TECHNISCHE UNIVERSITÄT MÜNCHEN

Institut für Informatik, Software & Systems Engineering

Wissen, wann sich Wiederverwendung lohnt:

e-FACTS, ein Modell zur einfachen ökonomischen Bewertung von Software-Produktlinien in der Automobilindustrie

Dirk-Michael Robert Alfred Reich

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Gudrun J. Klinker, Ph.D.

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Manfred Broy
2. Univ.-Prof. Dr. Klaus Pohl, Universität Duisburg-Essen

Die Dissertation wurde am 06.03.2012 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 21.09.2012 angenommen.

Zusammenfassung

Die Automobilindustrie befindet sich in einem faszinierenden Wandel. Software und die darauf aufbauenden Funktionen spielen eine zunehmend bedeutende Rolle. Bereits heute wird ein Großteil der Innovationen im Auto durch Software erst ermöglicht. In dem traditionell ingenieurgeprägten automobilen Umfeld führt dieser Wandel zu einigen Herausforderungen. Entsprechende IT-Kapazitäten sind vielfach nicht ausreichend vorhanden. Und dabei stellt die Erstellung von Software selbst die traditionellen Softwareunternehmen immer noch vor einige Probleme. Projekte sind oft verzögert und teurer als geplant. Eine vieldiskutierte Lösung ist eine stärkere Wiederverwendung von bekannten Komponenten. Im Hardwarebereich haben OEMs dieses Thema, z.B. durch Plattformkonzepte oder Module, erfolgreich gemeistert. Softwareunternehmen haben mit dem Konzept der Software-Produktlinien (SWPL) erfolgreich nachgezogen. Folgerichtig setzen auch einige Unternehmen in der Automobilindustrie SWPL versuchsweise ein. SWPL stellen dabei ein relativ neues Entwicklungsparadigma da, das mit einer Orientierung auf strategische Wiederverwendung von zentralen Komponenten, sogenannten Core Assets, die Entwicklungszeit zu verkürzen und dabei gleichzeitig die Qualität zu erhöhen verspricht.

Da die Einführung einer SWPL mit erheblichen Investitionen und zeitlichen Anpassungen verbunden ist, ist eine ökonomische Fundierung einer solchen strategischen Entscheidung notwendig. Die vorhandenen ökonomischen Modelle sind allerdings entweder auf einen spezifischen Anwendungsfall zugeschnitten. Oder die Modelle sind stark vereinfacht, z.B. durch die Verwendung von Standardparametern, über die in der Literatur keine Einigkeit besteht. Daher ist eine direkte Übernahme der bestehenden Modelle für die Autoindustrie nicht möglich. Das im Rahmen dieser Arbeit für die Automobilindustrie entwickelte e-FACTS Modell schließt diese Lücke. Es handelt sich um ein generisch anzuwendendes, ökonomisches Modell. Der Name steht für economic Feature Point Analysis with compatible timing for Software Product Lines. Die Zielsetzung war ein Ansatz, der einfach, mit geringem Zeitaufwand und zu einem sehr frühen Zeitpunkt in der SWPL- Entscheidung bei Automobilunternehmen anzuwenden ist. Als Ergebnis liefert e-FACTS fundierte finanzielle Daten, um die Entscheidung über eine SWPL ökonomisch sinnvoll treffen zu können.

Die Arbeit umfasst zunächst eine umfangreiche Vorstellung der wichtigsten Begriffe und Methoden und eine kritische Auseinandesetzung mit den bereits erwähnten bestehenden

ökonomischen Modellen, bevor das e-FACTS Modell im Detail vorgestellt und anhand eines durchgängigen Fallbeispiels erklärt wird. Das Modell bietet ein mehrstufiges Vorgehen an mit dem Ziel, den Aufwand durch möglichst frühzeitigen Ausschluss von ungeeigneten Anwendungen möglichst gering zu halten. Dazu erfolgt im ersten Schritt die Beantwortung eines grundlegenden Fragebogens, der das Vorhandensein von wesentlichen Erfolgskriterien überprüft. Im zweiten Schritt werden weitere ungeeignete Anwendungen mittels einer einfachen Analyse, basierend auf den bekannten ökonomischen Modellen, identifiziert.

Der dritte Schritt ist die eigentliche Detailanalyse des e-FACTS Modells. Diese verwendet eine detaillierte Aufwandsbestimmung mittels einer auf Functions Points basierenden Schätzmethode für Software. Darauf aufbauend erfolgt die Optmierung der Asset Base, dem Kernstück der SWPL, also dem Teil, der strategisch wiederverwendet werden soll. e-FACTS fokussiert dabei stärker auf ein ökonomisch optimiertes Ergebnis. Die tatsächliche technische Realisierbarkeit (Variabilität) ist zusätzlich zu prüfen.

Im Rahmen eines detaillierten Business Cases erfolgt abschließend die Bestimmung der ökonomischen Folgen einer SWPL Einführung, inkl. einer klaren Empfehlung für oder gegen die Umsetzung des Plans. Die hier ermittelten Werte können von den jeweiligen Unternehmen direkt für die eigentliche Entscheidungsfindung herangezogen werden. Das e-FACTS Modell ist so ausgelegt, dass auf Basis der hier vorliegenden Arbeit ein Unternehmen aus der Automobilindustrie in der Lage ist (evtl. unter Einbeziehung weiterführender Literatur, auf die jeweils verwiesen wird) alle notwendigen Schritte eingenständig durchzuführen.

Die reale Anwendbarkeit des e-FACTS Modells wurde anhand von zwei Fallstudien bei Siemens VDO (heute VDO) und dem Deutschen Zentrum für Luft- und Raumfahrt (DLR) Bereich für Verkehrssystemtechnik erfolgreich überprüft. Die Ergebnisse dieser Fallstudien werden am Ende der Arbeit vorgestellt. Zusätzlich finden sich dort umfangreiche Sensitivitätsanalysen, die der Überprüfung der Stabilität der Ergebnisse dienen.

Das e-FACTS Modell ist für Automobilunternehmen ausgelegt, die die Einführung einer SWPL planen, die damit verbundene ökonomische Unsicherheit jedoch verringern möchten. Es ermöglicht mit einem überschaubaren zeitlichen und personellen Aufwand, die Informationsbasis für eine qualifizierte Entscheidung hinsichtlich der SWPL-Einführung zu schaffen.

Danksagung

Auch wenn eine Doktorarbeit natürlich eine eigenständige Arbeit darstellt, wäre die Fertigstellung ohne die Unterstützung einer großen Zahl an Personen nicht möglich. Daher möchte ich die Chance nutzen, mich an dieser Stelle explizit zu bedanken:

An erster Stelle danke ich meinem Doktorvater Prof. Dr. Manfred Broy für die großzügige wissenschaftliche Betreuung meines Themas. Insbesondere die äußerst konstruktive inhaltliche Begleitung mit zahlreichen Kommentare und Verbesserungsvorschlägen hat maßgeblich zum Gelingen der Arbeit beigetragen. Ebensolcher Dank gebührt Prof. Dr. Klaus Pohl für die Übernahme der Rolle des Zweitgutachters dieser Arbeit.

Bedanken möchte ich mich ebenfalls bei den Kollegen vom Lehrstuhl für Software and Systems Engineering von Prof. Broy an der TU München. Hier wurde ich, obwohl externer Doktorand, äußerst freundlich aufgenommen. Ich habe mich dort immer wohl gefühlt. Dies ist insbesondere auch der unglaublich netten und hilfsbereiten Art von Marina und Silke zu verdanken.

Die beiden Fallbeispiele am Ende der Arbeit wären ohne die Unterstützung von Herrn Dr. Bocionek von VDO und Herrn Dr. Frank Köster vom DLR nicht möglich gewesen. Beide waren nicht nur für die Workshops, sondern auch für das Feedback zu den jeweiligen Kapiteln bereit, ihre Zeit zu investieren. Außerdem haben sie dankenswerterweise der Veröffentlichung der Ergebnisse im Rahmen der Arbeit zugestimmt.

McKinsey & Company hat mich in jeder Hinsicht durch den für eine solche Arbeit notwendigen Freiraum unterstützt. Ohne diese Unterstützung wäre ich das Wagnis einer Dissertation wohl nicht eingegangen.

Ebenso gilt meine tiefe Dankbarkeit Andrea, die nicht müde wurde, mit mir über einzelne Aspekte meiner Arbeit zu reden und mich immer wieder voranzubringen. Ein ganz besonderer Dank gilt meinen Eltern, die mich nicht nur tatkräftig unterstützt haben, sondern auch in schwierigen Zeiten - und die gibt es sicherlich bei (fast) jeder Dissertation - immer wieder motiviert haben. Abschließend möchte ich mich von tiefstem Herzen bei Anja für ihre Geduld und Motivation bedanken. Ohne deinen sanften Druck wäre die Arbeit heute wahrscheinlich noch immer nicht fertig.

Inhaltsverzeichnis

Teil A. Einführung und grundlegende Theorien 1

1. Einleitung 2
   1.1. Die Automobilindustrie als Erfinder der Produktlinie 2
   1.2. Motivation der Arbeit 5
   1.3. Aufgabenstellung und Zielsetzung 13
   1.4. Struktur der Arbeit 14
2. Software Produktlinien 16
   2.1. Produktlinien: Die Idee der strategischen Wiederverwendung 16
      2.1.1. Begriffsbestimmung 16
      2.1.2. Aktueller Forschungsstand 20
      2.1.3. Auswahl erfolgreicher Anwendungsbeispiele 22
   2.2. Software Produktlinien – ein bekanntes Thema im Bereich des Software Engineering 25
      2.2.1. Einleitung zu Softwareproduktlinien 25
      2.2.2. Unterscheidung zwischen SW-Produktfamilien und SW-Produktlinien 29
      2.2.3. Variabilität in Software 32
      2.2.4. Ansätze zur Bewertung der Reife von Prozessen in Unternehmen 35
         2.2.4.1. CMMI 35
         2.2.4.2. SPICE 37
         2.2.4.3. Auswirkungen auf die vorliegende Arbeit 38
      2.2.5. Entwicklungsprozesse und aktueller Forschungsstand 38
      2.2.6. Auswahl erfolgreicher Anwendungsbeispiele 49
         2.2.6.1. Philips Medical Systems (PMS) 50
         2.2.6.2. Hewlett Packard (HP) – Firmware für Drucker 50
         2.2.6.3. MAN Nutzfahrzeuge – Software als original Ersatzteil 51
      2.2.7. Abgrenzung des Produktlinienumfangs: Product Line Scoping 52
3. Embedded Systems 55
   3.1. Definition 55
   3.2. Bedeutung von Embedded Systems 57
   3.3. Eigenschaften von Embedded Systems 58
   3.4. Bedeutung der Software in Embedded Systems 60

3.5. Besonderheiten in der Automobilindustrie – Bereich PKW 61
4. Besonderheiten bei der Anwendung von Software Produktlinien in Embedded Systems 65
4.1. Hardware Fokus im Entwicklungsprozess 65
4.2. Nutzung von standardisierten Methoden und Werkzeugen 67
4.3. Beziehungen zwischen OEM und Zulieferern in der Automobilindustrie 69
4.4. Geschwindigkeit der technischen Entwicklung 71
5. Übersicht über Bewertungsmöglichkeiten des Entwicklungsaufwands bei Softwareprojekten 74
5.1. Lines of Code 79
5.2. Function Points 80
5.3. Mark II Function Points 88
5.4. COSMIC-FFP 90
5.5. Use Case Points und weitere Verfahren 94
5.6. Übersicht der heutigen Verwendung von SW Aufwandsschätzverfahren in der automobilen Praxis 96
5.7. Anpassung der Function Point Methode auf Embedded Systems 98
6. Zusammenfassung Teil A 104

Teil B. Ökonomische Betrachtung von SWPL im Automotive Bereich im Vorfeld einer Investitionsentscheidung 107

1. Einleitung 108
1.1. Motivation 108
1.2. Aufbau von Teil B 109
2. Die grundlegenden Unternehmensziele beeinflussen den möglichen Erfolg von Software Produktlinien 111
2.1. Welche Ziele verfolgen die Unternehmen? 111
2.2. Welchen Einfluss haben Software Produktlinien auf die Zielerreichung? 113
2.2.1. Operationale Ziele 114
2.2.1.1. Kosten 114
2.2.1.2. Zeit 116
2.2.1.3. Qualität 117
2.2.2. Formale Unternehmensziele 119
2.2.2.1. Gewinn 119
2.2.2.2. Wachstum 120
2.2.2.3. Rentabilität 121
2.2.2.4. Liquidität 122
2.2.2.5. Image und Innovationskraft 122

3. Analyse und Bewertung bestehender ökonomischer Modelle für Software Produktlinien 124
3.1. Grundsätzliches zu Return on Investment Ansätzen 124
3.2. COPLIMO und COCOMO II 126
3.3. Poulin's Measuring Software Reuse 134
3.4. PuLSE-Eco Ansatz 135
3.5. ABC Analyse 138
3.6. SIMPLE 139
3.7. Simulation – das ReValue Modell 143
3.8. Abschließende Betrachtung und Anforderungen an ein neues Modell 145
4. Wirkung von SWPL auf die Produktqualität 147
4.1. Wiederverwendung und Qualität 147
4.2. Einfluss der Variabilität bei SWPL auf die Qualität 148
4.3. Auswirkung einer SWPL auf ein Qualitätsmodell 149
4.4. Bedeutung der Qualität im Rahmen dieser Arbeit 153
5. Entwicklung eines umfassenden Modells zur frühzeitigen Bewertung von Produktlinienansätzen 155
5.1. Schritt 1: Fragenkatalog mit Basiskriterien 156
5.1.1. Kategorisierung der Fragen 156
5.1.1.1. Produktangebot 160
5.1.1.1.1. Anzahl der derzeitigen Produkte im Markt 160
5.1.1.1.2. Abdeckungsgrad der Produktlinie 161
5.1.1.1.3. Anzahl geplanter neuer Produkte in den nächsten 2 Jahren 163
5.1.1.1.4. Neuigkeitsgrad der relevanten Produktgruppe 164
5.1.1.1.5. Maximales Wiederverwendungspotenzial bei auf der SWPL basierenden Produkten 167
5.1.1.2. Produktentwicklung / Prozesse 168
5.1.1.2.1. Prozessreife des Unternehmens 169
5.1.1.2.2. Festlegung der Anforderungen an neue Produkte 170
5.1.1.2.3. Qualität der bestehenden Produkte 171
5.1.1.3. Organisation 174
5.1.1.3.1. Größe der SW Entwicklungsabteilung 174
5.1.1.3.2. Ausrichtung der SW Entwicklung 175
5.1.1.3.3. Anpassungsfähigkeit 176
5.1.1.3.4. Interne Unterstützung 177
5.1.1.3.5. Umsetzung einer HW Modularisierung 178

5.1.1.4. Architektur 179
5.1.1.4.1. Anteil von Standardkomponenten 180
5.1.1.4.2. Aktuelle Produktarchitektur 181
5.1.1.4.3. Integration neuer Funktionen 182
5.1.1.5. Business drivers and manners 183
5.1.1.5.1. Unternehmensziele 184
5.1.1.5.2. Verzögerung bei Produktentwicklung 185
5.1.1.5.3. Unternehmensbeteiligungen 186
5.1.1.5.4. Eigene SW Erstellung 188
5.1.2. Bewertung der Produktlinienfähigkeit der Unternehmen 189
5.2. Schritt 2: Vereinfachte Berechnung der ökonomischen Konsequenzen des Einsatzes einer Software Produktlinie 191
5.3. Das e-FACTS Modell 194
5.3.1. Beispiel Zugangssysteme im Automobil 196
5.3.2. Erhebung der notwendigen Basisdaten 201
5.3.3. Übersicht der geplanten Produktstruktur 207
5.4. Schritt 3: Einführung von Feature Points zur Komplexitätsmessung 212
5.4.1. Feature Point Zählung ohne Wiederverwendung 212
5.4.2. Berücksichtigung von Variabilität 221
5.4.2.1. Implementierungsunabhängige Variabilitätstypen 222
5.4.2.2. Realisierung von Variabilität 226
5.4.3. Feature Point Zählung mit Wiederverwendung 227
5.4.3.1. Ohne SWPL Architektur 228
5.4.3.2. Mit SWPL Architektur 233
5.4.4. Optimierung der Asset Base 236
5.4.5. Abschließende Betrachtung Schritt 3 241
5.5. Schritt 4: Business Case 241
5.6. Schritt 5: Anwendungsmöglichkeiten des Modells 250
5.6.1. Unterschiedliche Produktszenarien 251
5.6.1.1. Entwicklung SWPL auf Basis existierender Produkte 251
5.6.1.2. Integration neuer Produkte in eine bestehende SWPL 253
5.6.1.3. Evolution oder Revolution einer bestehenden SWPL 254
5.6.2. Unterschiedliche Unternehmensziele 254
5.6.2.1. Time-to-Market 254
5.6.2.2. Wachstum 256

5.6.2.3. Kundenbindung 257
6. Nachweis der Praktikabilität von e-FACTS und individuelle Anwendung 258
6.1. DOMINION des DLR 258
6.1.1. Besonderheiten des DLR 259
6.1.2. Beschreibung der Softwareumgebung DOMINION 260
6.1.2.1. Funktionsbeschreibung 260
6.1.2.2. Verwendung als Produktfamilie 261
6.1.2.3. Anwendungsmöglichkeiten des e-FACTS Modells 262
6.1.3. Detailbeschreibung der betrachteten Funktionen 263
6.1.3.1. ACC 263
6.1.3.2. Bewegungssteuerung 265
6.1.3.3. Fahrdynamikmodell 265
6.1.3.4. SeSiLa 265
6.1.3.5. Displaykonzept 266
6.1.3.6. Nebenaufgaben 266
6.1.3.7. Cabin 266
6.1.4. Aufwandsbewertung mittels FP 267
6.1.5. Hochrechnung auf das Gesamtsystem 270
6.1.6. Sensitivitätsanalyse 275
6.1.7. Abschließende Betrachtung 279
6.2. Motorsteuerungssoftware bei Siemens VDO 280
6.2.1. Projektbeschreibung 280
6.2.2. Komponente ZGA 282
6.2.3. Komponente ESR 284
6.2.4. Sensitivitätsanalyse 286
6.2.5. Abschließende Betrachtung 288
6.3. Allgemeine Bewertung und Sensitivitätsanalyse 290
7. Zusammenfassung und Ausblick 302
7.1. Zusammenfassung 302
7.2. Fazit 304
7.3. Weiterer Untersuchungsbedarf 305
Teil C. Anhang 307
Literatur 311

Abkürzungsverzeichnis

AA	:	Assessment and Assimilation
AAM	:	Adaptation Adjustment Modifier
ADC	:	Additional Development Cost
AAF	:	Adaptation Adjustment Factor
AG	:	Aktiengesellschaft
AL	:	Algorithmus
ARES	:	Architectural Reasoning for Embedded Systems
ASAM	:	Association for Standardization of Automation and Measuring Systems
AT	:	Automated Translation
BAPO	:	Business, Architecture, Process und Organisation
bspw.	:	beispielsweise
bzw.	:	beziehungsweise
CAFÈ	:	From Concept to Application in System-Family Engineering
CCS	:	Components, Connectors and Semantics
CM	:	Code Modified
CMM	:	Capability Maturity Model
CMMI	:	Capability Maturity Model Integration
COCOMO	:	Constructive Cost Model
COPLIMO	:	Constructive Product Line Investment Model
COR	:	Cost of Reuse
COSMIC	:	Common Software International Consortium
CoVAR	:	Component Selection considering Variability, Architectural Concerns, and Requirements
c.p.	:	ceteris paribus
DAH	:	Deutscher Automobilhersteller
DET	:	Data Element Type
DF	:	Data File
d.h.	:	das heißt
DM	:	Design Modified
DOCU	:	Documentation Match to Lifecycle Needs cost driver

DOR	:	Degree of Reuse
DPA	:	Domain Potential Analysis
DSI	:	Delivered Source Instructions
e-FACTS	:	economic Feature Point Analysis with compatible timing for Software Product Lines
ECU	:	Electronic Control Unit
EI	:	External Input
EIF	:	External Interface File
EM	:	Effort Multiplier
EO	:	External Output
EQ	:	External Inquery
ESAPS	:	Engineering Software Architectures, Processes, and Platforms for System Families
ESLOC	:	Equivalent Source Lines of Code
ESR	:	Einspritzungsrealisierung
evt.	:	eventuell
f.	:	folgende Seite
FAMILIES	:	Fact-based Maturity through Institutionalisation Lessons-learned and Involved Exploration of System-Family Engineering
FCM	:	Factors-Criteria-Metrics
ff.	:	fortfolgende Seiten
FP	:	Feature Point
FPs	:	Feature Points
FPA	:	Function Point Analysis
FTR	:	File Type Referenced
FUR	:	Functional User Requirements
ggfs.	:	gegebenenfalls
GPS	:	Global Positioning System
GQM	:	Goal-Questions-Metrics
HIS	:	Herstellerinitiative Software
HW	:	Hardware
IBM	:	International Business Machines
IESE	:	Institut für Experimentelles Software Engineering

IFPUG	:	International Function Point User Group
ILF	:	Internal Logical File
IM	:	Integration Modified
inkl.	:	inklusive
insb.	:	insbesondere
ISBSG	:	International Software Benchmarking Standards Group
ISO	:	International Organization for Standardization
ITEA	:	Information Technology for European Advancement
KDSI	:	Thousand Delivered Source Instructions
KSLOC	:	Thousand Source Lines of Code
LoC	:	Lines of Code
MDSD	:	Model-Driven Software Development
MECE	:	Mutually Exclusive, Collectively Exhaustive
Mk II	:	Mark II Function Point Analysis
NPV	:	Net Present Value
Nr.	:	Nummer
OEM	:	Original Equipment Manufacturer
OS	:	Operating System
PC	:	Personal Computer
PhD	:	Doctor of Philosophy
PLM	:	Product Line Mapping
PM	:	Personenmonat
PMS	:	Philips Medical Systems
PRAISE	:	Product Line Realization and Assessment in Industrial Settings
PuLSE	:	Product Line Software Engineering
RCA	:	Reuse Cost Avoidance
RCR	:	Relative Cost of Reuse
RCWR	:	Relative Cost of Writing for Reuse
RELY	:	Required Software Reliability
RET	:	Record Element Type
ROI	:	Return on Investment
RUSE	:	Development for Reuse
S.	:	Seite

SA	:	Sensitivitätsanalyse
SEI	:	Software Engineering Institute
SF	:	Scale Factor
SIMPLE	:	Structured Intuitive Model for Product Line Economics
SKH	:	Südkoreanischer Automobilhersteller
SMH	:	Société suisse de microélectronique et d'horlogerie SA (Swatch Gruppe)
SOP	:	Start of Production
SPLP	:	Software Product Line Practice
SPICE	:	Software Process Improvement and Capability Determination
SU	:	Software Understanding
SUV	:	Sport Utility Vehicle
SW	:	Software
SWPL	:	Software Produktlinie
TDEV	:	Time to Develop
TS	:	Institut für Verkehrssystemtechnik des DLR
u.a.	:	unter anderem
UML	:	Unified Modeling Language
UNFM	:	Programmer Unfamiliarity
VAF	:	Value Adjustment Factor
vgl.	:	vergleiche
z.B.	:	zum Beispiel
ZGA	:	Zusammensetzung der Gase im Auspuff

Abbildungsverzeichnis

Abbildung 1 – Produktfamilien im Rahmen der Variantenproblematik [Jung05, S. 5] 11

Abbildung 2 – Der Nutzen von Produktordnungssystemen [in Anlehnung an Wild04, S.39] 22

Abbildung 3 – PRAISE Referenzprozess [vgl. Lind02, S. 47] 40

Abbildung 4 – Übersicht der 3 wichtigen Aktivitäten bei SWPL [Clem02, S. 30] 42

Abbildung 5 – Überblick über PuLSE [Baye99, S.123] 44

Abbildung 6 – Das Software Product Line Engineering Framework [Pohl05, S.22] ... 46

Abbildung 7 – Zusammenhang zwischen Prozessreife und Softwarequalität [McKi06, S. 40] 69

Abbildung 8 – Produktlebenszyklus eines Fahrzeugs [nach Schä06, S. 21] 71

Abbildung 9 – Übersicht der zu erwartenden Schätzgenauigkeit zu verschiedenen Projektzeitpunkten [vgl. Wiec08, S. 206 und Brun01, S. 352] 78

Abbildung 10 – Die Erfassung von Functional User Requirements für Function Points [Lava09, S. 2] 81

Abbildung 11 - Ablauf der Function Point Berechnung 83

Abbildung 12 – Ablauf des Function Point Zählprozesses nach [Lava09, S. 3] 84

Abbildung 13 – Vergleich der Messungenauigkeiten und des notwendigen Zeitaufwands bei Function Point Methoden 86

Abbildung 14 – Darstellung der Anwendungsgrenzen [vgl. Cosm09a, S. 33] 91

Abbildung 15 – Der COSMIC Zählprozess nach [Lava09, S. 4] 93

Abbildung 16 – Empirische Daten zur Messungenauigkeit der FP Methodik 101

Abbildung 17 – Kostenvergleich Entwicklung mit und ohne SWPL [Clem03] 116

Abbildung 18 – Vergleich der Genauigkeit der Größen- und Kostenschätzung je Projektphase [Boehm00, S. 10, leicht angepasst] 128

Abbildung 19 – Übersicht COCOMO Modell-Familie [Smit06, S. 4] 132

Abbildung 20 – Überblick über die Grundstruktur des PuLSE-Eco Ansatzes [Schm04, S. 47] 136

Abbildung 21 – Grundsätzliches Szenario für SIMPLE Kalkulationen [vgl. Clem05, S. 5] 140

Abbildung 22 – Struktur des ReValue Simulationsprogramms [Stüt02, S. 151] 144

Abbildung 23 – Beispielhafter Graph eines SW Qualitätsmodells für die Wartbarkeit nach [Soft11].....151

Abbildung 24 – Entwicklung der Defektanzahl über die Lebenszeit von SWPL [SWPL11].....153

Abbildung 25 – Ablauf des e-FACTS Modells.....195

Abbildung 26 – Feature Modell eines automobilen Zugangssystems 198

Abbildung 27 – Bewertung der Time-to-Market [Schm03, S. 6] 255

Abbildung 28 – Übersicht der Dominion Architektur 261

Abbildung 29 – Funktionsweise der ACC Regelung.....264

Abbildung 30 – Übersicht der analysierten Funktionen auf Basis von Dominion 271

Abbildung 31 – Übersicht der Produktivitätsverteilung [vgl. auch McKi06, S. 25] ..272

Abbildung 32 – Übersicht der Produktivitätsverteilung nach Anwendungsklassen [vgl. auch McKi06, S. 25].....273

Abbildung 33 – Berechnung der ökonomischen Auswirkungen durch Dominion.....275

Abbildung 34 – Übersicht der Schnittstellen und Funktionen eines Motorsteuergeräts für Ottomotoren nach [Schä06, S. 9] 282

Abbildung 35 – Schematische Darstellung der ZGA Komponente.....282

Abbildung 36 - Schematische Darstellung der ESR Komponente 285

Abbildung 37 – Einfluss der Anzahl der Wiederverwendungen auf SWPL.....296

Abbildung 38 – Einfluss der Größe der Asset Base.....297

Abbildung 39 – Einfluss des Mehraufwands für Wiederverwendbarkeit.....299

Abbildung 40 – Einfluss des Aufwands zur tatsächlichen Wiederverwendung 300

Abbildung 41 – Basisfragebogen Teil 1.....309

Abbildung 42 – Basisfragebogen Teil 2.....310

Tabellenverzeichnis

Tabelle 1 – Commercial Software Estimation von 1974 bis 1986 nach Capers Jones [Jone07, S. 30] 79

Tabelle 2 – Function Point Werte je Funktionstyp und Komplexitätsgewicht nach IFPUG [Boeh00, S. 19] 83

Tabelle 3 – Feature Point Werte je Funktionstyp und Komplexitätsgewicht für Feature Point Berechnung 99

Tabelle 4 – Beispielhafte Produktkarte [Schm04, S. 48] 137

Tabelle 5 – Kategorie Produkte, Frage 1 161

Tabelle 6 – Kategorie Produkte, Frage 2 163

Tabelle 7 – Kategorie Produkte, Frage 3 164

Tabelle 8 – Kategorie Produkt, Frage 4 167

Tabelle 9 – Kategorie Produkt, Frage 5 168

Tabelle 10 – Kategorie Prozesse, Frage 1 170

Tabelle 11 – Kategorie Prozesse, Frage 2 171

Tabelle 12 – Kategorie Prozesse, Frage 3 173

Tabelle 13 – Kategorie Organisation, Frage 1 175

Tabelle 14 – Kategorie Organisation, Frage 2 176

Tabelle 15 – Kategorie Organisation, Frage 3 177

Tabelle 16 – Kategorie Organisation, Frage 4 178

Tabelle 17 – Kategorie Organisation, Frage 5 179

Tabelle 18 – Kategorie Architektur, Frage 1 181

Tabelle 19 – Kategorie Architektur, Frage 2 182

Tabelle 20 – Kategorie Architektur, Frage 3 183

Tabelle 21 – Kategorie Business drivers, Frage 1 185

Tabelle 22 – Kategorie Business drivers, Frage 2 186

Tabelle 23 – Kategorie Business drivers, Frage 3 188

Tabelle 24 – Kategorie Business drivers, Frage 4 189

Tabelle 25 – Basisdaten für Beispiel SWPL 207

Tabelle 26 – Produktübersicht SWPL Zugangssysteme 209

Tabelle 27 – Komponentenübersicht SWPL Zugangssysteme 210

Tabelle 28 – Bewertung der Komponenten als Domain Asset 211

Tabelle 29 – Worksheet zur Zählung der Feature Points je Komponente (gelbe Felder dienen der Dateneingabe, sonstige Daten dienen der Berechnung, L, M und H stehen für die unterschiedlichen Komplexitätsklassen) 213

Tabelle 30 – Komplexitätsklassen für EI 213

Tabelle 31 – Komplexitätsklassen für EO 214

Tabelle 32 – Komplexitätsklassen für EQ 214

Tabelle 33 – Komplexitätsklassen für AL 215

Tabelle 34 – Komplexitätsklassen für DF 216

Tabelle 35 – Komplexitätsklassen für EIF 216

Tabelle 36 – FP Zählung ohne Wiederverwendung für die GSM Komponente (ausgefülltes Worksheet, siehe auch Tabelle 29) 220

Tabelle 37 – Vollständige FPs für geplante Produkte in 5 Jahren 221

Tabelle 38 - FP Zählung Zugangssysteme als SWPL ohne Architektur (vollständig ausgefülltes Worksheet für Anwendungsbeispiel, siehe auch Tabelle 29) 232

Tabelle 39 – FP Bewertung SWPL Architektur im Anwendungsbeispiel 235

Tabelle 40 – FP Zählung Zugangssysteme als SWPL mit Architektur (vollständig ausgefülltes Worksheet für Anwendungsbeispiel, siehe auch Tabelle 29) 236

Tabelle 41 – Optimierung der Asset Base ohne Architektur Investment (gelber Felder erfordern einen individuellen Input) 239

Tabelle 42 – Optimierung der Asset Base unter Berücksichtigung der SWPL Architektur 240

Tabelle 43 – Business Case Anwendungsbeispiel Zugangssysteme (Werte in Tausend EUR, außer Prozent) 243

Tabelle 44 – Business Case mit Berücksichtigung von Optionen (Werte in Tausend EUR, außer Prozent) 250

Tabelle 45 – Feature Point Berechnung ohne Wiederverwendung (gelbe Felder zur Eingabe von Werten) 268

Tabelle 46 – Feature Point Berechnung mit Wiederverwendung (gelbe Felder zur Eingabe von Werten) 268

Tabelle 47 – FP Zählung ZGA als Asset Base (vollständig ausgefülltes Worksheet auf Basis Tabelle 29) 283

Tabelle 48 – FP Zählung ESR als Asset Base (vollständig ausgefülltes Worksheet auf Basis Tabelle 29) 285

Tabelle 49 – Übersicht der drei Szenarios für die SA 294

Tabelle 50 – FP Übersicht Zugangssysteme ohne Wiederverwendung 307

Tabelle 51 – FP Übersicht Zugangssysteme mit Wiederverwendung ohne Architektur 307

Tabelle 52 – FP Übersicht Zugangssysteme mit Wiederverwendung mit Architektur 308

Teil A. Einführung und grundlegende Theorien

1. Einleitung

1.1. Die Automobilindustrie als Erfinder der Produktlinie

Die industrielle Revolution änderte unser Verständnis von Produktion grundlegend. Handelte es sich zuvor bei Produkten fast ausschließlich um klassische Einzelanfertigungen oder Kleinstserien, gewann nun die Massenproduktion aufgrund der damit verbundenen erheblichen Kostenvorteile an Bedeutung. Hohe Produktivität verdrängte Individualität beinahe gänzlich als Unternehmensziel. Dies führte u. a. zum berühmten Zitat von Henry Ford "Any customer can have a car painted any color that he wants so long as it is black." [WiFo09][1]

Heute besteht für ein Unternehmen nur noch in wenigen Ausnahmesituationen, z.B. bei einer großen Innovation und einer daraus resultierenden kurzzeitigen monopolähnlichen Situation, die Möglichkeit zur reinen Standardisierung. Ansonsten sind die Kundenanforderungen zu vielfältig und das Konkurrenzangebot zu umfangreich, als dass ein einziges standardisiertes Produkt für einen größeren Markt ausreichen würde. Die von Porter [Port83] postulierte Unvereinbarkeit der grundsätzlichen Markstrategien Kostenführerschaft (durch starke Standardisierung) und Differenzierung ist daher zu hinterfragen und wurde auch bereits mehrfach kritisiert [vgl. Pill98, S. 47ff. und Pine93, S. 113ff.]. Porter hat diese Unvereinbarkeit später selbst relativiert. "Durch die Einführung bedeutsamer technologischer Innovationen kann ein Unternehmen gleichzeitig sowohl seine Kosten senken als auch die Differenzierung steigern und damit vielleicht beide Strategien verwirklichen. Neue automatisierte Fertigungsverfahren können diesen Effekt ebenso hervorrufen wie neue Informationssysteme zur rechnergestützten Logistik oder Produktgestaltung" [Port00, S. 48].

Um dennoch die Kostenvorteile der Massenproduktion nicht aufgeben zu müssen, ist es notwendig eine ökonomische Balance zwischen Individualität und Standardisierung zu finden. Davis hat dafür den Begriff der Mass Customization geprägt [Davi87][2]. Damit bezeichnet er die in großem Maßstab stattfindende Produktion an individuelle

---

[1] Tatsächlich wurde das T-Model von Ford zu Beginn der Produktion 1908 bis einschließlich 1914 in mehreren Farben angeboten, zu denen allerdings nicht Schwarz zählte. Ab 1914 war das Model dann nur noch in Schwarz erhältlich, angeblich da diese Farbe am schnellsten trocknete. In den letzten beiden Produktionsjahren 1926 und 1927 war der Wagen dann erneut in mehreren Farben erhältlich.

[2] Vgl. auch [Pine93] und [Pill98], wobei Piller den etwas intuitiveren Begriff der kundenindividuellen Massenfertigung benutzt.

Kundenwünsche angepasster Produkte. Für die Kunden hat dies direkte, positive Auswirkungen, da sie nun aus einem breiteren Produktangebot das genau auf ihre Bedürfnisse Passende wählen können. Für die Unternehmen ist dies allerdings mit einer signifikanten technischen Investition verbunden. Um die Auswirkung auf den Gewinn zu reduzieren und steigende Preise zu vermeiden, versuchen die Unternehmen möglichst viele Komponenten, die für die Individualisierung von geringer Bedeutung sind, standardisiert einzusetzen.

In den 1980er Jahren hat sich in der wissenschaftlichen Theorie eine Methode entwickelt, die den Ansatz der Standardisierung nach innen – im Unternehmen – und der Variabilisierung nach außen – zum Kunden – aufgenommen und erweitert hat: Product Line Engineering [vgl. Trig03, S. 3]. Wildemann hat in diesem Zusammenhang – allerdings mit Fokussierung auf die Betriebswirtschaftslehre – den Begriff der Produktordnungsysteme geprägt [Wild04]. In der englischsprachigen wissenschaftlichen Literatur wird dazu allgemein der Begriff der "product lines" benutzt, der im deutschen nicht einheitlich mit Produktlinie oder mit Produktfamilie übersetzt wird. Eine genaue Definition und Abgrenzung der einzelnen Begriffe erfolgt in Kapitel 2.2.2, eine detaillierte Darstellung der geschichtlichen Entwicklung findet sich in Kapitel 2.1.1.

Aufgrund der vielversprechenden und nachvollziehbaren Vorteile der Wiederverwendung gab es bereits früh erste Anwendungen in der Wirtschaft. Auch wenn die Automobilbranche nicht als der eigentliche "Erfinder" der Produktlinie in Form von Plattformen anzusehen ist, so sind insbesondere die deutschen, amerikanischen und japanischen Hersteller für das heutige Verständnis der Allgemeinheit von Wiederverwendung, Verwendung von Gleichteilen und insbesondere Plattformkonzepten verantwortlich. "Populärster Vertreter der Plattformstrategie in Europa ist der VW Konzern, bei dem auf einer Plattform bis zu sieben Fahrzeugmodelle mit eine Jahresstückzahl von ca. 1,8 Mio. Fahrzeugen basieren" [Jung05, S.13].

Die möglichen Kostenvorteile und die Reduktion der Komplexität sowie die damit verbundenen kürzeren Entwicklungszeiten sind der Auslöser für die Konzerne, sich dieses Themas anzunehmen. Als ein Grund für die Pioniertätigkeit der Automobilindustrie in diesem Bereich kann die gestiegene Anzahl der Produktvarianten Mitte der 1980er gesehen werden.[3] U. a. wurde diesem auf der Hersteller-Seite mit der Einführung von Plattformen begegnet [Pohl05, S. 5]. Bei [Cusu98] findet sich eine detaillierte Beschreibung der

[3] Während ein deutscher Autokunde Mitte der 1980er Jahre die Wahl zwischen 140 Modellen hatte, waren es 2005 bereits 330 trotz einer gleichzeitig sinkenden Anzahl an Anbietern [vgl. Absa05, S. 45]

Plattformaktivitäten von 9 großen Automobilherstellern in den 1990er Jahren.[4] Eine Plattform bezeichnet dabei den "für den Kunden weniger sichtbaren Fahrzeugteil, also Chassis, Fahrwerk, Motor und Getriebe" [Schn00, S. 171]. Volkswagen war 1998 der erste Autokonzern, der ein solch weitreichendes Plattformkonzept in der Großserie erfolgreich umgesetzt hatte. Modelle auf Basis einheitlicher Plattformen erzielten einen Anteil an den Gesamtverkäufen von ca. 50% [Schn00, S.171]. Zwar wurden bereits vorher einige Teile, wie z.B. Motoren oder Getriebe, in mehreren Modellreihen verwendet, aber die Anwendung auf die gesamte Fahrzeugplattform war neu.[5] Ziel war eine deutliche Kostenreduktion, da bis zu 80% der Entwicklungskosten allein auf die Plattform entfallen [Mark03, S. 325]. Und mittels einer Plattformstrategie lassen sich Gleichteilquoten von bis zu 40% realisieren [Wann05, S. 165][6], was wiederum zu einer Einsparung bei den Entwicklungskosten von bis zu 10% führen kann [Best05, S. 124]. Wiederverwendung ermöglichte es VW die Anzahl unterschiedlicher Türschlösser von 122 auf 28 und die Zahl der Anlasser von 46 auf 10 zu reduzieren [Bryl99, S. 4]. Ebenso steigerte VW die Anzahl der auf einer Plattform gefertigten Modelle signifikant.[7] Allerdings orientierte sich VW kurze Zeit später um. Anscheinend reichte die erzielte Gewinnung von Neukunden durch ein kundenspezifischeres und breiteres Angebot nicht aus, um den negativen Kannibalisierungseffekt[8] auszugleichen. Der mit der Diversifizierung des Produktangebots verbundene Mehraufwand und die höhere Komplexität erwiesen sich als zu groß. Daher wurde das reine Plattformkonzept 2004 zum Modulkonzept erweitert. Ein Modul bezeichnet eine Baugruppe eines Fahrzeugs (also z.B. den Motor oder die Aufhängung). Die Grundidee ist erneut eine breit angelegte Wiederverwendung über verschiedene Modellreihen und einen sehr langen Zeitraum von mehr als 10 Jahren hinweg. Allerdings nun nicht mehr spezifisch für eine bestimmte Fahrzeuggruppe und möglichst unsichtbar für den Kunden.

---

4 Der Bericht deckt folgende Automarken ab: Renault, Mitsubishi, Fiat, Chrysler, Honda, Mazda, Nissan, General Motors und Ford.

5 Brylawski beschreibt die Strategie von VW treffend: "Volkswagen employs this strategy to dramatic ends, with its destinctive New Beetle, Jetta, and Golf, Audi A3 and TT, and two other European models sharing a common architecture and 65% of the same components. Few in the public realize the high-end, "halo car" $35,000 TT is heavily related to the post-modern, mass-market $17,000 New Beetle. That's the point: the automaker can spread expensive design, engineering, and manufacturing costs across 1.5 million vehicles, while customers feel they're getting unique products." [Bryl99, S. 2]

6 Nach eigenen Angaben kommt der französische PSA Konzern bei seinen Modellen Peugeot 307 und Citröen C4 inzwischen sogar auf eine Gleichteilquote von 60% [vgl. Hand06].

7 Die Plattformstrategie wurde zuerst mit dem Golf IV bei Volkswagen eingeführt. Diese Basis wurde für insgesamt 8 Modelle verwendet, neben dem Golf noch Audi TT, Seat Leon, Seat Toledo, Skoda Octavia, Audi A3, VW New Beetle und VW Bora [vgl. auch Mark03, S. 325].

8 Kannibalisierung bezeichnet die Abwanderung von Käufern innerhalb des Unternehmens zu anderen insbesondere günstigeren Modellen, z.B. von VW zu Seat oder Skoda [vgl. vSom05, S. 158 und insb. Keri78].

Der Umfang der möglichen Wiederverwendung ist bei einem Fahrzeughersteller aufgrund der geringen Anzahl an unterschiedlichen Produkten begrenzt. Ein größeres Potenzial ergibt sich für die Zulieferer, da sie die wieder verwendbaren Teile über mehrere Fahrzeughersteller hinweg nutzen können. Hierfür finden sich in der Literatur zahlreiche Beispiele. So beschreibt Ward [Ward95] wie Nippondenso, ein führender Hersteller von Kühlern und Lichtmaschinen u.a. für Toyota, aufgrund des Produktfamilienansatzes seinen Kunden mehr als 700 unterschiedliche Varianten zu konkurrenzfähigen Preisen anbieten kann. Einen wichtigen weiteren Vorteil stellt die Reduzierung der benötigten Entwicklungszeiten dar. So konnte DaimlerChrysler den im Jahre 2000 vorgestellten PT Cruiser trotz seines damals radikal neuen Designs durch Nutzung der vorhandenen Neon Platform zu Kosten entwickeln, die "a fraction of the $700 million spent on the Neon" [Mark99, S. 46] betrugen.

Zahlreiche weitere Industrien haben den erfolgreichen Ansatz der strategisch geplanten und optimierten Wiederverwendung ebenfalls umgesetzt. Als ein Beispiel sei hier der finnische Mobilfunkgerätehersteller Nokia genannt. Dank einer konsequenten Standardisierung der inneren Bauteile und der Verwendung von Plattformen konnte Nokia im Jahre 2002 gegenüber seinem Konkurrenten Motorola Kostenvorteile in der Produktion von knapp 14% realisieren [Best05, S. 124]. Heute hat sich das Plattformkonzept bei Mobilfunkgeräten weitgehend durchgesetzt. Und das nicht allein im Bereich der Hardware, auch die Software, insbesondere das Betriebssystem (Operating System, OS) bei Smartphones, kann immer stärker als Plattform verstanden werden [Pohl05, S. 15]. Das bedeutendste OS ist mit einem Marktanteil von fast 50% Symbian.[9]

Für weitere Details zur Standardisierung, Plattformstrategie und Wiederverwendung sei auf die Ausführungen in Kapitel 2.1. verwiesen.

1.2. Motivation der Arbeit

Die Nutzung von Produktfamilien hat in vielen Bereichen der industriellen Fertigung zur Reduzierung des Entwicklungsaufwands, der Entwicklungszeit, der im Produkt enthaltenen Fehler bei der Markteinführung und zur Erhöhung der Flexibilisierung des Produktionsprozesses geführt [Pohl05, S. 9ff.]. Dies ermöglichte eine Änderung der Produktion weg von der reinen Massenfertigung hin zu einer Mass Customisation, also der

[9] Weitere wichtige OS sind Apple mit einem Anteil von 17% für sein iPhone, RIM mit 15% für BlackBerry und Microsoft mit 14% für Windows CE und Windows Mobile [Cana08].

massenhaften individuellen Fertigung von Standardprodukten (siehe dazu Punkt 2.1.1.). Allerdings ergeben sich bei der realen Umsetzung der Idee der Einführung einer Produktfamilie, oder anders gesagt die Idee der strategischen Wiederverwendung, zahlreiche Schwierigkeiten. Daher basiert der wirtschaftliche Erfolg dieses Ansatzes auf zahlreichen wissenschaftlichen Arbeiten, die seit Mitte der 1980er Jahre erstellt wurden. Denn erst die Ausformulierung von detaillierten Anweisungen zur Umsetzung der Idee der Wiederverwendung von Bauteilen schaffte die notwendige Basis für den Einsatz in der Praxis. Daher kann man heute auf ein umfangreiches Wissen über Produktfamilien im (Massen-)Fertigungsbereich zurückgreifen (vgl. Kapitel 2.1.2.).

Wiederverwendung ist nicht nur bei der industriellen Fertigung ein relevantes Thema. So hat sich insbesondere um das Thema Wiederverwendung von Software ein sehr aktives und weites wissenschaftliches Betätigungsfeld gebildet. Software hat, wenn auch vielfach vom Kunden unbemerkt, eine immer stärkere Bedeutung in zahlreichen Produkten. Bourgonjon machte in diesem Zusammenhang bereits 1995, also vor dem Boom des Internets und der New Economy, zwei empirische Beobachtungen [Bour95]:

- Die Größe von Embedded Software[10] in Konsumentenprodukten wird sich nach Moore's Law [Moor 65] alle zwei Jahre verdoppeln.
- Alle Konsumentenprodukte werden in Zukunft wohl Software enthalten und dabei die oben beschriebene Wachstumskurve verfolgen; lediglich der zeitliche Startpunkt kann variieren.

Eine Analyse der Embedded Software in High-end Fernsehern verdeutlicht diesen Trend. Laut [v.Omm04, S.1] stieg die durchschnittliche Menge an Software zwischen 1978 und 1996 von 1 KByte auf 1.000 KByte. In den folgenden 8 Jahren bis 2004 erfolgte erneut eine Verzehnfachung auf 10.000 KByte. Van Ommering schließt daraus, dass "every consumer product manufacturer will eventually face the problem of having to build an ever-growing amount of software, and thus becoming – at least partially – a software company." [v.Omm04, S. 1] Im Mobilfunkbereich hat sich dieser Trend weiter bestätigt. Noch im Jahre 2006 konnte das Nokia 6300 als aktuelles Business Mobiltelefon 1.000 Telefonbucheinträge verwalten, MP3 Dateien wiedergeben, Fotos mit einer 2 Megapixel Kamera machen und

[10] Embedded Software bezeichnet Software in einem eingebetteten System. Diese sind "application specific systems which contain hardware and software tailored for a particular task and are generally part of a larger system" [Kuma96, S.11]. Eine genaue Beschreibung von embedded Software und embedded Systems findet sich in Kapitel 3.

einige vorinstallierte Spiele spielen. Beim aktuellen iPhone 4 von Apple spielt das Thema Multimedia mit HD Video Wiedergabe und 5 Megapixel Kamera ein viel bedeutenderes Thema. So sind bis Mitte 2010 bereits über 250.000 Applikationen im iTunes Store verfügbar. Allein auf elektronischer Hardwareebene, also ohne Betriebssystem und Anwendungen umfasst das iPhone mehr als 1,7 Mio. Zeilen Code. Die Entwicklung der Software hat mit mehr als 150 Mannjahren fast so lange gedauert wie die Entwicklung der elektronischen Hardware mit 200 Mannjahren. Die ökonomischen Folgen davon zeigen sich z.B. am Ebit Anteil der 7 großen Mobilfunkanbieter. Während Apple in nur 3 Jahren von 2007 bis 2010 seinen Anteil von 1% auf über 48% steigern konnte, fiel der Marktführer Nokia von 63% auf nur noch 22% zurück, da „der Handyhersteller bei der Produktion von Smartphones einen zeitigen Paradigmenwechsel verschlafen“ [Fina10] hat.

Viele neue Produkte oder Innovationen in bestehenden Produkten sind überhaupt nur durch den verstärkten Einsatz von Software möglich, oder anders ausgedrückt die "Funktionalität wird mehr und mehr in Software realisiert" [Böck04a, S.3]. Hier seien nur einige wenige Beispiele genannt: Telefonieren über das Internetprotokoll (IP Telefonie); Konvergenz von Mobiltelefonen, Digitalkameras, MP3 Playern und Organizern; Flachbild-Fernseher mit digitaler Bildaufbereitung[11]; Web 2.0; Verschlüsselung der Kommunikation z. B. für Online Banking; usw.

Man könnte formulieren, dass in vielen Bereichen die Hardware immer mehr zu einem Commodity wird und die Innovationen stärker durch Software realisiert werden. Ein gutes Beispiel dafür ist der Personal Computer (PC). IBM glaubte zu Beginn des Computerzeitalters u.a. aufgrund der eigenen starken Wettbewerbsposition und der Besonderheiten im damals vorherrschenden Großrechnerbereich, dass die Hardware der entscheidende Faktor für zukünftigen Erfolg sein werde. Als IBM sich 1981 als damals größter Computerhersteller entschloss in den sich entwickelnden PC Markt einzusteigen, gingen die Verantwortlichen davon aus, dass die Bedeutung der Hardware auch in diesem Umfeld Gültigkeit haben würde. "IBM wanted the hardware business and did not care about the software" [Both97]. Als Folge daraus erlaubten sie Microsoft, eigene Kopien von MS-DOS zu lizenzieren. MS-DOS konkurrierte zu dieser Zeit mit CP/M um die Stellung als vorherrschendes Betriebssystem. Beide Systeme waren als Open Architecture gestaltet und

[11] Moderne Geräte verfügen teilweise über eine Bildwiederholfrequenz von 200 Hertz. Dabei werden zwischen zwei real übertragenen Bildern bis zu 4 Zwischenbilder berechnet, um eine flüssige Darstellung von Bewegungen zu ermöglichen.

beide liefen auf den frühen 8086 Chips von Intel. Während nun zunächst CP/M den Markt dominierte, höchstwahrscheinlich aufgrund der guten Versorgung mit günstiger oder sogar kostenloser Software von Heimentwicklern, änderte sich das Bild bis 1984 dramatisch. Zu diesem Zeitpunkt waren über 11.000 verschiedene Software Programme für MS-DOS verfügbar [vgl. Gand95, S. 20]. Hinzukam, dass IBM durch seine große Kundenbasis viele Geschäftskunden an das neue System heranführen konnte. Diese gewöhnten sich an MS-DOS und zwangen wenig später alle Anbieter von IBM-Klonen, MS-DOS bei Microsoft zu kaufen. Insgesamt wurden von MS-DOS schätzungsweise über 200 Millionen Lizenzen verkauft [Leit09]. "Der IBM-PC mit Intel-Prozessor und dem Microsoft-Betriebssystem verdrängte schnell andere PC-Designs und wurde zum internationalen Standard" [Beis06, S. 65].

Auch aus der jüngsten Zeit findet sich ein eindrucksvolles Beispiel. Der Erfolg von Apple seit 2001 ist laut dessen CEO Steve Jobbs weitgehend auf die Überlegenheit der Software des Unternehmens (im Zusammenspiel mit einem entsprechenden Marketing) zurückzuführen.[12] Zwar spielt insbesondere bei den iPods, den iMacs und den Apple MacBooks auch das Design der Hardware und die kürzlich hergestellte Kompatibilität mit dem IBM PC Standard eine wichtige Rolle, aber noch wichtiger ist die einfache und fehlerfreie Software, die die Nutzung der Hardware überhaupt erst ermöglicht, sei es das Leopard Betriebssystem oder auch iTunes zur Synchronisierung der Daten des iPods mit dem PC/Apple Computer [Lane06]. Die Musikindustrie hatte über Jahre vergeblich versucht, eine finanziell erfolgreiche Nutzung der neuen digitalen Medien zu etablieren ohne neue Technologien zu bekämpfen oder einzuschränken. Dies ist Apple mit iTunes gelungen [Kitt09, S. 19].

In immer mehr Embedded Software Bereichen – wenn auch noch lange nicht wie in dem Umfang bei PCs – lässt sich eine ähnliche Entwicklung erkennen. So ist für die Hersteller von Mobiltelefonen schon heute die Software das entscheidende Differenzierungsmerkmal. Die Mobilfunk-Komponenten werden weitgehend von den gleichen Zulieferern bezogen. Eine Differenzierung im Hardwarebereich ist lediglich noch in einigen Ausnahmen (z.B. bei Technologiesprüngen) möglich. So konnte Motorola 2004 mit seinem Handymodell RAZR einen weltweiten Erfolg verbuchen. Ein Hauptgrund dafür war die exklusive Lizenzierung der damals neuen ultra-slim GSM Frontends von Skyworks [vgl. Quir05, Triq06 und Cook05], wodurch Motorola in der Lage war, die damals dünnsten Mobilfunkgeräte herzustellen. Trotz

[12] Die Aussage stammt aus einem gemeinsamen Interview von Steve Jobbs und Bill Gates im Rahmen der vom Wall Street Journal jährlich veranstalteten Konferenz "D: All Things Digital" in den USA [Pacz07].

eines im Vergleich zu damaligen Konkurrenzprodukten geringen Funktionsumfangs und schlechter Benutzerführung konnte sich das RAZR am Markt durchsetzten und schaffte im Mobilfunkbereich das, was der iPod im Bereich der digitalen Musikabspielgeräte schaffte. [13] Apple lieferte mit der Einführung des iPhones ein weiteres Beispiel. Die innovative Bedienung über ein berührungsempfindliches Display bot völlig neue Möglichkeiten. Letztendlich erfolgreich wurde das Produkt allerdings durch die mächtige Software, die es verstand, die neue Hardware sinnvoll zu nutzen.[14]

Insgesamt kann man feststellen, dass die Bedeutung der Software signifikant zugenommen hat. Dies gilt insbesondere für die Automobilindustrie, da in einem modernen Fahrzeug bis zu 80 elektronische Steuergeräte (ECU)[15] mit der entsprechenden Software verbaut sind und bereits heute über 50% der Entwicklungskosten eines Steuergerätes auf den Softwareanteil entfallen [Kreu05, S. 70]. Daraus ergeben sich drei Herausforderungen für die Hersteller (OEMs):

- "Traditionally, carmakers focus on the design, planning, and assembly of hardware components, with software being largely an afterthought." [McKi06, S. 61]. Während viele Zulieferer sich in den vergangenen Jahren ein beachtliches Wissen in diesem Bereich erarbeitet haben, hinken die OEMs vielfach hinterher [Prof08 , S.217]. Die Verantwortung für Fehler trifft aber schließlich die Hersteller selbst. In Deutschland hängen "fast 50% aller PKW-Defekte mit der Software zusammen, vor allem in den Bereichen Infotainment, elektronische Fahrwerksabstimmung und Einspritzsystem. Bei Volkswagen werden bspw. 48% der Defekte durch Elektronik und Software verursacht." [Prof08, S. 217] Als Folge stieg die Zahl der durch Elektronik- und Softwarefehler verursachten Rückrufe in Deutschland zwischen 1995 und 2004 um jährlich 17 % [vgl. Prof08 , S.217]. Die Reduzierung der Produktfehler und damit die Verbesserung der Produktqualität muss ein vorrangiges Ziel für die Unternehmen sein.

---

[13] Im Gegensatz zum iPod könnte sich dieser Erfolg aber als kurzfristig erweisen. Denn ein ausgefallenes Design allein garantiert keine langfristige Kundenbindung über Modellreihen hinweg. Laut einer Umfrage unter Motorola Razr Nutzern würden 78% kein weiteres Mobiltelefon von Motorola kaufen. Hauptgrund ist die schlechte und zu langsame Benutzerführung. Unter Erstnutzern liegt der Prozentanteil sogar bei 85% [siehe User06].

[14] Einige Beispiele sind der intuitive 2 Finger Zoom, das einfache Scrollen, die gute Tastaturfunktion mit automatischer Erkennung der gewünschten Taste. Zusätzlich ist die Software iTunes zu nennen, die die Kommunikation des Gerätes mit dem PC oder Mac ermöglichte und ebenfalls umfangreiche Funktionen, wie z.B. den Zugang zum iTunes Store enthält.

[15] Ein Steuergerät, oft auch als Electronic Control Unit (ECU) bezeichnet, ist die physikalische Implementierung eines eingebetteten SW-Systems. Im Automobilbereich bilden ECU, Sensorik und Aktuatorik oft eine Einheit [vgl. Broy98a, S. 13].

- Die Kunden fordern eine immer stärkere Individualisierung der Produkte, für die ein kleines Käufersegment bereit ist mehr zu bezahlen.[16] Die Hersteller müssen einen Weg finden, die Produktvielfalt zum Kunden zu erhöhen, ohne damit einen signifikanten Mehraufwand hervorzurufen, um trotz der geringeren Verkaufszahlen pro Modell weiter profitabel produzieren zu können. Möglich ist dies bspw. durch eine klare Trennung von für den Kunden unterscheidungsrelevanten Teilen und Teilen, bei denen er keinen Unterschied feststellen kann oder bei denen kein Mehrwert aus einer Differenzierung resultiert. Dass dies auch falsch verstanden werden kann, zeigt ein Beispiel von Mercedes Mitte der 1990er Jahre. Damals wurde von der A- bis zur S-Klasse im Kofferraum das identische Plastikteil zur Anhebung der Reserverad-abdeckung verwendet. Dies kann zu Irritationen bei den Käufern der hochpreisigen Produkten führen.

- Trotz der zusätzlichen Komplexität gilt es, die Produktentwicklungszeiten weiter zu reduzieren, um damit den verkürzten Produktlebenszyklen zu begegnen [Kötz05, S. 488]. Die frühere Strategie deutscher Hersteller, ein Modell weitgehend unverändert über einen Zeitraum von fast einem Jahrzent zu verkaufen, ist heute nicht mehr möglich. Vielmehr muss es dass Ziel sein, die Produktzyklen entsprechend der Kundenwünsche zu reduzieren. Das Vorgehen asiatischer Automobilhersteller, die vielfach bereits nach ein paar Jahren ein scheinbar komplett neues Modell auf den Markt bringen, wobei die Unterschiede meist nur eine geringe Anzahl sichtbarer Teile betrifft, zeigt erste Lösungsoptionen auf. Aber auch die zunehmende Bedeutung von Innovationen aus dem elektronischen Bereich bedeutet, dass eine Annäherung an die kurzen Produktlebenszyklen der Elektronikindustrie erfolgen muss. Sehr deutlich wird dies im Bereich der Navigationssysteme. Mobile Geräte haben den Nachteil, dass sie aufgrund fehlender Daten aus den Fahrzeugsensoren (wie z.B. Lenkeinschlag und Geschwindigkeit) allein auf die GPS-Ortung angewiesen sind, was sich bspw. in Tunneln negativ auswirken kann. Andererseits bieten diese aber eine deutlich höhere Funktionsvielfalt zu deutlich niedrigeren Preisen. In diesem Bereich kommen jeden

[16] Fahrzeuge im Bereich Sport Utility Vehicle (SUVs), also Fahrzeuge, die zusätzlich zur Strasse auch eingeschränkt die Möglichkeit zur Bewegung im Gelände und/oder eine sehr sportliche Auslegung bieten, zielen auf eine Käufergruppe ab, die bereit ist signifikant höhere Preise zu zahlen. Allerdings ist diese Käufergruppe begrenzt, sodass sich eine eigenständige Entwicklung meist nicht lohnt. Subaru und Honda bspw. erzielten Ende der 1990er für die Modelle Subaru Outback und Honda CR-V ungefähr 5.000$ höhere Preise als für die Mainstream Modelle Subaru Legacy und Honda Civic [Bryl99, S.3].

Monat unzählige neue Produkte auf den Markt, während die Serienhersteller im Allgemeinen lediglich bei einem Modellwechsel oder einer größeren Modellpflege hier eine Anpassung vornehmen. Und selbst diese Produkte enthalten dann nicht den gleichen Funktionsumfang wie mobile Geräte. Der große Erfolg mobiler Geräte zeigt den Handlungsbedarf der Hersteller.

Ein möglicher Lösungsansatz sind die bereits angesprochenen Produktfamilien [Clem02], also die strategische Planung und Berücksichtigung von Wiederverwendung bei der Konzeption ganzer Produktreihen. Macht man sich zusätzlich bewusst, dass zukünftig der Großteil der Innovation im Bereich der Software erwartet wird[17], wäre eine Übertragung des Produktfamilienkonzepts auf den Bereich der Softwareentwicklung wünschenswert. Den Zusammenhang zwischen Variantenzahl, Stückzahlen pro Variante, Stück- und Gesamtkosten und den Möglichkeiten, die eine modulare Produktfamilienarchitektur hierbei bietet, erläutert Abbildung 1.

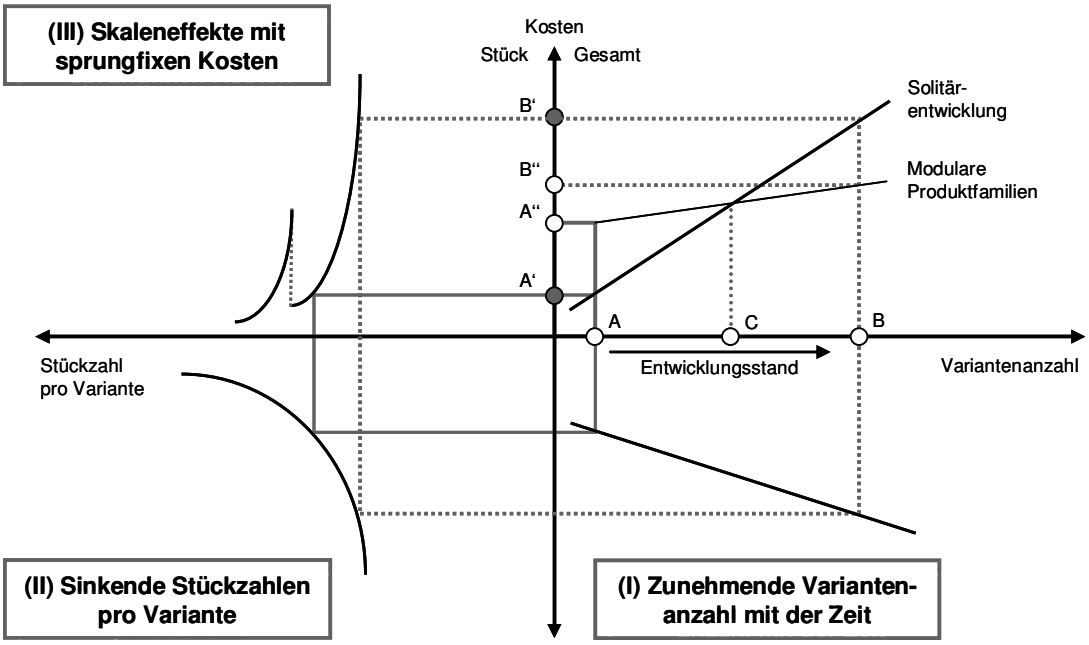

Abbildung 1 – Produktfamilien im Rahmen der Variantenproblematik [Jung05, S. 5]

Bei einer Zunahme der Variantenanzahl (I), also einer Veränderung von Punkt A zu Punkt B, ergeben sich bei ungefähr konstanter Absatzmenge niegrigere Stückzahlen je Variante (II). Umgekehrte und damit negative Skaleneffekte (III) führen zu höheren Stückkosten. Wenn

[17] In der wissenschaftlichen Literatur besteht über diese Aussage weitgehende Einigkeit. Schätzungen für den Anteil der auf Software basierenden Innovationen gehen von 70%/80% bis 90% aus [vgl. McKi06, S. 11 und Kötz05, S. 488].

sich die Stückkosten erhöhen, folgt direkt eine Erhöhung der Gesamtkosten. In der Abbildung wird dieser Zusammenhang durch die Veränderung von Punkt A' zu B' gezeigt. Die Produktfamilie weist im Gegensatz zu Einzelentwicklungen aufgrund der notwendigen Anfangsinvestition zunächst höhere Kosten auf (Punkt A"). Bei der angenommenen Zunahme der Variantenzahl ergeben sich schließlich jedoch geringere Gesamtkosten (Punkt B"). Die entscheidende Größe ist in diesem Fall die Variantenanzahl. Steigt diese über den Wert C, lohnt sich die Investition in eine Produktfamilie.

Zu diesem Thema sind unter der Überschrift Software-Produktfamilien oder auch Software-Produktlinien (SWPL) in den letzten Jahren zahlreiche Publikationen erschienen. Das Software Engineering Institute der Carnegie Mellon University ist hier führend. Aber auch in Europa sind mehrere Forschungsprojekte zu diesem Themenkomplex durchgeführt und zahlreiche Arbeiten verfasst worden. Im Detail wird der aktuelle Forschungsstand in Kapitel 2.2.5. besprochen.

Eine reine Beschreibung der Methodik reicht jedoch in der Praxis nicht aus. Insbesondere da die Einführung einer Produktlinie mit erheblichen Investitionen verbunden ist, bedarf es einfach anzuwendender ökonomischer Bewertungsmodelle, um die Auswirkungen einer Produktfamilie im Vorfeld besser abschätzen zu können. Für eine detaillierte Darstellung und Bewertung der vorhandenen wissenschaftlichen Arbeiten wird auf Teil B Kapitel 3 verwiesen. Eins sei aber vorweggenommen: Eine Gesamtbetrachtung der vorhandenen Literatur zeigt "more business case analyses are required" [Knau02, S. 44] und es lassen sich zwei grundlegende Verbesserungspotenziale identifizieren. Einerseits wird bei den bisherigen Modellen mit stark vereinfachten Annahmen gearbeitet, indem für elementare Einflußgrößen Standardwerte verwendet werden, über die in der Literatur keine Einigkeit besteht.[18] "Models are needed that operate on a more detailed level. For example, ROI models are needed that predict the ROI of a certain feature" [Pohl05, S. 438]. Und weiterhin decken die Modelle meist lediglich einen oder eine geringe Zahl von Spezialfällen ab. Wünschenswert wäre an dieser Stelle ein einfach anzuwendendes, umfassendes und generisches ökonomisches Modell zur frühzeitigen Bestimmung der Auswirkungen einer SWPL.

[18] So wird bspw. der Zusatzaufwand zur Herstellung wiederverwendbarer Software in der Literatur mit einem Faktor zwischen 1,5 und 9 bewertet [vgl. Broo95, S. 224, Jaco97, S. 23 und Trac95, S. xi].

1.3. Aufgabenstellung und Zielsetzung

Ausgehend von den in der wissenschaftlichen Literatur angesprochenen Verbesserungsmöglichkeiten, ist das Ziel dieser Arbeit die Erstellung eines umfassenden und generischen ökonomischen Modells zur frühzeitigen Bewertung von Software Produktlinienansätzen. Das Modell soll universell einsetzbar sein, im Rahmen der Arbeit erfolgt jedoch eine Fokussierung auf die Automobilindustrie, die auch die erste Zielgruppe dieser Arbeit darstellt. Dies spiegelt sich in der Auswahl der Praxispartner und zahlreicher Fallbeispiele wider. Unternehmen haben zwar schon jetzt die Möglichkeit auf eine Vielzahl von Modellen für eine ökonomische Bewertung eines Produktlinienansatzes zurückzugreifen. Allerdings handelt es sich dabei meist entweder um spezialisierte Ansätze mit engen Vorgaben oder um eine sehr starke Vereinfachung der Realität. Zusätzlich ist vielfach der nötige Zeitaufwand in der Praxis nicht realisierbar. Eine detaillierte Beschreibung der vorhandenen Ansätze und Modelle erfolgt in Teil B Kapitel 3.

Mit dem hier vorgestellten Ansatz werden die Unternehmen in die Lage versetzt, mit einem vertretbaren Aufwand die ökonomischen Folgen der Einführung einer Software Produktlinie abschätzen zu können – und zwar zu einem Zeitpunkt in der Planung, an dem noch keine oder nur geringe Investitionen angefallen sind.

Im Rahmen dieser Arbeit wird das entsprechende Modell ausführlich erklärt, sodass eine individuelle Anwendung auf Basis dieser Arbeit möglich ist. Die notwendigen Arbeitsblätter sind direkt im Text oder im Anhang abgedruckt. Zur Vereinfachung einer individuellen Anwendung können diese zusätzlich direkt beim Verfasser angefordert werden. Anhand dieser Hilfsmittel ist es jedem Unternehmen der Automobilindustrie möglich, eine erste individuelle Bewertung in einem Zeitrahmen von 3-5 Tagen durchzuführen. Die Schätzung basiert auf der Annahme, dass das Unternehmen Grundkenntnisse in der Anwendung der Function Point Methodik (siehe Kapitel 5.2.) hat. Dabei handelt es sich um ein Verfahren zur effizienten Abschätzung des Entwicklungsaufwands und der allgemeinen Komplexität einer Software. Das Modell basiert grundlegend auf dieser Methodik. Sollte dies nicht der Fall sein, kann eine notwendige Einarbeitung aufgrund umfangreich verfügbarerer Literatur[19] zeitnah und mit überschaubarem Zeitaufwand (ca. 1 Personenwoche) erfolgen.

[19] Zwar findet sich in Kapitel 5.2. der vorliegenden Arbeit eine kurze Übersicht zur Function Point Methodik, allerdings reicht diese Ausführung für ein grundsätzliches Verständnis des Vorgehens nicht aus. Es ist auch nicht das Ziel dieser

1.4. Struktur der Arbeit

Die Arbeit gliedert sich in zwei Hauptteile. Der erste Teil bietet eine Einleitung in die wichtigsten Themen und Begriffe dieser Arbeit. Zunächst erfolgt eine Einleitung in das große Thema der Produktlinien, die dann für den Softwarebereich spezifiziert werden. Im Anschluss werden Embedded Systems eingeführt. Schließlich erfolgt eine Zusammenführung der beiden Themengebiete Produktlinien und Embedded Systems mit speziellem Fokus auf die Besonderheiten der Automobilindustrie.

Das zweite Kapitel stellt vorhandene ökonomische Modelle allgemein und das im Rahmen der vorliegenden Arbeit entwickelte e-FACTS Modell[20] vor. Die e-FACTS Analyse folgt dabei den folgenden Teilschritten, die jeweils detailliert erklärt werden:

- **Fragebogen**: mit Hilfe eines kurzen Fragebogens wird die grundsätzliche Eignung oder auch Reife des Unternehmens zur Einführung einer Produktlinie ermittelt. Dies geschieht mit dem Ziel, möglichst früh ungeeignete Fälle auszuschließen, um die mit der Analyse verbunden Kosten zu minimieren. Der Fokus liegt hierbei allerdings klar auf der Minimierung der False/negative Erkennungen, d.h. es wird eher in Kauf genommen eine unnötige Detailanalyse durchzuführen als eine sinnvolle abzulehnen.
- **Einfache Analyse**: bei einem positiven Ergebnis des Fragebogens erfolgt eine erste schnelle Analyse der Produktstruktur gepaart mit ersten Ideen hinsichtlich der Auswahl der Produktlinien-Assets, also der wieder verwendbaren Codebestandteile. Bei einer zu geringen Asset Base oder einer zu geringen Anzahl Wiederverwendungsmöglichkeiten könnte die Analyse abgebrochen werden.
- **Detailanalyse**: dies erfordert eine genauere Analyse der geplanten zukünftigen Produkte in einem Zeitrahmen von 5 bis 10 Jahren. Am Ende steht eine Einteilung aller Funktionen oder Komponenten in wieder verwendbare und individuelle Softwarebestandteile.

Arbeit die Function Point Methodik im Detail zu erklären. Der Fokus liegt auf der Erklärung der Unterschiede zur hier verwendeten Feature Point Methodik und deren Herleitung. Für eine Einarbeitung in die Function Point Methodik sei auf [Jone07], [IFPU09] und die Internetseite der International Function Point Users Group (IFPUG) verwiesen [http://www.ifpug.org/].

[20] e-FACTS steht für economic Feature Point Analysis with compatible timing for Software Product Lines

- **Featurepoint Zählung**: Basis der ökonomischen Bewertung stellt die Feature Point[21] Zählung da. Es erfolgen zwei separate Zählungen, einmal mit und einmal ohne Wiederverwendung. Dies ermöglicht eine wesentlich genauere Abschätzung des notwendigen Mehraufwands bei der Erstellung einer Software Produktlinie im Vergleich zu einer Betrachtung der Lines of Code (LoC), wie sie vielfach bei ökonomischen Modellen verwendet wird. Hinzu kommt, dass eine LoC Betrachtung in den meisten Fällen erst zu einem späten Entwicklungszeitpunkt möglich ist.

- **Bewertung**: Zunächst ermöglicht die getrennte Erfassung der Erstellungsaufwände auf Komponenten- oder Funktionsebene eine mathematische Optimierung der Assetbase. Dabei ist allerdings noch die reale Durchführbarkeit dieser theoretisch optimalen Lösung zu überprüfen. Schließlich kann der Gesamtaufwand sowohl für eine individuelle Entwicklung als auch für die geplante Produktfamilie ermittelt werden.

- **Reale Beispiele**: Anhand von zwei realen Fallbeispielen werden die Praxisrelevanz und die einfache Anwendbarkeit des Modells illustriert. Gleichzeitig dient dieser Schritt dem Nachweis der Praktikabilität und der Qualität der Methodik, da in beiden Fällen bereits realisierte Software Produktfamilien betrachtet worden sind. Für diese lagen dementsprechend auch die tatsächlichen Werte nach der Umsetzung vor. Zum Ende erfolgt jeweils eine Sensitivitätsanalyse, um die Stabilität der Modell-Empfehlung zu untersuchen. Zusätzlich erfolgt eine allgemeine Betrachtung möglicher Fehler bei wichtigen Parametern des Modells.

Die Arbeit schließt mit einem Ausblick auf die notwendigen nächsten Schritte und die möglichen und sinnvollen Erweiterungen des vorgestellten Modells.

[21] Feature Point bezeichnen speziell auf embedded Software angepasste Function Points (siehe Kapitel 5.5). Im weiteren Verlauf wird die Abkürzung FP ausschließlich für Feature Points benutzt.

2. Software Produktlinien

2.1. Produktlinien: Die Idee der strategischen Wiederverwendung

2.1.1. Begriffsbestimmung

Wie bereits in der Einleitung zu Teil A erwähnt, unterlag die Art der Produktion von Gütern in den letzten 150 Jahren einem erheblichen Veränderungsprozess, von der individuellen Einzelfertigung über die reine Massenfertigung bis hin zur, heute in vielen Bereichen vorherrschenden, individualisierten Massenfertigung. In der Automobilindustrie z.B. zeigt sich diese Entwicklung an insgesamt drei Revolutionen, die zu erheblichen Veränderungen in der Industrie geführt haben. Die erste Stufe ist die Einführung der Fließbandfertigung durch Henry Ford 1913. Die zweite Revolution Ende der 1980er, Anfang der 1990er war die Einführung des Lean Managements. Während sich die beiden ersten Stufen klar auf Verbesserungen bei der Produktion konzentrierten, fokussiert die aktuell dritte Revolution auf den Entwicklungsbereich. Ziel ist die effiziente Variabilität, also die massenhafte Individualisierung von Standardprodukten entsprechend der Kundenwünsche [vgl. Hütt08]. Unzweifelhaft sind damit für den Kunden Vorteile verbunden, jedoch erzeugt dies bei der Fertigung eine deutlich höhere Komplexität und damit auch höhere Kosten, die normalerweise über höhere Preise an den Kunden weitergegeben werden. Es sei denn, es gelingt dem Unternehmen durch geeignete Maßnahmen die angesprochenen Kostensteigerungen zumindest teilweise zu kompensieren und eine "vollständige oder annähernde Kostenparität" [Port00, S. 41] zu seinen Wettbewerbern zu erreichen.

Trotz Individualisierung der Produkte gibt es viele Produktbestandteile, die nicht oder nur geringfügig zwischen den einzelnen Varianten variieren, wie z.B. der Produktkern. So hat für einen Fast-Food Kunden die Tatsache, dass das Brötchen für einen Burger mit Fleisch, Fisch oder vegetarisch identisch ist, wahrscheinlich keine kaufentscheidende Relevanz. Bei einem Auto bspw. dürfte der Käufer eher indifferent reagieren, wenn die Bodenplatte und die Federung auch in anderen Modellen verbaut werden. Viel wichtiger sind hier die durch den Kunden direkt wahrnehmbaren Elemente, wie z.B. Karosserie, Beleuchtung usw. Etwas allgemeiner formuliert geht es gerade beim Automobil um die Erfahrbarkeit der Veränderung. Eine Individualisierung ist tatsächlich nur bei solchen Eigenschaften sinnvoll, bei denen eine Differenzierung für den Kunden auch einen wahrnehmbaren Unterschied und damit einen Mehrwert bedeutet.

Wiederverwendung ist keine Erfindung der Produktlinie. Mit zunehmender Produktkomplexität ergab sich die Notwendigkeit zur Reduktion der Entwicklungsaufwände, Elemente so weit wie möglich weiterzuverwenden. Allerdings geschieht dies häufig zufällig und im Nachhinein, d.h. erst wenn die Neuentwicklung ansteht, macht man sich Gedanken über die Wiederverwendung. Die Chancen auf eine mögliche Wiederverwendung erhöhen sich deutlich, wenn dieses Thema strategisch über mehrere Produkte und Produktgenerationen hinweg betrachtet und die Wiederverwendung bereits zu Beginn mit kalkuliert wird.

Der Ansatz der Produktlinie basiert auf diesem Grundkonzept der strategischen Wiederverwendung, also der Verwendung von Gleichteilen, der Standardisierung von Schnittstellen, dem Baukastenprinzip und Modul- und Plattformstrategien. Eine Produktlinie ist allgemein ein Set von "systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way." [SEI09a] Es wird also auf einen Markt abgezielt, bei dem die Kunden ein ähnliches Bedürfnis haben, sich aber doch soweit unterscheiden, dass ein einziges Produkt nicht ausreichen würde. Ein Automobilhersteller kann unterschiedliche Kundenwünsche beispielsweise mit zwei komplett eigenständig entwickelten Modellen abdecken - dabei handelt es sich um ein klassisches Variantenmanagement. Bezieht der Hersteller bei dem Entwurf der beiden Modelle allerdings Überlegungen zu einer weitgehenden Wiederverwendung von einzelnen Komponenten mit ein, entspricht dies dem Grundgedanken einer Produktlinie.

Ein wichtiges Element von Produktlinien ist die Flexibilität, oft auch als Variabilität bezeichnet. Variabilität ist "the ability to change or customize a system" [Svah00, S.147]. Die reine Wiederverwendung von unveränderbaren Standardteilen würde die Möglichkeiten stark einschränken. Lässt man jedoch eine gewisse Flexibilität bei diesen Teilen zu, erhöhen sich die Anwendungsmöglichkeiten drastisch. Im oben angesprochenen Burger Beispiel könnte dies bedeuten, dass man zwar immer die gleichen Basis-Brötchen verwendet, es aber bspw. in der vegetarischen Variante mit Sesamsamen bestreut ist.

Produktlinien kombinieren das Konzept der Wiederverwendung mit einem gewissen Maß an Flexibilität, wobei im Rahmen von Produktlinien fast ausschließlich von Variabilität gesprochen wird. Dazu nutzen sie Plattformen, Gleichteile, standardisierte Schnittstellen und

die Idee der individualisierten Massenfertigung (Mass customization[22]). Die Entwicklung einer Produktlinie kann dem Software Engineering Institute (SEI) folgend in drei Schritten beschrieben werden [vgl. Clem02, S. 29]:

1. Product Management: Die erste Phase umfasst hauptsächlich ökonomische Aspekte. Ziel ist es, den Umfang der Produktlinie unter wirtschaftlichen Gesichtspunkten festzulegen. Dazu gehört die Definition eines Produktportfolios und einer Product Roadmap. Es wird teilweise argumentiert, dass diese erste Phase ein Bestandteil der zweiten Phase ist [siehe z.B. Pohl05], fest steht jedoch, dass dieser Schritt unabdingbar für die weiteren Phasen ist.

2. Domain Engineering: Domain Engineering ist eine "activity of collecting, organizing, and storing past experience in building systems or parts of systems in a particular domain in the form of reusable assets (i.e. reusable work products), as well as providing an adequate means for reusing these assets (i.e. retrieval, qualification, dissemination, adaptation, assembly, and so on) when building new systems" [Czar00, S. 20]. Hauptaufgabe ist demnach die Festlegung gemeinsamer und individueller Produktbestandteile, also die Aufteilung in einen Kern von wieder verwendbaren Elementen (der sogenannten Asset Base oder Domain) und der individuell zu entwickelnden Komponenten. Ein wichtiger Bestandteil ist dabei die Vereinbarung von einheitlichen Anforderungen an alle Bestandteile der Produktlinie. Und schließlich umfasst diese Phase auch noch die eigentliche Erstellung der Assets der Domain und deren Test.

3. Product/Application Engineering: Application Engineering beschreibt den "process of producing concrete members of a system family using the reusable assets developed during Domain Engineering" [Czar00, S.754]. Die eigentliche Produktentwicklung geschieht mittels Kombination existierender Assets, die über ihre eventuell eingebaute Flexibilität, angepasst werden können. Möglicherweise werden auch individuelle Komponenten hinzugefügt, die lediglich für ein einzelnes Produkt benötigt werden. Diese werden in der dritten Phase entwickelt und mit den entsprechenden Assets zu einem vollständigen Produkt verknüpft.

Ein interessantes Beispiel für die Auswirkungen einer Produktfamilie liefert die PC Industrie: Denn erst durch die Standardisierung der Schnittstellen und die Einführung einer

---

[22] Mass Customization wird verstanden, als ein Ansatz, "bei dem jeder Nachfrager als eigenständiges Marktsegment betrachtet wird, an dessen Wünschen und Bedürfnissen sich der Anbieter ausrichtet und für den er auf der Basis eines Massenproduktes ein individuelles Produkt fertigt; das Ziel besteht darin, den Kunden spezifische Problemlösungen anzubieten, ohne bei der Fertigung auf die Kostenvorteile der Massenproduktion zu verzichten." [Nies02, S. 1296] vgl. aber auch [Davi87].

Standardarchitektur erfolgte der breite wirtschaftliche Durchbruch, da die Kosten drastisch reduziert werden konnten [Bryl99, S. 4]. Apple brachte 1978 den ersten echten Personal Computer auf den Markt, den Apple II. Dieser zeichnete sich durch seine variable Aufrüstbarkeit über insgesamt 8 Steckkartenplätze aus.[23] Noch konsequenter wurde das Konzept der Produktlinie Anfang der 1980er Jahre von IBM verfolgt [vgl. Chip08, S. 26]. Der Erfolg von IBM war so groß, dass sich auch andere Hersteller gezwungen sahen, dieses Konzept zu übernehmen. Die dabei festgelegten Standards eines IBM PCs schafften den weltweiten Durchbruch und schufen einen Quasi-Industriestandard, der sogar Auswirkungen auf die heutige PC Architektur hat [vgl. Beis06, S.63 ff.]. Und dennoch ist es möglich mit dieser einheitlichen Struktur durch gezielte Differenzierung ein Preis-Premium zu erzielen, so wie es Apple mit dem iMac gelang [Bryl99, S. 4].

Das PC Beispiel zeigt, dass in einem sehr innovativen, aber auch unstandardisierten Markt eine gewisse Stabilität zum wirtschaftlichen Erfolg beiträgt. Man kann sogar formulieren, dass reife und daher stabile Märkte eine wichtige Bedingung für erfolgreiche Produktlinien sind. Denn der Erfolg einer Produktlinie basiert auf der möglichst häufigen Wiederverwendung der Assets. Handelt es sich um ein innovatives Produkt, ist davon auszugehen, dass der Anteil der wiederverwendbaren Komponenten geringer ist und die vorhandenen häufiger angepasst werden müssen, als bei Produkten, bei denen man auf eine lange Produktionserfahrung zurückblicken kann und die eine gewisse Stabilität erreicht haben. Aber selbst in sehr innovativen Märkten mit einer geringen Zahl an einzelnen Produkten kann der Produktlinienansatz in ausgewählten Bereichen sinnvoll sein. So z.B. in der Luftfahrtindustrie. Airbus setzt bei einem Großteil der aktuellen Baureihe fast unveränderte Cockpits ein. Bei den A330 und A340 Modellen wird sogar nur eine einzige Standard-Flugzeugzelle verwendet, auf dessen Basis insgesamt 6 verschiedene Konfigurationen mit 2 oder 4 Triebwerken gefertigt werden. Die Sitzplatzkapazität variiert von 240 (A340-200) bis zu 380 (A340-600). Und bei der Reichweite geht die Spanne von 10.500 km (A330-300) bis zu 16.700 (A340-500). Dabei haben alle A330 und A340 Modelle stark ähnliche Flugeigenschaften. Dadurch wird es Fluggesellschaften ermöglicht, die Flotte

[23] Interessanterweise kam damals die beliebteste Steckkarte von Microsoft. Die Softcard ermöglichte es, Bürosoftware auf dem damaligen Standardbetriebssystem CP/M laufen zu lassen.

mit einem einzigen Pool an Piloten zu operieren.[24] Detaillierte Ausführungen zum Thema der Stabilität von Anforderungen finden sich in Teil B Kapitel 5.1.1.1.

2.1.2. Aktueller Forschungsstand

Wildemann [Wild04] beschreibt mit seinem Konzept der Produktordnungssysteme einen umfassenden klassischen Produktlinienansatz, der es Unternehmen ermöglicht, über ihre gesamte Produktpalette hinweg, ein optimales und an die aktuellen Kundenbedürfnisse angepasstes Produktprogramm zu entwickeln und beizubehalten. Auch Wildemann fokussiert auf die Verbindung von kosteninduzierter Standardisierung und kundeninduzierter Individualisierung. Ziel des Ansatzes ist es, nach innen eine weitgehende Standardisierung, nach außen eine weitgehende Individualisierung zu erreichen. Basis ist eine stabile Produktgrundstruktur, die aber eine hohe Flexibilität in der Kombination der einzelnen Bestandteile aufweist.

Das Konzept fasst die 4 bekannten Einzelstrategien zur Erzielung einer höheren Wiederverwendung und einer Kostenreduktion in der Entwicklung und Fertigung zu einem ganzheitlichen Ansatz zusammen:

1. Gleichteile: Ein Bauteil wird in mehreren Produkten verwendet. Teilweise kann dazu eine Funktionsübererfüllung des Bauteils notwendig sein, d.h. das Bauteil ist in der Lage mehr als die Einzelanforderungen bei einer Verwendung zu erfüllen. Die damit verbundenen Mehrkosten werden durch verringerte Entwicklungs- und Komplexitätskosten kompensiert. Dieser Ansatz stößt insbesondere bei Massenprodukten aufgrund der hohen Zahl an Produkten und dem geringen Anteil der Entwicklungskosten an den finalen Produktgesamtkosten über die gesamte hergestellte Menge an seine Grenzen.

2. Plattformen: Viele unterschiedliche Produkte haben einen gemeinsamen Kern, oder auch Plattform. Gelingt es diese zu identifizieren, können zahlreiche Produktvarianten aufbauend auf dieser Plattform, die aus mehreren Einzelteilen besteht, durch Hinzufügen von zusätzlichen Modulen oder Systemen relativ einfach generiert werden. Dies erfordert natürlich einheitliche Schnittstellen.

---

[24] Airbus spricht von Mixed Fleet Flying und Cross Over Qualification. Dadurch ist der Wechsel für die Flugcrew zwischen den Modellen in 1 bis 3 Tagen möglich, während ein solcher Wechsel bei Wettbewerbsmodellen bis zu 25 Tage dauern kann [vgl. Airb10].

3. Module und Systeme: Ähnlich wie bei Plattformen werden hier mehrere Einzelteile zusammengefasst, allerdings beschränkt auf Teile für eine bestimmte Funktion (beim Automobil z.B. die Federung). Im Gegensatz dazu umfasst eine Plattform mehrere Funktionsbereiche (beim Automobil z.B. Chassis, Fahrwerk, Motor und Getriebe). Module werden gegenüber Systemen durch ihre physische Zusammengehörigkeit abgegrenzt.

4. Baukasten: Ein Produkt setzt sich aus den mit den Strategien 1 bis 3 erzeugten Einzelteilen (und eventuell einzelner weiterer Individualteile) zusammen. Mittels eines Baukastens werden die zulässigen Kombinationen festgelegt.

Gleichteile und Plattformen werden zusammenfassend auch als Bündelungsstrategien bezeichnet, da hier einzelne Bauteile über verschiedene Produkte verwendet werden sollen, also quasi gebündelt werden. Module und Systeme werden hingegen als Spaltungsstrategien bezeichnet, da hier zunächst eine Zerlegung des Produktes entlang seiner Funktionen erfolgt. Um die Einzelteile, Plattformen, Module und Komponenten mittels eines Baukastens zu einem fertigen Produkt zusammenfügen zu können, sind zwingend standardisierte Schnittstellen und eine stabile Produktarchitektur notwendig. "Das Gerüst des Produktes bleibt stabil, die Elemente im Gerüst lassen sich aufgrund standardisierter und stabiler Schnittstellen nach Bedarf austauschen." [Wild04, S. 39]

Die Umsetzung eines Produktordnungssystems erfolgt in 6 Teilschritten. Zu Beginn erfolgt die Analyse der Marktanforderungen mittels einer Conjoint-Analyse, um die Kundenbedürfnisse und die Zahlungsbereitschaft zu erfassen. Es folgt die Analyse der derzeit am Markt befindlichen Lösungen zur Befriedigung der Kundenbedürfnisse. Anschließend werden die einzelnen Faktoren, die eine Zusammenfassung oder Trennung von Bestandteilen erfordern, gespalten. Ergebnis ist eine Treibermatrix. Darauf aufbauend wird im 4. Schritt das Potenzial zur Zusammenfassung einzelner Elemente analysiert. In Phase 5 werden die Schnittstellen festgelegt, was eine kundenanonyme Vormontage ermöglicht. Dadurch wird der Variantenbestimmungszeitpunkt nach hinten verschoben. Abschließend erfolgt eine Gegenüberstellung des mit Hilfe der Conjoint-Analyse ermittelten Kundenutzens je Modul oder Plattform mit den erwarteten Herstellkosten. Der Kostenanteil sollte ungefähr dem Nutzenanteil entsprechen. Dieses Vorgehen bietet damit eine Möglichkeit zur Ableitung von Zielkosten. Das Vorgehen wurde weitgehend auch für Software-Produktlinien übernommen (vgl. Kapitel 2.2).

Die Ergebnisse dieses Ansatzes sind überzeugend. Basierend auf 21 Anwendungen konnte Wildemann einen deutlichen Kosten- und Bauteilerückgang bei gleichzeitig höherer Variantenzahl feststellen (siehe Abbildung 2).

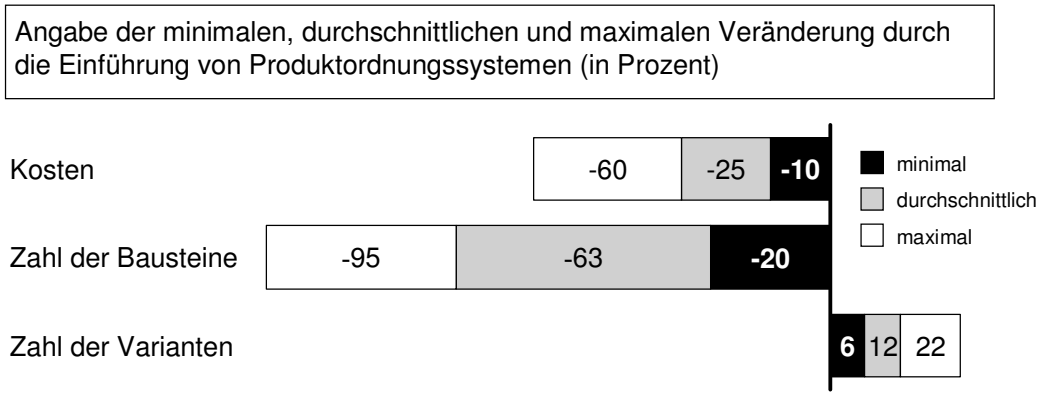

Abbildung 2 – Der Nutzen von Produktordnungssystemen [in Anlehnung an Wild04, S.39]

2.1.3. Auswahl erfolgreicher Anwendungsbeispiele

Neben der Beschreibung der Plattformstrategie von Volkswagen in Kapitel 1.1. sind im Kapitel 2.1.1. einige weitere Beispiel kurz angesprochen worden, die alle einzelne Aspekte einer Produktlinie abdecken, teilweise jedoch unbewusst, also ohne tatsächlich eine Produktlinie zu beabsichtigen. Zum Abschluss dieses Kapitels werden nun noch einige zusätzliche Beispiele kurz beschrieben, welche das Verständnis des Konzeptes durch praktische Anwendungen vertiefen sollen.

Swatch:

Mitte der 1970er Jahre befand sich die Schweizer Uhrenindustrie in einer tiefen Krise. Der ursprüngliche führende weltweite Marktanteil von 30% sank kontinuierlich. Den Massenmarkt schien man an die Konkurrenz aus Asien, insbesondere Japan zu verlieren. Die Japaner hatten den Trend der Zeit erkannt und zahlreiche günstige Quarz-Uhren auf den Markt gebracht. 1983 war der Schweizer Weltmarktanteil auf 9 % gesunken. Mit der Swatch Uhr revolutionierte SMH (Schweizer Gesellschaft für Mikroelektronik und Uhrmacherindustrie) den Uhrenmarkt. Das Erfolgsgeheimnis war eine rigorose Standardisierung und Reduzierung der Bauteile.[25] Dadurch wurde eine vollkommen neue Produktvielfalt ermöglicht, die genau die Kundenbedürfnisse nach einer erschwinglichen,

[25] Eine Swatch Uhr besteht aus lediglich 51 Komponenten, wohingegen normale Uhren aus mindestens 91 Komponenten oder oft sogar deutlich mehr bestehen.

qualitativ hochwertigen und gleichzeitig individuellen Uhr traf. Bis heute wird die Standard Swatch aus Plastik weitgehend unverändert produziert. Es wurden jedoch mehrere Produkterweiterungen hinzugefügt, wie z.B. Irony (Metalluhr), Skin (weltweit dünnster Chronograph) und Snowpass (Uhr mit weltweit nutzbarer Skipass-Funktion). Heute ist Swatch der größte Uhrenkonzern der Welt.

Das Konzept wurde mit dem Smart von Mercedes sogar in die Automobilindustrie übertragen [Jack09, S. 8f. und Jung05, S.181ff.]. Zwar stieg Swatch vor der Fertigstellung des Fahrzeugs aus dem Joint Venture aus, dennoch weist das Grundkonzept, insbesondere bei der weitgehenden modularen Fertigung, bei der die Zulieferer jeweils für ein komplettes Modul verantwortlich sind und die sich alle in direkter Nähe zur Endmontage des Fahrzeugs befinden, die grundlegenden Ideen der Swatch Uhr auf. Allerdings zeigte sich auch bald nach Markteinführung, dass eine direkte Übertragung noch lange nicht zum gleichen Erfolg führen muss, sodass Mercedes Anfang des 21. Jahrhunderts ernsthaft über eine Einstellung der Smart Produktion nachdachte [Jack09, S. 95].

Fertighäuser:
Der klassische Hausbau mit gemauerten Wänden und vor Ort gegossenen Decken ist ein aufwendiger und langwieriger Prozess. Anbieter von Fertighäusern haben dieses Geschäftsmodell radikal verändert. Durch eine weitgehende Standardisierung grundlegender Elemente, wie Wände und Decken, und durch die Verwendung von Holz ist eine Vorfertigung aller Hauselemente abseits der Baustelle möglich. Vielfach kann auch kundenanonym gefertigt werden. Der Kunde hat dennoch die Auswahl aus einer großen Zahl von individuellen Gestaltungsmöglichkeiten. So kann ein Kunde in Deutschland schätzungsweise zwischen mehr als 820.000 Varianten in Ausführung und Ausbau wählen [Fert09]. Als Vorteile ergeben sich niedrigere Kosten und eine deutlich reduzierte Bauzeit vor Ort. In der Regel steht das Haus innerhalb eines Tages oder maximal einer Woche (z.B. wenn die Fertigwände mit Beton ausgegossen werden), der anschließende Innenausbau erfordert nochmals 2 Wochen bis maximal 2 Monate (je nach Komplexität). Die schnelle Bauzeit wirkt sich nicht negativ auf die Qualität aus. Es wird angenommen, dass auch Fertighäuser eine Lebensdauer von 100 und mehr Jahren aufweisen.

LEGO Steine:
Erfunden wurden sie 1958 und ihr Erfolgsgeheimnis ist das Stecksystem, bei dem sich jeder Stein mit jedem beliebigen anderen fest verbindet, aber auch wieder leicht trennen lässt.

Dadurch konnten die eigentlichen Produkte des Unternehmens auf einige wenige Grundbausteine reduziert werden. Bis heute wurde das Grundsystem nicht verändert, wodurch heutige Steine immer noch einwandfrei mit Steinen von 1958 benutzt werden können [Lego09, S. 7]. Aufgrund der Standardbauteile und der standardisierten Schnittstellen ergeben sich die vielfältigen Anwendungsmöglichkeiten bei gleichzeitig später Variantenbestimmung, z.B. durch Bedruckung oder Verpackung (der Zusammenbau erfolgt ja durch den Kunden).[26]

Interessanterweise wurde bei den ersten Versuchen zur Wiederverwendung von Software ein "LEGO Stein" ähnlicher Ansatz gewählt. Man ging davon aus, dass es möglich sein sollte, fertige Codeblöcke weitgehend unverändert wiederverwenden zu können und dabei ein sinnvolles und funktionsfähiges System zu erhalten. Es zeigte sich jedoch, dass dieser Gedanke gerade wegen seiner unglaublichen Einfachheit nur auf Kinderspielzeug anwendbar war [vgl. Bosc00, S.7].

In der Literatur finden sich zahlreiche weitere erfolgreiche Beispiele, von denen einige kurz mit Quellenangabe aufgeführt werden:

- Toyota hat es früh verstanden, durch strategische Wiederverwendung die Anzahl verschiedener Teile drastisch zu reduzieren [Ward95].
- Einweg-Fotokameras wurden von Fuji erfunden. Aber Kodak hat es durch Verwendung einer Produktlinie verstanden, neue Produkte deutlich schneller und kostengünstiger zu realisieren als der eigentliche Erfinder Fuji und damit am Ende den Massenmarkt zu erobern [Robe99 und Clar95].
- Sony hat die Einführung des Walkman als Produktlinie gestaltet. Im Zeitraum zwischen 1980 und 1990 wurden mehr als 160 verschiedene Varianten auf den Markt gebracht, begleitet von 4 wichtigen technischen Innovationen. Und obwohl MP3 Player heute die Marktführung übernommen haben, ist der Name Walkman so bekannt, dass Sony seine MP3 Player immer noch als Walkman bezeichnet [Meye92, S. 3]
- Black and Decker, ein amerikanischer Werkzeughersteller, hatte in den 1970er Jahren ein sehr komplexes Produktportfolio mit mehreren Hundert Produkten. Diese

[26] LEGO stellt alle Reifen für seine Modelle selbst her. Insgesamt werden jährlich ca. 306 Millionen Reifen hergestellt. Dies bedeutet, dass LEGO als der weltgrößte Reifenhersteller angesehen werden kann [Lego09, S. 18].

verwendeten mehr als 30 verschiedene Motoren, 60 verschiedene Gehäuse und Dutzende Steuerungen. Durch die Einführung einer Produktfamilie wurde die Standardisierung deutlich gesteigert. Die Produktionskosten wurden um 50% gesenkt, wodurch ein dominanter Marktanteil erreicht werden konnte [Meye92, S. 4].

2.2. Software Produktlinien – ein bekanntes Thema im Bereich des Software Engineering

2.2.1. Einleitung zu Softwareproduktlinien

Wie bereits in Kapitel 1.2. angesprochen nimmt die Bedeutung der Software bei neuen Produkten kontinuierlich zu. Dies bedeutet einerseits, dass die Hersteller vieler Produkte, die ihr Wissen bisher hauptsächlich im Bereich der mechanischen Fertigung sahen, zu einem Umdenken und neuen Lernprozess gezwungen sind. Ebenso führt dieser allmähliche Wandel zu Schwierigkeiten hinsichtlich der Produktqualität und einer Verlängerung der Produktentwicklungszeiten, da die Unternehmen die notwendigen SW-spezifischen Prozesse oft noch nicht ausreichend beherrschen.

"The 1990s saw the widespread application of manufacturing process principles to the development of software" [Jone02, S. 388]. So bspw. auch den Versuch, das Konzept der Produktlinie auf den Softwarebereich zu übertragen. Dies erscheint insbesondere sinnvoll, da Studien gezeigt haben, dass zwischen 45% und 85% einer neuen Software auf Funktionen beruhen, die bereits in früheren Produkten enthalten waren [With96, S. 3]. Besonders klar wird die Bedeutung der Wiederverwendung in sich schnell entwickelnden Märkten. So kann bspw. im Bereich der Web Applications eine Wiederverwendungsrate von ca. 70% beobachtet werden [Jone07, S. 184]. Nur durch die intensive Nutzung von bereits entwickelter SW ist die hohe Entwicklungsgeschwindigkeit bei Web Applications überhaupt möglich.

„Reuse, as a software strategy for decreasing development costs and improving quality, is not a new idea, and software product lines definitely involve reuse – reuse, in fact, to the highest order.” [Clem02, S. 11] Aber Software lässt sich eben nicht so einfach wiederverwenden. Für eine erfolgreiche Wiederverwendung sind entsprechende Techniken und Verfahren in der Software Entwicklung notwendig, die erst kürzlich entwickelt wurden. In Erweiterung der Ideen von [Atki02, S.4] spielten dabei insbesondere die folgenden Entwicklungen eine entscheidende Rolle:

- **Objektorientierte Programmierung:** Methoden wie Vererbung und dynamische Bindung sind wichtige Grundvoraussetzungen für eine tatsächliche Wiederverwendung und wurden kurz nach ihrer Einführung vielfach als Silver Bullet angesehen. Allerdings stellte sich heraus, dass dieses Konzept, u.a. aufgrund der engen Bindung an Compiler, bei vielen wichtigen objektorientierten Sprachen nicht ausreichend ist. So sind bei C++ die Objekte nicht wirklich selbständig, da sie in das ausführbare Programm integriert werden. Dadurch wird die direkte Wiederverwendbarkeit erschwert. Denn diese würde eine einzelne Ausführbarkeit der Objekte vorraussetzen. Dieser Schritt wurde allerdings erst durch den komponentenbasierten Ansatz etabliert. Bei C++ ist jeweils eine erneute Kompilierung bei Wiederverwendung in einer anderen Umgebung erforderlich.

- **Komponenten:** „Components represent the smallest coherent packages of software that make sense as standalone, reusability entities“ [Atki02, S.5]. Komponenten stellen eine wichtige Erweiterung der objektorientierten Programmierung dar, denn damit wurden die Bausteine stärker gekapselt und tatsächlich unabhängig wiederverwendbar. Als wichtige aktuelle Technologien hierzu sind zu nennen: Enterprise JavaBeans, DCOM und die Erweiterung COM+ und schließlich CORBA.[27]

- **Architectural styles und design pattern:** Diese erweitern den komponentenorientierten Ansatz auf die nächste Granularitätsstufe, da hier die Strategien zur Verbindung der einzelnen Komponenten (sogenannte basic building blocks) ergänzt werden. Die Darstellung kann z.B. mittels UML erfolgen.

- **Reference Architecture:** Eine Referenzarchitektur liefert einen Vorschlag und grundsätzliche Hilfestellungen für die Entwicklung von Architekturen für eine bestimmte Domäne.[28] Aufbauend auf Erfahrungen stellt die Referenzarchitektur eine Sammlung der besten Umsetzungen für eine bestimmte Aufgabe dar. Ein wichtiges Ziel dieser Referenzarchitekturen ist ein hohes Maß an Communality, also der intensiven Verwendung von gleichen Elementen. Im Rahmen von SWPL spielt die Architektur eine entscheidende Rolle, allerdings ist der Ansatz zur Realisierung einer

---

[27] Für weitere Details sei auf [Clem02, S. 63] verwiesen.

[28] Die Java EE (Java Plattform Enterprise Edition) Architektur ist ein Beispiel für eine weit verbreitete Referenzarchitektur, die für zahlreiche Unternehmenssysteme und den dabei häufig anzutreffenden Problemen vorgefertigte Lösungsvorschläge bereithält.

SWPL deutlich umfangreicher. Eine Betrachtung der Architektur alleine reicht für eine erfolgreiche strategische Wiederverwendung nicht aus.

- **Object oriented Frameworks:** Auch bei Frameworks steht die Zusammenstellung von einzelnen Komponenten im Vordergrund, mit deren Hilfe neue Software durch Wiederverwendung einfacher Komponenten mit höherer Qualität entwickelt werden kann. In [Care02, S. 1] sind Frameworks definiert als „a set of components working together so they address a number of problems in one or more domains". Dabei gehen Frameworks über Class Libraries und Design Patterns (für Details siehe [Gamm95] und [Coop03]) hinaus. Es ist sozusagen der nächste Reifegrad der Wiederverwendung, da Frameworks immer mehrere Libraries und Design Patterns verbinden und gleichzeitig etwas über die jeweilige Implementierung aussagen. Wichtig ist bei Frameworks allerdings die klare Fokussierung auf einen spezifischen Bereich, um die Komplexität im Griff zu behalten und durch das Framework einen echten Mehrwert für eine einfachere Realisierung von Software zu erzielen. Insgesamt stellen Frameworks daher ein wichtiges Werkzeug für die Wiederverwendung von Software Komponenten dar. Für eine übergreifende strategische Wiederverwendung greift der Ansatz alleine allerdings zu kurz.

- **Plattformen:** Plattformen stellen eine Erweiterung einer standardisierten Infrastruktur dar und weisen damit einen höheren Reifegrad bei der Software Wiederverwendung auf. Dies geht allerdings gleichzeitig mit einer strengeren Anforderung an die Stabilität der Rahmenbedingungen einher, um eine Wiederverwendung zu ermöglichen. Allerdings abstrahieren diese Ansätze meist von der Notwendigkeit zur Variabilität, also der späteren Anpassung von Elementen und Entscheidungen. Bei Plattformen wird stärker angenommen, dass einmal entwickelte Komponenten in ihrer bestehenden Form über Standardschnittsellen weiterverwendet werden können.

All diese Methoden sprechen wichtige Aspekte zur strategischen Wiederverwendung an, alleine bilden sie aber immer nur Teilaspekte ab. So betrachtet z.B. das Component-Based Development die Entwicklung von Komponenten, die wieder verwendbar gestaltet werden. Eine strategische Betrachtung der notwendigen Variabilität erfolgt allerdings nicht bei der ersten Erstellung der Komponente, sondern erst in dem Moment, indem diese tatsächlich wiederverwendet werden soll. Außerdem werden so veränderte Komponenten im weiteren Verlauf meist auch unabhängig als eigenständige Varianten gewartet und weiterentwickelt.

Referenz-Architekturen und Frameworks dienen zwar dazu, in verschiedenen Umgebungen und Systemen erneut eingesetzt und entsprechend angepasst werden zu können, aber auch hier fehlt eine frühzeitige und ganzheitliche Sicht auf die später angestrebte Wiederverwendung. SWPL als ein übergreifendes Framework, wie z.B. bei [Clem02 und Pohl05] vorgestellt, sind hingegen deutlich umfassender und berücksichtigen bzw. erweitern all diese oben genannten Aspekte [vgl. dazu insb. Clem02, S. 11ff.]. Dieser umfassenden Betrachtungsweise soll in der vorliegenden Arbeit gefolgt werden. Daher wird die folgende Definition zu Grunde gelegt:

"A software product line is a set of software-intensive systems sharing a common managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way" [Clem02]. Es fällt auf, dass die Änderungen gegenüber der schon bekannten allgemeinen Produktliniendefinition aus Kapitel 2.1.1 gering sind. Auch bei SWPL stehen die optimale Nutzung der Gemeinsamkeiten und die Ausgestaltung der Variabilitäten im Mittelpunkt. Allerdings erfordert das Software Engineering in vielen Bereichen eine ganz andere Herangehensweise als bei klassischer Hardware. Daher ist eine reine Beschreibung der Idee zur tatsächlichen Umsetzung nicht ausreichend. Es werden vielmehr konkrete Ansätze zur Realisierung von Software-Produktlinien benötigt. Inzwischen finden sich in der Literatur zahlreiche, weitgehend ähnliche Ansätze, von denen die wichtigsten kurz vorgestellt werden sollen. Für eine detaillierte Betrachtung siehe [Ding09, S. 43ff.]:

- **Domain Engineering:** Mittels Feature-Modellen werden die erwarteten Fähigkeiten des Produktes aus Nutzersicht erfasst. Dadurch wird sowohl die Abstimmung zwischen Produktmanagement als Vertreter des Kunden und der Entwicklung vereinfacht, als auch die Erkennung von Wiederverwendungspotenzial erleichtert. Dies wird als Feature Modellierung bezeichnet. Durch Instanziierung des Feature-Modells können schließlich einzelne Varianten der Produktlinie erstellt werden.
- **Entscheidungsorientierter Ansatz:** Die Eigenschaften der Produktlinie werden in einem sogenannten Domänen-Modell erfasst. Dabei werden alle technischen und nicht-technischen Abhängigkeiten und Einschränkungen erfasst und zu einem Entscheidungsmodell zusammengefasst. Dieses Modell enthält alle möglichen Entscheidungen, die bei der Auswahl einer Variante zulässig sind. Durch das Treffen von expliziten Entscheidungen erfolgt die Zusammenstellung einer Variante.

- **Architekturbeschreibungssprachen:** Beim Thema der strategischen Wiederverwendung spielt insbesondere die gewählte SW-Architektur eine wichtige Rolle, da mit dieser viele grundlegende Entscheidungen getroffen werden, die die spätere SWPL beeinflussen. In der Literatur findet sich daher eine Vielzahl an Methoden zur Erstellung einer geeigneten SW Architektur. Allerdings erfordert der SWPL-Ansatz einige Besonderheiten, die aktuell von fünf Architekturbeschreibungsmethoden unterstützt werden: COPA, FAST, FORM, KobrA und QADA. Ein umfangreiches Evaluations-Framework zur Beschreibung und zum Vergleich dieser Ansätze findet sich bei [Mati04]. Zur weiteren Vertiefung der Architekturbeschreibungssprachen sei zusätzlich auf [Dash01] verwiesen.

- **UML-basierte Ansätze:** Einige Autoren haben konkrete Konzepte entwickelt, die den Industriestandard UML nutzen, um damit die Wiederverwendung von SW Requirements und Architekturen zu ermöglichen. Die Ansätze unterscheiden sich grundsätzlich nicht von den bereits oben genannten Vorgehen mit der Ausnahme, dass UML explizit als Modellierungswerkzeug vorgesehen ist. Für alle UML basierten Designmethoden und Prozesse, wie z.B. Use Cases, State Machines, Interaction Modelling, bieten diese Ansätze geeignete Konzepte und Techniken, um Einzelentwicklungen in SWPL umzuwandeln. Dabei werden die wichtigen Themen, wie Communality und Variability explizit in die UML Modellierung mit einbezogen. Für weitere Details wird auf den PLUS Ansatz (Productline UML-based Software Engineering) von Gomaa [Goma04] oder auch [Atki02] verwiesen.

Die Kombination von etablierten Methoden des Software Engineering und klassischer Produktlinien erzielte Erfolge, die in den nächsten Kapiteln im Detail vorgestellt werden. Zunächst erfolgt jedoch eine genauere Begriffsdifferenzierung.

2.2.2. Unterscheidung zwischen SW-Produktfamilien und SW-Produktlinien

In der Literatur werden die beiden Begriffe SW-Produktfamilie und SW-Produktlinie teilweise synonym, teilweise differenziert verwendet [vgl. Clem02, S. 15]. Aus der geschichtlichen Entwicklung der Begriffe heraus lässt sich eine Differenzierung herleiten, die Withey bereits 1996 in seinem Aufsatz "Investment Analysis of Software Assets for Product Lines" [With96] aufzeigt. So wird der Begriff Produktlinie insbesondere in der Industrie benutzt, um gleichartige Produkte aus der Sicht des Marktes zu kennzeichnen. Eine Produktlinie beschreibt:

- "1. a group of products that provide a core benefit yet differ along attributes which affect the buying behavior of different customer groups
- 2. a group of products that are closely related, either because they function in a similar manner, are sold to the same customer groups, are marketed through the same types of outlets or fall within given price ranges" [With96, S.58]

Hier zeigt sich der sehr klare Marktfokus einer Produktlinie. Die technische Realisierung der Wiederverwendung spielt nur eine nachrangige Rolle, im Gegensatz zu einer Produktfamilie. Diese werden häufig bezeichnet als das Set von:

- "products that share a common platform but have specific features and functionality required by different sets of customers" [Meye92, S.3]
- "different products that can be produced from a common design, shared assets, and an application engineering process" [With96, S.58]

Hier stehen die technischen Gemeinsamkeiten der Produkte, die verantwortlich für die mögliche Wiederverwendung sind, im Vordergrund.

[Parn09] unterscheidet klarer zwischen einer Produktlinie als kommerzielles Konzept, das durch Marktanalysen motiviert wird und durch die Marketingabteilung eines Unternehmens festgelegt wird. Erneut wird auf die Marktseite abgezielt. Die Produktlinie stellt dementsprechend allein die Abdeckung von individuellen Kundenwünschen mittels unterschiedlicher Produkte dar. Dies sagt allerdings noch nichts über die tatsächliche Realisierung der Produktlinie aus. Laut Parnas ist es in vielen Fällen nach einer genauen Analyse sinnvoll eine Produktlinie als Program Family zu designen. Den Begriff Program Family prägte Parnas bereits 1976. „Program families are defined (analogously to hardware families) as sets of programs whose common properties are so extensive that it is advantageous to study the common properties of the programs before analyzing individual members“ [Parn76, S.1]. Programmfamilien sind demnach Zusammenstellungen von Programmen, die man nicht unabhängig entwickeln sollte. Vielmehr sollte aufgrund des hohen Anteils an ähnlichen Bestandteilen zunächst allgemein überlegt werden, wie die Entwicklung am Besten organisiert werden könnte.

Wichtig ist zu beachten, dass bei all diesen Definitionen eine Produktlinie sowohl mehrere Produkt-/Programmfamilien beinhalten kann, als auch selbst Bestandteil von mehreren

Produkt-/Programmfamilien sein kann [Czar00, S. 37]. Anhand eines kurzen Beispiels soll dies verdeutlicht werden:

Bei einem Automobilhersteller, der mehrere Produktreihen anbietet, um unterschiedliche Kundensegmente anzusprechen, könnte man bei diesen Produktreihen von Produktlinien sprechen, da der Markt im Vordergrund der Produktdifferenzierung steht. Sollte das Unternehmen jedoch Gleichteile für mehrere Produkte verwenden, dann könnte man von einer Produktfamilie sprechen. Denn hier würden Kosteneinsparungen im Vordergrund stehen und im Idealfall bemerkt der Kunde von diesem Vorgehen nichts. Ein Gleichteil, also beispielsweise die Steuerung der Klimaautomatik, könnte nun in mehreren Baureihen, sprich Produktlinien, enthalten sein. Gleichzeitig kann eine Produktlinie die Basis für den Aufbau einer Produktfamilie sein, wenn aus den bestehenden Komponenten eine gemeinsame Basis erstellt werden soll.

Weiterhin zeigt sich eine regionale Begriffsdifferenzierung. Während man in der nordamerikanischen Literatur meist den Begriff Product Lines vorfindet, wird in Europa öfter von Produktfamilien gesprochen [Pohl05, Preface]. Dies spiegelte sich auch in den Namen der beiden Konferenzserien zum Thema Produktlinien oder -Familien in den USA (software product line conference series) und Europa (product family engineering workshop series) wider. Im Jahre 2004 wurden beide als Software Product Line Conference Series (SPLC) zusammengefasst. Die zunehmende Nutzung des Begriffes Produktlinie kann man zusätzlich an der weitverbreiteten operativen Bezeichnung Software Product Line Engineering erkennen. Der Begriff bezeichnet die Gesamtheit aller Ansätze, die die Wiederverwendung von Software mittels Nutzung von Software Plattformen, sprich dem geplanten, strategischen Reuse, und individualisierte Massenfertigung, also gemanagte Variabilität, zum Ziel haben [Pohl05, S.14]. Zahlreiche Autoren verwenden die beiden Begriffe daher inzwischen gleichbedeutend [vgl. Clem02, S. 14f.].

Es ist demnach zwar möglich, eine Begriffsdifferenzierung zu erreichen. Aufgrund der heutigen fast synonymen Verwendung der beiden Begriffe, ist der Nutzen allerdings fragwürdig. Im Rahmen dieser Arbeit wird daher der heute vielfach benutze Begriff Produktlinie (auch entsprechend des Titels der Arbeit) verwendet. Gemeint ist damit nicht allein die marktseitige Betrachtung, sondern die umfassende Betrachtung der strategischen Wiederverwendung von (Software) Bestandteilen bei mehreren Produkten. Sollte eine

Differenzierung im weiteren Verlauf sinnvoll oder sogar notwendig sein, wird explizit darauf hingewiesen.

2.2.3. Variabilität in Software

Nach [Svah00, S. 147] bezeichnet Variabilität in Software "the ability to change or customize a system". Die Möglichkeit zur Veränderung durch den Hersteller sollte bei jeder Software gegeben sein. Ansonsten könnte bspw. bereits eine gesetzliche Änderung[29] einen hohen Anpassungsaufwand nach sich ziehen. Aber auch technischer Fortschritt kann Anpassungen z.B. bei Schnittstellen notwendig machen. Wenn ein Unternehmen lediglich ein Produkt anbietet und dieses entsprechend der veränderten technischen oder gesetzlichen Bedingungen über Zeit anpasst, spricht man von Variability in Time [Pohl05, S. 65].[30] Diese Art der Variabilität ermöglicht eine Evolution der Software, also eine langfristige Verbesserung oder Optimierung.

Das Customizing, also die Anpassung eines Systems an die Kundenwünsche, kann nun wieder über den Zeitablauf erfolgen, d.h. es werden zwar Änderungen durchgeführt, aber es wird zu einem gegebenen Zeitpunkt immer nur ein Produkt angeboten[31], oder es werden mehrere Produkte gleichzeitig angeboten, wobei jedoch alle auf einem einheitlichen Kern basieren. Im letzten Fall spricht man von Variability in Space [Pohl05, S. 66]. Demnach ist jede Individualisierung eines Neuwagens über die vom OEM angebotenen Sonderausstattungen eine Variability in Space. Diese Form von Variabilität ist das entscheidende Merkmal von Produktlinien.

---

[29] Beispielweise könnte die Änderung der Mehrwertsteuer die Nutzung eines Warenwirtschaftssystem unmöglich oder zumindest deutlich aufwendiger machen, solange der Prozentsatz innerhalb des Programms nicht angepasst worden ist.

[30] Im Automobilbereich war eine solche Entwicklung durch die zunehmende Verbreitung von Musik im MP3 Format zu beobachten. Fast alle Hersteller bieten inzwischen eine Möglichkeit zum Anschluss eines externen MP3 Abspielgerätes, oder die Radiogeräte selbst sind in der Lage MP3 Formate über des CD oder DVD Laufwerk abzuspielen. Für den einzelnen OEM oder auch Zulieferer bedeutete diese Entwicklung allerdings einen unterschiedlich hohen Anpassungs- und Entwicklungsaufwand. Einige Hersteller hatten schon früh offene Schnittstellen in ihre Systeme eingebaut (z.B. als Cinch- oder Mini-Klinken-Eingang) wodurch nur geringe Anpassungen notwendig waren. Andere OEMs benutzen jedoch proprietäre Schnittstellen, die in vielen Fällen ein neues Steuergerät notwendig machten und dann auch nur mit einem bestimmten Abspielgerät funktionierten.

[31] Die kürzlich eingestellte G-Klasse von Mercedes wurde äußerlich weitgehend unverändert über einen Zeitraum von 29 Jahren gebaut. Im Allgemeinen finden große Modellwechsel in der Automobilindustrie durchschnittlich alle 4 bis 8 Jahre statt [McKi06, S. 8]. Um auch dazwischen Innovationen in die bestehenden Produkte einführen zu können, nutzen zahlreiche Automobilhersteller Modelljahre. Innerhalb eines Modelljahres finden sich nur sehr geringe bis gar keine Unterschiede, zwischen den Modelljahren kann es jedoch (obwohl es sich immer noch um das gleiche Produkt handelt) zu größeren Veränderungen kommen. In den USA wird ein Auto daher neben dem eigentlichen Namen auch immer mit dem Modelljahr bezeichnet. Zusätzlich nutzen die OEMs auch die sogenannte Modellpflege. Dabei werden umfangreichere Veränderungen eingeführt, die stärker das äußere Erscheinungsbild des Fahrzeugs betreffen. Es handelt sich aber immer noch um das gleiche Basisprodukt.

Es gilt: "Variabilität ist das zentrale Konzept in allen Artefakten von Software-Produktlinien" [Bühn04, S. 13]. Die Variabilität einer Produktlinie wird über Variationspunkte und Varianten abgebildet. "Ein Variationspunkt gibt die Stelle an, an der die Auswahl einer oder mehrerer Varianten möglich ist. Eine Variante beschreibt konkrete Ausprägungen in Bezug auf einen oder mehrere Variationspunkte" [Bühn04, S.17]. Variationspunkte beschreiben zumindest teilweise ein reales Objekt, das verschiedene Ausprägungen annehmen kann. Oder anders ausgedrückt: Variationspunkte bezeichnen eine Funktion, die im Produkt unterschiedlich ausgeprägt sein kann. Als Beispiel könnte man sich die Bedienung der Fensterheber in einem Automobil vorstellen. Die möglichen Ausprägungen der Funktion sind die Varianten, also z.B. elektrisch oder manuell. Interessant ist, dass sich Variationspunkte im Zeitablauf meist nicht verändern, die Varianten aufgrund technischer Veränderungen jedoch schon. Dies führt zu der Schwierigkeit, alle relevanten Variationspunkte bereits bei der Konzeption des Produktes vollständig zu berücksichtigen. Das bedeutet, dass hinsichtlich des Fensterheberbeispiels auch dann ein Variationspunkt eingeführt werden muss, wenn bei der Entwicklungsphase lediglich eine manuelle Bedienung möglich ist, aber später mit zusätzlichen Bedienmöglichkeiten gerechnet wird. Dies verdeutlicht die Bedeutung einer vollständigen systematischen Erfassung der Variabilität im Rahmen einer Produktlinie. Weiss hat diesen Zusammenhang bereits 1999 in seiner Definition von Variabilität als "an assumption about how members of a family may differ from one another" [Weis99, S.23] verdeutlicht

Es gibt viele unterschiedliche Arten von Variabilität in SWPL, die sich nach [Bühn04, S.14ff.] in folgende Typen gliedern lassen:

- **Variabilität in Features:** Produkte einer SWPL unterscheiden sich hinsichtlich der angebotenen Funktionen.
- **Variabilität in Abläufen:** Trotz gleicher Funktionen können Produkte in Prozessschritten variieren.
- **Variabilität im Datenumfang:** Verschiedene Produkte enthalten unterschiedliche Datenobjekte und Attribute.
- **Variabilität im Datenformat:** Produkte verwenden unterschiedliche Datenformate.
- **Variabilität im Systemzugang:** Funktionen können mehrere Zugangs- bzw. Nutzungsarten anbieten.

- **Variabilität in Schnittstellen:** Einerseits ermöglichen SWPL die individuelle Ausgestaltung der Benutzerschnittstelle für jedes Produkt, andererseits können aber auch mehrere Systemschnittstellen implementiert werden.
- **Variabilität in Qualität:** Produkte einer SWPL können unterschiedliche Qualitätsaspekte bieten.

Die einzelnen Variationspunkte und Varianten können untereinander Abhängigkeiten aufweisen, die berücksichtigt werden müssen. Wenn ein Variationspunkt vorhanden ist, dann kann für die einzelnen Varianten nach [Pohl05, S. 77f.] gelten:

- **Optional:** Die Variante kann vorhanden sein oder nicht
- **Alternative:** Eine und nur eine der Varianten muss vorhanden sein
- **Mandatory:** Diese Variante muss zwingend vorhanden sein

Ebenso können sich Variationspunkte und Varianten gegenseitig bedingen. So benötigt bspw. ein elektronisches Fahrstabilitätsprogramm die Daten mehrerer anderer Systeme, wie Antiblockiersystem, Querbeschleunigung, usw. Aber sie können sich auch ausschließen, z.B. bietet ein Fahrzeug, das allein über eine Chipkarte geöffnet und gestartet werden kann, eventuell keine Möglichkeit den Wagen zusätzlich über einen Schlüssel zu starten.

Weiterhin wird in der Literatur zwischen interner und externer Variabilität unterschieden. Externe Variabilität ist für den Kunden sichtbar. Demnach ist das oben angesprochene Fensterheberbeispiel eine externe Variabilität. Interne Variabilität ist für den Kunden unsichtbar, da sie keine Relevanz für die Kaufentscheidung hat. Ein Beispiel für interne Variabilität ist die Verwendung von unterschiedlichen Motorsteuergeräten in einzelnen Ländern, um die dort geltenden gesetzlichen Abgasvorschriften einzuhalten.

Für den erfolgreichen Einsatz einer Software-Produktlinie ist es wichtig, die Variabilität möglichst früh und vollständig zu erfassen. Es haben sich verschiedene Ansätze zur expliziten Repräsentation von Variabilität entwickelt [siehe Jaco97, Copl98 und Svah01]. Einerseits kann die Variabilität direkt in den bestehenden Entwicklungsartefakten als zusätzliche Information berücksichtigt werden. Andererseits bietet die Erfassung in einem separaten Variabilitätsmodell den Vorteil der vollständigen Sicht auf die Variabilität von einer Quelle aus. Hierzu hat [Pohl05, S. 74f.] den orthogonalen Variabilitätsbegriffs geprägt.

2.2.4. Ansätze zur Bewertung der Reife von Prozessen in Unternehmen

Bei der SW Erstellung zeigte sich in empirischen Analysen, dass den dabei verwendeten Prozessen eine hohe Bedeutung für die Produktivität und Qualität des Endproduktes zukommt [Jones07, S. 163 f.]. Unternehmen mit definierten Prozessen werden tendenziell, insbesondere bei sehr komplexen Projekten, eher eine höhere Produktivität und eine bessere Qualität aufweisen. Natürlich können sehr gute Mitarbeiter selbst in einem kleinen aber chaotischen Team ein gutes Resultat erzeugen, allerdings steigen die Chancen dafür bei einer guten Prozessunterstützung deutlich an [Jone07, S. 164]. Insbesondere in großen Unternehmen ist aufgrund der dortigen Projektkomplexität ein umfassendes Prozessmanagement unerlässlich.

Bei Produktlinien haben standardisierte und strikt eingehaltene Prozesse aufgrund der mehrfachen Verwendung von einzelnen Bestandteilen eine noch größere Bedeutung als bei Einzelentwicklungen. Individuelle Einzelveränderungen außerhalb der regulären Prozesse können signifikante Auswirkungen haben, z.B. wenn solche Veränderungen später zu unerklärbaren Fehlern führen. Daher wird im Folgenden detaillierter auf die existierenden Möglichkeiten eingegangen, um die Prozessreife in einem Unternehmen zu bewerten und ggfs. zu verbessern. Weiterhin wird das hier beschriebene Vorgehen zur Reifegradbestimmung später in abgewandelter und vereinfachter Form im Rahmen des e-FACTS Modells angewendet (Teil B Kapitel 5.1).

2.2.4.1. CMMI

1986 gab das US-amerikanische Department-of-Defense beim Software Engineering Institute (SEI) der Carnegie Mellon University ein Modell zur Bewertung der Prozessreife in Unternehmen in Auftrag. Daraus entstand 1991 das Capability Maturity Model (CMM). Dieses Verfahren misst über ein standardisiertes und umfangreiches Assessment den Reifegrad der Prozesse eines Unternehmens. Im Jahre 2000 wurde der Ansatz zum CMMI erweitert. Das I steht für Integration, da hier der Versuch unternommen wurde, alle bisherigen Modelle zu integrieren. Das Ergebnis ist ein sehr umfassendes, aber leider auch komplexes und zeitaufwendiges Verfahren. Ein komplettes CMMI Appraisal umfasst mehr als 450 detaillierte Fragen [McKi06, S.20], die unabhängig von mehreren Personen zu beantworten sind.

CMMI stuft Prozesse anhand einer 5-stufigen Skala ein:[32]

- **Initial:** Hier sind die Prozesse entweder gar nicht, oder nur gering definiert. Entscheidungen und Prozesse finden adhoc und eher chaotisch statt. Der Erfolg bei einem solchen Vorgehen hängt maßgeblich von der Qualität der einzelnen Mitarbeiter ab, auch darf das Team nicht zu groß sein.
- **Managed:** Ein Unternehmen erreicht diese Stufe, wenn es ein typisches Projektmanagement etabliert hat. Für jedes Projekt sind alle notwendigen Prozesse definiert, zwischen einzelnen Projekten kann es aber deutliche Unterschiede geben.
- **Defined:** Ein Unternehmen steigert sich auf Stufe 3 (entspricht ungefähr ISO 9001), wenn es gelingt von einzelnen Projekten zu einer Gesamtbetrachtung des Unternehmens zu gelangen. Es sind einheitliche Prozesse für die gesamte Unternehmung definiert.
- **Quantitatively managed:** Bei Stufe 4 ist es dem Unternehmen gelungen, konzernweit einheitliche Metriken und Kennzahlen einzuführen und mittels einer übergreifenden Datenbank sinnvoll zu nutzen. Daraus ergibt sich eine deutlich verbesserte Entscheidungsunterstützung.
- **Optimizing:** Um die letzte Entwicklungsstufe zu erreichen, ist es notwendig zusätzlich kontinuierliche Verbesserungen in die Prozesse zu integrieren. Dies umfasst die systematische Analyse noch bestehender Fehler und die systematische Selektion von Verbesserungsmaßnahmen.

Ein großer Vorteil von CMMI, insbesondere auch gegenüber der ISO 9001, ist die direkte Ableitung von Verbesserungsmöglichkeiten. Daher kann CMMI hilfreich für das Qualitätsmanagement im Software- und Systementwicklungsbereich sein. Zahlreiche Unternehmen haben durch eine Fokussierung ihrer Verbesserungsmaßnahmen auf CMM Erfolge erzielt [vgl. Paul95]. Für SWPL ist inbesondere relevant, dass "process improvement has a significant relationship to product line practice" [Jone02, S. 389], wobei diese Beziehung auch nicht überbewertet werden sollte, da "just because, say, your organization has institutionalized the CMMI process area of Configuration Management, it does not mean that

[32] Für eine detaillierte Beschreibung des Modells siehe [SEI00].

you have mastered the practice area of Configuration Management for product lines" [Jone02, S. 392].

2.2.4.2. SPICE

Ein ähnliches, wenn auch nicht ganz so weit verbreitetes, Verfahren ist SPICE (Software Process Improvement and Capability Determination). Dieses wurde 1998 als Vorabversion veröffentlicht und Anfang 2006 durch einen internationalen Standard (ISO 15504) ersetzt [ISO98]. Auch SPICE verfolgt 2 Ziele, einerseits die Verbesserung der eigenen Prozesse und zweitens die Bestimmung der Prozessfähigkeit von Lieferanten. Die Bewertung erfolgt hier ebenfalls in einem Assessment. Dabei werden alle relevanten Prozesse zerlegt in Zweck, Ergebnisse und Base Practices. Base Practices beschreiben die Aufgaben und Maßnahmen, die zum Erreichen des Prozesszwecks und der Prozessergebnisse erforderlich sind [VDA07, S. 37]. Für jeden Prozess wird individuell ein Reifegrad mittels einer 6-stufigen Skala bestimmt, aus denen sich ein Stärken-Schwächen-Profil ergibt. Schließlich erfolgt eine Gesamteinstufung des Unternehmens. Für Details wird auf [VDA07, S. 26f.] verwiesen.

- **Level 0 (incomplete):** Es werden nicht alle Prozesse durchgeführt, oder nicht alle Prozesse erfüllen ihren Zweck. Zumindest ein wichtiges Arbeitsprodukt fehlt.
- **Level 1 (performed):** Alle Dokumente und Prozesse sind vorhanden und diese erfüllen ihren Zweck.
- **Level 2 (managed):** Alle Prozesse werden systematisch geplant, gesteuert und kontrolliert, zumindest auf Projektebene.
- **Level 3 (established):** Es bestehen organisationsweite, einheitliche Standardprozesse. Zusätzlich ist sichergestellt, dass die Prozesse auch vollständig angewendet werden.
- **Level 4 (predictable):** Durch statistische Messung und Kontrolle der Prozesse wird ein Ablauf innerhalb klar definierter Grenzen sichergestellt.
- **Level 5 (optimizing):** Prozessinnovation und –Optimierung sind gewährleistet. Es findet eine ständige Kontrolle statt und notwendige Anpassungen sind aufgrund guter Vorbereitung zeitnah umzusetzen.

Im Jahr 2005 wurde SPICE durch die Automotive Special Interest Group[33] unter Führung der Herstellerinitiative Software[34] um Automotive SPICE erweitert. Dieses Modell hat das Ziel, die Leistungsfähigkeit der Entwicklungsprozesse von Steuergerätelieferanten in der Automobilindustrie zu bewerten. Die Motivation stammt aus der bereits in der Einleitung dieser Arbeit angesprochenen zunehmenden Bedeutung und Komplexität der Software im Automobil (siehe Kapitel 1.2.). Gegenüber SPICE wurden bei der Betrachtung der Beziehung zwischen OEM und Zulieferer einige Prozesse hinzugefügt. Ebenso wurden die Prozessdefinitionen, also der Zweck, die Ergebnisse und die Base Practices, an die Besonderheiten der Automobilindustrie angepasst.

2.2.4.3. Auswirkungen auf die vorliegende Arbeit

Trotz der starken Fokussierung von CMMI und SPICE auf die Prozessperspektive sind beide Ansätze bei der Evaluierung der ökonomischen Aspekte einer Produktlinie sinnvoll. Sie ermöglichen eine Abschätzung der derzeit gegebenen Qualität bei der Software- und Systementwicklung. Allerdings sind für eine Analyse der grundsätzlichen Eignung des Unternehmens für die Einführung einer Produktlinie weitere Fragenkategorien notwendig. Ebenso erfordern beide Ansätze einen erheblichen Zeitaufwand, der mit der grundsätzlichen Zielrichtung dieser Arbeit, einer einfachen und zeitnahen Analyse, die ein Unternehmen ohne fremde Hilfe durchführen kann, nicht vereinbar ist. Daher wird in Teil B Kapitel 5.1. ein Fragenkatalog vorgestellt, der sich zwar grundlegend an den Ideen von CMMI und SPICE orientiert, diese aber für die vorliegenden Zwecke an einigen Stellen erweitert, ansonsten aber weitgehend stark vereinfacht.

2.2.5. Entwicklungsprozesse und aktueller Forschungsstand

Softwareproduktlinien sind in der wissenschaftlichen Forschung kein neues Gebiet. Erste Arbeiten stammen aus den 1980er Jahren [Trig03, S.3]. Die Methodik hat sich in der wissenschaftlichen Forschung etabliert, wenn es um eine Senkung der Entwicklungskosten, der Entwicklungszeit, Verbesserung der Softwarequalität oder allgemein um eine Erhöhung der Variantenanzahl geht. Allerdings hat sich der Ansatz in der Praxis auf breiter Front bisher nicht durchsetzen können. Erst in den letzten Jahren nimmt die Zahl der Unternehmen zu, die

---

[33] Hierzu gehören die Firmen Audi AG, BMW Group, Daimler AG, Fiat Auto S.p.A., Ford Werke GmbH, Jaguar, Land Rover, Dr. Ing. h.c. F. Porsche AG, Volkswagen AG, Volvo Car Corporation und the procurement Forum.

[34] In HIS haben sich die großen deutschen Automobilhersteller zusammengeschlossen; Audi AG, BMW Group, Daimler AG, Dr. Ing. h.c. F. Porsche AG und Volkswagen AG.

diesen Ansatz (meist in direkter Zusammenarbeit mit einem wissenschaftlichen Institut) verfolgen oder zumindest testen. Eine weit verbreitete Verwendung erfolgt derzeit noch nicht. Dies ist umso erstaunlicher als dass inzwischen zahlreiche umfassende Entwicklungstools für Softwareproduktlinien verfügbar sind. Die aktuell bekanntesten Ansätze werden weiter unten kurz beschrieben.

Zuvor erfolgt jedoch noch eine Übersicht über die wissenschaftlichen Forschungsprojekte zu diesem Themenbereich in Europa [vgl. Kieb06, S. 11ff. und Trig03, S. 43f.]. Diese haben die wissenschaftliche Grundlage für die Frameworks gebildet, die im Anschluss vorgestellt werden:

ARES (Architectural Reasoning for Embedded Systems)
Von 1995 bis 2000 wurden im Rahmen des ARES Projektes, welches durch das europäische Esprit-Programm gefördert worden ist, die Grundlagen für die Verwendung von Produktlinien bei Eingebetteten Systemen erforscht. Das Ziel war, Software Entwickler in die Lage zu versetzen, explizit die Architekturen von Eingebetteten Softwareproduktlinien zu beschreiben, zu bewerten und zu managen. Es handelte sich also um die Erforschung von Grundlagen. Eine erste Anwendung und Validierung in der industriellen Praxis hat zwar stattgefunden, allerdings zeigt die Erfahrung, dass für diesen Schritt weitere Forschungsinitiativen notwendig sind.

PRAISE (Product-Line Realization and Assessment in Industrial Settings ESPRIT)
Das PRAISE Projekt deckte den gesamten Entwicklungsprozess einer Produktlinie ab, inklusive der Unterscheidung zwischen Domain und Application Engineering; ein Ansatz, der sich später durchgesetzt hat. Im Gegensatz zum ARES Projekt stand hier die tatsächliche Anwendbarkeit der Ergebnisse in einer industriellen Umgebung im Mittelpunkt. Ziel war es, Techniken zu entwickeln, mit denen Architekturen mit Variabilität und Kommunalität innerhalb einer Produktlinie entwickelt werden können. Aber auch die Abdeckung von funktionalen wie nicht-funktionalen Anforderungen (Requirements Engineering) wurde berücksichtigt. Ein zentrales Ergebnis war ein neuer Referenzprozess zur Entwicklung von Software Produktlinien, der ein duales Vorgehen, also die Aufspaltung in Domain und Application Engineering, vorsieht. Diese Aufteilung wurde erstmals von Weiss und Lai vorgeschlagen [vgl. Weis99].

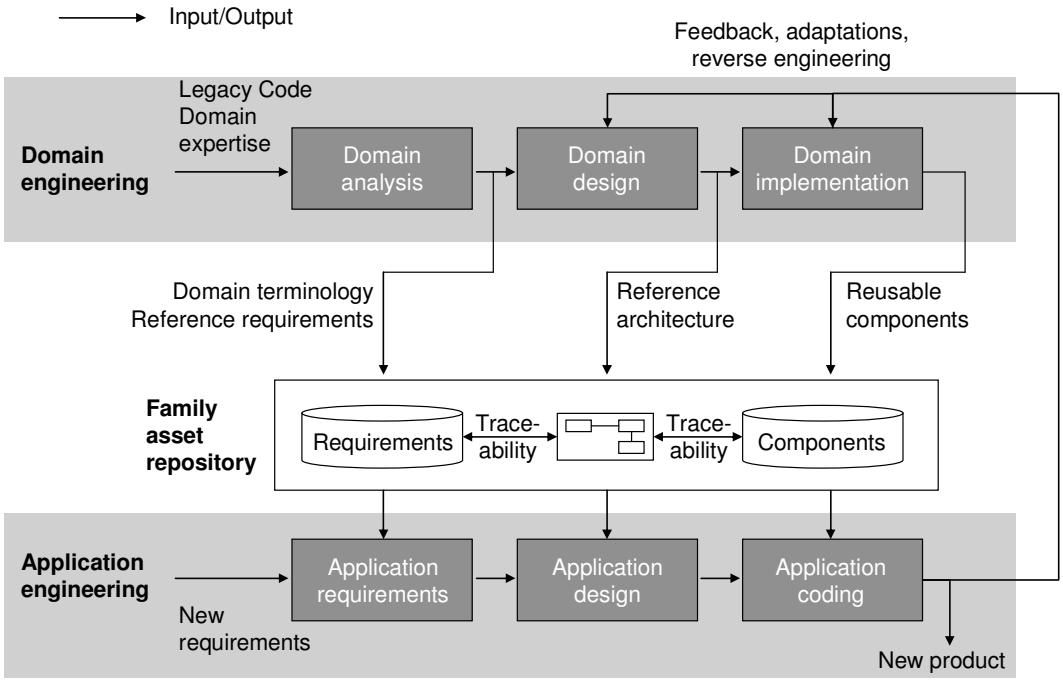

Abbildung 3 – PRAISE Referenzprozess [vgl. Lind02, S. 47]

Dieses Vorgehen hat sich bei zahlreichen weiteren europäischen Forschungsvorhaben in diesem Bereich etabliert [vgl. Böck04a, S.5 und Baye04a, S.1].

ESAPS (Engineering Software Architectures, Processes, and Platforms for System Families)
Im Rahmen des Information Technology for European Advancement (ITEA) Programms wurde das Esprit Programm weitergeführt. ESAPS hatte zum Ziel, den PRAISE Referenzprozess zu erweitern bzw. zu validieren und die Auswirkungen einer Produktlinie auf die Unternehmensorganisation zu untersuchen. Schließlich sollten für die einzelnen Schritte des Referenzprozesses Artefakte erstellt und gemanaged werden.

CAFÉ (From Concept to Application in System-Family Engineering)
Als Erweiterung der Ergebnisse des ESAPS Projektes hatte CAFÉ das Ziel, die Konzepte in eine Anwendung zu übertragen und direkt für die Industrie nutzbar zu machen. Besonderer Fokus galt den ökonomischen Auswirkungen und den Wirkungen auf die Organisation der Unternehmung. Ein wichtiges Ergebnis ist das Business Architecture Process Organisation (BAPO-) Modell. Es fast die 4 relevanten Einflussfaktoren auf eine Produktlinie zusammen: Auswirkungen auf die Gewinne des Unternehmens durch Produkte aus Produktlinien (Business), die Rollenverteilung innerhalb eines Unternehmens bei der Entwicklung einer SWPL (Process), die Übertragung der mit einer SWPL verbundenen Prozesse auf die

bisherige Unternehmensstruktur (Organisation) und schließlich die Architektur der Produktlinie selbst.

Anzumerken ist, dass sich diese 4 Dimensionen teilweise negativ beeinflussen, bspw. wenn die Einführung einer neuen Architektur ein hohes finanzielles und zeitliches Investment erfordert.

FAMILIES (Fact-based Maturity through Institutionalization Lessons-learned and Involved Exploration of System-Family Engineering)
FAMILIES schließt die Reihe der Forschungsprojekte im Rahmen des ITEA Programms ab. Ziel ist eine ganzheitlich Betrachtung von SWPL im Rahmen des BAPO-Modells mit besonderem Fokus auf die Standardisierung der Forschungsergebnisse, um diese leichter in die industrielle Praxis übertragen zu können.

Die genannten wissenschaftlichen Projekte haben die Vorraussetzung für zwei SWPL Frameworks aus Europa geliefert. Das Software Product Line Engineering Framework und PuLSE. Diese beiden werden im Folgenden zusammen mit einem Ansatz des Software Engineering Institutes aus den USA vorgestellt. Denn diese Frameworks ermöglichen aufgrund ihrer umfassenden Werkzeugunterstützung und ihrer formalen Ausgestaltung und Reife eine einfache industrielle Anwendung.

A Framework for Software Product Line Practice (SPLP):
Das Software Engineering Institute (SEI) der Carnegie Mellon University ist seit vielen Jahren eine der führenden Organisationen beim Thema Software Produktlinien. Das Institut kann auf eine umfangreiche Datenbank an SWPL Projekten zugreifen. Das SEI hat sich zum Ziel gesetzt, weitere Unternehmen von den Vorteilen einer systematischen Wiederverwendung durch Nutzung von SWPL zu überzeugen. Daher hat man sich entschlossen, dass Wissen über die Umsetzung von SWPL zu sammeln und zu veröffentlichen. Dies geschieht als ein "web-based, living document" [SEI09b]. Derzeit liegt Version 5 vor, wobei neue Erkenntnisse im Bereich der SWPL aufgrund des gewählten Konzeptes zeitnah eingebunden werden. Durch diesen iterativen und offenen Prozess wird eine hohe Aktualität erzielt.

Das Grundkonzept basiert auf drei Aktivitäten, die untereinander sehr stark miteinander verbunden sind. Einerseits handelt es sich um die bekannte Trennung zwischen Domain (Core Asset) und Application (Product) Engineering. Flankiert werden diese beiden Prozesse von

einem übergreifenden Management Prozess. Dieser ist maßgeblich für den eigentlichen Erfolg oder auch Misserfolg der SWPL verantwortlich (siehe Abbildung 4).

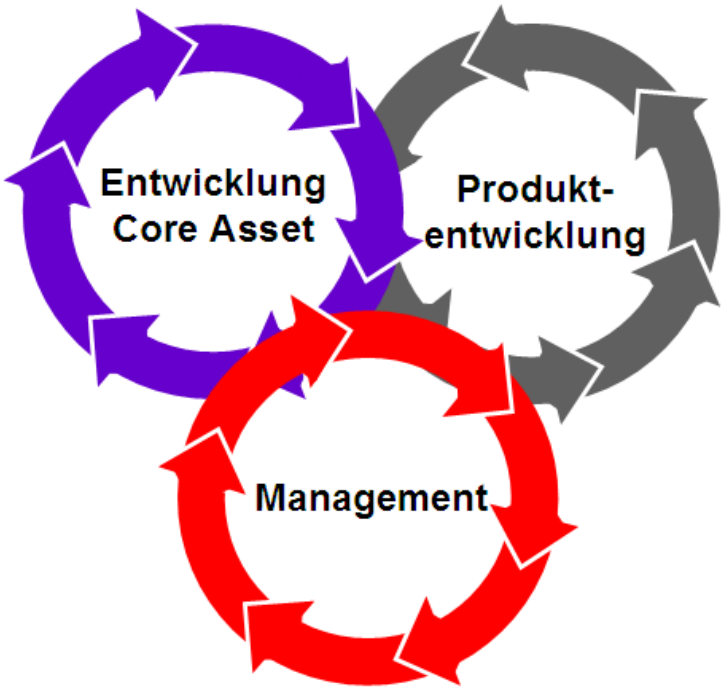

Abbildung 4 – Übersicht der 3 wichtigen Aktivitäten bei SWPL [Clem02, S. 30]

Die Core Asset Entwicklung liefert unter den Einschränkungen der bereits existierenden Produkte, der Unternehmensstrategie, und der Produktionsbedingungen einen möglichst optimalen Umfang der Produktlinie, die Core Asset Base selbst und einen grundlegenden Plan, wie die einzelnen individuellen Produkte auf Basis der Core Asset Base entwickelt werden können.

In der Produktentwicklung werden die Ergebnisse der Core Asset Entwicklung als Eingangsfaktoren genutzt, um durch intelligente Verknüpfung, Anpassung und Entwicklung individueller Komponenten unter Einhaltung des Produktionsplans die eigentlichen Produkte herzustellen. Entscheidendes Merkmal ist die direkte Rückkopplung zur Core Asset Entwicklung, denn aufgrund der hohen Komplexität der Produkte und der SWPL selbst kann es immer wieder vorkommen, dass sich der Produktionsplan nicht direkt umsetzen lässt. In einem solchen Fall ist die schnelle und direkte Kommunikation zwischen den Bereichen notwendig.

Diese Koordination wird durch die Management Ebene geleistet. Es hat sich in der Praxis gezeigt, dass dies der entscheidende Faktor bei der Realisierung einer SWPL ist. Das Management ist final für die Produktlinienstrategie verantwortlich. Neben der Kontrolle des Aufbaus der Produktlinie ist insbesondere die Weiterentwicklung der Produktlinie, sprich der Asset Base, die wichtigste Aufgabe. Denn ohne eine regelmäßige Erweiterung und Anpassung wird keine noch so gute Produktlinie lange bestehen können.

PuLSE (IESE):

PuLSE steht für Product Line Software Engineering und entstand 1999 am Fraunhofer Institut für experimentelles Software Engineering (IESE) als ein adaptierbarer, inkrementeller Ansatz zur Produktlinienentwicklung und –Einführung. Grund für die Entwicklung war die damals fehlende methodische Unterstützung bei der technischen Realisierung einer Produktlinie. Die meisten wissenschaftlichen Arbeiten fokussierten auf die Auswirkungen auf die Organisation und wie man diese managen sollte. Das IESE nutzt PuLSE, um Unternehmen bei der Einführung einer SWPL umfassend zu unterstützen.

PuLSE ist kein von Grund auf neuer Ansatz, aber er kombiniert zahlreiche existierende Ideen zu einem sinnvollen Gesamtkonzept. Im Folgenden wird dieses Konzept kurz vorgestellt und die Auswirkungen auf die hier vorliegende Arbeit diskutiert. Für eine detaillierte Erklärung sei auf [Baye99] verwiesen.

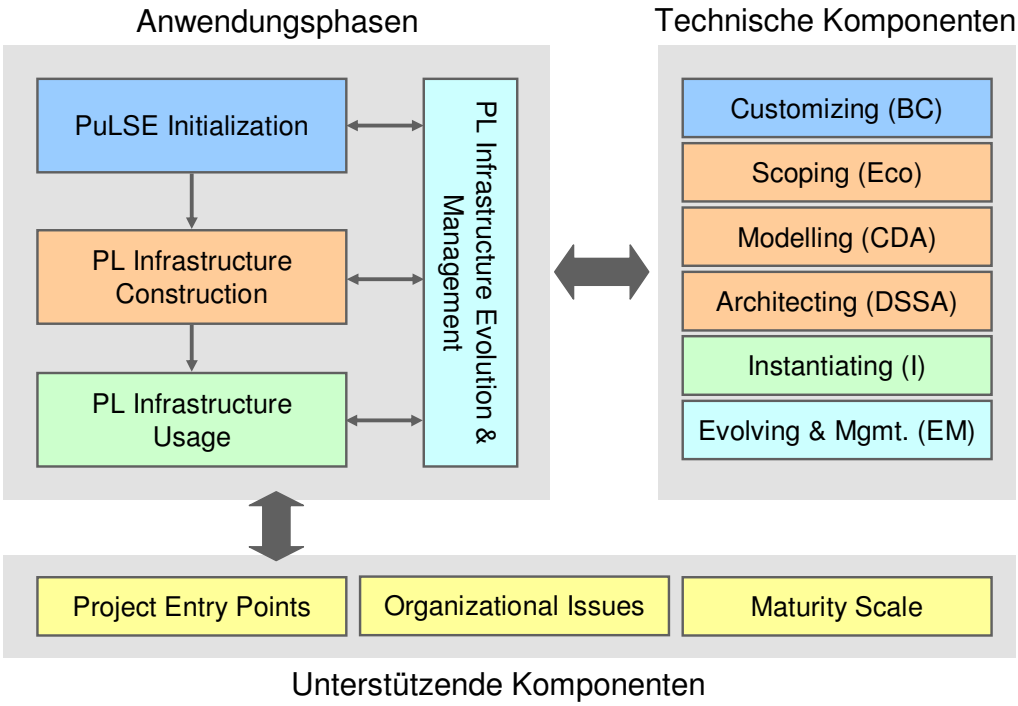

Abbildung 5 – Überblick über PuLSE [Baye99, S.123]

PuLSE besteht aus 3 Komponenten (siehe Abbildung 5): Anwendungsphasen, technische Komponenten und unterstützende Komponenten. Die technischen Komponenten sind dabei eindeutig bestimmten Anwendungsphasen zugeordnet, was durch die Farbkodierung gekennzeichnet wird. Insgesamt deckt das Modell damit alle Phasen der Einführung einer Produktlinie ab und bietet jeweils die notwendige methodische Unterstützung. Diese Phasen sind:

In der **Initialisierungsphase** wird die PuLSE Methode an die individuellen Bedürfnisse des Unternehmens angepasst. Beeinflussende Faktoren sind bspw. die Anzahl der verfügbaren Ressourcen oder der Anwendungstyp. Diese werden in einem Baselining zunächst ausgewählt und erfasst. Anschließend werden diese Faktoren ausgewertet (Evaluation). Aufgrund der möglichen Interdependenz wird hier auf Entscheidungsbäume zurückgegriffen. Das Ergebnis sind Baselining Profile, die dann zu einem Gesamtprozess ausgearbeitet werden (Customizing), an dem sich der gesamte weitere Ablauf orientiert.

Die **Konstruktionsphase** ist der umfangreichste Teil und wird daher in drei Segmente unterteilt, der ökonomischen Analyse, der Domain Analysis und der Reference Software Architecture. Die ökonomische Analyse wird durch PuLSE-Eco unterstützt. Hauptziel ist die wirtschaftlich fundierte Festlegung der Größe der Produktlinie (Scoping). Hierzu werden alle

existierenden und geplanten Produkte zusammen mit den Konkurrenzprodukten hinsichtlich ihrer Charakteristika in einer Product Map aufgenommen. Die Eignung der einzelnen Eigenschaften zur Integration in die Produktlinie wird mittels Benefit Functions analysiert. Obwohl es sich hier hauptsächlich um eine sehr einfache Scoping Methode handelt, wird dieser Schritt auch teilweise in der vorliegenden Arbeit behandelt (siehe Teil B Kapitel 3.4. und 5.2.).

Die Modellierung der Anwendungsdomäne (Domain Analysis) geschieht in der technischen Komponente CDA. CDA greift dabei zahlreiche vorhandene Domain Analysis Konzepte und Ideen des Requirements Engineering auf. Hauptarbeitsinstrument sind Storyboards, Context und Workflow Diagramme, die schließlich zu einem Entscheidungsmodell zusammengefügt werden.

Zum Abschluss der Constructionphase wird aufbauend auf dem Entscheidungsmodell und unter Verwendung von Szenarios, die aus den vorherigen Storyboards hergeleitet werden, die Referenzarchitektur inkrementell entwickelt. Dies wird durch die technische Komponente DSSA unterstützt. Als Ergebnis ergibt sich ein Konfigurationsmodell, welches das bisherige Entscheidungsmodell erweitert.

Die eigentliche Produktentwicklung erfolgt in der **Usagephase** mittels Parametrisierung der Software Architektur. Für die einzelnen Bestandteile gibt es 3 Möglichkeiten. Sie können direkt aus der vorhandenen Domäne stammen, Teil der noch zu entwickelnden Domäne sein, oder kein Bestandteil der Domäne sein. Diese werden eigens für ein Produkt entwickelt. In der Usagephase sind keine Änderungen an der eigentlichen Produktlinie möglich, sie wird allein genutzt, um ein Produkt zu realisieren.

Sollten doch einmal Änderungen notwendig sein, so wird dies an die **Evolution and Management Phase** weitergegeben. Diese Phase dient der Gesamtkoordination des Prozesses und sorgt für die andauernde Weiterentwicklung und Anpassung der Produktlinie.

Neben der Variabilität des Ansatzes, der Berücksichtigung der individuellen Unternehmensziele und der Möglichkeit zur kompletten Neuentwicklung als auch der systematischen Weiterverwendung bisheriger Produkte bietet PuLSE zusätzlich den Vorteil, dass viele Komponenten der Methode zunächst unabhängig voneinander im Unternehmen eingesetzt werden können. Es werden vier inkrementelle Stufen der Einführung von PuLSE beschrieben:

- **Initial**: Einzelne Komponenten (hauptsächlich Eco, CDA und DSSA) werden unabhängig verwendet.
- **Full**: Alle Elemente der Konstruktionsphase sind realisiert, allerdings sind diese noch nicht vollständig integriert, die Produktinstanziierung startet teilweise.
- **Controlled**: PuLSE wird über den gesamten Produktlinien Entwicklungsprozess angewendet, alle Phasen sind integriert.
- **Optimized**: Über mehrere Lebenszyklen einer Produktlinie hat sich das Verfahren im Unternehmen etabliert und weiterentwickelt.

Insgesamt bietet PuLSE eine umfangreiche Unterstützung über den gesamten Entwicklungsprozess einer Produktfamilie, allerdings sind nicht für alle Aspekte automatisierte Tools verfügbar, wodurch ein erheblicher manueller Aufwand bei der Nutzung von PuLSE entsteht.

Software Product Line Engineering Framework

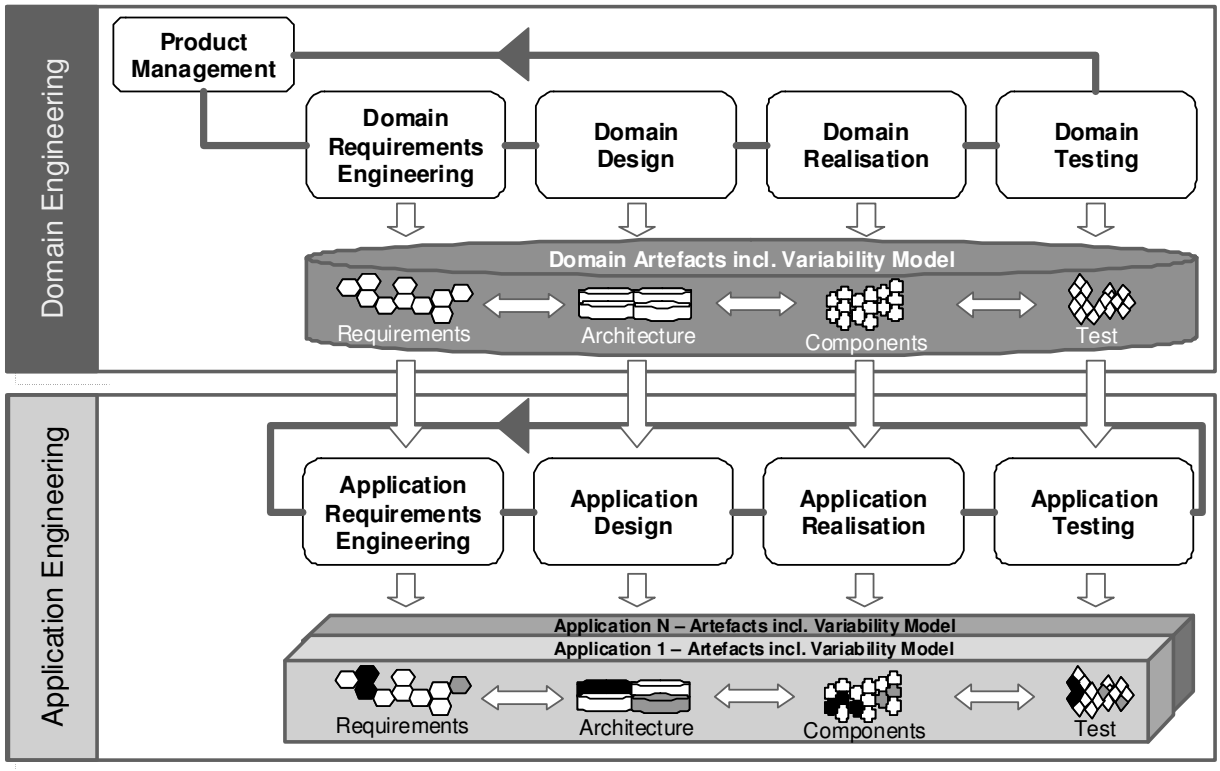

Abbildung 6 – Das Software Product Line Engineering Framework [Pohl05, S.22]

Das Software Product Line Engineering Framework basiert auf den beiden grundlegenden Konzepten der traditionellen Produktlinien Entwicklung Plattformen und Mass Customization

(siehe Kapitel 2.1.1.). Das Framework basiert auf den Ergebnissen der oben beschriebenen europäischen Projekte PRAISE, ESAPS, CAFÉ und FAMILIES. Der Ansatz greift zudem die Idee von Weiss und Lai [Weis99] auf, den Produktlinienentwicklungsprozess in zwei Teilprozesse zu trennen: Domain und Application Engineering. Damit werden direkt die Ergebnisse des PRAISE Projektes verwendet. Beide Phasen werden jeweils in 4 namensgleiche Teilprozesse unterteilt. Für das Domain Engineering gilt:

- **Requirements Engineering**: Basierend auf einer Planung für zukünftige Produktentwicklungen und –Innovationen werden für diese Produkte Anforderungen festgelegt. Dies geschieht nicht einzeln für jedes Produkt, sondern lediglich die gemeinsamen, produktübergreifenden Anforderungen und deren antizipierte Variabilität werden abgedeckt. Ergebnis ist ein Variabilitätsmodell der Produktlinie.

- **Design**: Es erfolgt eine Unterteilung der einzelnen Softwarebestandteile in wieder verwendbare Teile und notwendige Einzelentwicklungen. Basis dafür ist eine Referenzarchitektur, die für die gesamte Produktlinie gilt. D. h., dass alle individuellen Produkte die Variabilität dieser Referenzarchitektur nutzen und nur innerhalb dieses Rahmens von ihr abweichen können. Das anfängliche Variabilitätsmodell wird verfeinert und um interne Variabilität, also Variabilität die nicht für den Kunden, sondern nur für den Entwickler sichtbar ist, erweitert.

- **Realisation**: Aufbauend auf der Referenzarchitektur wird das Design verfeinert und die wieder verwendbaren Softwarebestandteile werden implementiert. Ergebnis sind lose gekoppelte und konfigurierbare Komponenten, die allerdings zusammen kein lauffähiges System ergeben.

- **Testing**: Dieser Teilprozess führt eine Validierung und Verifizierung der entwickelten Softwarebestandteile (der sogenannten Domain Artefakte) durch. Erschwert wird dies durch die Tatsache, dass sich eben kein lauffähiges System ergibt, sondern Bestandteile auf die Erfüllung der Anforderungen hin isoliert und inklusive noch nicht gebundener Variabilität überprüft werden müssen. Ergebnis sind Testartefakte, die selbst wieder verwendbar sind, indem sie später auf die fertigen individuellen Produkte angewendet werden können.

Inhaltlich erfolgt im Application Engineering jeweils die konkrete Ausgestaltung der Vorgaben aus dem Domain Engineering. Dies bedeutet, dass die Ergebnisse der Domain

Teilprozesse den Input für die Application Teilprozesse liefern. In jedem Teilprozess werden die Abweichungen des individuellen Produktes von den Vorgaben oder Möglichkeiten der Domain analysiert. Anschließend wird die enthaltene Variabilität gebunden, also festgelegt. Schließlich werden die individuellen, nicht wieder verwendbaren Artefakte (Application Artefacts) entwickelt. Am Ende steht das fertige und lauffähige Produkt.

Im Rahmen des Domain Engineering findet noch ein übergreifender Prozess, das **Product Management**, statt. Hier werden ökonomische Aspekte, das Produktportfolio des Unternehmens, die Marktstrategie und der gesamte Umfang der Produktlinie festgelegt. Die Ergebnisse dieses Prozesses sind eine Product Roadmap, der Product Line Scope und eine Übersicht der existierenden Software Bestandteile, die möglicherweise wiederverwendet werden können. Der Ansatz bezeichnet diese Ergebnisse nicht als Domain Artefakte, da es sich nicht um Software Artefakte im klassischen Sinn handelt. Vielmehr dienen diese Ergebnisse zur Lenkung der Entwicklungsprozesse in den anderen Bereichen. Daher liefert das Product Management in der Grafik keine direkten Artefakte als Ergebnis, sondern wird als ein übergreifender Steuerungsprozess verstanden.

Pohl, Böckle und van der Linden betrachten Variabilität als das wichtigste Konzept zur Realisierung von Wiederverwendung [Pohl05, S. 58 ff.]. Variabilität wird entsprechend ihres Frameworks im Domain Engineering definiert und im Application Engineering gebunden. Ein Produktlinienansatz wird nun dadurch erschwert, dass Variabilität in allen Teilprozessen auf unterschiedlicher Detailstufe definiert wird. Es fehlt eine ganzheitliche Sicht auf die Variabilität, wodurch es zu Unstimmigkeiten zwischen den einzelnen Prozessen kommen kann. Aus diesem Grund definiert [Pohl05, S. 72 ff.] mit dem **orthogonalen Variabilitäts-modell** einen übergreifenden Ansatz, der die Variabilität über den gesamten Produktlinien-entwicklungsprozess einheitlich definiert. Dadurch ist es möglich, bei Veränderungen an der Domain schnell alle davon betroffenen Bestandteile zu identifizieren und die notwendigen Veränderungen durchzuführen.

Pohl, Böckle und van der Linden haben mit ihrem Framework eine umfangreiche und vollständige Zusammenfassung des aktuellen Forschungsstandes zum Software Product Line Engineering geliefert. Wichtig ist anzumerken, dass es sich hierbei lediglich um ein generisches Framework handelt. Es werden zwar zahlreiche Beispiele mit erfolgreicher Umsetzung besprochen, bei dem eigentlichen Modell handelt es sich jedoch um einen reinen "High-Level" Ansatz. Dementsprechend werden zwar die Grundsätze für einen Einsatz einer

Software Produktlinie festgelegt, für einen unternehmensindividuellen Einsatz ist jedoch zuvor eine Detaillierung und Ausformulierung des Ansatzes für die jeweilige Anwendungsdomäne (z. B. Automotive) notwendig. Deutlich wird dies insbesondere aufgrund der fehlenden Toolunterstützung. Für weitere Details sei auf [Pohl05] verwiesen.

Als Schlussfolgerung aus der Analyse des aktuellen Forschungsstandes zu Software Produktlinien kann festgehalten werden, dass europäische und US-amerikanische Forschungsinstitute führend in diesem Bereich sind. Es gibt zahlreiche umfassende und aktuelle Arbeiten, die das Vorgehen zur Erstellung einer Software Produktlinie beschreiben. Im Bereich der Individualisierung des Ansatzes für eine spezielle Anwendungsdomäne und bei der Werkzeugunterstützung lassen sich jedoch Verbesserungspotenziale erkennen. Für die industrielle Praxis insbesondere von Interesse sind Aussagen über die ökonomischen Auswirkungen einer Software Produktlinie. Auch hier liegen bereits zahlreiche Arbeiten vor, die allerdings – wie bereits in der Einleitung angesprochen – nicht alle relevanten Aspekte abdecken. An dieser Stelle sollte lediglich eine Übersicht über den allgemeinen Forschungsstand zu SWPL gegeben werden, daher wurde auf eine Betrachtung der ökonomischen Aspekte und Modelle hier verzichtet. Stattdessen wird zur Vermeidung von Redundanz auf Teil B Kapitel 3 verwiesen, wo die Ansätze zur ökonomischen Bewertung im Detail besprochen werden.

2.2.6. Auswahl erfolgreicher Anwendungsbeispiele

Im Folgenden werden einige erfolgreiche Umsetzungen der Produktlinien-Idee im Softwarebereich vorgestellt. Ziel ist es nicht, einen vollständigen Überblick über die aktuelle Anwendungslandschaft zu geben, sondern lediglich zu zeigen, dass der Ansatz bereits in mehreren Fällen erfolgreich eingesetzt worden ist. Für weitere Informationen wird auf [Pohl05, S. 414ff.] verwiesen, der insgesamt 15 Beispiele über erfolgreiche Umsetzungen der SWPL Idee aufführt.

2.2.6.1.Philips Medical Systems (PMS)

Philips Medical Systems (PMS)[35] stellt Geräte her zur Aufzeichnung, Verarbeitung und Auswertung medizinischer Bilder in Krankenhäusern. Dies umfasst alle aktuell verfügbaren Technologien, wie z.B. Röntgen, Magnet-Resonanz-Tomographie (MRT), Ultraschall.

Die einzelnen Geräte unterscheiden sich in ihrer Funktionalität zwar deutlich, dennoch bieten alle ähnliche Funktionen zur Ver- und Bearbeitung von Bildern, z.B. laden, speichern, vergrößern und Informationen hinzufügen. PMS hat daher diese generischen Funktionen in einer Bildverarbeitungs-Middleware-Plattform (sprich einer Produktlinie) zusammengefasst. Die ersten Überlegungen hinsichtlich einer Produktlinie fanden 1997 statt. Die Architektur wurde bereits ein Jahr später festgelegt und im Jahre 2001 wurde die erste Version der Plattform veröffentlicht. Man wählte einen evolutionären Ansatz mit einer kleinen Basis zu Beginn, um die laufende Entwicklung möglichst wenig zu beeinflussen. Die Architektur basiert auf 3 Elementen: Components, Connectors und Semantics (CCS). Die Komponenten bündeln die Funktionalität, die von den einzelnen Produkten benötigt wird und stellen damit die wieder verwendbaren Elemente der Architektur dar. Die Connectors sind die Schnittstellen der Produkte an die Plattform. Diese sind von allen Produkten zu verwenden. Um diese daher stabil zu designen beschränken sich die Schnittstellen auf generische Datenzugriffsfunktionen. Die notwendige semantische Information über die Ausgestaltung der Schnittstellen liefern Information Models. Der eigentliche Datenaustausch geschieht in diesen Modellen und kann individuell angepasst werden.

Ende 2004 wurde die Plattform bereits bei mehr als 10 Produktgruppen verwendet. Es zeigte sich, dass der Entwicklungsaufwand um einen Faktor von 1,6 höher lag als eine Einzelentwicklung, aber natürlich konnten die Kosten auf mehrere Produkte umgelegt werden. Als weiterer positiver Effekt ging die Anzahl der berichteten Probleme deutlich zurück.

2.2.6.2.Hewlett Packard (HP) – Firmware für Drucker

Hewlett Packard (HP) ist einer der größten Druckerhersteller weltweit. Die verkürzten Produktlebenszyklen führten auch bei HP zu einem hohen Entwicklungsaufwand, u.a. musste die komplexe Firmware für jedes neue Gerät aufwendig angepasst werden. In Anbetracht der

---

[35] Heute wird dieser Bereich als Philips Healthcare bezeichnet und stellt neben Consumer Lifestyle und Lighting einen der drei wichtigen Hauptunternehmensbereiche von Philips dar.

zahlreichen, nicht erfolgreichen Versuche Software wiederzuverwenden, entschied sich HP für einen neuen produktlinienbasierten Ansatz zur Entwicklung der eigenen Owen Firmware, die in zahlreichen Druckern verwendet wird. Es wurde kein Reuse Team eingesetzt, sondern die bestehenden Firmware-Entwickler schlossen sich zu einer Firmware Kooperative zusammen. Es handelt sich also nicht um eine starre organisatorische Vorgabe, vielmehr basiert die Zusammenarbeit auf einem losen und freiwilligen Zusammenschluss unabhängiger Projektteams mit dem Ziel, die Entwicklungsaufwände gemeinsam zu reduzieren. Es kristallisierten sich 3 Erfolgsfaktoren heraus:

- Es bedarf einer guten, klaren komponenten-basierten Architektur, die eine Kopplung verschiedener Komponenten erlaubt, die notwendigen Verknüpfungen aber minimiert.
- Es ist ein passender Entwicklungsansatz notwendig. HP entschied sich für ein Firmware Asset Management basierend auf "change-sets". Jede Veränderung oder Hinzufügung eines Features wird in einem change-set zusammengefasst. Jedes Projektteam kann nun individuell entscheiden, welche change-sets es zusätzlich zur allgemeinen Basis für sein Projekt verwenden will. Dabei muss allerdings sichergestellt sein, dass wichtige und kritische Updates komplett berücksichtigt und umgesetzt werden.
- Für einen langfristigen Erfolg des Ansatzes müssen entsprechende Schlüsselrollen im Unternehmen definiert werden, um klare Verantwortlichkeiten zu haben und die kontinuierliche Verbesserung voranzutreiben.

HP erzielte mit diesem Vorgehen eine um 70% höhere SW-Wiederverwendungsrate gegenüber der herkömmlichen individuellen Entwicklung. Gleichzeitig wurde die Time-to-Market im Durchschnitt auf ein Drittel reduziert bei einer 25-fachen Verringerung der Zahl der Fehler.

2.2.6.3.MAN Nutzfahrzeuge – Software als original Ersatzteil

Auch in der Automobilindustrie haben erste Unternehmen erfolgreich Erfahrungen mit SWPL gesammelt. MAN hat für den Bereich Nutzfahrzeuge im Juli 2005 ein neues Geschäftsmodell eingeführt: "MAN-genuine Software". Im Rahmen dieses Geschäftsmodells wird Kunden die Möglichkeit geboten Software Updates genau wie sonstige Ersatzteile zu beziehen. Im Gegensatz zu Autos ändert sich bei Nutzfahrzeugen häufiger der Einsatzzweck, z.B. nach einem Verkauf. Daher sind auch Fahrzeuganpassungen üblich. Dies kann sogar Änderungen

an der HW-Mechanik betreffen, z.B. der Austausch einer starren Ladefläche durch eine Kippermulde. Die Bedienung muss in die bestehende Fahrzeugarchitektur integriert werden. Inzwischen werden auch bei Nutzfahrzeugen mehr und mehr Funktionen mittels SW umgesetzt. Als Beispiele sind Abstimmungen zwischen Betriebs- und Dauerbremsen, Kontroll-Anzeigen und die Funktion des Bremsassistenten[36] zu nennen.

Um SW als Ersatzteil verkaufen zu können, war es zunächst notwendig, die Steuergeräte-Architektur zu vereinheitlichen. Dies wurde mit der Baureihe TG bei MAN umgesetzt. "Merkmale dieser Struktur sind die einheitliche Vernetzung über CAN J1939 Bus und die Aufteilung der Subnetzwerke Motor-CAN, Triebstrang-CAN und Instrumentierungs-CAN" [Scht05, S. 822]. Dadurch ist MAN in der Lage einheitlich SW über alle Baureihen hinweg zu nutzen. Die Individualisierung erfolgt mittels Parametrisierung über ein Fahrzeug-Datenfile. Die Übertragung zum Fahrzeug geschieht per Flashprogrammierung über ASAM-MCF 2D. Erst diese weitgehende Vereinheitlichung erlaubt es MAN, den Kunden mit geringem Aufwand neue SW-Bausteine für jedes Fahrzeug anzubieten. Mit Hilfe eines entsprechenden Abrechnungsmodells wurde das Geschäftsmodell vervollständigt. Die Kunden erhalten einen Zusatznutzen, da nun auch SW bezogene Funktionen nachträglich individuell anpassbar sind. Für MAN ermöglicht das Geschäftsmodell, einen direkten Ertrag aus der SW Entwicklungsarbeit zu ziehen, was sich im Allgemeinen als eher schwierig erweist. Für weitere Details zu "MAN genuine Software" siehe [Scht05].

2.2.7. Abgrenzung des Produktlinienumfangs: Product Line Scoping

"Ein entscheidender Schritt einer erfolgreichen Produktlinienentwicklung ist es, zu identifizieren, welche Funktionalitäten einer Produktlinie wieder verwendbar unterstützt werden sollen und können. Erst dadurch, dass diese Auswahl geeignet getroffen wird, sind die enormen wirtschaftlichen Vorteile einer Produktlinie realisierbar." [Schm04, S. 43]

Schmid verdeutlicht damit die große Bedeutung der Abgrenzung des Produktlinienumfangs. Allerdings greift die klassische Scoping Definition aus dem Projekt Management zu kurz, da es sich bei Produktlinien ja eben nicht um ein abgrenzbares Projekt zur Realisierung eines Produktes handelt. In der Literatur hat sich daher eine Dreiteilung des Begriffes als geeignet herausgestellt:

---

[36] Bremsomat bei MAN Nutzfahrzeugen

Product Portfolio Scoping: Dieser Schritt ist eng mit der Produktportfoliodefinition verwandt. Meist im Rahmen von Workshops werden die Produkte identifiziert, die einerseits zu der Produktlinie aufgrund der strategischen Ausrichtung und der Marktposition des Unternehmens passen würden, die sich andererseits aber auch aufgrund ihrer grundsätzlichen Verwandtschaft für eine Produktlinie eignen, also bspw. Gleichteile oder ähnliche Komponenten enthalten, deren Unterschiede über Variabilität berücksichtigt werden können. Die Kriterien für eine positive Entscheidung sind die Implementierungseigenschaften und die technischen und ökonomischen Folgen.

Domain Scoping: Ziel ist es, Funktionsbereiche zu identifizieren, die sich für eine Wiederverwendung eignen. Die zu beantwortende Frage lautet, welche Funktionalität gehört zusammen und kann innerhalb einer Domäne abgebildet werden. Die meisten Ansätze orientieren sich hier an den Features einer Software. Die Identifikation dieser Features ist nicht in allen Fällen einfach, gelingt unter Einbeziehung der Entwickler jedoch meist.

Asset Scoping: Während sich die ersten beiden Stufen mit der eigentlichen Problemdefinition beschäftigt haben, geht es in der dritten Phase um die Lösung des Problems. Fokus sind die Gemeinsamkeiten der identifizierten Assets, die zu realisierenden Variabilitäten und die Auswirkungen auf die Wirtschaftlichkeit. Gerade die Frage nach der Wirtschaftlichkeit ist eng mit der Referenzarchitektur des Produktlinien Engineering verbunden. Da sich das Unternehmen bei der ersten Festlegung des Umfangs der Produktlinie meist in einer sehr frühen Phase der Produktentwicklung befindet, ist es oft nicht möglich, definitive Aussagen über die Produktarchitektur zu machen. Daher erscheint es zunächst sinnvoll, über eine Komponentenidentifikation die Wunschkomponenten für die Domain zu bestimmen. Die Überprüfung der technischen Realisierung erfolgt dann später, könnte aber dennoch einen Einfluss haben und damit das ökonomisch optimale Scoping ändern.

Der umfangreichste Ansatz zum Software Produktlinien Scoping findet sich als Bestandteil des bereits vorgestellten PuLSE Ansatzes (siehe Kapitel 2.2.5.), PuLSE-Eco. Der Ansatz unterteilt das Scoping in 3 Teile:

Product Line Mapping: Es erfolgt eine Erfassung aller existierenden und zukünftig (im Zeitrahmen der Produktlinie) geplanten Produkte. Zusätzlich werden die Produkte hinsichtlich ihrer enthaltenen Features unterschieden, wobei lediglich funktionale Features berücksichtigt werden. Nicht-funktionale Features werden erfasst, wenn sie direkt mit einem funktionalen

Feature verbunden sind, z.B. die nicht-funktionale Anforderung nach Ausfallsicherheit bei der funktionalen Anforderung des Anti-Blockier-Systems.

Domain Potential Analysis: Ziel ist die ungefähre Schätzung des Wiederverwendungspotenzials. Dieser Schritt greift auf etablierte Techniken des Prozess-Assessments, wie CMMI und SPICE, zurück (siehe Kapitel 2.2.4.). Allerdings wird lediglich die generelle Vorgehensweise der bekannten Verfahren genutzt. Die abgefragten Kategorien wurden neu definiert. Die 8 Hauptkriterien werden in insgesamt 100 Einzelfragen (statt 450 bei CMMI) unterteilt. In Verbindung mit einer unkomplizierteren Erfassung der Antworten ergibt sich ein reduzierter Umfang.

Reuse Infrastructure Scoping: Der dritte Teil versucht die optimale Ausgestaltung der Asset Base, also der wiederverwendbaren Bestandteile, festzulegen. Als Entscheidungsbasis dient der Return on Investment (ROI). Zur Bestimmung des ROI werden Nutzenfunktionen herangezogen. Denn der ROI wird maßgeblich durch die Unternehmensziele bestimmt. Es gilt also teilweise gegenläufige Zielsetzungen zu berücksichtigen, z.B. reduzierte Kosten, verkürzte Entwicklungszeit bei späteren Produkten bei verlängerter Entwicklungszeit für das erste produktlinienbasierte Produkt, gesteigerte Qualität und reduzierte Wartungskosten. PuLSE-Eco kombiniert die einzelnen Nutzenfunktionen zu einer gewichteten Summe. Anhand dieser Summe erfolgt eine Einteilung innerhalb einer 6-stufigen Skala, die die Eignung zur Wiederverwendung abbildet.

Der dritte Schritt als eigentliche Problemlösung erfordert mit Abstand den größten Aufwand. Daher entscheiden sich viele Unternehmen lediglich die ersten beiden Schritte durchzuführen. Im Rahmen der vorliegenden Arbeit spielt das Scoping ebenfalls eine entscheidende Rolle. Die ersten beiden Phasen des PuLSE-Eco Ansatzes finden sich dementsprechend wieder, sind jedoch für Embedded Systems im Automotive Bereich und für eine sehr frühe und schnelle Anwendbarkeit adaptiert. Hinsichtlich des dritten Schrittes wird der Standpunkt vertreten, dass der Nutzen den Mehraufwand rechtfertigt. Um jedoch dem Anspruch einer mit geringem Aufwand durchführbaren Analyse gerecht zu werden, wird dieser Teilschritt stärker angepasst. Insbesondere werden die Nutzenfunktionen stark vereinfacht, u.a. dadurch dass nicht alle möglichen Zielsetzungen berücksichtigt werden, sondern der wirtschaftliche Aspekt den Hauptfokus bildet. Für Details sei auf Teil B Kapitel 5 verwiesen.

3. Embedded Systems

3.1. Definition

Für eingebettete Systeme, die im Verlauf dieser Arbeit konsistent als Embedded Systems bezeichnet werden, finden sich in der aktuellen Liteartur zahlreiche leicht unterschiedliche Definitionen, von denen hier einige exemplarisch gezeigt werden sollen:

- Embedded Systems sind "application specific systems which contain hardware and software tailored for a particular task and are generally part of a larger system." [Kuma96, S. 11]
- Marwedel erweitert diese Sichtweise noch um den Aspekt der Sichtbarkeit, wenn er formuliert: „Eingebettete Systeme sind informationsverarbeitende Systeme, die in ein größeres Produkt integriert sind, und die normalerweise nicht direkt vom Benutzer wahrgenommen werden.“ [Marw07, S.1]
- In die gleiche Richtung geht Wolf, wenn er definiert: „What is an embedded computing system? Loosely defined, it is any device that includes a programmable computer but is not itself intended to be a general-purpose computer”. [Wolf08, S. 1]
- Auch Wilmshurst nennt die Sichtbarkeit in seiner Definition: “An embedded system is a system whose principal function is not computational, but which i scontrolles by a computer embedded within it. The computer is likely to be a microprocessor or microcontroller. The word embedded implies that it lies inside the overall system, hidden from view, forming an integral part of greater whole.” [Wilm01] Zusätzlich definiert er umfassend die wesentlichen Eigenschaften von Embedded Systems: “An embedded system is a microcontroller-based, software-driven, reliable, real time control system, autonomous, or human- or network-interactive, operating on diverse physical variables and in diverse environments, and sold into a competitive cost-conscious market” [Wilm01].

Bei Embedded Systems gibt es heute eine sehr große Bandbreite hinsichtlich der Komplexität der Systeme, von einfachen 8-Bit Einzelprozessorsystemen mit hardwarenah entwickeltem Code und einer starken Optimierung aufgrund von Kosteneinschränkungen bis hin zu hochkomplexen Multiprozessorsystemen. Solche Systeme nutzen entweder Standard PC

Hardware oder speziell angepasste Komponenten. Dabei wird im Unterschied zu klassichen PCs auch heute noch in vielen Fällen die Hardware auf Kosteneffizienz und bestmögliche Leistung optimiert, z.B. bei Verschlüsselungsaufgaben [vgl. insgesamt Kama03, S. 5ff.].

Die Definitionen verdeutlichen die Bedeutung der Integration von Embedded Systemen und die besonderen Anforderungen aufgrund der ungewöhnlichen Einsatzfelder. Allerdings kommt die Berücksichtigung der Kombination bei Embedded Systems aus Software, Elektronik und Mechanik zu kurz. Die Bedeutung der Integration der Systeme mit der Außenwelt zeigt sich stärker bei [Broy98c]. „Gerade bei engebetteten Systemen wird oft die Rolle von Software nicht wahrgenommen“ [Broy98c, S. 384]. Insbesondere beim Einsatz zur Steuerung von Prozessen besteht „ein besonders enger Zusammenhang zwischen Hardware und Software“ [Broy98c, S. 385], denn hier werden mittels Software „Prozessoren direkt über Sensoren und Aktuatoren mit physikalischen Vorgängen gekoppelt. Daten werden über die Sensoren gemessen und durch die Prozessoren verarbeitet. Abhängig von den Verarbeitungsergebnissen werden Steuergeräte angestoßen, die dann in die Mechanik der Geräte eingreifen“ [Broy98c, S. 385f.]. Ein Embedded System ist demnach als ein Softwaresystem zu verstehen, das mittels Sensoren analoge und digitale Informationen aus seiner Umwelt aufnimmt und über Aktuatoren selbst Einfluss auf seine Umwelt nehmen kann, oft mit regelungstechnischem Charakter. [Scho05, S. 7] zeigt dies am Beispiel des Airbagsystems moderner Fahrzeuge. Im Mittelpunkt steht hier also stärker die Software, die mechanische Teile kontrolliert und steuert.

Im Rahmen dieser Arbeit wird daher die folgende Definition zu Grunde gelegt:

- Embedded Systems sind Microcontroller-basierte, Software-gesteuerte Computersysteme, die Software, Elektronik und Mechanik kombinieren, um unter Optimierung der eingesetzten Ressourcen (wie Rechenleistung, Speicher und Strombedarf) mittels Sensoren Informationen aus der Umgebung aufnehmen, zu verarbeiten und mit den Berechnungsergebnissen über Aktoren die Umwelt zu beeinflussen. Dabei sind sie meist nicht als Rechensystem wahrnehmbar, da die Rechenleistung lediglich eine notwendige Voraussetzung zur Erfüllung des eigentlichen, spezifischen Einsatzzwecks ist.

Diese Definition ist durch die grundlegende Ausrichtung der Arbeit auf Embedded Systems im Automobil geprägt. Dies zeigt sich insbesondere durch die Betonung der mechanischen

Komponente und der Interaktion mit der Umwelt. Für den weiteren Verlauf ist diese Definition wichtig, da sie verdeutlicht, dass trotz der großen Ähnlichkeiten zahlreiche wichtige Unterschiede zwischen klassischen PCs und Embedded Systems bestehen. Dies wirkt sich ebenso auf die dafür entwickelte Software aus. Und damit haben diese Unterschiede eine direkte Auswirkung auf die Übertragbarkeit des SWPL Ansatzes auf Embedded Systems. Im weiteren Verlauf werden die spezifischen Elemente dieser Unterschiede weiter analysiert und betrachtet. Zunächst erfolgt aber noch eine etwas allgemeinere Betrachtung der Bedeutung von Embedded Systems.

3.2. Bedeutung von Embedded Systems

Embedded Systems sind aus unserem alltäglichen Leben nicht mehr wegzudenken.[37] In einer Vielzahl von Produkten sind heutzutage Funktionen eingebaut, die nur durch den Einsatz von elektronischen Komponenten und entsprechender Software möglich sind [vgl. z.B. Scho05, S.7]. So wird angenommen, dass ca. 90% aller komplexen Prozessoren in Embedded Systems zu finden sind [Marw07, S. 8]. Stiller bezieht sich auf die reinen Prozessor Verkaufszahlen und nennt einen embedded Anteil von 98% im Jahre 1999 [Stil00, S.52]. Dieser Wert wird bei [Scho05, S.2] auch für das Jahr 2002 bestätigt. Den Großteil machen dabei 8-Bit-Prozessoren aus, aber auch der embedded Anteil von 79% bei den 32-bit-Prozessoren ist beachtlich.

Embedded Systems zeigen einen Trend zu steigender Komplexität und zunehmendem Funktionsumfang [Schä06, S. 1]. Dies führt dazu, dass sich die Grenzen zwischen PCs und Embedded Systems aufzulösen beginnen. So konnte man bspw. die ersten Spielekonsolen eindeutig als Embedded Systems klassifizieren.[38] Ihr einziger Zweck diente der Berechnung eines Videobildes unter Berücksichtigung eines Nutzerinputs. Die Hardware war standardisiert und die Software genau beschrieben und nicht vom Nutzer veränder- oder anpassbar. Heutige Spielekonsolen haben sich hingegen hinsichtlich ihres Funktionsumfangs und der Variabilität stark ihren PC Kollegen angenähert. So erlauben die heutigen Konsolen von Sony (Playstation 3) und Microsoft (Xbox 360) das Herunterladen von Demos aus dem Internet, das Spielen mit menschlichen Gegnern weltweit und sie enthalten vielfach Standard PC Hardware, wie z.B. die eingebauten Festplatten. Auch befinden sich im Gegensatz zu

---

[37] Auf der IFA (Internationale Funkausstellung) in Berlin wurde 2008 erstmals ein Bereich für weiße Ware, also Haushaltsgeräte wie z.B. Kühlschränke, Waschmaschinen oder Herde reserviert.

[38] Zu den ersten Spielekonsolen gehörte Magnavox Odyssey, die fest verdrahtete Schaltkreise besaß, dafür aber keinen Mikroprozessor und keine Softwaremodule. Das System war nicht programmierbar und demnach eigentlich nicht als Computer anzusehen. Atari 2600 hingegen verfügte über Softwaremodule und war programmierbar.

früher gleichzeitig unterschiedliche Versionen der Hardware auf dem Markt (Bsp. Xbox mit und ohne Festplatte, Playstation 3 bis Ende 2009 insgesamt 4 Hardware-Generationen entwickelt, u.a. mit Änderung der Fertigungstechnik des Hauptprozessors und des Grafikchips). Dies ist u.a. auf die verkürzten Entwicklungszeiten zurückzuführen.

Eine ähnliche Entwicklung war auch bei Videorekordern zu beobachten. Mitte der 1970er Jahre kamen erste Videorekordersysteme für den Haushalt auf den Markt (VHS von JVC, Betamax von Sony und Video2000 von Grundig und Philips)[39]. Diese verwendeten zwar ein rein analoges Aufnahmeverfahren, aber auch diese Geräte enthielten bereits einige Mikrocontroller, u.a. zur Abstimmung der Laufgeschwindigkeiten der einzelnen Antriebe [WiVi09]. Mit dem Aufkommen der DVD ab dem Jahr 2000 stieg auch die Anzahl der elektronischen Komponenten stark an, u.a. aufgrund der notwendigen Rechenkapazitäten zur Dekomprimierung des digitalen Materials. Betrachtet man heutige Festplatten- und DVD-Rekorder, finden sich dort zahlreiche Standard-PC Bauteile, von der Festplatte über das DVD Laufwerk bis zu den Prozessoren.

Es ist anzunehmen, dass der allgemeine Trend zur weiteren Verbreitung von Elektronik und der gleichzeitigen Harmonisierung von Komponenten bei gleichzeitiger Reduktion der Entwicklungszyklen weiter zunehmen wird. Dies stellt die Hersteller vor große Herausforderungen, zu deren Lösung der SWPL Ansatz beitragen kann.

3.3. Eigenschaften von Embedded Systems

Trotz der zunehmenden Verwendung von Standard-PC Bauteilen weisen Embedded Systems einige spezifische Eigenschaften auf. Im Folgenden werden diese Charakteristiken basierend auf der Einteilung von Marwedel [Marw07, S.2ff.] kurz erläutert. Ebenso finden sich entsprechende Aufzählungen bei zahlreichen weiteren Autoren, vgl. z.B. [Wolf08, S. 5f.]. Für eine Einstufung als Embedded System genügt es bereits, wenn der Großteil der Eigenschaften zutrifft:

- **Verbindung zu Außenwelt:** Embedded Systems interagieren mit ihrer physikalischen Umwelt mittels Sensoren (Input) und Aktuatoren (Output).

[39] Die VHS Technik setzte sich schließlich trotz unterlegener technischer Fähigkeiten durch. Für weitere Details siehe [Cusu91].

- **Verlässlichkeit:** Bei sicherheitskritischen Anwendungen hat das System einen direkten Einfluss auf die Umgebung. Ein Ausfall kann dramatische bis hin zu katastrophalen Folgen nach sich ziehen. Daher müssen solche Systeme verlässlich funktionieren. Dazu sind folgende Eigenschaften erforderlich: Zuverlässigkeit und Wartbarkeit (um eine hohe Verfügbarkeit zu garantieren), Sicherheit (ein sicherer Zustand ist selbst bei Fehlfunktion oder Ausfall des Systems gewährleistet) und Integrität (Vertraulichkeit von Daten und sichere Kommunikation)

- **Effizienz:** An Embedded Systems werden vergleichsweise hohe Effizienzanforderungen gestellt. Einerseits basiert dies auf der mobilen Verwendung vieler Systeme, wodurch Energieverbrauch, Gewicht und Größe eine wichtige Rolle spielen. Andererseits werden solche Systeme meist in sehr hohen Stückzahlen gefertigt, wodurch eine starke Fokussierung auf die HW-Kosten entsteht. Daher werden lediglich die Ressourcen (Speicher und Rechenkapazität) integriert, die für einen Betrieb notwendig sind. Dies beeinflusst die Codegröße und die Effizienz der Berechnungen.

- **Single use:** Das System wurde für eine bestimmte Aufgabe entworfen. Dies umfasst explizit die Hardware und die Software.[40]

- **Benutzer Interface:** Vielfach nehmen Nutzer Embedded Systems nicht wahr, da sie keine Tastatur, sondern spezielle Benutzer-Interfaces, wie z.B. Knöpfe, Lenkrad, Pedale oder Mikrofon bieten.

- **Echtzeitsystem:** Wenn Systeme mit der Umwelt interagieren, ist meist eine Reaktion des Systems auf externe Eingaben innerhalb einer fest vorgegebenen Zeitspanne notwendig. Ein Überschreiten dieser Zeitspanne kann im leichten Fall zu Qualitätseinbußen (z.B. Bildausfälle bei einem digitalen TV-Receiver) führen. Hierbei spricht man von weichen Zeitbedingungen. Im schlimmsten Fall kann dies jedoch die Gefährdung der Benutzer und der Umwelt bedeuten (z.B. verzögerte Auslösung eines Airbags, verzögerte Umsetzung eines Bremsbefehls). Solche Anforderungen werden harte Zeitbedingungen genannt.

---

[40] Moderne, komplexe embedded Systems entfernen sich immer stärker von dieser Eigenschaft, schließlich erfordert die Integration zusätzlicher Funktionen und die zunehmende Konnektivität einzelner Komponenten eine zunehmende Variabilität.

- **Hybride Systeme:** Das Embedded System enthält sowohl analoge als auch digitale Schaltungen.

Zusätzlich zu den von Marwedel aufgeführten Eigenschaften können Embedded Systems vielfach auch hinsichtlich der folgenden Anforderungen abgegrenzt werden:

- **Robustheit:** Durch die Integration der Embedded Systems in ein Produkt ergeben sich meist deutlich härtere Anwendungsumgebungen, wie z.B. starke Erschütterungen bei Kinderspielzeug, Temperaturschwankungen bei Flugzeugen oder Staub und Schmutz bei Mobilfunkgeräten.
- **Systemstart:** Während PC-Nutzer gewohnt sind, dass das System eine gewisse Zeit zum Hochfahren benötigt, wird bei Embedded Systems im Allgemeinen eine sofortige Verfügbarkeit erwartet. Ein schneller Systemstart ist daher trotz zunehmender Komplexität des zugrundeliegenden Betriebssystems notwendig.[41]
- **Lebensdauer:** Neben der hohen Stückzahl werden einige Embedded Systems auch über einen relativ langen Zeitraum von mehreren Jahren bis hin zu einigen Jahrzehnten z.B. bei Flugzeugen gebaut. In Verbindung mit der Verwendung von bekannten, älteren Komponenten (z.B. Prozessoren) und der langen Nutzungs- und Wartungsphase stellt dies eine besondere Herausforderung an die Verfügbarkeit der elektronischen Bauteile.

3.4. Bedeutung der Software in Embedded Systems

Das erste Embedded System wurde vom amerikanischen Militär zur autonomen Steuerung von Raketen entwickelt. Das System aus der Minuteman Rakete ermöglichte eine wesentlich verbesserte Steuerung bei gleichzeitig reduziertem Gewicht durch weitgehenden Verzicht auf Verkabelung und analoge Technik. Zusätzlich konnte die Software nachträglich geändert und angepasst werden, was die Flexibilität erhöhte. Ein weiteres frühes aber deutlich komplexeres Embedded System war in den Apollo Raumkapseln installiert. Je Raumflug waren 2 Apollo Guidance Computer eingebaut, die die vollständige Steuerung übernehmen konnten. Dieses System wurde als eine der riskantesten Komponenten des Projektes angesehen.

---

[41] Die Ähnlichkeit von Blu-Ray-Playern und PCs kann man u.a. an der Startzeit der Geräte erkennen. Aktuelle Geräte benötigen bis zu 70 Sekunden, um mit dem Abspielen eines Films zu beginnen [Jurr09, S. 130 f.].

Die ersten massentauglichen Embedded Systems waren eher unterstützende Systeme in analogen Umgebungen, wie die Stabilisierung der Messung des Tankinhalts in Autos[42] oder die elektronische Steuerung der einzelnen Antriebsmotoren in Videorekordern (siehe Kapitel 3.1.). Sie übernahmen einfache Steuerungs-, Mess- und Kontrollfunktionen. Demzufolge waren die Anforderungen an die Software begrenzt. Aufgrund der hohen Kosten für Speicher und der beschränkten Rechenkapazität war das Ziel, einen sehr kompakten und performanten Programmcode zu erstellen. In den meisten Fällen war eine Programmierung direkt auf Hardwareebene notwendig. Durch die geringe Gesamtkomplexität ließ sich die Softwarequalität aber dennoch leicht sicherstellen. Wenn Probleme auftraten, waren sie meist in fehlerhafter Elektronik oder Mechanik begründet.

Die Verbesserung der Speicherkapazität und der Rechenleistung und die fortschreitende Miniaturisierung und Integration der Chips in Verbindung mit einem rapiden Preisverfall der Komponenten bot die Möglichkeit, den Funktionsumfang und damit die Komplexität der Systeme zu erhöhen. Gleichzeitig wurden die bisher einzelnen Systeme immer stärker vernetzt. Allmählich begann man, einige Funktionen komplett durch Software zu realisieren. Aktuelle Beispiele sind Fly-by-wire, also die elektronische Steuerung von Flugzeugen [vgl. From05], oder hochintegrierte Sicherheitssysteme im Automobil (z.B. Pre-Safe von Mercedes [SchV05]). Dadurch erhöhten sich sowohl die Komplexität der Software als auch die Qualitätsanforderungen. [Schä06, S. 21 f.]

Durch die stärkere Realisierung von Funktionen mittels Software ergeben sich für die Hersteller zahlreiche Vorteile (z.B. weniger Verkabelung, geringerer Bauraum, stärkere Individualisierung und Aktualisierung), aber auch einige Nachteile (z.B. unvollständige Testbarkeit, Gewährleistung der Sicherheit). Für weitere Details siehe z.B. [Schä06, S. 19 ff., Schä06, S. 142, Haus05 und Grot05, S. 268 f.].

3.5. Besonderheiten in der Automobilindustrie – Bereich PKW

Die Automobilindustrie weist begründet durch ihre Entstehungsgeschichte eine starke Ingenieursorientierung auf. In den letzten Jahren haben jedoch auch die OEMs und Zulieferer

[42] Durch die Fliehkräfte beim Fahren bewegt sich das Benzin im Tank, wodurch bei der Richtungs- oder Geschwindigkeitsänderung die Tankanzeige bei einer direkten Messung schwanken würde. Das Problem wurde durch eine Durchschnittsbildung über mehrere Messwerte gelöst, wodurch sich der angezeigte Wert trotz einzelner Abweichungen stabilisierte. Trotzdem kommt es heute immer noch vor, dass sie Tankanzeige sich verändert, wenn das Fahrzeig längere Zeit in einer schrägen Position steht.

die Bedeutung der Embedded Systems für ein modernes Automobil erkannt. Bereits 1998 startete BMW eine internationale Werbekampagne, die eine Apollo Rakete beim Start zeigte zusammen mit der Bildunterschrift „Wenn sie einen 7er BMW starten, aktivieren sie 20 Megabyte Rechenleistung (…) Das ist mehr als bei der Mondmission der Apollo 11." [Hoch00, S. 5]. Schätzungen gehen davon aus, dass rund 90% der zukünftigen Innovationen im Automobilbereich auf Embedded Systems basieren.[43] Bereits heute liegt der Anteil der Innovationen durch den Einsatz von Elektronik bei ca. 70 bis 90% [McKi06, S. 11]. Insbesondere bei Themen wie Verbrauchsreduzierung und Sicherheitssystemen sind kaum noch rein mechanische Innovationen möglich [vgl. Schä06, S.1].

Die bereits unter Punkt 3.3. angesprochenen Vorteile der Realisierung von Funktionen durch Software spielen bei Automobilen aufgrund des beschränkten Platzangebotes, der hohen Gewichtsbedeutung und der großen Stückzahlen eine noch wichtigere Rolle. Zusätzlich bieten elektronische Systeme durch den Wegfall mechanischer Komponenten eine höhere Zuverlässigkeit und haben vielfach einen geringeren Energiebedarf als mechanische Lösungen [Schä06, S.2]. Und schließlich bietet die Möglichkeit neue Funktionen durch Softwareupdates einzuspielen, ohne Hardwarekomponenten austauschen zu müssen, einen bedeutenden Vorteil.[44]

In der Automobilindustrie wurde die Verwendung von Embedded Systems zuerst im Bereich der Motorsteuerung vorangetrieben. Aufgrund des Öl-Shocks in den 1970ern und den steigenden Anforderungen an die Sauberkeit der Abgase ergab sich die Notwendigkeit, die einzelnen Vorgänge bei der Verbrennung wesentlich genauer zu steuern, als es rein mechanisch möglich wäre. Sophistizierte Steuerungsalgorithmen in Verbindung mit Sensoren und und Aktuatoren ermöglichten völlig neue Optimierungsmöglichkeiten [vgl. z.B. Wolf08, S. 2f.]

---

[43] Dieser Wert wird vielfach genannt, so z.B. von Hans-Georg Frischkorn, Vorstandsvorsitzender der BMW AG in München [vgl. Schä06, S. V].

[44] Aus der Fülle der hier möglichen realen Gegebenheiten soll lediglich eine kleine Auswahl von zwei nicht repräsentativen Beispielen angesprochen werden: BMW hat bei einer Überarbeitung seiner Motoren für die 7-er Serie die Leistung des 745d von 220 auf 242 KW gesteigert, bei gleichzeitiger Erhöhung des Drehmoments. Kunden, die noch die Version mit geringerer Leistung gekauft haben, können mittels Softwareupdate ebenfalls von der höheren Leistung profitieren. Bei VW ist es üblich, bei einer Inspektion die Software des Fahrzeugs auf den aktuellen Stand zu bringen. In vielen Fällen ergeben sich dadurch auch neue Funktionen. So wurde bspw. beim Touareg nachträglich die Motorsteuerung so angepasst, dass auch in einer extremen Höhe ein normales Ansprechverhalten des Motors gewährleistet war.

Heute betrifft der Einsatz von Embedded Systems alle Bereiche des Automobils. BMW hat z.B. die Blinkfunktion in seiner 7-er Modellreihe[45] ab Baujahr 2001 vollständig elektronisch realisiert. Die Steuerung erfolgt über das Kombiinstrument. Das Blinkergeräusch wird über die Lautsprecher des Radios wiedergegeben. Dadurch wird eine direkte Verkabelung aller Blinker unnötig, es genügt, sie an einen zentralen BUS und einen Stromanschluss anzuschließen. Ebenso ermöglicht dies eine automatische Rückmeldung im Falle einer Fehlfunktion, wie z.B. einer defekten Glühbirne. Allerdings war es in der ersten Modellgeneration z.B. möglich, dass das Blinkergeräusch nicht gleichmäßig wiedergegeben wurde, wenn das Navigationsgerät gerade eine Routenberechnung durchführte.

In einem Automobil gibt es zahlreiche Aufgaben, die zu einem festgelegten Zeitpunkt ausgeführt werden müssen, z.B. die Benzineinspritzung, die Zündung, der Bremseingriff des ABS Systems usw. Im Gegensatz zu zahlreichen Anwendungen aus dem Entertainmentbereich, die nicht zeitkritisch ausgeführt werden müssen, besteht hier eine funktionale Anforderung in der Einhaltung fester vorgegebener Zeitgrenzen für die Fertigstellung eines Prozesses. Dies wird als Echtzeitanforderung bezeichnet und bedeutet, dass ein elektronisches System eine Funktion in einem festgelegten Zeitintervall mit Sicherheit vervollständigt. Für weitere Informationen sei auf [Reif06, S.40] verwiesen. Bei einem ungleichmäßigen Blinkergeräusch ist dies sicherlich nicht kritisch, sondern höchstens irritierend. Was aber geschieht, wenn sich die Airbagauslösung aufgrund einer Routenberechnung um 2 Hundertstel Sekunden verzögert? Die Hersteller umgehen dies, indem sie echtzeitfähige Systeme für kritische Anwendungen verwenden und diese als eigenständige Systeme konzipieren.

Neben den im Beispiel genannten Nachteilen sind noch zahlreiche weitere Probleme beim Einsatz von Software im Automobil zu lösen. So sind die langen Produktlebenszyklen mit durchschnittlich 3 Jahren Entwicklungs- und 7 Jahren Produktionszeit [Schä06, S. 21] eine bedeutende Herausforderung in Anbetracht der schnellen Entwicklungszeiten im elektronischen Bereich. Hinzu kommt die Betriebs- und Servicephase nach Einstellung der Produktion, die nochmals bis zu 15 Jahre oder länger betragen kann [Grot05, S. 266]. So stellte bspw. der große Erfolg des mobilen MP3 Players iPod von Apple zahlreiche Automobilhersteller vor die Schwierigkeit, dem Kundenwunsch entsprechend eine einfache

[45] Das Beispiel bezieht sich auf die Baureihe E65, die von 2001 bis 2008 gebaut worden ist. Diese vierte Generation der 7er-Reihe führte erstmals das als iDrive bezeichnete Bedienkonzept von BMW in einem Serienfahrzeug ein.

Lösung für die Anbindung an die Musikanlage des Fahrzeugs anzubieten [Göth05, S. 260]. Die Anbieter versuchen dies durch eine stärker standardisierte Software-Architektur abzufangen, wie z.B. durch Entwicklung einer Middleware zur Entkopplung der Lebenszyklen von Fahrzeugen und externen elektronischen Geräten. Ebenso werden Software Funktionen stärker hardware-unabhängig realisiert, um ggfs. die Software schneller portieren zu können [Grot05, S. 267].

Abschließend sind die unterschiedlichen Einsatzbedingungen im Automobil zu nennen, die an elektronische Komponenten höchste Anforderungen stellen, sei es hinsichtlich Temperatur, Erschütterung, elektromagnetischer Verträglichkeit oder Stromverbrauch und variable Spannungsversorgung. Dabei kommt insbesondere der Berücksichtigung der Grenzen der im Automobil eingesetzten Batterie eine große Bedeutung zu [Case05, S. 662].

Als Einsatzbereiche von Embedded Systems im Automobil lassen sich 5 Bereiche unterscheiden: Infotainment/Multimedia (z.B. ein Navigations- oder Entertainmentsystem), aktive und passive Sicherheit (z.B. ABS und ESP), Motor/Antriebsstrang (z.B. Motorsteuerung, Lambda Sonde), Karosserie (z.B. Funkfernbedienung zur Zutrittkontrolle) und Fahrwerk (z.B. Active Body Control; Luftfederung). In der Literatur findet man teilweise auch 4 Bereiche, dann werden die Sicherheitskomponenten dem Fahrwerk und der Karosserie zugeordnet [vgl. Schä06, S.6].

Insbesondere neue Systeme zur Steigerung der aktiven Sicherheit greifen immer stärker in die Funktionen des Automobils ein, sei es bei Bremseingriffen, der Motorsteuerung oder sogar wie z.B. bei Lane-Assist- oder Einpark-Systemen in die aktive Steuerung des Fahrzeugs. Solche Eingriffe stellen natürlich ganz neue Anforderungen an die Sicherheit elektronischer Systeme und der darin enthaltenen Software. Eine Fehlfunktion bedeutet eine Bedrohung der Insassen. Daher bestehen strenge Anforderungen an die Funktionssicherheit von elektronischen Systemen in Fahrzeugen, wie z.B. DIN 19250 [DIN89] oder IEC 61508 [IEC05].

4. Besonderheiten bei der Anwendung von Software Produktlinien in Embedded Systems

Der Grundgedanke von Produktlinien, durch gezielte Produktanalyse diejenigen Teile zu identifizieren, bei denen sich eine strategische Wiederverwendung lohnt, ist intuitiv. Ebenso verwundert die Entwicklung der Idee aus dem Hardwarebereich heraus nicht. Die in der nahen Vergangenheit in vielen Bereichen gestiegene Variantenzahl, durch die versucht wurde, die individuellen Kundenwünsche besser abzudecken, erforderte eine neue Betrachtung der Wiederverwendung. Wurden bisher Wiederverwendungsentscheidungen meist zufällig oder adhoc getroffen, erschien nun ein strategisches Vorgehen bei der Planung der Wiederverwendung notwendig. Für die Übernahme dieser Idee in den Softwarebereich gab es neben der geänderten Nachfrage einen weiteren Impuls, der die erfolgreiche Umsetzung der Produktlinienidee erst ermöglichte. Objektorientierte Programmierung und die Idee der Modularisierung sind notwendige Voraussetzungen für Software Produktlinien. Erst dadurch wird es möglich Software in Teilbereiche zu zerlegen und trotzdem ein späteres einfaches Zusammenfügen von Komponenten zu einem neuen Produkt sicher zu stellen.

Interessanterweise dauerte es mehrere Jahre, bis die SWPL Idee auch für Embedded Systems aufgegriffen worden ist. Das aktuelle Interesse an strategischer Wiederverwendung liegt in der gestiegenen Bedeutung dieser Systeme und dem zu beobachtenden Anstieg der Systemkomplexität begründet. Es ergeben sich allerdings bei der Anwendung der Produktlinienidee für Embedded Systems besondere Herausforderungen, die im Folgenden aufgezeigt werden.

4.1. Hardware Fokus im Entwicklungsprozess

Bei klassischen PCs hat sich bereits früh ein Wandel von einer Hardware-Orientierung hin zu einem Software-Fokus ergeben. Einen nicht unerheblichen Anteil dürfte IBM durch die Schaffung eines Quasi-Industriestandards mit seinem IBM-PC gehabt haben. Zusätzlich wurde der Trend von der enormen Leistungssteigerung der Hardware unterstützt. Moderne PCs bieten durch ihre universelle Ausrichtung weit mehr Leistung als für die Standardtätigkeiten, wie Dokumente bearbeiten oder Videos abspielen, notwendig ist. Allein aktuelle Computerspiele und aufwendige Grafikanwendungen (z.B. Videobearbeitung) benötigen die Leistungen der aktuellen Produkte. Die universelle Ausrichtung ermöglicht es,

bei unveränderter Hardware komplett neue Funktionen mittels Software zu realisieren. Die reine Hardware wurde ein Commodity, also austauschbar. Die wirkliche Leistung entsteht allein durch die Funktionen der Software. Dies gilt auch immer stärker für Embedded Systems. Apple hat bei seinem revolutionären iPhone mit dem Multitouch-Screen zwar ebenfalls eine neue Hardware verwendet, der wirkliche Nutzen ergibt sich aber wieder aus der entsprechenden Software. Erst die neuen intuitiven Funktionen machten das iPhone so erfolgreich. Interessanterweise profitierten auch Nutzer der ersten iPhone Generation von den neuen Features der zweiten Generation des Produkts. Ein Großteil der Neuerungen ist softwarebasiert und damit auch auf das alte iPhone per Softwareupdate übertragbar.

Gerade bei Unternehmen mit einen Ingenieurhintergrund, die sich Jahrzehnte mit der Verbesserung von Fertigungsprozessen und der Verringerung von Fehlern in der Hardware beschäftigt haben, besteht ein geringes Problembewusstsein für Software. Es erfordert einige Zeit, bis die gewandelte Bedeutung der Software und die daraus resultierenden Möglichkeiten, aber auch Herausforderungen erkannt und akzeptiert werden. Ein leitender Mitarbeiter eines deutschen Automobilherstellers sagte bspw. "Software ist nicht unser Problem"[46]. Er vertrat die Ansicht, dass Probleme im Automobil meist auf fehlerhafte Hardware, ein loses Kabel, eine defekte Lötstelle oder einen Platinenfehler zurückzuführen sind. Software habe nach seiner Meinung daran nur einen sehr geringen Anteil.

Die ADAC Pannen-Statistiken deuten allerdings auf einen anderen Zusammenhang hin, insbesondere in den Jahren bis 2004. Der Anteil der durch Elektronikprobleme verursachten Ausfälle stieg von 1999 bis 2004 von 50,5% bis auf einen Wert von über 59% an [Dude05, S.20]. Durch einen hohen Testaufwand ist es den Herstellern seither gelungen diesen Wert wieder abzusenken auf ca. 40% in der aktuellen Pannenstatistik 2008 [ADAC09, S. 4]. Nur ist diese Vorgehensweise ökonomisch wenig sinnvoll. Die alte Sichtweise mit einem Fokus auf die technische Hardware und einer Betrachtung der Software als dazu passende einfach zu erstellende Lösung, muss abgelegt und neue Softwareentwicklungsmethoden übernommen werden. Mit ersten erfolgreichen Versuchen mit modellbasierter Entwicklung sowie im Bereich der Software-Produktlinien wird der zukünftig einzuschlagende Weg aufgezeigt.

Problematisch ist, dass die hohen Stückzahlen im Automobilgeschäft einen hohen Kostendruck bei der Hardware induzieren. Bei der Konzeption eines neuen Modells spielen

---

[46] Das Zitat entstammt einem persönlichen Gespräch im Rahmen der Erstellung dieser Arbeit, wobei die Person ungenannt bleiben möchte.

bereits Kostenpositionen von wenigen Cent eine entscheidende Rolle. Die Hardware wird entsprechend optimiert. In Verbindung mit der großen Zahl an Ausstattungsvarianten ergibt dies teilweise unlogisch erscheinende Zuordnungen von Funktionen auf einzelne Steuergeräte. Selbst für den Kunden sind die Auswirkungen teilweise spürbar, z.B. wenn einzelne Ausstattungen nur in Kombination mit anderen, inhaltlich völlig unabhängigen Funktionen verfügbar sind.

Für die Verwendung von Produktlinien bedeuten insbesondere die kostenbedingten Einschränkungen bei der Hardware eine grundlegende Schwierigkeit. Produktlinien ermöglichen Wiederverwendung durch Standardisierung von Komponenten und unter Inkaufnahme von Redundanzen. Diese Redundanzen führen allerdings regelmäßig zu höheren Anforderungen an die Hardware, insbesondere bei Speicher und Rechenleistung. Bedingt durch die hohen Stückzahlen im automobilen Umfeld müssen daher die angestrebten Einsparungen im Entwicklungsbereich deutlich höher liegen, um die eventuell resultierenden Zusatzkosten während der Produktionsphase zu rechtfertigen.

4.2. Nutzung von standardisierten Methoden und Werkzeugen

Mit der zunehmenden Komplexität der elektronischen Systeme im Automobil stieg auch der Bedarf an standardisierten Methoden und Werkzeugen zur Softwareerstellung. Grundlegende Ideen wie modellbasierte Entwicklung und integrierte Qualitätssicherung wurden aus der klassischen Softwareentwicklung für Bürosoftware weitgehend übernommen und entsprechend angepasst. Weit verbreitet in der Automobilindustrie ist heute der aus der allgemeinen Softwareentwicklung bekannte V-Ansatz [vgl. Reif06, S. 47ff.], bei dem sowohl eine Trennung zwischen System und Software, als auch eine Trennung zwischen Analyse/Design/Implementierung und Testing besteht. Eine Besonderheit im Bereich der automobilen Softwareentwicklung besteht darin, dass das V-Konzept in der Realität meist inkrementell und/oder iterativ durchlaufen wird. Dabei werden einzelne Schritte oder das gesamte V-Modell mehrfach durchlaufen [Schä06, S. 27].

Aktuell etabliert sich die modellbasierte Entwicklung (im englischen Model-Driven Software Development, MDSD), da hierdurch die interdisziplinäre Zusammenarbeit der Entwickler vereinfacht und unterstützt wird. MDSD hat das Ziel, "domänenspezifische Abstraktionen zu finden und diese einer formalen Modellierung zugänglich zu machen. Dadurch entsteht ein hohes Automatisierungspotenzial für die Fertigung von Software, und dies wiederum lässt

sich in den allermeisten Fällen in eine deutliche Produktivitätssteigerung umsetzen. Auch verbessern sich dadurch Qualität und Wartbarkeit von Softwaresystemen. Zudem können die Modelle von Domänenexperten verstanden werden." [Stah07, S.4] Eine Anwendung findet MDSD im Rapid-Prototyping, denn durch die formale Beschreibung aller Anforderungen ist es möglich bereits früh Simulationen des geplanten Produktes zu realisieren. Der Simulation kommt bei Embedded Systems eine wesentlich größere Bedeutung zu. Da die endgültige Spezifikation bei Entwicklungsbeginn oft nicht vollständig vorliegt, ist es notwendig mittels Laborfahrzeugen bereits früh erste Erfahrungen zu sammeln und die geplanten Software-Funktionen zu validieren. Erschwert wird dies durch die starke unternehmensübergreifende Aufgabenteilung bei der Fahrzeugentwicklung. Die endgültige Kalibrierung der Parameter erfolgt meist erst im fertigen Fahrzeug.

Der eigentliche Produktivitätsvorteil der MDSD erfolgt duch die Verwendung von Methoden zur automatischen Codegenerierung. Dadurch wird es möglich, den Aufwand für die Realisierung der Software in den Steuergeräten zu minimieren. Änderungen, die sich aus den zahlreichen Tests ergeben, können zeitnah und ohne großen Aufwand umgesetzt werden. Dadurch wird sichergestellt, dass die Tests immer mit einer möglichst aktuellen und fehlerfreien Version der Software durchgeführt werden.

Der Verwendung von standardisierten Methoden und Werkzeugen sind allerdings durch die Vielzahl unterschiedlicher Systeme bei Automobilen Grenzen gesetzt. Die Hersteller und Zulieferer haben dies bereits vor einiger Zeit erkannt. Aus diesem Grund wurde u.a. die Association for Standardization of Automation and Measuring Systems (ASAM)[47] gegründet. ASAM verfolgt das Ziel, die Schnittstellen, Protokolle und Datenformate im Automobil zu vereinheitlichen. Im Bereich nicht-wettbewerbsrelevanter Anwendungen wurden bereits einige Ziele erreicht. In Bereichen, die allerdings eine gewisse Relevanz für die Produktdifferenzierung bieten, konnten bisher keine signifikanten Erfolge erzielt werden. Dies gilt ebenso für den Versuch von AUTOSAR eine einheitliche Softwareentwicklungsumgebung zu realisieren (siehe dazu auch das folgende Kapitel). Die Gründe hierfür sind vielfältig, liegen aber überwiegend im Schutz eigener Wettbewerbsvorteile und der Vermeidung eines zu großen Wissenstransfers hin zu den Zulieferern. Aufgrund der klaren Vorteile einer weitgehenden Standardisierung von Schnittstellen ist jedoch in Zukunft von Erfolgen in diesem Bereich auszugehen.

[47] Für weitere Informationen sei auf die Webseite des Vereins verwiesen: www.asam.net.

Generell kann man festhalten, dass zahlreiche Methoden und Werkzeuge zur Softwareerstellung im automobilen Embedded Bereich verwendet werden. Ansonsten wäre die angestrebte Verkürzung der Entwicklungszeiten trotz einer zunehmenden SW-Systemkomplexität nicht zu realisieren. Andererseits ist der Reifegrad bei der Verwendung der Prozesse und Werkzeuge im Allgemeinen niedriger als im Vergleich mit der Softwareentwicklung für PCs. Dies ist umso bedeutender, wenn man bedenkt, dass sich ein exponentieller Zusammenhang zwischen Reifegrad der Prozesse und Werkzeuge innerhalb eines Unternehmens und der Fehleranzahl in der Software erkennen lässt (siehe Abbildung 7). Für die Anwendung von SWPL bedeutet dies, dass Unternehmen, die bereits einen hohen Reifegrad bei ihren Prozessen und Werkzeugen erreicht haben, generell weniger Schwierigkeiten bei der Umsetzung einer SWPL haben werden und die Erfolgsaussichten damit ansteigen.

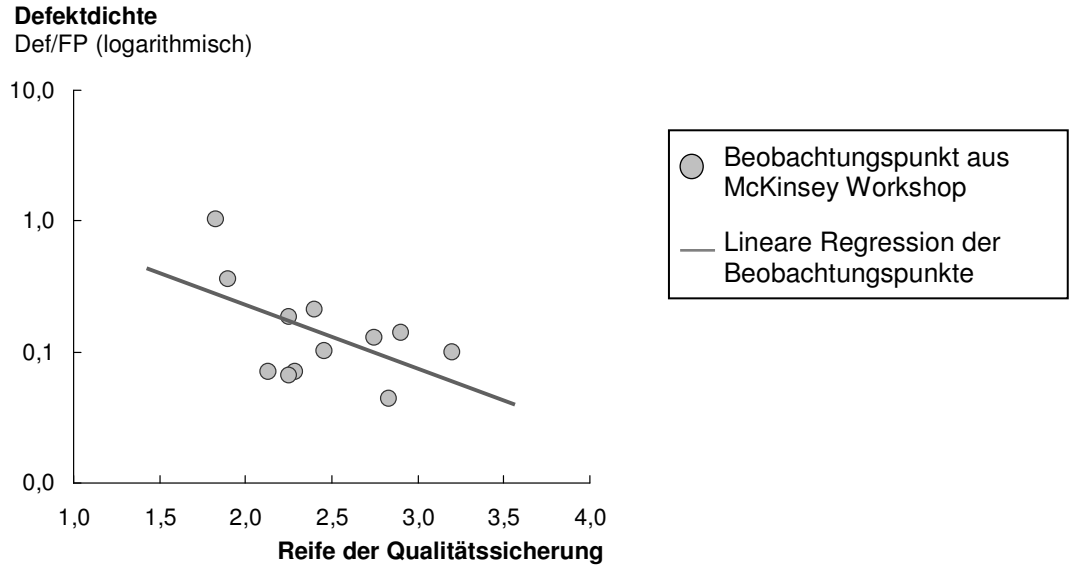

Abbildung 7 – Zusammenhang zwischen Prozessreife und Softwarequalität [McKi06, S. 40]

4.3. Beziehungen zwischen OEM und Zulieferern in der Automobilindustrie

In den letzten Jahrzehnten hat sich der Anteil der OEM-eigenen Wertschöpfung am fertigen Automobil immer weiter reduziert. Inzwischen werden ganze Baugruppen von Zulieferern entwickelt und produziert. Zukunftsweisende Innovationen in sicherheitskritischen Bereichen, wie bspw. ABS und ESP, wären für einen OEM alleine gar nicht realisierbar. Eine Kooperation mit einem oder mehreren erfahrenen Zulieferern ist zwingend notwendig. In der

letzten Zeit sind auch immer stärker Entwicklungskooperationen zwischen den OEMs selbst zu erkennen, insbesondere wenn es sich um den Bereich ihrer Kernkompetenzen handelt, wie z.B. Hybridantriebskonzepte.

Dennoch findet in der Automobilindustrie vielfach eine Auftragsvergabe nach dem Motto "ein Steuergerät – ein Zulieferer" statt. Ein wichtiges Erfolgskriterium für Produktlinien bei Zulieferern sind die deutlich häufigeren Wiederverwendungen der Assets der Produktlinie, da die Produkte mehreren OEMs angeboten werden können. Vielfach ist nun aber die Verteilung der Software auf die Steuergeräte nicht allein funktional orientiert, sondern vielfach historisch gewachsen, das Ergebnis eines komplexen Optimierungsprozesses zur Platz- und Kostenreduktion oder systembedingt[48]. Die Realisierung einer Produktlinie bei einem Steuergerät mit zahlreichen unterschiedlichen Funktionen ist problematisch, denn kaum ein Unternehmen kann alle teilweise willkürlich zusammengeführten Funktionen anbieten und zu einer Software-Produktlinie zusammenführen. Ebenso unwahrscheinlich erscheint derzeit die Realisierung einer Produktlinie über die Unternehmensgrenze hinweg.

Für eine solche SWPL wäre eine standardisierte Architektur wie z.B. bei AUTOSAR[49] notwendig. Da AUTOSAR inzwischen fertig spezifiziert ist und die OEMs beginnen, erste Anwendungen AUTOSAR-konform zu realisieren, dürfte dies allerdings nur noch eine Frage der Zeit sein. Dies ist umso wichtiger, als dass bereits heute ein Trend hin zu "eine Funktion – ein Zulieferer" erkennbar ist. Die Gründe hierfür sind die zunehmende Konsolidierung von Steuergeräten, maßgeblich angestoßen durch eine zunehmende Zahl an elektronischen Funktionen bei gleichbleibendem, begrenztem Bauraum. Nach [Buhr09] existieren zwei erfolgversprechende Kooperationsmodelle: einerseits die Wiederverwendung von Anforderungen und Testfällen durch Abstraktion. Die Assets werden soweit abstrahiert, dass sie herstellerunabhängig wiederverwendet werden können. Die Instanziierungen werden dann eindeutig als Variation gekennzeichnet. Dadurch wird der Zulieferer in die Lage versetzt, sein Asset häufig wiederzuverwenden. Dies zahlt sich für den OEM durch höhere Qualität und

---

[48] Eine solche systembedingte Kopplung ergibt sich bspw. bei Motorsteuergeräten, da diese in der unmittelbaren Nähe zum Motor angebracht werden müssen. Ansonsten könnten aufgrund der Latenzzeiten bei der kabelgebundenen Datenübertragung die notwendigen kurzen Steuerzeiten nicht erreicht werden.

[49] AUTOSAR steht für Automotive Open System Architecture. Das Konsortium wurde von zahlreichen Automobilherstellern, Zulieferern und Werkzeugentwicklern gegründet mit dem Ziel, die Wiederverwendbarkeit von Software im Automobil durch eine Trennung von Software und Hardware zu erhöhen. AUTOSAR bietet über die eigene Runtime Environment (RTE) eine Abstraktionsebene, die sämtliche Kommunikation der Software mit der Hardware steuert und darüber eine Unabhängigkeit herstellt. Für Details siehe [Scha05a, Hein04 und die Internetseite von AUTOSAR www.autosar.org]

niedrigere Kosten aus. Die zweite weitergehende Möglichkeit besteht in der Verknüpfung der Produktlinien des Zulieferers und des OEMs. "Kann man sich mit dem Zulieferer über eine klare Schnittstelle und ein eindeutiges Austauschformat einigen, können beide Produktlinien gekoppelt werden, wobei auftraggeberseitig Anforderungen, System- und Komponentenarchitektur sowie abstrakte Tests in die Produktlinie einfließen, während zuliefererseitig die Software-Architektur, Code und konkrete Tests bereitgestellt werden" [Buhr09, S.3]. Wieder werden unabhängige Assets ausgetauscht, wodurch der Zulieferer Entwicklungszeit und –Kosten sparen kann und von einer höheren Produktqualität ausgegangen werden kann. Wettbewerbsrelevante Informationen werden als Variation gekennzeichnet.

4.4. Geschwindigkeit der technischen Entwicklung

Ein Merkmal der Automobilindustrie sind die relativ langen Produktlebenszyklen. Obwohl sich die Produktionsdauer in den letzten Jahren deutlich reduziert hat, erreicht die Automobilindustrie immer noch beachtlich Zeiträume für ein Massenprodukt. Im Durchschnitt kann heute von einer Produktionsdauer von ca. 7 Jahren ausgegangen werden [Schä06, S. 20]. Größere Veränderungen finden nur im Rahmen der jährlichen Modellpflege statt. Die technische Entwicklung bzw. ihre Umsetzung im Automobil geschieht demnach langsamer als bei anderen Embedded Systems. Dies ist u.a. bedingt durch die erhöhten Sicherheitsanforderungen bei im Automobil verwendeten elektronischen Systemen. Beachtet werden sollte zusätzlich die Wartungsdauer. Einige Hersteller bieten bspw. für ihre Modelle nach Ende der Produktion noch eine garantierte Verfügbarkeit von Ersatzteilen von bis zu 20 Jahren. Im Durchschnitt ist jedoch mit einer Zeit für Betrieb und Service der Produkte nach Einstellung der eigentlichen Produktion von 10 bis 15 Jahren zu rechnen. Dieser Zusammenhang wird in Abbildung 8 dargestellt.

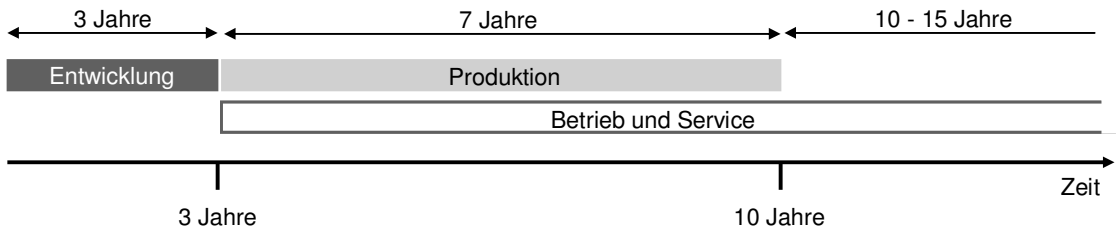

Abbildung 8 – Produktlebenszyklus eines Fahrzeugs [nach Schä06, S. 21]

Im Gegensatz zu Automobilen haben z.B. Mobiltelefone einen drastisch kürzeren Lebenszyklus von 6-12 Monaten [Grote05, S. 266]. Die großen Mobilfunkhersteller erstellen meist jedes Jahr komplett neue Hardware-Plattformen, um die technischen Veränderungen berücksichtigen zu können. Auch von einer langfristigen Wartung kann nicht gesprochen werden. Ersatzgeräte oder Austauschkomponenten stammen meist aus der eigentlichen Produktion. Nach Einführung eines neuen Modells erfolgt bei einem Reklamationsfall vielfach ein kompletter Austausch des Geräts. Spätestens bei der Kombination verschiedener technischer Geräte mit einem Automobil kann dies zu Komplikationen führen, bspw. wenn der Käufer eines Neuwagens sein Handy mit der Freisprecheinrichtung verbinden möchte. Durch die Nutzung von Standardschnittstellen, wie Bluetooth, konnte dies teilweise abgefangen werden [Göth05]. In anderen Bereichen besteht allerdings noch Handlungsbedarf. Hier seien nur zwei Beispiele genannt: Integration von Navigationsgeräten unterschiedlicher Hersteller in die Kommunikation des Fahrzeugs, Anschluss von externen Multimediageräten an das Soundsystem.

Auch wenn die langen Zyklen auf den ersten Blick für den Einsatz einer Produktlinie vorteilhaft erscheinen, stellt dies allerdings genauso eine Herausforderung dar. Denn aufgrund der längeren Zeiten zwischen zwei Updates fallen die Veränderungen deutlich größer aus. Dies würde für die Domain einer SWPL bedeuten, dass große Anpassungen notwendig wären, oder im Extremfall eine Wiederverwendung unmöglich ist. Zulieferer haben den Vorteil, dass sie die Domain vor einer technisch bedingten Anpassung aufgrund der Belieferung mehrerer OEMs stärker nutzen können. Für die OEMs besteht außerdem die Möglichkeit, die Systeme über verschiedene Fahrzeugklassen hinweg zu nutzen. Audi hat bspw. bereits im Jahr 1999 eine Basis-Architektur für die gesamte Elektronik entwickelt, die bis 2008 galt. Basierend auf dieser Basis wurden die Elektronik des Audi A8 (SOP[50] 2002), des Audi A3 (SOP 2002), des Audi A6 (SOP 2004) und des Audi Q7 (SOP 2005) entwickelt [Kötz05, S. 493]. Erkennbar ist dieses Vorgehen z. B. an Audis MMI System, das als zentrale Funktionsschnittstelle eine wichtige Rolle bei der Festlegung der Basis-Architektur spielte. Ebenso hat BMW bei der Entwicklung der Fahrzeuge der aktuellen 3er-Serie versucht, mittels einer Vereinheitlichung des Kommunikationsbordnetzes Funktionen aus den Modellen BMW 7er, BMW 6er und BMW 5er möglichst einfach auf den BMW 3er zu übertragen [Wagn05, S. 495 ff.].

[50] SOP steht für Start of Production und bezeichnet den Produktionsstart.

Eine andere Lösung besteht in der stärkeren Nutzung von hardwareunspezifischer und konfigurierbarer Software, wie sie z.B. in AUTOSAR angestrebt wird. Für eine Übersicht über die Erfolge und Herausforderungen von AUTOSAR siehe [Scha05a, S. 395 ff.]. Der Druck der Kunden, Innovationen zeitnah auch im Automobil realisiert zu sehen, wird wahrscheinlich die Produktlebenszyklen in der Automobilindustrie, zumindest bei den elektronischen Komponenten, weiter verkürzen. Dadurch werden SWPL für Automobilhersteller interessanter, denn mit deren Hilfe wäre eine größere Zahl individueller Produkte in kürzerer Zeit möglich. Allerdings sind die oben angesprochenen Maßnahmen, wie eine Standardisierung der Schnittstellen, notwendige Vorraussetzungen für einen Erfolg.

5. Übersicht über Bewertungsmöglichkeiten des Entwicklungsaufwands bei Softwareprojekten

Die Schwierigkeit einer jeden ökonomischen Analyse von Softwareprojekten stellt die frühzeitige Abschätzung des benötigten Entwicklungsaufwandes dar. Diese Information ist sowohl für den Entwickler als auch den Nutzer oder Kunden relevant. Beide Seiten interessieren sich für die vermutliche Entwicklungsdauer und die dabei entstehenden Kosten. Meistens verlassen sich die Beteiligten auf Erfahrungswerte, um zu einer ersten groben Schätzung zu gelangen [Stah05, S. 462]. Bei der Neu- oder Weiterentwicklung eines Produktes, bei der auf umfangreiche Erfahrung zurückgegriffen werden kann, kann sich dies als ausreichend genau erweisen. Bei der Entwicklung von innovativen Produkten besteht jedoch eine wesentlich größere Ungewissheit hinsichtlich der späteren Realisierung. Dies gilt ebenso für die Abschätzung der Auswirkungen von neuen innovativen Entwicklungsansätzen, wie z.B. Produktlinien.

Ein Automobilhersteller dürfte relativ genau abschätzen können, wie hoch die Kosten der Entwicklung eines neuen Motorenmodells unter gleichbleibenden Rahmenbedingungen sind, und welche Entwicklungszeit erforderlich ist. Dies betrifft ebenfalls die notwendige Software zur Motorensteuerung. Sollte sich das Unternehmen allerdings entschließen, die Entwicklung auf eine Produktlinie umzustellen und damit grundlegende Rahmenbedingungen zu ändern, dürfte eine Schätzung aus dem "Bauchgefühl" unmöglich oder zumindest beinahe erratisch sein. Zusätzlich erschwert das Fehlen einer standardisierten Methodik den Vergleich mit anderen Entwicklern oder Konkurrenzunternehmen. Zwar lassen sich die reine Zeit und der Personenaufwand vergleichen, die Komplexität des Produktes bleibt jedoch unbeachtet.

Interessanterweise finden sich in der wissenschaftlichen Literatur zahlreiche Ansätze zur Abschätzung des Entwicklungsaufwands physischer Produkte. Zu nennen wären hier bspw. Target Costing und Life Cycle Costing [vgl. Adam98, S. 192 ff.]. Das Target Costing orientiert sich am von Kunden akzeptierten Preis für eine Bündelung von Funktionen. Anschließend werden die Kosten je nach Nutzen der Teilfunktionen aufgeteilt. Daraus ergeben sich die erlaubten Kosten, die eine Komponente verursachen darf. Beim Life Cycle Costing werden zusätzlich die Kosten über die gesamte Nutzungszeit des Produktes berücksichtigt, z.B. Wartung und Entsorgung. Eine Übertragung auf den Softwarebereich ist unmöglich, da sich die genannten Verfahren auf die Produktionskosten und nicht die

Entwicklungskosten konzentrieren. Die reinen Produktionskosten bei Software betragen aber aufgrund der einfachen Reproduzierbarkeit beinahe null. Sinnvoller erscheint ein Vergleich mit Methoden aus dem Bereich der Industriegüter, wie z.B. Großanlagen, Kraftwerke, Containerschiffe, usw. Solche Produkte weisen in der Regel eine sehr niedrige Stückzahl auf, oft wird sogar nur ein Exemplar hergestellt. Die entsprechenden Verfahren zur Preisbestimmung werden Angebotskalkulationsverfahren genannt, da es sich meist um individuelle Angebote handelt. Bekannte Verfahren sind [vgl. Back99, S. 494 ff.]:

- Einfach oder Mehrfachregression über verschiedene Einflußgrößen (z.B. Kubikmeter umbauter Raum als Schätzgröße für die Herstellkosten eines Hauses)
- Modifikationspreisansatz (als Basis dienen alte Projekte und es werden lediglich Modifikationen in der Kalkulation durchgeführt, um Abweichungen vom Ausgangsprojekt zu berücksichtigen)
- Grobprojektierungsansatz (die Kalkulation basiert auf einem groben Mengengerüst mit Preisen für die einzelnen Komponenten, die weitgehend bekannt sind).

Es fällt auf, dass alle verwendeten Methoden auf Erfahrungswerte und Schätzgrößen zurückgreifen, um einen Näherungswert zu erhalten.

Eine direkte Übertragung dieser Ansätze auf Softwareprojekte ist nicht möglich [Back99, S. 503], aber die Grundidee des Rückgriffes auf Erfahrungswerte hat sich auch hier als zielführend erwiesen. Sämtliche Methoden, die sich in der Praxis als praktikabel und/oder nützlich erwiesen haben, greifen darauf zurück. Zu beachten ist, dass es sich bei allen Ansätzen lediglich um Näherungslösungen handelt, da die Softwareentwicklung ein komplexer Prozess ist und zusätzlich zahlreiche externe Einflussfaktoren eine Rolle spielen. [Jone07, S. 68ff.] unterscheidet 12 wesentliche Klassen von Fehlern bei der Aufwandsabschätzung:

- **Metrics errors:** Fehler aufgrund der Verwendung fehlerhafter Methoden, wie z.B. der Lines-of-Code. Obwohl diese Methode bereits Ende der 1970er Jahre nachweisbar als unzuverlässig eingestuft worden ist, wird sie bis heute vielfach verwendet.
- **Scaling errors:** Fehler durch die direkte Hochrechnung von Ergebnissen kleiner Projekte auf Projekte mit deutlich größerem Umfang, ohne die immanenten

Systemunterschiede zu berücksichtigen, wie z.B. die deutlich höhere Anzahl an allgemeinen Aufgaben (z.B. Koordination).

- **Executive and client errors:** Fehler durch das Eingreifen von hochrangigen Managern. Diese können valide Schätzungen teilweise aufgrund von politischen oder ökonomischen Motiven zurückweisen und nach ihren Vorstellungen anpassen lassen.
- **Sizing errors:** Fehler durch die falsche Voraussage des Umfangs benötigter Endprodukte, wie z.B. die Anzahl der Bildschirmseiten oder die Seiten der Dokumentation
- **Activity-selection errors:** Fehler durch die Nichtberücksichtigung von relevanten Aufgaben, wie z.B. die Erstellung der Dokumentation oder wichtiger Testschritte. Dies kann insbesondere bei der Vertragsgestaltung mit externen Dienstleistern eine wichtige Rolle spielen. Hier sollte ein gemeinsames Verständnis über den genauen Vertragsumfang erzielt werden.
- **Assignment-scope errors:** Fehler durch eine falsche Kalkulation des Arbeitsumfangs, der von einer Person erledigt werden kann. Dies führt zu einer Überbelastung der Mitarbeiter, die sich langfristig auf die Projektentwicklung durchschlagen wird.
- **Production-rate errors:** Eng verbunden mit dem vorherigen Punkt bezeichnet dies Fehler bei der Abschätzung der Produktivität der Mitarbeiter, also des Arbeitsumfangs, den eine einzelne Person innerhalb einer bestimmten Zeit erfolgreich erledigen kann.
- **Creeping user requirements:** Fehler durch die Nichtbeachtung von im Verlauf der Entwicklung anfallenden zusätzlichen Funktionen. Das Hinzufügen von neuen Anforderungen zu einem Softwareprojekt nach dem offiziellen Schluss der Anforderungsphase ist so häufig, dass dies direkt bei der Kalkulation des Aufwands berücksichtigt werden sollte.
- **Critical path errors:** Dies bezieht sich hauptsächlich auf zeitliche Verzögerungen von Softwareprojekten, da der kritische Zeitpfad nicht korrekt identifiziert und abgesichert worden ist. Jede Verzögerung innerhalb dieses Pfades hat direkte Auswirkungen auf den gesamten Projektverlauf. Diese Zusammenhänge sollten transparent gemacht werden, um entsprechend reagieren zu können.

- **Staffing build-up errors:** Bei zahlreichen Softwareprojekten verfügt das Unternehmen zu Projektbeginn nicht über die gesamte Zahl an notwendigen Mitarbeitern. Sollte es beim Aufbau der Kapazitäten (intern oder extern) zu Verzögerungen kommen, dann wirkt sich dies direkt auf die Projektplanung aus. In vielen Fällen wird dies nicht berücksichtigt.
- **Technology adjustment errors:** Software-Entwicklung ist verglichen mit anderen Bereichen noch ein relativ junges Gebiet, in dem sich schnell gravierende Veränderungen in der verwendeten Technologie und den entsprechenden Methoden ergeben können. Dies sollte bei der Aufwandsschätzung berücksichtigt werden. Vielfach sind die ersten Projekte bei Verwendung einer neuen Methodik (z.B. Objekt-orientierte Programmierung oder die Einführung von SWPL) nicht sonderlich produktiv. Aber aufgrund einer steilen Lernkurve dreht sich dies in vielen Fällen mit einem der Folgeprojekte um.
- **Special or unique situations:** Ein sehr schwer zu berücksichtigender Faktor sind außergewöhnliche Situationen, wie z.B. Naturkatastrophen, massenhafte Erkrankungen (Grippewelle) oder Streiks.

All diese Fehlerquellen können einen wichtigen Einfluss auf die gesamte Aufwandsschätzung haben. Teilweise reichen die Auswirkungen bis zu einem Faktor 10, wie z.B. bei Scaling errors oder Activity-selection errors. Das Ziel muss es daher sein, diese Fehler so weit wie möglich zu verhindern oder zumindest zu minimieren.

Ein zusätzliches Problem bei der Aufwandsschätzung liegt in der Anforderung, die Schätzung möglichst früh im Verlaufe des Softwareprojektes durchzuführen. Insbesondere zu Beginn des Projektes zeigen Aufwandsabschätzungen große Abweichungen. Abbildung 9 zeigt die Ergebnisse einer Untersuchung zu den zu erwartenden Genauigkeiten in verschiedenen Projektphasen. Die einzelnen Phasen bilden ein 5-stufiges Wasserfallmodell ab. Man erkennt, dass mit dem Beginn der Vorstudie, also der Anforderungsanalyse, eine Genauigkeit von lediglich +/-50% erreicht werden kann. Erst in der Hauptstudie, dem Systemdesign und der –Spezifikation, erhöht sich die Genauigkeit auf +/-20%. Dieser Wert entspricht dem von Boehm bereits 1981 geforderten Korridor für ausreichend genaue Aufwandsschätzungen für

Softwareprojekte [Boeh81].[51] Dies verdeutlicht erneut die Relevanz der Aussage, dass Aufwandsbewertungen lediglich eine Näherungsgröße sein können. Als Anwender dieser Analysen sind diese Besonderheiten zu berücksichtigen. So ist bspw. die Frage zu beantworten, welche Abweichung tolerierbar ist.

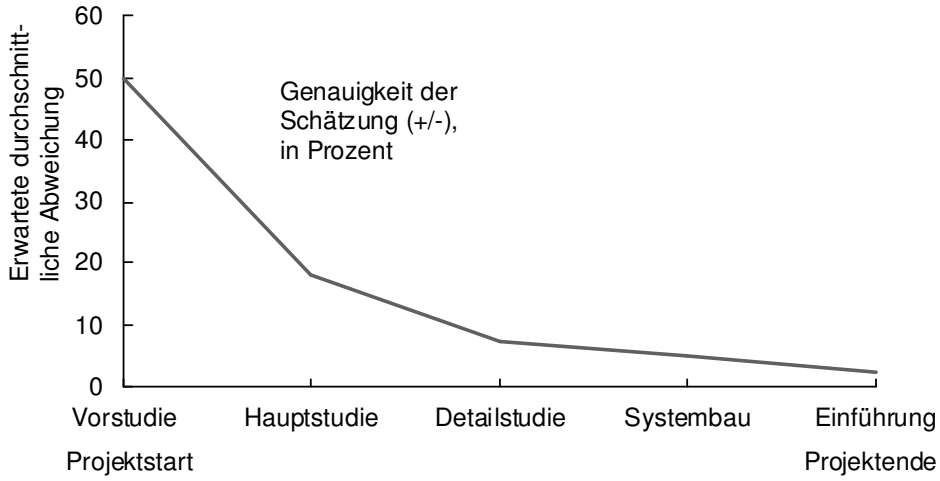

Abbildung 9 – Übersicht der zu erwartenden Schätzgenauigkeit zu verschiedenen Projektzeitpunkten [vgl. Wiec08, S. 206 und Brun01, S. 352]

Bereits in den 1960er Jahren hatte die Bedeutung von Aufwandsschätzungen das Top-Management erreicht, da zahlreiche Softwareprojekte die Zeit- und Budgetvorgaben nicht einhalten konnten. Seither wird intensiv an der Entwicklung geeigneter Schätzmethoden gearbeitet. In [Jone07, S. 23 ff.] findet sich eine umfassende Beschreibung der Entwicklung der Software Bewertungsmethoden. Tabelle 1 zeigt eine Zusammenfassung der relevanten Veröffentlichungen zu diesem Thema.

[51] Zu weiteren Infos zu diesem wichtigen +/-20% Korridor siehe auch Kapitel 5.2.

Erfinder	Bewertungsmethode oder -tool	Jahr
Dr. Randall Jensen	SEER	1974
Frank Freiman	PRICE-S	1977
Allan Albrecht	IBM function points	1978
Larry Putnam	SLIM	1979
Dr. Barry Boehm	COCOMO	1981
Tom DeMarco	DeMarco function points	1982
Charles Symons	Mark II function points	1983
Dr. Howard Rubin	ESTIMACS	1983
Allan Albrecht and Capers Jones	Backfiring (LOC to function points)	1984
Capers Jones	SPQR/20	1985
Allan Albrecht and Capers Jones	SPR feature points	1986

Tabelle 1 – Commercial Software Estimation von 1974 bis 1986 nach Capers Jones [Jone07, S. 30]

Im Folgenden werden die wichtigsten Ansätze kurz erläutert und hinsichtlich ihrer Eignung, insbesondere im Bezug auf die vorliegende Arbeit bewertet. Für eine vollständige Übersicht siehe [Lair06, S. 34 ff. und Jone07, S. 33 ff.]. Ebenso sei angemerkt, dass die hier vorgestellten Methoden lediglich die Basis für eine spätere ökonomische Analyse darstellen. Eine Übersicht über die vorhandenen Ansätze und Modelle zur ökonomischen Bewertung gibt Teil B Kapitel 3.

5.1. Lines of Code

Lines of Code (LoC) ist ein sehr einfaches, auf Durchschnitts- und Erfahrungswerten basierendes Verfahren und damit dem oben angesprochenen "Bauchgefühl" noch ziemlich ähnlich. Der Ansatz basiert auf der Annahme, dass es im Durchschnitt immer gleich lange dauert, um eine Zeile Code zu schreiben. Kennt man nun die ungefähre Anzahl an Codezeilen für ein Programm, lässt sich der erwartete Aufwand leicht ableiten. Dieser Wert gilt natürlich nur für einen bestimmten Softwaretyp und eine bestimmte Programmiersprache [Boeh00, S. 15]. So ermöglichen höhere Sprachen eine bestimmte Funktionalität mit weniger LoC zu realisieren, da bestimmte Basisdienste zur Verfügung stehen. Sicherheitskritische Anwendungen andererseits resultieren eher in einer geringeren LoC Zahl pro Zeiteinheit als Standardapplikationen, da hier höhere Sicherheitsanforderungen gestellt werden.

Eine tatsächliche LoC Zählung ist erst nach Fertigstellung der Software möglich. Eine grobe Abschätzung basierend auf Erfahrungswerten und Expertenschätzungen ist jedoch bereits zu einem frühen Projektzeitpunkt möglich. Dazu werden Projektbeschreibungen und Funktionsübersichten herangezogen. Die Vorteile von LoC Zählungen sind die weitgehende Automatisierbarkeit und die einfache Verständlichkeit, die kein intensives Training voraussetzen. Für eine Anleitung zum Zählen von LoC siehe [Park92, Lair06, S. 34ff. und Boeh00, S.75ff. und 394f.].

Andererseits ergeben sich auch zahlreiche Kritikpunkte, wie die Abhängigkeit der Ergebnisse (die Anzahl der Codezeilen) von der verwendeten Technologie (z.B. der Programmiersprache), obwohl für den Nutzer allein die realisierte Funktionalität von Bedeutung ist. Hinzukommt die Problematik der Abschätzung bei innovativen Produkten, da hier die notwendigen Erfahrungswerte fehlen. Auch Codegeneratoren stellen diesen Ansatz vor ein Problem. Schließlich vernachlässigt der Ansatz den Einfluss der Qualifikation und des Stils des Softwareentwicklers. Es ist zu erwarten, dass ein erfahrener Programmierer die identische Funktion mit deutlich weniger LoC realisieren kann. Die Größe der zugrundeliegenden Einheit (LoC) wird demnach von zahlreichen weiteren Faktoren neben der eigentlich entscheidenden Komplexität der zu erstellenden Software beeinflusst.

Als Fazit lässt sich festhalten, dass dieser Ansatz für eine frühzeitige zuverlässige Aufwandsanalyse bei Embedded Systems grundsätzlich nicht geeignet ist. Trotzdem wird dieser Ansatz weiterhin wegen seiner geringen Komplexität in vielen Kostenmodellen verwendet.

5.2. Function Points

Function Points wurden 1979 erstmals in dem Aufsatz von Allan Albrecht "New Way of looking at tools" erwähnt [Jone07, S. 26 f.]. Das Ziel war, den Umfang einer Software auf eine neue Art zu messen, die es präziser als der bis dahin bestehende Ansatz der LoC vermag, Aussagen hinsichtlich Produktivität und zu erwartender Kosten zur Realisierung abzugeben. Albrecht arbeitete damals für IBM, Ende der 1970er Jahre das größte Softwareunternehmen der Welt. IBM entwickelte Software in zahlreichen Bereichen, u.a. Betriebssysteme, Anwendungen, Werkzeuge, jeweils für verschiedene Programmiersprachen und Plattformen. IBM suchte daher eine Methode, die unabhängig von den technischen Bedingungen eine Aufwandsabschätzung ermöglichte. Daher änderte Albrecht die Perspektive von der

Betrachtung des tatsächlich geschriebenen Codes zu einer Betrachtung der Software aus Sicht des Nutzers. Function Points sind definiert als ein Maß für den Umfang der Functional User Requirements (FUR), also der funktionalen Anforderungen, die ein Nutzer an ein System hat. Aus dieser Betrachtung folgt, dass Function Points unabhängig sind von der verwendeten Technologie, der Entwicklungsmethode, der Programmiersprache oder der Hardware. Sie geben lediglich technologieunabhängig an, wie aufwändig die Kundenanforderungen in Software umzusetzen sind. Dies erfolgt allerdings mit einer standardisierten Methode. Denn aufbauend auf den Ideen von Albrecht wurde die IFPUG (International Function Point User Group) 1986 gegründet, die die Weiterentwicklung und Verbreitung des Ansatzes vorantreibt. Im Jahre 2003 wurde dieser Ansatz als erster zur Messung des Umfangs von Software von der International Organization for Standardization (ISO) als ISO 20926 [ISO03a] zertifiziert.

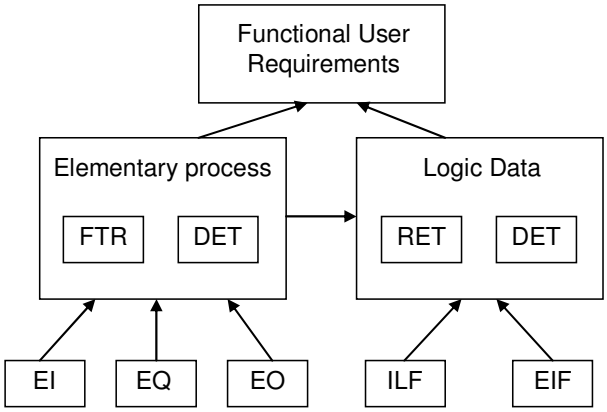

Abbildung 10 – Die Erfassung von Functional User Requirements für Function Points [Lava09, S. 2]

Der Prozess der Erhebung der Function Points wird als Function Point Analysis (FPA) bezeichnet. Das Standardvorgehen wird im Function Point Counting Practice Manual beschrieben, welches inzwischen in der Version 4.3 vorliegt [IFPU09]. Dabei wird zunächst unterschieden, ob es sich um eine Neuentwicklung oder eine Verbesserung bestehenden Codes handelt. Dabei werden lediglich die Function Points für die Veränderungen beachtet. Als dritte Alternative findet sich in der Literatur häufig noch die Installed Application. Diese bezieht sich auf die vollständige Zählung einer bereits vorhandenen Anwendung, die zunächst wie bei einer Neuentwicklung erfolgt, allerdings mit dem klaren Ziel, später Verbesserungen leichter ermitteln zu können als mit der reinen Zählung der Function Points für Veränderungen.

"On a conceptual level, function point analysis helps define two abstract levels of data – data at rest and data in motion" [Long01, S. 8]. Daten in Bewegung werden mittels Transaktionen erfasst, Daten in Ruhe sind Informationen, die von der Anwendung selbst oder einer anderen Anwendung verwaltet werden. Transaktionen werden verstanden als elementare Prozesse (Abbildung 10). Sie stellen die kleinste Aktivität, mit einem erkennbaren Nutzen für den Anwender dar. Beide Arten werden jeweils weiter untergliedert. Bei Daten unterscheidet man Internal Logical Files (ILF), also intern in der Software gespeicherte Datenstrukturen, und External Interface Files (EIF), also Datenstrukturen, die für die Anwendung zugänglich und notwendig sind, die aber außerhalb der betrachteten Systemgrenze liegen. Daher ist für die Nutzung dieser Quellen eine Schnittstelle notwendig. Transaktionen beziehen sich immer auf Interaktionen mit der externen Umgebung des Systems. Es werden Inputs (EI), also die Eingabe von Daten in das System, Outputs (EO), die Ausgabe von verarbeiteten Daten nach außen, und Inquiries (EQ), die Ausgabe von unverarbeiteten Daten nach außen, unterschieden. Diese insgesamt 5 Typen von Funktionen decken die gesamte Funktionsvielfalt von Software hinreichend ab. Allerdings weisen diese Funktionen unterschiedliche Komplexitätsgrade auf. IFPUG unterscheidet daher drei Kategorien: simpel, mittel und hoch komplex. Abbildung 10 zeigt die Faktoren, die über die Komplexitätseinstufung entscheiden. Bei den elementaren Prozessen oder Transaktionen sind das FTR und DET. FTR steht für File Type Referenced und bezeichnet die Anzahl der durch die Transaktion selbst genutzten ILFs und EIFs. DET bedeutet Data Element Type und ist ein Maß für die Menge an elementaren Informationen, die die Grenze der Anwendung überschreiten. DET sind dynamische und eben nicht statische Informationen, die für den Nutzer erkennbar sind. Außerdem müssen sie unique sein, oder bei einer wiederholten Verwendung wird lediglich das erste Erscheinen berücksichtigt. Für die Datentypen ergibt sich die Komplexität aus den RET und DET. DET sind identisch mit den für die Transaktionen beschriebenen DET. RET steht für Record Element Type und gibt die Typen von Information an, die im ILF oder EIF enthalten sind. Daher handelt es sich um vom Nutzer erkennbare Untergruppen von Datenelementen innerhalb der ILFs und EIFs. Eine komplette Anleitung zum Einsatz der FPA inkl. einiger Anwendungsbeispiele findet sich bei [Long01].

Die Komplexitätsgewichte zur Bewertung der jeweiligen Funktionen basieren auf der Erfahrung von mehreren tausend untersuchten Softwareprojekten. Die entsprechenden Function Point Werte je Funktionstyp und Komplexitätsklasse sind in Tabelle 2 dargestellt.

	Komplexitätsgewicht		
Funktionstyp	**Niedrig**	**Durchschnittlich**	**Hoch**
Internal Logical Files	7	10	15
External Interfaces Files	5	7	10
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6

Tabelle 2 – Function Point Werte je Funktionstyp und Komplexitätsgewicht nach IFPUG [Boeh00, S. 19]

Der Ablauf einer Function Point Zählung erfolg in Teilschritten (siehe Abbildung 11). Zuerst wird die Art der Zählung festgelegt, wobei nach Neuentwicklungen und Produkterweiterungen unterschieden wird. Die Zählung einer Produkterweiterung erfordert besondere Aufmerksamkeit, da lediglich die Funktionen in der Berechnung berücksichtigt werden dürfen, die tatsächlich verändert werden. Eine reine Übernahme vorhandener Funktionen erzeugt keinen Aufwand und wird daher nicht berücksichtigt, die Erweiterung bestehender Funktionen hingegen schon.

Abbildung 11 - Ablauf der Function Point Berechnung

Als zweites wird die Grenze der betrachteten Anwendung festgelegt. Eine Function Point Bewertung betrachtet Funktionen immer aus der Nutzersicht, demnach werden nur Funktionen bei der Berechnung berücksichtigt, die für den Nutzer einen Mehrwert schaffen. Die Nutzerperspektive kann aber variabel gestaltet werden. Wird bspw. die Entwicklung eines SW Entwicklungswerkzeugs untersucht, dann ist der Nutzer der spätere SW Entwickler. Gleichzeitig wird die Anwendung als abgeschlossenes System betrachtet. Die Systemgrenze legt fest, wie bestimmte Daten oder Transaktionen zu bewerten sind. So sind EIFs nur zu anderen Systemen möglich, die außerhalb der festgelegten Grenze liegen. ILFs müssen im Gegensatz dazu innerhalb der Anwendungsgrenzen gespeichert werden.

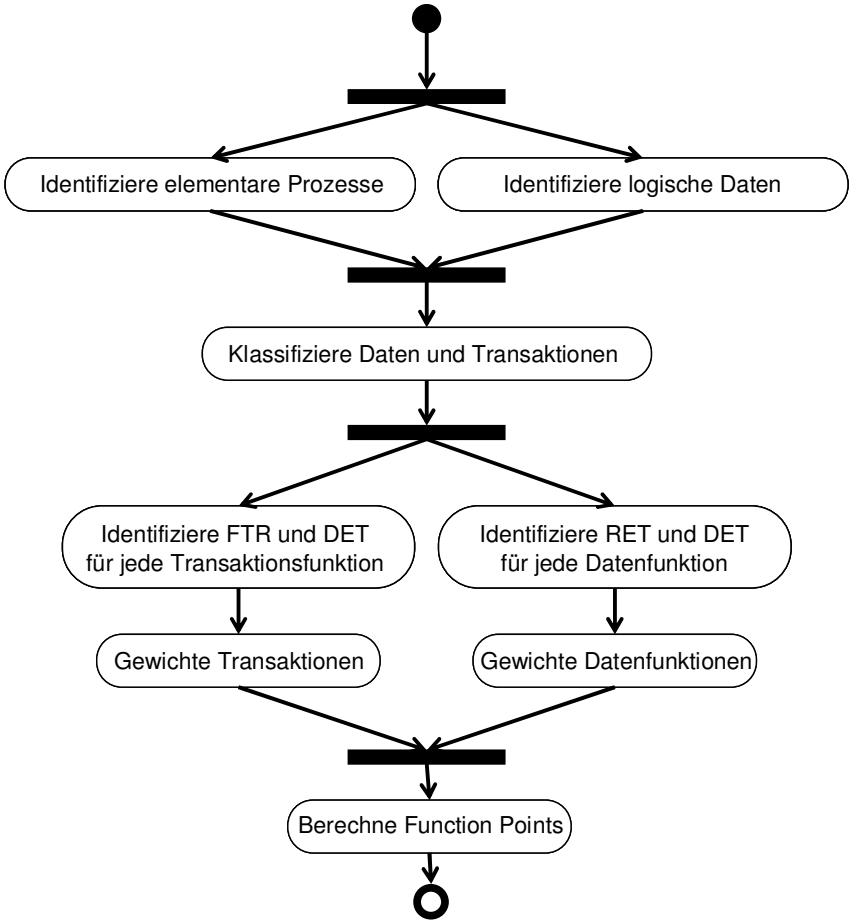

Abbildung 12 – Ablauf des Function Point Zählprozesses nach [Lava09, S. 3]

Anschließend beginnt der eigentliche Zählprozess (siehe Abbildung 12). Dazu werden alle Daten und Transaktionen für alle Funktionen entsprechend der Anwendungsgrenze identifiziert. Danach erfolgt eine Klassifizierung. Bspw. werden alle Eingabeprozesse für eine Funktion als External Inputs erfasst. Dabei ist entscheidend, dass die Eingabe von einem System außerhalb der festgelegten Grenzen erfolgt. Bis hierhin ist die Analyse mit vergleichsweise geringem Aufwand verbunden. Die Identifikation der Komplexitätstreiber im nächsten Schritt bedarf jedoch einer detaillierteren Betrachtungsweise. Für alle Elemente sind die DET und zusätzlich für jede Transaktion die entsprechenden FTR und für alle Daten die RET zu erfassen. Daraus ergibt sich die Komplexitätseinstufung anhand einer dreistufigen Skala mit den Kategorien gering, mittel und hoch. Anhand dieser Einstufung werden die Function Point entsprechend der Tabelle 2 vergeben. Wenn alle Funktionen analysiert worden sind, können die Function Points zum Gesamtwert addiert werden.

In einem letzten Schritt bietet die Methodik die Möglichkeit den ermittelten Wert mittels eines Value Adjustment Factors mit anderen Industrien und Anwendungsbereichen

vergleichbar zu machen. "The value adjustment factor (VAF) is based on 14 general system characteristics (GSC's) that rate the general functionality of the application being counted. Each characteristic has associated description to determine the degree of influence" [Long01, S. 65]. Dazu erfolgt zunächst die Bestimmung des VAF mittels der Beantwortung von 14 Fragen, die alle zu einer Einstufung zwischen 0 (nicht vorhanden, oder kein Einfluss) und 5 (durchgehend starker Einfluss) führen. Gleichung 1 zeigt die entsprechende Formel zur Berechnung des VAF. Zum Schluss wird der Function Point Wert mit diesem Faktor multipliziert, sodass sich die adjusted Function Points ergeben.

Gleichung 1

$$VAF = 0{,}65 + \left[\left(\sum_{i=1}^{14} C_i \right) / 100 \right]$$

Wobei: C_i = Einflussgrad jeder Charakteristik

i = läuft von 1 bis 14, repräsentiert die einzelnen Charakteristiken

Der Ansatz erlaubt es, mit einem relativ geringen und ökonomisch vertretbaren Zeitaufwand eine Schätzung über den zu erwartenden Aufwand zur Realisierung der geplanten Software bereits sehr früh im Entwicklungsprozess zu ermitteln. Die Nutzerperspektive ermöglicht es, aufbauend auf dem Pflichtenheft, eine erste Abschätzung zu geben. Ein deutlich genaueres Ergebnis liefert jedoch eine Betrachtung der tatsächlichen User Requirements, also die finale Funktionsübersicht der Software. Ebenso ermöglicht der Ansatz eine Messung der Produktivität nach Fertigstellung der Software. Ein Produktivitätswert erzielt seinen vollen Nutzen erst, wenn dieser mit anderen Abteilungen im Unternehmen, öffentlich zugänglichen Benchmarks oder über die Zeit mit Werten aus anderen internen Projekten verglichen wird. Dadurch wird es möglich, Verbesserungspotenzial direkt zu erkennen und die Wirkung von Maßnahmen zu kontrollieren.

In der Literatur finden sich zahlreiche kritische Äußerungen zum Function Point Ansatz [z.B. Lair06, S. 49 und Cosm09]. Meist wird angeführt, dass dieser Ansatz zu stark simplifiziert, z.B. aufgrund lediglich drei unterschiedlicher Komplexitätsgrade für Funktionen. Dies ist aber dadurch zu begründen, dass Function Points eine einfach zu erhebende Indikationsgröße darstellen, die keinen Anspruch auf absolute Genauigkeit erhebt. Der Ansatz sollte nur verwendet werden, wenn Ergebnisse mit einer Ungenauigkeit von maximal +/- 20% toleriert

werden können.[52] Bereits 1981 hat [Boeh81] einen Korridor mit dieser Genauigkeit für eine ausreichend genaue Schätzung von Entwicklungsaufwänden gefordert [vgl. auch Fent97, S. 105]. Abbildung 13 zeigt einen Vergleich der erzielbaren Messgenauigkeit im Verhältnis zur eingesetzten Zeit. Die Function Point Methodik nach IFPUG 4.2 bietet den besten Kompromiss aus Aufwand und Genauigkeit. Hinsichtlich der in dieser Arbeit angestrebten Zielsetzung ist die Fehlertoleranz dadurch erfüllt, dass eine positive Aussage hinsichtlich der SWPL Einführung nur dann getroffen wird, wenn die Sensitivität der Ergebnisse höher ist als die erwartete Unsicherheit (siehe dazu auch die detaillierten Ausführungen zur Sensitivitätsanalyse in Teil B Kapitel 6.3). Weiterhin ist zu beachten, dass die untersuchten Projekte eine gewisse Mindestgröße aufweisen müssen, da das Konzept der Function Points auf durchschnittlichen Beobachtungswerten basiert und bei zu isolierter Betrachtung größere Abweichungen auftreten können.

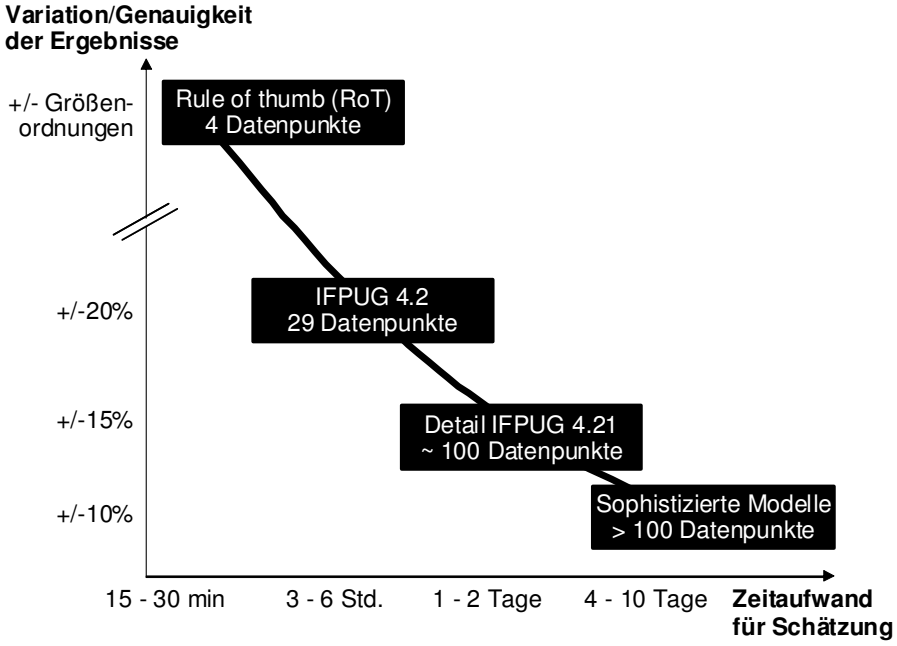

Abbildung 13 – Vergleich der Messungenauigkeiten und des notwendigen Zeitaufwands bei Function Point Methoden[53]

---

[52] Laut einer Studie in [Jone07, S. 261] erzielten zertifizierte IFPUG Function Point Zähler bei einem kontrollierten Test eine Abweichung von ungefähr 3%, während nicht-zertifizierte Zähler eine Abweichung von ca. 30% aufwiesen. Eine empirische Untersuchung – u.a. mit AT&T und GE Appliances – von [Conn90] zeigt, dass die durchschnittliche Abweichung bei Function Point Schätzunmgen nach IFPUG lediglich 10% betragen. Weiterhin zeigen empirische Daten, dass bei einer Genauigkeit der notwendigen Funktionen zum Zeitpunkt der Aufwandsbewertung von 5 bis 10%, „then the accuracy of the overall estimate can be quite good“ [Jone07, S. 9]. Sollten die Anforderungen sich jedoch im weiteren Verlauf stark ändern kann es zu Ungenauigkeiten von bis zu 50% kommen.

[53] Werte basieren auf [Jone07], [IFPU09] und Erfahrungen aus den im Rahmen der Erstellung der vorliegenden Arbeit durchgeführten Workshops und Function Point Zählungen.

Ebenso wird trotz zahlreicher Verbesserungen bemängelt, dass der Ansatz auf Ideen aus Anfang der 1980er Jahre beruht, sich in der Softwareentwicklung aber inzwischen zahlreiche neue Ideen, wie z.B. Objektorientierung, Entity-Relationship Diagramme oder UML, etabliert haben, die bei Function Points nicht umgesetzt sind. Dieser Kritik ist zuzustimmen, wenn es um eine Automatisierung des Zählprozesses und eine gleichzeitige Integration der Methode in die bestehenden Unternehmensabläufe geht. In diesem Fall ist eine Integration von UML und Entity-Relationship Diagrammen sinnvoll und wünschenswert. Jedoch liegen solche Dokumente zu dem in dieser Arbeit angestrebten frühen Zeitpunkt einer Analyse von geplanter Software in den meisten Fällen nicht vor.

Ein weiterer Kritikpunkt ist der Value Adjustment Factor (VAF). Dieser Wert setzt sich aus 14 Einzelbewertungen zusammen, die jeweils einen Aspekt der übergeordneten Komplexität des Systems erfassen sollen. Ziel des VAF ist es, trotz der Technologieunabhängigkeit des Ansatzes eine Vergleichbarkeit der Ergebnisse aus unterschiedlichen Industrien und beim Einsatz verschiedener Softwareentwicklungstechnologien zu ermöglichen. Auch wenn dieser Wert heute noch zum Vergleich mit Standardwerten aus der Literatur herangezogen wird, ist weitgehend akzeptiert, dass der VAF alleine für einen verläßlichen Vergleich von Werten verschiedener Anwendungen unzureichend ist. Dies ergibt sich nicht zuletzt aus dem limitierten Einfluss des Faktors. Der Wert kann maximal zwischen 0,65 und 1,35 schwanken. Das bedeutet, dass der unjustierte Wert lediglich um maximal 35% nach oben oder unten korrigiert werden kann. Konsequenterweise ist der VAF auch nicht in dem nach ISO 20926 standardisierten Ansatz enthalten. Im Rahmen der vorliegenden Arbeit wird der VAF ebenfalls nicht berücksichtigt.

Und schließlich vernachlässigt die Fokussierung auf die Nutzerperspektive interne Funktionen, z.B. Algorithmen[54]. Da diese Algorithmen insbesondere bei Embedded Systems eine wichtige Rolle spielen können, konnte die Function Point Methode im Rahmen dieser Dissertation nicht direkt übernommen werden. Die notwendigen Anpassungen werden in Kapitel 5.5 erläutert. Für eine genauere Begründung zur Verwendung der auf Function Points basierenden Methodik im Rahmen dieser Arbeit sei auf Teil B Kapitel 5.4 verwiesen.

---

[54] Broy defininiert einen Algorithmus als "ein Verfahren mit einer präzisen (d.h. in einer eindeutigen Sprache abgefassten) endlichen Beschreibung unter Verwendung effektiver (im Sinne von tatsächlich ausführbarer) Verarbeitungsschritte" [Broy98b, S.31].

Fazit: Die Function Point Analyse ist die mit Abstand am häufigsten angewendete Methode zur Aufwandsbestimmung bei SW-Projekten [vgl. Jone07, S. 245]. Dies ist mit einigen Vorteilen verbunden, wie z.B. einer umfangreichen Datenbasis, einem hohen Bekanntheitsgrad und einem großen Team, das an der Weiterentwicklung arbeitet.

5.3. Mark II Function Points

MarkII Function Points, auch bekannt als Mk II, wurden 1988 von Charles Symons mit dem Ziel entwickelt, die ursprüngliche Idee der Function Points kompatibel zu neuen Entwicklungsmethoden im Softwarebereich, wie Structured Analysis, Entity-Relationship-Diagramme und logical Transactions, zu machen. Bei der Anwendung der von Albrecht entwickelten Basismethode zeigte sich, dass eine weitgehend automatische Zählung der Function Points aufgrund der neuen Entwicklungsmethoden nicht möglich war. Symons neuer Ansatz sollte dies allerdings ermöglichen und dabei gleichzeitig noch einige weitere Kritikpunkte lösen. Bei Mk II werden logische Transaktionen betrachtet (Entitäten) und nicht wie beim ursprünglichen Ansatz logische Dateien (Logical Files). Logische Transaktionen werden definiert als "each logical transaction is triggered by a unique event of interest in the external world, or a request for information and, when wholly complete, leaves the application in a self consistent state in relation to the unique event." [UKSM98, S. 21]. Logische Transaktionen sind die "lowest level business processes supported by a software application" [UKSM98, S. 21].

Mk II geht davon aus, dass alle Anforderungen oder "User Functionality" als "Logical Transactions" ausgedrückt werden können, wobei jede dieser Transaktionen durch ein Ereignis aus der realen Welt oder eine Datenanfrage angestoßen wird, entsprechende Daten verarbeitet und das Ergebnis ausgibt. Dementsprechend besteht jede Transaktion aus einem Input, einer Verarbeitung und einem Output. Es werden also im Gegensatz zu Albrechts Methode lediglich 3 statt 5 unterschiedliche Komponenten betrachtet. Den umstrittenen VAF erweitert Symons um 6 weitere Faktoren, wie z.B. spezielle Sicherheitsfunktionen oder direkten Zugriff durch Dritte und bezeichnet ihn als Technical Complexity Adjustment (TCA). Wie auch bei Albrechts Function Points wird davon abgeraten, diese Anpassung zu benutzen. Zu beachten ist, dass Mk II ebenso wie Albrechts Methode keine qualitativen Anforderungen berücksichtigt. Die Auswirkungen besonderer Anforderungen hinsichtlich Antwortzeiten, Sicherheit oder Zuverlässigkeit sollten allein über den VAF, bzw. den TCA abgefangen werden. Eine Vernachlässigung dieser Faktoren erschwert die Vergleichbarkeit

der Ergebnisse zwischen verschiedenen Abteilungen innerhalb des Unternehmens und mit externen Benchmarks.

Die einzelnen Schritte zur Durchführung der Messung sind vergleichbar mit Albrechts Function Point Ansatz. Zunächst erfolgen die Festlegung des Zählzwecks, des Betrachtungspunktes und der Zählart (Neu- oder Weiterentwicklung). Daraus ergibt sich die Grenze der betrachteten Applikation. Anschließend werden alle logischen Transaktionen und Entitätstypen identifiziert. Es wird empfohlen vor der Anwendung von Mk II zunächst eine Entity-Relationship- oder Relational-Data-Analysis durchzuführen, um ein vollständiges Entity-Relationship Diagramm für die geplante Anwendung zu erhalten. Darauf aufbauend ist eine Mk II Zählung einfach durchzuführen, da für alle Transaktionen jeweils lediglich die Anzahl der Input und Output Datentypen und die referenzierten Entitäten zu erfassen sind.

Zur eigentlichen Messung der Komplexität der Software wird angenommen, dass der Umfang der Input und Output Komponenten jeweils proportional zu der Anzahl der damit verbundenen Data Element Typs ansteigt. Je mehr unterschiedliche Daten Elemente also bei einem In- oder Output verknüpft werden, desto komplexer wird das jeweilige Element und desto mehr Function Points werden ihm zugewiesen. Ebenso proportional verhält sich der Umfang der Verarbeitung zu der Anzahl der Entitätstypen, auf die während der Verarbeitung referenziert wird. Der Umfang der einzelnen Komponenten wird anschließend gewichtet und zu einer Gesamtgröße addiert. Diese Gewichtung unterscheidet nun im Gegensatz zu IFPUG keine unterschiedliche Komplexität, da die Funktionen bereits vorher in ihre kleinsten Einheiten (Datenelemente und verknüpfte Entitäten) zerlegt worden sind. Dies bedeutet, dass die Messung nicht direkt erfolgt, sondern abgeleitet über die jeweils verknüpften Entitäten und die übergebenen Datenelemente. Die Gewichtung lautet standardmäßig: 0,58 für einen Input Data Element Type, 1,66 für einen Entity Reference Type und 0,26 für einen Output Data Element Type.

Sowohl bei Albrechts Function Points als auch bei MK II handelt es sich um einen Ansatz der 1. Generation zur Software Messung. Obwohl Symons mit seinen Veränderungen einige Kritikpunkte aufgegriffen hat, bleibt die Fokussierung der Methodik auf Business Applications aus Sicht der vorliegenden Arbeit die größte Schwachstelle. Die vorgestellten Zählregeln lassen zwar eine Adaptierung zu, geben jedoch keinerlei Hilfestellung. Ebenso ist die verfügbare Datenbasis zum Vergleich der Ergebnisse zu gering. Daher wird Mk II in dieser Arbeit nicht verwendet.

5.4. COSMIC-FFP

In einem Versuch, eine einheitliche Methode zur Messung des Umfangs von Software zu schaffen, wurde 1998 das Common Software Measurement International Consortium (COSMIC) gegründet. Ziel war es, eine Methode zu begründen, die sowohl Business Software, Real-Time Software als auch hybride Mischungen korrekt analysieren kann. Ende 1999 wurde die erste Version veröffentlicht. Nach Aussage der Autoren handelt es sich dabei um die erste "2nd Generation functional sizing method". Dies wird mit der universellen Einsetzbarkeit der Methode begründet. Einzige Einschränkung ist der Bereich von Software, der der Manipulation von Daten dient. Gemeint sind komplexe wissenschaftliche Anwendungen, die hauptsächlich auf Algorithmen basieren, wie z.B. Wetter- oder Klimaberechnungen und Unfallsimulationen. Dies liegt in der Tatsache begründet, dass bis heute keine Methode zur Verfügung steht, die eine Messung von algorithmenbasierter Software ermöglicht.

Ebenso wie der Ansatz von IFPUG ist auch COSMIC-FFP in der Version 2.2 seit 2003 standardisiert: ISO/IEC 19761 [ISO03b]. Im September 2007 wurde die Version 3.0 veröffentlicht, womit gleichzeitig der Name auf COSMIC Methode geändert wurde.

Einer der größten Vorteile der COSMIC Methode wird in der Kompatibilität zu modernen Konzepten der Software Entwicklung gesehen (UML, Entity Relationship Analyse und Relationale Daten Analyse und sogar Objektorientierung). Dies unterstützt eine weitgehend automatische Erfassung und Zählung von Software Artefakten, sobald die grundlegenden Informationen, wie z.B. Entity-Relationship Diagramm oder UML, vollständig vorliegen. Zusätzlich erweitert der Ansatz die strenge Nutzerperspektive des IFPUG Konzeptes. COSMIC ermöglicht es, verschiedene Viewpoints einzunehmen und die Messung enstprechend anzupassen. Es ist jedoch zu beachten, dass Messungen mit unterschiedlichem Blickwinkel nicht miteinander verglichen oder addiert werden können. Den Nutzen einer solchen Unterscheidung kann man sich anhand eines Vergleiches mit einem Haus klar machen. Fragt man einen Passanten auf der Straße nach der Höhe eines Hauses wird er sicherlich die für ihn sichtbare Höhe verstehen. Für einen Architekten hingegen kann die tatsächliche Anzahl der Stockwerke, inklusive der unterirdischen Kellergeschosse jedoch von viel größerer Bedeutung sein. Der Entwickler-Blickpunkt kann z.B. helfen, mehrere Projekte mit unterschiedlichen Technologiebeschränkungen zu vergleichen. Man stelle sich vor eine bestimmte Funktion ist dreimal zu realisieren. Einerseits auf einer einzelnen Plattform, dann

auf einer mehrfach genutzten Plattform und schließlich in einer Umgebung, bei der zur Umsetzung auch Änderungen in Layern unterhalb der reinen Anwendungsschicht durchzuführen sind. Aus reiner Nutzersicht spielen diese technischen Beschränkungen keine Rolle und würden bei einer klassischen Zählung unberücksichtigt bleiben. Für den tatsächlichen Entwicklungsaufwand kann dies jedoch bedeutende Auswirkungen haben.

Die Zählung selbst basiert ähnlich wie bei IFPUG auf Functional User Requirements (FUR), die in der Mapping Phase auf das Generic Software Model angepasst werden. FUR enthalten keine Quality Requirements oder technischen Requirements. COSMIC definiert FUR als "a sub-set of the User Requirements. Requirements that describe what the software shall do, in terms of tasks and services." [Cosm09a, S. 18] Wichtig ist die klare Fokussierung auf den funktionalen Aspekt der Anforderung. Dies bedeutet, dass Nutzeranforderungen nicht als FUR betrachtet werden, die sich bspw. allein auf Qualitätsaspekte (Nutzbarkeit, Verlässlichkeit, usw.) beziehen. Anschließend werden die FUR differenziert in functional process types. Dabei handelt es sich um "an elementary component of a set of Functional User Requirements comprising a unique cohesive and independently executable set of data movement types." [Cosm09b, S. 15] Genauso wie bei IFPUG gibt es eine Anwendungsgrenze, die vorgibt, welche Teile zu berücksichtigen sind und welche vernachlässigt werden. Der Ausgangspunkt für die Bestimmung der Grenze ist der Nutzer der Software. Alle Funktionen, die er direkt nutzen kann, oder Funktionen, die für ihn nicht sichtbar sind, von denen er aber dennoch profitiert, befinden sich innerhalb der Untersuchungsgrenze. Der Zusammenhang wird in Abbildung 14 verdeutlicht.

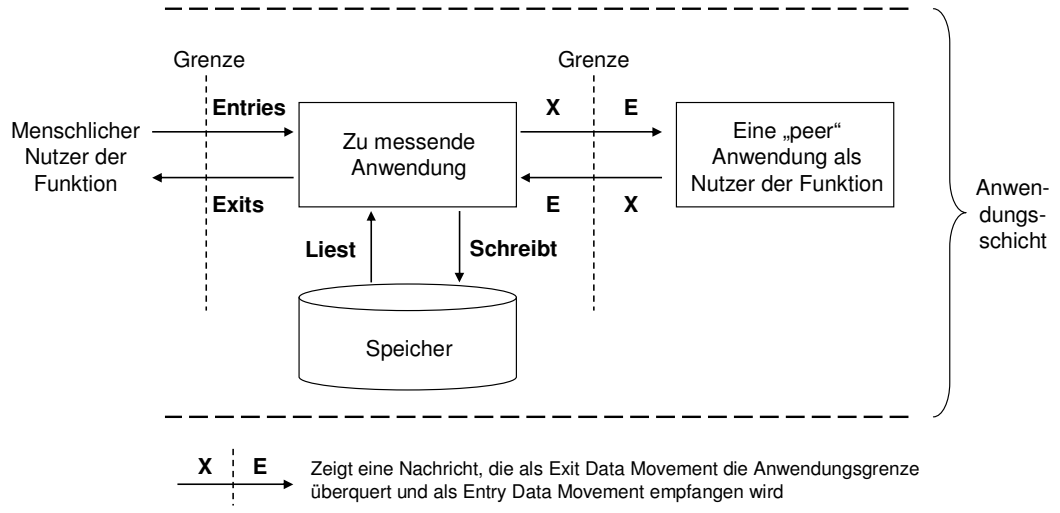

Abbildung 14 – Darstellung der Anwendungsgrenzen [vgl. Cosm09a, S. 33]

Die Zählung erfolgt in drei Teilschritten:

- **Strategy:** In dieser ersten Phase wird der Blickwinkel für die Messung festgelegt (z.B. Entwickler oder Nutzer). Ebenso wird die Boundary, also die Begrenzung, der Betrachtung bestimmt.
- **Mapping:** Die einzelnen zu betrachtenden Artefakte werden im Cosmic generic Software model zugewiesen. Dies stellt den schwierigsten Teil der Messung dar, da die unternehmensindividuellen Ausgangsdaten auf ein starres Modell angepasst werden müssen. Inhaltlich werden in diesem Schritt die identifizierten funktionalen Prozesse in ihre kleinsten Elemente zerlegt, entweder in Data Movements oder Data Manipulation. Für die weitere Messung werden lediglich die Data Movements näher betrachtet. Es wird angenommen, dass sich der Aufwand für die Data Manipulation proportional dazu verhält. Dies ist auch der Grund, warum algorithmenbasierte Software nicht mit COSMIC analysiert werden kann.
- **Measurement:** Die eigentliche Messung ist dann ein trivialer Schritt. Es erfolgt die Identifikation und Zählung aller Data Movements, wobei 4 Kategorien unterschieden werden (entry, exit, read and write). Read greift auf persistente Daten aus dem Speicher zu, Write schreibt die Daten in den Speicher und macht sie persistent über die Dauer des Prozesses hinaus. Entry und exit kommunizieren mit dem Nutzer, einer anderen SW außerhalb der Application Boundary oder einem anderen Gerät (z.B. einem Sensor). Jede Datenbewegung wird mit einem COSMIC Function Point (CFP) bewertet. Ein funktionaler Prozess besteht nun mindestens aus 2 CFPs ohne eine Begrenzung nach oben. Aufgrund der Annahme, dass sich der Aufwand für die Datenmanipulation proportional zu den Datenbewegungen verhält, berücksichtigt jede Datenbewegung einen gewissen Aufwand für die Manipulation. Schließlich werden die CFPs addiert. Abbildung 15 veranschaulicht den Ablauf der Zählung.

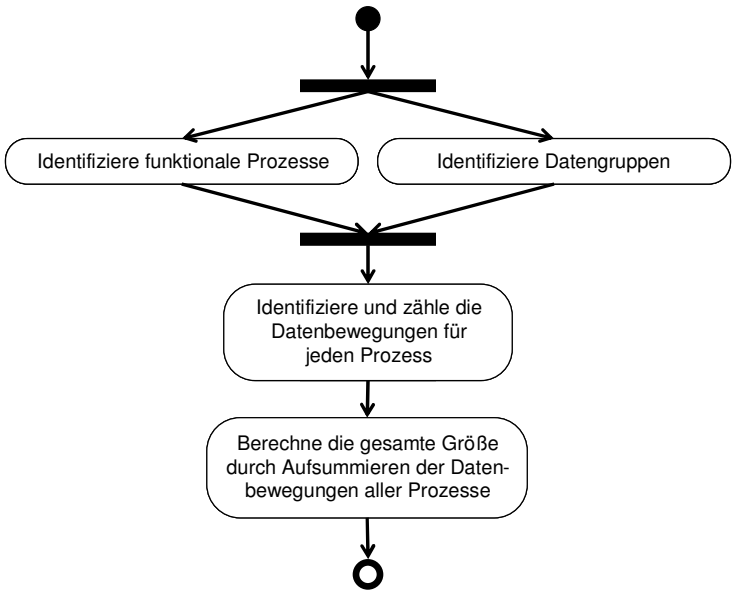

Abbildung 15 – Der COSMIC Zählprozess nach [Lava09, S. 4]

Interessanterweise weißt COSMIC explizit darauf hin, dass die Einführung eines SW Measurements als ein langfristiges Investment verstanden werden sollte. Nach ihrer Erfahrung sind ca. 10 Projekte notwendig, um eine ausreichende Datenbasis für sinnvolle und begründete Aussagen und Maßnahmen zu erstellen. Für diese Datenbank können natürlich auch bereits beendete Projekte herangezogen werden, allerdings zeigt die Erfahrung, dass die Datenlage in den Unternehmen dazu oft nicht ausreicht.

COSMIC nimmt für sich in Anspruch die IFPUG Methode an entscheidenden Stellen verbessert zu haben:

- Die lediglich 3 Komplexitätsklassen bei IFPUG ermöglichen keine ausreichende Berücksichtigung der realen Komplexitätsunterschiede bei einzelnen Elementen. So kann der maximale Komplexitätsunterschied einer Funktionsart lediglich bei ca. Faktor 2 liegen. COSMIC sieht lediglich eine Untergrenze von 2 CFPs je Prozess vor, jedoch keine Obergrenze. Im Bereich Avionic finden sich bspw. funktionale Prozesse mit bis zu 120 CFPs (z.B. Reaktion auf einen Raketenangriff im Militärjet)
- IFPUG berücksichtigt bei Updates das Ausmaß der Veränderung zu grob. Unabhängig vom Grad der Änderung an einer Komponente (z.B. einem Logical File) werden alle FPs erneut gezählt. COSMIC hingegen berücksichtigt lediglich die hinzugefügten Sub-Prozesse.

- Die IFPUG-Zählung basiert auf den vorhandenen Daten. Ein Logical File wird einmal gezählt und je nach Komplexität gewichtet. COSMIC zählt jedoch jeden Datenzugriff. Eine mehrfach benutzte Datei kann also auch mehrfach gezählt werden. Dieses Vorgehen orientiert sich nicht allein am Vorhandensein der Datei, sondern an der Anzahl der Aufrufe/Nutzung der Datei und stellt damit eine stärkere Prozessorientierung dar.

Diese Aufstellung beschränkt sich auf die wichtigsten Unterscheidungen. Für weitere Details zu Abweichungen zwischen den beiden Methoden sei auf [Cosm09c] verwiesen. Erste statistische Untersuchungen zeigten eine sehr gute Übereinstimmung der Messergebnisse zwischen IPFUG und COSMIC. Mit der Ausnahme von sehr kleinen Projekten unter 100 CFP ergibt sich laut [Fetk99 und Voge03] eine gute Übereinstimmung der Ergebnisse. Konkret ergab sich eine Abweichung von lediglich +/- 10 bis 20% bei einem Korrelationskoeffizienten von 99%. Allerdings sind die Messdaten nicht umfangreich genug, um eine allgemeine Aussage treffen zu können [Cosm09c]. Zusätzlich sind die erweiterten Möglichkeiten von COSMIC, wie geänderter Blickwinkel und Unterstützung für Real-Time Software, nicht berücksichtigt worden.

Hinsichtlich der Messdauer für beide Verfahren zeigt sich bei guter Dokumentation kein signifikanter Unterschied. Sollen jedoch erste Aussagen bereits zu Beginn des Entwicklungsprozesses getroffen werden, dann kann ein erfahrener Zähler trotz begrenzter Informationen und Dokumentation aufgrund der Komplexitäts-Kategorisierung bei IFPUG eine schnelle erste Schätzung ermitteln. Bei COSMIC ist in solchen Fällen eine Zählung zwar ebenfalls möglich, erfordert allerdings meist ein intensives Gespräch mit Experten zur Klärung der genauen Struktur.

5.5. Use Case Points und weitere Verfahren

Mit der Einführung von UML [Booc98] und deren zunehmenden Verbreitung ab ungefähr 2000 [Jone07, S. 267] bekamen Use Cases eine größere Bedeutung. Folgerichtig entwickelte sich auch ein Ansatz, der es ermöglichte auf Basis der Use Cases eine Function Point Abschätzung durchzuführen. Grundsätzlich basiert diese Methode auf dem IFPUG Ansatz, allerdings erfolgte hier keine Entwicklung von Regeln zur Zählung durch IFPUG selbst, wie z.B. für UML [Jone07, S. 181]. Für Use Cases wurden einige zusätzlichen Parameter eingeführt. So werden bspw. die Gesamtzahl der Akteure (damit sind die verschiedenen Arten

von Nutzern gemeint, also sowohl tatsächliche Personen, als auch andere SW) und die Anzahl der Verbindungen zu anderen Systemen zur Schätzung des Gesamtprojekts herangezogen. Dies ist auch der Grund dafür, warum eine Use Case Point Schätzung im Vergleich zu Function Points nach IFPUG im Allgemeinen ein höheres Ergebnis liefert. Hier kann man nach [Jone07, S. 267] von einem Zuwachs von ca. 25% ausgehen. Interessanterweise wächst diese Methode aktuell deutlich, obwohl sie weder standardisiert ist noch eine Zusammenarbeit mit IFPUG oder NESMA stattfindet. Zusätzlich ist diese Methode auf objektorientierte SW-Projekte beschränkt, was insbesondere die Vergleichbarkeit der Ergebnisse mit alten Daten erschwert. Der größte Vorteil dieses Ansatzes dürfte in der einfachen Anwendbarkeit liegen, vorausgesetzt die Use Cases liegen bereits vollständig vor. Für UML Spezifikationen wäre es technisch möglich eine weitgehend automatische Zählung durchführen zu lassen. Allerdings steht die Entwicklung entsprechender Werkzeuge noch am Anfang. Ziel könnte es aber sein, auch für Use Cases eine weitgehend automatisierte Zählung durchführen zu können.

Eine weitere Vereinfachung in der Anwendbarkeit stellt die Story Points Methode da. Diese wurde im Rahmen der Agilen SW Entwicklungsansätze und dem Extreme Programming (XP) entworfen. Nutzeranforderungen werden hier als User Stories erfasst. Diese entsprechen weitgehend dem Konzept der Use Cases, sind allerdings weniger formal. Der Vorteil dieses Ansatzes liegt in der einfachen Anwendbarkeit. Allein auf Basis einfacher und kurzer Beschreibungen von Funktionen aus Nutzersicht wird eine grobe Abschätzung ermöglicht. Allerdings bestehen für diese Methode aktuell noch wenige Erfahrungswerte und es fehlt eine Standardisierung. Wünschenswert wäre hier eine Zusammenarbeit mit IFPUG oder NESMA, um diese Methode stärker zu unterstützen.

Aufgrund des Alters der Function Point Methodik haben sich in der Zwischenzeit zahlreiche weitere Abwandlungen ergeben, wie z.B. 3D Function Points oder Full Function Points. Für weitere Details wird auf [Jone07, S. 260 ff.] verwiesen.

Insgesamt ist festzuhalten, dass die Function Point Methode aufgrund der regelmäßigen Anpassungen und Erweiterungen immer noch die mit weitem Abstand am häufigsten genutzte Methode zur Abschätzung des SW Entwicklungsaufwands darstellt. Alle modernen SW Entwicklungsansätze können mit der Methodik abgedeckt werden. Die vorhandenen Abwandlungen und neuen Ansätze zielen häufig auf spezifische Bereiche ab, für die sie optimiert worden sind. Grundsätzlich basieren aber alle Methoden auf den Grundüberlegungen der Function Points. In Kapitel 5.7 wird im Rahmen der vorliegenden

Arbeit ebenfalls eine leichte Anpassung der Standardmethodik vorgeschlagen, um Function Points besser für Embedded Systems nutzen zu können. Zunächst erfolgt aber eine Betrachtung der heutigen Verwendung von Schätzverfahren in der Automobilindustrie.

5.6. Übersicht der heutigen Verwendung von SW Aufwandsschätzverfahren in der automobilen Praxis

Im Rahmen der McKinsey Studie zur Produktivität bei Software für Embedded Systems im Automobil wurden mit mehr als 20 OEMs und Zulieferern Workshops durchgeführt, bei denen u.a. eine angepasste Function Point Methodik zur Aufwandsabschätzung verwendet wurde [McKi06]. Die Workshops wurden Ende 2005 und Anfang 2006 durchgeführt. Zu diesem Zeitpunkt waren Verfahren zur Aufwandsschätzung bei SW Projekten in der Automobilindustrie - zumindest bei den an der Studie teilnehmenden Unternehmen - weitgehend unbekannt. Bei keinem Teilnehmer waren Verfahren zur Aufwandsschätzung (ausgenommen von LoC) im Einsatz. Man kann festhalten, dass immer noch eine starke HW Orientierung sowohl bei den OEMs als auch den Zulieferern vorzufinden war. Ein ganz anderes Bild ergab sich bspw. bei den Herstellern von Mobilfunkgeräten. Hier war bei allen teilnehmenden Unternehmen ein detailliertes Verständnis der Aufwandsschätzmethoden vorhanden, die bei der Mehrzahl der Unternehmen aktiv eingesetzt werden. Der Grund dafür dürfte in der zu diesem Zeitpunkt bereits höheren Bedeutung der Software für Mobiltelefone gelegen haben. Schon 2005 war zu erkennen, dass Software Hardware als den entscheidenen differenzierenden Faktor ablösen würde. Gleichzeitig sahen sich viele Unternehmen von den typischen Problemen bei SW Projekten betroffen: zu lange Entwicklungszeit, Überschreitung des Budgets und zu hohe Fehlerzahl. Daher war der Einsatz der Schätzmethoden eine logische Konsequenz. Nach einer empirischen Analyse von Forrester aus dem Jahre 2008 bei SW Entwicklungsunternehmen und Unternehmen mit eigener SW Entwicklung [siehe Geru08, S. 8] nutzten mindestens 39% Function Points oder planten eine Einführung innerhalb des nächsten Jahres. Interessanterweise stimmt dieser Wert ziemlich genau mit den Angaben zur Erfassung der SW Wiederverwendungsrate in Unternehmen als KPI überein.

Inzwischen ist auch bei den Automobilherstellern und Zulieferern eine Verhaltensänderung beobachtbar. Auf Basis eigener Erfahrungen und einer Literaturrecherche finden sich im Folgenden einige aktuelle Beispiele zum Einsatz der Function Point Methodik und weiterer Schätzverfahren im Automobilen Umfeld:

- **BMW** setzt die FP-Methodik zusammen mit externen Partnern in mehreren Teilprojekten, teilweise in Kombination mit anderen Verfahren, erfolgreich ein. Jedoch erfolgt der Einsatz nur im Rahmen von Einzelprojekten. Eine institutionalisierte Verwendung über verschiedene Unternehmensbereiche hinweg erfolgte nicht. Konkret wurden Function Points im Rahmen der Anwendung von CoBRA, einer Kostenschätzmethode des Frauenhofer Instituts [Frau10] verwendet. Die BMW CarIT hingegen setzte bereits im Jahre 2004 Function Points und COCOMO kombiniert ein. Um die Frage zu beantworten wie hoch der Mehraufwand ist, sollten Daten für Memoryfunktionen (Spiegel, Sitz, Lenkrad, usw.) nicht mehr dezentral in den jeweiligen Systemen sondern zentral in einem Steuergerät gespeichert werden [Kamm04]. In beiden Fällen ergaben sich aussagekräftige Ergebnisse, allerdings mit der Einschränkung, dass diese nur für einen bestimmten Projekttyp und einen Unternehmensbereich verwendbar waren. Ein Vergleich über die Projektgrenzen hinweg war aufgrund der starken individuellen Rahmenbedingungen nicht möglich. Dieser Punkt entspricht den bereits weiter oben genannten bekannten Einschränkungen des Ansatzes, wird aber auch im weiteren Verlauf der Arbeit tiefer ausgearbeitet.

- **Ford und General Motors** nutzen laut Information der IFPUG Function Points ebenfalls aktiv. Mitte 2009 suchte Ford z.B. intensiv nach Function Point Experten. U.a. nutzte Ford Function Points erfolgreich zur Analyse der Software für Einspritzsysteme [Jone08, S. 124].

- **Renault** setzt seit 2008 COSMIC als Schätzmethode für die SW Entwicklung ein. Zuvor experimentierte das Unternehmen ohne Erfolg mit COCOMO. Zusätzlich wurde für den Bereich Business Software bereits seit einigen Jahren Functions Points mit Erfolg eingesetzt. Schließlich entschied sich Renault nach einem gemeinsamen Test von Function Points und COSMIC an der Motorsteuerungskomponente aufgrund der besseren Übereinstimmung mit den realen Werten für COSMIC. Renault verwendet die Schätzungen sowohl intern als auch als Basis für Preisverhandlungen mit Zulieferern. Insgesamt bewährte sich die Methodik laut Aussage von Renault und soll in weiteren Bereichen zum Einsatz kommen [Ster10].

- **Volkswagen und Audi** setzen keine Methoden zur Aufwandsschätzung ein, sondern verwenden ausschließlich erfahrungsbasierte Ansätze. Beide Unternehmen lehnen

jegliche Art der methodischen Aufwandsschätzung zur SW Erstellung ab, da sie in ihrer hauptsächlichen Rolle als Integrator die Verantwortung für den Erstellungsaufwand bei den Zulieferern sehen. VW und Audi erstellen detaillierte Anforderungen bei einer Ausschreibung und erzielen aus ihrer Sicht über die späteren harten Preisverhandlungen im Wettbewerb einen optimalen Preis. Allerdings ist auch hier in letzter Zeit ein Trend zu erkennen, dass sich die Angebotspreise der Zulieferer nach oben entwickeln. Und genau hier wäre eine eigene Aufwandsabschätzung durch den OEM vorteilhaft, um die Höhe der abgegebenen Angebote zumindest mit einer gewissen Ungenauigkeit einschätzen zu können.

Abschließend ist festzuhalten, dass sich mit der zunehmenden Bedeutung der SW im Automobil auch eine Veränderung der Einstellung der OEMs und Zulieferer zu beobachten ist. Es gibt erste deutliche Anzeichen, dass die in der Business SW Industrie erfolgreich eingesetzten Methoden zur Aufwandsabschätzung zunehmend auch in der Automobilindustrie Anwendung finden. Die geführte Diskussion über geeignete Methoden zur Aufwandsabschätzung ist vielfach akademisch geprägt. In vielen Fällen werden existierende Methoden, wie Function Points und COSMIC, zunächst aufgrund der Simplifizierung kritisiert. Vergleicht man aber die am Ende getroffene Entscheidung für ein Verfahren, zeigt sich, dass vielfach Methoden gewählt werden, die im Vergleich zu Function Points noch stärker vereinfachen, wie z. B. Use Case Points oder Story Points. Grund dafür ist meistens der geringere Erhebungsaufwand und die weitgehende Automatisierung der Zählung.

Nach aktuellem Kenntnisstand liegen keine allgemeinen Untersuchungen zur Verlässlichkeit der aktuellen Schätzmethoden vor. Dennoch konnte die Verwendung mit Hilfe von exemplarischen Beispielen unterstützt und begründet werden. Alle Beispiele, bei denen tatsächlich eine längerfristige Anwendung einer Schätzmethode erfolgte, zeigten einen erfolgreichen Einsatz.

5.7. Anpassung der Function Point Methode auf Embedded Systems

Die Function Point Methodik wurde, wie oben beschrieben, für eine Analyse von Business Software entwickelt. Obwohl IFPUG keine Einschränkung der Anwendbarkeit vorsieht, ist die direkte Nutzung der Methodik für Embedded Systems problematisch. Die Besonderheit bei Embedded Systems liegt in den vielfach einfach strukturierten Daten (Sensorsignale, Fahrereingaben, etc.) begründet, während die damit verbundenen Berechnungen oftmals eine

hohe Komplexität aufweisen. In der nach IFPUG definierten klassischen Function Point Methodik werden Daten und Algorithmen jedoch einheitlich als ein Element, den Internal Logical Files (ILFs), zusammengefasst. Die von McKinsey & Company entwickelte Feature Point Methode greift diesen Zusammenhang auf und trennt diese ILFs in ihre Daten- (Data File) und Berechnungskomponente (Algorithms).[55] Insgesamt ergeben sich damit 6 Typen von Funktionen. Die Gewichtung der einzelnen Komplexitätsgrade erfolgte direkt auf Basis der fundierten IFPUG Daten, indem die Gewichte eines ILFs im Verhältnis 2:1 auf die beiden neuen Kategorien Data File und Algorithm aufgeteilt worden sind. Bei Vorliegen eines Data Files und eines Algorithmus ergibt sich folglich kein Unterschied zwischen der ursprünglichen IFPUG Methodik und dem Feature Point Ansatz. Bei Embedded Systems ist dies allerdings nur selten der Fall, da es häufig Unterschiede zwischen der Komplexität der beiden Komponenten gibt. Ebenso ist das Verhältnis zwischen Daten und Algorithmen meist deutlich hinsichtlich mehr und komplexeren Algorithmen verschoben. In Tabelle 3 sind die einzelnen Komplexitätsgewichte aufgelistet.

Transaktions-/ Funktionstyp	Komplexitätsgewicht		
	Niedrig	Durchschnittlich	Hoch
Data File (DF)	5	7	10
Algorithms (AL)	2	3	5
External Interface Files (EIF)	5	7	10
External Inputs (EI)	3	4	6
External Outputs (EO)	4	5	7
External Queries (EQ)	3	4	6

Tabelle 3 – Feature Point Werte je Funktionstyp und Komplexitätsgewicht für Feature Point Berechnung

Diese sechs Typen decken alle notwendigen Bereiche für Embedded Systems im automobilen Bereich ab. Dies wurde bei der praktischen Anwendung der Methodik bei mehreren OEMs und Zulieferern im Rahmen der bereits genannten McKinsey Studie [McKi06] gezeigt. In der Literatur besteht keine Einigkeit hinsichtlich der Anwendbarkeit der Function Point Methodik bei Embedded Systems mit Real-Time Anforderungen. Während einige Quellen eine Eignung

[55] Bereits 1986 wurde eine Feature Point Methodik vom Begründer der Function Points, Allan Albrecht, vorgestellt. Auch dort wurde ein neuer Parameter für Algorithmen eingeführt, um die Methodik besser für Telekommunikations- und Real-Time-Anwendungen nutzbar zu machen [Jone07, S. 263].

bezweifeln [z.B. Cosm09c, siehe aber auch Kapitel 5.4] wird insbesondere von IFPUG und Capers Jones eine allgemeine Anwendbarkeit angenommen [vgl. IFPU09 und Jone07]. Auf Basis der persönlichen Anwendungserfahrung wird der letztgenannten Auffassung im Rahmen dieser Arbeit gefolgt. Schließlich wurde die Function Point Methodik bereits bei mehreren OEMs und Zulieferern erfolgreich bei automobilen Real-Time Embedded Systems eingesetzt, u.a. bei Motorsteuerungs-SW, Steuerungseinheit für Automatikgetriebe, Integration der Airbagsysteme in die Fahrzeugarchitektur und Hydraulikverstärker für Hybridfahrzeuge [vgl. McKi06 und Kapitel 5.6]. Hinsichtlich der Vergleichbarkeit der ermittelten Werte ist natürlich zu beachten, dass die Produktivität bei Real-Time Systemen unter ansonsten gleichen Rahmenbedingungen (ceteris paribus, c.p.) niedriger ausfallen wird als in anderen SW Projekten. Dieser Effekt soll zwar durch den VAF ausgeglichen werden, der in einer leicht angepassten Form für Embedded Systems und die Entwicklungsumgebung bei Automobilen ebenfalls zur Verfügung steht. Allerdings wird empfohlen, diesen lediglich zur Vergleichbarkeit mit externen Daten zu verwenden. Außerdem ist zu beachten, dass der VAF allein die Komplexität aller möglichen Unterschiede bei der SW Erstellung nicht abdecken kann. Für Vergleiche im Unternehmen und über die Zeit wird ausschließlich auf die unjustierten Werte zurückgegriffen. Auch die weiteren Anregungen und Verbesserungsvorschläge der functional sizing methods wurden weitgehend berücksichtigt. Bspw. wurde der starre Blickwinkel aus der Nutzerperspektive modifiziert, sodass vergleichbar zum COSMIC Ansatz auch Messungen aus Entwicklersicht möglich sind. Die dort genannten Vorgaben wurden übernommen. Bei Updates wurde ein prozentualer Veränderungsfaktor eingeführt, der verhindert, dass bei minimalen Veränderungen ein gesamter Typ erneut gezählt werden muss. Dieser Wert wird nur entsprechend des Veränderungsfaktors anteilig berücksichtigt. Dem bei COSMIC vorgebrachten Einwand einer unzureichenden Komplexitätsberücksichtigung wird basierend auf Anwendungserfahrungen des Autors entgegengebracht, dass eine dreiteilige Skala im Allgemeinen ausreicht. Voraussetzung ist eine ausreichend genaue Aufsplittung der betrachteten Funktionen oder Features. In manchen Fällen war es für die Entwickler sogar bei nur drei Komplexitätsklassen schwierig eine Einstufung vorzunehmen.

Das Vorgehen wurde empirisch bei der genannten Untersuchung der Firma McKinsey & Company für den Einsatz bei Embedded Systems im Automobilbereich getestet [McKi06]. Dabei erwies sich insbesondere die einfache und technologieunabhängige Anwendung der Methodik als vorteilhaft. Weiterhin erlaubt die Methodik einen sehr frühen Einsatz im

Entwicklungsprozess, da die finalen Entity-Relationship Diagramme zu diesem Zeitpunkt nicht notwendig sind. Stattdessen reicht eine allgemeine Aufstellung der geplanten Funktionen und ein entsprechendes Wissen über die geplante technische Umsetzung.

Eine Einschränkung des Ansatzes ist jedoch zu berücksichtigen: Aufgrund der verwendeten Durchschnittswerte des IFPUG Ansatzes eignet sich die Methodik nur für Projekte mit einer ausreichenden Größe. Eine Zählung von Projekten mit weniger als 15 Feature Points führt zu teilweise zufälligen Ergebnissen [Jone05, S. 269]. Aber selbst bei Projekten mit weniger als 100 Feature Points kann es zu Ungenauigkeiten aufgrund von außergewöhnlichen Effekten kommen. Dies ist u.a. bei der Bewertung von Einzelpersonen zu berücksichtigen. Meist ist der Umfang auf Einzelpersonenbasis nicht ausreichend, um eine verlässliche Aussage treffen zu können. Eine empirische Untersuchung von McKinsey hat gezeigt, dass die Abweichungen mit einiger Erfahrung im Zählen von Feature Points je Funktion im Bereich von +/-20% liegen. Die Abweichungen liegen damit etwas höher als bei der empirischen Untersuchung von Connolley [siehe Kapitel 5.2 und Conn90]. Die entsprechenden Ergebnisse sind in Abbildung 16 dargestellt.

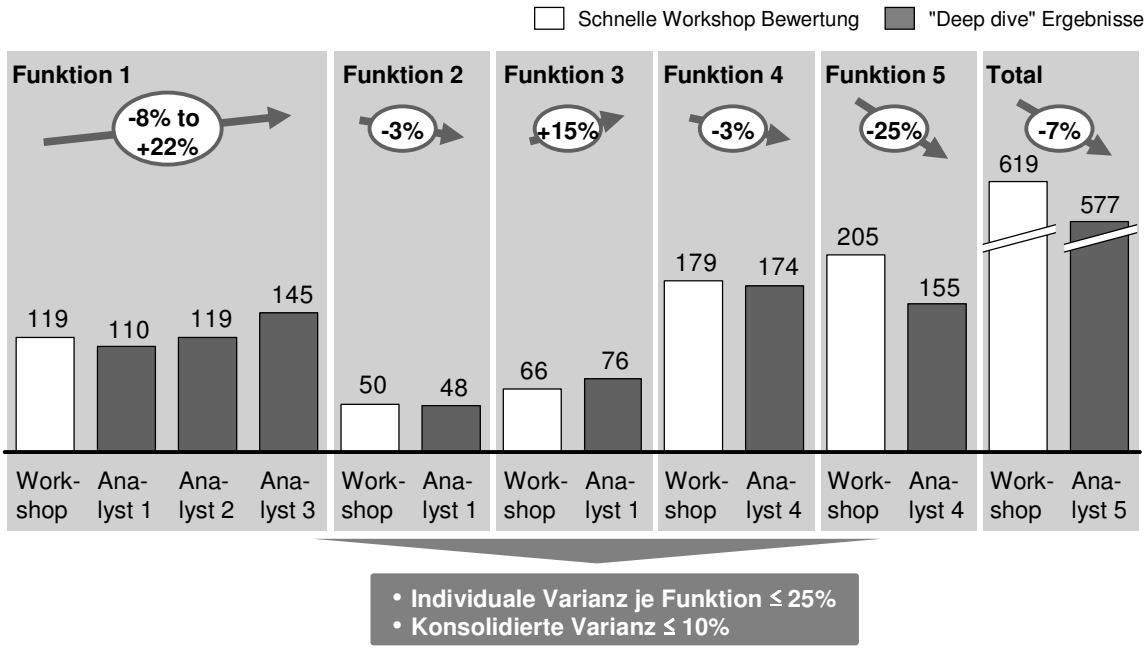

Abbildung 16 – Empirische Daten zur Messungenauigkeit der FP Methodik[56]

[56] Die Werte basieren auf einer unveröffentlichten Analyse von McKinsey und Company bei einem europäischen Mobilfunkhersteller.

Mit zunehmendem Umfang des analysierten Projektes werden einzelne Abweichungen durch individuelle Besonderheiten, wie z.B. einen besonders erfahrenen Entwickler, immer unbedeutender. Es steigt sogar die Wahrscheinlichkeit, dass sich positive und negative Abweichungen ausgleichen werden. Eben dieser Ausgleich ergibt am Ende ein realistisches und belastbares Ergebnis. In der empirischen Untersuchung zeigt sich dieser Zusammenhang in der geringeren Abweichung der Bewertungen im Durchschnitt über alle Funktionen. Hier lag die Abweichung bei weniger als 10%.

Die bisherigen Ergebnisse einer SWPL Anwendung zeigen, dass Verbesserungen nicht um einige Prozentpunkte, sondern ganze Größenordnungen möglich sind. "If an organization can realize a productivity gain of 200-500% through a product line approach, you don't need to worry if your model is off by some small integer factor. You aren't trying to convince someone who's worried about whether some value is 12% or 13%" [Clem05, S. 47].

Abschließend ist noch auf einen wichtigen Vorteil der Function Point Methodik hinzuweisen: Function Points sind die am weitesten verbreitete Aufwandsmessmethode. Bisher wurden sie bei mehr als 25.000 SW Projekten verwendet, während andere Metriken gerade einmal in 10 bis maximal 300 Projekten angewendet worden sind [Jone07, S. xix]. Der Bekanntheitsgrad[57] als auch die umfangreiche Datenbasis sind vorteilhaft bei der Verwendung von FPs. Allerdings bezieht sich der tatsächliche Vorteil im Rahmen dieser Arbeit lediglich auf die große Bekanntheit der Methodik, da die Ergebnisse der FP-Analyse ausschließlich innerhalb eines Unternehmens verwendet werden. Ein Vergleich mit externen Daten ist zwar grundsätzlich möglich, erfordert aber eine genaue Kenntnis der organisatorischen Strukturen, der internen Prozesse und der Rahmenbedingungen in den betroffenen Unternehmen. Der bereits oben angesprochene VAF reicht (und das zeigt auch die vielfach geäußerte Kritik) nicht aus, um die großen Unterschiede korrekt abzudecken und eine tatsächliche Vergleichbarkeit herzustellen. Dies wird auch bei der Überprüfung der Anwendbarkeit des Modells später in Teil B Kapitel 6 deutlich. Dort ist ein direkter Vergleich mit externen Benchmark-Werten nur bedingt möglich, da viele Besonderheiten zu berücksichtigen sind. Trotz dieser Schwierigkeiten wird ein solcher externer Vergleich in vielen Fällen von den Projektverantwortlichen oder dem Management gefordert. Dies ist auch der Grund für die

[57] Insbesondere die große Zahl an Personen, die in der Lage sind eine Function Point Analyse qualifiziert durchzuführen, bietet einen großen Vorteil bei der Akzeptanz des gesamten Modells.

Angabe eines solchen Vergleichs in Teil B Kapitel 6, allerdings mit klarer Aussage zu den damit verbundenen Schwierigkeiten und Ungenauigkeiten.

Eine Konzentration auf ein Unternehmen bietet hingegen den Vorteil, dass man keine Anpassungen an externe Vergleichswerte durchführen muss. Gleichzeitig ergibt sich dabei die Notwendigkeit, eigene Vergleichswerte zu generieren. Dies führt zu einem Mehraufwand, der allerdings aufgrund der höheren Qualität und einfacheren Vergleichbarkeit der Daten gut zu begründen ist. Die Vergleichswerte können bspw. erzeugt werden, indem ein Projekt vor der Einführung der SWPL zusätzlich untersucht wird. Dabei sollte darauf geachtet werden, dass für dieses Projekt weitgehend identische Rahmenbedingungen galten. In großen Unternehmen kann bereits der Wechsel der Abteilung oder des Unternehmensbereichs vergleichbare Auswirkungen haben wie ein Vergleich mit externen Daten. Meist wird es sich bei den ausgewählten Referenz-Projekten um bereits abgeschlossene Projekte handeln, wodurch die Menge der vorhandenen Daten grundsätzlich verbessert wird.

Das zugrundeliegende Vergleichsprojekt sollte eine c.p. Situation abbilden. Die Besonderheiten innerhalb eines Unternehmens werden damit berücksichtigt und der Fokus bei der Analyse der Veränderungen zwischen den beiden untersuchten Projekten kann allein auf die Einführung der SWPL gerichtet werden. Eventuell kann es notwendig sein, bekannte zusätzliche Faktoren, wie z.B. Reorganisationen oder eine größere Veränderung in der Entwicklungsabteilung (Outsourcing), getrennt zu berücksichtigen, um die c.p. Situation sicherzustellen.

Ermöglicht wird diese interne Fokussierung durch die Zielsetzung der Arbeit, die nicht darin besteht, eine allgemeine Aussage zur Vorteilhaftigkeit des SWPL Ansatzes abzuleiten. Ziel ist es vielmehr, für ein einzelnes Unternehmen und die dortige konkrete Anwendungssituation eine Empfehlung hinsichtlich des Einsatzes einer SWPL abzugeben. Und dafür ist eine alleinige Betrachtung des Unternehmens ausreichend. Vergleiche zu Benchmark-Daten können zusätzlich genutzt werden, um dem Management einen Anhaltspunkt über die eigene Produktivität und Aufstellung der Entwicklungsabteilung zu geben. Für die finale Empfehlung, ob sich die Einführung einer SWPL in der konkreten Situation für das Unternehmen lohnt, ist diese Benchmark-Betrachtung allerdings nicht notwendig.

6. Zusammenfassung Teil A

Im ersten Teil der Arbeit wurde eine Einführung in die grundlegenden Theorien gegeben. Dabei wurden wichtige Begriffe definiert und die bereits vorliegenden Arbeiten diskutiert. Zunächst wurde die historische Entwicklung und zukünftige Bedeutung von Embedded Systems im Automobil erläutert. Es zeigte sich, dass bereits heute ein Großteil der Innovationen auf elektronischen Systemen beruht und deren Anteil in Zukunft weiter deutlich zunehmen wird.

Umfassend wurde in Kapitel 2 die Grundidee der Produktlinie hergeleitet und der aktuelle Forschungsstand beschrieben. Anhand mehrerer Praxisbeispiele konnte das hohe Erfolgpotenzial aufgezeigt werden, wobei ebenfalls die möglichen Schwierigkeiten und Herausforderungen angesprochen worden sind. Schließlich wurde die Idee auf den Bereich der Software übertragen. Zahlreiche reale Anwendungen belegen das Potenzial der Idee, aber auch die besonderen Schwierigkeiten, die darin begründet sind, dass sich Software eben nicht wie ein technisches Produkt entwickeln lässt und die damit verbundene Unsicherheit zu Beginn der Entwicklung meist deutlich größer ist.

Ebenso war es notwendig die Besonderheiten von Embedded Systems aufzuzeigen. Trotz einer zunehmenden Angleichung von PCs und Embedded Systems bestehen immer noch klare Unterschiede, wie z.B. die stärkere Hardwareorientierung, die höhere Kostenbedeutung und die schwierigen Umweltbedingungen. Für den automobilen Bereich ergeben sich zusätzlich weitere Einschränkungen durch den geringen Bauraum, die nochmals verschärften Einsatzbedingungen und erhöhte Sicherheitsanforderungen. Der geringere Einsatz von standardisierten Methoden und Werkzeugen zusammen mit der längeren Produktlebensdauer bei gleichzeitig schneller werdender technischer Entwicklung bedeuten weitere Herausforderungen für die Verwendung von Software Produktlinien. Es ist evident, dass all diese Punkte zwar eine Verwendung von SW Produktlinien erschweren, aber keineswegs unmöglich machen. Die auch im embedded Bereich zu erwartenden und mit einigen ersten Beispielen bereits belegten Verbesserungspotenziale rechtfertigen eine tiefergehende Analyse.

Zum Schluss dieses Teils wurden Software-Bewertungsansätze diskutiert und deren Vor- und Nachteile erläutert. Eine solche Bewertung ist die Grundlage jeder ökonomischen Analyse. Trotz der bekannten Nachteile der am meisten verbreiteten Methode nach IFPUG wurde diese als Basis zur Bewertung für die vorliegende Arbeit ausgewählt. Im Rahmen einer empirischen

Studie der Unternehmensberatung McKinsey & Company wurde eine angepasste Methodik für Embedded Systems vorgestellt mit dem Namen Feature Points. Diese Methode ermöglicht wie auch die COSMIC Methode neben der reinen Nutzerperspektive weitere Betrachtungswinkel. Im Gegensatz zu COSMIC bleibt der Anwendungsaufwand allerdings selbst bei sehr frühen Zeitpunkten in der Entwicklung gering. Schließlich deuten erste Analysen auf eine Vergleichbarkeit der beiden Vorgehen mit einer geringen Fehlerwahrscheinlichkeit hin.

Teil A schafft damit die Grundlagen für die weiteren Schritte, die es einem Unternehmen mit vertretbarem Aufwand ermöglichen, eine ökonomische Bewertung des Produktlinienansatzes für automobile Software zu erstellen. Diese Schritte stellen den Hauptpunkt von Teil B dieser Arbeit dar.

Teil B. Ökonomische Betrachtung von SWPL im Automotive Bereich im Vorfeld einer Investitionsentscheidung

1. Einleitung

1.1. Motivation

In Teil A wurden die notwendigen Grundlagen für eine ausführliche Beschäftigung mit dem Thema SWPL im automobilen Umfeld gelegt. Diese werden in Teil B aufgegriffen und genutzt, um ein ökonomisches Modell zur Bewertung von Produktlinienansätzen in der Automobilindustrie im Detail herzuleiten.

Wie bereits in Teil A Kapitel 1 ausgeführt bietet die Idee der Software-Produktlinie eine Möglichkeit, die zunehmende Komplexität der elektronischen Systeme im Automobil zu beherrschen und gleichzeitig die Kosten und die Entwicklungszeit im Blick zu behalten. Allerdings stehen zahlreiche Unternehmen diesem Ansatz aufgrund des als hoch wahrgenommenen subjektiven Umsetzungsrisikos skeptisch gegenüber. Es fehlt die Möglichkeit, mit geringem Aufwand und zu einem frühen Planungszeitpunkt eine individuelle Risiko- und Nutzenabschätzung durchführen zu können. Dadurch würde dem SWPL Konzept eine breitere Akzeptanz verschafft, da eine größere Zahl an Unternehmen bereit wäre, den Ansatz umzusetzen. In der Wissenschaft wurde dieser Bedarf bereits vor einiger Zeit erkannt. Die entsprechende Literatur bezieht sich jedoch meist auf eine konkrete Ausgangssituation. Universell einsetzbare Konzepte liegen nicht vor. Wünschenswert wäre ein Ansatz, der sich leicht an die individuellen Bedürfnisse und Gegebenheiten des einzelnen Unternehmens anpassen lässt.

Genau dieses Ziel wird mit der Einführung des e-FACTS Modells verfolgt. **e-FACTS** steht für **economic Feature Point Analysis with convenient timing for Software Product Lines**. Teil B dient dazu, die Vorgehensweise des Modells im Detail zu erklären. Mit dieser Schritt-für-Schritt Anleitung werden Unternehmen allein auf Basis dieser Arbeit in die Lage versetzt, eine vollständige ökonomische Bewertung des SWPL Ansatzes für ihre konkrete Situation durchzuführen und damit eine fundierte Entscheidung hinsichtlich einer erfolgreichen Umsetzung zu treffen. Die einzige Voraussetzung ist die grundlegende Kenntnis der Function Point Methodik nach IFPUG, für die umfangreiche Literatur zur Verfügung steht. Die Arbeit leistet mit ihrer fokussierten operativen Ausrichtung einen wichtigen Beitrag in der wissenschaftlichen Forschung und schließt eine Lücke, durch die die Verwendung des SWPL Ansatzes außerhalb von universitären Projekten bisher behindert wurde.

1.2. Aufbau von Teil B

Im Folgenden wird die grobe Struktur von Teil B erläutert: Zu Beginn werden in **Kapitel 2** die unterschiedlichen Unternehmensziele diskutiert. Dies ist für die weitere Analyse obligatorisch, da das Unternehmensziel einen maßgeblichen Einfluss auf mögliche Investitionsentscheidungen, wie z.B. die Einführung einer SWPL, hat. Anschließend werden in **Kapitel 3** die bestehenden Modelle zur ökonomischen Bewertung von SWPL vorgestellt und verglichen. Kapitel 3 schließt mit einer Analyse der Schwachpunkte der einzelnen Methoden und einem Ausblick, wie diese durch das e-FACTS Modell angegangen werden. Neben der Betrachtung der ökonomischen Konsequenzen einer SWPL spielt zusätzlich die Qualität eine entscheidende Rolle. Dies wird im **Kapitel 4** im Detail diskutiert. **Kapitel 5** stellt den Hauptteil des Abschnitts B dar. Hier wird die eigentliche Methodik von e-FACTS vorgestellt. Der Aufbau dieses Kapitels folgt den eigentlichen Analyseschritten bei einer Anwendung des Modells. Zunächst wird der Basisfragebogen zur grundsätzlichen Abschätzung der Eignung des Unternehmens und dessen Produktpalette für eine mögliche SWPL vorgestellt. Anschließend folgt basierend auf den in Kapitel 3 vorgestellten wissenschaftlichen Arbeiten eine vereinfachte funktionsbasierte Berechnung. In Abschnitt 5.3 wird dann das Kernelement des e-FACTS Modells vorgestellt. Zum besseren Verständnis wird im gesamten weiteren Verlauf von Kapitel 5 auf ein einheitliches Anwendungsbeispiel zurückgegriffen, das hier eingeführt wird. In Kapitel 5.4 erfolgt die Komplexitätsmessung auf Basis der Feature Point Methodik. Ein entsprechendes Grundverständnis dieser Methodik wird für die weiteren Analyseschritte vorausgesetzt. Aufgrund der verfügbaren Literatur stellt dies jedoch keine Einschränkung dar.

Am Ende von Kapitel 5.4 ergibt sich bereits ein mögliches Ergebnis der Analyse: Eine ganzheitliche Bewertung einer geplanten oder bereits umgesetzten SWPL im Vergleich zu entsprechenden Einzelentwicklungen, und zwar unter Berücksichtigung aller relevanten individuellen Daten. Das vorliegende Modell benutzt zur Berechnung eben keine Durchschnittswerte, sondern greift soweit möglich auf die unternehmenseigenen Daten zurück. Dies ermöglicht im Vergleich zu den bereits vorhandenen Modellen eine deutlich bessere Abschätzung der späteren tatsächlichen Kosten.

In Abschnitt 5.5 erfolgt die Umwandlung der Feature Point Komplexitätswerte in monetäre Größen und eine Erweiterung zu einem vollständigen Business Case über die Laufzeit der SWPL. Abschnitt 5.6 beschließt das e-FACTS Modell mit einer Betrachtung der

Anwendungsmöglichkeiten in unterschiedlichen Unternehmenssituationen und bei unterschiedlichen Zielsetzungen. Das Basismodell unterstellt, dass das Unternehmen im Wettbewerb steht und daher die Bewertung hauptsächlich kosten- und zeitbasiert getroffen wird. Sollten aber zusätzlich weitere oder gänzlich andere Unternehmensziele im Vordergrund stehen (bspw. in einer monopolähnlichen Marktsituation), dann kann dies erhebliche Einflüsse auf die Bewertung haben.

Im Anschluss an die ausführliche Erklärung des Modells folgt in **Kapitel 6** eine Beschreibung der bisherigen Anwendungen des Ansatzes, die dem Nachweis der Praktikabilität des Modells dienen. Zwei Fallstudien werden detailliert besprochen und die gewonnenen Erfahrungen beschrieben, inkl. einer Sensitivitätsanalyse zur Berücksichtigung von Unsicherheit. Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick auf weitere Forschungsmöglichkeiten auf Basis dieser Arbeit.

2. Die grundlegenden Unternehmensziele beeinflussen den möglichen Erfolg von Software Produktlinien

2.1. Welche Ziele verfolgen die Unternehmen?

Jeder Tätigkeit von Unternehmen liegt ein bestimmter Zweck, ein Ziel zu Grunde. Ein Unternehmen hat dabei die Aufgabe, für seinen Betrieb eine möglichst einheitliche und übergreifende Zielfunktion zu formulieren [Wöhe00, S. 118]. Je nach Größe des Unternehmens ist dieses Oberziel in Unter- und Zwischenziele zu unterteilen. Nur dadurch werden die Ziele umsetzbar. Denn für die einzelne Abteilung sind die Folgen ihrer Entscheidungen schwer abzuschätzen, wenn allein ein Oberziel vorgegeben ist. Es bedarf einer genauen Abstimmung der einzelnen Unternehmensbereiche, um das gemeinsame Ziel zu erreichen. Ein wichtiger Grund dafür ist, dass die Zielsetzung für unterschiedliche Bereiche zu komplementären oder konkurrierenden Vorgaben führen kann, die sich gegenseitig negativ beeinflussen. Zusätzlich erhöhen die unterschiedlichen Entscheidungsträger die Komplexität der Zielfindung. So sind bei einem Unternehmen neben der Geschäftsführung oft auch Anteilseigner, Kapital- und Kreditgeber sowie die Arbeitnehmer beteiligt. In der Literatur wird daher von einem "Zielsystem" [Korn99, S. 36 und Bamb08, S. 26 ff.] gesprochen.

Gemeinnützige Vereine oder auch Stiftungen nehmen eine Sonderstellung beim Thema Ziele ein, da ein bestimmtes Ziel meist der Grund für ihre Gründung gewesen ist, z.B. Kleidung oder Lebensmittel an bedürftige Menschen kostengünstig zu verteilen. Es wird versucht, dieses Ziel unter den gegebenen Einschränkungen, insbesondere den verfügbaren finanziellen Mitteln, bestmöglich umzusetzen. Im Allgemeinen ist es für Unternehmen langfristig betrachtet allerdings notwendig, mehr zu erwirtschaften, als Kosten verursacht werden. Ansonsten verliert das Unternehmen seinen Finanzspielraum und kann schließlich seine Verbindlichkeiten nicht mehr bedienen. Das Unternehmen wäre in einem solchen Fall insolvent. Daraus abgeleitet ergibt sich: "Ein Unternehmer unterwirft sich dem Verlustrisiko, weil er die Chance zur Gewinnerzielung nutzen will. Weil er auf lange Sicht einen hohen Gewinn vorzieht, kann man die langfristige Gewinnmaximierung als oberstes Unternehmensziel ansehen" [Wöhe08, S. 389]. Allerdings handelt es sich bei der reinen Gewinnmaximierung als Hauptziel jeder marktwirtschaftlichen Tätigkeit um eine vereinfachte Darstellung, denn alle Entscheidungen sind unter Nebenbedingungen zu treffen. So ist bspw.

für einen Unternehmer, der ein Unternehmen aufgebaut hat, eine langfristige Beziehung entstanden. Hinsichtlich seiner persönlichen Gewinnmaximierung ist er kurzfristig auf die Erfolge des Unternehmens angewiesen und kann nicht alle Investitionsmöglichkeiten des Marktes nutzen. Er orientiert sich also an einem möglichen relativen Maximalgewinn unter Nebenbedingungen [Wöhe08, S. 18]. Diese Nebenbedingungen können sowohl als Minimalziele, die das Hauptziel Gewinnmaximierung kaum beeinflussen, als auch als Maximalziele, z.B. bei Umsatzmaximierung bei der der reine Gewinn in den Hintergrund tritt, auftreten [Bidl64, S. 99]. Daraus ergeben sich zahlreiche mögliche Zielkombinationen, die zwar alle eine Gewinnerzielungsabsicht voraussetzen, jedoch in unterschiedlichen Ausprägungen. Eine vollständige Auflistung findet sich bei [Bidl64, S. 103]. Die wichtigsten sind Gewinnlimitierung bei maximaler Macht (z.B. in monopolartigen Marktsituationen, oder zur Erreichung einer solchen Situation), Steigerung des Marktanteils, höhere Auslastung des Unternehmens (insbesondere relevant in Zeiten mit geringem Absatz und einem Überangebot an Produkten) und ein hohes Unternehmenswachstum. Wachstum ist dabei nicht gleichbedeutend mit einem höheren Marktanteil, denn hier spielt auch die Entwicklung des relevanten Marktes eine entscheidende Rolle.

Mitte der 1980er Jahre wurde ein neues Zielkonzept bedeutend: der Shareholder Value-Ansatz. Entwickelt wurde er von Alfred Rappaport [Rapp86]. Ziel ist die Maximierung des Unternehmenswertes, gemessen mittels einer Abzinsung der zukünftig erwarteten Zahlungsströme (Discounted Cash Flow). Dies ist gleichbedeutend mit einer Steigerung des Eigenkapitals eines Unternehmens. Dieser Ansatz stellt demnach die Eigenkapitalgeber, ohne Berücksichtigung des Fremdkapitals, in den Fordergrund. Dies ist der Hauptgrund für Kritik an diesem Ansatz [vgl. Hill96, S. 415ff.]. Der Stakeholder-Ansatz versucht dies zu umgehen, indem die Gruppe der zu berücksichtigenden Personen bei Unternehmensentscheidungen erweitert wird auf alle, die ein berechtigtes Interesse am Fortgang des Unternehmens haben, z.B. Arbeitnehmer, Kunden, Lieferanten und alle Kapitalgeber [Wöhe08, S. 55ff.].

In der Literatur finden sich zahlreiche Unterscheidungsmerkmale und Klassifizierungen von Unternehmenszielen, z.B. monetäre und nicht-monetäre Ziele, Zielhierarchien, die zeitliche Dimension, Zielbeziehungen, begrenzte und unbegrenzte Ziele, usw. [vgl. Wöhe08, S. 79ff.]. Im Folgenden wird eine Unterteilung in operationale und formale Unternehmensziele vorgenommen. Dabei werden die möglichen Ziele und die Auswirkungen, die die Einführung einer SWPL haben kann, erläutert. Im Rahmen der vorliegenden Arbeit wird davon

ausgegangen, dass insbesondere die finanzielle Zielkomponente für die anwendenden Unternehmen eine entscheidende Rolle spielt. Schließlich soll die vorgestellte Methode die Frage beantworten, ob die Einführung einer SWPL finanziell sinnvoll ist oder nicht. Um diese Frage zu beantworten, benötigt man Informationen über die Zielsetzung der Unternehmung. Neben der mit einer SWPL-Einführung verbundenen Investition spielt die zeitliche Komponente eine wichtige Rolle. Es ist von entscheidender Bedeutung, mit welcher Kraft das Unternehmen die neue Idee umsetzen will, d.h. welchen Zeithorizont man der SWPL einräumen möchte, um ihre finanziellen Vorteile auszuspielen. Zusammen mit den individuellen Gegebenheiten des einzelnen Unternehmens ergibt sich daraus eine erste mögliche Antwort. Zusätzlich existieren aber viele weitere Aspekte, u.a. die Bereitschaft eines Unternehmens in neue Methoden zu investieren, sowie Risiken einzugehen und die Stärke des Wettbewerbs im relevanten Markt.

2.2. Welchen Einfluss haben Software Produktlinien auf die Zielerreichung?

SWPL können eine Vielzahl von Unternehmenszielen unterstützen. Hauptsächlich basieren diese auf dem grundlegenden SWPL Konzept der strategischen Wiederverwendung. Für eine umfassende Übersicht der Vorteile von Wiederverwendung bei bestimmten Unternehmenszielen siehe auch [SPC93, S. 3-6].

Zu den klassischen Zielsetzungen einer SWPL gehört eine Erhöhung der Produktivität und damit bei gleicher Personalstärke eine Verkürzung der Entwicklungszeit, was sich in einer kürzeren Time-to-market widerspiegelt. Die eingesparte Entwicklungsarbeit würde dann direkt zu einer Reduzierung der Entwicklungskosten führen. Die freien Kapazitäten könnten dann für die Entwicklung zusätzlicher Produkte oder für Kosteneinsparungen genutzt werden. Es bestünde jedoch auch die Möglichkeit, diese Kapazitäten für zusätzliche Tests heranzuziehen und damit die Qualität der Produkte zu erhöhen. Eine spätere Wiederverwendung dieser ausführlich getesteten Komponenten würde sich ebenfalls in Zukunft positiv auf die Qualität auswirken. Bereits an diesen sehr einfachen Beispielen sieht man, dass eine SWPL je nach Zielsetzung unterschiedlich ausgestaltet werden kann.

Im Folgenden wird explizit auf die Auswirkungen einer SWPL auf unterschiedliche Unternehmensziele eingegangen. Dabei erfolgt eine Unterscheidung nach den operationalen Zielen (Kosten, Zeit und Qualität) und anschließend den formalen Zielen.

2.2.1. Operationale Ziele

Operationale Ziele sind direkt messbar. Sie bilden die Basis für alle weiteren, sozusagen abgeleiteten Unternehmensziele. In der Betriebswirtschaftslehre wird hierfür häufig vom Zieldreieck gesprochen. Dabei werden die drei Elemente Kosten, Zeit und Qualität betrachtet [vgl. Adam98, S. 28f.]. Allerdings sind diese nicht voneinander unabhängig. Jede Veränderung eines dieser Ziele bedingt eine gegenteilige Veränderung bei mindestens einem weiteren Ziel. Eine gleichzeitige Optimierung aller drei Zielsetzungen ist c.p. kaum möglich. Sollten sich diese Rahmenbedingungen jedoch ändern, dann wäre auch eine Verbesserung aller drei operativen Zieldimensionen möglich. Dies wäre bspw. durch einen technologischen Quantensprung möglich. Aufgrund der neuen Technologie werden die bisherigen Beziehungen zwischen Kosten, Qualität und Zeit auf eine neue Ebene gehoben. In der Automobilindustrie sind hier u.a. die Einführung der Fließbandfertigung und der Umstieg auf selbsttragende Karosserien zu nennen. Aber auch die bessere Ausschöpfung der Potenziale einer bestehenden Technologie ermöglicht die Verbesserung in allen drei Dimensionen. Der japanische Kaizen Ansatz[58] basiert auf dieser Überlegung. Hierbei wird durch eine unternehmensweite und andauernde Konzentration auf ständige Verbesserung versucht, das bestmögliche Ergebnis zu erreichen. Der Fokus liegt dabei zunächst auf einer hohen Produktqualität, ohne dabei allerdings höhere Kosten zu verursachen. Möglich wird dies durch die Nutzung von implizit im Unternehmen vorhandenem Wissen, z.B. durch ein internes Vorschlagswesen, gute Weiterbildung der Mitarbeiter und Einrichtung eines Qualitätsmanagementsystems.

Grundsätzlich gilt es zu beachten, dass eine Qualitätsverbesserung c. p. immer entweder eine Kostenerhöhung, eine Verlängerung der benötigten Zeit oder beides zur Folge haben wird. Die Einführung einer SWPL bedeutet eine Änderung der Rahmenbedingungen. Es wird möglich, die Zieldimensionen auf eine neue Ebene zu heben. Die individuellen Auswirkungen einer Produktlinie auf die operationalen Ziele werden im Folgenden besprochen.

2.2.1.1.Kosten

"Kosten sind der mit Preisen bewertete Verzehr von Produktionsfaktoren, der durch die betriebliche Leistungserstellung (und –Verwertung) verursacht wird" [Wöhe08, S. 305]. Dabei sind Kosten von den aus der Finanzbuchführung bekannten Aufwendungen

---

[58] Der Kaizen Ansatz ist auch bekannt als Kontinuierlicher Verbesserungsansatz oder Continuous Improvement. Für weitere Details siehe [Adam98, S. 64 ff.].

abzugrenzen. Aufwendungen "stellen die … periodisierten, d.h. auf die Abrechnungsperiode bezogenen Ausgaben dar, die aus dem Güterverbrauch, Leistungs- oder Werteverzehr oder sonstigen das Reinvermögen mindernden Ausgaben aus der jeweiligen Periode resultieren" [Bätg01, S. 3]. Aufwendungen entstammen der externen Rechnungslegung. Sie sind stärker periodenorientiert und an gesetzliche Vorgaben gebunden, die dazu dienen, die Vergleichbarkeit der Daten zu erhöhen. Kosten dienen hingegen eher der ökonomisch korrekten Darstellung und sollen daher im weiteren Verlauf der Arbeit verwendet werden. Für eine detaillierte Abgrenzung von Leistungen/Kosten und Ertrag/Aufwand siehe [Coen99, S.39].

Die bei der Produktherstellung anfallenden Kosten sind zu reduzieren. Hierbei sind zahlreiche Faktoren zu berücksichtigen, denn theoretisch würde man die niedrigsten Kosten bei einer völligen Einstellung der Produktion erzielen. Dies kann nicht das Ziel des Unternehmens sein. Ebenso spielt die zeitliche Dimension eine wichtige Rolle. So ist es bspw. in den meisten Fällen zunächst notwendig, eine Investition zu tätigen, um überhaupt in der Lage zu sein, eine Produktion zu starten. Die Grundidee einer Produktlinie ist die strategische Wiederverwendung von Produktbausteinen. Es geht also nicht allein darum, ein einzelnes Produkt herzustellen, sondern die Kosten über mehrere Produkte hinweg zu optimieren. Die Anfangsphase einer Produktlinie kann demnach ebenfalls als Investition angesehen werden. Die Entwicklung der einzelnen Komponenten ist aufgrund der angestrebten Wiederverwendbarkeit im Allgemeinen kostenintensiver als individuelle Lösungen. Dies liegt u.a. an dem notwendigen zusätzlichen Koordinationsaufwand, den zusätzlich standardmäßig vorhandenen Funktionen und dem komplexeren Testing. Nach Einrichtung der Produktlinie können neue Produkte basierend auf den vorhandenen Komponenten relativ einfach und mit geringen zusätzlichen Kosten erstellt werden. Es sind zwar individuelle Anpassungen und Tests notwendig, eine komplette Neuentwicklung, die mit deutlich höheren Kosten verbunden wäre, entfällt jedoch. Die Investition in eine Produktlinie lohnt sich, wenn die durch die mehrfache Verwendung über Zeit eingesparten Entwicklungskosten die anfänglichen Zusatzkosten übersteigen. In der Literatur finden sich zahlreiche Aussagen über die durchschnittlich zu erwartende Anzahl an Wiederverwendungen, nach denen sich eine Investition in eine SWPL lohnt. Dieser Break Even, also der Punkt an dem beide Entwicklungsmethoden die gleichen Kosten verursachen würden, wird im allgemeinen bei zwei bis drei, bei einigen wenigen Quellen auch vier Produktentwicklungen angenommen [vgl. Clem03, Weis99, S. 45ff.]. Hierbei ist natürlich zu beachten, dass es sich lediglich um

eine Durchschnittsbetrachtung handelt. Für einzelne Szenarien können die Werte aufgrund von zahlreichen individuellen Besonderheiten dennoch stark abweichen.

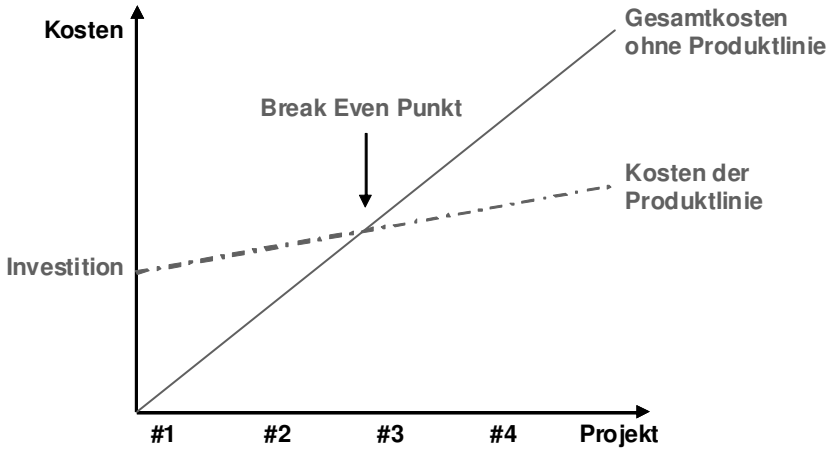

Abbildung 17 – Kostenvergleich Entwicklung mit und ohne SWPL [Clem03]

2.2.1.2.Zeit

Die Dimensionen Kosten und Zeit sind stark voneinander abhängig und lassen sich über den Faktor Arbeit gegenseitig beeinflussen. Je mehr Maschinen oder Personal zur Entwicklung eines Produktes eingesetzt werden, desto weniger Zeit wird dafür benötigt. Diese direkte Austauschbarkeit der beiden Faktoren unterliegt jedoch gewissen Grenzen. So erfordert die Zusammenarbeit in einem größeren Team auch immer eine komplexere Koordination. Ab einem bestimmten Niveau ist eine weitere Zeitverkürzung kaum noch oder lediglich zu unwirtschaftlichen Kosten möglich.

Wie jedoch bereits in der Einleitung zu Kapitel 2.2.1. erläutert, ändert die Einführung einer Produktlinie die Rahmenbedingungen und die Aufgabenstellung für das Unternehmen grundlegend. Während die Entwicklung sonst durch die Anzahl und den Umfang der zu entwickelnden Produkte bestimmt wurde, spielt bei einer Produktlinie der Anteil der möglichen Wiederverwendung eine entscheidende Rolle. Je höher dieser Anteil ist, desto stärker kann die individuelle Entwicklungszeit reduziert werden. Da nicht davon auszugehen ist, dass die Entwicklungskapazitäten eines Unternehmens kurzfristig deutlich vergrößert werden können, führt die Einrichtung einer Produktlinie beinahe zwangsläufig zu einer zeitlichen Verzögerung bei der Fertigstellung neuer Produkte. Nach erfolgreichem Aufbau einer einheitlichen Produktbasis ergibt sich dann eine reduzierte Entwicklungszeit für alle Folgeprodukte basierend auf der SWPL. Anschaulich ausgedrückt bedeutet dies: Das Erlernen einer Tastenkombination bei der Bedienung eines PC-Programms ist nicht so intuitiv wie die

Bedienung per Maus. Zusätzlich erfordert das Erlernen Zeit. Ist diese Hürde einmal überwunden, kann der Nutzer bei jeder Verwendung der Funktion Zeit einsparen. Es wird deutlich, dass eine solche Übung lediglich bei einer gewissen Nutzungshäufigkeit der Funktion sinnvoll ist. Dies gilt ebenso für die Anzahl der Wiederverwendungen der Komponenten einer SWPL.

Siemens VDO, ein deutscher Automobilzulieferer hat das SWPL-Konzept bereits 2005 umgesetzt. Nach einer 2-jährigen Entwicklungszeit, wobei die zusätzliche Entwicklungsarbeit in die normalen Arbeitsabläufe integriert wurde, reduzierte sich der durchschnittliche Entwicklungsaufwand für eine an die Bedürfnisse des Automobilherstellers angepasste Motorsteuerungssoftware ca. um die Hälfte.[59]

2.2.1.3.Qualität

Auch die Qualität steht in Abhängigkeit von den beiden anderen Zielgrößen. Besteht z.B. ein erheblicher Zeitdruck aufgrund eines festgelegten Vermarktungsstarts, ist es möglich, dass die Qualität zugunsten einer kürzeren Entwicklungszeit reduziert wird. Dies erreicht man durch eine Verringerung der Tests und der dafür vorgesehenen Zeiten. Sinnvoll erscheint ein solches Vorgehen allerdings nur, wenn mit einer verspäteten Fertigstellung erhebliche Kosten verbunden wären, da eine reduzierte Qualität aufgrund der zu erwartenden Garantiefälle ebenfalls erhebliche Ausgaben verursachen würde. Hinzu käme ein möglicher Imageschaden für das Unternehmen, dessen Auswirkungen nur schwer zu kalkulieren sind. Ein Hersteller von Mobiltelefonen erlebte dieses Dilemma 2005 bei der Entwicklung einer neuen Produktgeneration. Aufgrund eines sehr engen Zeitplans war es für die Entwicklung eines neuen Media Players notwendig, die Testzeit drastisch zu reduzieren. Das Produkt wurde zwar rechtzeitig den Netzbetreibern zur Verfügung gestellt, aber die Anzahl der bei den dortigen Operator Acceptance Tests festgestellten Fehler lag um das 3 bis 4-fache über den sonst üblichen Werten. Dies erforderte einen enormen zusätzlichen Testaufwand vor Auslieferung des Produktes. Als Folge lag der Test-Anteil am Gesamtaufwand zur Entwicklung des Produktes am Ende bei ca. 50%, während die reine Code-Entwicklung nur einen Anteil von etwas über 30% und die Spezifikation und das Design lediglich jeweils 10%

[59] Diese Daten basieren auf einem Workshop mit dem Entwicklungsleiter bei Siemens VDO Ende 2005. Eine ausführliche Darstellung findet sich im Kapitel 5.2, da es sich hierbei um eines der beiden Praxisbeispiele handelt.

der Entwicklungszeit betrug. [60] Während der Anteil für Spezifikation und Design bei Software Projekten im Durchschnitt liegt, wären beim Entwicklungsanteil normalerweise deutlich höhere Werte von 52 bis 64% zu erwarten. Dies geht zu Lasten des Testanteils der durchschnittlich zwischen 19 und 31% betragen sollte [vgl. Boeh00, S. 307ff.].

Umfangreiches Testen ist bei einer SWPL aufgrund der angestrebten Wiederverwendung von einzelnen Bestandteilen besonders wichtig. “One of the major economic advantages of software reusability is prefaced on the assumption that reusable code approaches the level of zero defects. If indeed a reusable module is at a zero-defect level, then its affect on productivity would be about 500 percent. It takes about four times as long to debug 1,000 lines of source code as it does to actually write the code.” [Jone07, S. 446] Ein Fehler in einem Core Asset einer SWPL würde deutlich höhere Kosten bei der Beseitigung erzeugen, als würde man den Code neu schreiben. Zusätzlich spielt die Verwendungshäufigkeit eine wichtige Rolle. Aufgrund der weiteren Verbreitung in mehreren Produkten könnte demnach allein durch die höhere Produktanzahl ein einzelner Fehler einen größeren Schaden verursachen. Der anfänglich höhere Testaufwand rechnet sich über die Laufzeit der SWPL, da die einmal getestete Komponente mit geringem zusätzlichem Integrations-Testaufwand in neuen Produkten verwendet werden kann. Gleichzeitig ist die spätere Anpassung von Standardteilen einfacher möglich, da die entsprechenden Abhängigkeiten geringer oder zumindest besser dokumentiert und damit kontrollierbar sind. Dafür ist ein geeignetes Variantenmanagement notwendig. Ebenso sind notwendige Anpassungen aufgrund externer Einflüsse, wie z.B. rechtliche Änderungen oder technische Neuentwicklungen, einfacher umsetzbar. Die Verwendung von standardisierten Bauteilen ermöglicht eine zentrale Implementierung der Änderungen bei einigen wenigen Komponenten anstelle bei jedem Produkt einzeln. Dabei kann sich das Unternehmen dennoch sicher sein, die Änderungen vollständig umgesetzt zu haben.

Aus den genannten Punkten ist mit einer gesteigerten Qualität nach Einführung einer SWPL zu rechnen. Allerdings ist es aufgrund der bereits oben erwähnten Austauschbarkeit von Qualität gegen Zeit oder Kosten nur schwer möglich, diese Größe wirklich genau zu messen. Eine Untersuchung der Unternehmensberatung McKinsey hat gezeigt, dass die Produktivität bei Embedded Software Projekten im Automotive Bereich mit der Reife der gesamten

---

[60] Die Werte stammen von persönlichen Erfahrungen im Rahmen eines Feature Point Counting Workshops bei dem entsprechenden Mobiltelefonhersteller.

Embedded Systems Entwicklung korreliert. Bei der Qualität (gemessen als Fehler pro Function Point) scheint zwar ebenso einen Zusammenhang zu existieren; dieser ist allerdings deutlich schwächer ausgeprägt [McKi06, S. 28]. Im Rahmen der vorliegenden Arbeit wird der Fokus daher verstärkt auf die Kosten- und Zeit-Dimension gelegt. Qualität wird eher als Hygienefaktor betrachtet, was zur Folge hat, dass ein gewisses Mindestniveau nicht unterschritten werden darf (siehe auch Teil B Kapitel 4).

2.2.2. Formale Unternehmensziele

2.2.2.1.Gewinn

Aus den operationalen Zielen lässt sich eine weitere Zielgröße ableiten: der Unternehmensgewinn. Allgemein wird zwischen kalkulatorischem Gewinn, der unternehmensintern in der Kostenrechnung verwendet wird und sich aus der Differenz von Erlösen und Kosten ergibt, und pagatorischem Gewinn, der zur externen Kommunikation in der Bilanz verwendet wird und die Differenz aus Aufwand und Ertrag wiedergibt, differenziert. Die Unterschiede beider Ansätze beziehen sich auf die zeitliche Zurechnung von Geschäftsvorfällen je nachdem für wen die Informationen bestimmt sind. Diese grundsätzlich verschiedenen Herangehensweisen in der Finanzbuchführung und im Controlling sollen im Rahmen der vorliegenden Arbeit jedoch nicht vertieft werden. Es genügt, Gewinn als Differenz zwischen bewertetem Ertrag und den für die Erzielung dieses Ertrags notwendigen bewerteten Einsatz von Produktionsfaktoren [Wöhe08, S. 304] zu verstehen. Allgemein kann man diesen Wert auch als Erfolg eines Unternehmens bezeichnen. Für eine ausführlichere Diskussion des Begriffes Gewinn wird auf die entsprechende Literatur verwiesen [z.B. Coen99, S. 23ff.].

Alle drei oben angesprochenen Dimensionen des Zieldreiecks führen kurz- oder langfristig zu einem gesteigerten Gewinn. Die Dauer, bis sich eine Veränderung im Gewinn zeigt, steigt dabei von Kosten, über Zeit bis hin zur Qualität deutlich an. Insbesondere die letztgenannte Dimension wirkt nur indirekt und damit zeitverzögert; nämlich erst dann, wenn die verkauften Produkte innerhalb der Garantiezeit weniger reklamiert werden oder (noch langfristiger) wenn die gesteigerte Qualität bei den existierenden und potenziellen Kunden eine positive Imagewirkung erzeugte und dadurch höhere Umsätze generiert und neue Kundengruppen adressiert werden.

Eine alleinige Fokussierung auf den Unternehmensgewinn reicht allerdings nicht aus. Natürlich ist langfristig jedes Unternehmen[61] auf einen ausreichenden Gewinn angewiesen, um handlungsfähig zu bleiben. Um dies zu erreichen, können zusätzlich zu den operationalen auch weitere formale Ziele gesetzt werden. Dabei wird meist bewusst – zunächst zeitlich begrenzt – ein geringerer Gewinn akzeptiert, um später mittels höherer Gewinne davon zu profitieren. Wie in den nachfolgenden Punkten dargestellt wird, führen daher auch die anderen formalen Ziele, wenn sie richtig verfolgt werden, langfristig zu einem höheren Unternehmensgewinn. Dieser stellt damit die übergreifende Zielgröße da.

2.2.2.2. Wachstum

Bei den weiteren möglichen Unternehmenszielen ist zunächst die Frage nach dem Grund für die Verfolgung dieses Ziels sinnvoll. Bei der Absicht, ein solides Wachstum zu erreichen, spielt insbesondere die Entwicklung des relevanten Marktes eine entscheidende Rolle. In einem stagnierenden Markt erweist sich ein Wachstum grundsätzlich als schwieriger im Vergleich zu einem stark expandierenden Markt. Ebenso sollte beachtet werden, was als Bezugsgröße für das angestrebte Wachstum herangezogen wird, z.B. die Unternehmensgröße, die Mitarbeiterzahl, der Umsatz oder der Marktanteil.

Unternehmen fokussieren ihre Zielsetzung meist auf Wachstum, wenn sich auch der entsprechende Markt stark positiv entwickelt. Ansonsten laufen Unternehmen Gefahr, durch einen sinkenden Marktanteil Nachteile zu erleiden, wie z.B. geringere Einkaufsmacht gegenüber Zulieferern, schlechtere Kostenbasis, höhere Entwicklungskosten pro Produkt, geringere Netzwerkeffekte für die eigenen Kunden. In einem solchen Markt ist mit starken Veränderungen der Produkte innerhalb kurzer Zeit zu rechnen. Vielfach wird es sich dabei um echte Produktrevolutionen handeln. Diese Rahmenbedingungen schränken den Einsatz von SWPL ein, da für einen erfolgreichen Einsatz eine gewisse Stabilität der grundsätzlichen Produkteigenschaften erforderlich ist. Sollte dies nicht der Fall sein, muss die Asset Base bei jeder Wiederverwendung zu stark angepasst werden, oder der Anteil der wiederverwendbaren Komponenten fällt zu klein aus. Zusätzlich erscheint es in einem sehr dynamischen Markt schwieriger, das notwendige zeitliche Investment für die Entwicklung der Asset Base leisten zu können. Die Unternehmen sind darauf konzentriert, die neuesten Funktionen rechtzeitig in ihre Produkte zu integrieren. Dies bedeutet nicht, dass eine SWPL grundsätzlich unmöglich

---

[61] Ausnahmen bilden Unternehmen, die staatliche Subventionen erhalten, oder deren Geschäftsmodell grundsätzlich als "non-profit" Organisation aufgebaut ist.

wäre. Jedoch sollten vor einem Einsatz die zusätzlichen Risiken genau analysiert werden. Das vorliegende Modell ist dafür geeignet, indem es für jede Komponente eine individuelle Wiederverwendungsanalyse basierend auf einer finanziellen Bewertung durchführt.

In gesättigten und entwickelten Märkten wird man ein Wachstumsziel eher bei neuen Marktteilnehmern antreffen. Ansonsten ist eine Wachstumsstrategie mit einem höheren Marktanteil verbunden. Dieser ist nur durch Verdrängung anderer Wettbewerber möglich. In einem solchen Marktumfeld ist der Wettbewerb meist intensiver und es herrscht ein höherer Kosten- und Innovationsdruck. Wobei Innovationen hier als langsame, teilweise marginale Produktverbesserungen zu verstehen sind. Hier sind die Vorteile der SWPL besonders sichtbar. Meist akzeptieren die Kunden eine kurze Verzögerung durch den Aufbau einer Asset Base bei der SWPL Einführung. Auf der anderen Seite führen die später möglichen kundenindividuellen Produkte in Verbindung mit hoher Qualität und niedrigen Kosten zu einem Wettbewerbsvorteil. Die kürzeren Entwicklungszeiten führen zudem zu einem Zeitvorsprung beim Angebot neuer Funktionen. Schließlich können insbesondere in einem gesättigten Markt neue Kundensegmente erschlossen werden, indem eine SWPL das Angebot eines individuell konzipierten Produktes ermöglicht.

2.2.2.3.Rentabilität

Die Rentabilität bezeichnet das Verhältnis des Periodenerfolgs eines Unternehmens zu einer bestimmten Kenngröße. Dabei kann es sich um das eingesetzte Kapital, sowohl Eigen- als auch Gesamtkapital, oder den Umsatz handeln [Wöhe08, S. 44]. Das Unternehmen kann die Rentabilität beeinflussen, indem die Einflussgrößen verändert werden. Diese sind nicht unabhängig voneinander, so bewirkt bspw. ein höherer Umsatz meist auch einen höheren Gewinn. Ohne eine grundsätzliche Veränderung weiterer Unternehmensparameter oder externer Einflussfaktoren (wie z.B. Preisdruck oder Rohstoffkosten) wird die Rentabilität unverändert bleiben. Für den Einfluss auf eine SWPL spielt die bereits oben angesprochene Frage nach dem Grund für diese Unternehmenszielsetzung eine wichtige Rolle. Es ist anzunehmen, dass Rentabilität aus Unternehmenssicht nicht das oberste Ziel ist, solange eine gesunde Rentabilität gesichert ist und das Unternehmen nicht mit jedem verkauften Produkt Verluste realisiert. Vielmehr wird ein solches Ziel extern vorgegeben, bspw. durch Aktionäre.[62] Durch eine SWPL kann ein Unternehmen, wie bereits oben erläutert, positive

[62] Ein aktuelles Beispiel ist die Finanzbranche, bei der in den letzten Jahren Eigenkapitalrenditen von über 20% üblich geworden sind. Solche Werte sind nur mit einem hohen Risiko und/oder einer geringen Eigenkapitalquote (sprich einer

Effekte auf Kosten, Zeit und Qualität und damit auf den Gewinn realisieren. Es ist anzunehmen, dass dies ebenso die Rentabilität eines Unternehmens positiv beeinflusst. Andere Maßnahmen, wie z.B. Kostensenkungen durch Rationalisierung, wirken allerdings deutlich direkter auf die Rentabilität und sind demnach vorzuziehen.

Ein weiterer Aspekt der Rentabilität im Zusammenhang mit Produktlinien betrifft die Sicht auf einzelne Produkte. Hier können sich durch eine SWPL deutliche Vorteile ergeben. So kann durch die kostengünstige Entwicklung zusätzlicher Varianten die Möglichkeit entstehen, neue Marktsegmente zu erschließen, für die die Entwicklung eines individuell angepassten Produktes bisher aufgrund der niedrigen Stückzahlen nicht rentabel erschien.

2.2.2.4.Liquidität

Jedes Unternehmen benötigt ein gewisses Maß an liquiden finanziellen Mitteln, um seinen aktuellen Zahlungsverpflichtungen nachkommen zu können. Im Allgemeinen ist für die Erstellung von Produkten zunächst der Ankauf von Rohstoffen oder Vorprodukten notwendig. Der Kapitalrückfluss zum Unternehmen erfolgt zeitlich verzögert erst nach dem Verkauf der eigenen Produkte. Für ein gesundes Unternehmen stellt die Sicherstellung einer soliden Finanzbasis in der Regel kein Problem dar. Erst in Krisenzeiten kann dies relevant werden, wenn bspw. plötzlich ein Großteil der erwarteten Bestellungen wegfällt. Dies bedeutet, dass es sich bei einem Fokus auf die Liquidität meist um ein kurzfristiges Ziel handelt. Eine erfolgreich umgesetzte SWPL wird zwar langfristig die Liquidität eines Unternehmens eher erhöhen. Unter Berücksichtigung der notwendigen Investition bei der Umsetzung einer SWPL erscheint dieser Ansatz jedoch zur kurzfristigen Sicherung oder Steigerung der Liquidität ungeeignet.

2.2.2.5.Image und Innovationskraft

Unter Images werden "objektiv und subjektiv verzerrte Vorstellungsbilder verstanden." [Meff98, S. 75]. Dabei beschreibt es die "Einstellung einer Person zu einem Meinungsgegenstand" [Andr76, S. 215]. Gleichzeitig kann ein Image als mehrdimensionales und ganzheitliches Bild einer Zielgruppe zu einem Gegenstand gesehen werden [Trom09, S. 155]. Das Image eines Unternehmens wird durch zahlreiche Faktoren beeinflusst, die nicht immer im direkten Einflussbereich des Unternehmens liegen. Eine SWPL kann jedoch in einigen Bereichen helfen, ein positives Image eines Unternehmens aufzubauen oder zu

---

hohen Verschuldung) realisierbar. Die Folgen haben sich im Jahre 2007 mit dem Beginn der weltweiten Finanzkrise gezeigt.

bestätigen. Die bessere Erfüllung der Kundenwünsche durch eine höhere Zahl individualisierter Produkte, eine schnelle Veröffentlichung von neuen Produkten, sowie niedrige Kosten bei gleichzeitig hoher Qualität, sind positive Imagefaktoren. Allerdings ist zu berücksichtigen, dass eine SWPL in einem hauptsächlich von Innovationen getriebenen Markt nicht ihr volles Potenzial realisieren kann. Dafür wären die notwendigen Anpassungen an der Asset Base, also dem für eine mehrfache Verwendung gedachten Funktionskern, zu groß oder die mögliche Asset Base selbst zu klein.

Demnach kann eine SWPL das Image eines Unternehmens positiv beeinflussen, bei der Fokussierung auf Innovationen sind dem Konzept allerdings Grenzen gesetzt.

3. Analyse und Bewertung bestehender ökonomischer Modelle für Software Produktlinien

Gemeinsam mit dem zunehmenden Interesse an Software-Produktlinien entwickelten sich auch Ansätze zur ökonomischen Bewertung von Produktlinien. Nachfolgend werden die wichtigsten Modelle vorgestellt und mit Hinblick auf die Nutzung für die weitere Arbeit kurz bewertet. Eine sehr umfangreiche Zusammenstellung ökonomischer Modelle zum Thema Wiederverwendung von Software findet sich bei [Wile99], der insgesamt 25 Modelle analysiert und vergleicht. Derzeit konzentrieren sich die in der Literatur vorhandenen Modelle fast ausschließlich auf den ökonomischen Aspekt. Weitere mögliche Zielsetzungen einer Software-Produktlinie, wie beispielsweise Qualitäts- oder Innovationsführerschaft, werden nicht explizit oder lediglich am Rande berücksichtigt. Daher wird hier neben der reinen ökonomischen Beschreibung und Bewertung auch die Möglichkeit des Ansatzes zur Bewertung weitergehender Unternehmensziele überblicksartig angesprochen.

3.1. Grundsätzliches zu Return on Investment Ansätzen

Grundsätzlich kann jede ökonomische Analyse auf einen Return on Investment Ansatz (ROI) zurückgeführt werden. Der notwendige Mehraufwand, um eine Software wieder verwendbar zu gestalten, wird dabei als Investment verstanden [Wile99, S. 3]. Die späteren Erträge des Investments ergeben sich aus den Einsparungen durch die tatsächliche Wiederverwendung im Gegensatz zu einer ansonsten notwendigen Neuentwicklung.

Die frühen Modelle zur Bewertung von Wiederverwendung basieren meist auf einer globalen Formel. Die Einflussgrößen werden jeweils als ein Durchschnittswert für das gesamte Projekt bestimmt und die daraus resultierenden Einsparungen ermittelt. Eines der ersten Modelle wurde von Gaffney und Durek entwickelt [Gaff89]. Die entsprechende Formel lautet:

Gleichung 2

$$C = R\left(b + \frac{E}{n} - 1\right) + 1$$

Wobei:

- C = Kosten für ein bestimmtes System mit Wiederverwendung, gemessen im Verhältnis zu den Kosten einer individuellen Entwicklung ohne Wiederverwendung
- R = Anteil des wiederverwendeten Codes am Gesamtsystem ($R \leq 1$)

- b = relative Kosten, die bei der Nutzung von existierendem Code in einem neuen System anfallen
- E = relative Kosten für die Erzeugung wieder verwendbaren Codes
- n = Anzahl der geplanten Wiederverwendungen (inkl. des ersten Produktes)

Die Logik des Modells lässt sich gut anhand eines einfachen Beispiels nachvollziehen. Nehmen wir an, ein Unternehmen plant die Entwicklung von 3 ähnlichen Produkten, die als SWPL realisiert werden könnten. Im Modell wird diese Produktanzahl direkt als die Anzahl der Wiederverwendungen (n) übernommen. Das Unternehmen nimmt an, dass die Entwicklung des ersten Produktes zu 50% (R) direkt auf Wiederverwendbarkeit ausgelegt wird und dieser Teil später auch tatsächlich bei allen 2 weiteren Produkten verwendet wird. Der Mehraufwand zur Erstellung wieder verwendbarer SW (E) beträgt 150% des normalen Aufwands für individuelle SW-Entwicklungen. Bei der tatsächlichen Wiederverwendung des Codes entsteht ein Anpassungsaufwand von ca. 10% des ursprünglichen Entwicklungsaufwands (b). Damit ergibt sich laut dem Modell von Gaffney und Durek ein Wert für C von 0,8. D.h. die Wiederverwendung ergibt eine Einsparung über die gesamte Laufzeit der SWPL von ca. 20% gegenüber einer Einzelentwicklung aller 3 Produkte.

Gleichung 3

$$C = 0,5 * (0,1 + 1,5/3 - 1) + 1 = 0,8$$

Das Modell verwendet jeweils nur einen Wert für das gesamte System. Es wird nicht nach einzelnen Komponenten unterschieden und als Basis für die Aufwandsbewertung werden Source Lines of Code (SLoC) verwendet. Ebenso werden weitere Einflüsse, wie z.B. Wartungskosten oder die zeitliche Abfolge der Zahlungsströme über mehrere Jahre hinweg nicht berücksichtigt. Zahlreiche Autoren haben eine solche vereinfachte Berechnung vorgeschlagen [vgl. Boec04]. Grund für dieses Vorgehen ist vielfach die sehr dünne Datenlage, die es erschwert detaillierte Aussagen hinsichtlich der zu erwartenden Kosten zu treffen. Grundsätzlich ist gegen die Idee der einfachen Umsetzung nichts einzuwenden. Insbesondere für eine sehr frühe Entscheidung ist ein einfacher Ansatz notwendig, da kaum belastbare Daten vorhanden sind. Bezugnehmend auf das Modell von Gaffney und Durek ist es selbst bei diesem vergleichsweise einfachen Modell schwierig, die wichtigen Parameter korrekt abzuschätzen, wenn keine individuelle Erfahrung im Bereich Wiederverwendung vorhanden ist. Ebenso ist zu beachten, dass eine SWPL aufgrund ihrer langen Laufzeit, der

getrennten Entwicklung von Domain und Produkten und den zeitlichen Verschiebungen spezifische Anforderungen an eine ökonomische Bewertung stellt, die mit einer solch simplen ROI Berechnung kaum zu beherrschen sind. Die im weiteren Verlauf vorgestellten Ansätze gehen daher weit über diesen einfachen Ansatz hinaus.

Als Vorgriff sei darauf hingewiesen, dass der in dieser Arbeit entwickelte Ansatz in einem ersten Analyseschritt einen einfachen Fragbogen zu einer ersten Go-/no Go Entscheidung benutzt, um früh eindeutig nicht geeignete Projekte zu identifizieren und den Analyseaufwand damit möglichst gering zu halten. Die tatsächlichen ROI Berechnungen basieren vollständig auf der kompletten FP Zählung der angestrebten Produktlinie, jeweils einmal mit und einmal ohne Wiederverwendung. Dies ermöglicht eine genaue Bestimmung des zu erwartenden Entwicklungsaufwands. Eine simple ROI Berechnung ohne diese verlässliche Berechnungsbasis, sprich ein Verständnis der zu erwartenden Komplexität der SWPL, erfolgt dennoch im Anschluss an den Fragebogen als optionaler Schritt. Unternehmen können dadurch eine erste grobe Idee von der Bedeutung der Wiederverwendungshäufigkeit und der grundsätzlichen Rahmenbedingung für ihre geplante SWPL erhalten.

3.2. COPLIMO und COCOMO II

COCOMO steht für COnstructive COst MOdel und wurde bereits 1981 in einer ersten Version von [Boeh81] vorgestellt. Ziel war es, ein umfassendes Modell zur ökonomischen Bewertung von Software Projekten zu erstellen, das Unternehmen unterstützt, diese Projekte im beabsichtigten Zeit- und Budgetrahmen fertig zu stellen. Es handelt sich um ein algorithmisches Kostenmodell, das einen Zusammenhang zwischen Software-Metriken und den zu erwartenden Kosten eines Projekts herstellt. Dadurch wird ebenso eine Abschätzung der zu erwartenden Projektdauer ermöglicht. Die in COCOMO verwendete Metrik ist Delivered Source Instructions (DSI). DSI zählt die manuell geschriebenen Computeranweisungen, die schließlich mit dem fertigen Produkt an den Kunden geliefert werden. Es werden demnach weder Testcode oder Hilfsprogramme noch automatisch generierter Code betrachtet. Die Bezeichnung Source Instructions geht auf die Anfang der 1980er Jahre noch üblichen Lochkarten zurück, weshalb Boehm diese auch als Card Images definiert. Die Aufwandsschätzung beginnt mit der Anwendung der Software Metrik, sprich der Erfassung der KDSI (Tausend DSI). Anschließend folgt eine Komplexitätsbestimmung des zu untersuchenden Projektes. Dafür stehen drei Klassen zur Auswahl: Organic Mode bezeichnet kleine und einfach zu handhabende Projekte, bei denen die Projektmitglieder eine

umfassende Erfahrung besitzen. Semi-detached fasst die Projekte mit mittlerer Größe (50 bis 300 KDSI) zusammen, die eine mittlere Komplexität aufweisen. Das Projektteam setzt sich aus Personen mit mittlerer Erfahrung und einigen Unerfahrenen zusammen. Die höchste Komplexitätsstufe beschreibt Boehm als Embedded. Hier finden sich insbesondere sicherheitskritische Projekte, die eine starre und unflexible Struktur aufweisen. Ebenso sind diese Projekte meist Neuentwicklungen. Die Aufwandsabschätzung erfolgt schließlich über eine Abschätzung der benötigten Personenmonate (PM) mittels folgender Formel:

Gleichung 4

$$PM = m * KDSI^{n}$$

Die Parameter m und n werden über die Komplexitätsklasse bestimmt und sind fest vorgegeben. Um die zu erwartende Projektdauer (TDEV = time to develop) zu bestimmen nutzt COCOMO die folgende Formel:

Gleichung 5[63]

$$TDEV = 2{,}5 * PM^{x}$$

Dies stellt die einfachste COCOMO Variante dar (Basic COCOMO) und ist trotz der begrenzten verfügbaren Information bereits in der Design-Phase eines SW Projektes anwendbar. Daneben gibt es noch zwei weitere Varianten (intermediate und detailed), die einsetzbar sind, sobald eine Entscheidung hinsichtlich der SW Architektur getroffen worden ist. Basis COCOMO wird dadurch um zahlreiche weitere Kostentreiber erweitert, wie z.B. die Enge des Zeitplans, die Qualität des Teams oder die geforderte Zuverlässigkeit des Systems. In ihrer gemeinsamen Wirkung haben diese weiteren Einflussfaktoren eine erhebliche Bedeutung. Die Bandbreite der möglichen Veränderung kann bis zu einem Faktor 6 liegen. Dies spiegelt sich auch entsprechend in den Beobachtungen von Boehm hinsichtlich der Ungenauigkeit bei der Aufwandsabschätzung in den verschiedenen Projektphasen wider (siehe Abbildung 18).

---

63 Der Parameter x wird erneut durch die Komplexitätsklasse fest vorgegeben. Man kann leicht erkennen, dass Boehm die Teamgröße unberücksichtigt lässt. Er ist der Auffassung, dass es eine optimale Teamgröße gibt, die nicht über- oder unterschritten werden sollte.

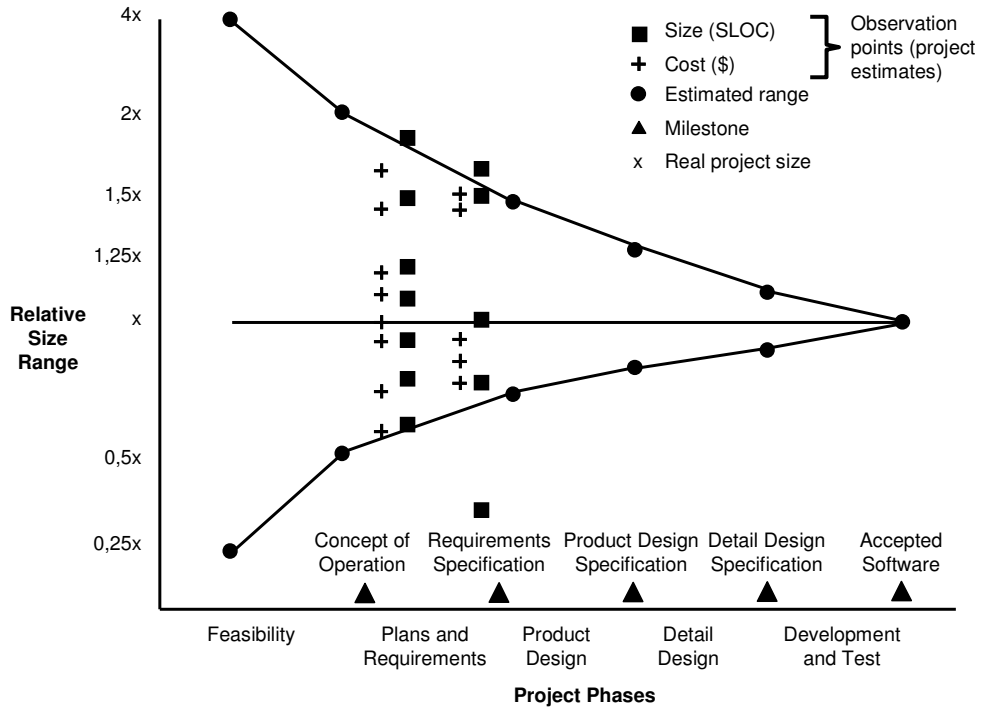

Abbildung 18 – Vergleich der Genauigkeit der Größen- und Kostenschätzung je Projektphase [Boehm00, S. 10, leicht angepasst]

Bereits bei der Veröffentlichung des ursprünglichen Modells wies Boehm auf die Schwächen seines Ansatzes hin. "Basic COCOMO is good for rough order of magnitude estimates of software costs, but its accuracy is necessarily limited because of its lack of factors to account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques, and other project attributes known to have significant influence on costs" [Boeh81]. COCOMO 81 wurde entwickelt für Projekte, die nach dem klassischen "Waterfall" Entwicklungsprozess aufgebaut sind. Moderne Software Engineering Methoden werden nicht unterstützt. Dennoch ist ein Großteil des ursprünglichen COCOMO Ansatzes nach Ansicht von Boehm heute immer noch aktuell. 1997 wurde den neuen Entwicklungen Rechnung getragen und eine Erweiterung und Überarbeitung fertig gestellt, die ca. 30% des ursprünglichen Inhalts betrifft [Boeh00, Preface]. Im Jahr 2000 wurde COCOMO II durch die Veröffentlichung von [Boeh00] einem größeren Publikum bekannt. Die Veränderungen betreffen in erster Linie die Variabilisierung des Modells, um es an die Bedingungen der neuen nicht-linearen Software-Entwicklungsansätze, wie beim Spiralmodell, und an kollaborative Software-Entwicklungsansätze anpassen zu können. Für die Aufwandsbewertung und –Abschätzung können sowohl KSLOC (Thousand Source Lines of Code, die neue Bezeichnung für DSI) als auch Function Points in der IFPUG Version

verwendet werden. FPs können allerdings nicht direkt verwendet werden, sondern müssen zunächst in KSLOC umgerechnet werden, da das Modell allein mit diesem Wert arbeitet. COCOMO II greift dazu auf bekannte Umrechnungstabellen zurück [Boeh00, S. 19].

Erneut gibt es mehrere Versionen der Methodik, die in unterschiedlichen Entwicklungsphasen des Projektes entsprechend der vorhandenen Informationen angewendet werden können: Early Design als ungefähre Abschätzung zu einem frühen Projektstadium und Post-Architecture als Detailmodell, das angewendet werden kann, sobald die SW-Architektur feststeht und die Entwicklung beginnen kann. Für beide Modelle gilt die identische Basisformel:

Gleichung 6

$$PM = A \times Size^{E} \times \prod_{i=1}^{n} EM_i$$

wobei:

A = eine Produktivitätskonstante, die für jedes Unternehmen zu Beginn individuell zu ermitteln ist

Size = die Größe der zu bewertenden Anwendung

E = das Ergebnis aus der Aggregation von 5 scale factors, die economies und deseconomies of scale für SW Projekte unterschiedlicher Größe berücksichtigen

EM = Effort multipliers, die unterschiedliche Aspekte des zu bewertenden Projektes berücksichtigen, um den nominalen Aufwand anzupassen

Unterschiede zwischen den beiden verschiedenen Komplexitätsgraden des Modells bildet die Formel allein über die individuelle Anzahl der Effort Multipliers (EM) ab. Während beim Post-Architecture 16 verschiedene Faktoren berücksichtigt werden, wurden diese für den Early Design Ansatz auf 6 zusammengefasst. Dabei wurde natürlich versucht, den Einfluss auf die PMs identisch abzubilden [Boeh00, S. 51]. Boehm unterteilt die Multiplikatoren in 4 Kategorien:

- **Produktbezogene Faktoren:** Die Charakteristik des zu entwickelnden Produktes hat maßgeblichen Einfluss auf den damit verbundenen Aufwand. Die wichtigste Größe ist

hierbei die Komplexität des Produktes. Zusätzlich werden Zuverlässigkeits-anforderungen und Wiederverwendungsgrad der Software berücksichtigt.

- **Plattformbezogene Faktoren:** Als Plattform bezeichnet Boehm die individuelle Kombination aus Hardware und Infrastruktursoftware, also die Umgebung, in der die zu entwickelnde Software laufen soll. Die Faktoren beinhalten die geplante Auslastung der Hardware, Einschränkungen bei der Speichernutzung und die Volatilität der Plattform. Weitere Faktoren, wie bspw. Parallelisierung, Eingebettete Systeme und Real-Time Anwendungen wurden nicht explizit berücksichtigt, sondern lediglich bei den produktbezogenen Faktoren unter der allgemeinen Produktkomplexität zusammengefasst.
- **Personenbezogene Faktoren:** Boehm schreibt diesen Faktoren neben der reinen Produktgröße den wichtigsten Einfluss auf den Projektaufwand zu. Sie bezeichnen dabei die Fähigkeiten und Erfahrungen des Entwicklungsteams. U.a. wird auch die Fluktuation des Mitarbeiterstabs berücksichtigt.
- **Projektbezogene Faktoren:** Schließlich werden alle Faktoren zusammengefasst, die direkt die Projektstruktur betreffen. Dies sind die Nutzung von Software-Werkzeugen, die geografische Verteilung der Entwicklungsabteilung und die geplante Projektdauer im Verhältnis zur durchschnittlich notwendigen Zeit.

Für eine detaillierte Beschreibung aller EMs sei auf [Boeh00, S. 36ff.] verwiesen.

Die Konstante A in der Formel bezeichnet eine Produktivitätskonstante, die für jedes Unternehmen individuell zu Beginn ermittelt werden sollte. Der Wert gibt an, wie viele PM durchschnittlich notwendig sind, um 1.000 SLOC zu erstellen. Basierend auf der COCOMO II Datenbank schätzt Boehm einen Standardwert für A von 2,94. Dieser Wert sollte jedoch für jedes Unternehmen mittels eigener Erfahrungswerte individuell bestimmt werden, um die Genauigkeit des Modells zu erhöhen. Size selbst gibt die Größe der zu erstellenden Software in KSLOC an. Dabei ist zu beachten, dass bei einer Überarbeitung eines bestehenden Produktes lediglich neue KSLOC gezählt werden. Daher enthält COCOMO II eine Berechnungsvorschrift, um wiederverwendeten Code in äquivalenten neuen Code umzurechnen. Dabei werden zwei Wiederverwendungsarten unterschieden: Eine reine Wiederverwendung ohne Anpassung (black box, reused code) und die Anpassung von existierendem Code (white box, adapted code). Der angepasste Code wird als equivalent

source lines of code (ESLOC) bezeichnet. COCOMO II berücksichtigt hierbei u.a. die zusätzlichen Kosten, die bei der Wiederverwendung entstehen, um den existierenden Code zu verstehen und die Schnittstellen zu überprüfen. Zusammengefasst wird dies in einem nicht-linearen Wiederverwendungsmodell, das insbesondere die Effekte Verständnis der wiederzuverwendenden Software, Grad der notwendigen Anpassung und Vertrautheit der Entwickler mit der Software berücksichtigt. Eine detailliertere Erklärung erfolgt weiter unten bei der Beschreibung der COPLIMO Erweiterung.

Der Exponent E gilt unverändert für beide Modelle, Early Design und Post-Architecture. Der Wert spiegelt Economies of Scale (Wert <1) oder eben Diseconomies of Scale (Wert >1) eines Projektes wider. Demnach wird angegeben, ob bei einer Verdopplung der Projektgröße mit einer unter- oder überproportionalen Entwicklung des Aufwandes zu rechnen ist. Eine überproportionale Vergrößerung des Aufwandes könnte bspw. in einer ineffizienten Kommunikation der Teammitglieder und einem größeren Overhead des Projektteams begründet sein. Die folgende Formel beschreibt die Berechnung:

Gleichung 7 – Berechnung des Exponenten E

$$E = B + 0{,}01 \times \sum_{j=1}^{5} SF_j$$

Die Konstante B wird mit 0,91 basierend auf der COCOMO II zugrundeliegenden Projektdatenbank angegeben. Die fünf Scale Factors (SF) sind: Precedentedness, Development Flexibility, Architecture/Risk Resolution, Team Cohesion und Process Maturity (nach CMM). Die Faktoren können jeweils Werte zwischen 0 und 5 bzw. 0 und 7,8 (je nach SF) annehmen. Insgesamt ergibt sich dadurch ein großer Einfluss auf den erwarteten Gesamtaufwand. Die Variation zwischen größtem und kleinstem möglichen Wert liegt bei einem Faktor von über 4. Boehm relativiert diesen Einfluss jedoch, da die Veränderung eines Faktors um ein Level lediglich eine Veränderung von ca. 6% bewirkt [Boeh00, S. 31]. Der Wert E mit seinen 5 exponentiellen Skalierungsfaktoren ersetzt in COCOMO II das in COCOMO 81 verwendete 3-stufige Komplexitätsmodell.

Inzwischen sind zahlreiche Erweiterungen des COCOMO Modells erschienen. Für eine detaillierte Erklärung der einzelnen Modelle siehe [Cent09]. Aktuelle Arbeiten versuchen aus den einzelnen Elementen eine integrierte COCOMO Suite zu machen. Eine Übersicht über die vorhandenen Mitglieder der COCOMO Familie gibt die folgende Grafik:

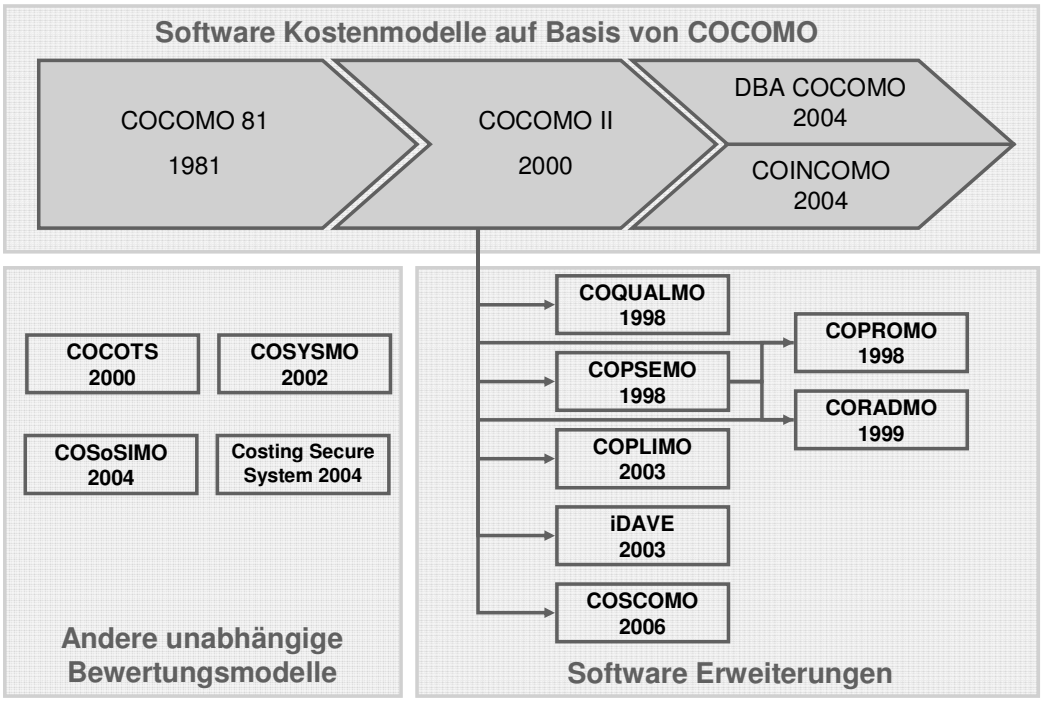

Abbildung 19 – Übersicht COCOMO Modell-Familie [Smit06, S. 4]

Hier soll im Detail auf die Erweiterung COPLIMO eingegangen werden. COPLIMO steht für COnstructive Product Line Investment MOdel und wurde 2004 vorgestellt [Boeh04]. Es basiert größtenteils direkt auf COCOMO II, hat allerdings einige Erweiterungen hinsichtlich Software-Wiederverwendung erfahren, um explizit für Software Produktlinien verwendbar zu sein. So war es notwendig, die Aufwandsberechnung in zwei Teilschritte zu zerlegen: die Erstellung der eigentlichen Produktlinie, inklusive Domain Assets und die annualisierte Betrachtung des Anwendungszeitraums der Produktlinie, inklusive Wartungs- und Anpassungsaufwänden. Die wichtigste Erweiterung ist die Hinzunahme von 2 zusätzlichen Einflussfaktoren bei der Wiederverwendung von Softwarebestandteilen:

- **Relative Cost of writing for Reuse (RCWR):** Die strategische Entwicklung wiederverwendbarer Software erzeugt einen höheren Aufwand verglichen mit der Entwicklung einer reinen Einzelanwendung. RCWR gibt diesen Wert als einen Multiplikator wider. Ein Wert von 1,5 bedeutet, dass eine wiederverwendbare Software den 1,5-fachen Aufwand zur Erstellung erzeugt wie eine Einzelanwendung. COPLIMO greift auf bereits vorhandene Variablen aus COCOMO II zur Berechnung von RCWR zurück. Dies sind die EM Development for Reuse (RUSE), Required Software Reliability (RELY) und Documentation Match to Life-Cycle Needs (DOCU). Diese drei Multiplikatoren waren bereits in COCOMO II miteinander

verknüpft. So impliziert ein RUSE Wert die Höhe der beiden anderen Werte [Boeh00, S. 42]. Allgemein wird für RUSE ein Wert von 15-20% angenommen. Dieser Wert liegt im Vergleich zu anderen Ansätzen sehr niedrig. Dies ist dadurch begründet, dass weitere Effekte (z.B. die beiden anderen Multiplikatoren) herausgerechnet sind und erst anschließend in der Formel multipliziert werden. Die Formel lautet:

Gleichung 8

$$RCWR = (1 + RUSE) * RELY * DOCU$$

- **Relative Cost of Reuse (RCR):** Bei dieser Berechnung wird ebenfalls auf bestehende Ergebnisse von COCOMO II zurückgegriffen. Wie bereits oben beschrieben enthält COCOMO II ein einfaches Wiederverwendungsmodell. Bei einer Black Box Wiederverwendung, also der reinen Wiederverwendung von existierendem Code ohne Anpassung, wird der Aufwand für das Suchen und Finden passender Software mittels eines Degree of Assessment and Assimilation (AA) beschrieben. Dieser Wert liegt zwischen 0,02 (wenn außer der Suche kein weiterer Aufwand entsteht) und 0,08 (dies schließt komplettes Testen und Evaluation mit ein). Bei einer White Box Wiederverwendung, wobei eine Anpassung der Software erfolgt, ergibt sich eine umfangreichere Berechnung:

Gleichung 9 – Berechnung der equivalenten SLOC

$$Equivalent\ KSLOC = KSLOC \times \left(1 - \frac{AT}{100}\right) \times AAM$$

AT zeigt den Anteil an automatisch generiertem Code. Dieser wird komplett abgezogen. AAM bezeichnet den Adaptation Adjustment Modifier. Zur Berechnung dieses Wertes werden das Verständnis der wiederverwendeten Software (Software Understanding, SU), die Vertrautheit der Programmierer (Programmer Unfamiliarity, UNFM) und ein gewichteter Wert für den Grad der Veränderung (Adaptation Adjustment Factor, AAF) herangezogen. AAF errechnet sich aus der prozentualen Veränderung des Designs (Percent Design Modified, DM), der prozentualen Codeveränderung (Percent Code Modified, CM) und dem Integrationsgrad der veränderten Software (Percent of Integration Required for Adapted Software, IM). Nimmt man nun an, dass bei einer existierenden Software ca. 25% angepasst werden und dass die anderen Variablen (mit Ausnahme der UNFM) eher im hohen Bereich

liegen, also die Software bekannt ist, dann ergibt sich ein AAM Wert von ca. 0,3. Das bedeutet, dass eine Wiederverwendung etwa 30% der Kosten einer kompletten Neuentwicklung verursachen würde.

Die eigentliche neue Funktion in COPLIMO ist die Betrachtung des gesamten Lebenszyklusses einer Produktlinie. Aufgrund der einheitlichen, formelmäßigen Berechnung ergeben sich dabei allerdings einige Einschränkungen, da zahlreiche Variablen über den gesamten Zeitraum konstant bleiben müssen. Dies betrifft bspw. alle EM und den Anteil der Software, der sich jährlich verändert. Die Aufwandsberechnung erfolgt recht einfach, indem zu dem ursprünglichen Aufwand für die Erstellung der Produktlinie die jährlichen Wartungsaufwände über die gesamte Laufzeit addiert werden. Die Berechnung der Aufwände greift erneut auf die grundlegenden Ideen von COCOMO II zurück und verwendet Aufwandsmultiplikatoren, die Systemverständnis und die Unkenntnis der Programmierer mit der Software berücksichtigen.

Abschließend kann festgehalten werden, dass COCOMO mit all seinen Erweiterungen einen nützlichen Ansatz bietet. Insbesondere die verschiedenen Komplexitätsstufen ermöglichen eine, der individuellen Datenlage angepasste, Genauigkeit des Modells. Allerdings erfordert der Ansatz auch eine hohe Einarbeitungszeit und das vollständige Modell setzt eine sehr umfangreiche Datenbasis voraus. Die angesprochene Kalibrierung der grundlegenden Faktoren, wie z.B. die individuelle Produktivität, ist notwendig und lässt sich nicht eliminieren. Die starre Verwendung von einmal gewählten Variablen erscheint bei der Berechnung des Aufwands für ein Projekt eher unproblematisch. Bei der Anwendung auf eine Produktlinie mit mehrjähriger Laufzeit ist dies jedoch als kritisch anzusehen. Ebenso basieren die empfohlenen Standardwerte auf der Datensammlung von COCOMO. Bei einer Anwendung auf Embedded Systems ist fraglich, ob diese einfach weiterverwendet werden können. Eine empirische Anpassung dürfte sich aufgrund der geringen verfügbaren Basisdaten allerdings als schwierig bis unmöglich erweisen.

3.3. Poulin's Measuring Software Reuse

Bereits 1997 stellte Poulin [Poul97a, Poul97b] einen zu COPLIMO sehr ähnlichen Ansatz zur ökonomischen Bewertung von Software Reuse vor. Er führt die beiden Begriffe Relative Cost of Reuse (RCR) und Relative Cost of Writing for Reuse (RCWR) ein. Diese entsprechen den gleichnamigen Begriffen bei COPLIMO. Die weiteren Berechnungsschritte unterscheiden

sich jedoch. So verwendet Poulin RCWR und RCR zur Berechnung von zwei weiteren Variablen:

- **Reuse Cost Avoidance (RCA):** Dies bezeichnet die Kosteneinsparung durch die Wiederverwendung von Software. Die Einsparung setzt sich aus zwei Teilen zusammen: Einsparungen bei der Entwicklung (Development Cost Avoidance, DCA) und Einsparungen bei der Wartung (Service Cost Avoidance, SCA). Die SCA ergeben sich aus der Anzahl der wiederverwendeten Codezeilen multipliziert mit der Fehlerrate und den Fehlerkosten.

Gleichung 10 – Formel zur Berechnung von DCA

$$DCA = RSI * (1 - RCR) * \left(\frac{C_{SU}}{LOC} \right)$$

wobei RSI die Reused Source Instructions und C_{SU} die Cost of single use code bezeichnet

- **Additional Development Cost (ADC):** Dies bezeichnet die bei einer von Anfang an geplanten Wiederverwendung zusätzlich entstehenden Entwicklungskosten. Die Berechnung erfolgt mittels des RCWR Wertes und des einmalig zu erstellenden Codes. Notwendig ist hier erneut die Kenntnis der durchschnittlichen Entwicklungskosten je Codezeile.

Anschließend berechnet Poulin einen einfachen ROI indem die RCAs über alle Wiederverwendungen hinweg addiert werden und einmalig die Zusatzkosten bei der Entwicklung für eine Wiederverwendung abgezogen werden.

Poulins Ansatz bietet sinnvolle Ideen, die z.B. bei COPLIMO aufgenommen und verfeinert wurden. Allerdings fehlt hier die explizite Berücksichtigung einer SWPL. Zusätzlich ergeben sich aufgrund der starken Ähnlichkeiten und der algorithmischen Herangehensweise die gleichen Schwächen, wie schon bei COCOMO beschrieben. Insbesondere die notwendigen Daten, wie Fehlerrate, Fehlerkosten und Entwicklungskosten pro Codezeile stellen eine Schwierigkeit in der Berechnung dar.

3.4. PuLSE-Eco Ansatz

Der PuLSE Ansatz (Product Line Software Engineering) wurde vom Frauenhofer Institut für experimentelles Software Engineering entwickelt und bietet eine Werkzeugunterstützung für

den gesamten Entwicklungsprozess einer SWPL an [Schm05, S. 544]. PuLSE-Eco [Schm03], als ein Teil dieser Werkzeuge, unterstützt das Scoping der Produktlinie durch eine ökonomische, funktionsorientierte Bewertung der SWPL-Bestandteile. Der Ansatz besteht aus 3 Schritten, die in Abbildung 20 wiedergegeben sind.

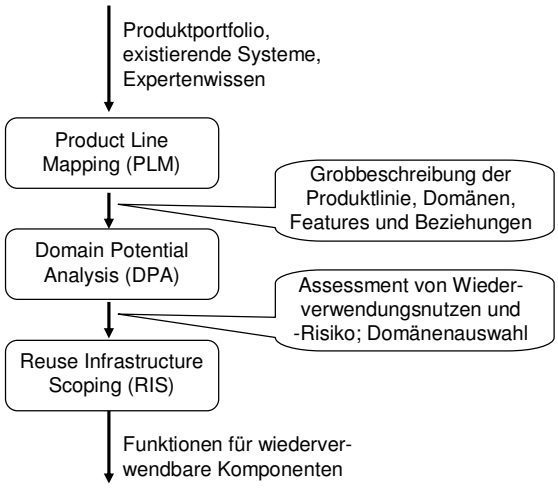

Abbildung 20 – Überblick über die Grundstruktur des PuLSE-Eco Ansatzes [Schm04, S. 47]

Nachfolgend werden die 3 Schritte beschrieben:

- **Produktlinienerfassung:** In einem ersten Schritt werden mit möglichst geringem Aufwand alle notwendigen Informationen über die geplante SWPL zusammengefasst. Dies geschieht zunächst nach Produkten getrennt, d.h. alle Produkte werden hinsichtlich ihrer Funktionalitäten und Unterschiede zu anderen Produkten systematisch erfasst. Wichtig ist hierbei, dass im Gegensatz zur klassischen Function Point Methodik alle Funktionalitäten erfasst werden, auch wenn diese für den Nutzer nicht direkt sichtbar sind. Eine Zusammenfassung der einzelnen Produkte erfolgt erst zu einem späteren Zeitpunkt. Dadurch wird ein inkrementelles Vorgehen ermöglicht. Mehrere Produkte können zeitgleich in kleineren Workshops erfasst werden, was zu einer Verringerung der Expertenbelastung führt. Zusätzlich erfolgt die Produkterfassung funktionsorientiert, was eine einheitliche Beschreibung über den gesamten Scoping Prozess erlaubt. Die Definition von mehrfach verwendeten Teilfunktionen erleichtert die Vermeidung von Doppelzählungen. In einem zweiten Schritt werden die für mehrere oder sogar alle Produkte wichtigen Funktionsbereiche identifiziert. PuLSE-Eco unterstützt diesen Schritt durch ein Erfassungsschema, das nicht nur die wesentlichen Funktionalitäten, sondern insbesondere auch die

Wechselwirkungen der Bereiche berücksichtigt. Als Ergebnis entsteht eine initiale Produktkarte, die alle wesentlichen Charakteristika der Produktlinie enthält. Tabelle 4 zeigt eine Produktkarte am Beispiel der Modellpalette von BMW. Ein x bedeutet jeweils, dass die Kombination aus Produkt und Funktion zu berücksichtigen ist.

			Existiert				In Entwicklung			Hypothetisch	
			3xx	5xx	7xx	…	550	770	…	200	…
Komfort	Kommunikation	Handy	x	x	x		x	x		x	
		Internet						x			
	Radiosystem	Navigation	x	x	x		x	x			
		…									
		Professional		x	x			x			
Sicherheit	Park-Distance-Control	vorne		x	x		x	x			
		hinten	x	x	x		x	x		x	
…	…	…									
Fahrwerk	Niveauregulierung							x	x		x
	…										

Tabelle 4 – Beispielhafte Produktkarte [Schm04, S. 48]

- **Domänenspezifische Potenzialanalyse**: Aufbauend auf den Produktbeschreibungen erfolgt nun eine Risikobewertung. Klassische Assessment Ansätze wie CMMI und SPICE bieten eine solche Risikobewertung, eignen sich allerdings nicht für eine direkte Anwendung bei produktlinienorientierter Wiederverwendung. Daher wurde im Rahmen von PuLSE-Eco ein eigenständiger Assessmentansatz entwickelt, der sich in zwei Hauptdimensionen gliedert: Machbarkeit und Nutzen. Insgesamt 93 Teilfragen bilden die Basis der Erhebung. Ein wichtiger Unterschied zu anderen Assessment-Ansätzen ergibt sich durch die funktionsorientierte Betrachtungsweise. Dadurch wird es möglich, das Wiederverwendungspotenzial individuell auf Funktionsebene zu bestimmen. Der Mehraufwand relativiert sich durch die deutliche Verbesserung der Ergebnisse, da sich die Funktionsbereiche einer Produktlinie hinsichtlich ihres Wiederverwendungspotenzials häufig deutlich unterscheiden.

- **Komponentendefinition:** Die Optimierung des Wiederverwendungspotenzials ist das Kernelement des Ansatzes. Auf Basis der identifizierten Features erfolgt eine Analyse der Eignung zur Wiederverwendung. Dazu werden zunächst die grundlegenden

Unternehmensziele, insbesondere mit Hinblick auf die Realisierung der Produktlinie, erfasst und anschließend durch einzelne Metriken, die direkt von der Organisation beeinflussbar sind, operationalisiert. Für jedes Geschäftsziel lassen sich dann die individuellen Wiederverwendungsnutzen berechnen. Eine Schwierigkeit stellt die notwendige Integration der Ergebnisse je Geschäftsziel dar. Optimal wäre eine Bewertung mittels des sich ergebenden Unternehmenswerts. Dieser Wert ist jedoch schwer zu bestimmen. Im Rahmen von PuLSE-Eco hat sich gezeigt, dass alternativ gute Ergebnisse über eine Gewichtung entsprechend der relativen Wichtigkeit der Ziele erreicht werden können. So könnte man bspw. annehmen, "eine Personenwoche weniger Entwickleraufwand hat den gleichen Wert wie einen Monat längere Time-to-Market" [Schm03, S. 13]. So erzielt man eine angemessene Approximation der Trade-offs der Unternehmensziele. Abschließend erfolgt basierend auf den berechneten Ergebnissen eine Einstufung in eine von 6 Wiederverwendungskategorien.

Der PuLSE-Eco Ansatz unterscheidet sich von anderen Scoping Ansätzen für SWPL einerseits durch seine Featureorientierung und andererseits durch die Berücksichtigung von unterschiedlichen Unternehmenszielen. Leider stellt die ökonomische Analyse nicht den Schwerpunkt des Ansatzes da. Das Scoping im Rahmen einer erfolgreichen SWPL Einführung steht im Vordergrund. Daher liefert dieser Ansatz einige sehr interessante Ideen, die im Rahmen dieser Arbeit weiter verfolgt werden. So findet zu Beginn der Analyse des e-FACTS Modells eine Bewertung der Ausgangssituation und des Potenzials basierend auf einem speziell für SWPL angepassten Fragebogen statt. Ebenso wurde das Vorgehen zur Berücksichtigung verschiedener Unternehmensziele übernommen. Und schließlich stellt die hier vorgestellte Funktionsorientierung den Grundpfeiler des Modells dieser Arbeit dar. Die eigentliche ökonomische Analyse wird jedoch über eine FP Zählung realisiert.

3.5. ABC Analyse

Cohen beschrieb in [Cohe03] einen einfachen Ansatz zur Berechnung eines Business Cases für den Produktfamilien Einsatz. Er unterscheidet drei Bereiche:

- **Application:** die Produkte, die im Rahmen der Produktfamilie entstehen sollen
- **Benefits:** die Vorteile, die sich das Unternehmen von diesem Ansatz verspricht, tangible und intangible

- **Costs:** die Kosten, die mit der Erstellung, Wartung und Verwendung der Produktfamilie verbunden sind

Als relevante Größen berücksichtigt er die Degree of Reuse (DOR), also den Umfang den die Domain Assets der Produktfamilie an allen zu entwickelnden Produkten haben, und Cost of Reuse (COR). Bei letzterem handelt es sich nicht um die Entwicklungskosten der Domain Assets, sondern ausschließlich um die Kosten, die mit der direkten Wiederverwendung der Komponenten verbunden sind (z.B. Testen, individuelle Requirements, Bestimmung der passenden Assets, Training der Entwickler). Und schließlich berücksichtigt er ebenfalls die Refresh Rate. Diese Größe gibt die Veränderungsgeschwindigkeit des Marktes an, also den Umfang an neuen Funktionen, die die nächste Generation aufweisen muss, oder anders gesagt, den Teil der Domain Assets, die erweitert oder angepasst werden müssen.

Für einen ersten sehr groben Überblick eignet sich dieses Verfahren sicherlich. Allerdings stellt es hohe Anforderungen an die Nutzer hinsichtlich ihrer Erfahrungen mit Produktfamilien. So wird es einem Techniker, der gerade seine erste Produktfamilie plant, kaum möglich sein, die einzelnen Werte auch nur ungefähr zu schätzen. Dieser Ansatz eignet sich daher besser für ein Beratungsprojekt als für eine eigene individuelle erste Einschätzung.

3.6. SIMPLE

Das Structured Intuitive Model for Product Line Economics (SIMPLE) ist ein ökonomisches Modell, das speziell für die Bewertung von SWPL entwickelt worden ist. Vorgestellt wurde der Ansatz 2005 von Clements, McGregor und Cohen in [Clem05]. Der Fokus lag auf einer strukturierten und dennoch intuitiv zu verstehenden und anzuwendenden Methodik. Dazu wurde ein "devide and conquer" Ansatz [Clem05, S. 6] gewählt, also die Zerlegung der Berechnung in handhabbare Teile und die spätere Addition der Teilergebnisse. Allgemein wurde SIMPLE für Unternehmen entwickelt, die die Frage beantworten wollen, ob es ökonomisch sinnvoll ist, für eine Menge neuer Produkte eine SWPL zu bauen, eine existerende SWPL zu erweitern oder die entsprechenden Produkte als Einzellösung zu realisieren. Abbildung 21 verdeutlicht dieses Grundverständnis.

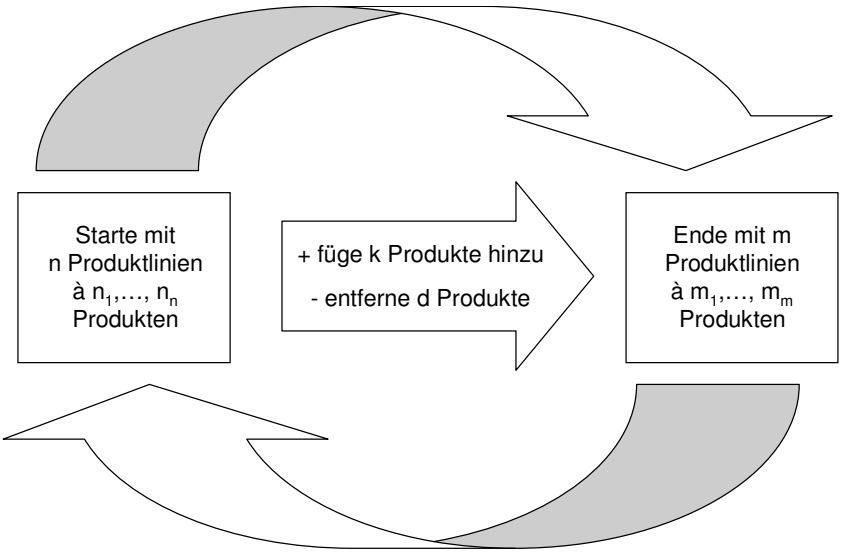

Abbildung 21 – Grundsätzliches Szenario für SIMPLE Kalkulationen [vgl. Clem05, S. 5]

Zur Zerlegung des Problems schlägt SIMPLE 4 Basic Cost Functions vor und eine nicht definierte Anzahl von Benefit Functions:

- $C_{org}()$ bezeichnet die Kosten, die anfallen, um eine Organisation so zu gestalten, dass sie fit für die Anwendung einer SWPL ist. Dies beinhaltet Kosten für Reorganisation, Prozessverbesserung, Training, usw.

- $C_{cab}()$ ermittelt die Kosten für die Entwicklung der Core Asset Base. Dies umfasst die Kosten für die Communality/Variability Analysis, also die Bewertung der einzelnen Funktionen hinsichtlich ihres Wiederverwendungspotenzials, die Definition des Umfangs der SWPL, das Design und die Evaluierung der SWPL Architecture und schließlich für die Erstellung einer Test-Architektur. Neben den reinen Kosten für die Erstellung von Software sind hier auch Kosten für die Erstellung von Plänen, Anweisungen und Dokumentationen berücksichtigt. C_{cab} kann sowohl für die Neugestaltung einer SWPL als auch für den Ersatz oder die Erweiterung einer bestehenden SWPL genutzt werden.

- $C_{unique}()$ enthält die Kosten für die Entwicklung der individuellen Teile eines Produktes, die nicht Teil der Core Asset Base sind. Im Extremfall kann dies 100% der Kosten abdecken, wenn ein individuelles Produkt komplett außerhalb der SWPL entwickelt wird.

- $C_{reuse}()$ ermittelt die Kosten für die Wiederverwendung eines Core Assets. Darin enthalten sind die Kosten für die Suche des Assets, der individuellen Anpassung und eines zusätzlichen Integrationstestings innerhalb des neuen Produktes.

Diese Basisfunktionen werden erweitert um die folgenden drei Kostenfunktionen:

- $C_{prod}()$ kalkuliert die Kosten, die anfallen würden, wenn die Software komplett eigenständig entwickelt werden würde. SIMPLE schlägt vor, für die eigentliche Kostenermittlung auf Erfahrungswerte oder etablierte Kostenkalkulationsmethoden wie z.B. COCOMO II zurückzugreifen.

- $C_{evo}()$ misst die Kosten, die die Weiterentwicklung einer bereits existierenden Software verursachen würde, die für ein neues Produkt angepasst werden muss. Dabei wird explizit davon ausgegangen, dass dies ohne Verwendung einer SWPL geschieht.

- $C_{cabu}()$ ermittelt die Kosten für die regelmäßige Anpassung der Elemente der Core Asset Base, wenn sie für ein neues Produkt tatsächlich wiederverwendet werden sollen. Anpassungen können sich ergeben, wenn ein neues Produkt Veränderungen an dem Asset erfordert hat, Fehler entdeckt worden sind oder zusätzliche Funktionen in der neuen Version als Commonality behandelt werden sollen. Sollte ein Asset ohne Veränderung verwendbar sein, wären die C_{cabu} gleich null.

SIMPLE berücksichtigt explizit die Zeit als Variable, denn nur so lassen sich sowohl evolutionär als auch revolutionär eingeführte SWPL abdecken. Dies betrifft insbesondere die Funktionen C_{cab} , C_{cabu} , C_{org} . Zusätzlich ist es möglich Kapitalkosten zu integrieren, indem über den Zeitablauf mit einem Kalkulationszinssatz zur Abzinsung gerechnet wird.

Ein wesentliches Unterscheidungsmerkmal gegenüber zahlreichen anderen ökonomischen Modellen ist die Einführung von Benefit Functions. Diese werden zur Berücksichtigung weiterer Vorteile aus der Einführung einer SWPL genutzt. Wichtig ist dabei, dass die gemessenen Effekte nicht doppelt gezählt werden, wie z.B. bei einer schnelleren Marktreife von Produkten durch kürzere Entwicklungszeiten, sprich niedrigere Entwicklungskosten. Die Anzahl der Benefit Functions ist nicht begrenzt, alle relevanten Aspekte können abgedeckt werden. Als Beispiele nennt SIMPLE u.a. bessere Qualität, einen höheren Marktanteil, gesteigerte Kundenzufriedenheit und höhere Produktivität. Gleichung 10 zeigt die vollständige SIMPLE Formel:

Gleichung 11

$$C_{org}(\ )+C_{cab}(\ )+\sum_{i=1}^{n}\left(C_{unique}(product_i)+C_{reuse}(product_i)\right)+\sum_{j=1}^{nbrBenefits}B_{ben_j}(t)$$

Diese Gleichung ermittelt zunächst einmal nur die Kosten und Benefits (B_{ben}) für die Realisierung einer SWPL, wobei n für die einzelnen auf Basis der SWPL realisierten Produkte steht. Um weitere Fragen beantworten zu können, wie z.B. lohnt sich die Einführung einer SWPL für ein Unternehmen, das x Produkte in den nächsten y Jahren entwickeln möchte, muss die Basisgleichung erweitert werden. SIMPLE bietet dafür insgesamt 9 Szenarios, die fast alle relevanten Anwendungsfälle abdecken sollten. Für Details siehe [Clem05, S. 13ff.]. Jedes Szenario bietet ein eigenes ökonomisches Modell basierend auf den vorgestellten Grundprinzipien.

Zu beachten ist, dass SIMPLE keine spezifischen Umsetzungen der Funktionen anbietet, sondern lediglich die grundsätzliche Bedeutung spezifiziert. Es obliegt dem Nutzer, diese selbst auf seine individuellen Anforderungen anzupassen. Dies ist als eine Schwäche des Modells anzusehen. SIMPLE hält zwar sein Versprechen intuitiv und verständlich zu sein. Wenn es aber an die reale Umsetzung geht, wird der Nutzer weitgehend allein gelassen. Der Ansatz basiert darauf, dass das Unternehmen die notwendigen Daten bereitstellen kann. Als mögliche Quellen werden genannt:

- Eigene historische Daten
- Benchmarks
- Nutzungsfunktionen, also die Aufteilung bekannter Fixkosten über die Nutzung
- Ableitung aus bekannten Daten
- Weitere eigene Zerlegung der genannten Funktionen[64]
- Einführung eines Homogenitätsmaßes, das wiedergibt wie ähnlich sich alle Funktionen sind.

[64] In [Pete04] wird vorgeschlagen, die C_{cab} weiter zu zerlegen in Domain Analysis, Target Architecture, Technology Standards (Einigung über zukünftig zu verwendende Technologien und Standards), Current Architecture (Erlangung eines Verständnisses der aktuellen Architektur) und Migration Strategy/Plan.

SIMPLE bietet ein gutes Grundkonzept, das auch in der vorliegenden Arbeit seine Verwendung findet (siehe Kapitel 4.2). Zur Ermittlung der notwendigen Daten wird an dieser Stelle auf die schon bekannte Feature Point Methodik zurückgegriffen. Damit kann der Gedanke der einfachen Verwendung auch auf die reale Umsetzung erweitert werden.

3.7. Simulation – das ReValue Modell

Aufgrund der Tatsache, dass für den Bereich der Wiederverwendung von Software nur eine geringe Anzahl an Beobachtungspunkten vorliegt, erscheint ein Ansatz zur Generierung von möglichst realistischen Punkten sinnvoll. Simulationen bieten eine solche Möglichkeit, um allgemeine Aussagen hinsichtlich der ökonomischen Vorteilhaftigkeit von wiederverwendbarer Software abzuleiten. Ein nützlicher Ansatz findet sich bei Stützle [Stüt02]. Ziel der Arbeit ist es, die Wiederverwendung von Software ökonomisch zu bewerten. Wiederverwendung wird dabei als Investition verstanden, d.h. ein Unternehmen entschließt sich, bereits vorhandene Software wiederverwendbar zu machen, oder neue Software durch einen zusätzlichen Arbeitsaufwand a priori wiederverwendbar zu gestalten. Das ReValue Modell berücksichtigt im Gegensatz zu den zahlreichen bereits vorhandenen Arbeiten die Unsicherheit von Investitionen. Über die Zeit werden mögliche Verläufe der Investition durchgespielt, sodass mittels eines einseitigen Konfidenzintervalls über die ermittelten Zufallsergebnisse der simulierten Investitionsverläufe eine entsprechend des gewählten Konfidenzintervalls sichere Mindestrendite der Investition ermittelt werden kann. Zur Simulation greift das Modell auf die Monte Carlo Simulation zurück. Der Ablauf der Simulation ist in Abbildung 22 veranschaulicht. Die ermittelte Zielgröße ist der interne Zinssatz, also die Verzinsung des eingesetzten Kapitals. Mit Hilfe dieses Modells sind allgemeine Aussagen über die Eignung bestimmter Softwareklassen zur Wiederverwendung von Softwarekomponenten möglich.

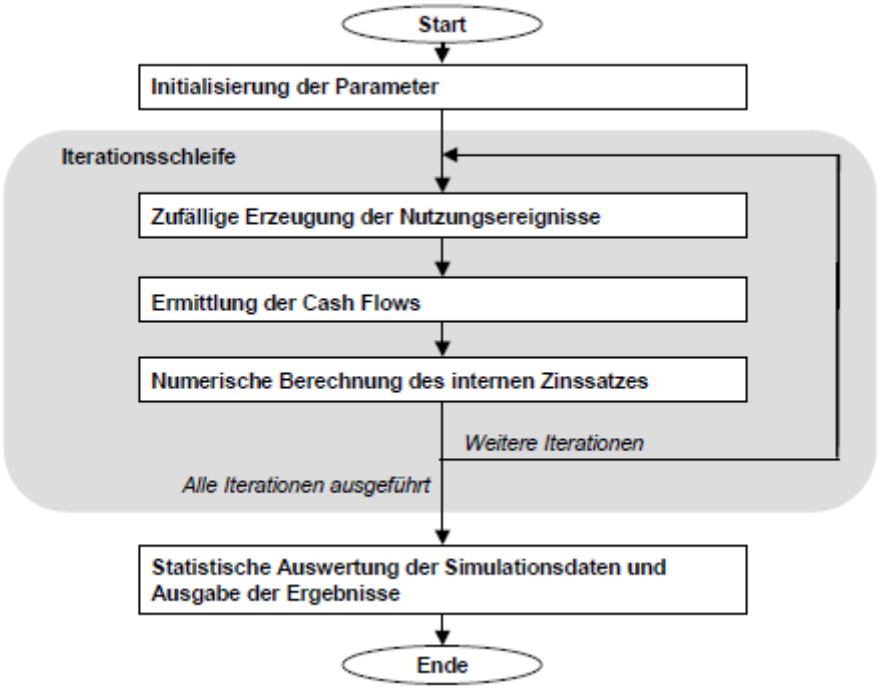

Abbildung 22 – Struktur des ReValue Simulationsprogramms [Stüt02, S. 151]

Das Modell benötigt folgende Eingangsdaten, mit denen die Simulation durchgeführt wird:

- **Investition in Wiederverwendbarkeit:** Die Erstellung wieder verwendbarer Software erzeugt einen Mehraufwand. Dieser wird als Faktor der ursprünglichen Entwicklungskosten als Einzelanwendung dargestellt.
- **Anfängliche Nutzungsrate:** Dieser Wert wird gemessen als Anzahl Nutzungen pro Jahr. Aufgrund einer angenommenen zeitlichen Abnutzung der Komponenten wird dieser Wert entsprechend dem physikalischen Zerfallsprozess simuliert.
- **Halbwertszeit:** Die Halbwertszeit gibt an, in welchem Zeitraum sich die Nutzungsrate halbiert haben wird.
- **Nutzungskostenfaktor:** Der Faktor gibt die zu erwartenden Kosten je Wiederverwendung als Prozentwert der ursprünglichen Entwicklungskosten an.

Das Modell bietet durch die Integration des Risikos in die Berechnung eindeutige Vorteile. Andererseits werden auch hier lediglich Standardwerte berücksichtigt. Insbesondere der wichtige Faktor für die Investition in Wiederverwendung ist zu grob. Im Hinblick auf die vorliegende Arbeit kann das ReValue Modell dennoch als nützlich angesehen werden, denn das Ergebnis dieser Arbeit sind genaue Daten hinsichtlich der Kosten für die

Wiederverwendung. ReValue kann dementsprechend genutzt werden, um Unsicherheiten hinsichtlich der zukünftigen Wiederverwendungsraten zu berücksichtigen und eine sichere Mindestrendite zu berechnen.

3.8. Abschließende Betrachtung und Anforderungen an ein neues Modell

Aufgrund der hohen Zunahme der Komplexität und auch der Bedeutung von Software hat sich bis heute ein großes Interesse an der Wiederverwendung von Software ergeben. Um die finanziellen Folgen einer strategischen Wiederverwendung von SW abzuschätzen, werden ökonomische Modelle benötigt. Die wissenschaftliche Literatur bietet dementsprechend eine breite Auswahl an sehr unterschiedlichen Modellen und Ansätzen. Im Kapitel 3 wurden die wichtigsten ökonomischen Modelle vorgestellt und bereits ihre Stärken und Schwächen angesprochen. Zusammenfassend kann festgehalten werden, dass "sizing methods for dealing with reusable artifacts are not yet perfected" [Jone07, S. 259]. Bevor im Kapitel 5 mit e-FACTS das eigentliche Modell dieser Arbeit eingeführt wird, soll noch auf die Einflüsse der bestehenden Modelle auf dieses neue Modell eingegangen werden.

Die wahrscheinlich größte Einschränkung der vorgestellten Modelle besteht in der Verwendung von Standardwerten. Es gibt zahlreiche Quellen für Werte zu den verschiedensten Themengebieten. Aber "the ranges of uncertainty are so great that users are well advised to replace generic 'industry averages' with specific values from their own enterprises for key parameters" [Jone07, S. xxiii]. So findet man in der Literatur bspw. für den Mehraufwand wiederverwendbarer Software Faktoren zwischen 1,5 und 9 [vgl. Broo95, S. 224, Jaco97, S. 23 und Trac95, S. xi]. Damit wird es möglich, allein über diesen Wert jede Anwendung positiv oder negativ zu beeinflussen. Bedenkt man weiterhin, dass "one of the most severe sources of software-estimation error centers around corporate politics, or the fact that senior executives and client executives have the authority to arbitrarily reject valid estimates" [Jone07, S. 72; siehe auch Teil A Kapitel 5], dann erkennt man die hohe Bedeutung einer stabilen und für das jeweilige Unternehmen gültigen Datenbasis.

Bei der Planung von SWPL stehen Unternehmen vor der Aufgabe, neben der Aufwandsschätzung für das Gesamtsystem auch eine Entscheidung hinsichtlich der Gestaltung der Asset Base zu treffen. Nur für wenige Komponenten des Systems wird diese Entscheidung einfach sein. Vielfach wird es von der individuellen Zahl der geplanten Wiederverwendungen und dem individuellen Mehraufwand je Komponente abhängen. An

dieser Stelle sollte ein ökonomisches Modell ansetzen. Mittels der bereits angesprochenen doppelten Feature Point Zählung auf Komponentenbasis ist es möglich, den Aufwand hinreichend genau abzuschätzen. Wichtig ist dabei allerdings, dass die Analyse dennoch zu einem möglichst frühen Zeitpunkt durchgeführt werden kann. Auch dies wird durch die Feature Point Methodik unterstützt. Im weiteren Verlauf der Arbeit werden zusätzliche Maßnahmen vorgestellt, mit denen der Analyseaufwand weiter gesenkt werden kann.

Hinsichtlich der vorgestellten Modelle bleibt festzuhalten, dass darin viele geeignete Ansätze und Ideen vorhanden sind, die auch für das neue Modell übernommen werden. So bietet sich PuLSE-Eco als Basis für die Bestimmung des SWPL Umfangs an. Der SIMPLE Ansatz liefert mit seiner offenen Struktur und seiner intuitiven Verständlichkeit den Rahmen für die eigentlichen finanziellen Berechnungen. Und schließlich enthält das ReValue Modell gute Ansätze zur Berücksichtigung des Risikos bei Investmententscheidungen. Das Modell wird sich jedoch gegenüber COPLIMO und Poulin's Measuring Software Reuse abgrenzen, da die dort verwendete LoC Methodik zur Bestimmung des Entwicklungsaufwands als zu ungenau abgelehnt wird.[65]

Ein wichtiger zusätzlicher Punkt neben der rein wirtschaftlichen Betrachtung ist die Auswirkung von SWPL auf die Qualität der erstellten Produkte. Bevor mit der Vorstellung des e-FACTS Modell begonnen wird, erfolgt im nächsten Kapitel zunächst eine detaillierte Betrachtung zur Auswirkung von SWPL auf die Produktqualität.

[65] Capers Jones schreibt dazu: "The most common error in software estimating is derived from the very common usage of lines-of-code (LOC) metrics. Although LOC-metrics were proven to be unreliable for software estimation as long ago as 1978, the technique is still among the most widely utilized" [Jone07, S. 69].

4. Wirkung von SWPL auf die Produktqualität

Bereits zu Beginn der Arbeit wurde auf die Produktion als Ideengeber für die Grundidee der strategischen Wiederverwendung verwiesen. Gerade im Bereich der physischen Produktion spielt die Herstellungsqualität eine entscheidende Rolle. Mit der Zeit haben sich zahlreiche Ansätze entwickelt, u.a. ISO-9000-Familie, Lean Manufacturing, Total Quality Management, Kaizen und Six Sigma. Für eine ausführliche Diskussion der Methoden siehe z.B. [Tout09, S. 10 ff.]. Insbesondere Six Sigma zielt explizit auf die Reduzierung von Fehlern bei produzierten Gütern unter Berücksichtigung der Kundenwünsche und einer effizienten Umsetzung. Referenzwert ist die erwartete durchschnittliche Fehlerzahl bei einer bestimmten Anzahl an Möglichkeiten, also hergestellten Produkten.

Würde man bspw. ein Six Sigma Qualitätslevel bei SW Projekten zugrundelegen, würde dies bedeuten, dass bei der Veröffentlichung eine Fehlerbeseitigungsquote von 99,99966% erreicht werden müsste [Tout09, S. 21], wenn man eine übliche Fehlerzahl in der heutigen SW-Entwicklung zu Grunde legt. Typischerweise erreichen SW Projekte laut Capers Jones aktuell lediglich Werte von durchschnittlich 85% [Jone07, S. 290]. Weniger als 0,1% der Projekte erreicht einen Wert von über 98%. Daraus lässt sich schließen, dass die Six Sigma Anforderungen aus dem Produktionsumfeld die heutigen Möglichkeiten bei der SW Entwicklung übersteigen [Jone07, S. 178].

Diese Tatsache in Verbindung mit der steigenden Bedeutung der SW in neuen Produkten zeigt die Bedeutung der SW Qualität. Daher wird im Folgenden zusätzlich zu dem bereits angesprochenen Produktivitätseffekt auch der Effekt von SWPL auf die Qualität der SW und der Produkte allgemein genauer untersucht.

4.1. Wiederverwendung und Qualität

Die Wiederverwendung von vorhandenen Komponenten kann die Produktivität deutlich erhöhen, wenn es gelingt, den Mehraufwand bei der ersten Entwicklung und den Aufwand bei der tatsächlichen Wiederverwendung gering zu halten. Grund dafür ist, dass direkt vorhandene Komponenten verwendet werden können. Bei der Wiederverwendung von Methoden und Konzepten kann aber auch bereits die Erfahrung eine große Rolle spielen. Wenn man nun bekannte Komponenten wiederverwendet, ist davon auszugehen, dass das Risiko eines zusätzlichen Fehlers deutlich reduziert ist. Schließlich verwendet man eine

Komponente die bereits intensiv getestet wurde oder man entwickelt etwas, bei dem die Entwickler bereits auf umfangreiche Erfahrung zurückgreifen können. Natürlich erübrigt sich dadurch kein Integrationstest, da die Komponente in anderen Rahmenbedingungen und mit anderen weiteren Komponenten einwandfrei funktionieren soll. Grundsätzlich kann aber angenommen werden, dass durch die erneute Verwendung einer bereits vorhandenen Komponente neben dem Produktivitätsgewinn auch ein Qualitätsgewinn verbunden ist.

Dabei muss allerdings berücksichtigt werden, dass für eine erfolgreiche Wiederverwendung bereits ein hohes Qualitätsniveau erforderlich ist. Ansonsten läuft man Gefahr, die fehlerhaften Komponenten immer wieder neu zu verwenden und entsprechenden Anpassungsaufwand zu generieren, wodurch sich die Vorteile der Wiederverwendung schnell kompensieren können. Allgemein gilt: „Reuse is only effective for high-quality deliverables, and is hazardous and uneconomical in the presence of excessive defect levels" [Jone07, S. 444]. "Unless software quality is at state of the art levels, any attempt to reuse materials will result in longer schedules and higher costs than you have right now!" [Jone07, S. 624].

4.2. Einfluss der Variabilität bei SWPL auf die Qualität

Bei SWPL ergeben sich zusätzlich zu der ohnehin bei SW bestehenden Problematik der Gewährleistung einer hohen Produktqualität weitere Herausforderungen durch die Variabilität der Komponenten und der späteren Gestaltung von Produkten aus einer Vielzahl einzelner Standardbausteine. Dadurch erhöht sich die Komplexität des Qualitätssystems. Während für ein einzelnes Produkt in den meisten Fällen die zu erreichenden Ziele hinsichtlich der Qualität weitgehend feststehen, besteht bei SWPL die zusätzliche Komplexität, dass man zum Zeitpunkt der Erstellung einer Komponente vielfach noch gar nicht genau abschätzen kann, für welche Produkte und damit für welche genauen Qualitätsanforderungen diese auszulegen ist. Es besteht demnach eine erhöhte Unsicherheit hinsichtlich der zukünftigen Qualitätsanforderungen. Daher ist es erforderlich diese Anforderungen im Vorfeld zu untersuchen und entsprechend zu planen, gleichzeitig aber auch für zukünftige Anpassungen und Erweiterungen vorzubereiten. Um dies zu gewährleisten, wurden verschiedene Modelle erarbeitet, die toolbasiert bei der Überprüfung der Qualitätsvorgaben unterstützen. U.a. stellen diese Tools sicher, dass nur hinsichtlich ihrer Qualitätsanforderungen zulässige Kombinationen von Komponenten möglich sind. Für eine kurze Übersicht der vorhandenen Tools inkl. einer Abschätzung der einzelnen Vor- und Nachteile siehe [Bart09].

Die Verwendung dieses zusätzlichen Qualitätstools hat ebenfalls positive Auswirkungen auf die Gesamtqualität der Produkte, da hierdurch eine explizite Beschäftigung mit den Qualitätsthemen ermöglicht und die Kommunikation hinsichtlich der Anforderungen zwischen Nutzern und Entwicklern vereinfacht wird. Für den Entwickler zeigen sich insbesondere positive Auswirkungen auf die Gestaltung der SWPL Architektur, da bereits zu einem sehr frühen Zeitpunkt eine intensive Beschäftigung mit den Qualitätsanforderungen stattfindet und diese direkt in die Gestaltung der Architektur einfließen können. Und schließlich können die Folgen einzelner Entscheidungen hinsichtlich der SW Qualität für den Nutzer transparent gemacht werden. Dadurch werden Trade-off Entscheidungen erleichtert und im Falle einer Unvereinbarkeit einzelner Anforderungen können Optimierungen und individuelle Anpassungen direkt aus dem Tool heraus abgeleitet werden.

4.3. Auswirkung einer SWPL auf ein Qualitätsmodell

Hinsichtlich der Messung der Qualität von SW-Produkten ergeben sich deutlich unterschiedliche Anforderungen im Vergleich zu physischen Produkten. Daher hat sich in der Literatur auch ein weites Feld entwickelt, das sich mit dem Themengebiet der Qualität von SW beschäftigt. Für weitere Details sei an dieser Stelle lediglich auf einige Literaturquellen als Ausgangspunkt für eine weitere Vertiefung verwiesen [z.B. Kan03, Broy05, Kitc96, Eber97].

Etwas ausführlicher soll an dieser Stelle auf Qualitätsmodelle eingegangen werden. Dabei wird im Rahmen dieser Arbeit „unter einem Qualitätsmodell eine systematische Zusammenfassung von Kriterien, welche zur Beurteilung von Qualitätseigenschaften bzw. Anforderungen an die Struktur von Software geeignet sind“, verstanden [Soft11]. In ISO/IEC 9126 werden wesentliche Begriffe zum Thema SW Qualität beschrieben [ISO01]. So wird bspw. zwischen externer, interner und der Nutzungs-Qualität unterschieden. Dort wird ebenfalls ein allgemeines SW Qualitätsmodell mit 6 Kategorien definiert, das eine grundlegende Messung der SW Qualität ermöglicht. Diesem Vorgehen wird im weiteren Verlauf gefolgt.

Die Aufstellung eines Qualitätsmodells beginnt mit der Erfassung der Anforderungen des Nutzers, d.h. es werden allgemeingültige Kriterien abgeleitet, die vom Nutzer an das System/Produkt üblicherweise gestellt werden. U.a. besteht die Möglichkeit, diese Kriterien über empirische Untersuchungen im Markt zu erheben, oder das Unternehmen kennt die

Anforderungen seiner Kunden aufgrund seiner Erfahrung im Markt selbst ausreichend genug. Schließlich besteht die Möglichkeit ein internes Qualitätsmodell einzurichten, mit dem bspw. einige Attribute überprüft werden, die für die eigenen Entwickler wichtig sind. Hierbei sollte die Erfassung der Zielkriterien natürlich bei den entsprechenden Mitarbeitern erfolgen.

Nach [Soft11] bietet ISO/IEC 9126 bereits eine Auswahl an möglichen allgemeingültigen Kriterien, die im Folgenden kurz aufgelistet werden:

- **Funktionalität**, mit den Teilmerkmalen Angemessenheit, Richtigkeit, Interoperabilität, Anforderungstreue, Zugriffssicherheit
- **Zuverlässigkeit**, mit den Teilmerkmalen Reife, Fehlertoleranz, Wiederherstellbarkeit
- **Verwendbarkeit**, mit den Teilmerkmalen Verständlichkeit, Erlernbarkeit und Benutzbarkeit
- **Effizienz**, mit den Teilmerkmalen Zeitverhalten und Ressourcenverhalten
- **Pflegbarkeit**, mit den Teilmerkmalen Analysierbarkeit, Änderbarkeit, Stabilität und Testbarkeit
- **Portierbarkeit**, mit den Teilmerkmalen Anpassbarkeit, Installierbarkeit, Einhaltung von Portierbarkeitsnormen und -konventionen und Ersetzbarkeit.

Das Ziel des Qualitätsmodells ist es, diese abstrakten Kriterien mittels eines Graphen in tatsächlich messbare Kriterien herunterzubrechen. Bei diesem Graphen handelt es sich üblicherweise um einen gerichteten, azyklischen Graphen. An der Wurzel findet man die angesprochenen allgemeinen Kriterien. Die Knoten sind die Qualitätsziele in unterschiedlichen Granularitätsstufen. Die Kanten, die die Knoten miteinander verbinden, zeigen einen kausalen Zusammenhang, mit dessen Hilfe die Ziele entsprechend verfeinert werden. Die Blätter am Ende des Graphen stehen damit für die Metriken, mit denen die Merkmale auf detaillierter Ebene gemessen werden können.

Diese Verfeinerung der allgemeinen Ziele in messbare Kriterien kann u.a. mittels der FCM[66] oder GQM[67] Methoden erfolgen. Ein vollständiges Qualitätsmodell zerlegt demnach eine

[66] FCM steht für Factors-Criteria-Metrics und definiert 3 Schritte zur Erstellung eines Qualitätsmodells. 1. Merkmale (Factors) identifizieren, die die Qualität im Sinne des Produktes und des Nutzers genauer beschreiben; 2. Teilmerkmale (Criteria) ableiten, die die Merkmale konkretisieren und 3. Ableitung von Metriken je Teilmerkmal zur genauen Messung der Ausprägung.

vom Nutzer gewünschte Eigenschaft (z.B. Wartbarkeit) in messbare Subkriterien, die die ursprüngliche Eigenschaft möglichst vollständig und überschneidungsfrei (MECE) abdecken sollten. Für jedes Subkriterium definiert das Qualitätsmodell eine Skala mit akzeptablen (ausreichende Qualität) und nicht mehr akzeptablen Wertebereichen (unzureichende Qualität). Zusätzlich sollte das Modell mittels Kalibrierung in der Lage sein, bestimmte Anomalien der Messwerte zu erkennen und entsprechend zu berücksichtigen oder Nutzerinput einzufordern.

Abbildung 23 zeigt einen beispielhaften Graphen für das Thema Wartbarkeit. Anzumerken ist, dass es sich in den meisten Fällen nicht um einen tatsächlichen Baum handelt, da einzelne Metriken mehrfach verwendet werden können.

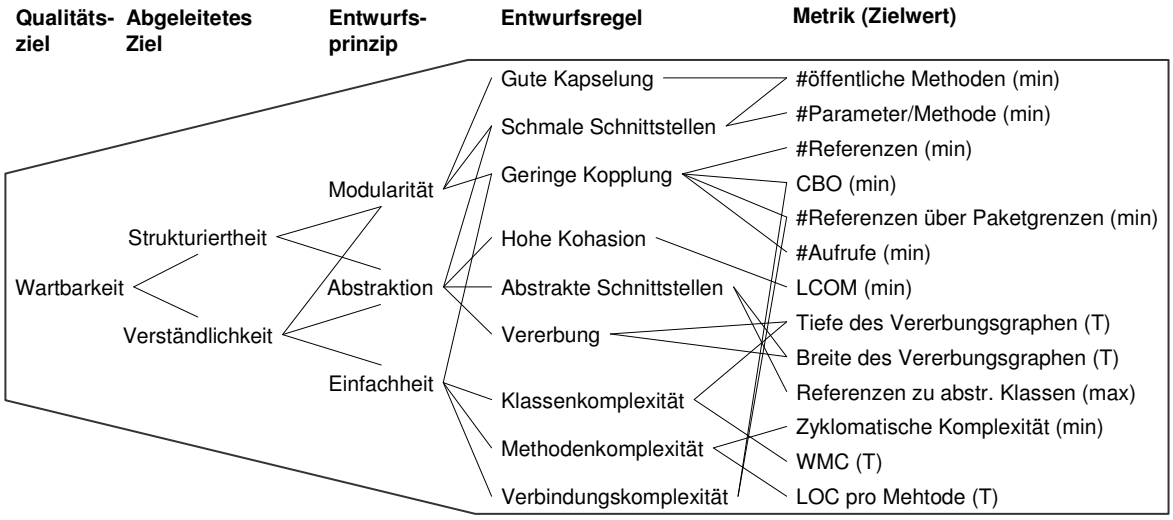

Abbildung 23 – Beispielhafter Graph eines SW Qualitätsmodells für die Wartbarkeit nach [Soft11]

In [Tren03] findet sich eine detaillierte Beschreibung der Optimierung eines Qualitätsmodells speziell für SWPL. Dabei werden 3 Hauptkriterien definiert, anhand derer bereits bestehende Ansätze für Qualitätsmodelle, wie ISO 9126 oder GQM, auf ihre Eignung für SWPL untersucht und schließlich die am besten geeigneten Methoden zu einem neuen Ansatz kombiniert werden:

---

[67] GQM steht für Goal-Question-Metrics und betont die Bedeutung von klar definierten Zielen als Ausgangspunkt für ein Qualitätsmodell. In einem ersten Schritt werden demnach die Ziele definiert, wobei 5 Merkmale beschrieben werden: Objekt (direkt mit dem Ziel inhaltlich verbunden, z.B. das fertige Produkt, also der SW Code), Aspekt des Objektes (Eigenschaft, von dem die Zielerreichung abhängt, z.B. die Wartbarkeit der SW), Absicht (wie soll der Aspekt des Objektes verändert werden, z.B. Vereinfachung der Wartbarkeit), Perspektive (wer ist der eigentliche Adressat der Zieldefinition, z.B. einfachere Wartung durch SW Techniker im Betrieb oder einfachere Nutzung durch den Endnutzer) und Kontext (Umgebungsvariablen mit einem signifikanten Einfluss auf die Zielerreichung, z.B. SW Entwicklung bei einem OEM oder Kooperation mit einem Zulieferer und dessen SW Entwicklungsabteilung). Anschließend werden die Ziele mittels Fragen konkretisiert, wobei für jede Frage eine Metrik zu definieren ist, die eine quantitative Beantwortung ermöglicht.

- **Flexibilität:** Beschreibt die Möglichkeit, das Modell im Verlauf der Entwicklung flexibel anpassen zu können. Dies ist insbesondere wichtig, um eine möglichst frühe Verwendbarkeit im Entwicklungsprozess zu ermöglichen, um somit Fehler so früh wie möglich zu erkennen, mit dem Ziel die damit verbundenen Kosten zu reduzieren. Natürlich ergibt sich dadurch die Notwendigkeit zu einer späteren Verfeinerung im Laufe des Entwicklungsprozesses.
- **Reusability:** Konsequenterweise sollte bei der Anpassung von Qualitätsmodellen auf die Bedürfnisse von SWPL auch auf die Wiederverwendung von Elementen des Qualitätsmodells geachtet werden. Ziel ist es, einen möglichst großen Teil der Elemente über die verschiedenen Entwicklungsstufen der SWPL zu verwenden und neue Erkenntnisse im Sinne eines Continuous Improvements möglichst einfach in de SWPL einfließen zu lassen.
- **Transparency:** Die Wiederverwendung des Qualitätsmodells wird vereinfacht, wenn Klarheit über die Einzelheiten bei der Erstellung bestehen. Insbesondere sollte ein klar definiertes Verständnis für die Ableitung und Detaillierung von Charakteristika sichergestellt sein.

Doch wie wirkt sich nun eine SWPL auf die Qualität der auf dieser Basis entwickelten Produkte aus? Zunächst einmal hat die Einführung einer SWPL keine direkte Auswirkung auf die Qualität der entwickelten SW, da es sich allein um einen Ansatz handelt, mit dem die Wiederverwendung von einzelnen Komponenten erhöht wird. Allerdings kann allgemein davon ausgegangen werden, dass eine höhere Wiederverwendung von einzelnen Bausteinen die Qualität der Produkte insgesamt steigert. Dies setzt natürlich voraus, dass die wiederzuverwendenden Komponenten selbst eine entsprechende Qualität aufweisen. Grund für diese Annahme ist, dass aufgrund der geplanten Wiederverwendung eine sorgfältigere Erstellung und ein intensiveres Testing der Komponenten ökonomisch sinnvoll sind. Bedingt durch die höheren Stückzahlen und der längeren geplanten Verwendung der Komponenten ergibt sich eine andere Kalkulationsbasis, die entsprechend höhere Kosten zur Erzielung einer höheren Qualität erlaubt. Dies ist allerdings nur eine implizite Schlussfolgerung und keine automatische Konsequenz aus der Anwendung des SWPL Ansatzes. Fraglich ist daher, ob sich diese rationalen Annahmen in der wirtschaftlichen Praxis tatsächlich nachweisen lassen. In der Praxis hat sich diese Annahme vielfach bestätigt. So hat bspw. HP durch die Verwendung von SWPL die Anzahl der Defekte in seinen Produkten um bis zu 96%

reduzieren können [SWPL11]. Das SEI der Carnegie Mellon University berichtet allgemeiner von einer durchschnittlichen Möglichkeit zur Verbesserung der Qualität von ca. 10 %, nennt aber zusätzlich ein konkretes Beispiel, bei dem das U.S. National Reconnaissance Office eine bereits bestehende SW Lösung für die Bodensteuerung von Satelliten zu einer SWPL erweitert hat. Als Ergebnis konnte eine 10-fach höhere Produktqualität beobachtet werden [SEI11]. Salion, ein SW Start-up Unternehmen, das sich auf die Erstellung von Unternehmens-Software-Systemen spezialisiert hat, mit dessen Hilfe Unternehmen die Front-end Prozesse bei der Bearbeitung von Angeboten (RFQs) verbessern können, hat bereits mehrere Generationen von SWPL realisiert. In Abbildung 24 sind die tatsächlich erkannten Fehler während des Produkt-Tests für einzelne Produkte in einer SWPL abgetragen. Es zeigt sich ein klarer Trend zu einer Reduktion der Fehler über die Laufzeit einer SWPL. Aber auch über die verschiedenen Generationen von SWPL (Release 1 bis 3) hinweg zeigt sich ein Lerneffekt und damit eine Reduktion der Fehler. So konnte die Anzahl der Fehler im ersten Produkt bei der 3. Generation von SWPL beinahe halbiert werden.

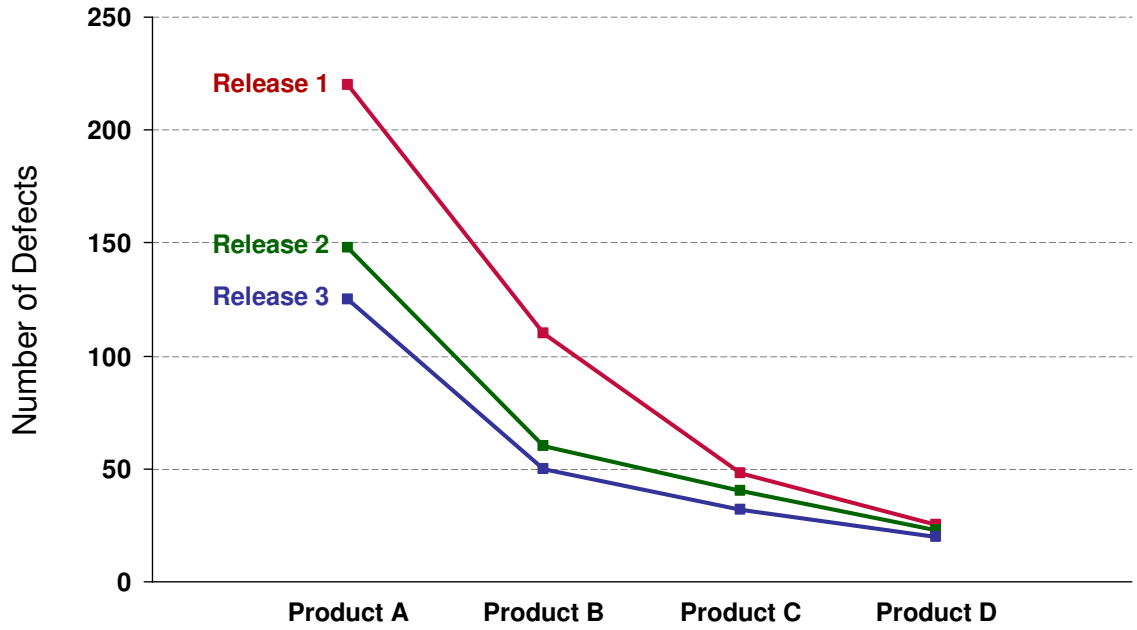

Abbildung 24 – Entwicklung der Defektanzahl über die Lebenszeit von SWPL [SWPL11]

4.4. Bedeutung der Qualität im Rahmen dieser Arbeit

Bevor im nächsten Kapitel mit der Vorstellung des e-FACTS Modells zur einfachen ökonomischen Bewertung von SWPL für einzelne, spezifische Unternehmenskonstellationen

begonnen wird, soll abschließend die Bedeutung der Qualität im Rahmen der hier vorliegenden Arbeit genauer erläutert werden.

Wie soeben beschrieben spielt Qualität insbesondere bei SWPL aufgrund der strategischen Wiederverwendung eine besondere Rolle. Ohne ausreichende Maßnahmen zur Sicherstellung einer hohen SW Qualität wird eine SWPL ökonomisch nicht erfolgreich sein können. Allerdings kann diese Anforderung als eine Grundbedingung für die erfolgreiche Einführung einer SWPL in einem Unternehmen verstanden werden, ebenso wie bspw. die Bereitschaft zu organisatorischen Anpassungen oder eine deutlich sichtbare Unterstützung des Vorhabens durch das Management. Das übergreifende Ziel bleibt in den meisten Fällen eine ökonomische Verbesserung, sei es durch eine Reduktion der Entwicklungszeiten oder der –Kosten.

Daher steht die Betrachtung der Qualität nicht im Fokus dieser Arbeit. Es wird vielmehr angenommen, dass die Qualität zumindest auf einem konstanten Niveau gehalten wird oder sich sogar aufgrund der Einführung einer SWPL verbessert. Die bisherigen Ausführungen aus Kapitel 4 unterstützen diese Schlussfolgerung. Eine direkte ökonomische Auswirkung oder Bewertung einer solchen Veränderung wird nicht berücksichtigt.

Im Rahmen eines Vorab-Fragebogens, der frühzeitig ungeeignete Anwendungsfälle identifizieren soll, um damit den Analyseaufwand zu minimieren, wird allerdings die Qualität in der SW Entwicklung des jeweiligen Unternehmens vor Einführung der SWPL abgefragt (siehe Kapitel 5.1.1.2.3). Nur wenn diese mindestens ein durchschnittliches Niveau erreicht, wird die weitere Verfolgung des SWPL Vorhabens empfohlen. Ansonsten sollte zunächst das Qualitätsproblem unabhängig von der Einführung einer SWPL gelöst werden. Schließlich hat sich in den bisherigen Erfahrungen mit SWPL gezeigt, dass dieser Ansatz gut dazu geeignet ist, um Schwierigkeiten in der Produktivität oder gesteigerte Kundenanforderungen hinsichtlich verkürzter Entwicklungszeiten zu begegnen. Es gab allerdings bisher kein erfolgreiches Beispiel einer SWPL Einführung mit dem Hauptziel die Produktqualität zu steigern. Nur in wenigen Fallstudien wird die Qualität als eigener Punkt genannt. Dann zwar immer mit einer signifikanten Verbesserung bedingt durch die SWPL Einführung, allerdings als zusätzlicher positiver Effekt meist in Verbindung mit einer Reduktion der Entwicklungszeit oder der –Kosten.

5. Entwicklung eines umfassenden Modells zur frühzeitigen Bewertung von Produktlinienansätzen

Aufgrund der diskutierten Vor- und Nachteile der derzeit verfügbaren Bewertungsmodelle soll im Folgenden nun ein neuer, erweiterter Ansatz zur Bewertung der ökonomischen Vorteilhaftigkeit des Einsatzes von Software Produktfamilien detailliert hergeleitet werden. Der Fokus liegt dabei auf einer einfachen, schnellen Anwendung für alle Unternehmen im Automobilbereich und einer Erfassung der Entwicklungsaufwände auf Funktionsebene, wie er in [Pohl05, S. 438] gefordert wird. Die Analyse setzt sich aus mehreren Schritten zusammen, die mit Ausnahme der SIMPLE-basierten Berechnung in Kapitel 5.2 nacheinander zu durchlaufen sind. Nach Vollendung eines Schrittes wird jeweils eine individuelle Analyse erfolgen, mit dem Ziel eindeutig nicht geeignete Projekte möglichst früh zu identifizieren und dadurch den Aufwand für die Unternehmen gering zu halten.

In **Kapitel 5.1** wird ein grundlegender Fragenkatalog vorgestellt, der die Fähigkeit eines Unternehmens zur Umsetzung und Realisierung einer SWPL analysiert. Eine möglicherweise negative Empfehlung hinsichtlich einer SWPL wird dabei um Verbesserungsvorschläge ergänzt, um den Unternehmen Ideen zu geben, wie eine zukünftige Realisierung dennoch positiv möglich sein könnte.

In **Kapitel 5.2** erfolgt eine vereinfachte erste ökonomische Berechnung auf Basis eines adaptierten SIMPLE Ansatzes. Aufgrund des niedrigen hiermit verbundenen Aufwandes erscheint dieses Vorgehen als zusätzlicher Konsistenzcheck sinnvoll, könnte aber falls nötig als einziger Analyseschritt übersprungen werden.

Im **Kapitel 5.3** wird das eigentliche e-FACTS Modell vorgestellt. Zunächst werden die benötigten Basisdaten erhoben und eine Produktstruktur erfasst. Zusätzlich wird ein Anwendungsbeispiel eingeführt, das stringent für die gesamte e-FACTS Beschreibung genutzt wird, um die einzelnen Analyseschritte zu veranschaulichen. Anschließend wird in **Kapitel 5.4** die Bewertung des Software Erstellungsaufwands mittels einer abgewandelten Function Point Methode nach IFPUG [IFPU09] durchgeführt. Die Feature Point Methode wurde in Teil A Kapitel 5.7 vorgestellt und wird im weiteren Verlauf als FP bezeichnet. Hierbei handelt es sich um den zeitlich aufwendigsten Teil der Analyse, denn es erfolgt eine doppelte FP Zählung, einmal als Einzelentwicklung und zweitens als Teil einer größtmöglichen Produktfamilie (sprich einer größtmöglichen Asset-Base). Diese beiden

Werte ermöglichen den ökonomischen Vergleich. Zusätzlich bietet dieses Vorgehen die Möglichkeit, eine teilautomatische Optimierung der Asset-Base durchzuführen.

Kapitel 5.5 beschreibt die Zusammenfassung der ermittelten Aufwandsdaten und deren Erweiterung zu einem vollständigen Business Case. Im abschließenden **Kapitel 5.6** werden die Anwendungsmöglichkeiten des Modells bei unterschiedlichen Produktszenarien (existierende Produkte, Erweiterung einer SWPL, usw.) erläutert. Es wird also gezeigt, welche Fragestellungen e-FACTS eigentlich beantworten kann. Abschließend wird gezeigt, welche Auswirkungen unterschiedliche Unternehmensziele auf die Anwendung des Modells haben können. So kann es bspw. für ein Unternehme mit einem klaren Wachstumsziel notwendig sein, neben den rein monetären Werten den Einfluss einer Produktlinieneinführung auf die Release Zeitpunkte neuer Produkte stärker zu gewichten, da diese einen starken Einfluss auf das Unternehmenswachstum haben könnten.

5.1. Schritt 1: Fragenkatalog mit Basiskriterien

5.1.1. Kategorisierung der Fragen

"Since the late 80ies, a number of capabilities evaluation models have been developed for various fields of assessment" [Lind04, S. 127]. Das heute bedeutendste ist sicherlich immer noch das CMMI Modell (siehe Teil A Kapitel 2.2.4.1.). Es kann als de facto Standard im Bereich der Bewertung von Prozessen zur SW Erstellung angesehen werden [Lind04, S. 119]. Speziell für SWPL haben sich aus den in Teil A Kapitel 2.2.5. beschriebenen SWPL Forschungsprojekten mehrere Modelle entwickelt, u.a. PLTP. "The SEI's Product Line Technical Probe (PLTP) allows examining an organisation's readiness to adopt or its ability to succeed with a software product family engineering approach" [Lind04, S. 18]. PLTP basiert auf der bereits beschriebenen SPLP, die insgesamt 29 Practice Areas unterscheidet, die wiederum grob in drei Kategorien aufgeteilt werden können: SW Engineering, Technical Management und Organisational Management [SEI10]. Der Fokus liegt zwar auf den Prozessen des Unternehmens, aber es werden auch andere Bereiche umfassend berücksichtigt, wie z.B. die Erstellung eines Business Cases. In [Lind04] wurde PLTP auf Basis des BAPO-Modells leicht erweitert, indem zu den drei Kategorien eine zusätzliche Businessperspektive hinzugefügt worden ist. BAPO steht für Business, Architecture, Process und Organisation. Sowohl PLTP als auch BAPO gehen demnach deutlich über den Prozessfokus des CMMI-Modells hinaus. Grund dafür ist, dass gute Prozesse allein keinen Umsetzungserfolg einer

SWPL garantieren, sondern dass alle vier genannten Kategorien einen signifikanten Einfluss haben.

Der in dieser Arbeit verwendete Fragebogen basiert daher auf den Ideen von PLTP und BAPO. Allerdings zielen diese beiden Ansätze auf eine vollständige Bewertung der Produktlinienfähigkeit eines Unternehmens ab und zwar zu einem Zeitpunkt, zu dem sich das Unternehmen bereits intensiv mit dem Thema und einer möglichen Umsetzung beschäftigt hat. So gilt bspw. für das BAPO Modell: „The evaluation framework is intended for use within software developing organisations to determine the status of their own software product family engineering and the priorities to improve" [Lind04, S. 110].

Der hier vorliegende Fragebogen hat eine davon abweichende Zielsetzung. Er dient dazu, den an einer Produktlinie interessierten Unternehmen sehr schnell eine erste grobe Antwort zu geben, ob eine genauere Beschäftigung mit diesem Thema überhaupt Sinn macht. Gleichzeitig zielt der Fragebogen explizit auf die Automobilindustrie ab. Auch wenn dies aufgrund der allgemeinen Fragestellung nicht bei jeder einzelnen Frage direkt ersichtlich ist und viele davon sicherlich auch für Unternehmen außerhalb der Automobilindustrie verwendet werden könnten. Grund dafür ist, dass der Großteil der notwendigen Eigenschaften für eine erfolgreiche Einführung einer SWPL unabhängig vom Unternehmenszweck ist. Daher bleiben die Fragen an einigen Stellen eher allgemein, allein schon aufgrund der Tatsache, dass der Fragebogen keine Unterscheidung nach Unternehmensgröße vornimmt. Allerdings erfolgt an einigen Stellen eine Differenzierung nach OEM und Zulieferern, da sich hier die deutlichsten Unterschiede bei der Anwendung einer SWPL zeigen.

Für das eigentliche Ziel einer frühzeitigen groben Abschätzung der Umsetzungschancen einer SWPL sind die genannten Einschränkungen jedoch nur von geringer Bedeutung. Es genügt, die klar ungeeigneten Fälle anhand von Erfahrungswerten zu erkennen und zusätzlich bei kritischen Themen vor einer weiteren Beschäftigung mit dem Thema Produktlinie zu einer intensiven Diskussion anzuregen.

Neben dem PLTP und dem BAPO-basierten Modell finden sich in der Literatur zahlreiche weitere Evaluierungsansätze, die die Fähigkeit von Unternehmen für Wiederverwendung von

SW im Allgemeinen oder speziell für SWPL bewerten.[68] Allerdings fokussieren diese Ansätze noch stärker auf die eigentliche Umsetzung und Realisierung. Auch setzen sie teilweise fundierte Kenntnisse zum Thema SWPL voraus [Schr04, S. 151]. Der in Kapitel 3.4 bereits vorgestellte PuLSE-Eco Ansatz kommt dem hier vorliegenden Fragebogen sehr nahe, speziell die Domain Potential Analysis, allerdings ist der Fokus ein anderer. PuLSE-Eco verwendet den Fragebogen, um die Produktbereiche mit dem höchsten Wiederverwendungspotenzial zu identifizieren. Daher ist zunächst eine Beschreibung der SWPL und ihrer Domäne notwendig [Schm04, S. 49ff.]. Mit einem Umfang von ca. 100 Einzelkriterien ist der Fragbogen gleichzeitig deutlich umfangreicher.

Im Gegensatz dazu ermöglicht der hier verwendete Fragebogen eine grundsätzliche Eignungsanalyse innerhalb von lediglich einer Stunde, und dies ganz zu Beginn einer SWPL Analyse. Dies wird durch eine Fokussierung der Fragen auf die wichtigsten Bereiche erreicht, insbesondere auf Themen, die bereits vor einer intensiven Beschäftigung mit dem Thema SWPL von einem Unternehmen beantwortet werden können und die trotzdem eine Aussagekraft über die späteren Erfolgsaussichten einer SWPL Einführung haben. So wurden bspw. einige der 29 Practice Areas des PLTP ausgelassen (z.B. Betrachtung eines Business Cases, da dies erst zu einem späteren Zeitpunkt in dieser Arbeit relevant ist) oder neu gruppiert und zusammengefasst (z.B. wurden die einzelnen, sehr detaillierten Elemente der aktuellen SW-Entwicklung, wie Component Development, Integration und Testing, zu einem Punkt SW-Entwicklung zusammengefasst). Auf die Entsprechungen und evtl. Abweichungen gegenüber den genannten Modellen wird bei der Detailbeschreibung der einzelnen Fragen jeweils hingewiesen.

Insgesamt umfasst der verwendete Fragebogen 20 Fragen, die in fünf Kategorien gruppiert sind. Bei der Definition der Kategorien wurde BAPO als Grundlage genommen, allerdings erweitert um eine zusätzliche Kategorie, das Produktangebot. Grund dafür ist, dass dieses Thema insbesondere in einer sehr frühen Phase wichtige Anhaltspunkte über die Komplexität der Umsetzbarkeit einer SWPL geben kann. Die restlichen 4 Kategorien entsprechen den Elementen des BAPO Modells. Nachfolgend werden die 5 Kategorien kurz erklärt: Das **Produktangebot** beschäftigt sich mit der aktuellen und zukünftigen Angebotssituation des Unternehmens und berücksichtigt die Anforderungen der spezifischen Wettbewerbssituation

[68] Zu nennen sind hier: SW-CMM V2.0 als Erweiterung der CMMI Methodik. Das Kapitel Organization Software Asset Commonality setzt dabei allerdings erhebliche Kenntnisse über SWPL voraus. SW-CMM V2.0 wurde niemals offiziell freigegeben [Schr04, S. 151]. Ein ebenfalls umfangreicher Bewertungsansatz ist das Reuse Adoption Guidebook [SPC93].

des Unternehmens. Die zweite Kategorie **Produktentwicklung** (im BAPO Modell als Process bezeichnet) bewertet generell den Reifegrad des Unternehmens bei der Erstellung von Software. Die interne **Organisation** und deren Eignung für eine Produktfamilie werden anschließend analysiert. Hier wird überprüft ob die aktuelle Organisation gut zu den Anforderungen einer SWPL passt oder ob die notwendigen Veränderungen zumindest leicht umzusetzen sind. Es folgt eine Bewertung der aktuellen Software **Architektur** mit einem Fokus auf die mögliche Weiterverwendbarkeit und leichte Anpassbarkeit an die Anforderungen einer SWPL. Am Ende folgt eine Bewertung der **Business Drivers**, also vergangene Geschäftsentscheidungen, die einen Einfluss auf eine SWPL haben.

Der Fragebogen ermittelt die Eignung eines Unternehmens für die Einführung einer SWPL, ausgedrückt als Prozentwert. Ebenso wie das BAPO-basierte Modell bietet auch dieser Fragebogen eine Leveleinstufung an, um Verbesserungsmöglichkeiten identifizieren zu können. Als Erweiterung wurden jedoch einige Fragen als "Ausschlusskriterien" definiert. Diese Fragen haben eine so große Bedeutung, dass die Unterschreitung eines bestimmten Schwellenwertes direkt und unabhängig von den anderen Antworten zu einem insgesamt negativen Urteil über eine mögliche Produktfamilie führt. Sollte eine solche Frage unterhalb des Schwellenwertes beantwortet werden, wird zusätzlich zur negativen Empfehlung der gesamte Prozentwert angegeben. Dieser kann durchaus über den ansonsten geltenden Grenzen liegen. Als Basis für eine Diskussion werden weiterhin die verfehlten Ausschlusskriterien explizit ausgewiesen. Dadurch soll eine fokussierte Entscheidung ermöglicht werden, ob und mit welchem Aufwand diese Barrieren für eine erfolgreiche SWPL Einführung überwunden werden können.

Der Großteil der Fragen muss beantwortet werden. Eine Nichtbeantwortung wird mit 0 Punkten bewertet. Zusätzlich gibt es einige optionale Fragen, die das Unternehmen möglichst beantworten sollte. Bei einer Nichtbeantwortung wird diese Frage jedoch für die Auswertung nicht berücksichtigt. Die Beantwortung einer optionalen Frage bedeutet keinen generellen Vorteil in der finalen Bewertung, da bei einer Beantwortung auch die mögliche Maximalpunktzahl und damit die Basis für die 100% Bewertung ansteigt. Eine Bewertung einer optionalen Frage mit einer Einstufung oberhalb des Durchschnitts aller bisherigen Antworten wirkt positiv, ansonsten verringert sich der Gesamtwert leicht oder bleibt unverändert.

Alle Fragen können mit einem Wert zwischen 1 und 5 bewertet werden, wobei 1 den niedrigsten und 5 den höchsten Wert darstellt. Die Unterschiede zwischen den einzelnen Stufen sind linear verteilt. Um eine Vergleichbarkeit und Verrechenbarkeit zu ermöglichen, wird die Einstufung je Frage mit einer individuellen Gewichtung bewertet. Diese liegt ebenfalls zwischen 1 und 5. Demnach ergibt sich für das Ergebnis einer einzelnen Frage je nach Gewichtung eine Bandbreite von 1 bis 25. Die jeweiligen Maximalergebnisse aller obligatorischen und der beantworteten optionalen Fragen werden addiert, um die Basis der 100% Bewertung zu erhalten. Zusätzlich zur Gesamtbetrachtung erfolgt eine Einzelbetrachtung je Kategorie. Zwar kann mittels der Gewichtung eine grundsätzliche Vergleichbarkeit der einzelnen Fragen erreicht werden, dennoch haben die zuvor definierten 5 Kategorien eine so große individuelle Bedeutung, dass auch hier jeweils ein Mindestwert erreicht werden sollte. Dadurch wird die Kompensation von vorhandenen Schwächen durch Stärken in anderen Bereichen eingeschränkt.

Die einzelnen Gewichte und eventuellen Ausschlusskriterien basieren auf einer umfangreichen Literaturanalyse der vorhandenen Ansätze und eigenen Erfahrungen. Die Begründungen und Herleitungen (z.B. der Gewichte) sind jeweils bei der Erklärung der Frage zu finden. Der vollständige Fragebogen ist im Anhang abgebildet.

5.1.1.1.Produktangebot

5.1.1.1.1. Anzahl der derzeitigen Produkte im Markt

Die Anzahl unterschiedlicher Produkte, die ein Unternehmen derzeit im Markt anbietet (und zwar inkl. aller SW Produktvarianten) gibt ein Indiz für die Wahrscheinlichkeit, eine SWPL erfolgreich etablieren zu können. Der Fokus sollte dabei natürlich auf dem Unternehmensbereich liegen, für den die Bewertung der SWPL Optionen durchgeführt werden soll. Bei großen Unternehmen ist demnach nicht die Gesamtzahl der angebotenen Produkte heranzuziehen, sondern die Anzahl der Produkte aus dem entsprechenden Bereich. Je umfangreicher das Angebot ist, desto größer ist die Wahrscheinlichkeit, dass sich ein passender Produktbereich findet, der zusätzlich die notwendige Größe aufweist. "Kaum ein Unternehmen wird direkt eine Produktlinie mit mehreren Produkten planen, ohne schon Systeme in dem entsprechenden Bereich realisiert zu haben" [John04, S. 154]. Natürlich könnte auch ein Unternehmen mit nur einem einzigen Produkt eine SWPL als Erweiterung des bisherigen Angebotes einführen. Die damit verbundenen Risiken steigen allerdings an im

Vergleich zu einem Unternehmen, was bereits breiter im Markt positioniert ist und lediglich eine begrenzte Erweiterung plant.

Die Grenzen für eine Einstufung mit 1 liegen bei 1 bis 4 Produkten. Bis einschließlich 8 Produkten erfolgt eine Bewertung mit 2 Punkten. 3 Punkte erhält ein Unternehmen mit weniger als 16 Produkten, bis 30 Produkte 4 Punkte. Und bei mehr als 30 angebotenen Produkten erfolgt die Höchstbewertung mit 5 Punkten. Die Einstufungskriterien können der nachfolgenden Tabelle entnommen werden. Für die weiteren Fragen wird dieses Vorgehen zur Vereinfachung weitestgehend bevorzugt.

Es ist anzunehmen, dass diese Frage für Zulieferer im Durchschnitt zu einer höheren Bewertung führt als für OEMs, da Zulieferer im Allgemeinen eine größere Anzahl an Produkten aufweisen. In einem konkreten Fall würde dies bedeuten, dass der Zulieferer eine höhere Wahrscheinlichkeit aufweist, die SWPL ökonomisch erfolgreich umzusetzen. Daher eignet sich diese Frage im Rahmen der vorgestellten Zielsetzung, auch wenn der Zusammenhang nur schwach ausgeprägt ist und eine hohe Produktanzahl nicht in jedem Fall einen Erfolg der SWPL bedeutet. Vielmehr handelt es sich um eines von zahlreichen Indizien, die mittels des Fragebogens erfasst werden sollen, um eine möglichst valide Gesamtaussage treffen zu können. Insgesamt erfolgt die Gewichtung mit einem Wert von 3. Bei einer sehr geringen Zahl an Produkten kann schließlich von einem höheren Risiko ausgegangen werden. Der Wert gibt demnach ein Indiz für mögliche Schwierigkeiten.

Frage	Einstufungen 1	2	3	4	5
Wieviele unterschiedliche Produkte bieten sie aktuell am Markt an?	1 bis 4	5 bis 8	9 bis 15	16 bis 30	mehr als 30

Tabelle 5 – Kategorie Produkte, Frage 1

5.1.1.1.2. Abdeckungsgrad der Produktlinie

Ergänzend zur ersten Frage ist es relevant zu wissen, wie Groß der Anteil der vorhandenen Produkte ist, der über die Produktlinie abgedeckt werden soll. Grundsätzlich kann angenommen werden, dass eine größere Abdeckung zu einer höheren Akzeptanz und Bedeutung der Produktlinie innerhalb des Unternehmens führen wird. Als Ergebnis lässt sich daraus eine höhere Erfolgswahrscheinlichkeit ableiten. Unternehmen, die lediglich einen kleinen Anteil ihrer Produkte in die Produktlinie integrieren wollen oder aufgrund von anderen Einschränkungen können, werden es schwerer haben, die erwarteten Vorteile zu realisieren. Die Frage zielt explizit auf die endgültige Größe der Produktlinie ab, deckt also

auch die schrittweise Einführung einer SWPL ab. Gemeint ist aber nicht der Umfang je Produkt, der über die SWPL realisiert wird, sondern der Anteil der Produkte, die Elemente der Produktlinie enthalten. Eine genauere Einschätzung hinsichtlich des tatsächlichen Abdeckungsgrads einzelner Funktionen durch die Asset Base dürfte zu diesem Zeitpunkt meist unmöglich sein, ist aber auch eindeutig nicht das Ziel dieses Fragebogens. Wie schon bei Frage 1 gilt auch hier möglicherweise nicht das Gesamtunternehmen als Bezugsbasis, sondern der Unternehmensteil, für den die Einführung der SWPL geplant ist. Es sollte allerdings beachtet werden, dass der entsprechende Unternehmensteil eine relevante Größe für das Unternehmen hat.

Sollte das Unternehmen weniger als 5% seiner bestehenden Produkte für die Produktlinie berücksichtigen, erfolgt die niedrigste Einstufung. Ab einem Anteil von über 50% wird der Höchstwert vergeben. Die Gewichtung erfolgt mit dem Höchstwert von 5. Grund dafür ist, dass dieser Anteilswert ein direktes Indiz für die zukünftige interne Bedeutung einer Produktlinie ist. Fast alle Umsetzungen, bei denen "man es nur mal ausprobieren wollte", sind gescheitert. Wichtig ist, dass es hier lediglich um eine sehr frühe Abschätzung aus Sicht des Unternehmens geht. Es soll sichergestellt werden, dass mit ambitionierten Zielwerten gestartet wird. Damit ist allerdings nicht die Einführungsgeschwindigkeit der SWPL gemeint. Auch wenn zunächst ein evolutionärer Ansatz mit einer lediglich langsamen Anwendung der SWPL in die bestehenden Produkte verfolgt wird, kann diese Frage mit einem hohen Wert beantwortet werden, da auf den finalen Wert abgezielt wird.

Für OEMs bedeutet eine weitreichende Verwendung der SWPL meist eine Integration über verschiedene Modelle hinweg, von der Luxusklasse bis zum Einstiegsmodell. Dadurch ergeben sich größere Herausforderungen als bei Zulieferern, die eine Produktklasse mehreren OEMs anbieten können. Allerdings besteht im letztgenannten Fall eine zusätzliche Herausforderung durch die Frage nach der Sicherung des Intellectual Property. Daher lehnen zahlreiche OEMs bei kritischen Komponenten den Einsatz von SWPL durch die Zulieferer ab, da sie befürchten einen Wettbewerbsvorteil zu verlieren.

Diese Frage stellt das erste Ausschlusskriterium dar. Der Grund dafür ist die hohe Bedeutung eines ambitionierten Zielwertes direkt bei der Planung einer SWPL. Sollte das Unternehmen hier mit einem zu niedrigen Ziel an den Start gehen, wird dadurch die erfolgreiche Umsetzung gefährdet, da möglicherweise keine ausreichende Unterstützung für diese wichtige strategische Aufgabe besteht. Ab einem Anteil von über 10%, also einer Einstufung von 3,

gilt dieses Minimum als erreicht. Zu Bedenken ist allerdings, dass dieser Wert stark von der Gesamtzahl der angebotenen Produkte abhängig ist. Dies sollte bei einem etwaigen Unterschreiten berücksichtigt und entsprechend diskutiert werden. Wie bereits erwähnt dienen die Ausschlusskriterien eben nicht dazu eine definitiv negative Empfehlung zu geben, sondern vielmehr auf die entscheidenden Erfolgsfaktoren hinzuweisen, die vor einer SWPL Einführung noch entsprechend anzupassen sind.

Frage	Einstufungen 1	2	3	4	5
Können sie bereits abschätzen, wieviele ihrer Produkte zumindest teilweise durch die SWPL abgedeckt werden sollen?	unter 5%	5 bis 10%	über 10 bis 20%	über 20 bis 50%	mehr als 50%

Tabelle 6 – Kategorie Produkte, Frage 2

5.1.1.1.3. Anzahl geplanter neuer Produkte in den nächsten 2 Jahren

Produktlinien eignen sich insbesondere in einer Umgebung, in der zahlreiche neue Produkte in der nahen Zukunft auf den Markt kommen, die sich nicht grundsätzlich unterscheiden. Eine typische Situation wäre ein gesättigter oder schrumpfender Markt, auf dem die Anbieter durch stärkere Kundenorientierung versuchen, ihren Marktanteil eher durch Verdrängung als durch wirkliches Wachstum zu erhöhen. In dieser Situation zahlen sich die Kostenvorteile einer Produktlinie durch strategische Wiederverwendung stärker aus [vgl. Pohl05, S. 175]. Ohne neue Produkte oder Produktvarianten verliert der Ansatz seine Grundlage. Daher stellt diese Frage ein weiteres Ausschlusskriterium dar. Sollte das befragte Unternehmen in den nächsten 2 Jahren kein neues Produkt planen, dann kann schon allein aus dieser Tatsache abgeleitet werden, dass eine Produktlinie ungeeignet ist. Die finale Empfehlung wäre demnach eine Ablehnung. Allerdings basiert diese zunächst nur auf einer einzigen Beobachtung. Um ein Missverständnis auszuschließen und den Unternehmen Handlungsempfehlungen mit auf den Weg zu geben, werden negativ beantwortete Ausschlusskriterien am Ende explizit aufgeführt und erneut diskutiert. Dies ermöglicht eine fundierte Entscheidung, ob z.B. die Zahl der neuen Produkte nicht doch höher liegen könnte.

Der enge Zeitrahmen von 2 Jahren wurde gewählt, um den Unternehmen die Beantwortung zu erleichtern, schließlich soll an dieser Stelle zunächst lediglich die generelle Eignung für eine Produktlinie geprüft werden. Für OEMs wäre aufgrund der langfristigen Produktentwicklung und der daraus resultierenden langfristigen Markteinführungsplanung sicherlich auch ein deutlich längerer Zeitraum von bis zu 5 Jahren möglich. Für die Zulieferer hingegen ist ein kürzerer Zeitraum in einigen Fällen notwendig, da die Planungsintervalle hier kürzer sind.

Die Grundlage zur Bestimmung der Anzahl neuer Produkte sollte je nach individueller Situation bestimmt werden. Für einen OEM muss es z.B. nicht zwingend die Anzahl der neu in den Markt gebrachten Fahrzeugmodelle sein. Bezieht sich die geplante SWPL auf einen bestimmten Produktbereich, wie z.B. die Infotainmentsysteme, dann könnte die Bewertungsbasis auch die Anzahl der neuen Systeme in diesem Bereich sein, unabhängig von der Anzahl der Fahrzeugmodelle.

Der allgemeine Zeitrahmen für die spätere ökonomische Betrachtung wird 5 Jahre betragen. Im Allgemeinen wird angenommen, dass aufgrund der schnellen technologischen Veränderung „the technology time horizon to no more than three to five years“ [Clem02, S. 329] beschränkt ist. Daher ergibt sich auch nur eine durchschnittliche Gewichtung dieser Frage mit dem Wert 3. Sollte kein neues Produkt geplant sein, erfolgt eine Einstufung mit 1, ab einer Neuproduktanzahl von mehr als 10 erfolgt die höchste Einstufung.

Frage	Einstufungen 1	2	3	4	5
Wie viele neue Produkte planen sie in den nächsten 2 Jahren auf den Markt zu bringen?	keins	1 bis 2	3 bis 5	6 bis 10	mehr als 10

Tabelle 7 – Kategorie Produkte, Frage 3

5.1.1.1.4. Neuigkeitsgrad der relevanten Produktgruppe

Der Neuigkeitsgrad spielt für SWPL eine entscheidende Rolle. Einerseits ist es das Ziel möglichst viele neue Produkte in kurzer Zeit auf den Markt zu bringen, um von der Idee der strategischen Wiederverwendung zu profitieren. Andererseits ist es notwendig, dass sich die einzelnen Produkte dabei nicht zu stark voneinander unterscheiden. "If everything changes every half-year in an unpredictable way, the investment costs never pay off" [Pohl05, S. 18]. Denn dann wären große Anpassungen an den Assets in der Domain, also der Sammlung der wiederverwendbaren SW-Bestandteile, notwendig, die eine ökonomisch sinnvolle Wiederverwendung bedrohen. "Systematische Varianz und ein hinreichendes Maß an Gemeinsamkeit zwischen den Systemen sind Voraussetzungen für erfolgreiche Wiederverwendung" [Schm04, S. 50].

Das PLTP Modell berücksichtigt dieses Element mit insgesamt 2 Practice Areas: der Market Analysis und dem Technology Forecasting. Hier wird versucht, durch geeignete Analysen ein besseres Verständnis über die zukünftigen Veränderungen zu erlangen und darauf aufbauend die SWPL entsprechend gestalten zu können. Allerdings sind diese Practice Areas im PLTP

der Kategorie Organizational Management zugeordnet (entspricht Kapitel 5.1.1.3; Organisation im Rahmen dieser Arbeit), da dort kein eigener Produktbereich existiert.

Basis für die Bewertung im Fragebogen ist die durchschnittliche historische Veränderung der am Markt angebotenen Produkte. Es kommt nicht auf die derzeitige Realisierung im Unternehmen an. D.h. selbst ein Unternehmen, dass derzeit jedes Produkt als Einzelentwicklung von Grund auf konzipiert, kann einen hohen Wert erreichen, wenn diese Tatsache allein auf der derzeitigen Umsetzung im Unternehmen beruht, der Markt aber auch eine stärkere Wiederverwendung zulassen würde. Entscheidend ist also, ob sich die technischen Rahmenbedingen so weit verändern, dass unabhängig von der Aufstellung des Unternehmens eine Neuentwicklung durch den Markt bedingt wird. In [Cohe03, S.6] wird diesbezüglich von der Refresh Rate gesprochen, die ein Indiz dafür ist, wie häufig Komponenten und Funktionen den Marktbedingungen angepasst werden müssen. Ein Indiz für einen stabilen Markt ist eine hohe Zahl an Standardfunktionen übergreifend über alle Produkte. Mobilfunkgeräte oder MP3 Player sind Beispiele für Produkte, die sich gerade von großen jährlichen Veränderungen hin zu einer stärkeren Standardisierung verändern. Während in den letzten Jahren fast alle Produkte dieses Bereichs komplette Neuentwicklungen waren, findet man heute immer häufiger Produkte, bei denen man die „Vorfahren“ erkennen kann. Bei denen man also eher von einer Evolution als einer Revolution in der Entwicklung sprechen kann. Hier könnte man insbesondere das iPhone von Apple als Beispiel nennen, das bei seiner Einführung eine neue Produktkategorie begründet hat. Aufgrund des großen Markterfolgs ist inzwischen ein großes Angebot an unterschiedlichen Software-Plattformen entstanden, die ähnliche Funktionalitäten anbieten.[69] Der Neuigkeitsgrad innerhalb dieser Klasse hat dabei abgenommen. Bspw. können Kunden der ersten iPhone Generation einen Großteil der neuen Funktionen ebenso nutzen, da diese mittels SW realisiert worden sind, allein auf etwaige zusätzliche HW, wie z.B. eine Kamera, oder generell eine höhere Rechenleistung muss verzichtet werden. Die tatsächlichen funktionalen Unterschiede sind jedoch gering.

Für die Automobilindustrie besteht aufgrund der Produktpolitik mit sich zwar aktuell verkürzenden, aber im Vergleich zu Elektronikprodukten im Heimbereich immer noch sehr langen Produktlebenszyklen auf Fahrzeugmodellebene eine besondere Herausforderung. Die

[69] Zu den Betriebsystemen, die weitgehend die iPhone Bedienung über Touchscreen und Finger übernommen haben, gehören Android, WebOS, Maemo, die alle auf einem Linux Kernel basieren, und die etablierteren OS wie Blackberry, Symbian OS und Windows Mobile [vgl. Barc09, S. 86ff.].

Veränderungen je Generation fallen durch die langen Laufzeiten entsprechend umfangreich aus. Allerdings bestehen hier je nach Produktklasse erhebliche Unterschiede. Während die Änderungen im Bereich Infotainment entsprechend groß ausfallen, sind die Anpassungen z.B. im Bereich Motorelektronik eher durch veränderte rechtliche Rahmenbedingungen bestimmt und fallen entsprechend geringer aus. Eine zweite Unterscheidungsebene besteht hinsichtlich der Wertschöpfungskette. Für die OEMs besteht aufgrund der begrenzten Anzahl an neuen Modellen die Herausforderung eher in einer geringeren Häufigkeit von Anpassungen, die allerdings im Durchschnitt einen größeren Umfang haben. Für die Zulieferer hingegen gilt es eher eine größere Zahl an neuen Produkten zu realisieren. Die Unterschiede je Produkt fallen dagegen im Durchschnitt eher geringer aus.

Für die eigentliche Bewertung sollen die Unternehmen den Grad der durchschnittlichen Veränderung als Prozentanteil der Bestandteile eines Produktes angeben, der je Generation verändert wird, basierend auf einer bestmöglichen Umsetzung und bedingt durch sich ändernde technische Bedingungen, Markt- und Kundenanforderungen. Unterschiede sollten sich hauptsächlich durch die oben genannten beiden Differenzierungsebenen, Produktklasse und Wertschöpfungskette, ergeben. Gleichzeitig soll eine Gewichtung nach der Bedeutung der Funktionen erfolgen. Dabei genügt eine sehr grobe Abschätzung. Dies spiegelt sich auch in der Beschreibung der Kategorien 1 und 2 wider, die lediglich verbal ohne prozentuale Angabe erfolgt. So ist für die niedrigste Einstufung entscheidend, dass jede Generation beinahe eine komplette Neuentwicklung ist. Für die zweite Kategorie können zwar einige Kernkomponenten weiterverwendet werden, aber der Großteil ist immer noch neu zu entwickeln. Bis zu dieser Stelle gibt es über die Einstufung als Ausschlusskriterium eine negative Bewertung. Eine solch starke Veränderung der Anforderungen erschwert die Einführung einer Produktlinie erheblich und macht sie in den meisten Fällen sogar unwirtschaftlich. Von einer weiteren Verfolgung der Pläne wird dementsprechend abgeraten.

Ab Kategorie 3, die immerhin noch eine durchschnittliche Veränderung der Produkte von ca. 50% annimmt und damit die besondere Situation in der Automobilindustrie mit größeren Veränderungen je Produktgeneration aufgrund der vergleichsweise langen Produktlebenszyklen berücksichtigt, wird eine ökonomische Perspektive des Ansatzes vermutet. Höhere Kategorien betrachten die Veränderungen je Generation im Vergleich zu den vorhergehenden eher als Evolution denn als Revolution. Die höchste Einstufung wird schließlich bei einer Veränderung von weniger als 20% vorgenommen, da dies am oberen

Ende der üblichen jährlichen Wartungskosten für Domain Assets liegt [vgl. Cohe03, S.12]. Aufgrund der genannten Besonderheiten der Automobilindustrie stellt dieser ansonsten eher durchschnittliche Wert hier eine außergewöhnliche Leistung dar. Ein solcher Wert dürfte wahrscheinlich eher von Zulieferern als von OEMs erreicht werden. Die Kategorie wird insgesamt mit dem Wert 4 gewichtet.

Frage	Einstufungen 1	2	3	4	5
Wie groß ist der Grad der Veränderung (Neuigkeit) je Generation oder neuem Produkt, wobei wesentliche Features stärker berücksichtigt werden sollten als kosmetische Veränderungen?	Jede Generation ist fast eine komplette Neuentwicklung.	Einige wenige grundlegende Komponenten können weiterverwendet werden; größtenteils jedoch neu.	Neuentwicklung und Verwendung bestehender Komponenten hält sich ungefähr das Gleichgewicht	Verändeurng ist eher als eine Evolution und nicht als eine Revolution zu verstehen, es finden sinnvolle Erweiterungen in geringem Umfang statt (zwischen 20 und 50%)	Nur sehr geringe Veränderungen (weniger als 20% der Komponenten betroffen), hauptsächlich Anpassungen, keine komplette Neuentwicklung.

Tabelle 8 – Kategorie Produkt, Frage 4

5.1.1.1.5. Maximales Wiederverwendungspotenzial bei auf der SWPL basierenden Produkten

Diese Frage knüpft an Frage 4 zum Neuigkeitsgrad an, projiziert die Antwort allerdings ausgehend von der Stabilität der betrachteten Produkte auf die zukünftige Produktlinie und die spezifische Unternehmenssituation. Gefragt ist, welche Wiederverwendungsrate das Unternehmen im Optimalfall aufgrund der Marktgegebenheiten mit einer Produktlinie für möglich hält. "Eine stabile Domäne ist Voraussetzung dafür, dass Investitionen in die Wiederverwendung auch für zukünftige Produktentwicklungen genutzt werden können" [Schm04, S. 50]. Sollte keine oder lediglich eine kleine gemeinsame Domain für realisierbar angesehen werden, wird erneut von einer Realisierung einer Produktlinie abgeraten. "Reuse will not have a substantial impact on development unless a sizable portion of your product is derived from reusable assets" [SPC93, S. A-6]. Zumindest ein relativ großer gemeinsamer Kern kritischer Funktionen wird als notwendige Bedingung angesehen (Ausschlusskriterium). Ab Einstufung 3 gilt dies als erfüllt.

Um eine fundierte Aussage zum Wiederverwendungspotenzial treffen zu können, bedarf es neben der Einschätzung der internen Umsetzungsmöglichkeiten zusätzlich einer Abschätzung der zukünftig kundenrelevanten Funktionen. Je besser ein Unternehmen die zukünftigen Marktveränderungen und Trends durch ein gutes Kundenverständnis (z.B. durch umfangreiche Marktanalysen und Kundenpannels) abschätzen und vorhersehen kann, desto größer ist die Wahrscheinlichkeit eine für diese Funktionen passende Architektur zu designen und damit die Stabilität der SWPL zu erhöhen.

Die Beschreibungen der einzelnen Stufen sind bewusst allgemein gehalten und verzichten auf konkrete Prozentangaben, da zu dem angepeilten frühen Zeitpunkt der Anwendung des Fragebogens eine detailliertere Beantwortung kaum möglich sein wird. Ganz im Gegensatz dazu wird dieses Thema beim PLTP Modell wesentlich detaillierter behandelt. Gleich mehrere Practise Areas beschäftigen sich damit, u.a. Mining Existing Assets, Make/Buy/Mine Commission Analysis. Für die Aufgabe dieses Fragebogens ist jedoch eine sehr allgemeine Beantwortung ausreichend, um den Analyseaufwand gering zu halten. Eine detaillierte Bewertung kann zu einem späteren Zeitpunkt erfolgen.

Die Gewichtung erfolgt mit Faktor 4 und nicht wie eigentlich zu erwarten mit dem Höchstwert von 5, um diese Ungenauigkeit zu reflektieren, die zwangsläufig in der Antwort enthalten ist. Dennoch stellt die Frage ein wichtiges Hygienekriterium dar, denn damit wird sichergestellt, dass sich das Unternehmen ausreichend mit dem Thema SWPL auseinandergesetzt hat, um zumindest eine ungefähre Abschätzung über die zukünftigen Realisierungsmöglichkeiten abgeben zu können.

Frage	Einstufungen 1	2	3	4	5
Wenn sie die maximale Wiederverwendungsrate abschätzen sollten, wie weit wäre eine Standardisierung von Bestandteilen aufgrund der Marktgegebeneheiten wahrscheinlich möglich?	Bei jedem Produkt handelt es sich weitgehend um eine Einzelentwicklung. Daher wird es quasi keine gleiche Basis geben können.	Die Produkte sind alle eigenständig, haben jedoch eine kleine gemeinsame Basis; dies wird sich aufgrund der sehr spezifischen Kundenanforderungen in Zukunft auch nicht ändern.	Es wird einen realtiv großen gemeinsamen Kern bei den Produkten der Produktfamilie geben, aber auch einen signifikanten Teil unterschiedlicher Funktionen je Produkt.	Zum Kunden hin werden sich die Produkte entsprechend der Positionierung teilweise deutlich unterscheiden, aber technisch sind sie weitgehend identisch	Die gemeinsame Basis (Core Assets) der Produktfamilie wird ein limitiertes Set enthalten, das alle Produkte abdecken wird. Der individuelle Anteil der Produkte wird (bis auf wenige Ausnahmefälle) sehr gering.

Tabelle 9 – Kategorie Produkt, Frage 5

5.1.1.2.Produktentwicklung / Prozesse

Das PLTP Modell sieht für die Prozessreife eine eigene Practice Area vor im Bereich Technical Management, nämlich Process Discipline. Im BAPO-Modell wird diese Kategorie stark an die Bewertungsmethodik der Prozessreife nach CMMI angelegt, da dies den de-facto Standard für die Prozessreife darstellt und sämtliche Aspekte weitgehend abdeckt. Diesem Ansatz wird auch hier gefolgt, weshalb die Anzahl der Fragen in dieser Kategorie im Vergleich zum vorherigen Kapitel reduziert ist. Um den Erfassungsaufwand zu reduzieren, kann ein bestehendes CMMI Assessment übernommen werden. Eine weitere Frage greift das Thema Requirements Engineering explizit noch mal auf, obwohl dies bei CMMI bereits berücksichtigt ist. Allerdings kommt diesem Punkt bei der Realisierung einer SWPL eine so hohe Bedeutung zu, dass dies zusätzlich als Einzelkriterium betrachtet und individuell

bewertet werden sollte. Die pauschale Berücksichtigung im Rahmen von CMMI reicht für eine SWPL Anwendung nicht aus.

5.1.1.2.1. Prozessreife des Unternehmens

Unternehmen weisen in der Regel stark unterschiedliche Reifegrade bei ihren Prozessen auf. "Lacking maturity in software engineering can affect the success of software product line engineering even more seriously than technological matters" [Pohl05, S. 17]. Je besser ein Unternehmen allerdings seine Prozesse organisiert hat, desto eher kann angenommen werden, dass auch die Einführung einer Produktfamilie das Unternehmen nicht vor eine unlösbare Herausforderung stellen wird. Geeignete Mittel zur Messung der Prozessreife sind standardisierte Referenzmodelle, wie z.B. CMMI[70]. Hat das Unternehmen bereits eine CMMI Einstufung durchführen lassen, ergibt sich die Antwort für diese Frage direkt aus dem Ergebnis der Einstufung. Die CMMI Level entsprechen den Kategorien. Aber auch ähnliche Modelle wie SPICE, das insbesondere im Automobilbereich aufgrund der Automotive SPICE Variante stärker verbreitet ist [vgl. Teil A Kapitel 2.2.4.2], lassen sich aufgrund der großen Ähnlichkeit leicht übertragen. Unternehmen ohne eine standardisierte Einstufung können auch eine Selbsteinschätzung vornehmen, wobei diese allerdings ein Grundverständnis der CMMI Methodik und eine realistische Einschätzung voraussetzt. Aufgrund der klaren inhaltlichen Abgrenzung der einzelnen Assessment-Stufen erscheint eine grobe Selbsteinschätzung möglich und für die Zielsetzung des Fragebogens ausreichend. Schließlich sollte einem Unternehmen die Bedeutung einer hohen Prozessreife vor Beginn der Umsetzung einer SWPL zumindest bewusst sein. Und es ist davon auszugehen, dass die meisten Verantwortlichen eine relativ gute Vorstellung von ihrer eigenen Unternehmensreife haben.

Die Gewichtung erfolgt mit dem Faktor 5. Allerdings handelt es sich nicht um ein Ausschlusskriterium. Grund dafür ist, dass ein Unternehmen mit hohem CMMI Level zwar Vorteile durch gut organisierte Prozesse hat und es auch nachgewiesen hat, Verbesserungen durchführen zu können. Aber bei der Übertragbarkeit auf die Umsetzung einer Produktlinie bestehen doch ganz neue Herausforderungen. Als Beispiel sei daran erinnert, dass eine CMMI Level 5 Zertifizierung für SW Outsourcing Anbieter inzwischen quasi zur Pflicht geworden ist, insbesondere im asiatischen Umfeld.[71] Die Erfahrung zeigt, dass diese Einstufung alleine

---

[70] Für Details siehe Teil A Kapitel 2.2.4.

[71] Im Jahre 2003 stammten von den 80 Software-Unternehmen, die weltweit mit CMM Level 5 bewertet wurden, 60 aus Indien [Scha05b, S. 6].

keinen Erfolg bei der Umsetzung eines Projektes garantiert. Die starke Prozessfokussierung der CMMI Evaluation deutet zwar auf hohe Effizienz, diese garantiert jedoch keine hohe Effektivität.

Frage	Einstufungen 1	2	3	4	5
Haben sie schon mal ein CMMI Assessment durchführen lassen? Wenn ja, welches Level haben sie erreicht? (Ansonsten auch vergleichbarer Wert, wie z.B. SPICE; oder auch Selbsteinschätzung)	CMMI Level 1 Keine festgelegten Prozesse, Vorgehen wird häufig adhoc abgestimmt	CMMI Level 2 Einige wenige Prozesse sind standardisiert, vielfach aber nicht genau kontrolliert	CMMI Level 3 ausreichendes Prozessmanagement etabliert	CMMI Level 4 alle Aktivitäten sind über standardisierte Prozesse abgedeckt, die klar kommuniziert und auch kontrolliert werden	CMMI Level 5 wie Level 4 zusätzlich ergänzt um ein exzellentes Verbesserungs-management

Tabelle 10 – Kategorie Prozesse, Frage 1

5.1.1.2.2. Festlegung der Anforderungen an neue Produkte

Die frühzeitige Festlegung der Anforderungen an neue Produkte ist ein wichtiger Schritt bei der Entwicklung von SW. Dafür hat sich der Begriff Requirements Engineering etabliert. Fehler in diesem Bereich werden meist erst spät in der weiteren Entwicklung erkannt und sind nur mit hohem Aufwand zu beheben. Ein gutes Requirements Engineering ist daher zur erfolgreichen Erstellung von SW in allen Bereichen notwendig. Hinsichtlich der eingesetzten Werkzeuge bei der SW Erstellung hat das Requirements Engineering den größten Einfluss auf den Erfolg eines Projektes [Stan01, S. 10]. Bei der Realisierung einer SWPL kommt diesem Schritt eine noch größere Bedeutung zu, da die Komponenten der Asset Base über einen längeren Zeitraum verwendet werden sollen. Nachträgliche Anpassungen erzeugen einen deutlich höheren Aufwand. Ebenso wäre es ökonomisch unsinnig, Funktionen zu entwickeln, die später nicht benötigt werden. "Thorough requirements engineering comprising the identification of commonality and variability is a major prerequisite for software product line engineering" [Pohl05, S.17]. Aus diesem Grund findet sich dieses Element ebenfalls im PLTP Modell als eigene und wichtige Practice Area im ersten Bereich, dem Software Engineering. Erweitert wird dies zusätzlich durch das Scoping im Technical Management.

Meist ist „das Anforderungsmanagement nicht unbedingt automobilspezifisch ausgeprägt. Dies kann auch am Einsatz von Standardwerkzeugen zum Anforderungsmanagement in Fahrzeugprojekten erkannt werden“ [Schä06, S. 136]. Es wird in der Literatur allerdings ergänzend darauf hingewiesen, dass einige automobilspezifische Eigenschaften, wie z.B. die standortübergreifende Zusammenarbeit oder die langen Produktlebenszyklen berücksichtigt werden sollten [vgl. Schä06, S. 136]. Die vorhandenen Standardmethoden decken diese jedoch vollständig ab und können daher verwendet werden.

Sollten die Anforderungen allein von einer Abteilung an die Entwicklung vorgegeben werden, dann entspräche dies Kategorie 1. Bei Level 3 besteht bereits die Forderung nach einem weitgehend standardisierten Prozess, allerdings noch unter Einbeziehung eines beschränkten Kreises an Verantwortlichen. Mit Level 4 erfolgt eine breitere Abstimmung der Anforderungen und damit einhergehend aufgrund der im Automobilbereich vielfach dezentralen Struktur der Unternehmen die Forderung nach einer standortübergreifenden Standardisierung der Prozesse. Und erst wenn die Anforderungen über einen vollständig standardisierten, dokumentierten und standortübergreifenden Prozess mit allen Stakeholdern abgestimmt werden, erfolgt die Höchstwertung. Die Gewichtung ist Faktor 4. Trotz der hohen Bedeutung erfolgt nicht die höchste Einstufung, da die Antwort lediglich die aktuelle Situation im Unternehmen widerspiegelt. Vor der Einführung einer SWPL wäre allein schon für die Definition der Architektur ein Requirements Engineering notwendig. Unternehmen, die bereist darüber verfügen, haben es einfacher. Die Einführung vor der SWPL Realisierung erscheint jedoch möglich. Die Frage wird daher nicht als Ausschlusskriterium betrachtet.

Frage	Einstufungen 1	2	3	4	5
Wie werden bei ihnen die Produktrequirements festgelegt?	Die Requirements werden vom Marketing vorgegeben, ohne dass es dafür ein standardisiertes Verfahren geben würde. Die Entwicklungsabteilung wird weitgehend mit fertigen high-level Requirements konfrontiert.	Marketing entwickelt die Requirements, stimmt diese aber nachträglich mit der Entwicklung ab; kein Standardprozess etabliert	Es gibt einen Standardprozess zur Aufstellung der Requirements; hauptsächlich eine Abteilung ist mit der Erstellung alleine beauftragt	Wie bei 3 allerdings sind mehrere Abteilungen in der Entwicklung der Requirements involviert	Über einen standardisierten Prozess werden die Requirements frühzeitig mit allen Stakeholdern abgestimmt. Spätere Anpassungen treten dadurch nur sehr selten auf.

Tabelle 11 – Kategorie Prozesse, Frage 2

5.1.1.2.3. Qualität der bestehenden Produkte

„The most expensive and time-consuming work of software development is the work of finding bugs and fixing them“ [Jone07, S. 10]. So wird im Allgemeinen von einer zu erwartenden Fehlerzahl von 5 pro Function Point ausgegangen. Dieser Wert berücksichtigt Fehler bei den Anforderungen, der Kodierung, der Dokumentation und Bad-Fixes, also die unbeabsichtige Erzeugung von neuen Fehlern bei der Beseitigung von bekannten Fehlern. Dies verdeutlicht erneut die Wichtigkeit der Verwendung des FP-Ansatzes als Basis für die Aufwandsbestimmung, da reine LoC Ansätze den Aufwand allein aufgrund von Fehlern außerhalb der Kodierung nicht oder nur schwer berücksichtigen können [Jone07, S. 114 f.].

Wie bereits in Kapitel 4 ausgeführt, sind SWPL nicht in erster Linie ein Ansatz zur Verbesserung der Qualität. Vielmehr ist ein ausreichendes Qualitätsniveau eine notwendige Voraussetzung, da ansonsten der zusätzliche Aufwand für die Produkttests bei Komponenten

mit einer hohen Fehlerzahl den Vorteil durch eine höhere Produktivität aufgrund der Wiederverwendung überkompensieren kann. Wenn aber vollständig überprüfte Komponenten häufiger verwendet werden, steigert dies die Qualität der Produkte und der darin enthaltenen SW im Besonderen. Zusätzlich kann die gesteigerte Produktivität für zusätzliche Integrationstest genutzt wird, wodurch sich das Qualitätsniveau im Allgemeinen weiter erhöhen sollte.

Da bei der Einführung von SWPL häufig das bisherige Produktangebot die Basis für die zukünftige SWPL Assets bildet, ist eine Betrachtung der aktuell erreichten Qualität bei der SW Erstellung notwendig. Als Maßstab für die Einstufung wird nicht auf die Fehler pro FP zurückgegriffen, da anzunehmen ist, dass viele Unternehmen aus der Automobilindustrie nicht über einen entsprechenden Wert verfügen. Zur möglichst einfachen Beantwortung des Fragebogens wird auf standardmäßige KPIs in der Welt der SW Erstellung zurückgegriffen. Typische Grenzwerte für die Anzahl von Fehlern finden sich bspw. bei Capers Jones [Jone07, S. 10]. So wird für die durchschnittliche Fehlerbeseitigung beim Verkaufsstart ein Wert von ca. 85% erwartet. Dieser sagt aus, dass bis zum Marktstart ca. 85% aller in der SW enthaltenen Fehler erkannt und beseitigt worden sind. Der Wert basiert auf einer Durchschnittsbetrachtung der amerikanischen SW Industrie und ist für eine Verwendung im Rahmen dieses Fragebogens für die Automobilindustrie leicht nach oben anzupassen, da im Vergleich zur PC-basierten SW Industrie an zahlreiche Komponenten deutlich höhere Qualitätsanforderungen gestellt werden, z.B. bei sicherheitskritischen Systemen [vgl. Schä06, S. 141 ff.]. Außerdem ist aktuell ein Update der Systeme noch nicht vergleichbar einfach sicherzustellen, wie z.B. heute durch die Sicherheitsupdates beim Windows Betriebssystem. Etwas eingeschränkt wird diese Betrachtung allerdings durch die Tatsache, dass inzwischen immer mehr Standardanwendungen aus dem Unterhaltungsbereich ihren Weg in das Automobil finden. Unter Berücksichtigung der genannten Faktoren wird für eine mittlere Einstufung eine Fehlerbeseitigung von ca. 90% zum Marktstart angenommen.

Eine reine Betrachtung der Fehlerfreiheit reicht allerdings insbesondere für die Automobilindustrie nicht aus, denn eine hohe Endqualität kann neben einer guten Entwicklung auch durch nachträgliche und damit sehr aufwändige und kostspielige Testverfahren und Korrekturen erreicht werden. Da ein solches Vorgehen weder effektiv (falsche oder unzureichende Testcases, usw.) noch effizient (Test zu spät im Entwicklungsprojekt, usw.) ist, wird dies über zwei zusätzliche KPIs überprüft. Die Kosten

für die Fehleridentifikation und –Beseitigung sollten gemessen am Gesamtprojektaufwand bei ca. 40% liegen, der Anteil der für die Test insgesamt aufgewendete Zeit sollte ungefähr 35% betragen. Als Basis für diese Grenzwerte wurden erneut die Durchschnittswerte nach Capers Jones für die amerikanische SW Industrie herangezogen und jeweils analog zum Anteil der Fehlerbeseitigung um 5% nach oben angepasst, um die Besonderheiten der Automobilindustrie entsprechend zu berücksichtigen.

Zusätzlich werden die KPIs mit Aussagen zu den verwendeten Prozessen zur Qualitätskontrolle kombiniert, um ein aussagekräftiges Bild über die aktuelle Ausgangslage des Unternehmens beim Thema SW-Qualität zu erhalten. Bei Stufe 1 ist keine Qualitätskontrolle vorhanden. Stufe 2 setzt bereits eine minimale Kontrolle voraus, die sich allerdings durch ein Fehlen von Pre-Test Inspektionen, unzureichende Fehlernachverfolgung und Tests durch Generalisten auszeichnet. In der höchsten Stufe werden im Unternehmen formale Design- und Code-Überprüfungen durchgeführt, es findet eine formale und automatisierte Fehlernachverfolgung statt, und das formale Testen erfolgt durch ausgebildete Testspezialisten, während Unit Tests durch die Programmierer selbst bereits beim Erstellen des Codes erfolgen.

Die generelle Aussage bei dieser Frage ist, dass Unternehmen, die ihre Qualität bereits gut beherrschen, bei der Einführung einer SWPL Vorteile haben. Sie können ihre existierenden Komponenten leichter wiederverwenden und auch die höheren Qualitätsanforderungen bei der Erstellung von Domain-Assets stellen eine geringere Herausforderung dar. Im PLTP Modell wird das Thema Qualität im Bereich Testing ebenfalls als eigene Practice Area behandelt. Die Gewichtung erfolgt mit Faktor 4, wodurch die hohe Bedeutung der Qualität im Rahmen eines SWPL Projektes gezeigt wird. Weiterhin handelt es sich um ein Ausschlusskriterium, da Unternehmen, die nicht zumindest ein durchschnittliches Qualitätsniveau erreichen, sich zunächst auf dieses Thema konzentrieren sollten bevor sie sich mit einem weiteren komplexen Thema wie SWPL beschäftigen.

Frage	Einstufungen 1	2	3	4	5
Wie hoch ist die Qualität, die sie derzeit bei der SW Erstellung mit ihren aktuellen Prozessen erreichen?	Es findet keine systematische Qualitätskontrolle statt. KPIs werden nicht ermittelt	Es erfolgt eine minimale Qualitätskontrolle, bei der aber zahlreiche wichtige Elemente fehlen oder verbessert werden können. Es können zumindest 2 der genannten 3 KPIs ermittelt werden, die allerdings unterhalb der Durchschnittswerte liegen.	Es erfolgt eine durchschnittliche Qualitätskontrolle und die relevanten KPIs liegen im Durchschnitt (Fehlerbeseitigung ca. 90%, Kosten Fehlerbeseitigung ca. 40% und Zeitaufwand ca. 30%)	Die Qualitätskontrolle berücksichtigt die wichtigsten Elemente, könnte aber noch weiter verbessert werden. Die aktuellen KPIs liegen aber bereits deutlich über dem Durchschnitt.	Die Qualitätskontrolle berücksichtigt alle wichtigen Elemente, wie z.B. Tests durch spezialisierte Mitarbeiter. Und die KPIs liegen deutlich über dem Durchschnitt.

Tabelle 12 – Kategorie Prozesse, Frage 3

5.1.1.3.Organisation

5.1.1.3.1. Größe der SW Entwicklungsabteilung

Im Allgemeinen korreliert die Bedeutung, die SW für ein Unternehmen spielt, zu einem gewissen Grad mit der Zahl der SW Entwickler. Zwar kann ein Unternehmen einen Großteil der Entwicklungsarbeit nach außen geben, aber die aktuellen Trends in der Automobilindustrie zeigen, dass die OEMs eher versuchen, eigene Kapazitäten im Bereich SW-Entwicklung aufzubauen. Ein Grund hierfür dürfte die zunehmende Bedeutung von SW im Automobil sein und die Erwartung, dass SW sich in Zukunft als wichtiger Faktor für die Produktdifferenzierung etablieren könnte. Am Beispiel von BMW zeigt sich dieser Trend z.B. am Aufbau eigener SW-Kompetenzcenter (Car IT und Softwarehaus) oder die Mitgliedschaft in allen wichtigen Standardisierungsgremien (z.B. Genivi oder Autosar).

Für ein Unternehmen mit einer sehr geringen Zahl von SW-Entwicklern ist es deshalb fraglich, ob eine SWPL sinnvoll eingesetzt werden kann. Eventuell bietet sich eine SWPL tatsächlich an, allerdings wäre ein Zulieferer davon viel stärker betroffen, da diese bisher den Großteil der Entwicklungsarbeit übernommen haben. Außerdem wäre für den OEM in diesem Fall aufgrund des hohen Outsourcing-Anteils nicht sichergestellt, dass er über die notwendigen Kompetenzen verfügt, eine SWPL sinnvoll einzusetzen oder zumindest zu steuern. Die Verantwortung hierfür dürfte wohl eher beim Zulieferer liegen. Die Abfrage der Entwicklerzahl dient also auch der Sicherung, dass sich alle relevanten Unternehmen mit der Idee der SWPL beschäftigen.

Die Frage ist ein Ausschlusskriterium und ergibt bei einer Einstufung von 1 oder 2 eine negative Empfehlung. Die Gewichtung erfolgt mit Faktor 3 und damit durchschnittlich. Dies ist dadurch begründet, dass die tatsächliche Größe der Entwicklungsabteilung für die Umsetzung einer SWPL weniger bedeutend ist. Entscheidend ist, dass eine bestimmte Mindestgröße nicht unterschritten wird, da ansonsten davon ausgegangen werden sollte, dass SW nicht im Fokus der entwickelten Produkte steht. Allerdings kann es Ausnahmen geben, wie z.B. beim Deutschen Zentrum für Luft- und Raumfahrtforschung (DLR), wo die Methodik bereits getestet worden ist. Aufgrund der besonderen Rahmenbedingungen einer eher universitär organisierten Einrichtung bestand die Möglichkeit, trotz einer Anzahl von 5

FTEs[72] erfolgreich eine SWPL im Automobilbereich zu erstellen. Unterstützt wurde dies durch den Umstand, dass keine marktreifen Produkte, sondern Testumgebungen für die Erprobung neuer Funktionen im Automobil erstellt werden. Die genauen Details finden sich in Kapitel 6.1.

Frage	Einstufungen 1	2	3	4	5
Wie viele Personen (gemesse als FTEs) arbeiten bei ihnen intern in der Software-Entwicklungsabteilung?	Weniger als 5	Zwischen 5 und 10	Mehr als 10, aber maximal 20	Mehr als 20, aber maximal 50	Mehr als 50

Tabelle 13 – Kategorie Organisation, Frage 1

5.1.1.3.2. Ausrichtung der SW Entwicklung

Jedes Unternehmen hat eine, wenn auch implizite, Struktur. Diese kann für bestimmte Vorhaben unterstützend wirken oder hinderlich sein. Augrund der mit der Einführung einer SWPL verbundenen zahlreichen Veränderungen hinsichtlich Verantwortlichkeiten ist auch eine notwendige Anpassung der bestehenden Organisationsstruktur zu erwarten. Im PLTP Modell wird dies mittels der Practice Area „Structuring the Organisation“ im Bereich Organisational Management berücksichtigt. Daher widmet auch dieser Fragebogen dem Thema eine eigene Frage mit SW Entwicklung Fokus.

Die generelle Entscheidung, ob eine Projektorientierung oder eine Featureorientierung für die Erstellung von SW besser geeignet ist, soll hier nicht getroffen werden. Für die Anwendung einer SWPL ergibt sich jedoch eine klare Empfehlung. Aufgrund der Zerlegung der eigentlichen Produkte in einen allgemein einheitlichen und einen individuellen Teil wird eine reine Projektorientierung erschwert. Zur Entwicklung und auch zur Wartung der Elemente der Asset Base ist eine Featureorientierung besser geeignet, da Überschneidungen vermieden werden. Für ein Unternehmen, das seine Produkte bisher allein als Einzelentwicklungen im Rahmen von Projekten erstellt hat, wird der Umstieg schwieriger sein als für ein Unternehmen, das seine Einzelprodukte bereits nach einzelnen Funktionen aufgeteilt entwickelt hat. Die bisherigen Ausführungen sind für kleine und mittlere Unternehmen, also insbesondere im Zuliefererbereich, anwendbar. Bei großen Unternehmen, z.B. OEMs oder Tier-1 Zulieferern, ergeben sich zusätzliche Schwierigkeiten, da oft mehrere Produkte gleichzeitig entwickelt werden und daher sowohl Produkt- als auch Fachwissen benötigt wird.

[72] FTE steht für Full Time Equivalent und bezieht sich nicht auf die reine Anzahl an beschäftigten Personen, sondern rechnet die Anzahl entsprechend ihrer prozentualen Beschäftigung um, inkl. Urlaubs- und Krankheitstage. Eine Halbzeitkraft würde entsprechend mit 0,5 FTEs berücksichtigt.

Aus diesem Grund haben sich hier vielfach Matrixorganisationen durchgesetzt [Böck04b, S. 33]. Funktionswissen wird häufig innerhalb von Kompetenzzentren gesammelt. Für die Beantwortung der Frage bedeutet dies, dass große Unternehmen mit einer Matrixorganisation angeben sollten, ob der Fokus stärker auf einer Projekt- oder Featureorientierung liegt.

Grundsätzlich ist es nicht ausgeschlossen, den Schritt von einer Projektorientierung hin zur Featureorientierung zeitnah zu bewerkstelligen. Die Komplexität der Umsetzung einer SWPL erhöht sich allerdings. Aus diesem Grund handelt es sich nicht um ein Ausschlusskriterium und die Gewichtung ist mit Faktor 3 durchschnittlich. In Kapitel 2.2.1.2 wurde bereits das Beispiel zur Motorsteuerungssoftware bei Siemens VDO kurz erwähnt. Der Zulieferer hat bei seiner Umstellung auf eine SWPL genau diese organisatorische Veränderung erfolgreich vollzogen.

Frage	Einstufungen 1	2	3	4	5
Wie würden sie die aktuelle Struktur bei der Produktentwicklung am ehesten beschreiben?	Reine Projektorientierung; es gibt keine oder kaum Schnittstellen zwischen den einzelnen Projekten oder übergreifende Teams	Projektorientierung mit vereinzelten übergreifenden Teams; ebenso gibt es mehrer Schnittstellen zwischen den Projektteams	Intensive Zusammenarbeit einzelner Teams etabliert; für einige Komponenten ist auch bereits ein projektunabhängiges Team eingesetzt (Featureorientierung)	Featureorientierung weitgehend umgesetzt	Reine Featureorientierung bei der Produkt-/ Komponentenentwicklung vollständig etabliert

Tabelle 14 – Kategorie Organisation, Frage 2

5.1.1.3.3. Anpassungsfähigkeit

"Produktlinienentwicklung setzt im Allgemeinen organisatorische Veränderungen voraus" [Schm04, S. 50]. Selbst für Unternehmen, die bereits eine Funktionsorientierung im Entwicklungsbereich umgesetzt haben, bedeutet die Einführung einer SWPL eine große organisatorische Umstellung. "Die Erfahrung zeigt, dass eine nicht zu den Prozessen und der Art des Engineering passende Organisationsstruktur den Erfolg einer Softwarefirma zunichte machen kann" [Böck04b, S. 26]. Es müssen Verantwortliche für das Management der Asset Base und die Kontrolle der gesamten SWPL gefunden werden. Ein Indiz für die Anpassungsfähigkeit gibt die Beantwortung der Frage: "How well does the organisation manage the distinct responsibilities and relationships occurring in the software family engineering: undifferentiated vs. specialised roles" [Lind04, S.124]. Für eine Bewertung verschiedener Organisationsformen hinsichtlich ihrer Eignung für eine SWPL siehe [Böck04b, S. 30ff.]. Die Anpassung der Struktur eines Unternehmens ist eine komplexe Aufgabe, die nicht unterschätzt werden sollte. Je einfacher ein Unternehmen aus der historischen Erfahrung heraus mit organisatorischen Veränderungen umgehen kann, desto leichter wird sich eine SWPL umsetzten lassen. Insbesondere für die OEMs ist eine solche

organisatorische Anpassung eine große Herausforderung. Für kleinere Zulieferer dürfte dies hingegen leichter möglich sein. Dies zeigt sich auch an den hier genannten Praxisbeispielen, die sich alle auf kleinere Unternehmen beziehen. Einzige Ausnahme ist BMW mit dem 150% Ansatz, bei dem mittels einer Funktionsübererfüllung versucht wird, die Wiederverwendung von Gleichteilen (insb. Steuergeräten mit identischer SW) zu erhöhen.

Die notwendigen organisatorischen Veränderungen können so gravierend sein, dass diese Frage als Ausschlusskriterium eingestuft ist. Die Einstufungen 1 und 2 führen zu einer negativen Empfehlung. Bevor weitere Überlegungen hinsichtlich SWPL angestellt werden, ist anzuraten, dieses Hemmnis zu beseitigen. Die Gewichtung mit Faktor 5 spiegelt die hohe Bedeutung der Veränderungsfähigkeit wider. Zusätzlich genügt es nicht, ein Mindestlevel zu erreichen, sondern die Einführung einer SWPL wird umso einfacher möglich sein, je offener ein Unternehmen gegenüber Veränderungen eingestellt ist.

Frage	Einstufungen 1	2	3	4	5
Wie einfach ist es, in ihrem Unternehmen Verantwortungen/Strukturen zu ändern?	äußerst schwierig bis unmöglich; es ist mit unüberbrückbaren Widerständen zu rechnen	schwierig; die Erfahrung zeigt, dass an diesem Thema schon des öfteren Projekte gescheitert sind	durchaus möglich, aber mit erheblichen Widerständen ist zu rechnen, die jedoch in der Vergangenheit immer überbrückt werden konnten	relativ leicht, obwohl natürlich nicht immer alle der selben Meinung sind. Neuerungen werden gründlich geprüft, statt sie grundsätzlich abzulehnen	ziemlich einfach, da eine Kultur der Veränderung gelebt wird und Veränderung nicht grundsätzlich als etwas schlechtes gesehen wird.

Tabelle 15 – Kategorie Organisation, Frage 3

5.1.1.3.4. Interne Unterstützung

Wenn die Entscheidung für eine SWPL gefallen ist, dann kommt der tatsächlichen Umsetzung und der Institutionalisierung der mit einer SWPL verbundenen Ansätze eine wichtige Bedeutung zu. Ein Unternehmen sollte die Rahmenbedingungen einer SWPL verinnerlichen. Wichtig ist im gesamten Unternehmen transparent zu machen, dass die Einführung einer SWPL eine grundsätzliche Veränderung der Art und Weise, wie Produkte erstellt werden, bedeutet. Im PLTP Modell wird dies innerhalb der Practice Area „Launching and Institutionalizing“ behandelt. Diese Kategorie deckt weitgehend sowohl die vorherige als auch diese Frage ab. Im Rahmen des Fragebogens liegt der Fokus jedoch stärker auf den aktuell vorhandenen Rahmenbedingungen, die eine mögliche Umsetzung einer SWPL behindern oder unterstützen könnten. Daher werden hier insbesondere die Aspekte der Anpassungsfähigkeit und der tatsächlichen internen Unterstützung für die SWPL Einführung betrachtet.

Die Frage nach der internen Unterstützung knüpft daher eng an die vorherige an, beleuchtet aber nicht die Einstellung zur Veränderung der Unternehmung als ganzes, sondern fokussiert

auf einzelne Personen, die eine SWPL Einführung behindern könnten. Daher ist diese Frage weitgehend automobilunspezifisch. Vielmehr ist sie für alle Unternehmen relevant, die eine SWPL planen oder in Erwägung ziehen. Sollten im Vorfeld einer SWPL Entscheidung Vorbehalte gegen diese Maßnahme bekannt sein, dann ist damit offensiv umzugehen. Es sollte versucht werden, die Gründe für die Vorbehalte zu entkräften. Möglicherweise sind die Einwände auch begründet und sollten in die Überlegungen mit einbezogen werden. Schwierig wird es, wenn zu Beginn nicht klar ist, dass es überhaupt ablehnende Haltungen gibt. Um dies möglichst zu verhindern, sollte offen diskutiert werden und alle Mitarbeiter ermuntert werden, ihre Meinungen mitzuteilen.

Aber selbst ohne tatsächliche Gegner kann es notwendig sein, grundsätzlich von SWPL abzuraten. Dies ist dann der Fall, wenn es keine aktive Unterstützung des Vorhabens durch das Management gibt. "Such a project can only be successful when top management shows strong commitment to the project and is actively involved" [Wapp00, S. 80]. Aufgrund der zahlreichen damit verbundenen Änderungen in den Arbeitsabläufen dauert es eine Zeit, bis sich der Grundgedanke wirklich etabliert hat und alle Mitarbeiter dies voll unterstützen. Eine Anweisung zur Umsetzung allein reicht dazu nicht aus. Nur wenn die Einführung der SWPL offensichtlich durch hochrangige Verantwortliche unterstützt wird, kann diese kritische erste Phase erfolgreich sein. In einer Untersuchung der Standish Group aus dem Jahr 2001 zeigte sich, dass Executive Support der wichtigste Erfolgsfaktor für erfolgreiche SW Projekte ist [Stan01, S. 4].

Die genannten Gründe führen zu einer Gewichtung mit Faktor 5 und zur Einstufung dieser Frage als Ausschlusskriterium. Erst ab Einstufung 3 erfolgt keine direkte negative Empfehlung.

Frage	Einstufungen 1	2	3	4	5
Wird die Entwicklung einer SWPL intern grundsätzlich unterstützt oder gibt es möglicherweise Gegner?	Es gibt zahlreiche Gegner, insbesondere auch in höheren Managementebenen	Es gibt zahlreiche Gegner, was aber vielfach auf zu geringe Information zurückzuführen ist	Es gibt einige Gegner, aber zumindest das obere Management unterstützt das Vorhaben komplett	alle Managementebenen unterstützen das Vorhaben, und auch sonst gibt es nur geringe Kritik, die wahrscheinlich durch zusätzliche Information gelöst werden kann	Komplett unterstützt, es gibt keine Gegner

Tabelle 16 – Kategorie Organisation, Frage 4

5.1.1.3.5. Umsetzung einer HW Modularisierung

Unternehmen der Automobilindustrie, insbesondere OEMs, sind führend in der Umsetzung von Produktlinien im Hardwarebereich (siehe Teil A Kapitel 1.1). Grund für die Anwendung

des Konzepts der Modularisierung und der Plattformen für einzelne Komponenten waren Zielsetzungen, die vergleichbar sind mit den Zielen, die durch eine SWPL erreicht werden sollen, z.B. eine Reduktion der Entwicklungszeit, eine Erhöhung der Flexibilität, eine erhöhte Standardisierung und Produktivität [vgl. Jung05, S. 100ff.]. Für die Bewertung der SWPL-Fähigkeit eines Unternehmens kann der Umfang der Modularisierung oder Plattformstrategie als grobes Indiz dienen, wie stark ein Unternehmen das Ziel der strategischen Wiederverwendung tatsächlich verfolgen wird. U.a. kann daran die Unterstützung durch das Management und die grundsätzliche Fähigkeit zur organisatorischen Anpassung erkannt werden. Eine erfolgreich umgesetzte Modularisierungs- oder Plattformstrategie lässt demnach vermuten, dass auch eine SWPL gute Umsetzungschancen hätte. Natürlich hängt der spätere Erfolg einer SWPL nicht allein vom Vorhandensein einer HW-basierten Produktlinie ab. Dazu sind die damit verbundenen Anforderungen einfach zu unterschiedlich.

Allerdings ist es das Ziel des Fragebogens möglichst viele Indizien zusammenzutragen, um eine fundierte Aussage über die SWPL-Fähigkeit treffen zu können. Für OEMs dürfte diese Anforderung häufiger erfüllt sein. Bei Zulieferern ergeben sich jedoch basierend auf der persönlichen Erfahrung des Verfassers weniger Beispielfälle. In vielen Fällen sind die Anforderungen der OEMs zu unterschiedlich, oder die OEMs behindern die Modularisierung und Wiederverwendung von Komponenten bewusst, um ihre bestehenden Wettbewerbsvorteile und Differenzierungsmerkmale gegenüber dem Wettbewerb nicht zu verlieren.

Die Gewichtung erfolgt mit einem Faktor 2. Grund ist, dass die Einführung einer HWPL eher als Indiz anzusehen ist, selbst aber keine Garantie für einen Erfolg bedeutet, da sich die beiden Projekte inhaltlich stark unterscheiden.

Frage	Einstufungen 1	2	3	4	5
Haben sie bereits außerhalb der SW-Entwicklung eine Modularisierung oder Plattform-Strategie verfolgt?	Eine Modularisierungs- und/oder Plattformstrategie wurde geplant/umgesetzt. Diese scheiterte jedoch und wird nicht weiter verfolgt.	Eine Modularisierungs- und/oder Plattformstrategie wurde bisher nicht geplant/umgesetzt.	Eine Modularisierungs- und/oder Plattformstrategie wurde für einen kleinen Bereich bereits erfolgreich umgesetzt.	Eine Modularisierungs- und/oder Plattformstrategie wurde für mehrere Bereiche bereits erfolgreich umgesetzt. Die Strategie wird bald/aktuell auf größere Komponenten ausgeweitet.	Wichtige HW-Komponenten sind heute bereits als Produktlinie gestaltet. Es wurden signifikante Erfolge damit erzielt und die Strategie wird weiter verfolgt

Tabelle 17 – Kategorie Organisation, Frage 5

5.1.1.4.Architektur

Sowohl im BAPO-Modell als auch bei der PLTP Analyse wird die Architektur als eines der entscheidenden Erfolgsfaktoren für eine SWPL betrachtet. Grundsätzlich spielt die

Architektur bei jedem SW-Projekt eine entscheidene Rolle, da diese bereits zu Beginn festgelegt wird und etwaige Fehler zu diesem Zeitpunkt im Allgemeinen erhebliche Auswirkungen auf die spätere Entwicklung haben werden. Für SWPL ist die Bedeutung nochmals größer, da SWPL Architekturen eine Doppelfunktion erfüllen: einerseits die allgemeine SWPL Architektur und die Architektur der einzelnen Produkte, die aus der ersten abgeleitet werden, z.B. mittels Variationspunkten.

Im Rahmen dieses Fragebogens spielt die spätere SWPL Architektur allerdings zunächst nur eine untergeordnete Bedeutung, da diese mit großer Wahrscheinlichkeit nicht einmal in Ansätzen festgelegt ist. Im späteren Verlauf sollte eine entsprechende Bewertung der Architektur vorgenommen werden. Dazu können dann die bekannten Bewertungsmodelle herangezogen werden. Ist es allerdings das Ziel, lediglich die Eignung eines Unternehmens für die Einführung einer SWPL zu ermitteln, ist es sinnvoller sich zunächst auf die vorhandenen und verwertbaren Informationen zu konzentrieren. Diese befassen sich stärker mit der aktuellen Architektur und evtl. vorhandenen ersten Erfahrungen der Unternehmen mit Wiederverwendung von SW-Komponenten. In diese Richtung zielen die folgenden Fragen und unterscheiden sich damit deutlich von den Ansätzen der bekannten Bewertungsmodelle, die sich stärker auf die tatsächliche SWPL Architektur fokussieren.

5.1.1.4.1. Anteil von Standardkomponenten

Bei Unternehmen mit mehreren Produkten am Markt ist anzunehmen, dass es einen gewissen Anteil an Standardkomponenten gibt. Vielfach wurde dieser bisher nicht im Sinne einer strategischen Wiederverwendung betrachtet. "Bereits existierende Artefakte können die Produktlinien begünstigen, wenn sie die Basis für den Aufbau der Plattform zur Verfügung stellen" [Schm04, S. 50]. Je höher der Anteil an Standardkomponenten oder existierender Artefakte ist, desto größer sind die Chancen für eine erfolgreiche Einführung einer SWPL. Die Asset Base kann größer sein und auch die Erstellung der Domain Komponenten selbst dürfte weniger Aufwand verursachen. Allerdings besteht auch ein gewisses Risiko, wenn die existierenden Komponenten "schwer erfüllbare Kompatibilitätsanforderungen mit sich bringen" [Schm04, S. 50]. In einem solchen Fall kann jedoch von einer Nutzung bestehender Komponenten abgesehen und eine Neuentwicklung durchgeführt werden.

In der Automobilindustrie wurde in der Vergangenheit SW oft als Resultat des Designs der elektronischen Komponenten betrachtet. Dies wurde u.a. auch durch die Besonderheiten, wie geringe Rechenkapazität, geringer Speicher und hardwarenahe Programmierung gefördert.

Aufgrund der zunehmenden Bedeutung der SW im automobilen Bereich ist seit einigen Jahren eine Verhaltensänderung festzustellen. Einige Unternehmen, insbesondere kleinere Zulieferer, haben die sich daraus ergebenden Möglichkeiten früh erkannt und auch entsprechend bereits erste Erfolge durch stärkere Standardisierung und Modularisierung erzielen können. Viele Unternehmen stehen jedoch noch am Beginn dieses Umwandlungsprozess. Daher dürfte diese Frage eine große Streuung an Antworten ergeben.

Auf die Vorgabe von exakten Einstufungsgrenzen wurde verzichtet, da es keine entsprechenden Werte in der Literatur gibt. Die persönlichen Erfahrungen des Autors reichen ebenfalls nicht aus, um entsprechende Werte ableiten zu können. Daher sollten die Unternehmen anhand ihrer eigenen Erfahrung eine Einstufung vornehmen. Dies soll in erster Linie ein stärkeres Bewusstsein für die Bedeutung standardisierter Komponenten im Rahmen einer SWPL erzeugen. Gleichzeitig soll aber auch die Basis für die spätere Detailanalyse der aktuellen SW Komponenten und deren Potenzial für eine mögliche Wiederverwendung gelegt werden. Die Gewichtung mit Faktor 3 ist durchschnittlich.

Frage	Einstufungen 1	2	3	4	5
Standardkomponenten im Softwarebereich bei heutigen Produkten?	Sehr gering	gering	mittel	hoch	sehr hoch

Tabelle 18 – Kategorie Architektur, Frage 1

5.1.1.4.2. Aktuelle Produktarchitektur

"Die systematische Entwicklung, Pflege und Dokumentation einer Architektur [spielt] für jedes (Software-)System eine herausragende Rolle. Im Kontext von Produktlinien steigt die Bedeutung dieser Rolle weiter an, da nur mittels der Architektur über alle Systeme einer Produktlinie kommuniziert werden kann" [Baye04b, S. 166]. Für den erfolgreichen Einsatz einer SWPL ist es demnach eine wesentliche Voraussetzung, dass die Entwicklung auf einer geeigneten Referenzarchitektur stattfindet [Hein02]. Entwurfsfehler in dieser Architektur haben im Allgemeinen Auswirkungen auf alle später zu entwickelnden Produkte und resultieren damit entweder in einem niedrigeren Wiederverwendungsgrad oder kostspieligen Anpassungen der SWPL [Thie04, S. 81]. Unternehmen, die bereits heute eine standardisierte, produktübergreifende SW Architektur einsetzen, werden bei der Einführung einer SWPL Vorteile haben, da sie einerseits auf die vorhandenen Komponenten zum Aufbau der Asset Base zurückgreifen können. Zusätzlich verfügen sie über Erfahrung bei der strategischen Wiederverwendung von SW.

Aufgrund der zunehmenden Komplexität der SW Komponenten im Automobil sind die OEMs und Zulieferer gezwungen sich stärker mit dem Thema Wiederverwendung von SW zu beschäftigen, da ansonsten die Anforderungen des Marktes nach neuen Funktionen und Verbesserungen nicht zu einem realistischen Preis und in einem geeigneten Zeitrahmen realisiert werden können. Daher werden Aspekte aus der klassischen SW-Entwicklung zum Thema Wiederverwendung übernommen, wie z.B. objektorientierte Programmierung [vgl. Schä06, S. 220]. Je stärker sich ein Unternehmen bereits mit diesen Themen beschäftigte, desto besser ist es auch auf den Einsatz einer SWPL vorbereitet, wobei hier ein wesentlich höheres Niveau der Wiederverwendung das Ziel ist. Es ist zu erwarten, dass auch hier große Unterschiede zwischen den einzelnen Unternehmen auftreten werden.

Die Frage ist als Ausschlusskriterium ausgelegt, um einen Sonderfall abzudecken. Falls das Unternehmen bereits früher versucht haben sollte, eine einheitliche SW Architektur einzuführen, dabei aber gescheitert ist, dann ist es sinnvoll, die Gründe dafür vor einer weiteren Analyse nochmals im Detail zu besprechen und zu klären, ob sich die Rahmenbedingungen inzwischen geändert haben. Ab Einstufung 2, also selbst wenn ein Unternehmen bisher keine Anstrengungen zur Vereinheitlichung der Architektur unternommen hat, erfolgt keine direkte negative Empfehlung mehr. Der Gewichtungsfaktor ist mit 3 durchschnittlich. Schließlich kann das aktuelle Produktportfolio eine einheitliche SW Architektur bisher unmöglich gemacht haben.

Frage	Einstufungen 1	2	3	4	5
Benutzen sie bereits heute eine standardisierte produktübergreifende Softwarearchitektur?	Wir haben es mal probiert, leider nicht erfolgreich	Nein, bisher nicht probiert	Für einige ausgewählte Komponenten im Einsatz	Ja, in weiten Teilen	Ja, komplett

Tabelle 19 – Kategorie Architektur, Frage 2

5.1.1.4.3. Integration neuer Funktionen

Requirements Engineering versucht, die Anforderungen an eine SW möglichst früh und vollständig zu erfassen, um spätere kostenintensive Änderungen und Anpassungen zu reduzieren. Dennoch kann aus verschiedenen Gründen die Integration neuer Funktionen in eine bestehende Architektur notwendig sein. Auch ohne den Einsatz einer SWPL ist es daher sinnvoll, solche Anpassungen vorzusehen, da ansonsten der Aufwand für eine Integration so hoch ausfallen könnte, dass eine Neuentwicklung vorzuziehen ist. Bei der Einführung einer SWPL kommt dieser Integrationsfähigkeit des Unternehmens eine noch größere Bedeutung zu. Schließlich wird es in den meisten Fällen notwendig sein, einen Teil der bereits

existierenden SW anzupassen und weiter zu nutzen. Unternehmen, die eine hohe Reife bei der Umsetzung von SWPL erreicht haben, beachten daher im Rahmen des SW Variability Managements folgendes: "the complete software life cycle is taken into account, determining when to introduce and bind which variability, and which mechanism should be used" [Lind04, S. 118]. Dadurch wird ein Höchstmaß an Flexibilität erreicht. Außerdem hat die Architektur bei diesen Unternehmen einen hohen Einfluss auf die eigentliche Entwicklung erreicht, denn nur bei einer durchgängig gültigen und viele Anwendungsbereiche abdeckenden Architektur können neue Funktionen schnell eingefügt werden.

Die vorhergehende Frage zielte allein auf das Vorhandensein einer produktübergreifenden Systemarchitektur ab. Eine hohe Einstufung führt im Allgemeinen auch zu einer einfachen Integration neuer Funktionen. Die Korrelation ist jedoch nur einseitig, denn selbst ohne eine solche Architektur kann ein Unternehmen in der Lage sein, neue Funktionen mit geringem Aufwand zu integrieren, z.B. bei einer soliden Zukunftsplanung und einem relativ stabilen technischen Umfeld. Daher erfasst diese Frage eine wichtige zusätzliche Eigenschaft. Die Gewichtung erfolgt mit dem Faktor 3.

Frage	Einstufungen 1	2	3	4	5
Wieviel Aufwand ist heute mit der Integration eines neuen Features durchschnittlich verbunden?	Fast alle neuen Features führen zu umfangreichen Änderungen an zahlreichen Stellen im Code; eine Aufwandsschätzung ist kaum möglich		Einige Features erzeugen hohen Aufwand, insbesondere, wenn diese nicht vorherzusehen gewesen sind; der Großteil ist jedoch einfach umsetzbar		Umsetzung ist in den meisten Fällen sehr einfach aufgrund der offenen Architektur und der guten Planung

Tabelle 20 – Kategorie Architektur, Frage 3

5.1.1.5.Business drivers and manners

Abschließend werden Faktoren berücksichtigt, die stärker auf das Unternehmen und den Markt fokussiert sind. Hier geht es in erster Linie nicht um die technische Realisierbarkeit, sondern um die sonstigen Rahmenbedingungen, die den Erfolg einer SWPL beeinflussen können. Im Rahmen des BAPO Modells wird dieser Aspekt ebenfall explizit im Bereich Business behandelt, während sich im PLTP Ansatz nur einige Hinweise auf diese Einflussfaktoren finden. Der Fokus dieses Ansatzes liegt stärker auf der tatsächlichen Realisierung als auf der frühzeitigen Bestimmung der Erfolgsaussichten.

Die Betrachtung der Unternehmensbeteiligungen und möglicher Auswirkungen bei verzögerten Neuprodukteinführungen im Rahmen der SWPL Umsetzung sind neue Aspekte, die bisher in keinem Bewertungsansatz zu finden sind. Allerdings können diese insbesondere

zu einem sehr frühen Bewertungszeitpunkt wichtige Indizien liefern, ob die Idee der SWPL weiterverfolgt werden sollte.

5.1.1.5.1. Unternehmensziele

Eine SWPL ist eine langfristige Unternehmensentscheidung. Um diese erfolgreich umsetzen zu können, ist ebenfalls eine konstante und langfristig ausgerichtete Unternehmenszielsetzung erforderlich. Die Erfahrung zeigt, dass Unternehmen mit einer strategischen Planung für die SWPL Entwicklung, klaren Roadmaps und stabilen langfristigen Zielen eine höhere Reife für SWPL aufweisen [Lind04, S. 114]. Ein Unternehmen, das innerhalb kurzer Zeit seine grundsätzliche Ausrichtung ändert, wird eher Probleme haben, die anfänglich festgelegten Ziele einer SWPL auch langfristig zu verfolgen. Damit verbunden ist eine sinkende Unterstützung für die SWPL. Dies wäre ökonomisch unsinnig, da sich eine SWPL aufgrund der Ausgestaltung als Investitionsobjekt erst über die Laufzeit rentiert.

"When a software family engineering approach is chosen, the business goals should be in alignment with the software family engineering goals in order to improve the competitiveness" [Lind04, S. 114]. Bei der Frage nach Unternehmenszielen ist anzugeben, ob es eine klare Ausrichtung auf eine der drei möglichen Zielrichtungen, Innovation, Kosten oder Qualität gibt. Während so gut wie alle Unternehmen im Automobilbereich diese Frage aus Gesamtunternehmenssicht wahrscheinlich leicht und eindeutig beantworten können, könnte sich durch den Fokus auf die reine SW-Perspektive vielfach eine ambivalente Sichtweise ergeben. Bei vielen OEMs und Zulieferern wird aktuell über die zukünftige Aufstellung hinsichtlich der SW-Entwicklung entschieden. Während Premium Hersteller eher dazu tendieren, wie z.B. BMW, selbst verstärkt eigene Ressourcen aufzubauen und die SW als eine zukünftige Kernkompetenz sehen, kooperieren Massenhersteller immer stärker mit großen SW-Herstellern, um insbesondere im Infotainmentbereich leistungsfähige Produkte anbieten zu können.[73] Bei den Zulieferern ergibt sich aufgrund der neu in den Markt eintretenden Wettbewerber eine gewisse Unsicherheit hinsichtlich des weiteren Vorgehens. Aktuell ist allerdings ein verstärkter Aufbau von SW-Entwicklungsressourcen zu beobachten.

Die Gewichtung erfolgt mit einem Faktor 2 etwas unterdurchschnittlich, da die Unternehmensziele keine direkte Auswirkung auf die technische Realisierung der SWPL

[73] Als Beispiel für eine Kooperation von OEMs und großen SW-Herstellern seien Ford und Microsoft genannt, die beim Entertainment- und Navigationssystem SYNC eng zusammenarbeiten und dadurch deutlich kostengünstigere Infotainmentprodukte anbieten können [vgl. Mora07].

haben. Sie beeinflussen lediglich den tatsächlich erzielbaren Erfolg einer eingeführten SWPL. Die Frage ist kein Ausschlusskriterium, da SWPL auch bei unklarer oder sich ändernder Zielvorgabe möglich sind. Allerdings erhöht sich das finanzielle Risiko, da bei einer frühzeitigen Einstellung aufgrund geänderter Ziele der Großteil der Erträge aus der SWPL Investition nicht mehr zu realisieren wäre.

Frage	Einstufungen 1	2	3	4	5
Welche Ziele verfolgt das Unternehmen maßgeblich?	Es sind keine langfristigen Ziele festgelegt. Bei der individuellen Produktentscheidung bilden eher kurzfristige oder/und produktindividuelle Überlegungen die Entscheidungsbasis.		Das Unternehmen hat eine klare Ausrichtung auf eine der drei möglichen Zielrichtungen, Innovation, Kosten oder Qualität, die auch weitgehend verfolgt wird.		Das Unternehmen hat eine klare Ausrichtung auf eine der drei möglichen Zielrichtungen, Innovation, Kosten oder Qualität. Diese wird konsequent bei allen Projekten/Produkten verfolgt.

Tabelle 21 – Kategorie Business drivers, Frage 1

5.1.1.5.2. Verzögerung bei Produktentwicklung

Die Einführung einer SWPL kann inkrementell, also in kleinen Schritten und damit quasi parallel zur bisherigen Produktentwicklung, verlaufen. Dies ist insbesondere dann sinnvoll, wenn der Erfolg einer SWPL als unsicher eingestuft wird, da dieses Vorgehen kaum Anfangsinvestitionen erfordert. Der Nachteil besteht in der geringen Nutzbarkeit einer solchen SWPL und der langen Dauer bis zur wirklichen Einsetzbarkeit, verbunden mit der "Gefahr, dass die Plattform häufig überarbeitet werden muss, um weiterhin eine geeignete Basis für neue Produkte zu bieten" [Knau04, S. 186]. Daher wird von diesem Vorgehen, mit Ausnahme einiger weniger Anwendungsbereiche, eher abgeraten.

Wenn sich ein Unternehmen allerdings entschließt, eine SWPL als big-bang und damit in einem großen Schritt einzuführen, ist dies meist unausweichlich mit einer Verzögerung der bisher geplanten zukünftigen Produktreleases verbunden. Schließlich erfordert der Aufbau der Asset Base einen Großteil der Entwicklungsressourcen des Unternehmens. "Können die notwendigen Investitionen in die Wiederverwendung erbracht werden? Gibt es Konflikte mit Produktentwicklungen?"[Schm04, S.50] sind zu beantwortende Fragen. Daher ist es relevant, ob der Markt eine begrenzte zeitliche Verzögerung neuer Produkte akzeptieren würde. Es wurde exemplarisch eine Zeitspanne von 6 Monaten gewählt. Befindet sich das Unternehmen in einem sehr aktiven Wettbewerb mit zahlreichen neuen Produkten über einen kurzen Zeitraum, dann ist eine SWPL eher ungeeignet. Zumindest eine revolutionäre Einführung erscheint schwierig.

Während sich OEMs mit SWPL meist im Bereich der Vorserienentwicklung beschäftigen und daher zeitliche Beschränkungen eher weniger bedeutend sind, spielt diese Frage insbesondere für die Zulieferer eine sehr wichtige Rolle. Werden die Kunden bereit sein, eine Verzögerung neuer Produkte zu akzeptieren? Andererseits kann eine große Menge an neuen Produkten, die in Zukunft innerhalb kurzer Zeit fertig zu stellen sind, wie im Beispiel von Siemens VDO auch genau zur Einführung einer SWPL führen. Das Unternehmen erkannte, dass mit der ursprünglichen Organisation und Produktentwicklung eine Steigerung der produzierten Motorsteuerungen entsprechend der Marktanforderungen nur bei einer deutlichen Steigerung der Mitarbeiterzahl möglich wäre. Dies wurde allerdings als kaum möglich angesehen. Daher entschied man sich für einen neuen Ansatz bei der SW-Erstellung, um die Kundenwünsche zu erfüllen. Aufgrund des anfänglichen zeitlichen Investments konnte die Neuproduktanzahl im weiteren Verlauf deutlich gesteigert werden. Allerdings musste dieses zeitliche Investment genau geplant werden, um die Auswirkungen auf die aktuellen Produkte so gering wie möglich zu halten. Bei Siemens VDO war dies weitgehend möglich. Je nach individueller Unternehmens- und Wettbewerbssituation kann sich eine andere Bewertung ergeben. Daher sollte sich das Unternehmen mit dieser Frage beschäftigt haben und eine Abschätzung abgeben können.

Der Gewichtungsfaktor ist 4 und zeigt gemeinsam mit der Einstufung als Ausschlusskriterium die hohe Bedeutung dieser Frage. Aufgrund der Möglichkeit einer inkrementellen Einführung ist es dennoch möglich, eine SWPL aufzubauen ohne Einfluss auf die aktuelle Produktentwicklung. Aufgrund der oben genannten Kriterien sollte dies jedoch genau abgewogen werden. Erneut zeigt sich der Charakter der Ausschlusskriterien: eine negative Beantwortung bedeutet kein endgültiges Aus, aber es erinnert daran, dass ein bestimmtes Risiko explizit berücksichtigt werden sollte.

Frage	Einstufungen 1	2	3	4	5
Ist es realistisch, dass der Markt eine Verzögerung der Markteinführung ihrer nächsten Produktgeneration um ca. 6 Monate akzeptiert?	So gut wie ausgeschlossen. Der Wettbewerbsdruck ist so hoch, dass wir uns eine solche Verzögerung nicht leisten können.		Der Markt verändert sich schnell, daher wird eine solche Verzögerung zu erheblichen kurzfristigen Umsatzeinbußen führen, wäre jedoch möglich		Eine Verschiebung der neuen Produkte ist durchaus möglich. Die zu erwartenden Einbußen halten sich in Grenzen.

Tabelle 22 – Kategorie Business drivers, Frage 2

5.1.1.5.3. Unternehmensbeteiligungen

Ein aus eigener Kraft wachsendes Unternehmen hat die beste Kontrolle über die eigene Produktlandschaft. Je stärker fremde Unternehmen und deren Produkte integriert werden

müssen, desto eher ist davon auszugehen, dass hinsichtlich einer standardisierten Architektur Zugeständnisse gemacht werden müssen. In Ausnahmefällen kann die klare Ausrichtung auf Zukäufe zur Erweiterung der Angebotsfamilie allerdings auch positive Effekte haben, z.B. wenn dies die Grundlage für die Einführung einer standardisierten SW Architektur gewesen ist, um die Integrationskosten möglichst gering zu halten.

Grundsätzlich ist jedoch eher davon auszugehen, dass es Unternehmen mit einem geringen Anteil an zu integrierender Fremd-SW einfacher haben werden, eine SWPL aufzubauen, da die Architektur stärker vereinheitlich sein sollte und die mögliche Wiederverwendungsrate der bestehenden SW-Komponenten höher ausfällt.

Diese Frage zielt explizit auf den für die SWPL relevanten Unternehmensbereich ab. Zukäufe in anderen Bereichen sind nicht von Bedeutung. Ziel ist es, zu erkennen, ob evtl. vor der Einführung einer SWPL Integrationsaufgaben für die bestehende SW-Architektur erforderlich sind. Die Beschäftigung mit dem Aufbau der Asset Base vor einer vollständigen Integration und Stabilisierung der Architektur kann langfristige, negative Auswirkungen auf die SWPL haben, da die Asset Base in einem solchen Fall tendenziell kleiner ausfallen wird. Hier ist dringend angeraten, die Optimierung in der korrekten Reihenfolge durchzuführen und erst eine stabile Basis für die Einführung einer SWPL zu schaffen. Es ist zu vermuten, dass diese Frage eher für Zulieferer von Bedeutung ist, da Zukäufe wegen der im Allgemeinen geringeren Unternehmensgröße eine größere Auswirkung haben. Bei OEMs steht aufgrund der weitgehenden Zusammenarbeit mit einer großen Zahl an Zulieferern ohnehin die Integrationskompetenz für SW-Systeme zu einem fertigen Produkt im Mittelpunkt. Und bedingt durch ihre Größe wäre eine SWPL vermutlich auch allein für ein Unternehmen und dessen Produkte ökonomisch sinnvoll, ohne die komplette Integration abzuwarten.

Die Gewichtung erfolgt mit dem Höchstwert 5 und es handelt sich um ein Ausschlusskriterium. Unternehmen mit einer Einstufung von 1 oder 2 sollten überlegen, ob die Einführung einer SWPL derzeit sinnvoll möglich ist, oder ob nicht zunächst eine vollständige Integration der neuen SW erfolgen sollte. Eventuell bietet eine SWPL aber auch die Möglichkeit, den ohnehin notwendigen Aufwand zur Integration mit einer Neuausrichtung der bestehenden SW-Komponenten zu verbinden. Genau diese Unsicherheit sollte im Vorfeld offen diskutiert werden.

Frage	Einstufungen 1	2	3	4	5
Hat ihr Unternehmen in letzter Zeit zahlreiche andere Unternehmen gekauft, oder sind solche Käufe geplant?	Es wurden in den letzten Jahren zahlreiche Fremdfirmen in das Unternehmen integriert. Und auch für die Zukunft ist dies geplant. Dieses Vorgehen hat natürlich zu zahlreichen Veränderungen der Produktlandschaft geführt, die bis heute auch noch nicht vollständig abgeschlossen sind.	Es wurden in den letzten Jahren zahlreiche Fremdfirmen erfolgreich in das Unternehmen integriert. In Zukunft wird dies weiterhin geschehen, allerdings mit reduzierter Geschwindigkeit. Die wichtigsten Lücken im Produktangebot sind geschlossen. Für eine mögliche Erweiterung des Angebots ist geplant, sich weiterhin eher auf Zukäufe als komplette Eigenentwicklungen zu konzentrieren.	Es wurden in den letzten Jahren einige Firmen gekauft und erfolgreich integriert. Der Grund war immer eine sinnvolle Erweiterung der Produktpalette, wobei es sich meist lediglich um Ergänzungen des Angebots handelt.	Es wurden in den letzten Jahren einige wenige Firmen gekauft und erfolgreich integriert. Der Grund war immer eine sinnvolle Erweiterung der Produktpalette. Allerdings handelt es sich dabei (auch in Zukunft) immer um Randprodukte.	Das Unternehmen wächst weitgehend aus eigener Kraft. Wenn es Zukäufe gegeben hat oder geben wird, dann lediglich um wichtige vereinzelte Lücken im Produktangebot zu schließen. Zukäufe sind auf jeden Fall die Ausnahme.

Tabelle 23 – Kategorie Business drivers, Frage 3

5.1.1.5.4. Eigene SW Erstellung

Die letzte Frage ist verbunden mit der Frage nach der Anzahl der Softwareentwickler, setzt den Fokus aber stärker auf die Tätigkeiten, die im Unternehmen selbst zur Erstellung von SW durchgeführt werden. Selbst bei einer großen Anzahl an SW Entwicklern ist es möglich, dass die Entwicklung weitgehend ausgelagert ist und sich das Unternehmen allein auf das Testen und Integrieren konzentriert. Bspw. hat ein IT-Tochterunternehmen eines großen deutschen Automobilherstellers die SW für eine neue Getriebesteuerung zunächst selbst bis zu einem fertigen Testprodukt entwickelt, da das Unternehmen keine Möglichkeit sah, die Anforderungen hinreichend genau zu spezifizieren. Die marktreife Entwicklung wurde an einen Zulieferer abgegeben. Dieser entwickelte das Produkt von Grund auf neu. Der Gesamtaufwand hätte durch eine bessere Spezifizierung der Anforderungen zu Beginn der Entwicklung anstelle einer kompletten Entwicklung einer nutzungsfähigen SW um ca. 25% reduziert werden können.[74]

Unternehmen, die selbst nur einen kleinen Anteil der SW entwickeln oder sich auf Testing und Integration konzentrieren, sollten den Fokus ihrer Arbeit nicht auf die Entwicklung einer SWPL lenken. Denn sie kennen zwar den Markt, aber ohne Kenntnis "der technischen Realisierungsmöglichkeiten für die geplanten Produkte ist es sehr schwer, eine neue Produktlinie aufzusetzen" [John04, S. 154]. Es wäre jedoch möglich, den Zulieferer zur Einführung einer SWPL zu ermutigen. Eventuell besteht die Möglichkeit, die SWPL gemeinsam zu entwickeln. Diese Frage ist ein Ausschlusskriterium, da ein Unternehmen ohne das technische Wissen keine SWPL erfolgreich einführen kann. Liegt der Entwicklungsfokus

[74] Die Daten stammen aus einem Workshop im Jahre 2005 im Rahmen der McKinsey Studie zu embedded Systemen [McKi06, S. 37]. Eine Nennung des Unternehmens ist daher leider nicht möglich.

auf der Integration von Komponenten, sollten die eigentlichen Entwickler bei einer SWPL Entscheidung mit einbezogen werden. Die Gewichtung erfolgt mit Faktor 4.

Frage	Einstufungen 1	2	3	4	5
Wie stellen sie die benötigte Software heute her?	Das Unternehmen stellt selbst gar keine Software her. Lediglich die grundlegenden Requirements und das Integrations-Testing wird vorgegeben bzw. durchgeführt.	Das Unternehmen kauft die Softwarebestandteile so gut wie ausschließlich zu. Nur ein sehr geringer Anteil der Arbeit wird intern erledigt. Dabei handelt es sich hauptsächlich um Testing und Integration. Requirements werden lediglich sehr grob vorgegeben.	Es werden alle verfügbaren Ressourcen genutzt. Eigenentwicklung dort, wo der meiste Nutzen generiert wird, allerdings fehlen klare Regelungen wie hier vorzugehen ist. Teilweise werden Entscheidungen adhoc getroffen. Als Minimum werden aussagekräftige Requirements erstellt und das Componenten- und Integrationstesting durchgeführt.	Wie bei 3 allerdings bestehen klare Regelungen, wie in einem bestimmten Fall vorzugehen ist. Lediglich in begründeten Ausnahmefällen darf von dieser Regelung abgewichen werden.	Als Kombination aus allen möglichen Quellen (COTS, mit Hilfe von externen Kontraktern, Eigenentwicklung). Der Schwerpunkt der eigenen Arbeit liegt dabei auf den wertschöpfenden Tätigkeiten und den Bereichen, in denen das Unternehmen über das notwendige Know-how verfügt. (Requirements Engineering, SW Entwicklung für Kernkompetenzen und das komplette Testing)

Tabelle 24 – Kategorie Business drivers, Frage 4

5.1.2. Bewertung der Produktlinienfähigkeit der Unternehmen

Das Ergebnis der Analyse ist zweigeteilt. Zunächst ergibt sich ein Score (in %) für jede Einzelkategorie und ein finaler Score (in %), der die Eignung eines Unternehmens für die Einführung einer SWPL über alle 5 Bewertungskategorien wiedergibt. Diese Werte werden ermittelt, indem für jede beantwortete Frage die Einstufung mit der Gewichtung multipliziert und über alle Fragen aufsummiert wird. Diese Summe wird schließlich durch die mögliche Höchstpunktzahl dividiert. Für unbeantwortete Fragen wird entweder kein Wert, wenn es sich um eine optionale Frage handelt, oder ein Wert von 0 bei allen anderen Fragen, berücksichtigt. 100% gibt demnach die höchste Übereinstimmung zwischen notwendigen Bedingungen für den Einsatz einer SWPL und der aktuellen individuellen Unternehmenssituation wieder. Das bedeutet jedoch nicht automatisch, dass dieses Unternehmen eine SWPL ökonomisch sinnvoll einsetzen kann, da eine solche Aussage nicht Ziel des Fragebogens ist. Es zeigt allerdings, dass die notwendigen Rahmenbedingungen erfüllt sind und damit gute Chancen auf ein positives Ergebnis einer detaillierten Analyse bestehen.

Bei einer Erfüllung von mehr als 50% der möglichen SWPL Bedingungen wird eine detailliertere Analyse zum Einsatz des Produktlinienansatzes empfohlen. Bei einem Ergebnis von unter 20% wird empfohlen, auf einen Einsatz von Produktlinien in der aktuellen Situation generell zu verzichten, da zu viele Punkte gegen einen erfolgreichen Einsatz sprechen. Für die restlichen Unternehmen mit Werten zwischen 20 und 50% werden individuelle Empfehlungen gegeben, die eine Realisierung des Produktlinienansatzes dennoch ökonomisch möglich

machen könnten. Nach erfolgreicher Umsetzung einiger dieser Hinweise sollte der Fragebogen erneut ausgefüllt werden.

Zusätzlich zum Gesamtergebnis sollten auch die Einzelbewertungen je Kategorie die oben genannten Kriterien erfüllen. Nur in diesem Fall erfolgt eine positive Gesamtbewertung. Dies dient dazu, die Kompensation zwischen den Kategorien einzuschränken. Schwächen eines Unternehmens, die Auswirkungen auf eine mögliche SWPL Realisierung haben, sollten klar aufgedeckt und angesprochen werden. Evtl. bietet es sich an, diese vor der weiteren Beschäftigung mit SWPL zu beseitigen.

Eine weitere Einschränkung ergibt sich durch die Ausschlusskriterien. Sollte eines oder mehrere dieser Kriterien mit einer zu geringen Einstufung bewertet worden sein, ergibt sich das Fazit "ein oder mehrere unbedingt notwendige Kriterien nicht ausreichend erfüllt". Diese Kriterien werden zusätzlich aufgeführt. Nach einer Diskussion der entsprechenden Punkte kann der ebenfalls individuell ermittelte Wert berücksichtigt werden. Sollte dieser über 50% liegen und die Diskussion der Ausschlusskriterien ergeben haben, dass sie in der spezifischen Unternehmenssituation unbedeutend sind[75], wird empfohlen die weiteren Analyseschritte durchzuführen.

Der hauptsächliche Nutzen des Fragebogens liegt in der möglichst frühzeitigen Entdeckung von eindeutig ungeeigneten Anwendungssituationen für SWPL. Da die anschließende Detailanalyse einigen Aufwand erzeugt, sollten Unternehmen diesen Schritt nur dann vornehmen, wenn eine realistische Chance auf ein positives Ergebnis besteht. Um den Aufwand an dieser Stelle der Analyse möglichst gering zu halten, wurde die Anzahl der Fragen bewusst niedrig gehalten. Ebenso wurde der Aufwand zur Beantwortung der Fragen reduziert. So ist es bspw. nicht erforderlich detaillierte Daten zu den geplanten zukünftigen Produkten, wie Umsatz, Stückzahlen, Entwicklungskosten usw., zu ermitteln. Der Test nimmt bewusst einen hohen Anteil falsch positiver Bewertungen in Kauf. Dennoch wird gewährleistet, dass Unternehmen mit unzureichender Aussicht auf eine erfolgreiche SWPL Einführung entsprechend gewarnt werden und dass kritische Erfolgsfaktoren aufgezeigt werden, um eine entsprechende Diskussion anzustoßen. Dadurch werden die Chancen für eine erfolgreiche Anwendung der weiteren Analyseschritte und der späteren SWPL Umsetzung erhöht.

---

[75] Wie es z.B. beim DLR und der Anwendung DOMINION der Fall war, siehe Kapitel 6.1

5.2. Schritt 2: Vereinfachte Berechnung der ökonomischen Konsequenzen des Einsatzes einer Software Produktlinie

In Kapitel 3.6 wurde der SIMPLE Ansatzes bereits vorgestellt. Sein großer Vorteil ist die intuitive Verständlichkeit. Ein Nachteil besteht in der Notwendigkeit der individuellen Ausarbeitung des Modells für die individuelle Unternehmenssituation. SIMPLE stellt nur die Grundstruktur der Analyse zur Verfügung. In Schritt 2 der Analyse wird diese einfache Grundstruktur genutzt, um Unternehmen eine grobe Idee von den finanziellen Auswirkungen einer SWPL zu geben. Hauptziel ist die bessere Erkennung von ungeeigneten Anwendungssituationen. Zu den Aspekten des Fragebogens fügt dieser Schritt die Betrachtung der Wiederverwendungshäufigkeit hinzu. Dieser Schritt ist optional, wird aber insbesondere für Unternehmen empfohlen, deren Eignung im Fragebogen eher durchschnittlich bewertet worden ist.

Die Analyse greift auf das in [Clem05, S. 13f.] beschriebene Szenario 2 zurück: "An organization wishes to choose between building a set of products as a software product line and building them as a set of stand-alone products that do not share core assets." Die Kosteneinsparungen bei Anwendung einer SWPL im Vergleich zu SW-Einzelentwicklungen werden mittels Gleichung 12 bestimmt.

Gleichung 12 – Berechnung der Kostenvorteile einer SWPL gegenüber Einzelprodukten

$$B_{SWPL} = \sum_{i=1}^{n} C_{prod} - \left(C_{org} + C_{cab} + \sum_{i=1}^{n} C_{reuse} \right)$$

wobei B_{SWPL} = Nutzen der SWPL

n = Anzahl der Produkte auf Basis der SWPL

Um den Aufwand so gering wie möglich zu halten, werden für fast alle Variablen die Standardwerte aus der Literatur vorgegeben, jeweils mit einer Bandbreite, aus der das Unternehmen einen individuellen Wert wählen kann. Dieser Wert wird als Standard für alle Produkte der SWPL angenommen. Die einzigen unternehmensspezifischen Daten sind die Anzahl der Produkte der SWPL n und der Grad der Abdeckung sämtlicher Funktionen durch die Domain s_{reuse}. Ein s_{reuse} von 100% bedeutet, dass die gesamte Funktionalität der SWPL als Domain Asset realisiert wird. Für die spätere Produktentwicklung wäre demnach keine Entwicklung von produktindividuellen Komponenten notwendig. Der Anteil der individuell

zu entwickelnden Funktionen kann als Differenz von s_{reuse} zu 100% aufgefasst werden. Außer der Anzahl der Wiederverwendung handelt es sich bei allen Daten um Prozentwerte, da zur Vereinfachung die tatsächliche Größe der SW vernachlässigt werden soll. Dies ist möglich, da der Umfang der SW sowohl die Entwicklung der SWPL als auch die Entwicklung der Einzelprodukte beeinflusst. C_{prod} wird auf 100% gesetzt und C_{cab} wird verstanden als Faktor, der den Mehraufwand einer SW-Entwicklung durch Integration von Variabilität ausdrückt. Aufgrund dieser Annahmen kann Gleichung 12 umgewandelt werden zu:

Gleichung 13 – Vereinfachte Berechnung der Kostenvorteile

$$B_{SWPL} = n * C_{prod} - \left(C_{org} + s_{reuse} * C_{cab} + n * \left((1 - s_{reuse}) * C_{prod} + s_{reuse} * C_{reuse}\right)\right)$$

Folgende Bandbreiten werden für die einzelnen Werte vorgegeben:

- C_{org}: Der Aufwand für die Anpassung der Organisation lässt sich schwer messen. In der Literatur finden sich fast keine entsprechenden Daten. Wenn Zahlen genannt werden, liegen diese um 10% der Entwicklungskosten als Einzelprodukte [vgl. Nóbr08, S. 20]. Anwendern wird daher eine Bandbreite von 5 bis 20% angeboten mit einem voreingestellten Wert von 10%.

- C_{cab}: Dieser Wert misst den Mehraufwand, um eine SW-Komponente wieder verwendbar zu gestalten, z.B. durch Integration von Variabilität. In der Literatur findet sich eine große Bandbreite, zwischen 150 und 900% [vgl. Broo95, S. 224, Jaco97, S. 23 und Trac95, S. xi]. Aufgrund von Erfahrungen aus der Anwendung des vorliegenden Modells erscheint ein Faktor über 4 zu hoch. Daher wird hier eine Bandbreite von 150 bis 400% angeboten mit einem voreingestellten Wert auf 200%.

- C_{reuse}: Die Kosten für die tatsächliche Wiederverwendung von Assets können als Anteil der gesamten Entwicklungskosten ausgedrückt werden. In der Literatur findet sich eine Bandbreite von 5 bis 15%, wobei der obere Wert für opportunistische Wiederverwendung angesetzt wird (vgl. Kapitel 4.4.4). Zu bedenken ist, dass sich dieser Wert nicht auf die Entwicklungskosten als Einzelprodukte, sondern explizit auf die Kosten der wiederverwendbaren Assets bezieht. Daher sollte der aus der Bandbreite zwischen 5 und 15%

gewählte Wert vor der Verwendung in der Formel mit C_{cab} multipliziert werden.

Nützliche Ergebnisse lassen sich ableiten, wenn die Kostenvorteile B_{SWPL} als Funktion in Abhängigkeit von der Anzahl der Wiederverwendungen betrachtet wird. Die Anwendung soll mit einem Beispiel veranschaulicht werden: Ein Unternehmen plant eine SWPL mit 3 zu entwickelnden Produkten. Die organisatorischen Veränderungen (C_{org}) werden mit 15% eingeschätzt. Der Mehraufwand zur Herstellung wieder verwendbarer Komponenten (C_{cab}) wird mit 200% bewertet. Die Kosten für die tatsächliche Wiederverwendung (C_{reuse}) werden mit 20% angesetzt. Dieser Wert berücksichtigt bereits den Faktor 2 für C_{cab}. Der Anteil der durch die Domain abgedeckten Funktionen (s_{reuse}) beträgt 50%. Jeweils eine Hälfte eines Produktes stammt aus der Domain und eine Hälfte wurde individuell entwickelt.

Damit ergibt sich die Funktion für die Kostenvorteile in Abhängigkeit von der Anzahl der Wiederverwendungen:

Gleichung 14 – Kostenvorteile in Abhängigkeit der Anzahl Wiederverwendungen (Bsp.)

$$B_{SWPL}(n) = 0{,}4 * n - 1{,}15$$

Bei n=3 ergibt sich ein Kostenvorteil von knappen 5%. Das Unternehmen sollte eine Erweiterung des SWPL Umfangs oder der Anzahl der Wiederverwendungen in Betracht ziehen. Bei 5 geplanten Produkten ergibt sich ein Vorteil von 85%. Selbst dieser Wert liegt deutlich am unteren Ende der Erwartungen. Generell werden durch die Einführung von SWPL "order-of-magnitude productivity gains" [Clem02, S. xxi] erwartet. Unternehmen, die an dieser Stelle lediglich geringe Vorteile berechnen, sollten ihr Vorhaben grundsätzlich überdenken oder zumindest einige wichtige Parameter anpassen.

Die in SIMPLE vorgeschlagenen Ertragsfunktionen bleiben an dieser Stelle zu Gunsten eines minimalen Analyseaufwands unbeachtet. Dennoch ermöglicht dieser sehr vereinfachte Ansatz den Unternehmen, einen ersten Eindruck von der Bedeutung der Anzahl der geplanten Wiederverwendungen zu erhalten. Analysen mit sehr geringen Wiederverwendungsraten können damit vor Beginn der eigentlichen Analyse und damit vor der Investition eines hohen Zeitaufwands beendet werden.

5.3. Das e-FACTS Modell

Die bisherigen Schritte der Analyse sind als Vorüberlegungen zu verstehen. Ziel ist es, frühzeitig Situationen zu erkennen und zu kommunizieren, bei denen sich aus der weiteren Analyse keine positive Investmententscheidung ergeben würde. Dadurch wird versucht, den Analyseaufwand gering zu halten. Der Fragebogen in Kapitel 5.1 stellt sicher, dass die wichtigsten Rahmenbedingungen für die erfolgreiche Einführung einer SWPL erfüllt sind. Zusätzlich werden durch die Ausschlusskriterien Verbesserungsbereiche aufgezeigt, die die Erfolgsaussichten steigern würden. Aufgrund des geringen Zeitaufwands zur Beantwortung der 20 Fragen wird eine Beantwortung zu Beginn der Analyse empfohlen. In Kapitel 5.2 wird das SIMPLE Modell in einer vereinfachten Form angewendet. Dieser Schritt ist optional, aber für Unternehmen mit einer mittleren Bewertung empfehlenswert, da sich das Risiko einer unnötigen Analyse reduzieren lässt.

In diesem Kapitel wird nun der Schwerpunkt der vorliegenden Arbeit vorgestellt: das e-FACTS Modell. e-FACTS steht für **e**conomic **F**eature Point **A**nalysis with **c**onvenient **t**iming for **S**oftware Product Lines. Das Modell ermöglicht es Unternehmen schnell und leicht verständlich zu einem sehr frühen Zeitpunkt im Entscheidungsprozess über die Umsetzung einer SWPL eine fundierte Aussage zu den ökonomischen Auswirkungen zu treffen. Der Ablauf umfasst 6 Teilschritte, die in Abbildung 25 dargestellt sind und im Folgenden kurz erläutert werden. In den nachfolgenden Kapiteln, deren Gliederung sich am Ablauf des Modells orientiert, erfolgt dann eine detaillierte Beschreibung. Veranschaulicht wird die Handhabung des Modells durch ein Fallbeispiel aus der Automobilindustrie. Anhand dieses Beispiels wird eine komplette e-FACTS Analyse exemplarisch und jeweils am Ende eines Kapitels durchgeführt.

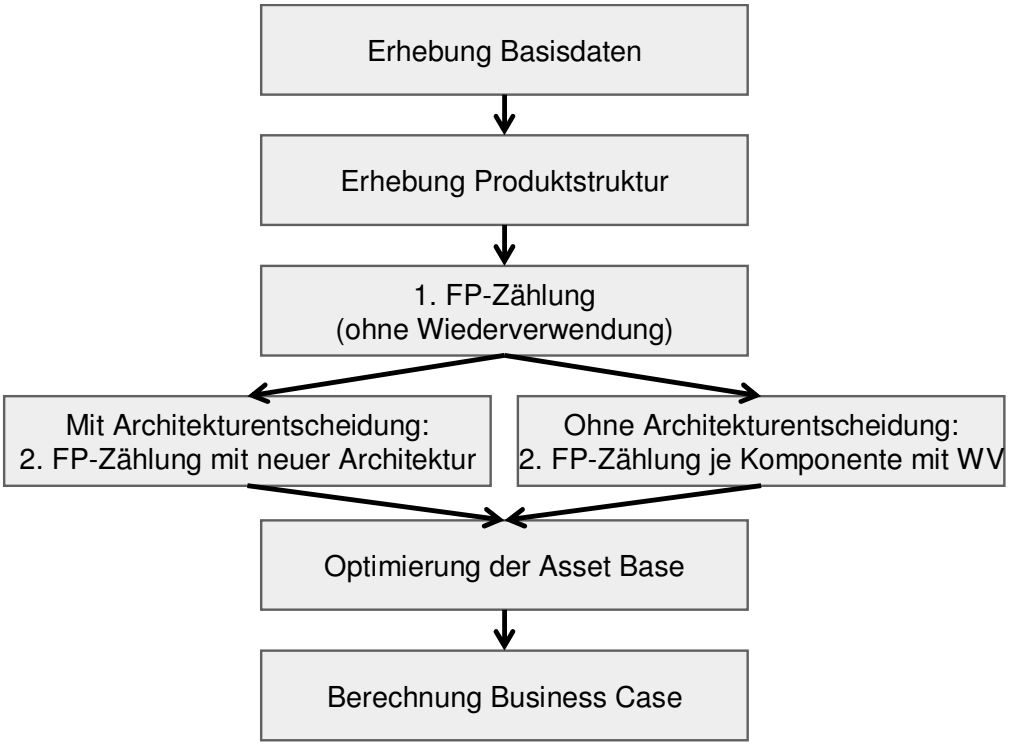

Abbildung 25 – Ablauf des e-FACTS Modells

Im ersten Schritt werden die zur Berechnung des späteren Business Cases notwendigen Daten erhoben. Anschließend erfolgt die Erfassung der geplanten Produktstruktur für die SWPL. In der ersten FP-Zählung werden alle geplanten Produkte mittels der Feature Point Methodik gezählt, als ob es sich um eine reine Einzelentwicklung handeln würde. Damit wird ein Referenzwert für den späteren ökonomischen Vergleich gewonnen. Der nächste Schritt bietet zwei Möglichkeiten: Unternehmen mit Erfahrungen im Bereich der strategischen Wiederverwendung von Software und entsprechenden Vorüberlegungen hinsichtlich einer grundsätzlich neuen Software Architektur für die SWPL können die zweite FP Zählung auf Basis dieser geplanten neuen Architektur durchführen. Die damit erzielten Ergebnisse decken die Wirklichkeit besser ab. Allerdings ist mit der Erstellung einer neuen Architektur ein nicht unerheblicher Aufwand verbunden. Wie bereits in Kapitel 5.1.1.5.2 erläutert sind Entwurfsfehler in dieser Architektur mit langfristigen Folgen für die gesamte SWPL verbunden. Daher bietet e-FACTS die Möglichkeit, die zweite FP-Zählung mit Wiederverwendung ohne Berücksichtigung der SWPL Architektur. Im vorletzten Schritt erfolgt eine Optimierung der Core Assets. Dazu werden die Erstellungsaufwände um Wartungs- und Wiederverwendungskosten erweitert und der Wiederverwendungsnutzen auf Komponentenebene erfasst. Zur Berücksichtigung von finanziell nicht erfassbaren Constraints

findet zusätzlich eine manuelle Evaluierung der automatischen Empfehlung statt. Abschließend erfolgt die Erstellung des Business Cases, der es ermöglicht anhand des Net Present Value (NPV) abzulesen, ob die Einführung der SWPL aus ökonomischer Sicht sinnvoll ist. In Kapitel 5.6 wird das Modell dann erweitert, um z.B. andere Unternehmenszielsetzungen aber auch verschiedene Szenarien zu berücksichtigen.

5.3.1. Beispiel Zugangssysteme im Automobil

Zur besseren Verständlichkeit des e-FACTS Modells wird ein Beispiel eingeführt, das im weiteren Verlauf konsistent genutzt wird. Jeweils im Anschluss an die formale Erklärung wird der entsprechende Arbeitsschritt auf das Beispiel angewendet. Das Anwendungsbeispiel basiert nicht auf einer konkreten Situation bei einem OEM oder Zulieferer, sondern wurde vom Autor selbst auf Basis seiner Erkenntnisse im Rahmen der Erstellung der vorliegenden Arbeit mit großer Sorgfalt erstellt. Daher ist es als fiktiv zu bezeichnen. An einigen Stellen werden zum Zwecke der Übersichtlichkeit und besseren Nachvollziehbarkeit Zusammenhänge aus der Praxis vereinfacht dargestellt. Auch werden aus Platzgründen nicht alle Teilergebnisse in der Arbeit gezeigt.

Das Beispiel beschreibt einen großen Automobilhersteller, der in den nächsten 5 Jahren seine gesamte Fahrzeugpalette erneuern wird. Insgesamt handelt es sich um 5 komplett neue Modelle. Für diese sind jeweils eigene Fahrzeugzugangssysteme zu entwickeln. Ein Fahrzeugzugangssystem wird dabei verstanden als die Gesamtheit aller Systeme, die den Zugang der Nutzer zum Fahrzeug aber auch dessen Funktionen kontrollieren. Eine Abgrenzung des Umfangs solcher Systeme kann über die für deren Nutzung notwendige Autorisierung des Nutzers erfolgen. Alle Systeme, für die eine Autorisierung erforderlich ist, werden im Folgenden mit berücksichtigt. Bei einem Automobil wäre dies der Zugang zum Fahrzeug selbst, die Sicherung gegen unbefugtes Benutzen und weitere Komfortfunktionen, für die eine Autorisierung notwendig ist, wie z.B. das Öffnen der Fenster über den Autoschlüssels. Der Zugriff auf diese Funktionen kann entweder bei direktem Kontakt mit dem Fahrzeug erfolgen, oder mittels externer Schnittstellen davon entfernt. Abbildung 26 zeigt eine komplette Übersicht der betrachteten Funktionen. Die Darstellung erfolgt als Feature Modell in der Notation nach [Czar00]. Ausgefüllte Kreise an den Funktionen selbst bedeuten, dass diese Funktion notwendig ist, im Gegensatz dazu steht ein leerer Kreis für ein optionales Feature. Die Bögen stehen für die Abhängigkeit der Funktionen unterhalb einer übergeordneten Funktion. Ein ausgefüllter Bogen deutet Oder-Features an. Es muss also

mindestens eins der möglichen Features vorhanden sein. Leere Bögen bezeichnen hingegen alternative Features. Von den genannten muss also genau eins ausgewählt und im Produkt vorhanden sein, wenn die übergeordnete Funktion ebenfalls vorhanden ist.

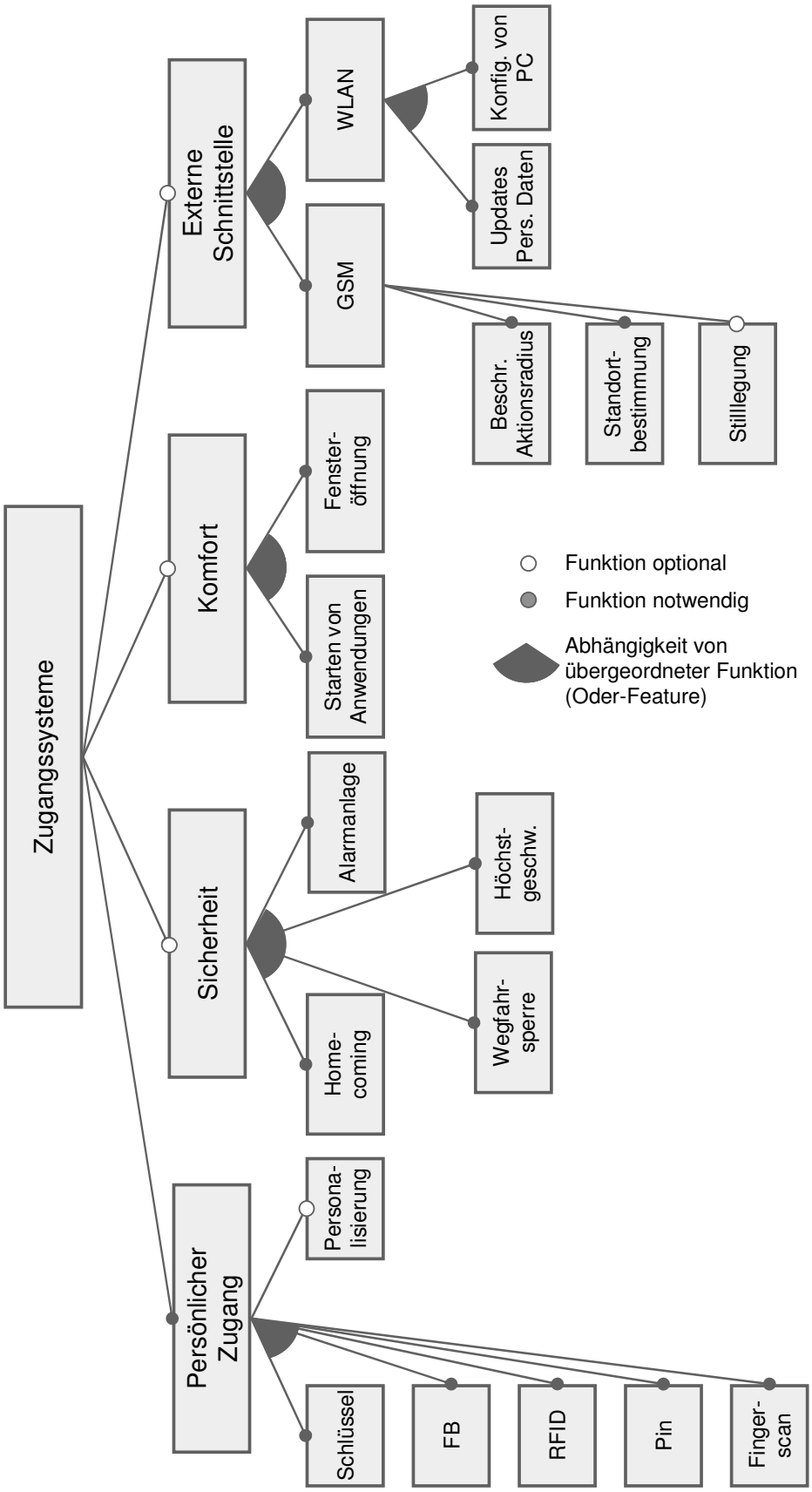

Abbildung 26 – Feature Modell eines automobilen Zugangssystems

Im Folgenden werden die einzelnen Funktionen kurz erklärt:

- **Persönlicher Zugang:** Die Komponente ermöglicht den Zugang zum Fahrzeug selbst, also das Öffnen der Türen. Dies kann alternativ über einen Schlüssel, eine Fernbedienung, eine Chip-Karte mittels RFID oder einen Fingerscan erfolgen. Als Vereinfachung wird im Folgenden Schlüssel als Synonym für das Zugangsmedium verwendet.
- **Personalisierung:** Mittels einer Personalisierung des Schlüssels ist es möglich einige Funktionen speziell für jeden Nutzer individuell anzupassen. Durch die Personalisierung erkennt das System, welcher Fahrer das Fahrzeug gerade nutzt.
- **Alarmanlage:** Eine Alarmanlage sichert das Auto gegen Diebstahl. Moderne Systeme schalten sich beim Abschließen des Fahrzeugs automatisch ein und beim autorisierten Öffnen wieder ab. Der Fahrer kann die Funktion zusätzlich manuell ausschalten.
- **Elektronische Wegfahrsperre:** In Deutschland sind seit 1998 elektronische Wegfahrsperren vorgeschrieben. Es sind unterschiedliche Generationen von Systemen im Einsatz, moderne Sperren funktionieren über einen digitalen Schlüssel, der sich in einem Chip im Schlüssel befindet. Über eine Abfrage des Fahrzeugs wird die Echtheit des Schlüssels überprüft. Erst danach wird das Motorsteuergerät freigegeben und der Wagen kann gestartet werden. Zur Erhöhung der Sicherheit wird inzwischen fast die gesamte Kommunikation verschlüsselt.
- **Homecoming Light:** Bei Dunkelheit können die Scheinwerfer des Fahrzeugs genutzt werden, um die Umgebung auszuleuchten, wenn der Fahrer das Fahrzeug verlassen hat. Diese Funktion ermöglicht es, das Licht beim Öffnen des Fahrzeugs einzuschalten und nach Verriegelung der Türen erst verzögert auszuschalten.
- **Höchstgeschwindigkeit:** Bereits heute ist es möglich, die Höchstgeschwindigkeit elektronisch zu begrenzen. Dies wird bspw. bei Winterreifen genutzt, die eine niedrigere Geschwindigkeitsklasse haben, als die Höchstgeschwindigkeit des Fahrzeugs. In Verbindung mit einer Personalisierung des Schlüssels ist es möglich individuelle Höchstgeschwindigkeiten vorzugeben.
- **Fensteröffnung:** Beim Öffnen des Fahrzeugs können die Fenster mit geöffnet werden. Dies ist bspw. an heißen Sommertagen sinnvoll, um den Innenraum schneller

abzukühlen. Beim Verschließen des Fahrzeugs können alle noch offenen Fenster auf Wunsch geschlossen werden.

- **Starten von Anwendungen:** In modernen Fahrzeugen finden sich zahlreiche Funktionen, die ähnlich wie ein PC zunächst hochgefahren werden müssen und nicht sofort zur Verfügung stehen. Ein Beispiel sind Navigations- oder Einparksysteme. Dies kann störend sein, z.B. wenn der Fahrer nach dem Starten des Wagens mit dem Ausparken warten muss, bis die Einparkhilfe funktionsbereit ist. Trotz erheblicher Anstrengungen ist es bei einigen Systemen noch nicht möglich, die Startzeiten auf ein unbedeutendes Zeitintervall zu reduzieren. Eine Umgehung dieses Problems besteht in der frühzeitigen Aktivierung der Funktion. So können kritische Systeme bereits beim Öffnen des Fahrzeugs gestartet werden anstatt beim Einschalten der Zündung.

- **Standortbestimmung:** Bei vorhandenem Navigationsgerät und einer Kommunikationsschnittstelle ist es möglich mit entsprechender SW eine Ortung des Fahrzeugs durchzuführen. Meist ist dies aufgrund des Stromverbrauchs eines solchen Systems lediglich bei eingeschalteter Zündung möglich. Die Ortung erfolgt mittels GPS Daten, die mittels einer SMS an eine gespeicherte Mobilfunknummer gesendet werden.

- **Stilllegung:** Als Erweiterung der Ortung besteht die Möglichkeit ein Auto aus der Ferne stillzulegen. Ein solcher Vorgang muss natürlich sicherstellen, dass dadurch keine Gefährdung entsteht. Ein plötzliches Abschalten aller Systeme scheidet demnach aus, aber eine Reduzierung der Antriebsleistung oder eine Abschaltung bei Stillstand sind möglich. Zusammen mit der Ortung bietet diese Funktion einen wirksamen Schutz gegen Diebstahl.[76]

- **Aktionsradius:** Das Fahrzeug kann für die Nutzung in einer bestimmten Umgebung freigegeben werden. Sollte das Fahrzeug diesen ausgewiesenen Bereich verlassen, erfolgt eine Mitteilung, z.B. per SMS, an den Besitzer. Diese Funktion setzt weitere Funktionen voraus, u.a. ein Navigationsgerät zur Bestimmung der derzeitigen Position und eine Kommunikationsschnittstelle, z.B. ein GSM Modul und eine entsprechende SIM Karte.

---

[76] Allerdings nur solange das Fahrzeug nicht abgeschleppt wird und die Sicherungssysteme später umgangen werden.

- **Update von Daten:** Heutige Fahrzeuge bieten umfangreiche Multimediafunktionen, inklusive Musik, Video, Kalenderfunktion und Internetzugriff. Über eine externe Schnittstelle ist es möglich, persönliche Updates einzuspielen und die Daten mit anderen Systemen zu synchronisieren. So könnte ein Mobilfunkgerät bspw. die Kontaktdaten oder Termine mit der Datenbank des Fahrzeugs abgleichen.
- **Konfiguration:** Die Einstellmöglichkeiten moderner Fahrzeuge werden immer umfangreicher. Die Konfiguration im Fahrzeug ist im Allgemeinen eher umständlich, da ein geeignetes Interface (Tastatur oder Maus) fehlt. Diese Funktion ermöglicht den Zugriff auf die individuellen Einstellungen mittels einer eigenen SW am PC oder Laptop. Die Veränderungen werden per WLAN oder über den Schlüssel selbst auf das Fahrzeug übertragen.

Als Vorabinformation für die spätere Produktaufteilung sei erwähnt, dass sich Komponenten und Funktionen teilweise überschneiden. Die Funktionen Standortbestimmung, Stilllegung und Aktionsradius sind Bestandteil der Komponente GSM. Die letzten zwei Funktionen Update von Daten und Konfiguration sind in der Komponente WLAN zusammengefasst. Alle anderen Funktionen entsprechen einer Komponente.

Die 5 geplanten neuen Produkte decken einen großen Marktbereich ab, vom Kleinwagen bis zur Luxuslimousine. Folglich wird nicht ein identisches Zugangssystem für alle 5 Modelle verwendet, sondern es erfolgt eine starke Differenzierung von einem reinen Basisprodukt, das allein die notwendigen Funktionen bietet bis hin zu einem Produkt mit vollem Funktionsumfang. Eine genaue Produktdefinition erfolgt in Kapitel 5.3.3. Für das Unternehmen stellt sich nun die Frage, ob es diese geplanten Produkte als eigenständige Produkte entwickeln soll, oder ob der SWPL Ansatz geeignet ist, um Entwicklungskosten und –zeit zu sparen.

5.3.2. Erhebung der notwendigen Basisdaten

Im ersten Schritt werden allgemein relevante und notwendige Daten erfasst. Dabei handelt es sich einerseits um Daten über die Größe der geplanten SWPL und um Daten, die für die spätere Business Case Berechnung benötigt werden. Diese Daten werden direkt zu Beginn der Analyse erhoben, um alle entsprechenden Daten an einer zentralen Stelle verfügbar zu haben. Zusätzlich ergibt sich ein Zeitvorteil, da die Erhebung komplexer Daten früh angestoßen werden kann. Teilweise können auch Informationen aus der Beantwortung des Fragebogens

genutzt werden, allerdings ist an dieser Stelle ein höheres Detailniveau erforderlich. Im Folgenden werden die abgefragten Daten aufgeführt, inkl. einer kurzen Erklärung der Bedeutung und Verwendung im Modell:

- **Unternehmensziel:** Welches Ziel verfolgt die Unternehmung vorrangig. Kosten, Qualität und Zeit sind die häufigsten Antworten. Ebenso könnte auch eine abgeleitete Gewinnorientierung vorliegen. Für diese Zielsetzungen eignet sich das e-FACTS Modell. Sollten jedoch weitere Ziele genannt werden, ist zu prüfen, ob das Modell dies abbilden kann. Im Detail werden die Anwendungsmöglichkeiten bei unterschiedlichen Zielsetzungen in Kapitel 5.6.2 diskutiert.

- **Anzahl Entwickler:** Die Anzahl der Entwickler gemessen als FTEs wird später benötigt, um die Entwicklungsdauer der SWPL abschätzen zu können. Anhand der mittels FP berechneten Aufwände kann über die durchschnittliche Produktivität und die verfügbare Anzahl von FTEs die benötigte Zeit abgeschätzt werden. Gefragt ist die Gesamtzahl der Entwickler im Unternehmen. Bei großen Unternehmen bezieht sich die Zahl allerdings allein auf den relevanten Unternehmensbereich.

- **Kosten pro Entwickler:** Zu ermitteln sind die durchschnittliche Kosten pro Entwickler pro Jahr als Vollkosten, also inklusive aller Lohnnebenkosten.[77] Dieser Wert wird benötigt, um die reine Anzahl an FPs in Verbindung mit einer durchschnittlichen Produktivität in reale Kosten umzurechnen.

- **Strafzahlungen:** In vielen Verträgen zwischen Zulieferern und OEMs ist ein fester Liefertermin vereinbart. Eine Verzögerung kann für den OEM Probleme bei der Einhaltung seines SOP Termins bedeuten. Daher sind diese Termine oft mit Strafzahlungen verbunden, die fällig werden, wenn ein Liefertermin nicht eingehalten werden kann. Wenn solche absehbaren finanziellen Verpflichtungen bestehen, sind sie in der Berechnung des Business Cases zu berücksichtigen.

- **Produktivitätsmessung:** Das e-FACTS Modell ermittelt FPs. Dieser Wert gibt insbesondere im Vergleich zwischen Entwicklung mit und ohne Wiederverwendung ein Indiz über die wirtschaftliche Vorteilhaftigkeit eines der beiden Verfahren. Um allerdings eine Aussage hinsichtlich der benötigten Zeit zur Entwicklung neuer

[77] Oft wird dieser Wert auch als Fully Loaded Personal Costs bezeichnet.

Produkte auf Basis einer SWPL treffen zu können, ist es notwendig, die individuelle Produktivität eines Unternehmens zu kennen. Sollte eine derartige Erfassung bereits durchgeführt worden sein, werden die Werte nun abgefragt. In der Literatur finden sich Tabellen für die Umrechnung verschiedener Aufwandsmetriken.[78] Aus den bereits genannten unterschiedlichen Stärken und Schwächen der einzelnen Metriken ergibt sich eine begründete Skepsis gegenüber Umrechnungstabellen. Dennoch bieten sie eine Gelegenheit, eine erste Idee von der Produktivität des Unternehmens bei der SW Erstellung zu bekommen und sollten daher genutzt werden. Um für die spätere Analyse über einen belastbaren Wert zu verfügen, wird allerdings vorgeschlagen, eine separate FP Zählung für ein bereits abgeschlossenes SW-Projekt durchzuführen. Für dieses abgeschlossene Projekt sind die tatsächlichen Aufwände bekannt und erlauben die Berechnung einer Produktivität.

- **Produktlinienumfang:** Anzugeben ist die Anzahl der geplanten Produktentwicklungen auf Basis der SWPL über die gesamte Laufzeit. Dies ist ein entscheidender Wert, da die Anzahl der Wiederverwendungen von Komponenten aus der Asset Base maßgeblich den ökonomischen Nutzen der SWPL bestimmt. Im weiteren Verlauf erfolgt eine Detaillierung der geplanten Produkte basierend auf dieser Angabe.

- **Laufzeit:** Eine SWPL hat eine begrenzte Laufzeit, da geänderte Anforderungen und Techniken in einem Markt Anpassungen an der Asset Base notwendig machen. Ab einer gewissen Grenze ist eine Neugestaltung der SWPL Architektur und der Asset Base kostengünstiger als eine Anpassung. Für Unternehmen, die eine SWPL planen, ist eine ungefähre Vorstellung von der möglichen Laufzeit notwendig, um eine ökonomische Analyse durchführen zu können.

- **Produkte pro Jahr:** Die Unternehmen sollen für jedes Jahr einzeln angeben, welche Produkte auf den Markt gebracht werden. Einerseits erlaubt dies eine Überprüfung der geplanten Zeitpunkte mit den Möglichkeiten einer SWPL. Wenn das Unternehmen bspw. verstärkt zu Beginn neue Produkte auf den Markt bringen möchte, ist dies kaum mit einer SWPL zu realisieren. Die Zahl der Markteinführungen neuer Produkte je Zeiteinheit sollte sich über die Laufzeit der SWPL erhöhen und kurz vor Ende das

[78] So bietet bspw. [Boeh00, S.20] eine Tabelle zur Umrechnung von SLOC in unadjusted Function Points.

Maximum erreichen. Zu Beginn verzögert der Aufbau der Asset Base die Entwicklung gegenüber einer Einzelentwicklung. Außerdem ermöglicht die Angabe der zeitlichen Abfolge von Produktveröffentlichungen eine finanzmathematisch korrekte Berücksichtigung der Zahlungsströme.

- **Nicht wiederverwendbare Komponenten:** Das vorhandene Wissen in Unternehmen über die Möglichkeiten der Wiederverwendung von Komponenten kann genutzt werden, um bereits im Vorfeld einer Detailanalyse einige Komponenten auszuklammern, bei denen eine Wiederverwendung grundsätzlich nicht möglich oder sinnvoll ist. Dadurch kann der weitere Analyseaufwand reduziert werden. Die Entscheidung zum Ausschluss von Komponenten sollte allerdings eher konservativ erfolgen, um das Potenzial der SWPL nicht zu stark einzuschränken. Dabei sind insbesondere zukünftige technische Veränderungen zu berücksichtigen, die Einschränkungen aufheben könnten.

- **Kalkulationszinssatz:** Im Allgemeinen werden SWPL über einen Zeitraum größer als ein Jahr genutzt. Dadurch ergeben sich zeitliche Differenzen der Ein- und Auszahlungsströme, die finanzmathematisch korrekt abzubilden sind. Die Geldflüsse sind auf einen einheitlichen Zeitpunkt auf- bzw. abzuzinsen. Am Häufigsten wird der Startpunkt der Investition verwendet und ein Kapitalwert der Investition berechnet [Wöhe08, S. 537]. Ebenso kann jedoch auch der Endpunkt gewählt werden [Grob06, S. 53ff.]. Zur Berechnung sollte der Kalkulationszinssatz verwendet werden. Das ist die Verzinsung, die ein Unternehmen im Durchschnitt von seinen Investitionen mindestens erwartet.[79]

- **Zusätzliche HW:** Aufgrund der hohen Stückzahlen in der Automobilindustrie ergibt sich eine besondere Bedeutung der Stückkosten. In Verbindung mit der hohen Anzahl an möglichen Varianten resultiert daraus die Notwendigkeit, Funktionen entsprechend zu kapseln, um nur die benötigte HW in einem spezifischen Automobil zu verbauen. SWPL gehen grundsätzlich von einer modularen Struktur aus, die über Variationspunkte individuell erweitert werden kann. Es ist jedoch zu beachten, dass

[79] Am Kapitalmarkt ergeben sich unterschiedliche Soll- und Habenzinssätze. Daher ergibt sich der Kalkulationszinssatz als Mischzinssatz oder auch Weighted Average Cost of Capital (WACC) entsprechend der eigenen Fremd- und Eigenkapitalanteile. Bei einer reinen Eigenkapitalfinanzierung entspricht der Kalkulationszins der entgangenen Verzinsung aus einer Alternativanlage (Opportunitätskosten), bei reiner Fremdfinanzierung dem Sollzins [vgl. Wöhe08, S. 573 und Grob99, S. 16].

dies ebenso für die HW gelten muss. Hier ist eine individuelle Anpassung der Leistungsfähigkeit an die Variabilität der SWPL notwendig, um Mehrkosten zu vermeiden. Ein Ansatz besteht in der schichtenübergreifenden Optimierung und der Konsolidierung von Steuergeräten [Hörn05, S. 417]. Im Detail ergibt sich die Notwendigkeit zusätzlicher HW erst durch die Ausgestaltung der SWPL, dennoch ist zu erwarten, dass einige Unternehmen dies bereits frühzeitig abschätzen können. Wenn dieser Punkt positiv beantwortet wird, sollte dies bei der Berechnung des Business Cases berücksichtigt werden.

- **Szenarios:** Die Planung über einen längeren Zeitraum enthält Risiken, da die zukünftige Entwicklung nicht mit Sicherheit vorhergesehen werden kann. Aufgrund der langen Laufzeiten von SWPL ist mit einer entsprechend hohen Unsicherheit zu rechnen. Eine Möglichkeit, dies zu berücksichtigen, sind Szenarien, mit denen unterschiedliche zukünftige Entwicklungen beschrieben und die finanziellen Auswirkungen auf die SWPL entsprechend berechnet werden. Gewichtet man die einzelnen Szenarien mit Wahrscheinlichkeiten, dann kann man einen risikoadjustierten Wert der Investition berechnen. An dieser Stelle sollen die Unternehmen angeben, ob es aus ihrer Sicht sinnvoll ist, Szenarios zu verwenden (inkl. einer kurzen Beschreibung). Die entsprechende Berechnung erfolgt in Kapitel 5.5.

- **Verwendung eingesparter Zeit:** Erfolgreiche SWPL reduzieren die Entwicklungszeit durch einen höheren Wiederverwendungsanteil von existierenden SW Komponenten. Die Unternehmen haben verschiedene Möglichkeiten, die damit freigewordenen Kapazitäten zu nutzen. Einerseits können zusätzliche Produkte entwickelt werden, die Tests können erweitert werden oder es können grundsätzlich die Entwicklungskosten reduziert werden. Die individuelle Zielsetzung eines Unternehmens kann bei der Berechnung des Business Cases berücksichtigt werden.

- **Preisvariabilität:** SWPL zielen neben einer Reduzierung der Entwicklungskosten darauf ab, Produkte stärker zu individualisieren, die Qualität zu steigern und mehr Funktionen zu integrieren. Für die Berechnung des Business Cases ist es von Interesse, ob die Kunden bereit sind, dies mit einem höheren Produktpreis zu honorieren. Die Antwort hat Auswirkungen auf die Auslegung der SWPL und bestimmt damit das finanzielle Ergebnis.

Insgesamt lassen sich die benötigten Daten innerhalb kurzer Zeit erheben. Aufgrund des frühen Zeitpunktes in der Gesamtanalyse ergibt sich zusätzlich die Möglichkeit weitere Quellen im Unternehmen anzustoßen, sollte eine Information nicht direkt verfügbar sein.

Beispiel Zugangssysteme:

Der gesamte Fragebogen inklusive der Antworten für das Beispiel der Zugangssysteme findet sich in Tabelle 25. Auf einige Antworten wird hier im Detail eingegangen. Im Beispiel erfolgt die geplante erste Produktveröffentlichung erst im zweiten Jahr der SWPL. Die weiteren Einführungen verteilen sich gleichmäßig über die Laufzeit mit einem Fokus im 4. Jahr. Eine solche Planung berücksichtigt die Möglichkeiten und Einschränkungen der SWPL. Weiterhin wird angenommen, dass die SWPL zu einer komplexeren HW führen wird. Dies ist im weiteren Verlauf, spätestens jedoch bei der Erstellung des Business Cases zu validieren und entsprechend zu berücksichtigen. Und schließlich werden zwei Szenarios als relevant erachtet, die hinsichtlich der Marktresonanz auf die Möglichkeit, über externe Schnittstellen auf Funktionen des Fahrzeugs zugreifen zu können, unterscheiden. Bei einer breiten Marktakzeptanz und entsprechend großer Nachfrage kann bei einer SWPL schnell eine Erweiterung der angebotenen Funktionen erfolgen. Dieser Effekt wird bei der Berechnung des Business Cases berücksichtigt.

	Optionen/Einheit	
Unternehmensdaten		
Hauptsächlich verfolgte Ziele des Unternehmens	Gewinn, Innovation, Marktführer, Preis	Gewinn
Anzahl der beschäftigten Entwickler	in FTEs (Full Time Equivalent; Teilzeitkräfte werden entsprechend auf eine Vollzeitkraft umgerechnet)	150
Durchschnittliche Kosten je Entwickler pro Jahr	in EUR als Vollkosten inkl. aller Lohnnebenkosten	100000
Haben sie sich vertraglich zu Strafzahlungen verplichtet, falls sie einen Liefertermin nicht einhalten können sollten?	Ja/nein	Nein
	Falls ja, bitte Angabe der Höhe in EUR	
Haben sie schon einmal eine Produktivitätsmessung ihrer Softwareerstellung durchführen lassen?	Ja/nein	Ja
	Falls ja, basierend auf welcher Metrik (z.B. LOC, FPs, ...); bitte auch Vergleichsskala benennen (wenn möglich)	FPs
	Ihre aktuelle Produktivität	5 FP/PM

Produktlinie		
Umfang der Produktlinie	Anzahl der geplanten Produkte auf Basis der Produktlinie	5
Geplante Laufzeit der Produktlinie	in Jahren	5
Welche Produkte sollen jedes Jahr auf den Markt kommen?	2009	-
	2010	Fzg. 1
	2011	Fzg. 2
	2012	Fzg. 3 und 5
	2013	Fzg. 4
Wenn aus Erfahrung bereits feststeht, dass einige Komponenten nicht wiederverwendbar zu realisieren sind: Bitte Angabe der individuell zu realisierenden Komponenten	Nennung der Komponenten	
Welchen Kalkulationszinssatz verwenden sie derzeit?	in Prozent	8%
Besteht die Möglichkeit, dass zusätzliche Hardware pro Produkt notwendig wird aufgrund der Wiederverwendung von Software?	Ja/nein, bitte mit kurzer Begründung	Ja, umfangreichere ECUs für volle Funktionalität

Branche/Industrie		
Erachten sie es für ihre Industrie und ihr Unternehmen für sinnvoll, bei der Berechnung unterschiedliche Szenarien zu berücksichtigen, um die Unsicherheit in der Zukunft abzubilden?	Ja/nein	Ja
Wieviele Szenarios halten sie für notwendig/sinnvoll? Und bitte geben sie eine kurze Beschreibung jedes Szenarios.	Anzahl und kurze Beschreibung	2, gute und schlechte Marktresonanz auf externen Zugriff auf Fzg.-Funktionen
Wenn sie im Entwicklungsprozess Zeit einsparen könnten, wofür könnten sie diese Zeit am besten verwenden?	Schnellere Markteinführung, besseres Testen, kleineres Entwicklerteam, bessere Erfüllung der Kundenwünsche, etc.	80% Kürzere Entwicklungszeiten, 20% mehr Testing
Würden ihre Kunden einen höheren Kundennutzen bei ihren Produkten (bessere Qualität, mehr Funktionen, besser auf den Kunden zugeschnitten) durch höhere Preise honorieren?	Ja/nein; bitte zusätzlich Angabe, in welchem Umfang	Unklar, bei wirklich innovativen Funktionen wahrscheinlich ja

Tabelle 25 – Basisdaten für Beispiel SWPL

5.3.3. Übersicht der geplanten Produktstruktur

Produkte sind das Ergebnis eines aufwendigen Entwicklungsprozesses, da mit der Einführung eines Produktes hohe Investitionen und entsprechende Risiken verbunden sind. Zusammengefasst werden alle Tätigkeiten, die die Entwicklung, Herstellung und Vermarktung eines Produktes umfassen, im Product Management. Im Allgemeinen wird Product Management definiert als: "Planning, organising, executing, and controlling of all tasks, which aim at a successful conception, production, and marketing of the products offered by a company" [Pohl05, S. 167]. Im Zusammenhang mit SWPL wurde diese Definition von Pohl angepasst: "Product management is the sub-process of domain

engineering for controlling the development, production, and marketing of the software product line and its applications" [Pohl05, S. 167]. Zu den Aufgaben des Product Managements gehört es ebenso, entsprechend der erwarteten Marktentwicklung ein Produkt-Portfolio und eine Product Roadmap zu entwerfen. Dies ist bereits bei Einzelentwicklungen notwendig, dennoch kann in einem solchen Fall einfacher ein zusätzliches Produkt in die Roadmap integriert werden. Für eine SWPL ist es entscheidend, dass zu Beginn alle Anforderungen an die zukünftigen Produkte bekannt sind. Die grundsätzlichen Rahmenbedingungen der Produktentwicklung werden über die SWPL Architektur weitgehend festgelegt und können nur mit sehr großem Aufwand geändert werden.

In diesem Kapitel werden daher alle Informationen zu den geplanten Produkten erfasst. Zunächst erfolgt eine reine Übersicht der geplanten Produkte während der Laufzeit der SWPL. Hierbei ist es wichtig, dass sämtliche Produkte des Unternehmens oder bei großen Unternehmen des entsprechenden Bereichs, die während dieser Zeit entwickelt werden, erfasst werden; auch Produkte, bei denen klar ist, oder es zumindest klar scheint, dass sie nicht Teil der SWPL werden können. Die Entscheidung über den tatsächlichen Scope der SWPL wird später getroffen und durch die Ergebnisse der e-FACTS Analyse unterstützt. Tabelle 26 zeigt die 5 geplanten Produkte des Anwendungsbeispiels. Zusätzlich zu einer groben funktionalen Beschreibung werden die geplante Verfügbarkeit des Produktes und die angestrebten Verkaufszahlen erfasst. Als letztes findet sich eine Spalte für den Anteil der tatsächlich neu entwickelten Funktionen eines Produktes. Schließlich wird es sich nicht bei jedem Produkt um eine komplette Neuentwicklung handeln, sondern es können bestehende Komponenten auch ohne die Verwendung einer SWPL genutzt werden. Diese Information wird später für die Aufwandsbewertung benötigt, da lediglich neue Funktionen als tatsächlicher Aufwand zu berücksichtigen sind.

Produkte				
Name	Beschreibung	Geplante Verfügbarkeit (SOP oder Relaunch)	Geplante Verkaufs-zahlen (in Stück)	Anteil Neuentwicklung bei Relaunch (in %)
Zugangssystem Fzg. 1	Basissystem, Zugang über Schlüssel, Wegfahrsperre und Möglichkeit Fenster zu öffnen	Anfang 2011	1.500.000	100%
Zugangssystem Fzg. 2	Sicherheit, Zugang über Fernbedienung, personalisierbar, Alarmanlage, Wegfahrsperre, Homecoming Light, und programmierbare Höchstgeschwindigkeit je Fernbedienung, Fensteröffnung	Mitte 2012	1.200.000	100%
Zugangssystem Fzg. 3	Komfort, Zugang über RFID, Wegfahrsperre, Personalisierung, Fensteröffnung möglich, Starten von Fkt. beim Einsteigen	Anfang 2013	700.000	100%
Zugangssystem Fzg. 4	Mittel, Zugang über Fernbedienung oder RFID, Wegfahrsperre, Alarmanlage, Fensteröffnung, Starten von Funktionen beim Einsteigen, Homecoming Light, GSM Schnittstelle	Herbst 2013	900.000	100%
Zugangssystem Fzg. 5	Komplett, Zugang über RFID, Alarmanlage, Wegfahrsperre, Homecoming Light, Personalisierung, programmierbare Höchstgeschwindigkeit, Fensteröffnung, Starten von Anwendungen beim Einsteigen, Externe Schnittstelle über GSM und WLAN nutzbar	Anfang 2014	200.000	100%

Tabelle 26 – Produktübersicht SWPL Zugangssysteme

In einem zweiten Schritt erfolgt die Aufspaltung der Produkte in die entsprechenden Komponenten. Je nach Unternehmen kann anschließend eine weitere Unterteilung entsprechend der Funktionen erfolgen. Dies ist aber stark von der unternehmensinternen Struktur abhängig und beeinflusst weniger das vorliegende Modell. Die Aufspaltung ist notwendig, um eine funktionale Zerlegung der Produkte zu erlangen. Auf Basis dieser Komponenten oder auch Funktionen erfolgt später die FP Bewertung und die Einstufung der Wiederverwendbarkeit. In einem ersten Schritt wird die Komponentensicht genutzt, um Ähnlichkeiten zwischen den Produkten besser zu erkennen. Sollten in der Komponentenübersicht Bezeichnungen mehrfach auftauchen, ist das ein erstes Anzeichen für eine bisher ungenützte Wiederverwendungsmöglichkeit. Vor der weiteren Analyse sollten diese Potenziale identifiziert werden. Eine korrekte Komponentenübersicht enthält keine ähnlichen Komponenten, außer in begründeten Ausnahmefällen, z.B. wenn sich die Funktionalität einer Komponente für zwei Produkte deutlich unterscheidet. Stattdessen werden Komponenten, die eine große Ähnlichkeit aufweisen, zusammengefasst und als Bestandteil von mehreren Produkten gekennzeichnet. Tabelle 27 zeigt die Komponentenübersicht aus dem Anwendungsbeispiel.

Komponenten										
		Bestandteil folgender Produkte					Ähnlichkeit mit anderer Komponente?		
Name	Beschreibung	Prod1	Prod2	Prod3	Prod4	Prod5	Ja/nein	Welche?	Ähnlichkeit in %
Zugang	Ermöglicht Zugang über verschiedene Wege, wie Schlüssel, Fernbedienung, RFID	x	x	x	x	x	nein		
Personalisierung	Ermöglicht die individuelle Anpassung von Funktionen über eine Identifizierung des Objektes zur Öffnung		x	x		x	nein		
Alarmanlage	Sichert das Auto gegen Diebstahl, wird bei Öffnung automatisch ab- und beim Schließen eingeschaltet		x		x	x	nein		
Wegfahrsperre	Codes durch den Chip in dem Zugangsobjekt wird Zündung und Motor freigegeben; Fkt. Stark in die Architektur integriert, und zur Sicherheit verschlüsselt	x	x	x	x	x	nein		
Homecoming Light	Nutzung der Scheinwerfer zur Ausleuchtung der Umgebung nach Öffnen oder Schließen des Fahrzeugs		x		x	x	nein		
Höchst-geschwindigkeit	Die Höchstgeschwindigkeit kann für jeden Schlüssel individuell festgelegt werden, kann nicht geändert werden		x			x	nein		
Fensteröffnung	Beim Öffnen oder Schließen können Fenster geöffnet oder geschlossen werden mittels Schlüssel	x	x	x	x	x	nein		
Starten von Anwendungen	Manche Anwendungen benötigen Zeit zum Starten, diese werden direkt beim Öffnen des Fzg. Gestartet			x	x	x	nein		
GSM	GSM bietet die Möglichkeit zur Kommunikation über das Mobilfunknetz und ermöglicht 3 Funktionen im Zusammenhang mit Zugangssystemen: Standortbestimmung, Stillegung, Aktionsradius				x	x	nein		
WLAN	WLAN bietet die Möglichkeit zur Kommunikation über private Netzwerke mit kurzen Reichweiten. Im Zusammenhang mit Zugangssystemen werden 2 Funktionen ermöglicht: Update von Daten Konfiguration des Fahrzeugs					x	nein		

Tabelle 27 – Komponentenübersicht SWPL Zugangssysteme

Es besteht eine deutliche Ähnlichkeit der Komponentenübersicht mit der in Kapitel 3.4 vorgestellten Produktkarte des PuLSE-Eco Ansatzes. Dies ist beabsichtigt, da PuLSE-Eco als Basis für das Scoping von SWPL weitgehend übernommen worden ist. Allerdings wurde der Ansatz auf die wesentlichen Elemente reduziert, um den Analyseaufwand möglichst gering zu halten. Im Falle einer unklaren Beschreibung der eigentlichen SWPL und der Domainabgrenzung, wird allerdings eine detaillierte Beschäftigung mit dem PuLSE-Eco Ansatz als Unterstützung bei der Anwendung des e-FACTS Modells empfohlen. Siehe dazu [Schm03, Schm04 und Schm05].

Zur Identifizierung von ähnlichen Komponenten sind die letzten drei Spalten gedacht. Während der ersten Erfassung werden zunächst alle bisherigen Komponenten aufgenommen. Im Anschluss erfolgt jedoch eine Analyse der Ähnlichkeiten. Diese werden eingetragen und die Komponentenübersicht anschließend angepasst. Zur Überprüfung dient die Spalte

Ähnlichkeit Ja/Nein. Bevor mit dem nächsten Analyseschritt begonnen werden kann, darf diese Spalte nur "Nein" enthalten.

In einem letzten Schritt im Rahmen der Produktübersicht bietet e-FACTS eine Übersicht über die Komponenten an. Diese zeigt die Anzahl der geplanten Wiederverwendungen der Komponenten in einzelnen Produkten. Zur besseren Übersichtlichkeit wird daneben der Anteil der Wiederverwendungen im Verhältnis aller neuen Produkte berechnet. Ziel dieser zusätzlichen Information ist die Möglichkeit zur Angabe, ob für die jeweilige Komponente eine FP-Analyse durchgeführt werden soll. Wenn Gründe dagegen vorliegen, werden sie hier erfasst. Der Vorteil ist, dass unnötige FP-Berechnungen dadurch vermieden werden. Die Anwender haben die Wahl, die Komponenten, die sich nicht für eine Wiederverwendung eignen, komplett von der weiteren Analyse auszuschließen. Möglich wird dies, da das Ergebnis der Analyse ein Vergleich zwischen Einzelentwicklung und SWPL ist. Komponenten, die für jedes Produkt individuell entwickelt werden müssen, können daher auf beiden Seiten der Gleichung vernachlässigt werden.

Es besteht jedoch die Möglichkeit, damit den Aufwand für die SWPL etwas zu gering zu berechnen. Denn die Einpassung der individuell entwickelten Komponente könnte einen Mehraufwand bedeuten, da die Rahmenbedingungen der SWPL eingehalten werden müssen. Zusätzlich lässt sich in diesem Fall kein Gesamtaufwand für die SWPL ermitteln. Das Ergebnis wäre einzig eine Aussage, welche Entwicklungsmethode im konkreten Fall ökonomisch vorteilhaft ist. Aus diesen Gründen wird empfohlen, die Komponenten bei der FP-Berechnung ohne Wiederverwendung mit einzubeziehen.

Komponenten				
Name	Wiederverwendungen in Produkten	**Eignung der Kompo-nente für Domain**	Analyse als Domain Asset (ja/nein)	Gründe die gegen eine Wiederverwendung sprechen
Zugang	5	100%		
Personalisierung	3	60%		
Alarmanlage	3	60%		
Wegfahrsperre	5	100%		
Homecoming Light	3	60%		
Höchstgeschwindigkeit	2	40%		
Fensteröffnung	5	100%		
Starten von Anwendungen	3	60%		
GSM	2	40%		
WLAN	1	20%		

Tabelle 28 – Bewertung der Komponenten als Domain Asset

5.4. Schritt 3: Einführung von Feature Points zur Komplexitätsmessung

Schritt 3 ist der umfangreichste Analyseschritt. Hier erfolgt die Bewertung des Entwicklungsaufwands auf Funktionsebene mittels der in Teil A Kapitel 5.7 vorgestellten Feature Point Methodik. Zuerst wird der Umfang der geplanten zukünftigen SW Produkte als Einzelentwicklungen ermittelt. Anschließend folgt in Kapitel 5.4.2 die Erklärung und Validierung, wie Variabilität von SWPL und der daraus resultierende höhere Entwicklungsaufwand mittels der FP Methodik erfasst werden können. Dieser Schritt ist notwendig, um im Kapitel 5.4.3 die FP Zählung mit Wiederverwendung durchführen zu können. Dabei stehen zwei unterschiedliche Vorgehen zur Wahl, die die Erfahrungen der Unternehmen mit dem Thema SWPL berücksichtigen. Im abschließenden Kapitel 5.4.4 erfolgt die Optimierung der Asset Base. Das Modell gibt eine ökonomisch basierte Empfehlung, welche Komponenten als Domain Asset wieder verwendbar gestaltet und welche als Einzellösungen realisiert werden sollten. Am Ende von Schritt 3 liegen die Entwicklungsaufwände auf Komponentenebene und eine Empfehlung für die Gestaltung der Domain Assets vor. Die vollständige ökonomische Bewertung der SWPL erfolg abschließend in Schritt 4, wenn die Daten in einer Finanzanalyse bzw. einem Business Case aufbereitet werden.

5.4.1. Feature Point Zählung ohne Wiederverwendung

Die Feature Point Methodik eignet sich zur Abschätzung des Entwicklungsaufwands von SW-Projekten im automobilen Bereich [McKi06]. Als Ausgangspunkt für eine Zählung wird eine Übersicht der kundenrelevanten Funktionen, die entwickelt werden sollen, benötigt. Die in Analyseschritt zwei erstellte Komponentenübersicht bildet eine solche Aufstellung und wird im weiteren Verlauf verwendet. Je nach Detailgrad der Übersicht korrelieren die Komponenten bereits mit den Funktionen oder es ist eine weitere Unterteilung notwendig. Je Komponente ist eine Feature Point Zählung durchzuführen. Dies geschieht mit Hilfe des in Tabelle 29 dargestellten Arbeitsblattes.

Feature Point Zählung ohne Wiederverwendung je Funktion einer Komponente oder eines Systems

Name der Komponente:

Nr.	Funktionsbezeichnung	EI			EO			EQ			AL			DF			EIF		
		L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
	Anzahl	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Gewichtung	3	4	6	4	5	7	3	4	6	2	3	5	5	7	10	5	7	10
	Feature Points	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
																Total FP unadjusted		**=**	**0**
	Funktion																		
1																			
2																			
3																			
4																			
5																			
6																			

Tabelle 29 – Worksheet zur Zählung der Feature Points je Komponente (gelbe Felder dienen der Dateneingabe, sonstige Daten dienen der Berechnung, L, M und H stehen für die unterschiedlichen Komplexitätsklassen)

Es werden alle von dieser Komponente gelieferten Funktionen aufgeführt. Jede Funktion wird dann einzeln bewertet. In der Praxis hat es sich bewährt, die Zählung entlang der in Tabelle 29 dargestellten Anordnung der Funktionstypen durchzuführen. Im Folgenden werden die einzelnen Funktionstypen definiert und die Komplexitätsklassen beschrieben:

- **EI:** External Input ist ein logischer, elementarer Geschäftsprozess, der die Anwendungsgrenze überquert, um auf Daten oder Algorithmen zuzugreifen. Im Sinne von Embedded Systems sind EIs zu verstehen als Transaktion oder Prozess, der einen Datenstrom für einen Algorithmus zur Verfügung stellt und ihn dadurch anstößt. Beispiele sind: Ein Nutzer, der das Fenster seines Fahrzeugs öffnen möchte oder das Navigationssystem, das die aktuelle Uhrzeit vom Kombiinstrument erfragt.

Komplexitätsklassen		
Niedrig	**Mittel**	**Hoch**
• 1 - 4 Parameter, Variablen oder Zustände, die eingegeben werden • 1 DF oder AL referenziert	• 5 - 15 Parameter, Variablen oder Zustände, die eingegeben werden • 2 DF oder AL referenziert	• > 15 Parameter, Variablen oder Zustände, die eingegeben werden • > 2 DF oder AL referenziert

Tabelle 30 – Komplexitätsklassen für EI

- **EO:** External Output ist ein logischer, elementarer Geschäftsprozess, der zur Weitergabe von Daten/Informationen an Systeme oder Nutzer außerhalb der Anwendungsgrenzen führt. Ein EO ist oft das Ergebnis eines verarbeiteten Inputs. Beispiele sind: Der Befehl zum Öffnen des Fensters oder die Anzeige einer Information auf dem Cockpitdisplay.

Komplexitätsklassen		
Niedrig	**Mittel**	**Hoch**
• 1 - 5 Parameter, Variablen oder Zustände in EO enthalten • 1 DF oder AL referenziert	• 6 - 19 Parameter, Variablen oder Zustände in EO enthalten • 2 - 3 DF oder AL referenziert	• > 19 Parameter, Variablen oder Zustände in EO enthalten • > 3 DF oder AL referenziert

Tabelle 31 – Komplexitätsklassen für EO

- **EQ:** External Query ist ein logischer, elementarer Geschäftsprozess, der aus einem "Datentrigger" besteht, gefolgt von der Abfrage von Daten, die im System gespeichert sind und die Anwendungsgrenzen verlassen. EQs sind demnach EI ähnlich, allerdings erfolgt bei EQs keine Verarbeitung von Daten. Eine Query bezeichnet allein die Anfrage und Ausgabe von unverarbeiteten, im System gespeicherten Daten. In Embedded Systems im Automotive Bereich kommt dieser Funktionstyp praktisch nicht vor. Ein Beispiel wäre die Abfrage von Statusinformationen durch einen Mechaniker in der Werkstatt.

Komplexitätsklassen		
Niedrig	**Mittel**	**Hoch**
• 1 - 5 Parameter, Variablen oder Zustände in Query und EO enthalten • 1 DF oder AL referenziert	• 6 - 19 Parameter, Variablen oder Zustände in Query und EO enthalten • 2 - 3 DF oder AL referenziert	• > 19 Parameter, Variablen oder Zustände in Query und EO enthalten • > 3 DF oder AL referenziert

Tabelle 32 – Komplexitätsklassen für EQ

- **AL:** Algorithmus ist eine Sammlung von Regeln, die komplett angewendet werden müssen, um ein bestimmtes berechenbares Problem zu lösen. Algorithmen sind demnach jede Art von Berechnungsanweisung zur Manipulation von Daten. Sie

werden meist durch einen Input getriggert und liefern oft einen Output. Beispiele sind: Die Berechnung, ob ein Fenster geschlossen werden kann, wenn der Fahrer den Wunsch äußert. Dazu ist zu überprüfen, ob das Fenster nicht bereits vollständig geschlossen ist und ob ein gefahrloses Schließen (Einklemmschutz) möglich ist. Komplexere Algorithmen werden im Bereich des Antriebsstranges verwendet, z.B. zur Berechnung des optimalen Einspritzzeitpunktes des Benzingemisches in den Verbrennungsraum durch die Motorsteuerung.

	Komplexitätsklassen		
	Niedrig	**Mittel**	**Hoch**
Daten Komplexität	• Bis zu 19 Parameter, Variablen oder Zustände bei Berechnung involviert • Einfache Beziehungen zwischen Daten	• 20 - 50 Parameter, Variablen oder Zustände mit 2 - 5 Substrukturen bei Berechnung involviert	• > 50 Parameter, Variablen oder Zustände mit zahlreichen Substrukturen bei Berechnung involviert • Komplexe Datei Struktur und Daten Interaktionen
Problem Komplexität	• Bis zu 16 Regeln im Algorithmus und bis zu 3 Faktoren oder Bedingungen pro Regel • Kein zusätzlicher Aufwand nötig zur Optimierung des Algorithmus	• 17 - 64 Regeln im Algorithmus und 4 Faktoren oder Bedingungen pro Regel • Zeitkritische Berechnung implementiert; zusätzlicher Aufwand	• > 64 Regeln im Algorithmus und > 4 Faktoren oder Bedingungen pro Regel • Echtzeit Optimierung; zusätzlicher Aufwand für hochkomplexe mathematische Funktionen notwendig

Tabelle 33 – Komplexitätsklassen für AL

- **DF:** Data File ist eine logische, persistente Entität, die von Funktionen oder Algorithmen genutzt wird. Dabei müssen DFs nicht immer physischen Komponenten entsprechen, wie Dateien. Vielmehr beziehen sich DFs auf Entities im Sinne eines ER-Diagramms.[80] Beispiele sind: Gespeicherte Statusinformationen, Daten zur Parametrisierung von Funktionen und Referenztabellen für Einspritzzeitpunkte bei Motorsteuerungen. Der Fahrer als eigene Entität liegt außerhalb der Anwendung und wird daher nicht als DF betrachtet.

[80] Entity Relationship-Diagramm; diese werden verwendet, um einen Ausschnitt aus der realen Welt semantisch so zu beschreiben, dass daraus Datenmodelle abgeleitet werden können. Für weitere Details siehe [Chen91 oder Stahl05, S. 166ff.]

Komplexitätsklassen		
Niedrig	**Mittel**	**Hoch**
• Bis zu 19 Parameter, Variablen und Zustände • Maximal eine Substruktur (z.B. Standards, Formatumwandlung)	• 20 - 50 Parameter, Variablen oder Zustände • 2 - 5 Substrukturen	• > 50 Parameter, Variablen oder Zustände • > 5 Substrukturen

Tabelle 34 – Komplexitätsklassen für DF

- **EIF:** External Interface File, eine logische, persistente Entität, die von einem System referenziert wird. Die Verwaltung dieser referenzierten Entität obliegt jedoch einer anderen Anwendung außerhalb der betrachteten Grenzen. Gemeint ist damit eine Schnittstelle zur Außenwelt. Ein Beispiel für ein EIF ist die Anzeige einer Warnleuchte bei Aktivierung des ABS Systems. Dabei nutzt das ABS System die Displayfunktion, ohne selbst dafür verantwortlich zu sein.

Komplexitätsklassen		
Niedrig	**Mittel**	**Hoch**
• Bis zu 19 Parameter, Variablen oder Zustände • Maximal eine Substruktur	• 20 - 50 Parameter, Variablen oder Zustände • 2 - 5 Substrukturen	• > 50 Parameter, Variablen oder Zustände • > 5 Substrukturen

Tabelle 35 – Komplexitätsklassen für EIF

Jeder identifizierte Funktionstyp wird zunächst einer Komplexitätsklasse zugeordnet und anschließend in der entsprechenden Spalte eingetragen. Die Tabelle sieht lediglich ein Feld pro Komplexitätsklasse und Funktion vor, weshalb die Anzahl bei mehr als einem Funktionstyp als Summe aufgenommen wird. Nachdem alle Funktionen analysiert worden sind, kann das Ergebnis, also die Summe der FP für diese Komponente abgelesen werden. Zusätzlich kann ein VAF berücksichtigt werden. Dies bleibt aber zunächst unbeachtet, da dieser Wert lediglich für einen Vergleich der Ergebnisse mit anderen Unternehmen und insbesondere anderen Branchen relevant ist.

Die vorher erfolgte Zerlegung der Produkte in Komponenten ermöglicht eine Reduzierung des Erhebungsaufwands, da nicht alle Produkte komplett gezählt werden müssen, sondern

lediglich alle Komponenten. Für die Aggregation der FP Werte auf Produktebene werden dann die Ergebnisse der im Produkt vorhandenen Komponenten aufsummiert. In einigen Fällen kann es dabei zu Ungenauigkeiten kommen, wenn die Komponente nicht genau den Funktionsumfang des Produktes abdeckt. In einer solchen Situation kann auf die Funktionsaufteilung der eigentlichen Zählung zurückgegriffen werden. Die FPs der nicht benötigten Funktion werden dann abgezogen.

In der Produktübersicht konnten die Unternehmen angeben, dass ein Produkt keine 100%ige Neuentwicklung, sondern eine Weiterentwicklung eines existierenden Produktes ist. Sollte dies der Fall sein, berücksichtigt die Zählung nur die neuen Funktionstypen. In vielen Fällen wird dies jedoch nicht möglich oder zu ungenau sein, da viele der vorhandenen Funktionstypen ebenfalls angepasst werden müssen. Der dabei entstehende Aufwand rechtfertigt oft aber keine komplette Berücksichtigung des Funktionstyps. Daher kann alternativ eine vollständige Zählung erfolgen. Im Anschluss erfolgt dann eine prozentuale Berücksichtigung der FPs entsprechend dem Anteil der Neuentwicklung.

Wichtig ist, dass bis zu diesem Punkt keine Berücksichtigung von Variabilität erfolgt, da die Messung den Aufwand für die Erstellung von Einzelprodukten ausdrücken soll. Die Besonderheiten bei der korrekten Erfassung des Mehraufwands durch Einbeziehung von Variabilität in SWPL werden im nächsten Kapitel besprochen. Davor erfolgt aber noch ein Blick auf die FP Berechnung für das Anwendungsbeispiel der Zugangssysteme im Automobil.

Anwendungsbeispiel:
Der OEM im Anwendungsbeispiel plant, in den nächsten 5 Jahren 5 neue Produkte auf den Markt zu bringen. Bei diesen Produkten handelt es sich jeweils um eigene Modellreihen, zwischen denen bisher keine Vereinheitlichung durchgeführt worden ist, d.h. alle derzeit verwendeten Komponenten sind individuelle Entwicklungen für die jeweilige Modellreihe. Es gibt keine Nutzung von Komponenten über Modellreihen hinweg. Weiterhin ist zu beachten, dass die erfassten FPs lediglich den Aufwand für den Teil der Funktionalität berücksichtigen, der mit dem Zugangssystem in Verbindung steht. Das bedeutet, dass nicht in allen Fällen die komplette Funktionalität der genannten Funktionen erfasst wird. Bspw. nutzen die Funktionen Aktionsradius und Standortbestimmung die Daten des Navigationssystems über die aktuelle Position. Die Positionsbestimmung ist demnach nicht in den FPs enthalten. Die Funktion Alarmanlage beschränkt sich auf die Aktivierung und die Deaktivierung des Systems beim

Öffnen oder Schließen des Fahrzeugs. Die Realisierung der Alarmanlage, also das Ansprechen der Warnblinkanlage und der Hupe und die Kontrolle der Funktion sind ebenfalls nicht erfasst. Die Funktion Höchstgeschwindigkeit fokussiert auf die Möglichkeit die Höchstgeschwindigkeit, über einen personalisierten Schlüssel, individuell einzustellen. Der eigentliche Vorgang der Geschwindigkeitskontrolle über das Motorsteuergerät wird nicht betrachtet.

Eine vollständige Erklärung der gesamten FP Zählung ist aus Platzgründen nicht möglich und auch nicht erforderlich. Um dennoch einen Einblick zu ermöglichen wird im Folgenden beispielhaft die GSM Komponente im Detail besprochen.

Die Komponente GSM setzt voraus, dass das Fahrzeug über ein GSM Modul verfügt. Grundsätzlich dient ein solches Modul dazu, dem Fahrer das Telefonieren komfortabel über die Nutzung der Radiolautsprecher zu ermöglichen, indem entweder eine eigene SIM-Karte im Fahrzeug genutzt wird, oder ein externes Mobilfunkgerät mit dem Fahrzeug per Kabel oder Bluetooth verbunden wird. Das System kann erweitert werden, um Telefonie-unabhängige Funktionen zu ermöglichen. Wie bereits oben erwähnt, ist die reine Telefonfunktionalität nicht der Fokus der FP Analyse, diese wird als vorhanden angenommen. Da es sich hier um eine Einzelentwicklung handelt, soll die Komponente als Einheit entwickelt werden. Der Kunde hat also später die Möglichkeit, sich für alle Funktionen zu entscheiden, oder keine zu ordern. Eine freie Wählbarkeit wird erst auf Basis der SWPL realisiert. Ebenso werden keine Anstrengungen unternommen, um die Komponente für zukünftige Erweiterungen vorzubereiten. Um eine klarere Trennung zwischen SWPL und Einzelentwicklung zu schaffen, wird diese Eigenschaft ebenfalls erst bei Realisierung der SWPL berücksichtigt.

Eine FP Analyse verläuft nun einzeln für jede von der Komponente angebotene Funktion. Zunächst muss die Funktion und die geplante Realisierung im Detail verstanden werden. Dazu eignet sich der Entwickler oder ein Experte für solche Funktionen. Zusammen mit einem erfahrenen FP-Zähler kann die Bewertung einer einzigen Funktion im Allgemeinen innerhalb von 15 bis 20 Minuten erfolgen. Die Zeit verlängert sich, wenn der Experte zunächst in die FP Methodik eingeführt werden muss. Im vorliegenden Fall bietet die GSM-Komponente eine allgemeine Kommunikationsschicht an. Über diese wird die gesamte Kommunikation der Komponente mit dem GSM Modul gesteuert. Diese Funktion weist daher keine EI oder EO auf. Diese erfolgen durch die eigentlichen Funktionen selbst. Die

Kommunikation bietet allerdings einen Algorithmus an, der die gesendeten Informationen jeweils umwandelt, entweder aus der Fahrzeugkommunikation in verständliche Signale für die GSM Komponente, oder aber die EOs der GSM Komponente werden umgewandelt, sodass sie über die Fahrzeugkommunikation versendet werden können. Aufgrund der klaren Abgrenzung der Einzelentwicklung sind die Anforderungen für die Umwandlung klar definiert. Dennoch ergeben sich eine Vielzahl von Regeln, weshalb die Komplexität als mittel einzustufen ist. Ein zweiter (komplexer) Algorithmus wird verwendet, um die Autorisierung der Anfragen, die von außerhalb in das System kommen, zu überprüfen. Die entsprechenden Codes sind bei den jeweiligen Funktionen berücksichtigt. Kommunikation filtert die Anfragen entsprechend und leitet lediglich diejenigen mit Berechtigung weiter. In einem Logfile werden Besonderheiten und Fehler für eine spätere Analyse erfasst, daher ist ein DF notwendig. Da dieses aus weniger als 19 Parameter und Variablen je Datensatz besteht, erfolgt die Komplexitätseinstufung als gering. Schließlich sind Schnittstellen zum Fahrzeug selbst erforderlich. Die Funktion Kommunikation übernimmt vollständig die Interaktion mit dem vorhandenen GSM Modul. Da zahlreiche Eigenschaften des GSM Moduls genutzt werden (z.B. SMS-Versand, Zugriff von Außen) erfolgt eine mittlere Einstufung der Komplexität. Weiterhin ermittelt Kommunikation die aktuelle Position des Fahrzeugs vom Navigationssystem. Diese GPS Position ist klar definiert, daher ist dieses EIF als gering komplex eingestuft. Andere Funktionen innerhalb der Komponente können auf diesen Wert zugreifen. Allerdings nicht als gespeicherter Wert, sondern bei jeder internen Anfrage ermittelt Kommunikation einen aktuellen Wert.

Die Standortbestimmung wird durch einen EI aktiviert. Dies wäre im konkreten Fall eine Anfrage des Besitzers per SMS an das GSM Modul des Fahrzeugs. Über Kommunikation wird diese Anfrage als EI an Standortbestimmung weitergeleitet. Die Anfrage ist klar abgegrenzt und enthält weniger als 4 Parameter und ist daher gering komplex. Mittels eines einfachen Algorithmus wird dann die Abfrage der aktuellen Position gestartet und daraus eine Antwort generiert. Dazu werden weniger als 16 Regeln benötigt, was eine geringe Komplexität bedeutet. Das System kann auf ein DF zugreifen, in dem Textbausteine für eine Antwort gespeichert sind. Außerdem findet sich dort die Autorisierungsinformation zur Überprüfung der Gültigkeit einer Anfrage. Zum Schluss erfolgt ein EO in Form einer SMS, die an den Besitzer des Fahrzeugs versendet wird. Dieser enthält weniger als 6 Parameter und wird daher als gering komplex eingestuft.

Für die Stilllegung des Fahrzeugs ist ebenfalls ein EI notwendig, der als SMS erfolgt. Zur Autorisierung ist ein DF vorhanden. Ein mittel komplexer Algorithmus überprüft, ob eine sichere Stilllegung erfolgen kann und kümmert sich um die Verschlüsselung der Kommunikation mit dem MotorECU. Daher wird auch ein EIF benötigt. Als Ergebnis generiert Stilllegung 2 EOs, zunächst die eigentliche Stilllegung des Fahrzeugs und zweitens wird eine SMS als Bestätigung an den Besitzer versendet.

Für die Funktion Aktionsradius sind 2 EIs möglich. Zunächst kann die Grenze des erlaubten Aktionsradius eingestellt werden. Weiterhin kann die Funktion mittels SMS ein- und ausgeschaltet werden. Beide EIs sind gering komplex. Für die Autorisierung gibt es erneut ein DF, das die benötigten Daten speichert. Dieses enthält zusätzlich Angaben zur erlaubten Grenze. Ein Algorithmus berechnet aus der aktuellen Position, die über Kommunikation vom Navigationssystem angefragt wird, und des erlaubten Gebiets, ob sich das Fahrzeug innerhalb der Grenzen befindet. Sollte dies nicht der Fall sein, erfolgt eine Mitteilung an den Besitzer (per SMS) und im Navigationssystem wird eine Warnung angezeigt. Dementsprechend sind zwei gering komplexe EOs zu verzeichnen.

Name der Komponente:	GSM																		
	EI			EO			EQ			AL			DF			EIF			FP/
Funktionsbezeichnung	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	Funktion
													Total FP unadjusted			=		144	
Funktion																			
GSM																			
>Kommunikation											1	1	1			1	1		25
>Standortbestimmung	1			1						1			1						14
>Stilllegung	1			2							1		1			1			24
>Aktionsradius	2			2						1	1		1						24

Tabelle 36 – FP Zählung ohne Wiederverwendung für die GSM Komponente (ausgefülltes Worksheet, siehe auch Tabelle 29)

Die FP Analyse wurde für das gesamte Zugangssystem durchgeführt. Für die 5 geplanten Produkte ergeben sich die entsprechenden FP Werte bei Einzelentwicklung aus Tabelle 37. Insgesamt entsteht ein Aufwand von 833 FPs.

Produkte Name	Beschreibung	Summe FPs
Zugangssystem Fzg. 1	Basissystem, Zugang über Schlüssel, Wegfahrsperre und Möglichkeit Fenster zu öffnen	65
Zugangssystem Fzg. 2	Sicherheit, Zugang über Fernbedienung, personalisierbar, Alarmanlage, Wegfahrsperre, Homecoming Light, und programmierbare Höchstgeschwindigkeit je Fernbedienung, Fensteröffnung	150
Zugangssystem Fzg. 3	Komfort, Zugang über RFID, Wegfahrsperre, Personalisierung, Fensteröffnung möglich, Starten von Fkt. beim Einsteigen	101
Zugangssystem Fzg. 4	Mittel, Zugang über Fernbedienung oder RFID, Wegfahrsperre, Alarmanlage, Fensteröffnung, Starten von Funktionen beim Einsteigen, Homecoming Light, GSM Schnittstelle	206
Zugangssystem Fzg. 5	Komplett, Zugang über RFID, Alarmanlage, Wegfahrsperre, Homecoming Light, Personalisierung, programmierbare Höchstgeschwindigkeit, Fensteröffnung, Starten von Anwendungen beim Einsteigen, Externe Schnittstelle über GSM und WLAN nutzbar	311

Tabelle 37 – Vollständige FPs für geplante Produkte in 5 Jahren

5.4.2. Berücksichtigung von Variabilität

Variabilität ist das entscheidende Merkmal von SWPL, da die daraus resultierende Flexibilität die strategische Wiederverwendung von Komponenten im Rahmen einer SWPL erst ermöglicht. Variabilität darf aber nicht einfach in den Systemen vorhanden sein. Es bedarf einer Abgrenzung der Variabilität, sodass die entsprechenden Bausteine frei zu einem neuen Produkt zusammengefügt werden können. "Die Isolierung von Variabilität in Komponenten lässt sich für einen Großteil aller Variabilitätsarten realisieren und entspricht letztendlich der reinen komponentenbasierten Entwicklung. Es ist der dominierende und in der Praxis am häufigsten anzutreffende Ansatz bei der Behandlung von Variabilität. Aus diesem Grund sind Produktlinien in der Regel auch komponentenbasierte Systeme" [Muth04, S. 112].

Dieser Idee wird auch in der vorliegenden Arbeit gefolgt. Die Realisierung von Variabilität kann daher bei einer zweiten FP Berechnung auf Komponentenbasis als zusätzlicher Aufwand berücksichtigt werden. Um eine vollständige Übersicht zu geben werden zunächst alle Variabilitätstypen (siehe Kapitel A 2.2.3) einzeln betrachtet und die Möglichkeiten der Berücksichtigung mittels FPs erklärt (5.4.2.1). FPs haben die Eigenschaft, dass sie den Aufwand unabhängig von der gewählten Implementierungstechnik ermitteln [Long01, S.11]. Sie ermitteln ein Maß für die Komplexität der zu realisierenden Funktionen aus Sicht des Anwenders. Demnach spielt es keine Rolle, welche Programmiersprache oder welche Programmiertechnik verwendet worden ist, um die Funktionen zu realisieren. Der tatsächliche

Aufwand kann dadurch natürlich abweichen. Daher können FP Ergebnisse aus verschiedenen Projekten oder insbesondere zwischen Unternehmen nicht ohne weiteres miteinander verglichen werden. Für die hier vorliegende Arbeit ist diese externe Vergleichbarkeit jedoch nicht relevant, da lediglich ein Vergleich zwischen Einzelproduktentwicklung und der Realisierung mittels SWPL erzielt werden soll. Und es kann davon ausgegangen werden, dass ein Unternehmen die gleiche Technik verwenden wird unabhängig davon, ob die Produkte einzeln entwickelt werden oder auf Basis einer SWPL. Dies ist Voraussetzung für die Anwendung des e-FACTS Modells. Um also einen Vergleich zu ermöglichen muss auch die Implementierung von Variabilität unabhängig von der gewählten Technik berücksichtigt werden. Dennoch wird zusätzlich ein Einblick in die Implementierung von Variabilität auf Code Ebene gegeben (5.4.2.2).

5.4.2.1. Implementierungsunabhängige Variabilitätstypen

[Bühn04, S.14ff.] nennt 6 Typen von Variabilität, für die im Folgenden jeweils dargelegt wird, wie der entsprechende Mehraufwand bei der Integration mittels der FP Methodik erfasst werden kann:

- **Variabilität in Features:** Auf Basis einer einheitlichen Asset Base können neue Produkte durch Auswahl verschiedener Komponenten erzeugt werden. Dabei besteht die Möglichkeit, eine Funktion verschieden auszuführen. Bei den beschriebenen Zugangssystemen wäre das bspw. die Möglichkeit bei einem neuen Produkt zwischen unterschiedlichen Zugangsmethoden (mittels Schüssel, Fernbedienung oder RFID) frei zu wählen. Der zusätzliche Aufwand entsteht nun nicht durch die neue Funktionalität. Diese kann wie schon bekannt mittels FP Berechnung erfasst werden und ist für Einzelprodukt oder SWPL gleich. Ein Unterschied ergibt sich, da bei einem Einzelprodukt genau definiert ist, welche Funktion integriert wird. Die Assets einer SWPL müssen jedoch die Möglichkeit bieten, flexibel in verschiedenen Produkten einsetzbar zu sein. Je nach konkreter Situation bedeutet dies, dass die Komponenten zusätzliche EIFs, zusätzliche EIs oder EOs und umfangreichere DFs aufweisen müssen.
  Schnittstellen könnten bspw. notwendig sein, wenn ein Zugangssystem sowohl in Autos mit CAN-Bus als auch mit dem neueren Flexray Bus eingebaut werden sollen. Wenn dies bei der Planung der SWPL bekannt ist, kann statt einer zusätzlichen Schnittstelle auch eine Middleware Schicht in der Architektur eingezogen werden, die

diese Schnittstellenaufgaben übernimmt. Dadurch ergibt sich zwar ein zusätzlicher Aufwand für die Erstellung dieser Middleware, die einzelnen Komponenten werden jedoch von der Aufgabe, mehrere Schnittstellen zur Verfügung zu stellen, entlastet. Umfangreichere DFs könnten notwendig sein, um eine Parametrisierung von Funktionen zu ermöglichen. Bietet ein OEM bspw. seinen Kunden die Möglichkeit, Funktionen auch nach dem Kauf freischalten zu lassen, dann ist ein entsprechendes Management der Freigaben erforderlich.
Und schließlich könnten zusätzliche EIs und EOs entstehen, wenn der Kunde die Möglichkeit hat, während der Benutzung zwischen den Funktionen umzuschalten. Bei einem Auto das sowohl eine Fernbedienung als auch RFID als Zugangsmedium nutzt könnte es möglich sein, eine der Alternativen auszuschalten.

- **Variabilität in Abläufen:** Dieser Typ ist insbesondere für Zulieferer relevant, die Produkte auf Basis einer SWPL an mehrere OEMs liefern möchten. Die OEMs geben für ihre Produkte die Abläufe vor. Die Zulieferer müssen ihre Produkte entsprechend anpassen. Bei einer SWPL bedeutet dies, dass bestimmte Prozesse nicht fest vorgegeben sein dürfen. Daraus folgt ein höherer Aufwand, da alle möglichen Kombinationen abbildbar sein müssen. Ein Prozess darf also nicht auf ein bestimmtes Ereignis warten, sondern muss alternative Lösungswege kennen. Im Sinne der FP Methodik sind diese Aufwände durch komplexere Algorithmen erfassbar. In einem DF sind die möglichen Abläufe zu speichern und die Alternativen zu nennen. Das Ergebnis aus den Berechnungen kann ein entsprechend komplexerer EO sein, der mehr Parameter und Variablen erhält als nötig, bspw. wenn dieser EO auch zu einem späteren Zeitpunkt möglich wäre mit dann zusätzlich verfügbaren Informationen.
Als Beispiel aus der Automobilindustrie kann das Starten von Anwendungen beim Öffnen des Fahrzeugs genutzt werden. Die Reihenfolge, in der die Systeme eines Autos beim Start zur Verfügung stehen, kann je nach OEM abweichen. Für den Zulieferer eines Navigationssystems kann es daher notwendig sein, die internen Startabläufe anzupassen, um eine möglichst geringe Startzeit zu erhalten. Ein Grund kann sein, dass notwendige Systeme nicht rechtzeitig zur Verfügung stehen. Ein Standardsystem würde entsprechend warten bis das System verfügbar ist. Ein flexibles System sieht Alternativen für diesen Fall vor. Im Ergebnis sind komplexere Algorithmen notwendig, die überprüfen, welche Systeme wann verfügbar sind. Die

entsprechenden Daten über die verschiedenen Systeme und alternative Startprozesse sind in einem zusätzlichen DF zu speichern.

- **Variabilität im Datenumfang:** Wenn mittels Variabilität unterschiedliche Datenobjekte oder Attribute verfügbar gemacht werden können, dann müssen diese zunächst alle existieren. Dieser Aufwand ist jedoch bei einer Einzelentwicklung und einer Entwicklung als SWPL Asset gleich. Allein der Aufwandsunterschied durch die Tatsache, dass das Asset mehrere Varianten anbieten kann, ist relevant. Im Sinne der FP Methodik manifestiert sich dieser durch komplexere DFs, da diese mehr Variabeln, Parameter, Zustände und Substrukturen enthalten können, oder durch zusätzlich notwendige DFs, die den jeweiligen Zustand und die zulässigen Konfigurationen enthalten. Eventuell sind auch zusätzliche oder komplexere ALs notwendig, wenn bspw. die Zulässigkeit von Konfigurationen während des Betriebs überprüft werden muss.
  Die Standortbestimmung von Fahrzeugen ist ein Beispiel: Aufgrund von unterschiedlichen Datenschutzbestimmungen kann es erforderlich sein, die Funktion der Standortbestimmung mit unterschiedlichen Attributen auszustatten, um den Detailgrad der gesendeten Information an die lokalen Vorgaben anzupassen. Dementsprechend gibt es ein zusätzliches DF mit der aktuellen Konfiguration. Zusätzlich ist der Algorithmus zur Erstellung des EO anzupassen, indem bspw. zusätzliche Regeln eingeführt werden.

- **Variabilität im Datenformat:** Wenn mehrere Datenformate unterstützt werden sollen, dann beeinflusst dies die DFs. Diese müssen Informationen zu den einzelnen Formaten enthalten, aber auch Informationen, welche Formate unterstützt werden und welche Wechsel in welcher Reihenfolge möglich sind. Gleichzeitig müssen die ALs in der Lage sein, unterschiedliche Formate zu erkennen und die weiteren Prozessschritte entsprechend anpassen. Ein zusätzliches Datenformat kann entweder einen neuen EI oder die Erhöhung der Komplexität des bestehenden EIs zur Folge haben.
  Als Beispiel wird erneut auf die Zugangssysteme im Automobil zurückgegriffen. Ein SWPL für Wegfahrsperren muss die unterschiedlichen Datenformate der Motorsteuergeräte beherrschen. Eine Lösung ist die Individualisierung über eine Konfigurationsdatei und eine entsprechende Parametrisierung. Auch die Algorithmen müssen entsprechend flexibel gestaltet sein.

- **Variabilität im Systemzugang:** Das obige Beispiel für Variabilität in Features beinhaltet ebenfalls unterschiedliche Systemzugänge. Daher können die zusätzlichen Aufwände ebenfalls mittels FP Methodik erfasst werden. Die möglichen Veränderungen sind mit denen bei Variabilität in Features identisch.
  Zur Veranschaulichung dient ein weiteres Beispiel: Bei Zugangssystemen kann die externe Schnittstelle als GSM oder WLAN realisiert werden. Wenn beide Schnittstellen zur Verfügung stehen sollen, ist eine Konfigurationsdatei notwendig, die Interaktionen zwischen diesen beiden Schnittstellen regelt, z.B. über welche Technik wird übertragen, welches Signal hat Vorrang in der Bearbeitung. Zusätzlich kann ein Umschalten durch den Nutzer oder aufgrund eines Signals einen zusätzlichen EI darstellen. Wird der Status des Systems angezeigt, bedeutet dies einen zusätzlichen EO.

- **Variabilität in Schnittstellen:** Wenn mehrere Schnittstellen zu berücksichtigen sind, kann dies über Konfigurationsdateien und Parametrisierung gelöst werden. Der zusätzliche Aufwand ergibt sich aus komplexeren ALs, um die Schnittstellen zu erkennen und auf die entsprechende Kommunikation umzustellen. Die Einführung einer Middleware hat sich in diesem Fall als vorteilhaft erwiesen, da diese die Aufgaben der Kommunikation über die Schnittstellen zentral übernehmen kann und die einzelnen Funktionen dadurch entlastet werden.
  Ein anschauliches Beispiel sind Navigationssysteme. Der Kern der Funktionalität, nämlich die eigentliche Routenberechnung, ist weitgehend einheitlich gelöst, die Anzahl möglicher Schnittstellen hingegen ist hoch. Dies umfasst die Darstellung der Route, z.B. mittels Pfeilen, als Kartensicht oder als 3D Navigation, die zusätzlichen Informationsquellen, z.B. TMC zur Stauumfahrung, aber auch die Bedienung des Systems, z.B. über Knöpfe, Touchscreen oder Spracheingabe. Hersteller von Navigationssystem stehen daher vor der Aufgabe, mittels einer geeigneten Architektur eine einfache Anbindung über verschiedene Schnittstellen zu ermöglichen. Die individuelle Anpassung eines Produkts kann über Konfigurationsdateien erfolgen (Parametrisierung).

- **Variabilität in Qualität:** Verschiedene Qualitäten der SW Produkte ergeben sich direkt aus den bereits besprochenen Variabilitätstypen. Bspw. können unterschiedliche

Qualitäten mittels mehrerer Funktionen abgebildet werden. Daher entfällt eine individuelle Betrachtung.

Es konnte gezeigt werden, dass die Mehraufwände durch alle Typen von Variabilität mittels der FP Systematik erfasst werden können. Die Herleitung beschränkte sich allerdings auf eine implementierungsunabhängige Sichtweise. Im folgenden Kapitel wird daher näher auf die tatsächliche Realisierung von Variabilität auf Code Ebene eingegangen.

5.4.2.2.Realisierung von Variabilität

[Halm02] bezeichnet die bisher verwendete Gliederung von Variabilitätstypen als fachliche Variabilität. Er fügt darüber hinaus die technische Variabilität hinzu. Diese wird in drei Bereiche weiter unterteilt: Variabilität im Bereich der IT Infrastruktur (z.B. den Resourcenbedarf), Zeitpunkt der Konkretisierung der Variabilität (früh bei der Produktdefinition oder spät bei der Installation oder sogar erst bei der Nutzung) und die Realisierung selbst. Gerade der letzte Punkt ist von besonderem Interesse, da "literature has already addressed how to create and instantiate generic product line assets, such as domain models and architectures to generate instance specific ones, yet little attention has been given on how to actually deal with this genericity at the code level" [Anas01, S. 109].

Obwohl die FP Methodik von der eigentlichen Implementierung abstrahiert, soll im Folgenden kurz gezeigt werden, dass die im Bereich der Automobilindustrie bedeutenden Methoden zur Realisierung von Variabilität mit FPs berücksichtigt werden können. [Anas01] beschreibt insgesamt 13 Ansätze zur Implementierung von Variabilität auf Code Ebene. Zahlreiche Ansätze wie Aspect-Oriented-Programming oder Dynamic Class Loading sind für die Software in Embedded Systems im Automobil weniger interessant, da sie stark auf moderne Programmiersprachen wie Java abzielen. Hier sind Ansätze wie Parametrisierung, Delegation, Vererbung und Conditional Compilation bedeutender. Diese werden kurz erklärt:

- **Parametrisierung:** Über Parameter im Code können mittels einer Konfigurationsdatei die Funktionen, die im Produkt enthalten sein sollen, ausgewählt werden. Die eigentliche Bindung der Variabilität kann zum Zeitpunkt der Produktdefinition oder bei Laufzeit erfolgen. Die FP Methodik kann den entsprechenden Mehraufwand über die notwendigen DFs und deren höhere Komplexität erfassen. Je nach Realisierung kann auch ein zusätzlicher AL notwendig sein, z.B. wenn die Bindung bei Laufzeit erfolgt und eine Überprüfung der zugelassenen Zustände notwendig ist.

- **Delegation:** In diesem Fall werden die Funktionen selbst nicht variabilisiert. Es erfolgt allerdings eine Erweiterung, sodass Funktionen, die eine bestimmte Anfrage nicht erfüllen können, die Anfrage an die geeignete Funktion weiterleiten. Dadurch wird es möglich, dass eine Funktion virtuell jegliche Funktionalität des Gesamtsystems abdeckt. Ermöglicht wird dies, indem die delegierende Funktion Referenzen zu den entsprechenden Funktionen bereithält, die angefragte Funktion erkennt und daraus das richtige Delegationsobjekt identifiziert. Dementsprechend lassen sich der Mehraufwand und die Representation per FPs ablesen, nämlich DFs und ALs.

- **Vererbung:** Vererbung benutzt Klassen, um Funktionen zu modularisieren. In Super-Klassen werden ähnlich der Logik bei SWPL nur die gemeinsamen Funktionen realisiert. In Unter-Klassen wird die hauptsächliche Funktionalität festgelegt, dabei sind allerdings die Rahmenbedingungen der Super-Klassen einzuhalten. Demnach wären die Funktionstypen bereits bei der Definition auf der Super-Klasse zu zählen. Allerdings enthalten diese meist keine realen Funktionen. Diese werden erst in den Unter-Klassen festgelegt. Die FP-Zählung sollte demnach in der Klasse erfolgen, die eine wirklich nutzbare Funktion enthält. Bei zusätzlichen Erweiterungen in Unter-Klassen können die Veränderungen über eine Steigerung der Komplexität der Funktionstypen abgedeckt werden. Sollten gänzlich neue Funktionstypen definiert werden, sind diese an der jeweiligen Stelle neu zu berücksichtigen. Wichtig ist, dass keine Berücksichtigung der gesamten Funktionalität erfolgen darf, lediglich die Änderungen sind zu berücksichtigen, da sonst eine Doppelerfassung vorliegen würde.

- **Conditional Compilation:** Per Anweisungen wird festgelegt, welche Codebestandteile bei einer bestimmten Kompilierung tatsächlich verwendet werden. Dies erfasst die FP Methodik durch ein zusätzlich notwendiges DF, in dem die Anweisungen gespeichert werden.

5.4.3. Feature Point Zählung mit Wiederverwendung

In Kapitel 4.4.2 wurde die notwendige Vorarbeit geleistet, um die FP Methodik zur Bewertung wiederverwendbarer SW anzuwenden. Die Methodik ist in der Lage, den durch die eingebaute Variabilität entstehenden Mehraufwand zu berücksichtigen und entsprechend in FPs abzubilden.

Für die Zählung mit Wiederverwendung können zwei Verfahren genutzt werden: mit und ohne Berücksichtigung einer speziellen SWPL Architektur. Da die FP Methodik Variabilität berücksichtigen kann, können direkt auf Basis der Komponentenliste Zählungen durchgeführt werden, die die notwendige Variabilität und den daraus resultierenden Mehraufwand bei der Erstellung berücksichtigen. Einige Beispiele aus den vorherigen Kapitel haben allerdings gezeigt, dass eine spezielle SWPL Architektur große Vorteile bieten kann, z.B. durch die Einführung einer Middleware, um häufig genutzte Tätigkeiten aus den Komponenten zu lösen und zentral anzubieten. Zählt man direkt die FPs mit Wiederverwendung, dann berücksichtigt man den Mehraufwand für die Integration der Variabilität, lässt aber eventuell Einsparpotenzial durch eine SWPL Architektur ungenutzt, da bspw. EIFs je Funktion zu realisieren sind und entsprechend gezählt werden müssen. Andererseits erzeugt die Erstellung einer SWPL Architektur selbst ebenfalls FPs. Aber die zur Verfügung gestellten Funktionen, z.B. Schnittstellen können von allen Funktionen genutzt werden und müssen nicht separat realisiert werden. Die Architektur-Methode setzt ein besseres Verständnis des SWPL Konzeptes und der geplanten SWPL im Speziellen voraus. Unternehmen, die nicht über dieses Wissen verfügen, können e-FACTS dennoch anwenden, wenn sie die einfache Berechnungsmethode ohne Berücksichtigung der SWPL Architektur benutzen. Das Ergebnis wird konservativer ausfallen, was aufgrund der mit einer SWPL Einführung verbundenen, grundsätzlichen Unsicherheit als Vorteil zu werten ist. Das alternative Vorgehen dient dazu, die sehr frühe Anwendbarkeit des Modells zu gewährleisten, indem keine vollständige Ausarbeitung der SWPL Architektur benötigt wird. Aufgrund der besseren Ergebnisse wird jedoch die Architektur-Methode empfohlen.

5.4.3.1.Ohne SWPL Architektur

Die zukünftigen Produkte wurden bereits festgelegt. Die Aufwände für eine Einzelentwicklung stehen ebenfalls fest. Allerdings ist offen, wie diese ökonomisch sinnvoll entwickelt werden können. Es fehlen die Aufwände zur Erstellung einer SWPL. Um eine möglichst exakte Aussage zu erhalten, muss die Struktur feststehen. Allerdings ist mit dem Design einer SWPL, insbesondere mit deren Architektur, ein erheblicher Aufwand verbunden. Das e-FACTS Modell hat das Ziel, eine möglichst frühe Entscheidungshilfe zu geben, bevor der Großteil der Entwurfsentscheidungen einer SWPL getroffen wurde. Und die Architektur einer SWPL ist die wichtigste Designentscheidung bei der Entwicklung. Eine Wiederverwendung ist auch ohne eine optimale SWPL Architektur möglich. Es ist nur nicht möglich, den optimalen Wert aus der Wiederverwendung der Assets zu ziehen. Grund dafür

ist, dass ohne eine optimale Architektur entweder mehr Komponenten variabel gestaltet worden sind als nötig, oder – und das ist der weitaus gewichtigere Punkt – es wurden nicht die kritischen Elemente identifiziert, die den Kern der Wiederverwendung bilden. Dieser Kern muss in einer Form vorliegen, die mit geringem Aufwand an unterschiedliche Rahmenbedingungen anpassbar ist. Je besser man diese Rahmenbedingungen beschreiben und einschränken kann und je besser die Separierung von Standard- und Einzelfunktionen gelingt, desto größer kann der Vorteil aus einer SWPL ausfallen.

Die FP Analyse ist eine Methode, die die Realität bewusst soweit vereinfacht, dass ein ausreichend genaues Ergebnis erzielt wird. Der Vorteil der Abstrahierung ist eine deutliche Reduzierung des Analyseaufwands. Diesen Weg geht auch das e-FACTS Modell bei der Analyse der Wiederverwendung ohne SWPL Architektur. Es wird bewusst eine Ungenauigkeit in Kauf genommen, um die Analyse zu beschleunigen. Dabei ist zu beachten, dass ein solches Vorgehen immer konservativ sein sollte. Dadurch wird eine Sicherheit eingebaut, die die erhöhte Ungenauigkeit des Ergebnisses kompensiert.

Zur Abschätzung des Aufwands einer SWPL ohne eine optimale Architektur wird angenommen, dass sämtliche Funktionen bzw. Komponenten variabel zu gestalten sind mit Ausnahme der Komponenten, die bereits während der Erstellung der Produktübersicht ausgeschlossen worden sind. Diese Annahme erhöht den Entwicklungsaufwand auf den Maximalwert, der notwendig wäre, um Wiederverwendung zu ermöglichen. Sollte ein Vergleich dieses Maximalaufwands mit dem Aufwand der Einzelentwicklungen eine Vorteilhaftigkeit der SWPL ergeben, dann ist die SWPL vorzuziehen. Allerdings sollte vor der Einführung eine gründliche SWPL Architekturdefinition erfolgen. Die daraus erzielbaren zusätzlichen Aufwandsreduzierungen versprechen einen deutlich größeren Vorteil als die dafür einzusetzende Zeit. Die Motorsteuerung liefert ein Beispiel für eine Funktion, die besser für jedes Produkt einzeln entwickelt wird: Wird die Kommunikationsschnittstelle aus der Domain einer Motorsteuerungs-SWPL herausgelöst, dann ist für jedes Produkt "eine produktspezifische Anpassung der Kommunikationsschnittstelle an die fahrzeugspezifische CAN-Matrix nötig" [Hein04, S. 205]. Die Schnittstelle wird also bewusst auf Produktebene erstellt, denn erst mit dem Produkt stehen die erforderlichen Parameter fest.

In Kapitel 5.4.2 wurde gezeigt, dass der zusätzliche Aufwand, um Variabilität in SW zu realisieren, mittels FPs erfasst werden kann. Dabei ist zu beachten, dass FPs per Definition den gesamten Aufwand eines SW Erstellungsprozesses abdecken, vom Design, über das

Schreiben des Codes bis zum Testing. FPs haben eine klare Funktionsorientierung, d.h. sie betrachten allein die Funktionen einer SW aus Sicht eines Nutzers, unabhängig von der tatsächlichen Realisierung. Aufwandsunterschiede durch die Wahl einer anderen Programmiersprache oder umständliche Programmierung können nicht erfasst werden. Ebenso ist die Erfassung auf Aufwände reduziert, die tatsächlich durch die Realisierung von Funktionen begründet sind. Dies bedeutet, dass FPs keine Auskunft über die Kosten der organisatorischen Veränderungen geben.

Weiterhin stellt die Erfassung des Aufwands durch die tatsächliche Wiederverwendung eines Assets eine Herausforderung dar. Hierbei werden keine neuen Funktionen erstellt, die über FPs erfasst werden könnten. Es erfolgt lediglich eine Anpassung vorhandener SW-Komponenten, um eine Nutzung in einem neuen Produkt zu ermöglichen. Dieser Aufwand kann daher nicht direkt über FPs erfasst werden. Es besteht jedoch die Möglichkeit einer prozentualen Abschätzung. Dies stellt gegenüber vorhandenen Methoden trotz der an früherer Stelle geäußerten Kritik an der Verwendung von Standard-Prozentsätzen eine Verbesserung dar, denn die Basis der Berechnung wurde durch die Verwendung von FPs auf Funktionsebene verbessert. Dadurch besteht die Möglichkeit, den Aufwand der tatsächlichen Wiederverwendung individuell je Funktion zu bestimmen. Dies erhöht den Aufwand, ermöglicht aber ein genaueres Ergebnis.

Wie in Kapitel 5.4.2 gezeigt eignen sich FPs für die Berechnung des Mehraufwands für die Erstellung wiederverwendbarer SW. Die FP-Veränderungen gegenüber Einzelentwicklungen können – wie bereits beschrieben – vielfältig sein, da auch die Möglichkeiten für die Integration von Variabilität zahlreich sind. In der Automobilindustrie wird häufig eine Parametrisierung verwendet, um Variabilität zu integrieren, z.B. wenn einzelne Funktionen variabel gestaltet werden sollen. Mittels Parametern im Code können bestimmte Codeelemente bei einer Kompilierung ausgelassen werden. Diese Parameter und die Kombinationsmöglichkeiten sind zu erfassen. In der FP Logik bedeutet dies zusätzliche DFs. Die SW, die nachher im Automobil verwendet wird, enthält dann lediglich die tatsächlich benötigten Elemente. Dieses Vorgehen passt zu speziellen Anforderungen für Embedded Systems im Automobil, wo Ressourcen knapp sind. Zur Variabilisierung von Funktionen in der SW des Fahrzeugs selbst wird Parametrisierung daher selten verwendet. Schließlich würde dies bedeuten, ungenützten Code zu speichern. Im Bereich Multimedia sind solche Ansätze allerdings möglich, da hier HW Ressourcen weniger beschränkt sind.

Wie bereits bei der Zählung ohne Wiederverwendung können auch anteilige Entwicklungskosten bei der Nutzung von bestehenden Produkten berechnet werden. Im Gegensatz zur Einzelentwicklung sind jedoch meist größere Anpassungen notwendig, wenn existierende Komponenten Teil der Asset Base werden sollen. Bei der ursprünglichen Entwicklung ist diese Anforderung nicht berücksichtigt worden.

Anwendungsbeispiel:
Bei der Variabilisierung der Komponenten ergeben sich mit der FP Methodik Veränderungen bei allen Funktionstypen, außer bei EQs, weil dieser Funktionstyp sehr selten verwendet wird. Die größten Veränderungen finden sich bei den EIFs und den EIs. Bei diesen spielt die Verknüpfung mit anderen Komponenten eine wichtige Rolle. EIFs sind bei einer SWPL im Allgemeinen komplexer aufgebaut, da die Anforderungen für eine Kommunikation mit anderen Komponenten des Fahrzeugs nicht fest definiert sind. Dadurch erhöht sich die Zahl der Parameter und Variablen, aber auch die Zahl der benötigten Substrukturen. Beim Starten von Anwendungen kann Variabilität z.B. bedeuten, dass Systeme angesprochen werden müssen, die in anderen BUS-Systemen organisiert sind.[81] Bei EIs zeigt sich die Variabilität ebenfalls durch eine höhere Zahl an Parametern und Variablen, wodurch der Komplexitätsgrad ansteigen kann. Je nach Fahrzeugkonfiguration kann ein EI abweichende Parameter enthalten, z.B. kann die Funktion Homecoming Light unterschiedliche Intervalle anbieten, bevor sich das Licht ausschaltet. In einer anderen Konstellation ist die Funktion nicht konfigurierbar, sondern kann bei Bedarf lediglich abgeschaltet werden. Das gleiche gilt für EOs, denn hier ist über zusätzliche Parameter und Variablen sicherzustellen, dass das Ergebnis im Fahrzeug korrekt verarbeitet wird. Bei der Funktion Höchstgeschwindigkeit erfolgt bspw. ein Warnhinweis für den Fahrer. Der EO muss Angaben dazu enthalten, wie diese Information anzuzeigen ist, z.B. nur im Kombiinstrument oder auch im Navigationssystem. Algorithmen können zusätzliche Berechnungen für Umwandlungen oder zusätzliche Regeln enthalten, mit denen überprüft wird, ob eine Aktion für eine bestimmte Ausgestaltung von Variabilitätspunkten überhaupt zulässig ist. Und schließlich können zusätzliche oder komplexere DFs notwendig sein, z.B. um eine zusätzliche Parametrisierung von Funktionen zu ermöglichen. Eine Übersicht der FPs für die SWPL Zugangssysteme gibt Tabelle 38.

---

[81] Bei modernen Fahrzeugen sind Multimediasysteme nicht mehr mittels eines CAN-Busses angeschlossen, sondern über den moderneren MOST- oder sogar einen Flexray-Bus. Diese bieten eine höhere Bandbreite, skalierbare Fehlertoleranzen und Echtzeitfunktionen. Für Details siehe z.B. [Reic05].

Nr.	Featurebezeichnung	FP \WV	EI			EO			EQ			AL			DF			EIF			FP/ Fkt.	Veränd.
			L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H		
			Total FP(resuse) unadjusted																=	415		
1	Zugang	25		1			1						1		1	1			1		31	24%
2	Personalisierung	19		1			1					1			1				1		23	21%
3	Alarmanlage	18		1		2						1							1		21	17%
4	Wegfahrsperre	26		1			1						1			1				1	29	12%
5	Homecoming Light	19	1			1						1				1			1		23	21%
6	Höchstgeschwindigkeit	29	2	1		2						1			1				1		32	10%
7	Fensteröffnung	14		1		1						1							1		17	21%
8	Starten von Anwendungen	17	1				1			1		1	1		1					1	32	88%
9	GSM	87																			117	34%
10	>Kommunikation	25											1	1	1	1			1	1	37	48%
11	>Standortbestimmung	14	1			1						1			2						19	36%
12	>Stilllegung	24	1			2							1		2				1		31	29%
13	>Aktionsradius	24	1	1		2						1	1		2						30	25%
14	WLAN	57																			90	58%
15	>Kommunikation	15											1		1	1				1	25	67%
16	>Update von Daten	20			1	1							1		2				1		30	50%
17	>Konfiguration	22			1	1							1		1		1		1		35	59%

Tabelle 38 - FP Zählung Zugangssysteme als SWPL ohne Architektur (vollständig ausgefülltes Worksheet für Anwendungsbeispiel, siehe auch Tabelle 29)

Die zusätzliche letzte Spalte zeigt die prozentuale Veränderung gegenüber der Einzelentwicklung. Die einzelnen Funktionen weisen alle eine höhere FP Zahl auf. Der Mehraufwand liegt zwischen 10 und 88%. Der höchste Wert wird von der Funktion Starten von Anwendungen erreicht. Hier ergeben sich durch die unterschiedlichen Prozessabläufe beim Starten eines Fahrzeugs hohe Anforderungen an die Variabilität, die zu zusätzlichen FPs führen. Ein zusätzlicher AL ist erforderlich, der überprüft, ob das Starten in dieser Reihenfolge überhaupt möglich ist. Ein DF speichert die erlaubten Abfolgen ab und liefert die zu beachtenden Einschränkungen.

Vor einem Vergleich der SWPL mit den Einzelentwicklungen ist nun noch der Aufwand für die tatsächliche Wiederverwendung der Assets zu bestimmen. Wie bereits erläutert, kann dies über eine prozentuale Bewertung von einzelnen Funktionstypen erfolgen. Dieser Schritt wird jedoch erst in Kapitel 5.4.4 bei der Optimierung der Assets besprochen. In der Zwischenzeit werden diese Kosten vernachlässigt. Im Beispiel beträgt die Summe der FPs für die SWPL 415 FPs im Vergleich zu 833 FPs bei Einzelentwicklung. Die Einsparung beläuft sich auf ca. 50%. Das Beispiel erhebt keinen Anspruch auf Vollständigkeit und dient hauptsächlich zur Veranschaulichung des Ablaufs. Daher können aus den Ergebnissen keine wirklichen Schlüsse gezogen werden. Würden diese Werte allerdings bei einer realen Analyse auftreten, dann wäre die SWPL bereits an dieser Stelle zu empfehlen, obwohl keine optimierte Architektur berücksichtigt worden ist. Die SWPL erreicht den Break Even, also den Punkt bei

dem sich die Integration von Variabilität auszahlt, mit dem vierten Produkt. Dies liegt etwas unter dem Durchschnittswert aus bisherigen Anwendungen. In der Literatur geht man im Allgemeinen davon aus, dass der Break Even Punkt zwischen der zweiten und dritten Wiederverwendung liegt [vgl. Clem03, Weis99, S. 45ff.]. Im hier beschriebenen Fall ist der späte Break Even Punkt nicht verwunderlich, da die Produkte einen sehr unterschiedlichen Funktionsumfang aufweisen und der Umfang mit späteren Produkten deutlich zunimmt. So weist Produkt 5 den fünffachen FP Umfang verglichen mit Produkt 1 auf.

Zum Abschluss ist festzuhalten, dass die Aufwandsbestimmung ohne eine dedizierte SWPL Architektur möglich ist, der Aufwand jedoch maximal geschätzt wird. Daher kann dieses Vorgehen für eine schnelle Analyse benutzt werden, wenn berücksichtigt wird, dass ein negatives Ergebnis zu einer falschen Entscheidung führen kann. Daher wird empfohlen, die SWPL Architektur bei der Bewertung zu berücksichtigen. Das entsprechende Vorgehen wird im nächsten Kapitel beschrieben.

5.4.3.2.Mit SWPL Architektur

Betrachtet man erfolgreiche Anwendungsbeispiele in der Literatur, dann zeigt sich, dass die SWPL Einführung in fast allen Fällen mit einer Änderung der SW Entwicklungsmethodik verbunden war.[82] Der Ansatz zur Erstellung der Software wurde grundsätzlich geändert. Erst dadurch werden die beobachteten, großen Verbesserungen möglich. Für einen Hersteller von Motorsteuerungssoftware bedeutet dies, die Ausrichtung grundsätzlich von einem Projektgeschäft in ein Produktgeschäft zu verändern. Nicht mehr die Anforderungen eines Kunden sind der Auslöser für die Produktentwicklung, sondern strategische Überlegungen des Herstellers hinsichtlich des Wiederverwendungspotenzials einzelner Komponenten. Maßgebend für diese Neuausrichtung ist die Entwicklung einer neuen SW Architektur. Durch die Identifikation von Gemeinsamkeiten und die klare Trennung von Wiederverwendung und Einzelentwicklung sind dramatische Verkürzungen der Entwicklungszeit möglich. Erst damit kann das volle Potenzial des SWPL Ansatzes realisiert werden.

---

[82] Die Robert Bosch GmbH hat für die Entwicklung ihrer Fahrerassistenzsysteme als SWPL eine neue variable Architektur entworfen. Dies wurde begleitet von einer neuen Entwicklungsstrategie, die darauf abzielte, "die Abhängigkeiten zwischen den entsprechenden Komponenten zu minimieren" [Hein04, S.200]. Cummins Inc., ein amerikanischer Hersteller von Diesel-Motoren für eine Vielzahl von Anwendungsgebieten, stellte bei der Einführung einer SWPL ebenfalls die komplette SW Architektur um, zumindest als Vision. In der Realität blieb die SWPL eine lose Ansammlung wieder verwendbarer Komponenten. "While their achievements under those conditions were laudable, they now understand that they failed to reap the benefits that a strong architecture would have afforded them. Therefore, they are now launching what they call the Core II architecture" [Clem02, S. 426]. Auch die beiden Anwendungsbeispiele in dieser Arbeit veränderten ihre Methodik der SW Erstellung im Rahmen der SWPL Einführung grundsätzlich (siehe Kapitel 6).

Wenn eine SWPL Architektur, vielleicht auch nur als grobe Idee, besteht, dann sollte sie für die Analyse benutzt werden. In der Berechnung der FPs führt dies zu einer zusätzlichen Komponente, der SWPL Architektur, deren Funktionen getrennt zu bewerten sind. Dieser zusätzliche Aufwand kann als Investition in die SWPL verstanden werden. Der Ertrag der Investition ergibt sich aus der Entlastung der eigentlichen Funktionen von Tätigkeiten, die in anderen Schichten der Architektur zur Verfügung gestellt werden. Für die FP Zählung bedeutet dies, dass einige Funktionstypen gegenüber einer Einzelentwicklung überflüssig werden, wie z.B. EIFs.

Zusätzlich zu einer neuen Architektur werden für eine SWPL häufig auch neue Werkzeuge verwendet, die die Entwicklung und den Umgang mit Variabilität in SW erleichtern. Werden diese als COTS[83] Systeme am Markt gekauft, dann kann der Kaufpreis direkt im Business Case berücksichtigt werden. Erfolgt hingegen eine Eigenentwicklung oder eine Anpassung einer COTS SW, dann ist der entsprechende Aufwand mittels einer FP Berechnung zu ermitteln. In der Automobilindustrie werden häufig Code Generatoren und Wrapping Tools zur Entwicklung eingesetzt. Weitere Werkzeuge dienen der Prozessunterstützung, wie z.B. Dokumentation und Konfigurationsmanagement.

Zur Berechnung der FPs einer SWPL unter Berücksichtigung einer SWPL Architektur wird der Aufwand zur Erstellung der Architektur explizit bewertet. Gemeint ist nicht allein der Entwurf, sondern die tatsächliche Umsetzung in Code. Denn allein die Funktionen, die sich daraus ergeben, können mit der FP Methodik bewertet werden. Anschließend sind die Bewertungen der einzelnen Funktionen durchzugehen und auf etwaige durch die Architektur verursachte Änderungen zu überprüfen. Die redundanten Funktionstypen werden gestrichen. Eventuell ergibt sich aber auch eine Reduktion der Komplexität, sodass eine Einstufung in eine niedrigere Komplexitätsklasse erfolgt, der Funktionstyp selbst aber erhalten bleibt.

Selbst entwickelte Werkzeuge können mittels FP Methodik bewertet werden. Allerdings kann die Wirkung dieser Werkzeuge nicht mittels FPs erfasst werden, da FPs Funktionalität messen. Ein Werkzeug, das dem Programmierer Informationen aufbereitet, wie z.B. eine bessere Übersicht über die zu erfüllenden Aufgaben oder ein Konfigurations-Management, ergibt keine mittels FP erfassbare Reduktion des Entwicklungsaufwands. Sollte der erwartete

[83] COTS steht für Commercial-off-the-shelf SW Produkte

Effekt signifikant sein, kann der Effekt indirekt über eine gestiegene Produktivitätskennzahl berücksichtigt werden.

Für die Berücksichtigung von Verbesserungen mittels der FP Methodik ergeben sich also zwei Möglichkeiten:

- Durch die SWPL Architektur reduziert sich die Anzahl oder die Komplexität der Funktionstypen. Da sowohl die FPs als auch der entsprechende Entwicklungsaufwand entfallen, verändert sich die Produktivität nicht.
- Die Einführung von Tools, deren Wirkung nicht über die Funktionstypen der FP Methodik erfassbar ist, verändert hingegen die Produktivität. Diese neue Produktivität nach Einführung eines neuen Tools ist mit einer separaten FP-Zählung zu bestimmen.

Dadurch ist sichergestellt, dass beide Verbesserungsarten vollständig berücksichtigt werden. Weitere Details dazu werden bei der Erstellung des Business Cases besprochen (Kapitel 5.5).

Beispiel Zugangssysteme: Der OEM hat im Anwendungsbeispiel eine Middleware in seine Architektur einbezogen. Diese Middleware bündelt alle Kommunikationsaufgaben der Zugangssysteme mit dem Fahrzeug selbst. Die individuelle Anpassung an ein bestimmtes Fahrzeug erfolgt individuell, ist also nicht Teil der SWPL. Dadurch müssen die einzelnen Komponenten keine oder fast keine Kommunikationsaufgaben erfüllen. Notwendige Funktionen zur Kommunikation sind über Bibliotheken verfügbar und können durch einfache Auswahl eingefügt werden. Die spätere Umsetzung in Code erfolgt mit Hilfe von Codegeneratoren automatisch. Der Entwickler muss lediglich für eine neue Funktion bestimmen, welche Daten von anderen Anwendungen benötigt werden. Gleiches gilt für die erzeugten Outputs. Diese werden der Middleware lediglich angemeldet und hinzugefügt, welche Anwendungen diese Daten benötigen. Dadurch entfallen bei fast allen Funktionen die EIFs. Auch bei den EIs und EOs ergeben sich dadurch erheblich Einsparungen. Allerdings fallen diese meist nicht vollständig weg, denn die Middleware stellt lediglich Funktionen bereit, die EIs und EOs leichter in das Kommunikationsnetz integrieren lassen, die Implementierung der eigentlichen Funktion muss dennoch erfolgen.

		EI			EO			EQ			AL			DF			EIF				
Featurebezeichnung	FP ohne WV	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H		
													Total FP(resuse) unadjusted					=	383	FP/ Fkt.	Veränd.
Architektur	n/a			1			1						1				8	2		72	n/a

Tabelle 39 – FP Bewertung SWPL Architektur im Anwendungsbeispiel

Die Middleware erzeugt im Anwendungsbeispiel einen Aufwand von 72 FPs (Tabelle 39). Auf der anderen Seite ergeben sich Einsparungen von 104 FPs bei den Funktionen. Im Ergebnis führt das zu einem um 32 FPs geringeren Aufwand ggü. der Entwicklung ohne Berücksichtigung einer eigenen SWPL Architektur. Der Gesamtaufwand beträgt nun 383 FPs und bedeutet gegenüber der Einzelentwicklung eine Einsparung von 54% (Tabelle 40). Erneut soll darauf hingewiesen werden, dass dieses Ergebnis lediglich illustrativen Charakter hat. Außerdem sind bisher keine Kosten für die aktuelle Wiederverwendung von Assets berücksichtigt und es gilt weiterhin die Annahme, dass alle Komponenten Teil der Asset Base sind. Diese beiden Bedingungen werden im nächsten Kapitel aufgehoben.

Nr.	Featurebezeichnung	FP ohne WV	EI			EO			EQ			AL			DF			EIF			FP/ Fkt.	Veränd.
			L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H		
			Total FP(resuse) unadjusted																=	383		
1	Zugang	25	1			1							1		1	1					22	-12%
2	Personalisierung	19	1			1						1			1						14	-26%
3	Alarmanlage	18	1			2						1									13	-28%
4	Wegfahrsperre	26	1				1						1			1					18	-31%
5	Homecoming Light	19	1			1						1				1					16	-16%
6	Höchstgeschwindigkeit	29	3			2						1			1						24	-17%
7	Fensteröffnung	14	1			1						1									9	-36%
8	Starten von Anwendungen	17	1			1				1		1	1		1						21	24%
9	GSM	87																			102	17%
10	>Kommunikation	25											1	1	1	1				1	30	20%
11	>Standortbestimmung	14	1			1						1			2						19	36%
12	>Stilllegung	24	1			2							1		2						24	0%
13	>Aktionsradius	24	2			2						1	1		2						29	21%
14	WLAN	57																			72	26%
15	>Kommunikation	15											1		1	1				1	25	67%
16	>Update von Daten	20		1		1							1		2						21	5%
17	>Konfiguration	22		1		1							1		1		1				26	18%
18	Architektur	n/a			1			1						1				8	2		72	n/a

Tabelle 40 – FP Zählung Zugangssysteme als SWPL mit Architektur (vollständig ausgefülltes Worksheet für Anwendungsbeispiel, siehe auch Tabelle 29)

5.4.4. Optimierung der Asset Base

Die Ergebnisse aus dem vorherigen Kapitel zeigen den Aufwand für die Realisierung der geplanten Produkte auf Basis einer SWPL, entweder mit oder ohne Berücksichtigung einer SWPL Architektur. In beiden Fällen wurde die Annahme getroffen, dass alle Komponenten und Funktionen wiederverwendbar entwickelt werden müssen und damit Bestandteil der Asset Base werden. Die einzige Ausnahme sind Komponenten, die bereits zu Beginn der Analyse als klar ungeeignet identifiziert und demnach in Schritt 3 nicht als wiederverwendbar betrachtet worden sind. Es ist unwahrscheinlich, dass diese Maximierung der Asset Base optimal ist. Teilweise wird dies sogar unmöglich sein oder zu einem extrem hohen Aufwand führen. Im ungünstigen Fall könnte dadurch sogar eine eigentlich vorteilhafte SWPL einen

höheren Aufwand erzeugen als die Einzelentwicklungen. Um dies zu verhindern, wird ein zusätzlicher Analyseschritt eingefügt, die Optimierung der Asset Base.

Aufgrund der Struktur der bisherigen Analyse stehen die FP Werte auf Funktionsebene zur Verfügung. Ebenso ist bekannt, wie oft eine bestimmte Komponente und damit auch die zugehörige Funktion in verschiedenen Produkten wiederverwendet werden soll, wobei dieser Wert hier nochmal individuell angepasst werden kann. Mit diesen Daten ist es möglich, eine aufwandsminimierte Asset Base auf Komponentenebene zu ermitteln. Dazu wird ein Vergleich der FP Werte je Komponente ohne und mit Wiederverwendung erstellt. Das entsprechende Arbeitsblatt ist inklusive der Daten des Anwendungsbeispiels in Tabelle 41 und 42 dargestellt.

Zusätzlich zu den bereits vorhandenen Daten verlangt das Arbeitsblatt die Angabe des Aufwands in FP, wenn die Komponente tatsächlich wiederverwendet wird. Der Aufwand ergibt sich aus der notwendigen Anpassung der Komponente für das spezifische Produkt, z.B. mittels Parametrisierung. Wie bereits oben erklärt, kann dieser Aufwand pauschal geschätzt werden oder über eine Detailanalyse der FPs ermittelt werden. Im Anwendungsbeispiel wurde ein pauschaler Wert von ca. 10 % der Asset-Entwicklungskosten gewählt, der je Komponente individuell leicht angepasst worden ist.[84] Weiterhin ist ein Wert für den jährlichen Aktualisierungsaufwand anzugeben. Assets veraltern über die Zeit, daher ist es sinnvoll, die Assets in regelmäßigen Abständen auf notwendige Anpassungen zu untersuchen. Die Anpassungen betreffen bestimmte Funktionstypen, z.B. ist ein Algorithmus anzupassen oder komplett neu zu erstellen. Es können auch Fehler behoben werden, die während der Anwendung identifiziert worden sind. Daher kann eine Bewertung mit der FP Methodik erfolgen. Allerdings ist zu beachten, dass die Veränderungen meist nur einen kleinen Teil eines Funktionstyps betreffen. Entsprechend darf in einem solchen Fall nicht die volle FP Anzahl als Wartungsaufwand gerechnet werden, sondern lediglich ein entsprechender Anteil. Eine solche Berechnung ist aufwendig, daher ist es an dieser Stelle möglich, den Wartungsaufwand grob zu schätzen. Ein wichtiger Einflussfaktor ist die Stabilität der Technik eines Produktes. Bei dem Beispiel der Zugangssysteme wird angenommen, dass über die Laufzeit der SWPL keine entsprechende Anpassung notwendig sein wird. In der Literatur finden sich verschiedene Bandbreiten. Das COCOMO Modell nennt Werte zwischen 5 und

[84] In der Literatur ist ein Wert von 15% der ursprünglichen Entwicklungskosten der wieder verwendbaren SW für opportunistische Wiederverwendung allgemein akzeptiert [Clem05, S. 14]. Das COCOMO und das COPLIMO Modell verwenden einen Wert von ungefähr 5% für strategische Wiederverwendung [vgl. Boeh00, S20 und Boeh04].

20% [Boeh81]. Bei [Cohe03] werden die jährlichen Maintenance Cost auf 10 bis 15% der Entwicklungskosten beziffert.

Abschließend wird das Risiko abgefragt, dass die Komponente nicht wiederverwendet werden kann. Dabei handelt es sich um einen Prozentwert, der die Wahrscheinlichkeit angibt, die Komponente nach einem Jahr komplett ersetzen zu müssen. Gründe dafür können technische Veränderungen oder Veränderungen in der Produktplanung sein, z.B. bedingt durch geänderte Unternehmensziele. Je stärker demnach die technischen Veränderungen einer Komponente sind und je unsicherer das weitere Angebot dieser Komponente innerhalb des Unternehmens ist, desto höher wird dieser Prozentwert ausfallen. Berücksichtigt wird dieser Wert wie ein negativer Zins. Dazu werden alle Beträge, die erst in Zukunft anfallen, über die Laufzeit entsprechend abgezinst.

Die Berechnung der Ersparnis erfolgt nach der Formel:

Gleichung 15

$$B_{AB} = -A_{AB} + \left[n * \left(A_E - A_r\right) - A_a * t\right] * \left(1 - r\right)^t$$

Wobei:

	B_{AB}	Ersparnis durch Realisierung als Asset im Vergleich zur Einzelentwicklung in FP
	A_{AB}	Aufwand zur Erstellung als wiederverwendbares Asset in FP
	A_E	Aufwand zur Erstellung als Einzelprodukt in FP
	A_r	Aufwand bei tatsächlicher Wiederverwendung in FP
	A_a	Aufwand für jährliche Aktualisierung
	n	Anzahl der Wiederverwendungen
	t	Laufzeit der SWPL in Jahren
	r	Risiko für Neuentwicklung in % pro Jahr

Ein positiver Wert für B_{AB} bedeutet, dass diese Komponente als Asset realisiert werden sollte. Bei einem negativen Wert oder einem Wert nahe null sollte überprüft werden, ob die Anzahl der Wiederverwendungen erhöht werden kann. Ist dies nicht möglich, sollte diese Komponente auf Produktbasis entwickelt werden. Diese Entscheidung gilt für die weitere

Analyse und den Business Case. Beim tatsächlichen Design der SWPL kann dies als Information genutzt werden, die endgültige Entscheidung sollte jedoch auf Basis einer umfangreicheren Scoping Analyse getroffen werden.

Nr.	Komponente	Aufwand ohne WV in FPs	Aufwand mit WV in FPs	Faktor in %	Anzahl WV #	Aufwand bei Reuse des Assets in FPs	Jährlicher Aufwand zur Aktualisierung in FPs	Wahrscheinlichkeit, dass Komponente komplett ersetzt wird in %	Ersparnis als Asset in FPs
1	Zugang	25	31	124%	5	3	0	0%	79
2	Personalisierung	19	23	121%	3	2	0	0%	28
3	Alarmanlage	18	21	117%	3	2	0	0%	27
4	Wegfahrsperre	26	29	112%	5	4	0	0%	81
5	Homecoming Light	19	23	121%	3	2	0	0%	28
6	Höchstgeschwindigkeit	29	32	110%	2	3	0	0%	20
7	Fensteröffnung	14	17	121%	5	1	0	0%	48
8	Starten von Anwendungen	17	32	188%	3	4	0	0%	7
9	GSM	87	117	134%	2	8	0	0%	41
10	WLAN	57	90	158%	1	6	0	0%	-39

Tabelle 41 – Optimierung der Asset Base ohne Architektur Investment (gelber Felder erfordern einen individuellen Input)

Für das Anwendungsbeispiel der Zugangssysteme ergibt sich, dass lediglich eine Komponente, nämlich WLAN nicht als Teil der Domain entwickelt werden sollte. Bei der Komponente Starten von Anwendungen ergibt sich zwar ein positiver Wert, allerdings ist die erwartete Einsparung relativ gering, sodass eine individuelle Entscheidung getroffen werden sollte. Die Ergebnisse für diese beiden Komponenten resultieren aus dem hohen Mehraufwand, um die Komponente wiederverwendbar zu gestalten. Bei der Komponente WLAN kommt hinzu, dass nur eine Verwendung geplant ist. Dadurch schließt sich ein wiederverwendbares Design aus. Bereits bei einer zweiten Verwendung wäre eine Realisierung als Asset vorteilhaft. Die Daten in Tabelle 41 geben die Ergebnisse für die Analyse ohne Berücksichtigung einer SWPL Architektur wider. Auch diese Ergebnisse sind sinnvoll anwendbar. Allerdings muss sich der Verwender der Gefahr bewusst sein, dass tendenziell auch einige sinnvoll wiederverwendbare Komponenten abgelehnt werden. Ohne Berücksichtigung einer SWPL Architektur wird ein größerer Aufwand als nötig prognostiziert, da es keine übergreifenden Schnittstellen gibt und die Realisierung jeweils mehrfach auf Komponentenebene erfolgen muss. Dadurch wird der Vorteil durch die Nutzung einer Komponente als Asset tendenziell unterschätzt. Dies führt zu einer durchschnittlich kleineren Asset Base, wodurch sich insgesamt ein Nachteil in der Bewertung der SWPL ergibt. Dies entspricht einem konservativen Vorgehen und kann akzeptiert werden. Wichtig ist allerdings, dass sich der Anwender dessen bewusst ist.

Tabelle 42 zeigt die Ergebnisse unter Berücksichtigung einer SWPL Architektur. Es fällt auf, dass einige Komponenten bereits einen höheren A_E als A_{AB} Wert aufweisen. Dies bedeutet, dass sich eine Wiederverwendung bereits bei der ersten Verwendung lohnt. Der geringere Aufwand trotz Wiederverwendbarkeit täuscht, da dies auf dem Einsatz einer SWPL Architektur beruht, die zusätzlichen Aufwand erzeugt hat.

Nr.	Komponente	Aufwand ohne WV	Aufwand mit WV	Faktor	Anzahl WV	Aufwand bei Reuse des Assets	Jährlicher Aufwand zur Aktualisierung	Wahrscheinlichkeit, dass Komponente komplett ersetzt wird	Ersparnis als Asset
		in FPs	in FPs	in %	#	in FPs	in FPs	in %	in FPs
1	Zugang	25	22	88%	5	3	0	0%	88
2	Personalisierung	19	14	74%	3	2	0	0%	37
3	Alarmanlage	18	13	72%	3	2	0	0%	35
4	Wegfahrsperre	26	18	69%	5	4	0	0%	92
5	Homecoming Light	19	16	84%	3	2	0	0%	35
6	Höchstgeschwindigkeit	29	24	83%	2	3	0	0%	28
7	Fensteröffnung	14	9	64%	5	1	0	0%	56
8	Starten von Anwendungen	17	21	124%	3	4	0	0%	18
9	GSM	87	102	117%	2	8	0	0%	56
10	WLAN	57	72	126%	1	6	0	0%	-21

Tabelle 42 – Optimierung der Asset Base unter Berücksichtigung der SWPL Architektur

Berücksichtigt man nun die Ergebnisse der Asset Base Optimierung, sollten alle Komponenten bis auf WLAN in die Asset Base integriert werden. Allgemein können die gewonnen Daten nun genutzt werden, um zunächst den Gesamtaufwand zur Erstellung aller geplanten Produkte mittels SWPL zu ermitteln. Dazu werden die Aufwände der Komponenten, die in die Asset Base einbezogen werden sollen, mit ihrem Wert für die Wiederverwendung addiert. Hinzugerechnet wird der Aufwand für die Entwicklung der Komponenten, die nicht Bestandteil der Asset Base sind. Dieser Wert kann dann in einem finalen Schritt mit dem Gesamtaufwand für die Einzelproduktentwicklung verglichen werden. In dem Anwendungsbeispiel der Zugangssysteme betrug der Aufwand für alle 5 Produkte als Einzelentwicklung 833 FPs. Eine Entwicklung auf Basis einer SWPL ohne Berücksichtigung einer individuellen SWPL Architektur ergibt einen Aufwand von 445 FPs inklusive des Aufwands zur Wiederverwendung und der Optimierung der Asset Base. Der Wert mit Berücksichtigung der Architektur beträgt 388 FPs. Diese Werte liegen höher als die zuvor ermittelten Gesamtentwicklungsaufwände mit allen Komponenten in der Asset Base. Der Grund ist, dass die neu berücksichtigten Wiederverwendungskosten höher liegen als der durch die Optimierung erzielte Vorteil. Denn lediglich die WLAN Komponente wird individuell realisiert. Damit ist der Einspareffekt der Optimierung im Anwendungsbeispiel limitiert. Der Break Even Punkt für die Einführung einer SWPL liegt in beiden Fällen (mit und ohne

Optimierung) zwischen dem dritten und vierten Produkt. Es ergibt sich demnach für das Anwendungsbeispiel durch die Optimierung keine Veränderung der Empfehlung.

5.4.5. Abschließende Betrachtung Schritt 3

In Schritt 3 des e-FACTS Modells wurde die FP Methodik eingeführt, um eine valide Berechnungsbasis für den Aufwand zur Erstellung einer SWPL zu erhalten. Es wurde gezeigt, dass Varibilitäten und der daraus resultierende Mehraufwand bei der Erstellung mittels FPs erfasst werden können. Anschließend wurde die zweifache Berechnung der FPs anhand eines Anwendungsbeispiels dargestellt. Für die Berechnung des Aufwands bei Wiederverwendung stehen zwei Alternativen zur Verfügung. Ein Unternehmen ohne SWPL Erfahrung, das schnell eine Empfehlung für seine individuelle Situation sucht, kann die Bewertung ohne eigene SWPL Architektur durchführen. Diese liefert ein konservatives Ergebnis und kann daher für eine erste Einschätzung genutzt werden. Die Methode unter Berücksichtigung einer speziellen SWPL Architektur liefert ein genaueres Ergebnis, ist jedoch aufwändiger durchzuführen.

Schließlich wurde eine Möglichkeit vorgestellt, das Scoping einer SWPL zu unterstützen. Für die weitere Analyse wurde damit eine Auswahl der Komponenten getroffen, die zu einem optimalen Bewertungsergebnis für die SWPL führt, indem nur auf die ökonomisch sinnvollen Komponenten für die Wiederverwendung in einer Domain abgestellt wurde. Am Ende von Schritt 3 stehen alle notwendigen Daten zur Verfügung, um einen Business Case zu erstellen. Dies erfolgt in Schritt 4.

5.5. Schritt 4: Business Case

"Ein Business Case ist ein Szenario zur betriebswirtschaftlichen Beurteilung einer Investition" [Brug05, S. 11]. Ebenso ist es "a tool for making a business decision because it predicts the organizational consequences of that decision" [Clem02, S. 220]. Sehr anschaulich definiert [Kitt09, S. 62]: "Business Case = Comparison of the costs associated with a product or project to the quantified economic benefits or value to be derived from it."

Bei der Ausgestaltung eines Business Cases bestehen weitgehende Freiheiten, denn es gibt keinen einheitlichen Standard. Ein Business Case kann genutzt werden, um die finanziellen Folgen einer Entscheidung in übersichtlicher Form darzustellen und damit eine Entscheidungsfindung zu unterstützen. Eine wichtige Eigenschaft ist die korrekte zeitliche

Erfassung aller finanziellen Ströme, die im Zusammenhang mit der Investition stehen. Häufig wird er genutzt, um eine Investition besser abschätzen zu können. Zusätzlich kann ein Business Case zahlreiche Finanzkennzahlen enthalten und Finanzmodelle oder Cash-Flow Berechnungen berücksichtigen. In der vorliegenden Arbeit wird ein Business Case verwendet, um die bisherigen Ergebnisse aus der reinen Welt der FPs in die betriebswirtschaftliche Welt der Kosten und Erträge zu übertragen und einige Erweiterungen einzuführen. Diese sind:

- Die finanzmathematisch korrekte Betrachtung von zeitlichen Zahlungsströmen
- SWPL spezifische Kosten, wie Wartung der Domain und die Integration von COTS-SW
- Automobilspezifische Kosten, wie eventuelle HW Kosten durch komplexere SW-Anforderungen und Garantiekosten
- Optionen, um das Investitionsrisiko besser berücksichtigen zu können.

Zunächst sind die ermittelten FPs in einen monetären Betrag umzuwandeln. Dies kann mit Hilfe der erhobenen Basisdaten zu Beginn der Analyse erfolgen. FPs stellen ein Maß für den Aufwand zur Erstellung von SW Funktionalität dar. Sie decken den gesamten Entwicklungsprozess vom Design bis zum Testing ab. Aus dem FP Wert kann eine Zeitdauer berechnet werden, sofern eine Produktivität bekannt ist. Um einen bestimmten Aufwand zu leisten, benötigen unterschiedliche Personen eine unterschiedliche Zeit, da sie eine individuelle Produktivität aufweisen. Je höher die Produktivität ist, gemessen als Einheiten, die in einem bestimmten Zeitintervall absolviert werden können, desto weniger Zeit wird zur Erledigung eines bestimmten Aufwands benötigt. Auch Unternehmen weisen deutlich abweichende Produktivitäten auf. Im Rahmen der Basisdatenerhebung werden die Unternehmen zu Beginn der Analyse befragt, ob sie bereits eine Produktivitätsanalyse durchgeführt haben. Liegt ein solcher Wert vor, kann er eventuell genutzt werden. Eine Einschränkung besteht, da der Aufwand lediglich in FPs bekannt ist. Ein verlässlicher Wert für die Produktivität müsste ebenfalls in FPs gemessen worden sein. Da es sich hierbei um ein proprietäres Verfahren handelt, das erst 2006 von McKinsey&Company veröffentlicht worden ist, ist dies nicht zu erwarten. Aufgrund der großen Ähnlichkeit zum Function Point Ansatz erscheint es möglich, eine darauf aufbauende Produktivität ebenfalls zu verwenden. In der Literatur finden sich zahlreiche Umrechnungstabellen, die eine Umwandlung nahezu aller

Produktivitätskennzahlen untereinander ermöglichen.[85] Allerdings weisen die Autoren darauf hin, dass es hierbei zu erheblichen Abweichungen kommen kann (vgl. auch Kapitel 5.3.2). Liegt keine geeignete Produktivitätskennzahl vor, dann ist es zur Erstellung des Business Cases notwendig, eine solche zu erheben. Dazu kann ein bereits abgeschlossenes Projekt herangezogen werden. Idealerweise sollte dieses aus der jüngeren Vergangenheit stammen und unter ähnlichen Bedingungen durchgeführt worden sein, wie sie für die Einführung der SWPL erwartet werden. Grund dafür ist, dass etwaige Nebenbedingungen, wie z.B. enge Zeitvorgaben oder eine falsche Teamorganisation, die Kennzahl beeinflussen können.[86]

Die Produktivität erlaubt es, aus der Anzahl der FPs ein Zeitintervall zu bestimmen. Dieses sagt aus, wie lange ein einzelner Entwickler für die Fertigstellung des entsprechenden Projektumfangs benötigen würde. In einem weiteren Schritt kann dieses Zeitintervall mit den durchschnittlichen Vollkosten eines Entwicklers multipliziert werden und es ergibt sich der monetär bewertete Aufwand. Tabelle 43 zeigt die entsprechenden Daten des Anwendungsbeispiels unter Berücksichtigung der SWPL Architektur. Die darin enthaltenen Erweiterungen werden im Folgenden erklärt.

Zeit (Jahr)	2010	2011	2012	2013	2014
Ereignisbeschreibung	Entwicklung Asset Base	Start Prod.1	Start Prod.2	Start Prod. 3+4	Start Prod.5
Kosten					
Domainentwicklung	398				
Produktentwicklung Wiederverwendung		13	28	63	48
Produktentwicklung neue SW		0	0	0	95
Domainwartung		0	0	0	0
Risiko Asset Neuentwicklung		0	0	0	0
COTS Verwendung		0	0	0	0
Höhere HW Kosten					
Benefits					
Aufwand bei kompletter Neuentwicklung		108	250	512	518
Berechnungen					
Summe Kosten	398	13	28	63	143
Summe Benefits	0	108	250	512	518
Gesamtbetrag	-398	95	222	448	375
Abzinsungsfaktor	8%	8%	8%	8%	8%
Wert im jeweiligen Jahr	-398	88	190	356	276
Barwert der Investition in 2010					511

Tabelle 43 – Business Case Anwendungsbeispiel Zugangssysteme (Werte in Tausend EUR, außer Prozent)

Der Business Case ist nach Geschäftsjahren aufgeteilt. Das dargestellte Zeitintervall orientiert sich an der Lebensdauer der SWPL. Für jedes Jahr sind die relevanten Ereignisse für die

[85] Siehe z.B. [Jone07, S. 107] oder [Boeh00, S. 20]

[86] Ebenso kann die Einführung von neuen Werkzeugen die Produktivität beeinflussen. Daher sollte regelmäßig eine Kontrolle des Produktivitätswerts erfolgen.

SWPL angegeben. Anschließend folgt die Erfassung der Kosten, jeweils in dem Jahr der Entstehung. Da eine SWPL als Investition verstanden werden kann, fallen zunächst Kosten für die Erstellung der Asset Base an. Anschließend werden die Kosten für die tatsächliche Wiederverwendung von Assets angegeben, gefolgt von Kosten für die Entwicklung von Komponenten, die nicht Teil der Asset Base sind, aber dennoch für ein spezielles Produkt benötigt werden. Die WLAN Komponente wird lediglich für Produkt 5 benötigt und entsprechend individuell entwickelt. Daher werden die Kosten für die Entwicklung der 57 FPs (vgl. Tabelle 42) erst im Jahr der Produkteinführung[87] aufgeführt. Die 57 FPs wurden über den Produktivitätskostenfaktor von 1,66 in Kosten umgerechnet (95 Tausend EUR). Der Faktor ergibt sich aufgrund der durchschnittlichen Kosten eines Entwicklers (100.000 EUR/Jahr) und der angenommenen Produktivität der Entwickler (5 FP/Monat, bzw. 60 FP/Jahr). Darunter werden die Kosten für die Wartung der Domain Assets gezeigt, die bei der Optimierung der Asset Base in Kapitel 5.4.4 ermittelt wurden. Dort wurden zusätzlich Wahrscheinlichkeiten bestimmt, dass eine Komponente bereits nach einem Jahr komplett ersetzt werden muss. Zur Umrechnung in einen Aufwandswert ist die Wahrscheinlichkeit mit dem FP Wert zur Erstellung des Assets zu multiplizieren. Dadurch wird das erwartete Risiko eines zusätzlichen Entwicklungsaufwands korrekt berücksichtigt.

In der nächsten Zeile findet sich ein neuer Wert, COTS. "COTS (commercial off-the-shelf) software plays a pivital role in almost all systems developed today. COTS products exist independent of a specific system and are developed by a commercial organization for commercial purposes" [Clem02, S. 90]. Aufgrund der zunehmenden Komplexität von SW-Systemen sind Unternehmen heute meist gar nicht in der Lage, sämtliche Komponenten selbst zu entwickeln. Ebenso ist es bei Standardfunktionen ökonomisch oft sinnvoller auf solche Lösungen zurückzugreifen, als proprietäre SW zu entwickeln. Betriebssysteme waren lange Zeit die bedeutendsten Beispiele für COTS-SW. Mit der zunehmenden Verbreitung von Middleware Technologien hat sich dieser Markt stark erweitert [Clem02, S. 90 f.]. Heute sind für fast alle Anwendungen standardisierte Lösungen am Markt vorhanden. Wer bspw. einen Webshop aufbauen möchte, benötigt kein IT Wissen, da alle notwendigen Bausteine am Markt verfügbar sind.

COTS-SW wird entweder während der Produktentwicklung für ein einzelnes Produkt berücksichtigt. Es kann aber auch Bestandteil der Domain Assets werden [Pohl04, S. 95]. Im

[87] Zur Vereinfachung wird angenommen, dass die Produktentwicklung und die –Einführung im gleichen Jahr stattfinden.

letzten Fall sollte ein umfangreicher Auswahlprozess stattfinden, da das Unternehmen mit COTS Komponenten die Verantwortung für Updates und Variabilität aus der Hand gibt und dadurch eine Abhängigkeit gegenüber dem Anbieter der COTS-SW entsteht. Eine Beschreibung des Auswahlprozesses findet sich bei [Clem02, S. 91ff.]. Zur Unterstützung der Auswahl von COTS-Komponenten als Bestandteil der Domain aus einer technischen Sicht existiert CoVAR.[88]

COTS Komponenten wurden bisher in der Analyse nicht berücksichtigt: Einerseits ist eine Erfassung der FPs für COTS-SW deutlich aufwändiger, da die Funktionen nicht so transparent vorliegen wie bei einer internen Entwicklung. Andererseits würde eine solche Erfassung einen unnötigen Zusatzaufwand produzieren, da die tatsächlichen Kosten für COTS-Komponenten bereits vorliegen. Ist die Entscheidung für eine COTS-SW gegenüber einer Eigenentwicklung eindeutig, bedarf es keiner weiteren Analyse. Allein der Aufwand zur Integration der SW in die Asset Base sollte separat erfasst werden. Anders verhält es sich, wenn eine Entscheidung zu treffen ist, ob eine Komponente extern zugekauft oder intern entwickelt wird. Hierbei bietet der FP Ansatz eine hilfreiche Unterstützung, indem die Kosten miteinander verglichen werden können. Auch besteht die Möglichkeit, den Preis einer COTS-SW zu bewerten, indem die FPs und damit der Umfang gemessen werden. Im Business Case sind schließlich die tatsächlich angefallenen Kosten für die Verwendung von COTS-Komponenten aufzunehmen.

Die letzte Kostenkategorie sind höhere HW Kosten. Diese können von Bedeutung sein, wenn bedingt durch die komplexeren Funktionen einer SWPL teurere HW verwendet werden muss.[89] Dies kann ein entscheidender Faktor sein, da durch die hohen Stückzahlen in der Automobilindustrie Entwicklungskosten oft überkompensiert werden. Pro Fahrzeug betragen die Entwicklungskosten für einzelne Systeme lediglich wenige Euros oder sogar Cents. Sollte dann zusätzliche HW notwendig sein, relativieren sich mögliche Einsparungen im Entwicklungsbereich sehr schnell. Macht die Einführung einer SWPL zusätzliche HW erforderlich, sollten die finanziellen Folgen im Detail untersucht und im Business Case

---

[88] CoVAR steht für Component Selection considering Variability, Architectural Concerns, and Requirements [siehe Pohl01 und Ulfa05].

[89] So gilt bspw. für Funktionen, die entsprechend der AUTOSAR Spezifikation realisiert werden: "Durch die Einführung neuer Schichten und das Splitten einzelner Schichten wird der Bedarf an Rechenleistung größer. Die große Anzahl von Basis-Software-Komponenten führt auch zu Mehrfachdatenhaltung und somit zu erhöhtem RAM-Verbrauch, sowie zu einem größeren ROM Verbrauch. […] Somit sind die heute teilweise noch weit verbreiteten 8-Bit Microcontroller aufgrund der geringen Rechenleistung und des stark eingeschränkten Speicherausbaus für AUTOSAR Systeme nicht mehr geeignet." [Hörn05, S. 417]

berücksichtigt werden. Zusätzliche HW durch eine SWPL kann ebenso bedeuten, dass bei Versionen mit geringem Funktionsumfang nicht wie bisher ECUs weggelassen werden können, sondern diese aufgrund von Basisanforderungen der SWPL grundsätzlich notwendig sind. Durch eine konsequente Modularisierung und Variabilisierung kann dieser Effekt jedoch reduziert werden.

Für das Anwendungsbeispiel wurde angenommen, dass zusätzliche HW erforderlich sein könnte. Eine genaue Bewertung liegt nicht vor. Aber mit Hilfe der Ergebnisse des Business Cases kann eine Aussage über die maximal möglichen Zusatzkosten je Fahrzeug getroffen werden. Dazu sind die erwarteten Ersparnisse durch die Anzahl der geplanten Fahrzeuge, für die eine zusätzliche HW benötigt wird, zu dividieren. Angenommen, dies trifft auf Fahrzeug 1 zu, dann dürften die Mehrkosten je Fahrzeug maximal 0,34 Euro betragen. Liegen die HW-Kosten höher, wäre die SWPL Einführung abzulehnen.

Damit eine Investition vorteilhaft ist, werden neben den notwendigen Investitionen positive Effekte erwartet, auch Benefits genannt. Hauptsächlich handelt es sich um eingesparte Kosten, da die Komponenten nicht mehrfach als Einzelprodukte entwickelt werden müssen. Daher werden im Business Case die Kosten einer kompletten Neuentwicklung als Benefit der SWPL aufgenommen. Zusätzlich können zahlreiche weitere Effekte berücksichtigt werden, wie bspw. reduzierte Garantiekosten durch höhere Qualität, höhere Kundenzufriedenheit und –Bindung, schnellere Time-to-Market und ein höherer Marktanteil. Diese Erweiterungen werden in Schritt 5 behandelt. Ein weiterer positiver Effekt wird im Rahmen des Business Cases nicht explizit berücksichtigt, soll aber dennoch zumindest kurz angedeutet werden: die Einsparungen bei der Wartung der embedded Systems im Feld. Wartung und die damit verbundenen Kosten wurden bereits mehrfach, allerdings ohne Differenzierung, angesprochen. Bisher wurde damit immer auf die Wartung der Asset Base der SWPL abgestellt. Diese Perspektive soll nun um den Aspekt der Wartung der bereits im Markt befindlichen Produkte, also sowohl die Instandhaltung als auch die Wiederinbetriebnahme nach einem Ausfall, erweitert werden. Während sich erstgenannter Punkt in zahlreichen Veröffentlichungen und auch ihm Rahmen dieser Arbeit als Thema wiederfindet (z.B. in der Abbildung 8 in Teil A Kapitel 4.4 oder im Fragebogen Teil B Kapitel 5.1.1.1.4 und hier beim Business Case selbst), liegen für das Thema SWPL und Wartung im Feld aktuell keine fundierten Erkenntnisse für den Bereich Automotive vor. Daher wird auf eine quantitative

Betrachtung verzichtet. Allerdings könnten sich durch die SWPL hier zusätzliche Einsparpotenziale ergeben, die im Folgenden qualitativ aufgeführt werden:

- Aufgrund der hohen Bedeutung der elektronischen Systeme im Automobil spielen embedded Systems und die dort installierte SW bereits heute bei der Instandhaltung und der Beseitigung von Problemen eine große Rolle. So werden bei einer Inspektion in vielen Fällen zahlreiche neue Funktionen durch das Flashen mit der neuesten SW Version hinzugefügt. Da dieses Flashen aufgrund des Datenumfangs bei aktuellen Fahrzeugen einen erheblichen Zeitaufwand erzeugt, wurden Maßnahmen ergriffen, um diesen Prozess zu beschleunigen (siehe z.B. [Schä06, S. 309ff.]). Zusätzlich ergibt sich aufgrund der zunehmenden Verknüpfung der einzelnen Systeme eine deutlich gesteigerte Komplexität hinsichtlich der Kompatibilität. Bei der Vielzahl an unterschiedlichen Systemen und Kombinationsmöglichkeiten in einer Fahrzeugvariante ist es kaum möglich, alle Variationen vollständig zu testen. Diesem Problem wird u.a. mit einer stärkeren Standardisierung der Schnittstellen begegnet. SWPL führen diese Idee konsequent fort. So ermöglicht bspw. die strategische Wiederverwendung von Kernkomponenten eine grundsätzliche Reduktion der Anzahl an unabhängigen Komponenten. Die Zahl der Varianten wird sich im Rahmen von SWPL zwar durchschnittlich erhöhen, aber die einzelnen Produkte basieren dabei auf standardisierten Assets mit standardisierten Schnittstellen. Der notwendige Testaufwand kann dadurch stärker nach vorne, in die Entwicklungsphase, verlegt und gleichzeitig reduziert werden. Die Varianten einer SWPL sind daher im Allgemeinen zu jeder Zeit kompatibel und können mit der aktualisierten Version verwendet werden. Für den Service im Feld bedeutet dies eine Vereinfachung beim Flashen, da nur eine einzige Updateversion notwendig ist. Gleichzeitig reduziert sich das Risiko von unbekannten Inkompatibilitäten und damit der Wartungsaufwand insgesamt. Auch die Fehleranalyse wird vereinfacht. Heute werden Fehler, die von den Systemen nicht erkannt werden oder die nicht dokumentiert sind, direkt an die Hersteller weitergeleitet, da für die Werkstätten keine Möglichkeit zur Fehlerbehebung besteht. Die Anzahl solcher für die Werkstatt und den Kunden zeitaufwändigen Vorfälle dürfte sich durch die Einführung von SWPL verringern.

- Neben den zu einem Zeitpunkt parallel angebotenen Produktvarianten bedeutet die lange Laufzeit der Produkte (siehe auch Teil A, Kapitel 4.4) im automobilen Umfeld

zusätzlich eine hohe Zahl an Varianten, die im Rahmen der Wartung zu betreuen sind. Dadurch ergibt sich für die Werkstätten die Herausforderung für eine große Zahl an Systemen die passenden Austauschkomponenten und SW-Versionen zur Verfügung zu haben. Durch eine SWPL kann die Anzahl der verschiedenen Varianten deutlich reduziert werden, da eine SWPL im Allgemeinen eine Laufzeit von mehreren Jahren aufweisen wird. Dies verringert nicht nur die Menge der notwendigen Ersatzteile - z.B. da identische HW-Komponenten durch SW-basierte Parametrisierung direkt in der Werkstatt individualisiert werden können - sondern es reduziert auch die Komplexität der Fehlerdiagnose, da es immer nur einen aktuellen SW-Stand für eine SWPL gibt. Zusätzlich wird damit sichergestellt, dass neue Funktionen in alle Fahrzeuge integriert werden können. Ohne SWPL ist es wahrscheinlicher, dass unterschiedliche HW- und SW-Versionen einen zusätzlichen Entwicklungs- und Anpassungsaufwand erzeugen würden und daher evtl. einige neue Funktionen nicht in allen Kombinationen verfügbar sind. Die Kompatibilität wäre zusätzlich zu prüfen.

- SWPL bieten eine Möglichkeit, den Umfang einer klassichen Fahrzeugwartung um neue After-Sales Möglichkeiten zu erweitern. Aufgrund der Modularität der SWPL ist es einfacher möglich, neue SW-basierte Funktionen hinzuzufügen. Diese Möglichkeit wird durch die bei SWPL stärker standardisierte HW unterstützt. Zusätzlich werden Kompatibilitätsprobleme durch SWPL vermieden. Dadurch ergeben sich neue, interessante Geschäftsmodelle für OEMs und Werkstätten, wie z.B. die einfache und damit margenträchtige Nachrüstung von umfangreichen Funktionspaketen (bspw. verbesserte Stauwarnung im Navigationsgerät nach Freischaltung und Aufspielen neuer SW oder ein verändertes Fahrverhalten durch aktualisierte Parameter der Motor-, Getriebe- und Fahrwerkselektronik). Technisch sind solche Updates seit einiger Zeit möglich. Allerdings erwiesen sich diese bisher aufgrund des umfangreichen Testaufwands für alle Kombinationsmöglichkeiten und der notwendigen Anpassungen für unterschiedliche HW-Konfigurationen als unwirtschaftlich.

Aufgrund der fehlenden empirischen Datengrundlage wird die Wartung im Feld im Business Case nicht berücksichtigt. Die anderen genannten Werte können aber mittels einfacher Addition und Multiplikation zu einem finalen Wert aggregiert werden. Dabei wird ein Abzinsungsfaktor verwendet, durch den die zu unterschiedlichen Zeitpunkten auftretenden Geldströme finanzmathematisch korrekt zu einem Zeitpunkt, der Gegenwart,

zusammengefasst werden. Der Barwert der Investition[90] bezeichnet den finanziellen Vorteil aus der Investition in die SWPL. Ein positiver Wert ist damit ein Ausdruck für die Vorteilhaftigkeit der SWPL. Bei einem negativen Ergebnis wäre die Einzelentwicklung vorzuziehen.

Für das Anwendungsbeispiel ergibt sich demnach ein deutlicher Vorteil der SWPL. Bei Gesamtkosten für die Einzelentwicklung von 1,1 Mio. Euro ermöglicht der SWPL Ansatz eine Einsparung von fast 50% der Entwicklungskosten.

"After initiation, the business case is maintained to examine new or alternative angles on the opportunity" [Clem02, S. 220]. Ein Business Case ist kein statisches Objekt, das einmal zu Beginn einer Investition zur Entscheidungsfindung herangezogen und anschließend nicht weiter verwendet wird. Im Gegenteil, der Business Case kann und sollte während der gesamten Laufzeit der Investition genutzt werden, um den aktuellen Fortschritt und die Einhaltung der Ziele zu kontrollieren. Investitionen decken einen langen Zeitraum ab, daher kommt es häufig während der Laufzeit zu Anpassungen. Diese sollten in den Business Case integriert werden. Dadurch können Entscheidungen, die im Laufe einer Investition anfallen, unterstützt werden. Bspw. wird die Frage, ob die SWPL um eine bestimmte Komponente erweitert werden soll, mittels eines Business Cases durch fundierte Finanzdaten leichter zu beantworten sein. Solche Fragen werden im nächsten Kapitel behandelt.

Die lange Laufzeit von Investitionen erhöht die Komplexität zusätzlich: Die Entscheidung ist mit einem Risiko verbunden, da Aussagen über Ereignisse in der Zukunft getroffen werden müssen, die nicht mit Sicherheit vorhergesagt werden können. So könnte eine technische Neuerung die Assets vor Ende der geplanten Nutzungsdauer unbrauchbar machen. Eine Möglichkeit zur Berücksichtigung von Unsicherheit sind Szenarios. Dabei werden mindestens zwei unterschiedliche zukünftige Entwicklungen prognostiziert und die entsprechenden Auswirkungen auf die Produktentwicklung berücksichtigt. Mittels einer Gewichtung der Szenarios mit Eintrittswahrscheinlichkeiten kann ein Erwartungswert ermittelt werden. In Tabelle 44 sind die Daten für das Anwendungsbeispiel dargestellt. Wie in den Basisdaten angegeben differenzieren die Szenarien nach der Marktresonanz auf die Einführung einer externen Schnittstelle im Fahrzeug. In Szenario 1 wird diese Schnittstelle zu einem Erfolg. Dementsprechend wird die WLAN Komponente nach zwei Jahren Laufzeit in die SWPL

[90] Englisch als Net Present Value (NPV) bezeichnet

integriert. Die Produkte 3 und 4 werden zusätzlich mit einer WLAN- und GSM-Schnittstelle ausgestattet. In Szenario 2 lehnen die Kunden die Schnittstelle mehrheitlich ab, weshalb von einer Implementierung der WLAN und GSM Komponenten in den Produkten 4 und 5 abgesehen wird. Szenario 1 wird als etwas wahrscheinlicher betrachtet und mit 60% gewichtet. Dementsprechend erfolgt die Gewichtung von Szenario 2 mit 40%. Aufgrund der höheren Wahrscheinlichkeit für eine umfangreichere Asset Base ist das Ergebnis positiver.

Zeit (Jahr)	2010	2011		2012		2013		2014	
Ereignisbeschreibung	Entwicklung Asset Base	Start Prod.1		Start Prod.2		Start Prod. 3+4		Start Prod.5	
Szenarios	1	1	2	1	2	1	2	1	2
Beschreibung		Externe Schnitt-stelle erfolgreich	Externe Schnitt-stelle nicht erfolgreich	WLAN integriert in SWPL		Prod. 3+4 beide mit WLAN und GSM	Prod. 4 ohne WLAN		Prod5 ohne WLAN und GSM
Wahrscheinlichkeit in %	100%	60%	40%	60%	40%	60%	40%	60%	40%
Kosten									
Domainentwicklung	398								
Produktentwicklung Wiederverwendung		13	13	28	28	97	50	48	35
Produktentwicklung neue SW		0	0	0	0	0	0	0	0
Domainwartung		0	0	120	0	0	0	0	0
Risiko Asset Neuentwicklung		0	0	0	0	0	0	0	0
COTS Verwendung		0	0	0	0	0	0	0	0
Höhere HW Kosten									
Benefits									
Aufwand bei kompletter Neuentwicklung		108	108	250	250	847	367	518	278
Berechnungen									
Summe Kosten	398	13	13	148	28	97	50	48	35
Summe Benefits	0	108	108	250	250	847	367	518	278
Gesamtbetrag	-398	95	95	102	222	750	317	470	243
Abzinsungsfaktor	8%	8%	8%	8%	8%	8%	8%	8%	8%
Wert im jeweiligen Jahr	-398	88	88	87	190	595	251	345	179
Gewichteter Wert im jeweiligen Jahr	-398	88		128		458		279	
Barwert der Investition in 2010								555	

Tabelle 44 – Business Case mit Berücksichtigung von Optionen (Werte in Tausend EUR, außer Prozent)

5.6.Schritt 5: Anwendungsmöglichkeiten des Modells

Im Fokus der bisherigen Analyse steht ein Unternehmen, das zu entscheiden hat, wie eine bestimmte Anzahl an Produkten ökonomisch sinnvoll entwickelt werden kann. Die Entwicklung kann entweder für jedes Produkt unabhängig oder im Sinne einer SWPL mit strategischer Wiederverwendung erfolgen. Ziel ist es, die bessere der beiden Alternativen zu identifizieren. Besser wurde dabei bisher als ökonomisch, sprich kosteneffizienter, verstanden. Der Business Case aus dem letzten Kapitel bewertete dazu am Ende die Differenz zwischen den Gesamtentwicklungskosten der beiden Alternativen. Dieses Vorgehen ist nicht auf alle Unternehmen und deren spezifische Situationen anwendbar. Teilweise existieren bereits Produkte, die nun weiterentwickelt werden müssen, oder ein Unternehmen nutzt bereits eine SWPL, die erweitert werden soll. Die Möglichkeiten, die das e-FACTS Modell zur Beantwortung solcher Fragen bietet, werden in Kapitel 5.6.1 erläutert. Ebenso stellen die Kosten nicht für alle Unternehmen die maßgebende Zielgröße dar. Die notwendigen

Anpassungen des Modells, um andere Unternehmensziele abzubilden, sind das Thema in Kapitel 5.6.2.

5.6.1. Unterschiedliche Produktszenarien

5.6.1.1. Entwicklung SWPL auf Basis existierender Produkte

Neue Produkte sind vielfach mehr oder weniger starke Veränderungen bereits bestehender Produkte. "Kaum ein Unternehmen wird direkt eine Produktlinie mit mehreren Produkten planen, ohne schon Systeme in dem entsprechenden Bereich realisiert zu haben. Ohne Kenntnis des Marktes und der technischen Realisierungsmöglichkeiten für die geplanten Produkte ist es sehr schwer, eine neue Produktlinie aufzusetzen und deren Variabilität direkt zu planen" [John04, S. 154]. Wenn ein Unternehmen bereits über einzelne Produkte verfügt, sollten diese bei der Analyse eines SWPL Ansatzes berücksichtigt werden. Grundsätzlich ist anzunehmen, dass sich der Entwicklungsaufwand durch die Existenz von bestehenden Produkten verringert, da ein gewisses Wiederverwendungspotenzial unterstellt wird. Sollten die Unterschiede zur geplanten SWPL Architektur allerdings zu groß sein, kann eine komplette Neuentwicklung vorteilhaft sein. Bei Einzelproduktentwicklung ist dies selten der Fall. Die notwendigen Veränderungen zur Erstellung einer neuen Produktversion sind meist geringer, da keine grundsätzlichen Architekturentwürfe berücksichtigt werden müssen.

Bei der Entwicklung von SWPL auf der Basis von existierenden Produkten können zwei Ansätze unterschieden werden. Entweder wird in einem ersten Schritt die vollständige Asset Base erstellt und anschließend die individuellen Produkte, oder es erfolgt ein schrittweiser Aufbau der Asset Base, indem neue Produkte zwar als Einzelprodukt entwickelt werden, dies aber entsprechend der Anforderungen an die SWPL Architektur geschieht. Dadurch ist eine nachträgliche Aufnahme der Produktkomponenten in die Asset Base möglich. Mit der Anzahl der neuen Produkte wächst die Asset Base, bis schließlich neue Produkte komplett aus der Asset Base heraus entwickelt werden können.

Der erste Ansatz bedingt meist eine zeitliche Verzögerung der Veröffentlichung von neuen Produkten, da die Entwicklungskapazitäten größtenteils für die Erstellung der Komponenten der Asset Base verwendet werden. Andererseits bietet dies den Vorteil, die SWPL direkt für das erste neue Produkt verwenden zu können. Für die Anwendung des e-FACTS Modells ergeben sich kaum Anpassungen. Es ist lediglich zu beachten, dass nicht die gesamte Funktionalität gezählt werden darf, da diese bereits als nutzbarer Code vorliegt. Die Erfassung

von zusätzlichen Funktionen ist mit der FP Methode unproblematisch. Komplexer ist die Berechnung, wenn Funktionstypen verändert werden. Dies erfordert meist eine anteilige Zählung. Zum Vergleich: SIMPLE bezeichnet die entsprechenden Kostenfunktionen als C_{cabu} für die SWPL und C_{evo} für Einzelentwicklungen. Zur Bestimmung von C_{evo} wird vorgeschlagen, einen anteiligen Wert der Entwicklungskosten des alten Produktes zu nehmen, z.B. 20%.

Eine schrittweise Einführung der SWPL ist in vielen Fällen durch Anforderungen des Marktes bedingt, die es unmöglich machen, die Veröffentlichung von neuen Produkten zu verschieben. Es werden also neue Produkte als Einzelprodukte entwickelt, da die SWPL nicht vollständig ist. Die Komponenten der Einzelprodukte werden in die Asset Base integriert, wodurch diese mit der Zeit wächst. Für das e-FACTS-Modell ergeben sich größere Veränderungen, da sich die Entwicklungsschritte zeitlich stark verschieben. Dies betrifft allerdings meist allein die zeitliche Betrachtung. Die Bewertung des Aufwands mittels FPs kann unverändert erfolgen. Ein Vorteil dieses Vorgehens ist die Verlagerung von Investitionen in die Zukunft, schließlich erfolgt der Aufbau der Asset Base nicht zu Beginn, sondern über einen gewissen Zeitraum. Das Unternehmen erhält dadurch die Möglichkeit, die Investitionsentscheidung zumindest teilweise zu revidieren. So könnte die Entwicklung der SWPL aus wichtigen Gründen eingestellt werden. Dadurch würden zusätzliche Aufwände für die Variabilisierung noch zu entwickelnder Assets eingespart, da diese Produkte als pure Einzelprodukte entwickelt werden könnten. Um diesen Effekt abzubilden bieten sich Szenarios an. Wie bereits vorgestellt bietet e-FACTS diese Möglichkeit. Abschließend sei darauf hingewiesen, dass ein langsamer Aufbau der SWPL die Gefahr birgt, dass einzelne Komponenten, die zwar schon Bestandteil der Asset Base sind, aber noch nicht im Rahmen einer SWPL angewendet werden, in den Einzelprodukten stark verändert werden und nicht alle Änderungen in die Asset Base zurückgespielt werden. In [Yosh06, S. 70] wird berichtet, dass in einer solchen Umgebung "the products become more or less completely stand-alone, although the products have so much in common with respect to the domain". Bei dem untersuchten Projekt erwies sich die Geschwindigkeit, mit der sich die Funktionen in ihrer Umsetzung unterschieden, als deutlich höher als erwartet. Daher erfordert eine schrittweise Einführung ein sehr diszipliniertes Vorgehen, um die Einhaltung der SWPL Architektur auch ohne verwendbare SWPL sicherzustellen [Eber03, S. 524].

5.6.1.2. Integration neuer Produkte in eine bestehende SWPL

Beim Design einer SWPL spielt die Festlegung der zukünftig mit Hilfe der SWPL zu entwickelnden Produkte eine entscheidende Rolle und sollte entsprechend sorgfältig durchgeführt werden. Dennoch ist es möglich, dass im späteren Verlauf weitere Produkte in die SWPL integriert werden sollen, z.B. bei einem großen Erfolg der SWPL oder durch nicht vorhergesehene technische Veränderungen, die einen neuen Produkttyp erzeugt haben.

Meist wird in diesen Fällen eine Anpassung der SWPL Architektur notwendig sein, da das entsprechende Produkt im damaligen Design nicht berücksichtigt wurde. Einige Assets werden ebenso anzupassen sein. Zusätzlich können neue Kosten durch etwaige organisatorische Veränderungen entstehen, z.B. wenn das zu integrierende Produkt bisher von einem anderen Team betreut wurde und nun ein Teil dieses Teams in die bestehende SWPL Organisation integriert werden muss.

Ansonsten sind die üblichen Kostenkategorien für die Entwicklung eines Produktes auf Basis einer SWPL zu berücksichtigen: Kosten durch die Wiederverwendung von Assets und die Kosten für die Entwicklung der produktindividuellen Bestandteile. Zur Beantwortung der Frage, ob e-FACTS dieses Szenario abbilden kann, ist zu klären, wie die Kosten für die Anpassung der Architektur und der bestehenden Assets erfasst werden können. Die FP Methodik ist in der Lage, diese Aufwände zu messen. Lediglich die organisatorischen Veränderungen können nicht direkt erfasst werden und sind, wie schon aus dem Basismodell bekannt, getrennt zu bewerten.

Eine mögliche Abwandlung dieses Szenarios besteht, wenn Unternehmen bereits mehrere SWPLs einsetzen. Dann ist zu fragen, in welche SWPL das neue Produkt integriert werden sollte. Ein einfacher Ansatz unter Verwendung des e-FACTS Modells besteht in der kompletten Berechnung der Produktintegration für alle SWPL und einem einfachen Vergleich der Ergebnisse. Es sollte diejenige SWPL ausgewählt werden, die die niedrigsten Gesamtkosten ermöglicht. Bei einer größeren Zahl an SWPL wird dieser Ansatz allerdings schnell undurchführbar. Als Alternative bieten sich Homogenitätsanalysen an[91], die SWPL hinsichtlich der Ähnlichkeit der enthaltenen Produkte bzw. Komponenten bewerten. Dies würde eine Vorauswahl ermöglichen, um den Analyseaufwand zur Anwendung des e-FACTS Modells zu reduzieren.

---

[91] Das SIMPLE Modell bietet eine solche Homogenitätsanalyse [vgl. Clem03, S. 29].

5.6.1.3.Evolution oder Revolution einer bestehenden SWPL

Der eigentliche Wert einer SWPL besteht in den wiederverwendbaren Assets. Über die Laufzeit der SWPL altern diese Assets genauso wie die grundlegende Architektur. Durch Wartung können notwendige Anpassungen umgesetzt werden, allerdings werden diese Wartungskosten mit der Zeit wahrscheinlich zunehmen. Dies bedeutet, dass ab einem gewissen Zeitpunkt eine komplette Neuentwicklung der SWPL kostengünstiger ist, als die fortwährende Wartung. Die Frage ist, ob e-FACTS Unternehmen bei der Bestimmung dieses Zeitpunktes unterstützen kann.

Das Modell ist in der Lage die Anpassungen einer Domain zu bewerten. Es können neue Produkte integriert werden (siehe Kapitel 5.6.1.2) und die Kosten für die Wartung der Assets können mittels FP Zählung oder pauschal durch Erfahrungswerte abgeschätzt werden (siehe Kapitel 5.4.4). Ebenso kann eine neue SWPL bewertet werden. Dies ist über die Basisfunktionalität des e-FACTS Modells abgedeckt, wenn die bisherigen Komponenten zum Großteil nicht weiterverwendet werden können, oder es ist Teil von Produktszenario 1.

5.6.2. Unterschiedliche Unternehmensziele

Bereits im Basisfragebogen (Kapitel 5.1.1.5.1) und bei der Erhebung der Basisdaten zu Beginn der Analyse (Kapitel 5.3.2) wurde die Zielsetzung der Unternehmung erfasst. Unternehmensziele sind die Grundlage unternehmerischen Handelns. Grundsätzlich sollte jede Tätigkeit auf die Erreichung dieser Ziele ausgerichtet sein. Daher haben unterschiedliche Ziele Auswirkungen auf die konkrete Ausgestaltung von SWPL. Dies kann soweit gehen, dass der Wiederverwendungsnutzen einzelner Funktionen in Abhängigkeit vom Unternehmensziel variiert [Schm03, S. 13]. Das e-FACTS Modell ermittelt eine ökonomische Bewertung eines SWPL Ansatzes für ein konkretes Unternehmen. Es berechnet die Vorteilhaftigkeit einer SWPL gegenüber einer Produkteinzelentwicklung. Grundsätzlich sind die damit getroffenen Aussagen zunächst einmal für Unternehmen korrekt, die eine Gewinnmaximierung anstreben. Weitere Unternehmensziele können vielfach durch eine Anpassung der Berechnungsparameter berücksichtigt werden. Dies wird im Folgenden beleuchtet.

5.6.2.1.Time-to-Market

Die Entwicklungskosten für ein Produkt sind quasi der Enabler, die die späteren Umsätze aus der Vermarktung des Produktes ermöglichen. Das Ziel minimaler Entwicklungskosten kann

aus Unternehmenssicht kontraproduktiv sein, z.B. wenn durch die Einführung einer SWPL und durch den dafür notwendigen Aufbau der Asset Base der Markteintritt verschoben wird. Der daraus resultierende Umsatzverlust kann die eingesparten Entwicklungskosten überkompensieren. Die Bedeutung einer schnellen Markteinführung hängt vom Reifegrad eines Marktes ab. In einem reifen Markt mit geringen Wachstumsraten führt eine verspätete Einführung zu vergleichsweise geringen Umsatzverlusten. Die gleiche Verzögerung in einem innovativen Markt zu Beginn des Lebenszyklus kann signifikant höhere Einbußen zur Folge haben. Der Unterschied zu einem reifen Markt verringert sich, je später die ursprüngliche Produkteinführung geplant war. In Abbildung 27 sind die Zusammenhänge grafisch veranschaulicht.

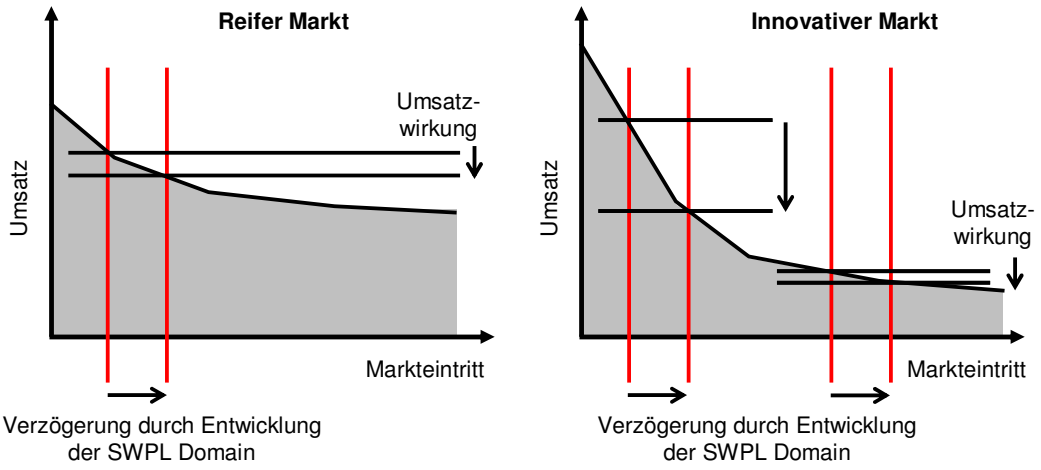

Abbildung 27 – Bewertung der Time-to-Market [Schm03, S. 6]

Es kann sich demnach lohnen, höhere Entwicklungskosten zu akzeptieren, um eine frühere Markteinführung zu ermöglichen. Im Rahmen einer SWPL bedeutet dies, dass die eigentliche Zielgröße nicht mehr die Erhöhung der Wiederverwendungsanzahl ist, sondern eine möglichst frühzeitige Fertigstellung der ersten Produkte. Bedenkt man nun, dass die Erstellung einer Komponente als Bestandteil der Asset Base im Allgemeinen einen höheren Entwicklungsaufwand erzeugt, kann es sinnvoll sein, bestimmte Komponenten zu Beginn bewusst als Einzelentwicklung zu realisieren, obwohl die ökonomische Analyse für eine Wiederverwendung spricht.

Dies kann im e-FACTS Modell durch eine Anpassung der Optimierungsphase berücksichtigt werden. Bisher war die Entscheidungsbasis der FP Wert über alle Wiederverwendungen im

Vergleich mit ebenso vielen Einzelentwicklungen. Dies ist anzupassen. Für Komponenten der ersten Produkte, für die ein früher Markteintritt als kritisch eingestuft wird, erfolgt eine getrennte Bewertung. Die Differenz zwischen FP minimaler Lösung für die gesamte SWPL und einer lokalen Minimierung der FP Anzahl für das erste Produkt wird berechnet. Dabei ist der Aufwand für eine nachträgliche Integration der Komponenten des ersten Produkts in die Asset Base bei den Kosten für die frühe Produktentwicklung zu berücksichtigen. Nach einer Umwandlung in Kosten kann dieser Wert mit den erwarteten Umsatzausfällen verglichen werden. Sollten die Umsatzausfälle größer sein als der Mehraufwand für eine schnellere Fertigstellung des ersten Produkts, sollte eine Einzelfertigung erfolgen.

5.6.2.2.Wachstum

Für Unternehmen mit sehr ambitionierten Wachstumszielen kann eine SWPL die einzige Möglichkeit sein, die gesteckten Ziele zu erreichen. So war die ökonomische Bewertung des SWPL Ansatzes für Cummins Inc., einen amerikanischen Hersteller von Dieselmotoren, "pretty straight-forward: they couldn't deliver to the existing product plan without it. It was as simple as that" [Clem02, S. 423]. Bei der gegebenen Menge an neu zu entwickelnden Produkten und der fehlenden Möglichkeit, zusätzliche Entwickler einzustellen, bot allein ein deutlich gesteigerter Wiederverwendungsanteil eine Lösung. Der Modellfokus verschiebt sich dadurch von einer reinen Kostenorientierung zu einer Produktivitätsorientierung. e-FACTS kann in diesem Zusammenhang genutzt werden, um SW-Werkzeuge zu bewerten. In einem ersten Schritt lassen sich die Aufwände für selbstentwickelte Werkzeuge ermitteln und mit Kosten bewerten. Für COTS Werkzeuge ist dieser Schritt nicht notwendig, da die Kosten unmittelbar verfügbar sind. Anschließend kann versucht werden, die Produktivitäts-veränderung durch den Einsatz des Werkzeugs mittels FP Methodik vorab zu schätzen. Dies wird allerdings nicht in allen Fällen möglich sein, da mittels FPs nur solche Veränderungen gemessen werden können, die Funktionstypen betreffen. Ein Werkzeug zur Vereinfachung der Arbeitsabläufe eines Entwicklers kann nicht erfasst werden. Hier bleibt die Möglichkeit, die Produktivitätsveränderung nachträglich zu erfassen und mit den entsprechenden Entwicklungs- oder Anschaffungskosten zu vergleichen.

SWPL unterstützen Wachstumsziele außerdem allgemein, da die notwendige SWPL Architektur bei ihrer Entwicklung eine verlässliche Antizipation zukünftiger Entwicklungen und Trends voraussetzt. Dies setzt antizipatives Denken und Handeln in den Unternehmen hinsichtlich der zukünftigen technischen Veränderungen voraus, denn eine spätere Anpassung

der Architektur wäre, wenn überhaupt möglich, mit hohen Kosten verbunden. Daraus resultiert eine umfassende Zukunftsanalyse, die die Wahrscheinlichkeit erhöht, in Zukunft mit den richtigen Produkte erfolgreich zu sein.

5.6.2.3.Kundenbindung

Gelingt es dem Unternehmen, seinen Kunden immer die richtigen Produkte zum richtigen Zeitpunkt und zum richtigen Preis anzubieten, sollte daraus langfristig eine hohe Kundenzufriedenheit und entsprechende –Bindung entstehen. Die genannten Punkte sind mögliche Ergebnisse einer SWPL. Allerdings erweist sich die Bewertung einer erhöhten Kundenbindung als komplex. Ein möglicher Ansatz findet sich bei [Clem03, S. 27f.]. Dort wird zunächst eine Segmentierung der Kunden durchgeführt, sodass sich hinsichtlich der zu untersuchenden Produktdimension[92] intern homogene, extern aber möglichst inhomogene Gruppen ergeben. Anschließend wird die Zufriedenheit der Kunden erfasst. Die direkte Messung der Kundenzufriedenheit ist nur begrenzt möglich, es existieren aber Indikatoren, die erfasst werden können. Ein Indikator ist der Anteil der Kunden, die wahrscheinlich beim nächsten Mal erneut ein Produkt des Unternehmens kaufen werden. Misst man diesen Wert vor und einige Zeit nach der Einführung einer SWPL kann man den Ertrag ermitteln, indem die Veränderung mit dem durchschnittlichen Umsatz der Kundengruppe multipliziert wird. Der Effekt einer SWPL ist demnach mit einigem Aufwand näherungsweise bestimmbar.

Eine weitere Veränderung der Methodik ist nicht notwendig, da die bereits angesprochenen Anpassungen für schnelle Produktentwicklung, eine schnelle Zunahme der angebotenen Produkte oder die reine Grundform des Modells mit einem Fokus auf eine Reduzierung der Entwicklungskosten ausreichen.

---

[92] Dabei könnte es sich um den Preis oder eine bestimmte Funktion handeln.

6. Nachweis der Praktikabilität von e-FACTS und individuelle Anwendung

In Kapitel 5 wurde das e-FACTS Modell im Detail hergeleitet und mögliche Anwendungen beschrieben. In Kapitel 6 erfolgt nun eine Plausibilisierung. Damit verbunden soll die Praktikabilität des Modells gezeigt werden. Es handelt sich explizit nicht um eine wissenschaftliche Validierung des Modells. Obwohl eine solche Validierung wünschenswert wäre, ließ die spezifische Situation bei Produktfamilien im automobilen Embedded Systems Bereich aufgrund der geringen Anzahl an verfügbaren Fallstudien keine echte Validierung zu. Es liegen derzeit keine ausreichenden Daten über abgeschlossene SWPL Projekte vor, die für eine solche Analyse herangezogen werden könnten. Schließlich bestehen hohe Anforderungen, damit ein Projekt verwendet werden kann. So müssen z.B. Vergleichswerte aus einem vorherigen Projekt im gleichen Unternehmen vorliegen. Wie bereits in Teil A Kapitel 5.5 angesprochen werden keine Benchmarkwerte herangezogen, sondern allein die individuellen Gegebenheiten des betroffenen Unternehmens berücksichtigt.

In der Literatur finden sich zwar zahlreiche Beispielanwendungen (siehe Teil A Kapitel 2.2.6), aber auch hier beschränkt sich die Betrachtung auf einzelne Anwendungsfälle. Eine einheitliche Datenbank mit einer signifikanten Anzahl an vergleichbaren SWPL Projekten ist nicht bekannt. Zusätzlich differieren die vorhandenen Beispiele meist stark, da sie ursprünglich mit unterschiedlichen Zielsetzungen verfolgt wurden.

Deshalb wurde das Modell im Rahmen dieser Arbeit in zwei realen Situationen empirisch getestet. Diese beiden Anwendungsbeispiele werden im Folgenden im Detail vorgestellt und anschließend bewertet. Die weitergehende Validierung des Modells kann in zukünftigen wissenschaftlichen Arbeiten verfolgt werden.

6.1. DOMINION des DLR

Zum Nachweis der praktischen Anwendbarkeit des im Rahmen dieser Arbeit entwickelten Modells wurde eine praktische Anwendung beim Deutschen Zentrum für Luft- und Raumfahrt e. V. (DLR) durchgeführt. Das DLR wurde 1967 gegründet und beschäftigt heute ca. 5.700 Mitarbeiter, die sehr unterschiedliche Themenbereiche bearbeiten. Neben dem bereits aus der Namensgebung ableitbaren Fokus auf Luft- und Raumfahrt, bilden heute Energie und Verkehr weitere Forschungsgebiete. Im Bereich Verkehr untersucht das DLR

Verkehrssysteme, Verkehrsmanagement und generell bodengebundene Fahrzeuge, sprich Straßen- und Schienenfahrzeuge.

Für die vorliegende Arbeit von besonderem Interesse sind die Arbeiten an Fahrerassistenzsystemen. Ein Ziel des DLR ist es, ein hochautomatisiertes Fahrzeug zu realisieren, das den Fahrer weitgehend beim Fahren unterstützt, ihn jedoch im Gegensatz zu vollautomatischen Ansätzen nicht ersetzt. Um dieses Ziel zu erreichen, werden sowohl Simulatoren als auch reale Fahrzeuge zu Testzwecken eingesetzt. Das duale Vorgehen stellt besondere Anforderungen an die Softwareentwicklung, da neue Anwendungen in beiden Umgebungen mit geringem zusätzlichem Aufwand einsetzbar sein müssen.

Zu diesem Zweck wurde am Institut für Verkehrssystemtechnik (TS) des DLR eine SOA inspirierte Entwicklungs- und Laufzeitumgebung für verteilte Embedded Systems im Automobil namens DOMINION entwickelt. Im Rahmen eines eintägigen Workshops wurde Dominion als Basis für eine Produktfamilie mit dem e-FACTS Modell analysiert. Genauer betrachtet handelt es sich dabei um einen Teil von Dominion, da innerhalb des TS inzwischen zahlreiche Erweiterungen von Dominion entwickelt wurden. Bspw. wird es nicht allein im Bereich Fahrzeugtechnik, sondern auch für Flugzeugsimulatoren eingesetzt. Eine genaue Abgrenzung von Dominion erfolgt im weiteren Verlauf.

6.1.1. Besonderheiten des DLR

Das DLR ist ein eingetragener Verein und kein klassisches gewinnorientiertes Unternehmen. Vielmehr versteht sich das DLR als Forschungszentrum. Daraus ergibt sich, dass nicht in erster Linie die Entwicklung eines marktreifen Produktes im Vordergrund steht, sondern vielmehr die Beantwortung von Fragen hinsichtlich Machbarkeit.

Bei der Analyse der Ausgangssituation erreicht das TS einen beachtlichen Wert von 85%. Dennoch gibt das Bewertungssystem eine negative Empfehlung, da eines der Ausschlusskriterien nicht erfüllt ist. Bei der Anzahl der in der Softwareentwicklung tätigen FTEs wird nur eine Einstufung von 1 oder 2 erreicht. Aufgrund der Ausrichtung des DLR als Forschungseinrichtung ist die Bestimmung der FTEs aufwendiger als in einem klassischen Unternehmen, da es keine klare Aufgabenzuordnung im Sinne eines typischen Softwareentwicklers gibt. Das Aufgabenspektrum der Mitarbeiter ist wesentlich weiter gefasst. Eine grobe Schätzung ergab einen Wert von ca. 5 FTEs, die mit der reinen Entwicklung von Software beschäftigt sind.

An dieser Stelle soll noch einmal daran erinnert werden, welchen Zweck die Ausschlusskriterien erfüllen. Es bedeutet nicht automatisch, dass der SWPL-Ansatz bei Nichterfüllung eines solchen Kriteriums nicht erfolgreich sein kann. Vielmehr stellt ein solcher Fall allein ein höheres Umsetzungsrisiko dar. Daher sollte vor einer Detailanalyse eine geeignete Einschätzung des damit verbundenen Risikos erfolgen. Im Falle des DLRs ergibt sich, dass die geringe Anzahl an Entwicklern aufgrund der Forschungsorientierung vernachlässigt werden kann. Bei der Umsetzung der SWPL zeigten sich dadurch keine negativen Auswirkungen.

6.1.2. Beschreibung der Softwareumgebung DOMINION

6.1.2.1. Funktionsbeschreibung

Das Institut für Verkehrssystemtechnik hat Dominion mit dem Ziel entwickelt, die Entwicklungszeiten für neue Funktionen deutlich zu reduzieren. Vor der Einführung von Dominion spielte die Abstimmung der Entwickler untereinander eine wichtige Rolle. Trotz der niedrigen Zahl von 5 FTEs entwickeln zahlreiche Personen im TS Software. Die notwendigen Detailinformationen zu allen benötigten Input- und Outputdaten, z.B. allen Schnittstellen für eine neue Funktion mussten in Gespräche mit den anderen Entwicklern gewonnen werden. Zusätzlich entstand Doppelaufwand, da die meisten Funktionen sowohl für die Simulations- als auch die Realumgebung entwickelt werden sollten.

Dominion fungiert nun als Middleware und stellt dem Entwickler standardisierte Funktionen zur Kommunikation zur Verfügung. Dadurch kann sich der Entwickler auf die reine Funktionalität statt der Implementierung konzentrieren. Dominion bietet eine automatische Codegenerierung, die spezifische Code-Templates für unterschiedliche Implementierungsumgebungen generiert. Der Entwickler erstellt dazu zunächst eine semiformale Beschreibung der neuen Funktion. Dadurch werden Daten, Services und deren Zusammenhänge implementierungsunabhängig definiert. Die Codegenerierung erfolgt anschließend automatisch. Zusätzlich werden etwaig notwendige Änderungen auf der Kommunikationsebene durch eine Neukompilierung automatisch übernommen. Ein großer Vorteil von Dominion ist seine Plattformunabhängigkeit. Derzeit werden folgende Betriebssysteme unterstützt: Windows, Linux, Mac OSX, Windows Mobile und iPhoneOS. Die Dominion Architektur ist in Abbildung 28 dargestellt.

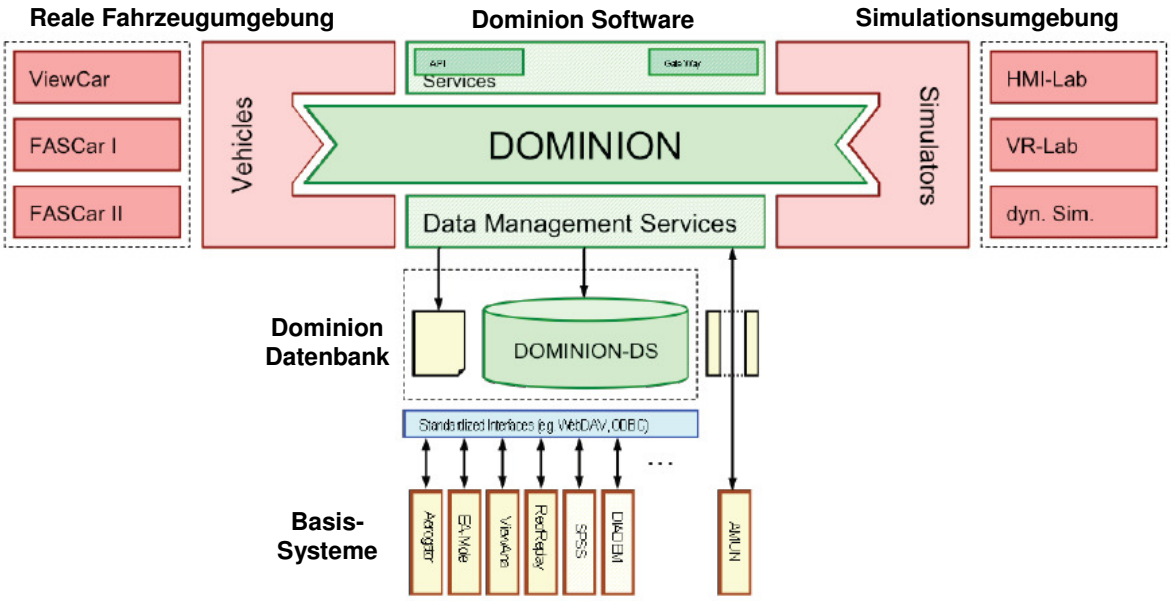

Abbildung 28 – Übersicht der Dominion Architektur

6.1.2.2. Verwendung als Produktfamilie

Das DLR beabsichtigte mit der Entwicklung von Dominion in erster Linie die Entwicklungszeiten für neue Funktionen durch Auslagerung und Automatisierung von Standardelementen zu reduzieren. Eine Produktfamilie war nicht das erklärte Ziel der Entwicklung. Dennoch eröffnet Dominion die Möglichkeit, mehrere Produkte auf dieser Basis zu entwickeln. Aufgrund der bereits angesprochenen Besonderheiten des DLR ist aktuell nicht absehbar, ob und wenn ja, wann Dominion und die darauf aufbauenden Produkte tatsächlich vermarktet werden. Unabhängig davon soll im Folgenden kurz erläutert werden, wie eine solche Produktfamilie aussehen könnte und warum sie sinnvoll wäre:

Zur Erprobung neuer Funktionen sind umfangreiche Tests erforderlich. Diese sind sowohl in einer kontrollierten Umgebung (Simulation) als auch im realen Fahrzeug im Straßenverkehr notwendig. Simulationen bieten insbesondere den Vorteil, dass sie ohne Risiko für den Tester oder andere Verkehrsteilnehmer durchgeführt werden können. Außerdem bieten sie die Möglichkeit, standardisierte Situationen immer wieder von neuem und unter identischen Bedingungen durchzuführen. Daher wäre Dominion für alle Entwickler von Fahrzeugsystemen sinnvoll. Die Umsetzung stößt allerdings an Grenzen, da für ein reales Fahrzeug Dominion allein nicht ausreicht. Da die zu testenden Systeme auf Sensordaten des Fahrzeugs und Daten des Fahrers zugreifen müssen, ist eine Anbindung an die elektronischen Systeme im Fahrzeug notwendig. Dieser Schritt erfordert derzeit manuelle Anpassungen, um

die Kommunikation mit der individuellen Fahrzeughardware zu ermöglichen. Das TS arbeitet bereits an einer automatischen Generierung dieser Schnittstelle. Einzige Voraussetzung wird sein, dass sich die OEMs an den Vector Standard halten.

Derzeit bietet sich eher die Simulationsumgebung als Bereich für mögliche Produktvarianten an. Zur Erprobung neuer Funktionen sind zahlreiche unterschiedliche Simulationsumgebungen im Einsatz. Dies reicht von Vollsimulatoren, bei denen der Fahrer in einem echten Fahrzeug sitzt, die realen Instrumente bedient, das Fahrzeug über Hydrauliksysteme bewegt und die Umgebung mittels Bildschirmen simuliert wird, bis hin zur reinen Sitzkiste mit einer minimalistischen Nachbildung eines Fahrzeugs. Der letztgenannte Simulatortyp wird bspw. eingesetzt, um 3D Modelle vom Inneren eines neuen Fahrzeugs zu testen. Daneben werden z.B. Kompaktsimulatoren eingesetzt. Diese unterscheiden sich von Vollsimulatoren dadurch, dass zahlreiche für den jeweiligen Test unwichtige Features weggelassen werden. Meist handelt es sich dabei um die hydraulische Bewegung des Simulators oder eine reduzierte grafische Darstellung der Umgebung. Weitere Unterscheidungsmerkmale von Simulatoren sind die Stereodarstellung, also die 3D Sicht für den Fahrer oder Headtracking. Letztgenanntes bezeichnet die Aufzeichnung der Kopf- und Augenbewegungen des Fahrers. Dadurch kann z.B. überprüft werden, wie stark die Ablenkung des Fahrers durch die Benutzung einer neuen Funktion ist.

6.1.2.3.Anwendungsmöglichkeiten des e-FACTS Modells

Für diese unterschiedlichen Anwendungsbereiche bietet Dominion als Middleware die Möglichkeit, den Entwicklungsaufwand für neue Funktionen deutlich zu reduzieren. Die Entwickler könnten sich auf die Funktionsentwicklung konzentrieren, unabhängig von der späteren konkreten Anwendungsumgebung. Somit wäre Dominion als Entwicklungsumgebung vermarktbar, alternativ könnten zahlreiche Produkte (mit dem Fokus auf Simulatoren) angeboten werden. e-FACTS kann diese Vorhaben auf zwei Arten unterstützen. Einerseits kann das DLR damit bestimmen, welchen Aufwand die Produkterstellung erzeugen würde und ob sich die Erstellung einer auf Dominion basierenden Simulationsproduktlinie lohnen würde. Eine solche Berechnung im Rahmen des eintägigen Workshops erschien allerdings aufgrund fehlender Pläne und weitergehender Daten nicht sinnvoll.

Andererseits besteht die Möglichkeit zu ermitteln, ob sich die Investition des DLR in die Erstellung der DOMINION Umgebung amortisiert hat.[93] Dieses Vorgehen wurde verfolgt. Dazu wurden insgesamt 7 zentrale Funktionen analysiert, die aufbauend auf Dominion entwickelt wurden. Im Folgenden werden diese Funktionen beschrieben.

6.1.3. Detailbeschreibung der betrachteten Funktionen

Der Fokus des TS liegt in der Entwicklung von Fahrerassistenzsystemen sowohl im Simulator- als auch im Echtbetrieb. Simulation deckt verschiedene Grade ab, z.B. Vollsimulatoren mit einem echten Fahrzeug, welches entsprechend der Simulationsumgebung und der Aktionen des Fahrers hydraulisch bewegt wird. Es besteht aber auch die Möglichkeit einzelne Elemente eines Fahrzeugs separat zu simulieren (Kompaktsimulatoren). Das TS entwickelt neben Funktionen für Serienfahrzeuge auch Anwendungen, die Besonderheiten einer Simulationsumgebung abdecken, wie z.B. die Simulation der realen Bewegungen eines Fahrzeugs in einem Simulator mittels hydraulischer Systeme oder die Simulation des Umgebungsverkehrs.

Innerhalb des Workshops wurde gemeinsam eine Auswahl von 7 Funktionen getroffen, die einen möglichst repräsentativen Querschnitt aller Funktionen abbilden. Dies soll sicherstellen, dass die Teilergebnisse nachher möglichst einfach auf die Grundgesamtheit hochgerechnet werden können. Der Anteil der 7 analysierten Funktionen am gesamten Entwicklungsaufwand beträgt ungefähr 10%.

6.1.3.1.ACC

ACC steht für Advanced Cruise Control. Dieses System unterstützt den Fahrer, indem er eine bestimmte Geschwindigkeit wählen kann, die das System dann selbständig einhält. Die Funktion Cruise Control (CC) oder Tempomat, die bereits länger im Einsatz ist, würde die Geschwindigkeit auch dann halten, wenn der Sicherheitsabstand zu einem vorausfahrenden Fahrzeug zu gering würde. In diesem Fall müsste der Fahrer die Geschwindigkeit durch Abschaltung des Systems oder Betätigung der Bremse verringern. ACC bietet zusätzlich die Möglichkeit, den vorausfahrenden Verkehr bei der Wahl der Geschwindigkeit zu berücksichtigen. Die gefahrene Geschwindigkeit ergibt sich also nicht allein aus der vom Fahrer gewählten Geschwindigkeit, sondern auch aus dem gewünschten zeitlichen Abstand zum vorausfahrenden Fahrzeug und dem aktuellen Abstand. Sollte dieser zu gering werden,

[93] Für den hier betrachteten Funktionsumfang von DOMINION sind 18 PM Entwicklungszeit aufgewendet worden.

würde das System das Fahrzeug selbständig verzögern. In diesem Fall sind mehrere Situationen denkbar. Einige Systeme unterstützen die Verzögerung lediglich bis zu einem bestimmten Prozentwert der maximalen Verzögerung. Sollte ein stärkeres Abbremsen des Fahrzeugs notwendig sein, z.B. beim Auffahren auf ein Stauende auf der Autobahn, wird der Fahrer akustisch und optisch gewarnt, muss die Aktion aber selbst ausführen. Neuere Systeme bieten die Möglichkeit, das Fahrzeug im Notfall bis zum Stillstand automatisch zu verzögern.

Das hier entwickelte ACC bietet all diese Möglichkeiten. Insgesamt sind drei Zustände möglich: Freie Fahrt (das System ist abgeschaltet), Tempomat und ACC. Für die Funktion Tempomat werden folgende Inputparameter benötigt: Gaspedalstellung, Bremspedalstellung, Einschlagwinkel des Lenkrads, gewünschte Geschwindigkeit (v_{des}) und die aktuelle Geschwindigkeit (v_x). Für die Funktion des ACC werden zusätzlich benötigt: gewünschter zeitlicher Abstand zum vorausfahrenden Fahrzeug (t_{des}) und der aktuelle Abstand zum Ziel auf der Fahrbahn (Δs_{act}). Das Ziel bezeichnet dabei das vorausfahrende Fahrzeug, zu dem der Abstand relevant ist (ein Fahrzeug auf der rechten Spur einer Autobahn wird als Ziel ignoriert, da ein Überholmanöver möglich ist).

Die Bestimmung der korrekten Entfernung zum Ziel erfolgt hier außerhalb des ACC Systems. Das ACC beschränkt sich auf die Berechnung der notwendigen Gas- und Bremspedalstellung. Dazu wird eine Positionskontrolle implementiert, die zu einem kaskadierten, geschlossenen Regelkreis führt. Der Ablauf wird in Abbildung 29 dargestellt. Die drei Begrenzer sorgen dafür, dass die berechneten Differenzen zwischen Ist- und Soll-Geschwindigkeit nicht unmittelbar, sondern unter Berücksichtigung physikalischer Grenzen für alle Insassen angenehm und entsprechend zeitlich leicht verzögert ausgeglichen werden.

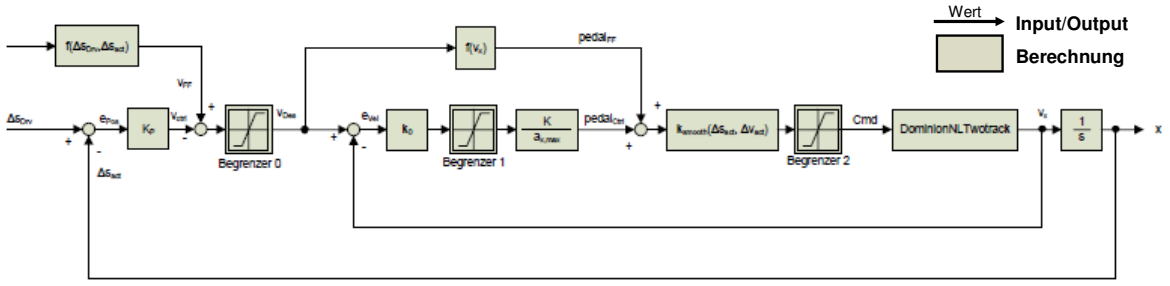

Abbildung 29 – Funktionsweise der ACC Regelung

6.1.3.2.Bewegungssteuerung

Ein Vollsimulator simuliert die Bewegung eines echten Fahrzeugs entsprechend der virtuellen Bewegungen in der Simulationsumgebung. Der Simulator entspricht weitgehend den bekannteren Flugsimulatoren. Es werden die Fliehkräfte beim Beschleunigen und Bremsen und auch bei Kurvenfahrten simuliert. Basierend auf den Inputdaten des Fahrers (Lenkeinschlag, Gas und Bremse) und den Umgebungsdaten der Simulation (Geschwindigkeit, Rotation usw.) ermittelt die Bewegungssteuerung die notwendigen Daten für die Ansteuerung der Hydraulik zur Bewegung des Testfahrzeugs im Simulator. Aufgrund der dreidimensionalen Bewegung sind hierfür zahlreiche Algorithmen notwendig, die teilweise hoch komplex ausfallen.

6.1.3.3.Fahrdynamikmodell

Mittels des Fahrdynamikmodells wird die Bewegung des Testfahrzeugs simuliert. Daher spielt diese Funktion lediglich für die Simulation eine Rolle. Es werden sowohl translatorische Bewegungen (Längs-, Quer-, und Hubbewegungen) als auch rotatorische Bewegungen (Gieren, Nicken und Wanken) berechnet. Diese Daten sind wichtig für zahlreiche weitere sicherheitskritische Systeme, wie z.B. ABS. Aber auch die Motorsteuerung und das Fahrwerk benötigen diese Informationen.

6.1.3.4.SeSiLa

SeSiLa steht für Sensorsimulation/Laserscanner. Diese Funktion wurde lediglich für die Simulation entwickelt. Ziel ist es, auch in der Simulationsumgebung möglichst realistische Daten für die Anwendung eines Laserscanners zur Verfügung zu haben. Dazu wird die tatsächliche Funktionalität eines Laserscanners imitiert. Als Basis dienen keine realen Gegenstände, sondern die virtuellen Objekte der Simulationsumgebung. In einem ersten Schritt werden die relevanten Objekte ausgewählt (bis zu 50 Einzelobjekte inklusive des eigenen Fahrzeugs) und in einem Koordinatensystem verordnet. Anschließend wird die Funktion des Lasers, das sogenannte Schießen, simuliert. In dem zugrundeliegenden realen System werden mittels zahlreicher Laserschüsse Punktewolken erzeugt, die jeweils die Situation vor dem Fahrzeug wiedergeben. Dabei wird der Abstand der Objekte zur Fahrzeugfront erfasst. Anschließend werden Objekte erkannt und Umgebungslinien für diese Objekte erzeugt. Schließlich ergeben sich Ergebniskonturlinien, die die realen Objektgrenzen wiedergeben. Diese Informationen werden dann in anderen Systemen weiterverarbeitet. Dies

ist lediglich eine sehr grobe Darstellung der Funktionsweise, soll aber für den Zweck dieser Arbeit genügen.

In der Simulation werden mittels der Daten über die virtuellen Objekte ebenso Konturlinien erzeugt, die dann in weiteren Systemen genutzt werden können. Dazu ist es notwendig jeden Schuss des Lasers zu simulieren und daraus die korrespondierenden Ergebniskonturlinien zu berechnen.

6.1.3.5.Displaykonzept

Die Funktion Displaykonzept berechnet Daten zur aktuellen Situation des Fahrzeugs im Verkehr und stellt diese anderen Systemen zur Verfügung. Diese Funktion ist auf die Simulationsumgebung beschränkt, da die entsprechenden Informationen lediglich hier benötigt bzw. ausgewertet werden. Es werden u.a. Daten über die Geschwindigkeit, die Rotation, den Abstand zu anderen Fahrzeugen und die Spur erfasst, berechnet und gespeichert. So bildet diese Funktion eine notwendige Grundbedingung für die Funktion ACC, da die Daten über den Abstand zum vorausfahrenden Fahrzeug hier generiert werden. In einem realen Testfahrzeug würde hierfür ein Laserscanner oder Radargerät genutzt. Eine grafische Bearbeitung erfolgt innerhalb der Funktion Displaykonzept nicht.

6.1.3.6.Nebenaufgaben

Diese Funktion wurde lediglich für die Anwendung im Simulator entwickelt. Ziel ist es, die Ablenkung des Fahrers durch verschiedene Aufgaben zu testen. Mehrere Test-Fahrer durchlaufen dabei eine standardisierte Testfahrt. Über ein Zusatzdisplay auf dem Beifahrersitz werden den Fahrern zahlreiche Fragen gestellt, z.B. welche Farbe haben die dargestellten Kreise, oder welcher der Kreise ist der größte. Die Beantwortung erfolgt über den Bildschirm, einen Touchscreen, selbst. Das System misst die Antwortzeiten, erfasst die Antworten und die Fahrt des Fahrers im Simulator. Diese Daten werden dann für eine spätere Auswertung bereitgestellt.

6.1.3.7.Cabin

Wie bereits oben erwähnt ist eine Anpassung der Software an die individuelle Hardware eines jeden Testfahrzeugs notwendig. Dies ergibt sich dadurch, dass Sensordaten und das Verhalten des Fahrers als Inputdaten benötigt werden, diese jedoch nicht standardisiert in allen Fahrzeugen vorhanden sind. Ebenso werden die Ergebnisse der Funktionen an das Fahrzeug zurückgespielt. Beim ACC erfolgt bspw. eine tatsächliche Beeinflussung der Beschleunigung

und Verzögerung des Fahrzeugs. Ebenso ist eine individuelle Realisierung für die Simulation notwendig, da hier die Umgebungsdaten lediglich simuliert werden und in das reale System des Fahrzeugs (bei Vollsimulatoren) übertragen werden müssen. Dies bedeutet, dass der Fahrer z.B. über das reale Kombiinstrument eine Information zur virtuellen Geschwindigkeit bekommt. Bei anderen Typen von Simulatoren kann es erforderlich sein, andere Schnittstellen zu bedienen, da möglicherweise keine reale Hardware eingesetzt wird, sondern lediglich eine Simulation erfolgt, z.B. ein virtueller Tacho integriert in die Darstellung der Umgebung. Natürlich sind auch alle Zwischenstufen denkbar und müssen unterstützt werden.

Aufgrund dieser Notwendigkeit zur individuellen Anpassung wurde Cabin nicht in Dominion integriert, sondern als separate Funktion realisiert. Cabin stellt daher die Schnittstelle zur jeweiligen Hard- oder auch Software her. Dies bedeutet insbesondere eine Beschäftigung mit den Kommunikationsnetzen in modernen Fahrzeugen, z.B. einem CAN-Bus. Sollte das TS ein anderes Testfahrzeug verwenden wollen, müsste auch die Anwendung Cabin neu entwickelt werden.

6.1.4. Aufwandsbewertung mittels FP

Der Fokus des Workshops lag auf der Bestimmung der FPs für 7 ausgewählte Funktionen. Für alle wurde sowohl eine Berechnung mit als auch ohne Wiederverwendung durchgeführt. Wiederverwendung bedeutet hierbei, dass die Funktion unter Nutzung von Dominion als Middleware realisiert worden ist. Die Ergebnisse sind in Tabelle 45 und 46 dargestellt. Da die einzelnen Funktionen jeweils für Dominion entwickelt wurden, musste die entsprechende Entwicklung ohne Dominion abgeschätzt werden. Die festgestellten Unterschiede beziehen sich in beiden Zählungen hauptsächlich auf die EI, EO und EIF. Bei Berücksichtigung der Funktionalität von Dominion wird dies leicht verständlich. Dominion erleichtert die Kommunikation mit den Systemen des Fahrzeugs oder der Simulationsumgebung, indem der Entwickler hier lediglich eine semiformale Beschreibung erstellt. Die eigentliche Codegenerierung erfolgt dann automatisch. Die Funktion Cabin stellt die eigentliche Anbindung an die Hardware im Fahrzeug her, so dass hier trotz Dominion EIs und EOs notwendig sind. Im Durchschnitt erfolgt aber auch hier eine Reduzierung des Aufwands z.B. für die CAN-Bus Implementierung um ca. 40%.

Nr.	Funktionsbezeichnung	EI			EO			EQ			AL			DF			EIF		
		L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
																Total FP unadjusted		=	333
1	Acc	9			1						1			1			1		
2	Bewegungssteuerung		5			1					12		3				1		
3	Fahrdynamikmodell	2				2						3		1			1		
4	Sensorsimulation / SeSiLa	1		1			1				1	3		1	1		1		
5	Displaykonzept	8				1						3		1			2		
6	Nebenaufgaben	2			2						1			1			1		
7	Cabin	4			7							1		2			2		

Tabelle 45 – Feature Point Berechnung ohne Wiederverwendung (gelbe Felder zur Eingabe von Werten)

Nr.	Featurebezeichnung	FP ohne WV	EI			EO			EQ			AL			DF			EIF		
			L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
																	Total FP(resuse) unadjusted		=	137
1	Acc	43										1			1					
2	Bewegungssteuerung	69										12		3						
3	Fahrdynamikmodell	35											3		1					
4	Sensorsimulation / SeSiLa	44										1	3		1	1				
5	Displaykonzept	53											3		1					
6	Nebenaufgaben	26										1			1					
7	Cabin	63	1			3							1		2			1		

Tabelle 46 – Feature Point Berechnung mit Wiederverwendung (gelbe Felder zur Eingabe von Werten)

Am Beispiel des ACC werden die Zählweise und die Unterschiede durch Dominion im Folgenden im Detail erklärt. Auf die Erklärung aller Funktionen kann verzichtet werden, da die Vorgehensweise jeweils sehr ähnlich ist.

Das ACC System benötigt Eingabedaten von 8 verschiedenen Systemen, wobei drei verschiedene CAN-Bus Systeme genutzt werden, nämlich der Comfort-CAN, der Antriebs-CAN und der System-CAN. Bei den Input Daten handelt es sich zunächst um die bereits unter 6.1.3 bei der Vorstellung des Systems genannten Werte, wie Gaspedal- und Bremspedalstellung, Lenkradwinkel, derzeitige und gewünschte Geschwindigkeit und der derzeitige und gewünschte Abstand zum vorausfahrenden Fahrzeug. Hinzu kommt der vom Fahrer gewünschte Modus des Systems, sprich in welchem Zustand das System arbeiten soll, ausgeschaltet, Cruise Control (CC) oder Advanced Cruise Control (ACC).

Quelle der Daten sind die unterschiedlichsten Systeme im Automobil, die über die entsprechenden Schnittstellen angesprochen werden müssen. Im hier vorliegenden Fall ist

dies jeweils ein CAN-Bus. Eine wichtige Voraussetzung für die Arbeit eines Entwicklers ist eine detaillierte Dokumentation der CAN-Bus Kommunikation, darüber hinaus ist eine intensive persönliche Diskussion mit dem jeweiligen Verantwortlichen und Experten notwendig. Durch die Nutzung von Dominion entfällt dieser Schritt weitgehend, da der Entwickler lediglich die einzubindenden Schnittstellen kennen muss. Der eigentliche Code wird später mit Hilfe der in Dominion gespeicherten Informationen automatisch generiert. So wird bspw. für die Daten über das vorausfahrende Fahrzeug eine Schnittstelle zum System "Traffic Front Vehicle" benötigt. Anstatt sich nun mit allen Details der bereits über CAN zur Verfügung stehenden Daten zu befassen und die notwendigen Inputs und Outputs zu implementieren, reicht es bei Verwendung von Dominion aus einen Link mit dem entsprechenden Namen der Schnittstelle anzugeben. Dieser verbleibende Restaufwand zum Erstellen einer Code Zeile wird hier vernachlässigt. Im Gegensatz zur klassischen FPA wird beim e-FACTS Modell eine gleichbleibende Produktivität angenommen. Die Unterschiede zwischen einer Realisierung der SWPL und einer Einzelentwicklung spiegeln sich explizit im Unterschied der Aufwände – gemessen in FPs – wider. Daher sind an dieser Stelle keine FPs für die automatische Generierung von In- und Outputs zu zählen. Die Möglichkeit dazu ergibt sich durch den Einsatz der SWPL, wodurch sich der Entwicklungsaufwand reduziert. Und genau dieser Effekt ist zu erfassen.

Neben In- und Outputs gelten die beschriebenen Vereinfachungen bei der Codeentwicklung ebenso für alle Schnittstellen. Auch hier bietet Dominion Unterstützung, indem die notwendigen Codebausteine für die Erstellung einer Schnittstelle zu einem anderen von Dominion abgedeckten System automatisch erstellt werden. Der ohne Dominion ansonsten notwendige Aufwand wird durch die zusätzlichen FPs für ein EIF bei Einzelentwicklung berücksichtigt. Als Output mit geringer Komplexität gibt ACC ein Datenpaket mit den entsprechenden Anpassungen für die Beschleunigung und Verzögerung des Fahrzeugs aus.

Erfolgt die Entwicklung basierend auf Dominion beträgt der Aufwand für alle Schnittstellen, In- und Outputs des ACC Systems 0 Feature Points. Es verbleibt lediglich der Aufwand zur Erstellung des Steuerungsalgorithmus, von dem in Abbildung 29 bereits ein Ausschnitt gezeigt worden ist. Der Algorithmus wurde in der Komplexität als einfach bewertet, da weniger als 19 Parameter und Variablen einbezogen und weniger als 16 Regeln angewendet werden. Zusätzlich enthält das System ein Data File, das den aktuellen Modus des Systems

und die aktuellen Ausgabewerte speichert. Daher handelt es sich hier ebenfalls um eine geringe Komplexität.

6.1.5. Hochrechnung auf das Gesamtsystem

Dominion als Einheit ist zu umfangreich, um komplett im Rahmen dieser Arbeit und insbesondere im Rahmen eines Workshops detailliert analysiert zu werden. Bemerkenswert ist jedoch, dass die tatsächliche Anwendung des Modells nach erfolgter Einführung und ein paar Anwendungen deutlich beschleunigt werden konnte, was einen Rückschluss auf die Anwendbarkeit des Modells in zukünftigen Analysen zulässt. Es zeigte sich, dass die Analyse einer Funktion, inklusive zweifacher FP-Zählung, von anfangs ca. 30 Minuten auf ca. 5 Minuten reduziert werden konnte. Darauf aufbauend wäre es möglich, die restlichen Funktionen von Dominion innerhalb eines weiteren Tages zu komplettieren.

Um eine valide Aussage über die ökonomischen Folgen der vollständigen Dominion Entwicklung treffen zu können, wurden die Ergebnisse der sieben untersuchten Funktionen auf die Gesamtheit aller seit Einführung von Dominion entwickelten Funktionen hochgerechnet. Dazu wurde in einem ersten Schritt die durchschnittliche Produktivität unter Einsatz von Dominion berechnet. Produktivität wird hier als die von einem Entwickler in einem Monat realisierbare Funktionalität, gemessen in FP, verstanden. Dabei beinhaltet realisierte Funktionalität den gesamten Entwicklungsprozess von der ersten groben Skizzierung bis zur anwendbaren Funktion, inklusive aller Tests. Die Berechnung ist in Abbildung 30 dargestellt.

	Entwicklungsaufwand in FPs			Aufwand	Produktivität
Anwendung	**DOMINION**	**Ohne DOMINION**	**Ersparnis**	PM	In FP/PM
• ACC	7	43	-84%	6	1,17
• Bewegungssteuerung	39	69	-43%	18	2,17
• Fahrdynamikmodell	14	35	-60%	NA	NA
• SeSiLa	23	44	-48%	0,75	30,67
• Displaykonzept	14	53	-74%	1	14,00
• Nebenaufgaben	7	26	-73%	0,5	14,00
• Cabin	33	63	-48%	2	16,50
Summe / Ø	137	333	-41%		4,35

Abbildung 30 – Übersicht der analysierten Funktionen auf Basis von Dominion[94]

Auffällig ist die hohe Bandbreite der gemessenen Produktivität. Diese reicht von 1,17 FP/PM für die Anwendung ACC bis zu einer Produktivität von knapp über 30 FP/PM für den Laserscanner. Während eine Produktivität etwas über 1 FP/PM für eine sicherheitskritische Anwendung im unteren Bereich der beobachteten Erfahrungswerte liegt, stellt die Produktivität des Lasersystems einen mit Abstand neuen Höchstwert dar. Die im Rahmen eines McKinsey Projektes gesammelten Erfahrungen deuten darauf hin, dass ein solcher Wert lediglich aufgrund besonderer Rahmenbedingungen möglich ist [vgl. McKi06]. Daher sollten an diesem Punkt der Analyse die Ergebnisse genau auf Plausibilität überprüft werden. Im hier vorliegenden Fall ergibt sich, dass der entsprechende Entwickler bereits über ein umfangreiches Expertenwissen hinsichtlich Sensorsysteme im Fahrzeug verfügte. Außerdem musste die hier betrachtete Anwendung nicht bis zur Marktreife entwickelt werden. Es wurde lediglich eine Emulation der Werte eines echten Systems für die Simulatorumgebung benötigt, wodurch der Testaufwand deutlich reduziert werden konnte. Insgesamt führen all diese Punkte zu der ungewöhnlich hohen Produktivität.

Solche Extremwerte sind bei der Anwendung der FP-Methodik zur Bestimmung des Entwicklungsaufwands durchaus möglich. Sie treten insbesondere bei kleinen Teilprojekten auf, wobei die verschiedensten Erklärungsmöglichkeiten greifen. Es ist nicht davon auszugehen, dass eine solch hohe Produktivität langfristig zu stabilisieren ist. Denn es ist unwahrscheinlich, dass die entsprechenden Sondereinflüsse bei jedem Projekt erneut

[94] Aufwand und Produktivität jeweils bei Realisierung mit der DOMINION Middleware

zutreffen. Für die Hochrechnung auf das Gesamtsystem ergibt sich, dass mit Durchschnittswerten gearbeitet werden sollte. Dadurch besteht die Möglichkeit einen möglichst realistischen Produktivitätswert für die Bewertung der restlichen Dominion Funktionen zu erhalten. Eine notwendige Bedingung ist, dass die Auswahl der Projekte nicht mit dem Ziel erfolgt, eine möglichst hohe Produktivität zu zeigen. Im Rahmen eines Workshops kann dies durch die Auswahl der Projekte zu einem Zeitpunkt, an dem die Teilnehmer die Methodik nicht vollständig kennen, verhindert werden.

Für das TS und Dominion ergibt sich selbst bei Nichtberücksichtigung des Extremwertes von über 30 FP/PM beim Produktivitätsunterschied immer noch ein Faktor von 14 (siehe Abbildung 31). Dies kann u.a. durch die thematisch unterschiedlichen Anwendungen erklärt werden. Es ist bspw. anzunehmen, dass die Produktivitäten von Anwendungen aus dem Entertainment Bereich grundsätzlich höher liegen, als solche aus sicherheitskritischen Bereichen. Allerdings zeigen Erfahrungen, dass solche Abweichungen eher im Bereich Faktor 2 bis 3 liegen. Der deutlich höhere Faktor bei Dominion hängt stärker mit der Arbeitsweise des TS und Dominion zusammen. Im Gegensatz zu großen Unternehmen werden hier Anwendungen stärker von einzelnen Personen, z.B. im Rahmen von Forschungsarbeiten, entwickelt als von großen Teams. Dadurch spielt natürlich die persönliche Erfahrung des einzelnen eine wesentlich größere Rolle. Durch die Verwendung von Durchschnittswerten kann dieser Effekt zumindest teilweise ausgeglichen werden.

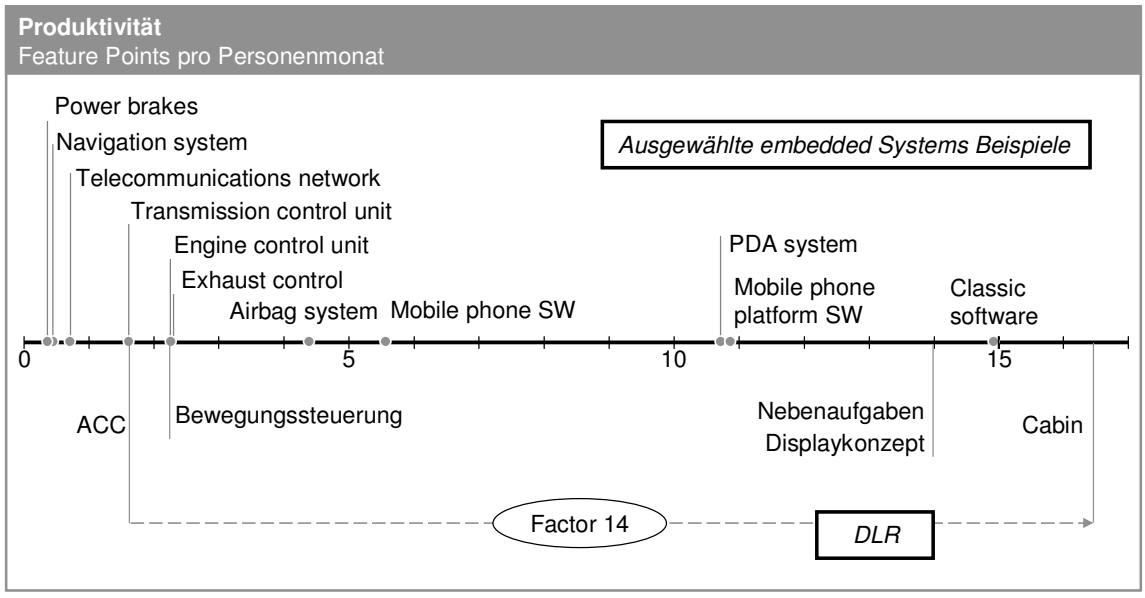

Abbildung 31 – Übersicht der Produktivitätsverteilung [vgl. auch McKi06, S. 25]

Im hier vorliegenden Fall wurde eine durchschnittliche Produktivität von 4,4 FP/PM erzielt. Für die Entwicklung von Embedded Systems im Automobil stellt dies einen überdurchschnittlich hohen Wert dar. In Abbildung 32 wird der Wert mit vorhandenen Daten aus früheren Untersuchungen verglichen.

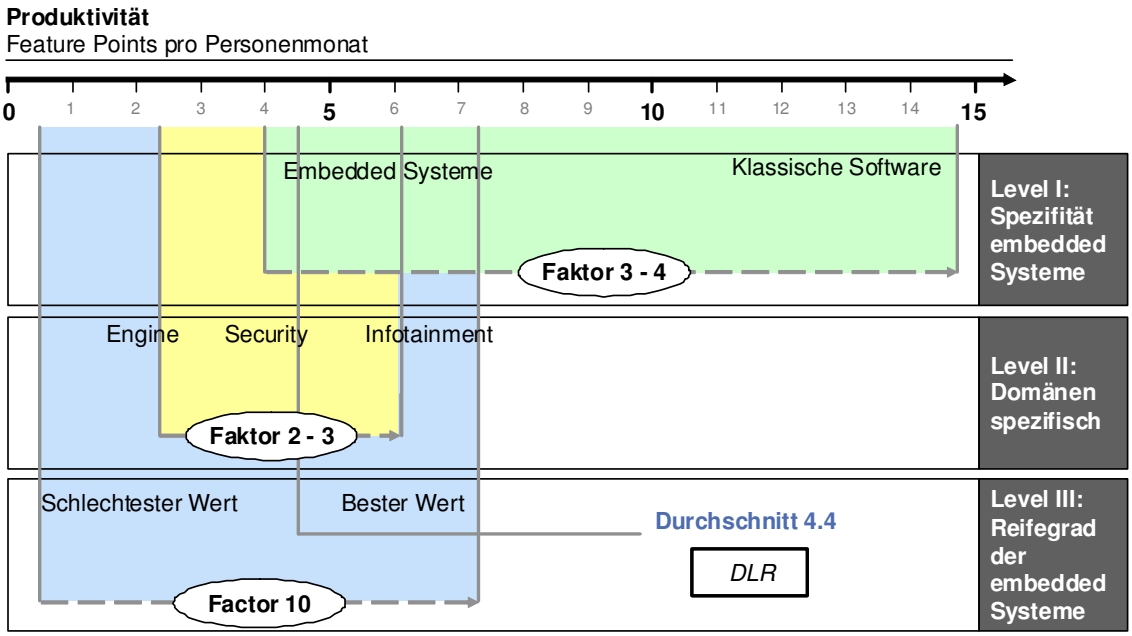

Abbildung 32 – Übersicht der Produktivitätsverteilung nach Anwendungsklassen [vgl. auch McKi06, S. 25]

Eine Produktivität von über 4 FP/PM liegt über dem beobachteten Durchschnitt für Embedded Systems, insbesondere da es sich hierbei auch um sicherheitskritische Systeme handelt. Eine deutlich darüber hinausgehende Produktivität ist eher typisch für den Bereich der klassischen Softwareentwicklung, da hier meist weitgehender auf Wiederverwendung ausgelegte Architekturen, Middleware und Entwicklungstools eingesetzt werden.[95] Bei der Betrachtung der Produktivität sollten drei weitere wichtige Punkte berücksichtigt werden, die einen Einfluss auf die Produktivität des DLR haben:

- Einerseits stellt das DLR keine marktreifen Produkte her. Natürlich wird die Software umfangreichen Tests unterzogen, nicht zuletzt da diese Software in Forschungsfahrzeugen genutzt wird. Aber dennoch ergäbe sich ein zusätzlicher Testaufwand, um die hier betrachteten Produkte tatsächlich am Markt absetzen zu können. Dieser Zusatzaufwand würde die Produktivität senken.

[95] Capers Jones bezeichnet eine Produktivität von 5 bis 10 Function Points/PM als die normale Bandbreite für SW-Projekte in den USA. Einzelne Projekte können aber auch bei über 20 Function Point/PM liegen [Jone07, S. 268].

- Andererseits erfolgt die Entwicklung eher forschungsorientiert als produktorientiert. Es stehen stärker Machbarkeitsstudien als reale Produkte im Vordergrund. Solche Vorhaben sind grundsätzlich mit einem höheren Entwicklungsaufwand verbunden, da in vielen Fällen zunächst einmal Grundlagenforschung betrieben werden muss. Daher ist anzunehmen, dass die Produktivität etwas niedriger als bei einem marktorientierten Unternehmen ausfallen wird.
- Und schließlich wird die Produktivitätsbetrachtung an dieser Stelle aufgrund des e-FACTS Modells leicht verfälscht, wodurch externe Vergleiche erschwert werden. Der Grund dafür ist, dass der tatsächlich realisierte Funktionsumfang – gemessen in FPs – höher liegt als der hier zugrundegelegte Wert. Schließlich werden die Schnittstellen, die Mittels DOMINION fast ohne zusätzlichen Entwicklungsaufwand implementierbar sind, nicht mit entsprechenden FPs berücksichtigt, um eine konstante Produktivität vor und nach Einführung des SWPL zu ermöglichen. Die tatsächliche Produktivität liegt demnach höher.

Für die Anwendung des hier vorgestellten Modells spielt diese Unterscheidung jedoch keine Rolle, da im weiteren Verlauf kein Vergleich mit anderen Unternehmen oder Benchmarks erfolgen soll, sondern lediglich ein unternehmensinterner Vergleich vorgenommen wird. Diese interne Betrachtungsweise ermöglicht es, die vorher genannten Besonderheiten eines jeden Unternehmens zu vernachlässigen und dennoch die Frage hinsichtlich der ökonomischen Folgen individuell beantworten zu können.

Dies bedeutet allerdings nur, dass man diese Effekte derzeit bei der Berechnung vernachlässigen kann. Sollte es für ein Unternehmen möglich sein, hier Veränderungen vorzunehmen (bspw. könnte sich das DLR entscheiden, in Zukunft marktreife Produkte anzubieten), ist im Einzelfall zu entscheiden, ob und wie diese Entscheidung in die Berechnung mit einbezogen werden kann. Wichtig ist, dass eine solche Veränderung zeitlich nicht zwischen dem Referenzprojekt und dem SWPL Projekt durchgeführt wurde. Ansonsten werden die sich daraus ergebenden Effekte mit der Wirkung der Einführung einer SWPL vermischt. Im Rahmen von DOMINION war dies nicht der Fall. Die Veränderungen waren vollständig auf die Einführung der SWPL zurückzuführen.

Die im Rahmen des eintägigen Workshops untersuchten 7 Funktionen bilden nur einen Teil der gesamten auf DOMINION basierten Entwicklungen ab. Seit der Einführung vor ca. 1 Jahr

sind bereits ungefähr 100 verschiedene Anwendungen entwickelt worden. Die 7 untersuchten Funktionen sind dabei etwas umfangreicher als der Durchschnitt. Es wurde geschätzt, dass der Anteil dieser Auswahl an dem Gesamtentwicklungsaufwand ca. 10 Prozent betragen hat. Legt man nun die Anzahl an FPs für die 7 betrachteten Anwendungen jeweils mit und ohne Dominion zu Grunde und rechnet diese auf alle dominionbasierten Anwendungen hoch, ergibt sich eine Einsparung von fast 2.000 FPs. Unter Berücksichtigung einer durchschnittlichen Produktivität von 4,4 FP/PM[96] und durchschnittlichen Personalkosten von 12.000 Euro pro Monat ergibt sich eine erzielte Ersparnis durch Dominion von mehr als 5,3 Mio. Euro. Die Kosten für die Erstellung von Dominion, besser gesagt, dem hier relevanten Teil von Dominion, betrugen dabei lediglich 216.000 Euro. Demnach hat sich die Investition in Dominion in nur einem Jahr nach Einführung bereist um den Faktor 24 amortisiert. Eine genauere Abschätzung über die gesamte Laufzeit ist allerdings nicht möglich, da bei der Entwicklung von Dominion keine entsprechende Planung erstellt worden ist. Die Details der Berechnung können der Abbildung 33 entnommen werden.

• Summe FPs für die 7 bewerteten Anwendungen auf Basis von DOMINION	137 FPs	
• Hochrechnung: Dies sind ca. 10% des Aufwands aller DOMINION Anwendungen		1.370 FPs
• Summe FPs ohne DOMINION	333 FPs	
• Hochrechnung Aufwand ohne DOMINION		3.330 FPs
• Ersparnis (Differenz mit/ohne DOMINION)		1.960 FPs
• Durchschnittliche Produktivität DLR		4,4 FP/PM
• PMs die durch DOMINION eingespart werden konnten		445 PM
• Aufwand zur Erstellung von DOMINION		18 PM
• Personalkosten /Person /Monat	12.000 Euro	
• Kosten zur Erstellung von DOMINION		**216.000 Euro**
• Ersparnis durch DOMINION nach 1 Jahr		**5.345.000 Euro**

Abbildung 33 – Berechnung der ökonomischen Auswirkungen durch Dominion

6.1.6. Sensitivitätsanalyse

Zur Berechnung der hier vorgestellten Ergebnisse mit dem e-FACTS Modell wurden einige Annahmen getroffen, Schätzungen vorgenommen und ein einfaches mathematisches Modell

[96] Die Veränderung des Entwicklungsaufwands durch Dominion wurde durch eine Verringerung der FPs berücksichtigt. Daher ist die Annahme einer konstanten Produktivität, sowohl mit als auch ohne Dominion, zulässig.

unter Unsicherheit erstellt. Auch wenn das Ergebnis eindeutig auf eine Vorteilhaftigkeit des DOMINION Ansatzes hinweist, handelt es sich hierbei allein um das Ergebnis für eine konkrete Ausgestaltung der gewählten Modellparameter, die nicht mit Sicherheit vorhergesagt werden können. Auch wenn im hier vorliegenden Fall aufgrund der a posteriori Betrachtung vielfach die tatsächlichen Werte verwendet werden können, sind einige Positionen dennoch zu bestimmen, z.B. der Aufwand in FPs oder der Anteil der untersuchten Funktionen am Gesamtentwicklungsumfang.

Ziel der wissenschaftlichen Erarbeitung einer Entscheidungsunterstützung sollte es sein, die in den Annahmen enthaltene Unsicherheit weitgehend zu reduzieren und damit die Stabilität der getroffenen Aussage zu erhöhen. Daher wird im Folgenden eine Sensitivitätsanalyse durchgeführt. An dieser Stelle, ebenso wie für das nächste Anwendungsbeispiel, wird die Sensitivitätsanalyse zunächst für konkrete Fallbeispiele herangezogen. Im Anschluss erfolgt dann in Kapitel 6.3 eine grundsätzliche Betrachtung und Bewertung der Sensitivitätsanalyse.

Laut Saltelli [Salt08, S.1] ist die Sensitivitätsanalyse (SA) "the study of how uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input". Demnach analysiert eine SA die Einflussstärke einzelner Input-Parameter eines Modells auf das Ergebnis, insbesondere wird untersucht, welche Unsicherheitsquellen den größten Effekt auf das Ergebnis haben. Dazu werden einzelne oder mehrere Parameter absichtlich ausgelenkt und die resultierenden Auswirkungen auf das Ergebnis betrachtet [siehe auch Stahl05, S. 304 und 395]. Schließlich kann der maximal tolerierbare Fehler für einen Parameter bestimmt werden, bei dem die Empfehlung unverändert bleibt. Wenn diese Fehlertoleranz groß genug ist, dann kann mit großer Wahrscheinlichkeit davon ausgegangen werden, dass selbst bei einer ursprünglich zu optimistischen Schätzung des Parameters die finale Empfehlung nach Korrektur des Fehlers durch die Messung der tatsächlichen Werte unverändert bleibt. Insgesamt handelt es sich bei der SA daher um ein Werkzeug, mit dessen Hilfe die Qualität einer Modellierung überprüft und gesichert werden kann. Dies kommt insbesondere bei komplexen Modellen zum Tragen, die als Einheit nicht mehr zu kontrollieren sind.

Für den konkreten Anwendungsfall der DOMINION Middleware sind die folgenden Parameter für die SA von Interesse, da sie trotz der a posteriori Betrachtung nicht eindeutig bestimmbar waren:

- Anteil der 7 analysierten Funktionen am Gesamtumfang aller seit Einführung von DOMINION darauf aufbauenden Entwicklungen: Im Rahmen des Workshops wurde eine Schätzung des Umfangs vorgenommen. Aufgrund der durchschnittlich etwas höheren Komplexität der betrachteten Funktionen wurde geschätzt, dass der Anteil 10% betragen müsste. Ein höherer Anteil reduziert das Einsparpotenzial durch die Verwendung von DOMINION, da weniger zusätzliche Funktionen mit reduziertem Aufwand entwickelt worden wären.

- Höhe des Einspareffekts bei den nicht individuell bewerteten Funktionen: in der Berechnung wird angenommen, dass sich die ermittelten Einsparungen aus der Einführung der DOMINION Middleware auch auf die weiteren darauf aufbauenden Funktionen identisch übertragen lassen. Sollten diese Einspareffekte geringer ausfallen, würde sich die Vorteilhaftigkeit des SWPL Ansatzes verringern.

- Fehler in der FP basierten Aufwandsabschätzung: Die Methode der FPs zur Aufwandsbestimmung ist eine Schätzmethode. Im Allgemeinen kann aufgrund von Erfahrungswerten mit dieser Methodik von einer Genauigkeit von +/-10% ausgegangen werden [Jone07, S. 9, 66 und 68]. Sollte bei der im Rahmen des eintägigen Workshops durchgeführten Schätzung ein größerer Fehler aufgetreten sein, könnte dies Auswirkungen auf die Empfehlung haben. Im Rahmen einer SA soll daher ermittelt werden, wie groß der Einfluss dieser Größe auf die Gesamtempfehlung ist und wie groß der Fehler der FP Schätzung sein dürfte ohne die Empfehlung ändern zu müssen.

Normalerweise werden SA verwendet, um komplexe mathematische Modelle hinsichtlich der Auswirkungen einzelner Inputfaktoren zu analysieren, da eine direkte Abschätzung durch den Inputgeber nur schwer oder überhaupt nicht möglich ist. Diese allgemeine Betrachtung erfolgt in Kapitel 6.3. An dieser Stelle wird aufgrund der begrenzten Zahl von lediglich 3 Einflussfaktoren eine direkte Auswertung ohne den Weg über mathematische Berechnungen gewählt. Es wird ein Worst Case Szenario angenommen, also eine Situation, die nur mit einer sehr kleinen Wahrscheinlichkeit eintrifft und die ungünstigsten gerade noch als realistisch anzunehmenden Rahmenbedingungen für einen Erfolg der SWPL vorgibt. Dabei handelt es sich konkret um die folgenden Werte:

- Der Anteil am Gesamtumfang wird statt mit 10% nun mit 25% angenommen.

- Der Einspareffekt durch DOMINION wird bei den verbleibenden 75% lediglich mit einem Viertel der Höhe bei den untersuchten Funktionen angenommen.
- Es wird angenommen, dass der mittels FP-Schätzung ermittelte Mehraufwand für die Entwicklungen ohne DOMINION um 100% überschätzt wurde.[97]

Damit kann eine Neuberechnung der eingesparten FPs durch DOMINION erfolgen. Dazu werden die berechneten Werte um die oben angeführten neuen Rahmenbedingungen korrigiert. Die 196 FPs Mehraufwand[98] für die Realisierung der Inputs, Outputs und Schnittstellen sind demnach um 50% zu reduzieren, da der Entwicklungsaufwand und damit die mögliche Einsparung durch die Verwendung der FP-Methodik zu hoch eingeschätzt wurde.[99] Diese 196 FPs zeigen allerdings nur die mögliche Einsparung bei den 7 im Detail untersuchten Funktionen. Die Wirkung ist zusätzlich auf das Gesamtsystem hochzurechnen. Aufgrund der getroffenen Worst Case Annahmen stellen diese Funktionen nun einen größeren Anteil am Gesamtsystem, wodurch sich das Einsparpotenzial verringert. Der Grund für diese Annahme ist, dass der im Workshop geschätzte Anteil der 7 Funktionen von 10% des Gesamtaufwands aller auf Basis von DOMINION im ersten Jahr entwickelten Funktionen lediglich eine grobe Schätzung darstellt. Die angenommenen 25% stellen eine sehr konservative Sichtweise da, was aufgrund der Zielsetzung einer Erhärtung der positiven SWPL Empfehlung zu begrüßen ist.

Stellen die 7 bewerteten Funktionen nun 25% des Gesamtaufwands, dann ist es korrekt anzunehmen, dass zusätzlich zur bereits berücksichtigten Einsparung noch mal der 3-fache Wert (Faktor 3) zu berücksichtigen ist. Als eine weitere konservative Annahme wird berücksichtigt, dass die Einsparungen bei den 75% weiteren Funktionen lediglich ein Viertel der im Detail ermittelten Einsparungen betragen (Faktor 0,25). Insgesamt ergibt sich damit eine Ersparnis von 196 FPs*0,5 + 196 FPs*0,5*3*0,25=172 FPs.

---

[97] Dies stellt eine zunächst ungewöhnliche Annahme im Rahmen von SWPL da. Man würde erwarten, dass in einem Worst Case Szenario eine zu geringe Aufwandsschätzung vorgenommen worden ist, da die Entwicklung von wieder verwendbaren Komponenten einen Mehraufwand verursacht. Allerdings ermöglicht DOMINION hier die Einsparung von Entwicklungsaufwand, da Schnittstellen einfacher integriert werden können. Aus diesem Grund ist die Annahme für die Worst Case Betrachtung in diesem Fall umzudrehen.

[98] 196 FPs sind die Differenz zwischen den 333 FPs, die ohne einen Einsatz von Dominion für die Entwicklung der 7 betrachteten Funktionen notwendig gewesen wären, und den 137 FPs, die durch den Einsatz von Dominion tatsächlich als Entwicklungsaufwand angefallen sind.

[99] In Teil A Kapitel 5 wurde mehrfach auf die angestrebte und tatsächlich beobachtete Genauigkeit der Function Point und Feature Point Methode von +/- 10% bis +/-20% eingegangen. Diese Ungenauigkeit kann daher in beide Richtungen beobachtet werden.

Dagegen ist nun die notwendige Investition für DOMINION zu rechnen. Bei der zugrundegelegten durchschnittlichen Produktivität von 4,4 FP/PM und einem Entwicklungsaufwand von 18 PM ergibt sich eine Investition von 18 PM*4,4 FP/PM=79FPs. Damit rechnet sich die Investition in die SWPL selbst im Worst Case Szenario bereits nach nur einem Jahr.

Es kann festgehalten werden, dass im Fall der DOMINION Middleware selbst bei Eintreffen der ungünstigsten Bedingungen eine Umsetzung der SWPL vorteilhaft gegenüber einer Einzelentwicklung ist. Zusätzlich zu den getroffenen Annahmen wäre die SWPL sogar bei einer Verwendung von nur einem Jahr vorteilhaft, obwohl die Lebensdauer auf mehrere Jahre ausgelegt worden ist.

6.1.7. Abschließende Betrachtung

Die hier vorgestellten Ergebnisse des eintägigen Workshops beim Institut für Verkehrssystemtechnik des DLR in Braunschweig zeigen die universelle Anwendbarkeit des Modells. Obwohl Dominion als plattformübergreifende Middleware zur Entwicklung von Embedded Systems für Versuchsfahrzeuge und Fahrzeugsimulatoren nicht automatisch als Produktfamilie zu erkennen ist, war es ohne Anpassungen möglich, das e-FACTS Modell anzuwenden. Aufgrund der unternehmensindividuellen Betrachtung war es möglich, selbst sehr individuelle Rahmenbedingungen, wie z.B. die Forschungsorientierung und das Fehlen marktreifer Produkte, zu berücksichtigen.

Ein wichtiger Faktor ist die benötigte Zeit. Trotz einer vollständigen Einführung aller Workshopteilnehmer in die FP-Methodik und das e-FACTS Modell war es möglich einen beträchtlichen Teil der Entwicklungsarbeit eines Jahres an nur einem Tag zu analysieren. Gleichzeitig wurde der hohe Lerneffekt der Teilnehmer sichtbar. Die benötigte Zeit zur Analyse einer Anwendung konnte im Verlauf des Workshops um über 80 Prozent gesenkt werden. Die Analyse der weiteren auf Dominion basierenden Anwendungen hätte demnach ungefähr einen weiteren Tag erfordert. Bedenkt man die gesamte Projektgröße von über 100 PM ist der Aufwand von 2 Tagen zur Analyse mehr als vertretbar.

Hinsichtlich der Genauigkeit der Ergebnisse konnte für den konkreten Anwendungsfall gezeigt werden, dass die modellimmanenten Ungenauigkeiten (z.B. durch die Verwendung der FP Methode zur Aufwandsschätzung) aufgrund der Höhe der ermittelten Vorteile vernachlässigbar sind. Selbst bei Annahme eines Worst Case Szenarios wäre die Umsetzung

einer DOMINION Middleware und damit die Schaffung der Möglichkeit einer SWPL mit großem Abstand vorteilhaft.

Die Ergebnisse wurden im DLR intensiv und mit Interesse diskutiert. Die Methodik wurde insgesamt positiv bewertet. Insbesondere die Möglichkeit, im Nachhinein eine ökonomische Bewertung der Entwicklungsleistung durchführen zu können, wurde als hilfreich betrachtet. Daher erwägt das DLR die Methodik in Zukunft weiterhin selbständig anzuwenden.

Abschließend zeigt dies, dass ein eintägiger Workshop ausreicht, um einen fundierten ersten Eindruck über die Anwendbarkeit der Methodik im jeweiligen Unternehmen zu erlangen. Gleichzeitig kann das Unternehmen die notwendigen Schritte zur Analyse anschließend selbständig durchführen.

6.2. Motorsteuerungssoftware bei Siemens VDO

Bereits früh im Verlaufe der Erstellung der vorliegenden Arbeit bestand die Möglichkeit, das e-FACTS Modell bei Siemens VDO[100], einem führenden deutschen Automobilzulieferer, zu testen. Zum Zeitpunkt des Workshops im Jahre 2006 war das Modell in seiner hier vorgestellten Form nicht vollständig vorhanden, da der Fragenkatalog mit Basiskriterien fehlte (Kapitel 5.1). Dennoch war eine vollständige Anwendung des e-FACTS Modells möglich. Die Ergebnisse werden im weiteren Verlauf vorgestellt. Der Workshop fand statt im Rahmen einer Zusammenarbeit mit der Unternehmensberatung McKinsey & Company. Aufgrund von Bedenken hinsichtlich vertraulicher und wettbewerbsrelevanter Ergebnisse wurden die angeführten Produkt- und Komponentennamen und auch die Kunden geändert oder verallgemeinert. Die genannten Zahlen entsprechen jedoch den tatsächlich erfassten Daten.

6.2.1. Projektbeschreibung

VDO bietet Produkte für fast alle Bereiche des Automobils an. Zur Analyse wurde die Motorsteuerung für PKWs ausgewählt, da das Unternehmen in diesem Bereich 2005 eine große Architekturinvestition vornam, bei der eine weitgehende Wiederverwendung der einzelnen Komponenten im Fokus stand. Bisher stellte das Unternehmen für jeden Kundenauftrag ein eigenes Team zusammen, das jeweils eine komplette Motorsteuerung

[100] Bedingt durch den Verkauf von Siemens VDO von Siemens an Continental im Jahre 2007 erfolgte im November 2007 die Umbenennung in VDO Automotive AG. Im weiteren Verlauf der vorliegenden Arbeit wird daher der Name VDO verwendet.

(Hardware und Software) entwickelte. Natürlich wurde dabei das im Unternehmen vorhandene Wissen genutzt, ebenso einzelne Komponenten wiederverwendet. Dabei handelte es sich allein um opportunistische Wiederverwendung. Dies wurde durch die hohe Anzahl von über 800 Komponenten erschwert. Eine durchgehende Dokumentation war nicht vorhanden. Um die Entwicklungsgeschwindigkeit zu erhöhen, entschloss man sich, den hohen Anteil gleicher Komponenten bei Motorsteuerungen zu nutzen und eine SWPL zu entwickeln, ohne dabei bewusst von einer SWPL zu sprechen. Im Rahmen eines Projektes wurden alle Komponenten hinsichtlich ihrer Eignung für eine strategische Wiederverwendung untersucht. Der Fokus lag auf der reinen Realisierung, eine Betrachtung der ökonomischen Folgen und der zu erwartenden Kosten für die neue Architektur vor Beginn des Projektes erfolgte nicht. Grund dafür war die Annahme, dass aufgrund der zunehmenden Kundenanforderungen eine Variabilisierung der Produktentwicklung zwingend erforderlich sein würde, um die Anfragen zukünftig mit den bestehenden Kapazitäten wettbewerbsfähig bedienen zu können. Die Motivation ergab sich demnach eher aus einem wirtschaftlichen, externen Zwang als aus einer fundierten ökonomischen Analyse.[101]

Die Analyse der Motorsteuerung bot den Vorteil, dass hier ein Vergleich zwischen der alten Entwicklung einer Motorsteuerung für einen deutschen Automobilhersteller (DAH) und einer aktuellen Entwicklung basierend auf der neuen Architektur für einen südkoreanischen Hersteller (SKH) gezogen werden konnte. Die Modellanwendung erfolgte im Rahmen eines eintägigen Workshops und mehrerer Vor- und Nachbereitungstermine. Die limitierte Zeit machte es erforderlich, aus der Liste aller Funktionen einer Motorsteuerung eine Auswahl zu treffen. Abbildung 34 zeigt eine Auflistung der wichtigsten Funktionen einer Motorsteuerung. Die weitere Analyse umfasst die Komponenten Einspritzrealisierung (ESR) und die Kontrolle der Zusammensetzung der Gase im Auspuff (ZGA).

[101] Nach Aussage von Herrn Dr. Bocionek, Leiter Funktions- und Softwareentwicklung für Motorsteuerung bei VDO, gibt es im Bereich der Motorsteuerung alle 20 Jahre einen großen Innovationssprung und alle 10 Jahre eine wichtige Investitionsentscheidung, wie z.B. die hier besprochene Entscheidung eine SWPL einzuführen. Dabei besteht ein externer Druck, diese Investition durchzuführen, um wettbewerbsfähig zu bleiben.

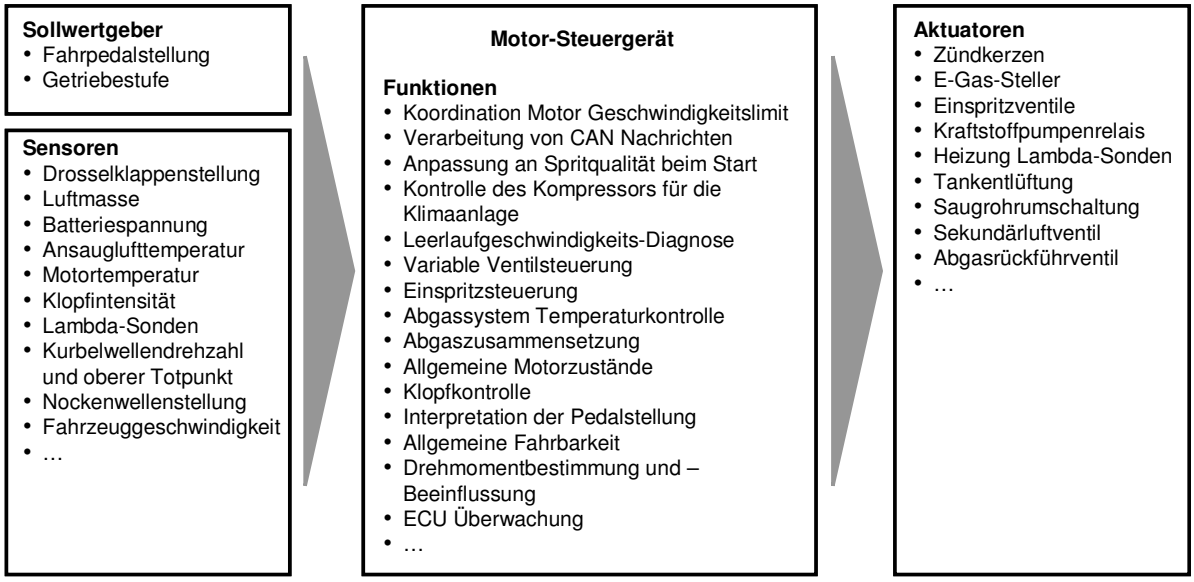

Abbildung 34 – Übersicht der Schnittstellen und Funktionen eines Motorsteuergeräts für Ottomotoren nach [Schä06, S. 9]

6.2.2. Komponente ZGA

Die Funktionsweise der ZGA Komponente ist in Abbildung 35 schematisch dargestellt. Die Komponente erhält verschiedene Inputparameter, wie z.B. die Temperaturen der Abgase an verschiedenen Stellen im System. Daraus werden Steuerungsparameter für die Benzineinspritzung berechnet, um den Schadstoffanteil im Abgas zu optimieren. Zusätzlich werden Diagnosewerte für die Lambda-Sonde und den Katalysator berechnet, welche anderen Anwendungen zur Verfügung gestellt werden.

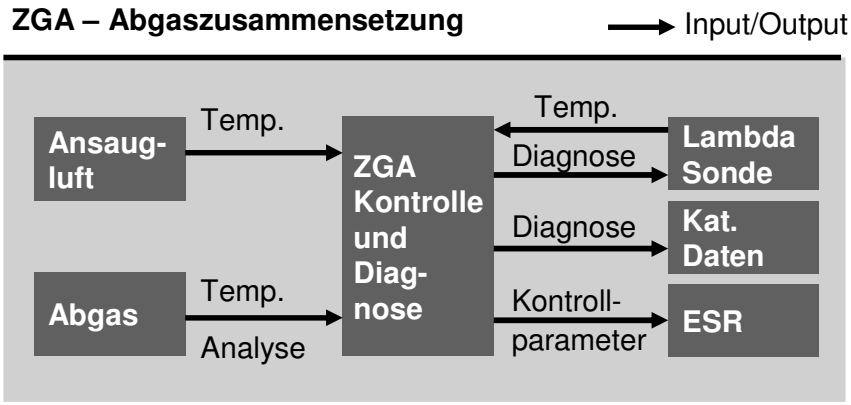

Abbildung 35 – Schematische Darstellung der ZGA Komponente

Sämtliche Funktionen der ZGA Komponente wurden mit FPs bewertet. Aus Zeitgründen wurde diese Zählung allein für die SKH-Komponente auf Basis der neuen SWPL Architektur durchgeführt, mit dem Ergebnis von 176 FPs. Tabelle 47 zeigt die Berechnung. Die Anzahl

der Total FP unadjusted ergibt sich durch Multiplikation der Anzahl der nach Komplexität getrennten Funktionstypen und den in Tabelle 3 (siehe Teil A Kapitel 5.5) genannten Komplexitätsgewichten. Zu den Details der Berechnung sei auf Kapitel 5.4.1 verwiesen.

		EI			EO			EQ			AL			DF			EIF		
Nr.	Feature	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
																Total FP unadjusted		=	176
1	Lambda Temperatur Messung und Kontrolle	1										1					1		
2	Lambda Messung und Kontrolle	4					1				2	1	1			1	3		
3	Lambda Heizung und Kontrolle	1			1						1	1		1			2		
4	Ansaugluft	1															1		
5	Abgastemperaturmessung	1															1		
6	Diagnose Katalysator					1					1								
7	Fehlerspeicher					1							1			1			
8	Konfiguration			1												1			
9	ICP und IP Generierung				2						1	1					2		

Tabelle 47 – FP Zählung ZGA als Asset Base (vollständig ausgefülltes Worksheet auf Basis Tabelle 29)

Die Funktionen sind weitgehend selbsterklärend, bis auf die letzte Funktion ICP und IP Generierung. IP bezeichnet einen Pumpstrom in der Lambda-Sonde, mit dem der Sauerstoffgehalt in der Messkammer geregelt wird. ICP ist eine Input-Capture Funktion. Diese wird zum Aufzeichnen und zur exakten Zeitmessung bei einer hohen Zahl an Messwerten innerhalb sehr kurzer Zeit verwendet.

Die fehlende FP-Bewertung für die ältere DAH-Komponente als Einzelentwicklung kann mit Hilfe der anderen verfügbaren Daten ausgeglichen werden. Dazu nimmt man an, dass die Produktivität für beide Projekte vergleichbar ist. Dies ist bei der Anwendung der FP Methodik im Rahmen des e-FACTS Modells dadurch zu begründen, dass der zusätzliche Aufwand zur Erstellung der wieder verwendbaren Komponente mittels zusätzlicher oder komplexerer Funktionstypen erfasst wird und sich eben nicht, wie sonst anzunehmen in einer niedrigeren Produktivität ausdrückt. Um die FPs für die Einzelentwicklung berechnen zu können werden die jeweiligen Aufwände in PM benötigt. Für SKH wurden 74 PM benötigt, wobei davon 13% auf die Neuausrichtung der Organisation entfielen. Demnach bleiben 64 PM für die effektive Funktionsentwicklung. Daraus ergibt sich mittels der Formel

Gleichung 16 – Berechnung der Produktivität

$$P = \frac{FP}{PM}$$

eine Produktivität von 2,73 FP/PM. Diese Produktivität wird nun auf das DAH Projekt übertragen. Der dort entstandene Aufwand beläuft sich auf 30 PM, wobei zu beachten ist, dass SKH 43% zusätzliche Funktionen enthält.[102] Der funktionsbereinigte Aufwand beträgt demnach 43 PM. Durch eine Multiplikation der Produktivität mit dem zeitlichen Aufwand lässt sich die Anzahl der FPs bestimmen, diese beträgt für DAH 117 FPs. Ein Vergleich der FP-Werte für beide Komponenten ergibt einen Mehraufwand für die Variabilisierung von 1,5. Das Unternehmen konnte die Funktionen soweit standardisieren und variabilisieren, dass eine Wiederverwendung dieser Komponente in einem neuen Produkt lediglich zu einem Aufwand von 1,5 PM führt. Damit wird der Break Even Point bereits mit der zweiten Verwendung erreicht. Das Unternehmen hat bei der Planung der SWPL mit 5 Verwendungen der Komponente gerechnet. Dies ergibt eine Einsparung von 135 PM oder über 60% gegenüber der Einzelentwicklung. Allerdings ist dabei die Investition in die SWPL Architektur nicht berücksichtigt.

Der Anteil der ZGA Komponente an der gesamten Funktionalität einer Motorsteuerung beträgt ungefähr 10%. D.h. selbst nach Einbeziehung der anteiligen Architekturentwicklungskosten rechnet sich die Variabilisierung der Komponente nicht um einige Prozent, sondern um Faktoren. Mit dieser Gewissheit können die Ungenauigkeiten der e-FACTS Methodik vernachlässigt werden. Weitere Details hierzu finden sich in der Sensitivitätsanalyse in Kapitel 6.2.4.

6.2.3. Komponente ESR

Die Funktionsweise der ESR Komponente ist in Abbildung 36 schematisch dargestellt. Die Komponente erhält verschiedene Inputparameter, wie Daten zur Güte des Kraftstoffs oder die Ergebnisse der Abgasanalyse der ZGA Komponente. Mittels dieser Daten wird die optimale Kraftstoffmenge bestimmt und die eigentliche Injektionsphase kontrolliert und bei Bedarf angepasst.

---

[102] Die Steuerung der Abgaszusammensetzung unterliegt starken Veränderungen, bspw. erfordern neue Abgasvorschriften wie die zum 1. September 2009 in Kraft getretene Euro5-Norm zusätzliche Funktionen, um die strengeren Vorgaben erfüllen zu können.

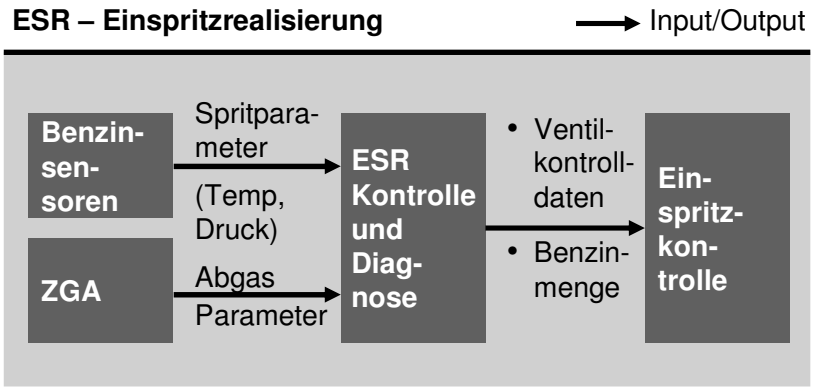

Abbildung 36 - Schematische Darstellung der ESR Komponente

Sämtliche Funktionen der ESR Komponente wurden mit FPs bewertet. Erneut musste aus Zeitgründen auf eine individuelle Zählung der DAH Komponente verzichtet werden. In Tabelle 48 ist das detaillierte Ergebnis der Zählung für das SKH Projekt dargestellt, mit einem Ergebnis von 151 FPs.

		EI			EO			EQ			AL			DF			EIF		
Nr.	Feature	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
																Total FP unadjusted		=	151
1	Einspritzmassen-Kontrolle	1			1						2	1		1			5		
2	Einspritzphasen-Kontrolle	1			1						1			1			4		
3	Einspritzventil-Kontrolle	1			1							1			1		4		
4	Ventil Diagnose	1			1						3			1				1	
5	Kalibrierung		1												1				

Tabelle 48 – FP Zählung ESR als Asset Base (vollständig ausgefülltes Worksheet auf Basis Tabelle 29)

Die Anzahl FPs für die ESR Komponente im DAH Projekt wird wie im vorherigen Kapitel abgeleitet. Die 151 FPs wurden mit 38 PM realisiert, dabei entfielen 7% auf die notwendigen organisatorischen Anpassungen. Mit den verbleibenden 35,3 PM ergibt sich eine Produktivität von 4,27 FP/PM. Es zeigen sich schon hier erhebliche Unterschiede gegenüber der ZGA Komponente. Dies unterstreicht die Bedeutung einer komponentenindividuellen Betrachtung.

Für die Erstellung der ESR Komponente im DAH Projekt ohne SWPL wurden 12 PM benötigt. Aufgrund der höheren Stabilität der Komponente haben sich hier keine signifikanten Unterschiede in der abgedeckten Funktionalität gezeigt. Daher ergibt sich ein Wert von 51,3 FPs. Der Faktor für Mehraufwand zur Erstellung wiederverwendbarer SW liegt bei 2,9. Der Aufwand für die Anpassung an ein neues Produkt ist erstaunlich gering und liegt bei lediglich 0,3 PM. Ein wichtiger Grund für diesen Wert ist die hohe Stabilität der Komponente. Die 0,3

PM entstehen fast ausschließlich durch Integrationstests. Daraus ergibt sich ein Break Even Point von fast genau 3 Verwendungen. Dies ist erneut ein erheblicher Unterschied gegenüber der ZGA-Komponente. Die Verwendung eines einheitlichen Wertes für diesen Mehraufwand hätte diese Unterschiede vernachlässigt.

Aufgrund des hohen Mehraufwands zur Integration der Komponente in die Asset Base könnte sich diese Komponente als Einzelentwicklung anbieten. Die Entscheidung wird allerdings maßgeblich durch die Anzahl der geplanten Verwendungen bestimmt. Auch hier wurde von 5 Anwendungen ausgegangen. Daher ermöglicht die Berücksichtigung als Asset eine Einsparung über die Laufzeit der SWPL von 217 PM oder fast 85%.

6.2.4. Sensitivitätsanalyse

Wie schon beim ersten Anwendungsbeispiel, der DOMINION Middleware, in Kapitel 6.1 soll auch für die untersuchte Motorsteuerungssoftware die Stabilität der Modellempfehlung mittels Sensitivitätsanalyse (SA) getestet werden. In diesem Fall werden 3 Eigenschaften, für die Werte im Rahmen der Berechnung angenommen worden sind, ausgelenkt und die Wirkung ermittelt. Bei diesen Werten wird eine große Wirkung auf das Gesamtergebnis angenommen. Im Folgenden wird allein auf die Veränderung dieser 3 Eigenschaften abgestellt und ein Worst Case Szenario angenommen:

- **Anzahl der Wiederverwendungen:** Es werden entweder nur eine einmalige Wiederverwendung angenommen (ZGA) oder 2 Wiederverwendungen (ESR).
- **Aufwand für Wiederverwendung:** Der angenommene Aufwand für die tatsächliche Wiederverwendung wird mit Faktor 2 bewertet
- **FP basierte Aufwandsschätzung:** Es wird angenommen, dass der mittels FP Schätzung bewertete Aufwand unterschätzt wurde. Dieser wird nun verdoppelt, was einer Korrektur von 100% entspricht.

Bei der ZGA Komponente ergibt sich bei Entwicklung einer SWPL und einer einmaligen Wiederverwendung ein Gesamtaufwand im Worst Case Szenario von 360 FPs. Dieser Aufwand setzt sich zusammen aus den 176 FPs für die Entwicklung der Basis der SWPL. Dieser Wert ist zu verdoppeln, da angenommen wurde, dass die FP-Schätzung den Aufwand um 100% zu niedrig bewertet hat. Zusätzlich ist der Aufwand für die tatsächliche Wiederverwendung von ursprünglich 1,5 PM zu verdoppeln und mittels der ermittelten

Produktivität in FPs umzurechnen. Als Ergebnis ermittelt man: 2*176FP+3PM*2,73FP/PM = 360 FPs.

Bei einer Einzelentwicklung beträgt der Umfang 468 FPs. Eine Einzelentwicklung wurde bei der Analyse mit 117 FPs bewertet. Dieser Wert ist aufgrund der angenommenen Unterschätzung des Aufwands zu verdoppeln. Zusätzlich fällt dieser Aufwand für die beiden zu entwickelnden Produkte (einmalige Wiederverwendung) an. Daraus ergibt sich der gesamte Aufwand von 2*117FP*2=468 FPs.

Für die ESR Komponente ist aufgrund des höheren Faktors für den Mehraufwand zur Herstellung wiederverwendbarer Software eine einmalige Wiederverwendung nicht ausreichend. Erst nach der zweiten Wiederverwendung rentiert sich die Investition. Unter den genannten Worst Case Annahmen (hier mit 2 Wiederverwendungen) ergeben sich die folgenden Umfänge, wobei die Berechnung analog zum oben beschriebenen Vorgehen erfolgt: 2*151 FP+0,6 PM*4,27 FP/PM*2=307 FPs für die SWPL und 2*51,3 FP*3=308 FP für die Einzelentwicklungen. Der Faktor 3 bei der Berechnung der Einzelentwicklungen ergibt sich durch die 2 Wiederverwendungen. Damit werden insgesamt 3 Produkte entwickelt.

Mittels der SA wurde der Punkt des maximalen Fehlers ermittelt. Sollte die angenommene Situation tatsächlich eintreffen, wären die beiden Entwicklungsansätze in diesem konkreten Anwendungsfall beinahe gleichwertig. Geht man nun davon aus, dass die für diese Berechnung geltenden Annahmen tatsächlich eine sichere Grenze darstellen, die nicht unterschritten werden wird, sondern sich mit großer Wahrscheinlichkeit positiver aus Sicht einer SWPL entwickeln wird, kann die Empfehlung zur Umsetzung einer SWPL als gesichert angesehen werden.

In beiden Fällen wurde ein Fehler der FP Schätzung von 100% angenommen. Dieser ist deutlich höher als alle Referenzwerte in der Literatur (vgl. Teil A Kapitel 5.5). Zusätzlich wurden auch die anderen unsicheren Parameter in Richtung eines Worst Case Szenarios ausgelenkt. Die Empfehlung blieb dabei stabil, hauptsächlich da die Empfehlung nicht auf einer leichten Vorteilhaftigkeit beruhte, sondern Unterschiede um einige Faktoren zwischen den beiden Lösungsalternativen bestehen.

6.2.5. Abschließende Betrachtung

Die Analyse der beiden Komponenten ZGA und ESR hinsichtlich ihres Wiederverwendungspotenzials ergab eine eindeutig positive Bewertung des SWPL Ansatzes. Die Gesamtkosten über die Laufzeit der geplanten SWPL liegen um Faktor 2,5 bzw. 6 niedriger als die entsprechenden Kosten bei Einzelentwicklung. Auffallend sind die großen Unterschiede zwischen den beiden Komponenten bei fast allen relevanten Daten. So ist das Einsparpotenzial für die ESR Komponente über die gesamte Laufzeit deutlich größer, obwohl der Faktor für die Mehrkosten für Wiederverwendbarkeit beinahe doppelt so hoch wie bei der ZGA Komponente liegt. Die Gründe dafür sind die niedrigen Aufwände für eine Anpassung an ein neues Produkt, die generell höhere Produktivität und der niedrigere Anteil der organisatorischen Kosten bei der Erstellung der Komponente als Asset. Dies zeigt die Vorteile des e-FACTS Ansatzes, der eine komponentenweise Erfassung und Auswertung der Daten bietet. Erst dadurch wird eine ökonomische Optimierung der Asset Auswahl ermöglicht, die als zusätzliche Information in das SWPL Scoping einfließen kann.

Eine Hochrechnung auf das Gesamtsystem beim VDO Beispiel erfolgte nicht, da der Anteil der ESR Komponente am Gesamtsystem nicht definiert ist. Aufgrund der großen Unterschiede zwischen den beiden analysierten Komponenten erscheint eine Hochrechnung allein auf Basis der ZGA Daten wenig sinnvoll. Generell deuten die Unterschiede auf die Notwendigkeit einer kompletten Komponentenbewertung hin.

Da die Einführung der SWPL, das DAH Projekt und einige weitere Motorsteuerungsprojekte auf Basis der neuen Architektur zum Analysezeitpunkt bei VDO bereits abgeschlossen waren, können generelle Aussagen über die erzielten Ergebnisse getroffen werden. Der Aufwand zur Erstellung der Software für eine an Kundenwünsche angepasste Motorsteuerung konnte durch die Einführung der SWPL ca. auf die Hälfte reduziert werden. Diese freie Kapazität wurde vollständig für neue und zusätzliche Funktionen genutzt, wie z.B. eine elektronische Ventilsteuerung. Der Gesamtaufwand zur Softwareerstellung blieb trotz geänderter Architektur im Durchschnitt je Projekt unverändert. Dies ist u.a. dadurch begründet, dass die OEMs den tatsächlichen Aufwand zur Erstellung der Motorsteuerung zahlen und nicht etwa einen fixen Preis für ein fertiges Produkt.[103]

---

[103] Laut Herrn Dr. Bocionek betragen die Gesamtkosten für die SW-Entwicklung einer Motorsteuerung ca. 2,5 Mio EUR.

Die komplette Entwicklungszeit einer neuen Motorsteuerung liegt zwischen 1,5 Jahren für eine einfache Variante (wie z.B. das SKH Modell) und 2 bis 2,5 Jahren für aufwändigere Ausgestaltungen (wie z.B. die DAH Variante).[104] Zwar könnte die reine SW-Entwicklungszeit aufgrund der Einführung der SWPL deutlich verkürzt werden, aber aufgrund des höheren Funktionsumfangs werden die erzielten Vorteile kompensiert. Zusätzlich ändern sich die Requirements im Laufe der Entwicklung typischerweise so häufig, dass eine Synchronisation der SW- mit der HW-Entwicklung notwendig ist.

Wie bereits oben erwähnt handelte es sich bei der Einführung der SWPL nicht um eine freie Entscheidung seitens VDO. Vielmehr wurde dies als notwendig angesehen, um auch zukünftig eine wettbewerbsfähige Position einnehmen zu können. Dennoch sollten die bei der Umsetzung der SWPL aufgetretenen Herausforderungen diskutiert werden. Besondere Probleme ergaben sich durch die hohe Auslastung der Entwicklungsabteilung zur Erstellung der neuen SWPL Architektur. Diese musste in die normale Entwicklung neuer Produkte integriert werden. Ein Ergebnis ist die lange Entwicklungszeit von 2 Jahren. Außerdem macht diese gleichzeitige Entwicklung eine Bestimmung des Aufwands zur Erstellung der SWPL-Architektur unmöglich. Trotz der Fokussierung auf die Fertigstellung neuer Produkte mussten einige Termine verschoben werden. Die daraus resultierenden Kosten, z.B. aus entgangenem Umsatz oder möglicher Strafzahlungen an OEMs, wurden nicht ermittelt. Als ein Risiko der SWPL betrachtet das Unternehmen heute die eingeschränkte Flexibilität der SWPL Architektur bei grundsätzlichen Innovationen im Bereich der Motorsteuerung, die unvorhersehbar sind und eventuell nicht mit der vorhandenen SWPL Architektur umgesetzt werden können.

Insgesamt betrachtet hat sich die Einführung der SWPL für VDO nach eigener Aussage jedoch mehr als rentiert. Dies war zu Beginn der Umsetzung noch ungewiss, da das Unternehmen keine ökonomische Analyse durchgeführt hat. Die Möglichkeit zur Gewinnung detaillierter ökonomischer Daten im Vorfeld einer tatsächlichen Entscheidung über die Einführung einer SWPL wurde von den Workshopteilnehmern insgesamt sehr positiv bewertet.

---

[104] Die Entwicklung einer Motorvariante eines bekannten Motors dauert ca. 9 bis 12 Monate.

6.3. Allgemeine Bewertung und Sensitivitätsanalyse

Das e-FACTS Modell wurde in zwei realen Unternehmenssituationen erfolgreich angewendet. Dies zeigt die grundsätzliche Anwendbarkeit der Methodik, auch wenn es sich hierbei nicht um eine tatsächliche Validierung handelt, also die wissenschaftliche Sicherstellung, dass das Modell die gestellten Anforderungen erfüllen kann. Eine Validierung ist bei den derzeitigen Rahmenbedingungen nur sehr schwierig oder sogar unmöglich. Es fehlt grundsätzlich an einer umfangreichen Datenbasis von vergleichbaren SWPL Projekten. Weiterhin würde dafür eine reine Datensammlung durchgeführter SWPL Projekte nicht ausreichen. Diese Projekte müssten zusätzlich relativ hohe Ansprüche erfüllen, um für eine Validierung des hier vorgestellten Modells in Frage zu kommen. Dazu gehört u.a. mindestens ein zusätzlich analysiertes Referenzprojekt vor Einführung der SWPL im gleichen Unternehmen, um belastbare individuelle Vergleichswerte zu erhalten. Zusätzlich wird eine Validierung dadurch erschwert, dass in der Literatur meist lediglich von den Ergebnissen berichtet wird, die mittels der Einführung einer SWPL erzielt worden sind. Um eine Anwendung des Modells zu ermöglichen ist jedoch eine wesentlich umfangreichere Datenbasis notwendig. Bei keinem Fallbeispiel, über das in der Literatur bisher berichtet worden ist, konnte eine Outside-in Anwendung[105] des Modells durchgeführt werden. Dazu wäre immer ein zusätzliches Gespräch oder Treffen mit den Projektverantwortlichen notwendig gewesen. Um dennoch zumindest die praktische Anwendbarkeit des Modells zu demonstrieren, wurden zwei Praxisbeispiele im Detail analysiert.

Die Belastbarkeit der Ergebnisse wurde jeweils durch Auslenkung einzelner Parameter erhärtet. Durch diese Sensitivitätsanalyse (SA) wird die Stabilität der getroffenen Empfehlung bei möglichen Fehlern wichtiger Parameter getestet. Zu den bedeutendsten Unsicherheitsfaktoren des e-FACTS Modells gehört die FP Bewertung. Schließlich handelt es sich dabei um eine Schätzmethode zur Bestimmung des notwendigen Aufwands und zwar zu einem sehr frühen Zeitpunkt im Projektverlauf. Eine solche Schätzung wird nicht 100%ig korrekt, sondern „immer schwierig und fehleranfällig“ [Pink02, S.75] sein. Auch wenn die Notwendigkeit der Aufwandsbestimmung allgemein anerkannt wird [vgl. z.B. Boeh00, S. 14, Pink02, S.76, Jone07, S. 230], besteht sowohl in der Literatur aber insbesondere auch in der

[105] Bei einer Outside-in Anwendung würden ausschließlich öffentlich zugängliche Daten genutzt, um das Modell anzuwenden.

Praxis[106] eine gewisse Skepsis gegenüber Aufwandsbewertungen bei Software-Projekten. „Da solche Rechenverfahren sehr ungenaue Werte liefern, werden sie in der betrieblichen Praxis kaum akzeptiert“ [Stah05, S.462]. Stattdessen wird vielfach auf Erfahrungen und vorhandene Vergleichswerte zurückgegriffen. Bei der Umsetzung einer neuen Entwicklungsmethode, wie z.B. der SWPL, ist dies allerdings nicht möglich.

Andererseits findet man in der Literatur auch Aussagen, dass „function points are useful estimators since they are based on information that is available early in the project life cycle“ [Boeh00, S.15]. Daher gilt es, die Vorbehalte durch nachprüfbare Argumente zu entkräften. Eine Möglichkeit dazu ist, die Ermittlung eines maximal möglichen Fehlers bei der Aufwandsbestimmung, ohne die endgültige Empfehlung zu beeinflussen. Man berücksichtigt quasi aktiv die systemimmanente Ungenauigkeit der FP-Analyse indem die Unsicherheit als maximal angenommen wird.

Lautet das Ergebnis des e-FACTS Modells, dass der SWPL Ansatz unter den individuellen Rahmenbedingungen ökonomisch sinnvoll ist, dann gilt diese Aussage zunächst unter der Annahme, dass die Aufwandsbewertungen mittels FP-Analyse korrekt sind. Eine Abweichung kann hier allein schon aufgrund des gewählten Ansatzes nicht ausgeschlossen werden[107]. Vielmehr gilt es die immanente Unsicherheit geeignet zu kontrollieren. Ein allgemeiner Fehler bei der Aufwandsbestimmung mittels FPs ist dabei allerdings unbedeutend, denn dieser würde sich sowohl auf die Berechnung des Aufwands zur Entwicklung der Einzellösungen als auch bei der SWPL auswirken. Die prozentuale Einsparung würde sich dabei nicht verändern. Selbst auf einer Einzelkomponentenebene würde sich lediglich die Höhe der Einsparung verändern, das prozentuale Verhältnis und damit auch die Empfehlung hinsichtlich einer Verwendung als Asset oder einer Einzelentwicklung blieben dabei identisch.

Relevant wird eine Fehlschätzung des Aufwands mittels FPs allerdings, wenn sich diese nicht gleichmäßig auf alle Elemente sondern insbesondere auf die Bewertung des Mehraufwands zur Variabilisierung von Komponenten und damit zur Erstellung von wiederverwendbaren

---

[106] Der Großteil der Entwickler, mit denen im Rahmen der Erstellung der vorliegenden Arbeit Gespräche geführt worden sind, war gegenüber Aufwandsbewertungen und insbesondere der Function Point Methodik zunächst negativ eingestellt. In allen Fällen überwog am Ende aber der erzielbare Informationsgewinn die mit der Methodik verbundene Unsicherheit.

[107] Ein möglicher Grund könnte bspw. die Unerfahrenheit der bewertenden Person mit der FP-Methodik sein. Laut einer Studie in [Jone07, S. 261] liegt die durchschnittliche Abweichung bei nicht-zertifizierten Zählern um den Faktor 10 höher als bei zertifizierten und damit gut ausgebildeten und mit der Methodik intensiv vertrauten Personen.

Assets für die Asset Base auswirken würde. Sollte hier ein zu geringer Mehraufwand geschätzt werden, würde die SWPL zu positiv bewertet. Im ungünstigsten Fall liegt jede Bewertung des durch die SWPL verursachten Mehraufwands innerhalb des angenommenen Sicherheitsbereichs am oberen Ende der Abweichung (Aufwands zu gering geschätzt) und jede Komponente der Einzelentwicklung am unteren Ende der Abweichung (Aufwand zu hoch geschätzt). Um diese Unsicherheit bewertbar zu machen, berechnet man die maximal mögliche Abweichung bei der Bewertung des Mehraufwands zur Herstellung der Wiederverwendbarkeit, ohne dass sich dadurch die Vorteilhaftigkeit des SWPL Ansatzes ändert. Die Betrachtung einer zu geringen Aufwandsschätzung bei Einzelentwicklungen kann entfallen, da diese immer auch die Basis für die Wiederverwendung darstellen und sich damit ebenfalls positiv auf die Entwicklung der Assets auswirken würde. Allein die Betrachtung des Faktors für den Mehraufwand zur Sicherstellung der Wiederverwendbarkeit der Komponenten ist relevant. Hier sollte der maximal mögliche Fehler ohne eine Änderung der finalen Empfehlung ermittelt werden.

Dieser maximale Fehlerwert kann dann mit der eigenen Einschätzung der zur erwarteten Ungenauigkeit der FP-Analyse verglichen werden. Oder anders gesagt, die beteiligten Personen können sich darauf verständigen, welches Sicherheitsniveau einzukalkulieren ist. In der Literatur zur FP-Methodik wird von einer durchschnittlichen Ungenauigkeit zwischen +/-10% und +/-20% ausgegangen (siehe Teil A Kapitel 5.7). Dieser Wert ist als untere Grenze anzusehen, da die Autoren eine intrinsische Motivation zur positiven Bewertung der FP-Methodik haben. Daher kann dieser Basiswert um einen individuellen Risikoaufschlag erhöht werden. Liegt der tolerierbare Fehler für die einzelne Projektbewertung deutlich höher als die individuell festgelegte Grenze, kann angenommen werden, dass das Modell selbst im ungünstigen Fall die richtige Empfehlung gegeben hat. Die mit der Verwendung der FP-Methodik verbundene Unsicherheit wird dadurch transparent und in die Überlegungen und Berechnungen ausreichend integriert, sodass von einer Entscheidung mit reduzierter Unsicherheit ausgegangen werden kann.

Zusätzlich zur FP-Bewertung sind je nach individueller Projektsituation weitere Unsicherheitsfaktoren zu berücksichtigen, wie z.B. bei der Hochrechnung der Teilergebnisse bei der Analyse der DOMINION Middleware auf das Gesamtsystem. Auch diese Parameter sollten bei der SA berücksichtigt und entsprechend ausgelenkt werden. Entweder können Worst-Case Szenarien angenommen werden oder man ermittelt wieder den maximal

tolerierbaren Fehler. Im ersten Fall wird die ungünstigste als realistisch anzunehmende Situation zu Grunde gelegt und das Modell mit den daraus resultierenden Werten berechnet. Das Ergebnis sollte im Idealfall unverändert bleiben. Im zweiten Fall der Maximalbetrachtung ist die Frage zu beantworten, ob der resultierende maximale Fehler als ausreichend sicher eingestuft wird. Wird dies mit Ja beantwortet, wurde die Unsicherheit ausreichend berücksichtigt und die Empfehlung des Modells bleibt bestehen.

Um die Auswirkungen der SA besser zu verdeutlichen werden im Folgenden drei beispielhafte SWPL Anwendungsbeispiele aufgezeigt und Trends abgeleitet. Tabelle 49 zeigt die drei Szenarien, die nun kurz beschrieben werden:

- **Szenario 1** bildet ein kleines SW Projekt ab, das zur Entwicklung einer einzelnen Komponente, möglicherweise im Rahmen eines größeren Projektes, dient. Der Anteil der wieder verwendbaren Assets ist mit 25% eher niedrig angesetzt und auch die Anzahl der Wiederverwendungen liegt mit lediglich 2 deutlich am unteren Ende der normalen Anwendungsfälle. Insgesamt bildet das Szenario damit ein kleines Entwicklerteam nach, das mit der Entwicklung einer einzelnen Komponente beauftragt worden ist, bisher noch keine Erfahrung mit SWPL gesammelt hat und den Ansatz daher zunächst einmal exemplarisch erproben möchte.
- **Szenario 2** steht für ein mittelgroßes SW-Projekt, wie z.B. eine einfache Motorsteuerung. Die Anzahl der geplanten Wiederverwendungen und der Anteil der Asset Base am Gesamtumfang liegen im mittleren Bereich. Dieses Szenario bildet ein Unternehmen ab, das bereits erste Erfahrungen mit SWPL gesammelt hat. Daraus resultieren auch der höhere angenommene Anteil der wieder verwendbaren Komponenten und die häufigeren Wiederverwendungen.
- **Szenario 3** ist typisch für ein großes SW-Projekt, wie man es bei einem OEM oder Zulieferer finden könnte. Erneut ist die Gesamtentwicklung eines Produktes abgedeckt, wie z.B. eine komplexe Motorsteuerung oder ein Navigationssystem. Das Szenario bildet ein erfahrenes Unternehmen im Umgang mit SWPL ab. Der Anteil der Asset Base liegt mit 75% recht hoch und 8 Wiederverwendungen sind ebenfalls eher im oberen erwarteten Bereich angesiedelt.

Szenario		1	2	3
Beschreibung	Einheit	Kleines SW Projekt, das meist lediglich eine Komponente in einem größeren SW-Projekt abdeckt	Mittelgroßes, vollständiges SW-Projekt	Größeres SW Projekt in einem größeren Unternehmen
Beispiel		Kleine SW Komponente, wie z.B. ESR bei VDO	Einfache Motorsteuerung	Komplexe Motorsteuerung oder einfaches Navigationssystem
Anzahl Wiederverwendungen	Anzahl	2	5	8
Umfang Einzelentwicklung	FPs	250	1.000	2.000
Anteil Asset Base bei SWPL	Prozent	25%	50%	75%
Mehraufwand für Wiederverwendbarkeit	Prozent	150%	200%	250%
Aufwand für tats. Wiederverwendung ggü. Einzelentwicklung	Prozent	20%	10%	5%
Aufwand Erstellung Asset Base	FPs	94	1.000	3.750
Berechnung				
Aufwand Einzelentwicklung	FPs	750	6.000	18.000
Aufwand SWPL	FPs	694	4.500	9.750
Vorteil SWPL auf Basis Einzel	Prozent	8%	25%	46%
Vorteil SWPL auf Basis SWPL	Prozent	8%	33%	85%

Tabelle 49 – Übersicht der drei Szenarios für die SA

Insgesamt werden im Rahmen der SA vier wichtige Parameter betrachtet, die einen signifikanten Einfluss auf die Bewertung der Vorteilhaftigkeit des SWPL Ansatzes haben können. Diese Parameter sind:

- Anzahl der Wiederverwendungen
- Prozentualer Anteil der Asset Base am Gesamtumfang der SW
- Mehraufwand für die Erzeugung der Wiederverwendbarkeit einer Asset Komponente
- Aufwand für die tatsächliche Wiederverwendung einer Komponente

Im Folgenden werden im Rahmen der SA jeweils für alle drei Szenarien nacheinander alle vier Parameter ausgelenkt. Die Veränderungen werden unter Beibehaltung aller weiteren Einflussfaktoren in einem Diagramm abgetragen und analysiert. Dabei wird jeweils die prozentuale Einsparung (gemessen in FPs) durch die Verwendung der SWPL ins Verhältnis gesetzt zum Gesamtaufwand (in FPs) für eine entsprechende Einzelentwicklung aller betrachteten Produkte auf Basis der SWPL. Das Ergebnis gibt die Vorteilhaftigkeit der SWPL an. Ein höherer Wert bedeutet, dass die SWPL Einsparung durch die Veränderung des Parameters verhältnismäßig gestiegen ist. Die prozentuale Betrachtung ist erforderlich, da ansonsten bspw. bei einer größeren Anzahl an Wiederverwendungen fast automatisch die Einsparung (gemessen in FP) ansteigt, allerdings bei einem gleichzeitig gesteigerten Gesamtaufwand, dessen Höhe aber nicht transparent gemacht wird.

Die Bandbreite der Auslenkung der Parameter basiert jeweils auf den Erkenntnissen des Verfassers in Verbindung mit den bereits im vorderen Teil der Arbeit angegebenen Vergleichswerten aus der Literatur. Dies ist erforderlich, um eine Vergleichbarkeit der

resultierenden Ausschläge im Rahmen der SA sicherzustellen und eine Aussage hinsichtlich der Einflussstärke der einzelnen Parameter im Vergleich treffen zu können.

Schließlich wird, wo es sinnvoll und möglich ist, ein Maximalwert, bzw. Minimalwert für den tolerierbaren Fehler angegeben, bei dem noch keine Änderung der Vorteilhaftigkeit erfolgt.

Anzahl der Wiederverwendungen:

Der Parameter gibt die Anzahl der Wiederverwendungen an. Diese Zahl ist jeweils um eins zu erhöhen, um die tatsächliche Anzahl an entwickelten Produkten ablesen zu können. Im Rahmen der SA wird ein Spektrum von 0 bis 10 Wiederverwendungen betrachtet, wodurch der Großteil der realen Anwendungsszenarien abgedeckt sein dürfte. Insgesamt steigt mit einer größeren Zahl an Wiederverwendungen die Vorteilhaftigkeit einer SWPL an. Abbildung 37 zeigt die Veränderung der prozentualen Vorteilhaftigkeit des SWPL Ansatzes in Abhängigkeit von der Anzahl der Wiederverwendungen.

Lediglich bei Szenario 1 ist bereits bei der ersten Wiederverwendung der Break-Even erreicht, bei den beiden anderen Szenarien erfolgt dies erst mit der zweiten Wiederverwendung. Die positivere Entwicklung bei Szenario 3 im Vergleich zu den beiden anderen Szenarien basiert auf dem höheren Asset Anteil, wodurch sich jede Wiederverwendung stärker auswirkt. Die degressive Entwicklung ergibt sich durch die Betrachtung der Einsparung im Verhältnis zum Gesamtaufwand aller vergleichbaren Einzelentwicklungen. Neben dem linearen Zuwachs bei dem eingesparten FPs je Wiederverwendung erhöht sich natürlich auch der entsprechende Gesamtaufwand, wodurch sich die prozentuale Einsparung reduziert. Dieser Prozentwert kann sich maximal dem Anteil der Asset Base am Gesamtentwicklungsumfang annähern, z.B. 75% bei Szenario 3.

Die Anzahl der Wiederverwendungen hat den größten Einfluss auf die Vorteilhaftigkeits-entscheidung, da hierbei die größte Wirkung auf die prozentuale Einsparung durch die SWPL erzielt wird. Wichtig zu beachten ist dabei, dass die Anzahl relativ leicht zu beeinflussen ist und außerdem ein zu Beginn gesetztes Ziel später auch rigoros verfolgt werden muss. Erst durch die tatsächliche Wiederverwendung der Assets in realen Produkten werden die positiven Rückflüsse der anfänglichen Investition ausgelöst. Daher wirkt sich eine spätere Veränderung dieses Parameters im Verlauf des Lebenszyklusses der SWPL direkt auf die ökonomische Vorteilhaftigkeitsbewertung aus.

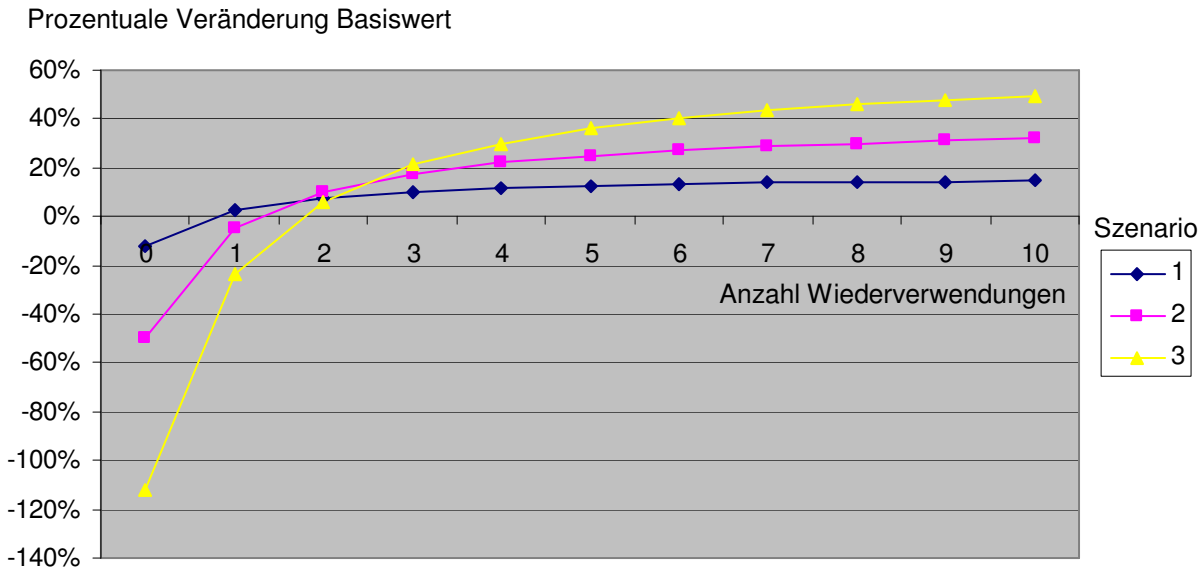

Abbildung 37 – Einfluss der Anzahl der Wiederverwendungen auf SWPL

Prozentualer Anteil der Asset Base am Gesamtumfang der SW:

Dieser Wert zeigt den Anteil der SW Komponenten an, gemessen als Anteil des Gesamtentwicklungsaufwands (in FPs), die im Rahmen der Asset Base wiederverwendet werden sollen. Ein Anteil von 50% bedeutet, dass genau die Hälfte der Komponenten (gewichtet mit ihrem Entwicklungsaufwand als Einzelentwicklung) als Asset wieder verwendbar entwickelt werden, die andere Hälfte jeweils für ein Produkt als Einzelentwicklung individuell entwickelt wird.

Ein höherer Anteil der Asset Base hat in allen drei Szenarien eine positive lineare Steigerung der prozentualen Vorteilhaftigkeit zur Folge (siehe Abbildung 38). Der stärkere Anstieg bei Szenario 3 ist durch die höhere Anzahl an Wiederverwendungen begründet, denn durch einen höheren Wiederverwendungsanteil steigt der Vorteil mit jeder Wiederverwendung stärker an. Neben der direkten, hier gemessenen Wirkung dient dieser Wert auch bei der Anzahl der Wiederverwendungen und dem Mehraufwand für Wiederverwendbarkeit als wichtiger Hebel. Die Empfehlung für oder gegen eine SWPL wird durch diesen Parameter nicht geändert, allein die Volatilität gegenüber Fehlern bei den anderen Parametern steigt mit größerem Anteil der Asset Base an.

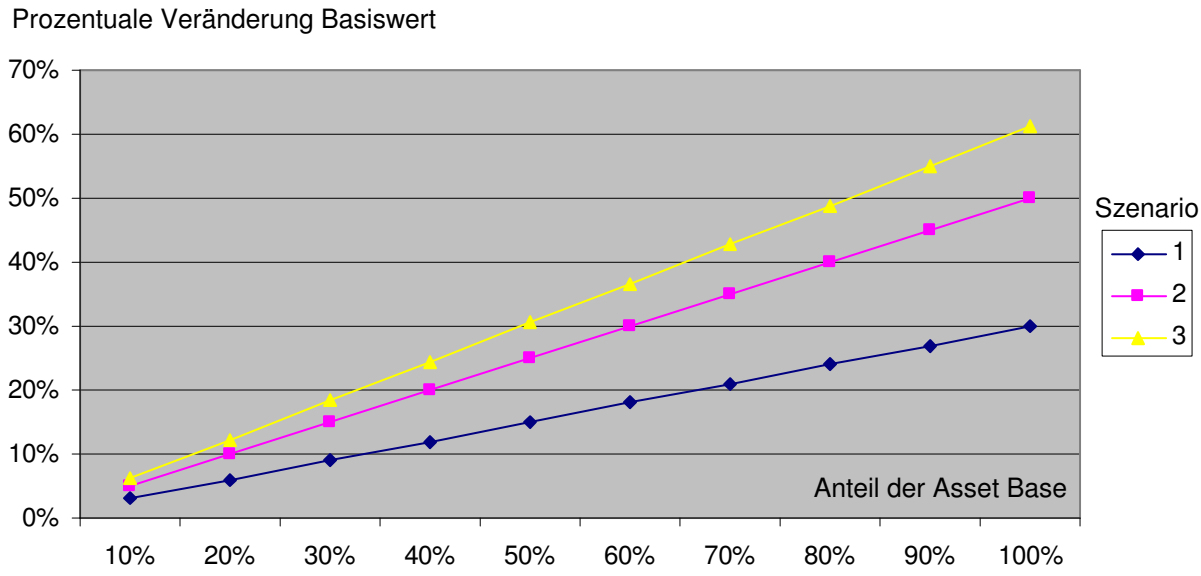

Abbildung 38 – Einfluss der Größe der Asset Base

Mehraufwand für die Erzeugung der Wiederverwendbarkeit einer Asset Komponente:

Die Erstellung einer wiederverwendbaren SW-Komponente ist im Allgemeinen aufwendiger als eine Einzelentwicklung. Die Höhe dieses Mehraufwands wird als Faktor angegeben, mit dem der FPs Wert einer Einzelentwicklung zu multiplizieren ist. Ein höherer Faktor bedeutet, dass die Erstellung der Asset Base aufwendiger ist, was sich negativ auf die Vorteilhaftigkeit der SWPL auswirkt.

Dieser Parameter dient, wie oben bereits erwähnt, dazu Fehler bei der Aufwandsbewertung mittels FPs zu berücksichtigen. Sollte der Mehraufwand für die Wiederverwendbarkeit zu gering geschätzt worden sein, würde sich dies auf die Optimierungsentscheidung auf Einzelkomponentenebene auswirken. Die Gesamtentscheidung für oder gegen eine SWPL kann dadurch entscheidend beeinflusst werden. Daher sollte dieser Parameter besonders sorgfältig auf ein mögliches maximales Fehlerpotenzial untersucht werden.

Zusätzlich Komplexität für die Analyse ergibt sich, da sich der tatsächliche Wert erst nach der Implementierung ermitteln lässt und die in der Literatur vorhandenen Vergleichswerte eine extrem große Streuung aufweisen, sodass signifikante reale Abweichungen möglicherweise erst zu spät erkannt werden (vgl. Kapitel 3.8).

Zur Erklärung, ein Wert von 100% in Abbildung 39 bedeutet, dass eine Einzelentwicklung und eine Entwicklung mit Wiederverwendung den gleichen Aufwand erzeugen würden. Ein

Wert von 200% bedeutet, dass eine wieder verwendbare SW im Rahmen einer SWPL den doppelten Aufwand im Vergleich zu einer Einzelentwicklung erzeugen würde.

Die Unterschiede zwischen den Szenarien ergeben sich maßgeblich durch die Anzahl der Wiederverwendungen und die Größe der Asset Base. So besteht in Szenario 1 schon bei einem geringen Anstieg des Mehraufwands kaum eine Chance diesen im Verlauf des Lebenszyklusses der SWPL durch Einsparungen auszugleichen. Das Sicherheitsniveau liegt hier bei einer maximalen Fehleinschätzung von 225%, was einer Abweichung von der ursprünglichen auf der doppelten FP Zählung basierenden Berechnung (150%) von genau 50% entsprechen würde. Dieser Sicherheitspuffer ist als eher gering anzusehen, obwohl wie bereits mehrfach erwähnt im Allgemeinen von einer Genauigkeit des FP Ansatzes von +/- 10 bis 20% ausgegangen wird. Bei konservativer Herangehensweise ist dennoch ein solcher Puffer als unzureichend abzulehnen.

Bei Szenario 2 beträgt der kritische maximale Wert 400%, was einer Verdopplung gegenüber dem ursprünglich ermittelten Wert (200%) entspricht. Bei einem solchen Puffer kann ein Unternehmen im Allgemeinen von einer sicheren Entscheidung ausgehen. Die finale Bewertung sollte aber immer entsprechend der individuellen Risikobereitschaft erfolgen.

Szenario 3 wird bei einem Faktor für den Mehraufwand von 643% negativ. Dies entspricht einem Puffer gegenüber dem ursprünglichen Wert (250%) von mehr als dem 2,6 fachen. Eine derart große Abweichung dürfte nur in seltenen Fällen auftreten, insbesondere wenn man sich erinnert, dass der ursprüngliche Wert eben nicht, wie bei vielen bisherigen Ansätzen zur ökonomischen Bewertung von SWPL, geschätzt, sondern auf Basis einer doppelten FP Bewertung individuell ermittelt worden ist.

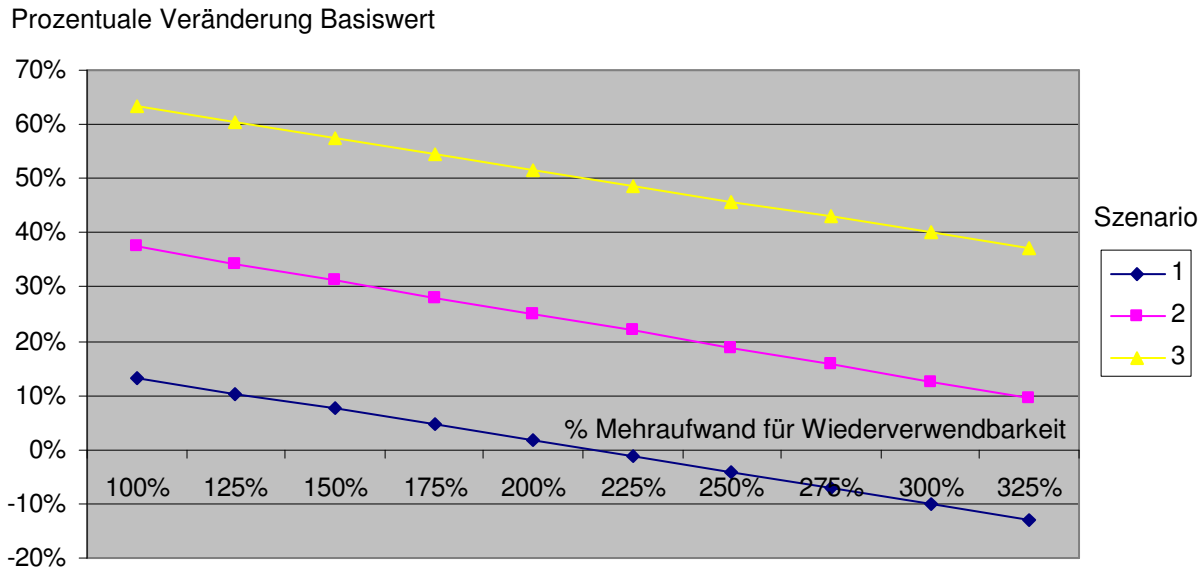

Abbildung 39 – Einfluss des Mehraufwands für Wiederverwendbarkeit

Aufwand für die tatsächliche Wiederverwendung einer Komponente:

Die Wiederverwendung einer vorhandenen Komponente erzeugt einen Aufwand, z.B. durch das Suchen und Identifizieren des Assets oder einen individuellen Integrationstest. Die Summe dieses Aufwands wird als Faktor des ursprünglichen Entwicklungsaufwands des Assets ausgedrückt. Bei der Berechnung des ursprünglichen Entwicklungsaufwands wird der Mehraufwands für die Erstellung der Wiederverwendbarkeit mit berücksichtigt.

Im Allgemeinen beträgt der Aufwand für die tatsächliche Wiederverwendung lediglich zwischen einigen wenigen und maximal 10% des ursprünglichen Entwicklungsaufwands[108], und in diesem Bereich zeigt sich kein signifikanter Einfluss des Parameters auf die Vorteilhaftigkeit der SWPL. Dennoch zeigt Abbildung 40, dass der Einfluss signifikant zunimmt bei steigender Anzahl an Wiederverwendungen und einer größeren Asset Base. Daher sollte dieser Parameter insbesondere bei großen SWPL Projekten genau beachtet werden. Erneut lässt sich der Wert erst a posteriori genau ermitteln. Diesmal erst nach der tatsächlichen Wiederverwendung, was nochmals deutlich später ist als bspw. beim Mehraufwand für Wiederverwendbarkeit, der bereits nach Erstellung der Asset Base bestimmt werden kann.

---

[108] COPLIMO geht bei den Relative Cost of Reuse (RCR) von einem üblichen Wert zwischen 2% (bei reinem Suchaufwand) und 8% (inkl. Testing und Integration) aus, vgl. Kapitel 3.2.

Positiv ist anzumerken, dass selbst bei einem großen SWPL Projekt, wie bei Szenario 3, erst eine Veränderung der Vorteilhaftigkeit eintritt, wenn jede Wiederverwendung einen zusätzlichen Aufwand von ca. 1/3 des ursprünglichen Entwicklungsaufwands erzeugt. Ein solcher Wert sollte ohnehin als ein deutliches Warnsignal gelten, da dies auf einen erheblichen Anpassungsbedarf der wieder verwendbaren Softwarekomponente hindeutet und eher bei white box Wiederverwendung, also bei einer direkten Anpassung des Codes, typisch ist (vgl. Kapitel 3.2).

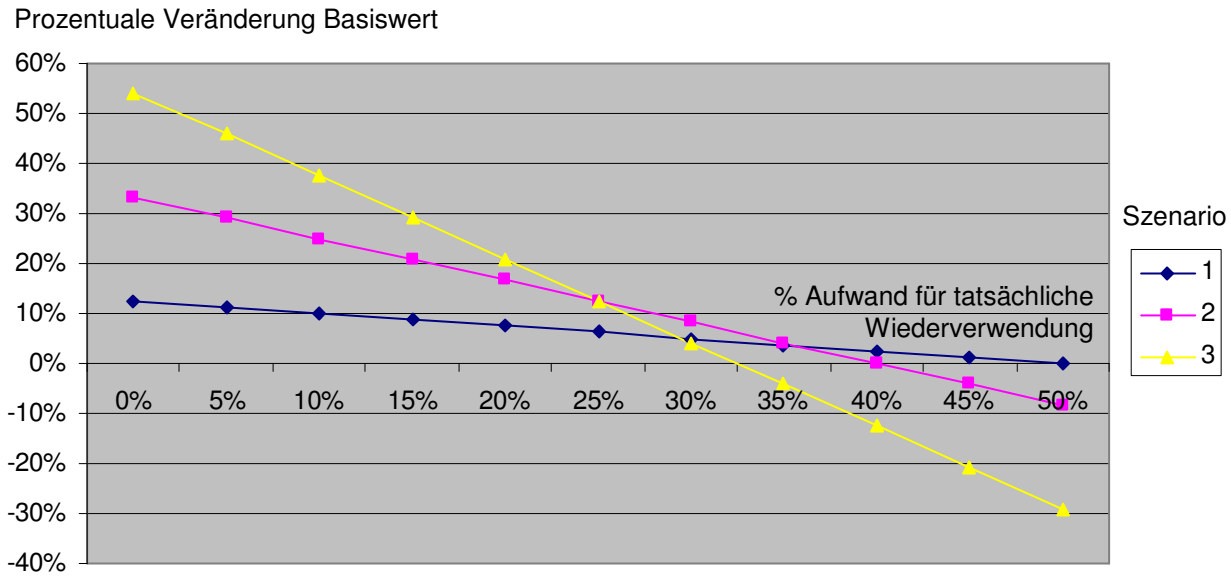

Abbildung 40 – Einfluss des Aufwands zur tatsächlichen Wiederverwendung

Abschließend lässt sich festhalten, dass die SA bei der Einführung einer SWPL relevante Aussagen zur Bedeutung von Fehlern bei den vier wichtigsten Parametern ermöglicht. So konnte bspw. gezeigt werden, dass die Anzahl der Wiederverwendungen den größten Einfluss auf die Vorteilhaftigkeit hat. Die Größe der Asset Base dient als wichtiger Hebel, der die Auswirkungen bei Fehlern in den anderen Parametern deutlich erhöht. Über den Mehraufwand für Wiederverwendbarkeit konnte die Unsicherheit aufgrund der Verwendung der FP Methodik transparent gemacht werden. Und schließlich konnte gezeigt werden, dass der Aufwand für die tatsächliche Wiederverwendung von Assets lediglich in Ausnahmefällen eine Rolle bei der Entscheidung für oder gegen eine SWPL spielen dürfte.

An dieser Stelle sei in Erinnerung gerufen, dass SWPL keine Einsparungen von einigen Prozent, sondern durch die verstärkte Wiederverwendung Verbesserungen um einige Faktoren ermöglichen sollen. Mit Hilfe dieses großen Verbesserungspotenzials können die

Unsicherheiten bei einzelnen Modellparametern kompensiert werden. Allerdings sollte dies im Einzelfall überprüft werden. Eine detaillierte SA bietet sich daher bei der Anwendung des e-FACTS Modells an, um die Unsicherheit zu reduzieren und die Entscheidung entsprechend des unternehmensspezifischen Risikoprofils abzusichern.

7. Zusammenfassung und Ausblick

7.1. Zusammenfassung

Die Zahl der elektronischen Systeme im Automobil ist in den letzten Jahren stark gestiegen. Ein Großteil der heutigen Innovationen basiert auf der Elektronik. Die Systeme und insbesondere die darin enthaltene SW haben inzwischen einen hohen Komplexitätsgrad erreicht. Dieser Trend führte zu einem Umdenken in der ansonsten eher ingenieurslastigen Automobilindustrie. Die Bedeutung von Embedded Systems und der SW ist gestiegen. In Verbindung mit der generellen Entwicklung in der Automobilindustrie zu einer stärkeren Produktdifferenzierung mit kürzeren Entwicklungszeiten sowie dem hohen Kostendruck führte dies zu zahlreichen Versuchen, bestehende SW wiederzuverwenden. SWPL sind ein geeigneter Ansatz, um eine hohe strategische Wiederverwendung zu erreichen. Produktlinien sind in der Automobilindustrie im HW-Bereich seit längerer Zeit etabliert, SWPL sind hier jedoch ein relativ neues Thema. Dies liegt u.a. an der nicht trivialen Umsetzung der SWPL-Idee in die reale Welt und fehlenden ökonomischen Daten, um die Entscheidung zur Einführung einer SWPL und insbesondere die damit verbundenen Risiken bewerten zu können.

Die vorliegende Arbeit hat zum Ziel, ein einfaches Modell zur ökonomischen Bewertung von SWPL-Ansätzen in der Automobilindustrie zu erstellen, das zu einem frühen Zeitpunkt anwendbar ist und dennoch ein möglichst detailliertes Ergebnis ermittelt.

Teil A der vorliegenden Arbeit liefert eine umfangreiche Einarbeitung in das Thema. Diese enthält die Definition und Abgrenzung der relevanten Begriffe, wie Embedded Systems, Produktlinien und SW-Wiederverwendung. Ebenso werden die wichtigsten Bewertungsmethoden für SW-Entwicklungsaufwände vorgestellt und analysiert. Teil A endet mit der Vorstellung der Feature Point Methodik, die im Rahmen der Arbeit zur Bewertung von SW-Aufwänden genutzt wird.

Teil B beschäftigt sich zunächst mit dem Einfluss der Unternehmensziele auf SWPL. Anschließend werden die wichtigsten ökonomischen Modelle für SWPL vorgestellt und ihre Stärken und Schwächen analysiert. Zusätzlich wurde die Wirkung von SWPL auf die Produktqualität analysiert. Darauf aufbauend wurde der Bedarf für ein neues Bewertungsmodell abgeleitet, das

- universell, aber zumindest für die gesamte Automobilindustrie,
- einfach und mit geringem Zeiteinsatz,
- zu einem frühen Zeitpunkt (bevor wichtige Entscheidungen getroffen sind) und
- mit einem ausreichenden Detailgrad

anwendbar ist. Das Ergebnis ist e-FACTS, die economic Feature Point Analysis with convenient timing for SWPL in Automotive. Dieses Modell adressiert alle geforderten Kriterien, insbesondere die detaillierte ökonomische Bewertung inkl. Komponenten- oder Funktionsebene. e-FACTS besteht aus 4 Schritten:

- **Basiskriterien:** Ein Katalog mit den wichtigsten Fragen ermittelt die grundsätzliche Eignung des Unternehmens für die Einführung einer SWPL mit dem Ziel, den Analyseaufwand für klar ungeeignete Fälle zu minimieren.
- **Einfache Berechnung:** Mittels einer auf dem SIMPLE Ansatz basierenden Bewertungsformel und der Vorgabe von Standardwerten können Unternehmen eine sehr grobe, erste rule-of-thumb Bewertung durchführen. Dies dient erneut der frühzeitigen Identifizierung ungeeigneter Fälle.
- **FP Berechnung:** Schritt 3 ist der umfangreichste und führt die FP Methodik zur Bewertung des Aufwands für Wiederverwendung auf Komponentenebene ein. Die FP Methodik ist ein von der Unternehmensberatung McKinsey&Company adaptierter Function Point Ansatz nach IFPUG.
- **Business Case:** Die Analyse schließt mit der Aufbereitung der gewonnen Daten in einem Business Case.

Die Einführung der FP Methodik zur Bewertung des Wiederverwendungsaufwands ermöglicht eine einfache Erfassung auf Komponenten- bzw. Funktionsebene. Die notwendigen Anpassungen und Herleitungen werden in Teil B Kapitel 5.4 im Detail erläutert, wodurch jeder Leser der vorliegenden Arbeit in die Lage versetzt wird, das e-FACTS Modell anzuwenden. Die universelle Anwendbarkeit wird durch die Beschreibung der Anwendungsmöglichkeiten bei unterschiedlichen Produktszenarien und bei unterschiedlichen Unternehmenszielen in Teil B Kapitel 5.6 ergänzt.

Die Arbeit schließt mit der Beschreibung der Ergebnisse aus der praktischen Anwendung des Modells in zwei Unternehmen aus dem Bereich der Automobilindustrie, Siemens VDO als deutscher Automobilzulieferer und dem Deutschen Zentrum für Luft und Raumfahrtforschung (DLR). Im ersten Fall wurde eine SWPL für Motorsteuerungs-SW betrachtet, beim DLR eine Middleware Plattform, die die Entwicklung von neuen Funktionen für die Automobilindustrie, erleichtert. Ergänzt werden die Anwendungsbeispiele um eine SA, mit deren Hilfe die Stabilität der mittels des e-FACTS Modells erzeugten Empfehlungen überprüft werden kann. Auf Basis von drei zusätzlichen Szenarien wurden die Möglichkeiten des Modells und der SA detailliert beschrieben mit dem Ziel eine individuelle Anwendung zu ermöglichen.

7.2. Fazit

Die beiden Fallstudien in Kapitel 6 dienen der Überprüfung der praktischen Anwendbarkeit des e-FACTS Modells. Sie verdeutlichen die umfangreichen Möglichkeiten, die das Modell bietet. Die Fallstudien decken zwei unterschiedliche automobile SW-Entwicklungsbereiche ab: Einerseits eine klassische Anwendungsentwicklung bei VDO und andererseits beim DLR die Entwicklung einer Middleware Plattform als Unterstützung für die Entwicklung der SW. Dies zeigt die universelle Einsetzbarkeit des Modells. Obwohl in beiden Fällen nur ein eintägiger Workshop zur Verfügung stand, konnte das Modell in beiden Fällen erfolgreich angewendet werden. Dies ist ein Indiz für die schnelle und einfache Anwendung des Ansatzes. Gleichzeitig belegen die gewonnen Ergebnisse die Relevanz einer Erfassung der Aufwände auf Komponentenebene statt – wie bei vielen Modellen bisher vorgesehen – auf Produktebene. Erst durch diese Detailbetrachtung können individuelle Besonderheiten, wie z.B. die außergewöhnlich hohe Produktivität beim Laserscanner des DLR auf Basis der DOMINION Middleware, erkannt und entsprechend korrekt berücksichtigt werden. Die festgestellten, teilweise erheblichen Unterschiede zwischen einzelnen Komponenten innerhalb eines Unternehmens oder sogar einer Abteilung verdeutlichen die Bedeutung dieses detaillierten Vorgehens, das durch e-FACTS unterstützt wird.

Die Anwendung beim DLR lieferte eine Möglichkeit zur Plausibilisierung der FP-Werte, da das Projekt bereits abgeschlossen war und die tatsächlichen Entwicklungsaufwände zum Vergleich vorlagen. Zusätzlich wurde für beide Fallstudien eine Sensitivitätsanalyse durchgeführt, um die Stabilität der ermittelten Empfehlung unter Berücksichtigung von etwaigen Fehlern bei der Ermittlung von Eingabeparametern zu bestimmen. Diese wurde mit

besonderem Fokus auf mögliche Ungenauigkeiten bei der Aufwandsabschätzung mittels der FP-Methodik vorgenommen. In beiden Fällen blieb die Empfehlung selbst bei Annahme eines Worst-Case Szenarios stabil. Grund dafür ist die große Vorteilhaftigkeit des SWPL Ansatzes, die keine Verbesserung von einigen Prozentpunkten sondern um Faktoren verspricht [siehe Clem02, S. xxi und Clem05, S. 47]. Zusätzlich konnten mit Hilfe von drei Szenarien die allgemeinen Vorteile einer SA bei der Anwendung des e-FACTS Modells gezeigt werden.

Das e-FACTS Modell hat demnach die gesetzten Anforderungen erfüllt. Es bietet eine nützliche Erweiterung gegenüber den bestehenden Modellen. Insbesondere die Erfassung der individuellen Aufwände auf Komponentenebene jeweils mit und ohne Wiederverwendung ermöglicht eine wesentlich detailliertere Datenbasis.

7.3. Weiterer Untersuchungsbedarf

e-FACTS erfüllt die zu Beginn der Arbeit gestellten Anforderungen. Es liefert damit einen relevanten Beitrag zur Verbesserung und Vereinfachung der ökonomischen Bewertung von SWPL Ansätzen für ein spezielles Unternehmen. Allerdings löst e-FACTS nicht sämtliche Herausforderungen einer ökonomischen Bewertung. Die folgenden Punkte deuten weitere Verbesserungsmöglichkeiten an, die die Basis für zukünftige Forschungsarbeiten bieten:

- Bei den verwendeten FPs handelt es sich um eine manuelle Bewertungsmethodik. Diese ermöglicht eine detaillierte Erfassung der Aufwände auf Komponentenebene, ist allerdings mit einem für größere Projekte nicht unerheblichen Zeitaufwand verbunden. Wünschenswert wäre eine Methodik, die die Aufwandsbestimmung zumindest teilweise automatisiert, ohne den Detailgrad zu verringern. Derzeit ist eine solche Methode nicht verfügbar.

- Die Verwendung von FPs führt dazu, dass einige Bereiche nicht direkt mit der gleichen Genauigkeit erfasst werden können. Dazu zählen bspw. die Kosten für die organisatorischen Veränderungen in Verbindung mit der Einführung einer SWPL oder die zusätzlichen Erträge (stärkere Kundenbindung, höhere Qualität). Zwar wurden im Rahmen der Arbeit Ideen aufgezeigt, wie die relevanten Kosten und Erträge berücksichtigt werden können. Detailliertere Ansätze wären hier allerdings sinnvoll.

- Die Anzahl untersuchter Projekte für eine SWPL Datenbasis ist immer noch gering. Ebenso unterscheiden sich die Detailgrade der einzelnen Projekte. Das e-FACTS

Modell unterstützt durch seine einfache und schnelle Anwendung eine breite und einheitliche Verwendung. Dadurch besteht die Möglichkeit, eine umfassende Datenbank für SWPL mit einem Detailgrad bis auf Komponentenebene zu erstellen. Eine solche Datenbank würde viele neue Analysen und Erkenntnisse zu Vorteilen von SWPL gegenüber der traditionellen SW Entwicklung ermöglichen. Zusätzlich wäre auf Basis einer solchen Datenbank in Zukunft auch eine wissenschaftliche Validierung des Ansatzes möglich.

Teil C. Anhang

Funktionsbezeichnung	EI			EO			EQ			AL			DF			EIF			
	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	
																Total FP unadjusted	=	311	
Funktion																			FP/ Funktion
Zugang	1			1							1		2			1			25
Personalisierung	1			1						1			1			1			19
Alarmanlage	1			2						1						1			18
Wegfahrsperre	1				1						1		1					1	26
Homecoming Light	1			1						1			1			1			19
Höchstgeschwindigkeit	3			2						1			1			1			29
Fensteröffnung	1			1						1						1			14
Starten von Anwendungen	1			1			1			1						1			17
GSM																			
>Kommunikation											1	1	1			1	1		25
>Standortbestimmung	1			1						1			1						14
>Stilllegung	1			2							1		1			1			24
>Aktionsradius	2			2						1	1		1						24
WLAN																			
>Kommunikation											1		1				1		15
>Update von Daten		1		1						1			1			1			20
>Konfiguration		1		1						1				1		1			22

Tabelle 50 – FP Übersicht Zugangssysteme ohne Wiederverwendung

Nr.	Featurebezeichnung	FP \WV	EI			EO			EQ			AL			DF			EIF				
			L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H		
																			=	415		
													Total FP(resuse) unadjusted								FP/ Fkt.	Veränd.
1	**Zugang**	25		1			1						1		1	1			1		31	24%
2	**Personalisierung**	19		1			1					1			1				1		23	21%
3	**Alarmanlage**	18		1		2						1							1		21	17%
4	**Wegfahrsperre**	26		1			1						1			1				1	29	12%
5	**Homecoming Light**	19	1			1						1				1			1		23	21%
6	**Höchstgeschwindigkeit**	29	2	1		2						1			1				1		32	10%
7	**Fensteröffnung**	14		1		1						1							1		17	21%
8	**Starten von Anwendungen**	17	1				1			1		1	1		1					1	32	88%
9	**GSM**	87																			117	34%
10	**>Kommunikation**	25											1	1	1	1			1	1	37	48%
11	**>Standortbestimmung**	14	1			1						1			2						19	36%
12	**>Stilllegung**	24	1			2							1		2				1		31	29%
13	**>Aktionsradius**	24	1	1		2						1	1		2						30	25%
14	**WLAN**	57																			90	58%
15	**>Kommunikation**	15											1		1	1				1	25	67%
16	**>Update von Daten**	20			1	1							1		2				1		30	50%
17	**>Konfiguration**	22			1	1							1		1		1		1		35	59%

Tabelle 51 – FP Übersicht Zugangssysteme mit Wiederverwendung ohne Architektur

Nr.	Featurebezeichnung	FP ohne WV	EI L	EI M	EI H	EO L	EO M	EO H	EQ L	EQ M	EQ H	AL L	AL M	AL H	DF L	DF M	DF H	EIF L	EIF M	EIF H	FP/ Fkt.	Veränd.
													Total FP(resuse) unadjusted							=	383	
1	Zugang	25	1			1							1		1	1					22	-12%
2	Personalisierung	19	1			1						1			1						14	-26%
3	Alarmanlage	18	1			2						1									13	-28%
4	Wegfahrsperre	26	1				1						1			1					18	-31%
5	Homecoming Light	19	1			1						1				1					16	-16%
6	Höchstgeschwindigkeit	29	3			2						1			1						24	-17%
7	Fensteröffnung	14	1			1						1									9	-36%
8	Starten von Anwendungen	17	1			1				1		1	1		1						21	24%
9	GSM	87																			102	17%
10	>Kommunikation	25											1	1	1	1				1	30	20%
11	>Standortbestimmung	14	1			1						1			2						19	36%
12	>Stilllegung	24	1			2							1		2						24	0%
13	>Aktionsradius	24	2			2						1	1		2						29	21%
14	WLAN	57																			72	26%
15	>Kommunikation	15											1		1	1				1	25	67%
16	>Update von Daten	20		1		1							1		2						21	5%
17	>Konfiguration	22		1		1							1		1		1				26	18%
18	Architektur	n/a			1			1						1				8	2		72	n/a

Tabelle 52 – FP Übersicht Zugangssysteme mit Wiederverwendung mit Architektur

Kate	Frage obligatorisch optional (freiwillig)	Einstufungen 1	2	3	4	5
Produktangebot	Wieviele unterschiedliche Produkte bieten sie aktuell am Markt an?	1 bis 4	5 bis 8	9 bis 15	16 bis 30	mehr als 30
	Können sie bereits abschätzen, wieviele ihrer Produkte zumindest teilweise durch die SWPL abgedeckt werden sollen?	unter 5%	5 bis 10%	über 10 bis 20%	über 20 bis 50%	mehr als 50%
	Wie viele neue Produkte planen sie in den nächsten 2 Jahren auf den Markt zu bringen?	keins	1 bis 2	3 bis 5	6 bis 10	mehr als 10
	Wie groß ist der Grad der Veränderung (Neuigkeit) je Generation oder neuem Produkt, wobei wesentliche Features stärker berücksichtigt werden sollten als kosmetische Veränderungen?	Jede Generation ist fast eine komplette Neuentwicklung.	Einige wenige grundlegende Komponenten können weiterverwendet werden; größtenteils jedoch neu.	Neuentwicklung und Verwendung bestehender Komponenten hält sich ungefähr das Gleichgewicht	Verändeurng ist eher als eine Evolution und nicht als eine Revolution zu verstehen, es finden sinnvolle Erweiterungen in geringem Umfang statt (zwischen 20 und 50%)	Nur sehr geringe Veränderungen (weniger als 20% der Komponenten betroffen), hauptsächlich Anpassungen, keine komplette Neuentwicklung.
	Wenn sie die maximale Wiederverwendungsrate abschätzen sollten, wie weit wäre eine Standardisierung von Bestandteilen aufgrund der Marktgegebeneheiten wahrscheinlich möglich?	Bei jedem Produkt handelt es sich weitgehend um eine Einzelentwicklung. Daher wird es quasi keine gleiche Basis geben können.	Die Produkte sind alle eigenständig, haben jedoch eine kleine gemeinsame Basis; dies wird sich aufgrund der sehr spezifischen Kundenanforderungen in Zukunft auch nicht ändern.	Es wird einen realtiv großen gemeinsamen Kern bei den Produkten der Produktfamilie geben, aber auch einen signifikanten Teil unterschiedlicher Funktionen je Produkt.	Zum Kunden hin werden sich die Produkte entsprechend der Positionierung teilweise deutlich unterscheiden, aber technisch sind sie weitgehend identisch	Die gemeinsame Basis (Core Assets) der Produktfamilie wird ein limitiertes Set enthalten, das alle Produkte abdecken wird. Der individuelle Anteil der Produkte wird (bis auf wenige Ausnahmefälle) sehr gering.
Produktentwicklung	Haben sie schon mal ein CMMI Assessment durchführen lassen? Wenn ja, welches Level haben sie erreicht? (Ansonsten auch vergleichbarer Wert, wie z.B. SPICE; oder auch Selbsteinschätzung)	CMMI Level 1 Keine festgelegten Prozesse, Vorgehen wird häufig adhoc abgestimmt	CMMI Level 2 Einige wenige Prozesse sind standardisiert, vielfach aber nicht genau kontrolliert	CMMI Level 3 ausreichendes Prozessmanagement etabliert	CMMI Level 4 alle Aktivitäten sind über standardisierte Prozesse abgedeckt, die klar kommuniziert und auch kontrolliert werden	CMMI Level 5 wie Level 4 zusätzlich ergänzt um ein exzellentes Verbesserungs-management
	Wie werden bei ihnen die Produktrequirements festgelegt?	Die Requirements werden vom Marketing vorgegeben, ohne dass es dafür ein standardisiertes Verfahren geben würde. Die Entwicklungsabteilung wird weitgehend mit fertigen high-level Requirements konfrontiert.	Marketing entwickelt die Requirements, stimmt diese aber nachträglich mit der Entwicklung ab; kein Standardprozess etabliert	Es gibt einen Standardprozess zur Aufstellung der Requirements; hauptsächlich eine Abteilung ist mit der Erstellung alleine beauftragt	Wie bei 3 allerdings sind mehrere Abteilungen in der Entwicklung der Requirements involviert. Aufgrund der dezentralen Struktur vieler Automobilunternehmen ist daher zusätzlich eine Standortunabhängigkeit der Prozesse notwendig.	Über einen standardisierten, standortübergreifenden Prozess werden die Requirements frühzeitig mit allen Stakeholdern abgestimmt. Spätere Anpassungen treten dadurch nur sehr selten auf.
	Wie hoch ist die Qualität, die sie derzeit bei der SW Erstellung mit ihren aktuellen Prozessen erreichen?	Es findet keine systematische Qualitätskontrolle statt. KPIs werden nicht ermittelt	Es erfolgt eine minimale Qualitätskontrolle, bei der aber zahlreiche wichtige Elemente fehlen oder verbessert werden können. Es können zumindest 2 der genannten 3 KPIs ermittelt werden, die allerdings unterhalb der Durchschnittswerte liegen.	Es erfolgt eine durchschnittliche Qualitätskontrolle und die relevanten KPIs liegen im Durchschnitt (Fehlerbeseitigung ca. 90%, Kosten Fehlerbeseitigung ca. 40% und Zeitaufwand ca. 30%)	Die Qualitätskontrolle berücksichtigt die wichtigsten Elemente, könnte aber noch weiter verbessert werden. Die aktuellen KPIs liegen aber bereits deutlich über dem Durchschnitt.	Die Qualitätskontrolle berücksichtigt alle wichtigen Elemente, wie z.B. Tests durch spezialisierte Mitarbeiter. Und die KPIs liegen deutlich über dem Durchschnitt.
Organisation	Wie viele Personen (gemesse als FTEs) arbeiten bei ihnen intern in der Software-Entwicklungsabteilung?	Weniger als 5	Zwischen 5 und 10	Mehr als 10, aber maximal 20	Mehr als 20, aber maximal 50	Mehr als 50
	Wie würden sie die aktuelle Struktur bei der Produktentwicklung am ehesten beschreiben?	Reine Projektorientierung; es gibt keine oder kaum Schnittstellen zwischen den einzelnen Projekten oder übergreifende Teams	Projektorientierung mit vereinzelten übergreifenden Teams; ebenso gibt es mehrer Schnittstellen zwischen den Projektteams	Intensive Zusammenarbeit einzelner Teams etabliert; für einige Komponenten ist auch bereits ein projektunabhängiges Team eingesetzt (Featureorientierung)	Featureorientierung weitgehend umgesetzt	Reine Featureorientierung bei der Produkt-/ Komponentenentwicklung vollständig etabliert
	Wie einfach ist es, in ihrem Unternehmen Verantwortungen/Strukturen zu ändern?	äußerst schwierig bis unmöglich; es ist mit unüberbrückbaren Widerständen zu rechnen	schwierig; die Erfahrung zeigt, dass an diesem Thema schon des öfteren Projekte gescheitert sind	durchaus möglich, aber mit erheblichen Widerständen ist zu rechnen, die jedoch in der Vergangenheit immer überbrückt werden konnten	relativ leicht, obwohl natürlich nicht immer alle der selben Meinung sind. Neuerungen werden gründlich geprüft, statt sie grundsätzlich abzulehnen	ziemlich einfach, da eine Kultur der Veränderung gelebt wird und Veränderung nicht grundsätzlich als etwas schlechtes gesehen wird.
	Wird die Entwicklung einer SWPL intern grundsätzlich unterstützt oder gibt es möglicherweise Gegner?	Es gibt zahlreiche Gegner, insbesondere auch in höheren Managementebenen	Es gibt zahlreiche Gegner, was aber vielfach auf zu geringe Information zurückzuführen ist	Es gibt einige Gegner, aber zumindest das obere Management unterstützt das Vorhaben komplett	alle Managementebenen unterstützen das Vorhaben, und auch sonst gibt es nur geringe Kritik, die wahrscheinlich durch zusätzliche Information gelöst werden kann	Komplett unterstützt, es gibt keine Gegner
	Haben sie bereits außerhalb der SW-Entwicklung eine Modularisierung oder Plattform-Strategie verfolgt?	Eine Modularisierungs- und/oder Plattformstrategie wurde geplant/umgesetzt. Diese scheiterte jedoch und wird nicht weiter verfolgt.	Eine Modularisierungs- und/oder Plattformstrategie wurde bisher nicht geplant/umgesetzt.	Eine Modularisierungs- und/oder Plattformstrategie wurde für einen kleinen Bereich bereits erfolgreich umgesetzt.	Eine Modularisierungs- und/oder Plattformstrategie wurde für mehrere Bereiche bereits erfolgreich umgesetzt. Die Strategie wird bald/aktuell auf größere Komponenten ausgeweitet.	Wichtige HW-Komponenten sind heute bereits als Produktlinie gestaltet. Es wurden signifikante Erfolge damit erzielt und die Strategie wird weiter verfolgt

Abbildung 41 – Basisfragebogen Teil 1

Architektur	Wie hoch ist der Anteil von Standardkomponenten im Softwarebereich bei heutigen Produkten?	Sehr gering	gering	mittel	hoch	sehr hoch
	Benutzen sie bereits heute eine standardisierte produktübergreifende Softwarearchitektur?	Wir haben es mal probiert, leider nicht erfolgreich	Nein	Für einige ausgewählte Komponenten im Einsatz	Ja, in weiten Teilen	Ja, komplett
	Wieviel Aufwand ist heute mit der Integration eines neuen Features durchschnittlich verbunden?	Fast alle neuen Features führen zu umfangreichen Änderungen an zahlreichen Stellen im Code; eine Aufwandsschätzung ist kaum möglich		Einige Features erzeugen hohen Aufwand, insbesondere, wenn diese nicht vorherzusehen gewesen sind; der Großteil ist jedoch einfach umsetzbar		Umsetzung ist in den meisten Fällen sehr einfach aufgrund der offenen Architektur und der guten Planung
Business Drivers	Welche Ziele verfolgt das Unternehmen maßgeblich, insbesondere im Hinblick auf die Software Entwicklung?	Es sind keine langfristigen Ziele festgelegt. Bei der individuellen Produktentscheidung bilden eher kurzfristige oder/und produktindividuelle Überlegungen die Entscheidungsbasis.		Das Unternehmen hat eine klare Ausrichtung auf eine der drei möglichen Zielrichtungen, Innovation, Kosten oder Qualität, die auch weitgehend verfolgt wird.		Das Unternehmen hat eine klare Ausrichtung auf eine der drei möglichen Zielrichtungen, Innovation, Kosten oder Qualität. Diese wird konsequent bei allen Projekten/Produkten verfolgt.
	Ist es realistisch, dass der Markt eine Verzögerung der Markteinführung ihrer nächsten Produktgeneration um ca. 6 Monate akzeptiert?	So gut wie ausgeschlossen. Der Wettbewerbsdruck ist so hoch, dass wir uns eine solche Verzögerung nicht leisten können.		Der Markt verändert sich schnell, daher wird eine solche Verzögerung zu erheblichen kurzfristigen Umsatzeinbußen führen, wäre jedoch möglich		Eine Verschiebung der neuen Produkte ist durchaus möglich. Die zu erwartenden Einbußen halten sich in Grenzen.
	Hat ihr Unternehmen in letzter Zeit zahlreiche andere Software Unternehmen aus dem Bereich, für den die SWPL geplant ist, gekauft und integriert, oder sind solche Käufe geplant?	Es wurden in den letzten Jahren zahlreiche Fremdfirmen in das Unternehmen integriert. Und auch für die Zukunft ist dies geplant. Dieses Vorgehen hat natürlich zu zahlreichen Veränderungen der Produktlandschaft geführt, die bis heute auch noch nicht vollständig abgeschlossen sind.	Es wurden in den letzten Jahren zahlreiche Fremdfirmen erfolgreich in das Unternehmen integriert. In Zukunft wird dies weiterhin geschehen, allerdings mit reduzierter Geschwindigkeit. Die wichtigsten Lücken im Produktangebot sind geschlossen. Für eine mögliche Erweiterung des Angebots ist geplant, sich weiterhin eher auf Zukäufe als komplette Eigenentwicklungen zu konzentrieren.	Es wurden in den letzten Jahren einige Firmen gekauft und erfolgreich integriert. Der Grund war immer eine sinnvolle Erweiterung der Produktpalette, wobei es sich meist lediglich um Ergänzungen des Angebots handelt.	Es wurden in den letzten Jahren einige wenige Firmen gekauft und erfolgreich integriert. Der Grund war immer eine sinnvolle Erweiterung der Produktpalette. Allerdings handelt es sich dabei (auch in Zukunft) immer um Randprodukte.	Das Unternehmen wächst weitgehend aus eigener Kraft. Wenn es Zukäufe gegeben hat oder geben wird, dann lediglich um wichtige vereinzelte Lücken im Produktangebot zu schließen. Zukäufe sind auf jeden Fall die Ausnahme.
	Wie stellen sie die benötigte Software heute her?	Das Unternehmen stellt selbst gar keine Software her. Lediglich die grundlegenden Requirements und das Integrations-Testing wird vorgegeben bzw. durchgeführt.	Das Unternehmen kauft die Softwarebestandteile so gut wie ausschließlich zu. Nur ein sehr geringer Anteil der Arbeit wird intern erledigt. Dabei handelt es sich hauptsächlich um Testing und Integration. Requirements werden lediglich sehr grob vorgegeben.	Es werden alle verfügbaren Ressourcen genutzt. Eigenentwicklung dort, wo der meiste Nutzen generiert wird, allerdings fehlen klare Regelungen wie hier vorzugehen ist. Teilweise werden Entscheidungen adhoc getroffen. Als Minimum werden aussagekräftige Requirements erstellt und das Componenten- und Integrationstesting durchgeführt.	Wie bei 3 allerdings bestehen klare Regelungen, wie in einem bestimmten Fall vorzugehen ist. Lediglich in begründeten Ausnahmefällen darf von dieser Regelung abgewichen werden.	Als Kombination aus allen möglichen Quellen (COTS, mit Hilfe von externen Kontraktern, Eigenentwicklung). Der Schwerpunkt der eigenen Arbeit liegt dabei auf den wertschöpfenden Tätigkeiten und den Bereichen, in denen das Unternehmen über das notwendige Know-how verfügt. (Requirements Engineering, SW Entwicklung für Kernkompetenzen und das komplette Testing)

Abschließende Bewertung

Empfehlung zur geplanten Realisierung eines Produktfamilienansatzes

Abbildung 42 – Basisfragebogen Teil 2

Literaturverzeichnis

[Absa05] Dossier Automobilmarketing: Emotionen oder Rabatte verkaufen? In: Absatzwirtschaft, 47. Jg., Nr. 9, 2005, S. 42-47.

[ADAC09] ADAC: ADAC Pannenstatistik 2008. April 2009, *http://www1.adac.de/images/25797_tcm8-250130.pdf* [05.11.2009]

[Adam98] D. Adam: Produktions-Management. 9. Auflage, Gabler, Wiesbaden, 1998.

[Airb10] Airbus: Maximum communality across six aircraft variants. *http://www.airbus.com/en/aircraftfamilies/a330a340/commonality* [11.11.2010]

[Anas01] M. Anastasopoulos, C. Gacek: Implementing Product Line Variabilities. In: ACM SIGSOFT Software Engineering Notes, Vol. 26, No. 3 (Mai 2001), ACM, New York, 2001, S. 109-117.

[Andr76] K. Andritzky: Die Operationalisierbarkeit von Theorien zum Konsumentenverhalten. Duncker & Humblot, Berlin, 1976.

[Atki02] C. Atkinson, et. al.: Component-Based Product Line Engineering with UML. Pearson Education Limited, Harlow, London, 2002.

[Back99] K. Backhaus: Industriegütermarketing. 6. Auflage, Vahlen, München, 1999.

[Bamb08] G. Bamberg, A.G. Coenenberg, M. Krapp: Betriebswirtschaftliche Entscheidungslehre. 14. Auflage, Vahlen, München, 2008.

[Barc09] A. Barczok, R. Opitz: Mobile Trümpfe: Smartphone-Betriebssysteme im Vergleich. In: c't Magazin für Computer Technik, Heft 23/2009, 2009, S. 86-97.

[Bart09] J. Bartholdt, M. Medak, R. Oberhauser: Integrating Quality Modeling with Feature Modeling in Software Product Lines. In: Proceedings of the Fourth International Conference on Software Engineering Advances, ICSEA, Porto, 2009, S. 265-370.

[Baye99] J. Bayer, et al.: PuLSE: A Methodology to Develop Software Product Lines. In: Proceedings of the Fifth ACM SIGSOFT Symposium on Software Reusability (SSR '99), Los Angeles, 1999, S. 122-131.

[Baye04a] J. Bayer, S. Kettemann, D. Muthig: Principles of Software Product Lines and Process Variants. PESOA Report No. 03/2004, Kaiserslautern, 2004.

[Baye04b] J. Bayer, et al.: Architekturentwicklung, basierend auf existierenden Systemen. In: G. Böckle et al. (Hrsg.), Software-Produktlinien: Methoden, Einführung und Praxis, dpunkt Verlag, Heidelberg, 2004, S. 165-176.

[Beis06] M. Beise: Die Lead-Markt-Strategie: Das Geheimnis weltweit erfolgreicher Innovationen. Springer, Berlin, 2006.

[Best05] E. Best, M. Weth: Geschäftsprozesse optimieren: Der Praxisleitfaden für erfolgreiche Reorganisation. 2. Auflage, Gabler, Wiesbaden, 2005.

[Bidl64] J. Bidlingmaier: Unternehmerziele und Unternehmerstrategien. Gabler, Wiesbaden, 1964.

[Böck04a] G. Böckle, E. Kamsties, K. Pohl: Einleitung. In: G. Böckle et al. (Hrsg.), Software-Produktlinien: Methoden, Einführung und Praxis, dpunkt Verlag, Heidelberg, 2004, S. 3-11.

[Böck04b] G. Böckle: Organisationsstrukturen. In: G. Böckle et al. (Hrsg.), Software-Produktlinien: Methoden, Einführung und Praxis, dpunkt Verlag, Heidelberg, 2004, S. 25-40.

[Boeh81] B. Boehm: Software Engineering Economics. Prentice Hall, Englewood Cliffs, USA, 1981.

[Boeh00] B. Boehm, et al.: Software Cost Estimation with COCOMO II. Prentice Hall, Upper Saddle River, USA, 2000.

[Boeh04] B. Boehm, et al.: A Software Product Line Life Cycle Cost Estimation Model. In: Proceedings of the 2004 International Symposium on Empirical Software Engineering, IEEE Computer Society, Los Alamitos, 2004, S. 156-164.

[Booc98] G. Booch, J. Rumbaugh, I. Jacobson: The Unified Modeling Language User Guide. Addison-Wesley, Boston, et al., 1998.

[Bosc00] J. Bosch: Design and Use of Software Architectures – Adopting and evolving a product-line approach. Addison-Wesley Longman, Amsterdam, 2000.

[Both97] D. Both: A Short History of OS/2. 1997. *http://www.os2bbs.com/OS2News/OS2History.html* [05.11.2009]

[Bour95] R. Bourgonjon: The Evolution of Embedded Software in Consumer Products. In: Proceedings of the International Conference on Engineering of Complex Computer Systems (ICECCS'95), Ft. Lauderdale, USA, 1995.

[Broo95] F. P. Brooks Jr.: The Mythical Man-Month: Essays on Software Engineering. Anniversary Edition, Addison-Wesley, Boston, et al., 1995.

[Broy98a] M. Broy, M. von der Beek, I. Krüger: Problemanalyse für ein Großverbundprojekt: Systemtechnik Automobil-Software für eingebettete Systeme. SOFTBED-Abschlussbericht, Technische Universität München, München, 1998.

[Broy98b] M. Broy: Informatik: Eine grundlegende Einführung. 1. Band, 2. Auflage, Springer, Berlin u.a., 1998.

[Broy98c] M. Broy: Informatik: Eine grundlegende Einführung. 2. Band, 2. Auflage, Springer, Berlin u.a., 1998.

[Broy05] M. Broy, F. Deißenböck, M. Pizka: A Holistic Approach to Software Quality at Work. Proceedings of the 3rd World Conference for Software Quality (3WCSQ), München, 2005.

[Brug05] R. Brugger: Der IT Business Case: Kosten ermitteln und analysieren, Nutzen erkennen und quantifizieren, Wirtschaftlichkeit nachweisen und realisieren. Springer, Berlin, 2005.

[Bryl99] M. Brylawski: Uncommon Knowledge: Automotive Platform Sharing's Potential Impact on Advanced Technologies, Conference pre-print for the 1st International Society for the Advancement of Material and Process

Engineering (SAMPE) Automotive Conference, 27-29. September, Detroit, USA, 1999.

[Bühn04] S. Bühne, et al: Variabilität in Software-Produktlinien. In: G. Böckle et al. (Hrsg.), Software-Produktlinien: Methoden, Einführung und Praxis, dpunkt Verlag, Heidelberg, 2004, S. 13-24.

[Buhr09] K. Buhr, R. T. Kolagari: Softwarebasierte Produktlinien – Szenarien für Automobilhersteller und Zulieferer. *http://pi.informatik.uni-siegen.de/stt/24_4/03_Technische_Beitraege/buhr.pdf* [14.01.2009]

[Cana08] Canalys – expert analysis for the high-tech industry: Global smart phone shipments rise 28% - Nokia retains lead, but Apple moves into number two position. *http://www.canalys.com/pr/2008/r2008112.htm* [21.10.2009]

[Care02] J. O. Carey, B. Carlson: Framework Process Patterns: Lessons Learned Developing Application Frameworks. Addison-Wesley Professional, Boston u.a., 2002.

[Case05] P. Caselitz, D. Lehmkuhl: Erhöhung der Zuverlässigkeit und Sicherheit elektrischer Fahrzeugsysteme durch reproduzierbare Untersuchungen und Tests mit virtuellen Battrien. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 661-671.

[Cent09] Center for Systems and Software Engineering: COCOMO Suite. *http://csse.usc.edu/csse/research/COCOMO_suite/* [05.11.2009]

[Chen91] P.S. Chen, H.-D. Knöll: Der Entity-Relationship-Ansatz zum logischen Systementwurf. B.I.-Wissenschaftsverlag, Mannheim, 1991.

[Chip08] Chip: 30 Jahre Hightech. In: Chip, 05/2008, S. 26 – 27.

[Clar95] K. Clark, S. Wheelwright: Leading Product Development. Free Press, New York, 1995.

[Clem02] P. Clements, L. Northrop: Software Product Lines: Practices & Patterns. SEI Series in Software Engineering, Addison-Wesley, Boston u.a., 2002.

[Clem03] P. Clements: It Takes Two. In: news@sei, Vol. 6, No. 4, 4. Quartal 2003. *http://www.sei.cmu.edu/news-at-sei/columns/software-product-lines/2003/4q03/software-product-lines-4q03.htm* [05.11.2009]

[Clem05] P.C. Clements, J.D. McGregor, S.G. Cohen: The Structured Intuitive Model for Product Line Economics (SIMPLE). Technical Report, CMU/SEI-2005-TR-003, Software Engineering Institute, Pittsburgh, 2005.

[Coen99] A. G. Coenenberg: Kostenrechnung und Kostenanalyse, 4. Auflage, Verlag Moderne Industrie, Landsberg am Lech, 1999.

[Cohe03] S. Cohen: Predicting when Product Line Investment Pays. Technical Note CMU/SEI-2003-TN-017, Carnegie Mellon University, Pittsburgh, 2003. *http://www.sei.cmu.edu/reports/03tn017.pdf* [06.11.2009]

[Conn90] M.J. Connolley: An Empirical Study of Function Points Analysis Reliability. Massachusetts Institute of Technology, 1990.

[Cook05] J. Cook: Motorola and Skyworks Sign Development Agreement for the Helios DigRF Radio Solution for Next-Generation EDGE Platforms, mobility today, veröffentlicht am 28.11.2005. *http://mobilitytoday.com/news/005683/mobile_Motorola_skyworks_Helios_Edge* [20.10.2009]

[Coop03] J. W. Cooper: C# Design Patterns: A Tutorial. Addison-Wesley, Boston u.a., 2003.

[Copl98] J. Coplien, D. Hoffman, D. Weiss: Communality and Variability in Software Engineering. In: IEEE Software, Vol. 15, No. 6, November/Dezember, IEEE Press, 1998, S. 37-45.

[Cosm09a] The Common Software Measurement International Consortium: COSMIC Functional Size Measurement Method Version 3.0.1, Measurement Manual, The COSMIC Implementation Guide for ISO/IEC 19761:2003. Montreal, 2009, *http://www.cosmicon.com/portal/public/COSMIC Method v3.0.1 Measurement Manual.pdf* [05.11.2009]

[Cosm09b] The Common Software Measurement International Consortium: COSMIC Functional Size Measurement Method Version 3.0.1, Documentation

Overview and Glossary of Terms. Montreal, 2009, *http://www.cosmicon.com/portal/public/COSMIC Method v3.0.1 Documentation Overview.pdf* [05.11.2009]

[Cosm09c] The Common Software Measurement International Consortium: Advantages of the COSMIC method. *http://www.cosmicon.com/advantagecs.asp* [21.10.2009]

[Cusu91] M. A. Cusumano, Y. Mylonadis, R. Rosenbloom: Strategic Maneuvering and Mass-Market Dynamics: The Triumph of VHS over Beta. International Center for Research on the Management of Technology, Working Paper 40-91, Massachusetts Institute of Technology, Cambridge, USA, 1991.

[Cusu98] M. A. Cusumano, K. Nobeoka: Thinking Beyond Lean: How Multi-Project Management is Transforming Product Development at Toyota and Other Companies. The Free Pres, New York, 1998.

[Czar00] K. Czarnecki, U. Eisenecker: Generative Programming — Methods, Tools and Applications. Addison-Wesley, Boston u.a., 2000.

[Dash01] E. M. Dashofy, A. van der Hoek, R. N. Taylor: A highly extensible, xml-based architecture description language. In: Working IEEE/IFIP Conference on Software Architecture (WICSA'01), Amsterdam, IEEE CS, 2001, S. 103-112.

[Davi87] S. M. Davis: Future Perfect. Addison-Wesley, Boston, 1987.

[DIN89] DIN Deutsches Institut für Normung e.V.: DIN 19250: Grundlegende Sicherheitsbetrachtungen für MSR-Schutzeinrichtungen, 1989.

[Ding09] U. Dinger: Integrierte und hybride Konstruktion von Software-Produktlinien. Dissertation an der technischen Universität Dresden, 2009.

[Dude05] F. Düdenhöffer, M. Krüger: Gemeinsam gegen Toyota? Deutsche Automobile im Qualitätsbund. In: QZ-Qualitätsmanagement in Industrie und Dienstleistung, Heft 10, 2005, S. 19-21.

[Eber97] C. Ebert: Dealing with Nonfunctional Requirements in Large Software Systems. In: Annals of Software Engineering, Vol. 3, Nr. 1, S. 367-395.

[Eber03] C. Ebert, M. Smouts: Tricks and Traps of Initiating a Product Line Concept in Existing Products. In: Proceedings of the 25th International Conference on Software Engineering, Portland, IEEE Computer Society, Washington, 2003, S. 520-525.

[Fent97] N. E. Fenton, S. L. Pfleeger: Software Metrics: A Rigorous & Practical Approach, 2. Auflage, PWS Publishing Company, Boston u.a., 1997.

[Fert09] Fertighaus.de: Es gibt keine Fertighäuser von der Stange. *http://www.fertighaus.de/nxs/2286///fertighaus/schablone1/Es-gibt-keine-Fertighaeuser-von-der-Stange* [21.10.2009]

[Fetk99] T. Fetke: The Warehouse Software Portfolio: a case study in functional size measurement. Technical report no. 1999-20, Software Engineering Management Research Laboratory, Univerité du Québec à Montréal, Canada, 1999.

[Fina10] Financial Times Deutschland Online: Fehleranalyse: Nokia stellt sich unverblümter Selbstkritik. Onlineartikel vom 16.09.2010. *http://www.ftd.de/it-medien/it-telekommunikation/:fehleranalyse-nokia-stellt-sich-unverbluemter-selbstkritik/50170745.html* (Abruf 26.11.2010)

[Frau10] Fraunhofer Institut: Referenzprojekt: Automobilhersteller BMW AG. *http://cobra.fraunhofer.de/deutsch/referenzen/deutschland/bmw/index.html* [18.11.2010]

[From05] P. Froment: A380 Avionics Architecture: The Solutions and Benefits. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 11-17.

[Gaff89] J. E. Gaffney Jr., T.A. Durek: Software Reuse – key to enhanced productivity: some quantitative models. In: Information and software technology, Vol. 31, No. 5, Juni 1989, S. 258-267.

[Gamm95] E. Gamma, et. al.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Boston u.a., 1995.

[Gand95] N. Gandal, S. Greenstein, D. Salant: Adoptions and Orphans in the Early Microcomputer Market, 1995, *http://129.3.20.41/eps/io/papers/9502/9502002.pdf* [20.10.2009]

[Geru08] M. Gerush: App Dev Leaders – Got Metrics? Forrester Research, 2008.

[Göth05] J. Göthel, P. Kohlschmidt: Consumer Electronics am Beispiel der Bluetooth-Schnittstelle. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 251-262.

[Goma04] H. Gomaa: Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures. Addison-Wesley, Boston, 2004.

[Grob06] H. L. Grob: Einführung in die Investitionsrechnung, 5. Auflage, Vahlen, München, 2006.

[Grot05] C. Grote, D. Weyl, R. Rau: Integration mobiler Endgeräte – Problemstellungen und mögliche Lösungsszenarien. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 263-276.

[Halm02] G. Halmans, K. Pohl: Modellierung der Variabilität einer Software Produktfamilie. In: Modellierung 2002, Modellierung in der Praxis – Modellierung für die Praxis, Proceedings der Arbeitstagung der GI, Vol. 12, 25.-27. März 2002, Tutzing, S. 63-74.

[Hand06] Handelsblatt online: Keine Gleichmacherei: Was "Plattformstrategie" bei Autos bedeutet. 10.03.2006. *http://www.handelsblatt.com/technologie/technik/was-plattformstrategie-bei-autos-bedeutet;1059943* [07.11.2009]

[Haus05] F. Hauser, et al.: Mehrstufiges Qualitätskonzept und neue Prüfverfahren zur Absicherung der Entwicklungsqualität komplexer Steuergeräte für Kraftfahrzeuge. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 163-174.

[Hein02] A. Hein, P. Lutz, S. Thiel: Systematisches Entwickeln softwareintensiver Produkte. In: Bosch-Zünder, Forschung-Entwicklung-Technik, Oktober/November 2002, S. 20.

[Hein04] H. Heinecke, et al.: AUTomotive Open System ARchitecture – An industry-wide initiative to manage the complexity of emerging Automotive E/E-Architectures Convergence 2004. International Congress on Transportation Electronics, Detroit, USA, 2004.

[Hoch00] D. J. Hoch, et al.: Erfolgreiche Software-Unternehmen. Die Spielregeln der New Economy. Hanser, München, Wien, 2000.

[Hörn05] H. Hörner, A. Raisch, O. Meili: Basis-Software-Komponenten in AUTOSAR – ein solides Fundament. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 409-419.

[Hütt08] M. Hüttenrauch, M. Baum: Effiziente Vielfalt. Die dritte Revolution in der Automobilindustrie. Springer, Berlin, Heidelberg, 2008.

[IEC05] IEC International Electrotechnical Commission: Functional Safety and IEC 61508, 2005. *http://www.iec.ch/zone/fsafety/pdf_safe/hld.pdf* [05.11.2009]

[IFPU09] International Function Point Users Group (IFPUG): IFPUG Counting Practices Manual. Release 4.3, Princeton Junction, USA, 2009.

[ISO98] ISO/IEC International Organization for Standardization/International Electrotechnical Commission: ISO/IEC 15504-1: Information technology – Software process assessment – Concepts and introductory guide. 1998.

[ISO01] ISO/IEC International Organization for Standardization/International Electrotechnical Commission: ISO/IEC 9126: International Standard, Software Engineering – Product quality – Part 1: Quality model. 2001.

[ISO03a] ISO/IEC International Organization for Standardization/International Electrotechnical Commission: ISO/IEC 20926:2003 Software engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting practices manual. 2003.

[ISO03b] ISO/IEC International Organization for Standardization/International Electrotechnical Commission: ISO/IEC 19761:2003 Software engineering – COSMIC-FFP – A functional size measurement method. 2003.

[Jack09] P. Jackson: The Little Book of Smart. Veloce Publishing, Dorchester, England, 2009.

[Jaco97] I. Jacobson, M. Griss, P. Jonsson: Software Reuse: Architecture, Process, and Organization for Business Success. Addison-Wesley, Boston, 1997.

[John04] I. John, J. Dörr: Requirements Engineering, basierend auf existierenden Systemen. In: G. Böckle et al. (Hrsg.), Software-Produktlinien: Methoden, Einführung und Praxis, dpunkt Verlag, Heidelberg, 2004, S. 153-163.

[Jone02] L. G. Jones: Software Process Improvement and Product Line Practice. In: P. Clements, L. Northrop, Software Product Lines: Practices and Patterns, Addison-Wesley, Reading, Massachusetts, USA, 2002, S. 388-392.

[Jone07] C. Jones: Estimating Software Costs: Bringing Realism to Estimating, 2. Auflage, McGraw-Hill, New York, et al., 2007.

[Jone08] C. Jones: Applied Software Measurement; global analysis of productivity and quality. Dritte Edition, McGraw-Hill, New York, et al., 2008.

[Jung05] M. Junge: Controlling modularer Produktfamilien in der Automobilindustrie: Entwicklung und Anwendung der Modularisierungs-Balanced-Scorecard. Dissertation an der Universität Mainz 2004, Gabler, Wiesbaden, 2005.

[Jurr09] N. Jurran: High Noon in Blau: Acht Blue-ray-Player bis 400 Euro gegen die PS3. In: c't Magazin für Computer Technik, Heft 6/2009, 2009, S.122-131.

[Kamm04] C. Kamm, et. al.: Systematische Aufwandsschätzung für Software im Fahrzeug. In: OBJEKTspektrum, Heft 4/2004, 2004, S. 60-65.

[Kan03] S. H. Kan: Metrics and Models in Software Quality Engineering. Addison-Wesley, Massachusetts, USA, 2003.

[Keri78] R.A. Kerin, M.G. Harvey, J.T. Rothe: Cannibalism and New Product Development. In: Business Horizons, Vol. 21, No. 5, October 1978, S. 25-31.

[Kieb06] S. R. Kiebusch: Metriken für prozessorientierte Software-System-Familien – Umfangskalkulation sowie Aufwandsprognose im Electronic Business und Automobilbereich. Dissertation an der Universität Leipzig, 2006.

[Kitc96] B. Kitchenham, S.L. Pfleeger: Software Quality: The Elusive Target. In: IEEE Software, Heft 13, 1996, S. 12-21.

[Kitt09] H.-B. Kittlaus, P. N. Clough: Software Product Management and Pricing: Key Success Factors for Software Organizations. Springer, Berlin, Heidelberg, 2009.

[Knau02] P. Knauer, G. Succi: Perspectives on Software Product Lines. In: Software Engineering Notes, Vol. 27, No. 2, March 2002, S. 40-45.

[Knau04] P. Knauer: Evolution. In: G. Böckle et al. (Hrsg.), Software-Produktlinien: Methoden, Einführung und Praxis, dpunkt Verlag, Heidelberg, 2004, S. 177-189.

[Kötz05] J. Kötz: Elektronik Architekturen über Baureihen – Chancen und Risiken. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 487-494.

[Korn99] W. Korndörfer: Unternehmensführungslehre: Einführung, Entscheidungslogik, Soziale Kompetenzen. 8. Aufl., Gabler, Wiesbaden, 1999.

[Kreu05] J. Kreuzinger: Softwareentwicklung im Automobilbau: Anforderungen, Standards und Vorbilder. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 69-78.

[Kuma96] S. Kumar, et al.: The Codesign of Embedded Systems – A unified Hardware/Software Representation. Kluwer Academic Publishers, Norwell, Massachusetts, USA, 1996.

[Lair06] L. M. Laird, M. C. Brennan: Software Measurement and Estimation: A Practical Approach. Wiley Interscience, New Jersey, USA, 2006.

[Lane06] S. Lane: Software remains Apple's secret sauce of success. AppleInsider *http://www.appleinsider.com/articles/06/09/13/software_remains_apples_secret_sauce_of_success___analyst.html* [20.10.2009]

[Lava09] L. Lavazza: Convertibility of Functional Size Measurements: New Insights and Methodological Issues. Proceedings of the 5th International Conference on Predictor Models in Software Engineering, Vancouver, Canada, ACM, New York, USA, 2009.

[Lego09] The LEGO Group: LEGO Company Profile – An introduction to the LEGO group 2009. *http://cache.lego.com/downloads/aboutus/LEGO_company_profile_UK.pdf* [05.11.2009]

[Leit09] B. Leitenberger: Die Geschichte von PC-DOS. *http://www.bernd-leitenberger.de/qdos.shtml* [06.11.2009]

[Lind02] F. van der Linden: Software Product Families in Europe: The Esaps & Café Projects. In: IEEE Software, Vol. 19, No. 4, Juli/August, 2002, S. 41-49.

[Lind04] F. van der Linden, et al.: Software Product Family Evaluation. In: R. L. Nord (Hrsg.), Software Product Lines, Proceedings of the third International Conference, SPLC 2004, Boston, USA, 30. August – 2. September, Springer, Berlin, Heidelberg, 2004, S. 110-129.

[Long01] D. H. Longstreet: Function Point Training and Analysis Manual. Longstreet Consulting Inc., 2001. *http://www.softwaremetrics.com/Function%20Point%20Training%20Booklet%20New.pdf* [06.11.2009]

[Mark99] F. Markus: 2001 Chrysler PT Cruiser. In: Car and Driver, 45. Jg., Nr. 1, Juli 1999, S. 44-49.

[Mark03] J. Markmann, A. Benze: Multichannel-Management in der Automobilindustrie. In: O.Merx, C. Bachem, Multichannel Marketing, Springer, Berlin, 2003.

[Marw07] P. Marwedel: Eingebettete Systeme – eine Einführung. Springer, Berlin, 2007.

[Mati04] M. Matinlassi: Comparison of software product line architecture design methods: COPA, FAST, FORM, KobrA, and QADA. In: 26th Intern. Conference on Software Engineering (ICSE'04), Edinburgh, Scotland, IEEE CS, 2004, S. 127-136.

[McKi06] McKinsey & Company: The Race to Master Automotive Embedded Systems Development, McKinsey & Company, Düsseldorf, 2006.

[Meye92] M. Meyer, J. Utterback: The Product Family and the Dynamics of Core Capability. The International Center for Research on the Management of Technology, Working Paper 77-92, Massachusetts Institute of Technology, Cambridge, USA, 1992.

[Moor65] G. E. Moore: Cramming more components onto integrated circuits. In: Electronics, Vol. 38, No. 8, 19. April, 1965, S. 114-117.

[Mora07] T. Moran: Out of Sync. The New York Times Online Ausgabe, 8.1.2007. *http://wheels.blogs.nytimes.com/2007/01/08/out-of-sync/* [29.12.2010]

[Muth04] D. Muthig: Implementierung von Variabilität. In: G. Böckle et al. (Hrsg.), Software-Produktlinien: Methoden, Einführung und Praxis, dpunkt Verlag, Heidelberg, 2004, S. 109-118.

[Nies02] R. Nieschlag, E. Dichtl, H. Hörschgen: Marketing, 19. Auflage, Duncker & Humblot, Berlin, 2002.

[Nóbr08] J. P. Nóbrega: An Integrated Cost Model for Product Line Engineering. Informatics Center of UFPE, 2008.

http://www.rise.com.br/research/seminars/riseday20080405/riseday20080405jarley.pdf [02.11.2009]

[Pacz07] J. Paczkowski: D5 Highlights: Steve Jobs and Bill Gates together. 2007. *http://d5.allthingsd.com/20070530/steve-jobs-and-bill-gates-together-part-1-of-7/* [05.11.2009]

[Park92] R. E. Park: Software Size Measurement: A Framework for Counting Source Statements. Technical Report, CMU/SEI-92-TR-020, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1992.

[Parn76] D. L. Parnas: On the Design and Development of Program Families. In: IEEE Transations on Software Engineering, Vol. SE-2, No. 1, S. 1-9, 1976.

[Parn09] D. L. Parnas: Multi-Dimensional Software Families: Document Defined Partitions of a Set of Products. Vortrag am 28. April 2009. *http://www.ptidej.net/teamlife/meetings/091119%20-%20DGIGL%20Seminar/091119%20-%20Dave%20Parnas%20-%20Product%20Line%20Document%20-%20Thumbs.pdf* (26.11.2010)

[Paul95] M. Paulk, et al.: The Capabiliy Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley, Reading, Massachussets, USA, 1995.

[Pill98] F. T. Piller: Kundenindividuelle Massenproduktion – die Wettbewerbsstrategie der Zukunft. Hanser, München, Wien, 1998.

[Pine93] B. J. Pine II: Mass Customization: The new frontier in Business Competition. Harvard Business School Press, Boston, 1993.

[Pink02] A. Pink, et al.: Software-Entwicklung für Kommunikationsnetze. Springer, Berlin, Heidelberg, 2002.

[Pohl01] K. Pohl, A. Reuys: Considering Variabilities during Component Selection in Product Family Development. In: Proceedings 4th International Workshop on Product Family Engineering (PFE-4), Bilbao, 2001.

[Pohl04] K. Pohl et al.: Anforderungsbasierte Selektion von COTS-Komponenten. In: G. Böckle et al. (Hrsg.), Software-Produktlinien: Methoden, Einführung und Praxis, dpunkt Verlag, Heidelberg, 2004, S. 93-107.

[Pohl05] K. Pohl, G. Böckle, F. v.d. Linden: Software Product Line Engineering: Foundations, Principles, and Techniques. Springer, Heidelberg, 2005.

[Port83] M. E. Porter: Wettbewerbstrategie: Methoden zur Analyse von Branchen und Konkurrenten. Campus, Frankfurt am Main, 1983.

[Port00] M. E. Porter: Wettbewerbsvorteile – Spitzenleistungen erreichen und behaupten. 6. Auflage, Campus, Frankfurt am Main, 2000.

[Poul97a] J. S. Poulin: Measuring Software Reuse. Addison-Wesley, Boston, 1997.

[Poul97b] J. S. Poulin: Fueling Software Reuse with Metrics. In: Object Magazine, Vol. 7, No. 7 (Sept. 1997), S. 42-47.

[Prof08] H. Proff, H. V. Proff: Dynamisches Automobilmanagement: Strategien für Hersteller und Zulieferer im internationalen Wettbewerb. Gabler, Wiesbaden, 2008.

[Quir05] G. A. Quirk: Semiconductor Insights: Mobile Handset DesignLine. *http://eetimes.com/news/semi/showArticle.jhtml?articleID=171200863* [Download 20.10.2009]

[Rapp86] A. Rappaport: Creating Shareholder Value: The new standard for business performance. Free Press, New York, 1986.

[Reic05] G. Reichart, et al.: Flexray – Eine Vision wird Realität. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 421-437.

[Reif06] K. Reif: Automobilelektronik: eine Einführung für Ingenieure. Vieweg, Wiesbaden, 2006.

[Robe99] D. Robertson, K. Ulrich: Produktplattformen: Was sie leisten, was sie erfordern. In: Harvard Business Manager, 21. Jg., Heft 4/1999, S. 75-85.

[Salt08] A. Saltelli, et al.: Global sensitivity analysis: the primer, John Wiley & Sons Ltd., West Sussex, England, 2008.

[Scha05a] T. Scharnhorst et. al.: AUTOSAR – Challenges and Achievements 2005. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 395-408.

[Scha05b] J. Schaaf: Outsourcing nach Indien: Der Tiger auf dem Sprung. Indien Spezial, Deutsche Bank Research, Frankfurt am Main, 2005.

[Schä06] J. Schäuffele, T. Zurawka: Automotive Software Engineering – Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen. 3. Auflage, Vieweg, Wiesbaden, 2006.

[Schm03] K. Schmid: Planung von Softwarewiederverwendung – Ein systematischer Ansatz zum Scoping von Softwareproduktlinien. IESE-Bericht Nr. 102.03/D, Fraunhofer IESE, 2003.

[Schm04] K. Schmid: Scoping als Basis optimierter Wiederverwendung. In: G. Böckle et al. (Hrsg.), Software-Produktlinien: Methoden, Einführung und Praxis, dpunkt Verlag, Heidelberg, 2004, S. 43-53.

[Schm05] K. Schmid et al.: Introducing the PuLSE Approach to an Embedded System Population at Testo AG. In: Proceedings of the 27th International Conference on Software Engineering, St. Louis, 2005.

[Schn00] W. Schneider: Die Weiterentwicklung der Marke Volkswagen PKW in Form selbstregelnder Organisationseinheiten. In: R. Berndt, Innovatives Management, Springer, Berlin, 2000, S. 167-190.

[Scho05] P. Scholz: Softwareentwicklung eingebetteter Systeme. Springer, Berlin, Heidelbeg, 2005.

[Schr04] A. Schreiber: Transitionsprozess. In: G. Böckle et al. (Hrsg.), Software-Produktlinien: Methoden, Einführung und Praxis, dpunkt Verlag, Heidelberg, 2004, S. 139-152.

[Scht05] M. Schmitt: Software als original Ersatzteil – Ein Modell zur kontinuierlichen Anpassung der Fahrzeugfunktionalität an die Kundenbedürfnisse. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 817-828.

[SchV05] V. Schmid, et al.: Eine neue Dimension der Aktiven und Passiven Sicherheit mit PRE-SAFE und Bremsassistent BAS PLUS in der neuen Mercedes-Benz S-Klasse. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 215-235.

[SEI00] Software Engineering Institute: CMMI – SE/SW, v1.0 Capability Maturity Model – Integrated for Systems Engineering/Software Engineering, Staged Representation, Version 1.0 (CMU/SEI-2000-TR-018). Carnegie Mellon University, Pittsburgh, 2000.

[SEI09a] Software Engineering Institute, Carnegie Mellon University, 2009. *http://www.sei.cmu.edu/productlines/* [20.10.2009]

[SEI09b] Software Engineering Institute, Carnegie Mellon University, 2009. *http://www.sei.cmu.edu/productlines/framework.html* [20.10.2009]

[SEI10] Software Engineering Institute, Carnegie Mellon University, 2010. *http://www.sei.cmu.edu/productlines/consulting/techprobe/* [22.12.2010]

[SEI11] Software Engineering Institute, Carnegie Mellon University, 2011 *http://www.sei.cmu.edu/productlines/* [25.03.2011]

[Smit06] R. Smith: COCOMO and SCORM: Cost Estimation Model for Web Based Training. Interservice/Industry Training, Simulation, and Education Conference, Dezember 2006. *http://www.modelbenders.com/papers/2474.pdf* [05.11.2009]

[Soft11] Kompetenzzentrzum Software Engineering des Bundesministeriums für Bildung und Forschung, Fraunhofer IESE, www.software-kompetenz.de ; http://www.softwarekompetenz.com/?11134&highlight=analyse [24.03.2011]

[SPC93] Software Productivity Consortium (SPC): Reuse Adoption Guidebook. SPC-92051-CMC, Version 02.00.05, Herndon, Virginia, 1993.

[Stahl05] P. Stahlknecht, U. Hasenkamp: Einführung in die Wirtschaftsinformatik. 11. Auflage, Springer, Berlin, et al., 2005.

[Stah07] Thomas Stahl, et al.: Modellgetriebene Softwareentwicklung. Techniken, Engineering, Management. 2. Auflage, dpunkt Verlag, Heidelberg, 2007.

[Stan01] The Standish Group International: Extreme Chaos. 2001. *http://www.standishgroup.com/sample_research/index.php* [26.10.2009]

[Ster10] S. Stern, O. Guetta: Manage the Automotive Embedded Software Development Cost by Using a Functional Size Measurement Method (COSMIC). Renault, France, 2010. *http://www.erts2010.org/Site/0ANDGY78/Fichier/PAPIERS%20ERTS%202010/ERTS2010_0018_final.pdf* [18.11.2010]

[Stil00] A. Stiller: Prozessorgeflüster - von Achterbahnen und Karussellfahrten. In: c't Magazin für Computer Technik, Heft 22/2000, 2000, S.52.

[Stüt02] R. Stützle: Wiederverwendung ohne Mythos: Empirisch fundierte Leitlinien für die Entwicklung wiederverwendbarer Software. Dissertation an der TU München, 2002.

[Svah00] M. Svahnberg, J. Bosch: Issues Concerning Variability in Software Product Lines. In: Proceedings of the 3rd International Workshop on Software Architectures for Product Families (IWSAPF), Las Palmas de Gran Canaria, Springer, 2000.

[Svah01] M. Svahnberg, J. van Gurp, J. Bosch: On the Notion of Variability in Software Product Lines. In: Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA 2001), 28-31. August, Amsterdam, 2001.

[SWPL11] Software Product Lines, 2011 *http://www.softwareproductlines.com/benefits/quality.html* [24.03.2011]

[Thie04] S. Thiel, A. Hein: Architekturentwicklung. In: G. Böckle et al. (Hrsg.), Software-Produktlinien: Methoden, Einführung und Praxis, dpunkt Verlag, Heidelberg, 2004, S. 81-92.

[Tout09] H. Toutenburg, P. Knöfel: Six Sigma. Methoden und Statisktik in der Praxis. 2. Auflage, Springer, Berlin, Heidelberg, 2009.

[Trac95] W. Tracz: Confessions of a used program salesman. Addison-Wesley, Massachusetts, 1995.

[Tren03] A. Trendowicz, T. Punter: Quality Modelling for Software Product Lines. In: Proceedings of the 7th ECOOP Workshop for Quantitative Approaches in Object Oriented Software Engineering, Darmstadt, 2003.

[Trig03] J.-C. Trigaux, P. Heymans: Software Product Lines: State of the art. Projekt: Product Line Engineering of food Traceability Software, FUNDP, Namur, Belgien, 2003.

[Triq06] TriQuint Semiconductor: WCDMA Front-end Modules and Their Critical Role in 3G Wireless Handsets. In: Microwave Journal, Vol. 49, No. 3, März 2006, S. 134.

[Trom09] V. Trommsdorff: Konsumentenverhalten. 7. Auflage, Kohlhammer, Stuttgart, 2009.

[UKSM98] United Kingdom Software Metrcis Association (UKSMA): MK II Function Point Analysis – Counting Practices Manual. Version 1.3.1, United Kingdom Software Metrics Association, 1998.

[Ulfa05] N. Ulfat-Bunyadi, E. Kamsties, K. Pohl: Considering Variability in a System Family's Architecture during COTS Evaluation. In: Proceedings of the 4th International Conference on COTS-Based Software Systems (ICCBSS 2005), Bilbao, Springer, Berlin, 2005.

[User06] Usernomics.com: Ease-of-use Moto: Dealing with usability problems after product release. *http://www.usernomics.com/news/2006/10/ease-of-use-dogs-moto.html* [20.10.2009]

[VDA07] Verband der Automobilindustrie: Automotive SPICE Prozessassessment – Prozessbewertung nach Automotive SPICE in der Entwicklung von Software-bestimmten Systemen. Heinrich Druck + Medien, Frankfurt am Main, 2007.

[Voge03] F. Vogelezang, A. Lesterhuis: Applicability of COSMIC Full Function Points in an administrative environment – Experiences of an early adopter. 13th International Workshop on Software Measurement, IWSM, Montréal, Canada, Sept. 23-25, Shaker Verlag, 2003.

[vOmm04] R. C. van Ommering: Building Product Populations with Software Components. Dissertation an der University of Groningen, 2004.

[vSom05] T. van Someren: Strategische Innovationen – So machen Sie Ihr Unternehmen einzigartig. Gabler, Wiesbaden, 2005.

[Wagn05] A. Wagner, J. Bielefeld, A. Balwé: Das Kommunikationsbordnet des neuen BMW 3er. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik, 12. Internationaler Kongress Elektronik Im Kraftfahrzeug, VDI-Berichte 1907, VDI Verlag, Düsseldorf, 2005, S. 495-510.

[Wann05] H. Wannenwetsch: Vernetztes Supply Chain Management – SCM-Integration über die gesamte Wertschöpfungskette. Springer, Berlin, 2005.

[Wapp00] T. Wappler: Remember the basics. In: P. Donohoe, Software Product Lines: Experience and Research Directions, Kluwer Academic Publishers, Massachusetts, 2000, S. 73-84.

[Ward95] A. Ward, J. Liker, D. Sobek II: The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster. In: Sloan Management Review, Vol. 36, No. 5, 1995, S. 43.

[Weis99] D. M. Weiss, C. T. R. Lai: Software Product-Line Engineering: A Family-Based Software Development Process. Addison-Wesley, Boston, 1999.

[WiFo09] Wikipedia: Ford Model T, 2009. *http://en.wikipedia.org/wiki/Ford_Model_T* [20.10.2009]

[Wile99] E. Wiles: Economic models of software reuse: A survey, comparison and partial validation. Technical Report, Department of Computer Science, Wales, 1999.

[Wild04] H. Wildemann: Stabil und doch flexibel – Produktordnungssysteme. In: Harvard Business Manager, 26. Jg., Februar 2004, S. 37-43.

[Wilm01] T. Wilmshurst: An Introduction to the Design of Small-Scale Embedded Systems. Palgrave Macmillan, New York, 2001.

[With96] J.Withey: Investment Analysis of Software Assets for Product Lines. Technical Report, CMU/SEI-96-TR-010, November 1996, *http://www.sei.cmu.edu/pub/documents/96.reports/pdf/tr010.96.pdf* [20.10.2009]

[WiVi] Wikipedia: Videorekorder. *http://de.wikipedia.org/wiki/Videorekorder* [06.11.2009]

[Wöhe08] G. Wöhe, U. Döring: Einführung in die Allgemeine Betriebswirtschaftslehre. 23. Auflage, Vahlen, München, 2008.

[Wolf08] W. Wolf: Computers as Components. Principles of embedded computing system design. Second Edition, Morgan Kaufmann, Burlington, 2008.

[Yosh06] K. Yoshimura, D. Ganesan, D. Muthig: Defining a Strategy to Introduce a Software Product Line Using Existing Embedded Systems. In: Proceedings of the 6th ACM & IEEE International Conference on Embedded Software, 22-25. Oktober, Seoul, ACM, New York, 2006, S. 63-72.