

**Rahmenwerk für einen selbstoptimierenden
Entwicklungsprozess fortschrittlicher
mechatronischer Systeme**

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

genehmigte
DISSERTATION

von
Dipl.-Wirt.-Ing. Sascha M. Kahl
aus *Kassel*

Tag des Kolloquiums: 20. Dezember 2012

Referent: Prof. Dr.-Ing. Jürgen Gausemeier

Korreferent: Prof. Dr.-Ing. Detmar Zimmer

Geleitwort

Das übergeordnete Ziel des von mir vertretenden Fachgebiets Produktentstehung ist die Steigerung der Innovationskraft von Industrieunternehmen. Ein Schwerpunkt der Arbeiten ist das Systems Engineering für mechatronische Systeme. Der stark gestiegene Anteil an Informations- und Kommunikationstechnik in diesen Systemen führt zu einer Steigerung der Komplexität dieser Systeme sowie von deren Entwicklungsprozess. Ziel ist, diese Komplexität mittels fachdisziplinübergreifender Entwicklungsmethoden und -werkzeuge zu beherrschen. Wesentliches Element ist der Aufbau eines ganzheitlichen Systemverständnisses bei den an der Entwicklung beteiligten Entwicklerinnen und Entwicklern. Dies umfasst das Verständnis über die Struktur und die Funktionsweise des zu entwickelnden Systems sowie über dessen projekt-, unternehmens- und industriebezogene Wechselwirkungen während des Entwicklungsprozesses. Im Ergebnis werden Abstimmungsaufwände und nachträgliche Änderungen reduziert, die Entwicklungszeit verkürzt und die Produktqualität gesteigert.

Für die fachdisziplinübergreifende Analyse und Synthese der Struktur und der Funktionsweise mechatronischer Systeme existieren bereits verschiedene modellbasierte Methoden und Werkzeuge, die derzeit ihren Weg in die industrielle Anwendung finden. Für die Handhabung der Wechselwirkungen zwischen dem zu entwickelnden System und seinem Entwicklungsumfeld existieren solche Methoden und Werkzeuge jedoch nur in Ansätzen. Es mangelt insbesondere an Mechanismen, mit denen plötzliche und unvorhergesehene Änderungen am zu entwickelnden System oder den durch das Unternehmen vorgegebenen operativen und strategischen Rahmenbedingungen strukturiert in entsprechende Anpassungen des Entwicklungsprozesses umgesetzt werden können.

Das von Herrn Kahl entwickelte Rahmenwerk stellt Strukturen und Mechanismen bereit, die zu jeder Zeit einen optimalen Entwicklungsprozess für das zu entwickelnde mechatronische System ermöglichen. Schlüsselement ist die Übertragung des Paradigmas der Selbstoptimierung auf das Management von Entwicklungsprozessen. Hierfür werden die Wechselwirkungen zwischen zu entwickelndem System, Entwicklungsprozess sowie operativen und strategischen Rahmenbedingungen des Unternehmens in einem systemischen Gesamtmodell abgebildet. Dieses wird durch Modellierungstechniken und Verfahren ergänzt, mit denen situationsspezifisch die aktuellen Entwicklungsprozessziele identifiziert und der Entwicklungsprozess entsprechend diesen Zielen unter Berücksichtigung der Struktur des zu entwickelnden Systems sowie den verfügbaren personellen und technischen Ressourcen angepasst wird. Mit seiner Dissertation leistet Herr Kahl einen wertvollen Beitrag zur Integration des modellbasierten Systems Engineering mit dem Projektmanagement und der Konstruktionswissenschaft.

Liste der veröffentlichten Teilergebnisse

- [KGD10] KAHL, S.; GAUSEMEIER, J.; DUMITRESCU, R.: Interactive Visualization of Development Processes in Mechatronic Engineering. In: Proceedings of the 1st International Conference on Modelling and Management of Engineering Processes (MMEP2010). Cambridge, UK, 19-20 July, 2010
- [GFD+09] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Specification Technique for the Description of Self-Optimizing Mechatronic Systems. In: Research in Engineering Design, Vol. 20, Number 4, November 2009, Springer-Verlag, London, 2009
- [GSD+09] GAUSEMEIER, J.; STEFFEN, D.; DONOTH, J.; KAHL, S.: Conceptual Design of Modularized Advanced Mechatronic Systems. In: Proceedings 17th International Conference On Engineering Design, 24-27 August, Stanford, USA, 2009
- [GSG+09] GAUSEMEIER, J.; SCHÄFER, W.; GREENYER, J.; KAHL, S.; POOK, S.; RIEKE, J.: Consistency Management between Domain-Spanning Specification and Domain-Specific Models. In: Proceedings of 17th International Conference on Engineering Design, 24-27 August, Stanford, USA, 2009
- [GRD+09] GEIGER, C.; RECKTER, H.; DUMITRESCU, R.; KAHL, S.; BERSSENBRÜGGE, J.: A Zoomable User Interface for Presenting Hierarchical Diagrams on Large Screens. 13th International Conference on Human-Computer Interaction, 19-24 Juli, San Diego, 2009

Zusammenfassung

Die derzeitig angestrebte Steigerung der Leistungsfähigkeit fortschrittlicher mechatronischer Systeme führt zu einer steigenden Dynamik und Komplexität von deren Entwicklungsprozess. Dieser kann nicht mehr beherrscht und das Leistungspotential fortschrittlicher mechatronischer Systeme nicht voll ausgeschöpft werden.

Zur Auflösung dieses Konflikts wird in der vorliegenden Arbeit ein *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* entwickelt. Es überträgt das Paradigma der Selbstoptimierung auf die Planung, Ausführung, Überwachung und Steuerung des Entwicklungsprozesses und macht diesen wieder beherrschbar. Das Rahmenwerk ist dreistufig aufgebaut. Der erste Bestandteil ist die *theoretische Übertragung des Paradigmas der Selbstoptimierung* auf den Entwicklungsprozess. Der zweite sind *Werkzeuge zur Umsetzung* des selbstoptimierenden Entwicklungsprozesses. Den dritten Bestandteil bildet ein *Vorgehensmodell zur Initialisierung und Nachbereitung* selbstoptimierender Entwicklungsprozesse mit dem das Rahmenwerk für ein konkretes Entwicklungsprojekt konfiguriert wird.

Die abschließende Verifikation des Rahmenwerks anhand des Demonstrators RailCab zeigt, dass es die in es gesetzten Erwartungen vollumfänglich erfüllt.

Summary

The currently postulated increase of performance of advanced mechatronic systems leads to an increased dynamic and complexity of their development process. The development Process cannot be managed anymore and thus it is not possible to make use of the whole capability of advanced mechatronic systems.

The work in hand presents a *framework for a self-optimizing development process of advanced mechatronic systems* to solve this conflict. It adapts the paradigm of self-optimization to the planning, execution, monitoring, and control of the development process and makes it again manageable. The framework consists of three components. First it performs the *theoretical transfer of the paradigm of self-optimization*. Secondly it offers *tools for the realization* of a self-optimizing development process. The *procedure model for initialization and postprocessing* is the third component. It is used, to configure the framework for a specific development project.

The framework is verified by the application example RailCab. The verification shows, that the framework completely satisfies all requirements.

Inhaltsverzeichnis		Seite
1	Einleitung.....	5
1.1	Problematik	5
1.2	Zielsetzung	6
1.3	Vorgehensweise	7
2	Problemanalyse	9
2.1	Begriffsdefinition und Einordnung der Arbeit	9
2.2	Fortschrittliche mechatronische Systeme	10
2.2.1	Mechatronische Systeme	10
2.2.2	Architektur mechatronischer Systeme	12
2.2.3	Leistungsstufen mechatronischer Systeme	14
2.3	Das Paradigma der Selbstoptimierung.....	17
2.3.1	Aspekte selbstoptimierender Systeme	17
2.3.2	Der Selbstoptimierungsprozess.....	20
2.3.3	Das Operator-Controller-Modul	23
2.3.4	Beispiele selbstoptimierender Systeme	24
2.4	Entwicklungsprozess mechatronischer Systeme.....	26
2.4.1	Rahmenbedingungen des Entwicklungsprozesses	28
2.4.2	Inhärente Eigenschaften des Entwicklungsprozesses.....	29
2.4.2.1	Menschmotivierte Eigenschaften.....	29
2.4.2.2	Strukturelle Eigenschaften.....	32
2.4.3	Strategien zur effizienten Entwicklung mechatronischer Systeme.....	39
2.5	Projektmanagement	40
2.5.1	Aufgabenkomplexe des Projektmanagements	41
2.5.2	Wissensgebiete des Projektmanagements.....	43
2.6	Problemabgrenzung	48
2.7	Anforderungen an ein Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess	51
3	Stand der Technik.....	53
3.1	Systemische Entwicklungsprozessmodelle	53
3.1.1	Allgemeine Technologie nach ROPOHL	53
3.1.2	Systemtechnik nach PATZAK	55
3.1.3	ZOPH-Modell.....	56

3.1.4	Integrierte Produkterstellung nach EHRENSPIEL.....	58
3.1.5	iPeM.....	59
3.2	Modellierungstechniken.....	61
3.2.1	Modellierungstechniken für Prozesse.....	62
3.2.1.1	ARIS.....	62
3.2.1.2	OMEGA.....	63
3.2.1.3	SADT.....	65
3.2.1.4	IDEF.....	66
3.2.1.5	Netzplantechnik.....	68
3.2.2	Sprachen zur Beschreibung des Systemmodells mechatronischer Systeme.....	70
3.2.2.1	OMG SysML™.....	70
3.2.2.2	CONSENS.....	72
3.2.2.3	Strukturmodellierung nach RIEPE.....	74
3.2.2.4	Modelica®.....	75
3.3	Dynamische und adaptive Planungs- und/oder Steuerungs- verfahren für Entwicklungsprozesse.....	76
3.3.1	Situative Entwicklungsaufgaben- und Methodenauswahl nach PONN.....	76
3.3.2	Methode zur flexiblen Gestaltung integrierter Entwicklungsprozesse nach BICHLMAIER.....	78
3.3.3	Adaptive Planung und Steuerung von integrierten Entwicklungs- und Planungsprozessen nach MURR.....	80
3.3.4	Verfahren zur Planung von Entwicklungsprozessen für fortgeschrittene mechatronische Systeme nach REDENIUS.....	81
3.3.5	Adaptive Entwicklungsprozessplanung nach SCHUMANN.....	84
3.3.6	Methoden zur dynamischen Planung und Steuerung von Produktenwicklungsprozessen nach DEMERS.....	85
3.3.7	Flexible Prozessunterstützung in der Produktentwicklung – Ergebnisse des Bayrischen Forschungsverbunds FORFLOW ..	86
3.3.8	Modellgestütztes Rahmenwerk zur Planung und Steuerung adaptiver Systementwicklungsprozesse nach LÉVÁRDY.....	90
3.4	Bewertung des Stands der Technik und Handlungsbedarf.....	93
4	Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess	97
4.1	Das Rahmenwerk im Überblick	97
4.2	Übertragung des Paradigmas der Selbstoptimierung auf den Entwicklungsprozess fortschrittlicher mechatronischer Systeme.....	98
4.2.1	Aspekte des selbstoptimierenden Entwicklungsprozesses fortschrittlicher mechatronischer Systeme.....	99
4.2.2	Ablauf des Selbstoptimierungsprozesses.....	102

4.2.3	Regelungssystem des selbstoptimierenden Entwicklungsprozesses	104
4.3	Umsetzung des Selbstoptimierungsprozesses	107
4.3.1	Modellierungstechniken	108
4.3.1.1	Abbildung des Systemmodells.....	108
4.3.1.2	Modellierung der Prozessbausteine	111
4.3.2	Verfahren zur Umsetzung.....	115
4.3.2.1	Verfahren zur systemmodellbasierten Prozessplanung.....	115
4.3.2.2	Verfahren zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen	122
4.3.2.3	Verfahren zur zielkonformen Planauswahl	131
4.4	Vorgehensmodell zur Initialisierung und Nachbereitung.....	133
5	Verifikation des Rahmenwerks	137
5.1	Software-Werkzeuge zur prototypischen Umsetzung des Rahmenwerks.....	137
5.1.1	Prozessplaner.....	138
5.1.2	Editoren für die Planauswahl	142
5.2	Anwendungsbeispiel RailCab	144
5.3	Durchführung der Verifikation und Ergebnisse	147
5.3.1	Aufbau des Verifikations-Setups.....	147
5.3.2	Verifikation des Selbstoptimierungsprozesses	154
5.4	Bewertung des Rahmenwerks.....	162
6	Zusammenfassung und Ausblick	165

Anhang

A1	Datenmodell der Prozessbausteine	A-1
A2	Verfahren zur Systemstrukturanalyse	A-3
A2.1	Systemstrukturanalyse – Dynamikindex von Systemelementen	A-3
A2.2	Systemstrukturanalyse – funktionsorientierte Reihenfolgebestimmung fachdisziplinübergreifender Integrations- und Validierungsaktivitäten.....	A-4
A3	Prozessbausteine für den Entwicklungsprozess fortschrittlicher mechatronischer Systeme	A-7

A4 Situationsspezifische Auswahl und Gewichtung von
Entwicklungsprozesszielen A-17

1 Einleitung

Die vorliegende Arbeit ist im Rahmen des Sonderforschungsbereichs (SFB) 614 „Selbstoptimierende Systeme des Maschinenbaus“¹ der Universität Paderborn entstanden. Gegenstand des SFB 614 sind maschinenbauliche Systeme, die ihr Verhalten in Eigenregie bestmöglich an ggf. geänderte Betriebs- und Umfeldbedingungen anpassen. Die Arbeit ordnet sich in den Bereich Entwurfsmethodik ein und leistet einen Beitrag zu der im SFB 614 adressierten „Neuen Schule des Entwurfs intelligenter technischer Systeme“. Gegenstand der Arbeit ist ein *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme*.

1.1 Problematik

Die Erzeugnisse des modernen Maschinenbaus und verwandter Branchen, wie z.B. der Automobilindustrie oder der Medizintechnik, durchdringen unseren Alltag mehr denn je und sind aus diesem nicht mehr wegzudenken. Unabhängig davon, ob als Konsum- oder Industriegüter beruhen sie heute vielfach auf dem synergetischen Zusammenwirken der Fachdisziplinen Mechanik, Elektrik/Elektronik, Regelungs- und Softwaretechnik. Der Begriff Mechatronik bringt dies zum Ausdruck. Mechatronische Systeme bringen Wettbewerbsvorteile, indem sie die Verbesserung der Funktionalität und des Betriebsverhaltens sowie die Reduktion von Kosten, Volumen und Gewicht technischer Erzeugnisse ermöglichen. Ferner können rundweg neue Funktionen in mechatronischen Systemen realisiert werden. Sie sind bspw. in der Lage zielgerichtet, innerhalb gewisser Grenzen, autonom und flexibel auf sich ändernde Betriebssituationen zu reagieren. [Ise08, S. 21ff.], [Möh04, S. 6ff.]

Den wachsenden Anforderungen und der rasanten Entwicklung der Informationstechnik Rechnung tragend, steigt die Leistungsfähigkeit mechatronischer Systeme permanent an. Ihr Verhalten bekommt zunehmend agierenden Charakter. Sie sind adaptiver, robuster und benutzerfreundlicher als herkömmliche mechatronische Systeme und erschließen neue Nutzenpotentiale; bis hin zu „teilintelligenten“ Verhaltensweisen. Diese Art mechatronischer Systeme wird als *fortschrittliche mechatronische Systeme* bezeichnet.

Den Vorteilen mechatronischer Systeme steht eine erhöhte Dynamik und Komplexität von deren Entwicklungsprozess gegenüber. Er hat zum einen eine höhere *inhärente* Dynamik und Komplexität als derjenige einer fachdisziplinspezifischen Entwicklung, da bei der Entwicklung mechatronischer Systeme Entwickler² verschiedener Fachdiszipli-

¹ gefördert durch die Deutsche Forschungsgemeinschaft (DFG)

² In der vorliegenden Arbeit wird ausschließlich aus Gründen der einfacheren Lesbarkeit die maskuline Form verwendet. Wird beispielsweise von Entwicklern gesprochen, sind selbstredend auch Entwicklerinnen gemeint.

nen zusammenarbeiten und verschiedene Technologien zusammengebracht werden müssen. Dadurch erfordert die Durchführung des Entwicklungsprozesses einen erhöhten Kooperations-, Kommunikations-, und Koordinationsaufwand [KFG07, S. 61]. Hinzu kommt die zunehmende *fremdbestimmte* Dynamik und Komplexität. Sie ergibt sich aus den direkten und indirekten Effekten des Verlangens verschiedener Stakeholder nach immer kürzeren Produktlebenszyklen, einer steigenden Anzahl von bereitzustellenden Varianten und dem Wunsch nach individualisierten Erzeugnissen [GW11, S. 27].

Die Realisierung fortschrittlicher mechatronischer Systeme ist von einem nochmaligen Sprung der inhärenten und fremdbestimmten Dynamik und Komplexität des Entwicklungsprozesses begleitet, deren Handhabung die Prozessverantwortlichen und die involvierten Entwickler zusehends überfordert. Um dennoch erfolgreich fortschrittliche mechatronische Systeme zu entwickeln, bedarf es neuer bzw. einer Erweiterung der bestehenden Management-Mechanismen. Dynamik und Komplexität des Entwicklungsprozesses müssen beherrschbar gemacht und ein weiterer Anstieg der Kontroll- und Koordinationsaufwände vermieden werden [Sch05, S. 3]. Eine Möglichkeit dies zu tun ist die Übertragung der Mechanismen zur Steigerung der technischen Leistungsfähigkeit mechatronischer Systeme auf das Management von deren Entwicklungsprozessen. Vor diesem Hintergrund wird in der vorliegenden Arbeit das Paradigma der Selbstoptimierung auf die Planung, Ausführung sowie Überwachung und Steuerung des Entwicklungsprozesses fortschrittlicher mechatronischer Systeme übertragen.

1.2 Zielsetzung

Ziel der vorliegenden Arbeit ist ein *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme*. Es soll das Paradigma der Selbstoptimierung auf die Planung, Ausführung sowie Überwachung und Steuerung des Entwicklungsprozesses übertragen und so dessen Dynamik und Komplexität beherrschbar machen.

Das Rahmenwerk folgt einem dreistufigen Aufbau. Zuerst hat die theoretische *Übertragung des Paradigmas der Selbstoptimierung* auf den Entwicklungsprozess fortschrittlicher mechatronischer Systeme zu erfolgen. Im nächsten Schritt müssen *Werkzeuge zur Umsetzung* eines selbstoptimierenden Entwicklungsprozesses bereitgestellt werden. Diese sollen die bestehenden Methoden und Werkzeuge des Entwicklungsprozessmanagements ergänzen bzw. erweitern. Abschließend bedarf es eines *Vorgehensmodells zur Initialisierung und Nachbereitung* selbstoptimierender Entwicklungsprozesse, mit dem die Bestandteile des Rahmenwerks auf ein konkretes Entwicklungsprojekt innerhalb eines Unternehmens angewendet werden können.

1.3 Vorgehensweise

Die im Kapitel 1.1 kurz umrissene Problematik wird in der Problemanalyse in **Kapitel 2** mit dem Ziel vertieft, dedizierte Anforderungen an ein Rahmenwerk für einen *selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* abzuleiten. Hierzu werden zunächst die für die vorliegende Arbeit relevanten Begriffe und Forschungsfelder abgegrenzt. Anschließend werden die Charakteristika und Leistungsstufen mechatronischer Systeme beschrieben. Da das Paradigma der Selbstoptimierung für die vorliegende Arbeit von besonderer Relevanz ist, wird es in einem eigenen Abschnitt umfassend erläutert. Im Anschluss werden die Eigenschaften und Herausforderung der Entwicklung mechatronischer Systeme analysiert. Aufbauend auf dieser Analyse werden die Aufgabenkomplexe des Projektmanagements als Mittel zur erfolgreichen Bewältigung der identifizierten Herausforderungen untersucht. Abschließend werden die Anforderungen an das in der vorliegenden Arbeit adressierte Rahmenwerk abgeleitet.

Die detaillierte Analyse des Stands der Technik bezüglich des in der vorliegenden Arbeit adressierten Rahmenwerks bzw. einzelner seiner Bestandteile ist Gegenstand des **Kapitels 3**. Zunächst werden systemische Entwicklungsprozessmodelle auf ihre Eignung hin untersucht, als Grundlage für das adressierte Rahmenwerk zu dienen. Anschließend werden Modellierungstechniken für die Beschreibung von Entwicklungsprozessen sowie fachdisziplinübergreifenden Systemmodellen mechatronischer Systeme analysiert. Danach folgt die umfassende Darstellung von dynamischen und adaptiven Planungs- und/oder Steuerungsverfahren für Entwicklungsprozesse mechatronischer Systeme. Zum Abschluss wird nachgewiesen, dass keines der im Stand der Technik untersuchten Modelle und Verfahren den in Kapitel 2 abgeleiteten Anforderungen gerecht wird und folglich Handlungsbedarf für ein *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* besteht.

Die Entwicklung des Rahmenwerks erfolgt in **Kapitel 4**. Zunächst werden die bisher für technische Systeme definierten theoretischen Grundlagen des Paradigmas der Selbstoptimierung auf den Entwicklungsprozess übertragen. Den zweiten Hauptbestandteil des Rahmenwerks bilden die Werkzeuge zur Umsetzung des selbstoptimierenden Entwicklungsprozesses. Sie setzen die theoretischen Grundlagen praktisch um und ermöglichen deren praktische Anwendung. Den dritten Bestandteil des Rahmenwerks bildet das Vorgehensmodell zur Initialisierung und Nachbereitung selbstoptimierender Entwicklungsprozesse. Mit diesem Vorgehensmodell werden das Rahmenwerk für ein konkretes Entwicklungsprojekt eines Unternehmens konfiguriert und nach Beendigung des Entwicklungsprojekts die gesammelten Erfahrungswerte für künftige Projekte aufbereitet.

Kapitel 5 dient der Verifikation des Rahmenwerks und seiner Bestandteile. Zunächst werden die prototypisch entwickelten Software-Werkzeuge zur Umsetzung des Rahmenwerks vorgestellt. Die anschließende Verifikation erfolgt anhand des neuartigen Schienenverkehrssystems RailCab. Das Anwendungsbeispiel eignet sich auf Grund sei-

ner hohen technischen Komplexität und der daraus resultierenden hohen Komplexität seines Entwicklungsprozesses sehr gut für die Verifikation des Rahmenwerks.

Eine Zusammenfassung der vorliegenden Arbeit und ein Ausblick auf zukünftige Arbeiten werden in **Kapitel 6** gegeben. Inhalt des **Anhangs** sind ergänzende Informationen sowie detailliertere Darstellungen der Bestandteile des in dieser Arbeit entwickelten Rahmenwerks.

2 Problemanalyse

In diesem Kapitel werden auf Basis einer detaillierten Problemanalyse die Anforderungen an das in der vorliegenden Arbeit adressierte *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* abgeleitet. Zunächst findet eine Abgrenzung der relevanten Begriffe und Forschungsfelder statt (Kapitel 2.1). Anschließend werden die Charakteristika und Leistungsstufen mechatronischer Systeme beschrieben (Kapitel 2.2). Die umfassende Darstellung des Paradigmas der Selbstoptimierung ist Gegenstand des Kapitels 2.3. Kapitel 2.4 stellt die Herausforderung bei der Entwicklung mechatronischer Systeme dar. Ihnen werden in Kapitel 2.5 die Aufgabenkomplexe des Projektmanagements als Mittel zur erfolgreichen Bewältigung der identifizierten Herausforderungen gegenübergestellt. Abschließend werden aufbauend auf den bis dato gemachten Ausführungen die Anforderungen an das in der vorliegenden Arbeit adressierte Rahmenwerk abgeleitet (Kapitel 2.7).

2.1 Begriffsdefinition und Einordnung der Arbeit

Gegenstand dieser Arbeit ist ein *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme*. Unter einem **Rahmenwerk** wird im Kontext dieser Arbeit ein Ordnungsschema verstanden, in dem einzelne Elemente zur Lösung einer bestimmten Aufgabe enthalten und entsprechend ihrer Eigenschaften und Beziehungen zueinander angeordnet werden. Es ist damit von dem in der Softwaretechnik mit „Rahmenwerk“ oder der englischen Übersetzung „Framework“ bezeichneten Menge aus Editoren, Dokumenten, Compilern, Bibliotheken und Entwicklungswerkzeugen zu unterscheiden [FH11, S. 345].

Der **Entwicklungsprozess** ist die Ausführung von ingenieurmäßigen Tätigkeiten unter Zuhilfenahme von Wissen, weiteren Informationen sowie personellen und materiellen Ressourcen zur Überführung einer durch Entwicklungsziele beschriebenen Problemstellung in die vollständige Beschreibung eines technischen Systems [Hub76, S. 5], [Ehr07, S. 57ff.], [Lin07, S. 16]. Die in der vorliegenden Arbeit betrachteten Entwicklungsprozesse dienen der Entwicklung **mechatronischer Systeme**, also technischer Systeme, die auf dem synergetischen Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrik/Elektronik und Informationstechnik basieren [VDI2206, S. 14].

Ein spezifischer Entwicklungsprozess wird als **Projekt** im Sinne eines in Randbedingungen und Inhalt einmaligen Vorhabens [DIN69901-5, S. 11] gesehen. Die Initiierung, Planung, Ausführung, Überwachung und Steuerung sowie der Abschluss eines Projekts sind Gegenstand des **Projektmanagements** [DIN69901-5, S. 14].

Vor diesem Hintergrund dient das adressierte Rahmenwerk der Übertragung des Paradigmas der Selbstoptimierung auf die Planung, Ausführung sowie Überwachung und Steuerung eines Entwicklungsprozesses. Die vorliegende Arbeit schlägt somit eine Brü-

cke zwischen der **Konstruktionswissenschaft** als der Ableitung von Regeln zur Entwicklung technischer Systeme auf Basis der Analyse des Aufbaus technischer Systeme und der Beziehungen zu ihrem Umfeld [PBF+07, S. 10] und dem **Projektmanagement**.

2.2 Fortschrittliche mechatronische Systeme

Die Erzeugnisse des modernen Maschinenbaus und verwandter Branchen wie der Automobilindustrie oder der Medizintechnik sind heute vielfach mechatronische Systeme. Die Mechatronik ermöglicht die Verbesserung der Funktionalität und des Betriebsverhaltens sowie die Reduktion von Kosten, Volumen und Gewicht technischer Erzeugnisse. Ferner können rundweg neue Funktionen in technischen Systemen realisiert werden; bis hin zu „teilintelligenten“ Verhaltensweisen [Ise08, S. 21ff.], [Möh04, S. 6ff.]. Im Folgenden werden die Historie, die Arten, die Architektur und die Leistungsstufen mechatronischer Systeme, soweit sie für die vorliegende Arbeit relevant sind, vorgestellt.

2.2.1 Mechatronische Systeme

Der Begriff Mechatronik ist ein Kofferwort, das erstmals 1969 von den Japanern T. MORI und K. KIKUCHI, zwei Mitarbeitern der YASKAWA Electric Corporation, geprägt wurde [Mor69]. Der Begriff bezog sich zunächst auf die Erweiterung mechanischer Systeme um elektronische Komponenten zur Realisierung neuer Funktionen. Er setzt sich aus den Bestandteilen „Mechanism“ für Mechanik oder allgemein Maschinenbau und „Electronics“ für Elektronik oder allgemein Elektrotechnik zusammen [VDI2206, S. 9]. Über die Jahre entstand eine Vielzahl verschiedener Definitionen für die Mechatronik, die sich hinsichtlich der hinzuzuzählenden Ingenieurdisziplinen bzw. Technologien und der Berücksichtigung verschiedener Phasen des Produktlebenszyklus unterscheiden. Einen Konsens bietet die 1996 von F. HARASHIMA, M. TOMIZUKA und T. FUKUDA vorgenommene Charakterisierung [HTF96, S. 1]. Demnach zeichnet sich die Mechatronik durch das synergetische Zusammenwirken der Disziplinen Mechanik, Elektrik/Elektronik, Regelungs- und Softwaretechnik beim Entwurf und der Fertigung eines industriellen Produkts aus. Die im Original englische Definition wurde für die VDI-Richtlinie 2206 „Entwicklungsmethodik für mechatronische Systeme“ übernommen und ins Deutsche übersetzt:

„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].

Die große Bandbreite der heute realisierten mechatronischen Systeme lässt sich nach GAUSEMEIER grundsätzlich in zwei Klassen unterteilen (siehe Bild 2-1) [Gau10, S. 15]. Die erste Klasse umfasst diejenigen Systeme, die auf der räumlichen Integration von Mechanik und Elektronik beruhen (integrierte mechatronische Systeme). Zur zweiten

Klasse gehören dynamischen Mehrkörpersysteme mit kontrolliertem Bewegungsverhalten.

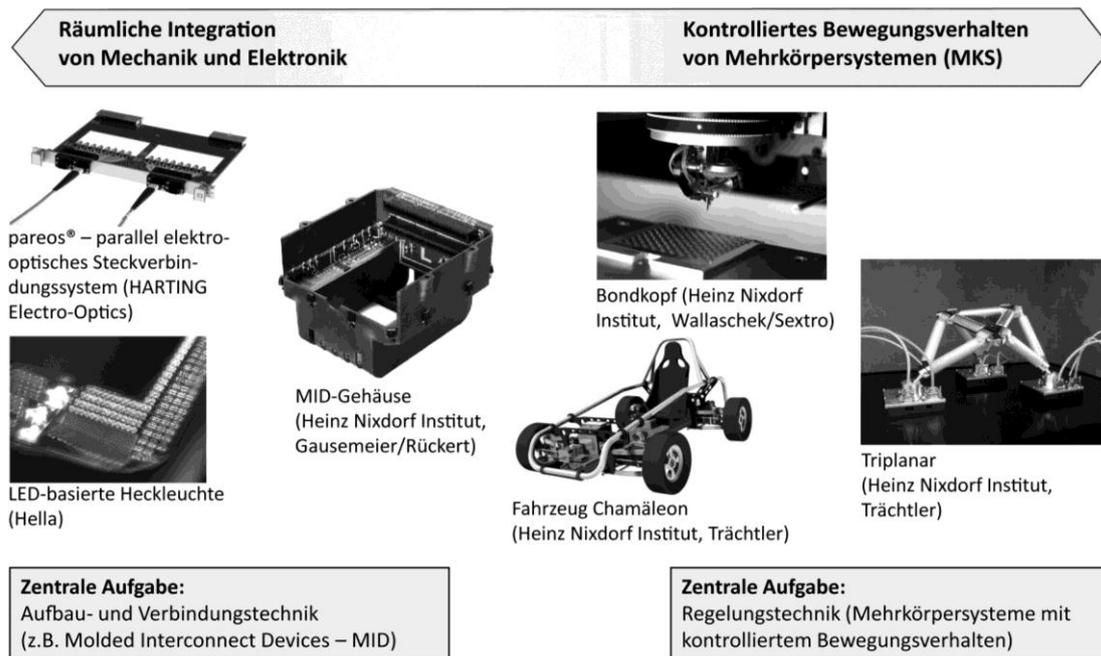


Bild 2-1 Klassen mechatronischer Systeme [Gau10, S. 15]

Ziel der **ersten Klasse** ist eine möglichst hohe Dichte mechanischer und elektrischer Funktionen auf kleinem Bauraum. Wesentliche Potentiale liegen in der Miniaturisierung und der Verringerung der Herstellkosten. Von zentraler Bedeutung ist die Aufbau- und Verbindungstechnik (AVT) wie z.B. die Technologie MID (Molded Interconnect Devices). Da sich bei dieser Klasse die eingesetzten Produkt- und Fertigungstechnologien sehr stark gegenseitig determinieren sind das Produkt und das zugehörige Produktionssystem integrativ im Wechselspiel zu entwickeln [Gau10, S. 14ff].

Die Verbesserung des Bewegungsverhaltens von Mehrkörpersystemen ist Ziel der **zweiten Klasse**. Die zentrale Rolle kommt hierbei der Regelungstechnik zu. Mit Hilfe von Sensoren werden Informationen über das System und dessen Umfeld gesammelt und basierend darauf adäquate Reaktion zur Verbesserung des Bewegungsverhaltens bestimmt, die dann von Aktoren umgesetzt werden [Gau10, S. 17].

Die beiden Klassen mechatronischer Systeme können auch in Kombination auftreten. Die integrierten Systeme der ersten Klasse sind dann oftmals Bestandteil der Systeme der zweiten Klasse; bspw. in Form intelligenter Sensoren (smart sensors) und Aktoren (smart actors) [Ise08, S. 24]. Die vorliegende Arbeit fokussiert primär die zweite Klasse mechatronischer Systeme, die die Basis für fortschrittliche mechatronische Systeme im Sinne dieser Arbeit bilden. Im Folgenden werden daher unter mechatronischen Systemen die Systeme der zweiten Klasse verstanden.

2.2.2 Architektur mechatronischer Systeme

Mechatronische Systeme bestehen basal aus vier Bestandteilen: dem Grundsystem (mechanische Struktur), Sensor(en), Aktor(en) und der Informationsverarbeitung (siehe Bild 2-2). Diese bilden gemeinsam das mechatronische Regelungssystem³, wobei in der einfachsten Ausbaustufe das Grundsystem die Regelstrecke, der Sensoren das Messglied, der Aktor die Stelleinrichtung und die Informationsverarbeitung den Regler darstellen (vgl. [DIN60050, S. 147]). Die Leistungsversorgung der vier Basisbestandteile kann systemintern oder -extern erfolgen. Sie werden grundsätzlich in einem Umfeld betrieben. Die Abläufe des Grundsystems und dessen Interaktion mit den Aktoren, Sensoren und dem Umfeld sind wert- und zeitkontinuierliche (analoge) Vorgänge. Die informationsverarbeitenden Komponenten heutiger mechatronischer Systeme arbeiten dem gegenüber in der Regel wert- und zeitdiskret (digital).

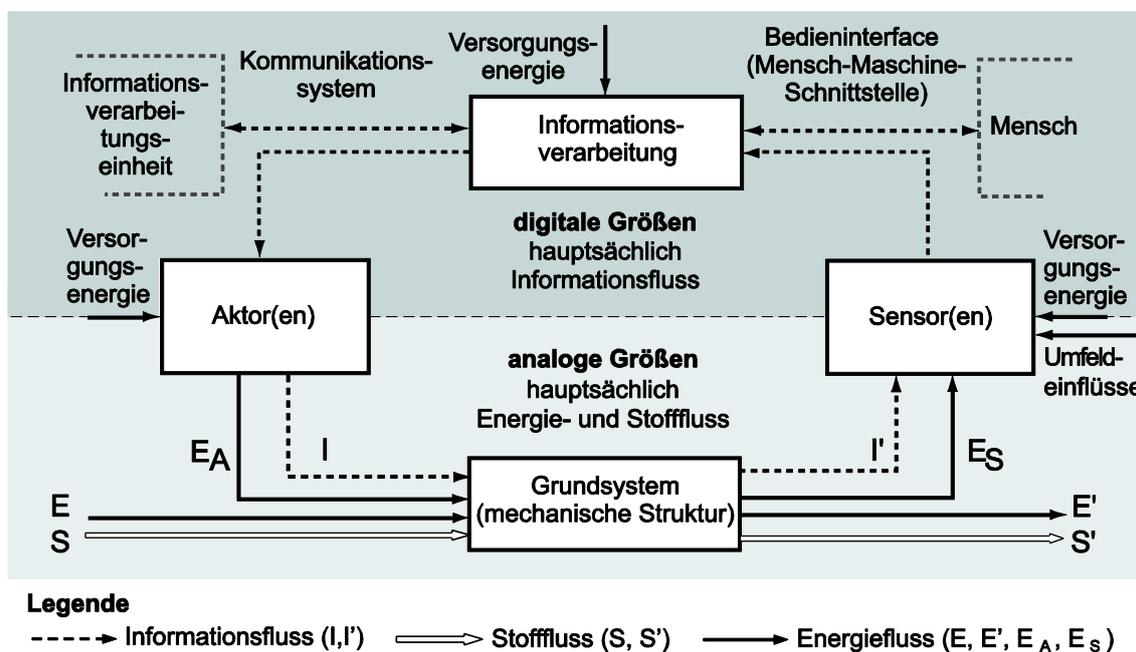


Bild 2-2 Grundstruktur eines mechatronischen Systems [in Anlehnung an GEK01, S. 32]

Das **Grundsystem** ist bspw. eine mechanische, elektromechanische, pneumatische oder hydraulische Struktur bzw. eine Kombination aus diesen. Je nach Zweck des mechatronischen Systems ist aber auch ein anderes physikalisches System vorstellbar; bspw. ein thermisches, thermodynamisches, optisches oder chemisches. Die **Sensoren** erfassen ausgewählte Zustandsgrößen⁴ des Grundsystems und bei Bedarf weitere Einflüsse aus

³ Regelungstechnische Begriffe werden in der vorliegenden Arbeit entsprechend geltender Norm DIN IEC 60050-351 [DIN 60050] verwendet.

⁴ Zustandsgrößen sind diejenigen physikalischen Größen, durch deren Wert zu einem beliebigen Zeitpunkt t_0 für alle $t > t_0$ der Ablauf eines technischen Systems eindeutig bestimmt ist, wenn für alle $t > t_0$ die Eingangsgrößen des Systems gegeben sind [Föl08, S. 389], [DIN60050, S. 7f.].

dem Umfeld des mechatronischen Systems. Sie können in Form physischer Messwertnehmer oder Software (sogenannte „Beobachter“ [Föl08, S. 501 ff.]) realisiert sein. Die mit Hilfe der Sensoren aufgenommenen Informationen dienen der **Informationsverarbeitung** zur Bestimmung derjenigen Einwirkungen, um das Grundsystem in der gewünschten Art und Weise zu beeinflussen. Hierbei können auch systemexterne Informationen über Schnittstellen zu den Informationsverarbeitungen anderer technischer Systeme oder dem Benutzer mit einbezogen werden. Die ermittelten notwendigen Einwirkungen werden durch **Aktoren** am Grundsystem umgesetzt [VDI2206, S. 14ff.].

Grundsystem, Sensoren, Aktoren und Informationsverarbeitung sind in Anlehnung an PAHL/BEITZ durch drei Arten von Flüssen miteinander verbunden [PBF+07, S. 43]:

Stoffflüsse: Sie beschreiben das Fließen von Stoffen verschiedener Aggregatzustände (feste Körper, Flüssigkeiten oder Gase); aber auch verschiedener Granularität (z.B. Staub etc.), verschiedener Reifegrade (z.B. Rohprodukt, Bauteil, geprüfter oder behandelter Gegenstand, Endprodukt) oder ähnlichem.

Energieflüsse: Hierzu gehören Flüsse unterschiedlicher Energieformen, wie z.B. mechanische, elektrische, chemische, thermische, optische oder Kernenergie. Ferner fallen darunter nicht stoffgebundene Flüsse, die die Energie eines Systemelements verändern können wie z.B. Kraft, Strom oder Wärme.

Informationsflüsse: Sie übertragen z.B. Messgrößen, Daten oder Steuerimpulse. Informationsflüsse sind in der Regel an Energieflüsse als Träger der Information gebunden.

In der Praxis weisen mechatronische Systeme die genannten Bestandteile oft mehrfach auf, die dann auf verschiedene Art und Weise miteinander vernetzt sind. Um die entstehenden komplexen Strukturen beherrschbar zu machen, schlägt LÜCKEL ein hierarchisches Ordnungsschema vor, das sich an der funktionalen Dekomposition des Bewegungsverhaltens eines mechatronischen Systems orientiert (siehe Bild 2-3) [LHL01, S. 124] [Hes06, S. 16ff.]:

- Die unterste Hierarchieebene bilden die **Mechatronischen Funktionsmodule (MFM)**. Sie beinhalten alle vier Basisbestandteile eines mechatronischen Systems. MFMs können mit anderen MFMs gleicher Hierarchieebene verbunden sein oder unterlagerte MFMs als Aktoren integrieren. Beispiele für MFM sind das Feder-/Neigemodul eines Schienenfahrzeugs oder ein aktives Federbein eines Automobils.
- Die nächst höhere Hierarchieebene ist das **Autonome Mechatronische System (AMS)**. Ein AMS entsteht durch die Vernetzung mehrere MFM innerhalb einer integrierenden Tragstruktur und stellt somit die nur einmal vorkommende, oberste Ebene eines physisch gekoppelten Systems dar. AMS sind zur Ausführung eigenständiger Handlungen innerhalb ihres Umfelds fähig. Hierfür verfügen sie über eine eigene Informationsverarbeitung und ggf. eigene Sensoren; bedienen sich aber der

Aktorik der unterlagerten MFMs. Ein autonomes Schienenfahrzeug oder ein modernes Automobil sind Beispiele für AMS.

- Die informationstechnische Kopplung mehrerer AMS führt auf oberster Hierarchieebene zu **Vernetzten Mechatronischen Systemen (VMS)**. Sie ermöglichen übergeordnete Aufgaben der Informationsverarbeitung. Hierfür muss das VMS nicht zwangsläufig eine eigene Rechen-Hardware besitzen. Die Funktionalität eines VMS kann auch auf der Hardware der integrierten AMS implementiert sein. Beispiele für VMS sind ein Konvoi von Schienenfahrzeugen mit „virtuellen“ Deichseln oder ein intelligentes Kreuzungsmanagement auf Basis miteinander kommunizierender Automobile.

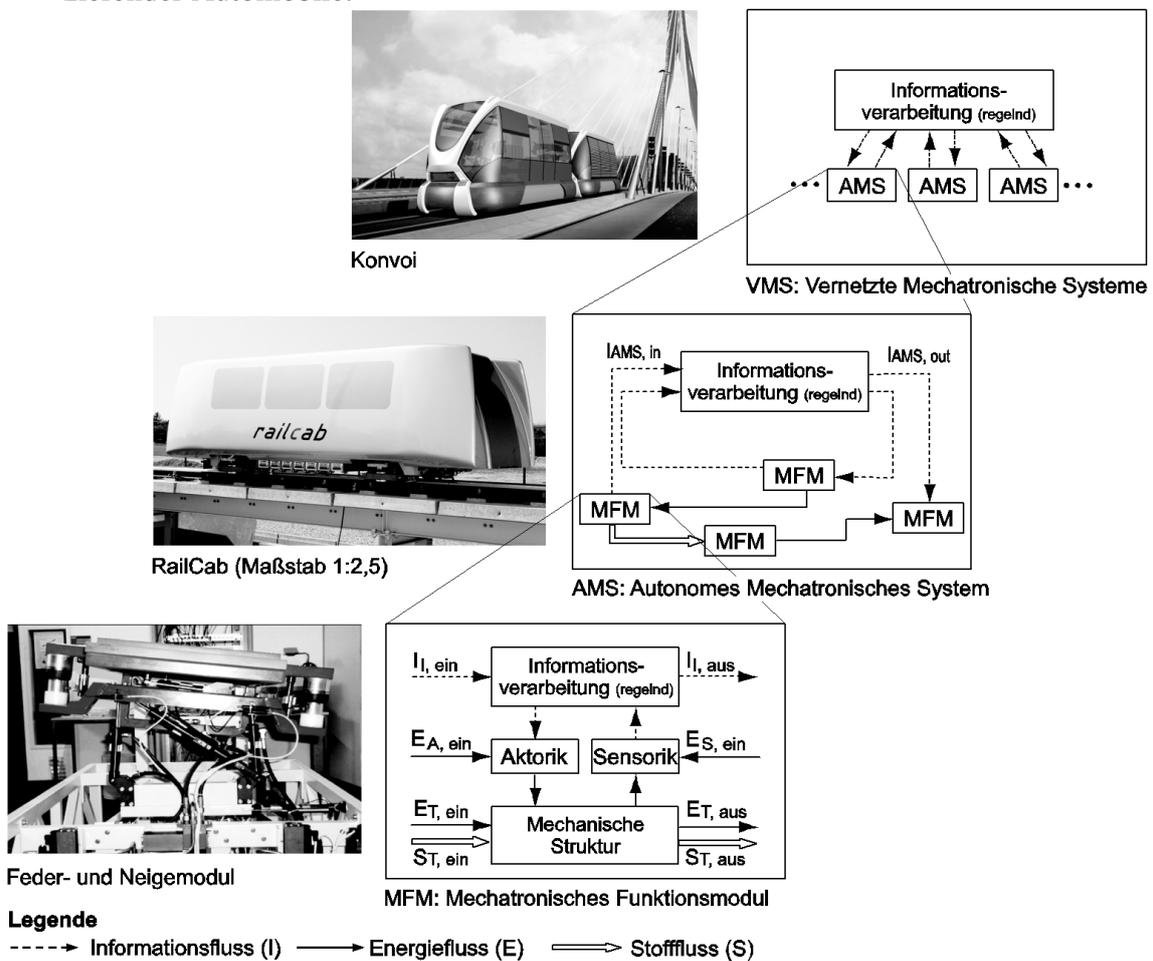


Bild 2-3 Strukturierung mechatronischer Systeme nach LÜCKEL (in Anlehnung an [Gau10, S. 20])

2.2.3 Leistungsstufen mechatronischer Systeme

Die frühen mechatronischen Systeme realisierten in der Regel einfache Regelungssysteme, um bspw. die Dämpfung der Bewegung eines technischen Systems aktiv zu beeinflussen bzw. zu verbessern. Die Systeme wurden für den Einsatz unter ganz bestimmten, während der Entwicklung bereits genau spezifizierten, Betriebsbedingungen

entwickelt und ausgelegt. Sie hatten statische Eigenschaften und zeigten reaktives Verhalten. Durch den Einsatz zunehmend verbesserter Informations- und Kommunikationstechnologie auf jeder der in Abschnitt⁵ 2.2.2 beschriebenen Hierarchieebenen sowie durch die informationstechnische Vernetzung dieser Ebenen wurden und werden mechatronische Systeme immer leistungsfähiger. Sie haben gegenwärtig einen Leistungsstand erreicht, der es ihnen erlaubt, zielgerichtet, innerhalb gegebener Grenzen, autonom und flexibel auf sich ändernde Betriebssituationen, Aufgabenstellungen oder Strategien zu reagieren bzw. sich diesen anzupassen [Ise08, S. 31].

Die bereits in den 1950er Jahren begonnene Entwicklung **adaptiver Regelungen** hat hierzu einen großen Teil beigetragen. Adaptive Regelungen haben flexibel parametrier- sowie strukturierbare Regeleinrichtungen und verfügen über die Fähigkeit, diese an veränderte Eigenschaften der Regelstrecke sowie veränderte Eingangs- respektive Störgrößen anzupassen [AW95, S. 1], [DIN60050, S. 66]. Dies ermöglicht in gewissen Grenzen die Entwicklung von Regelungen für solche mechatronischen Systeme, deren Eigenschaften und späteren Betriebsbedingungen während der Entwicklung nicht vollständig bekannt sind [KKK95, S. 1f.]. Es werden zwei Arten adaptiver Regelungen unterschieden (Bild 2-4):

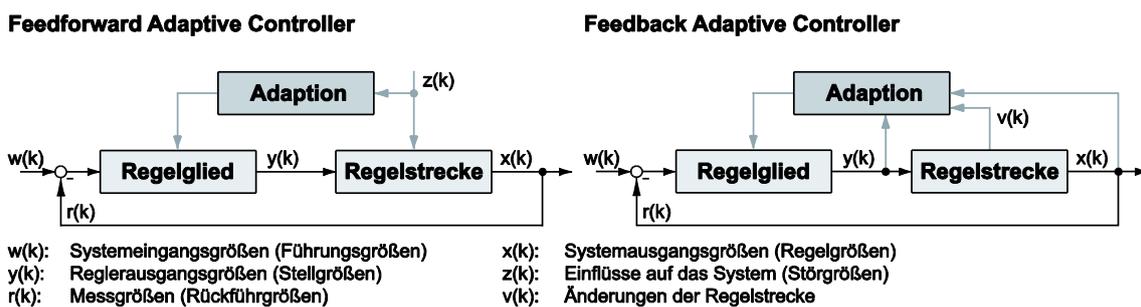


Bild 2-4 Gegenüberstellung des Prinzips von Feedforward und Feedback Adaptive Controller (in Anlehnung an [ILM92])

- Beim Prinzip des **Feedforward Adaptive Controller** muss der Einfluss der Änderung auf die Regelstrecke bekannt sein. Die Parameter oder die Struktur der Regeleinrichtung werden auf Basis dieses Wissens direkt angepasst.
- Beim Prinzip des **Feedback Adaptive Controller** wird der Einfluss einer Änderung auf die Regelstrecke und die Regelgröße zunächst durch den Adaptionsmechanismus bestimmt. Die anschließende Parameter- und Strukturanpassung erfolgt demnach indirekt.

⁵ In dieser Arbeit wird in Anlehnung an die Norm DIN 1421 nicht zwischen Kapiteln und Abschnitten unterschieden. Die Begriffe Unterkapitel und Unterabschnitt werden nicht verwendet [DIN1421].

Adaptive mechatronische Systeme sind als eine Erweiterung geregelter mechatronischer Systeme zu verstehen. Sie passen ihr Verhalten hinsichtlich während der Entwicklung festgelegter Optimierungsziele zur Laufzeit an.

Um die Leistungsfähigkeit mechatronischer Systeme weiter zu steigern, kommen immer höherwertigere regelungstechnische, informationstechnische und mathematische Verfahren zur Informationsgewinnung und -verarbeitung zum Einsatz. Die Systeme werden immer flexibler und ihr Verhalten erlangt mehr und mehr agierenden Charakter. Diese Art mechatronischer Systeme kann als **fortschrittliche mechatronische Systeme** bezeichnet werden. In unbekanntem Terrain oder komplexen Verkehrssituationen autonom handelnde Fahrzeuge sind Beispiele solcher Systeme [TMD+06], [MBB+08]. Die zu großen Teilen noch in der Erforschung befindlichen Ansätze zur Leistungssteigerung adressieren dabei unterschiedliche Hierarchieebenen mechatronischer Systeme.

Primär auf der Verbesserung und Vernetzung der Informationsverarbeitung auf MFM-bis hin zur AMS-Ebene aufbauende Ansätze sind bspw. **Selbstoptimierung**⁶ oder die **Integration kognitiver Fähigkeiten** in technische Systeme. Selbstoptimierende mechatronische Systeme sind eine Weiterentwicklung adaptiver mechatronischer Systeme und passen ihre Optimierungsziele und darauf aufbauend ihr Verhalten endogen an die jeweilige Betriebssituation an [ADG+09, S. 5ff.]. Mit der Integration kognitiver Funktionen in technische Systeme wird u.a. bezweckt, diesen Systemen die Fähigkeit zu geben, sich ihres Handelns bewusst zu werden und daraus Schlussfolgerungen zu ziehen; oder sich durch sogenannte Companion-Eigenschaften (Anpassungsfähigkeit, Verfügbarkeit, Individualität, Kooperativität, Vertrauenswürdigkeit) individuell auf den jeweiligen Nutzer einzustellen [BBW07, S. 19], [BW10, S. 335], [GRS03, S. 19 ff.].

Ein Beispiel für Ansätze, die auf VMS-Ebene angewendet werden, ist **Swarm Robotics**. Hier arbeitet eine große Anzahl von durch einfache Regeln geleiteten und mit relativ eingeschränkten physischen Fähigkeiten ausgestatteten Robotern gemeinsam an der Lösung einer Aufgabe, die ein einzelner Roboter nicht lösen kann. Dieser Ansatz ist durch das Verhalten von in sozialen Strukturen organisierten Insekten wie Ameisen oder Bienen inspiriert [SSW07, S. V]. Bei Ansätzen wie den **Cyber-Physical Systems** oder den **Intelligenten Objekten** werden mechatronische Systeme sowohl untereinander als auch mit ortsfesten informationstechnischen Infrastrukturen, wie z.B. fest installierten Videoüberwachungsanlagen oder über drahtlose Kommunikation mit dem Internet, verbunden. Hierdurch sollen u.a. der Sensorradius und die verfügbare Informationsbasis dieser Systeme erhöht werden, um deren Funktionalität zu erweitern und kontextsensitiv anzupassen [Aca09, S. 23f.], [Bro10, S. 21ff.], [HS09, S. 63ff.].

Neben den vorstehend genannten Ansätzen existieren noch weitere, deren Darstellung im Rahmen der vorliegenden Arbeit nicht geleistet werden kann. Die genannten Ansätze

⁶ Die ausführliche Erläuterung des für die vorliegende Arbeit besonders wichtigen Paradigmas der Selbstoptimierung erfolgt im nachfolgenden Kapitel 2.3.

sind daher als selektive Beispiele der umfassenden Bemühungen zur Leistungssteigerung mechatronischer Systeme zu sehen.

2.3 Das Paradigma der Selbstoptimierung

Der Begriff Selbstoptimierung wurde u.a. im Kontext adaptiver Regelungen im Jahr 1958 von KALMAN geprägt. Seiner Idee nach ist ein selbstoptimierendes System ein Regelungssystem, das abhängig vom aktuellen dynamischen Verhalten der Regelstrecke und deren Umfeld die Anforderungen an die Regeleinrichtung eigenständig bestimmt und diese entsprechend aus Standardelementen mit geeigneten dynamischen Eigenschaften zusammensetzt [Kal58]. Eine Weiterentwicklung der Idee der Selbstoptimierung für mechatronische Systeme beinhaltet die dem Sonderforschungsbereich (SFB) 614 „Selbstoptimierende Systeme des Maschinenbaus“ zugrundeliegende Definition:

„Unter Selbstoptimierung (self-optimization) eines technischen Systems wird die endogene Anpassung der Ziele des Systems auf veränderte Einflüsse und die daraus resultierende zielkonforme autonome Anpassung der Parameter und ggf. der Struktur und somit des Verhaltens dieses Systems verstanden. Damit geht Selbstoptimierung über die bekannten Regel- und Adaptionsstrategien wesentlich hinaus; Selbstoptimierung ermöglicht handlungsfähige Systeme mit inhärenter „Intelligenz“, die in der Lage sind, selbstständig auf veränderte Betriebsbedingungen zu reagieren.“ [ADG+09, S. 5].

Selbstoptimierung soll in der vorliegenden Arbeit entsprechend dieser Definition verstanden werden. Selbstoptimierung erstreckt sich demnach nicht nur auf die Anpassung der Regeleinrichtung, sondern auf die Verhaltensanpassung eines mechatronischen Systems im Ganzen [BSK+06, S. 4672]. Im Folgenden wird das Paradigma der Selbstoptimierung ausführlich erläutert. Hierzu werden die wesentlichen Aspekte eines selbstoptimierenden Systems und der Ablauf der Selbstoptimierung – der Selbstoptimierungsprozess – dargestellt.

2.3.1 Aspekte selbstoptimierender Systeme

Grundlage für die Betrachtung der verschiedenen Aspekte selbstoptimierender Systeme ist die Definition des Systembegriffs. Als **System** wird im Kontext der Selbstoptimierung und in Anlehnung an DIN 19226-1⁷ eine Anordnung von **Elementen** bezeichnet,

⁷ Die DIN-Norm 19226-1 ist im Jahr 2009 durch die in dieser Arbeit für regelungstechnische Begriffe maßgebliche Norm DIN IEC 60050-351 ersetzt worden. Da die grundlegenden Begriffe der Selbstoptimierung im Kontext des SFB 614 aber ursprünglich auf Basis der DIN 19226-1 festgelegt wurden, wird sie in diesem Abschnitt nach wie vor angewendet.

die durch eine physische oder gedankliche **Systemgrenze** von ihrem Umfeld abgegrenzt wurde. Entsprechend Bild 2-5 können die einzelnen Elemente selbst wiederum Systeme sein, wodurch eine hierarchische Strukturierung des Systems ermöglicht wird. Die Systemelemente sind durch gerichtete oder ungerichtete Beziehung miteinander oder mit Elementen des Umfelds verbunden. Die Beziehungen können u.a. nach den in Kapitel 2.2.2 eingeführten Flussarten nach PAHL/BEITZ unterschieden werden. Gerichtete Beziehungen zwischen Elementen des Systems und solchen des Umfelds werden entsprechend ihrer Flussrichtung in Eingangs- und Ausgangsgrößen unterschieden [ADG+09, S. 8], [DIN19226], [FGK+04, S. 16].

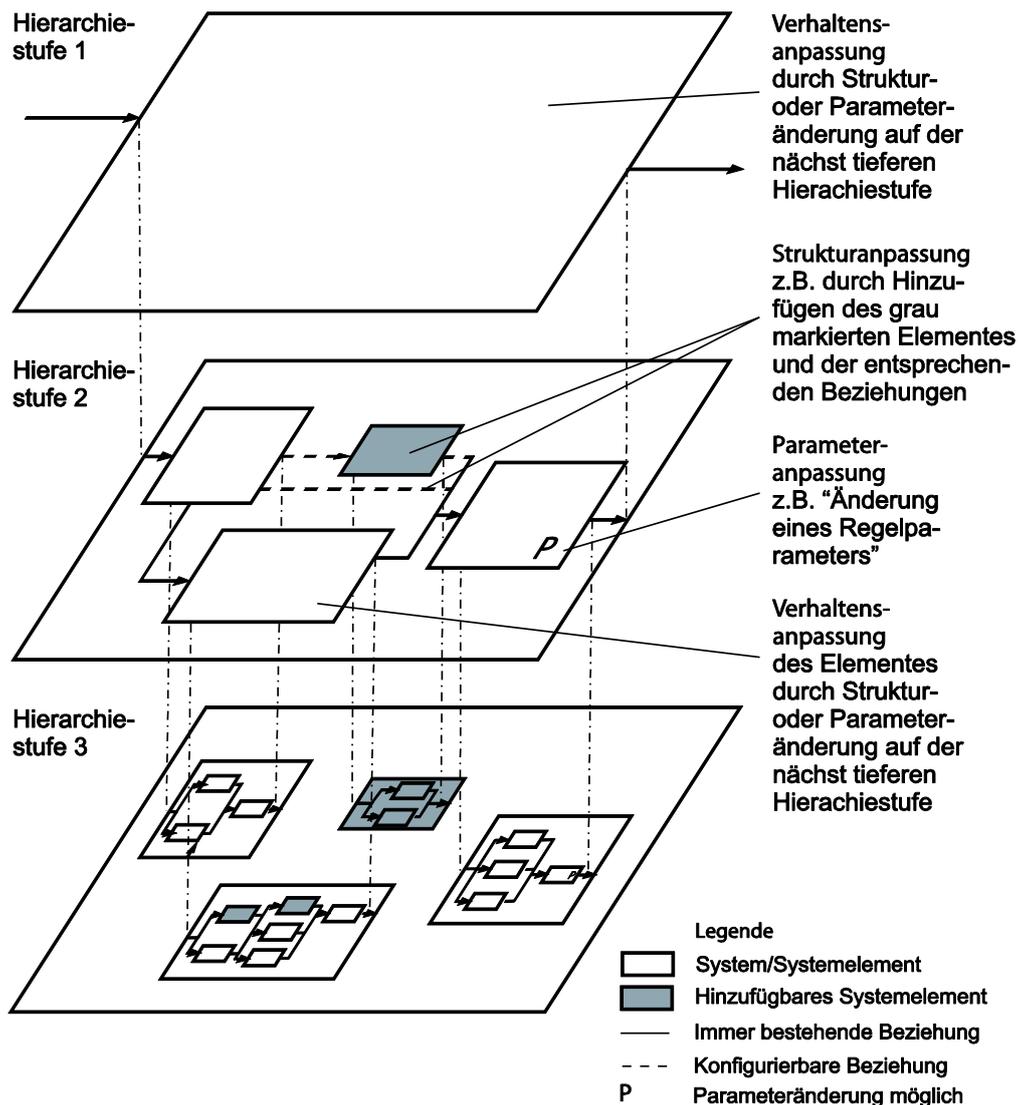


Bild 2-5 Hierarchiestufen selbstoptimierender Systeme und sich daraus ergebende Möglichkeiten zur Verhaltensanpassung [ADG+09, S. 7]

Auf das System wirkende Beziehungen werden als **Einflüsse** bezeichnet. Sie können nach Bild 2-6 vom Systemumfeld (der Umwelt, dem Benutzer oder anderen technischen Systemen) oder dem System selbst ausgehen. Einflüsse, die den Systemzweck behindern oder verhindern werden als Störgrößen angesehen. Ferner wird zwischen nicht

vorhersehbaren, nicht geplanten, unstrukturierten Einflüssen sowie geplanten und gezielten Einflüssen unterschieden. Geplante und gezielte Einflüsse werden als Vorgaben bezeichnet [ADG+09, S. 8], [FGK+04, S. 17].

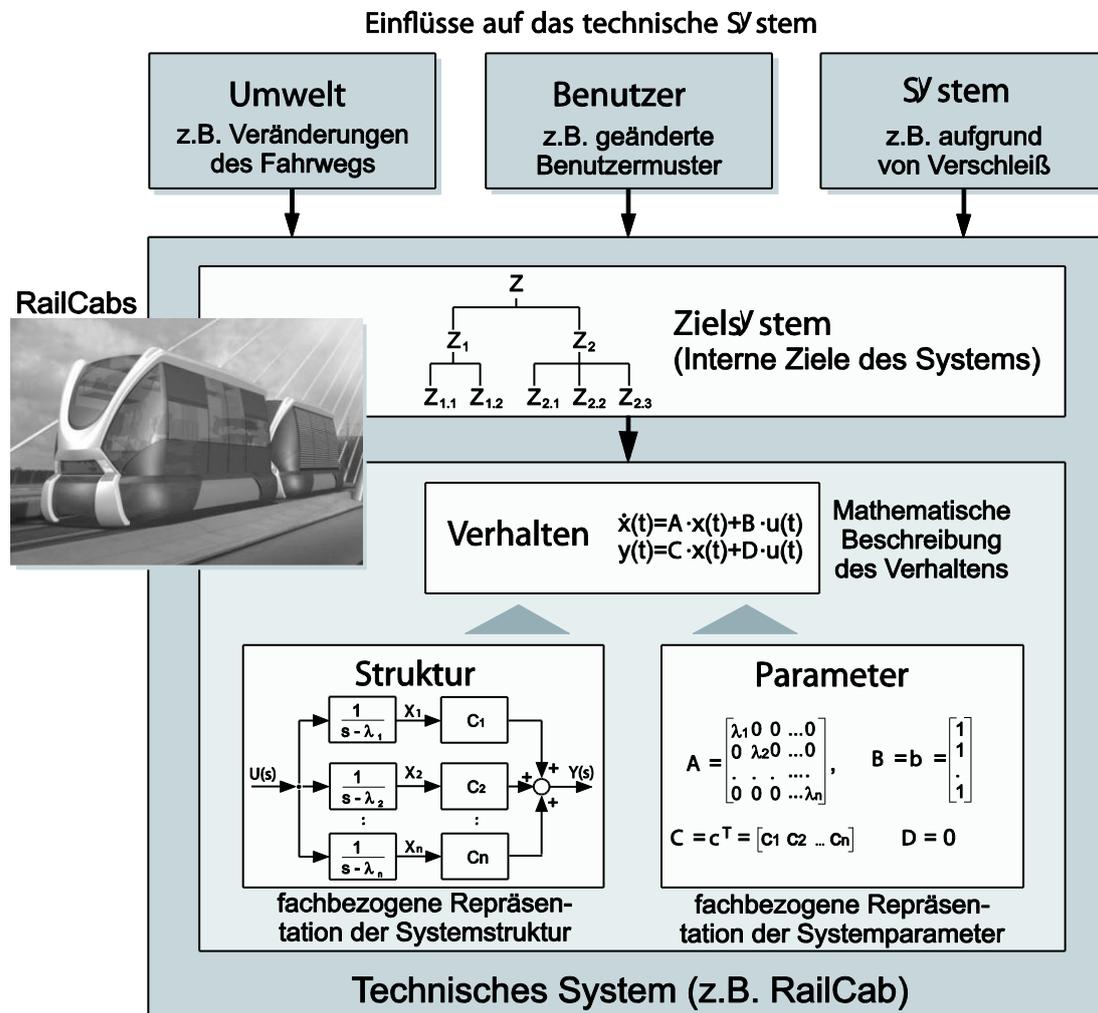


Bild 2-6 Aspekte eines Selbstoptimierenden Systems – Einflüsse auf das System bewirken eine Veränderung des Zielsystems und darauf aufbauend eine Anpassung des Systemverhaltens [SFB08, S. 18]

Die Einflüsse wirken sich bei selbstoptimierenden Systemen auf deren **Ziele** aus. Die Ziele formulieren die gewünschten, geforderten oder zu vermeidenden Eigenschaften eines Systems. Sie werden bereits bei der Entwicklung vorausgedacht. Im Unterschied zu Anforderungen werden die gewünschten Ausprägungen der Systemeigenschaften bei Zielen aber erst zur Laufzeit festgelegt. Grundsätzlich wird zwischen inhärenten, externen und internen Zielen unterschieden:

- Die **inhärenten Ziele** eines Systems spiegeln dessen Entwurfszweck wider und ergeben sich aus der Systemfunktionalität. Beispiele sind die Minimierung des Verschleißes oder die Maximierung der Energieeffizienz.

- **Externe Ziele** sind solche Ziele, die von außerhalb der Systemgrenze an das System herangetragen werden. Dies können bspw. Eingangsgrößen in Form von Vorgaben des Benutzers (z.B. gewünschter Fahrkomfort) oder anderer technischer Systeme (z.B. zu liefernde Leistung) sein.
- Aus den inhärenten und externen Zielen werden die **internen Ziele** des Systems gebildet. Sie sind die zu einem konkreten Zeitpunkt vom System verfolgten Ziele. Die Menge, das Gefüge und die Relevanz der internen Ziele wird dabei endogen durch das System selbst festgelegt.

Eine Menge zueinander in Beziehung stehender Ziele wird als **Zielsystem** bezeichnet. Wenn es sich dabei um eine lose Menge von Zielen handelt, wird das Zielsystem durch einen Zielvektor beschrieben. Im Falle hierarchischer Beziehungen zwischen den Zielen ist das Zielsystem eine Zielhierarchie. Hierbei werden Ziele höherer Hierarchieebenen in jeweils mindestens zwei Teilziele unterteilt. Die Ziele der höheren Ebene werden dann indirekt durch die der unteren Ebenen verfolgt. Stehen die Ziele durch komplexe Wechselwirkungen zueinander in Beziehung entspricht das Zielsystem einem Zielgraphen [ADG+09, S. 20f.], [Poo11, S. 82ff.].

Selbstoptimierende Systeme passen ihr Verhalten entsprechend ihres internen Zielsystems autonom an. Das **Verhalten** ist dabei der situationsabhängige Zusammenhang zwischen den Ein- und Ausgangsgrößen des Systems. Es hängt von der Struktur und den Parametern des Systems ab. Die **Struktur** wird durch die Gesamtheit der Elemente innerhalb der Systemgrenze und deren Beziehungen zueinander gebildet. Die Eigenschaften der Elemente und Beziehungen werden wiederum durch die **Parameter** des Systems beschrieben [FGK+04, S. 16]. Der genaue Ablauf der Verhaltensanpassung durch Selbstoptimierung wird im folgenden Kapitel 2.3.2 erläutert.

2.3.2 Der Selbstoptimierungsprozess

Selbstoptimierende Systeme sind in der Lage, sowohl ihre Parameter als auch ihre Struktur und daraus resultierend ihr Verhalten entsprechend ihres endogen gebildeten, internen Zielsystems autonom anzupassen. Sie führen hierzu aufeinanderfolgend und wiederkehrend drei Aktionen aus. Diese werden zusammengenommen als **Selbstoptimierungsprozess** bezeichnet [ADG+09, S. 6]:

- 1) **Analyse der Ist-Situation:** Hierbei wird der aktuelle Zustand des Systems und der Systemumgebung erfasst. Die Erfassung kann dabei direkt durch systemeigene Sensoren erfolgen oder indirekt durch Kommunikation mit anderen Systemen. Zum aktuellen Zustand werden u.U. auch bisher gemachte und gespeicherte Beobachtungen des Systems gezählt. Ein wesentlicher Aspekt bei der Analyse der Ist-Situation ist die Bestimmung des Erfüllungsgrades der aktuellen internen Ziele. Zudem wird geprüft, ob diese für die aktuelle Situation weiterhin angemessen sind.

- 2) **Bestimmung der Systemziele:** In diesem Schritt wird das interne Zielsystem bei Bedarf modifiziert. Für die Modifikation bestehen die Möglichkeiten der Auswahl, der Anpassung oder der Generierung. Bei der Auswahl werden die internen Ziele aus einer fest vorgegebenen, diskreten Menge möglicher externer und inhärenter Ziele entnommen. Bei der Anpassung der internen Ziele wird deren Ausprägung oder Gewichtung verändert. Im Falle der Generierung werden unabhängig von den bisherigen Zielen gänzlich neue gebildet. Grundsätzlich können die drei genannten Arten der Zielmodifikation auch als Mischform auftreten.
- 3) **Anpassung des Systemverhaltens:** Das Verhalten des Systems wird entsprechend dem modifizierten internen Zielsystems angepasst. Die Verhaltensanpassung auf einer Hierarchieebene des Systems kann nach Bild 2-5 durch Parameter- oder Strukturanpassung auf den jeweils unterlagerten Hierarchieebenen erfolgen. Bei der Parameteranpassung werden die Eigenschaften der Elemente oder der Beziehungen eines Systems verändert. Die Strukturanpassung wird in Rekonfiguration und kompositionale Anpassung unterschieden. Unter Rekonfiguration wird die Neuordnung oder Neuverknüpfung einer fest vorgegebenen Menge von Elementen verstanden. Bei der kompositionalen Anpassung werden neue Elemente hinzu genommen oder aktiviert bzw. bisher beinhaltete Elemente entfernt oder deaktiviert. Eine Übersicht der möglichen Arten der Verhaltensanpassung und ihre Zusammenhänge zeigt Bild 2-7. Zwischen den genannten Arten der Verhaltensanpassung sind auch Mischformen möglich [DHK+09, S. 3f.].

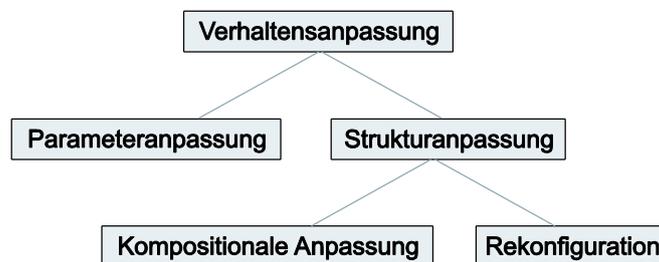


Bild 2-7 Arten der Verhaltensanpassung in selbstoptimierenden Systemen [ADG+09, S. 7]

Die drei Aktionen des Selbstoptimierungsprozesses können reaktiv oder proaktiv ausgelöst werden, müssen aber nicht immer in der genannten Reihenfolge durchlaufen werden. Zudem kann eine Aktion im Rahmen eines Selbstoptimierungsprozesses mehrfach vorkommen. Werden bspw. Planungsverfahren für die Realisierung des Selbstoptimierungsprozesses eingesetzt, kann es sein, dass das System verschiedene mögliche Situationen durchspielt, für die es dann jeweils interne Ziele und entsprechende Verhaltensanpassungen identifiziert. Es werden also wechselseitig die Analyse der Ist-Situation und die Bestimmung der Systemziele durchgeführt. Die Anpassung des Systemverhaltens erfolgt erst, wenn einer der Pläne tatsächlich ausgeführt wird. Einhergehend mit der Anpassung des Systemverhaltens ist ein Zustandsübergang des Systems.

Je nach dem auf welche Art und Weise die Elemente eines Systems am Selbstoptimierungsprozess beteiligt sind, werden zwei Arten von selbstoptimierenden Systemen unterschieden. Wenn der Selbstoptimierungsprozess von den Elementen eines isolierten Systems ausgeführt wird, ist es ein **individuell selbstoptimierendes System**. Besteht das isolierte System dabei aus weiteren Subsystemen, dürfen diese nicht mit eigenen Zielen am Selbstoptimierungsprozess des Systems beteiligt sein. Verfolgen die Subsysteme eigene interne Ziele, handelt es sich um ein **zusammengesetztes selbstoptimierendes System** [ADG+09, S. 9]. Solche Systeme können z.B. als Multi-Agenten-Systeme realisiert werden.

Unabhängig davon, um welche Art von selbstoptimierenden Systemen es sich handelt, sind Optimierungsverfahren die wesentliche Grundlage für den zweiten und dritten Schritt des Selbstoptimierungsprozesses. In beiden Schritten müssen in der Regel mehrere, einander widersprechende Ziele oder Stellhebel gleichzeitig optimiert werden. Durch die Widersprüchlichkeit der Ziele oder Stellhebel können typischerweise nur bestmögliche Kompromisse zwischen diesen im Sinne von Pareto-Punkten gefunden werden. Da diese zudem oftmals nicht eindeutig bestimmt sind, ergibt sich eine Menge von Kompromissen: die Paretomenge. Die eindeutige Auswahl eines Paretopunkts aus einer Paretomenge wird erst durch die gegenseitige Gewichtung der Ziele oder Stellhebel möglich. Diese Gewichtung ist i.d.R. nicht immer gleich, sondern situationsabhängig. Abhängig von dem der Optimierung zugrundeliegenden Modell werden zwei Arten von Optimierungsverfahren unterschieden [DHK+09, S. 9ff.]:

- Die **modellorientierte Optimierung** basiert auf einem ingenieurwissenschaftlichen (physikalischen) Modell des Systems. Die physikalischen Zusammenhänge zwischen den Ein- und Ausgängen des Systems sind bekannt und werden explizit beschrieben. Differentialgleichungssysteme zur Beschreibung der Systemdynamik sind Beispiele solcher Modelle. Auf diese können dann verschiedene mathematische Verfahren, bspw. der Mehrzieloptimierung oder der Optimalsteuerung, angewandt werden.
- Bei der **verhaltensorientierten Optimierung** wird auf die explizite Modellierung der zugrundeliegenden Physik verzichtet. Stattdessen wird das System als Black-Box betrachtet und die Auswirkungen der als relevant betrachteten Eingangsgrößen auf die jeweiligen Ausgangsgrößen direkt abgebildet. Die wesentlichen Verfahren im Rahmen der verhaltensorientierten Optimierung sind Lern- und Planungsverfahren. Lernverfahren zielen auf die Bestimmung der Auswirkungen der Ein- auf die Ausgangsgrößen und ggf. darauf aufbauend auf das Erlernen optimaler Reiz-Reaktions-Schemata. Planungsverfahren versuchen dagegen für die Bestimmung optimaler Verhaltensweisen neben der aktuellen Situation auch mögliche zukünftige Verhaltensweisen des Systems zu antizipieren.

Die Modelle beider Arten von Optimierungsverfahren können **kontinuierlich**, d.h. dass kleinste Veränderungen des Systemzustands in beliebig kleinen Zeitabständen abgebil-

det werden, oder **diskret** sein, d.h. Veränderungen des Systemzustands werden nur in festgelegten Zeitabständen und schrittweise abgebildet. Darüber hinaus gibt es **hybride** Modelle mit diskreten als auch kontinuierlichen Anteilen.

2.3.3 Das Operator-Controller-Modul

Ein derart mächtiges Paradigma wie die Selbstoptimierung erfordert für seine Realisierung in technischen Systemen ein geeignetes Strukturierungs- und Architekturkonzept für die Informationsverarbeitung dieser Systeme. Hierzu wurde das Konzept des **Operator-Controller-Moduls (OCM)** entwickelt. Entsprechend Bild 2-8 gliedert sich ein OCM in drei Ebenen: den Controller, den reflektorischen Operator und den kognitiven Operator.

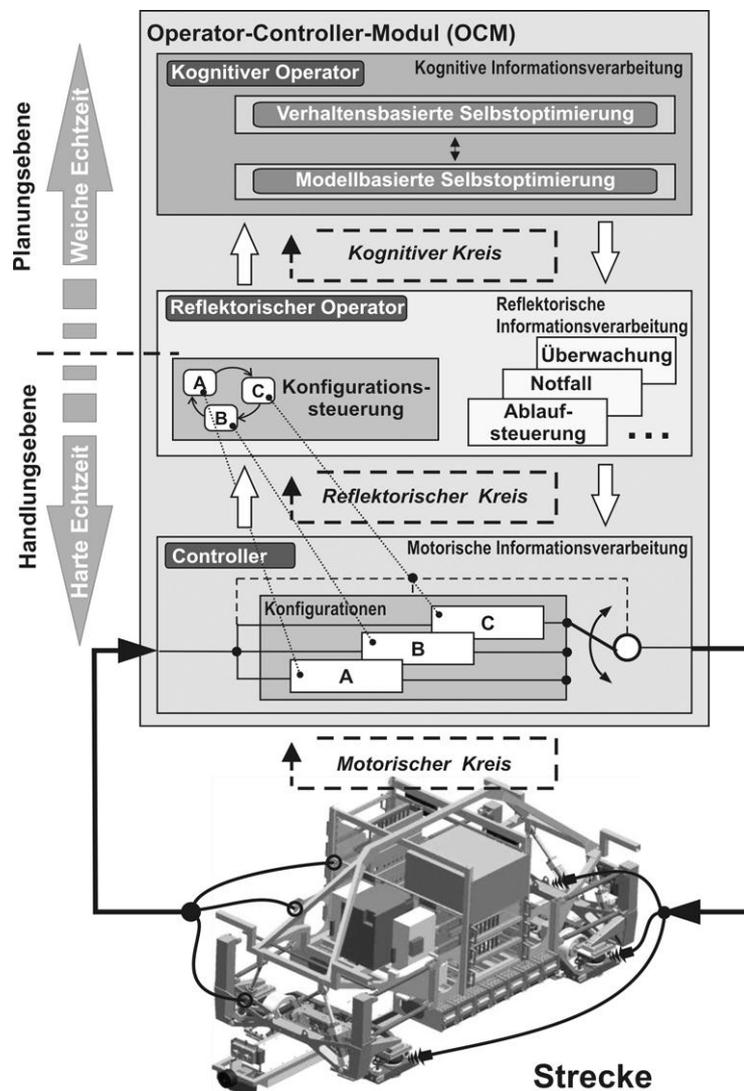


Bild 2-8 Architektur Operator-Controller-Modul (OCM) [ADG+09, S. 14]

- Die unterste Ebene des OCM bildet der **Controller**. Dieser Regelkreis verarbeitet in direkter Wirkkette die Messsignale, ermittelt Stellsignale und gibt diese aus. Er

entspricht dem klassischen Regelkreis und wird auch als „motorischer Kreis“ bezeichnet. Die Software auf dieser Ebene arbeitet quasi-kontinuierlich und muss harten Echtzeitbedingungen genügen. Der Controller kann sich aus mehreren Reglern zusammensetzen zwischen, denen umgeschaltet wird.

- Die nächst höhere Hierarchieebene des OCM bildet der **Reflektorischer Operator**. Er überwacht und steuert den Controller. Er greift dabei nicht direkt auf die Aktorik des Systems zu, sondern modifiziert den Controller indem er Parameter- oder Strukturänderungen initiiert (vgl. Abschnitt 2.3.2 Bild 2-7). Der reflektorische Operator arbeitet überwiegend ereignisorientiert. Als Verbindungselement zur kognitiven Ebene des OCM bietet der reflektorische Operator ein Interface zwischen den nicht echtzeitfähigen bzw. mit weicher Echtzeit arbeitenden Elementen des kognitiven Operators und dem Controller. Der reflektorische Operator filtert die ankommenden Signale und bringt sie in die unterlagerten Ebenen ein. Die Eingriffe in den Controller erfordern dabei harte Echtzeit. Der reflektorische Operator ist weiterhin für die Echtzeitkommunikation zwischen mehreren OCM verantwortlich, falls diese gemeinsam ein zusammengesetztes selbstoptimierendes System bilden.
- Die oberste Ebene des OCM ist der **kognitive Operator**. Der kognitive Operator kann durch die Anwendung vielfältiger Methoden (etwa Planungs- und Lernverfahren, modellorientierte Optimierungsverfahren oder den Einsatz wissensbasierter Systeme) Wissen über das System und dessen Umgebung zur Verbesserung des Systemverhaltens sammeln und anwenden. Verhaltensorientierte Verfahren werden in der Regel zur Planung und Bewertung der aktuellen Zielvorgaben eingesetzt. Die modellorientierten Verfahren werden allgemein für eine vorausschauende und vom realen System zeitlich entkoppelte Optimierung verwendet. Während sowohl Controller als auch reflektorischer Operator harten Echtzeitanforderungen unterliegen, kann der kognitive Operator auch asynchron zur Realzeit arbeiten. Dabei ist aber eine Antwort innerhalb einer gewissen Zeitspanne erforderlich, da die Ergebnisse des Selbstoptimierungsprozesses sonst aufgrund veränderter Umgebungsbedingungen nicht mehr verwendet werden können. Der kognitive Operator arbeitet deshalb in sogenannter weicher Echtzeit.

Im Rahmen der OCM-Architektur können die Aktionen des Selbstoptimierungsprozesses (1. Analyse der Ist-Situation, 2. Bestimmung der Systemziele und 3. Anpassung des Systemverhaltens) auf vielfältige Art und Weise realisiert werden [ADG+09, S. 14ff.], [GFD+08, S. 61f.].

2.3.4 Beispiele selbstoptimierender Systeme

Die maßgeblichen Demonstratoren für die erfolgreiche Übertragung des Paradigmas der Selbstoptimierung auf mechatronische Systeme wurden bis dato hauptsächlich im Rahmen des SFB 614 entwickelt (vgl. [ADG+09, S. 29ff.] und [GDD+10, S. 760ff.]). Bild 2-9 zeigt eine repräsentative Auswahl.

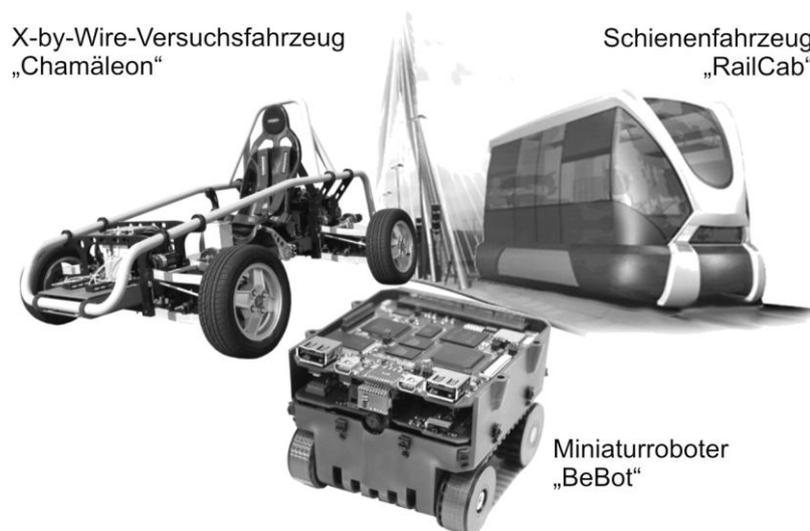


Bild 2-9 Demonstratoren des SFB 614 (in Anlehnung an [SFB08])

Der Hauptdemonstrator des SFB 614 ist ein innovatives Bahnsystem dessen Kern autonome Schienenfahrzeuge für den Personen- und Güterverkehr bilden; die sogenannten **RailCabs**⁸. Diese fahren nach Bedarf und nicht nach einem fest vorgegebenen Fahrplan. Ihr Verhalten ist proaktiv. Sie bilden bspw. auf längeren Streckenabschnitten selbstständig Konvois zur Reduzierung des Energiebedarfs. Zudem verfügen sie über die Fähigkeit Streckencharakteristika zu lernen und diese Information zur Verbesserung des Fahrkomforts zu nutzen. Der Antrieb der RailCabs erfolgt berührungslos mit Hilfe eines doppelt gespeisten elektromagnetischen Linearantriebs. Sein Stator befindet sich zwischen den Schienen, sein Rotor im Fahrzeug. Zudem ermöglicht der Antrieb die Energieübertragung in das Fahrzeug ohne Zuhilfenahme von Oberleitungen oder einer Stromschiene. [ADG+09, S. 29]

Der autonome Miniaturroboter **BeBot** stellt zum einen eine Versuchsplattform für neue Aufbau- und Verbindungstechniken dar, zum anderen dient er als Technologieplattform für die Forschung in den Bereichen dynamisch rekonfigurierbarer Systeme, Multiagentensysteme sowie Schwarmintelligenz⁹. Seine modulare Informationsverarbeitungsarchitektur und hohe Anzahl an Kommunikationsschnittstellen für die Vernetzung mit anderen BeBots erlaubt die Umsetzung und Erprobung der Selbstoptimierung auf verschiedenen Hierarchieebenen – insbesondere auf VMS-Ebene. Umgesetzte Szenarien, bei denen das Paradigma der Selbstoptimierung erfolgreich angewendet wurde, sind bspw. ein mobiles Ad-hoc-Kommunikationsnetzwerk zwischen mehreren BeBots oder

⁸ Website des Projekts „Neue Bahntechnik Paderborn/RailCab“ der Universität Paderborn: <http://www.railcab.de/>

⁹ Website des Projekts „Miniaturroboter BeBot“ am Heinz Nixdorf Institut in Paderborn: <http://www.whni.uni-paderborn.de/schwerpunktprojekte/miniaturroboter/>

die Kooperation innerhalb eines heterogenen Roboterteams bestehend aus BeBots und einer Roboterbasis mit stark andersartiger Physik. [ADG+09, S. 108ff.]

Der dritte dargestellte Demonstrator ist das vollständig elektrisch aktivierte X-by-Wire-Versuchsfahrzeug **Chamäleon**. Die Freiheitsgrade der Räder können individuell und mechanisch voneinander entkoppelt angesteuert werden. Dies ermöglicht eine hochgradig flexible Beeinflussung der Längs- und Querdynamik des Chamäleons. Im Kontext des SFB 614 dient das Chamäleon als Versuchsträger für eine selbstoptimierende Regelung der Vertikal- und Längsdynamik, des Energiemanagements sowie für die selbstoptimierende Rekonfiguration der Fahrwerksaktuatorik¹⁰. [GDD+10]

Die exemplarisch genannten Anwendungsbeispiele des Paradigmas der Selbstoptimierung beschreiben allesamt dessen Übertragung auf mechatronische, also technische Systeme. Auf Grund des in Kapitel 2.3.1 als Ausgangsbasis für die Definition des Paradigmas der Selbstoptimierung gewählten allgemeingültigen Systembegriffs ist aber eine Übertragbarkeit des Paradigmas auch auf nicht technische Systeme anzunehmen. Seine bisherige, hauptsächlich in mechatronischen Systemen stattfindende Anwendung ist der Fokussierung des SFB 614 auf diese Art von Systemen zuzuschreiben. Wie mechatronische Systeme entwickelt werden und welchen Herausforderungen die Entwickler dabei gegenüberstehen wird im nachfolgenden Kapitel 2.4 erläutert.

2.4 Entwicklungsprozess mechatronischer Systeme

Der Entwicklungsprozess mechatronischer Systeme ist ein soziotechnisches System (vgl. [Hit07]) in dem sich unterschiedliche Akteure, Tätigkeiten, Objekte, Einflüsse und Abhängigkeiten gegenseitig bedingen und effektuieren. Er kann hinsichtlich verschiedener Eigenschaften bzw. aus verschiedenen Blickwinkeln betrachtet werden und stellt demgemäß die Schnittmenge unterschiedlicher Dimensionen dar [Hup76, S. 3], [Lin07, S. 16ff.]. Da er unterschiedlichen, sich dynamisch verändernden Rahmenbedingungen unterliegt und stark vom zu entwickelnden Produkt und dessen Eigenschaften abhängt ist jeder Entwicklungsprozess einzigartig. Kein Entwicklungsprozess wird sich auf dieselbe Art und Weise wiederholen. Daher ist es zu Beginn der Entwicklung nicht möglich den Verlauf im Detail vorherzusagen [Alb10, S.4], [Lin07, S. 35].

Um diesen Sachverhalt zu unterstreichen, zeigt die nachfolgenden Tabelle 2-1 eine exemplarische Auswahl von im Rahmen der Konstruktionsforschung und verwandter Forschungsbereiche entstandenen Vorgehensmodellen die gemeinhin zum Ziel haben, die wichtigsten Tätigkeitsfolgen zur Entwicklung mechatronischer Systeme zu beschreiben. Die Tabelle erhebt keinen Anspruch auf Vollständigkeit. Sie zeigt aber deut-

¹⁰ Website des Schwerpunktprojekts „SFB 614 - Selbstoptimierende Systeme des Maschinenbaus“ des Heinz Nixdorf Instituts: <http://www.whni.uni-paderborn.de/schwerpunktprojekte/sfb-614/>

lich, dass die in den Vorgehensmodellen beschriebenen Tätigkeitsfolgen trotz des gleichen Anwendungsgebiets erheblich voneinander abweichen.

Tabelle 2-1 Auswahl von Vorgehensmodellen zur Beschreibung der Entwicklung er mechatronischer Systeme

Modell	Beschreibung	Quelle
SYSMOD	Inkrementelles Vorgehen zur Entwicklung der Struktur technischer Systeme auf Basis der Definition von Anwendungsfällen und des Systemkontexts.	[Wei06, S. 30f.]
Entwurfsschritte nach ISERMANN	Stufenweises Vorgehen, beginnend beim Systementwurf (Ergebnis: Labormuster), über die Systemintegration (Ergebnis: Funktionsmuster) zum Systemtest (Ergebnis: Vorserienprodukt).	[Ise08, S. 36]
3-Ebenen-Modell	Qualitätsorientiertes Vorgehensmodell, mit besonderer Betonung der Querbezüge zwischen Entwicklungs- und Prüftätigkeiten.	[Ben05, S. 45]
V-Modell der VDI 2206	Fallweise ausprägendes generisches Vorgehensmodell, das im Wesentlichen domänenübergreifenden Systementwurf, domänenspezifische Ausarbeitung und Systemintegration unterscheidet.	[VDI2206, S. 29]
Vorgehensmodell der VDI 2221	Sieben Arbeitsschritte umfassendes vierphasiges Vorgehen zur Entwicklung technischer Systeme das fallweise vollständig, zum Teil oder iterativ zu durchlaufen ist.	[VDI2221, S. 9]
Vorgehensmodell der VDI 2422	Vorgehensplan, bei dem ausgehend von einem Gerätekonzept parallel Software-, Schaltungs- sowie elektromechanischer Entwurf und Ausarbeitung erfolgen. Eine abschließende Erprobungsphase führt die Ergebnisse zusammen.	[VDI/VDE2224, S. 15]
Münchener Vorgehensmodell	Sieben Schritte (Elemente) umfassendes, flexibles Problemlösungsmodell. Die Schritte können sowohl sequentiell, als auch iterativ durchlaufen werden, wobei nicht logische Pfade ausgeschlossen sind.	[Lin07, S. 45ff.]
Problemlösungszyklus des Systems Engineering nach DAENZER ET AL.	Im Entwicklungsprozess wiederkehrend angewandte Problemlösungslogik, die im Wesentlichen Ziel-suche, Lösungssuche und Auswahl unterscheidet.	[DH02, S. 48]
TOTE-Schema	Auf einem Soll-Ist-Vergleich basierende grundsätzliche menschliche Verhaltenssequenz zur Problemlösung („Test-Operate-Test-Exit“).	[MGP73, S 34ff.]

Für einen tieferen Einblick in die Mechanismen des Entwicklungsprozesses mechatronischer Systeme werden nachfolgend die Rahmenbedingungen und inhärenten Eigenschaften des Entwicklungsprozesses mechatronischer Systeme dargestellt. Ferner werden ausgewählte Strategien zur effizienten Bearbeitung von Entwicklungsprozessen vorgestellt.

2.4.1 Rahmenbedingungen des Entwicklungsprozesses

Der Entwicklungsprozess mechatronischer Systeme ist nach Bild 2-10 Teil des Produktentstehungsprozesses¹¹. Dieser umfasst den gesamten Prozess von der Geschäftsidee bis zum erfolgreichen Serienanlauf und lässt sich in die drei Aufgabenbereiche strategische Produktplanung, Produktentwicklung und Produktionssystementwicklung unterteilen. Die drei Aufgabenbereiche bzw. die von ihnen implizierten Tätigkeiten bedingen sich gegenseitig und sind daher nicht sequentiell, sondern zyklisch zu bearbeiten.

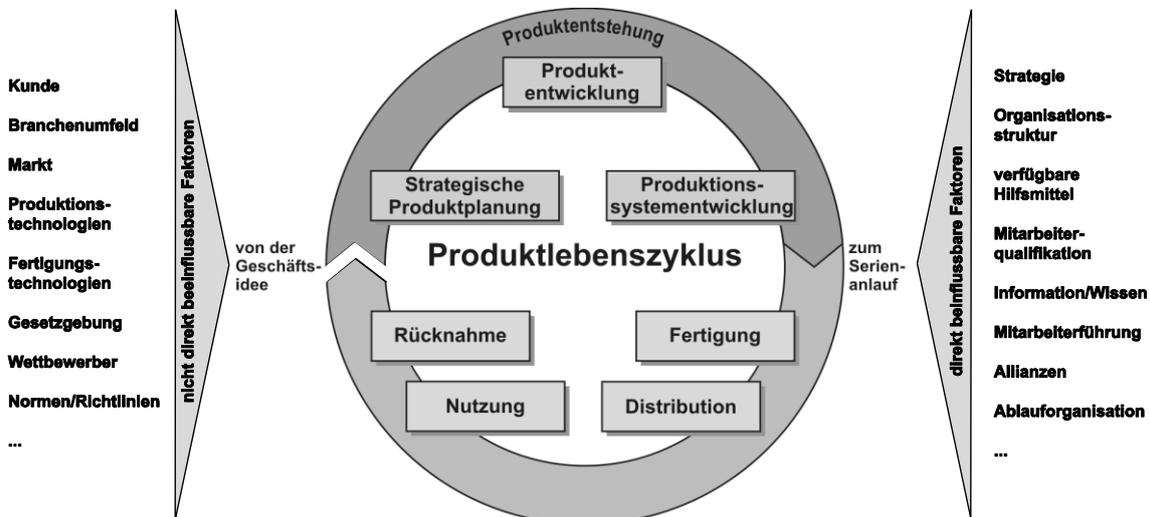


Bild 2-10 Rahmenbedingungen des Entwicklungsprozesses (in Anlehnung an [GLR+00, S. 3 +18])

Der Produktentstehungsprozess unterliegt Einflussfaktoren von denen einige Faktoren, wie bspw. die Organisationsstruktur, verfügbare Hilfsmittel oder Mitarbeiterqualifikation, direkt durch das Unternehmen beeinflusst werden können und andere Faktoren, wie bspw. das Branchenumfeld, Normen/Richtlinien oder der Kunde, nicht. Der Großteil dieser Einflussfaktoren greift z.T. direkt auf den Entwicklungsprozess durch.

Alle Einflussfaktoren des Entwicklungsprozesses können mit unterschiedlicher Dynamik unterschiedliche Ausprägungen annehmen. Im Ergebnis kommt es zu einer Vielzahl nur schwer vorhersehbarer, unterschiedlicher Rahmensituationen unter denen die

¹¹ Neben dem Begriff „Produktentstehungsprozess“ werden in der Literatur auch die Begriffe „Produktstellungsprozess“ [Ehr07, S. 1 ff.] oder „Kooperatives Produktengineering“ [GLR+00, S. 3 ff.] verwendet.

Produktentwicklung effizient durchgeführt werden muss [GLR+00, S. 3 + 18], [GPW09, S. 38f.], [Lin07, S. 8].

2.4.2 Inhärente Eigenschaften des Entwicklungsprozesses

Neben den vorstehend beschriebenen äußeren Rahmenbedingungen der Produktentwicklung hat der Entwicklungsprozess mechatronischer Systeme spezifische inhärente Eigenschaften, die ihn teils stark von anderen Unternehmensprozessen unterscheiden. Von diesen inhärenten Eigenschaften werden im Folgenden die menschl原因ierten und die strukturellen Eigenschaften des Entwicklungsprozesses betrachtet.

2.4.2.1 Menschmotivierte Eigenschaften

Die menschl原因ierten Eigenschaften ergeben sich durch den im Entwicklungsprozess schöpferisch tätigen Menschen. Der Entwickler beeinflusst durch seine Fähigkeiten und Verhaltensweisen den Entwicklungsprozess mechatronischer Systeme.

Der Entwicklungsprozess als Problemlösungs- und Lernprozess

Seinem Wesen nach ist der Entwicklungsprozess aus der Sicht des Entwicklers ein **Problemlösungs- und Lernprozess**. Unter Zuhilfenahme von Wissen, weiteren Informationen und materiellen Ressourcen wird eine durch Entwicklungsziele beschriebene Problemstellung in die vollständige Beschreibung eines technischen Systems, das diese Problemstellung löst, überführt.

Problem und zu entwickelndes System werden dabei auf **unterschiedlichen Abstraktions- und Detaillierungsebenen** sowie hinsichtlich **unterschiedlicher Aspekte** bearbeitet. Diese drei Dimensionen spannen den Entwurfsraum auf innerhalb dessen sich die Entwickler während des Entwicklungsprozesses bewegen. Die erarbeiteten Ergebnisse werden dabei zur Absicherung des weiteren Vorgehens immer wieder validiert. [Alb10, S. 5], [Ehr07, S. 57 ff.], [GFD+08, S. 62ff.], [GHK+06, S. 26ff.], [Hup76, S. 5], [Lin07, S. 16]

Rückgriff auf bekannte technische Lösungen

Zur Problembewältigung greifen die Entwickler in der Regel auf ihnen **bekannte technische Lösungen** zurück, wobei diese in unterschiedlich abstrakter Form, z.B. als Lösungsprinzip (physikalischer Effekt, informationsverarbeitendes Verfahren), Lösungsmuster oder konkretes Lösungselement vorliegen können [Dum11, S. 128ff.], [GFD+08, S. 63]. Die Strategien zur Problembewältigung werden unterteilt in ein *generatives Vorgehen*, bei dem mehrere gleichberechtigte Lösungen für ein Problem gesucht werden, zwischen denen dann ausgewählt wird, und ein *korrigierendes Vorgehen*, bei dem ausgehend von einer einmal gefundenen Lösung diese auf Schwachstellen analysiert und problemspezifisch verändert wird [Ehr07, S. 116f.].

Wiederholungen als Bestandteil des Entwicklungsprozesses

In einem Wechselspiel aus Analyse¹²- und Syntheseschritten¹³ nähern sich die Entwickler gleichermaßen einem immer detaillierteren Entwicklungsziel und der zugehörigen Lösungsbeschreibung an [Alb10, S. 4ff.], [Lin07, S. 20ff.]. Typischerweise führen dieses Wechselspiel und der damit verbundene Erkenntnisgewinn, z.B. dass die gefundene Lösung die Entwicklungsziele nicht erfüllt, zu **Wiederholungen** einzelner oder mehrerer Entwicklungsschritte. Rücksprünge in vorherige Arbeitsabschnitte zur wiederholten Betrachtung des gleichen Problems werden als „Iterationen“ bezeichnet. Die wiederholte Ausführung von Entwicklungsschritten auf unterschiedlichen Abstraktionsebenen mit dem Ziel einer immer detaillierteren Problemerkennung bzw. -lösung wird „Rekursion“ genannt [Ehr07, S. 91].

Wiederholungen sind der Entwicklung mechatronischer Systeme inhärent und lassen sich nicht vermeiden. KREHMER ET AL. unterscheiden in Abhängigkeit davon, wie weit im Entwicklungsprozess zurückgesprungen wird „kleine Wiederholungen“ und „große Wiederholungen“¹⁴. Kleine Wiederholungen sind insbesondere solche, bei denen sich der Entwickler der Lösung der Entwicklungsaufgabe iterativ annähert. Sie entsprechen dem natürlichen Problemlösungsverhalten des Menschen und sind entsprechend zu unterstützen. Die ihnen gegenüberstehenden großen Wiederholungen ergeben sich bspw. auf Grund unklarer Entwicklungsziele, geänderter Randbedingungen der Entwicklung oder der Feststellung, dass die bisher angestrebte Lösung sich nicht realisieren lässt. Große Wiederholungen bedeuten zwar wie bei kleinen einen Informationsgewinn für die Entwicklung, sind aber mit einem größeren Anteil an Wiederholarbeit verbunden. Ihr Anteil sollte daher nach Möglichkeit gering gehalten werden [KSM08, S. 3ff.], [RKL+08, S. 1ff.].

Unsicherheiten hinsichtlich Zielsetzung und Vorgehensweise

Der Entwicklungsprozess ist ein Lernprozess da zu Beginn u.U. noch **Unklarheit hinsichtlich der genauen Entwicklungsziele** herrscht und im Verlauf abhängig von den bisher erarbeiteten Entwicklungsergebnissen neue Ziele hinzukommen können bzw. schlimmstenfalls bestehende nicht erfüllt werden können und angepasst werden müssen [Alb10, S. 1ff.], [Lin07, S. 22]. Ferner ist den Entwicklern während der Entwicklung stellenweise nicht klar, welche **Vorgehensweisen und Hilfsmittel** sie für das vorlie-

¹² Unter „Analyse“ wird im Rahmen dieser Arbeit die „...systematische Untersuchung eines Gegenstandes od. Sachverhalts hinsichtlich aller einzelnen Komponenten od. Faktoren, die ihn bestimmen...“ [Dud01, S. 62] verstanden.

¹³ Unter „Synthese“ wird im Rahmen dieser Arbeit in Anlehnung an DAENZER ET AL. „...die erfolgreiche Ausführung einer konstruktiven und kreativen Tätigkeit zur Lösung eines analysierten Problems...“ verstanden [DH+02, S. 52].

¹⁴ KREHMER ET AL. schreiben statt „Wiederholung“ „Iteration“, da sie nicht zwischen „Iteration“ und „Rekursion“ unterscheiden.

gende Entwicklungsproblem anwenden sollen. Entsprechend dieser zwei Kriterien unterscheidet DÖRNER vier Problemklassen, mit denen sich Entwickler im Entwicklungsprozess auseinandersetzen müssen [Dör79, S. 14ff.]. FRICKE ordnet jeder Problemklasse jeweils eine Entwicklungsaufgabenklasse zu (Bild 2-11) [Fri93, S. 8ff.].

		Entwicklungsproblem Vollständigkeit und Klarheit der Entwicklungsziele	
		hoch	gering
Entwickler Anwendbarkeit von bekannten Vorgehensweisen und Hilfsmittel	hoch	Interpolationsproblem <i>Einfache Variantenkonstruktion</i> z.B. Variation in der Größe und/oder der Anordnung von Bauteilen	Dialektisches Problem <i>Einfache Entwicklungsstudie</i> z.B. Können Anforderungen für das Produkt nicht ermittelt werden
	gering	Syntheseproblem <i>Komplexe Anpassungskonstruktion</i> z.B. Anpassung bewährter Lösungsprinzipien und -muster an veränderte Rahmenbedingungen	Synthese und dialektisches Problem <i>Neukonstruktion</i> z.B. sind die Anforderungen und Lösungswege unklar

Bild 2-11 Problem- und Entwicklungsaufgabenklassifikation (in Anlehnung an [Dör79, S. 14ff.] und [Fri93, S. 8ff.]

- Wenn die Entwicklungsziele sowie zielführende Vorgehensweisen und Hilfsmittel dem Entwickler vollständig bekannt sind, liegt die Problemklasse eines *Interpolationsproblems* vor. Der Entwickler passt die Beschreibung des zu entwickelnden Systems hinsichtlich gegebener Zielgrößen mit ihm bekannten Vorgehensweisen und Hilfsmitteln an. Dies entspricht der Entwicklungsaufgabenklasse einer *Variantenentwicklung*, bei der in der Regel ausgehend von einem bestehenden System bestimmte Systemparameter skaliert werden, um das System der vorliegenden Problemstellung anzupassen.
- Im Falle eines *dialektischen Problems* weiß der Entwickler, wie er grundsätzlich vorgehen müsste und welche Hilfsmittel dabei zum Einsatz zu bringen sind, allerdings sind die Entwicklungsziele noch nicht hinreichend genau spezifiziert und bedürfen der weiteren Klärung. Diese Problemklasse korreliert mit der Entwicklungsaufgabenklasse der *Entwicklungsstudie*, bei der entsprechend bekannter Vorgehensweisen und Hilfsmittel die Problemstellung genauer analysiert und darauf aufbauend die Entwicklungsziele detailliert werden.
- Die Problemklasse des *Syntheseproblems* steht der des dialektischen Problems konträr gegenüber. Die Entwicklungsziele sind genau spezifiziert, der Entwickler ist aber im Unklaren darüber, welche Vorgehensweisen und Hilfsmittel er einsetzen muss, um diese zu erreichen. Der zugehörige Entwicklungsaufgabentyp ist die *An-*

passungsentwicklung, bei der bisher angewendete Lösungsprinzipien beibehalten werden, das zu entwickelnde System aber neu ausgestaltet wird.

- Die anspruchsvollste Problemklasse stellt die Mischung aus *Synthese und dialektischem Problem* dar. Hier sind weder die genauen Entwicklungsziele noch die zur Zielerreichung anwendbaren Vorgehensweisen und Hilfsmittel vollständig bekannt. Vielmehr nährt sich der Entwickler schrittweise einer genaueren Klärung der Entwicklungsziele an (dialektisches Problem) und versucht auf Basis dieser schrittweisen Klärung eine bessere Vorstellung über das im weiteren Verlauf anzuwendende Vorgehen zu bekommen (Syntheseproblem). Diese Problemklasse entspricht der Entwicklungsproblemklasse der *Neukonstruktion*, bei der neue Lösungsprinzipien oder -muster gefunden bzw. bestehende auf bisher nicht dagewesene Art miteinander kombiniert werden müssen.

2.4.2.2 Strukturelle Eigenschaften

Nach den vorstehend beschriebenen menschenmotivierten Eigenschaften von Entwicklungsprozessen mechatronischer Systeme werden in diesem Abschnitt ihre strukturellen Eigenschaften dargestellt. Sie beschreiben den inneren Aufbau des Entwicklungsprozesses. Eine schematische Übersicht gibt Bild 2-12.

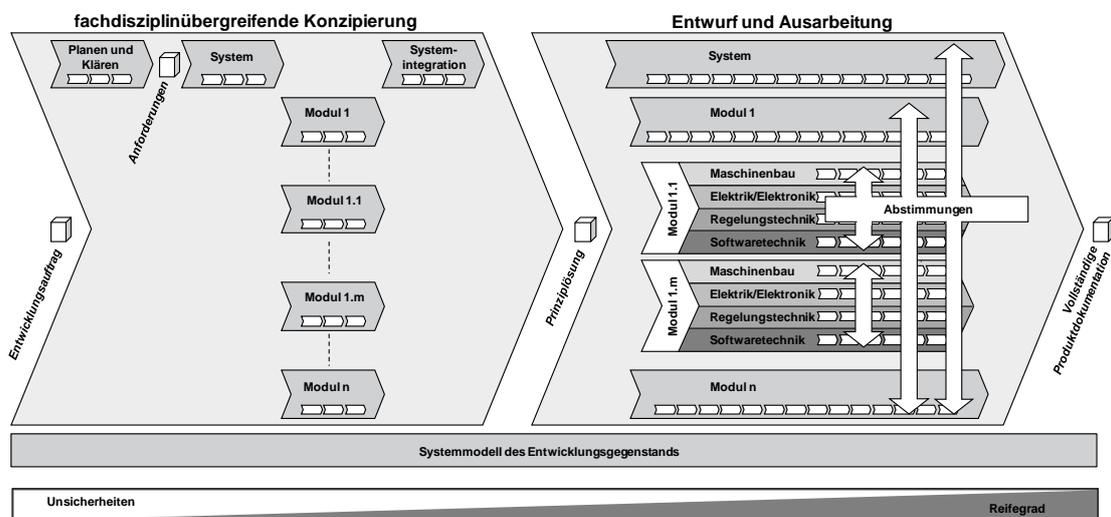


Bild 2-12 Strukturelle Eigenschaften des Entwicklungsprozesses mechatronischer Systeme

Der Entwicklungsprozess als Komposition von Entwicklungstätigkeiten

Der Entwicklungsprozess ist eine **Komposition von Entwicklungstätigkeiten**, die auf verschiedenen Detaillierungsebenen beschrieben werden können. Tätigkeiten der höheren Ebenen beschreiben grob die generellen Abschnitte bzw. Phasen des Entwicklungsprozesses, wie z.B. „fachdisziplinübergreifende Konzipierung“ oder „Entwurf und Ausarbeitung“. Da diese groben Phasen nicht ausführlich genug sind, um dem Entwickler

ausreichend Hilfestellung zu geben, wie er konkret vorgehen soll, werden sie durch Tätigkeiten auf einer niedrigeren Ebene detailliert bzw. setzen sich aus diesen zusammen [SHI+10, S. 1]. Bspw. kann die Phase „fachdisziplinübergreifende Konzipierung“ nach [GFD+08a, S. 96] in die Tätigkeiten „Planen und Klären der Aufgabe“, „Konzipieren auf Systemebene“, „Konzipieren auf Modulebene“ und „Konzepte integrieren“ unterteilt werden. In Bild 2-12 ist die Möglichkeit der Betrachtung unterschiedlicher Detaillierungsebenen durch das Verschachteln der durch Pfeile symbolisierten Entwicklungstätigkeiten dargestellt.

Bild 2-13 zeigt eine qualitative Gegenüberstellung von in der Literatur gängigen Detaillierungsrastern nach LINDEMANN [Lin07, S. 38], GIAPOULIS [Gia98, S. 103] und HUPKA [Hup76, S. 9]. Die Spanne der Detaillierung reicht von einzelnen Konstruktionsphasen oder -etappen auf oberster Stufe, wie z.B. „Konzipieren“ und „Konkretisieren“, bis zu elementaren Denk- und Handlungsweisen auf der untersten Stufe, wie z.B. „Analysieren“ oder „Synthetisieren“. Die Anzahl möglicher Detaillierungsstufen variiert bei den verschiedenen Autoren. Entsprechend überschneiden sich die Detaillierungsstufen und es gibt keine einheitliche Benennung bzw. Zuordnung.

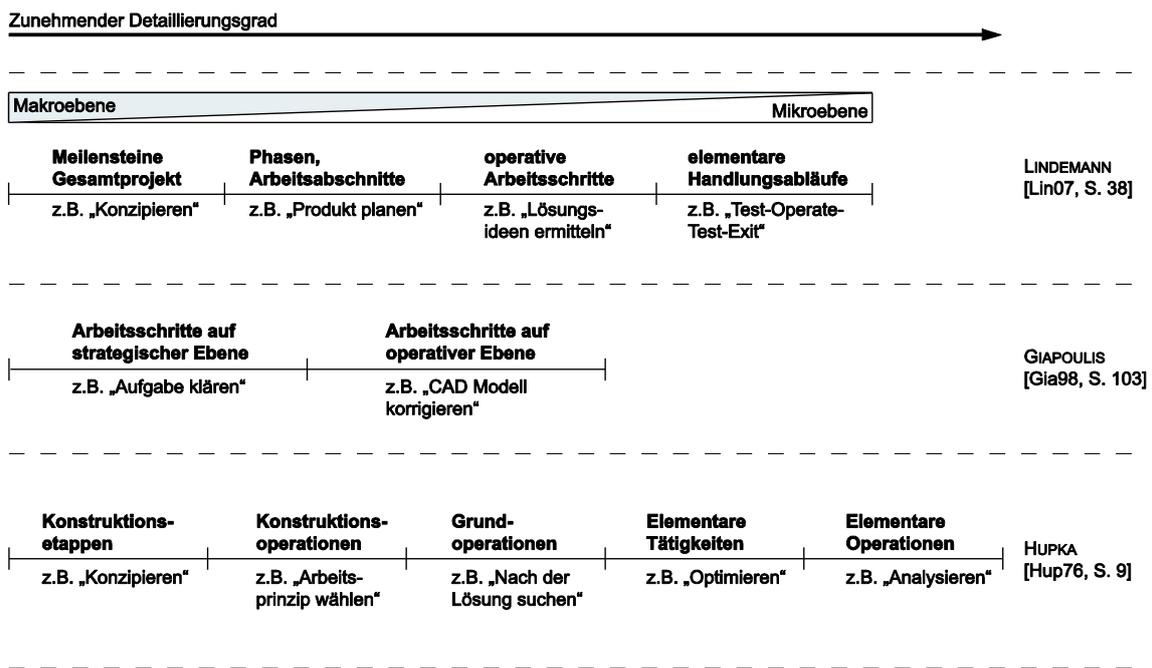


Bild 2-13 Qualitative Gegenüberstellung ausgewählter Detaillierungsraster von Entwicklungstätigkeiten (in Anlehnung an [Lin07, S. 38], [Gia98, S. 103] und [Hup76, S. 9])

Gleich bei allen drei Detaillierungsrastern ist, dass die Benennung der einzelnen Entwicklungstätigkeiten mit zunehmendem Detaillierungsgrad immer allgemeinerer Natur und damit wieder weniger problemspezifisch wird. BRAUN bestätigt diese Beobachtung anhand einer Klassifikation von Vorgehensmodellen zur Entwicklung mechatronischer Systeme. Er ordnet verschiedene Vorgehensmodelle den Detaillierungsstufen nach LINDEMANN zu und stellt fest, dass die Modelle höheren Detaillierungsgrades generischere

Inhalte haben als diejenigen niedrigeren Detaillierungsgrades. Die Vorgehensmodelle hohen Detaillierungsgrades mit entsprechend generischen Entwicklungstätigkeiten lassen sich auf eine große Anzahl verschiedener Problemstellungen anwenden. Die Vorgehensmodelle niedrigeren Detaillierungsgrades sind hingegen aufgabenspezifischer ausgeprägt [Bra05, S. 29]. Entsprechend können Vorgehensmodelle höheren Detaillierungsgrades in Tätigkeiten von denen niedrigeren Grades angewendet werden [Pul04, S. 79]. Bei der Planung des Entwicklungsprozesses für eine konkrete Entwicklungsaufgabe ist also ein Detaillierungsgrad zu wählen, bei dem die Tätigkeiten das Vorgehen für die Entwickler ausführlich genug beschrieben und gleichzeitig noch ausreichend spezifisch für die vorliegende Aufgabe sind.

Wechselwirkung mit der Struktur des zu entwickelnden Systems

Ähnlich wie der Entwicklungsprozess eine Komposition aus Teilprozessen ist, ist das **zu entwickelnde System selbst in Module (Subsysteme) unterteilt**. Die Strukturierung in Module erfolgt zur Reduktion der Komplexität der Entwicklungsaufgabe, indem das Gesamtproblem („zu entwickelndes System“) in Teilprobleme („zu entwickelndes Modul“) untergliedert wird. Die Module bestehen intern aus stark miteinander vernetzten Elementen während die externen Beziehungen zu anderen Modulen weniger intensiv sind. Für die Modularisierung mechatronischer Systeme existieren verschiedene Methoden, wie bspw. die auf EPPINGER und einen Kreis weiterer Autoren zurückgehende Design Structure Matrix (DSM) und ihre Weiterentwicklungen (vgl. [Bla99, S. 13ff.], [EWS+94, S. 3ff.], [LMB09, S. 50ff.], [LRZ06, S. 50ff.], [PE94, S. 345ff.], [SER07, S. 133ff.]), das Modular Function Deployment nach ERIXON [Eri98, S. 66ff.] oder das Verfahren zur Produktstrukturierung für fortgeschrittene mechatronische Systeme nach STEFFEN [Ste07, S. 87ff.]. Die Methoden werden typischerweise zu einem frühen Zeitpunkt im Entwicklungsprozess angewendet, wenn die Grundstruktur des zu entwickelnden Systems erarbeitet wird.

Die Modulstruktur ist zum einen Ergebnis der Entwicklungstätigkeiten, umgekehrt determinieren die Module aber auch die Struktur des Entwicklungsprozesses. Sie werden **parallel zueinander ausgearbeitet** [GFD+08a, S. 98]. Die Entwicklungstätigkeiten folgen dann der Modulstruktur. Die Verantwortung für ein Modul wird dabei in der Regel einem oder mehreren Entwicklern übertragen [Mül00, S. 44]. Der modulare Aufbau des zu entwickelnden Systems und die sich daraus ergebende Struktur des Entwicklungsprozesses wird in Bild 2-12 durch die vertikal untereinander angeordneten Entwicklungsprozesse einzelner Module (Modul 1 bis n) und Submodule (Modul 1.1 bis 1.m) dargestellt.

Kooperation verschiedener Fachdisziplinen

Da mechatronische Systeme auf der Symbiose verschiedener Fachdisziplinen beruhen müssen diese entsprechend zusammenarbeiten. Der Entwicklungsprozess unterteilt sich in Tätigkeiten, bei denen die Entwickler gemeinsam **fachdisziplinübergreifende** Ent-

wicklungsergebnisse erarbeiten und solche, bei denen sie **parallel zueinander fachdisziplinspezifische** Ergebnisse erarbeiten [GFD+08a, S. 96ff.].

Insbesondere zu Beginn der Entwicklung überwiegen die fachdisziplinübergreifenden Tätigkeiten. Die Entwickler erarbeiten gemeinsam ein **Systemmodell des Entwicklungsgegenstandes**, das den Aufbau und die Funktionsweise des zu entwickelnden Systems fachdisziplinübergreifend beschreibt. Jede Fachdisziplin trägt dabei ihren Teil zum Systemmodell bei. Hierfür hat es sich bewährt, auf Basis einer Umfeldanalyse des späteren Einsatzbereichs und von Anwendungsszenarien die genaue Problemstellung zu identifizieren und daraus die Anforderungen an das zu entwickelnde System abzuleiten. Aus den Anforderungen werden die Hauptfunktionen des zu entwickelnden Systems ermittelt, die dann hierarchisch in Teilfunktionen herunter gebrochen werden. Dies geschieht solange, bis für die einzelnen Teilfunktionen Lösungsmuster oder -elemente gefunden werden können (vgl. Abschnitt 2.4.2.1 „Rückgriff auf bekannte technische Lösungen“). Die Teillösungen werden zu Lösungsvarianten kombiniert, von denen die für die vorliegende Problemstellung am besten geeignete ausgewählt wird. Zur Komplexitätsreduktion und um die technische und wirtschaftliche Realisierbarkeit der Lösungsvarianten besser beurteilen zu können, wird das System mit den vorstehend beschriebenen Methoden modularisiert. Die Module werden von den Entwicklern gemeinsam verfeinert und anschließend integriert.

Die gemeinsamen Entwicklungstätigkeiten werden so lange betrieben, bis das fachdisziplinübergreifende Modell des Entwicklungsgegenstands einen Reifegrad erreicht hat, bei dem es alle Informationen für den Beginn der jeweiligen fachdisziplinspezifischen Ausarbeitungen umfasst. Dieser Reifegrad des Systemmodells des Entwicklungsgegenstands wird Prinziplösung genannt. Sie bildet den Startpunkt für die fachdisziplinspezifische Ausarbeitung der einzelnen Systembestandteile. Ziel der Ausarbeitung ist die eindeutige und vollständige Beschreibung des mechatronischen Systems durch Bau¹⁵- und Komponentenstruktur¹⁶. Hierfür setzen die Entwickler zum einen ihre fachdisziplinspezifischen Vorgehensweisen ein. Zum anderen stimmen sie ihre Entwicklungsergebnisse immer wieder anhand des Systemmodells aufeinander ab, um deren Konsistenz zu sichern. Eine beispielhafte Auswahl fachdisziplinspezifischer Vorgehensweisen zeigt Tabelle 2-2.

¹⁵ Unter einem „Bauteil“ wird im Rahmen dieser Arbeit ein gestaltbehaftetes Systemelement verstanden. Die Baustruktur beschreibt die logische Aggregation der Bauteile zu Baugruppen, sowie deren räumliche Anordnung [PBF+07, S. 56].

¹⁶ Unter einer „Komponente“ wird im Rahmen dieser Arbeit nach SZYPERSKI ein rein informationsverarbeitendes Systemelement verstanden [Szy02]. Die Komponentenstruktur beschreibt deren Gefüge [Oes98].

Tabelle 2-2 Beispiele für fachdisziplinspezifische Vorgehensweisen

Maschinenbau	Elektrik/Elektronik	Regelungstechnik	Softwaretechnik
Konstruktionsmethodik nach PAHL/BEITZ [PBF+07, S. 22]	Auslegung elektrischer Antriebe nach WEIDAUER [Wei11, S. 349ff.]	Lösungsweg für Regelungsaufgaben nach LUNZE [Lun10, S. 14ff.]	Der Rational Unified Process (RUP) [Bal08, S. 637ff.]
Konstruktionsmethodik nach KOLLER [Kol94, S. 75]	Mikroelektronik-Entwicklung nach SMITH [Smi97, S. 17]	Vorgehen zur Lösung regelungstechnischer Aufgabenstellungen nach MANN ET AL. [MSF09, S. 39ff.]	Scrum (Agile SW-Entwicklung) [Bal08, S. 670ff.]
Konstruieren mit Konstruktionskatalogen nach ROTH [Rot82, S. 23]	Auslegung elektrischer Antriebe nach BÖHM [Böh96, S. 98ff.]	Reglerentwurf nach FÖLLINGER [Föl08, S. 12f.]	Crystal Family nach COCKBURN [Coc02, S. 1ff.]
Arbeitsschritte des praktischen Konstruierens nach STEINWACHS [Ste76, S. 14]	Elektronikentwicklung nach WILCOX [Wil91, S. 34ff.]	Vorgehen zum Reglerentwurf nach KALLMEYER [Kal98, S. 39ff.]	Object Engineering Process OEP nach OESTEREICH [Oes98, S. 67ff.]
Konstruktiver Entwicklungsprozess nach HANSEN [Han74, S. 67ff.]	Y-Diagramm nach GAJSKI und KUHN [GK83, S. 13]	Methodik für die Reglersynthese auf Basis der Prinziplösung nach LOW [Low09, S. 89ff.]	V-Modell der Softwaretechnik [Bal08, S. 554]
Methodisches Entwickeln von Lösungsprinzipien nach VDI 2222 Blatt 1 [VI2222/1, S. 11 ff.]	Hardware-Software-Co-Design nach TEICH [Tei97, S. 6 ff.]		Spiralmodell [Boe88, S. 64ff.]
Methodisches Entwerfen nach VDI 2223 [VDI2223, S. 14ff.]			Wasserfallmodell [Roy70, S. 1ff.]

Darüber, wie das fachdisziplinübergreifende Systemmodell des Entwicklungsgegenstands zu repräsentieren ist, besteht derzeit keine einheitliche Meinung. Einige Autoren sehen dieses Modell als die Menge aller im Entwicklungsprozess entstehenden Entwicklungsobjekte¹⁷ [Alb10, S. 5], [Lin10, S. 22], [Neg98, S. 138ff.]. Die fachdisziplinübergreifenden Informationen liegen auf mehrere unterschiedliche Entwicklungsobjekte (z.B.

¹⁷ In Anlehnung an die Systemtheorie nennen ALBERS und NEGELE die Menge der Entwicklungsobjekte „Objektsystem“. LINDEMANN wählt die Bezeichnung „Entwicklungsmodell“.

CAD-Modelle, Stücklisten, Fertigungszeichnungen, Prototypen etc.) verteilt vor. Andere Autoren favorisieren einen zentralistischen Ansatz, bei dem die fachdisziplinübergreifend relevanten Informationen in einer zentralen Systemrepräsentation zusammengeführt werden, die dann auf fachdisziplinspezifische Repräsentationen verweisen und mit diesen synchronisiert werden. EVERSHEIM ET AL. entwickelten hierfür einen Ansatz zur fachdisziplinübergreifenden Produktdefinition mit Software-Agenten. Produktbeschreibende Informationen werden dabei nicht durch verschiedene Entwicklungsobjekte repräsentiert, sondern zu Datenobjekten zusammengefasst, die in einem PDM-System abgelegt werden. Die einzelnen Entwicklungsobjekte werden zur Konsistenzsicherung des Gesamtsystems mit diesen Datenobjekten online synchronisiert [ES05, S. 136 ff.]. GAUSEMEIER ET AL. definieren acht Partialmodelle und deren Verknüpfungen für die konsistente, fachdisziplinübergreifende Beschreibung eines fortgeschrittenen mechatronischen Systems [GFD+08a, S. 91 ff.]. Basierend auf diesem konsistenten System von Partialmodellen werden fachdisziplinübergreifend relevante Informationen mit Hilfe von Graphtransformationen mit fachdisziplinspezifischen Entwicklungsobjekten gespiegelt. Die Umsetzung des Synchronisationsmechanismus erfolgt durch Triple Graph Grammatiken [GSG+09, S. 5ff.]. ZEMAN ET AL. stellen einen generischen Ansatz zur Entwicklung eines zentralen mechatronischen Systemmodells (MSM) vor und nennen Beispiele für darin zu modellierende Inhalte (im MSM „Sichtweisen“ genannt) [ZFH+11, S. 381ff.], [ZFH+11a]. Die vollständige Beschreibung der Inhalte bleiben sie jedoch schuldig und begründen dies damit, dass ein MSM für verschiedene Klassen von mechatronischen Systemen unterschiedlich auszuprägen sei.

In Bild 2-12 sind die fachdisziplinübergreifenden und die fachdisziplinspezifischen Entwicklungstätigkeiten durch verschiedene Grautöne voneinander unterschieden. Die Unterscheidung ist für die einzelnen Hierarchieebenen des zu entwickelnden Systems nicht dogmatisch zu verstehen, sondern wurde qualitativ vorgenommen. Es wurden die jeweils überwiegenden Tätigkeiten in den Vordergrund gerückt. Zudem müssen nicht immer alle Fachdisziplinen bei der Entwicklung eines Moduls oder Submoduls beteiligt sein.

Bei der Unterscheidung in fachdisziplinübergreifende Konzipierung und wechselweise fachdisziplinspezifische und -übergreifende Konkretisierung ist zu beachten, dass diese ebenfalls nicht eindeutig voneinander getrennt werden können. Es ist davon auszugehen, dass während eines Entwicklungsprojekts der Reifegrad einzelner Module zum Teil stark divergiert. Während manche Module bereits detailliert ausgearbeitet sein können, ist nicht auszuschließen, dass für andere lediglich eine konzeptionelle Lösung vorliegt. Die zentrale Rolle des Systemmodells und dessen über den Entwicklungsprozess zunehmender Reifegrad werden durch die beiden Rechtecke am unteren Bildrand repräsentiert. Mit zunehmendem Reifegrad steigt das Wissen über das zu entwickelnde System und die Unsicherheiten, z.B. hinsichtlich dessen technischer Machbarkeit, nehmen entsprechend ab.

Abstimmungsbedarf im Entwicklungsprozess

Bei der Entwicklung mechatronischer Systeme müssen die verschiedenen Tätigkeiten der Entwickler fortwährend aufeinander abgestimmt werden (vgl. Bild 2-1). Die Abstimmungen unterteilen sich in Synchronisationen und Integrationen, wobei diese beiden Konzepte nicht disjunkt zueinander sind. Sie werden folgendermaßen definiert:

Die Synchronisation ist das zeitliche aufeinander Abstimmen zweier oder mehrere Vorgänge (in Anlehnung an [Dud01, S. 969]).

Synchronisationen sind durch die Ein- und Ausgangsobjekten zweier oder mehrerer Tätigkeiten bestimmt. Sie liegen überall dort vor, wo eine Tätigkeit erst beginnen kann, wenn die Ergebnisse zweier oder mehrerer vorher durchzuführender Tätigkeiten vorliegen. Eine solche Tätigkeit synchronisiert die vorhergehenden. Bspw. kann erst nach einem passenden Elektromotor gesucht werden, wenn der zur Verfügung stehende Bau-raum und die benötigten Leistungsdaten vorliegen. Synchronisationen können zwischen den Tätigkeiten verschiedener Fachdisziplinen, verschiedener Module oder Submodule sowie systemübergreifend über alle Module vorkommen.

Die Integration ist die „(Wieder-)Herstellung einer Einheit aus Differenziertem“ [Dud01, S. 447].

Integrationen sind Tätigkeiten, bei denen ein fachdisziplinspezifisches Entwicklungsobjekt einer höheren Hierarchieebene des zu entwickelnden Systems erstellt wird, bspw. beim Erstellen eines CAD-Modells des Gesamtsystems auf Basis der CAD-Modelle einzelner Bauteile. Gleiches könnte z.B. für die Komponentenstruktur der Software vorkommen. Diese Form von Integrationen korreliert folglich mit den verschiedenen im zu entwickelnden System vereinigten fachdisziplinspezifischen, hierarchischen Strukturen. Diese Strukturen sind in der Regel nicht deckungsgleich, sondern divergieren. Ferner zählen zu den Integrationen Tätigkeiten, bei denen neben den Entwicklungsobjekten, die verschiedene Fachdisziplinen (z.B. Mechanik, Hydraulik, Pneumatik, Thermodynamik, FEM, Schaltungslayout etc.) des zu entwickelnden Systems beschreiben, ein fachdisziplinübergreifendes Entwicklungsobjekt erstellt wird. Beispiele sind Multiphysics-Simulationen oder Hardware-in-the-Loop-Tests. Diese zweite Form von Integrationen kommt dort vor, wo verschiedene fachdisziplinspezifische Systemeigenschaften für die Erfüllung einer Systemfunktion zusammenwirken müssen. Integrationen können genau wie Synchronisationen zwischen verschiedenen Fachdisziplinen, verschiedenen Modulen oder Submodulen sowie systemübergreifend vorkommen. Wenn eine Integration auf Basis bestehender Entwicklungsobjekte einzelner Subsysteme oder Fachdisziplinen erfolgt, ist sie gleichzeitig auch Synchronisation. Findet die Integration simultan mit der Erstellung der Entwicklungsobjekte statt ist sie es nicht.

2.4.3 Strategien zur effizienten Entwicklung mechatronischer Systeme

Auf Grund der in den vorstehenden Abschnitten 2.4.1 und 2.4.2 beschriebenen Rahmenbedingungen und Eigenschaften des Entwicklungsprozesses mechatronischer Systeme ist dessen effiziente Bewältigung eine komplexe Herausforderung. In den vergangenen Jahren wurden verschiedene Strategien entwickelt, um mit dieser Herausforderung umzugehen. Die Strategien geben grundsätzliche Handlungsanweisungen für die Gestaltung des Entwicklungsprozesses und verwenden zum Teil ähnliche Ansätze. Allen gemeinsam ist das Ziel der Verbesserung der Effizienz des Entwicklungsprozesses. Nachfolgend werden ausgewählte Beispiele solcher Strategien kurz charakterisiert.

- Der Fokus des **Simultaneous Engineering** ist es, unterschiedliche Tätigkeiten parallel bzw. überlappend ablaufen zu lassen. Dies gilt insbesondere für die Tätigkeiten zur Produkt- und Produktionssystemplanung. Hierdurch sollen die Qualität des Produktes und des Produktionssystems gesteigert sowie die Entwicklungszeiten und -kosten reduziert werden [AE06, S. 661], [VDI2209, S. 34].
- **Concurrent Engineering** und Simultaneous Engineering sind hinsichtlich des Bestrebens nach der Parallelisierung von Tätigkeiten identisch. Beim Concurrent Engineering kommt hinzu, dass die Tätigkeiten weiter herunter gebrochen und von mehreren Bearbeitern verschiedener Fachdisziplinen gemeinsam ausgeführt werden [VDI2209, S. 34] [Wie00, S. 17].
- Eine auf den strukturierten Entwurf technischer Systeme ausgerichtete Strategie ist das **Axiomatic Design**. Dabei werden ausgehend von den Kundenwünschen für alle Fachdisziplinen die funktionalen Anforderungen an das zu entwickelnde System identifiziert. Auf dieser Basis werden dann die physischen Systemparameter erarbeitet und darauf aufbauend die Prozesse. Für einen guten Entwurf sind während dieser Vorgehensweise bei jeder Tätigkeit das Unabhängigkeits-Axiom (die einzelnen Systemeigenschaften sollen voneinander möglichst entkoppelt sein) und das Informations-Axiom (es sollen keine unnötigen Informationen vorgehalten werden) anzuwenden [Par07, S. 17f.].
- Beim **Quality Function Deployment** (QFD) werden alle Unternehmensbereiche in den Produkt- und Produktionssystementwicklungsprozess einbezogen. Ziel ist, Produkte zu entwickeln, die optimal auf die Kundenwünsche ausgerichtet sind. Hierfür werden die aufeinander aufbauenden Planungsschritte: Qualitätsplanung Produkt, Qualitätsplanung Konstruktion/Teile, Qualitätsplanung Prozess und Qualitätsplanung Produktion durchlaufen. Ein in jedem Planungsschritt unterstützendes Hilfsmittel ist das House of Quality [Sys06, S. 110ff.].
- Ein Ansatz, der auf spezifische Entwicklungsziele und -randbedingungen ausgerichtete Entscheidungsrichtlinien und -hilfsmittel für die Produktentwicklung zusammenfasst, ist das **Design for X**. Ziel ist, Produkt und Entwicklungsprozess mit Hilfe von Regelwerken effizient zu gestalten [CE05, S. 307f.].

- Das **Dynamic Product Development** ist eine Strategie, bei der die endgültige Produktkonzeption bewusst bis zum Ende des Entwicklungsprojekts offen gelassen wird. Sie wird als Vision des zu entwickelnden Produkts betrachtet, die sich parallel zur Ausgestaltung des Produkts und des zugehörigen Produktionsprozesses mitentwickelt. Diese Vision wird allen an der Entwicklung beteiligten Personen permanent vor Augen geführt und das Entwicklungsgeschehen entsprechend dynamisch angepasst [Ott04, S. 207ff.], [Hol07, S. 34ff.].

In Einklang mit der für die Gestaltung des Entwicklungsgeschehens gewählten Strategie müssen die Tätigkeiten der an der Entwicklung beteiligten Akteure koordiniert und zu einem erfolgreichen Abschluss geführt werden. Dies ist Aufgabe des Projektmanagements, das im nachfolgenden Abschnitt 2.5 detailliert erläutert wird.

2.5 Projektmanagement

Die Ausführungen zu den Charakteristika von Entwicklungsprozessen mechatronischer Systeme im vorstehenden Abschnitt 2.4 haben gezeigt, dass jeder Entwicklungsprozess eines mechatronischen Systems bedingt durch dynamische, nie gleiche Rahmenbedingungen und seine inhärenten Eigenschaften einmalig ist. Typischerweise wird die Entwicklung eines mechatronischen Systems daher als Projekt organisiert. Unter einem Projekt wird in der vorliegenden Arbeit entsprechend der geltende Norm DIN 69901-5 ein...

„...Vorhaben, das im Wesentlichen durch Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist“ [DIN69901-5, S. 11]

verstanden. Diese Einmaligkeit bezieht sich u.a. auf die individuellen Zielvorgaben, zeitliche, finanzielle und personelle Begrenzungen oder eine projektspezifische Organisation. Die Koordination und erfolgreiche Abwicklung von Projekten ist Gegenstand des Projektmanagements. Stellenweise wird in der Literatur, wie bspw. bei LINDEMANN [Lin07, S. 14], das Projektmanagement zu den Strategien zur effizienten Entwicklung technischer Systeme gezählt (vgl. Abschnitt 2.4.3). Dies wird im Rahmen der vorliegenden Arbeit nicht getan, sondern unter dem Projektmanagement die...

„...Gesamtheit von Führungsaufgaben, -organisation, -techniken und -mitteln für die Initiierung, [...] Planung, [eingefügt durch den Autor: Ausführung, Überwachung und] Steuerung [...] [eingefügt durch den Autor: sowie] den Abschluss von Projekten“ [DIN69901-5, S. 14]

verstanden [ANSI/PMI99-001-2004, S. 38]. Das Projektmanagement ist folglich u.a. ein Mittel zur Anwendung bzw. Umsetzung einer Strategie. Im Folgenden Abschnitt 2.5.1 werden die fünf genannten Funktionen des Projektmanagements näher beschrieben und deren gegenseitige Beeinflussung dargestellt. Anschließend werden in Abschnitt 2.5.2

die neun für die erfolgreiche Ausführung dieser Aufgaben wichtigen Wissensgebiete des Projektmanagements dargestellt.

2.5.1 Aufgabenkomplexe des Projektmanagements

Entsprechend der im vorstehenden Abschnitt gegebenen Definition des Projektmanagements sind dessen wesentlichen Aufgabenkomplexe die *Initiierung*, die *Planung*, die *Ausführung*, die *Überwachung* und *Steuerung* sowie der *Abschluss* von Projekten:

Gegenstand der **Initiierung** ist die formelle Freigabe eines neuen Projekts oder einer Projektphase. Diese wird in Form des Projektauftrags erteilt. Ferner werden zeitlicher, finanzieller, organisatorischer und produktbezogener Umfang und Inhalt des Projekts vorläufig definiert [ANSI/PMI99-001-2004, S. 43ff.].

In der **Planung** wird im Wesentlichen der Projektmanagementplan erstellt. Dieser legt fest, wie das Projekt ausgeführt, gesteuert und überwacht wird. Er integriert weitere Managementpläne, wie z.B. den Termin-, Kosten-, Qualitäts- und Personalmanagementplan. Basis bilden die Detaillierung des vorläufig definierten Umfangs und Inhalts des Projekts sowie die Erstellung des Projektstrukturplans. Ausgehend vom Projektstrukturplan werden die zur Verrichtung der Arbeit notwendigen Tätigkeiten und deren Reihenfolge bestimmt; eng verknüpft mit der Schätzung bzw. Planung von Kosten, Personal und Dauer bzw. Terminen. Parallel hierzu findet die Planung der Qualitätsstandards, der Kommunikation, der Beschaffung und des Vertragswesens statt. Sämtliche Planungen werden von einer Analyse der mit dem Projekt sowie den erstellten Plänen verbundenen Risiken und der darauf aufbauenden ggf. notwendigen Definition von Gegenmaßnahmen flankiert [ANSI/PMI99-001-2004, S. 46ff.].

Die beschriebenen Planungstätigkeiten werden in der Regel im Verlauf eines Projekts wiederholt ausgeführt und der Projektmanagementplan entsprechend angepasst. Ursache hierfür sind im Projektverlauf neu gewonnene Informationen mit der Möglichkeit einer detaillierteren Planung oder unvorhergesehene Änderungen, die eine Anpassung der ursprünglichen Planung erforderlich machen. Generell dienen solche Plananpassungen dem Zweck die zu Anfang festgelegten Projektinhalte und -umfänge im Ganzen einzuhalten. Dieses Prinzip der fortlaufenden iterativen Projektplanung wird auch als „rollierende Planung“ bezeichnet [ANSI/PMI99-001-2004, S. 46].

Konform zum Projektmanagementplan werden in der **Ausführung** Personal und weitere Einsatzmittel für den Vollzug der festgelegten Projekttätigkeiten gelenkt und deren Schnittstellen koordiniert. Hierbei werden die festgelegten Qualitätssicherungsmaßnahmen berücksichtigt. Für die innerhalb des Unternehmens auszuführenden Tätigkeiten werden entsprechende Projektteams zusammengestellt und deren weitere Entwicklung festgelegt. Für die extern zu vergebenden Aufgaben werden potentielle Lieferanten angefragt und ausgewählt [ANSI/PMI99-001-2004, S. 55ff.].

Während der Projektausführung ist eine permanente **Überwachung und Steuerung** erforderlich. Hierzu werden die Leistungsdaten und das Risiko des Projekts hinsichtlich Inhalt, Umfang, Kosten, Termine und Qualität kontinuierlich gemessen und ggf. deren weitere Entwicklung prognostiziert und bewertet. Wird ein Änderungsbedarf für das Projekt identifiziert und liegen dann vorzunehmende Änderungsmaßnahmen vor, werden diese im Rahmen der Steuerung des Projekts umgesetzt. Ferner gehören zu den Überwachungs- und Steuerungsfunktionen das Fortschrittsberichtswesen, die Kontaktpflege mit allen Stakeholdern des Projekts und das Abwickeln von Vertragsangelegenheiten [ANSI/PMI99-001-2004, S. 59ff.].

Der letzte Aufgabenkomplex des Projektmanagements ist der Projektabschluss. Der **Abschluss** beendet alle Tätigkeiten eines Projekts. Dies kann durch die Fertigstellung des Projektergebnisses oder durch die Entscheidung zum vorzeitigen Abbruch des Projekts motiviert sein. Einhergehend mit dem Abschluss werden ggf. noch offene Punkte von für das Projekt abgeschlossenen Verträgen geklärt und die Verträge beendet [ANSI/PMI99-001-2004, S. 66ff.].

Die beschriebenen fünf Aufgabenkomplexe des Projektmanagement beeinflussen sich gegenseitig und können in ihrem Zusammenschluss als Regelungssystem zur Projektdurchführung betrachtet werden. Die Überwachung und Steuerung ist von zentraler Bedeutung und interagiert mit allen anderen Aufgabenkomplexen (vgl. Bild 2-14).

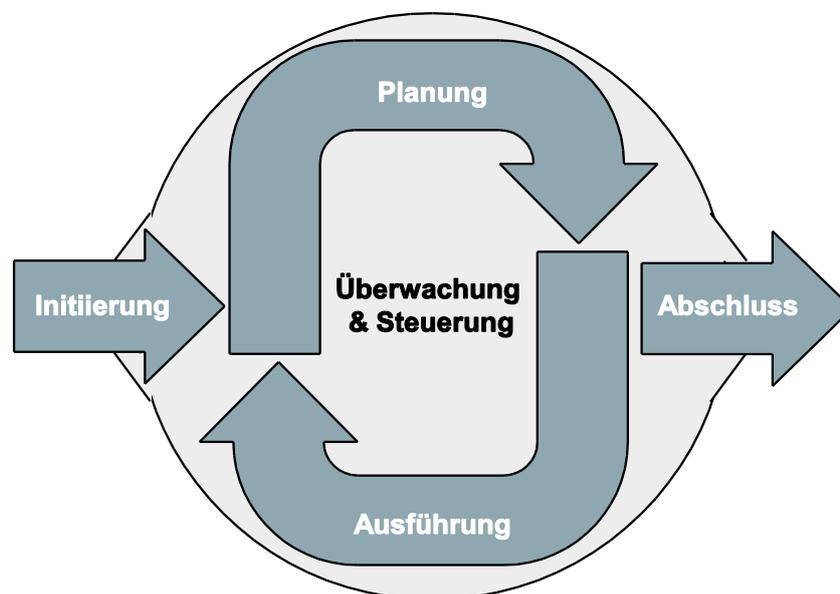


Bild 2-14 Schematische Darstellung des Zusammenwirkens der fünf Aufgabenkomplexe des Projektmanagements (in Anlehnung an [ANSI/PMI99-001-2004, S. 40])

2.5.2 Wissensgebiete des Projektmanagements

Nach dem PROJECT MANAGEMENT INSTITUTE werden im Projektmanagement neun Wissensgebiete unterschieden denen die Werkzeuge und Methoden für die Ausführung der fünf Aufgabenkomplexe des Projektmanagements zugeordnet werden können [ANSI/PMI99-001-2004, S. 70]. Die neun Wissensgebiete sind das *Integrationsmanagement*, das *Inhalts- und Umfangsmanagement*, das *Terminmanagement*, das *Kostenmanagement*, das *Qualitätsmanagement*, das *Personalmanagement*, das *Kommunikationsmanagement*, das *Risikomanagement* und das *Beschaffungsmanagement*. Tabelle 2-3 zeigt die Zuordnung der Wissensgebiete zu den Aufgabenkomplexen des Projektmanagements anhand der auszuführenden Tätigkeiten.

Tabelle 2-3 Zuordnung der Wissensgebiete zu den Aufgabenkomplexen des Projektmanagements [ANSI/PMI99-001-2004, S. 70]

Aufgabenkomplex / Wissensgebiet	Initiierung	Planung	Ausführung	Überwachung & Steuerung	Abschluss
Integrationsmanagement	Entwickeln des Projektauftrags Entwickeln der vorläufigen Beschreibung des Projektinhalts und -umfangs	Entwickeln des Projektmanagementplans	Lenken und Managen der Projektausführung	Überwachen und Steuern der Projektarbeit Integrierte Änderungssteuerung	Abschließen des Projekts
Inhalts- & Umfangsmanagement		Planung des Inhalts und Umfangs Definition des Inhalts und Umfangs Erstellen eines Projektstrukturplans		Verifizieren des Inhalts und Umfangs	

Aufgabenkomplex Wissensgebiet	Initiierung	Planung	Ausführung	Überwachung & Steuerung	Abschluss
Terminmanagement		Definition der Vorgänge Festlegen der Vorgangsfolgen Einsatzmittelbedarfschätzung für den Vorgang Schätzung der Vorgangsdauer Entwicklung des Terminplans		Steuerung des Terminplans	
Kostenmanagement		Kostenschätzung Kostenplanung		Steuerung der Kosten	
Qualitätsmanagement		Qualitätsplanung	Durchführen der Qualitätssicherung	Durchführen der Qualitätslenkung	
Personalmanagement		Personalbedarfsplanung	Zusammenstellen des Projektteams Entwickeln des Projektteams	Leiten des Projektteams	
Kommunikationsmanagement		Kommunikationsplanung	Informationsverteilung	Fortschrittsberichtsweisen Stakeholdermanagement	

Aufgabenkomplex Wissensgebiet	Initiierung	Planung	Ausführung	Überwachung & Steuerung	Abschluss
Risiko- management		Risikomanagement- planung Risikoidentifikation Qualitative Risiko- analyse Quantitative Risiko- analyse Risikobewältigungs- planung		Risikoüberwachung und -steuerung	
Beschaffungs- management		Planen der Einkäufe und Beschaffungen Planen des Vertragswesens	Lieferantenanfragen Lieferantenauswahl	Vertragsabwicklung	Vertragsbeendigung

Im Nachfolgenden werden die neun Wissensgebiete mit den jeweiligen für die vorliegende Arbeit maßgeblichen Methoden erläutert. Das **Integrationsmanagement** dient der Identifikation, Definition, Kombination, Vereinheitlichung und im Wesentlichen der Koordination der aus den anderen Wissensgebieten stammenden Tätigkeiten, die zu Beginn, während des Verlaufs und am Ende eines Projekts auszuführen sind. Ferner ist darüber zu entscheiden welche Ressourcen in welchem Umfang und wann für welche Tätigkeiten einzusetzen sind. Probleme sind vorherzusehen und Gegenmaßnahmen einzuleiten. Konkurrierende Zielsetzungen oder mögliche alternative Tätigkeitsfolgen sind gegeneinander abzuwägen und auszuwählen. Das Integrationsmanagement erstreckt sich folglich über alle Aufgabenkomplexe des Projektmanagements (vgl. Tabelle 2-3, Zeile 1).

Gegenstand des **Inhalts- und Umfangsmanagements** ist die Definition dessen, was zum Projekt gehört und was nicht (vgl. Tabelle 2-3, Zeile 2). Wesentliches Werkzeug hierfür ist der Projektstrukturplan. Er unterteilt die im Projekt auszuführende Arbeit in kleinere Arbeitspakete. Bei den Projektstrukturplänen wird zwischen objektorientiertem, funktionsorientiertem und ablauforientiertem Projektstrukturplan unterschieden.

Der objektorientierte Projektstrukturplan wird oft auch als produkt- oder erzeugnisorientierter Strukturplan bezeichnet. Die Definition der Arbeitspakete richtet sich nach der technischen Struktur des zu erstellenden Objekts. Beim funktionsorientierten Produktstrukturplan werden die Arbeitspakete nach den Funktionsbereichen der Entwicklung, wie z.B. Konstruktion, Prüftechnik oder Musterbau gegliedert. Der ablauforientierte Projektstrukturplan unterteilt die Arbeitspakete entsprechend der Phasen des Entwicklungsprozesses wie z.B. Konzipierung, Entwurf und Ausarbeitung. Gelegentlich treten auch Mischformen aus den drei Projektstrukturplantypen auf [Bur08, S. 161].

Der termingerechte Abschluss eines Projekts ist Aufgabe des **Terminmanagements**. Hierfür sind die auszuführenden Tätigkeiten festzulegen, deren Reihenfolge zu bestimmen und deren Dauern abzuschätzen, um auf dieser Basis den Terminplan des Projekts festzulegen und zu steuern (vgl. Tabelle 2-3, Zeile 3). Eine bewährte Methode zur Analyse, Planung und Kontrolle von Tätigkeitsfolgen ist die Netzplantechnik (siehe Abschnitt 3.2.1.5).

Wenn die logische Reihenfolge der auszuführenden Tätigkeiten festgelegt ist wird deren Dauer abgeschätzt. Hierfür können *algorithmische Methoden*, bei denen die Dauer mit Hilfe von mathematischen Methoden oder empirisch ermittelten Formeln bestimmt wird, *Vergleichsmethoden*, bei denen die Dauer durch Vergleich mit früheren Projekten ermittelt wird, oder mit *Kennzahlenmethoden*, bei denen die Dauer mit Hilfe von aus projekt- und produktspezifischen Messdaten extrahierten Kennzahlen ermittelt wird, angewendet werden [Bur08, S. 172ff.]. Auf dieser Basis wird dann der Projektterminplan erstellt. Dieser definiert, bis wann welche Aufgaben erledigt werden müssen. Für die Terminsteuerung während der Projektdurchführung können die Termine kontinuierlich durch einen Soll/Ist-Vergleich überwacht werden [Bur08, S. 354ff.].

Das **Kostenmanagement** umfasst die Tätigkeiten zur Schätzung, Planung und Steuerung der Projektkosten. Oberstes Ziel ist die Einhaltung des Projektbudgets (vgl. Tabelle 2-3, Zeile 4). Wie beim Terminmanagement können auch die Methoden zur Abschätzung der Kosten für ein Projekt in *algorithmische*, *vergleichs-* und *kennzahlenbasierte* Methoden unterteilt werden [Bur08, S. 172ff.]. Für die Steuerung der Projektkosten sind die Kosten des für das Projekt eingesetzten eigenen Personals inklusive der Sozialkosten, Arbeitsplatzkosten, Kommunikationskosten, arbeitsplatzbezogenen Dienstleistungskosten etc., die Kosten für fremdes Personal und weitere projektbezogene Kosten wie z.B. Rechner- und Maschinennutzung, Formen- und Musterbau, sonstige Dienstleistungen etc., zu erfassen. Die Kostenüberwachung kann dann über einen Soll/Ist-Vergleich der Kosten erfolgen [Bur08, S. 375ff.].

Die Kosten können auch als indirekte Kontrollgröße für die Ermittlung des Sachfortschritts eines Projekts dienen. Bei der Sachfortschrittskontrolle wird kontrolliert ob für die bisher im Projekt aufgewendeten Kosten auch eine adäquate Leistung vorliegt. Eine mögliche Kennzahl zur Bewertung des Sachfortschritts eines Projekts ist der Fertigstellungswert. Rechnerisch entspricht er dem Fertigstellungsgrad des zu generierenden Ob-

jekts multipliziert mit den Soll-Gesamtkosten des Projekts. Anhand des Vergleichs des Fertigstellungswertes mit z.B. den Ist-Kosten kann die Kostenabweichung des Projekts berechnet werden. Der Fertigstellungswert ist in der Literatur Basis für weitere Leistungs- und Bewertungskennzahlen [Bur08, S. 400ff.], [ANSI/PMI99-001-2004, S. 172ff.]. Eine Übersicht gibt die Norm DIN 69901-2 [DIN69901-2, S. 7].

Die Planung, Sicherung und Lenkung der Qualität des im Rahmen eines Projekts zu generierenden Objekts aber auch der zur Generierung des Projekts angewendeten Prozesse ist Gegenstand des **Qualitätsmanagements** (vgl. Tabelle 2-3, Zeile 5). Bei der Sicherstellung der Objektqualität kann zwischen der Validierung (generiere ich das richtige Objekt) und der Verifikation bzw. dem Testen (generiere ich das Objekt richtig) unterschieden werden. Verifikation bzw. Testen können im Kontext der Produktentwicklung entsprechend der möglichen Hierarchieebenen des zu entwickelnden Produkts z.B. in Komponententests, Modultests oder Systemtests unterschieden werden [Bur08, S. 411ff.], [Ben05, S. 45]. Zur Verbesserung der Prozessqualität können z.B. Reifegradmodelle wie das Capability Maturity Model Integration (CMMI) [CKS07, S. 45ff.], das Process and Enterprise Maturity Model (PEMM) [Ham07, S. 35ff.] oder das Prozess-Assessment nach dem derzeitigen Entwurf der DIN ISO/IEC 15504-2 [DIN ISO/IEC15504-2, S. 7ff.] herangezogen werden. Die Verbesserung der Prozessqualität wird in der Regel unternehmensweit betrieben und nicht isoliert auf ein Projekt. Aufbau und Verwirklichung eines umfassenden Projektmanagementsystems wird u.a. in den Normen der DIN-ISO-9000-Familie systematisiert [DIN EN ISO 9000, S. 4].

Im Rahmen des **Personalmanagements** wird der Mitarbeiterbedarf eines Projekts geplant, das Projektteam zusammengestellt, entwickelt und geleitet. Bei der Bedarfsplanung werden die im Projekt einzunehmenden Rollen, die Befugnisse, die Verantwortlichkeiten und die von den Mitarbeitern benötigten Kompetenzen festgelegt (vgl. Tabelle 2-3, Zeile, 6). Die Zusammenstellung des Projektteams erfolgt dann anhand dieser festgelegten Kompetenzen. Hierbei hat sich zur Unterstützung bei der Personalauswahl die Pflege von Mitarbeiterqualifikationsprofilen bewährt. Sie fassen die Fachkenntnisse, Fähigkeiten und Erfahrungen eines Mitarbeiters in einem einheitlichen Bewertungsrasster zusammen [Bur08, S. 306f.]. Die Qualifikationen können im Rahmen der Personalentwicklung durch gezielte Schulungsmaßnahmen verbessert werden. Ebenfalls dem Personalmanagement zuzuschreiben ist die Bewältigung ggf. auftretender Konflikte.

Die Organisation der Kommunikation der am Projekt beteiligten Stakeholder ist Aufgabe des **Kommunikationsmanagements** (vgl. Tabelle 2-3, Zeile 7). Bei Projekten mit geographisch verteilt arbeitenden Teams fällt hierunter auch die Bereitstellung geeigneter Kommunikationsnetze und -dienste [GHK+06, S. 39]. Ferner ist die Sammlung und bedarfsgerechte Bereitstellung von Wissen Gegenstand des Kommunikationsmanagements.

Zur Absicherung der Projektdurchführung wird ein kontinuierliches **Risikomanagement** benötigt (vgl. Tabelle 2-3, Zeile 7). Risiken sind mit Unsicherheit behaftete, pro-

jektintern oder -extern auftretende Ereignisse oder Bedingungen, die ein oder mehrere Projektziele negativ oder positiv beeinflussen können. Die potentiellen Risiken müssen identifiziert, bewertet und beobachtet werden. Für Risiken mit negativen Folgen sind Bewältigungsstrategien zu erarbeiten (z.B. Vermeidung, Übertragung oder Minderung), solche mit positiven Folgen sind als Chancen zu sehen und entsprechend zu fördern (z.B. durch Ausnutzung, Teilung oder Verbesserung). Wenn ein Projekt mit zu vielen Risiken mit negativen Folgen behaftet ist dann ist es Aufgaben des Risikomanagements die Beendigung des Projekts zu initiieren. Das Risikomanagement ist eines der am stärksten mit den anderen verwobenen Wissensgebiete.

Sämtliche von außerhalb des Projektteams benötigten Produkte, Dienstleistungen oder Ergebnisse werden im Rahmen des **Beschaffungsmanagements** erworben. Neben der Planung und Auswahl der Lieferanten von extern zu beschaffenden Objekten gehört auch die Gestaltung und Abwicklung des Vertragswesens zum Beschaffungsmanagement (vgl. Tabelle 2-3, Zeile 8).

Zur Unterstützung des Projektmanagements ist eine Vielzahl verschiedener Projektmanagement-Software-Werkzeuge entstanden die die Methoden eines oder auch mehrerer der beschriebenen Wissensgebiete umsetzen. Eine Aufstellung von in Deutschland verbreiteten Projektmanagement-Software-Werkzeugen zeigt bspw. BURGHARDT [Bur08, S. 592].

2.6 Problemabgrenzung

Die Entwicklung mechatronischer Systeme ist ein komplexer Prozess der einer Vielzahl von sich mit unterschiedlicher Dynamik ändernden Einflüssen unterliegt (vgl. Abschnitt 2.4). Ein Teil der Komplexität und Dynamik ist dem Entwicklungsprozess inhärent, ein Teil fremdbestimmt.

Die **inhärente Komplexität und Dynamik** resultieren aus dem Sachverhalt, dass der Entwicklungsprozess ein kreativer und iterativer Problemlösungs- und Lernprozess ist. Bei seiner Durchführung greifen die Entwickler teils auf bestehende technische Lösungen und Vorgehensweisen zurück, teils müssen sie neue Lösungen und Vorgehensweisen mit unsicherem Ausgang erarbeiten. Dies führt zu sich fortwährend aus dem Entwicklungsprozess heraus wandelnden Entwicklungszielen und in Folge dessen zu Veränderungen des aktuell in Entwicklung befindlichen Systems. Diesen Veränderungen muss der Entwicklungsprozess in seinen Zielen, seiner Struktur und seinem Ablauf folgen. Da bei der Entwicklung mechatronischer Systeme Entwickler und Technologien verschiedener Fachdisziplinen zusammenwirken erfordern Durchführung und Änderung des Entwicklungsprozesses zusätzlich einen erhöhten Kooperations-, Kommunikations-, und Koordinationsaufwand [KFG07, S. 61] gegenüber fachdisziplinspezifischen Entwicklungs- oder den i.d.R. weniger dynamischen übrigen Geschäftsprozessen eines Unternehmens.

Die **fremdbestimmte Komplexität und Dynamik** resultieren aus der Einbettung des Entwicklungsprozesses in ein durch das Unternehmen direkt beeinflussbares und ein nicht direkt beeinflussbares Umfeld des Entwicklungsprozesses. Zum direkt beeinflussbaren Umfeld gehören die finanziellen, materiellen, personellen und organisatorischen Rahmenbedingungen des Entwicklungsprozesses. Diese sind ebenfalls dynamisch da zum einen das die Entwicklung durchführende Unternehmen mit seinen Marktbegleitern in Konkurrenz steht und zum anderen auch das einzelne Entwicklungsprojekt mit anderen innerhalb des Unternehmens durchgeführten Entwicklungsprojekten um die verfügbaren Ressourcen konkurriert. Zum nicht direkt beeinflussbaren Umfeld gehören sämtliche weiteren Stakeholder mit Einfluss auf den Entwicklungsprozess, die allenfalls indirekt der Kontrolle des Unternehmens unterliegen.

Zusammengenommen führen die Bestandteile der inhärenten und fremdbestimmten Dynamik und Komplexität dazu, dass jeder Entwicklungsprozess einzigartig ist [Alb10, S. 4]. Ein spezifischer Entwicklungsprozess ist somit ein Projekt im Sinne eines in Randbedingungen und Inhalt einmaligen Vorhabens [DIN69901-5, S. 11] das einem teils mehr teils weniger, aber nie gänzlich zu vermeidenden, fortwährenden Wandel seiner Zielsetzungen, seiner Struktur, seines Ablaufs und seines Umfelds unterliegt. Die Ausführung von Entwicklungsprozessen erfordert daher eine fortlaufende Optimierung des Prozesses in Bezug auf die aktuelle Entwicklungssituation ähnlich dem Prinzip der „rollierenden Planung“ [ANSI/PMI99-001-2004, S. 46]. Der Schlüssel zum Erfolg bei dieser Managementaufgabe ist dabei die Handhabung der bei mechatronischen Systemen dramatisch angestiegenen Komplexität und Dynamik von Produkt und Prozess, um einen weiteren Anstieg der Kontroll- und Koordinationsaufwände zu vermeiden [Sch05, S. 3].

Die Gründe weshalb Unternehmen die Herausforderungen mechatronischer Entwicklungsprozesse dennoch annehmen liegen in dem gesteigerten **Nutzenpotential** und den damit einhergehenden **Wettbewerbsvorteilen** die **mechatronische Systeme** bieten. Die Mechatronik ermöglicht die Verbesserung der Funktionalität und des Betriebsverhaltens sowie die Reduktion von Kosten, Volumen und Gewicht technischer Erzeugnisse. Ferner können rundweg neue Funktionen in technischen Systemen realisiert werden; bis hin zu „teilintelligenten“ Verhaltensweisen. Sie sind bspw. in der Lage zielgerichtet, innerhalb gewisser Grenzen, autonom und flexibel auf sich ändernde Betriebssituationen zu reagieren [Ise08, S. 21ff.], [Möh04, S. 6ff.]. Die fortschreitende Entwicklung der Informations- und Kommunikationstechnologie wird die Leistungsfähigkeit und damit die Nutzenpotentiale mechatronischer Systeme noch weiter steigern. Ihr Verhalten wird zunehmend agierender Charakter bekommen. Diese Art mechatronischer Systeme kann als **fortschrittliche mechatronische Systeme** bezeichnet werden. Sie sind adaptiver, robuster, benutzerfreundlicher und entscheidungsfähiger als herkömmliche mechatronische Systeme und erschließen somit neue Nutzenpotentiale. Bspw. können sie noch flexibler und effizienter in dynamischen Umfeldern agieren. Eines der Schlüsselkonzepte auf dem Gebiet der fortschrittlichen mechatronischen Systeme sind **selbstoptimie-**

rende Systeme. Selbstoptimierende Systeme sind in der Lage auf Basis veränderter Betriebssituationen ihre Ziele und darauf aufbauend ihr Verhalten endogen anzupassen und zu optimieren [ADG+09, S. 5ff.].

Die Realisierung fortschrittlicher mechatronischer Systeme erfordert den Einsatz immer höherwertiger regelungstechnischer, informationstechnischer und mathematischer Verfahren und damit einhergehend die Integration weiterer Fachdisziplinen in den Entwicklungsprozess. Zudem steigt die Komplexität der Systeme an sich. Beides führt zu einem nochmaligen Sprung in der inhärenten als auch fremdbestimmten Dynamik und Komplexität des Entwicklungsprozesses fortschrittlicher mechatronischer Systeme.

Im direkten Vergleich bedeutet dies, dass die gewollte zunehmende Leistungsfähigkeit mechatronischer Systeme, flexibel auf komplexe und dynamische Betriebssituationen zu reagieren, zu ungewollt komplexen und dynamischen Situationen im Entwicklungsprozess dieser Systeme führen. Diese sind auf Grund der steigenden Kontroll- und Koordinationsaufwände nur schwer beherrschbar und verhindern damit einen effizienten Entwicklungsprozess und ggf. die vollständige Ausnutzung der Potentiale fortschrittlicher mechatronischer Systeme. Eine **Möglichkeit zur Auflösung dieses Konflikts** ist der **Umkehrschluss** vorstehender Ausführungen indem die Mechanismen zur Leistungssteigerung mechatronischer Systeme auf die Planung, Ausführung sowie Überwachung und Steuerung von deren Entwicklungsprozess übertragen werden und dieser im Sinne eines handelnden Systems flexibel auf komplexe und dynamische Entwicklungssituation reagiert.

Es besteht somit ein **Bedarf** nach einem *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* welches das Paradigma der **Selbstoptimierung** auf die Planung, Ausführung sowie Überwachung und Steuerung des Entwicklungsprozesses überträgt. Das Paradigma der Selbstoptimierung ist hierfür vor allen anderen Mechanismen zur Leistungssteigerung am besten geeignet. Seine Wirkungsweise, durch Anpassung der Systemziele an veränderte Betriebssituationen eine entsprechende Verhaltensanpassung zu bewirken, entspricht eins zu eins der Zielsetzung des Projektmanagements, bei einer veränderten Projektsituation eine Anpassung der Projektziele sowie der Projektplanung vorzunehmen und darauf aufbauend die Ausführung des Projekts zu verändern. Die Nutzung des Paradigmas der Selbstoptimierung ermöglicht es somit den Entwicklungsprozess endogen an veränderte Entwicklungssituationen anzupassen und diesen wieder beherrschbar zu machen. Im Sinne eines selbstverstärkenden Systems sollen so die Potentiale fortschrittlicher mechatronischer Systeme voll ausgeschöpft werden können.

Das Rahmenwerk soll drei Bestandteile beinhalten. Zum einen die theoretische *Übertragung des Paradigmas der Selbstoptimierung* auf den Entwicklungsprozess fortschrittlicher mechatronischer Systeme. Zudem muss es *Werkzeuge zur Umsetzung* eines selbstoptimierenden Entwicklungsprozesses bereitstellen die die bestehenden Methoden und Werkzeuge des Entwicklungsprozessmanagements ergänzen bzw. erweitern. Zu-

letzt bedarf es eines *Vorgehensmodells zur Initialisierung und Nachbereitung* selbstoptimierender Entwicklungsprozesse mit dem die Bestandteile des Rahmenwerks auf ein konkretes Entwicklungsprojekt innerhalb eines Unternehmens angewendet werden.

Ein derartiges Rahmenwerk adressiert Methoden für die Planung, Ausführung sowie Überwachung und Steuerung *eines* Entwicklungsprozesses innerhalb einer vorgegebenen Unternehmensorganisation. Ein **Multiprojektmanagement** [DIN69901-5, S. 10] oder der systematische Wandel der Unternehmensorganisation im Sinne der **Organisationsentwicklung** [Gab10, S. 331] sind nicht Gegenstand des Rahmenwerks.

2.7 Anforderungen an ein Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess

Vor dem Hintergrund des soeben aufgezeigten Handlungsbedarfs ergeben sich nachfolgende Anforderungen an ein *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme*:

A1 Ganzheitlichkeit: Der Entwicklungsprozess fortschrittlicher mechatronischer Systeme und sein Umfeld müssen als soziotechnisches System aufgefasst werden in dem unterschiedliche Akteure, Tätigkeiten, Objekte, Einflüsse und weitere Abhängigkeiten zusammenwirken und sich gegenseitig bedingen. Die einzelnen Subsysteme des Entwicklungsprozesses und ihre Wechselwirkungen sind zu strukturieren und voneinander abzugrenzen.

A2 Interdisziplinäre Wechselwirkungen: Die Fachdisziplinen der Mechatronik (Maschinenbau, Elektrik/Elektronik und Informationstechnik) und insbesondere ihre Wechselwirkungen müssen prozess- und systemmodelltechnisch abgebildet und beim Aufbau der Struktur des Entwicklungsprozesses berücksichtigt werden.

A3 Dynamische Vorausplanung: Die Vorausplanung des Entwicklungsprozesses muss dynamisch an die aktuell verfügbare Informationsbasis angepasst werden können. Die Planung soll mit wenig detaillierten Informationen über das zu entwickelnde System möglich sein wobei die Zunahme der Genauigkeit der Pläne mit steigender Informationsbasis unterstützt werden muss. Die Zunahme der Genauigkeit soll hierbei nicht nur die Auswahl geeigneter Entwicklungsschritte betreffen, sondern auch die der Zeit-, Kosten-, Hilfsmittel- und Ressourcenplanung

A4 Spezifität: Den Entwicklern müssen im Entwicklungsprozess Hilfestellungen in Form problemspezifischer Handlungsanweisungen gegeben werden. Diese sollen in der Vergangenheit erfolgreich angewandte Problemlösungsvorgehensweisen widerspiegeln wobei den Entwicklern noch Raum für eigene Problemlösungsstrategien bleiben muss.

A5 Problemunabhängigkeit: Das Rahmenwerk soll unabhängig von der Art des zu entwickelnden Systems oder der Entwicklungsproblemklasse angewendet werden können. Die beinhalteten Werkzeuge und Vorgehensweisen müssen entsprechend auf all-

gemeinen Eigenschaften mechatronischer Systeme aufbauen und ausgehend von dieser Basis weitere z.B. branchen- oder problemklassenspezifische Informationen integrieren.

A6 Selbstoptimierung: Die Methoden und Werkzeuge des Rahmenwerks müssen die situationsabhängige Anpassung der Ziele und des Ablaufes des Entwicklungsprozesses im Sinne des Paradigmas der Selbstoptimierung ermöglichen. Der Entwicklungsprozess soll seine Prozessziele aus sich selbst heraus managen und sich dadurch seinem aktuellen Zustand und den gegebenen Randbedingungen anpassen.

A7 Akzeptanz/Transparenz: Für die erfolgreiche Umsetzung des selbstoptimierenden Entwicklungsprozesses ist die Unterstützung der beteiligten Entwickler unabdingbar. Aus diesem Grund müssen die prozess- und systemmodelltechnischen Inhalte des Entwicklungsprozesses für alle beteiligten Entwickler transparent dargestellt werden ebenso wie die Schnittstellen zu den Werkzeugen die die Selbstoptimierung umsetzenden.

A8 Effizienz: Die Realisierung eines selbstoptimierenden Entwicklungsprozesses soll die Prozessverantwortlichen während der Durchführung entlasten und möglichst keinen Zusatzaufwand bedeuten. Aus diesem Grund sollen die Werkzeuge und Vorgehensweisen zur Realisierung des selbstoptimierenden Entwicklungsprozesses auf generell im Entwicklungsprozess verfügbare Informationen zugreifen und durch IT-Werkzeuge unterstützt werden.

3 Stand der Technik

In diesem Kapitel werden bestehende Ansätze mit Relevanz für ein *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* gemäß der in Kapitel 2.7 aufgestellten Anforderungen untersucht. Zuerst werden in Kapitel 3.1 Entwicklungsprozessmodelle analysiert, die den Entwicklungsprozess als ein aus Subsystemen bestehendes soziotechnisches System auffassen. In Kapitel 3.2 werden Modellierungstechniken zur Beschreibung von Prozessen und Systemmodellen dahingehend durchleuchtet inwieweit sie für die Beschreibung des Entwicklungsablaufs, von Entwicklungsziele und von Entwicklungsgegenstände geeignet sind. Anschließend folgt in Kapitel 3.3 die Untersuchung verschiedener dynamischer bzw. adaptiver Planungs- und Steuerungsverfahren für Entwicklungsprozesse. Kapitel 3.4 bewertet den Stand der Technik anhand der Anforderungen aus Kapitel 2.7 und zeigt den Handlungsbedarf für die vorliegende Arbeit auf.

3.1 Systemische Entwicklungsprozessmodelle

Wie in der Problemanalyse in Kapitel 2.4 dargestellt ist der Entwicklungsprozess fortschrittlicher mechatronischer Systeme ein Teil des Produktentstehungsprozesses. Er kann als soziotechnisches System aufgefasst werden (vgl. [Hit07]) in dem sich unterschiedliche Akteure, Tätigkeiten, Objekte, Einflüsse und Abhängigkeiten gegenseitig bedingen und effektuieren. Verschiedene Modelle beschreiben den Entwicklungsprozess daher aus einer systemischen Sicht, d.h. aus sich gegenseitig beeinflussenden Subsystemen bestehend. Diese systemischen Modelle bilden die Basis für ein besseres Verständnis des Entwicklungsprozesses und für die Entwicklung von Methoden und Werkzeugen zu dessen Beherrschung. Nachfolgend werden fünf systemische Modelle zur Beschreibung des Entwicklungsprozesses fortschrittlicher mechatronischer Systeme vorgestellt die als Grundlage für das in dieser Arbeit angestrebte Rahmenwerk dienen können.

3.1.1 Allgemeine Technologie nach ROHPOHL

Ein grundlegendes Modell mit dem das Entwicklungsgeschehen für ein technisches System beschrieben werden kann ist die erstmals 1978 veröffentlichte Modellvorstellung der Allgemeinen Technologie nach ROHPOHL [Rop09]. Sie bezieht sich auf die Technik, die

„...die gegenständlichen Artefakte oder Sachsysteme sowie deren Entstehung und deren Verwendung in soziotechnischen Systemen...“
[Rop09, S. 43]

umfasst. Technik ist also sowohl das technische System als Ergebnis (ingenieurmäßigen) Handelns, als auch das Handeln mit dem System selbst. ROHPOHL sieht die Tech-

nik als etwas von Menschen geschaffenes aber dennoch nicht gänzlich erfasstes und beherrschtes. Aus diesem Grund entwickelt er für den Umgang mit Technik bzw. die Entwicklung eines umfassenden Technikverständnisses auf Basis der allgemeinen Systemtheorie eine Systemtheorie der Technik. Das allgemeine Systemmodell wird in ein auf die Technik bezogenes Systemmodell überführt indem es als **Handlungssystem**, d.h. als

„...eine Instanz, die Handlungen vollzieht...“ [Rop09, S. 93]

interpretiert wird. Dabei können zwei Arten von Handlungssystemen unterschieden werden:

Zum einen menschliche Handlungssysteme. Hierzu gehören Individuen, Organisationen oder die Gesellschaft (bis hin zur Weltbevölkerung als globale Gesellschaft). Menschliche Handlungssysteme bestehen aus den drei Subsystemen: Ausführungssystem, Informationssystem und Zielsetzungssystem. Das Ausführungssystem verarbeitet stoffliche und energetische Eingangsgrößen in entsprechende Ausgangsgrößen. Das Informationssystem nimmt Informationen auf, verarbeitet diese und gibt sie weiter. Es ermöglicht somit die Kommunikation mit anderen Handlungssystemen. Das Zielsetzungssystem legt die **Ziele** des Handlungssystems intern fest. Ein Ziel ist dabei bewusst allgemein als der normative Ausdruck eines Sachverhaltes definiert. Hierunter fallen bspw. Wünsche, Bedürfnisse, Interessen, Zwecke, Normen oder Werte. Mehrere Ziele bilden gemeinsam ein Zielsystem.

Die zweite Art von Handlungssystemen sind künstliche Handlungssysteme. Diese werden als **Sachsysteme** bezeichnet. Sachsysteme sind:

„...nutzenorientierte, künstliche, gegenständliche Gebilde...“ [Rop09, S. 117],

d.h. durch menschliche Handlungssysteme oder andere Sachsysteme erstellte künstliche Dinge. Sie haben im Gegensatz zu menschlichen Handlungssystemen kein eigenes Zielsetzungssystem.

Beide Arten von Handlungssystemen stehen niemals allein, sondern sind immer in eine Umgebung eingebettet. Sie können aus weiteren Sub-Handlungssystemen bestehen oder Teil eines übergeordneten Handlungssystems sein. Zudem können menschliche Handlungssysteme und Sachsysteme gemeinsam ein Handlungssystem bilden. Dieses wird dann als soziotechnisches System bezeichnet.

Nach der Theorie der Allgemeinen Technologie ist der Entwicklungsprozess somit ein soziotechnisches System in dem menschliche Handlungssysteme und Sachsysteme zusammenwirken, um ein oder mehrere Sachsysteme entsprechend eines gemeinsamen Zielsystems zu erstellen.

Bewertung: Die Allgemeine Technologie nach ROPOHL beschreibt technische Systeme und ingenieurmäßig handelnde menschliche Systeme gleichermaßen. Die systemische

Strukturierung ist daher auf einer abstrakten Ebene gehalten ohne detaillierter auf die Strukturen mechatronischer Entwicklungsprozesse einzugehen. Zudem besteht in Teilen ein Widerspruch zwischen Rohpohls These, Sachsysteme würden kein eigenes Zielsetzungssystem besitzen, und dem Anspruch der Selbstoptimierung, technischen Systemen zu ermöglichen ihre Ziele eigenständig anpassen zu können.

3.1.2 Systemtechnik nach PATZAK

Zunehmende Komplexität, immer kürzere Innovationszyklen und fachdisziplinübergreifende Problemstellungen sind nach PATZAK [Pat82] die Hauptherausforderungen bei der Entwicklung technischer Systeme. Sie führen zu einer Überforderung des Planers und/oder der Entscheidungsträger im Entwicklungsprozess, da geeignete Methoden und Konzepte zur Strukturierung komplexer Problemstellungen und deren Lösungsablaufs fehlen. Um diesem Missstand Abhilfe zu leisten, entwickelt PATZAK seine Systemtechnik als Methodologie zur Planung komplexer Systeme mit der fachdisziplinspezifische Erkenntnisse und Methoden abstrahiert, verallgemeinert und auf beliebige andere Problemstellungen übertragen werden können um so

„...ein fachungebundenen Instrumentarium zur wirkungsvollen Behandlung und Lösung von komplexen interdisziplinären Problemstellungen zu schaffen.“ [Pat82, S. 2].

Der Entwicklungsprozess eines komplexen technischen Systems wird als Problemlösungsprozess verstanden in dem Zielvorstellungen in eine Realisierung zur Erfüllung der Ziele überführt werden. Der Entwicklungsprozess wird mit Hilfe von vier Systemtypen strukturiert (vgl. Tabelle 3-1).

Tabelle 3-1 Systemtypen des Entwicklungsprozesses nach PATZAK [Pat82, S. 30]

Systemtypen		Erscheinungsformen
- <u>Zielsysteme</u> auch: Bedarfssysteme	(abstrakt)	Pflichtenheft, Lastenheft, Anforderungsbeschreibung
- <u>Programmsysteme</u> auch: Aufgabensysteme	(abstrakt)	Projekt, Vorgehensplan, Programm, Prozeß
- <u>Wirksysteme</u> Auch: Handlungssysteme Arbeitssysteme	(konkret)	Organisation, Handlungsträger, Instrument, Sachmittel
- <u>Objektsysteme</u> Auch: Ergebnis, Problemlösung	(konkret/ abstrakt)	Arbeitsgegenstand, Produkt, Erzeugnis, Leistung, Ergebnis, Objekt, System schlechthin

Zielsysteme beschreiben das Handlungsergebnis im Endzustand. Sie sind abstrakte Systeme und können hierarchisch (Zielhierarchie) oder zeitlich (ablaufmäßige Zerlegung) strukturiert sein.

Die **Programmsysteme** beschreiben die zur Zielerreichung vorzunehmenden Aktivitäten. Sie sind ebenfalls abstrakte Systeme die die Art und Weise der auszuführenden Tätigkeiten in Form von Plänen beschreiben.

Wirksysteme führen dann die Aktivitäten aus und sind daher aus Sicht der Programmsysteme Mittel zum Zweck. Die Komponenten der Wirksysteme können belebt (z.B. Menschen) oder unbelebt (z.B. Sachmittel) sein. Treten belebte und unbelebte Komponenten gemeinsam in einem Wirksystem auf wird dies als soziotechnisches System bezeichnet. Das Zusammenwirken der Komponenten wird durch die Organisation im Sinne eines formellen oder informellen Beziehungsmusters festgelegt.

Die von den Wirksystemen zur Realisierung der Programmsysteme entsprechend der Zielsysteme ausgeführten Aktivitäten transformieren Objekte bzw. bringen diese hervor. Diese Objekte können beliebige konkrete oder abstrakte Systeme sein und werden im **Objektsystem** beschrieben.

Bewertung: Die Systemtechnik nach PATZAK fokussiert die Strukturierung des Entwicklungsgeschehens technischer Systeme mit dem Ziel dessen Komplexität beherrschbar zu machen. Die systemische Strukturierung des Entwicklungsgeschehens ist prozesstechnisch ausgerichtet und unterstützt sehr gut die Darstellung der im Entwicklungsprozess wirkenden Mechanismen. Konkrete Hilfestellungen bei der Abarbeitung einer spezifischen Entwicklungsaufgabe werden nicht gegeben.

3.1.3 ZOPH-Modell

Für die transparente Planung der bei der Produktentwicklung auszuführenden Tätigkeiten, die Abstimmung der dabei auftretenden Informationsflüsse und die Vernetzung aller verfügbaren Informationen entwickelt NEGELE [Neg06] das ZOPH-Modell als prozessorientierte Betrachtungsweise der Produktentwicklung. Die Buchstaben ZOPH stehen für **Z**iel-, **O**bjekt-, **P**rozess- und **H**andlungssystem. Diese vier in Anlehnung an die Systemtechnik nach PATZAK (s.o.) definierten Subsysteme bilden das in die Systemumwelt eingebettete Gesamtsystem Produktentwicklung, wobei das Prozesssystem das zentrale Element darstellt (vgl. Bild 3-1). Um das Gesamtsystem Produktentwicklung widerspruchsfrei modellieren zu können und die Elemente der Subsysteme einfach miteinander vernetzen zu können wird jedes Subsystem mit einer einheitlichen systemischen Modellierungssprache abgebildet.

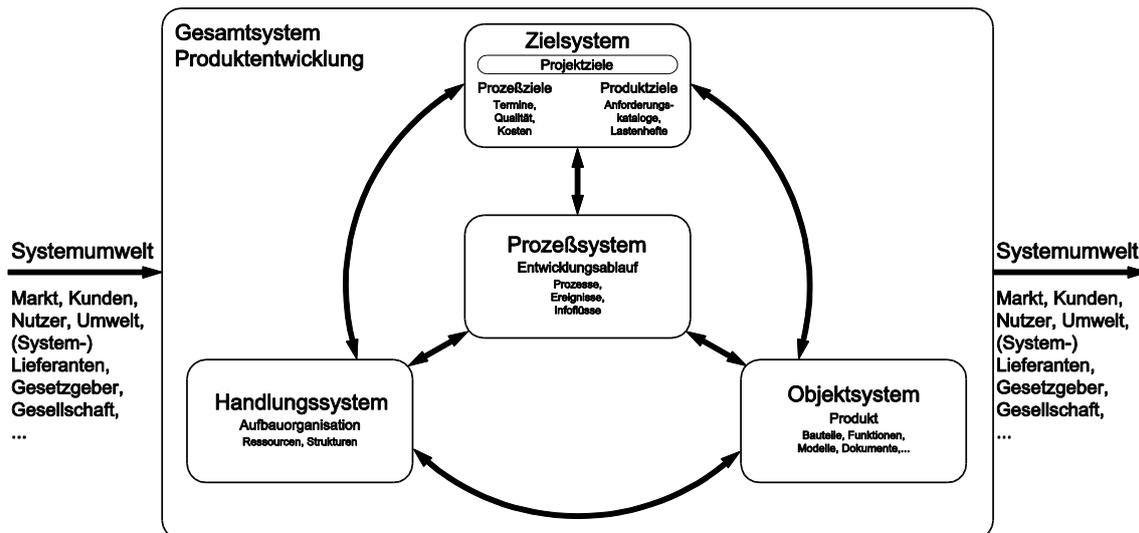


Bild 3-1 ZOPH-Modell der Produktentwicklung nach NEGELE (in Anlehnung an [Neg06], S. 139)

Die Beschreibung, welche Ergebnisse und unter welchen Randbedingungen diese Ergebnisse in einem Entwicklungsvorhaben oder (Teil-)Prozess erreicht werden sollen, ist Inhalt des **Zielsystems**. Es beschreibt alle Ziele bzw. Anforderungen und deren Zusammenhänge die für das Erarbeiten der richtigen Lösung relevant sind ohne die Lösung selbst zu beschreiben. Die beiden für diese Arbeit am bedeutendsten Kategorien von Zielen sind die Produktziele und die Prozessziele. Die Produktziele beschreiben die Anforderungen an das zu entwickelnde Produkt. Die Prozessziele betreffen den Entwicklungsablauf sowie die zeitlich-inhaltlichen, finanziellen und qualitätsbezogenen Erwartungen.

Im **Objektsystem** sind die Voraussetzungen, zu bearbeitenden Gegenstände und die Ergebnisse der Entwicklungstätigkeiten bzw. -prozesse beschrieben. Es beinhaltet also sowohl das (endgültige) Produkt als auch alle Artefakte, die die verschiedenen Evolutionsstufen des Produkts beschreiben.

Die während der Produktentwicklung auszuführenden Tätigkeiten, Prozesse oder Handlungen werden im **Prozesssystem** beschrieben. Ferner werden die verschiedenen Abhängigkeiten, Flüsse und Verknüpfungen zwischen den Prozesselementen abgebildet. Zusammenfassend werden die im Prozesssystem beschriebenen Tätigkeiten unter Zuhilfenahme der Elemente des Handlungssystems (s. u.) entsprechend der Vorgaben aus dem Zielsystem ausgeführt. Ihre Eingangs- und Ausgangsgrößen sind die Elemente des Objektsystems.

Das **Handlungssystem** beschreibt die sachlichen und personellen Ressourcen der Produktentwicklung als auch deren Abhängigkeiten in Form der Aufbauorganisation. Die Elemente des Handlungssystems können sowohl einzelne Aktionsträger oder deren Zusammenfassung zu einer Einheit, als auch sachliche Hilfsmittel zur Ausführung einer Tätigkeit sein.

Bewertung: Das ZOPH-Modell liefert eine sehr gute prozessorientierte Strukturierung des Entwicklungsprozesses und klassifiziert zudem die Bestandteile der Subsysteme. Durch die Verwendung der einheitlichen Modellierungssprache wird die Transparenz für die an der Entwicklung beteiligten Akteure gefördert. Das Modell fokussiert die Analyse operativ bearbeiteter Entwicklungsprozesses und nicht deren vorausschauende Planung und Anpassung für spezifische Entwicklungsaufgaben.

3.1.4 Integrierte Produkterstellung nach EHRENSPIEL

Die Integrierte Produkterstellung¹⁸ (IPE) nach EHRENSPIEL [Ehr07] ist ein durch die Systemtheorie inspirierter Ansatz mit dem Anspruch die Ziele und Handlungsmöglichkeiten aller an der Erstellung eines Produktes beteiligten Fachdisziplinen zu einem Gesamtsystem zu integrieren. Auf dieser Basis sollen allgemein anwendbare, organisatorische und sachgebundene Methoden der Produkterstellung vereinheitlicht und konsolidiert bzw. komplementär angewendet werden. Die IPE bezieht sich vom Anspruch her auf den gesamten Produktentstehungsprozess. Ihre konkrete Ausgestaltung stützt sich insbesondere auf EHRENSPIELS Erfahrungen in den Bereichen Entwicklung und Konstruktion sowie z.T. im Bereich Produktion.

Das modelltechnische Fundament der IPE bildet die Vorstellung, dass die Produkterstellung ein aus verschiedenen Ziel-, Handlungs- und Sachsystemen bestehendes Gesamtsystem ist.

Als **Zielsystem** bezeichnet EHRENSPIEL eine strukturierte Menge der Zielvorgaben. Es sind Anforderungen und deren Verknüpfungen. Sie können z.B. zeitlich oder hierarchisch geordnet sein und sind die Grundlage für die Bewertung des entwickelten Sachsystems (s. u.) oder des Handlungsprozesses.

Das **Sachsystem** sind die technischen Gebilde, die aus dem ingenieurmäßigen Wirken hervorgehen. Sie müssen nicht immer materieller Natur sein, sondern können bspw. wie bei Software auch immaterieller Natur sein. Ein oder mehrere Sachsysteme sind das Objekt des Handlungssystems (s. u.) d.h. sie werden in diesem transformiert oder gehen aus diesem hervor.

Das **Handlungssystem** beinhaltet sämtliche Elemente (bei EHRENSPIEL Aktivitäten genannt) mit denen ein Sachsystem entsprechend einem Zielsystem generiert wird. Die Elemente können sowohl Sachmittel, als auch Menschen oder Handlungen sein.

Die für die Produktentwicklung besonders wichtigen Ziel-, Handlungs- und Sachsysteme sind: das Sachsystem Produkt, das Sachsystem Produktion, das Ziel- und Handlungssystem Mensch, das Ziel- und Handlungssystem Entwicklung und Konstruktion

¹⁸ Der Begriff „Produkterstellung“ ist mit dem in dieser Arbeit bisher verwendeten Begriff der „Produktentstehung“ gleich zu setzen. Dem ursprünglichen Wortlaut EHRENSPIELS Rechnung tragend wird in diesem Abschnitt der Begriff „Produkterstellung“ verwendet.

sowie das Ziel- und Handlungssystem Kunde. Diese bilden gemeinsam mit den Ziel- und Handlungssystemen weiterer an der Produkterstellung beteiligter Bereiche (Bereich X) das Gesamtsystem der integrierten Produkterstellung. Dieses setzt letztendlich den Prozess der Produkterstellung um (vgl. Bild 3-2).

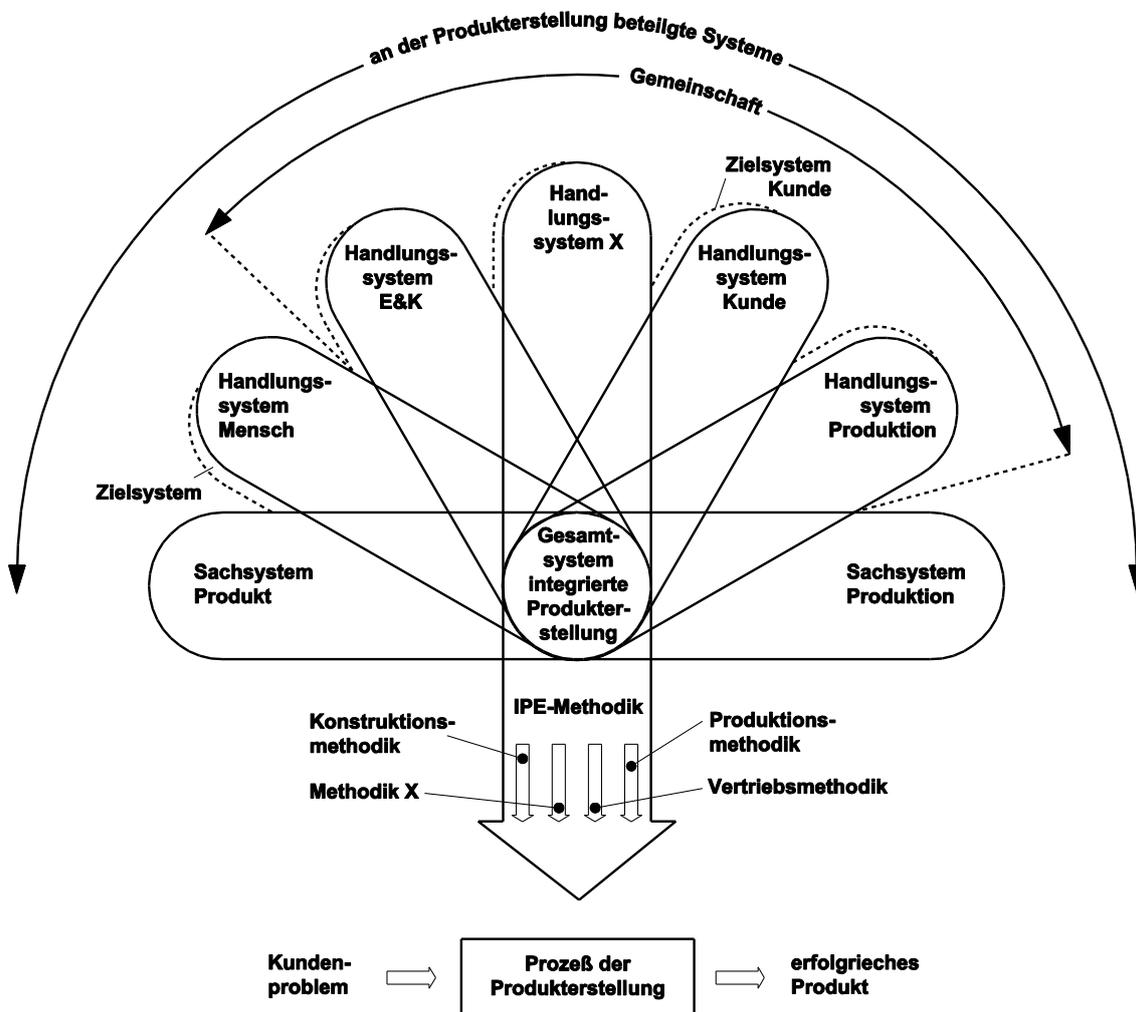


Bild 3-2 Systemische Bestandteile des Gesamtsystems der integrierten Produkterstellung nach EHRENSPIEL (in Anlehnung an [Ehr07])

Bewertung: Die Integrierte Produkterstellung nach EHRENSPIEL gibt eine gute Übersicht über die an der Entwicklung interdisziplinärer technischer Systeme beteiligten Fachdisziplinen und ihr grundsätzliches Zusammenspiel. Das Modell fokussiert dabei den Produktentstehungsprozess in Gänze wobei die einzelnen Elemente der Subsysteme und ihre Wechselwirkungen nur schematisch dargestellt werden.

3.1.5 iPeM

Das integrierte Produktentwicklungs-Modell, kurz iPeM, ist ein von ALBERS [Alb10] entwickeltes systemisches Gesamtmodell der Produktentwicklung. ALBERS selbst be-

zeichnet es als Metamodell der Produktentwicklung mit dem Entwicklungsprozesse einheitlich beschrieben, analysiert und verbessert werden sollen. Die Produktentwicklung wird in die drei Grundsysteme, Ziel-, Objekt- und Handlungssystem, unterteilt (vgl. Bild 3-3).

Das **Zielsystem** (System of Objectives) beschreibt die angestrebten Eigenschaften des zu entwickelnden Produkts und von Zwischenergebnissen für dessen Entwicklung. Es beschreibt folglich nicht das Produkt selbst, sondern die Produktziele, deren Zusammenhänge, Randbedingungen und Ursprünge. Während des Entwicklungsprozesses wird es permanent erweitert und verfeinert.

Das zu entwickelnde Produkt und alle mit ihm zusammenhängenden Ergebnisse werden im **Objektsystem** (System of Objects) beschrieben. Dies können sowohl physische Objekte wie z.B. Prototypen oder Teststände, als auch nicht physische Objekte wie z.B. Zeichnungen oder Simulationsmodelle sein. Entsprechend verlassen einige Objekte des Objektsystems am Ende des Entwicklungsprozesses das Unternehmen und andere nicht.

Der Entwicklungsprozess kann als die Überführung des Zielsystems in ein entsprechendes Objektsystem verstanden werden. Eine direkte Verbindung zwischen Ziel- und Objektsystem besteht nicht. Die Instanz die dies leistet ist das **Handlungssystem** (Operation System). Es ist ein soziotechnisches System, das in insgesamt vier Subsysteme unterteilt werden kann. Im ersten werden im Produktlebenszyklus abzuarbeitende Aufgabenkomplexe, sogenannte *Aktivitäten* der Produktentwicklung (Activities of Product Engineering), zusammengefasst. Diese sind nicht als top-down abzuarbeitendes Phasenmodell zu verstehen, sondern werden im Entwicklungsprozess z.T. wechselweise und iterativ durchlaufen. In jedem Aufgabenkomplex kommt die *allgemeine Problemlösungslogik* (Activities of Problem Solving) SPALTEN¹⁹ zur Anwendung (vgl. [ABM+05], S. 5). Mit Hilfe der sich aus den Aufgabenkomplexen und den einzelnen Schritten der Problemlösungslogik ergebenden Matrix (Activity Matrix) kann der logische Ablauf eines Entwicklungsprozesses dargestellt werden. Dieser wird mit den im dritten Subsystem beschriebenen *Ressourcen* (System of Resources) wie z.B. Personal, Maschinen, Finanzmittel, Informationen oder Verbrauchsmaterial realisiert. Die genauen Anfangs- und Endzeitpunkte der einzelnen Prozessschritte werden im vierten Subsystem, dem *Phasenmodell* (Phase Model), dargestellt. Das Phasenmodell beschreibt somit letztendlich einen spezifischen Entwicklungsprozess und kann für den Aufbau von Erfahrungswissen sowie die Klassifikation oder den Vergleich unterschiedlicher Entwicklungsprozesse verwendet werden.

¹⁹ SPALTEN ist ein generischer Problemlösungsprozess für technische Problemstellungen. Die einzelnen Buchstaben der Abkürzung stehen für: Situationsanalyse (Situation Analysis), Problemeingrenzung (Problem Containment), Alternative Lösungssuche (Search for Alternative Solutions), Lösungsauswahl (Selection of Solutions), Tragweitenanalyse (Analysis of the Level of Fulfillment), Entscheiden/Umsetzen (Make Decision/Implement) und Nacharbeiten/Lernen (Recapitulate/Learn).

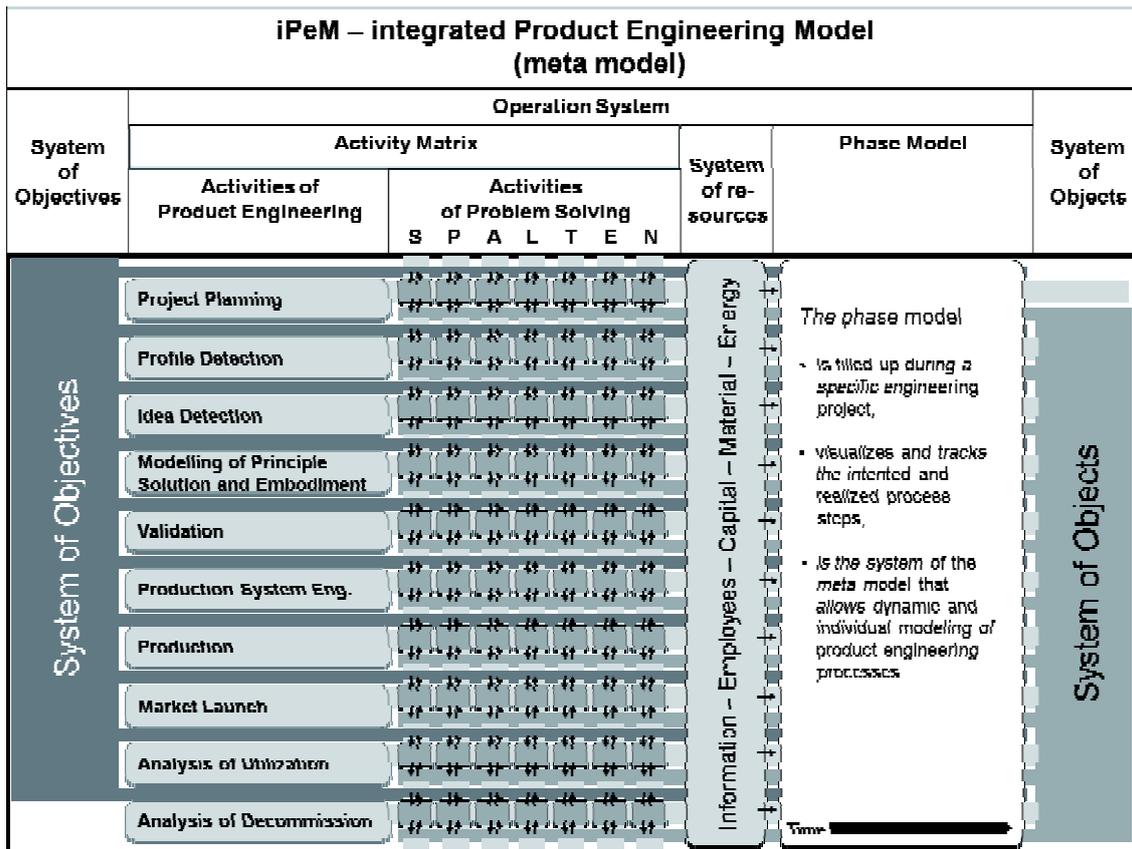


Bild 3-3 Das integrierte Produktentwicklungs-Modell nach ALBERS [Alb10, S. 7]

Bewertung: Das iPeM liefert eine sehr gute Strukturierung des Entwicklungsprozesses in Subsysteme und klassifiziert deren Inhalte. Es werden die grundsätzlichen Aufgabenkomplexe der Produktentwicklung und eine allgemeine Problemlösungsstrategie bereitgestellt, wobei eine problemspezifische Ausprägung möglich ist, aber nicht aktiv unterstützt wird. Das Phasenmodell gibt eine gute Übersicht über die prozesstechnischen Inhalte der Entwicklung. Hilfsmittel zur Darstellung der modelltechnischen Inhalte werden nicht bereitgestellt.

3.2 Modellierungstechniken

Nachfolgend werden Modellierungstechniken zur Beschreibung von Prozessen und fachdisziplinübergreifenden Systemmodellen dargestellt. Die Modellierungstechniken für Prozesse sollen die Darstellung und Planung des Entwicklungsablaufs sowie der darin eingesetzten Ressourcen und Entwicklungsobjekte ermöglichen. Die Modellierungstechniken für fachdisziplinübergreifende Systemmodelle sollen als Informationsbasis über alle Aspekte des zu entwickelnden Systems und deren Zusammenhänge dienen.

3.2.1 Modellierungstechniken für Prozesse

Die Modellierung von Prozessen unterstützt zum einen die Planung des Entwicklungsablaufs und dient zum anderen als Basis für die Koordination, Kommunikation und Kooperation der an der Entwicklung beteiligten Entwickler. Nachfolgend werden fünf Modellierungstechniken für Geschäftsprozesse in wertschöpfenden Unternehmen vorgestellt, die diese Aufgaben mit unterschiedlichem Schwerpunkt unterstützen.

3.2.1.1 ARIS

ARIS steht für „Architektur Integrierter Informationssysteme“ und ist eine Methode zur Modellierung von Unternehmen mit dem ursprünglichen Ziel aufbauend auf der Planung und Modellierung der Geschäftsprozesse eine geeignete Business-IT aufzubauen. Sie wird durch das kommerziell verfügbare Softwarewerkzeug „ARIS Toolset“ der IDS Scheer AG umgesetzt, kann aber auch unabhängig von davon angewendet werden. Das Gesamtmodell eines Unternehmens setzt sich dabei aus vier Sichten zusammen [GPW09, S. 278f.], [SS06, S. 609ff.], [Sei06, S. 11 ff.]:

In der Sicht **Funktionen** werden die Geschäftsprozesse (in ARIS Funktionen genannt) sowohl hierarchisch als auch ihrer Reihenfolge nach geordnet. Eine Funktion beschreibt die dem Unternehmenszweck entsprechende Transformation einer Eingangsgröße in eine Ausgangsgröße. Die Ausführung einer Funktion kann dabei ein (Geschäfts-)Ziel unterstützen oder durch dieses gesteuert werden. Entsprechend werden die Ziele ebenfalls in der Sicht Funktionen in Form von Zieldiagrammen abgebildet.

Die Aufbauorganisation des Unternehmens wird in der Sicht **Organisation** abgebildet. Sie beschreibt die verschiedenen Organisationseinheiten und diesen zur Verfügung stehende Ressourcen (z.B. Personal, Budget, Betriebsmittel etc.).

Funktionen auslösende oder beendende Ereignisse sowie physischen und nichtphysischen Eingangs- und Ausgangsobjekte der einzelnen Funktionen werden in der Sicht **Daten** mit Hilfe von Entity-Relationship-Diagrammen dargestellt. Ereignisse beschreiben dabei Zustände die den weiteren Prozessverlauf beeinflussen. Die physischen und nichtphysischen Objekte werden durch Entitytypen, ihre Eigenschaften durch Attributtypen beschrieben.

Die **Steuerungssicht** integriert die vorstehend genannten Sichten. Sie kann entweder als Vorgangskettendiagramm oder (erweiterte) ereignisgesteuerte Prozesskette (eEPK) modelliert werden. Bild 3-4 zeigt ein Beispiel für eine eEPK nach ARIS. Funktionen und Ereignisse sind wechselweise angeordnet und über Pfeile miteinander verbunden. Die die Funktionen ausführenden Organisationseinheiten sind auf der linken Seite angeordnet, die zur Ausführung eingesetzten Datenelemente auf der rechten.

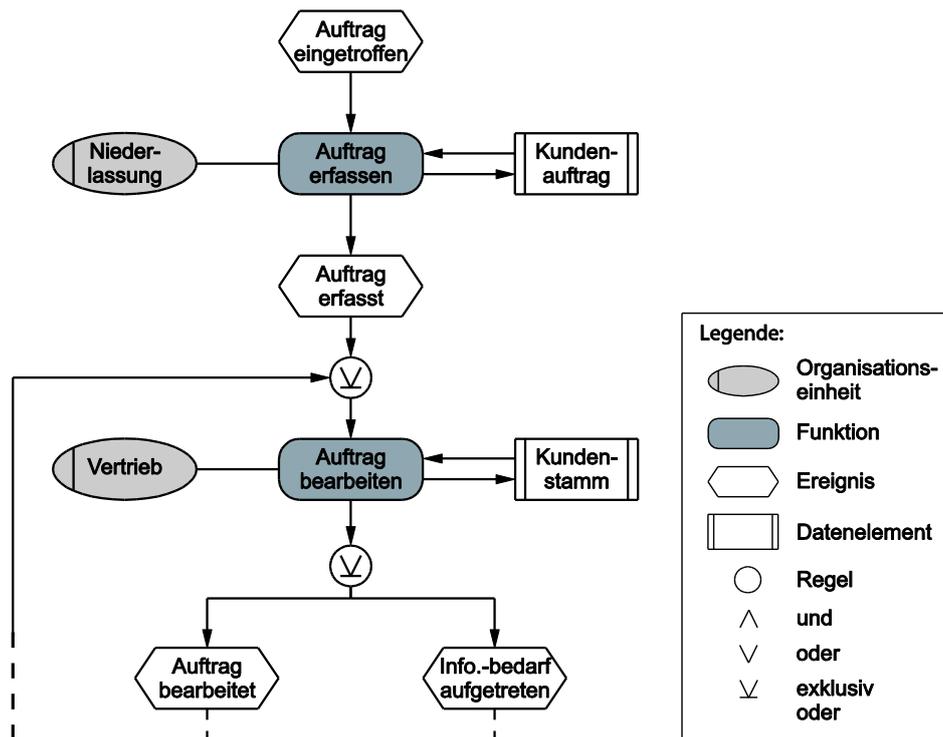


Bild 3-4 Beispiel für eine nach ARIS modellierte erweiterte ereignisgesteuerte Prozesskette (eEPK) [GPW09, S. 278]

Bewertung: ARIS unterstützt die Modellierung der Aufbauorganisation und der Geschäftsabläufe eines Unternehmens durch eine Vielzahl problemangepasster Modellierungskonstrukte. Es können sowohl deterministische als auch nichtdeterministische Abläufe modelliert werden. Der Schwerpunkt liegt auf der Darstellung und Analyse bestehender Prozesse, um darauf aufbauend eine unterstützende Business-IT aufzubauen. Die Vorausplanung und dynamische Anpassung des Prozesses ist nicht Fokus von ARIS.

3.2.1.2 OMEGA

Ziel der objektorientierten Methode zur Geschäftsprozessmodellierung und -analyse OMEGA ist ein intuitiv verständliches, vollständiges Modell der Ablauforganisation. Hierzu werden die Aufbau- und die Prozessorganisation in einem Diagramm abgebildet. Aus diesem lassen sich formale Prozessspezifikationen für Workflow- und Produktdatenmanagementsysteme ableiten. Die Symbolik von OMEGA umfasst die Konstrukte: Geschäftsprozesse und Aktivitäten, Organisationseinheiten, externe Objekte, Bearbeitungsobjekte, technische Ressourcen und Kommunikationsbeziehungen (siehe Bild 3-5). [Fah95], [GPW09, S. 281 ff.]

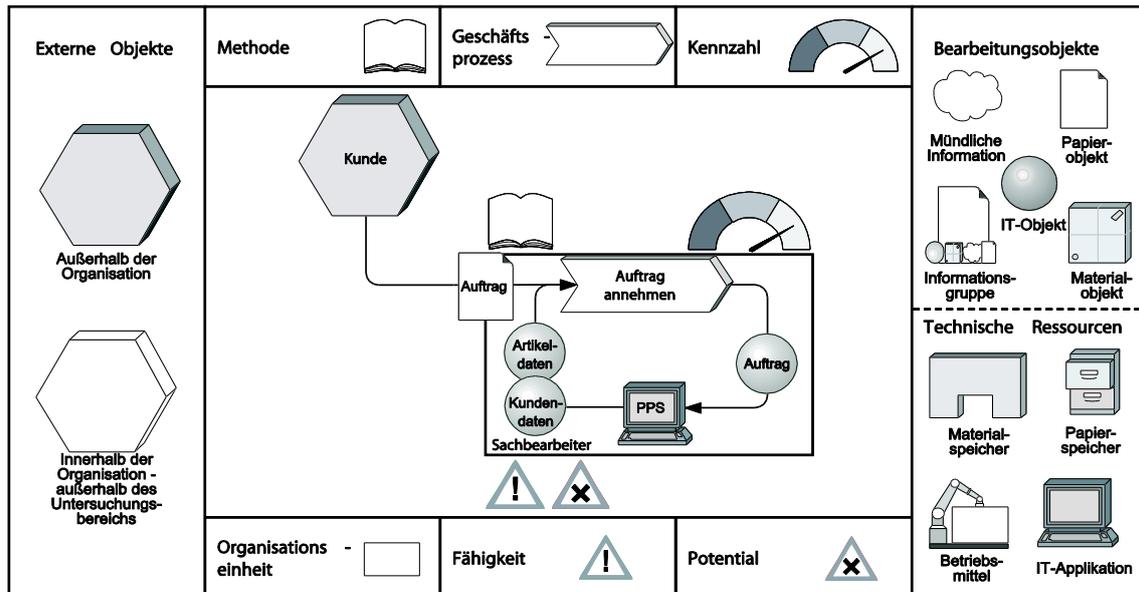


Bild 3-5 Überblick über die Konstrukte von OMEGA [GPW09, S. 282]

Als **Geschäftsprozess** wird eine Abfolge logisch zusammenhängender Aktivitäten bezeichnet. Er dient der Veränderung eines Objekts (Transformation) oder dem Erbringen eines Ergebnisses. Anfang und Ende sind durch Auslöser bzw. Input und Ergebnis bzw. Output fest definiert. Geschäftsprozesse können durch untergeordnete Geschäftsprozesse detailliert werden bzw. einem übergeordneten Geschäftsprozess zugeordnet sein.

Ausgeführt wird der Geschäftsprozess durch **Organisationseinheiten**. Sie repräsentieren eine Stelle in der Aufbauorganisation des Unternehmens (z.B. Abteilung, Person, Rolle etc.). Jede Organisationseinheit kann in einem Geschäftsprozess mehrfach verwendet werden.

Nicht zur betrachteten Organisation gehörende Stellen werden als **externe Objekte** modelliert. Hierzu gehören bspw. externe Personen, Firmen, Institutionen etc.

Die Input- und Output-Größen eines Geschäftsprozesses sind die **Bearbeitungsobjekte**. Sie werden durch den Geschäftsprozess transformiert bzw. erzeugt. Das Output-Objekt eines Geschäftsprozesses ist i.d.R. das Input-Objekt eines nachgelagerten Geschäftsprozesses. Neben den Geschäftsprozessen können technische Ressourcen und externe Objekte ebenfalls Bearbeitungsobjekte empfangen und abgeben. In OMEGA werden IT-Objekte, Papierobjekte, mündliche Informationen, Materialobjekte und Informationsgruppen (Zusammenfassung mehrerer Informationsobjekte) unterschieden.

Die zur Durchführung eines Geschäftsprozesses benötigten Hilfsmittel werden als **technische Ressourcen** bezeichnet. Hierzu gehören IT-Applikationen, Betriebsmittel, Papierspeicher und Materialspeicher. Technische Ressourcen können wie z.B. Organisationseinheiten ebenfalls mehrfach im Geschäftsprozess verwendet werden.

Die Geschäftsprozesse, externe Objekte und technische Ressourcen werden über **Kommunikationsbeziehungen** miteinander verbunden. Sie übermitteln die Bearbeitungsob-

jekte und machen so die Informations- und Materialflüsse im Prozess sichtbar. Jede Kommunikationsbeziehung hat dabei immer genau einen Sender und genau einen Empfänger.

Bewertung: OMEGA strukturiert und klassifiziert die einzelnen in einem Geschäftsprozess eingesetzten Objekte und stellt intuitiv verständliche Symbole zu deren Darstellung bereit. Spezifische Eigenschaften eines Prozessschrittes können über Kennzahlen abgebildet werden. Die Modellierungstechnik fokussiert die Abbildung und Analyse bestehender Prozesse sowie die darauf Aufbauende Prozessverbesserung und Vorgabe von Sollprozessen. Eine detaillierte Termin- oder Kostenplanung wird nicht explizit unterstützt.

3.2.1.3 SADT

Die Structured Analysis and Design Technique (SADT) von ROSS dient der Beschreibung, dem Verständnis, der Bearbeitung und der Nachverfolgung von Problemstellungen beim Systementwurf. Die Technik folgt dem Gedanken, dass der menschliche Verstand jedes noch so komplexe Problem erfassen und lösen kann wenn es in einfachere Teile zerlegt wird. Die Systembeschreibung erfolgt bei SADT mit Hilfe eines Datenmodells (Datagramm) und eines Aktivitätenmodells (Aktigramm). Für die Modellierung von Geschäftsprozessen eignet sich insbesondere das Aktigramm (vgl. Bild 3-6). Die Prozesse werden mit Hilfe der Konstrukte *Aktivität*, *Eingabe-/Ausgabedaten*, *Steuerungsdaten* und *Mechanismen* beschrieben. Die Aktivität entspricht der in einem Prozessschritt ausgeführten Tätigkeit. Einzelnen Aktivitäten sind über ihre Eingabe- und Ausgabedaten zu Prozessfolgen verbunden. Dies können sowohl Informations- als auch Materialobjekte sein. Für die Ausführung der Aktivitäten benötigte Hilfsmittel werden als Mechanismen modelliert. Die Art der Ausführung ist durch die Steuerungsdaten beschrieben. Welche Ereignisse eine Aktivität auslösen oder beenden wird in einem Aktigramm nicht abgebildet. SADT sieht vor nicht mehr als sechs Aktivitäten auf einmal abzubilden. Aus diesem Grund entstehen bei der Modellierung nach SADT oft tiefgegliederte Teilmodelle, die hierarchisch miteinander verbunden sind. [Ros85, S. 25ff.], [GPW09, S. 276 f.]

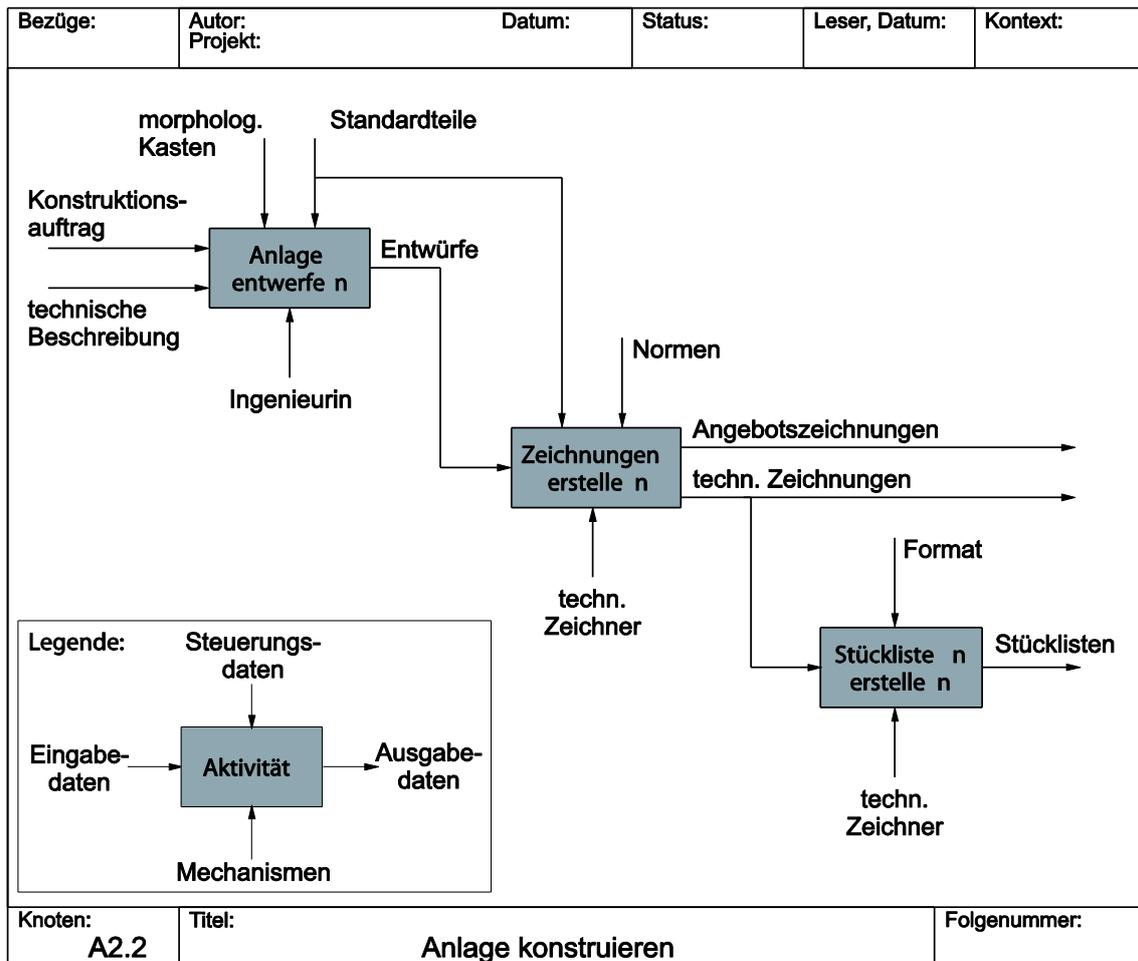


Bild 3-6 Beispiel für ein Aktigramm nach SADT [GPW09, S. 277]

Bewertung: SADT bildet die logische Abfolge von Geschäftsprozessen ab. Die Darstellung größerer Prozesse wird durch die Vorgabe erschwert nicht mehr als sechs Aktivitäten auf einmal abzubilden. Die geringe Anzahl an Symbolen und die u.U. tief gegliederten Prozessmodelle erschweren deren intuitive Erfassbarkeit.

3.2.1.4 IDEF

IDEF (**I**ntegrated **D**efinition) ist keine singuläre Modellierungssprache, sondern eine Gruppe von Modellierungssprachen zur Unterstützung des Systems Engineering. Die Ursprünge von IDEF liegen im ICAM-Projekt²⁰ der US-Luftwaffe Ende der 1970er Jahre. Angefangen mit der Veröffentlichung des NDEF0 Standards im Jahr 1993 sind in den vergangenen Jahren weitere Standards bis hin zu IDEF14 hinzugekommen, die kontinuierlich weiterentwickelt werden. Für die Modellierung von Geschäftsprozessen sind insbesondere die Standards IDEF0 und IDEF3 relevant. [CE05, S. 74ff.], [Gad10, S. 73]

²⁰ Integrated Computer-Aided-Manufacturing

IDEF0 baut auf SADT auf und nutzt ebenfalls *Aktivitäten, Eingabe-/Ausgabedaten, Steuerungsdaten* und *Mechanismen* als Modellierungskonstrukte (vgl. Abschnitt 3.2.1.3). Hinzugekommen ist ein unterhalb der Aktivität und von ihr wegführender Pfeil, mit dem auf ein weiteres Modell zur Detaillierung der Aktivität verwiesen bzw. dieses aufgerufen wird²¹. Der IDEF0 Standard kann genau wie SADT die logische Abfolge der Prozessschritte abbilden, nicht aber deren zeitlich-logische Abhängigkeiten. Die Darstellung der zeitlich-logischen Abhängigkeiten von Aktivitäten wird durch den IDEF3 Standard ermöglicht. Er verwendet eine weitestgehend eigene Symbolik, kann aber auf gemäß IDEF0 modellierte Objekte referenzieren. Den Kern bilden weiterhin die vom IDEF0 Standard übernommenen Tätigkeitsboxen. Diese werden über Kontrollflüsse miteinander verbunden. Die Kontrollflüsse können über Verbindungsboxen an verschiedene Bedingungen geknüpft werden, wodurch z.B. nachgelagerte Tätigkeiten als gleichzeitig oder zeitlich unabhängig voneinander modelliert werden können. Bild 2-1 zeigt ein Beispiel für einen gemäß IDEF3 modellierten Geschäftsprozess. [CE05, S. 74ff.], [Gad10, S. 73ff.]

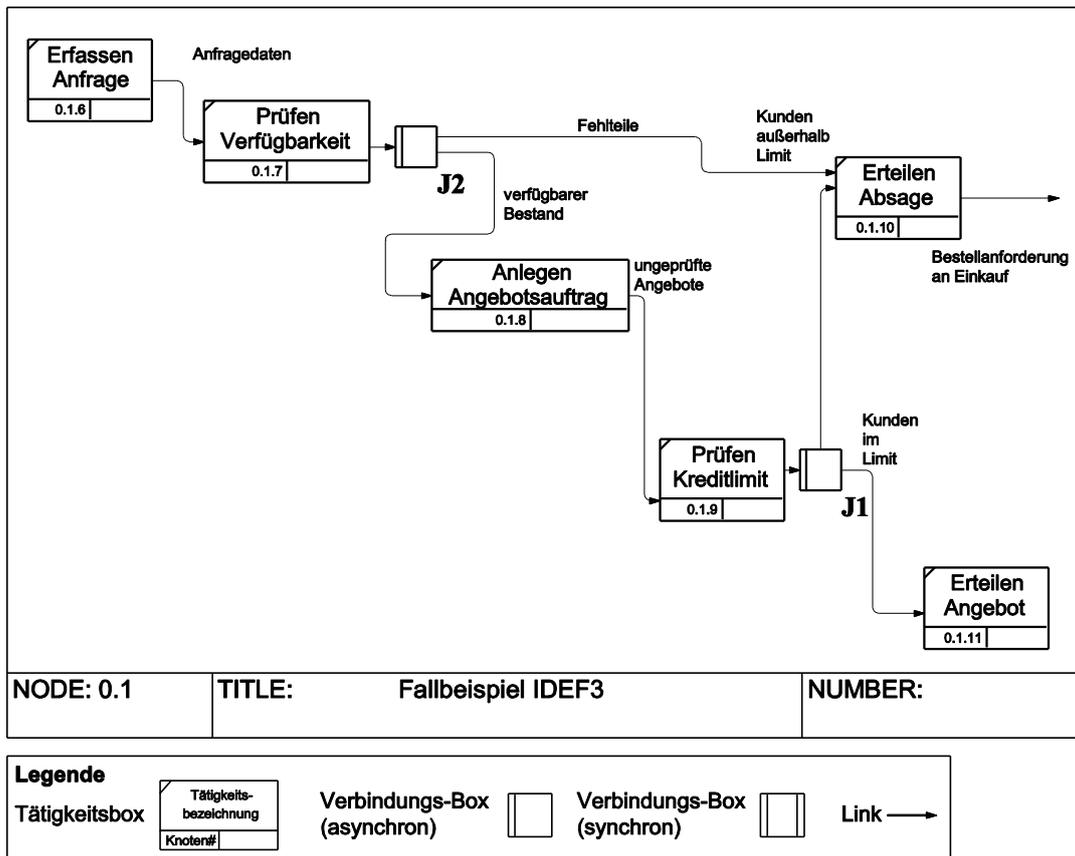


Bild 3-7 Beispiel für ein IDEF3-Diagramm nach [Gad10, S. 75]

²¹ Auf Grund des geringen Unterschiedes zu SADT wird an dieser Stelle auf die Darstellung der Symbolik gemäß IDEF0 verzichtet.

Bewertung: IDEF0 und IDEF3 leiden unter den gleichen Darstellungsschwächen wie SADT. Die Abbildung der zeitlich-logischen Abhängigkeiten durch IDEF3 ermöglicht die Abbildung deterministischer und nicht-deterministischer Abläufe. Zudem unterstützt IDEF die standardisierte Darstellung von für Planungen und Geschäftsprozesssimulationen benötigten Informationen. Konkrete Verfahren werden allerdings nicht bereitgestellt.

3.2.1.5 Netzplantechnik

Die Netzplantechnik hat ihren Ursprung im Projektmanagement und wird heute in verschiedenen Ausprägungen verwendet. Grundsätzlich werden in einem Netzplan *Vorgänge*, *Ereignisse* und *Anordnungsbeziehungen* unterschieden. Ein Vorgang ist eine zeitbehaftete Tätigkeit im Projekt. Ein Ereignis ist ein bestimmter Zustand im Projektablauf. Die Anordnungsbeziehungen definieren die zeitlichen, personellen und fachlichen Abhängigkeiten zwischen den einzelnen Vorgängen. Hinsichtlich des möglichen Durchlaufens eines Netzplans werden deterministische und stochastische Netzpläne unterschieden. Bild 3-8 zeigt auf der linken Seite Beispiele für deterministische Netzpläne (MPM-Netzplan²² und CPM-Netzplan²³), auf der rechten für stochastische (GERT-Netzplan²⁴ und PETRI-Netz²⁵) [Bur08, S. 245ff.]

Bei den **deterministischen Netzplänen** sind die im Netzplan beschriebenen Abläufe eindeutig. Dies bedeutet, dass alle Pfade des Netzplans durchlaufen werden. Beispiele für deterministische Netzplantechniken sind Vorgangspfeil-Netzpläne, Ereignisknoten-Netzpläne oder Vorgangsknoten-Netzpläne. Bei einem Ereignisknoten-Netzplan werden die Ereignisse in Form von Knoten dargestellt. Die Vorgänge zwischen den Ereignissen treten in den Hintergrund. Sie ergeben sich aus den sie begrenzenden Ereignissen. Ihnen gegenüber stehen die Vorgangsknoten-Netzpläne. Bei dieser Darstellungsform werden die Vorgänge als Knoten dargestellt. Die Pfeile zwischen den Knoten sind die logischen Anordnungsbeziehungen der Vorgänge, die deren Reihenfolge abbilden. Die dritte Dar-

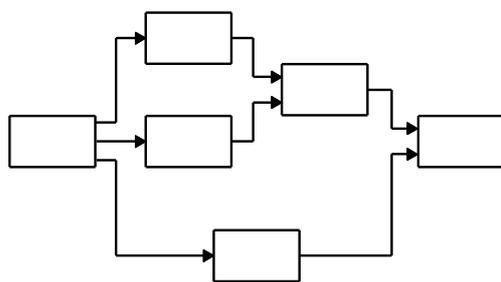
²² Bei einem MPM (Meta-Potential-Methode)-Netzplan werden die Vorgänge als Rechtecke und ihre Anordnungsbeziehungen als Pfeile dargestellt (Vorgangsknoten-Netzplan). Ereignisse werden nicht abgebildet. Sie entfallen bei Vorgangsknoten-Netzplänen. [Bur08, S. 247 ff.]

²³ Der CPM (Critical Path Method)-Netzplan ist ein Vorgangspfeil-Netzplan, bei dem die Vorgänge als Pfeil dargestellt werden und die den Vorgang auslösenden bzw. beendenden Ereignisse als kreisförmige Knoten. Anordnungsbeziehungen sind in Vorgangspfeil-Netzplänen nicht dargestellt. [Bur08, S. 247 ff.]

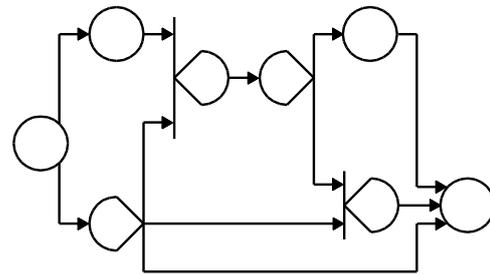
²⁴ GERT (Graphical Evaluation and Review Technique)-Netzpläne arbeiten mit Pfeilen zur Darstellung von Vorgängen und verschiedenen Symbolen für Knoten zur Darstellung von Ereignissen. Die Eingänge der Vorgänge können als disjunktive oder konjunktive Mengen, ihre Ausgänge als deterministische oder probabilistische Größen modelliert werden. [Bur08, S. 248]

²⁵ Bei Petri-Netzen werden die Zustände eines Prozesses über Stellen, Zustandsübergänge durch Transitionen dargestellt. Das zeitlich-dynamische Verhalten eines Prozesses wird durch sogenannte Marken abgebildet. [GPW09, S. 277]

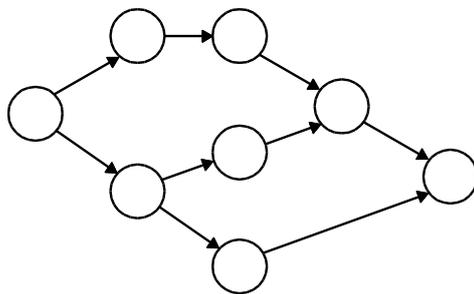
stellungsform sind die Vorgangspfeil-Netzpläne. Hierbei werden die Vorgänge als Pfeile dargestellt und die Ereignisse als Knoten.



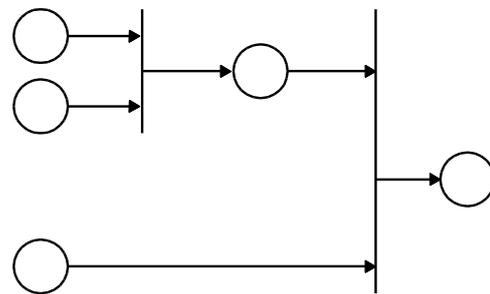
MPM-Netzplan



GERT-Netzplan



CPM-Netzplan



Petri-Netz

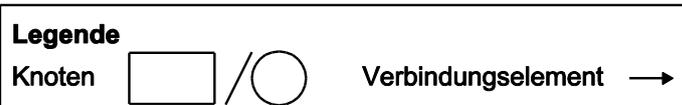


Bild 3-8 Schematische Darstellung verschiedener Netzpläne nach [Bur08, S. 246]

Bei den **stochastischen bzw. probabilistischen Netzplänen** gibt es mehrere alternative Wege durch den Netzplan. Abhängig von spezifischen Wahrscheinlichkeitswerten kann zwischen verschiedenen Möglichkeiten für den weiteren Verlauf gewählt werden. Solche Netzpläne werden auch als Entscheidungsnetzpläne bezeichnet. Bei ihnen müssen nicht zwangsläufig alle Pfade des Netzplans durchlaufen werden. Beispiele stochastischer Netzplantechniken sind die Methode Graphical Evaluation and Review Technique oder Petri-Netze.

Bewertung: Netzpläne ermöglichen abhängig von der Art des Netzplans die Planung und Simulation von Prozessverläufen. Sie ermöglichen bspw. Ablauf-, Termin-, Ressourcen- oder Kostenplanungen. Da sie nicht speziell auf Geschäftsprozesse fokussiert sind, sind ihre Konstrukte und Darstellungsmöglichkeiten abstrakt und wenig intuitiv verständlich gehalten.

3.2.2 Sprachen zur Beschreibung des Systemmodells mechatronischer Systeme

Um während der Entwicklung den Überblick über das zu entwickelnde mechatronische System zu behalten bedarf es einer Informationsbasis, die alle Aspekte des Systems und insbesondere deren Zusammenhänge fachdisziplinunabhängig beschreibt. Zu diesem Zweck sind in den letzten Jahren verschiedene Modellierungssprachen entstanden mit denen Struktur und Funktionsweise eines mechatronischen Systems in einem ganzheitlichen Systemmodell beschrieben werden. Die verschiedenen Sprachen setzen dabei unterschiedliche Schwerpunkte. Nachfolgend werden fünf Modellierungssprachen vorgestellt, die als Informationsbasis für die effiziente Entwicklung mechatronischer Systeme dienen können.

3.2.2.1 OMG SysML™

Die OMG Systems Modeling Language (OMG SysML™) ist eine semiformale, objektorientierte, grafische Modellierungssprache zur generellen Spezifikation, Modellierung, Analyse, Verifikation und Validierung technischer Systeme. Sie wurde Anfang 2007 zur Unterstützung des Systems Engineering gemeinsam von der Object Management Group, Inc. (OMG) und dem International Council on Systems Engineering (INCOSE) als Standard in der Version 1.0 veröffentlicht. Die aktuelle Version ist die im Juni 2010 verabschiedete Version 1.2. Die SysML basiert auf der für die Softwareentwicklung definierten Unified Modeling Language (UML) 2.3. Diese wurde für die SysML zum Teil erweitert, zum Teil wurden für die Beschreibung technischer Systeme unnötige Bestandteile der UML gestrichen. Ein Vorteil des Aufbaus auf der UML liegt darin, dass dadurch die für die UML verfügbaren Software-Werkzeuge für die SysML angepasst und genutzt werden können. [OMG10, S. 1], [Wei06, S. 157ff]

Mit der SysML wird ein technisches System anhand der Aspekte Struktur, Anforderungen und Verhalten beschrieben. Diese drei Aspekte werden mit verschiedenen Diagrammen entsprechend Bild 3-9 beschrieben. Die meisten Diagrammtypen sind aus der UML übernommen worden. Neu hinzugekommen sind das Anforderungsdiagramm und das Zusicherungsdiagramm.

In den *Strukturdiagrammen* werden die Blöcke aus denen das zu entwickelnde System besteht, ihre Konfiguration und ihre Eigenschaften beschrieben. Blöcke können sowohl informationsverarbeitende als auch physikalische Elemente des Systems sein, wobei die physikalischen Elemente exklusiv als Einheit bezeichnet werden. Die Blöcke, ihre Generalisierungen, Beziehungen und Assoziationen werden in *Blockdefinitionsdiagrammen* beschrieben. Die Beziehungen zwischen den internen Bestandteilen eines Blocks werden in den *internen Blockdiagrammen* durch Ports, Flüsse und Konnektoren abgebildet. Die Beziehungen zwischen den Eigenschaften verschiedener Blöcke sind separat in *Zusicherungsdiagrammen* abgebildet. Sie ermöglichen die formale Modellierung parametrischer Abhängigkeiten, wie z.B. physikalischer Zusammenhänge. Mit den *Pa-*

ketdiagrammen können die Elemente in hierarchischen Strukturen gruppiert werden. [OMG10, S. 21ff.]

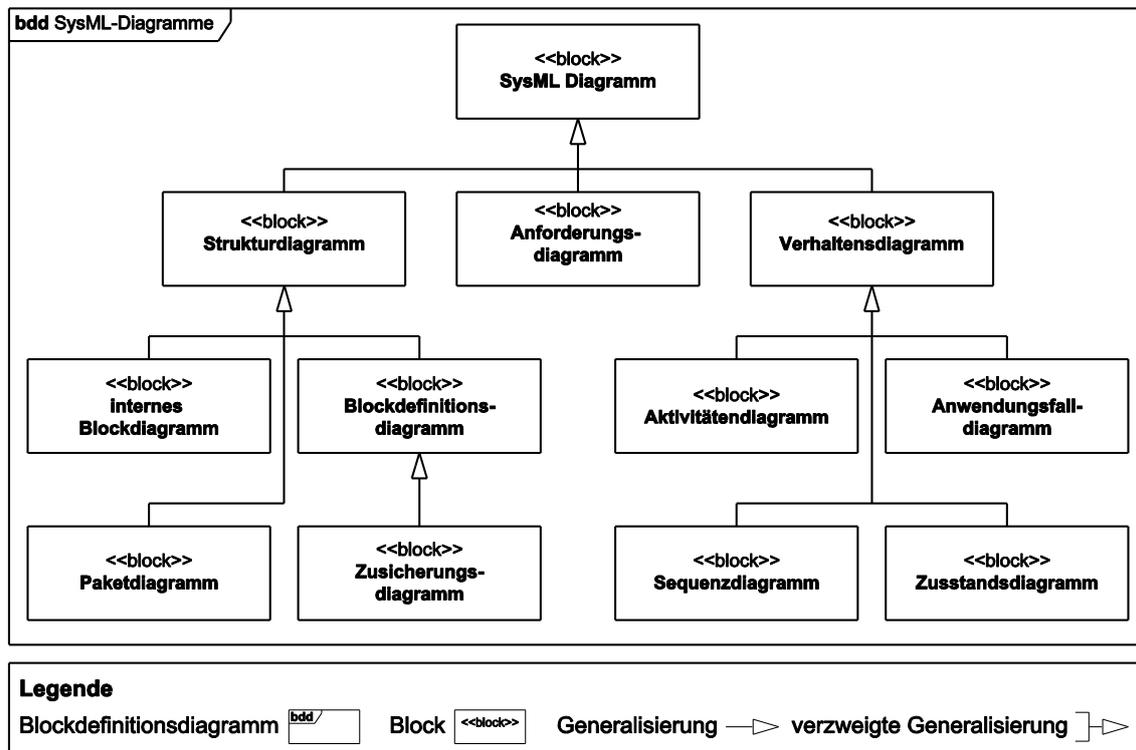


Bild 3-9 Die verschiedenen Diagramme der SysML nach [Wei06, S. 160]

Funktionale als auch nicht funktionale Anforderungen an das zu entwickelnde System werden in den *Anforderungsdiagrammen* beschrieben. Ferner werden deren Zusammenhänge in Form von Ableitungs-, Kopie-, Verfeinerungs-, Erfüllungs-, Enthält-, Prüf- und Verfolgungsbeziehungen spezifiziert. [OMG10, S. 125ff.]

Das Verhalten des zu entwickelnden Systems wird durch die verschiedenen *Verhaltensdiagramme* modelliert. Die Interaktion des Systems nach außen hin, z.B. mit dem Benutzer oder anderen Systemen, wird durch *Anwendungsfalldiagramme* beschrieben. Seine später möglichen Zustände und Zustandsübergänge sind Gegenstand der *Zustandsdiagramme*. Der Funktionsablauf im späteren Betrieb des Systems, inklusive der zugehörigen Ein- und Ausgangsgrößen, wird in *Aktivitätendiagrammen* losgelöst von den die Funktionen ausführenden Elementen beschrieben. Die einzelnen Funktionen können hierbei aus Unterfunktionen zusammengesetzt sein, wodurch eine hierarchische Strukturierung der Funktionen ermöglicht wird. Wie die einzelnen Elemente des Systems interagieren wird in den *Sequenzdiagrammen* beschrieben. [OMG10, S. 75ff.]

Bewertung: Die SysML™ unterstützt die fachdisziplinübergreifende Spezifikation eines technischen Systems. Die Modellierungstechnik zielt nicht speziell auf mechatronische Systeme ab. Sie können zwar mit der SysML™ abgebildet werden, die Verwendung der verschiedenen Konstrukte ist dabei aber teilweise beliebig. Eine intuitive Ver-

ständigkeit der Modellierungstechnik ist auf Grund ihres immensen Umfangs und der teilweisen sehr ähnlichen Konstrukte nicht gegeben.

3.2.2.2 CONSENS

Die semiformale, grafische Modellierungssprache CONSENS (**C**onceptual **D**esign **S**pecification Technique for the **E**ngineering of Complex Systems) [GDT+11] wurde im Rahmen des Sonderforschungsbereichs 614 „Selbstoptimierende Systeme des Maschinenbaus“²⁶ zur intuitiven und fachdisziplinübergreifenden Spezifikation der Prinziplösung selbstoptimierender mechatronischer Systeme entwickelt. Sie setzt auf den Arbeiten von FRANK, GAUSEMEIER und KALLMEYER auf [Fra06], [GEK01], [Kal98] und wurde parallel zum Sonderforschungsbereich 614 im Verbundforschungsprojekt „VireS – Virtuelle Synchronisation von Produktentwicklung und Produktionssystementwicklung“²⁷ weiter formalisiert und in dem Software-Werkzeug „Mechatronic Modeller“ umgesetzt (vgl. [BDT10], [GDK10]).

Die Beschreibung des zu entwickelnden Systems ist gemäß Bild 3-10 in die acht Aspekte: Umfeld, Anwendungsszenarien, Anforderungen, Funktionen, Wirkstruktur, Verhalten, Gestalt und Zielsystem gliedert. Jeder dieser Aspekte und deren Beziehungen untereinander werden rechnerintern in Form eines Partialmodells abgebildet. Ihre Gesamtheit ergibt die ganzheitliche und konsistente fachdisziplinübergreifende Beschreibung des zu entwickelnden Systems. Sie werden während der Entwicklung solange iterativ bearbeitet bis die Beschreibung des zu entwickelnden Systems vollständig ist. [GFD+08a]. Nachfolgend werden die acht Aspekte respektive Partialmodelle zusammenfassend erläutert.

Umfeld: Im Partialmodell Umfeld wird das zu entwickelnde System zunächst als Black-Box betrachtet, die über Beziehungen mit den Elementen des Umfelds verbunden ist zu denen sie in Wechselwirkung stehen. Die Beziehungen sind in Einflussbereiche (z.B. Witterung) und Einflüsse (z.B. Wärmestrahlung) unterteilt. Dem Systemzweck entgegenstehende Einflüsse werden als Störgrößen klassifiziert. Auf Basis einer Analyse der Wechselwirkungen zwischen den Einflüssen können konsistente Bündel von gemeinsam auftretenden Einflüssen gebildet werden. Diese Bündel sind spätere Betriebs-situationen, für die das System ausgelegt sein muss.

Anwendungsszenarien: Basierend auf dem im Partialmodell Umfeld identifizierten Betriebs-situationen wird in den Anwendungsszenarien das gewünschte Verhalten des zu entwickelnden Systems in den identifizierten Betriebs-situationen beschrieben. Dabei werden nicht nur die späteren Standardbetriebs-situationen betrachtet, sondern auch Fehlbedienungs- oder Notfallsituationen. Die Beschreibung der Anwendungsszenarien

²⁶ gefördert durch die Deutsche Forschungsgemeinschaft (DFG)

²⁷ gefördert durch das Bundesministerium für Bildung und Forschung (BMBF)

erfolgt zunächst prosaisch wird dann aber im weiteren Verlauf der Entwicklung insbesondere durch die Partialmodelle Funktionen, Wirkstruktur und Verhalten verfeinert.

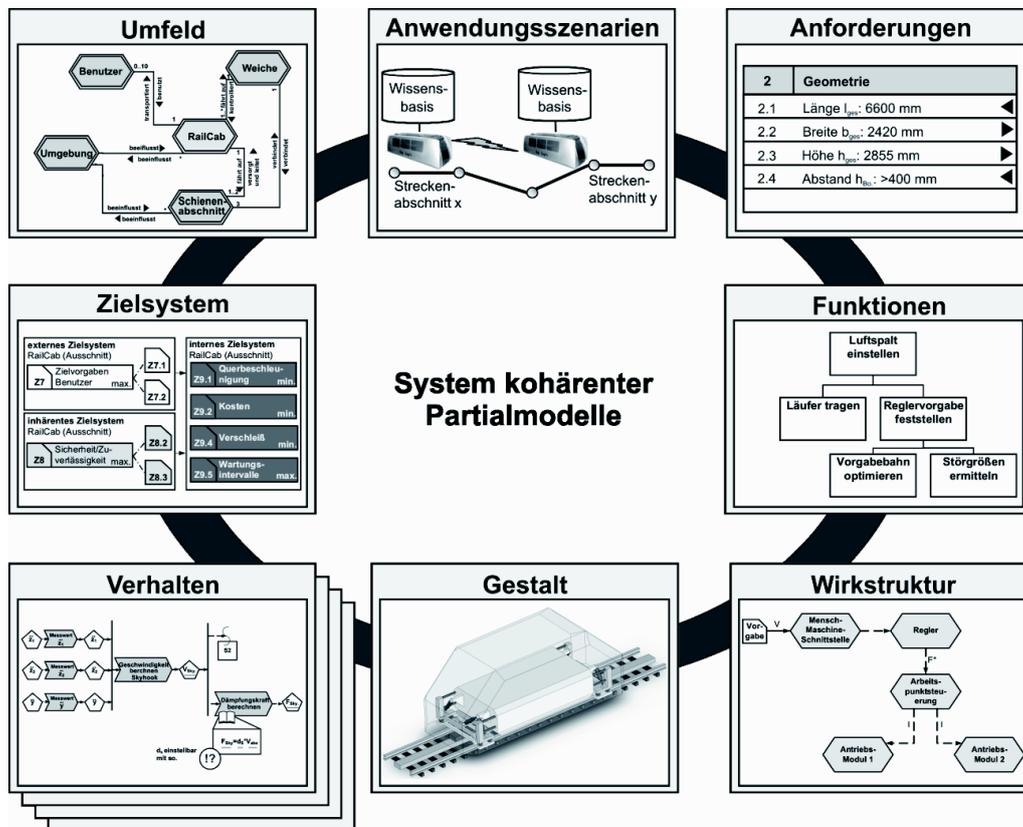


Bild 3-10 Partialmodelle zur Beschreibung der Prinzipiellösung selbstoptimierender Systeme (in Anlehnung an [Fra06, S. 80])

Anforderungen: Die Anforderungen werden mit Hilfe einer in der Praxis üblichen Anforderungsliste beschrieben. Sie werden in Wunsch- und Festforderungen unterschieden. Jede Anforderung hat eine prosaische Beschreibung und kann durch Attribute bzw. deren Ausprägungen weiter konkretisiert werden.

Funktionen: Die Funktionalität des zu entwickelnden Systems wird lösungsneutral in Form einer Funktionshierarchie beschrieben. Ausgehend von der Hauptfunktion des Systems wird diese solange in Teilfunktionen untergliedert bis Lösungsmuster oder -elemente zu deren Realisierung gefunden werden können.

Wirkstruktur: Sie ist das zentrale Partialmodell der Modellierungssprache. In ihr werden die Systemelemente, deren Eigenschaften und deren Beziehungen untereinander beschrieben. Die Systemelemente subsumieren bzw. sind abstrakte Systembestandteile, die im Laufe der Entwicklung durch Software-Komponenten und/oder Bauteile weiter verfeinert werden. Ziel ist die Abbildung der grundsätzlichen Struktur und aller möglichen Systemkonfigurationen.

Verhalten: Das Partialmodell Verhalten fasst verschiedene Verhaltensmodelle zusammen. Die drei wesentlichen sind das Partialmodell Verhalten – Zustände, das Partialmodell Verhalten – Aktivitäten und das Partialmodell Verhalten – Sequenz. Das Partialmodell Verhalten – Zustände bildet die Betriebsmodi, passiven Zustände und möglichen Zustandsübergänge des Systems ab. Ferner beinhaltet es die einen Zustandsübergang auslösenden Ereignisse. Im Partialmodell Verhalten – Aktivitäten werden die in einem bestimmten Zustand im System auftretenden Ablaufprozesse beschrieben. Das Partialmodell Verhalten – Sequenz dient der Darstellung der Interaktion zwischen einzelnen Systemelementen.

Gestalt: Im Partialmodell Gestalt werden die geometrischen Eigenschaften der gestaltbehafteten Systemelemente abgebildet. Wesentliche Inhalte sind hierbei die Hüllflächen, Wirkflächen, Wirkorte und Stützstrukturen.

Zielsystem: Das Partialmodell Zielsystem bildet speziell für selbstoptimierende Systeme deren externe, inhärente und interne Ziele ab. Ihre gegenseitige Beeinflussung kann mit Hilfe einer Einflussmatrix dargestellt werden.

Bewertung: Die Modellierungstechnik CONSENS wurde speziell für die fachdisziplinübergreifende Spezifikation mechatronischer und selbstoptimierender Systeme entwickelt. Ihre Symbole unterscheiden sich deutlich wodurch sich die mit den Symbolen beschriebenen Aspekte intuitiv voneinander unterscheiden lassen. Es existieren eindeutige Modellierungsanweisungen wie welcher Sachverhalt abzubilden ist.

3.2.2.3 Strukturmodellierung nach RIEPE

Eine ebenfalls auf den Arbeiten von KALLMEYER [Kal98] aufsetzende Modellierungssprache für mechatronische Systeme ist die Strukturmodellierung nach RIEPE [Rie03, S. 233ff]. Sie zielt insbesondere darauf die modulare Struktur und die Variantenbeziehungen des zu entwickelnden Produkts in den frühen Phasen der Produktentwicklung in einem ganzheitlichen, fachdisziplinübergreifenden Modell darzustellen. Hierzu werden alternative Systemelemente und Konfigurationen über Variantenstücklisten und Konfigurationsbedingungen spezifiziert. Ihr Zusammenwirken wird in der Wirkstruktur nach KALLMEYER abgebildet. Für die Analyse des entstehenden Modells bietet RIEPE Konzepte für die Bildung verschiedener Sichten (hierarchisch, variantenorientiert, funktionsorientiert, verbindungsorientiert) an. Ferner stellt er einen Ansatz für den Übergang zu domänenspezifischen Modellen vor.

Bewertung: Die Strukturmodellierung nach RIEPE eignet sich sehr gut für die graphikbasierte Generierung und Abbildung von strukturellen Varianten. Dadurch tritt aber die Abbildung weiterer fachdisziplinübergreifend relevanter Aspekte mechatronischer Systeme in den Hintergrund. Zudem werden die Diagramme sehr komplex und sind nicht mehr ohne weiteres intuitiv erfassbar.

3.2.2.4 Modelica®

Modelica® ist eine objektorientierte, gleichungsbasierte Multiphysik-Modellierungssprache. Sie wurde im Rahmen des ESPRIT-Projekts „Simulation in Europe Basic Research Working Group“ zur Modellierung großer, komplexer und physikalisch heterogener Systeme entwickelt. Ihr Ziel ist die Vereinheitlichung von Modellierungssprachen wie bspw. VHDL-AMS²⁸ und ObjectMath²⁹. Ihre Weiterentwicklung und Vermarktung wird durch die Modelica Association vorangetrieben. Diese stellt den Sprachstandard und Modelica-Standardbibliotheken kostenlos zur Verfügung. Die aktuelle Version der Sprachdefinition liegt seit März 2010 in der Version 3.2 vor. [EMO99], [Mod10]

Die Modellierung eines mechatronischen Systems erfolgt in Modelica® akausal mit Hilfe von Objektdiagrammen. Die einzelnen Komponenten werden topologieorientiert über Beziehungen miteinander verbunden und durch Parameter bzw. auf unterster Ebene mit mathematischen Gleichungen in Modelica®-Syntax konkretisiert. Mit Hilfe eines Modelica®-Übersetzers werden sie in ein differential-algebraisches bzw. hybrid differential-algebraisches Gleichungssystem überführt das anschließend gelöst und simuliert werden kann. Auf diese Weise können effizient Gleichungssysteme mit über einhunderttausend Gleichungen verwaltet werden. Die akausale Modellierung mit Modelica® fördert ferner die Bildung von Modellbibliotheken und den Austausch einzelner Modelle oder Modellbestandteile. Derzeit sind verschiedene Bibliotheken für mechanische, elektrische, pneumatische, hydraulische, thermische oder regelungstechnische Komponenten teils frei teils kommerziell verfügbar. Für die Modellierung und Simulation von Modelica®-Modellen bieten verschiedene Software-Hersteller Lösungen an. Beispiele sind Dymola von Dassault Systèmes, SimulationX von ITI oder LMS Imagine.Lab AMESim von LMS. [Mod10]

Bewertung: Modelica® fokussiert die Modellierung und Simulation der Physik geregelter mechatronischer Systeme. Die Modellierung der für fortschrittliche mechatronische Systeme besonders relevanten Fachdisziplin Softwaretechnik wird nur in Ansätzen unterstützt. Für die Abbildung eines Systemmodells, anhand dessen die übrigen Entwicklungsobjekte gespiegelt werden, ist Modelica® nicht geeignet.

²⁸ VHDL-AMS (VHSIC Hardware Description Language – Analog and Mixed Signal Extensions) ist eine textbasierte Spezifikation zur Beschreibung digitaler Systeme.

²⁹ ObjectMath ist eine objektorientierte Erweiterung der Computer-Algebra-Sprache „Mathematica“, mit der mathematische Modelle durch Gruppierung von Funktionen und Gleichungen strukturiert werden können.

3.3 Dynamische und adaptive Planungs- und/oder Steuerungsverfahren für Entwicklungsprozesse

In der Problemanalyse sind grundlegende Techniken und Strategien zur Planung und Steuerung von Entwicklungsprojekten aufgezeigt worden (vgl. Abschnitt 2.4.3 und 2.5.2). Nachfolgend werden speziell solche Verfahren untersucht, die die dynamische oder adaptive Planung und Steuerung von Entwicklungsprozessen adressieren.

3.3.1 Situative Entwicklungsaufgaben- und Methodenauswahl nach PONN

Die von PONN entwickelte Systematik unterstützt Produktentwicklungsingenieure dabei abhängig von der gegenwärtigen Entwicklungssituation die als nächstes zu bearbeitende Entwicklungsaufgabe und geeignete Methoden zur erfolgreichen Bewältigung dieser Aufgabe zu identifizieren und für ihr weiteres Handeln einzuplanen. Die Systematik fokussiert die Konzeptentwicklung technischer Produkte und umfasst die drei Bestandteile **Beschreibungsmodell**, **Anwendungsmethodik** und **Informationssammlung** [Pon07, S. 121ff.].

Das **Beschreibungsmodell** definiert, wie Entwicklungssituationen, Entwicklungsaufgaben, Methoden und deren Verknüpfung zu spezifizieren sind, damit ausgehend von der Analyse der Entwicklungssituation die Auswahl von Entwicklungsaufgabe und potentiell anzuwendenden Methoden erfolgen kann (vgl. Bild 3-11).

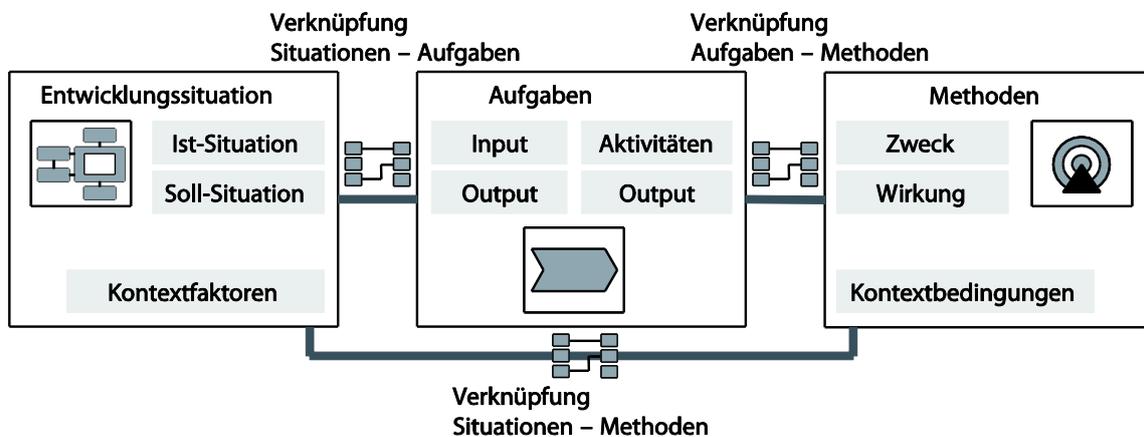


Bild 3-11 Beschreibungsmodell zur situativen Entwicklungsaufgaben- und Methodenauswahl [Pon07, S. 129]

PONN unterscheidet bei der Beschreibung von Entwicklungssituationen zwischen dem direkten und dem indirekten Kontext (vgl. Bild 3-12). Der direkte Kontext wird durch Faktoren beschrieben, die das gegenwärtige Betrachtungsobjekt des Entwicklungsingenieurs und dessen Wissensstand charakterisieren. Diese Faktoren können Ergebnisse, Ereignisse oder Erkenntnisse sein. Für jeden Faktor sind nach dem Ausführen einer Entwicklungsaufgabe der Ist- und der Soll-Zustand zu definieren. Der indirekte Kontext wird durch Faktoren beschrieben, die primär die Auswahl in Frage kommender Metho-

den betreffen und keinen direkten Einfluss auf die Entwicklungsaufgabenauswahl haben. Die Faktoren des indirekten Kontexts werden den Bereichen Entwicklungsaufgabe/Produkt, Einwicklungsprozess, Entwickler/Team und Rahmenbedingungen zugeordnet.

Entwicklungsaufgaben werden in Form von Prozessbausteinen mit einem standardisierten Raster mit dem Inhalt: Name, Kurzbeschreibung, Abbildung, Aktivitäten, Input und Output beschrieben. Die Methodenbeschreibung erfolgt in ähnlicher Form durch Methodensteckbriefe. Sie beinhalten Name, Kurzbeschreibung, Abbildung, Zweck, Voraussetzungen, Wirkung und Kontextbedingungen einer Methode. Die Kontextbedingungen adressieren dieselben Bereiche wie die Faktoren zur Beschreibung des indirekten Kontexts der Entwicklungssituation.

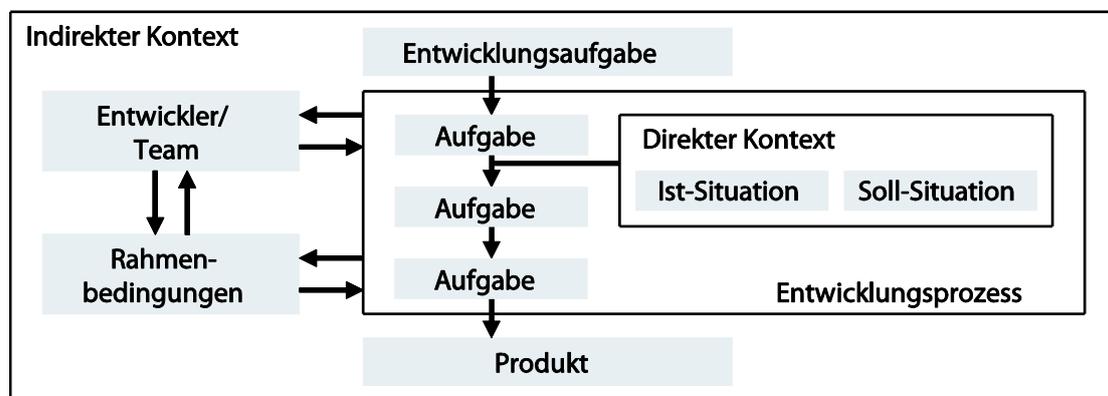


Bild 3-12 Beschreibung der Entwicklungssituation [Pon07, S. 123]

Die Verknüpfungen zwischen der Beschreibung von Entwicklungssituation, Entwicklungsaufgabe und Methode werden in der Verknüpfungsstruktur beschrieben. Es werden Entwicklungssituationen mit Entwicklungsaufgaben, Entwicklungsaufgaben mit Methoden und Entwicklungssituationen mit Methoden verknüpft (vgl. Bild 3-12). Die Entwicklungssituationen werden mit den Entwicklungsaufgaben über die Ist- und Sollzustände von deren Input- und Output-Größen verknüpft. Entwicklungsaufgaben und Methoden können direkt, über Tätigkeiten oder über Objekte verbunden werden. Die Verknüpfung der Entwicklungssituationen und Methoden erfolgt über die den indirekten Kontext der Entwicklungssituation beschreibenden Faktoren und die Kontextbedingungen der Methode.

Die **Anwendungsmethodik** beschreibt, wie in vier Schritten (Situationsanalyse, Aufgabenauswahl, Methodenauswahl und Methodenanwendung) ausgehend von der gegenwärtigen Entwicklungssituation die als nächstes auszuführende Entwicklungsaufgabe und die anzuwendende Methode identifiziert werden. Wenn auch die Reihenfolge der Schritte nicht beliebig ist, so betont PONN ausdrücklich, dass abhängig vom Erkenntnisstand des Entwicklungsingenieurs und den äußeren Umständen bei jedem der vier Schritte der Anwendungsmethodik eingestiegen werden kann.

In der **Informationssammlung** wurden die einzelnen Bestandteile des Beschreibungsmodells für die Konzeptentwicklung technischer Produkte ausgeprägt. Insgesamt definiert PONN 18 Prozessbausteine und 36 Methoden sowie deren Verknüpfungen anhand spezifischer Faktoren zur Beschreibung des direkten und indirekten Kontexts von Entwicklungssituationen. Zur effizienten Unterstützung der Entwicklungsingenieure wurden das Beschreibungsmodell und dessen spezifische Ausprägung in Form des Informationsmodells prototypisch in einem webbasierten Rechnerwerkzeug implementiert.

Bewertung: Die von PONN entwickelte situationsspezifische Entwicklungsaufgaben- und Methodenauswahl klassifiziert die vorliegende Entwicklungssituation und gibt dem Entwickler konkrete Handlungsanweisung für die als nächstes auszuführende Tätigkeit sowie die dabei einzusetzenden Methoden. Auf diese Weise werden die Entwickler während der mit großen Unsicherheiten behafteten Konzipierung situationsangepasst zur Problemlösung befähigt. Das Verfahren ist reaktiv ausgerichtet und sieht nicht die Planung des Entwicklungsprozess bis zu Ende vor. Eine ganzheitliche Optimierung des Entwicklungsprozesses wird nicht adressiert.

3.3.2 Methode zur flexiblen Gestaltung integrierter Entwicklungsprozesse nach BICHLMAIER

Für die schnelle Anpassung des Entwicklungsprozesses an unvorhergesehene Ereignisse und wechselnde Rahmenbedingungen entwickelt BICHLMAIER [Bic00] einen wissensbasierten Ansatz, bei dem der Entwicklungsprozess mit Hilfe von Entwicklungsprozessbausteinen bzw. kleinen Prozessketten zusammengesetzt und bei Bedarf weiter detailliert oder auch flexibel geändert werden kann. Der Fokus seiner Methode liegt auf der Stärkung der Zusammenarbeit der verschiedenen an der Entwicklung beteiligten Fachdisziplinen. Die von ihm zur Demonstration der Methode gewählten Fachdisziplinen sind die Konstruktion und die Montageplanung.

Die **Entwicklungsprozessbausteine** als kleinster Baustein eines Prozesses sollen inhaltliche Hilfestellung bei der Bearbeitung fachdisziplinübergreifender Arbeitsinhalte geben. Sie beschreiben entsprechend diese Inhalte in Form der auszuführenden Tätigkeiten und benennen deren Ein- und Ausgangsinformationen (vgl. Bild 3-13). Optional können die zur Ausführung der Tätigkeiten benötigten Kompetenzen, Ressourcen, Methoden und Werkzeuge angegeben werden. Diese Kriterien unterstützen bei der späteren Auswahl der Prozessbausteine.

Für die Vernetzung der Entwicklungsprozessbausteine zum auszuführenden Entwicklungsprozess gibt BICHLMAIER zwei Möglichkeiten an. Bei der **Vernetzung durch Strukturierung nach dem Vorgehenszyklus** werden die Entwicklungsprozessbausteine entsprechend dem allgemeinen Vorgehen beim Problemlösen zusammengestellt. Dies bedeutet, dass abhängig vom vorliegenden Teilproblem, Prozessbausteine, die Analyse-, Synthese-, Bewertungs- oder Auswahlsschritte darstellen, zu einem Prozessnetz kombiniert werden. Die zweite Möglichkeit ist die **Vernetzung durch Betrachtung**

ung der Schnittstellen. Hierbei werden die nach einem gewählten Entwicklungsprozessbaustein vorliegenden Ausgangsinformationen mit den notwendigen Eingangsinformationen der potentiell anschließend ausführbaren Entwicklungsprozessbausteine verglichen. Auf diese Weise können die als nächstes auszuführenden Entwicklungsschritte bestimmt werden. Das Prozessnetz wird in einer vorwärtsgerichteten Suche aufgebaut.

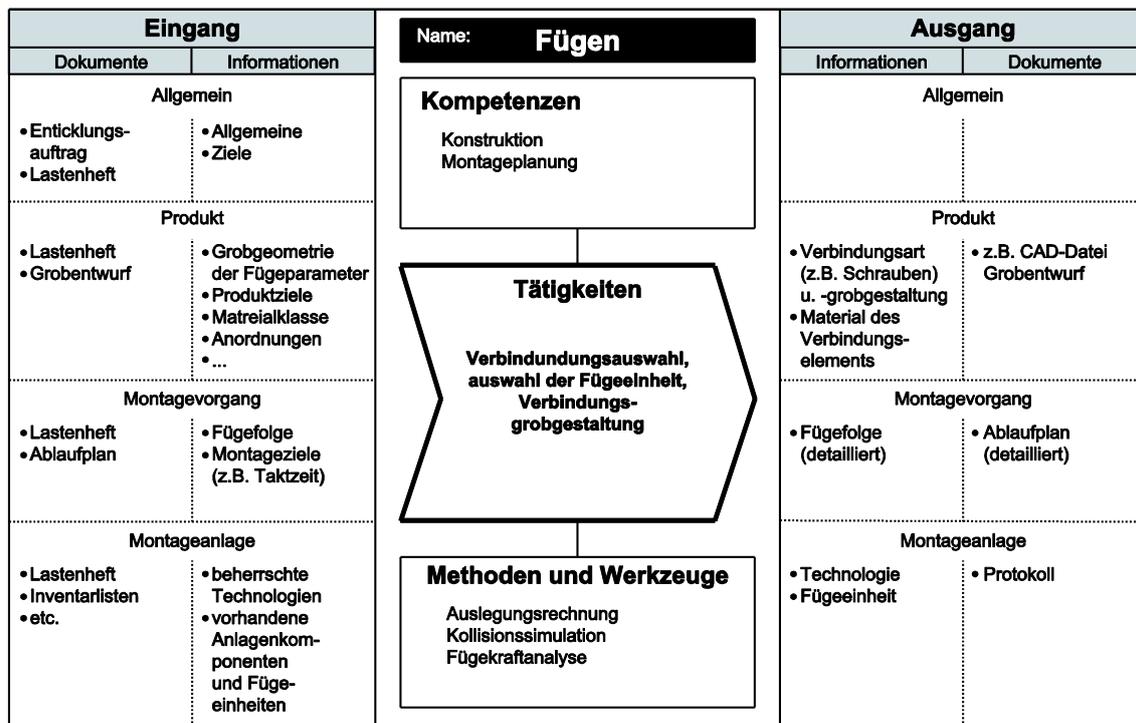


Bild 3-13 Beispiel eines Entwicklungsprozessbausteins nach Bichlmaier [Bic00, S. 80]

Die Methode ist software-technisch in Form eines Prozessbaukastens und eines Prozessmodellierers prototypisch implementiert. Im Prozessbaukasten sind beispielhaft verschiedene Entwicklungsprozessbausteine abgelegt. Die einzelnen Entwicklungsprozessbausteine können mit Hilfe des Prozessmodellierers miteinander kombiniert werden. Dieser liefert zur Unterstützung des Benutzers eine Liste der durch das aktuelle Prozessnetz bereitgestellten Ausgangsinformationen.

Bewertung: Der wissensbasierte Ansatz von BICHLMAIER unterstützt die dynamische Vorausplanung und ggf. notwendige Anpassung des Entwicklungsprozesses technischer Systeme. Die Orientierung am aktuell vorliegenden Entwicklungsgegenstand und der Entwicklungssituation ist nicht systematisiert. Die Auswahl und Vernetzung der Prozessbausteine ist durch die Entwickler zu leisten.

3.3.3 Adaptive Planung und Steuerung von integrierten Entwicklungs- und Planungsprozessen nach MURR

MURR [Mur99] entwickelt ein adaptives Geschäftsprozessmanagement zur Steigerung der Effizienz, Effektivität und Flexibilität von Entwicklungsprozessen. Prozessverantwortliche und -durchführende sollen mit dessen Hilfe dezentral auf nicht planbare Einflüsse und Störungen im Entwicklungsprozess reagieren können. Das adaptive Geschäftsprozessmanagement basiert auf der Verwendung von generischen Prozessbausteinen und einer Planungs- und Steuerungsmethodik zu deren Nutzung.

Die generischen **Prozessbausteine** bestehen aus der Benennung des Entwicklungs- oder Planungsschritts, dessen notwendigen Input und dem resultierenden Output (vgl. Bild 3-14). Input und Output werden dabei als Zustände von Basisobjekten aus den Bereichen Produkt, Produktionsprozess und Produktionsmittel erfasst. Die Basisobjekte erfahren bei der Ausführung der Entwicklungs- und Planungsschritte einen Zustandsübergang. Ein Entwicklungs- oder Planungsschritt kann abhängig von den Zuständen der vorliegenden Basisobjekte ausgeführt werden oder nicht. Der Zustand eines Basisobjekts wird mit Hilfe von Reifegradindikatoren beschrieben. Die Prozessbausteine werden über den Vergleich der Zustände der Input- und Output-Größen zum Entwicklungsprozess kombiniert. Zusätzlich zum Entwicklungs- und Planungsschritt sowie dessen Input- und Output-Größen werden in einem Prozessbaustein die zur Ausführung benötigten Methoden, zugehörigen Aufgaben, Ressourcen, sowie Zeit- und Kostenaufwände beschrieben.

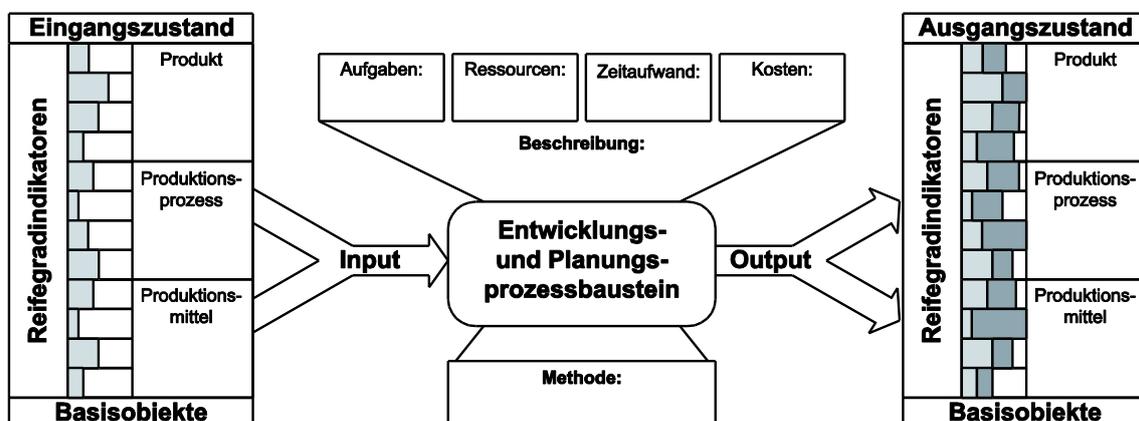


Bild 3-14 generischer Geschäftsprozessbaustein nach Murr (in Anlehnung an [Mur99, S. 62])

Der Entwicklungsprozess wird mit Hilfe der in Bild 3-15 dargestellten **Planungs- und Steuerungsmethodik** aufgebaut und bei Bedarf während der operativen Ausführung auch angepasst. Die Planungs- und Steuerungsmethodik sieht eine Planungs-, eine Steuerungs- und eine Monitoring-Komponente vor. Die Planungskomponente dient der Analyse des Entwicklungsvorhabens und der Definition eines adäquaten Referenzprozesses. Neben der Struktur- und Ablaufplanung soll hier auch eine Termin-, Ressourcen- und Kostenplanung stattfinden. In der Steuerungskomponente werden bei Abweichungen

des Ist-Verlaufs vom Soll-Verlauf Gegenmaßnahmen initiiert. Die Identifikation der Abweichungen ist Aufgabe der Monitoring-Komponente. Alle drei Komponenten orientieren sich bei der Erfüllung ihrer Funktion an einem Zielsystem in dem die zu erreichenden Merkmale des Produkts oder Prozesses in Form eines Zielwertes oder Zielkorridors quantitativ oder qualitativ festgelegt sind.

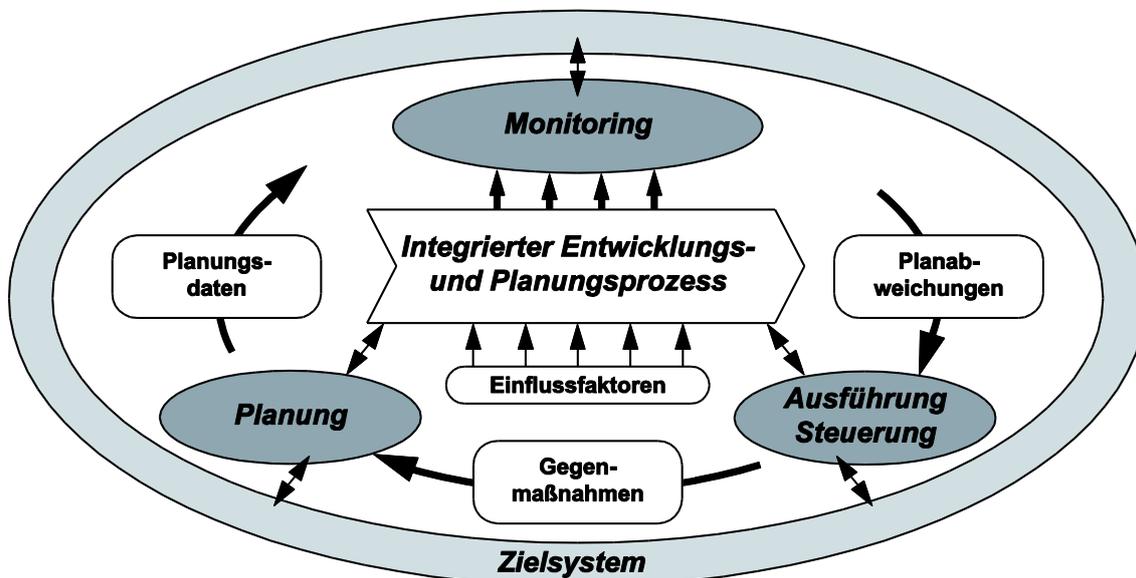


Bild 3-15 Planungs- und Steuerungsmethodik nach MURR [Mur99, S. 90]

Bewertung: Die adaptive Planung und Steuerung von integrierten Entwicklungs- und Planungsprozessen nach MURR beinhaltet die wesentlichen Prozessmanagementelemente zur situationsangepassten Planung, Steuerung und Überwachung von Produktentwicklungsprozessen wobei keine Modifikationen der Prozessziele im Sinne der Selbstoptimierung adressiert werden. Die von ihm spezifizierten Prozessbausteine ermöglichen eine detaillierte Vorausplanung des Entwicklungsprozesses. Gleichzeitig verhindert die Art der Prozessmodellierung mit dem beinhalteten Konzept der Transformation der Reifegradindikatoren die problemunabhängige Verwendung des Verfahrens.

3.3.4 Verfahren zur Planung von Entwicklungsprozessen für fortgeschrittene mechatronische Systeme nach REDENIUS

Gegenstand des von REDENIUS entwickelten Verfahrens ist die Generierung eines spezifischen Entwicklungsprozesses auf Basis von Erfahrungswissen aus bereits erfolgreich abgeschlossenen Entwicklungsprojekten. Der generierte Prozess soll den Entwicklern detaillierte Handlungsanweisungen hinsichtlich Vorgehensweise sowie einzusetzender Methoden und Hilfsmittel geben. Das Verfahren basiert auf den drei Bestandteilen: **Wissensbasen/-speicher, Auswahlmechanismen und Anpassungskonzepte.**

In der **Wissensbasis** ist das Erfahrungswissen aus bereits erfolgreichen abgeschlossenen Entwicklungsprojekten in Form von Prozessbausteinen, Prozesssträngen oder kompletten Entwicklungsprozessen abgelegt. Der Prozessbaustein ist die kleinste Einheit und stellt wie bei BICHLMAIER (vgl. Abschnitt 3.3.2 bzw. [Bic00, S. 65]) eine abgeschlossene Tätigkeitsfolge mit einem definierten Ergebnis dar. Mehrere Prozessbausteine werden zu einem Prozessstrang verbunden. Sie dienen der Entwicklung eines Moduls oder eines kompletten technischen Systems. Mehrere Prozessstränge ergeben den Entwicklungsprozess. In der Vergangenheit erfolgreich durchgeführte Entwicklungsprozesse werden entsprechend als Kombinationen von Entwicklungssträngen gespeichert.

Welche bisher erfolgreich angewendeten Prozesse in welcher Form für ein neu vorliegendes Entwicklungsproblem angewendet werden können, wird mit Hilfe der **Auswahlmechanismen** bestimmt. Das top-down-gerichtete Vorgehen bei der Auswahl ist in Bild 3-16 dargestellt.

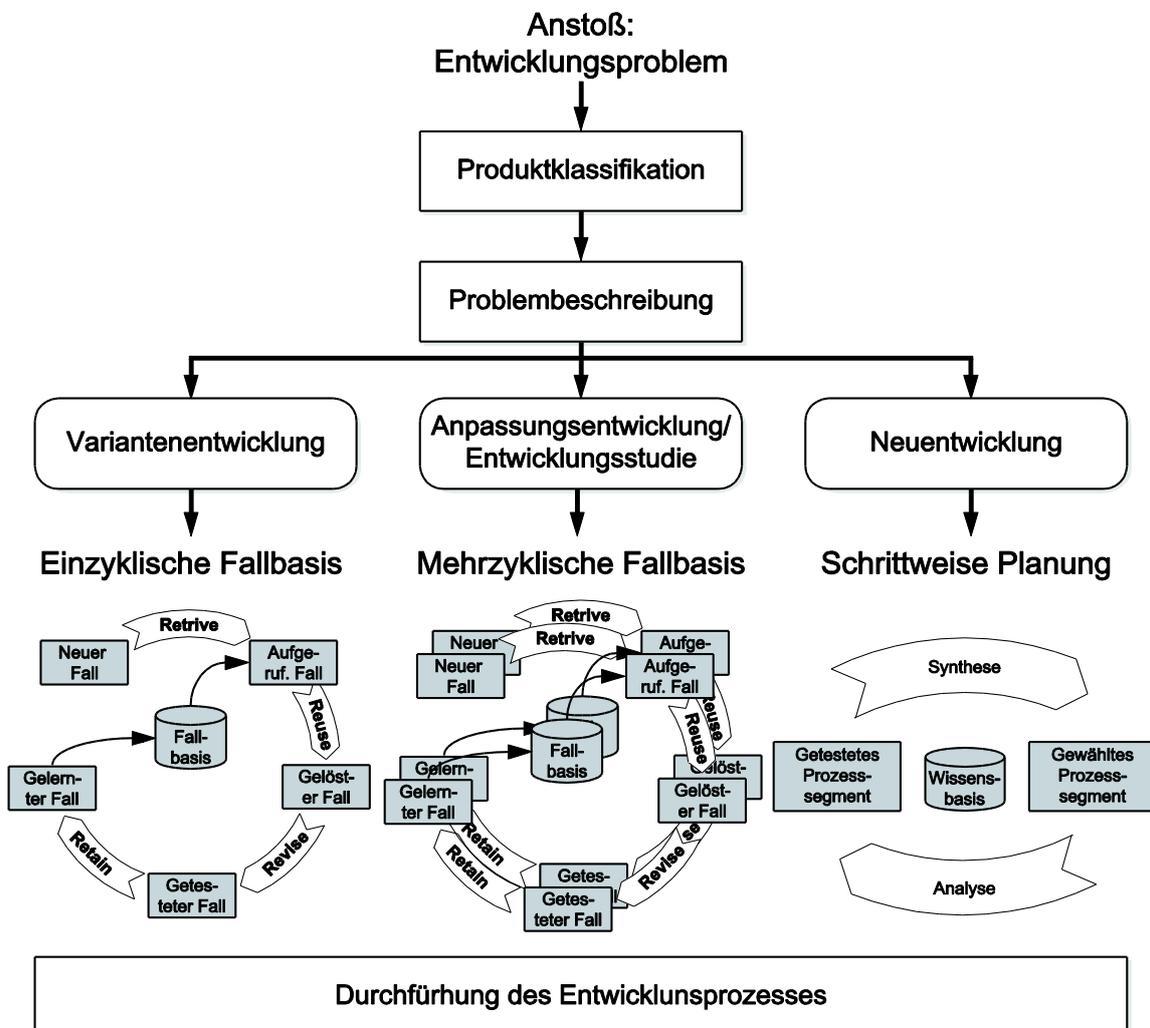


Bild 3-16 Vorgehen bei der Planungsunterstützung nach REDENIUS (nach [Red06, S. 111])

Der Ausgangspunkt ist die Klassifikation des zu entwickelnden Produkts nach dem Industriestandard eCl@ss. Den verschiedenen Produktklassen sind wiederum verschiedene Formblätter mit Merkmalen zur Beschreibung der Entwicklungsprobleme der jeweiligen Klasse zugeordnet. Die Merkmale sind in die Kategorien: Aufgabenstellung, Zielsystem und Randbedingungen unterteilt. Durch die Kombination von Produktklassifikation und Problembeschreibung wird das Entwicklungsproblem Stück für Stück immer detaillierter spezifiziert. Abhängig davon, ob es sich um eine Variantenentwicklung, eine Anpassungsentwicklung, eine Entwicklungsstudie oder um eine Neuentwicklung handelt, sind alle oder nur ein Teil der notwendigen Klassifikationsmerkmale bekannt. Es wird davon ausgegangen, dass bei einer Variantenentwicklung die meisten und bei einer Neuentwicklung die wenigsten der notwendigen Klassifikationsmerkmale bekannt sind. Sind alle Klassifikationsmerkmale bekannt, kann ein spezifischer Entwicklungsprozess innerhalb eines Planungszyklus ermittelt werden. Die Auswahl der entsprechenden Prozessstränge bzw. -bausteine erfolgt mit Hilfe des fallbasierten Schließens. Ist ein Teil der Klassifikationsmerkmale unbekannt, kann kein spezifischer dafür aber ein abstrakter Entwicklungsprozess ermittelt werden, der die Hauptphasen der Entwicklung beinhaltet. Mit fortschreitender Entwicklung und wachsender Problemkenntnis wird der abstrakte Entwicklungsprozess in weiteren Planungszyklen detailliert. Ist der Grad der Unklarheit des Entwicklungsproblems derart hoch, dass nicht mal ein abstrakter Entwicklungsprozess identifiziert werden kann, findet eine schrittweise Planung statt, bei der mit Hilfe einzelner Prozessbausteine die jeweils nächsten Entwicklungsschritte geplant werden. Es werden hierfür mögliche Prozessbausteine vorgeschlagen, deren konkrete Auswahl und Anordnung dem Entwickler überlassen wird.

In der Regel passt kein bisher angewandter Entwicklungsprozess eins zu eins für ein neu vorliegendes Entwicklungsproblem. Die vorgeschlagenen Prozesse können daher im Rahmen des vorgesehenen **Anpassungskonzepts** modifiziert werden. Die daraus ggf. neu entstehenden Prozessstränge werden mit ihrer zugehörigen Produkt- und Problemmklassifikation in der Wissensbasis abgelegt.

Das Verfahren wurde in Form des Software-Werkzeugs „WisePlan“ (Werkzeug zur erfahrungsbasierten Planung) prototypisch implementiert. Es unterstützt die Produkt- und Problemmklassifikation sowie den darauf aufbauenden Zugriff auf eine Microsoft ACCES Datenbank, in der verschiedene in Microsoft Visio modellierte Prozessmodelle hinterlegt sind.

Bewertung: Die Planung von Entwicklungsprozessen nach REDENIUS ermöglicht einen strukturierten Zugriff auf empirisches Lösungswissen und stellt den Entwicklern abhängig von der Güte der Beschreibung der Entwicklungsaufgabe spezifische Handlungsanweisungen intuitiv verständlich zur Verfügung. Die Verwendung des eCl@ss Standards als Grundlage für die Auswahl der anzuwendenden Entwicklungsschritte erschwert aber eine problemunabhängige Nutzung des Verfahrens. Die als Rückfallebene gedachten generischen Vorgehensweisen bilden die interdisziplinären Wechselwirkungen im Entwicklungsprozess nur rudimentär ab.

3.3.5 Adaptive Entwicklungsprozessplanung nach SCHUMANN

Der zunehmenden Dynamik, Komplexität und Parallelität von Entwicklungsprozessen Rechnung tragend entwickelt SCHUMANN [Sch94] ein dreigeteiltes dynamisches Modell zur adaptiven Planung, Überwachung und Steuerung von Entwicklungsprozessen. Das Modell gilt für Entwicklungsprozesse technischer Systeme mit einer gewissen Mindestkomplexität aber ohne Beschränkung auf eine spezifische Branche, spezifisches Produkt oder spezifischen Prozesstyp. Ziel ist die Steigerung der Flexibilität, Transparenz und Effizienz der Produktentwicklung. Es handelt sich um einen wissensbasierten Ansatz, bei dem das in einem Unternehmen vorhandene methodische Entwicklungswissen mit Hilfe verschiedener Teilmodell abgebildet wird und dieses Wissen dann zur wahlweise teil- oder vollautomatisierten Anpassung des Entwicklungsprozesses an die aktuellen Gegebenheiten genutzt wird.

Das dynamische Entwicklungsprozessmodell besteht aus den drei Teilmodellen: *Produktmodellierungsmodell*, *Metaproduktmodellierungsmodell* und *Koordinationsstrategiemodell*. Die Modelle sind jeweils unternehmensspezifisch auszuprägen.

Im **Produktmodellierungsmodell** werden die in einem Unternehmen möglichen Produktmodellierungsaktionen, die möglichen Gestaltungsgegenstände und Gestaltungsmerkmale abgebildet. Auf dieser Basis werden die bereits ausgeführten und noch auszuführenden Phasen und Zustände des operativen Entwicklungsprozesses und der zugehörigen Entwicklungsgegenstände abgebildet. Sie ergeben das sogenannte Produktmodellierungsprozessmodell.

Die Aktionen um das Produktmodellierungsprozessmodell zu generieren, zu überwachen und zu beeinflussen werden im **Metaproduktmodellierungsmodell** beschrieben. Dieses Modell beschreibt somit die Aktionen zum Managen des operativen Entwicklungsprozesses. Die Ausführung der Aktionen des Metaprozessmodellierungsprozessmodells werden durch die Zustände des Produktmodellierungsprozessmodells getriggert. Dadurch findet eine Integration von operativer Entwicklung und deren Management statt

Wie die Triggerung der Metaproduktmodellierungsaktionen erfolgt ist im **Koordinationsstrategiemodell** beschrieben. Hier sind bspw. eine iterative, durch den Benutzer gestützte Aktionsselektion, eine automatisierte reaktiven Vorwärtssuche oder eine automatisierte Rückwärtssuche denkbar. Das Modell ist für unternehmensspezifische Erweiterungen offen.

Bewertung: Der von SCHUMANN entwickelte Ansatz zur adaptiven Entwicklungsprozessplanung ist ein konzeptionelles Rahmenwerk, mit dem unternehmensinterne Entwicklungs- und Planungsmechanismen abgebildet und anschließend automatisiert werden können. Die beschriebenen Modelle und Verfahren eignen sich somit sehr gut, um standardisierte Anpassungsmechanismen zu automatisieren. Eine flexible Anpassung an veränderte Entwicklungszielsetzungen wird aber nicht unterstützt.

3.3.6 Methoden zur dynamischen Planung und Steuerung von Produktentwicklungsprozessen nach DEMERS

Da der konkrete Verlauf von Entwicklungsprozessen nicht vorhersehbar und damit nur schwer planbar ist, sieht DEMERS [Dem00] die Planung des Entwicklungsprozesses nicht als einmalige, zu Anfang der Entwicklung vollständig ausführbare Aufgabe, sondern als entwicklungsbegleitende Tätigkeit, die durch verschiedene Methoden zu unterstützen ist. Abhängig von der gegenwärtigen Entwicklungssituation wird der Entwicklungsprozess dynamisch angepasst und wenn möglich konkretisiert. Seine Methoden umfassen eine neu entwickelte **Modellierungssprache** und **Regeln** zu deren Anwendung sowie ein **Vorgehensmodell** zur Generierung, Analyse, Bewertung und Auswahl von Handlungsalternativen [Dem00, S. 3ff.].

Die entwickelte **Modellierungssprache** baut im Wesentlichen auf der Methode zur Problemformulierung des TRIZ Methodenbaukastens und weiteren Modellierungsmethoden für Geschäftsprozesse wie ARIS und SADT auf. Sie umfasst die Elemente: Aktivität, Ein- bzw. Ausgangsgröße, Steuergröße, Ressourcen bzw. Mechanismen, positives bzw. neutrales Ereignis und negatives Ereignis. Sie können über die verschiedenen Verknüpfungen: „wird benötigt für“, „verursacht“, „wurde eingeführt, um zu eliminieren“, „löst aus“ und „verhindert/behindert“ miteinander verbunden werden. Die Elemente können dabei die Zustände: Vergangenheit, Gegenwart und Zukunft einnehmen. Dadurch wird es möglich die Zusammenhänge zwischen dem zu entwickelnden Produkt, dem Entwicklungsprozess und weiteren Einflüssen auf die Entwicklung zeitbezogen abzubilden. In Bild 3-17 ist bspw. der Fall dargestellt, dass es bei einem Prototypentest zu Brummen der zu entwickelnden Maschine kommt. Daher wird eine spontane Ursachenanalyse geplant, die die Ursache der Brummgeräusche offenlegen soll. Welche Objekte durch welches Element und welche Elemente mit welchen Verknüpfungen verbunden werden dürfen, wird durch die **Regeln** zur Anwendung der Modellierungssprache festgelegt. Bspw. muss die *verursacht*-Verknüpfung immer auf ein negatives Ereignis zeigen [Dem00, S. 81ff.].

Um den Entwicklungsprozess an die gegenwärtige Entwicklungssituation anzupassen, wird diese im ersten Schritt des **Vorgehensmodells** zur Generierung, Analyse, Bewertung und Auswahl von Handlungsalternativen mit Hilfe der Modellierungssprache erfasst. Die Modellierung der Entwicklungssituation umfasst die Abbildung der ursprünglichen Planung, der Vergangenheit, der Gegenwart und der bisher zusätzlich zur ursprünglichen erstellten Planung. Das so entstehende Modell der Entwicklung erlaubt es sich gegenseitig negativ beeinflussende Elemente im Entwicklungsgeschehen zu erkennen und solche Aktivitäten im Entwicklungsprozess zu identifizieren, an denen die geplanten Ergebnisse bisher nicht erreicht wurden. Im nächsten Schritt des Vorgehensmodells werden Handlungsalternativen zur Verbesserung des Entwicklungsgeschehens erarbeitet. Hierbei werden zwei Strategien unterstützt. Die erste Strategie zielt darauf ab negative Beeinflussungen zwischen den Elementen oder blockierte Ergebnisse dadurch zu bereinigen, dass ein alternativer Entwicklungsprozess geplant wird, in dem die be-

Abhängig von der gegenwärtigen Entwicklungssituation sollen dem Entwickler mit Hilfe des Software-Werkzeugs „Prozessnavigator“ kontextsensitiv Handlungsoptionen sowie für seine Arbeit notwendige Produktdaten, Methoden, benötigtes Wissen und Werkzeuge bereitgestellt werden. Der Prozessnavigator hat dabei den Anspruch den Charakteristika von Produktentwicklungsprozessen – nicht deterministische, kreative und schwer strukturierbare Prozesse mit unvollständigen und unsicheren Daten – gerecht zu werden. Bild 3-18 visualisiert den beschriebenen Anspruch und die Rolle des Prozessnavigators.

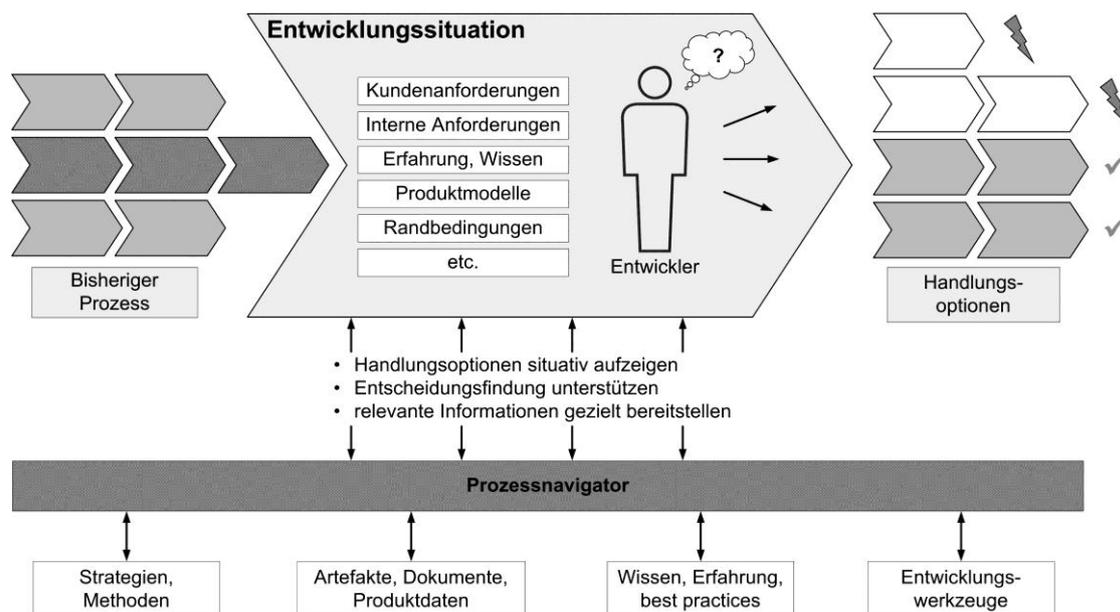


Bild 3-18 Schema der Prozess- und Workflowunterstützung zur Planung und Steuerung der Abläufe im Produktentwicklungsprozess des Forschungsverbunds FORFLOW [MHJ+09, S. 26]

Die situationsspezifische Prozessplanung und -konfiguration ist in der im Rahmen des Forschungsverbundes erarbeiteten Dissertation von ROELOFSEN [Roe11] behandelt worden. Ziel ist die methodische Unterstützung der situationsspezifischen Auswahl und Konfiguration von Entwicklungsprozessen in Kombination mit adäquaten Methoden. Hierfür wird ein **universelles, auf den spezifischen Anwendungsfall anpassbares Vorgehensmodell** für die Entwicklung multidisziplinärer Produkte, ein **Ansatz für die produktmodellbasierte Planung und Ausführung von Entwicklungsprozessen**, ein **Schema zur Analyse von Entwicklungssituationen** sowie **Methoden zur strategischen und operativen Planung von Entwicklungsprozessen** vorgestellt. Diese sind prototypisch im Prozessnavigators umgesetzt [MHJ+09, S. 48ff], [Roe11, S. 92ff].

Die Basis der situationsspezifischen Auswahl und Konfiguration von Entwicklungsprozessen bildet das **FORFLOW-Prozessmodell**. Es soll eine situationsspezifische Prozessplanung, die Integration von Iterationen und Lessons Learned, eine verbesserte Informationsversorgung der Entwickler, die Integration von CAX-Werkzeugen, die Absicherung des zu entwickelnden Produkts auf mehreren Hierarchieebenen, die Integration

von Simultaneous und Concurrent Engineering Aspekten, eine DfX-Unterstützung und die dynamische Verknüpfung des Prozesses mit den während der Entwicklung entstehenden Produktmodellen ermöglichen. Das Prozessmodell ist in drei Hierarchieebenen unterteilt. Die erste Ebene umfasst sechs Hauptschritte eines Entwicklungsprozesses, wie z.B. „Klären und Präzisieren der Aufgabenstellung“ oder „Systemgestaltung“. Die zweite und dritte Ebene präzisieren diese Hauptschritte und unterteilen sie jeweils weiter. Auf der dritten Ebene werden insgesamt 86 Schritte unterschieden. Für die Ebenen eins und zwei werden Reihenfolgen für die Bearbeitung der Schritte empfohlen aber nicht fest vorgeschrieben. Die Reihenfolge der Bearbeitung der Schritte auf Ebene drei ist bewusst offen gelassen und soll situationsspezifisch ausgeprägt werden [KEL+10, S. 60ff.], [MHJ+09, S. 27ff.], [Roe11, S. 92ff.].

Das generelle Vorgehen für die situationsspezifische Ausprägung und Ausführung des FORFLOW-Prozessmodells bildet der in Bild 3-19 schematisch dargestellte Ansatz **Product Model Driven Development (PMDD)**. Er unterscheidet zwei Ebenen für die Analyse der Entwicklungssituation und die Planung des Entwicklungsprozesses.

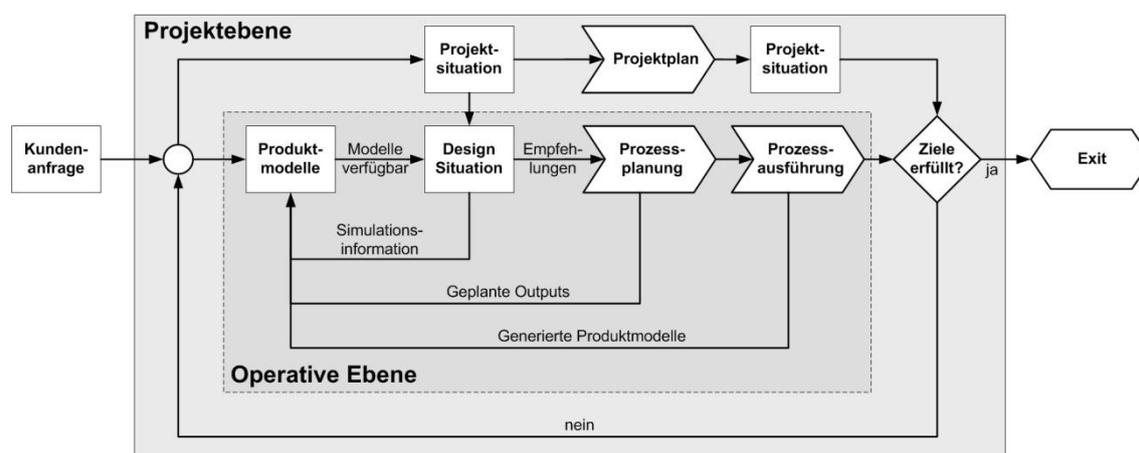


Bild 3-19 Product Model Driven Development [Roe11, S. 111]

Genereller Ansatz ist es den Entwicklungsprozess über die zu generierenden Produktmodelle zu steuern. Auf der obersten Ebene, der Projektebene, werden die Rahmenbedingungen der Entwicklung zur Projektsituation zusammengefasst und darauf aufbauend die Projektziele festgelegt und der Entwicklungsprozess auf der ersten und zweiten Hierarchieebene des FORFLOW-Prozessmodells geplant. Auf der zweiten Ebene des PMDD, der operativen Ebene, werden ggf. vorliegende und entsprechend dem bisherigen Prozessplan zu entwickelnde Produktmodelle, die Projektsituation und weitere Parameter zur Design-Situation kombiniert. Die Design-Situation dient der Ausgestaltung des FORFLOW-Prozessmodells auf der dritten Hierarchieebene hin zum auszuführenden Entwicklungsprozess. Für die kontinuierliche Anpassung des Entwicklungsprozesses werden die zu generierenden Produktmodelle und somit der auszuführende Entwicklungsprozess durch drei Rückführzweige der aktuellen Design-Situation sowie dem Status der Entwicklung angepasst [MHJ+09, S. 53f.], [RLL07, S. 24f.], [Roe11, S. 110ff.].

Die Analyse von Projekt- und Design-Situation erfolgt anhand des **Schemas zur Analyse von Entwicklungssituationen**. Insgesamt werden 16 Parameter aus drei Kategorien verwendet, die in der nachfolgenden Tabelle 3-2 dargestellt sind.

Tabelle 3-2 Parameter zur Beschreibung von Entwicklungssituationen nach [Roe11, S. 117]

Parameter auf Projektebene	Parameter auf operativer Ebene	Übergreifende Parameter
Struktur der Entwicklungsaufgabe	Integrationsgrad	Entwicklungsaufgabe
Projektrandbedingungen	Geplanter Output	Anforderungen
Produzierte Stückzahl	Verfügbare Dokumente	Prozessergebnisse
Kunde	Struktur des Teilproblems	Neuheitsgrad der Entwicklung
Risiko	Operative Randbedingungen	Komplexität der Aufgabe
	Prozessverlauf bis zum aktuellen Punkt	

Auf diesen Parametern bauen die **Methoden zur strategischen und operativen Prozessplanung** auf. Auf der strategischen Ebene wählt der projektplanende Entwickler regelbasiert die auszuführenden Schritte der ersten und zweiten Hierarchieebene des FORFLOW-Prozessmodells aus und erstellt daraus den vorläufigen Projektplan. Auf der operativen Ebene bestimmt der projektausführende Entwickler die Reihenfolge der auszuführenden Schritte auf der dritten Hierarchieebene des FORFLOW-Prozessmodells mit Hilfe von zeitbasierten Design Structure Matrizen wie sie von BROWNING beschrieben werden [Bro01, S. 293], [MHJ+09, S. 56f.], [RKL+08, S. 2f.], [Roe11, S. 122ff.].

Bewertung: Die von ROELOFSEN im Rahmen des Forschungsverbundes FORFLOW erarbeitete Methodik erlauben eine präzise Klassifikation der aktuellen Entwicklungssituation und das darauf aufbauende dynamische Tailoring des vordefinierten FORFLOW-Prozessmodells. Der für die Planung des weiteren Vorgehens entwickelte Ansatz des Product Model Driven Development hat den Anspruch sowohl die erstellten Produktmodelle als auch den bisherigen Verlauf des Entwicklungsprozesses zu berücksichtigen, vernachlässigt aber die innere Struktur des zu entwickelnden Systems.

3.3.8 Modellgestütztes Rahmenwerk zur Planung und Steuerung adaptiver Systementwicklungsprozesse nach LÉVÁRDY

Um der zunehmenden Komplexität technischer Systeme und deren Entwicklungsumfelds, den Unsicherheiten hinsichtlich der genauen Anforderungen an das System als auch unvorhergesehenen Ereignissen während der laufenden Entwicklung effektiv begegnen zu können, entwickelt LÉVÁRDY ein am Systems Engineering ausgerichtetes Rahmenwerk zur Planung und Steuerung adaptiver Systementwicklungsprozesse. Kern des Rahmenwerks ist ein zweifach geteilter, adaptiver Analyse-Regel-Kreis wie in Bild 3-11 dargestellt.

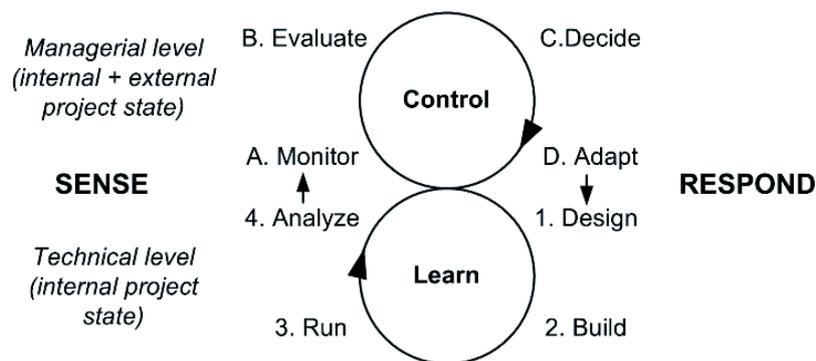


Bild 3-20 Adaptives System-Entwicklungs-Rahmenwerk [Lev06, S. 107]

Auf der unteren Ebene, dem technical level, wird das zu entwickelnde System in einem Wechselspiel aus Synthese und Analyse ausgestaltet, wobei immer wieder Zwischenergebnisse hinsichtlich der gestellten Anforderungen geprüft werden. Die Anforderungserfüllung wird auf der zweiten Ebene, der Management-Ebene, zusammen mit weiteren Faktoren aus dem Projektumfeld für die Bewertung des aktuellen Projektstatus genutzt, um auf dieser Basis ggf. notwendige Anpassungen der Entwicklung vorzunehmen. Für die Umsetzung dieses Analyse-Regel-Kreises in konkreten Entwicklungsprojekten schlägt Lévardy verschiedene Methoden vor, von denen die zwei wichtigsten eine **Methode zur Entscheidungsfindung in adaptiven Systementwicklungsprojekten** und eine **Methode zur Modellierung und Simulation adaptiver Systementwicklungsprozesse** sind [Lev06, S. 14ff.].

Die Methode zur **Entscheidungsfindung in adaptiven Systementwicklungsprojekten** basiert auf einer kontinuierlichen Chancen-Risiko-Bewertung anhand spezifischer technischer und managementorientierter Leistungskennzahlen. Bild 3-21 verdeutlicht, wie die Leistungskennzahlen den einzelnen Hierarchieebenen und Bestandteilen des zu entwickelnden Systems zugeordnet und zu Schlüsselkennzahlen verdichtet werden.

Auf der obersten Systemebene (Level 1) werden einige wenige Schlüsselkennzahlen, die Key Performance Parameter, festgelegt anhand derer der aktuelle Stand der Systementwicklung in technischer Hinsicht bewertet wird. Die Key Performance Parameter

werden entsprechend dem hierarchischen Aufbau des Systems in mess- und bewertbare Kenngrößen, die Measure of Performance, herunter gebrochen. Diese sind den einzelnen Modulen bzw. Komponenten des Systems zugeordnet (Level 2). Bei besonders komplexen Modulen bzw. Komponenten können diese nochmals unterteilt (Level 3) und deren Bestandteilen eigene Kenngrößen, die Technical Performance Measures, zugeordnet werden. Zur Überprüfung der Technical Performance Measures werden die sich nach den einzelnen Entwicklungstätigkeiten ergebenden Eigenschaften des zu entwickelnden Systems herangezogen (Level 4) [Lev06, S. 115ff.].

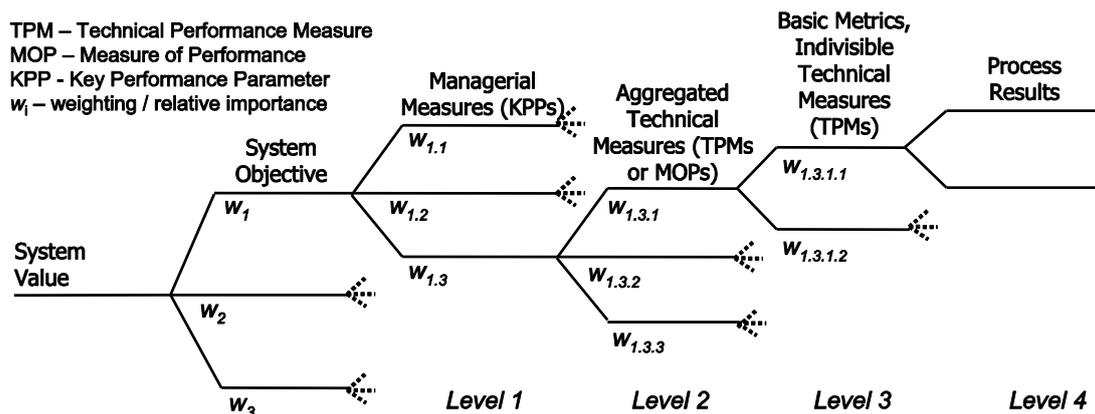


Bild 3-21 System von Schlüsselkennzahlen als Entscheidungskriterien [Lev06, S. 115]

Für jeden Key Performance Parameter wird eine Nutzenfunktion definiert, aus der der Nutzen bei Erfüllung, Nichterfüllung und Übererfüllung der für den Key Performance Parameter geltenden Anforderungen ermittelt werden kann. Aus dem gegenwärtigen Erfüllungsgrad der Anforderungen und der Nutzenfunktion wird nach dem Ansatz von BROWNING das aktuelle technische Risiko-Chancen-Verhältnis des jeweiligen Key Performance Parameters ermittelt [LB09, S. 600ff.]. Die einzelnen Risiko-Chancen-Verhältnisse werden in Form einer gewichteten Summe zum technischen Gesamt-Risiko-Chancen-Verhältnis R_{TP} zusammengefasst. Die Gewichtungsfaktoren in der Summe geben die jeweilige Relevanz des Key Performance Parameters aus Kundensicht an. Zusammen mit dem Risiko-Chancen-Verhältnis bezüglich der Projektlaufzeit R_S und dem Risiko-Chancen-Verhältnis bezüglich der Projektkosten R_C berechnet sich das Gesamt-Risiko-Chancen-Verhältnis des Systementwicklungsprojekts R_{PD} wiederum als gewichtete Summe aus R_{TP} , R_S und R_C . Hierfür sind die Nutzenfunktionen der Projektlaufzeit und der Projektkosten notwendig. Die Gewichtungen der einzelnen Risiko-Chancen-Verhältnis-Werte geben die Bedeutung von Zeit, Kosten und technischer Leistungsfähigkeit des zu entwickelnden Produkts aus Sicht des Projektmanagements an. Abhängig von R_{PD} entscheidet das Projektmanagement, ob ggf. die Entwicklungsziele oder der Entwicklungsprozess angepasst werden [Lev06, S. 113ff.].

Zur Entscheidungsunterstützung, wie der Entwicklungsprozess angepasst werden sollte, dient die **Methode zur Modellierung und Simulation adaptiver Systementwicklungsprozesse** „Adaptive System Development Process“ (ASDP). Hierbei werden ein-

zelne Entwicklungsschritte als Aktivitäten modelliert, die in Abhängigkeit des Zustands der Eingangsinformationen unterschiedliche Modi haben (vgl. Bild 3-22). Die möglichen Aktivitäten, ihre Modi und ihre Abhängigkeiten sind durch das Projektmanagement zu definieren.

Den Modi der Aktivitäten sind unterschiedliche Eigenschaften wie z.B. Dauer, Kosten oder Effektivität sowie Transformationsfunktionen zugeordnet. Die Transformationsfunktionen beschreiben, auf welche Art und Weise sich die Eigenschaften des zu entwickelnden Systems durch die Aktivität verändern. Dadurch kann die Entwicklung der Technical Performance Measures für unterschiedliche Aktivitätsfolgen und damit einhergehend das Gesamt-Chancen-Risiko-Verhältnis des aktuellen Systementwicklungsprojektzustands ermittelt werden. Dieses Verhältnis wird für die Ermittlung der Wahrscheinlichkeit von Iterationen im Entwicklungsprozess und, zusammen mit vorher festgelegten Risikoschwellenwerten, für die nutzenmaximierende Gewichtung der einzelnen Risiko-Chancen-Verhältnis-Werte von Zeit, Kosten und technischer Leistungsfähigkeit verwendet. Dadurch wird es möglich zukünftige Iterationen und an den Risikoschwellenwerten orientierte Entscheidungen des Projektmanagements bei der Simulation möglicher Entwicklungsprozesse zu berücksichtigen. Der effektivste Prozess wird nach der Simulation für die Anpassung des Entwicklungsprozesses verwendet [Lev06, S. 187ff.].

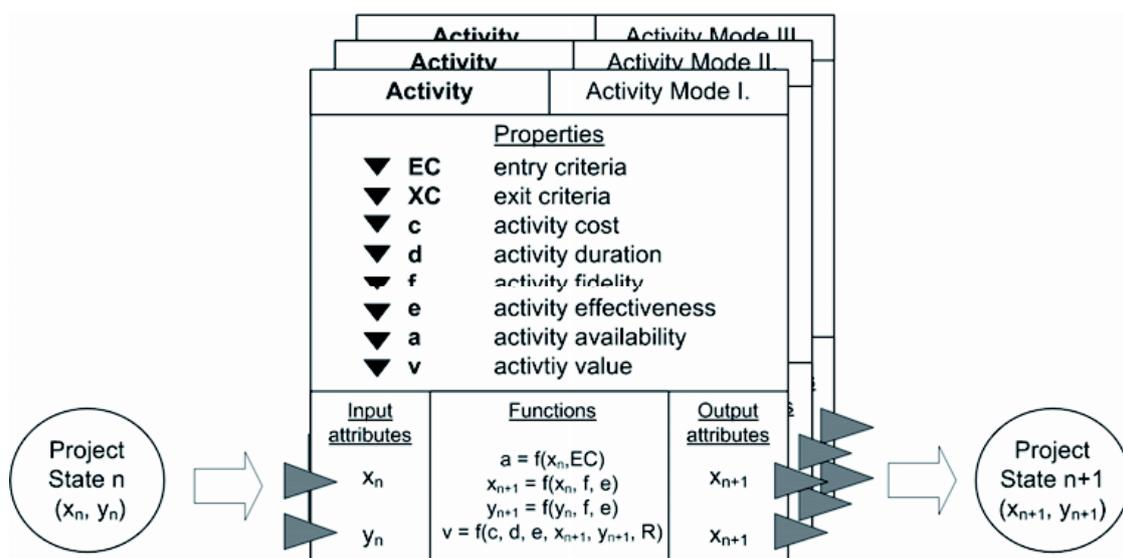


Bild 3-22 Beschreibung einer Aktivität in der ASDP-Methode [Lev06, S 188]

Bewertung: Das Rahmenwerk für adaptive Entwicklungsprozesse nach LÉVÁRDY kann als erster Schritt in Richtung des in dieser Arbeit adressierten Rahmenwerks für einen selbstoptimierenden Entwicklungsprozess gesehen werden. Die Verschiebung der Bedeutung von Zeit, Kosten und technischer Leistungsfähigkeit in Abhängigkeit des aktuellen Projektrisikos mit Hilfe verschiedener Nutzenfunktionen ermöglicht zwar keine vollständig selbstoptimierende aber zumindest adaptive Anpassung des Entwicklungsprozesses. Für die vollständige Umsetzung der Selbstoptimierung bedarf es weiterer

Mechanismen zur Berücksichtigung inhärenter Ziele, zur Auswahl von Projektzielen sowie zur Modifikation der Nutzenfunktionen. Als kritisch bezogen auf den notwendigen Aufwand sowie die Übertragbarkeit bisheriger Entwicklungstätigkeiten auf neue Problemstellungen ist wie bei dem Ansatz von MURR die verwendete Modellierungstechnik zu sehen. Im Vorfeld die durch einen Entwicklungsschritt bewirkten konkreten Änderungen der Eigenschaften des Entwicklungsgegenstands zu antizipieren ist nur schwer möglich.

3.4 Bewertung des Stands der Technik und Handlungsbedarf

Die Gegenüberstellung der in diesem Kapitel beschriebenen Ansätze des Stands der Technik mit den in Kapitel 2.7 definierten Anforderungen stellt sich wie folgt für die einzelnen Anforderungen dar:

A1 Ganzheitlichkeit: Insbesondere unter den systemischen Produktentwicklungsmodellen existieren einige sehr gute Ansätze für die Strukturierung der Subsysteme und Wechselwirkungen innerhalb des Produktentwicklungsprozesses. Hauptzweck dieser Modelle ist aber die Analyse des Entwicklungsgeschehens. Die auf der Analyse aufbauende Vorausplanung eines spezifischen Entwicklungsprozesses wird nicht adressiert.

A2 Interdisziplinäre Wechselwirkungen: Modelltechnische und prozesstechnische Abhängigkeiten zwischen den an der Entwicklung eines mechatronischen Systems beteiligten Fachdisziplinen können mit den bestehenden Modellierungstechniken abgebildet werden. Es existiert keine Technik die modell- und prozesstechnische Sicht vereint, was aber auch nicht unbedingt erforderlich ist. Kritisch ist, dass die bestehenden Planungs- und Steuerungsverfahren die konkret vorliegenden strukturellen Zusammenhänge des zu entwickelnden Systems kaum bzw. gar nicht nutzen.

A3: Dynamische Vorausplanung: Die bestehenden Planungs- und Steuerungsverfahren fokussieren meist eine auf der Klassifikation des zu entwickelnden Systems oder der Art des vorliegenden Entwicklungsproblems basierende Auswahl geeigneter Entwicklungsschritte und die darauf aufbauende Vorgangsfolgeplanung. Einige Ansätze lassen hierbei bereits eine mehrstufige Detaillierung der Klassifikation zu und wählen entsprechend geeignete Entwicklungsschritte aus. Eine Zeit-, Kosten-, Hilfsmittel- oder Ressourcenplanung wird wenn überhaupt nur bei wenigen Ansätzen unterstützt.

A4 Spezifität: Die Bereitstellung von spezifischen Handlungsanweisungen für die Entwickler wird insbesondere in solchen Ansätzen gut gelöst in denen auf Prozessbausteine zum Aufbau von Prozessnetzen zurückgegriffen wird. Die dort gewählten Detaillierungsgrade erlauben die Vorgabe der als nächstes zu bearbeitenden Aufgabe ohne die Problemlösungsstrategie einzugrenzen. Als problematisch ist bei diesen Verfahren aber die Übertragbarkeit auf bisher unbekannte Problemstellungen zu sehen.

A5 Problemunabhängigkeit: Das Rahmenwerk soll unabhängig von der Art des zu entwickelnden Systems oder der Entwicklungsproblemklasse angewendet werden kön-

nen. Die beinhalteten Werkzeuge und Vorgehensweisen müssen entsprechend auf allgemeinen Eigenschaften mechatronischer Systeme aufbauen und ausgehend von dieser Basis weitere z.B. branchen- oder problemklassenspezifische Informationen integrieren.

Problemunabhängigkeit und Spezifität der bereitgestellten Handlungsanweisungen sind bei den bisherigen Ansätzen gegenläufig und nicht zufriedenstellend gelöst. Auf der einen Seite stehen nicht problemunabhängigen Zugriffsmechanismen auf spezifischen Handlungsanweisungen. Auf der anderen Seite stellen die problemunabhängigen Ansätze nur unzureichend oder gar nicht spezifische Handlungsanweisungen bereit.

A6 Selbstoptimierung: Keiner der im Stand der Technik untersuchten Ansätze adressiert die endogene Anpassung der Prozessziele im Sinne des Paradigmas der Selbstoptimierung. Einzig das von LÉVÁRDY entwickelte Rahmenwerk macht einen ersten Schritt in diese Richtung. Der Ansatz hat nicht den Anspruch Selbstoptimierung umzusetzen und nimmt entsprechend keine endogene Modifikation der verwendeten Nutzenfunktionen vor.

A7 Akzeptanz/Transparenz: Einige der im Stand der Technik untersuchten Ansätze ermöglichen eine intuitiv verständliche Darstellung der modell- und prozesstechnischen Inhalte des Entwicklungsprozesses. Ähnlich wie bei den interdisziplinären Wechselwirkungen (Anforderung A2) gibt es aber auch hier kein Verfahren was beide Arten von Inhalten berücksichtigt bzw. abbildet.

A8 Effizienz: Die meisten der vorgestellten Verfahren und Methoden werden durch prototypische IT-Werkzeuge unterstützt und können für sich genommen entsprechend leicht angewendet werden. Oft sind die Verfahren und Methoden aber nicht konsequent genug mit den im Entwicklungsprozess verfügbaren Informationen integriert und bedeuten einen zusätzlichen Aufwand für die Prozessverantwortlichen.

Zusammenfassend wird deutlich, dass derzeit im Stand der Technik kein Ansatz existiert, der als Rahmenwerk für einen *selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* entsprechend der Anforderungen aus Kapitel 2.7 dienen kann (vgl. Zusammenfassung Bild 3-23). Auch eine einfache Kombination der bestehenden Ansätze würde dies nicht ermöglichen, da die Übertragung des Paradigmas der Selbstoptimierung auf die Planung, Ausführung sowie Überwachung und Steuerung des Entwicklungsprozesses bisher nicht adressiert wird. Weitere wesentliche Herausforderungen bei der Erfüllung der Anforderungen aus Kapitel 2.7 sind die konsequente Berücksichtigung der modell- und prozesstechnischen interdisziplinären Wechselwirkungen der an der Entwicklung beteiligten Fachdisziplinen sowie die Auflösung des Konflikts zwischen der Problemunabhängigkeit des Rahmenwerks und der Bereitstellung problemspezifischer Handlungsanweisungen für die Entwickler. Es besteht somit Handlungsbedarf für ein *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme*.

Bewertung der untersuchten Ansätze hinsichtlich ihrer Anforderungserfüllung ● voll erfüllt ◐ teilweise erfüllt ○ nicht erfüllt	Anforderungen							
	A1	A2	A3	A4	A5	A6	A7	A8
Syst. Produktentwicklungsmodelle								
Allgemeine Technologie nach ROPOHL	◐	◐	○	○	●	○	◐	○
Systemtechnik nach PATZAK	●	◐	◐	○	●	○	◐	○
ZOPH-Modell	●	●	○	○	●	○	◐	◐
Integrierte Produkterstellung nach EHRENSPIEL	◐	◐	○	○	●	○	○	○
iPeM	●	◐	○	◐	●	○	◐	○
Prozess-Modellierungstechniken								
ARIS	◐	◐	○	○	●	○	◐	◐
OMEGA	◐	◐	◐	○	●	○	●	◐
SADT	○	◐	○	○	●	○	○	○
IDEF	○	◐	◐	○	●	○	○	◐
Netzplantechnik	○	◐	◐	○	◐	○	○	●
System-Modellierungstechniken								
OMG SysML™	○	●	○	○	●	○	◐	●
CONSENS	○	●	○	○	●	○	●	●
Strukturmodellierung nach RIEPE	○	◐	○	○	◐	○	◐	○
Modelica®	○	◐	○	○	◐	○	◐	●
Planungs- und Steuerungsverfahren								
Situative Aufgaben- und Methodenauswahl nach PONN	◐	○	◐	●	○	○	○	◐
Methode zur flexiblen Gestaltung nach BICHLMAIER	○	○	◐	◐	○	○	◐	◐
Adaptive Planung und Steuerung nach MURR	◐	○	◐	◐	○	○	◐	○
Entwicklungsprozessplanung nach REDENIUS	○	○	◐	●	◐	○	●	◐
Adaptive Entwicklungsprozessplanung nach SCHUMANN	◐	○	◐	○	●	○	○	○
Dynamische Planung und Steuerung nach DEMERS	○	○	◐	○	●	○	◐	○
Flexible Prozessunterstützung - FORFLOW	◐	○	◐	◐	◐	○	◐	◐
Rahmenwerk für adaptive Entwicklungsprozesse	◐	◐	◐	◐	○	◐	◐	○

Bild 3-23 Bewertung des Stands der Technik

4 Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess

Der in Kapitel 3.4 offengelegte Handlungsbedarf macht deutlich, dass es derzeit kein *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozesses für fortschrittliche mechatronische Systeme* entsprechend den in Kapitel 2.7 definierten Anforderungen gibt. Ein solches Rahmenwerk ist der Kern der vorliegenden Arbeit. Seine Bestandteile werden in diesem Kapitel auf theoretischer Basis erläutert. Die Verifikation anhand eines ausgewählten Beispiels ist Gegenstand des nachfolgenden Kapitels 5.

4.1 Das Rahmenwerk im Überblick

Bild 4-1 zeigt die drei Hauptbestandteile des Rahmenwerks. Es beinhaltet sowohl die theoretische Übertragung als auch die Hilfsmittel und Vorgehensweisen zur praktischen Anwendung des Paradigmas der Selbstoptimierung auf den Entwicklungsprozess fortschrittlicher mechatronischer Systeme.

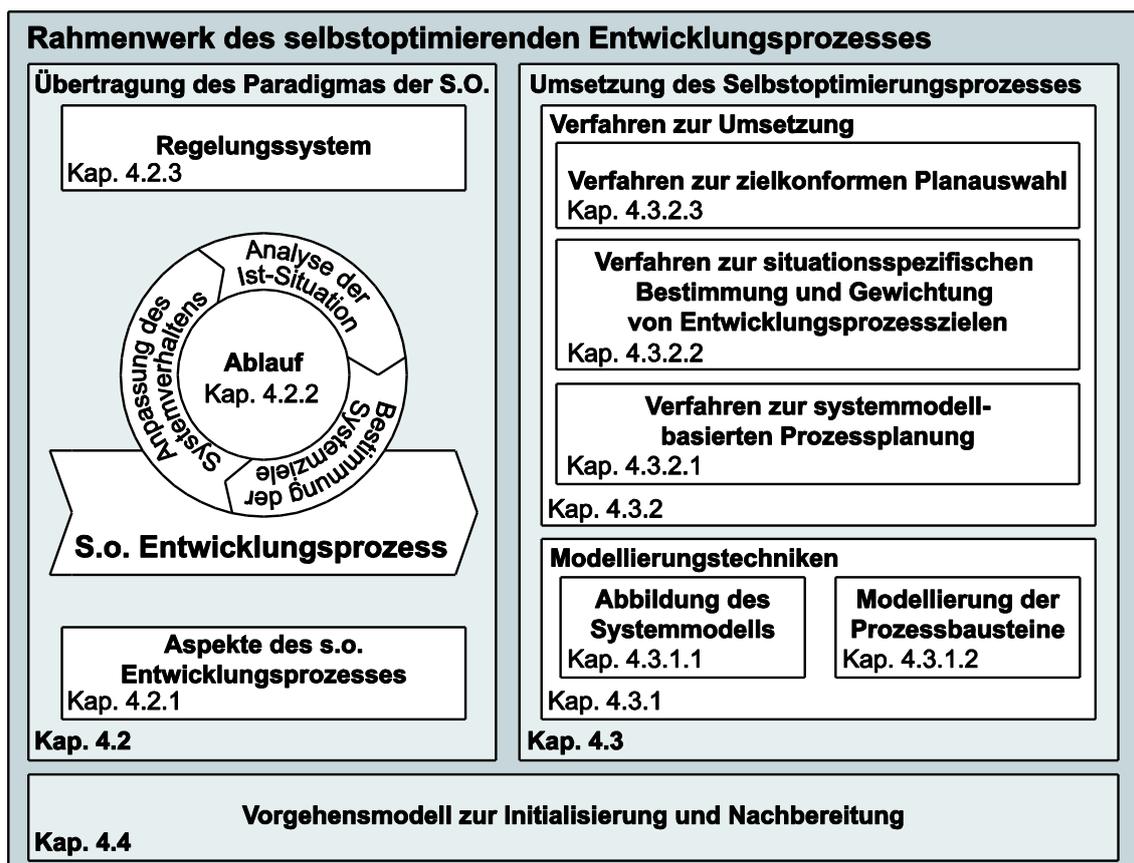


Bild 4-1 Bestandteile des Rahmenwerks für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme

Der erste Hauptbestandteil ist die theoretische **Übertragung des Paradigmas der Selbstoptimierung** auf den Entwicklungsprozess. Zunächst werden die bisher für technische Systeme definierten *Aspekte* der Selbstoptimierung auf den Entwicklungsprozess übertragen und in einem systemischen Gesamtmodell dargestellt (Kapitel 4.2.1). Dies bildet das Fundament für die Definition des *Ablaufs der Selbstoptimierung* des Entwicklungsprozesses; den sogenannten Selbstoptimierungsprozess (Kapitel 4.2.2). Der Selbstoptimierungsprozess selbst wird durch die Komponenten des *Regelungssystems* des selbstoptimierenden Entwicklungsprozesses umgesetzt (Kapitel 4.2.3).

Den zweiten Hauptbestandteil des Rahmenwerks bilden die **Werkzeuge zur Umsetzung** des selbstoptimierenden Entwicklungsprozesses (Kapitel 4.3). Sie setzen die theoretischen Grundlagen praktisch um und ermöglichen deren Anwendung für tatsächliche Entwicklungsprojekte. Die Werkzeuge gliedern sich in Modellierungstechniken und Verfahren. Zu den Modellierungstechniken gehören eine Technik zur *Abbildung des Systemmodells* des zu entwickelnden Systems (Kapitel 4.3.1.1) und eine Technik zur Modellierung von Entwicklungstätigkeiten in Form von *Prozessbausteinen* (Kapitel 4.3.1.2). Sie dienen dazu sämtliche Informationen, die für die Anwendung des Paradigmas der Selbstoptimierung auf den Entwicklungsprozess benötigt werden, konsolidiert bereitzustellen. Auf diesen Informationen setzen dann die den selbstoptimierenden Entwicklungsprozess umsetzenden Verfahren auf. Die in den einzelnen Phasen der Selbstoptimierung zum Einsatz kommenden Verfahren gliedern sich in das *Verfahren zur systemmodellbasierten Prozessplanung* (Kapitel 4.3.2.1), das *Verfahren zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen* (Kapitel 4.3.2.2) und das *Verfahren zur zielkonformen Planauswahl* (Kapitel 4.3.2.3).

Den dritten Bestandteil des Rahmenwerks bildet das **Vorgehensmodell zur Initialisierung und Nachbereitung** (Kapitel 4.4) selbstoptimierender Entwicklungsprozesse. Mit diesem Vorgehensmodell werden das Rahmenwerk für ein konkretes Entwicklungsprojekt eines Unternehmens konfiguriert und nach Beendigung des Entwicklungsprojekts die gesammelten Erfahrungswerte für künftige Projekte aufbereitet.

4.2 Übertragung des Paradigmas der Selbstoptimierung auf den Entwicklungsprozess fortschrittlicher mechatronischer Systeme

Nachfolgend werden analog zur Darstellung der Selbstoptimierung in der Problemanalyse in Kapitel 2.3 die theoretischen Grundlagen des Rahmenwerks gelegt. Sie schaffen das begriffstechnische, ablauftechnische und strukturelle Fundament für die Anwendung des Paradigmas der Selbstoptimierung auf den Entwicklungsprozess fortschrittlicher mechatronischer Systeme.

4.2.1 Aspekte des selbstoptimierenden Entwicklungsprozesses fortschrittlicher mechatronischer Systeme

Für die Übertragung des Paradigmas der Selbstoptimierung auf den Entwicklungsprozess fortschrittlicher mechatronischer Systeme werden nachfolgend die in Kapitel 2.3.1 dargestellten Aspekte selbstoptimierender Systeme auf den Entwicklungsprozess transferiert. Der Transfer erfolgt in Anlehnung an das ZOPH-Modell nach NEGELE [Neg06, S. 138ff.] (vgl. Abschnitt 3.1.3).

Das **System** ist der selbstoptimierende Entwicklungsprozess nach Bild 4-2. Die **Elemente**, aus denen er besteht, sind das *Prozessmanagement* und die *Entwicklungstätigkeiten*. Der Entwicklungsprozess ist durch eine gedankliche **Systemgrenze** vom *Entwicklungsprozessumfeld* getrennt. Das Entwicklungsprozessumfeld wird in die vier Bereiche *Unternehmensumfeld*, *unternehmensexterne Stakeholder*, *Entwicklungsgegenstand* und *weiteres Systemumfeld* unterteilt.

Sämtliche Elemente des Entwicklungsprozesses und des Entwicklungsprozessumfeldes sind über gerichtete und ungerichtete Beziehungen miteinander verbunden³⁰. Die über die Systemgrenze hinausgehenden, gerichteten Beziehungen werden entsprechend ihrer Richtung in Eingangs- und Ausgangsgrößen unterschieden. Alle auf den Entwicklungsprozess wirkenden Beziehungen werden als **Einflüsse** bezeichnet. Sie können Eingangsgrößen sein oder vom Entwicklungsprozess selbst kommen³¹. Beispiele sind der gewünschte Liefertermin oder das erlaubte Entwicklungsbudget. Sie kommen je nach Entwicklungssituation aus dem Unternehmensumfeld oder von den unternehmensexternen Stakeholdern, wie bspw. den Kunden. Den Entwicklungsprozess negativ beeinträchtigende Einflüsse sind *Störgrößen*. Ihnen gegenüber stehen die unterstützenden, gezielten bzw. geplanten Einflüsse, die sogenannten *Vorgaben* und *Objekte*. Die Summe der Ausprägungen der Einflüsse zu einem spezifischen Zeitpunkt ergibt die *Entwicklungssituation*.

Die Entwicklungstätigkeiten sind über Eingangsobjekt-Ausgangsobjekt-Zusammenhänge miteinander verbunden und können selbst wiederum aus mindestens zwei miteinander gekoppelten Entwicklungstätigkeiten bestehen. Sie werden durch bzw. mit Hilfe von Handlungsobjekten aus dem Unternehmensumfeld ausgeführt.

Die Handlungsobjekte werden ihrer Art nach in *Ressourcen* und *Hilfsmittel* eingeteilt. Zu den Ressourcen zählen bspw. personelle Ressourcen oder Maschinen. Hilfsmittel sind bspw. Methoden oder Software-Werkzeuge. Die Handlungsobjekte werden in der Regel durch das Unternehmensumfeld bereitgestellt.

³⁰ Auf die explizite Darstellung der Beziehungen zwischen den Elementen des Entwicklungsprozessumfeldes wurde in Bild 4-2 zu Gunsten einer einfacheren Lesbarkeit verzichtet.

³¹ Es wird davon ausgegangen, dass die aus dem weiteren Systemumfeld stammenden Einflüsse indirekt über die anderen drei Bereiche des Entwicklungsprozessumfeldes auf den Entwicklungsprozess wirken.

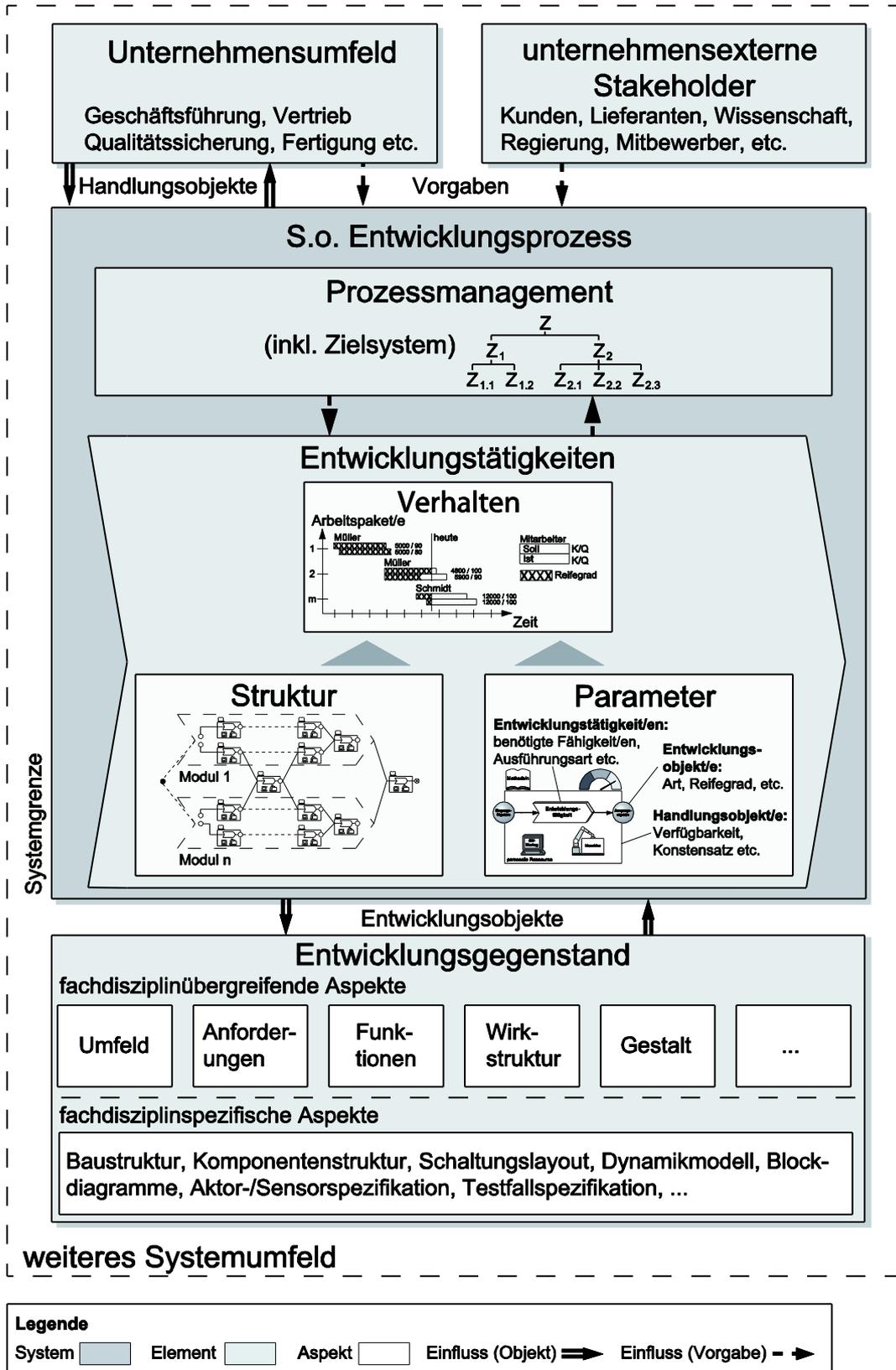


Bild 4-2 Systemisches Gesamtmodell des selbstoptimierenden Entwicklungsprozesses

Die Eingangs- und Ausgangsgrößen der Entwicklungstätigkeiten sind die Entwicklungsobjekte. Sie sind entsprechend der Definition des Objektsystems nach NEGELE:

„...Voraussetzung, (zu bearbeitender) Gegenstand bzw. Ergebnis der im Entwicklungsprozeß durchgeführten Aktivitäten.“ [Neg06, S. 147]

Ihre Gesamtheit beschreibt den Entwicklungsgegenstand. Sie können hinsichtlich der durch sie abgebildeten Fachdisziplinen in fachdisziplinübergreifende und fachdisziplinspezifische Entwicklungsobjekte unterteilt werden.

Die Transformation der Entwicklungsobjekte durch die Entwicklungstätigkeiten erfolgt entsprechend dem durch das Prozessmanagement festgelegten Zielsystem. Die im Zielsystem des Entwicklungsprozesses zusammengefassten **Ziele** beschreiben das gewünschte, geforderte oder zu vermeidende Verhalten des Entwicklungsprozesses bezogen auf die Ausführung der aktuell durchgeführten oder noch durchzuführenden Entwicklungstätigkeiten. Beispiele sind die geforderte Prozessgüte, der zeitlich-inhaltliche Fortschritt oder der Wunsch nach einer minimalen Anzahl von Personalwechseln. Sie werden von den „Produktzielen“ unterschieden, die die Anforderungen an den Entwicklungsgegenstand beschreiben [vgl. Neg06, S. 143]. Die Ziele werden durch das Prozessmanagement auf Basis der Vorgaben aus dem Unternehmensumfeld, den Vorgaben der unternehmensexternen Stakeholder und den Eigenschaften des Entwicklungsgegenstandes festgelegt. Sie werden in externe, inhärente und interne Ziele unterteilt:

- **Externe Ziele** werden von außerhalb der Systemgrenze in den Entwicklungsprozess eingebracht. Hierzu gehören zum einen Ziele, die aus den externen Vorgaben resultieren. Dies sind z.B. die Ziele bezüglich des Endtermins, der Prozessgüte oder der zulässigen finanziellen Aufwände. Zum anderen resultieren die externen Ziele aus den Eigenschaften des Entwicklungsgegenstandes³². Bspw. erfordern höher komplexe Module des zu entwickelnden Systems eine höhere Gewichtung des Ziels „Maximierung der Prozessgüte“.
- Die **inhärenten Ziele** spiegeln den Zweck des Entwicklungsprozesses wider. Sie beschreiben die Ziele jeder Entwicklung, wie z.B. die „Minimierung der Anzahl von Personalwechseln“ während der Entwicklung.
- Die zu einem spezifischen Zeitpunkt für den Entwicklungsprozess geltenden Ziele sind die **internen Ziele** des Entwicklungsprozesses. Ihre Menge, ihr Gefüge und die jeweilige Gewichtung werden im selbstoptimierenden Entwicklungsprozess abhängig von der Entwicklungssituation endogen angepasst.

Die Gesamtheit aus Entwicklungstätigkeiten, Handlungs- und Entwicklungsobjekten und ihre Beziehungen untereinander ergeben die **Struktur** des Entwicklungsprozesses. Ihre Eigenschaften stellen die **Parameter** des Entwicklungsprozesses dar. Zusammen bestimmen sie dessen **Verhalten**. Das Verhalten beschreibt die Ausführung der Ent-

³² Da die von den Entwicklungszielen zu unterscheidenden Produktziele für den Entwicklungsgegenstand definiert werden, wirken die Produktziele indirekt auf die inhärenten Entwicklungsprozessziele.

wicklungstätigkeiten zu einem spezifischen Zeitpunkt und die durch die Ausführung bewirkten Transformationen der Eingangs- in die Ausgangsgrößen.

Die vorstehenden Ausführungen zeigen die Übertragbarkeit der für technische Systeme definierten Aspekte der Selbstoptimierung auf den Entwicklungsprozess fortschrittlicher mechatronischer Systeme. Die Definition der Selbstoptimierung (vgl. Abschnitt 2.3) kann entsprechend für Entwicklungsprozesse adaptiert werden:

„Unter Selbstoptimierung (self-optimization) eines Entwicklungsprozesses wird die endogene Anpassung der Ziele des Entwicklungsprozesses auf veränderte Einflüsse und die daraus resultierende zielkonforme autonome Anpassung der Parameter und ggf. der Struktur und somit des Verhaltens des Entwicklungsprozesses verstanden.“

4.2.2 Ablauf des Selbstoptimierungsprozesses

Die Selbstoptimierung des Entwicklungsprozesses vollzieht sich in einer Abfolge aus drei Aktionen (vgl. Abschnitt 2.3.2). Ihre wiederkehrende Ausführung sorgt für eine endogene Anpassung des Zielsystems des Entwicklungsprozesses und darauf aufbauend für die Anpassung von dessen Verhalten. Die drei Aktionen für die Selbstoptimierung des Entwicklungsprozesses sind:

- 1) **Analyse der Ist-Situation:** Hierbei werden über die Überwachungsgrößen der aktuelle interne Zustand des Entwicklungsprozesses sowie über die Einflüsse die aktuelle Entwicklungssituation erfasst. Anhand der Überwachungsgrößen (z.B. Sachfortschritt der Entwicklungsobjekte, verstrichene Entwicklungszeit, bisher angefallene Kosten etc.) wird geprüft, ob der aktuell auszuführende Entwicklungsprozessplan und die internen Ziele des Entwicklungsprozesses untererfüllt, erfüllt oder übererfüllt sind bzw. ob sich dies in der Zukunft abzeichnet. Zudem werden die Verfügbarkeit der im Entwicklungsprozess einzusetzenden Handlungs- und Entwicklungsobjekte überprüft und ggf. vorhandene Veränderungen der Vorgaben erfasst. Basierend auf diesen Informationen werden bei Bedarf alternative Prozesspläne generiert.
- 2) **Bestimmung der Systemziele:** In diesem Schritt wird geprüft, ob das aktuelle interne Zielsystem der Entwicklungssituation angemessen ist. Die Menge und Gewichtung der Ziele wird auf Basis der in der Analyse der Ist-Situation gesammelten Informationen (Abweichungen der Überwachungsgrößen, Objekt-Einflüsse und Vorgaben-Einflüsse) überprüft und bei Bedarf modifiziert. Die wesentlichen Mechanismen für die Modifikation sind die Auswahl und Anpassung. Die in der Analyse der Ist-Situation generierten möglichen Prozesspläne werden anhand des internen Zielsystems bewertet. Passt einer der alternativen Prozesspläne besser als der aktuell ausgeführte zum internen Zielsystem wird eine Anpassung des Systemverhaltens angestoßen.

- 3) **Anpassung des Systemverhaltens:** Die Anpassung des Verhaltens des Entwicklungsprozesses wird durch die Vorgabe eines neuen Prozessplans bewirkt. Dieser kann die gleiche Struktur des vorherigen aber veränderte Parameter, eine veränderte Struktur mit gleichen Parametern oder eine geänderte Struktur mit geänderten Parametern haben.

Die vorstehend beschriebenen Aktionen des Selbstoptimierungsprozesses müssen nicht sequentiell in der angegebenen Reihenfolge durchlaufen werden. Abhängig davon, woher der Auslöser für die Ausführung des Selbstoptimierungsprozesses kommt, werden sie in unterschiedlicher Reihenfolge und unterschiedlichem Umfang durchlaufen. Die verschiedenen möglichen Auslöser haben dabei eine unterschiedliche Dynamik und bewirken auf unterschiedlichen Ebenen eine Anpassung des Entwicklungsprozesses. Beispiele für ausgewählte Auslöser sind:

- **Veränderung der strategischen Vorgaben:** Normalerweise steht die gewählte Strategie-Option eines Entwicklungsprojekts fest. Wird sie aber während des Entwicklungsverlaufs dennoch geändert, hat dies direkten Einfluss auf das Zielsystem des Entwicklungsprozesses und bewirkt einen Wechsel des auszuführenden Prozessplans. Dieser Auslöser ist als sehr selten anzusehen und kommt meist nur bei langfristigen Entwicklungsprojekten vor.
- **Veränderte Eigenschaften des Entwicklungsgegenstandes:** In den meisten Entwicklungsprojekten stehen die Struktur und einzusetzenden Technologien für das zu entwickelnde System zu Beginn nicht endgültig fest und unterliegen während der Entwicklung entsprechenden Änderungen. Diesen Änderungen muss im Entwicklungsprozess Rechnung getragen werden. Veränderte Eigenschaften des Entwicklungsgegenstandes bewirken i.d.R. sowohl eine Anpassung der Ziele als auch des Prozessplanes. Sie sind ein häufiger Auslöser des Selbstoptimierungsprozesses wobei ihre Häufigkeit im Verlauf der Entwicklung mit zunehmendem Reifegrad des zu entwickelnden Produkts abnimmt.
- **Veränderungen der Handlungsobjekte:** Während der Entwicklung auftretende Veränderungen der Handlungsobjekte sind bspw. der krankheitsbedingte Ausfall einer personellen Ressource oder der Erwerb eines zusätzlichen Hilfsmittels (z.B. Softwarelizenz, Prüfstand etc.). Veränderungen der Handlungsobjekte erzwingen bzw. ermöglichen alternative Prozessverläufe, wobei das Zielsystem unverändert bleibt. Bspw. können beim Ausfall einer personellen Ressource Entwicklungstätigkeiten nicht mehr parallel ausgeführt werden und müssen stattdessen nacheinander abgearbeitet werden. Eine Veränderung der Handlungsobjekte ist als häufiger als die Veränderung der strategischen Vorgaben und als weniger häufig als die Veränderung des Entwicklungsgegenstandes anzunehmen.
- **Planabweichungen:** Weicht der aktuelle Verlauf des Entwicklungsprozesses vom geplanten ab bewirkt dies eine Anpassung der Ziele und des Prozessplanes wenn

der Ursrungsplan mit den verbleibenden Mitteln nicht mehr abgearbeitet werden kann. Dieser Auslöser ist bei Entwicklungsprojekten als der häufigste anzunehmen.

Unabhängig davon welcher Auslöser den Selbstoptimierungsprozess anstößt, ist festzulegen, wie stark auf Veränderungen der Entwicklungssituation oder des internen Zustands des Entwicklungsprozesses reagiert wird. Grundsätzlich gilt, dass Entwicklungsprojekte mit hohem Risiko eng gesteuert werden und damit die den Selbstoptimierungsprozess umsetzenden Komponenten eine hohe Empfindlichkeit haben sollten. Bei Entwicklungsprojekten mit geringem Risiko ist dies entsprechend umgekehrt.

4.2.3 Regelungssystem des selbstoptimierenden Entwicklungsprozesses

Für die Realisierung des im vorangegangenen Kapitel 4.2.2 beschriebenen Selbstoptimierungsprozesses bedarf es eines entsprechenden Regelungssystems. Die Struktur eines solchen Regelungssystems ist in Bild 4-3 dargestellt. Es ist ein Netzwerk mehrerer, miteinander verzahnter und sich gegenseitig beeinflussender Regelkreise. Das Regelungssystem ermöglicht die endogene Anpassung der Ziele des Entwicklungsprozesses und die darauf aufbauende Anpassung dessen Verhaltens im Sinne der Selbstoptimierung. Seine Funktionalität bezieht sich rein auf das Prozessmanagement, d.h. die Elemente außerhalb der in Kapitel 4.2.1 beschriebenen Systemgrenze werden nicht beeinflusst. Das Regelungssystem besteht aus den vier Komponenten: *Überwacher*, *Steuerung*, *Planer* sowie *Ziel- und Ausführungsbestimmung*. Abhängig von der durch den Entwicklungsgegenstand, das Unternehmensumfeld und die unternehmensexternen Stakeholder geprägten Entwicklungssituation greifen diese vier Komponenten wahlweise direkt in die Ausführung des Entwicklungsprozesses ein oder indirekt über die Modifikation von dessen Zielsystem. Wesentliche Ein- und Ausgangsgrößen des selbstoptimierenden Entwicklungsprozesses sind hierbei die fachdisziplinübergreifenden als auch fachdisziplinspezifischen Entwicklungsobjekte.

Unter den vier Komponenten der Regeleinrichtung des selbstoptimierenden Entwicklungsprozessmanagements kommt dem **Überwacher** die Aufgabe zu, den Fortschritt des Entwicklungsprozesses und die Eigenschaften des Entwicklungsgegenstandes zu erfassen. Basierend auf einem Soll-Ist-Vergleich der Überwachungsgrößen (z.B. Sachfortschrittsgrad der Entwicklungsobjekte, verstrichene Entwicklungszeit, bisher angefallene Kosten etc.) werden die Abweichungen von der geplanten Ausführung des Entwicklungsprozesses identifiziert. Diese werden zusammen mit den Eigenschaften der Entwicklungsobjekte an die *Ziel- und Ausführungsbestimmung* zur Entscheidung über ggf. notwendige Änderungen am Prozessverlauf übergeben. Der Sachfortschrittsgrad

der Entwicklungsobjekte wird zudem an die *Steuerung* für die Freigabe der als nächstes auszuführenden Entwicklungstätigkeiten übermittelt³³.

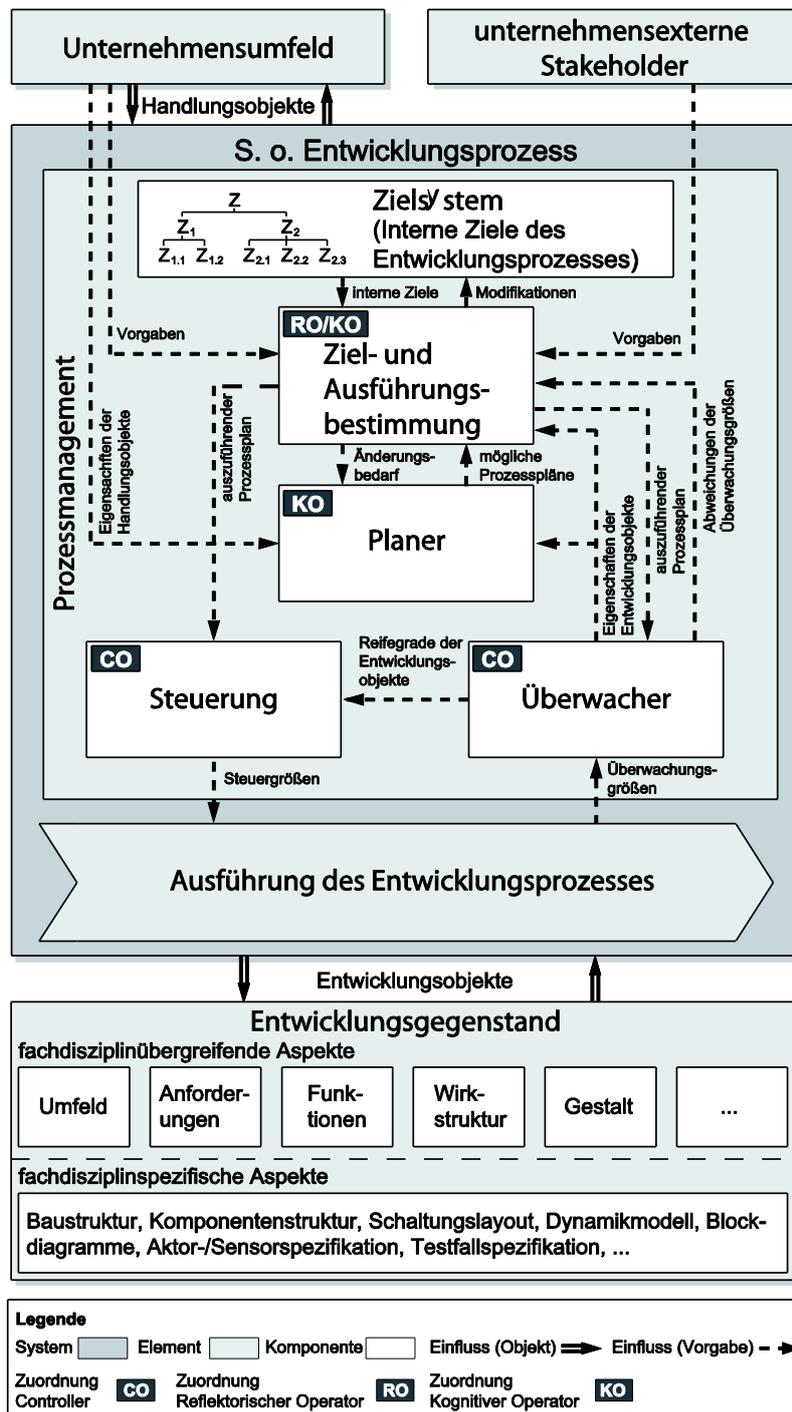


Bild 4-3 Regelungssystem des selbstoptimierenden Entwicklungsprozesses

³³ Es sei an dieser Stelle darauf hingewiesen, dass im Rahmen der vorliegenden Arbeit bei der Erfassung des Sachfortschrittgrads der Entwicklungsobjekte deren Qualität bezogen auf die Produktziele nicht explizit geprüft wird. Eine parallele Optimierung des Produkts liegt folglich nicht im Fokus des selbstoptimierenden Entwicklungsprozesses. Das in der vorliegenden Arbeit adressierte Rahmenwerk folgt dem Ansatz, durch eine hohe Prozessgüte eine hohe Produktgüte zu erreichen.

Aufgabe der **Steuerung** ist es, für die Entwickler die als nächstes auszuführenden Entwicklungstätigkeiten über die Steuergrößen freizugeben und falls die *Ziel- und Ausführungsbestimmung* einen neuen Prozessplan vorgibt die laufende Ausführung des Entwicklungsprozesses entsprechend anzupassen. Falls die tatsächliche Ausführung des Entwicklungsprozesses vom ursprünglichen Prozessplan abweicht und kein neuer Plan vorliegt, ist es Aufgabe der *Steuerung* die Freigabe der als nächstes auszuführenden Entwicklungstätigkeiten derart zu erteilen, dass die Entwickler zum ursprünglichen Plan zurückkehren.

Die Aufgabe des **Planers** ist die Generierung möglicher Prozesspläne, mit denen die aktuellen Entwicklungsobjekte in die vollständige Beschreibung eines technischen Systems überführt werden können. Zu diesem Zweck werden die Eigenschaften der Entwicklungsobjekte analysiert und darauf aufbauend eine Inhalts-, Vorgangsfolgen-, Termin-, Kosten-, Hilfsmittel- und Ressourcenplanung für die auszuführenden Entwicklungstätigkeiten vorgenommen. Das Ergebnis sind mehrere mögliche Prozesspläne mit unterschiedlichen Eigenschaften bspw. bezogen auf Dauer, Kosten und zu erwartende Prozessgüte.

Welcher der möglichen Pläne ausgeführt werden soll wird durch die **Ziel- und Ausführungsbestimmung** festgelegt. Sie stellt die Entscheidungskomponente des selbstoptimierenden Entwicklungsprozesses dar, die beschließt, ob Änderungsbedarf besteht und ggf. eine Neuplanung des Entwicklungsprozesses angestoßen werden muss. Die Zielauswahl und -gewichtung geschieht anhand der von der *Überwachung* erhaltenen Informationen über Abweichungen vom ursprünglichen Prozessplan, der Analyse der Eigenschaften der Entwicklungsobjekte und der Analyse der Einflüsse aus dem Unternehmensumfeld bzw. von den unternehmensexternen Stakeholdern. Das auf dieser Basis gebildete interne Zielsystem wird gegen die Gesamtheit der vom Planer erhaltenen möglichen Prozesspläne gespiegelt und der für die gegenwärtige Entwicklungssituation am besten geeignete Plan ausgewählt. Dieser wird an die *Steuerung* übergeben und die Ausführung des Entwicklungsprozesses entsprechend angepasst.

Generell können die vier beschriebenen Komponenten des Regelungssystems den drei Ebenen des Operator-Controller-Moduls (OCM) nach Kapitel 2.3.3 zugeordnet werden (vgl. Bild 4-3). Die Zuordnung ist qualitativ und darf in Teilen nicht uneindeutig sein, da für das Regelungssystem des selbstoptimierenden Entwicklungsprozesses keine harten Echtzeitanforderungen gelten wie für technische Systeme. Dem Controller als der untersten Ebene des OCM können die Steuerung und die Überwachung zugeordnet werden. Zusammen geben sie die Steuergrößen für die zu regelnde Strecke als Reaktion auf Abweichungen vom gewünschten Systemverhalten aus und bilden somit den motorischen Kreis des Regelungssystems. Dem reflektorischen Operator ist derjenige Teil der Ziel- und Ausführungsbestimmung zuzuordnen, welcher der Erfassung der Entwicklungssituation und der Entscheidung über Änderungsbedarf am Entwicklungsprozess dient. Der für die Zielauswahl und -gewichtung zuständige Teil der Ziel- und Ausführungsbestimmung bildet zusammen mit dem Planer den Kognitiven Operator.

Neben seiner Übertragbarkeit auf das OCM verbindet das beschriebene Regelungssystem des selbstoptimierenden Entwicklungsprozesses den Selbstoptimierungsprozess nach Kapitel 2.3.2 mit den Aufgabenkomplexen des Projektmanagements nach Kapitel 2.5.1. Bild 4-4 visualisiert diesen Brückenschlag.

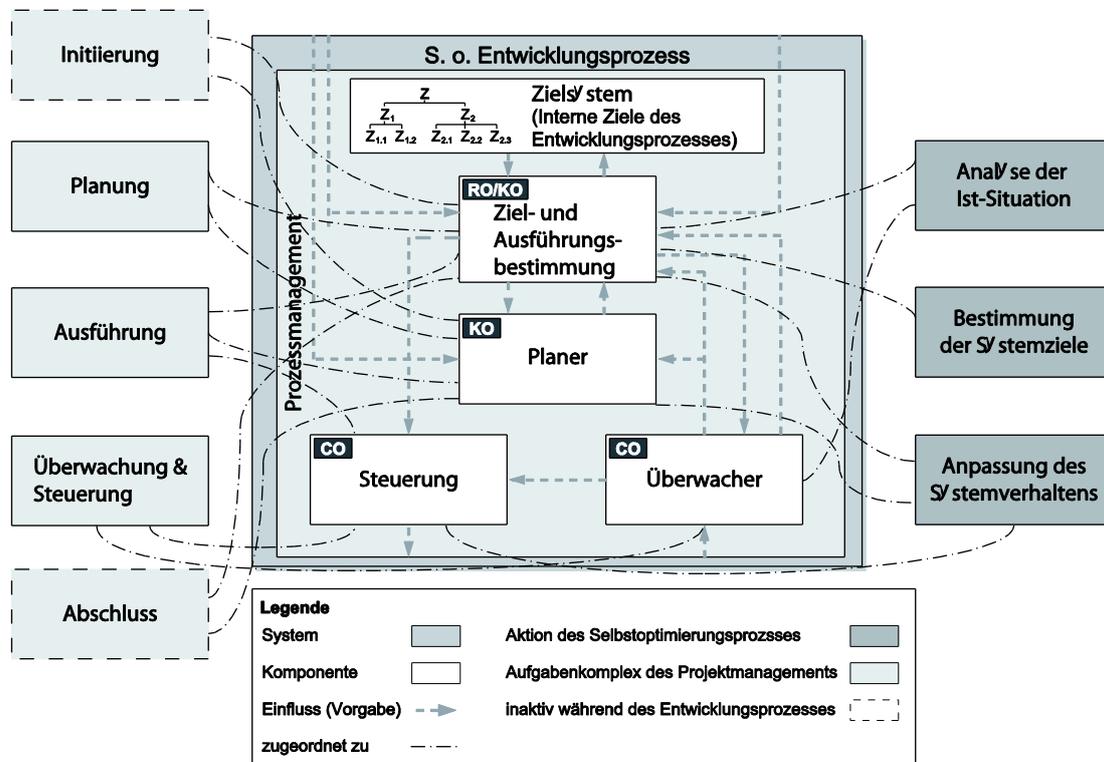


Bild 4-4 Zuordnung der Aufgabenkomplexe des Projektmanagements und der Aktionen des Selbstoptimierungsprozesses zu den Komponenten des Regelungssystems des selbstoptimierenden Entwicklungsprozessmanagements

Auf der linken Seite des Bildes sind die Aufgabenkomplexe des Projektmanagements dargestellt, auf der rechten die Aktionen des Selbstoptimierungsprozesses. Die gestrichelten Linien um die Aufgabenkomplexe „Initiierung“ und „Abschluss“, die zur Initiierung, Konfiguration und Nachbereitung eines Entwicklungsprojektes dienen (vgl. Abschnitt 2.5.1), machen deutlich, dass diese Tätigkeiten im laufenden Entwicklungsprozess in den Hintergrund treten. Sie werden im Vorgehensmodell zur Initialisierung und Nachbereitung des selbstoptimierenden Entwicklungsprozesses in Kapitel 4.4 beschrieben. Die durchgehend umrandeten Aufgabenkomplexe werden durch die Regeleinrichtung während des laufenden Entwicklungsprozesses umgesetzt. Gleiches gilt für die Aktionen des Selbstoptimierungsprozesses.

4.3 Umsetzung des Selbstoptimierungsprozesses

Zur Umsetzung des in Kapitel 4.2.2 beschriebenen Selbstoptimierungsprozesses bedarf es verschiedener **Verfahren** (Kapitel 4.3.2), die durch die Komponenten der in Kapitel 4.2.3 beschriebenen Regeleinrichtung des selbstoptimierenden Entwicklungsprozesses

ineinandergreifend ausgeführt werden. In diesem Kapitel werden daher die im Rahmen der vorliegenden Arbeit für die Umsetzung des Selbstoptimierungsprozesses zum Teil neuen, als auch aus bestehenden Ansätzen heraus entwickelten Verfahren zur *systemmodellbasierten Prozessplanung* (Kapitel 4.3.2.1), zur *situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen* (Kapitel 4.3.2.2) sowie zur *zielkonformen Planauswahl* (Kapitel 4.3.2.3) vorgestellt³⁴. Alle drei Verfahren stützen sich auf zwei **Modellierungstechniken** (Kapitel 4.3.1). Die eine dient der Abbildung aller fachdisziplinübergreifend relevanten Informationen über den Entwicklungsgegenstand in einem zentralen *Systemmodell* (Kapitel 4.3.1.1), die andere der Modellierung der im Entwicklungsprozess auszuführenden Tätigkeiten und deren Eigenschaften in Form von *Prozessbausteinen* (Kapitel 4.3.1.2). Nachfolgend werden zuerst die Modellierungstechniken als Grundbausteine der Verfahren zur Umsetzung des Selbstoptimierungsprozesses dargestellt.

4.3.1 Modellierungstechniken

In diesem Kapitel werden auf Basis bestehender Ansätze eine Modellierungstechnik für das Systemmodell eines fortschrittlichen mechatronischen Systems und eine Modellierungstechnik für Prozessbausteine, aus denen der Entwicklungsprozess fortschrittlicher mechatronischer Systeme aufgebaut werden kann, entwickelt. Die Anwendung beider Modellierungstechniken bildet die Informationsbasis für die im anschließenden Kapitel 4.3.2 erläuterten Verfahren zur Umsetzung des selbstoptimierenden Entwicklungsprozesses.

4.3.1.1 Abbildung des Systemmodells

Für die vorliegende Arbeit wird davon ausgegangen, dass die fachdisziplinübergreifend relevanten Informationen über den Entwicklungsgegenstand während des gesamten Entwicklungsprozesses in einem zentralen Systemmodell abgelegt und sämtliche fachdisziplinspezifischen Entwicklungsobjekte über dieses synchronisiert und konsistent gehalten werden (vgl. Abschnitt 2.4.2.2). Bezüglich der Konsistenzsicherung wird die Anwendung des von GAUSEMEIER ET AL. in [GSG+09, S. 5ff.] vorgestellten Verfahrens zur Konsistenzsicherung fachdisziplinübergreifender und fachdisziplinspezifischer Modelle angenommen³⁵. In dem zentralen Systemmodell sollen die für die Umsetzung des

³⁴ Die vorgestellten Verfahren erheben keinen Anspruch darauf, alleinig zur Umsetzung eines selbstoptimierenden Entwicklungsprozesses geeignet zu sein. Der Autor möchte an dieser Stelle explizit feststellen, dass das hier vorgestellte Rahmenwerk bei den Werkzeugen zur Umsetzung des Selbstoptimierungsprozesses die Intention hat, offen für weitere Verfahren, Methoden und Werkzeuge zu sein.

³⁵ Wird das Verfahren nach GAUSEMEIER ET AL. nicht auf den Entwicklungsgegenstand angewendet, schließt dies die Umsetzung eines selbstoptimierenden Entwicklungsprozesses keinesfalls aus. Es wird aber ungleich aufwendiger, da Konsistenzsicherungs- und Informationsbeschaffungsmechanismen entsprechend den Anforderungen des selbstoptimierenden Entwicklungsprozess über die Gesamtheit der Entwicklungsobjekte neu definiert werden müssten.

selbstoptimierenden Entwicklungsprozesses notwendigen Informationen über den Entwicklungsgegenstand abgebildet werden. Die Modellierung des Systemmodells erfolgt mit Hilfe der in Kapitel 3.2.2.2 vorgestellten Modellierungstechnik CONSENS³⁶ (vgl. [GFD+08a, S. 91ff.] und [GDT+11]), wobei nicht alle Partialmodelle für die Abbildung der notwendigen Informationen gleich wichtig sind. Die beiden bedeutendsten sind die Partialmodelle Wirkstruktur und Funktionen. Beide werden für das in dieser Arbeit entwickelte *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* erweitert.

Die **Wirkstruktur** beschreibt dabei den grundlegenden Aufbau und die Wirkungsweise des zu entwickelnden Systems. Die einzelnen Systembestandteile werden durch Systemelemente abgebildet. Da die Wirkstruktur sowohl physische als auch nicht physische Systemelemente beinhalten kann, vereint sie unterschiedliche Strukturen in einer aspektübergreifenden, hierarchischen Struktur. Die wesentlichen in der Wirkstruktur subsumierten aspektspezifischen Strukturen sind die Funktionsstruktur, die Kommunikationstopologie, die Komponentenstruktur und die Baustruktur (vgl. [Ste07, S. 117])³⁷. Sie ergeben sich aus unterschiedlichen Verbindungen zwischen den Systemelementen sowie den Verweisen auf die durch die Systemelemente realisierten Funktionen und deren hierarchische Zusammenhänge. Die aspektspezifischen Strukturen können für eine Fachdisziplin (z.B. wird die Komponentenstruktur eins zu eins der Fachdisziplin Softwaretechnik zugeordnet) oder auch mehrere Fachdisziplinen (z.B. ist die Kommunikationstopologie u.a. für die Fachdisziplinen Elektronik, Regelungstechnik und Softwaretechnik relevant) bedeutsam sein. Bild 2-1 zeigt die an den Systemelementen für das in dieser Arbeit entwickelte Rahmenwerk zusätzliche zu annotierenden Informationen:

Die **Verbindungen** zwischen den einzelnen Systemelementen werden mit Hilfe der drei in Kapitel 2.2.2 beschriebenen Flussarten nach PAHL/BEITZ: *Informations-* *Stoff-* und *Energiefluss* abgebildet [PBF+07, S. 43]. Sie werden durch die *mechanische Verbindung* ergänzt. Mit ihr werden statische Verbindungen zwischen physischen Systemelementen und somit die Baustruktur des Systems abgebildet. Sämtliche Verbindungen werden über an den Systemelementen befindlichen Ports angelegt. Dies ermöglicht eine eindeutige Schnittstellendefinition.

Bestehen Systemelemente aus weiteren Subsystemelementen kann dies in Form einer **hierarchische Detaillierung** der Systemelemente abgebildet werden. Systemelemente, die nicht durch weitere Subsystemelemente detailliert werden, werden im Kontext dieser Arbeit als Elementarsystemelemente bezeichnet.

³⁶ Die Modellierung kann alternativ auch mit der Modellierungssprache SysML erfolgen. Hierfür ist ein dem Metamodell von CONSENS entsprechendes Profil in SysML zu erstellen.

³⁷ STEFFEN spricht von der funktionsorientierten und der gestaltorientierten Erzeugnisgliederung. Die funktionsorientierte Erzeugnisgliederung fasst dabei die Funktions- und die Komponentenstruktur zusammen.

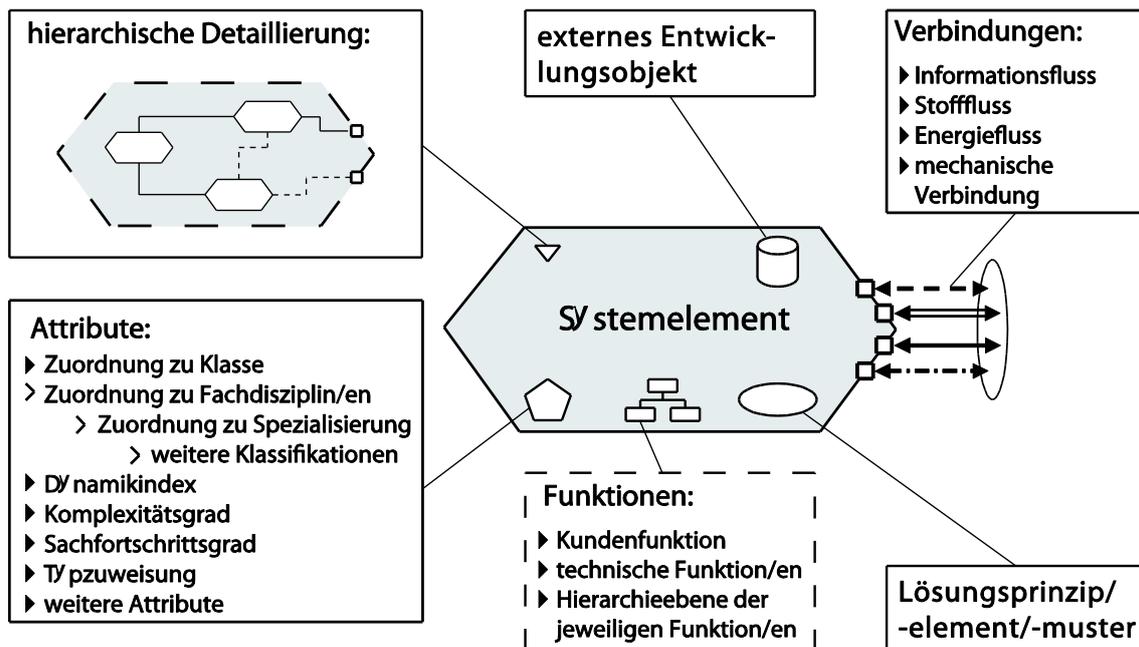


Bild 4-5 Systemelementbeschreibung im zentralen Systemmodell des Entwicklungsgegenstands

Besteht eine Beziehung zu einem oder mehreren **Entwicklungsobjekten**, die nicht in CONSENS abgebildet werden (z.B. Schaltungslayout, Dynamikmodell, Blockdiagramm etc.), wird dies durch eine entsprechende Verlinkung abgebildet. Solche Verlinkungen können auf allen Hierarchieebenen vorkommen.

Die Eigenschaften der Systemelemente werden durch deren **Attribute** beschrieben. Mit Hilfe hierarchischer Klassifikationsbäume wird durch die Entwickler definiert zu welcher *Klasse* von mechatronischem System das Systemelement gehört (dynamisches Mehrkörpersystem oder integriertes mechatronisches System), zu welcher/en *Fachdisziplin/en* (z.B. Maschinenbau, Elektrik, Elektronik, Regelungstechnik, Softwaretechnik) es gehört, ob es einer *Spezialisierung* innerhalb einer Fachdisziplin zugeordnet werden kann (z.B. Hydraulik oder Pneumatik als Spezialisierung des Maschinenbaus) usw. Der *Dynamikindex* gibt an wie stark ein Systemelement innerhalb des Gesamtsystems eingebunden ist bzw. dessen Verhalten beeinflusst. Er wird anhand der zu den Systemelementen der ersten Hierarchieebene bestehenden Verbindungen ermittelt. Eine detaillierte Beschreibung des Vorgehens zur Ermittlung des Dynamikindex ist Anhang A2.1 zu entnehmen. Der *Komplexitätsgrad* des Systems ist entweder durch die Entwickler abzuschätzen³⁸ oder alternativ aus der Struktur des betrachteten Systemelements zu ermit-

³⁸ Für die vorliegende Arbeit wird davon ausgegangen, dass für das Entwicklerkollektiv eines Unternehmens zumindest Einigkeit über eine Ordinalskala für den Komplexitätsgrad besteht, sodass dieser als Eingangsinformation für, im Unternehmen eingesetzte, Aufwandschätzverfahren dienen kann.

tel³⁹. Der *Fertigstellungsgrad* beschreibt, wie weit das Systemelement in der laufenden Entwicklung bereits ausgestaltet wurde. Die Verwendung des *Typkonzeptes* erlaubt es einzelne Systemelemente innerhalb des Systems mehrfach zu benutzen.

Greifen die Entwickler bei einem Systemelement auf ein bekanntes **Lösungsprinzip**, -**muster** oder konkretes **-element** zurück, wird dies ebenfalls annotiert. Auf Grund des bestehenden Erfahrungswissens über diese Elemente können einige Eigenschaften des Systemelements wie bspw. die Zuordnung zu einer oder mehreren Fachdisziplinen automatisiert erfolgen. Zu diesem Zweck ist das Lösungswissen einheitlich zu beschreiben, z.B. mit Hilfe der Lösungsmusterspezifikation nach DUMITRESCU [Dum11, S. 128]. Lösungselemente werden im Gegensatz zu Systemelementen nicht weiter detailliert. Ihr Sachfortschrittsgrad beträgt von Anfang an 100%.

Die Funktionalität des zu entwickelnden Systems wird im Partialmodell **Funktionen** in Form einer Funktionshierarchie beschrieben. Die Hauptfunktion des Systems wird hierarchisch in Teilfunktionen heruntergebrochen. Dies geschieht solange, bis ein oder mehrere Systemelemente zur Realisierung der Teilfunktion/en gefunden werden können. Jedem Systemelement sind folglich ein oder mehrere Funktionen zugeordnet, die von ihm realisiert werden. Die Funktionen werden in Kunden- und technische Funktionen unterschieden. Kundenfunktionen sind solche, die vom Kunden im späteren Betrieb bewusst wahrgenommen werden, wie z.B. das automatische Öffnen der Heckklappe bei einem Pkw. Technische Funktionen hingegen werden vom Kunden nicht bewusst wahrgenommen und dienen meistens der Realisierung von Kundenfunktionen. Ein Beispiel ist das Detektieren des Benutzerwunsches als Voraussetzung für das Öffnen der Heckklappe.

4.3.1.2 Modellierung der Prozessbausteine

Die zweite Informationsquelle, auf die sich die Verfahren zur Umsetzung des selbstoptimierenden Entwicklungsprozesses stützen, sind in Anlehnung an verschiedene Ansätze aus dem Stand der Technik ([Bic00, S. 78ff.], [Red06, S. 108ff.], [Lev06, S. 188] und [Mur99]) Prozessbausteine, aus denen der Entwicklungsprozess flexibel situationsspezifisch zusammengesetzt werden kann. Entsprechend der Anforderung aus Kapitel 2.7, für die Entwickler aufgabenspezifische Handlungsanweisungen adäquat bereit zu stellen, werden in den Prozessbausteinen fachdisziplinspezifische und fachdisziplinüber-

³⁹ In diesem Fall müssen für die Vergleichbarkeit der Komplexität verschiedener Systeme nach NEGELE zumindest beide Systeme in der gleichen Modellierungssprache abgebildet werden und eine Konvention über den Auflösungs- bzw. Detaillierungsgrad der Modelle bestehen [Neg06, S. 9]. Beide Forderungen werden im Kontext des hier beschriebenen Verfahrens insofern erfüllt, dass für das Systemmodell als einheitliche Modellierungstechnik CONSENS verwendet und das Modell soweit detailliert wird, bis alle fachdisziplinübergreifend relevanten Informationen darin abgebildet sind. Die formale Formulierung des Zusammenhangs zwischen Systemelementstruktur und Komplexitätsgrad wäre also grundsätzlich möglich. Eine allgemeingültige formale Definition des Komplexitätsgrads ist jedoch nicht Gegenstand dieser Arbeit.

greifende Entwicklungstätigkeiten auf den Detaillierungsstufen „Konstruktionsoperationen“ und „Grundoperationen“ nach HUPKA [Hup76, S. 9] modelliert⁴⁰. Abhängig von den verwendeten Hilfsmitteln können die in den Prozessbausteinen beschriebenen Entwicklungstätigkeiten auf unterschiedliche Art und Weise ausgeführt werden. Die Prozessbausteine werden entsprechend Bild 4-6 in zwei Arten unterteilt.

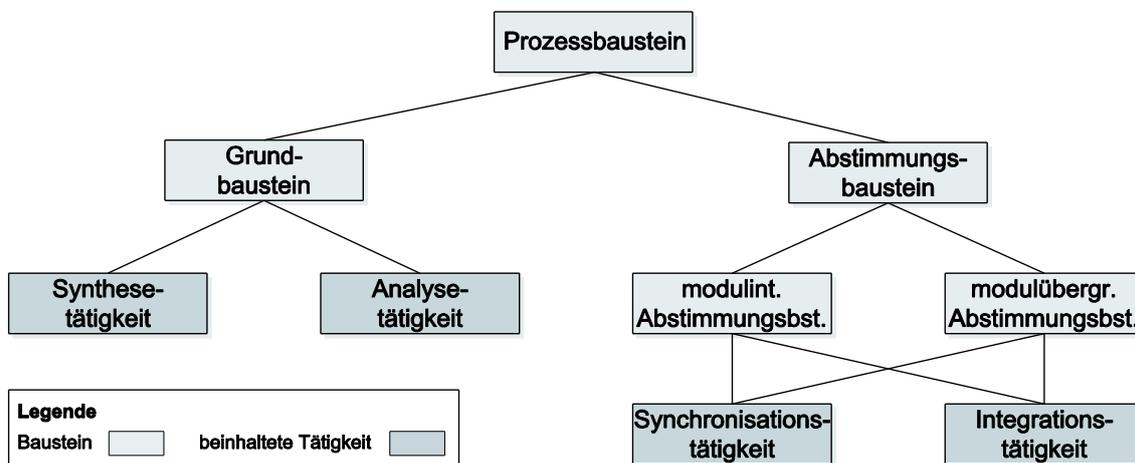


Bild 4-6 Arten von Prozessbausteinen

Die **Grundbausteine** bilden die Analyse- und Synthesetätigkeiten zur Entwicklung einer technischen Lösung ab. Sie bewirken maßgeblich die Wertschöpfung im Entwicklungsprozess und treiben den Reifegrad des Entwicklungsgegenstands maßgeblich voran. Da für die Systembestandteile des Entwicklungsgegenstands wiederkehrend ähnlich geartete Problemstellungen zu bearbeiten sind, kommen einzelne Grundbausteine im Entwicklungsprozess mehrfach vor.

Die auszuführenden Entwicklungstätigkeiten und ihre Ergebnisse müssen synchronisiert und entsprechend der Struktur des Entwicklungsgegenstands integriert werden. Die hierfür notwendigen Synchronisations- und Integrationstätigkeiten werden in den **Abstimmungsbausteinen** abgebildet (siehe auch Kapitel 2.4.2.2). Es werden modulinterne und modulübergreifende Abstimmungsbausteine unterschieden.

Beide Arten von Prozessbausteinen werden mit Hilfe der in Kapitel 3.2.1.2 bereits dargestellten Modellierungssprache OMEGA abgebildet. Um den Anforderungen des hier beschriebenen Rahmenwerks gerecht zu werden, wurde OMEGA in Teilen erweitert. Bild 4-7 zeigt die zur einheitlichen Beschreibung der verschiedenen Prozessbausteine in OMEGA abgebildeten Informationen. Das Datenmodell zur rechnerinternen Repräsentation der Prozessbausteine ist in Anhang A1 dargestellt. Nachfolgend werden die enthaltenen Informationen erläutert.

⁴⁰ Basierend auf den in Kapitel 2.4.2.2 beschriebenen fachdisziplinspezifischen Vorgehensweisen wurden zur Validierung des hier beschriebenen Verfahrens für die Fachdisziplinen Maschinenbau, Elektrik, Elektronik, Regelungstechnik und Softwaretechnik entsprechende Kataloge von Prozessbausteinen entwickelt (siehe Anhang A3).

Die **Eingangs- und Ausgangsobjekte** beschreiben die Entwicklungsobjekte und legen fest auf welche Art und Weise die Prozessbausteine zu unterschiedlichen Prozessnetzen verknüpft werden können. Jeder Prozessbaustein kann mehrere Eingangs- und Ausgangsobjekte besitzen, die entweder in Kombination oder alternativ zueinander vorliegen müssen bzw. durch die im Prozessbaustein ausgeführte Aktivität generiert werden.

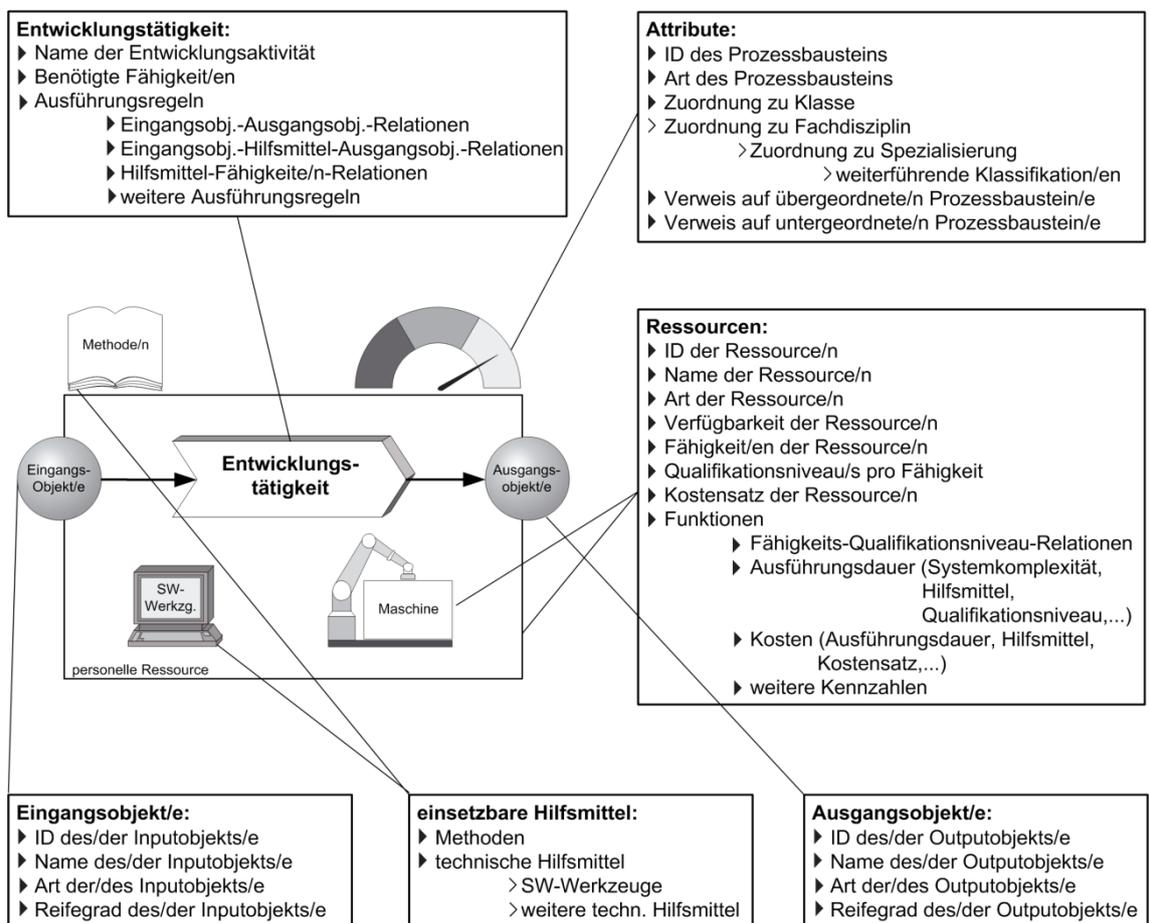


Bild 4-7 Modellierung eines Prozessbausteins

Die **Attribute** beschreiben die Eigenschaften eines Prozessbausteins. Zur Klassifikation (*Fachdisziplin, Spezialisierung* etc.) der Prozessbausteine werden die gleichen Klassifikationsbäume wie zur Klassifikation der Systemelemente verwendet. Ferner wird die *Art der Prozessbausteine* (Grundbaustein, modulinterner Abstimmungsbaustein oder modulübergreifender Abstimmungsbaustein) definiert. Wird eine Entwicklungstätigkeit durch eine oder mehrere weitere Entwicklungstätigkeiten detailliert, wird dies durch *Verweise* zwischen den Prozessbausteinen dieser Entwicklungstätigkeiten abgebildet. Die Verweise sind gerichtet und zeigen jeweils auf unter- oder übergeordnete Prozessbausteine. Detaillierungen ergeben sich z.B. dann wenn den einzelnen Ebenen der zur Eigenschaftsmodellierung verwendeten Klassifikationsbäume individuell Prozessbausteine zugeordnet werden und nicht nur den Wurzeln der Klassifikationsbäume.

Die zur Ausführung der Entwicklungstätigkeit **einsetzbaren Hilfsmittel** werden primär in *Methoden* und *SW-Werkzeuge* unterschieden. Diesen beiden Kategorien können bei Bedarf weitere hinzugefügt werden.

Abhängig vom Eingangsobjekt und dem eingesetzten Hilfsmittel ergeben sich unterschiedliche Möglichkeiten zur Ausführung der durch den Prozessbaustein repräsentierten **Entwicklungstätigkeit**. Jede dieser Ausführungsformen erfordert eine oder mehrere spezifische *Fähigkeiten* von den die Entwicklungstätigkeit ausführenden Ressourcen (z.B. personelle Ressource, Maschine etc.). Ein Beispiel für eine spezifische Fähigkeit ist ein CAD-Modell mit dem SW-Werkzeug Catia V5 erstellen zu können. Welche Eingangsobjekte mit welchen Hilfsmitteln in welche Ausgangsobjekte überführt werden können, legen die *Ausführungsregeln* fest. Sie beschreiben die basale Logik aus der sich die verschiedenen Möglichkeiten zur Ausführung der Entwicklungstätigkeit ergeben. Weiterhin wird durch die Prozessregeln festgelegt welche Fähigkeiten zur Ausführung der Entwicklungstätigkeit mit einem bestimmten Hilfsmittel benötigt werden. Grundsätzlich muss immer mindestens eine Möglichkeit modelliert sein, wie eine Entwicklungstätigkeit ausgeführt werden kann. Eine konkrete Ausprägungskombination von Eingangsobjekt, Entwicklungstätigkeit, Hilfsmittel, benötigte Fähigkeit und Ausgangsobjekt wird als Ausführungsmodus des Prozessbausteins bezeichnet.

Für jeden Ausführungsmodus eines Prozessbausteins werden eine oder mehrere **Ressourcen** benötigt. Ressourcen werden in personelle Ressourcen und Maschinen unterteilt. Jede Ressource hat eine spezifische *Verfügbarkeit*, verschiedene *Fähigkeiten* mit jeweiligem *Qualifikationsniveau* und einen *Kostensatz*. Durch Vergleich der von den Ressourcen beherrschten Fähigkeiten mit denen, die in einem Ausführungsmodus benötigt werden, können den Ausführungsmodi die verschiedenen Ressourcen zugeordnet werden, die diese ausführen können. Mit Hilfe der den Ressourcen zugeordneten *Funktionen* werden verschiedene Prozesskennzahlen, wie z.B. die *Dauer* der Ausführung eines Ausführungsmodus, die dabei entstehenden *Kosten* und weitere Kennzahlen ermittelt. Die Funktionen spiegeln die für die Entwicklungsprozessplanung einsetzbaren Aufwandschätzverfahren entsprechend Kapitel 2.5.2 (algorithmische Verfahren, Vergleichsverfahren oder Kennzahlenverfahren) wider⁴¹. Die Zuordnung zu den Ressourcen ermöglicht die spezifische Aufwandsermittlung für einen Ausführungsmodus in Abhängigkeit der eingesetzten Ressource. Für die Aufwandsermittlung über den Entwicklungsgegenstand benötigte Informationen können den Eigenschaften der Systemelemente des vorstehend beschriebenen Systemmodells entnommen werden.

⁴¹ Für die vorliegende Arbeit wird davon ausgegangen, dass in den Unternehmen entsprechende funktionale Zusammenhänge zwischen den für die Systemelemente hinterlegten Attributen und den zu ermittelten Prozesskennzahlen bekannt sind. Die Entwicklung spezifischer Aufwandschätzverfahren ist nicht Gegenstand dieser Arbeit.

4.3.2 Verfahren zur Umsetzung

Im Folgenden werden drei auf den im vorangegangenen Kapitel 4.3.1 vorgestellten Modellierungstechniken fußende Verfahren zur Umsetzung des Selbstoptimierungsprozesses nach Kapitel 4.2.2 vorgestellt. Im Einzelnen sind dies ein Verfahren zur systemmodellbasierten Prozessplanung (Kapitel 4.3.2.1), zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen (Kapitel 4.3.2.2) sowie zur zielkonformen Planauswahl (Kapitel 4.3.2.3).

4.3.2.1 Verfahren zur systemmodellbasierten Prozessplanung

Wesentliche Grundlage für die Anpassung des Entwicklungsprozesses im Sinne der Selbstoptimierung ist die Kenntnis über mögliche Abläufe wie der aktuell vorliegende Entwicklungsgegenstand in die vollständige Beschreibung eines technischen Systems, das sämtliche Produktziele erfüllt, überführt werden kann. Ziel des nachfolgend dargestellten Verfahrens sind daher automatisch generierte, entwicklungsgegenstandskonforme und paretooptimale Prozesspläne, die eine Vorgangsfolgen-, Termin-, Kosten-Hilfsmittel- und Ressourcenplanung beinhalten.

Das Verfahren ist wissensbasiert und nutzt die mit Hilfe der in Kapitel 4.3.1 vorgestellten Modellierungstechniken spezifizierten Inhalte für die Generierung möglicher Vorgangsfolgen (vgl. Bild 4-8). Für die Termin-, Kosten-, Hilfsmittel- und Ressourcenplanung wurde das von KLÖPPER entwickelte Lösungsverfahren zur Generierung paretooptimaler Vorgangsfolgen adaptiert [Klö10, S. 635ff.], [Klö11, S. 1ff.].

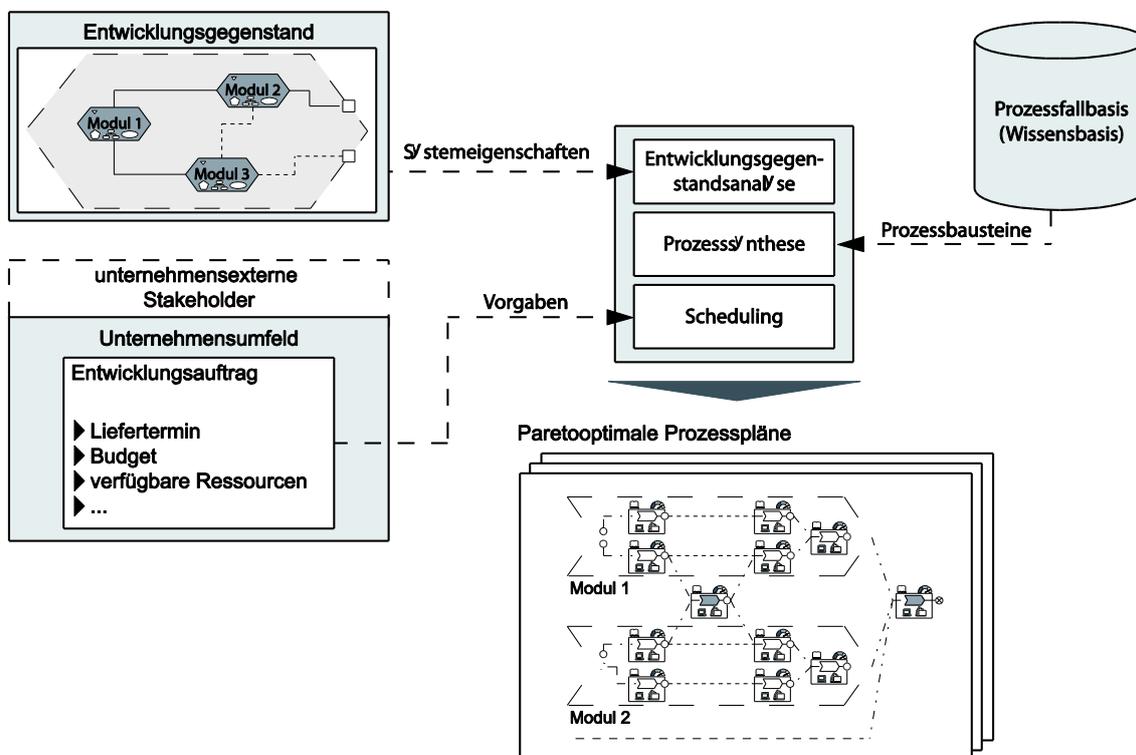


Bild 4-8 Bestandteile des Verfahrens zur systemmodellbasierten Prozessplanung

Bild 4-9 zeigt den Ablauf des Verfahrens zur systemmodellbasierten Prozessplanung in Form eines Phasen-Meilenstein-Diagramms. Insgesamt werden drei Phasen durchlaufen, an deren Ende Prozesspläne stehen, mit denen der aktuell vorliegende Entwicklungsgegenstand in die vollständige Beschreibung eines technischen Systems, das sämtliche Produktziele erfüllt, überführt werden kann. Die drei Phasen müssen nicht immer vollständig oder in der angegebenen Reihenfolge durchlaufen werden. Fällt bspw. eine Ressource aus, würden lediglich die Tätigkeiten der Phase „Scheduling“ erneut durchlaufen.

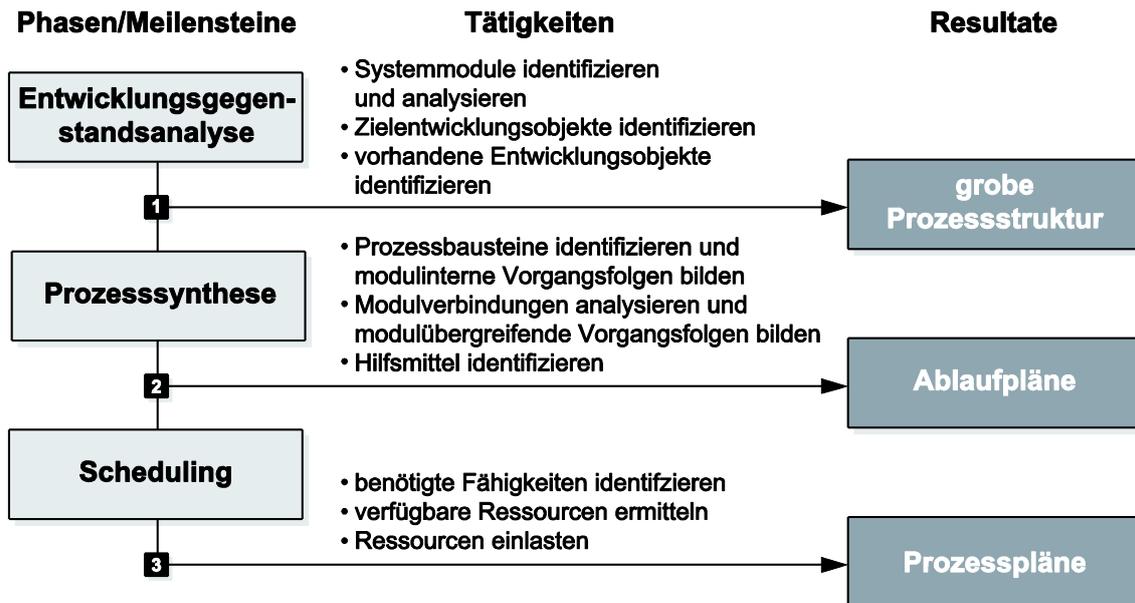


Bild 4-9 Ablauf des Verfahrens zur systemmodellbasierten Prozessplanung

Nachfolgend werden die drei Phasen sowie die in ihnen beinhalteten Tätigkeiten und Ergebnisse detailliert dargestellt.

Phase 1: Entwicklungsgegenstandsanalyse

Die erste Phase dient der Entwicklung einer groben objektorientierten Prozessstruktur. Zuerst werden die im Entwicklungsprozess parallel zueinander zu entwickelnden Module des Systems anhand des Systemmodells identifiziert. Die Module werden durch die Systemelemente auf der ersten Hierarchieebene der Wirkstruktur repräsentiert (vgl. Bild 4-10). Für jedes Modul wird ein separater Entwicklungsprozessstrang vorgesehen. Werden Module innerhalb des Entwicklungsgegenstands mehrfach verwendet, sind sie also vom gleichen Typ, wird für diese nur ein Entwicklungsstrang vorgesehen.

Um festzulegen, welche Entwicklungsobjekte am Ende eines Entwicklungsprozessstranges vorliegen müssen, wird anhand der den Systemelementen zugeordneten Klassifikationsbäume bestimmt, zu welcher Klasse von mechatronischem System (dynamisches Mehrkörpersystem oder integriertes mechatronisches System) und zu welcher/en Fachdisziplin/en (Maschinenbau, Elektrik/Elektronik, Regelungstechnik oder Software-

technik) sie gehören. Liegen Spezialisierungen innerhalb einer Fachdisziplin (z.B. Hydraulik, Pneumatik etc.) oder noch weiterführende Klassifikationen vor, werden diese Informationen ebenfalls verwendet. Für das hier beschriebene Verfahren muss mindestens die Zuordnung der Systemelemente zu einer Klasse von mechatronischen Systemen und einer oder mehreren Fachdisziplinen vorliegen.

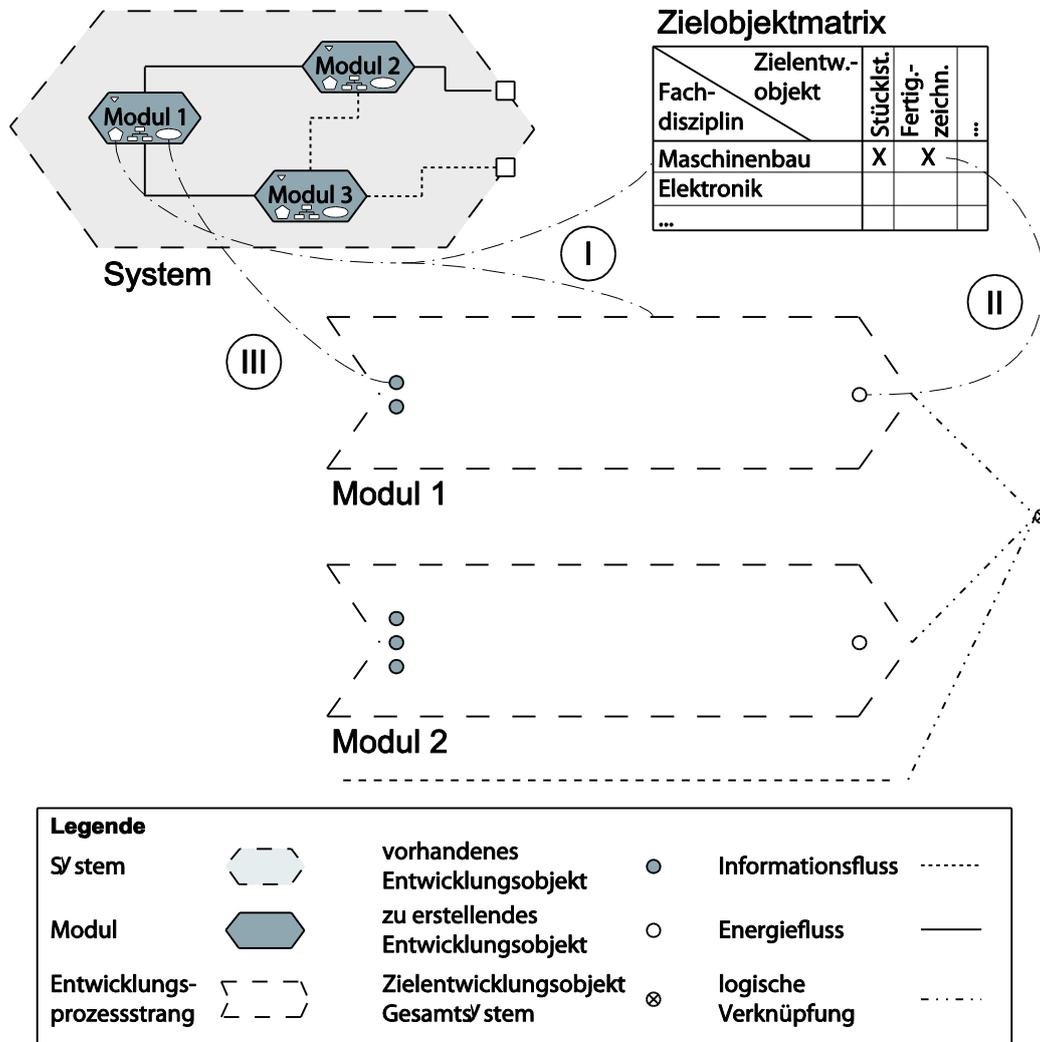


Bild 4-10 Phase 1 des Verfahrens zur systemmodellbasierten Prozessplanung

Anhand der vorliegenden Zuordnungen werden über eine **Zielobjektmatrix** diejenigen Entwicklungsobjekte identifiziert, die am Ende eines spezifischen Entwicklungsprozessstrangs vorliegen müssen. Diese Objekte werden als Zielentwicklungsobjekte bezeichnet. Bspw. müssen für ein dem Maschinenbau zugeordnetes Modul am Ende die Fertigungszeichnungen und die Modulstückliste vorliegen. Das immer gesetzte modulübergreifende Zielentwicklungsobjekt ist die vollständige Produktdokumentation auf Gesamtsystemebene. Sie sind die Senke des Entwicklungsprozesses.

Am Ende dieser Phase werden über die Verlinkungen an den Systemelementen die im Entwicklungsprozess bereits vorliegenden Entwicklungsobjekte erfasst und ebenfalls

den Entwicklungsprozesssträngen der jeweiligen Module zugeordnet. Im Ergebnis liegt ein grobes Prozessgerüst vor das die grundsätzlich vorkommenden Entwicklungsprozessstränge und deren Ein- und Ausgangsobjekte beinhaltet.

Phase 2: Prozesssynthese

In der Prozesssynthese werden mit Hilfe der Prozessbausteine mögliche Vorgangsfolgen gebildet, mit denen die vorliegenden Entwicklungsobjekte in die Zielentwicklungsobjekte überführt werden können. Ähnlich wie bei der Identifikation der Zielentwicklungsobjekte in Phase 1 werden die für die Entwicklung eines Moduls in Frage kommenden Prozessbausteine anhand eines Abgleich der Klassifikationsbäume der Systemelemente mit den Prozessbausteinattributen identifiziert. Die in Frage kommenden Prozessbausteine werden den Entwicklungssträngen der Module zugeordnet. Sie können sowohl Grund- als auch modulinterne Abstimmungsbausteine⁴² sein.

Anschließend werden die Prozessbausteine für jedes Modul entsprechend ihrer möglichen Eingangsobjekt-Ausgangsobjekt-Kombinationen in einer rückwärts gerichteten Breitensuche ausgehend vom Zielentwicklungsobjekt des Modulentwicklungsstrangs aneinandergereiht (vgl. Bild 4-11). Welche Kombinationen von Eingangs- und Ausgangsobjekten pro Prozessbaustein möglich sind, wird durch dessen Ausführungsregeln (Eingangsobjekt-Ausgangsobjekt-Relationen) festgelegt. Die Suche endet, wenn das Eingangsobjekt für den zuletzt hinzugefügten Prozessbaustein einem der für das Modul bereits vorliegenden Entwicklungsobjekte entspricht oder wenn es keinen Grund- oder Abstimmungsbaustein mehr gibt, der als Ausgangsobjekt eines der noch fehlenden Eingangsobjekte hat. Die so gefundenen alternativen Vorgangsfolgen ermöglichen die separate Entwicklung jedes Moduls des Entwicklungsgegenstands.

Um die verschiedenen Modulentwicklungsstränge miteinander zu verbinden, werden anhand der im Systemmodell zwischen den Modulen abgebildeten Verbindungen modulübergreifende Abstimmungsbausteine eingesetzt. Welche modulübergreifenden Abstimmungsbausteine dies sind, wird durch **Prozessregeln** festgelegt. Eine Prozessregel legt abhängig von der Klassifikation zweier oder mehrerer Module, den zwischen ihnen in der Wirkstruktur bestehenden Verbindungen und dem hierarchischen Zusammenhang zwischen den durch die Module realisierten Funktionen die einzusetzenden modulübergreifenden Abstimmungsbausteine fest. Besteht bspw. zwischen zwei dem Maschinenbau zugeordneten Modulen eine mechanische oder stoffliche Verbindung werden die Gestalt der Module integrierende Abstimmungsbausteine eingesetzt. Auf diese Weise ist es möglich, den verschiedenen in der Wirkstruktur zusammengefassten Strukturen (Funktionsstruktur, Kommunikationstopologie, Baustruktur, Komponentenstruktur etc.) bei der Planung des Entwicklungsprozesses gerecht zu werden.

⁴² Voraussetzung hierfür ist, dass das Modul mindestens zwei verschiedenen Fachdisziplinen zugeordnet wurde

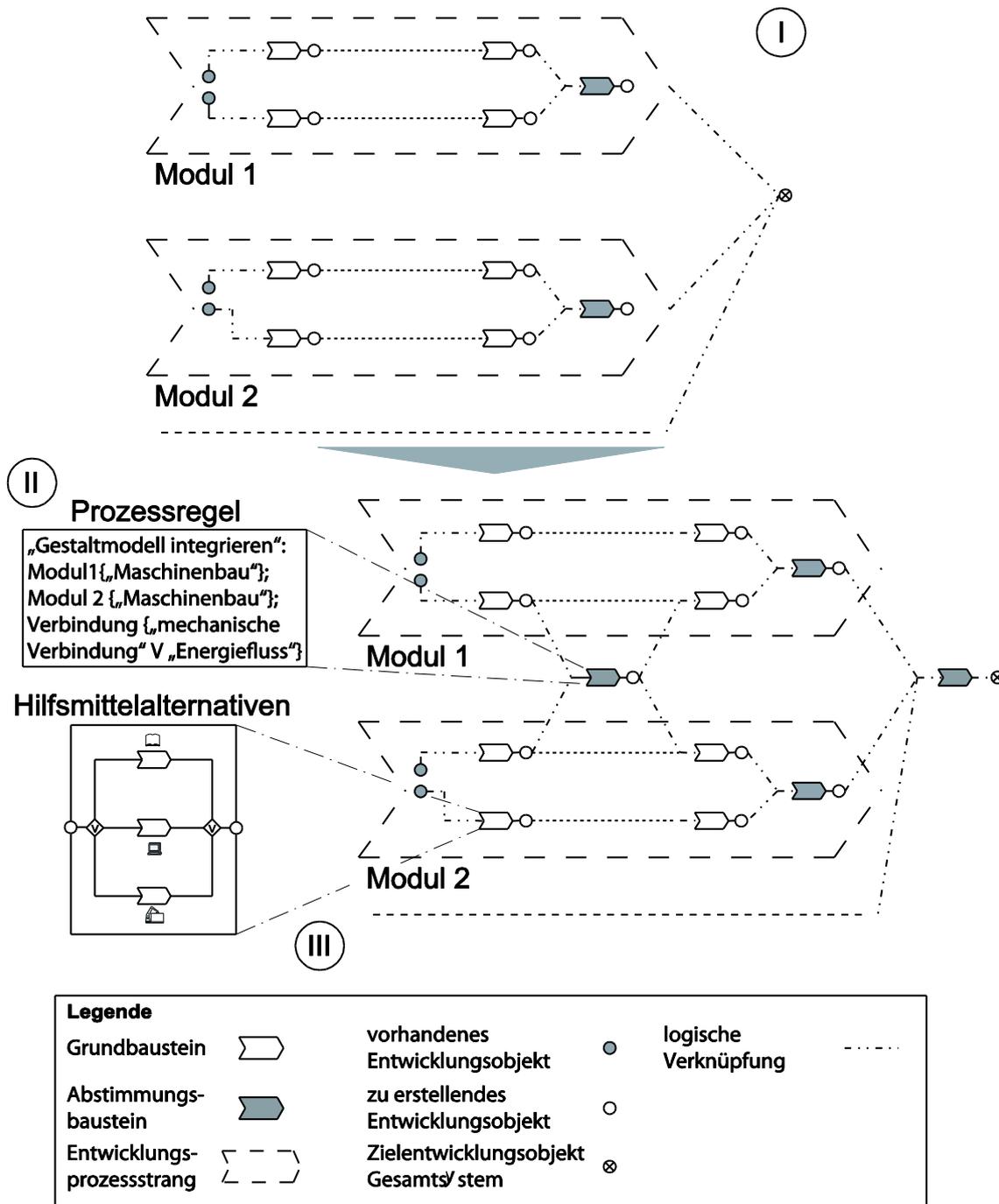


Bild 4-11 Phase 2 des Verfahrens zur systemmodellbasierten Prozessplanung

Unter die modulübergreifenden Abstimmungsbausteine fallen auch die im Entwicklungsprozess auszuführenden, mehrere Module umfassenden Validierungsaktivitäten. Die fachdisziplinspezifischen unter ihnen folgen aus den in der Wirkstruktur beinhaltenen aspektspezifischen Strukturen (Baustuktur, Komponentenstruktur etc.). Sie werden durch die fachdisziplinübergreifenden Validierungsaktivitäten ergänzt, deren Reihenfolge sich aus den hierarchischen Zusammenhängen zwischen den durch die Module realisierten Funktionen ergibt. Die Vorgehensweisen zur funktionsorientierten Reihen-

folgebestimmung der fachdisziplinübergreifenden Validierungsaktivitäten ist Anhang A2.2 zu entnehmen.

Ein immer einzusetzender modulübergreifender Abstimmungsbaustein ist der, der die Entwicklungsergebnisse der einzelnen Modulentwicklungsstränge zur vollständigen Produktdokumentation auf Gesamtsystemebene zusammenführt. Eine Auflistung der Entwicklungstätigkeiten, die im Rahmen dieser Arbeit beispielhaft als Prozessbausteine für die Fachdisziplinen Maschinenbau, Elektrik/Elektronik, Regelungstechnik und Softwaretechnik modelliert wurden, ist im Anhang A3 einzusehen.

Die modulübergreifenden Abstimmungsbausteine werden über ihre Ein- und Ausgangsobjekte mit den Prozessbausteinen in den Modulentwicklungssträngen verbunden. Kann ein Ausgangsobjekt eines modulübergreifenden Prozessbausteins nicht direkt mit einem Eingangsobjekt innerhalb der betrachteten Modulprozessstränge verbunden werden, werden in einer vorwärtsgerichteten Breitensuche unter Verwendung weiterer modulübergreifender Abstimmungsbausteine mögliche Vorgangsfolgen zu den im Modulprozessstrang enthaltenen Entwicklungsobjekten gesucht. Das Vorgehen gleicht dem zur Generierung der Vorgangsfolgen für die Modulprozessstränge.

Im nächsten Schritt werden die möglichen Hilfsmittel zur Ausführung der Prozessbausteine anhand der Ausführungsregeln der Prozessbausteine (Eingangsobjekt-Hilfsmittel-Ausgangsobjekt-Relationen) ermittelt. Kann eine Eingangsobjekt-Ausgangsobjekt-Kombination mit mehreren Hilfsmitteln realisiert werden, wird der betreffende Prozess in alternativ ausführbare Subprozesse unterteilt.

Abschließend werden die nicht verwendeten bereits im Entwicklungsprozess vorliegenden Entwicklungsobjekte gelöscht. Als Ergebnis liegen mehrere mögliche, alternative Vorgangsfolgen vor, mit denen die vollständige Produktdokumentation des Entwicklungsgegenstands erstellt werden kann. Sie bilden die logische Folge der auszuführenden Entwicklungstätigkeiten ab, die in der nachfolgenden Phase auf die zur Verfügung stehenden Ressourcen verteilt werden.

Phase 3: Scheduling

Für die Termin-, Kosten-, Hilfsmittel- und Ressourcenplanung werden die Vorgangsfolgen in Hierachical Precedence Graphs (HPG) nach KLÖPPER [Klö10, S. 635ff.], [Klö11, S. 1ff.] überführt und anhand der Ausführungsregeln der Prozessbausteine die von den einzusetzenden Ressourcen benötigten Fähigkeiten ermittelt. Anschließend werden die Ressourcen identifiziert, die die benötigten Fähigkeiten besitzen. Die verfügbaren Ressourcen werden mit Hilfe des von KLÖPPER auf Basis einer heuristischen Mehrzielsuche (vgl. [DCD99, S. 2ff.]) entwickelten Lösungsverfahrens zur Generierung pareto-optimaler Vorgangsfolgen eingelastet (vgl. Bild 4-12) [Klö10, S. 635ff.], [Klö11, S. 1ff.].

Hierachical Precedence Graph

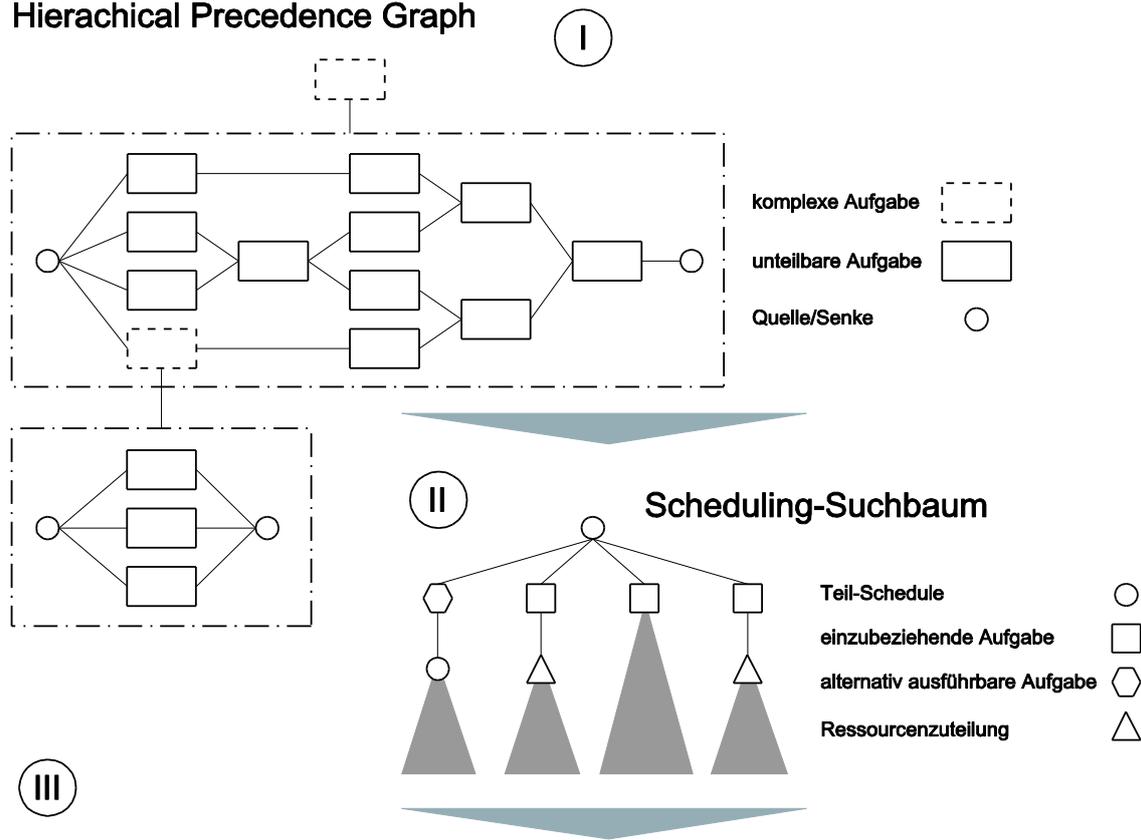


Bild 4-12 Phase 3 des Verfahrens zur systemmodellorientierten Prozessplanung

Bei der Einlastung einer Ressource für eine Entwicklungstätigkeit werden anhand der den Ressourcen zugeordneten Funktionen und der Moduleigenschaften die für das Scheduling benötigten Kennzahlen ermittelt. Dies sind z.B. die Dauer der Aktivität oder die dabei entstehenden Kosten. Es wird jeweils diejenige Ressource für eine Aufgabe eingelastet, die die besten Ergebnisse für die Kennzahlen liefert. Hierbei gibt es selten

eine Ressource, deren Leistungsfähigkeit die der anderen bei allen Kennzahlen übertrifft. Bspw. liefert ein sehr erfahrener Entwickler in der Regel ein Ergebnis höherer Prozessgüte, ist dafür aber bei gleicher Ausführungsdauer teurer als ein unerfahrener Entwickler. Liegt ein solcher Zwiespalt vor wird für jede Option ein eigener Plan erzeugt. Der eine Plan ist dann z.B. prozessgüten-, der andere kostenoptimal. Bei der nächsten einzuplanenden Entwicklungstätigkeit wird äquivalent verfahren und die Kennzahlen zusammengefasst. Jeder neu erstellte Plan wird mit den bereits vorliegenden verglichen. Werden bei diesem Vergleich Pläne entdeckt, bei denen eine oder mehrere Kennzahlen eines Plans besser als die des anderen und alle übrigen gleich sind (der bessere Plan dominiert den schlechteren), wird der schlechtere Plan verworfen. Im Ergebnis liegen mehrere pareto-optimale Pläne vor, d.h. mindestens eine Kennzahl ist besser als bei allen anderen und die übrigen so gut wie für diesen Prozessverlauf möglich [Klö10, S. 635ff.], [Klö11, S. 1ff.].

Das Verfahren wird auf die verschiedenen in Phase 2 generierten möglichen Vorgangsfolgen angewendet. Die dabei ermittelten Pläne werden zusammengetragen und aus dieser Menge die dominierenden extrahiert. Im Ergebnis liegen die pareto-optimale Prozesspläne über alle möglichen Vorgangsfolgen zur Erstellung der vollständigen Produktdokumentation des Entwicklungsgegenstands vor. Diese werden innerhalb der Regeleinrichtung des selbstoptimierenden Entwicklungsprojektmanagements an die Ziel- und Ausführungsbestimmung übergeben, die aus dieser Menge den in der aktuellen Entwicklungssituation auszuführenden Plan auswählt.

4.3.2.2 Verfahren zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen

Das in vorangegangenem Kapitel 4.3.2.1 beschriebene Verfahren zur systemmodellbasierten Prozessplanung liefert eine Menge pareto-optimale Prozesspläne, mit denen der aktuelle Entwicklungsgegenstand in eine vollständige Beschreibung des zu entwickelnden Systems überführt werden kann. Um den für die aktuelle Entwicklungssituation angemessenen Prozessplan aus der Menge der Möglichkeiten auszuwählen, bedarf es einer Entscheidungsgrundlage anhand derer die möglichen Prozesspläne bewertet und der am besten geeignete Plan ausgewählt werden kann. Diese Entscheidungsgrundlage ist das interne Zielsystem des Entwicklungsprozesses.

Mit Hilfe des nachfolgen dargestellten Verfahrens zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen werden die internen Ziele des Entwicklungsprozesses festgelegt. Dies geschieht unter Berücksichtigung der Entwicklungssituation, die durch die unternehmensexternen Stakeholder, das Unternehmensumfeld, den Entwicklungsgegenstand sowie den bisherigen Verlauf des Entwicklungsprozesses vorgegeben wird. Bild 4-13 zeigt schematisch die Bestandteile des Verfahrens und ihr Zusammenspiel. Eine alle Aufgaben, Informationsflüsse und eingesetzte Me-

thoden beinhaltende detaillierte Darstellung des Verfahrens ist dem Anhang A4 zu entnehmen.

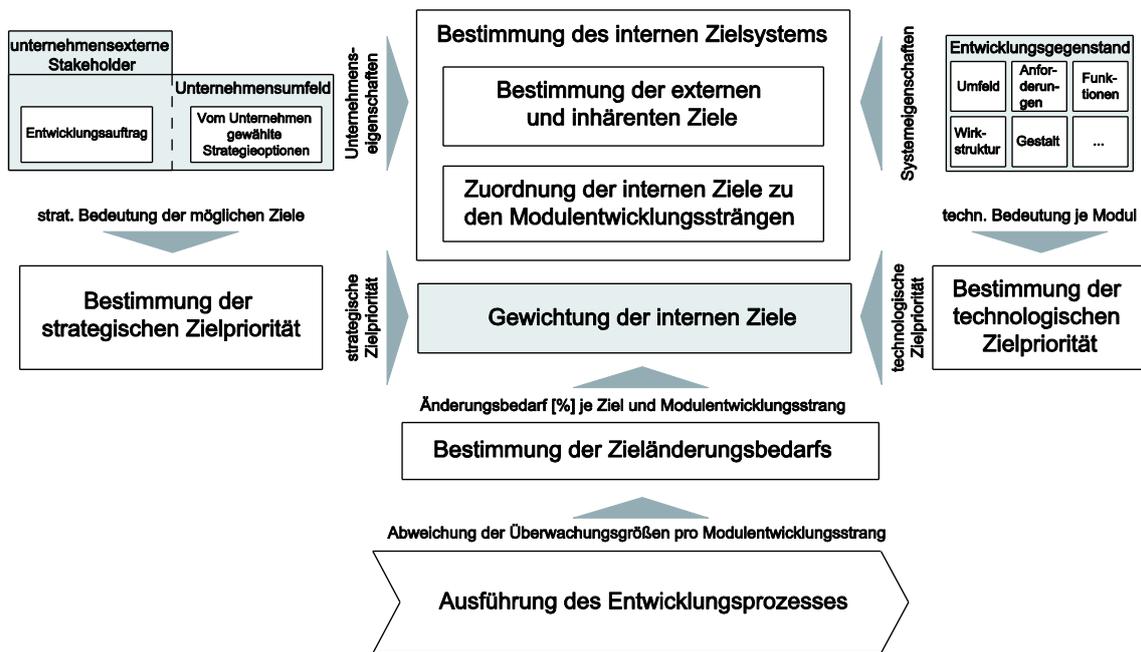


Bild 4-13 Bestandteile des Verfahrens zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen.

Der Ablauf des Verfahrens gliedert sich gemäß Bild 2-1 in fünf Phasen an deren Ende das situationsspezifisch ausgeprägte interne Zielsystem des Entwicklungsprozesses vorliegt. Je nach dem durch welchen Einfluss sich die Entwicklungssituation geändert hat werden alle oder nur vereinzelte Phasen des Verfahrens durchlaufen. Die fünf Phasen werden nachfolgend detailliert erläutert.

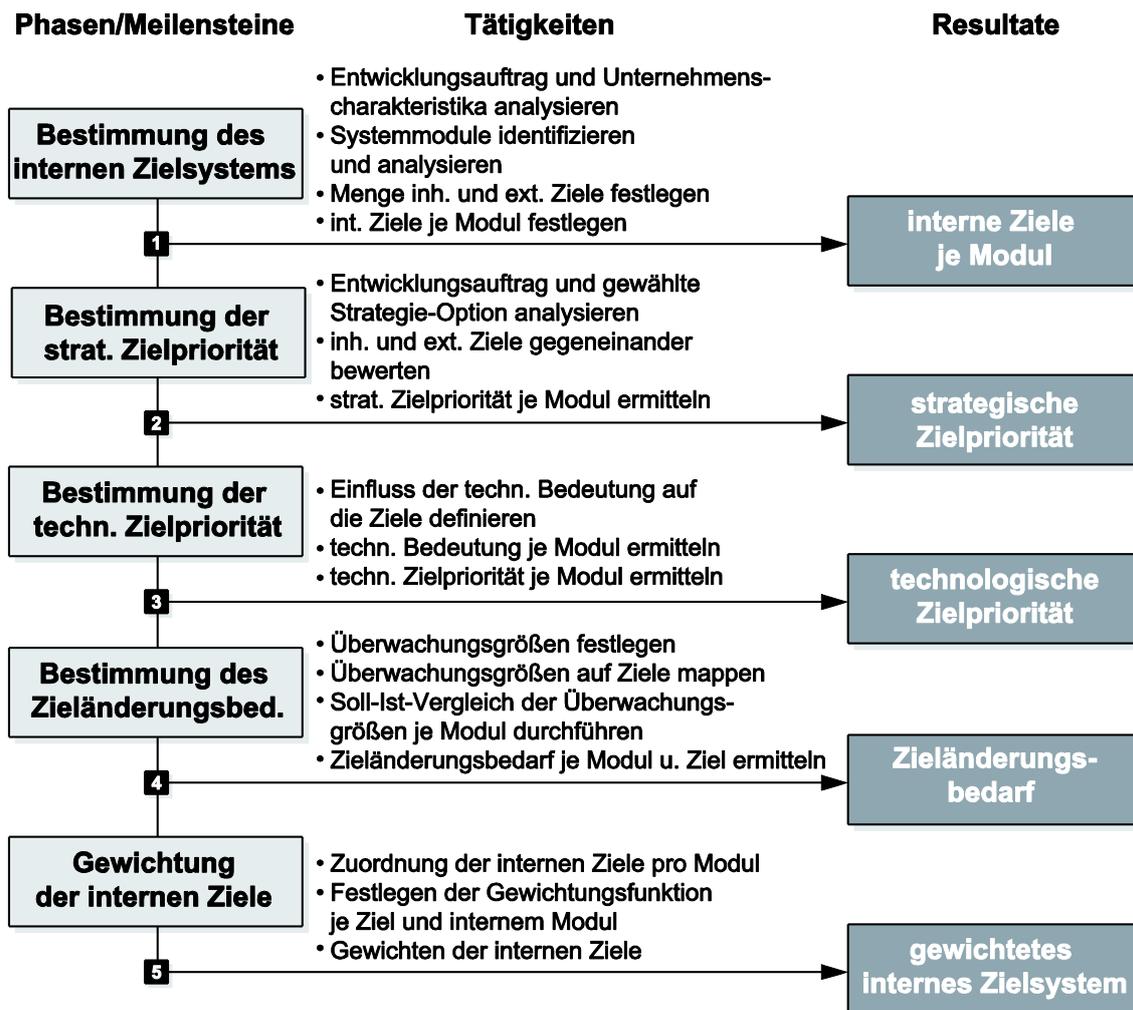


Bild 4-14 Ablauf des Verfahrens zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen

Phase 1: Bestimmung des internen Zielsystems

In der ersten Phase wird durch die Analyse der Eigenschaften des gegenwärtig vorliegenden Entwicklungsgegenstands und der Eigenschaften des Unternehmensumfelds, in dem er entwickelt werden soll, sowie der Zuhilfenahme allgemeinen Entwicklungs-Know-Hows die Grundmenge an externen und inhärenten Zielen des Entwicklungsprozesses bestimmt (vgl. Bild 4-15). Diese Grundmenge stellt die potentiellen internen Ziele des Entwicklungsprozesses dar.

Die externen Ziele ergeben sich aus dem vorliegenden Entwicklungsauftrag, der durch das Unternehmen gewählten Strategieoption und den Eigenschaften des gegenwärtig vorliegenden Entwicklungsgegenstands. Handelt es sich bei der Entwicklung um einen Kundenauftrag, beinhaltet der Entwicklungsauftrag neben den Zielen des Unternehmensumfeldes auch die Ziele der unternehmensexternen Stakeholder. Zu den externen Zielen des Entwicklungsprozesses gehören bspw. „minimiere Entwicklungszeit“, „minimiere Entwicklungskosten“, „maximiere Prozessgüte“, „minimiere technisches Ent-

wicklungsrisiko“ etc. Die sich aus den Eigenschaften des gegenwärtig vorliegenden Entwicklungsgegenstands ergebenden externen Ziele hängen von den Attributen (Anzahl beteiligter Fachdisziplinen, Komplexitätsgrad etc.) der zu entwickelnden Module ab. So gilt das Ziel „maximiere Anzahl modulinterner Abstimmungsprozesse“ nur für Module, an deren Entwicklung mehrere Fachdisziplinen beteiligt sind.

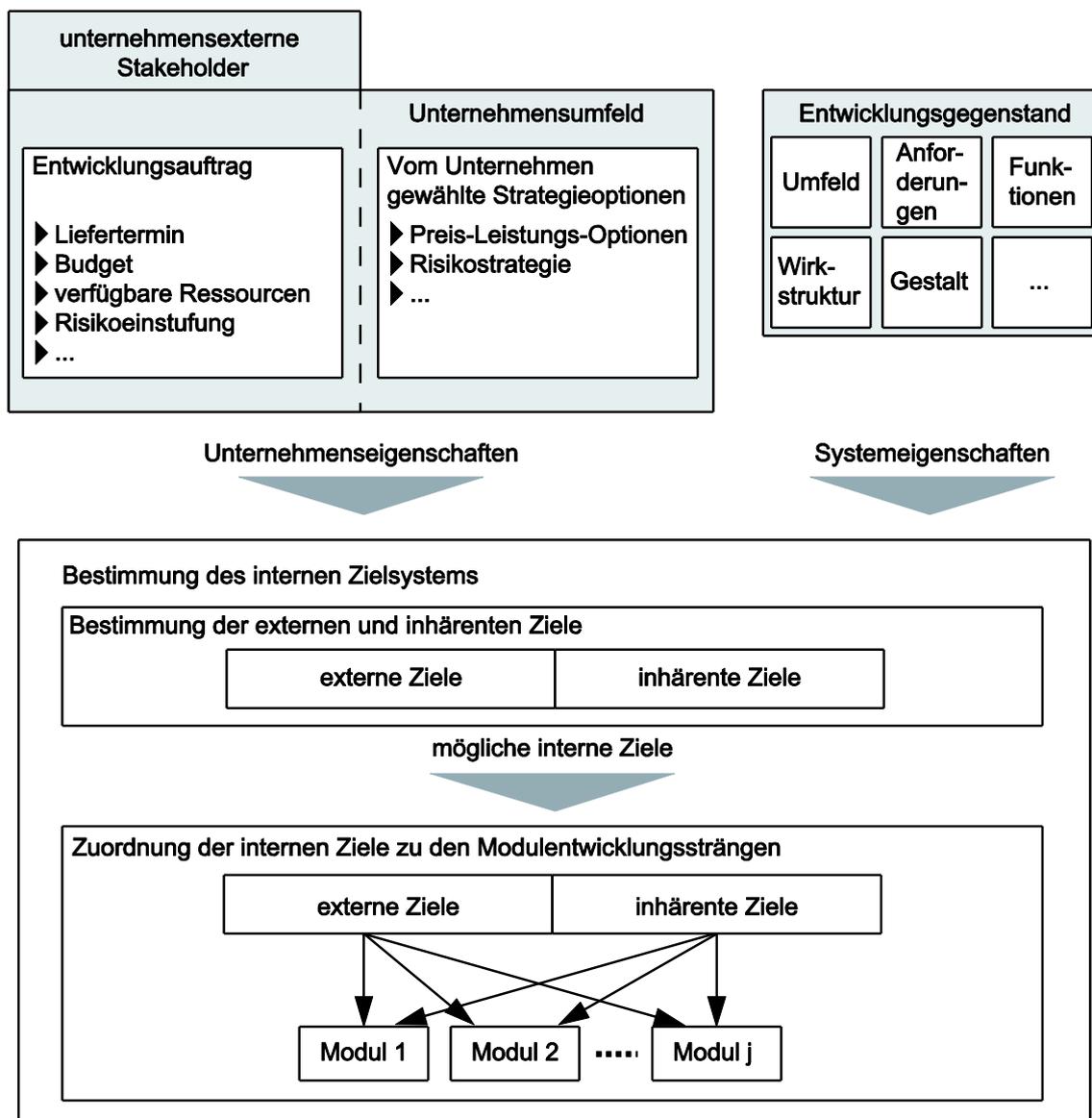


Bild 4-15 Phase 1 des Verfahrens zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen

Die inhärenten Ziele sind Erfahrungen guter ingenieurmäßiger Praxis und gelten für jede Entwicklung eines technischen Systems. Beispiele sind: „minimiere Personalwechsel“ oder „minimiere Medienbrüche“.

Sämtliche externen und inhärenten Ziele werden zusammengenommen und entsprechend den Eigenschaften des gegenwärtig vorliegenden Entwicklungsgegenstandes auf die Modulentwicklungsstränge heruntergebrochen. Die inhärenten Ziele sowie die ex-

ternen, die sich aus dem Unternehmensumfeld ableiten, können eins zu eins je Modul übernommen werden. Bei den externen Zielen, die sich aus den Attributen der Module ergeben haben, werden diese Ziele nur denjenigen Modulen zugeordnet, die die entsprechenden Attribute aufweisen.

Phase 2: Bestimmung der strategischen Zielpriorität

In dieser Phase wird die Priorität der Ziele untereinander für das jeweilige Entwicklungsprojekt aus strategischer Sicht festgelegt. Die in Phase 1 ermittelten externen und inhärenten Ziele werden in der strategieorientierten Relevanz-Matrix unter Berücksichtigung des Entwicklungsauftrags und der durch das Unternehmen gewählten Strategieoption paarweise miteinander verglichen (vgl. Bild 4-16).

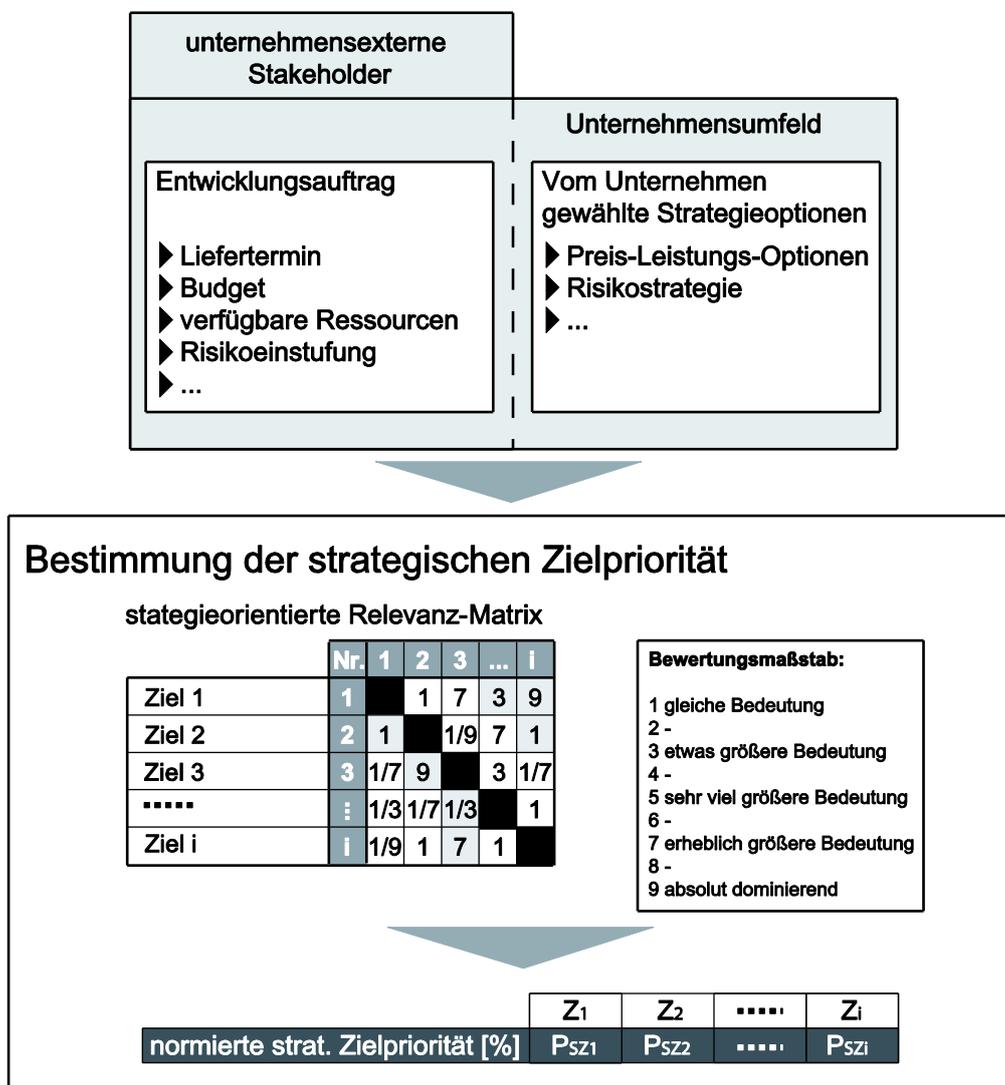


Bild 4-16 Phase 2 des Verfahrens zur situationspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen

Der paarweise Vergleich erfolgt entsprechend Ausschnitten aus dem Analytic Hierarchy Process (AHP) nach SAATY [Saa05, S. 348 ff.]. Hierbei wird paarweise festgelegt ob ein

Ziel im Kontext der durchgeführten Entwicklung eine gleich große oder größere Bedeutung als das andere hat. Die Spanne der Bedeutungsunterschiede reicht in neun Stufen von gleichbedeutend bis hin zu absolut dominierend. Durch Normierung des Eigenvektors der entstehenden Relevanz-Matrix ergeben sich die strategischen Zielprioritäten P_{SZ_i} der externen und inhärenten Ziele für die durchzuführende Entwicklung. Die strategischen Zielprioritäten stellen die grundsätzliche Gewichtung der externen und inhärenten Ziele aus Sicht des Unternehmensumfelds und der unternehmensexternen Stakeholder dar.

Phase 3: Bestimmung der technologischen Zielpriorität

Nach dem Priorisieren der Ziele aus strategischer Sicht werden die Ziele zusätzlich aus technischer Sicht priorisiert. Die technologische Zielpriorisierung trägt dem Sachverhalt Rechnung, dass die Eigenschaften der zu entwickelnden Module eines Systems Einfluss auf die Ziele des zugehörigen Entwicklungsprozesses haben. So ist z.B. bei einem im System stark vernetzten oder sehr komplexen Modul auf eine höhere Prozessgüte Wert zu legen als bei einem weniger vernetzten bzw. komplexen.

Für die Bestimmung der technologischen Zielpriorität ist zunächst die technologische Bedeutung (K_{M_j}) eines jeden Moduls M_j zu ermitteln. Sie wird anhand der im Systemmodell des Entwicklungsgegenstands hinterlegten Attribute bestimmt (vgl. Bild 4-17). Den größten Einfluss auf die technologische Bedeutung haben Komplexität und Dynamikindex eines Moduls. Der Dynamikindex beschreibt wie stark das Modul von anderen Modulen beeinflusst wird oder diese beeinflusst. Wie der Dynamikindex aus dem Systemmodell automatisch abgeleitet werden kann, ist im Anhang A2.1 beschrieben. Alternativ können weitere Kriterien, wie z.B. die Sicherheitsrelevanz eines Moduls, mit für die Bestimmung der technologischen Bedeutung genutzt werden.

Wie sich die technologische Bedeutung eines Moduls auf die Ziele des Entwicklungsprozesses auswirken wird mit Hilfe der technologieorientierten Einflussmatrix festgelegt. In ihr wird festgelegt, ob die technologische Bedeutung negativ oder positiv mit den in Phase 1 ermittelten externen und inhärenten Zielen korreliert. Das einleitend genannte Beispiel der hohen Prozessgüte bei hoher technischer Bedeutung hat demnach eine positive Korrelation. Minimale Entwicklungszeit und -kosten sind Beispiele für negativ mit der technologischen Bedeutung korrelierende Ziele.

Stehen technologische Bedeutung eines jeden Moduls und Korrelation der Ziele mit der technologischen Bedeutung fest wird hieraus die technologische Zielpriorität für jedes Ziel pro Modulentwicklungsstrang festgelegt. Hierfür werden die technologischen Bedeutungen aller Module auf das Intervall von minus Eins bis Eins normiert und mit Hilfe der in Bild 2-1 mittig dargestellten Priorisierungsfunktionen die technologische Priorität $P_{TM_jZ_i}$ für jedes Ziel Z_i über alle Module M_j bestimmt. Abschließend werden die technologischen Prioritäten $P_{TM_jZ_i}$ pro Modul normiert. Im Ergebnis liegen die normierten technologischen Zielprioritäten $P_{TnM_jZ_i}$ für jedes Modul in Prozent vor.

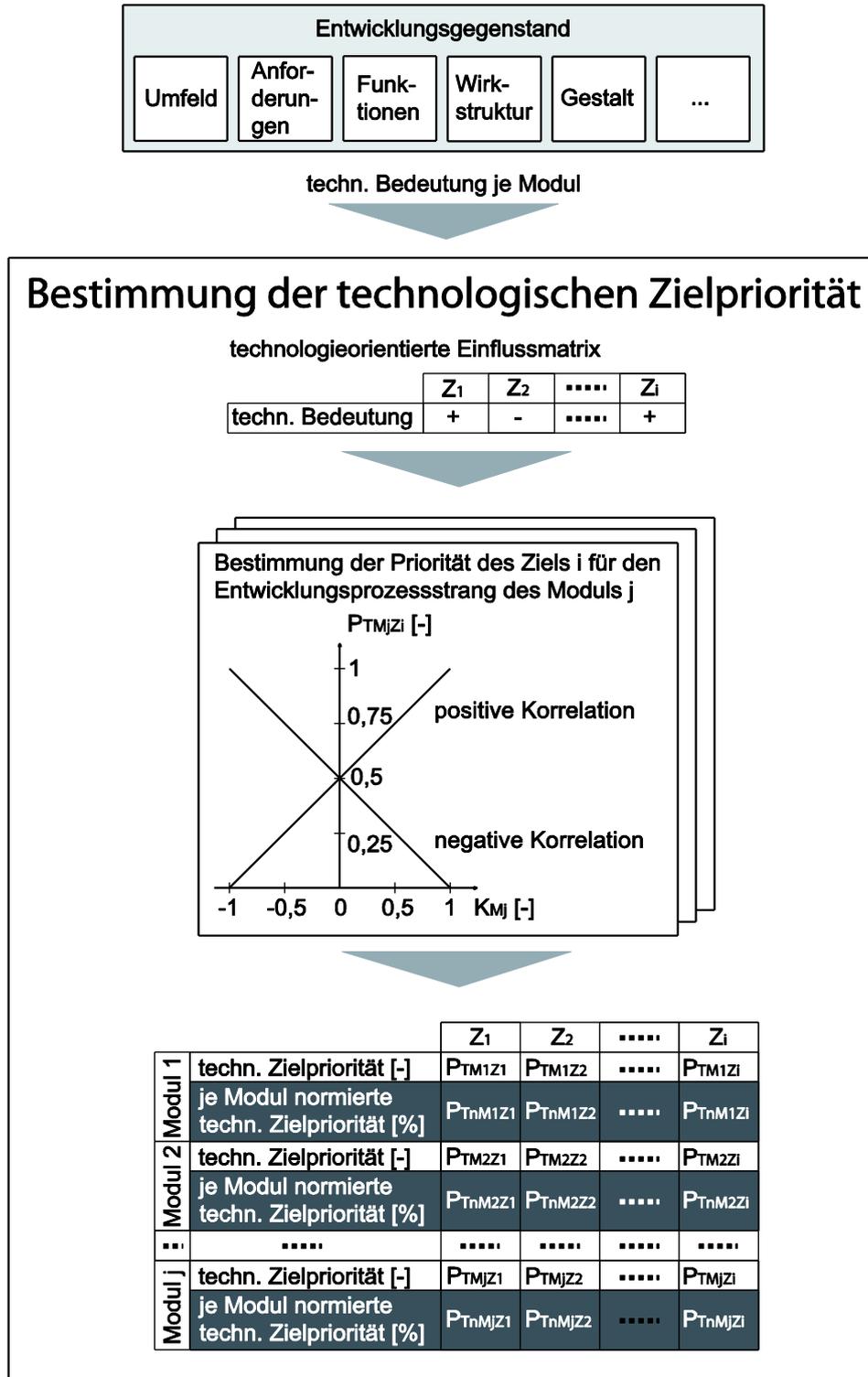


Bild 4-17 Phase 3 des Verfahrens zur situationspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen

Phase 4: Bestimmung des Zielgewichtungsänderungsbedarfs

Neben der Bestimmung der strategischen und technologischen Zielpriorität bedarf es für die situationspezifische Gewichtung der Ziele der Ermittlung der Abweichungen vom aktuelle umzusetzenden Projektplan und dem daraus resultierenden Änderungsbedarf der Gewichtung der Ziele (vgl. Bild 4-18). Hierfür werden die Überwachungsgrößen des Entwicklungsprojekts festgelegt, anhand derer die Ausführung des Entwicklungsprozesses aus Projektmanagementsicht erfasst wird. Beispiele für Überwachungsgrößen sind verbrauchtes Budget, vorliegender Sachfortschrittsgrad der Entwicklungsobjekte, verstrichene Zeit etc.

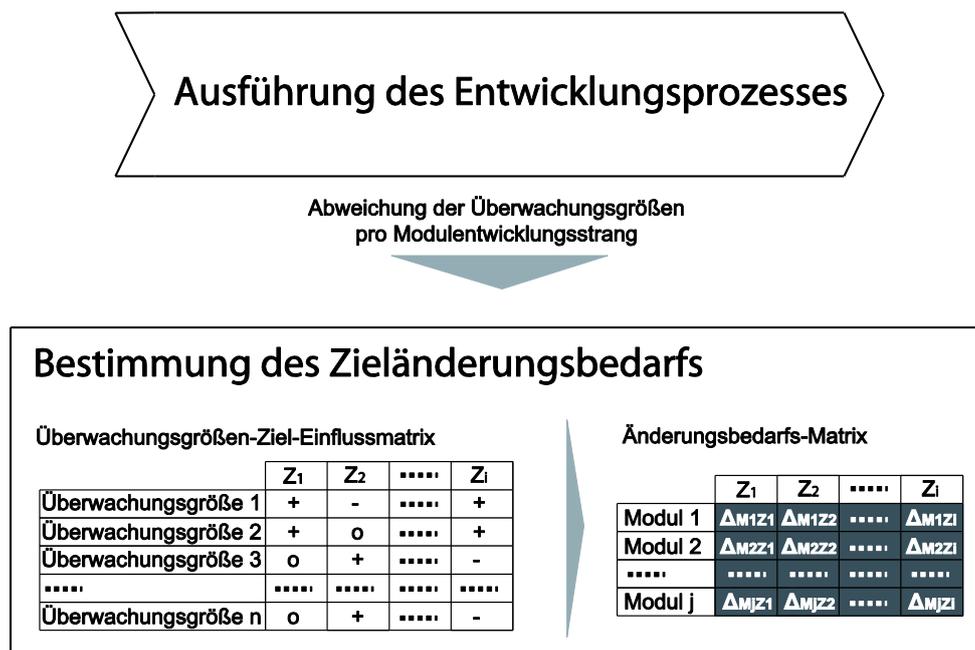


Bild 4-18 Phase 4 des Verfahrens zur situationspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen

Stehen die Überwachungsgrößen fest, wird mit Hilfe der Überwachungsgrößen-Ziel-Einflussmatrix definiert, wie sich die Soll-Ist-Abweichung einer Überwachungsgröße auf die Gewichtung eines oder mehrerer Ziele auswirkt. Die Soll-ist-Abweichung einer Überwachungsgröße kann sich dabei positiv, negativ oder nicht (neutral) auf den Änderungsbedarf der Zielgewichtung auswirken. Ist der Sachfortschritt eines Entwicklungsobjekts bspw. hinter dem Plan wirkt sich dies negativ auf die Gewichtung der Maximierung der Prozessgüte als auch die Maximierung der Anzahl der Personalwechsel aus. Es wirkt sich hingegen positiv auf die Gewichtung der Minimierung der Zeit und nicht auf die Gewichtung der Minimierung der Kosten aus.

Basierend auf der Definition des Einflusses der Überwachungsgrößen auf die Gewichtung der einzelnen Ziele und der Erfassung der aktuellen Soll-Ist-Abweichung der Überwachungsgrößen in Prozent für jeden Modulentwicklungsstrang wird der Zielgewichtungsänderungsbedarf Δ_{MjZi} je Ziel Z_i und Modul M_j ermittelt. Die Änderungsbe-

darfe werden in der Änderungsbedarfs-Matrix zusammengefasst. Der Änderungsbedarf kann in Richtung niedrigerer anzustrebender Gewichtung (Zielgewichtungsänderungsbedarf $< 0\%$), unveränderter Gewichtung (Zielgewichtungsänderungsbedarf $= 0\%$) oder höherer anzustrebender Gewichtung (Zielgewichtungsänderungsbedarf $> 0\%$) sein.

Phase 5: Gewichtung der internen Ziele

Die internen Ziele jedes Modulentwicklungsstranges werden anhand der in Phase 2 und 3 ermittelten strategischen und technologischen Zielpriorität sowie den in Phase 4 bestimmten, aktuellen Zielgewichtungsänderungsbedarfen gewichtet. Hierfür kommt die im nachfolgenden Bild 4-19 dargestellte Gewichtungsfunktion zur Anwendung⁴³. Der y-Achsenabschnitt b_{MjZi} ist das arithmetische Mittel aus gewichteter strategischer Zielpriorität P_{SZi} und gewichteter technologischer Zielpriorität P_{TnMjZi} des Ziels Z_i für Modul M_j . Die Steigung der Gewichtungsfunktion a ist für alle Gewichtungsfunktionen gleich und ergibt sich aus der Risikoeinstufung des Entwicklungsprojekts. Sie spiegelt die Empfindlichkeit des selbstoptimierenden Entwicklungsprozesses wider, mit der er auf Abweichungen in der Zielerfüllung reagiert. Je größer das Risiko ist, desto größer ist die Steigung. Dadurch wird dem Ansatz Rechnung getragen risikoreiche Projekte enger zu führen als solche geringeren Risikos.

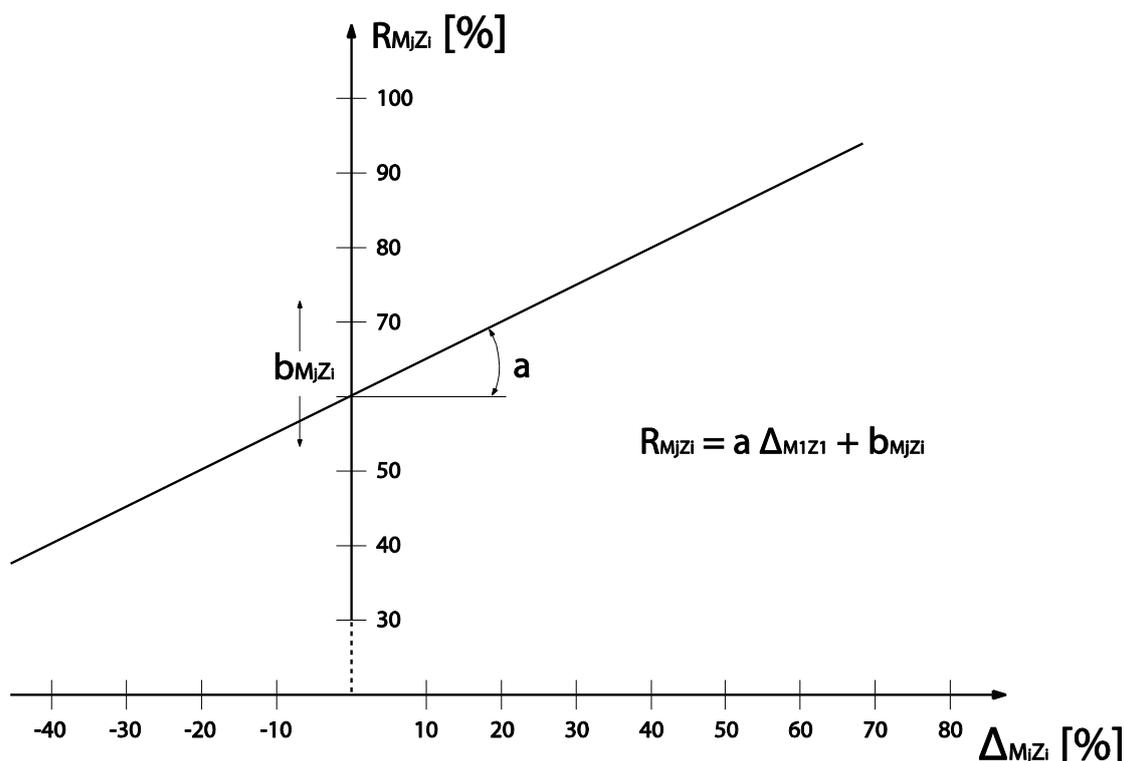


Bild 4-19 Gewichtungsfunktion zur Bestimmung der Bedeutung [-] des Ziels Z_i für den Entwicklungsstrang des Moduls M_j

⁴³ Alternativ zur dargestellten linearen Gewichtungsfunktion können auch andere Funktionen (z.B. exponentiell, Stufenfunktion etc.) verwendet werden.

Aus der Gewichtungsfunktion und dem Zielgewichtsänderungsbedarf Δ_{MjZi} wird die Relevanz R_{MjZi} je Ziel und Modul über alle Module ermittelt (vgl. Bild 4-20). Zielerfüllungsgrad und Steigung der Gewichtungsfunktion dürfen keine derart hohen Werte annehmen, dass sich ein negativer Wert für die absolute Relevanz eines Ziels ergeben würde. Um dies zu verhindern werden alle b_{MjZi} in einem solchen Fall gleichmäßig angehoben bis sich wieder ein positiver Wert für die absolute Relevanz einstellt.

Durch Normierung der absoluten Relevanzen pro Modul ergeben sich die Gewichtungen G_{MjZi} der internen Ziele je Modul in Prozent, anhand derer aus der Menge der mit dem in Kapitel 4.3.2.1 beschriebenen Verfahrens zur systemmodellbasierten Prozessplanung generierten Prozesspläne der für die gegenwärtige Entwicklungssituation optimale Prozessplan ausgewählt wird. Das Verfahren zur Auswahl des optimalen Prozessplans auf Basis des internen Zielsystems wird im nachfolgenden Kapitel 4.3.2.3 beschrieben.

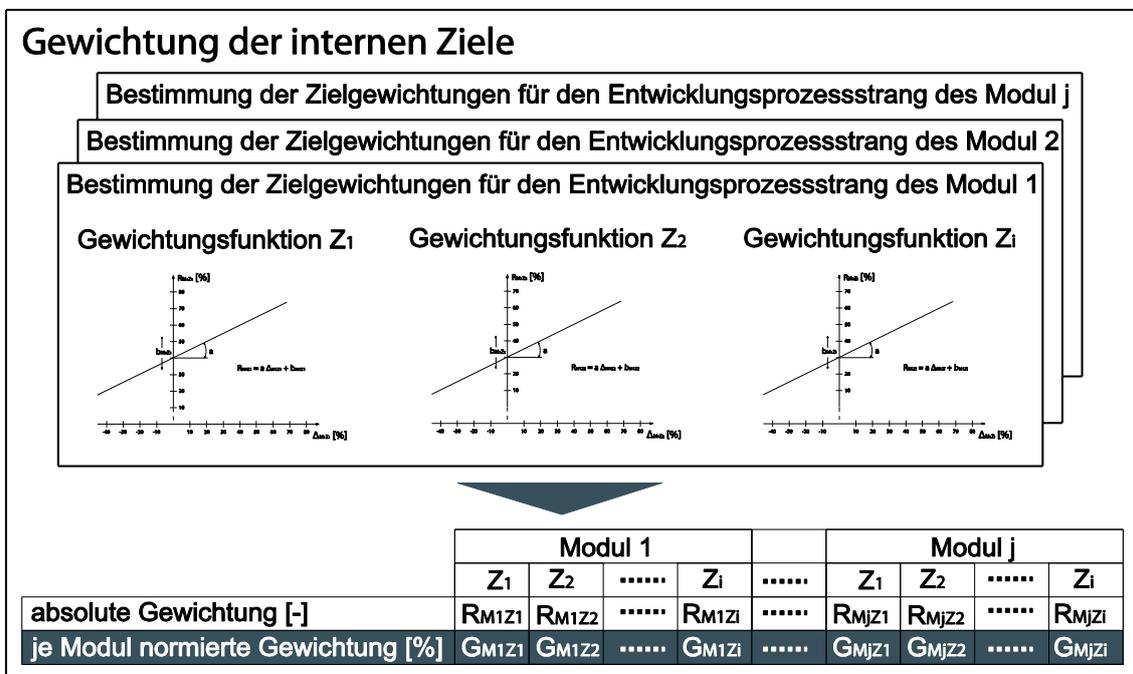


Bild 4-20 Phase 5 des Verfahrens zur situationspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen

4.3.2.3 Verfahren zur zielkonformen Planauswahl

Die in den beiden vorstehenden Kapiteln 4.3.2.1 und 4.3.2.2 beschriebenen Verfahren liefern für die gegenwärtige Entwicklungssituation paretooptimale Entwicklungsprozesse mit denen der aktuell vorliegenden Entwicklungsgegenstand in die vollständige Beschreibung des zu entwickelnden technischen Systems überführt werden kann sowie das situationspezifische interne Zielsystem des Entwicklungsprozesses. Mit dem in diesem Kapitel beschriebenen Verfahren wird nun basierend auf dem internen Zielsystem der

für die Entwicklungssituation optimale Plan aus der Menge der möglichen Pläne entsprechend dem in Bild 4-21 beschriebenen Vorgehens ausgewählt.

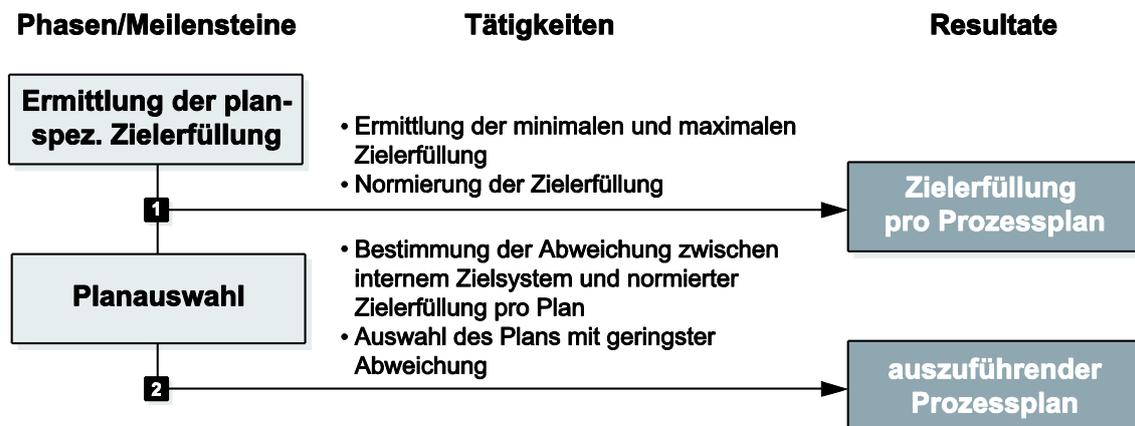


Bild 4-21 Ablauf des Verfahrens zur zielkonformen Planauswahl

Phase 1: Ermittlung der planspezifischen Zielerfüllung

Zunächst werden pro Modul die für die internen Ziele relevanten Eigenschaften der paretooptimalen Pläne ermittelt. Bspw. werden die in den Plänen implizierten minimal und maximal möglichen Dauern pro Modulentwicklungsstrang, deren minimal und maximal möglichen Kosten, deren minimal und maximal mögliche Prozessgüte etc. ermittelt. Prozesse, die mehrere Module betreffen, wie z.B. modulübergreifende Abstimmungsprozesse, werden hierbei anteilig den betroffenen Modulentwicklungssträngen zugeordnet. Anschließend werden die Werte aller Pläne mit Hilfe der identifizierten Minima und Maxima sowohl pro Ziel pro Modulentwicklungsstrang als auch über die Ziele eines Moduls pro Plan auf das Intervall von 0% bis 100% normiert.

Phase 2: Planauswahl

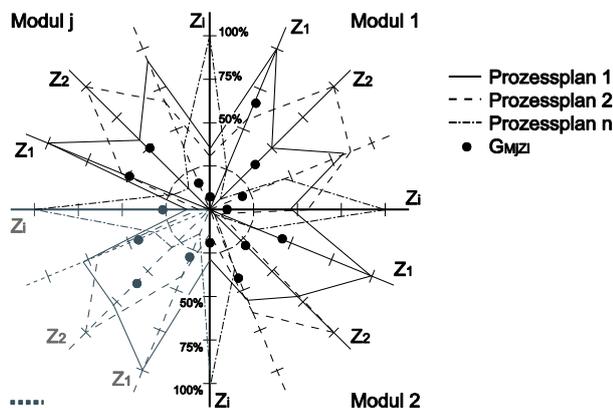
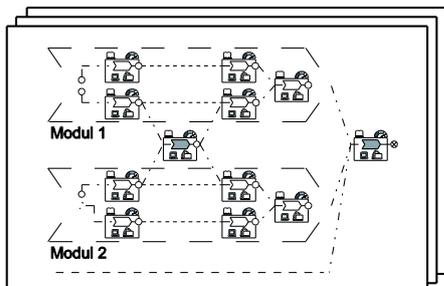
Im nächsten Schritt werden die Gewichtungen G_{MjZi} der internen Ziele herangezogen und für jeden der paretooptimalen Pläne der Abstand zu diesen Gewichtungen pro Ziel und Modulentwicklungsstrang berechnet. Im Ergebnis wird der Plan mit dem geringsten Gesamtabstand für die weitere Ausführung des Entwicklungsprozesses ausgewählt und initialisiert⁴⁴ (vgl. Spinnennetz-Diagramm Bild 4-22).

⁴⁴ Der Optimalitätsbegriff wird somit im Kontext des in dieser Arbeit beschriebenen selbstoptimierenden Entwicklungsprozesses als „die für die gegebene Entwicklungssituation beste Lösung im Rahmen des Möglichen“ verstanden. Die zentrale Planung bei der Prozessgenerierung garantiert hierbei die Optimalität auf Gesamtsystemebene gegenüber den Rahmenbedingungen des Entwicklungsprozesses. Die situations- und modulspezifische Zielgewichtung garantiert die Optimalität bezogen auf die Bedürfnisse der Modulentwicklungen. Der Autor hat die in dieser Arbeit zur Anwendung kommenden Verfahren bewusst derart konzipiert, dass die Gesamtsystemsicht die Modulsicht im Konfliktfall dominiert, da die Optimierung der einzelnen Modulentwicklungsstränge nicht zwangsläufig zu einer optimalen Entwicklung aus Gesamtsystemsicht führt.

gewichtetes internes Zielsystem

	Modul 1				Modul 2				Modul j				
	Z ₁	Z ₂	Z _i	Z ₁	Z ₂	Z _i	Z ₁	Z ₂	Z _i
absolute Gewichtung [-]	R _{M1Z1}	R _{M1Z2}	R _{M1Zi}	R _{M2Z1}	R _{M2Z2}	R _{M2Zi}	R _{MjZ1}	R _{MjZ2}	R _{MjZi}
je Modul normierte Gewichtung [%]	G _{M1Z1}	G _{M1Z2}	G _{M1Zi}	G _{M2Z1}	G _{M2Z2}	G _{M2Zi}	G _{MjZ1}	G _{MjZ2}	G _{MjZi}

Paretooptimale Prozesspläne



Ausführender Prozessplan

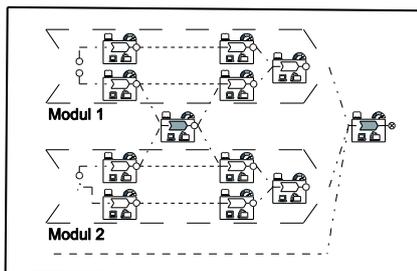


Bild 4-22 Struktur des Verfahrens zur zielkonformen Planauswahl

4.4 Vorgehensmodell zur Initialisierung und Nachbereitung

Die vorstehenden Ausführungen haben die theoretischen Grundlagen zur Anwendung des Paradigmas der Selbstoptimierung auf den Entwicklungsprozess fortschrittlicher mechatronischer Systeme sowie Werkzeuge zur Umsetzung der Selbstoptimierung in einem konkreten Entwicklungsprozess beschrieben. Gegenstand dieses Kapitels ist ein Vorgehensmodell mit dem die Rahmenbedingungen für die Umsetzung eines selbstoptimierenden Entwicklungsprozesses innerhalb eines Unternehmens geschaffen werden. Zudem beschreibt es wie die im Rahmen von Entwicklungsprojekten innerhalb eines Unternehmens gewonnenen Erfahrungen für künftige Entwicklungsprojekte nutzbar gemacht werden sollen. Es ist entsprechend den beiden bisher nicht betrachteten Aufgabenkomplexe des Projektmanagements „Initiierung“ und „Abschluss“ zuzuordnen (vgl. Abschnitt 4.2.3, Bild 4-4). Bild 4-23 zeigt das Vorgehensmodell zur Initialisierung und Nachbereitung des selbstoptimierenden Entwicklungsprozesses als bei der Planung von Entwicklungsprojekten in einem Unternehmen immer wieder zu durchlaufendes zyklisches Vorgehen.



Bild 4-23 Vorgehensmodell zur Initialisierung und Nachbereitung des selbstoptimierenden Entwicklungsprozesses

Den Einstieg bildet das **initiale Befüllen der Wissensbasis**. Hier werden die für das Unternehmen verfügbaren Entwicklungstätigkeiten in Form von Prozessbausteinen entsprechend Kapitel 4.3.1.2 definiert und dokumentiert. Einhergehend mit der Dokumentation der Tätigkeiten, aus denen der Entwicklungsprozess synthetisiert werden kann, werden auch die zur Ausführung der Tätigkeiten benötigten Fähigkeiten sowie verfügbaren Ressourcen und Hilfsmittel spezifiziert. Grundsätzlich müssen nicht nur innerhalb des Unternehmens verfügbare Tätigkeiten, Fähigkeiten, Ressourcen und Hilfsmittel betrachtet werden. Optimalerweise werden auch solche spezifiziert, auf die das Unternehmen nach Extern Zugriff hat, wie z.B. bei Partnerunternehmen oder Ingenieurdienstleistern wobei das Aufwand-Nutzenverhältnis einer solchen Spezifikation in Abhängigkeit der Regelmäßigkeit der Zusammenarbeit mit diesen externen Organisationseinheiten zu prüfen ist. Zudem werden im Unternehmen verwendeten Aufwandsschätzverfahren in den Funktionen der Ressourcen abgebildet. Im Ergebnis liegt ein Leistungsfähigkeits- und Kompetenzprofil des Unternehmens in Form der Prozessbausteine vor. Ergänzend werden die im Unternehmen zu erarbeitenden Zielentwicklungsobjekte definiert. Dies sind zum einen die Standardobjekte jeder Entwicklung eines technischen Systems wie z.B. Stücklisten oder Einzelteilzeichnungen, zum anderen können aber auch Zusatzdokumente, wie z.B. branchen- oder unternehmensspezifische Normen und Richtlinien hinzukommen.

Zur Sicherstellung eines Lerneffekts über die Dauer des Bestehens eines Unternehmens ist immer dann wenn ein spezifischer Entwicklungsprozess abgeschlossen ist – ggf. bei

langfristigen Projekten auch nach einem Prozessabschnitt – die **Wissensbasis** zu **aktualisieren**. Hierzu gehören bspw. die Anpassung des Qualifikationsniveaus einer Ressource, die Modellierung einer neuen Ein- und Ausgangsbeziehung für eine Tätigkeit, die weitere Detaillierung einer Aufwandsschätzfunktion oder die Definition einer gänzlich neuen Tätigkeit für eine neue Spezialisierung. Auf diese Weise werden ggf. entdeckte, effizientere Möglichkeiten zur Durchführung von Entwicklungsprozessen im Unternehmen für künftige Entwicklungsprojekte verfügbar gemacht.

Neben dem Aufbau der für die Umsetzung eines selbstoptimierenden Entwicklungsprozesses basalen Wissensbasis ist eine **unternehmensspezifische Konfiguration** des Rahmenwerks vorzunehmen. Hier werden die im Unternehmen über alle Entwicklungsprojekte potentiell anzuwendenden Ziele, Risiko-Metriken und Überwachungsgrößen festgelegt. Zudem wird festgelegt, wie ein Ziel mit der technologischen Bedeutung eines Moduls und den Überwachungsgrößen korreliert. Diese Phase legt also die grundlegenden Möglichkeiten und Wirkmechanismen des Rahmenwerks innerhalb des Unternehmens fest.

Sind die grundlegenden Möglichkeiten und Wirkmechanismen geklärt kann das Rahmenwerk konkret für einen spezifischen Entwicklungsprozess konfiguriert werden. Hierzu werden in der **projektspezifischen Konfiguration** die finanziellen, zeitlichen, personellen und materiellen Randbedingungen des Projekts definiert und aus der Menge potentieller Ziele und Überwachungsgrößen die für das Entwicklungsprojekt gültigen ausgewählt. Weiterhin werden die Mechanismen zur Bestimmung der strategischen und technologischen Zielpriorität sowie des Zielgewichtungsänderungsbedarfs konfiguriert. Hierzu werden strategieorientierte Relevanz-Matrix, technologieorientierte Einflussmatrix sowie die Überwachungsgrößen-Ziel-Einflussmatrix projektspezifisch ausgeprägt. Sind diese projektorganisatorischen Rahmenbedingungen geklärt, wird die initiale Produktspezifikation vorgenommen, die die erwartete grundlegende Struktur des zu entwickelnden Produkts und eine erste Einschätzung von dessen Charakteristika beinhaltet. Abschließend wird die Risikoklasse des Projekts abgeschätzt, um die Empfindlichkeit der Werkzeuge zur Umsetzung der Selbstoptimierung festzulegen.

Die anschließende Phase ist die **Ausführung des Entwicklungsprozesses**. Diese vollzieht sich entsprechend den in den vorangegangenen Kapiteln beschriebenen Mechanismen selbstoptimierend.

Es sei an dieser Stelle betont, dass das hier beschriebene Vorgehen einen grundlegenden Ablauf beschreibt. Es ist nicht ausgeschlossen, dass einzelne Tätigkeiten aus den Phasen des beschriebenen Vorgehensmodells auch während der Ausführung des selbstoptimierenden Entwicklungsprozesses wahrgenommen werden. Die Werkzeuge zur Umsetzung des selbstoptimierenden Entwicklungsprozess können derartige Vorkommnisse auffangen und in die Ausführung der Selbstoptimierung integrieren.

5 Verifikation des Rahmenwerks

Nachfolgend wird das im vorangegangenen Kapitel 4 entwickelten Rahmenwerk verifiziert. Zu diesem Zweck sind die den selbstoptimierenden Entwicklungsprozess umsetzenden Verfahren aus Kapitel 4.3.2 prototypisch in verschiedenen Software-Werkzeugen umgesetzt. Sie werden im nachfolgenden Kapitel 5.1 beschrieben. Die Verifikation erfolgt am Beispiel des in Kapitel 5.2 ausführlich beschriebenen Schienenfahrzeugs RailCab⁴⁵. Ziel der Verifikation ist der Funktionsnachweis des Rahmenwerks entsprechend dem in Kapitel 4.2.2 beschriebenen Ablauf der Selbstoptimierung. Hierfür werden vier Verifikationsszenarien untersucht, in denen die in Kapitel 4.2.2 beispielhaft genannten Auslöser des Selbstoptimierungsprozesses (veränderte strategische Vorgaben, veränderte Eigenschaften des Entwicklungsgegenstandes, Veränderungen der Handlungsobjekte und Planabweichungen) herangezogen und die durch sie angestoßenen Reaktionen mit dem gewünschten Ablauf entsprechend dem Selbstoptimierungsprozess verglichen werden. Die Verifikation stützt sich bezüglich des technischen Aufbaus und des logischen Ablaufs des Entwicklungsprozesses auf respektive gewonnene Real-Daten über das System RailCab. Da keine empirischen Daten in ausreichender Qualität über die Aufwände der Entwicklung vorhanden sind und die Entwicklung von Aufwandschätzverfahren nicht Gegenstand der vorliegenden Arbeit ist, wird an den Stellen, an denen die paretooptimalen Prozesspläne als Eingangsdaten dienen, auf fiktive Plandaten zurückgegriffen. Folglich sind die in diesem Kapitel vorgenommenen Betrachtungen nicht als empirische Validierung des Rahmenwerks für den industriellen Praxiseinsatz zu werten. Die Ergebnisse der Verifikation sind im nachfolgenden Kapitel 5.3 dargestellt.

5.1 Software-Werkzeuge zur prototypischen Umsetzung des Rahmenwerks

Für die Verifikation des Rahmenwerks für einen selbstoptimierenden Entwicklungsprozess wurden zwei prototypische Software-Werkzeuge entwickelt. Das erste ist der Prozessplaner (Kapitel 5.1.1), der das Verfahren zur systemmodellbasierten Prozessplanung aus Kapitel 4.3.2.1 prototypisch umsetzt. Die Verfahren zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen (Kapitel 4.3.2.2) und zur zielkonformen Planauswahl (Kapitel 4.3.2.3) werden mit Hilfe zweier Excel-Editoren umgesetzt. Der Aufbau der beiden Editoren ist in Kapitel 5.1.2 dargestellt.

⁴⁵ Siehe ergänzend Kurzbeschreibung des RailCabs als Beispiel für selbstoptimierende Systeme in Kapitel 2.3.4.

5.1.1 Prozessplaner

Der Prozessplaner ist in der Entwicklungsumgebung Eclipse⁴⁶ mit der Programmiersprache Java implementiert. Seine Struktur und Funktionsweise orientieren sich an den Bestandteilen des Verfahrens zur systemmodellbasierten Prozessplanung entsprechend Bild 4-8. Er beinhaltet eine Wissensbasis für generische Prozesse, eine Wissensbasis für personelle Ressourcen, einen Editor zur Spezifikation des zu entwickelnden mechatronischen Systems (Entwicklungsgegenstand), einen Netzplangenerator zur teilautomatisierten Erstellung des passenden Vorgangsfolgeplans und den Scheduler zur Generierung paretooptimaler Prozesspläne. Die einzelnen Bestandteile und deren Funktionsweise werden im Folgenden erläutert.

Wissensbasis für generische Prozesse

In der Wissensbasis für Prozesse sind Grundbausteine und Abstimmungsbausteine zur Generierung von Vorgangsfolgeplänen für die Entwicklung eines mechatronischen Systems im XML-Format hinterlegt (vgl. Abschnitt 4.3.1.2). Eine tabellarische Auflistung der als Prozessbausteine modellierten Entwicklungstätigkeiten ist dem Anhang A3 zu entnehmen. Die Grundbausteine sind in dieser prototypischen Implementierung bereits zu fachdisziplin- bzw. spezialisierungsspezifischen Prozesssträngen verknüpft. Abhängig von der oder den einem Modul des zu entwickelnden Systems zugeordneten Fachdisziplinen (Maschinenbau, Elektronik, Elektrik, Regelungstechnik oder Softwaretechnik) wird für das Modul eine Menge von Entwicklungstätigkeiten mit vorgegebener logischer Reihenfolge bereitgestellt. Ist ein Modul mehreren Fachdisziplinen zugeordnet, sind die modulinternen Abstimmungsbausteine bereits in den Prozesssträngen enthalten. Die modulübergreifenden Abstimmungsbausteine werden abhängig davon bereitgestellt welchen Fachdisziplinen oder Spezialisierungen zwei verschiedene Module zugeordnet wurden.

Jede Entwicklungstätigkeit ist als complex task modelliert. Sie kann hierarchisch in Sub-Tätigkeiten unterteilt werden. Diese können als sukzessiv oder alternativ zueinander ausführbare Entwicklungstätigkeiten modelliert werden. Mit diesem Mechanismus werden Detaillierungen von Entwicklungstätigkeiten oder deren unterschiedliche Ausführung mit z.B. unterschiedlichen Entwicklungswerkzeugen abgebildet. Für jede Entwicklungstätigkeit ist die Fähigkeit definiert, die benötigt wird, um diese Tätigkeit auszuführen. Mit Hilfe dieser benötigten Fähigkeit können die Ressourcen identifiziert werden, die für die Ausführung der Tätigkeit in Frage kommen.

Wissensbasis für personelle Ressourcen

Die für die Entwicklung des mechatronischen Systems zur Verfügung stehenden personellen Ressourcen werden in einer eigenen Wissensbasis in XML abgebildet. Jeder Resource sind entsprechend ihrer Qualifikation die vorstehend bereits erwähnten Fähigkei-

⁴⁶ <http://www.eclipse.org/>

ten zugeordnet, die zur Ausführung einzelner Entwicklungstätigkeiten benötigt werden. Für jede Fähigkeit können Funktionen zur Aufwandsschätzung⁴⁷ hinterlegt werden, die widerspiegeln wie lange die entsprechende Ressource für die Ausführung der mit der Fähigkeit ausgeführten Entwicklungstätigkeit braucht, wie viel dies kostet und welche Prozessgüte bei der Ausführung der Entwicklungstätigkeit zu erwarten ist. In den Funktionen kann neben der Qualifikation der personellen Ressource auch der bisherige Sachfortschrittsgrad, die Komplexität und der Dynamikindex (vgl. Anhang A2.1) des zu entwickelnden Moduls eingebracht werden. Hierdurch soll der Sachverhalt abgebildet werden, dass die Entwicklung eines Moduls länger dauert und mehr kostet, je komplexer es ist.

Editor zur Spezifikation des zu entwickelnden mechatronischen Systems

Struktur und Charakteristik der einzelnen Module des zu entwickelnden mechatronischen Systems werden mit Hilfe des in Bild 5-1 dargestellten Editors spezifiziert. Er verfügt hierfür über verschiedene Eingabemasken zur sukzessiven Beschreibung der Module.

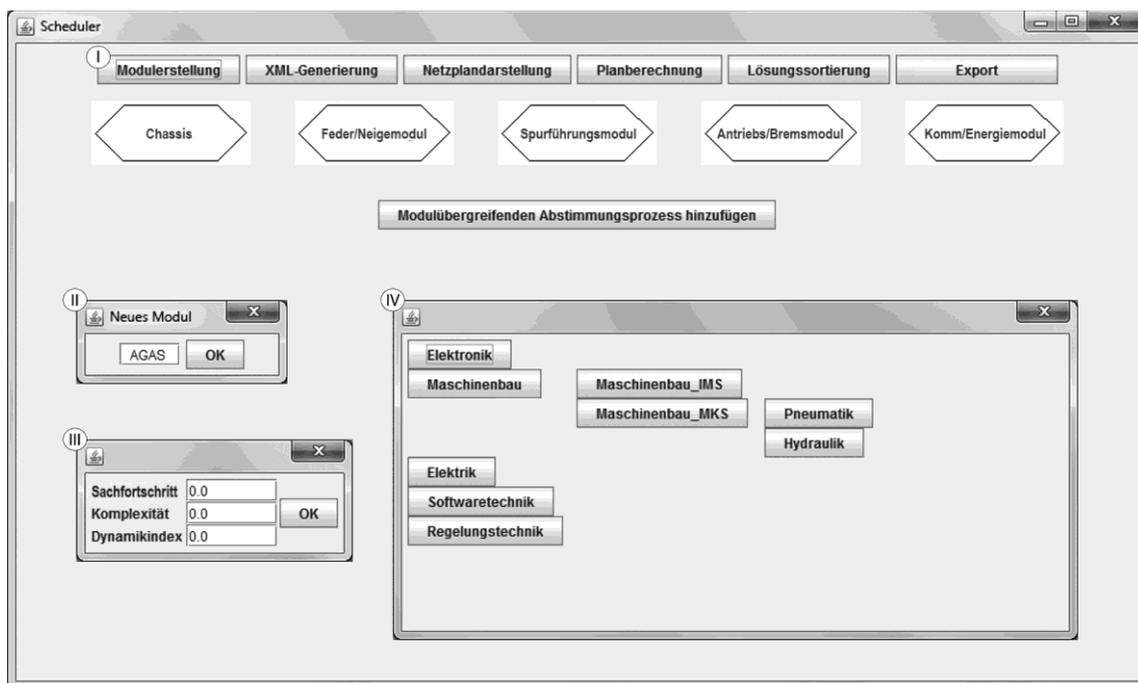


Bild 5-1 Spezifikation der Module des zu entwickelnden mechatronischen Systems

Mit dem Reiter „Modulerstellung“ wird ein neues Modul angelegt und benannt. Anschließend werden der bisher bei der Entwicklung erreichte Sachfortschrittsgrad des Moduls, dessen Komplexitätsgrad und dessen Dynamikindex eingegeben. Die Skala reicht

⁴⁷ An dieser Stelle sei nochmals betont, dass die Entwicklung von Aufwandsschätzverfahren nicht Gegenstand dieser Arbeit ist, weshalb in dem hier vorgestellten prototypischen Software-Werkzeug die Voraussetzung für die Hinterlegung solcher Verfahren geschaffen wurden, aber keine empirisch belegten Schätzfunktionen hinterlegt sind.

dabei von Null bis Drei. Sind die Werte wie im Beispiel jeweils Null, bedeutet dies einen geringen Sachfortschrittsgrad, Komplexitätsgrad und Dynamikindex. Im Anschluss an die Spezifikation der technischen Eigenschaften des Moduls erfolgt in der letzten Eingabemaske dessen Klassifikation. Ein Modul kann einer oder mehreren Fachdisziplinen (Maschinenbau, Elektronik, Elektrik, Regelungstechnik oder Softwaretechnik) zugeordnet werden. Zudem können in den einzelnen Fachdisziplinen auch Spezialisierungen ausgewählt werden. Beispielsweise kann ein Modul der Fachdisziplin „Maschinenbau“ die Spezialisierungen „Maschinenbau – Mehrkörpersystem“ und dort wiederum die Spezialisierungen „Pneumatik“ oder „Hydraulik“ haben. Sämtliche so angelegten Module erscheinen als Sechsecke im oberen Teil des Editors.

Wenn mindestens zwei Module spezifiziert sind schlägt der Editor in Abhängigkeit von den Charakteristika automatisch modulübergreifende Abstimmungstätigkeiten vor, mit denen die beiden Module während des Entwicklungsprozesses synchronisiert und integriert werden. Sind beispielsweise zwei Module der Fachdisziplin „Maschinenbau“ zugeordnet, werden die Abstimmungstätigkeiten „Kinematik auf Systemebene abstimmen“ oder „verfeinerte Gestalt auf Systemebene abstimmen“ vorgeschlagen. Die möglichen Abstimmungstätigkeiten werden über die Schaltfläche „Modulübergreifenden Abstimmungsprozess hinzufügen“ aufgerufen (siehe Bild 5-2).

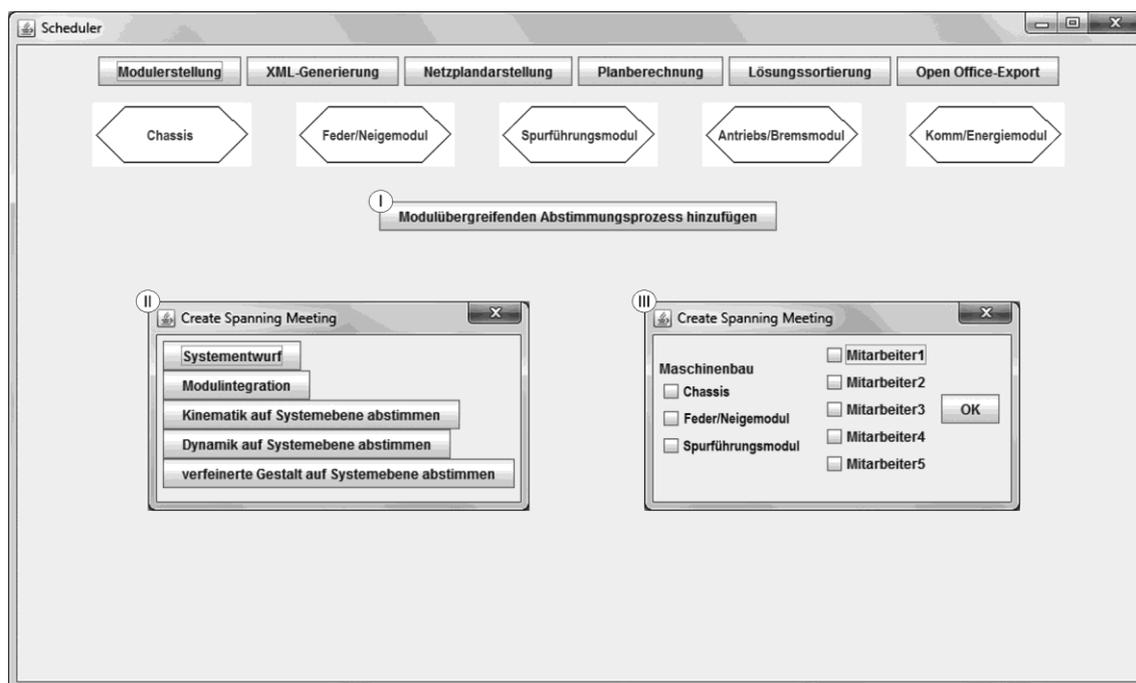


Bild 5-2 Auswählen modulübergreifender Abstimmungsbausteine

Der Benutzer wählt abhängig von der Struktur des zu entwickelnden Systems die jeweiligen Abstimmungstätigkeiten aus und legt in der nächsten Eingabemaske fest, für welche Module sie anzuwenden sind. Zudem werden diejenigen Mitarbeiter festgelegt, die für die Abstimmungstätigkeit notwendig sind.

Netzplangenerator

Sind sämtliche Module und mögliche Abstimmungstätigkeiten, die im Entwicklungsprozess berücksichtigt werden sollen, spezifiziert, erzeugt der Netzplangenerator einen entsprechenden objektorientierten Vorgangsfolgeplan. Hierfür werden für jedes Modul in Abhängigkeit von dessen Eigenschaften (Fachdisziplin, Spezialisierung usw.) die jeweils in der Wissensbasis für generische Prozesse hinterlegten Prozessstränge ausgewählt und modulspezifisch ausgeprägt. Zudem werden die ausgewählten Abstimmungstätigkeiten automatisch an die vordefinierten Stellen im Gesamtentwicklungsprozess eingesetzt. Bild 5-3 zeigt beispielhaft einen Ausschnitt aus dem jeweiligen Entwicklungsstrang zweier Module und die automatisch eingesetzten modulübergreifenden Abstimmungstätigkeiten. Der resultierende Vorgangsfolgeplan spiegelt sämtliche für die Entwicklung des gewünschten mechatronischen Systems notwendigen Entwicklungstätigkeiten und deren logische Reihenfolge wider. Er bildet die Grundlage für die Generierung der mit den vorhandenen Ressourcen möglichen zeit-, kosten- und prozessgüte-behafteten Prozesspläne.

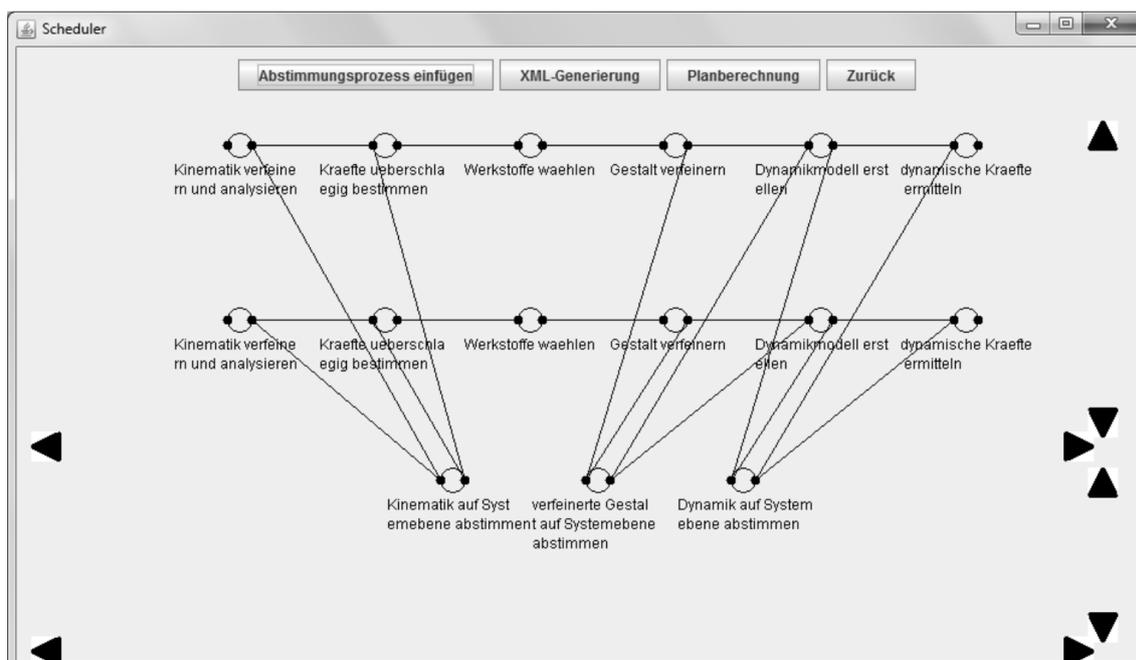


Bild 5-3 Automatisch im Prozessplaner generierter Netzplan

Scheduler zur Generierung paretooptimaler Prozesspläne

Ist der logische Vorgangsfolgeplan mit Hilfe des Netzplangenerators erstellt, wird dieser an den Scheduler übergeben. Der Scheduler arbeitet mit dem von KLÖPPER entwickelten Algorithmus zur Generierung paretooptimaler Prozesspläne [Klö10, S. 635ff.], [Klö11, S. 1ff.]. Dieser wurde dahingehend erweitert, dass für die Ausführung einer Entwicklungstätigkeit auch zwei oder mehr personelle Ressourcen herangezogen werden müssen und bei der Bewertung der Pläne neben Zeit, Kosten und Prozessgüte auch die Anzahl der Personalwechsel innerhalb eines Moduls berücksichtigt wird.

Zunächst werden die für die Umsetzung des Vorgangsfolgeplans einsetzbaren personellen Ressourcen anhand ihrer Fähigkeiten identifiziert. Die durch den Einsatz der jeweiligen Ressource für eine Entwicklungstätigkeit zu erwartende Zeit, die Kosten und die Prozessgüte können mit den für die Ressource hinterlegbaren Aufwandschätzfunktionen ermittelt werden. Es wird jeweils diejenige Ressource für eine Entwicklungstätigkeit eingelastet, die die besten Ergebnisse liefert. Zum Schluss liegen die pareto-optimalen Prozesspläne (Termin-, Kosten-, Hilfsmittel- und Ressourcenplanung) für die Entwicklung des mechatronischen Systems vor. Sie können wahlweise als Gantt-Diagramm angezeigt oder in tabellarischer Form exportiert werden. Beim Export werden Dauer, Kosten, Prozessgüte und Anzahl Personalwechsel der einzelnen Prozesspläne sowohl für jedes Modul als auch für das Gesamtsystem ausgegeben.

5.1.2 Editoren für die Planauswahl

Für die Auswahl eines für die aktuelle Entwicklungssituation optimalen Prozessplans wurden das Verfahren zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen (Kapitel 4.3.2.2) und das Verfahren zur zielkonformen Planauswahl (Kapitel 4.3.2.3) mit Hilfe zweier in Microsoft Excel realisierten Editoren umgesetzt. Der erste dient der situationsspezifischen Gewichtung der Entwicklungsprozessziele für jedes einzelne Modul des zu entwickelnden Systems. Im zweiten Editor werden die im ersten Editor ermittelten Gewichtungen der Entwicklungsprozessziele sowie die im Prozessplaner aus Kapitel 5.1.1 generierten paretooptimalen Prozesspläne importiert und auf dieser Basis der für die aktuelle Entwicklungssituation am besten geeignete Prozessplan ausgewählt.

Excel-Editor für die Bestimmung der Gewichtung der Entwicklungsprozessziele

Der Excel-Editor für die Umsetzung des Verfahrens zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen unterteilt sich entsprechend Bild 5-4 in vier Bereiche. Die Eingabemaske im oberen linken Bereich sowie die darunter dienen der Bestimmung der strategischen und der technischen Zielpriorität. Die Eingabemaske rechts dient der Spezifikation der Überwachungsgrößen und ihrer modulweisen Erfassung. Die Ausgabemaske im unteren Teil des Editors gibt das situationsspezifische interne Zielsystem des Entwicklungsprozesses aus. Grundsätzlich sind die weiß hinterlegten Felder durch den Benutzer auszufüllen. Sämtliche grau hinterlegten Felder werden automatisch berechnet.

In der Eingabemaske für die strategische Zielpriorität werden die für den Entwicklungsprozess geltenden internen und inhärenten Ziele angelegt und gegeneinander gewichtet. Der Editor berechnet mit Hilfe der AHP-Methode die strategische Relevanz jedes Ziels.

In der Eingabemaske für die technologische Zielpriorität werden die Module des zu entwickelnden Systems angelegt. Ihre Eigenschaften werden anhand von Dynamikindex und Komplexität spezifiziert, woraus der Editor die technologische Bedeutung jedes

Moduls berechnet. Wenn der Benutzer zudem die Korrelation zwischen den externen und inhärenten Zielen und der technologischen Bedeutung in der technologieorientierten Einflussmatrix im linken Bereich der Eingabemaske eingetragen hat, berechnet der Editor aus technologischer Bedeutung und technologieorientierter Einflussmatrix die technologische Zielpriorität für jedes Ziel über alle Module.

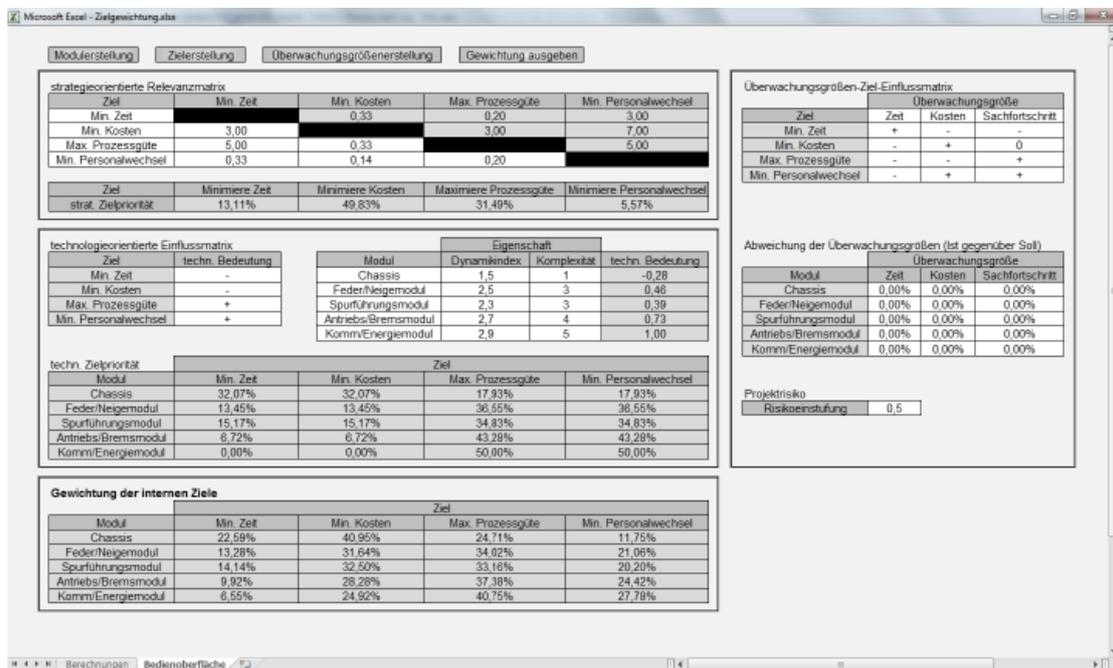


Bild 5-4 Bedienoberfläche des Excel-Editors für die Bestimmung der Gewichtung der Entwicklungsprozessziele

Die rechtsgelegene Eingabemaske dient der klassischen Projektüberwachung. Mit Hilfe der Überwachungsgrößen-Ziel-Einflussmatrix werden die Überwachungsgrößen des Entwicklungsprozesses spezifiziert und deren Einfluss auf die externen und inhärenten Ziele festgelegt. Mit Hilfe der unteren Tabelle der Eingabemaske werden die aktuellen Abweichungen der Überwachungsgrößen von deren Soll-Werten in Prozent eingegeben. Übersteigt der Ist- den Soll-Wert wird dies als positiver Wert eingetragen. Liegt der Ist-hinter dem Soll-Wert wird entsprechend ein negativer Wert eingetragen. Anhand der Inhalte der Überwachungsgrößen-Ziel-Einflussmatrix und den Abweichungen der Überwachungsgrößen zwischen Ist- und Soll-Wert errechnet der Editor den Zielgewichtungsänderungsbedarf je Ziel und Modul. Die Risikoklasse des Entwicklungsprozesses wird mit Hilfe des Wertes für die Risikoeinstufung spezifiziert.

Aus der strategischen und technologischen Zielpriorität sowie dem Zielgewichtungsänderungsbedarf und der Risikoeinstufung werden in der Ausgabemaske im unteren Teil des Editors die Gewichtungen der internen Ziele je Module ermittelt. Hierfür kommt standardmäßig die in Kapitel 4.3.2.2 beschriebene lineare Gewichtungsfunktion zum Einsatz. Abweichend hiervon ist es zudem möglich Exponential- oder Stufenfunktionen zu hinterlegen. Die Gewichtungsfunktionen können bei Bedarf für jedes Modul unter-

schiedlich gewählt werden. Die Gewichtungen der internen Ziele je Modul sind die Eingangswerte für den nachfolgend beschriebenen zweiten Excel-Editor.

Excel-Editor für die Auswahl des zielkonformen Plans

Der Excel-Editor für die Auswahl eines entwicklungszielkonformen Entwicklungsprozessplans ist in Bild 5-5 dargestellt. In die obere Maske werden die im vorstehend beschriebenen Excel-Editor bestimmten Gewichtungen der Entwicklungsprozessziele eingelesen. Zudem werden die für die internen Ziele relevanten Eigenschaften (Dauer, Kosten, Prozessgüte, Anzahl Personalwechsel) der im Prozessplaner aus Kapitel 5.1.1 generierten paretooptimalen Prozesspläne eingelesen. Für jede Eigenschaft werden die Minima und Maxima pro Modul bestimmt und auf das Intervall von 0 bis 100 Prozent normiert. Anschließend wird der euklidische Abstand jeder Eigenschaft zur Gewichtung des zugehörigen internen Ziels über alle Module berechnet. Die fünf besten Pläne werden, angefangen mit dem am besten geeigneten, mit ihren zugehörigen Abstandswerten in der unteren Ausgabemaske in Zahlen und als Spinnennetzgrafik dargestellt.

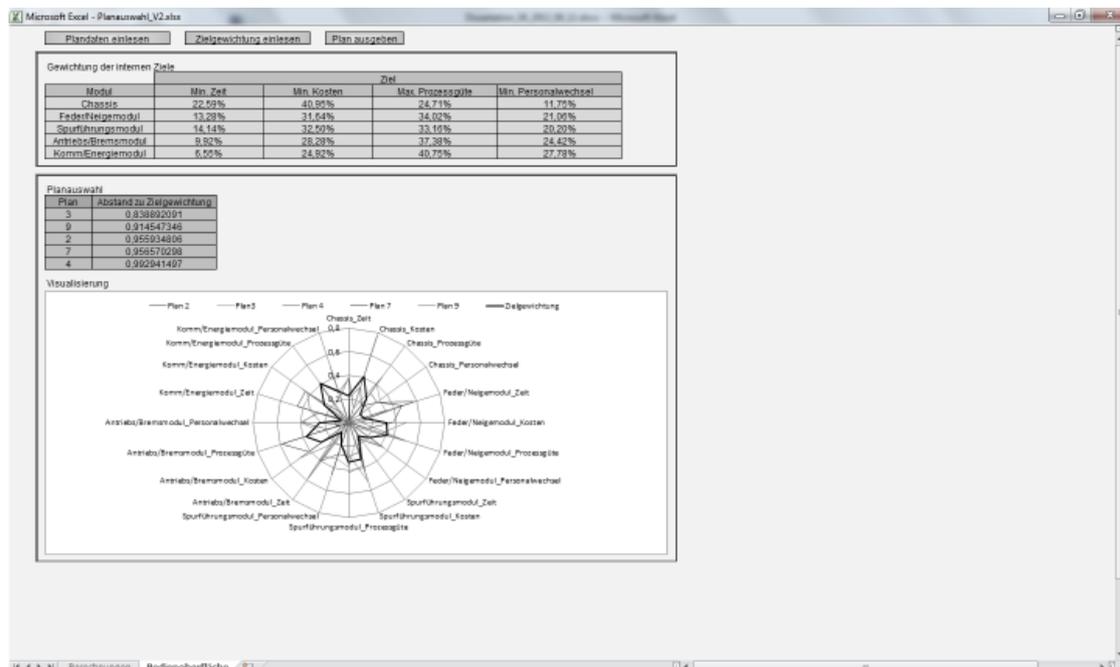


Bild 5-5 Bedienoberfläche des Excel-Editors für die Auswahl des zielkonformen Plans

5.2 Anwendungsbeispiel RailCab

Als Anwendungsbeispiel für die Verifikation des in dieser Arbeit entwickelten Rahmenwerks für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme dient das bereits in Kapitel 2.3.4 kurz vorgestellte innovative Bahnsystem RailCab. RailCab wird im Rahmen des Projekts „Neue Bahntechnik Paderborn“ entwickelt und ist einer der wesentlichen Demonstratoren des SFB 614 „Selbstoptimie-

rende Systeme des Maschinenbaus“. Ziel ist ein neuartiges, schienengebundenes Verkehrssystem für den Personen- und Gütertransport, das die bestehenden Trassen des Schienenverkehrs für den Individualverkehr nutzbar macht. Zu diesem Zweck werden keine fest miteinander gekoppelten Züge eingesetzt, sondern kleine autonome Fahrzeuge, sogenannte RailCabs (siehe Bild 5-6 links), die bedarfsgerecht und ohne Zwischenstopp vom Start- zum Zielpunkt fahren. Um energiebilanztechnisch nicht schlechter als fest gekoppelte Züge abzuschneiden, bilden die RailCabs auf längeren Streckenabschnitten Konvois zur Ausnutzung des Windschatteneffekts. Der Break-even-Point ist hier bereits ab einer Anzahl von vier RailCabs pro Konvoi erreicht. [Trä06], [ADG+09, S. 29ff.]

Der hohe Innovationsgrad des RailCabs spiegelt sich im Wesentlichen in zwei Punkten wider. Zum einen ist das System dezentral organisiert und weist einen hohen Grad an Autonomie und Selbstoptimierungsfähigkeit auf. Hierdurch kann es flexibel auf unterschiedliche Transportaufgaben unter verschiedenen Umfeldbedingungen reagieren. Zum zweiten ist das RailCab in Anlehnung an Systeme der Automobiltechnik modular aufgebaut. Die einzelnen Module sind aus mechanischer und informationstechnischer Sicht über standardisierte Schnittstellen miteinander verbunden. Hierdurch werden Kostenvorteile geschaffen und das RailCab kann bei Bedarf effizient für neue Transportaufgaben erweitert werden. Die Funktionsmodule des RailCabs sind in Bild 5-6 auf der rechten Seite dargestellt. [Trä06]

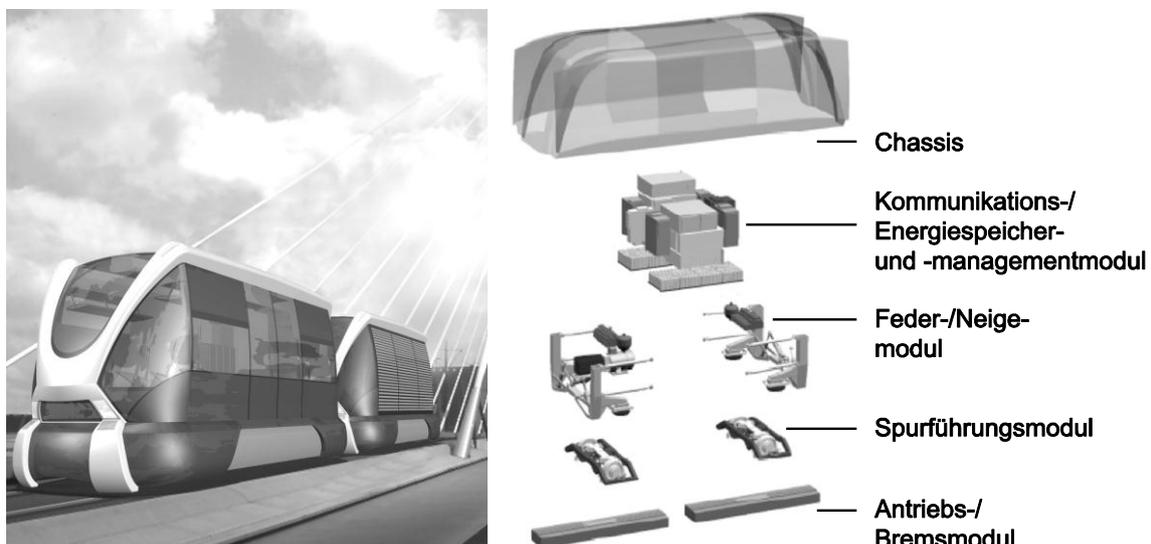


Bild 5-6 Autonomes Schienenfahrzeug RailCab: Design-Studie (links) und modularer Aufbau mit Funktionsmodulen (rechts) (in Anlehnung an [Trä06])

Das **Antriebs-/Bremsmodul** basiert auf einem doppelt gespeisten Linearmotor, bei dem sich ein Läufer vorne und einer hinten im Fahrzeug befindet und der Stator in der Mitte des Schienenbetts verlegt ist. Der Antrieb kann über- oder untersynchron betrieben werden, wodurch die Energieübertragung ins Fahrzeug ohne zusätzliche Oberleitungen entlang der Strecke ermöglicht wird. Zudem erlaubt es die individuelle Regelung des An-

triebs für jedes RailCab, dass zwei RailCabs mit unterschiedlichen Geschwindigkeiten auf demselben Statorabschnitt fahren können. Dies ist u.a. für die Konvoibildung notwendig [Trä06]. Da der Wirkungsgrad des Antriebs mit zunehmendem Abstand zwischen Läufer und Stator signifikant schlechter wird, ist für das Antriebs-/Bremsmodul eine aktive, elektromechanische Luftspaltverstellung (sogenanntes Air Gap Adjustment System – AGAS) vorgesehen, die den Abstand zwischen Läufer und Rotor durch Absenken oder Anheben des Läufers in Abhängigkeit vom Streckenverlauf variiert. [FGD+07, S. 935ff.]; [GZF+07, S. 1ff.]; [ADG+09, S. 45ff.]

Das RailCab verfügt über eine aktive Lenkung. Diese ist u.a. für die Konvoibildung und -auflösung bei voller Fahrt sowie auf Grund der dezentralen Organisationsstruktur des RailCab-Systems erforderlich. Sie ermöglicht ferner ein verschleißärmeres, leiseres Fahren sowie die Vermeidung des Spurkranzanlaufs bei Kurvenfahrten. Umgesetzt wird die aktive Lenkung durch das **Spurführungsmodul**. Es besteht aus je einer Losradachse hinten und vorne deren Räder ein zylindrisches Profil aufweisen. Der Lenkwinkel wird mit Hilfe von Hydraulikzylindern eingestellt. Neben dem Lenkwinkel kann zur Verbesserung der Führungskräfte zusätzlich der Radsturz individuell variiert werden. [Trä06]; [ADG+09, S. 54ff.]

Für die Realisierung eines hohen Fahrkomforts verfügt jedes RailCab über ein **Feder-/Neigemodul**, dessen Aufgabe es ist, die während der Fahrt auftretenden Vertikal- und Querkräfte in Abhängigkeit des Streckenverlaufs soweit wie möglich zu minimieren. Hierfür befinden sich vorne und hinten je ein aktives Federungssystem, die gemeinsam die Chassisbewegungen in Vertikal- und Querrichtung sowie ihren drei rotatorischen Freiheitsgraden bedämpfen. Ergänzt werden sie durch zwei aktive Neigesysteme, die gemeinsam die Seitenneigung des RailCabs bei Kurvenfahrten derart beeinflussen, dass höhere Kurvengeschwindigkeiten gefahren werden können. Federungs- und Neigesysteme sind jeweils hydraulisch aktuiert. [Trä06]

Je nach Streckenverlauf gibt es Betriebssituationen in denen nicht beliebig viel Leistung über den Antrieb ins Fahrzeug übertragen werden kann und somit der Leistungsbedarf des Bordnetzes nicht gedeckt werden kann. Ein Beispiel für eine solche Betriebssituation sind leichte Gefällstrecken, auf denen nur eine geringe Antriebs- bzw. Bremsleistung benötigt wird und damit auch die übertragene Leistung entsprechend gering ausfällt. Um dennoch jederzeit den Energiebedarf des RailCabs decken zu können, verfügt es über ein **Kommunikations-/Energiespeicher- und -managementmodul**. Dieses Modul hat sowohl eine große Speicherkapazität als auch eine hohe Leistungsdichte, was durch ein hybrides Energiespeichersystem bestehend aus einer Kombination aus Akkumulatoren und Doppelschichtkondensatoren ermöglicht wird. Es regelt zudem die Leistungsflüsse zu und von den übrigen Funktionsmodulen. Hierfür ist eine selbstoptimierende Betriebsstrategie vorgesehen, die basierend auf dem zu erwartenden Streckenprofil den optimalen Ladezustand des Energiespeichers und gemeinsam mit den übrigen Funktionsmodulen deren Soll-Betriebsmodi ermittelt. Das Streckenprofil wird durch Kommunikation mit an der Strecke verteilten Streckenabschnittsverwaltungen abgeru-

fen. Nach dem Überfahren eines Streckenabschnitts wird wiederum ein aktualisiertes Streckenprofil an der Streckenabschnittsverwaltung durch das RailCab hinterlegt. [ADG+09, S. 74ff.]

Das letzte Funktionsmodul des RailCab ist das **Chassis**. Es umfasst die Tragstruktur des RailCab, an der die übrigen Funktionsmodule befestigt werden, sowie den Fahrgastraum, der individuell für Personen und/oder Güter gestaltet werden kann.

Das RailCab ist bis dato als Prototyp im Maßstab 1:2,5 realisiert. Dieser Prototyp beinhaltet die vorstehend beschriebenen Funktionsmodule in ihrer Grundfunktionalität. Die Möglichkeit den Sturz der Spurführung zu verändern oder die aktive Verstellung des Luftspalts wurden hier noch nicht realisiert. Der Testbetrieb erfolgt auf einer 530m langen, ovalen Teststrecke. Die 1:1 Realisierung des Systems steht noch aus [Trä06]. Sie soll nachfolgend als Anwendungsbeispiel für das in der vorliegenden Arbeit entwickelte *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* dienen.

5.3 Durchführung der Verifikation und Ergebnisse

In diesem Kapitel werden die Durchführung der Verifikation des in der vorliegenden Arbeit entwickelten *Rahmenwerks für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* anhand des Beispiels RailCab und deren Ergebnisse beschrieben. Zunächst wird schrittweise die Erstellung des Verifikations-Setups entsprechend des Vorgehensmodells zur Initialisierung und Nachbereitung eines selbstoptimierenden Entwicklungsprozesses aus Kapitel 4.4 vorgenommen. Anschließend erfolgt die Verifikation des Selbstoptimierungsprozesses, wie er in Kapitel 4.2.2 beschrieben ist, auf Basis von vier verschiedenen Auslösern. Die Auslöser für den Selbstoptimierungsprozess sind: veränderte strategische Vorgaben, veränderte Eigenschaften des Entwicklungsgegenstandes, Veränderungen der Handlungsobjekte und Abweichungen des tatsächlichen Entwicklungsverlaufes gegenüber dem geplanten (vgl. Abschnitt 4.2.2). Die durch diese Auslöser angestoßenen Selbstoptimierungsprozesse werden den in Kapitel 4.2.2 beschriebenen erwarteten gegenübergestellt und bewertet.

5.3.1 Aufbau des Verifikations-Setups

Gemäß dem Vorgehensmodell zur Initialisierung und Nachbereitung eines selbstoptimierenden Entwicklungsprozesses aus Kapitel 4.4 sind vor der eigentlichen Ausführung des Entwicklungsprozesses die vier Phasen: „Wissensbasis initial befüllen“, „Wissensbasis aktualisieren“, „unternehmensspezifische Konfiguration vornehmen“ und „projektspezifische Konfiguration vornehmen“ zu durchlaufen. Die Ausführung des selbstoptimierenden Entwicklungsprozesses bildet die fünfte Phase.

Phase 1 und 2: Wissensbasis initial befüllen und Wissensbasis aktualisieren

Für die initiale Befüllung und Aktualisierung der Wissensbasis werden die im Anhang A3 aufgelisteten Prozessbausteine in dem in Kapitel 5.1.1 beschriebenen Prozessplaner hinterlegt. Zudem werden die möglichen personellen Ressourcen mit ihren jeweiligen Fähigkeiten modelliert, die die einzelnen Entwicklungstätigkeiten ausführen können. Auf die Modellierung spezifischer, ressourcenbezogener Aufwandschätzfunktionen wird verzichtet, da hierfür keine empirischen Daten in ausreichender Qualität vorhanden sind.

Zur Beschreibung der Grundbausteine, der modulinternen Abstimmungsbausteine sowie der modulübergreifenden Abstimmungsbausteine werden in Summe 153 Prozessbausteine modelliert, aus denen der Entwicklungsprozess des RailCab gebildet werden kann. Die durch die Prozessbausteine abgebildeten Fachdisziplinen sind: Maschinenbau, Elektrik, Elektronik, Regelungstechnik und Softwaretechnik. Für die Disziplin Maschinenbau werden zudem Prozessbausteine für die beiden Spezialisierungen Hydraulik und Pneumatik spezifiziert.

Phase 3: Unternehmensspezifische Konfiguration vornehmen

Als „unternehmensspezifische“ Konfiguration des selbstoptimierenden Entwicklungsprozesses werden für das Projekt „Neue Bahntechnik Paderborn“ die nachfolgenden Rahmenbedingungen angenommen. Die potentiell im Entwicklungsprozess anzuwendenden Ziele sind:

- **Minimiere Kosten (externes Ziel):** Eine hohe Gewichtung dieses Ziels bedeutet, dass die entwicklungsbezogenen Kosten für das RailCab möglichst gering gehalten werden sollen.
- **Minimiere Zeit (externes Ziel):** Eine hohe Gewichtung des Ziels „Minimiere Zeit“ bedeutet, dass die Entwicklung des RailCabs möglichst schnell erfolgen soll.
- **Maximiere Prozessgüte (externes Ziel):** Bei einer hohen Gewichtung dieses Ziel soll die Entwicklung des RailCabs mit möglichst hoher Prozessgüte erfolgen, um hierdurch indirekt eine möglichst hohe Produktgüte zu erreichen.
- **Minimiere Anzahl Personalwechsel (inhärentes Ziel):** Bei hoher Gewichtung des Ziels „Minimiere Anzahl Personalwechsel“ sollen nach Möglichkeit immer die gleichen personellen Ressourcen an einem Modul des RailCabs arbeiten. Häufige Wechsel zwischen verschiedenen Modulen sind nicht erwünscht.

Diese Ziele können grundsätzlich in Abhängigkeit von der vorliegenden Entwicklungssituation bei der Entwicklung des RailCab herangezogen und gegeneinander gewichtet werden. Die projektmanagementbezogene Erfassung der Entwicklungssituation kann mit Hilfe der drei nachfolgend definierten Überwachungsgrößen vorgenommen werden:

- **Zeit:** Die Überwachungsgröße Zeit beschreibt die seit Projektbeginn verstrichene Realzeit. Sie bezieht sich demnach nicht direkt auf den zeitlichen Aufwand für die Entwicklung eines Moduls, sondern auf eventuelle Terminüber- oder -unterschreitungen.
- **Kosten:** Die Kosten sind die kumulierten Aufwände für die Entwicklung eines Moduls.
- **Sachfortschritt:** Der Sachfortschritt beschreibt, ob den für die Entwicklung eines Moduls bisher aufgewendeten Kosten eine adäquate Leistung gegenübersteht.

Wie die potentiellen Entwicklungsziele und Überwachungsgrößen miteinander und mit den übrigen Rahmenbedingungen projektspezifisch zusammenspielen wird in der nachfolgenden Phase definiert.

Phase 4: Projektbezogene Konfiguration vornehmen

Für die Verifikation des in der vorliegenden Arbeit entwickelten Rahmenwerks sollen sämtliche in der vorstehenden Phase definierten Ziele und Überwachungsgrößen verwendet werden. Es wird davon ausgegangen, dass die weitere Entwicklung des RailCabs aufbauend auf dem bisherigen Entwicklungsstand einen Zeitrahmen von vier Jahren nicht überschreiten soll. Ferner sollen fünfzehn Entwickler mit unterschiedlichem Qualifikationsniveau durchgängig für die Entwicklung zur Verfügung stehen. Aus beiden Annahmen ergibt sich entsprechend das maximale Entwicklungsbudget⁴⁸. Das Projektrisiko wird als mittel eingestuft.

Neben den Projektrahmenbedingungen wird das Systemmodell des RailCabs im Reifegrad der Prinziplösung als Eingangsgröße für den selbstoptimierenden Entwicklungsprozess verwendet. Bild 5-7 zeigt einen vereinfachten Ausschnitt des Systemmodells, der entsprechend der in Kapitel 4.3.1.1 entwickelten Modellierungstechnik neben der grundlegenden Struktur auch eine Beschreibung der Charakteristika der einzelnen Module beinhaltet. Beschrieben sind die Klasse eines Moduls, die Fachdisziplinen, zu denen es gehört, sowie deren Komplexität und Dynamikindex.

Anhand der Zuordnung zu einer Klasse von mechatronischen Systemen und zu einer oder mehreren Fachdisziplinen sowie den zwischen den Modulen bestehenden Verbindungen kann mit Hilfe der in Phase 1 für den Prozessplaner definierten Prozessbausteine die logische Vorgangsfolge zur Entwicklung des RailCabs gebildet werden (siehe auch Abschnitt 4.3.2.1 und Abschnitt 5.1.1). Der entstehende Vorgangsfolgeplan enthält insgesamt 389 Prozessschritte. Da in Phase 1 keine spezifischen, ressourcenbezogenen Aufwandschätzfunktionen hinterlegt wurden, wird im Nachfolgenden mit fiktiven Plan-

⁴⁸ Die genannten Zahlen sind fiktive Werte, die lediglich für die hier beschriebene Verifikation herangezogen werden. Ggf. vorhandene Übereinstimmungen mit den realen Entwicklungsaufwänden für das RailCab im Projekt „Neue Bahntechnik Paderborn“ sind allenfalls zufällig und entziehen sich der Kenntnis des Autors.

Anhand der im Systemmodell spezifizierten Daten zu Komplexität und Dynamikindex der einzelnen Module wird mit Hilfe des in Kapitel 5.1.2 beschriebenen Excel-Editors für die Bestimmung der Gewichtung der Entwicklungsprozessziele deren jeweilige technologische Bedeutung berechnet (siehe auch Abschnitt 4.3.2.2). Die Ergebnisse sind im oberen Abschnitt in Bild 5-8 dargestellt. Zusammen mit der technologieorientierten Einflussmatrix wird aus der technologischen Bedeutung eines jeden Moduls die technologische Zielpriorität eines jeden Ziels pro Modul bestimmt. Die technologieorientierte Einflussmatrix des RailCabs zeigt, dass im Falle einer hohen technologischen Bedeutung eines Moduls für dieses Modul die Ziele „Minimiere Zeit“ und „Minimiere Kosten“ weniger hoch zu gewichten sind als bei den anderen Modulen. Bei den beiden Zielen „Maximiere Prozessgüte“ und „Minimiere Personalwechsel“ ist es genau umgekehrt. Die gewählte Konfiguration der technologieorientierten Einflussmatrix trägt somit dem Anspruch Rechnung bei komplexen bzw. für das RailCab technologisch bedeutsamen Modulen mehr auf Qualität als auf Zeit und Kosten zu achten. Demzufolge sind bei dem aus technologischer Sicht am anspruchsvollsten Kommunikations-/Energiespeicher- und -managementmodul die Ziele „Maximiere Prozessgüte“ und „Minimieren Personalwechsel“ mit jeweils 50% gewichtet, während die Ziele „Minimiere Zeit“ und „Minimiere Kosten“ mit 0% gewichtet sind. Beim technisch am wenigsten anspruchsvollen Chassis sind hingegen „Minimiere Zeit“ und „Minimiere Kosten“ mit jeweils 32% und „Maximiere Prozessgüte“ und „Minimieren Personalwechsel“ mit nur je 18% gewichtet.

	Chassis	Feder/Neigemodul	Spurführungsmodul	Antriebs/Bremsmodul	Komm/Energiemodul
Komplexität [1 - 5]	1	3	3	4	5
Dynamikindex [1 - 3]	1,5	2,5	2,3	2,7	2,9
technologische Bedeutung [-]	-0,28	0,46	0,39	0,73	1,00

technologieorientierte Einflussmatrix

Ziel	Min. Zeit	Min. Kosten	Max. Prozessgüte	Min. Personalwechsel
technologische Bedeutung	-	-	+	+

Bedeutung:
+ positive Korrelation
- negative Korrelation



technologische Zielpriorität [%]

Modul \ Ziel	Min. Zeit	Min. Kosten	Max. Prozessgüte	Min. Personalwechsel
Chassis	32%	32%	18%	18%
Feder/Neigemodul	13%	13%	37%	37%
Spurführungsmodul	15%	15%	35%	35%
Antriebs/Bremsmodul	7%	7%	43%	43%
Komm/Energiemodul	0%	0%	50%	50%

Bild 5-8 technologieorientierte Zielpriorität des RailCabs

Neben der technologischen Zielpriorität wird mit Hilfe der im nachfolgenden Bild 5-9 dargestellten strategieorientierten Relevanz-Matrix die strategische Zielpriorität der einzelnen Ziele mit Hilfe des Excel-Editors bestimmt (siehe auch Abschnitt 4.3.2.2 und Abschnitt 5.1.2). Die Matrix stellt die Ziele einander gegenüber und bewertet entsprechend dem rechts dargestellten Bewertungsmaßstab wie die strategische Bedeutung eines Ziels in einer Zeile zu dem in der Spalte ist. Insgesamt wird deutlich, dass aus

strategischer Sicht bei der Entwicklung des RailCabs die Minimierung der Kosten und die Maximierung der Prozessgüte im Vordergrund stehen sollen. Die Minimierung der Zeit und der Personalwechsel sind hingegen weniger von Bedeutung.

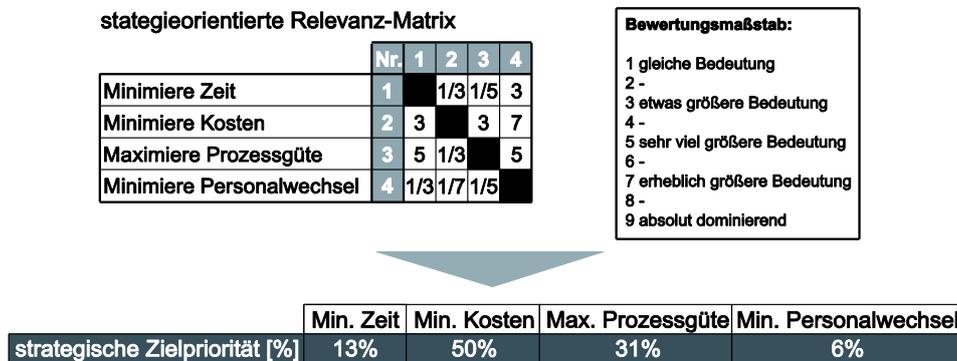
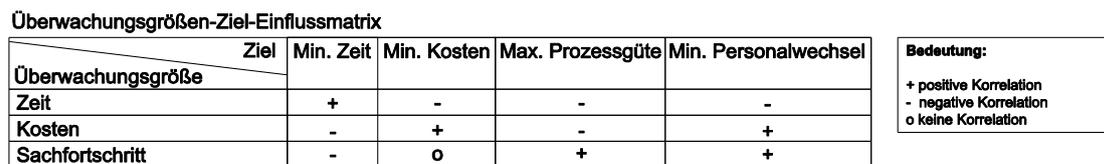


Bild 5-9 Strategieorientierte Relevanz-Matrix des RailCabs

Als drittes wird für die Nachverfolgung und Bewertung des Entwicklungsverlaufs des RailCabs festgelegt wie sich eine Über- bzw. Unterschreitung der Überwachungsgrößen auf die Gewichtung der Entwicklungsziele auswirken soll. Die Ergebnisse zeigt Bild 5-10.



Abweichung der Überwachungsgrößen (Ist gegenüber Soll) [%]

Überwachungsgröße \ Modul	Chassis	Feder/Neigemodul	Spurführungsmodul	Antriebs/Bremsmodul	Komm/Energiemodul
Zeit	0%	0%	0%	0%	0%
Kosten	0%	0%	0%	0%	0%
Sachfortschritt	0%	0%	0%	0%	0%

Änderungsbedarfs-Matrix

Modul \ Ziel	Min. Zeit	Min. Kosten	Max. Prozessgüte	Min. Personalwechsel
Chassis	0%	0%	0%	0%
Feder/Neigemodul	0%	0%	0%	0%
Spurführungsmodul	0%	0%	0%	0%
Antriebs/Bremsmodul	0%	0%	0%	0%
Komm/Energiemodul	0%	0%	0%	0%

Bild 5-10 Überwachungsgrößen und Änderungsbedarfs-Matrix des RailCabs

Die Überwachungsgrößen-Ziel-Einflussmatrix legt fest wie sich ein Über- oder Unterschreiten der Überwachungsgrößen bezogen auf den Ursprungsplan auf die Gewichtung der Entwicklungsziele auswirken soll (siehe auch Abschnitt 4.3.2.2). Für das RailCab gilt, dass im Falle einer Überschreitung der Zeit, wenn also ein Modul länger braucht

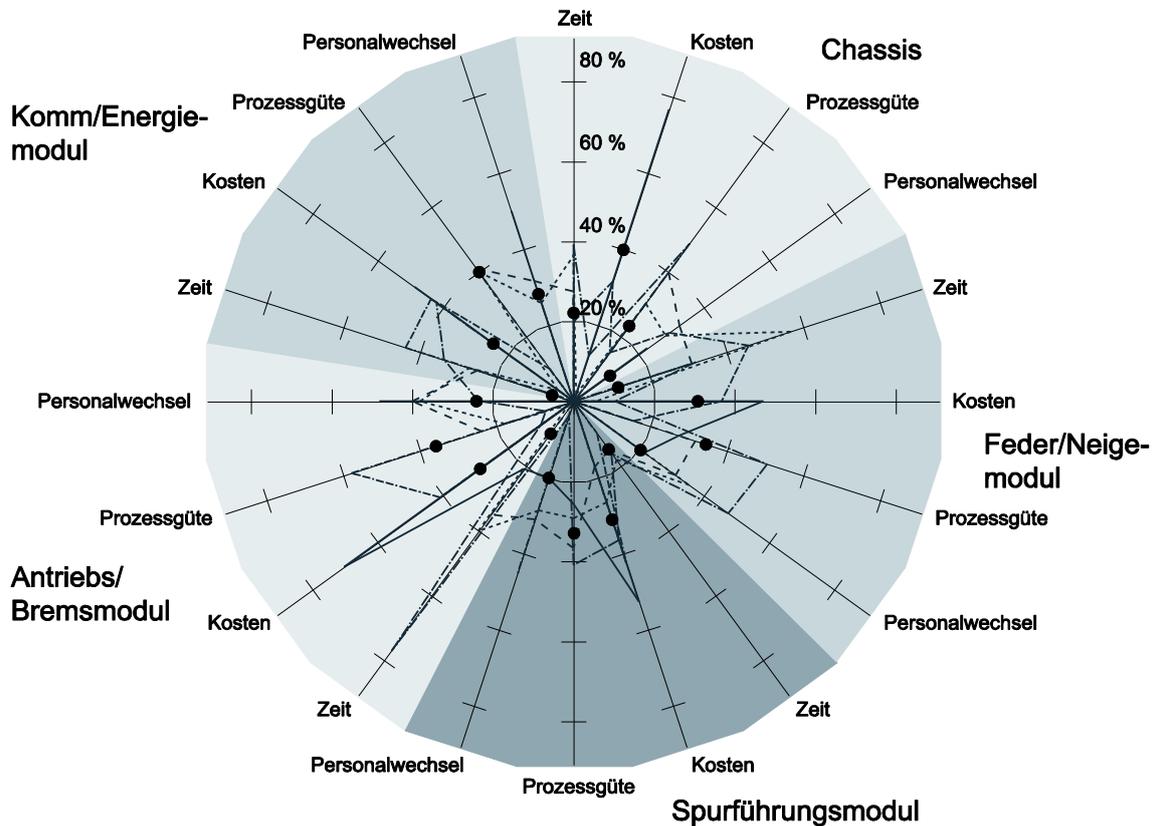
als geplant, das Ziel „Minimiere Zeit“ für dieses Modul höher gewichtet wird und die Ziele „Minimiere Kosten“, „Maximiere Prozessgüte“ und „Minimiere Personalwechsel“ weniger hoch. Der Zeitverlust soll also auf Kosten höherer Entwicklungsaufwände und geringerer Qualität wieder aufgeholt werden. Bei einer Kostenüberschreitung wird durch die gewählte Korrelation versucht mit dem verbleibenden Budget und den bereits eingearbeiteten Ressourcen auf Kosten einer geringeren Qualität und etwas längerer Entwicklungsdauer die Entwicklung möglichst effizient zu Ende zu bringen. Ist ein Modul innerhalb des Zeit- und Kostenrahmens weiter entwickelt als erwartet, sollen die dadurch frei gewordenen zeitlichen und kostenmäßigen Ressourcen zur Verbesserung von dessen Qualität verwendet werden.

Für das Verifikations-Setup wird zunächst davon ausgegangen, dass bis dato keine Planabweichungen bei der Entwicklung der Prinziplösung des RailCabs aufgetreten sind. Entsprechend ergibt sich kein aktueller Änderungsbedarf für die Gewichtung der einzelnen Ziele.

Zu guter Letzt wird mit Hilfe der in Kapitel 4.3.2.2 beschriebenen Gewichtungsfunktion aus technologischer und strategischer Zielpriorität sowie dem Zieländerungsbedarf das interne Zielsystem des Entwicklungsprozesses des RailCabs berechnet. Das interne Zielsystem ist im oberen Teil in Bild 5-11 dargestellt. Mit dem Excel-Editor für die Auswahl des zielkonformen Plans wird aus den fiktiven paretooptimalen Plänen der am besten für die aktuelle Entwicklungssituation geeignete ausgewählt (siehe auch Abschnitt 4.3.2.3 und Abschnitt 5.1.2). Die modulweisen Eigenschaften der Pläne und die Gewichtungen der einzelnen Entwicklungsziele sind im Spinnennetzdiagramm unterhalb des internen Zielsystems aufgeführt. Unterhalb des Spinnennetzdiagramms sind die fünf, den im internen Zielsystem gewünschten Zielgewichtungen am nächsten kommenden Pläne und ihre Rangfolge dargestellt. Für die aktuelle Entwicklungssituation ist demnach Plan 3 der am besten geeignete. Er zeichnet sich insbesondere durch eine hohe Prozessgüte aus.

gewichtetes internes Zielsystem

Modul \ Ziel	Min. Zeit	Min. Kosten	Max. Prozessgüte	Min. Personalwechsel
Chassis	22,59%	40,95%	24,71%	11,75%
Feder/Neigemodul	13,28%	31,64%	34,02%	21,06%
Spurführungsmodul	14,14%	32,50%	33,16%	20,20%
Antriebs/Bremsmodul	9,92%	28,28%	37,38%	24,42%
Komm/Energiemodul	6,55%	24,92%	40,75%	27,78%



Plan	Plan 2	Plan 3	Plan 4	Plan 7	Plan 9	Zielgewichtung
Linienart	-----	-----	-----	-----	-----	●
Abstand zur Zielgewichtung	0,955935	0,838892	0,992941	0,95657	0,914547	-
Rang	3	1	5	4	2	-

Bild 5-11 Ausgangssituation für die Entwicklung des RailCabs

5.3.2 Verifikation des Selbstoptimierungsprozesses

Für die Verifikation des Selbstoptimierungsprozesses, wie er in Kapitel 4.2.2 beschrieben ist, wird dieser nachfolgend auf vier verschiedene Arten ausgelöst. Die Auslöser sind: veränderte strategische Vorgaben, veränderte Eigenschaften des Entwicklungsgegenstandes, Veränderungen der Handlungsobjekte und Abweichungen des tatsächlichen Entwicklungsverlaufes gegenüber dem geplanten. Die Auslöser werden als während der Ausführung des Entwicklungsprozesses auftretend angenommen. Es werden jeweils die daraus gegenüber dem im vorstehend genannten Kapitel 5.3.1 generierten internen Zielsystem und dem zugehörigen optimalen Plan entstehenden Unterschiede beschrieben und bewertet.

Verifikations-Szenario 1: Veränderung der strategischen Vorgaben

Im ersten Verifikations-Szenario wird davon ausgegangen, dass zu Beginn der Weiterentwicklung des RailCabs nicht die im vorstehenden Kapitel 5.3.1 beschriebenen strategischen Vorgaben gemacht werden, sondern die im nachfolgenden Bild 5-12 beschriebenen. Die Änderungen gegenüber den ursprünglichen Vorgaben sind in der strategieorientierten Relevanz-Matrix hellgrau hinterlegt und fett. Im Unterschied zu der vorherigen qualitäts- und kostenorientierten Strategie wird jetzt eine Strategie bevorzugt, die eine Minimierung des zeitlichen Aufwands favorisiert. Entsprechend ändern sich die strategischen Zielprioritäten der einzelnen Ziele.



Bild 5-12 geänderte strategische Vorgaben für die Entwicklung des RailCabs

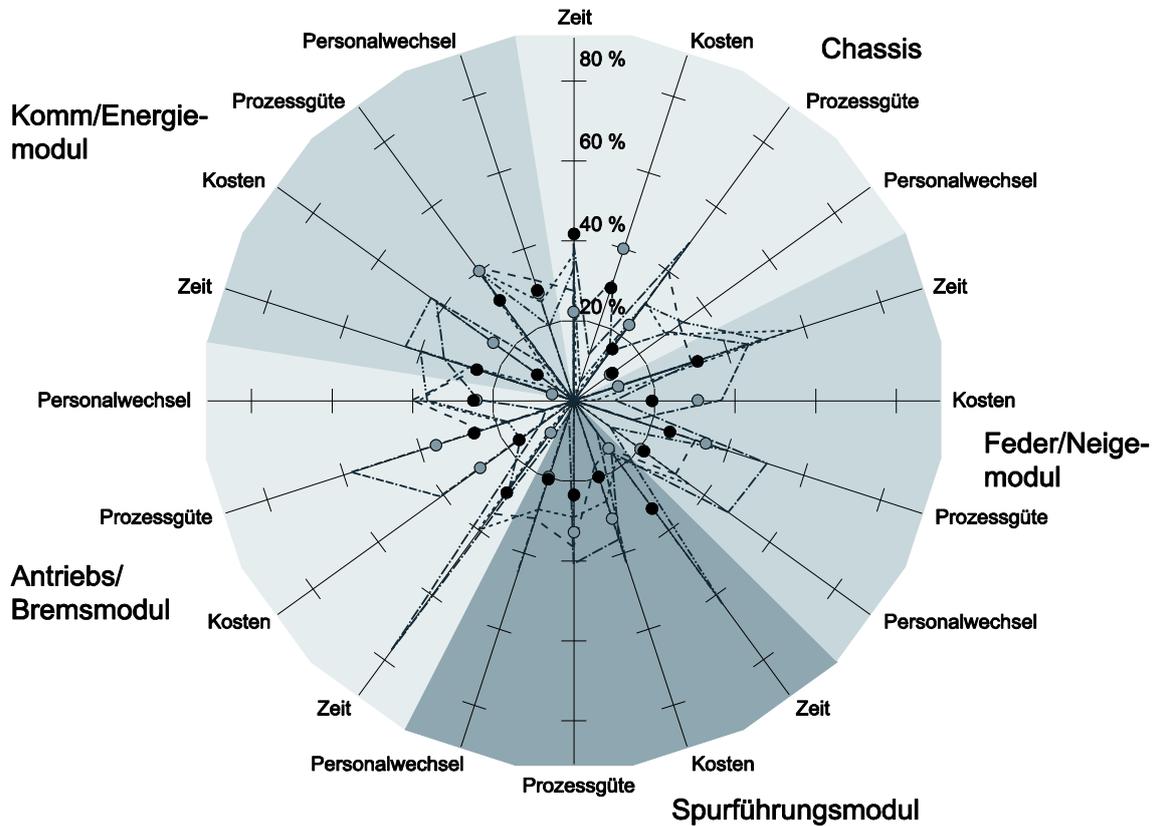
Die technologieorientierten Zielprioritäten und die einzelnen Prozesspläne bleiben von der Änderung der strategischen Vorgaben unberührt. An den technologieorientierten Zielprioritäten ändert sich nichts, da die Struktur und die technischen Charakteristika des RailCabs gleich geblieben sind. Der Entwicklungsgegenstand bleibt also unverändert. Die Prozesspläne bleiben unverändert, da weder die Struktur des RailCabs noch die verfügbaren personellen Ressourcen verändert wurden.

Das neue interne Zielsystem des Entwicklungsprozesses ist im nachfolgenden Bild 5-13 zu sehen. Im Spinnennetzdiagramm wird deutlich, dass die Gewichtungen der internen Ziele (schwarz ausgefüllte Kreise) über alle Module zu Gunsten einer höheren Gewichtung des Ziels „Minimiere Zeit“ gegenüber den alten Gewichtungen (grau ausgefüllte Kreise) verschoben wurden. Die paretooptimalen Prozesspläne bleiben unverändert. Für die neue Gewichtung der internen Ziele ist nun nicht mehr Plan drei der am besten geeignete, sondern Plan zwei. Dieser ist bezüglich Zeit und Kosten besser als Plan drei bei schlechterer Prozessgüte und Anzahl an Personalwechseln. Zudem gehört Plan neun nicht mehr zu den fünf am besten geeigneten Plänen. An seine Stelle tritt Plan eins.

Bezüglich dem nach Kapitel 4.2.2 zu erwartenden Ablauf des Selbstoptimierungsprozesses und seinem Ergebnis ist festzustellen, dass die in diesem Verifikations-Szenario erzielten Ergebnisse mit den Erwartungen übereinstimmen. Das interne Zielsystem wurde bei gleichzeitig unveränderten Prozessplänen angepasst und es wurde auf einen dem neuen Zielsystem entsprechenden Plan gewechselt.

gewichtetes internes Zielsystem

Modul \ Ziel	Min. Zeit	Min. Kosten	Max. Prozessgüte	Min. Personalwechsel
Chassis	42,03%	28,95%	16,32%	12,70%
Feder/Neigemodul	32,72%	19,64%	25,63%	22,01%
Spurführungsmodul	33,59%	20,50%	24,76%	21,15%
Antriebs/Bremsmodul	29,36%	16,28%	28,99%	25,37%
Komm/Energiemodul	26,00%	12,92%	32,35%	28,73%



Plan	Plan 1	Plan 2	Plan 3	Plan 4	Plan 7	Zielgewichtung
Linienart	-----	-----	-----	-----	-----	●
Abstand zur Zielgewichtung	0,69788	0,6283	0,62967	0,83992	0,98994	-
Rang	3	1	2	4	5	-

Bild 5-13 Selbstoptimierung des Entwicklungsprozesses des RailCabs nach der Änderung der strategischen Vorgaben

Verifikations-Szenario 2: Veränderung der Eigenschaften des Entwicklungsgegenstandes

Für die Analyse des Selbstoptimierungsprozesses bei Veränderung der Eigenschaften des Entwicklungsgegenstandes wird davon ausgegangen, dass beim Spurführungsmodul des RailCabs auf die Möglichkeit dessen Sturz zu variieren verzichtet wird. Zudem soll das Antriebs-/Bremsmodul ohne eine aktive Verstellung des Abstandes zwischen Läufer und Stator realisiert werden. Entsprechend reduzieren sich Komplexität und Dynamikindex beider Module (vgl. Bild 5-14). Der Komplexitätsgrad des Spurführungsmoduls sinkt von drei auf zwei, der des Antriebs-/Bremsmoduls von vier auf drei. Beim Dynamikindex sinkt derjenige des Spurführungsmoduls von 2,3 auf 2,1 und derjenige

des Antriebs-/Bremsmoduls von 2,7 auf 2,3. Durch diese Änderungen sinken auch die technologischen Bedeutungen beider Module und damit deren technologische Zielprioritäten. Die strategischen Zielprioritäten bleiben unverändert.

	Chassis	Feder/Neigemodul	Spurführungsmodul	Antriebs/Bremsmodul	Komm/Energiemodul
Komplexität [1 - 5]	1	3	2	3	5
Dynamikindex [1 - 3]	1,5	2,5	2,1	2,3	2,9
technologische Bedeutung [-]	-0,28	0,46	0,12	0,46	1,00

technologieorientierte Einflussmatrix

Ziel	Min. Zeit	Min. Kosten	Max. Prozessgüte	Min. Personalwechsel	Bedeutung:
technologische Bedeutung	-	-	+	+	+ positive Korrelation - negative Korrelation



technologische Zielpriorität [%]

Modul \ Ziel	Min. Zeit	Min. Kosten	Max. Prozessgüte	Min. Personalwechsel
Chassis	32%	32%	18%	18%
Feder/Neigemodul	13%	13%	37%	37%
Spurführungsmodul	22%	22%	28%	28%
Antriebs/Bremsmodul	13%	13%	37%	37%
Komm/Energiemodul	0%	0%	50%	50%

Bild 5-14 geänderte technologische Zielpriorität auf Grund veränderter technischer Charakteristika der Module des RailCabs

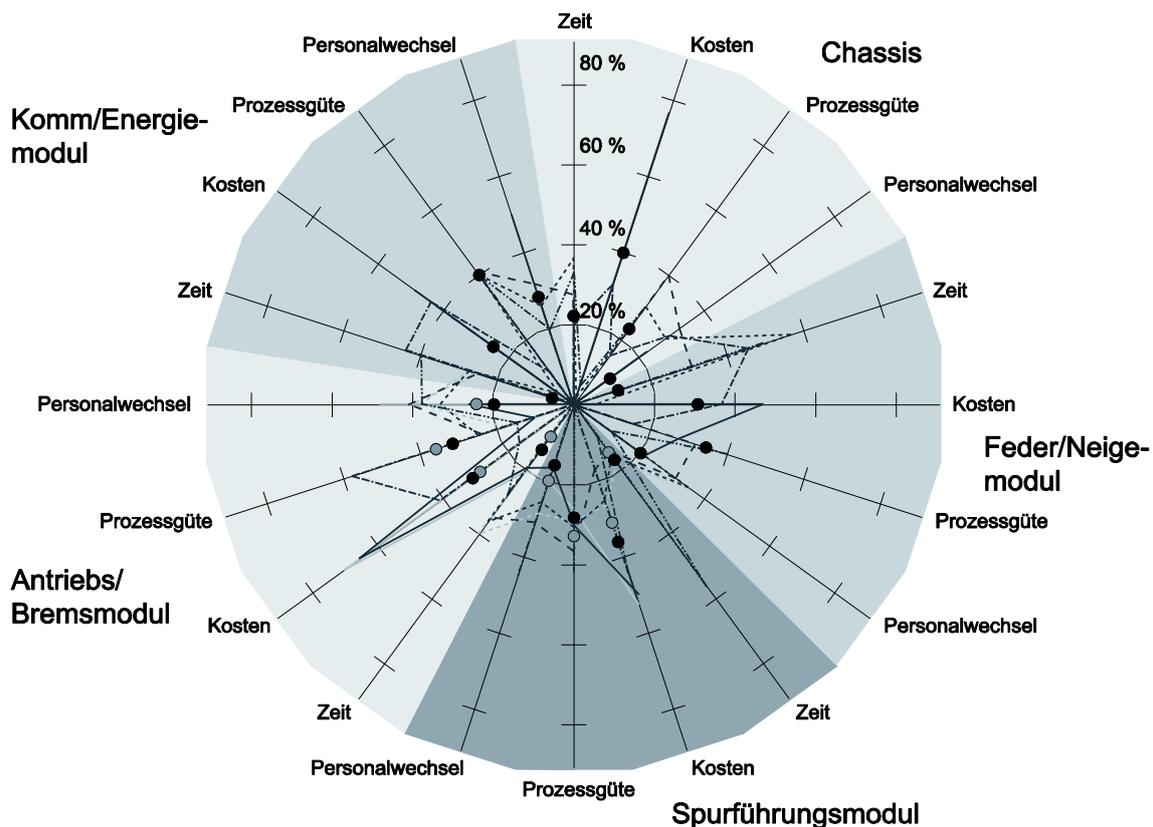
Die durch die Veränderung der technologischen Zielprioritäten des Spurführungsmoduls und des Antriebs-/Bremsmoduls hervorgerufenen Änderungen im internen Zielsystem des Entwicklungsprozesses des RailCab verdeutlicht Bild 5-15. Die Gewichtung der internen Ziele des Chassis, des Feder-/Neigemoduls sowie des Kommunikations-/Energiespeicher- und -managementmoduls bleiben unverändert. Die Gewichtungen der internen Ziele des Spurführungsmoduls und des Antriebs-/Bremsmoduls verschieben sich hingegen jeweils in Richtung einer höheren Gewichtung der Minimierung von Zeit und Kosten bei geringerer Gewichtung der Maximierung der Prozessgüte und der Minimierung der Anzahl der Personalwechsel. Diese Veränderungen des internen Zielsystems entsprechen den bei der Definition des Verifikations-Setups vorgenommenen Einträgen in der technologieorientierten Einflussmatrix. Einhergehend mit der Veränderung der technischen Charakteristika der beiden Module ändern sich auch die modulbezogenen Eigenschaften der paretooptimalen Prozesspläne. Die Veränderungen sind in der normierten Darstellung im Spinnennetzdiagramm in Bild 5-15 nicht so deutlich wie die Verschiebung der Zielprioritäten, da die Veränderungen der Pläne als annähernd gleichmäßig über die Ursprungspläne angenommen werden. Im Ergebnis ist für das neue interne Zielsystem jetzt Plan neun der am besten geeignete. Plan vier gehört nicht mehr zu den besten fünf und wird durch Plan eins ersetzt.

Bezüglich des nach Kapitel 4.2.2 zu erwartenden Ablaufs des Selbstoptimierungsprozesses und seinem Ergebnis ist für Verifikations-Szenario zwei festzustellen, dass die erzielten Ergebnisse ebenfalls mit den Erwartungen übereinstimmen. Die Veränderung der technischen Eigenschaften der einzelnen Module des RailCabs hat eine selektive Veränderung der technologischen Zielprioritäten dieser beiden Module bewirkt. Die

strategischen Zielprioritäten sind unverändert geblieben. In Folge dessen hat sich eine selektive Veränderung der internen Ziele des Spurführungsmoduls und des Antriebs-/Bremsmoduls ergeben, die zu einem Wechsel des Prozessplans geführt hat.

gewichtetes internes Zielsystem

Modul \ Ziel	Min. Zeit	Min. Kosten	Max. Prozessgüte	Min. Personalwechsel
Chassis	22,59%	40,95%	24,71%	11,75%
Feder/Neigemodul	13,28%	31,64%	34,02%	21,06%
Spurführungsmodul	17,50%	35,87%	29,80%	16,83%
Antriebs/Bremsmodul	14,14%	32,50%	33,16%	20,20%
Komm/Energiemodul	6,55%	24,92%	40,75%	27,78%



Plan	Plan 1	Plan 2	Plan 3	Plan 7	Plan 9	Zielgewichtung
Linienart	-----	-----	-----	-----	-----	●
Abstand zur Zielgewichtung	1,02621	0,9346	0,8673	0,95442	0,83483	-
Rang	5	3	2	4	1	-

Bild 5-15 Selbstoptimierung des Entwicklungsprozesses des RailCabs nach der Änderung der technischen Charakteristika der Module des RailCabs

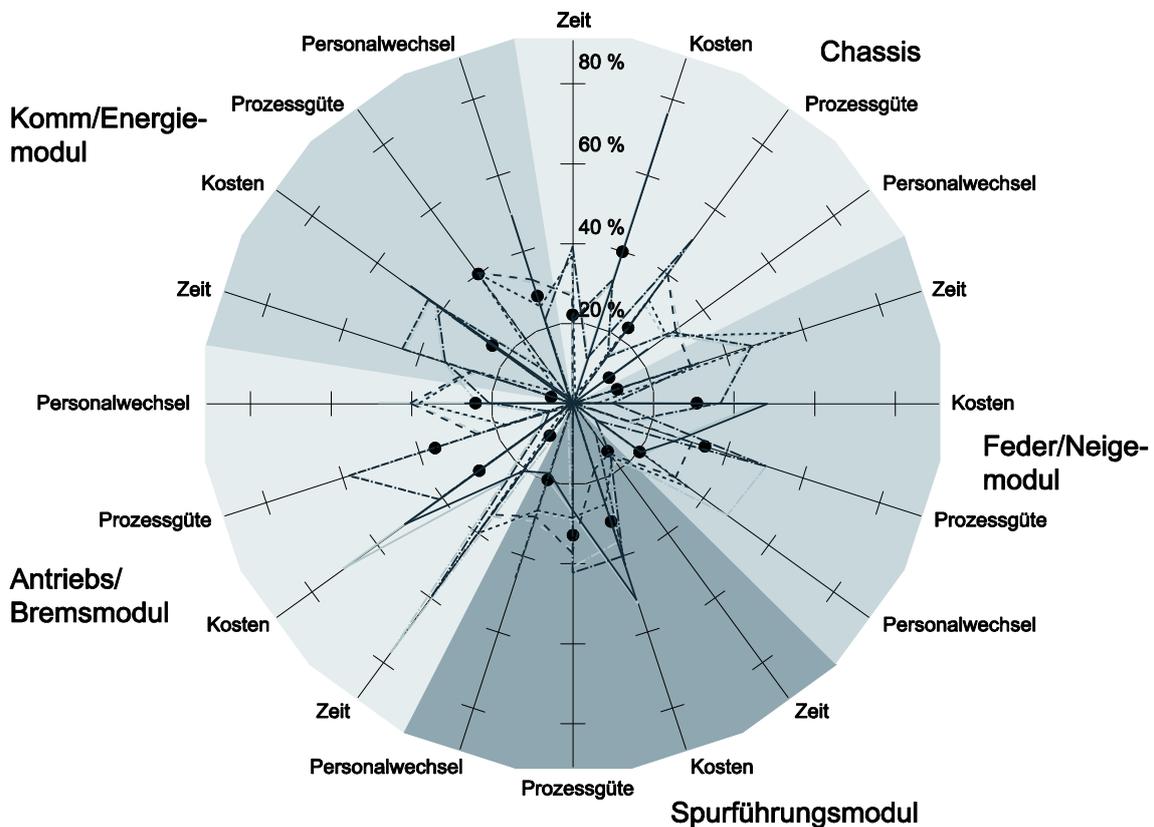
Verifikations-Szenario 3: Veränderung der Handlungsobjekte

Im Verifikations-Szenario drei wird davon ausgegangen, dass weniger personelle Ressourcen für die Weiterentwicklung des RailCabs zur Verfügung stehen als die angestrebten 15 Entwickler. Infolge dessen erhöht sich die Dauer über alle möglichen Prozesspläne annähernd homogen bei annähernd gleichbleibender Kostenstruktur, Prozessgüte und Anzahl an Personalwechseln.

Eine Veränderung des internen Zielsystem findet nicht statt, da sich weder die strategischen Vorgaben für die Entwicklung noch die technischen Charakteristika des RailCabs verändern. Im Ergebnis stellt sich die Selbstoptimierung des Entwicklungsprozesses wie in Bild 5-16 gezeigt dar. Plan drei ist nach wie vor der am besten geeignete Plan für die Entwicklung des RailCabs. Die gleichmäßige Veränderung der paretooptimalen Prozesspläne führt zwar zu einer Veränderung von deren Abständen zum internen Zielsystem, diese ist aber auch gleichmäßig, wodurch kein Wechsel zu einem anderen Prozessplan bewirkt wird.

gewichtetes internes Zielsystem

Modul \ Ziel	Min. Zeit	Min. Kosten	Max. Prozessgüte	Min. Personalwechsel
Chassis	22,59%	40,95%	24,71%	11,75%
Feder/Neigemodul	13,28%	31,64%	34,02%	21,06%
Spurführungsmodul	14,14%	32,50%	33,16%	20,20%
Antriebs/Bremsmodul	9,92%	28,28%	37,38%	24,42%
Komm/Energiemodul	6,55%	24,92%	40,75%	27,78%



Plan	Plan 2	Plan 3	Plan 4	Plan 7	Plan 9	Zielgewichtung
Linienart	-----	----	-----	-----	-----	●
Abstand zur Zielgewichtung	0,95609	0,83802	1,00558	0,95652	0,91865	-
Rang	3	1	5	4	2	-

Bild 5-16 Selbstoptimierung des Entwicklungsprozesses des RailCabs nach der Änderung der Anzahl der verfügbaren personellen Ressourcen

Auch wenn kein Wechsel des Prozessplans vorgenommen wird entsprechen die Ergebnisse des Verifikations-Szenarios 3 dem Selbstoptimierungsprozess in Kapitel 4.2.2.

Die Veränderung der Handlungsobjekte hat keine Veränderung des internen Zielsystems bewirkt. Dass die Veränderung des Handlungssystem keine besser geeigneten Prozesspläne und damit einen Planwechsel hervorruft, ist der gleichmäßigen Veränderung der Prozesspläne zuzuschreiben und für den Nachweis eines funktionierenden Selbstoptimierungsprozesses nicht notwendig.

Verifikations-Szenario 4: Planabweichungen gegenüber dem ursprünglichen Prozessplan

Die für Verifikations-Szenario 4 angenommenen Abweichungen gegenüber dem ursprünglich geplanten Verlauf des Entwicklungsprozesses des RailCabs sind in der Mitte in Bild 5-17 dargestellt. Demnach konnte das Chassis kostengünstiger entwickelt werden als ursprünglich angenommenen. Die Entwicklungsobjekte des Feder-/Neigemoduls wurden zu einem etwas späteren Zeitpunkt als erwartet und zudem zu einem höheren Preis erstellt. Ihr Sachfortschritt hat dafür die Erwartungen übertroffen. Die Entwicklungsobjekte des Spurführungsmoduls liegen vor dem vereinbarten Termin mit einem geringeren Sachfortschrittsgrad und höheren Kosten vor. Das Antriebs-/Bremsmodul liegt bezüglich der geplanten Zeit und Kosten im Soll und hat sogar einen höheren Sachfortschrittsgrad als erwartet. Für die Entwicklung des Kommunikations-/Energiespeicher- und -managementmoduls wurden mehr Zeit und Budget verbraucht als geplant, um den angestrebten Sachfortschrittsgrad zu erreichen.

Überwachungsgrößen-Ziel-Einflussmatrix

Ziel	Min. Zeit	Min. Kosten	Max. Prozessgüte	Min. Personalwechsel
Überwachungsgröße				
Zeit	+	-	-	-
Kosten	-	+	-	+
Sachfortschritt	-	0	+	+

Bedeutung:
+ positive Korrelation
- negative Korrelation
o keine Korrelation

Abweichung der Überwachungsgrößen (Ist gegenüber Soll) [%]

Modull	Chassis	Feder/Neigemodul	Spurführungsmodul	Antriebs/Bremsmodul	Komm/Energiemodul
Überwachungsgröße					
Zeit	0%	5%	-5%	0%	15%
Kosten	-5%	10%	15%	0%	10%
Sachfortschritt	0%	5%	-5%	10%	0%

Änderungsbedarfs-Matrix

Ziel	Min. Zeit	Min. Kosten	Max. Prozessgüte	Min. Personalwechsel
Modul				
Chassis	5%	-5%	5%	0%
Feder/Neigemodul	-10%	5%	-10%	0%
Spurführungsmodul	-15%	20%	-15%	0%
Antriebs/Bremsmodul	-10%	0%	10%	10%
Komm/Energiemodul	5%	-10%	-25%	-15%

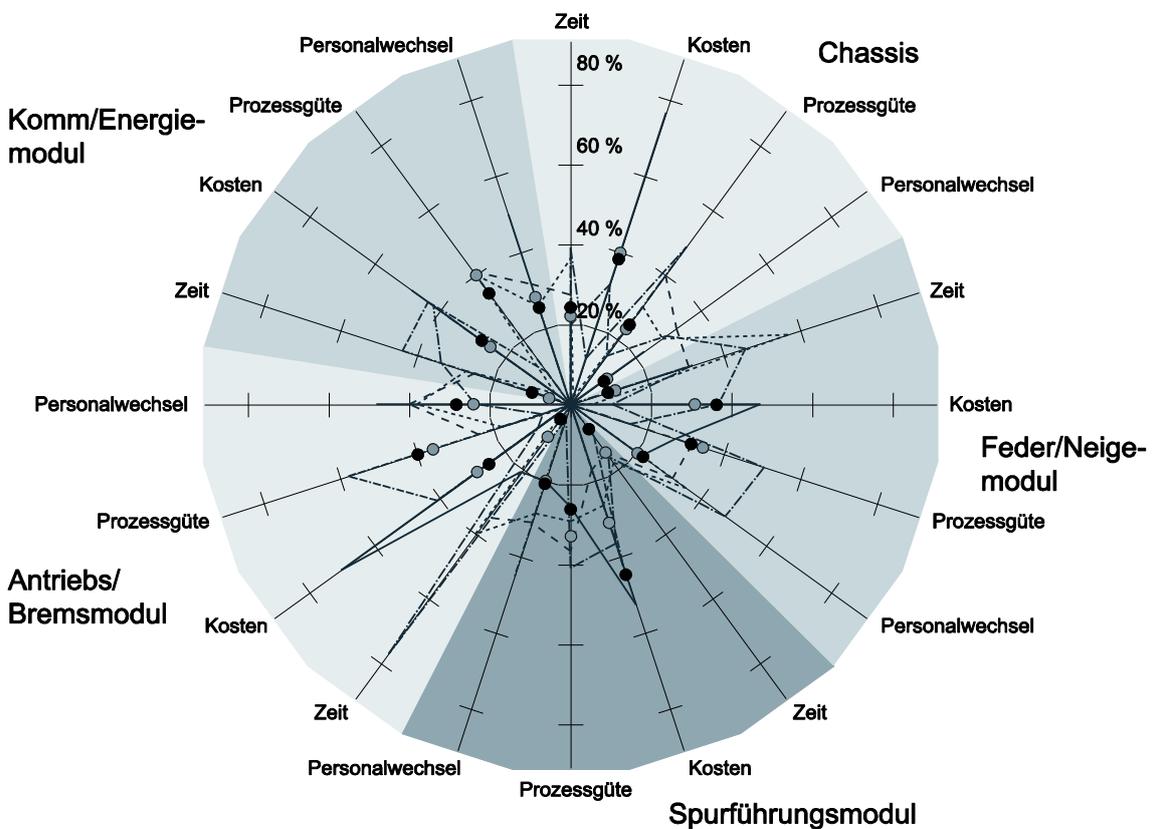
Bild 5-17 Abweichungen gegenüber dem ursprünglichen Prozessplan bei der Weiterentwicklung des RailCabs

Sämtliche dieser Planabweichungen verursachen entsprechend der in Phase 4 in Kapitel 5.3.1 definierten Überwachungsgrößen-Ziel-Einflussmatrix einen Änderungsbedarf der

einzelnen Zielgewichtungen für jedes Modul. Das daraus resultierende neue interne Zielsystem des Entwicklungsprozesses für das RailCab ist in Bild 5-18 dargestellt. Es wird deutlich, dass die Gewichtungen der internen Ziele individuell für jedes Modul verändert werden. Für die nun vorliegenden Gewichtungen ist der bisher nicht zur Anwendung gekommene Plan sieben der am besten geeignete.

gewichtetes internes Zielsystem

Modul \ Ziel	Min. Zeit	Min. Kosten	Max. Prozessgüte	Min. Personalwechsel
Chassis	24,48%	37,51%	26,55%	11,46%
Feder/Neigemodul	8,95%	36,91%	31,38%	22,77%
Spurführungsmodul	6,99%	44,74%	27,01%	21,26%
Antriebs/Bremsmodul	4,68%	26,93%	40,37%	28,02%
Komm/Energiemodul	11,32%	28,02%	35,31%	25,35%



Plan	Plan 2	Plan 3	Plan 4	Plan 7	Plan 9	Zielgewichtung
Linienart	-----	----	-----	-----	-----	●
Abstand zur Zielgewichtung	0,99689	0,90147	1,01528	0,89485	0,89962	-
Rang	4	3	5	1	2	-

Bild 5-18 Entwicklungssituation des RailCabs unter Berücksichtigung von Planabweichungen

Auch diese Ergebnisse entsprechen dem zu erwartenden Ablauf des Selbstoptimierungsprozesses nach Kapitel 4.2.2. Die Planabweichungen haben modulspezifische Veränderungen des internen Zielsystems für alle von Planabweichungen betroffenen Module bewirkt. Entsprechen dem neuen Zielsystem wurde ein Wechsel des Prozessplans bewirkt.

Zusammenfassend lässt sich über alle Verifikations-Szenarien feststellen, dass die dort erzielten Ergebnisse vollumfänglich dem Ablauf des Selbstoptimierungsprozesses gemäß Kapitel 4.2.2 entsprechen. Das in der vorliegenden Arbeit entwickelte *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* und die zu dessen Umsetzung definierten Verfahren arbeiten demzufolge wie angestrebt. Die abschließende Bewertung des Rahmenwerks anhand der in Kapitel 2.7 definierten Anforderungen ist Gegenstand des nachfolgenden Kapitels 5.4.

5.4 Bewertung des Rahmenwerks

Abschließend wird das in der vorliegenden Arbeit entwickelte *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* anhand der Anforderungen aus Kapitel 2.7 bewertet. Zu diesem Zweck wird für jede Anforderung erläutert wie sie durch die Bestandteile des Rahmenwerks erfüllt wird.

A1 Ganzheitlichkeit: Grundlage des Rahmenwerks ist ein systemisches Gesamtmodell des Entwicklungsprozesses fortschrittlicher mechatronischer Systeme, das diesen in unterschiedliche Subsysteme bzw. Aspekte unterteilt (vgl. Abschnitt 4.2.1). Die Aspekte und ihre Wechselwirkungen werden durch das Gesamtmodell strukturiert und gegeneinander abgegrenzt. Ergänzt wird das systemische Gesamtmodell durch das Regelungssystem für einen selbstoptimierenden Entwicklungsprozess (vgl. Abschnitt 4.2.3). Dieses integriert das Projektmanagement mit dem Entwicklungsprozess und stellt die Wirkmechanismen zwischen beiden ganzheitlich dar.

A2 Interdisziplinäre Wechselwirkungen: Die Verwendung eines zentralen Systemmodells zur Beschreibung des Entwicklungsgegenstands über den gesamten Entwicklungsprozess hinweg ermöglicht es, die modelltechnischen Inhalte und Wechselwirkungen sämtlicher an der Entwicklung beteiligten Fachdisziplinen abzubilden (vgl. Abschnitt 4.3.1.1). Prozesstechnische Wechselwirkungen werden durch Abstimmungsbau- steine beschrieben (vgl. Abschnitt 4.3.1.2). Sie bilden sowohl die modulinternen als auch die modulübergreifenden Synchronisations- und Integrationstätigkeiten des Entwicklungsprozesses ab. Das im Rahmen dieser Arbeit entwickelte Verfahren zur systemmodellbasierten Prozessplanung setzt die interdisziplinären Wechselwirkungen in eine adäquate Struktur des Entwicklungsprozesses um (vgl. Abschnitt 4.3.2.1).

A3 Dynamische Vorausplanung: Die für Systemmodell und Prozessbausteine entwickelten Modellierungstechniken erlauben durch die Verwendung hierarchisch gegliederter Attribute (Zuordnung zu Klasse → Zuordnung zu Fachdisziplin → Zuordnung zu Spezialisierung usw.) und das Hierarchiekonzept immer genauere Informationen über den Entwicklungsgegenstand und den Entwicklungsprozess abzubilden (vgl. Abschnitt 4.3.1). Hierdurch synthetisiert das Verfahren zur systemmodellbasierten Prozessplanung immer detailliertere Prozesspläne (vgl. Abschnitt 4.3.2.2). Damit einhergehend können immer genauere Hilfsmittel- und Ressourcenplanungen vorgenommen sowie genauere Prognosen für die Zeit- und Kostenplanung gemacht werden.

A4 Spezifität: Die mit den Bestandteilen des in dieser Arbeit entwickelten Rahmenwerks erstellten Prozesspläne geben den Entwicklern Handlungsanweisungen auf den Detaillierungsstufen „Konstruktionsoperationen“ und „Grundoperationen“ nach HUPKA [Hup76, S. 9]. Die Handlungsanweisungen sind damit problemspezifisch genug, um eine konkrete Hilfe darzustellen. Auf der anderen Seite lassen sie den Entwicklern genug Freiraum für eigene Problemlösungswege. Um erfolgreich angewandte Entwicklungsschritte auch für künftige Entwicklungsprojekte zu nutzen, ist im Vorgehensmodell zur Initialisierung und Nachbereitung des selbstoptimierenden Entwicklungsprozesses eine zyklische Aktualisierung der Wissensbasis vorgesehen (vgl. Abschnitt 4.4).

A5 Problemunabhängigkeit: Um das Rahmenwerk unabhängig von der Art des zu entwickelnden Systems oder der Entwicklungsproblemklasse anwenden zu können arbeiten die entwickelten Verfahren zur Umsetzung auf den im Systemmodell enthaltenen abstrakten Informationen (vgl. Abschnitt 4.3.2). Es wird keine Klassifikation nach Produktart oder Entwicklungsproblem vorgenommen. Die verwendeten Modellierungstechniken und Attribute beschreiben die allgemeinen Charakteristika mechatronischer Systeme, wie z.B. Fachdisziplin, Komplexität, strukturelle Zusammenhänge, Hierarchien usw. Hierdurch können die Verfahren auch auf bisher unbekannte Entwicklungsaufgaben angewendet werden.

A6 Selbstoptimierung: Die Selbstoptimierung des Entwicklungsprozesses wird durch alle Bestandteile des Rahmenwerks ermöglicht. Die beiden Schlüsselemente sind das Verfahren zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen und das Verfahren zur zielkonformen Planauswahl (vgl. Abschnitt 4.3.2.2 und Abschnitt 4.3.2.3). Sie ermöglichen die endogene Anpassung der Entwicklungsprozessziele und damit des Verhaltens des Entwicklungsprozesses. Die Funktionsfähigkeit der Selbstoptimierung wurde in Kapitel 5.3.2 nachgewiesen.

A7 Akzeptanz/Transparenz: Zur Sicherstellung der Akzeptanz und Transparenz des in dieser Arbeit entwickelten Rahmenwerks wurden für die Modellierung der benötigten Informationen mit CONSENS und OMEGA auf Modellierungstechniken aufgebaut, die sich in der Vergangenheit durch eine hohe intuitive Verständlichkeit hervorgetan haben (vgl. Abschnitt 3.2.1.2 und Abschnitt 3.2.2.2). Eine übersichtliche Darstellung der Wirkmechanismen der Selbstoptimierung ist durch das systemische Gesamtmodell des Entwicklungsprozesses sowie das Regelungssystem für einen selbstoptimierenden Entwicklungsprozess gegeben (vgl. Abschnitt 4.2.1 und Abschnitt 4.2.3).

A8 Effizienz: Die Effizienz des Rahmenwerks wird dadurch gewährleistet, dass seine Bestandteile während des Entwicklungsprozesses im Wesentlichen auf Informationen aufbauen, die sowieso im Verlauf der Entwicklung über den Entwicklungsgegenstand sowie für das Projektmanagement generiert werden müssen. Diese Informationen werden für die Anwendung des in dieser Arbeit entwickelten Rahmenwerks punktuell ergänzt, was mit den in Kapitel 5.1 prototypisch entwickelten Software-Werkzeugen mit vertretbarem Aufwand erfolgen kann.

6 Zusammenfassung und Ausblick

Der Großteil der gegenwärtigen maschinenbaulichen Konsum- und Industriegüter beruht auf dem synergetischen Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrik/Elektronik, Regelungs- und Softwaretechnik. Der Begriff Mechatronik bringt dies zum Ausdruck. Mechatronische Produkte bieten Wettbewerbsvorteile, indem sie die Verbesserung der Funktionalität und des Betriebsverhaltens sowie die Reduktion von Kosten, Volumen und Gewicht technischer Erzeugnisse ermöglichen. Gestiegene Marktanforderungen in Kombination mit den rasant wachsenden Möglichkeiten der Informationstechnik führen gegenwärtig zu einer signifikanten Zunahme der Leistungsfähigkeit mechatronischer Erzeugnisse. Ihr Verhalten bekommt zunehmend agierender Charakter; bis hin zu „teilintelligenten“ Verhaltensweisen. Ein Beispiel für solche fortschrittlichen mechatronischen Systeme sind selbstoptimierende Systeme. Sie sind in der Lage ihre Ziele und damit einhergehend ihr Verhalten optimal an veränderte Umwelteinflüsse und auch unvorhergesehene Betriebssituationen anzupassen.

Den genannten Vorteilen mechatronischer Systeme steht eine erhöhte, teils inhärente, teils fremdbestimmte Dynamik und Komplexität von deren Entwicklungsprozess gegenüber. Beide erfahren mit der Zunahme der Leistungsfähigkeit mechatronischer Systeme nochmals eine Erhöhung, sodass die gegenwärtigen Prozessmanagementmethoden für die effiziente Beherrschung des Entwicklungsprozesses nicht mehr ausreichend sind und die Prozessverantwortlichen zusehends überfordert werden. Es ist ein Zustand erreicht in dem das Leistungspotential fortschrittlicher mechatronischer Systeme nicht voll ausgeschöpft werden kann, weil die Dynamik und Komplexität von deren Entwicklungsprozess dies verhindert.

Zur Auflösung dieses Konflikts wird in der vorliegenden Arbeit ein *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* entwickelt. Das Rahmenwerk überträgt das bisher für technische Systeme definierte Paradigma der Selbstoptimierung auf die Planung, Ausführung sowie Überwachung und Steuerung des Entwicklungsprozesses. Hierdurch passt sich der Entwicklungsprozess endogen an veränderte Entwicklungssituationen an und die Überforderung der Prozessverantwortlichen wird eliminiert. Im Sinne eines selbstverstärkenden Systems sollen so die Potentiale fortschrittlicher mechatronischer Systeme voll ausgeschöpft werden können.

Das Rahmenwerk ist hierfür dreistufig aufgebaut. Der erste Hauptbestandteil ist die *theoretische Übertragung des Paradigmas der Selbstoptimierung* auf den Entwicklungsprozess. Hierfür werden die Aspekte der Selbstoptimierung auf den Entwicklungsprozess übertragen und in einem systemischen Gesamtmodell abgebildet. Das Systemmodell fasst den Entwicklungsprozess als soziotechnisches System auf und ist das Fundament des Rahmenwerks. Es ist auch die Basis für die Definition des Ablaufs der Selbstoptimierung des Entwicklungsprozesses, den sogenannten Selbstoptimierungs-

prozess. Der Selbstoptimierungsprozesses gliedert sich in die drei Aktionen: Analyse der Ist-Situation, Bestimmung der Systemziele und Anpassung des Systemverhaltens. Diese werden durch die Bestandteile eines hierfür entwickelten Regelungssystems für einen selbstoptimierenden Entwicklungsprozesses umgesetzt.

Den zweiten Hauptbestandteil des Rahmenwerks bilden die *Werkzeuge zur Umsetzung* des selbstoptimierenden Entwicklungsprozesses. Sie ermöglichen die praktische Umsetzung der theoretischen Grundlagen. Die Werkzeuge umfassen sowohl Modellierungstechniken als auch spezifische Verfahren. Die Modellierungstechniken dienen dem Zweck sämtliche Informationen, die für die Anwendung des Paradigmas der Selbstoptimierung auf den Entwicklungsprozess benötigt werden, konsolidiert bereitzustellen. Es werden sowohl eine Technik zur Abbildung des Systemmodells eines mechatronischen Systems als auch eine Technik zur Modellierung von Entwicklungstätigkeiten in Form von Prozessbausteinen benötigt. Auf der mit den Modellierungstechniken beschriebenen Informationsbasis setzen dann die spezifischen Verfahren zur Umsetzung des selbstoptimierenden Entwicklungsprozesses auf. Diese Verfahren sind: das Verfahren zur systemmodellbasierten Prozessplanung, das Verfahren zur situationsspezifischen Bestimmung und Gewichtung von Entwicklungsprozesszielen und das Verfahren zur zielkonformen Planauswahl. Zusammengenommen realisieren diese drei Verfahren die Selbstoptimierung des Entwicklungsprozesses.

Den dritten Bestandteil des Rahmenwerks bildet das *Vorgehensmodell zur Initialisierung und Nachbereitung* selbstoptimierender Entwicklungsprozesse. Es dient der Konfiguration des Rahmenwerks für ein konkretes Entwicklungsprojekt innerhalb eines Unternehmens und der abschließenden Aufbereitung der gesammelten Erfahrungswerte für künftige Projekte.

Der Nachweis der Funktionsfähigkeit des entwickelten Rahmenwerks und seiner Bestandteile wird im Rahmen der Verifikation am Beispiel des neuartigen Schienenverkehrssystem RailCab erbracht. Es werden vier verschiedene Auslöser für den Selbstoptimierungsprozess und die daraus durch das Rahmenwerk und seine Bestandteile generierten Ergebnisse untersucht. Die betrachteten Auslöser sind: veränderte strategische Vorgaben, veränderte Eigenschaften des Entwicklungsgegenstandes, Veränderungen der Handlungsobjekte und Abweichungen des tatsächlichen Entwicklungsverlaufes gegenüber dem geplanten. In allen vier Fällen kommt es zu einer endogenen Anpassung des Entwicklungsprozesses im Sinne der Selbstoptimierung. Das in dieser Arbeit entwickelte *Rahmenwerk für einen selbstoptimierenden Entwicklungsprozess fortschrittlicher mechatronischer Systeme* erfüllt somit die an es gestellten Anforderungen in vollem Umfang.

Das in dieser Arbeit entwickelte Rahmenwerk öffnet den Weg für **weiteren Forschungsbedarf**. Die für die Umsetzung des selbstoptimierenden Entwicklungsprozesses entwickelten Verfahren stellen eine Möglichkeit zur Realisierung der Selbstoptimierung dar. Sie erheben keinen Anspruch auf Alleingültigkeit. Beispielsweise lässt der Einsatz

probabilistischer Planverfahren bei der systemmodellbasierten Prozessplanung eine höhere Anzahl möglicher Prozesspläne und eine Verbesserung bei der Bestimmung der möglichen Problemlösungswege vermuten. Weiterhin beschränkt sich die Analyse des Entwicklungsgegenstandes für die Prozesssynthese bisher auf die oberste Hierarchieebene des Systemmodells. Eine Erweiterung über mehrere Hierarchieebenen hinweg im Sinne einer stufenweisen Prozesssynthese könnte ebenfalls einen Informationsgewinn und eine Verbesserung der Güte der generierten Pläne bewirken. Bezüglich der systemischen Betrachtung des Entwicklungsgeschehens wären eine Ausweitung des Projektkontextes auf mehrere Projekte im Sinne eines selbstoptimierenden Multiprojektmanagements und die Einbeziehung der Rückwirkungen des Entwicklungsprozesses auf das Handlungssystem die sich als nächstes anbietenden Forschungsfragen. Ferner wäre aufbauend auf der in dieser Arbeit geleisteten Verifikation die empirische Validierung des Rahmenwerks als neuer Beitrag zur empirischen Konstruktionsforschung wünschenswert.

Das in dieser Arbeit entwickelte Rahmenwerk und die vorstehend genannten Forschungsfragen für dessen zukünftige Weiterentwicklung unterstützen das übergeordnete Ziel einer neuen Schule des Entwurfs intelligenter technischer Systeme. Diese neue Schule ist Gegenstand des Sonderforschungsbereichs 614 „Selbstoptimierende Systeme des Maschinenbaus“ (SFB 614) der Universität Paderborn in dessen Rahmen die vorliegende Arbeit entstanden ist.

Abkürzungsverzeichnis

a	Steigung der Gewichtungsfunktion
AHP	Analytic Hierachy Process
AMS	Autonomes mechatronisches System
ARIS	Architektur Integrierter Informationssysteme
ASDP	Adaptive System Development Process
AVT	Aufbau- und Verbindungstechnik
bspw.	beispielsweise
CAX	Computer Aided X
CO	Controller
CONSENS	Conceptual Design Specification Technique for the Engineering of Complex Systems
CPM	Critical Path Method
d.h.	das heißt
DFG	Deutsche Forschungsgemeinschaft
DFX	Design for X
DIN	Deutsches Institut für Normung e.V.
eEPK	erweiterte ereignisgesteuerte Prozessketten
et al.	et alii – und andere
etc.	et cetera – und so weiter
GERT	Graphical Evaluation and Review Technique
ggf.	gegebenenfalls
G_{MjZi}	Gewichtung des internen Ziels i für Modul j
HPG	Hierachical Precedence Graph
i.d.R.	in der Regel
ICAM	Integrated Computer-Aided-Manufacturing
IDEF	Integrated Definition
IPE	Integrierte Produktentstehung

iPeM	integriertes Produktentstehungs-Modell
K_{Mj}	technologische Bedeutung des Modul j
KO	Kognitiver Operator
MFM	Mechatronisches Funktionsmodul
MID	Molded Interconnect Devices
MPM	Meta-Potential-Methode
MSM	Mechatronisches Systemmodell
OCM	Operator-Controller-Modul
od.	oder
OMEGA	Objektorientierte Methode zur Geschäftsprozessmodellierung und -analyse
OMG	Object Management Group
PDM	Produktdatenmanagement
PMDD	Product Model Driven Development
P_{Szi}	strategische Zielpriorität des Ziels i
P_{TMjzi}	technologische Zielpriorität des Ziels i für das Modul j
P_{TnMjzi}	normierte technologische Zielpriorität des Ziels i für das Modul j
R_{Mjzi}	Relevanz des Ziels i für das Modul j
RO	Reflektorischer Operator
SADT	Structur Analysis and Design Technique
SFB	Sonderforschungsbereich
s.u.	siehe unten
SysML	Systems Modeling Language
u.a.	unter anderem
u.U.	unter Umständen
UML	Unified Modeling Language
usw.	und so weiter
vgl.	vergleiche
VHDL	VHSIC Hardware Description Language

VMS	Vernetztes mechatronisches System
XML	Extended Markup Language
z.B.	zum Beispiel
z.T.	zum Teil
Z_i	Ziel i
ZOPH-Modell	Ziel-, Objekt-, Prozess- und Handlungssystem-Modell
$\Delta_{M_j Z_i}$	Zielgewichtsänderungsbedarf des Ziels i für Modul j

Literaturverzeichnis

- [ABM+05] ALBERS, A.; BURKHARDT, N.; MEBOLDT, M.; SAAK, M.: SPALTEN Problem Solving Methodology in the Product Development. In: Proceedings of the International Conference on Engineering Design, ICED05, August 15 - 18, Melbourne, 2005
- [Aca09] ACATECH – DEUTSCHE AKADEMIE DER TECHNIKWISSENSCHAFTEN: Intelligente Objekte – Klein, Vernetzt, Sensitiv – Eine neue Technologie verändert die Gesellschaft und fordert zur Gestaltung heraus. acatech BEZIEHT POSITION, Nr. 5, Springer, Berlin, 2009
- [ADG+09] ADELT, P.; DONOTH, J.; GAUSEMEIER, J.; GEISLER, J.; HENKLER, S.; KAHL, S.; KLÖPPER, B.; KRUPP, A.; MÜNCH, E.; OBERTHÜR, S.; PAIZ, C.; PORRMANN, M.; RADKOWSKI, R.; ROM-AUS, C.; SCHMIDT, A.; SCHULZ, B.; VÖCKING, H.; WITKOWSKI, U.; WITTING, K.; ZNAMENSHCHYKOV, O.: Selbstoptimierende Systeme des Maschinenbaus – Definition, Anwendung, Konzepte. HNI-Verlagsschriftenreihe, Band 234, Paderborn, 2009
- [AE06] ASSMUS, D.; EVERSHEIM, W.: Simultaneous Engineering (ab 1985). In: EVERSHEIM, W.; PFEIFER, T.; WECK, M. (Hrsg.): 100 Jahre Produktionstechnik – Werkzeugmaschinenlabor WZL der RWTH Aachen von 1906 bis 2006, Springer Verlag, Berlin, 2006
- [Alb10] ALBERS, A.: Five Hypothesis about Engineering Processes and their Consequences. In: Proceedings of the TMCE 2010, April 12-16, 2010, Ancona, Italy, 2010
- [AW95] ASTRÖM, K. J.; WITTENMARK, B.: Adaptive Control. Addison-Wesley, Reading, 2. Ed., 1995
- [Bal08] BALZERT, H.: Lehrbuch der Softwaretechnik: Softwaremanagement. Spektrum Akademischer Verlag, Heidelberg, 2. Auflage, 2008
- [BBW07] BEETZ, M.; BUSS, M.; WOLLHERR, D.: Cognitive Technical Systems – What is the Role of Artificial Intelligence? In: KI 2007: Advances in Artificial Intelligence, Lecture Notes in Computer Science, Volume 4667/2007, Springer, Berlin, 2007, pp. 19-42
- [BDT10] Brandis, R.; Dorociak, R.; Terfloth, A.: Softwareunterstützte Modellierung der Prinziplösung – Ein neuer Ansatz für eine integrative Produkt- und Produktionssystementwicklung. In: ProduktDatenJournal, Nr.2.2010, Darmstadt, ProSTEP iViP e.V., 2010
- [Ben05] BENDER, K. (Hrsg.): Embedded Systems – qualitätsorientierte Entwicklug. Springer, Berlin, 1. Auflage, 2005
- [Bic00] BICHLMAIER, C.: Methoden zur flexiblen Gestaltung von integrierten Entwicklungsprozessen. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, Herbert Utz Verlag, München, 2007
- [Bla99] Blackenfelt, M.: On the Development of Modular Mechatronic Products. Licentiate Thesis, Royal Insitut of Technology, KTH, Stockholm, Schweden, 1999
- [Boe88] BOEHM, B. W.: A Spiral Model of Software Development and Enhancement. In: IEEE Computer, Vol. 21, Issue 5, May 1988, IEEE Computer Society, 1988, pp. 61-72
- [Böh96] BÖHM, W.: Elektrische Antriebe. Vogel Buchverlag, Würzburg, 4. überarbeitete Auflage 1996
- [Bra05] BRAUN, T. E.: Methodische Unterstützung der strategischen Produktplanung in einem mittelständisch geprägten Umfeld. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, Verlag Dr. Hut, München, 2005
- [Bro10] BROY, M. (Hrsg.): Cyber-Physical Systems – Innovation durch software-intensive eingebettete Systeme. acatech DISKUTIERT, Springer, Berlin, 2010

- [Bro01] BROWNING, T. R.: Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions. In: IEEE Transactions on Engineering Management, Vol. 48, No. 3, August, 2001, pp. 292-306
- [BSK+06] BÖCKER, J.; SCHULZ, B.; KNOKE, T.; FRÖHLEKE, N.: Self-Optimization as a Framework for Advanced Control Systems. In: 32nd Annual Conference of the IEEE Industrial Electronics Society (IECON 06), November 7-10, 2006, Paris, 2006
- [Bur08] BURGHARDT, M.: Projektmanagement – Leitfaden für die Planung, Überwachung und Steuerung von Projekten, Publicis Corporate Publishing, Erlangen, 8. Wesentlich überarbeitete und erweiterte Auflage, 2008
- [BW10] BIUNDO, S.; WENDEMUTH, A.: Von kognitiven technischen Systemen zu Companion-Systemen. In: KI – Künstliche Intelligenz, Springer-Verlag, Volume 24, Number 4, August 2010, pp. 335–339
- [CE05] CLARKSON, J.; ECKERT, C.: Design process improvement – A review of current practice. Springer Verlag, London, 2005
- [CKS07] Chrissis, M. B.; Konrad, M.; Shrum, S.: CMMI – Guidelines for Process Integration and Product Improvement. Addison-Wesely, Upper Saddle River, New Jersey, USA, 2nd ed., 2007
- [Coc02] COCKBURN, A.: Agile software development. Addison-Wesley, Boston, 2002
- [DCD99] DASGUPTA, P.; CHAKRABARTI, P. P.; DESARKAR, S. C.: Multiojective Heuristic Search – An Introduction to intelligent Search Methods for Multicriteria Optimization. Vieweg & Sohn Verlagsgesellschaft, Bauschweig, 1999
- [Dem00] DEMERS, M. T.: Methoden zur dynamischen Planung und Steuerung von Produktentwicklungsprozessen. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, Verlag Dr. Hut, Band 40, München, 2000
- [DH02] DAENZER, W. F. (Hrsg.); HUBER, F. (Hrsg.); HABERFELLNER, R.; NAGEL, P.; BECKER, M.; BÜCHEL, A.; VON MASSOW, H.: Systems Engineering – Methodik und Praxis. Verlag Industrielle Organisation, Zürich, 11. durchgesehene Auflage, 2002
- [DHK+09] DELL’AERE, A.; HIRSCH, M.; KLÖPPER, B.; KOESTER, M.; KRUPP, A.; KRÜGER, M.; MÜLLER, T.; OBERTHÜR, S.; POOK, S.; PRIESTERJAHN, C.; ROMAUS, C.; SCHMIDT, A.; SONDERMANN-WÖLKE, C.; TICHY, M.; VÖCKING, H.; ZIMMER, D.: Verlässlichkeit selbstoptimierender Systeme - Potentiale nutzen und Risiken vermeiden. HNI-Verlagsschriftenreihe, Band 235, Paderborn, 2009
- [Dör79] DÖRNER, D.: Problemlösen als Informationsverarbeitung. Kolhammer, Stuttgart, 2. Auflage, 1979
- [Dud01] Duden – Das Fremdwörterbuch. WISSENSCHAFTLICHER RAT DER DUDENREDAKTION (Hrsg.), Bibliographisches Institut & F.A. Brockhaus AG, Mannheim, 2001
- [Dum11] DUMITRESCU, R.: Entwicklungssystematik zur Integration kognitiver Funktionen in fortgeschrittene mechatronische Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 286, Paderborn, 2011
- [Ehr07] EHRENSPIEL, K.: Integrierte Produktentwicklung – Denkabläufe, Methodeneinsatz, Zusammenarbeit. Carl Hanser Verlag, München, 3. aktualisierte Auflage, 2007
- [EM099] ELMQVIST, H.; MATTSON, S. E.; OTTER, M.: Modelica – A Language for Physical System Modeling, Visualization and Interaction. In: The 1999 IEEE Symposium on Computer-Aided Control System Design (CACSD99), August 22-29, Hawaii, IEEE Customer Service, Piscataway, 1999
- [Eri98] ERIXON, G.: Modular Function Deployment – A Method for Product Modularisation. PhD Thesis, Royal Insitut of Technology, KTH, Stockholm, Schweden, 1998

- [ES05] EVERSHEIM, W.; SCHUH, G.: Integrierte Produkt- und Prozessgestaltung – Ergebnisse des Sonderforschungsbereichs (SFB) 361 der Deutschen Forschungsgemeinschaft (DFG) an der RWTH Aachen. Springer Verlag, Berlin, 2005
- [EWS+94] EPPINGER, S. D.; WHITNEY, D.; SMITH, R.; GEBALA, D.: A Model-Based Method for Organizing Tasks in Product Development. In: Research in Engineering Design, Vol. 6, No. 1, March 1994, Springer Verlag, London, 1994, pp. 1-13
- [FGD+07] FRANK, U.; GAUSEMEIER, J.; DANGELMAIER, W.; SCHMIDT, A.; ZIMMER, D.; KLÖPPER, B.: Using Active Patterns for the Conceptual Design of Self-Optimizing Systems Exemplified by an Air Gap Adjustment System. In: 2007 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, September 4-7, Las Vegas, USA, 2007, pp. 935-944
- [FGK+04] FRANK, U.; GIESE, H.; KLEIN, F.; OBERSCHELP, O.; SCHMIDT, A.; SCHULZ, B.; VÖCKING, H.; WITTING, K.: Selbstoptimierende Systeme des Maschinenbaus – Definitionen und Konzepte. HNI-Verlagsschriftenreihe, Band 155, Paderborn, 2004
- [FH11] FISCHER, P.; HOFER, P.: Lexikon der Informatik. Springer-Verlag, Heidelberg, 15. überarbeitete Auflage, 2011
- [Föl08] FÖLLINGER, O.: Regelungstechnik – Einführung in die Methoden und ihre Anwendung. Hüthig Verlag, Heidelberg, 10. durchgesehenes Auflage, 2008
- [Fra06] FRANK, U.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 175, 2006
- [Fri93] FRICKE, G.: Konstruieren als flexibler Problemlösungsprozess – Empirische Untersuchung über erfolgreiche Strategien und methodische Vorgehensweisen beim Konstruieren. VDI-Verlag, Düsseldorf, 1993
- [Gab10] GABLER: Gabler Kompakt-Lexikon Wirtschaft – 4.500 Begriffe nachschlagen, verstehen, anwenden. Gabler Verlag, Wiesbaden, 10., vollständig überarbeitete und erweiterte Auflage, 2010
- [Gad10] Gadatsch, A.: Grundkurs Geschäftsprozess-Management - Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker. Vieweg + Teubner, Wiesbaden, 6., aktualisierte Auflage, 2010
- [Gau10] GAUSEMEIER, J. (Hrsg.): Frühzeitige Zuverlässigkeitsanalyse mechatronischer Systeme. Carl Hanser Verlag, München, 2010
- [GDD+10] GAUSEMEIER, J.; DONOTH, J.; DUMITRESCU, R.; TRÄCHTLER, A.; REINOLD, P.: Self-Optimization – An Approach for Intelligent Mechatronics Exemplified by an X-by-wire Vehicle. 8th IEEE International Conference on Industrial Informatics – INDIN2010, July 13-16, 2010, Osaka, Japan, pp. 757-762
- [GDK10] GAUSEMEIER, J.; DOROCIAC, R.; KAISER, L.: Computer-Aided Modeling of the Principle Solution of Mechatronic Systems – A Domain-Spanning Methodology for the Conceptual Design of Mechatronic Systems. In: Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE2010), August 15-18, Montreal, Canada, 2010
- [GDT+11] GAUSEMEIER, J.; DUMITRESCU, R.; TSCHIRNER, C.; STILLE, K. S.: Modellbasierte Konzipierung eines hybriden Energiespeichersystems für ein autonomes Schienenfahrzeug. Tag des Systems Engineering, 9.-11. November, Hamburg, 2011
- [GEK01] GAUSEMEIER, J.; EBBESMEYER, P.; KALLMEYER, F.: Produktinnovation – Strategische Planung und Entwicklung der Produkte von morgen. Carl Hanser Verlag, München, 2001
- [GFD+08] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme des Maschinenbaus (Teil 1). In: Konstruktion, Ausgabe 7/8-2008, Springer VDI-Verlag, Düsseldorf, 2008

- [GFD+08a] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme des Maschinenbaus (Teil 2). In: Konstruktion, Ausgabe 9-2008, Springer VDI-Verlag, Düsseldorf, 2008
- [GHK+06] GAUSEMEIER, J.; HAHN, A.; KESPOHL, H. D.; SEIFERT, L.: Vernetzte Produktentwicklung – Der erfolgreiche Weg zum Global Engineering Networking. Carl Hanser Verlag, München, 2006
- [GK83] GAJSKI, D. D.; KUHN, R. H.: Guest Editors' Introduction – New VLSI Tools. In: IEEE Computer, Vol. 16, No. 12, 1983, pp. 11-14
- [GLR+00] GAUSEMEIER, J.; LINDEMANN, U.; REINHARDT, G.; WIENDAHL, H.-P.: Kooperatives Produktengineering – Ein neues Selbstverständnis des ingenieurmäßigen Wirkens. HNI-Verlagsschriftenreihe, Band 79, Paderborn, 2000
- [GPW09] GAUSEMEIER, J.; PLASS, C.; WENZELMANN, C.: Zukunftsorientierte Unternehmensplanung – Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen. Carl Hanser Verlag, München, 2009
- [GRS03] GÖRZ, G.; ROLLINGER, C.-R.; SCHNEEBERGER, J. (Hrsg.): Handbuch der Künstlichen Intelligenz. Oldenburg Wissenschaftsverlag, München, 4. Auflage, 2003
- [GSG+09] GAUSEMEIER, J.; SCHÄFER, W.; GREENYER, J.; KAHL, S.; POOK, S.; RIEKE, J.: Consistency Management between Domain-Spanning Specification and Domain-Specific Models. In: Proceedings of 17th International Conference on Engineering Design, 24-27 August, Stanford, USA, 2009
- [GW11] GAUSEMEIER, J.; WIENDAHL, H.-P. (Hrsg.): Wertschöpfung und Beschäftigung in Deutschland. acatech DISKUTIERT, Springer-Verlag, Berlin, 2011
- [GZF+07] GAUSEMEIER, J.; ZIMMER, D.; FRANK, U.; POOK, S.; SCHMIDT, A.: Conceptual Design of Self-Optimizing Systems Exemplified by a Magnetic Linear Drive. In: International Conference On Engineering Design, ICED'07, August 28-31, Paris, France, 2007
- [Ham07] HAMMER, M.: Der große Prozess-Check. In: Harvard Business Manager, Mai 2007, S. 35-52
- [Han74] HANSEN, F.: Konstruktionswissenschaft – Grundlagen und Methoden. Carl Hanser Verlag, München, 1974
- [Hes06] HESTERMEYER, T.: Strukturierte Entwicklung der Informationsverarbeitung für die aktive Federung eines Schienenfahrzeugs. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, 2006
- [Hol07] HOLMDAHL, L.: Complexity Aspects of Product Development. Dissertation, Fakultät für Maschinenbau, Otto-von-Guericke-Universität, Magdeburg, 2007
- [HS09] HERZOG, O.; SCHILDHAUER, T. (Hrsg.): Intelligente Objekte – Technische Gestaltung – Wirtschaftliche Verwertung – Gesellschaftliche Wirkung. acatech DISKUTIERT, Springer-Verlag, Berlin, 2009
- [HTF96] HARASHIMA, F.; TOMIZUKA, M.; FUKUDA, T.: Mechatronics – „What Is It, Why, and How?“ – An Editorial. In: IEEE/ASME Transactions on Mechatronics, Volume 1, Nr. 1, 1996
- [Hup76] HUPKA, V.: Theorie der Konstruktionsprozesse – Analyse der Konstruktionstätigkeiten. Springer Verlag, Berlin, 1976
- [Ise08] ISERMANN, R.: Mechatronische Systeme – Grundlagen. Springer-Verlag, Berlin, 2. vollständig überarbeitete Auflage, 2008
- [Kal58] KALMAN, R. E.: Design of a self-optimizing control system. In: Transactions of the American Society of Mechanical Engineers, Volume 80, 1958, pp. 468-478

- [Kal98] KALLMEYER, F.: Eine Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 42, 1998
- [KEL+10] KREHMER, H.; ECKSTEIN, W.; LAUER, W.; ROELOFSEN, J.; STÖBER, C.; TROLL, A.; ZAPF, J.; WEBER, N.; MEERKAMM, H.; HEINRICH, A.; LINDEMANN, U.; RIEG, F.; WARTZACK, S.: Das FORFLOW-Prozessmodell zur Unterstützung der multidisziplinären Produktentwicklung. In: Konstruktion, Ausgabe 10-2010, Springer VDI-Verlag, Düsseldorf, 2010
- [KFG07] KRAUSE, F.-L.; FRANKE, H.-J.; GAUSEMEIER, J.: Innovationspotentiale in der Produktentwicklung, Carl Hanser Verlag, München, 2007
- [KKK95] KRSTIC, M.; KANELAKOPOULOS, I.; KOKOTOVIC, P.: Nonlinear and Adaptive Control Design. John Wiley & Sons, New York, 1995
- [Kl010] KLÖPPER, B.; ISHIKAWA, F.; HONIDEN, S.: Service Composition with Pareto-Optimality of Time-Dependent QoS Attributes In: Proceedings of the 8th International Conference on Service-Oriented Computing (ICSOC 2010), pp. 635-640, Spring Lecture Notes in Computer Science, 2010
- [Kl011] Klöpper, B.; Pater, J. P.; Honiden, S.; Dangelmaier, W.: A Multi-Objective Evolutionary Approach to Scheduling for Evolving Manufacturing Systems. Evolving Systems, Springer Verlag, Heidelberg, 2011, pp. 1-18
- [Kol94] KOLLER, R.: Konstruktionslehre für den Maschinenbau – Grundlagen zur Neu- und Weiterentwicklung technischer Produkte. Springer Verlag, Berlin, 3. völlig neubearbeitete und erweiterte Auflage, 1994
- [KSM08] KREHMER, H.; STÖBER, C.; MEERKAMM, H.: Approach on the control of iterations in the multidisciplinary development of technical systems. In: Proceedings of the 10th International Design Conference DESIGN 2008, May 19-22, 2008, Dubrovnik, Croatia, 2008, pp. 1303-1310
- [LB09] LÉVÁRDY, V.; BROWNING, T. R.: An Adaptive Process Model to Support Product Development. In: IEEE Transactions on Engineering Management, vol. 56, No. 4, November, 2009
- [Lev06] LÉVÁRDY, V.: Model-based Framework for the Adaptive Development of Engineering Systems. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, 2006
- [LHL01] LÜCKEL, J.; HESTERMEYER, T.; LIU-HENKE, X.: Generalization of the Cascade Principle in View of a Structured Form of Mechatronic Systems. In: International Conference on Advanced Intelligent Mechatronics (AIM 2001), IEEE/ASME, Como, Italy, 2001, pp. 123-128
- [Lin07] LINDEMANN, U.: Methodische Entwicklung technischer Produkte – Methoden flexibel und situationsgerecht anwenden. Springer Verlag, Berlin, 2. bearbeitete Auflage, 2007
- [LMB07] LINDEMANN, U.; MAURER, M.; BRAUN, T.: Structural Complexity Management – An Approach for the Field of Product Design. Springer Verlag, Berlin, 2007
- [Low09] LOW, C. Y.: A Methodology to Manage the Transition from the Principle Solution towards the Controller Design of Advanced Mechatronic Systems. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 254, Paderborn, 2009
- [LRZ06] LINDEMANN, U.; REICHWALD, R.; ZÄH, M. F.: Individualisierte Produkte – Komplexität beherrschen in Entwicklung und Produktion. Springer Verlag, Berlin, 2006
- [Lun10] LUNZE, JÄ. Regelungstechnik 1. Springer Verlag, Berlin, 2010
- [MBB+08] MONTEMERLO, M.; BECKER, J.; BHAT, S.; DAHLKAMP, H.; DOLGOV, D.; ETTINGER, S.; HAEHNEL, D.; HILDEN, T.; HOFFMANN, G.; HUHNKE, B.; JOHNSTON, D.; KLUMPP, S.; LANGER, D.; LEVANDOWSKI, A.; LEVINSON, J.; MARCIL, J.; ORENSTEIN, D.; PAEFGEN, J.; PENNY, I.; PETROVSKAYA, A.; PFLUEGER, M.; STANEK, G.; STAVENS, D.; VOGT, A.; THRUN, S.: Ju-

- nior – The Stanford Entry in the Urban Challenge. In: Journal of Field Robotics, Wiley Periodicals, Volume 25, Issue 9, September 2008, pp. 569–597
- [MGP73] MILLER, G. A.; GALANTER, E.; PRIBRAM, K. H.: Strategien des Handelns – Pläne und Strukturen des Verhaltens. Klett-Cotta, Stuttgart, 1. Auflage, 1973
- [MHJ+09] MEERKAMM, H.; HEINRICH, A.; JABLONSKI, S.; KRUMHOLTZ, H.; LINDEMANN, U.; RIEG, F. (Hrsg.): Flexible Prozessunterstützung in der Produktentwicklung: Prozesse – Daten – Navigation – Bayerischer Forschungsverbund FORFLOW für Prozess- und Workflowunterstützung zur Planung und Steuerung der Abläufe in der Produktentwicklung – Abschlussbericht 01.10.2006 – 30.09.2009. Shaker Verlag, Aachen, 2009
- [Mod10] Modelica Association: Modelica® A Unified Object-Oriented Language for Physical Systems Modeling – Language Specification – Version 3.2. Linköping, Sweden, 2010
- [Möh04] MÖHRINGER, S.: Entwicklungsmethodik für mechatronische Systeme. Habilitation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 156, Paderborn, 2004
- [Mor69] MORI, T.: Yaskawa Internal Trademark Application Memo, 21.131.01, July 1969
- [MSF09] MANN, H.; SCHIFFELGEN, H.; FRORIEP, R.: Einführung in die Regelungstechnik – Analoge und digitale Regelung, Fuzzy-Regler, Regler-Realisierung, Software, Carl Hanser Verlag, München, 11. neu bearbeitete Auflage, 2009
- [Mül00] MÜLLER, M.: Modularisierung von Produkten – Entwicklungszeiten und Kosten reduzieren. Carl Hanser Verlag, München, 2000
- [Mur99] MURR, o.: Adaptive Planung und Steuerung von integrierten Entwicklungs- und Planungsprozessen. In: REINHART, G.: Forschungsberichte iwv, Band 130, Herbert Utz Verlag, München, 1999
- [Neg06] NEGELE, H.: Systemtechnische Methodik zur ganzheitlichen Modellierung am Beispiel der integrierten Produktentwicklung, Herbert Utz Verlag, München, 2., unveränderte Auflage, 2006
- [Oes05] OESTERREICH, B.: Objektorientierte Softwareentwicklung. Analyse und Design mit der Unified Modeling Language. Oldenbourg Verlag, München, 4. Auflage, 1998
- [OMG10] OBJECT MANAGEMENT GROUP, Inc. (OMG): OMG Systems Modeling Language (OMG SysML™). Version 1.2 with change bars. OMG Document Number: formal/2010-06-02, URL: <http://www.omg.org/spec/SysML/1.2/>, 2010
- [Ott04] OTTOSON, S.: Dynamic product development – DPD. In: Technovation, Vol. 24, Issue 3, December 2004, Elsevier, 2004, pp. 207-217
- [Par07] PARK, G.-J.: Analytic Methods for Design Practice. Springer Verlag, London, 2007
- [Pat82] PATZAK, G.: Systemtechnik – Planung komplexer innovativer Systeme – Grundlagen, Methoden, Techniken. Springer Verlag, Berlin, 1982
- [PBF+07] PAHL, G.; BEITZ, W.; FELDHUSEN, J., GROTE, K. H.: Konstruktionslehre - Grundlagen erfolgreicher Produktentwicklung Methoden und Anwendung, Springer-Verlag, Berlin, 7. Auflage, 2007
- [PE94] PIMMLER, T. U.; EPPINGER, S. D.: Integration Analysis of Product Decompositions. In: Proceedings of the ASME Conference on Design Theory and Methodology, September 1994, Minneapolis, MN, USA, 1994, pp. 343-351
- [Pon07] PONN, J. C.: Situative Unterstützung der methodischen Konzeptentwicklung technischer Produkte. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, Verlag Dr. Hut, München, 2007

- [Poo11] POOK, S.: Eine Methode zum Entwurf von Zielsystemen selbstoptimierender mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 296, Paderborn, 2011
- [Pul04] PULM, U.: Eine systemtheoretische Betrachtung der Produktentwicklung. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, Verlag Dr. Hut, München, 2004
- [Rie03] RIEPE, B.: Integrierte Produktstrukturmodellierung in den frühen Phasen der Produktentstehung – Eine Methode zur Modularisierung variantenreicher mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Books on Demand GmbH, Norderstedt, 1. Auflage, 2003
- [Rop09] ROPOHL, G.: Allgemeine Technologie – Eine Systemtheorie der Technik. Universitätsverlag, Karlsruhe, 2009
- [RKL+08] ROELOFSEN, J.; KREHMER, H.; LINDEMANN, U.; MEERKAMM, H.: Using the Design-Structure-Matrix for the Avoidance of Unnecessary Iterations. In: Proceedings of the 10th International DSM Conference, November 11-12, 2008, Stockholm, Sweden, Carl Hanser Verlag, Munich, 2008
- [RLL07] ROELOFSEN, J.; LAUER, W.; LINDEMANN, U.: Product Model Driven Development. In: Proceedings of the 14th European Concurrent Engineering Conference ECEC 2007, April 25-27, 2007, Delft, Netherlands, 2007, pp. 20-26
- [Roe11] ROELOFSON, J. K. M.: Situationsspezifische Planung von Entwicklungsprozessen. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, Verlag Dr. Hut, München, 2011
- [Ros85] ROSS, D. T.: Applications and Extensions of SADT. In: Computer, IEEE Computer Society, Washington (DC), Volume 18, Issue 4, April 1985, pp. 25 - 34
- [Rot82] ROTH, K.: Konstruieren mit Konstruktionskatalogen – Systematisierung und zweckmäßige Aufbereitung technischer Sachverhalte für das methodische Konstruieren. Springer Verlag, Berlin, 1982
- [Roy70] ROYCE, W. W.: Managing the development of large Software Systems. In: Proceedings of the IEEE Western Electronic Show and Convention (WESCON), August 25-28, 1970, Los Angeles, California, USA, 1970, pp. A1-1 - A1-9
- [Saa05] SAATY, T. L.: The Analytic Hierarchy and Analytic Network Processes for the Measurement of Intangible Criteria and for Decision Making. In: FIGUEIRA, J.; GRECO, S.; EHRGOTT, M. (Eds.): Multiple Criteria Decision Analysis – State of the Art Surveys. Springer Verlag, Boston, 2005
- [Sch94] SCHUMANN, G.: Adaptive Planung des Produktentwicklungsprozesses. In: SPUR, G. (Hrsg.): Produktionstechnik – Berlin – Forschungsberichte für die Praxis, Band 146, Carl Hanser Verlag, Berlin, 1994
- [Sch05] SCHUH, G.: Produktkomplexität managen – Strategien, Methoden, Tools. Carl Hanser Verlag, München, 2., überarbeitete und erweiterte Auflage, 2005
- [Sei06] SEIDLMEIER, H.: Prozessmodellierung mit ARIS – Eine beispielorientierte Einführung für Studium und Praxis. Vieweg & Sohn Verlag, Wiesbaden, 2., aktualisierte Auflage, 2006
- [SER07] SOSA, M. E.; EPPINGER, S. D.; ROWLES, C. M.: Are Your Engineers Talking to One Another When They Should? In: Harvard Business Review, Vol. 85, No. 11, November 2007, Harvard Business Publishing, 2007, pp. 133-142
- [SFB08] SONDERFORSCHUNGSBEREICH 614: Selbstoptimierende Systeme des Maschinenbaus – Finanzierungsantrag 2. Halbjahr 2009 bis 1. Halbjahr 2013. Band 1, Universität Paderborn, 2008

- [SHI+10] STARK, R.; HAYKA, H.; ISRAEL, J. H.; KIM, M.; MÜLLER, P.; VÖLLINGER, U.: Virtuelle Produktentstehung in der Automobilindustrie. In: Informatik-Spektrum, Ausgabe 33, Springer Verlag, Berlin, 2010
- [Smi97] SMITH, M.: Application-Specific Integrated Circuits. Addison-Wesley, Berkeley, 1997
- [SS06] SCHEER, A.-W.; SCHNEIDER, K.: ARIS – Architecture of Intergrated Information Systems. In: BERNUS, P.; MERTINS, K.; SCHMIDT, G.: Handbook on Architectures of Information Systems (Ed.). Springer Verlag, Berlin, 2 Edition, 2006
- [SSW07] SAHIN, E.; SPEARS, W. M.; WINFIELD, A. F. T.: Swarm Robotics. In: Lecture Notes in Computer Science, Volume 4433/2007, Springer, Berlin, 2007
- [Ste07] STEFFEN, D.: Ein Verfahren zur Produktstrukturierung für fortgeschrittene mechatronische Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 207, Paderborn, 2007
- [Ste76] STEINWACHS, H. O.: Praktische Konstruktionsmethode – Anleitung zum methodischen Konstruieren. Vogel Verlag, Würzburg, 1976
- [Sys06] SYSKA, A.: Produktionsmanagement – Das A - Z wichtiger Methoden und Konzepte für die Produktion von heute, Gabler Verlag, Wiesbaden, 2006
- [Szy02] SZYPERSKI, C.: Component Software – Beyond Object-Oriented Programming. Addison-Wesley Longman, Amsterdam, 2. Auflage, 2002
- [Tei97] TEICH, J.: Digitale Hardware/Software-Systeme – Synthese und Optimierung. Springer Verlag, Berlin, 1997
- [TMD+06] THRUN, S.; MONTEMERLO, M.; DAHLKAMP, H.; STAVENS, D.; ARON, A.; DIEBEL, J.; FONG, P.; GALE, J.; HALPENNY, M.; HOFFMANN, G.; LAU, K.; OAKLEY, C.; PALATUCCI, M.; PRATT, V.; STANG, P.; STROHBAND, S.; DUPONT, C.; JENDROSSEK, L.-E.; KOELEN, C.; MARKEY, C.; RUMMEL, C.; VAN NIEKERK, J.; JENSEN, E.; ALESSANDRINI, P.; BRADSKI, G.; DAVIES, B.; ETTINGER, S.; KAEHLER, A.; NEFIAN, A.; MAHONEY P.: Stanley – the robot that won the DARPA Grand Challenge. In: Journal of Field Robotics, Wiley Periodicals, Volume 23, Issue 9, September 2006, pages 661–692
- [Trä06] TRÄCHTLER, A.: RailCab – mit innovativer Mechatronik zum Schienenverkehrssystem der Zukunft. In: VDE (Hrsg.): VDE Kongress 2006 – Innovations for Europe – Fachtagungsbericht der ITG/BMBF, 23. - 25. Oktober 2006, Aachen, VDE-Verlag, 2006
- [Wei06] WEILKIENS, T.: Systems Engineering mit SysML/UML – Modellierung, Analyse, Design. dpunkt.verlag, Heidelberg, 1. Auflage, 2006
- [Wei11] WEIDAUER, J.: Elektrische Antriebstechnik – Grundlagen, Auslegung, Anwendungen, Lösungen. Publicis Corporate Publishing, Erlangen, 2. überarbeitete Auflage, 2011
- [Wie00] WIENDL, W.: Produktintegritätskonzept zum Einsatz im Concurrent Engineering. Dissertation, Fachbereich Maschinenbau, Universität Hannover, 2000
- [Wil91] WILCOX, A. D.: Entwicklerleitfaden Elektronik. Carl Hanser Verlag, München 1991
- [ZFH+11] ZEMAN, K.; FOLLMER, M.; HEHENBERGER, P.; PUNZ, S.; ROSEN, R.: Systemmodelle zur Entwicklung mechatronischer Produkte. In: Tagungsband Internationales Forum Mechatronik 2011, 21. - 22. September, Cham, Chilipaper GmbH, 2011, S. 381-392
- [ZFH+11a] ZEMAN, K.; FOLLMER, M.; HEHENBERGER, P.; PUNZ, S.; ROSEN, R.: Approach for the creation of Mechatronic System Models. In: Proceedings of the International Conference on Engineering Design, ICED11, August 15 - 18, Copenhagen, Denmark, 2011

Normen und Richtlinien

- [VDI2206] VEREIN DEUTSCHER INGENIEURE (VDI): VDI-Richtlinie 2206 – Entwicklungsmethodik für mechatronische Systeme. Beuth Verlag, Berlin, Juni, 2004
- [VDI2209] VEREIN DEUTSCHER INGENIEURE (VDI): VDI-Richtlinie 2209 – 3-D-Produktmodellierung – Technische und organisatorische Voraussetzungen Verfahren, Werkzeuge und Anwendungen – Wirtschaftlicher Einsatz in der Praxis. Beuth Verlag, Berlin, März, 2009
- [VDI2221] VEREIN DEUTSCHER INGENIEURE (VDI): VDI-Richtlinie 2221 – Methodik zum Entwickeln und Konstruieren technischer Produkte. Beuth Verlag, Berlin, Juni, 1993
- [VDI2222/1] VEREIN DEUTSCHER INGENIEURE (VDI): VDI-Richtlinie 2222 – Methodisches Entwickeln von Lösungsprinzipien. Beuth Verlag, Berlin, Juni, 1997
- [VDI2223] VEREIN DEUTSCHER INGENIEURE (VDI): VDI-Richtlinie 2223 – Methodisches Entwerfen technischer Produkte. Beuth Verlag, Berlin, Juni, 2004
- [VDI/VDE2224] VEREIN DEUTSCHER INGENIEURE (VDI); VEREIN DEUTSCHER ELEKTROTECHNIKER: VDI/VDE-Richtlinie 2224 – Entwicklungsmethodik für Geräte mit Steuerung durch Mikroelektronik. Beuth Verlag, Berlin, Juni, 1994
- [DIN60050] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (DIN): DIN IEC 60050-351 – Internationales Elektrotechnisches Wörterbuch – Teil 351: Leittechnik. Beuth Verlag, Berlin, Juni, 2009
- [DIN69901-2] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (DIN): DIN 69901-2 – Projektmanagement – Projektmanagementsysteme – Teil 3: Methoden. Beuth Verlag, Berlin, Januar, 2009
- [DIN69901-5] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (DIN): DIN 69901-5 – Projektmanagement – Projektmanagementsysteme – Teil 5: Begriffe. Beuth Verlag, Berlin, Januar, 2009
- [DIN1421] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (DIN): DIN 1421 – Gliederung und Benennung in Texten – Abschnitte, Absätze, Aufzählungen. Beuth-Verlag, Berlin, 1983
- [DIN19226] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (DIN): DIN 19226 Teil 1 – Leittechnik, Regelungstechnik und Steuerungstechnik – Allgemeine Grundbegriffe. Beuth Verlag, Berlin, 1995
- [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., Newtown Square, PA, USA, 2004
- [DIN ISO/IEC15504-2] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (DIN): DIN ISO/IEC 15504-2 – Informationstechnik – Prozess-Assessment – Teil 2: Durchführung eines Assessments. Beuth Verlag, Berlin, Januar, 2010
- [DIN EN ISO 9000] DEUTSCHES INSTITUT FÜR NORMUNG E.V. (DIN): DIN EN ISO 9000 – Qualitätsmanagementsysteme – Grundlagen und Begriffe. Beuth Verlag, Berlin, Dezember, 2005

Anhang

Inhaltsverzeichnis	Seite
A1	Datenmodell der Prozessbausteine A-1
A2	Verfahren zur Systemstrukturanalyse A-3
A2.1	Systemstrukturanalyse – Dynamikindex von Systemelementen A-3
A2.2	Systemstrukturanalyse – funktionsorientierte Reihenfolgebestimmung fachdisziplinübergreifender Integrations- und Validierungsaktivitäten..... A-4
A3	Prozessbausteine für den Entwicklungsprozess fortschrittlicher mechatronischer Systeme A-7
A4	Situationsspezifische Auswahl und Gewichtung von Entwicklungsprozesszielen A-17

A1 Datenmodell der Prozessbausteine

Bild A-1 zeigt das Datenmodell zur Repräsentation der für die vorliegende Arbeit entwickelten Prozessbausteine. Es stellt eine Erweiterung des OMEGA-Datenmodells dar. Zur besseren Lesbarkeit des Bildes sind nur die für die vorliegende Arbeit maßgeblichen Objekte, Attribute, Funktionen und Beziehungen abgebildet.

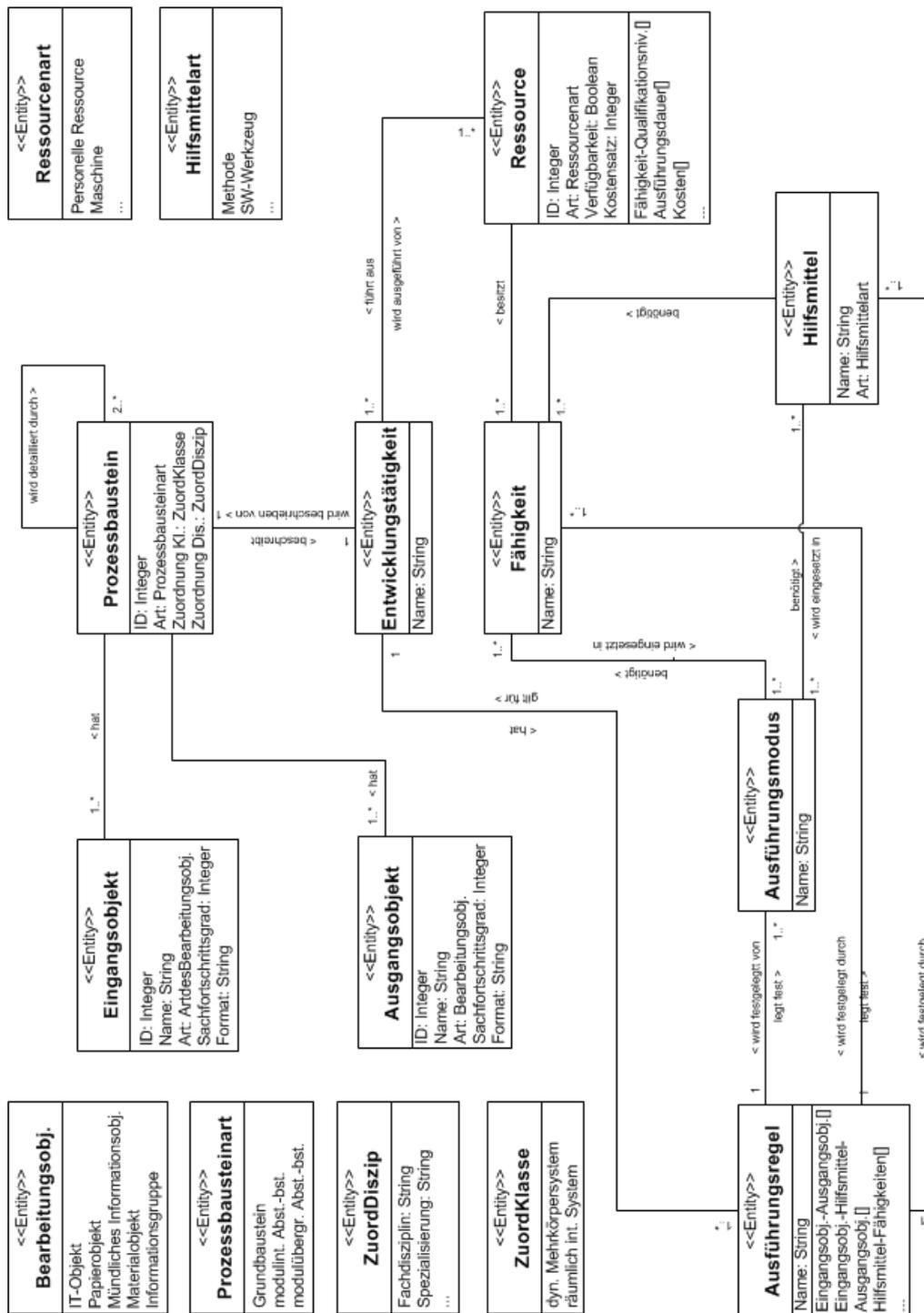


Bild A-1 Datenmodell zur Repräsentation von Prozessbausteinen

A2 Verfahren zur Systemstrukturanalyse

Nachfolgend werden zwei, im Rahmen der vorliegenden Arbeit entwickelte Verfahren zur Unterstützung der Systemstrukturanalyse fortschrittlicher mechatronischer Systeme vorgestellt.

A2.1 Systemstrukturanalyse – Dynamikindex von Systemelementen

Nachfolgend wird das in dieser Arbeit verwendete Verfahren zur Ermittlung des Dynamikindex eines Systemelements vorgestellt. Der Dynamikindex gibt an, wie stark ein Systemelement innerhalb des Gesamtsystems eingebunden ist bzw. dessen Verhalten beeinflusst. Das hier vorgestellte Verfahren ist folglich den Verfahren zur Systemstrukturanalyse zuzuordnen.

Ausgangsbasis der Dynamikindexermittlung ist eine, mit der Spezifikationstechnik des SFB 614⁴⁹ (vgl. [GFD+08a, S. 91ff.]) erstellte Wirkstruktur des zu analysierenden Systems. Es wird davon ausgegangen dass die Wirkstruktur einen homogenen Detaillierungsgrad über alle Systemelemente und deren Verbindungen besitzt. Der Dynamikindex wird auf Basis einer indirekten Einflussanalyse ermittelt (siehe Beispiel Bild A-2). Zunächst werden sämtliche Systemelemente, bis auf das aktuell betrachtete, auf ihre Elementarsystemelemente oder falls vorhanden Lösungselemente herunter gebrochen. Die Elemente werden in den Zeilen und Spalten einer direkten Einflussmatrix gegenübergestellt. Die Zahlenwerte in den Schnittpunkten der Zeilen und Spalten geben an, wie stark das Element in der Zeile das in der Spalte beeinflusst. Die Werte werden anhand der zwischen den Elementen innerhalb der Wirkstruktur bestehenden gerichteten und ungerichteten Verbindungen ermittelt. Verbindungen nach außerhalb der Systemgrenze werden nicht betrachtet, da sich die Funktionalität auf die systeminhärente Funktionalität bezieht. Eine Null bedeutet es besteht kein Einfluss bzw. keine Verbindung zwischen den jeweiligen Elementen. Eine Drei bedeutet, es besteht ein starker Einfluss bzw. viele Verbindungen zwischen den jeweiligen Elementen.

Im nächsten Schritt wird die direkte Einflussmatrix in eine indirekte Einflussmatrix überführt. Die indirekte Einflussmatrix berücksichtigt auch solche Einflüsse, die dadurch entstehen, dass ein Systemelement über ein zweites Systemelement auf ein drittes wirken kann, ohne direkt mit diesem verbunden zu sein. Um zu berücksichtigen, dass diese Beeinflussung nicht mehr so stark ist wie eine direkte, wird bei der Berechnung der indirekten Beeinflussung ein entsprechender Dämpfungsfaktor verwendet.

⁴⁹ Die Modellierung kann alternativ auch mit den Modellierungssprachen UML oder SysML erfolgen. Für beide Sprachen können Profile entsprechend dem Metamodell der Spezifikationstechnik des SFB 614 erstellt werden.

Nach der Berechnung der indirekten Beeinflussung werden die Zeilen- und Spaltensummen für das aktuell zu analysierende Systemelement in der indirekten Einflussmatrix berechnet. Die Zeilensumme ist die Aktivsumme des Systemelements. Sie gibt an, wie stark das Systemelement die anderen Elemente beeinflusst. Die Spaltensumme ist die Passivsumme. Sie ist ein Maß für die Beeinflussung des Systemelements durch die anderen Elemente. Der Dynamikindex ist durch die euklidische Norm aus Aktiv- und Passivsumme gegeben.

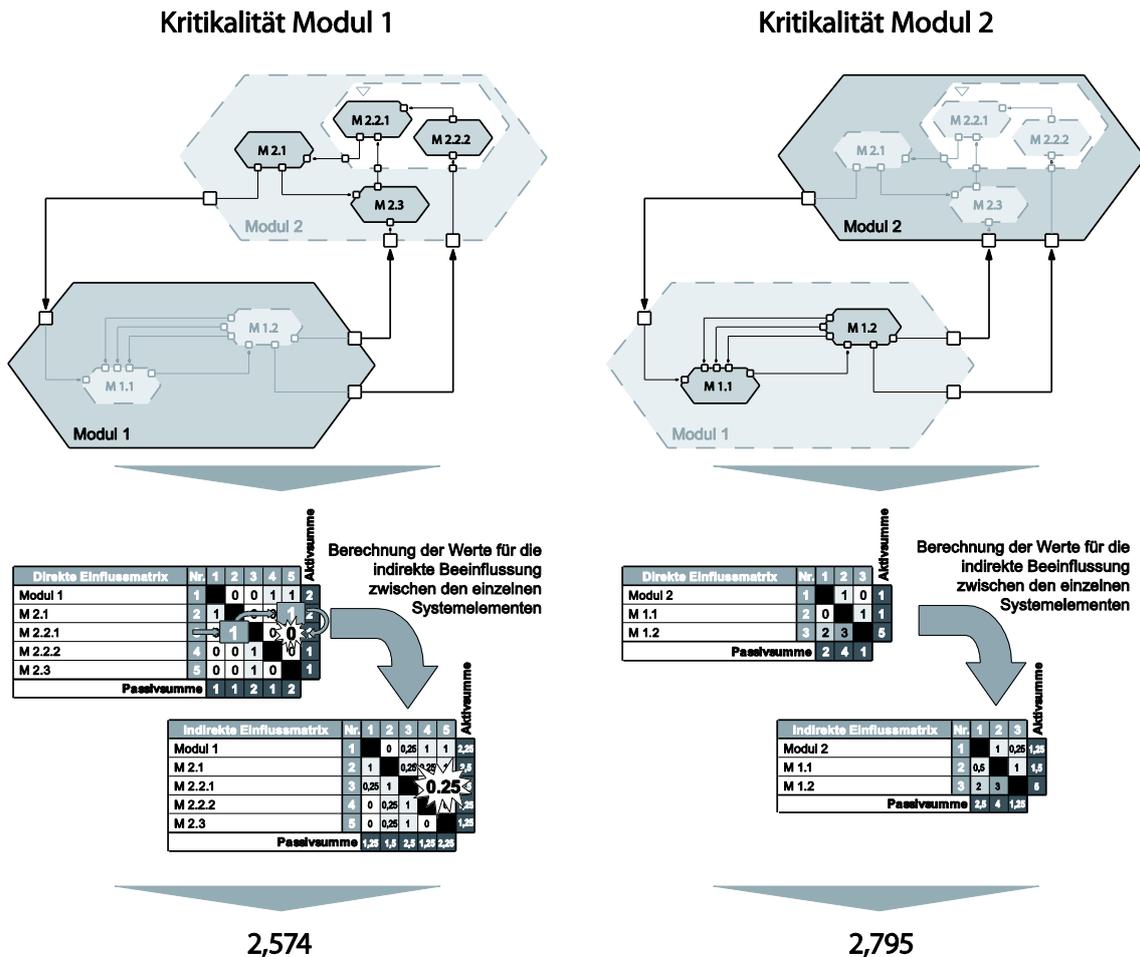


Bild A-2 Berechnung des Dynamikindex zweier Module

A2.2 Systemstrukturanalyse – funktionsorientierte Reihenfolgebestimmung fachdisziplinübergreifender Integrations- und Validierungsaktivitäten

Bei fortgeschrittenen mechatronischen Systemen wird der Großteil der Systemfunktionalität durch das synergetische Zusammenwirken der Elemente verschiedener Fachdisziplinen realisiert. Um sicher zu gehen, dass die Funktionalität vollständig gegeben ist, reicht es nicht aus, die Funktionsfähigkeit der einzelnen Elemente für sich zu validieren. Ergänzend muss auch das Zusammenspiel der Elemente aus den verschiedenen Fachdisziplinen validiert werden. In welcher Konstellation und Reihenfolge die Elemente

dabei zusammen getestet werden müssen, ist dadurch festgelegt, wie sie bei der Erfüllung der Gesamtsystemfunktionalität zusammenspielen. Nachfolgend wird ein Verfahren beschrieben, wie anhand der Wirkstruktur, der Funktionshierarchie und den Verweisen zwischen beiden die Reihenfolge für fachdisziplinübergreifende Validierungsaktivitäten bestimmt werden kann.

Ausgangsbasis des Verfahrens sind die mit Hilfe der Spezifikationstechnik des SFB 614⁵⁰ (vgl. [GFD+08a, S. 91ff.]) entsprechend dem Beispiel in Bild A-3 erstellte Wirkstruktur und Funktionshierarchie des zu analysierenden Systems als auch die Verweise zwischen beiden. Die Gesamtfunktion des Systems wurde solange hierarchisch in Teilfunktionen zerlegt, bis zu jeder Teilfunktion ein oder mehrere Systemelemente gefunden werden können, die diese Teilfunktion realisieren. Die zuletzt gefundenen Teilfunktionen bilden die Blätter der Funktionshierarchie. Die die Teilfunktionen realisierenden Systemelemente sind in der Wirkstruktur abgebildet und ebenfalls hierarchisch strukturiert. Die Systemelemente auf der ersten Ebene des Systems bilden dessen Module ab. Diese bestehen aus weiteren Systemelementen, von denen die Systemelemente auf der untersten Ebene über Verweise mit den Teilfunktionen verbunden sind, zu deren Realisierung sie beitragen. Die hierarchischen Strukturen von Funktionshierarchie und Wirkstruktur stimmen in der Regel nicht überein.

Im ersten Schritt wird das Element in der Wirkstruktur festgelegt, für das die Validierungsaktivitäten geplant werden sollen. Im Beispiel ist dies das System selbst. Für jedes Systemelement auf der ersten Ebene werden die Verweise auf die, durch die in diesem Systemelement beinhalteten Elemente realisierten Teilfunktionen, auf das Systemelement an sich bezogen. Im Beispiel werden also die Verweise auf die, durch die Systemelemente eines Moduls realisierten Teilfunktionen, auf das Modul selbst gezogen. So wird mit jedem Modul des Systems verfahren.

Anschließend werden an jede Teilfunktion der Funktionshierarchie das oder die sie nun realisierenden Systemelemente als n-Tupel annotiert. Im Beispiel werden die Modulnummern an die Teilfunktionen geschrieben. Nun wird analysiert, ob eine Funktion durch mindestens zwei oder mehr Systemelemente realisiert wird. Ist dies der Fall werden diese Tupel für die unterste Ebene der Funktionshierarchie vermerkt. Im Beispiel erfüllen Modul 1 und Modul 2 zusammen eine Teilfunktion der untersten Ebene der Funktionshierarchie. Das Tupel {1,2} wird entsprechend für Ebene 2 vermerkt. Alle übrigen Funktionen werden jeweils durch nur ein Modul realisiert. Im Folgenden werden die einzelnen Äste der Funktionshierarchie ausgehend von den Teilfunktionen der untersten Ebene von unten nach oben untersucht. An jedem Knoten respektive jeder aus mindestens zwei Teilfunktionen bestehenden Funktion wird ermittelt, ob die Elemente der bisherigen Tupel der Teilfunktionen disjunkt sind oder nicht. Sind sie disjunkt wird

⁵⁰ Die Modellierung kann alternativ auch mit den Modellierungssprachen UML oder SysML erfolgen. Für beide Sprachen können Profile entsprechend dem Metamodell der Spezifikationstechnik des SFB 614 erstellt werden.

eine neues Tupel gebildet, das die disjunkten Mengen vereinigt. Dieses Tupel wird für die Ebene der Funktionshierarchie, auf der sich der betrachtete Knoten befindet, vermerkt. Sind die Tupel nicht disjunkt, wird das die größte Menge von Systemelementen beinhaltende Tupel am Knoten annotiert und für die Ebene vermerkt. Durch die beschriebene Vorgehensweise ergeben sich für die einzelnen Ebenen der Funktionshierarchie verschiedene Tupel, die anzeigen welche Systemelemente zusammen auf gemeinsame Funktionserfüllung geprüft werden sollten. Sind auf verschiedenen Ebenen identische Tupel vorhanden, werden die der unteren Ebenen gelöscht. Die Reihenfolge auszuführender Validierungsaktivitäten ist den Ebenen der Funktionshierarchie von unten nach oben zu entnehmen. Im Beispiel werden zuerst Modul 1 und 2 sowie Modul 1 und 3 zusammen geprüft und dann alle drei Module zusammen.

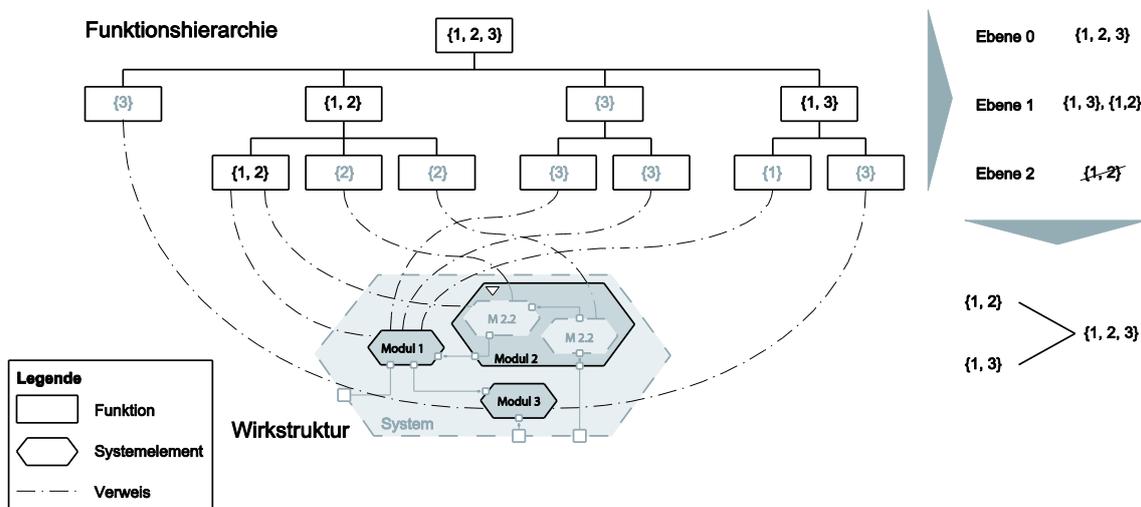


Bild A-3 Bestimmung der Reihenfolge von fachdisziplinübergreifenden Validierungsaktivitäten

A3 Prozessbausteine für den Entwicklungsprozess fortschrittlicher mechatronischer Systeme

Das in Kapitel 4.3.2.1 vorgestellte Verfahren zur systemmodellbasierten Prozessplanung fußt auf einer Wissensbasis, in der verschiedene Prozessbausteine zur Generierung problemangepasster Entwicklungsprozesse abgelegt sind. Die nachfolgenden Tabellen A-1, A-2, A-3 und A-4 listen die Entwicklungstätigkeiten und ihre Ein- und Ausgangsobjekte auf, die für die vorliegende Arbeit als Prozessbausteine entwickelt wurden. Sie basieren auf den in Kapitel 2.4.2.2 in Tabelle 2-2 zusammengefassten fachdisziplinspezifischen Vorgehensweisen. Die Sammlung versteht sich als Beispielsammlung, die bedarfsspezifisch erweitert oder angepasst werden kann bzw. muss.

Die Tabellen beinhalten die Grundbausteine, modulinterne Abstimmungsbausteine und modulübergreifende Abstimmungsbausteine (siehe auch Abschnitt 4.3.1.2). Für die Grundbausteine und die modulinternen Abstimmungsbausteine sind die Klassen von mechatronischen Systemen, die Fachdisziplinen und ggf. die Spezialisierungen angegeben, für die sie anzuwenden sind. Für die modulübergreifenden Abstimmungsbausteine gibt Tabelle A-4 die Prozessregeln an, wann diese in den Entwicklungsprozess zu integrieren sind. Zuordnung der Grundbausteine und modulinternen Abstimmungsbausteine sowie die Prozessregeln der modulübergreifenden Abstimmungsbausteine sind in Kurzform dargestellt, aus denen für die Implementierung der Prozessstränge im Prozessplaner (vgl. Abschnitt 5.1.1) die Langform der exakten Anwendungsbedingungen abgeleitet wurden.

Tabelle A-1 Grundbausteine für die Fachdisziplinen Maschinenbau, Elektrik, Elektronik, Regelungstechnik und Softwaretechnik

Nr.	Eingangsobjekt	Entwicklungstätigkeit	Ausgangsobjekt	Attribute		
				Zuordnung zu Klasse	Zuordnung zu Fachdisziplin	Zuordnung zu Spezialisierung
1	Prinziplösung \wedge Lösungskonzept auf Modulebene	Kinematik verfeinern und analysieren	Liste der Bewegungsmöglichkeiten und Stellwege	MKS	MB	
2	(Liste der Bewegungsmöglichkeiten und Stellwege \wedge Prinziplösung) \vee Kinematikmodell des Systems	Kräfte überschlägig bestimmen	statische Kräfte	MKS \vee IMS	MB	
3	statische Kräfte \wedge Prinziplösung	Werkstoffe wählen	Liste einzusetzender Werkstoffe	MKS \vee IMS	MB	
4	Liste einzusetzender Werkstoffe \wedge Prinziplösung	Gestalt der Hauptfunktionsträger maßlich und räumlich verfeinern	verfeinerte Gestalt der Hauptfunktionsträger	MKS \vee IMS	MB	
5	(verfeinerte Gestalt der Hauptfunktionsträger \wedge Prinziplösung abgestimmte) \vee mechanische Schnittstellen	Gestalt der Nebenfunktionsträger maßlich und räumlich verfeinern	verfeinerte Gestalt der Nebenfunktionsträger	MKS \vee IMS	MB	
6	(verfeinerte Gestalt der Hauptfunktionsträger \wedge verfeinerte Gestalt der Nebenfunktionsträger) \vee Gestaltmodell des Gesamtsystems	Durchdringungsanalyse durchführen (opt.)	Ergebnis Durchdringungsanalyse	MKS \vee IMS	MB	

Nr.	Eingangsobjekt	Entwicklungstätigkeit	Ausgangsobjekt	Attribute		
				Zuordnung zu Klasse	Zuordnung zu Fachdisziplin	Zuordnung zu Spezialisierung
7	(Liste der Bewegungsmöglichkeiten und Stellwege \wedge verfeinerte Gestalt der Hauptfunktionsträger \wedge verfeinerte Gestalt der Nebenfunktionsträger \wedge Prinziplösung) \vee (Liste der Bewegungsmöglichkeiten und Stellwege \wedge verfeinerte Gestalt der Hauptfunktionsträger \wedge verfeinerte Gestalt der Nebenfunktionsträger \wedge Prinziplösung \wedge Ergebnis Durchdringungsanalyse) \vee (Liste der Bewegungsmöglichkeiten und Stellwege \wedge verfeinerte Gestalt der Hauptfunktionsträger \wedge verfeinerte Gestalt der Nebenfunktionsträger \wedge Prinziplösung \wedge Ergebnis Durchdringungsanalyse \wedge Gestaltmodell des Gesamtsystems)	Dynamikmodell erstellen	Dynamikmodell	MKS	MB	
8	Dynamikmodell \vee Dynamikmodell auf Systemebene	dynamische Kräfte ermitteln	dynamische Kräfte	MKS	MB	
9	dynamische Kräfte	Aktorenergie bestimmen	benötigte Aktorenergien	MKS	MB	Hydraulik \vee Pneumatik
10	benötigte Aktorenergien \wedge Prinziplösung	Aktoren auswählen	Datenblätter der Aktoren	MKS	MB	Hydraulik \vee Pneumatik
11	Datenblätter der Aktoren \wedge Prinziplösung	Versorgungskomponente auswählen	Datenblätter der Versorgungskomponenten	MKS	MB	Hydraulik \vee Pneumatik
12	Dynamikmodell mit Aktorik und Sensorik	Hydraulikplan erstellen	Hydraulikplan	MKS	MB	Hydraulik
13	Dynamikmodell mit Aktorik und Sensorik	Pneumatikplan erstellen	Pneumatikplan	MKS	MB	Pneumatik
14	(dynamische Kräfte \vee Dynamikmodell mit interner Aktorik und Sensorik) \vee (Dynamikmodell mit gesamter Aktorik und Sensorik auf Systemebene)	Kollisionsanalyse durchführen (opt.)	Ergebnis Kollisionsanalyse	MKS	MB	
15	dynamische Kräfte \vee Dynamikmodell mit interner Aktorik und Sensorik \vee Ergebnis der Kollisionsanalyse \vee Erweiterte Gestaltmodelle der Haupt- und Nebenfunktionsträger \vee Dynamikmodell mit gesamter Aktorik und Sensorik auf Systemebene	Feingestalten der Hauptfunktionsträger	Ausgestaltete Hauptfunktionsträger	MKS \vee IMS	MB	
16	Ausgestaltete Hauptfunktionsträger	Feingestalten der Nebenfunktionsträger	Ausgestaltete Nebenfunktionsträger	MKS \vee IMS	MB	
17	Ausgestaltete Hauptfunktionsträger \wedge ausgestaltete Nebenfunktionsträger \wedge abgestimmte ausgestaltete mechanische Schnittstellen	FEM-Analyse durchführen (opt.)	Ergebnisse FEM-Analyse	MKS \vee IMS	MB	
18	(Ausgestaltete Hauptfunktionsträger \wedge Ausgestaltete Nebenfunktionsträger) \vee (Ausgestaltete Hauptfunktionsträger \wedge Ausgestaltete Nebenfunktionsträger \wedge Ergebnisse FEM-Analyse) \vee (Ausgestaltete Hauptfunktionsträger \wedge Ausgestaltete Nebenfunktionsträger \wedge Aufeinander abgestimmte Gestaltmodelle)	Erarbeiten der Einzelteilzeichnungen	Einzelteilzeichnungen	MKS \vee IMS	MB	
19	Einzelteilzeichnungen	Erarbeiten der Gruppenzeichnungen des Moduls	Gruppenzeichnungen	MKS \vee IMS	MB	
20	Gruppenzeichnungen	Modulprototyp konstruieren (opt.)	Konstruktionszeichnungen Modulprototyp	MKS \vee IMS	MB	
21	Konstruktionszeichnungen Modulprototyp \wedge Prinziplösung	Modulprüfstand konstruieren (opt.)	Konstruktionszeichnungen Modulprüfstand	MKS \vee IMS	MB	
22	Konstruktionszeichnungen Modulprototyp	Modulprototyp erstellen (opt.)	Modulprototyp	MKS \vee IMS	MB	
23	Konstruktionszeichnungen Modulprüfstand	Modulprüfstand erstellen (opt.)	Modulprüfstand	MKS \vee IMS	MB	
24	Modulprüfstand	Modulprüfstand in Betrieb nehmen (opt.)	betriebsbereiter Modulprüfstand	MKS \vee IMS	MB	
25	betriebsbereiter Modulprüfstand \wedge Modulprototyp	Test durchführen (opt.)	Testergebnisse Modulprototypentest	MKS \vee IMS	MB	
26	Testergebnisse Modulprototypentest	Test auswerten und Modul verbessern (opt.)	ausgewerteter Modulprototypentest	MKS \vee IMS	MB	
27	Gruppenzeichnungen \wedge ausgewerteter Modulprototypentest	Detaillieren und festlegen der Einzelteile	detaillierte Gestalt der Einzelteile	MKS \vee IMS	MB	

Nr.	Eingangsobjekt	Entwicklungstätigkeit	Ausgangsobjekt	Attribute		
				Zuordnung zu Klasse	Zuordnung zu Fachdisziplin	Zuordnung zu Spezialisierung
28	Einzelteilzeichnungen \wedge Gruppenzeichnungen \wedge Datenblätter der Aktoren \wedge Datenblätter der Sensoren \wedge Datenblätter der Versorgungskomponenten	Erarbeiten der Modulstückliste	Modulstückliste	MKS $\underline{\vee}$ IMS	MB	
29	Kommunikationsinfrastruktur auf Modulebene \vee Kommunikationsinfrastruktur auf Systemebene	Kommunikationschnittstellen festlegen	Kommunikationschnittstellen	MKS $\underline{\vee}$ IMS	ET	
30	vorl. Schaltungsplan \wedge Kommunikationschnittstellen	Schaltungsplan erweitern	erweiterter Schaltungsplan	MKS $\underline{\vee}$ IMS	ET	
31	vorl. Stromlaufplan \wedge Kommunikationschnittstellen	Stromlaufplan erweitern	erweiterter Stromlaufplan	MKS $\underline{\vee}$ IMS	ET	
32	Kommunikationstopologie des Moduls \wedge Kommunikationschnittstellen \wedge erweiterter Schaltungsplan \wedge erweiterter Stromlaufplan	Anforderung an die Elektronik detaillieren	Pflichtenheft der Elektronik	MKS $\underline{\vee}$ IMS	ET	
33	Pflichtenheft der Elektronik	Elektronik funktional beschreiben	Funktionsblöcke der Elektronik	MKS $\underline{\vee}$ IMS	ET	
34	Funktionsblöcke der Elektronik	Ein- und Ausgangsverhalten funktional modellieren	imperative Beschreibung des Ein- und Ausgangsverhaltens	MKS $\underline{\vee}$ IMS	ET	
35	imperative Beschreibung des Ein- und Ausgangsverhaltens	Register-Transfer-Ebene beschreiben	mit Makros der Digitaltechnik beschriebene Register-Transfer- Ebene	MKS $\underline{\vee}$ IMS	ET	
36	mit Makros der Digitaltechnik beschriebene Register-Transfer-Ebene	Binäre Logik der Elektronik spezifizieren	mit Registern und Gattern beschriebene Schaltungslogik	MKS $\underline{\vee}$ IMS	ET	
37	mit Registern und Gattern beschriebene Subblöcke	Logik verifizieren (opt.)	verifizierte Schaltungslogik	MKS $\underline{\vee}$ IMS	ET	
38	mit Registern und Gattern beschriebene Schaltungslogik $\underline{\vee}$ verifizierte Schaltungslogik	Logik auf Transistorebene herunterbrechen	Strukturbeschreibung auf Transistorebene	MKS $\underline{\vee}$ IMS	ET	
39	Strukturbeschreibung auf Transistorebene	Schaltungslayout erstellen	Schaltungslayout	MKS $\underline{\vee}$ IMS	ET	
40	Schaltungslayout	Layout verifizieren (opt.)	verifiziertes Schaltungslayout	MKS $\underline{\vee}$ IMS	ET	
41	Schaltungslayout $\underline{\vee}$ verifiziertes Schaltungslayout	benötigte Energien bestimmen	benötigte Energie der Elektronik	MKS $\underline{\vee}$ IMS	ET	
42	benötigte Energie der Elektronik	el. Versorgungskomponenten auswählen	benötigte el. Versorgungskomponenten	MKS $\underline{\vee}$ IMS	ET	
43	Schaltungslayout \wedge benötigte Versorgungskomponenten \wedge finaler Schaltungsplan \wedge finaler Stromlaufplan	Elektronikprototyp erstellen (opt.)	Prototyp der Elektronik	MKS $\underline{\vee}$ IMS	ET	
44	Prototyp der Elektronik	Elektronikprototyp testen (opt.)	getesteter Prototyp der Elektronik	MKS $\underline{\vee}$ IMS	ET	
45	(Schaltungslayout \wedge benötigte Versorgungskomponenten \wedge finaler Schaltungsplan \wedge finaler Stromlaufplan) $\underline{\vee}$ (Schaltungslayout \wedge benötigte Versorgungskomponenten \wedge finaler Schaltungsplan \wedge finaler Stromlaufplan \wedge getesteter Prototyp der Elektronik) $\underline{\vee}$ (Schaltungslayout \wedge benötigte Versorgungskomponenten \wedge finaler Schaltungsplan \wedge finaler Stromlaufplan \wedge abgeschlossene modulare Integrationstests)	Fertigungsunterlagen der Elektronik erstellen	Fertigungsunterlagen der Elektronik	MKS $\underline{\vee}$ IMS	ET	
46	Modulintern abgestimmte Kinematik \wedge Dynamische Kräfte	Antriebsart auswählen	Antriebsart	MKS $\underline{\vee}$ IMS	EL	Antriebs- technik
47	Antriebsart \wedge Dynamische Kräfte	Nennleistung bestimmen	Nennleistung des Antriebs	MKS $\underline{\vee}$ IMS	EL	Antriebs- technik
48	Antriebsart \wedge Nennleistung des Antriebs	elektrische Aktorik auswählen	Motortyp	MKS $\underline{\vee}$ IMS	EL	Antriebs- technik
49	(Nennleistung des Antriebs \wedge Motortyp) $\underline{\vee}$ ausgelegter el. Speicher	Überprüfen der thermischen Belastbarkeit (opt)	thermische Belastbarkeit	MKS $\underline{\vee}$ IMS	EL	Antriebs- technik
50	(Nennleistung des Antriebs \wedge Motortyp) $\underline{\vee}$ (Nennleistung des Antriebs \wedge Motortyp \wedge thermische Belastbarkeit)	elektrische Aktorik dimensionieren	Datenblätter des el. Motors	MKS $\underline{\vee}$ IMS	EL	Antriebs- technik
51	Datenblätter des Motors $\underline{\vee}$ ausgelegter el. Speicher	Leistungselektronik definieren	Leistungselektronik des el. Motors	MKS $\underline{\vee}$ IMS	EL	
52	Datenblätter des el. Motors \wedge Leistungselektronik des Motors \wedge finaler Schaltungsplan \wedge finaler Stromlaufplan	Elektronikprototyp erstellen (opt.)	Prototyp der Elektrik	MKS $\underline{\vee}$ IMS	EL	

Nr.	Eingangsobjekt	Entwicklungstätigkeit	Ausgangsobjekt	Attribute		
				Zuordnung zu Klasse	Zuordnung zu Fachdisziplin	Zuordnung zu Spezialisierung
53	Prototyp der Elektrik	Elektrikprototyp testen (opt.)	getesteter Prototyp der Elektrik	MKS <u>✓</u> IMS	EL	
54	(Datenblätter des el. Motors \wedge Leistungselektronik des Motors \wedge finaler Schaltungsplan \wedge finaler Stromlaufplan) <u>✓</u> (Datenblätter des el. Motors \wedge Leistungselektronik des Motors \wedge finaler Schaltungsplan \wedge finaler Stromlaufplan \wedge getesteter Prototyp der Elektrik) <u>✓</u> (Datenblätter des el. Motors \wedge Leistungselektronik des Motors \wedge finaler Schaltungsplan \wedge finaler Stromlaufplan \wedge abgeschlossene modulare Integrationstests Elektrik und Mechanik)	Fertigungsunterlagen der Elektrik erstellen	Fertigungsunterlagen der Elektrik	MKS <u>✓</u> IMS	EL	
55	benötigte Leistungs- und Energiedichte	Elektrische Speichermedien dimensionieren	ausgelegter el. Speicher	MKS <u>✓</u> IMS	EL	Speichertechnik
56	Prinziplösung \wedge Lösungskonzept auf Modulebene	Reglerfunktionen des Moduls extrahieren	Reglerfunktionen des Moduls	MKS <u>✓</u> IMS	RT	
57	Prinziplösung \vee Lösungskonzept auf Modulebene	Art der Aktorik und Sensorik der Reglerfunktionen des Moduls extrahieren	Art der Aktorik und Sensorik der Reglerfunktionen	MKS <u>✓</u> IMS	RT	
58	Prinziplösung \wedge Kommunikationstopologie des Moduls \wedge Reglerfunktionen des Moduls \wedge Art der Aktorik und Sensorik der Reglerfunktionen	Hierarchie der Reglerfunktionen des Moduls festlegen	Hierarchie der Reglerfunktionen	MKS <u>✓</u> IMS	RT	
59	Prinziplösung \wedge Hierarchie der Reglerfunktionen	Grundanforderungen der Regelung bestimmen (Stabilität, stationäre Genauigkeit etc)	Grundanforderungen der Regleraufgabe	MKS <u>✓</u> IMS	RT	
60	Prinziplösung \wedge Grundanforderungen der Regleraufgabe	technische Randbedingungen ermitteln (z.B vorgegebene Beschränkungen, Störbereich, Regelbereich, Führungsbereich)	technische Randbedingungen der Regleraufgabe	MKS <u>✓</u> IMS	RT	
61	Prinziplösung \wedge technische Randbedingungen der Regleraufgabe	qualitative Anforderungen an die Dynamik bestimmen	qualitative Anforderungen an die Dynamik der Regelung	MKS <u>✓</u> IMS	RT	
62	(Dynamikmodell mit interner Aktorik und Sensorik \wedge technische Randbedingungen der Regleraufgabe) <u>✓</u> (Prinziplösung \wedge technische Randbedingungen der Regleraufgabe)	Ersatzmodell der Regelstrecke erstellen	Ersatzmodell der Regelstrecke	MKS <u>✓</u> IMS	RT	
63	Ersatzmodell der Regelstrecke	quantitative Anforderungen an die Dynamik der Regelung ermitteln	quantitative Anforderungen an die Dynamik der Regelung	MKS <u>✓</u> IMS	RT	
64	Grundanforderungen der Regleraufgabe \wedge technische Randbedingungen der Regleraufgabe \wedge qualitative Anforderungen an die Dynamik der Regelung \wedge Ersatzmodell der Regelstrecke \wedge quantitative \wedge Anforderungen an die Dynamik der Regelung Aufteilung diskret-kontinuierliches Verhalten	Reglerstrukturen festlegen	Reglerstruktur	MKS <u>✓</u> IMS	RT	
65	Reglerstruktur \wedge auf Systemebene abgestimmte Reglerstruktur	Reglertyp wählen	Reglertyp	MKS <u>✓</u> IMS	RT	
66	Ersatzmodell der Regelstrecke \wedge Reglerstruktur \wedge Reglertyp	Parameter für Regler bestimmen	Prototyp der Regelung	MKS <u>✓</u> IMS	RT	
67	An Sensorparameter angepasster Prototyp der Regelung \wedge HW-SW Liste der Regelung	Reglercode generieren	Reglercode	MKS <u>✓</u> IMS	RT	
68	Prinziplösung \wedge Lösungskonzept auf Modulebene	Komponentenstruktur des Moduls extrahieren	Komponentenstruktur des Moduls	MKS <u>✓</u> IMS	SW	
69	Prinziplösung \wedge Komponentenstruktur des Moduls	modulspezifisches Verhalten extrahieren	modulspezifisches Verhalten	MKS <u>✓</u> IMS	SW	
70	Prinziplösung \wedge modulspezifisches Verhalten	Use Cases des Moduls konkretisieren	Use Cases des Moduls	MKS <u>✓</u> IMS	SW	
71	Use Cases des Moduls	Strukturierung der Use Cases	Strukturierte Use Cases des Moduls	MKS <u>✓</u> IMS	SW	
72	Komponentenstruktur des Moduls \wedge Kommunikationstopologie des Moduls \wedge auf Systemebene abgestimmte Kommunikationstopologie	Komponentenstruktur des Moduls verfeinern	verfeinerte Komponentenstruktur des Moduls	MKS <u>✓</u> IMS	SW	
73	modulspezifisches Verhalten \wedge strukturierte Use Cases des Moduls	Zustände verfeinern	verfeinerte Zustände des Moduls	MKS <u>✓</u> IMS	SW	

Nr.	Eingangsobjekt	Entwicklungstätigkeit	Ausgangsobjekt	Attribute		
				Zuordnung zu Klasse	Zuordnung zu Fachdisziplin	Zuordnung zu Spezialisierung
74	modulspezifisches Verhalten \wedge strukturierte Use Cases des Moduls \wedge verfeinerte Zustände des Moduls	Aktivitäten verfeinern	verfeinerte Aktivitäten des Moduls	MKS $\underline{\vee}$ IMS	SW	
75	modulspezifisches Verhalten \wedge strukturierte Use Cases des Moduls \wedge verfeinerte Zustände des Moduls \wedge Sequenzdiagramme des Moduls	Sequenzen festlegen	Sequenzdiagramme des Moduls	MKS $\underline{\vee}$ IMS	SW	
76	modulspezifisches Verhalten \wedge verfeinerte Zustände des Moduls \wedge Sequenzdiagramme des Moduls	Zeitverhalten festlegen	vorgegebenes Zeitverhalten der Software	MKS $\underline{\vee}$ IMS	SW	
77	Use Cases des Moduls \wedge verfeinerte Zustände des Moduls \wedge vorgegebenes Zeitverhalten der Software	Design formal verifizieren (opt)	verifiziertes Design	MKS $\underline{\vee}$ IMS	SW	
78	(vorgegebenes Zeitverhalten der Software \wedge Reglercode) $\underline{\vee}$ (verifiziertes Design \wedge Reglercode)	Design implementieren	implementiertes Design	MKS $\underline{\vee}$ IMS	SW	
79	implementiertes Design	Testcode generieren (opt)	Testcode	MKS $\underline{\vee}$ IMS	SW	
80	Testcode	Testcode simulieren (opt)	getestetes Design	MKS $\underline{\vee}$ IMS	SW	
81	implementiertes Design $\underline{\vee}$ getestetes Design	Finalen Code generieren	Finaler Code	MKS $\underline{\vee}$ IMS	SW	

Tabelle A-2 Modulinterne Abstimmungsbausteine für die Fachdisziplinen Maschinenbau, Elektrik, Elektronik, Regelungstechnik und Softwaretechnik

Nr.	Eingangsobjekt	Entwicklungstätigkeit	Ausgangsobjekt	Attribute		
				Zuordnung zu Klasse	Zuordnung zu Fachdisziplin	Zuordnung zu Spezialisierung
1	modularisiertes grobes Lösungskonzept auf Systemebene	Entwicklungsaufgabe für das Modul analysieren	grobe Anforderungsliste für das Modul \wedge geklärte Aufgabenstellung für das Modul	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
2	grobe Anforderungsliste für das Modul \wedge geklärte Aufgabenstellung für das Modul	Umfeld des Moduls analysieren	Umfeldmodell des Moduls	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
3	Umfeldmodell des Moduls \wedge geklärte Aufgabenstellung	Anwendungsszenarien des Moduls festlegen	Anwendungsszenarien des Moduls	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
4	Anwendungsszenarien des Moduls \wedge grobe Anforderungsliste für das Modul	Grundstrukturtyp der Entwicklungsaufgabe des Moduls	Grundtyp der Entwicklungsaufgabe für das Modul	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee	
5	Grundtyp der Entwicklungsaufgabe für das Modul	relevante Entwurfsregeln für das Modul bestimmen	relevante Entwurfsregeln für das Modul	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
6	relevante Entwurfsregeln für das Modul \wedge grobe Anforderungsliste für das Modul \wedge geklärte Aufgabenstellung für das Modul \wedge Anwendungsszenarien des Moduls	Anforderungsliste mit Modulanforderungen erweitern	mit den Anforderungen des Moduls erweiterte Anforderungsliste	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
7	mit den Anforderungen des Moduls erweiterte Anforderungsliste	Funktionshierarchie mit Modulfunktionen erweitern	Funktionshierarchie für das Modul	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
8	Funktionshierarchie für das Modul \wedge Anwendungsszenarien des Moduls	Funktionshierarchie anwendungsszenariospezifisch für Modul modifizieren	für Modul anwendungsszenariospezifisch erweiterte Funktionshierarchien	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
9	für Modul anwendungsszenariospezifisch erweiterte Funktionshierarchien	potentielle Lösungsmuster für Modul suchen	potentielle Lösungsmuster für die Anwendungsszenarien des Moduls	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	

Nr.	Eingangsobjekt	Entwicklungstätigkeit	Ausgangsobjekt	Attribute		
				Zuordnung zu Klasse	Zuordnung zu Fachdisziplin	Zuordnung zu Spezialisierung
10	für Modul anwendungsszenariospezifisch erweiterte Funktionshierarchien	potentielle Lösungselemente für Modul suchen	potentielle Lösungselemente für die Anwendungsszenarien des Moduls	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
11	potentielle Lösungsmuster für die Anwendungsszenarien des Moduls \wedge potentielle Lösungselemente für die Anwendungsszenarien des Moduls	Lösungsvarianten je Anwendungsszenario des Moduls generieren	Lösungsvarianten je Anwendungsszenario des Moduls	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
12	Lösungsvarianten je Anwendungsszenario des Moduls	Bewertung und Auswahl einer Lösungsvariante je Anwendungsszenario des Moduls	Lösungsvariante je Anwendungsszenario des Moduls	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
13	Lösungsvariante je Anwendungsszenario des Moduls	Wirkstruktur pro Lösungsvariante für Modul aufstellen	Wirkstruktur je Anwendungsszenario des Moduls	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
14	Lösungsvariante je Anwendungsszenario des Moduls \wedge Wirkstruktur je Anwendungsszenario des Moduls	Gestalt pro Lösungsvariante des Moduls grob entwerfen	grobe Gestalt je Anwendungsszenario des Moduls	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
15	Lösungsvariante je Anwendungsszenario des Moduls	grundsätzliche Betriebsmodi des Moduls pro Lösungsvariante festlegen	grundsätzliche Betriebsmodi je Anwendungsszenario des Moduls	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
16	Lösungsvariante je Anwendungsszenario des Moduls \wedge grundsätzliche Betriebsmodi je Anwendungsszenario des Moduls	Ablaufverhalten des Moduls pro Lösungsvariante grob festlegen	grobes Ablaufverhalten je Anwendungsszenario des Moduls	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
17	Lösungsvariante je Anwendungsszenario des Moduls \wedge Wirkstruktur je Anwendungsszenario des Moduls \wedge grobe Gestalt je Anwendungsszenario des Moduls \wedge grundsätzliche Betriebsmodi je Anwendungsszenario des Moduls \wedge grobes Ablaufverhalten je Anwendungsszenario des Moduls	Lösung je Anwendungsszenario des Moduls auswählen	Lösung je Anwendungsszenario des Moduls	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
18	Lösung je Anwendungsszenario des Moduls	Lösungen zu Lösungskonzepten auf Modulebene zusammenführen	Lösungskonzepte auf Modulebene	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
19	Lösungskonzepte auf Modulebene	Bewertung und Auswahl der Lösungskonzepte auf Modulebene	Lösungskonzept auf Modulebene	MKS $\underline{\vee}$ IMS	MB \vee RT \vee SW \vee EL \vee ET	
20	Dynamikmodell mit interner Aktorik und Sensorik \vee Dynamikmodell mit gesamter Aktorik und Sensorik \vee benötigte el. Versorgungskomponenten \vee ausgelegter el. Speicher	Gestaltmodell um neue Komponenten erweitern	erweiterte Gestaltmodelle der Haupt- und Nebenfunktionsträger	MKS $\underline{\vee}$ IMS	MB \vee ET \vee EL	
21	Prinziplösung \wedge Lösungskonzept auf Modulebene Strukturierte \wedge Use Cases des Moduls	Kommunikationstopologie des Moduls extrahieren	Kommunikationstopologie des Moduls	MKS $\underline{\vee}$ IMS	ET \wedge EL \wedge SW	
22	Prinziplösung \wedge Kommunikationstopologie des Moduls \wedge auf Systemebene abgestimmte Kommunikationstopologie	vorl. Schaltungsplan erstellen	vorl. Schaltungsplan	MKS $\underline{\vee}$ IMS	ET \wedge EL	
23	Prinziplösung \wedge Kommunikationstopologie des Moduls \wedge auf Systemebene abgestimmte Kommunikationstopologie	vorl. Stromlaufplan erstellen	vorl. Stromlaufplan	MKS $\underline{\vee}$ IMS	ET \wedge EL	
24	(vorl. Stromlaufplan \wedge vorl. Schaltungsplan) (vorl. Stromlaufplan \wedge vorl. Schaltungsplan \wedge Prototyp der Regelung)	Entscheiden über HW und SW	Aufteilung der HW- und SW-Komponenten auf Modulebene	MKS $\underline{\vee}$ IMS	ET \wedge EL \wedge RT \wedge SW	
25	Prinziplösung \wedge Aufteilung der HW- und SW-Komponenten auf Modulebene \wedge Aufteilung diskret-kontinuierliches Verhalten	Kommunikationsinfrastruktur auf Modulebene abstimmen	Kommunikationsinfrastruktur auf Modulebene	MKS $\underline{\vee}$ IMS	ET \wedge SW	
26	(Schaltungslayout \wedge benötigte Versorgungskomponenten \wedge finaler Schaltungsplan \wedge finaler Stromlaufplan \wedge finaler Code) \vee (Schaltungslayout \wedge benötigte Versorgungskomponenten \wedge finaler Schaltungsplan \wedge finaler Stromlaufplan \wedge getesteter Prototyp der Elektronik \wedge finaler Code)	modulinterne Integrationstest SW und HW durchführen	abgeschlossene modulinterne Integrationstests SW und HW	MKS $\underline{\vee}$ IMS	ET \wedge SW \wedge SW	
27	Liste der Bewegungsmöglichkeiten und Stellwege	Kinematik modulintern abstimmen	modulintern abgestimmte Kinematik	MKS $\underline{\vee}$ IMS	EL \wedge MB	Antriebs- technik

Nr.	Eingangsobjekt	Entwicklungstätigkeit	Ausgangsobjekt	Attribute		
				Zuordnung zu Klasse	Zuordnung zu Fachdisziplin	Zuordnung zu Spezialisierung
28	Datenblätter der Aktoren \wedge Dynamikmodell Datenblätter des el. Motors \wedge Leistungselektronik des el. Motors	Dynamikmodell um modulerne Aktorik erweitern und analysieren	Dynamikmodell mit interner Aktorik \wedge Anforderungen an die Sensorik	MKS	MB \wedge EL	Hydraulik \vee Pneumatik \vee Antriebs- technik
29	Anforderungen an die Sensorik \wedge Prinziplösung	Sensoren auswählen	Datenblätter der Sensoren	MKS	MB \wedge EL	Hydraulik \vee Pneumatik \vee Antriebs- technik
30	Dynamikmodell mit Aktorik \wedge Datenblätter der Sensoren	Dynamikmodell um modulerne Sensorik erweitern und analysieren	Dynamikmodell mit interner Aktorik und Sensorik	MKS	MB \wedge EL	Hydraulik \vee Pneumatik \vee Antriebs- technik
31	(erweiterter Schaltungsplan \wedge benötigte Versorgungskomponenten) \vee (erweiterter Schaltungsplan \wedge benötigte Versorgungskomponenten \wedge erweiterte Gestaltmodelle der Haupt- und Nebenfunktionsträger)	Schaltungsplan vervollständigen	finaler Schaltungsplan	MKS \vee IMS	ET \wedge EL	
32	(erweiterter Schaltungsplan \wedge benötigte Versorgungskomponenten) \vee (erweiterter Schaltungsplan \wedge benötigte Versorgungskomponenten \wedge erweiterte Gestaltmodelle der Haupt- und Nebenfunktionsträger)	Stromlaufplan vervollständigen	finaler Stromlaufplan	MKS \vee IMS	ET \wedge EL	
33	Konstruktionszeichnungen Modulprototyp \vee ausgewerteter Modulprototypentest \vee getesteter Prototyp der Elektrik \vee (Datenblätter des el. Motors \wedge Leistungselektronik des Motors \wedge finaler Schaltungsplan \wedge finaler Stromlaufplan)	modulinterne Integrationstest Elektrik und Mechanik durchführen (opt.)	abgeschlossene modulerne Integrationstests Elektrik und Mechanik	MKS \vee IMS	EL \wedge MB	Antriebs- technik
34	Sequenzdiagramme des Moduls	diskret-kontinuierliches Verhalten abstimmen	Aufteilung diskret-kontinuierliches Verhalten	MKS \vee IMS	SW \wedge RT	
35	abgeschlossene modulerne Integrationstests SW und HW \wedge abgeschlossene modulerne Integrationstests Elektrik und Mechanik	Modulfunktionstest durchführen (opt.)	abgeschlossener Modulfunktionstest	MKS \vee IMS	(MB \wedge EL) \wedge (ET \wedge SW \wedge RT)	
36	Modulstückliste \wedge Fertigungsunterlagen der Elektronik \wedge Fertigungsunterlagen der Elektrik \wedge Reglercode \wedge Finaler Code \wedge abgeschlossener Modulfunktionstest	Dokumentationen des Moduls zusammenführen	vollständige Dokumentation des Moduls	MKS \vee IMS	MB \wedge EL \wedge ET \wedge SW \wedge RT	

Tabelle A-3 Modulübergreifende Abstimmungsbausteine für die Fachdisziplinen Maschinenbau, Elektrik, Elektronik, Regelungstechnik und Softwaretechnik

Nr.	Eingangsobjekt	Entwicklungstätigkeit	Ausgangsobjekt	Prozessregel
1	Entwicklungsauftrag	Entwicklungsaufgabe analysieren	grobe Anforderungsliste \wedge geklärte Aufgabenstellung	immer
2	grobe Anforderungsliste \wedge geklärte Aufgabenstellung	Umfeldanalyse durchführen	Umfeldmodell	immer
3	Umfeldmodell \wedge geklärte Aufgabenstellung	Anwendungsszenarien auf Systemebene festlegen	Anwendungsszenarien des Systems	immer
4	Anwendungsszenarien des Systems \wedge grobe Anforderungsliste	Grundstrukturtyp der Entwicklungsaufgabe bestimmen	Grundtyp der Entwicklungsaufgabe	immer
5	Grundtyp der Entwicklungsaufgabe	relevante Entwurfsregeln bestimmen	relevante Entwurfsregeln	immer
6	relevante Entwurfsregeln \wedge grobe Anforderungsliste \wedge geklärte Aufgabenstellung \wedge Anwendungsszenarien des Systems	Anforderungsliste erstellen	Anforderungsliste auf Systemebene	immer
7	Anforderungsliste auf Systemebene	Funktionshierarchie erstellen	Funktionshierarchie auf Systemebene	immer
8	Funktionshierarchie auf Systemebene \wedge Anwendungsszenarien des Systems	Funktionshierarchie anwendungsszenariospezifisch modifizieren	anwendungsszenariospezifisch erweiterte Funktionshierarchien	immer
9	anwendungsszenariospezifisch erweiterte Funktionshierarchien	potentielle Lösungsmuster suchen	potentielle Lösungsmuster für die Anwendungsszenarien	immer
10	anwendungsszenariospezifisch erweiterte Funktionshierarchien	potentielle Lösungselemente suchen	potentielle Lösungselemente für die Anwendungsszenarien	immer

Nr.	Eingangsobjekt	Entwicklungstätigkeit	Ausgangsobjekt	Prozessregel
11	potentielle Lösungsmuster für die Anwendungsszenarien \wedge potentielle Lösungselemente für die Anwendungsszenarien	Lösungsvarianten je Anwendungsszenario generieren	Lösungsvarianten je Anwendungsszenario	immer
12	Lösungsvarianten je Anwendungsszenario	Bewertung und Auswahl von Lösungsvarianten je Anwendungsszenario	Lösungsvariante je Anwendungsszenario	immer
13	Lösungsvariante je Anwendungsszenario	grobe Wirkstruktur pro Lösungsvariante aufstellen	Wirkstruktur je Anwendungsszenario	immer
14	Lösungsvariante je Anwendungsszenario \wedge Wirkstruktur je Anwendungsszenario	Gestalt pro Lösungsvariante grob entwerfen	grobe Gestalt je Anwendungsszenario	immer
15	Lösungsvariante je Anwendungsszenario	grundsätzliche Betriebsmodi pro Lösungsvariante festlegen	grundsätzliche Betriebsmodi je Anwendungsszenario	immer
16	Lösungsvariante je Anwendungsszenario \wedge grundsätzliche Betriebsmodi je Anwendungsszenario	Ablaufverhalten pro Lösungsvariante grob festlegen	grobes Ablaufverhalten je Anwendungsszenario	immer
17	Lösungsvariante je Anwendungsszenario \wedge Wirkstruktur je Anwendungsszenario \wedge grobe Gestalt je Anwendungsszenario \wedge grundsätzliche Betriebsmodi je Anwendungsszenario \wedge grobes Ablaufverhalten je Anwendungsszenario	Lösung je Anwendungsszenario auswählen	Lösung je Anwendungsszenario	immer
18	Lösung je Anwendungsszenario	Lösungen zu Lösungskonzepten auf Systemebene zusammenführen	grobe Lösungskonzepte auf Systemebene	immer
19	grobe Lösungskonzepte auf Systemebene	Bewertung und Auswahl des Lösungskonzepts auf Systemebene	grobe Lösungskonzept auf Systemebene	immer
20	grobe Lösungskonzept auf Systemebene	Modularisierung auf Systemebene	modularisiertes grobes Lösungskonzept auf Systemebene	immer
21	Lösungskonzept auf Modulebene	Lösungskonzepte auf Modulebene zusammenführen	Lösungskonzepte auf Systemebene	immer
22	Lösungskonzepte auf Systemebene	Lösungskonzepte auf Systemebene bewerten und Prinziplösung bestimmen	Prinziplösung	immer
23	Dynamikmodell mit interner Aktorik und Sensorik \vee benötigte Aktorenergien	Dynamikmodelle bzgl. modulexterne Energien abstimmen und analysieren	Dynamikmodell mit gesamter Aktorik und Sensorik	siehe Tabelle A-4
24	ausgestaltete Hauptfunktionsträger	mechanische Schnittstellen abstimmen	abgestimmte mechanische Schnittstellen	siehe Tabelle A-4
25	Datenblätter des el. Motors \vee benötigte el. Versorgungskomponenten	Benötigte Leistungs- und Energiedichte bestimmen	benötigte Leistungs- und Energiedichte	siehe Tabelle A-4
26	(Modulstückliste \wedge Fertigungsunterlagen der Elektronik \wedge Fertigungsunterlagen der Elektrik \wedge Reglercode \wedge Finaler Code \wedge vollständige Dokumentation des Moduls) \vee abgeschlossener Modulintegrationstest \vee abgeschlossener Systemfunktionstest	Moduldokumentationen zusammenführen	vollständige Produktdokumentation auf Systemebene	immer
27	Liste der Bewegungsmöglichkeiten und Stellwege	Kinematik auf Systemebene abstimmen	Kinematikmodell des Systems	siehe Tabelle A-4
28	verfeinerte Gestalt der Nebenfunktionsträger	verfeinerte Gestalt auf Systemebene abstimmen	Gestaltmodell des Gesamtsystems	siehe Tabelle A-4
29	Dynamikmodell	Dynamikmodell auf Systemebene erstellen und analysieren	Dynamikmodell auf Systemebene	siehe Tabelle A-4
30	Dynamikmodell mit gesamter Aktorik und Sensorik	Dynamikmodell auf Systemebene erweitern und analysieren	Dynamikmodell mit gesamter Aktorik und Sensorik auf Systemebene	siehe Tabelle A-4
31	Ausgestaltete Nebenfunktionsträger	ausgestaltete mechanische Schnittstellen aufeinander abstimmen	abgestimmte ausgestaltete mechanische Schnittstellen	siehe Tabelle A-4
32	Kommunikationstopologie des Moduls	Kommunikationstopologie auf Systemebene abstimmen und analysieren	auf Systemebene abgestimmte Kommunikationstopologie	siehe Tabelle A-4
33	Kommunikationsinfrastruktur auf Modulebene	Kommunikationsinfrastruktur auf Systemebene abstimmen	auf Systemebene abgestimmte Kommunikationsinfrastruktur	siehe Tabelle A-4
34	Reglerstruktur	Regelstruktur auf Systemebene abstimmen	auf Systemebene abgestimmte Reglerstruktur	siehe Tabelle A-4
35	abgeschlossener Modulfunktionstest	modulübergreifenden Integrationstest durchführen	abgeschlossener Modulintegrationstest	siehe Tabelle A-4
36	abgeschlossener Modulfunktionstest \vee abgeschlossener Modulintegrationstest \vee ausgewerteter Modulprototypentest	Systemfunktionstest durchführen	abgeschlossener Systemfunktionstest	immer

A4 Situationspezifische Auswahl und Gewichtung von Entwicklungsprozesszielen

Das nachfolgende Bild A-4 gibt eine vollständige Übersicht über die Aktionen, Informationsflüsse und eingesetzten Methoden des Verfahrens zur situationspezifischen Auswahl und Gewichtung von Entwicklungsprozesszielen.

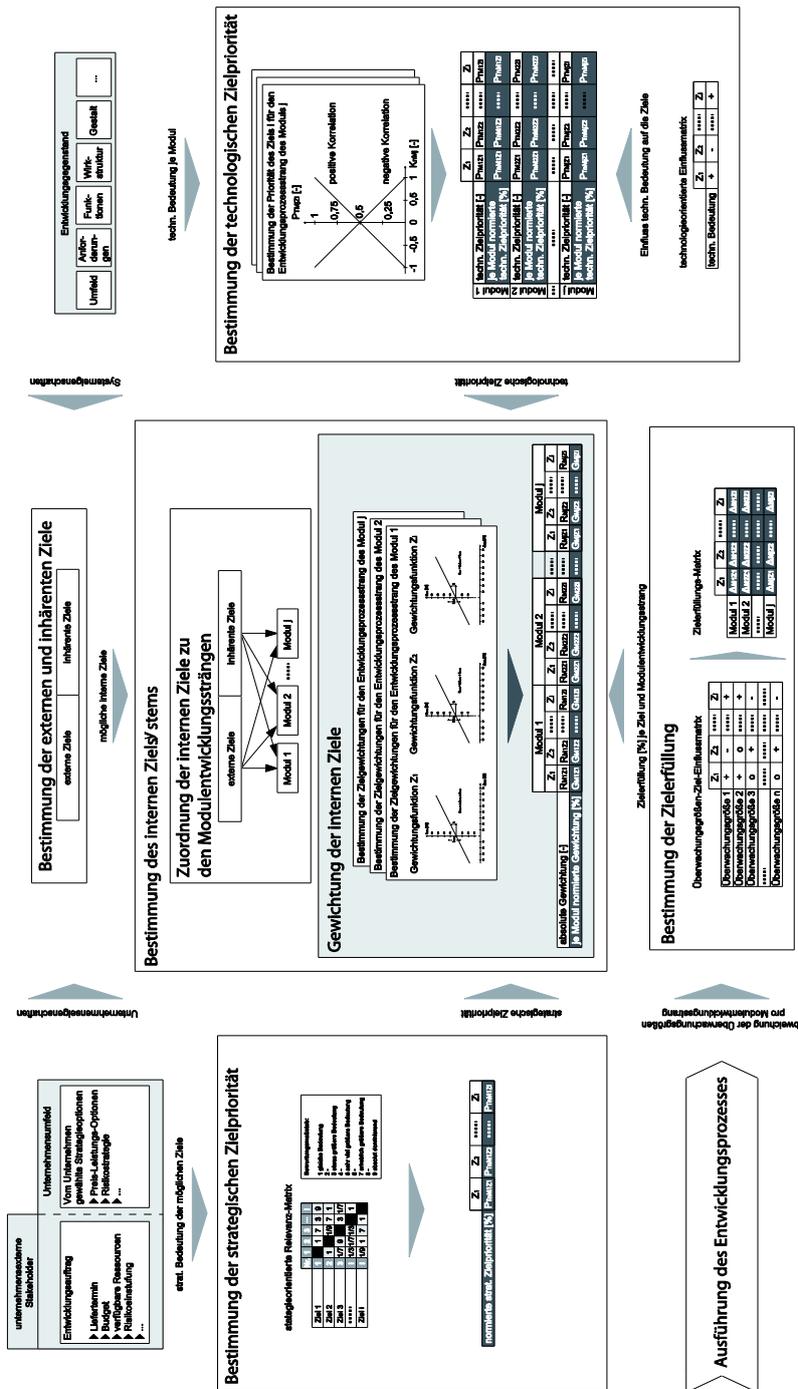


Bild A-4 Übersicht über das Verfahren zur situationspezifischen Auswahl und Gewichtung von Entwicklungsprozesszielen

