



[SKB+25] G. Schuh, A. Keuper, G. Bönsch, S. Schümmelfeder, H. Lauf, C. Wyrwich, S. Hillemacher, J. Michael, B. Rumpe, S. Stüber:
Leitfaden zur agilen, datenbasierten Produktentwicklung in der Windenergiebranche.
ISBN 978-3-98555-284-9, Apprimus Wissenschaftsverlag, May 2025.

FORSCHUNGSBERICHT

FORSCHUNGSBERICHT

FORSCHUNGSBERICHT

Die Windenergiebranche in Deutschland befindet sich in einem Umbruch. Hersteller sehen sich mit stark volatilen politischen Rahmenbedingungen sowie hohen Individualisierungsanforderungen konfrontiert. Um im stark wachsenden Markt für erneuerbare Energien bestehen zu können, müssen Unternehmen auf Veränderungen und neue Regularien entsprechend flexibel reagieren können. Die traditionellen plangetriebenen Produktentwicklungsprozesse zeichnen sich durch ihren deterministischen Charakter aus, welcher die Anpassungsfähigkeit und der damit verbundenen Reaktionsfähigkeit auf Umwelteinflüsse in späteren Entwicklungsphasen einschränkt. Agile Entwicklungsprozesse aus der Softwareentwicklung versprechen durch das iterative Vorgehen mehr Flexibilität und Anpassungsfähigkeit in der Produktentwicklung. Die direkte Übertragbarkeit von agilen Methoden der Softwareentwicklung ist unter Berücksichtigung der spezifischen Randbedingungen von Windenergieanlagen auf diese noch nicht gewährleistet. Die Agilisierung der Entwicklungsprozesse eines hochkomplexen Systems im Kontext der Windenergiebranche geht mit dem Aufbau eines Informationsmodells einher, um die benötigte Informationsgrundlage zu gewährleisten. Um die Vorteile agiler Produktentwicklungsmethoden zu nutzen, wurde ein Informationsmodell und ein neuer Entwicklungsprozess, welcher plangetriebene und agile Entwicklungsmethoden vereint, entwickelt und für die Anwendung im Windenergiesektor eingeführt. Das entwickelte Vorgehen ermöglicht Änderungsprozesse im Produktentwicklungsprozess effizient abzubilden, um dadurch dem volatilen Umfeld gerecht zu werden. Die Entwicklung eines solchen methodischen Vorgehens war Gegenstand des vom Bundesministerium für Bildung und Forschung geförderten und vom Projektträger Jülich betreuten Forschungsprojekts Agile Data Dev.

Der vorliegende Leitfaden zur agilen, datenbasierten Produktentwicklung in der Windenergiebranche ist ein Resultat des Projekts und beschreibt anwendungsnah die schrittweise Umsetzung des neuen Entwicklungsvorgehens in der Praxis. Er dient als praktische Hilfestellung für Unternehmen bei dem Aufbau einer datenzentrierten und agilen Produktentwicklung. Ein Fokus liegt dabei auf der Windenergiebranche.



Leitfaden zur agilen, datenbasierten Produktentwicklung
in der Windenergiebranche

Günther Schuh (Hrsg.)



Günther Schuh (Hrsg.)

Leitfaden zur agilen, datenbasierten Produktentwicklung in der Windenergiebranche



Leitfaden zur agilen, datenbasierten Produktentwicklung in der Windenergiebranche

Gefördert durch:



**Bundesministerium
für Wirtschaft
und Klimaschutz**

**aufgrund eines Beschlusses
des Deutschen Bundestages**

Autoren:

Dieser Leitfaden ist unter gemeinsamer Zusammenarbeit aus dem Werkzeugmaschinenlabor WZL, dem Lehrstuhl und Institut für Maschinenelemente und Systementwicklung IMSE, und dem Lehrstuhl für Software Engineering SE entstanden.

An dieser Stelle sind alle beteiligten Autoren die für dieses gemeinsame Werk zusammengearbeitet haben aufgeführt:

WZL: Günther Schuh, Alexander Keuper, Gereon Bönsch,
Sven Schümmelfeder und Hendrik Lauf

IMSE: Christian Wyrwich

SE: Steffen Hillemacher, Judith Michael, Bernhard Rumpe und
Sebastian Stüber

Danksagung

Das Forschungsprojekt „Agile datenbasierte Produktentwicklung von Windenergieanlagen“ wurde durch den Projektträger Jülich (PTJ) unter dem Förderkennzeichen 0324341B betreut. Die Finanzierung erfolgte aufgrund eines Beschlusses des Deutschen Bundestags durch das Bundesministerium für Wirtschaft und Klimaschutz, wofür wir uns herzlich bedanken.

Bibliografische Information der Deutschen Nationalbibliothek

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

Günther Schuh (Hrsg.):

Leitfaden zur agilen, datenbasierten Produktentwicklung in der Windenergiebranche

1. Auflage, 2025

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

Copyright Apprimus Verlag, Aachen, 2025

Wissenschaftsverlag des Instituts für Industriekommunikation und Fachmedien
an der RWTH Aachen

Steinbachstr. 25, 52074 Aachen

Internet: www.apprimus-verlag.de, E-Mail: info@apprimus-verlag.de

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany

ISBN 978-3-98555-238-2

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.1.1	Motivation aus Sicht der Praxis	3
1.1.2	Motivation aus Sicht der Forschung	5
1.2	Potenziale in der Produktentwicklung der Windenergiebranche	8
1.2.1	Effektive Produktentwicklung durch agile Entwicklungsmethoden.....	10
1.2.2	Effiziente Produktentwicklung durch eine echtzeitfähige Datenverfügbarkeit	11
1.3	Zielsetzung und Aufbau	13
1.4	Vorstellung des Forschungsprojekts „Agile_Data_Dev“	16
2	Konzeption einer agilen, datenbasierten Produktentwicklung	23
2.1	Erfolgsfaktoren der agilen, datenbasierten Produktentwicklung	23
2.1.1	Erfolgsfaktoren für eine agile Produktentwicklung.....	24
2.1.2	Erfolgsfaktoren für datenbasierte Produktentwicklung	27
2.1.3	Zwischenfazit.....	33
2.2	Grobkonzept eines Vorgehensmodells für die agile, datenbasierte Produktentwicklung	34
2.2.1	Vorgehen in der Analysephase	34
2.2.2	Vorgehen in der Konzeptionsphase.....	34
2.2.3	Vorgehen in der Detaillierungsphase.....	35
2.2.4	Vorgehen in der Implementierungsphase.....	35
3	Vorgehen zur Umsetzung einer agilen, datenbasierten Produktentwicklung	37
3.1	Analysephase.....	37
3.1.1	Analyse der Produktentwicklungsprozesse	37

3.1.2	Analyse der Informationssystemarchitektur.....	57
3.2	Konzeptionsphase.....	63
3.2.1	Konzeption des agilen Entwicklungsprozesses.....	63
3.2.2	Konzeption Informationsmodell	86
3.3	Detaillierungsphase.....	95
3.3.1	Methodische Unterstützung der Sprintdurchführung ..	95
3.3.2	Ableitung Datenmodell	110
3.3.3	Benutzung Datenmodell	116
3.4	Implementierungsphase	124
3.4.1	Organisatorische Verankerung	124
3.4.2	Gestaltung eines Implementierungsprozesses.....	139
4	Zusammenfassung	147
5	Literaturverzeichnis	149
A	Anhang	159
A	Ablauforganisatorische Struktur	159
B	Ausführliche Rollenbeschreibungen.....	168
C	Prozessbeschreibung mit Rollenzuordnung.....	176

Abbildungsverzeichnis

ABBILDUNG 1: ORDNUNGSRAHMEN DES FORSCHUNGSPROJEKTS „AGILE_DATA_DEV“	14
ABBILDUNG 2: ARBEITSPLAN FÜR DAS FORSCHUNGSVORHABEN „AGILE_DATA_DEV“	19
ABBILDUNG 3: ORDNUNGSRAHMEN DER ERFOLGSFAKTOREN AGILER PRODUKTENTWICKLUNG.....	25
ABBILDUNG 4: GEGENÜBERSTELLUNG DER EIGENSCHAFTEN DOKUMENTENBASIERTER UND MODELLBASIERTER ENTWICKLUNGSANSÄTZE	28
ABBILDUNG 5: ERFOLGSFAKTOREN FÜR EINEN DATENBASIIERTEN PRODUKTENTWICKLUNGSPROZESS	29
ABBILDUNG 6: AUSZUG DER FUNKTIONSORIENTIERTEN MODELLIERUNG EINER WIND- ENERGIEANLAGE NACH JACOBS ET AL.	32
ABBILDUNG 7: BEISPIELE FÜR MBSE-MODELLIERUNGSWERKZEUGEN.....	33
ABBILDUNG 8: MORPHOLOGIE ZUR BESCHREIBUNG VON ENTWICKLUNGSNETZWERKEN	39
ABBILDUNG 9: MORPHOLOGIE ZUR BESCHREIBUNG VON SYSTEMPARTNERN UND SCHNITTSTELLEN	41
ABBILDUNG 10: EXEMPLARISCHE PROZESSSTRUKTURIERUNG IM KONTEXT VON ENTWICKLUNGSTÄTIGKEITEN	43
ABBILDUNG 11: BEISPIELHAFTES GESCHÄFTSPROZESSE IN DER WINDENERGIEBRANCHE	44
ABBILDUNG 12: MORPHOLOGIE ZUR BESCHREIBUNG VON SCHNITTSTELLEN ZWISCHEN GESCHÄFTSPROZESSEN.....	46
ABBILDUNG 13: EXEMPLARISCHE DARSTELLUNG DER ENTWICKLUNGSRELEVANTEN GESCHÄFTSPROZESSSCHNITTSTELLEN.....	48
ABBILDUNG 14: DIMENSIONEN ZUR BEWERTUNG DES AGILISIERUNGSPOTENZIALS.....	53
ABBILDUNG 15: EXEMPLARISCHER AUSSCHNITT EINER WERTSTROMANALYSE IM KONTEXT DER WINDENERGIEANLAGENENTWICKLUNG	55
ABBILDUNG 16: AKTIVITÄTENKARTEN ZUR UNTERSTÜTZUNG DER FUNKTIONSORIENTIERTEN WERTSTROMANALYSE	56
ABBILDUNG 17: VORGEHEN ZUR GENERIERUNG DER DATENBANK.....	58
ABBILDUNG 18: AUSZUG AUS DEM AUFBAU EINER WINDENERGIEANLAGE.....	61
ABBILDUNG 19: SCHEMATISCHE GEGENÜBERSTELLUNG DER REIFEGRADORIENTIERTEN UND OBJEKTORIENTIERTEN PROZESSSTEUERUNG.....	65

ABBILDUNG 20: DEKOMPOSITION EINES ENTWICKLUNGSPROZESSES IN EINZELNE ENTWICKLUNGSZYKLEN	66
ABBILDUNG 21: GEGENÜBERSTELLUNG EINER PLANGETRIEBENEN UND AGILEN DURCHFÜHRUNG EINES ENTWICKLUNGSPROZESSES	68
ABBILDUNG 22: STACEY MATRIX FÜR DIE ENTSCHEIDUNG ZWISCHEN PLANGETRIEBENEM UND AGILEM VORGEHEN FÜR EINEN ENTWICKLUNGSPROZESS	69
ABBILDUNG 23: ABLAUF EINES SPRINTS.....	71
ABBILDUNG 24: FALLUNTERSCHIEDUNG BEI DER DURCHFÜHRUNG DER ENTWICKLUNGSPROZESSE	72
ABBILDUNG 25: DEKOMPOSITION EINES SYSTEMS "WINDTURBINE" IN MÖGLICHE MODULE	73
ABBILDUNG 26: SCHEMATISCHE DARSTELLUNG EINES SPRINTS DER SYSTEM-ENTWICKLUNG.....	74
ABBILDUNG 27: SCHEMATISCHE DARSTELLUNG EINES SPRINTS DER MODULENTWICKLUNG.....	75
ABBILDUNG 28: IDENTIFIZIERUNG VON PROZESSBAUSTEINEN IN EINER WERTSTROM-ANALYSE	76
ABBILDUNG 29: BESCHREIBUNG UND DOKUMENTATION EINES PROZESSBAUSTEINS ...	78
ABBILDUNG 30: AUFLÖSUNG DER DICHOTOMIE ZWISCHEN GENERIZITÄT UND SPEZIFIZITÄT DER PROZESSBAUSTEINE	80
ABBILDUNG 31: LEGENDE ZUR PROZESSMODELLIERUNGSSPRACHE IN ANLEHNUNG AN DIN 66001	81
ABBILDUNG 32: PROZESSRAHMENMODELL.....	81
ABBILDUNG 33: FORMALES PROZESSMODELL EINER ENTWICKLUNGSSITERATION MIT SYSTEM- UND MODULEBENE.....	83
ABBILDUNG 34: FORMALES PROZESSMODELL EINER ENTWICKLUNGSSITERATION MIT MODULEBENE	85
ABBILDUNG 35: VORGEHEN ZUR GENERIERUNG DER DATENBANK.....	86
ABBILDUNG 36: MODELLIERUNG ZWECKGEBUNDENER SICHTEN IN DER SYSML ÜBER DIAGRAMME NACH	87
ABBILDUNG 37: DIAGRAMMTYPEN IN SYSML.....	88
ABBILDUNG 38: NOTATION, SYNTAX UND SEMANTIK IN SYSML.....	89
ABBILDUNG 39: AUSZUG DES KONZEPTEES ZUR MODELLIERUNG DES INFORMATIONSMODELLS MIT SYSML.....	90

ABBILDUNG 40: MODELLIERUNG DES INFORMATIONSMODELLS ZUR ABBILDUNG DER PRODUKTSTRUKTUR	92
ABBILDUNG 41: MODELLIERUNG DES INFORMATIONSMODELLS ZUR ABBILDUNG DER ABHÄNGIGKEITEN	94
ABBILDUNG 42: VORGEHEN ZUR KONFIGURATION VON ENTWICKLUNGSSPRINTS.....	96
ABBILDUNG 43: ABLEITUNG TECHNISCHER FRAGESTELLUNG ANHAND DER PRÜFKRITERIEN FÜR ANFORDERUNGEN	97
ABBILDUNG 44: OPERATIONALISIERUNG VON FRAGESTELLUNGEN MITTELS KENNZAHLENSYSTEM UND GENERISCHER PRODUKTSTRUKTUR	99
ABBILDUNG 45: SPEZIFIKATION VON MVPs ZUR BEANTWORTUNG TECHNISCHER FRAGESTELLUNGEN	99
ABBILDUNG 46: ERZEUGUNG VON PBS-KETTEN ZUR KONFIGURATION VON ENTWICKLUNGSSPRINTS	100
ABBILDUNG 47: VERGLEICH EINHEITLICHER UND ZYKLUSspezifischer SPRINTLÄNGE	102
ABBILDUNG 48: DIE GESTAFFELTE SPRINTLÄNGE VEREINT DIE VORTEILE EINES EFFIZIENTEN ARBEITENS MIT EINER EFFIZIENTEN INFORMATIONSWERTBEREICHUNG ..	104
ABBILDUNG 49: VERTIKALE SYNCHRONISATION ZWISCHEN DEN ENTWICKLUNGSZYKLEN DURCH KOORDINATIONSPUNKTE.....	105
ABBILDUNG 50: HORIZONTALE SYNCHRONISATION INNERHALB DES ENTWICKLUNGS- ZYKLUS DURCH WECHSELPUNKTE	107
ABBILDUNG 51: CHECKLISTE FÜR DEN WECHSEL DES ENTWICKLUNGSVORGEHENS...	108
ABBILDUNG 52: KOORDINATIONS- UND WECHSELPUNKTE STRUKTURIEREN DIE VERTIKALE UND HORIZONTALE SYNCHRONISATION DES ENTWICKLUNGSVORGEHENS	109
ABBILDUNG 53: VORGEHEN ZUR GENERIERUNG DER DATENBANK.....	110
ABBILDUNG 54: REPRÄSENTATION DES GRAFISCH MODELLIERTEN SYSTEMMODELLS IN ECLIPSE PYPYRUS.....	112
ABBILDUNG 55: AUSZUG AUS DEM QUELLCODE DES ENTWICKELTEN DATENPARSERS	113
ABBILDUNG 56: TRANSFORMATION DES GRAFISCH MODELLIERTEN SYSTEMMODELLS IN DAS FORMAT CD4A.....	115
ABBILDUNG 57: VORGEHEN ZUR GENERIERUNG DER DATENBANK.....	116
ABBILDUNG 58: MONTIGEM FRAMEWORK	118
ABBILDUNG 59: DETAILANSICHT EINES BLADE	120

ABBILDUNG 60: PYTHON-CODE FÜR DIE ERSTELLUNG EINES NEUEN „BLADE“-OBJEKTS	121
ABBILDUNG 61: ANZEIGE EINER TABELLE MIT HEATMAP	123
ABBILDUNG 62: HANDLUNGSFELDER BEI DER GESTALTUNG DER AGILEN ORGANISATION	125
ABBILDUNG 63: ROLLEN IN DER AGILEN ENTWICKLUNG VON WINDENERGIEANLAGEN	128
ABBILDUNG 64: PAARWEISER VERGLEICH FÜR DIE ZUORDNUNG VON PERSONEN ZU ROLLEN	131
ABBILDUNG 65: GEGENÜBERSTELLUNG DER BEWERTUNG VON PLANGETRIEBENER, AGILER UND HYBRIDER PROJEKTORGANISATION	132
ABBILDUNG 66: ÜBERSICHT DER ROLLEN IN AGILEN PROJEKTTEAMS.....	134
ABBILDUNG 67: VORGEHEN ZUR PROJEKTBEGLEITENDEN EINBINDUNG ENTWICKLUNGS-EXTERNER UNTERSTÜTZUNG	135
ABBILDUNG 68: PROJEKTORGANISATION FÜR KLEINERE WEA-UNTERNEHMEN	136
ABBILDUNG 69: AUSWAHLMÖGLICHKEIT DER ORGANISATIONSFORM.....	137
ABBILDUNG 70: KONZEPT EINER DUALEN ORGANISATION.....	138
ABBILDUNG 71: UMSETZUNG DER DUALEN ORGANISATION MIT DER BESCHRIEBENEN AGILEN PROJEKTORGANISATION	138
ABBILDUNG 72: PROZESS FÜR DIE IMPLEMENTIERUNG UND INTENSITÄT DER VERURSACHTEN VERÄNDERUNGEN	140
ABBILDUNG 73: MATRIX ZUR AUSWAHL VON KOMMUNIKATIONSMETHODEN.....	142
ABBILDUNG 74: ABWEHRMECHANISMEN VON MITARBEITENDEN BEI VERÄNDERUNGEN	143

Abkürzungsverzeichnis

Bspw.	beispielsweise
BTP	Build to Print
BTS	Build to
EEG	Erneuerbare-Energien-Gesetz
EU	Europäische Union
MW	Megawatt
PLM	Product Lifecycle Management
PBS	Prozessbaustein
SysLM	System Lifecycle Management
VUCA	Volatility, Uncertainty, Complexity, Ambiguity
WEA	Windenergieanlage
WZL	Werkzeugmaschinenlabor
z.B.	zum Beispiel

1 Einleitung

1.1 Motivation

Zur Umsetzung des Pariser Klimaschutzabkommens¹ ist die Nutzung regenerativer Energien, insbesondere die Nutzung von Windenergie, von zentraler Bedeutung. Der Ausbau der Windenergie in Deutschland ist in den letzten Jahren stetig vorangeschritten. Windenergie bildet heute einen zentralen Baustein der Stromversorgung in Deutschland und nimmt auf dem Weg zur Klimaneutralität eine Schlüsselrolle ein. Bis Ende des Jahrzehnts müssen Windenergieanlagen nicht nur konventionelle Kraftwerke ersetzen, sondern auch zusätzlich den Strombedarf im Rahmen der Sektorenkopplung decken².

Ebenfalls verkörpern die aktuellen geopolitischen Entwicklungen die besondere Relevanz der Windenergiebranche als zentralen Baustein zur Erreichung des übergeordneten energiepolitischen Ziels der Versorgungssicherheit in der Bundesrepublik Deutschland. Die Windenergiebranche, insbesondere die Entwicklung von Windenergieanlagen, muss befähigt werden, um diese ambitionierten Ziele und die gestiegenen Anforderungen optimal begegnen zu können.

Regenerative Energien sind in Zeiten knapper werdender Ressourcen das Zukunftsthema. Die besondere energiepolitische Relevanz wird innerhalb des Koalitionsvertrags der Bundesregierung deutlich: Demnach sollen rund 80 % des deutschen Stromverbrauchs im Jahr 2030 mittels erneuerbarer Energien gedeckt werden.³ Unter den erneuerbaren Energien leistet zurzeit die Windenergie einen signifikanten Beitrag. Im dritten Quartal des Jahres 2021 entfielen knapp 17 % der gesamten Stromeinspeisung in der Bundesrepublik auf Windenergieanlagen. Ferner verfolgt die Bundesregierung ambitionierte Ausbauziele. Dazu sollen unter anderem 2 % der Landesfläche für Windenergie ausgewiesen werden. Zur Umsetzung dieses energiepolitischen Ziels muss der Ausbau von Windenergieanlagen forciert und die Entwicklung von neuen Anlagen fokussiert werden.

¹ United Nations 2015.

² Quentin 2021.

³ Die Bundesregierung 2021.

Folglich birgt die Windenergiebranche neben dieser hohen energie- und klimapolitischen Relevanz ebenfalls enormes wirtschaftliches Potenzial in sich. Die Erschließung dieses wirtschaftlichen Potenzials ist jedoch gleichzeitig mit zahlreichen Herausforderungen verbunden. Genau wie das gesamte heutige Geschäftsumfeld sehen sich auch Hersteller von Windenergieanlagen mit zunehmender Volatilität, Unsicherheit, Komplexität und Ambiguität (VUCA-Umwelt) konfrontiert. Im Maschinen- und Anlagenbau manifestiert sich diese maßgeblich durch eine fortschreitende Globalisierung und eine rasante Technologieentwicklung. Dabei ist die Windenergiebranche im Speziellen geprägt von kürzer werdenden Produktlebenszyklen, den gesteigerten Forderungen nach individuellen, auf die jeweiligen Standortbedingungen maßgeschneiderten Produkten sowie steigendem Kostendruck. Um auf dieses dynamische Umfeld bestmöglich zu reagieren und gleichzeitig wettbewerbsfähige Produkte anbieten zu können, müssen die Hersteller von Windenergieanlagen neue Strategien entwickeln und. Diese Strategien müssen eine optimale Flexibilisierung des entsprechenden Entwicklungsprozesses bei gleichzeitig sinkenden Stromerzeugungskosten ermöglichen. Es gilt also, im Sinne eines agilen Vorgehens, Transparenz und Veränderungsgeschwindigkeit zu erhöhen und den Einsatz der entwickelten Systeme zu beschleunigen. Jene Herausforderungen werden verstärkt durch die zunehmende Digitalisierung in produzierenden Unternehmen, insbesondere auch in der Forschung und Entwicklung im Maschinen- und Anlagenbau. Dort dominieren die disruptiven Einflussfaktoren einer rasant ansteigenden Menge an Daten und Informationen in „Velocity, Volume, Value, Variety, Veracity und Variability“ und induzieren einen signifikanten Komplexitätsanstieg.

Um Unternehmen der Windenergiebranche, insbesondere Hersteller von Windenergieanlagen, im Umgang mit den Herausforderungen der VUCA-Umwelt zu unterstützen, müssen Konzepte entwickelt werden, um Entwicklungsprozesse flexibel und transparent zu gestalten. Gleichzeitig müssen diese Prozesse eine zunehmende Menge an Informationen berücksichtigen und verarbeiten können, die im Kontext der Entwicklung einer Windenergieanlage anfallen. Es gilt, die zunehmende Menge an Daten zu wertschöpfungsorientierten Informationen zu verarbeiten und durch datenbasierte Entwicklungsprozesse Effizienzvorteile zu erschließen. Einen erfolgversprechenden Ansatz zur Bewältigung

dieser Herausforderungen stellt die Erarbeitung eines agilen, datenbasierten Entwicklungsprozesses für Windenergieanlagen dar.

1.1.1 Motivation aus Sicht der Praxis

Windenergieanlagenhersteller benötigen eine maximale Effektivität und Effizienz in der Entwicklung neuer Anlagevarianten unter volatilen Anforderungen. Weltweit werden jährlich etwa 20.000 bis 30.000 neue Windenergieanlagen (WEA) installiert. Je nach Größe des Unternehmens beträgt die Zahl der zu produzierenden WEA pro Jahr zwischen ca. 50 Stück und max. 2.000 Stück, was nach industriellen Maßstäben Kleinst- oder Kleinserien entspricht. Zeitgleich sind die Anforderungen an die Individualisierung der Anlagen in Abhängigkeit von gesetzlichen und politischen Rahmenbedingungen (z. B. normative Netzanforderungen, Sicherheits- und Kennzeichnungsanforderungen, Netzanschlussbedingungen etc.) sowie von kunden- und marktabhängigen Wünschen und Anforderungen (max. Transportabmessungen, klimatische Bedingungen, Netzfrequenz und -spannung, Schallanforderungen etc.) äußerst hoch. Folglich sehen sich die Hersteller von WEA heutzutage auf dem globalen Windmarkt zahlreichen und rasch ändernden Anforderungen an die Produkte ausgesetzt. Um in wirtschaftlich sinnvollen Stückzahlen zu fertigen, ist eine internationale Ausrichtung auf global verteilte Märkte notwendig. Dabei sieht sich die Branche jedoch mit teils enorm volatilen Märkten konfrontiert, die schnelle Markteintrittsstrategien und damit lokale Produktadaptionen notwendig machen. Bedingt durch den wachsenden Wettbewerbsdruck und die zunehmende Sättigung der Kernmärkte europäischer Hersteller sind die Kundenanforderungen im internationalen Umfeld möglichst individuell zu bedienen⁴. Aufgrund der standortübergreifenden Einbindung aller Fachabteilungen, eines stetig wachsenden Zuliefernetzwerks durch „Local-content“-Anforderungen, der verkürzten Entwicklungszeiten sowie des steigenden Kostendrucks bei erhöhter Produktqualität wird das Entwicklungsumfeld zusätzlich verschärft. Neben einer Zunahme der Produktkomplexität führen diese Faktoren zu einer steigenden Komplexität in den bestehenden Entwicklungsprozessen⁵. Einher mit diesem Komplexitätsanstieg in Entwick-

⁴ Schmelzer und Sesselmann 2004, S. 472.

⁵ Komus et al. 2020.

lungsprozessen geht eine Steigerung der Daten- und Informationsmengen, die durch Windenergieanlagenhersteller erfasst, systematisiert und verarbeitet werden müssen. Um effizient, zielorientiert und letztlich erfolgreich neue Anlagen zu entwickeln, bedarf es einer ausreichenden, zuverlässigen und konsistenten Datengrundlage, um Entwicklungsprozesse zielgerichtet zu unterstützen.

Die skizzierte Inhomogenität der Märkte wird durch die zunehmend dynamischere Entwicklung politischer Mehrheiten und konjunktureller Einflüsse verstärkt. Nahezu jedes Projekt verlangt letztendlich eine mehr oder weniger individuell zusammengestellte Windenergieanlage zur optimalen wirtschaftlichen Ausnutzung der jeweiligen gegebenen Rahmenbedingungen, sodass für die Hersteller ein hohes Maß an Flexibilität in ihren Produkten zwingend erforderlich ist. Eine Windenergieanlage stellt jedoch ein komplexes technisches System mit einer hohen Abhängigkeit und Wechselwirkungen der Komponenten untereinander dar. Marginale Änderungen an einer Komponente induzieren oftmals umfangreiche Änderungs- und Anpassungsmaßnahmen an verbundenen Komponenten.

Jene Herausforderungen werden durch immer kürzere Produktlebenszyklen im Maschinen- und Anlagenbau intensiviert. So haben sich speziell in der Windenergiebranche die Technologielebenszyklen neuer Anlagen im Vergleich zur Marktsituation von vor wenigen Jahren drastisch geändert. Neue Anlagevarianten veralten heute sehr viel schneller als früher. Dies stellt Hersteller am Markt vor die Herausforderung, dass sie kontinuierlich und kurzzyklisch (d. h. fast jährlich) auf diese sich ändernde Wettbewerbssituation reagieren müssen, obwohl die Entwicklungszeit für eine neue Anlagengeneration aktuell noch zwischen 18 und 24 Monaten beträgt. Zugleich sind die Hersteller von Windenergieanlagen auf dem globalen Windmarkt zahlreichen und sich rasch ändernden Anforderungen an die Produkte ausgesetzt. Volatile Marktumfeld fordern schlussfolgernd Agilität in Entwicklungstätigkeiten, schnelle Markteintrittsstrategien und lokale Produktadaptionen. Bedingt durch den wachsenden Wettbewerbsdruck und die zunehmende Sättigung der Kernmärkte europäischer Hersteller sind die Kundenanforderungen möglichst individuell zu bedienen⁶. Aufgrund der standortübergreifen-

⁶ Schuh 2002.

den Einbindung aller Fachabteilungen, eines stetig wachsenden Zuliefernetzwerks durch „Local-content“-Anforderungen, der verkürzten Entwicklungszeiten sowie des steigenden Kostendrucks bei erhöhter Produktqualität wird das Entwicklungsumfeld zusätzlich verschärft.

Letztlich führen diese Faktoren zu einer steigenden Komplexität in den bestehenden Entwicklungsprozessen. Die komplexen Wechselwirkungen aus Volatilität, Dynamik und Unsicherheit führen dazu, dass bei der Initiierung eines Entwicklungsprojekts einer Anlage die Wettbewerbssituation, die Marktanforderungen sowie die Zielmärkte noch gar nicht abschließend definierbar sind. Das Risiko dieser „Moving Targets“, d. h. sich ändernder Anforderungen, kann dazu führen, dass am Ende eines Entwicklungsprozesses das initial definierte WEA die Kunden- und Marktanforderungen nicht mehr hinreichend erfüllt. Um auf dieses dynamische Umfeld bestmöglich zu reagieren und gleichzeitig wettbewerbsfähige Produkte anbieten zu können, müssen die Hersteller von Windenergieanlagen neue Strategien entwickeln. Diese müssen eine optimale Flexibilisierung des entsprechenden Entwicklungsprozesses bei gleichzeitig sinkenden Herstellungs-, Betriebs- und letztlich Stromgestehungskosten ermöglichen.

Zusammenfassend muss der Entwicklungsprozess einer Windenergieanlage aus Sicht der Praxis also stärker kundenorientiert sein und damit flexibel auf sich ändernde Produkthanforderungen reagieren können sowie einen kontinuierlichen Informationsaustausch zwischen den Entwicklungsabteilungen bspw. bei der Konstruktion von voneinander abhängigen Komponenten ermöglichen.

Einen erfolversprechenden Ansatz zur Bewältigung dieser komplexen Herausforderungen stellt die Erarbeitung eines agilen, datenbasierten Entwicklungsprozesses für Windenergieanlagen dar. Hierdurch sollen schnellere und aufwandsärmere Anpassungen der zu entwickelnden Produkte an sich ändernden Rahmenbedingungen und Anforderungen im laufenden Entwicklungsprozess ermöglicht werden. Bislang besteht in der Praxis jedoch mangelnde Kenntnis, wie ein agiler, datenbasierter Entwicklungsprozess auszugestalten und zu implementieren ist.

1.1.2 Motivation aus Sicht der Forschung

Wie bereits aus praxisorientierter Sicht dargestellt, stellt die heutige VUCA-Umwelt hohe Anforderungen an die Produktentwicklungspro-

zesse im Bereich der Windenergieanlagen. Entwicklungsprozesse müssen zunehmend in der Lage sein, eine stärkere Kundenorientierung zu fokussieren und flexibel auf sich dynamisch ändernde Produkthanforderungen reagieren zu können. In Anbetracht des wachsenden Digitalisierungstrends und der damit verbundenen gesteigerten Menge und Vielfalt an Daten und Informationen zeichnet sich ein Komplexitätsanstieg ab. Zusätzlich stellt eine Windenergieanlage ein komplexes System dar und weist zahlreiche voneinander abhängige Komponenten auf. Der Entwicklungsprozess muss schlussfolgernd einen kontinuierlichen Informationsaustausch zwischen Entwicklungsabteilungen für den Entwurf eben dieser Komponenten ermöglichen.

Die Mehrzahl der Unternehmen im Bereich des Maschinen- und Anlagenbaus, insbesondere der Windenergiebranche, verwendet verschiedene plangetriebene Ansätze innerhalb ihrer Produktentwicklung. Plangetriebene Entwicklungsprozesse gehen u. a. auf das Stage-Gate-Modell nach COOPER⁷ oder das Wasserfallmodell⁸ zurück. Speziell im Maschinen- und Anlagenbau beschreibt das weitverbreitete V-Modell (VDI 2206) bspw. die Produktentwicklung durch Problemzerlegung, -lösung und -integration der Lösungen in den Bereichen Mechanik, Elektrotechnik und Softwaretechnik⁹. Allen diesen Entwicklungsprozessmodellen ist gemeinsam, dass sie ein deterministisches und konsekutives Entwicklungsvorgehen skizzieren. Hieraus resultiert ein steigender Bedarf nach größerer Flexibilität, um Unternehmen die Möglichkeit zu geben, auf sich ändernde Rahmenbedingungen zu reagieren und einen durchgängigen, iterativen Informationsfluss zwischen verschiedenen am Entwicklungsprozess partizipierenden Bereichen sicherzustellen.

In den letzten Jahren haben viele Unternehmen aus dem Maschinen- und Anlagenbau begonnen, agile Entwicklungsansätze aus der Softwareentwicklung zu adaptieren. Agile Ansätze fokussieren interaktive, inkrementelle und anpassungsfähige Prozessabläufe und versprechen flexiblere Reaktionen auf sich ändernde Anforderungen bei gleichzeitig kürzeren Entwicklungszeiten. Dabei stützen sich Unternehmen zumeist auf Ansätze wie Scrum¹⁰ oder das Agile-Stage-Gate Hybrid Modell¹¹.

⁷ Cooper 1990, S. 44.

⁸ Royce 1970.

⁹ Verein Deutscher Ingenieure e.V. 2020.

¹⁰ Schwaber 1997.

¹¹ Cooper und Sommer 2016, S. 513.

Aus Sicht der Forschung beschreiben diese Ansätze zwar eine agile Produktentwicklung, transformieren aber nicht systematisch den bestehenden Produktentwicklungsprozess in ein neues Prozesskonzept. Aus diesem Grund sind diese agilen Prinzipien für Unternehmen bisher nicht ohne Weiteres anwendbar. Ebenfalls berücksichtigen sie nicht den hohen Bedarf und die Anforderungen des Informationsaustauschs, der bei der Entwicklung komplexer technischer Systeme, wie etwa Windenergieanlagen, aufgrund der hohen Interdependenz der Komponenten notwendig ist.

Typischerweise stellen agile Ansätze einen prototypischen Zielprozess für die Softwareentwicklung dar. Der Übergang von plangetriebenen zur agilen Produktentwicklung und die Berücksichtigung branchenspezifischer Besonderheiten ist zumeist nicht Teil etablierter agiler Entwicklungsansätze und in der gegenwärtigen Literatur nur unzureichend betrachtet. Unternehmen benötigen eine Methodik, um von ihren derzeitigen plangetriebenen Prozessen zu einer auf ihre Bedürfnisse zugeschnittenen agilen Entwicklung überzugehen.

Ferner ist in der wissenschaftlichen Literatur nur unzureichend beschrieben, wie eine optimierte Datengrundlage und -verfügbarkeit agile Prozesse im Maschinen- und Anlagenbau ermöglichen kann. Aus wissenschaftlicher Sicht ist es von hoher Relevanz zu untersuchen, wie eine echtzeitfähige, datenbasierte Entwicklung neuer Produktvarianten unter Berücksichtigung volatiler Markt- und Kundenanforderungen auszugestalten ist. Es fehlt an einem Vorgehen zur Erstellung eines einheitlichen Datenmodells als Grundlage für eine datenbasierte Entscheidungsunterstützung im Kontext agiler Entwicklungsprozesse.

Um diesen Herausforderungen zu begegnen, ist es notwendig, einen Ansatz zu entwickeln, der es Unternehmen der Windenergiebranche ermöglicht, ein entsprechendes Vorgehen für die eigenen Entwicklungstätigkeiten, unter Berücksichtigung des spezifischen Unternehmenskontexts, abzuleiten. Dieser Ansatz soll auf der Identifikation individueller Potenziale und Restriktionen für die Einführung und Anwendung eines agilen, datenbasierten Entwicklungsprozesses basieren.

1.2 Potenziale in der Produktentwicklung der Windenergiebranche

Bezüglich der Produktentwicklung stoßen aktuelle, deterministische und plangetriebenen Entwicklungsansätze in Anbetracht der skizzierten Herausforderungen im Windenergieanlagenbau an Grenzen. Wie gezeigt, ist den plangetriebene Entwicklungsprozessmodellen gemein, dass sie ein rigides Vorgehen präferieren, das Unternehmen häufig hindert, bedarfsgerecht auf sich ändernde Rahmenbedingungen reagieren zu können. Wie ein Großteil des Maschinen- und Anlagenbaus verwendet auch die Mehrheit der Unternehmen der Windenergiebranche plangetriebene Ansätze in der Produktentwicklung. Gerade Unternehmen auf dem Windenergiemarkt haben aufgrund der Inflexibilität in ihren bisherigen Entwicklungsprozessen jedoch zunehmend Schwierigkeiten, dynamische Ziele zu erreichen.

Das beschriebene Marktumfeld bedingt, dass es für die Hersteller von Windenergieanlagen zunehmend schwieriger wird, die relevanten Kundenanforderungen und Marktbedingungen zu Beginn eines Entwicklungsprojektes vollumfänglich zu identifizieren.

Erfolgversprechend scheinen hingegen die in der Softwareentwicklung bereits erfolgreich eingesetzten agilen Vorgehensweisen. So beschreibt etwa das agile Projektmanagement-Framework Scrum¹² die Entwicklung als eine Abfolge von Sprints mit entsprechenden Aufgaben¹³. Zum Ziel hat diese Vorgehensweise, ein lieferbares Produktinkrement zu entwickeln, das an definierten Zielen getestet und mit dem Kunden diskutiert werden kann. In diesem Zusammenhang erfordert bspw. „Extreme Programming“ nach BECK eine iterative Entscheidungsfindung, welche Anforderungen im nächsten Release umgesetzt werden, wobei in jeder Iteration alle klassischen Phasen der Produktentwicklung durchlaufen werden¹⁴. Kanban visualisiert den Prozess und verbessert diesen inkrementell, um das jeweilige Risiko zu minimieren¹⁵. Eine Kombination aus Scrum und Kanban, Scrumban, stellt Sprints in den visualisierten und begrenzten Kanban-Kontext¹⁶.

¹² Pichler 2011.

¹³ Schwaber 1997.

¹⁴ Beck 2010.

¹⁵ Anderson 2010.

¹⁶ Ladas 2008.

Die Skalierung der Grundprinzipien der agilen Produktentwicklung auf komplexere Projekte mit mehreren Entwicklungsteams wird durch Scrum-Frameworks wie Large Scale Scrum, SAFe oder Scrum@Scale beschrieben. Large Scale Scrum (LeSS) bietet ein Framework, das den Scrum-Ansatz auf zwei bis acht Teams überträgt, die ganzheitlich an einem gemeinsamen, insgesamt lieferbaren Produktinkrement arbeiten, das mit dem Kunden bewertet werden kann¹⁷. SAFe (Scaled Agile Framework) beschreibt die Anwendung eines agilen Entwicklungsansatzes auf Programmebene und die Synchronisation zwischen verschiedenen Entwicklungsteams¹⁸. Der Scrum@Scale-Ansatz führt ein Scrum of Scrums ein, das die Ergebnisse von unabhängig voneinander arbeitenden Teams in ein ganzheitliches Produktinkrement integriert¹⁹. Einige agile Entwicklungsansätze stellen sich den spezifischen Herausforderungen des Maschinen- und Anlagenbaus, da sie sich mit der hohen Komplexität seiner aus interdisziplinären, voneinander abhängigen Komponenten bestehenden Produkte auseinandersetzen. Die Flexible Produktentwicklung von SMITH²⁰, das Agile-Stage-Gate Hybrid Modell von COOPER und SOMMER²¹ und das Scaling Agile Development in Mechatronic Organizations von EKLUND und BERGER²² übertragen die Prinzipien der agilen Entwicklung auf die Entwicklung von Mechatronik. Aufgrund unterschiedlicher Restriktionen in der Entwicklung mechatronischer Produkte, wie z.B. Sicherheitsnormen oder Energieeffizienzstandards, insbesondere in der Windenergiebranche, lassen sich diese Ansätze jedoch nicht ohne weiteres übertragen. Zusätzlich beschreiben diese Ansätze zwar Prinzipien und Methoden der agilen Produktentwicklung, bieten allerdings kein Vorgehensmodell oder eine Verfahrensanleitung, wie konkret bestehende plangetriebene Produktentwicklungsansätze um agile Prinzipien angereichert- bzw. vollständig in agile Prozesse überführt werden können. Die Operationalisierung der Ansätze und deren Implementierung in Unternehmensstrukturen mit bestehenden Prozessen stellt zahlreiche Unternehmen der Windenergiebranche aktuell vor Herausforderungen.

¹⁷ Larman und Vodde 2009.

¹⁸ Stojanov et al. 2015.

¹⁹ Dingssoeyr et al. 2019.

²⁰ Smith 2007.

²¹ Cooper und Sommer 2016.

²² Eklund und Berger 2017.

1.2.1 Effektive Produktentwicklung durch agile Entwicklungsmethoden

Nach FARNBACH ist die Anwendung eines detailliert geplanten, traditionellen Stage-Gate-Prozesses in volatilen und turbulenten Umfeldern nicht nur ineffektiv, sondern kontraproduktiv und kann die heutige Marktdynamik längst nicht mehr vollständig adressieren²³. Ein Ansatz, um die Effektivität in der Entwicklung zu steigern sowie der Volatilität der Kundenanforderungen zu begegnen, stellt die angeführte agile Produktentwicklung dar²⁴. Die agile Produktentwicklung ist eine Menge iterativer Entwicklungsmethoden, die nachweislich zu einem höheren Zielerreichungsgrad führen. Somit können sie den Projekterfolg deutlich steigern²⁵. Neben einer verstärkten Kollaboration in cross-funktionalen Teams und einer reduzierten Vorausplanung am Anfang eines Projektes, setzt die agile Produktentwicklung auf iterative Entwicklungszyklen und eine kontinuierliche Validierung der Entwicklungsergebnisse mit definierten Kundengruppen. Damit unterstützt eine agile Vorgehensweise in der Produktentwicklung insbesondere die Adaptionfähigkeit des Entwicklungsprozesses. Durch eine kontinuierliche Adjustierung auf sich ändernde Randbedingungen, ermöglicht eine agile Vorgehensweise zudem den Umgang mit unvollständigen oder teilweise unbekanntem Anforderungen zu Beginn eines Entwicklungsprojektes²⁶.

Da der Einsatz rein agiler Methoden nicht für jeden Entwicklungsprozess geeignet ist²⁷, empfiehlt sich stattdessen die Nutzung von integrierten Prozessen als eine Kombination aus agilen und plangetriebenen Entwicklungsvorgehen. Somit kann ein ganzheitlicher agiler Entwicklungsprozess gestaltet werden, welcher auf der einen Seite flexibel ist, gleichzeitig auf der anderen Seite durch die plangetriebenen Elemente auch die notwendige Stabilität garantiert. Die Einführung agiler Ansätze bedeutet also nicht, dass gänzlich auf die Vorteile einer plangetriebenen Entwicklung verzichtet werden muss.

²³ Farnbach 2011, 37 ff.

²⁴ Komus et al. 2020, S. 3.

²⁵ Serrador und Pinto 2015, 1 ff.

²⁶ VersionOne 2017, S. 8.

²⁷ Komus et al. 2020, S. 20.

1.2.2 Effiziente Produktentwicklung durch eine echtzeitfähige Datenverfügbarkeit

Die Herausforderungen resultierend aus der oben beschriebenen systematischen Aufnahme von Kundenanforderungen für eine agile Produktentwicklung als auch die Steuerung eines durch bspw. „Local-content“-Anforderungen wachsendes Entwicklungs- und Zuliefernetzwerks, können nur bewältigt werden, wenn alle benötigten Produktdateninformationen in durchgängiger, einheitlicher und stets aktueller Form vorliegen. Gerade in der Qualität der Informationen sowie deren Verfügbarkeit für alle am Entwicklungsprozess beteiligten Stakeholder liegen bei gemeinsamer und simultaner Nutzung wesentliche Potenziale für den Entwicklungsprozess. Insbesondere in der frühen Phase der Wertschöpfung, in der maßgeblich Stammdaten zu Produkten generiert und verändert werden, sind die Effektivität und die Effizienz der Prozesse stark von der Nutzung verfügbarer Informationen abhängig²⁸. Schließlich werden unternehmerische Ressourcen nur dann effizient eingesetzt, wenn Redundanzen und Inkonsistenzen in der Datenhaltung vermieden werden. Die an den Geschäftsprozessen teilnehmenden internen und externen Akteure sollten folglich auf eine einheitliche, gemeinsame Datenbasis zurückgreifen. In diesem Kontext gilt es, ein digitales Abbild der Windenergieanlagen zu erzeugen, einen sogenannten „Digitalen Schatten“. Der Digitale Schatten ist ein virtuelles Abbild von relevanten Daten in der Produktion, Entwicklung und angrenzenden Bereichen. Die Daten des Digitalen Schattens beinhalten die durchlaufene Historie, den aktuellen Status und mögliche Wege und Prozesse eines angestrebten Status²⁹.

Der Konzeptionierung einer solchen Datenbasis sowie der auf dieser Datenbasis aufbauenden Geschäftsprozesse, widmet sich das Forschungsgebiet des Product Lifecycle Managements (PLM). Dabei werden Konzepte für eine einheitliche Verwaltung von Produktdaten, für die Integration von Prozessen sowie für ein Anwender-Rechtssystem über den gesamten Produktlebenszyklus definiert³⁰. Nach SCHUH umfasst ein vollständiges Produktlebenszykluskonzept einen Ansatz zur ganzheitlichen, unternehmensübergreifenden Verwaltung und Steuerung aller

²⁸ Schuh 2015, S. 19.

²⁹ Bauernhansl et al. 2016.

³⁰ Schuh 2015, S. 20.

Produktdaten und Prozesse des gesamten Produktlebenszyklus entlang der erweiterten Logistikkette³¹. Das PLM unterstützt die abteilungs- und standortübergreifende Verwaltung und zielgerichtete Bereitstellung aller während des Produktlebenszyklus erzeugten Daten und Informationen und integriert auf diese Weise Menschen, Prozesse und Informationen³². Eine solche gemeinsame Struktur für Produktinformationen, die allen Lebenszyklusphasen einer Windenergieanlage gerecht wird und die relevanten informationsspezifischen Sichten auf das Produkt vereint, ermöglicht eine anforderungsgerechte und systematische Bereitstellung von Informationen. EIGNER beschreibt diese zielgerichtete Bereitstellung von Produktinformationen detailliert in verschiedenen Produktstrukturen und Sichten auf das Produkt. Ein Resultat dessen sind verschiedene Stücklistenarten, die ebenfalls charakterisiert und verglichen werden können³³. Einen aktuellen Lösungsansatz für die Fragestellungen rund um die Sichten auf das Produkt entlang dessen Lebenszyklus bietet darüber hinaus das System Lifecycle Management (SysLM)³⁴. Im Gegensatz zum PLM, das aus der mechanischen Produktentwicklung stammt, erweitert SysLM diesen Gedanken. Hier liegt ein stärkerer Fokus auf der Modellierung des Gesamtsystems, einschließlich verschiedener Entwicklungsdomänen sowie der Erfassung von Informationen zu Anforderungen und Konzepten. Für ein lebenszyklusübergreifendes Management von Produktinformationen muss also eine Datenstruktur vorliegen, die sowohl den agilen Entwicklungscharakteristika genügt, als auch die Informationen in den Geschäftsprozessen aller beteiligter Akteure im Sinne des PLM zur Verfügung stellen kann³⁵. Nur mit einer durchgängigen Datenstruktur können die oben genannten Potenziale einer agilen und datengetriebenen Produktentwicklung mit denen einer stets aktuellen, konsistenten und redundanzfreien Informationslage im Sinne einer „Single Source of Truth“ realisiert werden.

Trotz des enormen Potenzials, das agile und datenbasierte Ansätze in Sachen Effizienz, Effektivität und Flexibilität in sich bergen, existiert ins-

³¹ Schuh 2015, S. 20.

³² Eigner und Stelzer 2009, S. 23.

³³ Eigner und Stelzer 2009, 42 ff.

³⁴ Sendler 2013, 16 ff.

³⁵ Cha 2014, S. 421.

besondere in der Windenergiebranche bisher kein agiles Prozessmodell oder standardisiertes Vorgehensmodell zur Nutzung agiler Entwicklungsansätze im Kontext der Entwicklung von Windenergieanlagen. Es gilt die Potenziale in der Produktentwicklung durch die Nutzung agiler Ansätze zu nutzen.

Konsequenterweise wurde der wesentliche Befähiger zur systematischen Optimierung der Entwicklungsprozesse innerhalb der Windenergiebranche identifiziert in Form einer effektiven Produktentwicklung, durch den Einsatz agiler Methoden, in Kombination mit einer effizienten Produktentwicklung, durch eine echtzeitfähige Datenverfügbarkeit.

1.3 Zielsetzung und Aufbau

Basierend auf den dargestellten Potenzialen für Praxis und Wissenschaft wurde das Forschungsvorhaben „Agile_Data_Dev“ formuliert. Dessen Ziel war die Entwicklung eines agilen, datenbasierten Entwicklungsprozesses, verbunden mit der Erarbeitung eines Leitfadens im Sinne einer Vorgehensbeschreibung zur Implementierung in der Windenergiebranche. Der Zielprozess für die Produktentwicklung zeichnet sich durch die Integration von agilen und plangetriebenen Methoden sowie deren datenbasierten Unterstützung aus. Somit soll (1) eine Steigerung der Effektivität durch eine zielgerichtete Adressierung von Anforderungen im Sinne des agilen Gedankens erreicht werden, (2) eine Verbesserung der Effizienz durch die Vermeidung von ungewollten Iterationen aufgrund einer verbesserten Datentransparenz unterstützt werden, (3) eine datenbasierte Entscheidungsunterstützung durch eine durchgängige Daten- und Systemarchitektur ermöglicht werden und (4) eine Verkürzung der Time-to-Market bei gleichzeitiger Steigerung des Market-Fit erreicht werden.

Der in Abbildung 1 dargestellte Ordnungsrahmen des Forschungsprojekts identifiziert die zu betrachtenden Handlungs- und Themenfelder, um diese Projektziele zu erreichen. Das Zielbild setzt sich dabei aus zwei Handlungsfeldern, der Analyse der Prozesse und Tätigkeiten sowie der Analyse der Informationssystemarchitektur und Datenstrukturen, sowie dem Handlungsfeld zur Gestaltung des agilen, datenbasierten Entwicklungsprozesses für Windenergieanlagen zusammen. Prozessseitig gilt es zunächst das Entwicklungsnetzwerk zu beschreiben und dort Systempartner und Schnittstellen zu bestimmen. In Anbetracht

der Komplexität und Interdisziplinarität der Fragestellungen innerhalb eines Entwicklungsprojekts in der Windenergiebranche erfolgt diese unter Einbezug verschiedener Akteure, die im Gesamtsystem der Energieerzeugung agieren. Konkret erfolgt die WEA-Entwicklung zumeist in einem Netzwerk, in dem Entitäten wie Hersteller, Zulieferer und Forschungseinrichtungen kooperativ an Konzeption, Konstruktion und Optimierung der WEA arbeiten.

Aufbauend ist eine Geschäftsprozessanalyse durchzuführen, um relevante Geschäftsprozesse zu ermitteln und Schnittstellen zu identifizieren. Abschließend ist eine funktionsorientierte Wertstromanalyse durchzuführen, um Prozesse zu modellieren und Tätigkeiten sowie Aktivitäten zu bestimmen. Informationsseitig müssen die am Prozess beteiligten Systeme, Objekte und Personen und deren Informationsbedarf identifiziert werden. Aufbauend auf den gewonnenen Erkenntnissen kann der agile, datenbasierte Entwicklungsprozess für Windenergieanlagen gestaltet werden. Zentrale Handlungsfelder sind die Definition agiler Prozessbausteine, die Ermöglichung einer Prozesskonfiguration und die Umsetzung eines Daten- und Informationsmodells.

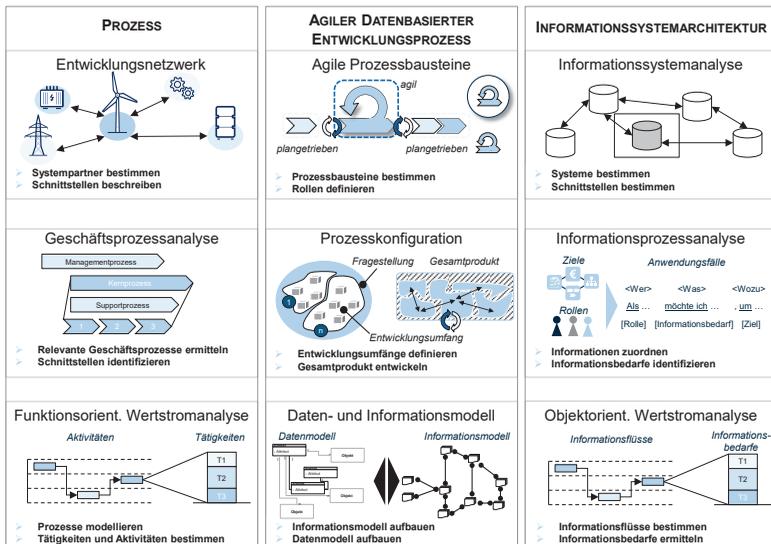


Abbildung 1: Ordnungsrahmen des Forschungsprojekts „Agile_Data_Dev“

Die übergeordneten Ziele des Forschungsvorhabens waren insbesondere, die Entwicklungszeit, Fehlentwicklungen und damit spezifischen

Kosten innerhalb der Entwicklung einer Windenergieanlage zu senken, um den wirtschaftlichen Ausbau der Windenergie zu forcieren. Der im Projekt entwickelte agile, datenbasierte Entwicklungsprozess ermöglicht es Unternehmen der Windenergiebranche, sich weltweit immer stärker individualisierenden Marktanforderungen zielgerichteter stellen zu können. Durch die Anwendung dessen werden Unternehmen in die Lage versetzt, den wirtschaftlichen Ausbau der Windenergie zu befähigen und den Technologiestandort Deutschland zu stärken. Dazu gilt es im Weiteren theoretische Modelle zur Bewertung, Neuentwicklung und Optimierung von Windenergieanlagen zu entwickeln. In diesem Zusammenhang wurde ein Entwicklungsprozess erarbeitet, welcher anhand ausgewählter Teilprojektumfänge in einem vorwettbewerblichen Kontext validiert wurde. Somit konnte unmittelbar getestet werden, inwiefern eine Prozessflexibilisierung zur Adressierung der zuvor genannten Herausforderungen bei der Entwicklung von Windenergieanlagen erfolgen kann. Dafür wurden Konzepte zur Verknüpfung unterschiedlicher Modelle und Datenbasen erarbeitet. Zur Stärkung der Aussagekraft dieser Modelle wurden Initiativen zur Modellierung und Simulation von Komponenten und kompletten Windenergieanlagen bzw. Windparks zur vereinfachten Weiterentwicklung bestehender Konzepte sowie zur Entwicklung von neuen Konzepten unternommen. Um einen agilen, datenbasierten Entwicklungsprozess modellieren zu können und folglich um das Verfahren einsetzen und erproben zu können, wurden verschiedene Simulationsumgebungen miteinander verknüpft sowie mit den bestehenden Datenbasen synchronisiert. So wurde es möglich, die derzeit bestehenden Produktentwicklungsprozesse, um neue Konzepte zu erweitern, zu verbessern und zu beschleunigen. Insgesamt kann mit Hilfe eines agilen, datenbasierten Ansatzes mittel- bis langfristig die Gesamtwirtschaftlichkeit des Anlagenportfolios gesteigert werden.

Zusammenfassend ergeben sich die übergeordneten Ziele:

- Steigerung der Effektivität durch eine zielgerichtete Adressierung von Anforderungen im Sinne des agilen Gedankens
- Verbesserung der Effizienz durch die Vermeidung von ungewollten Iterationen aufgrund einer verbesserten Datentransparenz
- Datenbasierte Entscheidungsunterstützung durch eine durchgängige Daten- und Systemarchitektur

- Verkürzung der Time-to-Market bei gleichzeitiger Steigerung des Market-Fit

Diese übergeordneten Teilziele lassen sich in spezifische wissenschaftliche und technische Arbeitsziele untergliedern:

- Erarbeitung einer Methodik zum Aufbau eines Datenmodells als Grundlage für die datenbasierte Entscheidungsunterstützung
- Erarbeitung einer Methodik zur Definition von agilen und plange-triebenen Prozessbausteinen sowie unternehmensspezifischen agilen Rollen
- Erarbeitung einer Methodik zur kontextbezogenen Konfiguration von agilen und plangetriebenen Prozessbausteinen zur Steigerung von Effizienz und Effektivität
- Entwicklung eines Leitfadens zur Implementierung eines agilen, datenbasierten Entwicklungsprozesses im Anlagenbau mit Fokus auf die Windenergiebranche

1.4 Vorstellung des Forschungsprojekts „Agile_Data_Dev“

Die Erkenntnisse des Leitfadens wurden im Rahmen eines gemeinsamen Forschungsprojekts des Werkzeugmaschinenlabors WZL der RWTH Aachen University und eines Unternehmens aus der Windenergiebranche erarbeitet. Das Unternehmen agiert als Auftragsentwickler für Windenergieanlagen sowohl in Deutschland als auch im internationalen Umfeld. Der Lehrstuhl für Produktionssystematik des Werkzeugmaschinenlabors WZL der RWTH Aachen University befasst sich unter der Leitung von Prof. Dr.-Ing. Dipl.-Wirt. Ing. Günther Schuh seit vielen Jahren in verschiedenartigen Forschungs- und Industrieprojekten mit Fragestellungen rund um Themen des Innovationsmanagements. Dieses kann wiederum in die Bereiche des Komplexitätsmanagements, des Entwicklungsmanagements, des Data Intelligence Managements und des Prototypings aufgegliedert werden. Die Gestaltung und Optimierung von agilen, datenbasierten Entwicklungsprozessen hat sich neben der Baukastengestaltung als eine der Kernkompetenzen des Lehrstuhls in zahlreichen Forschungs- und Industrieprojekten erwiesen. Un-

terstützt wurde das Werkzeugmaschinenlabor durch das Institut für Maschinenelemente und Systementwicklung MSE und den Lehrstuhl für Software Engineering SE der RWTH Aachen University.

Im Rahmen des Projekts wurde ein systematisches Vorgehen erarbeitet, mit dem Unternehmen der Windenergiebranche einen agilen, datenbasierten Entwicklungsprozess entwickeln, implementieren und umsetzen können. Der vorliegende Praxisleitfaden unterstützt dabei die Einführung eines solchen Entwicklungsprozesses, indem das Vorgehen detailliert erläutert wird.

Dabei gliedert sich das Forschungsprojekt in die vier determinierenden Projektphasen Analyse, Konzeption, Detaillierung und Validierung. Die Arbeiten wurden dabei in je vier Arbeitspakete für den Praxispartner aus der Windenergiebranche und die Forschungsinstitute aufgeteilt:

- AP1 Analysephase Praxisunternehmen
- AP2 Analysephase WZL
- AP3 Konzeptionsphase Praxisunternehmen
- AP4 Konzeptionsphase WZL
- AP5 Detaillierungsphase Praxisunternehmen
- AP6 Detaillierungsphase WZL
- AP7 Validierungsphase Praxisunternehmen
- AP8 Validierungsphase WZL

Die vier betrachteten Projektphasen bauen inhaltlich aufeinander auf, wobei gewisse Vorarbeiten der einzelnen Projektphasen bereits vor Abschluss der vorhergehenden Projektphasen begonnen werden konnten. Einzige Ausnahme bilden die Projektphasen 1 und 2, die weitgehend parallel durchgeführt wurden. In diesen beiden Projektphasen wurden die konzeptionellen Grundlagen für den agilen, datenbasierten Entwicklungsprozess gelegt, bevor diese dann in Phase 3 in einen konfigurierbaren agilen Prozess übersetzt werden konnten. In Projektphase 4 erfolgte schließlich die ganzheitliche Erarbeitung des vorliegenden Leitfadens zur Implementierung der agilen, datenbasierten Produktentwicklung. In diesem Zusammenhang wurden Teilelemente der Methodik darüber hinaus in vorwettbewerblichen

Projekten validiert. Im Folgenden findet sich eine kurze Darstellung der Arbeitsziele der jeweiligen Projektphase:

Arbeitsziele der Phase 1: Analyse

- Erarbeitung einer Methodik zur prozesseitigen und systemseitigen Modellierung von Entwicklungsprozessen
- Beschreibung von Rollen innerhalb der Produktentwicklung
- Aufnahme von Schwachstellen in Entwicklungsprozessen als Basis für die Integration von agilen Methodenbausteinen
- Ermittlung von Anforderungen eines durchgängigen Datenmodells zur Schaffung von Transparenz als Basis einer datenbasierten Produktentwicklung

Arbeitsziele der Phase 2: Konzeption

- Definition von agilen und plangetriebenen Prozessbausteinen als Basis für die Agilisierung von Entwicklungsprojekten unter Berücksichtigung von benötigten Informationen
- Erarbeitung von Rollen zur Umsetzung der agilen und plangetriebenen Prozessbausteine
- Konzipierung eines Informationsmodells basierend auf den Anforderungen der agilen Prozessbausteine sowie der zugehörigen Rollen

Arbeitsziele der Phase 3: Detaillierung

- Erarbeitung einer Methodik zur Konfiguration eines agilen, datenbasierten Entwicklungsprozesses bestehend aus agilen und plangetriebenen Prozessbausteinen
- Auswahl von Produktumfängen, welche agil oder plangetrieben zu entwickeln sind
- Synchronisation von agilen und plangetriebenen Entwicklungssträngen zur Maximierung der Effizienz und Effektivität

- Erarbeitung von Visualisierungsmöglichkeiten als Basis der Entscheidungsunterstützung innerhalb eines Entwicklungsprojektes
- Gestaltung eines Datenmodells als Basis für die datenbasierte Produktentwicklung

Arbeitsziele der Phase 4: Validierung

- Entwicklung einer Veränderungskommunikation anhand unternehmensspezifischer Anforderungen
- Anwendung der Entwicklungsmethodik in einem vorwettbewerblichen Kontext zur Validierung der erarbeiteten Methodenelemente

Nachfolgend sind in Abbildung 2 die Arbeitsphasen in ihrem Ablauf und mit ihren Inhalten kurz dargestellt.

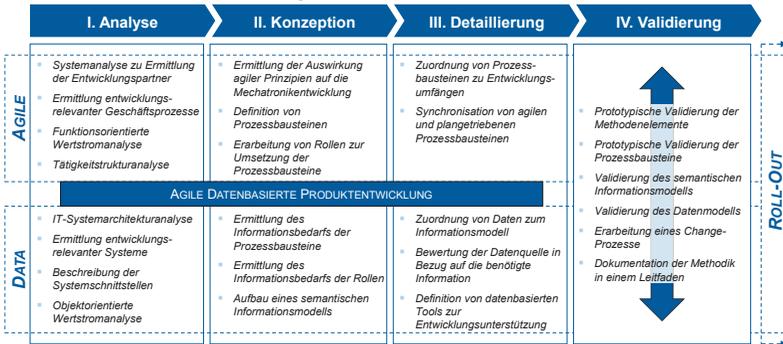


Abbildung 2: Arbeitsplan für das Forschungsvorhaben „Agile_Data_Dev“

Analysephase

Innerhalb der ersten Phase wurde zusammen mit dem Praxispartner ein Vorgehen zur Analyse des aktuellen Status der Entwicklungsprozesse systematisch erarbeitet, um die Eignung dieser Prozesse für agile Entwicklungsmethoden zu ermitteln. Hierzu wurden das Entwicklungsnetzwerk, Partner sowie Schnittstellen systematisch beschrieben. Des Weiteren wurde der entwicklungsbezogene Wertstrom über alle Wertschöpfungsstufen hinweg aufgenommen, um so ein detailliertes Bild der Prozesslandschaft zu erhalten. Zusätzlich lieferten die Beleuchtung und Analyse von Schnittstellen eine wesentliche Grundkenntnis für Ansatzpunkte der kurzzyklischen und iterationsbasierte Entwicklung. Ferner lieferte diese Phase Erkenntnisse bezüglich der Rollen und Rechte der

Akteure über alle Prozesse hinweg und ermöglichte so ein Verständnis für die Nutzung von Ressourcen.

Die Analyse hat sich zudem über die IT-Systeminfrastruktur erstreckt, um insbesondere die Datennutzung innerhalb der Produktentwicklung transparent zu machen. Dazu gehört die Ermittlung der für die Entwicklung relevanten Systeme sowie die Beschreibung der zwischen den Systemen bestehenden Schnittstellen. Zudem wurden anhand einer objektorientierten Wertstromanalyse die Wege von Entwicklungsartefakten durch den Entwicklungsprozess, dessen Aktivitäten und die dazugehörigen Systeme geprüft, welche Informationsbedarf bei den an der Entwicklung beteiligten Akteuren entstehen und in welche Objekte die benötigten Informationen zu welchem Zeitpunkt einfließen. Dementsprechend schließt die Analyse der Informationssysteme auch die Ermittlung der im Prozess vorhandenen systemseitigen Rollen und deren Rechte ein.

Konzeptionsphase

Auf Basis der in der Analysephase herausgearbeiteten Prozesse, Aufgaben und Rollen wurden in der zweiten Phase die Grundlagen für den agilen datenbasierten Entwicklungsprozess auf Prozess- und Informationssystemebene definiert.

Für die konzeptionelle Definition eines Entwicklungsprozesses, in dem Teilsysteme einer Windenergieanlage datenbasiert und agil entwickelt werden können, wurde die Auswahl eines sowohl prozess- als auch produktabhängigen agilen Konzepts vorbereitet. Dazu wurden auf Basis der Wertstrom- und Entwicklungsprozessanalyse des vorangegangenen Arbeitspakets diejenigen agilen Konzepte ausgewählt, die sich zur Adaption innerhalb der beschriebenen Unternehmensprozesse eignen. Die agilen Wirkmechanismen, die bereits in der Softwarebranche Anwendung finden, wurden hierbei auf eine Übertragbarkeit in den Anlagenbau überprüft und für eine Implementierung in die bestehende Prozesslandschaft konzeptioniert. In diesem Zusammenhang wurden standardisierte Prozessbausteine beschrieben, welche eine Kombination der agilen und der plangetriebenen Entwicklung ermöglichten und sich zu einem ganzheitlichen, agilen Entwicklungsprozess konfigurieren lassen.

Zusätzlich wurde in dieser Phase die systemseitige Unterstützung der zuvor definierten agilen und plangetriebenen Prozessbausteine konzipiert. Für einen echtzeitfähigen Wechsel zwischen unterschiedlichen Sichten auf das Produkt und die dahinterliegenden Strukturen ist die Konzeption generischer, produktbeschreibender Perspektiven notwendig. Weiterhin wurden die Informationsbedarfe der definierten agilen Rollen identifiziert, um abschließend ein ganzheitliches Informationsmodell für einen agilen datenbasierten Entwicklungsprozess ableiten zu können. Aufbauend ließ sich ein entsprechendes Sichtenmanagement konstruieren, um funktionsübergreifenden Teilnehmern eines agilen Teams während des Entwicklungszyklus die entsprechenden Sichtweisen auf das zu entwickelnde Inkrement (zu entwickelnder Teilumfang des Produktes innerhalb der iterativen Entwicklungszyklen) zu ermöglichen.

Detaillierungsphase

Innerhalb der Detaillierungsphase wurden die bisher konzeptionell ausgearbeiteten Prozessbausteine für die Umsetzung in einem ganzheitlichen agilen Entwicklungsprozess spezifiziert. Schlussfolgernd wurde ein Vorgehen für die Auswahl der definierten Prozessbausteine anhand der zu entwickelnden Produktumfänge definiert. Anhand von Agilitätsindikatoren erfolgte eine methodisch unterstützte Bewertung der einzelnen Produktumfänge zur fundierten Auswahl von agilen oder plangetriebenen Entwicklungsansätzen je Produktumfang. Zusätzlich erfolgte die Bestimmung des horizontalen und vertikalen Synchronisationsbedarfs zwischen den einzelnen Prozessphasen.

Gleichzeitig wurde die Umsetzbarkeit der Datenunterstützung sichergestellt. Hierzu erfolgte die Detaillierung der notwendigen Sichten und Informationsbedarfe durch die Identifikation der Daten zur Unterstützung des Entwicklungsprozesses. So wurde ein Vorgehen entwickelt, um den im vorangegangenen Arbeitspaket identifizierten Informationsbedarfen die relevanten Datenbedarfe zuzuordnen und diese in ein Datenmodell für einen ganzheitlichen agilen Entwicklungsprozess zu überführen. Verbunden damit wurde die Entwicklung eines Rollen- und Rechtekonzeptes für den agilen, datenbasierten Entwicklungsprozess umgesetzt. Abschließend erfolgte die Festlegung entsprechender Anforderungen an eine datenbasierte Visualisierungsunterstützung, zum einen zum Fortschrittscontrolling im Entwicklungsvorgehen und zum anderen zur

Entscheidungsunterstützung bei der Analyse unterschiedlicher entwicklungsrelevanter Fragestellungen.

Validierungsphase

Die Validierung des erarbeiteten agilen, datenbasierten Entwicklungsprozesses erfolgte anhand von Entwicklungsumfängen mit vorwettbewerblichem Charakter und ermöglichte die Bewertung des zu erwartenden Nutzens, die Verifikation der beschriebenen Methodik sowie die Überprüfung der Umsetzbarkeit der Datenunterstützung basierend auf bestehenden Systemen. Begleitet wurde dieser Validierungsprozess durch die Entwicklung und Umsetzung eines entsprechenden Change-Managementprozesses. Als Ergebnis dieser abschließenden Validierungsphase ist der vorliegende Leitfaden für die Windenergiebranche hervorzuheben, als Instrument, das weitere Unternehmen befähigt, agile, datenbasierte Entwicklungsprozesse anzuwenden.

2 Konzeption einer agilen, datenbasierten Produktentwicklung

2.1 Erfolgsfaktoren der agilen, datenbasierten Produktentwicklung

Unternehmen befinden sich derzeit in einem Spannungsfeld von zunehmendem Kostendruck und steigenden Kundenanforderungen. Dieser Wettbewerbsdruck fordert von Unternehmen eine Verschlanung der Prozesse und die Steigerung der internen Effizienz. Darüber hinaus führen die individuellen Kundenwünsche und erhöhte Ansprüche an die technische Leistungsfähigkeit der Produkte zu einem Wachstum der angebotenen Variantenvielfalt sowie der daraus resultierenden Produktkomplexität. Um die interne, produktinduzierte Komplexität wirtschaftlich zu realisieren und gleichzeitig den externen Anforderungen des Marktes gerecht zu werden, bedarf es demzufolge eines geeigneten Managementansatzes.

Das Konzept einer agilen, datenbasierten Produktentwicklung im Maschinen- und Anlagenbau ermöglicht Unternehmen ein Vorgehensprinzip zu etablieren, das auf aktuelle Entwicklungsaktivitäten, an sinnvollen Stellen, agile Prinzipien systematisch überträgt. Zusätzlich befähigt die datenbasierte Arbeit in der Produktentwicklung einen schnelleren, transparenten Austausch von Information sowie eine Reduktion der Fehler bei Informationsweitergaben

Die drei zentralen Erfolgsfaktoren einer agilen datenbasierten Produktentwicklung liegen in der Abbildbarkeit von wiederkehrenden Prozessmodulen, der Zuordnung dieser Module zu physikalischen Modulen und in einem auf den Prozess adaptierten Informations- und Datenmodell. Im Folgenden werden die zentralen Erfolgsfaktoren für eine agile Produktentwicklung beleuchtet, gefolgt von einer Analyse der Erfolgsfaktoren für eine datenbasierte Produktentwicklung. Hierbei wird aufgezeigt, welche Aspekte den Entwicklungsprozess positiv beeinflussen. Abschließend wird, innerhalb eines Zwischenfazit, herausgearbeitet, welche Synergiepotenziale sich innerhalb der beiden Themengebiete ergeben.

2.1.1 Erfolgsfaktoren für eine agile Produktentwicklung

Die Transformation zur agilen Produktentwicklung stellt Unternehmen in der Praxis, trotz der dargestellten existierenden Ansätze, häufig vor große Herausforderungen.

Spezielle Herausforderungen, mit denen sich Unternehmen insbesondere im Bereich der Windenergiebranche zu Beginn der Integration agiler Prinzipien in die Produktentwicklung konfrontiert sehen, lassen sich in drei übergeordnete Bereiche einteilen: Prozess, Organisation & Infrastruktur und Kultur³⁶. Aus prozessualer Sicht gilt es, agile Ansätze zu adaptieren und eine prozessuale Anpassung der Entwicklungsorganisation zu forcieren. Auf dem Bereich Organisation & Infrastruktur ist v.a. die Zusammenarbeit in interdisziplinären Teams und die Selbstbestimmtheit dieser Teams sicherzustellen. Die Anpassung der Prozesse und Rollen werden in Unternehmen im Kontext der agilen Transformation mit verschiedenen kulturellen Herausforderungen verbunden. Es gilt ein Bewusstsein für agile Prinzipien zu erarbeiten und dieses Mindset nachhaltig zu verankern.

Ein Blick in die gegenwärtige Situation der Produktentwicklung in der Windenergiebranche und die Untersuchung der eingesetzten Methoden offenbart, eine notwendige Voraussetzung für die Umsetzung der vorwiegend genutzten klassischen plangetriebenen Entwicklung ist die vollständige Anforderungsdefinition und Prozessstrukturierung zu Projektbeginn. Damit ermöglicht ein klassisches Vorgehen zwar einen sicheren Umgang mit der typischerweise in der mechatronischen Produktentwicklung hohen Vielfalt an Systemelementen, gleichzeitig wird aber die Veränderungsdynamik des Entwicklungsprozesses eingeschränkt. Unter Berücksichtigung der heutigen Markt- und Wettbewerbsbedingungen stoßen plangetriebene Ansätze in der Windenergiebranche daher vermehrt an ihre Grenzen. Dabei bedeutet die Einführung agiler Ansätze nicht, dass gänzlich auf die Vorteile einer plangetriebenen Entwicklung verzichtet werden muss. Während in der Vergangenheit die beiden gegensätzlichen Ansätze als miteinander unvereinbar eingestuft wurden, rückt dieser empfundene Widerspruch heute mehr und mehr in den Hintergrund. Um Verluste im Hinblick auf die Prozesseffektivität und -effizienz zu vermeiden, fokussieren produzierende Unternehmen stattdessen die gezielte Verknüpfung plangetriebener

³⁶ Riesener et al. 2020.

und agiler Vorgehensweisen und Methoden, um die jeweiligen Stärken der Ansätze vollumfänglich ausnutzen zu können. Ziel dieser Kombination ist es, die Produktentwicklung sowohl flexibel und schnell als auch stabil und strukturiert zu gestalten. Die Adaption dieses Vorgehensprinzips in der Windenergiebranche verspricht also vor dem Hintergrund der vorherrschenden „Moving targets“ und hohen Systemkomplexität viele Vorteile.

Wie eine vorangegangene Studie des Werkzeugmaschinenlabors WZL der RWTH Aachen mit 21 beteiligten Unternehmen des Maschinen- und Anlagenbaus³⁷ gezeigt hat, lassen sich im Zusammenhang mit den zuvor aufgezeigten Herausforderungen (Prozess, Organisation & Infrastruktur und Kultur) bei der Integration agiler Prinzipien in die Produktentwicklung übergeordnete Erfolgsfaktoren ableiten, um diese Hindernisse zu überkommen. Konkret wurden innerhalb der Studie des Werkzeugmaschinenlabors WZL der RWTH Aachen die Gestaltungsfelder Kultur und Mindset, Organisation, Prozess, Methoden und Werkzeuge, Infrastruktur und Prototyping als zentrale, übergeordnete Handlungsfelder der erfolgreichen agilen Produktentwicklung zu identifizieren³⁸ (s. Abbildung 3).

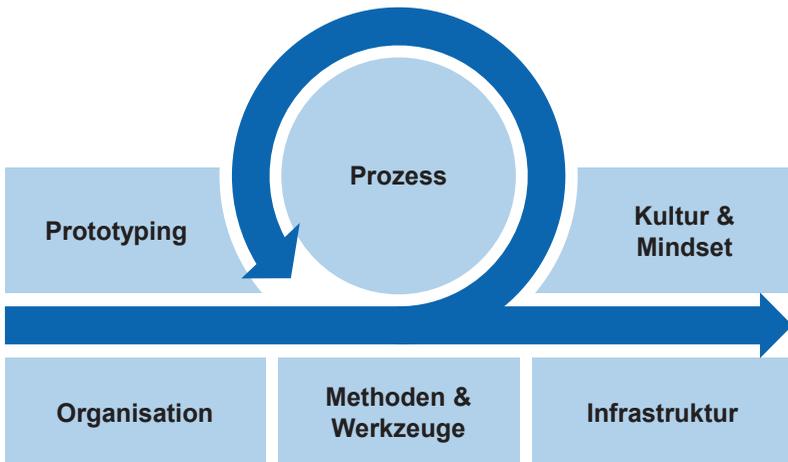


Abbildung 3: Ordnungsrahmen der Erfolgsfaktoren agiler Produktentwicklung

³⁷ Werkzeugmaschinenlabor WZL der RWTH Aachen University 2018.

³⁸ Schuh et al. 2021.

Zentrale Erfolgsfaktoren sind dabei im Handlungsfeld „Kultur und Mindset“ die Trennung der fachlichen und disziplinarischen Führung sowie die Überwindung der Vollständigkeitsparanoia. Zusätzlich gilt es, der Kommunikation zwischen den Entwicklungsbeteiligten ausreichend Bedeutung zuzumessen und diese entsprechend zu fördern.

Im Handlungsfeld „Organisation“ bilden Schulungen zu Grundlagen des agilen Arbeitens für Mitarbeiter*innen einen Schlüsselfaktor zur erfolgreichen Gestaltung einer agilen Produktentwicklung. Dabei sind ausreichend Freiräume zur Adressierung von Problemen vorzusehen, um geeignete Lösungen zu finden. Zusätzlich sind verantwortungsvolle Teams langfristig zu besetzen und cross-funktional aufzustellen.

Wesentlicher Erfolgsgarant im Handlungsfeld Prozess ist eine zielführende Kombination plangetriebener und agiler Vorgehensweisen, um eine hybride Produktentwicklung zu ermöglichen. Das richtige Maß an Agilität trägt zu einer effizienten Produktentwicklung bei.

Hinsichtlich der „Methoden und Werkzeuge“ ist es von besonderer Relevanz, Transparenz, Verbindlichkeit und eindeutige Priorisierungen sowie regelmäßige Synchronisation zu schaffen. Der Einsatz der spezifischen Methoden und Werkzeuge ist stets an den jeweiligen Zielen auszurichten.

In Sachen „Infrastruktur“ bilden die Steigerung der Team-Performance durch gegenseitiges Verständnis, ein kontinuierlicher und offener Austausch durch eine offene Raumstruktur und Transparenz zur Steigerung der Produktivität entscheidende Erfolgsfaktoren.

Ferner gilt es im Handlungsfeld „Prototyping“ spezifische Prototypen zielgerichtet einzusetzen, um schnell und proaktiv konkretes Feedback einzuholen. Prototypen bilden dabei bewusst kein vollständiges, detailliertes Produkt, sondern zunächst Teilumfänge, die sich an zu erprobenden Lösungsprinzipien orientieren. Technologien wie 3D-Druck oder die Nutzung von Simulationsumgebungen unterstützen bei der schnellen Realisierung der Prototypen. Dabei sind diese Handlungsfelder ganzheitlich und konsequent durch eine entsprechende Top-Management-Unterstützung zu forcieren.

Die größte Hausforderung für Unternehmen der Windenergiebranche ist jedoch der Übergang von einer planorientierten zur agilen Entwicklung, da es an einer umfassenden Methode fehlt, mit der sie ihre derzeitigen Entwicklungsaktivitäten in einen agilen Ansatz überführen kön-

nen. Um einen Transfer der dargestellten Erfolgsfaktoren und Handlungsfelder der agilen Produktentwicklung auf die Windenergiebranche umzusetzen, werden Prozessbausteine definiert, die es Unternehmen ermöglichen, agile Entwicklungssprints auf Basis der Aktivitäten ihres bisherigen plangetriebene Entwicklungsansatzes zu gestalten³⁹. Als Bausteine dienen generische Prozessmodule, die prozessuale Transformationen von spezifischen Inputs in spezifische Outputs darstellen⁴⁰. Die Verwendung dieser Prozessmodule für die Planung von Entwicklungssprints stellt sich als ein zentraler Erfolgsfaktor für die agile Produktentwicklung dar. Prozessbausteine versetzen Unternehmen dabei in die Lage, Prozesse zu konfigurieren, die auf die Entwicklung spezieller Produktinkremente zugeschnitten sind und die Integration agiler Ansätze ermöglichen, ohne eine umfangreiche Neugestaltung des gesamten Entwicklungsprozesses vorauszusetzen.

2.1.2 Erfolgsfaktoren für datenbasierte Produktentwicklung

Die Erzeugnisse produzierender Unternehmen des Maschinenbaus haben in den letzten Jahrzehnten einen kontinuierlichen Wandel hin zu miteinander vernetzten mechatronischen Systemen, durchlaufen⁴¹. Diese Entwicklung resultiert in stetig steigenden Elektrik- und Softwareanteilen innerhalb moderner Produkte⁴². So besaß ein Auto 2010 laut Bosch etwa zehn Millionen Zeilen Software-Code. Heutige nicht-autonom fahrende Autos hingegen enthalten bereits 100 Millionen Zeilen Software-Code⁴³.

Die zuvor aufgeführten Herausforderungen können mit herkömmlichen Methoden der Produktentwicklung nicht mehr bewältigt werden⁴⁴. Insbesondere die steigenden Wechselwirkungen der Fachdisziplinen Mechanik, Elektrik und Software erfordern Modellierungsansätze, die die transparente Abbildung dieser Abhängigkeiten über sämtliche Phasen der Produktentwicklung hinweg ermöglichen⁴⁵. Diese Anforderungen erfordern eine Transition von einer traditionellen dokumentenbasierten

³⁹ Riesener et al. 2022.

⁴⁰ Riesener et al. 2021.

⁴¹ Dumitrescu et al. 2021.

⁴² Eigner et al. 2014.

⁴³ Bosch-Presse 21.07.2020.

⁴⁴ Jacobs et al. 2022.

⁴⁵ Alur 2015; Graessler und Hentze 2020.

Produktentwicklung hin zu einer domänenübergreifenden, datenbasierten Produktentwicklung in der die Zusammenhänge und das domänen-spezifische Wissen strukturiert bereitgestellt werden. Diesen Ansatz verfolgt die funktionsorientierte, modellbasierte Produktentwicklung in der sämtliche beteiligte Fachdomänen durchgängig in Entwicklungsmodellen verknüpft sind⁴⁶. Spezifisch dem Model-Based Systems Engineering (kurz: MBSE) als Ausprägung einer solchen modellbasierten Produktentwicklung wird dabei viel Potenzial zugesprochen und ist bereits erfolgreich umgesetzt. Aufgrund der bereits existierenden Erkenntnisse zur Umsetzung wird MBSE im Folgenden als Methode der daten- und modellbasierten Produktentwicklung genutzt. Eine Gegenüberstellung des Betrachtungsrahmens zur dokumentenbasierten Produktentwicklung ist in Abbildung 4 aufgebaut. Im Folgenden werden die Grundzüge des MBSE und wesentliche Voraussetzungen für die Systemmodellierung erläutert.

Dokumentenbasierte Produktentwicklung	Model-Based Systems Engineering
<ul style="list-style-type: none"> ▪ Zeichnungen und Stücklisten im Fokus ▪ Dokumentenablage im PLM-System ▪ Sequenzieller Modellstatus ▪ Management der Entwicklungsprozesse bis SOP 	<ul style="list-style-type: none"> ▪ Modelle im Fokus (Wissensmanagement) ▪ Vernetzung domänenspezifischer Modelle (SysML) ▪ Evolutionäres Systemmodell mit Parameter-SPOT ▪ Management des gesamten Product Life Cycle

Abbildung 4: Gegenüberstellung der Eigenschaften dokumentenbasierter und modellbasierter Entwicklungsansätze⁴⁷

Das INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING (kurz: INCOSE) definiert MBSE als „formalisierte Anwendung von Modellbildung, um die Aktivitäten der Anforderungserfassung, Entwicklung, Analyse, Verifikation und Validierung eines Systems beginnend in der Konzeptphase, über die Entwicklungsphase bis zu späteren Lebenszyklusphasen zu unterstützen“.⁴⁸

Durch die Schaffung eines interdisziplinären Systemmodells werden im MBSE sämtliche Modelle und Informationen des Entwicklungsprozesses zentral als Single Point of Truth verwaltet und funktional miteinander verknüpft⁴⁹. Als Kernelement des MBSE bildet das Systemmodell damit

⁴⁶ Jacobs et al. 2022.

⁴⁷ Eigner et al. 2017.

⁴⁸ International Council on Systems Engineering 2007.

⁴⁹ Albers et al. 2015.

eine formalisierte Abbildung des Produkts⁵⁰ und beschreibt domänenübergreifend dessen funktionales Verhalten.

Die transparente Abbildung des ganzheitlichen Zusammenwirkens technischer Systeme stellt einen Erfolgsfaktor bei Windenergieanlagenentwicklung dar⁵¹. Aufgrund der steigenden Aufmerksamkeit für das Thema ist zudem zu erwarten, dass sich die Leistungsfähigkeit des MBSE stetig erhöht. Insbesondere muss in den kommenden Jahren der Paradigmenwechsel von der dokumentenzentrierten hin zur modellbasierten Produktentwicklung durch MBSE in der industriellen Praxis etabliert werden⁵².

In der Konsortialbenchmarking Studie Model Based Systems Engineering des Werkzeugmaschinenlabors WZL der RWTH Aachen⁵³ und des Center for Systems Engineering wurden zehn Erfolgsfaktoren für die Einführung sowie Anwendung einer MBSE-basierten Produktentwicklung identifiziert. Diese verteilen sich entsprechend der Abbildung 5 auf die drei Themenfelder organisatorische Voraussetzungen, durchgängige Systemarchitektur und technologische Voraussetzungen:

Organisatorische Voraussetzungen	Kulturwandel fördern u. fordern	Mehrwert aufzeigen	Durchhaltevermögen zeigen	ExpertInnen implementieren	Ausbildungen strukturieren
Durchgängige Systemarchitektur	Strukturen schaffen	Methodik priorisieren	Standards schaffen		
Technische Voraussetzungen	MBSE-Tool nutzen	Toolunabhängigkeit u. -fokussierung gewährleisten			

Abbildung 5: Erfolgsfaktoren für einen datenbasierten Produktentwicklungsprozess

Nachfolgend sollen mit Hinblick auf den Paradigmenwechsel von dokumenten- zu datenbasierter Produktentwicklung die besonders relevanten Themenfelder „Durchgängige Systemarchitektur“ und „Technische Voraussetzungen“ näher beschrieben werden.

Identisch zu den Erfolgsfaktoren der agilen Produktentwicklung bildet die organisatorische Weiterentwicklung die Basis für eine erfolgreiche Umsetzung einer datenbasierten Produktentwicklung. Dabei gilt es

⁵⁰ Konrad et al. 2019; Albers et al. 2015.

⁵¹ Jacobs et al. 2022.

⁵² Dumitrescu et al. 2021.

ebenso durch eine geeignete Kommunikation Mehrwerte und ein Zielbild den Mitarbeiter*innen aufzuzeigen, um im Sinne einer übergeordneten Vision langfristig den Kulturwandel anzustoßen und festgelaufene Abläufe aufzubrechen. Hierzu müssen allerdings mittel- und langfristig entsprechende Stellen geschaffen und Ausbildungen strukturiert oder angepasst werden.

Aufbauend auf der Komponente Mensch ist modellseitig eine durchgängige Systemarchitektur für die Beherrschung fachdisziplinübergreifender Abhängigkeiten unabdingbar und stellt die Basis des MBSE dar. MBSE umfasst dabei die Definition von Anforderungen, die Konzeptentwicklung sowie das Validieren und Testen des zu entwickelnden Systems. Um die Durchgängigkeit erzeugen zu können, wird eine standardisierte Sprache benötigt, in der sämtliche Fragmente miteinander kommunizieren können. Daher wird im Kontext von MBSE zum größten Teil mit der Modellierungssprache SysML gearbeitet und als Standard verwendet.

Die grafische **Modellierungssprache** Systems Modeling Language (kurz: SysML) wurde für die Beschreibung technischer Systeme entwickelt, um einen standardisierten Rahmen für die Entwicklung einer Systemarchitektur zu bilden. SysML basiert auf der in der Softwareentwicklung eingesetzten Unified Modeling Language (kurz: UML) und enthält spezifische Erweiterungen für die Systementwicklung⁵⁴. Dies ermöglicht die gemeinsame Entwicklung mechatronischer Systeme durch System- und Softwareingenieure⁵⁵. SysML bietet somit eine standardisierte Notation zur Darstellung von Systemzusammenhängen, die von allen am Entwicklungsprozess beteiligten Parteien interpretiert werden kann⁵⁶. SysML legt somit die Basis zur Erstellung eines einheitlichen, logischen Modells.⁵⁷ Als Nachfolger von SysML gibt es seit 2023 SysML v2, welche die Konzepte von SysML verfeinern und seine Notationssyntax verbessern soll. Die verbesserte Syntax bietet den Vorteil, dass sie sowohl grafisch als auch textuell notiert werden kann und damit neue und bessere Modellierungshilfen ermöglicht. Zu Beginn des Forschungsprojekts stand diese Version jedoch noch nicht zu Verfügung.

⁵⁴ Object Management Group 2019.

⁵⁵ Object Management Group 2019.

⁵⁶ Bursać 2016.

⁵⁷ Object Management Group 2019, Für eine ausführliche Beschreibung sei an dieser Stelle auf die offizielle Definition der SysML gemäß OMG Systems Modeling Language verwiesen.

Aufbauend wird ein Vorgehen benötigt, das die Systemmodellierung an sich strukturiert. Die Definition einer solchen **Modellierungsmethode** zur Modellierung eines konsistenten MBSE-Systemmodells gilt dabei als Schlüsselfaktor. Als Methode für die Ausgestaltung der einzelnen Produktentwicklungsschritte und des Architektur-Rahmenwerks haben sich beispielsweise Modellierungsmethoden, wie FAS⁵⁸, SYSMOD⁵⁹, SPES⁶⁰, Consens⁶¹, FAS4M, MecPro² sowie die am Institut für Maschinenelemente und Systementwicklung der RWTH Aachen University entwickelte motego-Methode (WiGeP) etabliert⁶². Ein Alleinstellungsmerkmal der motego-Methode stellt die durchgängige, formalisierte Beschreibung von Produkten bis hin zur Abbildung elementarer physikalischer Zusammenhänge dar⁶³. Diese Detailtiefe ermöglicht die parameterbasierte Anbindung von Entwicklungsmodellen unterschiedlicher fachlicher Domänen an das Systemmodell. Diese Anbindung wiederum ermöglicht das frühzeitige und kontinuierliche Testen von Anforderungen und Produktfunktionen⁶⁴.

Die motego-Methode umfasst die vier Ebenen Requirements, Functions, Solutions und Product sowie die durchgängige parametrische Verknüpfung der Ebenen (siehe Abbildung 6)⁶⁵. Zunächst werden Anforderungen an das zu entwickelnde Produkt definiert. Die Anforderungen werden anschließend mit einer funktionalen Dekomposition des Produktes verknüpft, die das Verhalten des Produktes lösungsneutral beschreibt. Solutions konkretisieren, wie das funktionale Verhalten physikalisch umgesetzt wird. Als zentrale Elemente des Ansatzes, sind die Solutions mit den Entwicklungsmodellen und Workflows der relevanten fachlichen Domänen verknüpft und ermöglichen so das funktionale Testen in frühen Phasen des Entwicklungsprozesses. In mechanischen Systemen werden Funktionen durch den Kontakt von mindestens einer Komponente mit einer weiteren realisiert. Die Wirkflächen an denen es

⁵⁸ Lamm und Weilkiens 2014.

⁵⁹ Weilkiens 2014.

⁶⁰ Pohl et al. 2016.

⁶¹ Gausemeier et al. 2012.

⁶² Jacobs et al. 2022; Zerwas et al. 2021.

⁶³ Jacobs et al. 2022.

⁶⁴ Zerwas et al. 2021; Höpfner et al. 2021.

⁶⁵ Jacobs et al. 2022.

zum Kontakt kommt, sind innerhalb der Solutions definiert. Aus den separaten Wirkflächen wiederum entstehen die physischen Produktkomponenten, die auf der Produkt-Ebene definiert werden.⁶⁶

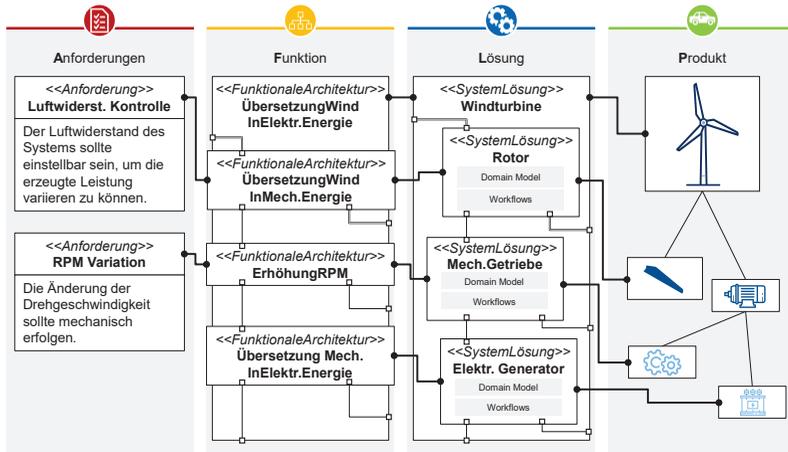


Abbildung 6: Auszug der funktionsorientierten Modellierung einer Windenergieanlage nach JACOBS ET AL.⁶⁷

Neben den Aufbau geeigneter Prozesse und Rahmenbedingungen in der Organisation gilt es ebenso die technische Seite zu beachten. Dazu wird empfohlen auf die Unterstützung eines Modellierungswerkzeugs (engl.: modeller) für den Aufbau eines Systemmodells bei der Anwendung in SysML mithilfe der definierten MBSE-Methode zurückzugreifen. MBSE-Modeller stellen Software-Tools dar, welche die Handhabung der Sprachelemente erlauben und damit den Aufbau von Systemmodellen ermöglichen⁶⁸. Abbildung 7 zeigt im Folgenden einen Auszug am Markt verfügbarer MBSE-Tools in denen Modellierungswerkzeuge integriert sind und führt deren wesentliche Eigenschaften auf.

⁶⁶ Jacobs et al. 2022.

⁶⁷ Jacobs et al. 2022; Axair Fans UK Limited 2020.

⁶⁸ Munker 2016; oose Innovative Informatik eG.

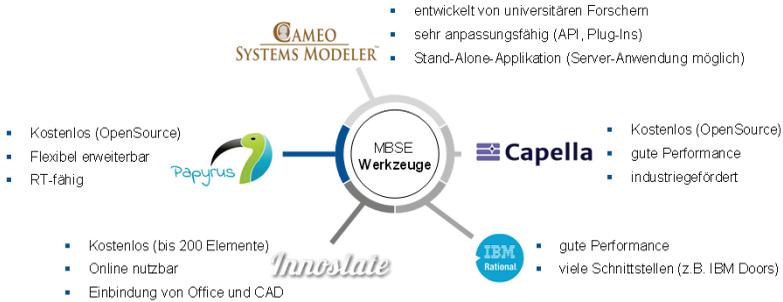


Abbildung 7: Beispiele für MBSE-Modellierungswerkzeugen

2.1.3 Zwischenfazit

Hersteller von Windenergieanlagen sehen sich derzeit mit einer Vielzahl von Herausforderungen konfrontiert. Dynamische und volatile Entwicklungsumgebungen induzieren einen bedeutenden Komplexitätsanstieg. Insbesondere sich schnell ändernde Anforderungen, also das Risiko der „Moving-Targets“ und inhomogene globale Märkte sind für diesen Anstieg der Unsicherheit verantwortlich. Kürzere Produkt- und Technologielebenszyklen verstärken diese Effekte. Zusätzlich ist eine Windenergieanlage, als ein komplexes technisches System mit voneinander abhängigen Komponenten, im Entwicklungsprozess darauf angewiesen, dass Informationsflüsse zwischen Entwicklungsabteilungen konsistent und auf einer zuverlässigen Datengrundlage basieren, um Informationsbrüchen oder -defiziten vorzubeugen.

Die Integration agiler Prinzipien und datenbasierter Ansätze birgt erhebliches Potenzial zur Verbesserung der Effektivität und Effizienz von Entwicklungsprozessen. Es ist zu prüfen, wie diese Prinzipien in die Windenergiebranche eingebunden werden können, um den dargestellten Herausforderungen zu begegnen und die Branche zu befähigen den Ausbau der Windenergie voranzutreiben. Konkret sind Erfolgsfaktoren und eine Herangehensweise für die Einführung eines agilen datenbasierten Entwicklungsprozesses in der Windenergiebranche zu erarbeiten.

2.2 Grobkonzept eines Vorgehensmodells für die agile, datenbasierte Produktentwicklung

Im Folgenden wird ein Grobkonzept für ein Vorgehensmodell für die agile, datenbasierte Produktentwicklung erörtert. Dieses Modell kombiniert agile Prinzipien mit einem datenbasierten Entwicklungsprinzip. In den folgenden Abschnitten werden die zentralen Phasen dieses Vorgehensmodells - Analyse, Konzeption, Detaillierung und Implementierung - betrachtet und dabei spezifische Aspekte der agilen, datenbasierten Produktentwicklung herausgearbeitet.

2.2.1 Vorgehen in der Analysephase

Ziel der Analysephase ist die Erarbeitung einer Methodik zur Modellierung und Analyse von Entwicklungsnetzwerken, Geschäfts- und Entwicklungsprozessen. Dazu wird zunächst die Identifizierung von Agilisierungspotenzialen in Entwicklungsnetzwerken fokussiert. Diese Identifizierung basiert auf einer umfassenden Systemanalyse des Entwicklungsnetzwerkes, einer Differenzierung von Kooperationsformen und der Klassifikation von Systempartnern und Schnittstellen. Anschließend ist eine Geschäftsprozessanalyse vorzunehmen. Diese Analyse ist anhand der übergeordneten Bestandteile der Identifikation entwicklungsrelevanter Geschäftsprozesse und einer korrespondierenden Geschäftsprozessschnittstellenbeschreibung zu strukturieren.

Zudem wird in der Analysephase die im Unternehmen bestehende Informationssystemarchitektur untersucht. Grundlage für ein effektives Informations- und Datenmodell ist Kenntnis über die im Rahmen der Produktentwicklung auszutauschenden Datentypen und deren Struktur. Aus diesem Grund widmet sich die Analysephase insbesondere der strukturierten Aufarbeitung der im Entwicklungsprozess vorliegenden Datenstrukturen.

2.2.2 Vorgehen in der Konzeptionsphase

Innerhalb der Konzeptionsphase wird die agile Entwicklung auf übergeordneter Ebene strukturiert. Dazu wird der Entwicklungsprozess zunächst dekomponiert, um Prozessabschnitte gemäß ihres zuvor identifizierten Agilisierungspotenzials entweder plangetrieben oder agil

durchzuführen. Sodann wird das Vorgehen insbesondere in den Phasen „Planung“, „Entwicklung“ und „Validierung“ der agilen Entwicklung beschrieben. Durch die Definition von Prozessbausteinen werden Grundelemente für den zeitlich effizienten Aufbau individueller Entwicklungssprints bereitgestellt. Ein Prozessrahmenmodell strukturiert die Steuerung und Organisation des Entwicklungsprozesses und bildet die entsprechenden Aktivitäten ab.

Parallel dazu wird basierend auf den zuvor analysierten Datenstrukturen ein Informationsmodell erstellt. Dazu wird die Produktstruktur einer Windenergieanlage als Strukturdiagramm in der SysML modelliert. Dieses Modell umfasst die Abhängigkeiten der Systemelemente untereinander und bildet die Grundlage für den Zugriff auf ausgewählte Systeminformationen.

2.2.3 Vorgehen in der Detaillierungsphase

Die konkrete Durchführung der agilen datenbasierten Entwicklung steht im Fokus der Detaillierungsphase. Dabei wird zunächst aufgezeigt, wie Fragestellungen für die agile Entwicklung ausgehend von unvollständigen Anforderungen abgeleitet werden. Mithilfe von Prozessbausteinen werden geeignete Entwicklungssprints zur Beantwortung der Fragestellungen konfiguriert. Da ein Entwicklungsprozess mehrere parallele Entwicklungszyklen umfasst, innerhalb derer zwischen plangetriebenen und agilen Vorgehensweisen situativ gewechselt werden kann, ist eine vertikale wie horizontale Synchronisation des Vorgehens nötig. Entsprechende Koordinations- und Wechsellpunkte werden beschrieben.

Die Ableitung des Datenmodells erfolgt über einen Datenparser basierend auf dem Informationsmodell. Das Datenmodell wiederum stellt die Grundlage für den Aufbau der Datenbank dar. Diese ist über ein Webinterface zugänglich und bildet die „Single.Source-of-Truth für agile, datenbasierte Produktentwicklung.

2.2.4 Vorgehen in der Implementierungsphase

Die Implementierungsphase widmet sich der langfristigen und nachhaltigen Verankerung der agilen, datenbasierten Produktentwicklung im Unternehmen. Diese Verankerung wird durch zwei zentrale Maßnahmen erreicht, der organisatorischen Verankerung und Ausrichtung der

Unternehmensstrukturen sowie einem systematischen Change-Prozess.

Die organisatorische Verankerung adressiert hierbei drei Handlungsfelder. Zunächst werden für die agile Produktentwicklung im Kontext der Windenergiebranche geeignete Rollenprofile definiert, die die Durchführung des definierten agilen Entwicklungsprozesses ermöglichen und Aufgaben, Kompetenzen und Verantwortlichkeiten verteilen. Für die operative Produktentwicklung wird zudem eine agile Projektorganisation definiert, die den agilen Entwicklungsprozess in den Gesamtkontext des Unternehmens einbettet. Abschließend wird dargestellt, wie für Unternehmen die Struktur der Aufbauorganisation gestaltet werden kann, um eine agile Produktentwicklung zu ermöglichen.

Der Change-Prozess wiederum begleitet die Einführung der agilen Produktentwicklung und soll einen kulturellen Wandel in plangetriebenen ausgerichteten Unternehmen initiieren. So wird auf der einen Seite ein Beispielprozess mit Prozessphasen und möglichen Aktivitäten vorgestellt, der genutzt werden kann, um eine agile Produktentwicklung im eigenen Unternehmen einzuführen. Zudem wird beschrieben, wie der kulturelle Wandel bei der Einführung der agilen Produktentwicklung ausgestaltet sein muss, um agile Entwicklungsprinzipien in der täglichen Arbeitsroutine verankern zu können.

3 Vorgehen zur Umsetzung einer agilen, datenbasierten Produktentwicklung

Das angeführte Grobkonzept des Vorgehens in der Analyse-, Konzeptions- und Detaillierungs- und Implementierungsphase wird nachfolgend hinsichtlich des methodischen Vorgehens zur Umsetzung eines agilen, datenbasierten Produktentwicklungsprozesses detailliert. Hierzu werden die im Rahmen des Forschungsvorhabens entwickelten Vorgehensweisen und Methoden vorgestellt.

3.1 Analysephase

Die Analysephase dient der Aufnahme eines Status-quo der Produktentwicklung eines Unternehmens aus der Windenergiebranche. Diese Analyse setzt sich zusammen aus der Untersuchung der Produktentwicklungsprozesse in 3.1.1 hinsichtlich der Identifizierung von Potenzialen zur Einführung agiler Wirkprinzipien sowie einer Analyse der Informationssystemarchitektur in Kapitel 3.1.2 hinsichtlich der Einführung eines Informationsmodells zur gegenseitigen Bereitstellung von Informationen im Kontext des Entwicklungsprozesses.

3.1.1 Analyse der Produktentwicklungsprozesse

Den initialen Schritt im Kontext der Gestaltung eines agilen, datenbasierten Entwicklungsprozesses innerhalb der Windenergiebranche bildet die Identifikation von Agilisierungspotenzialen im bestehenden Entwicklungsnetzwerk und der Prozesslandschaft des Unternehmens, das eine agile, datenbasierte Entwicklung umsetzen möchte. Im Folgenden wird in Kapitel 3.1.1.1 das Vorgehen zur Systemanalyse des Entwicklungsnetzwerkes in Anlehnung an ALPAR et al. beschrieben und eine Morphologie zur Klassifikation bereitgestellt⁶⁹. Aufbauend auf dieser Systemanalyse wird in Kapitel 3.1.1.2 ein Vorgehen zur Geschäftsprozessanalyse mit Fokus auf die Produktentwicklung in der Windenergiebranche entlang einer übergeordneten Prozessklassifikation in Management-, Kern- und Unterstützungsprozesse dargelegt. Diese Geschäftsprozesse werden in Kapitel 3.1.1.3 im Kontext des Entwicklungsnetzwerkes,

⁶⁹ Alpar et al. 2019.

der dazugehörigen Systempartner und dem eigentlichen Entwicklungsziel hinsichtlich ihres Agilisierungspotenzials untersucht. Abschließend liefert die Anwendung der in Kapitel 3.1.1.4 beschriebene funktionsorientierte Wertstromanalyse eine Dokumentation und Modellierung von Material- und Informationsflüssen zwischen Aktivitäten innerhalb der Prozesslandschaft.

3.1.1.1 Identifizierung von Agilisierungspotenzialen in Entwicklungsnetzwerken

Die Systemanalyse des Entwicklungsnetzwerks erlaubt die Evaluation der Zusammenarbeit der verschiedenen Entwicklungspartner. Unterstützt wird diese Systemanalyse durch die Nutzung einer Morphologie zur Klassifikation von Systemen. Unter einem System wird in diesem Zusammenhang eine Menge von miteinander verknüpften Elementen verstanden, die sich insgesamt von ihrer Umgebung abgrenzen lassen. Systeme innerhalb eines Systems sind folglich unter dem Begriff Subsystem aufzufassen. Prinzipiell eignen sich Morphologien in diesem Kontext, um die Struktur und Form von Systemen zu beschreiben und zu untersuchen.

Das Ziel der dargestellten Morphologie ist die modellhafte Abbildung von Entwicklungsnetzwerken in der Windenergiebranche. Unter einem Modell wird hier „das Ergebnis eines Konstruktionsprozesses [verstanden], das die Wahrnehmung von Inhalten eines ausgewählten Gegenstands zweckorientiert repräsentiert“⁷⁰. Eine Morphologie mit Merkmalen und Ausprägungen zur Beschreibung eines Entwicklungsnetzwerks als Systems ist in Abbildung 8 dargestellt.

Systeme können hinsichtlich ihrer Entstehung natürlich- (z.B. Ökosysteme) oder, als künstliche Systeme, durch Menschen geschaffen worden, sein (z.B. Maschinen und Anlagen). Die Elemente eines Systems lassen sich anhand der Ausprägungen natürlich, maschinell oder gemischt beschreiben. Dabei kann ein System tangibel existieren und damit konkret sein oder nur ein abstraktes Gedankenkonstrukt darstellen. Systeme stehen zumeist in Interaktion mit ihrer Umwelt und können auf diese reagieren. Reagiert ein System auf äußere Einflüsse, so ist es als offen zu bezeichnen. Reagiert ein System nicht auf äußere Änderungen lässt es sich als geschlossen charakterisieren. Hinsichtlich des Verhaltens eines Systems kann zwischen voraussagbaren, deterministischem

⁷⁰ vom Brocke 2003.

Verhalten, dem stochastischen Verhalten entsprechend einer Wahrscheinlichkeitsfunktion oder einem rein zufälligen Verhalten unterschieden werden. Hinsichtlich seiner Anpassung können Systeme auf Umwelteinflüsse oder andere äußere Einflüsse adaptiv oder nicht-adaptiv sein. Abschließend lassen sich Systeme hinsichtlich ihrer Steuerbarkeit unterscheiden. Wird bspw. eine Zielvorgabe auf Einhaltung überprüft und Abweichungen werden festgestellt, so lassen sich bei Systemen mit Rückkopplung über Input/Output Größen Veränderungen steuern. Systeme ohne Rückkopplung lassen sich nicht modifizieren.

Systemklassifikation		Legende	
		<input checked="" type="checkbox"/>	Beispiel Windenergieanlage
Merkmal	Ausprägung		
Entstehung	natürlich	<input checked="" type="checkbox"/>	künstlich
Komponenten	maschinell	natürlich	<input checked="" type="checkbox"/> maschinell und natürlich
Existenz	abstrakt	<input checked="" type="checkbox"/>	konkret
Umweltinteraktion	<input checked="" type="checkbox"/> offen		geschlossen
Verhalten	deterministisch	<input checked="" type="checkbox"/>	stochastisch
Anpassung	<input checked="" type="checkbox"/> adaptiv		nicht-adaptiv
Steuerung	<input checked="" type="checkbox"/> mit Rückkopplung		ohne Rückkopplung

Abbildung 8: Morphologie zur Beschreibung von Entwicklungsnetzwerken

Zudem stellt Abbildung 8 die Charakterisierung eines Entwicklungsnetzwerks in der Windenergiebranche dar. Das Unternehmen mit seinem Netzwerk ist ein künstlich erschaffenes System, das über reale Mitarbeiter und Maschinen verfügt. Es agiert mit anderen Unternehmen und zeigt dabei branchenübliche Verhaltensmuster. Auf Veränderungen wird mit entsprechenden Maßnahmen und Erfolgskontrollen reagiert.

Im Kontext der Analyse des Entwicklungsnetzwerks müssen zudem die an der Entwicklung beteiligten Entwicklungspartner identifiziert und klassifiziert werden. Zu den relevanten Partnern gehören hier alle an der Entwicklung beteiligten Lieferanten, Entwicklungsunternehmen, Dienstleister und Kunden. Zur Ermittlung der Systempartner können folgende Interviewfragen genutzt werden, die den an der Entwicklung beteiligten Personen gestellt werden können:

- Mit welchen **Lieferanten** arbeiten Sie bereits während der Produktentwicklung zusammen?

- Mit welchen **Entwicklungspartnern** oder **Dienstleistern** arbeiten Sie bereits während der Produktentwicklung zusammen?
- Mit welchen **Kunden** arbeiten Sie bereits während der Produktentwicklung zusammen?

Die auf diese Weise ermittelten Systempartner können hinsichtlich der jeweiligen Kooperationsform, die zwischen Unternehmen und Partner etabliert ist, und hinsichtlich der Schnittstelle klassifiziert werden. Eine Morphologie zur Beschreibung von Systempartnern und Schnittstellen ist in Abbildung 9 dargestellt.

Die Richtung der Kooperation zwischen Unternehmen und Partner lässt sich anhand der Ausprägung vertikal, horizontal und diagonal beschreiben. Unter dem Begriff horizontale Kooperationen ist die Zusammenarbeit zwischen Unternehmen auf der gleichen Wertschöpfungsstufe aufzufassen. Innerhalb von vertikalen Kooperationen arbeiten Unternehmen unterschiedlicher Wertschöpfungsstufen zusammen. Abschließend verbinden diagonal Kooperation zumeist unterschiedliche Wertschöpfungsketten. Bezüglich des Zeithorizonts lassen sich Beziehungen unter Berücksichtigung der Ausprägung langfristig, mittelfristig und kurzfristig unterscheiden. Ferner beschreibt das Merkmal „Zugang zu Daten“, ob die Systempartner begrenzten oder unbegrenzten Zugriff auf gegenseitige Datenbestände haben. Das Merkmal „Fixierung von Absprachen“ beschreibt den Formalisierungsgrad der Beziehung. Dies kann anhand von Verträgen, als formalisierte und rechtlich bindende Schriftstücke, Spielregeln, also Leitlinien der Zusammenarbeit auf Basis eines gemeinsamen Konsenses oder mündlichen Absprachen, also informellen und spontanen Vereinbarungen, geschehen. Abstimmungen können hinsichtlich ihrer Frequenz entlang der Kriterien Häufigkeit und Regelmäßigkeit unterschieden werden. Es ergeben sich einmalige Abstimmungen, regelmäßiger Austausch, etwa turnusgemäß, und kontinuierliche Austausch in Form eines engen, stetigen Kontakts verbunden mit kurzen Abstimmungszyklen. Zusätzlich ist die Art der Informationsübermittlung vertiefend zu betrachten. So fokussiert eine reine Informationsübermittlung eine einseitige Übermittlung von Informationen, ein Informationsaustausch einen bidirektionalen Prozess und ein Informationsaustausch mit Abstimmungsprozess einen bidirektionalen Prozess mit Rücksprache und Austausch. Aufbauend finden sich System-

partner in den Rollen des Kunden, Lieferanten, Branchen-Wettbewerber oder des branchenfremden-Wettbewerber. Abschließend kann der Typ der Lieferantenbeziehung entlang der Ausprägung Build to Print (BTP), also ein Auftragsfertiger mit Fertigung von Produkten gemäß kundenseitiger Arbeitsanweisungen (bspw. Komponentenfertigung), oder Build to Spec (BTS), also einer Fertigung nach kundenvorgegebenen Spezifikationen und der Entwicklung von Lösungen für Probleme unter bestimmten Rahmenbedingungen, unterschieden werden. Zudem zeigt Abbildung 9 die beispielhafte Ausfüllung der Morphologie zur Beschreibung der Kooperation und Schnittstelle zwischen einem Auftragsentwickler für Windenergieanlagen und einem Lieferanten für Generatoren.

Systempartner- und Schnittstellenklassifikation				Legende
Merkmal	Ausprägung			(x) Beispielhafte Ausprägung
Richtung der Beziehung	(x) vertikal	horizontal	diagonal	
Zeithorizont	(x) langfristig	mittelfristig	kurzfristig	
Zugang zu Daten	unbegrenzt		(x) begrenzt	
Fixierung von Absprachen	(x) Verträge	Spielregeln	Mündliche Absprache	
Frequenz der Abstimmung	einmalige Abstimmung	regelmäßiger Austausch	(x) kontinuierlicher Austausch	
Informationsübermittlung	reine Informationsübermittlung	Informationsaustausch	Informationsaustausch mit Abstimmungsprozess	
Systempartner	Kunde	(x) Lieferant	Branchen-Wettbewerber	Branchenfremder-Wettbewerber
Lieferantenbeziehung	BTP (Build to Print)		(x) BTS (Build to Spec)	

Abbildung 9: Morphologie zur Beschreibung von Systempartnern und Schnittstellen

3.1.1.2 Geschäftsprozessanalyse

Nach der Analyse des Entwicklungsnetzwerks und der Beschreibung der Partner im Entwicklungsnetzwerk anhand der etablierten Schnittstellen folgt die Beschreibung und Analyse der internen Analyse des bestehenden Entwicklungsprozesses mithilfe der Geschäftsprozessanalyse. Ziel und Zweck der Geschäftsprozessanalyse ist die Ermittlung und Bewertung der relevanten Geschäftsprozesse aus Entwicklungssicht. Im Kontext der Windenergiebranche wurde dazu eine Methode zur Ermittlung entwicklungsrelevanter Prozesse unter Berücksichtigung der Geschäftsprozesse der Systempartner entwickelt. Aufbauend

wurde eine Methode zur Analyse von Schnittstellen zwischen entwicklungsrelevanten Geschäftsprozessen entwickelt.

Geschäftsprozesse können über Abteilungs- und Betriebsgrenzen hinweggehen und gehören zur Ablauforganisation eines Unternehmens. Sie unterstützen ein unternehmensbezogenes Ziel und sind damit kontextgebunden. Sie bestehen aus mehreren Einzelschritten bzw. Aktivitäten. Sie werden regelmäßig unter der Beteiligung mehrerer Personen, Bereiche oder Abteilungen durchlaufen und erfordern in der Regel Ressourcen⁷¹. Dabei verarbeiten Geschäftsprozesse Informationen (Input) zu einem vom Unternehmen gewünschten Ergebnis (Output)⁷². Zusammenfassend ist ein Geschäftsprozess also eine Reihe an verbundenen Aktivitäten, an deren Ende die Bereitstellung einer Dienstleistung oder eines Produkts für den (internen oder externen) Kunden steht.

Aktivitäten werden ausgelöst, um im Rahmen des Geschäftsprozesses ein spezifisches Ergebnis zu erzeugen. Sie sind zeitlich und aufgabenbezogene Handlungen und im Gegensatz zu Geschäftsprozessen nicht kontextgebunden, sondern können in unterschiedlichen Kontextsituationen auftreten.

Zur effizienten Identifikation der Geschäftsprozesse eignet sich die Durchführung von Interviews⁷³. Innerhalb des Forschungsprojekts wurden vier zentrale Interviewleitfragen formuliert, um die entwicklungsrelevanten Geschäftsprozesse zu identifizieren.

- Welche **Abteilungen** in Ihrem Unternehmen sind an der Produktentwicklung beteiligt? (Ergebnis: Abteilung, Personen)
- Welche **Prozesse** sind in Ihrem Unternehmen im Kontext der Produktentwicklung definiert worden und welche Abteilungen sind daran beteiligt? (Ergebnis: Prozess)
- Welcher **Input** wird von der Produktentwicklung benötigt? Wer stellt den benötigten Input zur Verfügung? (Ergebnis: Input)
- Welcher **Output** wird von der Produktentwicklung generiert? Wer ist Empfänger dieses Outputs? (Ergebnis: Output)

⁷¹ Gadatsch 2020, 5 ff.

⁷² Hilmer 2016, 267 f.

⁷³ Gadatsch 2020, 5 f.

Die auf Basis dieser Leitfragen identifizierten entwicklungsrelevanten Geschäftsprozesse können anschließend in Management-, Kern- und Unterstützungsprozesse unterteilt werden⁷⁴, wobei jede Art von Geschäftsprozessen spezifische Merkmale aufweist⁷⁵. Kernprozesse sind kundenorientiert, haben eine hohen Wertschöpfungsgrad, decken die gesamte Wertschöpfungskette ab und sind entscheidend für den Unternehmenserfolg. Konkret lassen sich hier zentrale Prozesse in Verbindung mit der Entwicklung einer Windenergieanlage anführen. Managementprozesse steuern Kernprozesse, stellen eine Struktur für die Organisation bereit, haben keinen direkten Wertschöpfungsbeitrag und sind den Kernprozessen übergeordnet. Als Beispiel lassen sich etwa Prozesse in Verbindung mit der Geschäftsführung anführen. Unterstützungsprozesse tragen mittelbar zur Wertschöpfung bei, stellen die notwendige Infrastruktur für die Kernprozesse bereit, sind den Kernprozessen untergeordnet, im Allgemeinen nicht sichtbar für den Kunden und können prinzipiell auch ausgelagert werden. Als Beispiel lässt sich etwa der Einkaufsprozess anführen. Eine Übersicht über die drei Geschäftsprozessarten mit einer Unterscheidung der Kernprozesse hinsichtlich ihrer zeitlichen Abfolge ist in Abbildung 10 dargestellt.



Abbildung 10: Exemplarische Prozessstrukturierung im Kontext von Entwicklungstätigkeiten

⁷⁴ Hilmer 2016, 267 ff.

⁷⁵ Schmelzer und Sesselmann 2004, 46 ff.

Im Zusammenhang mit der Identifizierung von relevanten Entwicklungsprozessen und -aktivitäten und der Ableitung des Agilisierungspotenzials ist stets ein konkreter Wertschöpfungsfokus, das heißt die Vermeidung von Verschwendung zu fokussieren und ein Kundenfokus, in Form der Ausrichtung sämtlicher Unternehmensaktivitäten auf den Kunden, zu beachten und sicherzustellen. Vor diesem Hintergrund ist ein Kernelement der Geschäftsprozessbewertung die Unterteilung der identifizierten Geschäftsprozesse in Kern-, Management- und Unterstützungsprozesse. Die weitere Analyse zur Identifikation von Agilisierungspotenzialen erfolgt stets aus Kundensicht und sollte sich daher insbesondere auf die im Unternehmen identifizierten Kernprozesse beziehen.

Hierzu sind zunächst die identifizierten Geschäftsprozesse mit Entwicklungsbezug in Management-, Kern- und Unterstützungsprozesse zu unterteilen. Aufbauend auf dieser Unterteilung sind im Anschluss, bei Bedarf, übergeordnete Prozesse in weitere Teilprozessschritte aufzulösen, um eine ausreichend feine Granularität der Betrachtungsebene zu gewährleisten. Als kleinste Einheit ist hierbei die Abteilung zu wählen, da hier konkrete Entwicklungsumfänge definiert werden können und eine Funktionsorientierung sichergestellt ist.

Abbildung 11 zeigt eine beispielhafte Prozessanalyse der entwicklungsrelevanten Geschäftsprozesse in einem Unternehmen der Windenergiebranche.

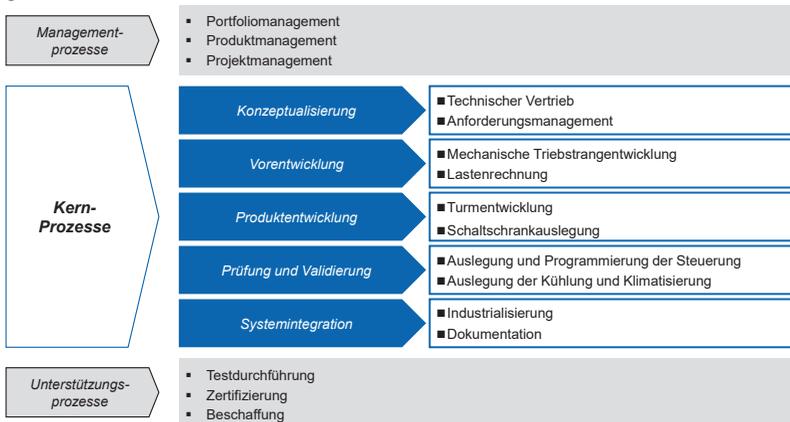


Abbildung 11: Beispielhafte Geschäftsprozesse in der Windenergiebranche

Zur Realisierung des Unternehmenszwecks und damit insbesondere für die Erstellung einer Leistung ist das Zusammenwirken der identifizierten

Geschäftsprozesse nötig. Die dabei entstehenden Schnittstellen innerhalb des Unternehmens erweisen sich dabei oft als Reibungspunkte, an denen Konflikte auftreten, Informationsverluste entstehen oder zusätzliche Kosten verursacht werden. Insbesondere vor dem Hintergrund der Agilisierung von Prozessen müssen daher insbesondere die internen Schnittstellen besonders berücksichtigt werden, da die agile Arbeitsweise einen starken Einfluss auf die Ausgestaltung dieser Schnittstellen haben kann.

Im Kontext der Betrachtung von Entwicklungsaktivitäten im Kontext der Windenergiebranche kann das folgende im Forschungsprojekt entwickelte Deskriptionsmodell in Abbildung 12 unterstützen. Der Standardisierungsgrad der Schnittstelle gibt an, wie standardisiert der Prozess an der jeweiligen Schnittstelle ist. Zeichnet sich der Prozess durch eine vollständige Standardisierung aus, lässt dies auf eine eher plangesteuerte Schnittstelle schließen, da Prozesse und Abläufe im Vorfeld größtenteils bekannt sind. Andererseits zeichnet sich eine nicht-standardisierte Schnittstelle durch eine hohe Dynamik und vorherrschende Unsicherheit aus. Die Schnittstelle weist demnach ein hohes Agilisierungspotenzial auf. Lassen sich sowohl Standardisierungen als auch dynamische Verhaltensmuster erkennen ist von einer teilweise standardisierten Schnittstelle zu sprechen. Hinsichtlich der Art der Informationsübermittlung ist zwischen einer Informationsübermittlung ausschließlich in eine Richtung, ohne einen Informationsaustausch, und einer beidseitigen Schnittstelle, mit bidirektionalem Informationsaustausch, zu unterscheiden. Ferner kann dieser bidirektionale Informationsausschluss um einen verbundenen Abstimmungsprozess erweitert und intensiviert werden. Darüber hinaus kann eine Schnittstelle anhand ihres zeitlichen Rahmens charakterisiert werden. Langfristig bestehende Schnittstellen setzen eine gewisse Planbarkeit voraus, während kurzfristige Schnittstellen eher für Dynamik und Flexibilität stehen und somit eher agil sind. Die Art des übergebenen Objekts lässt sich anhand des physischen Charakters unterscheiden. Entweder das Objekt ist physisch und wird physisch übergeben, bspw. in Form eines Prototyps, oder das Objekt ist virtuell und wird transferiert, bspw. Daten. Dabei kann das übergebene Objekt einen geringen Informationswert, bspw. in Form von (unverarbeiteten) Daten, einen mittleren Informationswert, bspw. in Form von Informationen, oder einen hohen Informationswert, bspw. in Form

von Wissen, besitzen. Im Weiteren sind Schnittstellen anhand ihrer benötigten Flexibilität zu unterscheiden. Hohe Flexibilität wird benötigt, wenn bspw. Ansprechpartner häufig wechseln oder unterschiedliche Dateiformate vorliegen. Der Bedarf an mittlerer Flexibilität wird bspw. durch teilweise standardisierte Dateiformate hervorgerufen. Geringe Flexibilität benötigt eine Schnittstelle hingegen, wenn Austauschformate standardisiert sind und definierte Ansprechpersonen festgelegt sind. Zusätzlich kann die Relevanz der Schnittstelle klassifiziert werden. Ist diese als hoch einzustufen, werden zumeist wichtige Objekte ausgetauscht. Eine mittlere Relevanz liegt vor, wenn relevante Objekte ausgetauscht werden. Eine geringe Relevanz ist im Austausch von eher unwichtigen Objekten begründet. Ferner gibt es Schnittstellen, an denen der Informationsaustausch sehr präzise sein muss, da das Risiko großer wirtschaftlicher Schäden im Falle von Missverständnissen vorherrscht. In diesen Fällen ist von einem hohen Grad der Korrektheit zu sprechen. Andere Schnittstellen dienen eher zur Information über Ereignisse oder Sachverhalte, ohne dass diese notwendigerweise in gänzlicher Korrektheit beschrieben werden müssen. Hier lassen sich die Ausprägungen mittel oder gering in Bezug auf das Merkmal des Grads der Korrektheit aufzeigen.

Beschreibung von Geschäftsprozessschnittstellen		Legende	
		(x)	Beispielhafte Ausprägung
Merkmal	Ausprägung		
Standardisierungsgrad der Schnittstelle	nicht-standardisiert	(x) teilweise standardisiert	voll-standardisiert
Art der Informationsübermittlung	Reine Informationsübermittlung	Informationsaustausch	(x) Informationsaustausch mit Abstimmungsprozess
Zeitlicher Rahmen der Schnittstelle	(x) langfristig	mittelfristig	kurzfristig
Art des übergebenen Objekts	physisch		(x) virtuell
Informationswert des übergebenen Objekts	Daten	Information	Wissen
Benötigte Flexibilität an der Schnittstelle	hoch	mittel	(x) gering
Relevanz der Schnittstelle	(x) hoch	mittel	(x) gering
Grad der Korrektheit	(x) hoch	mittel	gering

Abbildung 12: Morphologie zur Beschreibung von Schnittstellen zwischen Geschäftsprozessen

Beispielhaft wurde dieses Deskriptionsmodell im Kontext der Entwicklung einer Windenergieanlage zur Beschreibung der Schnittstelle zwischen der Blattentwicklung und Lastenrechnung in Abbildung 12 angewendet. Die Blattentwicklung übergibt das 3D Modell des Rotorblatts an die Lastenrechnung, die daraus ein Lastenmodell entwickelt. In diesem anonymisierten Anwendungsbeispiel ist die Schnittstelle nur teilweise standardisiert, da zusätzlich zur Ablage des 3D Modells auf den Servern der Lastenrechnung auch eine E-Mail von der Blattentwicklung an die Lastenrechnung mit weiteren unstrukturierten Informationen übermittelt wird. In Form einer iterativen Abstimmung wird die Blattentwicklung fortgeführt, bis das finale Blatt entwickelt ist. Der zeitliche Rahmen dieser exemplarischen Schnittstelle ist als langfristig zu kennzeichnen, da die Schnittstelle über eine lange Zeit und für viele verschiedene Projekte genutzt wird. Das übergebene Objekt ist eine Datei und daher virtuell. Übergeben wird an dieser Schnittstelle Wissen darüber, ob die Blattkonstruktion den Belastungsfällen entsprechend ausgelegt ist oder angepasst werden muss. Das Wissen besteht in der notwendigen Erfahrung zur Auswertung der Ergebnisse der Lastenrechnung. Die Flexibilität der betrachteten Schnittstelle ist gering, da der Austausch stetig gleich abläuft. Die Schnittstelle ist von hoher Relevanz, da sie essenziell für die Entwicklung des Blattes ist und sicherheitstechnische Belange adressiert. Der Grad der Korrektheit ist abschließend ebenfalls als hoch zu beurteilen, da eine fehlerhafte Übermittlung einen erheblichen Einfluss auf das gesamte Entwicklungsprojekt hat und letztlich zur einer Funktionsbeeinträchtigung des Produkts, mit hohen Folgekosten, führen kann.

Nachfolgend findet sich in Abbildung 13 das exemplarische Ergebnis der Anwendung des im Forschungsprojekt entwickelten Deskriptionsmodells zur Beschreibung der entwicklungsrelevanten Geschäftsprozessschnittstellen in Bezug auf die Art der Informationsübermittlung eines Windenergieanlagenentwicklers. Als Darstellungsform wurde ein grafisch, heuristisches Verfahren der Ressourcenplanung in Form des Kreisverfahrens nach SCHWERTDFEGER⁷⁶ gewählt. Dieses Verfahren nutzt die Ergebnisse der vorangegangenen Geschäftsprozessanalyse und lässt Rückschlüsse auf die Beziehungen der einzelnen Prozesse untereinander zu.

⁷⁶ Schwerdtfeger 1999.

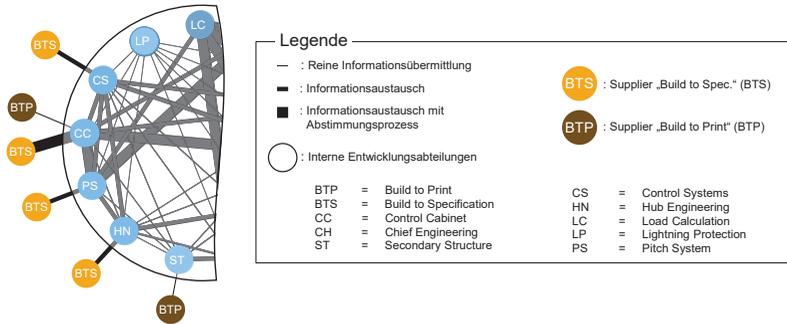


Abbildung 13: Exemplarische Darstellung der entwicklungsrelevanten Geschäftsprozessschnittstellen

Aus der Beschreibung der identifizierten entwicklungsrelevanten Geschäftsprozesse und Schnittstellen wird im weiteren Verlauf das Agilisierungspotenzial der jeweiligen Geschäftsprozesse und an den entsprechenden Schnittstellen abgeleitet und als Input für die Konzeption eines agilen datenbasierten Produktentwicklungsprozesses genutzt.

3.1.1.3 Bewertung des Agilisierungspotenzials von Geschäftsprozessen und Schnittstellen

Im Kontext der Bewertung des Agilisierungspotenzials wurden im Forschungsprojekt unterschiedliche relevante Dimensionen mit zugehörigen Beschreibungsmerkmalen identifiziert. Die Dimensionen Kunde, Dynamik, Umfang, Kritikalität, Output, Personal und Organisation dienen als Indikatoren, um das Agilisierungspotenzial bestimmter Geschäftsprozesse im Kontext der Produktentwicklung innerhalb der Windenergiebranche zu bestimmen.

Die Dimension „**Kunde**“ betrachtet Schnittstellen zu anderen Geschäftsprozessen und zu externen Partnern. Der Kunde ist der Empfänger des im Prozess generierten Outputs. Dabei kann es sich um interne Kunden, also bspw. das Team eines nachfolgenden Geschäftsprozesses oder externe Kunden, wie etwa die Empfänger des entwickelten Produkts handeln. Während im klassisch-plangetriebenen Entwicklungsprozess Kunden ausschließlich als Auftraggebende auftreten, mit denen ein Vertrag ausgehandelt wird, dessen Erfüllung im weiteren Projekt einziger Berührungspunkt zwischen Entwicklung und Kunden ist, zielt die agile Projektorganisation auf eine intensivierete *Zusammenarbeit*. Die Kunden werden aktiv in den Entwicklungsprozess einbezogen.

Unabhängig vom Verhältnis zu den Kunden ist ihre *Verfügbarkeit* relevant. Ein agiles Vorgehen erfordert die regelmäßige Abstimmung, wenn die entsprechenden Kapazitäten auf Kundenseite nicht vorhanden sind, können sie nicht aktiv in agile Prozesse einbezogen werden und an agilen Prozessen partizipieren. Im Weiteren ist die jeweilige *Vorgehensaffinität* zu berücksichtigen. So haben bspw. Kunden mit Erfahrung im Bereich der plangetriebenen Ansätze ein grundlegendes Verständnis für die Projektplanung anhand definierter Meilensteine und formaler Prozesse. Auf der anderen Seite sind Kunden, die selbst Erfahrung mit agilen Prozessen haben, eher bereit bspw. iterative Änderungen an der ursprünglichen Planung nachzuvollziehen und im Rahmen der agilen Methoden mitzugestalten.

Die Dimension **Dynamik** beschreibt die zeitliche Veränderung des Entwicklungsauftrags und umfasst alle Formen von Unsicherheit bei der Erstellung von Anforderungen an das Produkt. Sind die Anforderungen an das Ergebnis eines Entwicklungsschrittes umfassend beschrieben und damit die Problemstellung klar definiert, eignet sich eher der Einsatz klassischer, plangetriebener Ansätze. Eine Entwicklungsaufgabe mit offener Aufgabenstellung erfordert hingegen ein hohes Maß an iterativer, flexibler Problemlösungsarbeit. Dabei eignet sich eher die Anwendung agiler Entwicklungsmethoden. Neben der *Vollständigkeit* der Anforderungen ist auch ihre *zeitliche Stabilität* zu berücksichtigen. Gerade die Windenergiebranche zeichnet sich durch die in Kapitel 1.1.1 erläuterten „Moving Targets“ aus. Wie gezeigt, weisen hier klassische, plangetriebene Ansätze eine Limitation in Bezug auf die Reaktionsfähigkeit und Adressierung der veränderten Zielgrößen auf. Auch der *Innovationsgrad* oder die Neuartigkeit des zu entwickelnden Produkts wird betrachtet. Für Projekte mit hohem Neuentwicklungsgrad bietet sich eher die Nutzung agiler Ansätze an⁷⁷. Ist das Problem hingegen grundsätzlich bekannt und nur zu einem kleinen Grad neu, besteht in der Regel eine Anzahl bewährter Lösungsverfahren für das Entwicklungsprojekt. Diese können klassisch, plangetrieben durchlaufen werden⁷⁸.

Die Dimension „**Umfang**“ erfasst den zu erbringenden Aufwand. Agile Methoden weisen einen höheren Wirkungsgrad auf, wenn die *Anzahl*

⁷⁷ Wessels et al. 2019.

⁷⁸ Klein 2016.

der involvierten Akteure begrenzt und gering ist. In umfangreichen Projekten mit einer Vielzahl beteiligter Akteure erweisen sich prinzipiell plangetriebene Ansätze als zielführender⁷⁹. BREHM et al. sehen in der Entwicklungszeit einen Faktor zur Bestimmung der Eignung der Anwendung agiler Methoden. Kürzere *Entwicklungsdauern* ließen prinzipiell auf die Anwendbarkeit von agilen Prinzipien schließen, bei längerfristigen Prinzipien sei die Integration plangetriebener Ansätze zu prüfen⁸⁰. Nach SCHUH lässt sich grundsätzlich zwischen repetitiven und kreativen Aktivitäten im Produktentwicklungsprozess unterscheiden⁸¹. Während *kreative Tätigkeiten* durch einmalige Bedingungen und eine geringe Standardisierung gekennzeichnet sind, werden repetitive Tätigkeiten im Rahmen vieler Entwicklungsaufgaben in ähnlicher oder identischer Weise bearbeitet. Für die Durchführung repetitiver Aufgaben eignen sich daher eher standardisierte Vorgehensweisen innerhalb plangetriebener Vorgehensmodelle, während für kreative Tätigkeiten agile Methoden Nutzenpotenzial bieten.

Unter der Dimension „**Kritikalität**“ wird die Wichtigkeit eines anforderungsgerechten Ergebnisses des Geschäftsprozesses betrachtet. Hierunter fallen insbesondere Folgen für das Projekt, das Unternehmen oder die Umwelt, sollten Fehler im Output der Aktivität vorliegen. Je größer der potenzielle *Schaden eines Fehlers für das Gesamtprojekt* ist, desto wichtiger ist die Anwendung umfangreicher, standardisierter Kontrollmaßnahmen⁸². Zusätzlich sind bei Prozessen mit besonderer Wichtigkeit für den weiteren Projektverlauf standardisierte, plangetriebene Prozessauslegungen zu fokussieren, um einen anforderungsgerechten Ablauf der Aktivität sicherzustellen⁸³. Nach COCKBURN sind hinsichtlich der *Folgen von Fehlern für das Unternehmen* die Stufen „Unannehmlichkeiten, Verlust ersetzbarer Gelder, Verlust essenzieller Gelder und Verlust von Menschenleben“ zu unterscheiden⁸⁴. Gemäß dieser Klassifikation der Schwere von Fehlerfolgen liegt die größte Kritikalität dann vor, wenn die Auswirkungen von Fehlern unmittelbar zur Schädigung der Gesundheit oder des Lebens von Menschen führt. Auf die Entwicklung einer Windenergieanlage projiziert, tritt eine hohe Kritikalität also

⁷⁹ Boehm und Turner 2004, S. 33.

⁸⁰ Brehm et al. 2017, 34 f.

⁸¹ Schuh 2013, 15 f.

⁸² Cockburn 2003, S. 126.

⁸³ Boehm und Turner 2004, S. 23.

⁸⁴ Cockburn 2003, S. 126.

insbesondere dann auf, wenn sicherheitsrelevante Systeme betrachtet werden. Hier sind plangetriebene Mechanismen, zwecks Standardisierung, Kontrolle, Dokumentation und Konformität mit Vorschriften, zu bevorzugen. Ergänzend ist zu evaluieren, wann (potenzielle) Fehler erkannt werden können. Ist zu erwarten, dass Fehler frühzeitig erkannt werden, können diese etwa in folgenden Iterationen behoben werden. Erfolgt eine *Fehlererkennung* nicht oder erst später im Entwicklungsprozess, sind entsprechende standardisierte und plangetriebene Prozesse, die die Fehlerwahrscheinlichkeit minimieren können, zu präferieren. Innerhalb der Dimension „**Output der Aktivität**“ wird zunächst betrachtet, inwiefern *Regularien* zu entsprechen ist. Liegen beispielsweise genormte plangetriebene Vorgehensweisen vor, um ein Produkt entsprechend exogen vorgegebener Richtlinien zu entwickeln, können agile Prozessstrukturierungen keine Anwendung finden. Der große Aufwand zur Herstellung von Produktinkrementen in der physischen Produktentwicklung stellt ein wesentliches Hindernis für die Anwendung agiler Methoden dar⁸⁵. Da eine agile Prozessstrukturierung regelmäßig funktionsfähige Prototypen bereitstellt und evaluiert, ist der *Erstellungsaufwand für Prototypen* ein wichtiges Entscheidungskriterium zum Einsatz agiler oder plangetriebener Ansätze⁸⁶. Die Anwendung agiler Methoden ist immer dann eingehender zu prüfen, wenn für die Entwicklung von Bauteilen Entwicklungsumfänge mit einer Mischung aus mechanischen, elektrischen und informationstechnischen Elementen vonnöten sind. Die Dimension „**Personal**“ betrachtet die Eigenschaften des Entwicklungsteams. Wesentlich für die Eignung zur Nutzung agiler Methoden ist die Fähigkeit zur selbstverwalteten, kooperativen Erfüllung der gestellten Aufgabe⁸⁷. Nur wenn alle *Kompetenzen* vorhanden sind, ist das Team in der Lage, selbstverwaltet für alle an das Produkt gestellten Anforderungen Lösungen zu erarbeiten⁸⁸. Sind die benötigten Kompetenzen nicht durchgängig im Team verfügbar, eignet sich eine agile Prozessgestaltung eher weniger und plangetriebene Ansätze mit einer örtlichen und zeitlichen Ressourcenplanung sind zu präferieren⁸⁹. Inner-

⁸⁵ Brehm et al. 2017, S. 31.

⁸⁶ Diels 2017, 153 f.

⁸⁷ Siakas und Siakas 2007, S. 602.

⁸⁸ Chow und Cao 2008.

⁸⁹ Boehm und Turner 2004.

halb von Entwicklungsprojekten entlang eines plangetriebenen Ansatzes führen oftmals einzelne Beteiligte Teilaktivitäten allein und individuell durch. Agile Vorgehensweisen weisen das Potenzial auf, durch einen kontinuierlichen Austausch *personelle Fluktuationen* besser zu kompensieren. Zur erfolgreichen Anwendung agiler Arbeitsweisen benötigen die Mitarbeitenden *Methodenkenntnisse* im Bereich der Agilen Prinzipien, bzw. müssen die Eignung aufweisen schnell neue methodische Fähigkeiten zu entwickeln.

Die Dimension **Unternehmensorganisation** betrachtet die Gesamtheit der Arbeitsweisen unterschiedlicher struktureller Bestandteile eines Unternehmens. Agile Vorgehensweisen erfordern innerhalb einzelner Projekte *Selbstständigkeit* gegenüber einer vorgegebenen Aufbauorganisation. Ist die Organisation stark linienorientiert und fordert ständige Rechenschaft gegenüber einer Kontrollinstanz, erschwert dies die Arbeit eines agilen Entwicklungsteams und spricht eher für ein plangetriebenes Vorgehen⁹⁰. Die Anwendung agiler Methoden setzt häufige, intensive Kommunikationsprozesse voraus, verbunden mit regelmäßigen Meeting-Formaten und enger Zusammenarbeit. Eine gute *Vernetzung* der Mitglieder des Projektteams durch die Bereitstellung physischer und/oder digitaler Infrastruktur ist folglich der Anwendung agiler Methoden zuträglich. So genannte demokratische Unternehmen bieten durch flache *Hierarchie* und die Charaktereigenschaften der Flexibilität, Spontaneität und Eigenverantwortung den besten Rahmen für die Anwendung agiler Prinzipien⁹¹. Je stärker hierarchische und rigide Strukturen im Unternehmen und dessen Projektmanagement ausgeprägt sind, desto aufwändiger gestaltet sich die Integration agiler Mechanismen.

Die sieben Dimensionen zur Bewertung des Agilisierungspotenzials sind in Abbildung 14 zusammenfassend dargestellt. Dabei sind die Dimensionen Personal und Organisation insofern als veränderlich zu verstehen, als dass Unternehmen die Ausprägungen dieser jeweils anpassen können, um einem agilen bzw. plangetriebenen Vorgehen zu entsprechen. Die Dimensionen Kunde, Dynamik, Umfang, Kritikalität und Output sind entsprechend als nicht-veränderlich zu verstehen, weil ihre Ausprägungen wesentlich durch die konkrete Entwicklungsaufgabe und damit extern determiniert sind. Deshalb werden Sie auch als Agilitätsindikatoren bezeichnet.

⁹⁰ Boehm und Turner 2004, S. 50.

⁹¹ Diels 2017, S. 185.

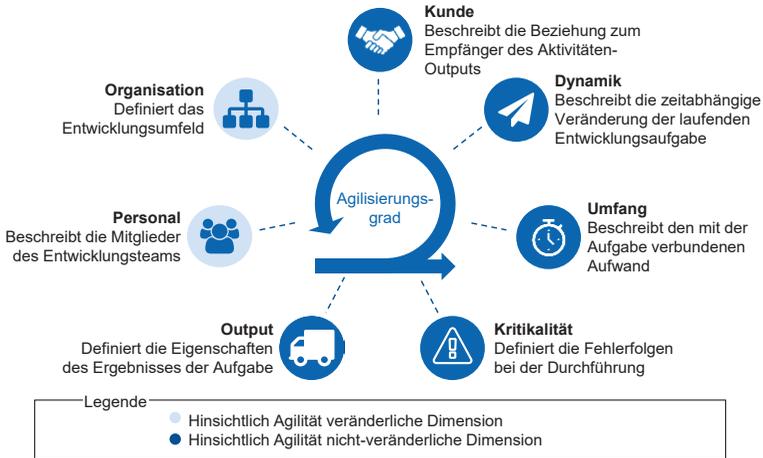


Abbildung 14: Dimensionen zur Bewertung des Agilisierungspotenzials

Um das geeignete Entwicklungsvorgehen für eine gegebene Entwicklungsaufgabe zu bestimmen, werden die Ausprägungen der Agilitätsindikatoren bestimmt. Je nachdem, ob sich ein agiles oder plangetriebenes Vorgehen besser eignet, werden die Dimensionen Organisation und Personal entsprechend angepasst. Das Vorgehen wird in Kapitel 3.3.1.3 näher erläutert. Nach der Evaluierung des Potenzials zur Einführung agiler Methoden ist eine Wertstromanalyse durchzuführen. Die Wertstromanalyse ermöglicht eine Untersuchung der bestehenden Geschäftsprozesse, indem sie Engpässe, Verschwendungen und potenzielle Verbesserungsbereiche aufzeigt. Diese detaillierte Analyse bietet eine Grundlage für die erfolgreiche Implementierung agiler Methoden. Im Kontext der Analysephase wird die Wertstromanalyse insbesondere eingesetzt, um den aktuellen Entwicklungsprozess zur Erzeugung einer Windenergieanlage transparent zu dokumentieren und die Aktivitätenfolge und Informationsflüsse zu modellieren.

3.1.1.4 Funktionsorientierte Wertstromanalyse

Die Wertstromanalyse, im Deutschen auch Wertstromdesign, im Englischen value mapping, value stream design oder value stream analysis genannt, besitzt ihren Ursprung in dem bei Toyota verwendeten „material and information flow diagram“, das dazu diente Zulieferer der Toyota-Werke im Umgang mit dem „Toyota Production System“ zu

schulen⁹². Die folgende Weiterentwicklung der Methode fokussierte eine Darstellung und Analyse von Produktionsprozessen. Wertstromanalysen werden im gegenwärtigen Zeitgeschehen häufig adaptiert und finden Anwendung in einem breiten Spektrum im Kontext von Prozessanalysen. Ziel ist stets, die Verbesserung von Geschäftsprozessen durch eine ganzheitliche Systemperspektive und Kenntnis über einzelne Teilprozesse zu ermöglichen.

Das Instrument der Wertstromanalyse ist vor allem in der Produktion weit verbreitet und erweist sich dort als leistungsfähiges Tool zur Optimierung von Wertschöpfungsprozessen. Auch in der Produktentwicklung hat dieses Verfahren in den vergangenen Jahren stark an Bedeutung gewonnen. Die Methode basiert auf der Identifikation von generischen Aktivitätentypen mit einer wertoptimalen, idealtypischen Ausführungsweise. Analog zur Definition eines idealen Soll-Prozesses, bieten diese Soll-Aktivitäten eine Hilfestellung, um Verschwendung erkennen zu können.

Die funktionsorientierte Wertstromanalyse dient zur Aufnahme und einheitlichen Beschreibung von Aktivitäten sowie deren Interaktionen. Die Funktionsorientierung impliziert dabei, dass Aktivitäten je beteiligter Abteilung aufgenommen und dargestellt werden. Dabei nutzt die funktionsorientierte Wertstromanalyse die folgenden Elemente:

- Aktivitäten (Reale, zielgerichtete Handlungsabläufe, die innerhalb des von den Aufgaben vorgegebenen Handlungsspielraums ausgeführt werden.) Aktivitäten sind kontextungebunden und werden häufig auch als Tätigkeiten bezeichnet.
- Prozess (Reihe von Aktivitäten, die aus einem definierten Input ein definiertes Ergebnis erzeugen.) Prozesse sind stets an ihren Kontext gebunden.
- Swimlanes (Abteilungen oder Personen in deren Verantwortungsbereich die Durchführung der unterschiedlichen Aktivitäten fällt.)

Ein Beispiel für eine funktionsorientierte Wertstromanalyse in Abbildung 15 dargestellt. Fokus der Wertstromanalyse sind dabei die als Kernprozesse identifizierten Geschäftsprozesse eines Unternehmens, die für

⁹² Liker 2004, S. 275.

die Überführung in einen agilen Entwicklungsprozess dokumentiert werden müssen. Die Aufnahme erfolgt interviewbasiert.

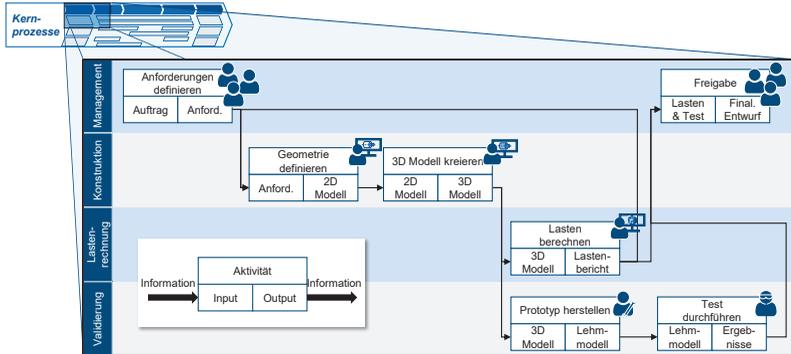


Abbildung 15: Exemplarischer Ausschnitt einer Wertstromanalyse im Kontext der Windenergieanlagenentwicklung

Als methodische Unterstützung eignet sich die Nutzung von Aktivitätenkarten. Mittels Aktivitätenkarten können Prozesse strukturiert dokumentiert und modelliert werden. Eine Aktivitätenkarte, die im Rahmen des Forschungsvorhabens für die Aufnahme des Entwicklungsprozesses in der Windenergiebranche entwickelt wurde, ist in Abbildung 16 beispielhaft dargestellt. Die Aktivitätenkarte nimmt die für die Wertstromanalyse notwendigen Merkmale der Aktivitäten in standardisierter Form auf. Beschreibungsmerkmale, welche eine Aktivitätenkarte unabhängig vom Leitbild aufnehmen muss, sind der Titel der Aktivität, eine eindeutige Identifikationsnummer für die Auswertung, die verantwortliche Abteilung oder Rolle, der Wertschöpfungsgrad, die Dauer, der Input sowie der Output. Daneben können, je nach Leitbild, weitere Informationen wie der Standardisierungsgrad, das verwendete IT-System, die Art und Weise der Informationsaufnahme und Weitergabe, die Wartezeit, beteiligte Abteilungen oder weitere Prozessparameter wie Fehlerquote oder Effizienzkennzahlen sein.

Nr: 2	Aktivität: Benchmarking	Dauer: 8 Stunden	Kr. Pfad 
		Wertschöpfungsgrad:	
Verantwortliche Rollen/Abteilungen: Produktentwicklung		Hauptaktivität:	
Beteiligte Rollen/Abteilungen: Vorentwicklung / Technologieleiter		Nebenaktivität:	
IT-System: MS Office		Organisatorische Aktivität:	
Input: Marktanalysedaten	Input-Datenformat: Excel-Datei	Input-Quelle: Mail	
Output: Tabelle mit Ergebnissen de	Output-Datenformat: Excel-Datei	Daten-Ausgabe: Dateiablage auf internem Server	

Abbildung 16: Aktivitätenkarten zur Unterstützung der funktionsorientierten Wertstromanalyse

Ziel der Prozessgestaltung/-optimierung im Anschluss an eine Wertstromanalyse ist die Vermeidung von Verschwendung und die Erhöhung des Wertschöpfungsanteils innerhalb des Prozesses. Dabei können vier Leistungsarten von Aktivitäten unterschieden werden, die sich hinsichtlich ihres Wertschöpfungsanteils und ihrer Verschwendung unterscheiden:

- Wirkleistung (Direkt wertschöpfende Aktivitäten, bspw. konstruieren in CAD)
- Stützleistung (Indirekt wertschöpfende Nebenaktivitäten, bspw. Dokumentation des Projektfortschritts)
- Blindleistung (Aktivitäten, die nicht notwendig sind und somit keinen Wertbeitrag leisten, bspw. ungeplante Iterationen)
- Fehlleistung (Aktivitäten, die den Wert des Produkts reduzieren, bspw. fehlerhafte Informationsweitergabe)

Um Schwachstellen und Verschwendung im Prozess systematisch identifizieren zu können, gliedert sich die Methode in drei Schritte:

- Informationsaufnahme
- Auswertung

- Ableitung von Gestaltungsmaßnahmen

Zunächst ist eine IST-Prozess Analyse durchzuführen. Diese zielt auf die Identifikation von Aktivitäten und Prozessen. Als Best-Practice hat sich im Forschungsprojekt die Durchführung von workshopbasierten Prozessaufnahmen verbunden mit der Identifizierung und Diskussion von Schwachstellen im Prozess erwiesen. Es empfiehlt sich in Kleingruppen-Workshops insbesondere die Mitarbeiter zu beteiligen, die bei der Ausführung der entsprechenden Aktivitäten selbst teilhaben. Das Ergebnis findet sich in einem visualisierten Prozess mit Aktivitäten und Interaktionen. Ebenfalls ermöglicht diese Form der Prozessaufnahme eine Visualisierung des Informationsflusses entlang des Prozesses und die Darstellung von Medienbrüchen. Darüber hinaus findet sich eine strukturierte Darstellung von Schwachstellen und Potenzialen entlang des aktuellen Entwicklungsprozesses. Aufbauend auf dieser Dokumentation der Ergebnisse lassen sich Teilprozesse für tiefgehende und detaillierte Analysen identifizieren. Die aufgenommenen Wertströme sind Input für die Gestaltung eines agilen Entwicklungsprozesses. Im Folgenden sind diese Ergebnisse, im Sinne eines datenbasierten Vorgehensprinzips, mit der Analyse der Informationssystemarchitektur zu verbinden.

3.1.2 Analyse der Informationssystemarchitektur

Der Entwicklungsprozess für Windenergieanlagen ist durch den Austausch von Entwicklungsdaten zwischen den beteiligten Simulationsmodellen geprägt. Wird beispielsweise der Generator ausgelegt, werden Informationen über Drehzahl der Eingangswelle benötigt. Diese Drehzahl resultiert aus der Drehzahl des Rotors und dem Übersetzungsverhältnis des in der Entwicklung ausgewählten Getriebes. Der Austausch dieser Abhängigkeiten zwischen den verantwortlichen Entwicklungsteams und damit zwischen den genutzten Simulationsmodellen zur Gestaltung einer Windenergieanlage erfolgt dokumentenbasiert. Demzufolge müssen für die Ausgestaltung die für die modellbasierte Abbildung relevanten Komponenten und Module einer Windenergieanlage sowie die Beziehungen zwischen diesen Elementen untereinander identifiziert und eine für deren Abbildung nutzbare Datenbank generiert werden. Die Anwendung sollte über eine webbasierte Datenbank inklusive grafischer Nutzerschnittstelle erfolgen, sodass der Zugriff für alle

Beteiligten möglich ist. Daher werden im Folgenden ein Vorgehen zur Generierung der Datenbank in Abschnitt 3.1.2.1 sowie die Analyse der Informationssystemarchitektur in Abschnitt 3.1.2.2 erläutert.

3.1.2.1 Vorgehen zur Generierung der Datenbank

Das Vorgehen zur Generierung der Datenbank für den agilen Entwicklungsprozess untergliedert sich in die drei Phasen Analyse, Konzeption und Detaillierung. Die darin enthaltenen vier Schritte sind in Abbildung 17 mit Tätigkeiten, Ergebnissen, Tooling sowie Verweis auf das dazugehörige Kapitel dieses Leitfadens zusammengefasst und werden im Folgenden erläutert.

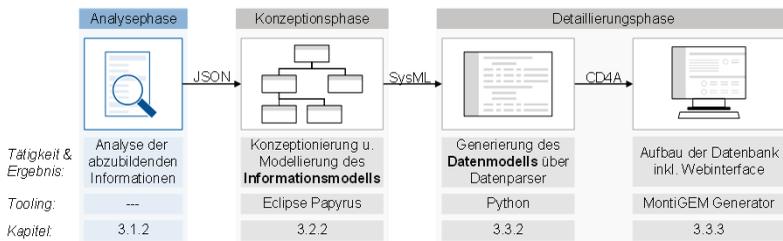


Abbildung 17: Vorgehen zur Generierung der Datenbank

Schritt 1: Analyse der abzubildenden Informationen: Die Erfassung der relevanten Daten und Informationen für den agilen Entwicklungsprozess einer Windenergieanlage stellt den initialen Schritt dar.

Schritt 2: Konzeptionierung und Modellierung des Informationsmodells: Auf Basis der in Schritt 1 identifizierten abzubildenden Informationen erfolgt die Konzeptionierung des Informationsmodells. Das Informationsmodell muss in der Lage sein, die identifizierten Informationen zur Entwicklung einer Windenergieanlage formalisiert abzubilden. Die Umsetzung des Informationsmodells erfolgt durch die Modellierung eines MBSE-Systemmodells mit der grafischen Systems Modelling Language (kurz: SysML). Der Fokus der Modellierung liegt dabei auf der Abbildung der Struktur einer Windenergieanlage. Zudem erfolgt die Modellierung der Wechselwirkungen der Komponenten sowie Module der Windenergieanlagenstruktur untereinander, sodass erforderliche Auslegungsparameter innerhalb des Entwicklungsprozesses zwischen den Simulationsmodellen übergeben werden können.

Schritt 3: Generierung des Datenmodells über Datenparser: Das in SysML modellierte Informationsmodell stellt die Datengrundlage zur Generierung der Datenbank dar. Die Erstellung der Datenbank erfolgt

mit Hilfe des MontiGem-Datenbankgenerators, der von dem Software Engineering Institut der RWTH Aachen University entwickelt wurde. Der MontiGem-Datenbankgenerator verarbeitet als Input das Datenformat Class Diagram for Analysis (kurz: CD4A). Daher besteht die Notwendigkeit die in SysML modellierten Daten des Informationsmodells in die Sprache CD4A zu transformieren. Für diese Aufgabe wurde ein Datenparser entwickelt.

Schritt 4: Aufbau der Datenbank inklusive Webinterface: Aus dem Datenmodell im Format CD4A entsteht in Schritt 4 die Datenbank. Der Zugriff auf die Datenbank erfolgt dabei durch ein eigens angepasstes Webinterface.

Innerhalb des vorliegenden Kapitels 3.1.2 wird der in Abbildung 17 blau hervorgehobene erste Schritt „Analyse der abzubildenden Informationen“ im Folgenden detaillierter betrachtet. Das Vorgehen zur Konzeptionierung und Modellierung des Informationsmodells mit der grafischen Modellierungssprache SysML als zweiter Schritt wird innerhalb von Kapitel 3.2.2 aufgezeigt. In der anschließenden Detaillierungsphase werden die in SysML modellierten Informationen des Systemmodells durch einen Datenparser in das Datenformat CD4A übersetzt und somit das Datenmodell generiert (siehe Kapitel 3.3.2). Das Datenmodell bildet wiederum die Grundlage zur Generierung der Datenbank über Anwendung des MontiGem-Generators (siehe Kapitel 3.3.3).

3.1.2.2 Analyse der bestehenden Datenstruktur und Identifikation der abzubildenden Informationen

Bevor die Erstellung des Systemmodells beginnen kann, muss initial analysiert werden, welche Informationen innerhalb des Entwicklungsprozesses einer Windenergieanlage abgebildet werden müssen. Abbildung 18 stellt im oberen Bereich einen Auszug der Baugruppen einer Windenergieanlage dar. Die übergeordneten Einheiten Rotor, Nacelle und Tower können dabei in weitere Submodule unterteilt werden. So enthält die Nacelle beispielsweise den mechanischen Triebstrang (englisch: Mechanical Drive Train), der wiederum das Getriebe (englisch: Gearbox) und den Generator enthält.

Angelehnt an die Produktstruktur einer Windenergieanlage, untergliedert sich die Entwicklung in so genannte Fields (deutsch: Fachgebiete). Für jedes dieser Fachbereiche wird innerhalb des Windows-Explorers ein Ordner erstellt, sodass sich die im unteren Bereich von Abbildung

18 auszugsweise dargestellte Ordnerstruktur ergibt. Bereits an den Ordner-Bezeichnungen dieses dateibasierten Ablagesystems ist erkennbar, dass Informationen aus unterschiedlichen fachlichen Domänen abgebildet werden müssen, woraus sich der Aufbau eines MBSE-Systemmodells motiviert. So interagieren unter anderem die fachlichen Domänen Mechanik, Elektrotechnik, Elektronik, Software und Aerodynamik innerhalb einer Windenergieanlage miteinander. Die in Abbildung 18 dargestellte Ordnerstruktur zeigt ebenfalls, dass neben den Komponenten und Modulen der Windenergieanlage auch die Betriebsbedingungen zu betrachten sind. Diese werden in dem Anwendungsbeispiel etwa durch Loads sowie Informationen des Typs WindClass beschrieben.

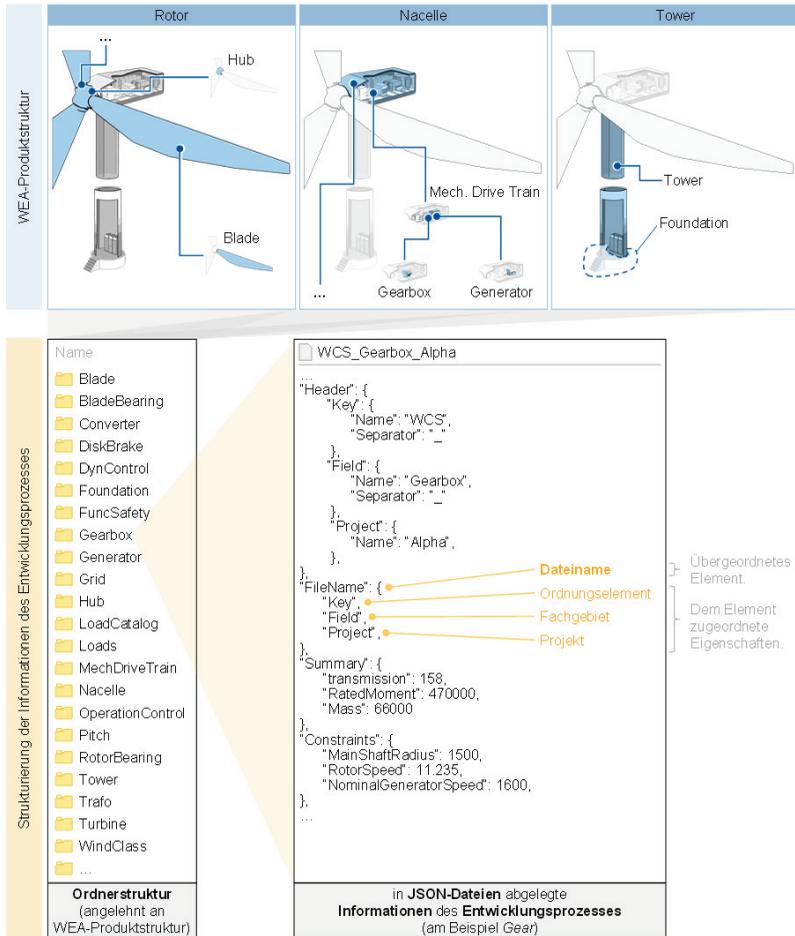


Abbildung 18: Auszug aus dem Aufbau einer Windenergieanlage⁹³

Welche Informationen die Fachgebiete enthalten und wie diese Informationen abgelegt sind, zeigt der untere rechte Bereich von Abbildung 18 für das Beispiel des Getriebes. So liegt innerhalb des Fachgebietsordners Gearbox die Java Script Object Notation (kurz: JSON) Datei WCS_Gearbox ab. JSON beschreibt ein kompaktes Datenformat in

⁹³ Axair Fans UK Limited 2020.

Textform, das sich aufgrund seiner Unabhängigkeit von Programmiersprachen für den Datenaustausch zwischen Anwendungen eignet⁹⁴. Abbildung 18 zeigt anhand eines Auszuges der JSON-Datei des Getriebes, in welchem Schema die Informationen abgelegt sind. Das Schema soll im Folgenden kurz anhand der standardisierten Definition der Dateinamen (FileName) erläutert werden. Die entsprechenden Informationen sind innerhalb von Abbildung 18 farblich hervorgehoben. Der FileName setzt sich aus den Elementen Key, Field und Project zusammen. Diese Elemente wurden zuvor innerhalb der JSON-Datei definiert. So enthält etwa das Field die Eigenschaften Name und Separator. Wird der FileName aus den Elemente Key, Field und Project zusammengefügt, ergibt sich WCS_Gearbox_Alpha als Benennung der JSON-Datei.

Durch die Analyse der bereitgestellten JSON-Dateien der Windenergieanlagenentwicklung, können die zu modellierenden Informationen in Kategorien eingeteilt werden. Tabelle 3-1 stellt die identifizierten Kategorien dar und erläutert diese jeweils. Mit Hilfe der durchgeführten Analyse der Entwicklungsdaten kann somit identifiziert werden, welche Informationen für den agilen Entwicklungsprozess einer Windenergieanlage abgebildet werden müssen. Auf dieser erarbeiteten Grundlage erfolgt im Anschluss die Konzeptionierung sowie Modellierung des Informationsmodells (siehe Kapitel 3.2.2).

Nachdem in der Analysephase die vorherrschenden Bedingungen herausgearbeitet worden sind, gilt es im folgenden Kapitel aufbauend auf den gesammelten Analysen das agile Konzept zu entwickeln.

⁹⁴ JSON. Zuletzt geprüft am 12.08.2022

Information	Beschreibung
Constraints	Fest definierte Abhängigkeiten zu anderen Fachgebieten.
ControlValue	Parameter, die während der Konfiguration einer Windenergieanlage zwischen den Fachabteilungen ausgetauscht werden.
FileName	Aufbau des Headers zur eindeutigen Dateibezeichnung.
Header	Informationen zu Beginn jeder JSON-Datei, um den Inhalt über einen Bezeichnungsschlüssel eindeutig definieren zu können.
Link	Eindeutige Verweise auf <i>ModelData</i> oder <i>ControlValues</i> innerhalb eines Fachgebietes.
ModelData	Modelldaten zur Ausführung spezifischer Simulationsumgebungen der Fachgebiete.
Summary	Zusammenfassung wesentlicher Eigenschaften des jeweiligen Fachgebietes.
...	...

Tabelle 3-1: Abzubildende Informationen des Entwicklungsprozesses einer Windenergieanlage

3.2 Konzeptionsphase

Die Konzeptionsphase dient der konzeptionellen Umsetzung einer agilen, datenbasierten Produktentwicklung. An dieser Stelle wird insbesondere die Konzeption des agilen Entwicklungsprozesses in Kapitel 3.2.1 zur Umsetzung der agilen Wirkprinzipien und der Konzeption des Informationsmodells in Kapitel 3.2.2 betrachtet.

3.2.1 Konzeption des agilen Entwicklungsprozesses

Die Konzeption des agilen Entwicklungsprozesses widmet sich zunächst der Übertragung der agilen Vorgehensweisen und Wirkprinzipien auf die Windenergieanlagenentwicklung in Kapitel 3.2.1.1. Grundlage für diese Übertragbarkeit ist die Existenz von wiederkehrenden Prozessbausteinen, die in Kapitel 3.2.1.2 anhand einer Definition und einer Methode zur Identifizierung von Prozessbausteinen auf Grundlage der Wertstromanalyse eingeführt werden. In Kapitel 3.2.1.3 wird der resultierende agile Entwicklungsprozess für die Windenergieanlagenentwicklung als formales Prozessmodell eingeführt.

3.2.1.1 Agile Vorgehensweise für die Windenergieanlagenentwicklung

Nachfolgend wird die Vorgehensweise für den agilen Entwicklungsprozess für Windenergieanlagen konzeptioniert. Ein Entwicklungsprozess

ist nach VDI 2221 ein „interdisziplinärer Unternehmensprozess zur Entwicklung eines marktfähigen Produkts, basierend auf der Definition initialer Ziele und Anforderungen an das Produkt, die im Lauf des Prozesses kontinuierlich weiterentwickelt und iterativ angepasst werden.“⁹⁵ Insgesamt wird im Rahmen dieses Leitfadens vorgestellt, wie sich agile Arbeitsweisen auf den Entwicklungsprozess für Windenergieanlagen übertragen lassen. Sowohl die Modelle für agile als auch für klassische plangetriebene Vorgehensweisen definieren jeweils phasenorientierte Prozesse, die sich aus zeitlichen Abschnitten zusammensetzen. Grundlage für diese Segmentierung des Entwicklungsprozesses ist das regelmäßige Review der bisherigen Ergebnisse. Die plangetriebene und agile Arbeitsweise unterscheidet sich durch die Interpretation des Inhalts dieser Reviews. Die plangetriebene Entwicklungsprozesssteuerung beruht auf einer sequenziellen Reihung der übergeordneten Entwicklungsphasen Planung, Konzeption, Entwicklung, Ausarbeitung und Validierung. Der Abschluss einer Phase kann nur durch die Erreichung eines definierten Reifegrads des Ergebnisses der jeweiligen Projektphase erreicht werden. Die agile Prozesssteuerung beruht hingegen auf der iterativen Bearbeitung und abschließenden Beurteilung inhaltlicher, funktionaler Entwicklungsobjekte. Die reifegradorientierte und objektorientierte Prozesssteuerung sind in Abbildung 19 gegenübergestellt.

⁹⁵ VDI 2221.

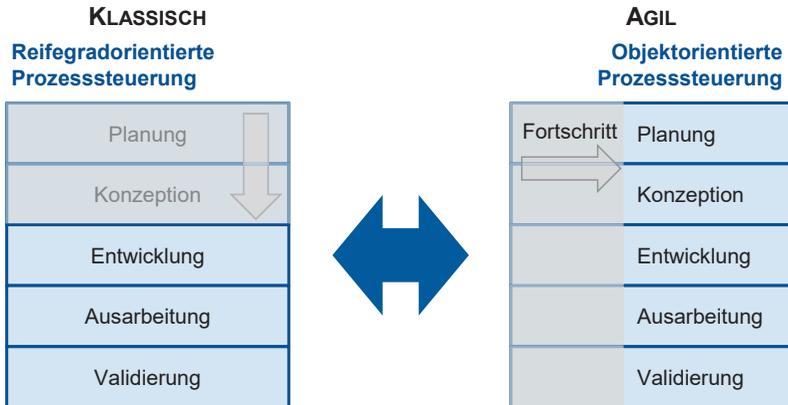


Abbildung 19: Schematische Gegenüberstellung der reifegradorientierten und objektorientierten Prozesssteuerung⁹⁶

Bei der Übertragung der agilen Produktentwicklung zugrundeliegenden Ansätze und Vorgehensweisen auf die Entwicklungsumgebung der Windenergiebranche wurde festgestellt, dass die Entwicklung einer Gesamtanlage aufgrund der Komplexität des Produkts zumeist nicht als homogenes Gesamtprojekt betrachtet werden kann. Bereits bei der klassischen Produktentwicklung ließen sich sowohl zeitlich sequenzielle als auch parallele heterogene Teilprojekte beobachten, die definierte Produktumfänge erzeugen. Diese Teilprojekte unterscheiden sich dabei sowohl hinsichtlich des Reifegrads des jeweiligen Projektergebnisses als auch hinsichtlich der benötigten Kompetenzen und Projektbesetzung.

Um der Produktkomplexität einer Windenergieanlage auch bei der agilen Entwicklung Rechnung zu tragen, wurde zunächst definiert, dass ein Gesamtentwicklungsprojekt aus zu Beginn des Projekts zu definierenden Entwicklungszyklen besteht. Die beispielhafte Dekomposition eines Entwicklungsprozesses in Entwicklungszyklen zeigt Abbildung 20. Grundlage und gleichzeitig Input jedes Produktentwicklungsprozesses sind dabei Anforderungen oder Fragestellungen⁹⁷. Ein Entwicklungszyklus behandelt die Bearbeitung einer inhaltlich gebündelten Teilmenge

⁹⁶ VDI 2221.

⁹⁷ Für ein genaueres Verständnis von Fragestellungen als Möglichkeit zur Kompensation von Unsicherheit im Entwicklungsprojekt und lösungsneutralen Anforderungsaufnahme sei an dieser Stelle auf Abschnitt 3.3.1.1 verwiesen

dieser Anforderungen und Fragestellungen. Innerhalb eines Entwicklungszyklus erarbeitet ein interdisziplinäres Team, welches kompetenzbasiert zusammengestellt wird, einen Minimum Viable Product (kurz MVP, beispielsweise Anlagenkonzept, Lastenmodell, Betriebsführung, virtueller Prototyp, Netzmodell). Im Kontext dieses Leitfadens wird ein MVP als minimales technisches Erzeugnis verstanden, welches unter Berücksichtigung des zur Realisierung notwendigen Zeit- und Ressourceneinsatzes die relevanten Eigenschaften abbildet, welche zur hinreichend genauen Beantwortung einer oder mehrerer Fragestellungen in Abhängigkeit ihrer Kritikalität benötigt werden. Ein MVP entsteht als Ergebnis eines Zyklus und besteht nicht nur aus einem Set von Features, sondern beinhaltet auch deren Umsetzung. Bei der initialen Definition von MVPs muss berücksichtigt werden, dass diese in ihrem Umfang einen brauchbaren Nutzen für den Kunden oder andere Stakeholder darstellen, der am Ende jedes Zyklus mit den jeweiligen Personen validiert werden kann.⁹⁸ Auf Basis dieser Validierung und dem Feedback der Stakeholder werden im Sinne der agilen Produktentwicklung Erkenntnisse für den weiteren Entwicklungsprozess gewonnen, die in spätere Zyklen einfließen können. Die Gesamtheit der entwickelten MVPs ergibt dann das abschließende Gesamtprojektergebnis.

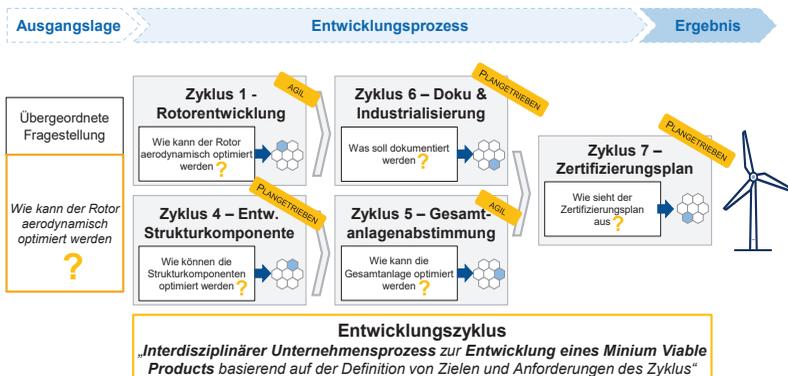


Abbildung 20: Dekomposition eines Entwicklungsprozesses in einzelne Entwicklungszyklen

⁹⁸ Maurya 2022.

Hinsichtlich der Durchführung der definierten Entwicklungszyklen kann wiederum zwischen einer agilen und plangetriebenen Herangehensweise unterschieden werden. In der Praxis bieten beide Herangehensweisen individuelle Vorteile, die für unterschiedliche Projekte oder Projektphasen im Einzelnen bewertet werden können, um eine Entscheidung hinsichtlich der zu bevorzugenden Herangehensweise herbeizuführen. Plangetriebene Prozesse zeichnen sich durch eine hohe Effizienz bei einer klaren Zielvorstellung und bekannten Lösungsansätzen aus. Ihnen fehlt jedoch bei höherer Volatilität oder Unsicherheit der Rahmenbedingungen die nötige Flexibilität. Dahingegen eignet sich die iterativ inkrementelle Vorgehensweise agiler Entwicklungsprozesse für Projekte mit höherem Innovationsgrad, einer unklaren Zielvorstellung und einem unbekanntem Lösungsweg mit hohem Abstimmungsbedarf aufgrund der Vielzahl der Schnittstellen. Die Übertragung dieser beiden Vorgehensweisen ist in Abbildung 21 dargestellt. Bei einem plangetriebenen Entwicklungszyklus ist das Entwicklungsziel klar definiert und die nötigen Entwicklungsschritte zur Erzeugung des definierten MVP stehen bereits zu Beginn fest. Im Gegensatz dazu muss das Entwicklungsziel bei der agilen Entwicklung noch nicht feststehen. Die Entwicklungsschritte werden kurzzyklisch, iterativ geplant. Das MVP wird dementsprechend iterativ in Sprints mit einem Zeithorizont von zwei bis vier Wochen entwickelt, die jeweils eine Teilmenge der initial definierten Fragestellungen abdecken. Innerhalb eines Sprints wird ein Produktinkrement (beispielsweise ein Schaltplan, 3D-Modell oder Geometrieentwurf) erzeugt, das intern validiert werden kann, um Erkenntnisse für die nachfolgenden Sprints abzuleiten. Die Gesamtmenge der erzeugten Produktinkremente ergibt am Ende des Entwicklungszyklus das MVP.

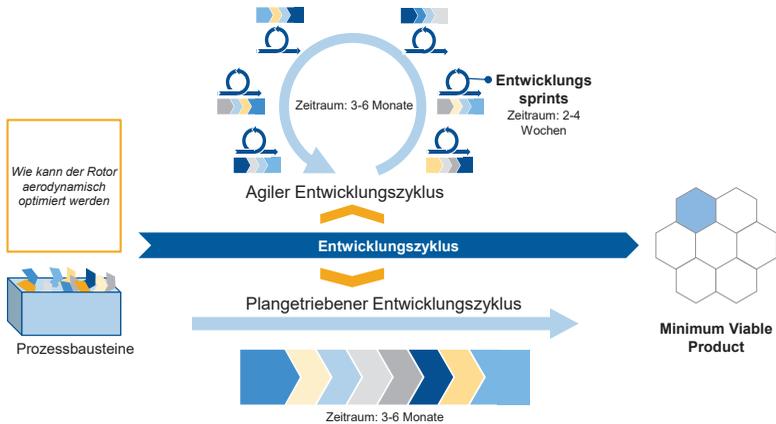


Abbildung 21: Gegenüberstellung einer plangetriebenen und agilen Durchführung eines Entwicklungszyklus

Die Entscheidung für ein agiles oder plangetriebenes Vorgehen beruht also darauf, welches Vorgehen für die aktuelle Aufgabe vorteilhafter ist. Eine Indikation hierzu gibt die sog. Stacey Matrix (vgl. Abbildung 22) in welcher die Anforderungsstabilität über der Technologiekenntnis aufgetragen wird. Je stabiler, d. h. bekannt und weitgehend unveränderlich die Anforderungen an das Entwicklungsergebnis sind, desto besser eignet es sich für die plangetriebene Entwicklung. Umgekehrt sind agile Ansätze für volatile Anforderungen aufgrund der vorgehaltenen Flexibilität besser geeignet. Ähnlich verhält es sich mit der Technologie: Sollen bekannte Technologien zum Einsatz kommen, wird oft plangetrieben entwickelt, sind unbekannte Technologien zu nutzen, empfiehlt sich das agile Vorgehen. Eine detailliertere Bewertung des Agilisierungspotenzials mithilfe von Agilitätsindikatoren wurde bereits in Kapitel 3.1.1.3 vorgestellt. Kapitel 3.3.1.3 geht detaillierter auf das Vorgehen bei der initialen, insbesondere aber bei der Neubewertung der Vorgehensweise sowie den Wechseln zwischen den Vorgehensweisen ein.

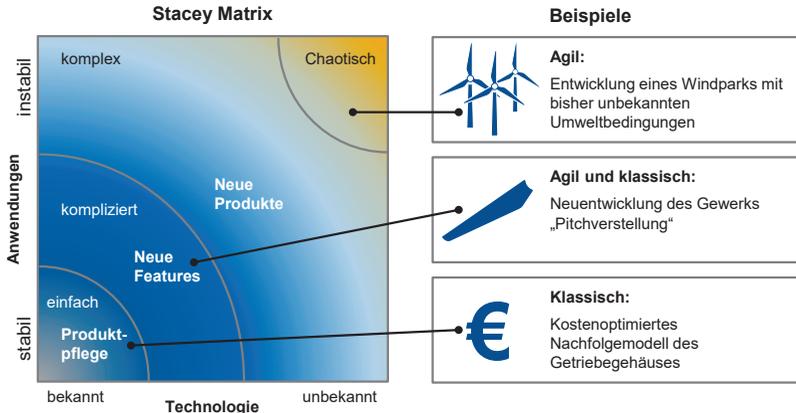


Abbildung 22: Stacey Matrix für die Entscheidung zwischen plangetriebenem und agilem Vorgehen für einen Entwicklungszyklus

Der vorliegende Leitfaden konzentriert sich auf die Darstellung der Durchführung der agilen Entwicklungszyklen. Der Fokus liegt dabei auf der Darstellung der Sprintdurchführung. Ein Sprint ist typischerweise zwei bis vier Wochen lang, wobei sich die spezifische Sprintlänge je Unternehmen, Projekt und sogar Projektphase unterscheiden kann. Der Ablauf eines Sprints folgt dabei den charakteristischen Schritten Planung, Entwicklung und Validierung, die über den gesamten Entwicklungszyklus iterativ durchlaufen werden. Ein solcher Sprint ist schematisch in Abbildung 23 dargestellt. Nachfolgend werden die drei Sprintphasen kurz erläutert:

Planung:⁹⁹ Im Rahmen eines Sprints wird eine Teilmenge der Gesamtmenge der Fragestellungen eines agilen Entwicklungszyklus beantwortet. Die Gesamtmenge der Fragestellungen und Anforderungen wird im Backlog gesammelt. Zu Beginn des Sprints erfolgt zunächst eine aktuelle Priorisierung der Fragestellungen sowie eine Auswahl der am

⁹⁹ Eine ausführliche Darstellung und methodische Erläuterung der Planungsphase erfolgt in Abschnitt 3.3.1.1 zur Konfiguration von Entwicklungssprints

höchsten priorisierten Fragstellungen. Anhand der ausgewählten Fragestellungen werden notwendige Entwicklungsaktivitäten¹⁰⁰ sowie je Aktivität ein Kriterium für den erfolgreichen Abschluss (engl. „Definition of Done“) definiert. Zudem wird je Entwicklungsaktivität der nötige Aufwand abgeschätzt.

Entwicklung: Im Anschluss an die Definition der notwendigen Entwicklungsaktivitäten erfolgt die eigentliche Entwicklung eines Produktinkrements. Da aufgrund der hohen Produktkomplexität einer Windenergieanlage die Bereitstellung eines nutzenstiftenden MVP für die Validierung mit allen Stakeholdern nach jedem Sprint technisch nicht möglich ist, erfolgt dessen Entwicklung inkrementell. Ein Produktinkrement kann dabei sowohl einen Teilumfang des MVP oder auch den gesamten MVP mit geringerer Reifegradentwicklung abbilden. Das Produktinkrement definiert sich durch die ausgewählten Fragestellungen und soll diese beantworten. Wie das Produktinkrement entwickelt wird, wird zu Beginn des Sprints nicht festgelegt. Stattdessen entwickelt das interdisziplinäre Team das Produktinkrement in enger Kooperation autonom in Selbstorganisation und stimmt sich während der Entwicklungszeit nicht bezüglich der Anforderungserfüllung mit anderen Stakeholdern ab. Die in der Planung definierten Entwicklungsaktivitäten werden nicht zugeteilt, sondern jede Person des Entwicklungsteams übernimmt eigenverantwortlich Aufgaben. Während der Entwicklung stimmt sich das Entwicklungsteam in regelmäßigen Zeitabständen (beispielsweise in Form eines Stand-Up-Meeting zu Beginn jedes dritten Arbeitstages) zum Fortschritt der Entwicklung ab.

Validierung: Nach der Entwicklung des Produktinkrements erfolgt die Validierung innerhalb des Entwicklungsteams. Die Validierung teilt sich in zwei Phasen, die Validierung des Ergebnisses sowie das Review des Vorgehens. Im Rahmen der Validierung des Ergebnisses wird das entwickelte Produktinkrement hinsichtlich der Beantwortung der ausgewählten Fragestellungen geprüft. Dabei können einerseits Fragestellungen als beantwortet abgeschlossen werden und andererseits auf Basis

¹⁰⁰ Für die Entwicklung komplexer Produkte wie beispielsweise der Entwicklung von Windenergieanlagen wird empfohlen, eine Liste möglicher Entwicklungsaktivitäten mit definierten Input und Output zur Verfügung zu stellen. Im Rahmen dieses Leitfadens werden diese vordefinierten Entwicklungsaktivitäten als Prozessbausteine definiert. Eine ausführliche Beschreibung der Prozessbausteine sowie den Methoden zur Identifizierung und Dokumentation von Prozessbausteinen ist in Abschnitt 3.2.1.2 dargestellt.

der gewonnenen Erkenntnisse neue Fragestellungen definiert oder bestehende Fragestellungen angepasst werden. Diese Fragestellungen werden in den Backlog überführt. Die Validierung des Produktinkrements kann intern im Entwicklungsteam ohne die jeweiligen Stakeholder durchgeführt werden. Andererseits erfolgt innerhalb des Entwicklungsteams ein Review des zurückliegenden Sprints, um die Vorgehensweise kontinuierlich zu verbessern. Dazu wird in einem Sprint-Review-Meeting offenes Feedback innerhalb des Entwicklungsteams ausgetauscht und Lessons Learned werden dokumentiert.

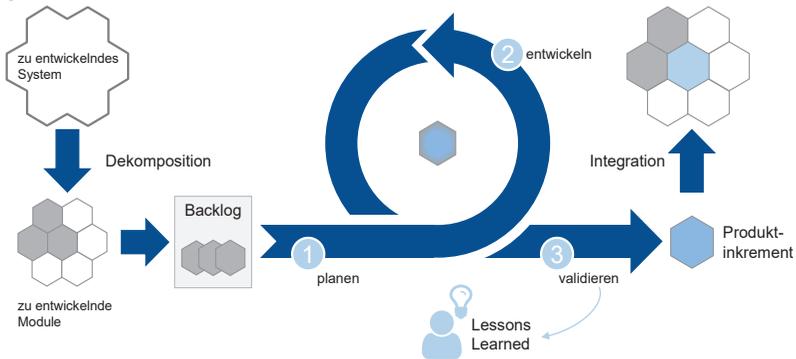


Abbildung 23: Ablauf eines Sprints

Zuvor wurde bereits dargestellt, dass insbesondere die hohe Produktkomplexität einer Windenergieanlage im Konflikt mit der agilen Vorgehensweise steht. Durch die Einführung von Entwicklungszyklen wurden in diesem Kontext bereits vorgestellt, wie Herausforderungen bei der Bereitstellung eines testbaren MVP adressiert werden können. Allerdings erschwert die hohe Produktkomplexität zusätzlich die Kooperation innerhalb der Entwicklungszyklen. Bei der Analyse der Entwicklungsprozesse wurde festgestellt, dass die Entwicklung sehr arbeitsteilig ausgestaltet ist und die Entwicklung der Module stark parallelisiert ist. Dieser Parallelisierung folgend soll auch die agile Produktentwicklung eine parallele Entwicklung differenzierter Module innerhalb eines Entwicklungszyklus ermöglichen. Dazu muss zunächst eine Fallunterscheidung vorgenommen werden. An dieser Stelle sollen zwei Zyklustypen unterschieden werden. Die beiden Typen unterscheiden sich hinsichtlich der Komplexität des zu entwickelnden Produktumfangs. So kann im Rahmen eines Entwicklungszyklus entweder die Entwicklung

eines komplexen Systems mit einer parallelisierten Entwicklung mehrerer Module und andererseits die Entwicklung eines einzelnen Moduls ohne Parallelisierung von mehreren Modulentwicklungen betrachtet werden. Diese Unterscheidung ist nachfolgend in Abbildung 24 dargestellt.

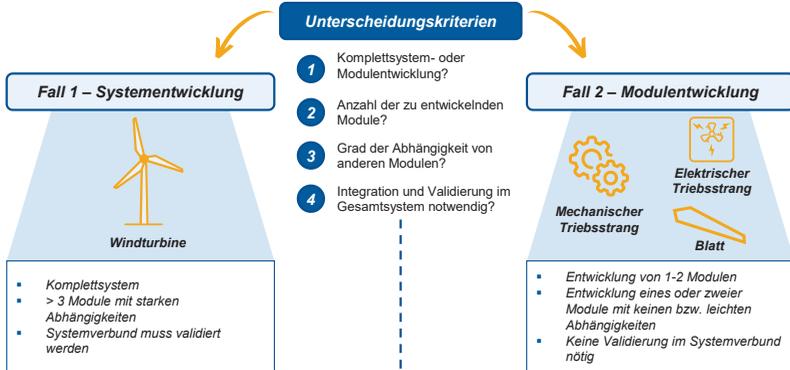


Abbildung 24: Fallunterscheidung bei der Durchführung der Entwicklungszyklen

An dieser Stelle soll zunächst auf die Systementwicklung eingegangen werden. Als System kann an dieser Stelle nicht nur eine gesamte Windenergieanlage, sondern insbesondere auch ein Teil einer Anlage verstanden werden, der aufgrund seiner Komplexität wiederum in mehrere Module unterteilt werden muss. Ein Modul kann projektindividuell definiert werden, jedoch sollte es klar abgegrenzt sein und einen eigenständig entwickelbaren Umfang besitzen. So können beispielsweise das Blatt, das Pitchsystem, der elektrische oder der mechanische Triebstrang vorstellbare Module sein. Sofern vorhanden kann eine Dekomposition des Systems in Anlehnung an die generische Produktstruktur der Windenergieanlage erfolgen.¹⁰¹ Eine mögliche Dekomposition ist in Abbildung 25 dargestellt.

¹⁰¹ Schuh 2022.

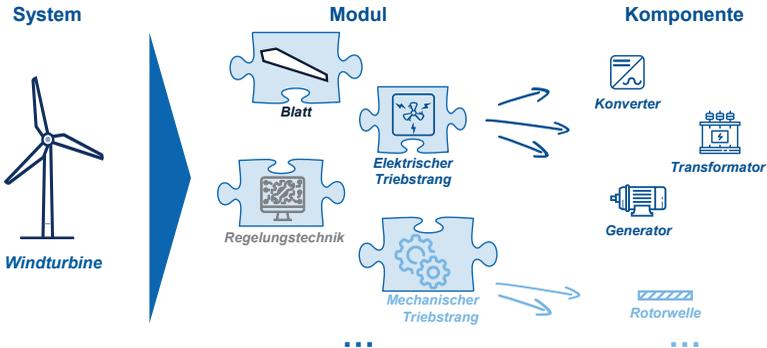


Abbildung 25: Dekomposition eines Systems "Windturbine" in mögliche Module

Diese Unterteilung wird genutzt, um Entwicklungsumfänge aufzuteilen, die Größe des an dem einzelnen Sprint beteiligten Team zu begrenzen und Abstimmungsaufwände während des Sprints zu reduzieren. Die Abfolge eines Sprints in der Systementwicklung ist in Abbildung 26 dargestellt.

Jeder Sprint beginnt dabei mit einer Planung auf Systemebene. Kundenseitige Fragestellungen und Anforderungen richten sich in diesem Prozessmodell stets an das zu entwickelnde System und werden in der Planung priorisiert. Im Anschluss erfolgt die Übergabe der für den Sprint priorisierten Fragestellungen von der Systemebene an die Modulebene und die einzelnen Entwicklungsteams, die jeweils ein Modul betreuen. Jedes Team leitet aus den priorisierten Fragestellungen auf Systemebene Fragestellungen auf Modulebene ab und überführt diese in den jeweiligen Modul Backlog. Auf Basis des Modul Backlogs erfolgt die eigenverantwortliche Entwicklung innerhalb der Teams je Modul, indem Teilumfänge der Module entsprechend der priorisierten Modulfragestellungen inkrementell weiterentwickelt werden. Hilfsmittel bei der strukturierten Auswahl von Entwicklungsaktivitäten zur Erzeugung gewünschter Entwicklungsergebnisse können Prozessbausteine (PBS) sein, die in Abschnitt 3.2.1.2 eingeführt werden. Abschließend werden die Ergebnisse der Modulentwicklungen auf Modulebene validiert, um beispielsweise für kommende Sprints modulindividuelle Verbesserungspotenziale ableiten und an die Systemebene übergeben zu können. Auf der Systemebene erfolgt eine Integration der Ergebnisse aller Modulsprints und eine Validierung zur Prüfung der Beantwortung der priorisierten

Systemfragestellungen. Auf Basis dieser Validierung des kombinierten Inkrements auf Systemebene werden Erkenntnisse und neue Fragestellungen für kommende Sprints auf Systemebene abgeleitet.

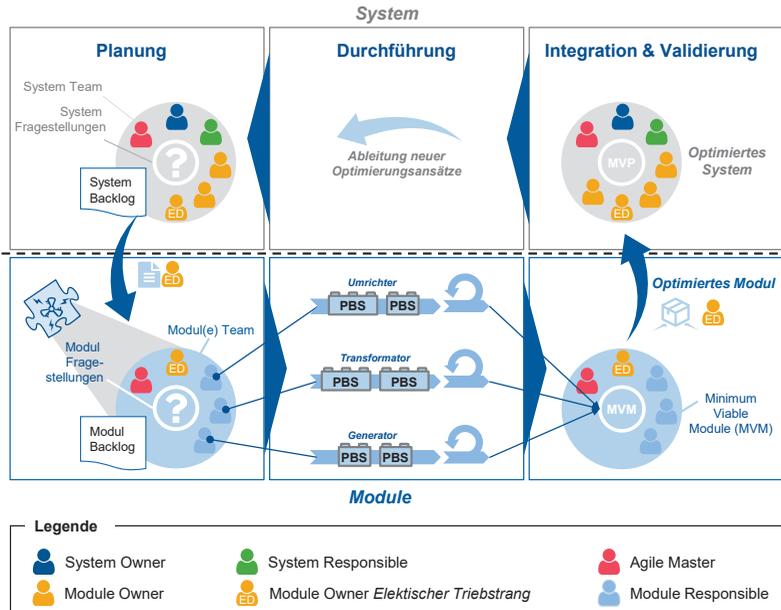


Abbildung 26: Schematische Darstellung eines Sprints der Systementwicklung¹⁰²

Analog zu der schematischen Darstellung der Systementwicklung zeigt Abbildung 27 den Ablauf der Modulentwicklung als Sonderfall der Systementwicklung. In dem Fall der Modulentwicklung findet keine Dekomposition des zu entwickelnden Umfangs statt. Stattdessen ist die Modulentwicklung eng an die oben beschriebene Sprintdurchführung mit den drei Phasen Planung, Entwicklung und Validierung angelehnt. Die Formulierung der Fragestellungen adressiert bei der Modulentwicklung direkt den gesamten Entwicklungsumfang des jeweiligen Entwicklungszyklus. Anschließend folgt die eigentliche Entwicklung und Validierung des Modulinkrements. Entgegen der Benennung bezieht sich die Bezeichnung „Modulentwicklung“ insbesondere auf das Prozessmodell als

¹⁰² Die dargestellten Rollen sind an dieser Stelle beispielhaft zu verstehen und werden in Abschnitt 3.4.1.1 detailliert beschrieben.

Ausschnitt aus der Systementwicklung. Das Prozessmodell der Modulentwicklung kann auch für größere Entwicklungsumfänge wie beispielsweise die frühe Konzeption der Hauptmaße einer Windenergieanlage eingesetzt werden, die ebenfalls in einem kleinen Team erledigt werden können.

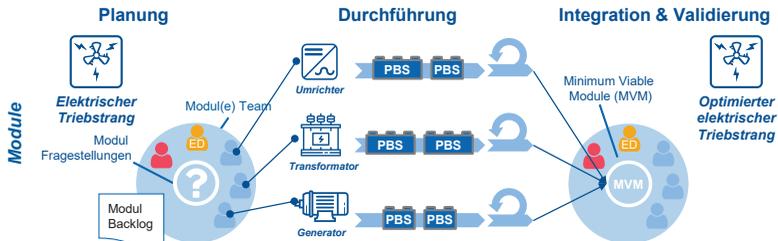


Abbildung 27: Schematische Darstellung eines Sprints der Modulentwicklung

Nachdem die grundsätzliche Vorgehensweise in der agilen Entwicklung vorgestellt ist, wird im Weiteren auf die bereits erwähnten Prozessbausteine eingegangen.

3.2.1.2 Definition von Prozessbausteinen

Um Unternehmen des Maschinen- und Anlagenbaus zur zielgerichteten Konfiguration von einzelnen Entwicklungssprints zu befähigen, wird an dieser Stelle das Konzept der Prozessbausteine eingeführt. Durch die Einführung von Prozessbausteinen wird der übergeordnete Prozess des Sprints in einfach handhabbare Teile zerlegt um die initiale Konfiguration zu erleichtern. Ein Prozessbaustein ist ein abgeschlossener Prozessabschnitt, der von einer Einzelperson oder einem Team in einem abgeschlossenen Zeitraum durchgeführt werden kann. Ein Prozessbaustein wird definiert durch die durchzuführende Aufgabe sowie durch die Input-Output-Beziehung. Die Definition von Prozessbausteinen erfolgt ergebnisorientiert und bildet die Grundlage aus der ein Sprint konfiguriert werden kann.

Die Definition einer Gesamtmenge verfügbarer Prozessbausteine ist dabei Grundlage für die Konfiguration eines Produktentwicklungssprints, sodass die Erzeugung eines gewünschten Outputs sichergestellt werden kann. Prozessbausteine sind in diesem Kontext für wiederkehrende Handlungsmuster von unterschiedlichen Entwicklungsprozessen oder innerhalb eines Entwicklungsprozesses definiert. Eine

Grundmenge aller Prozessbausteine wird anhand der in Abschnitt 3.1.1.4 beschriebenen Wertstromanalyse identifiziert. Diese Identifizierung ist in Abbildung 28 anhand einer beispielhaften Wertstromanalyse dargestellt. Die Identifizierung der Prozessbausteine ist dabei in hohem Maße unternehmensindividuell und muss spezifisch für jedes Unternehmen durchgeführt werden. Dabei sollten insbesondere jene Ketten von Aktivitäten als Prozessbaustein identifiziert werden, die ein für sich stehendes, relevantes Entwicklungsartefakt erzeugen. Eine Aktivitätenkette kann nicht als zusammengehörender Prozessbaustein identifiziert werden, wenn Zwischenprodukte von anderen Prozessbausteinen als Input benötigt werden oder für sich eine Relevanz für den Entwicklungsprozess besitzen. In diesem Fall sollte die Aktivitätenkette in mehrere Prozessbausteine aufgeteilt werden. Ebenso sollte für einen Prozessbaustein keine 1 zu 1 Beziehung mit einem anderen Prozessbaustein existieren. Eine 1 zu 1 Beziehung entsteht, wenn ein erzeugtes Artefakt eines Prozessbausteins lediglich der spezifische Input für einen anderen Prozessbaustein darstellt. Liegt eine 1 zu 1 Beziehung zwischen Prozessbausteinen vor können diese zusammengeführt werden. Die auf diese Weise initial definierte Liste aller Prozessbausteine kann dabei nicht als final angesehen werden, da die aufgenommenen Wertstromanalysen nicht den Anspruch besitzen, alle potenziellen Entwicklungssituationen abbilden zu können. Aus diesem Grund kann und muss die Gesamtliste der Prozessbausteine innerhalb der Entwicklungssprints um neue Prozessbausteine stetig angepasst und erweitert werden, sodass diese für die Konfiguration künftiger Entwicklungssprints zur Verfügung stehen.

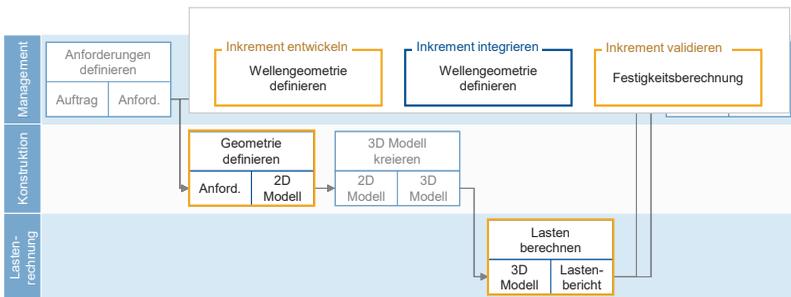


Abbildung 28: Identifizierung von Prozessbausteinen in einer Wertstromanalyse

Die Prozessbausteine können entsprechend ihrer im Entwicklungsprozess übergeordneten Funktion typisiert werden. Im Rahmen des Forschungsvorhabens wurden die Typen „Inkrement entwickeln“, „Inkrement integrieren“ und „Inkrement validieren“ identifiziert. Ein Prozessbaustein des Typs „**Inkrement entwickeln**“ leistet dabei einen tatsächlichen Wertschöpfungsbeitrag, indem Unsicherheit hinsichtlich der Ausgestaltung des Endprodukts aktiv beseitigt wird. Ein typischer Prozessbaustein ist beispielweise die Berechnung einer Leistungsgröße oder die Konstruktion eines Bauteils. „**Inkrement integrieren**“ bezeichnet Prozessbausteine, die keinen zusätzlichen Informationsgehalt zur Produktentwicklung generieren, sondern ein definiertes Format erzeugen oder Ergebnisse aggregieren und auf diese Weise bereitstellen. Typische Prozessbausteine dieses Typs sind das Ableiten einer Zeichnung oder das Zusammenführen einer Baugruppe. Abschließend dienen Prozessbausteine des Typs „**Inkrement validieren**“ dazu, das Entwicklungsergebnis mit einem gewünschten Zustand abzugleichen. Mögliche Prozessbausteine sind die Durchführung einer Simulation oder eine Kollisionsprüfung des Systems.

Zudem können Prozessbausteine dahingehend differenziert werden, ob der Objektbereich der Anwendung die in Abschnitt 3.2.1.1 eingeführte Modul- oder Systemebene adressiert. So können insbesondere für den Fall der Systementwicklung die Prozessbausteine „Modulinkrement entwickeln“, „Modulinkrement integrieren“, „Modulinkrement validieren“, „Systeminkrement integrieren“ und „Systeminkrement validieren“ unterschieden werden. Eine Entwicklungstätigkeit auf Systemebene findet typischerweise nicht statt. Stattdessen dient die Systemebene der Zusammenführung der Modulinkremente und der übergeordneten Validierung des vollständigen Systeminkrements.

Zur Beschreibung und Dokumentation eines Prozessbausteins werden an dieser Stelle zunächst vier Beschreibungsdimensionen mit insgesamt sechs beschreibenden Merkmalen vorgeschlagen, die in Abbildung 29 beispielhaft dargestellt sind. Die vier Beschreibungsdimensionen sind Transformationstätigkeit, Transformationseingang, Transformationsmittel und Transformationsergebnis. Die **Transformationstätigkeit** mit dem Merkmal Handlung kann dabei als Titel und zur Identifizierung des Prozessbausteins genutzt werden, da hier die eigentliche Entwicklungshandlung (bspw. 3D-Modell ableiten) dokumentiert wird. In

der Dimension **Transformationseingang** wird der Prozessbaustein anhand eingehenden Artefakten beschrieben. Dabei definiert das Merkmal Input, welche Artefakte für die Durchführung der Entwicklungstätigkeit notwendig sind. Das Merkmal Information nimmt Artefakte auf, die die Ausführung des Prozessbausteins optional unterstützen können, allerdings für die Durchführung nicht zwingend notwendig sind. Unter **Transformationsmittel** werden die Merkmale Hilfsmittel und Kompetenzen zusammengefasst. Unter Hilfsmittel werden Systeme, Werkzeuge oder andere Ressourcen (bspw. CAD-Software) definiert, die für die Durchführung des Prozessbausteins genutzt werden. Kompetenzen sind jene Fähigkeiten, die Mitarbeitende für die Ausführung des Prozessbausteins (bspw. CAD-Kompetenz) benötigen. Abschließend wird im **Transformationsergebnis** definiert, welcher Output als Ergebnis bei der Durchführung des Prozessbausteins (bspw. 3D-Modell) erwartet wird.

Verwendungsziel

 **Fragestellung:** Erzwingt die Durchführung eines Prozessbausteins

Transformationsergebnis

 **Output:** Erwartetes Ergebnis der Transformationstätigkeit zur Beantwortung der Fragestellung

Transformationseingang

 **Input:** Notwendige „Betriebsstoffe“ zur Durchführung der Transformationstätigkeit

 **Informationen:** Optionale/kontextbezogene „Betriebsstoffe“ zur Durchführung der Transformationstätigkeit

Transformationsmittel

 **Kompetenzen:** Notwendige Personalfähigkeiten zur Durchführung der Transformationstätigkeit

 **Hilfsmittel:** Notwendige Systeme/Werkzeuge zur Durchführung der Transformationstätigkeit

Transformationstätigkeit

 **Handlung**

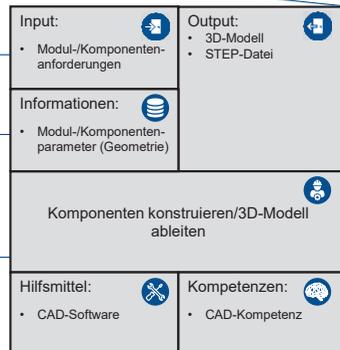


Abbildung 29: Beschreibung und Dokumentation eines Prozessbausteins

An dieser Stelle wird darauf hingewiesen, dass bei der Formulierung von Prozessbausteinen eine Dichotomie zwischen Generizität und Spezifität vorherrscht. Diese Dichotomie resultiert daher, dass Prozessbausteine vornehmlich genutzt werden sollen, um Entwicklungssprints fragestellungs-spezifisch zu konfigurieren.

Vor dem Hintergrund dieser Anforderungen sollten Prozessbaustein möglichst spezifisch dokumentiert werden und idealerweise fragestellungs- und anwendungsbezogen sein, um eine eindeutige Auswahl zu ermöglichen. Allerdings mündet diese Spezifität der Prozessbausteine in einer hohen Anzahl und würde einen komplexen Auswahlprozess zur Folge haben.

Um den Auswahlprozess zu vereinfachen, bietet sich eine möglichst generische Formulierung der Prozessbausteine an, um deren Anzahl zu minimieren und eine schnelle Konfiguration innerhalb des Sprints zu ermöglichen. Allerdings wären die resultierenden Prozessbausteine zu einem hohen Maß mehrdeutig. Dementsprechend ergibt sich die Dichotomie zwischen der Generizität und Spezifität der Prozessbausteine, die aufzulösen ist.

An dieser Stelle wird die Empfehlung ausgegeben, die Dichotomie aufzulösen, indem, wie in Abbildung 30 dargestellt, Prozessbausteine vom Verwendungsziel und Anwendungskontext entkoppelt werden. Das Verwendungsziel eines Prozessbausteins ist die zu beantwortende Fragestellung, welche den Lösungsraum des Prozessbausteins hinsichtlich des formulierten Ziels einengt. Der Anwendungskontext eines Prozessbausteines ergibt sich aus dem betrachteten Produktumfang, in dessen Kontext der Prozessbaustein eingesetzt wird. Durch die Entkopplung der Prozessbausteine von diesen beiden Dimensionen kann die Vielfalt der zu dokumentierenden Prozessbausteine effektiv eingeschränkt werden. Eine Spezifizierung der Prozessbausteine findet dann durch den Anwendungskontext der Prozessbausteine in der Praxis statt. Allerdings muss an dieser Stelle erwähnt werden, dass die Auflösung dieser Dichotomie nicht bedeutet, dass keine spezifischen Prozessbausteine formuliert werden dürfen. So gilt es verwendungsziel- und anwendungsspezifische Prozessbausteine zu formulieren, wenn diese nur in einem definierten Kontext genutzt werden.

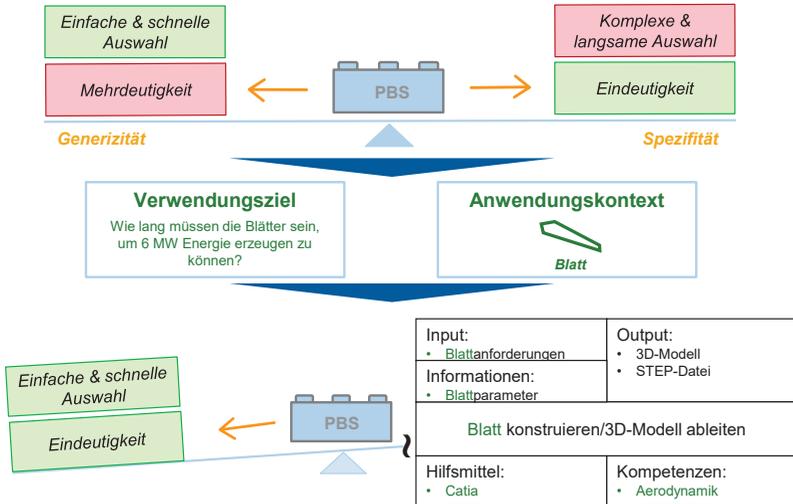


Abbildung 30: Auflösung der Dichotomie zwischen Generizität und Spezifität der Prozessbausteine

Abschließend lässt sich sagen, dass die Definition geeigneter Prozessbausteine eine verbesserte Strukturierung von Sprints und eine verbesserte Transparenz des Entwicklungsprozesses ermöglicht.

3.2.1.3 Agiler Entwicklungsprozess für die Windenergieanlagenentwicklung

Nachfolgend wird die Ablauforganisation des agilen Entwicklungsvorgehens formal dargestellt und erläutert. Zu diesem Zweck wird eine Modellierungssprache zur Prozessmodellierung in Anlehnung an DIN 66001 verwendet. Eine Legende zur verwendeten Prozessmodellierungssprache ist in Abbildung 31 dargestellt.

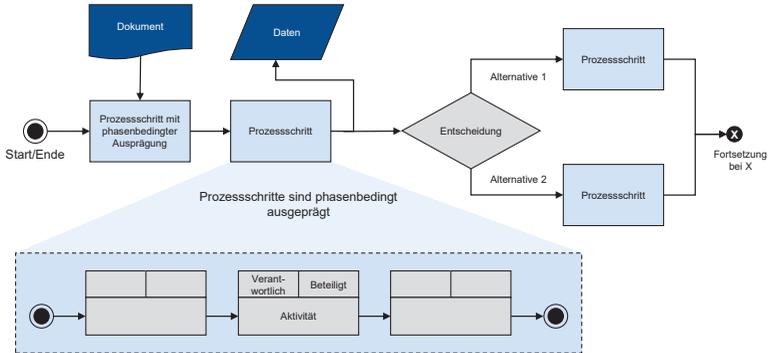


Abbildung 31: Legende zur Prozessmodellierungssprache in Anlehnung an DIN 66001

Im Folgenden soll zunächst das Prozessrahmenmodell eingeführt werden. Als Prozessrahmen werden an dieser Stelle die Initiierung und der Abschluss eines Projekts definiert, die den eigentlichen Entwicklungsprozess steuern und organisieren. Aus diesem Grund wurde in Abbildung 32 eine Projekt- und Produktebene eingeführt, welche die projektbezogenen Organisationstätigkeiten und die produktbezogenen Entwicklungstätigkeiten trennt. Auf der Projektebene wurden fünf Prozessschritte definiert, die an dieser Stelle kurz erläutert werden. Eine Darstellung der Prozessschritte als Abfolge von Aktivitäten und den entstehenden Artefakten je Aktivität ist im Anhang A dargestellt.

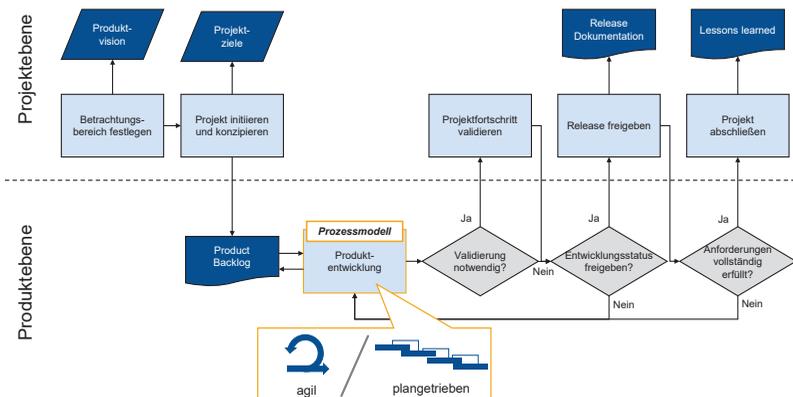


Abbildung 32: Prozessrahmenmodell

Der Prozessschritt „**Betrachtungsbereich festlegen**“ hat das Ziel die Rahmenbedingungen des Projekts zu klären und eine Vision für das zu entwickelnde Produkt festzulegen. Dazu gehört unter anderem die Analyse der Ausgangssituation, die Identifizierung geeigneter Marktsegmente und einer potenziellen Marktlücke hinsichtlich der Randbedingungen, Lösungen, Preise und Technologien sowie die Ableitung einer Produktvision und Absatzprognose. Die Produktvision soll als übergeordnetes Zielbild fungieren und den angestrebten Nutzen für die identifizierte Marktlücke beschreiben.

Im darauffolgenden Prozessschritt „**Projekt initiieren und konzipieren**“ wird die definierte Produktvision genutzt, um die eigentliche Produktentwicklung vorzubereiten und alle dafür notwendigen Informationen zu aggregieren. Dazu gehört die Formulierung und Priorisierung von User Stories, eine Kosten- und Absatzprognose, die Formulierung der Projektziele, der Aufbau einer Funktionsstruktur und die Festlegung des technischen und personellen Projektrahmens durch Produktarchitektur und -entwurf sowie der Teamzusammensetzung. Abschließend wird der initiale Product Backlog als Sammlung der ersten Fragestellungen und Anforderungen formuliert und ein Releaseplan in Form zu entwickelnder MVPs definiert.

An die Konzeption und Initiierung des Projekts anschließend folgt der Prozessschritt der **Produktentwicklung** auf der Produktebene. Diese soll nachfolgend im Detail erläutert werden.

Nach der Produktentwicklung wird, sofern eine Validierung notwendig ist, im nächsten Prozessschritt, dem „**Projektfortschritt validieren**“, geprüft, inwiefern der aktuelle Stand der Produktentwicklung den definierten Projektzielen genügt. Dabei geschieht ein Abgleich des aktuellen Entwicklungsstatus mit dem initial definierten Release-Plan, um die Zielerreichung zu validieren. Ergebnis dieser Validierung kann einerseits die weitere Produktentwicklung oder andererseits die Freigabe des aktuellen Release für den Kunden sein. Bei der Fortsetzung der Produktentwicklung sollte zudem geprüft werden, inwiefern die Adaption des geplanten Entwicklungsvorhabens hinsichtlich der Ziele, der Teamzusammensetzung oder dem Release Plan notwendig ist.

Sollte im vorigen Prozessschritt die Validierung positiv ausfallen und die Entscheidung getroffen werden, den Entwicklungsstatus freizugeben, folgt der Prozessschritt „**Release freigeben**“. Dieser Schritt umfasst ne-

ben der Erstellung der prüfungsrelevanten Dokumente und der gegebenenfalls anfallenden Zertifizierung die Erzeugung einer finalen Stückliste sowie Finalisierung der Produktdokumentation und Freigabe zur Fertigung.

Sofern alle Anforderungen durch diese Dokumentation erfüllt werden, folgt der finale Prozessschritt „**Projekt abschließen**“ und im Sinne der agilen Vorgehensweise eine kritische Reflexion des Projektvorgehens. Insbesondere sollen dabei mögliche Verbesserungen anhand des Projektverlaufs identifiziert und in dokumentierten Lessons Learned verfügbar gemacht werden, bevor das Projekt final abgeschlossen wird.

An dieser Stelle soll ein formales Prozessmodell für die **Produktentwicklung** vorgestellt werden, welches auf der in Abschnitt 3.2.1.1 beschriebenen Entwicklungsiteration auf System- und Modulebene aufbaut und in Abbildung 33 dargestellt ist. Diese Entwicklungsiteration und die darin enthaltenen Prozessschritte werden nachfolgend erläutert. Eine detaillierte Erläuterung der Prozessschritte findet sich in im Anhang A.

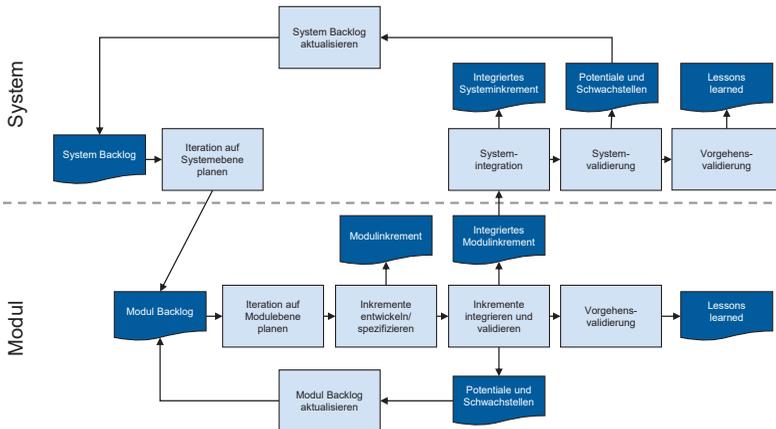


Abbildung 33: Formales Prozessmodell einer Entwicklungsiteration mit System- und Modulebene

Die Entwicklungsiteration beginnt typischerweise mit dem Schritt „**Iteration auf Systemebene planen**“, bei dem anhand der Gesamtmenge aller noch nicht erfüllten Anforderungen und noch nicht beantworteten Fragestellungen im System Backlog jene Elemente priorisiert werden, die im nachfolgenden Sprint adressiert werden sollen. Zudem sollten je

explizite Entwicklungsergebnisse definiert sowie ein dazugehöriges Abnahmekriterium definiert werden, anhand dessen das erzeugte Inkrement am Sprintende bewertet werden kann. Zudem muss anhand der priorisierten Fragestellungen geprüft werden, wie die zu entwickelnden Module von diesen Fragestellungen betroffen sind und zu deren Beantwortung beitragen können, um entsprechende Modulfragestellungen ableiten zu können. Modulfragestellungen sind somit die Fragen, die jedes Modul für sich beantworten muss, um gesamtheitlich die priorisierte Fragestellung zu beantworten. Für die Modulfragestellungen werden ebenfalls zu entwickelnde Teilergebnisse definiert und ein Umsetzungsaufwand je Teilergebnis abgeschätzt. Die Abschätzung des Umsetzungsaufwands ist insbesondere für die Arbeit im Multiprojektmanagement eine hohe Relevanz, da nicht nur geprüft wird, inwiefern der einzelne Entwicklungssprint durch das Team zeitlich erledigt werden kann, sondern auch Konflikte hinsichtlich der zeitlichen Auslastung einzelner Personen zwischen Projekten identifiziert und aufgelöst werden können, indem die Sprintplanung angepasst wird. Sofern keine Konflikte bestehen, werden die abgeleiteten Modulfragestellungen in die jeweiligen Modul Backlog überführt.

Im folgenden Schritt **„Iteration auf Modulebene planen“** werden zunächst die Aktivitäten analog zur Systemebene für die Definition von Erfüllungskriterien und die Identifizierung betroffener und damit anzupassender Sub-Module durchgeführt. Allerdings erfolgt nun die Konfiguration des Sprints, indem für die Erzeugung des Teilergebnisses notwendige Prozessbausteine ausgewählt und hinsichtlich des Umsetzungsaufwands abgeschätzt werden, um zeitliche Planung des Sprints zu ermöglichen. An dieser Stelle sei angemerkt, dass innerhalb des Modulsprints nicht ausschließlich die initial priorisierten Fragestellungen behandelt werden müssen, sondern auch moduleigene Fragestellungen oder nicht beantwortete Fragestellung aus vorigen Sprints relevant sein können.

Im Anschluss an die Planung auf Modulebene folgt die Modulentwicklung durch die Schritte **„Inkrement entwickeln“**, **„Inkrement integrieren“** und **„Inkrement validieren“**, welche durch die ausgewählten Prozessbausteine ausgefüllt werden. Ergebnis dieser Schritte ist das geprüfte Modulinkrement, das in der **„Systemintegration“** mit den anderen entwickelten Modulinkrementen zusammengeführt wird und die

„**Systemvalidierung**“ durchläuft. Bei der Validierung sowohl des Modulinkrements und insbesondere des Systeminkrements werden Schwachstellen und Potenziale aufgedeckt, die in Modul und System Backlog durch die Prozessschritte „**Modul Backlog aktualisieren**“ und „**System Backlog aktualisieren**“ überführt werden. Abschließend erfolgt in jedem Sprint eine „**Vorgehensvalidierung**“ auf Modul- und Systemebene, um den beteiligten Personen die Möglichkeit einzuräumen, prozessuale Verbesserungspotenziale kurzzyklisch einfließen zu lassen. Eine ausführliche formale Dokumentation dieser Prozessschritte findet sich im Anhang A.

Abschließend ist zudem ein Sprint in der Modulentwicklung als Sonderfall der Systementwicklung in Abbildung 34 dargestellt. Die Schritte sind dabei analog zu den Schritten der Systementwicklung, wobei eine Dekomposition der Fragestellung und eine Integration der Inkremente für die reine Modulentwicklung nicht notwendig ist.

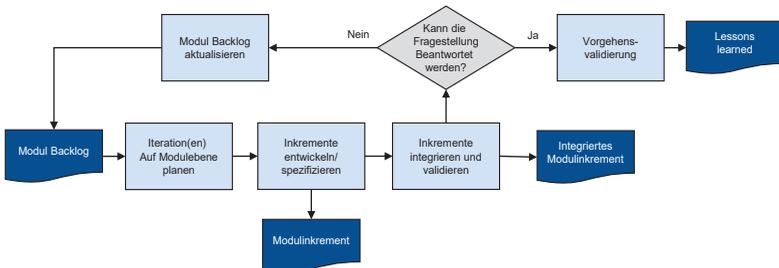


Abbildung 34: Formales Prozessmodell einer Entwicklungsiteration mit Modulebene

Mit der Detaillierung der Modulebene als letzter Teil des vollständigen Prozessmodells ist die Konzeption für die Übertragung der agilen Vorgehensweisen auf die Windenergieanlagenentwicklung vollständig. Zusätzlich zur Definition der Entwicklungsprozesse wurde in der Konzeptionsphase das im Kapitel 3.2.2 dargestellte Informationsmodell entwickelt, um das definierte Prozessmodell zu befähigen.

3.2.2 Konzeption Informationsmodell

In Kapitel 3.1.2 konnten innerhalb der Analysephase die abzubildenden Informationen zur Beschreibung einer Windenergieanlage identifiziert werden. Auf dieser Grundlage kann das Konzept des Informationsmodells erstellt und anschließend modelliert werden (siehe Abbildung 35). Dieses Modell stellt anschließend die Grundlage für die Datenbank dar, auf welche von Nutzern über Schnittstellen zugegriffen werden kann.

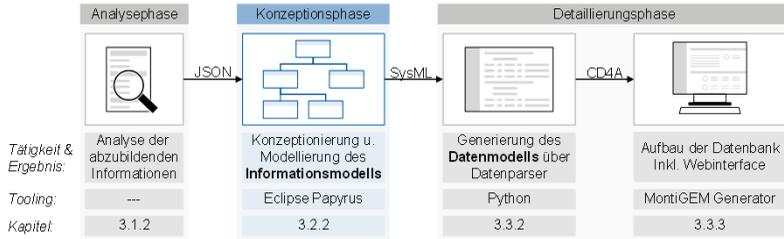


Abbildung 35: Vorgehen zur Generierung der Datenbank

Während der Analyse der abzubildenden Informationen (siehe Kapitel 3.1.2) konnte gezeigt werden, dass vielfältige technische Domänen an dem Entwicklungsprozess einer Windenergieanlage beteiligt sind. Wie die Vielzahl moderner mechatronischer Produkte, verwendet die Windenergiebranche eine Vielzahl disziplinspezifischer Modelle, die jeweils eine individuelle Sichtweise auf das Produkt darstellen¹⁰³. Einen Ansatz, um sämtliche Sichtweisen sowie ihre jeweiligen Modelle innerhalb der Produktentwicklung zu verbinden, stellt wie bereits beschrieben das Model-Based-Systems Engineering dar. Durch die Schaffung eines interdisziplinären Systemmodells werden im MBSE sämtliche Modelle und Informationen des Entwicklungsprozesses zentral verwaltet und funktional miteinander verknüpft¹⁰⁴. Aufgrund dieser Eigenschaften stellt die Überführung der Informationen aus den JSON-Dateien in ein MBSE-Systemmodell, den Lösungsansatz zur Generierung eines Informationsmodells für den datenbasierten Entwicklungsprozess einer Windenergieanlage dar. Im Folgenden werden daher Grundzüge des MBSE erläutert, auf deren Basis der Aufbau des Informationsmodells

¹⁰³ Kaufmann und Pfenning 2014.

¹⁰⁴ Albers et al. 2015.

konzeptioniert wird. Anschließend werden die identifizierten abzubildenden Informationen durch Anwendung des definierten Konzeptes modelliert.

3.2.2.1 Modellierung mit SysML

In der Modellierungssprache SysML erfolgt die Darstellung der beteiligten Entwicklungssichten auf das Produkt über Diagramme¹⁰⁵. Die Diagramme stellen zweckgebundene Sichten dar, die die Modellierung und den Zugriff auf ausgewählte Informationen aus dem Systemmodell ermöglichen. Abbildung 36 zeigt exemplarisch die Modellierung der Struktur eines Getriebes in Diagramm 1, während Diagramm 2 ein Modell zur Beschreibung der Verzahnungsauslegung des Getriebes beschreibt.¹⁰⁶

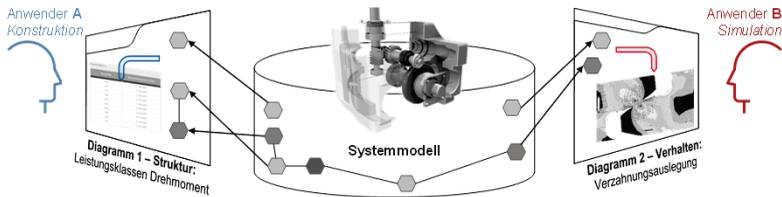


Abbildung 36: Modellierung zweckgebundener Sichten in der SysML über Diagramme nach CASSE¹⁰⁷

Mit den Strukturdiagrammen, den Anforderungsdiagrammen sowie den Verhaltensdiagrammen existieren in SysML drei wesentliche Diagrammtypen. Die für die Konzeption und Modellierung des Informationsmodells insbesondere relevanten Strukturdiagramme sind in Abbildung 37 farblich hervorgehoben und werden im Folgenden detaillierter betrachtet.

Die Modellierung der Struktur eines Modells erfolgt über das Blockdefinitionsdiagramm (engl.: block definition diagram, kurz: bdd). Innerhalb eines Blockdefinitionsdiagramms können die Modellelemente der Struktur dargestellt sowie Eigenschaften definiert und miteinander in Beziehung gesetzt werden¹⁰⁸. Das Grundelement der Struktur in SysML stellt

¹⁰⁵ Casse 2017.

¹⁰⁶ Alt 2012.

¹⁰⁷ Casse 2017; Alt 2012; Eigner et al. 2016.

¹⁰⁸ Friedenthal et al. 2015.

der Block dar. Blöcke können zur Modellierung jeder Art von Systemelementen verwendet werden. Dazu können Blöcke zudem Eigenschaften zugewiesen bekommen, die als „Properties“ bezeichnet werden.¹⁰⁹

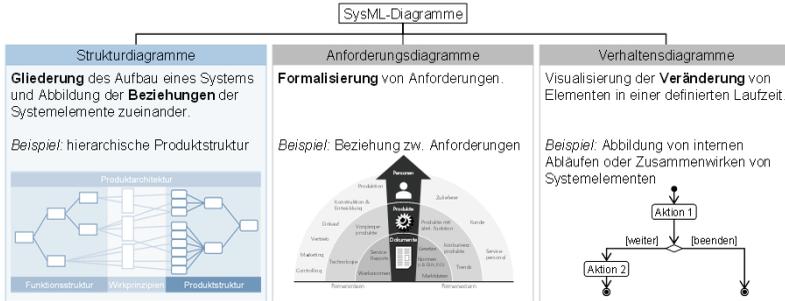


Abbildung 37: Diagrammtypen in SysML¹¹⁰

Um den Aufbau eines Systems zu beschreiben, werden die definierten Blöcke in Beziehung zueinander gesetzt¹¹¹. Die Abbildung dieser Abhängigkeiten erfolgt dabei ebenfalls grafisch. In Hinblick auf die Modellierung von Strukturdiagrammen werden vorrangig die folgenden zwei Abhängigkeitsarten verwendet¹¹²:

Generalisierung: Eine Generalisierung beschreibt eine Vererbungsbeziehung. Generalisierungen werden verwendet, falls Blöcke Gemeinsamkeiten aufweisen. Diese Gemeinsamkeiten werden an die erbenenden Elemente weitergegeben und können zusätzlich durch spezielle Merkmale des erbenenden Elements ergänzt werden.

Komposition: Die Komposition beschreibt eine gerichtete Beziehung zwischen zwei Elementen. So lässt sich modellieren, dass eine Windenergieanlage aus einem Rotor, einem Maschinenhaus und einem Turm besteht.

Die für die Modellierung mit SysML nutzbaren Elemente werden somit durch standardisierte Zeichen (Notationen) beschrieben. Die Notationen können nach den definierten Regeln (Syntax) in SysML miteinander kombiniert werden, sodass sich innerhalb des Systemmodells eine Bedeutung (Semantik) ergibt.

¹⁰⁹ Delligatti 2014.

¹¹⁰ Casse 2017; Delligatti 2014; Object Management Group 2019.

¹¹¹ Friedenthal et al. 2015.

¹¹² Friedenthal et al. 2015; Casse 2017; Alt 2012.

Abbildung 38 fasst die wesentlichen SysML-Elemente zur Beschreibung der Struktur zusammen und zeigt die exemplarische Anwendung für eine Windenergieanlage.¹¹³

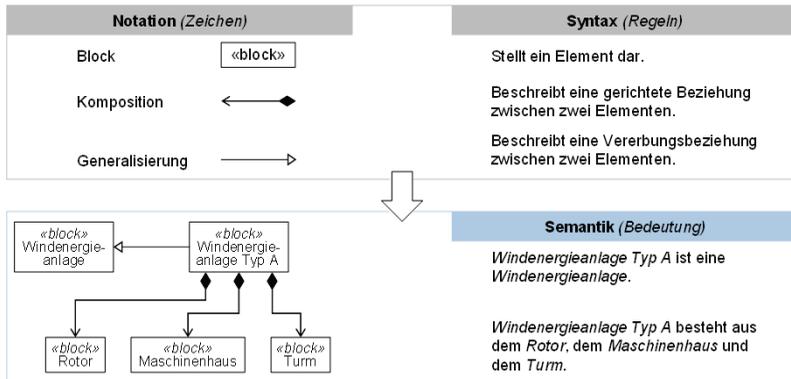


Abbildung 38: Notation, Syntax und Semantik in SysML¹¹⁴

Aufbauend auf der beschriebenen Syntax und Semantik wird im Folgenden das Informationsmodell aufgebaut.

3.2.2.2 Konzeptionierung des Informationsmodells

Durch die Analyse der zur Verfügung gestellten JSON-Files, wurden die abzubildenden Informationen zur formalisierten Beschreibung der WEA-Produktentwicklung identifiziert. Auf der Grundlage dieser relevanten Informationen kann im Folgenden ein Konzept zur Modellierung des Informationsmodells mit SysML erstellt werden. Die Grundstruktur des Informationsmodellkonzepts stellt die Abbildung der wesentlichen baulichen Zusammenhänge der Komponenten und Baugruppen einer Windenergieanlage dar. Im Anschluss können die für die Entwicklung einer Windenergieanlage notwendigen Abhängigkeiten und Parameter der Komponenten und Baugruppen abgebildet werden. Abbildung 39 zeigt einen exemplarischen Auszug des resultierenden Konzeptes zur Modellierung des Informationsmodells in SysML.

¹¹³ Friedenthal et al. 2015.

¹¹⁴ Friedenthal et al. 2015.

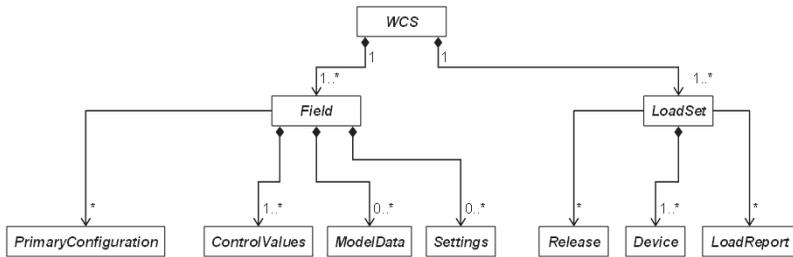


Abbildung 39: Auszug des Konzeptes zur Modellierung des Informationsmodells mit SysML

Das Vorgehen zum Aufbau des SysML-Systemmodells mit dem open-source MBSE-Modeller „Eclipse Papyrus“ wurde in einem Modellierungsleitfaden dokumentiert. Dieser Leitfaden enthält unter anderem die Installation von Eclipse Papyrus unter Berücksichtigung des SysML-Plugins, ein Rahmenwerk zur Anwendung von SysML sowie das Vorgehen zur Modellierung des Informationsmodells. Im Folgenden wird exemplarisch die Modellierung des Informationsmodells zusammengefasst.

3.2.2.3 Modellierung des Informationsmodells

Die Modellierung des Informationsmodells beginnt mit der Abbildung der Produktstruktur der Windenergieanlage. Dabei erfolgt die Unterteilung der Windenergieanlage nach dem Vorbild der durch das anwendende Unternehmen definierten Fachbereiche (engl.: fields). Daher wird ein Block angelegt, der das generische Produkt Windenergieanlage beschreibt und zusätzlich jeweils ein Block pro Fachbereich. Exemplarisch hervorgehoben sei an dieser Stelle der Block „WCS_Gearbox“ zur Modellierung eines generischen Getriebes einer Windenergieanlage zur Erhöhung der Rotordrehzahl vor Eintritt der Welle in den Generator. Die Formalisierung der Beziehungen zwischen der Windenergieanlage und den Fachbereichen erfolgt im Anschluss über Kompositionen, die innerhalb eines Blockdefinitionsdiagramms grafisch modelliert werden (siehe Abbildung 40).

Nach der Abbildung der grundlegenden Windenergieanlagenstruktur werden die einzelnen Fachbereiche weiter detailliert. Abbildung 40 zeigt hierzu Auszüge der SysML-Modellierung des spezifischen Getriebes „WCS_Gearbox_Alpha“ die innerhalb eines weiteren Blockdefinitionsdiagramms erfolgt. Innerhalb des bdd ist über eine Generalisierung modelliert, dass es sich bei dem Getriebe „WCS_Gearbox_Alpha“ um ein

Element vom Typ „WCS_Gearbox“ handelt. Da der Block „WCS_Gearbox“ in unterschiedlichen SysML-Diagrammen verwendet wird, zeigt sich bereits der in Abbildung 36 erläuterte Ansatz der Modellierungssichten.

Dem spezifischen Getriebe „WCS_Gearbox_Alpha“ werden im Nachgang weitere Blöcke über jeweils eine Komposition zugewiesen. Die abzubildenden Informationen sind innerhalb des Auszuges der JSON-Datei auf der linken Seite von Abbildung 40 aufgeführt. Neben dem Dateinamen FileName, Summary und Constraints, zeigt Abbildung 40 die modellierten Blöcke „ModelData“ sowie „ControlValues“. Die zwei Blöcke enthalten jeweils weitere Eigenschaften. Mit diesem Vorgehen können die in den JSON-Dateien beschriebenen Elemente einer Windenergieanlage in SysML angelegt werden. Neben dem Aufführen der Elemente, werden zudem beschreibende Informationen modelliert. Im Anschluss muss beschrieben werden, welche Eigenschaften der Blöcke innerhalb der Produktentwicklung einer Windenergieanlage aufgrund bestehender Abhängigkeiten untereinander ausgetauscht werden müssen.

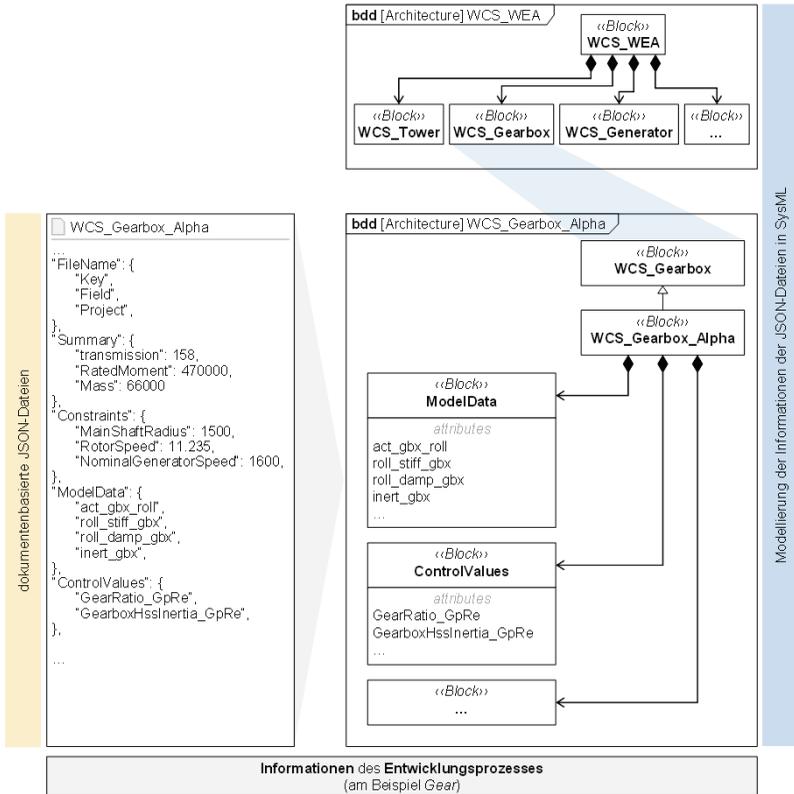


Abbildung 40: Modellierung des Informationsmodells zur Abbildung der Produktstruktur

Die Beschreibung der Abhängigkeiten der zuvor als Blöcke abgebildeten Elemente einer Windenergieanlage ist zum aktuellen Stand ebenfalls über JSON-Dateien abgebildet. So beschreiben Configuration-Files die Wechselwirkungen, die sich beispielsweise zwischen den Simulationswerkzeugen in den Fachdisziplinen ergeben. Im Folgenden wird zur Veranschaulichung das Beispiel der Pitch-Anlage verwendet. Die Pitch-Anlage bezeichnet das Verstellsystem der Rotorblätter. Über das System werden die Rotorblätter relativ zum Hub um die Längsachse verdreht, sodass die Rotor-Vorderkante in die Anströmung hinein oder aus der Anströmung hinaus gedreht werden kann.

Aus der Veränderung des Anstellwinkels resultiert eine Vergrößerung beziehungsweise Verkleinerung der Auftriebskräfte und damit der Leistung der Windenergieanlage.

Das Blockdefinitionsdiagramm der „Pitch_Configuration“ zeigt Abbildung 41 auf der rechten Seite. Zusätzlich ist mit dem bdd „PitchElectric“ die Struktur der Elektrik der Pitch-Anlage dargestellt. Diese enthält mit den „PitchActuatorTorqueLimits“ etwa Angaben über die zulässigen Drehmomente. Werden diese Grenzwerte innerhalb der Auslegung eines Pitch-Verstellsystems nicht eingehalten, ist die entwickelte Windenergieanlage nicht funktionsfähig. Vergleichbare Abhängigkeiten sind in sämtlichen dargestellten Blöcken vorgesehen. So enthält die „PitchMechanic“ Angaben zu strukturellen Kräften und das „PitchSafetySystem“ Definitionen für die Ausführung von Schutzeinrichtungen.

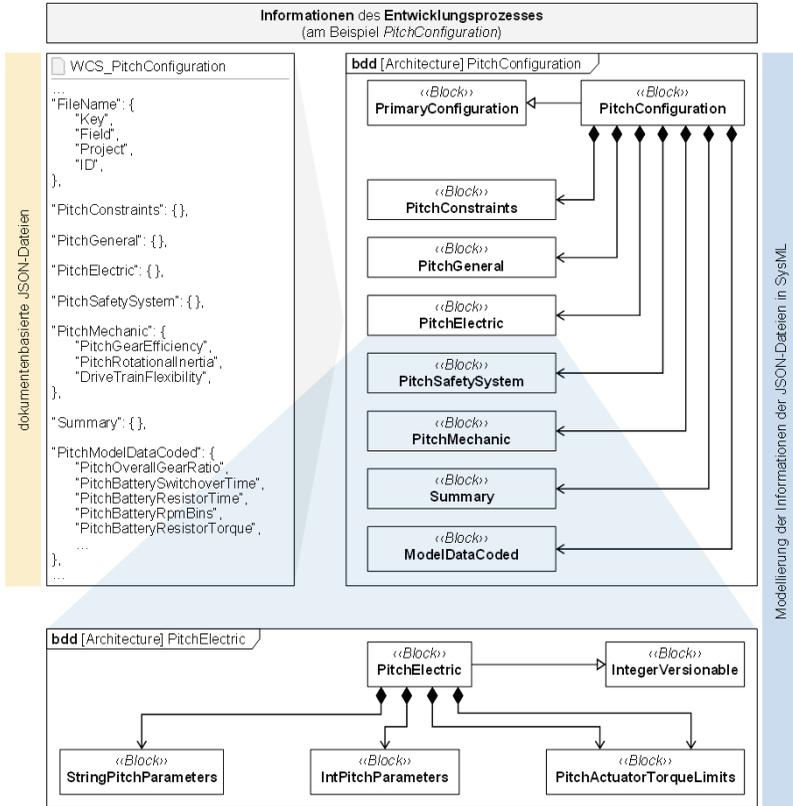


Abbildung 41: Modellierung des Informationsmodells zur Abbildung der Abhängigkeiten

Durch Anwendung des entwickelten Konzeptes auf sämtliche Fachgebiete der seitens des Projektpartners bereitgestellten JSON-Dateien konnte das Informationsmodell der datenbasierten Entwicklung einer Windenergieanlage in SysML modelliert werden. Diese Datenbasis wird im nächsten Schritt von SysML in CD4A transformiert, um die modellierten Informationen in den MontiGem Datenbank-Generator einlesen zu können.

3.3 Detaillierungsphase

Im Rahmen der Detaillierungsphase werden die zuvor skizzierten Konzepte auf Prozess- und Datenebene weiter konkretisiert und um Hinweise für die praktische Durchführung der agilen datenbasierten Produktentwicklung ergänzt. Dabei gibt das Kapitel 3.3.1 zunächst konkrete Hinweise für die erfolgreiche Sprintdurchführung. Dabei wird insbesondere auf die Konfiguration und Synchronisation von Entwicklungssprints eingegangen. Kapitel 3.3.2 beschreibt, wie ausgehend von einem Informationsmodell in SysML (vgl. Kapitel 3.2.2) das Datenmodell in CD4A abgeleitet wird. Der anschließende Aufbau der Datenbank mithilfe des MontiGem-Generators wird in Kapitel 3.3.3 tiefergehend erläutert.

3.3.1 Methodische Unterstützung der Sprintdurchführung

Als methodische Unterstützung bei der Sprintdurchführung wird im Folgenden auf drei Themenbereiche näher eingegangen, die sich im Verlauf des Forschungsprojektes als besonders wichtig für das erfolgreiche agile Entwickeln herausgestellt haben. Zunächst wird in Kapitel 3.3.1.1 die Sprintplanung betrachtet und detailliert beschrieben, wie von unvollständigen Produkthanforderungen auf die durchzuführenden Entwicklungstätigkeiten geschlossen werden kann. Ausgehend von der Anforderungsanalyse wird schrittweise ein geeigneter Sprint als Kette von Prozessbausteinen konfiguriert. Kapitel 3.3.1.2 geht auf die Herausforderung der Informationsweitergabe zwischen parallelen Entwicklungszyklen ein. Die Bestimmung eines übergreifenden Takts sowie die Ausgestaltung von Koordinationspunkten zur vertikalen Synchronisation werden erläutert. Komplementär zur vertikalen Synchronisation beschreibt Kapitel 3.3.1.3 Wechsellpunkte als Mittel zur horizontalen Synchronisation im Entwicklungsprozess. Diese Wechsel des Entwicklungsvorgehens sind die Grundlage für eine hybride Produktentwicklung, die sich den Erfordernissen der jeweiligen Entwicklungsaufgabe anpasst. Insgesamt führen die betrachteten Themenbereiche zu einer strukturierteren Planung von Entwicklungssprints.

3.3.1.1 Konfiguration der Entwicklungssprints

Das Vorgehen zur Konfiguration der Entwicklungssprints folgt dem in Abbildung 42 dargestellten Vorgehen beginnend mit der Aufnahme von

Anforderungen. Sodann werden technische Fragestellungen abgeleitet, priorisiert und operationalisiert. Die Zuordnung von Entwicklungsobjekten zu Fragestellungen bildet die Grundlage für die anschließende Auswahl von Prozessbausteinen. Die einzelnen Schritte werden im Folgenden weiter detailliert.

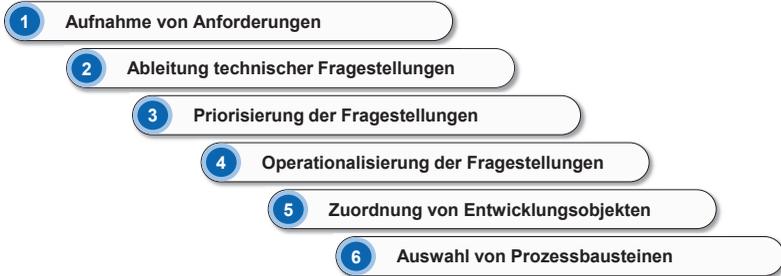


Abbildung 42: Vorgehen zur Konfiguration von Entwicklungssprints

Im ersten Schritt sind die Anforderungen aufzunehmen, denen das zu entwickelnde Produkt entsprechen muss. Diese können beispielsweise durch Angaben in einem Lastenheft vorgegeben sein oder einer User Story entstammen. Auch rechtliche Vorgaben und Erkenntnisse aus Kundengesprächen fließen in die Anforderungsdefinition ein. Resultat ist eine Liste von Anforderungen. Meist wird diese Liste im Zuge der Produktentwicklung laufend modifiziert. Nach Abschluss der Anforderungsaufnahme werden die vorliegenden Anforderungen durch Prüfkriterien nach ihrer Totalität bewertet. Diese Kriterien sind Realisierbarkeit, Korrektheit, Eindeutigkeit, Vollständigkeit und Verifizierbarkeit. Realisierbarkeit ist gegeben, wenn die technische Lösung ausreichend spezifiziert ist. Korrektheit bezieht sich auf die Übereinstimmung der Anforderung mit den Kundenwünschen und ist etwa bei Widersprüchen zwischen Lastenheft und Kundenbefragungen nicht gegeben. Eindeutigkeit herrscht, wenn der Interpretationsspielraum in der Formulierung von Anforderungen ausreichend klein ist und damit bei unterschiedlichen Personen zum gleichen Ergebnis führen würde. Die umfängliche Beschreibung der in der Anforderung zugrundeliegenden Funktionen wird durch die Vollständigkeit gewährleistet. Verifizierbarkeit ist gegeben, wenn eine Prüfung der umgesetzten Anforderungen spezifiziert ist. Das Ergebnis des ersten Schritts ist demnach ein Anforderungskatalog inklusive der Bewertung der Prüfkriterien für die jeweilige Anforderung.

Im zweiten Schritt werden technische Fragestellungen abgeleitet. Da agiles Arbeiten insbesondere zur zielgerichteten Reduktion von Unsicherheiten geeignet ist, sollen die Fragestellungen Unsicherheiten im bestehenden Anforderungskatalog adressieren. Deshalb werden sie anhand der nicht erfüllten Prüfkriterien abgeleitet. Technische Fragestellungen umfassen stets eine eindeutige Bezeichnung, eine konkrete Hypothese bzw. Lösungsmöglichkeit und ein Ziel, das sich aus dem adressierten Prüfkriterium der Fragestellung ableitet. Lautet eine formulierte Anforderung beispielsweise „Als Offshore-Windenergieanlagenbesitzer möchte ich eine möglichst wartungsfreie Windenergieanlage haben, damit ich geringe Wartungsaufwände habe“, so sind die Kriterien der Korrektheit, der Eindeutigkeit und der Verifizierbarkeit erfüllt. Da keine technische Umsetzung spezifiziert wurde und die Reduktion der Wartungsaufwände nicht quantifiziert ist, sind die Realisierbarkeit und die Vollständigkeit jedoch nicht gegeben. Hier werden entsprechende Fragestellungen abgeleitet, um etwa die konkrete Hypothese einer getriebelosen Ausführung der Windenergieanlage zu untersuchen. Das exemplarische Vorgehen ist auch in Abbildung 43 dargestellt. Ergebnis des zweiten Schritts ist eine Liste an technischen Fragestellungen.

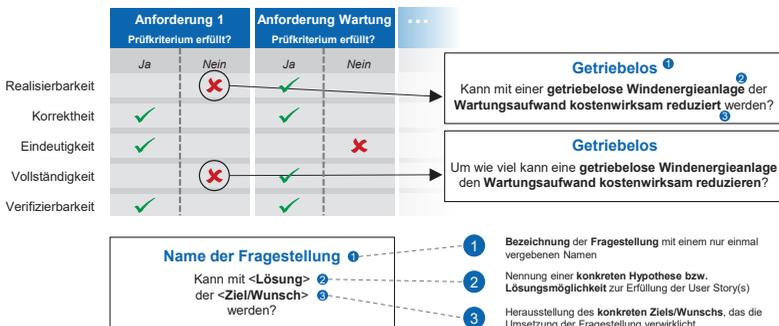


Abbildung 43: Ableitung technischer Fragestellung anhand der Prüfkriterien für Anforderungen

Im dritten Schritt erfolgt die Priorisierung der zuvor abgeleiteten Fragestellungen. Dadurch können die weiteren, teils sehr aufwendigen Schritte zunächst für die am höchsten priorisierten Fragestellungen durchgeführt werden. Hierzu sind drei wesentliche Sichten zu berücksichtigen: die Kundensicht, die Projektsicht und die formale Sicht. Aus

Kundensicht sind die Fragestellungen zu priorisieren, welche die höchste Auswirkung auf den Kundennutzen haben. Darüber hinaus kann es auch Fragestellungen geben, die dem Kunden unabhängig quantifizierbarer Performance wichtig sind. Auch diese Präferenzen sind im Sinne der Kundennähe zu berücksichtigen. Wichtigkeit, Dringlichkeit und Aufwände zur Beantwortung der Fragestellung schlagen sich in der Projektsicht nieder. Besonders komplexe Fragestellungen mit hohem Arbeitsaufwand sollten nicht zu weit verschoben werden. In der formalen Sicht werden logische Zusammenhänge der Fragestellungen untereinander berücksichtigt. Bspw. kann mithilfe einer Design Structure Matrix dargestellt werden, welche Fragestellungen aufeinander aufbauen oder zusammengefasst werden können. Ergebnis des dritten Schritts ist die priorisierte Liste der technischen Fragestellungen. Im vierten Schritt sind die Fragestellungen zu operationalisieren, um sie für den weiteren Entwicklungsprozess handhabbar zu machen. Dazu wurde ein branchenspezifisches Kennzahlensystem für die Windenergiebranche erstellt (vgl. Abbildung 44 dargestellten beispielhaften Kennzahlen). Wesentliche Bezugsgröße ist die Levelized Cost of Energy (LCoE), welche die Kosten zur Erzeugung einer Megawattstunde Energie beziffert. Weitere Kennzahlen stehen mit der LCoE in einem hierarchischen Zusammenhang, können also auf sie zurückgeführt bzw. aus ihr hergeleitet werden. Die unterste Hierarchieebene des Kennzahlensystems umfasst Kennzahlen, die auf einzelne Produktumfänge der generischen Produktstruktur¹¹⁵ referenzieren. Zu operationalisierende Fragestellungen werden mit einer oder mehreren Kennzahlen aus dem System assoziiert. Über die hierarchischen Zusammenhänge können die relevanten Kennzahlen der untersten Hierarchieebene und die entsprechend relevanten Produktumfänge identifiziert werden. Die o.g. Fragestellung zur getriebelosen WEA kann somit etwa mit dem Anteil der Instandhaltungskosten an den Betriebskosten der Anlage assoziiert werden, welche u.a. mit der Ausgestaltung des Antriebsstrangsystems zusammenhängen.

Ergebnis des vierten Schritts sind jeweils operationalisierte Fragestellungen mit konkretem Produktbezug.

¹¹⁵ Eine generische Produktstruktur für die Windenergiebranche wurde im Forschungsprojekt WEA-Baukasten entwickelt.

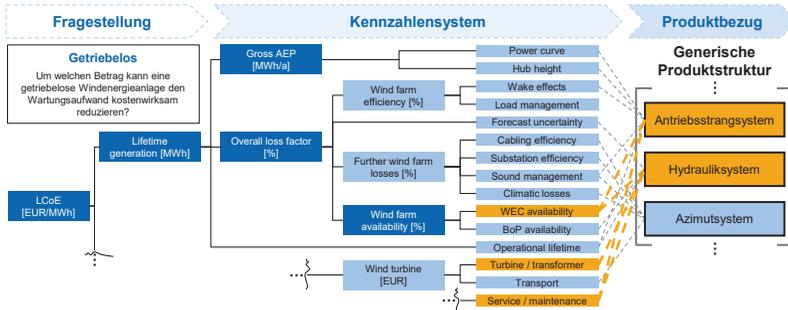


Abbildung 44: Operationalisierung von Fragestellungen mittels Kennzahlensystem und generischer Produktstruktur

Im fünften Schritt werden den Fragestellungen Entwicklungsobjekte zugeordnet. Dazu sind jene Entwicklungsobjekte auszuwählen, deren Erstellung Aufschluss über die Fragestellung gibt, indem sie zur Ermittlung der jeweils produktumfangbezogenen Kennzahlen beitragen. Beispielsweise kann die Ermittlung der Stückliste für den Antriebsstrang Grundlage für die Ermittlung der in die Instandhaltungskosten einfließenden Kosten für Ersatzteile sein. Basierend auf den zu ermittelnden Kennzahlen ergibt sich eine Menge zu erstellender Entwicklungsobjekte, im Beispiel können etwa auch die Kosten zugelieferter Teile von Belang sein. Für die Menge zu erstellender Entwicklungsobjekte werden geeignete MVPs definiert, welches die zu erzeugenden Entwicklungsobjekte, die zu ermittelnden Kennzahlen, die zu beantwortenden Fragestellungen spezifiziert (vgl. Abbildung 45). Ergebnis des fünften Schritts sind damit MVPs für die Beantwortung technischer Fragestellungen.

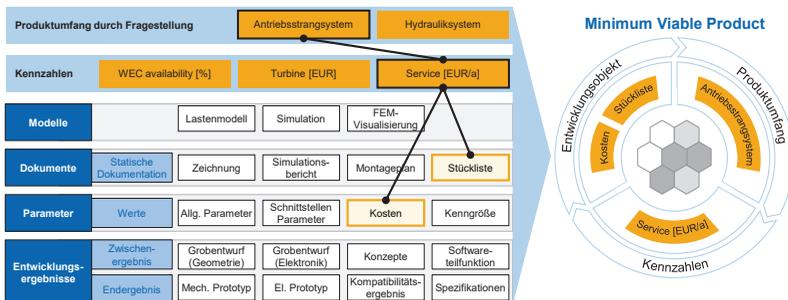


Abbildung 45: Spezifikation von MVPs zur Beantwortung technischer Fragestellungen

Im sechsten Schritt werden Prozessbausteine (PBS) für die Erzeugung der MVPs ausgewählt. Die aus Kapitel 3.2.1.2 bekannten Prozessbausteine sind als ergebnisorientierte Transformationen über ihre Input/Output-Beziehung definiert. Dies erlaubt die Identifikation relevanter PBS, da die im MVP spezifizierten Entwicklungsobjekte Output von PBS sein müssen. Da diese PBS wiederum einen Input benötigen, können passenden Bausteine identifiziert werden, die hier einen entsprechenden Output liefern. Die so entstehenden PBS-Ketten werden solange fortgeführt, bis sie auf bekannte Inputobjekte verweisen und konstituieren den zur Erzeugung des MVPs durchzuführenden Entwicklungssprint. Das Verfahren ist ausführlich an anderer Stelle beschrieben worden und ist exemplarisch in Abbildung 46 dargestellt¹¹⁶.

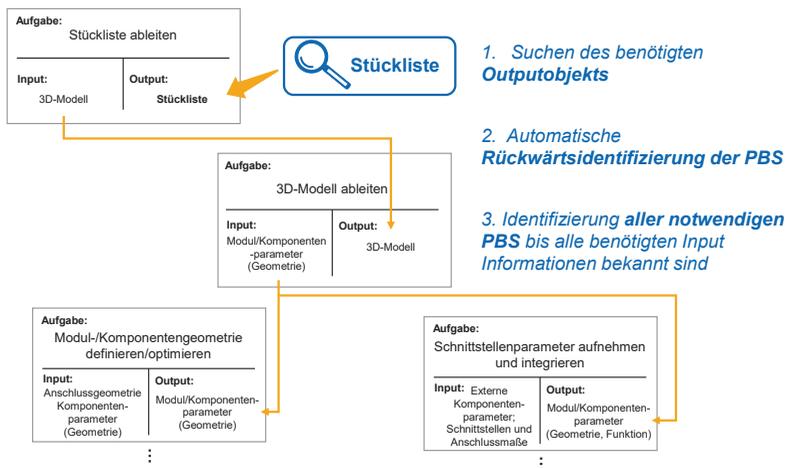


Abbildung 46: Erzeugung von PBS-Ketten zur Konfiguration von Entwicklungssprints

Im Beispiel ist der PBS „Stückliste ableiten“ als notwendig für die Erzeugung des Outputs „Stückliste“ identifiziert. Dementsprechend ist das Input-Objekt „3D-Modell“ dieses PBS zu erzeugen, was durch den PBS „3D-Modell ableiten“ abgebildet werden kann, welcher wiederum Inputs benötigt usw. Ergebnis des sechsten Schritts ist der aus Prozessbausteinen konfigurierte Entwicklungssprint. Da zeitgleich mehrere Sprints

¹¹⁶ Riesener et al. 2022.

durchgeführt werden gilt es neben der Ausarbeitung der einzelnen Sprints auch eine übergeordnete Struktur für die Abstimmungen zwischen den Sprints zu realisieren. Diese Synchronisation wird im Folgenden detailliert beschrieben.

3.3.1.2 Synchronisation und Taktung

Der Begriff der Synchronisation bezeichnet die zeitliche Abstimmung und inhaltliche Koordination von Vorgängen. Das interdisziplinäre, parallele Arbeiten im agilen Vorgehen macht diese Abstimmung und Koordination notwendig, um Widersprüche, unnötige Anpassungen und Fehler zu vermeiden. Dabei hat sich die Einführung einer Taktung als vorteilhaft erwiesen. Unter einer Taktung versteht man die Unterteilung des Zeitraums zwischen zwei Synchronisationspunkten in einheitliche Zeiteinheiten.

Der agile Entwicklungsprozess besteht, wie in Kapitel 3.2.1.1 beschrieben, aus Entwicklungszyklen innerhalb derer das jeweilige Team ein MVP erstellt. Dabei kann die Erstellung unterschiedlicher MVPs unterschiedliche Aufwände induzieren, weshalb sich entsprechend variable Längen der zugehörigen Entwicklungszyklen ergeben. Innerhalb der Entwicklungszyklen ist das Vorgehen in Sprints fixer Länge eingeteilt. Innerhalb eines Sprints wird ein Produktinkrement erzeugt. Synchronisationspunkte sollten idealerweise am Sprintende bzw. vor Sprintbeginn liegen, um die Arbeitsergebnisse an andere Teams zu kommunizieren und die aus der Synchronisation gewonnen Erkenntnisse berücksichtigen zu können. Um einen geeigneten Takt bzw. eine geeignete Platzierung der Synchronisationspunkte zu finden ist also die Festlegung der Sprintlänge(n) der beteiligten Entwicklungszyklen entscheidend.

Grundsätzlich hängt die geeignete Sprintlänge von unternehmens- und branchenspezifischen Randbedingungen ab, etwa dem für die Tätigkeiten im Sprint zu erwartenden Aufwände oder dem geforderten Reifegrad des im Sprint erzeugten Produktinkrements. Gerade im Maschinen- und Anlagenbau ergeben sich jedoch je beteiligter Disziplin stark unterschiedliche Randbedingungen. In der Windenergiebranche ist etwa in der Softwareentwicklung eine Sprintdauer von zwei Wochen üblich, wohingegen die Tätigkeiten in der Lastenrechnung einem zwei Wochen Intervall nicht entsprechen können, weil die Erstellung der Berichte mit Rechenzeiten verbunden ist und sich nicht zur Bereitstellung von Zwischenergebnissen eignet.

Der in Abbildung 47 gezeigte Vergleich zwischen der Option einer einheitlichen Sprintlänge über die verschiedenen Entwicklungszyklen hinweg und einer zykluspezifischen Sprintlänge verdeutlicht das sich ergebende Dilemma. Eine einheitliche Sprintlänge ermöglicht die effiziente Weitergabe von Informationen. Sprints, die zum selben Zeitpunkt starten, enden auch am selben Zeitpunkt, sodass nach jedem Sprint Ergebnisse weitergegeben und vor jedem Spint empfangene Ergebnisse berücksichtigt werden können. Allerdings kann das Arbeiten innerhalb der Zyklen ineffizient sein. Muss die Softwareentwicklung bspw. eine deutlich längere Sprintlänge adaptieren, erhält sie weniger häufig und weniger unmittelbar Feedback zu den erstellten Inkrementen. Damit steigt das Risiko für Fehlentwicklungen. Auf die umgekehrten Schwierigkeiten der Lastenrechnung bei einer zu kurz gewählten Sprintlänge ist oben bereits eingegangen worden.

Eine zykluspezifische Sprintlänge ermöglicht hingegen ein möglichst effizientes Arbeiten innerhalb des jeweiligen Teams, indem eine präzise auf die jeweiligen Randbedingungen angepasste Sprintlänge gewählt wird. Die Weitergabe von Informationen gestaltet sich jedoch schwieriger. Aufgrund der unterschiedlichen Sprintlängen enden parallel begonnene Sprints zu unterschiedlichen Zeitpunkten. Die erzeugten Ergebnisse können damit nicht direkt weitergegeben werden und erst nach Abschluss der noch laufenden Sprints Berücksichtigung finden. Für den Start des nächsten Sprints muss auf dem letzten bekannten Stand aufgesetzt werden. Dieser kann schon zum Zeitpunkt der Sprintplanung veraltet sein, weil sich in den laufenden Sprints neue Erkenntnisse ergeben haben.

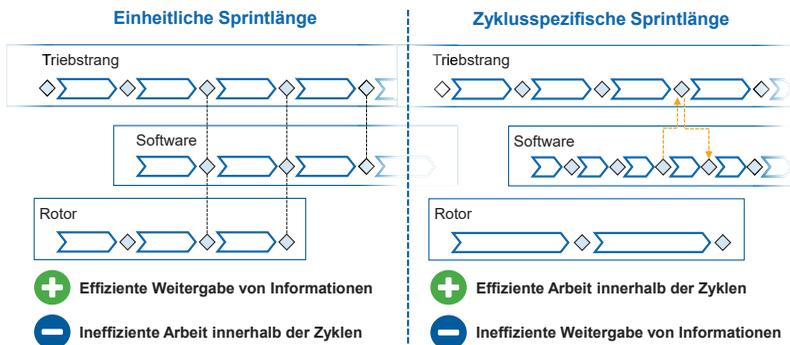


Abbildung 47: Vergleich einheitlicher und zykluspezifischer Sprintlänge

Die Lösung liegt in der in Abbildung 48 dargestellten gestaffelten Sprintlänge. Jede beteiligte Abteilung wählt ihre Sprintlänge als Vielfaches des führenden Entwicklungszyklus. Dieser gibt die Taktung der Synchronisation vor. Damit arbeiten einerseits die einzelnen Abteilungen möglichst nahe ihres Optimums. Andererseits kann die Synchronisation koordiniert erfolgen, indem sie am Sprintende des führenden Entwicklungszyklus erfolgt. Im gezeigten Beispiel ist die Triebstrangentwicklung der führende Zyklus mit einer Sprintlänge von 4 Wochen. Diese Sprintlänge ist ebenfalls für die Rotorentwicklung geeignet, die Lastenrechnung entscheidet sich hingegen für eine 8-wöchige Sprintdauer und kann damit nur zu jeder zweiten Synchronisation Ergebnisse beisteuern. Für die Softwareentwicklung liegt die Sprintlänge bei zwei Wochen, sodass nach jedem zweiten Sprint eine Synchronisation erfolgt.

Der führende Entwicklungszyklus kann anhand von vier Kriterien bestimmt werden. Er sollte zunächst prägend für den Entwicklungsverlauf sein. Führende Entwicklungszyklen werden über weite Teile des Entwicklungsprozesses betrieben und betreffen i. d. R. zentrale Produktumfänge aus technischer Sicht sowie hinsichtlich der vom Unternehmen abgebildeten Wertschöpfung. Eng mit diesem Kriterium verknüpft ist auch das weitere Kriterium des hohen Vernetzungsgrads. Die Ergebnisse des führenden Zyklus sind von hoher Relevanz für die anderen Zyklen und beeinflussen diese. Da die Sprintlänge im führenden Entwicklungszyklus anhand der spezifischen Randbedingungen gewählt wird und dieser damit am nächsten an seinem individuellen Optimum arbeitet, sind führende Entwicklungszyklen zudem oft zeitkritisch. Das vierte Kriterium bildet die Kundenrelevanz. Da agiles Arbeiten sich grundsätzlich durch seine Kundenorientierung auszeichnet, entwickeln führende Zyklen oft auf Kundensicht besonders relevante Umfänge. Insbesondere ist festzuhalten, dass der führende Entwicklungszyklus nicht notwendigerweise die längste oder kürzeste Sprintlänge aufweist.

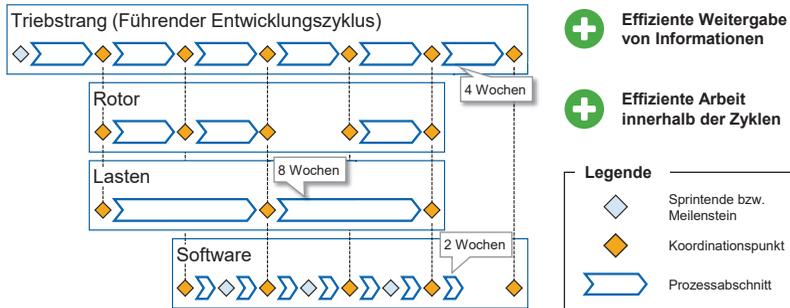


Abbildung 48: Die gestaffelte Sprintlänge vereint die Vorteile eines effizienten Arbeitens mit einer effizienten Informationsweitergabe

Die Synchronisation zwischen den Zyklen, auch vertikale Synchronisation genannt, erfolgt in Koordinationspunkten. In diesen Punkten tragen alle Abteilungen ihre jüngsten Ergebnisse bei, wobei sich jeweils andere Notwendigkeiten in der Ausgestaltung ergeben je nachdem, ob im vergangenen Zeitraum plangetrieben oder agil entwickelt wurde (vgl. Kapitel 3.2.1.3 und Kapitel 3.3.1.3). Im plangetriebenen Vorgehen genügt es die in ihrer Form erwartbaren Ergebnisse durch Ablage oder Weiterleitung bereitzustellen. Im agilen Vorgehen gewonnene Erkenntnisse sind ggf. zu diskutieren, um die entsprechenden Implikationen abzuleiten. Ziel ist es, die Entwicklungsergebnisse zu sichern, abzugleichen und ggf. Konflikte aufzulösen, damit auf einer gemeinsamen Basis weiterentwickelt werden kann. Die Koordinationspunkte stellen insbesondere keine zusätzliche Validierung der Produktinkremente im Sinne des Sprint-Review dar. Für den Sonderfall von Sprintlängen, die größer als die des führenden Prozesses sind, gilt die gemeinsame Basis nur eingeschränkt (bzw. nur für die mit dem Sprintende des länger getakteten Zyklus). Hier sind ggf. mehr Konflikte aufzulösen und Ergebnisse zu revidieren. Dieser Umstand wird aber dafür in Kauf genommen, parallel entwickeln zu können. Abbildung 49 zeigt den prozessualen Ablauf der vertikalen Synchronisation mithilfe von Koordinationspunkten, wobei ein besonderes Augenmerk auf die technische Befähigung durch ein übergreifendes Datenmanagement gelegt wird.

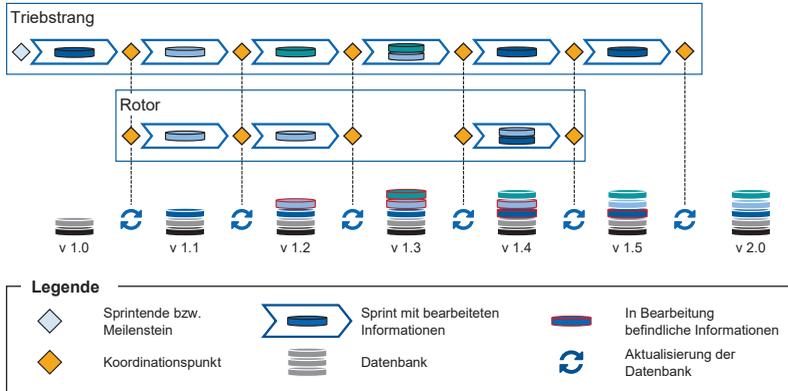


Abbildung 49: Vertikale Synchronisation zwischen den Entwicklungszyklen durch Koordinationspunkte

Für die Entwicklungstätigkeiten im Sprint werden Informationen benötigt und durch eine Datenbank bereitgestellt. Im Sprint selbst werden neue Informationen erzeugt oder bestehende Umfänge verändert, daher sind entsprechende Umfänge in der Datenbank als in Bearbeitung befindlich zu markieren. Im Koordinationspunkt wird die Datenbank aktualisiert. Dabei werden sowohl die neu erzeugten Informationen abgelegt als auch Aktualisierungen an bestehenden Informationen vorgenommen, bspw., wenn sich aufgrund einer Geometrieänderung am Blatt ein neues Strömungsverhalten einstellt. Im Zuge dieser Aktualisierung müssen auch mögliche Konflikte aufgelöst werden. Setzt etwa die besagte Geometrieänderung eine Blattlänge von 64 bis 69 m an, die Strömungssimulation ist aber nur für Blattlängen bis 67 m aussagefähig, so kann entweder die Blattlänge entsprechend begrenzt, die Simulation entsprechend erweitert oder die Blattlänge im Überschneidungsbereich zwischen 64 und 67 m ausgewählt werden. Um die Auflösung von Konflikten zu vermeiden, können Änderungen an bestehenden Informationen auch strikt sequentiell erfolgen.

3.3.1.3 Wechsellpunkte für die Änderung der Vorgehensweise

Nachdem im vorangegangenen Kapitel die vertikale Synchronisation zwischen verschiedenen Entwicklungszyklen betrachtet wurde, behandelt dieses Kapitel die horizontale Synchronisation innerhalb eines Entwicklungszyklus. Dabei liegt der Fokus auf dem Übergang von agilem zu plangetriebenem Vorgehen und umgekehrt. Geeignete Zeitpunkte

für einen solchen Wechsel der Vorgehensweise sind die Sprintenden bzw. Meilensteine. Sie markieren Ruhepunkte im Prozess, sind aufgrund der übergreifenden Taktung ohnehin vorhanden und auch inhaltlich ist die Wahl dieser Zeitpunkte sinnvoll, da hier die Planungstätigkeiten für den jeweils nächsten Projektschritt verortet sind. In diese Planungstätigkeiten ist auch die Neubewertung der Agilitätseignung zu integrieren.

In Kapitel 3.2.1.1 wurde bereits diskutiert, dass sich das agile wie das plangetriebene Vorgehen jeweils für unterschiedliche Entwicklungsaufgaben gut eignen und dass die Bewertung mittels Stacy Matrix hier erste Anhaltspunkte liefert. Zur detaillierteren Bewertung werden die Agilitätsindikatoren zur Bestimmung des Agilisierungspotenzials aus Kapitel 3.1.1.3 herangezogen. Eine initiale Bewertung der Indikatoren Kunde, Dynamik, Umfang, Kritikalität und Output gibt dementsprechend Aufschluss darüber, ob im nächsten Prozessabschnitt bzw. Sprint plangetrieben oder agil gearbeitet werden soll.

Das Vorgehen zur horizontalen Synchronisation innerhalb des Entwicklungszyklus ist in Abbildung 50 dargestellt. Am Sprintende bzw. Meilenstein erfolgt eine Neubewertung der Agilitätsindikatoren. Im Beispiel sinkt etwa nach zwei agilen Sprints die Dynamik, da unvollständige Anforderungen basierend auf den durchgeführten Entwicklungstätigkeiten ergänzt werden konnten. Das Vorgehen wechselt von agil zu plangetrieben. Aufgrund neuer Kundenanforderungen steigen jedoch im Folgenden Dynamik, Kritikalität und Output an und begründen den Wechsel zurück zum agilen Vorgehen.

Die Neubewertung am Sprintende kann grundsätzlich in derselben Weise erfolgen wie die initiale Bewertung. In der Praxis hat sich jedoch eine weniger umfangreiche Bewertung bewährt. Dabei werden nur diejenigen Indikatoren betrachtet, bei denen eine Änderung vermutet wird und die Änderung auch im Wesentlichen grob abgeschätzt, statt eine vollständige Neubewertung vorzunehmen. Dadurch können Ressourcen gespart werden. Ebenfalls hat sich speziell in den Prozessen des Praxisunternehmens gezeigt, dass die Arbeit in den Zyklen i. d. R. agil startet und dann bei ausreichend reduzierter Unsicherheit in das plangetriebene Vorgehen wechselt.

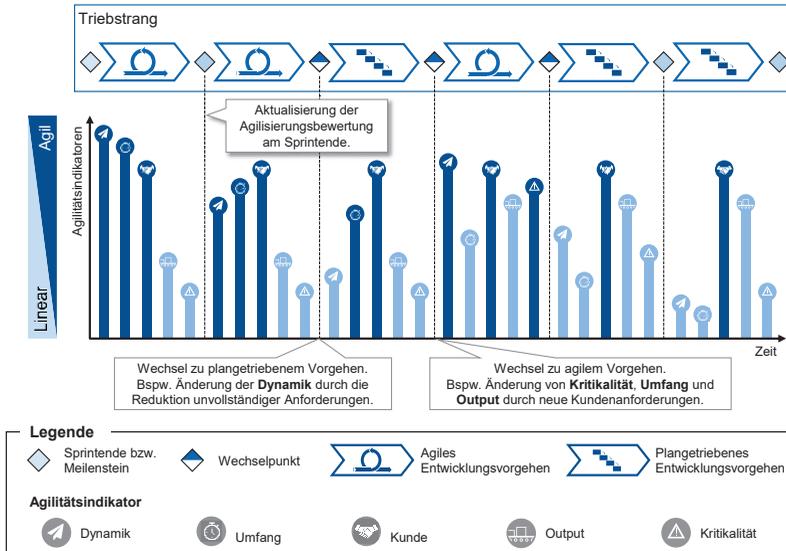


Abbildung 50: Horizontale Synchronisation innerhalb des Entwicklungszyklus durch Wechsellpunkte

Beim Vorgehenswechsel werden die veränderlichen Dimensionen des Agilisierungspotenzials angepasst. Abbildung 51 zeigt eine beispielhafte Checkliste, mithilfe derer ein effizienter Wechsel unterstützt wird. Die Teamzusammensetzung kann beim Wechsel von einem agilen zu einem plangetriebenen Vorgehen derart geändert werden, dass nur die tatsächlich benötigten Fachkompetenzen vorgehalten werden. Domänenexpert*innen verlassen das Team oder mehrere Teams werden zusammengeschlossen und arbeiten stark arbeitsteilig. Einzelne Nachfragen werden teamextern geklärt, interdisziplinärer Austausch ist weniger relevant. Eine effiziente Ausführung der Tätigkeiten steht im Vordergrund.



Abbildung 51: Checkliste für den Wechsel des Entwicklungsvorgehens

Beim Wechsel von einem plangetriebenen Vorgehen zu einem agilen Vorgehen muss der prozesseitige Planungshorizont reduziert werden; im Vordergrund steht klar der aktuelle bzw. jeweils nächste Sprint. Die Führung als Projektorganisation wird forciert und die demokratische, selbstbestimmte Teamorganisation gewinnt an Bedeutung.

Auch das Umfeld wird angepasst. Beispielsweise ist beim Wechsel hin zum plangetriebenen Vorgehen eine globale Verortung der Teammitglieder leichter realisierbar, weil Tätigkeiten unabhängig voneinander durchgeführt werden können. Entsprechend sinkt auch die Absprachehäufigkeit. Gegebenenfalls tritt ein zweiwöchiger Regeltermin an die Stelle von täglichen Stand-ups. In der Regel definiert eine Organisation jeweils für sich, welche Aspekte beim einem Vorgehenswechsel im Entwicklungsprozess geändert werden sollen und welche explizit beibehalten werden, um eine gewisse Konsistenz zu sichern.

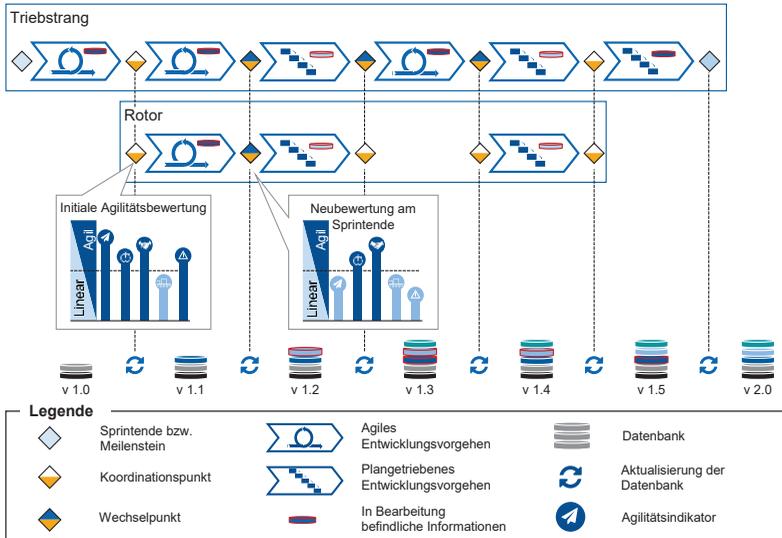


Abbildung 52: Koordinations- und Wechselpunkte strukturieren die vertikale und horizontale Synchronisation des Entwicklungsvorgehens

Das sich ergebende Gesamtzielbild für die vertikale und horizontale Synchronisation ist in Abbildung 52 dargestellt. Der übergreifende Takt strukturiert das Vorgehen. Koordinationspunkte ermöglichen den Informationsaustausch am Sprintende bzw. Meilenstein. Legt die Überprüfung der Agilitätsindikatoren einen Wechsel des Entwicklungsvorgehens nahe, so erfolgt dieser im Wechselpunkt. Damit folgt das Vorgehen dem aktuellen Agilisierungspotenzial. Die Synchronisation ermöglicht ein effektives und effizientes paralleles Arbeiten. Der dafür benötigte Austausch an Informationen wird insbesondere durch die zugrundeliegende Datenbank unterstützt, welche entsprechend aktualisiert wird. Der Aufbau der Datenbank wird dabei im anschließenden Unterkapitel 3.3.2 mithilfe des Datenmodell näher beleuchtet.

3.3.2 Ableitung Datenmodell

Aus dem Informationsmodell wird innerhalb der Ableitung des Datenmodells die Grundlage zur Generierung der Datenbank mit dem MontiGem-Datenbankgenerator erstellt. Der MontiGem-Datenbankgenerator verarbeitet als Input das Datenformat Class Diagram for Analysis (kurz: CD4A). Daher besteht die Notwendigkeit die in SysML modellierten Daten des Informationsmodells in die Sprache CD4A zu transformieren. Für diese Aufgabe wurde ein Datenparser entwickelt. Die Erstellung des Datenparsers erfolgt mit der Programmiersprache Python, um die zukünftige Wartbarkeit und Erweiterbarkeit gewährleisten zu können. Durch die Transformation des Eclipse Papyrus-Informationsmodells in eine Beschreibung basierend auf Klassendiagrammen, entsteht damit das konkrete Datenmodell zur Generierung der Datenbank.

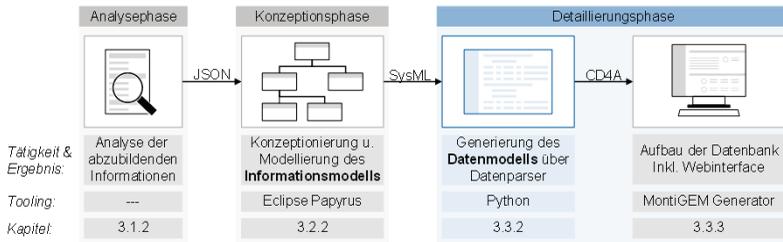


Abbildung 53: Vorgehen zur Generierung der Datenbank

Abbildung 53 hebt die Ableitung des Datenmodells innerhalb des Gesamtworkflows zum Aufbau der Datenbank inklusive Webinterface farblich hervor. Im Folgenden wird die durchgeführte Datentransformation zu einem CD4A Format detaillierter betrachtet. Dazu wird zunächst der Input in Form einer .uml-Datei als Auszug des SysML-Systemmodells betrachtet, welche im weiteren mithilfe eines Datenparsers ins CD4A Format transformiert wird

3.3.2.1 Analyse der Repräsentation des Systemmodells in Eclipse Papyrus

Die Dokumentation grafisch modellierter Informationen des Systemmodells erfolgt in Eclipse Papyrus textuell in einer .uml-Datei. Diese Datei dient als Input für den Datenparser, da somit kein separater Datenexport erforderlich ist und alle erforderlichen Informationen direkt und ohne Datenverlust zugänglich sind. Abbildung 54 zeigt einen Auszug des grafisch modellierten Blockdefinitionsdiagramms „WCSTMaster“ sowie die Repräsentation des bdd in der .uml-Datei. Dabei sind sich entsprechende Elemente innerhalb beider Darstellungsformen hervorgehoben. Im Folgenden wird beispielhaft der SysML-Block „WCSTMaster“ im Detail betrachtet werden:

- Anlegen des SysML-Blocks WCSTMaster: Das Anlegen eines SysML-Blocks wird in der Datei WCS.uml als „packagedElement“ repräsentiert. Im Anschluss erfolgt die Zuweisung des Namens über den Ausdruck „name“.
- Zuweisen der Eigenschaft „wcsVersion“ mit der Einheit Integer zum Element „WCSTMaster“: Die grafisch modellierte Eigenschaft „wcsVersion“ vom Datentyp Integer wird in der WCS.uml als „ownedAttribute“ bezeichnet.
- Vererbungsbeziehung zwischen den Elementen „IntegerVersionable“ und „WCSTMaster“: Der SysML-Block „WCSTMaster“ ist über eine Generalisierungsbeziehung mit dem Block „IntegerVersionable“ verbunden. „WCSTMaster“ erbt somit sämtliche Eigenschaften von „IntegerVersionable“. In der WCS.uml wird diese Beziehung als „generalization“ abgebildet.

Anders als die zuvor erläuterte Vererbungsbeziehung werden Kompositionen in der Repräsentation des grafisch modellierten SysML-Systemmodells als „ownedAttribute“ aufgeführt. Besitzt die Komposition wie im Falle der Verbindung zwischen „WCSTMaster“ und „ControlValueDefinition“ Namen (hier: Blade) und Multiplizitäten (hier: *), werden diese ebenfalls textuell beschrieben.

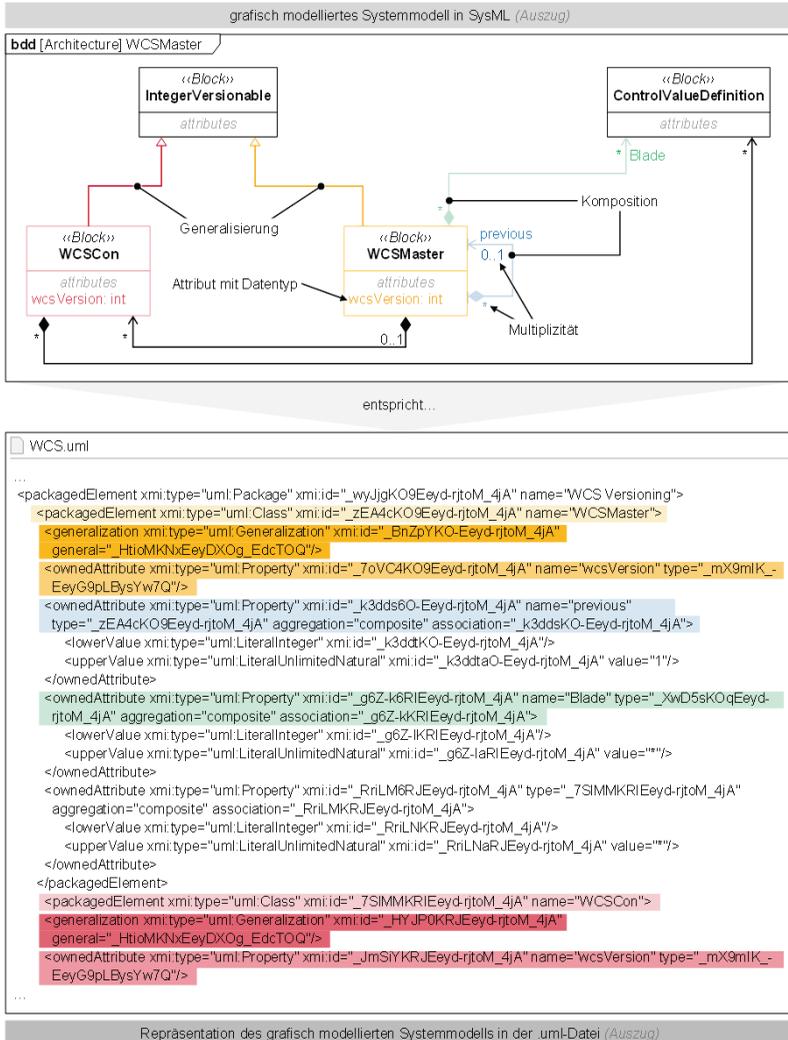


Abbildung 54: Repräsentation des grafisch modellierten Systemmodells in Eclipse Papyrus

So kann die Repräsentation des grafisch modellierten Systemmodells in dem MBSE-Modeller Eclipse Papyrus analysiert werden. Ein identischer oder an dieses Vorgehen angelehnter Schritt müsste bei einer Übertragung des dieses Leitfadens auf andere Projekte ebenfalls durchgeführt werden. Auf Basis der Ergebnisse muss im nächsten Schritt ein Datenparser entwickelt werden, der die Datengrundlage des SysML-Systemmodells in das Datenformat CD4A transformiert.

3.3.2.2 Datenparser zur Transformation von SysML zu CD4A

Der Datenparser identifiziert die grafisch mit SysML modellierten Systemmodell-Elemente sowie deren Verknüpfungen untereinander und transformiert diese in ein zusammenhängendes Klassendiagramm im Format CD4A. Abbildung 55 fasst diese grundsätzliche Aufgabe des Datenparser zusammen und zeigt einen Auszug des Quellcodes zur Ausleitung der in Abbildung 54 exemplarisch gezeigten Informationen aus der .uml-Datei des SysML-Systemmodells.

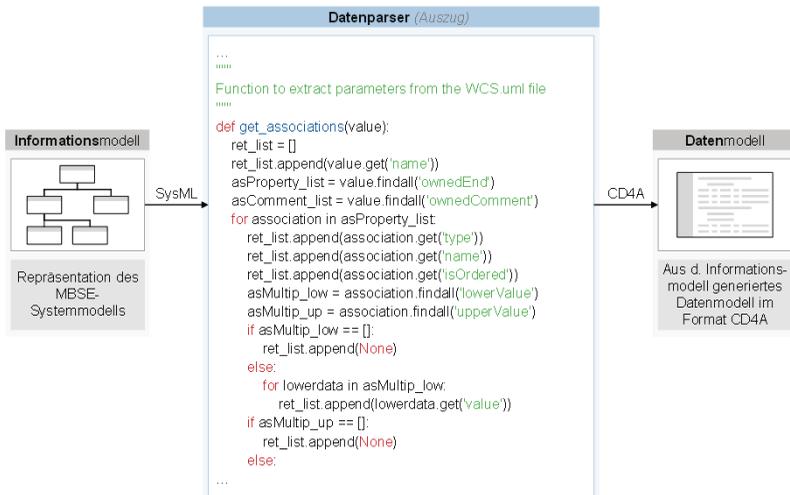


Abbildung 55: Auszug aus dem Quellcode des entwickelten Datenparsers

Die Datentransformation wird durch die Ausführung der Python-Datei „main“ durchgeführt. Dazu verwendet der Datenparser die WCS.uml-Datei welche in dem gleichen Ordner hinterlegt ist. Das erstellte Datenmodell im Format CD4A als generierter Output des Parsers wird in dem Ordner Output als Datei „out“ abgespeichert.

Im Folgenden wird zur Veranschaulichung der Datentransformation eine beispielhafte Anwendung des Datenparsers betrachtet. Abbildung 56 zeigt dazu die Transformation des SysML-Blocks „WCSMaster“ und seiner Beziehungen in die CD4A-Klasse „WCSMaster“. Die farblich hervorgehobenen Datentransformationen von SysML in CD4A stellen einen Auszug der Datenparser-Anwendung dar und umfassen beispielhaft die folgenden Übersetzungen:

- SysML-Blöcke werden als Klassen (engl.: class) in CD4A überführt. Die Namen der Blöcke werden identisch übernommen. Eigenschaften der Blöcke werden den Klassen zugewiesen und der jeweilige Datentyp wird angegeben.
- Vererbungsbeziehungen des Systemmodells werden in den Ausdruck „erbende Klasse extends vererbende Klasse“ transformiert.
- Kompositionsbeziehungen aus der SysML werden in CD4A über den Ausdruck association gekennzeichnet. Angefügte Multiplizitäten werden in eckigen Klammern übernommen.

Repräsentation des grafisch modellierten Systemmodells in der uml-Datei (Auszug)

```

WCS uml
...
<packageElement xmi:type="uml:Package" xmi:id="_vyJgkO9Eeyd-rjtoM_4jA" name="WCS Versioning">
  <packageElement xmi:type="uml:Class" xmi:id="_zEA4ckO9Eeyd-rjtoM_4jA" name="WCSMaster">
    <generalization xmi:type="uml:Generalization" xmi:id="_BnZpYKO-Eeyd-rjtoM_4jA"
      general="_HtioMkNxEeyDXOg_EdcTOQ?/">
      <ownedAttribute xmi:type="uml:Property" xmi:id="_7oVC4KO9Eeyd-rjtoM_4jA" name="wcsVersion" type="mX9mIK_
        EeyG9pLEysYw7Q?/">
      <ownedAttribute xmi:type="uml:Property" xmi:id="_k3dds6O-Eeyd-rjtoM_4jA" name="previous"
        type="_zEA4ckO9Eeyd-rjtoM_4jA" aggregation="composite" association="_k3ddsKO-Eeyd-rjtoM_4jA">
        <lowerValue xmi:type="uml:LiteralInteger" xmi:id="_k3ddtKO-Eeyd-rjtoM_4jA"/>
        <upperValue xmi:type="uml:LiteralUnlimitedNatural" xmi:id="_k3ddtaO-Eeyd-rjtoM_4jA" value="1"/>
      </ownedAttribute>
      <ownedAttribute xmi:type="uml:Property" xmi:id="_g6Z-k6RIEeyd-rjtoM_4jA" name="Blade" type="XwD5sKOqEeyd-
        rjtoM_4jA" aggregation="composite" association="_g6Z-kKRIEeyd-rjtoM_4jA">
        <lowerValue xmi:type="uml:LiteralInteger" xmi:id="_g6Z-lKRIEeyd-rjtoM_4jA" value="1"/>
        <upperValue xmi:type="uml:LiteralUnlimitedNatural" xmi:id="_g6Z-lKRIEeyd-rjtoM_4jA" value="1"/>
      </ownedAttribute>
      <ownedAttribute xmi:type="uml:Property" xmi:id="_RrILM6RJEeyd-rjtoM_4jA" name="ControlValueDefinition"
        aggregation="composite" association="_RrILMKRJEeyd-rjtoM_4jA">
        <lowerValue xmi:type="uml:LiteralInteger" xmi:id="_RrILNKRJEeyd-rjtoM_4jA" value="1"/>
        <upperValue xmi:type="uml:LiteralUnlimitedNatural" xmi:id="_RrILNKRJEeyd-rjtoM_4jA" value="1"/>
      </ownedAttribute>
    </packageElement>
  <packageElement xmi:type="uml:Class" xmi:id="_7S1MMKRIEeyd-rjtoM_4jA" name="WCSCon">
    <generalization xmi:type="uml:Generalization" xmi:id="_HYJP0KRJEeyd-rjtoM_4jA"
      general="_HtioMkNxEeyDXOg_EdcTOQ?/">
      <ownedAttribute xmi:type="uml:Property" xmi:id="_JmSIYKRJEeyd-rjtoM_4jA" name="wcsVersion" type="mX9mIK_
        EeyG9pLEysYw7Q?/">
    </packageElement>
  ...

```

Datentransformation

```

out.cd
...
class WCSMaster extends IntegerVersionable {
  int wcsVersion;
}

association [*] WCSMaster -> (previous) WCSMaster [0..1];
association [*] WCSMaster -> (Blade) ControlValueDefinition [*];
<<ordered>> association [0..1] WCSMaster -> WCSCon [*];

class WCSCon extends IntegerVersionable {
  int wcsVersion;
}

<<ordered>> association [*] WCSCon -> ControlValueDefinition [*];
...

```

Output des Datenparsers im Format CD4 (Auszug)

Abbildung 56: Transformation des grafisch modellierten Systemmodells in das Format CD4A

Das somit erarbeitete Datenmodell stellt den Input des MontiGem-Datenbankgenerators dar. Der Aufbau der Datenbank wird im folgenden Kapitel 3.3.3 aufgezeigt.

3.3.3 Benutzung Datenmodell

Nach der Generierung eines Datenmodells wird daran anschließend eine Datenbank mittels Webinterface generiert. Abbildung 57 zeigt die Einordnung des Methodenschrittes im Gesamtkontext. Für die Nutzung der Datenbank wird in Abschnitt 3.3.3.1 das zugrundeliegende Datenschema und in Abschnitt 3.3.3.2 MontiGem als Framework für den Aufbau der Nutzerschnittstelle beschrieben. Genauere funktionale Beschreibungen der Datenbank folgen in Abschnitt 3.3.3.3 mit der Generierung von Webseiten, in Abschnitt 3.3.3.4 mit der Detaillierung der Python-Schnittstelle, der Darstellung der Versionierung in Abschnitt 3.3.3.5 und der Beschreibung, wie Simulationsdatenvisualisiert werden können, in Abschnitt 3.3.3.6.

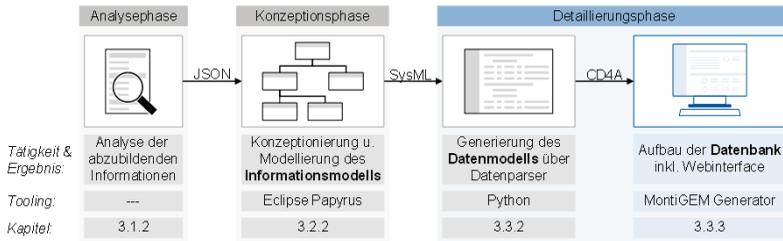


Abbildung 57: Vorgehen zur Generierung der Datenbank

3.3.3.1 Zugrundeliegendes Datenschema

Das Datenschema wird in dem cd4a-Format gespeichert. Im Vergleich zu Alternativen, welche z.B. JSON oder XML basiert sind, ist das cd4a Format einfacher zu lesen und schreiben. Formate, welche nicht maschinenlesbar sind, wie PDF oder PowerPoint, können nur mit großem Aufwand in der modellbasierten Entwicklung eingesetzt werden.

Auch ermöglicht cd4a die Verwendung von automatischen Tools. Aus dem Datenschema wird 1) ein Server-Backend mit Datenbank und API-Schnittstellen 2) ein grafisches Frontend als HTML-Website und 3) Python-Code generiert.¹¹⁷

Diese Generierung ist vollständig automatisiert. Wenn der Domain-Experte z.B. eine neue Datenklasse hinzufügt, ist diese sofort auf der Website sichtbar und kann mit der Python API gelesen und verändert werden.

¹¹⁷ GitHub 2022.

Für den produktiven Einsatz der modellgetriebenen Methodik mit Daten aus dem Produktionssystem ist eine Analyse der möglichen Änderungen des Datenschemas, und deren möglichen Auswirkungen wichtig. Die folgenden Fragen helfen dabei:

- Wie häufig tritt diese Änderung auf?
- Wie dringend ist die Änderung?
- Ist eine Datenmigration notwendig / wie aufwendig ist eine Migration?
- Kann existierendes Tooling weiterhin genutzt werden?
- Ist die Archivierung weiterhin gegeben?

Änderungen, welche sehr häufig auftreten, sollten durch das Tool automatisiert ablaufen können. Dies bedeutet, dass die Datenmigration über Schnittstellen und in ihrer Ausprägung möglichst standardisiert abbildbar sein sollten. Für seltene Änderungen, bspw. die Migration einer Datenbank, ist auch eine manuelle Durchführung denkbar.

Bei allen Schritten ist zu beachten, dass bereits existierende Software mitbedacht werden muss, um sicherzustellen das APIs existieren über die ein Austausch der relevanten Daten aufwandsarm möglich ist.

3.3.3.2 Genutztes Framework: MontiGem

Das am Software-Engineering Lehrstuhl der RWTH Aachen entwickelte Framework MontiGem¹¹⁸ wurde als Fundament zum Aufbau einer Systemarchitektur genutzt und um spezifische Features mithilfe des Python API erweitert. In Abbildung 58 wird der MontiGem Generierungsprozess visualisiert.

¹¹⁸ Adam et al. 2018; Gerasimov et al. 2020a; Gerasimov et al. 2021; Gerasimov et al. 2020b.

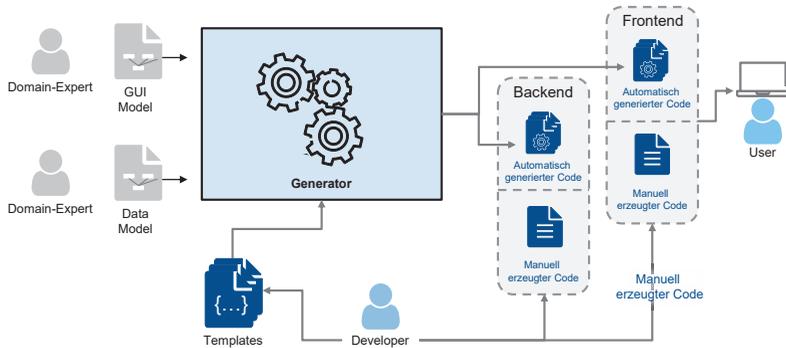


Abbildung 58: MontiGem Framework¹¹⁹

Im Folgenden werden die drei wichtigsten Systeme, der Generator, das Backend und das Frontend, vorgestellt:

Der **Generator** verarbeitet das Datenmodell und erzeugt Programmcode, welcher später im Backend & Frontend verwendet wird. Der Generator baut auf der Language Workbench MontiCore¹²⁰ auf. Er ist größtenteils in Java geschrieben und für die Generierung wird FreeMarker eingesetzt.

Aus jeder Klasse in dem Datenschema wird folgende Java-Funktionalität generiert:

- Datenklasse zum Speichern in Datenbank
- Builder zum Erstellen der Objekte
- Transfer-Klasse für die Übertragung von Daten
- Visitor-Klasse zum einfachen Iterieren über Objekte
- Create / Read / Update / Delete Commands für die API
- Validatoren zum Erkennen von Fehlern und unerlaubten Änderungen

Das **Backend** übernimmt das Speichern und Laden von Daten. Über eine REST-Schnittstelle können mit CRUD-Operatoren die Daten verändert werden. Zudem ist eine Nutzerverwaltung und Authentifizierung integriert.

¹¹⁹ Adam et al. 2018.

¹²⁰ RWTH Aachen University 2021.

Im Backend ist eine effiziente und parallele Bearbeitung der Anfragen wichtig, damit mehrere Nutzer gleichzeitig Arbeiten können. Java-Frameworks wie „Spring“ sind auf eine parallele Abarbeitung ausgelegt und unterstützen somit die Entwicklung.

Das **Frontend** ist eine statische Website, welche Daten von dem Backend lädt und grafisch anzeigt. Weil das Frontend statisch ist, kann es sehr einfach gehostet werden und beansprucht nur wenige Computer-Ressourcen. Als Framework wurde Angular genutzt, sodass eine große Anzahl an vordefinierten Komponenten eingebunden werden kann.

Das Frontend ist nur notwendig, wenn Daten grafisch einsehbar oder änderbar sein sollen. Wenn nur ein Datenspeicher gebraucht wird, kann das Frontend deaktiviert werden.

Aus den generierten MontiGem-Komponenten Backend und Frontend werden Docker-Container gebaut. Docker erleichtert die Verwaltung auf dem Server. Ohne Docker müsste der Server Administrator z.B. manuell die richtige Java Version installieren und selbst den Webserver konfigurieren. Für die Datenbank wird ein PostgreSQL-Docker-Container verwendet. Die Container werden mit Docker-Compose verbunden. Docker-Compose verwaltet hierbei die Kommunikation zwischen den Containern und startet und stoppt die Container in der richtigen Reihenfolge. Durch den beschriebenen Aufbau konnte das Fundament für die Systemarchitektur auf der Softwareseite gelegt werden. Hervorzuheben ist, dass die Integration einer Python-Schnittstelle für die Implementierung der Daten aus den gängigen Entwicklungstools für Windenergieanlagen benötigt wurde und zudem für die Abbildung der Änderungsanfragen aus den agilen Prozessen eine teils automatisierte Lösung in Back- und Frontend gewählt worden ist. Dies könnte bei Übertragung des beschriebenen Vorgehens auf andere Projekte ebenfalls angedacht werden. Diese wird im Folgenden kurz beschrieben.

3.3.3.3 Automatische Generierung von Webseiten

Das reine MontiGem-Framework erzeugt eine minimale Website, welche in weiteren Entwicklungsschritten manuell um Funktionalität erweitert wird. Es wurde eine Erweiterung genutzt, welche für jede Klasse automatisch Seiten zur Suche, Anzeige von Daten und Änderungen von Daten erstellt.

Aus der Klasse „Blade“ wird die Übersichtsseite und die Detailansicht (vgl. Abbildung 59) generiert. Auf der Übersichtsseite werden alle Objekte des Typs „Blade“ mit den wichtigen Attributen angezeigt. Eine Suchfunktion ermöglicht das Filtern. Mit einem Klick auf ein Objekt gelangt man zu der Detailansicht des Objektes. Dort werden alle Attribute angezeigt und können bearbeitet werden.



Abbildung 59: Detailansicht eines Blade

Diese Zusammenführung einer automatisierten Seitenerstellung, welche manuell erweitert werden kann, bietet die Grundlage für Nutzerschnittstellen innerhalb des MontiGem-Frameworks. Folglich können Schnittstellen zu anderen höheren Programmiersprachen, wie beispielsweise Python geschaffen werden.

3.3.3.4 Python-Schnittstelle

Neben der visuellen Anzeige über die Webseite müssen die Daten des Servers auch automatisch zugreifbar sein. Zum Beispiel sollen Simulationen mit den aktuellen Werten gefüllt und die Ergebnisse der Simulation auf dem Server gespeichert werden.

Viele der eingesetzten CAD- und Simulationsprogramme sind kompatibel mit Python. Deshalb wurde eine Python-Schnittstelle zu dem MontiGem Server entwickelt. Die Schnittstelle funktioniert sowohl in Python2 (z.B. IronPython 2.7) als auch in Python3 (z.B. CPython). Die Schnittstelle wurde dokumentiert und mittels Sphinx lässt sich daraus eine HTML-Webseite erstellen.

```
obj = ADDObject_Factory.new(type_name="Blade",
                             data={"model": "VX2-A",
                                    "mass": 30,
                                    "length": 60})

obj.push() // Speichern auf dem MontiGem-Server
```

Abbildung 60: Python-Code für die Erstellung eines neuen „Blade“-Objekts

Neben den Funktionen „Speichern“ und „Laden“ wurde die Python-Schnittstelle in seiner Funktionalität erweitert. So ist der Umgang mit Binärdateien effizient möglich und erste Checks werden auf den Daten aufgerufen, um Fehler frühzeitig zu erkennen.

Das entwickelte Datenschema ist kompatibel zu den legacy-Modellen, was eine Integration von existierenden JSON-Daten ermöglicht. Dies wird benötigt um die Informationen aus den Fachgebieten und deren Systemen zu integrieren, wozu ein Austausch im JSON Format angestrebt wurde. Nachdem umfassend der Aufbau von Nutzerschnittstellen und die somit ermöglichte Zugänglichkeit der Daten dargestellt wurde, wird im Weiteren die Versionierung der Daten beleuchtet.

3.3.3.5 Versionierung

Für die Archivierung müssen Daten über mehrere Jahre zugreifbar sein und dürfen nicht verändert werden. Um dies zu gewährleisten, wurde ein Versionierungssystem in MontiGem eingebaut. Nach einem „release“ können Objekte nicht mehr verändert werden.

Es wurden verschiedene Arten der Versionierung implementiert und evaluiert. Das IntegerVersionable speichert den Versionsstatus als ein Integer Wert. Sobald der Wert größer als 100 ist, sind keine Änderungen an den Attributen erlaubt. Der Integer Wert kann jedoch weiterhin erhöht werden, um z.B. Meta-Eigenschaften wie „outdated“ anzuzeigen. Das BooleanVersionable ist äquivalent, aber kann keine weiteren Meta-Eigenschaften speichern. Zuletzt gibt es das ReleasedVersionable, welches sofort nach dem Erstellen als „released“ gilt und niemals Änderungen erlaubt.

Je nach Anwendungsfall kann eine Versionierung ausgewählt werden. Auf diese Weise können beispielsweise automatisch erstellte Simulationsdaten vor Änderungen geschützt und sofort archiviert werden.

3.3.3.6 Visualisierung von Simulationsdaten

Wie oben beschrieben, können mittels der Python-Schnittstelle Simulationsdaten auf dem MontiGem-Server gespeichert werden. Welche im Rahmen der Implementierung das Webinterface genutzt werden, um die Darstellung der Analysen und der Daten zu ermöglichen.

Bei der graphischen Ausgestaltung ist eine enge Zusammenarbeit zwischen Nutzer und Entwickler notwendig. So sollte der Entwickler die Bedürfnisse des Nutzers gut kennen, damit z.B. relevante Daten gut sichtbar sind. Als Nutzer ist an dieser Stelle der Anwender der Softwareanwendung für den die Visualisierung ein Werkzeug zum einfacheren Verständnisses der Modellinformationen bieten soll. Im Fall für die Entwicklung der Windenergieanlagen wurde ein klares Beschreibungsmodell durch die Fachabteilung benötigt, um relevante Charakteristiken der Produktkomponenten auszuarbeiten und in der Visualisierung für die Nutzer abzubilden. Für die verschiedenen Prozesse und den daran beteiligten Nutzergruppen ergeben sich unterschiedliche Informationsbedarfe und -perspektiven, die mithilfe der Anwendung abgedeckt werden. Daher ist es wichtig ein umfangreiches Beschreibungsmodell als Datengrundlage auszuarbeiten mit dem alle relevanten Informationen abgedeckt werden können. Die Informationen sind daher mit den Fachabteilungen abzustimmen um ein vollständiges Bild der Komponenten aufzubauen. Eine solche umfassende Gestaltung der Komponenten ermöglicht auch Darstellungen konfigurierbar zu machen, um den Nutzer Kontrolle über die Ansicht zu geben. Prototyping und Feedbackrunden sollten dabei zur Verifizierung und zur frühzeitigen Erkennung von Problemen und Optimierungspotenzialen mit bedacht werden.

Als Visualisierungselement wurde auf eine flexible Darstellung von Tabellen gesetzt (vgl. Abbildung 61), um die Daten übersichtlich anzuzeigen. Mithilfe von Filtern und Änderung der Sortierung werden flexible Gestaltungen der Tabellen ermöglicht. Außerdem können Heatmaps über die Datensätze gelegt werden, um einfach Minima und Maxima zu erkennen und ein schneller Überblick über die Daten ermöglicht werden.

A	B	C	D	E
↑ ↓ √	↑ ↓ √	↑ ↓ √	↑ ↓ √	↑ ↓ √
Eintrag 0	0	-2,5	0	0
Eintrag 1	1	-2	1	0
Eintrag 2	2	-1,5	4	0
Eintrag 3	3	-1	9	0
Eintrag 4	4	-0,5	16	0
Eintrag 5	5	0	25	0
Eintrag 6	6	0,5	36	0
Eintrag 7	7	1	49	0
Eintrag 8	8	1,5	64	0
Eintrag 9	9	2	81	0
Eintrag 10	10	2,5	100	0

Kennwerte

Maximum	36
Minimum	9
Summe	86
Anzahl	4
Durchschnitt	21,5
Median	20,5

Abbildung 61: Anzeige einer Tabelle mit Heatmap

Der detaillierte Einblick auf MontiGem als Framework für die Generierung einer Datenbank wird hiermit abgeschlossen. Es zeigen sich die Potenziale der Methode um die Umsetzung eines agilen, datenbasierten Produktentwicklungsprozesses zu ermöglichen. Auf diese Implementierung wird im Folgenden Kapitel 3.4 ausführlich eingegangen.

3.4 Implementierungsphase

Nach der Konzeption sowie der Detaillierung des agilen, datenbasierten Entwicklungsprozesses für Windenergieanlagen muss die Umsetzung und Verankerung im jeweiligen Unternehmen folgen. Zu diesem Zweck ist nachfolgend in Kapitel 3.4.1 beschrieben, wie einerseits die agile, datenbasierte Entwicklung im Unternehmen organisatorisch verankert werden kann. Andererseits definiert Kapitel 3.4.2 einen exemplarischen Implementierungsprozess für die agile, datenbasierte Entwicklung mit dem Ziel, den zuvor adressierten Kulturwandel herbeizuführen.

3.4.1 Organisatorische Verankerung

Ein zentrales Handlungsfeld für die nachhaltige Verankerung der agilen Arbeitsweise in einem Unternehmen ist die Etablierung einer geeigneten Organisation im Unternehmen, die definierten Prozesse und den damit verbundenen Verantwortlichkeitsteilungen ermöglicht und unterstützt. In diesem Sinne erfordert Agilität mehr als reines Methodenwissen, sondern es erfordert auch strukturelle und kulturelle Veränderungen, die nachfolgend beschrieben werden sollen.

Die dafür benötigten strukturellen Veränderungen werden an dieser Stelle in Anlehnung an die Ergebnisse von GUTSCH¹²¹ in drei Handlungsfelder unterteilt deren aktive Gestaltung von Unternehmen bei der Transformation zur agilen Arbeitsweise zu adressieren sind:

- **Rollenprofile:** Definition agiler Rollenprofile mit entsprechenden Aktivitäten, Kompetenzen und Verantwortlichkeiten
- **Projektorganisation:** Etablierung einer agilen Projektarbeitsweise und projektübergreifenden Kollaborationsmöglichkeiten
- **Organisationsstruktur:** Bildung flexibler Strukturen für die Förderung agiler Arbeitsweisen

Um Handlungsbedarf in den drei Feldern Rollenprofile, Projektorganisation und Organisationsstruktur ableiten zu können, empfiehlt sich die Prüfung der aktuellen Positionierung des Unternehmens hinsichtlich des Agilitätsgrades.

Zu diesem Zweck eignet sich beispielsweise die Durchführung einer interviewbasierten Aufnahme des Status-quo, um neben organisatorischen auch kulturelle Aspekte der Gestaltung der Aufbauorganisation

¹²¹ Gutsch et al. 2018.

aufnehmen zu können. Als Interviewpartner sollten Ansprechpersonen aus unterschiedlichen Fachbereichen¹²² des Unternehmens und unterschiedlichen Hierarchieebenen gewählt werden. Zudem hat sich gezeigt, dass eine längere Zugehörigkeit und damit höhere Projekterfahrung eine differenziertere Einschätzung des Handlungsbedarfs ermöglicht.

Die drei aufgeführten Handlungsfelder Rollenprofile, Projektorganisation und Organisationsstruktur sind für die folgenden Ausführungen die strukturgebenden Punkte. In Abbildung 62 sind diese zusammenfassend dargestellt und werden in den nachfolgenden Teilkapiteln näher erläutert.



Abbildung 62: Handlungsfelder bei der Gestaltung der agilen Organisation

3.4.1.1 Definition der Rollenprofile

In der Gestaltung der Aufbauorganisation ist das Element auf der untersten Detaillierungsebene, welches hier betrachtet wird, die Definition der für die agile Produktentwicklung geeigneten Rollenprofile. Nach ABELS ist eine Rolle die „Identifizierung des Zusammenhangs generalisierter Erwartungen“.¹²³ Hinsichtlich der Beschreibung von Rollen lassen sich nach BECKER zwei Perspektiven differenzieren, die operationelle Rolle und die Rolle im Team. Operationelle Rollen sind auf die formale Organisationsstruktur und die darin definierte Arbeitsteilung zurückzuführen und erklären sich aus den unterschiedlichen Qualifikationen der Personen. Die Rollen im Team beschreiben, wie Individuen sich

¹²² Hier sollten insbesondere die direkt von der agilen Transformation betroffenen Entwicklungsbereiche berücksichtigt werden

¹²³ Abels 2019, S.111.

in bestimmten Situationen im Team verhalten und beschreiben die soziale Komponente der Zusammenarbeit und das Teamgefüge aus Führungs- und Gruppenrollen¹²⁴.

Die im Folgenden beschriebenen Rollen besitzen sowohl eine fachliche als auch soziale Dimensionen und werden anhand des AKV-Prinzips beschrieben. Das AKV-Prinzip sieht vor, dass jede Rolle hinsichtlich ihrer Aktivitäten, Kompetenzen und Verantwortung beschrieben wird. Eine Aktivität ist eine „von Menschen [...] ausgeführte Tätigkeit, die Leistung erbringt“¹²⁵ und ist demnach eine Zusammenfassung des Tätigkeitsgebiets der Rollen. Kompetenz ist der „Oberbegriff für Fähigkeiten, Fertigkeiten und Wissen, manchmal auch synonym zu jeweils einer der drei Begriffe“¹²⁶ und beschreibt an dieser Stelle insbesondere den Umfang der Befugnisse der Rollen. Eine Verantwortung ist eine „Verpflichtung und Berechtigung zum Zwecke der Erfüllung einer Aufgabe [...] selbstständig zu handeln“¹²⁷ und definiert an dieser Stelle insbesondere den inhaltlichen Zuständigkeitsbereich einer definierten Rolle. Dem AKV-Prinzip folgend werden Rollen an dieser Stelle insbesondere als inhaltliche und soziale Erwartungen an Individuen interpretiert, die den Rollen Handlungsfreiräume und Handlungsverpflichtungen zuweisen.

Die Definition der Rollen erfolgt durch eine Zusammenführung der Erkenntnisse einer Literaturrecherche sowie der Prüfung der bisher in der Organisation der Anwendungspartner im Forschungsprojekt berücksichtigten Rollen. Als Quellen wurden im Rahmen der Literaturrecherche unter anderem die Arbeiten von BÖHM¹²⁸, HOFERT¹²⁹, KUSAY-MERKLE¹³⁰, LEICHT ET AL.¹³¹ und MEYER UND REHER¹³² berücksichtigt. Auf Basis dieser Literaturrecherche konnte ein grundlegendes Rollengerüst aufgebaut werden, das die Selbstorganisation der agilen Arbeit und die

¹²⁴ Becker und Gieselmann 2018.

¹²⁵ Schwarzer und Krcmar 1995.

¹²⁶ Herold 2011, S.215.

¹²⁷ Suchanek 2021.

¹²⁸ Böhm 2019, S37-40.

¹²⁹ Hofert 2018, S.54.

¹³⁰ Kusay-Merkle 2021, S.11-12.

¹³¹ Leicht-Deobald et al. 2020, S.43.

¹³² Meyer und Reher 2016, S.83-85.

Einhaltung der agilen Prinzipien sicherstellt. Insbesondere ist dabei die klare Trennung einer methodischen Führung des Teams, einer inhaltlichen Führung des Teams und der operativen Entwicklungsarbeit vorgezogen.

Zudem wurden die bisher bestehenden Rollen dahingehend geprüft, welche Aktivitäten, Kompetenzen und Verantwortungen bisher im Rahmen des Entwicklungsprozesses bei der Entwicklung von Windenergieanlagen nötig waren. Grundlage hierfür war die Aufnahme des Wertstroms in der Produktentwicklung und damit einhergehend die Dokumentation von den an der Produktentwicklung beteiligten Rollen. Aus der vorherrschenden Arbeitsstruktur ergab sich eine Differenzierung anhand der Hierarchieebenen. Insgesamt konnten drei Rollenprofile mit jeweils unterschiedlichen Aktivitäten, Kompetenzen und Verantwortungen als übergeordnete Kategorien identifiziert werden. Spezifisch wurden die Rollen mit Managementaufgaben, Rollen mit projektspezifischen Aufgaben und Rollen mit fachlichen Expertenaufgaben unterschieden.

Aus diesen drei Perspektiven ergeben sich die Abbildung 63 dargestellten Rollen auf den drei dargestellten Hierarchieebenen. Zudem wurde im Rahmen der Rollenbeschreibung ebenfalls die für die Windenergieanlagenentwicklung elementare Unterscheidung der System- und Modulebene unterschieden, die sich entsprechend auch in den Rollenbeschreibungen wiederfinden lässt.

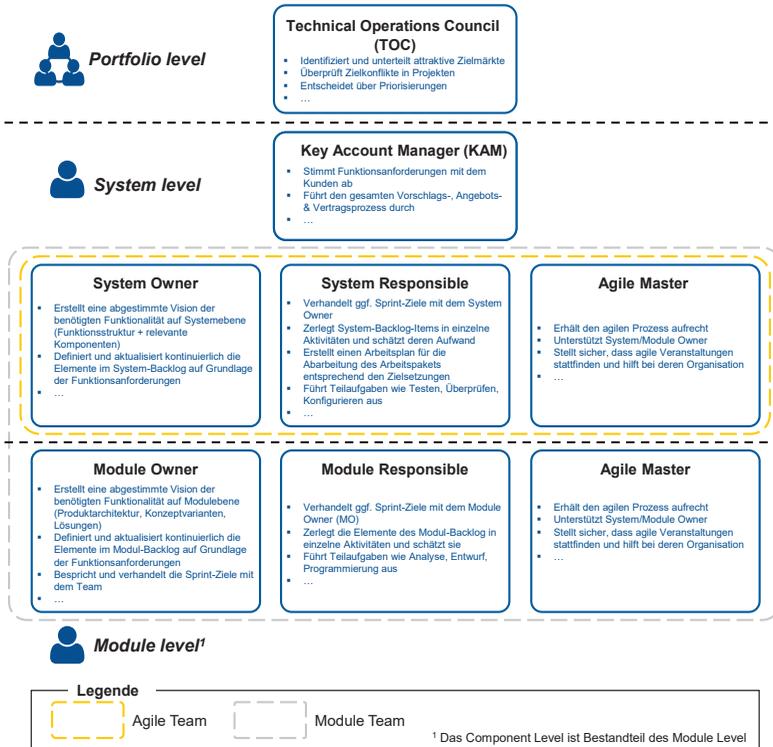


Abbildung 63: Rollen in der agilen Entwicklung von Windenergieanlagen

Nachfolgend werden die Rollen, Technical Operations Council, Key Account Manager, Agile Master, System Owner, System Responsible, Module Owner und Modul Responsible, kurz dargestellt. Für eine ausführliche Beschreibung der nach dem AKV-Prinzip beschriebenen Rollenprofile befindet sich im Anhang B.

Das **Technical Operations Council** ist eine Rolle der strategischen Produkt- und Projektorganisation, die das Gesamtportfolio verantwortet. Zentrale Aktivität ist dabei die Steuerung und Organisation aller Produktentwicklungstätigkeiten im Unternehmen. Demnach obliegt dem Technical Operations Council die Verteilung der für die Bearbeitung notwendigen Ressourcen und damit auch die Priorisierung von Projekten bei Zielkonflikten sowie Schaffung geeigneter Rahmenbedingungen für eine Projektdurchführung.

Der **Key Account Manager** bildet die Kundenschnittstelle in der Produktentwicklung und zentralisiert insbesondere die Kommunikation. Durch diese Rolle soll zudem eine vereinfachte Einbindung des Kunden in den Entwicklungsprozess über die Stellvertretungsfunktion des Key Account Managers erreicht werden, der Funktionsanforderungen abstimmt und diese in die Steuerung der Entwicklungstätigkeiten einfließen lässt.

Der **System Owner** trägt die inhaltliche Projektverantwortung und ist im Rahmen des Projekts insbesondere für die Einsteuerung von Anforderungen und Fragestellungen in den Entwicklungsprozess zuständig. Der System Owner repräsentiert damit alle für die Anlage relevanten Stakeholder sowie deren Prioritäten bei der Weiterentwicklung, harmonisiert diese, überträgt diese in den System Backlog des Projekts und steuert sie schließlich in das Entwicklungsprojekt ein.

Der **System Responsible** übernimmt die ingenieurstechnische Sicht und führt Entwicklungstätigkeiten auf System- bzw. Anlagenebene durch. Dabei liegt der Hauptfokus des System Responsible in der operativen Sprintdurchführung und der Sicherstellung erfolgreicher Inkremente auf System- bzw. Anlagenebene. Insbesondere die Integration der Modulumfangs am Sprintende zur Entwicklung und Bereitstellung der definierten Inkremente ist dabei zu beachten.

Der **Module Owner** verantwortet projektübergreifend einen definierten Produktumfang (bspw. das Rotorblatt oder den mechanischen Triebstrang) und dessen markt- und kundenorientierte Weiterentwicklung. Der Module Owner ist daher für die Definition der Sprintziele für das verantwortete Modul auf Basis der Sprintziele für das Gesamtsystem verantwortlich.

Der **Module Responsible** führt die operative Entwicklungsarbeit auf Modulebene durch und ist damit einem spezifischen Modul (bspw. dem Rotorblatt) zugeordnet. Dabei zerlegt er die durch System und Module Owner definierten Sprintziele in Aktivitäten und führt diese während der Sprintdurchführung eigenverantwortlich aus. Am Ende des Sprints stellt der Module Responsible ein testfähiges Inkrement des Moduls zur Verfügung.

Der **Agile Master** unterstützt die Prozessdurchführung und stellt die Einhaltung der agilen Arbeitsweise und Prinzipien sicher. Dazu gehört

insbesondere die Leitung der Regeltermine während der Sprintdurchführung. Der Agile Master ist inhaltlich nicht in die Entwicklungsarbeit eingebunden.

Die dargestellte Rollenstruktur orientiert sich dabei an den bei den Unternehmen aus der Windenergiebranche zumeist vorherrschenden OEM-Strukturen. Zudem wurde eine verkürzte Rollenstruktur entwickelt, die insbesondere für kleinere Unternehmen gültig ist, die weniger arbeitsteilig entwickeln und dementsprechend auf eine integrierte Struktur angewiesen sind. Daher wurden die Rollen Technical Operations Council, Key Account Manager, System Owner und Module Owner zum Project Owner und System Responsible und Module Responsible zum Fields Responsible zusammengefasst. Dementsprechend ergeben sich zwei neue Rollenprofile:

Der **Project Owner** übernimmt im Sinne des System und Module Owners die Verantwortung für die Inhalte der Produktentwicklung und den Funktionsumfang, indem er die Ziele je Sprint definiert und im Sinne der Stakeholder priorisiert. Zudem hat der Project Owner direkten Kundenkontakt und stimmt sich mit anderen Project Owner bezüglich der Priorisierung der Projekte untereinander bei der Ressourcenallokation ab.

Der **Fields Responsible** führt die Entwicklungstätigkeiten im Projekt durch, indem aus den Zielen Entwicklungstätigkeiten abgeleitet und eigenverantwortlich ausgeführt werden. Am Ende des Sprints wird durch den Fields Responsible ein Produkt- bzw. Projektinkrement zur Verfügung gestellt, welches hinsichtlich der Erfüllung der Sprintziele geprüft werden kann.

Für eine systematische Überführung der bestehenden Rollenstruktur in die neue Rollenstruktur muss ein strukturierter Vergleich der bisher in den Entwicklungsprozess eingebunden Personen und der Anforderungen der definierten Rollen auf Basis des AKV-Prinzips durchgeführt werden. Zu diesem Zweck wird je Dimension ein paarweiser Vergleich der Personen und den neuen Rollen durchgeführt und bewertet, ob die Person hinsichtlich der Aktivitäten, Kompetenzen oder Verantwortlichkeiten geeignet ist, die jeweilige Rolle auszufüllen. Eine entsprechende Matrix, die den paarweisen Vergleich beispielhaft darstellt, findet sich in Abbildung 64. Eine Zuordnung der Rollen erfolgt auf Basis der Bewertung in allen drei Dimensionen, indem die jeweils geeignete Person ausgewählt wird.

Vergleich des AKV-Rollenprofils		Rollenkonzept in der agilen Entwicklung						
		TOC	Key Account Manager	Agile Master	System Owner	System Responsible	Module Owner	Module Responsible
Bestehende Rollen	Abteilungsleiter	3	1	1	2	1	2	1
	Portfolio-manager	2	1	1	3	1	3	1
	Projektmanager	1	1	3	3	1	3	1
	Ingenieur	1	1	2	1	3	3	1
	...							

Legende

1 = Keine Übereinstimmung; 2 = teilweise Übereinstimmung; 3 = große Übereinstimmung; ○ = Mögliche Zuordnung

Abbildung 64: Paarweiser Vergleich für die Zuordnung von Personen zu Rollen

An dieser Stelle sei anzumerken, dass die vorgestellten Rollenprofile sich an der Entwicklung von Windenergieanlagen orientieren und für einen praktikablen Einsatz in diesem Umfeld gestaltet worden sind. Dennoch können Abweichungen von den vorgeschlagenen Rollenprofilen anwendungsspezifisch sinnvoll sein.

In der Praxis führen die dargestellten Rollen den in den Kapiteln 0 und 3.3 Produktentwicklungsprozess durch. Die Verantwortlichkeit der Rollen für Aktivitäten während des Produktentwicklungsprozesses sind im Anhang C dargestellt. Aufbauend auf den vorgestellten Rollenprofilen soll im Folgenden die Auswahl der Projektorganisation im Sinne einer agilen Zusammenarbeit detailliert werden.

3.4.1.2 Gestaltung einer agilen Projektorganisation

Die Wahl einer geeigneten Projektorganisation definiert in hohem Maße, wie ein Team entsprechend den definierten Strukturen agieren kann und muss. Dabei ist die Wahl der richtigen Projektorganisation grundlegend von externen Rahmenbedingungen der Branche des jeweiligen Unternehmens abhängig. Durch die Wechselbeziehung zwischen der Umwelt und den Kundenaufträgen mit den vorhandenen Projekten und Produkten ist auch das Projektteam durch die sich daraus ergebenden Strukturen im Unternehmen geprägt. Unternehmen aus der Windenergiebranche werden dennoch oftmals durch starre Strukturen charakterisiert, die typischerweise für Unternehmen auf einem durch

den Verkäufer dominierten Markt vorherrschen, da sie hohe Effizienz bei definierten Anforderungen versprechen. Die Rahmenbedingungen der Windenergiebranche werden von den Anwendungspartnern eher als Käufermarkt charakterisiert, auf dem Anbieterunternehmen sich in ihrer Auftragsabwicklung stark am Kunden orientieren müssen. Von Entwicklungsunternehmen der Branche wird demnach eine hohe Flexibilität, auch hinsichtlich der Individualisierung und kurzfristigen Änderung von Kundenanforderungen als auch der Innovationskraft ihrer Produkte gefordert. In Abbildung 65 ist dargestellt, inwiefern sich agile, hybride oder plangetriebene Projektorganisationen hinsichtlich der Erfüllung der externen Rahmenbedingungen der Windenergiebranche eignen. Der Kriterien und der Erfüllungsgrad dieser wurde anhand einer Literaturrecherche aufgestellt und mithilfe der beteiligten Partner verifiziert. Durch die variable Umsetzung der hybriden Projektorganisation wird für diese keine finale Eignung zu den Kriterien angegeben, sondern ein Intervall in dem sich diese je nach Ausprägung bewegen kann. Anhand der Auswertung kann festgehalten werden, dass typische Projekte in der Windenergieanlagenentwicklung agil oder hybrid durchzuführen sind. Bei anderen Projektrahmenbedingungen kann sich wiederum anwendungsfallspezifisch auch eine plangetriebene Herangehensweise eignen.

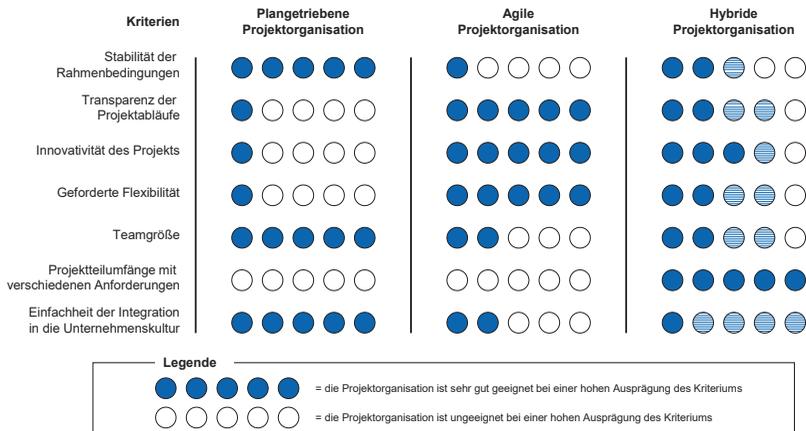


Abbildung 65: Gegenüberstellung der Bewertung von plangetriebener, agiler und hybrider Projektorganisation¹³³

¹³³ Boehm und Turner 2003; Klein 2016; Aldushyna und Engstler 2015.

Die agile Projektorganisation definiert insbesondere die Zusammenarbeit der im Produktentwicklungsprozess beteiligten Personen. Zunächst wird daher die Zusammensetzung eines Projektteams für agile und hybride Projektdurchführungen und die darin vorgesehene Aufgabenteilung eingegangen.

Zur Erklärung der Teamzusammensetzung wird zwischen System und Module Level unterschieden. Auf System Level besteht das Team aus einem System Owner, einem Key Account Manager, System Responsibles und je im Projektumfang berücksichtigtem Modul einem Module Owner. Das Entwicklungsteam auf System Level hat dabei weniger eine entwickelnde als vielmehr eine steuernde Funktion. Der Key Account Manager verantwortet den kontinuierlichen Kundenkontakt im Verlauf der Systementwicklung und vertritt dementsprechend den Kunden. Der System Owner ist für die Gesamtanlage verantwortlich und steuert deren Entwicklung inhaltlich durch die iterative Priorisierung des Backlogs. Eine direkte Weisungsbefugnis für das Projektteam hat der System Owner jedoch nicht. Die System Responsibles führen alle Entwicklungen auf Systemebene, wie beispielsweise die Bereitstellung modulbezogener Daten, die Bauraumaufteilung oder die Integration der Systeminkremente aus den Modulinkrementen sowie deren Validierung durch. Die Module Owner bilden die Schnittstelle des Entwicklungsteams auf Systemebene zu den Entwicklungsteams auf Modulebene, da sie gemeinsam mit dem System Owner die priorisierten Sprintziele auf Systemeben in Sprintziele für ihre jeweiligen Module übersetzen.

Auf Modulebene besteht ein Projektteam aus jeweils einem Module Owner und je nach Arbeitsumfang vier bis acht Module Responsibles. Der Module Owner verwaltet dabei den modulspezifischen Backlog und definiert auf Basis der Anforderungen des System Owner modulspezifische Sprintziele. Es bietet sich an, dass ein Module Owner das Modulteam mehrerer Projekte verantwortet, um eine projektübergreifende inhaltliche Weiterentwicklung der Module nachhaltig sicherzustellen. Auch in diesem Fall ist der Module Owner gegenüber den Module Responsibles nicht weisungsbefugt, sondern definiert lediglich das inhaltliche Ziel, welches die Module Responsibles durch ihre Entwicklungstätigkeiten eigenverantwortlich umsetzen.

Zudem gehört zu jedem Projektteam ein Agile Master, der die Regeltermine innerhalb des Projekts leitet und das Projekt mit seiner Vorgehens-

und Methodenkompetenz der agilen Produktentwicklung begleitet, allerdings inhaltlich nicht in das Projekt involviert ist. Zudem steht der Agile Master als Schlichtungspartei für Probleme zwischen den Personen im Projektteam zur Verfügung. Ein Agile Master kann zudem mehrere Teams als Agile Master begleiten oder sogar in anderen Teams in anderer Rolle operativ mitarbeiten.

Eine Übersicht über die Zusammensetzung eines agilen Projektteams mit dem Team auf Systemebene und beispielhaften Teams auf Modulebene ist in Abbildung 66 dargestellt.

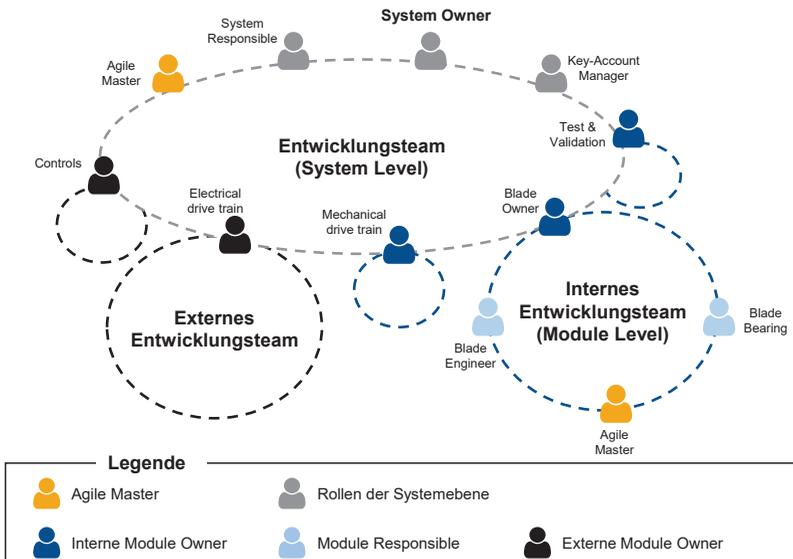


Abbildung 66: Übersicht der Rollen in agilen Projektteams

Neben der internen Projektorganisation muss auch die projektübergreifende Organisation und Koordination sichergestellt werden. Zu diesem Zweck muss ein projektübergreifendes Entwicklungsteam definiert werden, welches durch das oben beschriebene Technical Operations Council geleitet wird und sich idealerweise in einem zwei- bis vierwöchigen Rhythmus¹³⁴ abstimmt. Das projektübergreifende Team hat insbesondere koordinierende Aufgaben. Dazu gehört die Identifizierung und Auflösung von Konflikten bei der Ressourcenallokation zwischen

¹³⁴ Der ideale Rhythmus der Abstimmung ist abhängig von den für die Projekte gewählten Sprintlängen und sollte der kürzesten Sprintlänge angepasst sein, um sicherzustellen, dass pro Sprintdurchlauf eine Abstimmung des übergeordneten Projektteams stattfindet.

der Entwicklungsteams ohne Systemlevel¹³⁵ sowie die Integration von Entwicklungspartnern und Kunden im Unternehmensnetzwerk.

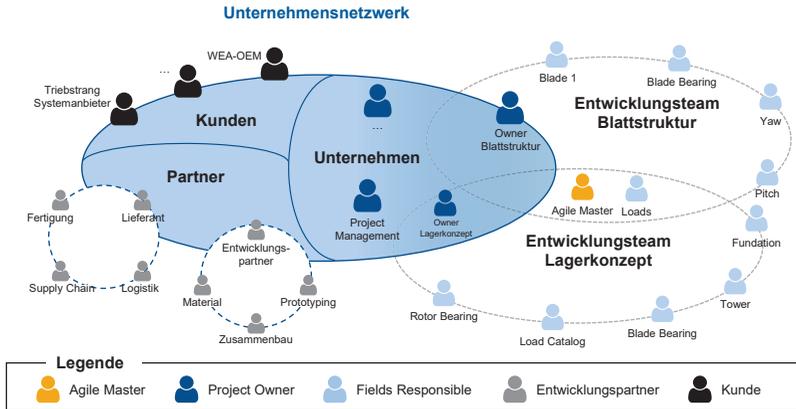


Abbildung 68: Projektorganisation für kleinere WEA-Unternehmen

3.4.1.3 Gestaltung einer agilen Aufbauorganisation für die Produktentwicklung

Eine Aufbauorganisation ist das „statische System der organisatorischen Einheiten einer Unternehmung, das die Zuständigkeiten für die arbeitsteilige Erfüllung der Unternehmensaufgaben regelt“¹³⁶. Dabei kann zwischen der Primärorganisation als Grundgerüst, die permanent existiert und den Betrieb sicherstellt, und der Sekundärorganisation, die die Primärorganisation temporär beispielsweise für die Durchführung von Projekten überlagert, unterschieden werden¹³⁷.

Bei der Wahl der richtigen Aufbauorganisation, insbesondere innerhalb der Produktentwicklung, muss berücksichtigt werden, wie sich das zugrundeliegende Unternehmen in dem in Abbildung 69 dargestellten Spannungsfeld zwischen einer plangetriebenen und agilen Projektdurchführung aufstellen möchte. Organisationsformen, die sich für plangetriebene Projekte eignen, weisen eine hohe Effizienz auf, wenn das Projekt einen hohen Standardisierungsgrad besitzt und der Lösungsweg weitgehend vorgegeben ist. Organisationsformen für die agile Projektdurchführung besitzen dahingegen die nötige Flexibilität, um insbesondere

¹³⁵ Die Empfehlung der optimalen Teamgröße von vier bis acht Fields Responsibles zzgl. des Project Owner und Agile Master bleibt bestehen

¹³⁶ Schewe 2018.

¹³⁷ Schlick et al. 2018; Mattmüller und Tunder 2012.

Projekte mit hohem Innovationsgrad umsetzen zu können. Beispielhaft sind an dieser Stelle die Linienorganisation, Matrixorganisation, duale Organisation und die temporären Projektteams eingeordnet.



Abbildung 69: Auswahlmöglichkeit der Organisationsform

Aufgrund der in der Windenergiebranche vorliegenden Rahmenbedingungen eignet sich insbesondere die in Abbildung 70 dargestellte duale Organisation, um agile Produktentwicklung im Unternehmen zu ermöglichen. In der dualen Organisation existiert je Mitarbeitendem eine disziplinarisch vorgesetzte Führungsperson sowie inhaltlich vorgesetzte Projektleitung. In der dualen Organisation ist grundsätzlich die Mitarbeit in mehreren Projekten möglich. Sie vereint auf diese Weise die Vorteile der Linienorganisation durch die hohe Effizienz im linienbasierten Tagesgeschäft sowie der Flexibilität der temporären Projektteams für Projekte mit hohem Innovationscharakter. Die duale Organisation eignet sich damit insbesondere für hybride Projektdurchführungen.

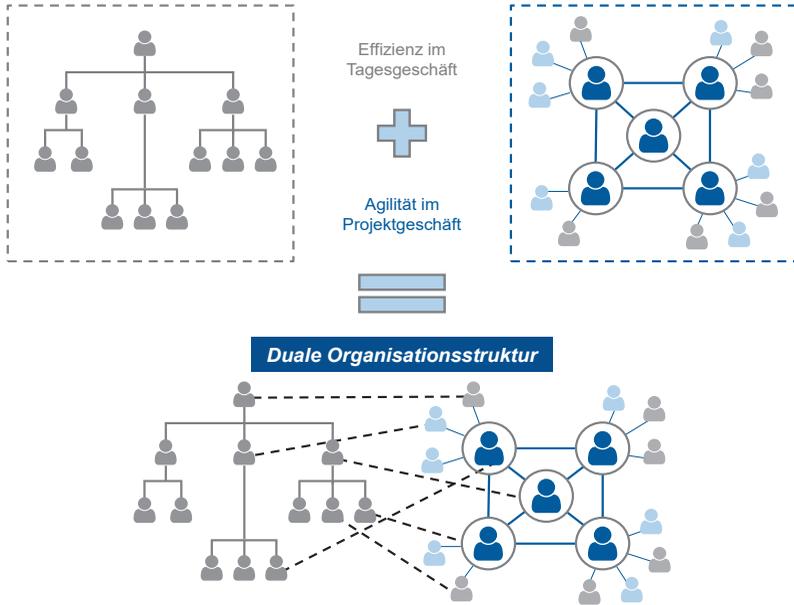


Abbildung 70: Konzept einer dualen Organisation

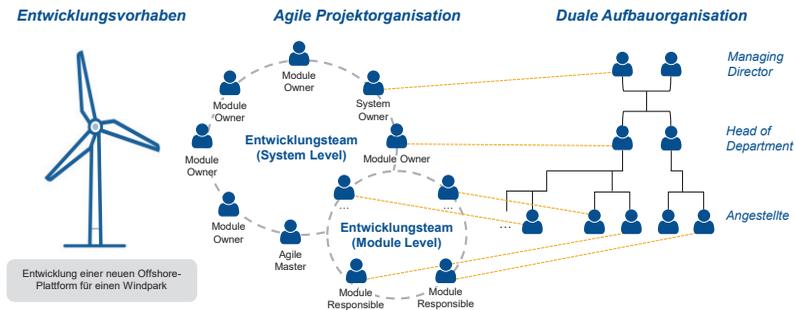


Abbildung 71: Umsetzung der dualen Organisation mit der beschriebenen agilen Projektorganisation

Abschließend soll die Umsetzung der dualen Organisation für die in Kapitel 3.4.1.2 dargestellte Projektorganisation in Abbildung 71 skizziert werden. Die Besetzung des Projektteams bedient sich dabei an den unterschiedlichen Ebenen der Aufbauorganisation entsprechend der fachlichen Qualifikation der jeweiligen Personen. Eine disziplinarische Führung mündet dabei nicht unbedingt in einer fachlichen Führung inner-

halb des jeweiligen Projekts. Die Rollenzuweisung ist zudem projektspezifisch, kann aber auch projektübergreifend verstanden werden. Insbesondere die Rolle des Module Owner eignet sich für die Durchführung in mehreren Projekten, um die inhaltliche Weiterentwicklung eines bestimmten Moduls übergreifend im Unternehmen verankern zu können.

3.4.2 Gestaltung eines Implementierungsprozesses

Nachdem in den vorigen Kapiteln insbesondere das Zielbild für eine agile Produktentwicklung umfassend beschrieben wurde, soll nachfolgend in Kapitel 3.4.2.1 detailliert dargestellt werden, wie der Prozess zur Implementierung der agilen, datenbasierten Produktentwicklung ausgestaltet werden kann und welche Erfolgsfaktoren hierbei im Rahmen des Forschungsvorhabens identifiziert wurden. Dazu werden zunächst mögliche Aktivitäten für den im Unternehmen notwendigen Change-Prozess vorgestellt. Zudem wird in Kapitel 3.4.2.2 dargestellt, welchen kulturellen Wandel innerhalb des Unternehmens die agile Entwicklung voraussetzt.

3.4.2.1 Phasen und Aktivitäten der Implementierung

Die Umsetzung einer agilen Arbeitsweise im Arbeitsalltag des Unternehmens bedeutet im Gegensatz zur plangetriebenen Arbeitsweise eine umfangreiche Umstellung der Arbeitsabläufe und der dazugehörigen Organisationsstrukturen. Aufgrund der hohen Intensität der bevorstehenden Veränderungen ist die Planung und Durchführung eines systematischen Change-Prozesses notwendig. Dieser Prozess folgt im Optimalfall der Logik einer stufenweisen Implementierung, um insbesondere die Intensität der Veränderungen zu Beginn des Prozesses (siehe Abbildung 72) aktiv zu steuern. Der zugrundeliegende Prozess gliedert sich hierbei in fünf Phasen, die nachfolgend jeweils kurz beschrieben werden.

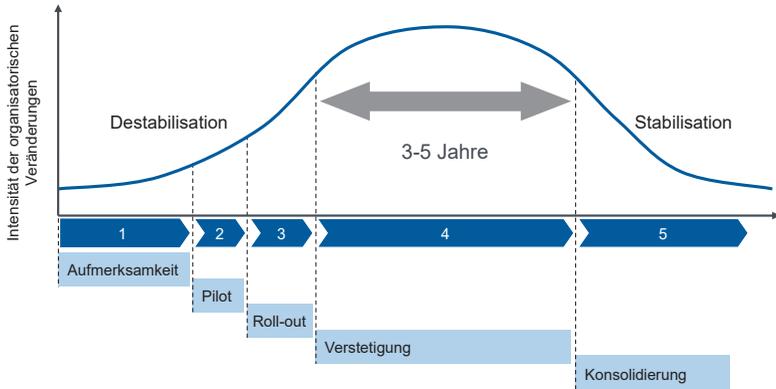


Abbildung 72: Prozess für die Implementierung und Intensität der verursachten Veränderungen

Der definierte Implementierungsprozess startet mit der Phase „**Aufmerksamkeit**“, im Rahmen derer zunächst Schwachpunkte in der bisherigen Prozessdurchführung und die Vorteile einer agilen, datenbasierten Produktentwicklung identifiziert und transparent gemacht werden müssen. Hierzu eignen sich insbesondere die in Kapitel 3.1 vorgestellten Methoden zur Analyse des Ist-Zustands.

Die darauffolgende Phase „**Pilot**“ widmet sich der initialen Kommunikation der bevorstehenden Veränderung und auch der Initiierung eines ersten prototypischen Projekts. Hierbei sollte insbesondere die Ausarbeitung eines grundlegenden Prozess- und Organisationsverständnisses für die agile Produktentwicklung definiert und erprobt werden, um einerseits im Sinnbild eines Leuchtturms Aufmerksamkeit für das Thema zu erregen und andererseits unternehmensspezifische Adaptionsbedarfe identifizieren zu können.

Die dritte Phase „**Roll-Out**“ adressiert die nach der initialen Demonstration aufkommenden Zweifel innerhalb der Organisation. Aufgrund des prototypischen Charakters der frühen Phase muss damit gerechnet werden, dass sich nicht alle Vorteile der agilen Produktentwicklung sofort einstellen lassen und erste Widerstände aufkommen. Ziel der dritten Phase ist die Beseitigung der eventuellen Bedenken gegenüber der agilen Produktentwicklung durch Kommunikation und Adaption, sodass die Mitglieder der Organisation aktiv am Veränderungsprozess partizipieren.

Die vierte Phase „**Verstetigung**“ bildet den Schwerpunkt des Implementierungsprozesses und ist zugleich kritisch für dessen Erfolg. Nachdem mit hohem Aufwand erste kleine Erfolge erreicht wurden, besteht die Gefahr, dass über einen längeren Zeitraum (üblicherweise kann diese Phase 3 bis 5 Jahre dauern) durch den großen Veränderungsgrad bei der Rückkehr ins Alltagsgeschäft die Frustration bei den Mitarbeitenden wächst. Aus diesem Grund muss an dieser Stelle die nachhaltige Verankerung der agilen Produktentwicklung in der Organisation und Kultur erreicht werden, um dieser Entwicklung entgegen zu wirken. Beispielsweise durch die kontinuierliche Kommunikation der Vorteile und Gründe weshalb die agile Vorgehensweise eingesetzt wird und dem Vorleben sowie Belohnen eines agile Vorgehens durch das Management, kann die Unternehmenskultur stetig angepasst werden. Abschließend wird in der fünften Phase „**Konsolidierung**“ der Implementierungsprozess abgeschlossen. Dazu gehört einerseits das Aufzeigen der erreichten Potenziale und andererseits die transparente Aufarbeitung der Lessons-Learned, um im Sinne der agilen Produktentwicklung einen kontinuierlichen Verbesserungsprozess anzustoßen. Über den gesamten Implementierungsprozess ist Kommunikation ein zentrales Instrument für die Überzeugung der Mitarbeitenden. Jedoch ist die Auswahl der richtigen Kommunikationsmethode nicht trivial, sondern bedarf der Berücksichtigung der gewünschten Reichweite und Wirkungstiefe. Abbildung 73 stellt mögliche Kommunikationsmethoden zur Auswahl auf Basis der gewünschten Wirkungstiefe und der Reichweite in einer Matrix dar.

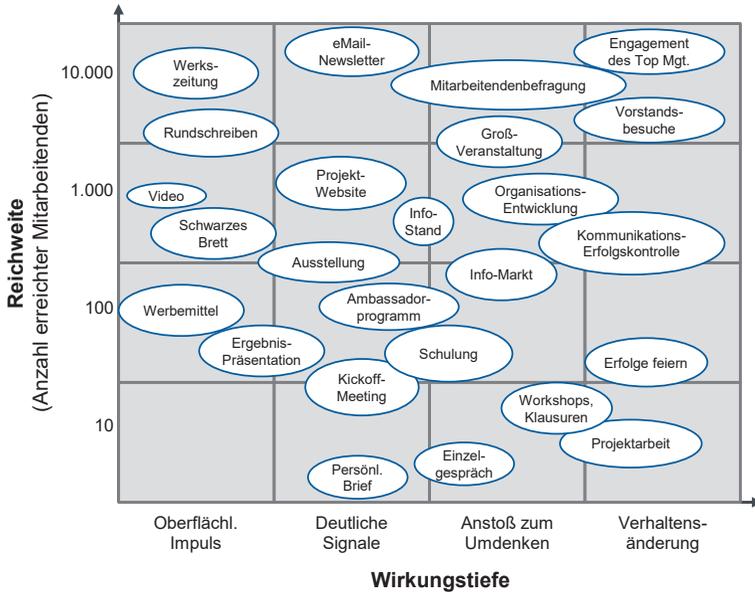


Abbildung 73: Matrix zur Auswahl von Kommunikationsmethoden

3.4.2.2 Gestaltung des kulturellen Wandels

Neben einem organisationalen Wandel, der durch einen Implementierungsprozess gesteuert wird, muss auch ein kultureller Wandel bei der Einführung der agilen Produktentwicklung berücksichtigt werden. In diesem Kontext sollen Abwehrmechanismen, die bei einem derartigen Veränderungsprozess bei Mitarbeitenden auftreten könnten, in Abbildung 74 zusammengefasst werden, um diesen bei der Planung der kulturellen Veränderung in einem Unternehmen proaktiv begegnen zu können.



Abbildung 74: Abwehrmechanismen von Mitarbeitenden bei Veränderungen

Im Rahmen des Forschungsvorhabens sowie in begleitenden Aktivitäten wurden anhand von Beobachtungen und abschließenden Interviews die Eigenschaften der Organisation und deren Auswirkung auf die Umsetzung einer agilen Produktentwicklung festgehalten. Hinsichtlich des kulturellen Wandels konnten drei Eigenschaften identifiziert werden, deren Umsetzung für die Verankerung einer agilen Arbeitsweise benötigt worden sind:

- Neues Teamverständnis
- Neues Führungsverständnis
- Neues Tätigkeitsverständnis

Beim kulturellen Wandel muss ein neues **Teamverständnis** ggü. der plangetriebenen Produktentwicklung etabliert werden. Insbesondere zeichnet sich das Team durch eine cross-funktionale Zusammenarbeit aus. Entgegen dem bisherigen stark zwischen den Disziplinen (Mechanik, Elektronik und Software) getrennten Entwicklungsprozess müssen Teams für die agile Produktentwicklung mit einem diversifizierten Kompetenzportfolio ausgestattet sein, sodass die anfallenden Entwicklungsaufgaben eigenständig durch das Team erarbeitet werden können. Das Team übernimmt in diesem Kontext Verantwortung für die jeweiligen Arbeitsaufgaben und definiert die Lösung eigenständig ohne Rücksprache mit der anforderungsgebenden Instanz während eines Sprints. Auf diese Weise wird die notwendige Eigenständigkeit und das nötige Verantwortungsbewusstsein des Teams für die Entwicklung unterstützt. Zudem muss sich im Kontext der agilen Entwicklung auch das **Führungsverständnis** ändern. Führung darf im agilen Kontext nicht als rein

disziplinarische oder fachliche Führung verstanden werden, sondern muss insbesondere das entwickelnde Team während den Sprints befähigen, die gestellten Aufgaben eigenständig und insbesondere ohne Störungen von außen zu erfüllen. Das inkludiert auch ein inhärentes Vertrauen in das Team und die vom Team erarbeiteten Lösungen, da die Führungskraft während eines Sprints nicht in die Arbeit eingreifen sollte. Das Team benötigt aus diesem Grund einen Umfang an Planungsreserve, um unerwartete Störungen oder Komplikationen eigenständig ausgleichen zu können.

Auch das **Tätigkeitsverständnis** muss sich im Kontext agiler Arbeit weiterentwickeln und stärker die Adaptionfähigkeit betonen. Weil klassische plangetriebene Entwicklungsprozesse an definierten Meilensteine vollständige Anforderungslisten, Modelle oder Zeichnungen benötigen, hat sich dieser Wunsch nach Vollständigkeit auch im Denken der am Prozess beteiligten Personen gefestigt. Diese Vollständigkeitsparanoia sollten müssen für eine agile Arbeitsweise überwunden werden, die durch kurze Entwicklungsiterationen die Fähigkeit besitzt, auch mit unvollständigen Informationen umzugehen, indem beispielsweise sinnvolle Annahmen getroffen werden, die am Ende einer Iteration verifiziert oder falsifiziert werden können. Dazu gehört zudem die Adaptionfähigkeit der am Prozess beteiligten Personen sowie eine offene Fehlerkultur in der gesamten Organisation. Während in der plangetriebenen Produktentwicklung ein hoher Zeitanteil für die Planung aufgewendet wird, um die Zahl der Fehler zu minimieren, können Fehler im agilen Entwicklungsprozess genutzt werden, um am Iterationsende neue Erkenntnisse zu gewinnen.

Um diese neuen Verständnisse aufzubauen und zu verankern sind im Rahmen dieses Projekts unterschiedliche Workshops und die Definition einer Kommunikationsstrategie zum Einsatz gekommen. In wie fern die Veränderungen auch nachhaltig bleiben wird sich allerdings erst in Zukunft retrospektiv festhalten lassen. Die Art und Weise mit der der kulturelle Wandel angestoßen wird, ist sicherlich unternehmensspezifisch zu bewerten. Als grundlegendes Vorgehen ist es sinnvoll einerseits ein Bewusstsein für den Bedarf eines kulturellen Wandels im Top-Management zu schaffen und andererseits eine Strategie zu überlegen mit der die Veränderungen transparent aufgezeigt werden und auf über die Zeit auch präsent gehalten werden kann. Auch kann ein dediziertes

Leuchtturmprojekt zu Beginn der Veränderung für Verständnis sorgen und Ansprechpersonen für Fragen und Probleme aufbauen.

Den insbesondere Veränderungen von Prioritäten und der Umgang mit der Umverteilung von Verantwortlichkeiten, die es beim Übergang zur agilen Vorgehensweise benötigt, gilt es durch positive Verstärkung in der Unternehmenskultur zu befähigen.

Dadurch lässt sich eine Verstärkung, sowie eine schlussendliche Konsolidierung der prozessualen Veränderung und eine angepasste Unternehmenskultur effektiv erreichen.

4 Zusammenfassung

In den vorangegangenen Kapiteln wurde ein Vorgehensmodell zur Implementierung eines agilen, datenbasierten Entwicklungsprozesses für Unternehmen der Windenergiebranche beschrieben.

Aktuelle Herausforderungen fordern von Unternehmen mehr Flexibilität und Anpassungsfähigkeit an sich ändernde Rahmenbedingungen in der Produktentwicklung. Dabei stoßen die gängigen plangetriebene Produktentwicklungsprozesse aufgrund ihres deterministischen Charakters und der daraus folgenden Inflexibilität an ihre Grenzen. Stattdessen könnten agile Entwicklungsprozesse aus der Softwareentwicklung durch Unternehmen der Windenergiebranche adaptiert werden, um durch eine iterative Vorgehensweise die Anpassungsfähigkeit des Prozesses zu verbessern. Allerdings ist aufgrund des geänderten Objektbereichs keine direkte Übertragbarkeit gewährleistet. Zu diesem Zweck wurde im Rahmen des Forschungsvorhabens ein Vorgehensmodell entwickelt, das von Unternehmen zu diesem Zweck adaptiert werden kann.

Die Agilisierung der Entwicklungsprozesse geht dabei mit einer Zentralisierung der Informationsablage innerhalb der Produktentwicklung einher. Da agile Entwicklungsprozesse im Gegensatz zu plangetriebene Prozessen eine höhere Interaktion und auch Informationsaustausch zwischen den teilnehmenden Personen erfordern, scheitern unsystematische Informationsverwaltungen aufgrund fehlender Transparenz und hoher Fehleranfälligkeit durch uneindeutige Ablageorte oder Unklarheit über den aktuellen Entwicklungsstand. Daher muss der agile Entwicklungsprozess datenbasiert sein, um einen effektiven Informationsaustausch innerhalb der Entwicklungsiterationen zu ermöglichen.

Das erarbeitete Vorgehensmodell hat vier Schritte. Zunächst findet eine Aufnahme des Status-quo des jeweiligen Unternehmens statt. Im zweiten Schritt erfolgt die Konzeption des agilen, datenbasierten Entwicklungsprozesses hinsichtlich der prozessualen Ausrichtung und der Struktur des Informationsmodells. Der dritte Schritt detailliert einerseits den Prozess durch eine methodische Unterstützung und andererseits das den Informationsaustausch unterstützende Datenmodell, dessen Aufbau sowie Benutzung. Im letzten Schritt wird eine Verankerung der agilen Produktentwicklung im Unternehmen angestrebt. Diese erfolgt

durch die Definition einer agilen Aufbauorganisation mit Rollen sowie eines Change-Prozess.

Im Rahmen des Forschungsvorhabens wurde der dargestellte Entwicklungsprozess gemeinsam mit dem Anwendungsunternehmen in unterschiedlichen Kontextsituationen erprobt und zur Verbesserung der Praktikabilität adaptiert. So wurden unter anderem Vorschläge für die Anpassung an kleinere Unternehmen entwickelt. Auch die organisatorische Verankerung oder der Change-Prozess wurden praxisnah vor dem Hintergrund der Unternehmenssituation im Praxiskontext entwickelt.

5 Literaturverzeichnis

- Abels, Heinz (2019): Rolle: Was vom Individuum erwartet wird und wie es damit umgeht. In: Heinz Abels (Hg.): Einführung in die Soziologie. Wiesbaden: Springer Fachmedien Wiesbaden, S. 105–134.
- Adam, Kai; Netz, Lukas Stephan Michael; Varga, Simon; Michael, Judith; Heuser, Patricia; Letmathe, Peter (2018): Model-Based Generation of Enterprise Information Systems.
- Albers, Albert; Scherer, Helmut; Bursac, Nikola; Rachenkova, Galina (2015): Model Based Systems Engineering in Construction Kit Development – Two Case Studies. In: *Procedia CIRP* 36, S. 129–134. DOI: 10.1016/j.procir.2015.01.044.
- Aldushyna, Anna; Engstler, Martin (2015): Erfolgsfaktoren bei der Umsetzung hybrider Projekte - Ergebnisse einer Befragung und praktische Empfehlungen zur Umsetzung. In: Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser, Martin Mikusz und Alexander Volland (Hg.): Projektmanagement und Vorgehensmodelle 2015. Bonn: Gesellschaft für Informatik e.V, S. 39–54.
- Alpar, Paul; Alt, Rainer; Bensberg, Frank; Weimann, Peter (2019): Anwendungsorientierte Wirtschaftsinformatik. Strategische Planung, Entwicklung und Nutzung von Informationssystemen. 9. überarbeitete und aktualisierte Auflage. Wiesbaden: [Heidelberg] (Lehrbuch).
- Alt, Oliver (2012): Modellbasierte Systementwicklung mit SysML. München: Carl Hanser Fachbuchverlag. Online verfügbar unter <http://www.hanser-elibrary.com/doi/book/10.3139/9783446431270>.
- Alur, Rajeev (2015): Principles of cyber-physical systems. Cambridge, Massachusetts, London, England: The MIT Press.
- Anderson, David J. (2010): Kanban. Successful evolutionary change for your technology business. Sequim, Wash.: Blue Hole Press.
- Axair Fans UK Limited (2020): Wind Turbine Ventilation & Cooling. Online verfügbar unter <https://www.axair-fans.co.uk/wp-content/uploads/2022/10/Renewable-Energy-Wind.pdf>.
- Bauernhansl, T.; Krüger, J.; Reinhart, G.; Schuh, Günther (2016): WGP-Standpunkt Industrie 4.0: Wissenschaftliche Gesellschaft für Produktionstechnik WGP e.V.

- Beck, Kent (2010): Extreme programming explained. Embrace change. 2. ed., 8. print. Boston: Addison-Wesley (The XP series).
- Becker, Fred G.; Gieselmann, Jana M. (2018): Anreizsysteme. In: Hans Corsten, Ralf Gössinger und Thomas S. Spengler (Hg.): Handbuch Produktions- und Logistikmanagement in Wertschöpfungsnetzwerken: De Gruyter, S. 387–412.
- Boehm, B.; Turner, R. (2003): Using risk to balance agile and plan-driven methods. In: *Computer* 36 (6), S. 57–66. DOI: 10.1109/MC.2003.1204376.
- Boehm, Barry W.; Turner, Richard (2004): Balancing agility and discipline. A guide for the perplexed. 6. printing. Boston, Mass.: Addison-Wesley.
- Böhm, Janko (2019): Erfolgsfaktor Agilität. Wiesbaden: Springer Fachmedien Wiesbaden.
- Bosch-Presse (21.07.2020): Bosch kompakt. Wussten Sie schon, dass...? Fakten rund um Software und Elektronik im Fahrzeug. Annett Fischer. Online verfügbar unter https://www.bosch-presse.de/pressportal/de/media/dam_images/pi11181/pi11181_de.pdf.
- Brehm, Lars; Feldmüller, Dorothee; Rieke, Tobias (2017): Konfiguration des hybriden Projektmanagements für die Entwicklung technischer, physischer Produkte.
- Bursać, Nikola (2016): Model Based Systems Engineering zur Unterstützung der Baukastenentwicklung im Kontext der Frühen Phase der Produktgenerationsentwicklung. Dissertation. Karlsruher Institut für Technologie (KIT). Institut für Produktentwicklung, zuletzt geprüft am 22.03.2021.
- Casse, Olivier (2017): SysML in action with Cameo systems modeler. London: ISTE Press Ltd (Implementation of model based system engineering set).
- Cha, Jianzhong (2014): Moving integrated product development to service clouds in the global economy. Proceedings of the 21st ISPE Inc. International Conference on Concurrent Engineering, September 8-11, 2014 volume 1. Online verfügbar unter <https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1640390>.

- Chow, Tsun; Cao, Dac-Buu (2008): A survey study of critical success factors in agile software projects. In: *Journal of Systems and Software* 81 (6), S. 961–971. DOI: 10.1016/j.jss.2007.08.020.
- Cockburn, Alistair (2003): Agile software development. 4. print. Boston, Mass., Munich: Addison-Wesley (The Agile software development series).
- Cooper, Robert G. (1990): Stage-gate systems: A new tool for managing new products (33).
- Cooper, Robert G.; Sommer, Anita F. (2016): The Agile-Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity. In: *J Prod Innov Manag* 33 (5), S. 513–526. DOI: 10.1111/jpim.12314.
- Delligatti, L. (2014): A Brief Guide to the Systems Modeling Language: China Machine Press, Beijing.
- Die Bundesregierung (2021): Koalitionsvertrag. Online verfügbar unter <https://www.bundesregierung.de/breg-de/service/gesetzesvorhaben/koalitionsvertrag-2021-1990800>.
- Diels, Frederic (2017): Indikatoren für die Ermittlung agil zu entwickelnder Produktumfänge. Dissertation. 1. Auflage (Produktionssystematik, 2018, Band 3).
- Dingsoeyr, Torgeir; Falessi, Davide; Power, Ken (2019): Agile Development at Scale: The Next Frontier. In: *IEEE Softw.* 36 (2), S. 30–38. DOI: 10.1109/MS.2018.2884884.
- Dumitrescu, R.; Albers, A.; Riedel, O.; Stark, R.; Gausemeier, J. (2021): Engineering in Deutschland—status quo in Wirtschaft und Wissenschaft, Ein Beitrag zum advanced systems engineering. In: *Fraunhofer IEM, Paderborn*, S. 88–92.
- Eigner, Martin; Dickopf, T.; Schulte, T.; Schneider, M. (2016): mecPro² - Entwurf einer Beschreibungssystematik zur Entwicklung cybertronischer Systeme mit SysML. In: Sven-Olaf Schulze: Tag des Systems Engineering. Ulm, 11. –13. November 2015. Hg. v. Sven-Olaf Schulze und Christian Muggeo. München: Hanser (Hanser eLibrary).
- Eigner, Martin; Koch, Walter; Muggeo, Christian (2017): Modellbasierter Entwicklungsprozess cybertronischer Systeme. Berlin, Heidelberg: Springer Berlin Heidelberg.

- Eigner, Martin; Roubanov, Daniil; Zafirov, Radoslav (2014): Modellbasierte virtuelle Produktentwicklung. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Eigner, Martin; Stelzer, Ralph (2009): Produktdatenmanagement-Systeme. Ein Leitfaden für Product Development und Life Cycle Management. 2., Aufl. Berlin: Springer-Verlag GmbH & Co. KG.
- Eklund, Ulrik; Berger, Christian (2017): Scaling agile development in mechatronic organizations - a comparative case study, S. 173–182. DOI: 10.1109/ICSE-SEIP.2017.25.
- VDI 2221, 2019: Entwicklung technischer Produkte und Systeme - Modell der Produktentwicklung.
- Farnbach, J. S. (2011): Why companies aren't jumping on the agile bandwagon. In: *Product Development & Management Association* 35 (3), S. 37–39.
- Friedenthal, Sanford; Moore, Alan; Steiner, Rick (2015): A practical guide to SysML. The systems modeling language. 3. ed. Amsterdam, Waltham, Mass.: Elsevier; Morgan Kaufmann.
- Gadatsch, Andreas (2020): Grundkurs Geschäftsprozess-Management. Wiesbaden: Springer Fachmedien Wiesbaden.
- Gausemeier, Ing Jürgen; Christian, Dipl-Wirt-Ing M. Eng; Gaukster, Dipl-Wirt-Ing Tobias; Dumitrescu, Ing Roman; others (2012): Integrative Konzipierung von Produkt und Produktionssystem als Basis für eine erfolgreiche Produktentstehung.
- Gerasimov, Arkadii; Heuser, Patricia; Ketteniß, Holger; Letmathe, Peter; Michael, Judith; Netz, Lukas Stephan Michael et al. (2020a): Generated Enterprise Information Systems: MDSE for Maintainable Co-Development of Frontend and Backend. Hg. v. Modellierung 2020 Conference. Vienna, Austria. Online verfügbar unter <https://publications.rwth-aachen.de/record/784771>, zuletzt geprüft am 23.01.2023.
- Gerasimov, Arkadii; Michael, Judith; Netz, Lukas; Rumpe, Bernhard (2021): Agile Generator-Based GUI Modeling for Information Systems. In: Ajantha Dahanayake, Oscar Pastor und Bernhard Thalheim (Hg.): *Modelling to Program*, Bd. 1401. Cham: Springer International Publishing (Communications in Computer and Information Science), S. 113–126.

- Gerasimov, Arkadii; Michael, Judith; Netz, Lukas; Rumpe, Bernhard; and Varga, Simon (2020b): Continuous Transition from Model-Driven Prototype to Full-Size Real-World Enterprise Information Systems. Hg. v. AMCIS 2020. Online verfügbar unter https://aisel.aisnet.org/amcis2020/systems_analysis_design/systems_analysis_design/2, zuletzt geprüft am 23.01.2023.
- GitHub (Hg.) (2022): MontiCore/cd4analysis. Online verfügbar unter <https://github.com/MontiCore/cd4analysis>, zuletzt geprüft am 23.01.2023.
- Graessler, Iris; Hentze, Julian (2020): The new V-Model of VDI 2206 and its validation. In: *at - Automatisierungstechnik* 68 (5), S. 312–324. DOI: 10.1515/auto-2020-0015.
- Gutsch, Stefan; Beisl, Maximilian; Konopik, Jens; Souskengphet, Anna (2018): Agile Champions. Wie agil müssen Familienunternehmen morgen sein? Weismann-Studie 2018. Hg. v. Weissman & Cie. GmbH & Co. KG. Weissman & Cie. GmbH & Co. KG. Nürnberg. Online verfügbar unter https://www.weissman.de/media/Ergebnisse_Weissman-Studie_Agile-Champions.pdf, zuletzt geprüft am 23.01.2023.
- Herold, Susann (2011): Ausbildung von „Universalgenies“? Zum Kompetenzbegriff und translatorischen Kompetenzmodellen. In: *Lebende Sprachen* 55 (2). DOI: 10.1515/les.2010.017.
- Hilmer, Christian (2016): Prozessmanagement in indirekten Bereichen. Empirische Untersuchung und Handlungsempfehlungen. Wiesbaden: Springer Gabler (SpringerLink Bücher).
- Hofert, Svenja (2018): Das agile Mindset. Wiesbaden: Springer Fachmedien Wiesbaden.
- Höpfner, Gregor; Jacobs, Georg; Zerwas, Thilo; Drave, Imke; Berroth, Joerg; Guist, Christian et al. (2021): Model-Based Design Workflows for Cyber-Physical Systems Applied to an Electric-Mechanical Coolant Pump. In: *IOP Conf. Ser.: Mater. Sci. Eng.* 1097 (1), S. 12004. DOI: 10.1088/1757-899X/1097/1/012004.
- International Council on Systems Engineering (2007): SYSTEMS ENGINEERING VISION 2020. Hg. v. Technical Operations INCOSE.
- Jacobs, Georg; Konrad, Christian; Berroth, Joerg; Zerwas, Thilo; Höpfner, Gregor; Spütz, Kathrin (2022): Function-Oriented Model-Based Product Development. In: Dieter Krause und Emil Heyden

- (Hg.): Design Methodology for Future Products. Cham: Springer International Publishing, S. 243–263.
- JSON: Introducing JSON. Online verfügbar unter <https://www.json.org/json-en.html>, zuletzt geprüft am 17.08.2022.
- Kaufmann, Uwe; Pfenning, Michael (2014): Was die Produkt-von der Softwareentwicklung lernen kann & durchgängige Integration disziplinspezifischer Modelle durch den Einsatz von Modellierungssprachen. In: *Tag des Systems Engineering*, S. 329.
- Klein, Thorsten (2016): Agiles Engineering im Maschinen- und Anlagenbau. Dissertation. München (Forschungsberichte IWB, Band 323).
- Komus, Ayelt; Kuberg, Moritz; Schmidt, Sonja; Rost, Lisa; Koch, Claus-Peter (2020): Ergebnisbericht: Status Quo (Scaled) Agile 2019/20. Hochschule Koblenz. Online verfügbar unter <https://www.hs-koblenz.de/bpm-labor/status-quo-scaled-agile-2020>.
- Konrad, Christian; Jacobs, Georg; Rasor, Rik; Riedel, Ricarda; Katzwinkel, Tim; Siebrecht, Justus (2019): Enabling complexity management through merging business process modeling with MBSE. In: *Procedia CIRP* 84, S. 451–456. DOI: 10.1016/j.procir.2019.04.267.
- Konsortialbenchmarking Studie Model Based Systems Engineering. Online verfügbar unter <https://www.wzl.rwth-aachen.de/cms/wzl/wirtschaft/praxisprojekte/konsortial-benchmarks/~dcstc/model-based-systems-engineering/>, zuletzt geprüft am 21.11.23.
- Kusay-Merkle, Ursula (2021): Agiles Projektmanagement im Berufsalltag. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Ladas, Corey (2008): Scrumban. And other essays on Kanban System for Lean Software development. Saetle, WA: Modus Cooperandi Press (Modus Cooperandi Series).
- Lamm, Jesko G.; Weilkiens, Tim (2014): Method for Deriving Functional Architectures from Use Cases. In: *Syst. Engin.* 17 (2), S. 225–236. DOI: 10.1002/sys.21265.
- Larman, Craig; Vodde, Bas (2009): Scaling lean & agile development. Thinking and organizational tools for large-scale Scrum. Upper Saddle River, NJ: Addison-Wesley.

- Leicht-Deobald, Ulrich; Schafheitle, Simon Daniel; Weibel, Antoinette; Ebert, Isabel Laura; Kasper, Gabriel; Schank, Christoph (2020): No stone left unturned? Towards a framework for the impact of datafication technologies on organizational control. In: *AMD*. DOI: 10.5465/amd.2019.0002.
- Liker, Jeffrey K. (2004): *Toyota way. 14 management principles from the world's greatest manufacturer*. First edition. New York, N.Y.: McGraw-Hill Education; McGraw Hill (McGraw-Hill's AccessEngineering). Online verfügbar unter <https://www.accessengineering-library.com/content/book/9780071392310>.
- Mattmüller, Roland; Tunder, Ralph (2012): Primär- und Sekundärorganisation im Handel. In: Joachim Zentes, Bernhard Swoboda, Dirk Morschett und Hanna Schramm-Klein (Hg.): *Handbuch Handel*. Wiesbaden: Springer Fachmedien Wiesbaden, S. 833–846.
- Maurya, Ash (2022): *Running lean. Iterate from plan A to a plan that works*. Third edition. Beijing, Boston, Farnham, Sebastopol, Tokyo: O'Reilly (The Lean series). Online verfügbar unter <https://e-bookcentral.proquest.com/lib/kxp/detail.action?docID=6894541>.
- VDI 2221, 1993: *Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*.
- Meyer, Helga; Reher, Heinz-Josef (Hg.) (2016): *Projektmanagement*. Wiesbaden: Springer Fachmedien Wiesbaden.
- Munker, Florian (2016): Ein Ansatz zur anwenderorientierten Systemmodellierung für die interdisziplinäre Produktentwicklung = A User-Oriented Concept of Systems Modeling for Interdisciplinary Product Engineering. In: *Forschungsberichte. IPEK*. DOI: 10.5445/IR/1000063012.
- Object Management Group (2019): *OMG Systems Modeling Language (OMG SysML)*. Online verfügbar unter <https://www.omg.org/spec/SysML/1.6/>, zuletzt geprüft am 16.02.2021.
- oose Innovative Informatik eG. (Hg.): *SysML-Werkzeuge*. Online verfügbar unter <https://www.oose.de/nuetzliches/sysml-werkzeuge/>.
- Pichler, Roman (2011): *Agile product management with Scrum. Creating products that customers love*. 4th printing. Upper Saddle River, NJ: Addison-Wesley (The Addison-Wesley signature series).

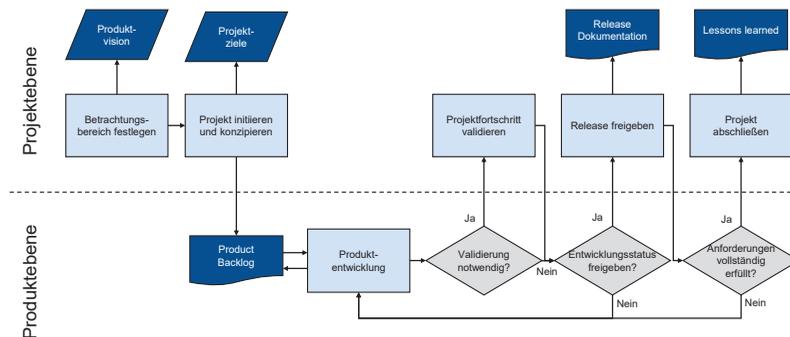
- Pohl, Klaus; Broy, Manfred; Daembkes, Heinrich; Hönninger, Harald (2016): Advanced Model-Based Engineering of Embedded Systems. In: Klaus Pohl, Manfred Broy, Heinrich Daembkes und Harald Hönninger (Hg.): Advanced Model-Based Engineering of Embedded Systems. Cham: Springer International Publishing, S. 3–9.
- Quentin, Jürgen (2021): Bedeutung der Windenergie. Online verfügbar unter <https://www.windindustrie-in-deutschland.de/publikationen/aktuell/bedeutung-der-windenergie-fuer-deutschland>.
- Riesener, Michael; Dölle, Christian; Becker, Annika; Schuh, Günther (2020): Agile Transformation in produzierenden Unternehmen. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 115 (6), S. 367–370. DOI: 10.3139/104.112350.
- Riesener, Michael; Dölle, Christian; Lauf, Hendrik; Schuh, Günther (2021): Framework for an agile, databased development. In: *Procedia CIRP* 100, S. 343–348. DOI: 10.1016/j.procir.2021.05.084.
- Riesener, Michael; Kuhn, Maximilian; Lauf, Hendrik; Bönsch, Gereon C.; Schuh, Günther (2022): Systematic derivation of customized development sprints for an agile development of wind turbines. In: *Procedia CIRP* 109, S. 299–304. DOI: 10.1016/j.procir.2022.05.253.
- Royce, Winston W. (1970): Managing the Development of Large Software Systems (1970), S. 321–332. DOI: 10.7551/mitpress/12274.003.0035.
- RWTH Aachen University (2021): MontiCore Language Workbench and Library Handbook. Unter Mitarbeit von Bernhard Rumpe, Katrin Hölldobler und Oliver Kautz. Online verfügbar unter <https://monti-core.de/download/handbook.pdf/>, zuletzt geprüft am 23.01.2023.
- Schewe, Gehard (2018): Aufbauorganisation. Hg. v. Gabler Wirtschaftslexikon. Online verfügbar unter <https://wirtschaftslexikon.gabler.de/definition/aufbauorganisation-31264/version-254826>.
- Schlick, Christopher; Bruder, Ralph; Luczak, Holger (2018): Arbeitswissenschaft. 4. Aufl. 2018. Berlin, Heidelberg: Springer Berlin Heidelberg. Online verfügbar unter <http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1521821>.
- Schmelzer, Hermann J.; Sesselmann, Wolfgang (2004): Geschäftsprozessmanagement in der Praxis. Produktivität steigern, Wert erhöhen, Kunden zufrieden stellen. 4., erw. Aufl. München, Wien: Hanser.

- Schuh, Günther (2002): Serienanlauf in branchenübergreifenden Netzwerken.
- Schuh, Günther (2013): Lean Innovation. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Schuh, Günther (2015): Leitfaden zur Baukastengestaltung. Ergebnisse des Forschungsprojekts Gestaltung innovativer Baukasten- und Wertschöpfungsstrukturen (GiBWert) ; ... innerhalb des Rahmenkonzeptes "Foschung für die Produktion von morgen". Stuttgart: VDMA Verl.
- Schuh, Günther (Hg.) (2022): Leitfaden zur Baukastengestaltung in der Windenergiebranche. 1. Aufl. Aachen: Apprimus-Verl. (Forschungsbericht).
- Schuh, Günther; Dölle, Christian; Kreutzer, Ramon; Ays, Johanna; Ebi, Mael; Schwarberg, Thomas et al. (2021): Agile Innovation. Gestaltungsfelder der agilen Produktentwicklung. Aachen: Werkzeugmaschinenlabor WZL der RWTH Aachen.
- Schwaber, Ken (1997): SCRUM Development Process, S. 117–134. DOI: 10.1007/978-1-4471-0947-1_11.
- Schwarzer, Bettina; Krcmar, Helmut (Hg.) (1995): Grundlagen der Prozessorientierung. Wiesbaden: Deutscher Universitätsverlag.
- Schwerdtfeger, A. (1999): Methoden zur materialflußgerechten Zuordnung von Betriebsbereichen und -mitteln. Berlin: Beuth.
- Sendler, Ulrich (2013): Book cover Book cover Industrie 4.0 pp 1–19 Cite as Industrie 4.0– Beherrschung der industriellen Komplexität mit SysLM (Systems Lifecycle Management). In: Ulrich Sendler (Hg.): Industrie 4.0. Berlin, Heidelberg: Springer Berlin Heidelberg (Xpert.press), S. 1–19.
- Serrador, Pedro; Pinto, Jeffrey K. (2015): Does Agile work? — A quantitative analysis of agile project success. In: *International Journal of Project Management* 33 (5), S. 1040–1051. DOI: 10.1016/j.ijproman.2015.01.006.
- Siakas, Kerstin V.; Siakas, Errikos (2007): The agile professional culture: A source of agile quality. In: *Softw. Process: Improve. Pract.* 12 (6), S. 597–610. DOI: 10.1002/spip.344.

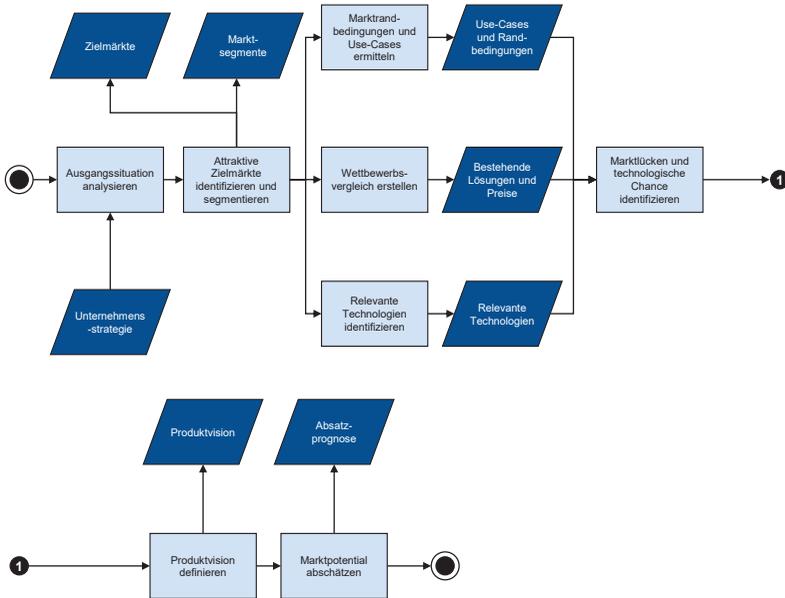
- Smith, Preston G. (2007): Flexible product development. Building agility for changing markets. San Francisco: John Wiley & Sons; Jossey-Bass.
- Stojanov, Igor; Turetken, Oktay; Trienekens, Jos J.M. (2015): A Maturity Model for Scaling Agile Development, S. 446–453. DOI: 10.1109/SEAA.2015.29.
- Suchanek, Andreas (2021): Gabler Wirtschaftslexikon: Verantwortung. Definition: Was ist "Verantwortung"? Hg. v. Gabler Wirtschaftslexikon. Gabler Wirtschaftslexikon. Online verfügbar unter <https://wirtschaftslexikon.gabler.de/definition/verantwortung-50418/version-384796>, zuletzt geprüft am 23.01.2023.
- United Nations (2015): Paris Agreement.
- Verein Deutscher Ingenieure e.V. (2020): Entwicklung cyber-physischer mechatronischer Systeme (CPMS).
- VersionOne (2017): 11th annual state of agile report. Online verfügbar unter <https://digital.ai/resource-center/analyst-reports/state-of-agile-report/>, zuletzt aktualisiert am 01.12.2022.
- vom Brocke, Jan (2003): Referenzmodellierung. Gestaltung und Verteilung von Konstruktionsprozessen. 2., unveränderte Auflage. Berlin: Logos Verlag (Advances in information systems and management science, 4).
- Weilkiens, Tim (2014): Systems Engineering mit SysML/UML. Anforderungen, Analyse, Architektur. 3., überarb. und aktualisierte Aufl. Heidelberg: dpunkt.verl.
- Werkzeugmaschinenlabor WZL der RWTH Aachen University (2018): Agile Innovation - Design Fields of Agile Product Development. Aachen.
- Wessels, Holger; Heimicke, Jonas; Rapp, Simon; Grauberger, Patric; Richter, Thilo; Matthiesen, Sven; Albers, Albert (2019): Sprint planning in mechatronic system development based on reference system elements.
- Zerwas, Thilo; Jacobs, Georg; Spütz, Kathrin; Höpfner, Gregor; Drave, Imke; Berroth, Joerg et al. (2021): Mechanical concept development using principle solution models. In: *IOP Conf. Ser.: Mater. Sci. Eng.* 1097 (1), S. 12001. DOI: 10.1088/1757-899X/1097/1/012001.

A Anhang

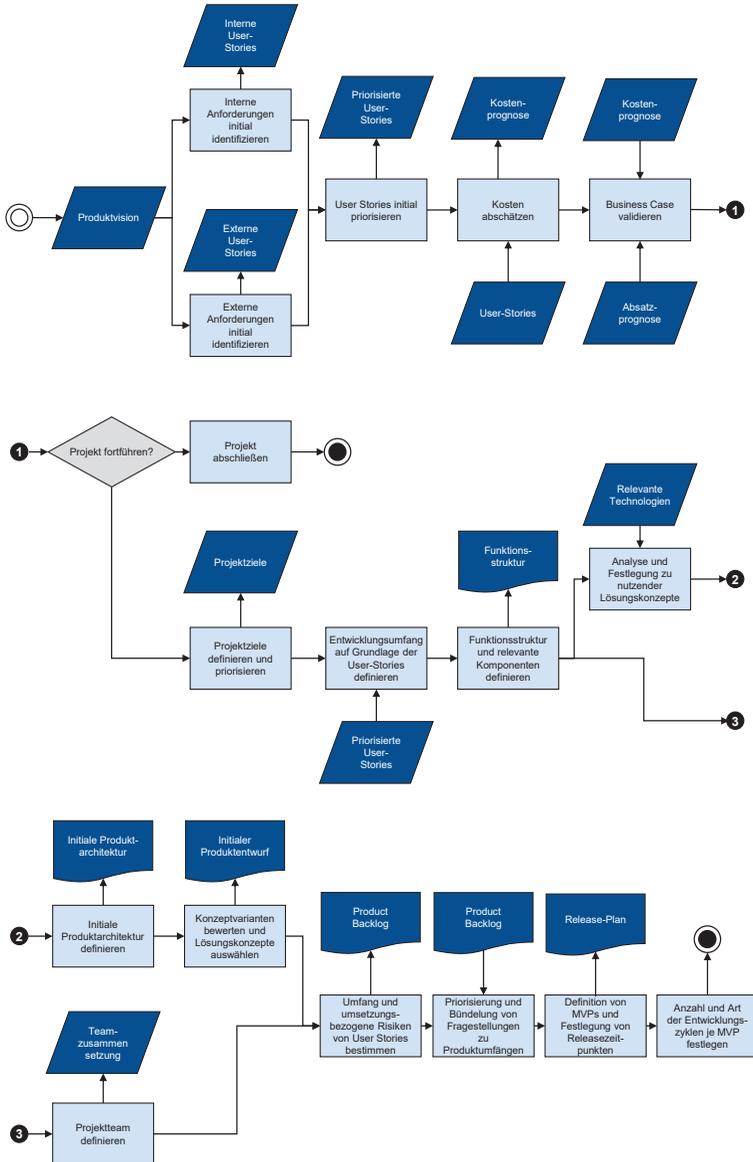
A Ablauforganisatorische Struktur



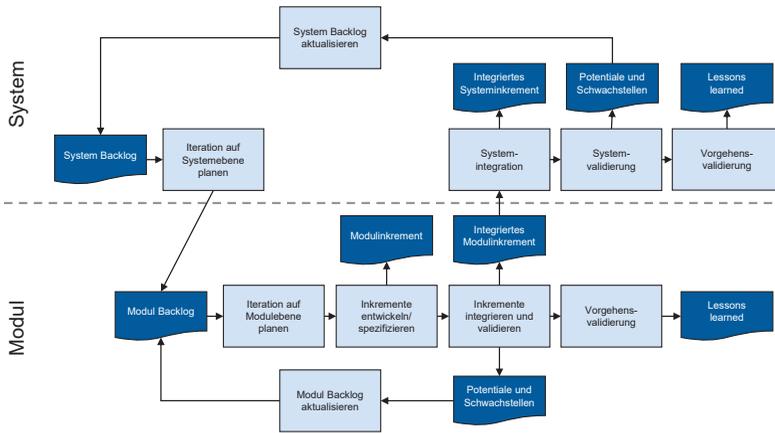
Anhang A1.1 Ablauforganisatorische Struktur des übergeordneten Produktentwicklungsprozesses



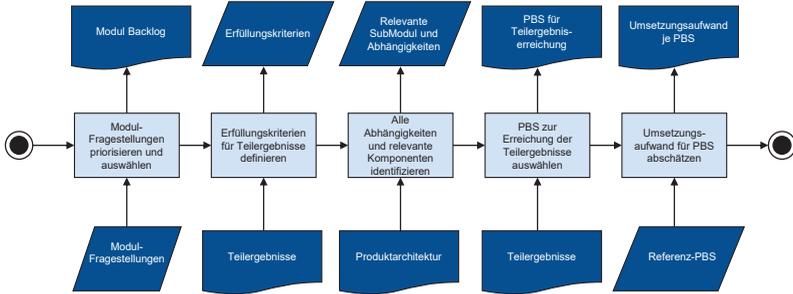
Anhang A1.2 Ablauforganisatorische Struktur für „Betrachtungsbereich festlegen“



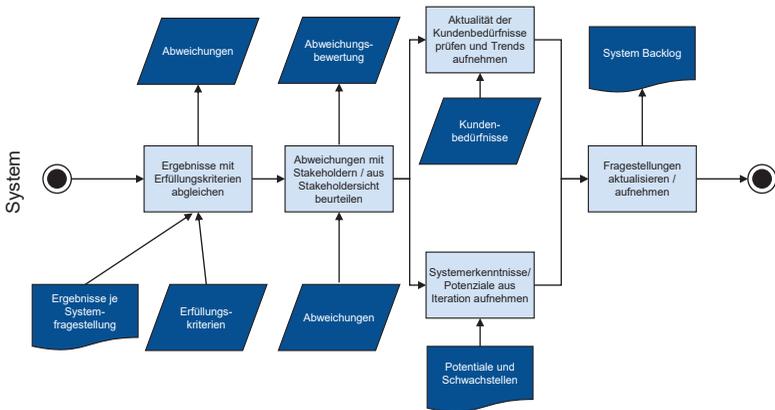
Anhang A1.3 Ablauforganisatorische Struktur für „Projekt initiieren und konzipieren“



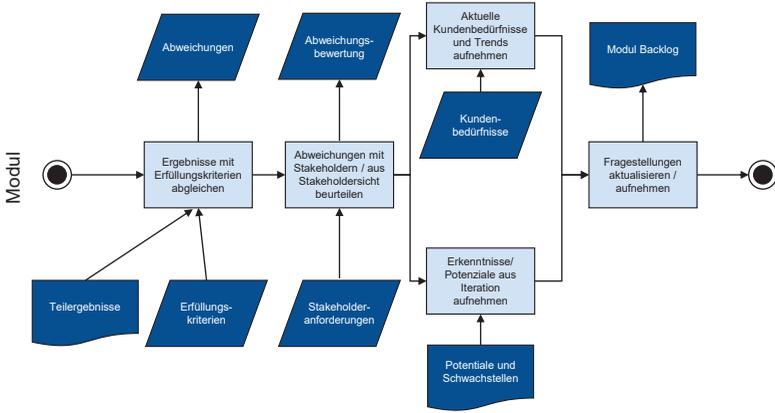
Anhang A1.4 Generische ablauforganisatorische Struktur der Produktentwicklungsebene



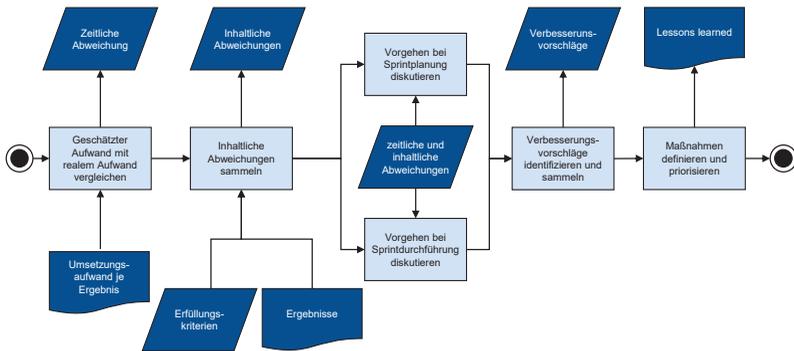
Anhang A1.6 Ablauforganisatorische Struktur für „Iteration auf Modulebene planen“



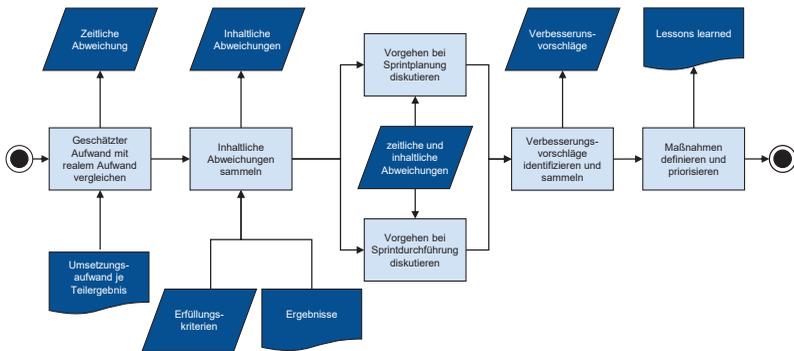
Anhang A1.7 System Backlog aktualisieren



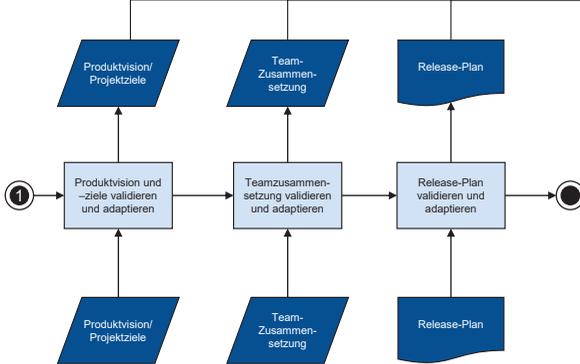
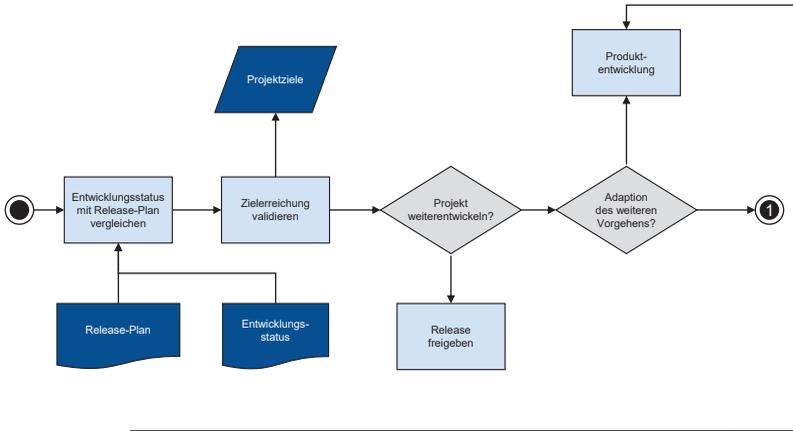
Anhang A1.8 Modul Backlog aktualisieren



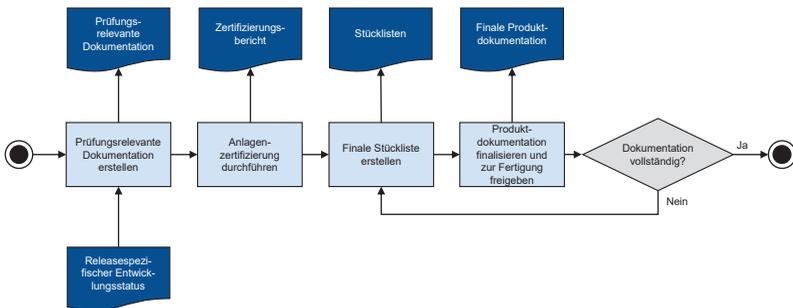
Anhang A1.9 Vorgehensvalidierung auf Systemebene



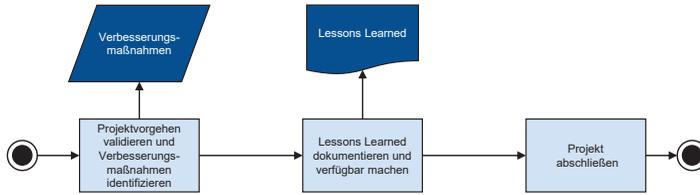
Anhang A1.10 Vorgehensvalidierung auf Modulebene



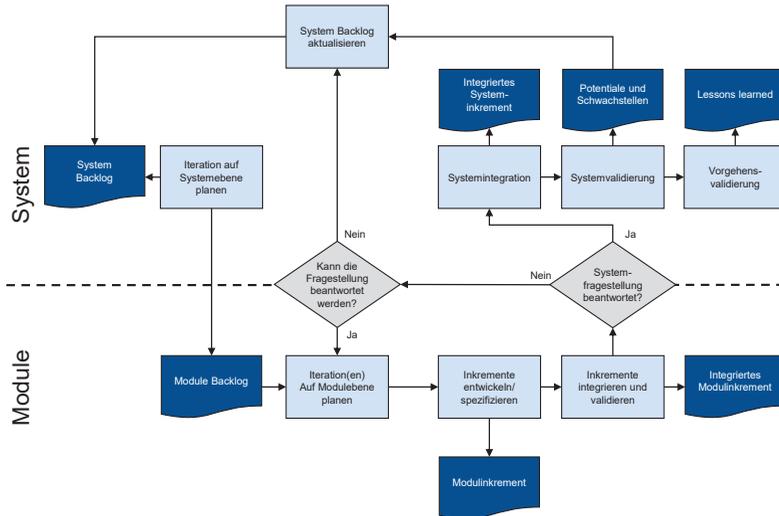
Anhang A1.11 Ablauforganisatorische Struktur für „Projektfortschritt validieren“



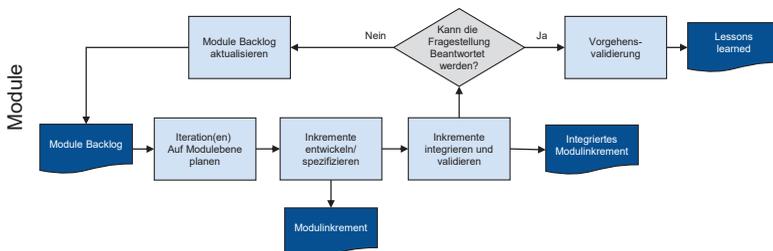
Anhang A1.12 Ablauforganisatorische Struktur für „Release freigeben“



Anhang A1.13 Ablauforganisatorische Struktur für „Projekt abschließen“



Anhang A1.14 Generische ablauforganisatorische Struktur der Produktentwicklung – Systemebene



Anhang A1.15 Generische ablauforganisatorische Struktur der Produktentwicklung – Modulebene

B Ausführliche Rollenbeschreibungen

Technical Operations Council (TOC)		
Aktivitäten	Kompetenzen	Verantwortlichkeit
<ul style="list-style-type: none"> ▪ Identifiziert attraktive Zielmärkte sowie Produktvarianten (Portfolio-Ebene) ▪ Überprüft Zielkonflikte in Projekten ▪ Informiert die System Owner (SO) der betroffenen Teams sowie den Key Account Manager (KAM) über die (Neu-)Priorisierung von Sprints/Entwicklungszyklen 	<ul style="list-style-type: none"> ▪ Hat die Kompetenz, bei Zielkonflikten zwischen verschiedenen Sprints/Entwicklungszyklen ein vorgegebenes Eskalationsformat in Form von Entscheidungsvorlagen (einschließlich Lösungsalternativen) zu erwarten ▪ Hat die Kompetenz, strategische Überlegungen bei der Prioritätensetzung zu berücksichtigen 	<ul style="list-style-type: none"> ▪ Ist dafür verantwortlich, die zugewiesene Ressourcenkapazität für einzelne Teams mindestens pro Sprint/Planungsperiode bereitzustellen ▪ Ist verantwortlich für die Lösung von Zielkonflikten zwischen verschiedenen Projekten ▪ Ist verantwortlich für die Schaffung notwendiger Rahmenbedingungen zur Erleichterung der Projektdurchführung (z.B. Personalausstattung, Ressourcenzuweisung)

Tabelle A2.1 Beschreibung der Rolle „Technical Operations Council“

Key Account Manager (KAM)		
Aktivitäten	Kompetenzen	Verantwortlichkeit
<ul style="list-style-type: none"> ▪ Stimmt Funktionsanforderungen mit dem Kunden ab ▪ Leitet die Schätzung von Funktionsanforderungen hinsichtlich Aufwand und Zeit ein ▪ Führt den gesamten Vorschlags-, Angebots- und Vertragsprozess durch ▪ Stimmt mit dem Kunden ein grobes Budget, den Zeitplan und den groben Umfang einer Funktionsanforderung ab ▪ Nimmt an der Demonstration und Überprüfung des Projekts Sprint teil ▪ Führt die interne und externe Projektberichterstattung durch und kommuniziert sie 	<ul style="list-style-type: none"> ▪ Hat die Befugnis, vom System Owner die erforderlichen Informationen für erste Schätzungen der Funktionsanforderungen anzufordern ▪ Hat die Befugnis, die Priorität eines Kunden vom System Owner ändern zu lassen ▪ Hat die Befugnis, Informationen anzufordern, um den Erfolg eines Projekts zu überwachen 	<ul style="list-style-type: none"> ▪ Ist verantwortlich für die Übergabe von Funktionsanforderungen und notwendigen Informationen an den System Owner zur Ausarbeitung der User Story ▪ Ist dafür verantwortlich, die Stimme des Kunden zu vertreten ▪ Ist dafür verantwortlich, die Projekte erfolgreich zu gestalten, indem er gültige Aufwands-, Kosten- und Wertabschätzungen vorlegt ▪ Ist dafür verantwortlich, sich an agilen Arbeitsweisen auszurichten ▪ Ist dafür verantwortlich, den Kunden bei der Aufteilung von Funktionsanforderungen in kleinere und priorisierte Teile zu beraten

Tabelle A2.2 Beschreibung der Rolle „Key Account Manager“

System Owner (SO)		
Aktivitäten	Kompetenzen	Verantwortlichkeit
<ul style="list-style-type: none"> ▪ Erstellt eine abgestimmte Vision der benötigten Funktionalität, indem Funktionsanforderungen mit dem Kunden und KAM abgeklärt werden ▪ Definiert affektive Funktionen und relevante Komponenten ▪ Definiert das Projektteam ▪ Definiert und aktualisiert ständig die Elemente im System-Backlog auf Grundlage der Funktionsanforderungen ▪ Koordiniert Abhängigkeiten mit anderen SO ▪ Eskaliert in Abstimmung mit KAM die Risiken des Entwicklungszyklus an den TOC 	<ul style="list-style-type: none"> ▪ Hat die Befugnis, System-Backlog-Items zu priorisieren ▪ Hat die Befugnis, einen System Sprint abzubrechen, wenn das System-Sprint-Ziel veraltet ist ▪ Hat die Macht, System Backlog Items jederzeit zu aktualisieren, falls erforderlich ▪ Hat die Macht, Behinderungen in Abstimmung mit dem KAM und/oder dem Management an den TOC zu eskalieren ▪ Hat die Befugnis, System-Backlog-Items, die nicht innerhalb des Entwicklungszyklus fertiggestellt werden können, an den TOC zu eskalieren 	<ul style="list-style-type: none"> ▪ Ist verantwortlich für die Priorisierung der Elemente des System Backlog, um gesteckte Ziele und Aufgaben bestmöglich zu erfüllen ▪ Ist verantwortlich für die Pflege und Kommunikation einer aktuellen Roadmap für den Entwicklungszyklus basierend auf priorisierten Backlog-Items für den laufenden Zyklus ▪ Ist dafür verantwortlich, einen kritischen Pfad eines Entwicklungszyklus unter Berücksichtigung von Abhängigkeiten sichtbar zu machen ▪ Ist dafür verantwortlich, den Arbeitsfortschritt für alle Beteiligten transparent zu machen

Tabelle A2.3 Beschreibung der Rolle „System Owner“

Module Owner (SO)		
Aktivitäten	Kompetenzen	Verantwortlichkeit
<ul style="list-style-type: none"> ▪ Erstellt eine abgestimmte Vision der benötigten Funktionalitäten, indem Funktionsanforderungen mit dem System Owner abgeklärt werden ▪ Bricht die initiale Produktarchitektur auf Modulebene herunter ▪ Bewertet Konzeptvarianten und wählt Lösungskonzepte aus ▪ Definiert und aktualisiert ständig die Elemente im Modul-Backlog auf Grundlage der Funktionsanforderungen ▪ Koordiniert Abhängigkeiten mit anderen Module Ownern ▪ Eskaliert Risiken des Entwicklungszyklus an den Systemeigentümer 	<ul style="list-style-type: none"> ▪ Hat die Befugnis, die Elemente des Modul-Backlog zu priorisieren ▪ Hat die Befugnis, einen Module Sprint abzubrechen, wenn das Modul Sprint-Ziel obsolet wird ▪ Hat die Möglichkeit, Modul-Backlog-Einträge bei Bedarf jederzeit zu aktualisieren ▪ Hat die Macht, Behinderungen an den System-eigentümer zu eskalieren ▪ Hat die Befugnis, Modul Backlog Items, die nicht innerhalb des Entwicklungszyklus fertiggestellt werden können, an den System Owner zu eskalieren 	<ul style="list-style-type: none"> ▪ Ist verantwortlich für die klare Darstellung der Modul Backlog Items und bereitet sie für die Planung des Entwicklungszyklus vor ▪ Ist verantwortlich für die Priorisierung der Modul Backlog Items, um gesteckte Ziele und Aufgaben bestmöglich zu erfüllen ▪ Ist verantwortlich für die Pflege und Kommunikation einer aktuellen Roadmap für den Entwicklungszyklus basierend auf priorisierten Backlog-Items für den laufenden Zyklus ▪ Ist dafür verantwortlich, den Arbeitsfortschritt für alle Beteiligten transparent zu machen ▪ Ist dafür verantwortlich, dem Systemeigentümer Aufwandsschätzungen für die Entwicklung von Funktionen zur Verfügung zu stellen

Tabelle A2.4 Beschreibung der Rolle „Module Owner“

System Responsible (SR)		
Aktivitäten	Kompetenzen	Verantwortlichkeit
<ul style="list-style-type: none"> ▪ Zerlegt System Backlog Items in einzelne Aktivitäten und schätzt sie ▪ Erstellt einen Arbeitsplan für die Abarbeitung des Arbeitspakets entsprechend den Zielsetzungen ▪ Führt Sprints aus ▪ Identifiziert Teilaufgaben, schätzt Teilaufgaben ab, meldet sich für Teilaufgaben an, führt Teilaufgaben aus und verfolgt den Status bis zum Abschluss ▪ Führt Teilaufgaben wie Analyse, Entwurf, Programmierung, Konfiguration, Test, Überprüfung aus ▪ Führt kundenspezifische Fehlerbehebung durch 	<ul style="list-style-type: none"> ▪ Hat die Kompetenz, selbständig Lösungen für gegebene Probleme zu finden ▪ Hat die Befugnis, den Aufwand von System Backlog Items unabhängig von den Stakeholdern zu schätzen ▪ Hat die Befugnis, die Anzahl der Elemente aus dem System Backlog für den System Sprint auf der Grundlage von Kapazität und Durchführbarkeit auszuwählen ▪ Hat die Befugnis, Arbeitsaufgaben abzulehnen, wenn die "Definition of Ready" für bereits begonnene Aufgaben nicht erfüllt ist ▪ Hat die Befugnis, den SO bei Unklarheiten um Klärung zu bitten 	<ul style="list-style-type: none"> ▪ Ist dafür verantwortlich, am Ende jedes System Sprints ein potenziell releasefähiges Inkrement des Endprodukts zu liefern ▪ Ist dafür verantwortlich, die Qualität des Codes, der Tests und der Dokumentation jedes potenziell releasefähigen Inkrements sicherzustellen ▪ Ist dafür verantwortlich, alle Änderungswünsche während des System Sprints von außerhalb des Teams an den Agile Master zu melden ▪ Ist verantwortlich, die eigene Arbeit gemäß TOC-Beschlüssen zu priorisieren

Tabelle A2.5 Beschreibung der Rolle „System Responsible“

Module Responsible (MR)		
Aktivitäten	Kompetenzen	Verantwortlichkeit
<ul style="list-style-type: none"> ▪ Zerlegt die Elemente des Modul-Backlog in einzelne Aktivitäten und schätzt sie ▪ Erstellt einen Arbeitsplan für die Abarbeitung des Arbeitspakets entsprechend den Zielsetzungen ▪ Führt Sprints aus ▪ Identifiziert Teilaufgaben, schätzt Teilaufgaben ein, meldet sich für Teilaufgaben an, führt Teilaufgaben aus und verfolgt den Status bis zum Abschluss ▪ Führt Teilaufgaben wie Analyse, Entwurf, Programmierung, Test, Überprüfung aus ▪ Leitet das Stand-up Meeting 	<ul style="list-style-type: none"> ▪ Hat die Kompetenz, selbständig Lösungen für gegebene Probleme zu finden ▪ Hat die Befugnis, den Aufwand von Modul Backlog Items unabhängig von den Stakeholdern zu schätzen ▪ Hat die Befugnis, die Anzahl der Elemente aus dem Modul Backlog für den Module Sprint auf der Grundlage von Kapazität und Durchführbarkeit auszuwählen ▪ Hat die Befugnis, Arbeitsaufgaben abzulehnen, wenn die "Definition of Ready" nicht erfüllt ist ▪ Hat die Befugnis, den MO bei Unklarheiten um Klärung zu bitten 	<ul style="list-style-type: none"> ▪ Ist dafür verantwortlich, am Ende jedes Module Sprints ein potenziell releasefähiges Inkrement des Endprodukts zu liefern ▪ Ist dafür verantwortlich, die Qualität des Codes, der Tests und der Dokumentation jedes potenziell releasefähigen Inkrements sicherzustellen ▪ Ist dafür verantwortlich, alle Änderungswünsche während des Module Sprints von außerhalb des Teams an den Agile Master zu melden ▪ Ist verantwortlich, die eigene Arbeit gemäß TOC-Beschlüssen zu priorisieren

Tabelle A2.6 Beschreibung der Rolle „Module Responsible“

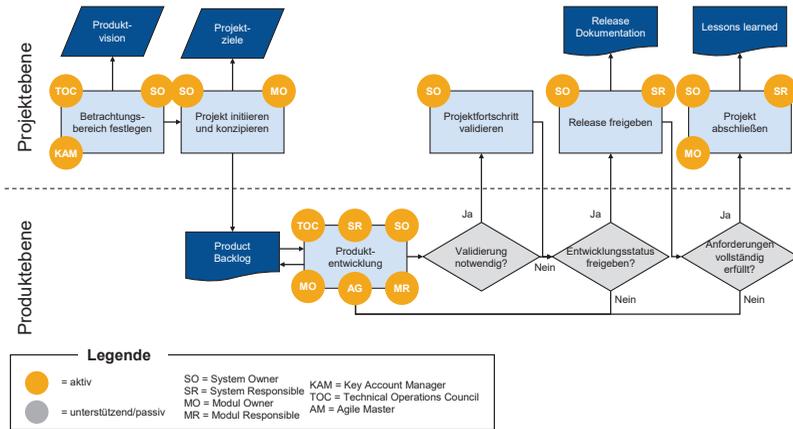
Agile Master (SM)		
Aktivitäten	Kompetenzen	Verantwortlichkeit
<ul style="list-style-type: none"> ▪ Erhält den agilen Prozess aufrecht, hilft, agile Prozesse zu verstehen ▪ Stellt sicher, dass agile Veranstaltungen stattfinden, und hilft auf Anfrage bei deren Organisation ▪ Unterstützt (SO/MO) bei der Verwaltung des System-/Modul-Backlog und hilft bei der Priorisierung ▪ Hilft SO/MO, den Teammitgliedern die Vision, die Ziele und die Elemente des System-/Modul-Backlog klar zu vermitteln ▪ Coacht das Team in Selbstorganisation und funktionsübergreifender Zusammenarbeit 	<ul style="list-style-type: none"> ▪ Hat die Macht, die Tagesordnung bei allen agilen Veranstaltungen zu moderieren und einzuhalten ▪ Hat die Macht, dafür zu sorgen, dass die Regeln für die Zusammenarbeit im Team eingehalten werden ▪ Hat die Befugnis, neue Anforderungen während des laufenden Sprints abzulehnen, wenn sie nicht mit dem SO/MO vereinbart sind ▪ Hat die Macht, Störungen bei allen agilen Veranstaltungen zu vermeiden ▪ Hat die Macht, Hindernisse für SO/MO und/oder das Management zu eskalieren 	<ul style="list-style-type: none"> ▪ Ist dafür verantwortlich, auftretende Hindernisse zu beseitigen, indem das Team dazu befähigt wird ▪ Ist verantwortlich für die Förderung der Agilität gemäß dem eingeführten agilen methodischen Rahmenwerk und den entsprechenden Strukturen, Werkzeugen, Regeln und Verhaltensaspekten ▪ Ist dafür verantwortlich, dass agile Ereignisse innerhalb der Zeitfenster bleiben ▪ Als Servant Leader des Teams ist er/sie dafür verantwortlich, das Team in die Lage zu versetzen, seinen/ihren Wert zu maximieren

Tabelle A2.7 Beschreibung der Rolle „Agile Master“

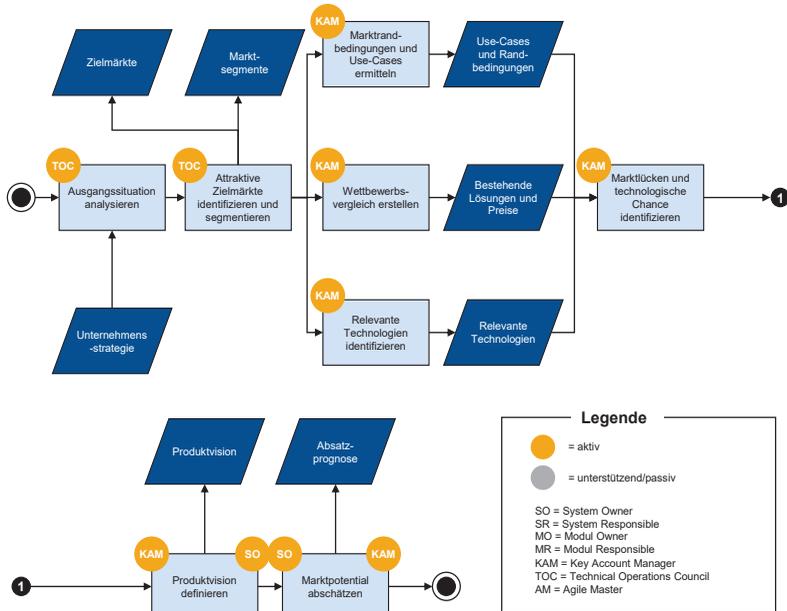
Agile Team (SO, MO, SR, MR)		
Aktivitäten	Kompetenzen	Verantwortlichkeit
<ul style="list-style-type: none"> ▪ Definiert die "Definition of Ready" auf der Grundlage gegebener Rahmennormen ▪ Definiert die "Definition of Done" auf der Grundlage gegebener Rahmennormen ▪ Führt die Sprint-Planung, Sprint-Review und Sprint-Retrospektive durch ▪ Führt Planungssitzung der individuellen Planungsperiode durch 	<ul style="list-style-type: none"> ▪ Hat die Befugnis zu entscheiden, wie und wann die Verfeinerung des System-/Modul-Backlog durchgeführt wird ▪ Hat die Entscheidungsgewalt über die agile Methodik auf Teamebene: (z.B. Scrum mit Sprint-Länge 2 Wochen) 	<ul style="list-style-type: none"> ▪ Ist verantwortlich für die Planung und Durchführung von Planung, Review und Retrospektive ▪ Ist dafür verantwortlich, sich selbst zu organisieren und die Arbeit zu erledigen ▪ Ist dafür verantwortlich, "Fertig"-Produkte schrittweise zu liefern, um sicherzustellen, dass nach jedem Sprint immer eine potenziell nützliche Version eines funktionierenden Produkts verfügbar ist. ▪ Ist dafür verantwortlich, die eigene Effektivität beständig zu steigern

Tabelle A2.8 Beschreibung der Rolle eines agilen Teams im Gesamtkontext

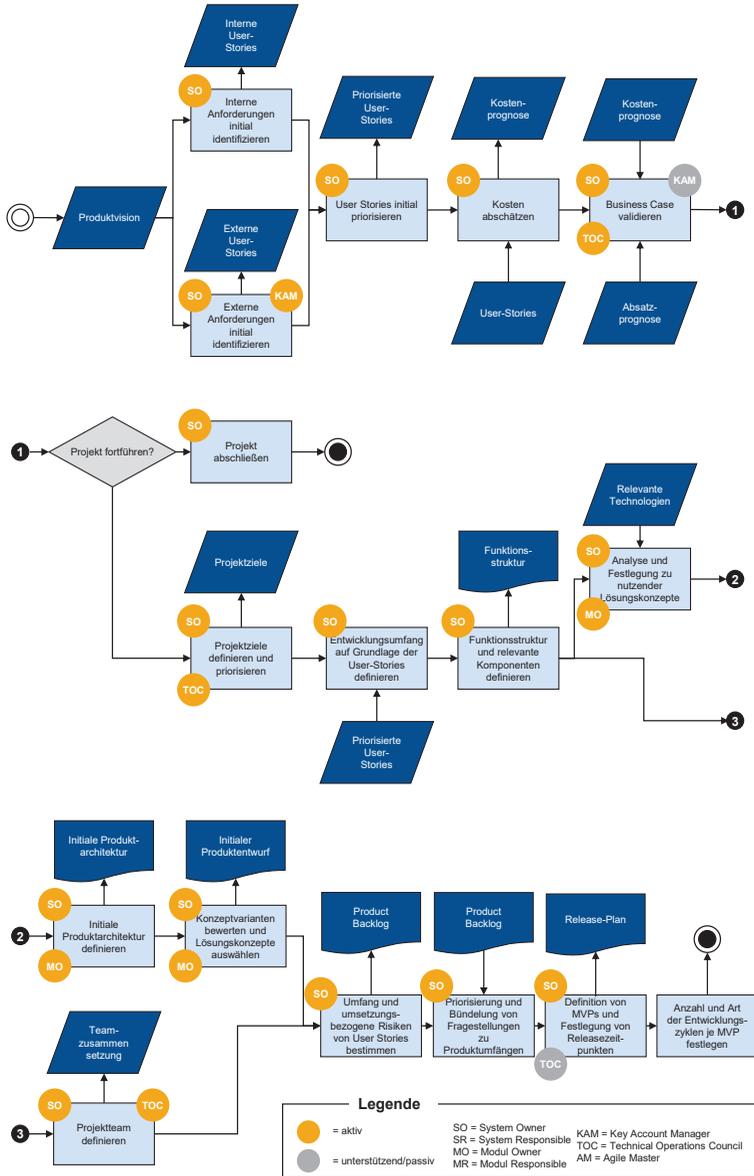
C Prozessbeschreibung mit Rollenzuordnung



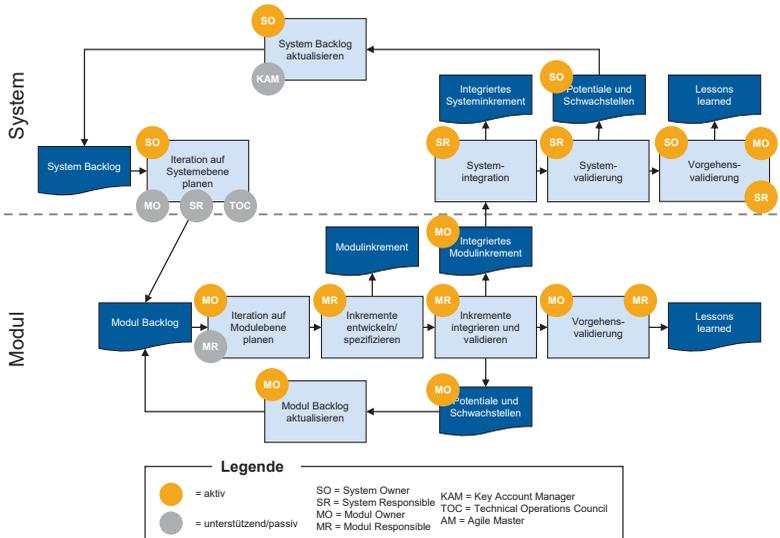
Anhang A3.1 Ablauforganisatorische Struktur des übergeordneten Produktentwicklungsprozesses (inkl. Rollenbeschreibung)



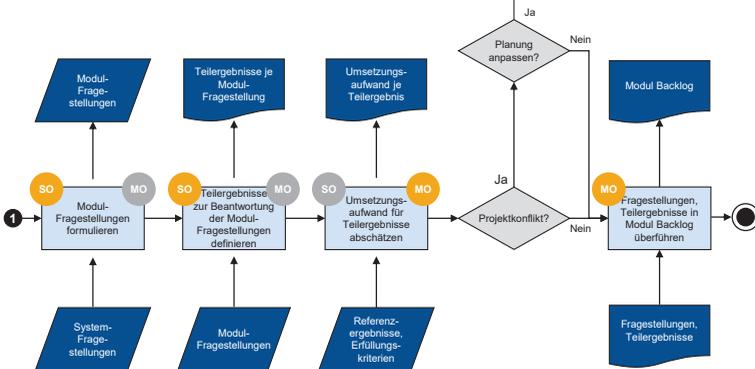
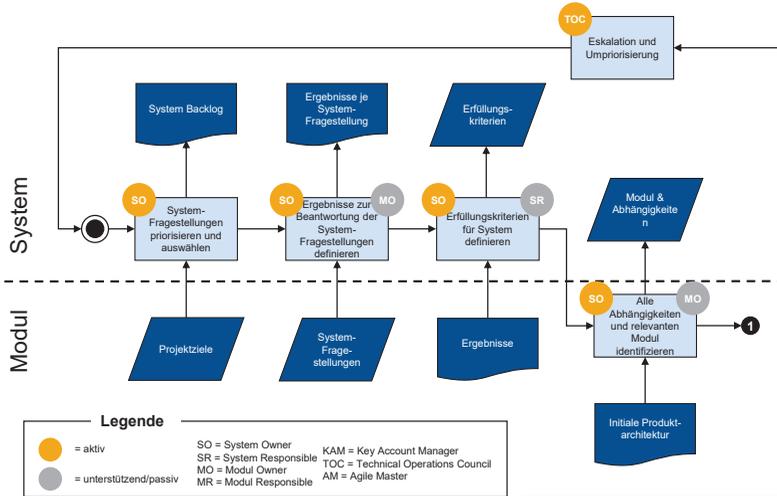
Anhang A3.2 Ablauforganisatorische Struktur für „Betrachtungsbereich festlegen“ (inkl. Rollenbeschreibung)



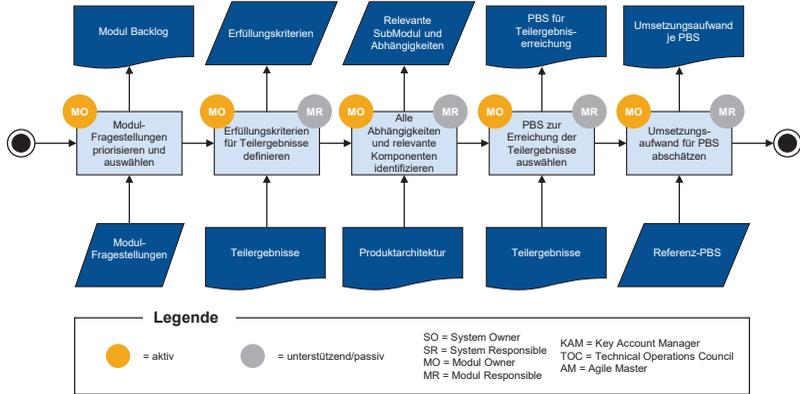
Anhang A3.3 Ablauforganisatorische Struktur für „Projekt initiieren und Projekt konzipieren“ (inkl. Rollenbeschreibung)



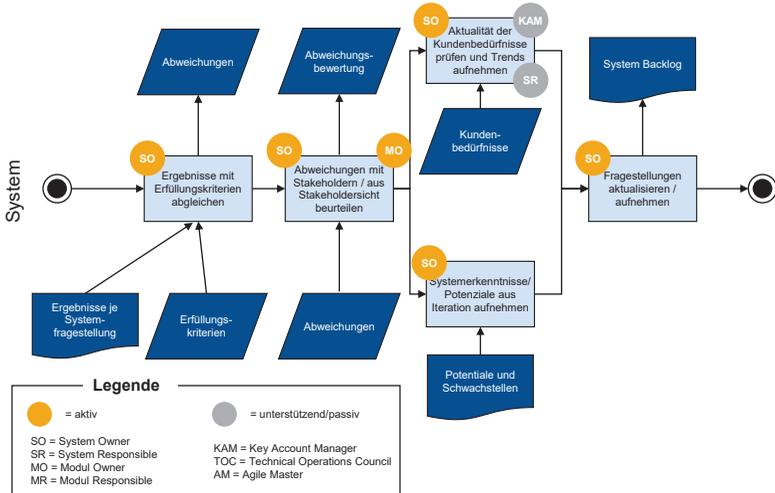
Anhang A3.4 Ablauforganisatorische Struktur der Produktentwicklungsebene (inkl. Rollenbeschreibung)



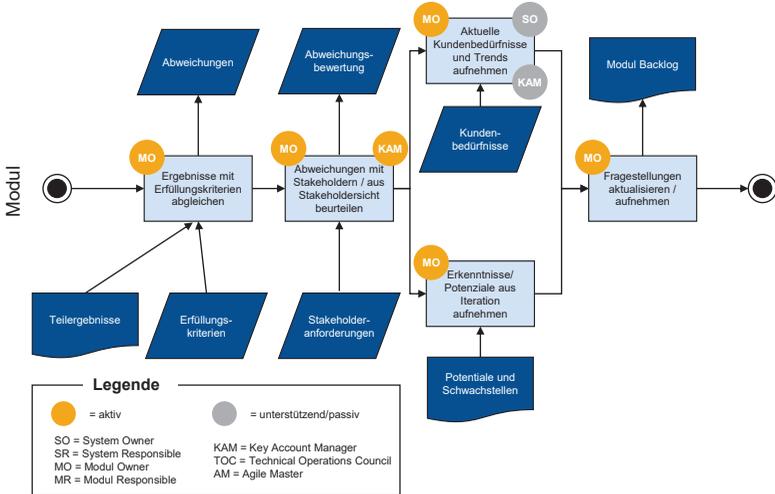
Anhang A3.5 Ablauforganisatorische Struktur für „Iteration auf Systemebene planen“ (inkl. Rollenbeschreibung)



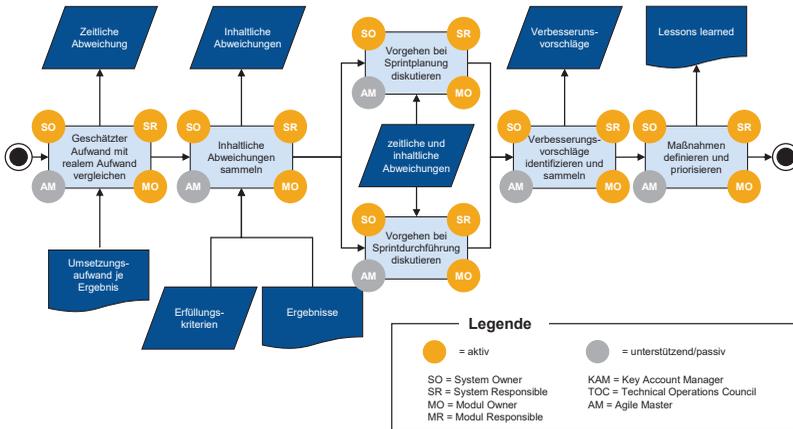
Anhang A3.6 Ablauforganisatorische Struktur für „Iteration auf Modulebene planen“ (inkl. Rollenbeschreibung)



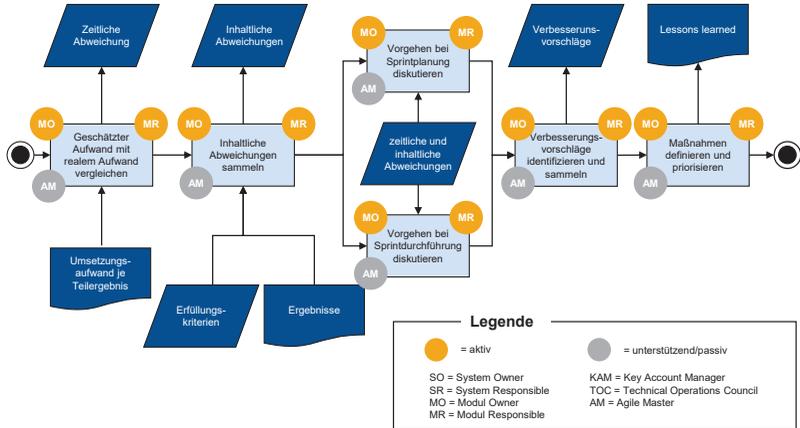
Anhang A3.7 System Backlog aktualisieren (inkl. Rollenbeschreibung)



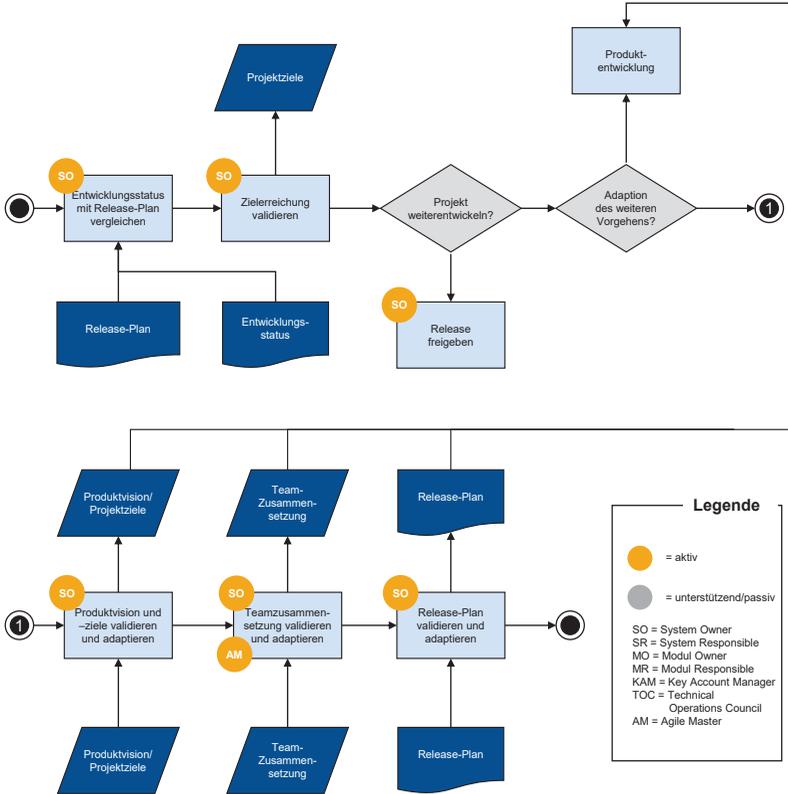
Anhang A3.8 Modul Backlog aktualisieren (inkl. Rollenbeschreibung)



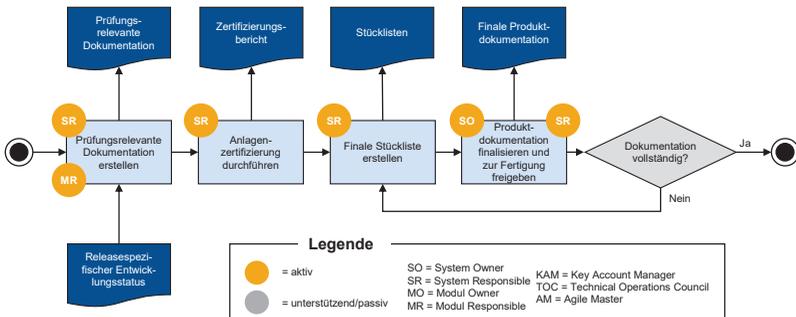
Anhang A3.9 Vorgehensvalidierung auf Systemebene (inkl. Rollenbeschreibung)



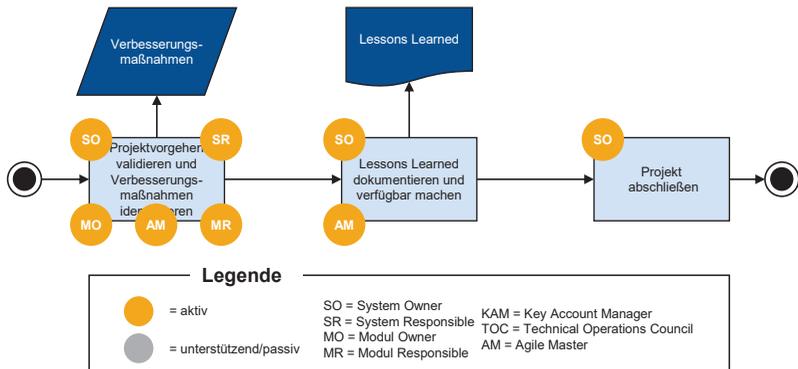
Anhang A3.10 Vorgehensvalidierung auf Modulebene (inkl. Rollenbeschreibung)



Anhang A3.11 Ablauforganisatorische Struktur für „Projektfortschritt validieren“ (inkl. Rollenbeschreibung)



Anhang A3.12 Ablauforganisatorische Struktur für „Release freigeben“ (inkl. Rollenbeschreibung)



Anhang A3.13 Ablauforganisatorische Struktur für „Projekt abschließen“ (inkl. Rollenbeschreibung)

