen Bezeichnung aus der Funktionalen Sicht einer Entität zugeordnet, die wiederum aus einem farbig markierten Rechteck besteht. Eine Entität ist dabei das jeweilige System, auf dem diese Funktion oder Teile der Funktion vorhanden sein sollen. Ein Modul kann dabei mehreren Entitäten zugeordnet werden, wenn es auf mehreren Entitäten vorhanden sein muss oder Teile des Moduls auf der einen und andere Teile des Moduls auf einer anderen Entität laufen müssen.

5.3.3 Use Case Sicht

Die Use Case Sicht ist im Kontext der RAII ähnlich umgesetzt worden wie die Szenarien von Kruchten. Die dort enthaltenen Anwendungsfälle beschreiben die wichtigsten Aspekte der Architektur und dienen dazu, Architekturelemente der anderen Sichten zu identifizieren. Für die hier erstellte Referenzarchitektur ist es jedoch vonnöten, unterschiedliche Use Cases von unterschiedlichen Unternehmenstypen abzubilden, um alle Aspekte eines IIS hinsichtlich generischer und optionaler Elementen darzustellen. Daher sind z. T. ähnliche Use Cases verschiedener Unternehmen vorhanden. Die Darstellung der Use Cases orientiert sich dabei an der textuellen Vorgabe von Cockburn (2002).

5.3.4 Prozesssicht

Diese Use Cases bilden auch die Grundlage für die Modellierung in der Prozesssicht. Dort werden sie per UML-Aktivitätsdiagramm so modelliert, dass eine Verbindung zwischen den Use Cases, Funktionaler Sicht und Verteilungssicht einfach ersichtlich ist. Dies geschieht für alle Use Cases. Anhand dieser Informationen werden weitere generische und optionale Teile von Prozessen der RAII identifiziert. Im Falle der RAII wird ebenso eine Grundaktivität identifiziert, welche die grundlegende Funktionsweise eines IIS umspannt und die Use Cases vereinheitlicht. In dieser Grundaktivität werden die identifizierten generischen und optionalen Teile der Prozesse aggregiert und mit den übrigen Sichten verbunden. Hierdurch wird eine weitere Abstraktionsebene erreicht.

Mit dieser Form der Darstellung unterscheidet sich die Prozesssicht stark von der bei Kruchten verwendeten, da ein höherer Abstraktionsgrad und eine zweite Form der Verknüpfung der übrigen Sichten erreicht wird. Sie stellt damit die generische Überführung der Use Cases mitsamt Verknüpfung zu den restlichen Sichten dar und ist die Basis für eine iterative Weiterentwicklung der Architektur. Weiterhin wird anhand dieser Sicht das Wissen über die Gesamtarchitektur punktgenau dargelegt, sodass eine Evaluation der restlichen Sichten durch

das Wechselspiel erreicht werden kann. Im Kontext der RAII wird insbesondere durch diese Sicht die Kommunikation zwischen allen Stakeholdern ermöglicht.

Bei der Darstellung dieser Sicht orientiert sich die RAII weitestgehend am Standard für UML-Aktivitätsdiagramme, jedoch werden komplexere Notationen aus dieser vermieden, Teile erweitert und vereinfacht. Dies hat zum Ziel, eine bessere Verständlichkeit zu erreichen und Informationen der RAII gezielter und aggregiert darzustellen. Aus den Aktivitäten sollte hervorgehen, zu welchen Modulen sie gehören und auf welcher Entität sie ausgeführt werden. Die angepassten Elemente sind in Abbildung 17 exemplarisch dargestellt.

Aktivitätsname (Legende)

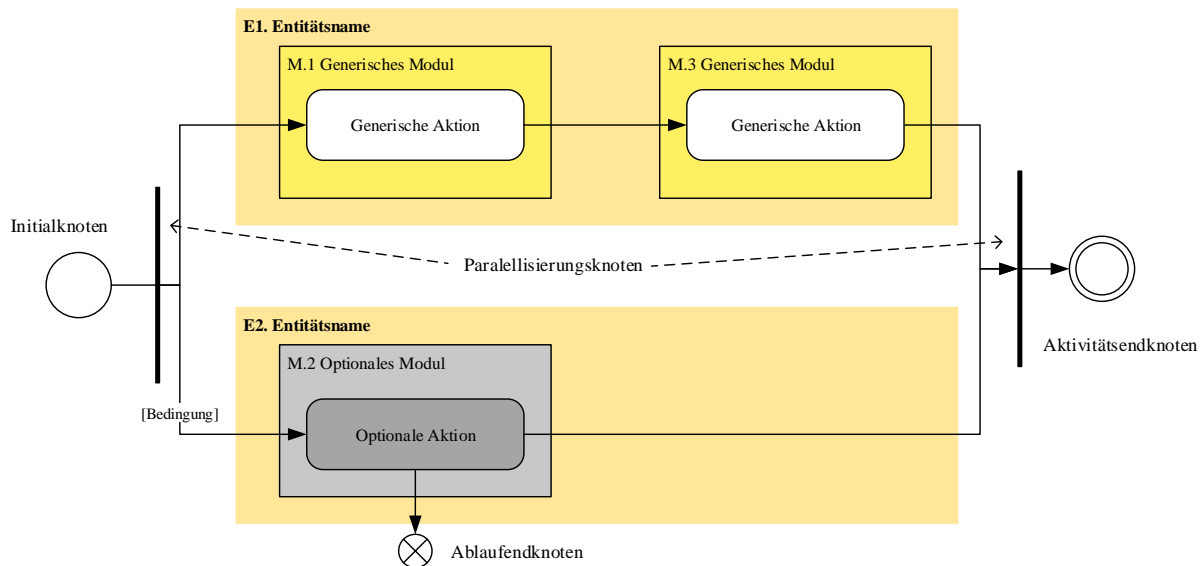


Abbildung 17: Darstellung Prozesssicht – Veränderte Darstellungselemente

Quelle: Eigene Darstellung

Folgende Anpassungen sind für die Darstellung der Prozesssicht in der RAII nötig:

- Auf die Darstellung einer Aktivität und der typischen umschließenden Rahmen diese wird verzichtet. Stattdessen wird nur ein Aktivitätsname angegeben. Die Aktivität umspannt immer das komplette Aktivitätsdiagramm. Weiter wird auf die Darstellung von Parameter, Objektknoten und Datenknoten verzichtet.
- Das Rechteck um eine Aktion gibt an, zu welchem Modul sie gehört. Durch diese Art der Darstellung ist die Zugehörigkeit einer Aktion zum jeweiligen Modul ersichtlich und die Verbindung zu der Funktionalen Sicht wird plastisch dargestellt.
- Die großen Rechtecke, die mehrere Module und Aktionen umfassen, sind Entitäten, die den Ort des jeweiligen Moduls angeben. Durch diese Visualisierung ist die Verbindung zu der Verteilungssicht gewahrt und Aktionen zwischen mehreren Entitäten leicht darstellbar.
- An einigen Punkten wird auf Entscheidungsknoten verzichtet, falls diese solange eine Schleife nach sich ziehen würden, bis eine bestimmten Bedingung erfüllt wird. Aus Gründen der Übersichtlichkeit wird in diesen Fällen darauf verzichtet, da diese Bedingung sich aus dem jeweiligen Kontext ergeben wird. Weiter wird nach einem öffnenden Parallelisierungsknoten als Vereinfachung eine Bedingung ohne Entscheidungsknoten für den Fall verwendet, dass diese Bedingung eine optionale Aktion nach sich zieht.

- Graue Rechtecke stellen entweder eine optionale Aktion oder ein komplett optionales Modul dar. Eine Aktion kann dabei in einem generischen Modul optional sein, jedoch kann es niemals eine generische Aktion in einem optionalen Modul geben.

5.3.5 Weitere Besonderheiten bei der Darstellung

Variationen

Eine weitere Änderung ist die Einführung der Darstellung von Variabilität. Diese Art der Darstellung ist Teil vieler Referenzarchitekturen und Produktlinienarchitekturen (vgl. Kapitel 2). Diese erleichtert die Darstellung von optionalen Komponenten einer Architektur und zeigt potentielle Architekturentscheidungen auf. Hierdurch wird die Erweiterbarkeit und die Zukunftssicherheit der Referenzarchitektur gesichert. Die in dieser Arbeit behandelte Referenzarchitektur beinhaltet laut Klassifikation Variationspunkte, die an die jeweilige Art der Referenzarchitektur und ihrer Darstellung angepasst werden, da keine Einheitlichkeit hinsichtlich der Darstellung und der Einordnung von Variationen existiert, die für jede Art von Architektur passend wäre (Nakagawa et al., 2014a, S. 151). Die Art der Variationen sind ferner vergleichbar mit den Anpassungen von konfigurierbaren Referenzmodellen, die Becker et al. (2007) behandeln. Jedoch wird nicht mit Perspektiven gearbeitet, sondern mit vereinfachter integrierter Darstellung, da dies stärker den Ansprüchen des modularisierten und technischeren Aufbaus von Referenzarchitekturen entspricht. Zusätzlich wird durch die hier verwendete Modellierung die Verständlichkeit des generischen Kerns eines Instandhaltungssystems betont.

Die Variationen sind bei der RAI daher möglichst einfach und intuitiv verständlich in den jeweiligen Sichten dargestellt. Sie äußern sich dabei wie folgt:

- In der Funktionalen Sicht sind die Variationspunkte durch die jeweiligen Module beschrieben. Neben den Anforderungen sind auch Module der RAI entweder generisch oder optional. Generisch bedeutet, dass diese Module bei der Implementierung eines IIS zwangsläufig vorhanden sein müssen. Sind Module hingegen optional, so stellen sie einen möglichen Variationspunkt dar. Optional sind Module, wenn sie nur für bestimmte Unternehmensgruppen oder Anwendungsfälle zutreffen, eine geringe Priorität besitzen oder nur einen Zusatznutzen darstellen. Diese Einteilung wird anhand der Verteilung von generischen und optionalen Anforderungen innerhalb des Moduls, deren Priorisierung und einer Einschätzung der Umsetzbarkeit des Moduls vorgenommen. Neben der Optionalität eines gesamten Moduls können auch generische Module optionale Aspekte besitzen. Diese werden durch einzelne optionale Anforderungen innerhalb eines generischen Moduls abgebildet. Dort sind sie beheimatet, da sie nicht isoliert von der ursprünglich generischen Funktion betrachtet werden können. Bei der späteren Entwicklung können sie allerdings je nach Bedarf eingebunden werden. Dadurch wird ersichtlich, welche Module und Anforderungen in einem zu entwickelnden IIS für die Instandhaltung enthalten sein müssen und welche optional – je nach Anforderung beim zugrundeliegenden Fall – in das System integriert werden müssen. Wird dieser Variationspunkt bei der Erstellung von konkreten Architekturen ausgestaltet, so können durch die beschriebenen Verbindungen in den jeweiligen Modulen die nötigen Stellen identifiziert werden, die eine Anpassung benötigen.
- In der Prozesssicht sind Variationspunkte innerhalb eines Prozesses beschrieben und werden, wie in der Legende im vorherigen Kapitel dargestellt, als graue Elemente

gekennzeichnet. Sind Elemente als optional deklariert bzw. grafisch in Grau gehalten, so kann dieser Prozessschritt in Abhängigkeit von der Implementierung des jeweiligen optionalen Moduls getätigt werden.

- Die Verteilungssicht stellt einen Empfehlungscharakter dar und durch die Beschreibung wird ersichtlich, dass die Grenzen zwischen den Endgeräten bei Bedarf aufgehoben werden können. Variabilität ist daher direkt in die Art der Darstellung integriert. Auch hier gilt jedoch, dass die aufgezeigten Module, je nachdem ob sie generisch oder optional sind, ebenso Variationspunkte darstellen.

Integrationsmöglichkeiten

Die Integrationsmöglichkeiten von bestehenden Systemen, welche in Kapitel 3 vorgestellt wurden, sind nur teilweise direkt in der Referenzarchitektur enthalten. Da die Existenz dieser Systeme und deren Ausprägung stark individuell ist, wird deren Integration nur textuell innerhalb der relevanten Module beschrieben und durch bestimmte Module wie die Pluginintegration bedacht. Jedoch können auch relevante Teile der Referenzarchitektur und damit generische wie auch optionale Module direkt durch andere Systeme ersetzt und somit integriert werden. Hierbei gilt es jedoch zu beachten, dass das ganzheitliche Konzept trotz dieser Integration weiterverfolgt wird.

Anforderungen und Domänenwissen

Neben der Darstellung der jeweiligen Sichten, welchen in Kapitel 6 inhaltlich vorgestellt werden, bilden noch weitere Elemente einen Teil der Referenzarchitektur. Dies wären zum einen die Anforderungen aus Kapitel 4 und zum anderen das Domänenwissen aus Kapitel 3. Beide Elemente stellen wesentliche Bestandteile der Referenzarchitektur dar. Sie sind textuell Bestandteil der in dieser Arbeit vorliegenden Form der Referenzarchitektur oder können bei Reidt et al. (2016a) nachgelesen werden. Beide Elemente werden in der Darstellung der Module behandelt, sind jedoch aufgrund der Komplexität ihrer Beschreibung und der angestrebten Unabhängigkeit von den jeweiligen Instandhaltungssystemen nicht direkt in die Sichten integriert worden.

5.4 Zusammenfassung

In diesem Kapitel wurde die Darstellungsart der RAII vorgestellt. Zuerst wurden die Konzepte präsentiert, auf denen die Darstellung basiert und erläutert, wie diese im Zusammenhang mit der in Kapitel 2 erstellten Klassifikation stehen. Dabei wurde die Darstellungsart des 4+1 Sichtenkonzept nach Kruchten vorgestellt. Aufgrund der Art der hier vorgestellten Referenzarchitektur und der damit verbundenen Ziele wurden jedoch Änderungen am ursprünglichen Sichtenkonzept vorgenommen, sodass nur 3+1 Sichten in dieser Arbeit verwendet und inhaltlich und darstellerisch auf die Bedürfnisse der hier vorliegenden Referenzarchitektur angepasst wurden. Diese Änderungen hängen mit der Abstraktheit und den potentiellen Variationspunkten der Referenzarchitektur zusammen, die ebenso erläutert wurden. Die hier vorgestellten Informationen dienen damit als Grundlage zum Verständnis des Inhalts und der Darstellung der im nächsten Kapitel vorgestellten einzelnen Sichten der RAII.

6 Referenzarchitektur RAI

6.1 Einleitung

Nachdem das Konzept der Darstellung und der einzelnen Sichten im vorherigen Kapitel dargelegt worden ist, wird aufbauend auf diesem Wissen in diesem Kapitel die RAI mitsamt ihrer 3+1 Sichten vorgestellt. Die Inhalte dieser vier Sichten werden nacheinander dargestellt und in Verbindung zueinander gesetzt. Die hier dargestellten Informationen sind eine erweiterte Zusammenfassung der in Reidt et al. (2016a) vorgestellten Referenzarchitektur. Zur sprachlichen Vereinfachung wird im Folgenden von einem IIS gesprochen, als ob es eine zu entwickelnde Applikation darstellt.

Das Kapitel ist nach den 3+1 Sichten unterteilt: In Kapitel 6.2 wird die Funktionale Sicht vorgestellt, welche die einzelnen logischen Einheiten eines IIS beinhaltet. Anschließend wird in Kapitel 6.3 die Verteilungssicht dargestellt, welche die Intelligenzverteilung der Module auf verschiedene Entitäten beschreibt. In Kapitel 6.4 wird die Prozesssicht vorgestellt. Diese enthält die wichtigsten Prozesse der RAI, angelehnt an die Use Cases aus der Use Case Sicht, sowie eine generische Grundaktivität, die den Kern eines IIS darstellt. Anhand vereinfachter UML Aktivitätsdiagramme werden die Prozesse der RAI mit den jeweiligen Modulen und Entitäten aus den anderen Sichten verbunden. In Kapitel 6.5 wird schließlich die Use Case Sicht dargestellt, in der die wichtigsten Use Cases des IIS enthalten sind. Nach der Vorstellung der einzelnen Sichten wird die RAI, ihre Designentscheidungen während der Entwicklung und potentielle Auswirkungen von anderen Entscheidungen kritisch diskutiert und reflektiert. Diese kritische Würdigung der RAI erfolgt in Kapitel 6.6. Am Ende des Kapitels in 6.7 wird eine Zusammenfassung gegeben.

6.2 Funktionale Sicht

Die Funktionale Sicht beinhaltet insgesamt 40 logische Module, die aus 39 generischen und 31 optionalen Anforderungen bestehen. Einige Module sind in fünf unterschiedlichen Bausteinen eingruppiert. Wie in Kapitel 2.3.2 in Schritt 6 erläutert entstehen die einzelnen Module und ihr Schnitt in einem iterativen Prozess. Die in Kapitel 4.3.3 extrahierten Anforderungen aus der Praxis werden zu logischen Einheiten zusammengefasst, die ein Modul bilden. Dabei wird nach dem Prinzip der Kapselung und der Separations of Concern verfahren. Ähnliche Anforderungen und Module werden nach logischen Gesichtspunkten zusammengefasst. Dies wird solange wiederholt bis Module auch aus einer möglichst generischen technischen Sichtweise so zusammengefasst sind, dass sie einzelne Arbeitspakete bzw. technische Einheiten darstellen können. Die Beschreibung richtet sich dann nach einer abstrahierten technischen Funktionsweise verbunden mit den jeweiligen Anforderungen des Moduls. Dabei kann auch der Fall auftreten, dass ein generisches Modul, optionale Anforderungen beinhaltet, die sich jedoch nicht von dem gesamten Modul trennen lassen. Aus diesem Grund und um eine Aufteilung einer logischen Einheit in kleinere Einheiten zu ermöglichen, wie bspw. Microservices, wird die Funktionsweise, wie in 5.3 beschrieben, auch in Unterfunktionen dargestellt.

Die Besonderheit dieser Module rührt daher, dass sie jeweils Kernfunktionalitäten darstellen, die die validierten Anforderungen aus Kapitel 4.3.3 erfüllen. Dadurch können die generischen Module in nahezu jedem Instandhaltungssystem wiedergefunden werden. Der umfassenden Gültigkeit drückt sich dahingehend aus, dass die Module mit sehr hoher Wahrscheinlichkeit alle Elemente eines Instandhaltungssystems umfassen. Das heißt ein Instandhaltungssystem für Produktionsanlage wird zwingend die Module bzw. die generischen enthalten müssen. Weiter werden durch die optionalen Module alle erkannten Anforderungen abgedeckt, die für bestimmte Sonderfälle gelten, so dass die nachfolgenden Informationen bei der Erstellung jedweder Art von Instandhaltungssystemen für Produktionsanlagen behilflich sein können.

In Abbildung 18 wird die visuelle Gesamtübersicht aller Module und Bausteine der RAII anhand des in Kapitel 5.3 vorgestellten Schemas dargestellt. Die Anforderungen, die die einzelnen Module enthalten werden aus Übersichtlichkeitsgründen in dieser Abbildung nicht dargestellt. Die Module sind hinsichtlich ihrer Klassifikation als generisch oder optional dargestellt. Module in Grün sind generisch, Module in Grau sind optionale Module. Von den dort gezeigten Modulen sind 23 generisch und 17 optional, wobei einzelne Aspekte eines Moduls in unterschiedlicher Weise und teilweise auch optional ausgebaut werden können. Die RAII wird darüber hinaus auf Abbildung 19 komplett dargestellt, das heißt auch mit den jeweiligen Anforderungen. Die Darstellung richtet sich dabei ebenso nach dem in Kapitel 5.3 vorgestellten Schema.

Nach den beiden Grafiken werden in den folgenden Kapiteln, die in den Abbildungen dargestellten Module, einzeln in tabellarischer Darstellungsform beschrieben.

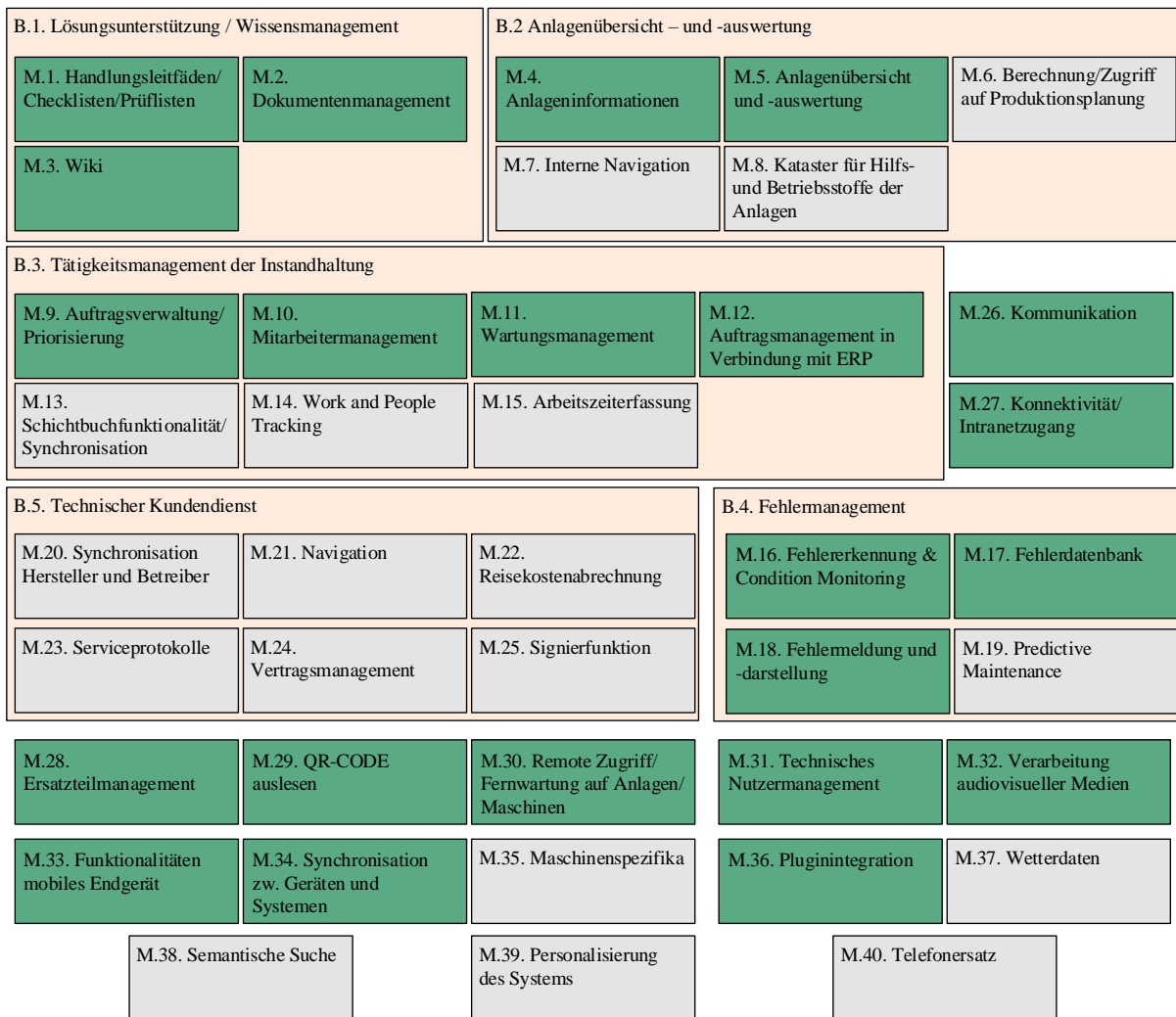


Abbildung 18: Übersicht über generische/optionale Module und Bausteine

Quelle: Eigene Darstellung

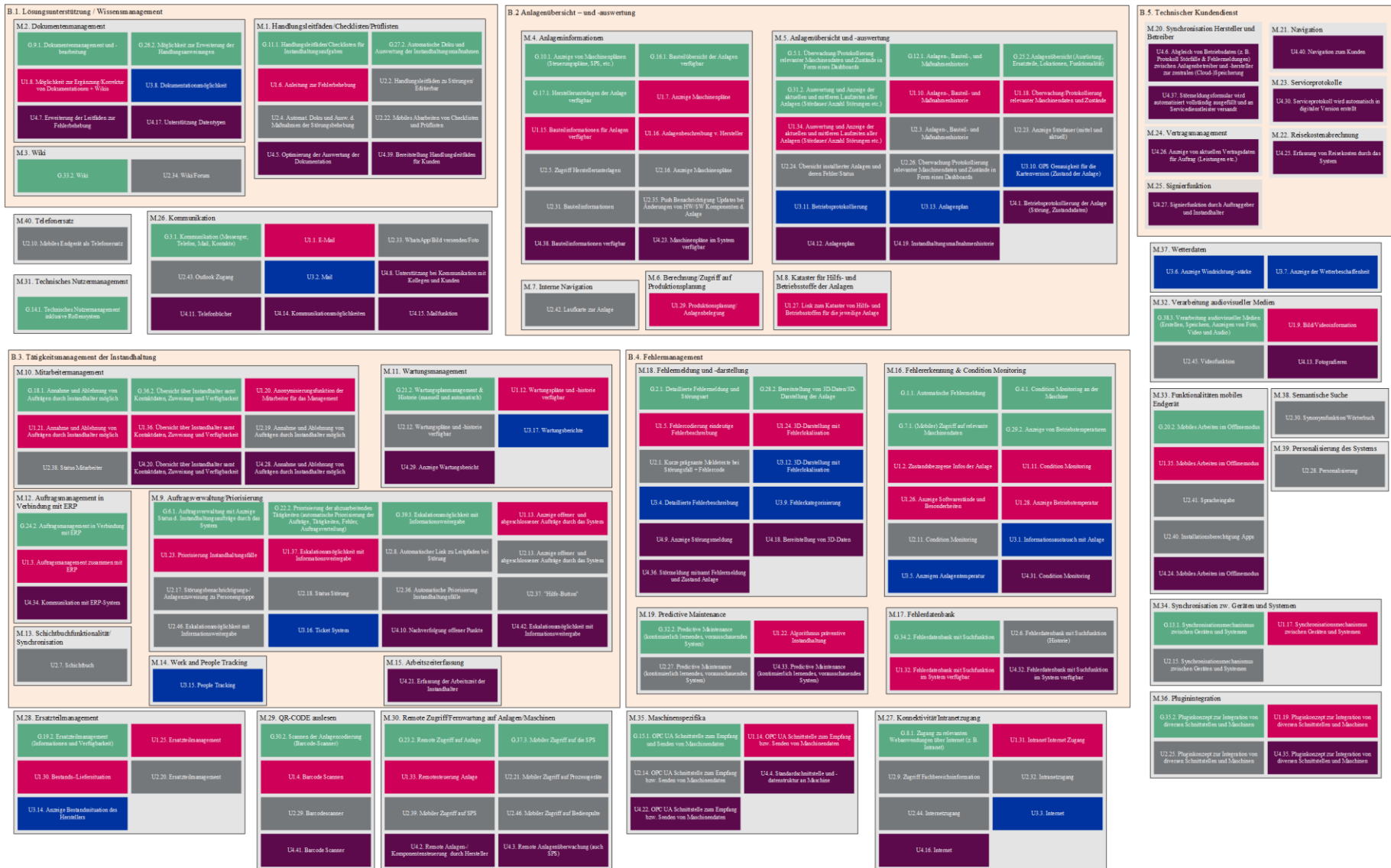


Abbildung 19: Gesamtübersicht Module/Bausteine der RAI
 Quelle: In Anlehnung an Reidt et al. (2016a, S. 54)

6.2.1 M.1. Handlungsleitfäden/Checklisten/Prüflisten

Komponente	M.1. Handlungsleitfäden/Checklisten/Prüflisten										
Darstellung in Referenzarchitektur	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center; padding: 5px;">M.1. Handlungsleitfäden/Checklisten/Prüflisten</td> </tr> <tr> <td style="background-color: #4CAF50; color: white; padding: 5px;">G.11.1. Handlungsleitfäden/Checklisten für Instandhaltungsaufgaben</td> <td style="background-color: #4CAF50; color: white; padding: 5px;">G.27.2. Automatische Doku und Auswertung der Instandhaltungsmaßnahmen</td> </tr> <tr> <td style="background-color: #E91E63; color: white; padding: 5px;">U1.6. Anleitung zur Fehlerbehebung</td> <td style="background-color: #9E9E9E; padding: 5px;">U2.2. Handlungsleitfäden zu Störungen/ Editierbar</td> </tr> <tr> <td style="background-color: #9E9E9E; padding: 5px;">U2.4. Automat. Doku und Ausw. d. Maßnahmen der Störungsbehebung</td> <td style="background-color: #9E9E9E; padding: 5px;">U2.22. Mobiles Abarbeiten von Checklisten und Prüflisten</td> </tr> <tr> <td style="background-color: #8E24AA; color: white; padding: 5px;">U4.5. Optimierung der Auswertung der Dokumentation</td> <td style="background-color: #8E24AA; color: white; padding: 5px;">U4.39. Bereitstellung Handlungsleitfäden für Kunden</td> </tr> </table>	M.1. Handlungsleitfäden/Checklisten/Prüflisten		G.11.1. Handlungsleitfäden/Checklisten für Instandhaltungsaufgaben	G.27.2. Automatische Doku und Auswertung der Instandhaltungsmaßnahmen	U1.6. Anleitung zur Fehlerbehebung	U2.2. Handlungsleitfäden zu Störungen/ Editierbar	U2.4. Automat. Doku und Ausw. d. Maßnahmen der Störungsbehebung	U2.22. Mobiles Abarbeiten von Checklisten und Prüflisten	U4.5. Optimierung der Auswertung der Dokumentation	U4.39. Bereitstellung Handlungsleitfäden für Kunden
M.1. Handlungsleitfäden/Checklisten/Prüflisten											
G.11.1. Handlungsleitfäden/Checklisten für Instandhaltungsaufgaben	G.27.2. Automatische Doku und Auswertung der Instandhaltungsmaßnahmen										
U1.6. Anleitung zur Fehlerbehebung	U2.2. Handlungsleitfäden zu Störungen/ Editierbar										
U2.4. Automat. Doku und Ausw. d. Maßnahmen der Störungsbehebung	U2.22. Mobiles Abarbeiten von Checklisten und Prüflisten										
U4.5. Optimierung der Auswertung der Dokumentation	U4.39. Bereitstellung Handlungsleitfäden für Kunden										
Zweck/Ziel	<p>Das Modul soll dem Instandhalter bei seinen konkreten Tätigkeiten zur Wartung/Reparatur/Fehleranalyse oder ähnlichen Maßnahmen unterstützen. Dies wird durch eine grafische Darstellung von Prüf-, Checklisten und Handlungsleitfäden für die jeweiligen Systeme erreicht. Diese können durch den Instandhalter interaktiv abgearbeitet und protokolliert werden. Diese Maßnahmen, die der Instandhalter durchzuführen hat, werden dadurch einfach verständlich für ihn aufbereitet, sodass die Durchführung effektiv unterstützt wird. Weiterhin kann anhand der Protokollierung der einzelnen Arbeitsschritte eine automatische Dokumentation der durchgeführten Instandhaltungsmaßnahmen generiert werden.</p>										
Generisch/optional	Generisch										
Funktionsbeschreibung	<p>Das Modul beinhaltet Handlungsleitfäden, Prüf- und Checklisten für Instandhaltungsaufträge. Diese Dokumente werden grafisch interaktiv aufbereitet, so dass der Instandhalter optimal unterstützt wird. So werden ihm verschiedene Schritte zur Behebung von Fehlern angezeigt und er kann die Abarbeitung jedes einzelnen Schrittes protokollieren. Die einzelnen Schritte können dabei Erklärungen, 3D-Modelle oder sonstige Medien beinhalten bzw. auf diese verweisen, die zum Verständnis beitragen und die Darstellung erleichtern. Des Weiteren müssen diese Vorgehensweisen in Verbindung mit dem Modul M.2. Dokumentenmanagement in mehreren Fällen veränderbar sein.</p> <p>Diese Fälle wären:</p> <ul style="list-style-type: none"> • Erstellen von Handlungsleitfäden und Checklisten – es muss durch das Modul in Verbindung mit dem Modul M.2. Dokumentenmanagement die Möglichkeit bestehen, zu einem bestimmten Fehler oder Aufgaben eine Prüf-, Checkliste und/oder einen Leitfaden zu erstellen, die mit diesem Fehler bzw. der Aufgabe verbunden ist/sind. • Abändern von bestehendem Handlungsleitfäden/Checklisten – bestehende Inhalte müssen einfach geändert werden können. Entweder direkt beim Bearbeiten eines Auftrages oder zentral nach dem Abarbeiten in Form von Lessons Learned oder ähnlicher Evaluation der angewandten Handlungsleitfäden/Checklisten. Hier könnte bspw. der Instandhalter während der Bearbeitung direkt Fotos hinzufügen oder dies am Ende des Tages zentral bewerkstelligen. Dies ist in Kombination mit dem Dokumentenmanagement zu durchzuführen. 										

	<p>Ein weiterer wichtiger Punkt ist, dass die Inhalte offline verfügbar sein sollten, damit es möglich ist, bestimmte anstehende Aufträge vorzuladen und abrufbar zu machen. Bearbeitungen der Leitfäden/Checklisten müssen hierbei jeweils synchronisiert werden, sodass zentral immer die aktuellste Version verfügbar ist. Gegebenenfalls ist eine Version einzuführen.</p> <p>Abhängig von den gewünschten Darstellungsarten muss es möglich sein, eine Vielzahl von möglichen Inhalten, wie z. B. Videos, 3D Modelle, Handbücher, textuelle Checklisten, interaktive Anleitungen etc. in diesem Modul darzustellen.</p> <p>Voraussetzungen für eine ordnungsgemäße Funktionsweise des Moduls wäre es eine gepflegte Datenbasis und eine leichte Editierbarkeit bei der Umsetzung zu erreichen, die die Akzeptanz bei den Nutzern fördert. Zu jedem Fehler und jeder Maßnahme müssen Inhalte händisch hinterlegt werden und häufig aktualisiert werden. Die grafische Darstellung und Anwendung der Handlungsleitfäden muss hohen ergonomischen Anforderungen entsprechen, damit die Arbeit der Instandhalter erleichtert wird und die Akzeptanz bei den Anwendern hoch ist.</p>
<p>Mögliche Unterfunktionen</p>	<ul style="list-style-type: none"> • Erstellen und Verändern von Handlungsleitfäden/Checklisten/Prüflisten • Darstellung bzw. grafische Aufbereitung von Anweisungen und Listen • Abruf von weitergehenden Informationen zu einzelnen Schritten • Standardvorgehen erstell- und veränderbar für bestimmte Fehlergruppen machen • Protokollierung der einzelnen Schritte • Mechanismus zur automatischen Dokumentation nach Benutzung von Handlungsleitfäden/Checklisten
<p>Verbindungen zu anderen Modulen</p>	<p>Baustein B.1. Lösungsunterstützung/Wissensmanagement:</p> <ul style="list-style-type: none"> • Modul M.2. Dokumentenmanagement – Dieses Modul beinhaltet die Editierfunktion für die hier zugrundeliegenden Inhalte. <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung – Dieses Modul ruft das vorliegende Modul auf. • Modul M.10. Mitarbeitermanagement • Modul M.11. Wartungsmanagement • Modul M.15. Arbeitszeiterfassung • Modul M.16. Fehlererkennung & Condition Monitoring • Modul M.18. Fehlermeldung und -darstellung • Modul M.23. Serviceprotokolle • Modul M.26. Kommunikation • Modul M.28. Ersatzteilmanagement • Modul M.29. QR-CODE auslesen
<p>Verbundene Aktivitäten</p>	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A1 – Inspektion/Wartung Total Productive Maintenance • Aktivität A2 – Inspektion Wartung Instandhalter • Aktivität A3 – Instandsetzung • Aktivität A4 – Vereisung der Windenergieanlage (WEA)

	<ul style="list-style-type: none"> • Aktivität A5 – Ereignisgesteuerte Instandsetzung • Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung • Aktivität A7 – Zustandsabfrage Wartung (Condition Monitoring)
Aufteilung nach Entität	Dieses Modul befindet sich auf dem mobilen Endgerät, wie auch im Backend (Editierfunktion).

6.2.2M.2. Dokumentenmanagement

Komponente	M.2. Dokumentenmanagement								
Darstellung in Referenzarchitektur	<table border="1"> <tr> <td colspan="2">M.2. Dokumentenmanagement</td> </tr> <tr> <td>G.9.1. Dokumentenmanagement und -bearbeitung</td> <td>G.26.2. Möglichkeit zur Erweiterung der Handlungsanweisungen</td> </tr> <tr> <td>U1.8. Möglichkeit zur Ergänzung/Korrektur von Dokumentationen + Wikis</td> <td>U3.8. Dokumentationsmöglichkeit</td> </tr> <tr> <td>U4.7. Erweiterung der Leitfäden zur Fehlerbehebung</td> <td>U4.17. Unterstützung Datentypen</td> </tr> </table>	M.2. Dokumentenmanagement		G.9.1. Dokumentenmanagement und -bearbeitung	G.26.2. Möglichkeit zur Erweiterung der Handlungsanweisungen	U1.8. Möglichkeit zur Ergänzung/Korrektur von Dokumentationen + Wikis	U3.8. Dokumentationsmöglichkeit	U4.7. Erweiterung der Leitfäden zur Fehlerbehebung	U4.17. Unterstützung Datentypen
M.2. Dokumentenmanagement									
G.9.1. Dokumentenmanagement und -bearbeitung	G.26.2. Möglichkeit zur Erweiterung der Handlungsanweisungen								
U1.8. Möglichkeit zur Ergänzung/Korrektur von Dokumentationen + Wikis	U3.8. Dokumentationsmöglichkeit								
U4.7. Erweiterung der Leitfäden zur Fehlerbehebung	U4.17. Unterstützung Datentypen								
Zweck/Ziel	Das Modul dient dem Hinzufügen, Editieren, Löschen und Versionieren von Dokumenten, Artikeln und Medien, die für die Instandhaltung von Relevanz sind. Es ermöglicht anderen Modulen, Dokumente und Medien einfach einzubinden und auszulesen.								
Generisch/optional	Generisch								
Funktionsbeschreibung	Jegliche eigenen Inhalte können über das Modul bearbeitet, hinzugefügt, gelöscht und versioniert werden. Dazu muss das Modul zu einer Art internen Content-Management System bzw. Dokumentenmanagementsystem (DMS) werden. Vorliegende Content Management Systeme können auch direkt implementiert werden. Neben den dateibasierten Funktionen können auch Textdokumente oder Artikel hinterlegt und verwendet werden. Schnittstellen und Verbindungen zu anderen Modulen sind zu integrieren, indem diese einfach auf die Funktionalitäten, Dateien zugreifen können.								
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Auffinden und Bearbeiten von Dokumenten • Umwandeln von Informationen in verschiedene Formate • Versionierung von Dokumenten • Hinzufügen, Editieren, Löschen von Dateien und Dokumenten • Bereitstellen von Schnittstellen und Funktionen zum Aufrufen aus anderen Modulen 								
Verbindungen zu anderen Modulen	Baustein B.1. Lösungsunterstützung/Wissensmanagement: <ul style="list-style-type: none"> • Modul M.1. Handlungsleitfäden/Checklisten/Prüflisten • Modul M.3. Wiki 								

	Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen: <ul style="list-style-type: none"> • Modul M.4. Anlageninformationen • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.17. Fehlerdatenbank • Modul M.18. Fehlermeldung und -darstellung
Verbundene Aktivitäten	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A1 – Inspektion/Wartung Total Productive Maintenance • Aktivität A2 – Inspektion Wartung Instandhalter • Aktivität A3 – Instandsetzung • Aktivität A4 – Vereisung der Windenergieanlage (WEA)
Aufteilung nach Entität	Das Modul Dokumentenmanagement ist zentral im Backend vorhanden, die Dokumente sind jedoch von mobilen Endgerät erreichbar.

6.2.3M.3. Wiki

Komponente	M.3. Wiki
Darstellung in Referenzarchitektur	
Zweck/Ziel	<p>Dieses Modul implementiert ein Wissensmanagement in Form von Wikis zur Konservierung und Konsumierung von Wissen zur Unterstützung der Instandhaltung. Das übergeordnete Ziel ist, dass die Effizienz der Instandsetzung von Maschinen und Anlagen nicht mehr spürbar abhängig vom Wissenstand des einzelnen Instandhaltungsmitarbeiters ist. Ein Wiki ist hierbei ein Hypertextsystem für Webseiten, d. h. eine erweiterbare Menge von mit- und untereinander hierarchisch verlinkten Webseiten. Diese können durch die Nutzer in selbstorganisierter und kollaborativer Weise erstellt, mit zusätzlichen Medien wie Bildern und Videos erweitert und falls nötig editiert werden. Es entsteht eine sich stetig weiterentwickelnde Ansammlung von Expertenwissen zu den Themen, die für die Mitarbeiter und ihr Unternehmen relevant sind. Wichtige Details zu Prozessen und Vorgehensweisen, die normalerweise an einzelne Mitarbeiter gebunden sind, werden nun für alle verfügbar. Die Herausforderungen bei der Nutzung ergeben sich durch das offene System. Jeder Mitarbeiter und/oder eine Redaktion trägt die Verantwortung qualitativ hochwertige Inhalte zu erstellen, wie auch geheime Daten und Informationen nur für die berechtigten Personen sichtbar zu machen.</p>
Generisch/optional	Generisch, jedoch abhängig von den bisherig existierenden Lösungen im Unternehmen.
Funktionsbeschreibung	<p>Die Verwendung sollte sich möglichst nahe an der Benutzung bekannter Wikis wie http://www.wikipedia.org ausrichten. Nutzer können über eine Menüleiste die einzelnen Seiten des Wikis bearbeiten. Dazu zählt das Verändern und Erweitern eines bereits bestehenden Textes per WYSiWYG (What You See Is What You Get)-Editor, das Anlegen neuer und das Löschen veralteter Artikel, wie auch das Betrachten der bisher getätigten Veränderungen aller Nutzer. Über Personalisierungsfunktionen kann der Nutzer die Farbgebung, den Kontrast sowie</p>

	<p>die verfügbaren Schriften und deren Größe auswählen. Gewünschte Seiten können in andere Formate, z. B. PDF, umgewandelt und ausgedruckt werden.</p> <p>Medien und Dokumente sind in Verbindung mit dem Modul M.2. Dokumentenmanagement zu realisieren.</p> <p>Voraussetzungen für die Funktionsfähigkeit</p> <p>Zum Gelingen eines Wikis und um die Akzeptanz zu erhöhen, müssen technische wie menschliche Aspekte beachtet werden. Dazu zählt eine feingranulare Zugriffskontrolle, um spezielles Wissen nur den berechtigten Personen zugänglich zu machen. Daneben sollten Einträge überprüft und bewertet werden können. Zusätzlich sollte es die Möglichkeit geben, Gruppen, Bereiche und Rollen zu erstellen und zu verwalten. Eine Versionskontrolle mit Rollback Möglichkeit muss ebenso verfügbar sein. Für die Nutzer ist ein sogenannter „WYSIWYG“-Editor die beste Wahl. Bei Editor, der auch als Echtbilddarstellung bekannt ist, wird ein Dokument während der Bearbeitung am Bildschirm genauso angezeigt, wie es bei der Ausgabe mit einem Drucker aussehen würde bzw. bei der Ausgabe auf einer Webseite. Weiterhin ist das Einbinden von Anhängen, Bildern und Videos möglich. Die Ausgabe der Daten in anderen Formaten und das Ausdrucken von Dokumenten muss ebenfalls möglich sein.</p> <p>Um die nötige Akzeptanz zu erreichen, müssen dem Nutzer bestimmte Funktionen zur Verfügung stehen und spezielle Anreize gesetzt werden, die die Weiterentwicklung des Wikis fördern. Hier ist der problemlose und möglichst leichte Zugang zum Wiki, wie auch eine möglichst selbsterklärende Nutzung der grafischen Benutzeroberfläche, zu nennen. Diese sollte hinsichtlich Schriftgröße, Kontrast und eventuell bestehender Farbschwächen personalisiert werden können. Eine Bewertungsfunktion für Seiten und Inhalte ist nötig, um einen Qualitätsstandard zu etablieren und einen gewissen Ehrgeiz zu erzeugen das Wiki zu verbessern. Benachrichtigungen über Änderungen können per E-Mail, Pushnachricht auf das mobile Endgerät oder RSS-Feed dem Benutzer zugestellt werden. Schlussendlich erhöht eine gut funktionierende Suchfunktion die generelle Nutzbarkeit und Nützlichkeit enorm.</p> <p>Eine mögliche Technologie für das Modul wäre z. B. semantische Wikis. Diese ermöglichen Metadaten und ihre Relationen festzuhalten. Diese Wissensbasis kann maschinell gelesen und aufbereitet werden. Damit werden automatische Schlussfolgerungen über die Daten möglich, welche die Datenbasis automatisch erweitern und die Suche nach Informationen erleichtern.</p>
<p>Mögliche Unterfunktionen</p>	<ul style="list-style-type: none"> • Anzeige von vorhandenen Einträgen • Kommentieren von Einträgen • Logfunktion für das Editieren der Informationen • Hinzufügen von Dokumenten und Medien zu Wiki-Artikeln • Bewertungsfunktion von Artikeln
<p>Verbindungen zu anderen Modulen</p>	<p>Baustein B.1. Lösungsunterstützung/Wissensmanagement:</p> <ul style="list-style-type: none"> • Modul M.2. Dokumentenmanagement <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung
<p>Verbundene Aktivitäten</p>	<p>Keine Aktivitäten spezifiziert.</p>

Aufteilung nach Entität	Dieses Modul befindet sich im Backend und auf dem mobilen Endgerät.
--------------------------------	---

6.2.4M.4. Anlageninformationen

Komponente	M.4. Anlageninformationen														
Darstellung in Referenz-architektur	<table border="1"> <tr> <td colspan="2">M.4. Anlageninformationen</td> </tr> <tr> <td>G.10.1. Anzeige von Maschinenplänen (Steuerungspläne, SPS, etc.)</td> <td>G.16.1. Bauteilübersicht der Anlagen verfügbar</td> </tr> <tr> <td>G.17.1. Herstellerunterlagen der Anlage verfügbar</td> <td>U1.7. Anzeige Maschinenpläne</td> </tr> <tr> <td>U1.15. Bauteilinformationen für Anlagen verfügbar</td> <td>U1.16. Anlagenbeschreibung v. Hersteller</td> </tr> <tr> <td>U2.5. Zugriff Herstellerunterlagen</td> <td>U2.16. Anzeige Maschinenpläne</td> </tr> <tr> <td>U2.31. Bauteilinformationen</td> <td>U2.35. Push Benachrichtigung Updates bei Änderungen von HW/SW Komponenten d. Anlage</td> </tr> <tr> <td>U4.38. Bauteilinformationen verfügbar</td> <td>U4.23. Maschinenpläne im System verfügbar</td> </tr> </table>	M.4. Anlageninformationen		G.10.1. Anzeige von Maschinenplänen (Steuerungspläne, SPS, etc.)	G.16.1. Bauteilübersicht der Anlagen verfügbar	G.17.1. Herstellerunterlagen der Anlage verfügbar	U1.7. Anzeige Maschinenpläne	U1.15. Bauteilinformationen für Anlagen verfügbar	U1.16. Anlagenbeschreibung v. Hersteller	U2.5. Zugriff Herstellerunterlagen	U2.16. Anzeige Maschinenpläne	U2.31. Bauteilinformationen	U2.35. Push Benachrichtigung Updates bei Änderungen von HW/SW Komponenten d. Anlage	U4.38. Bauteilinformationen verfügbar	U4.23. Maschinenpläne im System verfügbar
M.4. Anlageninformationen															
G.10.1. Anzeige von Maschinenplänen (Steuerungspläne, SPS, etc.)	G.16.1. Bauteilübersicht der Anlagen verfügbar														
G.17.1. Herstellerunterlagen der Anlage verfügbar	U1.7. Anzeige Maschinenpläne														
U1.15. Bauteilinformationen für Anlagen verfügbar	U1.16. Anlagenbeschreibung v. Hersteller														
U2.5. Zugriff Herstellerunterlagen	U2.16. Anzeige Maschinenpläne														
U2.31. Bauteilinformationen	U2.35. Push Benachrichtigung Updates bei Änderungen von HW/SW Komponenten d. Anlage														
U4.38. Bauteilinformationen verfügbar	U4.23. Maschinenpläne im System verfügbar														
Zweck/Ziel	Das Modul bietet Zugriff auf Anlageninformationen, die im Zusammenhang mit der Instandhaltung stehen. Diese Informationen umfassen direkte Informationen zu einer Anlage, wie bspw. die Lokation, Ausrüstung, Bauteile, Funktion der Anlage und dient als zentraler Ort für anlagenbezogene Dokumente wie die Herstellerunterlagen, Maschinenpläne aber auch SPS-Pläne. Dabei ist der zentrale Zugangspunkt für die Form der Informationen entscheidend.														
Generisch/optional	Generisch														
Funktionsbeschreibung	Das Modul stellt Informationen zu einzelnen Anlagen zur Verfügung, die helfen, Probleme und Fehler bei Anlagen zu identifizieren. Im Unterschied zu Modul M.5. Anlagenübersicht und -auswertung sind jedoch keine Auswertungen oder Historien in diesem Modul enthalten, sondern weitestgehend statische Informationen, wie Lokation, Ausrüstung, Bauteile, Funktion der Anlage und Dokumente wie Herstellerunterlagen, Bauteilinformationen von einzelnen Teilen der Anlage und Maschinenpläne oder dazugehörige SPS-Programme. Diese werden zentral zur Verfügung gestellt. Der Gedanke bei der Bereitstellung der Informationen in dieser Form ist es, einen einzelnen Ort zu haben, an dem diese Informationen gebündelt verfügbar sind. Falls diese Informationen jedoch auch dezentral an der Anlage bereitgestellt werden soll, so bedarf es einer fortwährenden Synchronisation, um nicht verschiedene Versionen eines Dokuments im System zu haben.														

	<p>Die Informationen, die in diesem Modul enthalten sind, können dahingehend erweitert werden, sodass sie als Grundlage für den digitalen Zwilling der Anlage benutzt werden können.</p> <p>Das Updaten und Bearbeiten der Dokumente kann mithilfe des Moduls M.2. Dokumentenmanagement umgesetzt werden.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Bereitstellung von Herstellerunterlagen • Bereitstellung von Maschinenplänen und SPS-Programmen • Bereitstellung der Information über Bauteile einer Anlage • Zentraler Datenspeicherort z. B. für Messwerte
Verbindungen zu anderen Modulen	<p>Baustein B.2. Anlagenübersicht und -auswertung:</p> <ul style="list-style-type: none"> • Modul M.5. Anlagenübersicht und -auswertung • Modul M.6. Berechnung/Zugriff auf Produktionsplanung • Modul M.7. Interne Navigation • Modul M.8. Kataster für Hilfs- und Betriebsstoffe der Anlagen <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.2. Dokumentenmanagement • Modul M.19. Predictive Maintenance • Modul M.20. Synchronisation Hersteller und Betreiber • Modul M.29. QR-CODE auslesen
Verbundene Aktivitäten	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A1 – Inspektion/Wartung Total Productive Maintenance • Aktivität A2 – Inspektion/Wartung Instandhalter • Aktivität A3 – Instandsetzung • Aktivität A4 – Vereisung der Windenergieanlage (WEA) • Aktivität A5 – Ereignisgesteuerte Instandsetzung • Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung • Aktivität A7 – Zustandsabfrage/Wartung (Condition Monitoring)
Aufteilung nach Entität	<p>Mobiles Endgerät, Maschine und Backend.</p>

6.2.5M.5. Anlagenübersicht und -auswertung

Komponente	M.5. Anlagenübersicht und -auswertung																					
Darstellung in Referenz-architektur	<table border="1"> <thead> <tr> <th colspan="3" data-bbox="483 320 1362 342">M.5. Anlagenübersicht und -auswertung</th> </tr> </thead> <tbody> <tr> <td data-bbox="483 353 778 432">G.5.1. Überwachung/Protokollierung relevanter Maschinendaten und Zustände in Form eines Dashboards</td> <td data-bbox="786 353 1082 432">G.12.1. Anlagen-, Bauteil-, und Maßnahmenhistorie</td> <td data-bbox="1090 353 1362 432">G.25.2. Anlagenübersicht (Ausrüstung, Ersatzteile, Lokationen, Funktionalität)</td> </tr> <tr> <td data-bbox="483 443 778 521">G.31.2. Auswertung und Anzeige der aktuellen und mittleren Laufzeiten aller Anlagen (Stördauer/Anzahl Störungen etc.)</td> <td data-bbox="786 443 1082 521">U1.10. Anlagen-, Bauteil- und Maßnahmenhistorie</td> <td data-bbox="1090 443 1362 521">U1.18. Überwachung/Protokollierung relevanter Maschinendaten und Zustände</td> </tr> <tr> <td data-bbox="483 533 778 611">U1.34. Auswertung und Anzeige der aktuellen und mittleren Laufzeiten aller Anlagen (Stördauer/Anzahl Störungen etc.)</td> <td data-bbox="786 533 1082 611">U2.3. Anlagen-, Bauteil- und Maßnahmenhistorie</td> <td data-bbox="1090 533 1362 611">U2.23. Anzeige Stördauer (mittel und aktuell)</td> </tr> <tr> <td data-bbox="483 622 778 701">U2.24. Übersicht installierter Anlagen und deren Fehler/Status</td> <td data-bbox="786 622 1082 701">U2.26. Überwachung/Protokollierung relevanter Maschineendaten und Zustände in Form eines Dashboards</td> <td data-bbox="1090 622 1362 701">U3.10. GPS Genauigkeit für die Kartenversion (Zustand der Anlage)</td> </tr> <tr> <td data-bbox="483 712 778 790">U3.11. Betriebsprotokollierung</td> <td data-bbox="786 712 1082 790">U3.13. Anlagenplan</td> <td data-bbox="1090 712 1362 790">U4.1. Betriebsprotokollierung der Anlage (Störung, Zustandsdaten)</td> </tr> <tr> <td data-bbox="483 801 778 880">U4.12. Anlagenplan</td> <td data-bbox="786 801 1082 880">U4.19. Instandhaltungsmaßnahmehistorie</td> <td data-bbox="1090 801 1362 880"></td> </tr> </tbody> </table>	M.5. Anlagenübersicht und -auswertung			G.5.1. Überwachung/Protokollierung relevanter Maschinendaten und Zustände in Form eines Dashboards	G.12.1. Anlagen-, Bauteil-, und Maßnahmenhistorie	G.25.2. Anlagenübersicht (Ausrüstung, Ersatzteile, Lokationen, Funktionalität)	G.31.2. Auswertung und Anzeige der aktuellen und mittleren Laufzeiten aller Anlagen (Stördauer/Anzahl Störungen etc.)	U1.10. Anlagen-, Bauteil- und Maßnahmenhistorie	U1.18. Überwachung/Protokollierung relevanter Maschinendaten und Zustände	U1.34. Auswertung und Anzeige der aktuellen und mittleren Laufzeiten aller Anlagen (Stördauer/Anzahl Störungen etc.)	U2.3. Anlagen-, Bauteil- und Maßnahmenhistorie	U2.23. Anzeige Stördauer (mittel und aktuell)	U2.24. Übersicht installierter Anlagen und deren Fehler/Status	U2.26. Überwachung/Protokollierung relevanter Maschineendaten und Zustände in Form eines Dashboards	U3.10. GPS Genauigkeit für die Kartenversion (Zustand der Anlage)	U3.11. Betriebsprotokollierung	U3.13. Anlagenplan	U4.1. Betriebsprotokollierung der Anlage (Störung, Zustandsdaten)	U4.12. Anlagenplan	U4.19. Instandhaltungsmaßnahmehistorie	
M.5. Anlagenübersicht und -auswertung																						
G.5.1. Überwachung/Protokollierung relevanter Maschinendaten und Zustände in Form eines Dashboards	G.12.1. Anlagen-, Bauteil-, und Maßnahmenhistorie	G.25.2. Anlagenübersicht (Ausrüstung, Ersatzteile, Lokationen, Funktionalität)																				
G.31.2. Auswertung und Anzeige der aktuellen und mittleren Laufzeiten aller Anlagen (Stördauer/Anzahl Störungen etc.)	U1.10. Anlagen-, Bauteil- und Maßnahmenhistorie	U1.18. Überwachung/Protokollierung relevanter Maschinendaten und Zustände																				
U1.34. Auswertung und Anzeige der aktuellen und mittleren Laufzeiten aller Anlagen (Stördauer/Anzahl Störungen etc.)	U2.3. Anlagen-, Bauteil- und Maßnahmenhistorie	U2.23. Anzeige Stördauer (mittel und aktuell)																				
U2.24. Übersicht installierter Anlagen und deren Fehler/Status	U2.26. Überwachung/Protokollierung relevanter Maschineendaten und Zustände in Form eines Dashboards	U3.10. GPS Genauigkeit für die Kartenversion (Zustand der Anlage)																				
U3.11. Betriebsprotokollierung	U3.13. Anlagenplan	U4.1. Betriebsprotokollierung der Anlage (Störung, Zustandsdaten)																				
U4.12. Anlagenplan	U4.19. Instandhaltungsmaßnahmehistorie																					
Zweck/Ziel	<p>Das Modul dient der Anlagenauswertung und -übersicht. Es stellt eine Übersicht in Form eines typischen Dashboards über alle vorhanden Anlagen dar, verbindet diese mit nötigen Kennzahlen zur Kontrolle und Auswertung. Zudem werden historische Daten zu Aufträgen der jeweiligen Anlagen dargestellt. Weiterhin werden die Betriebsprotokolle der Anlagen in diesem Modul gespeichert und zur Verfügung gestellt.</p>																					
Generisch/optional	<p>Generisch</p>																					
Funktionsbeschreibung	<p>Die Funktionalitäten des Moduls gliedern sich in mehrere Teilbereiche, die nachfolgend aufgelistet werden:</p> <ul style="list-style-type: none"> • Bereitstellung einer durchsuchbaren Anlagenübersicht. Diese lässt sich einfach durchsuchen, filtern und kann bei Bedarf besondere Ereignisse hervorheben. So können Anlagen mit Unterschreitung bestimmter Kennzahlwerte gesondert dargestellt werden. Einzelne Anlagen können hierdurch ausgewählt werden, um genauere Informationen zu diesen Anlagen zu bekommen und um auf die Informationen aus dem Modul M.4. Anlageninformationen zuzugreifen. • Weiterhin dient dieses Modul zum Zugriff auf die Daten des Moduls M.4. Anlageninformationen. Lokationen, Ausrüstungen und Funktionen der anderen Anlagen können dadurch angezeigt und nach diesen Kriterien gefiltert werden. • Bereitstellung von Anlagen-, Bauteil- und Maßnahmenhistorien von Anlagen. • Beziehen der benötigten Daten für die Auswertung und Abspeicherung der Daten für die Betriebsprotokollierung. • Bereitstellung von vorkonfigurierbaren Dashboardansichten über bestimmte Anlagen und Anlagengruppen. Durch dieses Modul können Daten und Werte von Anlagen in Kombination mit dem Modul M.16. Fehlererkennung & Condition Monitoring angezeigt werden und so zielgerichtet Informationen für eine bestimmte Benutzergruppe leicht dargestellt werden. Verschiedene Ansichten müssen dazu bereitgestellt werden. 																					

	<p>Beispiele für die Auswertung von Kennzahlen sind bspw. die Berechnung der MTTR oder die MTBF, die Kosten bei Stillstand und Wartung sowie vergleichbare Kennzahlen.</p> <p>Um die benötigten Daten zur Anlagenauswertung darzustellen und zu archivieren, werden die nötigen Schnittstellen zu dem jeweiligen Datenhalter und den Maschinen, in Verbindung mit Modul M.16. Fehlererkennung & Condition Monitoring, benötigt. Gegebenenfalls ist die Integration von vorhandenen MES-Systemen zu prüfen, ob diese Teile der Funktionalitäten abdecken.</p> <p>Auswertungen zu den Anlagen werden anhand der vorliegenden Daten angefertigt. Dazu muss jede Anlage/Maschine mit den jeweiligen Daten eindeutig verknüpft werden. Beispielsweise muss es möglich sein die Historie von Aufträgen, welche durch das Modul M.9. Auftragsverwaltung/Priorisierung erstellt und bearbeitet werden, hier anzeigen zu lassen. Darüber hinaus müssen sich aus diesen Daten die Stillstandszeiten, Betriebszeiten, der Output und vergleichbare Auswertungen berechnen und intuitiv darstellen lassen. Eine grafische und tabellarische Auswertung sowie Sortierung nach mehreren Kriterien muss möglich sein. Hier ist zu beachten, dass die gewünschten Auswertungen spezifisch für das Unternehmen sind und deren nötige Indikatoren angepasst werden können.</p> <p>Voraussetzungen für die Funktionsfähigkeit</p> <p>Schnittstellen zu den Systemen, welche die genannten Daten halten, sind zu implementieren. Zugriff zu diesen muss möglich sein oder es müssen eigene Datenspeicher und -pflegemöglichkeiten implementiert werden. Die Auswertungen müssen an das Unternehmen angepasst werden können und es müssen genügend Daten verfügbar sein, um sinnvolle Analysen erstellen zu können.</p>
<p>Mögliche Unterfunktionen</p>	<ul style="list-style-type: none"> • Berechnen und grafisches Auswerten von Indikatoren anhand der jeweiligen Werte der Anlagen. • Darstellung eines übersichtlichen, editierbaren Anlagenplans. • Anzeige von Anlagenhistorie (Störungen, Maßnahmen, Ersatzteile, Laufzeiten). • Anpassen der Anzeige, z. B. Sortierung von Anlagenwerten, Auswertungen, neue Indikatoren. • Betriebsprotokollierung der Anlagen • Bereitstellung von konfigurierbaren Dashboards • Konfiguration und Komposition von Dashboards
<p>Verbindungen zu anderen Modulen</p>	<p>Baustein B.2. Anlagenübersicht und -auswertung:</p> <ul style="list-style-type: none"> • Modul M.4. Anlageninformationen • Modul M.6. Berechnung/Zugriff auf Produktionsplanung <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.11. Wartungsmanagement • Modul M.13. Schichtbuchfunktionalität/Synchronisation • Modul M.16. Fehlererkennung & Condition Monitoring • Modul M.17. Fehlerdatenbank • Modul M.19. Predictive Maintenance • Modul M.20. Synchronisation Hersteller und Betreiber

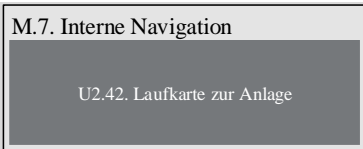
	<ul style="list-style-type: none"> • Modul M.26. Kommunikation • Modul M.29. QR-CODE auslesen
Verbundene Aktivitäten	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A1 – Inspektion/Wartung Total Productive Maintenance • Aktivität A2 – Inspektion Wartung Instandhalter • Aktivität A3 – Instandsetzung • Aktivität A4 – Vereisung der Windenergieanlage (WEA) • Aktivität A5 – Ereignisgesteuerte Instandsetzung • Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung • Aktivität A7 – Zustandsabfrage Wartung (Condition Monitoring) • Aktivität A8 – Verfügbarkeit (technisch/organisatorisch) • Aktivität A9 – Auswertung Fehlerhistorien einer Anlage
Aufteilung nach Entität	Dieses Modul muss vom mobilen Endgerät und dem Backend aus erreichbar sein, wobei aufwendigere Berechnungen auf dem Backend stattfinden und sich das mobile Endgerät vorwiegend auf die Darstellung konzentrieren sollte.

6.2.6M.6. Berechnung/Zugriff auf Produktionsplanung

Komponente	M.6. Berechnung/Zugriff auf Produktionsplanung
Darstellung in Referenzarchitektur	
Zweck/Ziel	Das Modul Berechnung/Zugriff auf Produktionsplanung stellt einen Zugriff auf die Produktionsplanung des Unternehmens her. Dies geschieht bspw. zur Identifikation von produktionsfreien Zeiten, damit Wartung oder Instandhaltung zeitlich besonders effizient eingeplant werden können. Eine besondere Kritikalität von Fehlern im Produktionsprozess kann so ebenso frühzeitig erkannt werden, wie der Zusammenhang eines Produktionsplans und das Entstehen von Fehlern.
Generisch/optional	Optional
Funktionsbeschreibung	<p>Daten aus bestehenden Systemen zur Produktionsplanung müssen integriert werden, sodass produktionsfreie Zeiten identifiziert werden können. Diese Informationen werden durch das Modul M.9. Auftragsverwaltung/Priorisierung benutzt, um Instandhaltungsmaßnahmen bevorzugt zu diesen Zeiten einzuplanen. Weiterhin können Maschinen identifiziert werden, die aufgrund des Produktionsplanes nicht ausfallen dürfen. Weitere Funktionen und die genaue Umsetzung der beschriebenen Funktionen sind individuell auf das jeweilige Unternehmen zugeschnitten.</p> <p>Voraussetzung für die Funktionstüchtigkeit des Moduls wäre es, Zugriff bzw. offene Schnittstellen für bestehende Produktionsplanungssysteme zu besitzen</p>
Mögliche Unterfunktionen	Keine Unterfunktionen spezifiziert

Verbindungen zu anderen Modulen	Baustein B.2. Anlagenübersicht und -auswertung: <ul style="list-style-type: none"> • Modul M.4. Anlageninformationen • Modul M.5. Anlagenübersicht und -auswertung Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen: <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.19. Predictive Maintenance
Verbundene Aktivitäten	<ul style="list-style-type: none"> • Grundaktivität
Aufteilung nach Entität	Backend und mobiles Endgerät.

6.2.7M.7. Interne Navigation

Komponente	M.7. Interne Navigation
Darstellung in Referenzarchitektur	 <p>The diagram shows a box labeled 'M.7. Interne Navigation' nested within a larger box labeled 'U2.42. Laufkarte zur Anlage'.</p>
Zweck/Ziel	Das Modul Interne Navigation stellt einen editierbaren Lageplan und eine interne Navigation innerhalb eines Unternehmens zur Verfügung. Es ist für große Unternehmensgelände und Hallen gedacht, bei denen eine Navigation nicht mit öffentlich verfügbarem Kartenmaterial und Methoden umgesetzt werden kann.
Generisch/optional	Optional
Funktionsbeschreibung	Das Modul stellt interne Lagepläne von Anlagen zur Verfügung. Das heißt, interne Fabrikgelände und die Lage der Maschinen sind dort abgebildet. Diese Informationen können editiert werden, sodass Orte von Maschinen, Wegbeschreibungen und ähnliche Informationen hinterlegt werden können, sodass eine Wegfindung und Navigation innerhalb der Hallen und dem Gelände angeboten wird. Diese kann statisch oder durch Mechanismen zur Indoornavigation dynamisch umgesetzt werden.
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Lageplanbereitstellung mit Editierfunktion • Möglichkeit Maschinen/Anlagen in die Pläne einzutragen • Wegfindung zu Anlagen/Maschinen
Verbindungen zu anderen Modulen	Baustein B.2. Anlagenübersicht und -auswertung: <ul style="list-style-type: none"> • M.4. Anlageninformationen
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.

Aufteilung nach Entität	mobiles Endgerät.
--------------------------------	-------------------

6.2.8M.8. Kataster für Hilfs- und Betriebsstoffe der Anlagen

Komponente	M.8. Kataster für Hilfs- und Betriebsstoffe der Anlagen
Darstellung in Referenzarchitektur	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> <p>M.8. Kataster für Hilfs- und Betriebsstoffe der Anlagen</p> <p>U1.27. Link zum Kataster von Hilfs- und Betriebsstoffen für die jeweilige Anlage</p> </div>
Zweck/Ziel	Das Modul stellt ein Kataster für Hilfs- und Betriebsstoffe von Produktionsanlagen zur Verfügung. Eine fehlerhafte Auswahl von Hilfs- und Betriebsstoffen kann somit durch den Instandhalter überprüft und eingeschätzt werden.
Generisch/optional	Optional
Funktionsbeschreibung	<p>Das Modul gibt eine Übersicht über die Hilfs- und Betriebsstoffe einer Produktionsanlage. Diese werden bei Auswahl einer Anlage durch das Modul M.4. Anlageninformationenseparat dargestellt, sodass klar ersichtlich ist, welche Stoffe zu einer Anlage gehören und mit welchen diese benutzt werden soll.</p> <p>Entweder wird ein bestehendes Kataster integriert oder kann selbst für jede Anlage/Maschine angelegt werden. Bedingung für die Funktionstüchtigkeit des Moduls ist, dass die Liste von Hilfs- und Betriebsstoffen bekannt ist.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Extraktion von Hilfs- und Betriebsstoffen aus vorhandenem Kataster. • Darstellung von Hilfs- und Betriebsstoffen einer Anlage. • Editierung von Hilfs- und Betriebsstoffen.
Verbindungen zu anderen Modulen	<p>Baustein B.2. Anlagenübersicht und -auswertung</p> <ul style="list-style-type: none"> • Modul M.4. Anlageninformationen
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Backend und mobiles Endgerät.

6.2.9M.9. Auftragsverwaltung/Priorisierung

Komponente	M.9. Auftragsverwaltung/Priorisierung																							
Darstellung in Referenz-architektur	<table border="1"> <tr> <td colspan="3" data-bbox="475 306 1372 331">M.9. Auftragsverwaltung/Priorisierung</td> </tr> <tr> <td data-bbox="488 338 778 416">G.6.1. Auftragsverwaltung mit Anzeige Status d. Instandhaltungsaufträge durch das System</td> <td data-bbox="786 338 1077 416">G.22.2. Priorisierung der abzuarbeitenden Tätigkeiten (automatische Priorisierung der Aufträge, Tätigkeiten, Fehler, Auftragsverteilung)</td> <td data-bbox="1085 338 1369 416">G.39.3. Eskalationsmöglichkeit mit Informationsweitergabe</td> </tr> <tr> <td data-bbox="488 427 778 506">U1.13. Anzeige offener und abgeschlossener Aufträge durch das System</td> <td data-bbox="786 427 1077 506">U1.23. Priorisierung Instandhaltungsfälle</td> <td data-bbox="1085 427 1369 506">U1.37. Eskalationsmöglichkeit mit Informationsweitergabe</td> </tr> <tr> <td data-bbox="488 517 778 595">U2.8. Automatischer Link zu Leitpfaden bei Störung</td> <td data-bbox="786 517 1077 595">U2.13. Anzeige offener und abgeschlossener Aufträge durch das System</td> <td data-bbox="1085 517 1369 595">U2.17. Störungbenachrichtigungs-/Anlagenzuweisung zu Personengruppe</td> </tr> <tr> <td data-bbox="488 607 778 685">U2.18. Status Störung</td> <td data-bbox="786 607 1077 685">U2.36. Automatische Priorisierung Instandhaltungsfälle</td> <td data-bbox="1085 607 1369 685">U2.37. "Hilfe-Button"</td> </tr> <tr> <td data-bbox="488 696 778 775">U2.46. Eskalationsmöglichkeit mit Informationsweitergabe</td> <td data-bbox="786 696 1077 775">U3.16. Ticket System</td> <td data-bbox="1085 696 1369 775">U4.42. Eskalationsmöglichkeit mit Informationsweitergabe</td> </tr> <tr> <td colspan="2" data-bbox="488 786 1077 864">U4.10. Nachverfolgung offener Punkte</td> <td data-bbox="1085 786 1369 864"></td> </tr> </table>			M.9. Auftragsverwaltung/Priorisierung			G.6.1. Auftragsverwaltung mit Anzeige Status d. Instandhaltungsaufträge durch das System	G.22.2. Priorisierung der abzuarbeitenden Tätigkeiten (automatische Priorisierung der Aufträge, Tätigkeiten, Fehler, Auftragsverteilung)	G.39.3. Eskalationsmöglichkeit mit Informationsweitergabe	U1.13. Anzeige offener und abgeschlossener Aufträge durch das System	U1.23. Priorisierung Instandhaltungsfälle	U1.37. Eskalationsmöglichkeit mit Informationsweitergabe	U2.8. Automatischer Link zu Leitpfaden bei Störung	U2.13. Anzeige offener und abgeschlossener Aufträge durch das System	U2.17. Störungbenachrichtigungs-/Anlagenzuweisung zu Personengruppe	U2.18. Status Störung	U2.36. Automatische Priorisierung Instandhaltungsfälle	U2.37. "Hilfe-Button"	U2.46. Eskalationsmöglichkeit mit Informationsweitergabe	U3.16. Ticket System	U4.42. Eskalationsmöglichkeit mit Informationsweitergabe	U4.10. Nachverfolgung offener Punkte		
M.9. Auftragsverwaltung/Priorisierung																								
G.6.1. Auftragsverwaltung mit Anzeige Status d. Instandhaltungsaufträge durch das System	G.22.2. Priorisierung der abzuarbeitenden Tätigkeiten (automatische Priorisierung der Aufträge, Tätigkeiten, Fehler, Auftragsverteilung)	G.39.3. Eskalationsmöglichkeit mit Informationsweitergabe																						
U1.13. Anzeige offener und abgeschlossener Aufträge durch das System	U1.23. Priorisierung Instandhaltungsfälle	U1.37. Eskalationsmöglichkeit mit Informationsweitergabe																						
U2.8. Automatischer Link zu Leitpfaden bei Störung	U2.13. Anzeige offener und abgeschlossener Aufträge durch das System	U2.17. Störungbenachrichtigungs-/Anlagenzuweisung zu Personengruppe																						
U2.18. Status Störung	U2.36. Automatische Priorisierung Instandhaltungsfälle	U2.37. "Hilfe-Button"																						
U2.46. Eskalationsmöglichkeit mit Informationsweitergabe	U3.16. Ticket System	U4.42. Eskalationsmöglichkeit mit Informationsweitergabe																						
U4.10. Nachverfolgung offener Punkte																								
Zweck/Ziel	<p>Das Modul M.9. Auftragsverwaltung/Priorisierung organisiert und verwaltet die Instandhaltungsmaßnahmen und dient als Eingangspunkt für Fehlermeldungen bzw. Maßnahmenmeldungen. Das Modul erstellt Abfolge, Priorisierung und Zeitpunkte von Instandhaltungs- bzw. Wartungsaufgaben. Es ist für die Funktionstüchtigkeit des IIS essentiell, da es den Arbeitsablauf und die -festlegung steuert und ermöglicht, den Status aller Instandhaltungsaufträge zu verfolgen.</p> <p>Darüber hinaus benötigt das Modul eine hohe Individualisierbarkeit und aggregiert eine große Menge an Daten aus anderen Modulen bzw. Systemen, um die Aufträge zu priorisieren bzw. festzulegen. Insbesondere durch die Möglichkeit eine objektive Priorisierung anzugeben, die sich klar nach Managementregeln richtet, ergeben sich enorme Verbesserungspotentiale für die Aufgabenverteilung innerhalb der Instandhaltung.</p>																							
Generisch/optional	Generisch – bestimmte Teilfunktionalitäten dieses Moduls sind jedoch optional.																							
Funktionsbeschreibung	<p>Das Modul untergliedert sich in verschiedene funktionale Teilbereiche. Diese werden in den Aktivitäten A1-A7 und der Grundaktivität (vgl. Kapitel 6.4.1) prototypisch in einer festen Ordnung durchlaufen. Jedoch gibt es in der Praxis viele Anwendungsfälle bei denen unterschiedliche Einstiegspunkte nötig sind. Denn das Modul muss von mehreren Stellen aus aufgerufen werden können und nicht alle Teilbereiche werden zwangsläufig durchlaufen. Prinzipiell kann als Einstiegspunkt zwischen dem Standardvorgang, wie in der Grundaktivität beschrieben, der manuellen oder automatischen Auftragsverwaltung und Priorisierung unterschieden werden. Diese Teilbereiche werden nachfolgend beschrieben.</p> <p>Eingang neuer Aufträge</p> <p>Das Modul ist standardmäßig Eingangspunkt neuer Aufträge/Meldungen. Ein sogenannter Maßnahmenauftrag wird erstellt, sobald Fehlermeldungen oder Wartungsmeldungen aus den Modulen M.11. Wartungsmanagement oder M.16. Fehlererkennung & Condition Monitoring empfangen werden. Dieser Maßnahmenauftrag ist ein zu dem Zeitpunkt noch nicht mit Informationen und Arbeitsaufträgen angereicherter Auftrag. Dieser Auftrag wird erstellt, sobald Informationen empfangen werden, die eine Instandhaltungsmaßnahme nach sich ziehen müssen. Hierunter fallen die besagten Fehler- und Wartungsmeldungen oder Meldungen aus dem Condition Monitoring aufgrund unvorhergesehener Abnutzung.</p>																							

Nach dem Erstellen dieses Maßnahmenauftrages, wird dieser durch spätere Funktionen mit Informationen zur Durchführung angereichert und wandelt sich von einem Maßnahmenauftrag zu einem konkreten Arbeitsauftrag.

Eine weitere Möglichkeit neue Aufträge zu generieren, kann durch manuellen Auftragserstellung geschehen. Hier meldet ein Bediener oder Instandhalter einen Auftrag durch eine Eingabemaske. Dadurch wird automatisch ein dementsprechender Maßnahmenauftrag samt Fehler- bzw. Wartungsmeldung erstellt.

Auftragsverwaltung und Historie

Das Modul kann zur Verfolgung und Verwaltung der Aufträge genutzt werden. Der aktuelle Status der Aufträge kann nachvollzogen, die aktuelle Bearbeitung und abgelaufene Aufträge können betrachtet werden. Durch diese Funktionalität kann das Management von Instandhaltungsaufträgen deutlich verbessert werden, da das Management offene Aufträge immer im Blick hat.

Datenaggregation

Zur Festlegung, Priorisierung bzw. dem Managen der Instandhaltungsaufträge werden weitere Informationen über den eingegangenen Auftrag oder den aktuellen Status aller Aufträge benötigt, die über die ursprüngliche Meldung der Maschine bzw. des Wartungsmanagements hinausgehen. Diese Daten werden vom Auftragsverwaltungsmodul aus anderen Modulen extrahiert, sodass z. B. Daten zu einem Fehler oder der beauftragte Mitarbeiter bzw. dessen Identifikator im System mit dem Arbeitsauftrag in Verbindung gesetzt werden können. Diese Daten werden für den Arbeitsauftrag bzw. die Aufträge oder auch für die komplette Neuplanung zusammengestellt. Dazu zählen können u. a.:

- Anlagenpläne
- Produktionspläne
- Maschinenbelegungen
- Pufferbestände vor und nach der Maschine
- Mitarbeiterpläne, -verfügbarkeit und deren Schichten
- Nötige Ersatzteile und deren Verfügbarkeit
- Kostenberechnung von Stillständen
- Nötige Werkzeuge
- (Wetterdaten)
- Kundenaufträge – Enddaten bestimmter Aufträge
- Aufbereitete Fehlerinformationen
- Handlungsleitfäden/Checklisten
- Anlageninformationen und -auswertungen

Diese Informationen werden zwischengespeichert oder fest im Modul hinterlegt. Unterschiede bei dieser Entscheidung könnten sich stark auf die Performance des Systems auswirken, da ein häufiges Laden aus anderen Modulen z. T. zu starken Latenzen führen kann.

Darüber hinaus ist die Methodik des Synchronisierens mit dem mobilen Endgerät für das Modul von hoher Bedeutung, da der Instandhalter den Auftrag möglichst mit allen Details zur Verfügung gestellt bekommen sollte. Der Instandhalter sollte vor Ort dann entsprechende Daten neu laden oder auf eine vorher geladene Offlinekopie zurückgreifen können. Je nach Anwendungsfall ist daher die Entscheidung der Implementierung bzgl. der Zwischenspeicherung bzw. des Neuladens zu treffen.

Mit all diesen Informationen ist es dem Modul möglich, eine Auftragsplanung zu vollziehen oder diese zu ändern. Des Weiteren sind Informationen für den

Instandhalter und seine Aufträge, wie Handlungsleitfäden oder Fehlerinformationen, vorgeladen und können im Auftrag hinterlegt werden. Die Informationen, die für einen Auftrag zu laden sind, haben hier jedoch keinen Anspruch auf Vollständigkeit. Sie können von Unternehmen zu Unternehmen, genauso wie die Anforderungen an eine Priorisierung, deutlich abweichen. Daher muss je nach Anforderungen entschieden werden, welche Daten für Priorisierung und Auftragsmanagement nötig sind.

Automatische Festlegung des Zeitpunktes/Priorisierung

Die Festlegung des Zeitpunktes der Instandhaltungsmaßnahmen kann erfolgen, nachdem alle nötigen Daten dafür zur Verfügung stehen. Hierbei sind mehrere Funktionen nötig diese zu priorisieren.

Algorithmus zur Priorisierung

Geeignete Algorithmen/Heuristiken für eine schnelle Zuweisung von Arbeitsaufträgen und generelle Neuplanung (Komplettplanung) müssen identifiziert und implementiert werden. Diese müssen in Abhängigkeit an die Anforderungen zur Geschwindigkeit und der Komplexität der Berechnungen getroffen werden. Es bedarf hierbei zwei Arten der maschinellen Priorisierung, einer ad-hoc Einplanung von kritischen Fehlern und einer generellen rechenintensiven Großplanung. Dies hängt damit zusammen, dass die optimale Auftragsplanung bei einer gewissen, meist vorherrschenden Komplexität, bis dato extrem laufzeitintensiv ist. Es muss jedoch möglich sein, besonders kritische Fehler kurzfristig in einen bestehenden Auftragsplan einzuordnen.

Managementansatz Priorisierung

Es müssen Regeln zur Priorisierung festgelegt werden können, so dass bspw. bestimmte Aufträge vorrangig priorisiert werden, wenn es Enddaten für Kundenaufträge bzw. Konventionalstrafen nötig machen. Darüber hinaus muss festgelegt werden, welche von den oben genannten Daten Vorrang gegenüber anderen haben. Daher müssen eindeutige Regeln für die Terminvergabe erstellt werden.

Auswahlmöglichkeiten Priorisierung

Zwischen verschiedenen Priorisierungsmöglichkeiten sollte ein schneller Wechsel möglich sein. Verschiedene Profile mit vorkonfigurierten Einstellungen wären möglich.

Zeitraum für Komplettplanung

Zeiträume für die komplette Neuplanung oder Komplettaktualisierung müssen festgelegt werden. Insbesondere wenn deren zeitlicher Aufwand aufgrund langer Laufzeiten und anderen Randbedingungen, wie eine nötige Synchronisation mit anderen Geräten, zu Komplikationen führen kann.

Ein weiterer Faktor für die Priorisierung bzw. Auftragsplanung könnte sein, dass eine Mitarbeiterzuweisung berechnet werden muss, die die Verfügbarkeit von Mitarbeitern sowie die nötigen Fähigkeiten oder Schulungen zur Instandhaltung der Mitarbeiter betrachtet.

Manuelle Priorisierung

Neben der automatischen Priorisierung muss es dem Instandhaltungsmanagement auch möglich sein, Aufträge manuell zu priorisieren. Dazu ist eine grafische Oberfläche nötig, in der den einzelnen Aufträgen eine Priorisierung direkt zugeteilt werden kann. Zusätzlich sollte eine direkte Verteilung an bestimmte Instandhalter zu bestimmten Zeiten möglich sein.

Bestätigung durch Instandhalter

Neben der Planung und Priorisierung versendet das Modul einen Arbeitsauftrag an das Modul M.10. Mitarbeitermanagement. Diesen müssen die jeweiligen

Instandhalter bestätigen. Daraufhin wird die Antwort an das Modul geschickt und eine dazu passende Entscheidung getroffen:

- Entweder wird eine Neuplanung mit Neuauswahl des Mitarbeiters angestoßen oder es folgt die Buchung des Mitarbeiters in Verbindung mit Modul M.10. Mitarbeitermanagement.
- M.10. Mitarbeitermanagement zur Benachrichtigung der Verantwortlichen.

Bearbeitung des Auftrages durch Instandhalter

Bevor der Instandhalter seine Tätigkeit an der Maschine beginnt, muss er in dem Modul den Beginn der Instandhaltungstätigkeit festhalten. Des Weiteren wird jeder einzelne Schritt der Instandhaltung, der händisch protokolliert werden soll, in diesem Modul in Verbindung mit dem Modul M.1.

Handlungsleitfäden/Checklisten/Prüflisten aufgezeichnet. Das Ende der Tätigkeit wird ebenfalls festgehalten und das Modul setzt ggf. bestimmte Wartungszähler herunter und verweist auf evtl. zu ändernde Dokumente. Hier wäre es möglich, die Informationen aus dem Auftrag mit den optionalen Modulen M.13.

Schichtbuchfunktionalität/Synchronisation und M.15. Arbeitszeiterfassung zu synchronisieren.

Eskalation des Auftrages

Das Modul beinhaltet eine Eskalationsmöglichkeit, die entweder automatisch oder manuell ausgeführt werden kann. Die automatische Eskalation kann bspw. nach einer gewissen Zeit eines Fehlers oder der Bearbeitung eines Auftrags erfolgen oder in Abhängigkeit von den Änderungen bestimmter Zustandswerte einer Maschine. Die Manuelle Eskalation kann durch den Instandhalter oder den Bearbeiter des Instandhaltungsplans durchgeführt werden. Dies kann bspw. bei Überlastung oder bei Aufträgen der Fall sein, bei denen der zuständige Instandhalter eine externe Firma oder den Maschinenhersteller beauftragt den aktuellen Auftrag zu übernehmen. Denkbar ist aber auch eine interne Eskalationsmöglichkeit, wenn mehrere Eskalationsstufen der Instandhaltung intern existieren. Besonders sind hier die Möglichkeiten zur Informationsweitergabe in Form von Auftragsdaten oder Maschinendaten zu beachten. Insbesondere die Weitergabe an Fremdsysteme von externen Partnern bedarf ein sicheres Vorgehen.

Synchronisation

Die Synchronisation zwischen verschiedenen Systemen stellt eine große Herausforderung in diesem Modul dar. Änderungen an den Plänen müssen je nach individueller Ausprägung von Maschinen und mobilen Endgeräten mit den Backend abgeglichen werden. Dazu sind Mechanismen zu implementieren, die eine sofortige Synchronisation zwischen verschiedenen Systemen und Endgeräten ermöglichen.

Voraussetzungen für die Funktionsfähigkeit

Die Daten der jeweiligen anderen Module müssen vorhanden und verfügbar sein. Zusätzlich müssen Verbindungen zu den beteiligten Systemen bestehen und eine ausreichend hohe Rechenleistung zur Berechnung der Aufträge muss vorhanden sein

Mögliche Unterfunktionen

- Aufnahme und Speicherung neuer eingehender Aufträge bzw. Meldungen
- Anlegen von Instandhaltungsaufträgen
- Manuelle Bearbeitung von Aufträgen
- Manuelle Priorisierung von Aufträgen
- Löschen von Aufträgen
- (Automatische) Priorisierung und zeitliche Festlegung von Instandhaltungsaufträgen

	<ul style="list-style-type: none"> • Import und Erstellung neuer Prioritätsregeln und Profilen mit vordefinierten Einstellungen • Austausch der benutzen Algorithmen • Schnellpriorisierung (für dringende Aufträge, statt kompletter neuer, laufzeitintensiver Priorisierung) • Automatische vollständige Priorisierung in Abhängigkeit von Neuplanungsbedarf • Extraktion von benötigten Daten zur Priorisierung aus anderen Modulen • Erstellen und Weiterleiten von Arbeitsaufträgen • Protokollierung der Instandhaltungstätigkeit • Darstellung des Status von aktuellen Aufträgen (bspw. offen, in Bearbeitung, gelöst) • Eskalation eines Auftrages (manuell oder automatisch) • Informationsweitergabe von Auftragsdaten an externe Unternehmen
Verbindungen zu anderen Modulen	<p>Baustein B.3. Tätigkeitsmanagement der Instandhaltung:</p> <ul style="list-style-type: none"> • Modul M.10. Mitarbeitermanagement • Modul M.11. Wartungsmanagement • Modul M.12. Auftragsmanagement in Verbindung mit ERP • Modul M.13. Schichtbuchfunktionalität/Synchronisation • Modul M.14. Work and People Tracking • Modul M.15. Arbeitszeiterfassung <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.1. Handlungsleitfäden/Checklisten/Prüflisten • Modul M.2. Dokumentenmanagement • Modul M.3. Wiki • Modul M.5. Anlagenübersicht und -auswertung • Modul M.6. Berechnung/Zugriff auf Produktionsplanung • Modul M.16. Fehlererkennung & Condition Monitoring • Modul M.17. Fehlerdatenbank • Modul M.18. Fehlermeldung und -darstellung • Modul M.19. Predictive Maintenance • Modul M.21. Navigation • Modul M.22. Reisekostenabrechnung • Modul M.23. Serviceprotokolle • Modul M.24. Vertragsmanagement • Modul M.25. Signierfunktion • Modul M.26. Kommunikation • Modul M.29. QR-CODE auslesen • Modul M.28. Ersatzteilmanagement

	<ul style="list-style-type: none"> • Modul M.37. Wetterdaten
Verbundene Aktivitäten	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A1 – Inspektion/Wartung Total Productive Maintenance • Aktivität A2 – Inspektion Wartung Instandhalter • Aktivität A3 – Instandsetzung • Aktivität A4 – Vereisung der Windenergieanlage (WEA) • Aktivität A5 – Ereignisgesteuerte Instandsetzung • Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung • Aktivität A7 – Zustandsabfrage Wartung (Condition Monitoring) • Aktivität A8 – Verfügbarkeit (technisch/organisatorisch) • Aktivität A10 – Synchronisation
Aufteilung nach Entität	Backend.

6.2.10 M.10. Mitarbeitermanagement

Komponente	M.10. Mitarbeitermanagement												
Darstellung in Referenzarchitektur	<table border="1"> <tr> <td colspan="3">M.10. Mitarbeitermanagement</td> </tr> <tr> <td>G.18.1. Annahme und Ablehnung von Aufträgen durch Instandhalter möglich</td> <td>G.36.2. Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit</td> <td>U1.20. Anonymisierungsfunktion der Mitarbeiter für das Management</td> </tr> <tr> <td>U1.21. Annahme und Ablehnung von Aufträgen durch Instandhalter möglich</td> <td>U1.36. Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit</td> <td>U2.19. Annahme und Ablehnung von Aufträgen durch Instandhalter möglich</td> </tr> <tr> <td>U2.38. Status Mitarbeiter</td> <td>U4.20. Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit</td> <td>U4.28. Annahme und Ablehnung von Aufträgen durch Instandhalter möglich</td> </tr> </table>	M.10. Mitarbeitermanagement			G.18.1. Annahme und Ablehnung von Aufträgen durch Instandhalter möglich	G.36.2. Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit	U1.20. Anonymisierungsfunktion der Mitarbeiter für das Management	U1.21. Annahme und Ablehnung von Aufträgen durch Instandhalter möglich	U1.36. Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit	U2.19. Annahme und Ablehnung von Aufträgen durch Instandhalter möglich	U2.38. Status Mitarbeiter	U4.20. Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit	U4.28. Annahme und Ablehnung von Aufträgen durch Instandhalter möglich
M.10. Mitarbeitermanagement													
G.18.1. Annahme und Ablehnung von Aufträgen durch Instandhalter möglich	G.36.2. Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit	U1.20. Anonymisierungsfunktion der Mitarbeiter für das Management											
U1.21. Annahme und Ablehnung von Aufträgen durch Instandhalter möglich	U1.36. Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit	U2.19. Annahme und Ablehnung von Aufträgen durch Instandhalter möglich											
U2.38. Status Mitarbeiter	U4.20. Übersicht über Instandhalter samt Kontaktdaten, Zuweisung und Verfügbarkeit	U4.28. Annahme und Ablehnung von Aufträgen durch Instandhalter möglich											
Zweck/Ziel	Das Modul dient der Organisation der an der Instandhaltung beteiligten Mitarbeitern. Zum einen durch eine Selbstverwaltung der eigenen Aufträge durch die Instandhalter selbst, zum anderen durch eine Übersicht für das Management, bei der die aktuelle Zuweisung und Verfügbarkeit der Instandhalter dargestellt wird.												
Generisch/optional	Generisch, mit einer optionalen Möglichkeit zur Anonymisierung der Mitarbeiter.												
Funktionsbeschreibung	<p>Zur Organisation der Instandhalter und zur Darstellung der aktuellen Tätigkeiten bedarf es mehrere Teilfunktionen. Diese müssen je nach Bedarf komplett selbst implementiert werden oder können durch bereits bestehende Systeme abgedeckt werden.</p> <p>Beispielsweise wird eine Kalenderfunktion mit Terminen/Aufgaben der einzelnen Mitarbeiter in diesem Modul bereitgestellt, so dass klar ersichtlich wird, für welche Aufgaben die Instandhalter eingeplant sind und wann sie noch bestimmte Arbeiten ausführen können. Diese Kalenderfunktion kann selbst implementiert oder im Verbund mit bestehenden unternehmenseigenen Kalendersystemen umgesetzt werden.</p> <p>Eine Synchronisation zwischen Mitarbeitern und deren Terminen aus diesem Modul muss mit den zugehörigen Aufträgen, aus Modul M.9.</p>												

Auftragsverwaltung/Priorisierung, stattfinden. Ein Instandhaltungsbezogener Termin eines Mitarbeiters stellt einen Auftrag dar, der den Mitarbeiter für einen gewissen Zeitraum „belegt“. Der Auftrag aus dem besagten Modul muss also direkt dem Termin zugeordnet werden können. Die Kalenderfunktion muss dazu durch andere Module angesprochen und neue Termine müssen hinzugefügt werden können.

Diese Funktion ist die Grundlage für eine Managementsicht, bei der die aktuelle Zuweisung aller Instandhalter übersichtlich dargestellt wird. Eine Verteilung der Instandhalter auf einzelne Aufgaben kann hier per Drag-and-Drop durch die zuständigen Personen getätigt werden, sofern dieses nicht automatisch durch das System erledigt werden soll. Weiterhin ist insbesondere bei Unternehmen mit hohen Datenschutzanforderungen eine Möglichkeit der Anonymisierung der Mitarbeiter wichtig, sodass Aufgaben bspw. rollenbezogen oder schichtbezogen vergeben werden können. Letztere Funktion ist jedoch davon abhängig, ob die Anforderung an eine anonyme Darstellung der eigenen Mitarbeiter von Relevanz ist.

Zusätzlich ist es optional möglich, durch eine Verbindung mit den Modulen Modul M.14. Work and People Tracking und Modul M.15. Arbeitszeiterfassung, den Ort/Status und Arbeitszeit mit dem Profil des Mitarbeiters zu verknüpfen. Dies ist jedoch stark unternehmensindividuell und bedarf einer Abstimmung mit Mitarbeitervertretern.

Neben der managementorientierten Sicht dient dieses Modul auch dazu, dem einzelnen Instandhalter Funktionalitäten anzubieten, die seine Arbeit erleichtern. Neben der Bereitstellung des eigenen Kalenders, einer Anzeige der aktuellen Aufträge und Termine, kann der Instandhalter eingehende Aufträge annehmen oder ablehnen.

Neben der den genannten Funktionen muss es dem Management bzw. den Administratoren möglich sein, neue Mitarbeiter anzulegen, optional deren Kompetenzen und Schulungen festzuhalten und diese jeweils in die Hierarchie des Unternehmens einzutragen. So müssen diese in Verbindung mit dem rein technischen Modul M.31. Technisches Nutzermanagement in ein unternehmensspezifisches Rollenmodell eingetragen werden können, um bspw. Eskalationsstufen der Instandhaltung abzubilden. Weiterhin sollte es möglich sein, die Instandhalter bestimmten Anlagengruppen zuzuweisen, für diese die dann vordergründig verantwortlich sind.

Voraussetzungen für die Funktionsfähigkeit

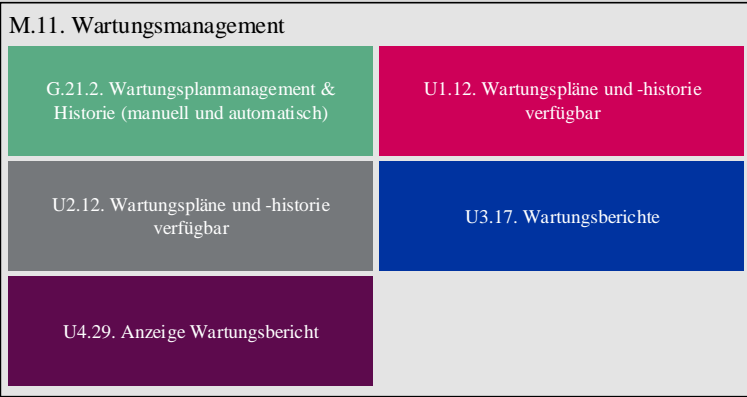
Die Mitarbeiter müssen in Verbindung mit dem Modul M.31. Technisches Nutzermanagement eingetragen und gepflegt werden.

Mögliche Unterfunktionen

- Bereitstellung einer Kalenderfunktionalität
- Terminplanung der Mitarbeiter
- Termine für den Tag anzeigen
- Anlegen von Mitarbeitern in Verbindung mit technischer Anlage von Benutzern
- Einordnen der Mitarbeiter in Rollenmodell des Unternehmens
- Annehmen und Ablehnen von Aufträgen durch Instandhalter
- Import und Export von Kalendern in Verbindung mit evtl. bestehenden Lösungen
- Anzeigen der verfügbaren Mitarbeiter
- Anzeige der aktuellen Zuweisung der Mitarbeiter
- Verbindung der Termine mit Aufträgen herstellen

	<ul style="list-style-type: none"> • Anonymisierungsfunktion für Mitarbeiter
Verbindungen zu anderen Modulen	<p>Baustein B.3. Tätigkeitsmanagement der Instandhaltung:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.13. Schichtbuchfunktionalität/Synchronisation • Modul M.14. Work and People Tracking • Modul M.15. Arbeitszeiterfassung <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.1. Handlungsleitfäden/Checklisten/Prüflisten • Modul M.26. Kommunikation • Modul M.30. Remote Zugriff/Fernwartung auf Anlagen/Maschinen
Verbundene Aktivitäten	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A1 – Inspektion/Wartung Total Productive Maintenance • Aktivität A2 – Inspektion Wartung Instandhalter • Aktivität A3 – Instandsetzung • Aktivität A4 – Vereisung der Windenergieanlage (WEA) • Aktivität A5 – Ereignisgesteuerte Instandsetzung • Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung • Aktivität A7 – Zustandsabfrage Wartung (Condition Monitoring)
Aufteilung nach Entität	<p>Das Modul befindet sich im Backend als auch auf dem mobilen Endgerät. Komplexe Berechnungen sind, abhängig von den technischen Gegebenheiten, auf dem Backend zu platzieren.</p>

6.2.11 M.11. Wartungsmanagement

Komponente	M.11. Wartungsmanagement
Darstellung in Referenzarchitektur	
Zweck/Ziel	<p>Das Modul übernimmt das Wartungsmanagement von allen beteiligten Anlagen. Das Wartungsmanagement beinhaltet das Anlegen und Pflegen von Wartungsplänen von Bauteilen, Maschinen und Anlagen. Daneben werden durch das Bereitstellen von Wartungsberichten und der Zurverfügungstellung von Wartungshistorien von Anlagen dem Instandhalter wichtige Informationen zur Verfügung gestellt.</p>

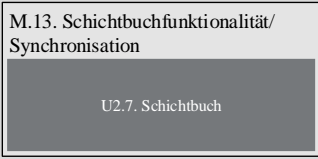
Generisch/optional	Generisch
Funktions- beschreibung	<p>Das Modul Wartungsmanagement regelt die Aktivitäten, welche die Wartung betreffen. Es umspannt viele dezentrale Aktivitäten, welche auf der jeweiligen Maschine vorhanden sind oder sein können, sowie viele Funktionen, die zentral optimale Wartungspläne erstellen und Wartungsmeldungen an das Auftragsmanagement weitergeben.</p> <p>Eine Kernfunktion des Moduls ist das Management und das Bereitstellen von Wartungszyklen/Wartungsplänen für bestimmte Komponenten der Maschine. Damit wird z. B. beschrieben, wann welches Bauteil gewechselt oder gewartet werden soll. Dies könnte sich nach einer bestimmten Anzahl an Vorgängen und/oder nach einer bestimmten Zeitspanne richten. Die Daten werden dabei im Zusammenspiel mit dem aktuellen Status aus dem Condition Monitoring (Modul M.16. Fehlererkennung & Condition Monitoring) angepasst, sodass der wirtschaftlich gesehen beste Moment für die Wartung angepeilt wird. Dabei können die Wartungspläne auch manuell geändert werden.</p> <p>Der automatisierte Teil des Prozesses und die daraus berechneten Wartungspläne können dezentral auf der Maschine stattfinden, indem dort lokal Wartungsgrenzen angepasst werden und der Wartungsplan dezentral gehalten wird. Dieser lokale Wartungsplan bedarf jedoch ggfs. einer zentralen Synchronisation. Diese spielt eine wichtige Rolle, sofern das Wissen über Wartungsgrenzen und Abnutzung über mehrere Maschinen und Anlagen geteilt werden soll. Wird erkannt, dass ein Bauteil besonders häufig gewartet werden muss, so können die Wartungsgrenzen bei allen ähnlichen Maschinen angepasst werden.</p> <p>Aufgrund der Erkenntnisse und Wartungspläne werden Wartungsaufträge in Verbindung mit Modul M.9. Auftragsverwaltung/Priorisierung erstellt. Dabei besitzen diese Wartungsaufträge meist einen frühesten und spätesten Zeitpunkt, an dem dieser Auftrag durchgeführt werden sollte. Die Einordnung in den bestehenden Instandhaltungsplan wird vom Modul M.9. Auftragsverwaltung/Priorisierung erledigt. Das Zeitfenster der Wartung wird jedoch von dem vorliegenden Modul berechnet.</p> <p>Nach Fertigstellung eines Auftrages, bei dem Wartungstätigkeiten durchgeführt worden sind, werden zum einen nicht nur die Wartungspläne in Abhängigkeit der festgestellten Abnutzung angepasst, sondern zum anderen auch die Wartungszähler zurückgesetzt. Als Zähler sind Werte gemeint, die den Status von Wartungszyklen darstellen. Dies kann ein zeitlicher Wert sein oder auch ein Wert, welcher Iterationen einer Produktion oder Ähnliches darstellt.</p> <p>Im Sinne eines CPS ist es angedacht, dass sich das Wartungsmanagement auch dezentral auf der Maschine befindet, so dass eine Maschine von selbst erkennen kann, ob und wann sie und ihre Komponenten Wartung bedürfen und eine entsprechende Meldung an die zuständigen Personen und Systeme senden kann. Des Weiteren besteht die Möglichkeit, dass die Maschine teils autonom bestimmte Wartungsvorgänge selbst vornimmt. Die Implementierung dieser Funktionalitäten hängt stark von der Anpassbarkeit der Maschinen ab.</p> <p>Des Weiteren sind in dem Modul die Wartungshistorien vorhanden und darstellbar. Dedizierte Wartungsberichte können als optionale Funktion, verbunden mit oder ohne den zugehörigen Auftrag, in dem Modul erstellt und hinterlegt werden.</p> <p>Voraussetzungen für die Funktionsfähigkeit</p> <p>Je nach Ausprägung der Funktionalitäten müssen die beteiligten Maschinen so angepasst werden können, dass ein dezentraler Wartungsplan mit dem zentralen Wartungsplan synchronisiert werden kann, und, dass Wartungsmeldungen von diesen – ähnlich wie Fehlermeldungen – versendet werden können. Können die Funktionalitäten nicht dezentral erstellt werden, so müssen die nötigen Daten zur</p>

	Beurteilung des Zustandes der Maschine von diesen direkt bzw. indirekt über das Modul M.16. Fehlererkennung & Condition Monitoring gelesen und zentral gepflegt werden.
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • (Automatisch) Wartungsbedarf melden bzw. anzeigen • Zurücksetzen und Einstellen von Wartungszählern • Erstellen von dedizierten Wartungsberichten • Wartungszyklen für Maschinen und Bauteile festlegen • (Automatisches) Anpassen von Wartungsplänen anhand von Abnutzungsveränderungen • Automatischer Abgleich des zentralen Wartungsplans mit Wartungsmeldungen und Wartungsplänen der Maschinen • Bestimmung von Schwellwerten für Condition Monitoring in Abhängigkeit von der Abnutzung • Selbstständige Wartungslösung auf Maschine (z. B. Selbstreinigung)
Verbindungen zu anderen Modulen	<p>B.3. Tätigkeitsmanagement der Instandhaltung:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.13. Schichtbuchfunktionalität/Synchronisation <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.1. Handlungsleitfäden/Checklisten/Prüflisten • Modul M.5. Anlagenübersicht und -auswertung • Modul M.16. Fehlererkennung & Condition Monitoring • Modul M.19. Predictive Maintenance • Modul M.29. QR-CODE auslesen
Verbundene Aktivitäten	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A1 – Inspektion/Wartung Total Productive Maintenance • Aktivität A2 – Inspektion Wartung Instandhalter • Aktivität A3 – Instandsetzung • Aktivität A4 – Vereisung der Windenergieanlage (WEA) • Aktivität A5 – Ereignisgesteuerte Instandsetzung • Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung • Aktivität A7 – Zustandsabfrage Wartung (Condition Monitoring)
Aufteilung nach Entität	Dieses Modul befindet sich auf der Anlage und Backend. Zugreifbar sind die Informationen auch vom mobilen Endgerät aus.

6.2.12 M.12. Auftragsmanagement in Verbindung mit ERP

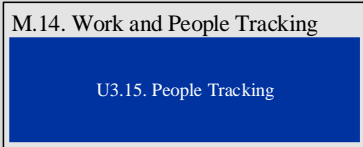
Komponente	M.12. Auftragsmanagement in Verbindung mit ERP
Darstellung in Referenzarchitektur	<p>M.12. Auftragsmanagement in Verbindung mit ERP</p> <p>G.24.2. Auftragsmanagement in Verbindung mit ERP</p> <p>U1.3. Auftragsmanagement zusammen mit ERP</p> <p>U4.34. Kommunikation mit ERP-System</p>
Zweck/Ziel	Dieses Modul stellt eine Erweiterung des Moduls M.9. Auftragsverwaltung/Priorisierung dar. Es dient dazu, das Auftragsmanagement mit einem bereits vorhandenen ERP-Systems zu verbinden und den notwendigen Datenaustausch zu managen.
Generisch/optional	Generisch, außer es besteht kein ERP.
Funktionsbeschreibung	<p>Soweit ein ERP-System vorliegt, werden die nötigen Daten aus dem IIS-eigenen Auftragsmanagement automatisch mit dem ERP Auftragsmanagement synchronisiert bzw. abgeglichen. Dadurch können Funktionalitäten aus dem ERP übernommen und gegebenenfalls in bestehende CMMS innerhalb des ERPs eingebunden werden.</p> <p>Die exakte Art der Synchronisation/Verbindung kann nur in Abhängigkeit von den verwendeten Lösungen bestimmt werden.</p> <p>Voraussetzungen für die Funktionsfähigkeit</p> <p>Es muss eine Verbindung zum Auftragsmanagement des ERPs bestehen. Das Bereitstellen aller benötigten Daten sollte ohne Zeitverzug, mehrfacher Datenhaltung oder Buchungsfehler stattfinden.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Aufträge aus ERP erhalten, abgleichen und generell synchronisieren • Auftragsdaten aus Instandhaltungssystem und ERP nach Beendigung abgleichen
Verbindungen zu anderen Modulen	<p>Baustein B.2. Tätigkeitsmanagement der Instandhaltung:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Dieses Modul befindet sich im Backend.

6.2.13 M.13. Schichtbuchfunktionalität/Synchronisation

Komponente	M.13. Schichtbuchfunktionalität/Synchronisation
Darstellung in Referenzarchitektur	
Zweck/Ziel	Das Modul Schichtbuchfunktionalität stellt die Funktionalität eines Schichtbuches zur Verfügung bzw. synchronisiert die Daten zwischen einem Schichtbuch und dem vorliegenden IIS.
Generisch/optional	Optional
Funktionsbeschreibung	<p>Der Zweck eines Schichtbuches ist es, der aktuellen und nachfolgenden Schicht, die bspw. mit der Produktion von bestimmten Teilen beschäftigt ist, den aktuellen Zustand des Produktionsprozesses und damit einhergehend der Anlagen zu verschaffen. Dies soll in einer möglichst einfachen Art und Weise geschehen.</p> <p>Zur Umsetzung dieser Funktionalität müssen diverse Daten aus anderen Modulen extrahiert werden, um diese in einem Schichtbuch einzutragen und darzustellen. Die zu extrahierenden Daten variieren von Unternehmen zu Unternehmen, typischerweise umfassen sie folgende Informationen:</p> <ul style="list-style-type: none"> • Aktuelle Meldungen in dieser Schicht • Aktuelle offene Meldungen anzeigen • Auflistung und • Protokollierung der durchgeführten Maßnahmen • Anlagenauswertung <p>Diese werden aus den jeweiligen Modulen, die diese Informationen beinhalten geladen und in diesem Modul grafisch übersichtlich dargestellt. Das besondere Augenmerk dieses Moduls liegt auf der einfachen Bedienung und Darstellung auf einer Touchoberfläche.</p> <p>Weitere mögliche Funktionen wären die</p> <ul style="list-style-type: none"> • Schichtübergabe mit dedizierten Protokoll mit Signatur • Abnahme der durchgeführten Maßnahmen <p>Diese Funktionen müssen durch das Modul implementiert oder in Verbindung mit dem optionalen Modul M.25. Signierfunktion integriert werden.</p> <p>Neben dem Fall, dass diese Funktionen selbst implementiert werden, muss es die Möglichkeit geben, die Daten des vorliegenden IIS automatisch mit bestehenden Schichtbüchern zu synchronisieren.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Darstellung aktuelle Meldungen dieser Schicht für die zugehörigen Anlagen • Aktuelle offene Meldungen • Auflistung der durchgeführten Maßnahmen
Verbindungen zu anderen Modulen	<p>B.3. Tätigkeitsmanagement der Instandhaltung:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.5. Anlagenübersicht und -auswertung

	<ul style="list-style-type: none"> • Modul M.10. Mitarbeitermanagement • Modul M.11. Wartungsmanagement <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.16. Fehlererkennung & Condition Monitoring • Modul M.25. Signierfunktion
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Backend und mobiles Endgerät.

6.2.14 M.14. Work and People Tracking

Komponente	M.14. Work and People Tracking
Darstellung in Referenzarchitektur	 <p>The diagram shows a box labeled 'M.14. Work and People Tracking' containing a smaller box labeled 'U3.15. People Tracking'.</p>
Zweck/Ziel	Das Modul dient dazu, den Aufenthaltsort und die Arbeit von internen wie auch externen Instandhaltern zu tracken, um anhand ihrer Position Arbeitsaufträge zu vergeben und den Arbeitsfortschritt nachzuvollziehen.
Generisch/optional	Optional
Funktionsbeschreibung	<p>Der Instandhalter muss anhand eines mobilen Endgerätes seine Position kontinuierlich an das Backend senden, damit der Status des Instandhalters samt seiner Position angezeigt werden kann. Dazu muss das mobile Endgerät entweder in einem Intervall oder – zur Reduzierung des Datenverkehrs – bei Änderung der Position seine Koordinaten an das Backend senden.</p> <p>In Verbindung mit dem Modul M.10. Mitarbeitermanagement kann mit diesen Daten eine Übersicht erstellt werden, in der alle Instandhalter samt ihrer Positionen dargestellt werden. Anhand dieser Position kann der Instandhalter, der sich in nächster Nähe befindet, bei auftretenden kritischen Fehlern direkt zu dem Auftrag gesendet werden. Weiter kann in Verbindung mit dem Modul M.9. Auftragsverwaltung/Priorisierung die Position in die Optimierung der Instandhaltungsplanung integriert werden. Diese Funktionen sind dann vonnöten, wenn die Anlagen und die Instandhalter, eine sehr hohe räumliche Distanz zueinander aufweisen.</p> <p>Darüber hinaus kann durch die Funktion der Arbeitsort des Instandhalter getrackt werden. Mit der Positionübergabe mitsamt Zeitstempel können in Verbindung mit dem Modul M.9. Auftragsverwaltung/Priorisierung die Zeiten und die Orte der Auftragsabwicklung automatisch gespeichert werden. Diese Funktionalität kann dabei bei Bedarf integriert werden.</p> <p>Voraussetzungen für die Funktionsfähigkeit</p> <p>Ein mobiles Endgerät muss die Position des Instandhalter übermitteln können. Des Weiteren dürfen datenschutzrechtliche Bedenken in dem Unternehmen nicht gegen diese Funktion sprechen.</p>

Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Positionsbestimmung des Instandhalters • Änderungen der Position bestimmen • Weiterleitung der Position des Instandhalters • Protokollierung der Positionen • Arbeitstracking
Verbindungen zu anderen Modulen	Baustein B.3. Tätigkeitsmanagement der Instandhaltung: <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.10. Mitarbeitermanagement • Modul M.15. Arbeitszeiterfassung
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Backend und mobiles Endgerät.

6.2.15 M.15. Arbeitszeiterfassung

Komponente	M.15. Arbeitszeiterfassung
Darstellung in Referenzarchitektur	
Zweck/Ziel	<p>Das Modul dient der Arbeitszeiterfassung der Instandhalter. Wird die Instandhaltung als Service angeboten, kann durch die Erfassung der Arbeitszeit die Abrechnung mit dem Kunden erleichtert und durch diese Genauigkeit Vertrauen aufgebaut werden. Falls die Instandhaltung intern betrieben wird, kann hierdurch die Arbeitszeit der Instandhalter automatisch dokumentiert werden.</p>
Generisch/optional	Optional
Funktionsbeschreibung	<p>Das Modul stellt eine Zusatzfunktionalität dar, die zur Protokollierung der eigenen Arbeitszeit genutzt wird. Das Augenmerk liegt darauf, dass der Instandhalter seinen Arbeitsstart und sein Arbeitsende protokollieren kann. Weiterhin sollen seine Aktivitäten während der Arbeitszeit hinterlegt werden können. Dazu wird ein einfaches Bedienfeld bereitgestellt, an dem die Arbeit gestoppt und gestartet werden kann. Nach Beendigung der Arbeit können automatisch Notizen eingefügt werden. All diese Informationen sind für einen gewissen Zeitraum im Nachhinein durch den Instandhalter bearbeitbar.</p> <p>Viele der Informationen zu Aufträgen, abgearbeiteten Schritten und weiteren Informationen können auch automatisiert in Verbindung mit den Modulen M.9. Auftragsverwaltung/Priorisierung, M.10. Mitarbeitermanagement und Modul M.1. Handlungsleitfäden/Checklisten/Prüflisten integriert werden. Zu der geleisteten Arbeitszeit wären dann die Informationen aus diesen Modulen verfügbar, sodass erledigte Arbeitsschritte und Aufträge nicht händisch protokolliert werden müssten.</p>

Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Arbeitszeitmanagement • Automatische Tracking der Arbeitszeit • Bearbeitung der Arbeitszeiten • Extraktion und Integration der Auftragsinformationen in Daten zu Arbeitszeit • Extraktion von Arbeitszeitdaten aus Auftragsinformationen (auch aus Metadaten) • Hinterlegen von Kommentaren und Notizen zu Arbeitstagen und -zeiten
Verbindungen zu anderen Modulen	<p>B.3. Tätigkeitsmanagement der Instandhaltung:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.10. Mitarbeitermanagement • Modul M.14. Work and People Tracking <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.1. Handlungsleitfäden/Checklisten/Prüflisten • Modul M.22. Reisekostenabrechnung • Modul M.23. Serviceprotokolle
Verbundene Aktivitäten	Grundaktivität
Aufteilung nach Entität	Backend und mobiles Endgerät.

6.2.16 M.16. Fehlererkennung & Condition Monitoring

Komponente	M.16. Fehlererkennung & Condition Monitoring														
Darstellung in Referenzarchitektur	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center; background-color: #f2f2f2;">M.4. Anlageninformationen</td> </tr> <tr> <td style="background-color: #4CAF50; color: white; text-align: center;">G.10.1. Anzeige von Maschinenplänen (Steuerungspläne, SPS, etc.)</td> <td style="background-color: #4CAF50; color: white; text-align: center;">G.16.1. Bauteilübersicht der Anlagen verfügbar</td> </tr> <tr> <td style="background-color: #4CAF50; color: white; text-align: center;">G.17.1. Herstellerunterlagen der Anlage verfügbar</td> <td style="background-color: #E91E63; color: white; text-align: center;">U1.7. Anzeige Maschinenpläne</td> </tr> <tr> <td style="background-color: #E91E63; color: white; text-align: center;">U1.15. Bauteilinformationen für Anlagen verfügbar</td> <td style="background-color: #E91E63; color: white; text-align: center;">U1.16. Anlagenbeschreibung v. Hersteller</td> </tr> <tr> <td style="background-color: #9E9E9E; color: white; text-align: center;">U2.5. Zugriff Herstellerunterlagen</td> <td style="background-color: #9E9E9E; color: white; text-align: center;">U2.16. Anzeige Maschinenpläne</td> </tr> <tr> <td style="background-color: #9E9E9E; color: white; text-align: center;">U2.31. Bauteilinformationen</td> <td style="background-color: #9E9E9E; color: white; text-align: center;">U2.35. Push Benachrichtigung Updates bei Änderungen von HW/SW Komponenten d. Anlage</td> </tr> <tr> <td style="background-color: #673178; color: white; text-align: center;">U4.38. Bauteilinformationen verfügbar</td> <td style="background-color: #673178; color: white; text-align: center;">U4.23. Maschinenpläne im System verfügbar</td> </tr> </table>	M.4. Anlageninformationen		G.10.1. Anzeige von Maschinenplänen (Steuerungspläne, SPS, etc.)	G.16.1. Bauteilübersicht der Anlagen verfügbar	G.17.1. Herstellerunterlagen der Anlage verfügbar	U1.7. Anzeige Maschinenpläne	U1.15. Bauteilinformationen für Anlagen verfügbar	U1.16. Anlagenbeschreibung v. Hersteller	U2.5. Zugriff Herstellerunterlagen	U2.16. Anzeige Maschinenpläne	U2.31. Bauteilinformationen	U2.35. Push Benachrichtigung Updates bei Änderungen von HW/SW Komponenten d. Anlage	U4.38. Bauteilinformationen verfügbar	U4.23. Maschinenpläne im System verfügbar
M.4. Anlageninformationen															
G.10.1. Anzeige von Maschinenplänen (Steuerungspläne, SPS, etc.)	G.16.1. Bauteilübersicht der Anlagen verfügbar														
G.17.1. Herstellerunterlagen der Anlage verfügbar	U1.7. Anzeige Maschinenpläne														
U1.15. Bauteilinformationen für Anlagen verfügbar	U1.16. Anlagenbeschreibung v. Hersteller														
U2.5. Zugriff Herstellerunterlagen	U2.16. Anzeige Maschinenpläne														
U2.31. Bauteilinformationen	U2.35. Push Benachrichtigung Updates bei Änderungen von HW/SW Komponenten d. Anlage														
U4.38. Bauteilinformationen verfügbar	U4.23. Maschinenpläne im System verfügbar														
Zweck/Ziel	<p>Dieses Modul dient der Fehlererkennung bzw. -diagnose und des Condition Monitorings auf Maschinen- bzw. Anlagenebenen. Hierzu zählt, neben dem Erkennen des Zustandes einer Anlage/Maschine/Bauteils, auch die Möglichkeit diesen abzurufen und Fehler- oder Wartungsmeldungen automatisch an zuständige Empfänger in entsprechenden Formaten zu versenden. Zusätzlich soll dieses Modul illustrieren, wie diese Funktionalität im Hinblick auf die effiziente Ausnutzung von Dezentralität im Sinne von CPS umgesetzt werden kann.</p> <p>Es wird daher ein generisches Diagnose- und Condition Monitoring System beschrieben, das auf die Möglichkeiten von CPS eingeht und so im Kontext zukünftiger Instandhaltungsszenarien nutzbar ist. Ziel dieses Systems ist es, den Grad an reaktiver Instandhaltung zu senken und die Transparenz im Anlagenbetrieb zu erhöhen.</p>														
Generisch/optional	Generisch														
Funktionsbeschreibung	<p>Das Modul M.16. Fehlererkennung & Condition Monitoring erzeugt Warnungen/Meldungen über existierende Fehler oder einen zukünftig entdeckten Wartungsbedarf aufgrund von Condition Monitoring und Fehlererkennung. Es ist in der Lage, diese an weitere Komponenten zu versenden. Hinsichtlich der technischen Umsetzung kann ein Diagnosesystem bspw. auf einer dezentralen, standardisierten Recheneinheit platziert sein, wenn die Maschine diese Erweiterung nicht bereits unterstützt. Das Modul wird insbesondere aufgrund folgender Punkte benötigt:</p> <ul style="list-style-type: none"> • Diagnose: Bei allen auftretenden Problemen müssen Instandhalter genaue Fehlerursachen identifizieren. Die Lösung ist ein intelligentes Diagnosesystem, das einen Soll-/Istwert-Abgleich für eine effiziente Fehlerdiagnose anbietet. Das Vorgehen bzgl. der Diagnose beschränkt sich nicht nur auf Fehler, sondern auch auf die Überprüfung von physikalischen Messgrößen durch das Condition Monitoring. Hier werden Abweichungen identifiziert, die „noch“ nicht als Fehler klassifiziert sind. Diese 														

Abweichungen können jedoch auch Instandhaltungsmaßnahmen nach sich ziehen.

- Unterstützung der Diagnose von komplexen Fehlerbildern und Zusammenhängen an Maschinen und Anlagen, die von klassischen Diagnosesystemen bisher nicht unterstützt oder angeboten werden.
- Automatisches Versenden von Fehlermeldungen und Wartungsbedarfen muss durch das Modul unterstützt werden.
- Relevante Daten müssen aus einer Vielzahl proprietärer Diagnosesysteme gesammelt werden. Informationen der Maschinen und Anlagen bzw. Diagnosefunktionalitäten stehen durch das Modul standardisiert auf Anlagenebene zur Verfügung, so dass jedes relevante Objekt bzw. CPS Informationen anbieten bzw. beziehen kann. Dazu kann ein leichtgewichtiges Format verwendet werden. Diese Daten müssen für die restlichen Module bereitgestellt werden.

Die Berechnung von typischen Kennzahlen der Maschinen und Anlagen, wie bspw. Mean Time To Repair (MTTR) oder Mean Time Between Failure (MTBF), ist allerdings weiterhin Aufgabe eines evtl. bereits vorhandenem MES-Systems bzw. des Moduls M.5. Anlagenübersicht und -auswertung des IIS. Dazu sind nötigen Kennzahlen zu erstellen bzw. aus anderen System zu integrieren. Das gleiche gilt für eventuell vorhandene Condition Monitoring Lösungen.

Die Fehlerdiagnose unterstützt zwei Arten der Diagnose, die im Bereich Instandhaltung notwendig sind. Diese Teile werden nachfolgend getrennt voneinander beschrieben.

Maßnahmediagnose auf Basis eines kontinuierlichen Soll-Ist-Vergleichs

Für die Durchführung der Fehler- und Wartungserkennung einer Maschine oder Komponente ist es einerseits essentiell den Zustand dieser zu bestimmen. Andererseits muss dieser Zustand regelmäßig einem Soll-Ist-Vergleich unterzogen werden, damit sichergestellt ist, dass der Zustand einen „normalen“ Betrieb ermöglicht. Dafür ist es notwendig vorab Zustände zu definieren, die die Grenzen des Solls bestimmen. Der abstrakte Begriff des Zustandes kann definiert werden als „die Gesamtheit aller Eigenschaften oder Attribute, die zur Abgrenzung und Unterscheidung des jeweils betrachteten Objekts von anderen Objekten nötig sind“ (Endig et al., 2010, S. 129). In der DIN 31051:2012-09 wird der Zustand indirekt über den Abnutzungsvorrat beschrieben. Der Zustand wäre hier die genaue Einschätzung der Koordinaten der Maschine auf der Abnutzungskurve. Mögliche Sollwerte wären Punkte kurz vor dem Verbrauch des Abnutzungsvorrates und damit kurz vor dem Ausfall der Maschine (vgl. Kapitel 3.2).

Im einfachsten Fall können Messwerte von Sensoren direkt als Zustandsabstraktion verwendet werden. Durch die kontinuierliche Messwertaufnahme in Condition Monitoring Systemen hat man Zugang zu dem zeitlichen Verlauf des aktuellen Zustandes. Dieser kann mit einem Soll-Wert bzw. Referenzwert verglichen werden. Wollschlaeger et al. (2015, S. 3) unterscheiden drei Methoden für die Implementierung von Sollwerten bzw. Sollfunktionen. Zum einen kann der Referenz- bzw. Soll-Wert als absolutes Intervall mit oberer Grenze (oG) und unterer Grenze (uG) angesehen werden. Dabei darf sich der Ist-Wert nur in diesem Intervall bewegen. Ein weiterer Modus ist die Hinterlegung einer Referenzfunktion mit Eingriffsgrenzen, die mithilfe von prozentualen Auf- und Abschlägen berechnet werden. Abschließend legt er einen allgemeinen Fall vor, in dem die Grenzen dynamisch und in Abhängigkeit der Zeit eingestellt werden.

Diese vorgestellten Methoden lassen sich dahingehend erweitern, dass die Einstellung der Grenzwerte nicht nur über die Zeit gesteuert werden, sondern Taskorientiert. Unter Taskorientiert wird hier verstanden, dass die Grenzwerte bezogen auf den gerade ausgeführten Arbeitsschritt (Task) festgelegt werden und nicht zeitbasiert. Damit folgt das Condition Monitoring dynamisch den

Arbeitsschritten und kann intelligent auf Änderung im Ablauf reagieren. Dies ermöglicht eine höhere Flexibilität, da auch bei einer Umstellung des Produktionsprozesses die Grenzwerte nicht angepasst werden müssen. Eine Kombination der angesprochenen Methoden ist denkbar.

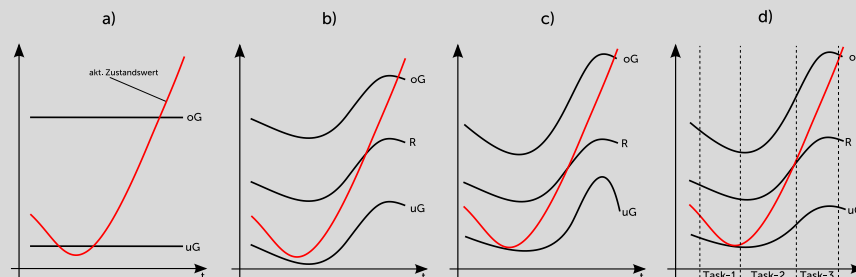


Abbildung 20: Soll-Ist Vergleich auf Basis dynamischer Prozesseingriffsgrenzen

Quelle: In Anlehnung an Wollschlaeger et al. (2015, S. 3)

Abbildung 20 zeigt die drei Fälle nach Wollschlaeger et al. (2015) (a - c) sowie den Taskorientierten Ansatz (d). Die rote Linie zeigt den aktuellen Verlauf des Ist-Zustandes an. Mit (R) wird der Verlauf des Soll-Wertes dargestellt. Da im ersten Fall (a) ein zeitlich konstantes Intervall angegeben ist, gibt es hier keine Referenzkurve. In der zweiten Grafik werden die oG und uG durch Prozentsätze vom Referenzverlauf berechnet und sind somit von R abhängig. In Abbildung (c) besitzen die Grenzkurven komplexere Funktionen, die abhängig vom Produktionsprozess gewählt werden. Abschließend wird in (d) der Taskorientierte Ansatz dargestellt. Die Grenzkurven sind hier bezogen auf die einzelnen Tasks und nicht mehr nur zeitabhängig.

Weiterhin müssen mehrere Sollwerte festgelegt werden. Bestimmte Sollwerte zeigen eine Wartung an, wohingegen ein Überschreiten anderer Sollwerte eine sofortige Instandhaltung oder gar einen Stopp der Produktion nach sich ziehen müsste. Bei der Implementierung ist es wichtig, die nötigen Informationen mitsamt evtl. Fehlercodes direkt an die zuständigen Module zu senden, damit Instandhaltungsaufträge erstellt und Wartungspläne angepasst werden können.

Fehler-/Maßnahmediagnose durch herstellerepezifische Fehlercodes

Die Fehler-/Maßnahmenerkennung soll zusätzlich durch die Nutzung der herstellerepezifischen Anwendungen bzw. Diagnoseintelligenz geschehen. Aufgrund der heterogenen Schnittstellen und Logik dieser Lösungen gilt es zu untersuchen, wie diese in das IIS integrierbar sind. In diesem Fall sind die Ausgangspunkte für das Diagnosesystem die Fehlerbeschreibungen bzw. Fehlercodes der Maschinen und Anlagen und nicht Sensor- oder Betriebsdaten. Diese müssen entweder direkt weitergeleitet werden oder anhand des vorgestellten Konzepts überführt werden. Beispiele für Architekturen solcher Systeme finden sich bspw. bei Fleischmann et al. (2016).

Neben der Diagnose muss das Modul auch in diesem Fall bei erkanntem Handlungsbedarf automatisch Fehlermeldungen und unerwartete Wartungsbedarfe an das Modul M.9. Auftragsverwaltung/Priorisierung senden, woraus dort Maßnahmenaufträge erstellt werden.

Voraussetzungen für die Funktionsfähigkeit

Das Modul und damit das Diagnosesystem benötigen lesenden Zugriff auf die Maschine bzw. auf die aktuellen Logdateien, die Betriebsdatenerfassung oder das Modul muss direkt in die Maschine integriert werden. Die Auswertung sollte in Echtzeit oder mit kurzer Verzögerung erfolgen. Weiterhin muss das Diagnosesystem Fehlermeldungen und Aufträge/Meldungen absenden können.

Bei Nutzung der Hersteller-codes müssen diese zusätzlich auf Anlagenebene vorhanden sein. Das Diagnosesystem benötigt folglich Zugang zu den wesentlichen

	<p>Log-Maschinendaten sowie Fehlercodes und -beschreibungen, um eine Diagnose zu ermöglichen.</p> <p>Insbesondere muss jedoch die Datenweitergabe an das Backend des IIS spezifiziert werden. Hier ist zu definieren, wie die Datenaggregatoren Informationen von den Maschinen und Anlagen beziehen.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Generierung und automatische Versendung von Fehlermeldungen • Generierung und Versenden von Maßnahmenaufträgen • Automatisches Erkennen und Bewerten von Soll/Ist-Abweichungen (Condition Monitoring) und plötzlich auftretenden Fehlern auf Anlage(n) • Auslesen von proprietären Diagnosesystemen • Bereitstellung von relevanten Maschinendaten • Soll-Ist-Vergleich des Zustandes der einzelnen Komponenten
Verbindungen zu anderen Modulen	<p>Sobald das Modul an der Maschine Fehler, Warnungen oder sonstige Maßnahmenhinweise erkennt, muss eine Meldung generiert und an das Auftragsmanagement versandt werden, damit ein Maßnahmenauftrag erstellt werden kann. Im Zusammenspiel mit einem möglichen Predictive Maintenance kann auch die Diagnose verbessert werden. Daten über Grenzwerte oder Soll-/Ist-Abgleiche können daher getauscht werden. Dies stellt jedoch eine optionale Verbindung dar.</p> <p>B.4. Fehlermanagement:</p> <ul style="list-style-type: none"> • Modul M.17. Fehlerdatenbank • Modul M.18. Fehlermeldung und -darstellung • Modul M.19. Predictive Maintenance <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.1. Handlungsleitfäden/Checklisten/Prüflisten • Modul M.5. Anlagenübersicht und -auswertung • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.11. Wartungsmanagement • Modul M.13. Schichtbuchfunktionalität/Synchronisation • Modul M.29. QR-CODE auslesen
Verbundene Aktivitäten	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A1 – Inspektion/Wartung Total Productive Maintenance • Aktivität A2 – Inspektion Wartung Instandhalter • Aktivität A3 – Instandsetzung • Aktivität A4 – Vereisung der Windenergieanlage (WEA) • Aktivität A5 – Ereignisgesteuerte Instandsetzung • Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung • Aktivität A7 – Zustandsabfrage Wartung (Condition Monitoring)
Aufteilung nach Entität	<p>Das hier vorgestellte Modul wird sich an der Maschine befinden.</p>

6.2.17 M.17. Fehlerdatenbank

Komponente	M.17. Fehlerdatenbank
Darstellung in Referenzarchitektur	
Zweck/Ziel	<p>Dieses Modul stellt eine Verbindung zwischen den Anlagen, in welchen Fehler auftreten, und den zugehörigen Fehlerkodierungen her, sofern diese nicht selbst aussagefähige Fehlermeldungen beinhalten. Darüber hinaus wird eine Historie von aufgetretenen Fehlern angelegt, die von anderen Modulen ausgelesen und ausgewertet werden können. Die Datenbank ist hierbei als wichtiger Teil des Wissens- und Fehlermanagements zu sehen. Sie ermöglicht Fehler festzuhalten, zu dokumentieren und auszuwerten. Anhand dieser Daten können Maßnahmen eingeleitet werden, um wiederkehrende Fehler zu verhindern oder deren Lösung zu beschleunigen. Dazu wird die Datenbank initial mit den schon bekannten Fehlern und Ursachen befüllt und der Prozess des Sammelns dieser Daten im normalen Arbeitsablauf kontinuierlich weitergeführt.</p>
Generisch/optional	Generisch
Funktionsbeschreibung	<p>Voraussetzung für die systematische Fehlererfassung ist die eindeutige und einheitliche Beschreibung von Fehlerereignissen mit einem eindeutigen Fehlerschlüssel. Dieser soll die Daten aus den Fehlercodes der Hersteller bzw. der einzelnen Anlagen mit einem Fehlercode des IIS verknüpfen. Dieses Vorgehen führt dazu, dass keine Dopplungen von Fehlercodes bei einer Vielzahl von Anlagen entstehen können, die unterschiedliche Bedeutungen haben. Es müssen also Maschinenhersteller spezifische Fehlercodes auf IIS interne Fehlercodes transformiert werden, so dass Fehler eindeutig zuzuordnen sind.</p> <p>Dies geschieht, indem aus Fehlercode plus einem eindeutigen, unternehmensinternen Identifikator der Anlage, ein Schlüssel für den internen Fehlercode des IIS gebildet wird.</p> <p>Eindeutige Fehlercodes hinzufügen</p> <p>Die Bildung der eindeutigen Fehlercodes kann auf zwei Arten geschehen:</p> <p>Zum einen sollte die manuelle Eingabe der Fehlercodes möglichst einfach über eine grafische Schnittstelle möglich sein. Vorstellbar wäre z. B. ein Dateneingabeformular, in der die Fehlerdaten direkt in die einzelnen Formularspalten eingetragen werden können. Bisher nicht bekannte Fehler einer bestehenden oder neuen Anlage könnten dabei so eingepflegt werden. Des Weiteren ist auch eine automatisierte Befüllung seitens der Anlagen möglich, d. h. es sollen automatisch Einträge zu neuen Fehlercodes erstellt werden, falls unbekannte Fehlercodes auftreten, die bisher nicht in der Fehlerdatenbank hinterlegt sind. Dies kann ggfs. nach Erstellen eines Eintrags einer (manuellen) Prüfung unterzogen werden, um die Konsistenz der Datenbank zu bestätigen.</p> <p>Fehlermeldungen hinzufügen</p> <p>Neben dieser rein technischen Datenhaltung und -identifikation beinhaltet dieses Modul noch eine Fehlerdatenbank, welche die vergangenen Fehler von allen Anlagen speichert. Neben dem Hinzufügen von eindeutigen Fehlercodes, dient daher das Modul dazu, Fehlerhistorien aufzubauen und die Informationen der jeweiligen</p>

	<p>Fehler zu hinterlegen. Dazu werden alle eindeutig erkannten Fehlermeldungen aus dem Modul M.16. Fehlererkennung & Condition Monitoring direkt mit dem jeweiligen Fehleridentifikator in die Datenbank übertragen.</p> <p>Diese Datenbank kann anhand verschiedener Kategorien durchsucht und gefiltert werden. Sie bietet zusätzlich die Möglichkeit, die Fehler mit Informationen aus den Modulen M.9. Auftragsverwaltung/Priorisierung und M.18. Fehlermeldung und -darstellung anzureichern. Der Unterschied zu letzterem Modul besteht darin, dass in dem vorliegenden Modul Fehlerdaten und Informationen historisch archiviert und zugeordnet werden. Es sind jedoch nicht alle potentiellen Informationen zu Fehlern, wie bspw. 3D-Darstellung in dieser hinterlegt, um (kontextabhängige) Fehlerdarstellungen von der reinen Historie zu separieren.</p> <p>Nachfolgend werden Begriffe vorgeschlagen, die zur Fehlerkategorisierung dienen könnten. Durch die Fehlerdatenbank mitsamt Historie können dann Aussagen über Häufigkeiten von Fehlern und Fehlerschwerpunkten getroffen werden. Die Klassen des Fehlerschlüssels sind:</p> <ul style="list-style-type: none"> • Fehlercodierung des Herstellers • Fehlerart und -schwere • Interne IIS-spezifische Anlagen-ID • Eine eindeutige Fehler-ID (interne Fehlerzuordnung, jede Art von Fehler besitzt genau eine ID) • Eine eindeutige Fehlernummer, welche sich von der Fehler-ID dadurch unterscheidet, dass mehrere identische Fehler mit einer Fehler-ID unterschiedliche Fehlernummern besitzen, um deren zeitliches und häufiges Auftreten zu unterscheiden. Jedes Auftreten eines Fehlers wird daher mit einer neuen Fehlernummer versehen • Uhrzeit/Datum (Timestamp) • Programmbezeichnung mit Versionsnummer bzw. Entwicklungsstand (falls Softwarefehler) • Fehlerstatus z. B. offen, in Analyse, in Behebung, erledigt oder auch keine Maßnahme ergriffen • Weitere Fehlerattribute und Informationen <p>Voraussetzungen für die Funktionsfähigkeit</p> <p>Die Fehlerdatenbank muss von fachkundigem Personal, i. d. R. Instandhalter, befüllt werden können. Falls Fehler auftreten, die sich noch nicht in der Datenbank befinden, müssen diese möglichst sofort nachgetragen werden. Um den Nutzen zu maximieren, sollte die Datenbank möglichst aktuell gehalten und darin enthaltene Informationen mit Hilfe von Funktionen aus anderen Modulen, z. B. Visualisierung aus Modul M.18. Fehlermeldung und -darstellung, genutzt werden.</p>
<p>Mögliche Unterfunktionen</p>	<ul style="list-style-type: none"> • (Automatisches) Eintragen und Bearbeiten von Fehlern in Datenbank • Auswertungsfunktionen für Fehler • Archivierung von Fehlerdaten • Suchfunktion für Fehler (Anlage, Fehlerart, etc.) • Fehlercode oder einfache Fehlermeldung in Verbindung mit vorhandenen Fehlerinformationen setzen
<p>Verbindungen zu anderen Modulen</p>	<p>Baustein B.4. Fehlermanagement:</p> <ul style="list-style-type: none"> • Modul M.16. Fehlererkennung & Condition Monitoring

	<ul style="list-style-type: none"> • Modul M.18. Fehlermeldung und -darstellung • Modul M.19. Predictive Maintenance <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.2. Dokumentenmanagement • Modul M.5. Anlagenübersicht und -auswertung • Modul M.9. Auftragsverwaltung/Priorisierung
Verbundene Aktivitäten	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A1 – Inspektion/Wartung Total Productive Maintenance • Aktivität A2 – Inspektion Wartung Instandhalter • Aktivität A3 – Instandsetzung • Aktivität A4 – Vereisung der Windenergieanlage (WEA) • Aktivität A5 – Ereignisgesteuerte Instandsetzung • Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung • Aktivität A8 – Verfügbarkeit (technisch/organisatorisch) • Aktivität A9 – Auswertung Fehlerhistorien einer Anlage
Aufteilung nach Entität	Dieses Modul befindet sich auf dem mobilen Endgerät wie auch im Backend.

6.2.18 M.18. Fehlermeldung und -darstellung

Komponente	M.18. Fehlermeldung und -darstellung														
Darstellung in Referenzarchitektur	<table border="1"> <thead> <tr> <th colspan="2" data-bbox="555 309 1302 342">M.18. Fehlermeldung und -darstellung</th> </tr> </thead> <tbody> <tr> <td data-bbox="555 342 932 456">G.2.1. Detaillierte Fehlermeldung und Störungsart</td> <td data-bbox="932 342 1302 456">G.28.2. Bereitstellung von 3D-Daten/3D-Darstellung der Anlage</td> </tr> <tr> <td data-bbox="555 456 932 571">U1.5. Fehlercodierung eindeutige Fehlerbeschreibung</td> <td data-bbox="932 456 1302 571">U1.24. 3D-Darstellung mit Fehlerlokalisierung</td> </tr> <tr> <td data-bbox="555 571 932 685">U2.1. Kurze prägnante Meldetexte bei Störfall + Fehlercode</td> <td data-bbox="932 571 1302 685">U3.12. 3D-Darstellung mit Fehlerlokalisierung</td> </tr> <tr> <td data-bbox="555 685 932 799">U3.4. Detaillierte Fehlerbeschreibung</td> <td data-bbox="932 685 1302 799">U3.9. Fehlerkategorisierung</td> </tr> <tr> <td data-bbox="555 799 932 913">U4.9. Anzeige Störungsmeldung</td> <td data-bbox="932 799 1302 913">U4.18. Bereitstellung von 3D-Daten</td> </tr> <tr> <td data-bbox="555 913 932 1037">U4.36. Störmeldung mitsamt Fehlermeldung und Zustand Anlage</td> <td data-bbox="932 913 1302 1037"></td> </tr> </tbody> </table>	M.18. Fehlermeldung und -darstellung		G.2.1. Detaillierte Fehlermeldung und Störungsart	G.28.2. Bereitstellung von 3D-Daten/3D-Darstellung der Anlage	U1.5. Fehlercodierung eindeutige Fehlerbeschreibung	U1.24. 3D-Darstellung mit Fehlerlokalisierung	U2.1. Kurze prägnante Meldetexte bei Störfall + Fehlercode	U3.12. 3D-Darstellung mit Fehlerlokalisierung	U3.4. Detaillierte Fehlerbeschreibung	U3.9. Fehlerkategorisierung	U4.9. Anzeige Störungsmeldung	U4.18. Bereitstellung von 3D-Daten	U4.36. Störmeldung mitsamt Fehlermeldung und Zustand Anlage	
M.18. Fehlermeldung und -darstellung															
G.2.1. Detaillierte Fehlermeldung und Störungsart	G.28.2. Bereitstellung von 3D-Daten/3D-Darstellung der Anlage														
U1.5. Fehlercodierung eindeutige Fehlerbeschreibung	U1.24. 3D-Darstellung mit Fehlerlokalisierung														
U2.1. Kurze prägnante Meldetexte bei Störfall + Fehlercode	U3.12. 3D-Darstellung mit Fehlerlokalisierung														
U3.4. Detaillierte Fehlerbeschreibung	U3.9. Fehlerkategorisierung														
U4.9. Anzeige Störungsmeldung	U4.18. Bereitstellung von 3D-Daten														
U4.36. Störmeldung mitsamt Fehlermeldung und Zustand Anlage															
Zweck/Ziel	<p>Das Modul Fehlermeldung und -darstellung dient der eindeutigen Beschreibung und Darstellung von Fehlern, so dass deren Ursachen schneller und akkurater erkannt werden können. Dies wird mithilfe exakter und detaillierter Fehlermeldungen mitsamt Kategorisierung und 3D-Darstellung erreicht. Mit diesen Informationen wird die Kritikalität von Fehlern und die Schritte zur Behebung dieser schneller ersichtlich, wodurch sich die Instandhaltungszeit verkürzt.</p>														
Generisch/optional	<p>Generisch</p>														
Funktionsbeschreibung	<p>Das Modul stellt z. T. funktionale Anforderungen an Fehlermeldungen sicher und reichert die meist simplen Fehlermeldungen aus Modul M.16. Fehlererkennung & Condition Monitoring mit weiteren Informationen an. Daher baut dieses Modul auf dem eindeutigen Fehlerformat für die Fehlerdarstellung auf, welches in Modul M.16. Fehlererkennung & Condition Monitoring implementiert wird. Mit diesem kann ein einheitliches Format zur Darstellung und Codierung der Fehler innerhalb des IIS genutzt werden. Neben der reinen Übersetzung und eines einheitlichen Formats des Fehlers, das auch in anderen Modulen angewendet werden muss, soll in diesem Modul vor allem eine Datenbasis aufgebaut werden, die Anlage, Fehlercode, Sensoren und Fehlerursache bzw. die allgemeinen Informationen und Beschreibungen miteinander in Verbindung setzen kann. Diese Datenquelle muss neben den reinen Informationen auch mit klar verständlichen Beschreibungen des Fehlers angereichert werden.</p> <p>Die Grundbasis der Fehlerdaten in diesem Modul entstammt dem Modul M.17. Fehlerdatenbank. In der Verbindung mit diesem Modul und den daraus extrahierten Fehlerdaten sollen hier alle weiteren relevanten Informationen hinterlegt werden. Dies muss, falls die Fehler nicht maschinenseitig eindeutig vom Hersteller beschrieben werden, in Verbindung mit dem Modul M.2. Dokumentenmanagement nachträglich über Eingabemasken eingegeben werden können.</p>														

	<p>Neben rein textueller Beschreibung wird der Fehler und deren Ursache so hinterlegt, dass eine 3D-Darstellung der Anlage und des Fehlers möglich ist. Das heißt abhängig von der 3D-Darstellung und deren Format, werden die Koordinaten des Fehlers innerhalb dieser Darstellung hinterlegt.</p> <p>Der Unterschied zu dem Modul Fehlerdatenbank besteht darin, dass neben den Anforderungen zur Darstellung, die Fehler aus der Datenbank in diesem Modul mit zusätzlichen Informationen angereichert werden, um visuelle, kontextsensitive Darstellungen und Auswertungen zu ermöglichen.</p> <p>Voraussetzungen für die Funktionsfähigkeit</p> <p>Die 3D-Daten der Anlagen müssen vorliegen und die Darstellung von Fehlerlokationen ermöglichen</p>
<p>Mögliche Unterfunktionen</p>	<ul style="list-style-type: none"> • Fehlerlokalisierung (grafisch) durchführen • Editieren von Fehlerinformationen • Anzeigen Fehlerbeschreibung • Anreicherung des Fehlercodes mit zusätzlichen eindeutigen Beschreibungen • Grafische Modellierung des Fehlers innerhalb der Anlage
<p>Verbindungen zu anderen Modulen</p>	<p>Baustein B.4. Fehlermanagement:</p> <ul style="list-style-type: none"> • Modul M.16. Fehlererkennung & Condition Monitoring • Modul M.17. Fehlerdatenbank <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.1. Handlungsleitfäden/Checklisten/Prüflisten • Modul M.2. Dokumentenmanagement • Modul M.9. Auftragsverwaltung/Priorisierung
<p>Verbundene Aktivitäten</p>	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A1 – Inspektion/Wartung Total Productive Maintenance • Aktivität A2 – Inspektion Wartung Instandhalter • Aktivität A3 – Instandsetzung • Aktivität A4 – Vereisung der Windenergieanlage (WEA) • Aktivität A5 – Ereignisgesteuerte Instandsetzung • Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung • Aktivität A9 – Auswertung Fehlerhistorien einer Anlage
<p>Aufteilung nach Entität</p>	<p>Dieses Modul befindet sich auf dem mobilen Endgerät wie auch im Backend.</p>

6.2.19 M.19. Predictive Maintenance

Komponente	M.19. Predictive Maintenance
Darstellung in Referenzarchitektur	
Zweck/Ziel	<p>Bei der Datenaufbereitung sollen im IIS neue Verfahren zur Prognose von anfallenden Wartungs- und Fehlerfällen entwickelt werden. Dies beginnt mit dem frühzeitigen Erkennen von potentiellen Fehlern und deren Ursachen aufgrund umfassender Analyseverfahren. Diese meist aus dem Machine Learning bzw. Data Mining stammende Verfahren werden auf möglichst umfassenden Daten der jeweiligen Maschinen angewendet. Dadurch ist es möglich (statistische) Zusammenhänge zu erkennen, die vorher nicht bekannt waren. Der Unterschied zu den Modulen M.16. Fehlererkennung & Condition Monitoring und M.11. Wartungsmanagement ist daher zum einen, dass im Vergleich zu ersterem keine Vorhersagen getroffen bzw. Prognosen berechnet werden. Zum anderen unterscheidet sich das vorliegende Modul in der Wahl der Mittel bzw. der umfassenden Datenanalyse, die auch Daten einbezieht, die nicht konkret dem Condition Monitoring zuzuordnen sind. Zusammen mit dem Absenden von geeigneten Statusmeldungen oder Warnmeldungen von Maschinen, werden Mechanismen definiert, die Prüfungsbedarf anzeigen, Wartungsintervalle ableiten und ggf. auch schon Zeitpläne für die Umsetzung im vorhandenen Rahmen festlegen.</p> <p>Das Modul Predictive Maintenance wird daher für die vorzeitige Erkennung von Instandhaltungsbedürfnissen diverser Maschinen benötigt. Teil dieses Moduls sind</p> <ul style="list-style-type: none"> • die Klassifikation, ob Instandhaltungsbedarf vorliegt, • die verwendete Methode und Algorithmen zur Berechnung von Vorhersagen, • Methoden zur kontinuierliche Verbesserung der eigenen Methoden und des Abgleichs mit mehreren Datenspeichern, z. B. Maschinen, • und die technische Umsetzung der Synchronisierung zwischen einem Lerner und eines Klassifikators. <p>Dabei konzentriert sich das Modul nicht auf das reine Condition Monitoring, sondern klassifiziert Zusammenhänge verschiedenster Maschinendaten.</p>
Generisch/optional	<p>Optional. Dieses Modul wird als optional angesehen, obwohl es generische Anforderungen besitzt. Dies liegt daran, dass eine Umsetzung dieses Moduls als sehr herausfordernd anzusehen ist und daher nicht zwangsläufig in jedem IIS enthalten sein muss.</p>
Funktionsbeschreibung	<p>Dieses Modul unterteilt sich in zwei Teile, die die Mechanismen des Predictive Maintenance kennzeichnen. Diese Teile werden nachfolgend in Form abstrakter Designmuster beschrieben:</p> <p>Der Klassifikator</p> <p>Als Klassifikator wird hier ein meist dezentral bei der Maschine beheimateter, leichtgewichtiger Entscheider bezeichnet. Dieser prüft bspw. die aktuellen Sensor-/Logdaten und bestimmt anhand dieser Logdaten, ohne große Rechenzeit, den Bedarf nach Instandhaltung. Der Instandhalter muss jede falsche Entscheidung des</p>

Klassifikators protokollieren. Der Klassifikator hängt von dem später beschriebenen Lerner dahingehend ab, dass der Lerner den Klassifikator erzeugt und diesen in gewissen Abständen aktualisiert. Eine Verbindung zum Lerner muss nur für die Aktualisierung oder Datenweitergabe erfolgen.

Der Klassifikator erzeugt Warnungen für bevorstehende Fehler und ist in der Lage, diese an weitere Komponenten zu versenden. Weiterhin wird eine Wahrscheinlichkeit für das Auftreten einer Störung errechnet, z. B. aus der Korrelierung von Sensordaten und dem Maschinenzustand. Aus vergangenen Daten können so Schlüsse für die Zukunft gezogen werden, die für einen Menschen in der Kürze zu komplex sind, um erfasst zu werden. Auf dieser Basis ist es möglich, Vorhersagen über den Verschleiß von Teilen und etwaige Fehler zu treffen.

Der Lerner

Als Lerner wird hier eine zentrale Intelligenz verstanden, die zum einen alle zur Bewertung von Instandhaltung nötigen Daten diverser Maschinen abgreifen und speichern kann. Zum anderen diese Daten nutzt, um die Berechnung von Klassifikatoren zu verbessern. Das Beschaffen der Daten sollte nicht in Echtzeit vollzogen werden, da dies zu einer hohen Belastung der Netzwerke führen könnte. Anhand all dieser Daten und eventuell einem markierten (Fehler ja/nein, Fehler x/y) Testset, können Berechnungen per unterschiedlichster Algorithmen durchgeführt werden, deren Ziel die Identifikation von Fehler- bzw. Wartungsursachen und -zusammenhänge darstellt. Aufbauend auf diesen Informationen werden verbesserte Klassifikatoren erstellt. Dies heißt in dem Fall, dass Prognosen für Fehler anhand dieser Daten errechnet und Prognosen in Form des Klassifikators heruntergebrochen werden. Ein Beispiel für den Lerner wäre ein typischer Machine Learning Algorithmus mit einem Lerner/Klassifikator, der mit einem Testset trainiert wird und am Ende einen Klassifikator für neue Anwendungsfälle erzeugt. Jeder Lauf des Lerners, der in einer signifikanten Veränderung des Lerners mündet, führt zu einem Update aller angeschlossener Klassifikatoren.

Die Zerteilung zwischen Klassifikator und Lerner wird aufgrund folgender Punkte als nötig angesehen:

- Datendurchsatz
- Rechenintensität
- Dezentrales Design
- Einfaches Aufspielen der Diagnoseintelligenz auf gleichartige Maschinen bzw. Anlagen
- Diagnoseintelligenz ebenfalls bei fehlendem Netzzugang
- Erhöhung der Effektivität, da Anlagen ggf. von mehreren Anlagen lernen können
- Statistische Validierung bzw. Absicherung von erlernten Klassifikatoren

Der Datendurchsatz spielt in vielen Produktionsnetzwerken eine große Rolle, da sich diese Netzwerke im Normalbetrieb schon nahe der Komplettauslastung bewegen. Aus diesem Grund sollten die benötigten Daten für den Lerner so übermittelt werden, dass möglichst wenige Auswirkungen auf das Netzwerk zu befürchten sind. Beispielsweise könnten Daten für den Lerner in Niedriglastzeiten übertragen werden. Wann der Lerner die Daten der Maschinen bekommt, spielt für dessen Einsatz keine Rolle. Er sollte jedoch in möglichst regelmäßigen Abständen aktualisiert werden.

Die Berechnung eines neuen Klassifikators ist in der Regel sehr rechen- und damit auch zeitintensiv. Sie kann nicht an der Maschine vorgenommen werden und muss daher zentral geschehen, so dass keine Produktionsressourcen der einzelnen Maschinen betroffen sind. Ein weiterer Grund hierfür ist, dass für die Berechnung des Klassifikators Daten aus mehreren Maschinen aggregiert werden können.

	<p>Im Sinne von CPS wäre dies ein Beispiel vom Wechselspiel zwischen dezentraler und zentraler Intelligenz mehrerer Systeme. Der Klassifikator an der Maschine entscheidet dezentral vor Ort ohne Verbindung zum zentralen System. Er wird jedoch, wenn nötig, von der zentralen Instanz aktualisiert. Die zentrale Instanz des Lernalers kann hierbei beliebig viele dezentrale Klassifikatoren aktualisieren, sofern diese sich mit dem identischen Problem beschäftigen. Eine beispielhafte Implementierung dieses Konzeptes ist bei (Fleischmann et al., 2016b) zu finden.</p> <p>Voraussetzungen für die Funktionsfähigkeit</p> <p>Der Klassifikator benötigt lesenden Zugriff auf die Maschine bzw. auf die aktuellen Logdateien oder muss direkt in die Maschine integriert werden. Die Auswertung sollte in Echtzeit oder mit kurzer Verzögerung erfolgen. Weiterhin muss der Klassifikator Fehlermeldungen absetzen können.</p> <p>Der Lernaler benötigt Zugang zu den wesentlichen Log-/Maschinendaten, um die Klassifikatoren zu erzeugen. Dies muss nicht in Echtzeit geschehen, auch könnte er die Daten zu einem bestimmten Zeitpunkt importieren.</p> <p>Insbesondere muss jedoch die Datenweitergabe zwischen Maschine und Lernaler bzw. Klassifikator spezifiziert werden. Darüber hinaus muss eine Methode erdacht werden, wie die Synchronisierung zwischen Lernaler und Klassifikator stattfindet.</p>
<p>Mögliche Unterfunktionen</p>	<ul style="list-style-type: none"> • Extraktion der Maschinendaten und Sensorwerte • Automatische Klassifizierung aus Korrelierung von Sensordaten und Maschinenzustand • Synchronisation zwischen Lernaler und Klassifikator durchführen • Klassifikator trainieren • Update aller verbundenen Klassifikatoren • Austausch der benutzten Algorithmen und Verfahren • Automatisches Senden von potentiellen Wartungs- und Fehlermeldungen
<p>Verbindungen zu anderen Modulen</p>	<p>Baustein B.4. Fehlermanagement:</p> <ul style="list-style-type: none"> • Modul M.16. Fehlererkennung & Condition Monitoring • Modul M.17. Fehlerdatenbank <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.4. Anlageninformationen • Modul M.5. Anlagenübersicht und -auswertung • Modul M.6. Berechnung/Zugriff auf Produktionsplanung • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.11. Wartungsmanagement • Modul M.37. Wetterdaten
<p>Verbundene Aktivitäten</p>	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A7 – Zustandsabfrage Wartung (Condition Monitoring)
<p>Aufteilung nach Entität</p>	<p>Das hier vorgestellte Modul befindet sich an der Maschine und im Backend. Dabei besteht eine Zweiteilung, welche im Text zuvor beschrieben worden ist.</p>

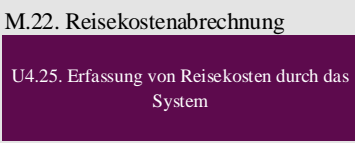
6.2.20 M.20. Synchronisation Hersteller und Betreiber

Komponente	M.20. Synchronisation Hersteller und Betreiber
Darstellung in Referenz-architektur	<p>M.20. Synchronisation Hersteller und Betreiber</p>
Zweck/Ziel	<p>Das Modul ermöglicht die Synchronisation von Betriebsdaten der Anlagen/Maschinen zwischen Hersteller/Serviceerbringer und Betreiber. Mit diesen Daten können anhand von Logdateien und Fehlerdaten zukünftig Instandhaltungen zielgerichteter, schneller und vor allem vorausschauender durchgeführt werden. Zusätzlich erweitert sich die Datenbasis beim Hersteller bzgl. der Verwendung und dem Verhalten der eigenen Maschinen. Anhand dieser Informationen können die eigenen Maschinen weiterentwickelt und verbessert werden.</p>
Generisch/optional	Optional
Funktionsbeschreibung	<p>Die Daten des Anlagenbetreibers in Form von Betriebsprotokollen mitsamt bestimmter Sensorwerte können direkt an den Hersteller gesendet werden. Dies kann entweder kontinuierlich geschehen, so dass der Hersteller bzw. der Serviceerbringer vorausschauend Wartung und Fehler erkennen kann. Oder die Daten werden gesammelt und erst nach Freigabe durch den Betreiber gesendet. Die Daten können auf Basis der Module M.5. Anlagenübersicht und -auswertung sowie M.4. Anlageninformationengesammelt an den Hersteller gesendet werden.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Senden von Betriebsprotokollen • Einrichten einer Freigabe für den Hersteller • Freigabe von Daten an den Hersteller • Einstellungen über die zu übertragenden Daten • Verbindung zu internen Datenhaltern
Verbindungen zu anderen Modulen	<p>B.5. Technischer Kundendienst:</p> <ul style="list-style-type: none"> • Modul M.23. Serviceprotokolle <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.4. Anlageninformationen • Modul M.5. Anlagenübersicht und -auswertung
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Maschine und mobiles Endgerät.

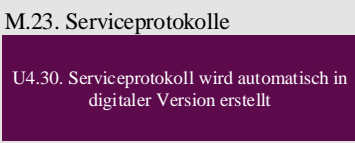
6.2.21 M.21. Navigation

Komponente	M.21. Navigation
Darstellung in Referenzarchitektur	
Zweck/Ziel	Das Modul stellt eine Navigation zur Verfügung, damit der Instandhalter per Routenplanung direkt zu dem Instandhaltungsort geleitet wird.
Generisch/optional	Optional
Funktionsbeschreibung	<p>Der Routenplaner stellt nach dem Eingeben der gewünschten Zielposition und eventuellen Zwischenzielen eine Route vom aktuellen Standort aus dar. Anhand von GPS-Ortsbestimmung wird der eigenen Standort bestimmt. Die aktuelle Position auf der Route wird dann automatisch verfolgt und eine stetig aktualisierte Zeitangabe über das Eintreffen angezeigt. Diese Funktion wird typischerweise auf dem mobilen Endgerät bereits durch bestehende Lösungen seitens Google, Apple, HERE oder ähnlichen Anbietern gelöst. Eine mögliche Integration der Routenplanung bei der Auftragsplanung erfordert die Auslesbarkeit dieser Lösungen und die berechneten Fahrtwege und -zeiten zu bestimmten Tageszeiten.</p> <p>Voraussetzungen für die Funktionsfähigkeit</p> <p>Der Anwender benötigt ein mobiles Endgerät, welches über eine ausreichende Hardware/Software verfügen sollte, wodurch vielfältige Funktionen unterstützt und auf dem ausgewählte Apps installiert werden können.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Import und Update von Kartenmaterial • Routensimulation • Navigation • Berechnung von Routen
Verbindungen zu anderen Modulen	<p>B.5. Technischer Kundendienst:</p> <ul style="list-style-type: none"> • Modul M.22. Reisekostenabrechnung <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	mobiles Endgerät.

6.2.22 M.22. Reisekostenabrechnung

Komponente	M.22. Reisekostenabrechnung
Darstellung in Referenzarchitektur	
Zweck/Ziel	Das Modul dient zur Reisekostenabrechnung des Instandhalters. Diese können teilautomatisch berechnet werden und die Kosten können direkt den Instandhaltungsaufträgen und Kunden zugerechnet werden. Dies beschleunigt den Prozess der Reisekostenabrechnung und damit verbunden die Abrechnung mit dem Servicenehmer der Instandhaltung.
Generisch/optional	Optional
Funktionsbeschreibung	Arbeitszeiten, Fahrtwege und Übernachtungskosten sowie Belege können per Maske in dem Modul eingetragen und mit den zugehörigen Tätigkeiten verbunden werden. Bestimmte Informationen zur Reisekostenberechnungen können in Verbindung mit den Modulen M.9. Auftragsverwaltung/Priorisierung, M.24. Vertragsmanagement, M.21. Navigation und M.15. Arbeitszeiterfassung automatisch oder teilautomatisch hinzugefügt werden. Dies wäre bspw. der zugehörige Auftrag, die Arbeitszeit, die zurückgelegte Strecke und bestimmte Vertragsdaten des Kunden. Anhand dieser Daten berechnet das Modul automatisch die angefallenen Reisekosten. Diese können direkt an andere Module weitergereicht und so Aufträgen zugeordnet werden.
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Eingabemaske für Reisekosten • Berechnung der Reisekosten • Zuordnung Reisekosten zu durchgeführten Aufträgen • Freigabe von Reisekosten
Verbindungen zu anderen Modulen	<p>B.5. Technischer Kundendienst:</p> <ul style="list-style-type: none"> • Modul M.21. Navigation • Modul M.24. Vertragsmanagement • Modul M.25. Signierfunktion <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.15. Arbeitszeiterfassung
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Backend und mobiles Endgerät.

6.2.23 M.23. Serviceprotokolle

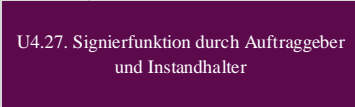
Komponente	M.23. Serviceprotokolle
Darstellung in Referenzarchitektur	 <p>M.23. Serviceprotokolle</p> <p>U4.30. Serviceprotokoll wird automatisch in digitaler Version erstellt</p>
Zweck/Ziel	Das Modul bietet die Möglichkeit, Serviceprotokolle nach der Beendigung einer Instandhaltungstätigkeit zu erstellen und dem Kunden zur Verfügung zu stellen. Viele Informationen in diesem Serviceprotokoll können automatisiert ausgefüllt werden.
Generisch/optional	Optional
Funktionsbeschreibung	<p>Nach Beendigung einer Instandhaltungstätigkeit kann per Formular ein Serviceprotokoll vom Instandhalter erstellt werden. Falls bereits Daten für dieses Formular wie Kundename, Arbeitszeit oder durchgeführte Tätigkeit in anderen Modulen vorliegen, können diese automatisch integriert werden. Die benötigten Daten hängen von unternehmensindividuellen Anforderungen ab. Beispiele wären:</p> <ul style="list-style-type: none"> • Geleistete Arbeitszeit • Durchgeführte Tätigkeiten • Auftragsnamen und Nummer • Beteiligte Personen • Verbrauchte Ersatzteile <p>Anschließend kann der Instandhalter noch nicht vollständige Informationen per Maske ein- und freigeben.</p> <p>Ist das Formular ausgefüllt, kann es vom Auftraggeber mithilfe der Module M.25. Signierfunktion und M.20. Synchronisation Hersteller und Betreiber abgezeichnet werden. Anschließend wird es als Protokoll an den Auftraggeber versendet.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Bereitstellen von Eingabemaske Serviceprotokoll • Automatisches Befüllen von Daten aus anderen Modulen • Versenden an Auftraggeber/Kunden
Verbindungen zu anderen Modulen	<p>Baustein B.5. Technischer Kundendienst:</p> <ul style="list-style-type: none"> • Modul M.15. Arbeitszeiterfassung • Modul M.20. Synchronisation Hersteller und Betreiber • Modul M.24. Vertragsmanagement • Modul M.25. Signierfunktion <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.1. Handlungsleitfäden/Checklisten/Prüflisten • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.28. Ersatzteilmanagement
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.

Aufteilung nach Entität	Backend und mobiles Endgerät.
--------------------------------	-------------------------------

6.2.24 M.24. Vertragsmanagement

Komponente	M.24. Vertragsmanagement
Darstellung in Referenzarchitektur	
Zweck/Ziel	Das Modul dient der Darstellung der Vertragsdetails und Beziehung zum Servicenehmer. Es dient der Informationsgewinnung des Instandhalters, so dass dieser abhängig von den Vertragsdetails Instandhaltungsmaßnahmen vornehmen kann und die Grundlagen zur Abrechnung der eigenen Leistungen auf einen Blick hat.
Generisch/optional	Optional
Funktionsbeschreibung	Die Vertragsdetails müssen in das System abhängig vom Kunden hinterlegt werden können. Diese müssen daher eindeutig dem Kunden zugeordnet werden können. Diese Daten können ggfs. aus einem bestehenden CRM-System geladen oder können in Form eines Formulars editiert und eingesehen werden. Alternativ können auch PDFs mit den aktuellen Daten hochgeladen und eingesehen werden.
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Management der Vertragsdokumente der Kunden • Eintragen und Bearbeiten von Kundendaten • Darstellung von Vertragsdaten
Verbindungen zu anderen Modulen	<p>Baustein B.5. Technischer Kundendienst:</p> <ul style="list-style-type: none"> • Modul M.22. Reisekostenabrechnung • Modul M.23. Serviceprotokolle <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Backend und mobiles Endgerät.

6.2.25 M.25. Signierfunktion

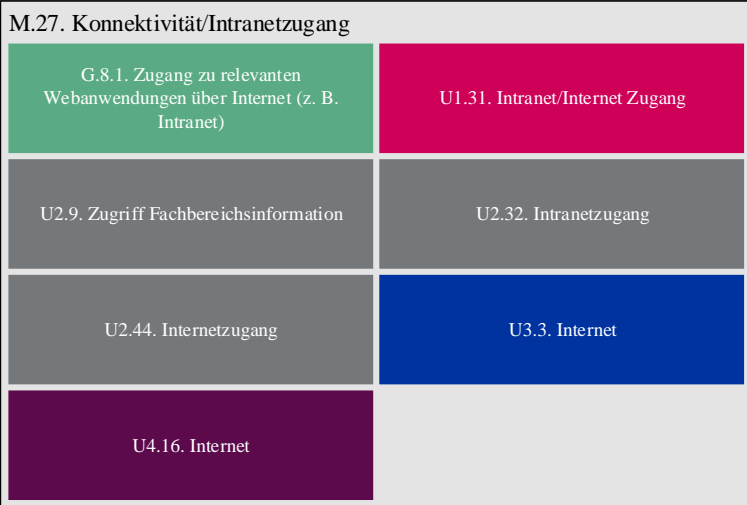
Komponente	M.25. Signierfunktion
Darstellung in Referenzarchitektur	<p>M.25. Signierfunktion</p> 
Zweck/Ziel	Das Modul stellt eine (rechtsverbindliche) Signierfunktion bereit. Mit dieser können Aufträge von Kunden und Instandhaltern per Unterschrift verbindlich quittiert werden.
Generisch/optional	Optional
Funktionsbeschreibung	<p>Das Modul stellt die Möglichkeit zur Verfügung, eine Unterschrift einer Person aufzuzeichnen und abzuspeichern. Einem Auftrag, Serviceprotokoll oder sonstigen Aktionen kann diese Unterschrift zugeordnet werden.</p> <p>Dies wird damit erreicht, dass die Signierfunktion aus anderen Modulen aufgerufen werden kann, indem ein Name und die zugehörige Aktion übergeben werden. Zu diesem Aufruf wird die Signatur per Toucheingabe oder per Stift gespeichert. Eine Umwandlung der Eingabe in ein digitales Format wird vom mobilen Endgerät übernommen.</p> <p>Anschließend wird die Signatur gespeichert und an das aufrufende Modul weitergeleitet.</p> <p>Voraussetzungen für die Funktionsfähigkeit</p> <p>Voraussetzung zur Funktionstüchtigkeit ist die Eingabemöglichkeit per Touch oder Stift. Das Endgerät muss auf diese Art von Eingaben reagieren und in rechtsverbindliche Genauigkeit speichern können.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Aufzeichnen einer Signatur • Sicheres Ablegen der Signatur in Verbindung mit einem (Kunden-)Auftrag
Verbindungen zu anderen Modulen	<p>Baustein B.5. Technischer Kundendienst:</p> <ul style="list-style-type: none"> • Modul M.22. Reisekostenabrechnung • Modul M.23. Serviceprotokolle <p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.13. Schichtbuchfunktionalität/Synchronisation
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Backend und mobiles Endgerät.

6.2.26 M.26. Kommunikation

Komponente	M.26. Kommunikation												
Darstellung in Referenz-architektur	<table border="1"> <tr> <td colspan="3" data-bbox="478 304 1380 333">M.26. Kommunikation</td> </tr> <tr> <td data-bbox="478 333 778 427">G.3.1. Kommunikation (Messenger, Telefon, Mail, Kontakte)</td> <td data-bbox="778 333 1078 427">U1.1. E-Mail</td> <td data-bbox="1078 333 1380 427">U2.33. WhatsApp/Bild versenden/Foto</td> </tr> <tr> <td data-bbox="478 427 778 521">U2.43. Outlook Zugang</td> <td data-bbox="778 427 1078 521">U3.2. Mail</td> <td data-bbox="1078 427 1380 521">U4.8. Unterstützung bei Kommunikation mit Kollegen und Kunden</td> </tr> <tr> <td data-bbox="478 521 778 616">U4.11. Telefonbücher</td> <td data-bbox="778 521 1078 616">U4.14. Kommunikationsmöglichkeiten</td> <td data-bbox="1078 521 1380 616">U4.15. Mailfunktion</td> </tr> </table>	M.26. Kommunikation			G.3.1. Kommunikation (Messenger, Telefon, Mail, Kontakte)	U1.1. E-Mail	U2.33. WhatsApp/Bild versenden/Foto	U2.43. Outlook Zugang	U3.2. Mail	U4.8. Unterstützung bei Kommunikation mit Kollegen und Kunden	U4.11. Telefonbücher	U4.14. Kommunikationsmöglichkeiten	U4.15. Mailfunktion
M.26. Kommunikation													
G.3.1. Kommunikation (Messenger, Telefon, Mail, Kontakte)	U1.1. E-Mail	U2.33. WhatsApp/Bild versenden/Foto											
U2.43. Outlook Zugang	U3.2. Mail	U4.8. Unterstützung bei Kommunikation mit Kollegen und Kunden											
U4.11. Telefonbücher	U4.14. Kommunikationsmöglichkeiten	U4.15. Mailfunktion											
Zweck/Ziel	<p>Die Instandhalter haben es mit Maschinen und Anlagen höchster Komplexität und steigender Spezialisierung zu tun. Dies führt dazu, dass auftretende Fehler eine große Komplexitäts- und Kombinationsvielfalt besitzen und oft nicht ausreichend von einem einzelnen Mitarbeiter diagnostiziert und behoben werden können. Eine Kommunikation mit zuständigen Experten des Unternehmens oder des Maschinenherstellers unterstützt den Instandhalter. Die nötigen Informationen können so mitgeteilt und zusammen mit eventuell zusätzlichen Fernwartungsoptionen eingesetzt werden.</p> <p>Daher umfasst das Modul Kommunikation Möglichkeiten der Informationsweitergabe und -empfangs sowie typische Funktionen, um diese zu unterstützen. Diese umfassen Messenger-, Telefonie- und Mailfunktionalitäten verbunden mit den jeweiligen Kontakten.</p>												
Generisch/optional	Generisch												
Funktionsbeschreibung	<p>Die angesprochenen Funktionen bilden die Basis eines IIS, um mit anderen Akteuren der Instandhaltung in Kontakt zu treten und einen Informationsaustausch durchzuführen.</p> <p>Die Funktionalitäten umspannen Messaging, Telefonie und Mail. Um diese Funktionalität effizient umzusetzen, bedarf es auswählbare und pflegbare Kontakte. Die meisten der Funktionen werden normalerweise durch Systeme außerhalb des IIS abgedeckt, sei es durch ein E-Mail-System, Apps auf dem mobilen Endgerät oder der dort vorhandenen Telefoniefunktion bzw. unternehmensweite VoIP-Lösungen. Diese bestehenden Lösungen gilt es so in das IIS zu integrieren, dass eine medienbruchlose Kommunikation stattfinden kann. Damit die Informationen bzw. die Kommunikation zu einem bestimmten Auftrag direkt im IIS angezeigt oder diese direkt von diesem aus ausgerufen werden können, müssen die Funktionen dort integriert werden. Eine Möglichkeit der Protokollierung der Kommunikation und eine Zuordnung zu bestimmten Aufträgen oder Fehlern ist zu prüfen.</p> <p>Die technologische Umsetzung hängt dabei stark von den zur Verfügung stehenden unternehmensinternen Lösungen ab (E-Mail, VoIP).</p> <p>Ausnahme bildet hier die Messagingfunktion. Diese stellt eine typische Funktionalität dar, die bei Apps wie WhatsApp oder Facebook Messenger zu finden sind. Diese Apps können meist nicht in die eigene Lösung integriert werden. Daher muss eine ähnliche Lösung implementiert werden, die direkt aus der Auftragsdurchführung aufgerufen werden kann und diese Kommunikation in den Auftrag integriert.</p> <p>Ein weiterer wichtiger Funktionspunkt stellt das Kontaktmanagement dar. Bestehende unternehmensinterne Kontakte der zuständigen Personen (z. B. weitere Instandhalter oder externe Ansprechpartner) sollen in das IIS integriert und direkt</p>												

	<p>auswählbar gemacht werden. Diese können dann ausgewählt werden, um ihnen Nachrichten, Anrufe oder E-Mails zu senden.</p> <p>Voraussetzungen für die Funktionsfähigkeit</p> <p>Sofern die Funktionen von bestehenden Applikationen ohne Medienbrüche direkt in das IIS integriert oder implementiert werden sollen, müssen die Schnittstellen zu den bestehenden Funktionalitäten offen sein und integriert werden können.</p> <p>Falls die Funktionen nicht im IIS direkt implementiert werden und der Anwender diese auf seinem mobilen Endgerät nutzen will, so muss dieses über eine ausreichende Hardware/Software verfügen, das die vielfältigen Funktionen unterstützt und auf dem ausgewählte Apps installiert werden können.</p> <p>Dazu zählen der Mail-Client, eine Telefonie-Funktion und die benötigten Kontaktmanagementfunktionen. Zusätzlich wäre ein Messenger nötig, mit dem die aufgenommenen Daten mit anderen Akteuren geteilt werden können.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Import und Export von Kontaktdaten, z. B. Telefonbucheinträge und Mailadressen in Verbindung mit evtl. bestehenden Lösungen wie Outlook o. Ä. • Kontaktdatenmanagement • Nachrichtenaustausch • Auswahl Kommunikationsmöglichkeiten in Verbindung mit Tätigkeit • Telefonie • Senden von Nachrichten mit Anhang • Einbindung Outlook & diverse Apps • Archivierung von Nachrichten • Suchfunktion für Nachrichten
Verbindungen zu anderen Modulen	<p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.1. Handlungsleitfäden/Checklisten/Prüflisten • Modul M.5. Anlagenübersicht und -auswertung • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.10. Mitarbeitermanagement • Modul M.30. Remote Zugriff/Fernwartung auf Anlagen/Maschinen
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Mobilen Endgerät und Backend.

6.2.27 M.27. Konnektivität/Intranetzugang

Komponente	M.27. Konnektivität/Intranetzugang
Darstellung in Referenzarchitektur	 <p>The diagram illustrates the structure of M.27. Konnektivität/Intranetzugang, which is composed of several sub-modules:</p> <ul style="list-style-type: none"> G.8.1. Zugang zu relevanten Webanwendungen über Internet (z. B. Intranet) - Green box U1.31. Intranet/Internet Zugang - Pink box U2.9. Zugriff Fachbereichsinformation - Grey box U2.32. Intranetzugang - Grey box U2.44. Internetzugang - Grey box U3.3. Internet - Blue box U4.16. Internet - Purple box
Zweck/Ziel	<p>Das Modul "Internetzugang/Intranetzugang" ermöglicht die Konnektivität, sowohl für das mobile Endgerät als auch für das Backend-System und damit die Nutzung von externen Diensten außerhalb des IIS, bspw. interne Webanwendungen. Damit bildet dieses Modul die Grundlage für das zuvor beschriebene Modul M.26. Kommunikation und weitere Module, die einen Zugriff über das Internet benötigen. Mit diesem Modul können gesicherte oder je nach Anwendungsfall auch ungesicherte Verbindungen aufgebaut werden. Insbesondere für den Zugriff auf das Intranet von außen müssen entsprechende Sicherheitsvorschriften der jeweiligen Unternehmen bezüglich Authentifizierung und Autorisierung berücksichtigt werden.</p>
Generisch/optional	<p>Generisch</p>
Funktionsbeschreibung	<p>Der Benutzer will bspw. über eine Anwendung zusätzliche Informationen nachladen, die nicht lokal auf dem Endgerät vorliegen. Zu diesem Zweck muss eine Verbindung in das Internet oder Intranet aufgebaut werden. Diese kann entweder über das Mobilfunknetz (GPRS, UMTS, LTE) oder über ein Unternehmensnetzwerk (LAN, WLAN) erfolgen. Je nach Sicherheitsanforderungen (Virtuelles Privates Netzwerk, Zertifikate, Zugangsdaten, etc.) müssen noch weitere Dienste genutzt oder Einstellungen vorgenommen werden. Im Allgemeinen ist die Funktion des Moduls die Bereitstellung einer Internet-/Intranetverbindung, über die höherwertige Anwendungen genutzt werden können, um Daten abzurufen, die nicht lokal vorliegen.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Herstellung von stabiler Internetverbindung • Verbindung zu Intranet (auch extern) • Zugang zu Fachbereichsinformationen
Verbindungen zu anderen Modulen	<p>Dieses Modul muss zu fast jedem Modul Verbindungen besitzen, dies ist jedoch davon abhängig, wann und wo das Modul benötigt bzw. eingesetzt werden muss. Es ist daher als übergeordnetes Modul zu verstehen, sodass Verbindungen nicht explizit genannt werden.</p>
Verbundene Aktivitäten	<p>Keine Aktivitäten spezifiziert.</p>

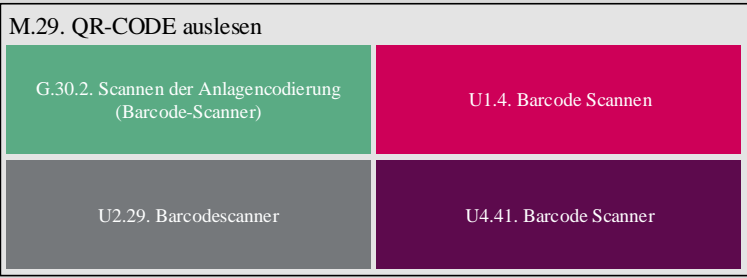
Aufteilung nach Entität	Möglichst jeder Entität sollte Konnektivität besitzen, solange dies nicht gegen die Sicherheitsregularien verstößt. Die Konnektivität ist hier im Speziellen auf das mobile Endgerät gerichtet.
--------------------------------	---

6.2.28 M.28. Ersatzteilmanagement

Komponente	M.28. Ersatzteilmanagement
Darstellung in Referenzarchitektur	
Zweck/Ziel	Das Modul dient dem Ersatzteilemanagement in Verbindung mit Instandhaltungstätigkeiten. Hierdurch wird erreicht, dass die nötigen Ersatzteile, deren Verfügbarkeit und ggfs. die Bestandssituationen bei Zulieferern und Herstellern, mit Instandhaltungstätigkeiten in Verbindung gesetzt werden können.
Generisch/optional	Generisch
Funktionsbeschreibung	<p>Das Modul bietet die folgende Funktionalität:</p> <ul style="list-style-type: none"> • Ersatzteile für Aufträge zusammenstellen • Bestandssituation prüfen • Ersatzteile nachbestellen (optional) <p>Bevor ein Instandhalter seine Instandhaltungstätigkeiten beginnt, müssen die Ersatzteile für seinen Auftrag zusammengestellt werden. Darüber hinaus muss der Bestand der Ersatzteile geprüft werden. Dies geschieht je nach Anwendungsfall bei den zuständigen internen Stellen oder durch eine Verbindung direkt beim Hersteller. Je nach Verfügbarkeit, Ort und Lieferzeit ist den späteren Aufträgen in Verbindung mit dem Modul M.9. Auftragsverwaltung/Priorisierung eine Priorität und ein passender Termin zuzuordnen. Bei einem Auftrag verwendete Ersatzteile müssen direkt nach Abschluss des Auftrages bzw. vor Beginn des Auftrages bestellt bzw. nachbestellt werden. Sind benötigte Teile nicht verfügbar, so müssen die betroffenen Aufträge zurückgestellt werden.</p> <p>Eine weitere Funktion ist, dass ein Instandhalter nach Ersatzteilen suchen und deren Bestand prüfen kann.</p> <p>Voraussetzungen zur Funktionsfähigkeit</p> <p>Der Zugriff auf Datenbanken, in denen die Verfügbarkeit bzw. der Bestand von Ersatzteilen hinterlegt wird, muss möglich sein. Dies kann entweder intern oder extern erfolgen. Darüber hinaus muss es für jeden Auftrag möglich sein die dafür benötigten Ersatzteile zu bestimmen. Dieser Zusammenhang muss händisch hergestellt oder später erweitert werden.</p>

Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Bestandssituation von (benötigten) Ersatzteilen anzeigen • Bestandssituation von (benötigten) Ersatzteilen bei Zulieferer/Hersteller/Serviceanbieter anzeigen • (Automatische) Rückmeldung bei fehlenden Ersatzteilen für bestimmte Arbeitsaufträge bzw. generelle Lagerengpässe • (Automatisches) Zusammenstellen des für den Instandhaltungsauftrag benötigten Ersatzteile • Bestellung von Ersatzteilen • Buchung von Ersatzteilen nach Verbrauch • Verbindung zu Herstellerersatzteilsystem
Verbindungen zu anderen Modulen	<p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.1. Handlungsleitfäden/Checklisten/Prüflisten • Modul M.9. Auftragsverwaltung/Priorisierung – Dieses Modul ruft das vorliegende Modul auf • Modul M.23. Serviceprotokolle
Verbundene Aktivitäten	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A2 – Inspektion Wartung Instandhalter • Aktivität A3 – Instandsetzung • Aktivität A4 – Vereisung der Windenergieanlage (WEA) • Aktivität A5 – Ereignisgesteuerte Instandsetzung • Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung • Aktivität A7 – Zustandsabfrage Wartung (Condition Monitoring)
Aufteilung nach Entität	Dieses Modul befindet sich im Backend wie auch auf dem mobilen Endgerät.

6.2.29 M.29. QR-CODE auslesen

Komponente	M.29. QR-CODE auslesen
Darstellung in Referenzarchitektur	
Zweck/Ziel	Das Modul „QR-CODE auslesen“ dient des vereinfachten, manuellen Abrufs der aktuellen Maschinendaten inklusive des Maschinenstatus sowie der Anzeige der abzuarbeitenden Tätigkeiten an dieser Maschine durch ein mobiles Endgerät. Das Modul implementiert die Fähigkeit QR-Codes zu scannen, deren Daten auszulesen und dadurch Informationen über die Maschine lokal, wie auch aus dem Backend zu erlangen. Die zu übertragenden Daten können z. B. die Wartungs- und Betriebsdaten,

	<p>Fehlerlisten und Handlungsleitfäden umfassen, die aus anderen Modulen geholt bzw. auf diese verwiesen werden muss. Diese Informationen stehen dem Abrufen, in aller Regel Instandhalter oder Bediener, nach dem Einlesen des Codes und der Auswahl der möglichen Daten zeitnah zur Verfügung. Dies ermöglicht eine schnelle und intuitive Übersicht bzgl. Instandhaltungstätigkeiten bzw. Maßnahmen.</p> <p>Durch die kürzere Erkennungs- und Informationszeit während des normalen Betriebs erhöht sich die Effizienz der Mitarbeiter, da benötigte Informationen sofort digital zur Verfügung stehen und nicht mehr händisch langwierig von verschiedenen Stellen bzw. Archiven beschafft werden müssen. Zusätzlich verringern sich die Fehlentscheidungen aufgrund unvollständiger oder ungenauer, oft nur mündlicher Daten.</p>
Generisch/optional	Generisch, außer es besteht keine Nutzung von einem mobilem Endgerät.
Funktionsbeschreibung	<p>Der Anwender liest mit dem mobilen Endgerät den QR-Code des jeweiligen Gegenstandes der Instandhaltung ein. Der Code wird analysiert und in einen systemkonformen, eindeutigen Identifikator der Maschine oder des betrachteten Gegenstandes umgewandelt. Dies könnte bspw. ein Integer sein, der in einer Datenbank mit den Informationen der Anlage verbunden ist. Daraufhin nimmt das mobile Endgerät über eine geeignete und verfügbare Verbindungsart, bspw. WLAN, Kontakt mit dem Backend auf und lädt die aktuellen und aufbereiteten Informationen auf das mobile Endgerät. Dieses vermittelt ihm die gewünschten, übersichtlich dargestellten Maschinendaten und den Status der Anlage. Diese Informationen werden nur durch dieses Modul aufgerufen. Die Informationen selbst werden durch andere Module bereitgestellt.</p> <p>Die entscheidende Funktion des Moduls stellt das grafische Auslesen des QR-Codes per Kamera des mobilen Endgeräts dar. Eine Datenbank bzw. ein Vorgehen zum Erstellen des QR-Codes, das zu einem Mapping QR-Code zu Anlage führt, muss implementiert werden.</p> <p>Voraussetzungen zur Funktionsfähigkeit</p> <p>Der Anwender benötigt ein mobiles Endgerät mit Kamera bzw. Videofunktion mit Möglichkeit zur QR-Code Erkennung. Jede Anlage muss zudem mit einem QR-Code versehen werden, der mit den Daten im Backend verknüpft und eindeutig zuordenbar sein muss. Eine in Bandbreite ausreichende Verbindungsart muss zur Verfügung stehen, um die eventuell umfangreichen Daten vom Backend auf das mobile Endgerät senden zu können.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Scannen von QR-Code • Herstellen der Zuordnung von QR-Code zu Anlage bzw. Bauteil • Aufruf von Informationen (auf die der Code verlinkt)
Verbindungen zu anderen Modulen	<p>Nachdem der QR-Code eingelesen worden ist, müssen die gewünschten Informationen zur gescannten Anlage aus folgenden Modulen im Backend geholt bzw. diese Module mitsamt dem Parameter der Anlage und der gewünschten Darstellung aufgerufen werden:</p> <ul style="list-style-type: none"> • Modul M.1. Handlungsleitfäden/Checklisten/Prüflisten • Modul M.4. Anlageninformationen • Modul M.5. Anlagenübersicht und -auswertung • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.11. Wartungsmanagement • Modul M.16. Fehlererkennung & Condition Monitoring

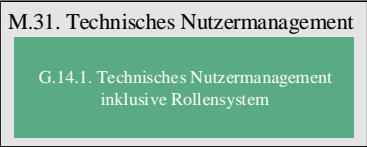
	<ul style="list-style-type: none"> • Modul M.30. Remote Zugriff/Fernwartung auf Anlagen/Maschinen
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Dieses Modul befindet sich auf dem mobilen Endgerät.

6.2.30 M.30. Remote Zugriff/Fernwartung auf Anlagen/Maschinen

Komponente	M.30. Remote Zugriff/Fernwartung auf Anlagen/Maschinen
Darstellung in Referenzarchitektur	
Zweck/Ziel	<p>Dieses Modul ermöglicht die Nutzung eines Remote Zugriffs, um auf Funktionen und Informationen der einzelnen Anlagen zuzugreifen und diese gegebenenfalls zu bearbeiten. Auf diese Weise können Benutzer Daten von bestimmten Anlagen oder Maschinen abrufen, ohne direkt vor Ort sein zu müssen, wodurch eine Fernwartung oder Instandhaltung erreicht werden kann. Zu diesen Daten gehören u. a. Fehlerbeschreibungen und Statusmeldungen. Neben dem Abruf von Informationen kann über dieses Modul auch eine Steuerung und ggf. eine Programmierung der Anlage initiiert werden.</p>
Generisch/optional	Generisch, außer ein Remotezugriff soll aus Sicherheitsgründen nicht möglich sein.
Funktionsbeschreibung	<p>Erfolgt ein entfernter Aufruf durch das IIS auf eine entsprechende Anlage, so prüft dieses Modul zunächst, ob der Benutzer autorisiert ist, die gewünschte Funktion auszuführen. Die hierfür notwendige Authentifizierung kann mit Unterstützung des Moduls M.10. Mitarbeitermanagement realisiert werden. Ein elementarer Punkt bei der Implementierung der Funktionalität ist vorab zu klären ob nur lesender oder auch schreibender Zugriff auf den Anlagen möglich sein soll. Des Weiteren ist zu klären, für welche Gruppe von Maschinen ein Remote Zugriff sinnvoll erscheint.</p> <p>Je nach Rolle des Benutzers und abhängig von den Sicherheitseinstellungen und unternehmensspezifischen Anforderungen können verschiedene Einschränkungen bzgl. des Abrufs von Informationen oder auch der Kontrolle von Anlagen berücksichtigt werden.</p>

	<p>Weiterhin gilt, dass mögliche manuelle Sicherheitsmechanismen an der Anlage installiert werden, sodass sichergestellt werden kann, dass nicht unbefugt auf die Anlagen zugegriffen werden kann.</p> <p>Im Zusammenhang mit der Fernwartung soll es Benutzern auch möglich sein einen Benutzer vor Ort, bspw. bei einer Instandhaltung, zu unterstützen.</p> <p>Voraussetzungen zur Funktionsfähigkeit</p> <p>Zur Nutzung des Remote Zugriffs muss eine Verbindung von außerhalb zur Maschine aufgebaut werden. Das IIS muss dafür eine geeignete Schnittstelle nach außen zur Verfügung stellen und die unternehmenseigene IT muss mit ihren Sicherheitseinstellungen und -regularien dafür sorgen, dass diese Schnittstellen benutzbar sind. Denkbar wäre, dass diese nur für bestimmte Zeiten offen sein könnten.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Aufbau einer Remote-Verbindung • Authentifizierung des Users • Freigabe der Maschine/Anlage durch Anlagenbesitzer/-betreiber • Remote Zugriff auf bestimmte Maschinenfunktionalitäten (lesend) • Remote Zugriff auf bestimmte Maschinenfunktionalitäten (schreibend) • Manuelle Freigabe an der Maschine, die zwingend aktiviert werden muss für den Remotezugriff • Remote Zugriff abhängig von den benötigten Use Cases (SPS, Bedienelemente, Prozessgeräte etc.)
Verbindungen zu anderen Modulen	<p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.10. Mitarbeitermanagement • Modul M.26. Kommunikation • Modul M.29. QR-CODE auslesen
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Dieses Modul befindet sich auf dem mobilen Endgerät, der Maschine und im Backend.

6.2.31 M.31. Technisches Nutzermanagement

Komponente	M.31. Technisches Nutzermanagement
Darstellung in Referenzarchitektur	
Zweck/Ziel	<p>Dieses Modul ist zuständig für die zentrale technische Verwaltung, der mit dem System interagierenden Benutzer. Dies schließt zum einen die Aufnahme von Nutzerdaten ein, wodurch sich die Personen eindeutig im technischen wie im Unternehmenskontext identifizieren lassen. Zum anderen ist auch die komplette feingranulare Administrierung von Verantwortlichkeiten, Rollen und Gruppen mit</p>

	den zugehörigen Berechtigungen vorgesehen, die essentiell für Kontrolle und Steuerung eines Informationssystems für die Instandhaltung ist. Zudem erlaubt dieses Modul die Anbindung/Nutzung von bestehenden Benutzerverwaltungssystemen. Diese Eingliederung der Nutzerdaten verringert die Komplexität und hilft das Informationssystem in den Kontext der unternehmenseigenen IT einzugliedern.
Generisch/optional	Generisch
Funktionsbeschreibung	<p>Administratoren und Anwender haben über eine grafische Benutzeroberfläche u. a. folgende Aktionen zur Auswahl:</p> <ul style="list-style-type: none"> • Benutzer anzulegen, deren Daten zu bearbeiten und diese wieder zu löschen • Anfertigen von Rollen und Gruppen • Erstellen von Berechtigungskonzepten, Zugriffskontrolllisten und Kennwortregeln <p>Die Verwaltung der Benutzer, insbesondere das massenweise Ändern von Daten, geschieht hierbei durch entsprechende Auswahlmöglichkeiten und nimmt somit nur sehr wenig Zeit in Anspruch. Die Verknüpfung bzw. Einteilung des Personals zu den passenden Rollen und Gruppen ist entweder durch „Drag and Drop“ oder durch das manuelle Markieren und Einfügen in den entsprechenden Teil der grafischen Maske möglich. Berechtigungskonzepte und Zugriffskontrolllisten können manuell implementiert werden.</p> <p>Darauf aufbauend müssen die vorhandenen Rollen und Gruppen im System eingegeben werden können bzw. es muss möglich sein diese zu erstellen. Weiterhin wird ein Regelsystem bzw. ein Berechtigungskonzept benötigt, das festlegt, welche Personen bestimmte Aktionen, hier insbesondere Schreib/Lese-Zugriff, ausführen oder Zugang zu Systemen bzw. Applikationen, z. B. Internetanwendungen, bekommen können.</p> <p>Außerdem muss es dem administrativen Personal möglich sein, diese Stammsätze und Berechtigungsprofile einerseits zu verbinden und andererseits mit geeigneten Funktionen zur Benutzer- und Rollenpflege zu bearbeiten. Die ganze Anwendung sollte ergonomischen Grundsätzen genügen, möglichst einfach, selbsterklärend und grafisch zu bedienen sein sowie eine barrierefreie Gestaltung innehaben.</p> <p>Das Modul M.10. Mitarbeitermanagement steht in enger Verbindung zu dem vorliegenden Modul, wobei in diesem Modul nur technische Fragen bzw. Berechtigungsfragen geklärt werden. Organisatorische Daten wie Termine, Positionen, auftragsspezifische Funktionen oder weitere Informationen, welche die Instandhaltung betreffen, sind dort zu finden.</p> <p>Voraussetzungen zur Funktionsfähigkeit</p> <p>Dieses Modul kann bzw. muss zu fast jedem Modul Verbindungen besitzen, dies ist jedoch abhängig davon wann und wo das Modul benötigt bzw. eingesetzt wird. Es ist daher als übergeordnetes Modul zu verstehen, sodass Verbindungen nicht explizit genannt werden.</p> <p>Es müssen die Daten aller Nutzer, die Zugriff auf das System benötigen, entweder schon vorhanden sein, z. B. in einem bereits existierenden und über bspw. LDAP (Lightweight Directory Access Protocol) angebundenes Verzeichnis, oder digital erfasst und in ein Nutzerverzeichnis eingetragen werden.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Benutzerdaten anlegen, löschen, verändern • Anlegen von Rollen und Gruppen • Erstellung von Berechtigungskonzepten und Kennwortregeln • Integration mit vorhandenem Benutzerverwaltungssystem

Verbindungen zu anderen Modulen	Dieses Modul kann bzw. muss zu fast jedem Modul Verbindungen besitzen, dies ist jedoch abhängig davon wann und wo das Modul benötigt bzw. eingesetzt wird. Es ist daher als übergeordnetes Modul zu verstehen, sodass Verbindungen nicht explizit genannt werden.
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Dieses Modul befindet sich im Backend und auf dem mobilen Endgerät.

6.2.32 M.32. Verarbeitung audiovisueller Medien

Komponente	M.32. Verarbeitung audiovisueller Medien
Darstellung in Referenzarchitektur	
Zweck/Ziel	Das Modul dient dazu, dass der Instandhalter audiovisuelle Medien erstellen, bearbeiten und konsumieren kann. Diese bilden eine Grundlage zur Dokumentation und dem Austausch von Informationen. Das Modul kann innerhalb anderer Module aufgerufen werden oder für sich separat stehen. Die erstellten Dateien stehen dem Instandhalter sofort zur Verfügung.
Generisch/optional	Generisch
Funktionsbeschreibung	<p>Bilder und Videos werden mit der im mobilen Endgerät eingebauten Kamera und Applikationen aufgenommen und auf diesem gespeichert. Dabei werden die nativen Applikationen des mobilen Endgeräts benutzt.</p> <p>Die Dateien auf dem mobilen Endgerät können durch Verweise in den anderen Modulen eingefügt oder direkt die Applikationen aufgerufen werden.</p> <p>Werden die audiovisuellen Medien innerhalb anderer Module erstellt und dort hinzugefügt, werden diese in Verbindung mit dem Modul M.2. Dokumentenmanagement archiviert.</p> <p>Voraussetzungen zur Funktionsfähigkeit</p> <p>Dieses Modul kann bzw. muss zu fast jedem Modul Verbindungen besitzen, dies ist jedoch abhängig davon, wann und wo das Modul benötigt bzw. eingesetzt werden muss. Es ist daher als übergeordnetes Modul zu verstehen, sodass Verbindungen nicht explizit genannt werden.</p> <p>Der Anwender benötigt ein mobiles Endgerät, das über eine ausreichende Hardware/Software verfügen sollte, wodurch vielfältige Funktionen unterstützt und auf dem ausgewählten Apps installiert werden können.</p> <p>Damit dies alles gleichzeitig auf dem mobilen Endgerät verfügbar ist, muss genügend interner oder externer Speicher für die anfallenden Bilder und Videos</p>

	<p>vorhanden sein. Das ist abhängig von der verwendeten Bildkompression und dem Codec, der für die Aufnahme der Videos verwendet wird.</p> <p>Die Qualität der hierbei verwendeten Verfahren muss hoch genug sein, damit auch in Umgebungen mit schwierigen Licht- und Farbverhältnissen aussagekräftige Aufnahmen gemacht werden können. Ebenfalls sollten die verwendeten Dateiformate eine hohe Verbreitung haben oder offen sein, damit eine Betrachtung mit einer Vielzahl an Systemen möglich ist.</p> <p>Unabhängig vom Gerät muss eine ausreichende Verbindung zum Netz, z. B. WLAN, und ein befriedigender Mobilfunkempfang für die nötige Send- und Empfangsqualität sichergestellt sein. Zusätzlich sollten die Funktionen weitgehend mit den derzeit bestehenden Applikationen in Aussehen und Funktion übereinstimmen, damit zum einen die Einarbeitungszeit minimiert und zum anderen eine hohe Ergonomie bei der Bedienung erreicht wird. Ebenso wird die Akzeptanz durch ein solches Vorgehen gefördert.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Aufnahme von Bildern, Audio- und Videomitschnitten • Automatisches Archivieren von Dateien
Verbindungen zu anderen Modulen	Dieses Modul kann bzw. muss zu fast jedem Modul Verbindungen besitzen, dies ist jedoch abhängig davon, wann und wo das Modul benötigt bzw. eingesetzt werden muss. Es ist daher als übergeordnetes Modul zu verstehen, sodass Verbindungen nicht explizit genannt werden.
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	mobiles Endgerät.

6.2.33 M.33. Funktionalitäten mobiles Endgerät

Komponente	M.33. Funktionalitäten mobiles Endgerät
Darstellung in Referenzarchitektur	
Zweck/Ziel	Dieses Modul beschäftigt sich mit speziellen Anforderungen, die an das mobile Endgerät des IIS gestellt werden und keiner übergeordneten Funktionalität zugerechnet werden können.
Generisch/optional	Generisch – Der Offlinemodus wird als generisch betrachtet, jedoch beinhaltet dieses Modul die optionalen Anforderungen der Spracheingabe und die

	Installationsberechtigung für Apps auf dem mobilen Endgerät. Sofern kein mobiles Endgerät vorhanden ist, kann dieses Modul als optional betrachtet werden.
Funktions- beschreibung	<p>Eine Kernfunktionalität des mobilen Endgeräts ist es, dem mobilen Arbeiter auch im Offlinemodus das Arbeiten zu ermöglichen. Besonders an den Orten der Instandhaltung wo keine Konnektivität herrscht, ist es nötig, dem Instandhalter Handlungsleitfäden, Checklisten auch offline zur Verfügung zu stellen. Die Funktionalität ermöglicht daher von den wichtigsten Daten eines Auftrages temporär Offlinekopien anzufertigen. Weiterhin werden weitere Offlinefunktionalitäten bei Bedarf implementiert. Sobald das mobile Endgerät wieder Konnektivität besitzt, müssen die Daten synchronisiert und nicht mehr benötigte Dateien wieder gelöscht werden.</p> <p>Neben der Offlinefunktionalität beinhaltet das Modul zwei optionale Funktionalitäten:</p> <ul style="list-style-type: none"> • Die erste Funktionalität ist die Installation von weiteren mobilen Applikationen bzw. Apps. Diese Apps sollten ohne großen Aufwand schnell zu installieren und anzuwenden sein. Zusätzlich müssen dabei die benötigten Berechtigungen, insbesondere Lese- und Schreibzugriff, dieser Apps überprüft werden, damit keine Sicherheitsrisiken eingegangen werden. Dies erfordert womöglich extra Schulungen oder spezielles Personal, um Qualitäts-, Datenintegritäts- und Sicherheitsanforderungen einzuhalten. Eine mögliche Liste an sicherheitsgeprüften Apps müsste bereitgestellt werden bzw. die Installation anderer verboten bzw. ausgeschlossen werden. • Die zweite Funktionalität ist die Spracheingabe, die zum Steuern bestimmter Funktionalitäten und als Diktierfunktion möglich benutzt werden kann. Beispielsweise könnten Protokolle mündlich verfasst oder einzelne Schritte des Handlungsleitfadens bestätigt werden. <p>Voraussetzungen zur Funktionsfähigkeit</p> <p>Ein qualitativ ausreichendes Mikrofon für die Spracheingabe das Störgeräusche, die in einer Industrieumgebung auftreten, herausfiltern kann. Die Aufnahme muss mithilfe eines geeigneten Codecs erfolgen, welcher zum einen die nötige Qualität besitzt und zum anderen keinen zu hohen Speicherbedarf aufweist. Zusätzlich wäre ein ausreichender dimensionierter interner oder durch zusätzliche Steckplätze erweiterbarer Speicher nötig um, die anfallenden Daten zwischen zu speichern, bevor diese synchronisiert werden.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Installationsmöglichkeit von Apps • Liste von freigeschalteten Apps empfangen • Sprachsteuerung IIS • Diktierfunktion bei Protokollen
Verbindungen zu anderen Modulen	Keine Verbindungen spezifiziert.
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Dieses Modul befindet sich auf dem mobilen Endgerät.

6.2.34 M.34. Synchronisation zw. Geräten und Systemen

Komponente	M.34. Synchronisation zw. Geräten und Systemen
Darstellung in Referenzarchitektur	<p>M.34. Synchronisation zw. Geräten und Systemen</p> <p>G.13.1. Synchronisationsmechanismus zwischen Geräten und Systemen</p> <p>U1.17. Synchronisationsmechanismus zwischen Geräten und Systemen</p> <p>U2.15. Synchronisationsmechanismus zwischen Geräten und Systemen</p>
Zweck/Ziel	<p>Das Modul Synchronisation zw. Geräten und Systemen sorgt für den Daten- und Informationsabgleich zwischen den verteilten Geräten des IIS. Dadurch wird die Konsistenz der Daten, die dem Benutzer auf dem mobilen Endgerät angezeigt werden und die im Backend vorhandenen sind, gewährleistet. Ebenso sollen Änderungen, die ein Instandhalter bspw. an den Handlungsleitfäden oder Checklisten vornimmt, auch im Backend übernommen werden, damit anderen Instandhaltern bei der Auftragserstellung die aktuellen Informationen bzw. Datensätze zur Verfügung stehen. In der Regel wird der Synchronisationsprozess von einer mobilen Anwendung initiiert. Je nach Anbindung der mobilen Endgeräte an das IIS ist auch eine Synchronisation vom Backend zum Endgerät möglich.</p>
Generisch/optional	Generisch
Funktionsbeschreibung	<p>Für den Datenabgleich muss zunächst eine Verbindung aufgebaut oder eine bestehende Verbindung genutzt werden, um Daten und Informationen auszutauschen. Solange keine zuverlässige Verbindung besteht kann keine Synchronisation erfolgen und der Verbindungsaufbau wird in regelmäßigen Abständen wiederholt. Anschließend werden von diesem Modul die relevanten Daten identifiziert und mit denen des entfernten Systems verglichen. Über einen Synchronisationsprozess werden die jeweils aktuellsten Informationen ausgetauscht und beide Entitäten auf den gleichen Stand gebracht. Mögliche Konflikte zwischen den unterschiedlichen Versionen können entweder automatisch oder durch den Benutzer gelöst werden.</p> <p>Des Weiteren sollen in dem Modul Einstellungen vorgenommen werden können, die bestimmen, wie oft welche Daten zwischen den verschiedenen Geräten und Modulen synchronisiert werden müssen.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Synchronisationsbedarf feststellen • Synchronisation durchführen • Updateverteilung • Synchronisationsoptionen je nach Modul, System und Daten konfigurieren
Verbindungen zu anderen Modulen	<p>Dieses Modul kann bzw. muss zu fast jedem Modul Verbindungen besitzen, dies ist jedoch abhängig davon wann und wo das Modul benötigt bzw. eingesetzt werden muss. Es ist daher als übergeordnetes Modul zu verstehen, sodass Verbindungen nicht explizit genannt werden.</p>
Verbundene Aktivitäten	<p>Sämtliche Use Cases, die eine Interaktion mit dem Mobilten Endgerät oder eine Auslagerung von relevanten Funktionen auf einzelne Maschinen beinhalten.</p> <ul style="list-style-type: none"> • Aktivität A10 – Synchronisation

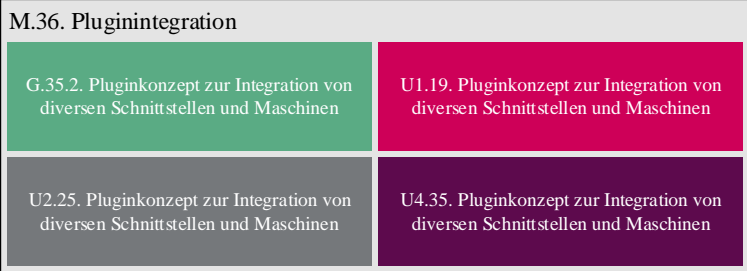
Aufteilung nach Entität	Dieses Modul befindet sich auf dem mobilen Endgerät, auf der Maschine wie auch im Backend.
--------------------------------	--

6.2.35 M.35. Maschinenspezifika

Komponente	M.35. Maschinenspezifika
Darstellung in Referenzarchitektur	
Zweck/Ziel	Dieses Modul beschäftigt sich mit Maschinenspezifika des Ressourcencockpits und angeschlossenen Maschinen bzw. der Anlagen.
Generisch/optional	Optional, denn obwohl die OPC UA Schnittstelle bei den untersuchten Unternehmen eine generische Anforderung darstellt, kann dies je nach Unternehmen stark abweichen. Darüber hinaus lassen sich bestehende Maschinenanbindungen nicht ändern.
Funktionsbeschreibung	<p>Es sollen eine Standardschnittstelle und ein standardisiertes Protokoll an den Maschinen implementiert werden, welche eine vereinfachte Integration der Maschinen ermöglicht.</p> <p>Das Protokoll, das in diesem Modul zu implementieren ist, soll folgende Funktionen und Eigenschaften bieten:</p> <ul style="list-style-type: none"> • Hohe Effizienz und teilweise Echtzeitfähigkeit • Geringer Implementierungsaufwand in die bestehende und geplante Infrastruktur („Plug and Work“) • Ausgeprägte Modularität und Erweiterbarkeit (z. B. Cloud Applikationen) • Nutzerfreundliche Bedienbarkeit über bekannte Schnittstellen • Sicherheit gegen Angriffe von außen wie innen • Moderne Verschlüsselung- und Authentifizierungsmethoden wie auch Zugriffskontrollen • Hoher Verbreitungsgrad und Akzeptanz in der Industrie durch eine Standardisierung • Offener, moderner und aktiv weiterentwickelter Standard • Möglicher Einsatz als Feldbus oder Middleware zwischen den Modulen und damit eingehende hohe Interoperabilität (vertikale und horizontale Kommunikation) <p>In den untersuchten Unternehmen wird als Machine to Maschine Kommunikationsprotokoll standardmäßig OPC UA zur Bereitstellung einiger dieser</p>

	<p>angesprochenen Funktionen verwendet. Für die untersuchten Unternehmen soll die Verwendung von OPC UA eine Standardanforderung bei neuen Maschinen/Anlagen werden. Aus diesem Grund wurde die Verwendung von OPC UA als generische Anforderung klassifiziert.</p> <p>Voraussetzungen zur Funktionsfähigkeit</p> <p>Um die volle Funktionalität nutzen zu können, ist eine möglichst störungsfreie und dauerhafte Verbindung zwischen allen Entitäten nötig, die mit dem IIS überwacht, kontrolliert oder gesteuert werden sollen. Die Art der Konnektivität ist hierbei nicht explizit spezifiziert und zudem maschinen- bzw. anlagenabhängig. Dabei sollte insbesondere das Auslesen der benötigten Informationen möglichst schnell bzw. in Echtzeit erfolgen können, was ein Protokoll mit geringem Overhead erforderlich macht. Für die Anbindung der Maschinen und Anlagen, die Einrichtung bzw. die Konfigurierung der Server sollte geschultes Personal zur Verfügung stehen, um die korrekte und schnelle Installation und Betriebsbereitschaft sicherzustellen. Die Sicherheit muss sichergestellt sein, was mehrere Aspekte umfasst: Zum einen den Schutz vor Angriffen von außen wie innen und dem Verhindern des Auslesens von Betriebsgeheimnissen durch geeignete Maßnahmen. Zum anderen die Abwehr von Manipulationsversuchen, die Mensch und Maschine gefährlich werden könnten.</p>
Mögliche Unterfunktionen	Keine Unterfunktionen spezifiziert.
Verbindungen zu anderen Modulen	Keine Verbindungen spezifiziert.
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Dieses Modul befindet sich an der Maschine.

6.2.36 M.36. Pluginintegration

Komponente	M.36. Pluginintegration
Darstellung in Referenzarchitektur	
Zweck/Ziel	Das Modul dient dem Zweck eine Pluginfunktionalität für technische Konverter (engl. Converter) bereit zu stellen. Diese dienen dem Umwandeln von maschinen- oder systemspezifischen Formaten bzw. Standards in ein Format, mit dem das IIS umgehen kann. Durch dieses Modul wird die Erweiterbarkeit durch Maschinen, Komponenten und Protokollen sichergestellt.
Generisch/optional	Generisch

Funktions- beschreibung	Dieses Modul implementiert ein Pluginsystem, das ermöglicht eine Ansammlung von Plugins in das IIS einzubinden und zu pflegen. Diese Plugins wandeln z. B. die Daten aus unterschiedlichen Anlagen in ein Format um, welches das IIS benötigt, um diese Daten zu bearbeiten. Neben dem Umwandeln von Maschinendaten sollen hier Konverter aus beliebigen Altsystemen oder z. B. Schnittstellen zu Herstellern integrierbar sein. Dies führt dazu, dass Daten von unterschiedlichen Maschinen und Systemen direkt empfangen und gesendet werden können, ohne dass die systemeigenen Formate angepasst werden müssen. Eine genauere Beschreibung findet hier nicht statt, da die Art der Plugins und der zu konvertierenden Daten zu individuell sind. Jedoch sollten alle Plugins in einem zentralen Punkt gesammelt werden, aus Gründen der Übersichtlichkeit und Wartbarkeit.
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Pluginsystem für das Einbinden verschiedener Plugins zum Senden und Empfangen von Daten aus unterschiedlichsten Quellen • (Automatisches) Umwandeln von proprietären Datenformaten in ein einheitliches Format, das im IIS Verwendung findet
Verbindungen zu anderen Modulen	Dieses Modul kann bzw. muss zu fast jedem Modul Verbindungen besitzen, dies ist jedoch abhängig davon, wann und wo das Modul benötigt bzw. eingesetzt werden muss. Es ist daher als übergeordnetes Modul zu verstehen, sodass Verbindungen nicht explizit genannt werden.
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Diese Modul befindet sich im Backend.

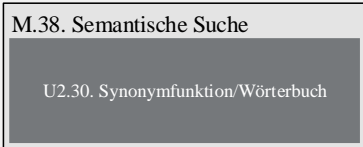
6.2.37 M.37. Wetterdaten

Komponente	M.37. Wetterdaten
Darstellung in Referenz- architektur	
Zweck/Ziel	Das Modul Wetterdaten dient dem Abrufen und anschließenden Bereitstellen von meteorologischen Daten aus verschiedenen Quellen. Mit diesen Daten können Anlagen überwacht und mögliche witterungsbedingte Gefährdungen wie Vereisung und Sturmschäden erkannt werden. Zusätzlich ermöglichen die Daten die genauere Planung, sprich Zeitpunkt und Priorität, von Instandhaltungsoperationen. Damit können Reparaturen und Inspektionen in geeignete Phasen, wie z. B. bei Windenergieanlagen passend in Schwachwindzeiten gelegt und die notwendige Zeit für die Ursachenforschung verringert werden. Ebenfalls sinkt der Anteil an reaktiver Instandhaltung und schlussendlich verlustreicher Stillstandszeiten.
Generisch/optional	Optional – nur benötigt, falls Windenergieanlagen im Einsatz sind
Funktions- beschreibung	Der Input dieses Moduls stammt aus zwei verschiedenen Domänen, die separat behandelt werden müssen. Daten von der Anlage:

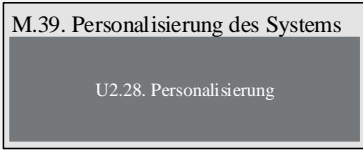
	<p>Das System hat lesenden Zugriff auf die Sensoren, die Umgebungs- bzw. Wetterdaten an der Anlage messen. Dies kann entweder direkt über die Maschine stattfinden, z. B. per Webservice, oder mithilfe eines Systems, welches die Daten von der Maschine in regelmäßigen Abständen abrufen.</p> <p>Daten von Wetterportalen:</p> <p>Webportale bieten eine in Qualität wie auch Quantität sehr unterschiedliche Auswahl von Informationen. Der Zugriff auf Daten mehrerer Portale sollte per Webzugriff möglich sein. Ein Beispiel für ein solches Portal wäre der Deutsche Wetterdienst (http://www.dwd.de/). Bei diesem rangieren die Informationen von Wetterkarten und Warnmeldungen über präzise Vorhersagen und Prognosekarten bis hin zu Klimainformationen. Zusätzlich werden anwenderspezifische Zusatzdienste angeboten. Die Anbindung wäre bei diesem Dienst u. a. mit Webservices oder Standardwebschnittstellen möglich. Technisch wäre eine Integration der Wetterdaten daher relativ einfach. Bei der Implementierung wäre zu bestimmen in welchen Intervallen die Daten abgerufen werden müssen. Entweder werden bei jeder Benutzung der Wetterdaten Abfragen gestellt oder diese werden regelmäßig im Hintergrund aktiviert. Dies hängt von den jeweiligen Möglichkeiten bzgl. der Internetverbindung, aufzubringenden Kosten und anderen Anforderungen bei den Unternehmen vor Ort ab.</p> <p>Die Herausforderung dieses Moduls liegt im Wesentlichen in der Akquisition und Verknüpfung von Daten unterschiedlicher Herkunft. Diese kommen einerseits von heterogenen Anlagen, die von verschiedenen Herstellern stammen und unterschiedlich strukturierte Daten und Informationen liefern. Diese Daten sind meist lokale Umgebungsdaten, die durch Sensorik an den jeweiligen Anlagen ermittelt werden. Andererseits sollen aktuelle Wetterdaten und -prognosen von Wetterdiensten und -portalen aufgenommen werden, um mit einer breiteren Datenbasis den Instandhaltungsprozess effektiver zu gestalten.</p> <p>Voraussetzungen zur Funktionsfähigkeit</p> <p>Um die volle Funktionalität nutzen zu können, muss eine (Remote-)Verbindung zu den Anlagen bestehen. Zusätzlich benötigt das System lesenden Zugriff auf die aktuellen Logdateien bzw. die aktuellen Systemdaten, die brauchbare Sensordaten über Temperatur/Klima enthalten. Diese Informationen müssen bereitgestellt bzw. erstellt werden. Die Auswertung sollte dabei schnell aber nicht zwangsweise in Echtzeit erfolgen. Weiterhin benötigt das System einen Internetzugang mit ausreichender Datenübertragungsgeschwindigkeit, um sich zusätzliche meteorologische Daten von Wetterportalen beschaffen zu können.</p>
<p>Mögliche Unterfunktionen</p>	<ul style="list-style-type: none"> • Abrufen von meteorologischen Daten aus verschiedenen Portalen • (Kontinuierlicher) Abruf aktueller Wetter- und Umgebungsdaten von Anlagen • Umwandlung der Daten in ein einheitliches Format • Weiterleitung von Wetterwarnungen
<p>Verbindungen zu anderen Modulen</p>	<p>Verbindungen außerhalb des eigenen Bausteins zu anderen Modulen:</p> <ul style="list-style-type: none"> • Modul M.9. Auftragsverwaltung/Priorisierung • Modul M.19. Predictive Maintenance
<p>Verbundene Aktivitäten</p>	<ul style="list-style-type: none"> • Grundaktivität • Aktivität A4 – Vereisung der Windenergieanlage (WEA)

Aufteilung nach Entität	Dieses Modul befindet sich auf dem mobilen Endgerät wie auch im Backend.
--------------------------------	--


6.2.38 M.38. Semantische Suche

Komponente	M.38. Semantische Suche
Darstellung in Referenzarchitektur	 <p>The diagram shows a box labeled 'M.38. Semantische Suche' nested inside a larger box labeled 'U2.30. Synonymfunktion/Wörterbuch'.</p>
Zweck/Ziel	Das Modul dient dazu, den Suchprozess nach Bezeichnungen, Fehlern und weiteren Begrifflichkeiten zu vereinfachen. Dies wird damit erreicht, dass automatisch Synonyme erkannt werden und Suchanfragen vervollständigt werden.
Generisch/optional	Optional
Funktionsbeschreibung	<p>Suchanfragen nach Fehlern, Begrifflichkeiten müssen auch mit Synonymen umgehen können. Aus diesem Grund muss in diesem Modul eine Datenbank mit Synonymfunktion eingeführt und gepflegt werden können. Diese soll anhand von Suchanfragen automatisch angelernt werden können. Weiterhin sollen ähnliche Suchanfragen gruppiert und vervollständigt werden können. Dies wird dadurch erreicht, dass Datenbankanfragen schon bei der Eingabe zur Verfügung stehen. Die Implementierung einer unscharfen Suche, die Wörter findet, die ähnlich der eingegebenen Zeichenkette sind, verbessert das Suchergebnis ebenso.</p> <p>Voraussetzungen zur Funktionsfähigkeit</p> <p>Ausreichende Datenbasis zur Bildung von Synonymen vorhanden.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Zuordnung von Synonymen zu Begrifflichkeiten • Automatische Synonymlernfunktion • Unterstützung einer Autovervollständigung bei der Suche
Verbindungen zu anderen Modulen	Keine Verbindungen spezifiziert.
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Backend und mobiles Endgerät.

6.2.39 M.39. Personalisierung des Systems

Komponente	M.39. Personalisierung des Systems
Darstellung in Referenzarchitektur	
Zweck/Ziel	Das Modul erlaubt die Personalisierung des IIS in Hinblick auf Veränderung der Darstellung und um eigene Notizen für Instandhalter zu hinterlegen.
Generisch/optional	Optional
Funktionsbeschreibung	<p>Das Modul stellt einen persönlichen Bereich der Anwendung zur Verfügung, bei der die Darstellung in Hinblick auf Schriftart, -größe und anderen optischen Merkmalen angepasst werden kann. Diese Änderungen werden anschließend auf das gesamte IIS der einzelnen Personen angewandt. Es wird also ein Mapping des Nutzeraccounts zu den Einstellungen erstellt.</p> <p>Weiterhin beinhaltet die Funktionalität einen eigenen Bereich für jeden Nutzer, wo persönliche Vermerke o. ä. hinterlegt werden können.</p>
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Bereitstellung eines personalisierten Bereichs des IIS • Grafische Einstellungsmöglichkeiten • Abspeichern von persönlichen Notizen
Verbindungen zu anderen Modulen	Dieses Modul kann bzw. muss zu fast jedem Modul Verbindungen besitzen, dies ist jedoch abhängig davon, wann und wo das Modul benötigt bzw. eingesetzt werden muss. Es ist daher als übergeordnetes Modul zu verstehen, sodass Verbindungen nicht explizit genannt werden.
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Auf Backend und mobilen Endgerät vorhanden.

6.2.40 M.40. Telefonersatz

Komponente	M.40. Telefonersatz
Darstellung in Referenzarchitektur	
Zweck/Ziel	Das Modul Telefonersatz stellt eine Anforderung an das mobile Endgerät dar. Das mobile Endgerät soll demnach die komplette Funktionalität eines Telefons abbilden, um dieses zu ersetzen.
Generisch/optional	Optional

Funktions- beschreibung	Das mobile Endgerät sollte ein typisches Unternehmenstelefon ersetzen. Dazu muss es alle Funktionalitäten der in dem jeweiligen Unternehmen verwendeten Mobilgeräte beinhalten.
Mögliche Unterfunktionen	<ul style="list-style-type: none"> • Telefonfunktion • Installationsberechtigung Apps
Verbindungen zu anderen Modulen	Dieses Modul kann bzw. muss zu fast jedem Modul Verbindungen besitzen, dies ist jedoch abhängig davon, wann und wo das Modul benötigt bzw. eingesetzt werden muss. Es ist daher als übergeordnetes Modul zu verstehen, sodass Verbindungen nicht explizit genannt werden.
Verbundene Aktivitäten	Keine Aktivitäten spezifiziert.
Aufteilung nach Entität	Mobiles Endgerät.

6.3 Verteilungssicht

Die Verteilungssicht legt dar, wie die Intelligenzverteilung der Module der RAI, insbesondere im Hinblick des zukünftigen Fokus auf CPS, stattfinden sollte. Grundlage zur Erstellung dieser Sicht bzw. der Intelligenzverteilung sind die Bedürfnisse der untersuchten Unternehmen sowie die zukünftige Entwicklung, die zu einer tendenziell dezentraleren Verteilung zu der Maschine bzw. dem Maschinensystem hinführt. Beide Punkte, zukünftige Entwicklungen und Status Quo, werden in dieser Sicht berücksichtigt. Grundlage dieses Kapitels sind die Ergebnisse von Reidt und Krcmar (2016), welche die Intelligenzverteilung eines CPS zur Unterstützung der Instandhaltung beschreiben. Diese Erkenntnisse werden an die Art und den Inhalt der hier vorgestellten Module angepasst.

Die Entscheidungen, welche Module auf welchem Gerät implementiert werden bzw. von welchen sie ausführbar sind, haben jedoch keinen Anspruch auf eine festgelegte und sakrosankte Ordnung. Sie gelten vielmehr als Empfehlungen. Abweichungen können und sollen abhängig von den individuellen Bedürfnissen und Gegebenheiten abgeändert werden. Denn die konkrete Ausprägung der unternehmensindividuellen Anforderungen bestimmt die Auswahl, den Zuschnitt und die Verteilung der Module. Hier gilt es zu prüfen, inwiefern eine Funktionalität auf den jeweiligen Entitäten komplett vorhanden, nur abrufbar oder überhaupt existieren muss. Als Entität wird hier der Ort der Intelligenz, auf dem sich die Funktionalität befindet, bezeichnet. Des Weiteren wird sich diese Sicht mit zunehmendem technologischen Fortschritt dahingehend ändern, dass mehr Funktionen/Module von der Maschine übernommen werden.

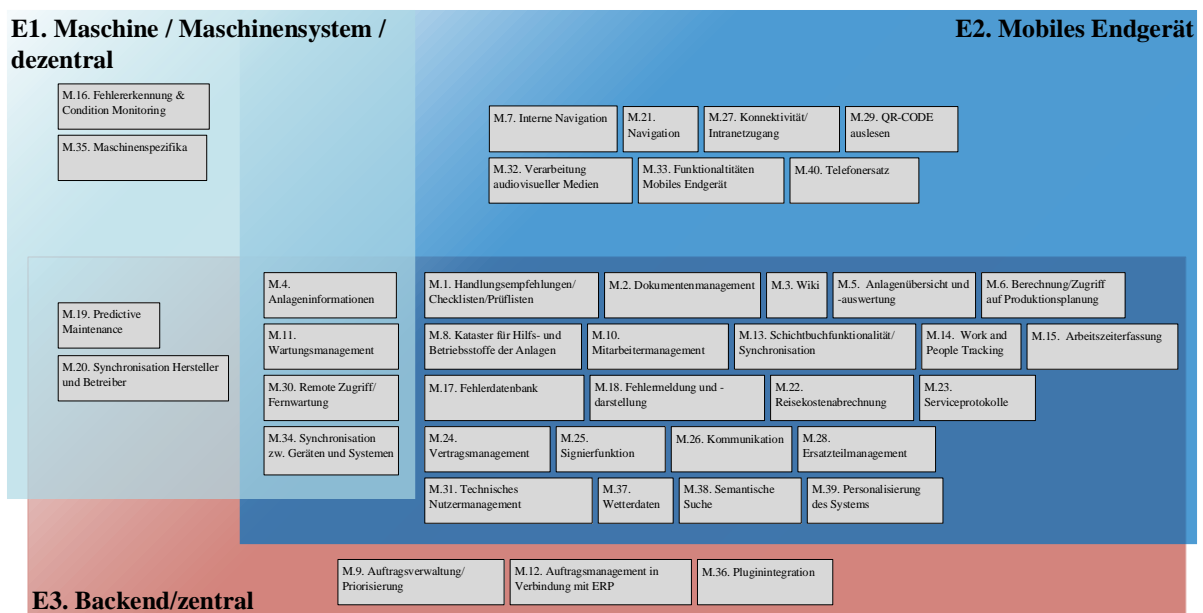


Abbildung 21: Verteilungssicht

Quelle: In Anlehnung an Reidt et al. (2016a, S. 164)

In Abbildung 21 ist die Modulverteilung der Referenzarchitektur dargestellt. Zu sehen sind drei Systeme, auf denen sich Module/Funktionen befinden. Diese Systeme werden als Entitäten bezeichnet und mit „E“ abgekürzt. Jeder Entität wird ein farbiger Block zugeordnet, der die Module auf dieser Entität umschließt. Die drei Entitäten sind:

- **E1. Maschine/Anlage** – Hiermit sind maschinennahe Systeme gemeint, die sich entweder in der Maschine selbst, oder an einem direkt an der Maschine lokal angeschlossenen System, befinden.
- **E2. Mobiles Endgerät** – Hiermit ist ein mobiles Endgerät gemeint, das der Instandhalter, Bediener oder eine dem Instandhaltungsmanagement zugeordnete Person zur Unterstützung von Instandhaltungsaufgaben nutzt. Typischerweise wird das mobile Endgerät während der Instandhaltung genutzt. Es zeichnet sich weiterhin durch ein kleineres Display und einfachere Bedienung bspw. durch Touch oder Sprache aus. Im Kontext der untersuchten Unternehmen wird ein Tablet präferiert.
- **E3. Backend** – Das Backend stellt den zentralen Server dar, auf dem die meisten Informationen zusammenlaufen. Es ist nicht spezifiziert, welche Leistung oder welche Art von Rechner das Backend haben bzw. sein sollte. Der Zugang zu dem Backend wird in der Regel durch einen Desktop PC oder Laptop erlangt. Hierfür ist ein Inter- bzw. Intranet Zugang notwendig.

Die Entitäten sind in der Darstellung überlappend angeordnet. Die so erzeugten Schnittmengen bilden Sektionen, in denen Module auf verschiedenen Entitäten parallel vorhanden sind. Es existieren demnach Module, die bspw. im Backend, wie auch auf dem mobilen Endgerät vorhanden sind, oder Module, welche auf allen drei Entitäten zu finden sind. Dies ist jedoch nicht gleichbedeutend damit, dass die identische Funktionalität auf allen Geräten vorhanden ist, sondern es kann auch sein, dass sich bestimmte Teile eines Moduls auf dem mobilen Endgerät befinden, wohingegen andere nur im Backend vorhanden sind. Als Beispiel wären hier das Modul Mitarbeitermanagement zu nennen. Auf dem mobilen Endgerät würden hauptsächlich die Funktionalitäten vorhanden sein, die dem einzelnen Instandhalter dienen, wie bspw. die Auftragsannahme oder die -anzeige. Im Backend hingegen würde die Übersicht über die aktuelle Aufgabenverteilung der Mitarbeiter vorhanden sein.

Das Modul M.11. Wartungsmanagement hingegen befindet sich im Zwischenbereich von Entität 1, 2 sowie 3. Das Modul hätte demnach Funktionalitäten auf allen drei Entitäten. Beispielsweise wäre, wie in Abbildung 29 auf Seite 211 dargestellt, ein Teil des Wartungsplans lokal auf der Maschine vorhanden. Eine Sensormessung an der Maschine würde zu einer Änderung an dem lokalen, dezentralen Wartungsplan führen. Dies wird per Synchronisation an das Backend gemeldet, was zu einer globalen Neuberechnung der Wartungspläne führen könnte. Dies würde im Backend geschehen, da nur zentral der globale Wartungsplan berechnet werden kann. Der Instandhalter sieht hingegen bei Beginn seines Arbeitstages auf dem mobilen Endgerät, welche Tätigkeiten des Wartungsplans er heute zu absolvieren hat. Weitere Informationen und Berechnungen sind auf dem mobilen Endgerät nicht erforderlich. Dieser Fall zeigt die unterschiedliche Funktionsverteilung eines Moduls auf drei Entitäten.

6.4 Prozesssicht

Die Prozesssicht bildet die Prozesse der RAI ab, die typischerweise bei der Nutzung eines IIS durchlaufen werden. Diese Prozesse werden anhand bestehender Prozesse der jeweiligen untersuchten Unternehmen und den aufgenommenen Use Cases gebildet. Hier zeigt sich, dass trotz der großen Unterschiede der Unternehmen generische Prozesse existieren, welche bis auf kleinere, spezifische Abweichungen bei allen untersuchten Unternehmen zu finden sind und alle Entitäten umspannt. Insbesondere kann in Verbindung mit den untersuchten Unternehmen eine sogenannte Grundaktivität identifiziert werden, welche den Hauptanwendungsfall des IIS abdeckt. Diese Grundaktivität – in diesem Dokument auch Referenzaktivität genannt – beinhaltet den generischen Kern der Nutzung eines IIS zur Instandsetzung bzw. Wartung. Als generischer Kern wird hier die Schnittmenge aller sehr hoch priorisierter Use Cases angesehen und die Identifikation von generischen Aktionen innerhalb dieses Prozesses. Weiterhin wird dieser Prozess von allen untersuchten Unternehmen als Kern eines IIS betrachtet und als wichtigster Umsetzungspunkt bzw. Fixpunkt bei der Entwicklung angesehen. Damit stellt diese Grundaktivität das Grundgerüst der RAI dar.

Die dargestellten Prozesse werden mit den Informationen, aus der in Kapitel 6.2 vorgestellten Funktionalen Sicht und der Verteilungssicht aus Kapitel 6.3, angereichert und in Verbindung gesetzt. Durch diese Verknüpfung geben die Prozesse in ihrer Modellierung ein aggregiertes Gesamtbild der Informationen der RAI wider. Dies geschieht mithilfe von in Kapitel 5.3 vorgestellten angepassten UML Aktivitätsdiagrammen. Durch diese wird klar erkennbar dargestellt, welche Aktionen der jeweiligen Aktivität auf welcher Entität und auf welchem Modul ausgeführt wird. So ist es möglich, die Datenweitergabe zwischen verschiedenen Modulen, Prozessen, Geräten und die Interaktion der Instandhalter zu verfolgen.

6.4.1 Grundaktivität

Die Grundaktivität der RAI illustriert den typischen Ablauf innerhalb des IIS zur Unterstützung der Instandhaltung für den Fall der Instandsetzung und der Wartung. Entstanden ist die Grundaktivität durch das Vereinheitlichen und Zusammenfassen der bei den untersuchten Unternehmen vorhanden Prozessen der Instandhaltung sowie aus später in diesem Kapitel vorgestellten Aktivitäten A1-A6 und A10, die allesamt auf den in Kapitel 6.5 vorgestellten Use Cases basieren. Diese Aktivitäten wurden dahingehend untersucht, ob und wann Gemeinsamkeiten zwischen diesen Aktivitäten auftreten und welche Aktionen vereinheitlicht werden können. Dieser Schritt wurde mehrmals iterativ mit Unternehmensfeedback durchlaufen. Es wurde ersichtlich, dass die untersuchten Aktivitäten einen ähnlichen Ablauf besitzen, welcher sich in der Grundaktivität widerspiegelt. Die Aktivitäten A7-A8 stellen Prozesse auf einer niedrigeren Abstraktionsstufe dar, dienen vor allem der Auswertung, sind nur indirekt an der Instandsetzung und der Wartung beteiligt und werden daher separat aufgeführt. Die Grundaktivität ist zweigeteilt: In einen generischen Kern des Prozesses, der immer ausgeführt wird und in weiß dargestellte Aktionen beinhaltet sowie in einen optionalen Teil, der z. B. nur den Wartungsfall betrifft oder der Teile beinhaltet, die nur bei der Implementierung von bestimmten Modulen aufgrund von spezifischen Anforderungen nötig sind. Dies wäre bspw. bei dem Modul, dass das Abrufen von Wetterdaten ermöglicht, der Fall. Dieses Modul wäre bspw. bei der Instandsetzung von Windenergieanlagen ein Kernelement des Prozesses. Aus diesem Grund wurden diese optionalen Module trotz ihrer Spezifität abgebildet. Eine Besonderheit stellen die optionalen Wartungsaktionen dar. Sie

stellen einen generischen Aspekt der jeweiligen Module dar, jedoch wird bei den untersuchten Unternehmen und Prozessen die Instandsetzung gegenüber der Wartung als bevorzugt angesehen und daher mit höherer Priorität eingestuft. Daher werden in der Grundaktivität die Wartungsaktionen als optionaler Teil der Grundaktivität deklariert.

Der Verlauf der Grundaktivität ist auf Abbildung 22 dargestellt. Die Aktivität ist in drei Sektionen geteilt, die den Entitäten aus Kapitel 6.3 entsprechen: dem zentralen Backend, dem mobilen Endgerät und der Maschine bzw. dem maschinennahen System. Auf jeder Sektion werden Aktionen ausgeführt, die sich in einem Modul befinden können und durch einen Aktivitätsfluss verbunden sind. Die Aktivität besitzt einen Start- sowie mehrere Endknoten.

Anhand der Abfolge von Aktionen wird der generische Aktivitätsfluss von einem spezifischen Startpunkt aus durch die jeweiligen Entitäten anhand einzelner Funktionen der jeweiligen Module auf diesen Entitäten abgebildet. Durch diese Darstellung wird klar ersichtlich, welche Funktionen und Module dazu beitragen, dass die Grundaktivität eines Informationssystems zur Unterstützung der Instandhaltung durchgeführt werden kann. Nachfolgend wird dieser Aktivitätsfluss Schritt für Schritt beschrieben.

Der Startpunkt der Aktivität befindet sich am oberen rechten Rand der Aktivität. Dieser spaltet sich in drei Aktionen auf, die jeweils zu einer Erstellung eines Maßnahmenauftrages führen können (4). Diese laufen kontinuierlich ab und wären:

- Die Entdeckung von Fehlern, Störungen, kritischen Zuständen oder generellen Handlungsbedarfen auf Seiten der Maschine/Anlage durch eine kontinuierliche Messung der Parameter zum Zustand der Anlage innerhalb des Moduls M.16. Fehlererkennung & Condition Monitoring. Dieses Modul sendet eine Meldung an das Modul M.9. Auftragsverwaltung/Priorisierung, sofern ein Fehler gefunden wird. Alternativ kann hier auch eine starke und unerwartete Abnutzung einer Komponente erkannt werden. Diese führt zu einer Anpassung des Wartungsplans innerhalb des Moduls M.11. Wartungsmanagement und in die Fortführung der Aktivität durch Punkt (3).
- (Optional) Eine Analyse der Sensorwerte und Logs auf der jeweiligen Anlage durch den Klassifikator des Moduls M.19. Predictive Maintenance erkennt das baldige Auftreten eines potentiellen Fehlers, woraufhin eine Meldung an das Modul M.9. Auftragsverwaltung/Priorisierung gesendet wird.
- Eine Überprüfung des Wartungsplans auf dem Backend durch das Modul M.11. Wartungsmanagement führt dazu, dass eine dringende Wartung erkannt wird. Diese muss in nächster Zeit durchgeführt werden, da ansonsten die Maschine nicht mehr sicher betrieben werden kann. Bei diesem Fall wird ebenso eine Meldung an das Modul M.9. Auftragsverwaltung/Priorisierung gesendet.

Sofern keine Auffälligkeiten oder Instandhaltungsbedarfe durch diese drei Aktionen erkannt werden, wiederholen sich die Aktionen kontinuierlich. Dieser Teilablauf der Grundaktivität ist daher eine Endlosschleife, da die Überwachung der Maschinen in der Praxis kontinuierlich fortläuft. Wird jedoch eine Meldung durch die drei erwähnten Fälle erzeugt, wird der angesprochene Maßnahmenauftrag im Backend durch das Modul M.9. Auftragsverwaltung/Priorisierung erstellt (4) und mit den nötigen Informationen aus einer Vielzahl an Modulen angereichert. Diese sind:

- M.17. Fehlerdatenbank - Hier wird der Fehlercode der Anlage in Verbindung mit den internen Informationen zu dem Fehler gesetzt und in einen internen, für das IIS übergreifende Fehlercode übersetzt. Dieser Punkt wird im Wartungsfall nicht absolviert.
- M.18. Fehlermeldung und -darstellung – Anhand des internen Fehlercodes aus M.17. Fehlerdatenbank und den Grunddaten des Fehlers aus der Fehlerdatenbank werden weitere Fehlerinformationen und -beschreibungen zu dem eingegangenen Fehler aus internen Quellen extrahiert. Dieser Punkt wird im Wartungsfall nicht absolviert.
- M.1. Handlungsleitfäden/Checklisten/Prüflisten – Handlungsleitfäden/Checklisten/Prüflisten zu dem Maßnahmenauftrag werden entsprechend dem Fehler/der Wartung extrahiert und mit dem Auftrag verbunden. Optional können auch Werkzeuglisten für den Auftrag bereitgestellt werden.
- M.10. Mitarbeitermanagement – die für die Instandhaltung zuständigen Personen für die Anlage/Fehler werden mitsamt allen beteiligten Personen auf verschiedenen Hierarchiestufen identifiziert und deren Verfügbarkeit überprüft.
- M.28. Ersatzteilmanagement – die für die Instandhaltung nötigen Ersatzteile werden zusammengestellt, deren Verfügbarkeit überprüft und ggfs. reserviert sowie deren Lagerort hinterlegt.
- M.4. Anlageninformationen – Herstellerunterlagen zu dem Auftrag und der Anlage werden mitsamt den aktuellen Maschinendaten wie Logs mit dem Auftrag verbunden.
- M.5. Anlagenübersicht und -auswertung - Auswertungen, Historien zu der betrachteten Anlage werden mit dem Auftrag verbunden.
- M.6. Berechnung/Zugriff auf Produktionsplanung – (Optional) Der Produktionsplan wird zur wirtschaftlich optimalen Festlegung des Zeitpunktes der Instandhaltungstätigkeit extrahiert.
- M.37. Wetterdaten – (Optional) Die Wetterdaten werden für den Zeitraum der Entstehung des Problems sowie zum voraussichtlichen Zeitpunkt der Instandhaltung bestimmt.

Des Weiteren werden Informationen zu alten und ähnlichen Aufträge durch das Modul M.9. Auftragsverwaltung/Priorisierung überprüft. Der optimale Zeitpunkt der Instandhaltung wird berechnet. Optional werden anschließend Inhalte bestimmt, die offline auf dem mobilen Endgerät des Instandhalters verfügbar sein sollen. Diese werden direkt an den Auftrag angehängt und können vor dem Beginn des Maßnahmenauftrages auf dem mobilen Endgerät heruntergeladen werden. Aus dem noch nicht auf konkrete Personen fixierten Maßnahmenauftrag, ist nach diesen Schritten ein konkreter Arbeitsauftrag entstanden, der an die beteiligten Personen gesendet wird.

Durch das Modul M.10. Mitarbeitermanagement wird auf dem mobilen Endgerät des ausgewählten Instandhalters der Auftrag fortan angezeigt. Dort bestätigt der Instandhalter den Auftrag. Daneben werden weitere Verantwortliche (Schichtleiter etc.) über den Auftrag durch das Modul informiert. Die Übernahme des Auftrages durch den Instandhalter wird hinterlegt und dessen Verfügbarkeit während der Instandhaltung aktualisiert. Diese Informationen werden ebenso an die verantwortlichen Personen für die Organisation der Instandhaltung bzw. Produktion weitergereicht. Auf den Fall der potentiellen Ablehnung des Auftrages durch den Instandhalter wird zur Vereinfachung in der Grundaktivität nicht eingegangen.

Nach diesen Schritten und vor der konkreten Durchführung der Instandhaltung (zu beachten ist hierbei, dass der Zeitpunkt der Durchführung der Instandhaltung deutlich von der Annahme der

Tätigkeit bzw. der Auftragserstellung abweichen kann) werden drei Aktionspfade synchron auf dem mobilen Endgerät durchlaufen:

- Die Ersatzteile werden nochmalig überprüft – sind Ersatzteile für den Instandhaltungsauftrag (noch) fehlend wird der Instandhaltungsauftrag zurückgestellt und die Aktivität endet. Dies wird zum zweiten Mal überprüft, um eine unvorhergesehene Änderung des Bestandes zu überprüfen und um den Instandhalter den aktuellen Ort der Ersatzteile anzuzeigen.
- Die Handlungsleitfäden/Checklisten etc. werden zu dem aktuellen Auftrag heruntergeladen, sodass die einzelnen Schritte abgearbeitet werden können.
- Optional werden vor Beginn der Instandhaltung die nötigen Offlinedaten, die im Auftragsmanagement hinterlegt sind, auf das mobile Endgerät heruntergeladen.

Sind die Ersatzteile verfügbar, die Handlungsleitfäden, Checklisten etc. sowie optional die Offlinedaten auf dem mobilen Endgerät vorhanden, kann der Instandhalter mit seiner Tätigkeit beginnen. Der Beginn der Tätigkeit wird im Auftragsmanagement protokolliert, die Maschine wird, falls notwendig, gestoppt. Ist die Durchführung der Instandhaltung für die durchführende Person sicher, fängt der Instandhalter mit dieser an, indem die einzelnen Schritte des Handlungsleitfadens nach und nach abgearbeitet, protokolliert und quittiert werden. Ist die Instandhaltung abgeschlossen, wird dies im Auftragsmanagement protokolliert und der Auftragsstatus wird aktualisiert.

Nach Beendigung der Instandhaltung werden folgende Aktionen ausgeführt:

- Es werden Ersatzteile nachbestellt, sofern diese verbraucht wurden.
- Die Handlungsleitfäden können durch den Instandhalter bestätigt oder angepasst werden.
- Lessons Learned in Form von Notizen können zu dem Auftrag hinterlegt werden.
- Optional kann eine Arbeitszeiterfassung des Instandhalters erfolgen.
- Im Wartungsfall wird zusätzlich im Backend der Wartungszähler zurückgesetzt, bei Bedarf der Wartungsplan angepasst und dieser mitsamt den resultierenden Änderungen gespeichert.
- Die Auftragsquittierung und die Informationen zu dem Auftrag werden an die zuständigen Personen gesendet und der Auftrag wird archiviert. Optional können Bearbeitungszeiten zu dem Fehler/zu der Wartung bzw. dem Auftrag gespeichert werden. Daneben kann optional eine Auftragszusammenfassung erstellt werden.

Nach dem Senden der Auftragsquittierung wird die Maschine wieder in Produktion genommen bzw. freigegeben und die Aktivität endet.

Während all dieser Schritte wird die Teilaktivität Aktivität A10 – Synchronisation ausgeführt, die dafür sorgt, dass bei jeder Änderung des Datenbestandes eine Synchronisation zwischen alle Entitäten erfolgt.

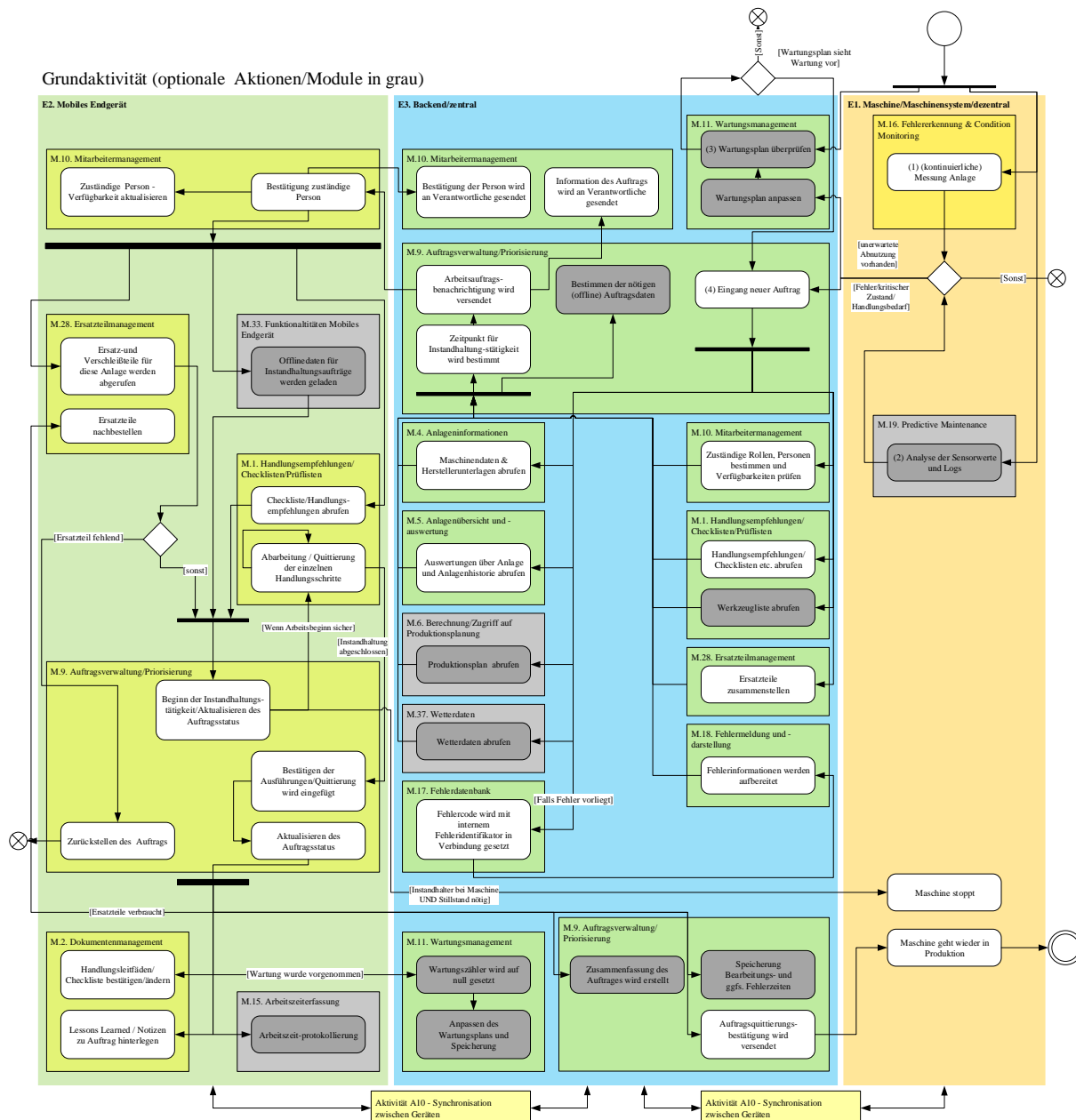


Abbildung 22: Aktivität – Grundaktivität (graue Aktionen/Module sind optional)
 Quelle: In Anlehnung an Reidt et al. (2016a, S. 171)

6.4.2 Aktivität A1 – Inspektion/Wartung Total Productive Maintenance

Die Aktivität A1 orientiert sich an dem Use Case UC1-U1: Inspektion/Wartung Total Productive Maintenance. Alle Aktionen dieser Aktivität sind in der Grundaktivität beschrieben und werden daher hier nicht einzeln aufgeführt. Die Unterschiede betreffen die unterschiedliche Zusammensetzung der Aktionen. In dieser Aktivität sind nicht alle Aktionen der Grundaktivität vorhanden und einige optionale Aktionen aus der Grundaktivität sind hier direkt abgebildet. Dieser Sachverhalt ist in Abbildung 23 dargestellt.

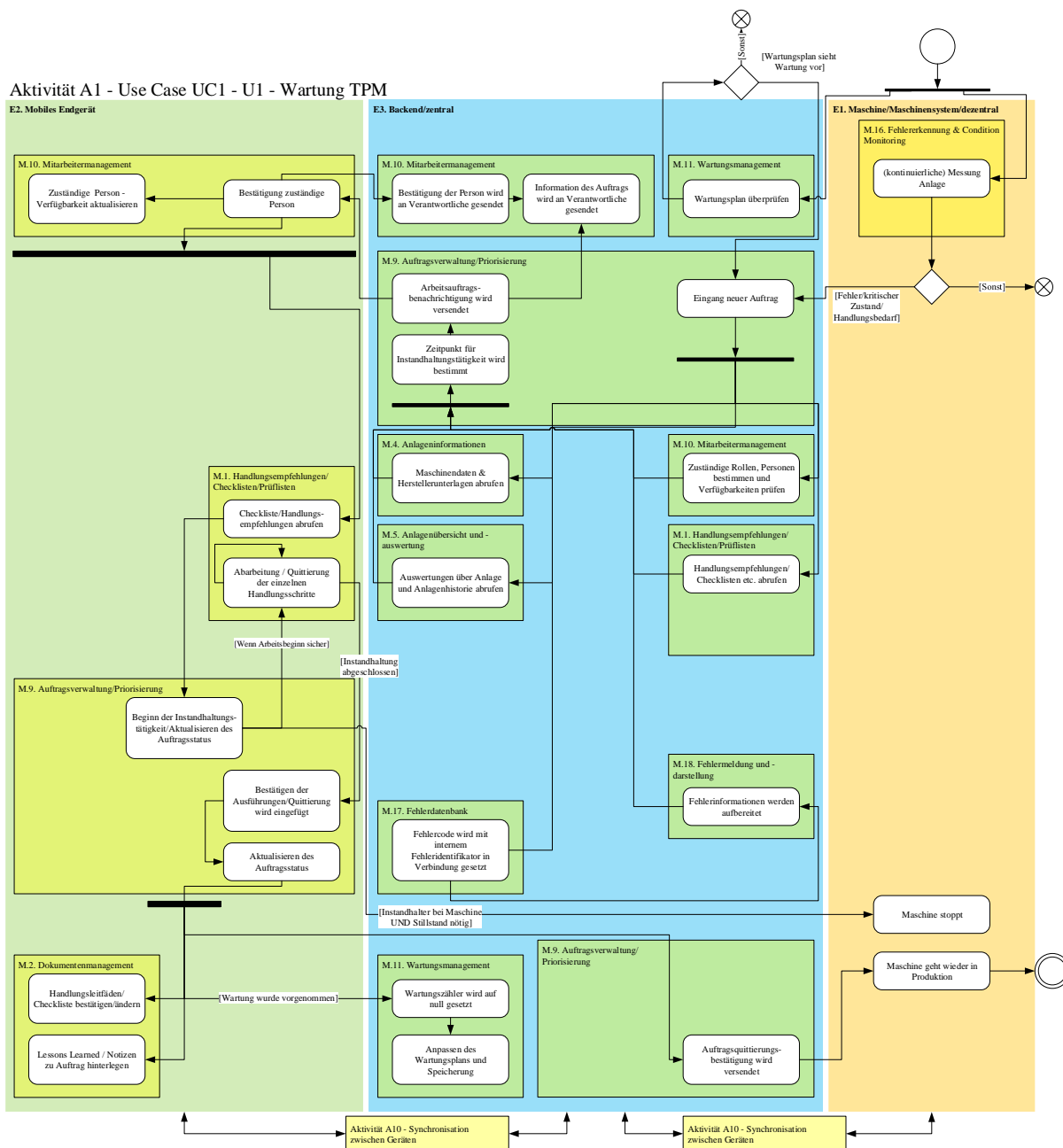


Abbildung 23: Aktivität A1 – Use Case UC1- U1 – Inspektion/Wartung TPM
 Quelle: In Anlehnung an Reidt et al. (2016a, S. 172)

6.4.3 Aktivität A2 – Inspektion Wartung Instandhalter

Die Aktivität A2 orientiert sich an den Use Case UC2-U1: Inspektion Wartung Instandhalter. Sie ist ähnlich der vorherigen Aktivität, wird jedoch vom Instandhalter und nicht vom Bediener ausgeführt. Alle Aktionen dieser Aktivität sind in der Grundaktivität beschrieben und werden daher hier nicht einzeln aufgeführt. Die Unterschiede betreffen die unterschiedliche Zusammensetzung der Aktionen. In dieser Aktivität sind nicht alle Aktionen der Grundaktivität vorhanden und einige optionale Aktionen aus der Grundaktivität sind hier direkt abgebildet. Dieser Sachverhalt ist in Abbildung 24 dargestellt.

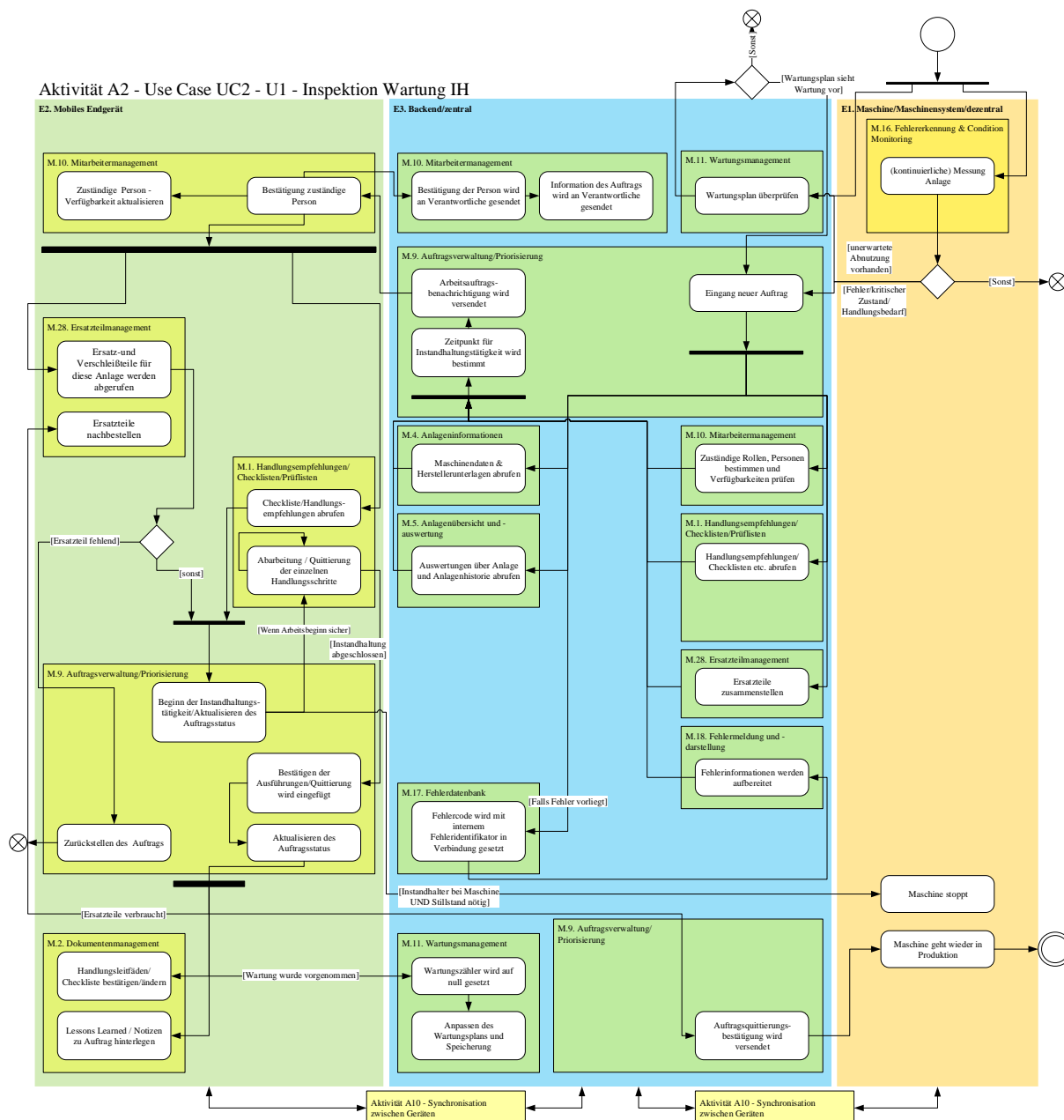


Abbildung 24: Aktivität A2 – Use Case UC2-U1 – Inspektion Wartung IH

Quelle: In Anlehnung an Reidt et al. (2016a, S. 173)

6.4.4 Aktivität A3 – Instandsetzung

Die Aktivität A3 orientiert sich an den Use Case UC3-U1: Instandsetzung. Alle Aktionen dieser Aktivität sind in der Grundaktivität beschrieben und werden daher hier nicht einzeln aufgeführt. Die Unterschiede betreffen die unterschiedliche Zusammensetzung der Aktionen. In dieser Aktivität sind nicht alle Aktionen der Grundaktivität vorhanden und einige optionale Aktionen aus der Grundaktivität sind hier direkt abgebildet. Dieser Sachverhalt ist in Abbildung 25 dargestellt.

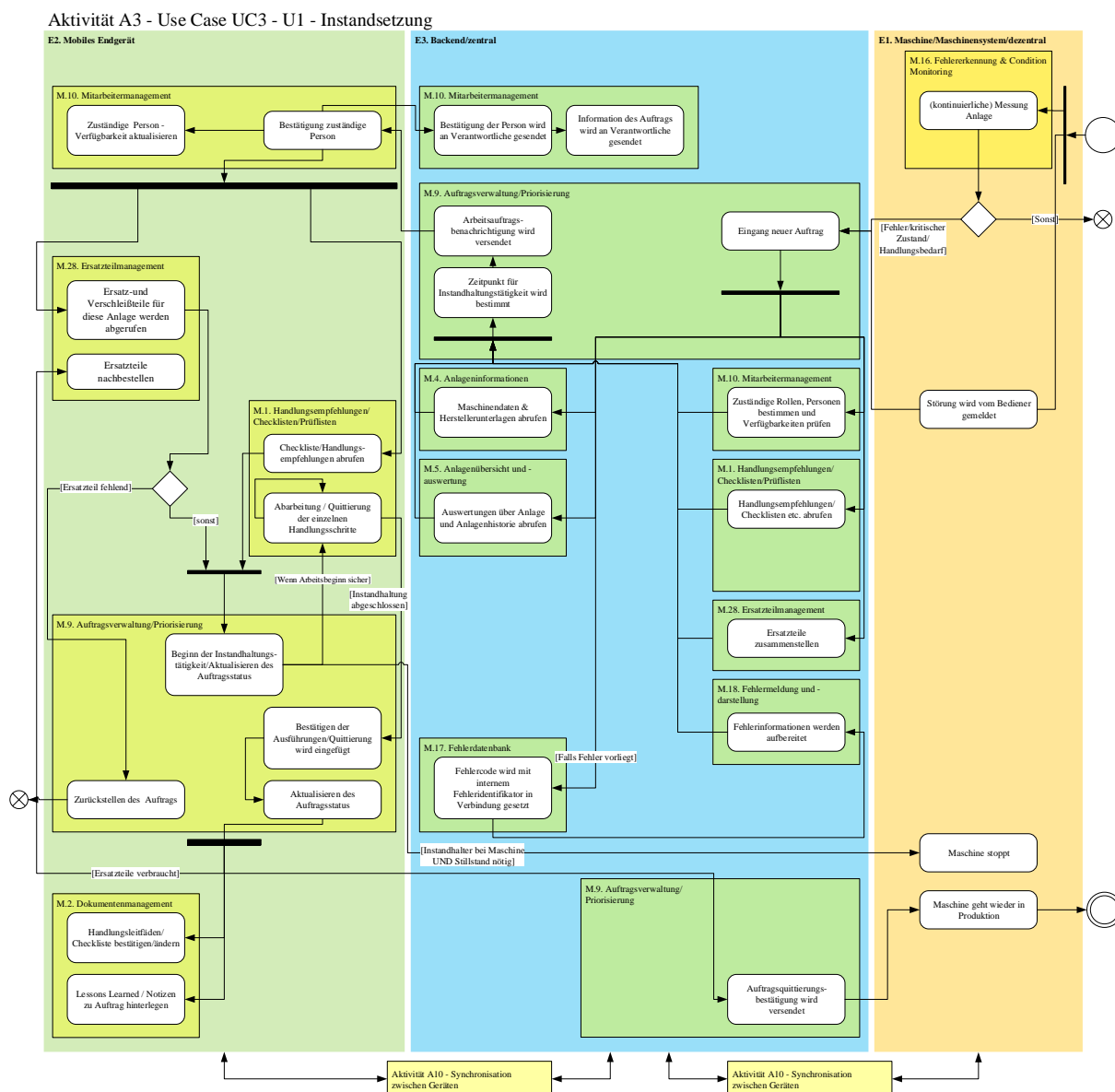


Abbildung 25: Aktivität A3 – Use Case UC3-U1 – Instandsetzung
 Quelle: In Anlehnung an Reidt et al. (2016a, S. 174)

6.4.5 Aktivität A4 – Vereisung der Windenergieanlage (WEA)

Die Aktivität A4 orientiert sich an den Use Case UC4-U3: Vereisung einer Windenergieanlage. Alle Aktionen dieser Aktivität sind in der Grundaktivität beschrieben und werden daher hier nicht einzeln aufgeführt. Die Unterschiede betreffen die unterschiedliche Zusammensetzung der Aktionen. In dieser Aktivität sind nicht alle Aktionen der Grundaktivität vorhanden und einige optionale Aktionen aus der Grundaktivität sind hier direkt abgebildet. Dieser Sachverhalt ist in Abbildung 26 dargestellt.

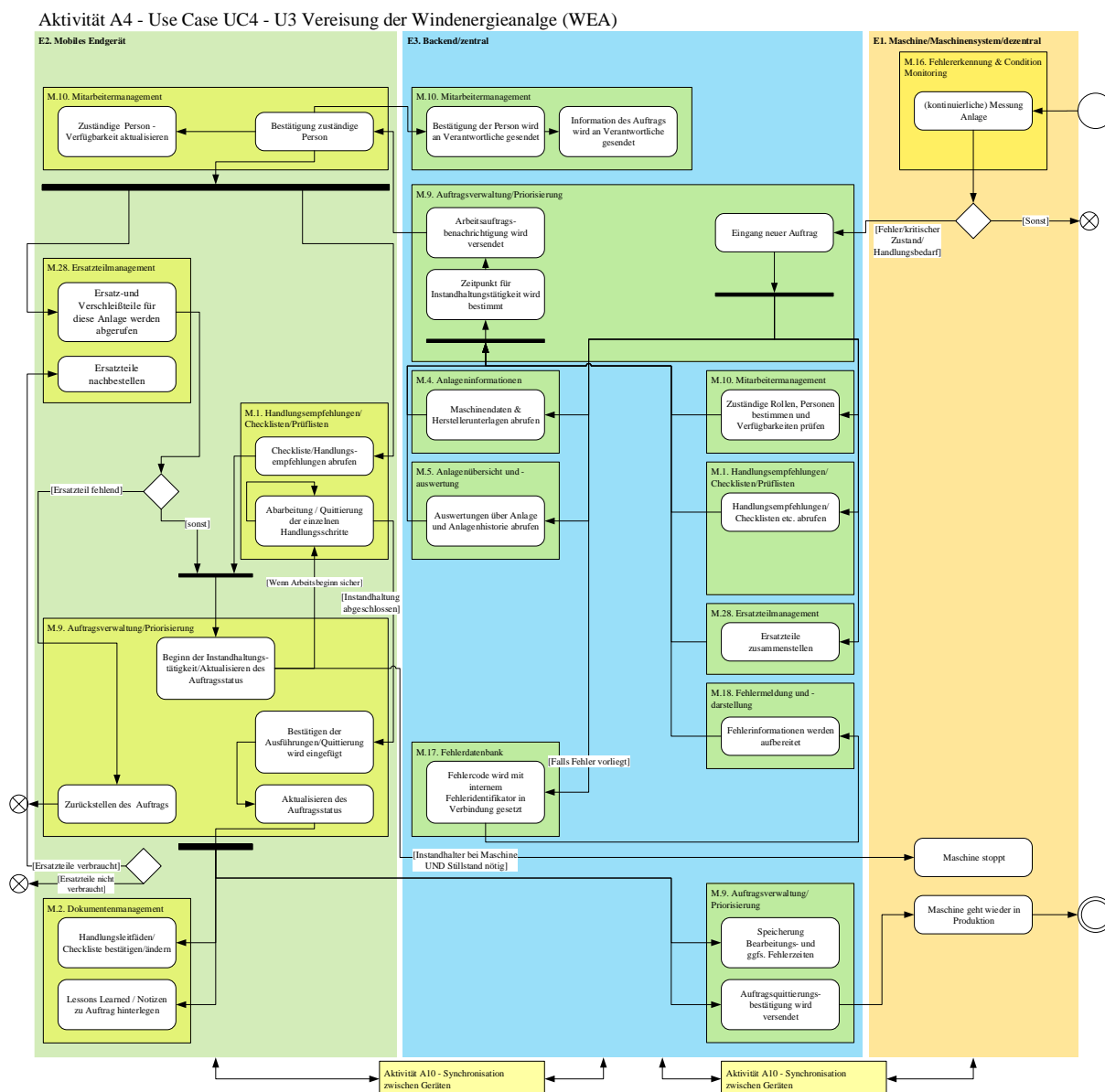


Abbildung 26: Aktivität A4 – Use Case UC4-U3 – Vereisung der Windenergieanlage (WEA)

Quelle: In Anlehnung an Reidt et al. (2016a, S. 175)

6.4.6 Aktivität A5 – Ereignisgesteuerte Instandsetzung

Die Aktivität A5 orientiert sich an den Use Case UC5-U2: Ereignisgesteuerte Instandsetzung von Anlagen. Alle Aktionen dieser Aktivität sind in der Grundaktivität beschrieben und werden daher hier nicht einzeln aufgeführt. Die Unterschiede betreffen die unterschiedliche Zusammensetzung der Aktionen. In dieser Aktivität sind nicht alle Aktionen der Grundaktivität vorhanden und einige optionale Aktionen aus der Grundaktivität sind hier direkt abgebildet. Dieser Sachverhalt ist in Abbildung 27 dargestellt.

Aktivität A5 - Use Case UC5 - U2 - Ereignisgesteuerte Instandsetzung von Anlagen

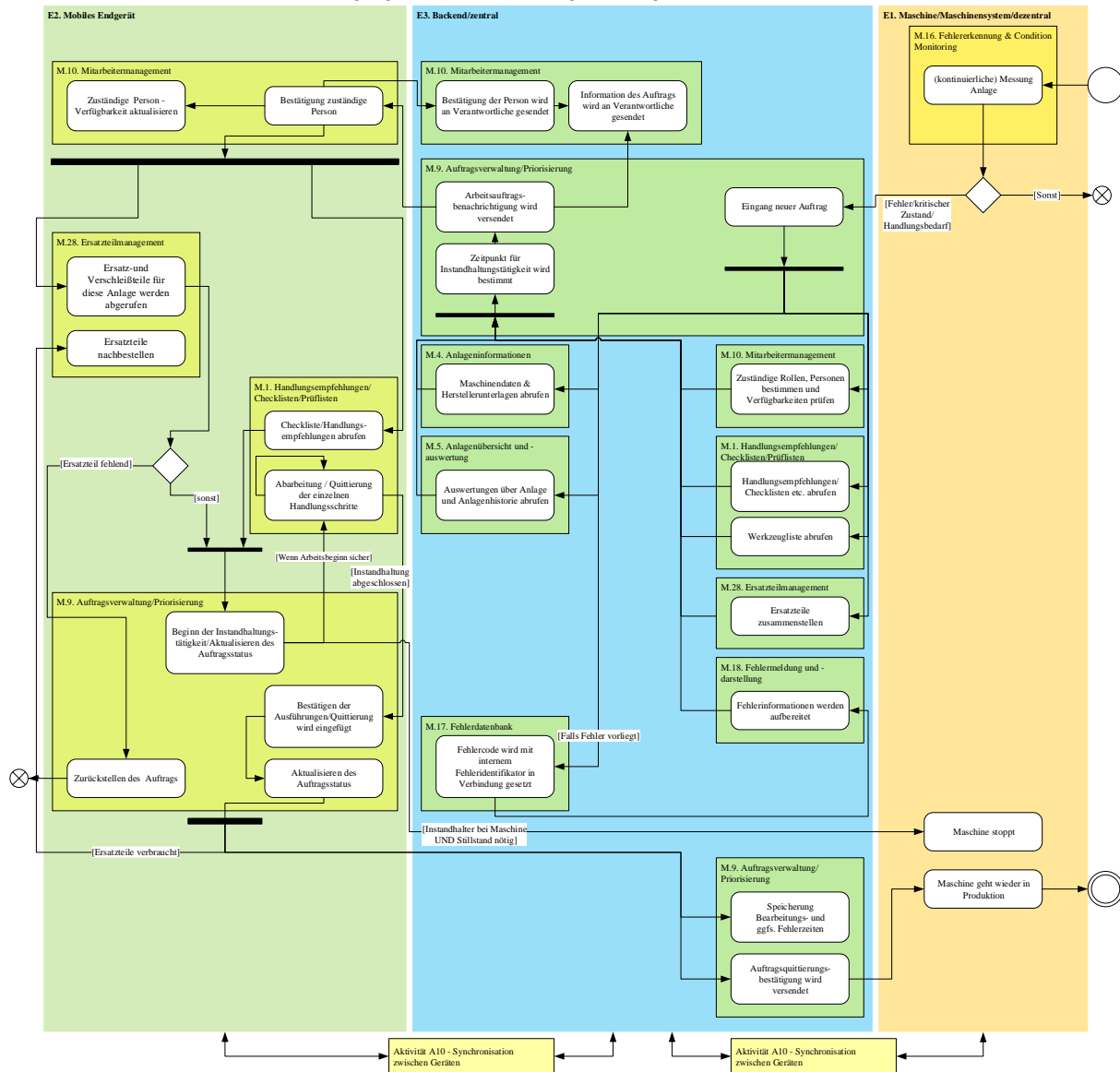


Abbildung 27: Aktivität A5 - Use Case UC5-U2 – Ereignisgesteuerte Instandsetzung von Anlagen

Quelle: In Anlehnung an Reidt et al. (2016a, S. 176)

6.4.7 Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung

Die Aktivität A6 orientiert sich an den Use Case UC6-U2: Ereignisgesteuerte Wartungsaufforderung von Anlagen. Alle Aktionen dieser Aktivität sind in der Grundaktivität beschrieben und werden daher hier nicht einzeln aufgeführt. Die Unterschiede betreffen die unterschiedliche Zusammensetzung der Aktionen. In dieser Aktivität sind nicht alle Aktionen der Grundaktivität vorhanden und einige optionale Aktionen aus der Grundaktivität sind hier direkt abgebildet. Dieser Sachverhalt ist in Abbildung 28 dargestellt.

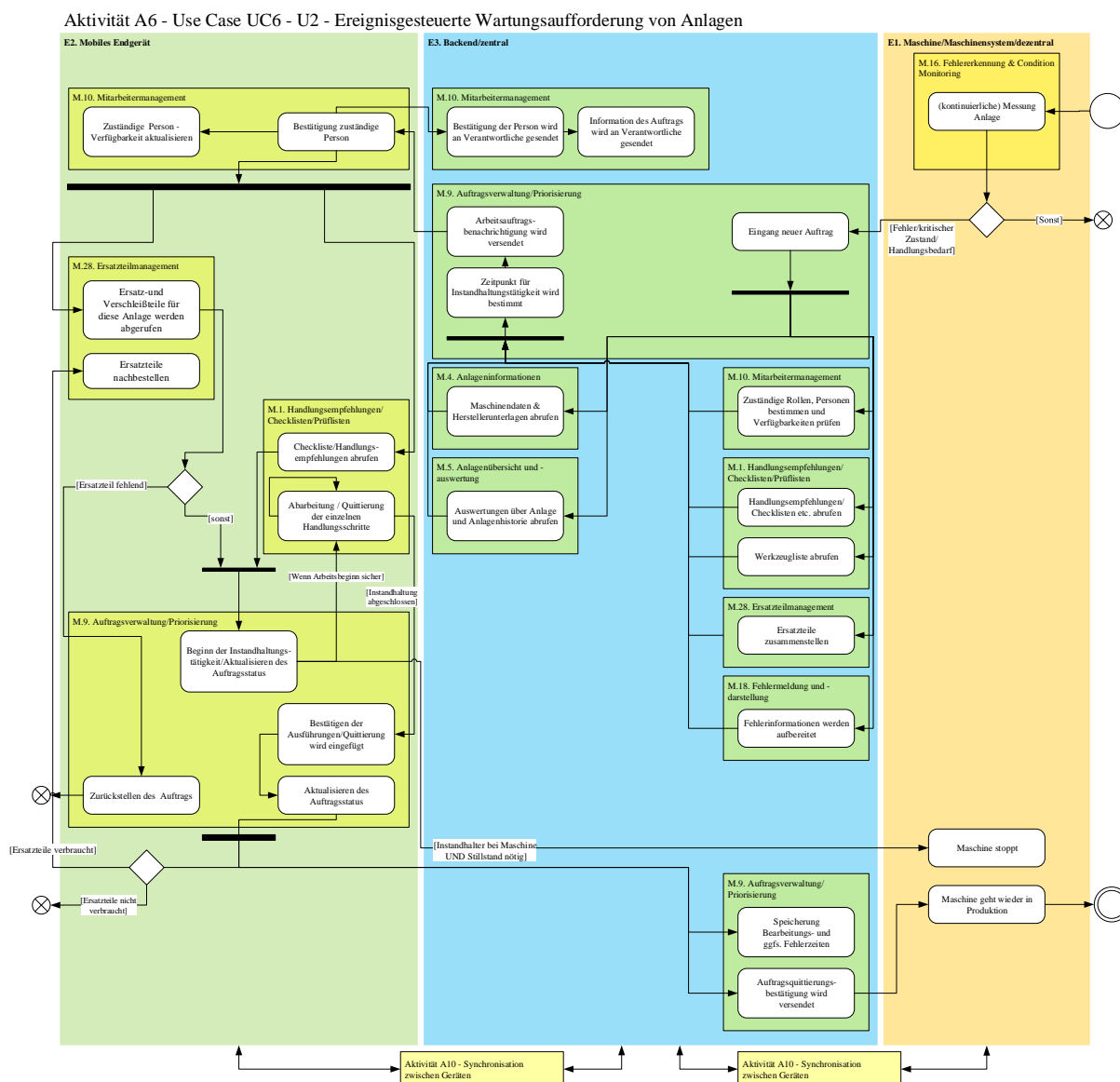


Abbildung 28: Aktivität A6 – Ereignisgesteuerte Wartungsaufforderung

Quelle: In Anlehnung an Reidt et al. (2016a, S. 177)

6.4.8 Aktivität A7 – Zustandsabfrage Wartung (Condition Monitoring)

Die Aktivität A7 ist nicht Teil der Grundaktivität. Sie ist eine detaillierte Aktivität auf niedrigerem Abstraktionsniveau und dient, wie Aktivität A8 und Aktivität A9, primär der Auswertung. Sie basiert auf dem Use Case UC7-U2: Zustandsabfrage Wartung (Condition Monitoring). Die Aktivität startet mit der Anforderung den aktuellen Status einer Anlage zu überprüfen. Der Ablauf ist in Abbildung 29 dargestellt und beginnt wie folgt:

Diese Anfrage startet typischerweise im Backend bzw. wird über diese Entität abgewickelt. Die besagte Maschine führt im Rahmen des Moduls M.16. Fehlererkennung & Condition Monitoring einen Soll-Ist-Vergleich der zu untersuchenden Komponenten durch, wodurch der Zustand aller Komponenten verfügbar gemacht wird. Der aktuelle Zustand wird anschließend mit dem ursprünglich prognostizierten Zustand verglichen. Anschließend wird der lokale Wartungsplan, der sich auf der Maschine befindet, abgerufen. Abhängig von den Veränderungen des Zustandes der Komponenten ist zu bestimmen, ob der Wartungsplan

angepasst werden muss. Sind keine Anpassungen nötig, werden für die untersuchte Anlage die Daten ermittelt, die für eine grafische Auswertung notwendig sind.

Aktivität A7 - Use Case UC7 - U2 – Zustandsabfrage Wartung (Condition-Monitoring)

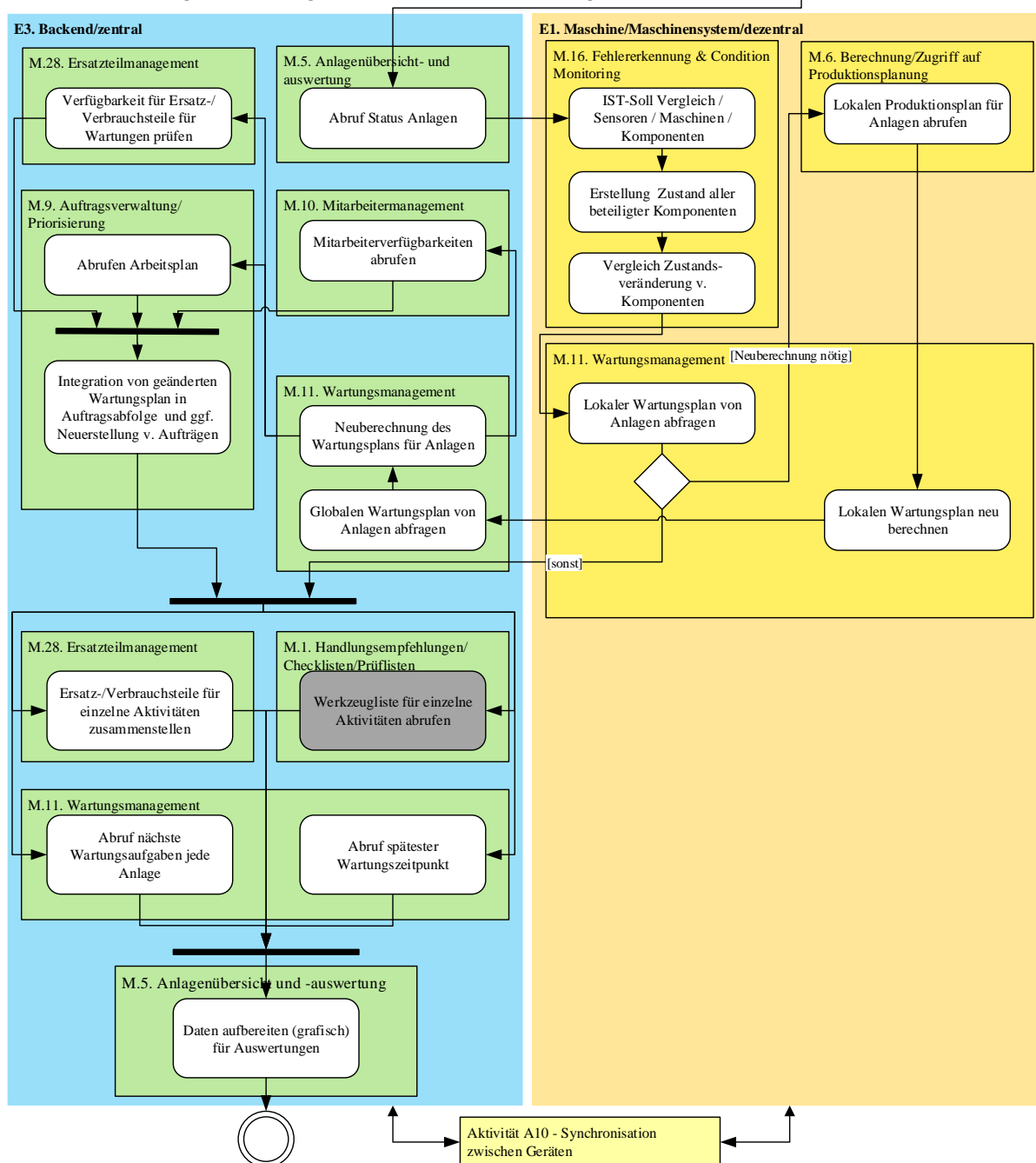


Abbildung 29: Aktivität A7 – Use Case UC7-U2 – Zustandsabfrage Wartung (Condition Monitoring)

Quelle: In Anlehnung an Reidt et al. (2016a, S. 179)

Für den Fall, dass Änderungen nötig sind, da der aktuelle Zustand schlechter als vorhergesehen ist, wird zusätzlich der lokale Produktionsplan der Maschine durch das Modul M.6. Berechnung/Zugriff auf Produktionsplanung abgerufen. Ist dieser, ebenso wie der Wartungsplan, nicht auf der Maschine vorhanden, so sind diese beiden Aktionen im Backend zu verorten. Anschließend wird der lokale Wartungsplan der Maschine durch M.11.

Wartungsmanagement abhängig von den Änderungen bzw. dem Verschleiß der Komponenten und den Produktionsplänen neu berechnet, sodass dieser den neuen Gegebenheiten gerecht werden kann.

Der neue lokale Wartungsplan wird im nächsten Schritt mit dem globalen Wartungsplan im Backend synchronisiert und neu berechnet, um die Erkenntnisse über unerwarteten Verschleiß aus dem lokalen in den globalen Wartungsplan zu integrieren. Nachfolgend werden Daten aus der Auftragsplanung, der Mitarbeiterplanung und die Verfügbarkeit von Ersatzteilen in den untersuchten Zeiträumen für die zu unternehmenden Tätigkeiten geladen, um den optimalen globalen Wartungsplan mitsamt der Auftragsplanung zu synchronisieren. Verstößt der lokale Wartungsplan gegen globale Anforderungen, wird der lokale Plan überschrieben.

Ist die Berechnung abgeschlossen, werden für die untersuchte Anlage die Daten ermittelt, die für eine grafische Auswertung nötig sind. Um diese bereit zu stellen, werden vorab die nächste Wartung, die benötigten Ersatzteile, optional Werkzeugteile und der spätestmögliche Zeitpunkt einer Wartung aufgrund der Zustandsveränderung abgerufen. Der spätestmögliche Zeitpunkt entspricht dem letzten Zeitpunkt, an dem noch keine Störung aufgrund einer unterlassenen Wartung auftritt. Diese Informationen werden anschließend so aufbereitet, dass sie für eine grafische Auswertung im Backend oder im mobilen Endgerät zur Verfügung stehen.

Dieser exemplarische Fall stellt die Zustandsabfrage und die grafische Auswertung einer Anlage dar. Bei der Konzeption dieser Aktivität ist zu beachten, dass typischerweise bei jedem Abruf keine komplette Neuberechnung des lokalen/globalen Wartungsplans durchzuführen ist, sondern dies eigenständig in bestimmten Intervallen durchgeführt wird oder Mechanismen zu implementieren sind, die den Aufwand geringhalten, so dass ein Soll-Ist-Vergleich keine großen Ressourcen kostet.

6.4.9 Aktivität A8 – Verfügbarkeit (technisch/organisatorisch)

Die Aktivität A8 ist nicht Teil der Grundaktivität. Sie ist eine detaillierte Aktivität auf niedrigerem Abstraktionsniveau und dient wie Aktivität A7 und A9 primär der Auswertung. Die Grundlage für diese Aktivität bildet der Use Case UC8-U2: Auswertungen Verfügbarkeit (technisch/organisatorisch). Die Aktivität startet mit der Anforderung, dass eine Auswertung bzgl. der Verfügbarkeit über Maschinen bzw. Anlagen gestellt wird. Dabei wird in dieser Aktivität nicht zwischen der Möglichkeit unterschieden eine oder mehrere Anlagen auszuwählen. Daraufhin werden im Modul M.5. Anlagenübersicht und -auswertung die dafür nötigen Indikatoren abgerufen bzw. berechnet. Diese sind bspw. die Bruttobetriebszeit, Gesamtstillstandszeiten aufgrund von Störungen, Gesamtstillstandzeit der Anlagen exklusive von Störungen sowie eine Kategorisierung der Ausfallzeiten der Anlagen nach bestimmten Fehlern XY. Diese Daten werden aufbereitet und anschließend wird eine grafische und tabellarische Auswertung erstellt.

Aktivität A8 - Use Case UC8 - U2 - Auswertungen
Verfügbarkeit einer Anlage (technisch/organisatorisch)

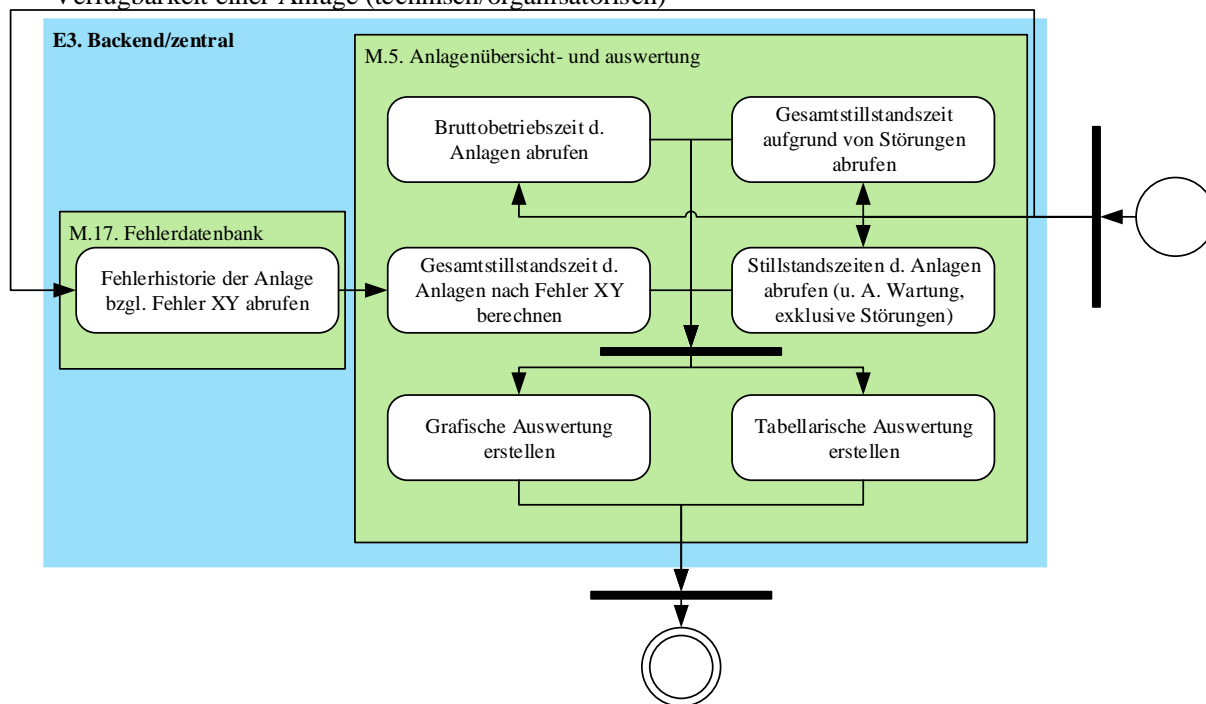


Abbildung 30: Aktivität A8 – Use Case UC8/UC9 – Auswertungen Verfügbarkeit (technisch/Organisatorisch)

Quelle: In Anlehnung an Reidt et al. (2016a, S. 181)

6.4.10 Aktivität A9 – Auswertung Fehlerhistorien einer Anlage

Die Aktivität A9 ist nicht Teil der Grundaktivität. Sie ist eine detaillierte Aktivität auf niedrigerem Abstraktionsniveau und dient wie Aktivität A7 und A8 primär der Auswertung. Die Grundlage für diese Aktivität bildet der Use Case UC9-U2: Auswertungen Fehlerhistorie einer Anlage. Die Aktivität startet mit der Anforderung, dass eine Auswertung bzgl. der Fehlerhistorie einer Anlage erstellt werden soll. Zuerst wird in Modul M.5. Anlagenübersicht und -auswertung eine Anlage ausgewählt, anschließend wird die Auswertung der Fehlerhistorie angefordert bzw. die Übersicht über diese ausgewählt. Aus dem Modul M.17. Fehlerdatenbank wird die Fehlerhistorie der jeweiligen Anlage geladen, und aus dem Modul M.18. Fehlermeldung und -darstellung werden zusätzliche Informationen und Beschreibungen zu den jeweiligen aufgetretenen Fehlern extrahiert. Anschließend werden in der Anlagenauswertung die Fehler nach deren Häufigkeit und verursachten Stillstandszeiten ausgewertet. Diese Daten werden aufbereitet und anhand dieser Daten werden eine grafische und tabellarische Auswertung erstellt.

Aktivität A9 - Use Case UC9 - U2 - Auswertungen Fehlerhistorie einer Anlage

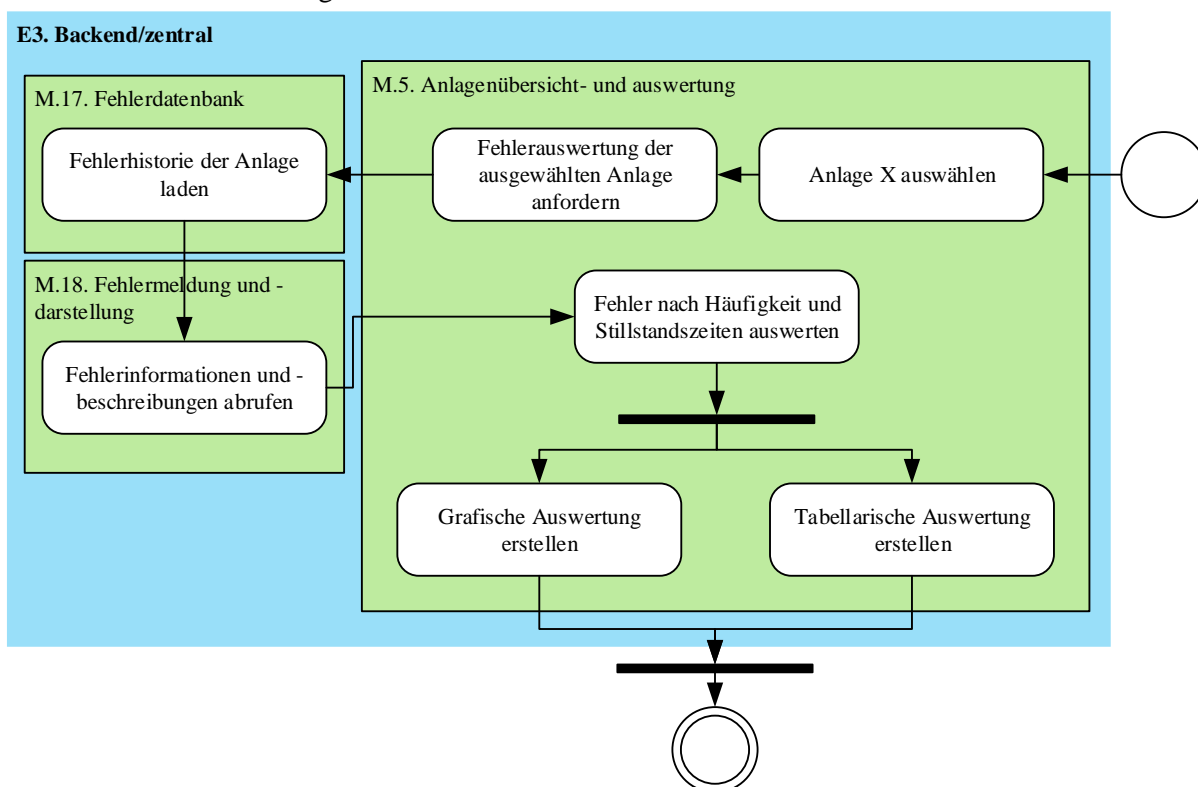


Abbildung 31: Aktivität A9 – Use Case UC9-U2: Auswertungen Fehlerhistorie einer Anlage

Quelle: Eigene Darstellung

6.4.11 Aktivität A10 – Synchronisation

Diese Aktivität ist eine Unteraktivität, der in den vorigen Kapiteln beschriebenen Aktivitäten. Sie wird immer dann aufgerufen, wenn eine Änderung im mobilen Endgerät, Backend oder der Maschine vollzogen wird, die eine wechselseitige Auswirkung auf die jeweiligen Entitäten der Referenzarchitektur besitzt. Die Aktivität wiederholt sich, bis es zu einer Verbindung und damit zu einer Synchronisation kommt. Der Start der Aktivität beginnt mit dem Aufbau einer Verbindung. Ist eine Verbindung vorhanden, wird überprüft, ob lokale Änderungen durchgeführt werden sollen oder Remote Änderungen auf den anderen Entitäten vollzogen werden. Anschließend werden die Änderungen zusammengeführt, Konflikte gelöst und die Änderungen aktualisiert.

Aktivität A10 - Synchronisation zwischen Geräten

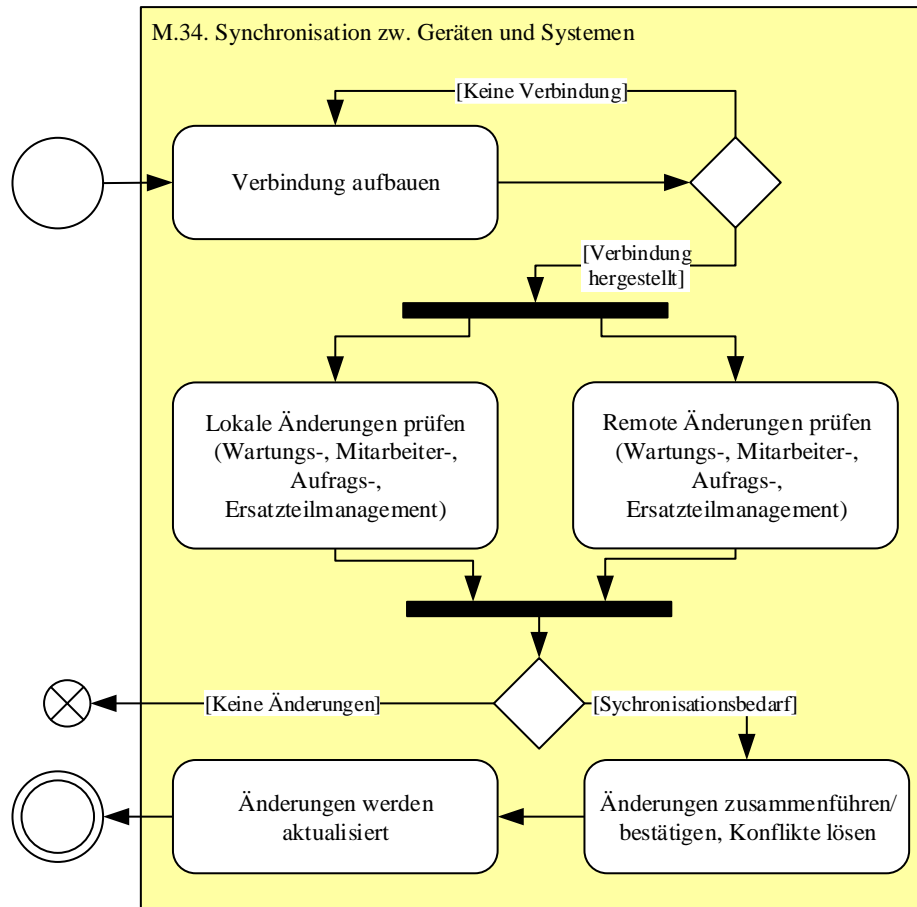


Abbildung 32: Aktivität A10 – Use Case Unteraktivität – Synchronisation

Quelle: In Anlehnung an Reidt et al. (2016a, S. 182)

6.5 Use Case Sicht

In dieser Sicht werden die Use Cases der RAII vorgestellt. Diese Use Cases bilden die wichtigsten Prozesse eines IIS innerhalb der untersuchten Unternehmen ab. Durch die große Vielfalt der Unternehmen und deren Ausprägung der Instandhaltung, können die vorgestellten Use Cases einen allgemeinen Blick auf die Instandhaltung per IIS geben. Es wird sich auf die Use Cases konzentriert, welche den Kern des Systems darstellen und die höchste Allgemeingültigkeit besitzen. Die Use Cases sind in ihrer Ausprägung jedoch abhängig von den jeweiligen Unternehmen dargestellt. Da mehrere Unternehmen betrachtet wurden, ist es ersichtlich, dass manche Use Cases Ähnlichkeiten aufweisen können. Diese ähnlich vorkommenden Use Cases sind in der RAII vorhanden, wenn deren explizite Ausführungen einen Mehrwert für die RAII mit sich bringt.

Die hier vorgestellten Anwendungsfälle fließen als Grundlage für die in Kapitel 6.4 vorgestellte Prozesssicht ein. Neben einer Eins-zu-eins-Abbildung der Use Cases als Aktivitäten, wird eine Grundaktivität erstellt, die den generischen Kern aller Use Cases darstellt. Diese Grundaktivität fasst damit die wichtigsten Use Cases in einer Visualisierung zusammen.

Der Fokus bei den untersuchten Anwendungsfällen ist, eine große Allgemeingültigkeit abzubilden, ohne dabei die spezifischen Anforderungen der betreffenden Branchen zu vernachlässigen. Aus den Analyse geht hervor, dass trotz der unterschiedlichen Branchen und Geschäftsmodelle ein großer Prozentsatz der Use Cases Ähnlichkeiten aufweisen.

Im Folgenden werden die erstellten Use Cases in aggregierter Form beschrieben. Auf die vollständige Darstellung der Use Cases wurde im Rahmen dieser Arbeit verzichtet, da die Kerninformationen durch die Prozesssicht und die aggregierte Darstellung in diesem Kapitel transportiert wird. Die vollständigen Use Cases sind bei Reidt et al. (2016a) nachlesbar. Die Use Cases sind von UC1 bis UC9 nummeriert und dem jeweiligen Unternehmen U1 bis U4 zugeordnet.

UC1-U1: Inspektion/Wartung Total Productive Maintenance

An der Maschine ist eine vorbeugende Inspektion/Wartung durchzuführen, damit ein ungeplanter Stillstand der Maschine vermieden wird. Die Tätigkeit wird vom Maschinenbediener bzw. Einrichter selbst durchgeführt. Dabei kann es notwendig sein, dass die Maschine kurzzeitig abgeschaltet werden muss. Die durchzuführenden Tätigkeiten werden vom IIS vorgegeben. Der Anwendungsfall ist abgeschlossen, sobald alle Tätigkeiten erledigt sind.

UC2-U1: Inspektion Wartung Instandhalter

An der Maschine ist eine vorbeugende Inspektion/Wartung durchzuführen, damit ein ungeplanter Stillstand der Maschine vermieden wird. Im Gegensatz zum Use Case UC1-U1 wird die Tätigkeit nicht vom Maschinenbediener ausgeführt, sondern von einem Instandhalter mit spezieller Schulung. Die notwendigen Tätigkeiten werden vom IIS vorgegeben. Der Anwendungsfall ist abgeschlossen, sobald alle Tätigkeiten erledigt sind.

UC3-U1: Instandsetzung

Eine Maschine kommt aufgrund eines Fehlers zum Stillstand. Dies wird automatisch oder durch den Bediener der Maschine an die Instandhaltung weitergeleitet. Je nach Priorität wird der

Instandsetzungsfall zeitlich und personell zugeordnet. Der zugeordnete Instandhalter bestätigt den Auftrag, arbeitet diesen ab und quittiert die Abarbeitung. Nach Abschluss der Instandsetzung ist der Fehler behoben und die Maschine ist wieder produktiv.

UC4-U3: Vereisung einer Windenergieanlage

Das Condition Monitoring System erkennt den Fehler „Vereisung Windenergieanlage“. Abhängig von Zustandsdaten der Anlage sowie aktuellen Wetterdaten, muss entschieden werden, ob die Anlage trotz Eisbildung weiterlaufen darf. Wenn notwendig, wird ein Außendienstmitarbeiter des Instandhaltungsunternehmens zu der Anlage geschickt, um deren konkreten Zustand zu bestimmen.

UC5-U2: Ereignisgesteuerte Instandsetzung von Anlagen

Das IIS wird in Echtzeit über das Auftreten einer Anlagenstörung informiert. Dabei werden alle benötigten Fehlinformationen bereitgestellt, um den Instandhaltungsprozess zu beschleunigen (Eine Senkung der Mean Time to Repair (MTTR) (Gajewski, 2004, S. 97) soll erfolgen). Zu den notwendigen Informationen gehören Anlagentyp, Fehlerdiagnose sowie Ersatzteile, falls diese für den Fehlerfall notwendig sind. Der Endzustand ist erreicht, sobald die Anlage wieder in den Betriebszustand „produktionsbereit“ versetzt wird.

UC6-U2: Ereignisgesteuerte Wartungsaufforderung von Anlagen

Das IIS wird in Echtzeit über das Erreichen eines Grenzwertes informiert. Die Maschine ist zu diesem Zeitpunkt noch einsatzfähig, benötigt aber zeitnah eine Wartung. Ziel ist es, den Abnutzungsvorrat wiederherzustellen, um einen ungeplanten Stillstand zu vermeiden (Steigerung Mean Time Between Failures (MTBF) (Mikler, 2015, S. 238)). Der Endzustand ist erreicht, sobald der Abnutzungsvorrat der Anlage wieder in den Ausgangszustand versetzt wird.

UC7-U2: Zustandsabfrage Wartung (Condition Monitoring)

Der Anwendungsfall wird verwendet, um einen Wartungsplan zu erstellen, der auf Basis des dezentral an der Anlage erfassten und prognostizierten Zustands, berechnet wird. Am Ende soll ein Wartungsplan stehen, der Wartungen so einplant, dass die Wartungen möglichst in produktionsfreien Zeiten durchgeführt werden und dass die spätestmöglichen Zeitpunkte bekannt sind an denen Wartungen durchzuführen sind.

UC8-U2: Auswertungen Verfügbarkeit (technisch/organisatorisch)

Analyse der Anlagen auf Engstellen hinsichtlich der Verfügbarkeiten und deren Verläufe. Analyse des Zielerreichungsgrades hinsichtlich der Instandhaltungsoptimierung. Auf Basis dieser Daten können Instandhaltungsverbesserungen nachvollzogen oder Bedarfe abgeleitet werden.

UC9-U2: Auswertungen Fehlerhistorie einer Anlage

Analyse der Anlagen auf Verbesserungen und Verläufe hinsichtlich der Störungsreduzierung. Dazu werden Fehlerhistorien von Anlagen erstellt, abgerufen und ausgewertet. So können kritische und häufige Fehler entdeckt werden.

Aus den vorgestellten Use Cases wird ersichtlich, dass diese in fünf Kategorien eingeteilt werden können:

- Instandsetzung von Anlagen
- Wartung/Inspektion von Anlagen durch Bediener
- Wartung/Inspektion von Anlagen durch Instandhalter
- Zustandsabfrage und Wartungsbedarfsanzeige von Anlagen
- Auswertung über die Verfügbarkeit der Anlagen sowie die
- Auswertung über die Fehlerhistorien

6.6 Diskussion Referenzarchitektur

Die in den vorherigen Kapiteln vorgestellte Referenzarchitektur zeigt in verschiedenen Sichten die Grundstruktur eines umfassenden und IIS. Bestimmte Entscheidungen, die bei der Erstellung getroffen wurden, können und wurden bei deren Erstellung intensiv diskutiert. Zum Zeitpunkt und in der Nachbetrachtung lassen viele der Punkte durchaus andere Designentscheidungen zu. Nachfolgend werden einzelne Punkte der Referenzarchitektur diskutiert, ggfs. die Designentscheidung begründet und andere Entscheidungsmöglichkeiten skizziert. Darüber hinaus wird auch auf mögliche Erweiterungen eingegangen.

Konkretheit/Abstraktheit der Referenzarchitektur

Eine grundlegende Entscheidung bei der Referenzarchitektur, deren Darstellung, der gesamten Beschreibung der Prozesse und Module war es, eine möglichst einheitliche Form der Abstraktheit im Allgemeinen bzw. eine einheitliche Abstraktionsstufe für jede Sicht zu wählen. Durch eine adäquate Abstraktion sollte zum einen ermöglicht werden, dass die Referenzarchitektur für eine möglichst große Menge an IIS gültig ist. Zum anderen sollte durch die Abstraktion die Verständlichkeit und dadurch das Hervorheben bestimmten Aspekte sowie domänenübergreifenden Diskussionen zwischen Personen erleichtert werden. Im Rahmen der RAII sollte dazu u. a. das Verständnis der Anforderungen erhöht und deren Verbindung zu den typischen Aktivitäten von Instandhaltungsfällen dargestellt werden.

Ein weiteres Ziel bei der Erstellung der RAII war es, dass die Referenzarchitektur technologieunabhängig sein soll und ebenso die Wahl verschiedener Architekturtypen ermöglicht. Dieses Vorgehen wurde gewählt, um insbesondere Altsysteme oder -architekturen leichter integrierbar zu machen und nicht einem bestimmten Architekturmuster zu folgen bzw. einen Architekturtypus vorzugeben, der dem aktuell bei Unternehmen vorherrschenden Typus widerspricht. Dadurch musste die Referenzarchitektur so modelliert werden, dass sie von Entscheidungen bzgl. bestimmter Architekturmuster abstrahiert. Diese grundlegenden Ziele führen unweigerlich zu einem hohen Grad an Abstraktion.

Dabei wird in der Ausprägung der RAII trotz des generellen Bestrebens eine einheitliche Abstraktionsstufe zu erreichen, durchaus variiert. Diese unterschiedlichen Abstraktionsstufen wurden in den Fällen gewählt, bei denen bestimmte Elemente der RAII hervorgehoben werden sollten oder weil in manchen Modulen die generischen Anforderungen eine konkretere Ausgestaltung ermöglichte als in anderen. Die Wahl der Abstraktion hätte jedoch auch anders gewählt werden können, je nachdem wie die Ausrichtung der RAII gewesen wäre und welche Elemente betont werden sollten. Auch wäre es bei der aktuellen Version möglich gewesen, Aspekte abstrakter, aber auch konkreter zu fassen.

Nachfolgend werden bestimmte Aspekte diskutiert, die unmittelbare Auswirkungen auf die Wahl der Abstraktion haben bzw. die Wahl der Abstraktion auf diese Elemente:

Allgemeingültigkeit: Je abstrakter die Elemente gewählt werden, desto höher ist die Wahrscheinlichkeit, dass eine Allgemeingültigkeit bzw. eine breite Gültigkeit erzielt wird. Auf höheren Ebenen der Abstraktion können Unterschiede abstrahiert werden, sodass die Kernfunktionalität von bspw. Modulen identisch bleiben. Werden jedoch Designentscheidungen konkret explizit definiert, sprechen diese evtl. gegen bestimmte Technologien, Systeme oder passen nicht mehr auf bestimmte Anwendungsfälle. Nichtsdestotrotz wäre es an einigen Stellen aber möglich gewesen, bei der RAII viele Aspekte

konkreter zu formulieren oder mehrere Prozesse abzubilden bzw. diese Schritt für Schritt zu konkretisieren. Auch hätten verschiedene Arten von Entscheidungen in Form von Variationen bereitgestellt werden können. So wäre eine höhere Konkretheit bei gleichzeitiger Offenheit möglich gewesen. Bei bestimmten Modulen wurde dies auch diskutiert und skizziert. Ein solche Entscheidung bedarf jedoch, dass alle Variationspunkte bekannt sind und sie steigert den Umfang von der Referenzarchitektur immens.

Technologieunabhängigkeit: Die Abstraktheit, die gewählt wurde, ermöglicht weitestgehend eine Technologieunabhängigkeit, sodass weder Programmiersprache, Architekturkonzepte, Schnittstellen etc. festgeschrieben werden. Die Abstraktion ermöglicht es, dass die RAII nahezu keinerlei Einschränkungen festlegt und die Design- und Technologieentscheidungen bei den jeweiligen Entwicklern bzw. Entscheidern vor Ort liegt. Weiter hat dies den Vorteil, dass getroffene Aussagen unabhängig von Technologietrends bestehen bleiben. Würde sich die Abstraktion auf einem anderen Level befinden, wäre diese Offenheit nicht immer möglich gewesen, da bei vielen Modulen Entscheidungen getroffen werden müssten, die nur unter Beachtung einer bestimmten Technologie sinnvoll gewesen wären. Grundsätzlich denkbar wäre es aber auch, auf bestimmte Technologiekonzepte zu setzen bzw. Technologien zu fixieren. Dies würde jedoch zu den im vorherigen Abschnitt erwähnten Einschränkungen hinsichtlich der Gültigkeiten der Referenzarchitektur führen. Insbesondere bei der Wahl von Architekturmustern wären solche Konkretisierungen möglich gewesen. Diese werden im weiteren Verlauf des Kapitels näher diskutiert.

Bei der RAII wurden jedoch auch zwei Einschränkungen vorgenommen: Es wurde angenommen, dass ein mobiles Endgerät Teil eines integrierten Instandhaltungssystems ist. Ein in heutiger Zeit aus Sicht des Autors aber fast allgemeingültiger Aspekt, da Instandhalter meist eine mobile Unterstützung bedürfen. Weiter wurde **OPC UA** explizit als Technologie für den Datentransfer integriert. Im Bereich der untersuchten sowie weiteren Unternehmen ist dieses Maschinenprotokoll ein aktueller Trend und galt im deutschsprachigen Raum für produzierende Unternehmen während der Entwicklung als zukünftiger Standard. Daher war OPC UA bei allen zu entwickelnden Prototypen als zu unterstützender Standard gesetzt. Eine alternative Wahl wäre hier jedoch auch möglich gewesen. Die RAII wurde trotz dieser expliziten Nennung von OPC UA dahingehend konzipiert, dass prinzipiell ein anderes Maschinenprotokoll gewählt werden kann (siehe M.36. Pluginintegration). Beispielsweise war bei der Erstellung der RAII lange die Verwendung bzw. die zusätzliche Verwendung des Message Queue Telemetry Transport (MQTT) Protokolls (OASIS Committee, 2017) oder RabbitMQ (Pivotal Software Inc., 2018) im Gespräch. Jedoch wurde davon abgesehen, da die bisherige Struktur die Nutzung bzw. Integration schon erlaubt und keine Technologie explizit genutzt werden sollte, die nicht explizit in den Anforderungen genannt wurde. Weiter ist es möglich, mehrere Protokolle parallel zu benutzen.

Ausrichtung: Die Wahl der Abstraktheit beeinflusst ebenso die allgemeine Ausrichtung der Referenzarchitektur. Wären Teile der RAII auf Codebasis modelliert worden, so hätte der Fokus deutlich stärker auf technischer Ebene gelegen und viele Entscheidungen hinsichtlich Technologie und Designentscheidungen müssten schon vorab getroffen worden. Die Referenzarchitektur wäre damit eine beispielhafte Implementierung auf Basis einer bestimmten Technologie. Eine Ableitung wäre für andere Entwickler, die ebenso auf diese Technologie setzen, deutlich vereinfacht. Auch hätten dadurch viele Entscheidungen bzgl. der Anforderungen getroffen werden müssen, da eine Implementierung von Anforderungen diese genauer abbildet. Das hätte den Vorteil, dass die Anforderungen auch eindeutiger wären, den

starken Nachteil, dass der Interpretationsspielraum eingeschränkt worden wäre und somit auch die Gültigkeit der Referenzarchitektur.

Verständlichkeit: Ein weiterer wichtiger Punkt bei der Wahl der Abstraktion ist die Verständlichkeit, die durch eine hohe Abstraktion in bestimmten Fällen gefördert werden kann. Durch die Reduzierung auf bestimmte Elemente könnten diese schneller erfasst und diskutiert werden. Jedoch wäre es bei der RAII durchaus möglich gewesen, die einzelnen Aktivitäten feingranularer festzulegen und deutlicher zu unterteilen. Ebenso wäre dies bei Modulen möglich gewesen und demzufolge plausibel, einige Module noch konkreter zu formulieren, weiter zu detaillieren und Lösungswege noch ausführlicher zu beschreiben. Bei Diskussionen während der Entwicklung wurde die dargestellte Abstraktionsstufe von allen Beteiligten präferiert, da durch das Weglassen bzw. Offenlassen bestimmter Details das Wesentliche besser fokussiert wird. Auch sind dadurch die entscheidenden Informationen schneller aufzufassen. Jedoch existiert bei der Erstellung einer Referenzarchitektur ein gewisser Grad an Subjektivität bzgl. der Wahl des Fokus und bei der daraus resultierenden Darstellungsart. Durch die Evaluation der RAII konnte sichergestellt werden, dass die getroffenen Darstellungsentscheidungen zur besseren Verständlichkeit und zur Unterstützung der domänenübergreifenden Entwicklung führen.

Umfang: Mit der im vorherigen Punkt angesprochenen Verständlichkeit geht der Aspekt des Umfanges einer Referenzarchitektur Hand in Hand. Prinzipiell wäre es möglich gewesen, viele Elemente und insbesondere die Module deutlich konkreter und mit festen Variationen anzureichern. Genauso wäre es möglich gewesen, die einzelnen Diagramme konkreter zu machen und feingranularer aufzubereiten. Der Umfang der Referenzarchitektur wäre dadurch jedoch stark angestiegen und hätte die angesprochenen Nachteile einer reduzierten Verständlichkeit. Die Gewichtung der Darstellung bzw. den Umfang der Darstellung hätte je nach Fokus der zu entwickelnden Referenzarchitektur geändert werden können, sodass eine größere Anzahl an Use Cases und Aktivitäten in die Referenzarchitektur hätte integriert werden können. Es wurde sich für diese reduzierte Darstellung entschieden, um so viel Umfang und erweiterbare Ansichten zu haben, dass die RAII leicht konsumierbar bleibt.

Form der Anforderungen: Während der Erstellung der Referenzarchitektur und insbesondere während der Auswahl der Anforderungen in Literatur und Praxis war die Form der Anforderungen bzw. der Moduldarstellungen und Beschreibungen ein Thema, das fortlaufend diskutiert wurde. Die Anforderungen wurden in mehreren Stufen ausgewählt, aggregiert und mehrmals validiert. Dabei musste sich in jeder Stufe auf eine Form der Anforderungsbeschreibung festgelegt werden. Hier wurden verschiedene Arten von Beschreibung getestet. Von Beschreibungen mit Identifikator, Beschreibung, Problembeschreibung, Quelle und Abnahmekriterium wie bspw. bei Robertson und Robertson (2008, S. 264) bis hin zu formalen Sprachbeschreibungen. Um die Anforderungen leichter darzustellen und weil aufgrund des Vorgehens viele Anforderungen aus der Praxis bzw. auch aus der Literatur wegen der Aggregation und fehlenden dortigen Beschreibung eine große Offenheit besitzen, wurde nur die Kurzversion von Anforderungen wiedergegeben. Diese Entscheidung kann kritisch hinterfragt werden, da dadurch eine gewisse Mehrdeutigkeit der Anforderungen entsteht. In der RAII werden die Anforderungen nur textuell und innerhalb der Module und durch die einzelnen Sichten weiter spezifiziert, das heißt durch die Beschreibungen in der RAII werden die Anforderungen erst konkreter, lassen aber dennoch bewusst Interpretationsmöglichkeiten. Wären die Restriktionen bzgl. Umfang und Verständlichkeit

nicht gewesen, hätte es durchaus sinnvoll gewesen sein können, den Anforderungsteil deutlich aufzuwerten und die Anforderungen wohldefinierter zu beschreiben.

Die genannten Punkte und Designentscheidungen können bei der Erstellung von Referenzarchitekturen je nach angestrebtem Ziel und Zweck unterschiedlich getroffen werden. Im Weiteren werden konkret genannte Entscheidungen, die indirekt die Abstraktion betreffen haben und konkrete Entscheidungen bzgl. Darstellung und Modulaufteilung diskutiert.

Wahl von Architekturmustern

Eine weitere Designentscheidung bei der RAII war es, keine starre Form von Architekturmustern auszuwählen. Die Architektur in der jetzigen Form hat Bestand, weitestgehend unabhängig davon, ob später eine ereignisgesteuerte oder monolithische Architekturform bei der Implementierung gewählt wird. Es wäre prinzipiell auch möglich gewesen, eine Art dieser Architekturmuster vorab zu wählen und die RAII darauf auszurichten. Während der Entwicklung der RAII wurde lange darüber diskutiert, eine Kombination aus **Microservicearchitektur** mitsamt einer **ereignisgesteuerten Architektur** anzustreben.

Prinzipiell wäre es eine denkbare Lösung gewesen, die RAII auf Basis eines bestimmten Architekturmusters zu entwerfen. Damit wäre eine Realisierung der Referenzarchitektur auf einem deutlich konkreteren Level möglich gewesen. Hier wären jedoch einige Module nötig geworden, die in keinen Anforderungen genannt worden wären, da es sich nur um rein für das Muster notwendige Module gehandelt hätte. Dies hätte Einfluss auf die Sichten und den Zuschnitt der Module gehabt: Module für einen Nachrichtenbus und weitere spezifische Konzepte aus dem Bereich von ereignisgesteuerten Architekturen hätten eingebunden werden können. Ebenso wäre der Schnitt der Module bei Microservicearchitekturen deutlich feingranularer möglich gewesen. Dies hätte jedoch, typisch bei Microservicearchitekturen, zu einer deutlich geringeren Verständlichkeit geführt.

Datenbanken in Modul oder extern

Ein weiterer Aspekt, der von der Referenzarchitektur nicht direkt behandelt wird, ist die Bestimmung von Datenbankschemata oder Datenbankkonzepten. Das heißt: wo befinden sich die Daten für bestimmte Module, existieren Formen von Data Lakes oder gibt es zentrale oder dezentrale Formen von Datenhaltung. Hier wäre es ebenso möglich gewesen, feste Orte vorzuschreiben oder allgemeine Konzepte zu beschreiben. Diese Punkte hätten aber ähnlich wie die Designentscheidungen bei den Architekturmustern Einfluss auf die gesamte Architektur und würden dazu führen, dass die Gültigkeit beschränkter wäre. Weiter müssten auch neue Module eingeführt werden, wenn bspw. dedizierte Formen von Data Lakes eingeführt werden sollten.

Aufteilung der Module

Die Aufteilung der Module wurde im Laufe des Entwicklungsprozesses iterativ entwickelt. In vielen Schritten, insbesondere bei der Aggregation, waren mehrere Möglichkeiten der Aufteilung und des Schnitts der Module möglich.

Das Modul M.2. Dokumentenmanagement ist aktuell für die Ablage und Speicherung von Dokumenten und ähnlichem zuständig. Es wäre genauso denkbar, dass dieses Modul gestrichen wird und die einzelnen Module jeweils ihre eigenen Daten verwalten. Das würde beispielsweise zu einer deutlich geringeren Kopplung führen. Nachteil bei dieser Entscheidung wäre jedoch,

dass bestimmte Implementierungen bzw. Funktionalitäten bei vielen Modulen vorkommen würden, womit es zu Code-Duplizierungen kommen würde.

Ein weiterer Punkt betrifft die Unterteilung von M.18. Fehlermeldung und -darstellung sowie M.17. Fehlerdatenbank. Hier könnten fehlerrelevante Funktionen auch gemeinsam in ein Modul hinzugefügt werden. Eine Trennung von Fehlerdatenbank und Fehlermeldung könnte aus logischer Sicht konstruiert wirken. Technisch gesehen wäre diese Aufteilung jedoch sinnvoll, da verschiedene Daten separat gehalten werden.

Weiter könnten andere Entscheidungen bzgl. der aktuellen Rolle des Moduls M.9. Auftragsverwaltung/Priorisierung getroffen werden. Dieses Modul besitzt aktuell eine sehr dominante Rolle in der Referenzarchitektur, da es mit den meisten anderen Modulen in Verbindung steht. Dies könnte bei der Implementierung zu Performanceschwierigkeiten und zu einem eventuell zu dominanten Modul der Architektur führen. Jedoch ist ein solches Modul, welches die relevanten Daten bei Bedarf aggregiert und Metadaten auswerten kann, von großer Notwendigkeit. Die Art der Implementierung ist insbesondere in Hinblick auf ereignisgesteuerten Architekturen und die Einführung eines Nachrichtenbusses von hoher Bedeutung, um die Nachteile einer solchen Designentscheidungen zu umgehen und die Vorteile zu maximieren. Zusätzlich wäre es bei diesem Modul denkbar, die Priorisierung der Aufträge bzw. die zuständigen Dienste hierfür in ein separates Modul auszulagern. Dies hätte den Vorteil, dass diese Dienste bspw. in die Cloud ausgelagert werden könnten, da die Priorisierung meist sehr rechenintensiv ist.

Viele weitere Aspekte bei der Modulunterteilung sind ebenso diskussionswürdig, so sind manche Module sehr detailliert ausgearbeitet oder gar technisch relativ konkret, da sie eine hohe Relevanz besitzen. Andere Module hingegen sind aufgrund tendenziell geringerer Relevanz nicht in der gleichen Konkretheit beschrieben. Hier könnten alle Module auf eine einheitliche Basis gesetzt werden, welche jedoch die Lesbarkeit reduzieren und eine Gewichtung der relevanten Aspekte nicht genauso ermöglichen würde.

Beim Modulaufbau ist es verbunden mit dem Schnitt prinzipiell denkbar, auch die Module von den Anforderungen zu lösen bzw. diese aufzuteilen. Eine der Grundbedingungen der RAI war es, die Anforderungen jeweils genau einem Modul zuzuordnen. Diese Anforderung führt unweigerlich zu Designentscheidungen, bei denen manche Module sehr umfassend sind. Abstrakte Anforderungen bzw. stark aggregierte Anforderungen führen daher in manchen Fällen ebenso zu stark aggregierten und komplexeren Modulen. Zwar wurde im Laufe der Anforderungserhebung und der Entwicklung der RAI dieser Aspekt reduziert, da einige Anforderungen in der Literatur und bei den Praxisbefragungen unterteilt werden konnten, das Grundproblem besteht jedoch weiterhin. Eine Unterteilung in viel kleinere Module wäre ohne die Anforderung also möglich gewesen.

Anderer Schnitt von generischen Modulen

Die Aufteilung zwischen generischen und optionalen Modulen ist darüber hinaus ein Aspekt, der im Nachhinein auch anders entschieden werden kann. Generische Module sind im Wesentlichen die Module, die mindestens eine hochpriorisierte generische Anforderung beinhalten. Nur zwei Module wurden als optional eingestuft, obwohl sie eine generische Anforderung besitzen. Das Modul M.35. Maschinenspezifika und M.19. Predictive Maintenance. Beide Module können aufgrund technischer Anforderungen und Diskussionen als zusätzliche Aufbaustufe eines integrierten Instandhaltungssystems angesehen werden.

Zwangsläufig notwendig sind sie jedoch nicht, trotz der hohen Relevanz der Anforderungen. So konnte neben den technischen Aspekten auch im Gespräch mit den beteiligten Stakeholdern klar erkannt werden, dass diese Eigenschaften zwar bei neueren Systemen angestrebt werden, jedoch eine erweiterte Ausbaustufe darstellen.

Ähnliche Punkte betreffen die folgenden Module. Sie könnten als optional deklariert werden, sofern kein mobiles Endgerät benutzt werden sollte bzw. dies nur darstellenden Charakter hätte. Also keine Informationen aufnehmen müsste.

- M.33. Funktionalitäten mobiles Endgerät
- M.29. QR-CODE auslesen
- M.26. Kommunikation

Zusätzlich kann auch das Modul M.3. Wiki als optional klassifiziert werden, da in den meisten Fällen ein Wikisystem bereits in den Unternehmen besteht und nicht als essentiell notwendig für die Instandhaltung erscheint, sofern die anderen Module bereits implementiert wurden. Ähnlich gelagert ist die Entscheidung bei dem Modul M.12. Auftragsmanagement in Verbindung mit ERP. Zwar ist es bei den meisten Unternehmen selbstverständlich, ein ERP zu besitzen und diese Informationen sollten zwangsläufig mit dem Instandhaltungssystem abgeglichen werden, jedoch existieren hier sicherlich auch andere Ansätze. Hierdurch wäre auch das Modul optional.

Sichten der Referenzarchitektur

Neben den Aspekten der Abstraktion und der Verständlichkeit, die im vorherigen Abschnitt diskutiert wurden, können wie in Kapitel 5 angeschnitten, die Sichten je nach Einsatzzweck angepasst werden. Die Begründung für die Wahl der Darstellung ist dort nachzulesen. Jedoch ist die Aufteilung und die Beschränkung auf die in der RAII enthaltenen Sichten nicht notwendig. So können andere Sichten erstellt werden, um bestimmte Designentscheidungen hervorzuheben und bspw. Entwicklungen zu unterstützen. Denkbar wäre eine zusätzliche Sicht für Rollenkonzepte und Verteilung von Funktionalitäten auf bestimmte Rollen. Ebenso könnten grafische Elemente oder Datenverteilungen genauer angesprochen und in neuen Sichten zusammengefasst werden. Neben dem Hinzufügen von Sichten wäre es ebenso denkbar, eine komplett andere Darstellungsform zu wählen. Schichtenmodelle oder explizite Codeteile können bei konkreten Referenzarchitekturen benutzt werden oder die Darstellung könnte in Abhängigkeit der Klassifikation aus Kapitel 2 geändert werden.

Begrenzung auf bestimmte Prozesse

Ein weiterer Punkt betrifft die Darstellung der Prozesse in Kapitel 6.4. Die Reduzierung auf elf Aktivitäten dient dazu, eine möglichst geringe Zahl an besonders wichtigen Aktivitäten zu haben. Weiter dienen die meisten Aktivitäten dazu, die Grundaktivität zu belegen und zu verfeinern. Die Reduzierung auf diese elf Aktivitäten wird nur durch die angestrebte Komplexitätsreduzierung und Fokussierung auf einen mobilen Instandhaltungsfall begründet. Es wäre daher durchaus möglich, noch eine beliebig hohe Anzahl an Prozessen in die Referenzarchitektur zu integrieren. Für die Entwicklung könnte dies einen Mehrwert besitzen, jedoch bestünde die Gefahr, die RAII damit zu stark zu überfrachten und die Übersichtlichkeit enorm einzuschränken.

Webfokussierung/Appfokussierung/Mobiles Endgerät

Eine Designentscheidung bzw. eine Restriktion wurde bei dem Aufbau des integrierten Instandhaltungssystems getroffen: Als ein zusätzliches Endgerät wurde von einem mobilen Endgerät als Entität ausgegangen. Weder sind alle Instandhaltungssysteme für eine mobile Nutzung ausgelegt, noch werden mobile Endgeräte standardmäßig verwendet. Es wurde sich dennoch für die Integration für mobilen Endgeräten in der RAII entschieden, da die mobile Nutzung von Instandhaltungssystemen gerade für den Instandhalter enorme Potentiale besitzt. Sei es, weil sich die Produktionsanlagen an externen Orten befinden, oder weil der Instandhalter Informationen aus dem Instandhaltungssystem benötigt, während er seine Tätigkeiten an einer Anlage durchführt. Weiter kann aus der aktuellen Entwicklung von Informationssystemen zweifellos von einer zunehmenden Wichtigkeit von mobilen Endgeräten und insbesondere der Optimierung von Systemen für mobile Endgeräten ausgegangen werden. Um alle Instandhaltungssysteme abzudecken, ist die Entscheidung jedoch prinzipiell diskussionswürdig.

Weiter behandelt die RAII Darstellungsarten geringfügig. Es wäre daher möglich gewesen, die RAII in verschiedene Teile wie eine Weboberfläche, eine mobile App und weitere Darstellungsarten zu unterteilen. Da die Entwicklung von Weboberflächen, Desktopanwendungen und Apps sich immer mehr annähern, wurde darauf verzichtet. Auch sollten keine Arten der Darstellung vorweggenommen werden, da diese insbesondere der technischen Entwicklung und allgemeinen Trends unterworfen ist.

In dieser Hinsicht ist auch die Beschränkung auf exakt drei Entitäten durchaus diskussionswürdig. Es wäre denkbar, mehr oder weniger Entitäten festzulegen. Zum Beispiel könnte zwischen verschiedenen mobile Endgeräten (Smartphone, Tablet, Augmented Reality fähige Brillen, Wearables im Allgemeinen) unterschieden werden. Es erschien für den Einsatzzweck der RAII und die Verallgemeinerbarkeit von Konzepten auf mobile Endgeräte sinnvoll, sich auf diese zu beschränken und nicht weiter zu unterteilen.

Die genannten Aspekte und Diskussionspunkte sind im Laufe der Erstellung und bei der Evaluation hervorgetreten. Viele der getroffenen Designentscheidungen können je nach Ausrichtung und je nach anstrebten Zweck einer Referenzarchitektur anders getroffen werden. Dieses Kapitel dient daher zur Reflektion der getroffenen Entscheidungen und auch als Möglichkeit, bei der Ableitung von neuen Referenzarchitekturen oder Implementierungen die Designentscheidungen im eigenen Sinne zu treffen.

6.7 Zusammenfassung

In diesem Kapitel wurde der Inhalt der vier Sichten der RAII zusammengefasst präsentiert. Zuerst wurde die Funktionale Sicht, die alle logischen Einheiten des IIS in Module unterteilt und deren Funktionsweise illustriert. Die Verteilung der einzelnen Module wird in der Verteilungssicht dargestellt, sodass klar ersichtlich ist, aus welcher Entität sich welches Modul befinden muss oder kann, damit der Instandhalter ideal unterstützt wird. Aufbauend auf den anderen Sichten wurde die Prozesssicht dargestellt. Diese enthält eine an das UML-Aktivitätsdiagramm angelehnte Modellierung der einzelnen Use Case Kategorien sowie den daraus extrahierten Referenzprozess. Dieser stellt den Kernprozess des Systems dar und verbindet einzelne Aktivitäten mitsamt den jeweiligen Modulen, den benutzten Geräten und den Use Cases. Diese Sicht stellt also eine umspannende Sicht dar, weil sie auf den Informationen aller anderen Sichten basiert, diese einfach ersichtlich in Kontext setzt und als Vorlage für die Entwicklung benutzt werden kann. Anschließend wurde die Use Case Sicht erläutert, die die wichtigsten Use Cases des zu entwerfenden Systems aus den vier untersuchten Unternehmen darstellt. Diese werden komprimiert dargestellt und in verschiedene Kategorien unterteilt. Anhand der einzelnen Sichten ist es möglich, sich Komponenten für das eigene System auszuwählen und die Informationen als Grundlage zur Entwicklung zu benutzen.

Weiter wurde inhaltlich abschließend die RAII und ihre Designentscheidungen, die während der Entwicklung der RAII getroffen wurden, kritisch gewürdigt und ausgiebig diskutiert. Viele dieser Designentscheidungen können je nach Ausrichtung einer Referenzarchitektur anders getroffen werden. Die Entscheidungen würden jedoch teilweise den Kern der Referenzarchitektur umgestalten und zur Folge haben, dass sich bestimmte Eigenschaften, wie z. B. Verständlichkeit, Umfang, Technologieunabhängigkeit verändert werden würden. Die einzelnen Aspekte und ihre Folgen wurden daher anhand verschiedener Designentscheidungen exemplarisch vorgestellt. Auf Basis dieser Informationen können die Entscheidungen besser nachvollzogen werden und bei Bedarf aufbauend auf diesen Punkte Veränderungen an der Referenzarchitektur herbeigeführt werden.

Das Kapitel umfasst demnach die RAII und eine kritische Würdigung der jeweiligen Entscheidungen, die zu der hier vorgestellten Ausprägung geführt haben.

7 Evaluation der Referenzarchitektur RAI

7.1 Einleitung

March und Smith (1995, S. 254) bezeichnen die Evaluation als zweites Element der gestaltungsorientierten Forschung, die nach der Entwicklung des Artefakts zur systematischen Bewertung der entwickelten Artefakte durchgeführt wird. In dem in Kapitel 1 vorgestellten Forschungsdesign wird die Evaluation durch die Evaluationsphase abgedeckt. Ziel der Evaluation ist es, das Artefakt dahingehend zu bewerten, ob es imstande ist, die zugrundeliegende Problemstellung zu adressieren (Riege et al., 2011, S. 73). Im vorliegenden Fall soll überprüft werden, inwiefern die Referenzarchitektur den Entwicklungsprozess für ein Instandhaltungssystem unterstützt. Die Beantwortung dieser Frage wird anhand drei Evaluationsmethoden durchgeführt, welche einen umfassenden Blick auf die Nützlichkeit des Artefakts ermöglichen. Die Ergebnisse und das Vorgehen werden in diesem Kapitel vorgestellt.

In Kapitel 7.2 werden hierfür die Ziele, das Vorgehen sowie sämtliche Kriterien der Evaluation vorgestellt. In Kapitel 7.3 werden jene Anforderungen an die RAI vorgestellt, durch deren Erfüllung eine Nützlichkeit nachgewiesen werden kann. Diese Anforderungen werden logisch, aus der Literatur und aus praktischen Erfahrungen während der Arbeit mit der Referenzarchitektur abgeleitet und dienen als Grundlage für die in Kapitel 7.4 vorgestellte analytisch merkmalsorientierte Evaluation. Die Anforderungen stellen dabei Merkmale dar, nach denen die Nützlichkeit analytisch untersucht wird. In Kapitel 7.5 wird die Evaluation anhand des Anwendungsfalls „Ressourcen-Cockpit“ dargestellt. Dieser Anwendungsfall umfasst zwei verschiedene Arten von Evaluationen: Eine prototypische Umsetzung der RAI in einen prototypischen Demonstrator, der als „Ressourcen-Cockpit“ bezeichnet wird und eine Evaluation anhand Experteninterviews, bei denen die Experten die RAI hinsichtlich ihrer Nützlichkeit bei der Entwicklungsunterstützung bewerten. Im Anschluss werden in Kapitel 7.6 die Ergebnisse zusammengefasst.

7.2 Ziele und Vorgehen der Evaluation

Im Rahmen der gestaltungsorientierten Forschung soll das Artefakt bzgl. der Qualität, Effizienz und Nützlichkeit mit nachvollziehbaren Evaluationsmethoden bewertet werden (Hevner/Chatterjee, 2010, S. 12). Im Rahmen dieser Arbeit stellt das Artefakt, das mit diesen Methoden evaluiert werden kann, im Sinne von Hevner et al. (2004) die Darstellung der Modellierung (siehe Kapitel 5) und die inhaltliche Architektur an sich dar (siehe Kapitel 6). Dieses Artefakt gilt es zu gestalten und dahingehend zu evaluieren, ob es das übergreifende Ziel erfüllt, die Entwicklung von IIS und den damit verbundenen Entwicklungsprozess im Speziellen zu erleichtern.

Eine systematische Evaluation erfordert nach Frank (2000, S. 40) zunächst die Definition von Anforderungen, die das Artefakt erfüllen soll. Diese Anforderungen müssen dabei so ausgewählt werden, dass durch ihre Erfüllung von einer positiven Auswirkung auf das Erreichen des übergeordneten Gesamtziels ausgegangen werden kann. Anhand dieser Anforderungen soll die Evaluation nach wissenschaftlich nachvollziehbaren Methoden erfolgen. Hevner et al. (2004, S. 86) präsentieren die in Tabelle 33 dargestellten Methoden der Artefaktevaluation in der gestaltungsorientierten Forschung und kategorisieren sie nach beobachtenden, analytischen, experimentellen, testenden und beschreibenden Methoden.

Art	Methode	Beschreibung
Beobachtend	Fallstudie	Untersuchung des Artefakts im jeweiligen Geschäftsumfeld
	Feldstudie	Beobachtung der Nutzung des Artefakts in Projekten
Analytisch	Statische Analyse	Untersuchung der Struktur des Artefakts
	Architekturanalyse	Untersuchung der Integrationsfähigkeit des Artefakts in die technische Infrastruktur
	Optimierung	Demonstration der Optimalität des Artefakts für den Zweck oder zum Aufzeigen der Grenzen der Optimierung
	Dynamische Analyse	Untersuchung des Laufzeitverhaltens des Artefakts
Experimentell	Kontrolliertes Experiment	Untersuchung von Artefakteigenschaften unter kontrollierten Bedingungen
	Simulation	Ausführen des Artefakts unter Nutzung nicht-realer Daten
Testend	Funktionale Tests	Verwenden der Schnittstellen des Artefakts, um Fehler zu finden (Black Box Tests)
	Strukturelle Tests	Testen der inneren Funktionsweise des Artefakts um Fehler zu finden (White Box Tests)
Beschreibend	Expertenwissen/Informiertes Argument	Begründen der Nützlichkeit des Artefakts durch überzeugende Argumentation auf Basis von Informationen aus der Wissensbasis
	Szenarios	Erstellung von Verwendungsszenarios, um die Nützlichkeit des Artefakts zu zeigen

Tabelle 33: Kategorisierte Übersicht von Evaluationsmethoden

Quelle: In Anlehnung an Hevner et al. (2004, S. 86)

Um eine Evaluation des Artefakts aus unterschiedlichen Perspektiven zu ermöglichen wird diese mit einem multimethodalen Ansatz durchgeführt. Angelehnt an die Einordnung der

Methoden aus Tabelle 33 können die später vorgestellten Evaluationsmethoden folgenden Kategorien zugeordnet werden: Zum einen werden beschreibende Methoden in Form von Szenarios verwendet, um durch deren Implementierung die Nützlichkeit aufzuzeigen. Zum anderen werden beschreibenden Methoden in Verbindung mit der analytischen statischen Architekturanalyse genutzt, um per Expertenwissen logische Schlüsse hinsichtlich der Nützlichkeit des Artefakts zu ziehen. Weiter hat die Anwendung der RAI bei der Entwicklung des Ressourcen-Cockpits und der anschließenden Bewertung anhand von Experteninterviews nach der Kategorisierung von Hevner et al. (2004, S. 86) Fallstudiencharakter. Insgesamt kann die Evaluation der Referenzarchitektur dreigeteilt werden und wird in Abbildung 33 anhand der verwendeten Methodik dargestellt.

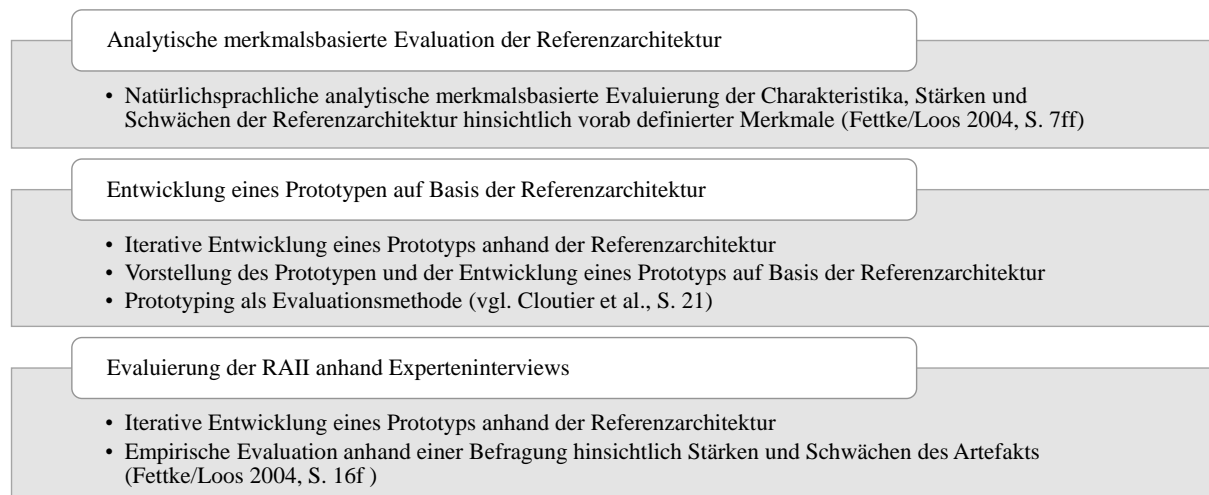


Abbildung 33: Unterteilung des Evaluationsvorgehen in drei Abschnitte

Quelle: Eigene Darstellung

Der erste Teil der Evaluation baut auf Evaluationsmethoden für Referenzmodelle auf. Fettke und Loos (2004a, S. 7ff) unterscheiden in ihrer Übersicht analytischer Evaluationsmethoden für Referenzmodelle zwischen einer natürlichsprachlichen und einer merkmalsbasierten Evaluation. Bei der natürlichsprachlichen Evaluation werden Charakteristika, Stärken und Schwächen eines Referenzmodells rein verbal diskutiert (Fettke/Loos, 2003, S. 83). Bei der merkmalsbasierten Evaluation erfolgt die Evaluation anhand von ad-hoc definierten Merkmalen. In der Literatur werden diese beiden Ansätze oft kombiniert (Fettke/Loos, 2004a, S. 8). Diesem Ansatz folgend und angepasst auf die Charakteristik einer Referenzarchitektur erfolgt eine analytische natürlichsprachliche merkmalsbasierte Evaluation der RAI. Dabei wird die Erfüllung bzw. Nichterfüllung der jeweiligen Merkmale der Referenzarchitektur argumentativ in natürlicher Sprache diskutiert. Im weiteren Verlauf wird die Kombination abkürzend als analytische merkmalsbasierte Evaluation bezeichnet, da merkmalsbasierte Evaluationen oft in Verbindung mit natürlichsprachlichen Evaluationen erfolgen bzw. diese oft beinhalten.

Nach Cloutier et al. (2009, S. 21) stellt eine iterative Entwicklung anhand von Prototyping einen geeigneten Ansatz dar, um innovative Referenzarchitekturen zu validieren und damit zu evaluieren. Da die RAI in weiten Teilen als forschungsgetrieben klassifiziert wurde (siehe 2.2.4.4), stellt sie somit eine innovative Referenzarchitektur im Sinne von Cloutier et al. (2009, S. 21) dar. Aus diesem Grund bildet das zweite Element der Evaluation der RAI die Implementierung eines Prototyps anhand der Referenzarchitektur.

Im dritten Teil werden an der Entwicklung und Anforderungsaufnahme beteiligte Experten hinsichtlich der Nützlichkeit der Referenzarchitektur befragt und bewerten die Entwicklung des Prototyps auf Basis der Referenzarchitektur. Diese Beurteilung findet im Rahmen eines semi-strukturierten Leitfadeninterview statt. Die Vorgehensweise entspricht der empirischen Evaluationsmethode der Befragung nach Fettke und Loos (2004a, S. 16f).

7.3 Anforderungen an die RAI

Die Merkmale, an denen sich Teile der Evaluation ausrichten, sind in dieser Arbeit Anforderungen, die an die RAI und ihre Modellierung gestellt werden. Durch deren Erfüllung wird das übergeordnete Ziel, den Entwicklungsprozess zu erleichtern, von der RAI erfüllt. Für die Referenzarchitektur eines IIS im Kontext von Trends wie Industrie 4.0, CPS und der Digitalisierung existieren Anforderungen, die aus verschiedenen Domänen und Themengebieten resultieren:

- **Referenzarchitektur:** Es wären insbesondere Anforderungen zu nennen, die aus dem Artefakt einer Referenzarchitektur und deren Klassifikation resultieren. Referenzarchitekturen sind eine besondere Art von Softwarearchitekturen, die besondere Anforderungen hinsichtlich Darstellung, Allgemeingültigkeit und Anpassbarkeit besitzen, da die Erstellung einer Referenzarchitektur meist aus bestimmten Gründen und mit festgelegten Zielen angestrebt wird. Diese Anforderungen, die aus der in Kapitel 2 vorgestellten Klassifikation resultieren, betreffen bspw. Anforderungen hinsichtlich besonders leichter Verständlichkeit (Muller, 2007, S. 8).
- **Industrie 4.0/Digitalisierung:** Die RAI als Blaupause einer Applikation für die Industrie 4.0 bzw. Digitalisierung besitzt in diesem Kontext spezielle Anforderungen, die sich von einer Standardapplikation unterscheiden. Industrie 4.0 und die Digitalisierung verlangen insbesondere eine fachbereichsübergreifende Kommunikation und Entwicklung sowie neuartiges Denken. Eine Förderung der domänenübergreifenden Entwicklung und Aufnahmegabe durch die RAI können daher als Anforderungen genannt werden. Weiterhin erfordert die zunehmende Dezentralität von neuartigen Systemen eine neuartige Form der Architektur und Darstellung.
- **Instandhaltung:** Die Instandhaltung als Einsatzort des IIS umfasst weitere spezifische Anforderungen. Diese resultieren insbesondere aus dem für die Entwicklung benötigten Domänenwissen der Instandhaltung und dem Bedarf, nahezu mit jedem Informationssystem eines produzierenden Unternehmens zusammenzuarbeiten. Weiter ergeben sich aus der oftmals unternehmensübergreifenden Zusammenarbeit bei der Instandhaltung auch Anforderungen, die eine unternehmensübergreifende Referenzarchitektur erfordern.

Die Anforderungen der Themenbereiche überschneiden sich teilweise sehr deutlich und können daher nicht immer klar voneinander abgegrenzt werden. Aufbauend auf diesen Themengebieten lassen sich viele Anforderungen an das beteiligte Anforderungsmanagement während der gesamten Lebensdauer des Entwicklungsprojektes ableiten. Diese Anforderungen spielen bei der RAI ebenso eine große Rolle. Zum einen, da die RAI die Entwicklung auch bei Aufnahmegabe und -management unterstützen soll. Zum anderen ist ein IIS oftmals ein Grundbestandteil eines PSS, wenn dieses insbesondere die Instandhaltung als elementaren Teil des PSS bzw. des Geschäftsmodells begreift. Viele dieser Herausforderungen und der daraus resultierenden Anforderungen sind von Berkovich (2011, S. 129) im Kontext von PSS Anforderungsmanagement Ansätzen identifiziert worden und können auf die RAI übertragen werden:

- Die Kommunikation zwischen den an der Entwicklung beteiligten Personen und Unternehmensbereichen spielt eine große Rolle bei der Entwicklung von PSS. Die Unternehmensbereiche und beteiligten Personen können unterschiedliche Vorstellungen haben und unterschiedliches Hintergrundwissen besitzen, sodass Anforderungen unterschiedlich interpretiert werden. Das Anforderungsmanagement für ein PSS muss daher die domänenübergreifende Zusammenarbeit fördern (Berkovich et al., 2010, S. 4). Im Kontext der Referenzarchitektur kann diese Herausforderung noch intensiviert werden, da alle an der Instandhaltung und Produktion (bspw. Instandhalter, Produktionsverantwortliche, IT-Verantwortliche) Beteiligten noch hinzugezogen werden müssen, weil in viele bestehende Prozesse und Systeme eingegriffen werden muss.
- Weiter stellt die unterschiedliche Granularität von Anforderungen eine große Herausforderung dar. Stakeholder drücken ihre Anforderungen üblicherweise auf unterschiedlichen Abstraktionsniveaus aus. Aufgabe des Anforderungsmanagements ist es daher, die einzelnen sehr konkreten Anforderungen mit den allgemeinen Anforderungen zu verbinden. Diese unterschiedlichen Arten von Anforderungen müssen sich auch in der Referenzarchitektur wiederfinden. Ebenso muss eine grundlegende Basis an Anforderungen definiert werden, die domänenübergreifend verständlich ist. Logische Verbindungen der Anforderungen untereinander und der Anforderung mit der Architekturlösung der Referenzarchitektur sollten weiterhin aufgezeigt werden, um eine Verständlichkeit und Rückverfolgbarkeit der Anforderungen während des gesamten Entwicklungsprozesses zu erleichtern.
- Weiterhin existiert oftmals eine konzeptionelle Lücke zwischen Anforderungen und der Entwicklung, sodass Anforderungen aus Kunden- bzw. Benutzersicht erst in die Sprache der Entwickler übersetzt und auf Korrektheit überprüft werden müssen (Adler, 1994, S. 18). Dieser Aspekt gewinnt insbesondere bei PSS an Bedeutung, da durch die Komplexität eines PSS keine Spezifikation ohne Entwicklungsentscheidung getroffen werden kann (Berkovich, 2011, S. 129).

Ursprünglich aus dem Bereich des reinen Anforderungsmanagements eines PSS, müssen diese Herausforderungen auch für eine Referenzarchitektur und insbesondere die RAI, welche im besonderen Maße Anforderungen inkludiert, adressiert werden.

Weiter sollte eine Referenzarchitektur entsprechend der in Kapitel 2 vorgestellten Definition die Kommunikation fördern. Ein Artefakt im Allgemeinen ermöglicht unter Beachtung bestimmter Rahmenbedingungen, dass über das Artefakt mit anderen Personen kommuniziert werden kann (Dix, 1994). Ein Artefakt, das von mehreren Beteiligten genutzt wird und über visuelle Komponenten verfügt, dient dabei als Mittel zur Kommunikation und kann diese dadurch fördern (Ramduny-Ellis et al., 2005, S. 23f). Solche kommunikationsfördernden Aspekte resultieren ebenso in Anforderungen an RAI. Diese betreffen vor allem die Bereitstellung von Domänenwissen und eine adäquate Darstellung zur Förderung von Anforderungsdiskussionen und -management sowie Diskussionen über die Entwicklung an sich.

Nachfolgend werden auf Basis der genannten Hintergründe und Herausforderungen 14 Anforderungen definiert. Diese sind hinsichtlich Anforderungskomplexen gruppiert und befinden sich teilweise auf unterschiedlicher Aggregationsebene. Dies hat zur Folge, dass bestimmte Anforderungen Teile anderer Anforderungen darstellen, da diese nur einen

Teilaspekt deutlich konkreter abdecken. Diese Anforderungen und deren Zusammenhang werden nachfolgend vorgestellt.

Anforderung 1: Domänenübergreifende Kooperation

Die RAII soll die domänenübergreifende Kooperation zwischen allen an der Entwicklung beteiligten Stakeholdern ermöglichen. Ein Kriterium für den Erfolg einer Referenzarchitektur besteht darin, dass diese von möglichst vielen Stakeholdern verstanden wird (Guessi et al., 2014, S. 2). Ein gegenseitiges Verständnis aller Beteiligten des Systems, dessen Anforderungen und der Entwicklung ist von Beginn an meist nicht gegeben. Dies ist jedoch von entscheidender Bedeutung für das Gelingen eines solchen Entwicklungsprozesses, da typischerweise Personen aus den Bereichen IT, Produktion und Instandhaltung sowie deren Wissen notwendig ist. Weiterhin sind die Anforderungen hinsichtlich der Architektur und des Systems bei jeder Anwendungsgruppe unterschiedlich und werden auf andere Weise ausgedrückt. Durch eine gezielte Diskussion anhand der RAII wird vermieden, dass nur eine kleine Gruppe an der Entwicklung partizipieren kann und sich so eine isolierte Sicht auf das System durchsetzen könnte. Die Elemente dieses Artefakts sollten daher als grundlegende Verständnisbasis zur Kommunikation fungieren. Diese Anforderung ist ebenso eine Kernanforderung für das Anforderungsmanagement für PSS von Berkovich et al. (2009, S. 2579) Die Anforderung 1 kann durch die Anforderung 2,3 und 4 unterstützt werden und umfasst diese implizit.

Daher lässt sich folgende Anforderung an die RAII formulieren: **Die RAII muss eine Kooperation aller an der Entwicklung des Instandhaltungssystems beteiligten Personen ermöglichen und fördern, indem ein domänenübergreifendes Verständnis der Elemente des Systems, dessen Komponenten sowie den verbundenen Anforderungen über den Entwicklungszyklus hinweg möglich ist.**

Anforderung 2: Bereitstellung von Domänenwissen

Ein weiterer Aspekt, der mit eng mit Anforderung 1 verbunden ist, jedoch einen eher inhaltlichen Fokus besitzt, ist die Bereitstellung von Domänenwissen durch die RAII. Referenzarchitekturen sind nach Muller und van de Laar (2009, S. 3) reich an technischen und kontextabhängigen Domäneninformationen. Hierdurch soll für alle Beteiligten eine möglichst einheitliche Wissensbasis geschaffen werden. Vielen Entwicklern sind die Anforderungen und Gegebenheiten der Instandhaltung sowie der Produktion, insbesondere die involvierten Systeme und Prozesse, nicht bekannt. Weiterhin fehlt es Personen aus der Instandhaltung und der Produktion oft an Domänenwissen aus der IT und den daraus resultierenden Anforderungen und Möglichkeiten. Durch die Bereitstellung eines einheitlichen Domänenwissens wird die Grundlage für ein einheitliches Verständnis des Systems geschaffen. Darüber hinaus soll durch die Referenzarchitektur Domänenwissen im Hinblick auf das System und dessen Komponenten an sich zur Verfügung gestellt werden, sodass der Einstieg in die Domäne und die Entwicklung an sich beschleunigt werden.

Daher lässt sich die Anforderung 2 an die RAII wie folgt formulieren: **Die RAII soll Domänenwissen über die Instandhaltung, Instandhaltungssysteme und die zugrundeliegende Referenzarchitektur mitsamt generischer sowie optionaler Komponenten bereitstellen.**

Anforderung 3: Bereitstellung einheitlicher Domänensprache

Eine einheitliche Domänensprache zwischen allen Beteiligten ist notwendig, um Missverständnisse und damit verbundene Fehlinterpretationen über Anforderungen und Eigenschaften des Systems zu reduzieren. Daher ist der Aufbau einer einheitlichen Begriffswelt unerlässlich, da diese in unterschiedlichen Domänen sehr verschieden sein kann (Berkovich et al., 2009, S. 2579). Zusammen mit Anforderung 1 und 2 bildet diese Anforderung einen Themenkomplex, der eine eindeutigere Zusammenarbeit zwischen den Beteiligten fördern soll, wozu die Referenzarchitektur als Ausgangsbasis dient.

Aus diesem Grund kann folgende Anforderung an die RAII gestellt werden: **Die RAII muss eine einheitliche Domänensprache über das zu entwickelnde System und die dazugehörige Domäne für alle Beteiligten der Entwicklung schaffen.**

Anforderung 4: Leichte Verständlichkeit

Viele der genannten Anforderungen fördern oder fordern die domänenübergreifende Zusammenarbeit in größeren Teams. Ein wichtiger Aspekt dieser Zusammenarbeit besteht darin, die zur Kommunikation und Entwicklung in diesem Kontext zu benutzenden Mittel nur dann akzeptiert und eingesetzt werden, wenn die Anwendung als hilfreich und einfach zu benutzen angesehen wird. Weiter ist es eine essentielle Notwendigkeit in Softwareprojekten, dass das gesamte System oder die Domäne von den erstellenden Akteuren verstanden wird (Batory et al., 1995, S. 35). Im Falle domänenübergreifender Projekte betrifft dies nicht nur Entwickler mit IT-Hintergrund. Daher muss eine Referenzarchitektur für eine große Anzahl an heterogenen Stakeholdern verständlich sein (Muller, 2007, S. 8). Referenzarchitekturen benutzen dazu die Abstraktion als Stilmittel, um ein umfassendes, für viele Stakeholder verständliches Bild einer Architektur ohne unnötige Details zu vermitteln. Als Nebenbedingung müssen Referenzarchitekturen daher hinsichtlich ihrer Größe limitiert sein, um nützlich und wartbar zu bleiben (Muller/van de Laar, 2009, S. 7). Aus diesen Gründen muss die RAII eine einfache Verständlichkeit besitzen und kein großes Hintergrundwissen erfordern, um mit ihr Arbeiten zu können.

Daher lautet die Anforderung wie folgt: **Die RAII soll sich durch eine leichte Verständlichkeit der Darstellung auszeichnen, sodass das Beherrschen von komplexen Notationen und Hintergrundwissen keine Notwendigkeit darstellt, um mit ihr zu arbeiten.**

Anforderung 5: Abbildung des generischen Kerns

Referenzarchitekturen werden oft dazu benutzt, bestehende Architekturen zu vergleichen und Gemeinsamkeiten aufzuzeigen, um daraus einen generischen Kern bzw. den Kern einer Domäne bzw. einer Applikation abzuleiten. Im Falle der RAII soll der Kern eines IIS dargelegt werden, der bei unterschiedlichen Anwendungsfällen bestehen bleibt. Dazu muss zuerst gezeigt werden, dass ein einheitlicher Kern über verschiedene Anwendungsfälle hinweg existiert und welche Elemente zu ihm gehören. Durch die Benutzung eines einheitlichen Kerns wird die Basis gelegt, individuelle Applikationen so zu entwickeln, dass eine gemeinsame Basis wiederverwendet wird, ohne das explizit Standards eingeführt werden müssen. Hierdurch können Softwareentwickler leichter unterschiedliche Lösungen mit wiederverwendbaren Komponenten entwickeln oder bestehende Lösungen lassen sich leicht an individuellen Gegebenheiten anpassen.

Daher lautet die Anforderung: **Die RAII soll den generischen Kern eines IIS über unterschiedliche Instandhaltungsszenarien und Fokusse hinweg darstellen, sodass bei unterschiedlichen mit der RAII entwickelten IISs eine einheitliche Basis vorhanden ist.**

Anforderung 6: Fokussierung auf Standardprozesse

Ein besonderer Aspekt der Fokussierung auf dem generischen Kern einer Applikation ist, dass Standardprozesse bzw. Referenzprozesse identifiziert werden. Diese Referenzprozesse sollten, wenn möglich bei jeder Applikation im Kern identisch sein. Durch die Identifikation solcher Prozesse lassen sich Funktionsweisen von Applikationen leicht verständlich und klar ersichtlich erläutern. Anhand dieser Prozesse können Standardkomponenten identifiziert werden. Darüber hinaus ermöglichen Referenzprozesse Unternehmen die Entwicklung und spätere Nutzung des zu entwickelnden Systems an diesen Prozessen auszurichten. Die Referenzprozesse sind angelehnt an die generischen Szenarios, die beim Vorgehen nach Anastasopoulos et al. (2000) und Bayer et al. (2004) zur Erstellung von Referenzarchitekturen benutzt werden. Die Anforderung, sich auf solche Referenzprozesse zu fokussieren, stellt dabei eine Teilerfüllung der Anforderung einen generischen Kern in der RAII zu haben dar.

Daher: **Die RAII sollte Standardprozesse eines Instandhaltungssystems über unterschiedliche Instandhaltungsszenarien und Fokusse hinweg darstellen.**

Anforderung 7: Anpassbarkeit auf optionale Fälle

Eine Referenzarchitektur muss nach Vogel et al. (2009, S. 257) an konkrete Bedürfnisse angepasst werden, sodass diese ggfs. schrittweise erweitert werden kann, wenn neue Anforderungen entstehen. Neben dem Fall der Erweiterung soll die Referenzarchitektur von Beginn an darauf ausgelegt sein, sich an spezifische Fälle anzupassen und diese von Beginn an aufzuzeigen. Diese Anpassung sollte durch Variationspunkte direkt in die Referenzarchitektur integriert werden, sodass die optionalen Fälle bei der Ableitung direkt ausgewählt werden können. Dies ist insbesondere bei der Instandhaltung nötig, da hier sehr unterschiedliche Anforderungen je nach Geschäftsmodell oder Betriebsart der Instandhaltung vorkommen. Graaf et al. (2005, S. 350) evaluieren bspw. in ihrer Arbeit eine Referenzarchitektur im Bereich der Embedded Software und sehen einen entscheidenden Nachteil in den fehlenden Variationspunkten, was dazu führt, dass eine Instanziierung der Referenzarchitektur stark erschwert wurde. Die Referenzarchitektur muss daher ermöglichen, dass die Applikation so um den generischen Kern herum erweitert werden kann, dass bereits bekannte und unbekannte optionale Anwendungsfälle integriert werden können.

Daher lautet die Anforderung: **Die RAII soll es ermöglichen, dass die Architektur basierend auf dem generischen Kern um optionale, bereits bestehende und zukünftig hinzukommende Anwendungsfälle angepasst bzw. erweitert werden kann.**

Anforderung 8: Allgemeingültigkeit

Ein Teil des Ausdrucks Referenzarchitektur beinhaltet das Wort Referenz und bezeichnet damit gleichzeitig eine Eigenschaft, die Referenzarchitekturen in vielen Fällen besitzt: Die Referenzarchitektur muss als Referenz dienen und damit eine gewisse Allgemeingültigkeit besitzen. Dumitrescu et al. (2015, S. 16) fordern für Referenzarchitekturen im Kontext von Industrie 4.0 eine Allgemeingültigkeit im Sinne dessen, dass Unternehmen aus unterschiedlichen Branchen und Größen sie einsetzen und für ihren Fall anpassen können.

Diese und andere Sichtweisen wurden in Kapitel 2 ausgiebig beleuchtet und mündeten in das dort vorgestellte Klassifikationsschema der RAI. Nach dieser Klassifikation wird die Allgemeingültigkeit dahingehend verstanden, dass die RAI nicht nur auf die vier untersuchten Unternehmen anwendbar sein sollte, sondern auch auf andere Unternehmen und deren Anwendungsfälle. Die RAI soll dabei einen Empfehlungscharakter besitzen, sodass sie prinzipiell als Grundlage für die Entwicklung von unterschiedlichen IISs nutzbar ist. Dabei wird nicht gefordert, dass die RAI alle potentiellen Anwendungsfälle verallgemeinert und auf diese angewendet werden kann. Diese Beweisführung ist für die Erfüllung des Ziels der Referenzarchitektur nicht notwendig und wäre im Rahmen dieser Arbeit nicht zu erbringen. Siehe hierzu die Diskussion bzgl. des Begriffes der Allgemeingültigkeit in Kapitel 2 und ergänzend dazu im Kontext des Begriffes des Referenzmodells bei vom Brocke (2015, S. 31ff).

Die Anforderung der unternehmensübergreifenden Gültigkeit gilt als erfüllt, wenn die Anforderungen 5, 6 und 7 erfüllt sind. Die hier vorgestellte Anforderung fasst diese im Hinblick zusammen und bewertet sie in Bezug auf die Allgemeingültigkeit. Erst wenn ein generischer Kern sowie Standardprozesse unabhängig von den untersuchten Unternehmen vorhanden sind und diese um optionale Komponenten erweitert werden können, sodass spezielle Bedürfnisse abgedeckt werden können, kann von einer Allgemeingültigkeit nach dem hier verwendeten Verständnis gesprochen werden.

Daher lautet die Anforderung wie folgt: **Die RAI muss über die untersuchten Unternehmen hinaus als Referenzarchitektur für ein IIS anwendbar sein.**

Anforderung 9: Rückverfolgbarkeit von Anforderungen in der technischen Lösung

Die Verbindung von technischen Komponenten und Anforderungen dient dem Verständnis der einzelnen Anforderung, wie auch dem Verständnis der Komponente. Durch diese Beziehung können Anforderungen direkt in technische, funktionale Module übersetzt werden, da deren Zusammenhang klar ersichtlich ist. Die Umsetzung von Modulen bzw. Teilkomponenten sowie die Priorisierung von Anforderungen werden dadurch plastisch dargestellt. Weiter kann durch die Verbindung das Zusammenwirken mehrerer Anforderungen in einem technischen Modul illustriert werden. Durch eine solche Verbindung können die Stakeholder der Entwicklung darüber hinaus die Umsetzung ihrer Anforderungen erkennen und so durchgängig über den Entwicklungsprozess hinweg deren Erfüllung beurteilen. Das Verständnis der Stakeholder bzgl. der technischen Lösung und der Validierung ihrer Anforderung soll dadurch gestärkt und Missverständnis frühzeitig minimiert werden.

Daher ist die Anforderung folgendermaßen: **Die RAI soll Anforderungen an ein IIS mitsamt funktionalen Modulen so verbinden, dass sie über den gesamten Entwicklungsprozess hinweg in diesen Modulen nachverfolgbar sind.**

Anforderung 10: Besseres Verständnis von generischen Anforderungen

Anforderungen, die einen generischen Aspekt eines IIS betreffen, müssen bei jeder Implementierung umgesetzt werden. Ein genaues Verständnis und eine mögliche Beschreibung der Umsetzung dieser Anforderung erleichtern den Entwicklungsprozess in zweierlei Hinsicht: Zum einen hilft das Verständnis, die eigenen Anforderungen mit den generischen Anforderungen abzugleichen und neue bisher unbeachtete zu identifizieren. Zum anderen können Anforderungen an ein IIS aufgrund einer genaueren Basis durch die generischen Anforderungen mit allen beteiligten Stakeholdern leichter diskutiert und abgeglichen werden.

Die generischen Anforderungen nehmen hierbei einen besonderen Status ein, da die Kenntnis und das Verständnis dieser Anforderungen elementar zur Planung eines Projektes zur Entwicklung eines Instandhaltungssystems sind.

Daher ist Anforderung 10: **Die RAII soll zum besseren Verständnis generischer Anforderungen beitragen.**

Anforderung 11: Besseres Verständnis der eigenen Anforderungen

Analog zu Anforderung 10 hilft ein genaues Verständnis eigener Anforderungen, diese konkret zu formulieren und die Entwicklung frühzeitig auf diese auszurichten. Konkrete Beispiele und bestehende Anforderungen helfen dabei, indem sie als Referenz und Vorlagen dienen. Dadurch eröffnen sie die Möglichkeit, eigene Anforderungen von diesen abzuleiten oder sich davon abzugrenzen. Eine höhere Güte der eigenen Anforderungen und deren Beschreibung kann hiermit erzielt werden. Unterstützt die RAII Unternehmen darin, eigene Anforderungen abzuleiten, zu formulieren und verschiedene Aspekte dieser schneller aufzunehmen, gilt diese Anforderung als erfüllt.

Aus diesem Grund ist Anforderung 11 an die RAII: **Die RAII soll es Unternehmen ermöglichen, eigene Anforderungen an ein IIS besser zu identifizieren, zu verstehen und zu formulieren.**

Anforderung 12: Implementierbarkeit/Umsetzbarkeit in funktionale Applikationen

Um Vorteile aus einer Referenzarchitektur zu ziehen und diese als „gut“ zu bezeichnen, erheben Vogel et al. (2009, S. 257) die Anforderung an eine Referenzarchitektur, dass diese erfolgreich in einen ähnlichen Kontext eingesetzt worden ist, um ihre Praxistauglichkeit zu beweisen. Cloutier et al. (2009, S. 21) sehen darüber hinaus die Möglichkeit, innovative Referenzarchitekturen durch eine erfolgreiche Implementierung zu validieren. Ziel in dieser Arbeit ist es daher, anhand der RAII prinzipiell praxistaugliche Applikationen zu entwickeln.

Daher gilt die praktische Anforderung: **Auf Basis der RAII und der in ihr enthaltenen Prinzipien, Komponenten und Gesamtarchitektur sollen funktionale, praxistaugliche Applikationen erstellt werden können.**

Anforderung 13: Kompatibilität zwischen entwickelten Lösungen

Referenzarchitekturen werden in manchen Ausprägungen dazu benutzt, Softwareapplikationen, deren Schnittstellen oder Teilkomponenten dieser zu standardisieren. Der Fokus der RAII ist nicht die komplette Standardisierung, da dies dazu führen würde, dass die daraus entstehende Software sich nicht an die spezifischen Gegebenheiten von Unternehmen und den bei diesen eingesetzten Produktionssystemen anpassen könnte. Ein Unternehmen müsste in diesem Fall seine Produktionssysteme und Schnittstellen auf die Funktionsweise der RAII anpassen. Eine solche Annahme würde die Akzeptanz der RAII verringern, da diese nur mit hohem Aufwand umgesetzt werden könnte. Mit Kompatibilität im Kontext der RAII ist daher gemeint, dass es aufgrund des generischen Kerns möglich ist, Module zwischen zwei Applikationen mit wenig Anpassung auszutauschen und aufgrund der ähnlichen Architektur Schnittstellen zwischen zwei auf RAII basierenden Systemen einfach anzupassen bzw. zu implementieren, sodass ein Potential für Interoperabilität entsteht.

Kurzum: **Die Verwendung der RAII als Basis von zu entwickelnden IISs soll die Kompatibilität zwischen diesen IISs dahingehend sicherstellen, dass Daten und Module mit geringen Anpassungen austauschbar sind.**

Anforderung 14: Unterstützung des Entwicklungsprozesses

Das vordergründige Ziel der RAII ist es, die Entwicklung von IISs zu erleichtern, beschleunigen und bis zu einem gewissen Grad zu vereinheitlichen, um damit die Entwicklung zu unterstützen. Dies geschieht zum einen durch die Erfüllung von bereits genannten Anforderungen wie bspw. der Bereitstellung von Domänenwissen, Einbindung der Stakeholder und der Bereitstellung des generischen Kerns. Zum anderen soll dem Entwickler eine Blaupause des zu entwickelnden Systems zur Verfügung gestellt werden, was die Entwicklung im Endeffekt stark unterstützt. Diese Blaupause soll einen architektonischen Rahmen darstellen, der die später zu entwickelnde konkrete Softwarearchitektur umreißt und deren Erstellung fördert. Dieses einfach zu kommunizierende Bild des Systems (Muller/van de Laar, 2009, S. 7) soll den Entwicklungsprozess unterstützen. Diese Anforderung ist dahingehend eine Aggregation der vorherigen Anforderungen und untersucht die Unterstützung des Entwicklungsprozesses umfassend über die genannten Anforderungen hinweg und auch danach, ob ein konsistentes Bild über die Referenzarchitektur hinweg präsentiert wird.

Zusammenfassend wird die Anforderung wie folgt bezeichnet: **Die RAII soll den Entwicklungsprozess von Instandhaltungssystemen unterstützen, indem ein architektonischer Rahmen des zu entwickelnden Systems bereitgestellt wird.**

Nachfolgend sind die genannten Anforderungen in Tabelle 34 mit Nummer, Kurzfassung und Beschreibung dargestellt.

Anforderung	Beschreibung
Anforderung 1: Domänenübergreifende Kooperation	Die RAII muss eine Kooperation aller an der Entwicklung des Instandhaltungssystems beteiligten Personen ermöglichen und fördern, indem ein domänenübergreifendes Verständnis der Elemente des Systems, dessen Komponenten sowie den verbundenen Anforderungen über den Entwicklungszyklus hinweg möglich ist.
Anforderung 2: Bereitstellung von Domänenwissen	Die RAII soll Domänenwissen über die Instandhaltung, Instandhaltungssysteme und die zugrundeliegende Referenzarchitektur mitsamt generischer sowie optionaler Komponenten bereitstellen.
Anforderung 3: Bereitstellung einheitlicher Domänensprache	Die RAII muss eine einheitliche Domänensprache über das zu entwickelnde System und die dazugehörige Domäne für alle Beteiligten der Entwicklung schaffen.
Anforderung 4: Leichte Verständlichkeit	Die RAII soll sich durch eine leichte Verständlichkeit der Darstellung auszeichnen, sodass das Beherrschen von komplexen Notationen und Hintergrundwissen keine Notwendigkeit darstellt, um mit ihr zu arbeiten.
Anforderung 5: Abbildung des generischen Kerns	Die RAII soll den generischen Kern eines IIS über unterschiedliche Instandhaltungsszenarien und Fokusse hinweg darstellen, sodass bei unterschiedlichen mit der RAII entwickelten IISs eine einheitliche Basis vorhanden ist.
Anforderung 6: Fokussierung auf Standardprozesse	Die RAII sollte Standardprozesse eines Instandhaltungssystems über unterschiedliche Instandhaltungsszenarien und Fokusse hinweg darstellen.

Anforderung	Beschreibung
Anforderung 7: Anpassbarkeit auf optionale Fälle	Die RAII soll es ermöglichen, dass die Architektur basierend auf dem generischen Kern um optionale, bereits bestehende und zukünftig hinzukommende Anwendungsfälle angepasst bzw. erweitert werden kann.
Anforderung 8: Allgemeingültigkeit der Referenzarchitektur	Die RAII muss über die untersuchten Unternehmen hinaus als Referenzarchitektur für ein IIS anwendbar sein.
Anforderung 9: Rückverfolgbarkeit von Anforderungen in der technischen Lösung	Die RAII soll Anforderungen an ein IIS mitsamt funktionalen Modulen so verbinden, dass sie über den gesamten Entwicklungsprozess hinweg in diesen Modulen nachverfolgbar sind.
Anforderung 10: Besseres Verständnis von generischen Anforderungen	Die RAII soll zum besseren Verständnis generischer Anforderungen beitragen.
Anforderung 11: Besseres Verständnis der eigenen Anforderungen	Die RAII soll es Unternehmen ermöglichen, eigene Anforderungen an ein IIS besser zu identifizieren, zu verstehen und zu formulieren.
Anforderung 12: Implementierbarkeit	Auf Basis der RAII und der in ihr enthaltenen Prinzipien, Komponenten und Gesamtarchitektur sollen funktionale, praxistaugliche Applikationen erstellt werden können.
Anforderung 13: Kompatibilität zwischen entwickelten Lösungen	Die Verwendung der RAII als Basis von zu entwickelnden IISs soll die Kompatibilität zwischen diesen IISs dahingehend sicherstellen, dass Daten und Module mit geringen Anpassungen austauschbar sind.
Anforderung 14: Unterstützung des Entwicklungsprozesses	Die RAII soll den Entwicklungsprozess von Instandhaltungssystemen unterstützen, indem ein architektonischer Rahmen des zu entwickelnden Systems bereitgestellt wird.

Tabelle 34: Anforderungen an die RAII

Quelle: Eigene Darstellung

7.4 Analytische Evaluation der RAI

In diesem Kapitel wird die natürlichsprachliche, merkmalsbasierte Evaluation nach Fettke und Loos (2004b, S. 7) durchgeführt. Die Anforderungen aus dem vorherigen Kapitel stellen dabei entsprechend jene Merkmale der Evaluation dar, hinsichtlich denen die RAI evaluiert wird. Der Nummerierung der Anforderungen aus Tabelle 34 folgend werden diese einzeln hinsichtlich ihrer Erfüllung durch die RAI bewertet. Die Bewertung der Anforderungserfüllung erfolgt dabei argumentativ.

Anforderung 1: Domänenübergreifende Kooperation – Analytische Evaluation

Die RAI stellt für alle Beteiligten eine gemeinsame Wissensbasis dar, welche die domänenübergreifende Zusammenarbeit fördert. Domänenwissen und eine einheitliche Domänensprache sind Teil der RAI, was dazu führt, dass die beteiligten Personen eine gemeinsame Begriffswelt teilen und anhand dieser gezielter diskutieren können. Die Nennung der Anforderungen und deren Klassifikation in generisch und optional bietet daneben weitere Anknüpfungspunkte zur Kommunikation zwischen den beteiligten Personen aus verschiedenen Domänen, wodurch sachliche und zielführende domänenübergreifende Diskussionen unterstützt werden. Anhand der Auswahl generischer und optionaler Anforderungen und der damit leichteren Identifikation von eigenen Anforderungen wird die Zusammenarbeit ebenfalls gefördert.

Weiter ermöglicht die Nachverfolgbarkeit der Anforderungen über die verschiedenen Entwicklungszyklen hinweg die Möglichkeit für alle Beteiligten, bei jedem Schritt anhand des bereitgestellten Wissens zu diskutieren und die eigenen Vorstellungen zu konkretisieren. Interdependenzen zwischen domänenspezifischen Anforderungen werden durch die Zuordnung zu Modulen durch die Referenzarchitektur klar herausgestellt, sodass die beteiligten Domänen die Abhängigkeiten zwischen ihren Anforderungen erkennen und an den entsprechenden Punkten diese diskutieren können.

Die RAI ermöglicht weiterhin, dass durch sie als Artefakt kommuniziert werden kann, da ihre simplifizierte Darstellungsform gezielt Informationen auf verständliche Art präsentieren kann und somit den Entscheidungsprozess unterstützt. Durch diese Darstellung wird wie von Guessi et al. (2014, S. 2) gefordert eine leichte Verständlichkeit über alle Stakeholder hinweg erreicht, indem kein tiefes Verständnis notwendig ist, um die grundlegenden Elemente zu verstehen. Die schwerpunktspezifischen Sichten der RAI zeigen in übersichtlicher Art und Weise, welche Verteilungen, Anforderungen und Komponenten das zu entwickelnde System besitzt. Der Einstieg in jede Sicht wird durch einfache, klar zu kommunizierende Illustrationen gegeben. Anhand dieser können die jeweiligen Stakeholder ohne Fachkenntnisse über die für sie elementaren Teile der Architektur diskutieren, da auf fachspezifische Darstellungen und Notationen weitestgehend verzichtet wird. Bei Bedarf wird auch die inhaltliche Tiefe durch ausgiebigere Informationen bereitgestellt, jedoch kann dies sukzessiv je nach Bedarf des Lesers geschehen. Weiter ermöglichen die Prozessdarstellungen und deren Verbindungen mit den anderen Sichten einen leichten Einstieg in die spätere konkrete Benutzung des Systems. Durch diese gewollte Reduktion des Inhalts bei bedarfsgesteuerter Steigerung der Komplexität wird ermöglicht, dass sich Beteiligte aus unterschiedlichen Domänen und mit unterschiedlichem Wissensstand an der Diskussion sowie der Arbeit an der Entwicklung beteiligen können.

Zusammenfassend fördert die RAI durch eine gemeinsame Wissensbasis, einfache Illustration und generische sowie optionale Anforderungen die domänenübergreifende Zusammenarbeit. Die bereitgestellten Informationen helfen den einzelnen Beteiligten dabei, eine gemeinsame Sicht und Kommunikationsbasis anhand der RAI aufzubauen und über die einzelnen Punkte dieses Artefakts zu diskutieren.

Diese einzelnen Punkte werden zusammen mit der Erfüllung der Anforderungen 2, 3 und 4 durch die RAI abgeleistet.

Anforderung 2: Bereitstellung von Domänenwissen – Analytische Evaluation

Die RAI beinhaltet Domänenwissen hinsichtlich folgender Themenbereiche:

- Darstellung und Kontext von Referenzarchitekturen (siehe Kapitel 2)
- Instandhaltung, Instandhaltungsarten und -systemen bzw. Systemen der Produktion (siehe Kapitel 3)
- Generische und optionale Anforderungen an ein IIS (siehe Kapitel 4)
- Funktionale Module, deren Interdependenzen und die abstrakte Architektur des zu entwickelnden Systems inklusive Intelligenzverteilung, Standardprozessen und zugrundeliegender Use Cases (siehe Kapitel 6)

Durch diese Punkte wird ersichtlich, dass die RAI Domänenwissen bereitstellt.

Anforderung 3: Bereitstellung einheitlicher Domänensprache – Analytische Evaluation

In Verbindung mit Anforderung 2 wird einheitliches Domänenwissen mit klar definierten Bezeichnungen hinsichtlich Begrifflichkeiten der Instandhaltung, Referenzarchitekturen und innerhalb der Inhalte der RAI selbst bereitgestellt. Durch die feste Benennung der Anforderungen, der einzelnen Module und eine einheitliche Namenskonvention innerhalb der RAI wird eine einheitliche Domänensprache gewährleistet. Diese Begriffe können direkt übernommen werden und bilden eine gemeinsame Begriffswelt bei allen Beteiligten.

Anforderung 4: Leichte Verständlichkeit – Analytische Evaluation

Die RAI verfolgt das Prinzip der Reduktion von Komplexität durch Abstraktion und einer konsequenten simplifizierten Darstellung. Einfache Illustrationen erläutern die jeweiligen Sichten und damit auch einen Komplex der Architektur. Es wird kein tiefes Verständnis benötigt, um die grundlegenden Elemente zu verstehen, da auf fachspezifische Darstellungen und komplexe Notationen weitestgehend verzichtet wird. Die schwerpunktspezifischen Sichten der RAI zeigen in übersichtlicher Art und Weise, welche Verteilungen, Anforderungen und Komponenten das zu entwickelnde System besitzt. Anhand dieser können die jeweiligen Stakeholder ohne Fachkenntnisse über die für sie elementaren Teile der Architektur diskutieren.

Durch die Verbindung mit den natürlichsprachigen Anforderungen können die (technischen) Funktionsweisen der Blöcke einfach verstanden werden. Die Unterteilung in verschiedene Sichten ermöglicht unterschiedlichen Stakeholdern einzelne Themengebiete zu verstehen, ohne dass das Wissen der anderen Sichten zwingend nötig ist. Zusätzlich wird durch das zur Verfügung gestellte Domänenwissen der Aufbau eines grundlegenden Verständnisses aller Beteiligten ermöglicht.

Anforderung 5: Abbildung des generischen Kerns – Analytische Evaluation

Der generische Kern der RAII zeigt sich z. B. in der Funktionalen Sicht, da dort alle Module so unterteilt werden, dass zwischen generischen und optionalen Modulen unterschieden werden kann. Die generischen Komponenten können auf diese Weise ganz klar von den optionalen getrennt werden. Das gleiche gilt für die Anforderungen. Diese werden in generische wie auch optionale Anforderungen unterteilt. So kann auf einer niedrigeren Abstraktionsebene direkt zwischen nötigen und optionalen Anforderungen unterschieden werden. Der dritte Punkt des generischen Kerns betrifft die Prozesssicht. Hier wird ein Referenzprozess ausgearbeitet, der bei allen untersuchten Anwendungsfällen besteht und somit den generischen Kern zusammen mit den generischen Modulen und Anforderungen abrundet und klar begrenzt.

Dabei basieren die Herleitungen auf einem umfassenden Wissensfundament aus Literatur und Praxis: Zum einen durch eine umfassende Literaturanalyse über verschiedene Instandhaltungssysteme und deren Einsatzzwecke, zum anderen durch die dedizierte Aufnahme von Anforderungen an Instandhaltungssysteme in vier Unternehmen, die in sehr unterschiedlicher Art Instandhaltung betreiben.

Mit diesen Aspekten wird sichergestellt, dass die RAII den generischen Kern einer Architektur eines IIS abbildet.

Anforderung 6: Fokussierung auf Standardprozesse – Analytische Evaluation

Die RAII stellt diverse Prozesse aus unterschiedlichen Unternehmen und Domänen dar. Aus diesen generischen, für die einzelnen Unternehmen/Domänen typischen Prozessen und in der Literatur gefunden Standardprozessen wird ein Referenzprozess gebildet, der die Grundlage der RAII und ihre Abläufe darstellt. Dieser Prozess wurde von mehreren Experten dahingehend evaluiert, ob er den Kernprozess eines IIS abbildet. Weiter wird dieser Referenzprozess mit existierenden Prozessen der untersuchten Unternehmen verglichen und dahingehend angepasst, dass sichergestellt wird, dass der Referenzprozess den Kern der übrigen Prozesse abstrahiert beinhaltet. Aus diesem Grunde besitzt die RAII Standardprozesse, welche durch ihre Darstellung anhand von adaptierten UML-Aktivitätsdiagrammen mit jedem Aktivitätsschritt eine Verbindung mit den funktionalen Modulen und Entitäten aufweisen. Hierdurch ist die Grundarchitektur eng mit den Standardprozessen verzahnt. Durch diese Verzahnung und einer Ausrichtung der restlichen Sichten auf diesen Prozess wird die Entwicklung anhand der RAII auf diese Standardprozesse fokussiert. Als Limitation dieses Referenzprozesses kann jedoch die Beinhaltung eines mobilen Endgeräts konstatiert werden.

Anforderung 7: Anpassbarkeit auf optionale Fälle – Analytische Evaluation

Analog zur Erklärung des generischen Kerns erläutert die RAII nicht nur eben jenen, sondern zeigt ebenso optionale Anforderungen, Module, Prozesse und Use Cases an. Die optionalen Anforderungen und Module können bei Bedarf übernommen und so auf den spezifischen Fall angepasst werden und sind dabei Variationspunkte der Architektur. Der modulare Aufbau der RAII führt dazu, dass die Grundarchitektur unabhängig von den gewählten Modulen bleibt, sodass auch eine spätere Erweiterung durch nicht angedachte Module möglich ist. Auf Use Cases für spezielle Anwendungsfälle wie bspw. der Wartung von Windparks kann zurückgegriffen werden. Der Referenzprozess bietet Einstiegspunkte für optionale Prozessschritte und definiert diese schon mit, kann jedoch beliebig erweitert werden. Ein Beispiel hierfür ist die Erweiterung des Instandhaltungssystems durch den Baustein des technischen Kundendienstes. Diese Module können ausgewählt und umgesetzt werden, falls dieses Geschäftsmodell beim umsetzenden Unternehmen relevant ist.

Die RAII bildet jedoch nicht in jeder Granularität die Erweiterungspunkte an. Einzelne selbstgenerische Module können mehrere Ausbaustufen haben, die sich jedoch in der Komplexität stark unterscheiden. Eine Anpassung an den eigenen Fall ist mit der RAII daher zwar möglich, jedoch bedarf es trotzdem einer genauen Analyse der einzelnen Module, deren Ausbaustufen und einer Priorisierung bei der Umsetzung. Auch erhebt die RAII nicht den Anspruch, jeden erdenklichen Anwendungsfall abzudecken. Es werden nur die in der Literatur und Praxis am häufigsten vorkommenden optionalen Anwendungsfälle abgebildet.

Zusammenfassend kann die RAII dazu genutzt werden, ein individuell auf den eigenen Anwendungsfall angelegtes IIS zu entwickeln.

Anforderung 8: Allgemeingültigkeit – Analytische Evaluation

Die RAII basiert auf der aktuellen Literatur im Kontext der Instandhaltungsunterstützungssysteme und zusätzlich auf den Anforderungen und Prozessen von vier sich stark unterscheidenden Unternehmen mit verschiedenen Betriebsmodellen der Instandhaltung. Gemeinsamkeiten der untersuchten Unternehmen und der Literatur sind daher mit sehr hoher Wahrscheinlichkeit allgemeingültig im Sinne eines Empfehlungscharakters. Weiter passen die Referenzarchitektur und ihr abstrahierter generischer Kern auf nahezu alle untersuchten Instandhaltungsfälle. Darüber hinaus ist die Änderung und Erweiterung der RAII direkt in deren Grundstruktur verankert und der Abstraktionsgrad so gewählt worden, dass Erweiterungen und Anpassungen leicht möglich sind und diese Änderungen keine Anpassung des Grundkonzepts nach sich ziehen müssen. Hierdurch kann von einer prinzipiellen Anwendbarkeit der Referenzarchitektur über die untersuchten Fälle hinaus ausgegangen werden.

Anforderung 9: Rückverfolgbarkeit von Anforderungen in der technischen Lösung – Analytische Evaluation

Die Module beinhalten generische und optionale Anforderungen. Diese Module sind in der technischen Lösung nach dem Separations of Concern (Laplante, 2007) auch in der logischen Struktur umzusetzen, wie durch die RAII vorgesehen. Hierdurch kann sichergestellt werden, dass die umgesetzten Anforderungen mit der technischen Lösung verbunden und dort klar darstellbar sind. Eine Anforderung wird immer nur durch ein Modul umgesetzt. Dadurch ist es möglich, über den gesamten Entwicklungsprozess hinweg Anforderungen zu verfolgen. So kann der Wandel der Anforderungen, der kausale Zusammenhang zwischen bereitgestellten und implementierten Funktionen sowie jegliche Änderung innerhalb des Entwicklungsprozesses klar nachvollzogen werden.

Anforderung 10: Besseres Verständnis von generischen Anforderungen – Analytische Evaluation

Allgemeingültige/generische Anforderungen werden in der RAII beschrieben, mit funktionalen Modulen verbunden und deren Umsetzung beschrieben. Interdependenzen mit anderen Modulen/Anforderungen werden aufgezeigt und Teilaspekte in Form von Unterfunktionen beschrieben. Zusätzlich wird das Wirken der Module/Anforderungen durch Prozesse und Referenzprozesse auf unterschiedlichen Entitäten illustriert. Hierdurch wird eine Wissensbasis bereitgestellt, die dazu dient, dass die generischen Anforderungen besser verstanden und deren Umsetzung zielgerichteter beurteilt werden kann.

Anforderung 11: Besseres Verständnis der eigenen Anforderungen – Analytische Evaluation

Durch die genaue Darstellung und Illustration von generischen und optionalen Anforderungen, deren Umsetzung und Verbindung mit der Architektur wird eine Inhaltsbasis geschaffen, die es ermöglicht, eigene Anforderungen besser zu verstehen und dadurch klarer formulieren zu können. Eigene Anforderungen können mit jenen der RAII abgeglichen und sogleich in die Referenzarchitektur eingruppiert werden. Das Vorliegen von optionalen Anforderungen und spezifischen Ausprägungen kann durch die RAII direkt kategorisiert werden. Wird die Instandhaltung bspw. als Service Dritten angeboten, so können die eigenen Anforderungen direkt vom optionalen Baustein des technischen Kundendienstes verglichen und ggfs. übernommen werden. Die Auswirkung der Anforderungen kann durch die Verbindung mit den Modulen daher im Vorhinein viel genauer abgeschätzt werden. Weiterhin können fehlende Anforderungen durch jene der RAII leichter gefunden werden.

Anforderung 12: Implementierbarkeit/Umsetzbarkeit in funktionale Applikationen – Analytische Evaluation

Auf Basis der RAII ist ein Prototyp namens „Ressourcen-Cockpit“ entwickelt worden, der von mehreren Unternehmen in der Instandhaltung eingesetzt werden kann. Dieser Prototyp wird in Kapitel 7.5.1 vorgestellt. Mit dieser Implementierung gilt die Anforderung 12 als erfüllt.

Anforderung 13: Kompatibilität zwischen entwickelten Lösungen – Analytische Evaluation

Die RAII gibt Begrifflichkeiten, die Grundstruktur eines IIS und Variationspunkte vor. Diese ermöglichen, dass der Kern einer auf Basis der RAII entwickelten Lösung sich gleicht. Weiterhin werden optionale Module definiert, die ein Pluginkonzept beinhalten, wodurch eine einfache Integration von zusätzlichen Plugins zur Integration externer Module möglich ist. Ein weiterer Faktor, der zur Kompatibilität beiträgt, ist die Bereitstellung von OPC UA Anbindungen, sodass Maschinen, welche sich auf diesen Standard fokussieren, in jedes IIS einfach zu integrieren sind. Zusammenfassend kann trotz dieser Punkte jedoch von keiner „Out-of-the-box“ Kompatibilität zwischen IISs gesprochen werden, die auf der RAII basieren. Denn der generische Kern zwischen den jeweiligen Applikationen erlaubt durch seine Abstraktheit und Technologiefreiheit große individuelle Anpassungen bei der Ableitung konkreter Lösungen und damit auch unterschiedliche Entwicklungen von Schnittstellen. Durch die standardisierte Maschinenanbindung, das Pluginkonzept und den identischen generischen Kern wird nur eine einfache Anpassung hin zur Kompatibilität durch die RAII ermöglicht.

Anforderung 14: Unterstützung des Entwicklungsprozesses – Analytische Evaluation

Die RAII benennt Anforderungen an ein IIS, funktionale Module und deren Funktionsweise. Weiter werden Interdependenzen und Verbindungen zu Standardprozessen beschrieben. Mehrere Faktoren unterstützen die Entwicklung durch diese Informationen:

- **Anforderungserhebung und -management:** Durch die Bereitstellung von generischen und optionalen Anforderungen können eigene Anforderungen einfach abgeleitet oder mit Informationen aus den RAII-Anforderungen angereichert werden. Hierdurch wird die Qualität der Anforderungen erhöht und die Wahrscheinlichkeit, wichtige Anforderungen zu spät zu erkennen, wird verringert. Zusätzlich werden die

Anforderungen direkt in funktionale Module überführt, wodurch die Verfolgbarkeit und Plastizität der Anforderungen gesteigert wird. Diese Punkte erhöhen die Anforderungsgüte hinsichtlich der Erhebung und des Managements, wodurch die Entwicklung verbessert wird, da diese konkreteren Anforderungen und nachträglichen Anforderungsänderungen belastet werden.

- **Architekturentscheidungen:** Die Grundarchitektur des Systems wird durch die funktionalen Module und deren Verteilung sowie den Referenzprozess vorgegeben. Dabei wird zwischen einem generischen Kern und optionalen Elementen unterschieden. Hierdurch wird ein exakteres Bild des Systems, dessen Komponenten und Variationspunkten gezeichnet, als bei einer reinen Sammlung von Anforderungen. Das Grundkonzept des Systems kann daher für die Entwicklung übernommen und bei Bedarf abgewandelt werden. Die Datenhaltung und Verteilung der Entitäten kann ebenso übernommen oder als Diskussionsgrundlage genutzt werden. Daher wird die Entwicklung bei der Erstellung einer eigenen Architektur unterstützt, indem sie die Referenzarchitektur als Blaupause benutzt.
- **Design und Umsetzungsentscheidungen:** Durch die Beschreibung und Erklärung der jeweiligen Module wird das Vorgehen zur Implementierung teilweise vorweggenommen. Dadurch wird die Implementierung der einzelnen Features erleichtert, auch können Interdependenzen mit anderen Features leichter erkannt werden. Die Bereitstellung von Variationspunkten in Form von optionalen Elementen ermöglicht es, aus einer bestehenden Auswahl eigene spezifische Lösungen abzuleiten. Durch die Bereitstellung dieses Wissens und der Vermittlung von Interdependenzen wird die Entwicklung entlastet, da Designentscheidungen bzgl. der konkreten Architektur erleichtert werden oder aufgrund der besseren Informationslage gründlicher und zielgerichteter getroffen werden können.
- **Steuerung des Entwicklungsprozesses:** Durch die zur Verfügung gestellten Informationen lässt sich der Entwicklungsprozess einfacher steuern und planen. Die Module besitzen alle generischen und optionalen Komponenten und sind selbst in optional und generisch eingeteilt. Zusätzlich beinhalten die Module Unterfunktionen. Anhand dieser Informationen kann die Entwicklung vorab leichter in kleine Einzelpakete unterteilt werden und durch die Unterteilung in generisch und optional werden Hilfen zur Priorisierung dieser Einzelpakete gegeben.
- **Kommunikation mit den Fachbereichen:** Ein weiterer, schon mit Anforderung 1 genannter Punkt, spielt in die Unterstützung der Entwicklung hinein. Durch die Erleichterung der domänenübergreifenden Kooperation der Referenzarchitektur wird auch die Kommunikation der Entwickler mit den jeweiligen Fachbereichen oder den Nutzern des Systems deutlich erleichtert. Dies geschieht durch die Bereitstellung einer einheitlichen Basis zur Kommunikation und die einfache Art der Visualisierung des zukünftigen Systems durch die RAI.
- **Auswahl von Systemen:** Ein weiterer Punkt, der einen positiven Einfluss auf die Entwicklung besitzt, ist, dass die Struktur der RAI es ermöglicht, dass beliebige Module prinzipiell von unterschiedlichen Systemen erbracht werden können. Mit diesem Wissen und den Modulen und Anforderungen als Vorlage können externe Systeme gezielter ausgewählt, hinzugekauft und in ein IIS integriert werden.

All diese Punkte führen dazu, dass die RAI das Ziel der Unterstützung der Entwicklung erfüllt.

7.5 Evaluation der RAII anhand des Anwendungsfalles „Ressourcen-Cockpit“

In diesem Abschnitt wird die Anwendung der RAII anhand des Anwendungsfalles des „Ressourcen-Cockpits“ gezeigt. Im Rahmen des dreijährigen Forschungsprojekts „Ressourcen-Cockpit für Sozio-Cyber-Physische Systeme“ (S-CPS) sollten individuelle IISs für vier Unternehmen mit unterschiedlichen Betriebsformen der Instandhaltung entwickelt werden. Ziel hierbei war es, die Instandhaltung und daran beteiligte Menschen adäquat durch modernste Technik und neuartiger Architekturformen, basierend auf den Möglichkeiten und Fähigkeiten von CPS, zu unterstützen. Teil des Projektes war nicht nur die Untersuchung, wie diese Applikationen gestaltet sein müssen, sondern auch, wie eine möglichst hohe Synergie zwischen diesen einzelnen Applikationen zu erzielen ist. Weiterhin wurde untersucht, wie ein solch anspruchsvoller Entwicklungsprozess mit kurzen Entwicklungszyklen unterstützt werden kann, um alle beteiligten Stakeholder einzubinden.

In diesem Zusammenhang stellt eine Referenzarchitektur eine essentielle Grundlage für die zielgerichtete Entwicklung und Umsetzung unterschiedlichster Prototypen dar. Durch Referenzarchitekturen können im Kern einheitliche und trotzdem für den Einzelfall individuelle Architekturen abgeleitet werden. Zudem kann durch die Fokussierung auf einheitliche Komponenten, Prozesse und Anforderungen der Entwicklungsprozess gesteuert werden. Weiterhin kann der Umgang mit der Referenzarchitektur und Feedback zu dieser genutzt werden, um so die Referenzarchitektur iterativ weiterzuentwickeln. Der Kontext eignet sich daher ideal, um die zur Entwicklung mithilfe einer Referenzarchitektur zu unterstützen und die Benutzung dieser bei der Entwicklung zu untersuchen.

Daraus ergaben sich zwei Möglichkeiten, die Referenzarchitektur zu evaluieren. Zum einen durch die Erstellung des Prototyps des Ressourcen-Cockpits, der auf verschiedene Unternehmen angepasst wurde. Zum anderen konnte die Benutzung der Referenzarchitektur durch die Stakeholder der einzelnen Unternehmen untersucht und durch Experteninterviews evaluiert werden. Nachfolgend werden beide Evaluationen vorgestellt.

7.5.1 Prototyp Ressourcen-Cockpit

Auf Basis der RAII wurde der Prototyp des Ressource-Cockpits entwickelt. Dies geschah in mehreren Iterationen über einen Zeitraum von 1,5 Jahren. In diesem Abschnitt wird das veröffentlichte Endergebnis hinsichtlich der Funktionalitäten, Grundkonzepte der Architektur und verwendeten Technologien beschrieben. Anschließend werden die Funktionalitäten anhand von Screenshots dargelegt. Die Ausführungen in diesen zwei Abschnitten basieren vornehmlich auf der veröffentlichten Publikation über den Prototypen von Horbach und Trommler (2017), in der die wichtigsten Aspekte des entwickelten Prototypens beschrieben werden. Diese Aspekte werden mit zusätzlichen Informationen angereichert und am Ende des Abschnitts wird die Architektur des Prototyps mit der RAII verglichen.

7.5.1.1 Technische Architektur

Die Anwendung des Ressourcen-Cockpits wurde als Anwendung so konzipiert, dass sie betriebssystemunabhängig von beliebigen Endgeräten aus erreichbar und benutzbar ist. Sie

sollte, wie von der RAI dargestellt, auf Endgeräten wie dem Backend, normalen Desktop-PCs als auch mobilen Endgeräten darstellbar sein. Voraussetzung für die Benutzung ist aufgrund der Verwendung aktueller Webtechnologien lediglich die Unterstützung aktueller Browser durch das Endgerät. Durch die Verwendung von Webtechnologien kann von einer besonderen Langlebigkeit der Applikation ausgegangen werden.

Das Ressourcen-Cockpit läuft auf den Microsoft Internet Services als Webserver, der in Kombination mit ASP.Net den Rahmen für die Anwendung vorgibt. Für das Frontend wurden die Technologien HTML5, CSS3 und JavaScript sowie Verbindungen zum Backend in der Razor-Syntax C# eingesetzt. Die Darstellung wurde für mobile Endgeräte optimiert. Um die Datenstruktur des Semantic Web abzubilden, wurde als Datenbank die dem Resource Description Framework (RDF) folgende BrightstarDB verwendet. Die Kommunikation mit dem Ressourcen-Cockpit erfolgt über JSON und XML. Der Applikation liegt das Entwurfsmuster des Model-View-Controllers zugrunde, wodurch eine klare Trennung von der Präsentations- und Logikschicht erreicht wird. Dies ermöglicht eine flexible Erweiterungs- und Änderungsmöglichkeit des Systems. Weiter wird durch dieses Entwurfsmuster die Wiederverwendbarkeit von Komponenten gefördert.

7.5.1.2 Beschreibung der Funktionalität des Ressourcen-Cockpits

Managementfunktionen

Der Prototyp verfügt über eine Reihe an Managementfunktionen, die eine dynamische Anpassung des entwickelten Ressourcen-Cockpits ermöglichen. Dargestellt sind diese in vier Grafiken, die in Tabelle 35 zusammengefasst sind. Der Screenshot oben links zeigt eine Übersicht über mehrere Managementfunktionen. Deren Ausprägungen werden beispielhaft auf den restlichen Screenshots der Tabelle illustriert:

- Dazu gehören ein frei konfigurierbares Rollensystem (siehe Screenshot unten links) und eine ausgiebige Benutzerverwaltung (siehe Screenshot rechts oben), die mit dem Rollensystem verbunden ist. Durch dieses System ermöglicht das Ressourcen-Cockpit, dass Rechte, Darstellungen und Funktionen auf den Benutzer und seine Rolle zugeschnitten werden können. Die einzelnen Benutzer können weiterhin Qualifikationen und Zertifikate zugeordnet werden, die in den jeweiligen Bereichen verwaltet werden können.
- Neben der benutzerorientierten Verwaltung können auch Anlagen und Anlagengruppen erstellt und verwaltet werden, die als Ressourcen und Ressourcen-Gruppen bezeichnet werden. So können spezifischen Namen, Lokationen und weitere Informationen zu den jeweiligen Anlagen gepflegt werden (vgl. Grafik rechts unten).

Eine weitere Funktion zur Verwaltung betrifft sogenannte Plugins. Diese ermöglichen es, dass Daten von Maschinen und Fremdsystemen in das Ressourcen-Cockpit eingelesen werden können. Das Ressourcen-Cockpit kann dabei um beliebige Arten von Plugins erweitert werden, sodass eine Interoperabilität mit bestehenden aber auch späteren Systemen mit geringerem Aufwand gewährleistet werden kann. Diese Plugins können zur Fernwartung und -steuerung von Anlagen, Maschinen und Systemen genutzt werden oder zur Integration von bestehenden ERPs, damit das Ressourcen-Cockpit Daten austauschen kann.

The screenshot displays the 'Management' interface, divided into two main sections: 'Benutzerverwaltung' (User Management) and 'Rolle bearbeiten' (Edit Role) / 'Ressource bearbeiten' (Edit Resource).

Benutzerverwaltung (User Management):

- Navigation menu on the left: Benutzer, Rollen, Qualifikationen, Zertifikate, Ressourcen, Ressourcengruppen, Pluginordner, Demo Features.
- Header: 'Management'.
- Buttons: '+ neuen Benutzer anlegen'.
- Table with columns: Verfügbar, Benutzername, Vorname, Nachname, Rollen, Qualifikation.

Verfügbar	Benutzername	Vorname	Nachname	Rollen	Qualifikation
<input checked="" type="checkbox"/>	Admin	Admin		Super User	keine Qu
<input checked="" type="checkbox"/>	mec	Finn	Faist	Default Instandhalter	FA Mec
<input checked="" type="checkbox"/>	H.Hertel	Henry	Hertel	Administrator Benutzer	keine Qu
<input checked="" type="checkbox"/>	mec3	Andre	Jinkov	Default Instandhalter	FA Mec
<input checked="" type="checkbox"/>	S.Klebber	Sven	Klebber	Instandhalter H1	keine Qu

Rolle bearbeiten (Edit Role):

- Name: Administrator Benutzer
- Navigation: Berechtigungen, Ressourcengruppen, Benutzer
- Table with columns: Name, Beschreibung

Name	Beschreibung
AdministrateUsers	Benutzer verwalten
ViewManagement	Ansicht Management

Ressource bearbeiten (Edit Resource):

- Name: Anlage 1
- Hersteller: CBS
- Modellnummer: Modell A
- Beschreibung: Prüfanlage
- Seriennummer: A1000
- Standort: Location: H1B1
- Breitengrad: 0,1
- Längengrad: 0,2
- Pluginname: CBS SCPS UAConnector
- ConnectionUrl: opc.tcp://localhost:4840/
- RootNodeName: A&C|R01_Nieten01
- Buttons: Plugin starten, Plugin wechseln

Tabelle 35: Darstellung der Managementfunktionen des Ressourcen-Cockpits
 Quelle: Horbach und Trommler (2017, S. 87ff)

Fehlermanagement

Ein wichtiger Aspekt des Ressourcen-Cockpits betrifft das Fehlermanagement. Nach Auftreten eines Fehlers in einer Anlage wird dieser direkt an potentiell zuständige Instandhalter versendet oder die Fehler können diesen zugeordnet werden. Auf Abbildung 34 ist in der oberen linken Hälfte ein Ausrufezeichen für den Eingang eines neuen Auftrages bzw. eines neuen Fehlers dargestellt, den der Benutzer (in dem Fall ein Instandhalter) bearbeiten soll. Unter dieser Anzeige im Tab Fehlerkatalog können auch historische Fehler dieser Anlage bzw. über alle Anlagen hinweg dargestellt und durchsucht werden. Nach einzelnen Aspekten wie Stör-codes, Hersteller und Firmware der Anlage sowie Varianten kann die Darstellung sortiert werden. Hierdurch können Auswertungen erstellt oder ähnliche Fehler zu einem aktuellen Instandhaltungsauftrag gefunden werden. Zu jedem Fehler wird darüber hinaus eine genaue Beschreibung dargestellt. Es besteht weiterhin die Möglichkeit zu jedem Fehler die benötigten Ersatzteile, Qualifikationen und Zertifikate für Instandhalter zur Behebung anzeigen zu lassen, sofern diese Daten schon eingetragen worden sind. Daneben können Benutzer auch neue Fehler anlegen, falls bisher unbekannte Fehler entdeckt werden oder der Hersteller neue Fehlercodes für seine Anlage aufspielt.

Hersteller	Variante	Firmware	Störcode	Beschreibung	Letztes Auftreten	
	K9TZA1130030R01NZ-1KU1		710002	Vernietung n.l.O.: untere Fehlerkurve	16.06.2016 15:41:00	ansehen
	K9TZA1130030R01NZ-1KU1		2111	Hydraulik: Paketkontrolle nicht in Ordnung: Material zu dick	16.06.2016 15:51:00	ansehen

Abbildung 34: Fehlerkatalog und Fehlermeldung

Quelle: In Anlehnung an Horbach und Trommler (2017)

Wie in Abbildung 35 dargestellt, können auch sämtliche aktuelle offenen Fehler verbunden mit den jeweiligen Aufträgen dargestellt werden. Dort kann zudem angezeigt werden, welcher Instandhalter aktuell für den Fehler verantwortlich ist, wo er wann aufgetreten ist und ob die notwendigen Werkzeuge und Ersatzteile für die Behebung des Fehlers vorhanden sind. Weiterhin können hier durch die Auswahl des Lupensymbols weitere Details zum Fehler sowie die genaue Fehlerbeschreibung bis hin zur Handlungsempfehlung zur Fehlerbehebung aufgerufen werden.

Type	Anlagen	Störcode	zugewiesener Benutzer	Auftrag erstellt	Werkzeuge	Ersatzteile	Status	
✘	STÄUBII ALPHA 500	STA500		12.10.2016 09:01:10	✓	✓	11m 17s	ansehen
✘	Anlage 2	IES2		12.10.2016 09:01:10	✓	✓	11m 17s	ansehen
✘	Anlage 2	IES1		12.10.2016 09:01:10	✓	✓	11m 17s	ansehen
✘	Anlage 2	IES2C		12.10.2016 09:01:10	✓	✓	11m 17s	ansehen
✘	Anlage 1	IMS2C		12.10.2016 09:01:10	✘	✘	21m 17s	ansehen

Abbildung 35: Anzeige aktueller Fehler mit Status

Quelle: Horbach und Trommler (2017, S. 94)

Auftragsverwaltung & Mitarbeitermanagement

Neben der Fehlerverwaltung und -historie existiert eine dedizierte Auftragsverwaltung und zielgruppenspezifische Darstellung von Aufträgen. In Abbildung 36 sind die aktuellen und vergangenen Aufträge zu erkennen und die jeweilige Person, die diese bearbeitet hat oder an diesem Auftrag arbeitet. Darüber hinaus sind die Aufträge wie auch die Fehler mit Informationen zu Werkzeugen und Ersatzteilen verbunden. Ebenso sind genauere Informationen zu diesem Auftrag erreichbar und können wiederum durch das Lupensymbol ausgewählt werden.

Aktuelle Aufträge							
Type	Anlage	zugewiesener Benutzer	Auftrag erstellt	Werkzeuge	Ersatzteile	Status	
✘	Anlage 1		11.10.2016 11:48:51	✓	✓	5m 22s	🔍 ansehen

Auftragshistorie				
Type	Anlage	zugewiesener Benutzer	Auftrag quittiert	
✘	Anlage 2	Lopp, Lenard	11.10.2016 11:49:13	🔍 ansehen
✘	Anlage 1	Lopp, Lenard	11.10.2016 11:49:22	🔍 ansehen

Abbildung 36: Anzeige aktuelle Aufträge und Auftragshistorie

Quelle: Horbach und Trommler (2017, S. 93)

Nach Auswahl des Lupenicons gelangt der Nutzer zu der in Abbildung 37 dargestellten Nutzermaske. In dieser erweiterten Informationsseite werden Handlungsempfehlungen zu diesem Auftrag bzw. dem Fehler sowie weitere Informationen wie Benutzerhandbücher, historische Daten, Ersatzteile passend zu dem Auftrag sowie Schaltpläne der Anlage angezeigt. Die Handlungsempfehlungen sind in diesem Prototyp als PDF darstellbar. Weiterhin ist es möglich, die Aufträge zu kommentieren und insbesondere fehlerhafte Beschreibungen zu verbessern.

Abbildung 37: Ansicht Handlungsanweisungen zur Fehlerbehebung Ressourcen-Cockpit

Quelle: In Anlehnung an Horbach und Trommler (2017)

Abbildung 38 zeigt eine andere Darstellung der Auftragsverwaltung. Diese Darstellung ist das persönliche Dashboard eines Instandhalters, welches dieser zur Arbeitsplanung und als Informationshub nutzen kann. Hier sieht der Instandhalter die nötigen Informationen zu den

von ihm durchzuführenden Wartungen und Fehlerbehebungen. Weiter können neu ankommende Aufträge vom Instandhalter angenommen oder abgelehnt werden. Mitarbeiter besitzen ebenso die Möglichkeit durch Auswahl ihrer Aufträge, diese abzubrechen, zu pausieren oder nach Beendigung des Auftrags zu quittieren. Zusätzlich kann durch das Dashboard eine Übersicht neuer Nachrichten dargestellt werden, die nach der Auswahl direkt gelesen werden können.

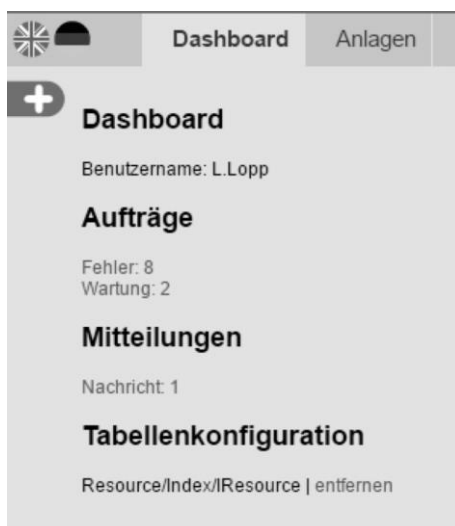


Abbildung 38: Dashboard eines Mitarbeiters mit Übersicht der aktuellen Aufträge und Mitteilungen

Quelle: Horbach und Trommler (2017, S. 86)

Anlagenübersicht

Durch den entwickelten Prototyp ist es möglich, Daten über das Einbinden von maßgeschneiderten Plugins von Anlagen abzugreifen. Umgesetzt wird diese Art der Integration für Anlagen mit OPC-UA Schnittstellen. Exemplarisch dargestellt ist diese Funktion in Abbildung 39. Hier ist eine Darstellung der Betriebsdaten der Anlage inklusive Betriebstemperatur und Auswertung hinsichtlich der Gesamtnutzung und der Betriebsstunden zu sehen. Darüber hinaus ist es möglich, auf die Wartungsinformationen dieser Anlagen zuzugreifen, die zukünftige und vergangene Wartungen anzeigt, auf aktuelle und vergangene Aufträge der Anlage direkt zuzugreifen und vorhandene Dokumente zu dieser Anlage anzeigen zu lassen.

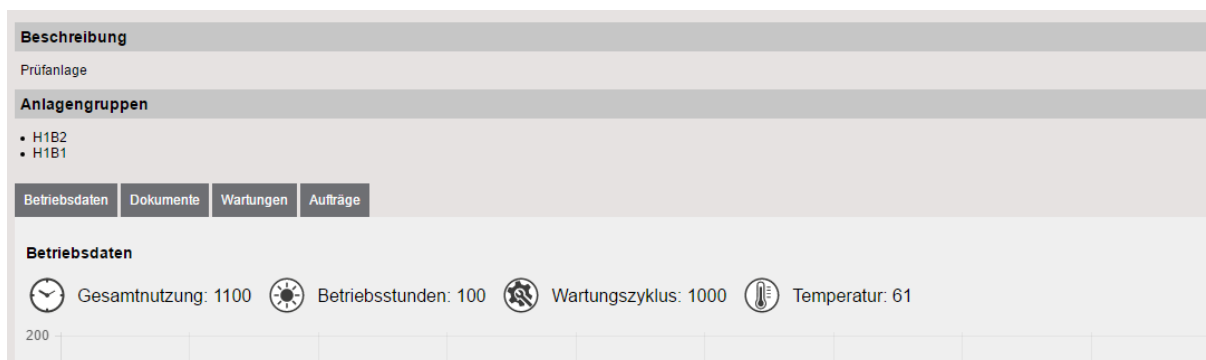


Abbildung 39: Anlageninformationen und Betriebsdaten

Quelle: Horbach und Trommler (2017, S. 91)

Wartungsmanagement

Die besagten Wartungen können in der Nutzermaske, die in Abbildung 40 dargestellt wird, verwaltet werden, sodass regelmäßige Wartungen eingeführt und deren Durchführung überprüft werden können. Die Art der Wartung, der Wartungscode, die dazugehörige Anlage, der aktuelle Status der Wartung und die Verfügbarkeit von Werkzeugen und Ersatzteilen für die jeweilige Wartung können angezeigt und in dieser Sicht dargestellt werden. Entweder haben diese Wartungen einen geplanten Start und/oder ein geplantes Ende, können auf unbestimmte Zeit durchgeführt werden oder werden einmalig angefordert. Weiterhin existiert die Möglichkeit, die Wartungsinformationen dezentral in der Maschinensteuerung zu speichern und bei einem Wartungsfall an das Ressourcen-Cockpit senden zu lassen. Hierdurch wird die Intelligenz des Systems in Richtung Maschine verschoben. Neben diesen Funktionalitäten existiert im Ressourcen-Cockpit die Möglichkeit, den Wartungsplan als Kalender anzeigen zu lassen.

Type	Anlage	Wartungscode	zugewiesener Benutzer	Geplanter Start	Geplantes Ende	Werkzeuge	Ersatzteile	Status	Aktiv	
⚙️	Anlage 1	Hauptuntersuchung		30.09.2016 13:48:29	10.10.2016 18:30:00	✓	✓	🛑	✗	🔍 ansehen
⚙️	Anlage 2	Motordrehzahl prüfen		10.10.2016 13:48:29	10.10.2016 19:48:29	✓	✓	▶️	✗	🔍 ansehen
⚙️	Anlage 1	Werkzeugprüfung	Lopp, Lenard	11.10.2016 07:48:29	12.10.2016 10:48:29	✓	✓	▶️	✓	🔍 ansehen
⚙️	Anlage 1	Spannungsmessung		12.10.2016 13:48:29		✓	✓	⚡	✓	🔍 ansehen
⚙️	Anlage 2	Laufzeitprüfung		13.10.2016 13:48:29		✓	✓	⚡	✗	🔍 ansehen

Abbildung 40: Wartungsmanagement

Quelle: Horbach und Trommler (2017, S. 96)

Ersatzteile, Werkzeuge und Nachrichtenhub

Neben dem Wartungsmanagement existiert auch eine Verwaltung und dedizierte Anzeige von Ersatzteilen und Werkzeugen, die mit den bereits vorgestellten Aufträgen verbunden sind und deren Verfügbarkeit angezeigt wird. Die Anzeige für Ersatzteile ist in Abbildung 41 dargestellt, die Anzeige für Werkzeuge ist analog. Hier ist eine Übersicht über Ersatzteile für einen Auftrag bzw. eine generelle Übersicht von Ersatzteilen dargestellt. Die Ersatzteile können, falls sie nicht verfügbar sind, mit potentiellm Lieferdatum und der Möglichkeit Ersatzteile nachzubestellen dargestellt werden. Daneben sind Inventarnummern der Ersatzteile bzw. Werkzeuge sowie die Anzahl der benötigten Stückzahl der bestimmten Art von Ersatzteilen dargestellt.

Name	Beschreibung	Externe ID	Inventar	Benötigt		
Schraube	Schraube M4 35mm	SM04355656	1123	10	✓	🔍 ansehen
Niete	4mm Niete 20mm für Panelverbindungen	RS3616515	124	3	✓	🔍 ansehen

Abbildung 41: Ersatzteilmanagement des Ressourcen-Cockpits

Quelle: In Anlehnung an Horbach und Trommler (2017)

Auf Abbildung 42 ist der Nachrichten Hub des Prototyps zu erkennen, in dem sich die an der Instandhaltung beteiligten Personen Nachrichten senden und diese mit Aufträgen verbinden

können. So kann bzgl. der Instandhaltung ohne Medienbruch kommuniziert und frühere Unterhaltungen und Kommentare bzgl. eines Auftrags erneut aufgegriffen werden.

Nachrichten Hub - Posteingang

Postausgang Neue Nachricht

Betreff	Von	Empfangen		
Wartungszyklus Anlage 3	C.Richard	07.10.2016 11:51:36	👁 ansehen	
Anlage 2 streikt!	H.Hertel	07.10.2016 11:49:11	👁 ansehen	
Krank	H.Hertel	07.10.2016 11:45:57	👁 ansehen	
Auftrag für Anlage 1 - Werkzeuge	S.Klein	07.10.2016 11:44:44	👁 ansehen	✖ entfernen

Abbildung 42: Nachrichtenhub Ressourcen-Cockpit

Quelle: Horbach und Trommler (2017, S. 97)

Neben den hier vorgestellten Funktionalitäten beinhaltet der Prototyp weitere Funktionen wie die Bereitstellung eines Wikis, Kalenderfunktionen, ERP-Integration und viele weitere kleinere Funktionen. Ein Vergleich und eine Auflistung der Fähigkeiten mit der Referenzarchitektur sind im folgenden Abschnitt zu finden. Für weitere Informationen sei auf den Beitrag von Horbach und Trommler (2017) verwiesen.

7.5.1.3 Evaluation des Prototyps

In Tabelle 36 ist ein Vergleich des Prototyps und dessen Funktionen mit den Modulen der RAI dargestellt. Dieser dient der Evaluation der RAI, da zwar durch den Prototypen die Implementierbarkeit der RAI nachgewiesen ist, jedoch noch zu klären ist, inwieweit sich die Struktur der RAI in den Prototypen und dessen Funktionen wiederfindet. Zu erkennen ist, dass 16 von den insgesamt 22 generischen Modulen durch das Ressourcen-Cockpit direkt in zumeist einfacher Ausbaustufe (die Grundfunktionalitäten des Moduls sind vorhanden) umgesetzt wurden. Ein Modul wurde von den benutzten Anlagen implementiert und zwei Module sind so umgesetzt worden, dass sie vom benutzten mobilen Endgerät ohne direkte Integration des Ressourcen-Cockpits erfolgen. Nur drei generische Module wurden für den Prototyp nicht umgesetzt, jedoch ist deren Umsetzung für weitere Entwicklungsstufen des Prototyps im Hinblick auf den produktiven Betrieb geplant.

Modul	Status
M.1. Handlungsleitfäden/Checklisten/Prüflisten	Implementiert
M.2. Dokumentenmanagement	Implementiert, einfache Ausbaustufe
M.3. Wiki	Implementiert
M.4. Anlageninformationen	Implementiert
M.5. Anlagenübersicht und -auswertung	Implementiert, Auswertung nur einfache Ausbaustufe
M.9. Auftragsverwaltung/Priorisierung	Implementiert, einfache Ausbaustufe – manuelle Priorisierung
M.10. Mitarbeitermanagement	Implementiert
M.11. Wartungsmanagement	Implementiert
M.12. Auftragsmanagement in Verbindung mit ERP	Implementiert, aktuell nur für ERP MS Dynamics

Modul	Status
M.16. Fehlererkennung & Condition Monitoring	Aktuell von Anlage implementiert, Aspekt Dezentralität
M.17. Fehlerdatenbank	Implementiert
M.18. Fehlermeldung und -darstellung	Implementiert, einfache Ausbaustufe
M.26. Kommunikation	Teilweise implementiert, nur die Kommunikation per Nachrichtenaustausch wurde umgesetzt, der Rest durch Standardapplikationen des mobilen Endgeräts
M.27. Konnektivität/Intranetzugang	Wird durch das mobile Endgerät bereitgestellt
M.28. Ersatzteilmanagement	Implementiert in einfacher Ausbaustufe
M.29. QR-CODE auslesen	Nicht Implementiert
M.30. Remote Zugriff/Fernwartung auf Anlagen/Maschinen	Implementiert in einfacher Ausbaustufe
M.31. Technisches Nutzermanagement	Implementiert
M.32. Verarbeitung audiovisueller Medien	Nicht Implementiert
M.33. Funktionalitäten mobiles Endgerät	Werden durch das mobile Endgerät bereitgestellt, dort aber nur in rudimentärer Ausbaustufe
M.34. Synchronisation zw. Geräten und Systemen	Nicht Implementiert
M.36. Pluginintegration	Implementiert
Optionale Anforderungen	
M.39. Personalisierung des Systems	Implementiert, in einfacher Ausbaustufe
M.40. Telefonersatz	Wird durch das mobile Endgerät bereitgestellt.
M.8. Kataster für Hilfs- und Betriebsstoffe der Anlagen	Implementiert

Tabelle 36: Komponentenüberprüfung Ressourcen-Cockpit

Quelle: Eigene Darstellung

Weiterhin wurden drei optionale Anforderungen umgesetzt, eine davon durch das mobile Endgerät. Durch die Umsetzung des Prototypens und die klare Orientierung an der Referenzarchitektur wird ein erweiterbares, flexibel einsetzbares System geschaffen, welches darlegt, dass die Entwicklung anhand der Referenzarchitektur möglich ist. Der Standardprozess ist darüber hinaus mit diesem System abbildbar.

7.5.2 Vorgehen Experteninterview

Als dritte Form der Evaluation wird die RAII anhand von Experteninterviews bewertet. Experteninterviews sind semi-strukturierte Interviews, die anhand eines Gesprächsleitfadens durchgeführt werden (Atteslander, 2008, S. 124). Der Gesprächsleitfaden, der in diesen Experteninterviews zur Durchführung genutzt wird, impliziert Fragen, die nach verschiedenen Themenbereichen geordnet sind (Gläser/Laudel, 2010, S. 42). Bei dieser Variante von sogenannten Leitfadeninterviews werden fachliche Experten zu einem Thema befragt, um ihr Spezialwissen zu erschließen (Döring/Bortz, 2016, S. 373). Als Experten werden nach Scholl (2009, S. 69) Personen bezeichnet, die diesen Expertenstatus aufgrund ihrer Position oder Funktion innerhalb einer Organisation aufweisen. Nach Meuser und Nagel (2002, S. 73) ist ein Experte eine Person, die

- in irgendeiner Weise Verantwortung trägt für den Entwurf, die Implementierung oder die Kontrolle einer Problemlösung oder
- über einen privilegierten Zugang zu Informationen über Personengruppen oder Entscheidungsprozesse verfügt.

Ziel eines leitfadengestützten Experteninterviews ist dabei die Gewinnung von Erkenntnissen, die sich nicht aus anderen Quellen oder nur sehr eingeschränkt ermitteln lassen (Kaiser, 2014, S. 35). Nach Wengraf (2001, S. 51) können die Experteninterviews einerseits explorativ für die Ableitung eigener Forschungshypothesen benutzt werden („model-building“), andererseits auch zur Überprüfung von Theorien („model-testing“). Die Erkenntnisse werden daher entsprechend für eines der beiden Ziele benutzt. In dieser Arbeit wird durch die Durchführung der Experteninterviews die Überprüfung von Theorien, hier: der Referenzarchitektur angestrebt.

Die Experteninterviews in dieser Arbeit haben es zum Ziel, die RAI als Theorie zu überprüfen. Dabei soll die Leitfrage der Evaluation beantwortet werden, ob und inwiefern die RAI den Entwicklungsprozess unterstützt. Zur Beurteilung eines solchen Sachverhalts ist es wichtig, dass die Experten über alle relevanten Dimensionen der Lösung Auskunft geben können und sie in ihrer Breite kennen (Niederberger/Wassermann, 2015, S. 56). Denn die Rolle der Experten ist es in einem solchen Fall nicht, Auskunft über sich selbst zu geben, sondern Beurteilungen, Positionen und Geschehnisse wiederzugeben (Lauth et al., 2015, S. 178). Dazu müssen Personen als Experten ausgewählt werden, die mit der RAI länger gearbeitet haben, Erfahrungen im Umgang mit Industrie 4.0 Projekten besitzen und daher eine große Fachkenntnis in den Bereichen IT, Produktion bzw. Instandhaltungen. In diesem Fall, bei dem objektives Expertenwissen genutzt werden soll, reichen einige wenige Experteninterviews aus (Niederberger/Wassermann, 2015, S. 56). Die ausgewählten Experten und ihre Eignung werden im folgenden Kapitel nähergehend vorgestellt.

Die Experteninterviews wurden dabei auf Basis umfangreicher, in dieser Arbeit vorgestellter Literaturrecherchen vorbereitet und hatten die komplette RAI als Grundlage. Der Leitfaden untergliedert sich in verschiedene Themenblöcke, die an den Anforderungen aus Kapitel 7.3 angelehnt sind. Der Leitfaden ist im Anhang in Tabelle 42 dargestellt.

Alle Leitfadeninterviews wurden in deutscher Sprache vom selben geschulten Interviewer mit einem Beisitzer durchgeführt. Aufgrund der großen geografischen Verteilung der Experten wurden die Interviews telefonisch durchgeführt. Insgesamt wurden drei Interviews durchgeführt, die zwischen 25-45 Minuten dauerten und während der Durchführung dokumentiert und digital aufgezeichnet wurden. Im Anschluss fand anhand der Aufzeichnungen eine Transkription der Inhalte statt. Anhand dieser Daten wurden die Experteninterviews mithilfe der qualitativen Inhaltsanalyse nach Mayring (2010b) ausgewertet. Eine Vorstellung des Vorgehens dieser Analyse befindet sich in Kapitel 7.5.2.2, die Diskussion und Auswertung der Analyse wird in Kapitel 7.5.2.3 diskutiert.

7.5.2.1 Experten

Für die Beantwortung der vorliegenden Fragestellung bzgl. der Evaluation der Referenzarchitektur ist zum einen ein tiefgehendes Verständnis und Erfahrung im Bereich der Entwicklung von Industrie 4.0 Projekten im Bereich der Instandhaltung bzw. Produktion

erforderlich. Zum anderen müssen sich die Personen intensiv mit der RAI auseinandergesetzt und diese für die Entwicklung genutzt haben bzw. an dieser beteiligt gewesen sein. Die befragten Personen besitzen den erforderlichen Expertenstatus bzgl. 4.0 Projekten und weisen Arbeitserfahrungen im Umgang mit der RAI auf. Nachfolgend werden die Personen vorgestellt.

Person 1 (P1): Managerin der Produktions- und Prozessplanung mit Fokus auf der Digitalisierung von Produktionsplanungsprozessen. Ursprünglich Experte in der Fertigungsplanung und bei Automatisierungskonzepten mit Forschungshintergrund. Große Erfahrung mit Industrie 4.0 Projekten im Konzern, insbesondere mit Big Data Projekten in der Produktion. Vorhandenes Architekturwissen vor allem auf Maschinenbauseite und IT-Wissen durch Entwicklung eigener Tools in der Vergangenheit. Arbeitete zwei Jahre mit der Referenzarchitektur.

Person 2 (P2): Leiter des Kundendienstes eines Maschinenherstellers mit Erfahrung aus mehreren Digitalisierungsprojekten, die IT und Maschinenbau verbinden und somit auch mit Industrie 4.0 Projekten. Seit sieben Jahren in dieser Position im Unternehmen. Bzgl. Architekturen liegen hauptsächlich Erfahrungen aus Maschinenarchitekturen vor. Arbeitete zwei Jahre mit der Referenzarchitektur.

Person 3 (P3): Lokaler IT-Manager mit Verantwortung über die gesamte IT des dortigen Produktionsstandorts mit großer Erfahrung im Bereich Anforderungsaufnahme/IT-Projekten, Industrie 4.0-Projekten und Erfahrung bzgl. IT-Architekturen aus Betriebssicht. Diese Person war seit 13 Jahren in dem Konzern mit den besagten Tätigkeiten befasst. Arbeitete zwei Jahre mit der Referenzarchitektur.

Die drei Personen besitzen umfangreiches Wissen bzgl. Industrie 4.0 bzw. Digitalisierungsprojekten und der damit verbundenen Anforderungserhebung. Durch ihre Projekte, welche sich immer an den Schnittstellen zu IT und Maschinenbau bewegten, waren sie die idealen Ansprechpartner bzgl. der Durchführung eines Digitalisierungsprojektes anhand einer Referenzarchitektur. Darüber hinaus konnten Sie die Referenzarchitektur schon eine geraume Zeit nutzen, sodass auch Erfahrung im Umgang damit vorhanden war. Durch den unterschiedlichen Fokus der einzelnen Personen war es möglich, die Referenzarchitektur aus verschiedenen Sichtweisen zu evaluieren und vor allem deren Eigenschaft zur Vereinheitlichung und Verständnis über Domänen hinweg zu überprüfen.

7.5.2.2 Vorgehen qualitative Inhaltsanalyse

Für die Auswertung der Experteninterviews, die einer wissenschaftlichen Systematik folgen soll, wurde die qualitative Inhaltsanalyse nach Mayring (2010b) angewandt. Dieses systematische Vorgehen drückt sich durch die Orientierung an vorab festgelegten Regeln und Methoden der Textanalyse aus. Hierdurch wird eine Reproduzierbarkeit und Nachvollziehbarkeit nach wissenschaftlichen Kriterien sichergestellt. Mayring (2010b) unterteilt den Ablauf der qualitativen Inhaltsanalyse in mehrere einzelne Schritte, die jeweils einen Teilaspekt der Analyse umfassen. Nach dem Ablaufmodell werden diese Schritte in ihrer Ausprägung jeweils auf den Kontext der Analyse maßgeschneidert angepasst. Zusammenfassend ist dieses allgemeine inhaltsanalytische Ablaufmodell in Abbildung 43 dargestellt.

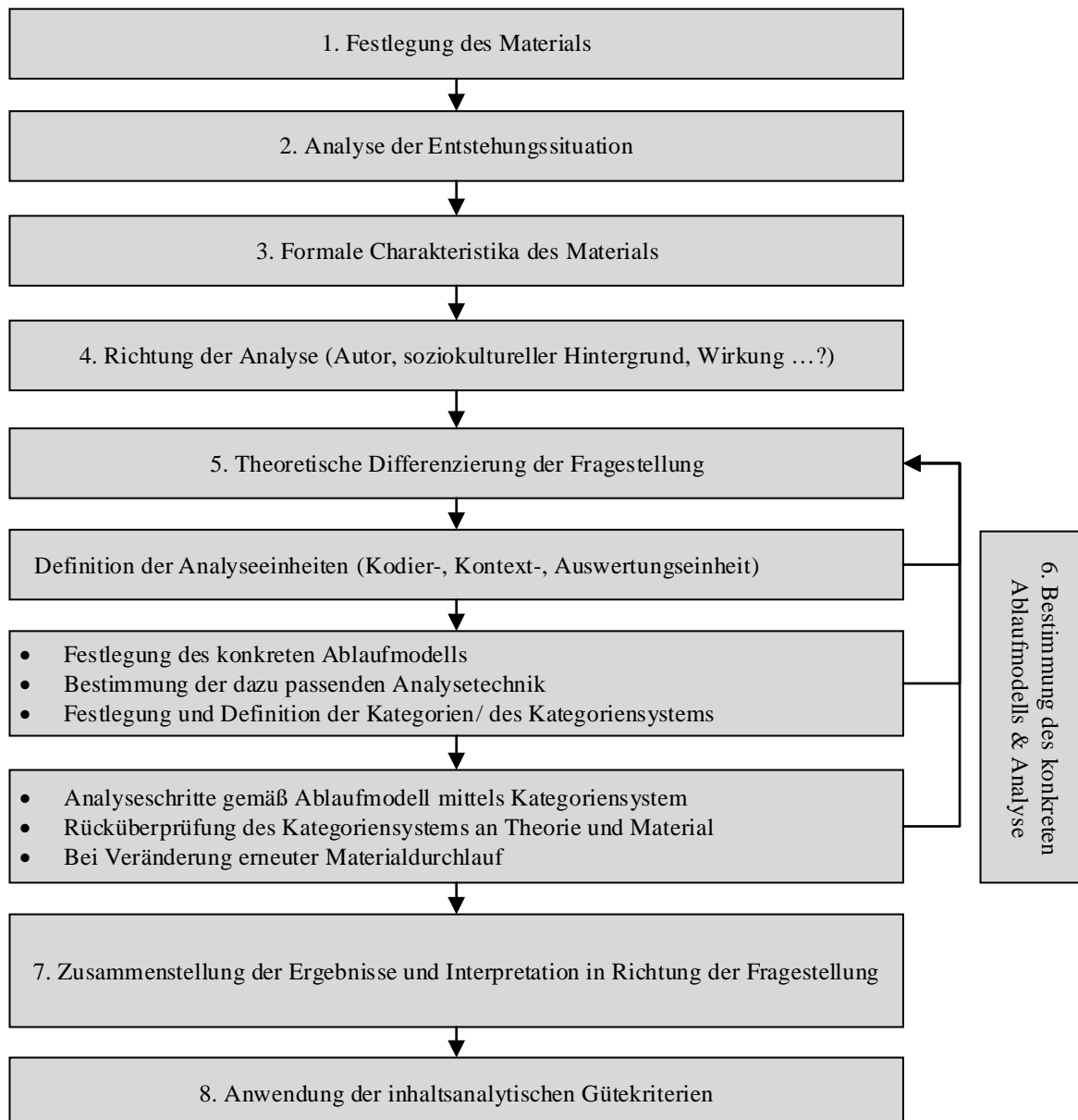


Abbildung 43: Allgemeines inhaltsanalytisches Ablaufmodell nach Mayring
 Quelle: In Anlehnung an Mayring (2010b, S. 62)

Nach dem Ablaufmodell wird zuerst eine Analyse des Ausgangsmaterials durchgeführt, um zu entscheiden, was hieraus interpretierbar ist. Dazu werden im Wesentlichen drei Analyseschritte unterschieden:

- **1. Festlegung des Materials:** Bestimmung des Materials, welches der Analyse zugrunde liegt.
- **2. Analyse der Entstehungssituation:** Beschreibung, unter welchen Bedingungen das Material produziert wurde z. B. hinsichtlich der Zielgruppe, des soziokulturellen Hintergrundes und der konkreten Entstehungssituation.
- **3. Die Beschreibung der formalen Charakteristika des Materials:** Beschreibung der Form, in der das Material vorliegt.

Nachdem das Ausgangsmaterial bestimmt, untersucht und beschrieben wurde, wird in den nächsten Schritten die Fragestellung der Analyse festgelegt. Damit wird bestimmt, was aus dem

Text herausinterpretiert werden soll (Mayring, 2010b, S. 58). Diese Analyse untergliedert sich in folgende Schritte:

- **4. Richtung der Analyse:** Hier soll bestimmt werden, was in den gesammelten Daten untersucht wird. Wird ein Gegenstand untersucht, den Experten beschreiben, oder wird der Experte an sich untersucht, um auf die Intention des Experten oder auf die Wirkung von etwas auf diesen Experten zu schließen.
- **5. Die theoriegeleitete Differenzierung der Fragestellung:** Bei diesem Schritt soll auf Basis von bestehenden Theorien die Fragestellung differenziert werden. Hierzu muss geklärt werden, an welche Theorien oder Forschung die Fragestellung der Analyse geknüpft ist bzw. auf welchen sie aufbaut, und wie diese Fragestellung weiter unterteilt werden kann.

Tabelle 37 gibt eine Übersicht der ersten fünf Schritte der Inhaltsanalyse und deren Ausprägung im Fall der Evaluation der RAI. In der linken Spalte ist der Schritt mit der jeweiligen Nummerierung genannt, in der rechten Spalte wird der Schritt in Hellgrau kurz zusammengefasst. Unter dieser Zusammenfassung wird die Ausprägung des jeweiligen Schrittes in dieser Evaluation der RAI beschrieben.

Schritt	Erläuterung
(1) Festlegung des Materials	<p>Genau Darstellung des der Analyse zugrundeliegenden Materials:</p> <p>Die Transkription der semi-strukturierten Interviews dreier Experten zur Bewertung ihrer Arbeit mit der RAI ist die Basis für die qualitative Inhaltsanalyse. Die Auswahl der Personen erfolgte nach dem Kenntnisstand bzgl. der domänenübergreifenden Industrie 4.0 Projekten und der Erfahrung im Umgang mit der Referenzarchitektur. Alle Beteiligten haben am Forschungsprojekt „S-CPS“ teilgenommen. Beschrieben werden die Personen in Kapitel 7.5.2.1..</p>
(2) Analyse der Entstehungssituation	<p>Genau Beschreibung der Bedingung, unter der das Material erstellt wurde. Wer ist der Verfasser/Sprecher, Hintergrund der Sprecher, Zielgruppe/Empfänger des Materials:</p> <p>Die Interviews wurden zwischen einem geschulten Interviewer und einem Beisitzer und den jeweiligen Experten per Telefon durchgeführt. Die Experten waren an der Entwicklung von Software-Prototypen anhand der RAI im Rahmen des Projektes „S-CPS“ maßgeblich beteiligt und konnten zwei Jahre mit der RAI arbeiten. Eine Vorstellung der Experten erfolgte in Kapitel 7.5.2.1.. Die Interviews wurde am Ende des Projektlebenszyklus von drei Jahren durchgeführt, um eine Evaluation der RAI durchzuführen. Die Teilnahme erfolgte freiwillig. Die Interviews wurden per Telefon anhand des in Tabelle 42: Leitfaden Telefoninterview</p> <p>Quelle: Eigene Darstellung dargestellten Leitfadens durchgeführt.</p>
(3) Formale Charakteristika des Materials	<p>Formale Beschreibung des Materials:</p> <p>Die Interviews wurden telefonisch durchgeführt und nach Absprache allesamt digital aufgezeichnet. Das Interview folgt einem vorab definierten Interviewleitfaden mit geschlossen und offenen Fragen, die durchgängig nummeriert sind. Analysiert wird der transkribierte Text. Der Text ist zeitlich und nach dem jeweiligen Sprecher und Frage sortiert. Jede Aussage wird einzeln dokumentiert und zeilenweise nummeriert. Dabei werden lange Sätze in mehrere Zeilen aufgeteilt. Weiterhin sind die Zeilen nach aktuellem Sprecher formatiert (P0 – P4, wobei P0 der Interviewer ist) und besitzen Textmarken (Fragen, Fragenummer und der jeweiligen Minute des Interviews) zur Navigation. Nachfragen, die direkt zu einer Frage zugeordnet werden können, werden mit kleinen Buchstaben dargestellt (z.</p>

Schritt	Erläuterung
	B. Frage 12 und 12a). Das Schema der Transkription Zeile-Minute-Frage-Sprecher-Zeileneinhalt.
(4) Richtung der Analyse	Was wurde analysiert? Inhalt im Text, Analyse des Verfassers (Sprecher), Analyse der Wirkung des Verfassers (Sprecher) auf den Rezipienten: Ziel der Analyse ist die Evaluierung der Referenzarchitektur durch die Benutzung der Aussagen bzgl. vorab definierter Kategorien/Anforderungen und offenen Fragen durch die befragten Personen. Durch die Interviews sollen die Experten dazu angeregt werden über ihre Benutzung der Referenzarchitektur zu reflektieren und diese genau zu beschreiben. Sie sollen ihre Erfahrung der Nutzung anhand der einzelnen Fragen wiedergeben und diese bei bestimmten Fragen bewerten.
(5) Theoretische Differenzierung der Fragestellung	Theoretische Verortung des Zwecks der Analyse: Die Erstellung der Referenzarchitektur hat den Zweck die Entwicklung von IISs zu erleichtern. Durch die Neuartigkeit dieser Art von Referenzarchitektur und ihrer Darstellung sollen die Benutzung und die Eigenschaften der Referenzarchitektur untersucht werden. Dazu sollen insbesondere folgende Unterfragestellungen anhand des Inhalts analysiert werden: <ul style="list-style-type: none"> • Inwiefern erfüllt die Referenzarchitektur die am Anfang der Erstellung festgelegten Anforderungen (vgl. Kapitel 7.3)? • Wie sind die Erfahrungen der Industrie 4.0 Experten mit der Arbeit der Referenzarchitektur? • Lassen sich aufgrund der Erfahrungen Verbesserungen der Referenzarchitektur ableiten? • Eignet sich die Referenzarchitektur dazu die Entwicklung zu erleichtern und Domänenwissen bereit zu stellen?

Tabelle 37: Schritte 1-5 des allgemeinen Ablaufmodells nach Mayring angewandt auf die Evaluation der RAI

Quelle: Eigene Darstellung in Anlehnung an Mayring (2010b, S. 62)

Basierend auf diesen grundlegenden Informationen wird im nachfolgenden 6. Schritt zuerst das konkrete **Ablaufmodell der Analyse** festgelegt. Dieses Ablaufmodell stellt eine der großen Stärken der qualitativen Inhaltsanalyse dar, denn dadurch wird sie nachvollziehbar und intersubjektiv überprüfbar (Mayring, 2010b, S. 61). Hierdurch werden einheitliche Regeln und Vorgehensweisen für die Kodierung des Interviews festgelegt, wodurch die Grundlage für Objektivität dieser und insbesondere eine Intercoderreliabilität hergestellt wird.

Als erster Teilschritt werden bei der Bestimmung des Ablaufmodells die **Analyseeinheiten festgelegt**, um die Präzision der Inhaltsanalyse zu erhöhen. Diese Analyseeinheiten werden nach Mayring (2010b, S. 61) in folgende Kategorien unterteilt:

*„Die **Kodiereinheit** legt fest, welches der kleinste Materialbestandteil ist, der ausgewertet werden darf, was der minimale Textteil ist, der unter eine Kategorie fallen kann.*

*Die **Kontexteinheit** legt den größten Textbestandteil fest, der unter eine Kategorie fallen kann.*

*Die **Auswertungseinheit** legt fest, welche Textteile jeweils nacheinander ausgewertet werden.“ (Mayring, 2010b, S. 61)*

Als **Kodiereinheit** wird bei der Evaluation der Referenzarchitektur ein Satz bzw. eine Aussage des Interviewten verwendet, der eine Anforderung oder Eigenschaft beschreibt oder bewertet.

Jede Aussage wird für spätere Referenzierungen mit der Fragennummer, Personen ID und Aussagennummer aus dem Transkript kodiert (Bsp. **06-P3-03**). Die Fragennummer bestimmt hierbei den Kontext der Antwort. Die Transkription wurde auf Basis der Fragen durchgeführt. Ist die Antwort eines Experten auf eine Frage in mehrere Aussagen aufteilbar, die getrennt voneinander ausgewertet werden können, so wurden diese als eigene Aussagen mit einer Aussagennummer markiert. Die Aussagennummer kennzeichnet somit, dass bei einer Frage im Interview mehr als eine Aussage getroffen wurde. Die **Kontexteinheit** und **Auswertungseinheit** sind bei der Zusammenfassung und Strukturierung jeweils das einzelne gesamte Interview einer Person. Die Interviews wurden nacheinander und geordnet nach dem Entstehungszeitpunkt analysiert.

Im nächsten Teilschritt der Festlegung des Ablaufmodells wird die **Auswahl der geeigneten Analysetechnik(en)** getroffen. Diese werden vor dem Kodieren festgelegt, um den wissenschaftlichen Umgang mit Texten auf ihre Grundstruktur hin zu überprüfen. Dabei werden verschiedene Grundformen des Interpretierens von Texten mit inhaltsanalytischen Regeln beschreibbar und überprüfbar gemacht (Mayring, 2010a, S. 602). Mayring (2010b, S. 67f) unterscheidet hierbei zwischen folgenden drei Grundformen der Analyse:

*„**Zusammenfassung:** Ziel dieser Analyse ist es, das Material so zu reduzieren, dass die wesentlichen Inhalte erhalten bleiben, durch Abstraktion einen überschaubaren Corpus zu schaffen, der immer noch Abbild des Grundmaterials ist.*

***Explikation:** Ziel der Analyse ist es, zu einzelnen fraglichen Textteilen (Begriffen, Sätzen, ...) zusätzliches Material heranzutragen, das das Verständnis erweitert, das die Textstelle erläutert, erklärt, ausdeutet.*

***Strukturierung (deduktive Kategorienanwendung):** Ziel der Analyse ist es, bestimmte Aspekte aus dem Material heraus zu filtern, unter vorher festgelegten Ordnungskriterien einen Querschnitt durch das Material zu legen oder das Material aufgrund bestimmter Kriterien einzuschätzen.“ (Mayring, 2010b, S. 67)*

Bei der Auswahl der Analysetechnik wurden mehrere Methoden kombiniert. Zuerst wurde ein Teil der Methode der Zusammenfassung verwendet, um die transkribierten Aussagen zu paraphrasieren. Hierdurch wurden die Aussagen auf ein einheitliches sprachliches Niveau gebracht und Phrasen wie bspw. „ähm“, „äh“ und „hmm“ entfernt. Dabei wurden jedoch Unterschiede in der Stärke der Zustimmung bzw. Ablehnung wie „ja das kann man so sagen“ (14-P1-01) und „ja, ein ganz klares ja“ (11-P2-01) beibehalten. Einzelnen Aussagen, die den Kontext der Frage nicht widerspiegeln, wie „ja, ein ganz klares ja“ (11-P2-01), wurde dem Fragenkontext hinzugefügt.

Für die Interviews wurde anschließend auf die Analysetechnik **skalierende Strukturierung** als Spezialform der Strukturierung zurückgegriffen. Diese Art der Strukturierung will zu einzelnen Einschätzungsdimensionen Ausprägungen in Form von Skalenpunkten definieren und das Material daraufhin einschätzen (Mayring, 2010b, S. 99f). Die Strukturierung basiert daher auf im Vorfeld festgelegte Kategorien, die sich im Kontext der durchgeführten Interviews aus den im Vorfeld definierten Anforderungen an die RAII ergeben (vgl. 7.3). Dabei werden Aussagen in den jeweiligen Interviews den Kategorien (hier Anforderungen) zugeordnet und dabei zusätzlich mit einer Zustimmungsausprägung bewertet. Zur Kodierung der untersuchten Interviews wurde als Einschätzungsdimension die Zustimmung bzw. Ablehnung zu einer Anforderung ausgewählt, da die Erfüllung der jeweiligen Anforderungen durch die RAII durch

die Experten beurteilt werden sollte. Sofern Aussagen von den jeweiligen Personen einer dieser Dimensionen einer Anforderung zugeordnet werden konnte, wurde dies eingetragen. Nach dieser Logik ist es auch möglich, dass eine Aussage bei mehreren Anforderungen unterschiedlich kodiert wird.

Zustimmungsausprägung	Definition	Ankerbeispiel	Kodierregeln
Z1: Volle Zustimmung	Komplette Erfüllung der Anforderung.	„ja, ein ganz klares ja“ (11-P2-01) „ja das kann man so sagen“ (14-P1-01) „Also es war für mich die wesentlichste Hilfestellung innerhalb des Projektes“ (01-P3-01)	Wenn einer Anforderung zugestimmt wird und keine Einschränkung in der Aussage enthalten ist (sonst Zustimmung mit Vorbehalt)-
Z2: Zustimmung mit Vorbehalt	Die Anforderung ist unter bestimmten Voraussetzungen erfüllt bzw. in bestimmten Grenzfällen nicht erfüllt	„Ja [...] aber, auf der Entwicklerseite wurde nicht alles so verstanden, wie es von den entsprechenden Partnern gewünscht wurde.“ (08-P3-01)	Die Aussage signalisiert Zustimmung hat aber Einschränkungen.
Z3: Ablehnende Haltung	Die Aussage besagt, dass die Anforderung nicht erfüllt ist	„Bei der Darstellung von Use Cases wäre eine reduzierte Darstellung sinnvoll.“ (23-P1-01)	Erfüllung der Anforderung wird abgelehnt bzw. ist unklar.
Z4: Bewertung nicht erschließbar	Keinerlei Aussage seitens der befragten Person möglich	„Das kann ich nicht beurteilen.“ (07-P2-01)	Wenn die Erfüllung/Nicht-Erfüllung einer Anforderung nicht beurteilt werden kann.

Tabelle 38: Übersicht Kodierung mit Ankerbeispiel und Kodierregeln

Quelle: Eigene Darstellung

Die Ausprägungen der Zustimmung wird über eine einfache Skalierung mit vier Zustimmungsausprägungen definiert, die zur Grundlage der Kodierung der jeweiligen Zustimmung dienen. Die vier gewählten Kategorien sind:

- **(Z1): Volle Zustimmung,**
- **(Z2): Zustimmung mit Vorbehalt,**
- **(Z3): Ablehnende Haltung und**
- **(Z4): Bewertung nicht erschließbar.**

Die Zustimmungsausprägungen sind das Ergebnis eines iterativen Prozesses. Hierbei wurde das Analysematerial mehrmals durchlaufen, wobei Anpassungen an den Dimensionen der Zustimmungsausprägung durchgeführt wurden.

Im nächsten Schritt werden zu jeder Dimension (Z1-Z4) Definitionen und ein Ankerbeispiel bereitgestellt. Außerdem werden Kodierregeln festgelegt, mit denen die Zuordnung der einzelnen Aussagen im Zweifel durchgeführt wird. Die Informationen sind in Tabelle 38 dargestellt.

7.5.2.3 Analyse der Ergebnisse

In der Durchführung der Inhaltsanalyse wurden die Aussagen der Experten den entsprechenden Anforderungen und Eigenschaften zugeordnet. Dabei wurde jede Aussage unabhängig einer oder mehreren Anforderungen zugeordnet. Eine Aussage kann eine Zustimmung für eine Anforderung enthalten und eine ablehnende Haltung für eine andere. Die Nummerierung wurde an die Nummerierung der Anforderungen aus Kapitel 7.3 angeglichen. Hierbei können mehrere Eigenschaften, die im Experteninterview als Kategorisierung benutzt wurden, einer Anforderung aus der merkmalsbasierten Analyse zugeordnet werden (z. B. die Kategorien 9.1 Rückverfolgbarkeit von Anforderungen und 9.2 Verbindung Anforderung mit technischer Lösung sind beide der Anforderung Rückverfolgbarkeit von Anforderungen aus Kapitel 7.3 zuzuordnen).

Veranschaulicht werden diese Ergebnisse in zwei Diagrammen, welche die Erfüllung der Anforderungen zeigen. Dabei werden die Aussagen pro Anforderung nach Kategorie (Z1-Z4) prozentual ausgewertet. In Abbildung 44 wird die Zustimmung getrennt nach Experte dargestellt und in Abbildung 45 aggregiert über alle Experten. Für die aggregierte Ansicht wird der Mittelwert der normierten Aussagen der einzelnen Experten verwendet. Für die Kategorie 6 und 9.1 konnte keine Aussage von P2 und für Anforderung 6 konnte keine Aussage von P3 zugeordnet werden.

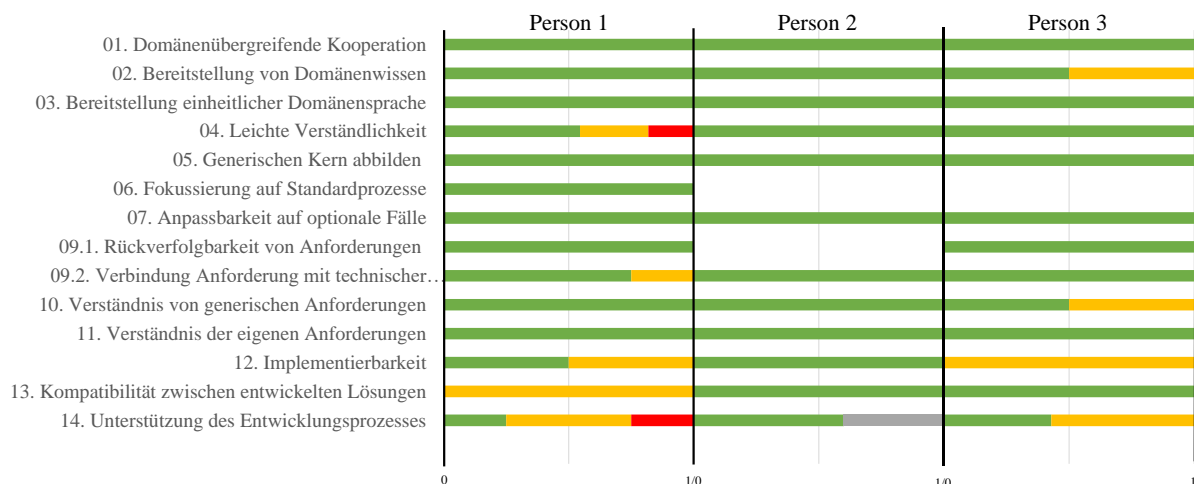


Abbildung 44: Auswertung Evaluation Experteninterview nach einzelnen Personen

Quelle: Eigene Darstellung

Im Ganzen ist klar zu erkennen, dass die Referenzarchitektur die an sie gestellten Anforderungen erfüllt. Lediglich bei der Anforderung „Unterstützung des Entwicklungsprozesses“ ist die Zustimmung nicht so stark ausgeprägt. So wird z. B. bestätigt, dass das Abstraktionsniveau der RAI für Nicht-Entwickler in Ordnung ist („das Abstraktionsniveau war für nicht-Entwickler völlig in Ordnung“ (07-P3-01)). Für die Entwicklung einer Instanz sind jedoch noch Spielräume vorhanden, die nochmals näher spezifiziert werden müssen. „Für Entwickler ist das Abstraktionsniveau wahrscheinlich schon zu allgemein. Bei der eigenen Entwicklung ist es dann auch so gekommen, dass es noch ein paar Fragestellung gab [...]“ (06-P3-03). Dieser Gegensatz ist zu erwarten gewesen, da sich die RAI, wie viele Referenzarchitekturen, im Spannungsfeld zwischen Abstraktion, Anwendung auf vielen Einsatzgebieten und Detaillierung, spezifische Anpassungen auf einen Anwendungsfall, bewegt.

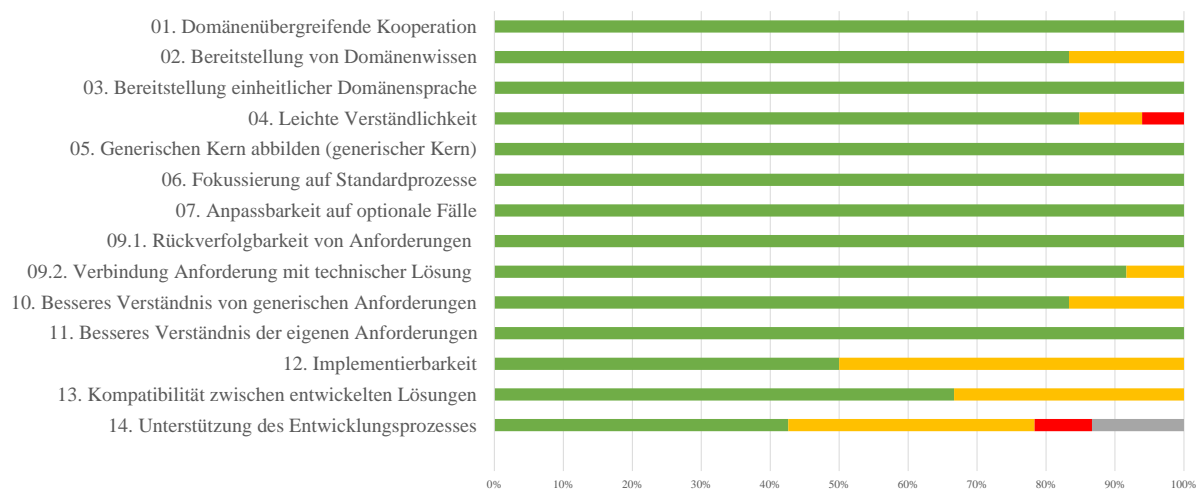


Abbildung 45: Auswertung Evaluation Experteninterview Gesamt

Quelle: Eigene Darstellung

Auch in Punkt 4 „Einfache Verständlichkeit“ ist das Abstraktionsniveau ausschlaggebend für die unterschiedlichen Aussagen. Hier stehen die Aussagen verschiedener Stakeholder in Widerspruch zu den im vorherigen Abschnitt getätigten Aussagen „Ja [das Abstraktionsniveau ist angemessen], aber für Nicht-Entwickler, bei denen die Prozesse im Vordergrund stehen, kann man bestimmte Details reduzieren“ (07-P1-01).

Hervorzuheben ist, dass die RAI bei der Anforderungsanalyse der beteiligten Unternehmen durchweg geholfen hat, die eigenen Anforderungen zu verstehen ebenso wie auch die zur Verfügung gestellten generischen Anforderungen. Weiter wird hervorgehoben, dass man mit der Referenzarchitektur Anforderungen „gut priorisieren und sich auf die wesentlichen Anforderungen konzentrieren und diese leicht identifizieren“ (03-P1-01) kann, wobei hier angemerkt wird, dass ein Grundverständnis der Domäne (Instandhaltung) vorhanden sein muss.

Die Referenzarchitektur hat daher den beteiligten Unternehmen einstimmig geholfen. Jedoch ist der Mehrwert der einzelnen Unternehmen unterschiedlich gewesen. P1 sieht bspw. vor allem die Erzeugung einer Priorisierung im Vordergrund, wohingegen P3 eher einen Leitfadeneffekt erkennt. Im Allgemeinen hilft die Referenzarchitektur den Themenkomplex zu strukturieren und alle Experten erwähnten, dass ein roter Faden durch die Referenzarchitektur für den gesamten Entwicklungsprozess gegeben wurde.

7.6 Zusammenfassung

In diesem Kapitel wurde die Evaluierung der RAII vorgestellt. Dafür wurden die Ziele und das Vorgehen zur Evaluation vorgestellt. Weiter wurden 14 Anforderungen genannt, deren Erfüllung mit der Evaluation überprüft werden sollte. Die Evaluation unterteilte sich dafür in drei Teile: Der erste Teil der Evaluation beinhaltete die analytische Evaluation der Referenzarchitektur anhand dieser Anforderungen. Diese Evaluation wurde argumentativ vollzogen, wodurch gezeigt wurde, dass die Referenzarchitektur die mit ihr verbundenen Ziele erreicht. Im zweiten Teil der Evaluation wurde die Implementierbarkeit der RAII aufgezeigt, indem der Prototyp „Ressourcen-Cockpit“ vorgestellt wurde, welcher auf der RAII basierte. Dieser wurde der Referenzarchitektur gegenübergestellt und es konnte damit gezeigt werden, dass eine Implementierung auf Basis der Referenzarchitektur möglich ist und dass der Kern der Referenzarchitektur bei mehreren Unternehmen zur Entwicklung von IIS Anwendung finden kann, was für eine Verallgemeinerbarkeit der Ergebnisse der RAII spricht. Anschließend wurden mit Industrie 4.0-Experten im Kontext der Instandhaltung Experteninterviews durchgeführt. Diese Personen arbeiteten einen längeren Zeitraum mit der Referenzarchitektur und sollten diese im Interview anhand verschiedener offener und geschlossener Fragen bewerten. Die Ergebnisse der Bewertung durch die Experten zeigte, dass die RAII die an sie gestellte Anforderungen und damit das Ziel, die Entwicklung eines IIS zu unterstützen, erfüllte. Hierdurch kann von einer hohen Nützlichkeit und einer Bestätigung des Empfehlungscharakters der Referenzarchitektur bei der Unterstützung der Entwicklung von Instandhaltungssystemen ausgegangen werden.

8 Fazit und Ausblick

8.1 Einleitung

Im letzten Kapitel werden die Ergebnisse der vorliegenden Dissertation rekapituliert. Zuerst werden in Kapitel 8.2 die Resultate der Arbeit anhand der forschungsleitenden drei Fragestellungen resümiert. Anschließend werden in Kapitel 8.3 die Implikationen der Ergebnisse dieser Arbeit für Praxis und Forschung in vorgestellt. Diese Ergebnisse unterliegen bestimmten Limitationen bei der Vorgehensweise zu Grunde, die in Kapitel 8.4 angegeben werden. Zum Abschluss der Arbeit werden in Kapitel 8.5 weitere Forschungsmöglichkeiten auf Basis der Referenzarchitektur in einem Ausblick vorgestellt.

8.2 Zusammenfassung und Ausblick

Ziel der vorliegenden Forschungsarbeit war die Konzeption und Erschaffung einer Referenzarchitektur, welche die Entwicklung von IIS erleichtern. Dies sollte u. a. durch die Bereitstellung von Domänenwissen und Architekturkonzepten zur Ableitung konkreter Architekturen geschehen. Während des Entwicklungsprozesses sollte die gemeinsame Zusammenarbeit aller beteiligten Stakeholder durch eine einfache Darstellung und breite Verständlichkeit der RAI gefordert werden. Als Zielgruppe wurden dabei vordergründig Softwareentwickler und -architekten adressiert, aber auch alle weiteren bei der Entwicklung beteiligten Stakeholder sollten von der Referenzarchitektur profitieren. Aus dieser Zielsetzung wurden drei forschungsleitende Fragestellungen abgeleitet. Die nachfolgenden Ausführungen reflektieren die Ergebnisse dieser Arbeit anhand der Beantwortung der drei Fragestellungen.

Forschungsfrage 1: Welche Anforderungen existieren an Informationssysteme zur Unterstützung der Instandhaltung in Literatur und Praxis

Zur Beantwortung der Forschungsfrage 1 wurde in Kapitel 3 die Domäne der Instandhaltung untersucht und deren Begrifflichkeiten definiert. Verschiedene Instandhaltungskonzepte und -arten wurden vorgestellt, um ein Überblick der Domäne, ihrer Rahmenbedingungen und deren Bedürfnisse zu geben. Anschließend wurden Informationssysteme vorgestellt und definiert, die in der Produktion typischerweise vorzufinden sind. Die Informationen aus diesen Systemen sind für die Durchführung der Instandhaltung notwendig, da mit ihnen auf Ursachen von Fehlern geschlossen werden kann oder generelle Informationen über die Maschinen erlangt werden können. Daneben wurde ein Überblick über aktuelle Systeme zur Unterstützung der Instandhaltung gegeben, der es ermöglicht, verschiedene Systeme und deren Eigenschaften voneinander zu unterscheiden.

Auf Basis der Namen für verschiedene Instandhaltungsarten und verschiedene Produktions- und Instandhaltungssysteme konnte eine zielgerichtete Literaturrecherche bzgl. der Anforderungen an existierenden Systemen zur Unterstützung der Instandhaltung durchgeführt werden. Publikationen wurden in folgenden Fällen als relevant für die Beantwortung dieser Frage eingestuft: Entweder behandelten die Publikationen Instandhaltungslösungen in Form von Prototypen, wodurch es möglich war, implizit Anforderungen aus diesen Lösungen zu extrahieren. Oder die Publikationen behandelten Vorgehensweisen bzw. Frameworks für Instandhaltungslösungen, aus denen ebenfalls Anforderungen an IT-Lösungen extrahiert werden konnten. Weiter wurden auch Publikationen identifiziert, welche explizit Anforderungen an bestimmte Instandhaltungssysteme behandelten. Mehrere Aspekte ergaben sich durch diese Literaturrecherche:

- Insgesamt konnten 56 Paper nach Abzug doppelter Nennungen identifiziert werden die Anforderungen an Instandhaltungssysteme behandelten oder aus denen sich Anforderungen an Instandhaltungssysteme ableiten lassen.
- Die Publikationen stammten aus unterschiedlichen Wissenschaftsdomänen, wodurch auf die Interdisziplinarität dieses Themas geschlossen werden kann. Am häufigsten waren die Publikationen den Domänen der Produktion bzw. Industrieautomatisierung, Informatik sowie der Wirtschaftsinformatik zuzuordnen.

- 62,5 % der Publikationen behandelten direkt spezifische Systeme der Instandhaltung und stellten diese vor.
- Es konnten systemübergreifend insgesamt 135 Anforderungen in der Literatur identifiziert werden und diese den folgenden Kategorien zugeordnet werden:
 - Technischer Kundendienst und Teleservice,
 - Remote Maintenance,
 - Auftragsmanagement und -informationen,
 - Wartungsmanagement,
 - Instandhaltungsplanung, -optimierung und Risikoklassifikation,
 - Anlagenübersicht und -information,
 - Ersatzteilmanagement,
 - Mitarbeitermanagement,
 - Kommunikation,
 - Mobiles System,
 - (Historische) Auswertungen,
 - Fehlermanagement,
 - Condition Monitoring und Predictive Maintenance,
 - Dokumentenmanagement,
 - Wissensmanagement,
 - Handlungsempfehlungen,
 - Externe Dienste und
 - Funktionen zur Systemadministration.
- Viele der Anforderungen wurden system- und publikationsübergreifend erwähnt, sodass bestimmte Anforderungen deutlich häufiger genannt wurden.
- Aufgrund der Häufigkeit von bestimmten Anforderungen konnte auf deren Wichtigkeit geschlossen werden, ebenfalls darauf, ob diese als generisch angesehen werden können.
- Die am häufigsten genannten Anforderungen waren das Condition Monitoring, das Auftragsmanagement, Fehlerhistorien sowie die Übersicht der verfügbaren Anlagen mit dazugehörigen Anlageninformationen.
- Manche dieser Anforderungen (z. B. Condition Monitoring) sind typischerweise mit einem speziellen System verbunden, wurden jedoch auch sehr häufig in Publikationen genannt, die nicht dieses spezielle System behandelten.
- Bestimmte Systeme wurden häufiger in der Literatur so behandelt, dass sich Anforderungen aus ihnen ableiten lassen. Dazu gehören insbesondere MSS, CMS, EMS sowie Systeme, die Predictive Maintenance und Remote Maintenance behandelten.

Aufbauend auf diesen wissenschaftlichen Ergebnissen wurde eine Untersuchung der Instandhaltung inklusive Anforderungsaufnahme an ein Instandhaltungssystem bei vier Unternehmen durchgeführt. Diese Unternehmen hatten unterschiedliche Betriebsmodelle und Ausprägungen der Instandhaltung, sodass von einer großen Bandbreite an potentiellen Anforderungen an Instandhaltungssystemen ausgegangen werden kann. 70 Anforderungen konnten bei diesen Unternehmen identifiziert werden, die in generische und optionale Anforderungen unterteilt wurden. Die Anforderungen aus der Praxis wiesen eine sehr hohe Übereinstimmung mit den Anforderungen aus der Literaturanalyse auf, unterschieden sich jedoch teilweise im Abstraktionsniveau, sodass Anforderungen aus der Praxis mehrere Anforderungen aus der Literatur umfassten und umgekehrt. Von den 28 häufigsten Anforderungen aus der Literatur wurden alle indirekt oder direkt von den generischen

Anforderungen aus der Praxis abgedeckt, wodurch diese bestätigt wurden. Die generischen und optionalen Anforderungen zusammengenommen deckten 61 % aller Anforderungen aus der Literatur ab. Nur wenige Anforderungen wurden in der Praxis aufgedeckt, die bisher nicht in der Literatur genannt wurden.

Als Ergebnis konnte eine umfassende Analyse der Anforderungen an Instandhaltungssysteme erstellt werden, wodurch die häufigsten relevantesten Anforderungen aus Wissenschaft und Praxis identifiziert und kategorisiert werden konnten. Daraus wurden generische und optionale Anforderungen an ein IIS abgeleitet.

Forschungsfrage 2: Welche funktionellen Module werden für ein integriertes Informationssystem benötigt, welches effektiv die Instandhaltung unterstützt?

Das IIS wird durch seine Funktionen definiert und deren Ausprägungen sind entscheidend für die Unterstützung der Instandhaltung. Weiterhin stellen funktionelle Module einen elementaren Teil einer Referenzarchitektur dar und bilden die Basis zur Verbindung mit dem zu entwickelnden System. Durch die Module wird konkret festgelegt welche Elemente die Referenzarchitektur enthalten und umfassen muss.

In Kapitel 2 wurde das Vorgehen zur Erstellung von Referenzarchitekturen auf Basis von spezifischen Anforderungen an Softwarelösungen vorgestellt. Dieses Vorgehen wurde genutzt, um die funktionellen Module für die RAI abzuleiten. Auf Basis der in Kapitel 4 identifizierten wissenschaftlichen und praktischen Anforderungen wurden generische und optionale Anforderungen abgeleitet. Sofern Anforderungen für das Funktionieren eines Instandhaltungssystems essentiell bei allen untersuchten Unternehmen und/oder der Literatur zu finden waren, wurden sie als generisch festgelegt. Optionale Anforderungen hingegen sind Anforderungen, die nur für einen bestimmten Anwendungsfall, ein bestimmtes Unternehmen oder nur bei einem Teilsystem bzw. speziellen Instandhaltungssystem vorkommen. Diese Anforderungen können zur Anforderungsaufnahme und als Vorlage oder Vorbild bei der Entwicklung oder dem Kauf von Informationssystem zur Unterstützung der Instandhaltung genutzt werden.

Diese generischen und optionalen Anforderungen wurden hinsichtlich ihrer Ähnlichkeit und/oder dem technischen Zusammenhang gruppiert und in Bausteine und Module überführt. Nach mehreren Iterationen und Feedbackschleifen konnten so 40 funktionelle Module erstellt werden, die jeweils beliebig viele Anforderungen zusammenfassen können. Auf Basis der enthaltenen Anforderungen und weiteren technischen Notwendigkeiten wurden die Module in Kapitel 6 hinsichtlich ihrer Funktionalität, möglichen Unterfunktionen, Schnittstellen und technischen Voraussetzungen beschrieben. Die Module konnten aufgrund ihrer Anforderungen und Funktionalität wiederum in generische und optionale Module eines integrierten Informationssystems zur Unterstützung der Instandhaltung unterteilt werden. Dabei bilden die generischen Module die Grundlage für IIS und die optionalen Module stellen Variationspunkte der Referenzarchitektur dar.

Forschungsfrage 3: Wie sieht eine Referenzarchitektur aus, die die Entwicklung eines integrierten Informationssystems zur Unterstützung der Instandhaltung erleichtert?

Referenzarchitekturen sind ein probates Mittel, um Entwicklungen von Softwaresystemen zu erleichtern und/oder zu standardisieren. Jedoch ist der Begriff der Referenzarchitektur nicht eindeutig definiert, genauso existiert kein eindeutiges Vorgehen zur Erstellung einer Referenzarchitektur, welche die Entwicklung erleichtert. Aus diesem Grunde wurde in Kapitel 2 die Domäne der Referenzarchitekturen untersucht, um das Begriffsverständnis von einer Referenzarchitektur und das nötige Vorgehen zur Erleichterung der Entwicklung von IIS zu bestimmen. Dazu wurden verwandte Begriffe aus dem Kontext von Referenzarchitekturen und verschiedene Definitionen und Eigenschaften von Referenzarchitekturen vorgestellt. Es zeigte sich, dass stark unterschiedliche Sichtweisen auf den Begriff Referenzarchitekturen existieren, welche jeweils unterschiedliche Eigenschaften jener hervorheben. Dabei konnte zwischen sehr technischen und eher abstrakteren Ansätzen unterschieden werden. Aufbauend auf diesen Informationen wurde eine eigene Definition sowie ein Klassifikationssystem für Referenzarchitekturen entwickelt, sodass diese sich besser bestimmen und vergleichen lassen. Aufbauend auf dem auf die RAII angewandten Klassifikationsschema wurden verschiedene Methoden zur Erstellung von Referenzarchitekturen diskutiert. Diese waren jedoch entweder zu generisch oder passten nicht zu der Art von Referenzarchitektur, der die RAII zuzuordnen ist und zu dem Ziel, das mit der RAII verfolgt werden sollte. Daher wurde das Vorgehen zur Referenzarchitekturerstellung auf Basis spezifischer Anforderungen an Softwaresysteme vorgestellt. Somit wird ermöglicht, Referenzarchitekturen auf Basis von spezifischen Anforderungen zu erstellen, die das Ziel haben, die Entwicklung zu erleichtern und forschungsgetriebenen sind.

Auf Basis des Vorgehens und der Forschungsfragen 1 und 2 wurde die Darstellungsart und Ausprägung der Referenzarchitektur bestimmt. Es wurde eine Sichtendarstellung der Architektur als Darstellungsart gewählt, da auf diese Weise verschiedene Aspekte des Instandhaltungssystems hervorgehoben und diskutiert werden können, ohne die Komplexität der Darstellung zu steigern. Zusätzlich wurde eine nicht formale Darstellung innerhalb der Sichten gewählt, sodass kein Expertenwissen notwendig ist, um die Referenzarchitektur zu verstehen und anhand dieser zu diskutieren. Dadurch fördert die RAII die domänenübergreifende Zusammenarbeit mehrere Stakeholder, wodurch auch die Entwicklung erleichtert wird. Des Weiteren wurde Domänenwissen, ein einheitliches Vokabular und Anforderungen an ein IIS in die RAII und deren Sichten inkludiert. Die Sichten unterteilen sich dabei wie folgt:

- **Funktionale Sicht:** Diese beinhaltet die aus FF2 entstammenden logischen Module, erklärt diese und zeigt grafisch und inhaltlich die Verbindung mit den jeweiligen generischen und optionalen Anforderungen an. Durch diese Art der Darstellung ist die Verbindung der Module mit den jeweiligen Anforderungen leicht erkennbar und eine Nachverfolgbarkeit und Auswahl ebenjener wird ermöglicht.
- **Verteilungssicht:** Sie stellt die jeweiligen Module so dar, dass erkennbar ist, auf welcher Entität sich die Intelligenz des zu entwickelnden Informationssystems befinden soll. Hierdurch wird der Aspekt der Dezentralität, der insbesondere im IoT-Umfeld zunehmend wichtig ist, betont und als Planungsaspekt mit in die Referenzarchitektur miteinbezogen.
- **Prozesssicht:** Diese Sicht stellt einen Referenzprozess und nebengelagerte Prozesse dar, die durch das IIS abgedeckt werden müssen. Durch diese Sicht kann der Prozess, der durch das Informationssystem unterstützt wird, mit den nötigen Modulen, Variationspunkten und Endgeräten abgeleitet werden.

- Zusätzlich werden die **Use Cases** dargestellt, die der Entwicklung des Systems zugrunde liegen und eine möglichst umfassende Nutzung des Systems aufzeigen sollen. Diese bilden die Basis der Prozesssicht.

Die Nützlichkeit hinsichtlich des verbundenen Ziels der Erleichterung der Entwicklung wurde durch eine umfangreiche Evaluation in Kapitel 7 nachgewiesen. Hierzu wurde die Referenzarchitektur analytisch und per Experteninterviews evaluiert. Anhand von 14 Merkmalen wurde die Referenzarchitektur beurteilt und insbesondere die Förderung der domänenübergreifenden Zusammenarbeit und die Nützlichkeit als Referenzpunkt zur Entwicklung und Anforderungsaufnahme betont. Zusätzlich wurde die Implementierbarkeit anhand der Referenzarchitektur eines Prototypen gezeigt. Auf Basis der Prototypen konnte nicht nur die Implementierbarkeit anhand der Referenzarchitektur nachgewiesen werden, es wurde auch gezeigt, dass dieser die Entwicklung erleichtert und dass ein Großteil des finalen Prototyps den funktionellen Modulen der Referenzarchitektur entspricht.

Weiter sind die vorgestellten Resultate im Sinne der gestaltungsorientierten Forschung nach Hevner et al. (2004, S. 87) neuartig, da außer der RAI keine Referenzarchitektur für IIS existiert. Daneben sind die Ergebnisse im Sinne eines Empfehlungscharakters und über die untersuchten Unternehmen hinweg verallgemeinerbar.

Mit diesem Ergebnis wurde gezeigt, wie eine Referenzarchitektur aussehen muss, welche die Entwicklung von IIS unterstützt. Dazu wurde aufgezeigt, wie ein Vorgang zur Erstellung dieser Art von Referenzarchitekturen, deren Klassifikation und die Einordnung in die wissenschaftliche Literatur über Referenzarchitekturen erfolgen kann

8.3 Implikationen für die Praxis und die Forschung

8.3.1 Theoretischer Beitrag und Forschungsergebnisse

Der theoretische Beitrag umfasst im Wesentlichen vier Wissensbausteine: Im Bereich der Instandhaltung wird Wissen über Instandhaltungsarten und -konzepte verbunden mit einer aktuellen Übersicht und Charakterisierung von Systemen der Produktion und zur Unterstützung der Instandhaltung dargelegt. Weiterhin erhebt und strukturiert der Beitrag wichtige Anforderungen der Domäne der Instandhaltung hinsichtlich der Unterstützung durch Informationssysteme. Die Wissensbasis hinsichtlich Instandhaltungssysteme und deren Anforderungen wird so durch eine Literaturübersicht wie auch eine Aufnahme von Anforderungen in der Praxis erweitert. Diese Anforderungen werden nach bestimmten Kategorien aufgeteilt und können in generische wie auch optionale Anforderungen unterteilt werden. Sie ermöglichen eine umfassende Sicht auf die Domäne der Instandhaltungssysteme.

Die zusammengestellten und entworfenen Module sind ein weiterer Beitrag zur Theorie, da sie aufzeigen, wie Instandhaltungssysteme bzw. IIS systematisch aufgeteilt werden können und wie eine Verbindung zwischen verschiedenen Anforderungen und Instandhaltungssystemen hergestellt werden kann. Ähnlich wie der Beitrag von generischen Anforderungen ist hier der Beitrag für generische Module zu sehen. Dieses Wissen erweitert das bisherige Verständnis der Kernaspekte von Instandhaltungssystemen.

Weiterhin wird Wissen zu der Domäne der Referenzarchitekturen hinzugefügt, durch die bereitgestellte Diskussion über Definitionen und Sichtweisen zur Ausgestaltung des Begriffs der Referenzarchitektur sowie der argumentativ abgeleiteten Definition und Klassifikation von Referenzarchitekturen. Es wird dadurch ein breiteres und tieferes Verständnis von möglichen Arten von Referenzarchitekturen, deren Eigenschaften sowie Vorgehen zur Erschaffung selbiger beigetragen. Die erarbeitete Klassifikation wird dabei helfen, Referenzarchitekturen in der Forschung einzuordnen und zu bewerten. Diese Klassifikation erlaubt insbesondere die Auswahl bestimmter Methoden, die zur Erstellung von Referenzarchitekturen herangezogen werden können. Die grundlegende Systematisierung des Begriffs hilft grundlegend, das wissenschaftliche Verständnis des Begriffs weiter zu konkretisieren.

Der Hauptbeitrag dieser Arbeit besteht jedoch darin, dieses Wissen in Form der übergreifenden, leicht verständlichen RAII bereitzustellen und mit Architekturwissen über funktionelle Module, Intelligenzverteilung und Instandhaltungsprozesse anzureichern. Das Wissen von vorherigen Systemen und Architekturansätzen wird in dieser abstrakten Referenzarchitektur aggregiert und in strukturierten Zusammenhang gebracht. So entsteht eine abstrakte Designtheorie (Gregor, 2006) für IIS, die für die weitere Konstruktion von Systemen in der Domäne der Instandhaltung als Rahmenwerk genutzt werden kann (Vgl. Kapitel 1). Weiter bilden die Referenzarchitektur und das Vorgehen zur Erstellung ebenjener Art von Referenzarchitektur ab, wie diese erschaffen und evaluiert werden können, so dass die Nützlichkeit bei der Erleichterung der Entwicklung im Fokus steht. Die RAII bildet dabei die Instanziierung der Theorie über Referenzarchitekturen ab und ist ein Artefakt der entwickelten Methode. Der theoretische Beitrag der Arbeit kann, wie in Tabelle 39 dargestellt, in die acht Kategorien der Designtheorie von Gregor und Jones (2007, S. 322) unterteilt werden.

Komponente	Beschreibung	Referenzarchitektur eines integrierten Informationssystems für die Instandhaltung
Kernkomponente		
1) Verwendungszweck und Geltungsbereich (causa finalis)	Für was ist das System? Welche Anforderungen und Ziele spezifizieren das Artefakt? Welcher Geltungsbereich gilt für das Artefakt?	Die RAII soll die Entwicklung von IIS sowie Teilsystemen erleichtern. Der Anwendungsbereich betrifft alle Informationssysteme zur Unterstützung der Instandhaltung im Kontext der industriellen Produktion.
2) Konstrukte (causa materialis)	Bausteine der Theorie	Domänenwissen der Instandhaltung, Anforderungen an einzelne Systeme der Instandhaltung, generische und optionale Anforderungen an IIS, Methodik zur Erstellung von Referenzarchitekturen, Darstellungsart der Referenzarchitektur, Module, Prozesse, Entitäten
3) Form- und Funktionsprinzipien (causa formalis)	Die abstrakte Blaupause, die Prinzipien oder die Architektur, die das Artefakt beschreibt	Die Referenzarchitektur gibt eine Blaupause eines IIS aus vier Sichten wider. Use Case-Sicht, Prozesssicht, Funktionale Sicht sowie Verteilungssicht. Anforderungen, Module und Prozesse können aus dieser abgeleitet und deren Zusammenhänge erkannt werden. Dezentrale Architekturkonzepte und funktionale Modulbeschreibungen sind in dem Artefakt enthalten.
4) Veränderlichkeit des Artefakts	Welche Art der Veränderungen und Anpassungen des Artefakts sind in der Theorie schon erfasst? Ist eine Generalisierbarkeit der Artefakts auf andere Domänen gegeben (Niemann, 2013, S. 12)?	Die Referenzarchitektur ist auf Veränderung und Erweiterung ausgelegt. Zum einen durch optionale Komponenten und Variationspunkte, die Erweiterungsmöglichkeiten der Grundfunktionalitäten vorsehen und durch beliebig viele bisher nicht enthaltene Module ergänzt werden können. Zum anderen erlaubt die hohe Abstraktion der Referenzarchitektur, dass einzelne Elemente erweitert und verändert werden können, ohne dass die Grundstruktur eine Veränderung erfahren muss.
5) Nachprüfbare Aussagen	Welche Aussagen der Theorie können nachgewiesen oder falsifiziert werden?	Ziel der Dissertation war es eine Referenzarchitektur für IIS zu erschaffen, die die Entwicklung von Instandhaltungssystemen erleichtert. Dazu sollte die Referenzarchitektur den Kern eines IIS und dessen Kontext so darstellen, dass der Entwicklungsprozess und die domänenübergreifende Zusammenarbeit erleichtert wird. Durch die Basis an Anforderungen an Instandhaltungssystemen aus Wissenschaft und Praxis wurde eine umfassende Sicht auf Instandhaltungssysteme in der Referenzarchitektur Rechnung getragen. Der Kontext wird durch Domänenwissen bereitgestellt und die restlichen Herausforderungen in Anforderungen an den Ansatz übersetzt. Die Erfüllung dieser Anforderungen und damit der Nachweis der Nützlichkeit der Referenzarchitektur hinsichtlich des Ziels wurde durch drei Arten der Evaluation nachgewiesen.
6) Rechtfertigbares Wissen	Die zugrundeliegende Theorie, die das Design des Artefakts ermöglichen bzw. dessen Ergebnis oder Methode rechtfertigen.	Die Methodik und das Design der vorgestellten Referenzarchitektur basieren auf den Anforderungen für Instandhaltungssysteme aus Literatur und Praxis sowie den aktuellen State of the Art hinsichtlich Referenzarchitekturen.

7) Umsetzungshinweise (causa efficiens)	Eine Beschreibung des Prozesses zur Implementierung der Theorie in einem speziellen Kontext.	Die Referenzarchitektur an sich und die einzelnen Übersichtsgrafiken können für Gruppendiskussionen zur Entwicklung und zur Anforderungserhebung genutzt werden. Eindeutige Bezeichnungen und Prozesse ermöglichen das eigene System und Anforderungen davon abzuleiten. Ein Implementierungsbeispiel wird im Rahmen dieser Arbeit gegeben.
8) Beispielhafte Inanziierung	Eine physische Implementierung des Artefakts, welche die Darstellung der Theorie erleichtern kann. Entweder zur Demonstration oder zum Testen der Theorie.	Die praktische Implementierung der Referenzarchitektur wurde realisiert und ausgiebig in Kapitel 7.5.1 diskutiert.

Tabelle 39: Elemente einer Designtheorie

Quelle: Gregor und Jones (2007, S. 322) in der Übersetzung nach Offermann et al. (2010, S. 289ff)

8.3.2 Praktischer Beitrag der Forschungsergebnisse

Die Ergebnisse der vorliegenden Arbeit helfen in erster Linie bei der Entwicklung von Informationssystemen zur Unterstützung der Instandhaltung. Die Anforderungsaufnahme wird dadurch unterstützt, dass die Referenzarchitektur schon generische wie auch optionale Anforderungen bereitstellt, die eine Vorlage für die Diskussion von potentiellen Anforderungen erlaubt und Wechselwirkungen vorab einschätzbar machen lassen. Zusätzlich bietet die Übersicht über Instandhaltungssysteme und die Bereitstellung von Domänenwissen und die Verbindung dieser Punkte die Möglichkeit, die eigene Wissensbasis bei der Anforderungserhebung deutlich zu steigern. Eigene Anforderungen können mit den Anforderungen aus der Referenzarchitektur abgeglichen werden und so fehlende identifiziert bzw. neue eigene der Referenzarchitektur hinzugefügt werden. Eine durch die Referenzarchitektur gewährleistete Nachverfolgbarkeit zwischen Modulen und Anforderungen ermöglicht zudem ein gesteigertes Verständnis von Interdependenzen zwischen Anforderungen und Funktionen des zu entwickelnden Systems.

Beim gesamten Entwicklungszyklus ermöglicht die RAI die Diskussionen für die domänenübergreifende Zusammenarbeit zu fördern und sich auf eine gemeinsame Sicht des zu entwickelnden Systems zu einigen. Durch die einfache Darstellung und die gemeinsame Begriffswelt können interdisziplinäre Entwicklungen mit höherer Partizipation und weniger Missverständnissen durchgeführt werden. Der Forderung der domänenübergreifenden Zusammenarbeit zwischen Fachabteilung und IT-Abteilung bzw. Entwicklern wird mit der vorliegenden Referenzarchitektur Rechnung getragen.

Softwareentwickler können wichtige Architekturentscheidungen und konkrete Implementierungsentscheidungen auf Basis der wohldefinierten Referenzarchitektur treffen. Durch die Blaupause der Referenzarchitektur können solche Entscheidungen auf Grundlage genauerer Informationen und einem Gesamtüberblick des Informationssystems bzw. potentiell zu entwickelnden weiteren Systemen getroffen werden. Des Weiteren können domänenfremde Softwareentwickler durch die bereitgestellten Informationen über die Domäne der Instandhaltung die Beweggründe mancher Anforderungen besser verstehen und Architekturentscheidungen mit größerer Sorgfalt treffen. Die enthaltenen Informationen über potentielle Produktions- und Instandhaltungssysteme lassen weiterhin eine genauere Aufnahme

der Situation vor Ort bei der Entwicklung von neuen Systemen zu, da eine genauere Landkarte der IT-Architektur gezeichnet werden kann. Dies lässt insbesondere eine frühzeitige Schnittstellenplanung zu, da die relevanten Systeme vorab bekannt sind.

Ein weiterer Punkt betrifft nicht die Entwicklung direkt, sondern die Auswahl von IT-Lösungen im Unternehmen zur Unterstützung der Instandhaltung. Durch den integrativen Blick auf Instandhaltungssysteme können zukünftige Informationssysteme so ausgewählt werden, dass sie in das Konzept der Referenzarchitektur passen und so integrierbar sind, dass zukünftige Erweiterungen aus der Referenzarchitektur mit den neuen Systemen möglich sind.

Zusammengefasst kann die Entwicklung anhand der Referenzarchitektur beschleunigt werden, da die Zusammenarbeit zwischen Stakeholdern gefördert, die Qualität und Geschwindigkeit der Anforderungserhebung gesteigert wird, Blaupausen für die Ableitung von konkreten Architekturen vorhanden sind und die RAII einen roten Faden für die Entwicklung von IIS liefert. Nicht zuletzt konnte der praktische Beitrag auch durch die Entwicklung verschiedener Prototypen bei unterschiedlichen Unternehmen dargelegt werden (siehe u. a. Kapitel 7.5).

Die Beiträge zu Theorie und Praxis in Verbindung mit den Artefakten für die jeweiligen Forschungsfragen sind zusätzlich aggregiert in Abbildung 46 dargestellt. Auf der linken Seite der Grafik ist das jeweilige Forschungsartefakt in Verbindung mit den jeweiligen Forschungsfragen dargestellt. In der mittleren Spalte sind die Beiträge zur Theorie des jeweiligen Forschungsartefakts dargestellt. Es sind jene Beiträge, die in vorherigen Kapitel angesprochen wurden aufgeteilt nach den jeweiligen Artefakten. In der letzten Spalte sind die Beiträge zur Praxis in Abhängigkeit zu den Artefakten der Forschungsfragen zusammengefasst. Die Beiträge zur Praxis des Forschungsartefakts für FF2 und FF3 sind dabei zusammen dargestellt, da sich die Beiträge beider Artefakte stark ähneln und das dritte Artefakt das zweite beinhaltet.

Forschungsartefakt		Theorie	Praxis
FF1	Anforderungen an ein IIS aus Forschung und Praxis	<ul style="list-style-type: none"> • Systematisierung von Instandhaltungsarten, -konzepten und -systemen • Literaturübersicht und Vergleich von Anforderungen aus Literatur und Praxis • Ableitung von generischen Anforderungen 	<ul style="list-style-type: none"> • Anforderungsübersicht für Instandhaltungssysteme & Unterstützung der Anforderungsaufnahme für IIS • Domänenwissen für die Instandhaltung
FF2	Funktionale Module für ein IIS	<ul style="list-style-type: none"> • Systematische Unterteilung eines integrierten IIS • Herstellung einer konzeptionellen Verbindung von bestehenden Instandhaltungssystemen zu einem IIS • Unterteilung in generische, wie auch optionale, Module eines IIS 	<ul style="list-style-type: none"> • Potentielle Ableitung & Vergleich von Anforderungen an ein IIS • Förderung der domänenübergreifenden Zusammenarbeit bei der Erstellung von IIS • Unterstützung der Entwicklung eines IIS bzw. von einzelnen Komponenten • Unterstützung bei dem Einkauf und Komposition von Instandhaltungsservices • Die Entwicklung von mehreren Prototypen auf Basis der Referenzarchitektur
FF3	Referenzarchitektur für ein IIS	<ul style="list-style-type: none"> • Systematisierung des Referenzarchitekturbegriffs • Klassifizierungssystem für Referenzarchitekturen • Methode zur Erstellung Evaluierung von Referenzarchitektur (Designtheorie) • Artefakt RAII • Förderung besseres Verständnis von IIS 	

Abbildung 46: Beitrag Forschung und Praxis

Quelle: Eigene Darstellung

Auf dieser Abbildung ist ersichtlich, dass Forschungsfrage 1 vor allem Beiträge für die Instandhaltung bzw. Instandhaltungs- und Produktionssysteme bietet. Dieses Wissen enthält einen theoretischen, wie auch praktischen Beitrag und lässt sich durch generische und optionale Anforderungen darstellen lässt.

Die Module helfen bei der systematischen Unterteilung von IIS und ermöglichen es, eine konzeptionelle Verbindung von Instandhaltungssystemen zu IIS darzustellen. Der praktische Beitrag wird zusammen mit dem der gesamten Referenzarchitektur dargestellt. Hier können die Aspekte hinsichtlich Entwicklung und Einkauf bzw. Selektion von Instandhaltungsdiensten und -systemen gemeinsam betrachtet werden. Der theoretische Beitrag des Artefakts von FF3 wird neben der eigentlichen Referenzarchitektur und der Designtheorie zur Erschaffung dieser mit dem hinzugefügten Wissen bzgl. der Domäne von Referenzarchitekturen angegeben.

8.4 Limitationen

In der vorliegenden Arbeit wurde eine Referenzarchitektur für ein integriertes Informationssystem zur Unterstützung der Instandhaltung konzipiert und deren Nützlichkeit bei der Erleichterung der Entwicklung von IIS durch eine ausführliche Evaluation demonstriert. Es existieren jedoch einige Limitationen bei der vorliegenden Arbeit und deren Vorgehen.

Die Referenzarchitektur wurde auf Basis von Anforderungen aus Wissenschaft durch Literaturrecherche und Anforderungen von vier Unternehmen aus der Praxis erstellt. Obwohl die Unternehmen große Unterschiede bei der Instandhaltung sowie in Größe und Geschäftsfeld aufweisen, können die gewonnenen Informationen nicht sicher verallgemeinert werden. Dies bedürfte einer deutlich größeren Anzahl an untersuchten Unternehmen.

Weiterhin ist die entwickelte Referenzarchitektur nur eine mögliche Art, die Entwicklung zu erleichtern. Es kann jedoch keinerlei Aussage darüber getroffen werden, ob dies die bestmögliche Referenzarchitektur darstellt. Ähnliche oder andere Lösungen sind denkbar.

Die Implementierbarkeit der Referenzarchitektur konnte mit dem in Kapitel 7 vorgestellten Prototypen aufgezeigt werden. Jedoch beinhaltete dieser nicht den kompletten Umfang der Referenzarchitektur und es konnte daher nicht aufgezeigt werden, wie die Kombination von denkbaren optionalen Modulen in der Praxis funktioniert. Neben diesem Aspekt wurde der Prototyp zwar bei einem Praxisunternehmen zu Testzwecken eingesetzt, jedoch befindet sich dieser nach aktuellem Stand noch nicht im Tagesbetrieb. Die Erprobung in diesem Kontext steht daher noch aus.

Eine weitere Limitation bei der Übertragbarkeit der Ergebnisse gilt der Domäne der Instandhaltung. Die entwickelte Referenzarchitektur zielt auf die Instandhaltung von Produktionsanlagen ab. Denkbar ist, dass die Ergebnisse auf andere Bereiche übertragbar sind, jedoch war dies nicht Gegenstand der vorliegenden Arbeit.

8.5 Ausblick

Die Arbeit liefert einen wesentlichen Beitrag zur verbesserten Entwicklung von integrierten Informationssystemen zur Unterstützung der Instandhaltung. Weiter bietet sie als Basis vielfältige Möglichkeiten zur weiteren Forschung und durch den rapiden Wandel der Instandhaltung besteht weiterer Forschungsbedarf. Die in dieser Arbeit präsentierte Referenzarchitektur fokussiert sich auf die Anforderungen und die integrierte, umfassende Sicht auf Instandhaltungssysteme. Sie bleibt jedoch in den meisten Aspekten abstrakt, um Freiraum für spezifische Implementierungen zu lassen. Teilaspekte können jedoch deutlich detaillierter erforscht und die Zusammenarbeit einzelner Komponenten durch vorgegebene Konzepte und Standards erleichtert werden. Des Weiteren könnte eine weitere Formalisierung der Referenzarchitektur eine Möglichkeit bieten, Teilaspekte formal zu verifizieren und gezieltere Implementierungshilfen zu geben.

Cloud-Dienste sind zunehmend populär und ermöglichen es, bestimmte Elemente des geschilderten IIS auszulagern. Zwar ermöglicht die Referenzarchitektur prinzipiell diese Auslagerung und es bedarf keiner Anpassung in ihrem Kern. Es stellt sich jedoch die Frage, wie einzelne nicht anpassbare Cloud-Dienste miteinander kombiniert werden könnten, sodass ein ad-hoc Austausch möglich ist. Beispielsweise müssen Cloud-Dienste für Predictive Maintenance und der Mitarbeiterplanung kombiniert werden, um Synergien zwischen diesen zu ermöglichen. Deren Ergebnisse müssen in die interne Struktur der Instandhaltungslösung integriert werden, um die Informationen mit internen Diensten zu kombinieren. Um einen Lock-in Effekt und starre Strukturen zu vermeiden, sollten jedoch Dienste von verschiedenen Anbietern austauschbar sein. Die dafür notwendigen Schnittstellen und Konzepte, die auch im Rahmen der Referenzarchitektur erwähnt werden, sollten daher erforscht werden. Die Konzeption von Ontologien und Minimalschnittstellen könnten hier eine Hilfe darstellen und Dienste schnell kombinierbar machen.

Ein weiterer Aspekt betrifft die Ausgestaltung von Teilaspekten der Architektur, so könnten Ontologien oder Teilsysteme für die generischen Komponenten untersucht und entwickelt werden. Erste Ansätze existieren hierzu in der wissenschaftlichen Literatur, sie behandeln jedoch oftmals nur isolierte Anwendungsfälle.

Damit verbunden ist die Bereitstellung von Referenzarchitektur für Predictive Maintenance Lösungen. Standardtemplates und Datenstrukturen für die schnelle Bearbeitung durch entsprechende Algorithmen könnten die Akzeptanz und die Implementierungsgeschwindigkeit deutlich steigern. Bisher wurden diese Aspekte jedoch nicht praxisgerecht aufbereitet und ergründet.

Ein weiterer Forschungsaspekt betrifft die Integration der vorliegenden Referenzarchitektur in weitere bestehende Referenzarchitekturen von Produktionssystemen und in übergeordnete Rahmenwerke. Die hier offen angelegte Ausgestaltung könnte durch zusätzliche Einbindung von Produktionsinformationen weiter konkretisiert werden.

Der Aspekt der Dezentralität und die damit verbundenen Sicherheitskonzepte bedürfen weiterer Untersuchung. Hier besteht die Herausforderung, dass mehrere Komponentenhersteller von Anlagen und Anlagenhersteller selber unterschiedliche Condition Monitoring und Remote Maintenance Konzepte verfolgen. Diese bedürfen Zugang zu den entsprechenden Netzwerken und Maschinen über wohldefinierte Schnittstellen und über Unternehmensgrenzen hinweg. Dies führt oftmals zum Bedarf seitens der Anlageneigentümer, diese Schnittstellen und den

Zugang zu harmonisieren und in sicherer Art und Weise bereitzustellen, ohne dass zusätzlicher Aufwand nötig ist. Offene Plattformkonzepte zur einheitlichen Bereitstellung von Condition Monitoring Daten und Fernwartung sind jedoch bisher kaum Beachtung geschenkt worden. Weiter wurden damit verbundene relevante Sicherheitsaspekte von Industrie 4.0 Applikationen für die Instandhaltung von Reidt et al. (2016b) im Kontext der RAI angesprochen, jedoch auf allgemeiner Ebene und nicht in die funktionellen Module integriert.

Die Übertragung der RAI auf andere Domänen wäre ebenso ein interessantes Forschungsfeld. Die hier vorgestellte Referenzarchitektur und das daraus resultierende integrierte Instandhaltungssystem könnte ebenso in den Bereich Health Care, Luftfahrt oder ähnliche übertragen werden. Welche Elemente angepasst werden müssen, wie sich unterstützende Systeme in dem Bereich und damit extrahierte Referenzarchitekturen von der RAI unterscheiden, wären sinnvolle Ansätze zur weiteren Forschung. Viele der Elemente, die durch ein IIS die Instandhaltung unterstützen, könnten auch Krankenschwestern, Ärzte, oder Instandhalter bzw. Techniker bei Flugzeugen darin hilfreich sein die Tätigkeiten zu verbessern. Der Kern Informationen kontextsensitiv aufzubereiten, Auszuwerten und Anzuzeigen verbunden mit einem außerordentlichen Wissensmanagement dürfte sich in weiten Teilen stark ähneln. Ebenso der Aufbau der jeweiligen Architektur. Die Übertragung von Wissen zwischen diesen Domänen wäre ein interessantes Forschungsfeld, selbst unabhängig von der Erstellung von Referenzarchitekturen in dem Bereich.

Vor dem Hintergrund dieser Arbeit wird die Entwicklung von Informationssystemen zur Instandhaltung besonders bei den Aspekten Predictive Maintenance, dezentrale Architekturen und AR ein spannendes Forschungsfeld bleiben. Die Arbeit und deren Ergebnisse ermöglichen den Blick auf die wesentlichen umfassenden Aspekte solcher Lösungen zu legen, ohne Freiheiten bei der Implementierung zu reduzieren.

Referenzen

- Abdennadher, K., Venet, P., Rojat, G., Retif, J.-M., Rosset, C. (2010).** A Real-Time Predictive-Maintenance System of Aluminum Electrolytic Capacitors Used in Uninterrupted Power Supplies. *IEEE Transactions on Industry Applications*, 46(4), 1644–1652.
- Addo, I. D., Ahamed, S. I., Yau, S. S., Buduru, A. (2014).** A Reference Architecture for Improving Security and Privacy in Internet of Things Applications. In *2014 IEEE International Conference on Mobile Services* (S. 108–115). IEEE.
- Adler, T. R. (1994).** The Innovation Process: Interpreting Customer Requirements. *IEEE Aerospace and Electronic Systems Magazine*, 9(6), 17–25.
- Adolphs, P., Bedenbender, H., Dirzus, D., Ehlich, M., Epple, U., Hankel, M., ... Wollschlaeger, M. (2015).** *Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)*. VDI/VDE Statusreport. Abgerufen von https://www.vdi.de/fileadmin/user_upload/VDI-GMA_Statusreport_Referenzarchitekturmodell-Industrie40.pdf
- Ahlemann, F., Gastl, H. (2007).** Process Model for an Empirically Grounded Reference Model Construction. In P. Fettke & P. Loos (Hrsg.), *Reference Modeling for Business Systems Analysis* (Bd. 49, S. 77–97). Hershey, PA, USA: Idea Group Inc.
- Ahuja, I. P. S., Khamba, J. S. (2008).** Total productive maintenance: literature review and directions. *International Journal of Quality & Reliability Management*, 25(7), 709–756.
- Al-Radhi, M., Heuer, J. (1995).** *Total Productive Maintenance*. München: Hanser Fachbuchverlag.
- Ali, A., Chen, Z., Lee, J. (2008).** Web-enabled platform for distributed and dynamic decision-making systems. *International Journal of Advanced Manufacturing Technology*, 38(11–12), 1260–1270.
- Anastasopoulos, M., Bayer, J., Flege, O., Gacek, C. (2000).** *A Process for Product Line Architecture Creation and Evaluation PuLSE-DSSA - Version 2.0* (Fraunhofer IESE-Report No. 038.00/E). Kaiserslautern.
- Anaya-Lara, O., Jenkins, N., McDonald, J. R. (2006).** Communications Requirements and Technology for Wind Farm Operation and Maintenance. In *International Conference on Industrial and Information Systems (ICIIS 2006)* (S. 173–178). Peradeniya, Sri Lanka: IEEE.
- Anderson, R. T., Neri, L. (1990).** *Reliability-Centered Maintenance: Management and Engineering Methods*. Dordrecht: Springer Netherlands.
- Angelov, S., Grefen, P. (2008).** An e-contracting reference architecture. *Journal of Systems and Software*, 81(11), 1816–1844.
- Angelov, S., Grefen, P., Greefhorst, D. (2009).** A Classification of Software Reference Architectures: Analyzing Their Success and Effectiveness. In *2009 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, WICSA/ECSA 2009* (S. 141–150). Cambridge, UK: IEEE.
- Angelov, S., Grefen, P., Greefhorst, D. (2012).** A framework for analysis and design of software reference architectures. *Information and Software Technology*, 54(4), 417–431.
- Angelov, S., Trienekens, J. J. M., Grefen, P. (2008).** Towards a Method for the Evaluation of Reference Architectures: Experiences from a Case. In *Proceedings of Software Architecture - Second European Conference, ECSA* (Bd. 5292 LNCS, S. 225–240). Paphos, Cyprus: Springer Berlin Heidelberg.
- Angelov, S., Trienekens, J. J. M., Grefen, P. (2014a).** *Extending and Adapting the*

Architecture Tradeoff Analysis Method for the Evaluation of Software Reference Architectures (Beta Working Paper series No. 443). Eindhoven.

- Angelov, S., Trienekens, J., Kusters, R. (2014b).** *A survey on design and usage of Software Reference Architectures* (Beta Working Paper series No. 444). Eindhoven.
- Arboleda, H., Casallas, R., Chavarriaga, J., Royer, J.-C. (2014).** Software Architecture for Product Lines. In M. C. Oussalah (Hrsg.), *Software Architecture 1* (S. 171–210). Chichester, UK: John Wiley & Sons, Ltd.
- Arnaiz, A., Emmanouilidis, C., Iung, B., Jantunen, E. (2006).** Mobile maintenance management. *Journal of International Technology and Information Management*, 15(4), 11–22.
- Aschenbrenner, D., Maltry, N., Kimmel, J., Albert, M., Scharnagl, J., Schilling, K. (2016).** ARTab - using Virtual and Augmented Reality Methods for an improved Situation Awareness for Telemaintenance. *Proceedings of the 4th IFAC Symposium on Telematics Applications*, 49(30), 204–209.
- Atteslander, P. (2008).** *Sozialforschung* (12. Aufl.). Berlin: Erich Schmidt Verlag.
- Bae, C., Kim, H., Son, Y., Lee, H., Han, S., Suh, M. (2009).** Development of a Web-based RCM system for the driverless Rubber-Tired K-AGT system. *Journal of Mechanical Science and Technology*, 23(4), 1142–1156.
- Bahga, A., Madiseti, V. K. (2012).** Analyzing massive machine maintenance data in a computing cloud. *IEEE Transactions on Parallel and Distributed Systems*, 23(10), 1831–1843.
- Baines, T. S., Lightfoot, H. W., Evans, S., Neely, A., Greenough, R., Peppard, J., ... Wilson, H. (2007).** State-of-the-art in product-service systems. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 221(10), 1543–1552.
- Bamberger, R., König, A., Pflaum, A. (2010).** *Mobile Servicewelten im Maschinenbau*. Frankfurt am Main: VDMA Verlag.
- Barkmeyer, E. J., Wallace, E. K. (2016).** *Reference Architecture for Smart Manufacturing Part 1: Functional Models* (NIST Advanced Manufacturing Series No. 300–1). Gaithersburg, MD, USA. Abgerufen von <http://nvlpubs.nist.gov/nistpubs/ams/NIST.AMS.300-1.pdf>
- Bass, L., Clements, P., Kazman, R. (2003).** *Software Architecture in Practice* (2. Aufl.). Boston, MA, USA: Addison Wesley.
- Bastos, P., Lopes, I., Pires, L. (2012).** A maintenance prediction system using data mining techniques. In *Proceedings of the World Congress on Engineering 2012 Vol III (WCE 2012)* (S. 2–7). London, U.K.
- Batory, D., Coglianese, L., Goodwin, M., Shafer, S. (1995).** Creating reference architectures. *ACM SIGSOFT Software Engineering Notes*, 20(SI), 27–37.
- Bauer, M., Boussard, M., Bui, N., De Loof, J., Magerkurth, C., Meissner, S., ... Walewski, J. W. (2013).** IoT Reference Architecture. In A. Bassi, M. Bauer, M. Fiedler, T. Kramp, R. van Kranenburg, S. Lange, & S. Meissner (Hrsg.), *Enabling Things to Talk* (S. 163–211). Berlin, Heidelberg: Springer.
- Bauernhansl, T., ten Hompel, M., Vogel-Heuser, B. (Hrsg.). (2014).** *Industrie 4.0 in Produktion, Automatisierung und Logistik*. Wiesbaden: Springer Fachmedien Wiesbaden.
- Bayer, J., Forster, T., Ganesan, D., Girard, J., John, I., Knodel, J., ... Muthig, D. (2004).** *Definition of reference architectures based on existing systems - WP 5.2, Lifecycle and Process for Family Integration* (IESE-Report No. 034.04/E). *Fraunhofer IESE-Report ...* Kaiserslautern. Abgerufen von <https://pdfs.semanticscholar.org/2a15/f323b2d898942d1b929eb3b55204b7a8fd4e.pdf>
- Becker, J. (2010).** Prozess der gestaltungsorientierten Wirtschaftsinformatik. In H. Österle, R.

- Winter, & W. Brenner (Hrsg.), *Gestaltungsorientierte Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz* (S. 13–17). Nürnberg: Infowerk AG.
- Becker, J., Delfmann, P., Knackstedt, R., Kuropka, D. (2007).** *Wissensmanagement mit Bordmitteln*. (J. Becker & R. Knackstedt, Hrsg.), *Wissensmanagement mit Referenzmodellen. Konzepte für die Anwendungssystem- und Organisationsgestaltung*. Berlin Heidelberg: Springer.
- Becker, J., Holten, R., Knackstedt, R., Niehaves, B. (2003).** *Forschungsmethodische Positionierung in der Wirtschaftsinformatik: Epistemologische, ontologische und linguistische Leitfragen* (Arbeitsberichte des Instituts für Wirtschaftsinformatik No. 93). Münster. Abgerufen von <http://hdl.handle.net/10419/59562>
- Bengtsson, M. (2004).** *Condition Based Maintenance Systems: An Investigation of Technical Constituents and Organizational Aspects*. Mälardalen University. Abgerufen von <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.122.2614&rep=rep1&type=pdf>
- Bergey, J., Cohen, S., Fisher, M., Campbell, G., Jones, L., Krut, R., ... Soule, A. (1999).** *Fourth DoD Product Line Practice Workshop Report* (Technical Report No. CMU/SEI-2001-TR-017 ESC-TR-2001-017). Pittsburgh, USA.
- Berkovich, M. (2011).** *Requirements Engineering für IT-gestützte Product Service Systems*. Technische Universität München. Abgerufen von <http://link.springer.com/10.1007/s12599-011-0192-2>
- Berkovich, M., Leimeister, J. M., Kremer, H. (2009).** An Empirical Exploration of Requirements Engineering for Hybrid Products. In S. Newell (Hrsg.), *Information systems in a globalising world: challenges, ethics and practices; ECIS 2009, 17th European Conference on Information Systems* (S. 2570–2582). Verona: Università di Verona, Facoltà di Economia, Dipartimento de Economia Aziendale.
- Berkovich, M., Leimeister, J. M., Kremer, H. (2010).** Ein Bezugsrahmen für Requirements Engineering hybrider Produkte. In M. Schumann, L. M. Kolbe, M. H. Breitner, & A. Frerichs (Hrsg.), *Multikonferenz Wirtschaftsinformatik 2010* (S. 2017–2029). Göttingen: Universitätsverlag Göttingen.
- Bichler, M. (2006).** Design Science in Information Systems Research. *Wirtschaftsinformatik*, 2(48), 133–142.
- Bienzeisler, B., Schletz, A., Gahle, A.-K. (2014).** Industrie 4.0 Ready Services Technologietrends 2020. Abgerufen 21. August 2015, von <http://wiki.iao.fraunhofer.de/images/studien/industrie-4-0-ready-service.pdf>
- Blanchard, B. S. (2004).** *Logistics Engineering and Management* (6. Aufl.). Upper Saddle River, New Jersey, USA: Prentice Hall.
- Bley, K., Leyh, C. (2016).** Status Quo der Digitalisierung deutscher Industrieunternehmen—Eine Studie ausgewählter Unternehmen. In V. Nissen, D. Stelzer, S. Straßburger, & D. Fischer (Hrsg.), *Tagungsband zur Multikonferenz Wirtschaftsinformatik 2016 (MKWI 2016)* (S. 1651–1662). Ilmenau: Universitätsverlag Ilmenau.
- Bohoris, G. A., Vamvalis, C., Trace, W., Ignatiadou, K. (1995).** TPM implementation in Land-Rover with the assistance of a CMMS. *Journal of Quality in Maintenance Engineering*, 1(4), 3–16.
- Bruns, R., Dunkel, J. (2010).** Event-Driven Architecture und Complex Event Processing im Überblick. In *Event-Driven Architecture* (1. Aufl., S. 47–82). Berlin, Heidelberg: Springer.
- Campos, J., Jantunen, E., Prakash, O. (2007).** Development of a maintenance system based on web and mobile technologies. *Journal of International Technology and Information Management*, 16(4), 1.
- Campos, J., Jantunen, E., Prakash, O. (2009).** A web and mobile device architecture for mobile e-maintenance. *International Journal of Advanced Manufacturing Technology*,

45(1–2), 71–80.

- Campos, J., Prakash, O., Jantunen, E. (2008).** A mobile Maintenance Decision Support System. *maintenance & asset management*, 23(2), 42–48.
- Candell, O., Karim, R., Parida, A. (2011).** Development of information system for e-maintenance solutions within the aerospace industry. *International Journal of Performability Engineering*, 7(6), 583–592.
- Candell, O., Karim, R., Söderholm, P. (2009).** eMaintenance—Information logistics for maintenance support. *Robotics and Computer-Integrated Manufacturing*, 25(6), 937–944.
- Cannata, A., Karnouskos, S., Taisch, M. (2009).** Dynamic e-maintenance in the era of SOA-ready device dominated industrial environments. In D. Kiritsis, C. Emmanouilidis, A. Koronios, & J. Mathew (Hrsg.), *Proceedings of the 4th World Congress on Engineering Asset Management (WCEAM 2009)*. Athens, Greece: Springer-Verlag London.
- Carnero, M. C., Novés, J. L. (2006).** Selection of computerised maintenance management system by means of multicriteria methods. *Production Planning & Control*, 17(4), 335–354.
- Carroll, D., Rose, M., Stritapan, V. (2013).** *Mobile Security Reference Architecture* (Digital Government Strategy). Abgerufen von <https://cio.gov/wp-content/uploads/downloads/2013/05/Mobile-Security-Reference-Architecture.pdf>
- Center for Intelligent Maintenance Systems. (o. J.).** Center for Intelligent Maintenance Systems. Abgerufen 13. Dezember 2016, von <http://www.imscenter.net/>
- Chand, G., Shirvani, B. (2000).** Implementation of TPM in cellular manufacture. *Journal of Materials Processing Technology*, 103(1), 149–154.
- Chen, I. J., Popovich, K. (2003).** Understanding customer relationship management (CRM): People, process and technology. *Business Process Management Journal*, 9(5), 672–688.
- Cheng, Z., Jia, X., Wang, L., Bai, Y. (2010).** A Framework for the Case-Based and Model-Based RCM Analysis. In *2010 International Conference on E-Product E-Service and E-Entertainment* (S. 1–4). Henan, China: IEEE.
- Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., Bone, M. (2009).** The Concept of Reference Architectures. *Systems Engineering*, 14(3), 14–27.
- Cockburn, A. (2002).** *Writing Effective Use Cases* (1. Aufl.). Boston [u.a.], USA: Addison-Wesley Professional.
- Crabb, C. (2014).** A Reference Stack for PHM Architectures. In *Annual conference of the Prognostics and Health Management Society* (S. 1–14). Fort Worth, Texas, USA.
- Crespo Marquez, A., Gupta, J. N. D. (2006).** Contemporary maintenance management: Process, framework and supporting pillars. *Omega*, 34(3), 313–326.
- Daeuble, G., Oezcan, D., Niemoeller, C., Fellmann, M., Nuettgens, M. (2015a).** Design of User-Oriented Mobile Service Support Systems - Analyzing the Eligibility of a Use Case Catalog to Guide System Development. In O. Thomas & F. Teuteberg (Hrsg.), *Wirtschaftsinformatik Proceedings* (S. 149–163). Osnabrück: Universität Osnabrück.
- Daeuble, G., Oezcan, D., Niemoeller, C., Fellmann, M., Nuettgens, M., Thomas, O. (2015b).** Information Needs of the Mobile Technical Customer Service -- A Case Study in the Field of Machinery and Plant Engineering. In *2015 48th Hawaii International Conference on System Sciences* (S. 1018–1027). Kauai, HI, USA: IEEE.
- Daneels, A., Salter, W. (1999).** What Is Scada? In D. Bulfone & A. Daneels (Hrsg.), *Accelerator and Large Experimental Physics Control Systems -Proceedings of the 7th international conference on accelerator and large experimental physics control systems* (S. 339–343). Trieste, Italy.
- Dey, P. K. (2004).** Decision Support System for Inspection and Maintenance: A Case Study of Oil Pipelines. *IEEE Transactions on Engineering Management*, 51(1), 47–56.

- DIN 13306:2015-09. (2015).** *Instandhaltung - Begriffe der Instandhaltung; Deutsche und Englische Fassung prEN 13306:2015 (Entwurf)*. (DIN Deutsches Institut für Normung e.V., Hrsg.). Berlin: Beuth Verlag.
- DIN 31051:2012-09. (2012).** *Grundlagen der Instandhaltung*. (DIN Deutsches Institut für Normung e.V., Hrsg.). Berlin: Beuth Verlag.
- DIN IEC 60050-351:2014-09. (2014).** *Internationales Elektrotechnisches Wörterbuch - Teil 351: Leittechnik (IEC 60050-351:2013)*. (DIN Deutsches Institut für Normung e.V., Hrsg.). Berlin: Beuth Verlag.
- Dix, A. (1994).** Computer Supported Cooperative Work: A Framework. In D. Rosenburg & C. Hutchison (Hrsg.), *In Design Issues in CSCW* (S. 9–26). London: Springer.
- Döring, N., Bortz, J. (2016).** *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften* (5.). Berlin, Heidelberg: Springer.
- Dumitrescu, R., Gausemeier, J., Kühn, A., Luckey, M., Plass, C., Schneider, M., Westermann, T. (2015).** *Auf dem Weg zu Industrie 4.0: Erfolgsfaktor Referenzarchitektur*. Paderborn. Abgerufen von http://www.its-owl.de/fileadmin/PDF/Informationsmaterialien/2015-Auf_dem_Weg_zu_Industrie_4.0_Erfolgsfaktor_Referenzarchitektur.pdf
- Ebke, M., Däuble, G. (2015).** Anforderungen an mobile Assistenzsysteme zur Unterstützung von Geschäftsprozessen - Literaturrecherche und empirische Untersuchung des Technischen Kundendienstes. In O. Thomas & F. Teuteberg (Hrsg.), *Wirtschaftsinformatik Proceedings 2015* (S. 46–60). Osnabrück: Universität Osnabrück.
- Efthymiou, K., Papakostas, N., Mourtzis, D., Chryssolouris, G. (2012).** On a Predictive Maintenance Platform for Production Systems. *Procedia CIRP*, 3(1), 221–226.
- Eigner, M., Roubanov, D., Zafirov, R. (Hrsg.). (2014).** *Modellbasierte virtuelle Produktentwicklung*. Berlin, Heidelberg: Springer.
- Eklund, U., Askerdal, Ö., Granholm, J., Alminger, A., Axelsson, J. (2005).** Experience of introducing reference architectures in the development of automotive electronic systems. *ACM SIGSOFT Software Engineering Notes*, 30(4), 1–6.
- Emmanouilidis, C., Jantunen, E., Gilabert, E., Arnaiz, A., Starr, A. (2011).** e-Maintenance update: the road to success for modern industry. In M. Singh, R. B. K. N. Rao, & J. P. Liyanage (Hrsg.), *Proceedings of the 24th International Congress on Condition Monitoring and Diagnostics Engineering Management* (S. 423–433). Stavanger, Norway: COMADEM International.
- Emmanouilidis, C., Pistofidis, P., Fournaris, A., Bevilacqua, M., Durazo-Cardenas, I., Botsaris, P. N., ... Starr, A. G. (2016).** Context-based and human-centred information fusion in diagnostics. *IFAC-PapersOnLine*, 49(28), 220–225.
- Endig, M., Freund, C., Götze, J., Hänsch, K., Kumetz, S., Ryll, F., Termath, W. (2010).** *Instandhaltung technischer Systeme*. (M. Schenk, Hrsg.). Berlin, Heidelberg: Springer.
- Enste, U., Mahnke, W. (2011).** OPC Unified Architecture. *at - Automatisierungstechnik*, 59(7), 397–404.
- Fay, A. (2009).** Anforderungen an Fern-Service-Systeme. *atp edition - Automatisierungstechnische Praxis*, 51(12), 38–43.
- Fellmann, M., Hucke, S., Breitschwerdt, R., Thomas, O., Blinn, N., Schlicker, M. (2011).** Informationssystemarchitekturen zur Unterstützung technischer Kundendienstleistungen. In *Wirtschaftsinformatik Proceedings 2011* (S. 252–161). Zürich, Schweiz.
- Fellmann, M., Özcan, D., Matijacic, M., Däuble, G., Schlicker, M., Thomas, O., Nüttgens, M. (2013).** Towards a Mobile Technical Customer Service Support Platform. In F. Daniel, G. A. Papadopoulos, & P. Thiran (Hrsg.), *Mobile Web Information Systems* (Bd. 8093 LNCS, S. 296–299). Berlin Heidelberg: Springer.
- Fernandez, E. B., Monge, R., Hashizume, K. (2016).** Building a security reference

- architecture for cloud systems. *Requirements Engineering*, 21(2), 225–249.
- Fettke, P., Loos, P. (2003).** Multiperspective Evaluation of Reference Models – Towards a Framework. In M. A. Jeusfeld & P. Ó. (Hrsg.), *Conceptual Modeling for Novel Application Domains. ER 2003. Lecture Notes in Computer Science vol. 2814* (S. 80–91). Berlin Heidelberg: Springer.
- Fettke, P., Loos, P. (2004a).** *Entwicklung eines Bezugsrahmens zur Evaluierung von Referenzmodellen - Langfassung eines Beitrages* (Working Papers of the Research Group Information Systems & Management No. 20). Mainz. Abgerufen von <https://dnb.info/972453296/34>
- Fettke, P., Loos, P. (2004b).** Referenzmodellierungsforschung. *Wirtschaftsinformatik*, 46(5), 331–340.
- Fissler, D., Reiners, R. (2015).** Prozessorientierte Überwachung in der Produktion. In D. Cunningham, P. Hofstedt, K. Meer, & I. Schmitt (Hrsg.), *INFORMATIK2015 LectureNotes in Informatics (LNI)* (S. 917–927). Bonn: Gesellschaft für Informatik.
- Fitouhi, M. C., Nourelfath, M. (2012).** Integrating noncyclical preventive maintenance scheduling and production planning for a single machine. *International Journal of Production Economics*, 136(2), 344–351.
- Fleischmann, H., Kohl, J., Franke, J. (2016a).** A modular web framework for socio-CPS-based condition monitoring. In *2016 IEEE World Conference on Factory Communication Systems (WFCS)* (S. 1–8). Aveiro, Portugal: IEEE.
- Fleischmann, H., Kohl, J., Franke, J., Reidt, A., Duchon, M., Krcmar, H. (2016b).** Improving maintenance processes with distributed monitoring systems. In *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)* (S. 377–382). Poitiers, France: IEEE.
- Fonseca, I., Farinha, T., Barbosa, M. (2008).** A computer system for predictive maintenance of wind generators. In N. E. Mastorakis, V. Mladenov, Z. Bojkovic, D. Simian, S. Kartalopoulos, A. Varonides, ... K. L. Man (Hrsg.), *ICCOMP'08 Proceedings of the 12th WSEAS international conference on Computers* (S. 928–933). Heraklion, Greece.
- Fowler, M. (2003).** Patterns. *IEEE Software*, 20(2), 56–57.
- Fowler, M. (2010).** *Analysis Patterns: Reusable Object Models* (21. Aufl.). Boston, MA, USA: Addison-Wesley.
- Frank, U. (2000).** Evaluation von Artefakten in der Wirtschaftsinformatik. In I. Häntschel & L. J. Heinrich (Hrsg.), *Evaluation und Evaluationsforschung in der Wirtschaftsinformatik* (S. 35–48). München, Wien: Oldenbourg.
- Frank, U. (2007).** Ein Vorschlag zur Konfiguration von Forschungsmethoden. In F. Lehner & S. Zelewski (Hrsg.), *Wissenschaftstheoretische Fundierung und wissenschaftliche Orientierung der Wirtschaftsinformatik* (S. 158–185). Berlin: GITO.
- Gabbar, H. a., Yamashita, H., Suzuki, K., Shimada, Y. (2003).** Computer-aided RCM-based plant maintenance management system. *Robotics and Computer-Integrated Manufacturing*, 19(5), 449–458.
- Gajewski, T. (2004).** *Referenzmodell zur Beschreibung der Geschäftsprozesse von After-Sales-Dienstleistungen unter besonderer Berücksichtigung des Mobile Business*. Heinz Nixdorf Institut, Universität Paderborn, Paderborn.
- Galar, D. (2014).** Context-driven Maintenance: an eMaintenance approach. *Management Systems in Production Engineering*, 3(15), 112–120.
- Gallagher, B. P. (2000).** *Using the Architecture Tradeoff Analysis Method SM to Evaluate a Reference Architecture : A Case Study* (Technical Note No. CMU/SEI-2000-TN-007).
- Galster, M. (2015).** Software Reference Architectures. In *Proceedings of the 1st International Workshop on Exploring Component-based Techniques for Constructing Reference Architectures - CobRA '15* (S. 5–8). New York, New York, USA: ACM Press.

- Galster, M., Angelov, S. (2015).** Understanding the Use of Reference Architectures in Agile Software Development Projects. In D. Weyns, R. Mirandola, & I. Crnkovic (Hrsg.), *Software Architecture 9th European Conference, ECSA 2015, Dubrovnik/Cavtat, Croatia, September 7-11, 2015. Proceedings* (S. 268–276). Berlin Heidelberg: Springer International Publishing.
- Galster, M., Avgeriou, P. (2011).** Empirically-grounded reference architectures. In *Proceedings of the joint ACM SIGSOFT conference -- QoSA and ACM SIGSOFT symposium -- ISARCS on Quality of software architectures -- QoSA and architecting critical systems -- ISARCS - QoSA-ISARCS '11* (S. 153). New York, New York, USA: ACM Press.
- Galster, M., Avgeriou, P., Tofan, D. (2012).** Constraints for the design of variability-intensive service-oriented reference architectures – An industrial case study. *Information and Software Technology*, 55(2), 428–441.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995).** *Design patterns: elements of reusable object-oriented software*. Boston, MA, USA: Addison-Wesley.
- Gläser, J., Laudel, G. (2010).** *Experteninterviews und qualitative Inhaltsanalyse: als Instrumente rekonstruierender Untersuchungen* (4. Aufl.). Wiesbaden: VS Verlag für Sozialwissenschaften.
- Goh, K. m., Tjahjono, B., Baines, T., Subramaniam, S. (2006).** A Review of Research in Manufacturing Prognostics. In *2006 IEEE International Conference on Industrial Informatics (INDIN)* (S. 417–422). Singapore, Singapore: IEEE.
- Gómez Fernández, J. F., Crespo Márquez, A. (2009).** Framework for implementation of maintenance management in distribution network service providers. *Reliability Engineering & System Safety*, 94(10), 1639–1649.
- Götzer, K., Schneiderath, U., Maier, B., Komke, T. (2004).** *Dokumenten-Management - Informationen im Unternehmen effizient nutzen* (3. Aufl.). Heidelberg: dpunkt.verlag.
- Graaf, B., van Dijk, H., van Deursen, A. (2005).** Evaluating an Embedded Software Reference Architecture - Industrial Experience Report -. In *Ninth European Conference on Software Maintenance and Reengineering* (S. 354–353). Manchester, UK: IEEE.
- Gregor, S. (2006).** The Nature of Theory in Information Systems. *MIS Quarterly*, 30(3), 611–642.
- Gregor, S., Jones, D. (2007).** The Anatomy of a Design Theory. *Journal of the Association for Information Systems*, 8(5), 312–335.
- Grosskurth, A., Godfrey, M. W. (2005).** A reference architecture for Web browsers. In *21st IEEE International Conference on Software Maintenance (ICSM'05)* (S. 661–664). Budapest, Hungary: IEEE.
- Grundnig, G., Raschhofer, S. (2010).** Erhöhung der Verfügbarkeit durch Einsatz des Diagnosesystems FDS. *SIGNAL + DRAHT*, 1+2(102), 29–33.
- Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. (2013).** Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645–1660.
- Guessi, M., Oquendo, F., Nakagawa, E. Y. (2014).** Variability viewpoint to describe reference architectures. In *Proceedings of the WICSA 2014 Companion* (S. 1–6). New York, New York, USA: ACM Press.
- Güntner, G., Benisch, M., Dankl, A., Isopp, J. (2015).** *Roadmap der Instandhaltung 4.0*. Salzburg, Österreich.
- Haarman, M., Delahay, G. (2013).** Value Driven Maintenance. *maintenance & asset management*, 13(13), 14–15.
- Haddaway, N. R., Collins, A. M., Coughlin, D., Kirk, S. (2015).** The Role of Google Scholar in Evidence Reviews and Its Applicability to Grey Literature Searching. *PLOS ONE*,

10(9), 1–17.

- Haft, M., Humm, B., Siedersleben, J. (2005).** The Architect's Dilemma – Will Reference Architectures Help? In *Quality of Software Architectures and Software Quality* (S. 106–122). Berlin, Heidelberg: Springer.
- Hameed, Z., Ahn, S. H., Cho, Y. M. (2010).** Practical aspects of a condition monitoring system for a wind turbine with emphasis on its design, system architecture, testing and installation. *Renewable Energy*, 35(5), 879–894.
- Hänsch, K., Endig, M. (2010).** Informationsmanagement in der Instandhaltung. In M. Schenk (Hrsg.), *Grundlagen der Instandhaltung* (S. 230–287). Berlin, Heidelberg: Springer.
- Harms, H. (2007).** Datenmanagementsystem für den Teleservice bei mobilen Arbeitsmaschinen. *Landtechnik*, 62(5), 328–329.
- Hausladen, I., Bechheim, C. (2004).** E-maintenance platform as a basis for business process integration. In *2nd IEEE International Conference on Industrial Informatics, 2004. INDIN '04.* (S. 46–51). Berlin, Germany: IEEE.
- Heinrich, B., Linke, P., Glöckler, M. (2015).** *Grundlagen Automatisierung*. Wiesbaden: Springer Vieweg.
- Heinrich, L. J. (2005).** Forschungsmethodik einer Integrationsdisziplin: Ein Beitrag zur Geschichte der Wirtschaftsinformatik. *NTM International Journal of History and Ethics of Natural Sciences, Technology and Medicine*, 13(2), 104–117.
- Hevner, A., Chatterjee, S. (2010).** Design Science Research in Information Systems. In *Design Research in Information Systems. Integrated Series in Information Systems, vol 22.* Springer (S. 9–22). Boston, MA: Springer.
- Hevner, A. R. (2007).** A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, 19(2), 87–92.
- Hevner, A. R., March, S. T., Park, J., Ram, S. (2004).** Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75–105.
- Himmler, F., Amberg, M. (2013).** Die Digitale Fabrik - eine Literaturanalyse. In *Wirtschaftsinformatik Proceedings 2013* (S. 165–179). Leipzig, Germany.
- Holmberg, K. (2005).** Maintenance, Condition Monitoring and Diagnostics. In S. Lahdelma & K. Palokangas (Hrsg.), *Maintenance, condition monitoring and diagnostics : proceedings of the 2nd international seminar* (S. 1–14). Oulu, Finland: Pohto.
- Holmberg, K., Adgar, A., Arnaiz, A., Jantunen, E., Mascolo, J., Mekid, S. (Hrsg.). (2010).** *E-maintenance*. London: Springer-Verlag London.
- Hora, M. E. (1987).** The unglamorous game of managing maintenance. *Business Horizons*, 30(3), 67–75.
- Horbach, S., Trommler, U. (2017).** Konzeption und Umsetzung eines Ressourcen-Cockpits für die Instandhaltung. In P. A. C. Bullinger-Hoffmann (Hrsg.), *Abschlussveröffentlichung S-CPS: Ressourcen-Cockpit für Sozio-Cyber-Physische Systeme* (S. 91–112). Chemnitz: Verlag aw&I - Wissenschaft und Praxis.
- Horváth, P. (2017).** Geschäftsmodellinnovationen durch Digitalisierung – Neue Herausforderungen an den Controller. In *Technologie, Strategie und Organisation* (S. 113–125). Wiesbaden: Springer Fachmedien Wiesbaden.
- Houy, C., Reiter, M., Fettke, P., Loos, P. (2012).** *Prozessorientierter Web-2.0-basierter integrierter Telekommunikationsservice (PROWIT) – Anforderungserhebung, Konzepte, Implementierung und Evaluation* (IW-Heft No. 196). Saarbrücken.
- Huang, Y., Gardoni, M., Coulibaly, A. (2009).** A Decision Support System Designed For Personalized Maintenance Recommendation Based on Proactive Maintenance. In U. Norell Bergendahl, M.; Grimheden, M.; Leifer, L.; Skogstad, P.; Lindemann (Hrsg.), *DS 58-4: Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 4, Product and Systems Design, Palo Alto, CA, USA, 24.-27.08.2009* (S. 191–202).

Palo Alto, USA.

- Huber, D., Kaiser, T. (2015).** Wie das Internet der Dinge neue Geschäftsmodelle ermöglicht. *HMD Praxis der Wirtschaftsinformatik*, 52(5), 681–689.
- Hung, M. H., Chen, K. Y., Ho, R. W., Cheng, F. T. (2003).** Development of an e-Diagnostics/Maintenance framework for semiconductor factories with security considerations. *Advanced Engineering Informatics*, 17(3–4), 165–178.
- Hurley, S., Wang, J. . (2014).** *IBM System x Reference Architecture for Hadoop: IBM InfoSphere BigInsights Reference Architecture* (IBM Redbooks). Poughkeepsie, NY, USA. Abgerufen von <http://www.redbooks.ibm.com/redpapers/pdfs/redp5009.pdf>
- Hwang, W. T., Tien, S. W., Shu, C. M. (2007).** Building an Executive Information System for Maintenance Efficiency in Petrochemical Plants—An Evaluation. *Process Safety and Environmental Protection*, 85(2), 139–146.
- Iung, B. (2003).** From remote maintenance to MAS-based e-maintenance of an industrial process. *Journal of Intelligent Manufacturing*, 14(1), 59–82.
- Iung, B., Levrat, E., Marquez, A. C., Erbe, H. (2009).** Conceptual framework for e-Maintenance: Illustration by e-Maintenance technologies and platforms. *Annual Reviews in Control*, 33(2), 220–229.
- Jin, X., Li, L., Ni, J. (2009).** Option model for joint production and preventive maintenance system. *International Journal of Production Economics*, 119(2), 347–353.
- Johnston, D. C. (2002).** Measuring RCM implementation. In *Annual Reliability and Maintainability Symposium. 2002 Proceedings* (S. 511–515). Seattle, WA, USA: IEEE.
- Jones, R. B. (1995).** *Risk-based Maintenance: a reliability centered approach*. New York, New York, USA: Routledge.
- Kaiser, R. (2014).** *Qualitative Experteninterviews*. Wiesbaden: Springer VS.
- Kamsu-Foguem, B., Mathieu, Y. (2014).** Software architecture knowledge for intelligent light maintenance. *Advances in Engineering Software*, 67(1), 125–135.
- Kamsu-Foguem, B., Noyes, D. (2013).** Graph-based reasoning in collaborative knowledge management for industrial maintenance. *Computers in Industry*, 64(8), 998–1013.
- Kans, M. (2008).** An approach for determining the requirements of computerised maintenance management systems. *Computers in Industry*, 59(1), 32–40.
- Karim, R. (2008).** *A service-oriented approach to e-maintenance of complex technical systems*. Luleå University of Technology. Abgerufen von <http://epubl.ltu.se/1402-1544/2008/58/LTU-DT-0858-SE.pdf>
- Karim, R., Söderholm, P., Candell, O. (2009).** Development of ICT-based maintenance support services. *Journal of Quality in Maintenance Engineering*, 15(2), 127–150.
- Karlsson, J., Wohlin, C., Regnell, B. (1998).** An evaluation of methods for prioritizing software requirements. *Information and Software Technology*, 39(14–15), 939–947.
- Karow, M., Pfeiffer, D., Räckers, M. (2008).** Empirical-Based Construction of Reference Models in Public Administrations. In M. Bichler, T. Hess, H. Krcmar, U. Lechner, F. Matthes, A. Picot, ... P. Wolf (Hrsg.), *Proceedings of the Multikonferenz Wirtschaftsinformatik 2008: München, Germany* (S. 1613–1624). Berlin: GITO-Verlag.
- Karray, M.-H., Chebel-Morello, B., Lang, C., Zerhouni, N. (2011).** A component based system for s-maintenance. In *Proceedings 2011 9th IEEE International Conference on Industrial Informatics (INDIN 2011)* (S. 519–526). Caparica, Lisbon, Portugal: IEEE.
- Kassner, L., Hirmer, P., Wieland, M., Steimle, F., Königsberger, J., Mitschang, B. (2017).** The Social Factory: Connecting People, Machines and Data in Manufacturing for Context-Aware Exception Escalation. In *Proceedings of the 50th Hawaii International Conference on System Sciences / 2017* (S. 1673–1682). Waikoloa, HI, USA.
- Kelly, A. (1997).** *Maintenance Organization and Systems*. Oxford: Butterworth-Heinemann.

- Klos, S., Patalas-Maliszewska, J. (2013).** The Impact of ERP on Maintenance Management. *Management and Production Engineering Review*, 4(3), 15–25.
- Koç, M., Ni, J., Lee, J., Bandyopadhyay, P. (2005).** Introduction to e- Manufacturing. In R. Zurawski (Hrsg.), *The Industrial Information Technology Handbook* (S. 1541–1549). Boca Raton, Florida, USA: CRC Press.
- Kruchten, P. (1995).** Architectural Blueprints—The “4+1” View Model of Software Architecture. *IEEE Software*, 12(6), 42–50.
- Kühl, I., Fay, A. (2010).** Towards common concepts of remote services. In *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2010*. Bilbao, Spain: IEEE.
- Kuhn, A., Schuh, G., Stahl, B. (2006).** *Nachhaltige Instandhaltung*. Frankfurt am Main: VDMA Verlag.
- Kumar, U., Galar, D., Parida, A., Stenström, C., Berges, L. (2013).** Maintenance performance metrics: A state-of-the-art review. *Journal of Quality in Maintenance Engineering*, 19(3), 233–277.
- Lachenmaier, J., Lasi, H., Kemper, H. (2015).** Entwicklung und Evaluation eines Informationsversorgungskonzepts für die Prozess- und Produktionsplanung im Kontext von Industrie 4.0. In O. Thomas & F. Teuteberg (Hrsg.), *Proceedings der 12. Internationalen Tagung Wirtschaftsinformatik (WI 2015)* (S. 1–15). Osnabrück.
- Laplante, P. A. (2007).** *What Every Engineer Should Know About Software Engineering*. Boca Raton: CRC Press.
- Lauth, H.-J., Pickel, G., Pickel, S. (2015).** *Methoden der vergleichenden Politikwissenschaft* (2. Aufl.). Wiesbaden: Springer Fachmedien Wiesbaden.
- Lee, J., Bagheri, B., Kao, H.-A. (2015).** A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3(January), 18–23.
- Lee, J., Chen, Y., Al-Atat, H., AbuAli, M., Lapira, E. (2009).** A systematic approach for predictive maintenance service design: methodology and applications. *International Journal of Internet Manufacturing and Services*, 2(1), 76–94.
- Lee, J., Ghaffari, M., Elmeligy, S. (2011).** Self-maintenance and engineering immune systems: Towards smarter machines and manufacturing systems. *Annual Reviews in Control*, 35(1), 111–122.
- Lee, J., Lapira, E. (2011).** TPM Gets Smart. *Manufacturing Engineering*, (June), 70–75.
- Lee, J., Lee, M., Lee, S., Oh, S., Kim, B., Nam, S., Jang, J. (2013).** Development of computerized facility maintenance management system based on reliability centered maintenance and automated data gathering. *International Journal of Control and Automation*, 6(1), 1–12.
- Lee, J., Ni, J., Djurdjanovic, D., Qiu, H., Liao, H. (2006).** Intelligent prognostics tools and e-maintenance. *Computers in Industry*, 57(6), 476–489.
- Lee, S., Akin, Ö. (2011).** Augmented reality-based computational fieldwork support for equipment operations and maintenance. *Automation in Construction*, 20(4), 338–352.
- Levrat, E., Iung, B. B. (2007).** TELMA: A full e-maintenance platform. In *Second World Congress on Engineering Asset Management and the Fourth International Conference on Condition Monitoring, WCEAM/CM 2007* (S. CDR0M). Harrogate, UK.
- Levrat, E., Iung, B., Crespo Marquez, A. (2008).** E-maintenance: review and conceptual framework. *Production Planning & Control*, 19(4), 408–429.
- Li, Y., Chun, L., Ching, Y., A.N. (2005).** An Agent-based Platform for Web-enabled Equipment Predictive Maintenance. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)* (S. 132–135). Compiegne, France: IEEE.

- Lin, S.-W., Miller, B., Durand, J., Joshi, R., Didier, P., Chigani, A., ... Witten, B. (2015).** *Industrial Internet Reference Architecture* (Technical Architecture No. tr. 001). *Technical Report*. Industrial Internet Consortium. Abgerufen von <http://www.iiconsortium.org/IIRA.htm>
- Ljungberg, Ö. (1998).** Measurement of overall equipment effectiveness as a basis for TPM activities. *International Journal of Operations & Production Management*, 18(5), 495–507.
- Maier, R. (2007).** *Knowledge Management Systems* (3. Aufl.). Berlin, Heidelberg: Springer.
- March, S. T., Smith, G. F. (1995).** Design and natural science research on information technology. *Decision Support Systems*, 15(4), 251–266.
- Marquez, A., Iung, B. (2008).** A review of e-maintenance capabilities and challenges. *Journal of Systemics, Cybernetics and Informatics*, 6(1), 62–66.
- Martínez-Fernández, S., Ayala, C., Franch, X., Ameller, D. (2013).** A Framework for Software Reference Architecture Analysis and Review. In M. Solari Buela & A. C. Dias Neto (Hrsg.), *CIBSE 2013 | X Workshop Latinoamericano Ingeniería de Software Experimental | ESELAW 2013* (S. 89–102). Montevideo, Uruguay.
- Martínez-Fernández, S., Ayala, C. P., Franch, X., Marques, H. M. (2013a).** Benefits and drawbacks of reference architectures. In *Software Architecture. ECSA 2013. Lecture Notes in Computer Science*, vol 7957 (S. 307–310). Berlin, Heidelberg: Springer.
- Martínez-Fernández, S., Ayala, C. P., Franch, X., Marques, H. M. (2013b).** REARM: A Reuse-Based Economic Model for Software Reference Architectures. In J. Favaro & M. Morisio (Hrsg.), *Safe and Secure Software Reuse. ICSR 2013. Lecture Notes in Computer Science*, vol 7925 (S. 97–112). Berlin, Heidelberg: Springer.
- Martínez-Fernández, S., Medeiros Dos Santos, P. S., Ayala, C. P., Franch, X., Travassos, G. H. (2015).** Aggregating Empirical Evidence about the Benefits and Drawbacks of Software Reference Architectures. In *2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (S. 154–163). Beijing, China: IEEE.
- Matijacic, M., Krammler, F., Nüttgens, M., Fellmann, M., Özcan, D., Thomas, O. (2013).** Elicitation and Consolidation of Requirements for Mobile Technical Customer Services Support Systems. In *Proceedings of the 34th International Conference on Information Systems (ICIS), Milan (Italy), December 15-18, 2013* (S. 1–16). Milan, Italy.
- Matyas, K. (2002).** Ganzheitliche Optimierung durch individuelle Instandhaltungsstrategien. *Industrie Management*, 18(2), 13–16.
- Mayring, P. (2010a).** Qualitative Inhaltsanalyse. In G. Mey & K. Mruck (Hrsg.), *Handbuch Qualitative Forschung in der Psychologie* (S. 601–613). Wiesbaden: Springer VS.
- Mayring, P. (2010b).** *Qualitative Inhaltsanalyse - Grundlagen und Techniken* (12. Aufl.). Weinheim, Basel: Beltz.
- McArthur, S. D. J., Strachan, S. M., Jahn, G. (2004).** The design of a multi-agent transformer condition monitoring system. *IEEE Transactions on Power Systems*, 19(4), 1845–1852.
- Meuser, M., Nagel, U. (2002).** ExpertInneninterviews — vielfach erprobt, wenig bedacht. In B. A., L. B., & M. W. (Hrsg.), *Das Experteninterview* (S. 71–93). Wiesbaden: VS Verlag für Sozialwissenschaften.
- Meyr, H., Wagner, M., Rohde, J. (2005).** Structure of Advanced Planning Systems. In H. Stadtler (Hrsg.), *Supply Chain Management and Advanced Planning* (S. 109–115). Berlin, Heidelberg: Springer.
- Mikler, J. (2015).** *On Improvement Of Maintenance Function - A Reference Model And Improvement Methodology*. Royal Institute of Technology Stockholm, Stockholm, Sweden.
- Mödinger, J. (2006).** *Entwicklung eines Bewertungs- und Entscheidungsunterstützungssystems*

für eine nachhaltige regionale Gewässbewirtschaftung (Mitteilungen No. 156). *Mitteilungen des Instituts für Wasserbau, Universität Stuttgart*. Eigenverlag des Instituts für Wasserbau der Universität Stuttgart, Stuttgart.

- Monnin, M., Voisin, A., Leger, J., Iung, B. (2011).** Fleet-wide health management architecture. In *Annual Conference of the Prognostics and Health Management Society 2011* (Bd. 1, S. 1–8). Montreal, Quebec, Canada.
- Moore, W. J., Starr, A. G. (2006).** An intelligent maintenance system for continuous cost-based prioritisation of maintenance activities. *Computers in Industry*, 57(6), 595–606.
- Moubray, J. (1996).** *RCM - Die Hohe Schule der Zuverlässigkeit von Produkten und Systemen*. Landsberg: Verlag Moderne Industrie.
- Moubray, J. (1997).** *Reliability-centred Maintenance* (2. Aufl.). Oxford, UK: Butterworth-Heinemann.
- Mouzoune, A., Taibi, S. (2013).** Towards an intelligence based conceptual framework for e-maintenance. In *2013 8th International Conference on Intelligent Systems: Theories and Applications (SITA)* (S. 1–8). Rabat, Morocco: IEEE.
- Muller, A., Crespo Marquez, A., Iung, B. B. (2008a).** On the concept of e-maintenance: Review and current research. *Reliability Engineering and System Safety*, 93(8), 1165–1187.
- Muller, A., Suhner, M. C., Iung, B. (2008b).** Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system. *Reliability Engineering and System Safety*, 93(2), 234–253.
- Muller, G. (2007).** *A reference architecture primer*. Kongsberg, Norway. Abgerufen von <http://www.gaudisite.nl/referencearchitectureprimerslides.pdf>
- Muller, G. (2008).** Right Sizing Reference Architectures; How to provide specific guidance with limited information. In *INCOSE International Symposium* (Bd. 18, S. 2047–2054). Utrecht, Netherlands.
- Muller, G., van de Laar, P. (2009).** Researching Reference Architectures and their relationship with frameworks, methods, techniques, and tools. In *7th Annual Conference on Systems Engineering Research 2009 (CSER 2009)* (S. 1–7). Loughborough, UK.
- Nakagawa, E. Y., Ferrari, F. C., Sasaki, M. M. F., Maldonado, J. C. (2011a).** An aspect-oriented reference architecture for Software Engineering Environments. *Journal of Systems and Software*, 84(10), 1670–1684.
- Nakagawa, E. Y., Guessi, M., Maldonado, J. C., Feitosa, D., Oquendo, F. (2014a).** Consolidating a process for the design, representation, and evaluation of reference architectures. In *Proceedings - 2014 IEEE/IFIP Conference on Software Architecture (WICSA)* (S. 143–152). Sydney, NSW, Australia: IEEE.
- Nakagawa, E. Y., Oliveira Antonino, P., Becker, M. (2011b).** Reference architecture and product line architecture: A subtle but critical difference. In I. Crnkovic, V. Gruhn, & M. Book (Hrsg.), *Software Architecture. ECSA 2011. Lecture Notes in Computer Science, vol 6903* (S. 207–211). Berlin, Heidelberg: Springer.
- Nakagawa, E. Y., Oquendo, F. (2013).** Perspectives and challenges of reference architectures in multi software product line. In *Proceedings of the 1st International Workshop on Multi Product Line Engineering (MultiPLE 2013), Aug 2013, Tokyo, Japan.* (S. 100–103). New York, New York, USA: ACM Press.
- Nakagawa, E. Y., Oquendo, F., Maldonado, J. C. (2014b).** Reference Architectures. In M. C. Oussalah (Hrsg.), *Software Architecture 1* (S. 258). London, U.K.: ISTE Ltd and John Wiley & Sons, Inc.
- Ni, J., Jin, X. (2012).** Decision support systems for effective maintenance operations. *CIRP Annals - Manufacturing Technology*, 61(1), 411–414.
- Niederberger, M., Wassermann, S. (Hrsg.). (2015).** *Methoden der Experten- und*

Stakeholdereinbindung in der sozialwissenschaftlichen Forschung. Wiesbaden: Springer Fachmedien Wiesbaden.

- Niemann, C. (2013).** *Behandlungsplanung in der Notaufnahme von Krankenhäusern.* Wiesbaden: Springer Fachmedien Wiesbaden.
- Niemöller, C., Metzger, D., Fellmann, M., Özcan, D., Thomas, O. (2016).** Shaping the Future of Mobile Service Support Systems – The Impact of Smart Glasses on Technical Customer Services. In H. Mayr & M. Pinzger (Hrsg.), *Informatik von Menschen für Menschen (Informatik 2016)* (S. 753–767). Klagenfurt, Österreich: Gesellschaft für Informatik.
- Niu, G., Yang, B. S., Pecht, M. (2010).** Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance. *Reliability Engineering and System Safety*, 95(7), 786–796.
- Nowlan, F. S., Heap, H. F. (1978).** *Reliability-Centered Maintenance.* San Francisco: United Airlines Publications.
- Nüttgens, M., Thomas, O., Fellmann, M. (2014).** *Dienstleistungsproduktivität.* (M. Nüttgens, O. Thomas, & M. Fellmann, Hrsg.). Wiesbaden: Springer Fachmedien Wiesbaden.
- OASIS Committee. (2012).** *Reference Architecture Foundation for Service Oriented Architecture Version 1.0* (Standards Track Work Product No. soa-ra-v1.0-cs01). (P. Brown, J. A. Estefan, K. Laskey, F. G. McCabe, & D. Thornthorn, Hrsg.). Abgerufen von <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/cs01/soa-ra-v1.0-cs01.html>
- OASIS Committee. (2017).** *MQTT Version 5.0.* (A. Banks, E. Briggs, K. Borgendale, & R. Gupta, Hrsg.).
- Offermann, P., Blom, S., Levina, O., Bub, U. (2010).** Vorschlag für Komponenten von Methodendesigntheorien. *Wirtschaftsinformatik*, 52(5), 287–297.
- Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., ... Sinz, E. J. (2010).** Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. *Zeitschrift für betriebswirtschaftliche Forschung*, 62(6), 662–672.
- Özcan, D., Fellmann, M., Matijacic, M., Däuble, G., Nüttgens, M., Thomas, O. (2013).** Use Case based description of IT-supported customer service processes. In *XXIII International Conference of the European Association for Research on Services (RESER 2013)* (S. 1–18). Aix en Provence, France.
- Pawellek, G. (2016).** *Integrierte Instandhaltung und Ersatzteillogistik* (2. Aufl.). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Payne, A., Frow, P. (2013).** A Strategic Framework for Customer Framework Relationship Management. *Journal of Marketing*, 69(4), 167–176.
- Persona, A., Regattieri, A., Pham, H., Battini, D. (2007).** Remote control and maintenance outsourcing networks and its applications in supply chain management. *Journal of Operations Management*, 25(6), 1275–1291.
- Pintelon, L., Parodi-Herz, A. (2008).** Maintenance: An Evolutionary Perspective. In *Complex System Maintenance Handbook* (S. 21–48). London: Springer London.
- Pistofidis, P., Emmanouilidis, C., Koulamas, C., Karampatzakis, D., Papathanassiou, N. (2012).** A layered e-maintenance architecture powered by smart wireless monitoring components. In *2012 IEEE International Conference on Industrial Technology (ICIT 2012)* (S. 390–395). Athens, Greece: IEEE.
- Pistofidis, P., Emmanouilidis, C., Papadopoulos, A., Botsaris, P. N. (2016).** Management of linked knowledge in industrial maintenance. *Industrial Management and Data Systems*, 116(8), 1741–1758.
- Pivotal Software Inc. (2018).** RabbitMQ - Messaging that just works. Abgerufen 11. Mai 2018, von <http://www.rabbitmq.com/>
- Pohl, K. (2010).** *Requirements Engineering - Fundamentals, Principles, and Techniques.*

Berlin, Heidelberg: Springer.

- Pohl, K., Böckle, G., van der Linden, F. (2005).** *Software Product Line Engineering*. Berlin, Heidelberg: Springer.
- Potes Ruiz, P. A., Kamsu-Foguem, B., Noyes, D. (2013).** Knowledge reuse integrating the collaboration from experts in industrial maintenance management. *Knowledge-Based Systems*, 50, 171–186.
- Potes Ruiz, P., Kamsu Foguem, B., Grabot, B. (2014).** Generating knowledge in maintenance from Experience Feedback. *Knowledge-Based Systems*, 68(1), 4–20.
- Projektseite „S-CPS“.** (2015). Abgerufen 7. September 2015, von www.s-cps.de
- Qingfeng, W., Wenbin, L., Xin, Z., Jianfeng, Y., Qingbin, Y. (2011).** Development and application of equipment maintenance and safety integrity management system. *Journal of Loss Prevention in the Process Industries*, 24(4), 321–332.
- Qu, R. (2006).** An Implementation of a Remote Diagnostic System on Rotational Machines. *Structural Health Monitoring*, 5(2), 185–193.
- Ramduy-Ellis, D., Dix, A., Rayson, P., Onditi, V., Sommerville, I., Ransom, J. (2005).** Artefacts as designed, artefacts as used: resources for uncovering activity dynamics. *Cognition, Technology & Work*, 7(2), 76–87.
- Rasovska, I., Chebel-Morello, B., Zerhouni, N. (2008).** A mix method of knowledge capitalization in maintenance. *Journal of Intelligent Manufacturing*, 19(3), 347–359.
- Rasovska, I., Chebel-Morello, B., Zerhouni, N. (2005).** Process of s-maintenance: decision support system for maintenance intervention. In *2005 IEEE Conference on Emerging Technologies and Factory Automation* (Bd. 2, S. 679–686). Catania, Italy: IEEE.
- Rausand, M. (1998).** Reliability centered maintenance. *Reliability Engineering and System Safety*, 60(2), 121–132.
- Reed, P. (2002).** Reference Architecture: The best of best practices. Abgerufen 15. Mai 2015, von <http://www.ibm.com/developerworks/rational/library/2774.html>
- Reichel, J., Müller, G., Mandelartz, J. (Hrsg.). (2009).** *Betriebliche Instandhaltung*. Berlin, Heidelberg: Springer.
- Reidt, A., Duchon, M., Krcmar, H. (2016a).** *Referenzarchitektur eines Ressourcen-Cockpits zur Unterstützung der Instandhaltung*. München: fortiss GmbH. Abgerufen von http://download.fortiss.org/public/scps/Referenzarchitektur_RC_Reidt_et_al_fortiss.pdf
- Reidt, A., Duchon, M., Krcmar, H. (2016b).** Sicherheitsaspekte von Industrie 4.0. In E. Müller & A. Bullinger-Hoffmann (Hrsg.), *Tagungsband „Smarte Fabrik & Smarte Arbeit – Industrie 4.0 gewinnt Kontur“ VPP2016 – Vernetzt planen und produzieren* (S. 445–456). Chemnitz: TU Chemnitz.
- Reidt, A., Duchon, M., Krcmar, H. (2017a).** Erstellung einer Referenzarchitektur anhand von individuellen Unternehmensanforderungen. In A. C. Bullinger-Hoffman (Hrsg.), *Abschlussveröffentlichung S-CPS: Ressourcen-Cockpit für Sozio-Cyber-Physische Systeme* (S. 23–42). Chemnitz: Verlag aw&I - Wissenschaft und Praxis.
- Reidt, A., Duchon, M., Krcmar, H. (2017b).** Referenzarchitektur eines Ressourcen-Cockpits zur Unterstützung der Instandhaltung. In A. C. Bullinger-Hoffman (Hrsg.), *Abschlussveröffentlichung S-CPS: Ressourcen-Cockpit für Sozio-Cyber-Physische Systeme* (S. 43–60). Chemnitz: Verlag aw&I - Wissenschaft und Praxis.
- Reidt, A., Krcmar, H. (2016).** Referenzarchitektur für Cyber-physische Systeme zur Unterstützung der Instandhaltung. In V. Nissen, D. Stelzer, S. Straßburger, & D. Fischer (Hrsg.), *Multikonferenz Wirtschaftsinformatik (MKWI) 2016* (S. 377–388). Ilmenau: Universitätsverlag Ilmenau.
- Reidt, A., Pfaff, M., Krcmar, H. (2018a).** Der Referenzarchitekturbegriff im Wandel der Zeit. *HMD Praxis Der Wirtschaftsinformatik*, 55(5), 893–906.

- Reidt, A., Schuhbäck, S., Krcmar, H. (2018b).** Comprehensive View on Architectural Requirements for Maintenance Information Systems. In *Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2018)* (S. 249–257). Madeira, Portugal: SCITEPRESS – Science and Technology Publications, Lda.
- Renner, J. (2006).** *Mobile Agenten für den Fernzugriff auf eingebettete Systeme*. Technische Universität Chemnitz, Chemnitz. Abgerufen von http://www.qucosa.de/fileadmin/data/qucosa/documents/5222/data/Dissertation_Web.pdf
- Riege, C., Saat, J., Bucher, T. (2011).** Systematisierung von Evaluationsmethoden in der gestaltungsorientierten Wirtschaftsinformatik. In J. Becker, H. Krcmar, & B. Niehaves (Hrsg.), *Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik* (S. 69–86). Heidelberg: Physica-Verlag.
- Robertson, S., Robertson, J. (2008).** *Mastering the Requirements Process* (2. Aufl.). Boston, MA, USA: Pearson Education.
- Rodrigues, M., Hatakeyama, K. (2006).** Analysis of the fall of TPM in companies. *Journal of Materials Processing Technology*, 179(1–3), 276–279.
- Rosqvist, T., Laakso, K., Reunanen, M. (2009).** Value-driven maintenance planning for a production plant. *Reliability Engineering and System Safety*, 94(1), 97–110.
- Roth, A. (Hrsg.). (2016).** *Einführung und Umsetzung von Industrie 4.0*. Berlin, Heidelberg: Springer.
- Savioja, P., Järvinen, P., Karhela, T., Siltanen, P., Woodward, C. (2007).** Developing a Mobile, Service-Based Augmented Reality Tool for Modern Maintenance Work. In S. R. (Hrsg.), *Proceedings Virtual Reality. ICVR 2007. Lecture Notes in Computer Science, vol 4563. Beijing, China, July 22-27, 2007*. (S. 554–563). Beijing, China: Springer.
- Schenk, M., Wirth, S., Müller, E. (2014).** *Fabrikplanung und Fabrikbetrieb*. Berlin, Heidelberg: Springer Vieweg.
- Schlicker, M. (2014).** *Dienstleistungsproduktivität*. (M. Nüttgens, O. Thomas, & M. Fellmann, Hrsg.), *Dienstleistungsproduktivität*. Wiesbaden: Springer Fachmedien Wiesbaden.
- Schmidt, J., Hoof, a Van. (2013).** Architektur einer Service Plattform zur Unterstützung des Betriebs erneuerbarer Energieanlagen. In *Wirtschaftsinformatik Proceedings 2013* (S. 1–15). Leipzig, Germany.
- Scholl, A. (2009).** *Die Befragung* (2. Aufl.). Stuttgart: UTB.
- Schröder, W. (2009).** *Ganzheitliches Instandhaltungsmangement*. (U. Bauer, H. Biedermann, & J. Wohinz, Hrsg.). Wiesbaden: Gabler.
- Schuh, G., Lorenz, B. (2009).** TPM – eine Basis für die wertorientierte Instandhaltung. In J. M. J. Reichel, G. Müller (Hrsg.), *Betriebliche Instandhaltung* (S. 75–87). Berlin, Heidelberg: Springer.
- Schwegmann, A. (1999).** *Objektorientierte Referenzmodellierung*. (J. Becker, H. L. Grob, & S. Klein, Hrsg.). Wiesbaden: Deutscher Universitätsverlag.
- Sharma, A., Yadava, G. S., Deshmukh, S. G. (2011).** A literature review and future perspectives on maintenance optimization. *Journal of Quality in Maintenance Engineering*, 17(1), 5–25.
- Stadtler, H. (2005).** Supply Chain Management – An Overview. In H. Stadtler & K. Christoph (Hrsg.), *Supply Chain Management and Advanced Planning* (3. Aufl.). Berlin, Heidelberg: Springer.
- Starr, A., Al-Najjar, B., Holmberg, K., Jantunen, E., Bellew, J., Albarbar, A. (2010).** Maintenance Today and Future Trends. In K. Holmberg, A. Adgar, A. Arnaiz, E. Jantunen, J. Mascolo, & S. Mekid (Hrsg.), *E-maintenance* (S. 5–37). London: Springer.
- Stenström, C., Parida, A., Kumar, U., Galar, D. (2013).** Performance indicators and

- terminology for value driven maintenance. *Journal of Quality in Maintenance Engineering*, 19(3), 222–232.
- Susto, G. A., Schirru, A., Pampuri, S., Beghi, A. (2012).** A predictive maintenance system based on regularization methods for ion-implantation. In *2012 SEMI Advanced Semiconductor Manufacturing Conference* (S. 175–180). Saratoga Springs, NY, USA: IEEE.
- Susto, G. A., Schirru, A., Pampuri, S., Pagano, D., McLoone, S., Beghi, A. (2013).** A predictive maintenance system for integral type faults based on support vector machines: An application to ion implantation. In *2013 IEEE International Conference on Automation Science and Engineering (CASE)* (S. 195–200). Madison, WI, USA: IEEE.
- Swanson, L. (2001).** Linking maintenance strategies to performance. *International Journal of Production Economics*, 70(3), 237–244.
- Swift, M., Aurisicchio, G., Pace, P. (2011).** New practices for railway condition monitoring and predictive analysis. In *5th IET Conference on Railway Condition Monitoring and Non-Destructive Testing (RCM 2011)*. Derby, UK: IET.
- Syafar, F., Gao, J. (2013).** Mobile collaboration technology in engineering asset maintenance: A Delphi study. In *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (S. 483–488). Whistler, BC, Canada: IEEE.
- Taylor, R. N., Medvidovic, N., Dashofy, E. M. (2009).** *Software Architecture: Foundations, Theory, and Practice*. Hoboken, N.J., USA: John Wiley & Sons.
- Trefke, J. (2012).** Grundlagen der Referenzarchitekturentwicklung. In H.-J. Appelrath, P. Beenken, L. Bischofs, & M. Uslar (Hrsg.), *IT-Architekturentwicklung im Smart Grid* (S. 9–30). Berlin, Heidelberg: Springer.
- Trommler, U., Horbach, S., Bullinger-Hoffmann, A. C., Löffler, T., Müller, E., Hopf, H. (2014).** Instandhaltung in der Industrie 4.0. *Industrie Management*, 30(6), 51–54.
- Tsang, A. H. C., Yeung, W. K., Jardine, A. K. S., Leung, B. P. K. (2006).** Data management for CBM optimization. *Journal of Quality in Maintenance Engineering*, 12(1), 37–51.
- Tschamler, H. (1996).** *Wissenschaftstheorie - Eine Einführung für Pädagogen* (3. Aufl.). Bad Heilbrunn: Julius Klinkhardt.
- Tsuchiya, S. (1992).** *Quality Maintenance: Zero Defects Through Equipment Management*. Cambridge, MA, USA: Productivity Press.
- Usländer, T., Epple, U. (2015).** Reference model of Industrie 4.0 service architectures. *at - Automatisierungstechnik*, 63(10), 858–866.
- VDI/VDE Fachausschuss Industrie 4.0. (2017).** *Industrie 4.0 Begriffe/Terms* (VDI-Statusreport VDI/VDE Fachausschuss 7.21 - Industrie 4.0). Düsseldorf. Abgerufen von <http://www.iosb.fraunhofer.de/?BegriffeI40>
- VDI 2895:2012-12. (2012).** Organisation der Instandhaltung - Instandhalten als Unternehmensaufgabe. In VDI Verein Deutscher Ingenieure e.V. (Hrsg.), *VDI-Handbuch Fabrikplanung und -betrieb, Band 1: Betriebsüberwachung/Instandhaltung* (S. 1–27). Berlin: Beuth Verlag.
- VDI 4500-3:2006-6. (2006).** *Technische Dokumentation - Erstellen und Verteilen von elektronischen Ersatzteilmformationen*. (VDI Verein Deutscher Ingenieure e.V., Hrsg.). Berlin: Beuth Verlag.
- VDI 5600-3:2013-7. (2013).** *Fertigungsmanagementsysteme (Manufacturing Execution Systems – MES) Logische Schnittstelle zur Maschinen- und Anlagensteuerung*. (VDI Verein Deutscher Ingenieure e.V., Hrsg.). Berlin: Beuth Verlag.
- Vogel, O., Arnold, I., Chughtai, A., Ihler, E., Kehrner, T., Mehlig, U., Zdun, U. (2009).** *Software-Architektur* (2. Aufl.). Heidelberg: Spektrum Akademischer Verlag.
- Voigt, K.-I., Steinmann, F., Bauer, J., Dremel, A. (2013).** Condition monitoring as a key

- technology – An analysis of requirements for new business models for remote services. In *9. Symposium für Vorausschau und Technologieplanung*. Berlin.
- vom Brocke, J. (2015).** *Referenzmodellierung - Gestaltung und Verteilung von Konstruktionsprozessen*. (J. Becker, H. L. Grob, S. Klein, H. Kuchen, U. Müller-Funk, & G. Vossen, Hrsg.) (2. Aufl.). Berlin: Logos Verlag.
- vom Brocke, J., Buddendick, C. (2004).** Organisationsformen in der Referenzmodellierung - Forschungsbedarf und Gestaltungsempfehlungen auf Basis der Transaktionskostentheorie. *Wirtschaftsinformatik*, 46(5), 341–352.
- Wang, L., Chu, J., Mao, W., Fu, Y. (2006).** Advanced Maintenance Strategy for Power Plants - Introducing Intelligent Maintenance System. In *Proceedings of the 6th World Congress on Intelligent Control and Automation* (Bd. 2, S. 7444–7448). Dalian, China: IEEE.
- Wang, W., Tse, P. W., Lee, J. (2007).** Remote machine maintenance system through Internet and mobile communication. *International Journal of Advanced Manufacturing Technology*, 31(7–8), 783–789.
- Wengraf, T. (2001).** *Qualitative research interviewing* (Bd. 30). London, U.K.: SAGE Publications, Ltd.
- Wilde, T., Hess, T. (2007).** Forschungsmethoden der Wirtschaftsinformatik Eine empirische Untersuchung. *Wirtschaftsinformatik*, 49(4), 280–287.
- Willmott, P., McCarthy, D. (2000).** *TPM - A Route to World Class Performance: A Route to World Class Performance*. Oxford, U.K.: Newnes.
- Winter, A., Wollschlaeger, M. (2015).** Condition Monitoring: Die Inhalte des VDMA-Einheitsblattes 24582. Abgerufen 6. September 2015, von <http://www.computer-automation.de/steuerungsebene/fernwirken/artikel/108262/>
- Winter, R. (2009).** Was ist eigentlich Grundlagenforschung in der Wirtschaftsinformatik? *Wirtschaftsinformatik*, 51(2), 223–231.
- Wollschlaeger, M., Theurich, S., Winter, A., Lubnau, F., Paulitsch, C. (2015).** A Reference Architecture for Condition Monitoring. In J. Proenza & T. Sauter (Hrsg.), *2015 IEEE World Conference on Factory Communication Systems (WFCS)* (S. 1–8). Palma de Mallorca, Spain: IEEE.
- Yamashina, H. (1995).** Japanese manufacturing strategy and the role of total productivemaintenance. *Journal of Quality in Maintenance Engineering*, 1(1), 27–38.
- Zerbst, M. (2000).** *Total Productive Maintenance*. Wiesbaden: Deutscher Universitätsverlag.
- Zhu, H., Gao, J., Li, D., Tang, D. (2012).** A Web-based product service system for aerospace maintenance, repair and overhaul services. *Computers in Industry*, 63(4), 338–348.
- Ziegler, J., Pfeffer, J., Urbas, L. (2011).** A mobile system for industrial maintenance support based on embodied interaction. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction (TEI '11)* (S. 181). New York, NY, USA: ACM Press.
- Zuccolotto, M., Pereira, C. E., Fasanotti, L., Cavalieri, S., Lee, J. (2015).** Designing an Artificial Immune Systems for Intelligent Maintenance Systems. *IFAC-PapersOnLine*, 48(3), 1451–1456.

Auflistung der relevanten Vorveröffentlichungen

- Fleischmann, H., Kohl, J., Franke, J., Reidt, A., Duchon, M.; Krcmar, H. (2016).** Improving maintenance processes with distributed monitoring systems. In *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)* (S. 377–382). Poitiers, France: IEEE.
- Reidt, A., Duchon, M.; Krcmar, H. (2016a).** *Referenzarchitektur eines Ressourcen-Cockpits zur Unterstützung der Instandhaltung*. München: fortiss GmbH.
- Reidt, A., Duchon, M.; Krcmar, H. (2016b).** Sicherheitsaspekte von Industrie 4.0. In E. Müller & A. Bullinger-Hoffmann (Hrsg.), *Tagungsband „Smarte Fabrik & Smarte Arbeit – Industrie 4.0 gewinnt Kontur“ VPP2016 – Vernetzt planen und produzieren* (S. 445–456). Chemnitz: TU Chemnitz.
- Reidt, A., Duchon, M.; Krcmar, H. (2017a).** Erstellung einer Referenzarchitektur anhand von individuellen Unternehmensanforderungen. In A. C. Bullinger-Hoffman (Hrsg.), *Abschlussveröffentlichung S-CPS: Ressourcen-Cockpit für Sozio-Cyber-Physische Systeme* (S. 23–42). Chemnitz: Verlag aw&I - Wissenschaft und Praxis.
- Reidt, A., Duchon, M.; Krcmar, H. (2017b).** Referenzarchitektur eines Ressourcen-Cockpits zur Unterstützung der Instandhaltung. In A. C. Bullinger-Hoffman (Hrsg.), *Abschlussveröffentlichung S-CPS: Ressourcen-Cockpit für Sozio-Cyber-Physische Systeme* (S. 43–60). Chemnitz: Verlag aw&I - Wissenschaft und Praxis.
- Reidt, A.; Krcmar, H. (2016).** Referenzarchitektur für Cyber-physische Systeme zur Unterstützung der Instandhaltung. In V. Nissen, D. Stelzer, S. Straßburger, & D. Fischer (Hrsg.), *Multikonferenz Wirtschaftsinformatik (MKWI) 2016* (S. 377–388). Ilmenau: Universitätsverlag Ilmenau.
- Reidt, A., Pfaff, M., & Krcmar, H. (2018).** Der Referenzarchitekturbegriff im Wandel der Zeit. *HMD Praxis Der Wirtschaftsinformatik*, 55(5), 893–906.
- Reidt, A., Schuhbäck, S.; Krcmar, H. (2018).** Comprehensive View on Architectural Requirements for Maintenance Information Systems. In *Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2018)* (S. 249–257). Madeira, Portugal: SCITEPRESS – Science and Technology Publications, Lda.

Anhang

8.6 Fragebogen Anforderungsaufnahme

Der vorliegende Fragebogen ordnet die Fragen in vier Priorisierungsstufen ein. Diese Priorisierungsstufen sind wie folgt definiert:

Priorisierung	Erläuterung
Priorisierung 1 - Muss:	Die Anforderung muss implementiert werden, sonst ist bietet das System keinen Mehrwert/Mindestanforderung
Erklärung Prio 2 - Soll:	Das System würde durch die Umsetzung dieser Anforderung deutlich effizienter/effektiver
Erklärung Prio 3 - Kann:	Es wäre gut, wenn das System diese Eigenschaft besitzen würde, es ist aber kein Muss
Erklärung Prio 4:	Keine Priorität

Tabelle 40: Priorisierung von Anforderungen
Quelle: Eigene Darstellung

Nr. FB.	Anforderung	Prio 1	Prio 2	Prio 3	Prio 4
1	Detaillierte Fehlermeldung mit Störungsart im System verfügbar				
2	Fehlerkategorisierung nach Schwere und Dringlichkeit im System verfügbar				
3	Mehrsprachigkeit von Fehlermeldungen				
4	Maschine sendet automatisch Fehlermeldung an konfigurierbaren Empfänger				
5	Fehlermeldung an der Maschine auslesbar				
6	Detaillierte Fehlermeldung komplett auf der Maschine vorhanden				
7	Fehlerlokalisierung auf 3D Modell der Anlage im System verfügbar				
8	Fehlerdatenbank mit Suchfunktion im System verfügbar				
9	Fehler bzw. Wartungsbedarf können manuell in das System eingetragen werden				
10	(Mobiler) Zugriff auf relevante Zustandsdaten der Maschinen				
11	Anzeige Betriebstemperaturen				
12	Maschinenplänen (Steuerungspläne, SPS) im System verfügbar				
13	Bauteilübersicht der Anlagen im System verfügbar				
14	Beschreibung und Informationen zu den anlagenspezifischen Bauteilen im System verfügbar				
15	Bauteilhistorie pro Anlage im System verfügbar				
16	3D Darstellung der Anlage im System verfügbar				

Nr. FB.	Anforderung	Prio 1	Prio 2	Prio 3	Prio 4
17	Informationen zur Produktionsplanung/Anlagenbelegung im System verfügbar				
18	Link zum Kataster der Hilfs- und Betriebsstoffe einer Anlage im System verfügbar				
19	Softwareversionen der Maschine im System verfügbar				
114	Instant-Feedback zum Erfolg der Reparatur (durch Kunde und/oder Maschine)				
20	Herstellerunterlagen für Anlagen im System verfügbar				
21	Kontaktinformationen für Service des Herstellers/Serviceanbieters der Anlage im System				
22	Anzeige aktuelle Störungen mitsamt Störungsmeldung pro Maschine im System verfügbar				
23	Maßnahmenhistorie (Instandhaltungsmaßnahmen) pro Anlage im System verfügbar				
24	Synchronisierung der Maßnahmenhistorie innerhalb einer Schicht mit einem evtl. vorhandenem elektronischem Schichtbuch				
25	Anzeige/Berechnung produktionsfreier Zeiten im System verfügbar				
26	Auswertung und Anzeige der aktuellen und mittleren Laufzeiten der Anlagen (Stördauer/Anzahl Störungen etc.) im System verfügbar				
27	Überwachung/Protokollierung relevanter Maschinendaten und Zustände in Form eines Dashboards o. Ä. im System verfügbar				
28	Änderungen an der HW-/SW-Konfiguration einer Anlage führen zu Benachrichtigungen durch das System an zuständige Personen				
30	Anlagenübersicht mit Statusinformation im System verfügbar				
29	Intranetzzugang bzw. Zugang zu relevanten firmeninternen Webanwendungen/Informationen				
31	Wiki im System verfügbar				
32	Anlagenplan (örtlich) im System verfügbar				
33	Navigation innerhalb des Standortes im System verfügbar				
35	Handlungsleitfäden/Checklisten für Instandhaltungsaufgaben im System verfügbar (Hinzufügen, Speichern, Editieren, Anzeigen)				
36	Bearbeitung, Hinzufügen von Dokumenten zu Handlungsleitfäden, Checklisten, Wiki (PDF, Word, etc.) im System möglich				
37	Handlungsleitfäden in einzelnen protokollierten Schritten abarbeitbar				
38	Darstellung, Abarbeitung und Speicherung der Handlungsleitfäden in Form von Entscheidungsbäumen möglich				
39	Fehlerunabhängige Standard-Handlungsempfehlungen verfügbar/bearbeitbar				
40	Automatische Erstellung von Handlungsleitfäden anhand des Vorgehens des Instandhalters				

Nr. FB.	Anforderung	Prio 1	Prio 2	Prio 3	Prio 4
41	Mehrsprachigkeit Handlungsempfehlungen				
42	Handlungsleitpfaden zur Fehlerdiagnose				
43	Abweichungen von Handlungsempfehlungen per Bestätigung im System zulassen				
44	Verfügbare Ersatzteile (intern)/Bestandsituation mit Lagerort im System verfügbar				
45	Anzeigen verfügbarer Ersatzteile (extern Bezugsquelle - Händler/Hersteller) im System verfügbar				
46	Verbindung benötigte Ersatzteile zu Instandhaltungsmaßnahmen im System verfügbar				
47	Nachbestellen von Ersatzteilen wird durch das System ermöglicht (fehlende Ersatzteile oder z. B. nach Verbrauch)				
48	Proaktives Bestellen der Ersatzteile durch das System				
49	Anzeige verfügbare Werkzeuge intern mit Lagerort im System verfügbar				
50	Verbindung benötigte Werkzeuge zu Instandhaltungsmaßnahmen im System verfügbar				
51	Anleitungen zur Benutzung und Beschreibung des Werkzeuges im System verfügbar				
52	(Automatische) Priorisierung der abzuarbeitenden Aufträge/Tätigkeiten durch das System				
53	Manuelle Auftragsverteilung/Priorisierung der abzuarbeitenden Aufträge/Tätigkeiten durch Management im System verfügbar				
54	Rückstellung Auftrag bei fehlenden Ersatzteilen durch das System [Wiedervorlagedatum durch Instandhalter]				
70	Mitarbeiter können im System offene Aufträge selbst auswählen				
55	Automatische Benachrichtigung zuständiger Personen über Auftragseingänge durch das System				
56	Automatische Dokumentation/Protokollierung der Maßnahmen der Instandhaltung (Wartung und Reparatur) und dazugehörigen Schritte der Handlungsleitfäden				
57	Auswertung mit anschließender Anzeige eines Berichts über die durchgeführten Maßnahmen der Instandhaltung nach Beendigung der Tätigkeit				
58	Manuelle Dokumentationsmöglichkeit nach abgearbeiteten Auftrag durch das System gegeben				
59	Protokollierung der auftragsbezogenen Kommunikation durch das System				
60	Anzeige offener Aufträge durch das System				
61	Auftragshistorie vom System darstellbar				
62	Abfrage und Anzeige der Status der Instandhaltungsaufträge durch das System				
63	Auftragsmanagement in Verbindung mit internem ERP				

Nr. FB.	Anforderung	Prio 1	Prio 2	Prio 3	Prio 4
64	Auswählbare Eskalation bei laufenden Aufträgen an nächsthöhere Stufe im System verfügbar				
65	Automatische (zeitlich abhängige) Eskalation bei laufenden Aufträgen an nächsthöhere Stufe durch System verfügbar				
66	Eskalationsmöglichkeit mitsamt Informationsweitergabe an Hersteller oder Fachfirma in System integriert				
34	Gesetzliche Regelungen zu Instandhaltungsauftrag im System verfügbar				
67	Anzeigen der Verfügbarkeit/aktuelle Zuweisung/der Instandhalter im System verfügbar				
117	Speicherung von Mitarbeiterdaten in Bezug auf Schulungen, Zulassungen, Fähigkeiten für bestimmte Maschinen mit Ablaufdatum				
68	Annehmen, Ablehnen von Instandhaltungsaufträgen durch Instandhalter im System verfügbar				
69	GPS zur Mitarbeiterkoordination/Anzeige im System verfügbar				
71	Zuweisung von Zuständigkeit von Personen bzw. Personengruppen für bestimmte Anlagen im System verfügbar				
72	Erfassung der Arbeitszeit durch das System				
73	Anonymisierungsfunktion der Mitarbeiter für das Management				
99	Predictive Maintenance (kontinuierlich lernendes, vorausschauendes System)				
116	Dedizierte Wartungsberichte sind im System verfügbar				
100	Synonymfunktion/Wörterbuch mit Lernfunktion im System verfügbar				
101	Wartungshistorie im System darstellbar und auswertbar				
102	Wartungsbenachrichtigung wird durch das System gesendet				
103	Berechnung des Wartungsplans durch das System				
104	Wartungsstand wird durch das System zurückgesetzt				
105	Wartungsgrenzen von Anlagen/Bauteilen werden durch das System berechnet				
106	Wetterdaten (Windrichtung, Temperatur) werden durch das System abgerufen				
107	Übertragung von Informationen in interne Systeme (z.B. PDM)				
115	Personalisierungsfunktion bei Darstellung (Geräteeinstellungen, Darstellung etc.)				
108	Automatische Selbstwartung/-reparatur der Maschine				
109	Condition Monitoring an der Anlage				
110	OPC UA Schnittstelle zum Empfang bzw. Senden von Maschinendaten				
111	Pluginkonzept zur Integration von diversen Schnittstellen und Maschinen				
112	Freihändige Bedienung des mobilen Endgerätes				

Nr. FB.	Anforderung	Prio 1	Prio 2	Prio 3	Prio 4
113	Augmented Reality Features				
74	Remotesteuerung Anlage				
75	Remotesteuerung Prozessgeräte				
76	Remotesteuerung Bedienpulte				
77	Manuelle Freigabe an Maschine für Remotezugriff erforderlich				
78	Mobiler Zugriff auf SPS				
79	Messengerfunktionalität (senden und Empfangen von Nachrichten mitsamt Bild, Ton und Video) im System verfügbar				
80	Mailfunktionalität (Zugang zu internen Mailservice, Schreiben, Senden, Empfangen von Mails) im System verfügbar				
81	Adressbuch/Kontakte im System verfügbar				
82	Aufgabenbezogene Kommunikation ohne Medienbrüche (Messaging, Anrufe aus Anwendung heraus)				
83	Scannen der Anlagencodierung und Identifikation im System				
84	Mobiles Endgerät als Telefonersatz				
85	Offline Abzugsmöglichkeit von Anlagendaten per mobilem Endgerät				
86	Synchronisationsmechanismus zwischen mobilem Endgerät, Backend sowie Anlage zur Konsistenzhaltung aller Datenbestände				
87	Installationsmöglichkeit Apps auf mobilen Device				
88	Internetfähigkeit des mobilen Endgerätes				
89	Mobiles Arbeiten im Offlinemodus				
90	Verarbeitung audiovisueller Medien (Erstellen, Speichern, Anzeigen von Foto, Video und Audio) durch das System				
91	GPS zur Navigation und Routenplanung				
92	(Semi-Automatisierter) Abgleich von Betriebsdaten (z. B. Protokoll Störfälle & Fehlermeldungen) zwischen Anlagenbetreiber und -hersteller durch System ermöglicht				
93	Benachrichtigung an den Kunden über Eintreffen des Mitarbeiters				
94	Erfassung von Reisekosten durch das System				
95	Automatische Erstellung von Leistungsberichten durch das System				
96	Freigabe von Handlungsempfehlungen an Kunden durch das System ermöglicht				
97	Anzeige von aktuellen Vertragsdaten für Auftrag (Leistungen etc.)				
98	Signierfunktion durch Auftraggeber und Instandhalter				

Tabelle 41: Fragebogen Anforderungserhebung

Quelle: Eigene Darstellung

8.7 Leitfaden Telefoninterview

Leitfaden zur Durchführung der Interviews zur Evaluierung der Referenzarchitektur		
Datum der Befragung		
Beteiligte Personen		
Aufnahme (Ja/Nein)		
Mitschrift (Ja/Nein)		
Telefoninterview(Ja/Nein)		
Name der befragten Person		
Unternehmen		
Position im Unternehmen		
Wie lange auf dieser Position tätig (in Jahren)		
Erfahrung im Bereich IT-Projekte/Anforderungsaufnahme		
Erfahrung im Bereich Industrie 4.0/Digitalisierung		
Erfahrung im Bereich Architektur		
Kategorie	Nr.	Frage (Abkürzungen RC=Ressourcen-Cockpit, RA= Referenzarchitektur)
Allgemein	1	Wie haben Sie mit der Referenzarchitektur gearbeitet?
Allgemein	2	Haben Sie Kenntnis über weitere offen verfügbare Architekturen für Instandhaltungssysteme? Wenn ja, wie empfinden sie diese im Vergleich zur RA?
Allgemein	3	Hilft die RA eine Diskussion über das zu entwickelnde System zu führen? Wenn ja, wie?
Allgemein	4	Wie empfanden Sie die Darstellung der Referenzarchitektur? War sie leicht verständlich? Für Entwickler, für nicht Entwickler?
Allgemein	5	Hilft die RA ein logisches, einfach zu kommunizierendes Bild des zu entwickelnden Systems zu vermitteln?
Allgemein	6	Wie bewerten Sie das Abstraktionsniveau der RA hinsichtlich der Entwicklung des RC? Zu Abstrakt für die Entwicklung oder zu konkret?
Allgemein	7	Wie bewerten Sie das Abstraktionsniveau der RA hinsichtlich allen Projektbeteiligten (Nicht Entwickler) des RC? Zu Abstrakt für die Entwicklung oder zu konkret?
Einheitlichkeit	8	Hilft die RA eine einheitliche Domänensprache zwischen den an der Entwicklung beteiligten Personen aufzubauen?
Einheitlichkeit	9	Hilft die RA dabei ein RC zu entwickeln, welches mit anderen RCs, die auf der RA aufbauen kompatibel ist?
Einheitlichkeit	10	Hilft die RA sich auf einen generischen Kern, den es umzusetzen gilt, zu fokussieren?
Anpassung	11	Hilft die RA unternehmensspezifische Anpassungen eines RCs vorzunehmen?
Anpassung	12	Hilft die RA die Prozesse zu identifizieren bei denen das RC den Instandhalter konkret helfen kann? (unabhängig vom Projekt)
Anforderungen	13	Hilft Ihnen die RA die (generelle) Anforderungen an ein RC leichter zu verstehen?

Anforderungen	14	Hilft die RA eigene Anforderungen an ein RC zu verstehen und klarer zu formulieren?
Anforderungen	15	Hilft die RA um zwischen generischen und optionalen Anforderungen eines an ein RC zu unterscheiden?
Anforderungen	16	Hilft die Verbindung zwischen Modul und Anforderungen den Entwicklern die Verbindung zw. Anforderungen und technischer Lösung zu erkennen?
Anforderungen	17	Hilft es einer zielorientierten Diskussion mit den Stakeholdern des Projektes direkt Anforderungen mit den Modulen zu verbinden?
Anforderungen	18	Hat die die Verbindung von Anforderungen zu Modul bei der Nachverfolgbarkeit von Anforderungen geholfen?
Anforderungen	19	War die verwendet visuelle Darstellung in der RA hilfreich? Verwenden Sie diese Art der Darstellung auch in anderen Projekten? Wenn Nein, was verwenden Sie stattdessen?
Anforderungen	20	Haben Sie weitere Anmerkungen zur Referenzarchitektur oder Verbesserungsvorschläge?

Tabelle 42: Leitfaden Telefoninterview

Quelle: Eigene Darstellung