

Eine rechnerintegrierte Methode zur Produktkonzeption

vorgelegt von
Magister-Ingenieur
Zbigniew Ragan
aus Busko-Zdrój, Polen

von der Fakultät V - Verkehrs- und Maschinensysteme
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing. H. Meyer
Berichter: Prof. Dr.-Ing. F.-L. Krause
Berichter: Prof. Dr.-Ing. J. Krüger

Tag der wissenschaftlichen Aussprache: 27.11.2007

Berlin 2008
D 83

VORWORT

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter an der Technischen Universität Berlin am Institut für Werkzeugmaschinen und Fabrikbetrieb im Fachgebiet Industrielle Informationstechnik.

Mein Dank gilt dabei insbesondere Herrn Prof. Dr.-Ing. F.-L. Krause, dem ehemaligen Leiter des Fachgebiets Industrielle Informationstechnik am IWF der TU-Berlin sowie des Bereichs „Virtuelle Produktentstehung“ am Fraunhofer Institut für Produktionsanlagen und Konstruktionstechnik in Berlin, für seine wohlwollende Unterstützung, wertvollen Anregungen und kritischen Hinweise im Rahmen der Erstellung dieser Arbeit. Darüber hinaus bin ich ihm außerordentlich dankbar für die Jahre angenehmer, abwechslungsreicher und überdies bereichernder Zusammenarbeit.

Für die Bereitschaft zur Übernahme des Vorsitzes im Promotionsausschuss bedanke ich mich bei Herrn Prof. Dr.-Ing. H. Meyer, Leiter des Fachgebiets Konstruktion von Maschinensystemen am Institut für Konstruktion, Mikro- & Medizintechnik der TU-Berlin. Herrn Prof. Dr.-Ing. J. Krüger, Leiter des Fachgebiets „Industrielle Automatisierungstechnik“ am IWF der TU-Berlin und des Bereichs „Automatisierungstechnik“ am Fraunhofer Institut für Produktionsanlagen und Konstruktionstechnik danke ich für die Durchsicht der Dissertation und wertvolle Anregungen..

Die Mitarbeiter des Fachgebiets Industrielle Informationstechnik gaben mir in konstruktiven Gesprächen ihre Anregungen, sorgten für ein gutes Arbeitsklima und erwiesen sich mir gegenüber stets hilfsbereit.

Meinem Kollegen Dr.-Ing. Stefan Dreher und ebenso meinem Freund Volker Schudak danke ich für fruchtbare Diskussionen, vielfältigen Gedankenaustausch und sprachliche Unterstützung.

Meiner Frau Renata, meinen kleinen Töchtern Hanna und Fiona sowie auch meinen Freunden gilt mein Dank für Geduld und Verständnis, für Ermahnung als auch Ermunterung, für Entbehrungen, arbeitsbedingte Terminverschiebungen und für manche (vorübergehende) Unkonzentriertheit meinerseits im Umgang miteinander. Alles hat schließlich zur Verwirklichung dieser Arbeit beigetragen.

Berlin, den 23. Februar 2008

Zbigniew Ragan

Zusammenfassung

In der vorliegenden Arbeit wird ein neuartiges Konzept vorgestellt, das alle notwendigen Aktivitäten für den Aufbau eines Produktmodells, für die Konzipierung eines neuen Produkts oder einer Produktpassung sowie die Einführung und Pflege der rechnerintegrierten Methode beschreibt.

Hierzu wird, basierend auf der Analyse der wissenschaftlichen Ansätze aus dem Bereich der Produktentwicklung und der derzeitigen Situation auf diesem Gebiet in kleinen und mittelständischen Unternehmen (KMU), die Notwendigkeit der Erarbeitung einer Methode zur rechnerintegrierten Konzipierung abgeleitet. Aus den abgeleiteten Zielen wird im Hauptteil dieser Arbeit die Methode selbst gestaltet. Der Anwendungsbereich des hier beschriebenen Ansatzes liegt in der methodischen Unterstützung der auftragsbasierten Produktentwicklung in KMU bei der Einzel- und Kleinserienfertigung sowohl von Maschinen als auch von Gebrauchsgütern.

Die prototypische Implementierung eines innovativen Softwaresystems zur Erfassung und Definierung von Produkthanforderungen, -funktionen und Funktionsträgern sowie ausgewählte industrielle Beispiele dienen der Konzeptevaluierung.

Abstract

In this thesis, a novel concept is introduced which describes all the necessary activities for the creation of a product model, for the conceptual design of a new product or a product adaptation as well as the introduction and maintenance of the computer integrated method.

Based on an analysis of the scientific approaches of product development and the current situation in this field in small and middle enterprises (SME), the need to formulate a method for computer integrated conceptual design is derived. The method itself is designed around the derived goals in the main part of the thesis. The application scope of the approaches described lies within the methodical support of order-based product development in SME, in the make-to-order and small batch production on the one hand, as well as of machines and consumer goods on the other.

The prototypical implementation of an innovative software system for the acquisition and definition of product requirements, functions and function owners as well as selected industrial examples provide the basis for a concept evaluation.

INHALTSVERZEICHNIS I

1. Einleitung	1
2. Ausgangssituation	2
2.1. Ansätze der Produktentwicklung.....	2
2.2. Anforderungserfassung und –definition	12
2.3. Funktionserfassung und –definition	18
2.4. Produktentwicklung in KMU	21
2.5. Erfahrungsgrundlage	22
3. Präzisierung der Aufgabenstellung	24
3.1. Motivation und Zielsetzung	24
3.2. Abgrenzung.....	26
3.3. Formulierung der Anforderungen an eine rechnerintegrierte Produktkonzeption	28
4. Entwicklung einer Methode zur rechnerintegrierten Produktkonzeption	30
4.1. Einführung.....	30
4.2. Bestimmung der Maßnahmen zur Einführung des Einsatzes..	32
4.3. Konzipierung eines neuen Produkts	50
4.4. Konzipierung einer Produkthanpassung.....	53
4.5. Maßnahmen zur Pflege der rechnerintegrierten Methode	59
5. Entwicklung eines Systemkonzeptes	65
5.1. Ableitung der Systemfunktionalitäten.....	65
5.2. Erarbeitung der Systemarchitektur	66
5.3. Gestaltung der graphischen Benutzungsoberfläche.....	77
6. Prototypische Systemimplementierung und Evaluierung des Ansatzes	83
6.1. Prototypische Systemimplementierung.....	83
6.2. Praxisbeispiel einer Montagemaschine	89
6.3. Fallbeispiel eines Autositzes.....	93
6.4. Fallbeispiel einer Verkaufstheke	98
7. Zusammenfassung	103
8. Literaturverzeichnis	105

INHALTSVERZEICHNIS II

1. Einleitung	1
2. Ausgangssituation	2
2.1. Ansätze der Produktentwicklung	2
2.1.1. Einführung.....	2
2.1.2. Konstruktionsmethodik nach Pahl und Beitz.....	4
2.1.3. Konstruieren mit Konstruktionskatalogen nach Roth.....	5
2.1.4. VDI-Richtlinien 2221/2222	5
2.1.5. Produktentwicklungsmethodik nach Ehrlenspiel.....	7
2.1.6. Münchener Vorgehensmodell nach Lindemann	8
2.1.7. Methodische Produktplanung nach Seidel.....	9
2.1.8. Unterstützung der frühen Phasen nach Schwankl.....	10
2.2. Anforderungserfassung und –definition	12
2.2.1. Einleitung	12
2.2.2. Anforderungserfassung nach Pahl und Beitz.....	12
2.2.3. Anforderungsklärung nach Roth	13
2.2.4. VDI-Richtlinien 2221/2222	13
2.2.5. Anforderungserfassung nach Ehrlenspiel	13
2.2.6. Anforderungsermittlung nach Lindemann	14
2.2.7. Anforderungen in der Systemermittlung nach Kruse	14
2.2.8. Erfassen und Handhaben von Produkthanforderungen nach Ahrens.....	15
2.2.9. Anforderungsmodellierung nach Gebauer	15
2.2.10. Anforderungsklärung nach Jung	16
2.2.11. Erfassung und Integration von Anforderungen nach Jörg ...	16
2.2.12. Weitere Ansetze zur Anforderungserfassung	17
2.3. Funktionserfassung und –definition	18
2.3.1. Einleitung	18
2.3.2. Pahl und Beitz	18
2.3.3. Funktionsermittlung nach Roth	18
2.3.4. VDI-Richtlinien 2221/2222	19
2.3.5. Funktionsmodellierung nach Ehrenspiel	19
2.3.6. Funktionsmodellierung und Lösungsfindung nach Huang...	20
2.3.7. Funktionsorientierter Entwurf nach Gehrke	20
2.3.8. Funktionsgetriebene Konstruktion nach Leemhuis	21
2.4. Produktentwicklung in KMU	21

2.5. Erfahrungsgrundlage	23
3. Präzisierung der Aufgabenstellung	24
3.1. Motivation und Zielsetzung	24
3.2. Abgrenzung	26
3.3. Formulierung der Anforderungen an eine rechnerintegrierte Produktkonzeption	28
4. Entwicklung einer Methode zur rechnerintegrierten Produktkonzeption.....	30
4.1. Einführung	30
4.2. Bestimmung der Maßnahmen zur Einführung des Einsatzes	32
4.2.1. Einleitung	32
4.2.2. Analyse der technischen Machbarkeit	33
4.2.3. Analyse der wirtschaftlichen Machbarkeit.....	34
4.2.4. Erstellung eines rechnerinternen Produktmodells	35
4.2.4.1. Einleitung	35
4.2.4.2. Erstellung des Anforderungsmodells	37
4.2.4.3. Erstellung des Funktionsmodells	41
4.2.4.4. Überprüfung des erweiterten Funktionsmodells	49
4.2.4.5. Evaluierung des Modells	50
4.3. Konzipierung eines neuen Produkts	50
4.4. Konzipierung einer Produktpassung.....	53
4.5. Maßnahmen zur Pflege der rechnerintegrierten Methode.....	59
5. Entwicklung eines Systemkonzeptes	65
5.1. Ableitung der Systemfunktionalitäten	65
5.2. Erarbeitung der Systemarchitektur	66
5.2.1. Einleitung	66
5.2.2. Fachdatenmodell	66
5.2.3. Schichtenmodell.....	70
5.2.4. Komponentenmodell	71
5.2.5. Strukturklassenmodell.....	72
5.3. Gestaltung der graphischen Benutzungsoberfläche.....	77
5.3.1. Einleitung	77
5.3.2. Benutzungsoberfläche zur Anforderungsmodellierung.....	77
5.3.3. Benutzungsoberfläche zur Funktionsmodellierung.....	80

6. Prototypische Systemimplementierung und Evaluierung des Ansatzes	83
6.1. Prototypische Systemimplementierung	83
6.2. Praxisbeispiel einer Montagemaschine	89
6.3. Fallbeispiel eines Autositzes.....	93
6.4. Fallbeispiel einer Verkaufstheke	98
7. Zusammenfassung	103
8. Literaturverzeichnis	105

ABBILDUNGSVERZEICHNIS

Bild 2-1	Produktphasen nach SPUR/KRAUSE /1/	2
Bild 2-2	Generelles Vorgehen beim Entwerfen und Konstruieren /9/ ...	6
Bild 2-3	Vorgehenszyklus zur Lösungssuche nach EHRENSPIEL /4/	8
Bild 2-4	Schritte der Planungsmethodik von SEIDEL /14/	10
Bild 2-5	Vorgehen zur Unterstützung der Frühen Phasen /15/	11
Bild 4-1	Metamethodisches Rahmenwerk von WEIGT /49/	30
Bild 4-2	Kontext der Entwicklung der rechnerintegrierten Methode in Anlehnung an WEIGT /49/	31
Bild 4-3	Einführung des rechnerintegrierten Methodeneinsatzes	33
Bild 4-4	Erstellung eines rechnerinternen Konzeptmodells	36
Bild 4-5	Erstellung eines Anforderungsmodells	38
Bild 4-6	Erstellung eines Funktionsmodells	42
Bild 4-7	Ableitung einer Produktfunktion	43
Bild 4-8	Definition einer Produktfunktion	45
Bild 4-9	Dekomposition einer Produktfunktion	49
Bild 4-10	Vorbereitung einer Neukonzipierung	51
Bild 4-11	Konzipierung eines neuen Produkts	52
Bild 4-12	Vorbereitung einer Produktpassung	54
Bild 4-13	Konzipierung einer Produktpassung	55
Bild 4-14	Überprüfung und Änderung eines Anforderungsmodells.....	57
Bild 4-15	Prüfung und Änderung eines Funktionsmodells	58
Bild 4-16	Maßnahmen zur Pflege der rechnerintegrierten Methode	60
Bild 4-17	Überprüfung und Änderung der Vorgehensweise	62
Bild 4-18	Überprüfung und Änderung eines Produkt(typs)modells.....	63
Bild 5-1	Fachdatenmodell des Softwaresystems	67
Bild 5-2	Schichtenmodell des Softwaresystems	70
Bild 5-3	Komponentenmodell des Softwaresystems.....	72
Bild 5-4	Paketenmodell des Softwaresystems	73
Bild 5-5	Strukturklassenmodell des Softwaresystems	76
Bild 5-6	Auslegung einer Benutzungsoberfläche zur Definition und Strukturierung von Anforderungen.....	78
Bild 5-7	Auslegung einer Benutzungsoberfläche zur Definition und Strukturierung von Funktionen und Funktionsträger.....	80
Bild 5-8	Auslegung eines Editors zur Definition von Funktionsobjekten oder Funktionsträger.....	81

Bild 6-1	Realisierung des Anforderungseditors.....	83
Bild 6-2	Realisierung des Funktionseditors.....	86
Bild 6-3	Formular zur Definition der Funktionseingangsgrößen.....	87
Bild 6-4	Formular zur Definition der Funktionsträger	88
Bild 6-5	Montagemaschine für Getriebewelle	89
Bild 6-6	Anforderungsmodell einer Montagemaschine	90
Bild 6-7	Funktionsmodell einer Montagemaschine	91
Bild 6-8	Fahrersitz	93
Bild 6-9	Basis-Funktionsmodell eines Autositzes	95
Bild 6-10	Funktionsmodelle eines Fahrersitzes	96
Bild 6-11	Kühltheke zur Aufbewahrung und Verkauf von Sandwiches ...	98
Bild 6-12	Funktionsmodell einer Verkaufstheke.....	100
Bild 6-13	Funktionsmodelle einer Kühltheke.....	101

GLOSSAR

Anforderung	<p>Eine Anforderung ist eine Aussage, die vom Auftraggeber oder vom Auftragnehmer getroffen wird. Seitens des Auftraggebers bezieht sich die Aussage auf die gewünschten Eigenschaften, Leistungen oder Anwendungen eines Produkts. Zusätzlich definiert der Auftragnehmer notwendige Eigenschaften oder Leistungen, die die vom Auftraggeber gewünschten Eigenschaften oder Leistungen präzisieren und der internen Umsetzung dienen.</p>
Anforderungsmodell	<p>Anforderungskategorien und Anforderungen an ein Produkt, die hierarchisch in einer Datenstruktur mittels eines Softwarewerkzeugs definiert, erfasst und verwaltet werden.</p>
Anwendungsebene	<p>Diese Abstraktionsebene dient der Beschreibung von realen betrieblichen Anwendungssituationen in einem bestimmten Unternehmen. Jene Situationen werden mit Hilfe von konkreten Prozessen, Softwaresystemen und Mitarbeitern realisiert.</p>
Funktion	<p>Eine Funktion im Kontext der Produktentwicklung ist eine möglichst lösungsneutrale Aussage über eine gewünschte Leistung oder Anwendung eines Produkts. Um diese zu realisieren, überführt die Funktion Eingangsgrößen in die Ausgangsgrößen.</p>
Funktionsmodell	<p>Funktionen eines Produkts, die hierarchisch in einer Datenstruktur mittels eines Softwarewerkzeugs definiert, erfasst und verwaltet werden.</p>
Funktionsträger	<p>Ein Produkt, eine Baugruppe oder ein Bauteil, welche(s) eine oder mehrere Funktionen erfüllt. In der Konzeption stellt der Funktionsträger die gesuchte Lösung einer Funktion dar.</p>
rechnerintegrierte Methode	<p>Ist eine planmäßige, zielgerichtete Abfolge von Aktivitäten, die mit Hilfe von einem oder mehreren Softwarewerkzeugen durchgeführt wird. Sie wird mittels Texten und Diagrammen beschrieben.</p>

Objektebene	Diese Abstraktionsebene dient der Beschreibung von generischen Anwendungssituationen und Systemkonzepten. Sie werden mit Hilfe verschiedener Abfolgen von Aktivitäten, Schichten, Komponenten und Klassen in Form von textuellen Beschreibungen sowie Diagrammen definiert.
Produktkonzept	Wird im Kontext der Produktentwicklung als rechnerinterne Datenmenge aufgefasst, die einerseits das gewollte Produkt definieren und andererseits eine mögliche Erfüllung der gewünschten Eigenschaften, Leistungen oder Anwendung beschreiben.
rechnerinternes Produktmodell	Sind Daten, die die Entstehung, Nutzung und Entsorgung eines Produktes beschreiben sowie mit Softwarewerkzeugen verarbeitet werden.
UML	Unified Modeling Language ist eine standardisierte Beschreibungssprache, die der Modellierung von Systemen und Vorgehensweisen dient.
XML	Extensible Markup Language ist eine Sprache zur Beschreibung und Verarbeitung von hierarchisch strukturierten Daten, die in Textdateien verwaltet werden.

1 Einleitung

Der Begriff Produkt, abgeleitet aus dem Lateinischen *producere* – hervorbringen, erzeugen, herstellen, – wird inhaltlich sehr unterschiedlich verwendet. Das Spektrum reicht von Erzeugnissen pflanzlicher Art, den Agrarprodukten, über solcher baulicher Art, den Bauprodukten, über alle Arten der Industrieprodukte bis hin zu Dienstleistungen oder geistigen Schöpfungen [1]. Die Entwicklung von Produkten erfordert eine zielgerichtete Planung, die Vorgabe von Handlungsanweisungen und methodischen Hilfsmitteln. Am Anfang muss ein Produkt definiert sein, um es später mittels Entwicklungsprozessen und Werkzeugen zu präzisieren und zu konstruieren. Dabei können verschiedene Konzepte und Modelle (bestehend aus Anforderungen, Funktionen und Bauteilen) des Produktes erstellt, beurteilt und geändert werden, ohne dabei große Kosten zu verursachen. Dies ist ein großes Hilfsmittel, um die zentralen betriebswirtschaftlichen Kriterien, wie Kosten, Zeit und Qualität, zu verbessern. Die vorliegende Arbeit konzentriert sich an der rechnerintegrierten Produktkonzeption als Teil der Produktentwicklung, deren Ergebnisse in der Konstruktionsphase bei der Gestaltung des Produktes benutzt werden.

Hierzu wird zunächst die Ausgangssituation im Bereich der Produktentwicklung dargestellt, mit Schwerpunkten auf der Anforderungs- und Funktionserfassung sowie der Berücksichtigung der Situation bei kleinen und mittelständischen Unternehmen (KMU). Aus einer kritischen Betrachtung der vorgefundenen Ansätze und deren Defizite werden notwendige Schwerpunkte identifiziert und definiert. Daraus ergeben sich die Ziele für ein Konzept der rechnerintegrierten Methode bei der Produktkonzeption. Im Hauptteil dieser Arbeit wird das Konzept selbst entwickelt.

Die prototypische Implementierung eines Softwaresystems zur Erfassung und Definition von Produktanforderungen, -funktionen und Funktionsträgern sowie ausgewählte industrielle Beispiele dienen der Konzeptevaluierung.

2 Ausgangssituation

2.1 Ansätze der Produktentwicklung

2.1.1 Einführung

Die Produktentwicklung verkörpert nach SPUR/KRAUSE /1/ den planerischen, gestalterischen und organisatorischen Anteil der Produktentstehung. Konkret kann sie in die Einzelschritte Produktplanung, Produktkonstruktion einschließlich Produktionsvorbereitung und Produkterprobung gegliedert werden (vgl. Bild. 2-1). Sie erfordert eine zielgerichtete Planung von Handlungsanweisungen. Bei komplexen Gebilden geschieht dies systematisch in Teilschritten. Entwickeln ist zweckgerichtetes Auswerten und Anwenden von Forschung und Erfahrung. Dies setzt auch eine ständige Verfügbarkeit des notwendigen Wissens voraus. /1/

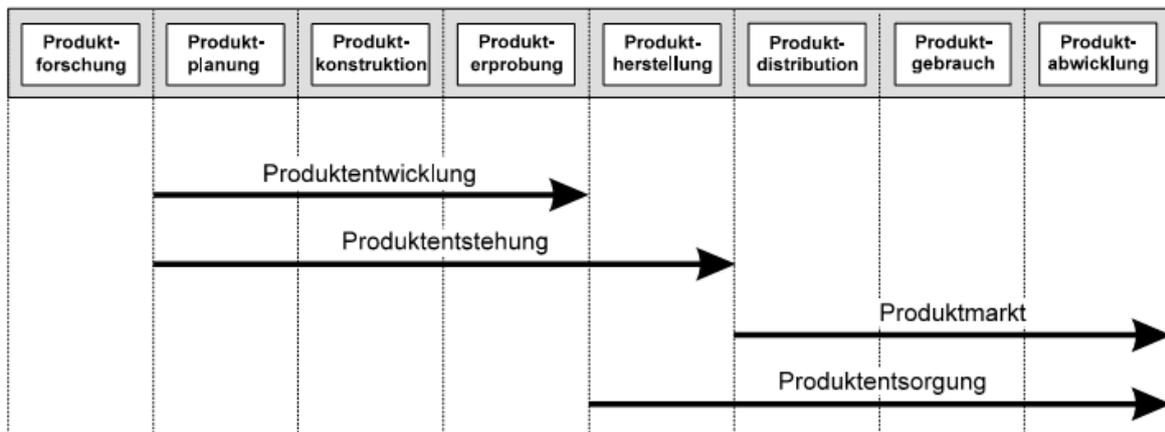


Bild 2-1: Produktphasen nach SPUR/KRAUSE /1/

Zur Optimierung des Produktentwicklungsprozesses ist eine informationstechnische Verknüpfung der Einzelaufgaben in Prozessketten und eine Nutzung von Produktmodellen erforderlich. Die Optimierungskriterien sind vorrangig auf kurze Entwicklungszeiten, niedrige Entwicklungs- und Fertigungskosten sowie hohe Qualität ausgerichtet. Als Erfolgskriterien können auch umweltorientierte Verträglichkeiten sowie die Erfüllung gesetzlicher Randbedingungen zum Tragen kommen. Die Komplexität produktspezifischer Interdependenzen bedeutet für den Produktentwicklungsprozess eine Ausrichtung auf verschiedenste, teilweise divergierende Optimierungsgrößen und stellt für den planerischen Ansatz eine große Herausforderung dar. /1/

In der Studie „Neue Wege der Produktentwicklung“ /2/ wurde auf der Basis einer umfangreichen Industriebefragung zu dem Schluss gekommen, dass

die Entwicklungsmethodik in der Wirtschaft als überaus bedeutsam angesehen wird. Von der Anwendung der Produktentwicklungsmethodik verspricht man sich eine Verbesserung der zentralen betriebswirtschaftlichen Kriterien, wie Kosten, Zeit und Qualität. Die Autoren GRABOWSKI u. a. /2/, schlagen vor, dass die Produkt-entwicklungsmethodik energisch vorangetrieben werden muss, um der zunehmenden Integration von Mechanik, Elektronik, Regelungstechnik und Software in technischen Erzeugnissen gerecht werden zu können.

HUTERER /3/ kommt in seiner Arbeit zu dem Schluss, dass Entwicklungsmethodik zu erfolgreicher Produktentwicklung führen kann, aber nicht unbedingt führen muss; denn neben erfolgreicher Methodenanwendung sind auch Misserfolge zu beobachten. Er schreibt weiter, dass viele Ursachen der Fehlschläge meist in den Schwierigkeiten liegen, die sich bei der Anwendung ergeben, und dass die Leistung der Methodenanwendung hinter den Erwartungen des Anwenders zurück bleibt. Er schreibt, dass die Komplexität und Theorielastigkeit, die Starrheit und Unflexibilität, der hohe Aufwand und der späte „Return of Invest“, der präskriptive Charakter und die fehl angenommene Allgemeingültigkeit als Ursachen für die eingeschränkte Leistungsfähigkeit verantwortlich sein können. Er schlussfolgert weiter, dass ein Entwickler nach erstmalig erfolgter Methodenanwendung vielleicht auch deshalb keinen Nutzen sieht, weil er sein Vorgehen ohne diese ähnlich gewählt hätte. Somit erscheint ihm die Methodenanwendung im Rückblick als überflüssig. Dadurch zieht er das Fazit, dass eine Methodik die Produktentwicklung verbessern kann, aber bislang wenig verbreitete Verankerung in der industriellen Praxis gefunden hat. Eine umfassende Ausarbeitung über die Ursachen der ausbleibenden Methodenanwendung ist bei HUTERER /3/ zu entnehmen; daher wird darauf an dieser Stelle nicht ausführlicher eingegangen.

Bei der Produktentwicklung werden Verlauf- und Strukturmodelle eingesetzt, um einerseits ein besseres Verständnis der Methode und der Problemstellung zu erreichen und um andererseits als Arbeitsgegenstände bei der Durchführung von Entwicklungsaktivitäten bearbeitet zu werden. Die Grundlage für die Erarbeitung oder Erstellung von Modellen bilden Informationen, deren Beschaffung bei der Problemanalyse bis zu 50% der Arbeitszeit eines Produktentwicklers in Anspruch nimmt (/2/, S. 71; /4/, S. 493). Während der Produktentwicklung treten auch Probleme bei der Einschätzung auf, ob und wann Modelle (Anforderungs-, Funktions-, Bauteil- oder Wirkmodelle) nützlich und wann die wirtschaftlich notwendigen unteren und oberen Grenzen der Detaillierung und Abbildung des Produktes oder Vorgehens erreicht sind.

FUCHS /5/ schreibt, dass strukturierte Modelle eine problemorientierte und nicht eine methodenorientierte Modellanwendung begünstigen sollen, d. h., die Auswahl eines Modells orientiert sich an den Besonderheiten

des Problems. Er schreibt weiter, dass die Einheit von Methode und Modell deutlich wird, wenn bei einem Problem häufig das Modell als Ergebnis einer Methodenanwendung steht. Weiterhin eignet sich ein Modell für die Dokumentation und Darstellung einer Aufgabe, weil die komplexen Zusammenhänge von Objekten durch Relationen wahrgenommen werden können. Eine umfassende Ausarbeitung über Modelle der Produktentwicklung ist bei FUCHS /5/ zu entnehmen; deshalb wird an dieser Stelle darauf nicht näher eingegangen.

2.1.2 Konstruktionsmethodik nach Pahl und Beitz

PAHL/BEITZ /6/ halten angesichts der zentralen Verantwortung des Konstrukteurs für die Produktentwicklung sowie der unternehmerischen Verantwortung zur rechtzeitigen Entwicklung marktfähiger Produkte ein Vorgehen zum Erreichen guter Lösungen für nötig, das planbar, flexibel, optimierbar und nachprüfbar ist. Sie beschreiben ein generelles Vorgehen zum Entwickeln und Konstruieren von Maschinen (basierend auf den Arbeitsschritten aus der VDI Richtlinie 2222, an der sie mitgearbeitet haben); vgl. /6/ das Bild 1.10 *Generelles Vorgehen beim Entwickeln und Konstruieren*, S. 28. und Bild 3.3 *Arbeitsschritte beim Planen und Konstruieren* S. 86. Dabei geht es um ein sequentielles Vorgehen, das aufgrund der Anordnung des Modells nur bedingt wieder gestartet und zur weiteren Konkretisierung der Aufgabestellung iterativ durchlaufen wird. Es besteht aus 4 Arbeitsschritten

1. Planen und Klären der Aufgabe,
2. Entwickeln der Prinzipiellen Lösung,
3. Entwickeln der Baustruktur und
4. Entwickeln der ausführungs- und Nutzungsunterlagen.

Im ersten Arbeitsschritt soll die Anforderungsliste ausgearbeitet werden, in dem darauf folgenden Schritt sollen die Funktionen, Wirkprinzipien und -strukturen ermittelt werden. Die Aufgabe zur Entwicklung eines Produktes - also der Entwicklungsauftrag - wird für Pahl und Beitz meist durch eine konkrete Bestellung eines Kunden oder eine interne Anregung an die Entwicklungsabteilung gestellt. Ein Entwicklungsauftrag enthält außer den Angaben zu Funktionalität und Leistungsumfang des Produkts auch wirtschaftliche und organisatorische Daten. Dem Entwickler ist überlassen, die Produkthanforderungen und Produktfunktionen zu erkennen, sie zu strukturieren und zu dokumentieren. Eine informationstechnische Unterstützung des Ansatzes wurde im Rahmen des SFB 203 /7/ realisiert. Dabei wurde das Konstruktionsanalyse- und -leitsystem KALEIT implementiert.

2.1.3 Konstruieren mit Konstruktionskatalogen nach Roth

ROTH /8/ beschreibt in seinem Ansatz „Konstruieren mit Konstruktionskatalogen“ eine Methodik, die auf den VDI-Richtlinien 2221 basiert. Dabei definiert er ein algorithmisches Auswahlverfahren zur Konstruktion mit Konstruktionskatalogen (AAK), in welcher er Handlungsanweisungen für die sieben definierten Arbeitsabschnitte und deren Arbeitsergebnisse präzisiert. Er behandelt als Schwerpunkte das Zusammenfassen, Ordnen und Wiederverwenden von Lösungen, die für die Konstruktion immer wieder brauchbar sind, und von Baugruppen und deren Verbindungsmöglichkeiten in so genannten Konstruktionskatalogen /8/. Er schlägt auch einen variablen Ablaufplan mit Einstiegsmöglichkeiten in verschiedenen Konstruktionsphasen /8/ S. 38 vor, um die Arbeit beim Auftreten von Änderungen sowie bei Anpassungs- und Variantenkonstruktionen zu gewährleisten. Die informationstechnische Unterstützung betrachtet er nicht näher. Die einfache Nutzung der Kataloge, Matrizen und Checklisten mit Office-Standardprogrammen gestaltet sich aber aufgrund deren Umfangs und der gegenwärtigen Bildschirmgröße ineffektiv.

2.1.4 VDI-Richtlinien 2221/2222

Die VDI-Richtlinie 2221 /9/ hat einen übergeordneten Charakter gegenüber der VDI-Richtlinie 2222 /10/, die nur ihre Teilaspekte ausführlicher behandelt. Sie schlägt allgemeingültige und branchenunabhängige Grundlagen für die methodische Entwicklung vor. Sie definiert dabei Arbeitsschritte und -ergebnisse, die in der Praxis eine Leitlinie für eine Vorgehensweise darstellen kann. Als methodische Grundlagen der VDI-Richtlinien dienen Ansätze der Systemtechnik zur generellen Problemlösungsmethodik sowie die bekannten Methodenvorschläge aus dem Maschinenbau und der Feinwerktechnik von Koller, Pahl und Beitz, Rodenacker sowie Roth. /9/

Die VDI-Richtlinie 2222 schlägt für den Entwicklungsprozess vier Modellierungsphasen vor: Anforderungs-, Funktions-, Prinzip- und Gestaltmodellierung (unterteilt in Entwerfen und Ausarbeiten). Sie werden dann in sieben Arbeitsabschnitten konkretisiert. Bild 2-2 stellt das Vorgehen dar.

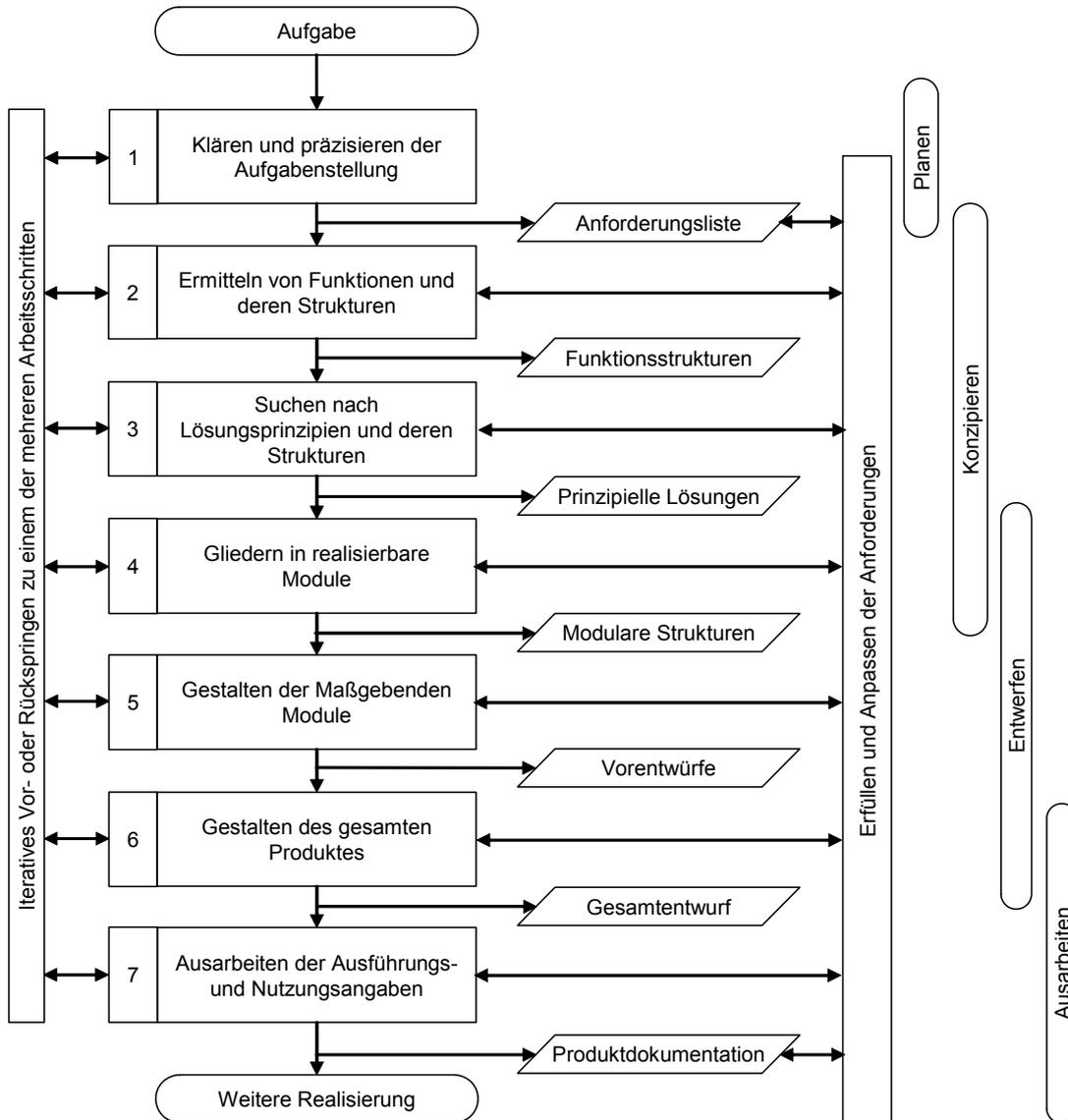


Bild 2-2: Generelles Vorgehen beim Entwerfen und Konstruieren /9/

Die VDI-Richtlinie 2222 vertieft die ersten drei Arbeitsschritte. Ausgehend von einer Entwicklungs- oder Konstruktionsaufgabe soll eine prinzipielle Lösung erarbeitet und dokumentiert werden. Besondere Aufmerksamkeit wird der systematischen Funktionsauswahl und der Variation der unterschiedlichen physikalischen Prinzipien gewidmet.

Eine informationstechnische Unterstützung des konstruktionsmethodischen Ansatzes der VDI-Richtlinien 2221/2222 wurde im Rahmen des BMBF Forschungsprojekts iViP - integrierte Virtuelle Produktentstehung im TP 3.2 „Funktionsorientiertes Entwerfen“¹ mit dem FOD-System prototypisch realisiert /11/.

¹ Forschungsprojekt koordiniert durch das Fraunhofer IPK Berlin, gefördert durch das Bundesministerium für Bildung und Forschung, Projektnr. 02PL10xxx, 1998-2002

Ergänzend zur VDI-Richtlinie 2221 wurde im Jahr 2004 die VDI-Richtlinie 2206 „Entwicklungsmethodik für Mechatronische Systeme“ publiziert /12/. Ziel der Richtlinie ist, das domänenübergreifende Entwickeln mechatronischer Systeme methodisch zu unterstützen. Der Schwerpunkt liegt dabei auf Methoden und Werkzeugen für die „Frühen Phasen“ des Entwickelns. Eine Richtschnur für das Vorgehen bildet das angepasste V-Modell, das die logische Abfolge wesentlicher Teilschritte bei der Entwicklung mechatronischer Systeme beschreibt. Da die VDI-Richtlinie 2206 eine Ergänzung der VDI 2221 ist, enthält sie keine signifikanten Inhalte, die für die vorliegende Forschungsarbeit detaillierter zu betrachten wären, so dass an der Stelle auf die Richtlinie nicht näher eingegangen wird.

2.1.5 Produktentwicklungsmethodik nach Ehrenspiel

EHRENSPIEL /4/ beschreibt in seinem wissenschaftlichen Ansatz ein Vorgehen, das den gesamten Produktlebenszyklus von der Ideensuche bzw. dem Auftrag bis zur Auslieferung des Produkts an den Nutzer unterstützen soll /4/. Er bezieht sich dabei auf das TOTE-Schema /4/ S. 82 und verwendet für seine Arbeit auch die VDI-Richtlinie 2221. In seinem Vorgehenszyklus beschreibt er eine Abfolge von drei Arbeitsschritten, die zu einer Lösung führen. Ausgangspunkt ist dabei eine Aufgabe bzw. ein Problem. Durch den Aufbau des Vorgehenszyklus soll die geforderte Erzeugung der Lösungsvielfalt innerhalb der ersten zwei Arbeitsschritte und die Einschränkung der Lösungsvielfalt im dritten Arbeitsschritt bis hin zu einer Lösungsauswahl unterstützt werden. Er zeigt dabei auch die zulässigen und notwendigen oder sinnvollen Iterationen und Rekursionen zwischen den einzelnen Arbeitsschritten. Mit Hilfe der Methodik versucht er, eine integrierende Denkweise zu vermitteln, die nicht nur die Zusammenarbeit der einzelnen Abteilungen im Unternehmen fördern, sondern auch die Festlegung aller Produkteigenschaften von der Nutzung bis zur Entsorgung besser unterstützen soll. Eine softwaretechnische Unterstützung des Vorgehens behandelt er nur implizit, betonend, dass eine logische Konstruktionsmethodik Voraussetzung für eine Rechnerunterstützung sein muss. Bild 2-3 zeigt den Vorgehenszyklus für die Lösungssuche mit den zugehörigen Strategien.

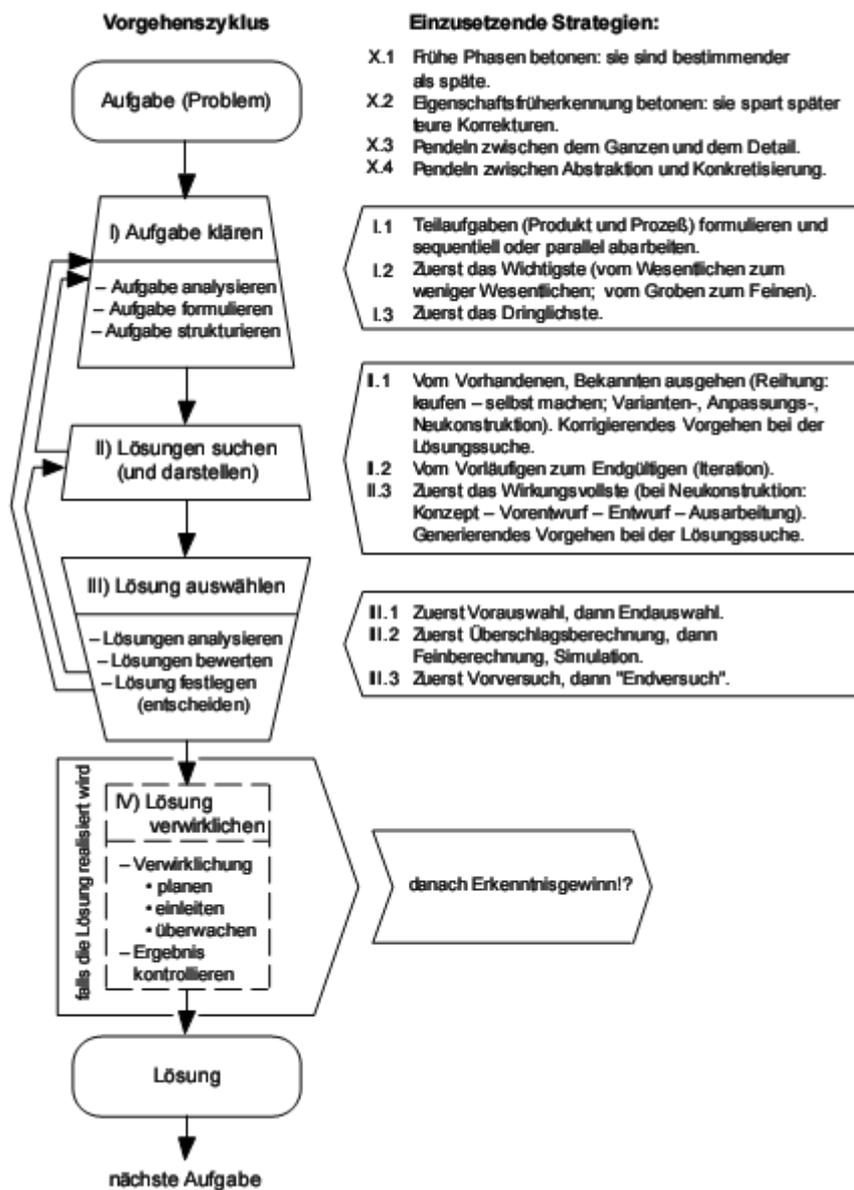


Bild 2-3: Vorgehenszyklus zur Lösungssuche nach EHRENSPIEL /4/

2.1.6 Münchener Vorgehensmodell nach Lindemann

Lindemann schlägt ein Vorgehensmodell zur Planung, Orientierung sowie zur Reflexion von Prozessen vor. /13/ Er unterteilt sein Münchener Vorgehensmodell (MVM) in sieben gleichwertig nebeneinander stehende Elemente, die nicht wie bisher üblich in einer strikt sequentiellen Abfolge abgearbeitet werden, sondern in Form eines Netzwerks dargestellt sind. Er schlägt einen Standardweg durch dieses Modell vor, das zu einer erfolgreichen Problemlösung führen soll. Der Standardweg folgt der Sequenz:

1. Ziel planen,
2. Ziel analysieren,

3. Ziel strukturieren,
4. Lösungsalternativen suchen,
5. Eigenschaften ermitteln,
6. Entscheidungen herbeiführen,
7. Ziel absichern.

Daneben gibt er noch Beispiele für alternative Wege, wie ein iteratives oder rekursives Vorgehen ablaufen soll. Er diskutiert in dem Arbeitsschritt „Zielstrukturierung“ die Funktionsmodellierung und schlägt dafür bereits existierende Ansätze (wie z. B. von Ehrlenspiel) vor.

Das Vorgehensmodell ist ziemlich abstrakt formuliert, so dass es nicht eindeutig klar ist, inwieweit es informationstechnisch unterstützt werden kann, obwohl der Autor die Benutzung von Methoden vorschlägt (z. B. Mind Mapping), die informationstechnische Unterstützung liefern.

2.1.7 Methodische Produktplanung nach Seidel

SEIDEL /14/ hat eine allgemeine Methodik zur Erstellung von Produktkonzepten erarbeitet. Diese besteht aus fünf aufeinander aufbauenden und durch gegenseitigen Informationsaustausch miteinander verknüpften Schritten, ergänzt um einen Vorbereitungs- und einen Nachbereitungsschritt /14/. Er beschreibt auch detaillierte Handlungsanweisungen für die Abarbeitung der einzelnen Schritte, die aber auf einer gewissen Abstraktionsebene bleiben. Er definiert als notwendige Eingangsgröße einen Planungsauftrag, mit dem der erste Schritt beginnt. Nach Abschluss des Prozesses erfolgt die Entscheidung über die Weiterverfolgung des erarbeiteten Konzepts. Er fokussiert seine Methodik auf Unternehmen, die zur Sicherstellung der Funktionserfüllung nötige Effekte technisch beherrschen und deren Produkte jeweils eine überschaubare Anzahl von Varianten aufweisen. Weiterhin sollen die adressierten Unternehmen definierte Prozesse für die Herstellung neuer Produkte haben und sich ebenso bereits mit Methoden und kundenorientierter Informationsgewinnung auseinandergesetzt haben /14/. Er schlägt in seinem Arbeitsschritt „Konzeptvarianten erzeugen“ vor, bei Bedarf eine Funktionsstrukturierung durchzuführen, um die Produktplanung methodisch zu unterstützen. Dazu gibt er die Nutzung der existierenden Ansätze vor, geht dann aber nicht näher auf die Funktion und Funktionsstrukturierung ein. Bei genauer Betrachtung der Arbeitsschritte wird ersichtlich, dass der Autor vor allem mit seinem Ansatz eine Neuentwicklung adressiert. Er macht über die informationstechnische Unterstützung seiner Methode keine Erläuterungen.

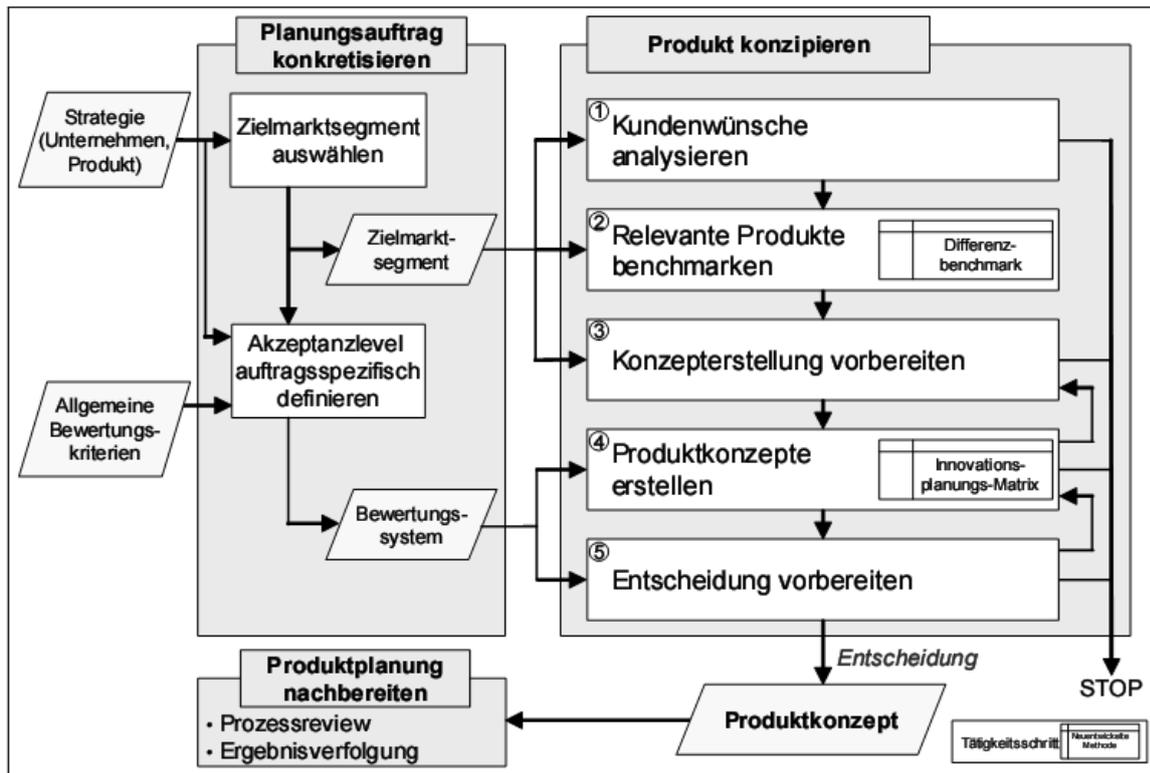


Bild 2-4: Schritte der Planungsmethodik von SEIDEL /14/

Es ist daher nicht klar, inwieweit die Methode bei einem Versuch der Rechnerunterstützung angepasst oder geändert werden soll. Bild 2-4 veranschaulicht die Arbeitsschritte seiner Planungsmethodik.

2.1.8 Unterstützung der Frühen Phasen nach Schwankl

SCHWANKL /15/ entwickelte ein erweitertes Vorgehensmodell zur Unterstützung der Frühen Phasen der integrierten Produktentwicklung, das die Bereiche Aufgabenklärung, Lösungssuche und -absicherung umfasst. Das Vorgehensmodell (Bild 2-5) beschreibt verschiedene Tätigkeiten, die auf dem Weg von der Aufgabenstellung bis zur Lösung abzuarbeiten sind. Er gibt nur einen groben Weg als orientierende Richtschnur vor und verzichtet auf eine genaue Festlegung von einzuhaltenden Arbeitsschritten und Iterationen. Ein situatives Vor- und Zurückspringen innerhalb der Teilschritte bzw. von einem in den nächsten und die wiederholte Bearbeitung mit unterschiedlicher Intensität sollen eine bedarfsgerechte Reaktion auf die jeweiligen Belange der Aufgabenstellungen und auf die gerade vorliegenden Randbedingungen erlauben. /15/ In seinem Ansatz ordnet er den einzelnen Tätigkeiten mehrere Methoden und Werkzeuge zu, deren Anwendung am besten unternehmensspezifisch (bzw. abteilungsspezifisch) erfolgen soll.

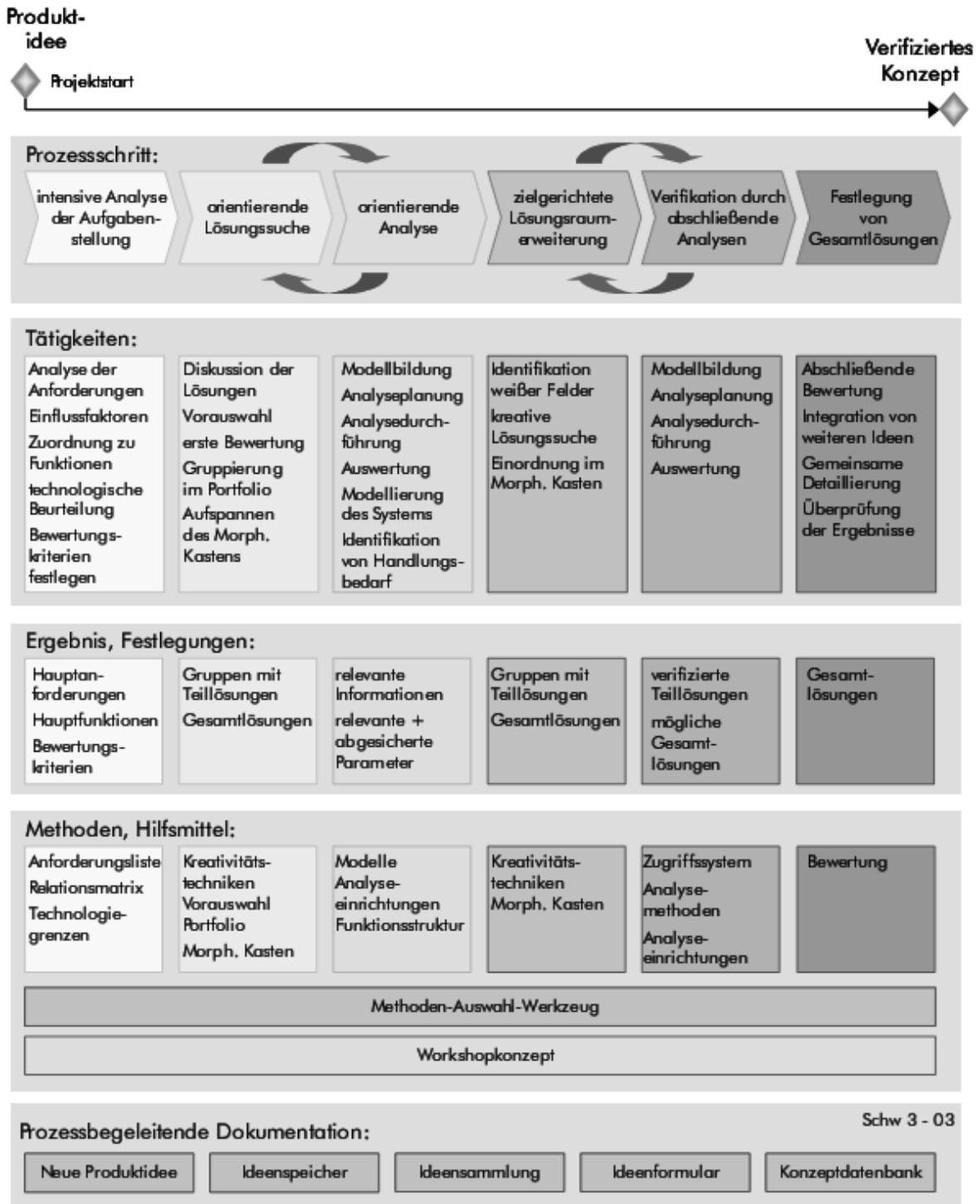


Bild 2-5: Vorgehen zur Unterstützung der Frühen Phasen /15/

Er argumentiert in diesem Zusammenhang, dass es von Vorteil ist, für jede Tätigkeit mehrere alternativ anzuwendende Methoden vorzugeben, aus denen dann die am geeignetesten erscheinende Methode gewählt werden soll. Dabei sollen bewusst einfache Methoden bevorzugt werden, die leicht zu erlernen, einfach anwendbar und flexibel anpassbar sind. Er definiert den Einsatzbereich des Vorgehens als „Anspruchvolle Aufgabenstellungen bei

Entwicklung von innovativen Produkten“. Aus der folgenden Detaillierung ist ersichtlich, dass eine Neukonstruktion adressiert wird. Er diskutiert kurz die Funktionsmodellierung, geht nicht weiter auf Details ein, sondern schlägt vor, dass diverse existierende Ansätze bedarfsgerecht angewandt werden sollen und listet diese im Anhang auf. Er schlägt auch existierende Rechnerwerkzeuge für die Unterstützung der Entwickler bei Dokumentation ihrer Tätigkeiten vor. Bild 2-5 stellt das erweitertes Vorgehensmodell zur Unterstützung der Frühen Phasen dar.

2.2 Anforderungserfassung und -definition

2.2.1 Einleitung

Da mit Hilfe der Anforderungen die von den Kunden angefragten Eigenschaften und Funktionen eines gewünschten Produkts expliziert und präzisiert werden können, ist eine entsprechende Anforderungserfassung und -definition von entscheidender Bedeutung und bildet somit eine Grundlage für die erfolgreiche Realisierung eines Entwicklungsauftrages. Durch die Erfassung der Anforderungen können die Kundenwünsche besser spezifiziert, dokumentiert sowie zur Kontrolle und Steuerung der richtigen Entwicklung eines Produkts benutzt werden. Die Erarbeitung einer Anforderungsstruktur erleichtert die Produktdokumentation und deren Nutzung in der Produktentwicklung, weil durch die Strukturierung eine Lösung systematisch erarbeitet und die Steuerung der Entwicklung effektiver durchgeführt werden können.

2.2.2 Anforderungserfassung nach Pahl und Beitz

Nach Pahl/Beitz /4/ enthält ein Entwicklungsauftrag neben den Angaben über Funktionalität und Leistungsumfang des Produkts, auch wirtschaftliche und organisatorische Daten. Dem Entwickler obliegt es nun, die lösungs- und gestaltbeeinflussenden Produktanforderungen zu erkennen, sie mit quantitativen Angaben zu versehen und in einer Anforderungsliste zu dokumentieren. Sie empfehlen, Anforderungen in einer Liste zu erfassen. Die vorgesehenen Felder sollen das Ändern und Weiterverarbeiten der Anforderungen erleichtern. Sie schreiben, dass eine Anforderungsliste zu Beginn einer Entwicklung niemals vollständig sein kann, sondern im Verlauf der Entwicklung wächst und sich ändert. Eine Anforderungsliste kann zu Beginn nicht alle notwendigen Anforderungen enthalten, sondern jeweils nur jene, die unbedingt zur Bearbeitung des gerade anstehenden Arbeitsschrittes notwendig sind, weshalb der Inhalt einer Anforderungsliste nicht nur produkt- sondern auch arbeitsschrittabhängig ist. /4/

2.2.3 Anforderungsklä rung nach Roth

ROTH /8/ gibt vor, bei der Anforderungsklä rung nicht nur verschiedene technische, wirtschaftliche oder organisatorische Kriterien zu beachten, sondern eine weitere Dimension in Form des Produktlebenslaufs von der Planung bis hin zum Recycling einzubeziehen. Er hat eine Suchmatrix zum Aufstellen der Anforderungsliste vorgeschlagen; sie enthält insgesamt 90 Felder mit darin eingetragenen Fragen. Für eine präzise Erstellung der Anforderungsliste erarbeitete er eine detaillierte Checkliste. Die Suchmatrix soll nach Roth im Entwicklungsprozess mehrfach durchlaufen werden, weil die Anforderungsliste immer erst für den gerade folgenden Konstruktionsabschnitt vollständig sein kann. Er beschreibt detaillierte Formblätter zur Erfassung der Anforderungen in der Liste. Auf Definition von Elementen einer Anforderung geht er aber nicht ein.

2.2.4 VDI-Richtlinien 2221/2222

Die VDI-Richtlinie 2221 /9/ enthält, in dem Arbeitsschritt „Klä ren und Präzisieren der Aufgabenstellung “ alle Informationen, wonach die zum Entwurf und zur Herstellung eines gewünschten Produkts in der Zukunft bekannt sind, zusammenzutragen und in einer Anforderungsliste zu erfassen sind. In der Liste sollen alle notwendigen Produkthanforderungen für weitere Arbeitsschritte bereitgestellt werden. Eine Struktur bzw. das Aussehen der Liste wird von den Autoren nicht vorgeschlagen; ebenso wird auch auf den Verlauf einer Definition für die Produkthanforderung nicht eingegangen.

2.2.5 Anforderungserfassung nach Ehrlenspiel

Nach EHRENSPIEL /4/ sollen alle gesammelten Informationen bei der Aufgabenanalyse und -strukturierung in einer Anforderungsliste erfasst werden, die er als eines der wichtigsten Dokumente beim methodischen Konstruieren bezeichnet, welche gelebt und immer wieder aktualisiert werden muss. Er geht kurz darauf ein, dass es unterschiedliche Vernetzungen zwischen Anforderungen geben kann (Zielkonflikt, Zielunabhängigkeit, Zielunterstützung), nennt aber wenige Möglichkeiten, wie diese zu ermitteln und vor allem zu handhaben sind. Es sollen weiterhin Checklisten und Leitlinien eingesetzt werden, um das Vergessen von Anforderungen zu vermeiden. Bei neuen Produkten und ungewohnten Situationen können aber Lücken in den Anforderungen nicht ausgeschlossen werden. Er schreibt vor, die Produkthanforderungen in einer Liste zu erfassen; dazu legt er Kriterien für deren Aufbau und Struktur fest. Auf die detaillierte Definition einer Anforderung und auf eine methodische Unterstützung für die Pflege der Anforderungsliste geht er nicht ein.

2.2.6 Anforderungsermittlung nach Lindemann

In dem Münchener Vorgehensmodell definiert LINDEMANN /13/ Schritte „Ziel analysieren“ und „Ziel strukturieren“ als Maßnahmen zur Anforderungsermittlung, und -erfassung. Er konkretisiert die Schritte mit einzelnen Elementen und gibt Empfehlungen bezüglich des Vorgehens und der einsetzbaren Methoden. Er legt fest, existierende Methoden entsprechend der zutreffenden Situation auszuwählen und bei Bedarf auch durch Modifikationen anzupassen. Der Schritt „Analyse des Ziels“ dient zur Klärung konkreter und detaillierter Anforderungen an das zu entwickelnde Produkt. Dabei schlägt er vor, die Anforderungen in einer Anforderungsliste zu erfassen. Damit Anforderungen auch effektiv verarbeitet und für die Entwicklung die richtigen Schwerpunkte gesetzt werden können, bespricht er in dem Schritt „Ziel strukturieren“ Methoden, die eine gute Strukturierung der Entwicklungsaufgabe ermöglichen.

2.2.7 Anforderungen in der Systementwicklung nach Kruse

KRUSE /16/ beschäftigt sich in seiner Arbeit mit der Klärung, Aufbereitung und Bereitstellung von Anforderungen in interdisziplinären Entwicklungsprojekten. Dabei weist er darauf hin, dass die Anzahl der Anforderungen noch komplexer wird, wenn ein Produkt durch Zusammenarbeit von Entwicklungsteams aus unterschiedlichen Fachdisziplinen realisiert werden soll. In diesem Falle ist eine Verständigung zwischen diesen Disziplinen oftmals schwierig, weil die Entwickler häufig nicht die gleiche Sprache sprechen bzw. die gleichen Begrifflichkeiten benutzen /16/. Kruse schlägt vor, Kundenanforderungen direkt an eine abstrakte oder konkrete Eigenschaft des Gesamtsystems, einer Systemkomponente oder eines Systemaspektes zu richten, und definiert Anforderungsobjekte. Wenn ein Zielobjekt der Anforderung noch nicht konzipiert ist (das ist bei Neuentwicklungen oft der Fall), soll ein entsprechendes „Entwicklungsobjekt“ definiert werden. Bei diesem Objekt muss es sich nicht zwingend um ein herstellbares Teil handeln, sondern man kann auch einen logischen Raum definieren. Kruse empfiehlt eine regelmäßige Qualitätsüberprüfung der Anforderungsstruktur, weil eine hohe Anzahl sich ausschließender Beziehungen eine mangelhaft abgestimmte Struktur zeigt, wodurch sich die Produktentwicklung erschweren kann und viele konkurrierende Beziehungen Iterationsschleifen in der Entwicklung erwarten lassen. Wenn aber zwischen den Anforderungen nur wenige Beziehungen existieren, kann dies bedeuten, dass entweder die Anforderungen sehr genau definiert sind oder die Beziehungsanalyse nur unzureichend durchgeführt wurde. Kruse beschreibt auch die Gewichtung von Anforderungen; er unterscheidet zwischen Fest-, Ziel- und Wunschforderungen. Festforderungen werden verwendet, um die nutzbaren Lösungen herauszufiltern Ziel- und Wunschforderungen

dienen der Wertung der nutzbaren Lösungen untereinander. Basierend auf diesen Überlegungen empfiehlt Kruse, Produkthanforderungen strukturiert zu erfassen, und liefert dafür entsprechende Kriterien. Weiterhin zeigt er eine informationstechnische Unterstützung der Anforderungsklä rung.

2.2.8 Erfassen und Handhaben von Produkthanforderungen nach Ahrens

AHRENS /17/ hat sich in ihrer Arbeit mit dem Erfassen und Handhaben von Produkthanforderungen auseinandergesetzt. Die Autorin hat ein iterativ anzuwendendes Vorgehensmodell zur Anforderungsklä rung vorgeschlagen. Sie sagt, dass der Kunde eine herausragende Bedeutung für die Qualität der Anforderungen hat; deshalb baut sie ihr Vorgehen auf den Kundenbedürfnissen auf. In ihrer Arbeit identifiziert sie vier Dimensionen, die die Anforderungserfassung und -handhabung beeinflussen: Kunden-, Transformations-, Konkretisierungs- und Handhabungsdimension. Sie schlägt eine informationstechnische Unterstützung vor, die die Methodenauswahl in den unterschiedlichen Phasen des Vorgehens sowie die Erfassung der Anforderungen hinsichtlich ihrer Dokumentation und Verwaltung ermöglichen soll. Darauf aufbauend beschreibt sie einen Softwareprototyp zur rechnerunterstützten Erfassung und Handhabung von Anforderungen, um ihre Vorgehensweise in der industriellen Praxis zu evaluieren. Das Softwaresystem ist Web-basiert und verwaltet Anforderungen in der Form von Listen. Dabei konzentriert sie sich auf die verteilte Anforderungsnutzung, die Suche nach Anforderungen und die Sicherheit in Bezug auf Freigaben und Änderungen. Dazu erarbeitete sie auch entsprechende Strategien.

2.2.9 Anforderungsmodellierung nach Gebauer

GEBAUER /18/ beschäftigt sich mit der kooperativen Produktentwicklung auf der Basis von verteilten Anforderungen. Ein zentraler Bestandteil seines Konzepts sind Anforderungsnetze und Anforderungsbibliotheken, die hierarchisch geordnet sind. Diese Ordnung kann sowohl in Bereichen als auch in Disziplinen erfolgen; sie können auch unternehmensspezifisch oder produktspezifisch erstellt werden. Sie dienen als Basis für die Auswahl oder Definition der Anforderungen an ein konkret zu entwickelndes Produkt. In den Anforderungsbibliotheken sind vordefinierte Anforderungsmuster gespeichert, die abhängig vom Produktentwicklungskontext ausgewählt werden können. Der Autor definiert Anforderungsmuster als Mengen von lösungsneutralen, d. h. noch keinem Produkt zugeordneten Anforderungen, die durch Beziehungen miteinander verknüpft sind. Die Beziehungen verbinden die Anforderungen zu so genannten semantischen Anforderungsnetzen, in denen einerseits Anforderungen die Knoten

abbilden und andererseits Anforderungsabhängigkeiten die Kanten des Netzes darstellen. Die semantischen Anforderungsnetze der Anforderungsmuster können zu größeren semantischen Netzstrukturen zusammengefügt werden. Sie sind in Bibliotheken abgelegt, und repräsentieren dabei lösungsneutral das gesammelte Wissen über die Zusammenhänge und Abhängigkeiten in Anforderungsmengen. Er entwickelte eine softwaretechnische Unterstützung (DIICAD-AES), die eine Weiterentwicklung des bestehenden Prototyps ist. Dabei sollen die Anforderungen strukturiert erfasst, modelliert und mit einer generischen Bauteilstruktur verknüpft werden.

2.2.10 Anforderungsklä rung nach Jung

JUNG /19/ befasst sich in seiner Arbeit mit der Anforderungsklä rung in interdisziplinärer Entwicklungsumgebung. Er entwickelte dabei ein Vorgehen zur relationalen iterativen Anforderungsklä rung. In seinem Ansatz betrachtet er das Produkt nicht nur in seiner Gesamtheit, sondern fordert unter Einbindung bekannter Methoden den Entwickler auf, zusammen mit Experten das zu entwickelnde Produkt iterativ und systematisch in seine einzelnen Elemente zu zerlegen und zu strukturieren. Dieses Vorgehen soll dem Entwickler eine Möglichkeit geben, Anforderungen in fremden Fachgebieten unter kontinuierlicher Einbindung eines Experten zu klären. Durch die systematische Abfrage der Relationen nach Anforderungen bei jeder weiteren Detaillierung des Systems erreicht das Anforderungssystem iterativ eine immer höhere Detaillierungsstufe. Damit wird angestrebt, dass die abgeleitete Anforderungsliste für den gerade folgenden Konstruktionsabschnitt vollständig ist.

Für seinen Ansatz schlägt er ein Softwarewerkzeug vor, das das Analysieren der Relationen zwischen den Systembestandteilen ermöglichen soll, um daraus Anforderungsfelder abzuleiten und die dazugehörigen einzelnen Anforderungen zu klären. Dabei ist zu kritisieren, dass in der Darstellung Anforderungen mit Bauteilen vermischt werden; das kann in vielen Fällen für den Entwickler irreführend sein.

2.2.11 Erfassung und Integration von Anforderungen nach Jörg

JÖRG /20/ stellt die Erfassung und Integration von Produkthanforderungen vor. Dabei hat er einen Ansatz entwickelt, mit dem es möglich ist, Anforderungen aus Texten (Dateien) zu extrahieren, zu interpretieren und in vereinzelter Form in das Anforderungsmodell zu überführen. Sie sollen dann in einer Anforderungsbibliothek produkt- und baugruppenbezogen erfasst werden. Er kommt zu der Erkenntnis, dass diese Anforderungen nicht 100%ig richtig erfasst wurden, weshalb sie auf ihre Richtigkeit zu

prüfen sind. Ferner hat der Autor ein Konzept für ein Anforderungsmanagementsystem vorgeschlagen, das Informationen mit anderen Systemen (CAD bzw. PDM) austauschen kann. Sein Werkzeug verwaltet Anforderungen mit Hilfe einer Baumstruktur. Der entwickelte Ansatz gilt für komplexe Produkte. JÖRG geht davon aus, dass Anforderungen weiterhin in Office-Dateien erfasst werden. Seine Softwarekomponente soll diese Anforderungen aus den Office-Dateien nur für Entwicklungszwecke in ein Anforderungsmanagementsystem extrahieren. Bei dem Extrahieren entsteht ein erheblicher Arbeitsaufwand. Strebt man jedoch an, die Anforderungen nur in einem einzelnen System zu erfassen und zu verwalten, kann jener Aufwand mit Sicherheit reduziert werden.

2.2.12 Weitere Ansätze zur Anforderungserfassung

Während im Maschinen- und Anlagenbau die Erstellung von Lasten- und Pflichtenheften zur Beschreibung der notwendigen Anforderungen genutzt werden /4/, wurde im Bereich der Softwareentwicklung die Methoden des Anforderungsmanagements erarbeitet. So ist auch verständlich, dass die beinhalteten Methoden an die Bedürfnisse der Entwicklung von Software angepasst sind. Zu diesen Ansätzen gehören „*Requirements Engineering und –Management*“ nach RUPP /21/ sowie die objektorientierte Modellierung der Anforderungen nach anderen Autoren /22/, /24/, /23/, /24/, /25/, /26/. Da die Ansätze nur für die Softwareentwicklung erarbeitet wurden, sind sie für den Einsatz in der Mechanik, Regelung und Elektrotechnik nicht direkt geeignet, so dass sie für die Anwendung in diesen Bereichen geändert werden müssen. Weiterhin existieren auf dem Markt Werkzeuge zum Anforderungsmanagement, wie *Doors* /27/, *RationalRequisite* /28/ und *RDT* /29/, die als Insellösungen zu sehen sind. Davon unterscheidet sich mit einem integrativen Ansatz *TeamcenterRequirements* /30/, mit dessen Hilfe man Anforderungen und Bauteile im PDM-System miteinander verlinken kann. Diese Links sind aber softwaretechnisch nicht auswertbar, und es fehlt die Möglichkeit der Funktionsmodellierung. Somit kann die durchgängige Produktentwicklung nicht unterstützt werden. In diesem Aspekt ist das *FOD-System* am weitesten fortgeschritten /31/.

2.3 Funktionserfassung und -definition

2.3.1 Einleitung

Zum Verlauf der Präzisierung und Konkretisierung der Aufgabenstellung bildet die Produktfunktion eine Schnittstelle zwischen der Formulierung des Auftrages mit den Anforderungen und der zu erarbeitenden Konstruktionslösung mit den Bauteilen. Durch die Erfassung und Explizierung der Produktfunktionen werden die Entwicklungsabsichten besser spezifiziert und dokumentiert, wodurch der Entwicklungsverlauf des Produkts besser nachvollziehbar wird. Die Erarbeitung der Funktionsstruktur eines Produkts erleichtert die Produktdokumentation und das Finden von optimalen Lösungen, weil durch die Strukturierung die Konstruktionslösung systematisch und mit weniger Fehlern sowie Entwicklungsiterationen erarbeitet werden können. Weiterhin wird durch die Dekomponierung der Hauptfunktionen eine Wiederverwendung von Teillösungen begünstigt.

2.3.2 Pahl und Beitz

Für den Teilschritt „*Ermitteln der Funktionen*“ definieren PAHL/BEITZ /7/ bei ihrem Vorgehen zuerst den Begriff „*Funktion*“ und geben dann Empfehlungen zum Aufstellen von Funktionsstrukturen, die mit einigen Beispielen illustriert werden. Sie beschreiben eine Funktion über „*Haupt- und Zeitwort*“ und unterscheiden zwischen Gesamt-, Haupt- und Nebenfunktionen. Die Funktion ist für sie eine Formulierung der Aufgabe auf einer abstrakten und lösungsneutralen Ebene. Als Eingangs- und Ausgangsgrößen einer Funktion wird der Umsatz von Energie, Stoff und Signal bezeichnet. Weiterhin schlagen die Autoren vor, die Funktionsstruktur mit Graphen zu beschreiben, in denen die Struktur, die Reihenfolge der Funktionen sowie der Fluss des Umsatzes dargestellt wird.

2.3.3 Funktionsermittlung nach Roth

ROTH /8/ erarbeitete mehrere Hilfsmittel zum „*Ermitteln von Funktionen und deren Strukturen*“, die er tabellarisch darstellt. Der Autor definiert neben den Gesamt-, Haupt- und Nebenfunktionen auch Aufgabe- und Sollfunktionen, die den Stoff, die Energie und die Informationen umsetzen. Er schlägt vor, die Funktionen in drei grundlegenden Funktionsstrukturen (allgemeine Funktionsstruktur, Intensitäts-, Qualitätsgrößen- und logischen Funktionsstruktur) mit Hilfe von Graphen zu modellieren. Die allgemeine Funktionsstruktur ist dazu da, auf möglichst hoher Abstraktionsstufe etwas über die Funktionen des zukünftigen Produkts auszusagen. Sie soll auch etwas über

die Zuordnung der Gesamtaufgabe zur Gesamtfunktion, Teilaufgabe zur Teilfunktion, über die festgelegte Flussrichtung sowie über eine mögliche Schaltordnung aussagen. Für ihre Erstellung sollen nur fünf allgemeine Funktionen benutzt werden. Überdies schreibt er, dass eine Intensitäts- und Qualitätsgrößen-Funktionsstruktur dazu dient, ein System von Kräften, Momenten, Geschwindigkeiten, Volumenstrom, Druck, Strom, Spannung oder Ladung zu beschreiben. Dazu hat er auch eine entsprechende Syntax erarbeitet. Die logische Funktionsstruktur sollte abstrakter als die allgemeine Funktionsstruktur sein und nicht auf Materie oder Energie, sondern nur auf Informationen angewiesen sein. Sie gründet auf Booleschen Verknüpfungsfunktionen und wird auf logischen Elementen realisiert, die auch mechanischer Natur sein können.

2.3.4 VDI-Richtlinien 2221/2222

In dem Arbeitsschritt nach VDI 2221 „*Ermitteln von Funktionen und deren Strukturen*“ sollen die Funktionen eines Produkts festgelegt zudem Überlegungen angestellt werden, in welcher Form diese in Unterfunktionen unterteilbar sind. Dabei ergibt sich in der Regel eine hierarchische Struktur aus Funktionen, die über Funktionsflüsse miteinander in Beziehung stehen. Die Funktionsmodellierung der VDI-Richtlinie 2222 Blatt-1 enthält eine Detaillierung mit zwei weiteren Schritten: *Abstraktion der Gesamtfunktion* und *Aufstellen der Funktionsstruktur*. Im ersten Schritt wird aus der Problemspezifikation (Anforderungsliste) die Gesamtfunktion abgeleitet. In der Regel soll diese jedoch sehr konkret sein und schränkt dadurch den möglichen Lösungsraum stark ein. Aus diesem Grund wird die Gesamtfunktion so weit abstrahiert, bis eine annähernd lösungsneutrale Beschreibung vorliegt. Sie soll in mehrere Unterfunktionen dekomponiert werden. Dies geschieht Schritt für Schritt auf mehreren Abstraktionsebenen. Nachdem nun die Gesamtfunktion abstrahiert und zusätzliche Funktionen spezifiziert wurden, sollen diese mit Hilfe verschiedener Flussarten miteinander verknüpft werden. Bei den Flussarten handelt es sich um Stoff, Energie und Informationen.

2.3.5 Funktionsmodellierung nach Ehrlenspiel

EHRENSPIEL /4/ schlägt für den Arbeitsschritt „*Ermitteln von Funktionen und deren Strukturen*“ gemäß VDI-Richtlinie 2221 eine Methode zur inhaltlichen Strukturierung nach Funktionen vor. Dabei geht er auf die Funktionsformulierung und auf das Erstellen von Funktionsstrukturen ein.

Eine Funktion ist für Ehrlenspiel eine lösungsneutrale Formulierung des gewollten Zwecks eines technischen Gebildes, welche Energie, Stoff und

Signal umsetzt. Der Autor führt neben den bekannten Gesamt-, Teil-, Haupt- und Nebenfunktionen noch die Begriffe Elementar-, Folge- und Geltungsfunktion ein. Dabei empfiehlt er, eine Funktion mit Substantiv und Verb zu formulieren, wobei das Substantiv das Umsatzprodukt und das Verb die Eigenschaftsänderung beschreiben sollen.

Eine Funktionsstruktur ist ein Mittel, um den Zweck des Systems zu beschreiben; eine weitere Unterscheidung macht der Autor nicht. Er schlägt vor, den Stoff-, Energie- und Signalfloss bei der Modellierung für die bessere Darstellung in Extra-Diagramme zu trennen. Für die Erstellung der Funktionsstruktur benennt er sechs lösungsneutrale und zwei logische Operationen sowie drei Arten von Relationen. Die Modellierung ist mittels Graphen durchzuführen.

2.3.6 Funktionsmodellierung und Lösungsfindung nach Huang

HUANG /32/ beschäftigt sich mit der wissensbasierten Funktionsstrukturmodellierung und Lösungsfindung mechatronischer Produkte. In ihrer Arbeit hat sie ein formalisiertes Vorgehen vorgeschlagen. Mit Hilfe ihres Vorgehens soll die Komplexität der Funktionsstrukturmodellierung durch *Kanonische Funktionen* reduziert werden. Die Autorin stellt die Modellierung *Kanonische Funktionsstruktur* vor und beschreibt, wie die Übergänge zwischen den einzelnen Funktionsstrukturen durchzuführen sind. Dabei hat sie insbesondere den Übergang zwischen der allgemeinen Funktionsstruktur und der speziellen Funktionsstruktur thematisiert. Außerdem hat sie Regeln zur Optimierung dieser Funktionsstruktur vorgeschlagen, insbesondere Regeln für eine Optimierung mittels informationsverarbeitender Funktionen.

Ein weiteres Ergebnis ihrer Arbeit ist eine konstruktionsgerechte Klassifizierung energetischer Größen, mit deren Hilfe die Kausalität verschiedener Fachdisziplinen sowie die Verbindungen zu den stofflichen Eigenschaften in einheitlicher, abstrahierter Form dargestellt werden können. Dazu hat Huang eine softwaretechnische Unterstützung – basierend auf einem kommerziellen Werkzeug – vorgeschlagen.

2.3.7 Funktionsorientierter Entwurf nach Gehrke

GEHRKE /33/ widmet sich dem Entwurf mechatronischer Systeme auf der Basis von Funktionshierarchien und Systemstrukturen. Im Rahmen dieser Arbeit hat er den Zusammenhang zwischen der Funktionshierarchie und der Systemstruktur auf der Grundlage von Graphtransmutationsregeln formal spezifiziert. Im Übrigen hat er die Modellierung der funktionalen Anforderungen mit Hilfe einer Funktionshierarchie und dem systematischen

Übergang zur Systemstruktur dargestellt. Darüber hinaus entwickelte er einen Suchalgorithmus, der in der Lage sein soll, für eine gegebene Funktion alle Systemelemente zu ermitteln, die diese Funktion realisieren können. Die Grundlage dafür bildet eine entsprechend große Menge von Systemelementen, auf die der Algorithmus zugreifen kann. Gehrke schlug eine softwaretechnische Unterstützung seines Ansatzes vor. Dabei hat er die statische Struktur mit Hilfe von UML-Klassendiagrammen und das Verhalten mit Hilfe umgangssprachlich definierter Regeln beschrieben.

2.3.8 Funktionsgetriebene Konstruktion nach Leemhuis

LEEMHUIS /34/ hat einen Ansatz erarbeitet, der es ermöglichen soll, auf der Grundlage der Konstruktionsmethodik sowie unter Nutzung einer geeigneten Systemumgebung Teilaufgaben der Produktentwicklung zu optimieren. Dies ist die Erweiterung des Ansatzes aus dem Forschungsprojekt iViP /11/ um Aspekte der Wissensmodellierung. Die Autorin versuchte, dabei sowohl das allgemeine als auch das firmenspezifische Konstruktionswissen mit Hilfe von Produktfunktionen in rechnerverarbeitbarer Form abzubilden. Dazu hat sie Konzepte eines Wissensmodells und eine Vorgehensweise vorgeschlagen sowie die rechnerunterstützte Abbildung und Nutzung des Ansatzes aufgezeigt.

2.4 Produktentwicklung in KMU

Kleine und mittelständische Unternehmen (KMU) sind solche Betriebe, die bestimmte Schwellenwerte nicht überschreiten. Nach der Definition der EU-Kommission /35/ haben diese Firmen nicht mehr als 250 Mitarbeiter und erreichen einen Jahresumsatz von weniger als 50 Mio. € bzw. eine Jahresbilanzsumme von nicht über 43 Mio. €. Demgegenüber definiert das Institut für Mittelstandsforschung, Bonn, spezifisch für Deutschland mittelständische Unternehmen mit 500 Mitarbeitern als Obergrenze sowie einen Umsatz bis 50 Mio. € und liefert dazu ausführliche Statistiken /36/.

Die KMU spielen sozial, politisch und wirtschaftlich eine sehr wichtige Rolle, da sie 99% aller Unternehmen in der EU darstellen, etwa 65 Millionen Arbeitsplätze bieten und eine wichtige Quelle für unternehmerische Initiativen und Innovationen sind /37/. In Deutschland beschäftigen die KMU 70,8 % der Erwerbstätigen /36/; somit finden mehr als zwei Drittel der arbeitenden Bevölkerung in mittelständischen Unternehmen ihren Lebensunterhalt. Aufgrund ihrer Flexibilität können KMU auf konjunkturell bedingte Nachfrageschwankungen kurzfristiger reagieren als Großunternehmen.

In KMU werden die Produktentwicklungen überwiegend durch den Auftrag eines Kunden angestoßen. Dabei wird möglichst auf vorhandene

Konstruktionen zurückgegriffen, um die geringen Entwicklungskapazitäten nicht zu überfordern. Sofern es überhaupt eine Entwicklungsabteilung gibt, ist diese in der Regel sehr klein (/38/ S. 75).

Besonders bei den kleinen KMU fehlt es oft am Entwicklungsmanagement bzw. an übergeordneter Leitungsebene, welche Fehler korrigieren oder die Vorgehensweise der Entwickler beeinflussen können (/38/ S. 66). Deswegen liegen das Projektmanagement und gleichzeitig die Festlegung der Vorgehensweise mit der Anwendung von Methoden in den Händen der Entwickler. Durch die starke Einbindung in das operative Geschäft finden sie im Gegensatz zu Großunternehmen selten die Zeit, ihre Entwicklungs- als auch Fertigungsprozesse fortlaufend zu verbessern und auf dem aktuellen Stand der Technik zu bleiben (/39/ S. 39).

ENGELMANN kam bei der Auswertung seiner Befragung zu dem Ergebnis, dass die kleinen Entwicklungsabteilungen der KMU die Konstruktionsmethodik oft nicht anwenden (/38/, S. 75). Wenn sie Methoden anwenden, dann bevorzugen sie jene Methoden, die wenig aufwendig sind und ein gutes Verhältnis zwischen Aufwand und Nutzen haben. Dabei führen diese nicht unbedingt zu optimalen Ergebnissen (/38/, S. 46).

Auch HESSLING postuliert, dass die Strategien, Methoden und Hilfsmittel zur Produktentwicklung und insbesondere für die integrierte Produktentwicklung in KMU nicht nur angepasster, sondern vor allem auch einfacher ausgestaltet werden müssen (/39/, S. 42). Die Wahrscheinlichkeit des Einsatzes von Methoden nimmt mit der Unternehmensgröße zu und ebenso die Zeit, die zur Erfüllung der Aufgabe zur Verfügung steht. Der Einsatz von Methoden führt in vielen Fällen zu positiven Auswirkungen auf Produktqualität, Entwicklungskosten und Entwicklungszeit sowie den Grad der Erfüllung der Kundenanforderungen. Zudem führt der Methodeneinsatz zu gut formalisierten, organisierten und dokumentierten Entwicklungsprozessen (zit. n. /39/, S. 41).

Bei der Produktentwicklung in KMU entstehen Probleme häufig durch ungeeignete Vorgehensweisen beim Management, nicht optimierte Entwicklungsprozesse, Mangel an qualifiziertem Personal, fehlende Disziplin zum Befolgen von Unternehmensrichtlinien und ungenügendes Methodentraining. Es fehlt auch der Kundenkontakt hinsichtlich der späteren Nutzung der Produkte, so dass nur wenige Konstrukteure nach Beendigung der Entwicklung noch mit dem Produkt involviert sind (/39/, S. 42).

Angesichts der beschriebenen Situation in vielen KMU werden daher große Potenziale für die Erarbeitung von KMU-gerechten Entwicklungsmethoden erkundbar. Dabei müssen die Methoden und Werkzeuge vor allem einfach spezifiziert und gestaltet werden.

2.5 Erfahrungsgrundlage

Die vorliegende Arbeit basiert auf den Erfahrungen und Erkenntnissen, die der Autor im Rahmen seiner wissenschaftlichen Tätigkeit am Lehrstuhl für Industrielle Informationstechnik am Institut für Werkzeugmaschinen und Fabrikbetrieb der Technischen Universität Berlin gewinnen konnte.

Die Möglichkeit, Problembereiche im Themengebiet der rechnerunterstützten Produktkonzeption zu erkennen, Zielsetzung und Lösungsansätze daraus abzuleiten und zu evaluieren, bot dem Autor in besonderem Maße das Forschungsprojekt für „integrierte Virtuelle Produktentstehung“ – iViP, Teilprojekt 3.2 „Funktionsorientiertes Entwerfen“². Im Rahmen dieses Teilprojekts hat der Autor den Prototyp eines Systems mitentwickelt, das auf den VDI-Richtlinien 2221/2222 basierend die Modellierung von Anforderungen, Funktionen und Baustrukturen in Hierarchieebenen und die Beschreibung von inneren oder äußeren Beziehungen der einzelnen Teilmodelle durch die Definition von Constraints unterstützt /40/, /41/, /42/. Wegen des großen Interesses an dem Softwareprototyp wurde das FOD-System von der Firma CADsys, Chemnitz, zum marktfähigen Produkt weiterentwickelt und vermarktet. Inzwischen gibt es zahlreiche industrielle Applikationen, die die ökonomische Relevanz dieses Systemkonzepts beweisen /43/.

Die weiteren Arbeiten sowohl in den industriellen Pilotprojekten des Systems als auch in den Forschungsprojekten, wie „Virtual Design and Automation Centre“ /44/, /45/, /46/ und „Mechatronik Plattform für anforderungsgesteuerte Prüfung und Diagnose“ /47/, gaben dem Autor ideale Bedingungen zur Identifizierung der Problemfelder zur Erfassung und Definition von Anforderungen und Funktionen sowie zur Erstellung von Produktmodellen bei mittelständischen Unternehmen. Dies hat die Entwicklung neuer Ansätze zur Lösung der identifizierten Probleme ermöglicht.

² Forschungsprojekt koordiniert durch das Fraunhofer IPK Berlin, gefördert durch das Bundesministerium für Bildung und Forschung, Projektnummer 02PL10xxx, 1998-2002

3 Präzisierung der Aufgabenstellung

3.1 Motivation und Zielsetzung

Bei der Produktplanung wird das herzustellende Produkt durch Anforderungen definiert. Beim Konzipieren wird versucht, eine prinzipielle – teilweise abstrakte – Lösung zu finden, die diese Anforderungen (insbesondere Funktionsanforderungen) am wahrscheinlichsten und optimalsten erfüllt /4/. Dabei können verschiedene Konzepte bzw. -modelle des Produktes erstellt, beurteilt und geändert werden, ohne dabei mehr Kosten zu verursachen, die bei eventueller Änderung des Produkts erst in der Phase der Herstellung oder der Montage entstehen würden. Deshalb ist es von entscheidender Bedeutung, effektive Methoden und Werkzeuge für die Produktkonzeption zu erarbeiten und anzuwenden, welche die Anforderungs-, Funktions- und Funktionsträgerdefinition wirksam unterstützen.

Methoden zur Unterstützung der Produktentwicklung, insbesondere der Konzeption und Konstruktion von Produkten, führen immer dann nur zu einer echten Verbesserung der Ergebnisse, wenn sie vom Entwickler akzeptiert und auch konsequent angewendet werden. Da die Entwicklungsmethodik zur erfolgreichen Produktentwicklung führen kann, aber nicht unbedingt führen muss /3/ und weil die Leistung der Methodik immer wieder hinter den Erwartungen des Anwenders zurück bleibt /3/, wird sie oft nicht akzeptiert und deshalb nicht angewendet /2/. Deswegen ist es notwendig, weiter an der Verbesserung der Anwendbarkeit der Produktentwicklungsmethoden zu arbeiten.

Zur Optimierung des Produktentwicklungsprozesses ist eine informationstechnische Verknüpfung der Einzelaufgaben in Prozessketten und eine Nutzung von Produktmodellen erforderlich /1/. Daher muss eine effektive Produktkonzeption auch die Nutzung von Software berücksichtigen. Die Ansätze aus der Produktentwicklung schlagen oft eine Softwareunterstützung vor (vgl. Kap. 2.1-3.); dabei impliziert der Begriff „*Unterstützung*“ die höhere Priorität der zugrunde liegenden Methodik des Vorgehens über die Softwarekomponenten. Dies resultiert oft in großen Schwierigkeiten bei der Entwicklung der passenden informationstechnischen Werkzeuge. Aufgrund der allgegenwärtigen Anwendung des Computers ist die Zeit gekommen, eine neue Form der Rechnerintegration anzustoßen. Dies unterscheidet sich durch die Gleichberechtigung des Vorgehens und der Software während der Entwicklung eines Ansatzes, so dass bei der Erarbeitung der Methodenschritte immer die softwaretechnischen Aspekte berücksichtigt werden.

Somit liegt das Ziel der Arbeit in der Entwicklung eines Ansatzes, bei dem das Vorgehen und die Software miteinander harmonisieren. Dadurch soll neben der besseren Effektivität auch die Akzeptanz der rechnerintegrierten Methode bei dem Anwender erreicht werden.

KMU haben für die Volkswirtschaft insofern eine herausragende Bedeutung, als aus ihnen insbesondere flexible Quellen für unternehmerische Initiativen und Innovationen hervorgehen (vgl. Kap. 2.4.). Jedoch verfügen KMU in aller Regel nur über kleine Entwicklungsabteilungen, die vorzugsweise nach jenen Vorgehensweisen arbeiten, die wenig aufwendig sind und ein gutes Verhältnis zwischen Aufwand und Nutzen haben (vgl. /38/, S. 46) und die Konstruktionsmethodiken oft nicht anwenden (vgl. /38/, S. 75). Es existieren auch kaum Entwicklungsansätze, die diese Anforderungen erfüllen und expliziert die spezifischen Vorgehensweisen in der Produktentwicklung in der KMU adressieren (vgl. Kap. 2.). Angesichts dieser Umstände ist das Ziel dieses Forschungsansatzes, eine einfach anwendbare, möglichst intuitive rechnerintegrierte Methode zu erarbeiten, die die spezifische Situation in den KMU berücksichtigt. Diese Methode soll das Verstehen der Problemstellung durch die Erarbeitung und Anwendung von rechnerinternen Produktmodellen fördern und die Lösungssuche durch Konkretisierung der Anforderungen mittels Funktionen und Funktionsträgern effektiv unterstützen. Sie soll einerseits ein methodisches Fundament zur rechnerintegrierten Konzipierung anbieten, andererseits aber den Abstand zwischen Theorie und Praxis in diesem Bereich möglichst gering halten. „Den Abstand gering halten“ bedeutet, dass die erarbeiteten generischen Anwendungssituationen, die auf einer Tätigkeits- und Objektebene beschrieben sind, auf konkrete betriebliche Anwendungssituationen in KMU sich effektiv und einfach abbilden lassen. Dadurch soll dieser Ansatz dem Entwickler in den Unternehmen zu einer zunehmend methodischen Problemlösefähigkeit verhelfen. Ein weiteres Ziel ist, die rechnerintegrierte Methode so zu erarbeiten, dass sie sowohl ihre Einführung als auch die stetige Weiterentwicklung der Vorgehensweise und des dazugehörigen Softwarewerkzeugs fördert.

Die gegenwärtigen wissenschaftlichen Ansätze, die sich mit Anforderungen beschäftigen, liefern viele methodische Hilfsmittel zur ihrer Ermittlung, Erfassung und Verwaltung. Die Vorgehensweisen und Randbedingungen in den KMU finden dabei jedoch kaum Berücksichtigung. Ferner befassen sich diese Ansätze mit der Definition von Anforderungen wenig, und wenn doch, dann nur auf der „Satzebene“ (vgl. Kap 2.2.). Eine Ausnahme bildet dabei ein Ansatz von RUPP /21/ zur Definition von funktionalen Anforderungen im Bereich der Softwareentwicklung, aber durch die Anwendung einiger weniger Anforderungselemente bleibt bei ihm ein großer Spielraum zur Fehldefinition und -interpretation insbesondere bei nichtfunktionalen Anforderungen. Aufgrund der beschriebenen Situation fehlt es dem Anwender an methodischen Mitteln, die ihm helfen, Anforderungen korrekt

und präzise zu definieren, und darüber hinaus auch an Kenntnissen, wie diese zusammengesetzt sind. Die genannten Umstände resultieren in den kaum zum Einsatz kommenden methodischen und softwaretechnischen Hilfsmitteln zur Anforderungsverarbeitung in den KMU. Da aber das herzustellende Produkt mittels Anforderungen definiert wird, sind diese von essentieller Bedeutung für die Produktentwicklung. Aus diesem Grunde besteht ein Ziel der vorliegenden Arbeit in der Entwicklung eines Vorgehens und eines Softwarewerkzeuges zur einfach gestalteten aber präzisen Erfassung, Definition und Modellierung von Anforderungen im Rahmen der rechnerintegrierten Produktkonzeption.

Da die Produktfunktion eine Schnittstelle zwischen den Anforderungen und den zu erarbeitenden Konstruktionslösungen bildet ebenso wie die Erarbeitung einer Funktionsstruktur eines Produktes das Finden von optimalen Lösungen erleichtert, ist diese von sehr großer Bedeutung für eine durchgängige Produktentwicklung. Die existierenden wissenschaftlichen Ansätze betrachten die Funktionsmodellierung fast nur aus der methodischen Perspektive. Sie schlagen oft eine informationstechnische Unterstützung vor (vgl. Kap. 2.3.), die aber mit Ausnahme des FOD-Systems leider in den KMU nicht angewendet wird. Während der durchgeführten Projektarbeiten – auch bei jenen unter Anwendung des o. g. FOD-Systems – konnte der Autor feststellen, dass die Funktionsmodellierung in gegenwärtiger Form mangels entsprechender Werkzeuge, wegen der für die Betroffenen kaum erkennbaren Vorteile der Anwendung und wegen des allgegenwärtigen Zeitdrucks einfach weggelassen wird. Diese Situation bringt später Inkonsistenzen bei den weiteren Entwicklungsarbeiten hervor, wodurch Arbeitsaufwand und Entwicklungskosten erhöht werden. Somit ist ein nächstes Ziel der vorliegenden Arbeit, im Rahmen der rechnerintegrierten Produktkonzeption einfache und effektive Hilfsmittel zur Erfassung, Definition und Modellierung von Produktfunktionen zu erarbeiten. Dabei wird auf die zukünftige Akzeptanz des Ansatzes bei den KMU besonders zu achten sein. Dies kann durch die Abbildung von Bauteilen und/oder Baugruppen in einem Funktionsmuster geschehen, um in den frühen Phasen die Produktfunktionen, Funktionsmerkmale und Produktstruktur zu erproben und die gesuchte Lösung effektiver einzugrenzen.

3.2 Abgrenzung

Die erarbeitete rechnerintegrierte Methode adressiert explizit die Unterstützung der durch einen Kundenauftrag initiierten Produktentwicklung. Typischerweise ist die Produktentwicklung in solchen KMU anzutreffen, in denen Maschinen, Gebrauchsgüter oder spezielle Produkte für den Kunden hergestellt werden.

Diese Methode ist besonders gut geeignet für eine gemischte Neu- und Anpassungskonstruktion, wie sie häufig bei Einzelentwicklungen und bei Kleinserienentwicklungen durch Baugruppenkonfiguration vorkommt. Bei einer sehr hohen Produktkomplexität oder in großen Entwicklungsabteilungen sind die Voraussetzungen für die Anwendung der Methode nicht gegeben. Die explizit gleichberechtigte Betrachtung des Vorgehens und des Softwaresystems unterscheidet diese Arbeit signifikant von den im Kap 2.1-3. vorgestellten Ansätzen. Eine solche Betrachtung bringt neue Erkenntnisse bezüglich der Erarbeitung von Produktentwicklungsmethoden.

Die Arbeit basiert auf dem im Leitprojekt iViP TP 3.2³ erarbeiteten Ansatz des FOD-Systems /11/; sie erweitert aber diesen um umfangreiche und präzise Möglichkeiten der Anforderungsdefinition und Produktfunktionsableitung, wodurch wesentliche Erleichterungen und eine Effektivitätssteigerung in der gesamten darauf folgenden Prozesskette der Produktentwicklung entstehen. Dies und die explizite Adressierung der KMU unterscheidet die Arbeit von den existierenden und im Kap. 2.2. dargestellten Lösungen.

Im Bereich der Funktionserfassung und -definition betrachtet die vorliegende Arbeit ausschließlich die Funktionshierarchie des Produkts, da diese keine festgelegte Reihenfolge (im Gegensatz zur Graphendarstellung) beinhaltet und somit als lösungsneutral angesehen werden kann. Durch die gleichzeitige Mitmodellierung der Funktionsträger beschreitet die Arbeit neue Wege, um den Nutzungsfaktor der Funktionsmodellierung erheblich zu steigern.

Mittels präziser Definition von Produkthanforderungen und -funktionen sowie mittels deren integrierter Modellierung wird eine bessere Dokumentation der Produkte erreicht. Auch die Kundenänderungen oder technologisch bedingten Änderungen können dadurch einfacher und schneller als bei den in der Ausgangssituation (Kap. 2.) vorgestellten Ansätzen vorgenommen werden.

Das erarbeitete Konzept eines Softwarewerkzeugs, das erstens die Auslegung der Benutzungsoberfläche beinhaltet und zweitens die Harmonisierung der Oberfläche mit der Vorgehensweise betrachtet, und die Funktionalitäten sowie die Bedienung des implementierten Prototyps setzen neue Maßstäbe in der Verbesserung der Akzeptanz eines wissenschaftlichen Ansatzes bei den Produktentwicklern.

³ „Funktionsorientiertes Entwerfen“ koordiniert durch das Fraunhofer IPK Berlin, gefördert durch das Bundesministerium für Bildung und Forschung, Projektnummer 02PL10xxx, 1998-2002

3.3 Formulierung der Anforderungen an eine rechnerintegrierte Produktkonzeption

Die folgenden Anforderungen an die rechnerintegrierte Produktkonzeption ergeben sich einerseits aus der Notwendigkeit, die dargestellten Defizite zu überwinden, und andererseits aus den Rahmenbedingungen der Arbeit mit einem Computer. Die formale und empirische Richtigkeit des Konzepts der rechnerintegrierten Methode sowie ihre Widerspruchsfreiheit und Konsistenz gehören zu den elementaren Anforderungen. Ein weiterer Aspekt ist die Forderung nach möglichst quantitativen, zuverlässigen und nachprüfbareren Ergebnissen, die nach Anwendung einer Methode oder Einsatz eines Systems zur Verfügung stehen /48/.

Da ein Entwicklungsprozess eine Präzisierung und Konkretisierung der Aufgabenstellung ist /4/, muss auch die Konzeption sich an diese Paradigmen halten. In vorliegender Arbeit dient die Formulierung der Anforderungen zur Konkretisierung jener in der Motivation und der Zielstellung eingeschlossener Inhalte.

Wie bereits im Kapitel 3. erwähnt, muss die rechnerintegrierte Konzeption die Entwicklung eines neuen Produkts und einer Produktpassung unterstützen. Sie muss auch in den mittelständischen Betrieben anwendbar sein.

Da eine rechnerintegrierte Methode die Nutzung eines Computers und rechnerinternen Produktmodells für die Berechnung und Darstellung der Arbeitsergebnisse voraussetzt, muss sie die Vorbereitung des Einsatzes und der Anwendung der Methode berücksichtigen. Es muss den Aufwand zur Umsetzung möglichst gering halten und sich an die vorhandenen Kapazitäten der KMU anpassen lassen. Eine Einführung darf nicht nur methodische Vorteile anstreben, sondern der für die Umsetzung nötige Aufwand muss durch die wirtschaftlichen Vorteile für das Unternehmen zu rechtfertigen sein.

Weiterhin muss die Vorgehensweise bei der Anwendung in KMU sich sehr flexibel an verschiedene Randbedingungen anpassen können. Die unkomplizierte Pflege der Einzelelemente des Ansatzes gehört zu den elementaren Forderungen an die rechnerintegrierte Produktkonzeption, um auf methodische und technologische Fortschritte entsprechend reagieren zu können. Außerdem muss sich die Methode systematisch auch bei den anwendenden KMU weiter entwickeln lassen.

Da die Vorgehensweise für die Nutzung durch einen Entwickler gedacht ist, muss sie für ihn verständlich, eindeutig, leicht erlernbar und einfach gestaltet sein, um ihm eine effektive und effiziente Arbeit zu ermöglichen.

Da der integrierte Einsatz mehr als nur eine normale softwaretechnische Unterstützung einer Vorgehensweise und die rechnerintegrierte Methode aus zwei Hauptbestandteilen zusammengesetzt ist, müssen Softwaresystem und Vorgehen gut zusammenpassen und vollständig integriert sein, um das große Potenzial der rechnerintegrierten Konzipierung ausschöpfen zu können.

Die rechnerintegrierte Methode ist so zu erarbeiten, dass sie die generischen Anwendungssituationen, die auf einer Tätigkeits- und Objektebene beschrieben sind, effektiv und einfach auf konkrete Prozessebene im KMU abbilden lässt.

Da das Softwaresystem mit der Vorgehensweise integriert sein muss, haben die Gestaltung der graphischen Benutzungsoberfläche und deren Bedienung mit den Konzipierungsschritten zu harmonisieren. Die Bedienung des Systems muss für die Anwender verständlich, eindeutig und leicht erlernbar sein. Dadurch soll sie eine effiziente und effektive Konzipierungsarbeit ermöglichen. Das System ist so zu entwickeln, dass es sich leicht administrieren lässt und wartungsarm ist. Wegen der eingeschränkten Ressourcen bei KMU ist dies eine unverzichtbare Forderung.

Weiterhin hat das Softwarewerkzeug eine einfach gestaltete, aber dennoch präzise Erfassung, Definition, Konkretisierung und Strukturierung von Anforderungen mit gegebenen Anforderungsbedingungen dem Entwickler zu ermöglichen.

Das System muss auch eine einfach gestaltete, präzise und eindeutige Formulierung und Modellierung von Funktionen unterstützen sowie deren Erfassung in Funktionsstrukturen ermöglichen. Außerdem hat das Softwarewerkzeug die Zuweisung der Funktionsträger zu entsprechenden Produktfunktionen zu unterstützen.

4 Entwicklung einer Methode zur rechnerintegrierten Produktkonzeption

4.1 Einführung

Die treffende Erarbeitung einer rechnerintegrierten Methode bedarf eines methodischen Fundaments, das sowohl Aktivitäts- als auch informationstechnische Aspekte der rechnerintegrierten Konzipierung eines Produkts entsprechend den formulierten Anforderungen berücksichtigt. Deshalb wurde in dieser Arbeit der metamethodische Ansatz zur Entwicklung von Ingenieurmethoden von WEIGT /49/ (Bild 4-1) zu Hilfe genommen. Er wurde allerdings um die für die vorliegende Forschungsarbeit notwendigen Aspekte der Rechnerintegration ergänzt.

Dieser metamethodische Ansatz nutzt eine informationszentrierte Sichtweise zur Methodenentwicklung und Prozessableitung und basiert auf der Identifikation der spezifischen Randbedingungen. Dementsprechend ist es möglich, sowohl technologische und organisatorische Aspekte, die die Definition und Optimierung von Prozessen zu beeinflussen, als auch psychologische Aspekte, die die Ausführung von Tätigkeiten betreffen, entsprechend zu berücksichtigen /49/.

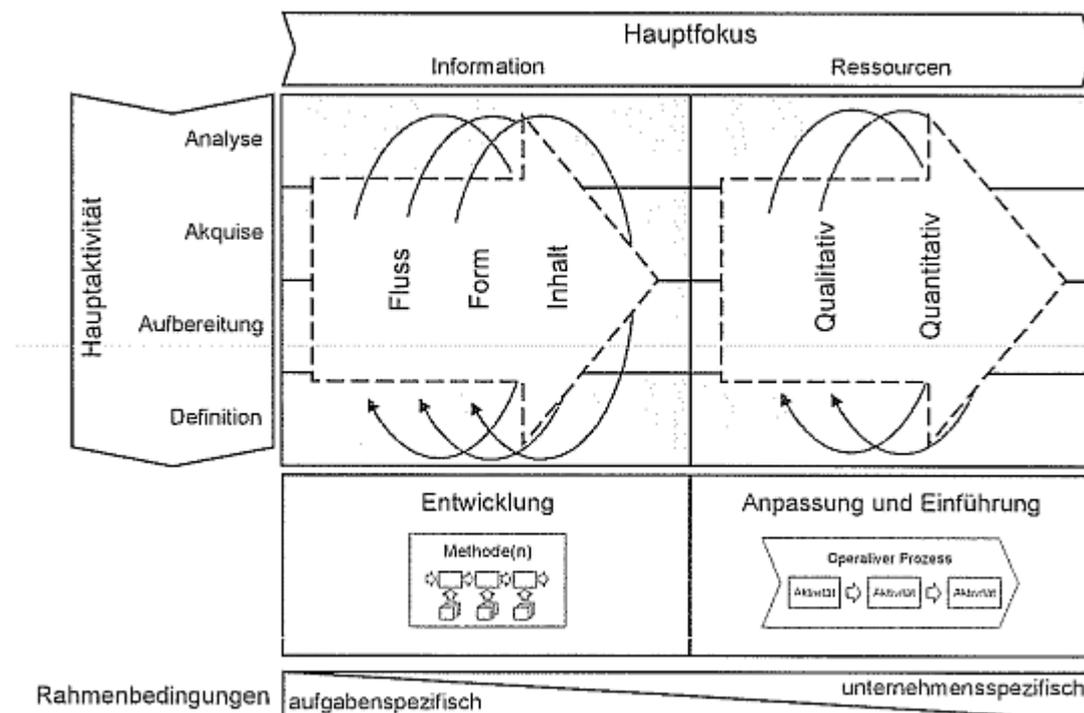


Bild 4-1: Metamethodisches Rahmenwerk von WEIGT /49/

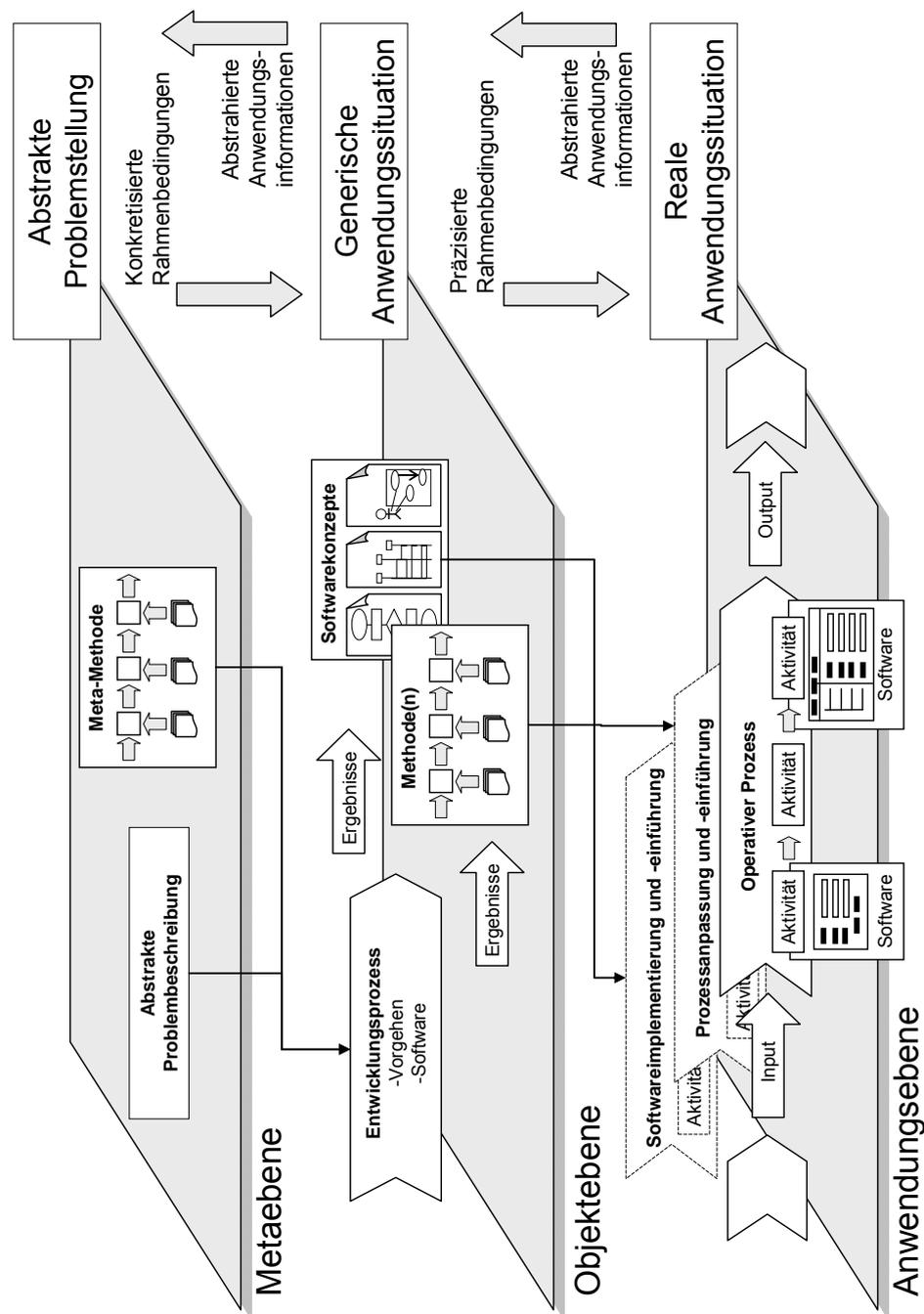


Bild 4-2: Kontext der Entwicklung der rechnerintegrierten Methode in Anlehnung an WEIGT /49/

Basierend auf diesem Ansatz werden in der vorliegenden Arbeit im Kontext zur Erarbeitung der rechnerintegrierten Methode verschiedene Abstraktionsebenen unterschieden, die im Bild 4-2 dargestellt sind. Auf der Metaebene wurde die Problemsituation bereits in vorigen Kapiteln definiert und Rahmenbedingungen gestellt.

Auf der Objektebene wird die rechnerintegrierte Methode basierend auf der generischen Anwendungssituation, wie z. B. „Konzipierung einer

Produktanpassung“, erarbeitet. Auf dieser Ebene werden zum einen verschiedene Aktivitätsfolgen definiert, die zusammengefasst ein Vorgehen entstehen lassen, zum anderen auch Schichten, Komponenten und Klassen, die ein Systemkonzept formen werden. Diese Ebene lässt sich mit Hilfe von Diagrammen darstellen. Auf der Anwendungsebene werden von Entwicklern konkrete Aufgabenstellungen und Informationen verarbeitet. Für die Lösung von realen Entwicklungsaufgaben werden aus der Methode konkretisierte Prozesse und ein implementiertes Softwaresystem abgeleitet.

4.2 Bestimmung der Maßnahmen zur Einführung des Einsatzes

4.2.1 Einleitung

Eine rechnerintegrierte Methode setzt die Nutzung eines Computers für die Berechnung und Darstellung der Arbeitsergebnisse voraus. Ein Computer verarbeitet Daten, die im Fall der rechnerintegrierten Produktkonzeption ein Computermodell eines Produkts bilden. Das rechnerinterne Produktmodell muss durch Abbildung oder Reduktion von Komplexität der Wirklichkeit des Produkts erstellt werden. Die Realisierbarkeit der Abbildung und ein Methodeneinsatz unter den ökonomischen Gesichtspunkten sind zunächst unsicher.

Die Abschätzung der Einsatzmöglichkeiten der Methode und die Erstellung eines Computermodells sollen somit das Ziel der Vorbereitung sein, die eine spätere produktive Anwendung der Methode ermöglicht. Dabei wird vorausgesetzt, dass ein Softwaresystem zur Verfügung steht, mit dem das Produktmodell erstellt werden kann.

Dabei sollen die technische und wirtschaftliche Machbarkeit des Vorhabens geklärt sowie ein Produkt(typ) als Computermodell abgebildet werden. Danach können die erstellten Anfangsdaten für einen bestimmten Fall der produktiven Anwendung vorbereitet werden.

Diese Vorbereitung treffen Mitarbeiter, die in die Entwicklung der relevanten Produkte involviert sind, da sie über das notwendige Know-how verfügen. Sie können in den Anfangsphasen durch externe Konsultanten unterstützt werden.

Wegen der Verfügbarkeit der notwendigen Daten zur Produktmodellierung und der Kommunikation zwischen den involvierten Mitarbeitern sollen Vorbereitungstätigkeiten vor allem in dem Unternehmen stattfinden, bei dem die Methode eingesetzt wird, und nicht bei dem, das die mögliche Consulting-Leistung erbringt.

Da die Gefahr besteht, dass das Computermodell nicht ausreichend detailliert erstellt werden kann, um inhaltlich und wirtschaftlich akzeptable Ergebnisse der Berechnung und Modellierung zu liefern, soll der Zeitpunkt der Vorbereitung so ausgewählt werden, dass die beteiligten Mitarbeiter ausreichend freien Raum für die Modellierungsaufgaben haben.

Ein Verlauf der Einführung des rechnerintegrierten Methodeneinsatzes wird im Bild 4-3 vorgeschlagen. Die einzelnen Tätigkeiten werden im Folgenden näher beschrieben.

Da nicht bei allen Produkten und Unternehmen der Einsatz der rechnerintegrierten Methode möglich oder auch sinnvoll ist, müssen die Produkte auf einzelne Merkmale hin untersucht werden.

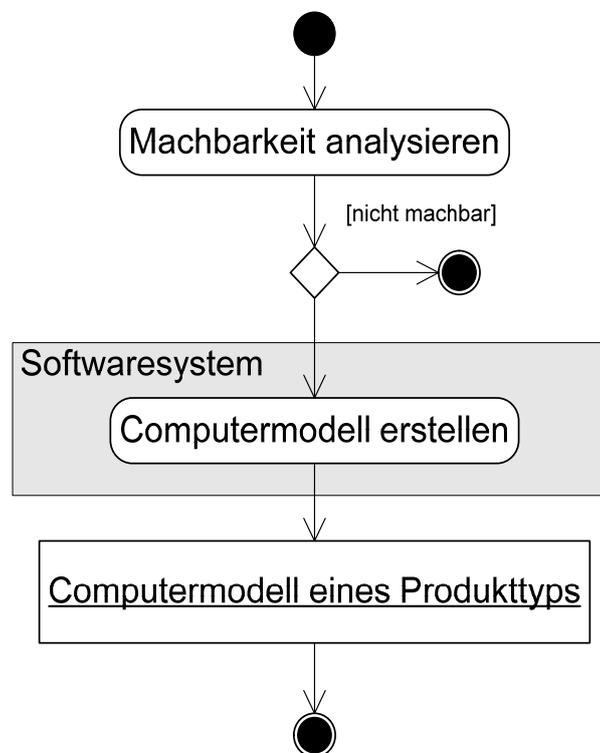


Bild 4-3: Einführung des rechnerintegrierten Methodeneinsatzes

4.2.2 Analyse der technischen Machbarkeit

Durch diese Analyse sollen mögliche Fehlentscheidungen im Bezug auf die Realisierung des Vorhabens vermieden werden. Besonders wichtig ist dabei die Erstellung des rechnerinternen Produktmodells. Dabei ist zu berücksichtigen, ob es sich um eine neue Entwicklung handelt oder die Abbildung eines vorhandenen Produkttyps ist. Es ist auch eine Übersicht über die Dokumentation und Konstruktionswissen zu erstellen, die dem Entwickler zur Verfügung stehen. Auch die Komplexität des Produkts findet hier Berücksichtigung mit der Folge, dass entschieden werden muss, ob ein

einziges oder mehrere Modelle zu erstellen sind, um das(den) Produkt(typ) abzubilden. Des Weiteren ist zu untersuchen, ob entsprechende Mitarbeiter vorhanden sind, die Konstruktionskenntnisse haben und die für die Modellierungsarbeit notwendige Softwarebedienung beherrschen. Auch das Vorhandensein entsprechender Software- und Rechnerinfrastruktur ist zu überprüfen.

4.2.3 Analyse der wirtschaftlichen Machbarkeit

Die Wirtschaftlichkeitsanalyse soll klären, ob die Erstellung des rechnerinternen Produktmodells, seine Anwendung sowie die gesamte Nutzung der rechnerintegrierten Methode wirtschaftlich sind. Dabei müssen die Kosten der Einführung, der Anwendungsvorbereitung und der Nutzung den Einsparungen gegenüber gestellt werden. Bei der Abschätzung der Einführung, der Vorbereitung und der Nutzung sind als Minimalanforderung folgende Faktoren zu berücksichtigen:

- Wie lange wird die Erstellung des rechnerinternen Modells dauern und wie viel wird sie kosten?
- Wie oft muss die Dokumentation für eine Anpassung oder Variante mit der Methode erstellt werden?
- Wie lange wird das Modell benutzt, um einen Produkttyp oder eine -serie zu erstellen?
- Wie lange und wie viel wird die Schulung der betroffenen Mitarbeiter dauern und kosten?
- Wie viele Lizenzen des Softwarewerkzeugs müssen gekauft werden?

Bei der Abschätzung der Einsparungen müssen folgende Aspekte beachtet werden:

- Wie lange dauert es, um eine Dokumentation konventionell oder mit Hilfe der Methode zu erstellen?
- Wie viele Mitarbeiter würden an der Erstellung der Dokumentation arbeiten, wenn sie konventionell oder rechnerintegriert erstellt wird?

Es muss auch geklärt werden, wie oft eine Variante oder Anpassung des Produkts verkauft wird. An der Wirtschaftlichkeitsanalyse sollen außer den Entwicklern und Konsultanten auch Mitarbeiter dem Vertrieb teilnehmen.

4.2.4 Erstellung eines rechnerinternen Produktmodells

4.2.4.1 Einleitung

Ein rechnerinternes Modell, das das Konzept eines Produkt(typs) darstellt, kann als eine Drehscheibe oder eine Basis für die rechnerintegrierte Methode angesehen werden.

Dieses Konzeptmodell soll dem Vertrieb und den Entwicklern zu der effizienten Berechnung der notwendigen Daten für ihre zukünftigen Tätigkeiten dienen. Seine Anwendung führt zur Steigerung der Arbeitseffektivität. Damit sollen die Mitarbeiter sehr schnell u. a. die Realisierbarkeit eines Auftrags beurteilen, ein Angebot erstellen und die Daten für die Konstruktion und Herstellung des bestellten Produkts berechnen können. Das rechnerinterne Modell eines Produkttyps kann die Verwendung von mehreren einzelnen Produktmodellen überflüssig machen, was die Transparenz der Daten sowie die Sicherheit der Produktentwicklung erhöht. Um die spezifischen Funktionen oder Funktionsmerkmale des Produkts zu untersuchen, können in dem rechnerinternen Modell zu den Funktionen einzelne Bauteile oder Baugruppen – die Funktionsträger – zugeordnet werden. Ein in dieser Weise erstelltes Konzeptmodell, mit dem das Zusammenspiel neuer Funktionen oder Komponenten überprüft wird, kann als erste Erprobungsstufe gesehen werden.

Der hier beschriebene Schritt soll erst dann unternommen werden, wenn es sichergestellt wurde, dass die Methode unter gegebenen Umständen in dem bestimmten Unternehmen machbar ist und wirtschaftlich eingesetzt werden kann. Außerdem müsste die Entscheidung der Verantwortlichen vorliegen, die die Methode einführen.

Das Konzeptmodell besteht aus einem Produkthanforderungsmodell und einem Produktfunktionsmodell, ergänzt um die Funktionsträger und liefert keine explizite visuelle Darstellung der geometrischen Gestalt. Diese soll später in der Konstruktionsphase mit Hilfe eines CAD-Systems erstellt werden.

Die Erstellung eines rechnerinternen Konzeptmodells soll iterativ erfolgen. Begonnen wird mit der Erstellung eines Anforderungsmodells, dem die Erstellung des Produktfunktionsmodells folgt. Bei den Modellierungsarbeiten muss am Anfang u. a. geklärt werden, welche Startparameter der Kunde im Auftrag vorgibt bzw. welche für die Berechnung von Varianten oder Anpassungen benötigt werden. In einem weiteren Schritt sind die Gültigkeitsbereiche der Parameter für die Modellierung zu bestimmen. Falls das eine Neu-Entwicklung ist, wird eine Anforderungsermittlung

durchgeführt. Der Verlauf der Erstellung eines rechnerinternen Modells, das ein Konzept eines Produkt(typs) darstellt, wird im Bild 4-4 gezeigt.

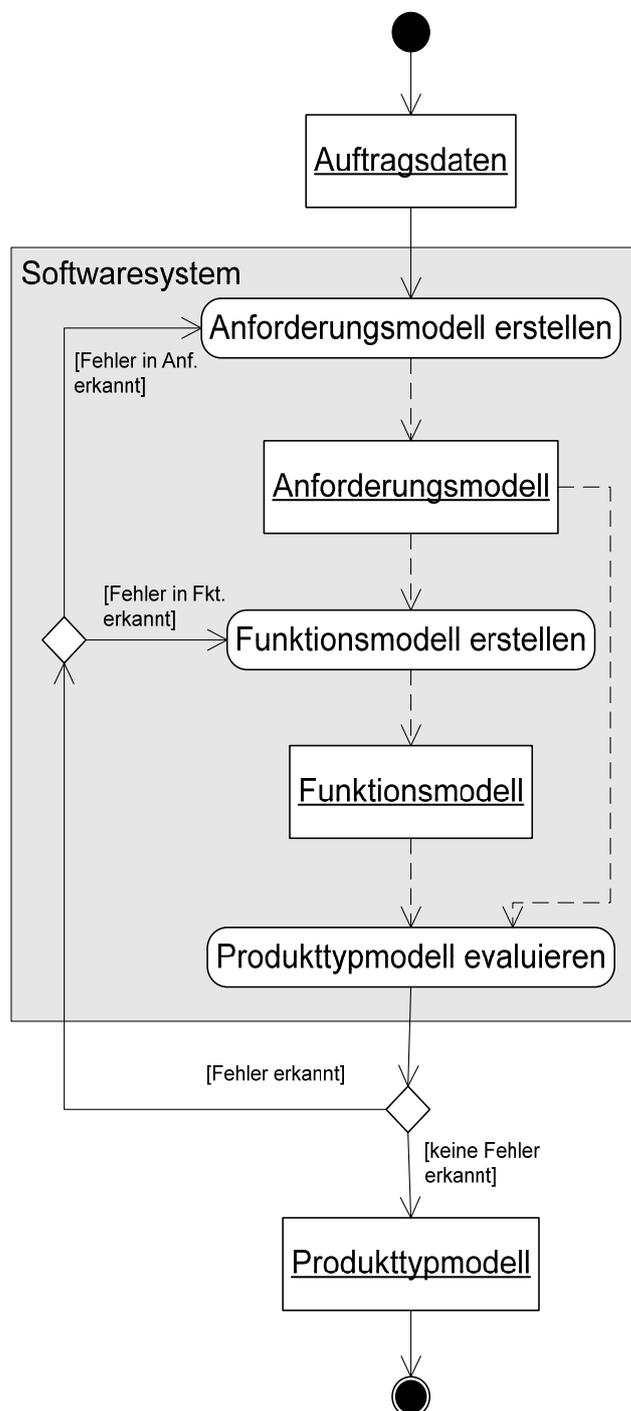


Bild 4-4: Erstellung eines rechnerinternen Konzeptmodells

Im Falle eines existierenden Produkt(-typs) sind die Produkthanforderungen analog auf dem Papier oder auch digital mit Hilfe eines Standard-Office-Programms erfasst. Bei einem neuen Produkt(-typ) existieren die Wünsche und Forderungen des Kunden im Bezug auf die zu erfüllenden Eigenschaften und Produktfunktionen teilweise im Auftrag oder auch

fragmentarisch in den Notizen der Entwickler. Die Verwaltung der so gesammelten Anforderungen ist ineffektiv. Diese können informationstechnisch nur schwer weiter verarbeitet werden, und deren Darstellung ist auch bei einer größeren Anzahl für den Entwickler nicht übersichtlich genug.

4.2.4.2 Erstellung eines Anforderungsmodells

Eine Hilfestellung für die Probleme liefert eine hierarchische Darstellung, in der alle Anforderungen, die für eine effektive Anbieterstellung, Produktentwicklung und -herstellung notwendig sind, strukturiert und konsistent in einem dafür geeigneten Editor erfasst werden können. Dadurch besteht die Möglichkeit, eine effektive und effiziente Verarbeitung der Anforderungen zu erreichen.

Es existieren bereits Ansätze zur Anforderungserfassung und -definition. Diese behandeln jedoch das Thema entweder aus der Sicht der Softwareentwicklung oder bleiben auf der „*Satzebene*“ (vgl. Kap. 2.2), was für eine präzise und eindeutige Definition sowie für eine effektive informationstechnische Verarbeitung der Anforderungen nicht ausreichend ist. In der vorliegenden Arbeit wird ein neuer Ansatz vorgeschlagen, Produkthanforderungen auf einer detaillierten „*atomaren*“ Ebene zu definieren, um der effektiven Produktentwicklung gerecht zu werden.

Die Erstellung eines Anforderungsmodells mittels eines Softwarewerkzeugs wird im Bild 4-5 dargestellt. In diesem Zusammenhang sind folgende Tätigkeiten durchzuführen:

Kategorien definieren

Bei Produkten, die aus einem oder mehreren Bauteilen bestehen, ist es möglich, die wenigen Anforderungen so aufzulisten, dass sie für Entwickler immer noch übersichtlich und verständlich bleiben. Bei komplexeren Produkten steigt natürlicherweise die Anzahl der Anforderungen und auch deren damit verbundene Unübersichtlichkeit und Inkonsistenz.

Durch Strukturierung mit Hilfe von Kategorien kann den Konstrukteuren geholfen werden, einen Überblick über die Anforderungen zu erhalten, um diese besser zu handhaben. Zuerst erfolgt die Festlegung der Kategorien, damit die Konstrukteure die Anforderungen optimaler einordnen können. Bei neuen Produkten sind zunächst die grundlegenden Kategorien zu bestimmen, dann können weitere mit Hilfe des Notwendigkeitsprinzips definiert werden. Die Namen der Kategorien sind bei vorhandenen Produkten oft schon vorgegeben. Die gängigen Bezeichnungen können

auch in der Literatur /8/ nachgeschlagen werden, diese sollen kurz und prägnant den Inhalt der Kategorie ausdrücken.

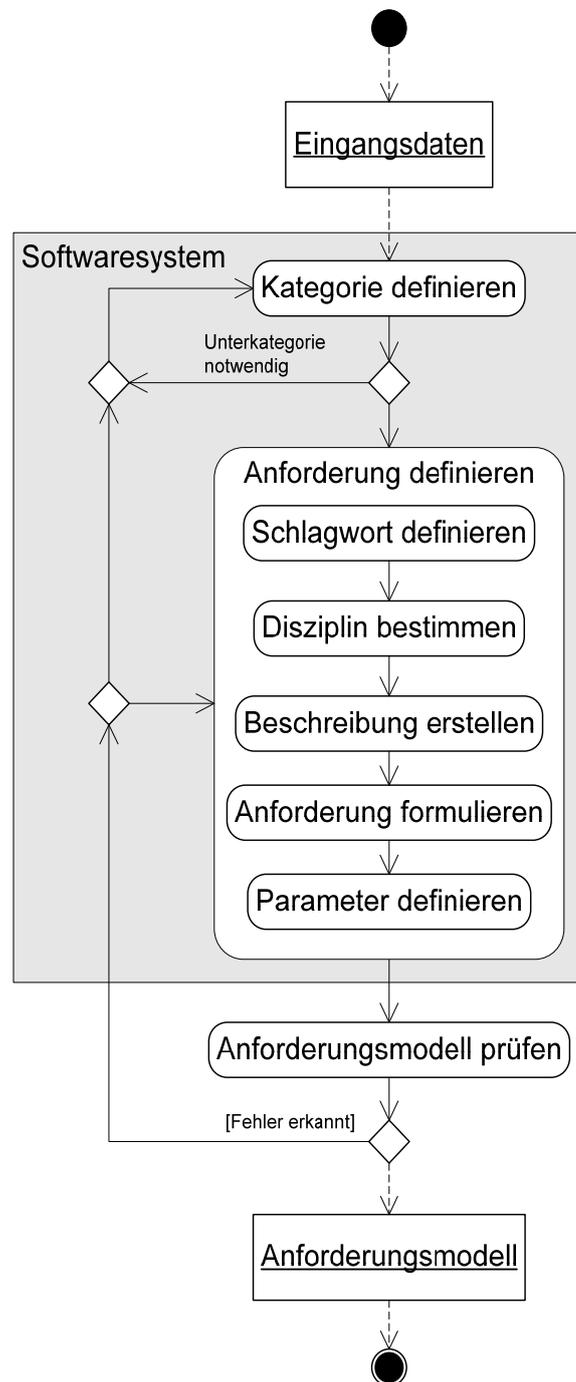


Bild 4-5: Erstellung eines Anforderungsmodells

Anforderung definieren

Ein Auftraggeber hat bestimmte Vorstellungen und Wünsche bezüglich der Eigenschaften seines Produkts. Um diese dem Auftragnehmer erfolgreich zu übermitteln, muss er sie in einer Form ausdrücken, die für den

Auftragnehmer verständlich und möglichst eindeutig ist. Die Formulierung der Anforderungen kann zuerst als Fliesstext erfolgen. Daraus kann ein Satz herausgefiltert werden, der eine bestimmte Produkteigenschaft definieren soll und als eine *formulierte Anforderung* bezeichnet wird.

Durch dieses Vorgehen besteht die Möglichkeit, den Auftrag schrittweise eindeutig zu klären; die Anforderungen bilden dann eine Grundlage für eine erfolgreiche Realisierung der gewünschten Produkteigenschaften.

Da die Formulierung der Anforderung für das einheitliche Verständnis unterschiedlicher Menschen entscheidend ist, sollen diese von Auftraggeber und Auftragnehmer gemeinsam erarbeitet werden.

Die Definition einer Anforderung soll aus verschiedenen Schritten bestehen, die in dem Übergang von unpräzise ausgedrückten Vorstellungen und Wünschen zu eindeutigen Formulierungen eine Hilfestellung bieten.

Schlagwort definieren

Wie bereits erwähnt, sollen die Anforderungen in einem Softwarewerkzeug strukturiert definiert und verwaltet werden. Für die Lesbarkeit und Verständlichkeit der Struktur ist von herausragender Bedeutung, dass ihre Elemente mit einem Blick auf dem Computerbildschirm erfasst und identifiziert werden können. Da eine Anforderung als ein Satz formuliert ist, die oft sehr lang sein kann, ist sie deshalb nicht geeignet, in einer hierarchischen Struktur vollständig abgebildet zu werden. Eine Lösung hierzu bietet ein Schlagwort, das kurz und prägnant den Inhalt einer Anforderung repräsentieren kann. Deren Anwendung zum Aufbau einer hierarchischen Darstellung hilft, die Anforderungsstruktur verständlich und übersichtlich zu gestalten.

Erst wenn eine Absicht, eine Skizze oder ein Fliesstext bezüglich der zu erfüllenden Eigenschaft zur Verfügung stehen, sollen Auftraggeber und Auftragnehmer im Konsens das Schlagwort der Anforderung passend zu der gegebenen Kategorie bestimmen.

Disziplin bestimmen

Da die heutigen Produkte oft mechatronischer Natur sind, kann die Zuordnung einer Anforderung zu einem bestimmten Disziplin wie Mechanik oder Elektronik helfen, die Kompetenzen zu explizieren und die Entwicklung des Produkts übersichtlich zu gestalten. Dies führt zur Steigerung der Effizienz bei den Entwicklungsarbeiten.

Sprachliche Beschreibung einer Anforderung erstellen

In der Anfangsphase haben Vorstellungen und Wünsche des Kunden bezüglich der zu erfüllenden Produkteigenschaften in der Regel noch nicht die Reife erreicht, dass sie eindeutig und präzise formuliert werden können. Die meisten Mitarbeiter der Vertriebs- und Konstruktionsabteilung in einem Unternehmen, die diese Anforderungen zu erfassen haben, sind meist keine ausgebildeten Berater mit linguistischen Fähigkeiten. Eine Beschreibung der Anforderungen, die die Motivation, das Umfeld, die Begründung und die Randbedingungen der gewünschten Produkteigenschaft beinhaltet, wird somit für die Konstrukteure eine wesentliche Hilfe sowie die Grundlage für eine weitere Präzisierung der Anforderung sein.

Anforderung in einer Schablone mit oder ohne Bedingung definieren

Wie bereits erwähnt wurde, sind die meisten Betroffenen, die eine Anforderung definieren sollen, in der Regel nicht in der Lage, gleich eine eindeutige, präzise formulierte und korrekte Anforderung definieren zu können. Der Grund dafür liegt einerseits in der fehlenden Ausbildung, und andererseits in der Schwierigkeit der linguistischen Formulierung, die die Vorstellung des Autors (z. B. Kunde) über die zu erfüllende Eigenschaft vollständig und korrekt wiedergibt. In dieser Situation wird die schablonenbasierte Vorgehensweise eine effektive Unterstützung bieten.

Eine Schablone – integriert in einem Softwarewerkzeug – kann dem Konstrukteur benannte Eingabefelder vorgeben, die mit einzelnen Begriffen ausgefüllt werden sollen. Die Namen dieser Felder sollen den sprachlichen Bestandteilen einer Anforderung entsprechen. Der Konstrukteur oder Kunde kann in der oben genannten Beschreibung und in der Schablone entsprechende Elemente miteinander assoziieren und somit eine Anforderung ableiten. Falls mit der Anforderung Bedingungen verknüpft sind, soll die Schablone entsprechend erweitert werden; das Vorgehen wird sich dadurch aber nicht ändern. Für diesen Ansatz reicht das Wissen über die Grammatik der deutschen Sprache, das in der normalen schulischen Ausbildung über Satzglieder der deutschen Syntax vermittelt wird. Dadurch wird einem Beteiligten geholfen, ohne spezielle Kenntnisse methodisch eine Anforderung präzise, eindeutig und korrekt auf einer sehr granularen Ebene auszudrücken.

Parameter definieren

Mit Parametern ist es möglich, eine in Form des sprachlich formulierten Satzes erstellte Anforderung noch präziser, eindeutiger und kürzer auszudrücken, wenn einer verständlichen Bezeichnung ein Wert und eine Einheit zugewiesen werden. Bei der einfachen Anforderung bezüglich Länge oder Breite eines Produkts ist es oft nicht notwendig, sie sprachlich zu formulieren. Für die Verarbeitung dieser Anforderung reicht bei der Produktentwicklung die parametrische Darstellung voll aus. Parameter können somit als die tiefste und letzte Ebene der Präzisierung einer Anforderung benutzt werden.

4.2.4.3 Erstellung eines Funktionsmodells

Das vorher erstellte Modell der Produkthanforderungen gibt vor allem die gewollten Produkteigenschaften wieder, beinhaltet jedoch viel zu wenige Informationen, die für Produktkonstruktion, -anpassung und -herstellung notwendig sind. Deswegen ist ein weiterer Präzisierungsschritt notwendig, in dem die Umwandlung des Modells abstrakter Produkteigenschaften in ein weniger abstraktes Modell der Produktfunktionen stattfindet. Dadurch wird es möglich, der gesuchten Lösung für das herzustellende Produkt einen Schritt näher zu kommen und mit wenigen Investitionen abzuschätzen, ob sie den gewollten Vorstellungen entspricht. Weiterhin ermöglicht die Funktionshierarchie die systematische Dekomposition der funktionalen Anforderungen in eindeutige Funktionen, denen Funktionsträger zu ihrer Realisierung zugewiesen werden können. Sie kann als lösungsneutral angesehen werden, weil sie im Gegensatz zu einem Funktionsgraph, wie z. B. beim PAHL/BEITZ /6/, keine modellierte Reihenfolgen enthält, die dadurch eine Lösung implizieren kann.

Bei der Erstellung des Funktionsmodells werden Produktfunktionen von Produkthanforderungen abgeleitet, ihre Hierarchie erarbeitet, danach bei Bedarf in Teilfunktionen zerlegt und schließlich den entsprechenden Funktionsträgern zugewiesen.

Die Erstellung des Funktionsmodells soll erst dann erfolgen, wenn das Anforderungsmodell die notwendige Reife erreicht hat, um mindestens die Gesamtfunktion ableiten zu können. Weitere Arbeiten können dann iterativ stattfinden. Dabei werden aus den funktionalen Anforderungen notwendige Produktfunktionen direkt abgeleitet, weiter strukturiert und um die konstruktions- und herstellungsrelevanten Informationen ergänzt. Der Verlauf zur Erstellung eines Funktionsmodells wird im Bild 4-6 dargestellt. Die einzelnen Schritte des Verlaufs werden nachfolgend beschrieben:

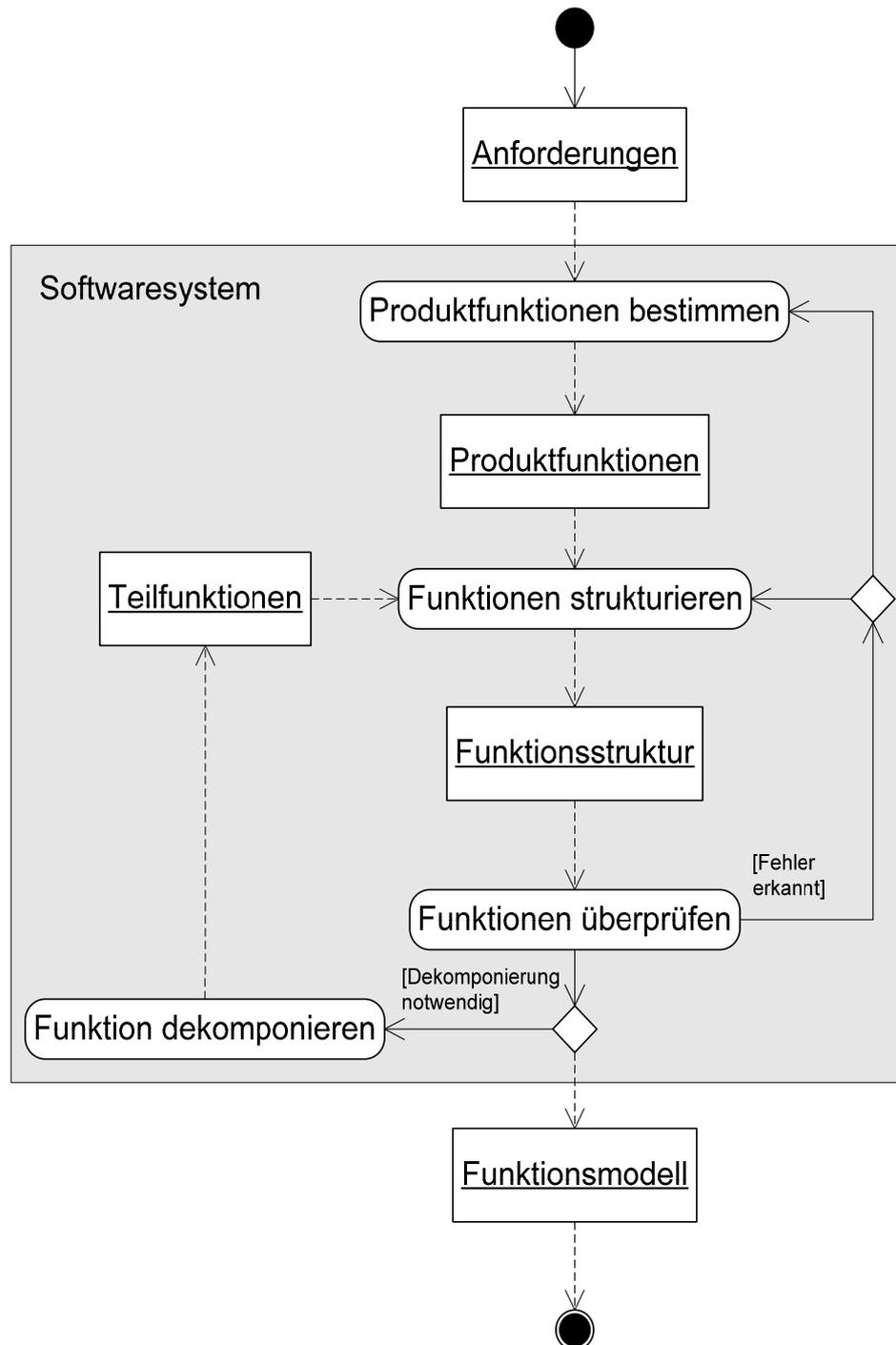


Bild 4-6: Erstellung eines Funktionsmodells

Produktfunktionen ableiten

Bei der Produktentwicklung, wie bereits erwähnt, ist die Lösungssuche die Konkretisierung des Abstrakten. Die gewollte Anwendung eines technischen Objekts, die als eine funktionale Anforderung abstrakt formuliert ist, soll so konkret ausgedrückt werden, dass sie immer noch ergebnisneutral formuliert bleibt. Diese Konkretisierung trägt dazu bei, schon die Lösung

oder Lösungsvarianten im Hinblick auf ihre Aufgabenerfüllung abzuschätzen und danach diese zu akzeptieren oder zu verwerfen. Sie setzt somit dem Konstrukteur die Grenzen zur Gestaltung und Optimierung des Funktionsträgers in dem Lösungsraum.

In dem hier entwickelten Ansatz wird bei der Ableitung einer Produktfunktion aus einer Produkthanforderung eine Definition dieser Funktion erarbeitet. Sie berücksichtigt folgende Elemente:

- Name der Funktion, bestehend aus dem Prozesswort und dem Objekt,
- sprachliche Beschreibung,
- Eingangs- und Ausgangsparameter sowie
- Zuweisung des Funktionsträgers.

Der Verlauf einer Ableitung wird im Bild 4-7 und einer Funktionsdefinition im Bild 4-8 dargestellt und nachfolgend detailliert beschrieben.

Prozesswort bestimmen

Die o. g. gewollte Produkthanwendung kann sprachlich nur durch ein Verb ausgedrückt werden, das einen Prozess, wie „befördern“ oder „wandeln“, beschreibt. So ist es möglich, einer Anwendung, die als eine funktionale Produkthanforderung formuliert wurde, genau ein Prozesswort zuzuordnen. Dadurch wird ihre eindeutige und präzise Konkretisierung gewährleistet, die für die weitere Gestaltung des Produkts notwendig ist. Damit ist ein Prozesswort der unveränderbare Kern einer Funktionsbezeichnung.

Ein Prozesswort kann ausgehend aus der funktionalen Produkthanforderung oder auch aus einer anderen (Produkt-)Funktion als eine Teil- oder Unterfunktion abgeleitet oder bestimmt werden.

Objekt zuweisen

In diesem Zusammenhang beschreibt ein Objekt einen Gegenstand, der sowohl den Eingang als auch den Ausgang einer Funktion darstellt und in einem Vorgang verarbeitet wird. Dadurch ergänzt es das Prozesswort um eine Information, die die gewollte Anwendung präziser formulieren lässt, beispielsweise „Passagiere befördern“. Durch das Objekt wird der Funktionsname vervollständigt.

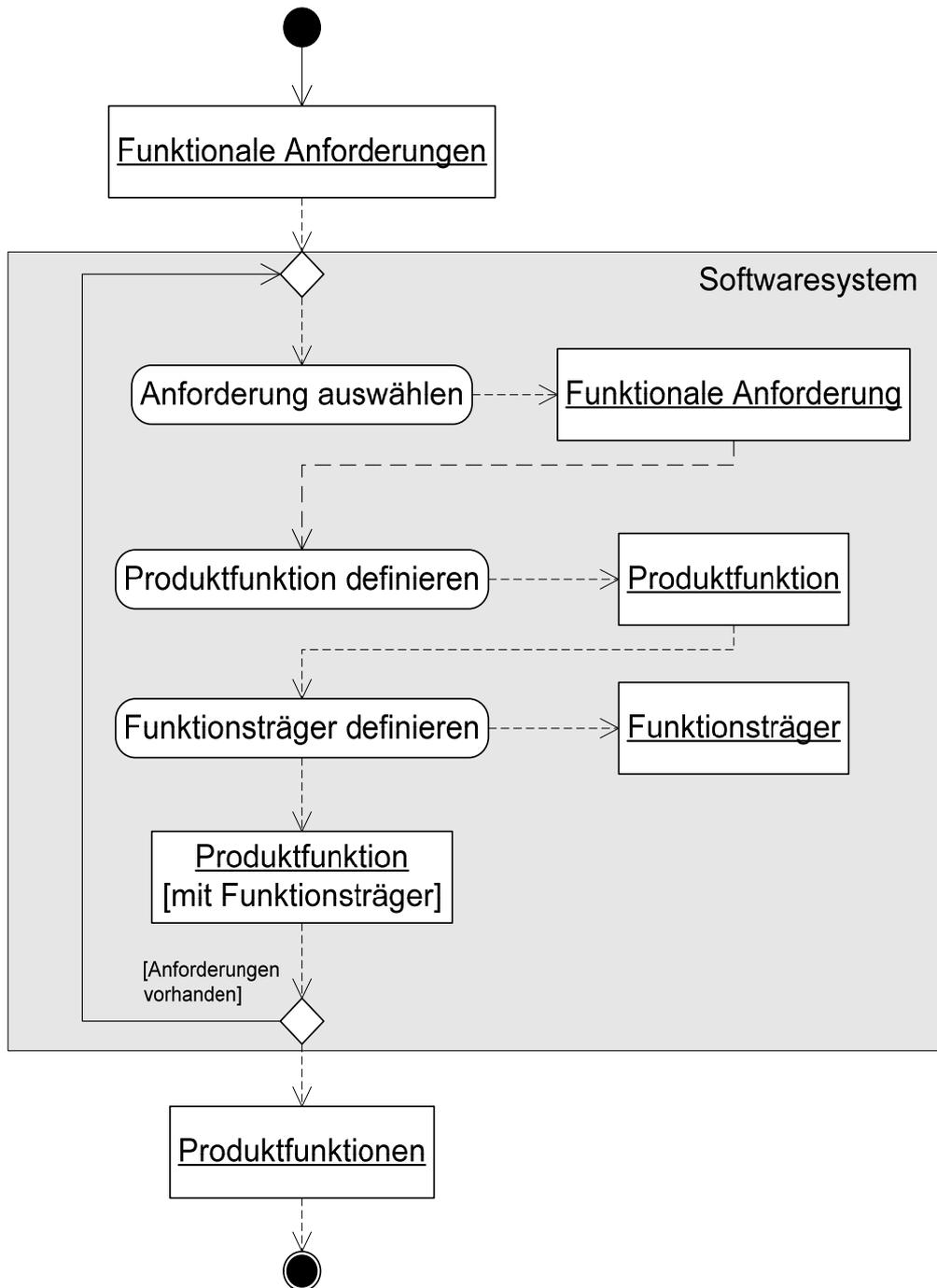


Bild 4-7: Ableitung einer Produktfunktion

Das Objekt kann von der Produkthanforderung übernommen oder abgeleitet werden. Bei der Ableitung besteht die Möglichkeit, es zu konkretisieren oder zu generalisieren. Hierdurch erfolgt die genauere Bestimmung der Grenzen zur Gestaltung und Optimierung des Funktionsträgers. Das Objekt kann durch Name und Parameter, die seine Eigenschaften abbilden, genau definiert werden; dazu kann hilfswise eine Beschreibung erstellt werden.

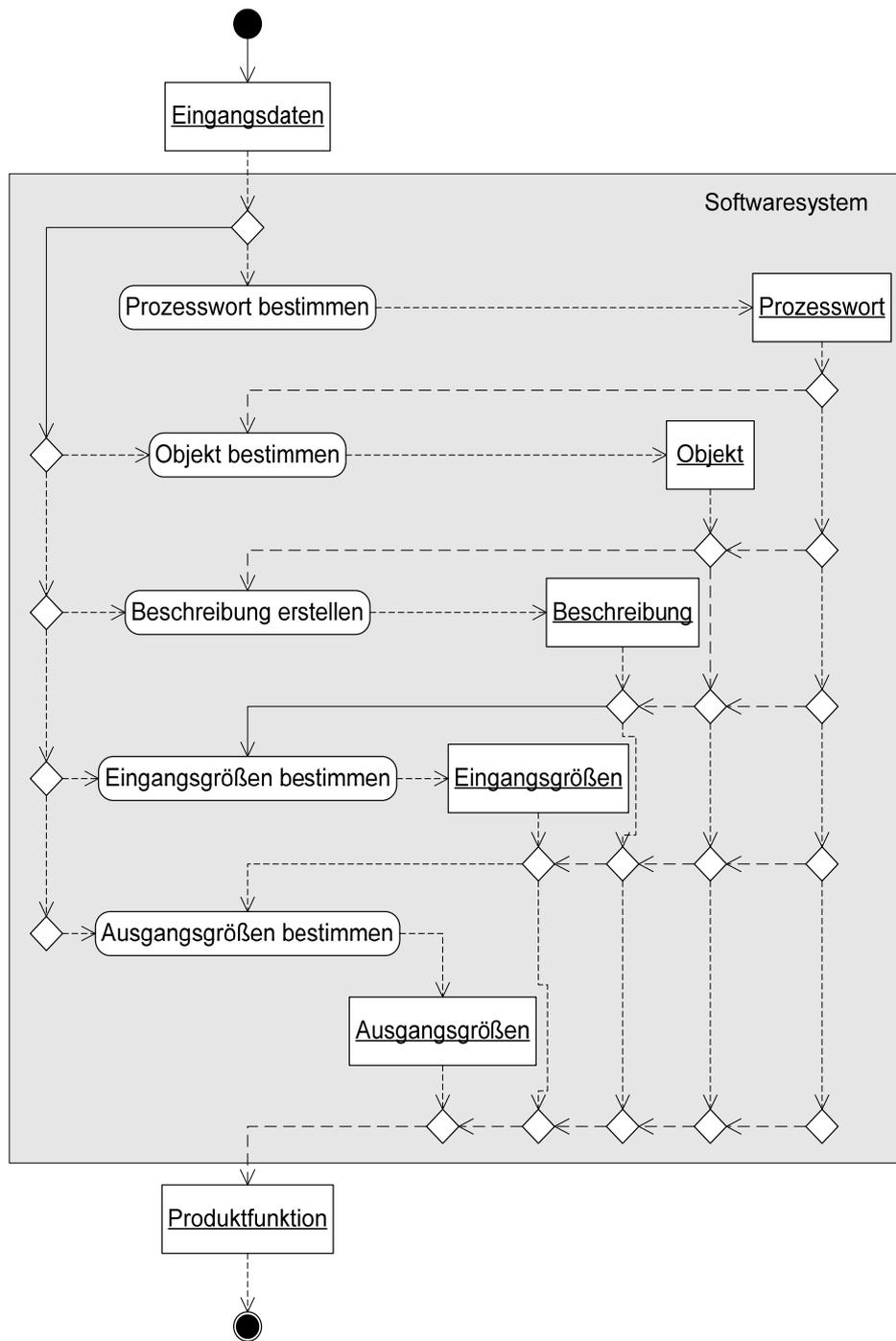


Bild 4-8: Definition einer Produktfunktion

Sprachliche Beschreibung einer Funktion erstellen

Eine Beschreibung, die Motivation, Umfeld, Begründung und Randbedingungen der Funktion beinhaltet, kann eine Hilfe für die Entwicklungsarbeiten der beteiligten Mitarbeiter sein. Sie soll eine Grundlage bilden, Fehler bei der Definition der Funktionsbestandteile zu vermeiden.

Eingangsgrößen bestimmen

Nach der Definition des Funktionsnamens ist die Bestimmung der Voraussetzungen und Startbedingungen (Eingangsgrößen) für die Verwirklichung der gewollten Anwendung des technischen Objekts ein weiterer Schritt bei der Konkretisierung und Präzisierung auf dem Weg zur Lösungsfindung.

Ausgangsgrößen bestimmen

Da die Anwendung eines technischen Objekts gewünschte Ergebnisse liefern soll, deren Werte von den Eingangsgrößen abhängig sind, und sie hinsichtlich der Erfüllung der Anforderungen überprüft werden sollen, müssen die Ausgangsgrößen sowohl hierfür als auch für die Vervollständigung der Funktionsdefinition abgebildet werden.

Die Eingangs- und Ausgangsgrößen können durch Parameter ausgedrückt werden, die durch Merkmale, wie Name, Wert, Einheit charakterisiert werden. Diese können erst dann definiert werden, wenn das Prozesswort feststeht. Anschließend sollen ergänzend dazu Parameter bestimmt werden, die sich auf das Objekt beziehen.

Funktionsträger definieren

Der Funktionsdefinition folgt dann ein nächster Präzisierungsschritt in der Lösungssuche. Hierbei wird ein Funktionsträger definiert, der eine zur weiteren Gestaltung des Produkts notwendige Lösung oder ein Lösungsprinzip darstellt.

Wenn der Funktionsträger in dieser konzeptionellen Phase gestaltneutral definiert und so einer Funktion zugewiesen wird, ergeben sich dadurch mehrere Vorteile. Da zu den Produktfunktionen, die in dem hier vorgeschlagenen Ansatz von den funktionalen Anforderungen abgeleitet und weiter zerlegt werden, wiederum ein Funktionsträger definiert wird, lässt sich zum einen sicherstellen, wodurch funktionale Anforderungen erfüllt werden können. Zum anderen hilft dies in der weiteren Entwicklung, effizienter zu überprüfen, ob alle funktionalen Anforderungen erfüllt worden sind. Durch die Zuweisung der Funktionsträger entsteht parallel zu der Funktionsstruktur eine Produktstruktur. Dadurch kann sichergestellt werden, dass alle Funktionen, die definiert wurden, durch entsprechende Funktionsträger auch erfüllt sind. Somit wird eine bessere Konsistenz der Entwicklung erreicht und Konstruktionsfehler werden nahezu vermieden.

Eine bestimmte Produktfunktionalität lässt sich durch verschiedene geometrische und technologische Gestaltungsmöglichkeiten realisieren. In dem hier vorgeschlagenen Ansatz können bei einer Funktion die Funktionsträger ausgetauscht werden. Dieser Ansatz hilft, die Vielfalt der Möglichkeiten zu begrenzen und die nicht relevanten auszuschließen. Außerdem erleichtert er die Wiederverwendung vorhandener Bauteile und Normteile. Weiterhin können dadurch effizient Lösungsalternativen abgeschätzt werden, um eine optimale Anpassung oder eine Variante zu bestimmen.

Wenn der Funktionsname bereits feststeht, sollen bei der Definition der Funktionsträger der Name, eine Beschreibung und die Parameter sowie die Zuordnung zu einer Disziplin wie Mechanik oder Elektrik bestimmt werden. Dabei kann auch ein zu erstellender Verweis auf eine vorhandene CAD-Datei die Konstruktion wesentlich beschleunigen.

Funktionen strukturieren

Die Produktfunktionen werden implizit durch die funktionalen Produkthanforderungen bestimmt, die wiederum in dem Anforderungsmodell kategorisiert, aber nicht hierarchisiert sind. Für die Lösungssuche und Realisierung der Aufgabe ist jedoch notwendig, eine Funktionsstruktur zu erarbeiten. Sie dient der weiteren Konkretisierung, in der zu den Funktionen in der Struktur die entsprechenden Funktionsträger zugewiesen werden. Dadurch entsteht eine Bauteilstruktur (Produktstruktur), die wiederum als Eingangsinformation zur weiteren Ausarbeitung in der Konstruktionsphase dient. Außerdem führt die Strukturierung dazu, dass durch die Erstellung alternativer Funktionsstrukturen ganze Lösungsfamilien dargestellt werden können, die zu einer schnellen Abschätzung und Variantenüberprüfung dienen.

Mit der Strukturierung kann begonnen werden, sobald mindestens eine funktionale Anforderung definiert wurde, so dass davon eine Produktfunktion abgeleitet werden kann, die in weitere Teilfunktionen zerlegt wird. Diese Vorgehensweise kann als ein iterativer Prozess gesehen werden, in dem Funktionszerlegung, Funktionsdefinition und Funktionsträgerzuweisung abwechselnd stattfinden.

Funktionen überprüfen

Da es möglich ist, dass bei den Modellierungsarbeiten verschiedene Inkonsistenzen und Fehler auftreten können, sollen die Funktionen in einem entsprechenden Vorgang permanent überprüft werden. Dabei sind auch Relevanz und Korrektheit der sprachlichen und inhaltlichen Formulierung zu kontrollieren und zu verbessern.

Funktionen dekomponieren

Da die Produktfunktionen abstrakt formuliert sein und komplexe Vorgänge ausdrücken können, sollen sie in möglichst eindeutige und atomare Vorgänge zerlegt werden. Dies ermöglicht, die funktionalen Produktanforderungen weitgehend vollständig abzubilden. Diese Abbildung dient wiederum dazu, die Lösung besser zu präzisieren.

Wenn die Funktionen atomar ausgedrückt werden, besteht die Möglichkeit, den Funktionsträger sogar auf den Ebenen des Bauteils zu bestimmen, was den Aufwand und die Zeit beim weiteren Konstruieren deutlich verkürzen kann.

Eine Funktion soll erst dann dekomponiert werden, wenn festgestellt wurde, dass sie aus anderen Vorgängen zusammengesetzt ist oder werden kann und die Zerlegung für die weitere Produktentwicklung notwendig ist.

Der Verlauf der Dekomposition einer Funktion wird im Bild 4-9 dargestellt. Die Tätigkeit „*Teilfunktion definieren*“ verläuft analog zu „*Funktion definieren*“ (vgl. Bild 4-8) und wird deswegen im Folgenden nicht explizit beschrieben.

Dekomposition überprüfen

Auf Grund verschiedener Inkonsistenzen, die auch beim Modellieren auftreten können, soll die Dekomposition auf eventuelle Fehler überprüft werden. Dabei sind die Zusammensetzung aus anderen Vorgängen und die weiteren Möglichkeiten im Hinblick auf die spätere Bestimmung des Funktionsträgers zu kontrollieren.

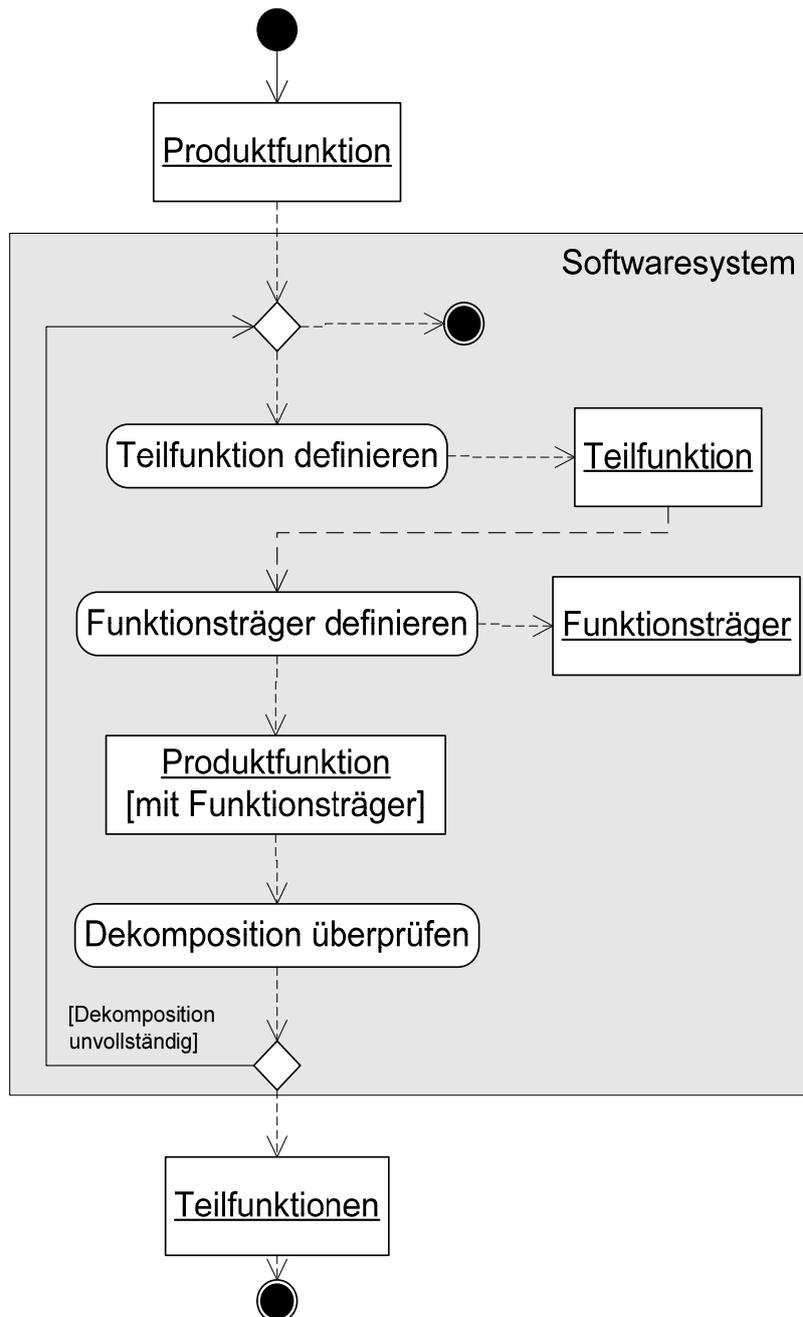


Bild 4-9: Dekomposition einer Produktfunktion

4.2.4.4 Überprüfung des erweiterten Funktionsmodells

Bestimmte Fehler, die in der Detailarbeit auftreten können, sind erst bei einer Gesamtüberprüfung feststellbar (der Schwerpunkt liegt dabei auf dem Aufbau des Modells). Daher ist ein entstandenes Funktionsmodell zu überprüfen, ob alle funktionalen Anforderungen durch Produktfunktionen abgebildet sind und ob es andere Anforderungen gibt, die zum einen die Funktionsdefinition und zum anderen die Wahl der Funktionsträger und Objekte beeinflussen können. Zu überprüfen sind mindestens folgende

Kriterien: Vollständigkeit, Konsistenz, Hierarchisierung, Relevanz, Änderbarkeit, Eindeutigkeit und Verstehbarkeit der Formulierungen.

4.2.4.5 Evaluierung des Modells

Da nach der Erstellung eines Produktmodells noch nicht sicher ist, ob die Wirklichkeit des Produkts korrekt abgebildet wurde, soll das Konzeptmodell anhand bekannter Informationen evaluiert werden. Im Unterschied zu Kapitel 4.2.4.4 ist in diesem Schritt festzustellen, ob die Nutzung des Modells zu gebrauchstauglichen Ergebnissen führt oder auch führen kann. In Bezug auf das gesamte Konzeptmodell sollen Notwendigkeit, Aktualität, Relevanz, Vollständigkeit, Gültigkeit, Korrektheit, Eindeutigkeit, Konsistenz, Realisierbarkeit, Umsetzbarkeit, Verstehbarkeit, Testbarkeit und Weiterverwendbarkeit untersucht werden.

Das Konzeptmodell kann evaluiert werden, wenn es fertig ist, oder – insbesondere bei komplexen Modellen – wenn ein Zwischenstand erreicht wird, der für eine Kontrolle geeignet ist.

4.3 Konzipierung eines neuen Produkts

Vorbereitung der Konzipierung

Auf Grund von Marktanforderungen oder Kundenaufträgen kann ein Unternehmen ein neues Produkt, d. h. ein Produkt, das es noch nicht in ihrem Portfolio hat, entwickeln. Hierfür können aber bereits vorhandene Erfahrungen, Computermodelle, Bauteile und Normteile einbezogen werden.

Bei der Erarbeitung der Neukonzipierung wird vorausgesetzt, dass die Einführung des rechnerintegrierten Ansatzes für bestimmte Produkte in dem betroffenen Unternehmen schon stattgefunden hat, d. h. es existieren bereits verschiedene rechnerinterne Produkttypmodelle.

Die Neukonzipierung bedarf einer Vorbereitung. Zuerst sollen die Auftragsdaten bezüglich der technischen und wirtschaftlichen Machbarkeit analysiert werden. Diese Tätigkeiten wurden bereits bei der Einführung des Einsatzes in Kapitel 4.2.2 und 4.2.3 detailliert beschrieben. Hier werden sie aber nicht für einen Produkttyp oder eine Produktreihe, sondern für ein ganz konkretes Produkt ausgeführt. Weiterhin soll die Ähnlichkeit des neuen Produkts zu denen bei dem Auftragnehmer bereits vorhandenen abgeschätzt werden, um gegebenenfalls relevante Produktdaten im Unternehmen zu identifizieren und aus diesen Daten wieder verwendbare Elemente, wie z. B. Anforderungen, Funktionen und Funktionsträger, zu

extrahieren. Die Vorbereitung dieser Neukonzipierung ist im Bild 4-10 dargestellt.

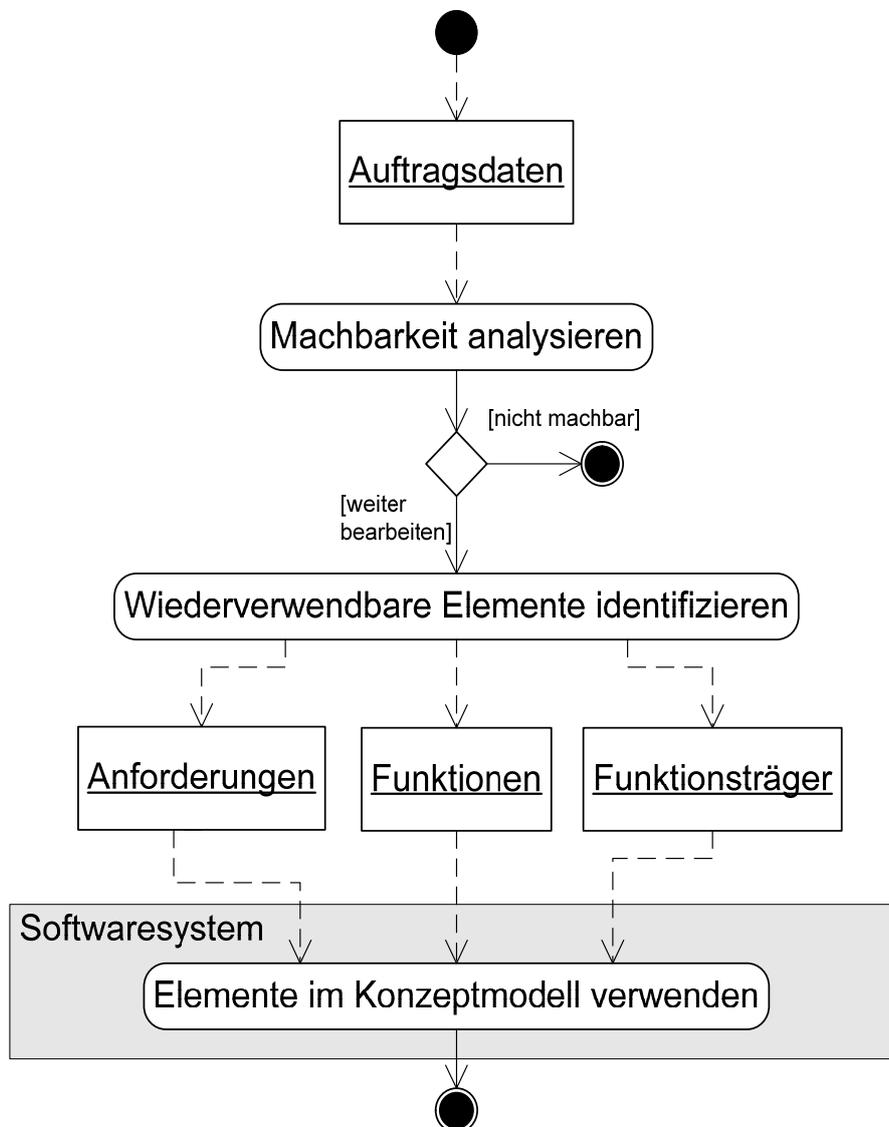


Bild 4-10: Vorbereitung einer Neukonzipierung

Neukonzipierung

In dem hier vorgestellten Ansatz sollen ein Anforderungs- und Funktionsmodell mit Funktionsträgern bei der Entwicklung eines neuen Produkts aus der Sicht des Auftragnehmers erstellt werden. Zum Einsatz kommt ein Softwaresystem, mit dem das rechnerinterne Produktmodell erarbeitet wird. In Kapitel 4.2.4 wurde die Erstellung eines Produktmodells schon beschrieben, die auch für die Konzipierung Anwendung finden kann. Dabei bestehen jedoch bestimmte Unterschiede: Das konzeptionelle Produktmodell kann nicht oder nur teilweise evaluiert werden, da keine oder kaum reale Daten vorhanden sind.

Das Anforderungsmodell und das Funktionsmodell können ähnlich erstellt werden wie es in Kapitel 4.2.4.2 und Kapitel 4.2.4.3 beschrieben wurde. Der Unterschied liegt jedoch darin, dass um die Anforderungs- und Funktionsmodelle nicht für einen Produkttypen, sondern für ein Produkt erarbeitet werden. Dies äußert sich darin, dass die Gültigkeitsbereiche der Parameter nicht explizit berücksichtigt werden müssen. Die Funktionen und Funktionsträger, die wieder verwendet werden können, sollen gegebenenfalls von anderen vorhandenen Produkten entnommen werden. Bild 4-11 stellt ein Konzept der Vorgehensweise dar.

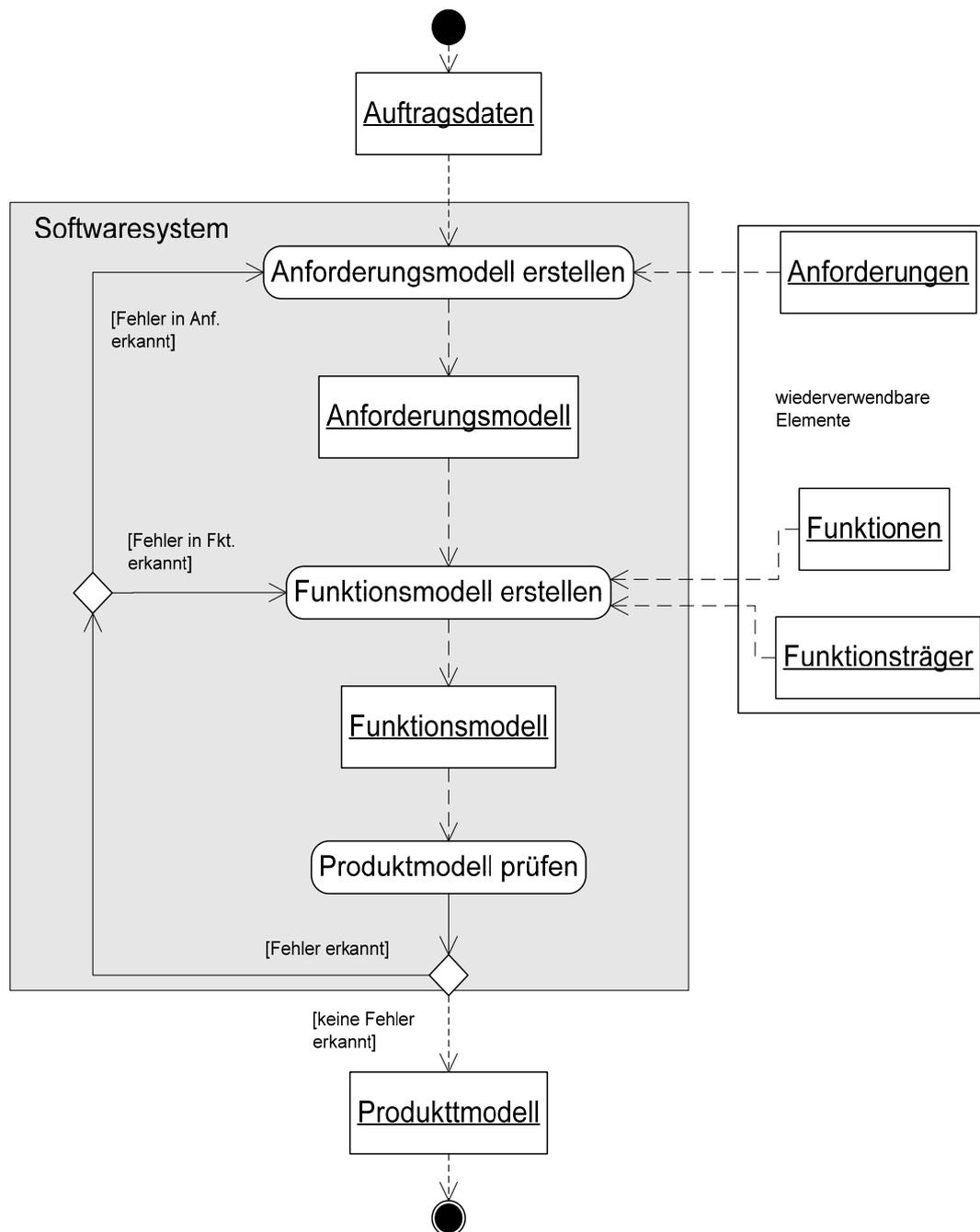


Bild 4-11: Konzipierung eines neuen Produkts

4.4 Konzipierung einer Produktanpassung

Vorbereitung

Aufbauend auf dem Kapitel 4.2, in dem die Erstellung der Infrastruktur für den Einsatz eines rechnerinternen Produktmodells beschrieben wurde, wird in diesem Kapitel dargestellt, wie das Konzeptmodell bei der Realisierung eines konkreten Auftrags angewendet werden kann. Die Anwendung bedarf aber einer Vorbereitung, weil es notwendig ist, aus den vorhandenen rechnerinternen Modellen, das Relevante oder auch die Bestandteile davon für die Modellierungsarbeit auszuwählen.

Zuerst erfolgt die Überprüfung der Auftragsdaten bezüglich der technischen und wirtschaftlichen Machbarkeit, wofür der Ansatz aus den Kapiteln 4.2.2 und 4.2.3 Anwendung finden kann. Die Daten lassen sich aber hierbei nicht so tief greifend analysieren, wie der Ansatz dies beschreibt. Weiterhin soll geprüft werden, ob es wiederverwendbare einzelne Elemente gibt, wie z. B. Anforderungen, Funktionen und Funktionsträger, die in dem Kontext der speziellen Anpassung relevant und wirtschaftlich anwendbar sind. Bild 4-12 stellt ein Konzept der Vorbereitung dar; nachfolgend wird auf die einzelnen Vorgänge näher eingegangen.

Wenn bei der Vorbereitung festgestellt wird, dass die Aufgabe mit der Durchführung von wenigen Veränderungen an einem im Portfolio des Unternehmens existierenden Produkt gelöst werden kann, soll dafür ein vorhandenes Produkttypmodell verwendet werden. Dies kann die Verwendung mehrerer einzelner Modelle überflüssig machen sowie die Transparenz der Daten erhöhen und dabei helfen, die Entwicklungsfehler zu vermeiden. Die Auswahl des Produkttypsmodells dient der möglichst präzisen Identifizierung der maßgeblichen Daten, die für die effiziente Lösungsfindung notwendig sind. Dabei werden die Anforderungen aus dem Auftrag mit denen aus den Modellen verglichen und die relevanten ausgewählt.

Wird erkannt, dass größere Änderungen vorgenommen werden müssen, wie z. B. Funktionsumfang, Ausstattung, Elektro- und Elektronikausstattung, wird nur das passende Anforderungsmodell zur weiteren Bearbeitung ausgewählt. Dieser Schritt erleichtert erfahrungsgemäß die Modellierung des Funktionsumfangs, da es einfacher und schneller ist, eine Funktionsstruktur neu zu modellieren, als eine komplexe, vorhandene zu analysieren und grundlegend zu ändern. Die Auswahl des Anforderungsmodells verläuft ähnlich wie bei dem Produktmodell, jedoch liegt der Schwerpunkt mehr auf dem Aufbau und den Kategorien des Modells.

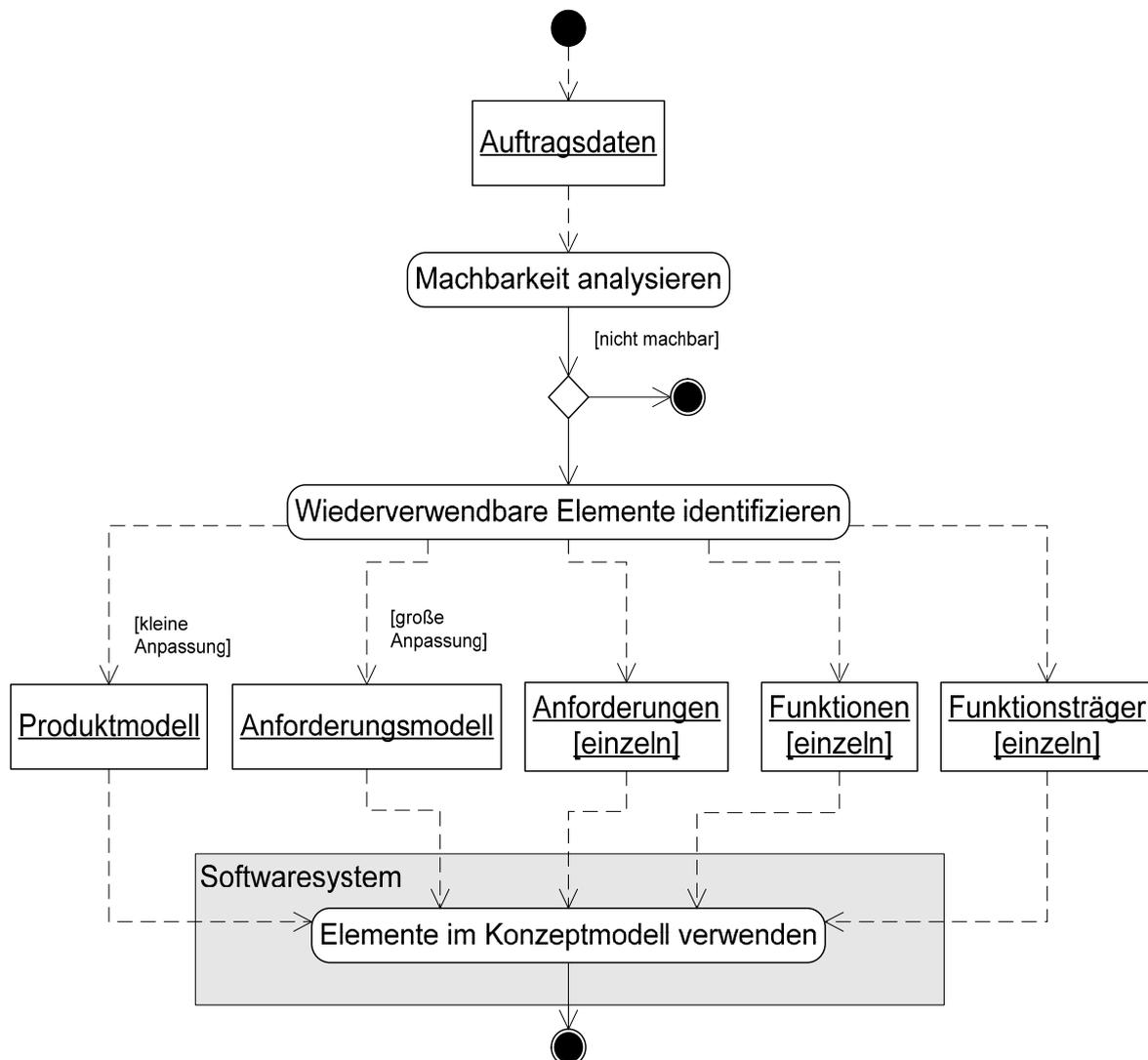


Bild 4-12: Vorbereitung einer Produkthanpassung

Anwendung

Während der rechnerintegrierten Konzipierung einer Produkthanpassung soll zuerst ein bereits erarbeitetes Anforderungsmodell anhand der Auftragsdaten geprüft und bei Bedarf modifiziert werden. Falls das Funktionsmodell nicht vorhanden ist, soll es neu erstellt werden. Wird Änderungsbedarf in dem bereits existierenden Funktionsmodell erkannt, dann soll das Modell überprüft und gegebenenfalls überarbeitet werden. Bei den Änderungen des Anforderungs- und Funktionsmodells sollen nach Möglichkeit wiederverwendbare Elemente aus der vorher aufgebauten Datenbank, wie z. B. einzelne Anforderungen, Funktionen und Funktionsträger, ausgewählt werden. Diese Tätigkeiten sind solange iterativ zu wiederholen, bis der Entwickler die Entscheidung trifft, dass die

Anforderungen durch die Funktionen und Funktionsträger konzeptionell erfüllt sind. Als Ergebnis dieses Schrittes soll ein angepasstes Produktmodell entstehen. Das Konzept der Vorgehensweise wird im Bild 4-13 dargestellt. Die erforderlichen Aktivitäten werden nachfolgend beschrieben.

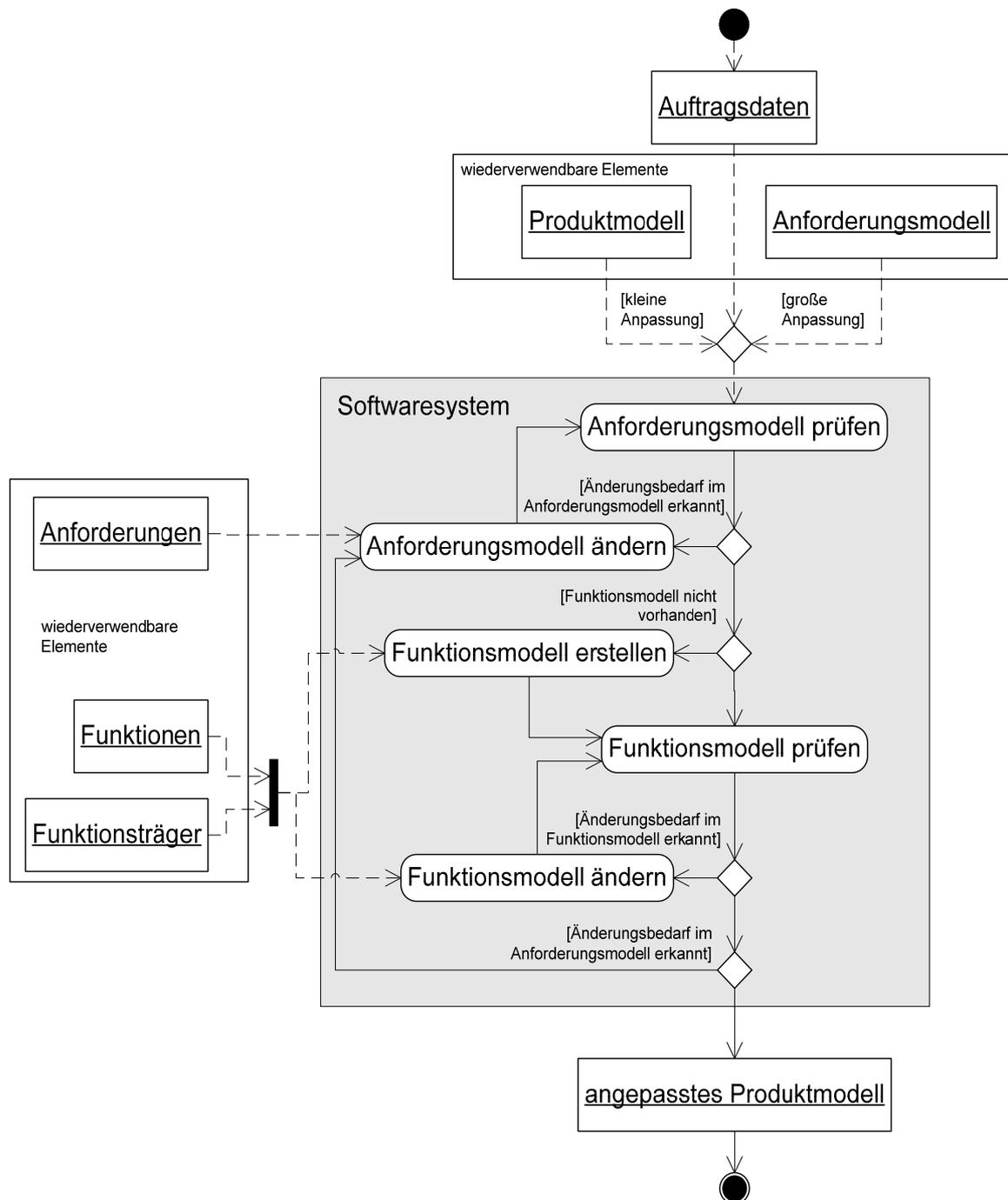


Bild 4-13: Konzipierung einer Produkthanpassung

Anforderungsmodell prüfen und ändern

Da sich die Produkthanforderungen auf Grund eines neuen Kundenauftrages geändert haben könnten, müssen sie zuerst auf Inkonsistenz und Korrektheit geprüft werden. In dem Anforderungsmodell erfolgt zuerst die Untersuchung der Struktur der Kategorien und Anforderungen, wobei die Beschreibungen und Schlagwörter zuvorderst kontrolliert werden. Danach werden die Formulierungen der Anforderungen einer Prüfung unterzogen. Dabei ist sehr wichtig zu wissen, ob eine Beschreibung nur eine konkrete Anforderung beinhaltet, und ob diese in der Anforderungsschablone richtig formuliert ist. Das ist besonders bei funktionalen Anforderungen von großer Bedeutung, bei denen das Verb und das Objekt später als der Name einer Produktfunktion übernommen werden können. Beim Prüfen sollen besonders abstrakte Anforderungen beachtet werden, die durch Subanforderungen präzisiert sein müssen. Danach soll der Ausdruck der Anforderungen mittels Parameter (vgl. Kapitel 4.2.4.2) untersucht werden. Bild 4-14 zeigt das Konzept der Überprüfung und Änderung des Anforderungsmodells.

Funktionsmodell erstellen

Falls bei dem betroffenen Unternehmen für eine konkrete Auftragsentwicklung kein entsprechendes Funktionsmodell vorhanden ist, soll ein neues Modell erstellt werden. Dafür kann die im Kapitel 4.2.4.3 dargestellte Vorgehensweise auch für die Zwecke der Produkthanpassung Anwendung finden. Dabei soll stets die mögliche Wiederverwendung der vorhandenen, realisierbaren Funktionen und Funktionsträger berücksichtigt werden.

Funktionsmodell prüfen und ändern

Wenn ein Anforderungsmodell als korrekt und konsistent eingestuft wird, soll im nächsten Schritt das existierende Funktionsmodell geprüft werden. Die entsprechende Vorgehensweise wurde schon im Kapitel 4.2.4.4 beschrieben und kann hierfür direkt übernommen werden kann.

Wurde bei der Überprüfung ein Änderungsbedarf identifiziert, sind die Bestandteile der Funktionsdefinition zu ändern. Bild 4-15 zeigt das Konzept der Prüfung und Änderung des Funktionsmodells, dessen Aktivitäten im Folgenden erläutert werden. Je nach analysiertem Bedarf kann das Funktionsmodell gründlich geprüft und überdacht, dann aber nicht mehr modifiziert, sondern muss komplett überarbeitet werden.

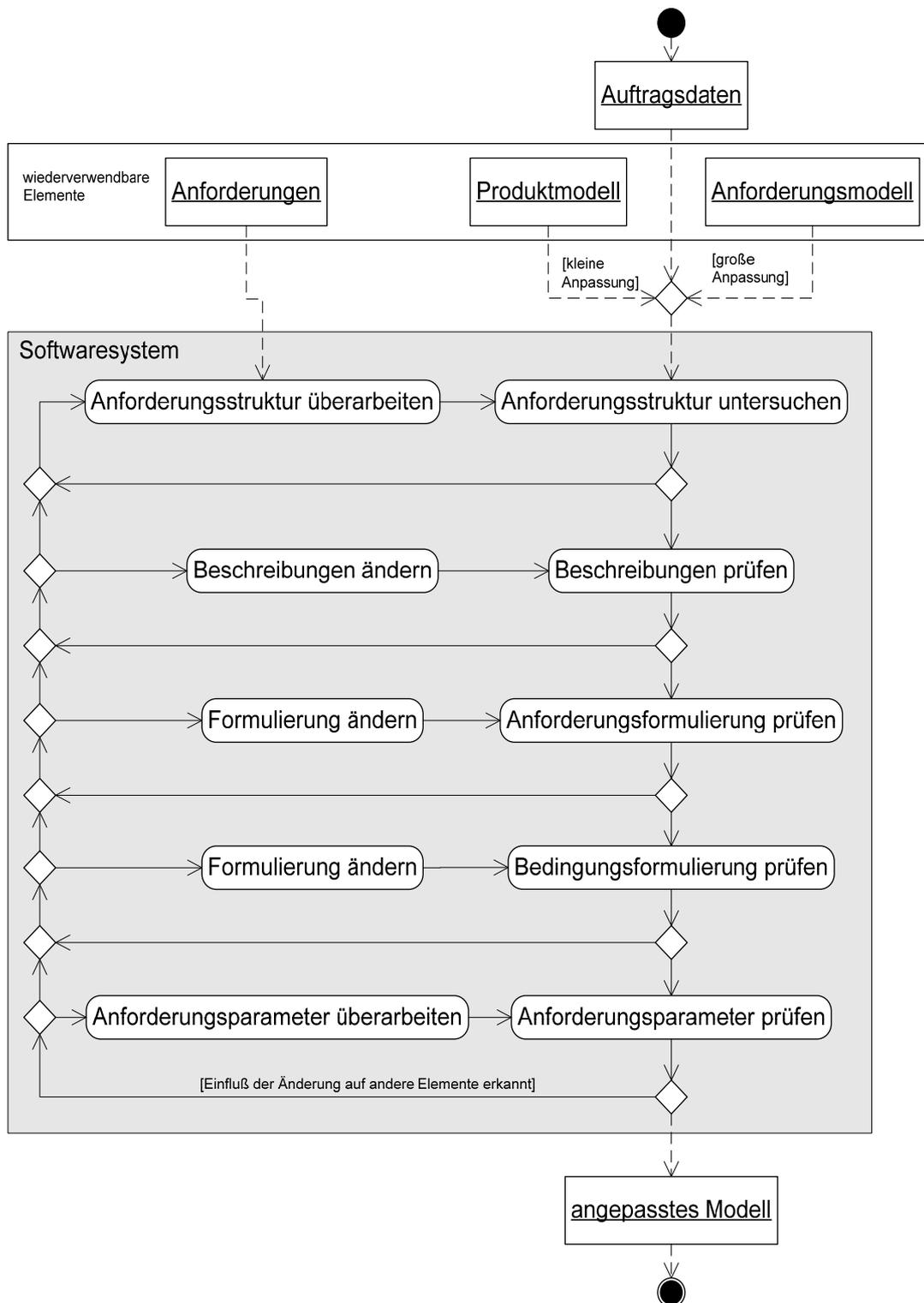


Bild 4-14: Überprüfung und Änderung eines Anforderungsmodells

Funktionsstruktur untersuchen und überarbeiten

Diese Aktivitäten wurden schon in Kapitel 4.2.4.3 erwähnt. Hier unterscheiden sich diese jedoch von denen wesentlich in dem Aspekt,

Objekte prüfen und wechseln

Falls bei einer funktionalen Anforderung das Objekt geändert wird, muss geprüft werden, ob bei der korrespondierenden Produktfunktion ein äquivalentes Objekt vorhanden ist. Gegebenenfalls muss dieses Objekt mit einem, das zu dem neuen Kontext der Anforderung relevant ist, ersetzt werden.

Funktionsparameter prüfen und überarbeiten

Basierend auf den Parametern der geänderten Anforderung werden in diesem Schritt die Eingangs- und Ausgangswerte der Funktion miteinander abgeglichen und nach Relevanz überprüft. Falls dabei Inkonsistenzen festgestellt werden, müssen diese Parameter geändert werden.

Funktionsträger kontrollieren und wechseln

Im letzten Schritt sollen die Funktionsträger auf Funktionserfüllung und Relevanz in dem gegebenen Kontext kontrolliert und die Funktionsträgerparameter vor allem auf Relevanz geprüft werden. Falls notwendig, sollen sie überarbeitet werden, und falls dies für die Funktionserfüllung noch nicht ausreichend ist, soll der Funktionsträger ausgewechselt werden.

4.5 Maßnahmen zur Pflege der rechnerintegrierten Methode

Der globale Markt, die zunehmenden Kundenwünsche, die unterschiedlichen Softwaresysteme sind die erschwerten Umstände, unter denen gegenwärtig ein Unternehmen agiert. Diese Bedingungen ändern sich in einem ständigen Prozess, auf den das Unternehmen oft wenig oder gar keinen Einfluss hat. Um in diesem Wettbewerb erfolgreich zu existieren, muss die betroffene Firma solche Faktoren anpassen und verbessern, die sie beeinflussen kann. Hierzu gehört die eingesetzte rechnerintegrierte Methode mit ihren Elementen, wie das Vorgehen, das Softwaresystem und das rechnerinterne Produktmodell. Diese sollen durch kontinuierliche Pflege immer auf einem aktuellen Stand gehalten werden, der eine effektive und effiziente Produktkonzeption gewährleistet.

Die Umsetzung der in dieser Arbeit entwickelten rechnerintegrierten Methode sowie ihre Anwendung sollen in dem Unternehmen, das sie einsetzt, mit Hilfe von Diagrammen und textuellen Beschreibungen erfasst werden. Es handelt sich vor allem um die Erfassung der Objekt- und der

Prozessebene (vgl. Bild 4-2). Die Dokumentation dient als Basis ihrer Pflege und soll nach durchgeführten Reviews aktualisiert und versioniert werden.

Die Pflege der rechnerintegrierten Methode soll permanent erfolgen und nicht erst dann, wenn Probleme aufgetreten sind. Die o. g. Reviews sollen in bestimmten, für ein konkretes Unternehmen spezifischen, Zeitintervallen, passieren. Sie sind auch als Teil der Qualitätssicherungspolitik eines Unternehmens zu verstehen. In einem Review haben die relevanten Mitarbeiter die Methode zu analysieren, entsprechende Verbesserungsmaßnahmen zu beschließen, die Betroffenen zu benachrichtigen und für die Realisierung der Verbesserungsmaßnahmen einen notwendigen Zeithorizont zu bestimmen.

Ein Konzept der Maßnahmen zur Pflege wird im Bild 4-16 vorgeschlagen. Die einzelnen Tätigkeiten werden nachfolgend beschrieben.

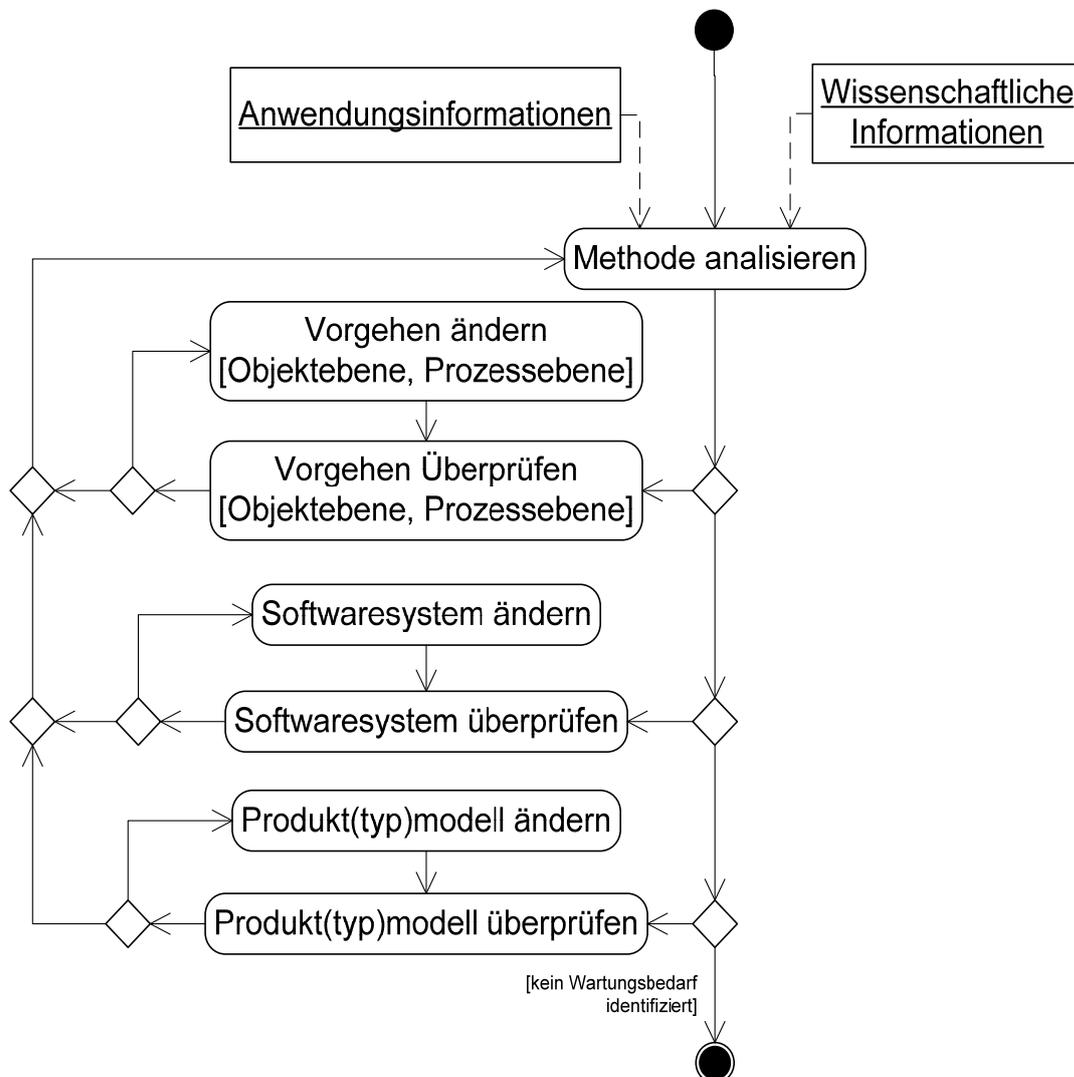


Bild 4-16: Maßnahmen zur Pflege der rechnerintegrierten Methode

Methode analysieren

Basierend auf Informationen aus der operativen Anwendung und der betrieblichen Umgebung – Kunden, Softwareanbieter und betroffene Mitarbeiter – sowie auch unter Berücksichtigung der aktuellen wissenschaftlichen Informationen und vorhandenen Dokumentation der Methode, sollen die dafür bestimmten Mitarbeiter die rechnerintegrierte Methode im Bezug auf ihre Effektivität, Effizienz, Aktualität, Relevanz, Kostenintensität und auf Korrektheit der erzielten Ergebnisse untersuchen. Dies soll passieren, um die erste Entscheidung zu treffen, welches von den drei Elementen der Methode – das Vorgehen, das rechnerinterne Produktmodell oder das Softwaresystem – detaillierter überprüft und ggf. geändert werden muss.

Vorgehen überprüfen und ändern

Falls bei der Analyse festgestellt wird, dass beim Vorgehen Verbesserungsbedarf besteht, soll dem Bedarf entsprechend auf der Objektebene und Anwendungsebene (vgl. auch Kapitel 4, Bild 4-2) das Vorgehen überprüft werden. Auf der operativen Anwendungsebene werden die Prozesse und auf der abstrahierten Objektebene die Tätigkeiten untersucht und geändert. Die entscheidenden Kriterien bilden dabei vor allem die Effektivität und Kostenintensität der Prozesse. Diese Untersuchungen erfolgen bei der Vorbereitung der Anwendung der rechnerintegrierten Methode sowie bei der eigentlichen Konzipierung eines Produkts. Dabei sollen das Vorgehen oder die Prozesse bedarfsgerecht um neue oder fehlende Tätigkeiten erweitert oder ergänzt, oder die bestehenden können in ihren Details aktualisiert werden. Falls aber bestimmte Aktivitäten vorhanden sind, die in der neuen gegebenen Situation aus bestimmten Gründen nicht mehr nutzbar oder anwendbar sind, können sie außer Acht gelassen werden. Die durchgeführten Änderungen sind bildlich sowie textuell zu dokumentieren und zu versionieren.

Die Objektebene und die Anwendungsebene beeinflussen sich gegenseitig und bedürfen der Harmonisierung. Dabei können die operative Anwendungsebene der direkten Wertschöpfung als Grundlage für Abstraktion und die erfasste abstrakte Objektebene als Kontrollelement für die methodische Richtigkeit der Entwicklung auf Prozessebene dienen. Ein Konzept der Maßnahmen ist im Bild 4-17 dargestellt.

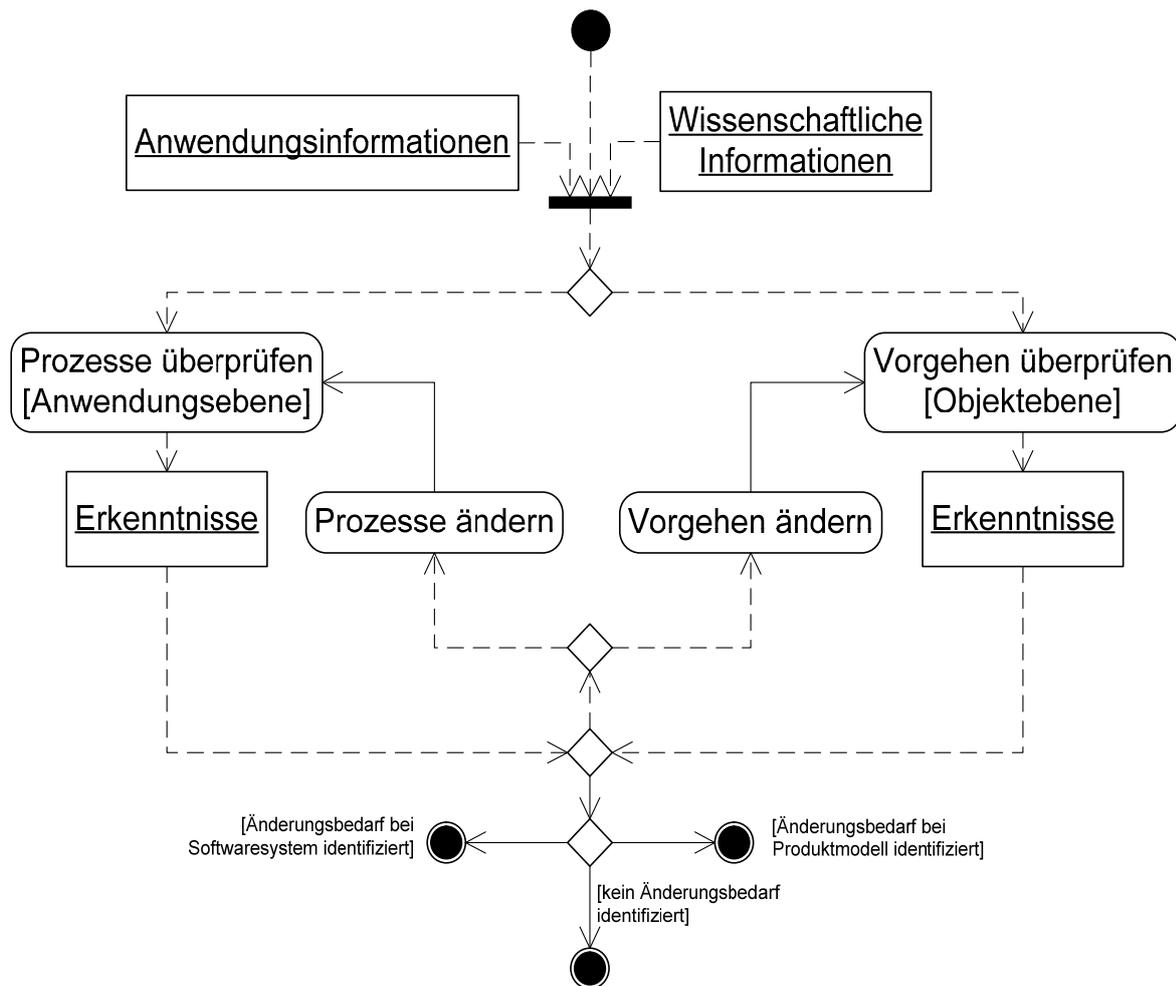


Bild 4-17: Überprüfung und Änderung der Vorgehensweise

Softwaresystem überprüfen und ändern

Wird bei der Analyse bzw. den Änderungen des Vorgehens oder des Produktmodells festgestellt, dass es bei dem Softwarewerkzeug Verbesserungspotenzial gibt, dann soll überprüft werden, welches von den Elementen, wie z. B. die Benutzungsoberfläche, die Bedienung, die vorhandenen Funktionalitäten, die Systemarchitektur, das Fachdatenmodell oder das Systemklassenmodell, zu ändern sind. Anschließend soll ein Konzept der Änderungen zusammen mit dem Softwareanbieter – falls das Softwarewerkzeug gekauft wurde – zunächst ausgearbeitet und dann umgesetzt werden.

Produkt(typ)modell überprüfen und ändern

Liefert die Analyse das Ergebnis, dass bei dem Produktmodell Verbesserungen notwendig sind, dann soll detailliert überprüft werden

(Bild 4-18), ob die Anforderungen, die Funktionen oder die Funktionsträger geändert werden müssen.

Hierbei handelt sich um Änderungen auf zwei unterschiedlichen Ebenen, d. h. sowohl inhaltliche, produktabhängige Änderungen der Produktdaten, wie z. B. neue Geometrieparameter, Materialparameter und Anforderungskategorien, als auch Änderungen an dem Modell selbst, wie z. B. zusätzliches Datenfeld für die Begründung der Anforderung oder Datenfeld für die Vergabe der Versionsnummer bei einem Funktionsträger. Die zweite Ebene löst die Notwendigkeit der Änderungen an dem Softwaresystem auf.

Bei den notwendigen Änderungen handelt es sich entweder um die Erweiterung des Produktmodells, wenn zusätzliche Informationen vorliegen und Daten vorhanden sind, die berücksichtigt werden müssen. Oder es geht um eine strategische Erweiterung des Modells an sich selbst um ein weiteres Teilmodell, wie z. B. ein Strukturmodell der Baugruppen und Bauteile, oder auch disziplinabhängige Teilmodelle der Anforderungen, Funktionen und Bauteilen. Falls bei der Analyse festgestellt wird, dass relevante Informationen in einem Teil des Produktmodells bis jetzt nicht berücksichtigt wurden oder fehlen, soll dieses Modell um solche Informationen ergänzt werden.

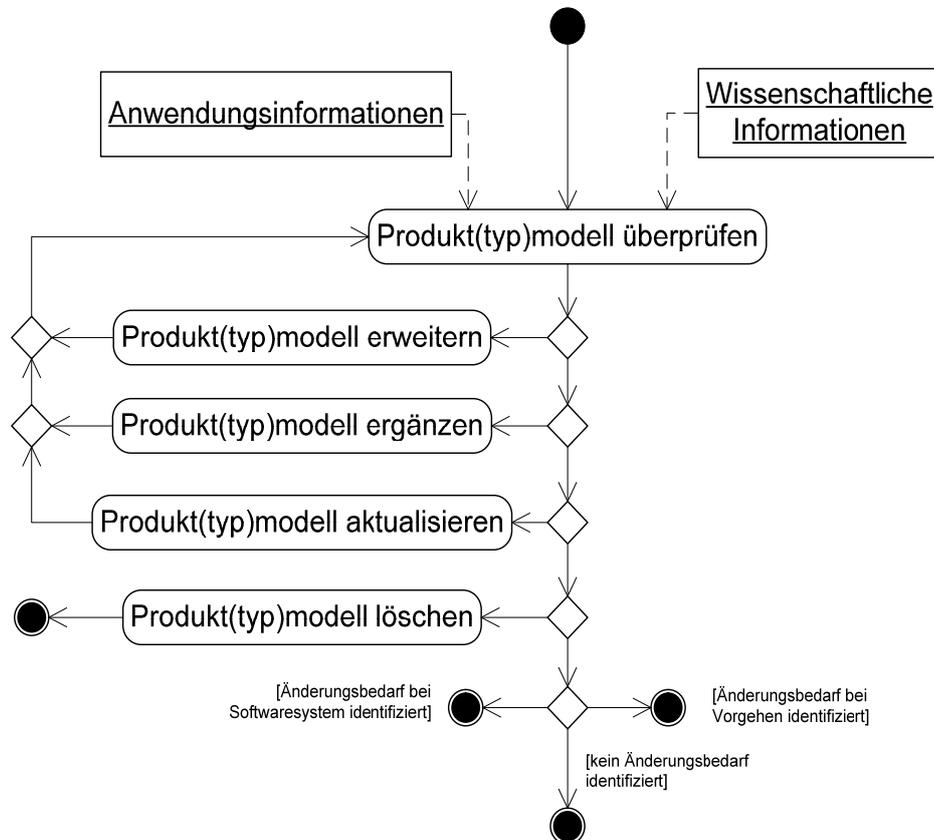


Bild 4-18: Überprüfung und Änderung eines Produkt(typs)modells

Kommen neuere Informationen aus der operativen betrieblichen Anwendung und von den Kunden hinzu, oder stehen aktuelle relevante wissenschaftliche Informationen zur Verfügung, soll das Produktmodell auf beiden o. g. Ebenen aktualisiert werden.

Falls aber bei der Überprüfung festgestellt wird, dass das rechnerinterne Produktmodell nicht mehr anwendbar oder nutzbar ist, kann es archiviert bzw. gelöscht werden. Das Konzept dieses Vorgehen ist im Bild 4-18 dargestellt.

5 Entwicklung eines Systemkonzepts

5.1 Ableitung der Systemfunktionalitäten

Ziel dieses Kapitels ist, eine generische Systemarchitektur des Softwarewerkzeugs zu entwickeln, die eine große Realisierungsfreiheit bei der Implementierung dieses Tools erlaubt.

Bei der Entwicklung eines Softwaresystems sind zuerst die Anforderungen zu definieren (vgl. auch Kapitel 3) /50/. Die Anwendungsfälle hierzu wurden schon in vorherigem Kapitel 4 beschrieben. Basierend auf diesen Vorarbeiten werden jetzt die wichtigsten und notwendigen Systemfunktionalitäten abhängig von den Anwendungsbereichen und Anwendungsfällen bestimmt.

In dem ersten Anwendungsbereich, das im Kapitel 4.2 „*Bestimmung der Maßnahmen zur Einführung des Einsatzes*“ beschrieben wurde, wird das System zur Realisierung des im Kapitel 4.2.4 definierten Anwendungsfalls „*Erstellung eines rechnerinternen Produktmodells*“ benutzt. Dies unterteilt sich in weitere Anwendungsfälle, wie „*Erstellung des Anforderungsmodells*“ und „*Erstellung des Funktionsmodells*“, die als allgemeine Funktionalitäten des Systems verstanden werden sollen. Diese Funktionalitäten können weiter zerlegt werden, was nachfolgend erörtert wird. Um mit Hilfe dieses Softwarewerkzeugs ein Anforderungsmodell erstellen zu können, muss es die im Kapitel 4.2.4.2 definierten Aktivitäten als Funktionen abbilden. Mit diesem System besteht auch die Möglichkeit, ein Funktionsmodell zu erstellen. Dazu muss es die im Kapitel 4.2.4.3 definierten Aktivitäten als Funktionalitäten abbilden. Diese können als Funktionen des zu entwickelnden Softwarewerkzeugs übernommen werden.

In dem zweiten Anwendungsbereich, das im Kapitel 4.3 „*Konzipierung eines neuen Produkts*“ beschrieben wurde, benötigt das System die gleichen Funktionalitäten, die schon oben definiert wurden.

In dem dritten Anwendungsbereich, das im Kapitel 4.4 „*Konzipierung einer Produktanpassung*“ beschrieben wurde, muss das System Anwendungsfälle wie „*Anforderungsmodell modifizieren*“, „*Funktionsmodell erstellen*“ und „*Funktionsmodell überarbeiten*“, realisieren. Im Kapitel 4.4 wurden Anwendungsfälle „*Produkt(typ)modell ändern*“ und „*Softwaresystem ändern*“ definiert, die das System umsetzen muss. Die Namen der Anwendungsfälle können direkt als Funktionalitäten des zu entwickelnden Systems übernommen werden.

Weiterhin gibt es allgemeine Funktionalitäten, die mit der Bedienung und Nutzung des Systems verbunden sind, wie z. B. *Modell speichern*, *Model laden*, *kopieren* oder *ausschneiden*.

5.2 Erarbeitung der Systemarchitektur

5.2.1 Einleitung

In diesem Konkretisierungsschritt wird für die oben beschriebenen Funktionalitäten des Softwarewerkzeugs das Konzept einer generischen Systemarchitektur erarbeitet. Die Modellierungssprache UML bietet hierfür mehrere Möglichkeiten /50/. Die vollständige Ausarbeitung der Modelle ist für die Darstellung des Konzepts nicht zwingend notwendig. Deshalb wurden zu diesem Zweck nur die Fachdaten-, Schichten-, Komponenten- und Strukturklassenmodelle für ausreichend gehalten.

5.2.2 Fachdatenmodell

Um die systemrelevanten Begriffe aus dem Bereich der rechnerintegrierten Konzipierung abzubilden, werden in diesem Datenmodell, basierend auf der Vorarbeiten, fachlich motivierte Klassen definiert. Sie werden mit Hilfe einer Klasse „*Projekt*“ verwaltet. Ein Fachdatenmodell des Softwaresystems wird im Bild 5-1 vorgeschlagen. Die wichtigsten Klassen sind im Folgenden genauer beschrieben, auf die anderen wird nur kurz eingegangen.

Produktanforderung

Entsprechend den Anforderungen aus dem Kapitel 3.1 „*Motivation und der Zielsetzung*“, muss das zu entwickelnde Programm eine eindeutige, präzise und konsistente Definition einer Anforderung methodisch unterstützen.

Eine Anforderung ist – linguistisch betrachtet – ein Modalsatz, der in der deutschen Sprache aus Satzgliedern, wie Subjekt, Prädikat, Objekt, Adverbialbestimmung und Attribut bestehen kann. Diese kann dann als eine Klasse „*Anforderung*“ abgebildet werden, die über Datenfelder verfügt. Die Datenfelder entsprechen den Satzgliedern – versehen mit Attributen. Wenn ein Satzglied nur einmal in dem Hauptsatz der Anforderung vorkommt, kann davon ausgegangen werden, dass sie mit großer Wahrscheinlichkeit auch konsistent und eindeutig formuliert ist. Wenn dies nicht der Fall ist, dann liegen die Fehler in der semantischen Auswahl der Wörter. Weitere notwendige Datenfelder sind *Stichwort*, *Beschreibung*, *Disziplin* und *Art* der Anforderung.

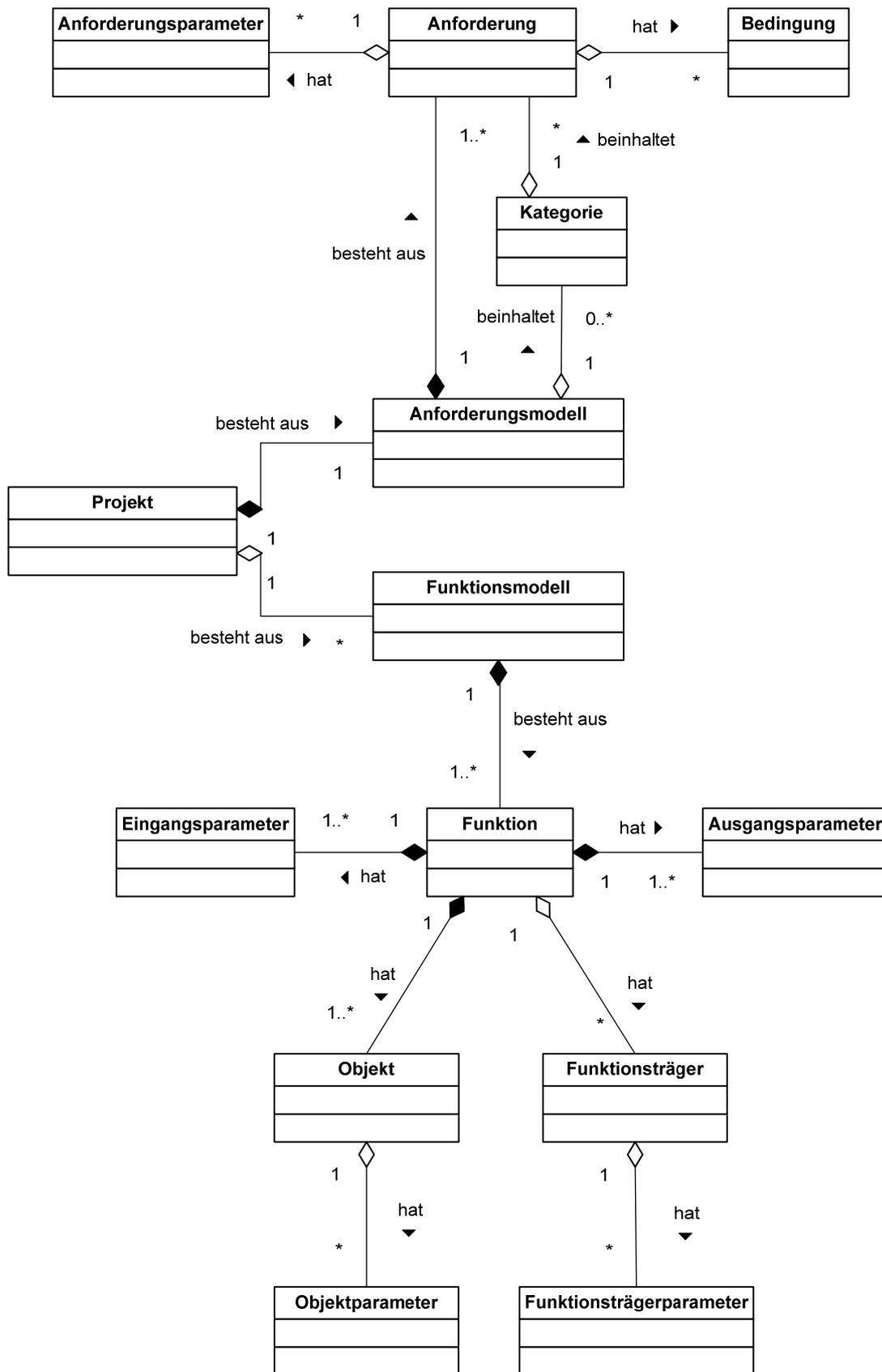


Bild 5-1: Fachdatenmodell des Softwaresystems

Diese Klasse soll dem Anwender auch Methoden für die Erzeugung und Verwaltung von Datenfeldern und für die Behandlung von Parametern und Bedingungen zur Verfügung stellen.

Dieser Ansatz ist zwar am Beispiel der deutschen Sprache dargestellt, aber er kann leicht erweitert und auf andere Sprachen übertragen werden. Weil die Datenfelder bei dem Programmstart erzeugt und mit Hilfe eines so genannten Behälters (z. B. ein Vektor oder eine Liste) verwaltet werden sollen, lassen sich deren Anzahl, Name und Reihenfolge schnell an die jeweilige Sprache anpassen.

Dieser Ansatz zerlegt eine Anforderung in „atomare“ Bestandteile, was die eindeutige und konsistente Definition der Anforderung unterstützt. Darauf basierend wird eine syntaktische Unterstützung der Formulierung mittels graphischer Benutzungsoberfläche erarbeitet. Diese ist für jeden Benutzer leicht zu erlernen, weil sie einem normalen Satzbau entspricht.

Anforderungsbedingung

Da die Erfüllung einer Anforderung mit verschiedenen Bedingungen verbunden sein kann, sollen diese im Fachdatenmodell auch als eine Klasse abgebildet werden. Jede Bedingung kann in Form eines Satzes ausgedrückt werden. Deshalb soll die Klasse „*Bedingung*“ die gleichen Datenfelder beinhalten wie die bereits beschriebene Klasse „*Produktanforderung*“. Eine Ausnahme bildet hier das Datenfeld „*Verbindlichkeit*“ (z. B. „muss“ oder „kann“), die bei der Klasse „*Bedingung*“ nicht mehr vorkommt, zusätzlich wird aber das Datenfeld „*Konjunktiv*“ (z. B. „wenn“) hinzugefügt. Diese Klasse wird durch die Klasse „*Produktanforderung*“ benutzt und soll die Methoden für die Handhabung ihrer Datenfelder beinhalten.

Produktfunktion

Die Klasse bildet eine gewünschte Produktfunktion ab (vgl. Kap. 4.2.4.3), und ermöglicht im weiteren Verlauf die Erstellung der Benutzungsoberfläche für die Unterstützung der Funktionsdefinition.

Eine Produktfunktion wird mittels eines Objekts und Prozessworts, wie z. B. „*Wasser pumpen*“ ausgedrückt. Deshalb sollte diese Klasse entsprechende Datenfelder für ihre Handhabung beinhalten. Da aber ein Objekt spezialisiert oder generalisiert werden kann, wie z. B. Flüssigkeit → Wasser, Benzin → Treibstoff, ist es vorteilhaft, für die Modellierung eine Austauschmöglichkeit zu schaffen (vgl. Kap. 4), so dass die Abbildung des Objekts durch eine weitere Fachklasse notwendig ist. Die Eingangs- und

Ausgangsgrößen werden auch als Klassen abgebildet, weil mehrere Daten für ihre Beschreibung erforderlich sind.

Für die Modellierung ist vorteilhaft, der Funktion so früh wie möglich einen Funktionsträger zuzuweisen (vgl. Kap. 4). Daher soll die Klasse „*Produktfunktion*“ über einen entsprechenden Behälter und notwendige Methoden verfügen.

Diese Klasse soll die Klassen „*Objekt*“, „*Funktionsträger*“, „*Eingangsparameter*“ und „*Ausgangsparameter*“ nutzen, verwalten und dafür die notwendigen Methoden dem Anwender zur Verfügung stellen.

Objekt

Das Objekt, das zusammen mit dem Prozesswort den Namen der Funktion ausmacht, wird mit Hilfe dieser Klasse abgebildet, und ermöglicht deren bessere Beschreibung und Modellierung. Sie soll für die Benennung, Beschreibung sowie Handhabung ihrer Eigenschaften über notwendige Methoden und Datenfelder verfügen.

Funktionsträger

Da es für die Modellierung nicht ausreichend ist, den Funktionsträger mit einem Datenfeld zu beschreiben, muss dafür eine Klasse mit entsprechenden Datenfeldern definiert werden. Sie soll Datenfelder für die Abbildung des Namens (z. B. Welle) oder Disziplin (z. B. Mechanik) haben sowie über einen Behälter für die Bauteilparameter (z. B. Länge oder Material) und Methoden für ihre Handhabung dieser Parameter verfügen.

Anforderungsmodell und Funktionsmodell

Diese Klassen dienen der Verwaltung der Produkthanforderungen und -funktionen. Sie beinhalten dafür entsprechende Methoden und Datenfelder und stellen die höchste Abstraktionsstufe der Modellierung mit Hilfe dieses Softwarewerkzeugs dar. Sie nutzen die Klassen „*Produkthanforderung*“, „*Produktfunktion*“ und „*Anforderungskategorie*“.

Anforderungskategorie

Verschiedene Anforderungskategorien, die bei der Strukturierung der Anforderungen benutzt werden können, sind mittels der Klasse abzubilden. Sie beinhaltet Datenfelder, wie *Stichwort*, *Beschreibung* sowie einen Behälter, in dem sich die Klassen „*Produkthanforderung*“ befinden.

Klassen der Anforderungs-, Objekt-, Funktionsträger-, Eingangs- und Ausgangsparameter

Die Parameterklassen dienen der Verwaltung der parametrischen Definition der Daten. Sie beinhalten die gleichen Felder wie „Name“, „Wert“, „Einheit“, „Beschreibung“ und „Typ“. Sie detaillieren die Beschreibung für Objekt, Funktionsträger und Anforderung. Die Klasse „Anforderungsparameter“ kann außerdem auch als die höchste Präzisierung der Anforderung angesehen werden. Alle diese Klassen müssen dem Anwender Methoden für die Handhabung ihrer Daten zur Verfügung stellen.

5.2.3 Schichtenmodell

Um ein grundsätzliches Konzept des Softwarewerkzeugs darzustellen, wird ein Schichtenmodell ausgewählt, das auf dem Fachdatenmodell und auf den Richtlinien für die Softwarearchitekturentwicklung, dem „*Model, View, Controller*“-Prinzip /25/, basiert. Das Modell beschreibt, aus welchen Hardware- oder/und Softwareschichten die auf dieser Architektur basierenden Lösungen bestehen /50/. Ein Konzept eines Schichtenmodells für ein Softwaresystem zur Produktkonzeption wird im Bild 5-2 dargestellt. Auf die einzelnen Schichten wird nachfolgend kurz eingegangen.

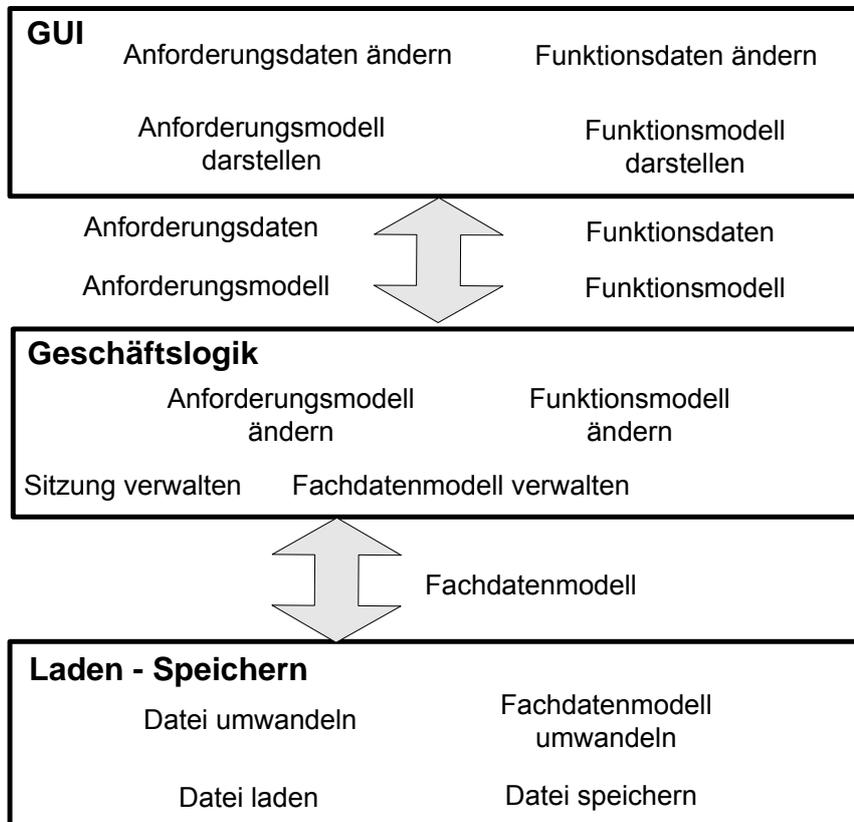


Bild 5-2: Schichtenmodell des Softwaresystems

Das zu entwickelnde Softwarewerkzeug muss über eine Benutzungsoberfläche (*GUI – Graphical User Interface*) verfügen. Mit deren Hilfe soll der Anwender die Anforderungs- und Funktionsmodelle aufbauen können. Die Oberfläche ermöglicht auch, die einzelnen Daten der Anforderungs- und Funktionsmodelle zu definieren, zu erfassen, zu modifizieren und zu verwalten. Sie beinhaltet auch Eingabe-, Ausgabe- sowie Darstellungskomponenten. Die GUI-Schicht soll mit der *Geschäftslogik-Schicht* über diese Anforderungs- und Funktionsmodelle kommunizieren.

Die Geschäftslogik hat die Arbeitssitzung sowie das Fachdatenmodell zu verwalten. Außerdem soll sie die internen Modellierungs- und Aktualisierungsabläufe koordinieren. Sie übergibt und bekommt von der *Laden-Speichern-Schicht* das Fachdatenmodell.

Diese Schicht soll die Umwandlung des Fachdatenmodells in eine speicherungsfähige Datenstruktur ermöglichen und den Speichervorgang durchführen. Sie soll also eine geladene Datei, die Arbeitsdaten beinhaltet, in ein Fachdatenmodell umwandeln, welches zur weiteren Verarbeitung an die Geschäftslogik-Schicht übergeben wird.

5.2.4 Komponentenmodell

Aufbauend auf den Schichten- und Fachdatenmodellen wird im Verlauf der weiteren Konkretisierung ein Konzept der notwendigen Komponenten vorgeschlagen sowie deren Anordnung und Abhängigkeiten identifiziert und benannt.

Das zu entwickelnde System soll aus einem Hauptprogramm – *Application* – bestehen, das von einem *Applicationmanager* verwaltet wird. Da aber das Softwaresystem erweiterungsfähig sein muss, wird der Manager einen „Behälter“ haben, der Komponenten beinhaltet, die dem Benutzer bereichsbezogene Funktionalitäten bereitstellen. Eine der möglichen Komponenten ist der zu entwickelnde *Konzeptmodellierer*, der wiederum über einen eigenen *Komponentenmanager* verfügt, welcher graphische Fachkomponenten, wie *GUI-Anforderungseditor* und *GUI-Funktionseditor* sowie den *Fachdaten-Import-Export* verwaltet. Die Benutzungsoberflächen der Editoren sollen visuelle Eingabe- und Darstellungskomponenten beinhalten, die für die Modellierungsarbeit notwendig sind.

Das Konzept des Komponentenmodells ist im Bild 5-3 dargestellt.

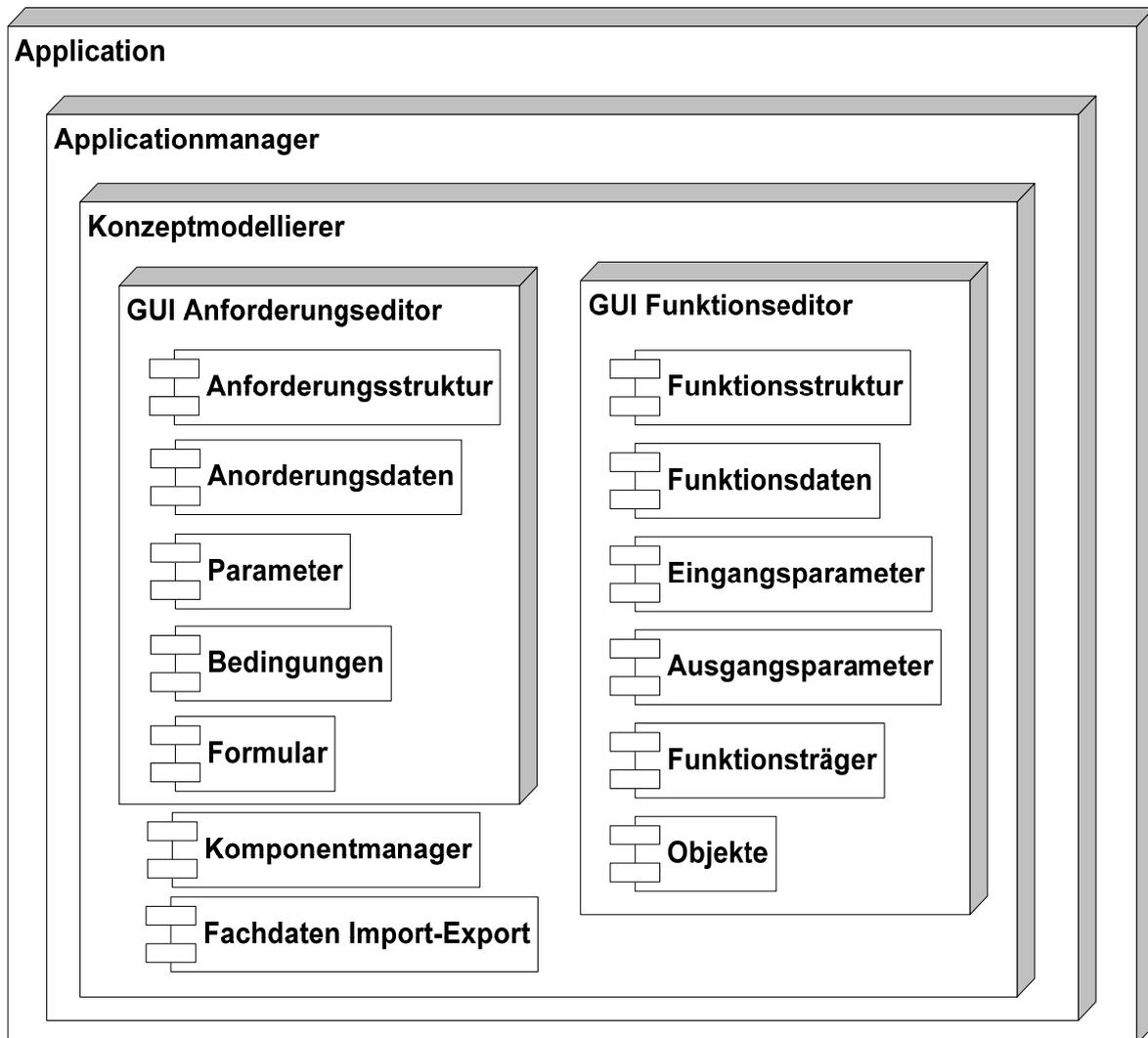


Bild 5-3: Komponentenmodell des Softwaresystems

5.2.5 Strukturklassenmodell

Basierend auf den oben dargestellten Modellen wird im Folgenden ein Strukturkonzept des zu entwickelnden Softwaresystems vorgeschlagen. Zu Beginn der Modellierung werden die Programmpakete (Bild 5-4) vom Komponentenmodell abgeleitet und durch die Definition der dazugehörigen Systemklassen präzisiert. Dieser Schritt führt dazu, dass bei der Entwicklung des Softwaretools die Fehleranfälligkeit auf ein Minimum reduziert wird.

In einem übergeordneten Paket *Application* werden Klassen definiert, die zur Verwaltung des Softwaresystems und zur Ausführung der allgemeinen Funktionen dienen. Die dazu notwendigen graphischen Komponenten der Benutzungsoberfläche werden in dem Unterpaket *GUI* definiert. Außerdem sind in dem Paket *Applikation* alle weiteren Unterpakete festzulegen. Diese

Aufteilung führt zur Erfüllung der grundlegenden Anforderung bei der Erweiterbarkeit von Softwaresystemen.

Das dem Paket *Application* untergeordnete *Paket Konzeptmodellierer* soll alle Klassen, die zur Verwaltung der Arbeit und der Komponenten des Modellierers dienen, sowie alle weiteren dafür notwendigen untergeordneten Pakete beinhalten. Weiterhin besitzt das Softwaresystem ein Fachdatenmodell, das im Kapitel 5.2.2 beschrieben wurde. Deren Klassen werden in dem Unterpaket *Fachdatenmodell* definiert, implementiert und verwaltet.

Das Softwaresystem muss über eine Benutzungsoberfläche zur Modellierung der rechnerinternen Produktkonzepte verfügen, welche wiederum aus einer *GUI* für die Anforderungs- und Funktionsmodellierung besteht. Die dafür notwendigen graphischen Komponenten werden in den Paketen *GUI-Anforderungseditor* und *GUI-Funktionsektor* definiert, implementiert und verwaltet.

Die notwendigen Klassen zum Einlesen/Importieren sowie zum Schreiben/Exportieren von Dateien mit rechnerinternen Konzepten werden in den Paketen *Import* und *Export* erstellt, implementiert und verwaltet.

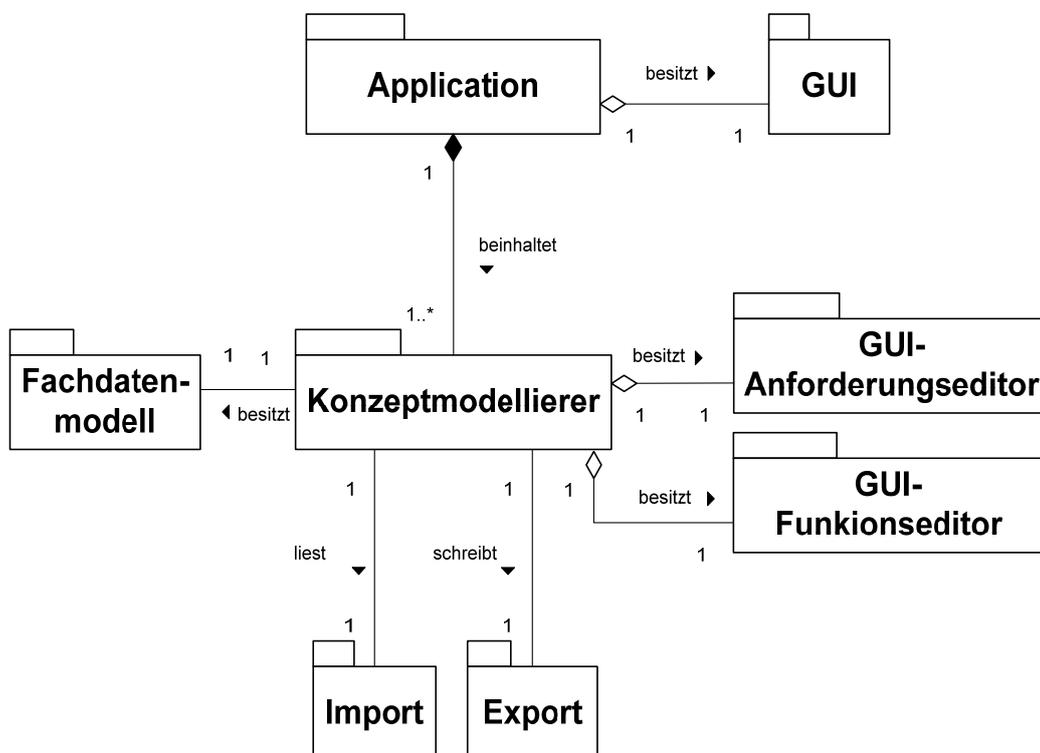


Bild 5-4: Paketenmodell des Softwaresystems

In einem abschließenden Konkretisierungsschritt müssen in den oben definierten Paketen Systemklassen definiert werden. Diese sollen die Abhängigkeiten und Hierarchien zwischen den Paketen konkretisieren und

darstellen. Damit wird ein detaillierter Aufbau des zu entwickelnden Systems erarbeitet. Ein Konzept der Systemklassen, deren Strukturierung und die dazugehörigen Namen werden im Folgenden näher beschrieben und sind im Bild 5-5 dargestellt.

Eine zentrale Anlaufstelle in dem Softwaresystem startet das Werkzeug, erleichtert seine interne Verwaltung und steuert sein gesamtes Verhalten. Für diese Aufgaben wird eine Klasse *Applicationmanager* in dem Paket *Application* definiert (vgl. Bild 5-3). Sie soll von den inhaltlichen Aufgaben des Werkzeuges (der rechnerinternen Konzipierung von Produkten) unabhängig funktionieren. Sie verwaltet ihrerseits eine grafische Klasse *ApplicationWindow* in dem Paket *GUI*, welche die Benutzungsoberfläche für allgemeine Verwaltungsaufgaben zur Verfügung stellt. Die Klasse *Applicationmanager* soll auch über einen Container für Menüs, die Iconleiste und weitere GUI-Klassen verfügen, die für die inhaltliche Arbeit der Anwendung notwendig sind.

Es muss in dem zu entwickelnden Softwaresystem auch eine Stelle geben, die die Verwaltung der inhaltlichen rechnerinternen Konzipierungsarbeit unterstützt. Hierfür wird die Klasse *Componentmanager* in dem Paket *Konzeptmodellierer* definiert. Dabei soll die Klasse *Componentmanager* die Komponenten zur Anforderungs- und Funktionsmodellierung sowie das Einlesen und Speichern von Dateien, in denen die rechnerinternen Konzeptmodelle gespeichert sind, über die Klasse *ImportExportManager* verwalten. Eine weitere Aufgabe dieser Klasse ist, die Klasse *Projekt* in dem Paket *Fachdatenmodell* (vgl. Bild 5-1), in dem alle Fachdaten erfasst und modelliert sind, zu verwalten.

Über die grafische Benutzungsoberfläche wird von dem Anwender die Anforderungsmodellierung initiiert, welche verschiedene Aufgaben beinhaltet, die in den grafischen Komponenten bearbeitet werden. Diese Komponenten sind in dem Paket *GUI-Anforderungseditor* zu definieren. Für ihre Verwaltung soll eine Klasse *Anf.EditorPanel* erstellt werden.

Die Aufgaben der Anforderungsmodellierung sind u. a. die Erfassung der verwaltungsbezogenen Daten, wie *Autor*, *Erstellungsdatum* oder *Stichwort*, welche mit Hilfe einer Klasse *Anf.DatenPanel* erfasst werden. Für die Darstellung der Anforderungsstruktur und deren grafische Modellierung soll eine *Anf.StrukturPanel* Klasse definiert werden. Für die Erfassung und Darstellung der Anforderungsparameter wird die Klasse *Anf.ParameterPanel* verwendet. Außerdem dienen die Klasse *Anf.BedingungPanel* zur Darstellung der Anforderungsbedingungen und die Klasse *Anf.FormularPanel* zur Definition sowie Darstellung der ausformulierten Anforderung.

Für die Verwaltung der Komponenten, mit denen die verschiedenen Tätigkeiten bei der Funktionsmodellierung realisiert werden, wird eine Klasse *Fkt.EditorPanel* definiert. Sie und die weiteren dafür notwendigen Komponenten lassen sich in dem Paket *GUI-Funktionseditor* erfassen. Die Darstellung der Funktionsstruktur erfolgt in der Klasse *Fkt.StrukturPanel*.

Der Name der Funktion und die verwaltungsbezogenen Daten sollen mit einer grafischen Komponente *Fkt.DatenPanel* dargestellt werden. Die Eingangs- und Ausgangsgrößen einer Funktion sind mit Hilfe der grafischen Komponenten *Fkt.InputParamPanel* und *Fkt.OutputParamPanel* zu erfassen und darzustellen.

Mit dem *Fkt.ObjektPanel* sollen die aufgabenspezifischen Komponenten verwaltet werden. Da einem Prozesswort unterschiedliche Objekte zugewiesen sein können, soll es mit der grafischen Komponente möglich sein, diese Objekte zu definieren und zu verwalten. Die Klasse *Fkt.ObjektListe* übernimmt die Aufgabe, die Objekte aufzulisten. Mit *Fkt.ObjektDatenPanel* sollen Name Beschreibung und andere verwaltungsbezogenen Daten erfasst werden. Mit *Fkt.ObjektParamPanel* kann das Objekt parametrisch beschrieben werden.

Da unterschiedliche Baugruppen oder Bauteile eine Funktion erfüllen können, soll die Möglichkeit bestehen, mehrere potenzielle Funktionsträger zu definieren und diese bedarfsgerecht einer Funktion zuzuweisen. Für die verschiedenen Tätigkeiten, die dabei auszuführen sind, werden entsprechende grafische Komponenten definiert und mit der Klasse *Fkt.TrägerPanel* verwaltet. Zur Darstellung der aufgelisteten Funktionsträger dient die Klasse *Fkt.TrägerListe*. Für die Definition und Darstellung des **Namens**, der Teilenummer und der Disziplin, aus der der Funktionsträger kommt, ist die Komponente *Fkt.TrägerDatenPanel* verantwortlich. Um die parameterische Beschreibung zu definieren, wird das Panel *Fkt.TrägerParamPanel* benutzt.

Wie oben bereits erwähnt, sollen die Aufgaben Lesen und Speichern durch die Klasse *ImportExportManager* umgesetzt werden. Dafür sind in dem Paket *Import* die Klasse *XMLImportReader*, die eine Datei lesen soll, und die Klasse *XML-ModellWandler*, die die eingelesenen Daten in ein internes Datenmodell umwandelt, zu definieren. In dem Paket *Export* sollen die Klasse *Model-XMLWandler*, die ein internes Datenmodell in eine XML Datenstruktur umwandelt, und eine Klasse *XML-ExportWriter*, die die Struktur auf die Festplatte oder zu einem PDM System zu speichern hilft, implementiert werden.

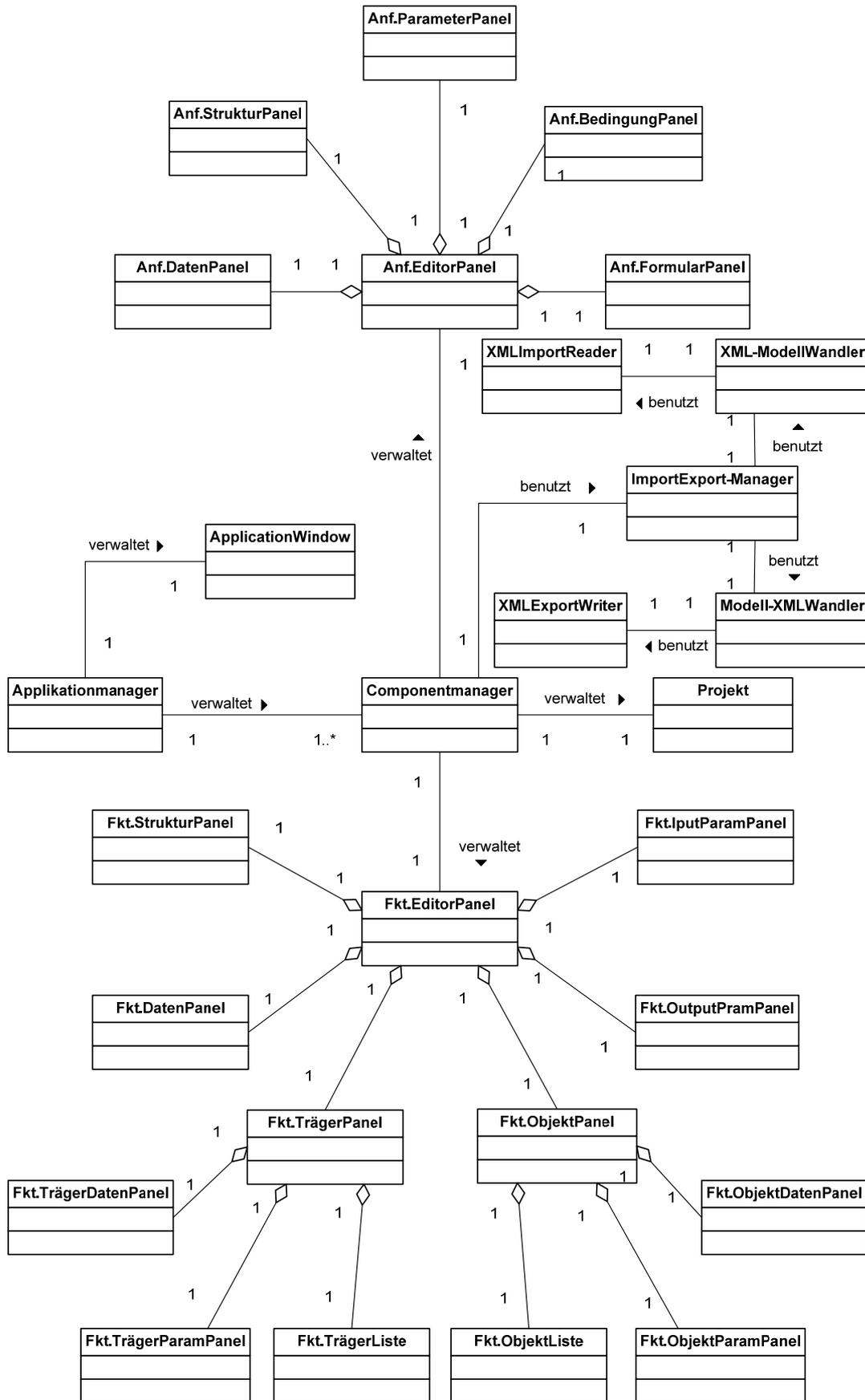


Bild 5-5: Strukturklassenmodell des Softwaresystems

5.3 Gestaltung der grafischen Benutzungsoberfläche

5.3.1 Einleitung

Die Benutzungsoberfläche ist das, was der Anwender von einer Software visuell wahrnehmen kann. Sie macht für ihn das Programm anfassbar und begreifbar. Sie vermittelt zwischen ihm und der Geschäftslogik. Deshalb ist ein gutes Zusammenwirken der Benutzungsoberfläche und der Geschäftslogik entscheidend für das Nutzen, die Gebrauchstauglichkeit und die Akzeptanz der Software durch den Anwender. Die Form der Oberfläche orientiert sich an der Funktion, aber auch an der Verwendung des Werkzeugs, weil die Funktion davon abhängt, wie, von wem und unter welchen Umständen das Werkzeug verwendet wird [51].

Aus den oben genannten Gründen ist es sowohl für den Anwender als auch für den Softwarehersteller von essentieller Bedeutung, eine Benutzungsoberfläche zu gestalten, die die Verwendung eines Softwaresystems mit seinen Funktionalitäten harmonisiert.

Der Einsatz einer grafischen Benutzungsoberfläche soll bei der Durchführung der rechnerintegrierten Konzipierung ihre Effektivität und Produktivität steigern. Dabei soll die GUI die Effizienz der Eingabe, der Erfassung der Definition und der Strukturierung der zur Konzipierung relevanten Daten erhöhen sowie einen möglichst vollständigen Überblick über die Darstellung der Arbeitsergebnisse unterstützen. Darüber hinaus muss die Oberfläche dem Anwender helfen, die rechnerintegrierte Vorgehensweise besser nachvollziehen zu können.

Eine Benutzungsoberfläche für allgemeine Aufgaben des Werkzeuges muss gewohnte Elemente und Anordnungen haben, um den Wiedererkennungseffekt zu erhöhen. Dadurch sollen der Einstieg in die Benutzung erleichtert und die Akzeptanz gesteigert werden. Für die allgemeine Funktionen, wie z. B. Datei öffnen, speichern, neues Projekt erstellen, kopieren, ausschneiden sowie einfügen, reichen die klassischen Untermenüs mit ihren Komponenten und deren gewohnte Reihenfolge völlig aus. Eine Symbolleiste unter den Menüpunkten soll Knöpfe haben, die erwartungsgemäß einen schnellen Zugang zu den meist benutzten Funktionen ermöglichen. Sie sollen mit Icons ausgestattet sein, die möglichst logisch die Funktion bildhaft darstellen.

5.3.2 Benutzungsoberfläche für die Anforderungsmodellierung

Für die Realisierung der Tätigkeiten, die in den Kapiteln 4.2.4.2, 4.3 und 4.4 beschrieben worden sind, muss ein Anforderungseditor eingesetzt werden. Die Funktionalitäten der Komponente wurden im Kapitel 5.1 beschrieben.

Innerhalb der Benutzungsoberfläche (GUI) sollen für die Erfassung der einfachen Daten kleine, einzeilige und für Anforderungsbeschreibungen große, mehrzeilige Eingabelemente verwendet werden. Die GUI soll über Darstellungskomponenten zum Anzeigen der Anforderungsstruktur und über Kontrollelemente (Knöpfe, Auswahlboxen) für die Kontrolle des Arbeitsflusses verfügen. Das Konzept zur Auslegung eines Anforderungseditors ist im Bild 5-6 zu sehen.

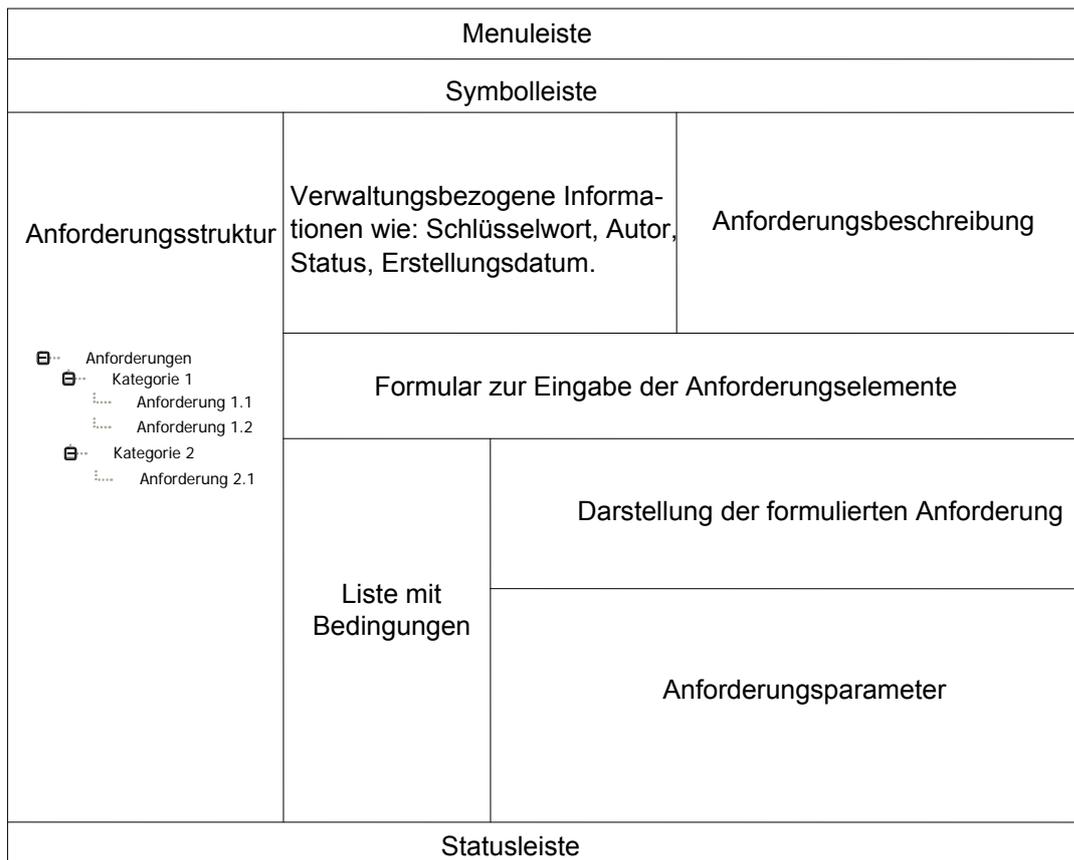


Bild 5-6: Auslegung einer Benutzungsoberfläche zur Definition und Strukturierung von Anforderungen

Anforderungsstruktur

Da bei der rechnerinternen Konzipierung Kategorien und Anforderungen definiert werden, soll eine Komponente deren baumartige Struktur abbilden. Sie ist auch der Einstieg in die Bearbeitung der Eigenschaften der Anforderungen. Da in Deutschland das lateinische Alphabet benutzt wird, sind die Menschen gewohnt, von links nach rechts zu schreiben. Außerdem liegen bei den meisten Programmen die Navigationselemente (Liste oder Strukturbaum) auf der linken Seite des Bildschirms. Daher soll auch die Anforderungsstruktur auf dieser Seite platziert sein, um den Wiedererkennungseffekt, die Einarbeitung und die Akzeptanz des Softwaresystems zu steigern (Bild 5-6).

Verwaltungsbezogene Informationen

Nach dem Prinzip „von oben (abstrakt) nach unten (konkret)“ sollen weiterhin verwaltungsbezogene Informationen, wie Schlüsselwort, Autor, Erstellungs-/ Änderungsdatum oder Version oben auf der rechten Seite der Struktur dargestellt werden. Rechts von den Informationen wird eine textuelle Beschreibung zu finden sein. Dieser Bereich stellt den abstrakten Teil der Definition einer Anforderung dar.

Formular für die Eingabe der Anforderungselemente

Weitergehend mit der Konkretisierung „nach unten“ soll das Formular mit der Schablone zur Definition von Anforderungselementen platziert sein. Als eine optimale Basis für eine Schablone eignet sich der normale Satzbau der deutschen Sprache, die nur über eine bestimmte Anzahl von Elementen verfügt (vgl. Kapitel 4.2.4.2). Um beispielsweise einen Modalsatz zu definieren, hat man in der deutschen Sprache nur acht Elemente, woraus auch eine Anforderung bestehen kann.

Liste mit Bedingungen

Neben dem o. g. Formular ist eine Stichwortliste mit Bedingungen einer Anforderung zu platzieren. Da die Bedingung auch einen Satz wie die Anforderung darstellt, ist es deshalb vorteilhaft, die Bedingungen auch mit dem gleichen Formular zu definieren, um den Entwicklungsaufwand zu reduzieren.

Ausformulierte Anforderung

Unterhalb des o. g. Formulars ist die ausformulierte Anforderung als nächste Konkretisierungsstufe darzustellen. Sie soll sich in Schrift und Farbe von dem Formular abheben, um den fertigen Status der Formulierung herauszustellen. Die Darstellung darf nicht editierbar sein, um die Konsistenz der Dateneingabe zu gewährleisten.

Anforderungsparameter

Im unteren Bereich des Editors befindet sich eine Tabelle mit Anforderungsparametern, die diese Anforderung abschließend präzisieren und eindeutig ausdrücken. Angefangen von der Beschreibung bis hin zu den Parametern, soll hierdurch die Präzisierung und Definition der Anforderung mit einem Blick gut nachvollziehbar sein.

5.3.3 Benutzungsoberfläche für die Funktionsmodellierung

Zur Realisierung der Tätigkeiten aus den Kapiteln 4.2.4.3, 4.3 und 4.4 soll ein Funktionseditor eingesetzt werden. Die Funktionalitäten dieses Editors wurden bereits im Kapitel 5.1 beschrieben. Die Auslegung der Elemente des Editors soll nach den gleichen Prinzipien wie bei dem Anforderungseditor erfolgen.

Funktionsstruktur

Die baumartige Funktionsstruktur kann mit Hilfe des Vorgehens aus dem Kapitel 4.2.4.3 erstellt werden. Die Darstellungskomponente der Funktionsstruktur soll aus den gleichen Gründen wie beim Anforderungseditor auf der linken Seite des Funktionseditors platziert sein. Sie hat aber die optionale Darstellung der zugewiesenen Funktionsträger sowie die Darstellung des zweiteiligen Funktionsnamens (Objekt und Prozesswort) zu berücksichtigen (Bild 5-7).

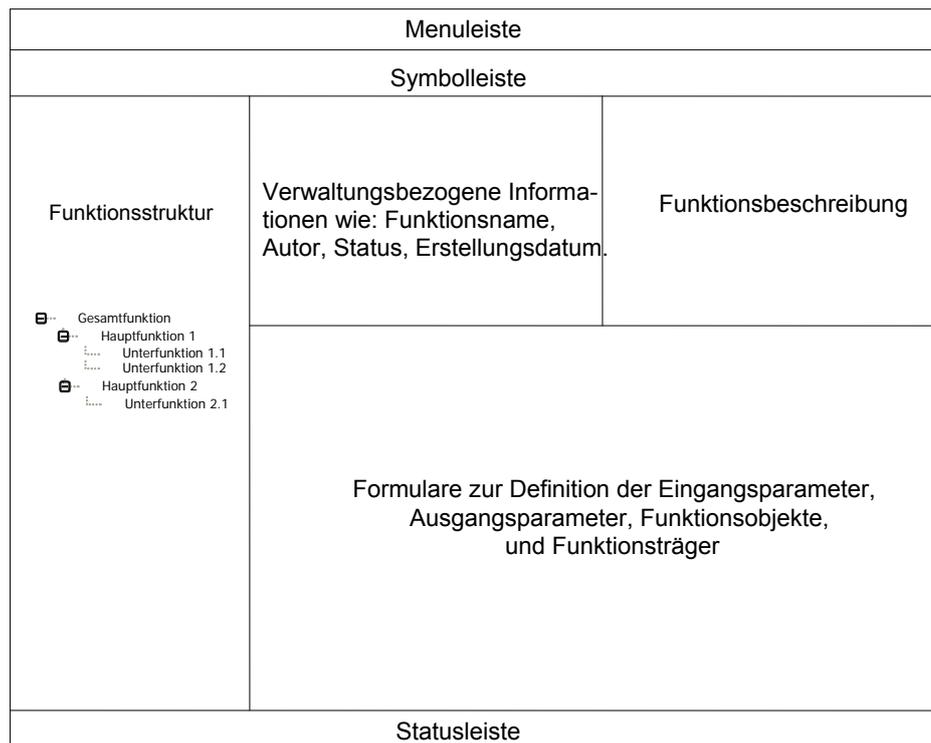


Bild 5-7: Auslegung einer Benutzungsoberfläche zur Definition und Strukturierung von Funktionen und Funktionsträger

Verwaltungsbezogene Informationen und Funktionsbeschreibung

Dieser Bereich ist dem Anforderungseditor gleich und soll aus ähnlichen Gründen an der gleichen Stelle des Funktionseditors zu finden sein.

Er unterscheidet sich jedoch von dem Anforderungseditor darin, dass hier der Funktionsname – bestehend aus zwei Elementen – definiert sein soll. Auch die Verwaltungsinformationen des Funktionsträgers können hier optional gezeigt werden.

Formulare zur Definition von Eingangs- und Ausgangsparametern

Um das methodische Vorgehen aus dem Kapitel 4.2.4.3 realisieren zu können, sollen auch Formulare zur Definition von Eingangs- und Ausgangsgrößen einer Produktfunktion benutzt werden. Zur Definition der Parameter eignen sich sehr gut tabellarische Softwarekomponenten, die gleichzeitig für die Eingabe als auch für die Darstellung der Daten eingesetzt werden können. Sie sollen unterhalb der Komponenten mit verwaltungsbezogenen Informationen und der Funktionsbeschreibung platziert sein.

Formular zur Definition der Funktionsobjekte

Da die Funktionsdefinition durch ein Funktionsobjekt generalisiert oder spezialisiert werden kann (vgl. Kapitel 4.2.4.3), muss sich in dem Funktionseditor auch ein entsprechendes Formular befinden, das die Definition, Verwaltung und Zuweisung von mehreren Funktionsobjekten unterstützt. Es soll unterhalb der Formulare zur Definition der Eingangs- und Ausgangsparameter liegen. Angefangen von links, soll es zunächst einen Bereich mit einer Liste der Funktionsobjekte, einen anderen Bereich mit verwaltungsbezogenen Informationen (Name, ID) sowie (darunter platziert) eine Beschreibung des Objekts haben. Auf der rechten Seite des Formulars soll die parametrische Beschreibung der Funktionsobjekte zu sehen sein, was für die Anpassung der Funktion an geänderte Produkthanforderungen sehr hilfreich ist. Bild 5-8 stellt die Auslegung des Formulars dar.

Liste mit Namen	Verwaltungsbezogene Informationen	Parametrische Beschreibung
	Natursprachliche Beschreibung	

Bild 5-8: Auslegung eines Editors zur Definition von Funktionsobjekten oder Funktionsträger

Formular zur Definition der Funktionsträger

Da mehrere Funktionsträger aus unterschiedlichen Disziplinen eine angeforderte Produktfunktion erfüllen können, muss zu ihrer Definition ein entsprechendes Formular dem Entwickler angeboten werden. Er soll über eine Liste mit den Namen der Funktionsträger verfügen, einen Bereich mit den verwaltungsbezogenen Informationen, die auch die Disziplin berücksichtigen, sowie textuelle Beschreibung haben. So wie bei dem o. g. Formular soll sich in ihm auch ein Bereich befinden, der die parametrische Definition der Funktionsträger anbietet. Eine zusätzliche Komponente kann hier ergänzend die Verknüpfung der Funktionsträger mit existierenden CAD-Dateien ermöglichen, um die Abschätzung der Relevanz für die Zuweisung zu einer Funktion visuell zu unterstützen. Das Formular soll unterhalb des Formulars zur Definition der Funktionsobjekte platziert sein (vgl. Bild 5-8).

6 Prototypische Systemimplementierung und Evaluierung des Ansatzes

6.1 Prototypische Systemimplementierung

Dieses Kapitel beschreibt die beispielhafte, prototypische Implementierung eines Softwarewerkzeugs zur rechnerintegrierten Konzipierung. Das Werkzeug wurde in der plattformunabhängigen Programmierungssprache Java™ /52/ implementiert, die über eine komfortable XML-Verarbeitung zur Datenspeicherung sowie über eine große Bibliothek mit grafischen Komponenten – Java Swing™ /53/ verfügt. Der Autor hat den Softwareprototyp selbständig implementiert.

Die Ausarbeitung und Implementierung der einzelnen Konzepte verlief iterativ und prototypenzentriert. Die Grundlagen zu den Arbeiten lieferten die im Kapitel 3.3 definierten Anforderungen, aus denen im Kapitel 5.1 die Systemfunktionalitäten abgeleitet wurden. Die im Kapitel 5.2 erarbeitete Architektur wurde um technologie-spezifische Klassen und Aspekte erweitert. Basierend auf diesen Vorarbeiten wurde die im Kapitel 5.3 konzipierte graphische Benutzeroberfläche realisiert.

Als Basis zur Implementierung des Anforderungseditors diente die im Bild 5-6 dargestellte Auslegung der Elemente. Die prototypische Implementierung des Editors ist im Bild 6-1 zu sehen. Im Folgenden wird auf die Realisierung der wesentlichen Elemente näher eingegangen. Diese sind auf dem Screenshot mit laufenden Nummern versehen.

Der Editor wurde mit Hilfe der Java Swing™-Klasse *JTabPane* realisiert. Diese Lösung erlaubt weitere Panels mit grafischen Oberflächen, wie z. B. Metriken zur Auswertung der Anforderungen oder Tabellen mit disziplinbezogen ausgefilterten Anforderungsparametern, zu implementieren.

Anforderungsstruktur

Die Anforderungsstruktur ist mit Hilfe der *JTree*-Komponente dargestellt (1) und um komfortablen Möglichkeiten der Strukturierung der Anforderungen erweitert. Um die Transparenz der Darstellung zu erhöhen, wurden verschiedene Icons zur Kennzeichnung der Kategorien, der funktionalen (2) und nichtfunktionalen (3) Anforderungen angewendet.

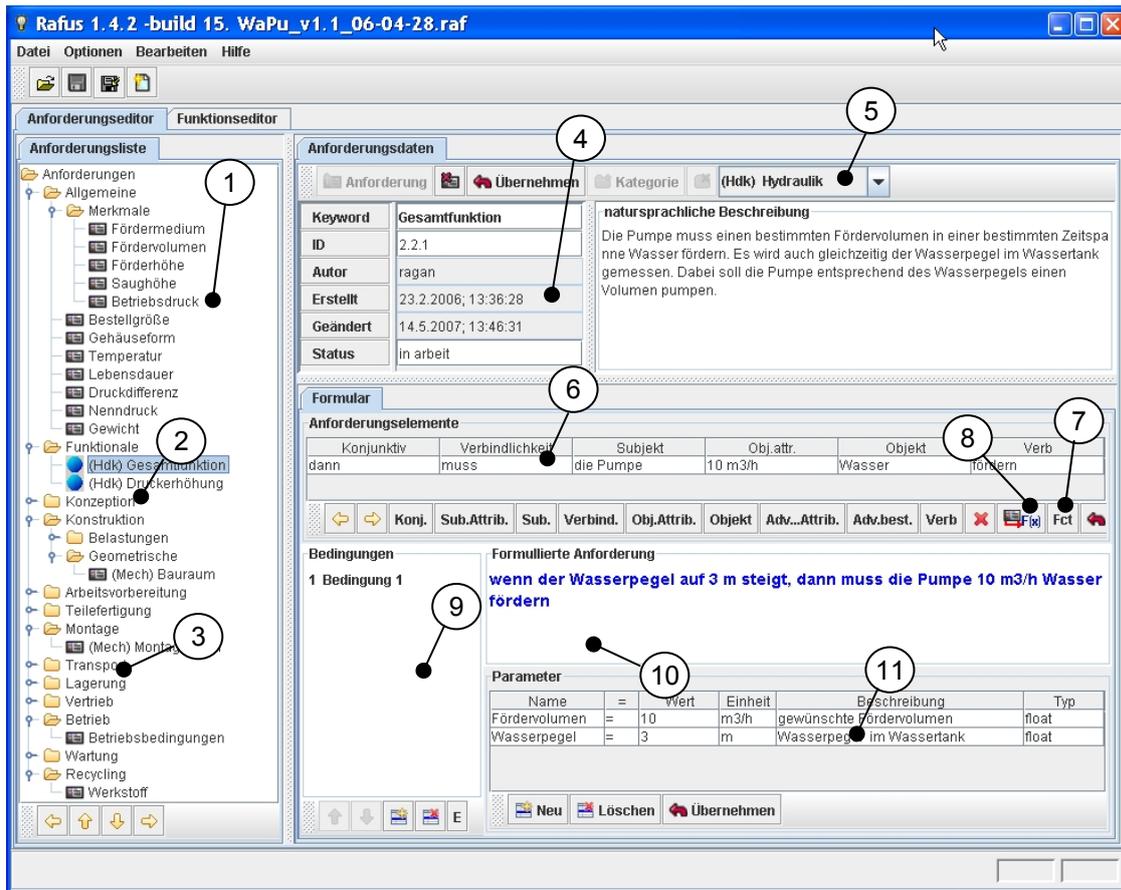


Bild 6-1: Realisierung des Anforderungsektors

Verwaltungsbezogene Informationen

Die verwaltungsbezogenen Informationen können mit der Komponente *JTextField* aufgenommen werden. Diese Informationen, wie z. B. *Autor*, *erstellt* und *geändert*, sind durch das Betriebssystem vorgegeben und können durch den Benutzer nicht editiert werden (4). Zusätzlich kann die Beschreibung der Anforderung mit Hilfe des *JEditorPane* (Bild 6-1, rechts) aufgenommen werden. Die Zuordnung einer Anforderung zu einer Disziplin wie Mechanik oder Hydraulik ist mit Hilfe einer *JComboBox*-Auswahl implementiert (5).

Formular für die Eingabe der Anforderungselemente

Das Formular zur Eingabe der Anforderungselemente (6) wurde mit Hilfe der JAVA-Klasse *JTable* erstellt. Es ermöglicht die Anordnung der Reihenfolge der Elemente, die Kennzeichnung der Anforderung als funktionale bzw. nichtfunktionale Anforderung (7) und die Ableitung einer Funktion von einer funktionalen Anforderung (8).

Liste mit Bedingungen

Die Liste mit Bedingungen einer Anforderung ist mittels *JList* realisiert (9). Auch die automatisch absteigende Sortierung der neu definierten Bedingungen wurde implementiert. Die Erfassung einer Bedingung erfolgt mit dem Formular zur Definition von Anforderungen, weil dafür die gleichen Elemente benutzt werden können.

Formulierte Anforderung

Die textuell formulierte Anforderung mit dazugehöriger Bedingung wird nachfolgend mit Hilfe des *JTextPane* (10) in blauer Farbe dargestellt, kann aber nicht in dem Panel editiert werden. Farbe und Schriftgröße wurden ausgewählt, um die Lesbarkeit der Anforderung zu erhöhen und auch hervorzuheben, dass sie fertig ist und einen offiziellen Charakter hat.

Anforderungsparameter

Die Anforderungsparameter werden mit Hilfe von *JTable* aufgenommen und dargestellt (11). Dabei gibt es die Möglichkeit, mehrere Parameter für eine Anforderung zu definieren, um die Bedingungen oder eine abstrakte Formulierung präziser auszudrücken.

Als Grundlage für die Implementierung des Funktionseditors diene die im Bild 5-7 dargestellte Auslegung der Elemente. Der Editor wurde mit *JTabPane* erstellt. Dies ermöglicht, weitere Komponenten wie z. B. zusätzliche verwaltungsbezogene Informationen, statistische Auswertungen, Darstellung der Funktionsstruktur mit Hilfe von Graphen zu implementieren. Seine prototypische Realisierung ist im Bild 6-2 dargestellt. Im Folgenden wird auf die Umsetzung der wesentlichen Elemente kurz eingegangen.

Funktionsstruktur

Die Funktionsstruktur ist mit Hilfe der *JTree*-Komponente dargestellt (1). Sie wurde mit komfortablen Möglichkeiten der Strukturierung ausgestattet und in ein *JTabPane* eingebettet. Dies ermöglicht, mehrere Komponenten mit Funktionsstrukturen zu verwalten. Daher wurden entsprechende Menüpunkte implementiert, um eine Funktionsstruktur als neue Variante in einem neuen Panel zu kopieren und weiter zu bearbeiten oder einen neuen leeren Panel zu erzeugen. Auf diese Weise besteht die Möglichkeit mehrere Funktionsstrukturvarianten komfortabel zu erstellen und schnell zu bewerten.

Weiterhin wurden entsprechende Menüpunkte mit Hilfe des *JButton* implementiert. Durch deren Ein- und Ausschalten werden die Funktionsobjekte (2) oder Funktionsträger (3) in der Funktionsstruktur gezeigt oder verborgen. Dadurch erhöht sich die Möglichkeit ihrer transparenten Darstellung und besseren Bewertung bei der Konzipierung.

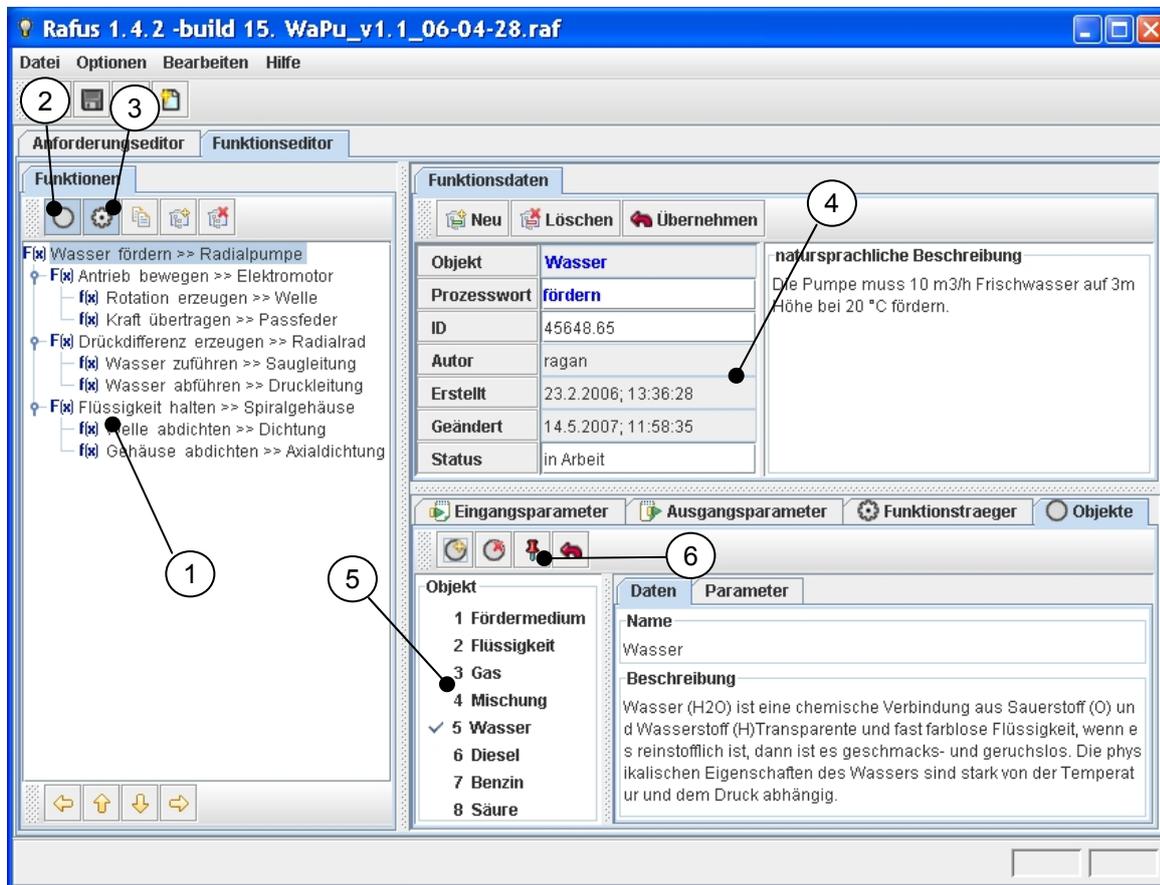


Bild 6-2: Realisierung des Funktionseditors

Verwaltungsbezogene Informationen und Funktionsbeschreibung

Die Definition des Prozesswortes der Funktion, die Darstellung des ausgewählten Funktionsobjekts sowie die Aufnahme der verwaltungsbezogenen Informationen einer Produktfunktion wurden mit Hilfe von *JTextField* implementiert (4). Allerdings darf der Name des Funktionsobjekts hier nicht geändert werden, weil dazu ein extra entwickeltes Formular bereit steht. Weiterhin kann die textuelle Beschreibung der Funktion mit Hilfe des *JEditorPane* durchgeführt werden.

Formular zur Definition der Funktionsobjekte

Dieses Formular wurde basierend auf der Auslegung aus dem Bild 5-8 mit Hilfe von *JTabPane* implementiert. Der Name des Funktionsobjekts wird mit

JTextField aufgenommen. Seine textuelle Beschreibung wird mit *JEditorPane* definiert, seine parametrische Beschreibung mit *JTable*. Die erfassten Funktionsobjekte werden als eine durchnummerierte Liste mit Hilfe von *JList* (5) dargestellt. Die Zuordnung eines Objekts zu einer Funktion wurde mit Hilfe des *JButton* (6) realisiert und mit einem Icon visuell gekennzeichnet. Die Darstellung des Funktionsobjekts in der Funktionsstruktur folgt erst dann, wenn deren Anzeige eingeschaltet worden ist (2).

Formulare zur Definition von Eingangs- und Ausgangsparameter

Basierend auf der Auslegung aus dem Bild 5-7 wurden die Formulare für die Definition der Eingangs- und Ausgangsgrößen (vgl. Bild 6-3) als Tabellen mit Hilfe von *JTable* implementiert (1). Sie wurden dann in einem *JTabPane* eingebettet. Diese Lösung ermöglicht die Erweiterung des Werkzugs um weitere Panels, die beispielsweise zur Definition von Constraints benutzt werden können. Weiterhin ist für jeden Eingangs- oder Ausgangsparameter die Möglichkeit der Kennzeichnung als Stoff, Energie oder Information mit Hilfe eines *JComboBox*-Auswahls (2) gegeben.

Name	=	Wert	Einheit	Beschreibung	Typ
Fördervolumen	=	10	m ³ /h	gewünschte Fördervolumen	integer
Förderhöhe	=	3	m	gewünschte Förderhöhe	float
Fördermedium	=	Wasser			text
Temperatur	<	90	°C	Maximale Betriebstemperatur	integer
Saughöhe	<	5	m	maximale Saughöhe	integer
Betriebsdruck	<	10	bar	maximale Betriebsdruck	integer
B_M	=	260	mm	Montageraumbreite	integer
H_M	=	300	mm	Montageraumhöhe	integer
T_M	=	550	mm	Montageraumtiefe	integer

Bild 6-3: Formular zur Definition der Funktionseingangsgrößen

Formular zur Definition der Funktionsträger

Die Implementierung dieses Formulars erfolgte mit Hilfe von *JTabPane* basierend auf der im Kap. 5.3.3 Bild 5-8 definierten Auslegung. Der Name des Funktionsträgers kann mit Hilfe von *JTextField* definiert werden. Die Zuordnung zu einer Disziplin findet mit einem *JComboBox*-Auswahl (1) statt. Die JAVA SWING™-Komponente *JEditorPane* dient zur Aufnahme der textuellen Beschreibung des Funktionsträgers.

Die parametrische Beschreibung des Funktionsträgers sowie die Definition von Verknüpfungen mit Dateien, wie z. B. CAD-Dateien, werden mit der Komponente *JTable* realisiert (2). Sie sind jeweils in ein *JTabPane* eingebettet. Diese Lösung ermöglicht, die Benutzungsoberfläche um

weitere Panels zu erweitern, die beispielsweise den Funktionsträger visuell darstellen können oder erlauben, Constraints zwischen den Parametern zu definieren.

Die somit erfassten Daten werden als Liste mit *JList* (3) dargestellt, in welcher der ausgewählte Funktionsträger mit einem entsprechenden Icon gekennzeichnet ist. Mit dem Menü-Punkt *JButton* wurde die Zuordnung eines Funktionsträgers zu einer Funktion implementiert (4). Der Aktivierung dieses Elementes kann auch die Darstellung des Funktionsträgers in der Funktionsstruktur folgen, allerdings unter der Bedingung, dass die Anzeige der Funktionsträger in der Funktionsstruktur eingeschaltet ist (vgl. im Bild 6-2 die Position (3)).

Name	=	Wert	Ein...	Beschr...	Typ
Q	=	10	m ³ /h	Förder...	int...
H_max	=	40	m	Förder...	float
P	=	2	kW	Leistung	float
DNS	=	65	mm	Flansc...	int...
DND	=	40	mm	Flansc...	int...
Pumpe...	=	GG25		Material	text
LaufRad	=	GG ...		Material	text
Gehäus...	=	Stahl		Material	text
Pumpe...	=	Stahl		Material	text

Bild 6-4: Formular zur Definition der Funktionsträger

6.2 Praxisbeispiel einer Montagemaschine

Anhand eines Praxisbeispiels aus der Auftragsentwicklung einer Montagemaschine wird im Folgenden die Anwendung der rechnerintegrierten Methode bei einer Neukonzipierung evaluiert. Ausgangspunkt des betrachteten Szenarios ist die Angebotsanfrage zur Erstellung einer Maschine für die Montage eines Rollenlagers auf eine Getriebewelle entsprechend einer vorgegebenen Montagereihenfolge. Ein Beispiel dieser Maschine ist im Bild 6-5 dargestellt.

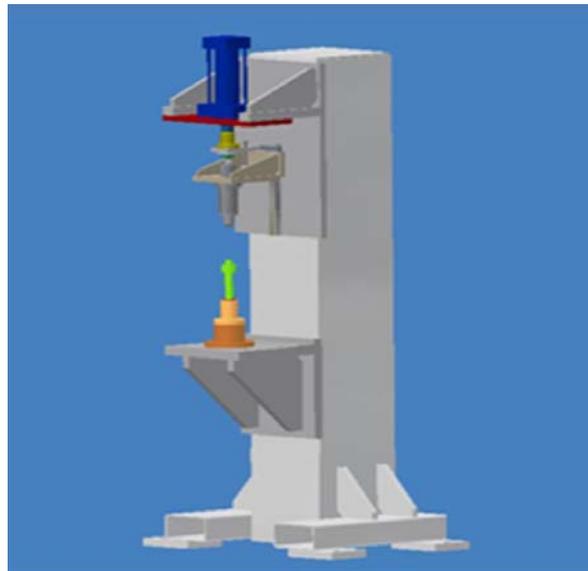


Bild 6-5: Montagemaschine für Getriebewelle⁴

Bei der Analyse der Kundenanforderungen haben sich Besonderheiten in Bezug auf die Linienkonformität, Fehlfunktionsprüfung, Messung der Fügekraft als Qualitätskriterium und Ausführung nach QM-Zertifizierung herausgestellt. Es wurde festgestellt, dass es eine derartige Maschine nicht in dem Portfolio des Unternehmens gibt, aber Ähnlichkeiten zu existierenden Maschinen bestehen und deswegen relevante Daten vorhanden sind, die man bei dem Auftrag nutzen kann. Daher ist die Maschine technisch und wirtschaftlich machbar.

Die rechnerintegrierte Konzipierung einer Neuentwicklung mit Hilfe des Werkzeug *Rafus*, dessen Prototyp im Kapitel 6.1 beschrieben wurde, beginnt mit der Erstellung eines neuen Anforderungsmodells.

Im ersten Arbeitsschritt gemäß der Vorgehensweise bei der Anwendung werden die relevanten Kategorien – u. a. „*Stationspflichtenheft*“, „*Steuerungshinweise*“ oder „*Maschinenabmessungen*“ – identifiziert und in dem *Rafus*-Werkzeug hierarchisch gegliedert.

⁴ Mit freundlicher Genehmigung der Firma CADsys GmbH, Chemnitz

Während der Analyse der Unterlagen wurden anhand der Kundenvorschriften die Anforderungen nacheinander identifiziert und in dem Softwarewerkzeug mittels eines Stichworts (wie z. B. „Gesamtfunktion“, „Fehlfunktion“ oder „Bediensystem“) sowie mit Hilfe der Fliesstext-Beschreibung erfasst. Dann werden diese als funktionale (z. B. „Fehlfunktion“) bzw. nichtfunktionale Anforderung (z. B. „Bediensystem“) gekennzeichnet.

Nachfolgend wurden die Anforderungen mit Hilfe des Formulars als Modalsätze präzisiert und ggf. durch weitere Anforderungen oder Bedingungen erweitert. Ein Beispiel des Anforderungsmodells und einer Anforderung in dem *Rafus*-Werkzeug wird im Bild 6-6 dargestellt. Das Formular zeigt, dass das Subjekt der Anforderung „Montagemaschine“ ein Objekt „Bediensystem BS32-S341“ haben muss, allerdings unter der Bedingung, dass ihr SPS-System ein „Siemens S7 300“ ist. Anschließend wurden die Anforderungen den unterschiedlichen Disziplinen, wie Hydraulik, Elektrotechnik, Mechanik oder Steuerung „(Stg) Bediensystem“ zugeordnet. Im letzten Schritt erfolgte bei Bedarf die parametrische Beschreibung, wie zum Beispiel „BSys=BS32-S341“.

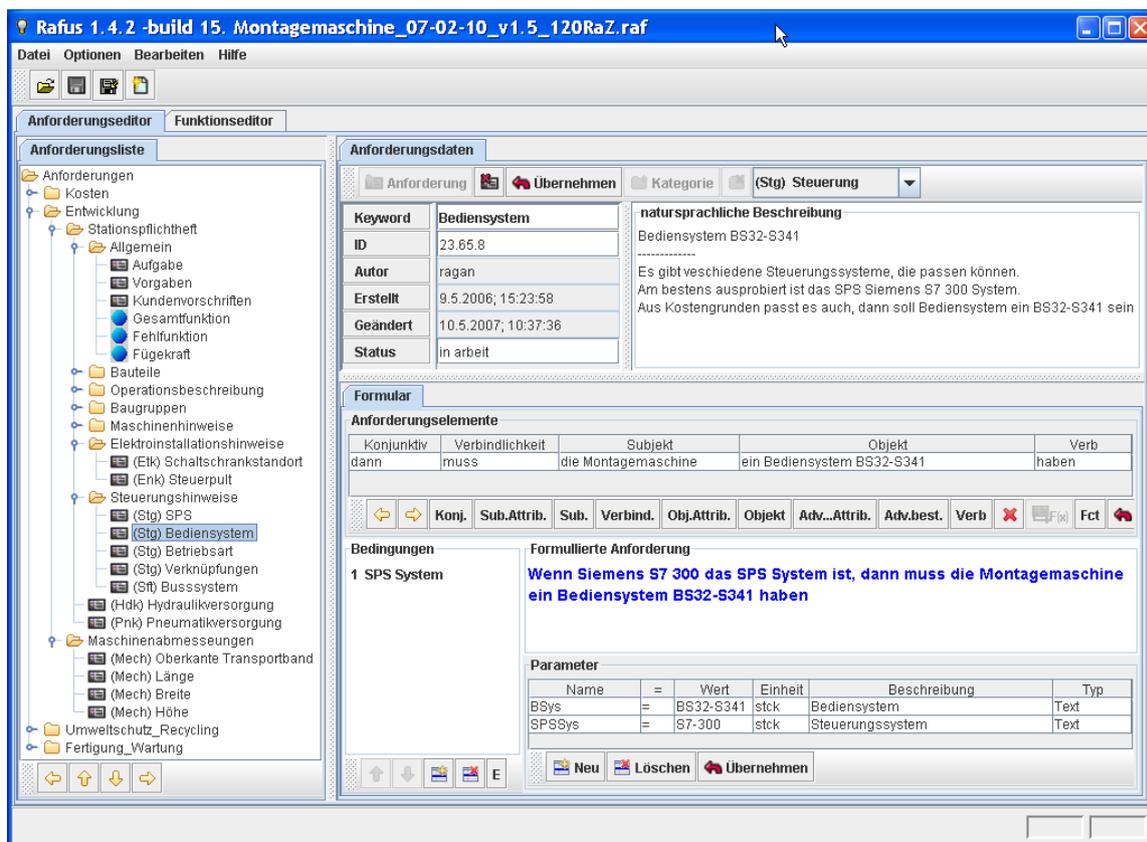


Bild 6-6: Anforderungsmodell einer Montagemaschine

Im nächsten Schritt der Konzipierung wurden aus den funktionalen Anforderungen die gewünschten Funktionen der Maschine, wie z. B.

„Lagerring auf Triebwelle fügen“, mit Hilfe des Werkzeugs *Rafus* automatisiert extrahiert, manuell zerlegt, durch andere Funktionen wie „Triebwelle positionieren“ vervollständigt und strukturiert. Dabei wurden die Verben, die in den Anforderungen auftreten, als Prozessworte in Funktionen und Objekte aus dem Formular anschließend als Objekte in den Funktionsnamen übernommen (vgl. Kapitel 4).

Weiterhin wurde für jede Funktion jeweils eine Beschreibung erstellt. Anschließend wurden bei den Funktionen teilweise aus den Anforderungen und teilweise aus der Erfahrung und bestimmten Dokumenten die Eingangsparameter, wie „Triebwelle = true“, „X_Referenzpunkt=0“ und Ausgangsparameter wie „Positioniert = true“, festgelegt. In dem nächsten Schritt wurden die Funktionsträger definiert, die eine Lösung oder ein Lösungsprinzip darstellten, die für die weitere Gestaltung des Produkts notwendig sind. Beispielsweise wurde für „Triebwelle positionieren“ der Funktionsträger „Zentrierpin“ bestimmt und parametrisch beschrieben.

Diese erste Konkretisierung half, die Lösung im Hinblick auf die Aufgabenerfüllung abzuschätzen. Anschließend wurden mehrere Funktionsmodelle der Maschine erstellt, analysiert und zum Schluss die vierte modellierte Variante als eine passende Funktionslösung ausgewählt. Das Beispiel des Funktionsmodells, der Funktion und des Funktionsträgers in dem *Rafus*-Werkzeug zeigt Bild 6-7.

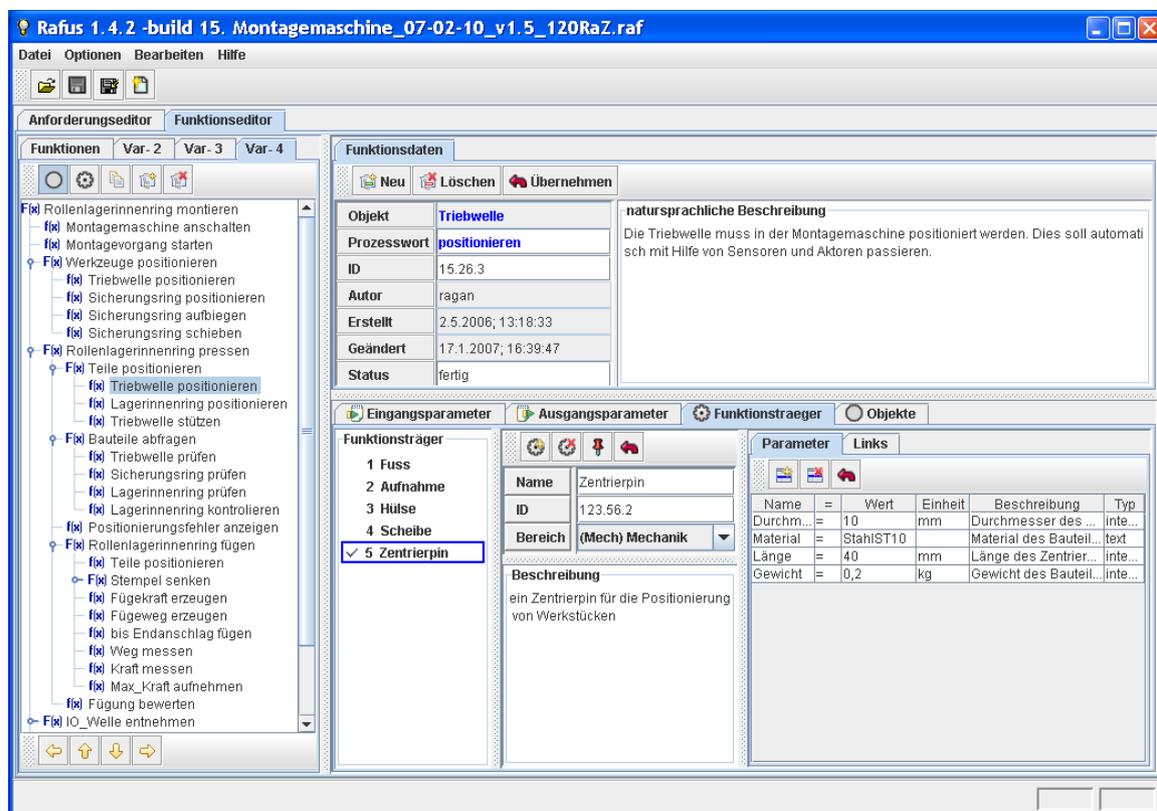


Bild 6-7: Funktionsmodell einer Montagemaschine

Die beispielhafte Modellierung der Montagemaschine anhand der erarbeiteten Vorgehensweise mit Hilfe des Softwareprototyps belegt, dass eine derartig präzise und benutzerfreundliche Definition der Anforderungen, die mit gegenwärtigen Werkzeugen nicht möglich ist, ein besseres Verständnis über die zu entwickelnde Maschine bewirkt. Dies führt wiederum im Verlauf der in dieser Weise methodisch formulierten Konzipierung, im Vergleich zu existierenden Ansätzen, schneller zu Funktionsmodellen, die adäquat Produkthanforderungen erfüllen sowie eine effektive und zeitsparende Eingrenzung der Lösung ermöglichen. Dadurch können bei einer derart verbesserten Produktentwicklung mehrere Iterationsschleifen vermieden, die Qualität der Ergebnisse erhöht, Zeit eingespart und Wirtschaftlichkeit gesteigert werden.

6.3 Fallbeispiel eines Autositzes

Die Anwendung des Ansatzes zur rechnerintegrierten Konzipierung einer Anpassung wird anhand des Praxisbeispiels zur Auftragsentwicklung eines Autositzes gezeigt. Ziel der Untersuchung ist, ein „minimales“ rechnerinternes Basismodell einer Sitzanlage zu schaffen, das als Grundlage für Auftragsentwicklungen von weiteren Fahrersitzen dienen kann. Durch die Erweiterung des rechnerinternen Basismodells kann das kundenspezifische Konzept eines Sitzes entwickelt werden, der alle Sicherheitsaspekte erfüllt und diese zusätzlich mit einem hohen Komfortwert verbindet. Da aber ein Komfort schwer formalisierbar ist und der gefühlte Komfort bei unterschiedlichen Personen zu sehr individuellen Aussagen führt, werden in der Konzeptphase immer wieder neue Wege gesucht, um eine Lösung möglichst präzise einzugrenzen. Ein effektiver Weg hierzu ist eine rechnerintegrierte, modellbasierte Konzipierung. Bild 6-8 zeigt einen Fahrersitz.



Bild 6-8: Fahrersitz⁵

⁵ Mit freundlicher Genehmigung der Firma INTEC GmbH & Co. KG, Poing (bei München)

Vorbereitung

Zunächst wurde untersucht, wie komplex die Erstellung eines rechnerinternen Modells eines Autositzes ist. Dabei stellte man fest, dass ein einziges minimales Grundmodell für den Fahrer- und Beifahrersitz vollkommen ausreichend ist und die Nutzung einer rechnerintegrierten Methode mit Sicherheit kostengünstig sein wird. Der Grad der Wirtschaftlichkeit lässt sich aber erst nach der Realisierung einiger Projekte eindeutig feststellen. Dabei spielt auch die Wiederverwendung der rechnerinternen Modelle eine entscheidende Rolle.

Gemäß der o. g. Vorgehensweise wurde nachfolgend ein rechnerinternes Produktmodell erstellt. In vorhandenen Unterlagen identifizierte und anschließend extrahierte man relevante Kategorien, wie z. B. „*Sicherheit*“, „*Recycling*“ oder „*Airbagintegration*“. Diese wurden in dem *Rafus*-Werkzeug hierarchisch gegliedert.

Während der weiteren Analyse des Produkts erfolgte die Identifizierung der Anforderungen mittels Stichwörtern, wie z. B. „*Längsverstellung*“, „*Elektrische Überprüfungen*“, „*Sitzbelegungserkennung*“, und deren Erfassung durch die Textbeschreibung in dem Softwarewerkzeug. Anschließend wurden diese Anforderungen als funktional bzw. nicht funktional gekennzeichnet und ggf. durch weitere Unteranforderungen oder Bedingungen präzisiert.

Im weiteren Verlauf der Erstellung des rechnerinternen Produktmodells wurden aus funktionalen Anforderungen allgemeine Funktionen eines Fahrer- und Beifahrersitzes mit Hilfe des Softwarewerkzeugs extrahiert, in Unterfunktionen zerlegt, durch zusätzliche Funktionen wie „*Kopf schützen*“ vervollständigt und dabei strukturiert. Außerdem erstellte man textuelle Beschreibungen der Funktionen. Darüber hinaus wurden die Eingangs- und Ausgangsparameter – basierend auf den Anforderungen und teilweise auf der Praxiserfahrung – definiert. Im letzten Schritt wurden die Funktionsträger“, die eine Lösung oder ein Lösungsprinzip wie „*Kopfstütze*“ oder „*Seitenairbag*“ darstellen können, festgelegt und parametrisch beschrieben.

Das Funktionsmodell eines Autositzes ist im Bild 6-9 dargestellt.

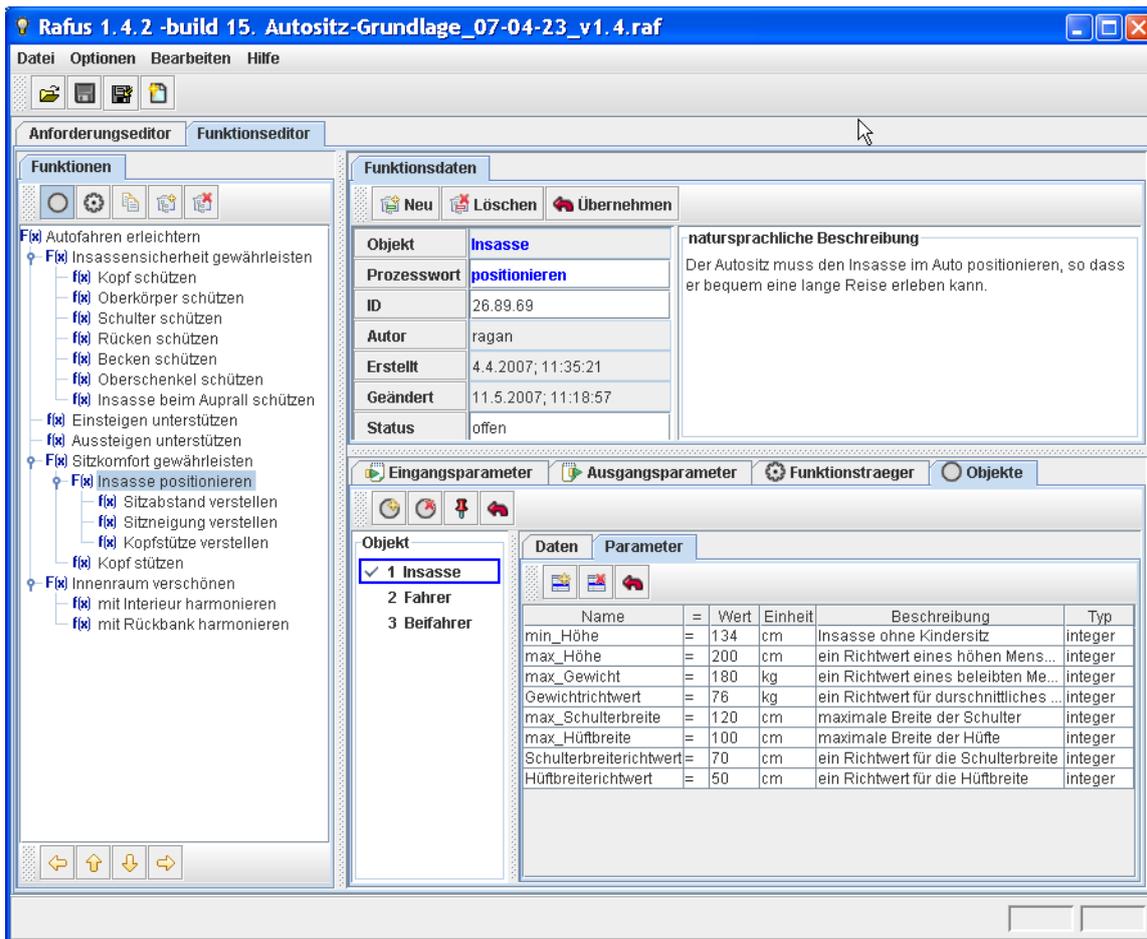


Bild 6-9: Basis-Funktionsmodell eines Autositzes

Anwendung

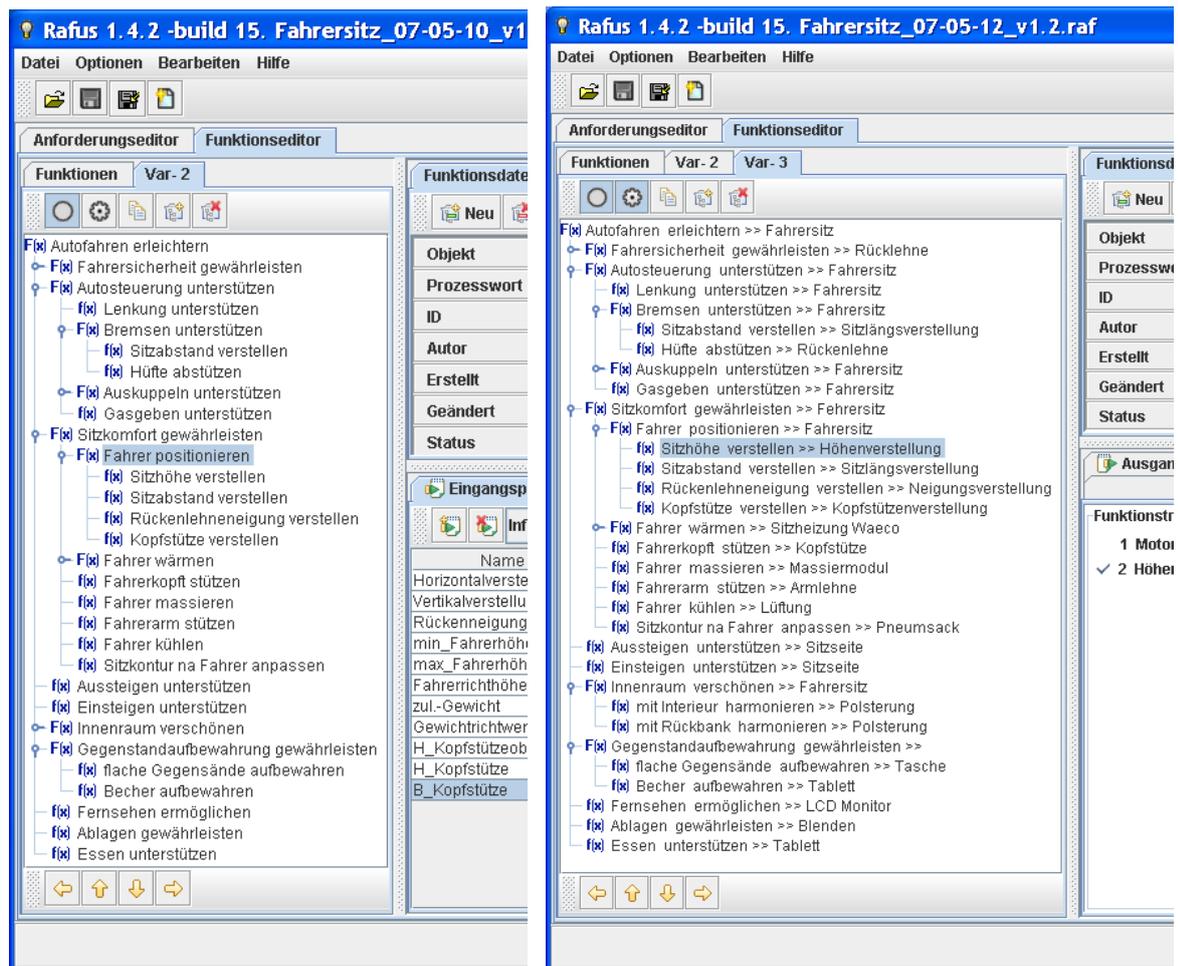
Der Ausgangspunkt in diesem Beispiel ist der Auftrag zur Entwicklung eines Sitzes für einen Autofahrer entsprechend den OEM-Anforderungen. Bei der Analyse der Kundenanforderungen wurden Besonderheiten im Bezug auf den Insassenschutz und das Komfortgefühl festgestellt. Weiterhin mussten Ergebnisse von Simulationen und Testläufen bei der Entwicklung stark berücksichtigt werden. Da die Basisfunktionen des Fahrersitzes stets unverändert bleiben, erlaubt dies, das vorhandene rechnerinterne Autositzmodell an neue Anforderungen schnell anzupassen und – darauf aufbauend – die Zusatzfunktionen und Funktionsträger zu variieren. Dieser Ansatz hilft, eine optimale Lösung schon in der Konzeptphase effektiv einzugrenzen.

Die rechnerintegrierte Konzipierung der Anpassung mittels des Softwarewerkzeugs beginnt mit der Überprüfung des vorhandenen rechnerinternen Modells vom Autositz anhand der vorgegebenen Auftragsdaten. Dabei wurden Anforderungsstruktur, Beschreibungen,

Formulierungen und Parameter untersucht und um auftragspezifische Inhalte bezüglich des Airbags, der Antriebe und der Sensorik ergänzt.

Weiterhin wurde das Funktionsmodell anhand von Anforderungen und Auftragsdaten überprüft. Hierbei fand eine Untersuchung der Funktionsstruktur statt, in der die Objekte, die Funktionsparameter, die Funktionsträger sowie Funktionsträgerparameter geprüft und erstmalig ergänzt wurden.

In virtuellen und physischen Simulationen sowie Tests wurde das Sitzverhalten im Crash mittels Messungen dokumentiert, sowie das Komfortgefühl mit verschiedenen Formen von Lehnen und Stützen, Bezugstoffen und Schaums erprobt. Die Ergebnisse der Erprobungen wurden jedes Mal in der rechnerintegrierten Konzipierung berücksichtigt. Diese führten – entsprechend der Kundenanforderungen – in wenigen iterativen Schritten mit Hilfe einer Variierung der Funktionsstruktur (Bild 6-10 a) und der Funktionsträger zur effektiven Eingrenzung der Lösung (Bild 6-10 b).



a) Angepasstes Funktionsmodell b) Um Funktionsträger erweitertes Modell

Bild 6-10: Funktionsmodelle eines Fahrersitzes

Die Evaluierung der Methode anhand des Autositzmodells und unter Anwendung des Prototyps demonstriert, dass ein auftragspezifisches rechnerinternes Konzeptmodell durch die Erweiterung des einmalig erarbeiteten „Basis“-Funktionsmodells um gewünschte Kundenfunktionen und durch dessen frühzeitige Überprüfung auf Anforderungskonformität effizienter mit der entwickelten Benutzungsoberfläche ausgearbeitet werden kann, als dieses mit gegenwärtig verfügbaren Mitteln erzielt werden könnte. Zwar muss man zuerst die Basis erarbeiten, aber der Aufwand bei derartig methodisch formuliertem Vorgehen ist gering verglichen mit den sich daraus ergebenden Einsparungen.

Der Zeitaufwand einer auftragspezifischen Entwicklung, und dadurch auch die Kosten, kann gegenüber der derzeitig eingesetzten Arbeitsweise stark reduziert werden. Ferner zeigt dieses Beispiel, dass Anwendung und Zusammenspiel des Vorgehens, des Softwarewerkzeugs und eines „minimalen“ Grundlagemodells zur rechnerintegrierten Konzipierung in einer Situation, in der Lösungen für die Kundenanforderungen erst untersucht bzw. erforscht werden müssen, den Änderungsaufwand reduzieren, die Lösung treffsicherer gestalten, als dieses ohne eine methodische Harmonisierung zwischen diesen drei Elementen möglich wäre. Dadurch werden sowohl die Qualität als auch die Effektivität einer derart gestalteten Entwicklungsarbeit signifikant gesteigert.

6.4 Fallbeispiel einer Verkaufstheke

Dieses Praxisbeispiel präsentiert die Anwendung der rechnerintegrierten Methode bei der Auftragsentwicklung einer Kühltheke ausgehend von dem Funktionsmodell einer Verkaufstheke. Ziel der Evaluierung war, ein „*maximales*“ rechnerinternes Modell einer Verkaufstheke so zu gestalten, dass es als umfangreiche Basis zur Auftragsentwicklungen von diversen Kühl-, Präsentations-, Verkostungstheken sowie von Theken zur Essenausgaben benutzt werden kann. Ein Kühlthekenbeispiel ist im Bild 6-11 dargestellt.



Bild 6-11: Kühltheke zur Aufbewahrung und Verkauf von Sandwiches⁶

Vorbereitung

Zuerst wurde die Komplexität der Erstellung eines rechnerinternen Modells für die Verkaufstheke untersucht. Man stellte dabei fest, dass ein umfangreiches Produktmodell mit allen gegenwärtig realisierbaren Funktionen als eine Entwicklungsbasis für Verkaufstheken erstellt werden kann und dass die Nutzung einer rechnerintegrierten Methode wirtschaftlich sein wird. Die Höhe der Kostenersparnisse lässt sich unter Berücksichtigung der Wiederverwendung des Produktmodells erst nach der Realisierung einiger Aufträge feststellen.

⁶ Mit freundlicher Genehmigung der Firma MOBIDEKO sp. z. o. o., Świerkówki, (Polen)

Schließlich wurde gemäß der Vorgehensweise ein rechnerinternes Produktmodell erstellt, in dem alle relevanten Anforderungskategorien aus mehreren vorhandenen Auftragsunterlagen, wie z. B. „*Design*“, „*Geometrische Anforderungen*“ oder „*Recycling*“, entnommen und dann in dem Softwarewerkzeug erfasst wurden. Da die Kundenanforderungen entscheidende Bedeutung haben, sind sie in einer extra Kategorie gesammelt und gegliedert worden. Dabei stellte man fest, dass das Design für den Kunden eine höhere Bedeutung als die Funktionalitäten hat.

Während der weiteren Analyse des Produkts wurden in erster Linie die Anforderungen identifiziert, die der Kunde bestimmt, und dann solche, über die der Entwickler entscheidet. Bei den Kundenanforderungen wurden die wichtigsten funktionalen Anforderungen eingegrenzt, die über die Bestimmung des Geräts entscheiden, und durch Stichwörter, wie „*Aufbewahrung*“, „*Lagerung*“ oder „*Präsentation*“, erfasst. Außerdem erfolgte die Erfassung solcher Anforderungen, die aus der Art des Lebensmittels hervorgehen und deshalb auf die Funktionalitäten der Verkaufstheke großen Einfluss haben. Nachfolgend wurden die Design-Anforderungen definiert, wie z. B. Farbe und Form des Korpus, des Aufsatzes und der Scharniere. Abschließend legte man die technischen Anforderungen für Steuerung, Verdampfer, Kompressor, Geräuschentwicklung und Material fest.

Im weiteren Verlauf der Erstellung des Basismodells wurden aus den funktionellen Anforderungen die Gesamtfunktion und die nutzungsbestimmenden Funktionen, wie z. B. „*Lebensmittel bis zum Verkauf aufbewahren*“, „*Lebensmittel präsentieren*“ und „*Lebensmittel lagern*“, mit Hilfe des Softwarewerkzeugs extrahiert sowie auf Unterfunktionen zerlegt und durch andere Funktionen, wie z. B. „*Lebensmittel kühl halten*“ oder „*Kondenswasser abführen*“, vervollständigt. Darüber hinaus erfolgten die Erstellung der textuellen Beschreibungen und die Definition der Eingangs- und Ausgangsparameter. Im letzten Schritt fand die Erweiterung der Funktionen um mögliche Funktionsträger, wie „*Steuergerät*“ oder „*Isolierung*“, und deren parametrische Beschreibung statt. Das Funktionsmodell einer Verkaufstheke ist im Bild 6-12 zu sehen.

Anwendung

Die Anwendung des rechnerinternen Produktmodells wird durch einen Kundenauftrag für eine Kühltheke initiiert. Da die bestellten Funktionen des Geräts innerhalb der vorher modellierten erwartet werden, erlaubt dies, ein vorhandenes rechnerinternes Produktmodell an neue Anforderungen schnell anzupassen und eine optimale Lösung effektiv einzugrenzen.

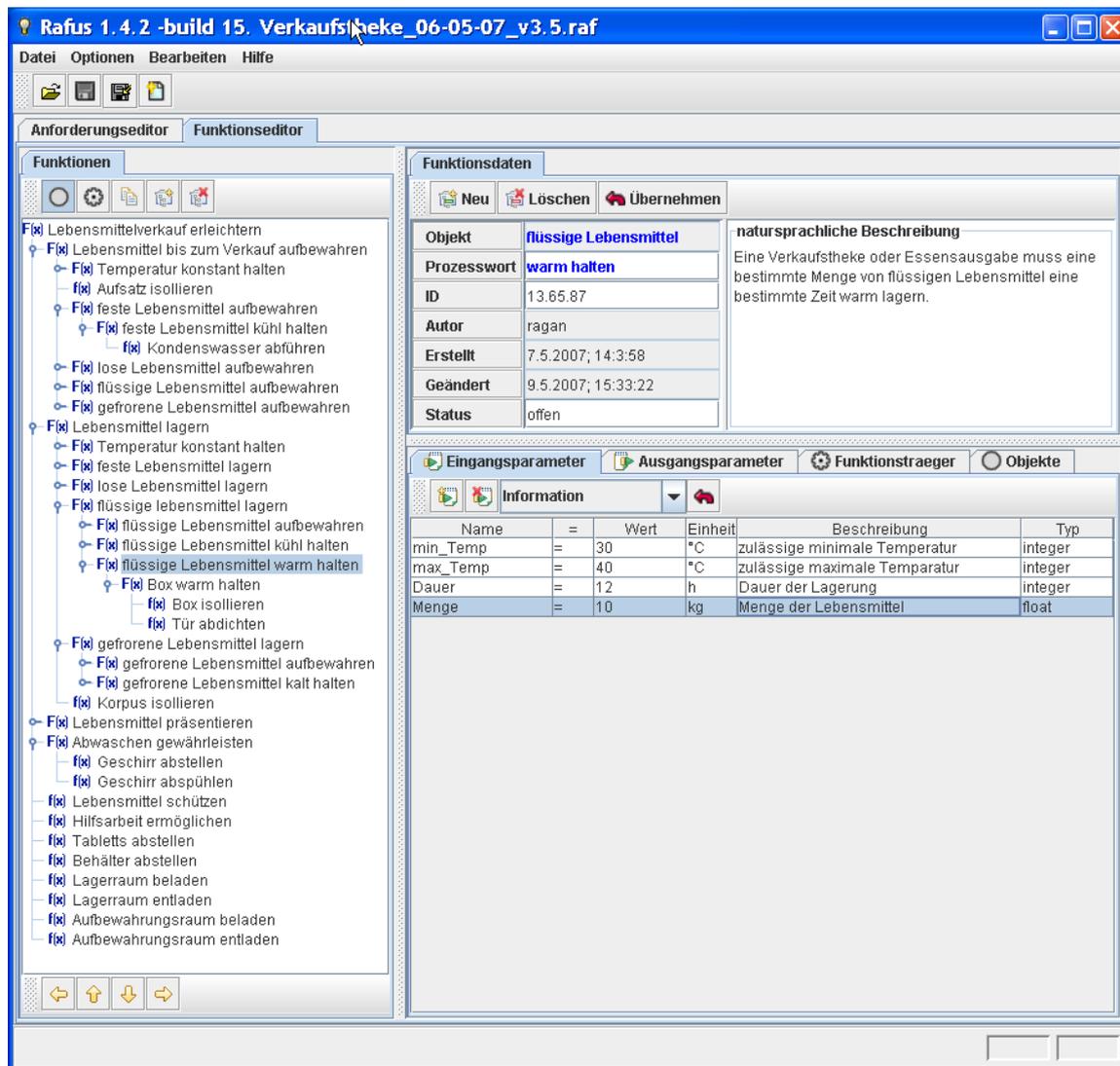
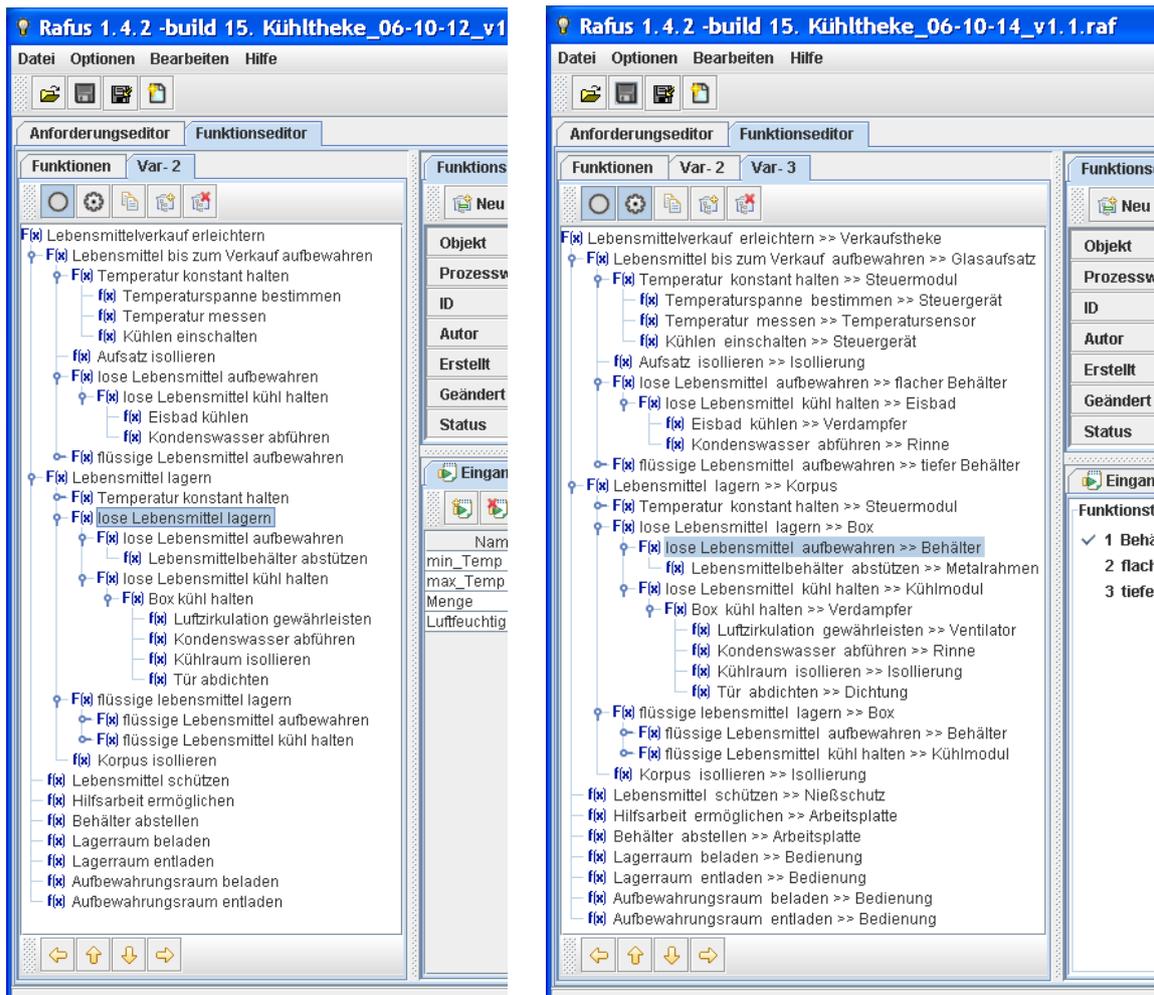


Bild 6-12: Funktionsmodell einer Verkaufstheke

Die rechnerintegrierte Konzipierung einer Kühltheke mit Hilfe des vorher beschriebenen Softwarewerkzeugs beginnt mit der Überprüfung des Basismodells einer Verkaufstheke anhand der Kundendaten. Diese bestehen sowohl aus den funktionalen und geometrischen Anforderungen als auch aus der Bestimmung der Anzahl der Kühlboxen, Türen und Steuerungspanels. All diese Daten werden auch unter den Designaspekten analysiert. Bei der Untersuchung der Anforderungsstruktur wurden die nicht relevanten Inhalte (z. B. „warm halten“ oder „präsentieren von Lebensmitteln“) entfernt.

Das Funktionsmodell wurde – basierend auf den Anforderungen und Auftragsdaten – in einem weiteren Schritt überprüft. Dabei wurden die Funktionen, die der Kunde nicht bestellt hat, aus dem rechnerinternen Modell der Verkaufstheke entfernt, so dass nur alle für eine Kühltheke relevanten übrig geblieben sind. Weiterhin wurden Funktionsparameter geprüft und ergänzt. Als Ergebnis dieses Schrittes entstand

ein auftragspezifisches Funktionsmodell einer Kühltheke (Bild 6-13 a). Dieses wird weiter dadurch konkretisiert, indem zu den Funktionen anhand der Anforderungen die Funktionsträger, wie z. B. Ventilatoren, Verdampfer und Dichtungen, zugewiesen werden (Bild 6-13 b). Diese werden zuerst teils konkret teils wegen notwendiger weiterer Abstimmung allgemein definiert. Die Abstimmungsergebnisse und die Variierung der Funktionsträger führten in wenigen iterativen Schritten sehr effizient zu abschließender Konkretisierung der Lösung auf der Konzeptionsebene.



a) Angepasstes Funktionsmodell b) Um Funktionsträger erweitertes Modell

Bild 6-13: Funktionsmodelle einer Kühltheke

Die exemplarische Modellierung der Kühltheke mittels des Systemprototyps zeigt, dass für die rechnerintegrierte Konzipierung von Produkten, die zu einem Produkttyp zusammengefasst werden können, ein umfassendes rechnerinternes Produktmodell erfolgreich anwendbar ist. Dieses muss einmalig erstellt werden. Der erforderliche Aufwand dafür ist aber gering im Vergleich zur Effizienzsteigerung, die einerseits auf der neuartigen rechnerintegrierten Arbeitsweise beruht und andererseits aus der langfristigen Nutzung des Produktmodells entsteht.

In dem Praxisbeispiel führten der Einsatz des erarbeiteten Vorgehens, des Softwareprototyps und die Reduzierung des „maximalen“ Produktmodells zum einen zu einer signifikanten Aufwandsminderung bei der Bestimmung des auftragsspezifischen Konzepts gegenüber den zurzeit eingesetzten Mitteln. Zum anderen kann die Lösung, im Vergleich zu gegenwärtig verfügbaren Werkzeugen und Methoden, durch die Anwendung des rechnerintegrierten Ansatzes effizienter ausgewählt und eine deutliche Effektivitäts- und Qualitätssteigerung der Entwicklungsarbeit erzielt werden. Bei der Erreichung der Verbesserungen spielte die methodische Auslegung der Benutzungsoberfläche des Softwareprototyps eine entscheidende Rolle.

7 Zusammenfassung

Da die kleinen und mittelständischen Unternehmen (KMU) die meisten Arbeitsplätze bieten und die Mehrheit der Unternehmen in der produzierenden Industrie darstellen, bilden sie eine tragende Säule der Wirtschaft und müssen eine rechnerintegrierte Unterstützung bekommen, die ihnen erlaubt, die zentralen betriebswirtschaftlichen Kriterien, wie z. B. Kosten, Entwicklungszeit und Qualität, ihrer Produkte zu optimieren und neue konkurrenzfähige Produkte zu entwickeln.

Basierend auf der Analyse der wissenschaftlichen Ansätze aus dem Bereich der Produktentwicklung und der derzeitigen Situation auf diesem Gebiet in KMU, ist die Notwendigkeit der Erarbeitung einer Methode zur rechnerintegrierten Konzipierung abgeleitet worden. Der Anwendungsbereich des hier beschriebenen Ansatzes liegt in der methodischen Unterstützung der auftragbasierten Produktentwicklung in KMU mit dem Schwerpunkt Neu- sowie Anpassungskonzipierung bei der Einzel- und Kleinserienfertigung sowohl von Maschinen als auch von Gebrauchsgütern.

In der vorliegenden Arbeit wurde ein umfassendes Konzept erarbeitet, das alle notwendigen Aktivitäten für den Aufbau eines Produktmodells, für die Konzipierung eines neuen Produkts und einer Produktpassung sowie die Einführung und Pflege der rechnerintegrierten Methode beschreibt. Weiterhin wurde für diesen Ansatz ein Softwarewerkzeuge konzipiert und prototypisch realisiert.

Besonderes Augenmerk wurde bei der Entwicklung des Ansatzes sowohl auf die von KMU bevorzugte einfache Gestaltung der Vorgehensweise und des dazugehörigen Softwaresystems sowie deren Integration als auch auf ein möglichst gutes Verhältnis zwischen Aufwand und Nutzen gelegt. Um diese Anforderungen zu erfüllen, wurden die Abstraktionsebenen bei der Erarbeitung und Darstellung der Methode strikt getrennt und von unnötigem Informationsballast befreit. So kann die neu entwickelte rechnerintegrierte Methode kompakt auf einer generischen Objektebene formuliert werden.

Durch die Erarbeitung des Vorgehens für den Aufbau eines Produktmodells wurde ein Grundstein für die rechnerintegrierte Konzipierung von mittelgroßen Produkten gelegt. Darauf basierend unterstützt diese Vorgehensweise, die Kundenanforderungen mit Hilfe eines Softwarewerkzeugs zu erfassen, zu kategorisieren und präzise zu definieren. Ausgehend von dem Anforderungsmodell wird der Entwickler dann in die Lage versetzt, effizient mehrere Funktionsmodelle mit Funktionsträgern zu erstellen und auch zu überprüfen, welche von

den Funktionsträgern die Anforderungen – insbesondere die funktionale Anforderungen – am wahrscheinlichsten erfüllen. Dadurch kann der Produktentwickler die gesuchte Lösung effizient eingrenzen.

Die prototypische Systemimplementierung nutzend, konnte der vorgestellte rechnerintegrierte Ansatz anhand ausgewählter Produktbeispiele evaluiert werden. Die Ergebnisse bestätigten die erwartete wesentliche Effektivitätssteigerung der Entwicklungsarbeit als auch die Reduzierung des Änderungsaufwandes und die Qualitätssteigerung der entwickelten Produkte. Die Systemeinführung und -bedienung mit entsprechend methodisch und benutzerfreundlich ausgelegter Benutzungsoberfläche bedürfen bei ausgebildeten Ingenieuren keiner Schulung.

Die zukünftige Aufgabe wird darin bestehen, den erarbeiteten Ansatz um eine Schlussfolgerungskomponente zur automatisierten Generierung der Funktionsmodelle mit Zuweisung der Funktionsträger sowie um eine Überprüfung der Konsistenz der Modelle zu erweitern. Eine weiterführende Aufgabe wird sein, Integrationskonzepte mit den CAx-Systemen der Konstruktionsphase zu erarbeiten. Diese Systeme können benötigte Informationen zur automatisierten Optimierung der Funktionsstruktur in der Konzeptionsphase liefern. Als Basis dafür kann die standardisierte CAD Services-Schnittstelle /54/ angewendet werden.

8 Literaturverzeichnis

- /1/ Spur, G.; Krause, F.-L.: Das Virtuelle Produkt: Management der CAD-Technik. Carl Hanser Verlag, München, 1997.
- /2/ Grabowski, H.; Geiger, K.: Neue Wege zur Produktentwicklung. Raabe, Stuttgart, 1997.
- /3/ Huterer, P.: Reflexive Dialoge und Denkbausteine für die methodische Produktentwicklung. München, 2005.
- /4/ Ehrlenspiel, K.: Integrierte Produktentwicklung. Carl Hanser Verlag, München, 1995.
- /5/ Fuchs, D. K.: Konstruktionsprinzipien für die Problemanalyse in der Produktentwicklung. TU-München Diss., 2005.
- /6/ Pahl, G.; Beitz, W.: Konstruktionslehre. Springer-Verlag, Berlin, 1997.
- /7/ Beitz, W.; u. a.: Strukturen rechnerunterstützter Konstruktionsprozesse. Teilprojekt B2 des Sonderforschungsbereiches SFB 203: Rechnerunterstützte Konstruktionsmodelle im Maschinenwesen. Abschlußbericht. TU-Berlin, Berlin, 1992.
- /8/ Roth, K.: Konstruktionslehre. Springer-Verlag, Berlin, 2000.
- /9/ VDI-Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. VDI-Verlag, Düsseldorf, 1993.
- /10/ VDI-Richtlinie 2222: Methodisches Entwickeln von Lösungsprinzipien. VDI-Verlag, Düsseldorf, 1993.
- /11/ Baumann, R.; Leemhuis, H.; Kaufmann, U.; Kühn, T.; Ragan, Z.; Swoboda, F.: Funktionsorientiertes Entwerfen. In: Leitprojekt integrierte Virtuelle Produktentstehung – Abschlußbericht, Fraunhofer IRB Verlag, Stuttgart, 2002.
- /12/ VDI-Richtlinie 2206: Entwicklungsmethodik für Mechatronische Systeme. Beuth Verlag, Berlin, 2004.
- /13/ Lindemann, U.: Methodische Entwicklung technischer Produkte. Springer-Verlag, Berlin [u.a.], 2007.
- /14/ Seidel, M.: Methodische Produktplanung. Univ. Karlsruhe Diss. Universitätsverlag Karlsruhe, Karlsruhe, 2005.
- /15/ Schwankl, L.: Analyse und Dokumentation in den frühen Phasen der Produktentwicklung. TU-München Diss., Verlag Dr. Hut, München, 2002.
- /16/ Kruse, P.: Anforderungen in der Systementwicklung-Erfassung, Aufbereitung und Bereitstellung von Anforderungen in interdisziplinären Entwicklungsprojekten. TU-Clausthal Diss., VDI-Verlag, Düsseldorf, 1996.
- /17/ Ahrens, G.: Das Erfassen und Handhaben von Produkthanforderungen. Dissertation TU-Berlin, Schriftreihe Konstruktionstechnik, Berlin, 2000.
- /18/ Gebauer, M.: Kooperative Produktentwicklung auf der Basis verteilter Anforderungen. Univ. Karlsruhe Diss., Shaker Verlag, Aachen, 2001.
- /19/ Jung, C.: Anforderungsklärun g in interdisziplinärer Entwicklungsumgebung. TU-München Diss. Verlag Dr. Hut, München, 2006.

- /20/ Jörg, M. A. J.: Ein Beitrag zur ganzheitlichen Erfassung und Integration von Produktanforderungen mit Hilfe linguistischer Methoden. Univ. Karlsruhe Diss., Shaker Verlag, Aachen, 2005.
- /21/ Rupp, Ch.: Requirements-Engineering und –Management. Carl Hanser Verlag, München, 2002.
- /22/ Shaler, S.; Mellor, S.J: Object-oriented system analysis. Yourdon Press, Englewood Cliffs, NJ, 1988.
- /23/ Raumbach, J.: Objektorientiertes modellieren und Entwerfen. Hanser-Verlag, München, 1993.
- /24/ Coad, P.; Yourdon, E.; Objektorientierte Analyse, 2 Auflage, Prentice Hall, München, 1996.
- /25/ Booch, G.: Objektorientierte Analyse und Design, Addison-Wesley, München, 2000.
- /26/ Jacobson, I.: Object oriented software engineering. ACM Press, Wokingham, 1992.
- /27/ Telelogic Inc: Telelogic DOORS, <http://www.telelogic.com/products/doorsers/doors/index.cfm>, 2006.
- /28/ IBM: Rational RequisitePro, <http://www-306.ibm.com/software/awdtools/reqpro/>, 2007.
- /29/ Igatech Systems: RDT, <http://www.igatech.com/rdt/index.html>, 2006.
- /30/ Electronic Data Systems EDS: Teamcenter Requirements http://www.ugs.com/products/teamcenter/sol_prod/requirements/, 2006.
- /31/ CADsys, FOD, <http://fod.cadsys.de>, 2007.
- /32/ Huang, M.: Funktionsmodellierung und Lösungsfindung mechatronischer Produkte. Univ. Karlsruhe Diss., Shaker Verlag, Aachen, 2002.
- /33/ Gehrke, M.: Entwurf mechatronischer Systeme auf Basis von Funktionshierarchien und Systemstrukturen. Univ. Paderborn Diss., Paderborn, 2005.
- /34/ Leemhuis, H.: Funktionsgetriebene Konstruktion als Grundlage verbesserter Produktentwicklung. TU-Berlin Diss., Berlin, 2005.
- /35/ EU-Kommission: Die neue KMU-Definition. EU Amt für Veröffentlichungen, Brüssel, 2006.
- /36/ IfM Bonn: Institut für Mittelstandforschung. <http://www.ifm-bonn.de/>, 2007.
- /37/ EU Kommission: KMU-Definition. http://ec.europa.eu/enterprise/enterprise_policy/sme_definition/index_de.htm, 2007.
- /38/ Engelmann, F.: Produktplanung und Produktentwicklung in kleinen und mittleren Unternehmen. Univ. Magdeburg Diss., Shaker Verlag, Aachen, 1999.
- /39/ Heßling, T.: Einführung der Integrierten Produktpolitik in kleinen und mittelständischen Unternehmen. TU-München Diss., Verlag Dr. Hut, München, 2006.
- /40/ Baumann R.; Kaufmann U.; Kühn T.; Leemhuis H.; Ragan Z.; Swoboda F.: Function Oriented Product Modelling Based on Feature Technology and Integrated Constraint Management. Product Data Technology Europe 2002 11th Symposium 7-9 May 2002 Turin, Italien. in PDT Europe 2002 Executive summary, Turin, 2002.

- /41/ Baumann, R.; Kaufmann, U.; Krause, F.-L.; Kühn, T.; Leemhuis, H.; Ragan, Z.; Swoboda, F.: Computer Aided Conceptual Design. The 36th CIRP-International Seminar on Manufacturing Systems, 03-05 June 2003, Saarbruecken, 2003.
- /42/ Baumann, R.; Grahl, W.; Kuehn, T.; Ragan, Z.: Konzeptionelle Entwicklung, ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb Nr. 97 (2003) 6, Berlin, 2003, str. 283-286, 2003.
- /43/ CADsys GmbH: FOD. http://www.cadsys.de/content/index.php?option=com_content&task=view&id=67&Itemid=124, 2007.
- /44/ Jansen H.; Ragan Z.; Weiss Z.: Systemowe Wspomaganie Projektowania, IV Wroclawskie Sympozjum "Automatyzacja produkcji" Tom II. „Referaty plakatywne”, Wrocław, 11-12.12.2003.
- /45/ Weiss, Z.; Oleszak, M.; Ragan, Z.; Zacharek, R.: Variant Design of a Water Pump. In: 1st International Conference “Virtual Design and Automation”, Publishing House of Poznań University of Technology, Poznań - Poland, 2004.
- /46/ Frad A.; Hermann Ch.; Krause F.-L.; Ragan Z.; Weiss Z.: New approach for the end of life oriented product conceptual design 2nd International Conference “Virtual Design and Automation” Poznań – Poland, 28-29 November 2005.
- /47/ Krause, F.-L.; Hayka, H.; Langenberg, D.: Actual und Future Needs for Collaboration Design Reviews. In: Proceedings of the International PACE Forum: Collaborative Visualisation, Darmstadt, S. 211-225, 15-17 Nov. 2006.
- /48/ Patzak, G.: Systemtechnik – Planung komplexer innovativer Systeme. Grundlagen, Methoden, Techniken. Springer-Verlag, Berlin, 1982.
- /49/ Weigt, M.: Komplexität als Herausforderung and die Entwicklung von Ingenieurmethoden – Problemanalyse und Lösungsansatz. In: H. Meerkamm (Hrsg.). Design for X. Beiträge zum 17. Symposium (199-210), Neukirchen, 12.-13. Oktober 2006.
- /50/ Oestereich, B.: Analyse und Design mit UML 2. Oldenbourg Verlag, München, 2005.
- /51/ Chlebek, P.: User Interface-Orientierte Softwarearchitektur. Friedrich Vieweg & Sohn Verlag, Wiesbaden, 2006.
- /52/ Sun Microsystems Inc.: Java Technology, <http://java.sun.com>, 2007.
- /53/ Sun Microsystems Inc.: JDK 6 Swing, <http://java.sun.com/javase/6/docs/technotes/guides/swing/index.html>, 2008.
- /54/ Object Management Group: CAD Services V1.1 Revision Task Force. <http://mantis.omg.org/mfgcadv1-1rtf.htm>, 2007

