

Systematik zur Release-Planung intelligenter technischer Systeme

zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)
der Fakultät Maschinenbau
der Universität Paderborn

genehmigte
DISSERTATION

von
Dipl.-Wirt.-Ing. Arno Theodor Kühn
aus Steinfurt

Tag des Kolloquiums: 27. Oktober 2016
Referent: Prof. Dr.-Ing. Jürgen Gausemeier
Korreferent: Prof. Dr.-Ing. Frank-Lothar Krause

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter und Gruppenleiter am Fraunhofer-Institut für Entwurfstechnik Mechatronik IEM. Sie ist das Ergebnis meiner wissenschaftlichen Arbeit im Rahmen von Forschungs- und Industrieprojekten.

Mein Dank gilt in erster Linie Herrn Professor Dr.-Ing. Jürgen Gausemeier für das entgegengebrachte Vertrauen, die übertragene Verantwortung und die stets fordernde und fördernde Zusammenarbeit. Sie und das von Ihnen geschaffene Umfeld haben meine fachliche und persönliche Entwicklung in den letzten Jahren maßgeblich geprägt! Für die Übernahme des Korreferats danke ich Herrn Professor Dr.-Ing. Frank-Lothar Krause, dem ehemaligen Leiter des Fachgebiets Industrielle Informationstechnik am Institut für Werkzeugmaschinen und Fabrikbetrieb der TU Berlin und des Bereichs Virtuelle Produktentwicklung am Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik IPK.

Mein herzlicher Dank gilt zudem Herrn Professor Dr.-Ing. Roman Dumitrescu, den ich als Chef, Mitstreiter und Freund schätzen gelernt habe. Du hast mich unermüdlich zum Abschluss dieser Promotion motiviert! Danke für die fachliche und persönliche Förderung, das Vertrauen und die stets inspirierende und freundschaftliche Zusammenarbeit.

Allen heutigen und ehemaligen Kolleginnen und Kollegen danke ich für die tolle Zusammenarbeit und den außergewöhnlichen Teamgeist. Besonders hervorheben möchte ich an dieser Stelle Harald Anacker: Danke für die langjährige Zusammenarbeit, den Zusammenhalt und den gemeinsamen Spaß bei der Arbeit. Stellvertretend für alle Studierenden, die mich als studentische Hilfskräfte oder durch ihre studentischen Arbeiten unterstützt haben: Danke, Alina Linden, Laban Asmar, Dominik Bronsterning, Thomas Eckelt, Fabian Günther und Christian Tewes! Darüber hinaus danke ich allen Kolleginnen und Kollegen der it's OWL Clustermanagement GmbH – allen voran Günter Korder.

Meinen Freunden danke ich für das Verständnis und die vielen unvergesslichen Momente während der Promotionszeit. Letztlich ist es aber meine Familie, der das allergrößte Dankeschön gebührt! In erster Linie sind das meine Eltern Martina und Arno sowie mein Bruder Kristof: Ich bin euch unendlich dankbar für eure bedingungslose Geduld, Unterstützung und Liebe! Euer Rückhalt hat es erst möglich gemacht, diese Arbeit erfolgreich abzuschließen. Eine besondere Erwähnung verdient zudem meine Freundin Anell, die mir gerade auf den letzten Metern die Zweifel nahm und den Rücken stärkte: Du bereicherst mein Leben – heute und zukünftig!

Liste der veröffentlichten Teilergebnisse

- [DFK14] DUMITRESCU, R.; FECHTELPETER, C.; KÜHN, A.: Systematische Berücksichtigung von Fertigungsanforderungen im Model-Based Systems Engineering. In: MAURER, M.; SCHULZE, S.-O. (HRSG.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2014
- [GDJ+14] GAUSEMEIER, J.; DUMITRESCU, R.; JASPERNEITE, J.; KÜHN, A.; TRSEK, H.: Der Spitzencluster it's OWL auf dem Weg zu Industrie 4.0. Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZWF), Ausgabe 5, 2014, S. 336-346
- [KBD+14] KÜHN, A.; BREMER, C.; DUMITRESCU, R.; GAUSEMEIER, J.: Feature models supporting trade-off decisions in early mechatronic systems design. NordDesign, Finnland, 2014
- [DBK+15] DUMITRESCU, R.; BREMER, C.; KÜHN, A.; TRÄCHTLER, A.; FRIEBEN, T.: Model-based development of products, processes and production resources – A state oriented approach for an integrated system model of objects, processes and systems. In: at Automatisierungstechnik 2015, Nr. 63, DE GRUYTER Oldenbourg, 2015, S. 844-857
- [KDG15] KÜHN, A.; DUMITRESCU, R.; GAUSEMEIER, J.: Managing Evolution from Mechatronics to Intelligent Technical Systems. 4th International Conference on Advances in Mechanical Engineering, German-Malaysian Workshop Series 2015 “Systems Engineering for Advanced Mechatronics”, ICAME 2015, Bali, Indonesia, 2015

Zusammenfassung

Der Wandel von mechatronischen zu intelligenten technischen Systemen verspricht vielfältige Innovationspotentiale. Diese ermöglichen es Unternehmen, die Attraktivität der eigenen Produkte kontinuierlich durch die Einführung neuer Produkt-Features zu steigern. Dies drückt sich in mannigfachen Produktversionen und -generationen aus, deren Veröffentlichung durch abgestimmte Innovations- und Anpassungsschritte gesteuert werden muss. Hier zeichnet sich die Weiterentwicklung technischer Systeme in Form von Release-Projekten als geeigneter Lösungsansatz ab. Dies setzt jedoch einen systematischen Planungsprozess voraus, der marktstrategische und technische Aspekte gleichermaßen ins Kalkül zieht.

Vor diesem Hintergrund wird in der vorliegenden Arbeit eine *Systematik zur Release-Planung intelligenter technischer Systeme* entwickelt. Diese strukturiert den Planungsprozess in eine strategische, taktische und operative Release-Planung. Kernaufgabe der strategischen Release-Planung ist die Strukturierung des Release-Plans und die damit verbundene Festlegung von Release-Typen und -Zeitpunkten. Die Zuordnung von neuen Produkt-Features und Änderungen zu diesen Releases erfolgt in der taktischen Release-Planung. Im Rahmen der operativen Release-Planung wird schließlich unter Berücksichtigung der verfügbaren Ressourcen der tatsächliche Release-Inhalt festgelegt. Vorgehensmodelle und Hilfsmittel als Bestandteile der erarbeiteten Systematik unterstützen die Umsetzung dieses hierarchisierten Release-Planungsprozesses. Die Anwendung der Systematik wird abschließend anhand eines elektrischen Regelventils beschrieben.

Summary

Technological change from mechanics to intelligent technical systems leads to a fast evolution of products, which is characterized by manifold product versions and generations. To cope with this change, companies need to systematically manage product evolution in terms of concerted innovation and adaptation steps. In this context, developing a technical system in release projects seems to be a promising approach. However, this requires a systematic planning process equally taking into account market strategic and technical aspects.

Against this background a *systematic approach for release planning of intelligent technical systems* is developed in this thesis. It structures the planning process in strategic, tactical and operational release planning. Strategic release planning schedules the initial release plan and determines release types and dates. Tactical release planning assigns new product features and technical changes to these releases. Operational release planning finally determines the actual release content taking into account the available resources. Necessary process models and tools support the implementation of this hierarchical release planning process. The approach is finally applied to an electric control valve.

Inhaltsverzeichnis

Seite

1	Einleitung.....	1
1.1	Problematik.....	1
1.2	Zielsetzung.....	3
1.3	Vorgehensweise	3
2	Problemanalyse	5
2.1	Begriffsabgrenzung und -definition	5
2.1.1	Systematik	5
2.1.2	Der Systembegriff	6
2.1.3	Komplexität und Unsicherheit	7
2.1.4	Varianz technischer Systeme.....	9
2.1.5	Produkt- und Technologielebenszyklus.....	10
2.1.6	Technische Änderungen und Produkt-Features.....	12
2.1.7	Release, Release-Planung und Release-Management	13
2.1.8	Invention und Innovation	15
2.2	Von mechatronischen zu intelligenten technischen Systemen.....	15
2.2.1	Grundstruktur und Klassen mechatronischer Systeme	16
2.2.2	Intelligente technische Systeme.....	18
2.2.3	Evolution intelligenter technischer Systeme.....	21
2.2.4	Fazit und Einordnung der zu entwickelnden Systematik	23
2.3	Der Produktentstehungsprozess nach GAUSEMEIER.....	23
2.4	Release-Planung für technische Systeme	26
2.4.1	Der Release-Plan im Kontext der Produktplanung.....	27
2.4.2	Das Konzept der Release-Planung	29
2.4.3	Nutzen einer systematischen Release-Planung.....	32
2.4.4	Voraussetzungen für die Entwicklung in Releases	36
2.4.5	Fazit und Einordnung der zu entwickelnden Systematik	37
2.5	Interdisziplinäre Systementwicklung	38
2.5.1	Grundlagen des Systems Engineering.....	38
2.5.2	Vorgehensmodelle der Systementwicklung	42
2.5.3	Gestaltung änderungsfreundlicher Systemarchitekturen.....	44
2.5.4	Der Umgang mit Änderungen in der Systementwicklung	46
2.5.5	Fazit und Einordnung der zu entwickelnden Systematik	50
2.6	Problemabgrenzung.....	50
2.7	Anforderungen an die Systematik	52

3	Stand der Technik	55
3.1	Ansätze zur Spezifikation technischer Systeme.....	55
3.1.1	Spezifikationstechnik CONSENS	55
3.1.2	Modellierungstechniken basierend auf SysML.....	58
3.1.3	A3 Architecture Overview.....	59
3.1.4	Matrixbasierte Produktmodellierung.....	60
3.1.5	Feature-Modelle	62
3.2	Ansätze zur Analyse von technischen Änderungen	64
3.2.1	Change Prediction Method nach CLARKSON ET AL.	64
3.2.2	Technology Infusion nach SUH ET AL.	66
3.2.3	Modellbasierter Ansatz zur Analyse von Änderungsauswirkungen nach NONSIRI ET AL.	67
3.2.4	Analyse von Änderungen mit SysML4Mechatronics	68
3.3	Ansätze zur Klassifizierung und Priorisierung von Änderungen.....	70
3.3.1	Kano-Modell zur Klassifizierung von Änderungen.....	70
3.3.2	Klassifizierung von Änderungen nach ABMANN.....	71
3.3.3	Klassifizierung von Änderungen im Rahmen der Produktgenerationsentwicklung nach ALBERS ET AL.	73
3.3.4	Priorisierungstechniken nach BERANDER und ANDREWS	75
3.4	Ansätze zur strategischen Produktplanung.....	76
3.4.1	Time-to-Market Management nach NIPPA und LABRIOLA	76
3.4.2	Entwicklung von Produkt-Technologiestrategien nach BRINK..	78
3.4.3	Entwicklung produktlebenszyklusorientierter Geschäftsmodell-Roadmaps nach PEITZ.....	80
3.4.4	Planung lebenszyklusgerechter Leistungsbündel nach HEPERLE.....	82
3.5	Ansätze zur Produkt-Release-Planung	84
3.5.1	Release-Planungsprozess nach ZORN-PAULI ET AL.	85
3.5.2	Modulbasierte Release-Planung nach SCHUH ET AL.	86
3.5.3	Gestaltung und Pflege zyklengerechter Modul- und Plattformstrategien nach MAURER ET AL.....	89
3.5.4	Release-Management nach BELENER	91
3.6	Bewertung und Handlungsbedarf.....	94
4	Vorstellung der Systematik zur Release-Planung.....	97
4.1	Grundidee und Lösungsansatz	97
4.2	Überblick über die Systematik.....	100
4.3	Die Release-Strategie	102
4.3.1	Gestaltungsfaktoren des Release-Plans	102
4.3.2	Ordnungsschema alternativer Release-Strategien	107
4.3.3	Auswahl der geeigneten Release-Strategie	109

4.4	Vorgehensmodelle für die Release-Planung.....	111
4.4.1	Strategische Release-Planung.....	111
4.4.2	Taktische Release-Planung	112
4.4.3	Operative Release-Planung	115
4.5	Das Systemmodell als Kooperationskern in der Release-Planung ..	116
4.5.1	Erweiterte Systemmodellierung für die Release-Planung	117
4.5.2	Beschreibung der Produktevolution	120
4.5.3	Unterstützung der Änderungsanalyse.....	121
5	Anwendung und Bewertung der Systematik zur Release-Planung.....	125
5.1	Strategische Release-Planung.....	127
5.1.1	Phase 1: Auswahl der Release-Strategie.....	127
5.1.2	Phase 2: Definition der Release-Typen.....	130
5.1.3	Phase 3: Strukturierung des Release-Plans	132
5.1.3.1	Planung der Major-Releases.....	134
5.1.3.2	Planung der Minor-Releases.....	136
5.2	Taktische Release-Planung	136
5.2.1	Phase 1: Technische Änderungsanalyse	138
5.2.1.1	Analyse der initialen Änderungsauswirkung.....	139
5.2.1.2	Analyse der Änderungsfortpflanzung	142
5.2.1.3	Analyse der Änderungsabhängigkeiten	143
5.2.2	Phase 2: Änderungsbewertung	144
5.2.3	Phase 3: Klassifizierung und Zuordnung zu Releases	150
5.3	Operative Release-Planung	152
5.3.1	Phase 1: Erhebung der operativen Rahmenbedingungen	152
5.3.2	Phase 2: Auswahl eines Änderungsbündels	154
5.3.3	Phase 3: Überprüfung von Umsetzungsalternativen	157
5.3.4	Phase 4: Umsetzung des Änderungsbündels	158
5.4	Kritische Bewertung der Systematik	160
6	Zusammenfassung und Ausblick.....	163
7	Abkürzungsverzeichnis.....	167
8	Literaturverzeichnis	169

Anhang

A1	Ergänzungen zur Problemanalyse.....	A-1
A1.1	Risiken und Trade-Offs in der Produktentwicklung	A-1
A1.2	Gruppen und Häufigkeiten von Änderungsursachen	A-1
A1.3	Gestaltungsprinzipien für die Systemarchitektur	A-2
A1.4	Vor- und Nachteile modularer Systemarchitekturen.....	A-4
A2	Ergänzungen zum Stand der Technik	A-7
A2.1	Kurzbeschreibung verschiedener Priorisierungstechniken.....	A-7
A2.2	Release-Planungsprozess EVOLVE II nach RUHE.....	A-8
A3	Ergänzungen zur Systematik.....	A-10
A3.1	Kriterien zur Klassifizierung von technischen Änderungen	A-10
A3.2	Qualitätskriterien zur Bewertung von Produktplänen nach DESCHAMPS ET AL.	A-12
A3.3	Beispiel für die Struktur einer Wettbewerbs-Roadmap	A-13
A3.4	Alternative Release-Typen.....	A-14

1 Einleitung

Die vorliegende Arbeit resultiert aus Forschungs- und Industrieprojekten, die am FRAUNHOFER-INSTITUT FÜR ENTWURFSTECHNIK MECHATRONIK IEM durchgeführt worden sind. Stellvertretend sei an dieser Stelle das Verbundprojekt Systems Engineering genannt, das als Querschnittsprojekt Teil der Technologieplattform des *BMBF-Spitzenclusters Intelligente Technische Systeme OstWestfalenLippe (it's OWL)* ist [GDJ+14]. Ziel des Projekts ist ein Instrumentarium zur effektiven und effizienten Planung und Entwicklung intelligenter technischer Systeme im Spannungsdreieck von Qualität, Kosten und Zeit. Die Arbeit ist Bestandteil dieses Instrumentariums und beschreibt eine *Systematik zur Release-Planung intelligenter technischer Systeme*.

1.1 Problematik

Deutschland nimmt sowohl als Produktionsstandort als auch als Fabrikaurüster eine im internationalen Vergleich führende Position ein. Diese Position gilt es zu behaupten und auszubauen. Nach wie vor sind dafür erhebliche Innovationsanstrengungen erforderlich. Bedeutende Innovationstreiber sind Informations- und Kommunikationstechnologien in industriellen Anwendungsfeldern, wie dem Maschinen- und Anlagenbau, der Elektroindustrie und der Medizintechnik [aca11, S. 5], [FA13, S. 5], [aca14, S. 5]. Basierend auf diesen Technologien zeichnet sich ein Trend hin zu intelligenten technischen Systemen ab, der in Stichworten wie Selbstoptimierung [GRS14], Internet of Things and Services [aca14] und Cyber-Physical Systems [GB12] zum Ausdruck kommt. Aktuelle Studien belegen das erhebliche Innovations- und Marktpotential derartiger Systeme [aca11, S. 5], [BSM+14], [GSK+14].

Mit dem Wandel von mechatronischen zu intelligenten technischen Systemen eröffnen sich faszinierende Perspektiven: adaptive, robuste, vorausschauende und benutzungsfreundliche Systeme, die einen erheblich gesteigerten Funktionsumfang versprechen [Dum10], [GAC+13]. Unternehmen sehen sich so in der Lage, die Attraktivität ihrer Produkte kontinuierlich zu steigern. Gleichzeitig stellt der rasante technologische Fortschritt die Unternehmen vor Herausforderungen in der Planung und Entwicklung dieser Produkte [GDS+13], [AGN15]. Intelligente technische Systeme zeichnen sich durch einen zunehmenden Anteil von Software und Elektronik aus. Dies führt einerseits zu einer erhöhten Interdisziplinarität in der Entwicklung und gleichzeitig zu verkürzten Innovationszyklen. Unternehmen sind gezwungen, die eigenen Produkte weit häufiger zu aktualisieren als bisher üblich – bei gleichzeitig steigender Komplexität [aca14, S. 21f.], [Pei15, S. 2f.]. Dies resultiert aber nur in den seltensten Fällen in der vollständigen Neuentwicklung eines Produkts. Vielmehr führen stetige Produktverbesserungen zu technisch bedingten Änderungen und somit zu einer evolutionären Weiterentwicklung bestehender

Systeme.¹ Diese Produktevolution drückt sich in mannigfachen Produktversionen und neuen Produktgenerationen aus, deren Veröffentlichung durch abgestimmte Innovations- und Anpassungsschritte gesteuert werden muss [Bor10], [GB12, S. 100], [ABW15, S. 1ff.].

Ein dafür geeignetes Management-Instrument ist die Release-Planung, die die Produktevolution in Form dedizierter Releases plant. Ein Release bündelt eine Menge neuer Produkt-Features und technischer Änderungen, die gemeinsam entwickelt, getestet und freigegeben werden [Sch05, S. 113], [Ruh10, S. 3]. Ein Release repräsentiert somit den Freigabestand eines Produkts, der immer auch zu einer neuen Produktversion führt. Nutzenpotentiale dieser stufenweisen Umsetzung von Änderungen sind eine gezielte Produktwertsteigerung über den Produktlebenszyklus und die Balance der zur Umsetzung notwendigen Entwicklungsressourcen. Ziel ist es, den sich verändernden Markt- und Kundenbedürfnissen unter Berücksichtigung von Restriktionen in Bezug auf Markt, Organisation und Technik bestmöglich gerecht zu werden [Ruh10, S. XII]. Für einen durchgängigen Ansatz zur Release-Planung ergeben sich folgende Fragestellungen:

- Wann und wie häufig bietet sich die Veröffentlichung neuer Releases an? Welche Kriterien sind bei der Festlegung der Release-Zeitpunkte zu berücksichtigen (z.B. marktspezifische Termine wie Messen und Wettbewerbsaktivitäten)?
- Wann ist der richtige Release-Zeitpunkt für ein neues Produkt-Feature oder eine technische Änderung? Welche Kriterien sind bei der Zuordnung der Änderungen zu einem Release zu berücksichtigen (z.B. Komplexität, Priorität der Änderung)?
- Welche Änderungen können innerhalb des nächsten Releases umgesetzt werden? Welche Kriterien sind bei der Bündelung von Änderungen zu einem umsetzbaren Release zu berücksichtigen (z.B. verfügbare Entwicklungsressourcen)?

Insbesondere bei der Bündelung neuer Produkt-Features und Änderungen zu einem Release gilt es, neben marktstrategischen und organisatorischen Aspekten auch technische Abhängigkeiten und Restriktionen zu berücksichtigen. So setzen sich einige Änderungen technisch bedingt voraus und müssen gemeinsam innerhalb eines Release umgesetzt werden. Andere Änderungen können hingegen separat umgesetzt und somit auf ein späteres Release verschoben werden (z.B. um Zeit und Entwicklungskosten zu sparen) [SAA13]. Gerade aufgrund des interdisziplinären Charakters intelligenter technischer Systeme gilt es, technische Abhängigkeiten und Restriktionen aus Mechanik, Elektronik und Software gleichermaßen zu berücksichtigen. Dies setzt ein umfassendes und interdisziplinäres Systemverständnis sowie das Wissen über die technischen Auswirkungen einzelner Änderungen auf das bestehende System voraus.

¹ ALBERS ET AL. kommen in einer aktuellen empirischen Analyse zu dem Ergebnis, dass sich lediglich 7% aller Entwicklungsaktivitäten der befragten Entwicklungsingenieure (N = 159) einer vollständigen Neuentwicklung zuordnen lassen. Bei den übrigen Aktivitäten handelt es sich um Weiterentwicklungen ausgehend von einem Referenzprodukt [ABW15, S. 8f.].

Die geschilderte Problematik führt zur Notwendigkeit, die Release-Planung intelligenter technischer Systeme methodisch zu unterstützen. Es bedarf eines Ansatzes zur durchgängigen Release-Planung – von der Definition der Release-Zeitpunkte bis hin zur Zuordnung und Bündelung neuer Produkt-Features und Änderungen zu Releases. Ein besonderes Augenmerk muss hierbei auf der Berücksichtigung von Änderungsauswirkungen und -abhängigkeiten liegen. Dies erfordert neben der Strukturierung der Planungsaktivitäten insb. geeignete Hilfsmittel zur System- und Änderungsanalyse.

1.2 Zielsetzung

Ziel dieser Arbeit ist eine *Systematik zur Release-Planung intelligenter technischer Systeme*. **Zielgruppe** sind in erster Linie Fachleute, die mit der Weiterentwicklung technischer Systeme befasst sind (z.B. Produktmanager², Systems Engineer). Diese sollen bei der Umsetzungsplanung neuer Produkt-Features und technischer Änderungen in Form von Releases unterstützt werden. Dazu soll die Systematik drei **wesentliche Aufgabenbereiche** unterstützen: 1) Die initiale Strukturierung des Release-Plans durch Festlegung der Release-Zeitpunkte, 2) die Zuordnung technischer Änderungen zu diesen Release-Zeitpunkten und 3) die Bündelung der Änderungen zu einem umsetzbaren Release unter Berücksichtigung von Marktbedürfnissen, Änderungsauswirkungen und -abhängigkeiten sowie verfügbaren Ressourcen. Den **Kern der Systematik** sollen aufeinander abgestimmte Vorgehensmodelle bilden, die die Tätigkeiten innerhalb der drei Aufgabenbereiche strukturieren und den Einsatz ausgewählter Methoden, Werkzeuge und Beschreibungstechniken steuern. Als **Resultat** sind die Umsetzungstermine technischer Änderungen in Form von Releases definiert und in einem Release-Plan dokumentiert. Der Release-Plan dient als Ausgangspunkt für die Weiterentwicklung des technischen Systems in nachfolgenden Entwicklungsprojekten.

1.3 Vorgehensweise

In **Kapitel 2** wird die bereits skizzierte Problematik durch eine fundierte Problemanalyse konkretisiert. Dazu werden die für das Verständnis der Arbeit relevanten Begriffe eingeführt und das Forschungsfeld eingegrenzt. Ausgangspunkt ist die Beschreibung des Wandels von mechatronischen zu intelligenten technischen Systemen. Es werden die wesentlichen Charakteristika dieser Systeme sowie die mit dem Wandel verbundenen Herausforderungen herausgestellt. Es folgt eine Einordnung der zu entwickelnden Systematik in den Produktentstehungsprozess sowie eine detaillierte Analyse der zwei zentralen Handlungsfelder dieser Arbeit: der Release-Planung für technische Systeme sowie der interdisziplinären Systementwicklung. In einer abschließenden Problemabgrenzung werden

2 Die Inhalte der vorliegenden Arbeit beziehen sich in gleichem Maße sowohl auf Frauen als auf Männer. Aus Gründen der besseren Lesbarkeit wird jedoch die männliche Form (Ingenieur, Konstrukteur etc.) für alle Personenbezeichnungen gewählt. Die weibliche Form wird dabei stets mitgedacht. Eine Ausnahme bilden Inhalte, die ausdrücklich auf Frauen bezogen werden.

wesentliche Nutzenpotentiale und Herausforderungen zusammengefasst und in Anforderungen an die zu entwickelnde Systematik überführt.

Kapitel 3 gibt einen Überblick über den Stand der Technik. Die Auswahl der hier analysierten Ansätze orientiert sich an den in Kapitel 2 herausgearbeiteten Herausforderungen und Anforderungen. Zunächst werden dedizierte Hilfsmittel vorgestellt, die zur Unterstützung von Teilaufgaben in der Release-Planung relevant sind. Hierbei handelt es sich um Ansätze zur Spezifikation technischer Systeme sowie um Ansätze zur Analyse, Klassifizierung und Priorisierung von technischen Änderungen. Davon ausgehend werden Ansätze zur strategischen Produktplanung analysiert, die einerseits als angrenzende Systematiken zu verstehen sind, gleichzeitig aber auch relevante Lösungsansätze für die Release-Planung liefern. Die Analyse bestehender Ansätze zur Release-Planung zeigt schlussendlich den wesentlichen Handlungsbedarf auf. Das Kapitel schließt mit einer Bewertung aller Ansätze anhand der in Kapitel 2 ermittelten Anforderungen.

Kapitel 4 bildet den Kern der Arbeit und beschreibt die entwickelte Systematik. Ausgehend von der Grundidee des verfolgten Release-Planungs-Ansatzes wird ein Überblick über die Bestandteile der Systematik gegeben. Als ein wesentlicher Bestandteil werden alternative Release-Strategien eingeführt, die der Strukturierung eines initialen Release-Plans dienen. Dieser ist die Grundlage für den darauf aufbauenden durchgängigen Release-Planungsprozess, der durch drei Vorgehensmodelle für die strategische, taktische und operative Release-Planung strukturiert wird. Komplettiert wird die Systematik durch die Darstellung des Systemmodells als Kooperationskern in der Release-Planung.

In **Kapitel 5** wird die durchgängige Anwendung der Systematik an einem Praxisbeispiel gezeigt. Hierzu wird das Vorgehen in den drei Ebenen strategische, taktische und operative Release-Planung einschließlich der einzusetzenden Hilfsmittel beschrieben. Das Anwendungsbeispiel ist so gewählt, dass die Erfüllung der aufgestellten Anforderungen und der Nutzen der Systematik nachgewiesen werden können. Darüber hinaus unterstreicht das Beispiel die Industrierelevanz der in dieser Arbeit erzielten Ergebnisse.

Im abschließenden **Kapitel 6** werden die zentralen Ergebnisse dieser Arbeit zusammengefasst und ein Ausblick auf zukünftige Forschungsfragen und -aktivitäten gegeben. Der **Anhang** umfasst ergänzende Informationen zur entwickelten Systematik.

Das wissenschaftliche Vorgehen in dieser Arbeit orientiert sich an den Richtlinien der Arbeitsgruppe *Strategische Produktplanung und Systems Engineering* des HEINZ NIXDORF INSTITUTS und der Fakultät für Maschinenbau der UNIVERSITÄT PADERBORN. Entstanden ist die Arbeit während der Tätigkeit des Autors als Mitarbeiter des FRAUNHOFER-INSTITUTS FÜR ENTWURFSTECHNIK MECHATRONIK IEM. Die während dieser Zeit geleiteten und bearbeiteten Forschungs- und Industrieprojekte bilden die Basis für die erarbeitete Systematik.

2 Problemanalyse

Ziel der Problemanalyse sind Anforderungen an eine *Systematik zur Release-Planung intelligenter technischer Systeme*. Dazu werden in **Abschnitt 2.1** zunächst die für das Verständnis dieser Arbeit relevanten Begriffe eingeführt und abgegrenzt. Davon ausgehend beschreibt **Abschnitt 2.2** den Wandel von mechatronischen Systemen zu intelligenten technischen Systemen, die den Betrachtungsgegenstand dieser Arbeit bilden. **Abschnitt 2.3** ordnet die zu entwickelnde Systematik in den Produktentstehungsprozess ein, ehe **Abschnitt 2.4** das Konzept, den Nutzen sowie Voraussetzungen der Release-Planung für technische Systeme detailliert. Basis für eine erfolgreiche Release-Planung ist ein umfassendes und interdisziplinäres Systemverständnis. Auf relevante Grundlagen der interdisziplinären Systementwicklung wird daher in **Abschnitt 2.5** näher eingegangen. Dies umfasst u.a. eine Einführung in das Systems Engineering sowie die damit verbundene Teildisziplin des Model-Based Systems Engineerings (MBSE). Auch der Umgang mit Änderungen in der Systementwicklung wird beleuchtet. Abschließend erfolgt in **Abschnitt 2.6** eine Problemabgrenzung, aus der in **Abschnitt 2.7** die Anforderungen an die zu entwickelnde Systematik resultieren.

2.1 Begriffsabgrenzung und -definition

In der Arbeit werden Begriffe verwendet, die in der Literatur verschieden definiert und erläutert werden. Für ein einheitliches Verständnis im Rahmen der Arbeit werden diese Begriffe präzisiert und inhaltlich abgegrenzt. Der Begriff *Systematik* gibt Aufschluss über die Bestandteile der Ergebnisse dieser Arbeit. Deren Betrachtungsgegenstand sind Produkte, die als technische *Systeme* aufgefasst werden und sich durch hohe *Komplexität* und *Unsicherheit* auszeichnen. Einen nicht unwesentlichen Beitrag zu dieser Komplexität leistet die räumliche und zeitliche *Varianz*. Dabei erfordert insb. der Aspekt der zeitlichen Varianz die Betrachtung von Konzepten zum *Produkt- und Technologielebenszyklus*. *Technische Änderungen* und *Produkt-Features* sind Bestandteil eines *Release* und Planungsgegenstand in der *Release-Planung*. Gleichzeitig sind sie die Grundlage für *Innovationen*, die durch die zu erarbeitende Systematik schlussendlich systematisch erschlossen werden sollen.

2.1.1 Systematik

Gegenstand dieser Arbeit ist eine *Systematik zur Release-Planung intelligenter technischer Systeme*. Der Begriff **Systematik** ist in der Wissenschaft insb. als Fachbereich in der Biologie etabliert. Die Biosystematik befasst sich mit der Kategorisierung, Bezeichnung und Identifizierung von Organismen [LL06], [Dum10, S. 5f.]. In der Statistik dient die Systematik der durchgängigen Ordnung von Erhebungs- und Darstellungseinheiten [Gab14-ol]. Im allgemeinen Sprachgebrauch bezeichnet die Systematik die *planmäßige, einheitliche Darstellung oder Gestaltung nach bestimmten Ordnungsprinzipien* [Dud14-

ol], [Bra14, S. 7]. Ausgehend von diesem Begriffsverständnis und in Anlehnung an DUMITRESCU bezeichnet eine Systematik ein universelles Rahmenwerk, bestehend aus nach aufgabenspezifischen Gesichtspunkten geordneten, reproduzierbaren **Vorgehensmodellen** sowie dedizierten Hilfsmitteln zu deren Unterstützung. **Hilfsmittel** können bspw. Methoden, Richtlinien, Spezifikationstechniken, Modellierungssprachen oder Werkzeuge sein [Dum10, S. 5f.]. Diesem Verständnis folgt die vorliegende Arbeit.

2.1.2 Der Systembegriff

Das heutige Systemverständnis fußt auf der allgemeinen, disziplinunabhängigen Systemtheorie nach BERTALANFFY [Ber49, S. 114ff.], [Ber69, S. 38]. Grundbegriffe der Systemtheorie sind in Bild 2-1 dargestellt.

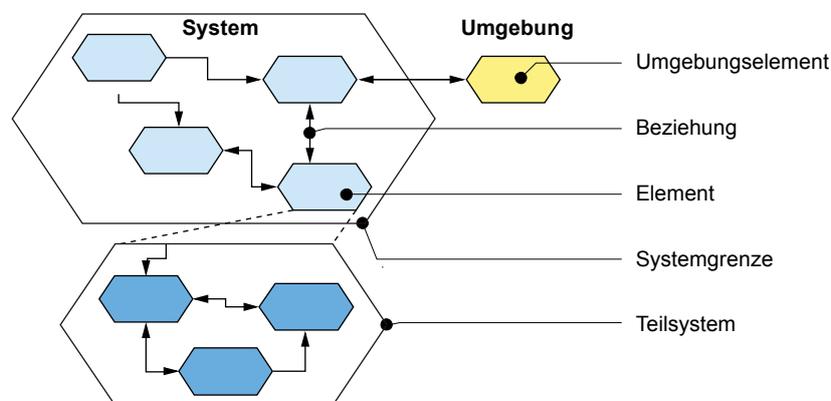


Bild 2-1: Grundbegriffe der Systemtheorie nach [HWF+12, S. 34f]

Ein **System** besteht aus miteinander in **Beziehung** stehenden **Elementen** (Systemelemente). Diese Elemente können ihrerseits wieder als Systeme bzw. **Teilsysteme** betrachtet werden. Sie sind über **Beziehungen** verbunden, bei denen es sich z.B. um Materialflüsse, Informationen und anderweitige Wirkzusammenhänge handelt. Die **Systemgrenze** trennt das System von seiner Umgebung, mit der es ebenfalls über Beziehungen verbunden ist. Die Anordnung der Elemente und Beziehungen beschreibt die Systemstruktur [Ehr09, S. 19ff.], [Rop09, S. 76ff.], [HWF+12, S. 34ff].

Diesen Grundbegriffen folgend weisen sowohl die Produktentstehung³ als auch Produkte einen systemischen Charakter auf und können folglich als System bezeichnet werden. Im Rahmen dieser Arbeit wird der Begriff System synonym mit dem **Produkt** genutzt. Diese werden in Sachleistungen, Dienstleistungen und hybride Leistungen unterschieden. Sie spiegeln das Angebot eines Unternehmens wider und dienen der Bedürfnisbefriedigung des Kunden [Gab15b-ol]. Bei den im Rahmen dieser Arbeit betrachteten intelligenten technischen Systemen handelt es sich um Sachleistungen mit einem signifikanten Anteil an physischen Komponenten.

³ Zum Verständnis des Begriffs Produktentstehung im Rahmen dieser Arbeit siehe Kapitel 2.3.

Das *Denken in Systemen* erleichtert das Verständnis und den Umgang mit komplexen Systemen; es wird in unterschiedlichen Fachbereichen angewendet (z.B. in der Managementlehre [Ves02], [Mal15]) und ist wichtiger Bestandteil des Systems Engineering-Konzepts [HWF+12, S. 33] (vgl. Abschnitt 2.5.1). In Bezug auf technische Systeme liegt der wesentliche Nutzen in der Vereinfachung von Planung, Entwicklung und Konstruktion [DH02, S. 9ff.], [Ehr09, S. 26], [Bra14, S. 8]. Ein wesentliches Prinzip ist dabei die Arbeit mit Modellen, die durch Abstraktion und Vereinfachung der Realität komplexe Strukturen und Zusammenhänge veranschaulichen [HWF+12, S. 41]. Dies wird im Kontext des Model-Based Systems Engineerings explizit aufgegriffen.

2.1.3 Komplexität und Unsicherheit

Komplexität ist ein im Unternehmenskontext und insb. in der Produktentstehung häufig verwendeter Begriff, der typischerweise zur Charakterisierung eines Systems genutzt wird. Dabei werden sowohl Produkte als auch die Entwicklung selbst als komplex bezeichnet [LMB09, S. 1]. Für eine differenzierte Betrachtung des Komplexitätsbegriffs bietet sich die Unterscheidung in zwei Dimensionen an [UP95], [Sch05, S. 5], [HWF+12, S. 40f]: 1) Vielfalt, Vielzahl, Größe und 2) Dynamik, Veränderbarkeit. Diese Dimensionen werden auch als strukturelle und dynamische Komplexität bezeichnet. Die Gegenüberstellung in einer Matrix ergibt vier Systemtypen (Bild 2-2).

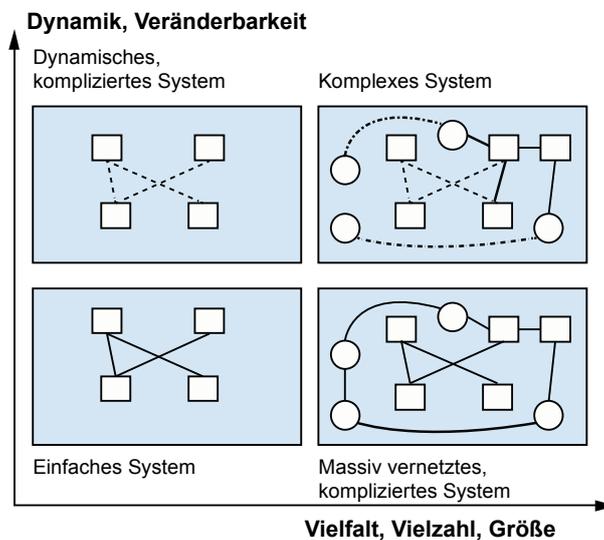


Bild 2-2: Systemtypen nach [HWF+12, S. 40f] und [UP95]

Einfache Systeme weisen lediglich wenige Elemente mit geringer Beziehungsintensität auf. Nimmt die Anzahl der Elemente und Beziehungen zu, handelt es sich um ein **massiv vernetztes, kompliziertes System**. Beide Systemklassen unterscheiden sich folglich lediglich in ihrer strukturellen Komplexität. Ein System mit hoher dynamischer Komplexität lässt sich hingegen als **dynamisches, kompliziertes System** bezeichnen. Es besteht aus Verbindungen, die sich zeitlich verändern. Bei hoher struktureller und dynamischer Komplexität werden Systeme als **komplex** bezeichnet [HWF+12, S. 40f].

Die Veränderlichkeit bzw. Dynamik eines Systems wird entscheidend beeinflusst durch die mit dem System verbundene **Unsicherheit**. Viele wissenschaftliche Ansätze betrachten Unsicherheit daher als einen Typ von Komplexität und bemühen sich um eine Klassifikation [Wil99], [EE05], [GMW10, S. 985ff]. EARL und ECKERT unterscheiden bspw. vier Klassen: bekannte und unbekante⁴ Unsicherheit sowie Datenunsicherheit und Unsicherheit in der Beschreibung bzw. Spezifikation eines Systems [EE05]. DE WECK ET AL. analysieren Unsicherheiten in Bezug auf ihren Einfluss auf die Produktentwicklung und gliedern diese nach ihrer Beeinflussbarkeit (Bild 2-3).

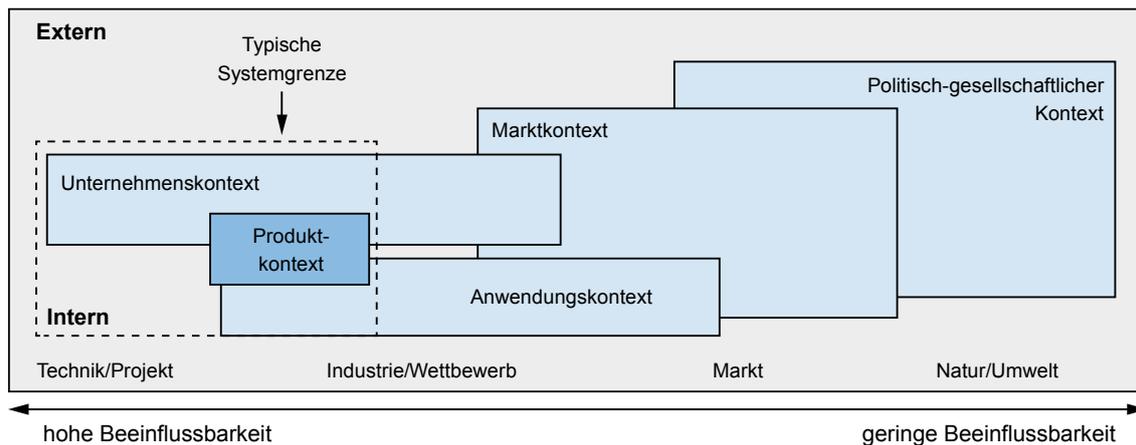


Bild 2-3: Interne und externe Unsicherheiten in der Produktentwicklung [CWC09, S. 3], [WEC07, S. 4]

Die Beeinflussbarkeit durch die Produktentwicklung sinkt mit Erweiterung des Betrachtungshorizonts. Gut beeinflussbare Unsicherheiten werden als **interne Unsicherheiten** bezeichnet. Der **Produktkontext** ist maßgeblich durch Unsicherheiten resultierend aus dem Neuheitsgrad eines Projekts geprägt. Diese Unsicherheiten treten trotz Wiederverwendung von Elementen aus vergangenen Entwicklungsprojekten auf. Gründe sind z.B. veränderte Schnittstellen zur Systemumgebung. Auch schlecht verstandene Wechselwirkungen zwischen Systemelementen sorgen z.B. durch Fehlerfortpflanzung für Probleme innerhalb des gesamten Produktlebenszyklus. Weitere Unsicherheiten ergeben sich aus dem **Unternehmenskontext**. Quellen sind Fehlplanungen in der Produktstrategie, späte Änderungsanforderungen und unklar definierte Projektziele [WEC07, S. 4f].

Neben internen Unsicherheiten beeinflussen **externe Unsicherheiten** die Produktentwicklung. Diese resultieren aus Anwendung, Markt, Politik und Umwelt. Unsicherheiten im **Anwendungskontext** ergeben sich z.B. aus der Nutzungsweise eines Produkts. Unsicherheiten im **Marktkontext** resultieren aus der zukünftigen Marktentwicklung und aus

⁴ Unbekannte Unsicherheiten sind nahezu unvorhersehbare Ereignisse. Selbst der Fakt, dass eine Unsicherheit existiert, ist in diesem Fall nicht bekannt. Stichworte sind *unknown unknowns* oder *black swans* [OS11]. TALEB diskutiert dieses Phänomen ausführlich in seinem Buch „Der Schwarze Schwan: Die Macht höchst unwahrscheinlicher Ereignisse“ [Tal10].

veränderten Anforderungsprofilen der Kunden. Eine weitere Quelle sind frühere und innovativere Wettbewerbsangebote. Vor allem im **politischen** und **kulturellen Kontext** sind Unsicherheiten kaum zu beeinflussen. Ein Beispiel sind gesetzlich vorgeschriebene Änderungen von Normen und Standards [WEC07, S. 5f].

Die Entwicklung mechatronischer Systeme ist bereits heute durch hohe strukturelle Komplexität und vielfältige Unsicherheiten geprägt. Sowohl die strukturelle Komplexität als auch die Unsicherheiten nehmen mit dem Wandel zu intelligenten technischen Systemen zu. Während sich das Systems Engineering der Beherrschung der strukturellen Komplexität widmet, zielt die Release-Planung auf die Beherrschung der dynamischen Komplexität ab. Hierauf wird in Abschnitt 2.2 näher eingegangen.

2.1.4 Varianz technischer Systeme

Technische Systeme bzw. Produkte zeichnen sich typischerweise durch hohe Varianz aus, die u.a. der Bedienung kundenindividueller Anforderungen und der Differenzierung im Wettbewerb dient. In der Literatur finden sich verschiedene Begrifflichkeiten zur Klassifizierung und Hierarchisierung von Produktvarianten. Ein wesentliches Merkmal ist die Unterscheidung in räumliche und zeitliche Varianz (Bild 2-4) [MI02].

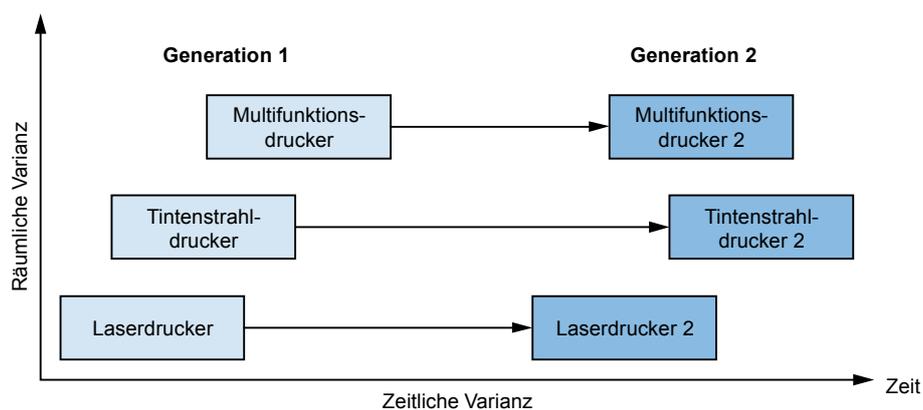


Bild 2-4: Räumliche und zeitliche Varianz nach [MI02]

Die **räumliche Varianz** entspricht der zu einem Zeitpunkt am Markt angebotenen Produktvielfalt [MI02]. Die Gesamtheit der durch ein Unternehmen am Markt angebotenen Produkte wird unter dem Begriff **Produktprogramm** bzw. Produktportfolio zusammengefasst [Ble11, S. 7f.], [PL11, S. 249]. Die Gliederung des Produktprogramms erfolgt durch **Produktfamilien**, in denen eine Menge von Produkten zusammengefasst wird, die sich durch gemeinsame Funktionalitäten und einen ähnlichen Anwendungsbereich auszeichnen [ML97], [Göp98], [Ble11, S. 7f]. Jedes spezifische Familienmitglied ist eine **Produktvariante** [JSS07, S. 7]. Volatile Märkte, der Trend zu kundenindividuellen Produkten und eine immer stärkere Marktsegmentierung führen in vielen Unternehmen zu einer Zunahme der räumlichen Varianz und enden häufig im sogenannten „Varianten-

Teufelskreis“⁵ [Rat93, S. 53], [Sch05, S. 148]. Der Bewältigung dieser Variantenvielfalt widmet sich das Variantenmanagement. Ziel des **Variantenmanagements** ist es, die vom Markt geforderten und Erfolg versprechenden Varianten zu antizipieren und die Entwicklung zu beauftragen, diesen durch konstruktive Maßnahmen wirtschaftlich zu begegnen. Typische Entwicklungsstrategien hierfür sind Baukastensysteme und Produktplattformen [GEK01, S. 166ff], [Sch05, S. 119].

Veränderte Randbedingungen, ausgelöst durch Kunden, Wettbewerber und technologische Weiterentwicklung, führen gleichzeitig jedoch auch zu einer Evolution⁶ des Produkts [BK13, S. 211]. Diese spiegelt sich in der Realisierung neuer Funktionen und zusätzlicher Anforderungen wider und führt zur **zeitlichen Varianz** bzw. Generationsvarianz. Zeitliche Varianten werden durch **Produktgenerationen** beschrieben, bei denen je Generation die Kernfunktionalität identisch bleibt; Technologien, Funktionen und Erscheinungsbild jedoch erweitert und aktualisiert werden [LK14, S. 79], [SL05], [KDK06], [ABW15, S. 8f.]. Eine Produktgeneration bildet die Basis für eine Menge sukzessiv abgeleiteter **Produktversionen**, die sich durch einen vergleichsweise geringeren technischen Änderungsgrad bzw. Innovationsbeitrag auszeichnen. In diesem Zusammenhang spricht die Literatur häufig auch von einem Facelift [HOL+11], [OHS+12, S. 126].

Die Planung von Produktgenerationen und -versionen erfolgt im Rahmen der Release-Planung und ist von den Aufgaben des Versions- und Konfigurationsmanagements zu unterscheiden. Das **Versionsmanagement** verwaltet in einer zeitlichen Reihenfolge stehende Versionen von Anforderungen und Ergebnisdokumenten, die zu einer Produktversion gehören. Das **Konfigurationsmanagement** verknüpft eine Produktversion mit der zugehörigen Konfiguration von Dokumenten und stellt sicher, dass alle Beteiligten die richtige Dokumentenversion erhalten [Sch12, S. 185], [GP14, S. 372].

2.1.5 Produkt- und Technologielebenszyklus

Der Begriff **Lebenszyklus** beschreibt die zeitliche Entwicklung der Existenz sowohl von Produkten als auch von Technologien. Unterschiedliche Sichtweisen auf diese zeitliche Entwicklung resultieren in drei Konzepten für den Produktlebenszyklus [FG13, S. 296f]: 1) den intrinsischen Produktlebenszyklus, 2) den betriebswirtschaftlichen Produktlebenszyklus und 3) den technologischen Produktlebenszyklus. Letzterer ist eng mit dem Technologielebenszyklus verknüpft.

5 Der „Varianten-Teufelskreis“ besagt, dass durch eine zunehmende Marktsegmentierung die Produktvielfalt im Unternehmen und damit die Stückkosten in der Produktion steigen. Konsequenz sind Wettbewerbsnachteile, die auf lange Sicht zum Verlust der Wettbewerbsfähigkeit führen [Rat93].

6 Evolution bezeichnet *den Prozess einer schrittweisen Änderung innerhalb eines Systems [...], Produkts, etc. [...] von einem einfacheren zu einem komplexeren oder fortschrittlicheren Zustand* [OED14-ol], [KST12, S. 107]. Die Evolution von Produkten meint somit die kontinuierliche Fortentwicklung und Verbesserung eines Produkts im Produktlebenszyklus.

Der **intrinsische Produktlebenszyklus** (Produktlebenslauf) beschreibt die Abfolge von Situationen, die ein Produkt von der ersten Idee bis zur Rücknahme durchläuft. Dies umfasst die strategische Produktplanung, Produkt- und Produktionssystementwicklung, Fertigung, Distribution und Nutzung [GLR+00, S. 3], [GK12, S. 42], [FG13, S. 297]. Verschiedene methodische Ansätze und Arbeitstechniken der Produktentwicklung (z.B. Product Lifecycle Management (PLM) und Lebenszykluskostenrechnung) basieren auf diesem Verständnis [AA13], [Hep13, S. 57ff.].

Der **betriebswirtschaftliche Produktlebenszyklus** adressiert die Phasen, die ein Produkt während seiner Marktpräsenz durchläuft. Jede Phase weist dabei eine charakteristische Entwicklung der Kenngrößen Umsatz, Ergebnis und Entwicklungskosten auf (Bild 2-5) [FG13, S. 296], [Pei15, S. 23].

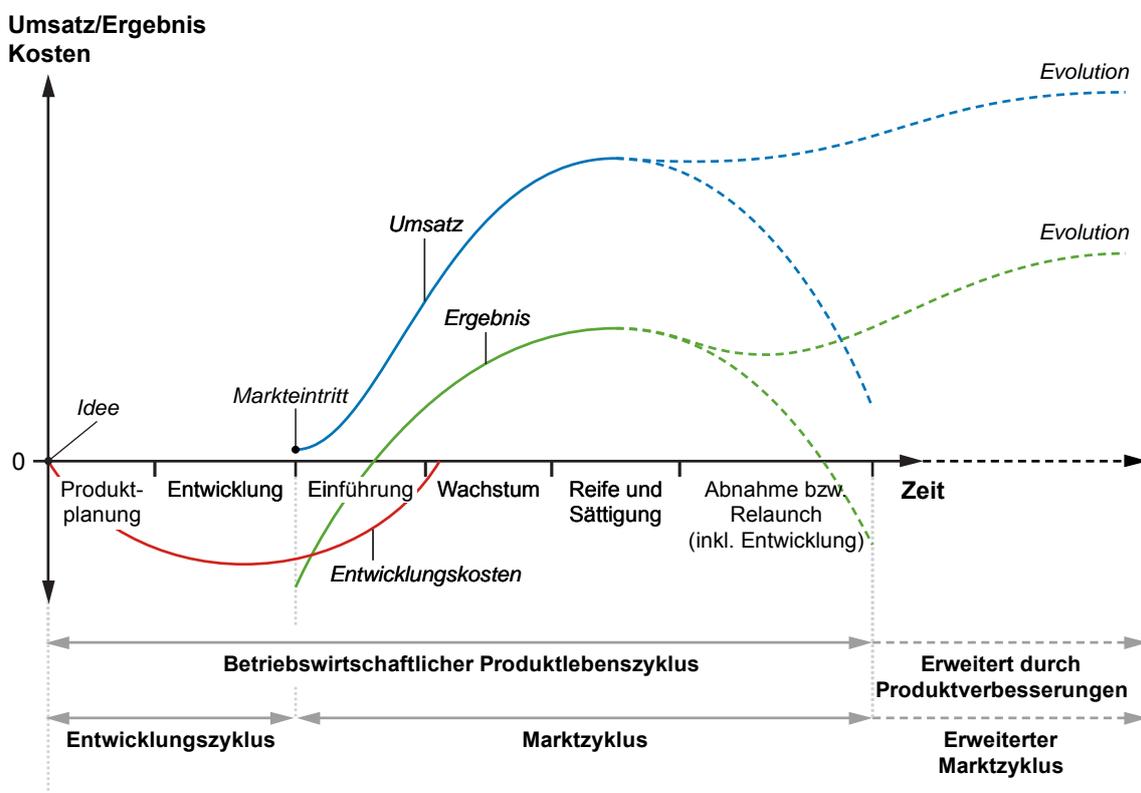


Bild 2-5: Betriebswirtschaftlicher Produktlebenszyklus nach [BH01, S. 134], [FG13, S. 296], [Pei15, S. 23]

Die mit der Planung und Entwicklung des Produkts verbundenen Kosten resultieren in einem Verlust, der mit Markteinführung durch erste Umsätze kompensiert wird. Bei einem erfolgreichen Produkt steigt dieser Umsatz in der Wachstums- und Reifephase kontinuierlich an. Ergebnis ist ein Gewinn, der in der Sättigungs- bzw. Verfallsphase durch sinkende Umsätze wieder abnimmt [FG13, S. 296], [BH01, S. 134]. Die **Dauer des betriebswirtschaftlichen Produktlebenszyklus** variiert je nach Produktart und Branche. Ziel von Maßnahmen zur Produktverbesserung und -erweiterung (Produktpflege) ist es, den Produktlebenszyklus zu verlängern und damit den wirtschaftlichen Gewinn zu vergrößern [FG13, S. 296]. Diesen Aspekt berücksichtigen FORTUIN und OMTA im

Produktgenerationslebenszyklus (engl. Product Generation Life Cycle), der auf dem Konzept des betriebswirtschaftlichen Produktlebenszyklus aufbaut. Der Produktgenerationslebenszyklus bezeichnet die Gesamtheit der Lebenszyklen aller Produktversionen einer Produktgeneration [FO07, S. 3]. In den vergangenen Jahren ist insb. in der Konsumgüterindustrie ein Trend zu **verkürzten Produktlebenszyklen** zu erkennen. Gründe sind ein verändertes Käufer- und Anbieterverhalten: Käufer adoptieren Neuerungen schneller; Anbieter sorgen bedingt durch den rasanten technologischen Fortschritt kontinuierlich für Innovationen, die vorhandene Produkte gänzlich ersetzen. Beides verkürzt die Zeit bis zur Verfallsphase eines Produkts [BH01, S. 136].

Der **technologische Produktlebenszyklus** beschreibt die technologische Weiterentwicklung eines Produkts einschließlich der daraus resultierenden Produktgenerationen. Dies basiert auf folgender Grundidee: Die Leistungsfähigkeit eines Produkts folgt zeitversetzt der Leistungsfähigkeit der für das Produkt charakteristischen Technologie. Der technologische Produktlebenszyklus ist somit eng mit dem Technologielebenszyklus verknüpft [FG13, S. 297]. Dem **Technologielebenszyklus** liegt die Annahme zu Grunde, dass sich die Leistungsfähigkeit einer Technologie entsprechend einer S-Kurve steigert. Die Technologie durchläuft dabei drei Phasen: Schrittmachertechnologie, Schlüsseltechnologie und Basistechnologie [Bul94, S. 113 ff.] [GP14, S. 132 f.]. Handelt es sich bei der für das Produkt charakteristischen Technologie also noch um eine Schrittmachertechnologie, ist davon auszugehen, dass auch die Leistungsfähigkeit des Produkts in den darauffolgenden Produktgenerationen noch erheblich gesteigert werden kann. Im Umkehrschluss ist im Falle einer Basistechnologie nur noch eine geringfügige Leistungssteigerung zu erwarten. In diesem Fall empfiehlt sich der Umstieg auf alternative, leistungsfähigere Technologien [FG13, S. 297].

Die Planung, Steuerung und Kontrolle von Lebenszyklen wird als **Zyklusmanagement** bezeichnet [VLR14]. Hierbei handelt es sich um ein aktuelles Forschungsfeld⁷, in das sich die zu entwickelnde Systematik einordnen lässt. So ist es das Ziel der Release-Planung, ein Produkt kontinuierlich unter Berücksichtigung des betriebswirtschaftlichen und technologischen Produktlebenszyklus weiterzuentwickeln. Beide Sichten auf den Produktlebenszyklus werden somit in dieser Arbeit gleichermaßen adressiert.

2.1.6 Technische Änderungen und Produkt-Features

Die meisten Produkte sind heute lediglich Weiterentwicklungen bestehender Systeme. Dies erlaubt die Fokussierung auf zentrale Produktverbesserungen und reduziert Risiken und Kosten des Entwicklungsprojekts [Cro89], [SFM+09, S. 1], [Bor10], [ABW15, S. 8f.]. Als Konsequenz prägen technische Änderungen an bestehenden Systemen das Innovations- und Entwicklungsgeschehen [JEC+11, S. 105f.].

⁷ Dem Forschungsfeld widmet sich u.a. der 2012 gestartete Sonderforschungsbereich 768 „Zyklusmanagement von Innovationsprozessen“ der TU München (Homepage: www.sfb768.tum.de).

Technische Änderungen (engl. Engineering Change) bezeichnen, dem Verständnis nach JARETT ET AL. folgend, Änderungen an Bauteilen, Zeichnungen oder Software, die bereits freigegeben wurden. Sie sind unabhängig vom Änderungsumfang und reichen von kleinen Modifikationen einzelner Komponenten bis hin zu großen Änderungen mit Auswirkungen auf das gesamte Produkt. Technische Änderungen treten während des gesamten Produktlebenszyklus auf, sind dabei allerdings explizit von typischen Iterationen in der Produktentwicklung zu unterscheiden [JEC+11, S. 105f.]. Auch Modifikationen, Verbesserungen und Aufrüstungen von im Feld befindlichen Produkten sind nicht gemeint. Grundsätzlich dient eine Änderung der Produktverbesserung. Lediglich die Änderungsursache variiert von der Problembhebung bis zur Anpassung des Produkts auf neue Kundenanforderungen [ECZ04, S. 1]. Es wird daher auch in **fehlerbedingte** und **neuheitsbedingte** Änderungen unterschieden [Con97], [LR98, S. 327].

Ein Grund für neuheitsbedingte Änderungen ist die Realisierung neuer **Produkt-Features**. Das Feature-Konzept stammt aus der Softwaretechnik und dient der Planung und Entwicklung von Produktlinien. Danach stellt ein Feature⁹ ein eindeutig identifizierbares Produktmerkmal dar [CE00], [KBD+14]. Hierbei kann es sich sowohl um mechanische, mechatronische als auch um rein softwarebasierte Produktmerkmale handeln. Ein Feature dient der Erfüllung eines konsistenten Bündels funktionaler und/oder nicht-funktionaler Anforderungen¹⁰ eines oder mehrerer Stakeholder [CZZ+05]. Features bilden die Brücke von der Marketing- zur Entwicklerperspektive. Im Bereich des Marketings werden Features als besondere Eigenschaften eines Produkts dargestellt – aus Entwicklersicht hingegen sind Features ein Bündel umzusetzender Produkthanforderungen [FHS02]. Die Deklaration eines Features erfolgt durch ein einzelnes Wort oder durch eine kurze Beschreibung. Das Feature-Konzept fördert die Beschreibung und Kommunikation der Variabilität von Produkten für interne und externe Stakeholder und unterstützt so die Strukturierung, Wiederverwendung und Variation eines Produkts über alle Phasen des Lebenszyklus [ABK+13, S. 18f.], [CZZ+05]. Features und die daraus resultierenden technischen Änderungen sind wesentlicher Planungsgegenstand in der Release-Planung.

2.1.7 Release, Release-Planung und Release-Management

Der Begriff **Release** fällt häufig im Zusammenhang mit Änderung und Features. Umgangssprachlich meint der Anglizismus die Veröffentlichung eines Produkts, insb. einer neuen oder überarbeiteten Software [Dud15-ol], [OED15-ol]. Die aus der Software-

⁸ Wenn im Rahmen dieser Arbeit von Änderungen gesprochen wird, sind stets technische Änderungen gemeint. Auf Abweichungen von dieser Regel wird explizit hingewiesen.

⁹ Der Begriff Feature im Kontext der Produktplanung ist von der Feature-Technologie als Unterstützungsmittel im Bereich der 3D-CAD-Technik zu unterscheiden (vgl. z.B. [FG13, S. 421ff.]).

¹⁰ Das Deutsche Institut für Normung (DIN) definiert eine Anforderung als: „*Beschaffenheit, Fähigkeit oder Leistung, die ein Produkt, Prozess oder die am Prozess beteiligte Person erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.*“ [DIN69901].

technik stammende Definition von SOMMERVILLE konkretisiert dies. Er bezeichnet ein Release als *eine Systemversion, die an den Kunden ausgeliefert wird* [Som07, S. 750]. RUHE hingegen legt den Blickwinkel auf den Inhalt bzw. den Umfang eines Release. Danach ist ein Release *eine Sammlung von neuen und/oder geänderten Features, die eine neue Version eines evolutionären Produkts bilden* [Ruh05]. Dies ähnelt dem Verständnis nach SCHUH ET AL. Diese betonen zusätzlich, dass es sich bei einem Release um ein *harmonisiertes Bündel von Verbesserungen eines Produkts* handelt [SDL+04], das gemeinsam entwickelt, getestet und freigegeben wird [Sch05, S. 113]. Diesem Verständnis folgend repräsentiert ein Release den Freigabestand eines Produkts, der immer auch zu einer neuen Produktversion führt.

Der Umfang eines Release wird im Rahmen der **Release-Planung** festgelegt [Ruh10, S. 3]. Diese ist nach RUHE ein Management-Instrument zur Planung der Evolution eines Produkts. Ziel ist es, den Markt- und Kundenbedürfnissen bestmöglich gerecht zu werden. Dabei erfolgt die Planung sowohl strategisch als auch operativ unter Berücksichtigung von Restriktionen in Bezug auf Markt, Technik und Organisation [Ruh10, S. XII]. Die Release-Planung stellt ein unternehmensweites Optimierungsproblem dar, in das eine Vielzahl von Stakeholdern involviert ist [Lin07, S. 2]. Die Release-Planung ist Teil des **Release-Managements**, welches nach SCHUH gleichbedeutend mit der Produktpflege ist. Es umfasst neben der Release-Planung auch das Konfigurations- und Versionsmanagement sowie das Testen [Sch12, S. 12].

Ergebnis der Release-Planung ist der **Release-Plan** (vgl. hierzu auch Abschnitt 2.4.1.). Dieser beschreibt die Evolution des Produkts durch Verbesserungen, Erweiterungen und Ergänzungen über den Produktlebenszyklus. Nach GERACIE und EPPINGER besteht er aus vier konstituierenden Elementen [GE13, S. 192]:

- 1) einem definierten **Planungshorizont**,
- 2) **Release-Terminen**, die in Abhängigkeit spezifischer Marktereignisse (z.B. Messen, Saisonabhängigkeiten) und anderer Deadlines definiert werden,
- 3) spezifischen **technischen Änderungen** und **Produkt-Feature**, die diesen Releases zugeordnet sind
- 4) sowie damit verbundenen **Entwicklungsaktivitäten** und **Ressourcen**.

Die Umsetzung eines Release erfolgt in einem **Release-Projekt**¹¹, das typischen Restriktionen im Hinblick auf Kosten, Zeit und Ressourcen unterliegt und, je nach Umfang des Releases, dem typischen Vorgehen in der Systementwicklung folgt – von der Anforderungsdefinition über die Systemintegration bis zum verbesserten bzw. erweiterten Produkt (vgl. hierzu auch Abschnitt 2.5.2).

¹¹ Die DIN 69901-5 definiert ein Projekt als ein *Vorhaben, das im Wesentlichen durch Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist* [DIN 69901-5].

2.1.8 Invention und Innovation

Der von SCHUMPETER eingeführte Innovationsbegriff (lat. innovatio = Erneuerung, Veränderung) unterliegt unterschiedlichen Definitionen und Abgrenzungen [Sch61, S. 91], [HS07, S. 3ff.], [BT07, S. 29ff.], [Sch12, S. 1ff.]. Gemeinhin werden dabei sowohl die hohe Bedeutung für den Unternehmenserfolg als auch die Unterscheidung von Invention und Innovation betont. Eine **Invention** ist nach BULLINGER *das Ergebnis von Technologieinsatz und Intuition des Erfinders* [Bul94, S. 35]. Ist diese Invention am Markt erfolgreich, handelt es sich um eine **Innovation** [HS07, S. 3ff.], [Sch12, S. 2]. Der **Innovationsgrad** eines Produkts bewegt sich auf einem Kontinuum zwischen einer geringfügigen und einer fundamentalen Änderung des bisher Dagewesenen [Ger05, S. 40f.]. HENDERSON und CLARK konkretisieren dies und klassifizieren Innovationen entsprechend ihrer Auswirkungen auf das Produkt. Sie unterscheiden dazu zwei Dimensionen (Bild 2-6): Die Erneuerung einzelner Systemelemente und die Veränderung der Beziehungen zwischen diesen Systemelementen [HC90, S. 11ff.].

Beziehungen zwischen Systemelementen/Systemarchitektur

verändert	Architektonische Innovation	Radikale Innovation
	Inkrementelle Innovation	Modulare Innovation
unverändert		
	bewährt	erneuert
	Systemelemente	

Bild 2-6: Innovationen klassifiziert nach dem Grad ihrer Änderungsauswirkungen auf das Produkt [HC90, S. 12]

Entsprechend dieser Klassifikation werden vier Innovationstypen unterschieden: **Inkrementelle Innovationen** sind Verfeinerungen und Erweiterungen eines bestehenden Produkts. Einzelne Systemelemente werden verbessert, ohne das Grundkonzept und die Beziehungen zwischen den Systemelementen zu verändern. Im Gegensatz dazu unterliegen bei **radikalen Innovationen** sowohl die Beziehungen als auch die Systemelemente selbst grundlegenden Änderungen. Erfolgt eine Änderung nur in einer dieser Dimensionen, wird von **modularen** bzw. **architektonischen Innovationen** gesprochen [HC90, S. 11ff.]. Änderungen der Struktur und der Systemelemente eines Produkts haben demnach hohen Einfluss auf den Innovationsgrad. Dies gilt es bei der Release-Planung und der damit verbundenen Klassifizierung und Umsetzung von technischen Änderungen zu berücksichtigen.

2.2 Von mechatronischen zu intelligenten technischen Systemen

Moderne technische Erzeugnisse zeichnen sich durch einen zunehmenden Anteil von Elektronik und Software aus. Diese sogenannten **mechatronischen Systeme** sind wesentliche Innovationstreiber – nicht nur in der Automobilindustrie, sondern auch in

verwandten Branchen wie dem Maschinen- und Anlagenbau und der Elektroindustrie. Beispiele sind Fahrerassistenzsysteme, automatische Werkzeugwechselsysteme und dezentrale Automatisierungskomponenten, die eine flexible Gestaltung von Fertigungsprozessen erlauben. Mechatronische Systeme bieten hohes Potential für Erfolg versprechende Produktinnovationen. Sie ermöglichen verbesserte Produktfunktionen, erhöhte Zuverlässigkeit und die Reduzierung von Kosten. Darüber hinaus bieten sie die Möglichkeit zur Integration vollständig neuer Funktionen [Möh04, S. 3], [Ise08, S. 21].

Mit der rasanten Entwicklung der Informations- und Kommunikationstechnologie zeichnet sich ausgehend von der Mechatronik ein Trend hin zu **intelligenten technischen Systemen** ab, der in Stichworten wie Selbstoptimierung [GRS14], Internet of Things and Services [aca14] und Cyber-Physical Systems [GB12] zum Ausdruck kommt. Charakteristika für intelligente technische Systeme sind eine inhärente Teilintelligenz sowie die hohe informationstechnische Vernetzung [GDJ+14]. Der technologische Wandel durch diese Systeme führt insb. in den oben genannten Branchen sowohl zu evolutionären als auch zu disruptiven Änderungen. Diese drücken sich in einer kontinuierlichen Weiterentwicklung und damit in einer **Evolution** technischer Systeme aus. Dabei gehen die Änderungen teilweise über reine Produktinnovationen hinaus und eröffnen das Potential für neue Märkte und innovative Geschäftsmodelle [KM05], [GB12, S. 9], [aca14]. Vor diesem Hintergrund erfolgt in den nächsten Abschnitten eine Diskussion dieses technologischen Wandels und seiner Konsequenzen für die Produktentstehung. Ausgehend von der Grundstruktur und den Klassen mechatronischer Systeme (**Abschnitt 2.2.1**) werden intelligente technische Systeme und die damit verbundenen Innovationspotentiale beschrieben (**Abschnitt 2.2.2**). Anschließend werden die wesentlichen Treiber diskutiert, die die Weiterentwicklung und damit die Evolution technischer Systeme beeinflussen (**Abschnitt 2.2.3**).

2.2.1 Grundstruktur und Klassen mechatronischer Systeme

Den Begriff **Mechatronik** als ein Kunstwort aus Mechanik und Elektronik prägten in seiner heutigen Definition HARASHIMA, TOMIZUKA und FUKUDA [HTF96, S. 1]. Die VDI-RICHTLINIE 2206 „Entwicklungsmethodik für mechatronische Systeme“ greift diese Definition auf und übersetzt sie wie folgt:

„Mechatronik bezeichnet das synergetische Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik beim Entwurf und der Herstellung industrieller Erzeugnisse sowie bei der Prozessgestaltung.“ [VDI2206, S. 14].

Sowohl der Begriff als auch die Definition betonen das Zusammenwirken klassischer Ingenieursdisziplinen zur Realisierung dieser Systeme. Einen gemeinsamen Bezugspunkt bildet die ebenfalls in der VDI-Richtlinie 2206 beschriebene **Grundstruktur** bzw. **Referenzarchitektur** mechatronischer Systeme (Bild 2-7).

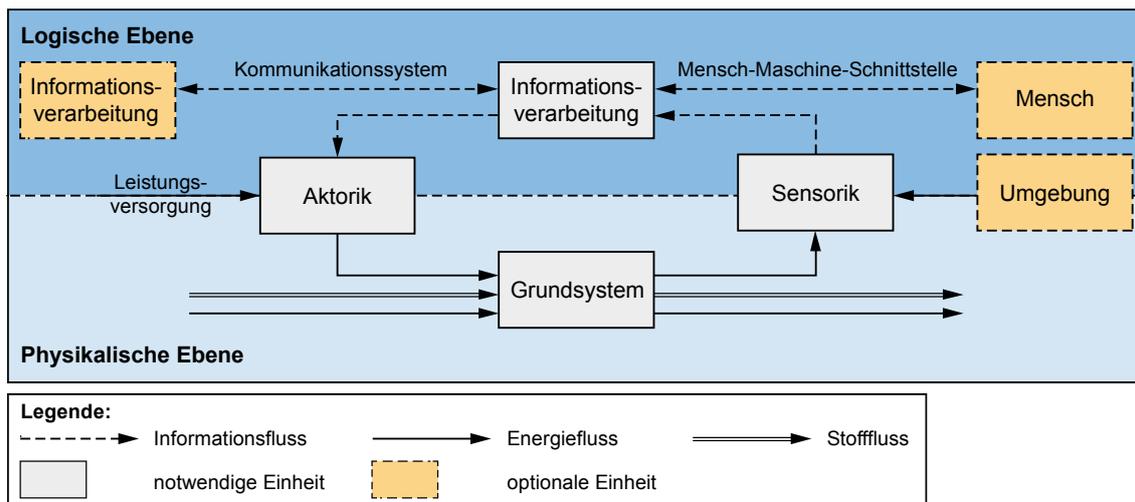


Bild 2-7: Grundstruktur bzw. Referenzarchitektur eines mechatronischen Systems [VDI2206, S. 14]

Demnach bestehen mechatronische Systeme in ihrer einfachsten Form aus vier Einheiten: einem Grundsystem, Sensor(en), Aktor(en) und der Informationsverarbeitung. Die vier Einheiten bilden das mechatronische Regelungssystem und sind immer in ein Umfeld eingebettet. Sie weisen **zwei Schnittstellen** nach außen auf: 1) eine Mensch-Maschine-Schnittstelle für die Interaktion mit dem Menschen und 2) ein Kommunikationssystem zum Informationsaustausch mit anderen (technischen) Systemen [VDI2206, S. 14f].

Das **Grundsystem** gibt die zentrale physikalische Grundausrichtung des Systems vor. Hierbei handelt es sich entweder um eine mechanisch, elektromechanisch, hydraulisch oder pneumatisch geprägte Struktur oder um eine Kombination daraus. Ausgewählte Zustandsgrößen des Grundsystems und optional auch äußere Einflüsse aus der Umgebung erfasst die **Sensorik**, die für unterschiedlichste physikalische Größen (z.B. Temperatur, Druck, Schall) verfügbar ist. Die erfassten Messwerte gehen, ergänzt um Informationen anderer technischer Systeme oder des Benutzers, als Eingangsgrößen an die **Informationsverarbeitung**. Diese bestimmt die Stellgrößen für die Aktorik. Die **Aktorik** setzt die Stellgrößen um und schließt den Regelkreis durch die geeignete Beeinflussung des Grundsystems [VDI2206, S. 14f]. Beispiele für Aktoren sind elektrische Antriebe, Pumpen und Pneumatikzylinder. Die beschriebenen Einheiten des mechatronischen Systems sind über Flüsse miteinander verbunden. PAHL/BEITZ unterscheiden drei Flussarten [PBF+07, S. 43]: 1) **Stoffflüsse** beschreiben den Transfer von Fluiden und Festkörpern, 2) **Energieflüsse** die Übertragung von Energien (z.B. Strom, Kraft) und 3) **Informationsflüsse** den Austausch von Stellgrößen, Messwerten und anderen Daten.

Heutige mechatronische Systeme lassen sich allesamt auf die beschriebene Grundstruktur zurückführen. Dennoch erlaubt die Vernetzung von mechatronischen Modulen die Realisierung eines breiten Spektrums unterschiedlicher Systeme. Dieses lässt sich grob in drei Klassen gliedern (Bild 2-8) [GAC+13], [GTS14, S. 26ff.].

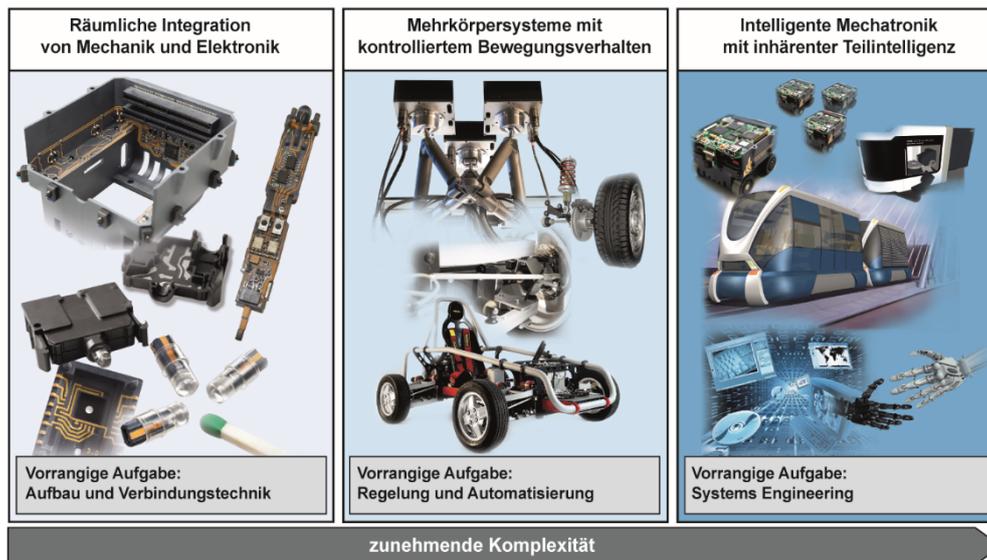


Bild 2-8: Klassen mechatronischer Systeme [GAC+13]

Die **erste Klasse** von Systemen integriert eine hohe Anzahl mechanischer und elektronischer Funktionsträger auf kleinem Bauraum. Dies resultiert in Vorteilen wie verringerten Herstellkosten, erhöhter Zuverlässigkeit, Miniaturisierung und Funktionsintegration. Die vorrangige Aufgabe ist die Aufbau- und Verbindungstechnik [GF06, S. 3f.]. Bei Systemen der **zweiten Klasse** handelt es sich um Mehrkörpersysteme mit kontrolliertem Bewegungsverhalten. Diese können gemäß der beschriebenen Grundstruktur mechatronischer Systeme selbstständig auf Veränderungen des Grundsystems und der Umgebung reagieren. Die vorrangige Aufgabe ist die ganzheitliche Optimierung des Systemverhaltens durch Regelung und Automatisierung [Gau10, S. 17]. Aus der sich abzeichnenden Entwicklung der Informations- und Kommunikationstechnik eröffnet sich die Perspektive für eine **dritte Klasse**: intelligente, vernetzte mechatronische Systeme [GTS14, S. 26ff.]. Diese Klasse mechatronischer Systeme steht im Fokus dieser Arbeit und wird daher im Folgenden gesondert betrachtet.

2.2.2 Intelligente technische Systeme

Die Bezeichnung *intelligente technische Systeme* steht im Rahmen dieser Arbeit stellvertretend für ein **Technologiekonzept**¹², das auf zwei grundlegenden Annahmen basiert: Zukünftige mechatronische Systeme besitzen eine **inhärente Teilintelligenz**, die weit über die heutige Informationsverarbeitung hinausgeht. Grundlage sind Fortschritte in der Informationstechnik und die Einbindung nichttechnischer Disziplinen¹³. Diese bringen

¹² Das hier beschriebene Technologiekonzept ist Grundlage der Forschungsarbeiten im Spitzencluster it's OWL [GDJ+14]. Es entspricht dem Paradigma cyber-physischer Systeme [GB12].

¹³ Wenn im Rahmen dieser Arbeit von Fachdisziplinen die Rede ist, schließt dies neben den klassischen Ingenieursdisziplinen (Mechanik, Elektrotechnik, Elektronik und Softwaretechnik) immer auch weitere an der Entwicklung beteiligte Disziplinen ein. Dies sind z.B. die Kognitionswissenschaften und die Neurobiologie, aber auch der Kunde bzw. das Produktmanagement.

vielfältige Ansätze hervor, mit denen neue sensorische, aktorische und kognitive Funktionen in technische Systeme integriert werden. Die zweite Annahme des Technologiekonzepts ist die zunehmende **Vernetzung der Systeme**. Getrieben durch Fortschritte in der Kommunikationstechnik kommunizieren und kooperieren zukünftige Systeme mit dem Ziel, eine gemeinsame Aufgabe zu erfüllen. Erst durch das Zusammenspiel der Einzelsysteme erschließt sich die Funktionalität des Gesamtsystems [GAC+13], [GDJ+14], [GRS14, S. 1]. Systeme dieser Art zeichnen sich durch folgende Eigenschaften aus [Dum10], [GAC+13]:

- **Adaptiv:** Die Systeme interagieren mit dem Umfeld und passen sich diesem autonom an. Dies erfolgt in einem vom Entwickler vorgesehenen Rahmen.
- **Robust:** Die Systeme bewältigen auch unerwartete und vom Entwickler nicht berücksichtigte Situationen. Unsicherheiten oder fehlende Informationen können bis zu einem gewissen Grad ausgeglichen werden.
- **Vorausschauend:** Auf der Basis von Erfahrungswissen antizipieren diese Systeme die künftigen Wirkungen von Einflüssen und mögliche Zustände. Auf diese Weise werden Gefahren frühzeitig erkannt und rechtzeitig die passenden Strategien zu ihrer Bewältigung ausgewählt.
- **Benutzungsfreundlich:** Die Systeme passen sich dem spezifischen Benutzerverhalten an und stehen in Interaktion mit dem Benutzer. Trotz steigender Komplexität bleibt ihr Verhalten für den Benutzer dabei stets nachvollziehbar.

PORTER und HEPPELMANN wählen eine andere Art der Charakterisierung intelligenter technischer Systeme und strukturieren deren Fähigkeiten in die vier Stufen Monitoring, Control, Optimization und Autonomy. Sie betonen dabei, dass jede dieser Fähigkeiten für sich bereits einen Kundennutzen bietet, gleichzeitig aber auch Voraussetzung für die nächste Stufe ist. So sind z.B. Überwachungsfunktionen (Monitoring) die Voraussetzung für die Steuerung und Optimierung des Produkts [PH14, S. 7ff.]. Bei der Einführung derartiger Fähigkeiten bzw. Features sind somit stets potentielle Abhängigkeiten zu überprüfen und zu berücksichtigen. Gleichzeitig deutet die Strukturierung in Stufen bereits auf den evolutionären Charakter intelligenter technischer Systeme hin: Bestimmte Produkt-Features und damit verbundene Fähigkeiten werden ausgehend von einer Basisfunktionalität sukzessive ergänzt.

Die beschriebenen Charakteristika eröffnen faszinierende **Innovations- und Nutzenpotentiale**. Zu deren Identifikation bietet sich die Unterscheidung in drei Suchfelder an: Funktionsverbesserung sowie horizontale und vertikale Funktionsintegration¹⁴ (Bild 2-9).

¹⁴ Das Schema der horizontalen und vertikalen Integration ist aus den Wirtschaftswissenschaften im Kontext von Unternehmenskonzentrationen übertragen. Hier meint die horizontale Integration den Zusammenschluss von Unternehmen gleicher Produktionsstufe, während die vertikale Integration Unternehmen vor- oder nachgelagerter Produktionsstufen zusammenfasst [Gab15a-01].

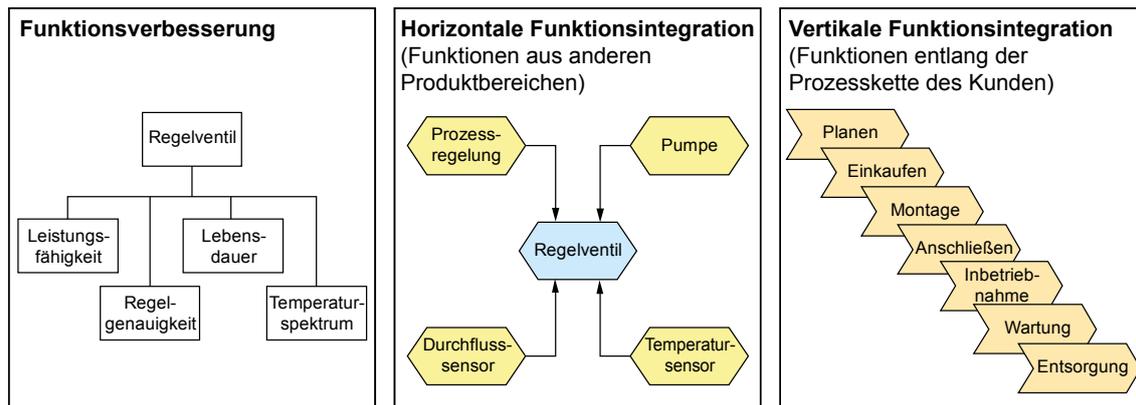


Bild 2-9: Funktionsverbesserung, horizontale und vertikale Funktionsintegration

Funktionsverbesserung: Diese zielt auf die Verbesserung der Kernfunktionen eines Systems im Hinblick auf Qualitätskriterien wie Leistungsfähigkeit, Zuverlässigkeit und Effizienz ab. Bspw. können durch die Integration von Selbstoptimierungsverfahren in einen Bohrprozess erhebliche Energieeinsparungen sowie eine verbesserte Bohrqualität und Effizienz erzielt werden [itsowl15, S. 18].

Horizontale Funktionsintegration: Ziel ist die Integration von Funktionen, die ursprünglich durch andere Systeme im Produktumfeld erfüllt werden. Beispiele sind Automatisierungskomponenten (z.B. Antriebe, Signalwandler), die durch eine eigene Informationsverarbeitung zunehmend Aufgaben einer übergeordneten Steuerung übernehmen [itsowl15, S. 17]. Ein weiteres Beispiel ist die Integration von Kameras in Smartphones.

Vertikale Funktionsintegration: Hiermit ist die Integration von Funktionen gemeint, die produktlebenszyklusrelevante Aufgaben unterstützen bzw. ersetzen. Typische Aufgaben finden sich in den Phasen Inbetriebnahme und Wartung. Stichworte sind Selbstkalibrierung und Predictive Maintenance bzw. Selbstdiagnose. Ein Beispiel sind selbstjustierende Fahrzeugscheinwerfer, die den ansonsten aufwändigen Justageprozess des Scheinwerfers automatisieren [itsowl15, S. 21].

Die Suchfelder für Innovationspotentiale gelten unabhängig von Produktart und Branche. **Anwendungsfelder** für intelligente technische Systeme finden sich somit in nahezu allen technikorientierten Branchen. Die Anwendungsfelder werden häufig unter dem Namenszusatz *Smart* diskutiert (z.B. Smart Home, Smart Mobility und Smart Health). Durch die massiven Auswirkungen auf die Zukunftsperspektiven der betroffenen Branchen widmen sich dem Thema Initiativen aus Industrie, Wissenschaft und Politik. Ein Beispiel ist das **Zukunftsprojekt Industrie 4.0** der Bundesregierung. Im Rahmen dieser Initiative werden die wirtschaftlichen, technologischen und gesellschaftlichen Auswirkungen des konsequenten und allgegenwärtigen Einsatzes intelligenter technischer Systeme in der Produktion diskutiert [FA13].

2.2.3 Evolution intelligenter technischer Systeme

Die Entwicklung technischer Systeme ist bereits heute durch hohe Komplexität geprägt, die mit dem Wandel zu intelligenten technischen Systemen weiter zunehmen wird. Schnittstellenvielfalt, Interdisziplinarität, Spezialisierung sowie die daraus folgende Zulieferintegration sind Beispiele für zunehmende strukturelle Komplexität auf Produkt- und Organisationsebene [Wil99], [LMB09, S. 3ff.]. Hinzu kommen hohe Dynamik und damit verbundene Unsicherheiten, die sich auf drei wesentliche Treiber zurückzuführen lassen: 1) technologischen Fortschritt, 2) dynamische Systemumgebungen und 3) volatile Märkte [WBF+97], [FS05], [LDA11, S. 2f.]. Alle drei Treiber führen zu einer rasanten Evolution technischer Systeme, die sich in mannigfachen Produktversionen und -generationen ausdrückt.

1) Technologischer Fortschritt: Die Anzahl und die Leistungsfähigkeit möglicher Produktfunktionen in einem System nimmt durch die Fortschritte insb. in der Informations- und Kommunikationstechnologie rapide zu. Die mit der Integration dieser Technologien verbundenen Innovationspotentiale müssen fortwährend ausgeschöpft werden, um ein Produkt konkurrenzfähig am Markt anbieten zu können [Jan98], [FS05]. Resultat sind häufige technologiebedingte Änderungen, die zu verkürzten Innovationszyklen führen. Diese verkürzten Innovationszyklen sind insb. auf den gestiegenen Anteil von Software und Elektronik zurückzuführen, die weitaus öfter aktualisiert werden als die Mechanik eines Systems (Bild 2-10).

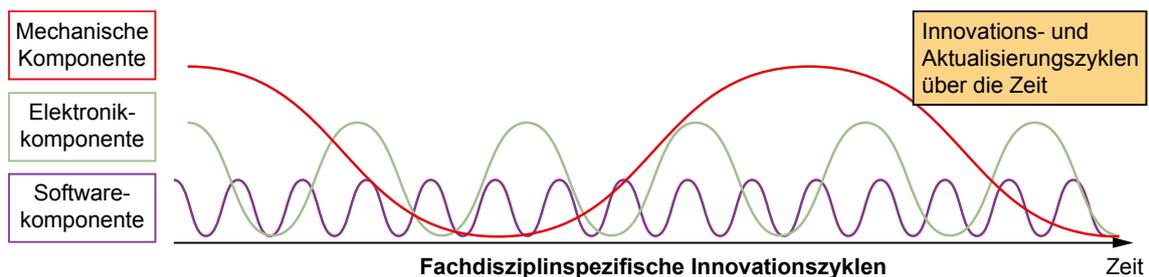


Bild 2-10: Qualitative Darstellung von Innovationszyklen nach fachdisziplinspezifischen Schwerpunkt in Anlehnung an [EB08, S. 2]

Während bspw. der mechanisch geprägte Antriebsstrang eines Fahrzeugs häufig für mehrere Jahre weitestgehend unverändert bleibt, halten elektronische Komponenten selten länger als 18 Monate [SE04]. Softwarekomponenten veralten oftmals sogar noch schneller. Die Richtung zeigen hier die Zyklen der Consumer-IT auf [aca14, S. 21f.]. Zwar sind deren Entwicklungszeiten nicht zwingend kürzer, doch unterliegen sie nicht im gleichen Maße technischen Restriktionen wie mechanische oder elektronische Komponenten. Gleichzeitig bedarf es nahezu keines Aufwands für die Produktion, so dass sich Software unkompliziert aktualisieren, vervielfältigen und in das Produkt integrieren lässt [Bro10, S. 20f.]. Das führt zu noch kürzeren Innovationszyklen und zum sogenannten „Innovation-Cycle-Dilemma“. Dieses beschreibt die Herausforderung, die Innovationszyklen

verschiedener Fachdisziplinen wie Mechanik, Elektronik und Software zu synchronisieren [LSS+07, S. 526ff.].

2) Dynamische Systemumgebungen: Intelligente technische Systeme sind in der Regel das Resultat der Komposition von Sub-Systemen und Komponenten. Diese werden teils von Zulieferern bezogen, die ihrerseits bestrebt sind, die eigenen Systeme weiterzuentwickeln. Abkündigungen und Innovationen bei Zukaufkomponenten führen so automatisch auch zu Änderungen am eigenen Produkt. Hinzu kommt, dass intelligente technische Systeme zunehmend vernetzt in einem sogenannten „System of Systems“¹⁵ agieren. Wechselseitige Beziehungen mit anderen technischen Systemen, die ebenfalls einem rasanten technologischen Wandel unterliegen, führen so ebenfalls zu unvorhersehbaren Änderungen [FS05].

3) Volatile Märkte: Globalisierung, Individualisierung und heterogene Nischenmärkte charakterisieren heute und zukünftig die Weltmärkte. Dies zwingt Unternehmen, die Produkte kontinuierlich auf sich ändernde und teils unvorhergesehene Kundenbedürfnisse anzupassen [FS05]. Gleiches gilt für Reaktionen auf verbesserte Wettbewerbsprodukte und die Änderung von Normen, Richtlinien und Gesetzen.

Die beschriebenen Komplexitätstreiber spiegeln die hohe Dynamik in der Systementwicklung wider, die zu stetig neuen Anforderungen führt. Kontinuierliche Änderungen, Anpassungen und Verbesserungen über den gesamten Produktlebenszyklus¹⁶ sind das Ergebnis. Schon aus Effizienzgründen kann diesen neuen Anforderungen nicht ausschließlich mit Neuentwicklungen begegnet werden, da die Investitionen in ein System von Jahr zu Jahr zunehmen. Deutlich macht dies bspw. der Softwareumfang technischer Systeme, der sich in Anlehnung an das Mooresche-Gesetz¹⁷ exponentiell entwickelt und stetig umfangreicher wird [LDA11, S. 2]. Die vollständige Ablösung und Neuentwicklung des Systems wird somit immer teurer und risikoreicher. Als Konsequenz prägt die **evolutionäre Weiterentwicklung** bestehender Systeme das Innovations- und Entwicklungsschehen [Cro89], [SFM+09], [Bor10], [LDA11].

Bei der Weiterentwicklung bestehender Systeme stehen inkrementelle Innovationen im Vordergrund, die zu neuen Sekundärfunktionen bei gleichbleibender Basisfunktionalität führen. Bestimmte Funktionseinheiten werden auf Basis neuer Lösungsansätze verändert oder ergänzt, während andere Teilsysteme unverändert von früheren Produktversionen übernommen werden. Hierbei bleiben die Struktur des Produkts und damit die Systemarchitektur häufig erhalten [ABW15, S. 1ff.]. Dennoch können die Aufwände zur Realisie-

¹⁵ „System of Systems“ (SoS) bezeichnen Systeme, deren Systemelemente wiederum selbst Systeme sind; durch die Vernetzung mehrerer heterogener und verteilter Systeme resultieren verschiedene interdisziplinäre Herausforderungen [INC12, S. 349].

¹⁶ Gemeint sind hier der betriebswirtschaftliche und technologische Produktlebenszyklus.

¹⁷ Benannt nach G. MOORE, einem der Mitbegründer der Firma INTEL, besagt das Gesetz, dass sich die Prozessorleistung alle 18 bis 24 Monaten verdoppelt.

rung einzelner Änderungen immens sein. So kann die Neugestaltung einzelner Teilsysteme in unerwarteten Aufwänden resultieren, die sich durch Änderungsfortpflanzungen aufgrund bekannter oder unbekannter technischer Abhängigkeiten ergeben [BB08]. Durch die zunehmende Komplexität intelligenter technischer Systeme nehmen diese technischen Abhängigkeiten zu, so dass die Prognose von Änderungsaufwänden noch schwieriger wird. Zur Vermeidung zusätzlicher Aufwände und ungewollter Nebeneffekte gilt es, die Weiterentwicklung technischer Systeme zu systematisieren. Dies resultiert in der Notwendigkeit, Rahmenbedingungen für die Evolution von Systemen zu schaffen und diese durch abgestimmte Innovations- und Anpassungsschritte gezielt zu steuern [FS05], [GB12, S. 100]. Die systematische Steuerung von Innovations- und Anpassungsschritten ist Aufgabe der Release-Planung.

2.2.4 Fazit und Einordnung der zu entwickelnden Systematik

Der Wandel von mechatronischen zu intelligenten technischen Systemen bietet vielfältige Innovationspotentiale, führt gleichzeitig jedoch zu einer hohen Komplexität und Dynamik in der Systementwicklung. Die Komplexität steigt durch die Einbindung verschiedener Fachdisziplinen sowie die zunehmende Vernetzung innerhalb der Systeme und zwischen den Systemen. Die Dynamik ist auf den rasanten technologischen Fortschritt, dynamische Systemumgebungen und volatile Märkte zurückzuführen. Resultat sind technische Änderungen, die es in zunehmend komplexeren Entwicklungsprojekten umzusetzen gilt. Dies stellt neue Anforderungen an die Planung und Durchführung dieser Entwicklungsprojekte. Es werden Methoden und Werkzeuge benötigt, die einerseits die systematische Weiterentwicklung technischer Systeme steuern und andererseits die Systementwicklung bei steigender Interdisziplinarität unterstützen. Dabei gilt es im Sinne eines interdisziplinären Systemverständnisses, die Abhängigkeiten zwischen Produkt-Features sowie die Auswirkungen der daraus resultierenden Änderungen zu berücksichtigen. Ansätze hierzu liefern die Release-Planung sowie das Systems Engineering als Grundlage für eine interdisziplinäre Systementwicklung. In diese Handlungsfelder ordnet sich die zu entwickelnde Systematik ein.

2.3 Der Produktentstehungsprozess nach GAUSEMEIER

Die zu entwickelnde Systematik dient der Unterstützung von Planung, (Weiter-)Entwicklung und Pflege intelligenter technischer Systeme. Dies macht die Einordnung in deren Entstehungsprozess erforderlich. Der **Produktentstehungsprozess** beschreibt nach GAUSEMEIER¹⁸ das Vorgehen von der Geschäfts- bzw. Produktidee bis zum

¹⁸ In der Literatur finden sich unterschiedliche Interpretationen des Produktentstehungsprozesses (PEP), die sich im Wesentlichen durch die berücksichtigten Phasen bzw. Teilprozesse unterscheiden. EHRLENSPIEL fasst bspw. auch die Fertigung als Teil des Produktentstehungsprozesses auf; der PEP endet hier mit der Auslieferung an den Kunden [Ehr09, S. 1]. Diese Arbeit folgt dem Verständnis nach GAUSEMEIER.

Serienanlauf; er gliedert sich in drei Hauptaufgabenbereiche: Strategische Produktplanung, Produktentwicklung und Produktionssystementwicklung (Bild 2-11) [GP14, S. 25].

Von der Geschäftsidee...

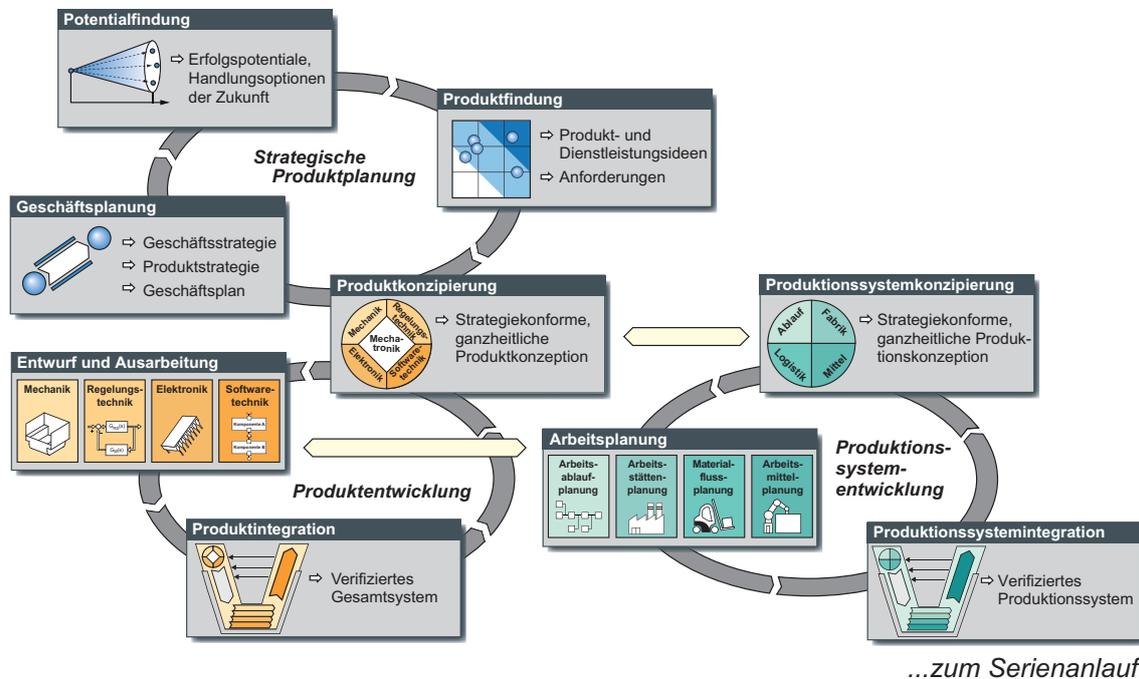


Bild 2-11: Drei-Zyklus-Modell der Produktentstehung [GP14, S. 26]

Die **strategische Produktplanung** ist der eigentlichen Produktentwicklung vorangestellt. Ziel ist die Konkretisierung von Erfolgspotentialen zu Erfolg versprechenden Produktkonzepten. Die **Potentiafindung** dient der Identifikation zukünftiger Erfolgspotentiale und entsprechender Handlungsoptionen. Hier gilt es, ausgehend von der heutigen Markt- und Wettbewerbssituation künftige Entwicklungen zu antizipieren, um externe Unsicherheiten insb. in der Geschäftsplanung berücksichtigen zu können. Zur Erschließung der identifizierten Potentiale werden in der **Produktfindung** Produktideen gesucht, ausgewählt und durch erste Anforderungsspezifikationen dokumentiert. Die Ideen reichen von vollständig neuen Produktideen (radikalen Innovationen) bis hin zu inkrementellen Verbesserungen bestehender Produkte. In der **Geschäftsplanung** wird davon ausgehend die Geschäftsstrategie erarbeitet. Diese definiert, welche Marktsegmente bearbeitet werden sollen. Auf dieser Grundlage erfolgt die Erarbeitung der Produktstrategie, welche u. a. Aussagen zur Gestaltung des Produktprogramms sowie zur Programmpflege über den Produktlebenszyklus enthält. Die Programm- bzw. Produktpflege ist Aufgabe des Release-Managements und zielt auf die fortwährende Verbesserung bereits am Marktpositionierter Produkte ab [Sch12, S. 12]. Die Produktstrategie mündet schließlich im Geschäftsplan, der den Nachweis über die Wirtschaftlichkeit anhand eines angemessenen Return on Investment (RoI) erbringen muss [GP14, S. 25f].

Die **Produktkonzipierung** ist das Bindeglied zum zweiten Zyklus – der **Produktentwicklung**. Ziel sind Erfolg versprechende Produktkonzepte, die sich zur Umsetzung der im Rahmen der Geschäftsplanung festgelegten Geschäfts- und Produktstrategie eignen [GP14, S. 25f]. Wesentliche Herausforderung im Rahmen der Produktkonzipierung ist die Gestaltung einer geeigneten Systemarchitektur¹⁹. Die Systemarchitektur hat weitreichende Effekte für den gesamten Produktlebenszyklus. Sie beeinflusst z.B. die Flexibilität gegenüber Änderungen und die Zuverlässigkeit des Systems [Ulr95]. Die Systemarchitektur sowie die damit verbundenen Produktkonzepte werden in Zusammenarbeit von Fachleuten aus Produktplanung, Vertrieb, Entwicklung/Konstruktion und Fertigungsplanung/Fertigung erarbeitet und dienen als Ausgangspunkt für den **Entwurf** und die **Ausarbeitung** in den Fachdisziplinen Mechanik, Regelungstechnik, Elektronik und Softwaretechnik. Im Rahmen der **Produktintegration** werden die Ergebnisse der einzelnen Fachdisziplinen zu einer Gesamtlösung integriert [GP14, S. 26]. Verschiedene Vorgehensmodelle (z.B. die VDI 2206) konkretisieren die typischen Aufgabenbereiche in der Produktentwicklung (vgl. Abschnitt 2.5.2).

Die parallele Anordnung des dritten Zyklus – der **Produktionssystementwicklung** – unterstreicht die Notwendigkeit einer integrativen Entwicklung von Produkt und Produktionssystem [GLL12]. Ausgangspunkt ist die **Produktionssystemkonzipierung**, die im Wechselspiel mit der Produktkonzipierung erfolgt. Die anschließende **Arbeitsplanung** integriert die vier Fachgebiete Arbeitsablaufplanung, Arbeitsstättenplanung, Materialflussplanung und Arbeitsmittelplanung. Analog zur Produktintegration werden im Rahmen der **Produktionssystemintegration** die einzelnen Ergebnisse der vier Fachgebiete zu einer Gesamtlösung integriert [GP14, S. 26f].

Die dargestellten Aufgabenbereiche sind in drei Zyklen gegliedert, die den iterativen Charakter und das Wechselspiel zwischen den Aufgaben unterstreichen. Je nach Konkretisierungsgrad des zu entwickelnden Systems sind mit den Aufgabenbereichen unterschiedliche Aktivitäten verbunden, die nach ALBERS und GAUSEMEIER aufgrund des systemischen Charakters der Produktentstehung als Ganzes betrachtet werden müssen [AG12, S. 21]. Den integrativen Charakter der Aufgabenbereiche spiegelt Bild 2-12 wider. In den Überschneidungsbereichen ergibt sich das Systems Engineering als gemeinsamer Kooperationskern. Dies wird in Kapitel 2.5 detailliert.

Fazit und Einordnung der zu entwickelnden Systematik: Die *Systematik zur Release-Planung intelligenter technischer Systeme* lässt sich den beiden Aufgabenbereichen Geschäftsplanung und Produktkonzipierung und damit der Schnittstelle von strategischer Produktplanung und Produktentwicklung zuordnen. Die strategische Produktplanung wird im Rahmen dieser Arbeit als ein kontinuierlicher Prozess interpretiert, der stetig Ideen und Konzepte für neue bzw. verbesserte Produkte hervorbringt und somit zu einer Aktualisierung der Produktstrategie führt. Originäre Aufgabe der Release-Planung ist es,

¹⁹ Der Begriff Systemarchitektur und dessen Bedeutung für den Produkterfolg wird in Abschnitt 2.5.3 diskutiert.

die Umsetzung der Produktstrategie in Form eines Release-Plans zu konkretisieren. Dies erfolgt im Wechselspiel mit der Produktkonzipierung, um die Berücksichtigung technischer Restriktionen durch das bestehende System sicherzustellen. Dabei sind technische Abhängigkeiten zwischen Änderungen sowie deren Auswirkungen auf das bestehende Produkt zu überprüfen. Dies setzt ein umfassendes Systemverständnis voraus, welches durch die Berücksichtigung von Ansätzen des Systems Engineering als gemeinsamen Kooperationskern sichergestellt wird.

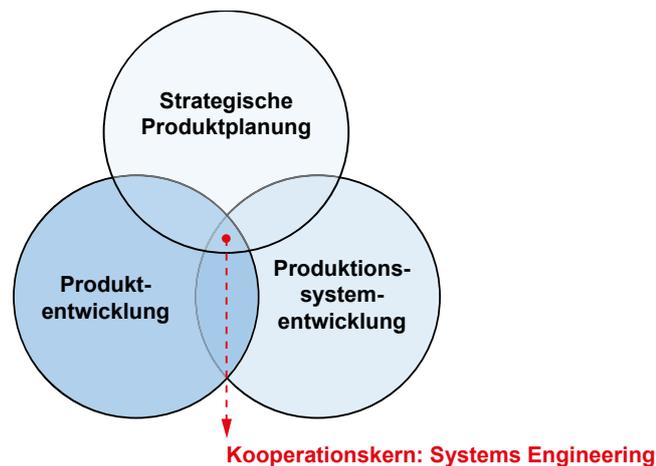


Bild 2-12: Integrativer Charakter der Produktentstehung [AG12, S. 21]

2.4 Release-Planung für technische Systeme

Ein vielversprechender Ansatz für die systematische Evolution technischer Systeme ist das Konzept der Release-Planung. Hierbei handelt es sich um ein Management-Instrument für die lebenszyklusorientierte Umsetzungsplanung von neuen Produkt-Features und technischen Änderungen. Das Konzept der Release-Planung wurde Anfang der 80er Jahre in der Softwareentwicklung geprägt und in jüngster Vergangenheit verstärkt in das produzierende Gewerbe übertragen [SE04], [SDL+04], [BK07], [Bel08], [SAA+13]. So verfolgt bspw. die DAIMLER AG seit mehreren Jahren mit den sogenannten „Änderungsjahren“ eine fest definierte Release-Strategie für alle Baureihen. Diese ermöglicht es, sowohl Innovationen als auch technische Änderungen baureihenübergreifend und gezielt einzusteuern, um so eine systematische Modellpflege zu betreiben [Bel08, S. 49]. Ein weiteres, häufig genanntes Beispiel sind die Smartphone-Serie iPhone und die Tablet-Serie iPad des amerikanischen Konzerns APPLE. Dies wird besonders an dem in 2011 veröffentlichten iPad 2 deutlich. Dieses kann aufgrund eines überschaubaren Innovationsgrades und der offensichtlichen Vorausplanung einzelner Produkt-Features²⁰ als Release der ersten Produktgeneration gewertet werden [Sch13, S. 212ff.].

²⁰ Das neue Produkt-Feature Video-Telefonie des iPad 2 wurde gezielt in der ersten Produktgeneration vorbereitet. So wurde für die notwendige Integration der erforderlichen Kamera sowohl eine spezielle Aufnahme vorgesehen als auch Bauraum ungenutzt gelassen [Sch13, S. 212ff.].

Grundsätzlich ist die Release-Planung als Teil der (strategischen) Produktplanung zu verstehen. Dies wird in **Abschnitt 2.4.1** anhand der Einordnung des zugehörigen Release-Plans in die verschiedenen Ebenen der Produktplanung verdeutlicht. In **Abschnitt 2.4.2** wird davon ausgehend das Grundkonzept der Release-Planung für technische Systeme erläutert. Der Nutzen einer systematischen Release-Planung und die für die Entwicklung in Releases notwendigen Voraussetzungen werden in den **Abschnitten 2.4.3** und **2.4.4** beschrieben. Insgesamt werden so das Handlungsfeld eingegrenzt und Anforderungen an die zu entwickelnde Systematik abgeleitet.

2.4.1 Der Release-Plan im Kontext der Produktplanung

Grundlage erfolgreicher und zielgerichteter Innovationen sind effektive Strategien und Planungen zu Produktportfolio und einzelnen Produkten [DNL96, S. 156]. Die Erarbeitung von Strategien und die Planung von Produkten erfolgen auf unterschiedlichen Ebenen im Unternehmen, so dass in der Literatur verschiedene, sich teils inhaltlich überschneidende Pläne im Kontext der Produktplanung existieren. Vor diesem Hintergrund bietet sich für ein einheitliches Verständnis die Einordnung in das Drei-Zyklen-Modell der Produktentstehung an. Danach erfolgt die Planung und Pflege des Produktprogramms im ersten Zyklus als Teil einer ganzheitlichen Produktstrategie. Davon ausgehend werden Technologie-, Produkt-, Release- und Projektpläne zur Operationalisierung der Strategie erarbeitet, die im zweiten Zyklus durch die Produktentwicklung umgesetzt werden. Bild 2-13 grenzt die Begriffe gegeneinander ab.

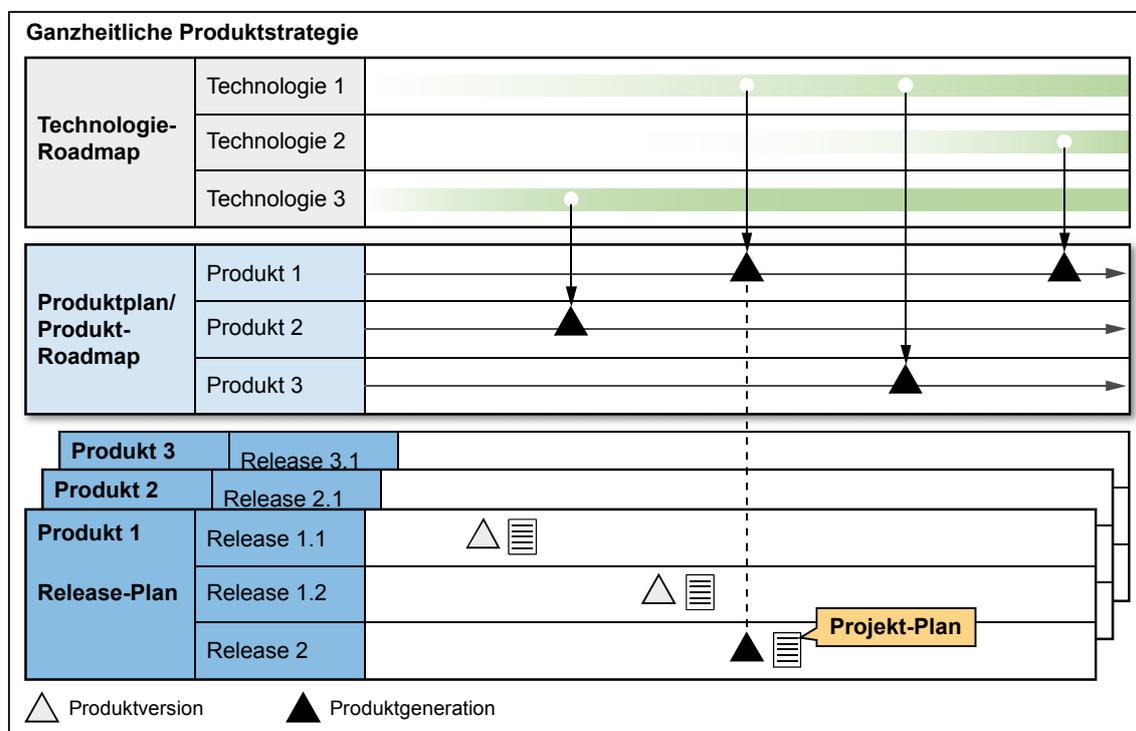


Bild 2-13: Abgrenzung von Produktstrategie, Technologie-Roadmap, Produktplan und Release-Plan in Anlehnung an [Ebe06, S. 859], [UE12, S. 55]

Mit der Unternehmens- bzw. Geschäftsstrategie²¹ werden das Produktprogramm, der Markt, der Wettbewerb, die abgeleitete strategische Stoßrichtung, notwendige Kompetenzen, Ziele, Risiken und Maßnahmen vorgegeben [Pei15, S. 21]. Dies dient als Input für die **Produktstrategie**, die für Produktfamilien oder einzelne Produkte erarbeitet wird. Dies ist meist Aufgabe des Produktmanagements²². Die besondere Bedeutung der Produktstrategie liegt in ihrem bereichsübergreifenden Bezug, durch den sie für alle am Produkt beteiligten Organisationsbereiche eine Handlungsleitlinie für den gesamten Produktlebenszyklus vorgibt [Soe02, S. 11 ff.]. Nach GAUSEMEIER ET AL. sollte eine Produktstrategie dazu Antworten auf folgende Fragen bieten [GEK01, S. 164ff]:

- „*Durch was erfolgt die Differenzierung im Wettbewerb?*“
- „*Wie ist der Entwicklungsprozess und Markteinführungsprozess zu gestalten?*“
- „*Wie kann die vom Markt geforderte Variantenvielfalt bewältigt werden?*“
- „*Wie soll der Wettbewerbsvorsprung über den Produktlebenszyklus erhalten bleiben?*“

Die Frage nach der Erhaltung des Wettbewerbsvorsprungs über den Produktlebenszyklus wird dabei üblicherweise in Produktplänen, Produkt-Roadmaps bzw. Release-Plänen beantwortet. Diese Begriffe sind in der Literatur nicht klar abgegrenzt, beschreiben aber gemeinhin, wann wesentliche Produkt-Features und Produktverbesserungen umgesetzt werden. Bezugspunkt sind dabei sowohl das gesamte Produktprogramm als auch einzelne Produkte. Eine Abgrenzung der Begriffe schlägt LINDGREN vor: Danach haben ein **Produktplan** bzw. die **Produkt-Roadmap** strategischen Charakter und einen starken Marktbezug [Lin08, S. 17ff.]. Sie definieren die zeitliche Umsetzung bereits antizipierter kundenrelevanter Produkt-Features typischerweise für ein ganzes Produktprogramm und dienen der Synchronisation mit der Technologieplanung und der zugehörigen Technologie-Roadmap.

Der **Release-Plan** hingegen dient der Pflege eines spezifischen Produkts bzw. einer Produktgruppe über den Produktlebenszyklus. Neben Vorgaben aus der Produktstrategie einschließlich des zugehörigen Produktplans werden auch fehlerbedingte Änderungen sowie technische Verbesserungen eingeplant, die sich aus der Markteinführung des Produkts ergeben [Lin08, S. 17ff.], [GE13, S. 199ff.]. Der Release-Plan dient somit als Brücke zwischen Produktstrategie bzw. kundenorientierter Produkt-Roadmap auf der einen Seite und der eigentlichen Umsetzung einzelner Produktentwicklungsprojekte auf der anderen Seite. Die Erarbeitung des Release-Plans erfolgt daher auf der Schnittstelle von Produkt- und Projektmanagement. Ziel ist die konkrete Planung der Evolution des Produkts in

21 Geschäftsstrategien sind nur bei größeren Unternehmen mit mehreren Geschäftsfeldern erforderlich und dienen der Konkretisierung der Unternehmensstrategie. Geschäftsstrategien beziehen sich nicht auf das gesamte Produktportfolio, sondern fokussieren eine Gruppe von Produkten, die in der Regel zu Produktfamilien zusammengefasst sind [GP14, S. 114].

22 Das Produktmanagement steuert und koordiniert sämtliche produktrelevante Themen und Aktivitäten (strategisch wie auch operativ) eines Produkts oder einer Produktgruppe – von der Beschaffung bis zu Marketing und Verkauf [Aum13, S. 2ff.].

Form von Produktversionen und -generationen. Dazu trägt der Release-Plan sowohl die Entwicklungsprojekte zur Umsetzung der Releases als auch die zugehörigen Release-spezifischen Markteinführungszeitpunkte auf einer Zeitachse auf. Im Rahmen der Release-Planung werden darüber hinaus durch die Zuordnung von Produkt-Features und technischen Änderungen die konkreten Release-Inhalte definiert. Dies erlaubt die Zuordnung von Entwicklungsressourcen und -budgets und unterstützt somit die effektive Planung von Entwicklungsprojekten [GE13, S. 191ff.].

Die Planung der jeweiligen Entwicklungsprojekte ist Aufgabe des Projektmanagements und mündet in einem Projektplan. Der **Projektplan** hat operativen Charakter und beinhaltet Vorgaben zu Terminen, Organisation, Struktur und Durchführung des Entwicklungsprojekts sowie eine grobe Abschätzung über voraussichtliche Entwicklungsaufwände und -kosten [ANSI/PMI99-001-2004]. Die Projektplanung für das nächste Release beeinflusst maßgeblich den Release-Plan: Häufig übersteigt der anvisierte Release-Umfang die verfügbaren Ressourcen und ist nicht im vorgegebenen Zeitrahmen zu realisieren. In diesem Fall können bestimmte Änderungen und Produkt-Features auf ein späteres Release verschoben werden, um das Risiko einer verzögerten Markteinführung zu reduzieren. Einen ähnlichen Effekt haben veränderte Geschäftsbedingungen: Bringt bspw. ein Wettbewerber ein neues Produkt auf den Markt, führt dies zu einer Aktualisierung der Produktstrategie und damit ebenfalls zur Veränderung der Prioritäten des Release-Plans. Beim Release-Plan handelt es sich daher um ein „lebendes“ Dokument, das in regelmäßigen Abständen in einem systematischen Planungsprozess aktualisiert werden muss [GE13, S. 191ff.].

2.4.2 Das Konzept der Release-Planung

Das **Konzept der Release-Planung** ist Teil des Release-Managements bzw. der Produktpflege, die mit Markteintritt beginnt und in einem kontinuierlichen Prozess Produkte verbessert und verändert. Gründe für Verbesserungen und Änderungen sind vielfältig. Beispiele sind fehlerbedingte Qualitätsverbesserungen, Kosteneinsparpotentiale, Ideen für neue Produkt-Features, Anpassungen an veränderte Kundenbedürfnisse und fortgeschrittene Technologieentwicklungen [JEC+11]. Für eine organisierte Umsetzung derartiger Änderungen²³ existieren in der Literatur verschiedene Richtlinien und Normen, die typischerweise den Umgang mit einer Änderung vom Änderungsantrag bis zur Umsetzung der favorisierten Lösung beschreiben [DIN199-4], [LR98, S. 372], [JEC04]. Die in der Literatur beschriebenen Prozesse unterstützen jedoch lediglich die Umsetzung *einer* Änderung und nicht die planmäßige und systematische Umsetzung *mehrerer* Änderungen. Genau hier setzt das **Kernprinzip der Release-Planung** an: Technische Änderungen werden gemeinsam betrachtet und gebündelt – mit dem Ziel der organisierten Einführung

²³ Auf den Umgang mit *einzelnen* Änderungen in der Systementwicklung wird in Abschnitt 2.5.4 explizit eingegangen, u.a. mit einer Analyse bestehender Änderungsprozesse.

neuer Produktversionen und -generationen. So verhindert die **Bündelung von Änderungen** zu Releases die unkoordinierte Entstehung unzähliger Produktversionen (aufgrund einzelner Änderungen) und reduziert den Gesamtaufwand für Folgeänderungen, notwendige Tests, Anpassungen der Produktdokumentation und die Koordination des Entwicklungsprozesses²⁴ [SE04]. Das abstrakte Modell in Bild 2-14 veranschaulicht dies.

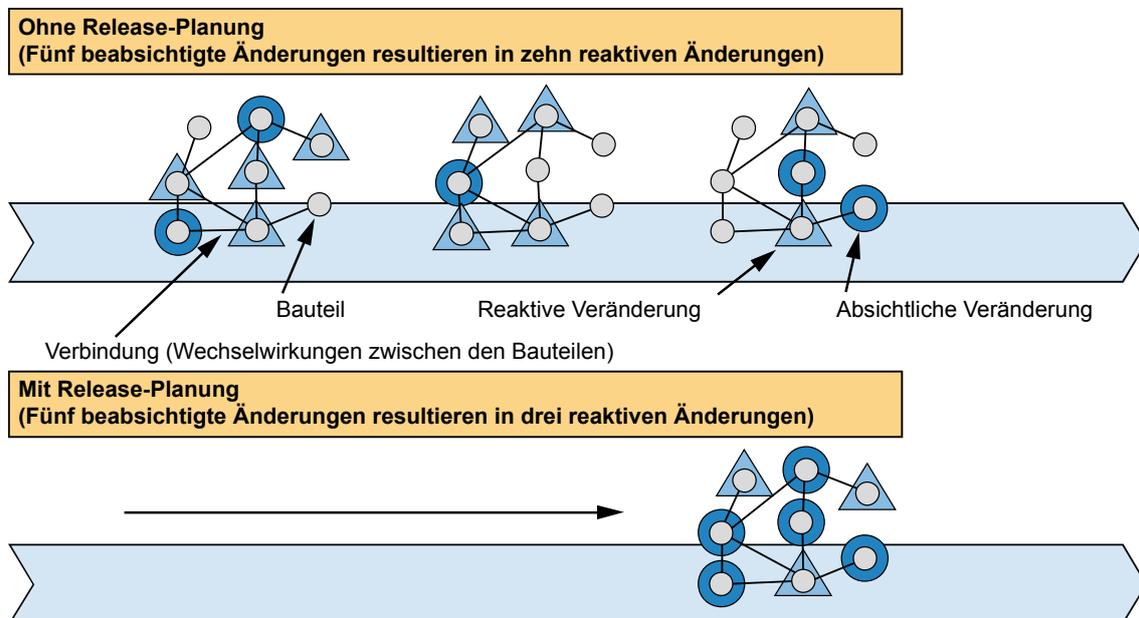


Bild 2-14: Bündelung von Änderungen im Rahmen der Release-Planung [SE04]

Jede geplante Änderung resultiert in Folgeänderungen (z.B. Anpassung weiterer Komponenten bzw. Module aufgrund von technischen Abhängigkeiten, Aktualisierung von Produktdokumentation und Marketingmaterial, Durchführung von Tests etc.). Diese können durch die Bündelung der Änderungen zu einem Release signifikant reduziert werden [SE04]. Die Erschließung der entstehenden Synergiepotentiale setzt allerdings eine systematische Umsetzungsplanung voraus, deren Charakteristika in Bild 2-15 dargestellt sind.

Ausgangspunkt eines systematischen Release-Planungsprozesses ist ein kontinuierlicher Zustrom technischer Änderungen, der sich aus verschiedenen Quellen ergibt (z.B. strategische Produktplanung, Service, Vertrieb, Gesetzgebung etc.). Die Änderungen werden analysiert, bewertet und zu Releases gebündelt. Dies setzt folgende **Entscheidungen im Planungsprozess** voraus:

²⁴ Eine detaillierte Diskussion der Nutzenpotentiale der Release-Planung findet sich in Abschnitt 2.4.3.

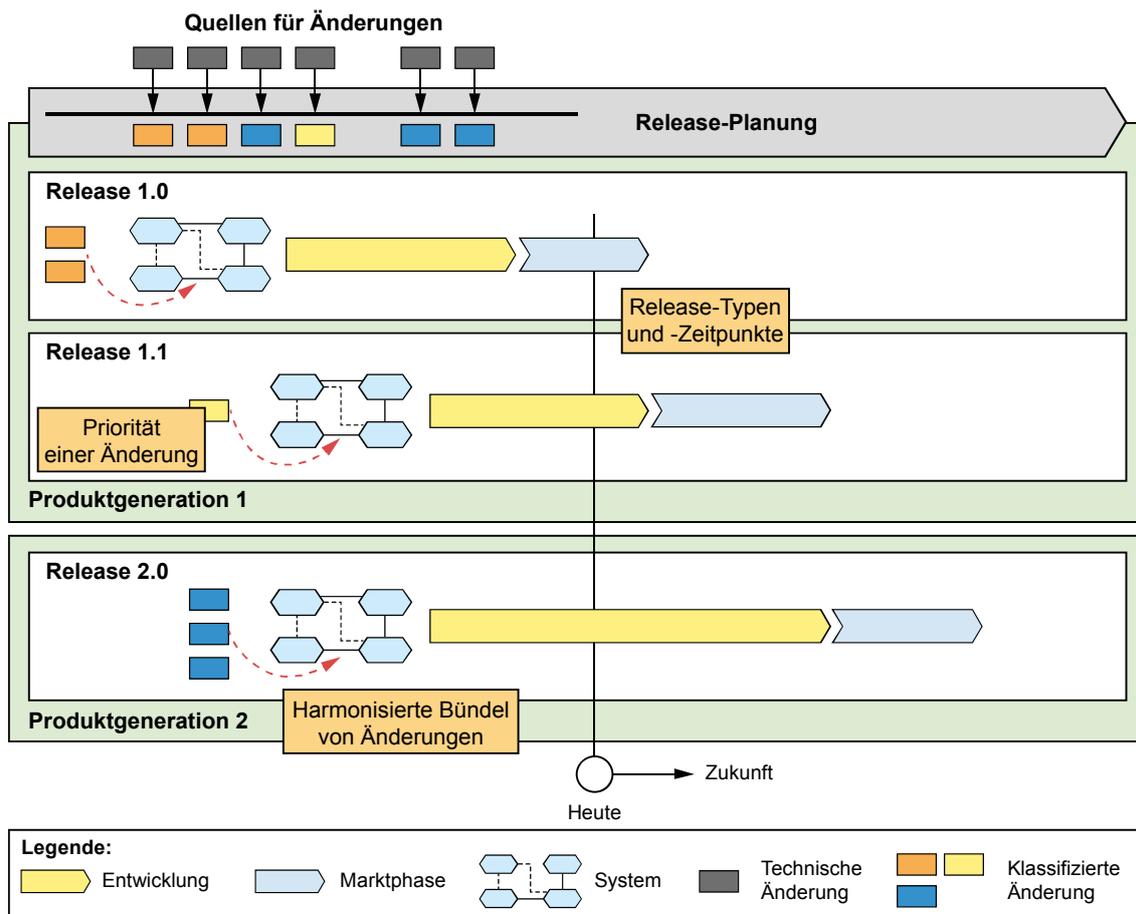


Bild 2-15: Grundkonzept der Release-Planung

- Release-Typen und -Zeitpunkte:** Es ist zu entscheiden, wann und in welchem Umfang neue (teils kundenrelevante) Releases in Form von Produktversionen und -generationen erscheinen. Dies ist auf unternehmensstrategische Ziele abzustimmen, z.B. durch die Berücksichtigung von Vorgaben aus der Produktstrategie [Bel08, S. 47f.]. Ziel ist eine initiale Strukturierung des Release-Plans.
- Priorität einer Änderung:** Die Release-Planung muss entscheiden, wann und in welcher Reihenfolge einzelne Änderungen in das Produkt einfließen. Herausforderung sind konkurrierende Stakeholder-Präferenzen (bspw. steht die von der Produktion angestrebte Kostenminimierung der Forderung des Marketings nach der Kundennutzenmaximierung entgegen) [Ruh10, S. 33ff.]. Darüber hinaus ist zu entscheiden, in welchen Fällen von der Umsetzung in Releases abgewichen wird und eine Änderung per Sofortmaßnahme umgesetzt wird.
- Harmonisierte Bündel von Änderungen:** Neben der Priorität der Änderung sind bei der Bündelung zu Releases weitere Aspekte zu berücksichtigen (z.B. Änderungsabhängigkeiten, Modulzugehörigkeit und Ressourcen-Restriktionen). Je nach strategischen Zielen der Release-Planung sind diese Aspekte bei der Bündelung zu berücksichtigen und zu gewichten [Ruh10, S. 85ff.].

Vor dem Hintergrund dieser notwendigen Entscheidungen dient die Release-Planung der Erhebung und Auswertung planungsrelevanter Informationen. Sie ist als Entscheidungsunterstützung bei der Festlegung von Releases und der darauf aufbauenden Bündelung von Änderungen zu betrachten.

2.4.3 Nutzen einer systematischen Release-Planung

Das Konzept der Release-Planung sieht die Verbesserung von Produkten durch neue Produkt-Features und Änderungen in Form dedizierter und regelmäßiger Release-Projekte vor. Jedes Release führt zu einer inkrementellen Verbesserung des bestehenden Produkts. Dies folgt dem Grundprinzip des Piecemeal Engineerings nach POPPER, wonach die Veränderung bzw. Verbesserung komplexer Systeme Schritt für Schritt anzugehen ist, um eine Nachvollziehbarkeit sicherzustellen [Pop57], [HWF+12, S. 246]. SMITH und REINERTSEN sprechen in diesem Zusammenhang auch vom Ansatz der inkrementellen Innovation [SR98, S. 67ff.]. Die **Nutzenpotentiale** einer systematischen Release-Planung lassen sich in zwei Sichten unterscheiden [Sch13, S. 214ff.]: Die markt- und wettbewerbsstrategische Sicht und die entwicklungsstrategische Sicht.

Markt- und wettbewerbsstrategische Vor- und Nachteile

Eine systematische Release-Planung zielt auf die stufenweise Erweiterung eines Produkts um neue Produkt-Features ab. Dies ermöglicht es, mit der Erstveröffentlichung des Produkts lediglich die Kernfunktionalität der geplanten Produktgeneration umzusetzen. Bestimmte Produkt-Features werden gezielt auf spätere Releases verschoben, mit dem **Vorteil** einer erheblichen Verkürzung der Zeit bis zum Markteintritt (Time-to-Market). Der verkürzte Markteintritt sichert dem Unternehmen frühes Kundenfeedback, auf das schnellstmöglich durch inkrementelle Verbesserungen in Form neuer Produkt-Releases reagiert werden kann [SR98, S. 67ff.]. SANCHEZ und SUDHARSHAN bezeichnen diese Strategie als „Real-Time Market Research“ [SS93]. Der geringere Entwicklungsumfang eines Release ermöglicht darüber hinaus die häufige und fortwährende Veröffentlichung von Produktverbesserungen im Sinne einer Produkt- bzw. „Kampfwertsteigerung“²⁵. Den Vorteil häufiger Releases veranschaulicht Bild 2-16.

Die kontinuierliche Einsteuerung von Innovationen mittels gezielter Release-Projekte sichert einerseits die Aktualität der eigenen Produkte und erhöht andererseits die Reaktionsfähigkeit auf veränderte Markt- und Wettbewerbsbedingungen [WC93], [DNL96, S. 329ff.], [Bel08, S. 49]. Jedes Produkt-Release führt zu einer Erhöhung des vom Kunden wahrgenommenen Innovationsbeitrags²⁶. Veränderte Kundenbedürfnisse können

25 Der Begriff Kampfwertsteigerung entstammt dem militärischen Kontext und meint die Verbesserung der Eigenschaften eines Waffensystems über den Produktlebenszyklus.

26 Nach SCHUH nimmt der wahrgenommene Innovationsbeitrag eines Produkts über die Zeit ab. Gründe sind neue Produkte des Wettbewerbs sowie die Diffusion innovativer Produkt-Features in untere Produktsegmente („Trickle-Down-Effekt“) [Sch05, S. 114].

genauso gezielt bedient werden wie notwendige Anpassungen aufgrund der Einführung von Wettbewerbsprodukten. Marktseitig ergibt sich darüber hinaus Potential für einen Imagegewinn durch stetige Produktverbesserung [SR98, S. 67ff.].

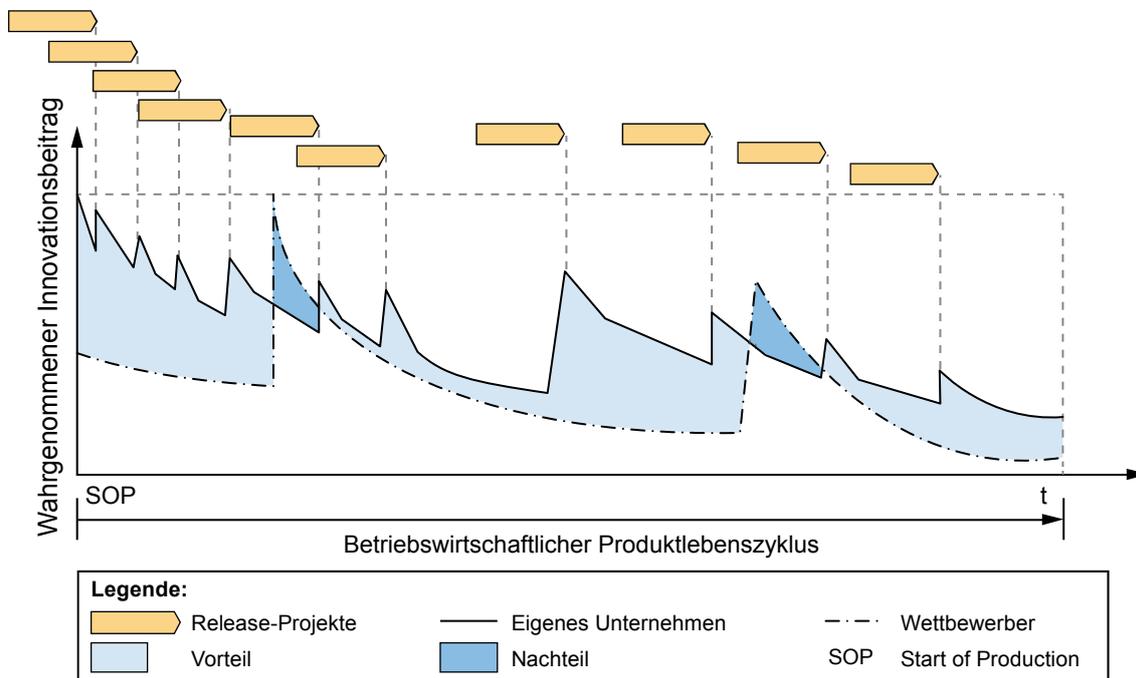


Bild 2-16: Markt- bzw. wettbewerbsstrategischer Vorteil häufiger Release-Projekte nach SCHUH [Sch13, S. 219]

Eine zu häufige Aktualisierung des Produkts durch kundenrelevante Produkt-Features kann allerdings auch zum **Nachteil** werden. Mit jeder Markteinführung entstehen zusätzliche Fixkosten, die insb. bei Produkten mit Zertifizierungsbedarf den Nutzen inkrementeller Verbesserungen übersteigen können. Darüber hinaus führen gerade erklärungsbedürftige Produkt-Features zu einer Überlastung des Vertriebskanals, da das Vertriebspersonal geschult und das Marketingmaterial aktualisiert werden muss. Auch den Kunden kann eine zu hohe Produktrate überfordern oder sogar aufgrund eines zu geringen Deltas zum Vorprodukt verärgern [SR98, S. 76ff.]. Hier gilt es einen vernünftigen Kompromiss zu finden, der unter Berücksichtigung von markt- und wettbewerbsstrategischen Faktoren die Rate kundenrelevanter Produkt-Releases definiert.

Interne entwicklungsstrategische Vor- und Nachteile

Die Vorteile aus markt- und wettbewerbsstrategischer Sicht liegen in einer aktiven Steuerung der Innovationsrate. Aus interner entwicklungsstrategischer Sicht ergeben sich die **Vorteile** durch die Änderungskonsolidierung, die Nutzung von Änderungssynergien und die damit verbundene Rationalisierung des Entwicklungs- und Änderungsprozesses. Die Release-Planung reduziert durch die Bündelung von Änderungen zu Releases die Aufwände für Folgeänderungen, notwendige Tests, die Pflege der Dokumentation und die Koordination des Entwicklungsprozesses. Änderungen werden nur in festgelegten Zeiträumen in die Produktion übergeben, so dass über einen längeren Zeitraum hinweg ohne

Störungen produziert werden kann. „Reibungsverluste“ werden vermieden. Zudem ergibt sich durch die kontrollierte Umsetzung eines abgegrenzten Release-Umfangs eine geringere technische Komplexität: Das Risiko im Umsetzungsprozess lässt sich minimieren, die Komplexität des Änderungsprozesses beherrschen [Lin07], [Bel08, S. 49], [Sch05, S. 112ff.]. Ein weiterer entscheidender Vorteil ist die flexible und gleichmäßige Auslastung der Entwicklungsressourcen. Entwicklungsprojekte haben stets das Ziel, möglichst hohe Qualität in kurzer Zeit bei möglichst geringen Entwicklungs- und Produktkosten zu realisieren (Bild 2-17).

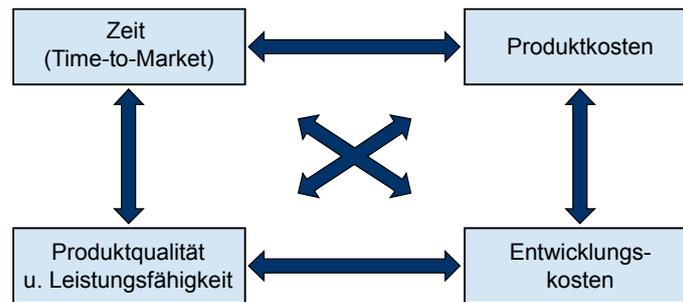


Bild 2-17: Zielkonflikte in der Produktentwicklung [SR98, S. 23ff.]

Diese Ziele stehen jedoch im Konflikt zueinander, so dass die Priorisierung des einen Ziels Auswirkungen auf das Erreichen der anderen Ziele hat. Diese Wechselwirkungen erfordern fortwährend Kompromisse („Trade-Offs“), die sich systematisch durch die Release-Planung erzielen lassen [KBD+14].²⁷ Ein Beispiel ist die Verkürzung des Markteintritts aufgrund der bewussten Verlagerung bestimmter Produkt-Features auf ein späteres Release. Dies folgt dem Grundsatz:

“At delivery time, it is better to be 100% complete with 80% of the most important product features than to be 80% complete with 100% of features and nothing to deliver.” [Fai09]

Einen weiteren Vorteil der Release-Planung sieht SCHUH in der systematischen Pflege bzw. Bereinigung modularer Produktprogramme. Durch Synchronisation und parallele Einführung aktualisierter Module in das Produktprogramm können Innovationen produktprogrammweit ausgerollt, gleichzeitig aber auch obsoletere Produkt- und Modulvarianten eliminiert werden [Sch13, S. 222f.]. Die Ausrichtung an neuen Kundenbedürfnissen kann so ohne eine unverhältnismäßige Zunahme der Produktvarianz erfolgen [SDL+04], [Bel08, S. 49]. Die Release-Planung dient folglich als Werkzeug zur vorsorglichen und direkten Kontrolle bzw. Reduzierung der produkt- und änderungsprozessbedingten Komplexität über den gesamten Produktlebenszyklus hinweg. Damit geht die Reduzierung von Zeit und Aufwand einher, was wiederum die Effizienz bei der Entwicklung und der Produktpflege verbessert [SDL+04, S. 37], [Sch13, S. 212ff.].

²⁷ Eine tabellarische Übersicht über Produktentwicklungsrisiken und damit verbundene Kompensationsmöglichkeiten findet sich in Anhang A1.1.

Aus entwicklungsstrategischer Sicht ergeben sich jedoch auch **Nachteile**. Dem eingesparten Aufwand in Entwicklung und Produktpflege steht ein aufwändigerer Planungsprozess für die Änderungsumsetzung gegenüber. Die Release-Planung erfordert einen hohen Abstimmungsaufwand sowohl innerhalb der Produktentwicklung als auch übergreifend in den Bereichen Produktmanagement, Einkauf, Qualitätssicherung, Arbeitsvorbereitung und Produktion. Darüber hinaus kann es zu Einschränkungen in der Änderungsumsetzung kommen. Statt einer sofortigen Umsetzung werden Änderungen auf das nächste Releases verschoben, wodurch z.B. Maßnahmen zur Kosteneinsparung erst später greifen. Die Nachteile verdeutlichen den Bedarf nach einem pragmatischen und systematischen Ansatz für die Release-Planung, der insb. den bereichsübergreifenden Abstimmungsaufwand reduziert. Tabelle 2-1 stellt die Vor- und Nachteile der systematischen Release-Planung zusammenfassend dar. Unabhängig davon bleibt festzuhalten, dass es gerade bei änderungsintensiven Systemen keine Alternative zu einer systematischen Release-Planung gibt. Das Ergebnis einer unsystematischen Umsetzung von Änderungen wäre, dass die Anzahl der Varianten explodiert und die resultierenden Komplexitätskosten die Rentabilität des Geschäfts zunichtemachen.

Tabelle 2-1: Vor- und Nachteile einer systematischen Release-Planung

	Vorteile	Nachteile
Markt- und wettbewerbsstrategische Sicht	<ul style="list-style-type: none"> • Beschleunigter Markteintritt • Frühes Kundenfeedback • Aktuelle Produkte • Flexible Reaktion auf veränderte Markt- und Wettbewerbsbedingungen • Imagegewinn durch stetige Produktverbesserung 	<ul style="list-style-type: none"> • Zusätzliche Fixkosten je Markteinführung • Überlastung der Vertriebskanäle • Überforderung und Verärgerung des Kunden • Langweilige Produkte
Interne entwicklungsstrategische Sicht	<ul style="list-style-type: none"> • Systematische und synergetische Umsetzung von technischen Änderungen • Weniger Änderungen im Produktionsprozess • Geringere technische Komplexität durch abgegrenzten Release-Umfang • Flexiblere und gleichmäßige Auslastung der Entwicklungsressourcen • Systematische Produktprogrammpflege 	<ul style="list-style-type: none"> • Aufwändiger Planungsprozess für die Änderungsumsetzung • Hohe bereichsübergreifende Abstimmungsaufwände • Weniger Freiheit bei der Änderungsumsetzung • Verspätete Umsetzung von Änderungen

2.4.4 Voraussetzungen für die Entwicklung in Releases

Zur Erschließung der mit der Entwicklung in Releases verbundenen Nutzenpotentiale sind im Unternehmen sowohl **organisatorisch** als auch **produktseitig Voraussetzungen** zu schaffen. Sind diese Voraussetzungen nicht gegeben, ist es wahrscheinlich, dass die zuvor beschriebenen Nachteile der Release-Planung überwiegen.

Strategische Produktplanung

Erfolgsentscheidend für die Release-Planung ist die enge Verzahnung mit unternehmens- und produktstrategischen Überlegungen [SR98, S. 82], [Sch13, S. 223f]. Grundlage hierfür ist die Etablierung der strategischen Produktplanung, die durch systematische Markt- und Technologiebeobachtung zukünftige Erfolgspotentiale antizipiert und so stetig Ideen und Konzepte für neue bzw. verbesserte Produkte hervorbringt. Wie in Abschnitt 2.2.2 herausgestellt, verspricht insb. der Wandel von mechatronischen zu intelligenten technischen Systemen enorme Potentiale. Diese gilt es, durch geeignete Methoden systematisch zu identifizieren und zu erschließen. Ein Beispiel für eine derartige Methode liefern ANACKER ET AL. [ADG+14]. Die Notwendigkeit der Weiterentwicklung des Produkts resultiert jedoch nicht nur aus technologischem Fortschritt: Auch veränderte Kundenbedürfnisse, Marktentwicklungen, Gesetzesänderungen oder Kundenbeschwerden sind zu berücksichtigen. Hier gibt die strategische Produktplanung nicht nur vor, welche Produkt-Features und Änderungen grundsätzlich umzusetzen sind, sondern priorisiert deren Umsetzung ausgehend von Markt- und Wettbewerbsbeobachtungen. Darüber hinaus stellt die strategische Produktplanung sicher, dass die Weiterentwicklung des technischen Systems bei Bedarf mit einer Anpassung des dazugehörigen Geschäftsmodells einhergeht [Pei15]. Grundlegende Voraussetzung ist, dass all diese Informationen in eine fundierte Produktstrategie einfließen, die Aussagen und Vorgaben über zukünftige Produkt- und Technologieentwicklungen trifft [GP14, S. 25f.], [GE13, S. 187]. Die Release-Planung ist auf diese Produktstrategie abzustimmen.

Abstimmung mit Marketingstrategie

Entsprechend der Produktstrategie sind auch Release-Planung und Marketingstrategie aufeinander abzustimmen. Durch die gezielte Vermarktung kundenrelevanter Produkt-Releases muss die damit einhergehende Produktwertsteigerung²⁸ dem Kunden vermittelt werden. So bietet eine geschickte Vermarktung das Potential, eine aus entwicklungstechnischer Sicht eher inkrementelle Verbesserung zu einem Markterfolg zu machen [Ebe06, S. 849], [Sch13, S. 212f.].

Entkoppelte Technologieentwicklung

Die Realisierung neuer Produkt-Features erfordert häufig die (Weiter-)Entwicklung bestehender oder neuer Technologien. Die Komplexität und der Neuheitsgrad dieser

²⁸ Entsprechend der VDI-Richtlinie 2800 ist der Produktwert das Verhältnis aus Bedürfnisbefriedigung und erforderlichen Aufwand. Aus Sicht des Kunden ergibt sich somit eine Produktwertsteigerung entweder durch eine verbesserte Bedürfnisbefriedigung oder durch einen reduzierten Preis [VDI2800].

Technologien resultiert in Risiken, die den Projekterfolg entscheidend beeinflussen und gefährden können. Ein Ansatz zur Reduktion dieser Risiken ist die Entkopplung von Technologie- und Systementwicklung [SR98, S. 79f], [SCF+00], [Lev06, S. 102ff]. Die mit einer Technologie verbundenen Unsicherheiten werden in individuelle Technologieentwicklungsprojekte verlagert, für die andere Rahmenbedingungen im Hinblick auf Zeit und Kosten gelten. Hat die Technologie einen entsprechenden Reifegrad erreicht, fließt sie in die Systementwicklung ein und kann in einem der nächsten Produkt-Releases realisiert werden. Bestehende Ansätze aus dem Technologiemanagement wie das Technologie-Roadmapping [SBK08, S. 3ff.], [GP14, S. 166f.], [LA06, S. 6] und Technologiereifegrade (engl. Technology Readiness Level) [Krö07, S. 100], [SRH+08] unterstützen die systematische Synchronisation von Produkt- und Technologieentwicklung. Diese Ansätze gilt es, für eine erfolgreiche Release-Planung zu etablieren.

Änderungsfreundliche Systemarchitektur

Eine weitere Voraussetzung für die Weiterentwicklung im Sinne der Release-Planung ist eine änderungsfreundliche Systemarchitektur [SR98, S. 83], [Sch13, S. 224]. Zur Gestaltung änderungsfreundlicher Systemarchitekturen sind verschiedene Gestaltungsprinzipien (z.B. Modularität, Einfachheit) einzuhalten [FS05], auf die in Abschnitt 2.5.3 näher eingegangen wird. Die Einhaltung dieser Gestaltungsprinzipien ermöglicht die Bündelung und weitestgehend unabhängige Weiterentwicklung von gemeinsam, innerhalb eines Releases zu veröffentlichenden Änderungen. Im Falle einer modularen Systemarchitektur beschränkt sich die Realisierung von Änderungen bspw. auf wenige Module mit leicht anpassbaren Schnittstellen und gleichzeitig geringeren Änderungsauswirkungen. Ein weiteres bewährtes Mittel ist die Überdimensionierung bzw. robuste Gestaltung von Modulen und Schnittstellen. Diese bildet die Basis für eine leichte Erweiterbarkeit des Produkts, wenngleich sie mit der Zunahme der Produktkosten verbunden ist. Hier gilt es, einen geeigneten Kompromiss zu finden [FS05, S. 345ff.].

2.4.5 Fazit und Einordnung der zu entwickelnden Systematik

Die Release-Planung und damit die zu entwickelnde Systematik bewegen sich im Spannungsfeld zwischen strategischer Produktplanung (Produktstrategie) und deren Operationalisierung in Form konkreter Entwicklungsprojekte (Projektplan). Ein systematischer Release-Planungsprozess verspricht dabei sowohl markt- und wettbewerbsstrategische als auch entwicklungsstrategische Nutzenpotentiale, deren Erschließung an Voraussetzungen geknüpft ist. Eine Voraussetzung für die Release-Planung ist ein kontinuierlicher Strom von neuen Produkt-Features und technischen Änderungen. Deren Identifikation und die Gewinnung relevanter Informationen über die Entwicklung von Märkten und Technologien ist nicht Gegenstand dieser Arbeit. Hier sei auf etablierte Methoden der strategischen Produktplanung verwiesen (vgl. auch Abschnitt 3.4). Auch die Marketingstrategie, die Technologieplanung und -entwicklung sowie die Gestaltung einer änderungsfreundlichen Systemarchitektur sind nicht Gegenstand dieser Arbeit.

2.5 Interdisziplinäre Systementwicklung

Für die systematische Umsetzungsplanung von Änderungen sind neben markt- und wettbewerbsstrategischen Überlegungen insb. ein umfassendes Systemverständnis sowie das Wissen über die Auswirkungen einzelner Änderungen auf das bestehende System erforderlich [SAA13]. Hierdurch weist die Release-Planung eine Schnittstelle zur interdisziplinären Systementwicklung auf, die bereits in Abschnitt 2.3 durch die Einordnung in den Produktentstehungsprozess aufgezeigt wurde. Im Folgenden wird diese Schnittstelle konkretisiert. Dazu wird in **Abschnitt 2.5.1** der Ansatz des Systems Engineerings einschließlich des Model-Based Systems Engineerings erläutert. Beide zeichnen sich als vielversprechende Ansätze zur Beherrschung der Komplexität in der Planung und Entwicklung komplexer technischer Systeme ab [GDS+13, S. 48ff.]. Die Umsetzung einzelner Releases erfolgt in Projekten, die sich an typischen Vorgehensmodellen der Systementwicklung orientieren. Als etabliertes Vorgehensmodell für die Entwicklung mechatronischer Systeme wird daher in **Abschnitt 2.5.2** die VDI-Richtlinie 2206 vorgestellt. Davon ausgehend wird in **Abschnitt 2.5.3** die Gestaltung änderungsfreundlicher Systemarchitekturen als Voraussetzung für erfolgreiche Release-Projekte konkretisiert und in den Kontext der Systementwicklung eingeordnet. Abschließend wird in **Abschnitt 2.5.4** die Schnittstelle der Release-Planung zum Änderungsmanagement und zum Umgang mit Änderungen in der Systementwicklung aufgezeigt.

2.5.1 Grundlagen des Systems Engineering

Der Ansatz des **Systems Engineerings** (SE) zeichnet sich als geeigneter Lösungsansatz zur Beherrschung der zunehmenden Komplexität moderner technischer Systeme ab. Unlängst dringt der Ansatz aus den klassischen Anwendungsfeldern der Luft- und Raumfahrt in andere Branchen wie die Automobilindustrie vor [GDS+13, S. 48ff.]. Es handelt sich um einen durchgängigen fachdisziplinübergreifenden Ansatz, der das zu entwickelnde System einschließlich des dazugehörigen Projekts adressiert [GDS+13, S. 20f.]. In der Literatur existiert eine Vielzahl verschiedener Definitionen, Ansätze und Vorgehensmodelle des Systems Engineerings [SR09], [KSS+11], [HWF+12]. Darüber hinaus widmen sich unterschiedliche Organisationen²⁹ durch die Definition eigener Standards, Normen und Richtlinien dem Thema. Die INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING (INCOSE) definiert Systems Engineering wie folgt:

„Systems Engineering ist ein interdisziplinärer Ansatz und soll die Entwicklung von Systemen methodisch ermöglichen. SE fokussiert ein ganzheitliches und zusammenwirkendes Verständnis der Stakeholder-Anforderungen, der Entdeckung von Lösungsmöglichkeiten und der Dokumentation von Anforderungen sowie das Synthetisieren, Verifi-

²⁹ Beispiele sind das „Systems Engineering Handbook“ der NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA) [NASA/SP-6105] und der „MIL-STD-498“ [MIL-STD-498] des DEPARTMENT OF DEFENSE (DoD). Einen Überblick über die Vielfalt der Standards und Normen gibt [Arm05].

zieren, Validieren und die Entwicklung von Lösungen. Das gesamte Problem wird während der Konzeptentwicklung bis zur Systementwicklung betrachtet. Das Systems Engineering stellt hierfür geeignete Methoden, Prozesse und Best Practices bereit.“ [INC12, S. 7].

Nach HABERFELLNER ET AL. sind für die Erarbeitung guter Lösungen unterschiedliche Faktoren wie Fachwissen, Situationskenntnis und Erfahrung relevant. Das Systems Engineering ergänzt diese um eine methodische Komponente und unterstützt die sinnvolle Abstimmung der Faktoren. Das Konzept des Systems Engineerings lässt sich in zwei übergeordnete Bereiche strukturieren (Bild 2-18): Die SE-Philosophie und den Problemlösungsprozess [HWF+12, S.28 ff.].

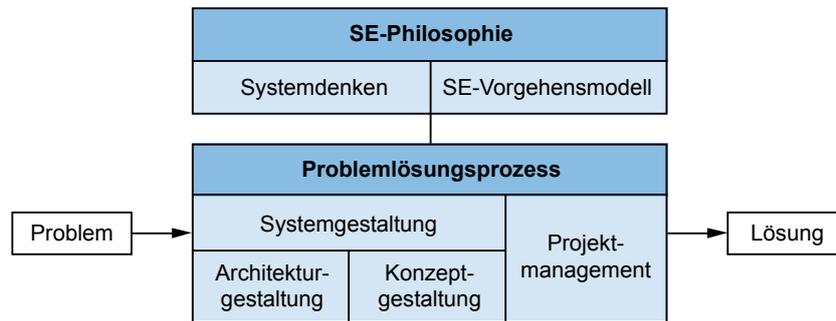


Bild 2-18: Konzept des Systems Engineering [vgl. HWF+12, S.28]

Die **SE-Philosophie** beschreibt das gedankliche Gerüst des Systems Engineerings und umfasst das Systemdenken und das SE-Vorgehensmodell. Das **Systemdenken** hilft komplexe Zusammenhänge und Systeme zu verstehen, zu strukturieren und letztlich zu gestalten. Das System wird dazu aus verschiedenen Blickwinkeln untersucht. Hierzu zählen die umfeld-, wirkungs- und strukturorientierte Betrachtungsweise. Je nach Problemstellung und Situation werden dazu unterschiedliche Abstraktionsniveaus gewählt, die von der Zweckmäßigkeit und der Problemrelevanz abhängen [HWF+12, S. 33ff.]. Das **SE-Vorgehensmodell** liefert Empfehlungen und Richtlinien zur Strukturierung der Lösungsfindung in beherrschbare Teilprozesse [HWF+12, S. 57ff.]. Neben dem SE-Vorgehensmodell existieren eine Reihe alternativer Vorgehensmodelle zur Systementwicklung, von denen in Kapitel 2.5.2 die VDI-Richtlinie 2206 als Quasi-Standard zur Entwicklung mechatronischer Systeme konkretisiert wird.

Der **Problemlösungsprozess** ist der zweite Bestandteil des SE-Konzepts und untergliedert sich in die Systemgestaltung und das Projektmanagement. Während das **Projektmanagement**³⁰ die erfolgreiche Abwicklung von Projekten aus organisatorischer Sicht sicherstellt, stehen bei der **Systemgestaltung** das Problem selbst sowie die Lösungsfindung im Vordergrund. Hierbei werden die Aufgabenbereiche Architektur- und Konzeptgestaltung unterschieden [HWF+12, S. 131ff.]. Im Rahmen der **Architekturgestaltung** wird

³⁰ Einen umfassenden Überblick über die Techniken des Projektmanagements gibt das PROJECT MANAGEMENT INSTITUTE mit dem „Project Management Body of Knowledge“ [ANSI/PMI99-001-2004].

die grundlegende Architektur eines Systems im Sinne eines vorteilhaften Lösungsprinzips festgelegt [HWF+12, S. 183ff.]. Die Architekturgestaltung ist von zentraler Bedeutung im Kontext der Release-Planung und wird daher in Kapitel 2.5.3 detailliert beschrieben. Die **Konzeptgestaltung** baut auf der Architektur auf und konkretisiert Subsysteme und Komponenten. Sowohl die Architektur- als auch die Konzeptgestaltung folgen dabei der Logik des Problemlösungszyklus in drei Phasen: Zielsuche, Lösungssuche und Auswahl [HWF+12, S. 195ff.]. Im Model-Based Systems Engineering wird diese Logik durch abstrakte modellhafte Abbildungen unterstützt.

Model-Based Systems Engineering

Der Ansatz des **Model-Based Systems Engineerings** (MBSE) bezeichnet das Paradigma einer fachdisziplinübergreifenden Entwicklung und Analyse von Systemen auf Basis von Modellen. Im Fokus steht der konsequente Einsatz eines Systemmodells, das durch eine ganzheitliche und allgemein verständliche Systembeschreibung als Verständigungsmittel zwischen den an der Entwicklung beteiligten Fachdisziplinen dient. Zunehmend wird das Systemmodell auch als Hilfsmittel zur Abstimmung mit dem Management, dem Vertrieb und dem Kunden genutzt [GDS+13, S. 36]. MBSE zielt auf den Wandel von einem dokumentenzentrierten zu einem modellbasierten Vorgehen ab [Kai13, S. 24ff.]. Dieses modellbasierte Vorgehen zeichnet sich durch zwei Charakteristika aus, die wesentlich zur Steigerung der Transparenz und damit zu einem verbesserten Systemverständnis in der Entwicklung beitragen: Die Dokumentation von Informationen durch Diagramme und die Verknüpfung dieser Informationen durch sogenannte Tracelinks³¹. Bild 2-19 veranschaulicht diese zwei Charakteristika.

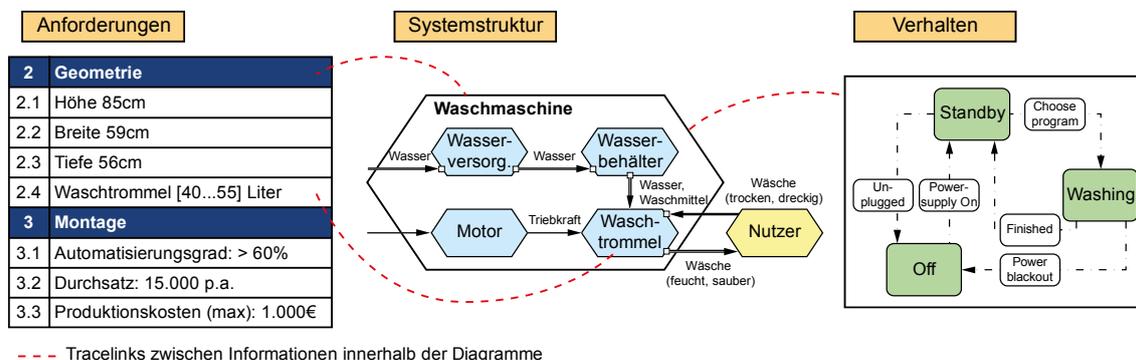


Bild 2-19: Diagramme und Tracelinks als Charakteristika des MBSE

Die Beschreibung aller wesentlichen fachdisziplinübergreifenden Informationen mit Hilfe von **Diagrammen** trägt zum einheitlichen Verständnis bei und dient als Kommunikationsmedium. Methoden³² (z.B. SYSMOD [Wei06], CONSENS [GLL12]) definieren

³¹ Tracelinks bezeichnen die Verknüpfung zweier Elemente (z.B. Anforderungen, Systemelemente, Funktionen) in der Modellierung. Sie dienen der Rückverfolgbarkeit von Informationen [PDK+11].

³² Einen umfassenden Überblick über Methoden, Sprachen und Werkzeuge des MBSE gibt [Est08]. Darüber hinaus werden die für diese Arbeit relevanten Modellierungstechniken technischer Systeme im Stand der Technik genauer untersucht (vgl. Abschnitt 3.1).

in Kombination mit einer Modellierungssprache (z.B. SysML [Wei06]), welche Aspekte eines Systems in welcher Art und Reihenfolge durch Diagramme beschrieben werden. **Tracelinks** hingegen verknüpfen die in den Diagrammen enthaltenen Informationen und gewährleisten die Rückverfolgbarkeit von Systemzusammenhängen. Grundsätzlich hilft diese Rückverfolgbarkeit dabei, Entscheidungen fundiert zu treffen und später nachvollziehen zu können. Ergänzend zu den Diagrammen fördern die Tracelinks somit das fachdisziplinübergreifende Systemverständnis [Bei13, S. 37]. Folgende Anwendungsfälle verdeutlichen dies: Die Verknüpfung von Anforderungen und Systemelementen erlaubt z.B. die verbesserte Analyse von Anforderungsänderungen und deren Auswirkungen auf das System [Alt12, S. 125ff]. Auch unterstützt die Verknüpfung die Wiederverwendung bestehender Lösungen und damit die Evolution des Produkts [FMS12, S. 20ff.], [Alt12, S. 136ff]. Beide Aspekte sind von hoher Relevanz für fundierte und nachvollziehbare Entscheidungen in der Release-Planung, so dass sich auch hier das Systemmodell zur Unterstützung des Entscheidungsprozesses anbietet.

Aspekte des Systemmodells

Das **Systemmodell** muss alle relevanten Informationen beinhalten, die für ein einheitliches Verständnis der beteiligten Fachdisziplinen erforderlich sind. Wesentlich sind hierzu die Aspekte Anforderungen, Systemarchitektur und Verhalten [Alt12, S. 8ff]. **Anforderungen** sind die „Messlatte“ für den Erfolg des zu entwickelnden Produkts [GLL12]. Sie werden häufig textbasiert in Anforderungslisten festgehalten, die jedoch zunehmend um weitere Ansätze zur Anforderungsspezifikation (z.B. Use Cases, Feature-Modelle) ergänzt werden. Das Requirements Engineering (RE) widmet sich der Ermittlung, Spezifikation, Analyse, Vereinbarung, Validierung und Verwaltung von Anforderungen [Ebe10, S. 32], [HWF+07], [Rup09, S. 14]. Neben den Anforderungen umfasst das Systemmodell die **Systemarchitektur**, deren Elemente im Sinne des MBSE über Tracelinks mit den Anforderungen verknüpft sind. Die Systemarchitektur ist immer im Zusammenhang mit den Anforderungen zu sehen und iterativ zu erarbeiten [Alt12, S. 11ff.]. Ein weiterer Aspekt des Systemmodells ist das **Verhalten**. Je nach Entwicklungsaufgabe sind unterschiedliche Arten des Systemverhaltens zu modellieren (z.B. Kinematik, Zustände). Dafür existieren verschiedene Konzepte und Technologien, die in der Regel auf einen spezifischen Anwendungsbereich spezialisiert sind [Alt12, S. 14f]. Beispiele sind MATLAB/Simulink [The11-ol] und Modelica [GTS14].

Anforderungen und Systemarchitektur sind immer der Startpunkt für die Erarbeitung des Systemmodells. Die Modellierung des Verhaltens folgt wahlweise in einem zweiten Schritt, um z.B. durch simulierbare Modelle Fehler im Systemverhalten frühzeitig zu identifizieren [Alt, S. 15]. Das initiale Systemmodell wird im Zusammenspiel aller beteiligten Disziplinen erarbeitet und kann unabhängig vom weiteren Vorgehen zur frühzeitigen Systemspezifikation im Sinne eines Lastenhefts eingesetzt werden. Das Vorgehen in der Systementwicklung definieren Vorgehensmodelle, die im Folgenden beschrieben werden.

2.5.2 Vorgehensmodelle der Systementwicklung

Die Entwicklung technischer Systeme sowie die Umsetzung einzelner Releases erfolgt in **Entwicklungsprojekten**, die üblicherweise in Projektphasen mit definierten Zielen und Meilensteinen unterteilt werden. Diese Projektphasen werden auch als Projektlebenszyklus bezeichnet und zielen durch die Einteilung in Arbeitspakete mit klar definierten Zielen auf ein verbessertes Management von Entwicklungsprojekten ab. Die Untergliederung in einzelne logisch und zeitlich voneinander getrennte Phasen geben Vorgehensmodelle vor [FMC05, S. 84ff.], [ANSI/PMI99-001-2004].

Gemeinhin decken **Vorgehensmodelle für die Systementwicklung** die notwendigen Phasen eines Entwicklungsprojekts von den Kundenanforderungen bis zum marktfähigen Produkt ab [FMC05, S. 84ff.]. Die Phasen unterscheiden sich jedoch je nach Produktart und Branche. Auch die Eigenschaften des Entwicklungsprojekts (z.B. Größe, Dynamik des Umfelds, Personal) beeinflussen die Wahl des richtigen Ansatzes, so dass heute verschiedene Vorgehensmodelle für die Systementwicklung existieren [Lev06, S. 31ff], [HWF+12, S. 109f.]. Beispiele sind das SE-Vorgehensmodell nach HABERFELLNER ET AL. [HWF+12, S. 65ff.] sowie das Wasserfallmodell [Som07, S. 96ff] und das Spiralmodell [Boe88], die beide ihren Ursprung in der Softwaretechnik haben.

Für die Entwicklung mechatronischer Systeme liefert die **VDI-Richtlinie 2206** ein etabliertes Vorgehen, das den minimalen Konsens der Fachwelt³³ zusammenfasst [Ehr09, S. 270f.]. Die Richtlinie fokussiert die methodische Unterstützung der fachdisziplinübergreifenden Entwicklung mechatronischer Systeme, um der Herausforderung einer steigenden Interdisziplinarität dieser Systeme zu begegnen [VDI2206, S. 8]. Sie versteht sich als Ergänzung zu etablierten Ansätzen der Fachdisziplinen (z.B. PAHL/BEITZ im Maschinenbau [PBF+07], Y-Diagramm in der Schaltungsentwicklung [GAG+13, S. 3f.]). Das Vorgehen greift das aus der Softwaretechnik bekannte V-Modell auf und wird durch wiederkehrende Prozessbausteine für die Bereiche Systementwurf, Modellbildung und -analyse, domänenspezifischer Entwurf, Systemintegration und Eigenschaftsabsicherung unterstützt [VDI2206, S. 26f.] (Bild 2-20).

Anforderungen resultieren aus einem konkreten Entwicklungsauftrag und sind der Ausgangspunkt für die Entwicklung. Sie präzisieren die Aufgabenstellung und dienen zugleich als Maßstab für die Bewertung des späteren Produkts. Im **Systementwurf** werden ausgehend von den Anforderungen und der funktionalen Beschreibung des Systems die wesentlichen physikalischen und logischen Wirkungsweisen des Systems festgelegt. Ergebnis ist ein fachdisziplinübergreifendes Lösungskonzept, das der Systemarchitektur des zu entwickelnden Systems entspricht.³⁴ Dieses Lösungskonzept wird im

³³ Neben der VDI-Richtlinie 2206 existieren weitere Ansätze, z.B. das 3-Ebenen-Modell nach BENDER [Ben05, S. 45] und die Entwurfsschritte nach ISERMANN [Ise08, S. 36]. Eine Übersicht und Klassifikation der Ansätze gibt MÖHRINGER [Möh04, S. 14ff.].

³⁴ Wird diese Phase im Sinne des MBSE explizit durch Modelle unterstützt, dann liegt mit dem Systementwurf bereits ein initiales Systemmodell vor.

domänenspezifischen³⁵ Entwurf unter Anwendung fachdisziplinspezifischer Entwicklungsmethoden konkretisiert. Die in den einzelnen Fachdisziplinen erarbeiteten Ergebnisse werden im Rahmen der **Systemintegration** zu einer Gesamtlösung zusammengeführt und kontinuierlich anhand der Anforderungen und dem zuvor erarbeiteten Lösungskonzept überprüft. Dies entspricht der **Eigenschaftsabsicherung**, die durch eine rechnergestützte **Modellbildung und -analyse** unterstützt wird. Im Ergebnis liegt das **Produkt** vor, das durch mehrmaliges Durchlaufen des V-Modells zunehmend konkretisiert wird und somit verschiedene Reifegrade durchläuft (z.B. Labormuster, Funktionsmuster, Vorserienprodukt, Serienprodukt) [VDI2206, S. 29f.].

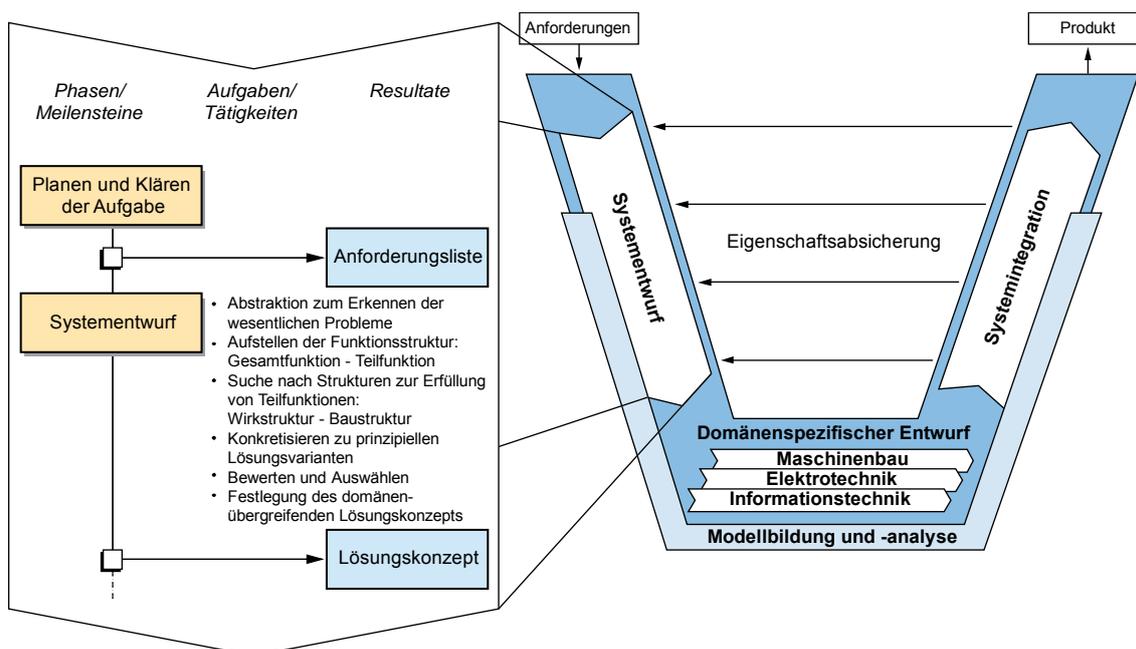


Bild 2-20: V-Modell und der Prozessbaustein Systementwurf [VDI2206, S. 32]

Insbesondere bei der **Weiterentwicklung** bestehender Systeme im Rahmen von Release-Projekten sind nicht alle Arbeitsschritte gleichermaßen zu durchlaufen. Statt eines vollständigen Systementwurfs sind bspw. nur die Auswirkungen der mit dem Release verbundenen Änderungen auf das System zu untersuchen. Die beschriebenen Arbeitsschritte werden dazu je nach Aufgabenstellung sowie Art und Komplexität des Entwicklungsauftrags ausgeprägt und auf die Bedürfnisse des Projekts angepasst [VDI2206, S. 31]. Das Vorgehensmodell nach der VDI-Richtlinie 2206 bietet somit die Möglichkeit einer inkrementellen Entwicklung und damit hohe Flexibilität in der Projektabwicklung. Bei der **inkrementellen Entwicklung** handelt es sich um eine **Entwicklungstaktik**, die auf verschiedene Vorgehensmodelle (z.B. V-Modell, Wasserfallmodell) angewendet werden kann [FMC05, S. 116ff.]. Die Kernidee der inkrementellen Entwicklung liegt in der schrittweisen Realisierung eines Systems durch einzelne Releases. Jedes Release verbessert dabei das bestehende System z.B. durch die Modifikation der Produktstruktur und -

35 Die Begriffe Fachdisziplin und Domäne werden in dieser Arbeit synonym verwendet.

gestalt sowie durch das Hinzufügen neuer Funktionalitäten bzw. Features [Smi07, S. 209ff.], [Coc08]. Ziel dieser Modifikationen sind bspw. die Steigerung der Zuverlässigkeit, Kostenreduzierungen oder auch Baugruppenvereinfachungen. Für die Anwendung der inkrementellen Entwicklung existieren zwei Voraussetzungen: 1) eine änderungsfreundliche Gestaltung der Systemarchitektur, die es ermöglicht, Inkremente in Form weiterer Releases hinzuzufügen sowie 2) eine systematische Release-Planung, die unter Berücksichtigung aller technischen und organisatorischen Restriktionen den Umfang und die Inhalte des nächsten Releases festlegt.

2.5.3 Gestaltung änderungsfreundlicher Systemarchitekturen

Die Gestaltung der **Systemarchitektur** erfolgt als erster Schritt im Prozess der Systementwicklung. Dazu werden die grundlegenden funktionalen und strukturellen Zusammenhänge eines Systems/Produkts im Sinne eines bestimmten Lösungsprinzips festgelegt. Wesentliche Elemente einer Systemarchitektur sind in Bild 2-21 dargestellt.

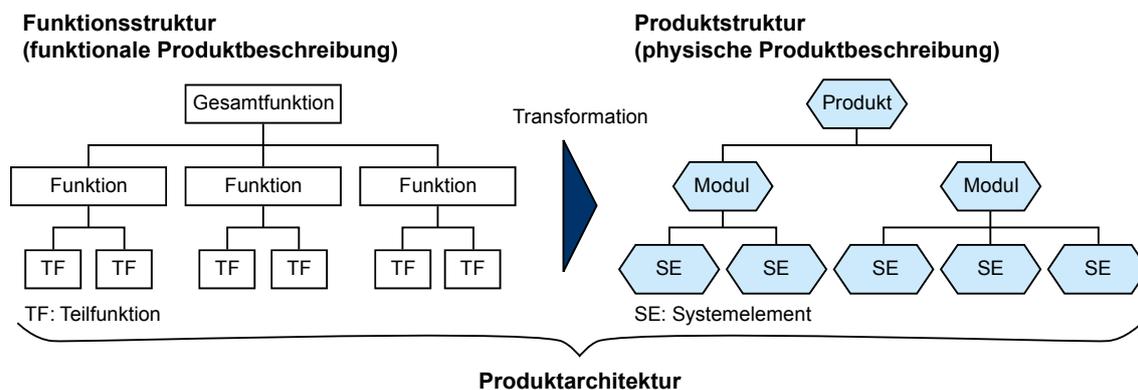


Bild 2-21: Elemente der Systemarchitektur nach ULRICH [Ulr95, S. 2]

In Anlehnung an ULRICH beschreibt eine Systemarchitektur die Struktur eines Systems in Form von *Funktionen, Systemelementen und deren Beziehungen zueinander* sowie auch die *Zusammenhänge zwischen Funktionen und Systemelementen* [Ulr95, S. 2], [Sch12, S. 117]. Die **Architekturgestaltung** (engl. system architecting) ist synonym zur Produktstrukturierung. Sie umfasst die Aufgaben der Allokation von Funktionen auf Systemelemente, die Anordnung dieser Elemente in einer Systemstruktur sowie die Definition der Schnittstellen zwischen diesen Elementen und zur Systemumwelt. Die Gestaltung der Systemarchitektur beeinflusst Eigenschaften wie die Leistungsfähigkeit des Produkts, Möglichkeiten der Variantenbildung und Bauteilstandardisierung, die ideale Entwicklungsorganisation sowie die Änderungsfreundlichkeit [HWF+12, S. 185]. Somit ermöglicht erst das Verständnis der funktionalen und strukturellen Zusammenhänge des Systems die effektive und effiziente Steuerung von komplexen Entwicklungsprojekten.

Der evolutionäre Charakter intelligenter technischer Systeme stellt besondere Anforderungen an die Systemarchitektur: Sie muss zeit- und kosteneffizient verändert, weiterentwickelt und erweitert werden können. Die Fachliteratur diskutiert diese Fähigkeit unter

dem Sammelbegriff „Changeability“³⁶ bzw. **Änderungsfreundlichkeit** [FS05], [RRH08]. Der Begriff fasst verschiedene Systemeigenschaften zusammen, die die Fähigkeit zur Anpassung an Veränderungen beeinflussen. Nach FRICKE und SCHULZ handelt es sich dabei um Flexibilität, Agilität, Robustheit und Anpassungsfähigkeit eines Systems [FS05]. Für die Beeinflussung dieser Eigenschaften ist die Gestaltung der Systemarchitektur von entscheidender Bedeutung [Ulr95], [CWE+04]. Dazu sind Gestaltungsprinzipien einzuhalten, deren Zusammenhänge mit den oben genannten Systemeigenschaften in Bild 2-22 dargestellt sind.

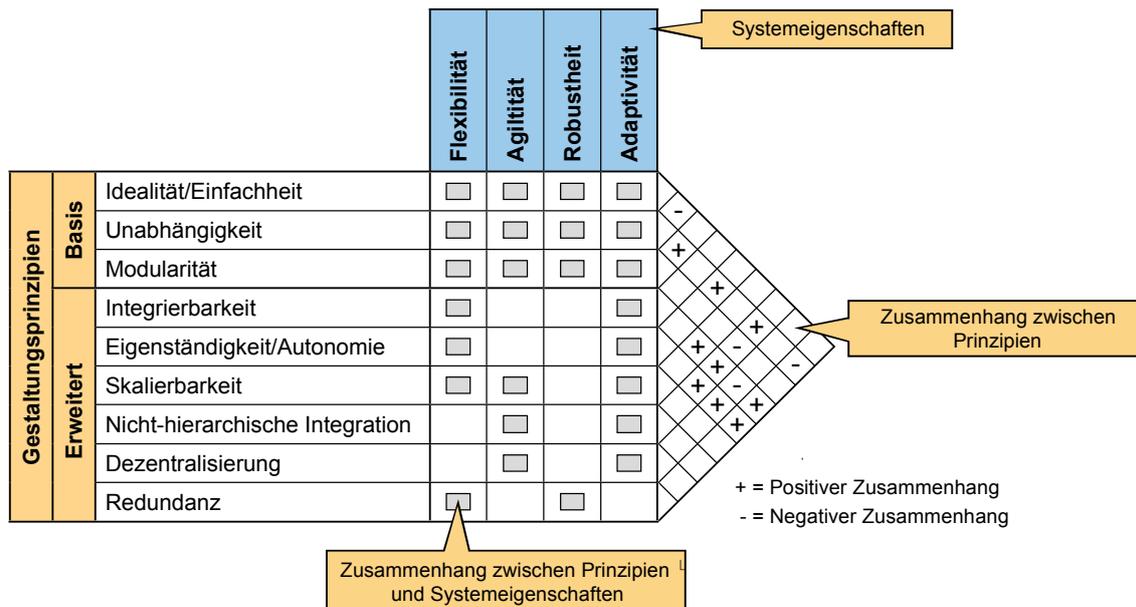


Bild 2-22: Gestaltungsprinzipien für änderungsfreundliche Systemarchitekturen [FS05, S. 348]

Die **Gestaltungsprinzipien**³⁷ erlauben die Realisierung flexibler, agiler, robuster und anpassungsfähiger Systeme. Dabei wird in zwei Kategorien unterschieden: Basisprinzipien, die alle vier Systemeigenschaften beeinflussen, und erweiterte Prinzipien, die jeweils nur ausgewählte Eigenschaften unterstützen. Die verschiedenen Prinzipien beeinflussen sich in ihrer Wirkung gegenseitig, so dass die Einhaltung zweier Prinzipien auch kontraproduktiv wirken kann (z.B. Prinzip Unabhängigkeit und Redundanz). Der Grad, in dem die einzelnen Prinzipien in einer Systemarchitektur berücksichtigt sind, kann mittels verschiedener Metriken gemessen werden [FS05, S. 347ff.].

In Literatur und Praxis kommt dem Gestaltungsprinzip der **Modularisierung** eine besondere Bedeutung zu. Die Modularisierung ist eine Art der Produktstrukturierung, die die Abhängigkeiten zwischen Systemelementen bzw. Modulen verringert bzw. die Schnittstellenvarianz reduziert [Rap99, S. 59f.], [Sch05, S. 130]. Wesentlicher Vorteil³⁸ ist die

36 Ein in diesem Kontext ähnlich verwendeter Begriff ist „Evolvability“ [Ste98], [BB08].

37 Die einzelnen Gestaltungsprinzipien werden im Anhang A1.3 erläutert.

38 Anhang A1.4 gibt einen Überblick über Vor- und Nachteile modularer Systemarchitekturen.

Reduzierung der Änderungsauswirkungen eines Moduls auf andere Module und damit die Entkopplung der Modullebenszyklen. Dies eröffnet über den gesamten Produktlebenszyklus vielfältige Potentiale: eine verbesserte Wiederverwendung in der Entwicklung, die Parallelisierung von Entwicklungsprozessen, vereinfachte Funktions- und Qualitätstests und die Konfiguration zu kundenindividuellen Produktvarianten [Ble11, S. 15]. Häufig genannte Beispiele für modulare Produkte sind Autos und Personal Computer (PC) [BC00].

Aufgrund des erheblichen Potentials widmet sich eine Vielzahl von Ansätzen der Modularisierung. Relevante Ansätze stammen bspw. von ERIXON, GÖPFERT, RAPP, SEKOLEC und STEFFEN [Eri98], [Göp98], [Rap99], [Sek05], [Ste07]. Trotz dieser Vielfalt an Ansätzen stellt FIXSON in einer Analyse von 160 Beiträgen aus 36 Journals fest, dass nur wenige Arbeiten das Thema Änderungsfreundlichkeit explizit adressieren [Fix06]. Hierzu gehören Arbeiten von BROWNING, LINDEMANN, MARTIN und SCHUH [EB08], [KSL12], [MI02], [SSA12]. Alle Arbeiten liefern Ansätze für die Identifikation, den Entwurf und die Entwicklung von Modulen mit dem Ziel einer möglichst änderungsfreundlichen Systemarchitektur.

Zusammenfassend lässt sich festhalten: Die Systemarchitektur definiert, wie einfach und effizient ein System geändert werden kann. Im Extremfall reduziert eine modulare Systemarchitektur die Auswirkungen einer Änderung auf ein einzelnes Systemelement bzw. Modul. Im anderen Extrem wirkt sich die Änderung auf alle übrigen Systemelemente aus, so dass ihre Umsetzung mit erheblichem Aufwand verbunden ist [Ulr95]. Eine änderungsfreundliche und damit modulare Systemarchitektur ist daher Voraussetzung für eine systematische und effiziente Weiterentwicklung technischer Systeme. Die Spezifikation der Systemarchitektur liefert den Entwicklern das notwendige Werkzeug, um Änderungsauswirkungen zu verstehen. Sie ist somit Grundlage für die systematische Umsetzung von Änderungen und damit für die Release-Planung.

2.5.4 Der Umgang mit Änderungen in der Systementwicklung

Technische Änderungen sind elementarer Bestandteil der Systementwicklung. Sie dienen wesentlich dazu, ein System evolutionär zu verbessern und an neue bzw. geänderte Kundenanforderungen anzupassen. Die **Änderungsursachen** sind vielfältig und reichen von Qualitätsverbesserungen für den Kunden bis hin zu Maßnahmen zur Kostenreduktion.³⁹ Entsprechend der Ursachen variieren auch die **Quellen** der Änderungen. Neben internen Abteilungen (z.B. Arbeitsvorbereitung, Konstruktion) sind die Kunden, der Wettbewerb oder die Gesetzgebung zu nennen [ECZ04], [Köh09, S. 13].

Unabhängig von Ursache und Quelle kann eine Änderung eine Reihe weiterer Änderungen nach sich ziehen. Dieses Phänomen wird als **Änderungsfortpflanzung** bezeichnet und in diversen Studien genauer untersucht [FGN+00], [CSE01], [ECE05]. Eine

³⁹ Anhang A1.2 liefert eine Übersicht verschiedener Änderungsursachen.

Änderung kann sich vom ursprünglich betroffenen Systemelement auf weitere Systemelemente ausbreiten. Dies gilt nicht nur für das Produkt selbst, sondern auch für andere Produkte; z.B. durch gemeinsame Plattformen, Produktionsprozesse oder Zulieferer [JEC04, S. 110f]. Durch die Änderungsfortpflanzung steigen Kosten und Dauer der Änderungsumsetzung. Verantwortlich dafür ist nicht nur die Änderung an sich, sondern der Aufwand für ihre Eingliederung in das Gesamtsystem. Dies gilt gleichermaßen für software-, elektronisch- und mechanisch-geprägte Änderungen.

Aufgrund der vielfältigen Ursachen für Änderungen und dem Phänomen der Änderungsfortpflanzung stellt das Änderungswesen⁴⁰ einen signifikanten Aufwand in der Systementwicklung dar [Sch13, S. 215]. Dem systematischen Umgang mit Änderungen widmen sich daher verschiedene Ansätze, die sich grob drei Strategien zuordnen lassen: 1) der Vermeidung, 2) der Früherkennung sowie 3) der organisierten Abwicklung von Änderungen [FGN+00], [Köh09, S. 25ff]. Die Gesamtheit dieser Ansätze wird auch als **Änderungsmanagement** bezeichnet [LR98, S. 372].

1) Vermeidung technischer Änderungen

Die Vermeidung von Änderungen zielt auf die Reduzierung fehlerbedingter Änderungen über den Produktlebenszyklus ab. Dies erfolgt bspw. durch ein systematisches Vorgehen in der Produktentwicklung im Sinne des Systems Engineering und des Virtual Prototyping⁴¹. Beides dient der Identifikation und Prävention von Fehlern bereits in der Entwicklung [Köh09, S. 28ff]. Auch die Entkopplung von Technologie- und Produktentwicklung folgt diesem Ziel. Durch den Einsatz ausgereifter und zuvor getesteter Technologien können technologiebedingte Änderungen reduziert werden [FGN+00].

2) Früherkennung technischer Änderungen

Während sich fehlerbedingte Änderungen durch ein systematisches Vorgehen in der Entwicklung vermeiden lassen, sind extern initiierte Änderungen z.B. durch technologischen Fortschritt oder veränderte Markt- und Kundenbedürfnisse kaum zu vermeiden. Hier setzt die Früherkennung technischer Änderungen an. Ziel ist die Antizipation potentieller, zukünftiger Änderungen und die Erarbeitung entsprechender Vorkehrungen. Zur Antizipation potentieller Änderungen kommen Methoden der Vorausschau zum Einsatz. Beispiele sind die Szenario-Technik und Trendanalysen [GP14, S. 41ff.]. Darauf aufbauend können notwendige Vorkehrungen zur vereinfachten Umsetzung der antizipierten Änderungen getroffen werden. Wesentlicher Stellhebel ist die Gestaltung änderungsfreundlicher Systemarchitekturen (vgl. Abschnitt 2.5.3).

⁴⁰ „Das Änderungswesen umfasst die innerbetriebliche Organisation und die zugehörigen Organisationsmittel zur Änderung von Gegenständen, z. B. von Unterlagen und Komponenten.“ [DIN199-4].

⁴¹ Virtual Prototyping meint die rechnerinterne Modellierung und Analyse eines in Entwicklung befindlichen Produkts. Ziel ist es, durch den Verzicht auf reale Prototypen Zeit und Geld zu sparen [SK97], [GP14].

3) Organisierte Abwicklung technischer Änderungen

Neben der Umsetzung vorbeugender Maßnahmen gilt es, die sich über den Produktlebenszyklus zwangsläufig ergebenden Änderungen in einem organisierten Prozess abzuwickeln. FRICKE ET AL. stellen daher die zwei Anforderungen *effektiv* und *effizient* an die Umsetzung technischer Änderungen. Effektiv meint, dass die Notwendigkeit der Umsetzung einer Änderung systematisch abgewogen werden muss; effizient meint, dass die Umsetzung anschließend unter bestmöglichem Einsatz der Ressourcen erfolgen muss [FGN+00, S. 174ff]. Zur Sicherung einer effektiven und effizienten Änderungssetzung existieren heute unterschiedliche Vorschläge für den Änderungsprozess. Beispiele sind die Handlungsempfehlungen des VERBANDS DER AUTOMOBILINDUSTRIE [VDA4965] und die ISO 10007:2003, die den Änderungsprozess als Teil des Konfigurationsmanagements komplexer Produkte beschreibt [ISO10007]. Im deutschsprachigen Raum wird zudem häufig der technische Änderungsprozess nach DIN 199-4 genannt [DIN199-4]. Dieser bezieht den Änderungsprozess jedoch ausschließlich auf die Produktentwicklung und sieht weder eine Lösungssuche noch eine Änderungsbewertung vor. Der Änderungsprozess nach JARRATT ET AL. greift diese Aspekte hingegen explizit auf und konsolidiert verschiedene bestehende Ansätze zu einem generischen Änderungsprozess [JEC04]. Im weiteren Verlauf wird daher stellvertretend der Änderungsprozess nach JARRETT ET AL. vorgestellt (Bild 2-23). Neben diesen etablierten Änderungsprozessen greifen aktuelle Forschungsarbeiten auch spezifischere Herausforderungen auf. So widmet sich bspw. ABRAMOVICI den Auswirkungen auf den Änderungsprozess, die sich durch den Wandel von rein physischen Produkten zu sogenannten „Produkt-Service-Systemen“ ergeben [ABG10], [AA15].

Der **Änderungsprozess** nach JARRETT ET AL. gliedert sich in sechs Schritte und umfasst ausgehend vom 1) Änderungsantrag 2) die Identifikation von Lösungsansätzen, 3) die Auswirkungsanalyse, 4) die Änderungsgenehmigung, 5) die Änderungsumsetzung und abschließend 6) die Überprüfung der Änderung. Entscheidend für einen erfolgreichen Änderungsprozess ist Schritt drei: die Untersuchung der Änderungsauswirkungen. Hier wird die für die Änderungsgenehmigung erforderliche Entscheidungsgrundlage erarbeitet. Dies umfasst insb. die Analyse der technischen Auswirkungen und die Identifikation der mit der Änderungsumsetzung verbundenen Risiken [JEC04], [Bei13, S. 167].

Trotz der hohen Relevanz der **Auswirkungsanalyse** belegen Studien, dass in der industriellen Praxis die Abschätzung von Auswirkungen häufig auf dem Erfahrungswissen einzelner Experten beruht [FGN+00, S. 174], [BFV00], [Bei13, S. 168]. Dies stellt insb. vor dem Hintergrund der zunehmenden Komplexität und Interdisziplinarität intelligenter technischer Systeme eine Schwierigkeit dar. Die Auswirkungen und Abhängigkeiten von Änderungen einschließlich der damit verbundenen Änderungsfortpflanzung lassen sich kaum mehr durch einzelne Entwickler überblicken. Abhilfe kann hier eine disziplinübergreifende und allgemein verständliche Systembeschreibung schaffen, die als Verständigungsmittel zwischen den an der Entwicklung beteiligten Fachdisziplinen den Status Quo des Systems einschließlich der Systemzusammenhänge aufzeigt [ECZ04, S. 20], [Bei13,

S. 163ff.]. Hier setzt das Model-Based Systems Engineering an, das durch Diagramme und Tracelinks Transparenz in der Systementwicklung schafft.

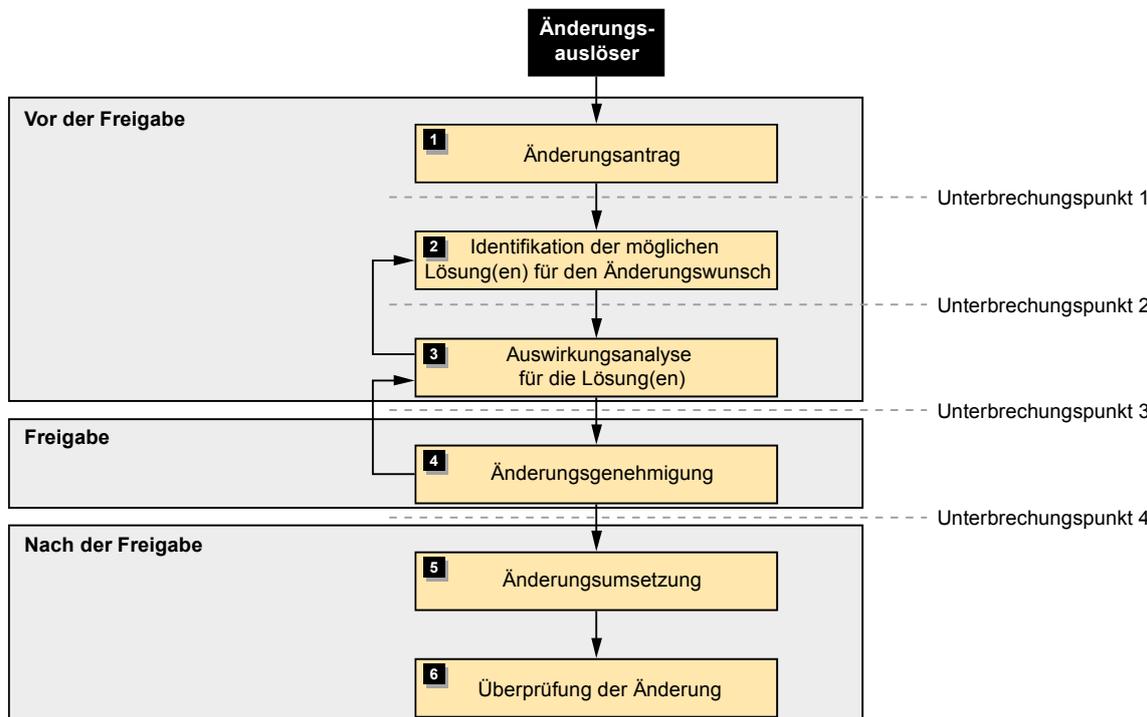


Bild 2-23: Änderungsprozess nach JARRATT ET AL. [JEC04]

Aufbauend auf der Auswirkungsanalyse erfolgt im Rahmen der **Änderungsgenehmigung** die Änderungsbewertung. Diese umfasst eine Kosten-Nutzen-Analyse, auf deren Basis ein Änderungskomitee über die Umsetzung der Änderung entscheidet [JEC04].

Die **Änderungsumsetzung** erfolgt anschließend je nach Wichtigkeit und Dringlichkeit. In Abhängigkeit von ihrer Ursache werden Änderungen sowohl von internen als auch von externen Stakeholdern als unterschiedlich wichtig und dringlich bewertet. Diese Bewertung entscheidet darüber, ob eine Änderung sofort umgesetzt werden muss oder ob sie in einem späteren Abschnitt des Produktlebenszyklus integriert werden kann [JEC04]. Studien belegen, dass lediglich ein kleiner Anteil an Änderungen als zeitkritisch einzuschätzen ist und daher sofort umgesetzt werden muss [DKL06, S. 817], [Sch13, S. 215f.]. Die Umsetzung des wesentlich größeren Teils technischer Änderungen kann daher systematisch geplant werden. Hier weist der Änderungsprozess eine Schnittstelle zur Release-Planung auf, die unter Berücksichtigung verschiedenster Rahmenbedingungen die Integration von Änderungen in verschiedenen Produktversionen und -generationen plant. Ein großer Teil der für die Release-Planung relevanten Informationen über eine Änderung wird jedoch schon im typischen Änderungsprozess erhoben, so dass dieser als Teil der zu entwickelnden Systematik betrachtet werden muss.

2.5.5 Fazit und Einordnung der zu entwickelnden Systematik

Die zu entwickelnde *Systematik zur Release-Planung intelligenter technischer Systeme* soll nahtlos an die etablierten Ansätze und Vorgehensmodelle der interdisziplinären Systementwicklung anknüpfen und sich diese zur Planung und Entscheidungsfindung zu Nutze machen. Dies gilt im Besonderen für das Paradigma des Model-Based Systems Engineerings, das die fachdisziplinübergreifende Entwicklung und Analyse technischer Systeme auf Basis eines Systemmodells propagiert. Das Systemmodell zeichnet sich als geeignetes Kommunikationsmittel in interdisziplinären Teams ab und bietet sich insb. bei komplexen technischen Systemen zur Analyse von Änderungsauswirkungen an. Schnittstellen weist die zu entwickelnde Systematik darüber hinaus zu bestehenden Vorgehensmodellen sowie zum typischen Änderungsprozess in der Systementwicklung auf. Diese Schnittstellen sind gezielt in die *Systematik zur Release-Planung intelligenter technischer Systeme* einzubetten. Die Gestaltung änderungsfreundlicher Systemarchitekturen ist hingegen nicht Gegenstand der Systematik. Gleichwohl ist das Wissen um die Modularisierung der Systemarchitektur von entscheidender Bedeutung für eine systematische Release-Planung.

2.6 Problemabgrenzung

Der Wandel von mechatronischen zu intelligenten technischen Systemen verspricht vielfältige Innovationspotentiale bei gleichzeitig steigender Systemkomplexität. Stetige Produktverbesserungen und die Integration neuer Produkt-Features führen zu technischen Änderungen und somit zu einer Evolution technischer Systeme. Diese Evolution gilt es, durch abgestimmte Innovations- und Anpassungsschritte gezielt zu steuern [GB12, S. 100]. Ein **Lösungsansatz** ist die systematische Weiterentwicklung technischer Systeme in Form von Release-Projekten. Unter Berücksichtigung verschiedener Faktoren, wie dem technologischen Neuheitsgrad, der Bedeutung für den Markterfolg und der Dringlichkeit, ist die Umsetzung der technischen Änderungen in Form von Releases zu planen. Ein Release bezeichnet ein harmonisiertes Bündel von Änderungen und Verbesserungen an einem Produkt, das gemeinsam entwickelt, getestet und freigegeben wird [SDL+04].

Die systematische Release-Planung verspricht vielfältige **Nutzenpotentiale**, die sich zusammenfassend in zwei Sichten unterscheiden lassen (vgl. Abschnitt 2.4.3): Aus markt- und wettbewerbsstrategischer Sicht bietet die Release-Planung insb. das Potential der gesteuerten Umsetzung von Innovationen. Aus entwicklungsstrategischer Sicht liegen die Potentiale in der Änderungskonsolidierung, der Nutzung von Änderungssynergien und der damit verbundenen Rationalisierung des Entwicklungs- und Änderungsprozesses [Sch13, S. 214]. Zur Erschließung dieser Nutzenpotentiale sind **Voraussetzungen** zu erfüllen, die neben der Etablierung der strategischen Produktplanung und einer entkoppel-

ten Technologieentwicklung insb. die Gestaltung einer änderungsfreundlichen Systemarchitektur umfassen (vgl. Abschnitt 2.4.4). Darüber hinaus sind für eine erfolgreiche Release-Planung die folgenden **Herausforderungen** zu meistern:

- **Abgestimmte Produktevolution:** Die Weiterentwicklung des Systems ist auf markt- und wettbewerbsstrategische Überlegungen abzustimmen, die u.a. die Rate und die Zeitpunkte einzelner Produkt-Releases bestimmen. Hierbei sind neben der Verfügbarkeit relevanter Technologien auch die Beschaffenheit der Systemarchitektur und die Komplexität des Systems zu berücksichtigen. Diese Faktoren beeinflussen maßgeblich den Aufwand für die Änderungsumsetzung und die Notwendigkeit der produktübergreifenden Synchronisation von Änderungen. Darüber hinaus schränken die durch die Entwicklungsorganisation vorgehaltenen Ressourcen den möglichen Änderungsumfang und die Häufigkeit ein, mit der neue Releases veröffentlicht werden können. Alle diese Aspekte gilt es, bei der Produktevolution und damit bei der Strukturierung des initialen Release-Plans zu berücksichtigen.
- **Konkurrierende Stakeholder-Präferenzen:** Neben der Strukturierung des Release-Plans ist die Bündelung von Produkt-Features und Änderungen zu Releases wesentliche Aufgabe der Release-Planung. Dabei liegt die Herausforderung in der Berücksichtigung unterschiedlicher und teils konkurrierender Stakeholder-Präferenzen. Eine zeitgleiche Umsetzung aller relevanten Änderungen ist aufgrund begrenzter Ressourcen meist ausgeschlossen. Zudem steht z.B. die von der Produktion angestrebte Kostenminimierung der Forderung des Marketings nach der Kundennutzenmaximierung entgegen. Es sind Kompromisse anzustreben, die systematisch nur durch eine fundierte Planungsunterstützung erzielt werden können.
- **Komplexität technischer Änderungen:** Die zunehmende Komplexität und Interdisziplinarität intelligenter technischer Systeme spiegelt sich auch in Änderungen an diesen Systemen wider. Die potentiellen Auswirkungen einzelner Änderungen einschließlich der damit verbundenen Änderungsförtpflanzungen sind fachdisziplinübergreifend zu überprüfen. Zudem bauen die Produkt-Features intelligenter technischer Systeme zunehmend aufeinander auf (vgl. Abschnitt 2.2.2). Auch diese Abhängigkeiten gilt es, bei der Umsetzungsplanung zu berücksichtigen. Zur Bewältigung dieser Herausforderungen ist ein umfassendes und fachdisziplinübergreifendes Systemverständnis unerlässlich.
- **Gemeinsamer Kooperationskern:** Die Release-Planung bewegt sich im Spannungsfeld von Produktplanung, Systementwicklung und Projektmanagement. Neben den verschiedenen Fachdisziplinen der Systementwicklung sind somit weitere Fachleute mit teils geringerem technischen Verständnis beteiligt. Dies erschwert die ohnehin schon aufwändige Kommunikation in der Planung und Entwicklung des technischen Systems. Erforderlich ist daher ein Kooperationskern, der die disziplinübergreifende Zusammenarbeit fördert und die verschiedenen Sichten auf das System vereint.

Aus den geschilderten Nutzenpotentialen und Herausforderungen ergibt sich der **Bedarf** für eine *Systematik zur Release-Planung intelligenter technischer Systeme*. Die Systematik soll Fachleute, die sich mit der Weiterentwicklung des technischen Systems befassen, bei der Umsetzungsplanung von Änderungen in Form von Releases unterstützen. Aufgrund der Interdisziplinarität der beteiligten Fachleute soll sich die Systematik an etablierten Standards der Produktplanung und Systementwicklung orientieren. Die Systematik soll folgende Bestandteile umfassen:

- **Strukturierte Vorgehensmodelle** müssen den Kern der Systematik bilden. Sie dienen als Leitfaden für die beteiligten Fachleute und geben die für die systematische Release-Planung notwendigen Arbeitsschritte vor.
- Zur Durchführung einzelner Arbeitsschritte sind **Methoden** und **Werkzeuge** vorzusehen, die z.B. die Auswirkungsanalyse und Klassifizierung von Änderungen unterstützen. Die Methoden und Werkzeuge dienen der Beschaffung von Informationen, die für eine fundierte Planung unerlässlich sind.
- Als Kooperationskern für die beteiligten Fachleute wird eine **disziplinübergreifende Systembeschreibung** benötigt, die eine kundenorientierte, funktionale Sicht mit einer technischen Sicht auf das System verbindet. Nur so sind Analyse und Planung technischer Änderungen unter Berücksichtigung des bestehenden Systems möglich.

2.7 Anforderungen an die Systematik

Aus der Problemanalyse resultieren folgende Anforderungen an eine *Systematik zur Release-Planung intelligenter technischer Systeme*:

A1) Durchgängiger Planungsprozess: Die Systematik muss verschiedene Planungsebenen mit unterschiedlichem Planungshorizont abdecken und die Wechselwirkungen zwischen diesen Ebenen berücksichtigen. Dies reicht von der langfristigen, strategischen Definition kundenrelevanter Releases bis hin zur kurzfristigen, operativen Umsetzungsplanung unter Berücksichtigung begrenzter Ressourcen.

A2) Strukturierung des Release-Plans: Die Systematik muss die Erarbeitung der grundlegenden Struktur des Release-Plans unterstützen. Die Struktur ist sowohl auf markt- und wettbewerbsstrategische Überlegungen als auch auf die Beschaffenheit der Systemarchitektur, die Technologieplanung und -entwicklung sowie die verfügbaren Entwicklungsressourcen abzustimmen.

A3) Gesteuerte Produktwertsteigerung: Änderungen dienen u.a. der Produktwertsteigerung aus Sicht des Kunden – entweder durch die verbesserte Bedürfnisbefriedigung oder durch eine Reduzierung des Verkaufspreises. Durch eine geeignete Bewertung müssen Änderungstypen unterschieden werden, so dass ihre Umsetzung im Sinne einer gezielten Produktwertsteigerung geplant werden kann.

A4) Berücksichtigung von Stakeholder-Präferenzen: Die Umsetzung einzelner Änderungen unterliegt teils konkurrierenden Stakeholder-Präferenzen. Die verschiedenen Stakeholder sind daher gezielt zu involvieren, so dass Präferenzen und Prioritäten bei der Umsetzungsplanung berücksichtigt werden können.

A5) Änderungsauswirkungen und -abhängigkeiten: Die Auswirkungen und Abhängigkeiten einer Änderung beeinflussen maßgeblich deren Umsetzungsplanung. Die Identifikation von Änderungsauswirkungen und -abhängigkeiten ist daher gezielt zu unterstützen, in eine Änderungsklassifikation zu überführen und im Ergebnis in der Release-Planung zu berücksichtigen.

A6) Harmonisierte Bündelung von Änderungen zu Releases: Änderungen lassen sich nach unterschiedlichen Kriterien zu einem Release bündeln (z.B. Dringlichkeit, Modulzugehörigkeit). Neben diesen Kriterien sind weitere Rahmenbedingungen und Restriktionen (z.B. begrenzte Ressourcen) zu beachten. Die Bündelung von Änderungen zu einem Release muss einer Logik folgen, die im Rahmen der Release-Planung abgestimmt auf strategische Überlegungen festzulegen ist.

A7) Interdisziplinarität: Um die Anwendbarkeit der Systematik auf intelligente technische Systeme zu gewährleisten, ist insb. der mit diesen Systemen verbundenen Interdisziplinarität Rechnung zu tragen. Neben den an der Systementwicklung beteiligten Fachdisziplinen sind daher die Kompetenzen der Produktplanung und des Projektmanagements zu integrieren.

A8) Unterstützung durch modellbasierte Systembeschreibung: Die Release-Planung erfordert im Spannungsfeld von Produktplanung, Systementwicklung und Produktmanagement einen gemeinsamen Kooperationskern. Im Sinne des MBSE bietet sich die modellbasierte Systembeschreibung zur Kommunikation, Koordination, Dokumentation und Entscheidungsunterstützung an. Die Systembeschreibung muss die kundenorientierte und technische Sicht gleichermaßen vereinen und darüber hinaus der zeitlichen Varianz bei der Weiterentwicklung eines technischen Systems Rechnung tragen.

A9) Systematisches Vorgehen und Praktikabilität: Die Release-Planung ist ein iterativer Prozess, der auf verschiedenen Planungsebenen mit unterschiedlichem Zeithorizont durchzuführen ist. Dem Begriffsverständnis einer Systematik folgend muss dieser Prozess durch ein systematisches und reproduzierbares Vorgehen sowie geeignete Hilfsmittel unterstützt werden. Um die Praktikabilität der Systematik sicherzustellen, ist dazu – wenn möglich – auf etablierte und rechnergestützte Methoden zurückzugreifen.

Die beschriebenen Anforderungen bilden die Rahmenbedingungen für die Erarbeitung der *Systematik zur Release-Planung intelligenter technischer Systeme*. Anhand dieser Anforderungen werden in Kapitel 3 bestehende Ansätze gespiegelt, um den wissenschaftlichen Handlungsbedarf herauszustellen.

3 Stand der Technik

Vor dem Hintergrund der Problemanalyse untersucht dieses Kapitel bestehende Ansätze und Arbeiten, die im Kontext der Release-Planung intelligenter technischer Systeme relevant sind. In **Abschnitt 3.1** werden zunächst Ansätze zur Spezifikation technischer Systeme analysiert. Davon ausgehend befassen sich die **Abschnitte 3.2** und **3.3** mit Ansätzen zur Analyse, Klassifizierung und Priorisierung von Änderungen. Aufbauend auf diesen Grundlagen werden in den folgenden Abschnitten bestehende Vorgehensmodelle zur Planung technischer Systeme analysiert. Die Release-Planung weist eine enge Schnittstelle zur strategischen Produktplanung auf. Bestehende Ansätze im Kontext der strategischen Produktplanung werden daher in **Abschnitt 3.4** diskutiert, ehe schließlich in **Abschnitt 3.5** etablierte Ansätze der Produkt-Release-Planung analysiert werden. Durch die Bewertung des Stands der Technik anhand der Anforderungen aus der Problemanalyse wird in **Abschnitt 3.6** der Handlungsbedarf für diese Arbeit hergeleitet.

3.1 Ansätze zur Spezifikation technischer Systeme

Wesentlicher Erfolgsfaktor für die Release-Planung ist ein interdisziplinäres Verständnis über die Auswirkungen von Änderungen auf das technische System, die damit verbundenen technischen Zusammenhänge sowie die entstehende Varianz. Vor diesem Hintergrund bedarf es einer Informationsbasis, die alle relevanten Aspekte eines technischen Systems fachdisziplinübergreifend beschreibt und die Analyse von Änderungsauswirkungen ermöglicht. Im Folgenden werden Ansätze vorgestellt, die mit jeweils unterschiedlichem Schwerpunkt einer disziplinübergreifenden Beschreibung technischer Systeme dienen. Schwerpunkte der Beschreibung sind die Aspekte Anforderungen, Struktur und Verhalten, aber auch die Varianz eines technischen Systems.

3.1.1 Spezifikationstechnik CONSENS

Zur mechatronischen Systembeschreibung wurde am HEINZ NIXDORF INSTITUT im Sonderforschungsbereich 614 die **Spezifikationstechnik CONSENS** (CONceptual design Specification technique for the Engineering of complex Systems) entwickelt [DDG+14]. Die Spezifikationstechnik dient heute als Basis für vielfältige weiterführende Forschungsarbeiten im Kontext des Model-Based Systems Engineerings [IKD+13], [KDB+13], [KDH+13], [KBD+14]. CONSENS umfasst eine Modellierungssprache sowie eine Methode zu deren Anwendung. Ziel ist die ganzheitliche und disziplinübergreifende Beschreibung eines fortgeschrittenen mechatronischen Systems im Rahmen des Systementwurfs, aber auch im Reengineering bestehender Systeme. Die Konzeption eines technischen Systems wird dazu durch sieben Aspekte beschrieben (Bild 3-1): Umfeld, Anwendungsszenarien, Anforderungen, Funktionen, Wirkstruktur, Gestalt und Verhalten. Alle Aspekte werden im Sinne des MBSE rechnerintern durch Partialmodelle be-

schrieben und sind wechselseitig durch partialmodellübergreifende Beziehungen verknüpft. Die Partialmodelle bilden die Ausgangsbasis für den fachdisziplinspezifischen Entwurf.

Eine **Methode** unterstützt die Anwendung der Modellierungssprache. Sie gibt eine grundsätzliche Reihenfolge zur Erarbeitung der Partialmodelle vor, die jedoch fallspezifisch angepasst werden kann. Ausgehend von Umfeld und Anwendungsszenarien werden die wesentlichen Anforderungen spezifiziert. Diese dienen als Basis für die Erarbeitung der Systemfunktionen und der davon abgeleiteten Wirkstruktur. Abschließend erfolgt die Spezifikation des Verhaltens und der Gestalt [DDG+14]. Im Folgenden werden die einzelnen Aspekte kurz erläutert.

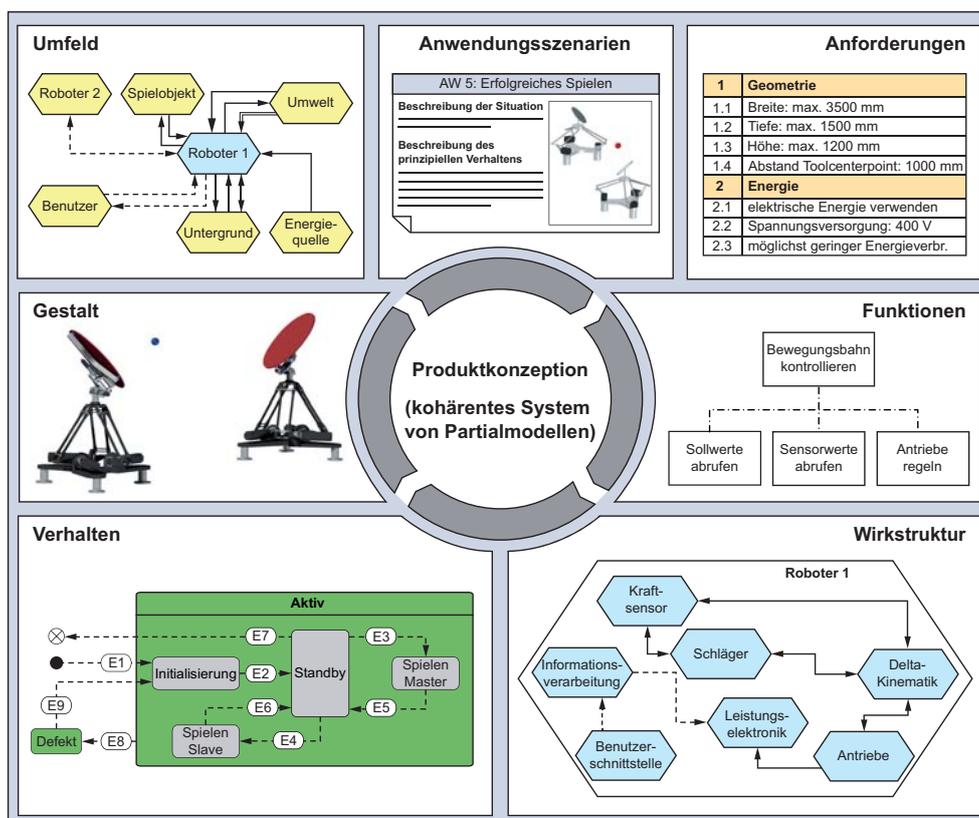


Bild 3-1: Partialmodelle der Spezifikationstechnik CONSENS [GTS+14]

Umfeld: Das Umfeldmodell definiert die Systemgrenzen und beschreibt die Beziehungen des zu entwickelnden Systems zum Umfeld. Das System wird dabei als Blackbox betrachtet; die Beziehungen werden über gerichtete Stoff-, Energie- und Informationsflüsse dargestellt. Beziehungen sind bspw. Benutzereingaben, Umgebungstemperaturen und Vibrationen [DDG+14].

Anwendungsszenarien: In diesem Partialmodell werden situationsspezifische Sichten auf das zu entwickelnde System beschrieben. Sie definieren, wie sich ein System in einem bestimmten Zustand oder einer bestimmten Situation verhalten soll. Mögliche Zustände und Situationen werden dabei über den gesamten Produktlebenszyklus antizipiert. Die

Anwendungsszenarien werden in Steckbriefen dokumentiert und dienen als Grundlage für die Anforderungsermittlung [DDG+14].

Anforderungen: Sämtliche Anforderungen an das System werden in einer Anforderungsliste strukturiert; diese repräsentiert das Partialmodell Anforderungen. Eine Konkretisierung der Anforderungen erfolgt durch qualitative und quantitative Angaben. Die Erfüllung der Anforderungen bildet den Maßstab für die Qualität des Systems [DDG+14].

Funktionen: Funktionen beschreiben lösungsneutral den Zusammenhang zwischen einer Eingangs- und einer Ausgangsgröße; ihr Ziel ist die Erfüllung einer spezifischen Aufgabe [PBF+07, S. 215]. Sie werden aus den zuvor spezifizierten Anforderungen des Systems hergeleitet, hierarchisch in einer Funktionshierarchie strukturiert und in Teilfunktionen untergliedert. Hierbei erfolgt die Untergliederung solange, bis sinnvolle Lösungen zur Erfüllung der Teilfunktion gefunden sind [DDG+14].

Wirkstruktur: Die Wirkstruktur beschreibt den grundsätzlichen Aufbau und die Wirkungsweise des Systems. Dazu werden die Systemelemente und deren Beziehungen modelliert. Je nach Abstraktionsgrad bzw. Entwicklungsfortschritt kann es sich bei den Systemelementen um abstrakte Wirkprinzipien, Module, Baugruppen, Bauteile oder Softwarekomponenten handeln. Die Beziehungen zwischen den Elementen werden analog zum Umfeldmodell durch Energie-, Stoff- und Informationsflüsse dargestellt [DDG+14]. BRANDIS ergänzt im Wechselspiel mit dem Partialmodell Gestalt einen weiteren Beziehungsaspekt – den Bauzusammenhang [Bra14, S. 116 ff.].

Verhalten: Bei der Entwicklung mechatronischer Systeme kommt dem Verhalten, also Systemzuständen, Zustandsübergängen und Ablaufprozessen, eine besondere Bedeutung zu. Im gleichnamigen Partialmodell werden diese daher explizit modelliert. Die Modelle dienen als Grundlage für den Software- und Reglerentwurf und spezifizieren sowohl das Verhalten des Gesamtsystems als auch einzelner Systemelemente [DDG+14].

Gestalt: Erste Modelle zur Gestalt sind bereits im Rahmen des Systementwurfs zu erarbeiten. Die Modellierung erfolgt mit Hilfe gängiger 3D-CAD-Systeme und spezifiziert Anzahl, Form, Lage, Anordnung und Art der Wirkflächen und Wirkorte [DDG+14].

Bewertung: Die Spezifikationstechnik CONSENS bildet mit den beschriebenen Aspekten in vielerlei Hinsicht eine gute Ausgangsbasis zur Unterstützung der Release-Planung. Die allgemeinverständliche und intuitive Systembeschreibung erlaubt den an der Release-Planung beteiligten Fachleuten die systematische Analyse komplexer technischer Zusammenhänge. Damit bietet sich CONSENS insb. zur fachdisziplinübergreifenden Analyse von Änderungsauswirkungen und -abhängigkeiten an. Schwächen offenbart die Spezifikationstechnik, wenn es um die Beschreibung der Variabilität eines Systems geht. So wird weder der räumlichen noch der zeitlichen Varianz Rechnung getragen. Es ist zu klären, inwieweit die Spezifikationstechnik CONSENS für den Einsatz im Kontext der Release-Planung modifiziert bzw. erweitert werden muss.

3.1.2 Modellierungstechniken basierend auf SysML

Die **Systems Modeling Language** (SysML) ist eine Sprache zur Modellierung komplexer technischer Systeme [Alt12], [FMS12]. Sie ist eine Erweiterung des UML-Standards⁴² (Unified Modeling Language) und wurde in einer gemeinsamen Initiative der OBJECT MANAGEMENT GROUP und INCOSE speziell für das Systems Engineering entwickelt. Zielsetzung war eine standardisierte Sprache zur Modellierung, Analyse und Verifikation komplexer technischer Systemen. Die erstmalige Standardisierung erfolgte 2006 mit der SysML 1.0. Aktueller Stand ist die Version 1.3 [OMG12]. Einen Überblick über die in der SysML definierten Diagramme zeigt Bild 3-2.

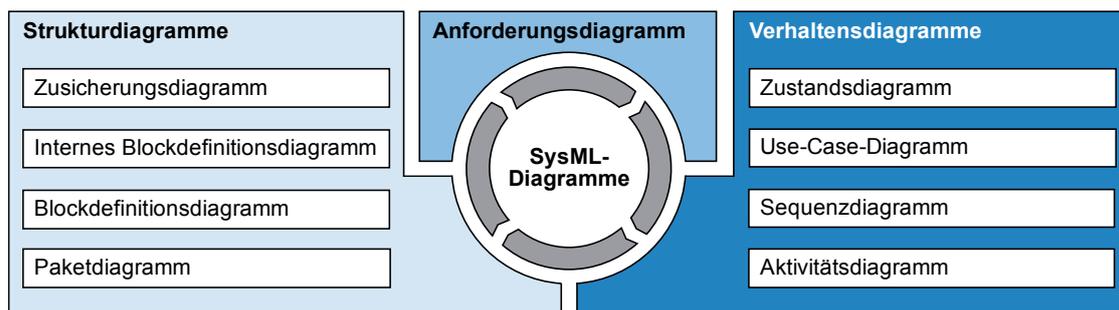


Bild 3-2: Diagramme der SysML [vgl. Wei06, S.160]

Die Diagramme sind auf oberster Ebene in drei Aspekte gegliedert: Struktur, Anforderungen und Verhalten. **Strukturdiagramme** beschreiben mit Hilfe von Blöcken informationsverarbeitende und physikalische Elemente eines Systems, einschließlich deren Zusammenhänge und Eigenschaften. Beziehungen zwischen Elementen werden mit Hilfe von Ports, Konnektoren und Flüssen festgehalten. Das **Anforderungsdiagramm** dient der Darstellung funktionaler und nicht funktionaler Anforderungen sowie damit verbundenen Beziehungen, wie Ableitungs-, Enthält-, Erfüllungs- und Verfolgungsbeziehungen. Die insgesamt vier **Verhaltensdiagramme** beschreiben das Systemverhalten mit unterschiedlichen Schwerpunkten. Das Use-Case-Diagramm spezifiziert bspw. die Interaktionen von Benutzern und/oder externen Systemen mit dem zu entwickelnden System [Wei06, S. 157ff.].

Im Gegensatz zu CONSENS ist die SysML unabhängig von einer Methode entwickelt worden, die die Anwendung der Sprache und der damit verbundenen Diagramme beschreibt. In den letzten Jahren ist rund um die SysML jedoch eine Community entstanden, die unterschiedliche **Methoden** zur Anwendung der Sprache vorschlägt. Die **FAS-Methode** (**F**unctional **A**rchitecture for **S**ystems) beschreibt ein sprachenunabhängiges Vorgehen von den Anforderungen bis zur Erstellung der funktionalen Architektur eines Systems. WEILKIENS und LAMM greifen die Methode auf und stellen ein darauf angepasstes SysML-Profil vor [LW10], [Kai13, S. 53f]. Dieser Ansatz wird auch mit **SYSMOD** und

⁴² Die UML wird von der OBJECT MANAGEMENT GROUP (OMG) entwickelt und ist in der Version 2.4.1 in der ISO 19505-1 und ISO 19505-2 standardisiert [ISO19505-1], [ISO19505-2].

OOSEM (Object-oriented Systems Engineering Method) verfolgt, bei denen es sich um weitere Methoden zur Modellierung komplexer Systeme basierend auf SysML handelt [Wei06], [FMS12]. Auch für die Spezifikationstechnik CONSENS liegt inzwischen ein solches SysML-Profil vor [IKD+13].

Bewertung: Ergänzt um die passende Methode ist die Modellierungssprache SysML eine ähnlich gute Ausgangsbasis zur Unterstützung der Release-Planung wie CONSENS. Aufgrund der Vielzahl der Konstrukte und des immensen Umfangs ist die Einarbeitung jedoch zeitaufwändig und nicht immer intuitiv. Dies ist insb. dann ein Hemmnis, wenn es um die Einarbeitung von Experten aus nicht-technischen Disziplinen wie dem Projekt- und dem Produktmanagement geht.

3.1.3 A3 Architecture Overview

Der Ansatz **A3⁴³ Architecture Overview** wurde an der Universität Twente speziell vor dem Hintergrund evolutionärer Systeme entwickelt. Ziel ist die Verbesserung der Kommunikation von Wissen bei der Weiterentwicklung komplexer technischer Systeme, insb. dem Wissen über die Architektur des Systems [Bor10, S. VII f.]. Mithilfe des Ansatzes wird die betrachtete Architektur eines Systems auf Vorder- und Rückseite eines DIN-A3-Blattes dargestellt. Die Beschränkung des verfügbaren Platzes auf ein DIN-A3-Blatt soll gewährleisten, dass nur relevante Informationen diskutiert werden [Bor10, S. 88]. Kern des Ansatzes ist das in Bild 3-3 dargestellte Dokumentationsschema.

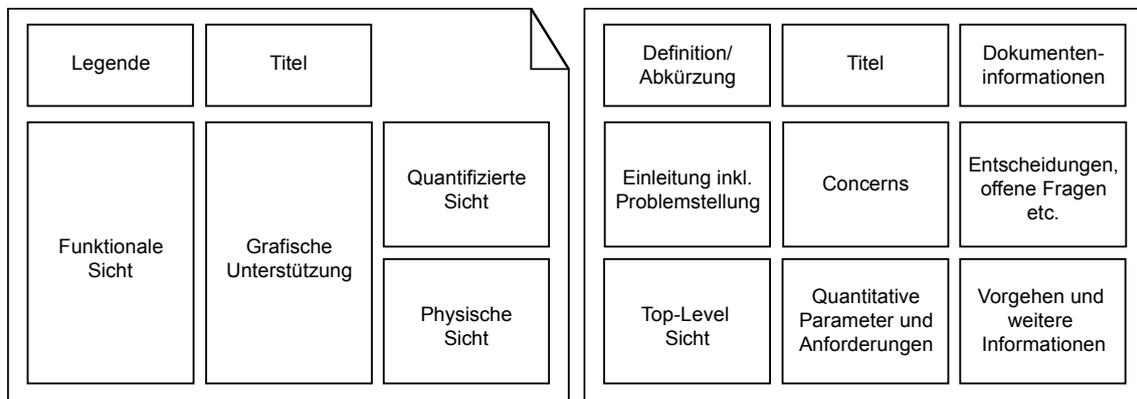


Bild 3-3: Aufbau der A3-Architekturdarstellung nach [Bor10, S. 94 f.]

Das **Dokumentationsschema** gliedert sich in zwei Bereiche: Auf der einen Seite des A3-Blattes wird ein strukturiertes Modell dokumentiert, das verschiedene, miteinander verbundene Sichten (z.B. funktionale und physische Sicht) auf das System darstellt. Für die Modellierung werden zwar vereinzelt Modellkonstrukte vorgegeben, es existiert jedoch keine explizite Modellierungssprache, insb. um unproduktive Diskussionen über die korrekte Darstellungsform zu vermeiden. Auf der anderen Seite des A3-Blattes werden mit-

43 A3 entspricht dem deutschen Standard-Papierformat DIN-A3 und damit den Maßen 297 x 420 mm.

tels einer textuellen Beschreibung des Systems strukturiert Hintergrundinformationen bereitgestellt. Auch die textuelle Beschreibung ist in verschiedene Elemente gegliedert und umfasst z.B. Informationen zu quantitativen Parametern, Anforderungen und offenen Fragen. Welches Element wie viel Platz in Anspruch nimmt, ist nicht festgelegt und kann je nach Bedarf angepasst werden [Bor10, S. 93 f.].

Die Anwendung des Dokumentationsschemas wird durch eine **Methode** zum sogenannten Reverse Architecting unterstützt. Die Methode gliedert sich in die drei Schritte Informationssammlung, Abstraktion und Präsentation, die iterativ durchlaufen werden. Ziel ist es, das in der Organisation verteilte, implizite Wissen über die Architektur eines bestehenden Systems zu sammeln und mittels des oben beschriebenen Dokumentationsschemas zu kommunizieren [Bor10, S. 73ff.].

Bewertung: Der A3 Architecture Overview wurde speziell vor dem Hintergrund der Evolution technischer Systeme entwickelt und zeichnet sich durch besondere Einfachheit bei der Beschreibung der Architektur eines technischen Systems aus. Dadurch ist der Ansatz besonders attraktiv für die industrielle Anwendung. Der Ansatz hat jedoch keinen Anspruch auf Vollständigkeit, Formalität und Transformierbarkeit der generierten Modelle, so dass er der Kernidee des MBSE widerspricht. Darüber hinaus werden mittels der im Dokumentationsschema beschriebenen Aspekte relevante technische Zusammenhänge nur unzureichend beschrieben, so dass eine Analyse von Änderungsauswirkungen nur bedingt möglich ist.

3.1.4 Matrixbasierte Produktmodellierung

Die **matrixbasierte Produktmodellierung** geht auf Forschungsarbeiten von STEWARD und weitere Autoren zurück, die in den 1960er Jahren matrixbasierte Methoden zur Modellierung und Analyse von Systemstrukturen und Prozessen entwickelt haben [Ste62], [Ste81]. Kern dieser Methoden ist die Strukturierung von Elementen und deren Beziehungen innerhalb einer Matrix sowie die Auswertung dieser Zusammenhänge mittels spezieller Algorithmen. Seit mehreren Jahren widmet sich eine eigene Forschungsgemeinschaft⁴⁴ der Weiterentwicklung und Anwendung dieser Methoden, die insb. zur Beschreibung und Analyse von komplexen technischen Systemen und den zugehörigen Entwicklungsprozessen zum Einsatz kommen [HC88], [Epp91], [LMB09], [EB12].

Einen Ansatz zur **Beschreibung der Systemarchitektur mechatronischer Systeme** mittels eines matrixbasierten Ansatzes schlagen LINDEMANN ET AL. vor [BDP+07], [HDL10]. Das betrachtete System wird dazu mit Hilfe von mehreren Teilmatrizen beschrieben, bei denen jeweils bestimmte Aspekte des Systems modelliert werden. Der Ansatz zielt darauf ab, die funktions- und die komponentenorientierte Sichtweise auf das

⁴⁴ Die Forschungsgemeinschaft trifft sich einmal jährlich auf der „International Design Structure Matrix Conference“ und präsentiert dort die neuesten Forschungsergebnisse auf dem Gebiet der matrixbasierten Produktmodellierung [DSM15-ol].

System in einem Modell zu vereinen und transparent miteinander zu verknüpfen. Resultat ist eine Multi Domain Matrix (MDM) (Bild 3-4).

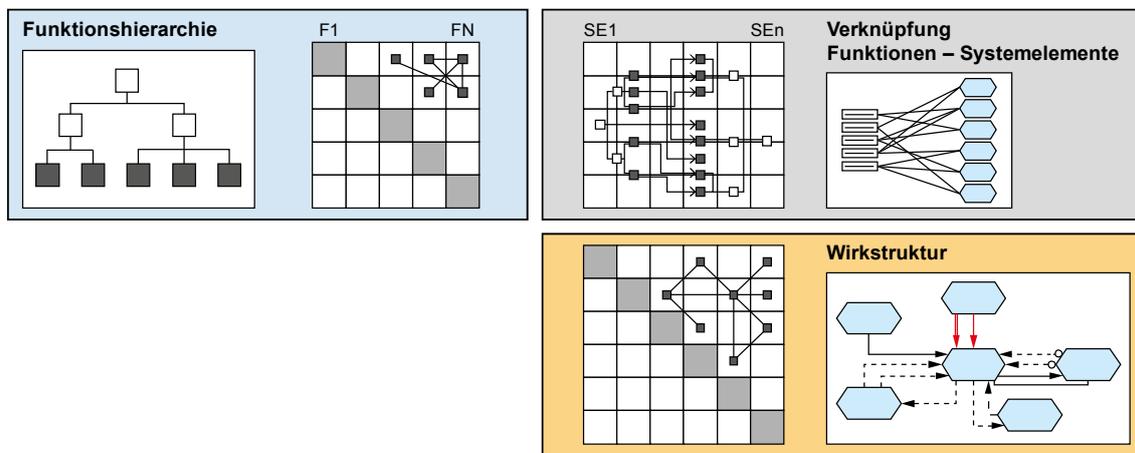


Bild 3-4: MDM-basierte Modellierung eines mechatronischen Systems nach [HDL10]

Die funktionsorientierte Sicht ergibt sich aus der Funktionshierarchie (Hauptfunktion bis hin zu Teilfunktion) und der Funktionsstruktur (funktionale Abhängigkeiten); die komponentenorientierte Sicht aus der Systemhierarchie (Module, Systemelemente) und der Systemstruktur (Wirkzusammenhänge der Systemelemente). Beide Aspekte werden in Teilmatrizen beschrieben, die jeweils nur eine Domäne (hier Funktionen und Systemelemente) umfassen und daher als Design Structure Matrix (DSM) bezeichnet werden. Die Verknüpfung der Aspekte Funktionen und Systemelemente erfolgt anschließend in einer zusätzlichen Matrix, in der die Zusammenhänge der einzelnen Funktionen und der zugehörigen Systemelemente abgebildet werden. Diese Matrix wird als Domain Mapping Matrix (DMM) bezeichnet [HDL10].

Bewertung: Die matrixbasierte Produktmodellierung erlaubt es, domänenspezifische Zusammenhänge (in Form einer DSM) und domänenübergreifende Zusammenhänge (in Form einer DMM) in einer Gesamtmatrix darzustellen. Die Matrixdarstellung ist leicht erweiterbar und kann zusätzliche Aspekte (bspw. die Produktion, die Abbildung von Varianz) einschließen. Ein wesentlicher Vorteil der Modellierung mechatronischer Systeme mittels Matrizen ist die Möglichkeit einer umfassenden Analyse der strukturellen Zusammenhänge. Inzwischen existiert eine Vielzahl von Methoden zur Analyse von Teilstrukturen und zur darauf aufbauenden strukturellen Optimierung. So gehen bspw. Arbeiten zur Modularisierung technischer Systeme [Ste07] und zur Auswirkungsanalyse von Änderungen auf matrixbasierte Ansätze zurück [CSE01]. Die matrixbasierte Modellierung setzt allerdings eine genaue Kenntnis über die Funktionen, Systemelemente und deren Abhängigkeiten voraus, die es in fachdisziplinübergreifenden Teams zu erheben gilt. Hier bieten die zuvor beschriebenen modellbasierten Ansätze wie CONSENS und SysML konkrete Hinweise zur Informationserhebung und darüber hinaus einen intuitiveren Zugang zum System. Aus diesem Grund bietet die matrixbasierte Modellierung eine sinnvolle Ergänzung zu den vorangegangenen Modellierungstechniken, ist jedoch als einzelner Ansatz für sich nicht ausreichend.

3.1.5 Feature-Modelle

Während die bisher genannten Ansätze zur Beschreibung technischer Systeme auf die Aspekte Anforderung, Struktur und Verhalten abzielen, handelt es sich beim **Feature-Konzept** um einen verbreiteten Ansatz zur Handhabung der Variabilität von Produkten. Es unterstützt die Strukturierung, Wiederverwendung und Variation eines Produkts über alle Phasen des Lebenszyklus [ABK+13, S. 18f.]. Jede Produktvariante wird dazu durch eine Feature-Kombination beschrieben, die sich durch bestimmte Features von anderen Kombinationen unterscheidet. Zur Spezifikation dieser Zusammenhänge eignen sich sogenannte Feature-Modelle⁴⁵. **Feature-Modelle** unterscheiden sich je nach Einsatzzweck in ihrer Notation. Bild 3-5 zeigt zwei alternative Ansätze, die mittels eines horizontalen bzw. vertikalen Baumdiagramms die Varianten eines Produkts darstellen.

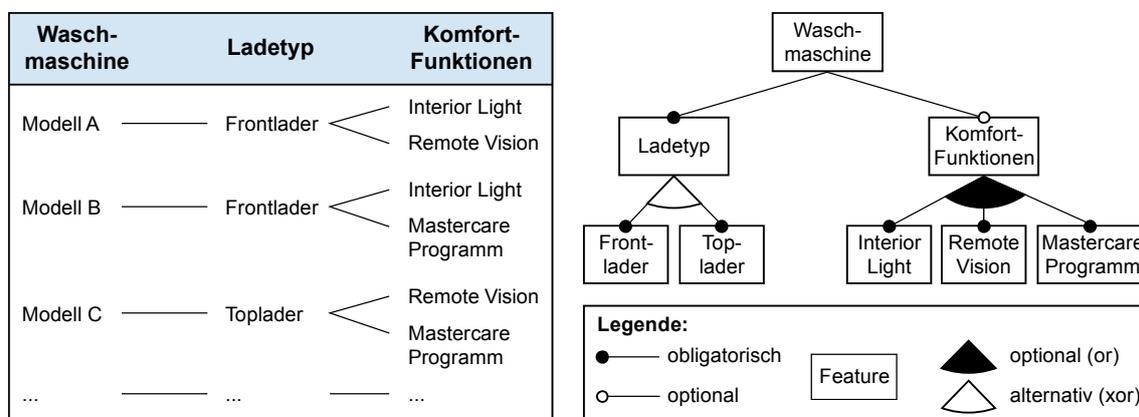


Bild 3-5: Alternative Ansätze zur Beschreibung der Variabilität technischer Systeme

Das Feature-Model links beschreibt die Varianten eines Produkts in einem horizontalen Baumdiagramm. Jede vertikale Ebene entspricht genau einem Produkt-Feature. Die Einträge je Ebene entsprechen den einzelnen Ausprägungen eines Features. Jede Ausprägung kann mehrfach in einem Baumdiagramm vorkommen, da verschiedene Varianten zum Teil identische Eigenschaften besitzen. Sie unterscheiden sich jedoch in mindestens einer Eigenschaft, so dass jeder Ast einer Produktvariante entspricht [Kes12].

Das Feature-Model rechts repräsentiert die Varianten eines Produkts in einer anderen Notation, die in dieser Form erstmalig von KANG ET AL. im Kontext der Feature-oriented Domain Analysis (FODA) vorgestellt wurde [KCH+90]. In einem vertikalen Baumdiagramm werden die Features hierarchisiert und verknüpft. Es wird dabei in obligatorische und optionale Features unterschieden, die wiederum eine Menge an Sub-Features umfassen. Ein weißer Bogen bedeutet, dass exakt ein Sub-Feature ausgewählt werden kann; ein schwarzer Bogen signalisiert, dass die Sub-Features dieser Gruppe kombiniert eingesetzt werden können. Aufgrund der einfachen und übersichtlichen Darstellung auch großer Variantenspektren ist diese Notation in verschiedensten Anwendungsfeldern verbreitet. Ins-

⁴⁵ Feature-Modelle werden auch als Merkmalsbäume bezeichnet [Kes12].

besondere in der Softwareentwicklung reichen diese von der strategischen Produktplanung über das Konfigurationsmanagement bis hin zur Release-Planung und zum Projektmanagement [MBD+05], [TS09].

Evolutionäre Feature-Modelle (EvoFM)

Feature-Modelle werden in der Regel zur Planung und Pflege von variantenreichen Produkten eingesetzt. Das Variantenspektrum eines Produkts ist jedoch üblicherweise nicht statisch, sondern verändert sich über die Zeit aufgrund technologisch bedingter Änderungen oder veränderter Kundenbedürfnisse. Diesem Aspekt widmen sich BOTTERWECK ET AL. [BPD+10]. Sie schlagen evolutionäre Feature-Modelle (EvoFM) vor, die die Entwicklung eines Produkts über die Zeit beschreiben. Dabei liegt der Fokus, wie beim herkömmlichen Feature-Modell, auf den Gemeinsamkeiten und den Varianten des Produkts. Die Notation unterscheidet sich jedoch, wie in Bild 3-6 dargestellt.

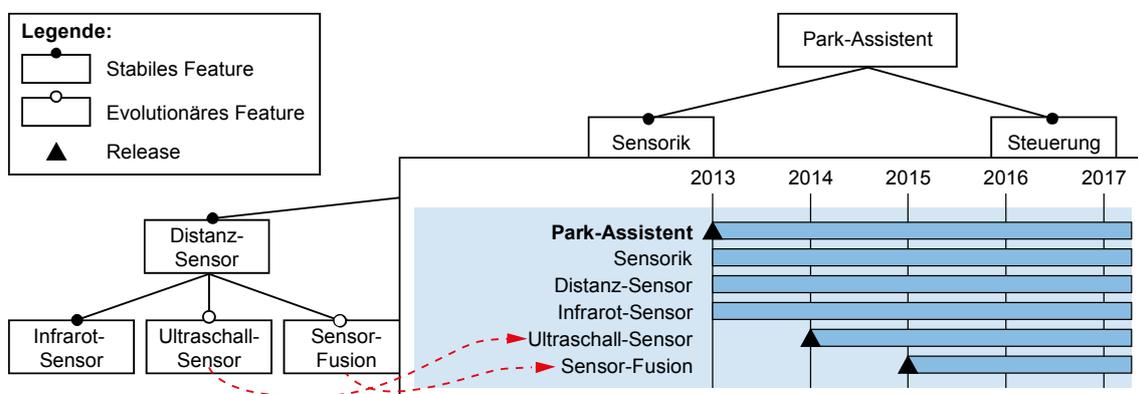


Bild 3-6: Evolutionäres Feature-Modell (EvoFM) nach [BPD+10]

In der Notation werden stabile und evolutionäre Features unterschieden: Stabile Features sind in der gesamten Produktgeneration enthalten; evolutionäre Features verändern sich über die Zeit bzw. sind erst ab einem bestimmten Zeitpunkt im Produkt enthalten. Um einen Überblick über die Zeitpunkte der Veränderungen zu erhalten, wird das evolutionäre Feature-Modell um eine Produkt-Roadmap ergänzt [BPD+10].

Bewertung: Feature-Modelle eignen sich sehr gut, um die Variabilität eines Produkts zu beschreiben. Die Notation ist einfach zu verstehen und intuitiv anwendbar, was sie zu einem geeigneten Kommunikationsmedium für Kunden, Produktmanagement und Produktentwicklung macht. Insbesondere evolutionäre Feature-Modelle bieten sich durch die Beschreibung der zeitlichen Varianz zur Unterstützung der Release-Planung an. Ergänzt um eine Produkt-Roadmap ermöglichen sie die modellbasierte Beschreibung der Weiterentwicklung eines technischen Systems über die Zeit. Durch den Fokus auf die Features eines Systems bleiben dabei allerdings technische Aspekte wie Anforderungen und Struktur unberücksichtigt. Eine Analyse von Änderungsauswirkungen ist nicht möglich. Vor diesem Hintergrund ist die Kombination der Feature-Modelle mit weiteren Ansätzen zur Systemspezifikation (z.B. CONSENS) zu prüfen.

3.2 Ansätze zur Analyse von technischen Änderungen

Die Weiterentwicklung eines technischen Systems resultiert in Änderungen, die sich sowohl direkt auf die zu ändernden Systemelemente als auch indirekt auf damit in Beziehung stehende Systemelemente auswirken. Das Wissen über diese Auswirkungen ist entscheidend, um die Weiterentwicklung eines technischen Systems systematisch planen zu können. Dabei geht es sowohl um die Identifikation der betroffenen Systemelemente als auch um eine grobe Abschätzung der mit der Änderung verbundenen Risiken insb. mit Blick auf den Änderungsaufwand. Der Analyse von Änderungsauswirkungen widmet sich daher eine Vielzahl von Ansätzen, die auch als Change-Impact-Analysen bezeichnet werden. Die Ursprünge der Change-Impact-Analyse liegen in der Softwareentwicklung, für die bereits in den 1990er Jahren verschiedene Analysetechniken vorgeschlagen wurden. Einen Überblick liefern BOHNER und ARNOLD [BA96]. Inzwischen existieren auch für die Analyse komplexer technischer Systeme verschiedene Ansätze. Einen Überblick liefern hier JARRETT ET AL. [JEC+11]. Im Folgenden werden ausgewählte Ansätze zur Analyse von Änderungen und den damit verbundenen Auswirkungen vorgestellt. Der Fokus liegt auf Ansätzen, die auf ein disziplinübergreifendes Verständnis technischer Änderungen abzielen.

3.2.1 Change Prediction Method nach CLARKSON ET AL.

Eine häufig zitiertes Ansatz zur Auswirkungsanalyse von Änderungen ist die Change Prediction Method (CPM) nach CLARKSON ET AL. [CSE01], die am CAMBRIDGE ENGINEERING DESIGN CENTER kontinuierlich weiterentwickelt [KEE+05], [AWC12], [KCC12] und inzwischen durch ein Softwarewerkzeug, den Cambridge Advanced Modeller (CAM), unterstützt wird [WWN+10]. Die Methode dient der Vorhersage von Änderungsförpflanzungen und ist damit Grundlage für die Planung und Umsetzung von technischen Änderungen. Ziel ist es, diejenigen Systemelemente zu identifizieren, die voraussichtlich von einer bestimmten Änderung betroffen sind. Die Methode geht von einem bestehenden Produkt aus und gliedert sich in drei Phasen, die wiederum aus einzelnen Arbeitsschritten bestehen [CSE01]. Die Methode ist in Bild 3-7 dargestellt und wird im Folgenden zusammenfassend erläutert.

Die **erste Phase** dient der Ausgangsanalyse und liefert im Ergebnis ein Änderungsmodell, das mittels einer Produkt-Risiko-Matrix visualisiert wird. Die Erarbeitung des Änderungsmodells gliedert sich in drei Schritte: Ausgehend von einer vorliegenden Produktdokumentation und den Kenntnissen der beteiligten Entwickler erfolgt im *ersten Schritt* die Produktmodellierung, die die Gliederung des Produkts in seine wesentlichen Teilsysteme beschreibt. Im *zweiten Schritt* werden die identifizierten Teilsysteme in Abhängigkeitsmatrizen überführt, in denen die Abhängigkeiten zwischen den Teilsystemen nach zwei Kriterien bewertet und festgehalten werden: 1) nach Wahrscheinlichkeit und 2) Auswirkung einer Änderungsförpflanzung. Dies dient im *dritten Schritt* als Grundlage zur

Berechnung⁴⁶ der Vorhersagematrizen und damit des Risikos der Fortpflanzung technischer Änderungen, welches mittels einer Produkt-Risiko-Matrix visualisiert wird (siehe Bild 3-7). Das Rechteck innerhalb der Matrix symbolisiert farblich das Änderungsrisiko eines Teilsystems. Dieses setzt sich aus der Fortpflanzungswahrscheinlichkeit (Breite des Rechtecks) und dem Umsetzungsaufwand (Höhe des Rechtecks) zusammen [CSE01].

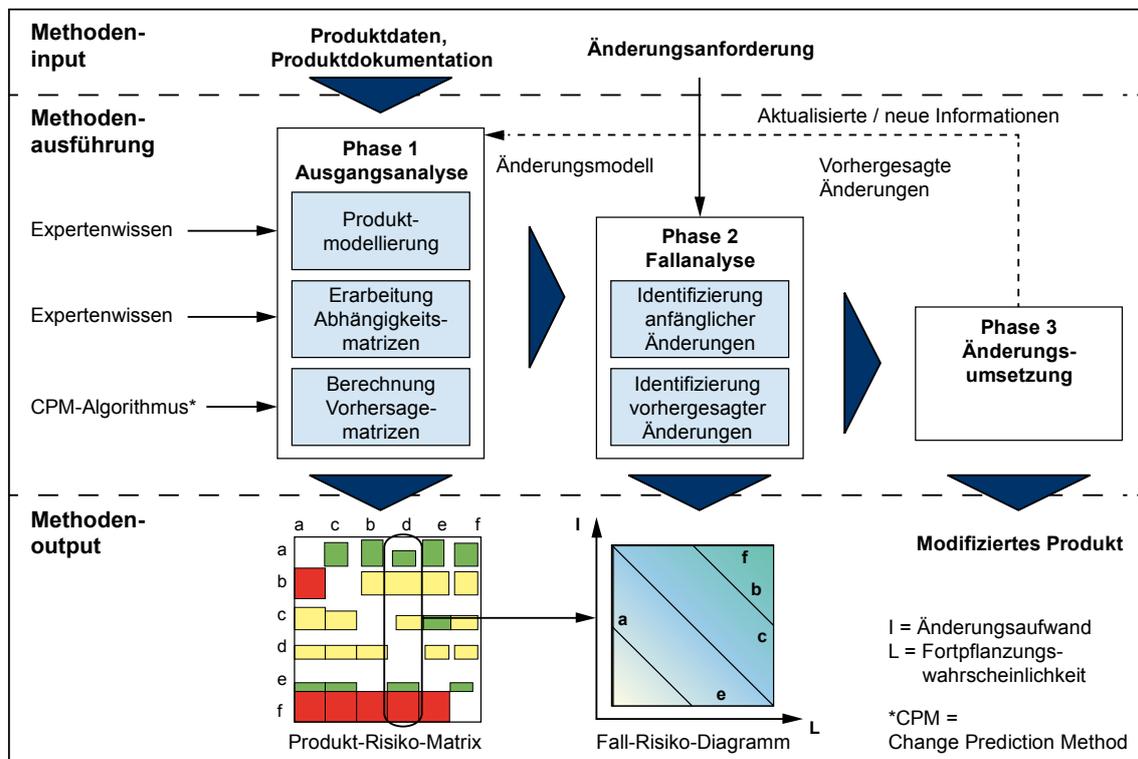


Bild 3-7: Vorgehen der Change Prediction Method nach [CSE01]

Die **zweite Phase** der CPM-Methode dient der eigentlichen Analyse von Änderungen im Sinne einer Fallanalyse. Eine Änderungsanforderung wird dazu zunächst hinsichtlich des primär betroffenen Teilsystems analysiert. Ausgehend vom zuvor erarbeiteten Änderungsmodell und der Produkt-Risiko-Matrix lässt sich anschließend das Risiko einer Änderungsfortpflanzung für die übrigen Teilsysteme ablesen. Zur verbesserten Priorisierung erfolgt eine Visualisierung mittels eines Fall-Risiko-Diagramms, in dem jedes Teilsystem hinsichtlich des Änderungsaufwands (I) sowie der Fortpflanzungswahrscheinlichkeit (L) dargestellt wird [CSE01].

Die **dritte Phase** der CPM-Methode ist die Änderungsumsetzung. Im Rahmen dieser Phase erfolgt eine Modifizierung des Produkts durch die Umsetzung der jeweiligen Änderung. Die Ergebnisse der vorangegangenen Phasen geben dabei Aufschluss über die

46 Die Berechnung der Änderungsfortpflanzung basiert auf speziellen Algorithmen, die neben direkten auch indirekte Änderungsauswirkungen im Sinne eines „Dominoeffektes“ einbeziehen. Für eine genauere Beschreibung dieser mathematischen Zusammenhänge sei an dieser Stelle auf die Primärquelle verwiesen.

voraussichtlich betroffenen Teilsysteme und unterstützen somit das Projektmanagement bei der Allokation der Entwicklungsressourcen [CSE01].

Bewertung: Die CPM-Methode nach CLARKSON ET AL. wurde speziell für die Analyse von Änderungsauswirkungen bei komplexen technischen Systemen entwickelt. Dabei ermöglicht das matrixbasierte Änderungsmodell in Kombination mit speziellen Algorithmen sowohl die Berücksichtigung direkter als auch indirekter Änderungsauswirkungen. Die Erzeugung des Änderungsmodells ist jedoch sehr aufwändig und erfordert viel Erfahrungswissen, insb. bei der Festlegung des Abstraktionsgrads, mit dem die Teilsysteme definiert werden. Die aufwändige Betrachtung einer einzelnen Änderung erschwert es darüber hinaus, Zusammenhänge und Abhängigkeiten zwischen Änderungen zu identifizieren.

3.2.2 Technology Infusion nach SUH ET AL.

Der bei XEROX entwickelte Technology Infusion Ansatz nach SUH ET AL. beschreibt ein Vorgehen zur Kosten-Nutzen-Bewertung bei der Implementierung neuer Technologien in einem bestehenden Produkt. Ein besonderer Fokus des Ansatzes liegt auf der Analyse technologiebedingter Änderungsauswirkungen und damit verbundener Aufwände. Die Implementierung einer neuen Technologie initiiert Änderungsanforderungen, deren Auswirkungen in drei Schritten analysiert werden (Bild 3-8).

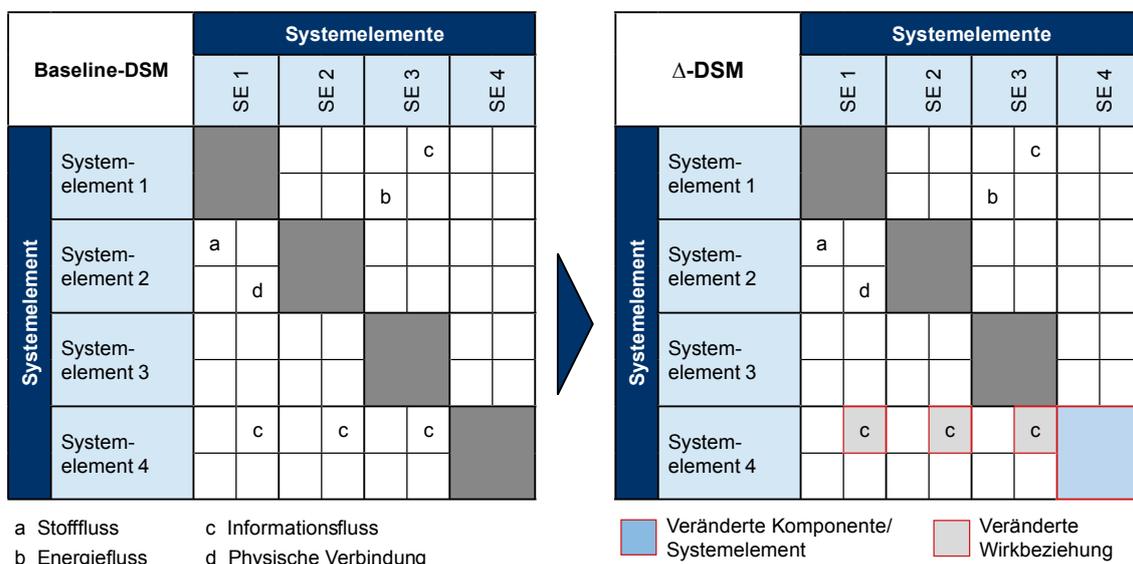


Bild 3-8: Änderungsanalyse mittels ΔDSM nach [SFM+09]

Im **ersten Schritt** wird eine sogenannte Baseline-DSM erarbeitet. Diese umfasst alle wesentlichen Teilsysteme des bestehenden Produkts sowie die Beziehungen zwischen den Teilsystemen. Es werden sowohl konstruktive Beziehungen als auch Stoff-, Energie- und Informationsflüsse innerhalb einer Matrix berücksichtigt und dokumentiert. Die so gewonnenen Produktinformationen dienen als Grundlage für die eigentliche Änderungsanalyse im **zweiten Schritt**. In einer Änderungs-DSM bzw. ΔDSM werden entsprechend

Bild 3-8 alle technologiebedingten Änderungen festgehalten. Dazu werden auf der Diagonalen der Δ DSM alle neuen bzw. zu modifizierenden Systemelemente dokumentiert; in den Zellen alle veränderten mechanischen Beziehungen sowie veränderte Stoff-, Energie- und Datenflüsse. Im **dritten Schritt** wird der Aufwand für die Umsetzung der technologiebedingten Änderungen durch die Berechnung des sogenannten TIE-Werts (Technology Infusion Effort) quantifiziert. Hierzu werden die identifizierten Änderungen (Einträge innerhalb der Δ DSM) summiert und durch die Anzahl bestehender Verbindungen und Elemente in der Baseline-DSM dividiert. Der TIE-Wert drückt somit den relativen Änderungsbedarf des Ausgangssystems im Zuge der Implementierung einer neuen Technologie aus. Dies ermöglicht den Vergleich der Änderungsauswirkungen alternativer Technologien und damit unterschiedlicher Änderungsanforderungen [SFM+09].

Bewertung: Ähnlich wie die CPM-Methode baut auch der Technology-Infusion Ansatz nach SUH ET AL. auf einer matrixbasierten Systembeschreibung auf. Für jede Änderung werden neben den betroffenen Systemelementen auch die Art und Anzahl betroffener Schnittstellen ermittelt. Die darauf basierende Berechnung des TIE-Werts erlaubt einen relativen Vergleich von Änderungen im Hinblick auf die Änderungsauswirkungen und damit ebenfalls Rückschlüsse auf den Änderungsaufwand.

3.2.3 Modellbasierter Ansatz zur Analyse von Änderungsauswirkungen nach NONSIRI ET AL.

Einen weiteren Ansatz zur Analyse von Änderungsauswirkungen präsentieren NONSIRI ET AL. [NCB+13]. Aufbauend auf der Modellierungssprache SysML und einer Design Structure Matrix (DSM) unterstützt der Ansatz die Analyse von Anforderungsänderungen und die Identifikation von Änderungsförpflanzungen in den frühen Phasen der Systementwicklung. Ergebnis ist ein quantitativer Wert, der den Vergleich von Änderungsaufwänden ermöglicht. Das Vorgehen gliedert sich in vier Phasen, die in Bild 3-9 mittels eines Phasen-Meilenstein-Diagramms beschrieben sind.

In der **ersten Phase** werden die Anforderungen aus vorhandenen Dokumenten in die Modellierungssprache SysML übertragen, hierarchisch gegliedert und klassifiziert. Das Ergebnis in dieser Phase ist ein Anforderungsdiagramm, das in der **zweiten Phase** in eine DSM übertragen wird. Durch die hierarchische Gliederung im Anforderungsdiagramm ist es möglich, Einflüsse zwischen den einzelnen Anforderungen festzuhalten. Ergebnis ist eine DSM nullter Ordnung. Um potentielle Fortpflanzungen geänderter Anforderungen zu identifizieren, werden in der **dritten Phase** übergeordnete Matrizen aufgestellt, die auch indirekte Abhängigkeiten zwischen den einzelnen Anforderungen berücksichtigen. So werden Einflüsse erster Ordnung rot, Einflüsse zweiter Ordnung gelb und Einflüsse dritter Ordnung grün dargestellt. In der **vierten Phase** wird ein quantitativer Wert – die sogenannte Total Change Number – berechnet, der eine Aussage über die Änderungsrate einer Anforderungsänderung bezogen auf das Gesamtsystem trifft. Auf Basis der gewonnenen Erkenntnisse lassen sich Aussagen über die Implementierungsaufwände

sowie über die Wiederverwendung von Anforderungen und den zugehörigen Teilsystemen oder Komponenten treffen [NCB+13].

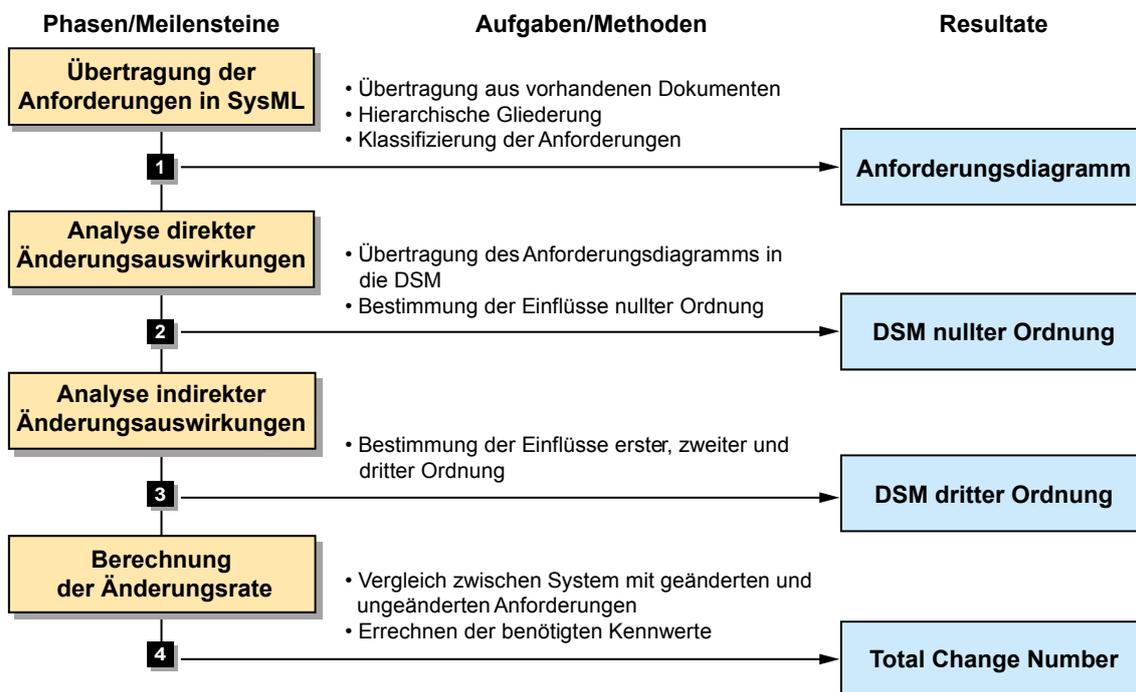


Bild 3-9: Vorgehensmodell zur Auswirkungsanalyse nach NONSIRI ET AL. [NCB+13]

Bewertung: Der Ansatz nach NONSIRI ET AL. kombiniert die modellbasierte Systembeschreibung mittels SysML mit einem matrixbasierten Anforderungsmodell. Die systematische Erhebung von direkten und indirekten Anforderungsabhängigkeiten mittels der DSM erlaubt Rückschlüsse auf die Fortpflanzung von Anforderungsänderungen. Dabei stellt der Ansatz jedoch keinen Bezug zu den betroffenen Systemelementen und damit zur Systemstruktur her. Das für die Release-Planung erforderliche Systemverständnis über die konkrete Änderungsauswirkung wird nicht ausreichend unterstützt.

3.2.4 Analyse von Änderungen mit SysML4Mechatronics

Einen weiteren Ansatz zur Änderungsanalyse präsentieren KERNSCHMIDT und VOGELHEUSER. Der Ansatz basiert auf der Methode SysML4Mechatronics, die das „Port Concept“ der aktuellen SysML Version 1.3 aufgreift. Ziel ist die Darstellung und Analyse von Änderungseinflüssen auf Produktionssysteme. Bild 3-10 beschreibt beispielhaft die Visualisierung von Änderungsauswirkungen im Blockdefinitionsdiagramm basierend auf SysML4Mechatronics [KH13].

In der Methode wird prinzipiell zwischen den drei Sichten Mechanik, Elektronik (E/E) und Software unterschieden. Das Blockdefinitionsdiagramm beschreibt die Zuordnung der Systemelemente zu den drei Sichten und die Schnittstellen zwischen den Systemelementen. Bei den Schnittstellen handelt es sich entweder um Stoff-, Energie- oder Informationsflüsse. Die Flüsse werden über eine Verbindungslinie visualisiert, die als „Proxy

Port“ bezeichnet und entsprechend ihrer übermittelten Flüsse benannt wird. Kompatibilitätsregeln werden für die einzelnen Ports definiert und in Form einer Tabelle festgehalten. Wird nun ein Systemelement verändert, wird dies farblich im Diagramm markiert. Im Beispiel unterliegt der Temperatursensor einer Änderung. Jede Schnittstelle mit diesem Systemelement muss nun über die Kompatibilitätsregeln der Tabelle analysiert werden. Dabei werden zwei Fälle der Änderungsauswirkung unterschieden: 1) Bestehen keine Inkompatibilitäten zwischen verändertem Systemelement und bestehender Schnittstelle, bedarf es keiner Änderung. Die Schnittstelle bleibt bestehen und der Port wird mit einem grünen Haken versehen. 2) Ergeben sich Inkompatibilitäten, wird die entsprechende Schnittstelle im Diagramm durch ein rotes Kreuz gekennzeichnet. An dieser Stelle müssen Überlegungen angestellt werden, in welcher Form die Schnittstelle angepasst werden kann bzw. ob ein Überdenken des veränderten Systemelements möglich ist [KH13].

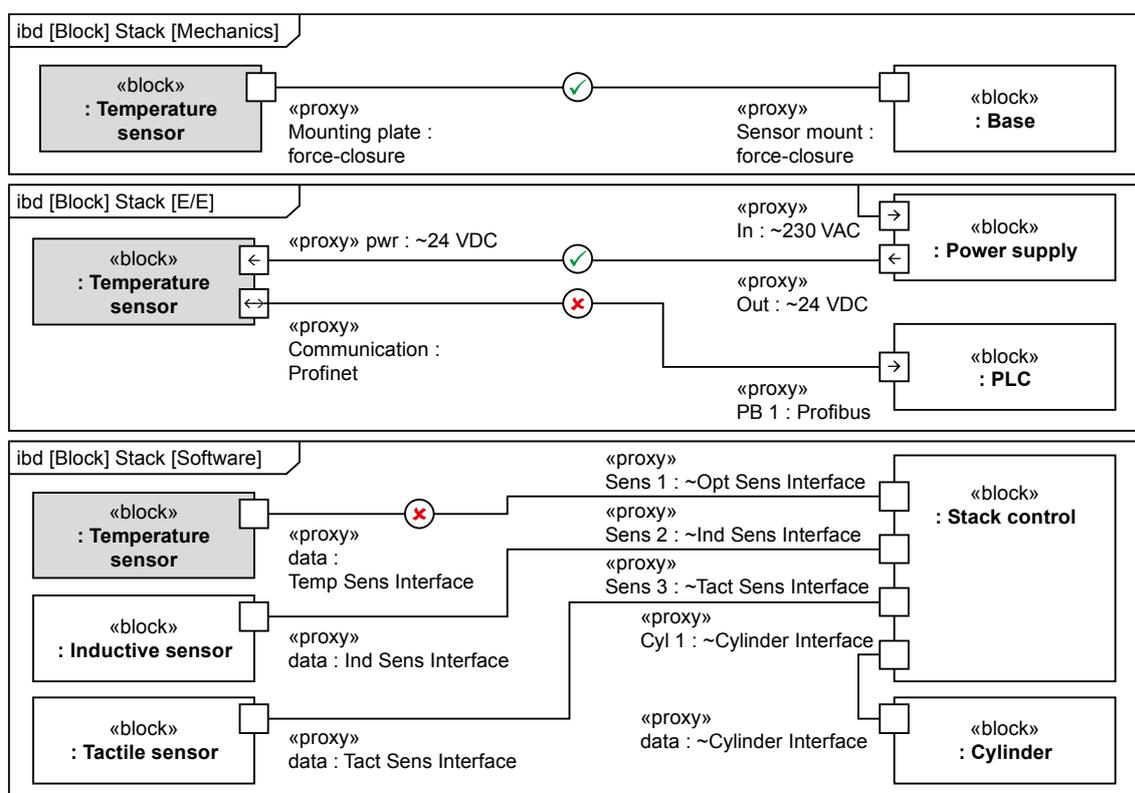


Bild 3-10: Visualisierung von Änderungsauswirkungen im Blockdefinitionsdiagramm nach [KH13]

Bewertung: Genau wie NONSIRI ET AL. greifen auch KERNSCHMIDT und VOGEL-HEUSER auf die SysML zur Systembeschreibung zurück. Dabei erfolgt die Änderungsanalyse allerdings basierend auf dem Blockdefinitionsdiagramm und damit im direkten Bezug zur Systemstruktur des betrachteten Produktionssystems. Der Ansatz unterstützt so die systematische und interdisziplinäre Analyse von Änderungen sowie deren Auswirkungen auf relevante Systemelemente und Schnittstellen.

3.3 Ansätze zur Klassifizierung und Priorisierung von Änderungen

Ausgehend von der Änderungsanalyse erfolgt im typischen Änderungsprozess die Änderungsgenehmigung mit der anschließenden Änderungsumsetzung. Für die Änderungsgenehmigung und die Frage nach dem richtigen Umsetzungszeitpunkt ist eine Klassifizierung und Priorisierung der Änderungen erforderlich. Die Klassifizierung schafft trotz hoher Heterogenität bei Änderungen eine Vergleichbarkeit, die es ermöglicht, konkrete Maßnahmen für die Abwicklung der jeweiligen Änderung abzuleiten. Die Priorisierung erlaubt das Abwägen einer Umsetzungsrangfolge unter Berücksichtigung unterschiedlicher Stakeholder-Präferenzen. Die für die Klassifizierung und Priorisierung erforderliche Änderungsbewertung basiert auf Merkmalen und Ausprägungen von Änderungen, die es im Rahmen des Änderungsmanagements bzw. der Release-Planung zu definieren gilt. Typische Merkmale sind der Änderungsaufwand und der Gegenstand bzw. die Art der Änderung [Bel08], [Ehr09]. Im Folgenden werden ausgewählte Ansätze zur Klassifizierung und Priorisierung von Änderungen vorgestellt.

3.3.1 Kano-Modell zur Klassifizierung von Änderungen

Das Kano-Modell dient der Strukturierung und Priorisierung von Kundenanforderungen durch deren Klassifizierung in Basis-, Leistungs- und Begeisterungsanforderungen. Typischerweise wird das Kano-Modell in der frühen Phase der Produktdefinition zur Planung der Funktionen/Eigenschaften eines Produkts eingesetzt [Lin05, S. 244]. Es kann aber ebenso zur Bewertung und Planung von Änderungsanforderungen in späteren Phasen des Produktlebenszyklus und somit in der Release-Planung eingesetzt werden. Ein klassisches Kano-Modell ist in Bild 3-11 exemplarisch dargestellt.

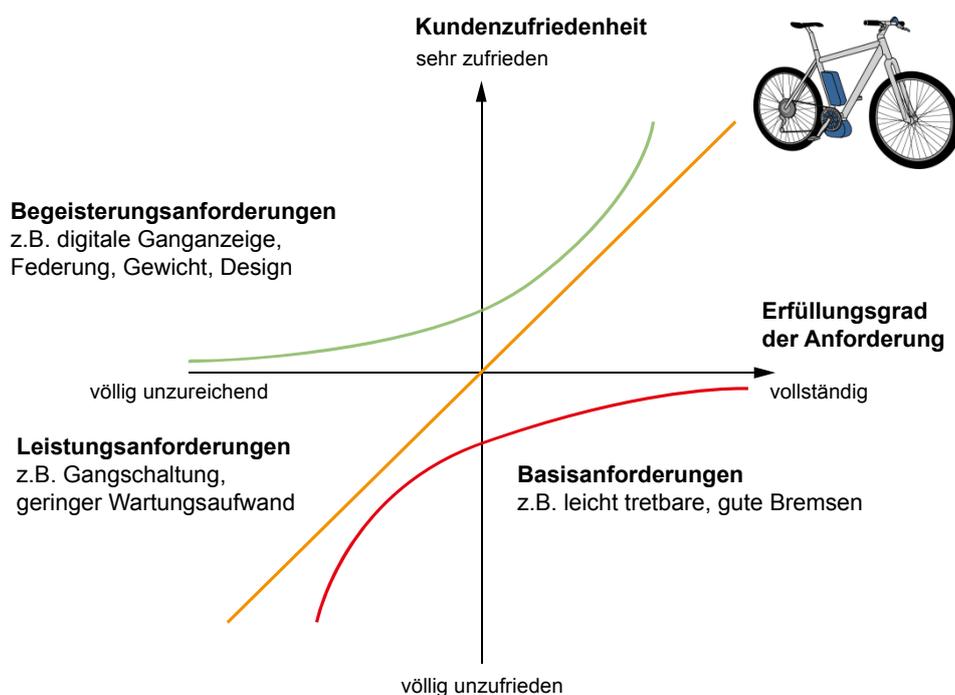


Bild 3-11: Kano-Modell am Beispiel eines Fahrrads [Lin05, S. 94]

Entsprechend des Modells werden die Anforderungen und damit auch Änderungsanforderungen eines Produkts in drei Klassen gegliedert: Basisanforderungen, Leistungsanforderungen und Begeisterungsanforderungen. Für die Erläuterung dieser Klassen bietet sich das Beispiel eines Fahrrads an. Die Erfüllung der **Basisanforderungen** (z.B. gute Bremsen) setzt der Kunde als selbstverständlich voraus. Werden diese Anforderungen nur unzureichend erfüllt, führt dies zur Unzufriedenheit des Kunden. **Leistungsanforderungen** (z.B. die Anzahl der Gänge) dienen dem Kunden zum Vergleich alternativer Angebote. Sie sind unter Berücksichtigung eines ausgewogenen Kosten-Nutzen-Verhältnisses zu definieren. Den größten Einfluss auf die Kundenzufriedenheit haben **Begeisterungsanforderungen** (z.B. eine digitale Ganganzeige). Diese werden vom Kunden eigentlich nicht erwartet, so dass sich die Erfüllung von Begeisterungsanforderungen überproportional positiv auf die Kundenzufriedenheit auswirkt. Hiermit lassen sich zumindest zeitlich befristet Wettbewerbsvorteile erzielen. Die zeitliche Befristung ergibt sich aus dem stetigen Wandel der Kundenbedürfnisse: Begeisterungsanforderungen entwickeln sich mit der Zeit zu Leistungsanforderungen und werden später als Basisanforderungen vom Kunden vorausgesetzt [Lin05, S. 244], [Kam12, S. 64].

Bewertung: Das Kano-Modell lässt sich im Rahmen der Release-Planung zur Klassifizierung kundenrelevanter Änderungen anwenden. Änderungen können hinsichtlich ihres potentiellen Innovationsbeitrags unterschieden werden, womit der Anforderung nach einer gesteuerten Produktwertsteigerung nachgekommen werden kann.

3.3.2 Klassifizierung von Änderungen nach ABMANN

Während das Kano-Modell einer markt- und kundenorientierten Klassifizierung von Änderungen dient, zielt der Ansatz nach ABMANN auf die Klassifizierung von Änderungstypen ab. Ziel ist es, durch die Zuordnung einer Änderung zu einem Änderungstyp eine differenzierte Abwicklung von Änderungen mittels vordefinierter Prozesse zu ermöglichen. Dazu werden die änderungsspezifischen Informationen (Problem, Lösung, mögliche Auswirkungen) mittels Klassifizierung zu einer Bewertungs- und Entscheidungsgrundlage verdichtet und abstrahiert. Typische Kriterien zur Klassifizierung sind die Änderungsdauer, Kosten, Komplexität, Neuigkeit und Nutzen. In Abhängigkeit vom ermittelten Änderungstyp werden anschließend vordefinierte Methoden und Freigabeprozesse zur Änderungsbearbeitung durchlaufen [Ass00, S. 92ff.]. Als wesentliches Hilfsmittel dient eine Checkliste zur Klassifizierung von Änderungen, die auf Arbeiten von EVERSHEIM ET AL. aufbaut [EWS97] (Bild 3-12).

Die in der Checkliste enthaltenen Kriterien (z.B. Problemteil, Dauer) sind unternehmensindividuell anzupassen. Dies sollte auf Basis einer Analyse von bereits abgewickelten Änderungen im Unternehmen erfolgen. Im Beispiel handelt es sich um eine Checkliste, die auf einen Automobilzulieferer ausgerichtet ist. Die Bewertung der Änderung entlang der einzelnen Kriterien und der zugehörigen Ausprägungen liefert ein Änderungsprofil, das einen Rückschluss auf den jeweiligen Änderungstyp zulässt. Wie in

Tabelle 3-1 dargestellt, sind diesen Änderungstypen schließlich konkrete Maßnahmen für die Änderungsbearbeitung zugewiesen [Ass00, S. 109ff.].

Checkliste zur Klassifikation von Änderungen		Ersteller:	Datum:
		G. Assmann	13.11.1999
Änderungsnummer: AE-37	Betroffenes Bauteil: Dichtung		Projekt: EA 47 - 11
Termin: 17.03.2000	Betroffene Lieferanten: Fa. Dichtungen Plum		Projektende: 31.07.2001
Änderungstyp	A	B	C
Beispiel	Neue Anforderung Funktionsausfall	Kostenreduzierung Funktionsstörung	Zeichnungskorrektur
Problemteil	ja	nein	unbedeutend
Dauer	Größer 4 Monate	1 bis 4 Monate	Kleiner 1 Monat
Kapazität	Größer 1 Personenmonat	Bis 1 Personenmonat	Kleiner 1 Personenwoche
Kosten	Veränderung des Teilepreises > 2,50	Veränderung des Teilepreises < 2,50	Keine Veränderung des Teilepreises
Komplexität	Mehr als 5 Bauteile	2 bis 5 Bauteile	1 oder 2 Bauteile
Neuigkeit	Ziel und Lösungsweg unbekannt	Ziel oder Lösungsweg unbekannt	Ziel und Lösungsweg bekannt
Technisches Risiko	Know-how nicht vorhanden, Versuche nötig	Kompetenz vorhanden, Versuche nötig	Kompetenz vorhanden, keine Versuche nötig
Wirtschaftl. Risiko	Hoch, z.B. kurze Projektlaufzeit	Mittel, z.B. lange Projektlaufzeit	Gering, z.B. geringe Kosten
Nutzen	Hoch	Mittel	Gering
Auswahl	A	B	C
Bemerkung/ Begründung	Die Änderung kann trotz hoher Dauer und hohem Kapazitätsbedarf dem Typ B zugeordnet werden, da die Komplexität gering und kein Problemteil betroffen ist.		

Bild 3-12: Checkliste zur Klassifizierung von Änderungen nach ABMANN [Ass00, S. 101]

Tabelle 3-1: Änderungstypen und entsprechende Maßnahmen [Ass00, S. 110]

Typ	Beschreibung	Maßnahme
A	Problem u. Lösung nicht bekannt, zeit- u. kostenintensive Änderung (z.B. neue Anforderung)	Problemlösung und Bewertung im Team
B	Problem und/oder Lösung teilweise bekannt, geringes wirtschaftliches und/oder technisches Risiko	Problemlösung durch Spezialisten, Bewertung im Team
C	Problem und Lösung vollständig bekannt, kein technisches Risiko, geringer Aufwand	Problemlösung und Bewertung durch Spezialisten

Bewertung: Der Ansatz zur Klassifizierung von Änderungen nach ABMANN zielt auf die differenzierte Bearbeitung von Änderungen ab. Trotz der Heterogenität von Änderungen werden diese durch die Klassifizierung standardisierten Bearbeitungsprozessen zugewiesen. Die standardisierte Abwicklung von Änderungen entspricht dem Grundgedanken der

Release-Planung, so dass sich die Klassifizierung nach Änderungstypen als geeignetes Hilfsmittel in der Release-Planung herausstellt. Hierfür ist jedoch eine Anpassung der Checkliste im Sinne eines speziellen Klassifikationsschemas für die Release-Planung erforderlich.

3.3.3 Klassifizierung von Änderungen im Rahmen der Produktgenerationsentwicklung nach ALBERS ET AL.

ALBERS ET AL. stellen ein Beschreibungsmodell für die Produktgenerationsentwicklung vor, das Entwicklungsprojekte anhand ihres Änderungsgrads charakterisiert. Sie definieren die Produktgenerationsentwicklung als die Entwicklung einer neuen Generation technischer Produkte, *die sowohl durch die Anpassung von Teilsystemen als Übernahmevariation als auch durch eine signifikante Neuentwicklung von Teilsystemen charakterisiert ist* [ABW15, S. 1]. Die Charakterisierung soll es ermöglichen, einen projektspezifischen Entwicklungsprozess in Abhängigkeit vom Änderungsgrad bedarfsgerecht zu planen und umzusetzen. Dem Beschreibungsmodell liegt die Annahme zu Grunde, dass neue Produkte häufig auf Vorgänger- oder Wettbewerbsprodukten basieren, die die grundsätzliche Struktur eines Systems vorgeben. Neuentwicklungsprojekte sind dann häufig nur Resultat der Änderung bzw. Variation einzelner Teilsysteme. Grundlage des Beschreibungsmodells ist die Formalisierung und Klassifizierung des Änderungsgrads mittels mathematischer Modelle [ABW15, S. 1ff.]. Wie in Bild 3-13 dargestellt, werden zur Klassifizierung einzelner Teilsysteme drei Änderungstypen unterschieden.

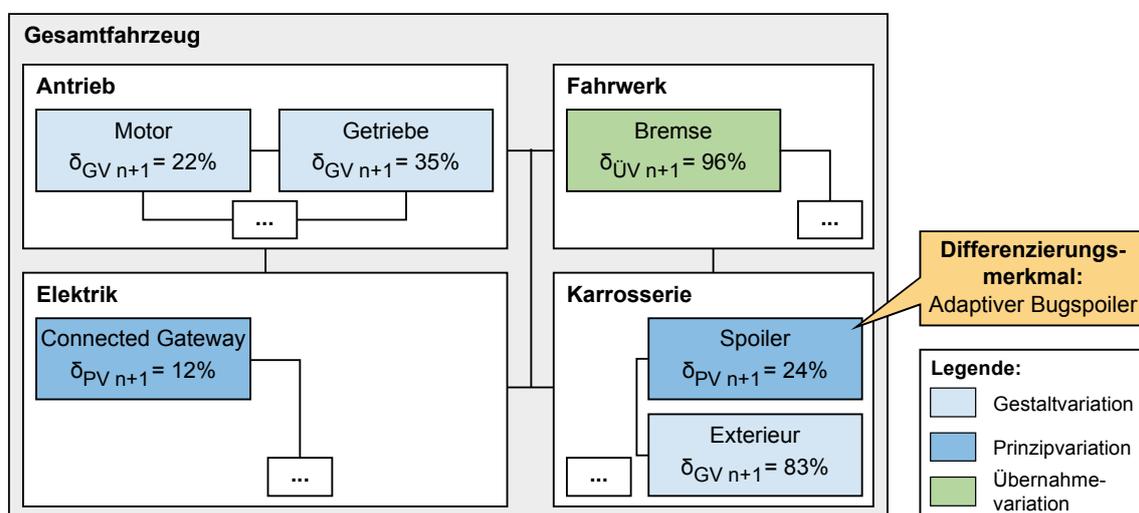


Bild 3-13: Unterscheidung von Gestalt-, Prinzip- und Übernahmevariation am Beispiel eines Fahrzeugs [ABW15, S. 7]

- **Gestaltvariation:** Hierbei handelt es sich um Neuentwicklungen, bei denen ein bekanntes Lösungsprinzip übernommen wird, z.B. aus einem Vorgängerprodukt. Die Variation entsteht dann durch Anpassung der funktionsbestimmenden Eigenschaften, z.B. der Leistungsfähigkeit oder Qualität. Als Beispiel führen die Autoren die

Steigerung der Leistungsdichte von Zahnradgetrieben durch Optimierung von Flankegeometrie, Werkstoff, Fertigungsverfahren und Schmierung an [ABW15, S. 4f.].

- **Prinzipvariation:** Diese beschreibt Neuentwicklungen von Teilsystemen durch alternative Lösungsprinzipien. Dazu werden Funktionen oder Eigenschaften von Produkten aus einem anderen Kontext adaptiert. Auch die systematische Anwendung alternativer Lösungsprinzipien ausgehend von Konstruktionskatalogen und Kreativitätstechniken fällt unter diesen Änderungstyp [ABW15, S. 4].
- **Übernahmevariation:** In diesem Fall werden bestehende Lösungen von Vorgängerprodukten übernommen. Nur die Schnittstellen werden angepasst, um die Systemintegration entsprechend der Anforderungen des neuen Systems zu ermöglichen.

Ausgehend von diesen Änderungsklassen wird der Variationsanteil einzelner Systeme bzw. Teilsysteme mittels eines mathematischen Modells bestimmt. Danach ergibt sich ein Entwicklungsprojekt (G_{n+1}) aus der Menge der Teilsysteme, die eine Übernahmevariation (ÜS), Gestaltvariation (GS) oder Prinzipvariation (PS) beinhalten. Damit gilt:

$$G_{n+1} = \text{ÜS}_{n+1} \cup \text{GS}_{n+1} \cup \text{PS}_{n+1}$$

Gleichung 3-1: Summe der Variationen [ABW15, S. 5]

Der Variationsanteil (in Prozent) für Prinzip-, Gestalt- und Übernahmevariationen eines Entwicklungsprojekts kann damit definiert werden als:

$$\delta_{\text{ÜV } n+1} = \frac{|\text{ÜS}_{n+1}|}{|G_{n+1}|} \quad \delta_{\text{GV } n+1} = \frac{|\text{GS}_{n+1}|}{|G_{n+1}|} \quad \delta_{\text{PV } n+1} = \frac{|\text{PS}_{n+1}|}{|G_{n+1}|}$$

Gleichung 3-2: Gleichungen zur Ermittlung des Übernahme-, Gestalt- und Prinzipvariationsanteils [ABW15, S. 5]

Die Bestimmung des Variationsanteils eines Teilsystems ermöglicht schließlich die Abschätzung von Risiko, Aufwand und Kosten des jeweiligen Entwicklungsvorhabens. Nach ALBERS ET AL. ist dabei festzuhalten, dass sich die Anteile von Prinzip- und Gestaltvariationen eines Entwicklungsvorhabens insb. auf kundenerlebbar Teilsysteme beziehen sollten, um Begeisterungs- und Leistungsanforderungen gerecht zu werden. Die für den Kunden nicht direkt erlebbaren Teilsysteme sollten hingegen im Sinne einer Übernahmevariation lediglich übernommen und angepasst werden [ABW15, S. 6ff.].

Bewertung: Ähnlich wie der Ansatz nach ABMANN zielt auch dieser Ansatz auf die Klassifizierung von Änderungstypen ab, um Rückschlüsse auf den passenden Entwicklungsprozess zu ziehen. Hierbei sticht insb. die differenzierte und quantifizierte Unterscheidung verschiedener Variationsanteile bei der Änderung des technischen Systems heraus. Die resultierende Änderungsklassifikation kann im Rahmen der Release-Planung als Hilfsmittel zur gezielten Zuordnung von Änderungen zu Releases genutzt werden. Offen lässt der Ansatz jedoch sowohl Methode als auch Beschreibungsmittel zur Auswirkungsanalyse von Änderungen.

3.3.4 Priorisierungstechniken nach BERANDER und ANDREWS

Ein wichtiger Faktor im Rahmen der Release-Planung ist die Entscheidung zwischen Alternativen. Aus Ressourcengründen können häufig nicht alle Änderungen wie gewünscht umgesetzt werden. In diesem Fall ist jedoch nicht immer eindeutig, welche der Umsetzungsalternativen die bessere ist, da verschiedene Aspekte wie Kundennutzen, Realisierungsaufwand und Risiko gleichermaßen berücksichtigt werden müssen. Hinzu kommen unterschiedliche Stakeholder-Präferenzen, bei denen bspw. die technische Perspektive der Entwickler entgegen der marktorientierten Perspektive des Produktmanagements steht. Für eine fundierte Entscheidung ist in diesem Fall eine systematische Priorisierung zwischen den Änderungen erforderlich, die die Perspektive unterschiedlicher Stakeholder berücksichtigt. Einen umfassenden Überblick über bestehende Techniken zur Priorisierung liefern BERANDER und ANDREWS [BA05, S. 69ff.]. Eine Auswahl der hier beschriebenen Ansätze⁴⁷ ist in Tabelle 3-2 dargestellt.

Tabelle 3-2: *Priorisierungstechniken und ihre Eigenschaften nach [BA05, S. 78]*

Priorisierungstechnik	Maßstab	Granularität	Aufwand
Analytic Hierarchy Process	Verhältniszahl	Fein	Sehr hoch
100 Dollar Test	Verhältniszahl	Fein	Hoch
Rangfolge	Ordinalzahl	Mittel	Mittel
Top-ten	Ordinalzahl	Grob	Gering

Grundsätzlich haben sämtliche Priorisierungstechniken den gleichen Zweck: Sie unterstützen die Entscheidungsfindung bei einer Menge an Alternativen. Sie unterscheiden sich jedoch im resultierenden Maßstab, in der Granularität und im Aufwand für die Durchführung. So handelt es sich beim Analytic Hierarchy Process (AHP) bspw. um eine aufwändige Priorisierungstechnik, die Alternativen sehr feingranular ins Verhältnis setzt. Die Priorisierungstechnik Top-ten unterstützt die Entscheidungsfindung hingegen pragmatisch mit geringem Aufwand, liefert jedoch ein wenig differenziertes Ergebnis und ist gerade bei komplexen Entscheidungsproblemen ungenau. Der Einsatz der passenden Priorisierungstechnik hängt daher von der Komplexität des Entscheidungsproblems und den zur Entscheidungsfindung verfügbaren Ressourcen ab [BA05, S. 69ff.].

Bewertung: Die Priorisierungstechniken nach BERANDER und ANDREWS sind ein probates Mittel zur Unterstützung der Release-Planung. Sie unterstützen die systematische Entscheidungsfindung bei konkurrierenden Änderungen. Auch kann durch die Einbindung verschiedener Stakeholder in den Priorisierungsprozess der Anforderung zur Berücksichtigung unterschiedlicher Stakeholder-Präferenzen im Rahmen der Release-Planung Rechnung getragen werden.

⁴⁷ Eine Kurzbeschreibung der Priorisierungstechniken findet sich in Anhang A2.1.

3.4 Ansätze zur strategischen Produktplanung

Die Release-Planung weist eine enge Schnittstelle zur Produktstrategie und damit zur strategischen Produktplanung auf (vgl. Abschnitt 2.3 und 2.4.1). Im Folgenden werden daher ausgewählte Ansätze der strategischen Produktplanung vorgestellt, die einerseits als angrenzende Systematiken zu verstehen sind, andererseits jedoch auch relevante Lösungsansätze für die zu entwickelnde Systematik liefern. Bei den Ansätzen nach NIPPA und LABRIOLA sowie nach BRINK handelt es sich um zwei roadmapbasierte Ansätze, die markt- und wettbewerbsstrategische Überlegungen mit der Technologieplanung vereinen, um über den Zeitpunkt der Markteinführung neuer Produkte zu entscheiden. Davon ausgehend werden mit den Ansätzen nach PEITZ und HEPPERLE zwei produktlebenszyklusorientierte Ansätze vorgestellt, die unter Berücksichtigung von Informationen über die Entwicklung von Märkten und Technologien die Weiterentwicklung technischer Systeme und zugehöriger Geschäfts- und Dienstleistungsmodelle planen.

3.4.1 Time-to-Market Management nach NIPPA und LABRIOLA

Das Time-to-Market Management (TtMM) nach NIPPA und LABRIOLA dient der Produkt- und Technologieplanung im Unternehmen. Ziel ist es, ausgehend von der integrierten Betrachtung von Market-Pull und Technology-Push bestmögliche Markteinführungszeitpunkte für Innovationen zu bestimmen und die Entwicklungsorganisation darauf auszurichten. Der Ansatz gliedert sich in drei Phasen entlang eines lang-, mittel- und kurzfristigen Planungshorizonts. Alle Phasen werden durch ausgewählte Methoden unterstützt und durch den Ansatz des Roadmappings synchronisiert [NL08, S. 297ff.]. Bild 3-14 zeigt das 3-stufige Vorgehensmodell.

Langfristiges Time-to-Market Management: In dieser Phase kommen technologieorientierte Methoden wie die Technologiefrüherkennung und marktorientierte Ansätze wie die Szenario-Technik zum Einsatz. Ziel ist die Identifikation technologie- und marktinduzierter Innovationsimpulse, bei denen es sich aufgrund des langfristigen Zeithorizonts schwerpunktmäßig um revolutionäre Innovationsströmungen handelt [NL08, S. 305].

Mittelfristiges Time-to-Market Management: In dieser Phase bildet ein integrierter Roadmapping-Ansatz den Kern, der sich in sechs Schritte gliedert (nicht in Bild 3-14 dargestellt). Der erste Schritt grenzt die Betrachtungsobjekte getrennt nach Technologien und Produktkategorien ein. Sowohl für die Technologien als auch für die Produktkategorien werden anschließend Roadmaps erzeugt. Die *Technologie-Roadmap* zeigt zukünftige Entwicklungspfade unter Berücksichtigung von Technologie-Lebenszyklen und Marktentwicklungen auf. Die *Produkt-Roadmap* bringt Verbesserungsinnovationen und revolutionäre Innovationen in einen Zeitverlauf. Die Synchronisation beider Roadmaps führt zur vorläufigen *Innovations-Roadmap*, die die geplanten Markteintrittszeiträume mit den entsprechend erforderlichen Technologieentwicklungsprojekten verknüpft. Die Verknüpfung erlaubt Rückschlüsse auf zu beschleunigende Entwicklungsprojekte bzw. auf voraussichtlich nicht einzuhaltende Markteintrittszeitpunkte. In einem weiteren Schritt wird

die vorläufige Innovations-Roadmap um eine technologie- und innovationsorientierte Wettbewerbsanalyse ergänzt, die in einer *Konkurrenz-Roadmap* mit den potentiellen Markteintrittszeiträumen vergleichbarer Konkurrenzprodukte mündet. Den Abschluss bildet die Überarbeitung der *Innovations-Roadmap*, die dann auf oberer Ebene den mittelfristigen und auf unterer Ebene den kurzfristigen Planungshorizont mit bereits laufenden Entwicklungsprojekten enthält [NL08, S. 317].

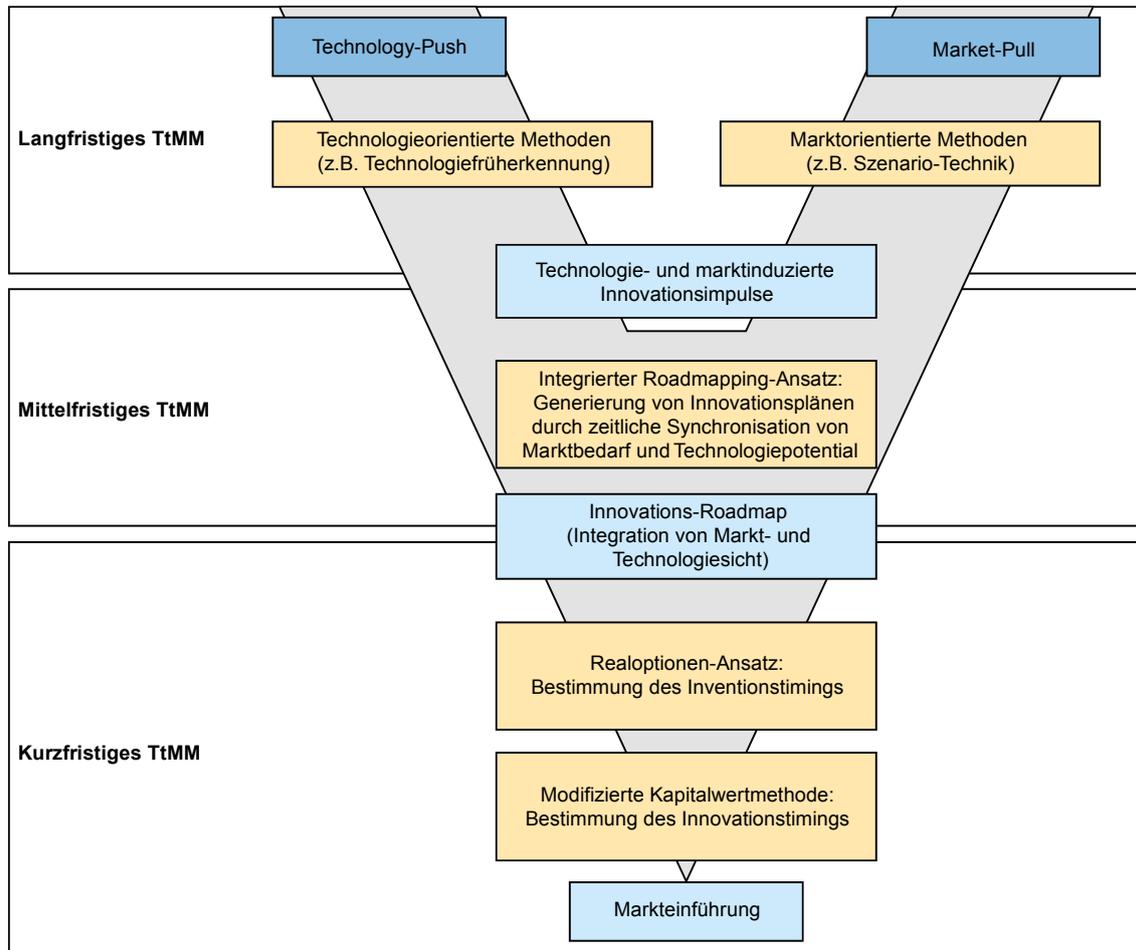


Bild 3-14: Drei Ebenen des Time-to-Market Managements nach NIPPA und LABRIOLA [NL08, S. 304ff.]

Kurzfristiges Time-to-Market Management: Unter Berücksichtigung von Entwicklungsressourcen und -kosten erfolgt in dieser Phase die Konkretisierung einzelner Innovationsprojekte. Im Fokus steht die zeitliche Feinplanung, mit dem Ziel eines optimalen Innovationstimmings und der damit verbundenen Festlegung des Markteinführungszeitpunkts. Hierzu kommt ein Modell zum Einsatz, das an den Realoptionen-Ansatz und die Kapitalwertmethode angelehnt ist [NL08, S. 318ff.].

Bewertung: Der TtMM-Ansatz nach NIPPA und LABRIOLA bietet ein kaskadiertes Vorgehen zur Eingrenzung der Markteinführungszeitpunkte einzelner Innovationen. Das Vorgehen zeichnet sich durch die umfassende Berücksichtigung markt- und technologie-

strategischer Überlegungen aus und stellt so eine mit der Unternehmens- und Produktstrategie abgestimmte Produktevolution sicher. Durch eine umfassende Analyse externer Einflüsse (z.B. Wettbewerb) wird darüber hinaus die Produktwertsteigerung gezielt gesteuert. Der Ansatz ist jedoch sehr allgemein gehalten und bezieht sich eher auf die Planung unternehmensweiter Produktinnovationen und weniger auf die gesteuerte Umsetzung von Änderungen. So werden weder Änderungsauswirkungen berücksichtigt, noch erfolgt eine systematische Bündelung von Änderungen zu Releases.

3.4.2 Entwicklung von Produkt-Technologiestrategien nach BRINK

Einen weiteren roadmapbasierten Ansatz zur Synchronisation von Produkt- und Technologieplanung schlägt BRINK vor. Anders als der Ansatz nach NIPPA und LABRIOLA dient hier eine konkrete Innovationsaufgabe als Ausgangspunkt. Für diese wird im weiteren Verlauf eine konsistente Produkt- und Technologiestrategie erarbeitet. Das Vorgehen ist in Bild 3-15 dargestellt und gliedert sich in sieben Phasen, die im Folgenden vorgestellt werden [Bri10, S. 97 ff.].

Analyse und Prognose des Geschäfts (Phase 1): Hier dienen Geschäftsstrukturanalyse und Kundensegmentierung als Basis für die Prognose des zukünftigen Geschäfts. Die Prognose liefert Anforderungen und Potentiale, die mittels Norm-Produktstrategien und Norm-Technologiestrategien zu Innovationsaufgaben für den jeweiligen Produktbereich konkretisiert werden. Im Ergebnis liegt eine priorisierte und um Anforderungen ergänzte Innovationsaufgabe vor [Bri10, S. 99 ff.].

Entwicklung von kundenspezifischen Anforderungsprofilen (Phase 2): Die Anforderungen der Innovationsaufgabe müssen zielgruppenspezifisch konkretisiert und priorisiert werden. Mittels Kundenbefragung werden dazu in dieser Phase kundensegmentspezifische Anforderungsprofile gebildet [Bri10, S. 117 ff.].

Anforderungsanalyse (Phase 3): Die Analyse der funktionalen Anforderungen resultiert in einer Funktionshierarchie. Orientiert an den zuvor erarbeiteten Anforderungsprofilen werden Nutzegewichte abgeleitet und den Funktionen zugewiesen [Bri10, S. 122 ff.].

Inventur des Technologieportfolios (Phase 4): Ausgangspunkt ist die Dokumentation bestehender, geplanter und beobachteter Technologien mittels Technologiesteckbriefen. Darauf aufbauend erfolgen eine Technologieverträglichkeitsanalyse und die Zuordnung der Technologien zu Standardfunktionen. Die Inventur des Technologieportfolios schafft so Transparenz über die derzeitige Technologieverwendung [Bri10, S. 131].

Bildung von Produkt-Technologie-Clustern (Phase 5): Entlang der zuvor identifizierten Technologien und den zugehörigen Standardfunktionen wird mittels eines morphologischen Kastens die Innovationsaufgabe konkretisiert. Dazu werden softwaregestützt konsistente Produkt-Technologie-Cluster gebildet, die sich zur Umsetzung der Innovationsaufgabe eignen [Bri10, S. 136].

Kosten- und Nutzenbewertung der Produkt- und Technologie-Cluster (Phase 6): Eine Kosten- und Nutzenbewertung ermöglicht die Auswahl des Erfolg versprechendsten Produkt-Technologie-Clusters. Für die enthaltenen Technologien sind anschließend Make-or-buy-Entscheidungen zu treffen [Bri10, S. 146 ff.].

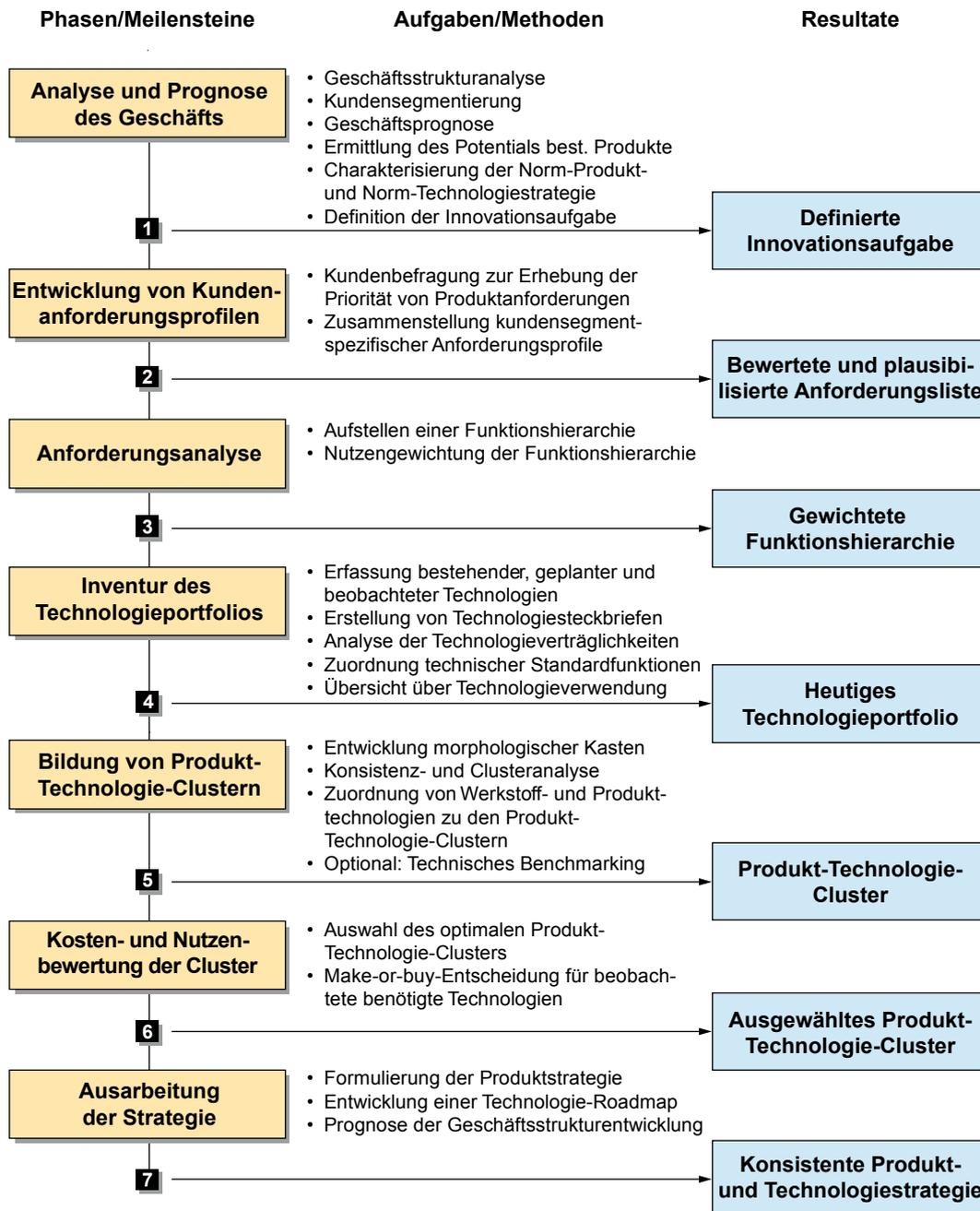


Bild 3-15: Vorgehensmodell zur Entwicklung konsistenter Produkt und Technologiestrategien [Bri10, S. 98]

Integration zur konsistenten Produkt- und Technologiestrategie (Phase 7): In der letzten Phase wird das ausgewählte Produkt-Technologie-Cluster zu einer konsistenten Produkt-Technologiestrategie integriert. Dazu werden Produktstrategie und Technologie-Roadmap konkretisiert. Die Prognose der Geschäftsstrukturentwicklung ermöglicht

schließlich die Definition von Aktivierungsmaßnahmen, die Vorgaben für die Modellpflege über den Produktlebenszyklus machen [Bri10, S. 154 ff.].

Bewertung: Ähnlich wie der TtMM-Ansatz nach NIPPA und LABRIOLA bietet das Verfahren nach BRINK eine umfassende Anleitung zur Abstimmung des Markteinführungzeitpunkts einer Produktinnovation auf die Verfügbarkeit der dafür relevanten Technologien. Somit liefert es einen sehr guten Ansatz zur strategischen Steuerung der Produktwertsteigerung, hat für den Einsatz zur Release-Planung intelligenter technischer Systeme allerdings ein zu hohes Abstraktionsniveau. So liegt der Fokus bei der Entwicklung der Produkt- und Technologiestrategie eher auf vollständigen Neuentwicklungen und weniger auf der abgestimmten Weiterentwicklung bestehender Systeme.

3.4.3 Entwicklung produktlebenszyklusorientierter Geschäftsmodell-Roadmaps nach PEITZ

Die Systematik nach PEITZ widmet sich der Herausforderung der strategiekonformen (Weiter-)Entwicklung von Produkt und zugehörigem Geschäftsmodell entlang des Produktlebenszyklus. Strategiekonformität wird dabei durch die kontinuierliche Synchronisation von Produktstrategie sowie Unternehmens- und Geschäftsstrategie sichergestellt. Grundlage ist eine fundierte Analyse der Ausgangssituation verbunden mit der Antizipation künftiger Markt- und Technologieentwicklungen [Pei15, S. 3]. Das zugehörige Vorgehensmodell ist in sieben Phasen untergliedert (Bild 3-16).

Analyse der Ausgangssituation (Phase 1): In dieser Phase werden der Handlungsrahmen eingegrenzt und erste Handlungsbedarfe identifiziert. Basis ist eine fundierte Analyse der Unternehmens- und Geschäftsstrategie sowie des originären Geschäftsmodells einschließlich des bestehenden Marktleistungskonzepts. Hierbei liegt das Hauptaugenmerk auf der Sammlung von Erfahrungen aus Produktion, Betrieb und Service. Ergänzt werden diese Erkenntnisse um eine Stakeholder-Analyse [Pei15, S. 90ff.].

Markt- und Technologievorausschau (Phase 2): Zur Berücksichtigung künftiger Markt- und Technologieentwicklungen werden in dieser Phase sowohl eine Markt- als auch eine Technologie-Roadmap erarbeitet. Beide Roadmaps dienen zusammen mit den Ergebnissen aus Phase 1 der Formulierung langfristiger Ziele für die Weiterentwicklung des Geschäftsmodells [Pei15, S. 98ff.].

Weiterentwicklung des Geschäftsmodells (Phase 3): In einem kreativen Prozess werden auf Basis der zuvor erarbeiteten Ergebnisse Geschäftsideen generiert und konzeptionell beschrieben. Eine paarweise Bewertung der Geschäftsideen liefert schließlich mittels Triangularisierung eine potentielle Realisierungsabfolge. Diese beschreibt die Evolution des Geschäftsmodells in Form von Geschäftsmodellstufen [Pei15, S. 112ff.].

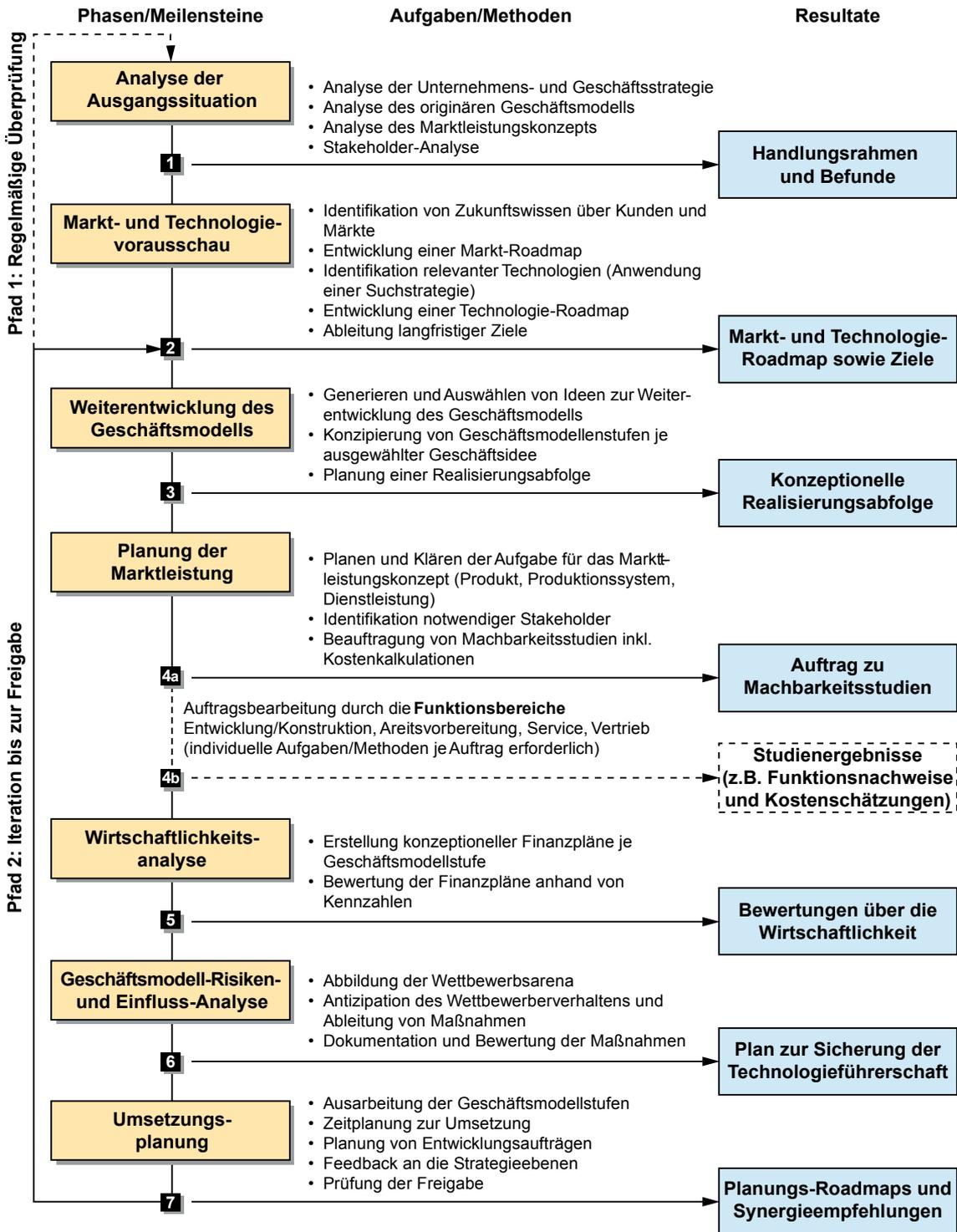


Bild 3-16: Vorgehensmodell zur Entwicklung einer produktlebenszyklusorientierten Geschäftsmodell-Roadmap nach PEITZ [Pei15, S. 88]

Planung der Marktleistung (Phase 4): Mit der Evolution des Geschäftsmodells geht die Weiterentwicklung der damit verbundenen Marktleistung einher. Diese Phase konzentriert sich daher auf Planung und Klärung von Entwicklungsaufgaben für Produkte, Produktionssysteme und Dienstleistungen. In Machbarkeitsstudien wird zudem die technische Umsetzbarkeit dieser Entwicklungsaufgaben überprüft [Pei15, S. 118ff.].

Wirtschaftlichkeitsanalyse (Phase 5): Neben der technischen Machbarkeit ist auch die Wirtschaftlichkeit der Entwicklungsaufgaben zu analysieren. Hierzu werden konzeptionelle Finanzpläne für jede Geschäftsmodellstufe erarbeitet und bewertet. Ergebnis ist deren Freigabe bzw. begründete Ablehnung [Pei15, S. 133ff.].

Geschäftsmodell-Risiken- und Einfluss-Analyse (Phase 6): Für sämtliche Geschäftsmodellstufen gilt es, potentielle Risiken bedingt durch den Wettbewerb zu identifizieren. Ausgehend von einer Analyse der Wettbewerbsarena wird dazu das Wettbewerbsverhalten antizipiert und in vorsorgliche und kausale Reaktionsmuster überführt. Im Ergebnis liegt ein Plan zur Sicherung der Technologieführerschaft vor [Pei15, S. 137].

Umsetzungsplanung (Phase 7): In dieser Phase werden die Geschäftsmodellstufen weiter ausgearbeitet und in eine Geschäftsmodell-Roadmap überführt. Die Geschäftsmodellstufen sind dazu mit der zuvor erarbeiteten Markt- und Technologie-Roadmap zu synchronisieren. Für den Übergang in die Entwicklung sind darüber hinaus Entwicklungsaufträge zu formulieren, die die (Weiter-)Entwicklung der Marktleistung in Abhängigkeit von der Geschäftsmodellstufe beschreiben [Pei15, S. 144ff.].

Bewertung: Die Systematik nach PEITZ ist ein sehr umfassendes Instrument zur lebenszyklusorientierten Geschäftsmodellentwicklung. Das Vorgehen eignet sich hervorragend zur Identifikation strategischer Vorgaben sowie zur Abstimmung der Produktwertsteigerung auf das jeweilige Geschäftsmodell. Das Vorgehen nach PEITZ liefert jedoch eher Input und keine ganzheitliche Lösung für den Kern der Release-Planung. Die Release-Planung ist im Sinne der Produktpflege in der Produktstrategie verankert und hat somit einen viel engeren Fokus. So fehlt es der Systematik nach PEITZ bspw. an einem Vorgehen zur Bündelung und Priorisierung spezifischer Änderungen für die Umsetzung in einem bestimmten Release. Auch die detaillierte Analyse von Änderungsauswirkungen und -abhängigkeiten ist nicht Gegenstand der Systematik.

3.4.4 Planung lebenszyklusgerechter Leistungsbündel nach HEPERLE

Der Ansatz nach HEPERLE zielt auf die Schaffung einer fundierten Entscheidungsgrundlage für die Planung lebenszyklusgerechter Leistungsbündel⁴⁸ ab. Er ist in sechs Phasen strukturiert, die kontextbezogen auf einen Modellraum planungsrelevanter Informationen

⁴⁸ Leistungsbündel werden z.B. auch als hybride Produkte und Produkt-Service-Systeme (PPS) bezeichnet. Sie sind gekennzeichnet durch die [...] Planung, Entwicklung, Implementierung, Erbringung und Nutzung von Sach- und Dienstleistungsanteilen [...] [MU12, S. 1ff.].

zugreifen und diesen mit neuen Ergebnissen anreichern. Der Modellraum wird in temporale und strukturelle Modelle unterschieden. Die temporalen Modelle werden in den ersten drei Phasen der Akquisition befüllt und beinhalten die zeitliche Abfolge von Informationen in Form von Lebenszyklusmodellen und Roadmapping-Ansätzen. Die strukturellen Modelle dienen in den darauffolgenden Phasen – der Analyse – der matrixbasierten Abbildung der Systemzusammenhänge [Hep13, S. 109]. Bild 3-17 gibt einen Überblick über den Ansatz.

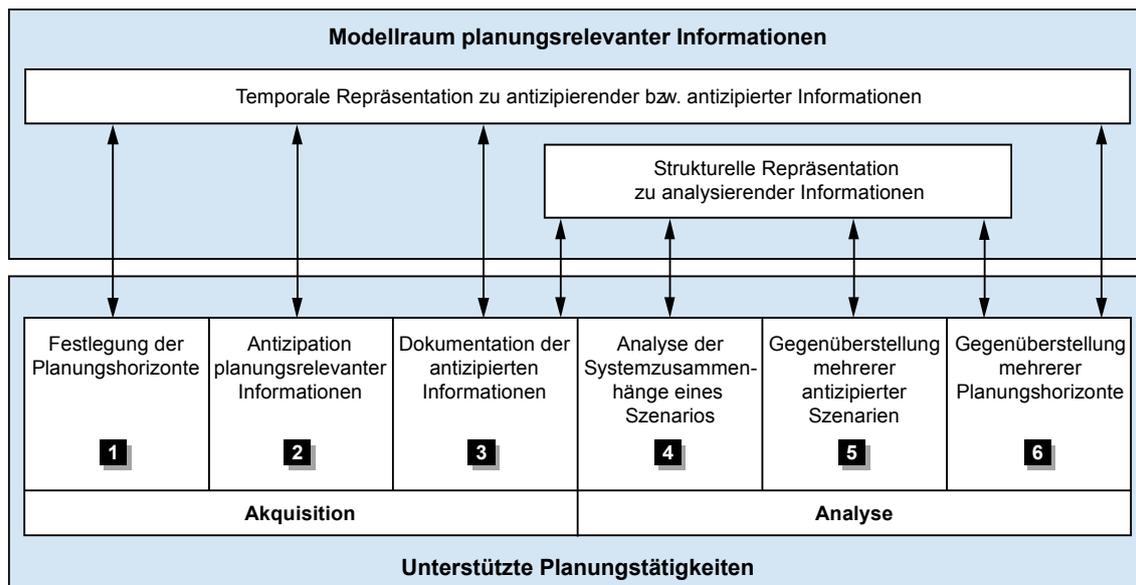


Bild 3-17: Unterstützende Planungstätigkeiten nach HEPERLE [Hep13, S. 108]

Festlegung der Planungshorizonte (Phase 1): In dieser Phase werden zunächst zeitliche Rahmenbedingungen für die Produktplanung identifiziert (z.B. der angestrebte Markteintrittszeitpunkt), die es ermöglichen, über mehrere Produktgenerationen hinweg Zeiträume abzuleiten (z.B. für den Start der Produktentwicklung). Die Kombination dieser Zeiträume mit einem idealtypischen Lebenszyklusmodell liefert im Ergebnis eine Lebenszyklus-Roadmap, die die planungsrelevanten Zeiträume und damit den Planungshorizont definiert [Hep 13, S. 109 ff.].

Antizipation planungsrelevanter Informationen (Phase 2): Unterstützt wird diese Phase durch ein Modell zur Erfassung von Kontextfaktoren. Hierbei handelt es sich um ein zweidimensionales Ordnungsschema, das die Lebenszyklusphasen verschiedenen Kategorien von Kontextfaktoren gegenüberstellt. Unter Einbezug relevanter Stakeholder dient das Ordnungsschema der Identifikation von Rahmenbedingungen und Potentialen, welche die Weiterentwicklung des Leistungsbündels beeinflussen [Hep13, S. 115ff.].

Dokumentation der antizipierten Informationen (Phase 3): In dieser Phase werden die zuvor identifizierten Informationen unter Berücksichtigung alternativer Szenarien strukturiert. Dazu werden sie einerseits unter Einsatz von Roadmapping-Ansätzen in einen zeitlichen Kontext gebracht. Andererseits werden sie in eine strukturelle, matrixbasierte Form überführt, die die spätere Analyse im Sinne des Komplexitätsmanagements

ermöglicht [Hep13, S. 110]. Relevante Informationen sind Ziele und Bedarfe des geplanten Leistungsbündels sowie dafür erforderliche Funktionen, Lösungselemente und Parameter [Hep13, S. 121].

Analyse der Systemzusammenhänge eines Szenarios (Phase 4): Mit Hilfe der zuvor erstellten Strukturmatrizen wird in dieser Phase die Kompatibilität der Leistungsbündelziele und der geplanten Lösungselemente ermittelt. Dies dient der Überprüfung der grundsätzlichen Realisierbarkeit der weiterzuverfolgenden Konzepte. Im Ergebnis liegen Aussagen über mögliche Kombinationen und die Umsetzbarkeit alternativer Leistungsbündel vor [Hep13, S. 110ff.].

Gegenüberstellung antizipierter Szenarien (Phase 5): In dieser Phase sollen Ähnlichkeiten in den zuvor ermittelten Szenarien überprüft werden. Die Basis dafür bilden erneut strukturbasierte Modelle, die bei richtiger Anwendung Aufschluss über vielversprechende Technologien und Lösungselemente versprechen [Hep13, S. 110ff.].

Gegenüberstellung mehrerer Planungshorizonte (Phase 6): Für jedes Szenario werden kurz- und mittelfristige Planungshorizonte gebildet und auf Konsistenz geprüft. Ein Portfolio-Ansatz zeigt die Existenz eines Lösungselements in mehreren Szenarien an und bringt die Relevanz für den mittel- bzw. langfristigen Planungshorizont zum Ausdruck. Dies dient als Grundlage für die anschließende Planung von Produktgenerationen und die damit verbundenen Produkt- und Technologieplanung [Hep13, S. 111ff.].

Bewertung: Der Ansatz nach HEPERLE unterstützt die Planung lebenszyklusgerechter Leistungsbündel im Sinne einer Produktstrategie. Dabei wird ein besonderer Schwerpunkt auf die zeitliche Strukturierung mittels verschiedener Roadmaps und Lebenszykluskonzepte gelegt. Insbesondere die Berücksichtigung verschiedener Kontextfaktoren in Phase 2 stellt eine wichtige Voraussetzung für die initiale Strukturierung des Release-Plans dar. Für die konkrete Umsetzungsplanung von Änderungen und die damit verbundene Änderungsanalyse hat der Ansatz jedoch einen zu hohen Abstraktionsgrad. Ferner ist aufgrund des hohen Aufwands für die Matrixerstellung und -analyse fraglich, ob sich der Ansatz für einen wiederkehrenden Prozess im Sinne der Release-Planung eignet.

3.5 Ansätze zur Produkt-Release-Planung

Das Konzept der Release-Planung entstammt ursprünglich der Softwaretechnik. Einen umfassenden Überblick über die Vielzahl der hier entstandenen Ansätze geben SVAHNBERG ET AL. [SGF+10]. Stellvertretend wird im Rahmen dieser Arbeit der Release-Planungsprozess nach ZORN-PAULI ET AL. vorgestellt. Ausgehend davon werden Ansätze vorgestellt, die das Konzept der Release-Planung auf mechatronische Systeme übertragen. Hier setzen die modulbasierte Release-Planung nach SCHUH ET AL., die Methodik zur zyklengerechten Erstellung von Modul- und Plattformstrategien nach MAURER ET AL. sowie das Release-Management nach BELENER an.

3.5.1 Release-Planungsprozess nach ZORN-PAULI ET AL.

ZORN-PAULI ET AL. beschreiben einen Release-Planungsprozess für Softwaresysteme, der die langfristige Release-Planung von Anforderungen auf unterschiedlichem Abstraktionsniveau ermöglicht [ZPB+13, S. 273]. Der Prozess orientiert sich am evolutionären Problemlösungsansatz EVOLVE II nach RUHE, der inzwischen Gegenstand einer Vielzahl weiterführender Forschungsarbeiten ist [Sal05], [Ruh10] (vgl. Anhang A2). Der Release-Planungsprozess ist in Bild 3-18 dargestellt.

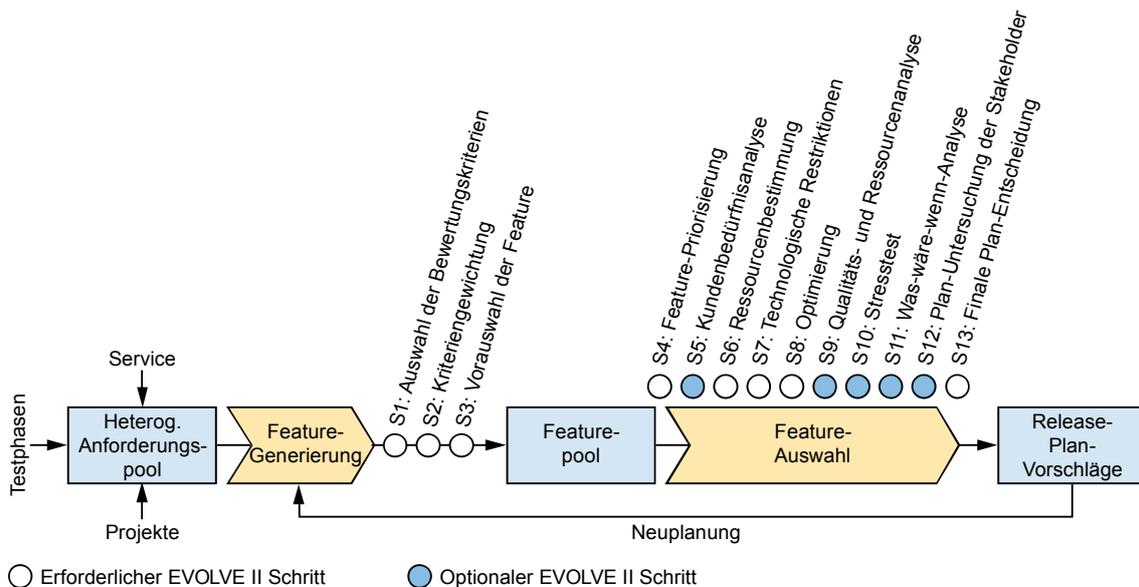


Bild 3-18: Strategischer Release-Planungsprozess nach ZORN-PAULI ET AL. [ZPB+13, S. 278]

Ausgangspunkt bildet ein **heterogener Anforderungspool**, in den kontinuierlich neue Anforderungen und Anforderungsänderungen aus verschiedenen Quellen einfließen (z.B. aus laufenden Projekten, Testphasen und dem Service). Die eingehenden Anforderungen besitzen unterschiedliche Abstraktionsniveaus. Sie betreffen z.B. die Umsetzung übergeordneter Geschäftsstrategien oder spezifische Verbesserungen am existierenden Produkt [ZPB+13, S. 274ff.]. Im Rahmen der **Feature-Generierung** gilt es daher, die heterogenen Anforderungen zu harmonisieren und zu klassifizieren. Dazu wird ein Modell zur Anforderungsabstraktion und Problemlösung als Erweiterung des Anforderungs-Abstraktions-Modells (RAM) nach GORSCHKE ET AL. verwendet [GW06]. Das Modell vereinheitlicht die heterogenen Anforderungen in Form von Features, wodurch die gemeinsame Betrachtung im Release-Planungsprozess ermöglicht wird [ZPB+13, S. 280f.].

Der eigentliche Release-Planungsprozess erfolgt anschließend mittels des Problemlösungsansatzes **EVOLVE II**. In drei vorbereitenden Schritten werden Bewertungskriterien ausgewählt, Bewertungsfaktoren gewichtet und eine Vorauswahl der zu planenden Features getroffen. Ergebnis ist eine **Feature-Pool**, aus der in den darauf folgenden Schritten systematisch die umzusetzenden Features ausgewählt werden (**Feature-Auswahl**). Hierzu wird auf das Softwarewerkzeug ReleasePlanner™ zurückgegriffen, das

unter Berücksichtigung vielfältiger Kriterien (z.B. Wichtigkeit, Risiko) die Features priorisiert und aufeinanderfolgenden Releases zuordnet. Dies erfolgt unter Berücksichtigung von vorhandenen Ressourcen sowie von Abhängigkeiten zwischen den Features. Im Ergebnis liegen alternative Release-Pläne vor (**Release-Plan-Vorschläge**), aus denen der Erfolgversprechendste ausgewählt werden muss. Entscheidungskriterien hierfür sind bspw. die Maximierung des Kundennutzens oder die unternehmensinterne Effizienz. Durch den kontinuierlichen Zustrom unterschiedlicher Anforderungen ist der Release-Planungsprozess in regelmäßigen Abständen zu durchlaufen. So sind bspw. Prioritäten und Aufwände bei Änderungen neu zu bestimmen [ZPB+13, S. 275ff.].

Bewertung: Der auf dem EVOLVE II-Ansatz basierende Release-Planungsprozess nach ZORN-PAULI ET AL. eignet sich aufgrund der Werkzeugunterstützung insb. für evolutionär zu entwickelnde Softwaresysteme mit einem hohen Änderungsaufkommen. Hervorzuheben ist der Rückgriff auf einen aufbereiteten Feature-Pool, in dem sämtliche Änderungen für eine sukzessive Realisierung gesammelt und harmonisiert werden. Das Vorgehen zur Generierung alternativer Release-Pläne stellt darüber hinaus eine harmonisierte Bündelung von Features zu Releases und die Berücksichtigung von Stakeholder-Präferenzen sicher. Der Ansatz gibt jedoch keinerlei Hilfestellung bei der Strukturierung des Release-Plans und der damit verbundenen Festlegung von Release-Zeitpunkten. Darüber hinaus werden zwar Abhängigkeiten zwischen Features berücksichtigt, Hinweise zu deren Identifikation bleiben jedoch aus. Auch eine Untersuchung der durch die Implementierung neuer Features resultierenden Änderungsauswirkungen erfolgt nicht.

3.5.2 Modulbasierte Release-Planung nach SCHUH ET AL.

SCHUH propagiert bereits seit einigen Jahren das Release-Engineering als Ansatz zur Beherrschung der zunehmenden Komplexität technischer Systeme [SE04], [SDL+04], [Sch05, S. 112ff.], [Sch12, S. 12f.] [Sch13, S. 209ff.]. In einer aktuellen Arbeit präsentiert er einen Ansatz zur modulbasierten Release-Planung, der auf die besonderen Herausforderungen komplexer Baukastensysteme ausgelegt ist. Der Ansatz zeichnet sich durch zwei Modelle aus: Im sogenannten *Release-Zyklen-Modell* werden ausgehend von der Flexibilität einzelner Module modulspezifische Release-Zyklen festgelegt. Diese dienen im *Änderungsmodell* als Grundlage für die Umsetzungsplanung technischer Änderungen. Bild 3-19 beschreibt das schrittweise Vorgehen des Ansatzes [SAA13, S. 1606ff.].

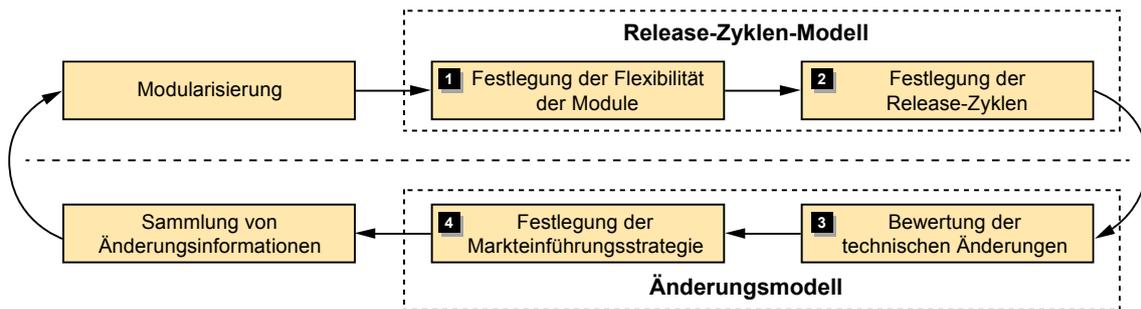


Bild 3-19: Vorgehen bei der modulbasierten Release-Planung für technische Änderungen nach SCHUH ET AL. [SAA13]

Das **Release-Zyklus-Modell** baut auf einer vorgegebenen modularen Produktstruktur auf und gliedert sich in zwei Schritte. Im ersten Schritt wird die Flexibilität der einzelnen Module gegenüber Änderungen bestimmt. Die Flexibilität dient als Indikator für den erforderlichen Aufwand zur Umsetzung von Änderungen an einem Modul und wird anhand der Kriterien Marktwahrnehmung, Modulwechselwirkungen, Komplexitätskosten und Technologieersatz für jedes Modul bestimmt. Ausgehend von einer qualitativen Gesamtbewertung der vier Kriterien werden die Module klassifiziert und in einem Zwiebschalenmodell visualisiert (Bild 3-20) [SAA13, S. 1610].

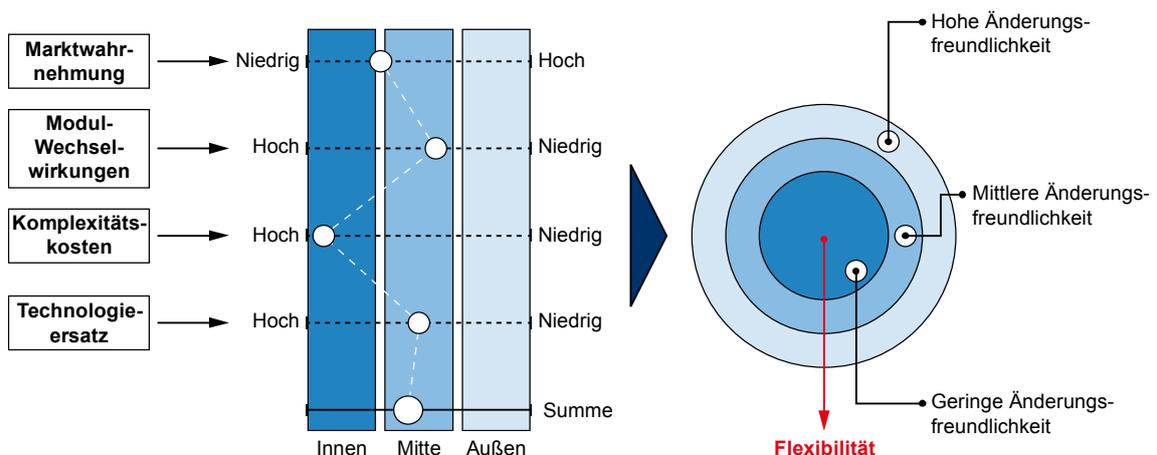


Bild 3-20: Zwiebschalenmodell zur Klassifizierung der Modulflexibilität nach SCHUH ET AL. [SAA13]

Die Module der äußeren Schale weisen eine höhere Flexibilität auf. Sie sind häufiger von Änderungen betroffen, lassen sich entsprechend ihrer Flexibilität aber einfacher überarbeiten. Das resultierende Ergebnis dient als Hilfsmittel für die im zweiten Schritt folgende Ableitung von Release-Zyklen. In Abhängigkeit von der Flexibilität bzw. der Änderungsfreundlichkeit eines Moduls wird der Überarbeitungszyklus für jedes Modul festgelegt. Dabei entspricht die Zyklusdauer immer einem Vielfachen voneinander, so dass flexible Module der äußeren Schale doppelt so häufig überarbeitet werden wie Module aus der mittleren Schale. Entsprechend selten werden Module der inneren Schale überarbeitet [SAA13, S. 1610].

Aufbauend auf den zuvor abgeleiteten Release-Zyklen beschreibt das **Änderungsmodell** ein Vorgehen zur Umsetzungsplanung technischer Änderungen. Das Vorgehen gliedert sich ebenfalls in zwei Schritte: die Bewertung der technischen Änderung sowie die Auswahl der passenden Einführungsstrategie. Bild 3-21 zeigt die Bewertungskriterien sowie die damit verbundenen Einführungsstrategien [SAA13, S. 1611f.].

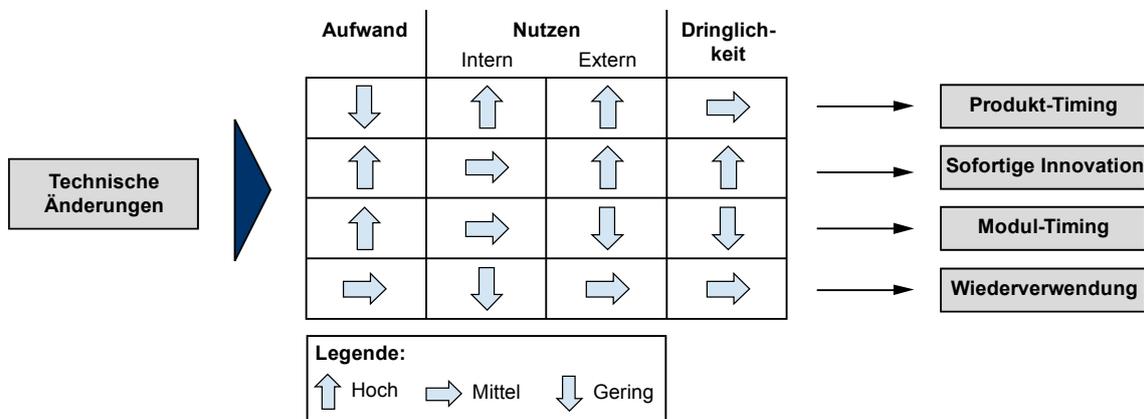


Bild 3-21: Änderungsbewertung zur Auswahl der Einführungsstrategie [SAA13]

Zur Auswahl der richtigen Einführungsstrategie werden technische Änderungen zunächst anhand der Kriterien Aufwand, Nutzen und Dringlichkeit bewertet. Die Gesamtbewertung weist am Ende ein Profil auf, das auf eine von vier Markteinführungsstrategien für ein neues oder angepasstes Modul hindeutet. Änderungen mit einem hohen Nutzen werden im Rahmen des **Produkt-Timings** umgesetzt. In diesem Fall werden die Releases für das gesamte Produktportfolio vorverlegt, um die gleichzeitige Einführung des geänderten Moduls in allen Produktvarianten sicherzustellen. Die **sofortige Innovation** zeichnet sich durch hohe Dringlichkeit aus und integriert das geänderte Modul, ohne die zeitliche Planung der Folge-Releases zu verändern. Weniger dringliche Änderungen werden im Rahmen des **Modul-Timings** erst zum nächstmöglichen Release-Termin umgesetzt. Im Gegensatz dazu erfolgt die Änderung im Rahmen der **Wiederverwendung** sofort, wird aber erst nach und nach in den verschiedenen Produktvarianten umgesetzt [SAA13, S. 1611ff.].

Bewertung: Die modulbasierte Release-Planung für technische Änderungen nach SCHUH ET AL. wurde speziell für Baukastensysteme entwickelt, bei denen gleiche Module in einer Vielzahl von Produktvarianten wiederverwendet werden. Speziell für diese Produktart unterstützt das Release-Zyklen-Modell eine initiale Strukturierung des Release-Plans durch die Festlegung modulspezifischer Release-Zyklen. Die Einführungsstrategien unterstützen darüber hinaus das Innovations-Timing und damit in Ansätzen eine gezielte Produktwertsteigerung. Änderungsauswirkungen und -abhängigkeiten werden hingegen nicht berücksichtigt. Auch Hilfsmittel zur Berücksichtigung von Stakeholder-Präferenzen sowie zur harmonisierten Bündelung von Änderungen fehlen.

3.5.3 Gestaltung und Pflege zyklengerechter Modul- und Plattformstrategien nach MAURER ET AL.

Der Ansatz nach MAURER ET AL. zielt auf die systematische Gestaltung von Plattformsystemen⁴⁹ und die roadmapgestützte Pflege bzw. Weiterentwicklung der darin enthaltenen Module ab. Die Plattformdefinition und die Generierung, Kontrolle und Steuerung von modulspezifischen Entwicklungs-Roadmaps gliedert sich in drei Phasen: Planung der Flexibilität, Operationalisierung und Lebenszyklusmanagement. Dabei berücksichtigt der Ansatz explizit zwei Perspektiven: Aus der Perspektive des Produktmanagements zielt ein *Top-Down-Ansatz* auf die Berücksichtigung und Steuerung der Produkt-Features im Plattformsystem ab. Aus der Perspektive der Entwicklung erfolgt die technische Realisierung des Modul- und Plattformsystems mittels eines *Bottom-Up-Ansatzes* [MBE+14, S. 141f.]. Bild 3-22 verdeutlicht den dreiphasigen Aufbau der Methodik zur Erstellung einer zyklengerechten Plattformstrategie.

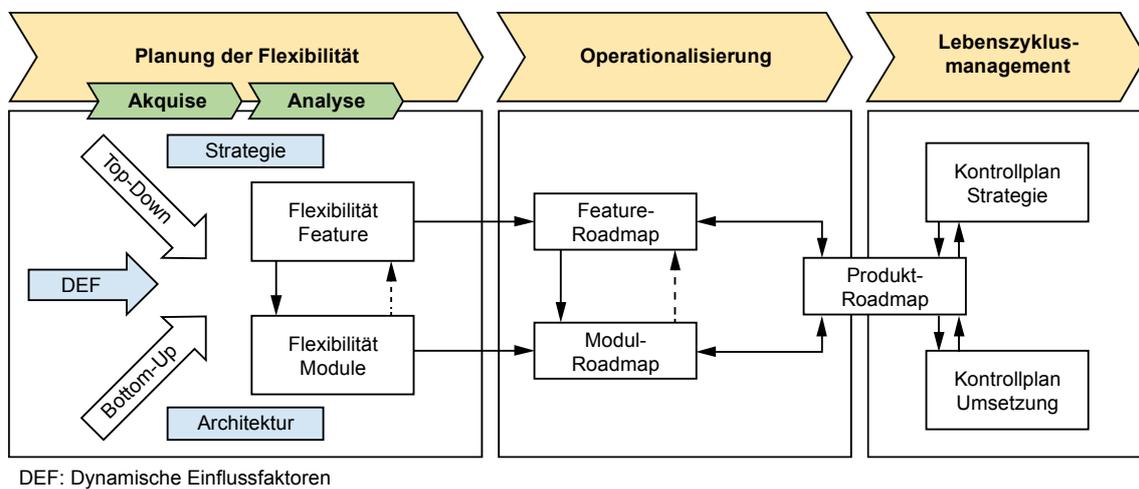


Bild 3-22: Methodik zur Erstellung und Pflege einer zyklengerechten Plattformstrategie nach MAURER ET AL. [MBE+14, S. 146]

Ziel der Phase **Planung der Flexibilität** ist die Gestaltung einer zyklenrobusten Plattform- und Modulararchitektur. Im Top-Down-Ansatz wird dazu die erforderliche Flexibilität des Plattformsystems aus strategischer Sicht festgelegt. Wesentliche Aufgaben sind die Antizipation dynamischer Einflussfaktoren (DEF) auf das Plattformsystem, die Festlegung von Änderungszyklen und die Quantifizierung der erforderlichen plattformsystemspezifischen Flexibilität auf Feature-Ebene. Dazu werden Produkt-Features in stabile Plattform-Features und flexible Features unterschieden. Im Bottom-Up-Ansatz ist es die Aufgabe der Produktentwicklung, die erforderliche Flexibilität möglichst effizient und kostengünstig zu realisieren. Hierzu wird durch die Verknüpfung der Produkt-Features mit den zugehörigen Bauteilen die erforderliche Flexibilität auf Modulebene ermittelt. Die anschließende Architekturgestaltung folgt dem Gedanken, dass stabile Bauteile mit

⁴⁹ Nach MAURER ET AL. umfasst der Begriff Plattformsystem neben der Produktarchitektur auch Prozesse und Organisationsstrukturen [BCE14, S. 11].

einer hohen Kritikalität und zeitlichen Robustheit innerhalb einer Plattform zu integrieren sind; Bauteile mit geringer Kritikalität und hoher Dynamik hingegen in flexiblen Modulen. Die Entscheidung je Bauteil wird durch das in Bild 3-23 dargestellte Kritikalitätsdynamik-Portfolio unterstützt [MBE+14, S. 150ff.], [BCE14, S. 11ff.].

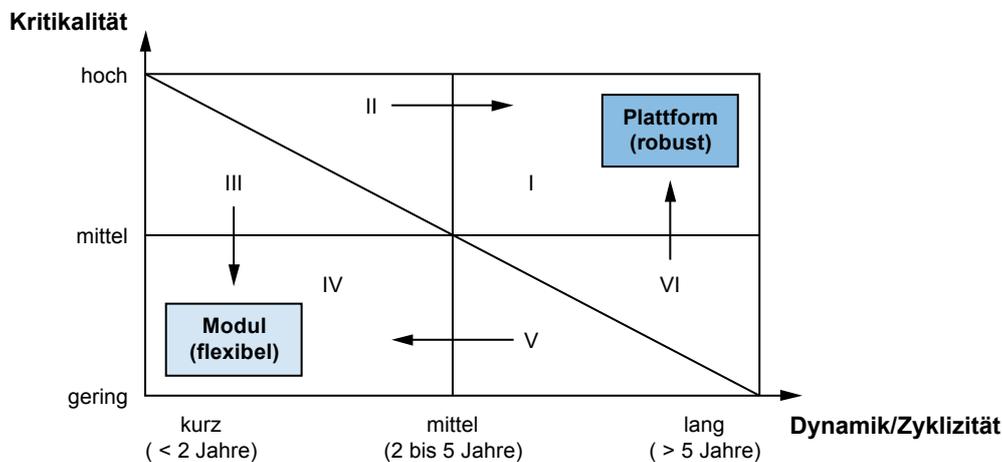


Bild 3-23: Kritikalitätsdynamik-Portfolio nach MAURER ET AL. [MBE+14, S. 151]

Im Rahmen der **Operationalisierung** erfolgt die Umsetzungsplanung unter Berücksichtigung der zuvor ermittelten dynamischen Einflussfaktoren und Änderungszyklen. In einer Feature-Roadmap werden dazu die Zeitpunkte festgelegt, an denen neue Produkt-Features für den Markt zugänglich sein sollen. Die Modul-Roadmap gibt die Zeiträume der zur jeweiligen Feature-Erfüllung erforderlichen Modulentwicklungen an. Die Synchronisation beider Roadmaps ergibt eine Produkt-Roadmap, die die Markteinführungstermine für zukünftige Produktversionen beschreibt. Informationen über verfügbare Entwicklungsressourcen und das Produktprogramm komplettieren die Produkt-Roadmap [MBE+14, S. 142ff.], [BCE14, S. 11ff.].

Im Rahmen des **Lebenszyklusmanagements** erfolgt die kontinuierliche Überwachung und Anpassung des Plattformsystems unter Berücksichtigung veränderter Rahmenbedingungen. Top-Down erfolgt die Kontrolle der marktseitigen Leistungsfähigkeit des Plattformsystems. Kennzahlen sorgen für den Abgleich der vom Markt verlangten Flexibilität mit der innerhalb des Plattformsystems vorhandenen Flexibilität (Kontrollplan Strategie). Bottom-Up erfolgt die Überwachung und Koordination des Produktprogramms sowie das Management der Überarbeitungs- und Änderungsaktivitäten. Kernaufgabe des Lebenszyklusmanagements ist eine synchronisierte und dauerhafte Pflege der Produkt-Roadmap (Kontrollplan Umsetzung) [MBE+14, S. 153], [BCE14, S. 11ff.].

Bewertung: Der Ansatz nach MAURER ET AL. bietet ein umfassendes Vorgehen zur Gestaltung und Pflege von Produktplattformen. Hervorzuheben ist dabei die Synchronisation von Markt- bzw. Strategie-Sicht (Top-Down) mit der Produktentwicklung bzw. Produktarchitektur (Bottom-Up). Das Vorgehen ermöglicht die gezielte Produktwertsteigerung und die systematische Berücksichtigung konkurrierender Stakeholder-Präferenzen. Zugleich unterstützen die Phasen *Planung der Flexibilität* und *Operationalisierung* in

Teilen die initiale Strukturierung des Release-Plans. Ähnlich wie beim Ansatz nach SCHUH ET AL. erfolgt jedoch keine interdisziplinäre Analyse der Änderungsauswirkungen und -abhängigkeiten. Eine harmonisierte Bündelung von Änderungen findet nicht statt.

3.5.4 Release-Management nach BELENER

BELENER stellt in seiner Dissertation ein ausführliches Konzept zum technischen Änderungsmanagement modularer Produkte und Prozesse vor. Das Konzept gliedert sich in vier wesentliche Bestandteile, die in Bild 3-24 gegliedert nach den Phasen Produkterstellung und Serienproduktion dargestellt sind [Bel08].

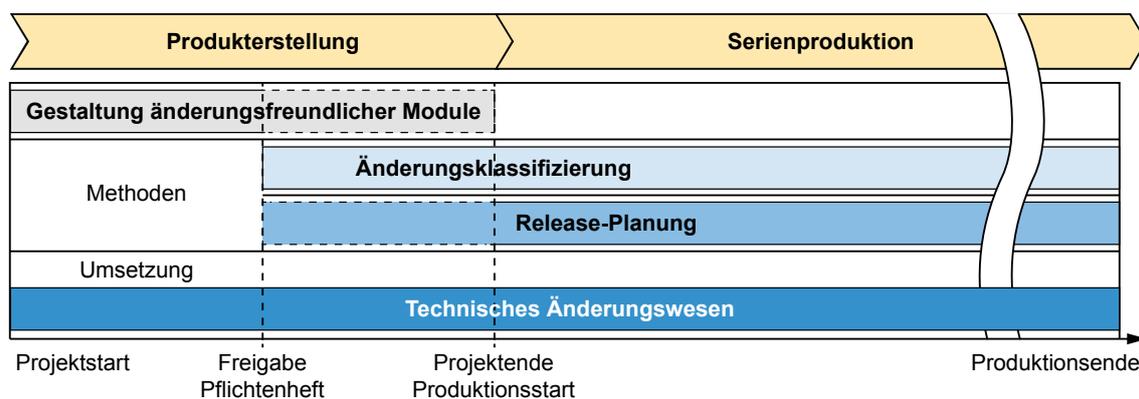


Bild 3-24: Übersicht zum technischen Änderungsmanagement nach BELENER [Bel08]

Mit dem Ziel eines vorsorgenden Änderungsmanagements wird ein Ansatz zur *Gestaltung änderungsfreundlicher Module* vorgestellt. Den zweiten Schwerpunkt bildet eine Methode zur *Änderungsklassifizierung* als Voraussetzung für die darauf aufbauende *Release-Planung*. Das *technische Änderungswesen* beschreibt die Aufbau- und Ablauforganisation im Unternehmen zur Umsetzung der übrigen Bestandteile [Bel08]. Im Folgenden liegt der Fokus auf den für diese Arbeit relevanten Bestandteilen der Änderungsklassifizierung und Release-Planung.

Die **Methode zur Änderungsklassifizierung** gliedert sich in die beiden Schritte 1) Bestandsaufnahme und Änderungsanalyse sowie 2) Änderungsbewertung. Im ersten Schritt – der Bestandsaufnahme und Änderungsanalyse – werden alle im Unternehmen bekannten Änderungen gesammelt und den betroffenen Modulen zugeordnet. Gleichzeitig erfolgt die Abschätzung der zur Umsetzung der Änderung benötigten finanziellen und zeitlichen Ressourcen. Im zweiten Schritt – der Änderungsbewertung – werden die erarbeiteten Änderungen anhand definierter Kriterien und Unterkriterien bewertet und gemäß dem Eisenhower-Prinzip nach ihrer Dringlichkeit und Wichtigkeit klassifiziert. Das Resultat ist ein Portfolio, das eine konkrete Empfehlung zur weiteren Bearbeitung der Änderungsanforderung umfasst (Bild 3-25) [Bel08, S. 75ff.].

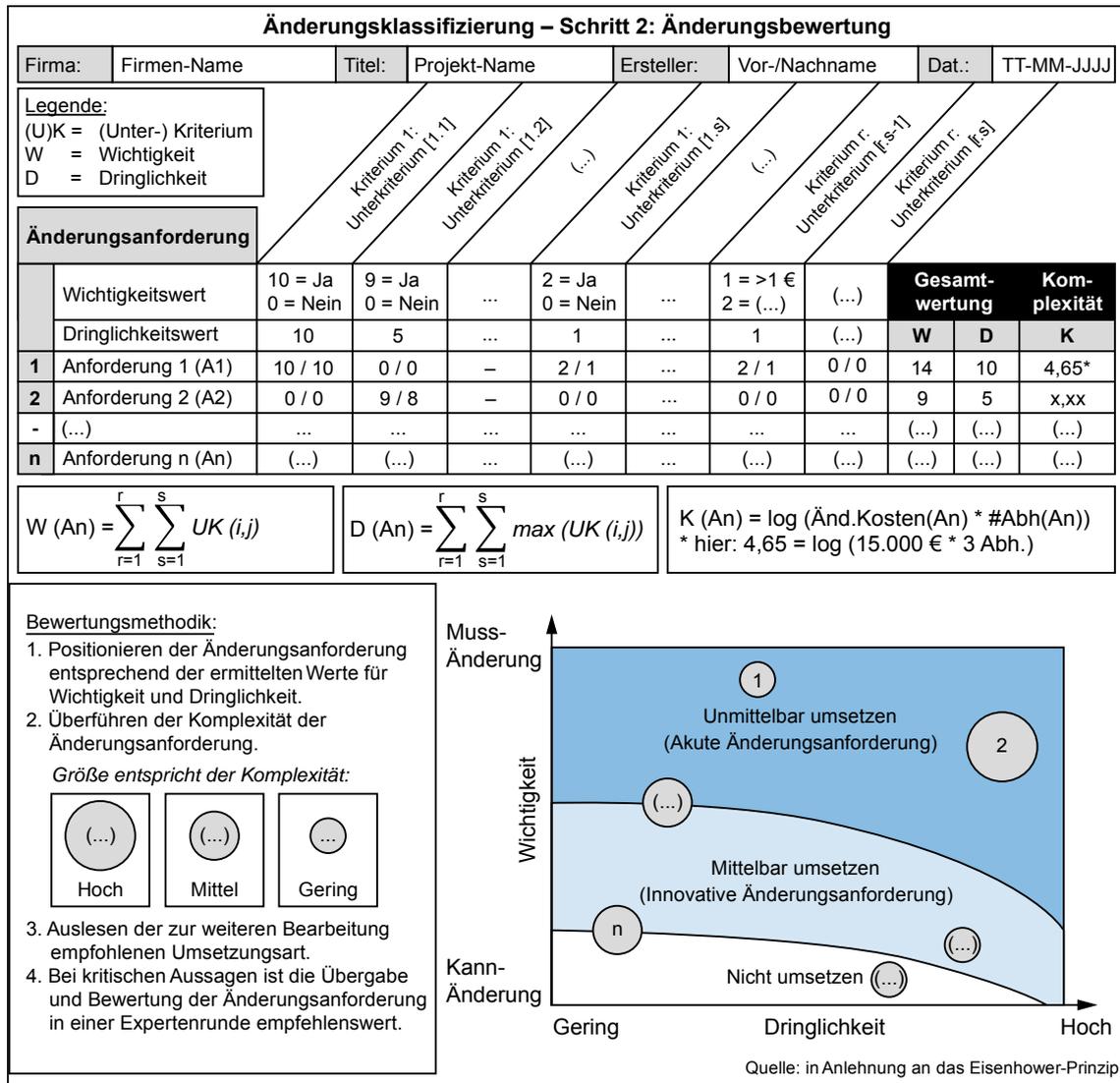


Bild 3-25: Änderungsbewertung nach BELENER [Bel08]

Die **Methode zur Release-Planung** baut als dritter Schritt auf den Ergebnissen der vorherigen Änderungsklassifizierung auf und bezieht sich auf alle Änderungen, die nicht unmittelbar umzusetzen sind. Diese gilt es, für eine gemeinsame Umsetzung in Form von Release-Einheiten zu bündeln. Dazu werden alle Änderungen aufgelistet und bzgl. ihrer Modul- und Prozesszugehörigkeit geordnet (Bild 3-26) [Bel08, S. 79f.].

Schritt 3: Release-Planung													
Firma:	Firmen-Name		Titel:	Projekt-Name		Ersteller:	Vor-/Nachname		Dat.:	TT-MM-JJJJ			
Gewichtung:			Summe der Gewichtungen ergibt den Wert 1.										
Wertung je Kriterium:			Modul-zugehörigkeiten		Kriterium 1: Dauer der Änderungsdurchführung		Kriterium 2: Zyklusdauer zum Folge-Release		(...)		Kosten der Änderungsdurchführung		
Anzahl der gemeinsamen Werte im Wertebereich			0,25	0,25	0,15	0,1	(...)		In TEUR	Release-Wertung	Nr.	Kosten: [Tsd. Euro]	Dauer: [Monate]
Änderungsanforderung	# Änd. Anf.	Modul Nr.		Monate (+- 3)	Monate (+- 6)	(...)		In TEUR	Release-Wertung	Nr.	Kosten: [Tsd. Euro]	Dauer: [Monate]	
1 Anforderung 1	3	2	6	4	12	-		14	12,9	1	42	10	
2 Anforderung 2		2	7	6	14	(...)		4					
3 Anforderung 5		2	7	6	21	(...)		4					
- (...)	(...)	(...)	(...)	(...)	(...)	(...)		(...)	(...)	(...)	(...)	(...)	
n Anforderung n	(...)	(...)	(...)	(...)	(...)	(...)		(...)	7,35	2	(...)	(...)	
Release-Wertung: $= \sum_{i=1}^n (WertungKrit(i) * GewichtungKrit(i)) / \text{ÄndAnf.}$													

Bild 3-26: Bewertungsschema zur Erzeugung von Release-Einheiten nach BELENER [Bel08, S. 80]

Zur Erzeugung potentieller Release-Einheiten wird anschließend die Verträglichkeit der Änderungen untereinander anhand verschiedener Kriterien bewertet (z.B. Modulzugehörigkeit, Dauer der Änderungsdurchführung). Sofern sich die Einzelbewertungen innerhalb eines ähnlichen Wertebereichs befinden, sind die Änderungen für das Kriterium als verträglich zu bewerten. Sind bspw. beim Kriterium Dauer der Änderungsdurchführung drei Monate angegeben, dann wird untersucht, wie viele Änderungen innerhalb der gleichen Zeitspanne liegen. Je nach Gewichtung der Kriterien ergeben sich unterschiedliche Release-Einheiten, über deren Umsetzung anschließend in einem Workshop zu entscheiden ist. Hierbei dient die sogenannte Release-Wertung als Indikator für die Verträglichkeit der in einem Release enthaltenen Änderungen [Bel08, S. 79f.].

Bewertung: Der Ansatz nach BELENER ordnet sich in ein umfassendes Konzept für das technische Änderungsmanagement modularer Produkte ein, das prinzipiell auf unterschiedliche technische Systeme übertragbar ist. Das Vorgehen zur Release-Planung zeichnet sich durch eine systematische Änderungsklassifizierung aus und bietet mit der Release-Wertung einen Ansatz zur systematischen Bündelung von Änderungen zu Releases. Hierbei bleibt allerdings unklar, wie die verschiedenen Releases erzeugt werden. Für die Strukturierung des Release-Plans gibt der Ansatz zwar einige Hinweise, eine systematische Unterstützung bleibt jedoch bei der Festlegung von Release-Zeitpunkten und Release-Typen aus.

3.6 Bewertung und Handlungsbedarf

Im Kapitel Stand der Technik wurde jeder beschriebene Ansatz anhand der in Abschnitt 2.7 abgeleiteten Anforderungen individuell bewertet. Im Folgenden erfolgt zusätzlich eine Bewertung je Anforderung, die in Bild 3-27 zusammenfassend dargestellt ist.

A1) Durchgängiger Planungsprozess: Anhaltspunkte für einen durchgängigen Planungsprozess liefern sowohl bestehende Ansätze der strategischen Produktplanung als auch der Release-Planung. Insbesondere das Vorgehen nach NIPPA und LABRIOLA trägt dieser Anforderung durch die Kaskadierung in eine lang-, mittel- und kurzfristige Innovationsplanung Rechnung. Für die Release-Planung ist dieser Ansatz jedoch zu abstrakt. Bestehende Ansätze im Kontext der Release-Planung reichen abschnittsweise von der strategischen Definition kundenrelevanter Releases (SCHUH ET AL., MAURER ET AL.) bis zur operativen Umsetzungsplanung von Releases (ZORN-PAULI ET AL., BELENER). Ein Planungsprozess, der sämtliche Planungsebenen verknüpft und zudem Wechselwirkungen zwischen den Planungsebenen berücksichtigt, existiert nicht. Dies ist in der zu entwickelnden Systematik aufzugreifen.

A2) Strukturierung des Release-Plans: Ansätze zur Strukturierung des Release-Plans speziell für Baukastensysteme und Plattformen liefern SCHUH ET AL. und MAURER ET AL. Beide Ansätze können im Rahmen der zu entwickelnden Systematik aufgegriffen werden. Die bestehenden Ansätze der strategischen Produktplanung liefern darüber hinaus Anhaltspunkte zur umfassenden Berücksichtigung von sowohl markt- und wettbewerbsstrategischen als auch technologischen Überlegungen. Davon ausgehend ist im Rahmen dieser Systematik eine Entscheidungsunterstützung zur unternehmens- und produktspezifischen Strukturierung des Release-Plans zu entwickeln.

A3) Gesteuerte Produktwertsteigerung: Das Kano-Modell eignet sich zur Klassifizierung von Änderungen nach ihrem Innovationsgrad und kann so als Hilfsmittel zur Steuerung der Produktwertsteigerung eingesetzt werden. Darüber hinaus greifen insb. die Ansätze der strategischen Produktplanung die Anforderung auf. Im Fokus steht hier jedoch eher die systematische Identifikation von Innovationen anhand antizipierter Markt- und Technologieentwicklungen. Die gezielte Einsteuerung von Änderungen unter Berücksichtigung technischer Auswirkungen und Abhängigkeiten wird hingegen nicht betrachtet. Lediglich der Release-Planungs-Ansatz nach MAURER ET AL. vereint beide Perspektiven durch die Synchronisation von Strategie-Sicht (Top-Down) mit der Produktentwicklung bzw. Produktarchitektur (Bottom-Up). Diese Denkweise ist in der zu entwickelnden Systematik aufzugreifen und anzupassen.

A4) Berücksichtigung von Stakeholder-Präferenzen: BERANDER und ANDREWS geben einen Überblick über verschiedene Priorisierungstechniken, die sich zur Berücksichtigung von Stakeholder-Präferenzen eignen. Ähnlich wie das Kano-Modell sind diese an geeigneter Stelle als Hilfsmittel in die zu entwickelnde Systematik einzubetten.

A5) Änderungsauswirkungen und -abhängigkeiten: Alle im Stand der Technik untersuchten Ansätze zur Analyse von technischen Änderungen liefern Ansätze zur Identifikation von Änderungsauswirkungen und -abhängigkeiten. Sie sind unter Berücksichtigung ihrer Vor- und Nachteile als Hilfsmittel in der zu entwickelnden Systematik aufzugreifen, anzupassen und zu erweitern. Die bestehenden Ansätze zur Release-Planung berücksichtigen Änderungsauswirkungen und -abhängigkeiten nur ansatzweise. Insbesondere Abhängigkeiten zwischen Änderungen fließen bisher nicht in die Release-Planung ein.

A6) Harmonisierte Bündelung von Änderungen zu Releases: Lediglich die Ansätze nach ZORN-PAULI ET AL. und BELENER bündeln Änderungen unter Berücksichtigung unterschiedlicher Kriterien, Rahmenbedingungen und Restriktionen zu Releases. In beiden Fällen bettet sich das Vorgehen jedoch nicht in einen durchgängigen Release-Planungsprozess ein. Hier besteht Handlungsbedarf.

A7) Interdisziplinarität: Mit Ausnahme des softwarespezifischen Ansatzes nach ZORN-PAULI ET AL. zielen alle untersuchten Ansätze zur strategischen Produkt- bzw. Release-Planung auf mechatronische Produkte ab. Eine Übertragbarkeit auf intelligente technische Systeme ist gegeben, so dass der Anforderung nach Interdisziplinarität grundsätzlich Rechnung getragen wird. Bedeutend ist diese Anforderung insb. im Kontext der Spezifikation technischer Systeme. Hier zeichnen sich CONSENS und SysML als geeignetes Beschreibungsmittel ab, um der Forderung nach Interdisziplinarität gerecht zu werden.

A8) Unterstützung durch modellbasierte Systembeschreibung: Keiner der untersuchten Ansätze zur Spezifikation technischer Systeme wird dieser Anforderung vollständig gerecht. Zwar liegen mit CONSENS und SysML zwei umfangreiche Ansätze zur Beschreibung technischer Systeme vor; diese vereinen kundenorientierte und technische Sicht jedoch nur unzureichend. Darüber hinaus bieten sie keinen zufriedenstellenden Ansatz zur Berücksichtigung der zeitlichen Varianz bei der Weiterentwicklung des technischen Systems. Hier weisen Feature-Modelle eine Stärke auf, so dass sich eine Kombination der Ansätze anbietet.

A9) Systematisches Vorgehen und Praktikabilität: Die meisten der vorgestellten Ansätze bieten zwar eine systematische und reproduzierbare Vorgehensweise, decken dabei aber nur Teilaspekte der Release-Planung ab. Eine umfassende Vorgehensweise existiert nicht. Darüber hinaus ist die Praktikabilität einiger Ansätze auf Grund ihrer Komplexität zweifelhaft.

Zusammenfassend kann festgehalten werden, dass keiner der untersuchten Ansätze alle Anforderungen aus Abschnitt 2.7 vollständig erfüllt. Auch eine einfache Kombination der Ansätze wird den Anforderungen nicht im vollen Umfang gerecht. Entscheidende Defizite sind die unzureichende Verzahnung der verschiedenen Planungsebenen sowie die nur sporadische Berücksichtigung von Änderungsauswirkungen und Änderungsabhängigkeiten. Es besteht somit dringender Handlungsbedarf, eine *Systematik zur Release-Planung intelligenter technischer Systeme* zu entwickeln.

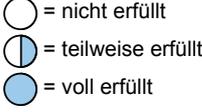
Bewertungsskala: 		Anforderungen								
		Durchgängiger Planungsprozess	Strukturierung des Release-Plans	Gesteuerte Produktwertsteigerung	Stakeholder-Präferenzen	Änderungsauswirkungen- u. abhängigkeiten	Harmonisierte Bündelung von Änderungen	Interdisziplinarität	Modellbasierte Systembeschreibung	Systematisches Vorgehen u. Praktikabilität
		A1	A2	A3	A4	A5	A6	A7	A8	A9
Ansätze zur Spezifikation technischer Systeme	Spezifikationstechnik CONSENS									
	Modellierungstechniken basierend auf SysML									
	A3 Architecture Overview									
	Matrixbasierte Produktmodellierung									
	Feature-Modelle									
Analyse technischer Änderungen	Change Prediction Method nach CLARKSON ET AL.									
	Technology-Infusion nach SUH ET AL.									
	Analyse von Änderungsauswirkungen nach NONSIRI ET AL.									
	Analyse von Änderungen mit SysML4Mechatronics									
Klassifizierung und Priorisierung von Änderungen	Kano-Modell									
	Klassifizierung von Änderungen nach ASSMANN									
	Klassifizierung von Änderungen nach ALBERS ET AL.									
	Priorisierungstechniken nach BERANDER und ANDREWS									
Strategische Produktplanung	Time-to-Market Management nach NIPPA und LABRIOLA									
	Entwicklung von Produkt-Technologiestrategien nach BRINK									
	Geschäftsmodell-Roadmaps nach PEITZ									
	Lebenszyklusorientierte Produktplanung nach HEPPERLE									
Produkt-Release-Planung	Release-Planungs-Prozess nach ZORN-PAULI ET AL.									
	Modulbasierte Release-Planung nach SCHUH ET AL.									
	Modul- und Plattformstrategien nach MAURER ET AL.									
	Release-Management nach BELENER									

Bild 3-27: Bewertung des untersuchten Stands der Technik anhand der in Abschnitt 2.7 abgeleiteten Anforderungen

4 Vorstellung der Systematik zur Release-Planung intelligenter technischer Systeme

“If you fail to plan, you are planning to fail!” – BENJAMIN FRANKLIN

Die Analyse des Stands der Technik in Kapitel 3 zeigt auf, dass es an einem durchgängigen Ansatz für die Release-Planung entsprechend der in Kapitel 2 abgeleiteten Anforderungen fehlt. Kapitel 4 greift diesen Handlungsbedarf auf und beschreibt die im Rahmen dieser Arbeit entwickelte *Systematik zur Release-Planung intelligenter technischer Systeme*. Dieses Kapitel bildet somit den Kern dieser Arbeit.

In **Abschnitt 4.1** wird zunächst die Grundidee der Release-Planung sowie der dazugehörige Lösungsansatz beschrieben, während **Abschnitt 4.2** einen Überblick über die Systematik und seine wesentlichen Bestandteile gibt. Für den Prozess der Release-Planung hat die initiale Strukturierung des Release-Plans weitreichende Bedeutung; sie folgt alternativen Release-Strategien, deren Gestaltung und Auswahl in **Abschnitt 4.3** diskutiert wird. Der Release-Planungsprozess gliedert sich in eine strategische, taktische und operative Planung, die jeweils durch ein spezifisches Vorgehensmodell unterstützt werden. Diese werden in **Abschnitt 4.4** erläutert. Zur Unterstützung der interdisziplinären Zusammenarbeit wird in **Abschnitt 4.5** das Systemmodell als Kooperationskern in der Release-Planung eingeführt. Hier wird die Spezifikationstechnik CONSENS (vgl. Abschnitt 3.1.1) um spezifische Anforderungen im Kontext der Release-Planung erweitert. Die eigentliche Anwendung der Systematik anhand konkreter Beispiele wird in Kapitel 5 beschrieben.

4.1 Grundidee und Lösungsansatz

Ausgangspunkt der Systematik ist der Bedarf nach einer systematischen Umsetzung von technischen Änderungen. Hier setzt die **Grundidee** der Release-Planung an: die Bündelung von Änderungen zu dedizierten Releases. Dies erfolgt im Spannungsfeld von strategischen Vorgaben, konkurrierenden Stakeholder-Präferenzen, Änderungsaufwänden und verfügbaren Ressourcen. Alle diese Aspekte gilt es, in einem systematischen Planungsprozess gleichermaßen zu berücksichtigen und auszubalancieren. **Lösungsansatz** ist die Hierarchisierung der Planung in aufeinander aufbauende und sich gegenseitig beeinflussende Planungsebenen (Bild 4-1).

Die Basis für die Hierarchisierung der Planungsebenen liefern Ansätze aus dem Innovations- und Entwicklungsmanagement. Diese unterteilen den Planungsprozess in eine strategische, taktische und operative Planung, die sich typischerweise durch den jeweiligen Planungshorizont unterscheiden [WC93, S. 69ff.], [NL08, S. 304ff.], [HHA12, S. 22]. Die entwickelte Systematik greift diese Hierarchisierung übertragen auf die Release-Planung intelligenter technischer Systeme auf.

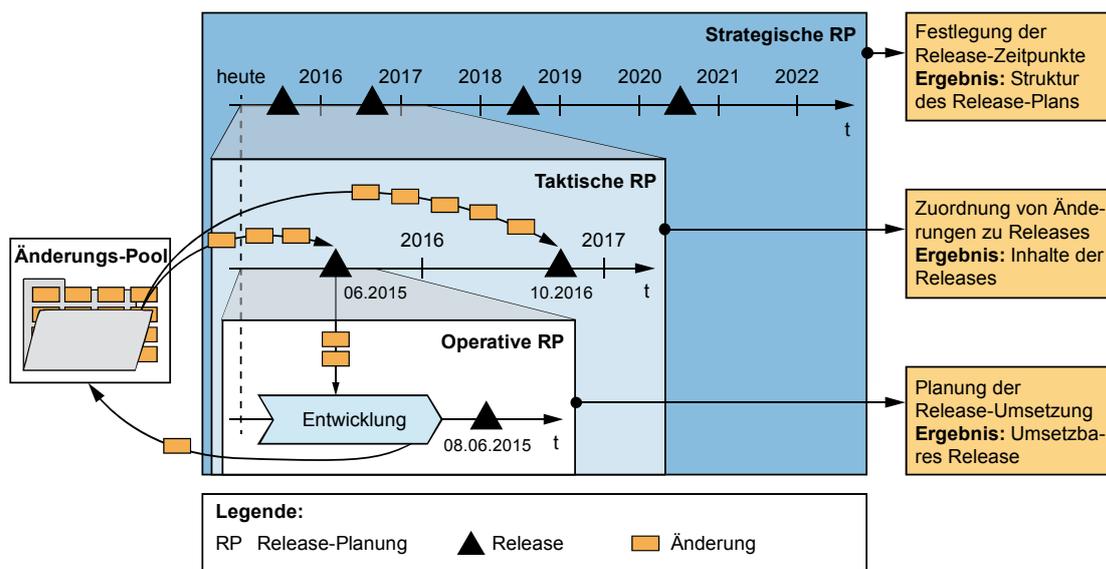


Bild 4-1: Planungsebenen der Systematik zur Release-Planung intelligenter technischer Systeme

Die **strategische Release-Planung** dient der langfristigen Release-Planung und weist je nach Produktart und Branche einen Planungshorizont von fünf bis 15 Jahren auf. Die Planungsgenauigkeit nimmt mit dem Zeitverlauf ab, so dass eine jährliche Aktualisierung im Sinne einer rollierenden Planung erforderlich ist. Kernaufgabe ist die Strukturierung des Release-Plans durch die Festlegung von Release-Typen und Release-Zeitpunkten. Grundlage hierfür ist eine zu erarbeitende Release-Strategie, die unter Berücksichtigung unternehmens- und produktspezifischer Eigenschaften u.a. Vorgaben zu Häufigkeit, Inhalt und zum Erscheinungszeitraum neuer Releases trifft (vgl. Abschnitt 4.3). Die Vorgaben der Release-Strategie sind mit denen der Unternehmens- und Produktstrategie abzugleichen und zu synchronisieren. So beruht die strategische Release-Planung stark auf der Antizipation der Entwicklung von Technologien (erwarteter Technology Push), Märkten (erwarteter Market Pull), Geschäftsumfeldern und der Wettbewerbsarena. Verantwortlich für die strategische Release-Planung ist der Produktmanager. Dieser plant auf Basis von produktstrategischen Vorgaben der Unternehmensleitung (z.B. Messeauftritte) und stellt in Zusammenarbeit mit dem Systemarchitekten die Berücksichtigung technischer Rahmenbedingungen sicher (z.B. Art und Häufigkeit neuer Releases, Verfügbarkeit von Ressourcen). Aufgrund des langen Planungshorizonts und der damit verbundenen Unsicherheiten trifft die strategische Release-Planung Aussagen zu Zeiträumen für die Marktveröffentlichung einzelner Releases, zur Definition von Release-Typen bzw. Änderungsklassen und zur Berücksichtigung langläufig geplanter Innovationsprojekte. Ergebnis des strategischen Planungsprozesses ist folglich ein vorstrukturierter Release-Plan.

Die **taktische Release-Planung** hat einen weitaus kürzen Planungshorizont, der sich auf ein bis zwei Jahre bzw. zwei bis vier Releases im Voraus beschränkt. Es handelt sich um

einen kontinuierlichen Planungsprozess, der sich an den generischen Änderungsprozess nach JARRETT ET AL. anlehnt (vgl. Abschnitt 2.5.4), diesen aber für die Release-Planung ausprägt und erweitert. Die Kernaufgabe der taktischen Release-Planung liegt in der Zuordnung von Änderungen und Features zu potentiellen Releases und damit in der Umsetzungsplanung. Grundlage hierfür ist ein Änderungs-Pool⁵⁰, der kontinuierlich mit Änderungsanforderungen (engl. Change Requests) sowie noch nicht realisierten Produkt-Features gefüllt wird. Diese werden in einem systematischen Bewertungsprozess klassifiziert und den in der strategischen Release-Planung definierten Releases zugeordnet. Die Bewertung erfolgt entsprechend der strategischen Vorgaben und berücksichtigt bspw. Änderungsauswirkungen und -abhängigkeiten sowie Dringlichkeit. Im Ergebnis liegt ein inhaltlich verfeinerter Release-Plan vor, der darüber hinaus den Zeitraum für die Veröffentlichung der Releases gegenüber der strategischen Planung weiter eingrenzt. Hauptverantwortlich für den Prozess der taktischen Release-Planung ist in der Regel ein Änderungsmanager, der sämtliche Änderungen verwaltet. Für die Bewertung und Umsetzungsplanung werden jedoch relevante Stakeholder hinzugezogen, die marktseitig vor allem durch den Produktmanager und aus technischer Sicht durch den Systems Engineer repräsentiert werden.

Die **operative Release-Planung** hat kurzfristigen Charakter und fokussiert den Planungshorizont auf das nächste umzusetzende Release. Sie ist die Schnittstelle zur anschließenden Projektplanung. Kernaufgabe ist die Planung der Release-Umsetzung unter Berücksichtigung der verfügbaren Ressourcen, die u.U. die Anzahl der umsetzbaren Änderungen einschränken. Ziel ist daher die harmonisierte Bündelung von Änderungen, die über eine Priorisierung die konkreten Release-Inhalte definiert. Grundlage dieser Bündelung sind unterschiedliche Kriterien (z.B. Dringlichkeit, Modulzugehörigkeit) sowie Restriktionen (z.B. begrenzte Ressourcen), die sowohl aus strategischen als auch aus taktischen Vorgaben resultieren. Im Falle unzureichender Ressourcen werden Änderungen von geringer Priorität zurück in den Änderungs-Pool gespielt und in einem der nächsten Releases auf Umsetzbarkeit geprüft. Aufgrund der hohen Planungsgenauigkeit insb. hinsichtlich der verfügbaren Ressourcen legt die operative Release-Planung zugleich den finalen Markteinführungszeitpunkt fest. Hauptverantwortlich hierfür ist der Projektmanager, der im Zusammenspiel mit Produktmanager und Systems Engineer den Planungsprozess steuert.

Das Ergebnis des gesamten Planungsprozesses ist ein konkreter Release-Plan, der als Basis für eine gesteuerte und systematische Änderungsumsetzung zu sehen ist. Der Release-Plan enthält alle erforderlichen Informationen für die Planung des nächsten Release-Projekts sowie erste Informationen über die Inhalte der darauffolgenden Releases. Damit

50 Der Änderungs-Pool steht in dieser Arbeit repräsentativ für eine Datenbank, in der Änderungen in einem systematischen Prozess erfasst, bewertet, dokumentiert und verwaltet werden. Dies wird typischerweise in PLM-Systemen mit Hilfe spezieller Module für das Änderungsmanagement realisiert. Ein Beispiel ist das Modul CIM DATABASE ECM der Firma CONTACT Software.

eignet sich der ausgearbeitete Release-Plan mit seinen verschiedenen Planungshorizonten als interne und externe Gesprächs- und Diskussionsgrundlage. Tabelle 4-1 fasst die wesentlichen Aspekte der drei Planungsebenen gegliedert nach Planungshorizont, Aufgaben und hauptverantwortlichen Rollen zusammen. Der Planungsprozess wird auf allen drei Ebenen durch Vorgehensmodelle unterstützt, die in Abschnitt 4.4 im Überblick erläutert und in Kapitel 5 exemplarisch angewendet werden.

Tabelle 4-1: Wesentliche Aspekte der strategischen, taktischen und operativen Release-Planung intelligenter technischer Systeme

	Strategische Release-Planung	Taktische Release-Planung	Operative Release-Planung
Planungshorizont	<ul style="list-style-type: none"> • Je nach Produkt und Branche fünf bis 15 Jahre • Mit dem Zeitverlauf abnehmende Planungsgenauigkeit • Jährliche Aktualisierung 	<ul style="list-style-type: none"> • Ein bis zwei Jahre bzw. zwei bis vier Releases im Voraus • Kontinuierliche Aktualisierung 	<ul style="list-style-type: none"> • Immer das nächste Release • Dauer der Release-Umsetzung bzw. des Entwicklungsprojekts
Aufgaben	<ul style="list-style-type: none"> • Festlegung der Release-Strategie • Erstellung des initialen Release-Plans • Abgleich des Release-Plans mit Produkt- und Technologie-strategie 	<ul style="list-style-type: none"> • Änderungsanalyse und -klassifizierung • Zuordnung von Änderungen zu Releases 	<ul style="list-style-type: none"> • Planung der Release-Umsetzung unter Berücksichtigung verfügbarer Ressourcen • Harmonisierte Bündelung von Änderungen • Priorisierung von Änderungen
Hauptverantwortliche Rollen	<ul style="list-style-type: none"> • Produktmanager • Unternehmensleitung • Systems Engineer 	<ul style="list-style-type: none"> • Änderungsmanager • Produktmanager • Systems Engineer 	<ul style="list-style-type: none"> • Projektmanager • Produktmanager • Systems Engineer

4.2 Überblick über die Systematik

Die entwickelte Systematik beansprucht, dem aufgezeigten Handlungsbedarf aus Problemanalyse und Stand der Technik gerecht zu werden. Sie beschreibt die konkrete Umsetzung des hierarchisierten Release-Planungsprozesses und unterstützt die beteiligten Fachleute durch geeignete Vorgehensmodelle und Hilfsmittel. Dabei zielt sie nicht auf die Automatisierung des Planungsprozesses ab. Vielmehr dient sie der Entscheidungsunterstützung durch die situationsgerechte Aufbereitung planungsrelevanter Informationen. Die wesentlichen Bestandteile der Systematik und ihr grundsätzliches Zusammenwirken sind in Bild 4-2 dargestellt.

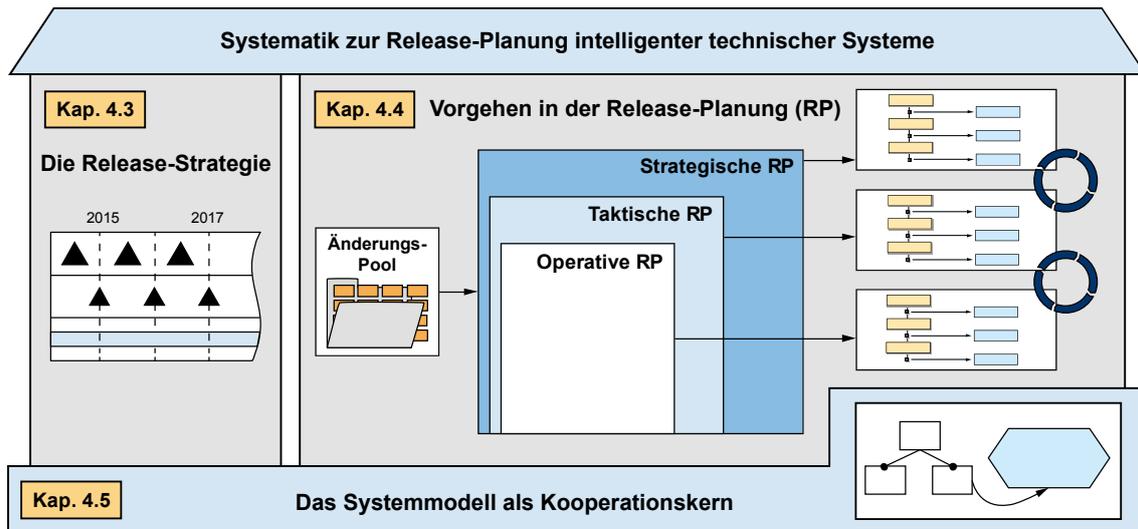


Bild 4-2: Bestandteile der Systematik zur Release-Planung intelligenter technischer Systeme

Basis für die Release-Planung ist die Auswahl einer geeigneten **Release-Strategie**. Diese macht über verschiedene Gestaltungsfaktoren und zugehörige Ausprägungen Vorgaben für die initiale Strukturierung des Release-Plans im Rahmen der strategischen Release-Planung. Diese Vorgaben gilt es, unter Berücksichtigung verschiedener Einflüsse zu konkretisieren (z.B. markt- und wettbewerbsstrategische Überlegungen, Beschaffenheit der Systemarchitektur, verfügbare Entwicklungsressourcen) (Abschnitt 4.3).

Vorgehensmodelle beschreiben und strukturieren die Tätigkeiten entlang der drei Ebenen des Release-Planungsprozesses – von der strategischen über die taktische bis zur operativen Release-Planung. Die Vorgehensmodelle definieren die planungsrelevanten Ergebnisse und integrieren die dafür notwendigen Hilfsmittel, z.B. eine Methode zur Analyse von Änderungsauswirkungen und eine Methode zur Klassifizierung von Änderungen (Abschnitt 4.4).

Unterstützt wird der Planungsprozess durch ein **Systemmodell als Kooperationskern** in der Release-Planung. Zu diesem Zweck wird auf die Spezifikationstechnik CONSENS zurückgegriffen, die entsprechend der Anforderungen aus Kapitel 2 erweitert wird. Im Kern umfasst dies die Erweiterung um ein Feature-Modell und den Release-Plan; beide ermöglichen im Zusammenspiel die Beschreibung der Produktevolution und gleichzeitig eine marktorientierte Sicht auf das technische System (Abschnitt 4.5).

Aufbauend auf der folgenden Beschreibung der Systematik und seiner Bestandteile wird in Kapitel 5 deren Anwendung vorgestellt. Anhand konkreter Beispiele werden dazu die einzelnen Phasen der Vorgehensmodelle durchlaufen.

4.3 Die Release-Strategie

Bereits in der Problemanalyse und in der Analyse des Stands der Technik konnte gezeigt werden, dass die Erarbeitung und Festlegung des initialen Release-Plans im Rahmen der strategischen Release-Planung einigen grundlegenden Gestaltungsfaktoren folgt. Beispiele sind Release-Einheiten, Release-Typen und der Rhythmus, mit dem die Releases veröffentlicht werden. Die Gestaltungsfaktoren müssen in Abhängigkeit von unternehmens- und produktstrategischen Überlegungen sowie unter Berücksichtigung der bestehenden Systemarchitektur unterschiedlich ausgeprägt werden. Diese Ausprägung wird im Folgenden als **Release-Strategie** bezeichnet. Die Release-Strategie gibt Rahmenbedingungen vor, die die Strukturierung des initialen Release-Plans im Hinblick auf zeitliche Planung, Inhalte und Rahmenbedingungen einzelner Releases unterstützen.

Um die Auswahl der für das jeweilige Produkt und die Unternehmenssituation passenden Release-Strategie zu unterstützen, wird in diesem Kapitel ein Ordnungsschema⁵¹ beschrieben. Das Ordnungsschema strukturiert alle relevanten Gestaltungsfaktoren und schafft so einen Überblick über die möglichen Ausprägungen alternativer Release-Strategien. In **Abschnitt 4.3.1** werden dazu ausgehend vom Stand der Technik die wesentlichen Gestaltungsfaktoren und entsprechende Ausprägungen extrahiert und strukturiert. Basis hierfür sind grundlegende Arbeiten zur Release-Planung ([SE04], [SDL+04], [Bel08], [Ruh10], [Wri12], [SAA13], [MBE+14]) und Erkenntnisse aus Release-Planungs-Ansätzen aus der Praxis. Das resultierende Ordnungsschema wird in **Abschnitt 4.3.2** beschrieben. Abschließend wird in **Abschnitt 4.3.3** ein Hilfsmittel zur Auswahl der passenden Release-Strategie vorgestellt, das sowohl unternehmensinterne als auch externe Einflussfaktoren berücksichtigt. Grundlage hierfür sind drei zuvor definierte Norm-Release-Strategien.

4.3.1 Gestaltungsfaktoren des Release-Plans

In Abschnitt 2.1.7 wurden vier konstituierende Elemente eines Release-Plans beschrieben: ein definierter Planungshorizont, Release-Zyklen, damit verbundene Release-Inhalte und zugehörige Entwicklungsaktivitäten bzw. Ressourcen. Diese Elemente sind bei der Erarbeitung des initialen Release-Plans unter Berücksichtigung strategischer Rahmenbedingungen auszuprägen. Ergänzende Vorgaben hierzu machen Gestaltungsfaktoren, die in Bild 4-3 exemplarisch anhand eines Release-Plans für eine Produktfamilie dargestellt sind. Im Folgenden werden die Gestaltungsfaktoren einschließlich ihrer möglichen Ausprägungen erläutert.

⁵¹ Nach DREIBHOLZ bezeichnet ein Ordnungsschema ein System zur Strukturierung von Lösungsalternativen in einer tabellenartigen Struktur und damit zur Schaffung eines Überblicks über den Lösungsraum [Dre75].

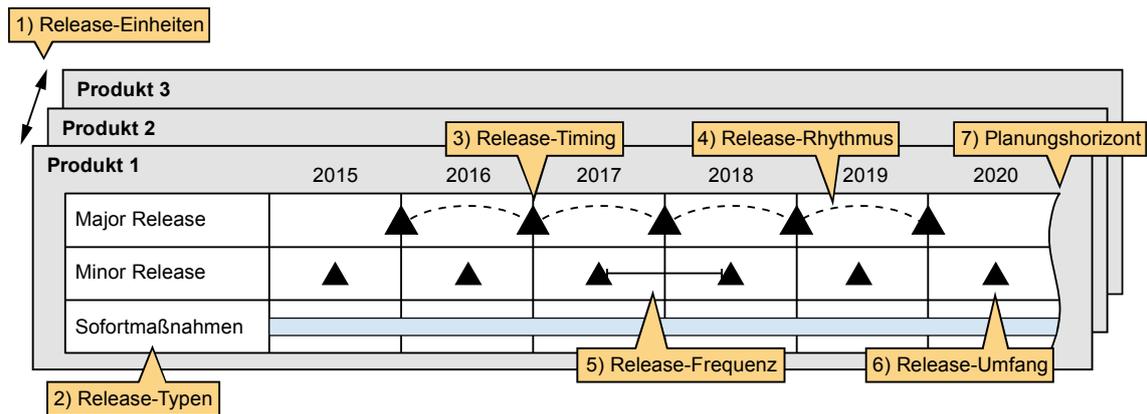


Bild 4-3: Überblick über die Gestaltungsfaktoren des Release-Plans

1) Release-Einheiten: Das Prinzip der Release-Planung kann auf verschiedene Zielsysteme angewendet werden. Dies reicht von der Release-Planung für ein dediziertes Produkt bis hin zur Planung für eine ganze Produktfamilie. Hierbei gilt: Je komplexer das Zielsystem, desto eher bietet sich die Gliederung in Release-Einheiten an. Eine **Release-Einheit**⁵² beschreibt ein Teilsystem des Produkts bzw. der Produktfamilie, dessen Veränderungen im Rahmen der Release-Planung eigenständig geplant, entwickelt, getestet und freigegeben werden. Entsprechend Bild 4-4 können zwei wesentliche Typen von Release-Einheiten unterschieden werden: Module und Produkte (bzw. Produktgruppen). In Spezialfällen bietet sich auch eine Kombination an (in Bild 4-4 nicht dargestellt).

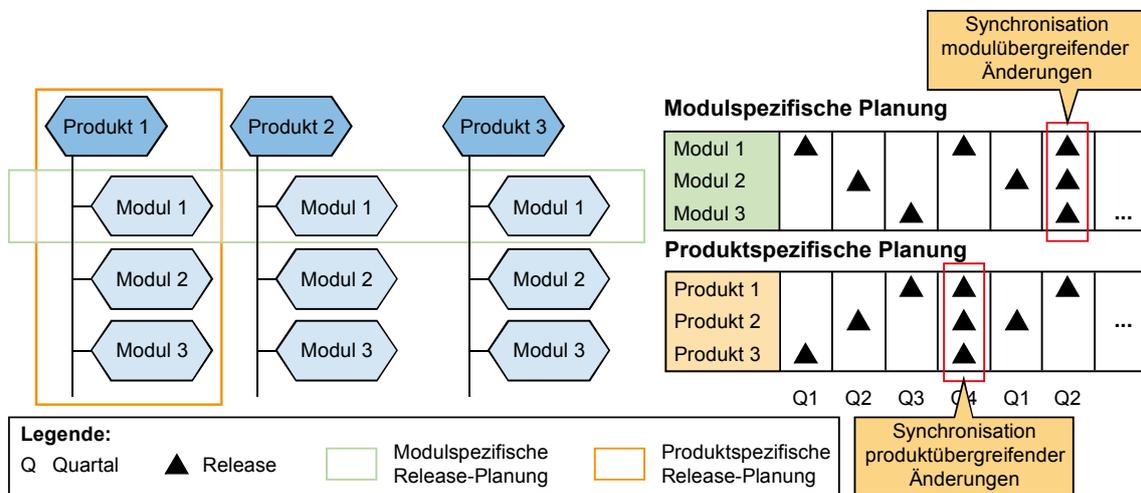


Bild 4-4: Gliederung einer Produktfamilie in alternative Release-Einheiten

Die kleinstmögliche Release-Einheit sind **Module**, die eine Menge an Systemelementen (Komponenten, Baugruppen und -teile) zusammenfassen. In diesem Fall werden Änderungen modulspezifisch geplant und umgesetzt. SCHUH ET AL. sprechen von einer modulspezifischen Release-Planung [SAA13]. Liegen modulübergreifende Änderungen vor, ist

⁵² Das Prinzip der Release-Einheiten entstammt dem Release-Engineering nach SCHUH und EVERSHEIM [SE04].

eine Synchronisation zwischen den einzelnen Modulen erforderlich. Eine Gliederung in Module ist sowohl für ein einzelnes Produkt als auch für eine Produktfamilie möglich. Insbesondere im Fall von Baukastensystemen mit einem hohen Grad an Kommunalität (Gleichheit) bietet sich die Gliederung in modulare Release-Einheiten an.

Der modulatorientierten Gliederung steht die Gliederung einer Produktfamilie in **Produkte** bzw. **Produktgruppen** (Zusammenfassung ähnlicher Produkte) gegenüber. In diesem Fall erfolgt die Planung und Umsetzung von Änderungen in Form von produkt- bzw. produktgruppenspezifischen Release-Einheiten. Die verfügbaren Entwicklungsressourcen werden so systematisch auf die Weiterentwicklung der einzelnen Produkte bzw. Produktgruppen verteilt. Im Falle von produktübergreifenden Gleichteilen ist auch hier eine Synchronisation zwischen den Produkten bzw. Produktgruppen erforderlich. Dies erfolgt durch geringfügige Anpassungen an den übrigen, betroffenen Produkten. Die produkt- bzw. produktgruppenspezifische Release-Planung bietet sich daher insb. für Produktfamilien an, die nur über einen geringen Anteil an Gleichteilen verfügen.

In Spezialfällen ist auch eine **Kombination** von modul- und produktspezifischer Planung möglich. In diesem Fall erfolgt ein Großteil der Entwicklungsaktivitäten produkt- bzw. produktgruppenspezifisch. Lediglich Gleichteile werden produktübergreifend zusammengefasst und im Sinne eines Moduls eigenständig geplant, entwickelt, getestet und freigegeben.

Die Gliederung von Produkten und Produktfamilien in geeignete Release-Einheiten ist erfolgsentscheidend für die Release-Planung und insb. bei komplexen Produktfamilien mit hoher räumlicher Varianz eine Herausforderung. Aus diesem Grund widmet sich diesem Thema eine Vielzahl aktueller Forschungsarbeiten (z.B. [SAA13], [MBE+14]).

2) Release-Typen: Im Beispiel in Bild 4-3 sind in regelmäßigen Abständen unterschiedliche Release-Typen eingeplant, die sich in drei Klassen unterscheiden: Major-Releases, Minor-Releases und Sofortmaßnahmen. Diese Unterscheidung dient der **differenzierten Umsetzung** von Änderungen. So werden im Beispiel jährlich sogenannte *Major-Releases* eingeplant, mit denen sowohl kundenrelevante Änderungen⁵³ als auch rein technische Verbesserungsmaßnahmen umgesetzt werden. Major-Releases werden auch als Facelift bezeichnet und intensiv vermarktet. Bekannt sind bspw. die Facelifts der Automobilindustrie, mit denen nach ca. der Hälfte des Modelllebenszyklus eine grundlegende Aufwertung des Fahrzeugs erfolgt. *Minor-Releases* erfolgen hingegen zeitlich versetzt. Sie umfassen hauptsächlich technische Verbesserungsmaßnahmen ohne Kundenrelevanz und werden nicht vermarktet. Die Umsetzung dringlicher Änderungen erfolgt im Sinne eines Bypasses durch sogenannte *Sofortmaßnahmen*. Hierdurch wird der eigentliche Gedanke der Release-Planung umgangen und die Änderung sofort umgesetzt und freigegeben. Die

⁵³ Im Rahmen dieser Arbeit fassen kundenrelevante Änderungen sämtliche Änderungen zusammen, die kundenerlebbare Produktmerkmale oder den Preis des Produkts beeinflussen und als Teil der Marketingstrategie des Unternehmens vermarktet werden. Dies sind neben funktionalen Änderungen z.B. auch gestaltorientierte Änderungen mit Auswirkungen auf das Produktdesign.

Zuordnung von Änderungen zu Release-Typen erfolgt mittels eines Klassifikationsschemas entlang der Eigenschaften einer Änderung. Das Klassifikationsschema und die damit verbundenen Release-Typen sind unternehmensindividuell im Rahmen der strategischen Release-Planung festzulegen. Hierbei sind Unternehmens- und Produktspezifika zu berücksichtigen. Einen Überblick über alternative Release-Typen gibt Anhang A3.4. Beispiele sind neben den genannten Major- und Minor-Releases sogenannte Modul-Releases, Plattform-Releases und Prozess-Releases.

Der differenzierten Umsetzung von Änderungen mittels unterschiedlicher Release-Typen steht die **universelle Umsetzung** von Änderungen gegenüber. In diesem Fall werden die Releases nicht in alternative Release-Typen unterschieden. Stattdessen wird für jedes Release individuell entschieden, welche Arten von Änderungen umgesetzt werden. Dies schafft ein gewisses Maß an Flexibilität insb. bei der Umsetzung von kundenrelevanten Änderungen und damit bei der gezielten Produktwertsteigerung.

3) Release-Timing: Dieser Gestaltungsfaktor definiert die Stringenz, mit der die Veröffentlichung eines neuen Releases zu erfolgen hat. Hierbei können grundsätzlich zeitliche, inhaltliche und hybride Gesichtspunkte unterschieden werden. Beim **zeitbasierten Release-Timing** steht die Einhaltung des Zeitpunkts der Marktveröffentlichung ungeachtet des Inhalts an erster Stelle. Im Zweifelsfall werden für die Einhaltung des Zieltermins ausgewählte Änderungen auf spätere Releases verschoben. Diese Stringenz in der Planung vereinfacht die Synchronisation zwischen verschiedenen Release-Einheiten. Demgegenüber fokussiert das **inhaltsbasierte Release-Timing** die Unterbringung eines gewissen Änderungsumfangs innerhalb eines Releases ohne zeitliche Relevanz. Hierbei werden Planabweichungen für die Realisierung zuvor definierter Änderungen und Produkt-Features explizit in Kauf genommen. Das **hybride Release-Timing** gewichtet beide Extremformen gleichermaßen [Wri12, S. 7f.].

4) Release-Rhythmus: Unabhängig vom Release-Typ erfolgt die Veröffentlichung neuer Releases entweder in zyklischen Intervallen oder aber fallbezogen. Im Falle mehrerer Release-Einheiten ist auch eine hybride Form möglich. Bild 4-5 stellt die drei Ausprägungen vergleichend gegenüber.

Im Fall der Ausprägung **zyklische Intervalle** erfolgt die Veröffentlichung eines Releases in regelmäßigen Abständen. Diese beschränken sich jedoch nicht auf eine Intervalllänge. Es ist ebenso möglich, dass verschiedene Änderungszyklen für unterschiedliche Release-Einheiten (Module, Produkte, Produktgruppen) geplant werden. Ein Hilfsmittel zur Planung modulspezifischer Release-Zyklen schlagen z.B. SCHUH ET AL. und MAURER ET AL. vor (vgl. Abschnitt 3.5.2 und 3.5.3.). Jedem Modul wird in Abhängigkeit von einem Flexibilitätsindikator ein Release-Intervall zugewiesen. Die Release-Intervalle entsprechen jeweils Vielfachen voneinander, so dass sich zu bestimmten Zeitpunkten Synchronisationspunkte ergeben. Diese werden genutzt, um modul- bzw. produktübergreifende Ände-

rungen umzusetzen [SAA13], [MBE+14]. Grundsätzlich ist zu bedenken, dass durch Sofortmaßnahmen von zyklischen Intervallen abgewichen werden kann. Dringende Änderungen können so auch außerhalb der geplanten Intervalle umgesetzt werden.

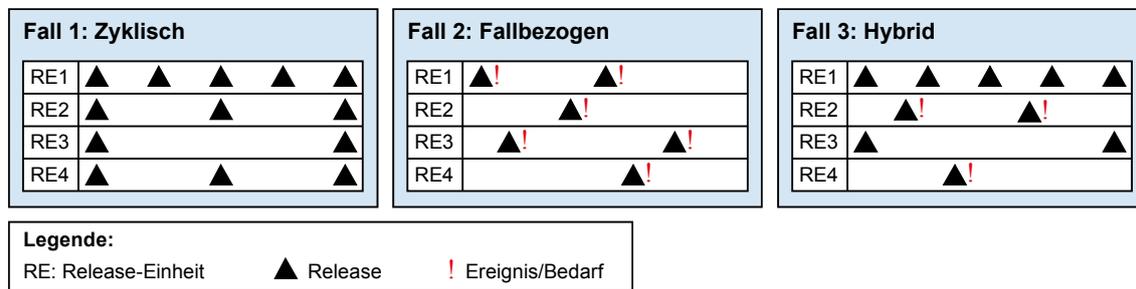


Bild 4-5: Gegenüberstellung der drei Ausprägungen für die Veröffentlichung neuer Releases

Der **fallbezogene Rhythmus** hingegen führt zur Veröffentlichung eines Release zu unregelmäßigen Zeitpunkten. Dabei sorgen verschiedene extern oder intern getriebene Ereignisse für ein neues Release. Als Beispiele sind unregelmäßig stattfindende Messen, Marktaktionen durch den Wettbewerber, saisonale Besonderheiten oder Lieferantenwechsel zu nennen. Fallbezogene Veröffentlichungen werden somit sowohl aktiv als auch reaktiv hervorgerufen. Der Planer entscheidet in diesem Fall, welche Ereignisse ein neues Release verursachen.

Erfolgt die Release-Planung für mehrere Release-Einheiten, ist auch eine **hybride Form** möglich. In diesem Fall erfolgt die Planung für ausgewählte Release-Einheiten zyklisch, während die übrigen fallbezogen veröffentlicht werden. Die Auswahl orientiert sich z.B. an der Häufigkeit, mit der eine bestimmte Release-Einheit geändert werden muss. Je häufiger Änderungen vorliegen, desto eher bietet sich eine zyklische Planung an.

5) Release-Frequenz: Dieser Gestaltungsfaktor gibt Aufschluss über die Häufigkeit, mit der neue Releases veröffentlicht werden. Auf diesem Wege wird automatisch der Lebenszyklus jedes einzelnen Release bestimmt. Grundsätzlich ist die Frequenz für verschiedene Release-Typen insb. für kundenrelevante und nicht kundenrelevante Releases zu unterscheiden. Die Ausprägung in **selten**, **mittel** und **häufig** ist bewusst auf einem 3-stufigen, qualitativen Niveau gehalten; sie muss je nach Branche und Produkt angepasst werden. Während die Frequenz kundenrelevanter Releases bei Smartphones bspw. bei einem Jahr liegt, ist diese gerade bei industriellen Produkten weitaus geringer. Auch die Frequenz rein technischer Releases hängt stark vom Produkttyp ab. Hier entscheidet insb. die Komplexität des Systems über die Änderungshäufigkeit und damit über die Anzahl notwendiger Releases. Die Release-Frequenz ist eng mit dem Rhythmus der Veröffentlichung abzustimmen, insb. im Fall zyklischer Intervalle.

6) Release-Umfang: Für jedes Release wird ein Änderungsumfang geplant, der die Anzahl und den Aufwand der umzusetzenden Änderungen unter Berücksichtigung der verfügbaren Ressourcen (z.B. Entwicklungsdauer, Personalverfügbarkeit) festlegt. Hierbei

können zwei Fälle unterschieden werden: ein definierter sowie ein variabler Release-Umfang. Im Fall eines **definierten Release-Umfangs** steht ein festes Budget für die Umsetzung des Release zur Verfügung, das bereits im Vorfeld definiert wird (z.B. im Rahmen der strategischen Release-Planung). Unter Berücksichtigung des verfügbaren Budgets ist dann der Änderungsumfang für das jeweilige Release festzulegen. Typischerweise werden verschiedenen Release-Typen unterschiedliche Budgets zugewiesen, so dass für die Realisierung eines Major-Releases in der Regel mehr Ressourcen zur Verfügung stehen als für die eines Minor-Release. Im Falle **variabler Release-Umfänge** wird das Budget für jedes Release individuell festgelegt. Restriktionen ergeben sich hierbei lediglich durch die unternehmensweit verfügbaren Ressourcen.

7) Planungshorizont: Der oben beschriebene Release-Plan weist einen Planungshorizont von knapp sechs Jahren auf. Im Kontext der Release-Planung gibt der Planungshorizont den Zeitraum vor, für den Release-Zeitpunkte und damit verbundene Release-Typen vorausgeplant und fest definiert werden. Die Festlegung dieses Zeitraums orientiert sich an der Dauer des Produktlebenszyklus, der wiederum stark von der Art des Produkts und der Branche abhängt. Bei der Festlegung des Planungshorizonts werden zwei mögliche Ausprägungen unterschieden. Im Falle eines **fixierten Planungshorizonts** umfasst dieser den gesamten Produktlebenszyklus, für den bereits zu Beginn sämtliche Releases und die damit verbundenen Release-Zeitpunkte definiert werden. Dies bietet sich für Produkte mit sehr kurzen Produktlebenszyklen an (ca. zwei bis vier Jahre) oder ergibt sich durch einen hohen Grad an Standardisierung in der Release-Planung (zyklische Veröffentlichung). Insbesondere bei Produkten mit einem zeitlich nicht definierten Produktlebenszyklus erfolgt die Planung **rollierend**. In diesem Fall werden die Releases und die damit verbundenen Release-Zeitpunkte bis zu einem definierten Bezugspunkt festgelegt. Der Bezugspunkt variiert je nach Release-Typ. Hierbei kann es sich z.B. um das übernächste kundenrelevante Release handeln.

4.3.2 Ordnungsschema alternativer Release-Strategien

Die Strukturierung der Gestaltungsfaktoren und der damit verbundenen Ausprägungen ergibt das in Bild 4-6 dargestellte Ordnungsschema. Die Kombination einer oder mehrerer Ausprägungen je Gestaltungsfaktor führt zu einer spezifischen Release-Strategie, die über ein charakteristisches Profil gezielt Vorgaben für die initiale Strukturierung des Release-Plans macht. Prinzipiell ist die Kombination aller Ausprägungen möglich, so dass eine hohe theoretische Anzahl alternativer Release-Strategien denkbar ist. Die Ausprägungen bedingen sich in Teilen jedoch gegenseitig. So bietet sich im Falle einer modulorientierten Release-Planung (Release-Einheit) eine zyklische Veröffentlichung der Re-

leases (Release-Rhythmus) an, um die Synchronisation bei modulübergreifenden Änderungen zu systematisieren. Die Analyse dieser Abhängigkeiten⁵⁴ liefert drei Grundtypen von Release-Strategien, die im Folgenden als Norm-Release-Strategien bezeichnet werden (Bild 4-6). Diese werden im Folgenden kurz beschrieben.

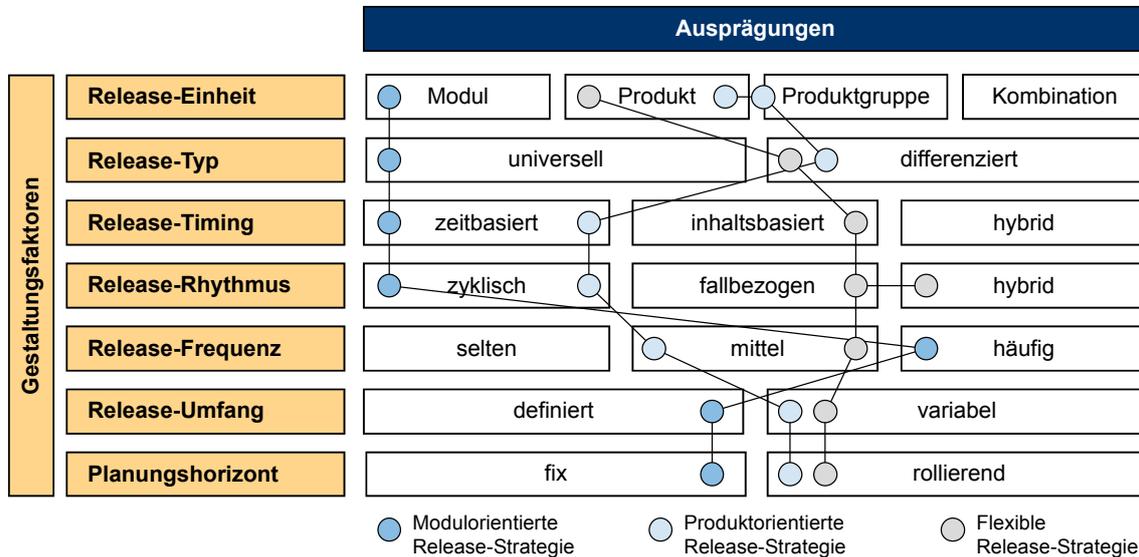


Bild 4-6: Ordnungsschema alternativer Release-Strategien

I) Die modulatorientierte Release-Strategie zeichnet sich durch einen hohen Grad an Standardisierung über einen fest definierten Planungshorizont aus: Releases werden für einzelne Module zeitbasiert, zyklisch und mit einem definierten Umfang geplant. Die Einschränkung der zeitlichen Freiheitsgrade resultiert dabei aus der Notwendigkeit, modulübergreifende Änderungen bei möglichst geringem Koordinationsaufwand zu synchronisieren. Ein gewisses Maß an Flexibilität ergibt sich durch universelle Releases, die im Hinblick auf Release-Inhalte keine Einschränkungen vornehmen. Die modulatorientierte Release-Strategie bietet sich insb. für variantenreiche (Baukasten-)Systeme mit einem hohen Änderungsaufkommen sowie einem hohen Grad an Kommunalität an. Sie fokussiert die Ausschöpfung interner entwicklungsstrategischer Nutzenpotentiale (vgl. Abschnitt 2.4.3). Ziel sind eine synergetische Umsetzung von Änderungen und eine systematische Produktprogrammpflege.

II) Im Kontext der produktorientierten Release-Strategie entsprechen die Release-Einheiten entweder einer spezifischen Produktvariante oder einer Gruppe ähnlicher Produktvarianten (Produktgruppe). Ähnlich wie bei der modulatorientierten Release-Strategie wird durch eine zeitbasierte und zyklische Veröffentlichung der Releases ein gewisser Grad an Standardisierung in der zeitlichen Planung erreicht. Zusammen mit einer Standardisie-

⁵⁴ Als Hilfestellung bei der Analyse der Abhängigkeiten bietet sich die Konsistenzanalyse an [GP14, S. 62ff.]. Hierbei werden die alternativen Ausprägungen paarweise auf Stimmigkeit überprüft. Über statistische Verfahren lassen sich schließlich konsistente Kombinationen von Ausprägungen ableiten.

nung der Release-Inhalte mittels differenzierter Release-Typen erlaubt dies die systematische Umsetzung von Änderungen bei einem hohen Änderungsaufkommen. Gleichzeitig wird durch die Standardisierung der Koordinationsaufwand für produktübergreifende Änderungen und damit für die Synchronisation von Gleichteilen innerhalb einer Produktfamilie reduziert. Dies ist allerdings nur bis zu einem gewissen Grad an Kommunalität praktikabel, so dass sich die produktorientierte Release-Strategie in erster Linie für Produktfamilien bzw. Produktprogramme mit wenigen Gleichteilen eignet. Auch hier steht die Ausschöpfung interner entwicklungsstrategischer Nutzenpotentiale im Vordergrund. Diese drücken sich insb. in einer gleichmäßigen Auslastung der Entwicklungsressourcen sowie einer systematischen Produktprogrammpflege aus. Gleichzeitig erlaubt die differenzierte Umsetzungsplanung verschiedener Release-Typen die Erschließung markt- und wettbewerbsstrategischer Nutzenpotentiale durch eine systematische Produktwertsteigerung.

III) Die flexible Release-Strategie dient der Umsetzungsplanung von Änderungen für ein spezifisches Produkt. Durch eine inhaltsbasierte und fallbezogene Veröffentlichung der Releases zeichnet sich diese Strategie durch ein hohes Maß an Flexibilität aus. Lediglich durch die Differenzierung von Release-Typen erfolgt eine Standardisierung des Release-Planungsprozesses im Hinblick auf die Release-Inhalte. Die Strategie eignet sich hauptsächlich für Produkte mit geringer bis gar keiner räumlichen Varianz und entsprechend wenigen Gleichteilen. Im Fokus steht die Erschließung markt- und wettbewerbsstrategischer Nutzenpotentiale. Dies ermöglicht eine systematische Produktwertsteigerung bei gleichzeitiger Erschließung von Änderungssynergien durch die gezielte Bündelung von Änderungen zu Releases.

4.3.3 Auswahl der geeigneten Release-Strategie

Ziel ist die Auswahl einer für das jeweilige Produkt und die Unternehmenssituation geeigneten Release-Strategie. Dabei sind zwei Fragestellungen von Bedeutung: 1) Wie hoch ist die Wahrscheinlichkeit von Änderungen aufgrund externer und interner Unsicherheiten (dynamische Komplexität)? 2) Wie hoch ist die Komplexität des Zielsystems, für das die Umsetzungsplanung von Änderungen im Rahmen der Release-Planung erfolgen soll (strukturelle Komplexität)?⁵⁵ Zur Einordnung der spezifischen Situation eignet sich eine Bewertungssystematik im Sinne eines Portfolioansatzes. Im Portfolio werden die Änderungswahrscheinlichkeit und die Komplexität des Zielsystems bewertet. Die Einteilung des Portfolios in Felder liefert schließlich Hinweise auf die empfohlene Release-Strategie (Bild 4-7). Im Folgenden werden die beiden Bewertungsdimensionen kurz beschrieben.⁵⁶

⁵⁵ Zur Bedeutung von dynamischer und struktureller Komplexität vgl. Abschnitt 2.1.3.

⁵⁶ Die exemplarische Bewertung eines Produkts anhand gewichteter Bewertungskriterien erfolgt in Abschnitt 5.1.1 im Rahmen der Anwendung der Systematik.

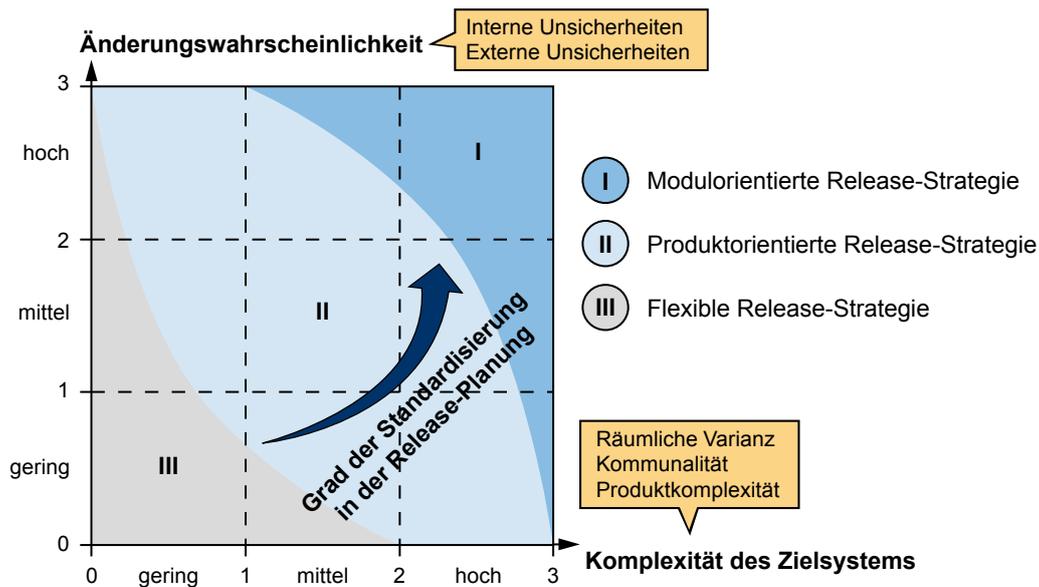


Bild 4-7: Portfolio zur Auswahl der geeigneten Release-Strategie

Änderungswahrscheinlichkeit: Entscheidend für die Auswahl der Release-Strategie ist die Wahrscheinlichkeit, mit der sich Änderungen über den Produktlebenszyklus ergeben. Je höher dieser Faktor ausfällt, desto systematischer und standardisierter muss die Umsetzungsplanung von Änderungen erfolgen. In diesem Fall bieten sich in Abhängigkeit von der Komplexität des Zielsystems insb. die Release-Strategien I und II an. Die Änderungswahrscheinlichkeit wird maßgeblich durch externe und interne Unsicherheiten beeinflusst (vgl. Abschnitt 2.1.3). Zur Bewertung der **externen Unsicherheiten** werden Kriterien aus den Gruppen Markt- und Kundenumfeld (Kundendynamik, Innovationsanspruch), Wettbewerb (Wettbewerbsintensität, Finanzkraft des Wettbewerbs, Innovationsgeschwindigkeit) und Gesetzgebung (Beständigkeit der Gesetzgebung) herangezogen. Die Bewertung der **internen Unsicherheiten** ergibt sich aus dem Produktreifegrad, der Dynamik des Lieferantenumfelds und der organisatorischen Komplexität.

Komplexität des Zielsystems: Neben der Änderungsintensität beeinflusst die Beschaffenheit des Zielsystems die Auswahl der geeigneten Release-Strategie. Auch hier gilt: Je komplexer das Zielsystem, desto höher die erforderliche Standardisierung im Release-Planungsprozess. Drei Kriterien beeinflussen die Komplexität des Zielsystems maßgeblich: räumliche Varianz, Kommunalität und Produktkomplexität. Die **räumliche Varianz** berücksichtigt die Bandbreite der Produkte, für die die Release-Planung erfolgen soll. Diese bewegt sich zwischen zwei Extremen: 1) Die Release-Planung erfolgt für ein einzelnes Produkt (keine räumliche Varianz), oder 2) die Release-Planung erfolgt für eine gesamte Produktfamilie (hohe räumliche Varianz). Je höher die räumliche Varianz, desto höher fällt die Komplexität des Zielsystems aus. Gleiches gilt für das Kriterium **Kommunalität**. Dieses gibt Aufschluss über die Anzahl an Gleichteilen und damit über die Häufigkeit produktübergreifender Änderungen. Die **Produktkomplexität** bewertet die Komplexität des Kernprodukts in Abhängigkeit von Anzahl und Vielfalt der Module bzw. Systemelemente. Die Bewertung der zwei Dimensionen Änderungswahrscheinlichkeit

und Komplexität des Zielsystems für ein spezifisches Produkt bzw. Produktprogramm erlaubt schließlich die Auswahl einer geeigneten Release-Strategie. Diese ist im weiteren Planungsprozess zu einem initialen Release-Plan zu konkretisieren.

4.4 Vorgehensmodelle für die Release-Planung

Wesentlicher Bestandteil der *Systematik zur Release-Planung intelligenter technischer Systeme* sind drei **Vorgehensmodelle**, die die Tätigkeiten entlang der drei Ebenen des Release-Planungsprozesses beschreiben und strukturieren. Die Vorgehensmodelle definieren die Reihenfolge der Tätigkeiten, integrieren die dafür notwendigen Hilfsmittel und geben einen Überblick über die planungsrelevanten Ergebnisse. Es handelt sich um eine sequentielle und idealtypische Darstellung in Phasen und Meilensteinen, die Iterationen erfordert und Rücksprünge in frühere Phasen zulässt. Im Folgenden werden die Vorgehensmodelle im Überblick beschrieben. So beschreibt **Abschnitt 4.4.1** das Vorgehen in der strategischen Release-Planung, **Abschnitt 4.4.2** die taktische Release-Planung und **Abschnitt 4.4.3** die operative Release-Planung. In Kapitel 5 werden die Vorgehensmodelle exemplarisch angewendet.

4.4.1 Strategische Release-Planung

Ziel der **strategischen Release-Planung** ist ein Release-Plan, der sämtliche Releases entlang des Planungshorizonts terminiert. Dazu werden, ausgehend von einer unternehmens- und produktspezifischen Release-Strategie, notwendige Release-Typen und Release-Zeitpunkte festgelegt. Das Vorgehen ist eng mit der strategischen Produktplanung (insb. der Erarbeitung der Produktstrategie) abzustimmen und gliedert sich in drei Phasen (Bild 4-8).

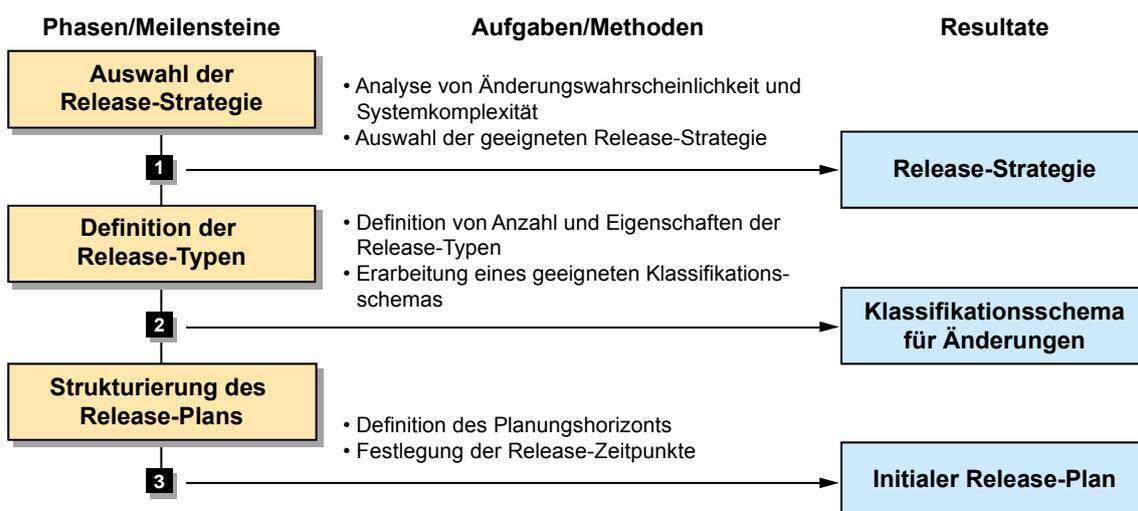


Bild 4-8: Vorgehensmodell für die strategische Release-Planung

Das Vorgehen startet in Phase 1 mit der **Auswahl der Release-Strategie**. Zur Auswahl werden Änderungswahrscheinlichkeit und Komplexität des technischen Systems bestimmt. Entsprechend des Ordnungsschemas aus Abschnitt 4.3.2 erlaubt dies Rückschlüsse auf die unternehmens- und produktadäquate Release-Strategie. Diese gibt letztlich wesentliche Strukturmerkmale für die weitere Erarbeitung des Release-Plans vor (vgl. exemplarische Anwendung in Abschnitt 5.1.1).

Aufbauend auf der ausgewählten Release-Strategie dient Phase 2 der **Definition der Release-Typen** und damit der Konkretisierung des gleichnamigen Gestaltungsfaktors. Ausgehend von einer Analyse bisheriger Änderungen ist festzulegen, wie viele Release-Typen prinzipiell unterschieden werden sollen; eine Möglichkeit ist ein 2-stufiges Release-Konzept, das Release-Typen in Major- und Minor-Releases unterscheidet. Ferner sind für jeden Release-Typ die zugehörigen Eigenschaften *Änderungstyp* und *Verfügbarkeit* festzulegen. Der Änderungstyp definiert die Änderungen, die innerhalb eines Release-Typs umgesetzt werden. Die Verfügbarkeit legt fest, mit welcher Frequenz ein bestimmter Release-Typ veröffentlicht wird (z.B. halbjährlich, jährlich). Darauf aufbauend wird ein Klassifikationsschema erarbeitet, das im Rahmen der taktischen Release-Planung die Zuordnung von Änderungen zu einem bestimmten Release-Typ ermöglicht (vgl. Abschnitt 5.1.2).

Die dritte Phase **Strukturierung des Release-Plans** greift die Ergebnisse der vorherigen Phasen auf und erfolgt in enger Abstimmung mit der Produktstrategie. Der Release-Plan wird in zwei Schritten konkretisiert: In einem ersten Schritt wird der Planungshorizont definiert. Dieser gibt den Zeitraum vor, für den die Release-Zeitpunkte bereits konkret geplant werden. Im zweiten Schritt werden die Releases terminiert. Grundlage sind die Analyse einer übergeordneten Produkt- und Technologie-Roadmap, eine Wettbewerbsanalyse und die Berücksichtigung branchen- und marktspezifischer Termine. Das **Resultat** ist ein auf unternehmens- und produktstrategische Überlegungen abgestimmter Release-Plan. Hierbei handelt es sich um „lebendes“ Dokument, das in regelmäßigen Abständen aktualisiert werden muss. Anders als die Phasen 1 und 2 ist die Phase 3 *Strukturierung des Release-Plans* daher mindestens in einem jährlichen Turnus zu durchlaufen (vgl. Abschnitt 5.1.3).

4.4.2 Taktische Release-Planung

Die **taktische Release-Planung** erfolgt für eine einzelne Änderung. Ziel ist die Terminierung der Änderungsumsetzung durch die Zuordnung zu einem Release. Dazu ist jede Änderung in einem systematischen Bewertungsprozess hinsichtlich ihrer technischen Auswirkungen und Abhängigkeiten sowie weiterer Kriterien wie Dringlichkeit und Wichtigkeit zu analysieren. Die Änderungsanalyse erfolgt kontinuierlich und lässt sich in den typischen Änderungsprozess eines Unternehmens einbetten. In Bild 4-9 ist dies stellvertretend am Änderungsprozess nach JARRETT ET AL. dargestellt.

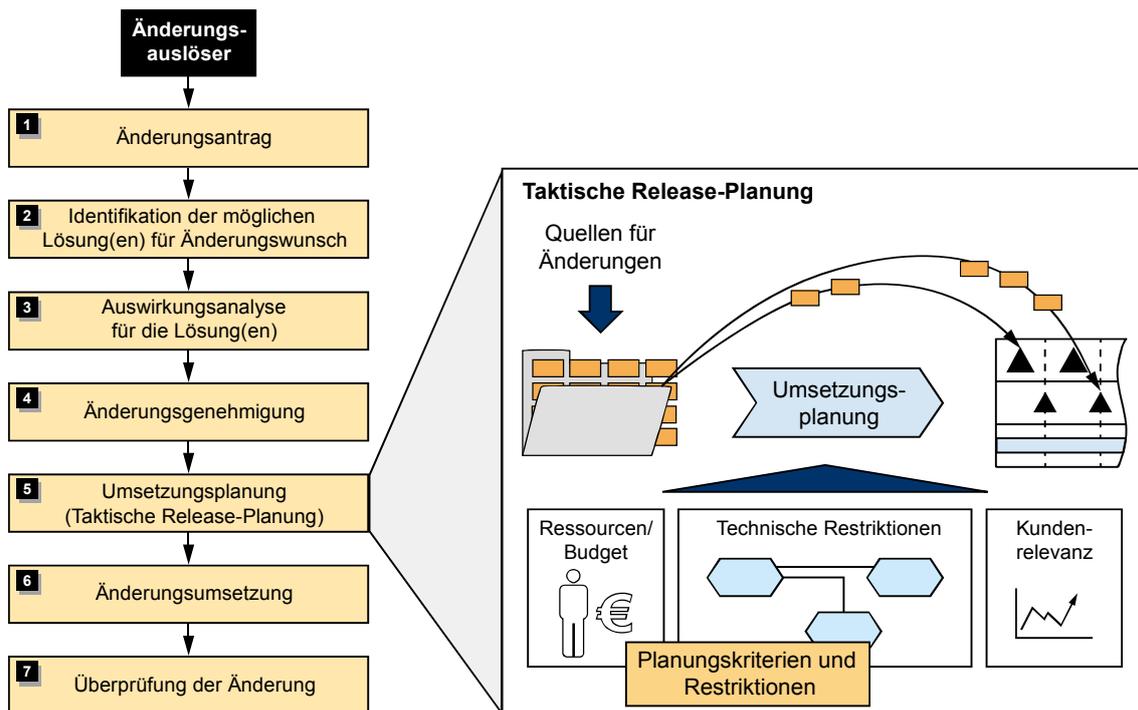


Bild 4-9: Einordnung der taktischen Release-Planung in den Änderungsprozess nach JARRETT ET AL.

Die Phasen 1 bis 4 des Änderungsprozesses dienen der systematischen Erfassung einer Änderung und der Beantwortung der Frage, ob eine Änderung überhaupt umgesetzt wird. Sie sind nicht Teil der taktischen Release-Planung und stehen damit auch nicht im Fokus dieser Arbeit. Vielmehr wird der Änderungsprozess im Rahmen der Release-Planung um die Phase 5 – **Umsetzungsplanung** – ergänzt. In dieser Phase werden sämtliche Informationen über eine Änderung zusammengetragen, die deren Umsetzungszeitpunkt beeinflussen (z.B. Kundenrelevanz, technische Restriktionen). Je nach unternehmensindividueller Gestaltung des Änderungsprozesses werden einige dieser Informationen bereits in früheren Analyseschritten erhoben (z.B. Phase 3 – Auswirkungsanalyse). Für andere Informationen sind weitere Analyseschritte erforderlich. Losgelöst von einem bestehenden Änderungsprozess zeigt Bild 4-10 das für die Erhebung aller planungsrelevanten Informationen erforderliche Vorgehen. Dieses gliedert sich in drei Phasen.

Basierend auf dem Systemmodell (vgl. Abschnitt 4.5) erfolgt in Phase 1 die **technische Änderungsanalyse**. Diese liefert in drei Schritten ein disziplinübergreifendes Verständnis über die Charakteristika der Änderung und die mit der Umsetzung verbundenen technischen Auswirkungen und Abhängigkeiten. Im ersten Schritt dient eine initiale Auswirkungsanalyse der Identifikation des primär betroffenen Systemelements. Davon ausgehend wird in einem zweiten Schritt die Änderungsf Fortpflanzung analysiert, die Aufschluss über weitere, von der Änderung betroffene Systemelemente liefert. In einem abschließenden dritten Schritt werden Änderungsabhängigkeiten ermittelt. Hierbei ist zu überprüfen, inwieweit sich Änderungen gegenseitig bedingen bzw. beeinflussen. So setzen sich einige Änderungen technisch bedingt voraus und *müssen* gemeinsam innerhalb

eines Release umgesetzt werden; andere Änderungen bieten lediglich Synergiepotentiale in der gemeinsamen Umsetzung. Änderungsabhängigkeiten beeinflussen somit die Umsetzungsabfolge und sind daher als Restriktion bei der Zuordnung einer Änderung zu einem Release zu berücksichtigen. Grundsätzlich können drei Typen von Abhängigkeiten unterschieden werden, auf die in Abschnitt 4.5.3 genauer eingegangen wird: technische Abhängigkeiten sowie Abhängigkeiten in Nutzen und Aufwand.

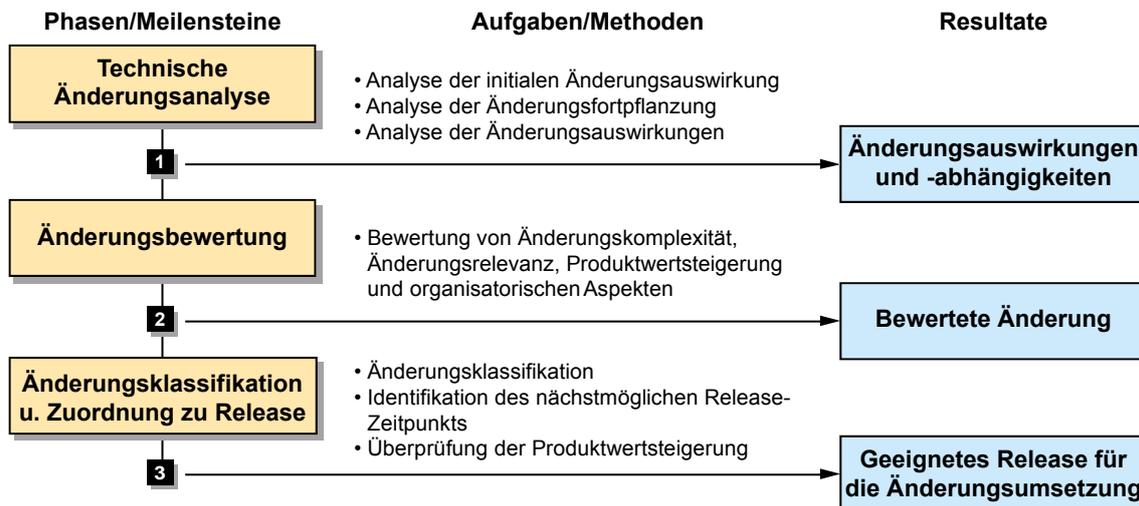


Bild 4-10: Vorgehensmodell für die taktische Release-Planung

In Phase 2 schließt sich die **Änderungsbewertung** an. Hierbei wird die Änderung unter Berücksichtigung von vier Aspekten bewertet: Änderungskomplexität, Änderungsrelevanz, Produktwertsteigerung und Organisatorisches. Die Änderungsbewertung führt sämtliche planungsrelevante Informationen zusammen und dient als Basis für die Änderungsklassifizierung in der darauffolgenden Phase.

In der letzten Phase – Phase 3 – erfolgen die **Änderungsklassifizierung** und die eigentliche **Zuordnung zu einem Release**. Grundlage hierfür ist das in der strategischen Release-Planung erarbeitete Klassifikationsschema, das eine Zuordnung der Änderung zu einem Release-Typ ermöglicht. Im Abgleich mit dem Release-Plan kann darauf aufbauend der nächstmögliche Release-Zeitpunkt identifiziert werden. Im Falle kundenrelevanter Änderungen ist eine gesonderte Überprüfung der mit einem Release verbundenen Produktwertsteigerung erforderlich. Es ist sicherzustellen, dass ein Release ein ausgewogenes Verhältnis an kundenrelevanten Änderungen umfasst. Das heißt, dass im Sinne einer kontinuierlichen Produktwertsteigerung ein Mindestmaß an kundenrelevanten Änderungen erreicht wird, gleichzeitig aber noch genug technische Neuerungen für Folge-Releases verfügbar sind. Im **Ergebnis** von Phase 3 ist jede Änderung einem zur Umsetzung geeigneten Release zugeordnet.

4.4.3 Operative Release-Planung

Die **operative Release-Planung** fokussiert das nächste Release. Ziel ist die abschließende Bündelung der Änderungen, die in diesem Release umgesetzt werden sollen. Dies erfolgt unter Berücksichtigung operativer Rahmenbedingungen (z.B. Verfügbarkeit von Ressourcen), die die Anzahl der umsetzbaren Änderungen einschränken. Bspw. werden im Falle unzureichender Ressourcen Änderungen von geringer Priorität zurück in den Änderungs-Pool gespielt und erst in einem der nächsten Releases umgesetzt. Das Vorgehen zur operativen Release-Planung gliedert sich in vier Phasen (Bild 4-11).

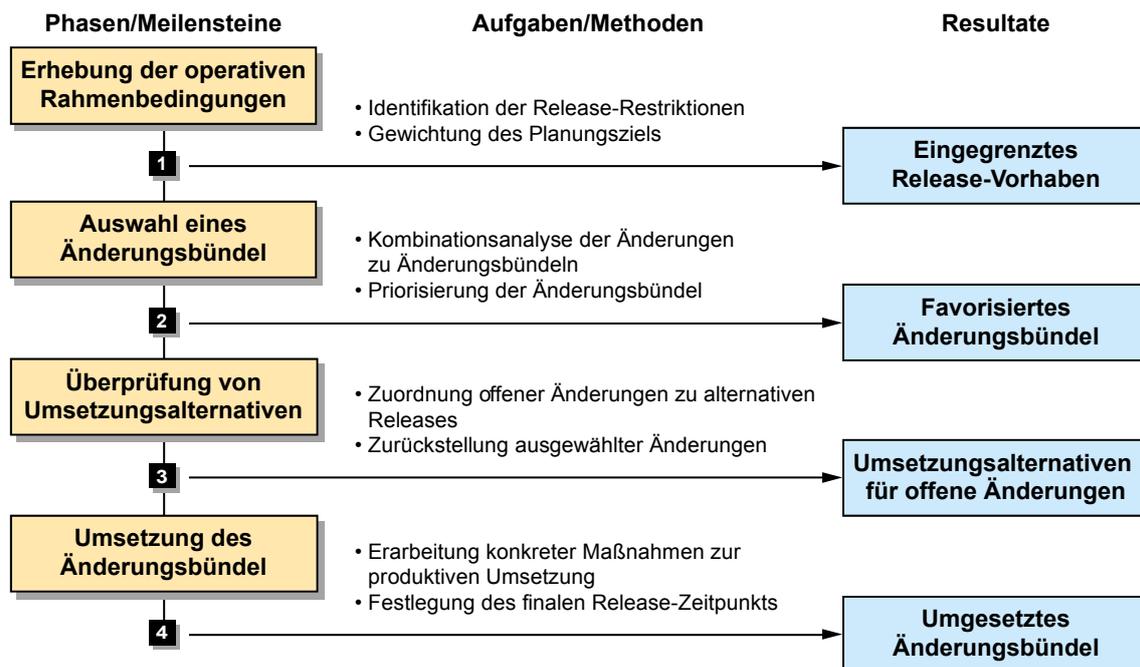


Bild 4-11: Vorgehensmodell für die operative Release-Planung

Phase 1 dient der **Erhebung der operativen Rahmenbedingungen** für das Release-Vorhaben. Im Fokus stehen die Identifikation von Release-Restriktionen sowie die Definition des Planungsziels. Release-Restriktionen geben Aufschluss über die verfügbaren Personalkapazitäten und die geplante Umsetzungsdauer für das jeweilige Release. Das Planungsziel gibt wesentliche Randbedingungen für die Bündelung der Änderungen zu einem Release vor. Je nach Zielsetzung kann der Fokus bspw. auf der Umsetzung der dringlichsten Änderungen oder aber auf der Erschließung möglichst vieler Synergiepotentiale in der Änderungsumsetzung liegen. Mit Abschluss der Phase 1 ist das Release-Vorhaben hinsichtlich seiner operativen Rahmenbedingungen eingegrenzt.

In Phase 2 folgt die **Auswahl eines Änderungsbündels**. Ausgangspunkt hierfür sind die Änderungen, die dem jeweiligen Release im Rahmen der taktischen Release-Planung zugeordnet wurden. Diese Änderungen werden unter Berücksichtigung der operativen Rahmenbedingungen zu Releases gebündelt, die sämtliche Restriktionen im Hinblick auf verfügbare Ressourcen und Umsetzungsdauer erfüllen. Es ergeben sich alternative Kombinationen von Änderungen – sogenannte Änderungsbündel –, die entsprechend des zuvor

festgelegten Planungsziels bewertet und anhand eines Release-Werts priorisiert werden. Es kristallisiert sich das favorisierte Änderungsbündel heraus.

Im Fokus von Phase 3 – der **Überprüfung von Umsetzungsalternativen** – stehen Änderungen, die aufgrund begrenzter Ressourcen und technischer Abhängigkeiten im favorisierten Änderungsbündel nicht berücksichtigt werden können. Für jede dieser Änderungen wird das nächstmögliche Release ermittelt und entschieden, ob eine Zurückstellung der Änderungsumsetzung unter Berücksichtigung aller Stakeholder akzeptabel ist. Im positiven Fall wird die Umsetzung der Änderung auf eines der nächsten Releases verschoben. Ist die Änderungsumsetzung hingegen zwingend erforderlich, muss Phase 2 erneut durchlaufen werden. Es werden erneut Änderungsbündel ermittelt, die die jeweilige Änderung zwingend enthalten.

Mit Phase 4 – der **Umsetzung des Änderungsbündels** – schließt die operative Release-Planung für das jeweilige Release ab. An der Schnittstelle zum Projektmanagement werden konkrete Maßnahmen zur produktiven Umsetzung des Releases festgelegt und der finale Release-Zeitpunkt bestimmt. Im **Ergebnis** liegen sämtliche Informationen zur Entwicklung und Einführung des Releases und damit des zuvor bestimmten Änderungsbündels vor.

4.5 Das Systemmodell als Kooperationskern in der Release-Planung

Aufgabe der Release-Planung ist die Umsetzungsplanung von neuen Produkt-Features und technischen Änderungen. Dies erfordert die Expertise verschiedener Fachleute, die jeweils unterschiedliche Sichten auf die resultierenden Änderungen und die damit verbundenen Auswirkungen haben. So kann der Softwareentwickler bspw. die Auswirkungen auf die Software und den damit verbundenen Quellcode abschätzen, bringt aber nur wenig Verständnis für die Auswirkungen auf Mechanik und Regelungstechnik auf. Der Produktmanager hingegen beurteilt die Änderungen aus Sicht von Markt und Kunde, interessiert sich jedoch nur eingeschränkt für die technischen Details. Für eine effektive und effiziente Release-Planung gilt es, die verschiedenen Sichten der Fachleute zu vereinen und zu planungsrelevanten Informationen zu verdichten. Dies erfordert die Kommunikation und Kooperation zwischen allen beteiligten Fachleuten.

Zur Unterstützung der Kommunikation und Kooperation im Release-Planungsprozess bietet sich im Sinne des MBSE-Paradigmas ein Systemmodell als gemeinsamer Kooperationskern an (vgl. Abschnitt 2.5.1). Das **Systemmodell** dient über den gesamten Planungsprozess hinweg als Informationsbasis, die kontinuierlich erweitert und gepflegt wird. Zur verbesserten Unterstützung der Release-Planung wird in **Abschnitt 4.5.1** eine erweiterte Systemmodellierung für die Release-Planung beschrieben. Davon ausgehend werden in **Abschnitt 4.5.2** und **Abschnitt 4.5.3** zwei relevante Anwendungsfälle beschrieben, die den Einsatz der erweiterten Systemmodellierung im Kontext der Release-Planung verdeutlichen.

4.5.1 Erweiterte Systemmodellierung für die Release-Planung

Im Rahmen dieser Arbeit wird davon ausgegangen, dass im Sinne des MBSE sowohl der Entwicklungsprozess als auch der gesamte Produktlebenszyklus durch ein Systemmodell begleitet werden. Das **Systemmodell** umfasst sämtliche fachdisziplinübergreifend relevanten Informationen und dient als Synchronisationspunkt für alle fachdisziplinspezifischen Entwicklungsaktivitäten. Es ist nicht nur Ausgangspunkt für den Systementwurf in der Neuproduktentwicklung, sondern auch Ausgangspunkt für die Planung und Analyse von technischen Änderungen über den Produktlebenszyklus. Im Verlauf eines Entwicklungsprojekts wird das Systemmodell durch fachdisziplinübergreifende Informationen angereichert und konkretisiert. Diese Informationsbasis wird im weiteren Produktlebenslauf durch technische Änderungen erweitert und aktualisiert. Das Systemmodell repräsentiert somit zu jedem Zeitpunkt im Produktlebenszyklus die aktuelle Konfiguration des technischen Systems aus einer fachdisziplinübergreifenden Sicht.

Als Ausgangsbasis zur Modellierung des Systemmodells dient die Spezifikationstechnik CONSENS⁵⁷ (vgl. Abschnitt 3.1.1). Hierbei sind nicht alle bestehenden Partialmodelle gleichermaßen relevant. Vielmehr liegt der Fokus auf der Wirkstruktur. Anforderungen und Funktionen werden als Hilfskonstrukt hinzugezogen. Für die Release-Planung wird die Systemmodellierung darüber hinaus um zwei Diagramme erweitert: das Feature-Modell und den Release-Plan. Bild 4-12 zeigt die für die Release-Planung relevanten Diagramme und deren Zusammenwirken.

Technische Sicht

Anforderungen sind das Bindeglied zwischen der Beschreibung des Systems aus Kunden- bzw. Marktsicht und technischer Sicht. Sie legen aus Sicht aller beteiligten Stakeholder den Maßstab für das fest, was das System leisten muss. Im Systementwurf dienen die funktionalen Anforderungen der Herleitung der Funktionen des technischen Systems. Die **Funktionen** werden in einer Funktionshierarchie strukturiert und solange in Teilfunktionen untergliedert, bis sinnvolle Lösungen zur Erfüllung der Teilfunktionen gefunden werden können. Damit dient das Partialmodell *Funktionen* gerade im Systementwurf als Hilfskonstrukt zur Herleitung eines technischen Lösungsansatzes, der im Partialmodell *Wirkstruktur* beschrieben wird.

⁵⁷ Grundsätzlich kann die Systemmodellierung auch mit der Modellierungssprache SysML erfolgen. Hierzu ist auf ein entsprechendes CONSENS-Profil für die SysML zurückzugreifen, das in [IKD+13] beschrieben wird.

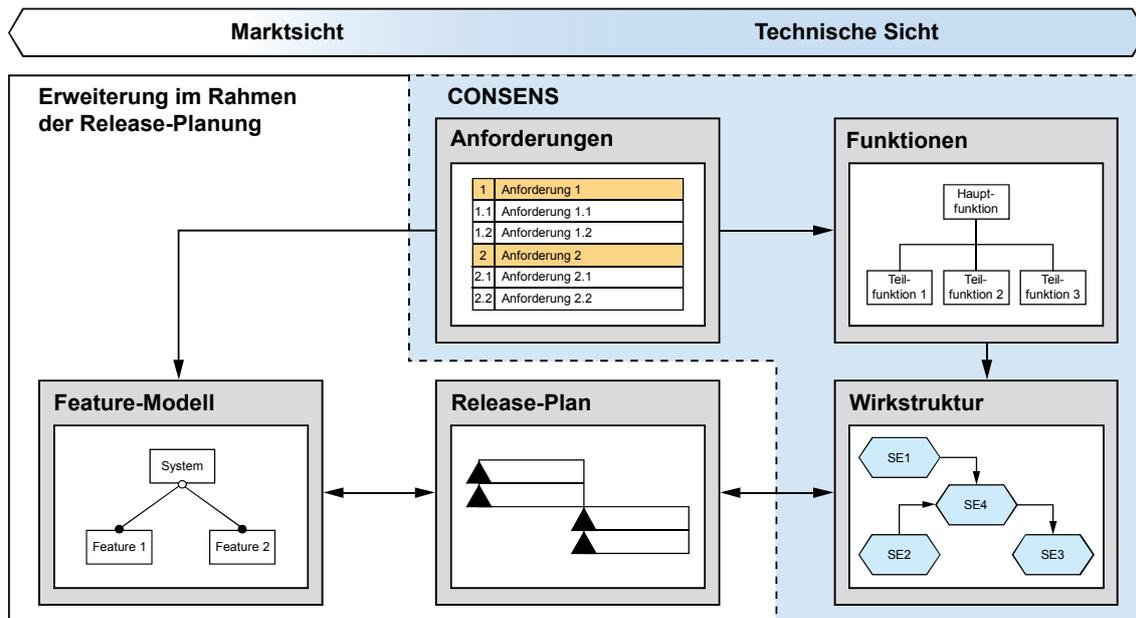


Bild 4-12: Zusammenwirken relevanter Diagramme zur Systemmodellierung in der Release-Planung

Die **Wirkstruktur** beschreibt schließlich den grundlegenden Aufbau und die Wirkungsweise des technischen Systems in einer fachdisziplinübergreifenden Weise. Kern dieser Beschreibung sind Systemelemente, bei denen es sich je nach Abstraktionsgrad um Module, Baugruppen, Bauteile und auch Softwarekomponenten handelt. Durch die Berücksichtigung von physischen und nicht-physischen Systemelementen vereint die Wirkstruktur verschiedenen Strukturen wie die Funktionsstruktur, Komponentenstruktur und Baustruktur, so dass die verschiedenen technischen Disziplinen wie Elektronik, Softwaretechnik und Regelungstechnik gleichermaßen berücksichtigt sind. Jedes Systemelement trägt zur Erfüllung einer oder mehrerer Funktionen bei, so dass über die Funktionen eine direkte Verknüpfung der Systemelemente zu den Anforderungen gegeben ist [Kah12, S. 109].

Erweiterung um Feature-Modell und Release-Plan

Für die Release-Planung wird neben der technischen Sicht auch eine Beschreibung des Systems aus **Marktsicht** benötigt. Das Systemmodell muss als Brücke zwischen der eigentlichen Systementwicklung und der Produktplanung fungieren. Hierbei liegt der Fokus weniger auf technischen Details als vielmehr auf der Beschreibung kundenrelevanter Produktmerkmale als Gegenstand einer innovationsorientierten Release-Planung. Diesem Anspruch werden die teilweise sehr feingranularen Anforderungen nicht gerecht, so dass sich hier eine abstrahierte Sicht auf die Anforderungen in Form von Features anbietet (vgl. Abschnitt 2.1.6). Features bündeln eine Menge funktionaler und/oder nicht-funktionaler Anforderungen zu relevanten Produktmerkmalen. Diese Produktmerkmale sind Diskussionsgrundlage für die Weiterentwicklung eines Systems (vgl. hierzu [KBD+14]). Sie werden durch Feature-Modelle beschrieben und strukturiert (Bild 4-13).

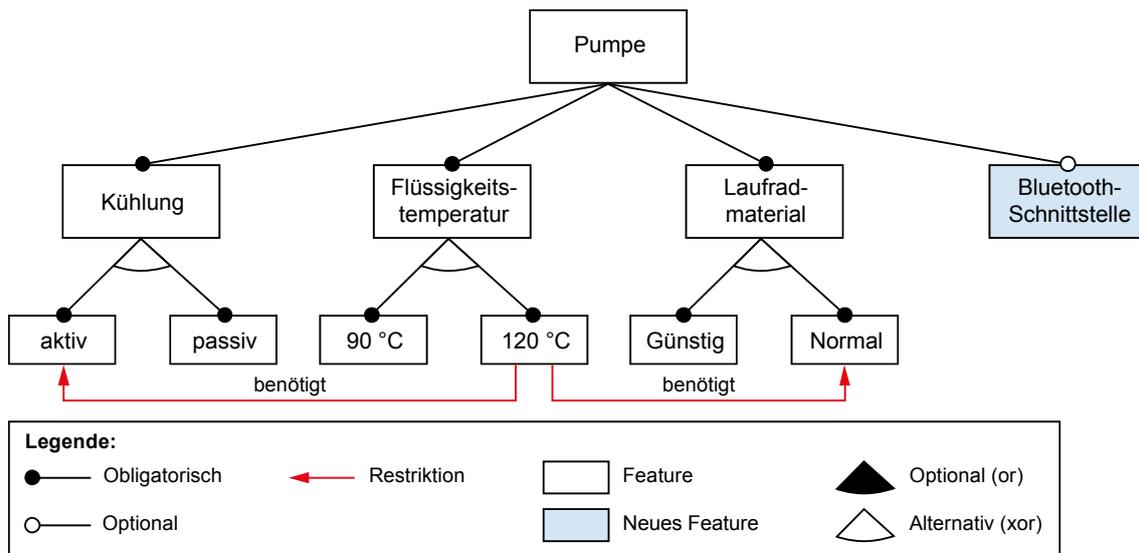


Bild 4-13: Feature-Modell zur Beschreibung eines Systems aus Marktsicht

Feature-Modelle ermöglichen die Beschreibung von Produktkonfigurationen aus zwei Perspektiven (vgl. Abschnitt 3.1.5): 1) Produktvarianten unterscheiden sich durch optionale und alternative Features⁵⁸; 2) Produktversionen zeichnen sich durch Hinzufügen neuer Features und die Aktualisierung alter Features aus. Im Beispiel in Bild 4-13 handelt es sich um eine Bluetooth-Schnittstelle, die in einer neuen Produktversion hinzugefügt wird. Darüber hinaus erlauben Feature-Modelle die Annotation von Abhängigkeiten, die es bei der Release-Planung zu berücksichtigen gilt. So setzt im Beispiel eine Flüssigkeitstemperatur von 120 °C eine aktive Kühlung sowie das normale Laufradmaterial voraus. Durch die Verknüpfung jedes einzelnen Features mit den Anforderungen ist eine direkte Rückverfolgbarkeit bis zur Wirkstruktur des technischen Systems gegeben. So können Zusammenhänge zwischen relevanten Produkt-Features und deren technischen Umsetzung identifiziert werden.

Bei der Weiterentwicklung des Systems über den Produktlebenszyklus werden Features hinzugefügt und aktualisiert. Diesem Aspekt wird mit der Erweiterung der Systembeschreibung um den Release-Plan Rechnung getragen. Der **Release-Plan** beschreibt die Weiterentwicklung des Systems über die Zeit. Hierbei können zwei Sichten unterschieden werden: die strukturorientierte und die inhaltsorientierte Sicht auf den Release-Plan (Bild 4-14).

Die **strukturorientierte Sicht** auf den Release-Plan beschreibt die Veröffentlichung unterschiedlicher Release-Typen über die Zeit. Diese Sicht dient in der strategischen Release-Planung der Strukturierung des Release-Plans unabhängig vom Release-Inhalt. Die

⁵⁸ Im Rahmen dieser Arbeit liegt der Fokus auf der Beschreibung von Produktversionen, so dass auf die Beschreibung der räumlichen Varianz im Zusammenspiel mit CONSENS nicht näher eingegangen wird.

inhaltorientierte Sicht auf den Release-Plan beschreibt hingegen die konkreten Release-Inhalte. Jedem Release werden dazu die Produkt-Features und Änderungen zugeordnet, die mit der Veröffentlichung umgesetzt werden sollen. Die inhaltsorientierte Sicht wird insb. in der taktischen und operativen Release-Planung eingesetzt. Mit der inhaltsorientierten Sicht auf den Release-Plan können sowohl das Feature-Modell als auch die Wirkstruktur mit einem bestimmten Zeitpunkt im Produktlebenszyklus verknüpft werden. Das Zusammenspiel von Release-Plan, Feature-Modell und Wirkstruktur wird im folgenden Abschnitt beschrieben.

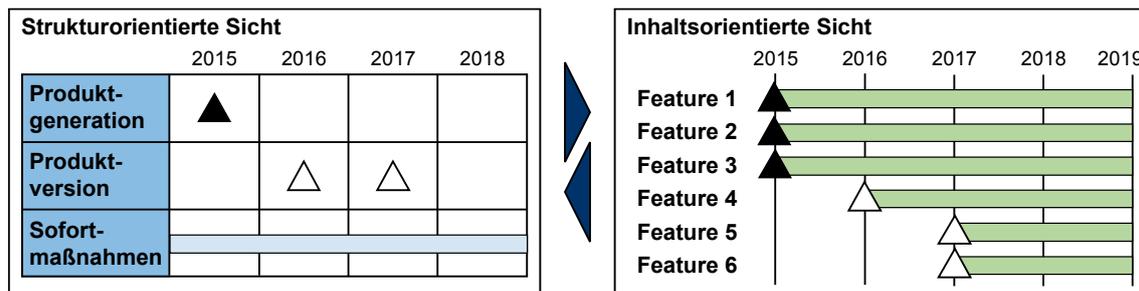


Bild 4-14: Zwei Sichten auf den Release-Plan

4.5.2 Beschreibung der Produktevolution

Eine zentrale Anforderung an das Systemmodell im Kontext der Release-Planung ist die Beschreibung der Weiterentwicklung des technischen Systems. Dies erlaubt zum einen die Planung und Diskussion zukünftiger Releases, zum anderen stellt es die Rückverfolgbarkeit von früheren Entscheidungen in der Release-Planung sicher. Durch die Kopplung von Feature-Modell und Wirkstruktur werden dabei marktorientierte und technische Sicht gleichermaßen berücksichtigt. Die Verknüpfung mit dem entsprechenden Release und dem damit verbundenen Zeitpunkt der Produktveröffentlichung wird über den Release-Plan sichergestellt (Bild 4-15).

Die Kombination aus Feature-Modell und Wirkstruktur beschreibt das technische System zu einem bestimmten Zeitpunkt. Dabei kann ein Feature durch ein oder mehrere Systemelemente erfüllt werden. Im Beispiel wird mit Release R1 ein weiteres Feature F3 hinzugefügt, das zu einer technischen Änderung an Systemelement SE3 führt. Das spätere Release R2 aktualisiert Feature F2 und hat somit Änderungen an den Systemelementen SE2 und SE3 zur Folge. Damit liegen beide Systemelemente in neuen Versionen vor, die explizit dem Release R2 zugeordnet werden können.

Sowohl Feature-Modell als auch Wirkstruktur können sehr umfangreich werden. Der Umfang der Modelle nimmt bei einer hohen Anzahl von Änderungen im Produktlebenslauf überdies zu. Je nach Komplexität des betrachteten Systems sind daher sowohl Feature-Modell als auch Wirkstruktur angemessen zu hierarchisieren. So bietet es sich an, bestimmte Features und damit verbundene Änderungen nur auf Modul- bzw. Subsystemebene zu berücksichtigen. In diesem Fall werden sowohl Feature-Modell als auch Release-Plan zusätzlich modul- bzw. subsystemspezifisch erstellt. Auf Systemebene werden

dann nur noch kundenrelevante Features berücksichtigt, die als Teil der Marketingstrategie hohe Relevanz für Produktmanagement, Vertrieb und Marketing haben.

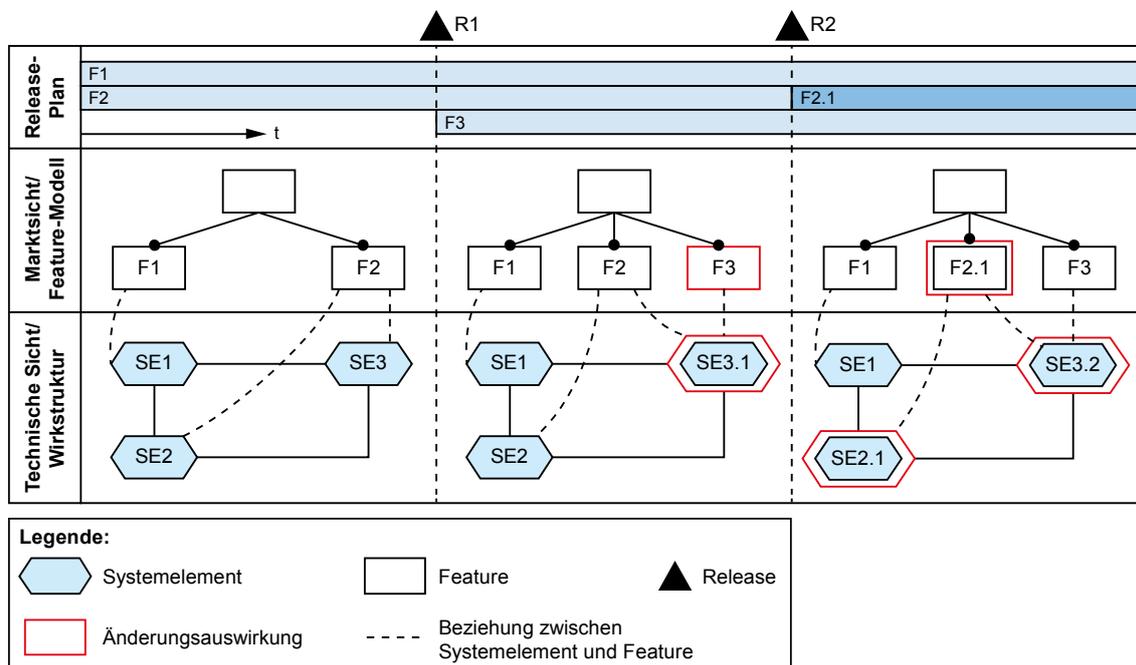


Bild 4-15: Beschreibung der Produktevolution mittels Release-Plan, Feature-Modell und Wirkstruktur

4.5.3 Unterstützung der Änderungsanalyse

Eine weitere Anforderung an das Systemmodell im Kontext der Release-Planung ist neben der Beschreibung der Produktevolution mittels Feature-Modell und Release-Plan die Unterstützung der technischen Änderungsanalyse. Die Änderungsanalyse ist von zentraler Bedeutung, da die Eigenschaften einer Änderung die Release-Entscheidung maßgeblich beeinflussen. So resultieren sowohl die technischen Auswirkungen als auch die Abhängigkeiten einer Änderung in Restriktionen, die es in der Release-Planung zu berücksichtigen gilt.

Änderungsauswirkungen können grob wie folgt unterschieden werden: lokale Änderungen an einer Komponente bzw. einem Modul, nicht-lokale Änderungen durch modulübergreifende Änderungsfortpflanzung sowie architektonische Änderungen durch vollständig neue Komponenten bzw. Module. Je höher die architektonische Auswirkung, desto wahrscheinlicher ist der Aufschub der Änderung auf ein Major-Release bzw. eine neue Produktgeneration, die eine umfassende Überarbeitung des Systems umfasst.

Neben den Auswirkungen gilt es, mögliche Abhängigkeiten zwischen den Änderungen zu berücksichtigen. Hierbei können drei Typen von **Änderungsabhängigkeiten** unterschieden werden:

- **Technische Abhängigkeit:** Änderungen können sich technisch in ihrer Umsetzung bedingen. So setzt bspw. die Realisierung einer Diagnosefunktion die vorherige Integration eines Sensors voraus. Im Falle technischer Abhängigkeiten sind zwei mögliche Restriktionen zu unterscheiden: 1) Eine Rangfolge, die die Umsetzung der einen Änderung vor der anderen festlegt sowie 2) die feste Kopplung der beiden Änderungen für den Fall, dass sich beide Änderungen gegenseitig bedingen. In diesem Fall sind beide Änderungen im weiteren Verlauf der Release-Planung als *eine* Änderung zu betrachten.
- **Abhängigkeit im Aufwand:** Änderungen können sich gegenseitig in ihrem Umsetzungsaufwand beeinflussen. Die gemeinsame Umsetzung von zwei Änderungen, die sich auf das gleiche Modul beziehen, reduziert bspw. den Aufwand für den notwendigen Modultest. Auch wenn die Erschließung dieser Synergieeffekte empfehlenswert ist, ist sie nicht zwingend. Es ist immer noch möglich, eine der Änderungen z.B. aus Marketinggründen oder auf Grund mangelnder Ressourcen auf ein späteres Release zu verschieben.
- **Abhängigkeit im Nutzen:** Ähnlich wie beim Aufwand können sich Änderungen auch gegenseitig in ihrem Nutzen beeinflussen. So besteht die Möglichkeit, dass eine Änderung zwar technisch ohne eine andere Änderung zu realisieren ist, sie aber ohne Nutzen für den Kunden bleibt. Ein Beispiel ist auch hier der Zusammenhang zwischen einer Diagnosefunktion und der Integration des zugehörigen Sensors. So kann der Sensor bereits integriert werden, während die Diagnosefunktion erst durch ein späteres Release hinzugefügt wird. Der Sensor bleibt so vorerst ohne Nutzen. Genau wie beim Aufwand ist die Berücksichtigung von Abhängigkeiten im Nutzen nicht zwingend, aber empfehlenswert.

Sowohl die Änderungsauswirkungen als auch die Abhängigkeiten gilt es, im Rahmen der Release-Planung durch eine systematische Änderungsanalyse zu berücksichtigen. Die Änderungsanalyse wird maßgeblich durch das Systemmodell unterstützt. Eine zentrale Rolle nimmt dabei die Wirkstruktur ein. Sie ermöglicht die fachdisziplinübergreifende Verortung von Änderungen und unterstützt die Auswirkungsanalyse durch spezifische Attribute. Diese spiegeln zentrale Eigenschaften der Systemelemente wider (Bild 4-16). Jedes Systemelement wird den *betroffenen Fachdisziplinen* (z.B. Regelungstechnik, Softwaretechnik, Mechanik) zugewiesen und hinsichtlich des *Komplexitätsgrads*⁵⁹ bewertet [Kah12, S. 110f]. Das Attribut *Mehrfachverwendung* gibt an, ob das Systemelement bzw. Modul als Gleichteil in anderen Produktvarianten bzw. -familien genutzt wird. Dies deutet auf potentielle Änderungsfortpflanzungen hin und ist im Rahmen der Änderungsanalyse weiter zu untersuchen.

⁵⁹ Bei der Ermittlung des *Komplexitätsgrads* folgt diese Arbeit dem Ansatz nach KAHL [Kah12, S. 110f.]. Hier wird davon ausgegangen, dass innerhalb eines Unternehmens zumindest Einigkeit über eine Ordinalskala für die Bewertung des Komplexitätsgrads besteht.

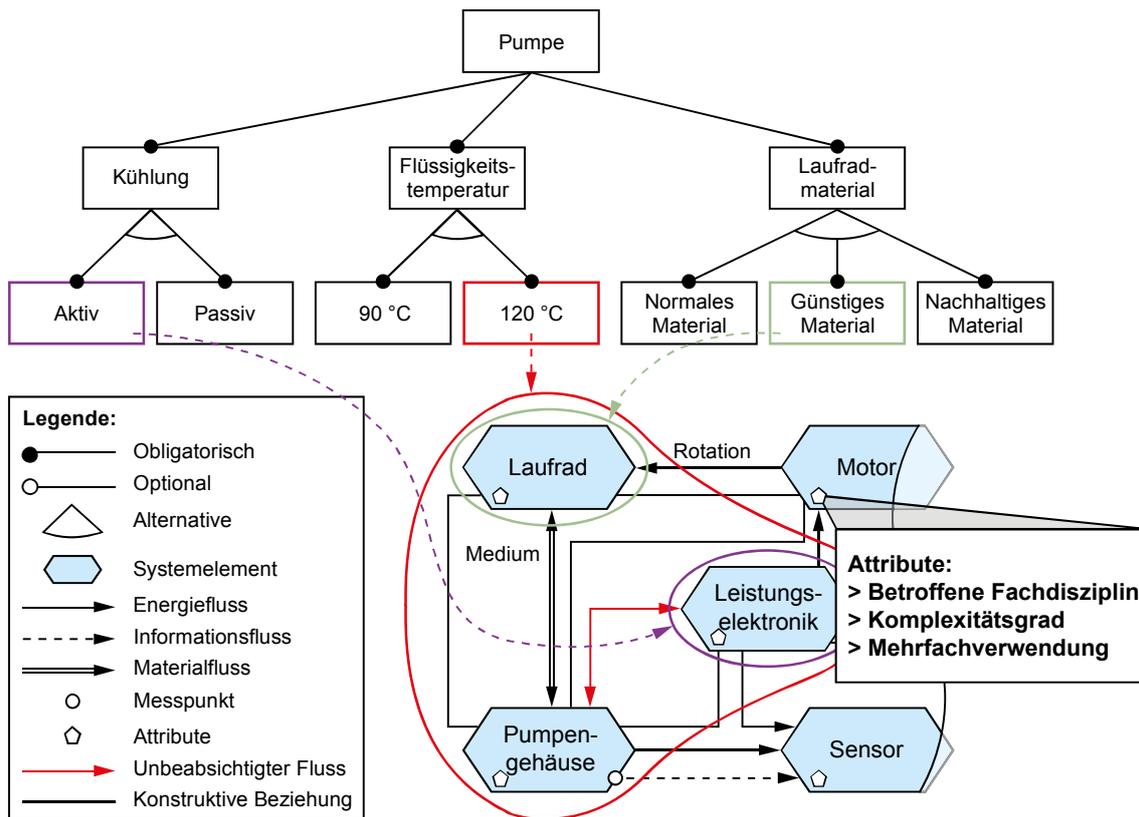


Bild 4-16: Analyse von Auswirkungen und Abhängigkeiten von Änderungen auf Basis des Systemmodells

Ausgehend vom initial betroffenen Systemelement und den zugewiesenen Attributen können in interdisziplinären Workshops mögliche Änderungsauswirkungen und -abhängigkeiten diskutiert werden. Gerade im Workshop hilft hierbei die Darstellung der Wirkstruktur in einem Diagramm. Zur systematischen Dokumentation der Änderungsauswirkungen kann zusätzlich auf die Darstellung in einer Matrix zurückgegriffen werden (vgl. Abschnitt 3.1.4). Das entsprechende Vorgehen zur Änderungsanalyse ist Gegenstand der taktischen Release-Planung und wird daher in Abschnitt 5.2.1 wieder aufgegriffen und vertieft.

5 Anwendung und Bewertung der Systematik zur Release-Planung

In diesem Kapitel wird die entwickelte Systematik angewendet. Die **Abschnitte 5.1 bis 5.3** orientieren sich dazu an den drei Ebenen der Release-Planung, die jeweils entlang des zugehörigen Vorgehensmodells erläutert werden. Im Rahmen der einzelnen Arbeitsschritte wird der Einsatz der Spezifikationstechnik sowie der Methoden und Hilfsmittel erläutert. Abschließend wird in **Abschnitt 5.4** die entwickelte Systematik anhand der Anforderungen aus der Problemanalyse bewertet (vgl. Abschnitt 2.7).

Zur Veranschaulichung und besseren Verständlichkeit der Systematik wird diese anhand von Ergebnissen aus Validierungsprojekten beschrieben. Als durchgängiges **Anwendungsbeispiel**⁶⁰ dient ein elektrisches Regelventil (siehe Bild 5-1).

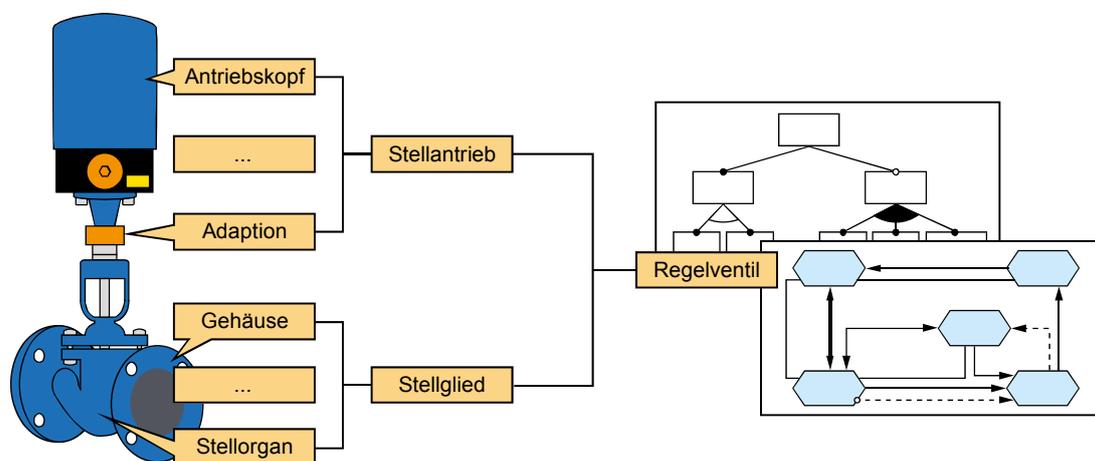


Bild 5-1: Anwendungsbeispiel elektrisches Regelventil

Regelventile sind Paradebeispiele für mechatronische Produkte. Als Aktoren in prozessautomatisierten, verfahrenstechnischen Anlagen verändern sie durch Öffnen, Schließen oder teilweises Absperren Medienströme. Dazu sind sie informationstechnisch in ein übergeordnetes Prozessleitsystem eingebunden. Gleichzeitig besitzen sie als Teil des Rohrleitungssystems eine mechanische Schnittstelle zum verfahrenstechnischen Prozess. Sie sind in der Regel modular aufgebaut und lassen sich in die Teilsysteme Stellglied und Stellantrieb gliedern. Jedes dieser Teilsysteme besteht wiederum aus einer Vielzahl weiterer Sub-Systeme, wie dem Antriebskopf, der Adaption, dem Gehäuse und dem Stellor-

⁶⁰ Voraussetzung für die Anwendung der Systematik ist die Verfügbarkeit einer Dokumentation des betrachteten Systems in der Spezifikationstechnik CONSENS. Dementsprechend wurden für das Anwendungsbeispiel Regelventil die relevanten Diagramme aufgenommen. Konkret handelt es sich um die Funktionshierarchie, die Wirkstruktur, das Feature-Modell und den Release-Plan.

gan. In Abhängigkeit von Einsatzort und Einsatzzweck werden verschiedene Regelventile eingesetzt, die sich z.B. bzgl. ihrer Nennweiten, Materialien und Antriebsarten unterscheiden [KE08, S. 2f.].

Eine im Rahmen eines Validierungsprojekts durchgeführte Potentialanalyse zeigt, dass sich der in Abschnitt 2.2 beschriebene Wandel vom mechatronischen zum intelligenten technischen System auch auf das Regelventil – und hier insb. auf den elektrischen Stellantrieb – übertragen lässt [HK15]. Dies veranschaulicht Bild 5-2.

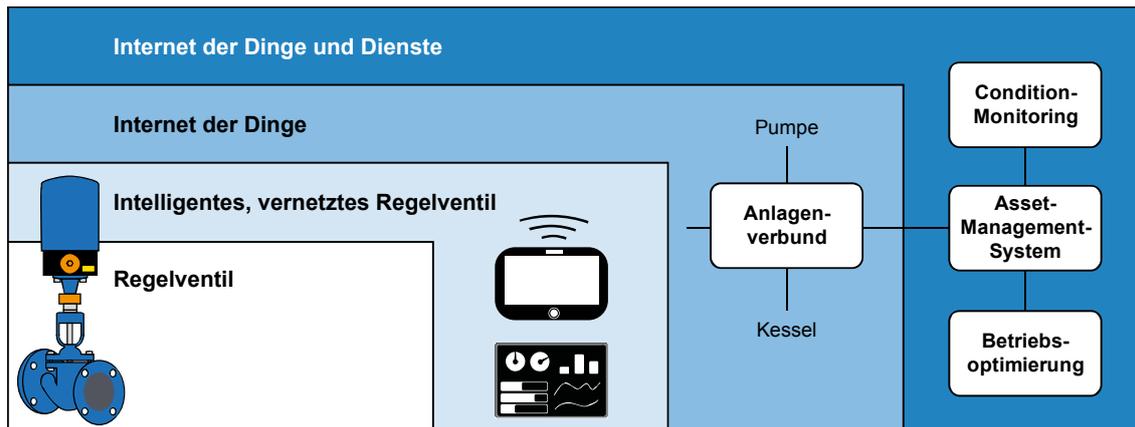


Bild 5-2: Typische Evolution technischer Systeme am Beispiel des Regelventils in Anlehnung an PORTER und HEPPELMANN [PH14, S. 7ff.]

Schon seit einigen Jahren ist die Weiterentwicklung des Regelventils von einem zunehmenden Anteil von Elektronik und Software geprägt. Dies hat in den letzten Jahren die Integration neuer Funktionen ermöglicht (z.B. automatische Kalibrierung, einfache Diagnosefunktionen und eine übergeordnete Kommunikation mit dem Leitsystem). Nicht zuletzt durch die zunehmende Vernetzung der Systeme im Sinne des Internets der Dinge und Dienste zeichnen sich jedoch auch für das Regelventil verschiedene Innovationspotentiale ab. Diese reichen von der Einbindung des Regelventils in übergeordnete Asset-Management-Systeme⁶¹ bis hin zu Funktionen für die prädiktive Wartung. Resultat sind technische Änderungen⁶². Diese wirken sich sowohl auf die Mechanik als auch auf Elektronik und Software des Systems aus und weisen darüber hinaus erhebliche Abhängigkeiten untereinander auf. Dies macht eine stufenweise und abgestimmte Umsetzung dieser Änderungen erforderlich. Hierfür bietet sich ein systematischer Release-Planungsprozess an, der im Folgenden als Anwendungsbeispiel für die im Rahmen dieser Arbeit entwickelte Systematik dient.

61 Der Begriff Asset-Management-System ist in diesem Kontext von dem Begriffsverständnis aus der Vermögensverwaltung bzw. dem Investmentbanking zu unterscheiden. Es handelt sich um ein Managementsystem, das der Steigerung von Effektivität und Effizienz beim Betrieb von Anlagegütern (Assets) dient. In der Prozessindustrie wird auch von Plant-Asset-Management (PAM) gesprochen [VDI2651].

62 Im Fokus dieser Arbeit stehen Änderungen an den Systemelementen eines technischen Systems. Änderungen ohne direkte Auswirkungen auf das Produkt werden nicht betrachtet (z.B. organisatorische Änderungen, Prüfmitteländerungen oder Verpackungsänderungen).

Das Regelventil als Anwendungsbeispiel wird darüber hinaus durch Ergebnisse und Erfahrungen aus weiteren Validierungsprojekten ergänzt, die insb. im Maschinen- und Anlagenbau durchgeführt wurden (z.B. Pumpen, Geldautomaten, Windenergieanlagen). Aus Gründen der Vertraulichkeit sind sämtliche Ergebnisse ausschließlich beispielhaft und auszugsweise dargestellt, ohne jedoch das Verständnis der beschriebenen Systematik einzuschränken.

5.1 Strategische Release-Planung

Grundlage für den Planungsprozess ist die initiale Strukturierung des Release-Plans. Dieser gibt die Leitplanken für die taktische und operative Release-Planung vor und ist entscheidend für deren Erfolg. Die Strukturierung des Release-Plans erfolgt in drei Phasen: Aufbauend auf der Auswahl der Release-Strategie in Phase 1 (**Abschnitt 5.1.1**) werden in Phase 2 die Release-Typen definiert (**Abschnitt 5.1.2**). Anschließend werden in Phase 3 verschiedene Einflussfaktoren analysiert, die die Strukturierung des Release-Plans beeinflussen (**Abschnitt 5.1.3**). Dies erfolgt stets unter Berücksichtigung der Gestaltungsfaktoren aus der Release-Strategie. Ergebnis ist ein Release-Plan, der alle relevanten Release-Termine entlang des Planungshorizonts definiert. Während die Phasen 1 und 2 in der Regel nur einmalig durchlaufen werden, sind die Ergebnisse aus Phase 3 im Sinne einer rollierenden Planung jährlich zu verfeinern und zu aktualisieren. Dies erfolgt im Wechselspiel mit der Erarbeitung und Aktualisierung der Produkt-Roadmap (vgl. Abschnitt 2.4.1). Auf die Einbettung in die strategische Produktplanung und damit auf die Schnittstellen zwischen Produkt-Roadmap und strategischer Release-Planung wird daher im Folgenden explizit eingegangen.

5.1.1 Phase 1: Auswahl der Release-Strategie

Die Release-Strategie bildet das Fundament für die weitere Strukturierung des Release-Plans. Ziel dieser Phase ist eine auf das jeweilige Produkt und die Unternehmenssituation abgestimmte Release-Strategie. Die Auswahl der Release-Strategie folgt dem in Abschnitt 4.3.3 beschriebenen Schema und sieht eine Bewertung des technischen Systems hinsichtlich seiner Änderungswahrscheinlichkeit und Komplexität vor.

Die **Änderungswahrscheinlichkeit** wird durch ein Expertenteam entlang der zwei Dimensionen externe und interne Unsicherheit bestimmt und mittels des in Bild 5-3 dargestellten Portfolios visualisiert. Hierzu werden die interne Unsicherheit auf der Abszisse und die externe Unsicherheit auf der Ordinate aufgetragen.

Insgesamt ist die Änderungswahrscheinlichkeit des Anwendungsbeispiels *Regelventil* als *mittel* einzustufen. Dies ergibt sich aus dem arithmetischen Mittel aus externer und interner Unsicherheit. Die *externe Unsicherheit* wird aufgrund eines wenig dynamischen Kundenumfelds mit gering bis mittel bewertet. In einem eher konservativen Markt fordern und adaptieren die Kunden Innovationen nur zögerlich. Gleichzeitig zwingt ein intensiver

Wettbewerb das Unternehmen jedoch immer wieder zur Differenzierung, die die technische Weiterentwicklung des Regelventils erforderlich macht. Die hohe *interne Unsicherheit* ergibt sich im Wesentlichen aus dem Neuheitsgrad des Produkts für das Unternehmen. Dieser ist als hoch einzustufen, so dass im Verlauf des Produktlebenszyklus einerseits mit fehlerbedingten Änderungen zu rechnen ist, andererseits aber auch noch erhebliches Weiterentwicklungspotential erschlossen werden kann. Sowohl externe als auch interne Unsicherheiten resultieren in einer Vielzahl potentieller Änderungen, die systematisch umgesetzt werden müssen.

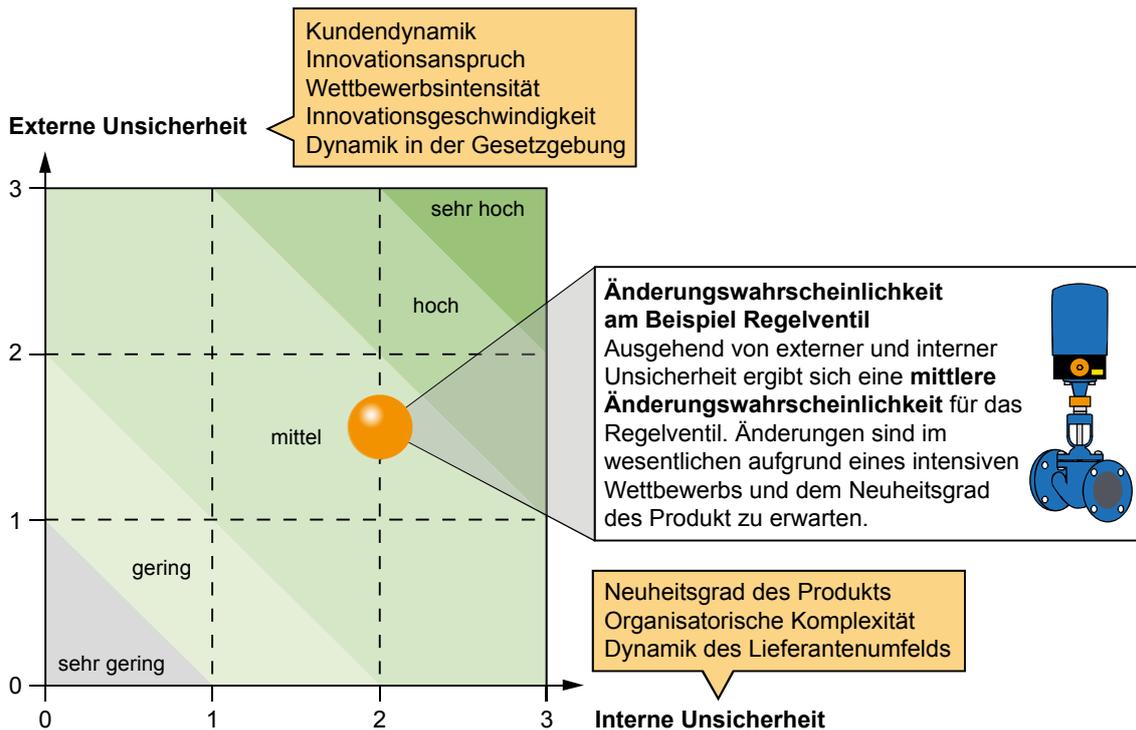


Bild 5-3: Portfolio zur Bewertung der Änderungswahrscheinlichkeit in Abhängigkeit von externen und internen Unsicherheiten

Grundsätzlich erlaubt die Bestimmung der Änderungswahrscheinlichkeit erste Rückschlüsse auf die Release-Frequenz: Je höher die Änderungswahrscheinlichkeit, desto eher bietet sich eine häufige Veröffentlichung neuer Releases an. Dies kann bspw. quartalsweise, halbjährlich oder jährlich erfolgen. Durch die Differenzierung in externe und interne Unsicherheit kann dabei die Release-Frequenz für kundenrelevante und rein technische Releases individuell festgelegt werden. Im Falle hoher externer Unsicherheit ist durch Reaktionen auf veränderte Markt- und Kundenanforderungen mit einer Vielzahl kundenrelevanter Releases zu rechnen. Die interne Unsicherheit erlaubt hingegen Rückschlüsse auf die Wahrscheinlichkeit fehlerbedingter Änderungen, die letztlich in rein technischen Releases beseitigt werden. Dies wird in der Strukturierung des Release-Plans in den Abschnitten 5.1.2 und 5.1.3 erneut aufgegriffen.

Neben der Änderungswahrscheinlichkeit ist für die Auswahl der Release-Strategie die **Komplexität des Zielsystems** zu bestimmen. Grundlage sind die drei Kriterien räumliche

Varianz, Kommunalität und Produktkomplexität. Im Anwendungsbeispiel ist die Komplexität des Zielsystems *Regelventil* als insgesamt gering bis mittel einzuschätzen. So handelt es sich um eine dedizierte Produktgruppe, die nur einem kleinen Teil des gesamten Produktprogramms des Unternehmens entspricht und damit eine geringe *räumliche Varianz* aufweist. Innerhalb der Produktgruppe liegen einige Varianten vor, für die durch die Nutzung von Gleichteilen Synergieeffekte erzielt werden können. Dennoch zeichnet sich das System durch geringe Kommunalität aus. Für die Bewertung der *Produktkomplexität* ist eine Analyse der Systemarchitektur erforderlich, die bspw. auf Basis der Spezifikationstechnik CONSENS erfolgen kann. Hier gilt es ausgehend von der Anzahl und Vielfalt der Module bzw. Systemelemente den Komplexitätsgrad für das betrachtete Zielsystem abzuschätzen. Für das Anwendungsbeispiel wird im Expertenteam eine mittlere Komplexität festgelegt.

Die Zusammenführung der zwei Dimensionen Änderungswahrscheinlichkeit und Komplexität des Zielsystems erlaubt schließlich die Auswahl einer geeigneten Release-Strategie. Bild 5-4 zeigt exemplarisch das Bewertungsergebnis für das Regelventil, dargestellt in einem Portfolio. Die Achsen des Portfolios sind die Komplexität des Zielsystems (Abzisse) sowie die Änderungswahrscheinlichkeit (Ordinate).

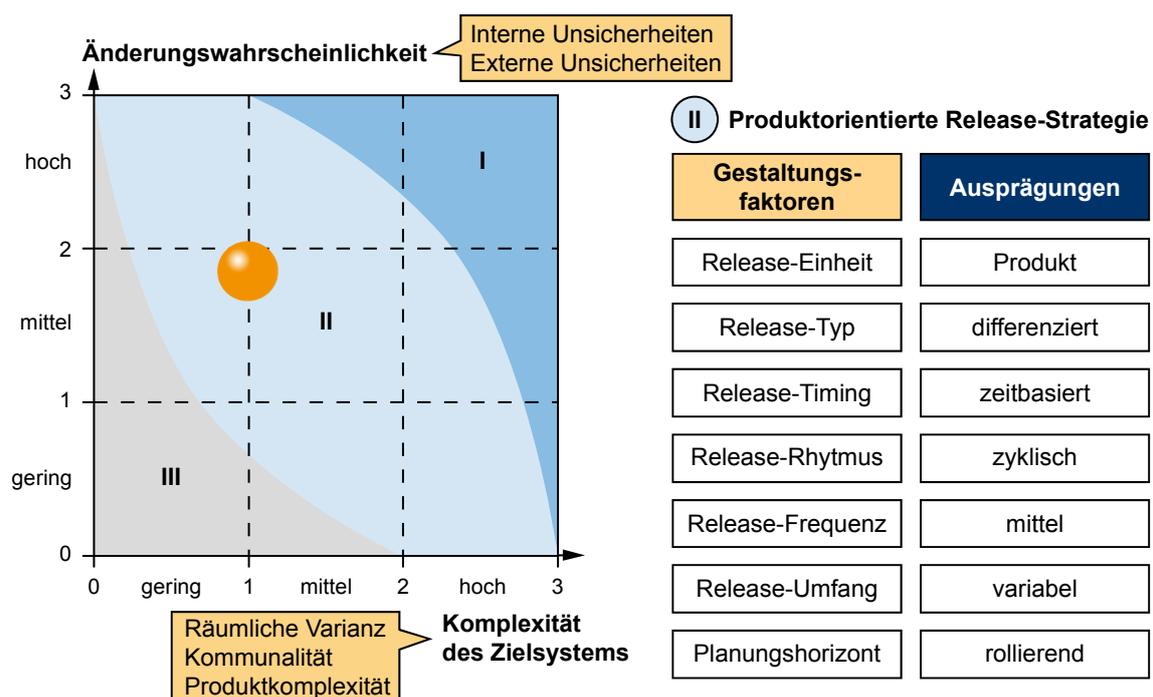


Bild 5-4: Auswahl der Release-Strategie für das Regelventil mittels Portfolioansatz

In der Auswertung empfiehlt sich für das betrachtete Regelventil die **produktorientierte Release-Strategie** (Strategie II), die sich durch die im Bild 5-4 rechts dargestellten Ausprägungen der Gestaltungsfaktoren auszeichnet. Die Strategie gibt wesentliche Rahmenbedingungen für die Definition der Release-Typen und die weitere Strukturierung des Release-Plans in den Phasen 2 und 3 vor. Demzufolge wird das Regelventil als Release-Einheit aufgefasst; eine weitere Gliederung in Module erfolgt nicht. Die Differenzierung

von Release-Typen erlaubt eine systematische Umsetzung sowohl kundenrelevanter als auch rein technischer Änderungen. Darüber hinaus standardisiert die zeitbasierte und zyklische Veröffentlichung der Releases den Planungsprozess, wodurch dem zuvor antizipierten mittleren Änderungsaufkommen entgegengewirkt werden kann. Dies erlaubt mit Blick auf die markt- und wettbewerbsstrategischen Nutzenpotentiale der Release-Planung insb. eine gezielte Produktwertsteigerung bei gleichzeitiger Nutzung von Änderungssynergien in der Entwicklung (vgl. Abschnitt 2.4.3). Die Auswahl der Release-Strategie beeinflusst somit maßgeblich die Planung des Produktlebenszyklus und damit die Produktstrategie einschließlich der darin enthaltenen Produkt-Roadmap.

5.1.2 Phase 2: Definition der Release-Typen

Ausgehend von der ausgewählten Release-Strategie gilt es in dieser Phase, den Gestaltungsfaktor **Release-Typ** zu konkretisieren. Die Ausprägung *differenziert* erfordert dazu die Definition unterschiedlicher Release-Typen durch das Unternehmen. Dazu ist zu entscheiden, wie viele Release-Typen unterschieden werden sollen; es ist ein Release-Konzept zu wählen. Ferner sind für jeden Release-Typ die zugehörigen Eigenschaften Änderungstyp und Verfügbarkeit zu konkretisieren. Ziel ist ein Klassifikationsschema, das die Zuordnung einer Änderung zu einem bestimmten Release-Typ im Rahmen der taktischen Release-Planung ermöglicht.

Die Unterscheidung von Release-Typen zielt auf die differenzierte Umsetzung verschiedener Änderungstypen durch standardisierte Bearbeitungsprozesse ab. Eine sinnvolle **Anzahl unterschiedlicher Release-Typen** ist daher abhängig von Anzahl und Häufigkeit bestimmter Änderungen. Die Anzahl hängt maßgeblich von der Heterogenität der Änderungen ab, die bspw. von einfachen Änderungen an einzelnen Bauteilen bis hin zu neuen Vertriebsanforderungen und Produktfunktionen reichen können. Die Häufigkeit wird maßgeblich durch die in Phase 1 bestimmte Änderungswahrscheinlichkeit beeinflusst. Um eine sinnvolle Anzahl an Release-Typen festzulegen, bietet sich daher eine Analyse bereits abgewickelter Änderungen im Unternehmen nach *Änderungstyp* und *Häufigkeit* an. Ausgehend davon können die in Anhang A3.4 beschriebenen Release-Typen zu einem Release-Konzept kombiniert werden. Im Anwendungsbeispiel Regelventil ergibt sich das in Tabelle 5-1 dargestellte 2-stufige Release-Konzept⁶³.

Während Sofortmaßnahmen und die Ablösung eines Produkts durch eine neue Produktgeneration Bestandteil jedes Release-Konzepts sind, erfolgt die Differenzierung über die Release-Typen. Für diese sind die **Eigenschaften** Änderungstyp, Verfügbarkeit und Release-Umfang festzulegen. Durch die Unterscheidung in Minor- und Major-Releases differenziert das im Anwendungsbeispiel Regelventil gewählte Konzept *Änderungstypen*

63 Die DAIMLER AG verfolgt mit den sogenannten „Änderungsjahren“ bspw. ein 3-stufiges Release-Konzept, das neben Sofortmaßnahmen sogenannte Releases (quartalsweise), Änderungsjahre (jährlich) und Modellpflege (MOPF) (ca. drei Jahre) unterscheidet [BG04, S. 21].

insb. im Hinblick auf die Umsetzungsplanung kundenrelevanter Änderungen. Gleichzeitig wird die Komplexität einer Änderung bei der Zuordnung zu einem Release berücksichtigt. Die *Verfügbarkeit* der Releases wird ausgehend von der in Phase 1 ermittelten Änderungswahrscheinlichkeit festgelegt. Aufgrund der mittleren Änderungswahrscheinlichkeit werden Minor-Releases halbjährlich und Major-Releases ca. alle zwei Jahre veröffentlicht. Sowohl für Minor- als auch für Major-Releases ist ein *Änderungsumfang* festzulegen, der die Anzahl und den Aufwand der umzusetzenden Änderungen unter Berücksichtigung der verfügbaren Ressourcen festlegt (z.B. Entwicklungsdauer, Personalverfügbarkeit). Entsprechend der Release-Strategie wird der Release-Umfang variabel gehalten und das Budget für jedes Release individuell im Rahmen der operativen Release-Planung festgelegt. Restriktionen ergeben sich hierbei lediglich durch die unternehmensweit verfügbaren Ressourcen.

Tabelle 5-1: 2-stufiges Release-Konzept als Basis für die Strukturierung des Release-Plans für das Anwendungsbeispiel Regelventil

		Änderungstyp	Verfügbarkeit
Sofortmaßnahmen		Sofortmaßnahmen umfassen sämtliche Änderungen, die von der Release-Planung ausgeschlossen sind und direkt umgesetzt werden. Dies sind dringliche Änderungen z.B. aufgrund gesetzlicher Maßnahmen. Auch geringfügige Änderungen z.B. an der technischen Dokumentation werden via Sofortmaßnahme umgesetzt.	Bei Bedarf (als Bypass zur Release-Planung)
Release-Typen	Stufe 1: Minor-Release	Minor-Releases umfassen rein technische Änderungen ohne Kundenrelevanz. Es handelt sich um weniger komplexe Änderungen zur Qualitätsverbesserungen und Kosteneinsparung.	Zyklisch, abgestimmt auf Major-Releases; halbjährlich
	Stufe 2: Major-Release	Major-Releases sind kundenrelevant. Sie umfassen signifikante Produktverbesserungen oder neue Produkt-Features. Auch komplexere Änderungen werden in diesem Rahmen umgesetzt.	Zyklisch; ca. alle 2 Jahre
Produktgeneration		Neue Produktgenerationen umfassen Änderungen, die sich durch einen besonders hohen Innovationsgrad auszeichnen z.B. aufgrund neuer Technologien oder einer Revolution im Design.	Bei Bedarf; bei grundlegenden (architektonischen) Änderungen

Für die Zuordnung der Änderungen zu den definierten Release-Typen ist in einem letzten Schritt ein **Klassifikationsschema** zu erarbeiten, das änderungsspezifische Informationen zu einer Bewertungs- und Entscheidungsgrundlage verdichtet und abstrahiert.

Typische Bewertungskriterien⁶⁴ zur Klassifizierung der Änderungen sind Änderungsdauer, Kosten, Änderungskomplexität, Neuheitsgrad und Nutzen. Im Anwendungsbeispiel wurde ein Portfolio-Ansatz⁶⁵ als Klassifikationsschema gewählt, der entsprechend des Release-Konzepts die zwei Dimensionen Änderungsrelevanz und Änderungskomplexität unterscheidet (Bild 5-5). Der Umgang mit dem hier dargestellten Klassifikationsschema wird in Abschnitt 5.2.3 konkretisiert.

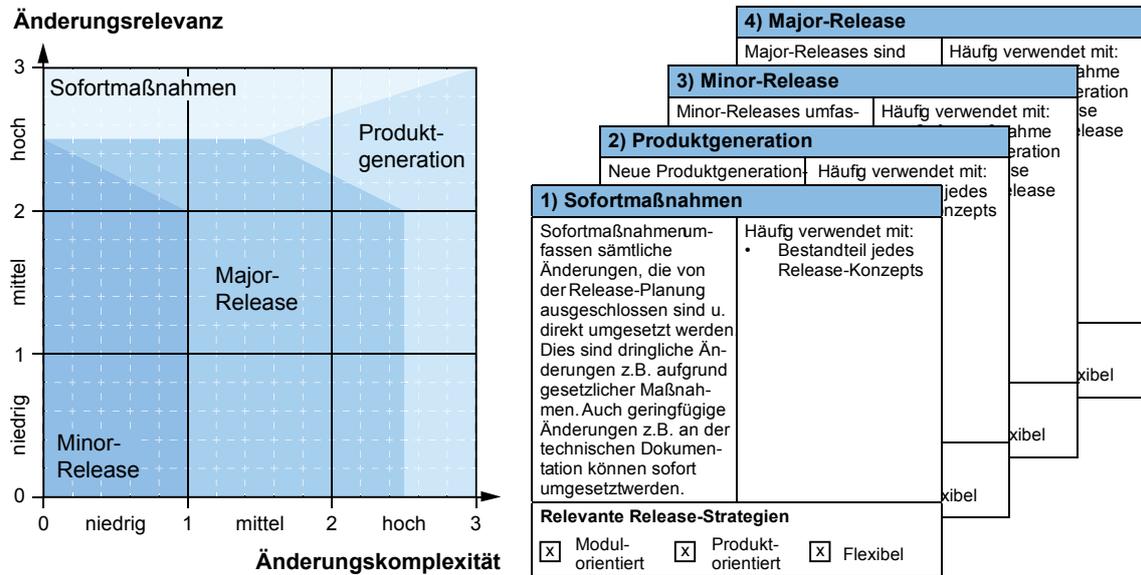


Bild 5-5: Klassifikationsschema zur Zuordnung technischer Änderungen zu den zuvor definierten Release-Typen

5.1.3 Phase 3: Strukturierung des Release-Plans

Kernaufgabe der strategischen Release-Planung ist die Strukturierung des Release-Plans. Wesentliche Vorgaben ergeben sich aus der zuvor ausgewählten Release-Strategie und der Definition der Release-Typen. In den nächsten Schritten sind nun der Planungshorizont (Abschnitt 0) sowie die Einführungszeitpunkte für Major-Releases (Abschnitt 5.1.3.1) und Minor-Releases (Abschnitt 5.1.3.2) festzulegen. Insbesondere zur Festlegung der Major-Releases erfolgt ein enger Abgleich mit planungsrelevanten Informationen aus der strategischen Produktplanung⁶⁶ bzw. der Produkt-Roadmap. Ziel ist ein Release-Plan, der Aufschluss über Markteinführungszeitpunkte sämtlicher Releases gibt. Grundsätzlich gilt dabei, dass die Planungsgenauigkeit mit zunehmendem Zeithorizont

64 Eine Übersicht potentieller Kriterien zur Klassifizierung von Änderungen gibt Anhang A3.1. Diese können als Grundlage zur Erarbeitung eines unternehmensindividuellen Klassifikationsschemas genutzt werden.

65 Eine Alternative zum Portfolio-Ansatz ist die von ABMANN vorgestellte Klassifizierung mittels Checkliste (vgl. Abschnitt 3.3.2). Auch sogenannte „Entscheidungsbäume“ können zur Klassifizierung genutzt werden [BV08, S. 273ff.].

66 Zur Informationsbeschaffung im Rahmen der strategischen Produktplanung eignen sich insb. die in Abschnitt 3.4 vorgestellten Ansätze.

abnimmt. Bild 5-6 zeigt exemplarisch den initialen Release-Plan für das Anwendungsbeispiel Regelventil.

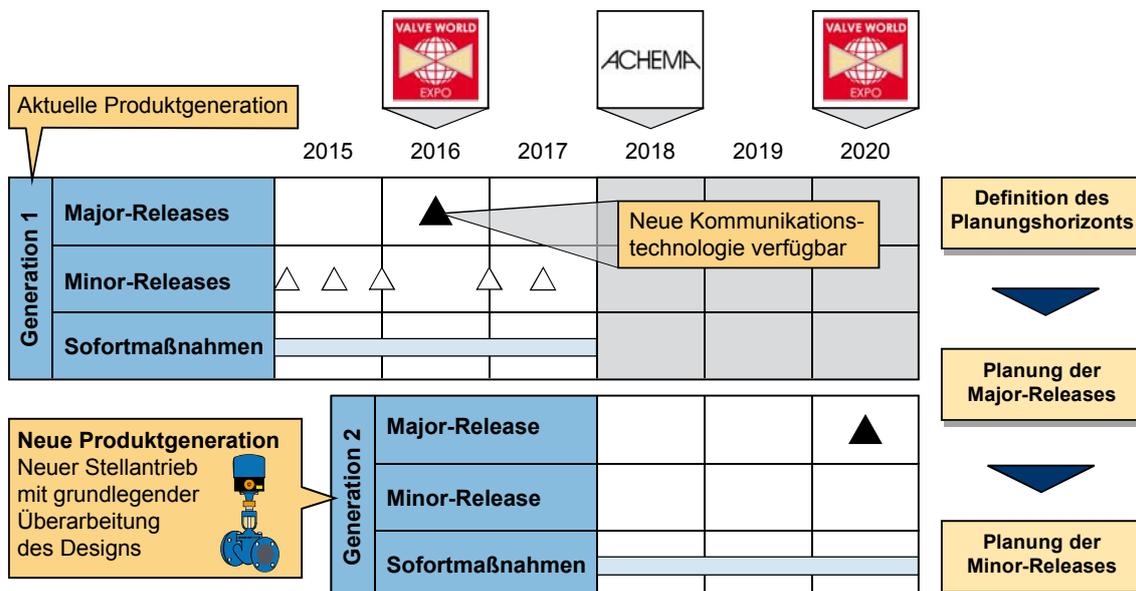


Bild 5-6: Initialer Release-Plan als Ergebnis der strategischen Release-Planung

Der **Release-Plan** weist einen Planungshorizont von sechs Jahren auf. Mit der Ablösung der aktuellen Produktgeneration im Jahr 2018 wird eine zusätzliche Planungsebene aufgemacht, für die bis ins Jahr 2020 weitere Releases nur noch grob vorausgeplant werden. Über Sofortmaßnahmen können dringliche Änderungen (z.B. Fehlerbehebungen, hohe Kosteneinsparpotentiale) jederzeit umgesetzt werden. Die Strukturierung des Release-Plans folgt dem in Bild 5-6 dargestellten 3-stufigen Vorgehen. Dies wird im Folgenden erläutert. Definition des Planungshorizonts

Die Definition des **Planungshorizonts** hängt stark vom Produktlebenszyklus und damit von Produkttyp und Branche ab. Je höher die Innovationsdynamik einer Branche, desto kürzer der Produktlebenszyklus und die Zeitspanne, für die eine sinnvolle Planung erfolgen kann. Als grobe Faustregel in der strategischen Produktplanung gilt, dass sich der Planungshorizont auf einen Zeitraum von mindestens zwei Produktgenerationen bezieht. Während die erste Produktgeneration dabei möglichst detailliert geplant und fixiert werden sollte, kann die Planung der zweiten eher eine grobe Schätzung sein [DNL96, S. 167], [Pei15, S. 99]. Die detaillierte Release-Planung schließt somit typischerweise mit der Ablösung einer Produktgeneration ab. Die grobe Schätzung über die nächste Produktgeneration erfolgt dann im Rahmen der strategischen Produktplanung als Teil der Produktstrategie und der darin enthaltenen Produkt-Roadmap.

Der Zeitpunkt für die Ablösung der aktuellen Produktgeneration legt somit den Planungshorizont für die Release-Planung fest. Für die Definition dieses Zeitpunkts bestehen zwei Möglichkeiten: 1) Der Lebenszyklus einer Produktgeneration und damit der Zeitpunkt ihrer Ablösung sind zeitlich fest vorgegeben. Dies ist bspw. in der Automobilindustrie der Fall. 2) Ein konkretes Produkt-Feature führt zu einer technisch bedingten Änderung,

die eine grundlegende Überarbeitung des Systems und damit eine neue Produktgeneration zur Folge hat. Beides ist typischerweise der Produkt-Roadmap zu entnehmen, so dass in diesem Zuge eine Synchronisation von Produkt-Roadmap und Release-Planung unerlässlich ist.

Im Anwendungsbeispiel in Bild 5-6 führt das bereits bekannte Vorhaben eines neuen Stellantriebs für das Jahr 2018 zu einer vollständigen Überarbeitung des Produkts und damit zu einer neuen Produktgeneration. Die Detailplanungen schließen somit im Jahr 2018 ab. Lediglich bestehende Informationen aus der Produkt-Roadmap werden darüber hinaus in den Release-Plan überführt. Im Beispiel handelt es sich hierbei um das bereits geplante Major-Release für 2020.

5.1.3.1 Planung der Major-Releases

Im zweiten Schritt sind die Markteinführungszeitpunkte für die Major-Releases zu planen. **Major-Releases** führen kundenrelevante Änderungen ein, die sich entweder durch signifikante Produktverbesserungen, neue Produkt-Features oder aufwändigere Änderungen am System auszeichnen. Teilweise sind die Inhalte einzelner Major-Releases bereits bekannt; teilweise werden diese aber auch erst im weiteren Verlauf der Release-Planung festgelegt. In jedem Fall sind die eingeplanten Major-Releases als Meilensteine im Rahmen der Vermarktung des Produkts zu betrachten. Aufgrund der Kundenrelevanz ist erneut eine enge Abstimmung mit der Produktstrategie erforderlich. Planungsrelevante Informationen liefern eine übergeordnete Produkt- und Technologie-Roadmap, eine Wettbewerbsanalyse und die Analyse branchen- und marktspezifischer Termine (z.B. Messen, Saisonabhängigkeiten) (vgl. prinzipielle Darstellung in Bild 5-7). Zudem gibt die Analyse der externen Unsicherheiten aus Phase 1 einen ersten Eindruck über die Frequenz, mit der kundenrelevante Releases erscheinen sollten (im Anwendungsbeispiel ca. alle zwei Jahre).

Analyse der übergeordneten Produkt- und Technologie-Roadmap: Wie in den Abschnitten 2.4.1 und 2.4.4 herausgestellt, sind sowohl eine systematische Produkt- als auch Technologieplanung wesentliche Erfolgsfaktoren für die Release-Planung. Beide enthalten Meilensteine im Produktlebenszyklus, die als Vorgaben für den Release-Plan übernommen werden müssen. Dies können einerseits produktstrategische Vorhaben sein, wie im Anwendungsbeispiel das Konzept eines neuen Stellantriebs. Andererseits geht es um die Verfügbarkeit relevanter Technologien, die die geplante Markteinführung eines Releases maßgeblich beeinflussen. Bedingt durch die Entkopplung von Technologie- und Systementwicklung bedarf es hier der Synchronisation beider Planungsebenen (vgl. Abschnitt 2.4.4). Im Anwendungsbeispiel ergibt sich hieraus die Verfügbarkeit einer Technologie, die als erweiterte Kommunikationsschnittstelle bereits jetzt in das Major-Release für 2016 eingeplant wird (Bild 5-6).

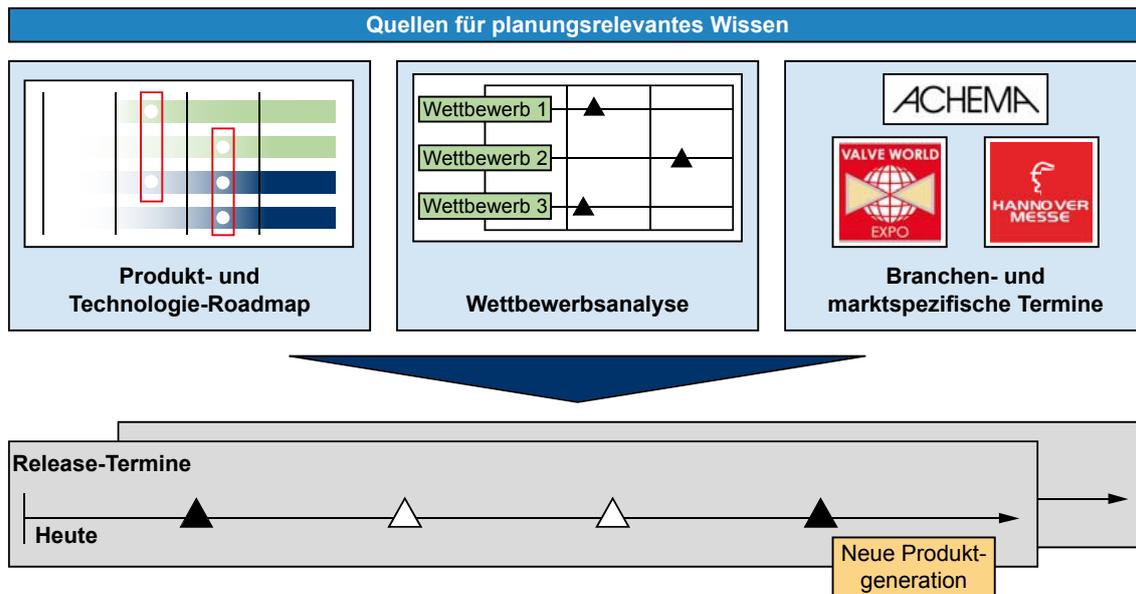


Bild 5-7: Informationsquellen für die Planung kundenrelevanter Releases (prinzipielle Darstellung)

Wettbewerbsanalyse: Neben der eigenen Produkt- und Technologieplanung beeinflussen Wettbewerbsaktivitäten die Veröffentlichung kundenrelevanter Produkt-Releases. Diesem Aspekt kann mit einer eigenen Wettbewerbs-Roadmap Rechnung getragen werden, die Informationen über Wettbewerbsaktivitäten (z.B. über voraussichtliche Markteinführungstermine) in eine strukturierte und zeitliche Abfolge bringt. Da es sich hierbei meist um vertrauliche Informationen handelt, ist deren Beschaffung aufwändig. Quellen sind Patentanmeldungen der Konkurrenz, Beiträge in Fachzeitschriften sowie Informationen aus Gesprächen mit Lieferanten und Kunden. Wenngleich diese Informationen nie zu einer exakten und vollständigen Wettbewerbs-Roadmap führen, können sie dennoch die eigene Planung kundenrelevanter Releases maßgeblich beeinflussen [NL08, S. 316f.]. Ein Beispiel für die prinzipielle Struktur einer Wettbewerbs-Roadmap liefert Anhang A3.3.

Berücksichtigung branchen- und marktspezifischer Termine: Ein weiterer Erfolgsfaktor bei der Planung der Major-Releases ist eine umfassende Berücksichtigung branchen- und marktspezifischer Termine. Typischerweise handelt es sich hierbei um relevante Messen und saisonale Abhängigkeiten, die maßgeblich die Vermarktung kundenrelevanter Produkt-Releases beeinflussen. So bietet die geschickte Vermarktung im besten Fall das Potential, eine eher inkrementelle Verbesserung zu einem Markterfolg zu machen [Ebe06, S. 849], [Sch13, S. 212f.]. Im Anwendungsbeispiel in Bild 5-6 sind die Major-Releases daher entsprechend branchenrelevanter Messen terminiert. Insbesondere die Einführung der neuen Produktgeneration in 2018 entfällt auf die Leitmesse des Unternehmens, so dass die Produkteinführung in das unternehmensweite Marketingkonzept eingebettet werden kann. Für einige Branchen und Produkte bietet es sich darüber hinaus an, die Planung der Major-Releases in einem **externen Release-Plan** ausgewählten Kun-

den zu kommunizieren. Dieser Release-Plan enthält nur wenige Details und ist im weitesten Sinne unverbindlich. Nichtsdestotrotz bietet er die Möglichkeit, frühes Feedback von Kunden und weiteren Partnern zu sammeln und in die weiteren Planungen einfließen zu lassen [GE13, S. 193f.].

5.1.3.2 Planung der Minor-Releases

Zeitversetzt zur Planung der Major-Releases sind in einem abschließenden Schritt die **Minor-Releases** zu planen. Diese dienen der Einführung weniger aufwändigerer Änderungen, die z.B. der Fehlerbehebung, der Berücksichtigung von Produktionsanforderungen oder der Kosteneinsparung dienen. Diese Änderungen sind nicht Teil der Vermarktungsstrategie des Unternehmens und können somit in regelmäßigen Abständen in die Produktionsumgebung überführt werden. Entscheidend ist die Häufigkeit, mit der Minor-Releases eingeführt werden. Hierfür ist die in Phase 1 festgelegte interne Änderungswahrscheinlichkeit ein erster Indikator. Durch eine Einordnung des Produkts in den technologischen Produktlebenszyklus (vgl. Abschnitt 2.1.5) kann dies untermauert werden. Bei einem noch jungen Produkt befindet sich die Kerntechnologie womöglich noch in der Wachstumsphase, so dass erhebliches Potential für inkrementelle Verbesserungen in Form neuer Produktversionen zu erwarten ist. In diesem Fall sollten Minor-Releases quartalsweise bis halbjährlich erfolgen. Dies gilt insb. im Anschluss an die Einführung einer neuen Produktgeneration, bei der gerade in der Hochlaufphase relevante Änderungen z.B. aus der Produktion zu erwarten sind. Handelt es sich hingegen bereits um eine reife Kerntechnologie, sind im weiteren Lebenszyklus nur noch wenige interne Verbesserungspotentiale zu erwarten. Die Frequenz der Minor-Releases kann dann z.B. auf einen jährlichen Turnus reduziert werden. Durch seinen elektrischen Antrieb basiert das Anwendungsbeispiel Regelventil auf einer noch jungen Kerntechnologie. Somit ist davon auszugehen, dass Verbesserungspotentiale weiterhin zur kontinuierlichen Weiterentwicklung beitragen werden. Minor-Releases sind daher im initialen Release-Plan alle sechs Monate eingeplant.

Es wird deutlich, dass das Vorgehen zur Strukturierung des Release-Plans vielfältigen externen wie auch internen Einflussfaktoren unterliegt, die immer wieder Anpassungen und Korrekturen fordern. Beispiele sind Wettbewerbsaktivitäten und neue Technologien. Beim Release-Plan handelt es sich daher um ein „lebendes“ Dokument, das in regelmäßigen Abständen in einem systematischen Planungsprozess aktualisiert werden muss. Die Phase 3 *Strukturierung des Release-Plans* ist daher mindestens in einem jährlichen Turnus zu durchlaufen.

5.2 Taktische Release-Planung

Aufgabe der taktischen Release-Planung ist die Zuordnung von Änderungen zu Releases. Ziel ist somit die Terminierung der Änderungsumsetzung. Es handelt sich um einen kontinuierlichen Planungsprozess, der für jede Änderung individuell durchlaufen wird.

Der Prozess entspricht einem Vorgehen mit drei Analyseschritten, das der Erhebung der planungsrelevanten Informationen über eine Änderung dient. Die drei Schritte reichen von der technischen Änderungsanalyse bis hin zur Änderungsklassifizierung. In den **Abschnitten 5.2.1 bis 5.2.3** wird das Vorgehen exemplarisch durchlaufen; die Erhebung aller für die Umsetzungsplanung erforderlichen Informationen wird aufgezeigt. Im Ergebnis ist jede Änderung einem vorläufigen Release-Termin zugeordnet, den es im Rahmen der operativen Release-Planung zu verfeinern und zu bestätigen gilt.

Ausgangspunkt für die taktische Release-Planung ist ein **Änderungs-Pool**, der kontinuierlich mit Änderungsanforderungen sowie noch nicht realisierten Produkt-Features gefüllt wird. Quellen für Änderungen sind z.B. die strategische Produktplanung, Verbesserungsvorschläge aus dem Service, Änderungen von Gesetzen, Normen bzw. Richtlinien und Produktmängel. Entsprechend Bild 5-8 sind sämtliche Änderungen systematisch zu erfassen.

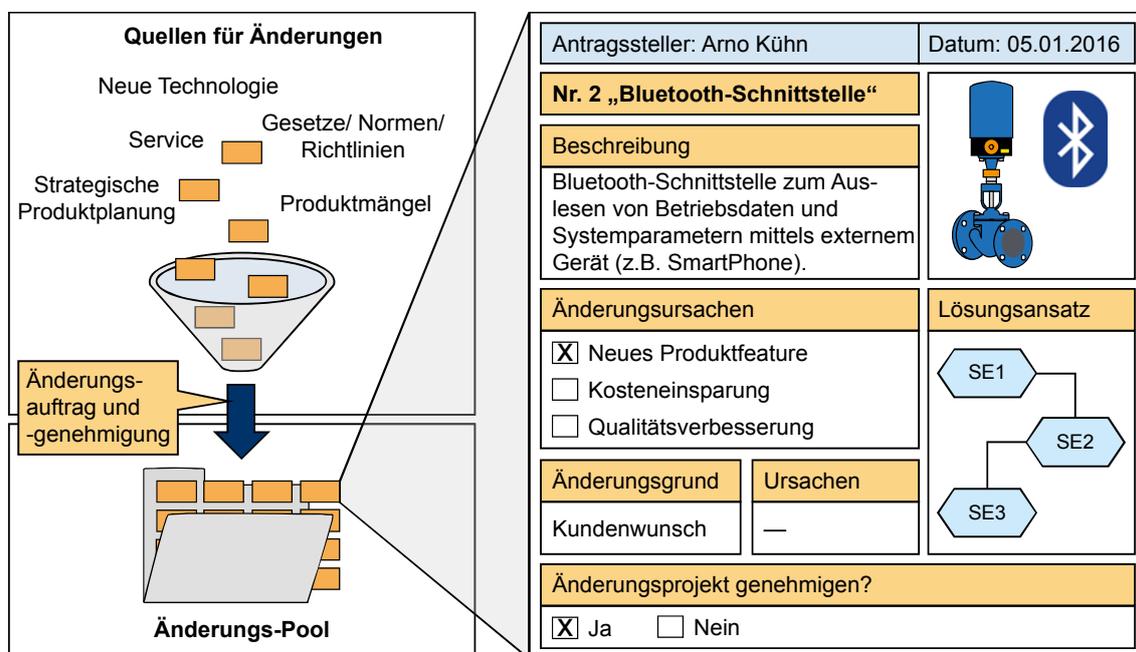


Bild 5-8: Standardisierte Aufnahme und Genehmigung von Änderungen

Zur Aufnahme von Änderungen in den Änderungs-Pool liegen im Unternehmen typischerweise Templates vor, die relevante Informationen wie Änderungsursache und -grund sowie einen ersten Lösungsansatz erfassen. Für die Änderungsumsetzung sind anschließend alternative Lösungsansätze zu erarbeiten und zu bewerten⁶⁷. Die Bewertung ist Entscheidungsgrundlage für die Genehmigung des favorisierten Lösungsansatzes. Mit

⁶⁷ Die Änderungsbewertung als Grundlage für die Änderungsgenehmigung ist unternehmensindividuell festzulegen. Typischerweise wird unter Berücksichtigung von Kriterien wie Kosten, Nutzen und Risiko über die Änderungsumsetzung entschieden. Dazu werden vordefinierte Genehmigungsprozesse durchlaufen, die sich je nach Änderungstyp grundlegend im Aufwand unterscheiden können. Die Änderungsbewertung als Basis für die Änderungsgenehmigung ist nicht Gegenstand dieser Arbeit. Hier sei auf relevante Literatur verwiesen (z.B. [LR98, S. 204ff.], [Bel08, S. 76ff.]).

der Änderungsgenehmigung stellt sich die Frage nach dem richtigen Umsetzungszeitpunkt. Diese wird im Rahmen der taktischen Release-Planung beantwortet.

Voraussetzung sowohl für die Änderungsgenehmigung als auch für die anschließende Umsetzungsplanung ist, dass die Änderungsvorschläge einen vergleichbaren Konkretisierungsgrad aufweisen. Wenig fundierte Produktideen und Verbesserungsvorschläge ohne Lösungsansatz sind daher zuvor auf Umsetzbarkeit innerhalb eines Entwicklungsprojekts zu überprüfen, in ein Lösungskonzept zu überführen und erst dann in den Änderungs-Pool aufzunehmen. Gerade bei neuen Technologien erfolgt dies typischerweise in Vorentwicklungs- bzw. Technologieentwicklungsprojekten (vgl. Abschnitt 2.4.4). Diese sind der Änderungsumsetzung vorgelagert und mit der Release-Planung zu synchronisieren.

Sowohl die Änderungsbewertung als auch die anschließende Umsetzungsplanung sind Aufgabe einer sogenannten **Routinerunde**⁶⁸. Hierbei handelt es sich um ein interdisziplinäres Team bestehend aus Produktmanager, Änderungsmanager, Systems Engineer und Teilnehmern der einzelnen Fachbereiche wie Softwaretechnik, Mechanik etc. Fallspezifisch kann das Team um weitere Spezialisten erweitert werden (z.B. aus der Fertigung). Die Routinerunde analysiert, bewertet und klassifiziert sämtliche Änderungen mit dem Ziel der Genehmigung und Terminierung der Änderungsumsetzung.

5.2.1 Phase 1: Technische Änderungsanalyse

Entscheidend für die systematische Umsetzungsplanung von Änderungen ist die Kenntnis über die Charakteristika einer Änderung einschließlich deren Auswirkungen auf das bestehende technische System. Im Rahmen der **technischen Änderungsanalyse** werden diese Informationen erhoben. Dazu sind folgende Fragen zu beantworten:

- Handelt es sich um eine Änderung an einem Bauteil bzw. einer Komponente oder ist gar ein ganzes Modul zu überarbeiten?
- Reicht eine geringfügige Anpassung der betroffenen Elemente aus oder ist eine Neuentwicklung erforderlich?
- Inwieweit wirkt sich die Änderung über bestehende Schnittstellen auf angrenzende Systemelemente aus (Änderungsfortpflanzung)?
- Welche Fachbereiche sind von der Änderung betroffen? Handelt es sich um eine disziplinspezifische oder um eine disziplinübergreifende Änderung?
- Inwieweit sind Varianten von der Änderung betroffen und wie wirkt sie sich auf die Produktion bzw. auf den Lieferanten aus?

⁶⁸ Der Begriff Routinerunde ist angelehnt an BELENER [Bel08, S. 84].

- Bestehen Abhängigkeiten zwischen Änderungen, die eine gemeinsame Umsetzung zwingend erforderlich machen?

Die Gesamtheit dieser Charakteristika beeinflusst maßgeblich die Zuordnung zu einem passenden Release. Handelt es sich um eine geringfügige Änderung ohne Auswirkungen auf angrenzende Komponenten, bietet sich bspw. die Umsetzung innerhalb eines Minor-Releases an (geringe Änderungskomplexität). Bei umfangreicheren Änderungen erfolgt die Umsetzung innerhalb eines Major-Releases oder gar durch die Ablösung der bestehenden Produktgeneration (hohe Änderungskomplexität). Ziel dieser Phase ist die Bestimmung der Änderungsauswirkung und -abhängigkeiten, die im weiteren Verlauf der taktischen Release-Planung zur Änderungsklassifizierung dienen und damit die Zuordnung zu einem Release ermöglichen.

Die Beantwortung der genannten Fragen setzt ein umfassendes Systemverständnis und ein systematisches Vorgehen bei der Änderungsanalyse voraus. Als Hilfsmittel dienen das Systemmodell (vgl. Abschnitt 4.5.3) und das in Bild 5-9 dargestellte Vorgehen.

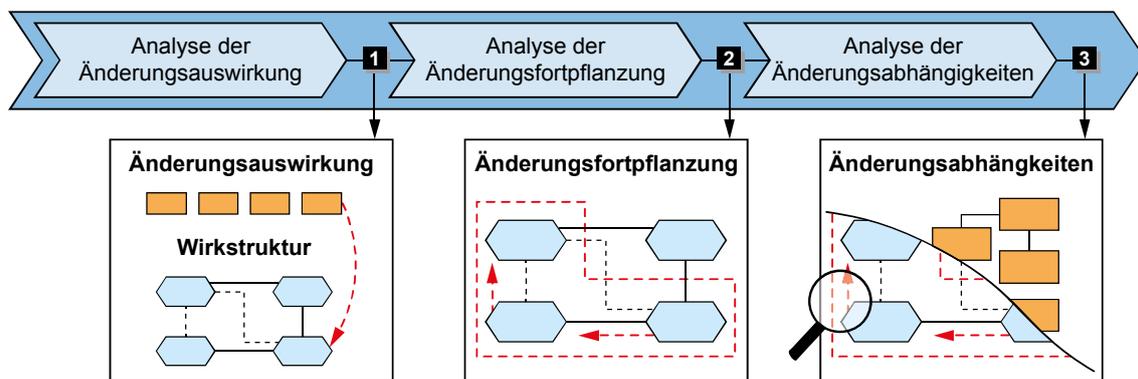


Bild 5-9: Vorgehen bei der technischen Änderungsanalyse

Während die Schritte 1 und 2 zum grundlegenden Verständnis der Änderung und ihrer Auswirkungen beitragen, werden Änderungsabhängigkeiten in einem abschließenden dritten Schritt ermittelt. Das Systemmodell unterstützt hierbei maßgeblich die Kommunikation, Koordination und Dokumentation in einem interdisziplinären Team. Im Folgenden werden die drei Schritte einschließlich der Nutzung des Systemmodells exemplarisch beschrieben.

5.2.1.1 Analyse der initialen Änderungsauswirkung

Im Rahmen der technischen Änderungsanalyse ist zu überprüfen, inwieweit sich eine Änderung auf die verschiedenen Fachdisziplinen wie Softwaretechnik, Elektrotechnik/Elektronik und Mechanik auswirkt. Dies gilt es, in einem interdisziplinären Team unter Teilnahme der verschiedenen Fachbereiche zu überprüfen und zu planungsrelevanten Informationen zu verdichten. Voraussetzung ist ein gemeinsames Systemverständnis, für das sich die Diskussion der Änderung am Systemmodell anbietet (vgl. Abschnitt 4.5.3). Insbesondere der Wirkstruktur kommt hierbei eine gewichtige Rolle zu, da sie

sowohl physische als auch nicht-physische Systemelemente vereint und so die verschiedenen technischen Disziplinen gleichermaßen berücksichtigt. Als Diskussionsgrundlage ist die Änderung daher in einem ersten Schritt in der Wirkstruktur zu verorten.

Die initiale Zuweisung zu einem oder mehreren Systemelementen in der Wirkstruktur unterstützen Tracelinks, die durch die Verknüpfung von Informationen die Rückverfolgbarkeit (Traceability) von Systemzusammenhängen gewährleisten (vgl. Abschnitt 2.5.1) [Bei13, S. 37]. Das Vorgehen unterscheidet sich je nach Änderungstyp (Bild 5-10).

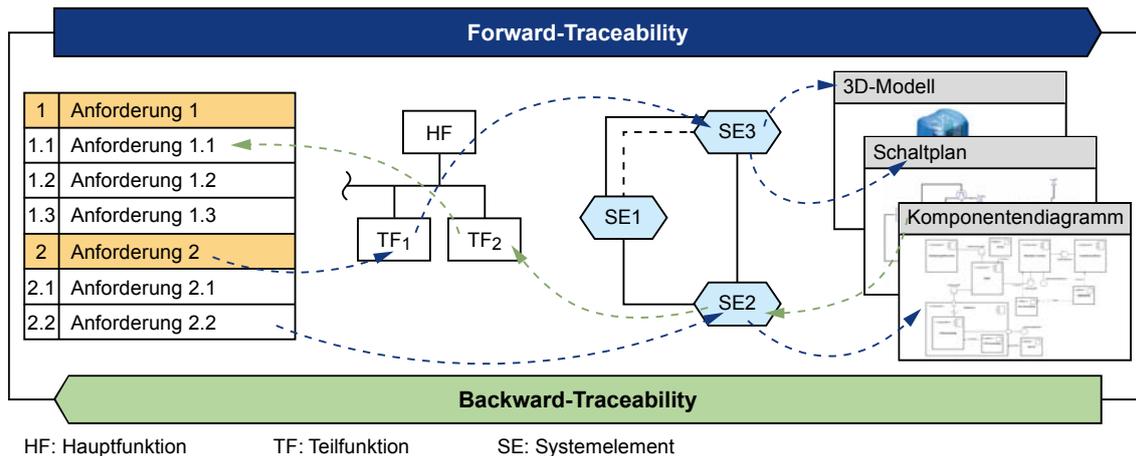


Bild 5-10: Forward- und Backward-Traceability zur Identifikation der initialen Änderungsauswirkung in der Wirkstruktur (prinzipielle Darstellung)

Handelt es sich um eine Anforderungsänderung, kann über Tracelinks direkt oder indirekt (über die Funktionen) auf die betroffenen Systemelemente geschlossen werden (Forward-Traceability). Im Falle fachdisziplinspezifischer Änderungen (z.B. an einer Softwarekomponente) gilt es, die Änderungen über das Systemmodell auf ein Systemelement zurückzuverfolgen (Backward-Traceability). Im Team können dann mögliche Auswirkungen der Änderungen auf andere Fachdisziplinen bzw. weitere Systemelemente diskutiert werden. Ein Sonderfall sind neue Anforderungen bzw. Funktionen. Hier ist entsprechend des Systementwurfs ein neuer Lösungsansatz zu erarbeiten, der entweder zur Veränderung bestehender Systemelemente oder zur Entwicklung neuer führt. In diesem Fall werden die einzelnen Diagramme des Systemmodells ergänzt. Bild 5-11 zeigt drei Änderungen im Kontext des Anwendungsbeispiels Regelventil: die Realisierung einer sichtbaren Zustandsanzeige (rot), einer Bluetooth-Schnittstelle (grün) und einer angepassten Antriebsspindel (blau). Die Änderungen demonstrieren exemplarisch das unterschiedliche Vorgehen bei der Analyse der initialen Änderungsauswirkung.

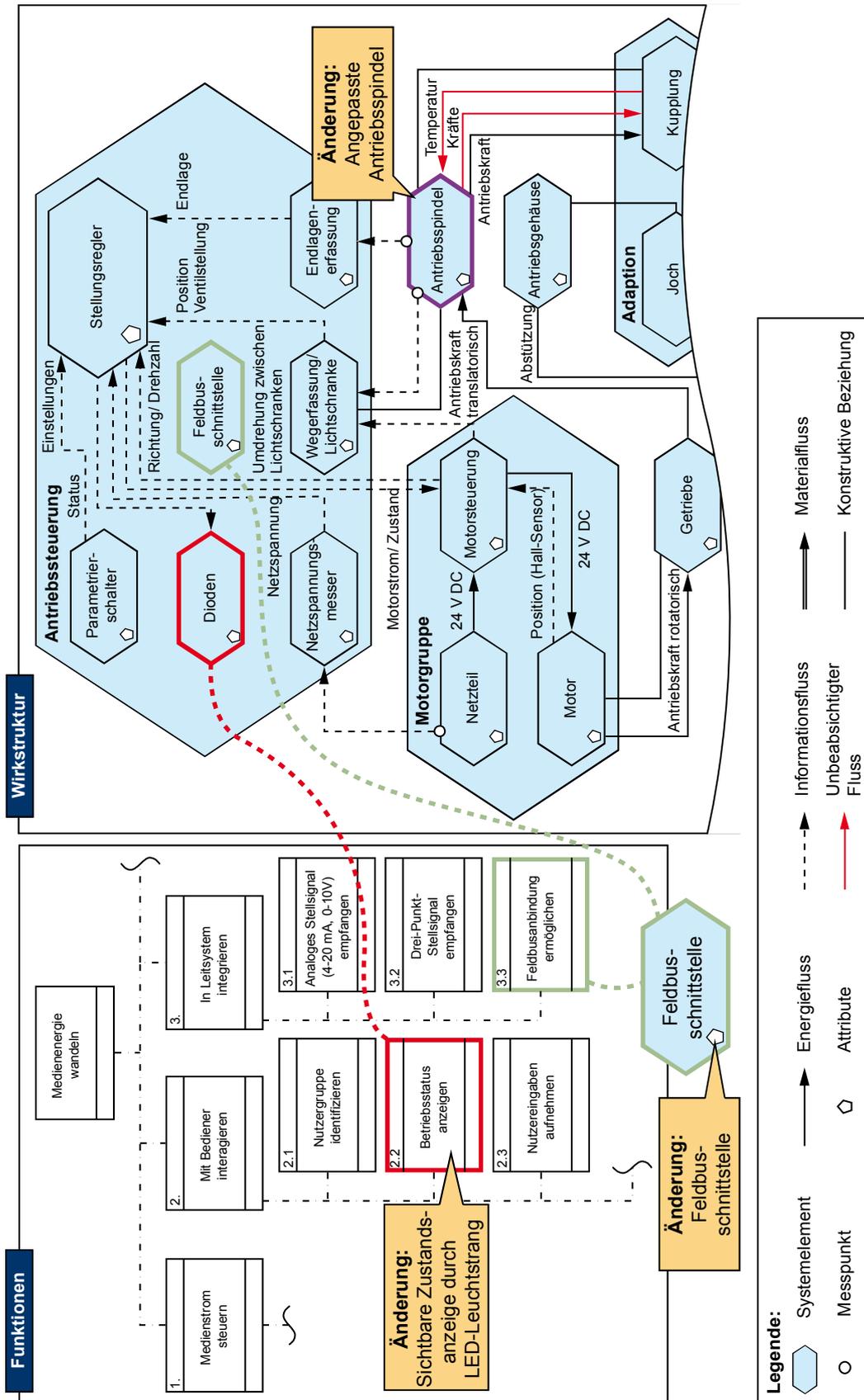


Bild 5-11: Analyse der initialen Änderungsauswirkung (Auszug aus Funktionshierarchie und Wirkstruktur des Regelventils)

Auf Grundlage des Systemmodells wird im Workshop die initiale Änderungsauswirkung jeder Änderungsanforderung analysiert. Die Änderung *sichtbare Zustandsanzeige* ändert die Funktion *Betriebsstatus anzeigen*. Bisher wird die Funktion durch einfache *Dioden* erfüllt, die nicht sichtbar auf der zentralen Antriebssteuerung unter dem Gehäuse des Antriebs installiert sind. Zukünftig soll ein im Gehäuse integrierter *LED-Leuchtstrang* den Betriebsstatus auch von außen sichtbar anzeigen. Ausgehend von der Funktion *Betriebsstatus anzeigen* kann somit über den zugehörigen Tracelink direkt auf das von der Änderung betroffene Systemelement *Dioden* geschlossen werden. Die Änderung *Feldbuschnittstelle* führt hingegen zu einer neuen Funktion, die die Integration eines zusätzlichen Systemelements erfordert. Es ist zu überprüfen, wie sich die Integration des Systemelements *Feldbuschnittstelle* auf das Gesamtsystem auswirkt. Bei der Änderung *angepasste Antriebsspindel* handelt es sich um eine fachdisziplinspezifische Änderung, die aus der Mechanik resultiert. Die initiale Änderungsauswirkung wird im Workshop an der *Antriebsspindel* verortet.

5.2.1.2 Analyse der Änderungfortpflanzung

Im zweiten Schritt ist ausgehend von der initialen Änderungsauswirkung die Änderungfortpflanzung zu überprüfen. So kann es sich um lokale Änderungen an einem Systemelement handeln, um nicht-lokale Änderungen durch systemelementübergreifende Änderungfortpflanzung sowie architektonische Änderungen durch vollständig neue Systemelemente. Als Grundlage für die Analyse der Änderungfortpflanzung dient erneut die Wirkstruktur, die zwecks Dokumentation der Änderungfortpflanzung um eine matrixbasierte Darstellung ergänzt wird. Dies ist in Bild 5-12 exemplarisch für die Änderung *angepasste Antriebsspindel* dargestellt.

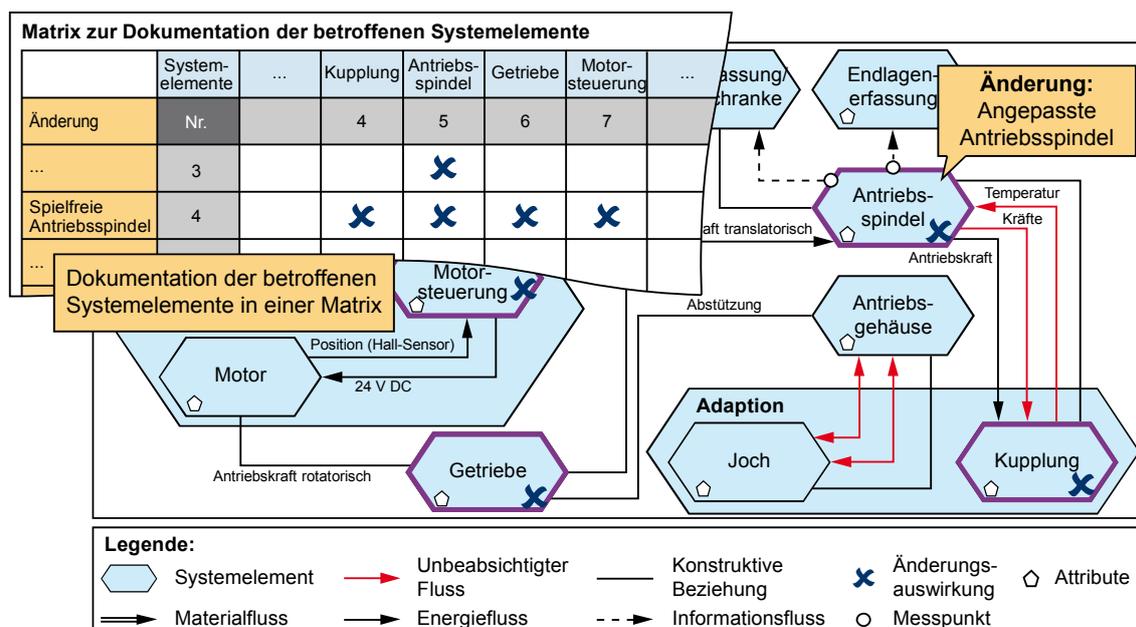


Bild 5-12: Analyse und Dokumentation der Änderungfortpflanzung

Ausgehend vom initial betroffenen Systemelement *Antriebsspindel* ist die Änderungsfortpflanzung auf weitere Systemelemente zu überprüfen. Im Workshop werden dazu sämtliche Systemelemente analysiert, die durch ein- bzw. ausgehende Beziehungen in direkter Wechselwirkung mit der *Antriebsspindel* stehen. Im Beispiel handelt es sich um die *Kupplung*, das *Getriebe*, die *Wegerfassung* und die *Endlagenerfassung*. Die konstruktive Anpassung der *Antriebsspindel* resultiert in Änderungen an *Kupplung* und *Getriebe*. *Wegerfassung* und *Endlagenerfassung* bleiben hingegen unverändert. Neben der direkten Änderungsfortpflanzung gilt es in einem zweiten Schritt, auch indirekte Änderungen an Systemelementen zu antizipieren, die keine direkte Wechselwirkung zum initial betroffenen Systemelement aufweisen. In der Analyse stellt sich heraus, dass durch die Anpassung des *Getriebes* auch die *Motorsteuerung* geringfügig angepasst werden muss. Die von der Änderung betroffenen Systemelemente werden in einer Matrix dokumentiert.

5.2.1.3 Analyse der Änderungsabhängigkeiten

Aufbauend auf der vorangegangenen Änderungsanalyse sind in einem dritten Schritt potentielle Abhängigkeiten mit anderen Änderungen zu identifizieren. Auch hier dient die Wirkstruktur als Diskussionsgrundlage, um im Expertenteam sämtliche Änderungen im Hinblick auf drei Typen von Abhängigkeiten zu überprüfen (vgl. Abschnitt 4.5.3):

- **Abhängigkeiten im Aufwand:** Reduziert sich der Aufwand für die Umsetzung einer Änderung, wenn vorher eine andere Änderung umgesetzt wird?
- **Abhängigkeiten im Nutzen:** Setzt eine Änderung eine andere Änderung voraus, um ihren eigentlichen Nutzen zu entfalten?
- **Technische Abhängigkeiten:** Setzt eine Änderung technisch bedingt eine andere Änderung voraus?

Vor dem Hintergrund dieser Abhängigkeiten werden in einer Matrix die Änderungen paarweise miteinander bewertet mit der Fragestellung „Setzt eine Änderung i (Zeile) eine andere Änderung j (Spalte) voraus?“ Ein „X“ wird immer dann eingetragen, wenn die Frage positiv beantwortet wird, z.B. setzt die Änderung der *Antriebsspindel* die Änderung *neuer Stellantrieb* voraus. Andernfalls wird die Zelle freigelassen. Bild 5-13 zeigt exemplarisch einen Auszug aus der Änderungsabhängigkeitsmatrix für das untersuchte Anwendungsbeispiel.

Der Auszug aus der Änderungsabhängigkeitsmatrix kann so interpretiert werden: Die Änderungen *Simulation Fehlerzustand*, *Feldbusanbindung* und *Parametrierung über Bluetooth-Schnittstelle* werden in Teilen durch neue Funktionsbausteine in der Software realisiert. Schon deshalb macht es Sinn, die geplante Änderung einer *neuen Softwarearchitektur* im Vorfeld umzusetzen, um den Aufwand für die Einbindung der Funktionsbausteine zu reduzieren. Es handelt sich um eine Abhängigkeit im Aufwand. Zur Umsetzung der Änderung *Parametrierung über Bluetooth-Schnittstelle* ist darüber hinaus ein

überarbeitetes Bedieninterface erforderlich. So muss sichergestellt werden, dass die Parametrierung über Bluetooth zu keinen Konflikten mit den direkt am System vorgenommenen Einstellungen führt. Es handelt sich um eine Abhängigkeit im Nutzen.

Änderungsabhängigkeitsmatrix										
Setzt eine Änderung i (Zeile) eine Änderung j (Spalte) voraus?										
<input checked="" type="checkbox"/> = ja <input type="checkbox"/> = nein, die Änderungen sind unabhängig										
Änderung (i)	Änderung (j)	Angep. Antriebsspindel	Neue Softwarearchitektur	Simulation Fehlerzustand	Feldbusanbindung	Bluetooth-Schnittstelle	Neuer Stellantrieb	Regelung	Überarb. Bedieninterface	
Nr.		4	6	7	10	11	16	17	18	
Angep. Antriebsspindel	4	■					✕			
Neue Softwarearchitektur	6		■							
Simulation Fehlerzustand	7		✕	■	✕					
Feldbusanbindung	10		✕		■					
Parametrierung Bluetooth	11		✕			■				✕
Neuer Stellantrieb	16						■	✕		
Regelung	17							■		
Überarb. Bedieninterface	18								■	

Bild 5-13: Auszug aus der Änderungsabhängigkeitsmatrix für das Anwendungsbeispiel

Aus der Dokumentation der Änderungsabhängigkeiten zwischen der Änderung j und der Änderung i ergibt sich eine Umsetzungsrangfolge, die im weiteren Verlauf der Release-Planung zwingend zu berücksichtigen ist. So folgen für die spätere Bündelung der Änderungen zu einem Release drei mögliche Konsequenzen: 1) Die Änderungen i und j werden gemeinsam in einem Release umgesetzt, 2) nur Änderung j wird umgesetzt und 3) das Release enthält keine der beiden Änderungen i und j. Diese technischen Abhängigkeiten werden in Abschnitt 5.3.2 als Restriktionen bei der Bündelung von Änderungen zu Releases wieder aufgegriffen.

5.2.2 Phase 2: Änderungsbewertung

Aufbauend auf den Ergebnissen der technischen Änderungsanalyse erfolgt in Phase 2 eine qualitative Bewertung jeder einzelnen Änderung. Grundlage hierfür sind die zuvor erlangten Ergebnisse sowie das Fachwissen und die Erfahrung der interdisziplinären Routerunde. Die Bewertung erfolgt mittels spezifischer Bewertungsschemata in den Dimensionen Änderungskomplexität, Änderungsrelevanz und Produktwertsteigerung. Darüber hinaus werden organisatorische Aspekte berücksichtigt. Die Bewertungssche-

mata vereinfachen die Entscheidungsfindung im Falle konkurrierender Stakeholder-Präferenzen und verbessern die Reproduzierbarkeit der Änderungsbewertung. Über die Auswahl und Gewichtung der Kriterien sind die Schemata produkt- oder zumindest unternehmensspezifisch anzupassen. Im Ergebnis liegen sämtliche Informationen für die Änderungsklassifizierung und Release-Zuordnung in Phase 3 vor. Im Folgenden wird jede Bewertungsdimension einzeln erläutert.

Bewertung der Änderungskomplexität

Die Änderungskomplexität ist ein Indikator für den Grad der Veränderung des Systems und den mit der Umsetzung der Änderung verbundenen Aufwand. Sie beeinflusst maßgeblich, ob die Änderung in einem Minor-Release, in einem Major-Release oder im Rahmen einer neuen Produktgeneration umgesetzt wird. Zur Bewertung der Änderungskomplexität werden die fünf Kriterien Änderungsauswirkung, Änderungfortpflanzung, Interdisziplinarität, betroffene Varianten und Auswirkung auf Produktion/Lieferant herangezogen. Ausschlaggebend für die Bewertung ist die Diskussion der beteiligten Fachleute im Rahmen der technischen Änderungsanalyse aus Phase 1.

- **Änderungsauswirkung:** Bewertet wird hier der Änderungsgrad der betroffenen Systemelemente. Berücksichtigt wird, ob es sich vornehmlich um wenige einfache Anpassungen oder gar um eine vollständige Überarbeitung handelt.
- **Änderungfortpflanzung:** Neben dem Änderungsgrad wird hier bewertet, inwieweit sich die Änderung auf angrenzende Systemelemente auswirkt. Dies reicht von einer lokalen Änderung bis hin zu grundlegenden architektonischen Änderungen, die die Anpassung weiterer Systemelemente zur Folge haben.
- **Interdisziplinarität:** Mit der Anzahl der an der Änderungsumsetzung beteiligten Fachdisziplinen steigt der Kommunikationsaufwand und damit auch die Komplexität der Änderung. Diesem Aspekt wird mit dem Kriterium *Interdisziplinarität* Rechnung getragen. Unterstützt wird dieser Bewertungsschritt durch das Attribut *beteiligte Fachdisziplinen*, das jedem Systemelement im Systemmodell zugewiesen ist (vgl. Abschnitt 4.5.3).
- **Betroffene Varianten:** Dieses Kriterium nimmt Bezug auf die Mehrfachverwendung der betroffenen Systemelemente in verschiedenen Varianten. Dies ist im Systemmodell dem Attribut *Mehrfachverwendung* zu entnehmen (vgl. Abschnitt 4.5.3). Im einfachsten Fall sind keine weiteren Varianten von der Änderung betroffen. Komplexer wird es, wenn von der Änderung eine Vielzahl weiterer Varianten betroffen sind, die individuell angepasst werden müssen.
- **Auswirkung auf Produktion:** Mit diesem Kriterium werden Änderungsauswirkungen auf den Fertigungs- und Montageprozess berücksichtigt. Hierbei kann es sich z.B. um Werkzeug-, Vorrichtungs- oder Prüfmitteländerungen handeln. Teilweise können Änderungen ohne grundlegende Anpassungen im Fertigungs-

bzw. Montageprozess umgesetzt werden. Aufwändiger wird es, wenn neue Produktionstechnologien erforderlich sind, die neben hohen Investitionen insb. auch erhöhte Komplexität für die Änderungsumsetzung bedeuten.

Unterstützt wird die Bewertung der fünf Kriterien durch das in Tabelle 5-2 dargestellte Bewertungsschema. Im Beispiel ist die Bewertung der Änderung *angepasste Antriebs- spindel* dargestellt.

Tabelle 5-2: Schema für die abschließende Bewertung der Änderungskomplexität im interdisziplinären Expertenteam (Beispiel: Angepasste Antriebsspindel)

Bewertungsschema Änderungskomplexität	Gew.	Bewertungsmaßstab			
		0	1	2	3
Änderungsauswirkung	0,25	Wenige, einfache Anpassungen	Viele, einfache Anpassungen	Umfassende Überarbeitungen	Vollständige Überarbeitung/ Neuentwicklung
Änderungfortpflanzung	0,25	Lokale Änderung an einzelnen Komponenten	Komponentenübergreifende Änderung an einem Modul	Modulübergreifende Änderung	Architektonische Änderung durch neues Modul
Interdisziplinarität	0,1		Disziplinspezifische Änderung	Zwei betroffene Fachdisziplinen	Mehr als zwei betroffene Fachdisziplinen
Betroffene Varianten	0,2	Keine Varianten betroffen	Ausschließlich einige, wenige Gleichteile betroffen	Kleinere, individuelle Anpassungen der Varianten erforderlich	Umfangreichere, individuelle Anpassungen der Varianten erforderlich
Auswirkung auf Produktion	0,2	Keine Anpassungen erforderlich	Geringfügige Anpassungen des Fertigungs- bzw. Montageprozesses	Umfangreiche Anpassungen des Fertigungs- bzw. Montageprozesses	Neue Produktionstechnologie erforderlich
Änderungskomplexität	1				

Änderung:
Angepasste
Antriebsspindel

Bewertung der Änderungsrelevanz

Neben der Änderungskomplexität beeinflusst auch die Änderungsrelevanz die Zuordnung zu einem Release-Typ. Die Änderungsrelevanz ist ein Indikator für die Priorität, mit der eine Änderung umgesetzt werden muss. Bspw. werden Änderungen mit besonders hoher Relevanz in Sofortmaßnahmen umgesetzt. Typischerweise werden Änderungen jedoch sowohl von internen als auch von externen Stakeholdern als unterschiedlich relevant betrachtet. Hinzu kommt das von BELENER beschriebene Phänomen, *dass vermeintlich dringende Änderungen häufig höchste Priorität genießen, diese jedoch selten besonders wichtig sind* [Bel08, S. 76]. Für die Release-Planung ist daher eine systematische Bewertung der Änderung im Hinblick auf Wichtigkeit und Dringlichkeit erforderlich. Die Wichtigkeit ergibt sich ausgehend vom Wert der Änderung für das Unternehmen (z.B. durch Kosteneinsparpotentiale, Qualitätsverbesserungen oder gesteigerten Kundennutzen). Die Dringlichkeit berücksichtigt hingegen den Faktor Zeit und damit die Notwendigkeit der Änderungsumsetzung [Bel08, S. 76f.]. Zur Bewertung der beiden Dimensionen wird das in Tabelle 5-3 dargestellte Bewertungsschema herangezogen.

Tabelle 5-3: Bewertungsschema für die Änderungsrelevanz einer Änderung in Anlehnung an [Bel08, S. 148]

Kriterium	Unterkriterium	Änderungsrelevanz	
		Wichtigkeit	Dringlichkeit
Produktfehler	Gefahr für Personen oder Sachen	3 = ja 0 = nein	3 = ja 0 = nein
	Funktionsfähigkeit nicht gewährleistet	3 = ja 0 = nein	3 = ja 0 = nein
	Qualitätsbeeinträchtigung bei Kunden	3 = hoch (direkt sichtbar) 1 = gering (kaum erkennbar)	3 = Häufigkeit größer 10% 1 = Einzelfall
Gesetze, Normen, Richtlinien	Verkauf gefährdet (Approbiation gefährdet)	3 = ja 0 = nein	3 = Kurzfristig erforderlich bis 1 = Zeitpuffer vorhanden
	Verkauf beeinträchtigt (aber weiter möglich)	2 = stark bis 0 = kaum	3 = Kurzfristig erforderlich bis 1 = Zeitpuffer vorhanden
Veränderte Kundenanforderung (nach KANO-Modell)	Ist bzw. wird Basisanforderung	3 = ja 0 = nein	3 = Wettbewerb bereits am Markt bis 1 = noch nicht am Markt absehbar
	Ist bzw. wird Leistungsanforderung	2 = ja 0 = nein	3 = Wettbewerb bereits am Markt bis 0 = noch nicht am Markt absehbar
	Ist bzw. wird Begeisterungsanforderung	1 = ja 0 = nein	3 = Wettbewerb bereits am Markt bis 0 = noch nicht am Markt absehbar
Imageverbesserung		2 = stark bis 0 = kaum	1 = ja 0 = nein
Rationalisierungs- oder Kosteneinsparpotential		> 50.000 EUR p.a. = 3 > 25.000 EUR p.a. = 2 > 10.000 EUR p.a. = 1 < 10.000 EUR p.a. = 0	> 50.000 EUR p.a. = 3 > 25.000 EUR p.a. = 2 > 10.000 EUR p.a. = 1 < 10.000 EUR p.a. = 0

Im Bewertungsschema werden fünf Kriterien unterschieden, die teilweise in weitere Unterkriterien gegliedert sind. Jede Änderung kann ausgehend von ihrer Ursache einem Kriterium zugewiesen werden und im Hinblick auf Wichtigkeit und Dringlichkeit bewertet werden. Fehlerbedingte Änderungen werden bspw. wie folgt unterschieden: Gefahr für Personen bzw. Gegenstände, Funktionsbeeinträchtigung oder Qualitätsbeeinträchtigung. Je nach Gewichtung der beiden Dimensionen Wichtigkeit und Dringlichkeit ergibt sich schließlich ein Wert zwischen 0 und 3 für die Änderungsrelevanz.

Bewertung der Produktwertsteigerung

Ein zentrales Nutzenpotential der Release-Planung liegt in einer gesteuerten und kundenorientierten Produktwertsteigerung. Im Fokus stehen Änderungen, die durch eine verbesserte Bedürfnisbefriedigung und/oder Kosteneinsparungen den Produktwert aus Sicht des Kunden beeinflussen. Diese Änderungen gilt es im Rahmen der Release-Planung über Major-Releases und neue Produktgenerationen gezielt einzusteuern. Ziel ist dabei eine kontinuierliche Produktwertsteigerung: Nicht alle kundenrelevanten Änderungen sollen mit einem Schlag auf den Markt gebracht werden, so dass anschließend eine lange Durststrecke folgt. Vielmehr sollen die Änderungen systematisch auf die verschiedenen kundenrelevanten Releases verteilt werden. Hierfür ist jede Änderung im Hinblick auf ihre

Kundenrelevanz zu überprüfen und in ihrem Beitrag zur Produktwertsteigerung zu bewerten. Für die pragmatische Bewertung der Produktwertsteigerung⁶⁹ einer Änderung aus Sicht des Kunden wird im Rahmen dieser Arbeit auf das Bewertungsschema in Tabelle 5-4 zurückgegriffen. Es unterscheidet die beiden Bewertungskriterien Verbesserung der Bedürfnisbefriedigung und kundenrelevante Kostenreduktion.

Tabelle 5-4: Schema für die pragmatische Bewertung der Produktwertsteigerung einer Änderung

Bewertung der Produktwertsteigerung	Bewertungsmaßstab			
	0	1	2	3
Verbesserung der Bedürfnisbefriedigung	Keine Verbesserung	Geringfügige Verbesserung (Basisanforderung)	Starke Verbesserung (Leistungsanforderung)	Sehr starke Verbesserung (Begeisterungsanforderung)
Kundenrelevante Kostenreduktion	Keine Kostenreduktion	Geringfügige Kostenreduktion	Hohe Kostenreduktion	Sehr hohe Kostenreduktion

- **Verbesserung der Bedürfnisbefriedigung:** Im Expertenteam wird hier antizipiert und bewertet, ob eine Änderung eine verbesserte Bedürfnisbefriedigung zur Folge hat. Dies kann bspw. das Resultat einer neuen Produktfunktion oder ein moderneres Produktdesign sein. Für eine Abstufung orientiert sich die Bewertung am bewährten Kano-Modell (vgl. Abschnitt 3.3.1). Zur Absicherung der Bewertung bieten sich darüber hinaus ergänzende Methoden der Markt- bzw. Kundenforschung an (z.B. Kundenbefragungen, Analyse von Beschwerden und Kundenbeobachtungen).
- **Kundenrelevante Kostenreduktion:** Gemeint sind Rationalisierungs- und Kosteneinsparpotentiale einer Änderung, die durch eine Preisreduktion an den Kunden weitergegeben werden können.

Bewertung organisatorischer Aspekte

In einem letzten Schritt ist die Änderung unter organisatorischen Aspekten zu bewerten. Diese Informationen dienen im Rahmen der operativen Release-Planung als Grundlage für die Bündelung von Änderungen zu Releases, die unter Berücksichtigung von Zeit- und Ressourcenrestriktionen umgesetzt werden können. Hierbei werden zwei wesentliche Kriterien berücksichtigt⁷⁰:

⁶⁹ In der Literatur existieren vielfältige, teils sehr aufwändige Ansätze zur Analyse und Bewertung des Produktwerts. So widmet sich bspw. die Wertanalyse bzw. das Value Engineering der systematischen Analyse des Produktwerts von Bewertungsobjekten [VDI2800]. Da die Entwicklung derartiger Methoden nicht im Fokus dieser Arbeit steht, wird in diesem Fall auf ein pragmatisches Vorgehen zurückgegriffen.

⁷⁰ Alternativ zur absoluten Aufwandsabschätzung ist auch ein einfacher relativer Vergleich der Änderungsaufwände denkbar. In dieser Arbeit wird jedoch davon ausgegangen, dass im Unternehmen für die Kenngrößen Dauer und Kosten einer Änderung entsprechende Schätzverfahren etabliert sind. Die Entwicklung derartiger Verfahren ist nicht Gegenstand dieser Arbeit.

- **Umsetzungsdauer:** Gemeint ist der mindestens erforderliche Zeitraum für die Umsetzung einer Änderung. Dies schließt neben der produktiven Änderungs-umsetzung insb. notwendige Tests und Zertifizierungen ein.
- **Erforderliche Ressourcen:** Hier werden die erforderlichen Ressourcen abge-schätzt, die für die Änderungsumsetzung erforderlich sind. Eine grobe Richtung gibt hier die Bewertungsdimension Änderungskomplexität vor. Die Kalkulation der erforderlichen Ressourcen kann z.B. in Personalaufwand oder in Änderungs-kosten erfolgen.

Sämtliche Bewertungsdimensionen werden schließlich in der sogenannten Änderungsbe-wertungsmatrix konsolidiert. In dieser Matrix wird jede Änderung entlang sämtlicher Kri-terien bewertet. Tabelle 5-5 zeigt einen Auszug aus der Änderungsbewertung des Anwen-dungsbeispiels Regelventil.

Tabelle 5-5: Abschließende Bewertung der Änderungen

Änderungsbewertungsmatrix Bewertung (B) der Änderung (i) entlang der Kriterien (K) auf einer Skala von 0 bis 3. Für eine genauere Erläuterung des Bewer-tungsmaßstabs siehe zugehörige Bewertungs-schemata.	Gewichtung (G)	Änderung (i)											
		Nr. 1 „Sichtbare Zustandsanzeige“		Nr. 2 „Bluetooth-Schnittstelle“		Nr. 3 „Endlagen- erfassung“		Nr. 4 „Angepasste Antriebspindel“		Nr. 5 „Dichtung für Gehäuse“		..	
		B	GxB	B	GxB	B	GxB	B	GxB	B	GxB	B	G
Änderungskomplexität													
Änderungsauswirkung	0,25	1	0,25	2	0,5	2	0,5	3	0,75	0	0		
Änderungsfortpflanzung	0,25	1	0,25	1	0,25	2	0,5	2	0,5	0	0		
Interdisziplinarität	0,1	2	0,2	2	0,2	2	0,2	2	0,2	1	0,1		
Betroffene Varianten	0,2	1	0,2	1	0,2	2	0,4	3	0,6	1	0,2		
Auswirkung Produktion	0,2	2	0,4	1	0,2	1	0,2	3	0,6	0	0		
Summe		1,3		1,35		1,8		2,65		0,3			
Änderungsrelevanz													
Wichtigkeit	0,5	1	0,5	2	1	2	1	1	0,5	3	1,5		
Dringlichkeit	0,5	1	0,5	3	1,5	1	0,5	1	0,5	2	1		
Summe		1		2,5		1,5		1		2,5			
Produktwertsteigerung													
Kundenrelevant		ja		ja		ja		nein		nein			
Verbesserung der Bedürfnisbefriedigung		3		2		1		-		-			
Kundenrelevante Kostenreduktion		0		0		0		-		-			
Summe		3		2		1		-		-			
Organisatorisches													
Minimale Umsetzungsdauer [in Monaten]		6		6		6		12		1			
Erforderliche Ressourcen [in Personenmonaten]		3		4		3		18		0,25			

Aus der Gewichtung der einzelnen Kriterien resultieren einheitenlose Summen, die die Bewertungsdimensionen Änderungskomplexität, Änderungsrelevanz und Produktwert-steigerung quantifizieren. Die organisatorischen Aspekte Umsetzungsdauer und erforderliche Ressourcen werden in Monaten bzw. Personenmonaten abgeschätzt. Im Beispiel

kristallisiert sich insb. die *angepasste Antriebsspindel* mit einer erheblichen Änderungskomplexität heraus. Von besonders hoher Relevanz ist hingegen die *Bluetooth-Anbindung*. Diese ist bei den Wettbewerbern bereits verfügbar und wird daher als besonders dringlich bewertet. Ein besonders hoher Beitrag zur Produktwertsteigerung wird durch die *sichtbare Zustandsanzeige* erwartet. Diese verbessert nicht nur die eigentliche Funktion, sondern trägt als Designelement auch zur Anmutung des Produkts bei. Bei der Änderung *Dichtung für Gehäuse* handelt es sich um eine sehr einfache Änderung, die mit einem geringen Umsetzungsaufwand verbunden ist. Da die Änderung jedoch eine Qualitätsbeeinträchtigung beim Kunden ausmerzen soll, wird sie mit einer sehr hohen Änderungsrelevanz bewertet.

5.2.3 Phase 3: Änderungsklassifizierung und Zuordnung zu Releases

Die Bewertung hinsichtlich Änderungskomplexität und Änderungsrelevanz aus dem vorherigen Abschnitt dient als Eingangsgröße für die in Phase 3 folgende Änderungsklassifizierung. Die Klassifizierung fußt auf dem in der strategischen Release-Planung erarbeiteten Klassifikationsschema (Abschnitt 5.1.2). Darin werden die Änderungen mittels eines Portfolio-Ansatzes einem Release-Typ zugeordnet. Diese Zuordnung dient als Ausgangspunkt für die anschließende Umsetzungsplanung der Änderung durch die zeitliche Verortung im Release-Plan. Im Ergebnis von Phase 3 ist jede Änderung einem Release zugeordnet und somit in ihrer planmäßigen Umsetzung terminiert. Bild 5-14 zeigt einen Auszug der Änderungsklassifizierung aus dem Anwendungsbeispiel Regelventil.

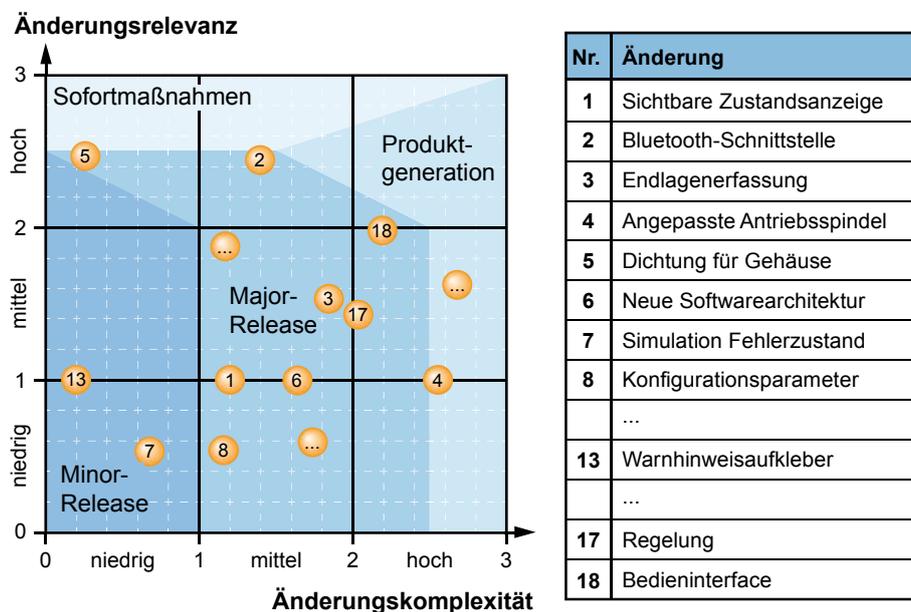


Bild 5-14: Klassifizierung der Änderungen zur Zuordnung zu Release-Typen

Die differenzierte Bewertung der Änderung entlang der beiden Dimensionen Änderungskomplexität und Änderungsrelevanz erlaubt die Zuordnung zu einem von vier Release-Typen. Dabei können zwei Extremfälle unterschieden werden: Im Fall einer sehr hohen

Änderungskomplexität empfiehlt sich die Umsetzung der Änderung im Rahmen einer neuen Produktgeneration. Im anderen Extrem – einer sehr hohen Änderungsrelevanz – ist die Änderung außerplanmäßig mittels Sofortmaßnahme umzusetzen. In den übrigen Fällen erfolgt in Abhängigkeit von Komplexität und Relevanz eine Zuordnung zum nächsten Minor- bzw. Major-Release. Im Anwendungsbeispiel wird ausgehend von der hohen Komplexität der Änderung *angepasste Antriebsspindel* die Umsetzung auf die nächste Produktgeneration verschoben. Mit dem nächsten Major-Release werden hingegen z.B. die Änderungen *sichtbare Zustandsanzeige* und *Bluetooth-Schnittstelle* eingeführt. Die Änderung *Dichtung für Gehäuse* wird aufgrund ihrer hohen Relevanz auf eine sofortige Umsetzung hin überprüft, letztlich aber doch dem nächsten Minor-Release zugeordnet.

Mit der Zuordnung der Änderung zu einem Release-Typ kann die eigentliche Umsetzungsplanung erfolgen. Dazu wird die Änderung anhand des Release-Plans terminiert. Typischerweise erfolgt hierbei eine 1:1-Zuordnung der Änderung entsprechend des Release-Typs. Grundsätzlich kann die Änderung jedoch auch dem nächstgrößeren Release zugeordnet werden. Wird bspw. als nächstes ein Major-Release eingeführt, können diesem auch geringfügige Änderungen zugeordnet werden, die üblicherweise im Rahmen eines Minor-Releases umgesetzt werden würden. Am Ende dieser Phase ist jede Änderung in ihrer planmäßigen Umsetzung terminiert.

Eine Besonderheit ergibt sich bei der Planung kundenrelevanter Änderungen, die in einem Major-Release umgesetzt werden sollen. Die Einführung von Major-Releases erfolgt abgestimmt auf die Erfordernisse von Markt und Kunden mit dem Ziel einer kontinuierlichen Produktwertsteigerung. Um diese sicherzustellen, ist bei der Zuordnung kundenrelevanter Änderung die mit einem Major-Release verbundene Produktwertsteigerung zu überprüfen. Dies ist in Bild 5-15 exemplarisch dargestellt.

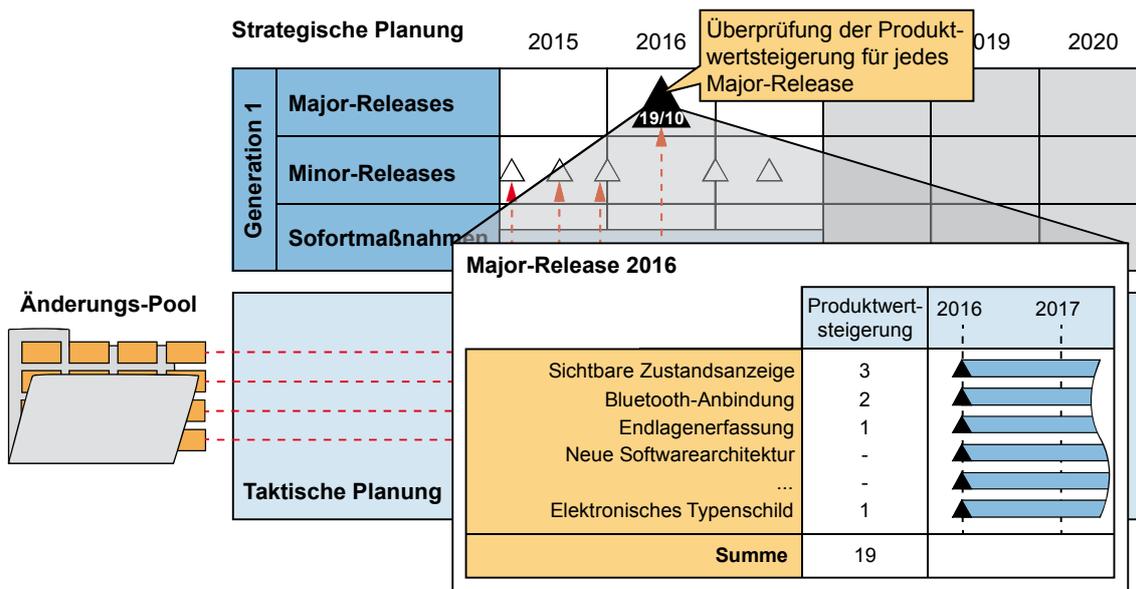


Bild 5-15: Berücksichtigung der Produktwertsteigerung bei der Umsetzungsplanung kundenrelevanter Änderungen

Im Rahmen der Änderungsbewertung wurde für jede Änderung der Beitrag zur Produktwertsteigerung abgeschätzt. Summiert über alle Änderungen ergibt dies die mit einem Major-Release verbundene Produktwertsteigerung. Durch die Festlegung eines Zielwerts für jedes Release kann die Zuordnung kundenrelevanter Änderungen nun gesteuert werden: Enthält das Release bereits ausreichend kundenrelevante Änderungen, kann unter strategischen Gesichtspunkten eine Verschiebung von Änderungen auf ein späteres Release überprüft werden. Eine deutliche Unterschreitung des Zielwerts signalisiert hingegen Handlungsbedarf. Es ist zu überprüfen, in welcher Form die geforderte Produktwertsteigerung erreicht werden kann. In beiden Fällen ist eine enge Abstimmung mit der strategischen Produktplanung und der jeweiligen Produkt-Roadmap erforderlich. Im Anwendungsbeispiel wird der Zielwert bei Weitem überschritten. Kundenrelevante Änderungen können ohne zu zögern auf ein späteres Release verschoben werden. Dies erfolgt im Rahmen der operativen Release-Planung unter der Berücksichtigung entsprechender Restriktionen.

5.3 Operative Release-Planung

Im Rahmen der taktischen Release-Planung werden sämtliche Änderungen klassifiziert und dem nächstmöglichen Release zugeordnet. Unberücksichtigt bleiben dabei Restriktionen (z.B. verfügbare Ressourcen, technische Abhängigkeiten). Insbesondere Ressourcenengpässe können allerdings dazu führen, dass nicht alle Änderungen im Rahmen eines Release umgesetzt werden können. Hier setzt die operative Release-Planung an. Sie fokussiert das nächste, planmäßig umzusetzende Release mit dem Ziel, die Änderungen unter Berücksichtigung sämtlicher Restriktionen zu einem umsetzbaren Release zu bündeln. Die Planung erfolgt auf Basis aller bis zu diesem Zeitpunkt gesammelten Informationen in vier Phasen. Diese werden in den **Abschnitten 5.3.1 bis 5.3.4** exemplarisch erläutert. Im Ergebnis liegen sämtliche Informationen zur Entwicklung und produktiven Umsetzung des Release und den damit verbundenen Änderungen vor.

5.3.1 Phase 1: Erhebung der operativen Rahmenbedingungen

Die Umsetzung eines Release erfolgt in einem Entwicklungsprojekt, das typischen projektspezifischen Rahmenbedingungen unterliegt. Diese gilt es, in einem ersten Schritt zu erheben. Wie Bild 5-16 am Beispiel des Major-Release zeigt, sind hierbei Release-Restriktionen und Planungsziel zu unterscheiden.

Release-Restriktionen schränken aus Sicht des Projektmanagements die Anzahl der Änderungen ein, die innerhalb eines Release umsetzbar sind. Je nach Release-Strategie⁷¹ sind die Restriktionen für jeden Release-Typ fest definiert oder variabel und damit individuell festzulegen. Die **Umsetzungsdauer** definiert den Zeitraum, der zur Umsetzung

⁷¹ Dies ergibt sich aus dem Gestaltungsfaktor Release-Umfang, der im Anwendungsbeispiel die Ausprägung *variabel* hat.

einer Änderung von der Idee bis zum erfolgreichen Test zur Verfügung steht. Die Umsetzungsdauer hängt vom Release-Typ ab. Während für die schnell aufeinanderfolgenden Minor-Releases typischerweise nur wenige Wochen zur Verfügung stehen, erfolgt die Vorbereitung eines Major-Releases parallel mit weitaus mehr Vorlaufzeit (im Beispiel acht Monate). Änderungen, die nicht innerhalb dieses Zeitraums zu realisieren sind, müssen auf ein späteres Release verschoben werden. Die **verfügbaren Ressourcen** schränken hingegen die Anzahl der Änderungen ein, die innerhalb eines Release umgesetzt werden können. Als wesentlicher Richtwert für die Planung wird im Rahmen dieser Arbeit die Personalverfügbarkeit⁷² herangezogen, die für jedes Release abgestimmt auf die unternehmensweit verfügbaren Ressourcen festgelegt werden muss. Ausgehend hiervon können unter Berücksichtigung des Aufwands je Änderung umsetzbare Kombinationen von Änderungen – sogenannte Änderungsbündel – ermittelt werden.

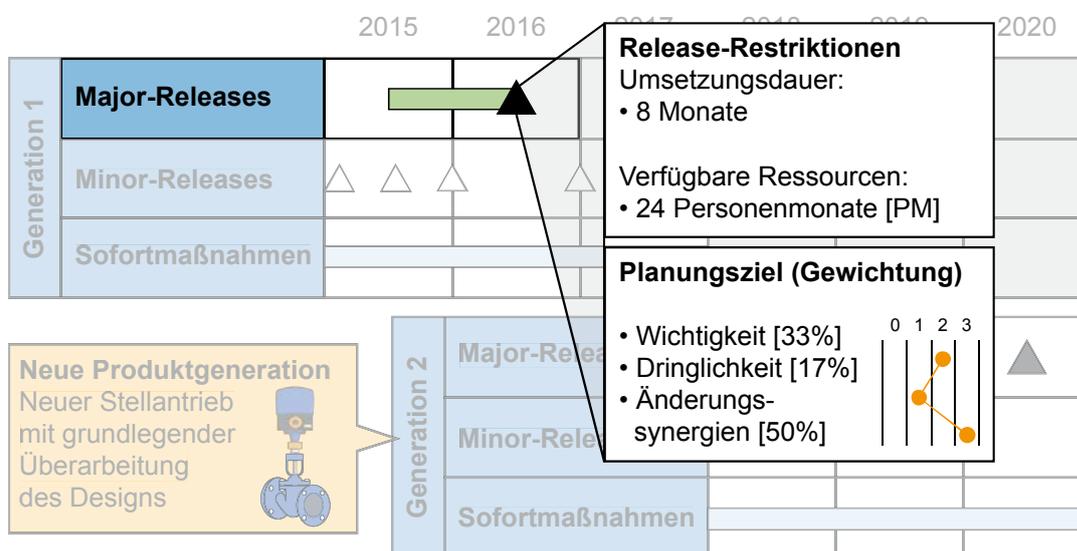


Bild 5-16: Erhebung der operativen Rahmenbedingungen für das nächste Release

Das **Planungsziel** ermöglicht es, die Auswahl des umzusetzenden Änderungsbündels auf strategische Überlegungen abzustimmen. So können mit einem Änderungsbündel bspw. nur die wichtigsten oder dringlichsten Änderungen umgesetzt werden. Zur Festlegung des Planungsziels werden ausgewählte Kriterien gewichtet, die es ermöglichen, für jedes Änderungsbündel einen Nutzwert – den sogenannten Release-Wert – zu ermitteln. Unter Berücksichtigung der Release-Restriktionen gilt es dann, das Änderungsbündel mit dem höchsten Release-Wert auszuwählen. Im Beispiel werden die drei Kriterien⁷³ Wichtigkeit, Dringlichkeit und Änderungssynergien herangezogen und gewichtet. Die Kriterien

⁷² Alternativ kann als Restriktion bspw. auch ein maximales Budget für die Änderungskosten festgelegt werden.

⁷³ Grundsätzlich sind weitere Kriterien zur Priorisierung denkbar (z.B. Kundennutzen, Kosten und Entwicklungsrisiko). Einen umfassenden Überblick liefert Anhang A3.1. Die Berücksichtigung weiterer Kriterien ist für das Verständnis der Systematik jedoch nicht förderlich.

Wichtigkeit und Dringlichkeit ergeben sich aus der Bewertung jeder einzelnen Änderung im Rahmen der taktischen Release-Planung (vgl. Abschnitt 5.2.2). Sie spiegeln die marktstrategische Sicht auf das Release wider. Das Kriterium Änderungssynergien bewertet hingegen die Konsistenz eines Änderungsbündels aus technischer Sicht. Es gilt die Annahme: Je weniger Systemelemente von einem Änderungsbündel betroffen sind, desto höher sind die potentiellen Änderungssynergien. Dies folgt dem Grundgedanken der Release-Planung (vgl. Abschnitt 2.4.2).

5.3.2 Phase 2: Auswahl eines Änderungsbündels

Lassen sich sämtliche Änderungen im Rahmen der verfügbaren Ressourcen umsetzen, entfallen die Phasen 2 und 3. Andernfalls ist in dieser Phase zunächst ein Änderungsbündel zu bestimmen, das mit dem Release umgesetzt werden soll. Der Aufwand zur Änderungsumsetzung darf dabei die verfügbaren Ressourcen nicht überschreiten. Gleichzeitig ist der Release-Wert unter Berücksichtigung des Planungsziels zu maximieren. Da es sich hierbei um ein mathematisches Optimierungsproblem⁷⁴ handelt, wird diese Phase durch ein Softwarewerkzeug⁷⁵ unterstützt. Im Ergebnis liegt eine Empfehlung für die umzusetzenden Änderungen vor. Gleichzeitig sind die Änderungen bekannt, die aufgrund begrenzter Ressourcen vorerst nicht berücksichtigt werden können. Das prinzipielle Vorgehen zur harmonisierten Bündelung der Änderungen ist in Bild 5-17 dargestellt.

Als **Eingangsinformationen** dienen die operativen Rahmenbedingungen und die in der strategischen und taktischen Release-Planung erhobenen Informationen über die einzelnen Änderungen. Während die obligatorischen Änderungen, technische Abhängigkeiten und die verfügbaren Ressourcen die Ermittlung potentieller Änderungsbündel beeinflussen, dienen die Änderungsbewertung und das Planungsziel der anschließenden Priorisierung.

74 Es handelt sich um das sogenannte „Rucksackproblem“ (engl. knapsack problem) – ein Optimierungsproblem aus der Kombinatorik. Ausgangspunkt ist eine Menge von Objekten (Änderungen), die jeweils ein Gewicht (Personalaufwand) und einen Nutzwert (z.B. Dringlichkeit, Wichtigkeit) haben. Hieraus gilt es, eine Teilmenge (Änderungsbündel) auszuwählen, deren Gesamtgewicht eine vorgegebene Grenze (verfügbarer Personalaufwand) nicht überschreitet. Gleichzeitig gilt es, den Nutzwert der ausgewählten Objekte zu maximieren [MT90, S. 1 ff.].

75 Ein prototypisches Werkzeug für die Auswahl des favorisierten Änderungsbündels wurde im Rahmen dieser Arbeit in Microsoft Excel umgesetzt. Es sind jedoch auch kommerzielle Werkzeuge wie der ReleasePlanner™ der Firma EXPERT DECISIONS INC. erhältlich [Ruh10].

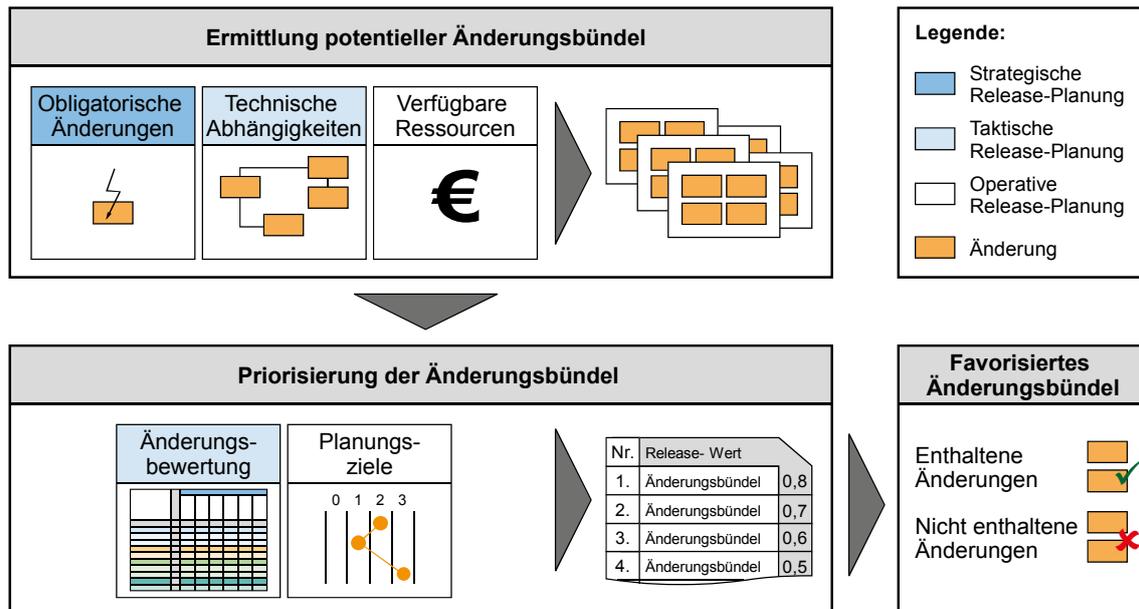


Bild 5-17: Vorgehen zur harmonisierten Bündelung von Änderungen (prinzipielle Darstellung)

Die **Ermittlung potentieller Änderungsbündel** ist eine kombinatorische Herausforderung, bei der mit jeder zusätzlichen Änderung die Anzahl⁷⁶ möglicher Änderungsbündel steigt. Eingeschränkt wird dies jedoch durch folgende Restriktionen:

- **Obligatorische Änderungen:** Teilweise sind bestimmte Produkt-Features bzw. Änderungen für ein Release zwingend zu berücksichtigen. Dies gilt speziell im Fall kundenrelevanter Releases, bei denen die Vorgaben z.B. aus der Produkt-Roadmap oder der Marketingstrategie resultieren. Vor diesem Hintergrund wurde im Anwendungsbeispiel in der strategischen Release-Planung eine neue Kommunikationsschnittstelle eingeplant (vgl. Abschnitt 5.1.3.1). Dies entspricht der Änderung Nr. 2 *Bluetooth-Schnittstelle*. Diese ist bei der Ermittlung potentieller Änderungsbündel zwingend zu berücksichtigen. Durch die Berücksichtigung obligatorischer Änderungen kann nicht zuletzt auch die in Abschnitt 5.2.3 betrachtete Produktwertsteigerung sichergestellt werden.
- **Technische Abhängigkeiten:** Zwischen zwei Änderungen kann sich technisch bedingt eine Rangfolge ergeben, in der Änderung „A“ vor Änderung „B“ umgesetzt werden muss (vgl. Abschnitt 4.5.3). Hieraus folgen drei mögliche Konsequenzen: Das Änderungsbündel enthält 1) beide Änderungen, 2) nur Änderung „A“ und 3) keine der beiden Änderungen. Im Anwendungsbeispiel ist dies z.B. für die Änderungen *Parametrierung über Bluetooth-Schnittstelle* und die *Überarbeitung des Bedieninterfaces* zu berücksichtigen (vgl. Abschnitt 5.2.1.3).

⁷⁶ Schon bei einer sehr kleinen Anzahl von zehn Änderungen sind – ohne Berücksichtigung von Restriktionen – 1.024 unterschiedliche Änderungsbündel denkbar.

- **Verfügbare Ressourcen:** Relevant sind sämtliche Kombinationen von Änderungen, die die verfügbaren Ressourcen nicht überschreiten und somit innerhalb des nächsten Release umsetzbar sind. Hat die Release-Planung zum Ziel, die verfügbaren Entwicklungsressourcen möglichst gleichmäßig auszulasten, sollten die notwendigen Ressourcen für das favorisierte Änderungsbündel möglichst nah am geplanten Grenzwert liegen.

In einem zweiten Schritt folgt die **Priorisierung der Änderungsbündel**. Hierzu wird auf die Änderungsbewertung aus der taktischen Release-Planung und das zuvor definierte Planungsziel zurückgegriffen. Es wird für jedes Änderungsbündel der Release-Wert (RW) bestimmt, der sich unter Berücksichtigung der drei Kriterien Wichtigkeit (W), Dringlichkeit (D) und Änderungssynergien (ÄS) ergibt. Der Wert für die Änderungssynergie ergibt sich aus der Anzahl der enthaltenen Änderungen (n) dividiert durch die Anzahl der betroffenen Systemelemente⁷⁷ (ASE) je Änderungsbündel. Berücksichtigt werden darüber hinaus die entsprechenden Gewichtungen (G), die durch das zuvor definierte Planungsziel vorgeben sind:

$$RW = \frac{\sum_{i=1}^n W_i}{n} * G_W + \frac{\sum_{i=1}^n D_i}{n} * G_D + \frac{n}{ASE} * G_{\text{ÄS}}$$

Gleichung 5-1: Ermittlung des Release-Werts je Änderungsbündel

Im Ergebnis liegt die in Tabelle 5-6 dargestellte Rangfolge **alternativer Änderungsbündel** vor. Die Tabelle enthält sämtliche – unter den gegebenen Restriktionen erlaubten – Änderungsbündel sortiert nach dem Release-Wert. Für jedes Änderungsbündel können der Tabelle folgende Informationen entnommen werden: Wie viele und welche Änderungen sind je Änderungsbündel enthalten bzw. nicht enthalten? Wie hoch sind die zur Umsetzung des Änderungsbündels erforderlichen Ressourcen? Wie viele Systemelemente sind von den enthaltenen Änderungen betroffen? Wie hoch sind Wichtigkeit, Dringlichkeit und Änderungsrelevanz des Änderungsbündels? Und letztlich: Wie hoch ist die mit dem Änderungsbündel verbundene Produktwertsteigerung?

Entsprechend der Rangfolge in der Tabelle sind die erstgelisteten Änderungsbündel die vielversprechendsten mit Bezug auf die Planungsziele. Allerdings sollte dabei nicht blind den Daten vertraut werden. Vielmehr stellt die Rangfolge im Sinne der Entscheidungsunterstützung eine Diskussionsgrundlage dar, auf deren Basis im Expertenteam Alternativen diskutiert und verglichen werden können. Im Anwendungsbeispiel zeigt sich, dass neben der als obligatorisch definierten *Bluetooth-Schnittstelle* insb. die Änderung *neue Softwarearchitektur* in allen hochpriorisierten Änderungsbündeln enthalten ist. Dies ist auf die Berücksichtigung der Änderungsabhängigkeiten zurückzuführen. So ist die

⁷⁷ Die Anzahl der betroffenen Systemelemente je Änderungsbündel wird auf Basis der in Abschnitt 5.2.1.2 analysierten Änderungsfortpflanzung ermittelt. Für jede Änderung wurden hier die betroffenen Systemelemente identifiziert.

Überarbeitung der Softwarearchitektur Voraussetzung für eine Vielzahl nachfolgender Änderungen. Die Änderungen *Endlagenerfassung* und *Regelung* sind hingegen in keinem der favorisierten Änderungsbündel enthalten. Dies ist im Wesentlichen auf die geringe Relevanz der Änderungen bei gleichzeitig hohem Umsetzungsaufwand zurückzuführen.

Tabelle 5-6: Bewertung und Priorisierung möglicher Änderungsbündel entlang des Release-Werts

	Änderungsbündel	Änderungen						Erforderliche Ressourcen	Betroffene Systemelemente	Anzahl Änderungen	Wichtigkeit	Dringlichkeit	Änderungsrelevanz	Produktwertsteigerung	Release-Wert
		Nr. 1 Sichtbare Zustandsanzeige	Nr. 2 Bluetooth-Schnittstelle	Nr. 3 Endlagenerfassung	Nr. 6 Neue Softwarearchitektur	Nr. 17 Regelung	Nr. 18 Überarbeitetes Bedieninterface								
1		X		X			X	22,5	3	7	9	10	4,7	7	1,83
2	X	X		X			X	24	3	7	9	9	4,5	9	1,81
3		X		X				20	3	7	9	7	4,2	7	1,76
4	X	X		X			X	23,5	3	6	8	11	4,5	9	1,75
5		X		X			X	24	3	6	9	8	4,3	7	1,72
6		X		X			X	21,5	3	6	8	10	4,3	7	1,72
7	X	X		X			X	23,5	3	6	8	9	4,2	10	1,69
8	X	X		X			X	23,5	3	6	8	9	4,2	9	1,69
9															

X Änderung im Änderungsbündel enthalten

Im Anwendungsbeispiel entfällt die Diskussion auf die ersten beiden Änderungsbündel, die sich mit Blick auf den Release-Wert nur geringfügig unterscheiden. Beide Änderungsbündel enthalten sieben Änderungen, die die Überarbeitung von drei betroffenen Systemelementen erfordern. Der Unterschied im Release-Wert ist auf eine leichte Differenz in der Änderungsrelevanz zurückzuführen. In der Diskussion entfällt die Entscheidung schließlich auf das zweite Änderungsbündel, da dies trotz etwas geringerer Änderungsrelevanz eine höhere Produktwertsteigerung aufweist und gleichzeitig die verfügbaren Ressourcen vollumfänglich ausnutzt.

5.3.3 Phase 3: Überprüfung von Umsetzungsalternativen

Mit der Priorisierung der Änderungsbündel liegt eine Empfehlung für die umzusetzenden Änderungen vor. Gleichzeitig stehen jedoch auch die Änderungen fest, die aufgrund begrenzter Ressourcen und technischer Abhängigkeiten nicht berücksichtigt werden können. Neben den bereits genannten Änderungen *Endlagenerfassung* und *Regelung* gilt dies für eine Vielzahl weiterer Änderungen. Die Überprüfung von Umsetzungsalternativen für diese Änderungen steht im Fokus von Phase 3. Hierzu wird in einem **ersten Schritt** für

jede nicht berücksichtigte Änderung das nächstmögliche Release ermittelt. Dies ist exemplarisch in Bild 5-18 dargestellt.

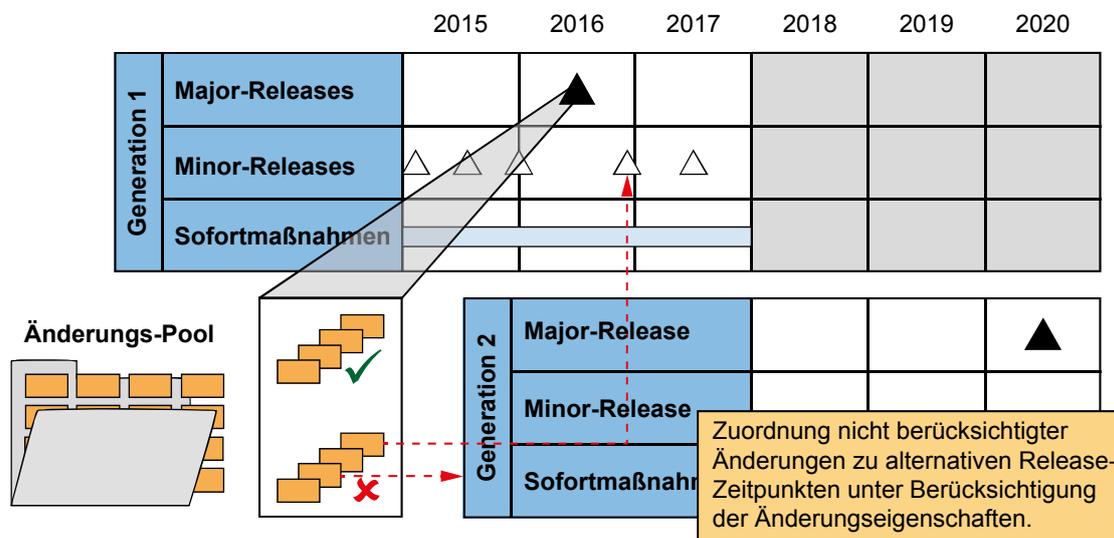


Bild 5-18: Überprüfung von Umsetzungsalternativen einzelner Änderungen

Die Überprüfung des Release-Plans ergibt zwei alternative Umsetzungstermine: Weniger komplexe Änderungen können auf das nachfolgende Minor-Release Anfang 2017 verschoben werden. Für alle anderen Änderungen ist der nächstmögliche Umsetzungszeitpunkt die Veröffentlichung der neuen Produktgeneration in 2018. Im Anwendungsbeispiel trifft dies für die Änderungen *Endlagenerfassung* und *Regelung* zu, die im favorisierten Änderungsbündel nicht enthalten sind. Beide Änderungen wurden im Rahmen der Änderungsklassifizierung dem Release-Typ Major-Release zugeordnet (vgl. Abschnitt 5.2.3) und können somit frühestens mit der neuen Produktgeneration in 2018 umgesetzt werden.

In einem **zweiten Schritt** ist nun unter Berücksichtigung aller Stakeholder zu überprüfen, ob eine Zurückstellung der betroffenen Änderungen akzeptabel ist. Im Anwendungsbeispiel wurde entschieden, die beiden Änderungen auf ein späteres Release zu verschieben. Ist die Änderungsumsetzung hingegen zwingend erforderlich, kann das zu einer vollständigen Neuplanung führen. In diesem Fall werden die betroffenen Änderungen als obligatorisch definiert und das Vorgehen zur harmonisierten Bündelung von Änderungen aus Phase 2 erneut durchlaufen. Bei der Auswahl des umzusetzenden Änderungsbündels handelt es sich daher um ein iteratives Vorgehen, das zur Entscheidungsunterstützung mehrfach durchlaufen werden kann.

5.3.4 Phase 4: Umsetzung des Änderungsbündels

Mit Phase 4 – der Umsetzung des Änderungsbündels – schließt die operative Release-Planung für das jeweilige Release ab. An der Schnittstelle zum Projektmanagement werden konkrete Maßnahmen zur produktiven Umsetzung des zuvor definierten Änderungsbündels festgelegt. Bei komplexen Änderungsvorhaben bietet sich dazu die Spezifikation

von Entwicklungsaufträgen⁷⁸ an, die den Rahmen für die anschließende Umsetzung der Änderungen mittels etablierter Vorgehensmodelle der Systementwicklung bilden (vgl. Abschnitt 2.5.2 und 2.5.4). Weniger komplexe Änderungen können hingegen ausgehend vom zuvor spezifizierten Änderungsauftrag direkt umgesetzt werden.

Neben der eigentlichen Umsetzung des Änderungsvorhabens gilt es darüber hinaus, die Weiterentwicklung des technischen Systems im Systemmodell nachzuhalten. Nur so kann die Konsistenz für die Planung zukünftiger Releases sichergestellt werden. Auch können frühere Entscheidungen so vereinfacht nachvollzogen werden. Für das im Anwendungsbeispiel fokussierte Major-Release zeigt dies exemplarisch Bild 5-19.

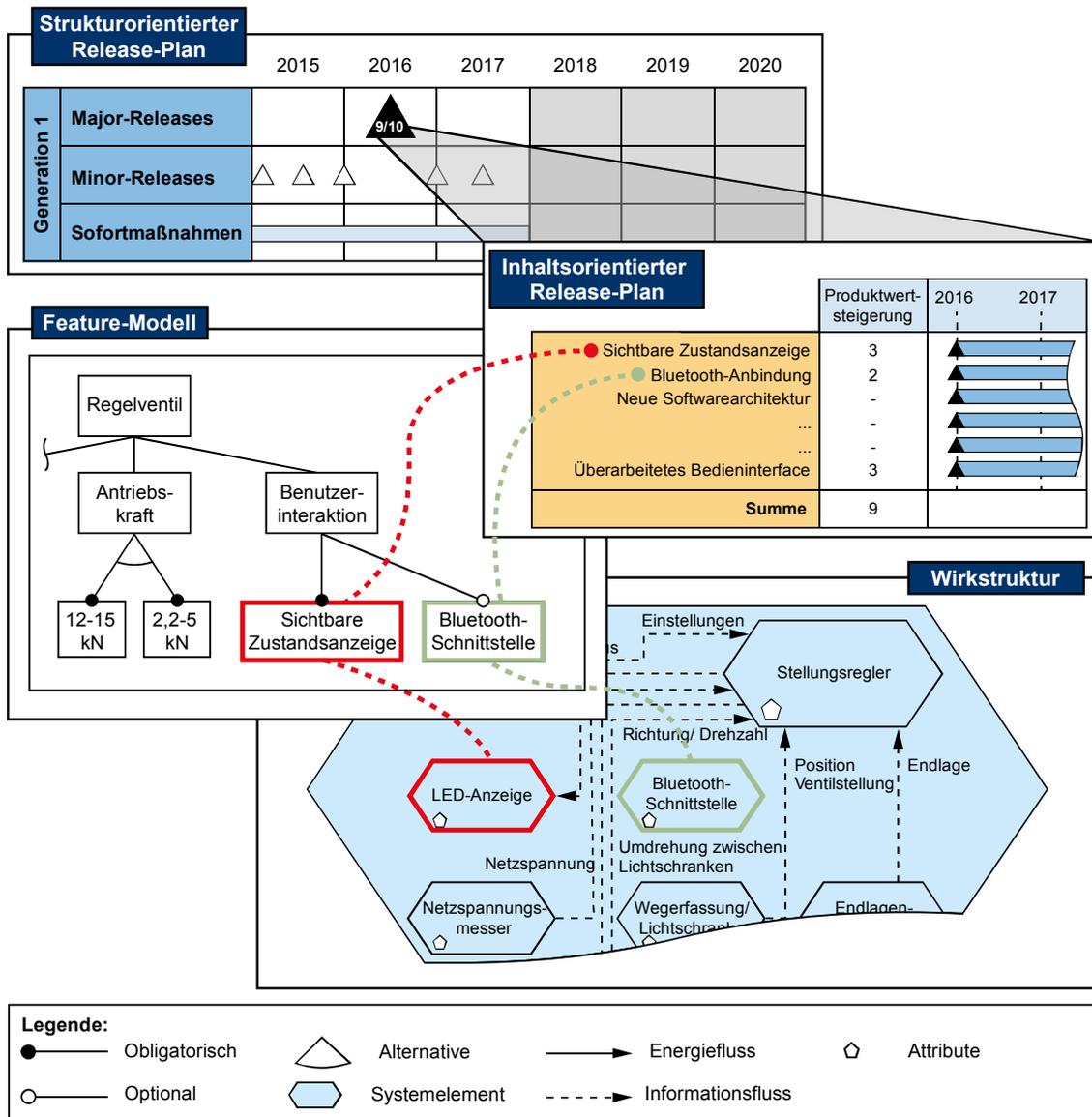


Bild 5-19: Beschreibung der Produktevolution mittels Release-Plan, Feature-Modell und Wirkstruktur als Resultat der Umsetzung des Änderungsbündels

78 Ein etabliertes Schema zur Spezifikation von Entwicklungsaufträgen liefert bspw. [Pei15, S. 148].

Die Produktevolution wird entsprechend Abschnitt 4.5.2 durch die Kopplung von Release-Plan, Feature-Modell und Wirkstruktur beschrieben. Der strukturorientierte Release-Plan verknüpft das umzusetzende Release mit einem konkreten Zeitpunkt für die Produktveröffentlichung – im Beispiel Mitte 2016. Die Konkretisierung des Releases erfolgt durch die Überführung in den inhaltsorientierten Release-Plan, der die umzusetzen den Änderungen näher spezifiziert. Die hier enthaltenen kundenrelevanten Änderungen werden wiederum als neue Produkt-Features in das Feature-Modell aufgenommen, das entsprechend des zugehörigen Release versioniert wird. Gleiches gilt für den zugehörigen Stand der Wirkstruktur, die die technische Umsetzung der Produkt-Features und damit der jeweiligen Änderungen beschreibt. Im Beispiel sind die Änderungen *Bluetooth-Schnittstelle* und *sichtbare Zustandsanzeige* hervorgehoben. Beide wurden zur Umsetzung im Major-Release Mitte 2016 eingeplant und werden somit als neue Produkt-Features im zugehörigen Feature-Modell spezifiziert. In der Wirkstruktur führt die Integration beider Features zu Änderungen an den jeweiligen Systemelementen. In der Release-Planung sind die hier beschriebenen Sichten letztlich für alle relevanten Änderungen zu erzeugen.

5.4 Kritische Bewertung der Systematik

In diesem Abschnitt findet abschließend eine Bewertung der erarbeiteten *Systematik zur Release-Planung intelligenter technischer Systeme* statt. Hierzu wird für jede Anforderung aus Abschnitt 2.7 überprüft, inwieweit sie durch einzelne Bestandteile bzw. das Zusammenwirken der Systematik erfüllt wird. Den Bezug der Anforderungen zur Systematik stellt Bild 5-20 her.

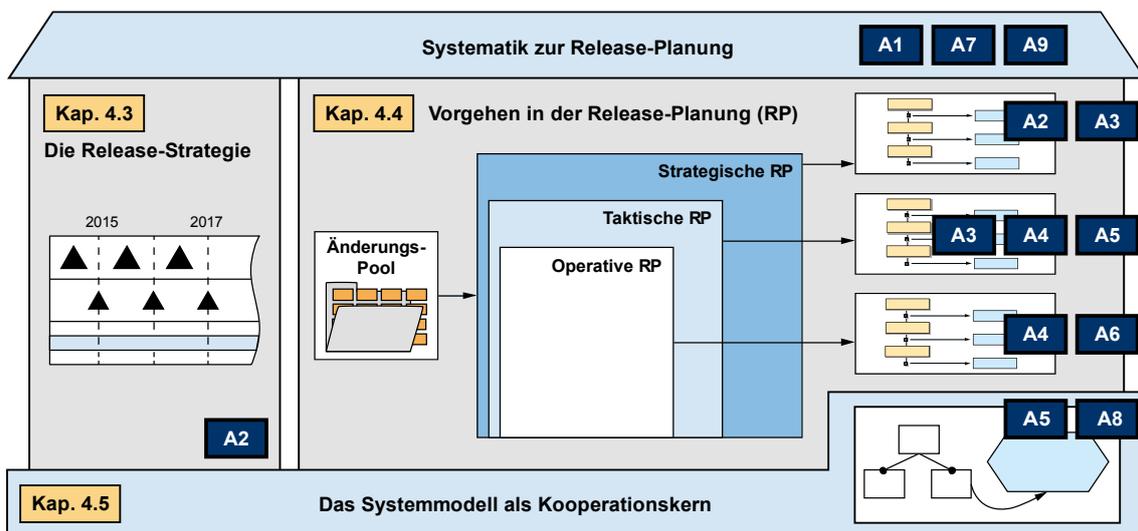


Bild 5-20: Erfüllung der Anforderungen durch die Systematik

A1) Durchgängiger Planungsprozess: Die Systematik beschreibt einen durchgängigen Planungsprozess – von der strategischen Definition kundenrelevanter Releases bis zur kurzfristigen, operativen Umsetzungsplanung unter Berücksichtigung verfügbarer Res-

sources. Grundlage hierfür ist die Gliederung des Planungsprozesses in die drei Ebenen strategische, taktische und operative Release-Planung (Abschnitt 4.1), deren Zusammenspiel über drei dedizierte Vorgehensmodelle konkretisiert wird (vgl. Abschnitt 4.4). Die Anwendbarkeit der Vorgehensmodelle wurde anhand des Beispiels elektrisches Regelventil belegt (Kapitel 5).

A2) Strukturierung des Release-Plans: Ausgehend von der Analyse der Gestaltungsfaktoren eines Release-Plans wurde ein Ordnungsschema erarbeitet, das die gezielte Ableitung einer Release-Strategie ermöglicht. Die Release-Strategie macht über ein charakteristisches Profil Vorgaben für die initiale Strukturierung des Release-Plans (Abschnitt 4.3). Im Zusammenspiel mit dem Vorgehen in der strategischen Release-Planung bietet die Systematik somit eine Entscheidungsunterstützung zur unternehmens- und produkt-spezifischen Strukturierung des Release-Plans (Abschnitt 5.1).

A3) Gesteuerte Produktwertsteigerung: Die Systematik steuert die Produktwertsteigerung durch das enge Zusammenspiel von strategischer und taktischer Release-Planung. Abgestimmt auf die übergeordnete Produktstrategie definiert die strategische Release-Planung die Markteinführungszeitpunkte für Major-Releases, die der Umsetzung kundenrelevanter Änderungen dienen (Abschnitt 5.1.3.1). Mittels des Schemas zur Änderungsbewertung werden diese Änderungen im Rahmen der taktischen Release-Planung identifiziert und den relevanten Releases gezielt zugeordnet (Abschnitt 5.2). Neben der Rate kundenrelevanter Releases kann somit der Beitrag jedes Release zur Produktwertsteigerung überprüft und gesteuert werden.

A4) Berücksichtigung von Stakeholder-Präferenzen: Den teils konkurrierenden Stakeholder-Präferenzen wird in der taktischen und operativen Release-Planung Rechnung getragen. So erfolgen sowohl Änderungsbewertung als auch die anschließende Umsetzungsplanung in einer Routinerunde, die sich aus einem interdisziplinären Team zusammensetzt (Abschnitt 5.2). Durch die Gewichtung des Planungsziels im Rahmen der operativen Release-Planung können bei der abschließenden Bündelung von Änderungen zu Releases ebenfalls verschiedene Interessen berücksichtigt werden. So erfolgt die Priorisierung der Änderungsbündel bspw. nach Wichtigkeit, Dringlichkeit und technischen Synergien (Abschnitt 5.3).

A5) Änderungsauswirkungen und -abhängigkeiten: Zur systematischen Berücksichtigung von Änderungsauswirkungen und -abhängigkeiten wurde ein umfassendes Vorgehen zur Änderungsanalyse im Kontext der Release-Planung erarbeitet (Abschnitt 5.2.1). Das Vorgehen wird methodisch durch den Einsatz des Systemmodells unterstützt (Abschnitt 4.5.3). Es mündet in einer Änderungsklassifikation, die im Rahmen der taktischen Release-Planung die gezielte Zuordnung von Änderungen zu Releases ermöglicht (Abschnitt 5.2). Bei der operativen Release-Planung werden darüber hinaus die identifizierten Änderungsabhängigkeiten als Restriktionen für die Ermittlung potentieller Änderungsbündel berücksichtigt (Abschnitt 5.3.2).

A6) Harmonisierte Bündelung von Änderungen zu Releases: Als Teil der operativen Release-Planung wurde ein Vorgehen zur Ermittlung potentieller Änderungsbündel erarbeitet (Abschnitt 5.3.2). Die harmonische Bündelung der Änderungen wird durch die Berücksichtigung der notwendigen Restriktionen (z.B. begrenzte Ressourcen, technische Abhängigkeiten) und des Planungsziels sichergestellt. Die mit dem Planungsziel vorgenommene Gewichtung ermöglicht es darüber hinaus, die abschließende Auswahl des umzusetzenden Änderungsbündels auf strategische Überlegungen abzustimmen (Abschnitt 5.3.1).

A7) Interdisziplinarität: Die Release-Planung intelligenter technischer Systeme ist eine interdisziplinäre Herausforderung, die markt- und entwicklungsstrategische Überlegungen gleichermaßen berücksichtigen muss. Die Systematik trägt dieser Herausforderung Rechnung, indem entlang des gesamten Planungsprozesses beide Dimensionen einbezogen werden. Besonders deutlich wird dies durch die Änderungsklassifikation, die nach den beiden Dimensionen Marktrelevanz und technische Auswirkung einer Änderung erfolgt (Abschnitt 5.2.3). Auf diese Weise kann sichergestellt werden, dass die Kompetenzen aus Produktplanung und Systementwicklung gleichermaßen berücksichtigt werden. Durch operative Restriktionen (z.B. verfügbare Ressourcen) bei der Ermittlung potentieller Änderungsbündel wird darüber hinaus der Perspektive des Projektmanagements Rechnung getragen (Abschnitt 5.3).

A8) Unterstützung durch modellbasierte Systembeschreibung: Als Kooperationskern in der Release-Planung stützt sich die Systematik auf ein Systemmodell, das auf der Spezifikationstechnik CONSENS basiert (Abschnitt 4.5). Zur Vereinigung der kundenorientierten und technischen Sicht sowie zur Beschreibung der zeitlichen Varianz wurde CONSENS um Feature-Modell und Release-Plan erweitert. Im Release-Planungsprozess unterstützt dies die zielgruppengerechte Kommunikation zwischen Produktplanung, Systementwicklung und Projektmanagement.

A9) Systematisches Vorgehen und Praktikabilität: Ein systematisches und reproduzierbares Vorgehen bei der Release-Planung wird durch die beschriebenen Vorgehensmodelle sichergestellt. Sie beschreiben notwendige Aufgaben und verknüpfen diese mit dedizierten Hilfsmitteln, die größtenteils speziell für die Systematik erarbeitet wurden. Die prägnante Darstellung der Hilfsmittel (z.B. durch Portfolios und Bewertungstabellen) trägt darüber hinaus zur Praktikabilität der Systematik bei. Vorgehen und Hilfsmittel bestätigen sich im Anwendungsbeispiel als sinnvoll und angemessen.

Die entwickelte *Systematik zur Release-Planung intelligenter technischer Systeme* erfüllt damit die gestellten Anforderungen in vollem Umfang. Die Systematik unterstützt Fachleute aus Produktplanung, Systementwicklung und Projektmanagement bei der Umsetzungsplanung neuer Produkt-Features und technischer Änderungen in Form von Releases. Anhand des elektrischen Regelventils wurde die erfolgreiche Anwendung der Systematik nachgewiesen.

6 Zusammenfassung und Ausblick

Der Wandel von mechatronischen zu intelligenten technischen Systemen eröffnet faszinierende Perspektiven: adaptive, robuste, vorausschauende und benutzungsfreundliche Systeme, die einen erheblich gesteigerten Funktionsumfang und damit vielfältige Innovationspotentiale versprechen. Unternehmen sind so in der Lage, die Attraktivität ihrer Produkte kontinuierlich durch die Einführung innovativer Produkt-Features zu steigern. Mit dem Wandel zu intelligenten technischen Systemen steigt jedoch auch die Komplexität in der Systementwicklung. Dies ist im Wesentlichen auf den zunehmenden Anteil von Software und Elektronik zurückzuführen, der einerseits zu einer erhöhten Interdisziplinarität in der Entwicklung führt und andererseits weit häufiger aktualisiert werden muss als die Mechanik eines Systems. Resultat sind komplexe, technisch bedingte Änderungen, die zu einer kontinuierlichen Weiterentwicklung des technischen Systems führen. Diese Weiterentwicklung drückt sich in mannigfachen Produktversionen, Facelifts und neuen Produktgenerationen aus, deren Veröffentlichung durch abgestimmte Innovations- und Anpassungsschritte gezielt gesteuert werden muss.

Ein **Lösungsansatz** ist die systematische Weiterentwicklung technischer Systeme in Form von Release-Projekten. In einem Release werden Änderungen an einem Produkt gebündelt und anschließend gemeinsam entwickelt, getestet und freigegeben. **Nutzenpotentiale** der Weiterentwicklung in Releases sind eine gesteuerte Produktwertsteigerung und die Rationalisierung des Änderungsprozesses durch die Nutzung von Änderungssynergien. Die Erschließung dieser Potentiale setzt jedoch eine systematische Release-Planung voraus, die die folgenden **Herausforderungen** meistern muss:

- Eine auf markt- und wettbewerbsstrategische Überlegungen abgestimmte Weiterentwicklung des Systems.
- Die Berücksichtigung unterschiedlicher und teils konkurrierender Stakeholder-Präferenzen.
- Den Umgang mit einer zunehmenden Komplexität technischer Änderungen einschließlich der Berücksichtigung von Änderungsabhängigkeiten.
- Die Schaffung eines gemeinsamen Kooperationskerns für die interdisziplinäre Zusammenarbeit von Produktplanung, Systementwicklung und Projektmanagement.

Um diesen Herausforderungen zu begegnen, muss die Umsetzungsplanung von Änderungen in Form von Releases methodisch unterstützt werden. Es werden ein systematisches Vorgehen und dedizierte Hilfsmittel zur Release-Planung intelligenter technischer Systeme benötigt. Ein besonderes Augenmerk muss hierbei auf der Berücksichtigung von Änderungsauswirkungen und -abhängigkeiten liegen. Dies erfordert neben der Strukturierung der Planungsaktivitäten insb. geeignete Hilfsmittel zur interdisziplinären System- und Änderungsanalyse.

Vor diesem Hintergrund wurde in Kapitel 3 der **Stand der Technik** zur methodischen Unterstützung der Release-Planung untersucht. Hierbei wurden dedizierte Hilfsmittel für einzelne Teilaufgaben genauso berücksichtigt wie bestehende Systematiken zur Release-Planung. Die Analyse des Stands der Technik liefert partiell Ansätze, die genutzt und erweitert werden können. Eine durchgängige Systematik, die alle Anforderungen abdeckt, existiert jedoch nicht.

Als dedizierte Hilfsmittel wurden Ansätze zur Spezifikation technischer Systeme sowie zur Analyse, Klassifizierung und Priorisierung von technischen Änderungen untersucht. Zur Spezifikation eines technischen Systems kristallisiert sich die Spezifikationstechnik CONSENS als geeignetes Hilfsmittel heraus. Sie erfüllt wichtige Anforderungen insb. im Hinblick auf eine interdisziplinäre Kommunikation und Kooperation im Rahmen der Release-Planung. Die untersuchten Ansätze zur technischen Änderungsanalyse sowie zur Priorisierung und Klassifizierung von Änderungen werden in der zu entwickelnden Systematik partiell aufgegriffen, angepasst und erweitert. Als besonders hilfreich stellt sich das von ABMANN entwickelte Klassifikationsschema heraus, das der Zuordnung von Änderungen zu standardisierten Bearbeitungsprozessen dient. Die standardisierte Abwicklung von Änderungen mittels eines Klassifikationsschemas wird als Hilfsmittel in der taktischen Release-Planung aufgegriffen.

Über die dedizierten Hilfsmittel hinaus wurden ganzheitliche Systematiken zur strategischen Produktplanung und zur Release-Planung von technischen Systemen untersucht. Besonders hervorzuheben ist hier der Ansatz zur strategischen Planung nach NIPPA und LABRIOLA. Dieser kaskadiert den Innovationsprozess in eine lang-, mittel- und kurzfristige Innovationsplanung und bietet so einen Anhaltspunkt für die Strukturierung des Release-Planungsprozesses. Die harmonisierte Bündelung von Änderungen zu Releases wird zwar in den Ansätzen nach ZORN-PAULI ET AL. und BELENER aufgegriffen, die untersuchten Systematiken erfüllen die definierten Anforderungen letztlich aber nur partiell. Entscheidende Defizite sind die unzureichende Verzahnung der verschiedenen Planungsebenen sowie die nur sporadische Berücksichtigung von Änderungsauswirkungen und -abhängigkeiten. Aus diesen Gründen bestand **Handlungsbedarf** zur Entwicklung einer durchgängigen Systematik, die die Release-Planung intelligenter technischer Systeme unter Berücksichtigung von Änderungsauswirkungen und -abhängigkeiten unterstützt.

Die letztlich in Kapitel 4 beschriebene Systematik greift vereinzelt Aspekte des untersuchten Stands der Technik auf, erweitert bzw. ergänzt diese um neu entwickelte Hilfsmittel und verknüpft alles zu einem durchgängigen Ansatz. Grundidee ist die Strukturierung des Planungsprozesses in eine strategische, taktische und operative Release-Planung. Kernaufgabe der strategischen Release-Planung ist die initiale Strukturierung des Release-Plans und die damit verbundene Festlegung der Release-Typen und Release-Zeitpunkte. Die Zuordnung von neuen Produkt-Features und Änderungen zu diesen Releases erfolgt in einem kontinuierlichen Prozess – der taktischen Release-Planung. Im Rahmen der operativen Release-Planung wird schließlich unter Berücksichtigung der verfügbaren Ressourcen der tatsächliche Release-Inhalt festgelegt. Zur Umsetzung dieses

Planungsprozesses liegt im **Ergebnis** eine Systematik vor, die im Kern drei übergeordnete Bestandteile umfasst:

- Ein **Ordnungsschema** für alternative Release-Strategien sortiert die Gestaltungsfaktoren und möglichen Ausprägungen eines Release-Plans. Ergänzt um ein dediziertes Vorgehen unterstützt es die unternehmens- und produktspezifische Strukturierung eines initialen Release-Plans im Rahmen der strategischen Release-Planung.
- Drei **Vorgehensmodelle** unterteilen die durchzuführenden Tätigkeiten der strategischen, taktischen und operativen Release-Planung in aufeinander aufbauende Phasen mit zugehörigen Meilensteinen. Die Vorgehensmodelle integrieren dedizierte Hilfsmittel. Dazu zählen ein Vorgehen zur technischen Änderungsanalyse, verschiedene Schemata zur Bewertung und Klassifizierung von Änderungen und ein Ansatz zur harmonisierten Bündelung von Änderungen.
- Eine Technik zur **Systembeschreibung** dient als Kooperationskern in der Release-Planung. Basis ist die methodische und sprachliche Erweiterung der Spezifikationstechnik CONSENS für die technische Änderungsanalyse und zur Beschreibung der zeitlichen Varianz eines technischen Systems.

Die **Anwendung** der Systematik in Kapitel 5 erfolgte anhand der Release-Planung eines elektrischen Regelventils. Hierzu wurden die Vorgehensmodelle der Systematik vollständig durchlaufen, die neu entwickelten Hilfsmittel angewendet und die so erzielten Resultate vorgestellt. Das gewählte Anwendungsbeispiel belegt die Praxistauglichkeit und Industrierelevanz der Arbeit. Zusammenfassend kann festgehalten werden, dass die erarbeitete *Systematik zur Release-Planung intelligenter technischer Systeme* die aufgestellten Anforderungen in gefordertem Umfang erfüllt.

Mit Blick auf die Release-Planung intelligenter technischer Systeme besteht **weiterer Forschungsbedarf**. Die Anzahl zeitlicher Varianten eines technischen Systems potenziert sich mit der Anzahl der räumlichen Varianten. Vor diesem Hintergrund sollten sich zukünftige Forschungsarbeiten der Gestaltung einer wirtschaftlichen Varianz ganzer Produktportfolios widmen. Dies erfordert neben einer geeigneten Systematik insb. die Weiterentwicklung der Spezifikationstechnik CONSENS. Ziel muss eine differenzierte Beschreibung von räumlichen und zeitlichen Varianten im Systemmodell sein, was neben sprachlichen und methodischen Anpassungen insb. die Realisierung einer geeigneten Werkzeugunterstützung umfasst.

Die Release-Planung zeichnet sich zudem als geeigneter Ansatz zur Synchronisation der verschiedenen Innovationszyklen von Mechanik, Elektronik und Software ab. Durch eine modulatorientierte Release-Planung können dynamische, sich häufig verändernde Systemanteile von einem eher statischen Grundanteil entkoppelt werden. In der Automobilindustrie wird so bspw. der Herausforderung verkürzter Innovationszyklen durch Consu-

mer-Electronics entgegengewirkt. Forschungsbedarf besteht hier insb. bei der Strukturierung des Systems in geeignete Module. Erste Ansätze zur Gestaltung entsprechender Systemarchitekturen lassen sich bei SCHUH ET AL. und MAURER ET AL. erkennen (vgl. Abschnitt 3.5.2 und 3.5.3).

Ferner zeichnet sich ein Trend zur Produktwertsteigerung im Betrieb ab. In diesem Fall werden neue Produktfunktionalitäten technisch vorbereitet und erst durch ein späteres Software-Update im Betrieb des Systems aktiviert. Auch hier ist die Architekturgestaltung ein wesentlicher Erfolgsfaktor. Forschungsbedarf besteht insb. bei der integrativen Gestaltung von System- und Softwarearchitektur. Zudem nimmt hierdurch die Bedeutung der IT-Sicherheit zu.

Das HEINZ NIXDORF INSTITUT und das FRAUNHOFER-INSTITUT FÜR ENTWURFSTECHNIK MECHATRONIK IEM greifen diese Themen bereits heute in weiterführenden Forschungsaktivitäten auf. Gemeinsames und übergeordnetes Ziel ist letztlich eine neue Schule des Entwurfs intelligenter technischer Systeme. Die in dieser Arbeit entwickelte Systematik liefert einen Baustein auf dem Weg zu diesem Ziel.

7 Abkürzungsverzeichnis

BMBF	Bundesministerium für Bildung und Forschung
bspw.	beispielsweise
bzgl.	bezüglich
bzw.	beziehungsweise
insb.	insbesondere
CAD	Computer Aided Design
CONSENS	CONceptual design Specification technique for the ENgineering of complex Systems
CPS	Cyber-physisches System / Cyber-physical system
DfV	Design for Variety
DIN	Deutsches Institut für Normung
DMM	Domain Mapping Matrix
DoD	Department of Defense
DSM	Design Structure Matrix
engl.	Englisch
et al.	lateinisch et alii = und andere
INCOSE	International Council on Systems Engineering
IT	Informationstechnik
it's OWL	Intelligente Technische Systeme OstWestfalenLippe
MBSE	Model-Based Systems Engineering
MDM	Multi Domain Matrix
lat.	Lateinisch
NASA	National Aeronautics and Space Administration
OMG	Object Management Group
PAM	Plant-Asset-Management
PC	Personal Computer

PEP	Produktentstehungsprozess
PLM	Produktlebenszyklusmanagement (engl. Product-Lifecycle-Management)
RE	Requirements Engineering
RoI	Return on Investment
SE	Systems Engineering
SoS	System of Systems
u.a.	unter anderem
VDI	Verein Deutscher Ingenieure
vgl.	vergleiche
z.B.	zum Beispiel

8 Literaturverzeichnis

Publikationen

- [AA13] ABRAMOVICI, M.; AIDI, Y.: Next Generation Product Lifecycle Management (PLM). In: FATHI, M. (EDS.): Integration of Practice-Oriented Knowledge Technology: Trends and Perspectives. Springer Verlag, Berlin, Heidelberg, 2013, S. 143-156
- [AA15] ABRAMOVICI, M.; AIDI, Y.: A Knowledge-based Assistant for Real-time Planning and Execution of PSS Engineering Change Processes. Proceedings of the 7th CIRP Conference on Industrial Product-Service Systems, IPS² 2014, Saint-Étienne, Frankreich, 2014, S. 445-450
- [ABG10] ABRAMOVICI, M.; BELLALOUNA, F.; GÖBEL, J.: Adaptive Change Management for Industrial Product-Service Systems. In: Journal of Mechanical Engineering, Nr. 56, 2010, S. 696-706.
- [ABK+13] APEL, S.; BATORY, D.; KÄSTNER, C.; SAAKE, G.: Feature-Oriented Software Product Lines - Concepts and Implementation. Springer Verlag, Berlin, Heidelberg, 2013
- [ABW15] ALBERS, A.; BURSAC, N.; WINTERGERST, E.: Produktgenerationsentwicklung – Bedeutung und Herausforderungen aus einer entwicklungsmethodischen Perspektive. In: Binz, H.; Bertsche, B.; Bauer, W.; Roth, D. (Hrsg.): Stuttgarter Symposium für Produktentwicklung 2015. Stuttgart, 2015
- [aca11] ACATECH – DEUTSCHE AKADEMIE DER TECHNIKWISSENSCHAFTEN (Hrsg.): Cyber-Physical Systems. Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion (acatech POSITION). Springer Verlag, Berlin, Heidelberg, 2011
- [aca14] ACATECH – DEUTSCHE AKADEMIE DER TECHNIKWISSENSCHAFTEN (Hrsg.): Smart Service Welt – Umsetzungsempfehlungen für das Zukunftsprojekt Internetbasierte Dienste für die Wirtschaft. März 2014
- [ADG+14] ANACKER, H.; DUMITRESCU, R.; GAUSEMEIER, J.; IWANEK, P.; SCHIERBAUM, T.: Methodology for the identification of self-optimizing potentials for mechatronic systems. Proceedings of 2nd Joint Symposium on System-integrated Intelligence: New Challenges for Product and Production Engineering (SysInt), July 2-4, Bremen, Deutschland, 2014, S. 17-26
- [AG12] ALBERS, A.; GAUSEMEIER, J.: Von der fachdisziplinorientierten Produktentwicklung zur vorausschauenden und systemorientierten Produktentstehung. In: Anderl, R.; Eigner, M.; Sandler, U.; Stark, R. (Hrsg.): Smart Engineering – interdisziplinäre Produktentstehung. acatech Diskussion, April 2012
- [AGN15] ABRAMOVICI, M., GÖBEL, J.C., NEGES, M.: Smart Engineering as Enabler for the 4th Industrial Revolution. In: Fathi, M. (Eds.): Integrated Systems: Innovations and Applications. Springer Verlag, Berlin, 2015, S. 163-170
- [Alt12] ALT, O.: Modellbasierte Systementwicklung mit SysML. Carl Hanser, München, 2012
- [Arm05] ARMSTRONG, J.R.: A Systems Approach to Process Infrastructure. INCOSE Symposium, Virginia, USA, 2005
- [Ass00] ABMANN, G.: Gestaltung von Änderungsprozessen in der Produktentwicklung. Dissertation, Technische Universität München, Herbert Utz Verlag – Wissenschaft, München, 2000
- [Aum13] AUMAYR, K.: Erfolgreiches Produktmanagement – Tool-Box für das professionelle Produktmanagement und Produktmarketing. Springer Gabler Verlag, Wiesbaden, 3. Auflage, 2013
- [AWC12] AHMAD, N.; WYNN, D.C.; CLARKSON, P.J.: Change impact on a product and its redesign process: a tool for knowledge capture and reuse. Research in Engineering Design, 2012, S. 1-26

- [BA96] BOHNER, S.A.; ARNOLD, R.S.: Software change impact analysis. IEEE Computer Society Press, Los Alamitos, USA, 1996
- [BA05] BERANDER, P.; ANDREWS, A.: Requirements Prioritization. In: Aurum, A.; Wohlin, C. (Eds.): Engineering and Managing Software Requirements. Springer Verlag, Berlin, Heidelberg, 2005, S. 69-94
- [BB08] BORCHES, P.D.; BONNEMA, G.M.: On the origin of evolvable systems. Proceedings of the TMCE 2008, April 21-25, Kusadasi, Türkei, 2008
- [BC00] BALDWIN, C.; CLARK, K.: Design rules: The power of modularity. MIT Press, Cambridge, 2000
- [BDP+07] BRAUN S. C; DIEHL, H.; PETERMANN, M.; HELLENBRAND, D.; LINDEMANN, U.: Function Driven Process Design for the Development of Mechatronic Systems. Proceedings of the 9th International DSM Conference, October 16-18 2007, Shaker Verlag, München, 2007
- [Bei13] BEIER, G.: Verwendung von Traceability-Modellen zur Unterstützung der Entwicklung technischer Systeme. Dissertation, Fakultät V - Verkehrs- und Maschinensysteme, Technische Universität Berlin, Berlin, 2013
- [Bel08] BELENER, P.M.: Technisches Änderungsmanagement modularer Produkte und Prozesse. Shaker Verlag, Aachen, 2008
- [Ben05] BENDER, K.: Embedded Systems – Qualitätsorientierte Entwicklung. Springer Verlag, Berlin, 2005
- [Ber49] BERTALANFFY, L.: Zu einer allgemeinen Systemlehre. Biologia Generalis 19(1), 1949, S. 114-129
- [Ber69] BERTALANFFY, L.: General System Theory – Foundations, Development, Applications. George Braziller, New York, 1969
- [BFV00] BIANCHI, A.; FASOLINO, A.; VISAGGIO, G.: An exploratory case study of the maintenance effectiveness of traceability models. Proceedings of the 8th International Workshop on Program Comprehension, June 10-11 2000, Limerick, Ireland, IEEE Computer Society, Washington, USA, 2000, S. 149-158
- [BG04] BENDRICH, D.; GLOGER, E.: Neues Änderungsmanagement Serie – weniger ist oft mehr. Daimler intern, Global Supplier, 4/2004
- [BH01] BEA, F. X.; HAAS, J.: Strategisches Management. Lucius & Lucius Verlag, Stuttgart, 3. Auflage, 2001
- [BK13] BAHNS, T.; KRAUSE, D.: Pflege modularer Produktfamilien nach dem Markteintritt durch die Produktentwicklung. In: Krause, D.; Paetzold, K.; Wartzack, S. (Hrsg.): Design for X – Beiträge zum 24. DfX-Symposium. TuTech Verlag, Hamburg, 2013
- [Ble11] BLEES, C.: Eine Methode zur Entwicklung modularer Produktfamilien. Dissertation, Technische Universität Hamburg-Harburg, TuTech Verlag Hamburg, Hamburger Schriftenreihe Produktentwicklung und Konstruktionstechnik, Band 3, Hamburg, 2011
- [Boe88] BOEHM, B.W.: A Spiral Model of Software Development and Enhancement. In: IEEE Computer, 21/1988, S. 61-72
- [Bor10] BORCHES, P.D.: A3 Architecture Overview – A tool for effective communication in product evolution. Dissertation, Department of Engineering Technology (CTW), Universität Twente, Enschede, Netherlands, 2010
- [Bra14] BRANDIS, R.: Systematik für die integrative Konzipierung der Montage auf Basis der Prinzipienlösung mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 325, Paderborn, 2014

- [Bri10] BRINK, V.: Verfahren zur Entwicklung konsistenter Produkt- und Technologiestrategien. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriften-reihe, Band 280, Paderborn, 2010
- [BPD+10] BOTTERWECK, G.; PLEUS, A.; DEEPAK, D.; POLZER, A.; KOWALEWSKI, S.: EvoFM: Feature-driven Planning of Product-line Evolution. Proceedings of the 2010 ICSE Workshop on Product Line Approaches in Software Engineering, May 2 2010, Cape Town, South Africa, 2010, S. 24-31
- [Bro10] BROY, M. (Hrsg.): Cyber-Physical Systems: Innovation durch softwareintensive eingebettete Systeme. Springer Verlag, Berlin, Heidelberg, 2010
- [BSM+14] BAUER, W.; SCHLUND, S.; MARRENBACH, D. GANSCHAR, O.: Industrie 4.0 – Volkswirtschaftliches Potential für Deutschland. Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V., Berlin, 2014
- [BT07] BESSANT, J., TIDD, J.: Innovation and Entrepreneurship. John Wiley and Sons, Chichester, 2007
- [Bul94] BULLINGER, H.-J.: Einführung in das Technologiemanagement – Modelle, Methoden, Praxisbeispiele. B.G. Teubner Verlag, Stuttgart, 1994
- [BV08] BANKHOFER, U.; VOGEL, J.: Datenanalyse und Statistik – Eine Einführung für Ökonomen im Bachelor. Gabler, Wiesbaden, 2008
- [CE00] CZARNECKI, K.; EISENECKER, U.: Generative programming: Methods, tools, and applications. ACM Press/Addison-Wesley, New York, 2000
- [Coc08] COCKBURN, A.: Using both incremental and iterative development. CrossTalk – The Journal of Defense Software Engineering, 2008, S. 27-30
- [Con97] CONRAT NIEMBERG, J.-I.: Änderungskosten in der Produktentwicklung. Dissertation, Technische Universität München, München, 1997
- [Cro89] CROSS, N.: Engineering Design Methods. Wiley, Chichester, 1989
- [CSE01] CLARKSON, P.J.; SIMONS, C.; ECKERT, C.M.: Predicting change propagation in complex design. Proceedings of ASME design engineering technical conferences, September 9-12 2001, Pittsburgh, USA, 2001
- [CWC09] CHALUPNIK, M. J.; WYNN, D. C.; CLARKSON, P. J.: Approaches to mitigate the impact of uncertainty in development processes. Proceedings of the 17th International Conference on Engineering Design, August 24-27 2009, Palo Alto, USA, 2009, S. 459-570
- [CWE+04] CRAWLEY, E.; DE WECK, O.; EPPINGER, S.; MAGEE, C.; MOSES, J.; SEERING, W.; SCHINDALL, J.; WALLACE, D.; WHITNEY, D.: The influence of architecture in engineering systems. Engineering systems monograph, MIT Press, 2004
- [CZZ05] CHEN, K; ZHANG, W.; ZHAO, H.; MEI, H.: An approach to constructing feature models based on requirements clustering. Proceeding International Conference on Requirements Engineering (RE), August 29 September 2 2005, IEEE Computer Society, 2005, S. 31-40
- [DDG+14] DOROCIAC, R.; DUMITRESCU, R.; GAUSEMEIER, J.; IWANEK, P.: Specification Technique CONSENS for the Description of Self-optimizing Systems. In: Gausemeier, J.; Rammig, F.; Schäfer, W. (Eds.): Design Methodology for Intelligent Technical Systems, Springer Verlag, Berlin, 2014
- [DH02] DAENZER, W. F.; HUBER, F. (Hrsg.): Systems Engineering – Methodik und Praxis. Verlag Industrielle Organisation, Zürich, 11. Auflage, 2002
- [DKL06] DEUBZER, F.; KREIMEYER, M.; LINDEMANN, U.: Exploring Strategies in Change Management – Current Status and Activity Benchmark. International Design Conference, Dubrovnik, Kroatien, 2006, S. 815-822

- [DNL96] DESCHAMPS, J-P.; NAYAK, P.R.; LITTLE, D.A.: Produktführerschaft – Wachstum und Gewinn durch offensive Produktstrategien. Campus Verlag, Frankfurt/Main, 1996
- [Dre75] DREIBHOLZ, D.: Ordnungsschemata bei der Suche von Lösungen. *Konstruktion*, 27/1975, S. 233-240
- [DSM15-ol] DSM CONFERENCE HOMEPAGE. Unter: <http://www.dsm-conference.org/>, 4. Dezember 2015
- [Dud14-ol] DUDEN ONLINE: Systematik. Unter: <http://www.duden.de/rechtschreibung/Systematik>, 4. Dezember 2015
- [Dud15-ol] DUDEN ONLINE: Release. Unter: http://www.duden.de/rechtschreibung/Release_Version_Software, 4. Dezember 2015
- [Dum10] DUMITRESCU, R.: Entwicklungssystematik zur Integration kognitiver Funktionen in fortschrittliche mechatronische Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 286, Paderborn, 2010
- [EB08] ENGEL, A.; BROWNING, T.R.: Designing systems for adaptability by means of architecture options. *Systems Engineering*, 11(2), 2008, S. 125-146
- [EB12] EPPINGER, S.D.; BROWNING, T.R.: Design Structure Matrix Methods and Applications. The MIT Press, Cambridge Massachusetts, London, England, 2012
- [Ebe06] EBERT, C.: The impacts of software product management. *The Journal of Systems and Software*, 80/2007, S. 850-861
- [Ebe10] EBERT, C.: Systematisches Requirements Engineering – Anforderungen ermitteln, spezifizieren, analysieren und verwalten. dpunkt.verlag, Heidelberg, 3. Auflage, 2010
- [ECE05] ECKERT, C.M.; CLARKSON, P.J.; EARL, C.F.: Predictability of change in engineering: a complexity view. Proceedings of the 2005 ASME international design engineering technical conferences, September 24-28 2005, Long Beach, California, USA, 2005, S. 341-350
- [ECZ04] ECKERT, C. M.; CLARKSON, P. J.; ZANKER, W.: Change and customisation in complex engineering domains. *Research in Engineering Design*, 15(1), 2004, S. 1-21
- [EE05] EARL, C. J.; ECKERT, C.M.: Complexity in Design process improvement – a review of current practice, Springer Verlag, London, 2005, S. 174-197
- [Ehr09] EHRENSPIEL, K.: Integrierte Produktentwicklung – Denkabläufe, Methodeneinsatz, Zusammenarbeit. Carl Hanser Verlag, München, 4. Auflage, 2009
- [Epp91] EPPINGER, S.D.: Model-based Approaches to Managing Concurrent Engineering. *Journal of Engineering Design*, 2/1991, S. 283-290
- [Eri98] ERIXON, G.: Modular Function Deployment – A Method for Product Modularisation. Dissertation, Royal Institute of Technology, KTH, Stockholm, 1998
- [Est08] ESTEFAN, J. A.: Survey of Model-Based Systems Engineering (MBSE) Methodologies. IN-COSE MBSE Focus Group, 2008
- [EWS97] EVERSHEIM, W.; WARNKE, L.; SCHRÖDER, T.: Änderungsmanagement in Entwicklungsoperationen. *VDI-Z* 139/1997, S. 60-63
- [FA13] FORSCHUNGSUNION WIRTSCHAFT-WISSENSCHAFT; ACATECH – DEUTSCHE AKADEMIE DER TECHNIKWISSENSCHAFTEN (HRSG.): Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0, 2013
- [Fai09] FAIRLEY, R.E.: Managing and leading Software Projects. Wiley, New Jersey, USA, 2009
- [FHS02] FERBER, F.; HAAG, J.; SAVOLAINEN, J.: Feature Interaction and Dependencies – Modeling Features for Reengineering a Legacy Product Line. Springer-Verlag, Berlin-Heidelberg, 2002

- [Fix06] FIXSON, S. K.: Modularity and Commonality Research: Past Developments and Future Opportunities. Sloan School of Management, Massachusetts Institute of Technology, Boston, 2006
- [FG13] FELDHUSEN, J.; GROTE, K.-H.: PAHL/BEITZ (HRSG.): Konstruktionslehre – Methoden und Anwendung erfolgreicher Produktentwicklung. Springer Verlag, Berlin, 8. Auflage, 2013
- [FGN+00] FRICKE, E.; GEBHARD, B.; NEGELE, H.; IGENBERGS, E.: Coping with changes: causes, findings and strategies. Systems Engineering, 3(4), 2000, S. 169-179
- [FMC05] FORSBERG, K.; MOOZ, H.; COTTERMAN, H.: Visualizing Project Management: Models and Frameworks for Mastering Complex Systems. Wiley Verlag, New Jersey, USA, 3. Auflage, 2005
- [FMS12] FRIEDENTHAL, S.; MOORE, A.; STEINER, R.: A practical guide to SysML: The Systems Modeling Language. Morgan Kaufmann Press, Boston, USA, 2. Ausgabe, 2012
- [FO07] FORTUIN, F. T. J. M.; OMTA, S. W. F.: The Length of the Product Generation Life Cycle as a Moderator of Innovation Strategy: A Comparative Cross-Industry Study of Ten Leading Technology-Based Companies. Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS'07), January 3-6 2007, Waikoloa, Hawaii, USA, 2007
- [FS05] FRICKE, E.; SCHULZ, A.: Design for Changeability (DfC): Principles To Enable Changes in Systems Throughout Their Entire Lifecycle. Systems Engineering, 8(4), 2005, S. 342-359
- [Gab14-ol] SPRINGER GABLER VERLAG (HRSG.): Gabler Wirtschaftslexikon – Stichwort: Systematik. Unter: <http://wirtschaftslexikon.gabler.de/Archiv/17721/systematik-v8.html>, 15. März 2014
- [Gab15a-ol] SPRINGER GABLER VERLAG (HRSG.): Gabler Wirtschaftslexikon – Stichwort: Unternehmenskonzentration. Unter: <http://wirtschaftslexikon.gabler.de/Definition/unternehmenskonzentration.html>, 4. Dezember 2015
- [Gab15b-ol] SPRINGER GABLER VERLAG (HRSG.): Gabler Wirtschaftslexikon – Stichwort: Produkt. Unter: <http://wirtschaftslexikon.gabler.de/Definition/produkt.html>, 5. Januar 2016
- [GAC+13] GAUSEMEIER, J.; ANACKER, H.; CZAJA, A.; WASSMANN, H.; DUMITRESCU, R.: Auf dem Weg zu intelligenten technischen Systemen. In: Gausemeier, J.; Dumitrescu, R.; Rammig, F.-J.; Schäfer, W.; Trächtler, A. (Hrsg.): 9. Paderborner Workshop Entwurf mechatronischer Systeme, 18.-19. April 2013, HNI-Verlagsschriftenreihe, Band 310, Paderborn, 2013
- [GAG+13] GAJSKI, D.D.; ABDI, S.; GERSTLAUER, A.; SCHIRNER, G.: Embedded System Design – Modeling, Synthesis and Verification. Springer Verlag, New York, USA, 2009
- [Gau10] GAUSEMEIER, J. (HRSG.): Frühzeitige Zuverlässigkeitsanalyse mechatronischer Systeme. Carl Hanser Verlag, München, 2010
- [GB12] GEISBERGER, E.; BROY, M.: agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems (acatech Studie). Springer Verlag, Heidelberg, 2012
- [GDJ+14] GAUSEMEIER, J.; DUMITRESCU, R.; JASPERNEITE, J.; KÜHN, A.; TRSEK, H.: Der Spitzencluster it's OWL auf dem Weg zu Industrie 4.0. Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZWF), 5/2014, S. 336-346
- [GDS+13] GAUSEMEIER, J.; DUMITRESCU, R.; STEFFEN, D.; CZAJA, A.; WIEDERKEHR, O.; TSCHIRNER, C.: Systems Engineering in der industriellen Praxis. Heinz Nixdorf Institut, Fraunhofer-Projektgruppe für Entwurfstechnik Mechatronik, UNITY AG, Paderborn, 2013
- [GE13] GERACIE, G.; EPPINGER, S. D.: The Guide to the Product Management and Marketing Body of Knowledge (ProdBOK). Product Management Educational Institute, Carson City, 2013
- [GEK01] GAUSEMEIER, J.; EBBESMEYER, P.; KALLMEYER, F.: Produktinnovation – Strategische Planung und Entwicklung der Produkte von morgen. Hanser Verlag, München, 2001

- [Ger05] GERPOTT, T.J.: Strategisches Technologie- und Innovationsmanagement. Schäffer-Poeschel Verlag, Stuttgart, 2. Auflage, 2005
- [GLL12] GAUSEMEIER, J.; LANZA, G.; LINDEMANN, U.: Produkte und Produktionssysteme integrativ konzipieren – Modellbildung und Analyse in der frühen Phase der Produktentstehung, Hanser Verlag, München, 2012
- [GF06] GAUSEMEIER, J.; FELDMANN, K.: Integrative Entwicklung räumlicher elektronischer Baugruppen. Carl Hanser Verlag, München, 2006
- [GK12] GAUSEMEIER, J.; KOKOSCHKA, M.: Schutzmaßnahmen vor Produktpiraterie. In: Gausemeier, J.; Glatz, R.; Lindemann, U.: Präventiver Produktschutz – Leitfäden und Anwendungsbeispiele, Carl Hanser Verlag, München, 2012
- [GLR+00] GAUSEMEIER, J.; LINDEMANN, U.; REINHART, G.; WIENDAHL, H-P.: Kooperatives Produktengineering – Ein neues Selbstverständnis des ingenieurmäßigen Wirkens. HNI-Verlagsschriftenreihe, Band 79, 2000
- [GMW10] GERALDI, J.; MAYLOR, H.; WILLIAMS, T.: Now, let's make it really complex (complicated) - A systematic review of the complexities of projects, International Journal of Operations & Production Management, 31(9), 2011, S. 966-990
- [Göp98] GÖPFERT, J.: Modulare Produktentwicklung – Zur gemeinsamen Gestaltung von Technik und Organisation. Gabler Verlag, Wiesbaden, 1998
- [GP14] GAUSEMEIER, J.; PLASS, C.: Zukunftsorientierte Unternehmensgestaltung – Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen. Carl Hanser Verlag, München, 2. Auflage, 2014
- [GRS14] GAUSEMEIER, J.; RAMMIG, F.; SCHÄFER, W. (HRSG.): Design Methodology for Intelligent Technical Systems, Springer Verlag, Berlin, 2014
- [GSK+14] GEISSBAUER, R.; SCHRAUF, S.; KOCH, V.; KUGE, S.: Industrie 4.0 – Chancen und Herausforderungen der vierten industriellen Revolution. PricewaterhouseCoopers Aktiengesellschaft (Hrsg.), Frankfurt am Main, 2014
- [GTS14] GAUSEMEIER, J.; TRÄCHTLER, A.; SCHÄFER, W.: Semantische Technologien im Entwurf mechatronischer Systeme: Effektiver Austausch von Lösungswissen in Branchenwertungsketten. Carl Hanser Verlag, München, 2014
- [GW06] GORSCHKE, T.; WOHLIN, C.: Requirements Abstraction Model. Requirements Engineering, 11(1), 2006, S. 79-101
- [HHA12] HAHN, A.; HÄUSLER, S.; GROBE AUSTING, S.: Quantitatives Entwicklungsmanagement: Modellbasierte Analyse von Produktentwicklungsprozessen. Springer Verlag, Berlin, 2012
- [HC88] HAUSER, J. R.; CLAUSING, D.: The House of Quality. Harvard Business Review, 66/1988, S. 63-73
- [HC90] HENDERSON, R.; CLARK, K.: Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. Administrative Science Quarterly, Special Issue: Technology, Organizations, and Innovation, 35(1), 1990, S. 9-30
- [HDL10] HELLENBRAND, D.; DANILIDIS, C.; LINDEMANN, U.: Integrierte und funktionsorientierte Produkt- und Prozessmodellierung mechatronischer Produkte. In: Gausemeier, J. (Hrsg.): 7. Paderborner Workshop Entwurf mechatronischer Systeme, 18.-19. März 2010, HNI-Verlagsschriftenreihe, Band 272, Paderborn, 2010
- [Hep13] HEPPELE, C.: Planung lebenszyklusgerechter Leistungsbündel. Verlag Dr. Hut, München, 2013
- [HK15] HARTING, K.; KÜHN, A.: Umsetzungsplanung via Mechatronikroadmap. develop³ - Systems Engineering, 3/2015, Konradin-Verlag, Leinfelden-Echterdingen, 2015

- [HOL+11] HEPPELLE, C.; ORAWSKI, R.; LANGER, S.; MÖRTL, M.; LINDEMANN, U.: Temporal aspects in lifecycle-oriented planning of product-service-systems. ICORD 11: Proceedings of the 3rd International Conference on Research into Design Engineering, January 10-12 2011, Bangalore, Indien, 2011
- [HS07] HAUSCHILDT, J.; SALOMO, S.: Innovationsmanagement. Vahlen Verlag, München, 4. Auflage, 2007
- [HTF96] HARASHIMA, F.; TOMIZUKA, M.; FUKUDA, T.: Mechatronics – What is it, why, and how?, An Editorial, IEEE/ASME Transactions on Mechatronics, 1996, S. 1-4
- [HWF+07] HOOD, C.; WIEDEMANN, S.; FICHTINGER, S.; PAUTZ, U.: Requirements Management: The Interface between Requirements Development and all other Systems Engineering Processes. Springer Verlag, Berlin, 2007
- [HWF+12] HABERFELLNER, R.; WECK DE, O. L.; FRICKE, E.; VÖSSNER, S.: Systems Engineering – Grundlagen und Anwendung. Orell Füssli, Zürich, 2012
- [Ian98] IANSITI, M.: Technology Integration: Making Critical Choices in a Turbulent World. Harvard Business School Press, Boston, 1998
- [IKD+13] IWANEK, P.; KAISER, L.; DUMITRESCU, R.; NYBEN, A.: Fachdisziplinübergreifende Systemmodellierung mechatronischer Systeme mit SysML und CONSENS. In: Maurer, M.; Schulze, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2013
- [INC12] INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING (INCOSE): INCOSE Systems Engineering Handbuch – Deutsche Übersetzung. GfSE, 2012
- [Ise08] ISERMANN, R.: Mechatronische Systeme – Grundlagen. Springer Verlag, Berlin, 2008
- [itsow15] IT'S OWL CLUSTERMANAGEMENT GMBH: Wie die Intelligenz in die Maschine kommt. Projektübersicht des Spitzenclusters it's OWL, März 2015
- [JEC04] JARRATT, T.A.W.; ECKERT, C.M.; CLARKSON, P.J.: Development of a Product Model to support Engineering Change Management. Proceedings of the TMCE 2004, April 12-16, Lausanne, Switzerland, 2004
- [JEC+11] JARRATT, T.A.W.; ECKERT, C.M.; CALDWELL, N.H.M.; CLARKSON, P.J.: Engineering change – an overview and perspective on the literature. Research in Engineering Design, 22(2), 2011, S. 103-124
- [JJS07] JIAO, J.; SIMPSON, T. W.; SIDDIQUE, Z.: Product family design and platform-based product development: a state-of-the-art review. Journal of Intelligent Manufacturing, Springer Science + Business Media, 18(1), 2007, S. 5-29
- [Kah12] KAHL, S.M.: Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 308, Paderborn, 2012
- [Kai13] KAISER, L.: Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 327, Paderborn, 2013
- [Kam12] KAMISKE, G. F. (HRSG.): Handbuch QM-Methoden – Die richtige Methode auswählen und erfolgreich umsetzen. Carl Hanser Verlag, München, 2012
- [KBD+14] KÜHN, A.; BREMER, C.; DUMITRESCU, R.; GAUSEMEIER, J.: Feature models supporting trade-off decisions in early mechatronic systems design. Proceedings of NordDesign 2014, August 27-29 2014, Finland, 2014
- [KCC12] KOH, E.C.Y.; CALDWELL, N.H.M.; CLARKSON, P.J.: A method to assess the effects of engineering change propagation. Research in Engineering Design, 23(4), 2012, S. 329-351

- [KCH+90] KANG, K.; COHEN, S.G.; HESS, J.A.; NOVAK, W.E.; PETERSON, S.A.: Feature-Oriented Domain Analysis (FODA) – Feasibility Study. Technical Report CMU/SEI-90-TR-21, Carnegie-Mellon University, 1990
- [KDB+13] KAISER, L.; DUMITRESCU, R.; BREMER, C.; GAUSEMEIER, J.: Sichten der Systemstruktur im Model-Based Systems Engineering für mechatronische Systeme. In: Stuttgarter Symposium für Produktentwicklung, Stuttgart, 2013
- [KDH+13] KAISER, L.; DUMITRESCU, R.; HOLTSMANN, J.; MEYER, M.: Automatic Verification of Modeling Rules in Systems Engineering for Mechatronic Systems. In: Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, August 4-7 2013, Portland, Oregon, USA, 2013
- [KDK06] KRANKEL, R.M.; DUENYAS, I.; KAPUSCINSKI, R.: Timing Successive Product Introductions with Demand Diffusion and Stochastic Technology Improvement. *Manufacturing & Service Operations Management*, 2/2006, S. 119-135
- [KE08] KIESBAUER, J.; ERBEN, S.: Integration kommunikationsfähiger Stellgeräte in Leitsysteme. Sonderdruck atp – Automatisierungstechnische Praxis, 50/2008
- [KEE+05] KELLER, R.; EGER, T.; ECKERT, C.; CLARKSON, P. J.: Visualising change propagation. In: Samuel, A.; Lewis, W. (Eds.): Proceedings of the 15th International Conference on Engineering Design, ICED, August 15-18 2005, Melbourne, Australien, 2005, S. 1-12
- [Kes12] KESPER, H.: Gestaltung von Produktvariantenspektren mittels matrixbasierter Methoden. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, 2012
- [KH13] KERNSCHMIDT, K.; VOGEL-HEUSER, B.: An interdisciplinary SysML based modeling approach for analyzing change influences in production plants to support the engineering. IEEE International Conference on Automation Science and Engineering (CASE), August 17-20 2013, S. 1113-1118
- [KM05] KIM, W. C.; MAUBORGNE, R.: Der blaue Ozean als Strategie – Wie man neue Märkte schafft wo es keine Konkurrenz gibt. Carl Hanser Verlag, München, 2005
- [Köh09] KÖHLER, C.: Technische Produktänderung – Analyse und Beurteilung von Lösungsmöglichkeiten auf Basis einer Erweiterung des CPM/PDD-Ansatzes. Dissertation, Universität des Saarlandes, Schriftenreihe Produktionstechnik, Band 45, Saarbrücken, 2009
- [Krö07] KRÖLL, M.: Methode zur Technologiebewertung für eine ergebnisorientierte Produktentwicklung. Dissertation, Fakultät Maschinenbau, Universität Stuttgart, Jost-Jetter Verlag, Heimsheim, 2007
- [KSL12] KISSEL, M.; SCHRIEVERHOFF, P.; LINDEMANN, U.: Design for Adaptability – Identifying Potential for Improvement on an Architecture Basis. Proceedings of NordDesign 2012, August 22-24 2012, Aarlborg, Denmark, 2014
- [KSS+11] KOSSIAKOFF, A.; SWEET, W.N.; SEYMOUR, S.; BIEMER, S.M.: Systems Engineering Principles and Practice. Wiley Verlag, New Jersey, USA, 2. Auflage, 2011
- [KST12] KIVI, A.; SMURA, T.; TÖYLI, J.: Technology product evolution and the diffusion of new product features. *Technological Forecasting & Social Change*, 79/2012, S. 107-126
- [LA06] LAUBE, T.; ABELE, T.: Technologie-Roadmap – Strategisches und taktisches Technologiemanagement: Ein Leitfaden. Fraunhofer IRB Verlag, Stuttgart, 2006
- [LDA11] VAN DE LAAR, P.; DOUGLAS, A.U.; AMERICA, P.: Research Evolvability. In: van de Laar, P.; Punter, T. (Eds.): Views on evolvability of embedded systems. Springer Verlag, Dordrecht, 2011
- [Lev06] LÉVÁRDY, V.: Model-based Framework for the Adaptive Development of Engineering Systems. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, 2006

- [Lin05] LINDEMANN, U.: Methodische Entwicklung technischer Produkte – Methoden flexibel und situationsgerecht anwenden. Springer Verlag, Heidelberg, 2005
- [Lin07] LINDGREN, M.: Release Planning in Industry – Interview Data. MRTC Report ISSN 1404-3041, 2007
- [Lin08] LINDGREN, M.: Towards A Capability Model For Release Planning Of Software Intensive Systems. Dissertation, Mälardalen University Press, Band 67, 2008
- [LK14] LIN, C.-Y.; KREMER, G.E.O.: Strategic decision making for multiple-generation product lines using dynamic state variable models: The cannibalization case. *Computers in Industry*, 65/2014, S. 79-90
- [LL06] LECOINTRE, G.; LE GUYADER, H.: Biosystematik – Alle Organismen im Überblick. Springer Verlag, Berlin, 2006
- [LMB09] LINDEMANN, U.; MAURER, M.; BRAUN, T.: Structural complexity management. Springer Verlag, München, 2009
- [LMK05] LEHTOLA, L.; KAUPPINEN, M.; KUJALA, S.: Linking the Business View to Requirements Engineering: Long-Term Product Planning by Roadmapping. Proceedings of the 13th IEEE International Conference on Requirements Engineering, August 29 - September 2 2005
- [LR98] LINDEMANN, U.; REICHWALD, R. (HRSG.): Integriertes Änderungsmanagement. Springer Verlag, Berlin, 1998
- [LW10] LAMM, J. G.; WEILKIENS, T.: Funktionale Architekturen in SysML. In: MAURER, M.; SCHULZE, S.-O. (HRSG.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2010
- [LSS+07] LUCKE, H.; SCHAPER, D.; SIEPEN, P.; UELSCHEN, M.; WOLLBORN, M.: The Innovation Cycle Dilemma. In: Herzog, O.; Rödiger, K.-H.; Ronthaler, M.; Koschke, R. (Eds.): *Lecture Notes in Informatics*. 110/2007, S. 526-530
- [Mal15] MALIK, F.: Strategie des Managements komplexer Systeme: Ein Beitrag zur Management-Kybernetik evolutionärer Systeme. Haupt Verlag, Bern, 11. Auflage, 2015
- [MBD+05] VON DER MABEN, T.; BACHER, H.V.; DÖRR, J.; GEISBERGER, E.; HOUDEK, F.; MACGREGOR, J.; MÜLLER, K.; PAECH, B.; SINGH, H.; WUBMANN, H.: Einsatz von Features im Software-Entwicklungsprozess – Abschlussbericht des GI-Arbeitskreises Features. RWTH Aachen, Aachener Informatik Berichte, 2005
- [MBE+14] MAURER, M.; BAUER, W.; ELEZI, F.; CHUCKOŁOWSKI, N.: Methodik zur Erstellung zyklengerechter Modul- und Plattformstrategien. In: Vogel-Heuser, B.; Lindemann, U.; Reinhart, G. (Hrsg.): *Innovationsprozesse zyklensorientiert managen: Verzahnte Entwicklung von Produkt-Service-Systemen*. Springer Verlag, Berlin, 2014, S. 139-154
- [ML97] MEYER, M.; LEHNERD, A. P.: The power of product platform – building value and cost leadership. Free Press, New York, 1997
- [BCE14] BAUER, W.; CHUCHOŁOWSKI, N.; ELEZI, F.: Zyklensorientiertes Modul- und Plattformentdenken – Ein Leitfaden für Praktiker. Lehrstuhl für Produktentwicklung, TU München, 2014
- [MI02] MARTIN, M.V.; ISHII, K.: Design for variety – Developing Standardized and Modularized Product Platform Architectures. *Research in Engineering Design*, 13(4), 2002, S. 213-235
- [Möh04] MÖHRINGER, S.: Entwicklungsmethodik für mechatronische Systeme. Habilitation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 156, Paderborn, 2004
- [MT90] MARTELLO, S.; TOTH, P.: Knapsack problems: Algorithms and computer interpretations. Wiley-Interscience, 1990
- [MU12] MAYER, H.; UHLMANN, E.: Hybride Leistungsbündel – ein neues Produktverständnis. In: *Integrierte Industrielle Sach- und Dienstleistungen – Vermarktung, Entwicklung und Erbringung hybrider Leistungsbündel*, Springer Verlag, Berlin, 2012

- [NCB+13] NONSIRI, S.; COATANEA, E.; BAKHOUYA, M.; MOKAMMEL, F: Model-based approach for change propagation analysis in requirements. Systems Conference (SysCon), IEEE International, April 15-18 2013, S. 497-503
- [NL08] NIPPA, M.; LABRIOLA, F.: Der Roadmapping-Ansatz als integrative Planungsmethode im Rahmen eines situationsgerechten Time-to-Market Management. In: Möhrle, M.G.; Isenmann, R. (Hrsg.): Technologie-Roadmapping: Zukunftsstrategien für Technologieunternehmen. Springer Verlag, Berlin, 3. Auflage, 2008
- [OED14-o1] OXFORD ENGLISH DICTIONARY (OED) ONLINE: Evolution.
Unter: <http://www.oed.com/view/Entry/65447?redirectedFrom=evolution>, 20. Oktober 2014
- [OED15-o1] OXFORD ENGLISH DICTIONARY (OED) ONLINE: Release.
Unter: <http://www.oxforddictionaries.com/definition/english/release>, 4. Dezember 2015
- [OHS+12] ORAWSKI, R.; HEPPELLE, C.; SCHENKL, S.; MÖRTL, M.; LINDEMANN, U.: Life-Cycle Oriented Requirement Formalization and Traceability. In: Rivest, L.; Bouras, A.; Louhichi, B. (Eds.): Product Lifecycle Management – Towards Knowledge-Rich Enterprises. IFIP WG 5.1 International Conference, PLM 2012, July 9-11 2012, Montreal, Kanada, 2012, S. 124-133
- [OMG12] OBJECT MANAGEMENT GROUP: Systems Modeling Language (OMG SysML™) – Version 1.3, 2012
- [OS11] OEHMEN, J.; SEERING, W.: Risk-Driven Design Processes – Balancing Efficiency with Resilience in Product Design. In: Birkhofer, H. (Eds.): The Future of Design Methodology. Springer Verlag, London, 2011, S. 47-54
- [PBF+07] PAHL, G.; BEITZ, W.; FELDHUSEN, J.; GROTE, K.-H. (HRSG.): Konstruktionslehre – Grundlagen erfolgreicher Produktentwicklung – Methoden und Anwendung. Springer Verlag, Berlin, 7. Auflage, 2007
- [PDK+11] PAIGE, R.F.; DRIVALOS, N.; KOLOVOS, D.S.; FERNANDES, K.J.; POWER, C.; OLSEN, G.K.; ZSCHALER, S.: Rigorous identification and encoding of trace-links in model-driven engineering. Software & Systems Modelling, 10(4), Springer Verlag, 2011, S. 469-487
- [Pei15] PEITZ, C.: Systematik zur Entwicklung einer produktlebenszyklusorientierten Geschäftsmodell-Roadmap. Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlags-schriftenreihe, Band 337, Paderborn, 2002
- [PH14] PORTER, M.E.; HEPPELMANN, J.E.: How Smart, Connected Products Are Transforming Competition. Harvard Business Review, 2014, S. 1-23
- [PL11] PONN, J.; LINDEMANN, U.: Konzeptentwicklung und Gestaltung technischer Produkte – Systematisch von Anforderungen zu Konzepten und Gestaltlösungen. Springer Verlag, Berlin, 2. Auflage, 2011
- [Pop57] POPPER, K. R.: The Poverty of Historicism. Beacon Press, 1957
- [Rap99] RAPP, T.: Produktstrukturierung: Komplexitätsmanagement durch modulare Produktstrukturen und –Plattformen. Gabler Verlag, Wiesbaden, 1999
- [Rat93] RATHNOW, P.J.: Integriertes Variantenmanagement: Bestimmung, Realisierung und Sicherung der optimalen Produktvielfalt. Vandenhoeck & Ruprecht, Göttingen, 1993
- [Rei98] REINERTSEN, D.G.: Managing the Design Factory – a product developer’s toolkit. The Free Press, New York, 1997
- [Rop09] ROPOHL, G.: Allgemeine Systemtheorie der Technik, KIT Scientific Publishing, Karlsruhe, 2009
- [RRH08] ROSS, A.M.; RHODES, D.H.; HASTINGS, D.E.: Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining System Lifecycle Value. Systems Engineering, 11(3), 2008, S. 246-262

- [Ruh05] RUHE, G.: Software Release Planning. In: Handbook of Software Engineering and Knowledge Engineering, 3/2005, S. 1-21
- [Ruh10] RUHE, G.: Product Release Planning – Methods, Tools and Applications. Auerbach Publications, Boca Raton, Florida, USA, 2010
- [Rup09] RUPP, C.: Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis. Carl Hanser Verlag, München, 5. Auflage, 2009
- [SAA13] SCHUH, G.; ALEKSIC, S.; ARNOSCHT, J.: Module Based Release Planning for Technical Changes. Technology Management in the IT-Driven Services (PICMET), Proceedings of PICMET'13, IEEE, July 28 – August 1 2013, S. 1604-1617
- [Sal05] SALIU, M. O.: Software Release Planning For Evolving Systems. PhD Research Proposal, Department of Computer Science, University of Calgary, Calgary, Alberta, 2005
- [SBK08] SCHUH, G.; BECKERMANN, S.; KLAPPERT, S.: Technologie-Roadmapping. In: Gassmann, O.; Sutter, P. (Hrsg.): Praxiswissen Innovationsmanagement – Von der Idee zum Markterfolg. Carl Hanser Verlag, München, 2008
- [Sch61] SCHUMPETER, J. A.: Konjunkturzyklen – Eine theoretische, historische und statistische Analyse des kapitalistischen Prozesses. Vandenhoeck & Ruprecht, Göttingen, 1961
- [Sch05] SCHUH, G.: Produktkomplexität managen - Strategien, Methoden, Tools. Carl Hanser Verlag, München, 2. Auflage, 2005
- [Sch12] SCHUH, G. (HRSG.): Innovationsmanagement: Handbuch Produktion und Management. Springer Vieweg Verlag, Berlin, 2. Auflage, 2012
- [Sch13] SCHUH, G.: Lean Innovation. Springer Verlag, Berlin, 2013
- [SCF+00] SCHULZ, A.P.; CLAUSING, D.P.; FRICKE, E.; NEGELE, H.: Development and Integration of Winning Technologies as Key to Competitive Advantage. Systems Engineering, 3(4), 2000, S. 180-211
- [SDL+04] SCHUH, G.; DESOI, J. C.; LENDERS, M.; WITTE, V.: Release-Engineering: An Innovative Approach to Handle Complexity of Mechatronic Products. Proceedings of the 1st International Conference on Electrical/Electromechanical Computer Aided Design & Engineering, 15-16 November 2004, S. 36-40
- [SE04] SCHUH, G.; EVERSHEIM, W.: Release Engineering – An Approach to Control Rising System-Complexity. CIRP-Annals Manufacturing Technology, 53(1), 2004, S. 167-170
- [Sek05] SEKOLEC, R.: Produktstrukturierung als Instrument des Variantenmanagements in der methodischen Entwicklung modularer Produktfamilien. Dissertation, ETH Zürich, VDI-Verlag, Band 172, 2005
- [SFM+09] SUH, S.; FURST, M.; MIHALYOV, K.; DE WECK, O.: Technology Infusion for Complex Systems – A Framework and case Study. Systems Engineering, 13(2), Wiley Periodicals, Inc., 2009
- [SGF+10] SVAHNBERG, M.; GORSCHKE, T.; FELDT, R.; TORKAR, R.; BIN SALEEM, S.; SHAFIQUE, M.U.: A systematic review on strategic release planning models. Information and Software Technology, 52(3), 2010, S. 237-248
- [SK97] SPUR, G.; KRAUSE, F.-L.: Das virtuelle Produkt. Carl Hanser Verlag, München, 1997
- [SL05] SÄÄKSJÄRVI, M.; LAMPINEN, M.: Consumer perceived risk in successive product generations. European Journal of Innovation Management, 8(2), 2005, S. 145-156
- [Smi07] SMITH, P.G.: Flexible Product Development: Building Agility for Changing Markets. Wiley, New Jersey, USA, 2007

- [Soe02] SOETEBEER, M.: Methode zur Modellierung, Kontrolle und Steuerung von Produktstrategien. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlags-schriftenreihe, Band 106, Paderborn, 2002
- [Som07] SOMMERVILLE, I.: Software Engineering. Pearson Education, New York, 2007
- [SR98] SMITH, P.G.; REINERTSEN, D.G.: Developing products in half the time: new rules, new tools. John Wiley & Sons, New York, 1998
- [SR09] SAGE, P.A.; ROUSE, W.B.: Handbook of Systems Engineering and Management. Wiley Verlag, New York, 2. Auflage, 2009
- [SRH+08] SAUSER, B.J., RAMIREZ-MARQUEZ, J.E., HENRY, D.; DIMARZIO, D.: A System Maturity Index for the Systems Engineering Life Cycle. International Journal of Industrial and Systems Engineering, 3/2008
- [SS93] SANCHEZ, R.; SUDHARSHAN, D.: Real-time Market Research. Marketing Intelligence & Planning, 11(7), 1993, S. 29-38
- [SSA12] SCHUH, G.; SCHIFFER, M.; ARNOSCHT, J.: Scenario Based Development of Robust Product Architectures. Technology Management for Emerging Technologies (PICMET), July 29 – August 2 2012, S. 2542-2549
- [Ste62] STEWARD, D.: On an Approach to the Analysis of the Structure of Large Systems of Equations. SIAM Review, 5/1962, S. 321–342
- [Ste81] STEWARD, D.: The Design Structure System: A Method for Managing the Design of Complex Systems. IEEE Transaction on Engineering Management, 1981, S. 79-83
- [Ste98] STEINER, R.: System Architectures and Evolvability: Definitions and Perspective. INCOSE International Symposium, 8(1), 1998, S. 137-141
- [Ste07] STEFFEN, D.: Ein Verfahren zur Produktstrukturierung für fortgeschrittene mechatronische Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlags-schriftenreihe, Band 207, Paderborn, 2007
- [Tal10] TALEB, N.N.: Der Schwarze Schwan: Die Macht höchst unwahrscheinlicher Ereignisse. Deutscher Taschenbuch Verlag, München, 2010
- [The11-ol] THE MATHWORKS: MATLAB/Simulink. Homepage: <http://de.mathworks.com>
- [TS09] THÖRN, C.; SANDKUHL, K.: Feature Modelling: Managing Variability in Complex Systems. In: Tolk, A.; Jain, L.C. (Eds.): Complex Systems in Knowledge-based Environments. Springer Verlag, Berlin, 2009, S. 129-161
- [UE12] ULRICH, K. T.; EPPINGER, S. D.: Product Design and Development. Fifth Edition, McGraw-Hill, New York, 2012
- [Ulr95] ULRICH, K.: The role of product architecture in the manufacturing firm. Research policy 24(3), 1995, S.419-440
- [UP95] ULRICH, H.; PROBST, G.J.: Anleitung zum ganzheitlichen Denken und Handeln – Ein Brevier für Führungskräfte. Haupt, Bern, 4. Auflage, 1995
- [Ves02] VESTER, F.: Die Kunst vernetzt zu denken: Ideen und Werkzeuge für einen neuen Umgang mit Komplexität. Ein Bericht an den Club of Rome. Deutscher Taschenbuch Verlag, München, 2002
- [VLR14] VOGEL-HEUSER, B.; LINDEMANN, U.; REINHART, G. (HRSG.): Innovationsprozesse zyklusorientiert managen: Verzahnte Entwicklung von Produkt-Service Systemen. Springer Verlag, Berlin, 2014
- [WBF+97] WENZEL, S.; BAUCH, T.; FRICKE, E.; NEGELE, H.: Concurrent engineering and more – a systematic approach to successful product development. Proceedings of 7th International Symposium, INCOSE, Los Angeles, USA, 1997

- [WC93] WHEELWRIGHT, S.; CLARK, K.: Revolution der Produktentwicklung – Spitzenleistungen in Schnelligkeit, Effizienz und Qualität durch dynamische Teams. Campus Verlag, Frankfurt/New York, 1993
- [WEC07] DE WECK, O.; ECKERT, C.M.; CLARKSON, J.: A Classification Of Uncertainty For Early Product And System Design. International conference on engineering design, ICED'07, August 8-31 2007
- [Wei06] WEILKIENS, T.: Systems Engineering mit SysML/UML: Modellierung, Analyse, Design. Dpunkt-Verlag, Heidelberg, 2006
- [Wil99] WILLIAMS, T.M.: The need for new paradigms for complex projects. International Journal of Project Management, 17(5), 1999, S. 269-273
- [Wri12] WRIGHT, K. H.: Release Engineering Processes, Their Faults and Failures. Dissertation, Faculty of the Graduate School of the University of Texas at Austin, USA, 2012
- [WWN+10] WYNN, D.; WYATT, D.F.; NAIR, S.; CLARKSON, P.J.: An introduction to the Cambridge Advanced Modeller. Proceedings of the 1st International Conference on Modelling and Management of Engineering Processes, MMEP '10, July 19-20, Department of Engineering, University of Cambridge, UK, 2010, S. 1-4
- [ZPB+13] ZORN-PAULI, G.; PAECH, B.; BECK, T.; KAREY, H.; RUHE, G.: Analyzing an Industrial Strategic Release Planning, Process – A Case Study at Roche Diagnostics. In: Doerr, J.; Opdahl, A.L. (Eds.): Requirements Engineering: Foundation for Software Quality (REFSQ), April 8-11 2013, Springer Verlag, 2013, S. 269-284

Normen und Richtlinien

- [ANSI/PMI99-001-2004] AMERICAN NATIONAL STANDARDS INSTITUTE (ANSI); PROJECT MANAGEMENT INSTITUTE (PMI): A Guide to the Project Management Body of Knowledge (PMBOK Guide). Project Management Institute, Inc., Newton Square, USA, 2004
- [DIN1421] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (DIN): DIN 1421. Gliederung und Benennung in Texten – Abschnitte, Absätze, Aufzählungen. Beuth Verlag, Berlin, 1983
- [DIN199-4] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (DIN): DIN 199 – Teil 4: Begriffe im Zeichnungs- und Stücklistenwesen - Änderungen. Beuth Verlag, Berlin, 1981
- [DIN69901] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (DIN): DIN69901. Projektmanagement – Projektmanagementsystem. Beuth Verlag, Berlin, 2009
- [DIN69901-5] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (DIN): DIN69901. Projektmanagement – Projektmanagementsysteme – Teil 5: Begriffe. Beuth Verlag, Berlin, 2009
- [ISO10007] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO): Quality management systems – Guidelines for configuration management. 2003
- [ISO19505-1] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO): Object Management Group Unified Modeling Language (OMG UML) – Part 1: Infrastructure. 2012
- [ISO19505-2] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO): Object Management Group Unified Modeling Language (OMG UML) – Part 2: Superstructure. 2012
- [NASA/SP-6105] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA): Systems Engineering Handbook. Washington, USA, 2007

-
- [MIL-STD-498] DEPARTMENT OF DEFENSE (DOD): Military Standard 498 – Software Development and Documentation. Washington, USA, 1994
- [VDA4965] VERBAND DER AUTOMOBILINDUSTRIE (VDA): Engineering Change Management Recommendation. Verband der Automobilindustrie, Frankfurt/Main, 2009
- [VDI2206] VEREIN DEUTSCHER INGENIEURE (VDI): Entwicklungsmethodik für mechatronische Systeme. VDI-Richtlinie 2206, Beuth Verlag, Berlin, 2004
- [VDI2651] VEREIN DEUTSCHER INGENIEURE (VDI): Plant Asset Management (PAM) in der Prozessindustrie – Definition, Modell, Aufgabe, Nutzen. VDI-Richtlinie 2651, Beuth Verlag, Düsseldorf, 2009
- [VDI2800] VEREIN DEUTSCHER INGENIEURE (VDI): Wertanalyse. VDI-Richtlinie 2800, Beuth Verlag, Berlin, 2010

Anhang

Inhaltsverzeichnis	Seite
A1 Ergänzungen zur Problemanalyse.....	A-1
A1.1 Risiken und Trade-Offs in der Produktentwicklung	A-1
A1.2 Gruppen und Häufigkeiten von Änderungsursachen	A-1
A1.3 Gestaltungsprinzipien für die Systemarchitektur	A-2
A1.4 Vor- und Nachteile modularer Systemarchitekturen.....	A-4
A2 Ergänzungen zum Stand der Technik	A-7
A2.1 Kurzbeschreibung verschiedener Priorisierungstechniken.....	A-7
A2.2 Release-Planungsprozess EVOLVE II nach RUHE.....	A-8
A3 Ergänzungen zur Systematik.....	A-10
A3.1 Kriterien zur Klassifizierung von technischen Änderungen	A-10
A3.2 Qualitätskriterien zur Bewertung von Produktplänen nach DESCHAMPS ET AL.	A-12
A3.3 Beispiel für die Struktur einer Wettbewerbs-Roadmap	A-13
A3.4 Alternative Release-Typen.....	A-14

A1 Ergänzungen zur Problemanalyse

A1.1 Risiken und Trade-Offs in der Produktentwicklung

Compensation approach	Risk Area			
	Expense overrun	Unit cost overrun	Performance shortfall	Schedule delay
Expenses		<ul style="list-style-type: none"> Reengineer high cost subsystems 	<ul style="list-style-type: none"> Reengineer low performance subsystems 	<ul style="list-style-type: none"> Increase program resources Pay for priority service
Unit cost	<ul style="list-style-type: none"> Relax unit cost goals Reuse subsystems 		<ul style="list-style-type: none"> Relax unit cost goals Use more expensive subsystems Reuse subsystems 	<ul style="list-style-type: none"> Accept high initial costs Reuse subsystems
Performance	<ul style="list-style-type: none"> Relax performance goals Drop some features Resegment market Use two step introduction 	<ul style="list-style-type: none"> Relax performance goals Drop some features Resegment target market 		<ul style="list-style-type: none"> Relax performance goals Drop some features Resegment target market Use two step introduction
Schedule	<ul style="list-style-type: none"> Slip the schedule Reduce team size Use cheap but slow services 	<ul style="list-style-type: none"> Slip the schedule Fine tune the design Wait for costs to improve 	<ul style="list-style-type: none"> Slip the schedule Wait for technology to improve 	

Bild A-1: Risiken und Trade-Offs in der Produktentwicklung [Rei98, S. 231]

A1.2 Gruppen und Häufigkeiten von Änderungsursachen

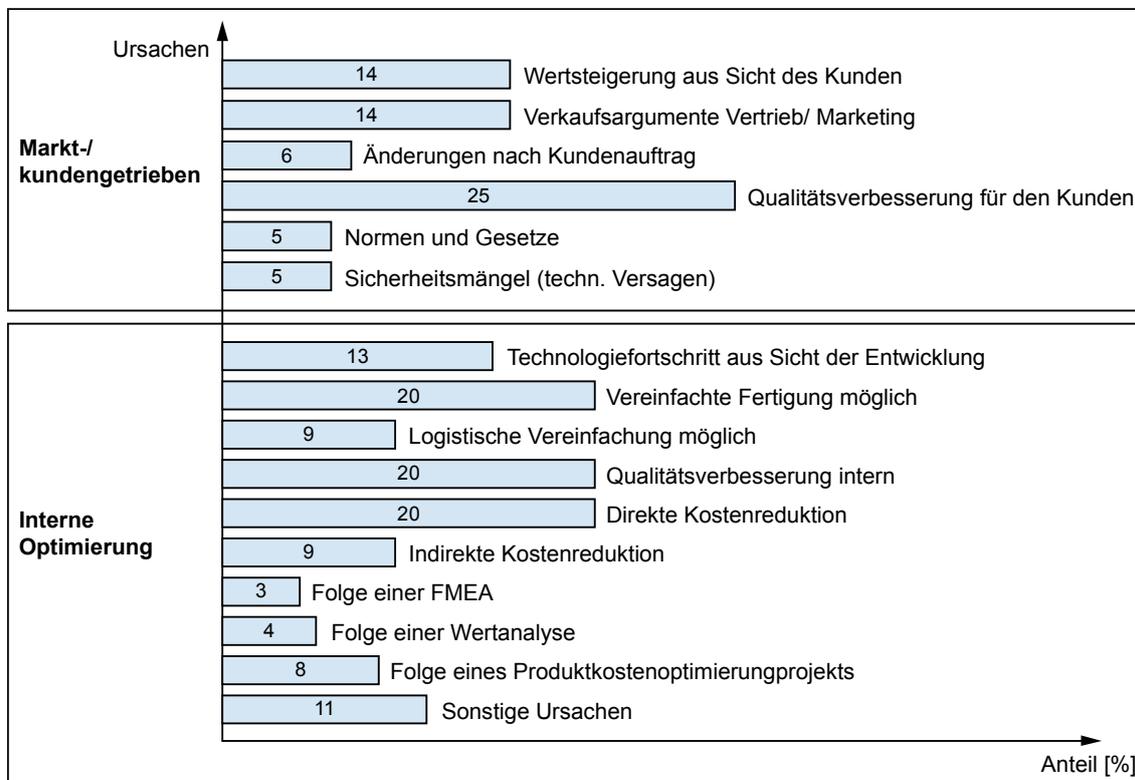


Bild A-2: Gruppen und Häufigkeiten von Änderungsursachen (Mehrfachnennungen möglich) [Sch13, S. 216]

A1.3 Gestaltungsprinzipien für die Systemarchitektur

Beschreibung	Visualisierung																																																																																																												
<p>Idealität/Einfachheit Ziel, die Komplexität des Systems zu verringern; abgeleitet von TRIZ Basis-Modell; nutzt vorhandene Ressourcen und wendet Praktiken der Konstruktionsoptimierung (-verschlankeung) an; Bsp. BMW i-Drive Konzept; nachteilige Wechselwirkung mit Unabhängigkeitsprinzip</p>	<p>a) Funktion wurde auf ein anderes Subsystem übertragen so dass ein vorhandenes Subsystem überflüssig ist</p> <p>b) Funktionsobjekt wurde eliminiert und Subjekt wird neuer Funktionsträger</p> <p>c) Funktionsobjekt wurde eliminiert und Systemumgebung wird zum Funktionsträger</p>																																																																																																												
<p>Unabhängigkeit Ziel, die Auswirkungen der Änderungen von Entwurfsparametern zu minimieren; jede Systemfunktion oder funktionelle Anforderung muss durch ein unabhängige Entwurfsparameter erfüllt werden, d.h. die Änderung eines Parameters beeinflusst kein verwandtes/verbundenes Parameter; vorteilhafte Wechselwirkung mit Prinzipien der Modularität/ Abkapselung, Eigenständigkeit (Autonomie) und nicht-hierarchische Integration</p>	<table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td></td><td>Funktion A</td><td>Funktion B</td><td>Funktion C</td><td>Funktion D</td><td>Funktion E</td></tr> <tr><td>Parameter A</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> <tr><td>Parameter B</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> <tr><td>Parameter C</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> <tr><td>Parameter D</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> <tr><td>Parameter E</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> </table> <p>a) Gekoppeltes Design</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td></td><td>Funktion A</td><td>Funktion B</td><td>Funktion C</td><td>Funktion D</td><td>Funktion E</td></tr> <tr><td>Parameter A</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> <tr><td>Parameter B</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> <tr><td>Parameter C</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> <tr><td>Parameter D</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> <tr><td>Parameter E</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> </table> <p>b) Entkoppeltes Design</p> <table border="1" style="display: inline-table;"> <tr><td></td><td>Funktion A</td><td>Funktion B</td><td>Funktion C</td><td>Funktion D</td><td>Funktion E</td></tr> <tr><td>Parameter A</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> <tr><td>Parameter B</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> <tr><td>Parameter C</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> <tr><td>Parameter D</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> <tr><td>Parameter E</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr> </table> <p>c) Losgelöstes Design</p>		Funktion A	Funktion B	Funktion C	Funktion D	Funktion E	Parameter A	●	●	●	●	●	Parameter B	●	●	●	●	●	Parameter C	●	●	●	●	●	Parameter D	●	●	●	●	●	Parameter E	●	●	●	●	●		Funktion A	Funktion B	Funktion C	Funktion D	Funktion E	Parameter A	●	●	●	●	●	Parameter B	●	●	●	●	●	Parameter C	●	●	●	●	●	Parameter D	●	●	●	●	●	Parameter E	●	●	●	●	●		Funktion A	Funktion B	Funktion C	Funktion D	Funktion E	Parameter A	●	●	●	●	●	Parameter B	●	●	●	●	●	Parameter C	●	●	●	●	●	Parameter D	●	●	●	●	●	Parameter E	●	●	●	●	●
	Funktion A	Funktion B	Funktion C	Funktion D	Funktion E																																																																																																								
Parameter A	●	●	●	●	●																																																																																																								
Parameter B	●	●	●	●	●																																																																																																								
Parameter C	●	●	●	●	●																																																																																																								
Parameter D	●	●	●	●	●																																																																																																								
Parameter E	●	●	●	●	●																																																																																																								
	Funktion A	Funktion B	Funktion C	Funktion D	Funktion E																																																																																																								
Parameter A	●	●	●	●	●																																																																																																								
Parameter B	●	●	●	●	●																																																																																																								
Parameter C	●	●	●	●	●																																																																																																								
Parameter D	●	●	●	●	●																																																																																																								
Parameter E	●	●	●	●	●																																																																																																								
	Funktion A	Funktion B	Funktion C	Funktion D	Funktion E																																																																																																								
Parameter A	●	●	●	●	●																																																																																																								
Parameter B	●	●	●	●	●																																																																																																								
Parameter C	●	●	●	●	●																																																																																																								
Parameter D	●	●	●	●	●																																																																																																								
Parameter E	●	●	●	●	●																																																																																																								
<p>Modularität/Abkapselung Ziel, eine Systemarchitektur aufzubauen, in der die Systemfunktionen durch Module erfüllt werden; lose Koppelung zwischen den Modulen und starker Zusammenhalt in den Modulen; unterstützt die Wiederverwendung von Elementen, Modulen oder ganzen Abschnitten einer Architektur; Implementierung von Plattformkonzepten oder Nutzung als Referenzarchitekturen möglich; weitverbreitetes Prinzip bei Systemarchitekturen; Zuverlässigkeit durch schwächste Komponente bestimmt</p>	<p>a) Komponenteneinteilung Modularität</p> <p>b) Komponententausch Modularität</p> <p>c) Fertigung nach Maß Modularität</p> <p>d) Gemischte Modularität</p>																																																																																																												
<p>Integrierbarkeit Ziel, Flexibilität und Anpassungsfähigkeit zu gewährleisten; Kompatibilität und Interoperabilität durch Verwendung gemeinsamer, offener und generischer Schnittstellen; nötig in sich schnell ändernden Umgebungen; nachteilige Wechselwirkung mit Prinzip der Dezentralisierung, vorteilhafte Wechselwirkung mit Prinzipien der Skalierbarkeit, nicht-hierarchischen Integration und Redundanz, Bsp. USB-Port, Bluetooth-Standard, AUTOSAR</p>	<p>a) Proprietäre Schnittstellen (alle Geräte besitzen spezifische Schnittstellen und sind nicht austauschbar)</p> <p>b) Generische und allgemeine Schnittstellen (alle Geräte besitzen eine gemeinsame Schnittstelle und sind austauschbar)</p>																																																																																																												

Bild A-3: Gestaltungsprinzipien für die Systemarchitektur Teil 1 [FS05]

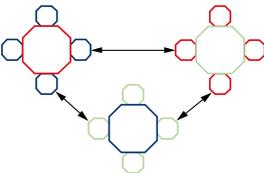
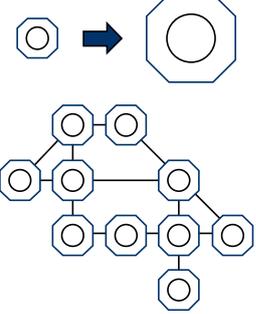
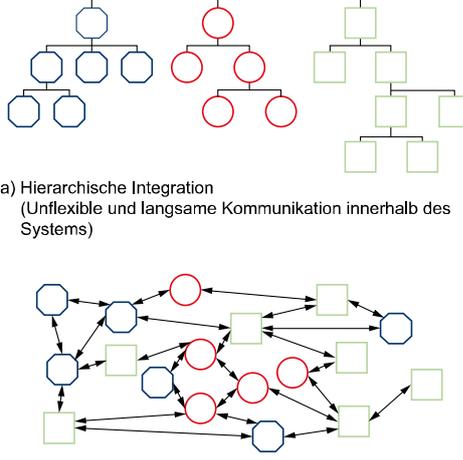
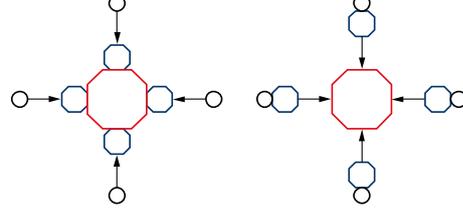
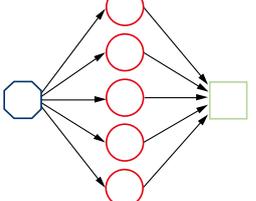
<p>Eigenständigkeit (Autonomie) Wichtig für Flexibilität und Anpassungsfähigkeit; gekennzeichnet durch Objekte, die grundlegende Funktionen bereitstellen, um die Unabhängigkeit von eingebetteten Systemen zu gewährleisten; vorteilhafte Wechselwirkung mit Unabhängigkeitsprinzip; Bsp. Satellitensysteme ändern automatisch ihren Modus, sofern spezifische Ereignisse auftreten</p>	 <p>Jedes Objekt erfüllt Basisfunktionalitäten, um die Unabhängigkeit von eingebetteten Systemen zu gewährleisten</p>
<p>Skalierbarkeit Durch vom Umfang unabhängige Einheiten gekennzeichnet; wichtig für Agilität, Flexibilität und Anpassungsfähigkeit; Architekturen können auf- und abwärts skaliert werden; Verlinkung vieler identischer Elemente oder Skalierung eines Elements; uneingeschränkte Erhöhung oder Verringerung der gesamten Einheiten innerhalb des Systems erforderlich; vorteilhafte Wechselwirkung mit Prinzip der Integrierbarkeit; Bsp. Mehrkernprozessoren</p>	 <p>a) Einzelelement durch Up-/Downsizing seiner charakteristischen Parameter skaliert b) Erhöhung/Verringerung der Anzahl mehrerer identischer Elemente, die verknüpft sind, um eine skalierte Architektur zu ermöglichen</p>
<p>Nicht-hierarchische Integration Gekennzeichnet durch Verknüpfung einzelner Einheiten über das Gesamtsystem, ohne Bezug auf jede Art der Modularität oder Einkapselung; wichtig für Agilität und Anpassungsfähigkeit; Schnittstellen auf gleicher hierarchischer Ebene ermöglichen eine schnelle, direkte und flexible Kommunikation und Interaktion zwischen einzelnen Einheiten/Objekten; kritisch ist die Verknüpfung von Einheiten/Objekten in verschiedenen Modulen oder Schichten der Systemarchitektur; Bsp. Internet (TCP/IP Protokoll)</p>	 <p>a) Hierarchische Integration (Unflexible und langsame Kommunikation innerhalb des Systems) b) Nicht-hierarchische Integration (Interaktionsschnittstellen auf gleicher Ebene gewährleisten eine schnelle und flexible Kommunikation)</p>
<p>Dezentralisierung Gekennzeichnet durch eine dezentrale Verteilung der Steuerung, Informationen, Ressourcen, Attribute und Eigenschaften innerhalb der Systemarchitektur; wichtig für Agilität und Anpassungsfähigkeit; Dezentralisierung verstärkt die Fähigkeit des Systems, sich schnell seiner Umgebung anzupassen und anonym auf wechselnde Anforderungen zu reagieren; Bsp. Fly-by-wire Flugsteuerung</p>	 <p>a) Zentralistisches System (alle Ressourcen befinden sich an einem Ort) a) Dezentrales System (auf geeignete Orte verteilte Ressourcen)</p>
<p>Redundanz Wichtig für Flexibilität und Robustheit, da das Prinzip Kapazitäts-, Funktionalitäts- und Performanceoptionen sowie Fehlertoleranz ermöglicht; vorteilhafte Wechselwirkung mit Prinzipien der Skalierbarkeit und Integrierbarkeit; nachteilige Wechselwirkung mit Prinzip der Unabhängigkeit</p>	 <p>Redundante Elemente erhöhen die Fehlertoleranz und Ausfallsicherheit</p>

Bild A-4: Gestaltungsprinzipien für die Systemarchitektur Teil 2 [FS05]

A1.4 Vor- und Nachteile modularer Systemarchitekturen

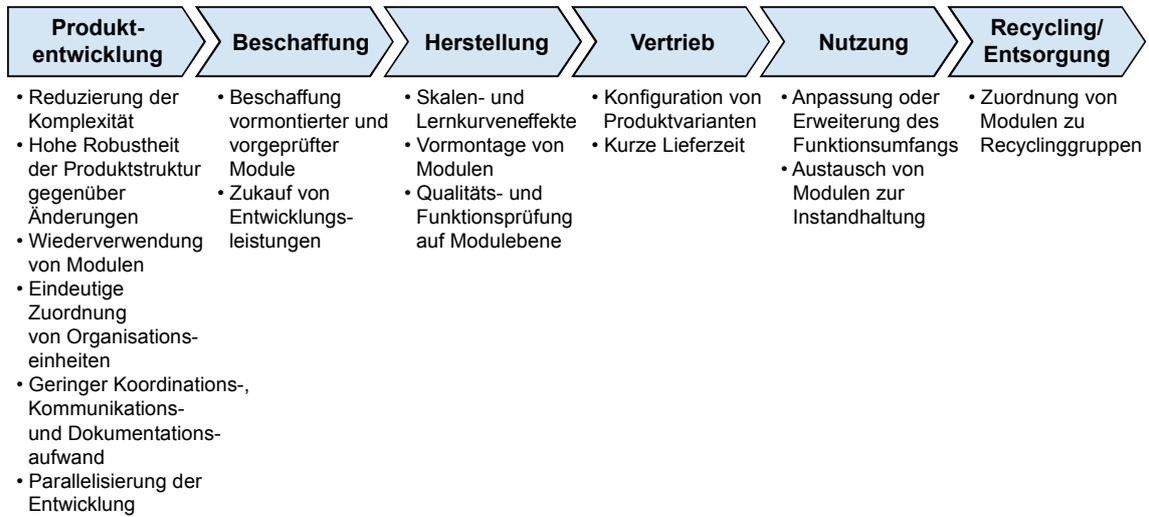


Bild A-5: Vorteile modularer Systemarchitekturen entlang des Produktlebenszyklus [Ble11]

Modulare Produktstrukturen		
Merkmale	Vorteile	Nachteile
Produkt Systemumfang Produktleistung Produktqualität und -zuverlässigkeit Erweiterbarkeit	<ul style="list-style-type: none"> Hierarchische Strukturierung erleichtert das Produktverständnis Einfachere Dokumentation Fehlerreduzierung durch Modulprüfung Vereinfachter Funktionstest des Gesamtsystems Einfache Identifikation defekter Module Reparatur durch Austausch von Modulen Nachträgliche Erweiterung der Funktionalität 	<ul style="list-style-type: none"> Hoher Aufwand für Strukturierung Zusatzaufwand für Spezifikation Notwendigkeit redundanter Funktionen Platz-, Leistungs- und Effizienzverlust durch Schnittstellen Tendenz zur Überdimensionierung von Modulen und Schnittstellen Gefahr von unerwünschten Seiteneffekten (Einzelkontrolle der Module garantiert nicht Gesamtfunktion) Beschränkung der Reparierbarkeit auf Modulaustausch Vorsehen von Schnittstellen mit Tendenz zur Überdimensionierung
Produktprogramm Wiederverwendung Mehrfachverwendung Standardisierbarkeit	<ul style="list-style-type: none"> Stabile ausgereifte Struktur auch bei Veränderung einzelner Module Wiederverwendung bewährter Module Einfache Variantenbildung durch Baukastensystem (Kombinierbarkeit) Reduktion des Entwicklungsaufwands Skalen- und Synergieeffekte Individualisierbarkeit der Produkte Anpassbarkeit an Länderspezifika Standardisierung existierender Lösungen durch einheitliche Schnittstellen Hohe Verfügbarkeit und kurze Lieferzeit (Zukauf von Standardlösungen) 	<ul style="list-style-type: none"> Gefahr der späten Entdeckung von Konzeptfehlern Vorplanung aller Varianten Geringe Differenz der Varianten Kostennachteile durch Leistungsreserven in Modulen/Schnittstellen Eingeschränkte Individualisierbarkeit Substituierbarkeit von Modulen/Ersatzteilen Geringe Originalität des Produktes
Produktentstehung Entwicklung Fertigung/Montage	<ul style="list-style-type: none"> Parallelisierung der Entwicklung durch Entkopplung und definierte Schnittstellen Reduktion der Teilekomplexität Beschleunigung der Produktion Reduzierter Montageaufwand durch vorgefertigte Module mit wenigen physischen Schnittstellen Vereinfachte Demontage 	<ul style="list-style-type: none"> Aufwändige Spezifikation der Schnittstellen Hoher Aufwand für die initiale Definition der Produktstruktur

Bild A-6: Vor- und Nachteile modularer Systemarchitekturen nach [Ste07, S. 25]

A2 Ergänzungen zum Stand der Technik

A2.1 Kurzbeschreibung verschiedener Priorisierungstechniken

Analytical Hierarchy Process (AHP)

Der Analytical Hierarchy Process ist eine systematische Methode der Entscheidungsunterstützung für die Priorisierung zwischen Alternativen. Dabei werden alle möglichen Paare hierarchisch klassifizierter Alternativen miteinander verglichen. Ziel ist die Bestimmung einer Prioritätsreihenfolge (1 = gleich bedeutend, 9 = sehr viel wichtiger). Da paarweises Vergleichen auf jeder Hierarchieebene für jede Alternative getätigt wird, ergibt sich oftmals eine sehr hohe Anzahl an Vergleichen. Um die Ermittlung der Priorität zu vereinfachen, wird die Anzahl der Vergleiche mithilfe von verschiedenen Methoden um bis zu 75% reduziert [BA05, S. 75f.].

100 Dollar Test

Der 100 Dollar Test ist eine einfache Priorisierungsmethode, bei der den Stakeholdern 100 imaginäre Einheiten gestellt werden (z.B. Geld oder Stunden), um diese priorisiert auf alle Alternativen zu verteilen. Das Ergebnis dieser Methode wird in einer Verhältnisskala abgebildet. Im Falle einer zu hohen Anzahl an zu bewertender Alternativen empfiehlt es sich hierbei, die Anzahl an zu verteilenden Einheiten auf z.B. 1.000, 10.000 oder 100.000 zu erhöhen [BA05, S. 76].

Rangfolge

In dieser Methode werden vorhandene Alternativen bestimmten Rängen zugewiesen, wobei der erste Rang als am wichtigsten eingestuft wird und der n-te Rang als unwichtigster gilt. Das Ergebnis wird hierbei auf einer ordinalen Skala abgebildet. Es können keine Auskünfte über den Abstand zwischen den einzelnen Rangfolgen getroffen werden, da hier nur subjektiv, ohne die Verteilung von Punkten, gearbeitet wird. Die Anordnung der Alternativen kann über verschiedene Wege ermittelt werden, wie z.B. durch das Bubble-sort-Verfahren oder binäre Suchbaum-Algorithmen [BA05, S. 77].

Top-Ten

In diesem Verfahren wählt der Stakeholder aus allen verfügbaren Alternativen seine persönliche Top-Ten aus. Ziel ist es, die Gleichberechtigung verschiedener Stakeholder zu gewährleisten. Dabei ist es für die Auswertung wichtig, essentielle Alternativen, die jedem Stakeholder wichtig sind, herauszustellen und eine Balance zwischen den ausgewählten Alternativen zu erhalten, um möglichst jeden Stakeholder zufrieden stellen zu können [BA05, S. 77].

A2.2 Release-Planungsprozess EVOLVE II nach RUHE

EVOLVE II nach RUHE beschreibt ein Vorgehen für die Entscheidungsunterstützung bei der Planung eines Produkt-Release. Ziel ist es, unter Berücksichtigung von Feature-Abhängigkeiten und Ressourcen-Restriktionen ein harmonisiertes Bündel von Features für die spätere Umsetzung auszuwählen [Ruh10, S. 105ff.]. Bild A-7 gibt einen Überblick über die 8 obligatorischen Schritte der Methode.

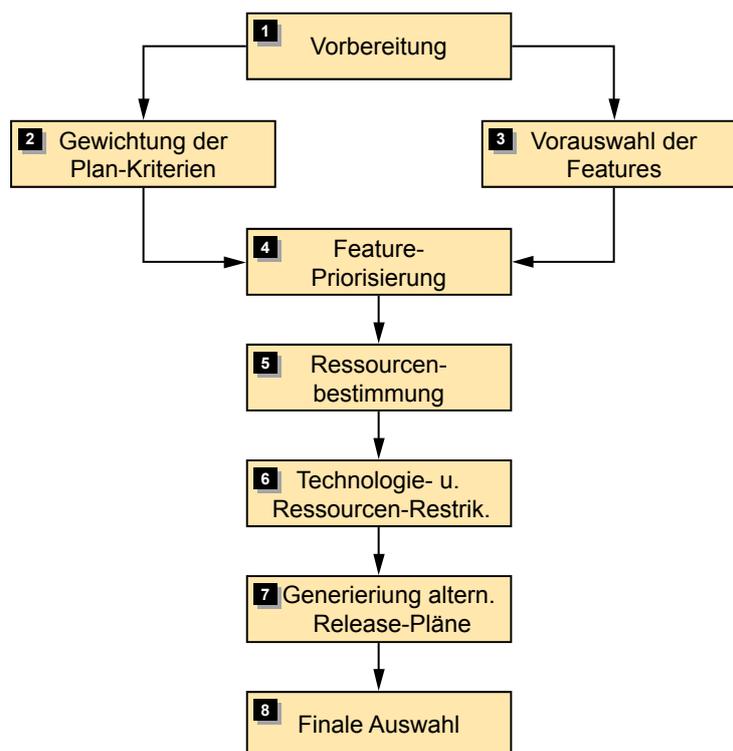


Bild A-7: Vorgehensmodell EVOLVE II nach RUHE [Ruh10, S. 110]

Schritt 1 – Vorbereitung: Im ersten Schritt werden, basierend auf der Unternehmensstrategie und den verfügbaren Projektinformationen, Ressourcenfragen geklärt und Schlüsselparameter für die darauf aufbauende Planung definiert. Ergebnis dieser Phase sind die Kriterien für die Planung, die Anzahl der Releases und der Planungsumfang. Ebenfalls werden Stakeholder spezifiziert und hinsichtlich ihrer Bedeutung gewichtet [Ruh10, S. 110f.].

Schritt 2 – Gewichtung der Plan-Kriterien: In diesem Schritt werden die zuvor definierten Kriterien ihrer Bedeutung nach gewichtet. Je nach Komplexität des Sachverhalts geschieht dies durch den Produktmanager oder als Ergebnis einer Gruppendiskussion [Ruh10, S. 111f.].

Schritt 3 – Vorauswahl der Features: Um Komplexität zu reduzieren, fließen nicht alle Features in den Planungsprozess ein. In diesem Schritt erfolgt daher eine Vorauswahl der Features, die in die Planung einfließen sollen [Ruh10, S. 112].

Schritt 4 – Feature-Priorisierung: Zur Auswahl der im Release zu realisierenden Features müssen diese durch die Stakeholder priorisiert werden. Dazu schlägt RUHE die Multi-Score-Methode vor [Ruh10, S. 47]. Die Bewertung der Features findet dabei anhand der in Schritt 2 definierten Kriterien statt [Ruh10, S. 112f.].

Schritt 5 – Ressourcenbestimmung: In diesem Schritt wird ermittelt, wie viele Ressourcen für die Umsetzung eines Features erforderlich sind [Ruh10, S. 114].

Schritt 6 – Analyse von Technologie- und Ressourcen-Restriktionen: Technologische Restriktionen ergeben sich aus Abhängigkeiten zwischen Features. Genauso gilt es, Restriktionen hinsichtlich der verfügbaren Ressourcen zu berücksichtigen. In diesem Schritt werden diese Restriktionen formalisiert und als Rahmenbedingungen für die Generierung alternativer Release-Pläne herangezogen [Ruh10, S. 115].

Schritt 7 – Generierung alternativer Release-Pläne: Über computergestützte Algorithmen wird ein Portfolio von bis zu fünf alternativen Release-Plänen generiert. Diese beschreiben unter Berücksichtigung der Feature-Priorität und der Restriktionen mögliche Bündel von Features sowie den Zeitpunkt der Umsetzung [Ruh10, S. 115f.].

Schritt 8 – Finale Auswahl: Unter Berücksichtigung optionaler Analysen (z.B. What-If-Analyse) wird in diesem Schritt der finale Plan ausgewählt [Ruh10, S. 119f.].

A3 Ergänzungen zur Systematik

A3.1 Kriterien zur Klassifizierung von technischen Änderungen

	Kriterium	Beschreibung	Beispielhafte Ausprägungen			
1	Dringlichkeit	Kriterien beschreiben die Notwendigkeit, alle benötigten Ressourcen für eine Änderung sofort bereitzustellen	Sehr hoch	Hoch	Mittel	Gering
1.1	Wirtschaftlicher Schaden	Dringlichkeit bedingt durch wirtschaftliche Konsequenzen der Nicht-Änderung	Möglicher Schaden bis 500.000 €	Möglicher Schaden bis 100.000 €	Möglicher Schaden bis 50.000 €	
1.2	Wirtschaftlicher Nutzen	Dringlichkeit bedingt durch Rationalisierungs- oder Kosteneinsparpotential bei Änderungsumsetzung	Einsparung größer 500.000 €	Einsparung größer 100.000 €	Einsparung größer 50.000 €	Einsparung kleiner 50.000 €
1.3	Produktfehler	Dringlichkeit bedingt durch Konsequenzen des Produktfehlers bei Nicht-Änderung	Gefahr für Personen oder Sachen	Funktionsfähigkeit nicht gewährleistet	Qualitätsbeeinträchtigung bei Kunden	Durch Kunden nicht wahrnehmbar
1.4	Imageverlust	Dringlichkeit bedingt durch Imageverlust bei Nicht-Änderung	Signifikanter Imageverlust	Mittlerer Imageverlust	Minimaler Imageverlust	Keine Auswirkung auf das Image
1.5	Gesetzgebung	Dringlichkeit bedingt durch gesetzliche Bestimmungen	Sofortige Adaption	Adaption bis 2017	Adaption bis 2020	Adaption bis 2025
1.6	Wettbewerb	Dringlichkeit bedingt durch Aktivitäten des Wettbewerbs	Wettbewerb bereits am Markt	Durch Wettbewerber angekündigt	Noch nicht am Markt absehbar	
2	Wichtigkeit	Kriterien beschreiben das Nutzenpotential der Änderung aus Sicht des Unternehmens	Sehr hoch	Hoch	Mittel	Gering
2.1	Wirtschaftlicher Nutzen	Nutzen für das Unternehmen bedingt durch Rationalisierungs- oder Kosteneinsparpotential bei Änderungsumsetzung	Sehr hoch	Hoch	Mittel	Gering
2.2	Kundenanforderung	Nutzen durch verbesserte Erfüllung der Kundenanforderung	Ist bzw. wird Begeisterungsanforderung	Ist bzw. wird Leistungsanforderung	Ist bzw. wird Basisanforderung	
2.3	Imageverbesserung	Nutzen durch Imageverbesserung bei Änderung	Signifikante Imageverbesserung	Mittlere Imageverbesserung	Geringe Imageverbesserung	Keine Auswirkung auf das Image
2.4	Komfortverbesserung	Die Änderung fördert den Komfort	Sehr hoch	Hoch	Mittel	Gering
2.5	Sicherheitsverbesserung	Die Änderung fördert die Sicherheit				
2.6	Designverbesserung	Die Änderung fördert das optische Erscheinungsbild des Produkts				
2.7	Funktionsverbesserung	Die Änderung bewirkt eine verbesserte Performance des Produkts				
3	Änderungskomplexität	Kriterien beschreiben die Komplexität der Änderung aus technischer Sicht	Sehr hoch	Hoch	Mittel	Gering
3.1	Änderungsauswirkung	Intensität der Änderung an den betroffenen Systemelementen	Vollständige Überarbeitung/Neuentwicklung	Umfassende Überarbeitung	Viele, einfache Anpassungen	Wenige, einfache Anpassungen
3.2	Änderungsfortpflanzung	Art der Änderungsfortpflanzung	Architektonische Änderung durch neues Modul	Modulübergreifende Änderung	Komponentenübergreifende Änderung	Lokale Änderung an einzelnen Änderungen

	Kriterium	Beschreibung	Beispielhafte Ausprägungen			
3.3	Anzahl betroffener Systemelemente	Alternative Bewertung der Änderungsfortpflanzung	Mehr als 5 Systemelemente	2 bis 5 Systemelemente	1 Systemelement	
3.4	Interdisziplinarität	Die Anzahl der an der Änderung mitwirkenden Fachdisziplinen	Mehr als 2 Disziplinen	2 Disziplinen	Disziplinspezifisch	
3.5	Betroffene Varianten	Anzahl und Änderungsgrad betroffener Produktvarianten	Umfangreiche, individuelle Anpassung	Kleinere, individuelle Anpassungen	Wenige betroffene Gleichteile	Keine Varianten betroffen
3.6	Auswirkung auf Produktion	Aufwand der Änderungsumsetzung aus Sicht der Produktion	Neue Produktionstechnologie erforderlich	Umfangreiche Anpassung des Fertigungs- bzw. Montageprozesses	Geringfügige Anpassung des Fertigungs- bzw. Montageprozesses	Keine Anpassungen erforderlich
4	Risiko	Beschreibt das mit der Änderungsumsetzung verbundene Risiko	Sehr hoch	Hoch	Mittel	Gering
4.1	Wirtschaftliches Risiko	Potentieller wirtschaftlicher Schaden	Sehr hoch	Hoch	Mittel	Gering
4.2	Technisches Risiko	Unsicherheit in der technischen Umsetzung der Änderung	Know-how nicht vorhanden, Versuche notwendig	Kompetenzen vorhanden, Versuche notwendig	Kompetenz vorhanden, keine Versuche notwendig	
4.3	Neuigkeit	Neuheitsgrad des erforderlichen Lösungskonzepts zur Umsetzung der Änderung	Ziel und Lösungsweg unbekannt	Ziel oder Lösungsweg unbekannt	Ziel und Lösungsweg unbekannt	
5	Weitere Klassifikationsmerkmale					
5.1	Modulzugehörigkeit	Beschreibt die von einer Änderung betroffenen Module	Keine Zugehörigkeit	Modul X	Modul Y	Modul Z
5.2	Gegenstand der Änderung	Beschreibt den Gegenstand der Änderung	Spezifikation/Dokumentation	Bauteil/Komponente	Material	Betriebsmittel
5.3	Baureihen-zugehörigkeit	Beschreibt die von der Änderung betroffenen Baureihen	Baureihe X	Baureihe Y	Baureihe Z	Baureihenübergreifend
5.4	Umsetzungsdauer	Dauer für die Änderungsumsetzung	> 4 Monate	1 bis 4 Monate	< 1 Monat	
5.5	Stakeholder	Auswirkung auf die Prozesse/Produkte von weiteren Stakeholdern	Kunde	Kooperationspartner 1	Kooperationspartner 2	Zulieferer X
5.6	Auslöser	Quelle der Änderungsanforderung	Kunde	Service	Marketing	Entwicklung
5.7	Technologie-reifegrad	Reifegrad der Technologie, die der Änderung zu Grunde liegt (z.B. in Technology-Readiness-Level)	TRL 9	TRL 8	...	TRL 1
5.8	Innovationsgrad	Innovationsgrad ausgehend vom Grad der Änderungsauswirkung	Radikale Innovation	Architektonische Innovation	Modulare Innovation	Inkrementelle Innovation
5.9	Zeitpunkt	Zeitpunkt zu dem die Änderungsanforderung bekannt wurde	Konzeptphase	Entwicklung	Serienvorbereitung	Serie

A3.2 Qualitätskriterien zur Bewertung von Produktplänen nach DESCHAMPS ET AL.

- 1 Segmentierung:** Fußt der Zyklusplan auf allen Produkt-/Markt-Kombinationen und nicht nur auf hauptsächlichen Produktsegmenten? Schließt er besondere Maßnahmen zur Abdeckung von Nischensegmenten mit ein, die in der Produktstrategie ausgewiesen sind?
- 2 Zeithorizont:** Erstreckt sich der Plan auf eine Zeitdauer von wenigstens zwei Lebenszyklen der Produkte? Zumindest der erste Produktzyklus sollte in allen Einzelheiten fixiert sein; der zweite kann eher eine Schätzung sein.
- 3 Art der Produktmaßnahmen:** Grenzt der Plan verschiedene Arten von Produktmaßnahmen gegeneinander ab, die von unterschiedlichen strategischen Auswirkungen und Ressourcen ausgehen?
- 4 Extern beschaffte Produkte:** Umfasst der Plan alle Produkte im Sortiment des Unternehmens, also auch extern beschaffte? Macht er Angaben zur vorläufigen Bezugsquelle und zur Wertschöpfung durch Unternehmen (Direkteinkauf/Weiterverkauf; nach technischen Daten des Unternehmens gefertigt; vom Unternehmen entwickelt etc.)?
- 5 Termin des Projektbeginns und der Produkteinführung:** Weist der Plan zumindest das Quartal aus, in dem das Produkt auf den Markt kommt, und sagt er – unter Berücksichtigung normaler Entwicklungszeiten für verschiedene Produktmaßnahmen und ihrer erwarteten Verkürzung – etwas über den ungefähren Zeitpunkt des Projektbeginns aus?
- 6 Produktmerkmale:** Beschreibt der Plan zumindest für die erste Hälfte des Planungshorizonts in groben Zügen den Inhalt jeder ins Auge gefassten Produktmaßnahme? Dazu gehört eine Grundskizze zu Konzept und Positionierung des Produkts (wenn neu) sowie zu Innovationsmerkmalen und Kostenzielen.
- 7 Reaktionen der Konkurrenz:** Geht der Plan unter Einbeziehung wahrscheinlicher Planungen der Hauptkonkurrenten auf deren erwartete Schritte ein? Spricht er Produkte an, bei denen der Zeitpunkt der Einführung im Vergleich zur Konkurrenz entscheidend ist?
- 8 Wirtschaftlichkeit des Produkts:** Beschreibt und rechtfertigt der Plan nach Möglichkeit Erwartungen zu Umsatzentwicklung, Preis, Kosten, Gewinnspanne, Investition und Kapitalertrag? Dies sollte zumindest auf die detaillierte erste Hälfte des Zeitplans für jede Produktmaßnahme zutreffen.
- 9 Entwicklungsressourcen:** Veranschlagt der Plan Entwicklungskapazitäten auf der Basis von normalen Entwicklungszeiten, so dass die Auswirkungen der einzelnen Produktmaßnahmen auf die vorhandenen Kapazitäten deutlich hervortreten? Weist er Kapazitätsmängel und Mittel zu ihrer Bewältigung aus?
- 10 Vorbereitung der Produktplanung:** Sind mit der Produktplanung ausdrücklich Manager aus Marketing sowie Forschung und Entwicklung betraut worden, die gemeinsam für die Vorbereitung, Quantifizierung und Bestätigung des Plans verantwortlich sind? In fortschrittlichen Unternehmen wird diese Aufgabe oft von einem aus der Marketingabteilung stammenden Produktplaner und einem Planer für Technologie und Entwicklung aus der F&E-Abteilung wahrgenommen.
- 11 Bestätigung:** Enthält der Prozess zwei voneinander abgegrenzte Phasen, und zwar die Ausarbeitung eines „Wunschplans“ (die zeitliche Entwicklung der Produktlinie nach Auffassung des Marketings) sowie den optimierten Produktplan, auf den sich alle Funktionen nach Abwägung sämtlicher Einflussfaktoren verständigt haben?
- 12 Genehmigung:** Wird der Produktplan einem höherrangigen, funktionsübergreifenden Managementausschuss (dem Strategieausschuss oder Produktkomitee) zur Genehmigung vorgelegt? Ist er dann bis zum nächsten angesetzten Überprüfungstermin verbindlich?
- 13 Modifizierung:** Hat die Unternehmensleitung die Voraussetzungen und den Prozess für eine Modifizierung des Produktplans zwischen zwei normalen Überprüfungsphasen genau geregelt und geht sie selbst mit gutem Beispiel strikter Plandisziplin voran?
- 14 Verknüpfung mit Geschäftsplänen:** Verwendet die Unternehmensleitung die Hauptelemente des Produktplans (Umsatzhöhe, Gewinnspannen, Investitionen, Entwicklungskapazitäten etc.) als Grundlage für den Geschäftsplan? Ist sie davon überzeugt, dass sie den einen Plan nicht abändern kann, ohne gleichzeitig den anderen zu verändern?

Bild A-8: Kriterien zur Bewertung von Produktplänen nach [DNL96, S. 167ff.]

A3.3 Beispiel für die Struktur einer Wettbewerbs-Roadmap

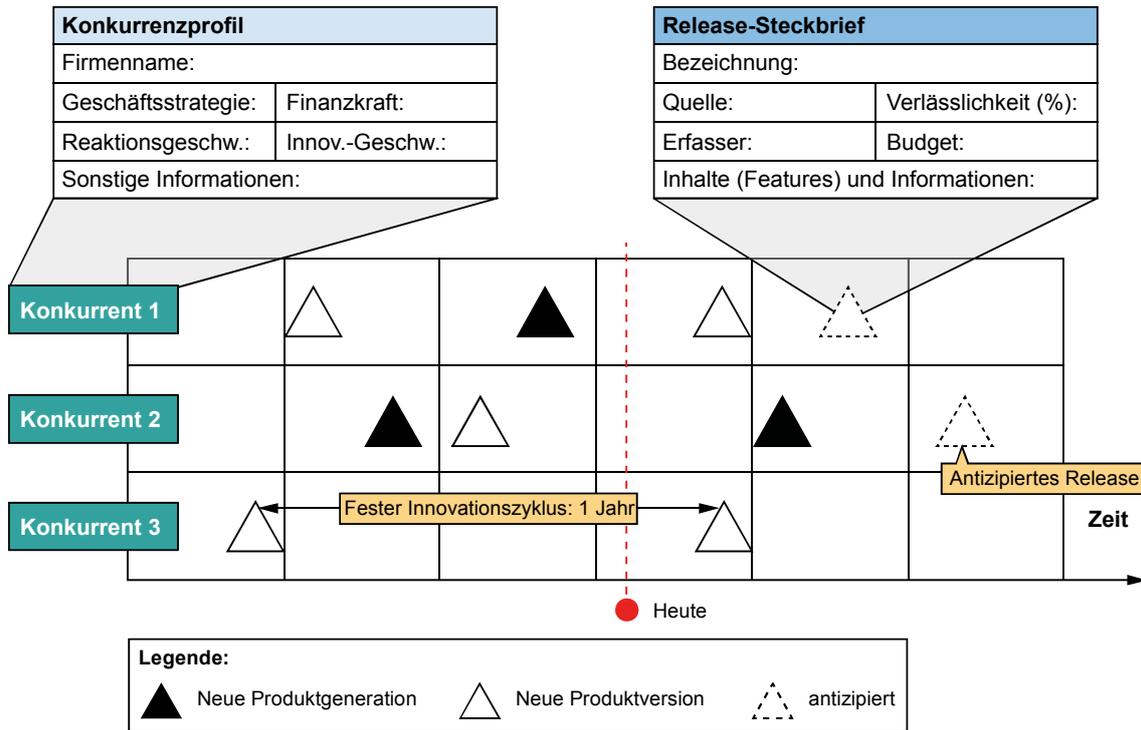


Bild A-9: Wettbewerbs-Roadmap ergänzt um Konkurrenzprofile und Release-Steckbriefe

A3.4 Alternative Release-Typen

<p>1) Sofortmaßnahmen</p> <p>Sofortmaßnahmen umfassen sämtliche Änderungen, die von der Release-Planung ausgeschlossen sind und direkt umgesetzt werden. Dies sind dringliche Änderungen z.B. aufgrund gesetzlicher Maßnahmen. Auch geringfügige Änderungen z.B. an der technischen Dokumentation können via Sofortmaßnahme umgesetzt werden.</p> <p>Häufig verwendet mit:</p> <ul style="list-style-type: none"> • Bestandteil jedes Release-Konzepts <p>Relevante Release-Strategien</p> <p><input checked="" type="checkbox"/> Modulorientiert <input checked="" type="checkbox"/> Produktorientiert <input checked="" type="checkbox"/> Flexibel</p>	<p>2) Produktgeneration</p> <p>Neue Produktgenerationen umfassen Änderungen, die sich durch einen besonders hohen Innovationsgrad auszeichnen z.B. aufgrund neuer Technologien oder einer Revolution im Design. In der Regel sind neue Produktgenerationen auch mit einer Überarbeitung der Systemarchitektur verbunden.</p> <p>Häufig verwendet mit:</p> <ul style="list-style-type: none"> • Bestandteil jedes Release-Konzepts <p>Relevante Release-Strategien</p> <p><input checked="" type="checkbox"/> Modulorientiert <input checked="" type="checkbox"/> Produktorientiert <input checked="" type="checkbox"/> Flexibel</p>
<p>3) Minor-Release</p> <p>Minor-Releases umfassen rein technische Änderungen ohne Kundenrelevanz. Die Releases werden nicht explizit vermarktet und umfassen typischerweise weniger komplexe Änderungen zur Qualitätsverbesserung und Kosteneinsparung.</p> <p>Häufig verwendet mit:</p> <ul style="list-style-type: none"> • Sofortmaßnahme • Produktgeneration • Major-Release • Software-Release <p>Relevante Release-Strategien</p> <p><input type="checkbox"/> Modulorientiert <input checked="" type="checkbox"/> Produktorientiert <input checked="" type="checkbox"/> Flexibel</p>	<p>4) Major-Release</p> <p>Major-Releases sind kundenrelevant und typischerweise Teil der Marketingstrategie. Sie umfassen signifikante Produktverbesserungen oder neue Produktfeature. Auch komplexere Änderungen werden in diesem Rahmen umgesetzt.</p> <p>Häufig verwendet mit:</p> <ul style="list-style-type: none"> • Sofortmaßnahme • Produktgeneration • Minor-Release • Software-Release <p>Relevante Release-Strategien</p> <p><input type="checkbox"/> Modulorientiert <input checked="" type="checkbox"/> Produktorientiert <input checked="" type="checkbox"/> Flexibel</p>
<p>5) Software-Release</p> <p>Software-Releases umfassen Änderungen, die im Rahmen eines gezielten Software-Updates ohne Auswirkung auf die Produktion umzusetzen sind. Software-Releases werden typischerweise eigenständig durch die Softwartechnik geplant und in der Regel nochmals in Minor- und Major-Releases sowie Bugfixes unterschieden.</p> <p>Häufig verwendet mit:</p> <ul style="list-style-type: none"> • Sofortmaßnahme • Produktgeneration • Minor-Release • Major-Release <p>Relevante Release-Strategien</p> <p><input checked="" type="checkbox"/> Modulorientiert <input checked="" type="checkbox"/> Produktorientiert <input checked="" type="checkbox"/> Flexibel</p>	<p>6) Produkt-Release</p> <p>Produkt-Releases fassen Änderungen zusammen, die sich auf das technische System auswirken (z.B. Änderungen am Design, des Materials, der Funktionalität). Hierzu gehören Änderungen von Mechanik, Elektronik und Software. Ausgenommen sind z.B. organisatorische Änderungen und Änderungen im Montageprozess.</p> <p>Häufig verwendet mit:</p> <ul style="list-style-type: none"> • Sofortmaßnahme • Produktgeneration • Prozess-Release • Software-Release <p>Relevante Release-Strategien</p> <p><input type="checkbox"/> Modulorientiert <input checked="" type="checkbox"/> Produktorientiert <input checked="" type="checkbox"/> Flexibel</p>

7) Prozess-Release	8) Modul-Release		
<p>Im Gegensatz zum Produkt-Release fassen Prozess-Releases Änderungen ohne direkten Bezug zum technischen System zusammen. Hierzu zählen z.B. organisatorische Änderungen, Änderungen im Fertigungs- und Montageprozess, Verpackungsänderungen sowie Prüfmitteländerungen.</p>	<p>Häufig verwendet mit:</p> <ul style="list-style-type: none"> • Sofortmaßnahme • Produktgeneration • Produkt-Release 		
<p>Modul-Releases umfassen Änderungen mit Auswirkung auf das gleiche Modul. Die Strukturierung des Produkts in Module kann dabei unter verschiedenen Gesichtspunkten erfolgen (z.B. Strukturierung nach Testbarkeit, nach Technologiezyklen und Wiederverwendbarkeit) [Ste07].</p>	<p>Häufig verwendet mit:</p> <ul style="list-style-type: none"> • Sofortmaßnahme • Produktgeneration 		
Relevante Release-Strategien <input type="checkbox"/> Modul-orientiert <input checked="" type="checkbox"/> Produkt-orientiert <input checked="" type="checkbox"/> Flexibel		Relevante Release-Strategien <input checked="" type="checkbox"/> Modul-orientiert <input type="checkbox"/> Produkt-orientiert <input type="checkbox"/> Flexibel	
9) Plattform-Release	10) Gleichteil-Release		
<p>Verfolgt das Unternehmen eine Plattformstrategie, bietet sich die Umsetzung von Änderungen in Form von Plattform-Releases an. In diesem Fall werden sämtliche Änderungen zusammengefasst, die sich direkt auf die Plattform des Produkts auswirken.</p>	<p>Häufig verwendet mit:</p> <ul style="list-style-type: none"> • Sofortmaßnahme • Produktgeneration • Modul-Release 		
<p>Gleichteil-Releases bieten sich an, wenn die Planung produktübergreifend erfolgt. In diesem Fall werden alle Änderungen zusammengefasst, die sich produktübergreifend auf mehrfach verwendete Komponenten auswirken. So kann die Synchronisation zwischen verschiedenen Produktgruppen sichergestellt werden.</p>	<p>Häufig verwendet mit:</p> <ul style="list-style-type: none"> • Sofortmaßnahme • Produktgeneration • Produkt-Release 		
Relevante Release-Strategien <input checked="" type="checkbox"/> Modul-orientiert <input type="checkbox"/> Produkt-orientiert <input type="checkbox"/> Flexibel		Relevante Release-Strategien <input type="checkbox"/> Modul-orientiert <input checked="" type="checkbox"/> Produkt-orientiert <input type="checkbox"/> Flexibel	

Das Heinz Nixdorf Institut – Interdisziplinäres Forschungszentrum für Informatik und Technik

Das Heinz Nixdorf Institut ist ein Forschungszentrum der Universität Paderborn. Es entstand 1987 aus der Initiative und mit Förderung von Heinz Nixdorf. Damit wollte er Ingenieurwissenschaften und Informatik zusammenführen, um wesentliche Impulse für neue Produkte und Dienstleistungen zu erzeugen. Dies schließt auch die Wechselwirkungen mit dem gesellschaftlichen Umfeld ein.

Die Forschungsarbeit orientiert sich an dem Programm „Dynamik, Mobilität, Vernetzung: Eine neue Schule des Entwurfs der technischen Systeme von morgen“. In der Lehre engagiert sich das Heinz Nixdorf Institut in Studiengängen der Informatik, der Ingenieurwissenschaften und der Wirtschaftswissenschaften.

Heute wirken am Heinz Nixdorf Institut neun Professoren mit insgesamt 150 Mitarbeiterinnen und Mitarbeitern. Pro Jahr promovieren hier etwa 20 Nachwuchswissenschaftlerinnen und Nachwuchswissenschaftler.

Heinz Nixdorf Institute – Interdisciplinary Research Centre for Computer Science and Technology

The Heinz Nixdorf Institute is a research centre within the University of Paderborn. It was founded in 1987 initiated and supported by Heinz Nixdorf. By doing so he wanted to create a symbiosis of computer science and engineering in order to provide critical impetus for new products and services. This includes interactions with the social environment.

Our research is aligned with the program “Dynamics, Mobility, Integration: Enroute to the technical systems of tomorrow.” In training and education the Heinz Nixdorf Institute is involved in many programs of study at the University of Paderborn. The superior goal in education and training is to communicate competencies that are critical in tomorrows economy.

Today nine Professors and 150 researchers work at the Heinz Nixdorf Institute. Per year approximately 20 young researchers receive a doctorate.

Zuletzt erschienene Bände der Verlagsschriftenreihe des Heinz Nixdorf Instituts

- Bd. 342 GAUSEMEIER, J.; GRAFE, M.; MEYER AUF DER HEIDE, F. (Hrsg.): 12. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung. Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 342, Paderborn, 2015 – ISBN 978-3-942647-61-8
- Bd. 343 GAUSEMEIER, J.; DUMITRESCU, R.; RAMMIG, F.; SCHÄFER, W.; TRÄCHTLER, A. (Hrsg.): 10. Paderborner Workshop Entwurf mechatronischer Systeme. Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 343, Paderborn, 2015 – ISBN 978-3-942647-62-5
- Bd. 344 BRÖKELMANN, J.: Systematik der virtuellen Inbetriebnahme von automatisierten Produktionssystemen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 344, Paderborn, 2015 – ISBN 978-3-942647-63-2
- Bd. 345 SHAREEF, Z.: Path Planning and Trajectory Optimization of Delta Parallel Robot. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 345, Paderborn, 2015 – ISBN 978-3-942647-64-9
- Bd. 346 VASSHOLZ, M.: Systematik zur wirtschaftlichkeitsorientierten Konzipierung Intelligenter Technischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 346, Paderborn, 2015 – ISBN 978-3-942647-65-6
- Bd. 347 GAUSEMEIER, J. (Hrsg.): Vorausschau und Technologieplanung. 11. Symposium für Vorausschau und Technologieplanung, Heinz Nixdorf Institut, 29. und 30. Oktober 2015, Berlin-Brandenburgische Akademie der Wissenschaften, Berlin, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 347, Paderborn, 2015 – ISBN 978-3-942647-66-3
- Bd. 348 HEINZEMANN, C.: Verification and Simulation of Self-Adaptive Mechatronic Systems. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 348, Paderborn, 2015 – ISBN 978-3-942647-67-0
- Bd. 349 MARKWART, P.: Analytische Herleitung der Reihenfolgeregeln zur Entzerrung hochauslastender Auftragsmerkmale. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 349, Paderborn, 2015 – ISBN 978-3-942647-68-7
- Bd. 350 RÜBBELKE, R.: Systematik zur innovationsorientierten Kompetenzplanung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 350, Paderborn, 2016 – ISBN 978-3-942647-69-4
- Bd. 351 BRENNER, C.: Szenariobasierte Synthese verteilter mechatronischer Systeme. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 351, Paderborn, 2016 – ISBN 978-3-942647-70-0
- Bd. 352 WALL, M.: Systematik zur technologieinduzierten Produkt- und Technologieplanung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 352, Paderborn, 2016 – ISBN 978-3-942647-71-7
- Bd. 353 CORD-LANDWEHR, A.: Selfish Network Creation - On Variants of Network Creation Games. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 353, Paderborn, 2016 – ISBN 978-3-942647-72-4
- Bd. 354 ANACKER, H.: Instrumentarium für einen lösungsmusterbasierten Entwurf fortgeschrittener mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 354, Paderborn, 2016 – ISBN 978-3-942647-73-1
- Bd. 355 RUDTSCH, V.: Methodik zur Bewertung von Produktionssystemen in der frühen Entwicklungsphase. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 355, Paderborn, 2016 – ISBN 978-3-942647-74-8

Zuletzt erschienene Bände der Verlagsschriftenreihe des Heinz Nixdorf Instituts

- Bd. 356 SÖLLNER, C.: Methode zur Planung eines zukunftsfähigen Produktportfolios. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 356, Paderborn, 2016 – ISBN 978-3-942647-75-5
- Bd. 357 AMSHOFF, B.: Systematik zur musterbasierten Entwicklung technologie-induzierter Geschäftsmodelle. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 357, Paderborn, 2016 – ISBN 978-3-942647-76-2
- Bd. 358 LÖFFLER, A.: Entwicklung einer modellbasierten In-the-Loop-Testumgebung für Waschautomaten. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 358, Paderborn, 2016 – ISBN 978-3-942647-77-9
- Bd. 359 LEHNER, A.: Systematik zur lösungsmusterbasierten Entwicklung von Frugal Innovations. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 359, Paderborn, 2016 – ISBN 978-3-942647-78-6
- Bd. 360 GAUSEMEIER, J. (Hrsg.): Vorausschau und Technologieplanung. 12. Symposium für Vorausschau und Technologieplanung, Heinz Nixdorf Institut, 8. und 9. Dezember 2016, Berlin-Brandenburgische Akademie der Wissenschaften, Berlin, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 360, Paderborn, 2016 – ISBN 978-3-942647-79-3
- Bd. 361 PETER, S.: Systematik zur Antizipation von Stakeholder-Reaktionen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 361, Paderborn, 2016 – ISBN 978-3-942647-80-9
- Bd. 362 ECHTERHOFF, O.: Systematik zur Erarbeitung modellbasierter Entwicklungsaufträge. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 362, Paderborn, 2016 – ISBN 978-3-942647-81-6
- Bd. 363 TSCHIRNER, C.: Rahmenwerk zur Integration des modellbasierten Systems Engineering in die Produktentstehung mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 363, Paderborn, 2016 – ISBN 978-3-942647-82-3
- Bd. 364 KNOOP, S.: Flachheitsbasierte Positionsregelungen für Parallelkinematiken am Beispiel eines hochdynamischen hydraulischen Hexapoden. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 364, Paderborn, 2016 – ISBN 978-3-942647-83-0
- Bd. 365 KLIEWE, D.: Entwurfssystematik für den präventiven Schutz Intelligenter Technischer Systeme vor Produktpiraterie. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 365, Paderborn, 2017 – ISBN 978-3-942647-84-7
- Bd. 366 IWANEK, P.: Systematik zur Steigerung der Intelligenz mechatronischer Systeme im Maschinen- und Anlagenbau. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 366, Paderborn, 2017 – ISBN 978-3-942647-85-4
- Bd. 367 SCHWEERS, C.: Adaptive Sigma-Punkte-Filter-Auslegung zur Zustands- und Parameterschätzung an Black-Box-Modellen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 367, Paderborn, 2017 – ISBN 978-3-942647-86-1
- Bd. 368 SCHIERBAUM, T.: Systematik zur Kostenbewertung im Systementwurf mechatronischer Systeme in der Technologie Molded Interconnect Devices (MID). Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 368, Paderborn, 2017 – ISBN 978-3-942647-87-8
- Bd. 369 BODDEN, E.; DRESSLER, F.; DUMITRESCU, R.; GAUSEMEIER, J.; MEYER AUF DER HEIDE, F.; SCHEYTT, C.; TRÄCHTLER, A. (Hrsg.): Intelligente technische Systeme. Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 369, Paderborn, 2017 – ISBN 978-3-942647-88-5