

1.3 Zielsetzung des Arbeitskreises

Die Zielsetzung des Arbeitskreises war es, das Thema „Einsatz von Features bei der Entwicklung software-basierter Systeme“ aus einer praxisorientierten Sichtweise aufzuarbeiten. Im Einzelnen wurden die nachfolgenden Teilziele verfolgt.

- *Ermitteln von Einsatzszenarien für Features*, d.h. Identifizieren von Gebieten, in denen Features sinnvoll und nutzbringend eingesetzt werden können. Dies sollte ohne einen Anspruch auf Vollständigkeit erfolgen. Vielmehr sollte das Augenmerk auf bereits erfolgreichen Einsätzen liegen. Neben dem eigentlichen Einsatzszenario sollten auch die beteiligten Stakeholder und deren jeweiligen Aufgaben identifiziert werden.
- *Begriffsklärung*. Der Begriff Features wird (wie auch viele andere Begriffe im Bereich der Softwareentwicklung) in unterschiedlicher Weise verwendet. Versuche, eine allgemeingültige Definition zu finden, endeten bisher in wenig griffigen Beschreibungen. Daher sollte versucht werden, in Abhängigkeit eines gegebenen Einsatzszenarios eine griffigere Definition zu finden. Im Zuge dieser Begriffsklärung sollten auch die Begriffe Featureliste und Featuremodell sinnvoll einsortiert werden.
- *Ermitteln von Qualitätskriterien für Features*, d.h. Identifizieren von Anhaltspunkten für „gute“ oder „schlechte“ Features bzw. Featuremodelle. Natürlich hängen auch diese Qualitätskriterien stark vom jeweiligen Einsatzszenario ab.

Wichtig war dabei allen Beteiligten, dass die Bearbeitung obiger Zielsetzung nicht akademisch, sondern auf Basis realer Praxisbeispiele erfolgte.

1.4 Aufbau des Berichts

Der vorliegende Bericht ist wie folgt aufgebaut.

In *Kapitel 2* wird der Featurebegriff näher betrachtet. Basierend auf einer Literaturstudie werden bekannte Definitionen des Featurebegriffs einander gegenübergestellt. Darüber hinaus werden bekannte Einsatzszenarien dargestellt.

Kapitel 3 präsentiert die Einsatzgebiete aus der Praxis, die im Rahmen der verschiedenen Arbeitsgruppen-Treffen vorgestellt wurden, und die mit einer zentralen Grundlage für die hier vorgestellten Ergebnisse bilden.

Kapitel 4 stellt die Ergebnisse der Diskussionen des Arbeitskreises vor. Über die in *Kapitel 3* vorgestellten Einsatzgebiete aus der Praxis hinaus präsentiert *Kapitel 4* zudem weitere Szenarien zum Einsatz von Features. Weiterhin folgt der Versuch einer Begriffsdefinition des Featurebegriffs. Den Kern des Kapitels bildet ein Einsatz-Modell, in dem die verschiedenen Einsatzgebiete für Features zusammenhängend dargestellt werden. Eine erste Diskussion von Gütekriterien für Features beschließt das Kapitel.

Kapitel 5 fasst die Ergebnisse des Berichts kurz zusammen.

2 Der Featurebegriff

Wie bereits in Abschnitt 1.1 erläutert wurde, gewinnen Featurelisten und Featuremodelle als Mittel zur Definition von Produkten, aber auch zur Planung und Verfolgung von Entwicklungsprozessen zunehmend an Bedeutung.

Ein Vielzahl von Ansätzen aus Industrie und Wissenschaft haben sich mit der Feature-Modellierung auseinandergesetzt. Eine detaillierte Übersicht bieten die Arbeiten von Lichter et. al. [Lichter03] und Dörr [Doerr02]. Um einen detaillierten Vergleich der Ansätze zu ermöglichen, ist auch eine eingängigere Betrachtung des in den Ansätzen verwendeten *Feature*-Begriffs sinnvoll. Dabei fällt auf, dass nahezu alle Ansätze durch den FODA-Ansatz [Kang90] geprägt oder doch zumindest stark beeinflusst wurden. Die Definition des Begriffs

Feature betreffend äußert sich dies darin, dass, sofern eine explizite Definition des Feature-Begriffs gegeben wird, diese entweder die Begriffs-Definition des FODA-Ansatzes direkt referenziert [Kang02][Fey02][Hein00] oder sich bei einer eigenen Definition des Begriffs sehr stark auf diejenige des FODA-Ansatzes stützt [Griss98]. Auch wenn eine solch explizite Definition nicht gegeben wird [Savolainen01] und man die Bedeutung des Begriffs *Feature* nur implizit ableiten kann, lässt sich dies beobachten.

Daher ist es nicht verwunderlich, dass der Mehrheit der Ansätze ein - wenn auch ein weit gefasstes und sehr abstraktes - gemeinsames Verständnis des Begriffs *Feature* zugrunde liegt. FODA definiert ein Feature als

„*a prominent or distinctive user visible aspect, quality or characteristic of a software system or systems*“.

Trotz dieser recht vagen Definition, lassen sich einige Eigenschaften eines Features ableiten. Zunächst soll es sich laut obiger Definition um eine markante und wichtige Charakteristik einer Domäne oder eines Softwaresystems handeln, welche zudem explizit unterscheidbar zu anderen Charakteristiken sein soll und sich somit klar von diesen abgrenzt. Weiterhin soll es durch den Benutzer des Systems wahrgenommen werden können. Allgemeiner ausgedrückt, soll ein Feature für den Benutzer erlebbar sein und für ihn einen Mehrwert darstellen. Die Erlebbarkeit durch den Endkunden steht hier im Widerspruch zu anderen Definitionen, die der Literatur entnommen werden können. Dort wird explizit die Erlebbarkeit nicht auf den Benutzer bzw. Endkunden eingeschränkt, sondern erweitert auf beliebige Stakeholder (vgl. [Doerr02], Seite 5ff).

Diese Definition wird in FODA weiterhin durch eine Klassifizierung von Features nach der Zugehörigkeit zu den folgenden Kategorien näher spezifiziert:

- *Service/Dienste*: Diese Kategorie enthält Features, die Dienstleistungen des Systems repräsentieren
- *Betriebsumgebung*: Features, welche besondere Anforderungen an Software oder Hardwareumgebung beschreiben
- *Domänentechnik*: Features für domänenspezifische Merkmale, wie festgelegte Standards, Verfahren und Empfehlungen
- *Implementierungstechnik*: Diese Features stehen für spezielle Implementierungstechniken, durch die sich Produkte der Domäne unterscheiden. Im Vergleich zur Domänentechnik sind diese Features allerdings generisch und können auch in anderen Domänen eingesetzt werden.

Die Ausprägung des Feature-Begriffs in den einzelnen Ansätzen unterscheidet sich insbesondere dadurch, dass entweder eine bestimmte Klasse von Features besonders betont oder der Feature-Begriff sogar vollständig auf eine bestimmte Klasse von Features eingeschränkt wird. Dies ist teilweise bereits beim FODA-Ansatz selbst zu beobachten, wo in der validierenden Fallstudie nur die Service/Dienst-Features untersucht und in

- *Funktionale Features*: stehen für die unterschiedlichen fachlichen Funktionalitäten, welche das System dem Benutzer zur Verfügung stellt
- *Operationale Features*: beschreiben Benutzer-Interaktionen
- *Präsentations-Features*: beziehen sich auf die Art der Darstellung von Informationen für den Benutzer

weiter unterschieden werden.

Wenn auch bereits im ursprünglichen FODA-Ansatz explizit darauf hingewiesen wird, dass die dort getroffene Einschränkung nur aus rein zeitlichen Gründen stattfand, so hat dies die nachfolgenden Ansätze in sofern entscheidend mitgeprägt, dass sich eine starke Betonung der funktionalen Features bei nahezu allen anderen Ansätzen abzeichnet. Besonders stark kommt

dies in [Kang98] zum Ausdruck, wo ein Feature aus Sicht des Endanwenders als eine *unterscheidbare funktionale Abstraktion eines Systems* betrachtet wird (abweichend davon wird der Feature-Begriff im Ansatz aber durchaus anders eingesetzt, so dass beispielsweise auch Implementierungstechniken berücksichtigt werden). Auch in [Griss98] wird diese Sicht vertreten, wobei als Abgrenzungskriterium eines Features gegenüber einem Use Case der systemübergreifende Charakter betont wird, d.h. ein Feature beschreibt eine Funktionalität einer bestimmten Domäne, nicht eines einzelnen Systems.

Neben dieser sehr starken Betonung der funktionalen Features lässt sich, getrieben durch den Einsatz der Feature-Modellierung in der Produktlinienentwicklung, bei nahezu allen Ansätzen eine starke Fokussierung auf den Unterscheidungscharakter eines Features beobachten, d.h. ein Feature wird als ein unterscheidendes Merkmal von Produkten betrachtet. Dies hängt damit zusammen, dass die Feature-Modellierung zur Modellierung der Variabilitäten in einer Produktlinie eingesetzt wird und kommt vor allem in [Hein00] besonders zum Ausdruck, wo die Unterscheidung zwischen einem Feature und einer *Anforderung* darin gesehen wird, dass ein Feature zur Modellierung der Variabilitäten und eine Anforderung zur Beschreibung der Gemeinsamkeiten der Produkte einer Produktlinie eingesetzt wird. Eine Sonderstellung bezüglich der Begriffsdefinition nimmt der von Czarnecki [Czarnecki98] beschriebene Ansatz ein, der die FODA-Begriffsdefinition nicht weiter einschränkt, sondern noch erweitert. Die von Czarnecki gelieferte Begriffsdefinition sieht ein Feature als eine *unterscheidende Eigenschaft eines Konzepts* und legt den Schwerpunkt nicht auf die Sichtbarkeit eines Features für den Endnutzer sondern, wie bereits oben erwähnt, allgemeiner auf die Relevanz eines Features für eine Gruppe von Interessenten (*Stakeholdern*).

Diese etwas weiter gefasste Begriffsdefinition ist stark durch den ODM-Ansatz (Organizational Domain Modeling) geprägt, wo die Nachverfolgbarkeit (engl. *traceability*) jedes Konzepts oder Features bezüglich der zugehörigen Interessensgruppe explizit gefordert wird.

Interessanterweise lassen sich von den doch recht allgemein gehaltenen Definitionen unterschiedliche Verwendungszwecke für Features feststellen. Vergleicht man die verschiedenen Ansätze, so lassen sich eine Reihe von Gemeinsamkeiten, aber auch Unterschiede bezüglich der Motivation für die Feature-Modellierung feststellen. Die mit der Feature-Modellierung verfolgten Ziele, spiegeln sich ebenfalls in den verwendeten Modellierungselementen wieder.

In den in [Lichter03] untersuchten Ansätzen lassen sich die folgenden Verwendungszwecke identifizieren:

- Feature-Modellierung als Technik der Domänenanalyse und –dokumentation
 - Kommunikation zwischen Entwicklern und Domänenexperten verbessern
- Dokumentation einer Produktlinie
 - Ableitung von (potentiellen) Produktkonfigurationen

Trotz dieser unterschiedlichen Einsatzgebiete lassen sich doch einige Gemeinsamkeiten bzgl. der verwendeten Modellierungstechniken und –elemente erkennen:

- Modellierung von Funktionalitäten und qualitativer Charakteristiken des Anwendungsbereichs
- Modellierung struktureller Beziehungen zwischen Features
 - Komposition
 - Spezialisierung
 - Implementierung
- Modellierung von Abhängigkeiten zwischen Features
 - Implizierung
 - Ausschluss

- Modellierung von Gemeinsamkeiten und Variabilitäten
 - Obligatorische Features
 - Optionale Features (0..1 aus N Auswahl)
 - Alternative-Features (1 aus N Auswahl)
 - Oder-Features (1..N aus N Auswahl)

Die unterschiedlichen Ansätze verwenden zum Teil unterschiedliche Notationselemente, um die oben beschriebenen Variabilitätseigenschaften, Beziehungen und Abhängigkeiten zu visualisieren. Sie besitzen hingegen alle die mit FODA eingeführte Semantik und erweitern diese gegebenenfalls. Tabelle 2 gibt eine Übersicht über die verwendeten Notationselemente und deren Semantik.



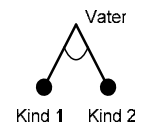
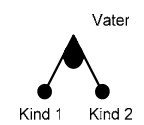
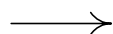
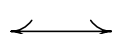
Beziehung	Typ	Semantik	Charakteristik	Notation
Domänenbeziehung	Obligatorisch	Wenn das Vater-Feature ausgewählt wird, so muss das Kind-Feature ebenfalls ausgewählt werden.		
	Option	Wenn das Vater-Feature ausgewählt wird, so kann das Kind-Feature wahlweise ausgewählt werden.		
	Alternative	Wenn das Vater-Feature ausgewählt wird, muss genau ein Kind-Feature aus der Menge alternativer Kind-Features ausgewählt werden.	Impliziter wechselseitiger Ausschluss zwischen alternativer Kind-Features.	
	Oder	Wenn das Vater-Feature ausgewählt wird, so muss mindestens ein Kind-Feature ebenfalls ausgewählt werden.		
Abhängigkeit	Implikation	Die Auswahl eines Features impliziert die Auswahl eines anderen Features.	Transitiv	
	Ausschluss	Zwei Features schließen sich wechselseitig aus und können nicht zusammen Bestandteil einer Produktdefinition sein.	Symmetrisch	

Tabelle 2: Notationselemente eines Feature-Diagramms

Abbildung 1 zeigt ein einfaches Beispiel eines Feature-Modells aus dem Automobilkontext:

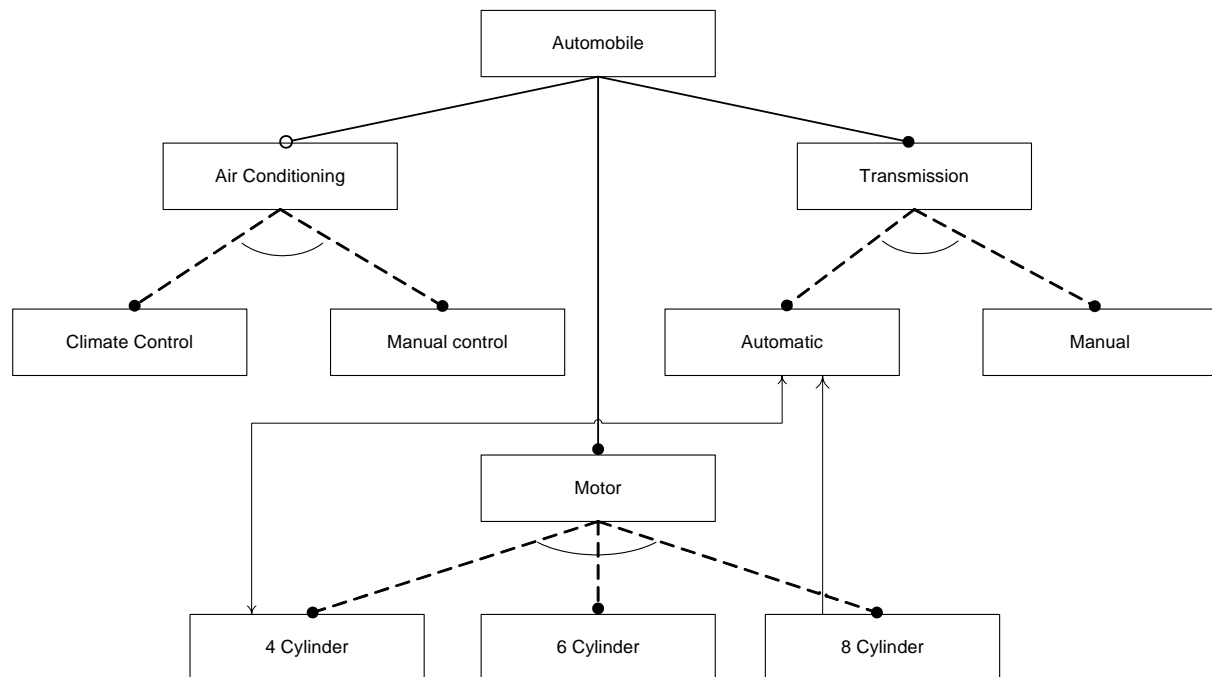


Abbildung 1: Feature-Diagramm.

Neben den in [Lichter03] untersuchten Ansätzen, die die Feature-Modellierung für die Domänenmodellierung und zur Produktkonfiguration einsetzen, wurden durch die Mitglieder des Arbeitskreises weitere Einsatzgebiete aufgezeigt. Diese werden im folgenden Abschnitt vorgestellt.

3 Einsatzberichte aus der Praxis

In den folgenden Abschnitten werden Szenarien skizziert, welche in den jeweiligen Arbeitsgebieten der Mitglieder des Arbeitskreises auftreten. Die im Folgenden beschriebenen Szenarien stellen somit realer Szenarien dar, in denen Features im Kontext der industriellen Softwareentwicklung zum Einsatz kommen.

3.1 Einsatz von Features zum Projektmanagement (DaimlerChrysler)

Features werden im Zuge der Fahrzeugentwicklung an unterschiedlichen Stellen im Entwicklungsprozess und zu unterschiedlichen Zielen eingesetzt. Im Folgenden findet sich eine Beschreibung des Einsatzes von Features, genauer gesagt von Featurelisten, zur Planung und Steuerung einer Steuergeräte-Entwicklung bei der DaimlerChrysler AG.

In modernen PKWs findet sich eine Vielzahl von Steuergeräten. Diese software-basierten Systeme kommunizieren über verschiedene Busse miteinander. Darüber hinaus sind diverseste Sensoren und Aktuatoren direkt mit einem einzelnen Steuergerät verbunden. Das gesamte Steuergerät wird typischerweise nicht vom Automobilhersteller entwickelt, sondern auf Basis umfangreicher Lastenhefte von einem Zulieferer. Durch die hohe Vernetzung der verschiedensten Systeme im Fahrzeug übernimmt der Steuergeräte-Verantwortliche auf Seiten des Automobilherstellers sehr stark die Rolle eines Koordinators. Er sammelt sämtliche Anforderungen an das betrachtete Steuergerät, führt Abstimmungen und Klärungen herbei und leitet die abgestimmten und verfeinerten Anforderungen an den Lieferanten weiter.

Ein Lastenheft für ein Steuergerät umfasst in der Regel mehrere hundert Seiten. Darüber hinaus gibt es noch eine Vielzahl an weiteren Vorgabedokumenten, in der allgemeine Dinge, wie beispielsweise EMV-Prüfbedingungen, Produkt-Kennzeichnungen oder die verwendende Betriebssystem-Software geregelt werden.

Abbildung 2 stellt schematisch den zugrunde liegenden Prozess der Steuergeräte-Entwicklung aus Requirements Engineering Sicht dar.

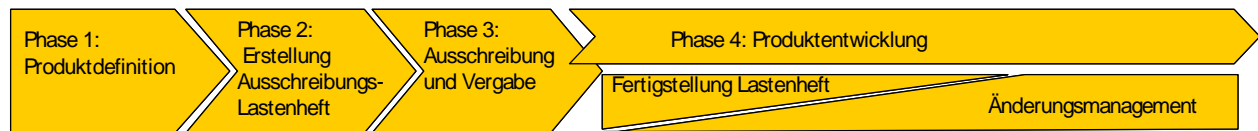


Abbildung 2: Eingesetzter Requirements Engineering Prozess.

Über den Entwicklungsverlauf, der von der Vergabe an der Erstellung der Ausschreibungsunterlagen bis zum Serienanlauf einer neuen Fahrzeugbaureihe in der Regel drei bis fünf Jahre umfasst, ergeben sich eine Vielzahl an Änderungen. Treiber für Änderungen sind mannigfaltig. Beispiele sind technologische Neuerungen (z.B. neuer Sensor verfügbar), Änderungen gesetzlicher Vorgaben (z.B. Abgas-Gesetzgebung), oder sind durch den Wettbewerber verursacht (Wettbewerber X hat das Feature Y angekündigt).

Für die Planung und Steuerung einer Steuergeräte-Entwicklung wird daher ein Mittel benötigt, dass es erlaubt, den aktuellen Umfang des Steuergeräts hinreichend genau zu beschreiben und gleichzeitig übersichtlich zu sein. Hierzu haben sich so genannte Featurelisten als Mittel der Wahl herausgestellt.

Abbildung 3 zeigt exemplarisch ein Beispiel für eine Featureliste. Die Liste ist in verschiedene inhaltliche Bereiche gegliedert und identifiziert alle wesentlichen Eigenschaften oder Merkmale, die das Produkt in seiner Gesamtheit beschreiben. Wesentliches Augenmerk liegt dabei auf so genannten Kostentreibern, d.h. Merkmalen, die Auswirkung auf die Kosten des Steuergeräts haben.

Die Liste wird in der Phase 1 (Produktdefinition) erstellt und dient hier als primäres Abstimmungsmedium. Dieser Phase der Produktentwicklung ist durch eine Vielzahl an Abstimmungen und daraus resultierenden Änderungen charakterisiert. Ein ständiges Nachziehen des gesamten Lastenhefts wäre hierbei so gut wie unmöglich.

In Phase 2 (Erstellung Ausschreibungslastenheft) dient die Featureliste nun als Übersicht über das zu erstellende Vergabe-Lastenheft. Für jeden Eintrag der Featureliste werden alle zur Vergabe benötigten Informationen bereitgestellt.

In Phase 3 sind nun die verschiedenen potentiellen Zulieferer aufgefordert, ihre Angebote abzugeben, wobei sie zu den einzelnen Positionen der Featureliste Stellung beziehen

müssen. Nachdem ein Zulieferer ausgewählt wurde, beginnt die eigentliche Produktentwicklung (Phase 4). Zu Beginn dieser Phase wird auf Basis der Featureliste eine Releaseplanung

Feature-list model XYZ	
1.	Functional Features <ul style="list-style-type: none"> • Tachometer • Speedometer • Fuel tank • Odometer • Outside temperature
2.	Hardware Features <ul style="list-style-type: none"> • Lighting • Display • Warning lamps • Stepper motors • Dial faces
3.	HMI Features <ul style="list-style-type: none"> • Display outside temperature • Menu <ul style="list-style-type: none"> • Trip computer <ul style="list-style-type: none"> • Odometer • Trip meter <ul style="list-style-type: none"> • Reset trip meter • Telephone <ul style="list-style-type: none"> • Telephone book <ul style="list-style-type: none"> • List of phonebook entries • Last dialed numbers <ul style="list-style-type: none"> • List of last 3 dialed numbers

Abbildung 3: Beispiel für eine Featureliste.

durchgeführt, d.h. es wird für jedes Feature festgelegt, zu welchem Musterstand dieses umgesetzt werden muss. Diese Planung dient auch als Grundlage für die Fertigstellung der Details im Lastenheft.

Während der Produktentwicklung wird in der Featureliste ständig der Fertigstellungsgrad der Lastenheft-Details, die aktuelle Releaseplanung sowie die Umsetzung und der erfolgte Test dokumentiert.

Systemseitig sind diese Featurelisten häufig in DOORS hinterlegt. In dedizierten Spalten wird für jedes Feature neben dessen Namen eine Kurzbeschreibung hinterlegt. Zudem finden sich Angaben über etwaige Ansprechpartner, den Fertigstellungsgrad der Spezifikation, Daten zur Releaseplanung und zur Produktumsetzung.

Featurelisten werden mittlerweile in verschiedenen Serien-Entwicklungsprojekten erfolgreich eingesetzt. Ein flächendeckender Einsatz ist bisher jedoch noch nicht zu beobachten. Wenngleich die gewählte Form der Liste (oder genauer gesagt, die Form eines flachen Baumes; die Einträge der Liste sind typischerweise in Featuregruppen, Features und Sub-Feature organisiert) es nicht unterstützt, Abhängigkeiten zwischen Features explizit darzustellen, ist gerade diese Form der Darstellung sehr erfolgreich. Ein Grund hierfür könnte sein, dass die existierenden Listen sehr leicht und ohne weitere Einführung verstanden werden können. Das Wissen über und die Beachtung von etwaigen Abhängigkeiten bleibt Aufgabe der jeweiligen Produktverantwortlichen.

In der Anfangsphase der Nutzung von Featurelisten konnte man beobachten, dass die Listen häufig umstrukturiert wurden. Mittlerweile haben sich gewisse Strukturen etabliert, die bisher jedoch vom jeweiligen Projekt abhängen. Der Aufwand zur Erstellung und Pflege der Featurelisten ist vergleichsweise gering. Typischerweise wird die Liste im Zuge laufender Regelkommunikationen überarbeitet.

Das eingesetzte Werkzeug DOORS ist zur Verwaltung von Featurelisten sicherlich nicht optimal, bietet sich aber an, wenn auch die übrigen Spezifikationsinhalte in DOORS vorliegen.

3.2 Einsatz von Features zum Projektmanagement (Bosch)

Die Robert Bosch GmbH liefert unter anderem elektronische Systeme und Komponenten für Kraftfahrzeuge. Dieser Markt ist charakterisiert durch hohe Stückzahlen und wird im Wesentlichen durch den Preis für die Hardware getrieben. Andererseits nehmen die Komplexität der Systeme und deren Vernetzung im Fahrzeug und damit der Anteil der Funktionalität, die in Software abgebildet wird, ständig zu. In eingebetteten Systemen werden Funktionen und Merkmale in Hardware und Software realisiert.

Steigende Komplexität der Systeme gefährdet die Einhaltung der vereinbarten Liefertermine und -umfänge. Durch die zunehmende Vernetzung der Systeme wird deren Einhaltung immer wichtiger, damit das Zusammenspiel der Systeme frühzeitig geplant und getestet werden kann. Dafür muss die Qualität der einzelnen Systeme stimmen, d.h. die Systeme müssen Merkmale und Funktionen im vereinbarten Umfang beinhalten.

Das Produkt Anzeigesystem-Steuergerät zeichnet sich im Gegensatz zu anderen eingebetteten Systemen im Kraftfahrzeug dadurch aus, dass viele Funktionen und Merkmale visuell überprüft werden können. Im Anzeigesystem werden dem Fahrer die Zustände des Fahrzeugs angezeigt (zum Beispiel Geschwindigkeit, Motortemperatur, Defekte usw.). Die anzuzeigenden Informationen werden meist von anderen Steuergeräten im Kraftfahrzeug geliefert, das Anzeigesystem ist mit nahezu allen anderen Steuergeräten im Kraftfahrzeug vernetzt. Für den Entwicklungsprozess folgt daraus, dass Änderungen an anderen Systemen häufig zu Änderungen der Anforderungen an das Anzeigesystem führen. Grundsätzlich betrachtet zeigt ein Anzeigesystem unabhängig vom Fahrzeugtyp immer die gleichen Zustände an. Andererseits wird die Anzeige dieser Zustände für jedes Modell in Abhängigkeit von der Fahrzeug-

Konfiguration individuell gelöst. Jedes Fahrzeug hat als Gesamtsystem unterschiedliche Funktionalitäten realisiert, von denen neben den Standard-Parametern Zustände dargestellt werden.

Features werden in der Anzeigesystem-Entwicklung eingesetzt, um den Ablauf der Entwicklung zu steuern. Mit den Features wird benannt, welches die wesentlichen Merkmale des Anzeigesystem-Steuergerätes sind und zu welchem (Zwischen-)Liefertermin oder Programmstand welche Merkmale realisiert sein sollen. Damit kann geplant werden, zu welchen Zeitpunkten im voraus Anforderungen und Informationen aus anderen Systemen bzw. Steuergeräten vorliegen müssen. Features sind ein zwischen Kunden und Zulieferer üblicher Terminus, um den Leistungsumfang des Anzeigesystems zu beschreiben. Neu ist in diesem Umfeld die konsequente Ausrichtung der gesamten Projektsteuerung auf Liefertermine mit festgelegten Featureumfängen.

In der Angebotsphase werden die Features festgelegt, die in dem Anzeigesystem realisiert werden sollen. Sie bilden die Grundlage für die Entwicklungsaufwände und sind damit eine wesentliches Ordnungsschema für die Preisbildung. Die Anforderungen an das Produkt werden vom Kunden als Lastenheft als Word-Dokument oder in Form eines Datenbankstandes aus einem Requirements Management Werkzeug bereitgestellt. Dieses Lastenheft ist zum Zeitpunkt der Angebotserstellung noch nicht präzise bis in jedes Detail ausformuliert. Die genauere Spezifikation der Anforderungen erfolgt im iterativen Prozess der Entwicklung eines Kraftfahrzeugs nach und nach. Lastenhefte für Anzeigesysteme enthalten ca. 25000 Einträge. In einem Analyse-Schritt werden davon ca. 8000 Anforderungen an das Produkt, welche relevant für die Software sind, extrahiert. Bei jedem neuen Stand des Lastenhefts muss der Analyseschritt wiederholt werden. Der Aufwand für diesen Analyseschritt hängt demnach davon ab, wie häufig neue Stände des Lastenhefts geliefert werden und wie intelligent der Änderungsprozess an Lastenheften organisiert ist.

Pro Jahr erfolgen mehrere Ablieferungen, damit der Kunde neue Features des Kombi-Instruments validieren und im Gesamtsystem erproben kann. Der Umfang jedes Liefertermins wird anhand der Features durch Absprache zwischen Kunde und Zulieferer festgelegt. Daraufhin untersuchen die Zulieferer die aktuell gültige Version des Lastenhefts nach Anforderungen, die für die Umsetzung des von ihnen zu verantworteten Features relevant sind und legen im RM-Werkzeug eine Verbindung zwischen Feature und Lastenheft-Anforderung ab (vgl. Abbildung 4). Dann wird überprüft, welche Komponenten für die Umsetzung des Features benötigt werden. Die Spezialisten für die jeweilige Komponente übersetzen die Anforderungen im Lastenheft in Anforderungen an ihre Komponente. Damit wird sichergestellt,

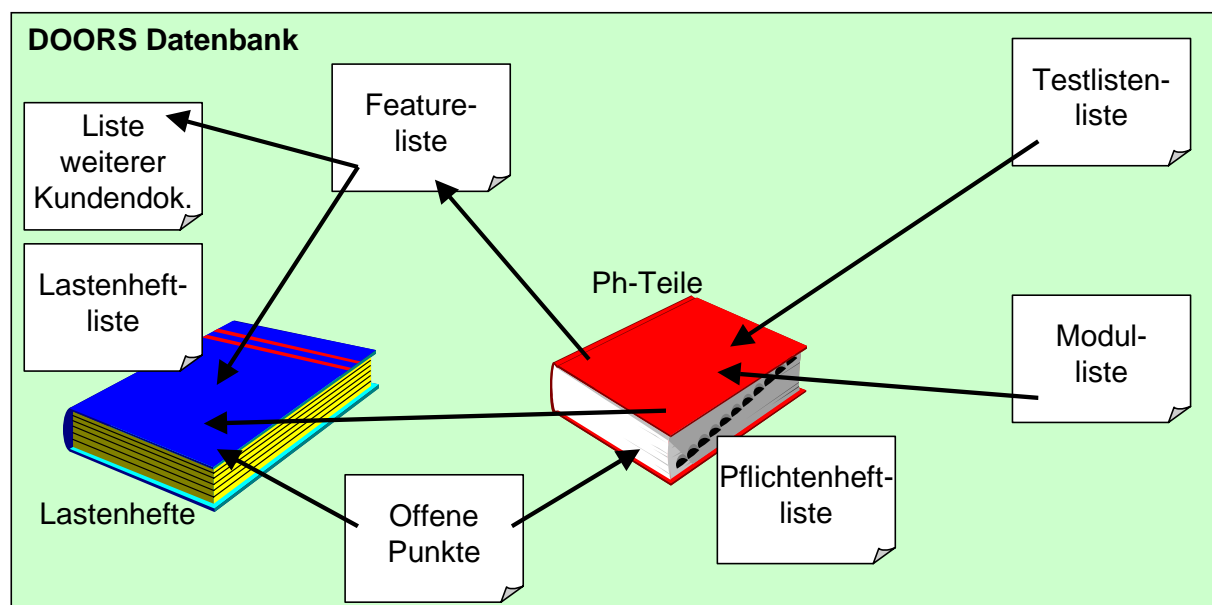


Abbildung 4: Zusammenhang zwischen Lasten- und Pflichtenheft

dass entsprechend der Liefertermine die Features aus Lastenheft-Anforderungen in Pflichtenheft-Anforderungen umgearbeitet werden. Das Pflichtenheft wird mit dem Kunden abgestimmt, Änderungen werden in einer offenen Punkteliste erfasst und mit den jeweiligen Einträgen im Lasten- und Pflichtenheft verbunden, zu dem die Unklarheit besteht. Weiterhin dient das Pflichtenheft als Grundlage für die Erstellung der Testlisten für den Systemtest.

Insgesamt werden mit diesem Vorgehen mehrere Entwicklungsprojekte gesteuert. Die vorhandenen Featurelisten bilden die Basis für Weiterentwicklungen, die während des Lebenszyklus eines Fahrzeugmodells bei Modellpflegen erfolgen. Das eingesetzte Werkzeug DOORS ist für das Vorgehen geeignet, muss aber durch einige aufwändige Skripte angepasst werden. Besonders fehlende Beibehaltung der IDs beim einspielen neuer Versionen der Lastenheft-Module musste angepasst werden. Die intensive Nutzung von Verbindungen der Elemente in den verschiedenen Modulen erfordert Prozessdisziplin in einem Umfeld, in dem häufige Änderungen üblich sind. Das Werkzeug DOORS ist hierfür eine gangbare Lösung. Wiederverwendung spielt in dieser Anwendung keine Rolle.

3.3 Einsatz von Features zur strategischen Produktplanung (Siemens)

Um Produkte einer neuen Mobiltelefongeneration definieren zu können, müssen sowohl strategische markt- als auch technologiebezogene Anforderungen berücksichtigt werden.

Diese Anforderungen werden bei Siemens Mobile gesammelt, gefiltert, gebündelt und schließlich zu Featurelisten kompiliert.

Um der Fülle an Informationen aus verschiedensten Quellen beider Perspektiven gerecht zu werden, wurden einerseits ein kontinuierlicher Marketingprozess, das sog. M-Programm und andererseits ein kontinuierlicher Technologieprozess, das sog. T-Programm aufgesetzt (siehe auch Abbildung 5).

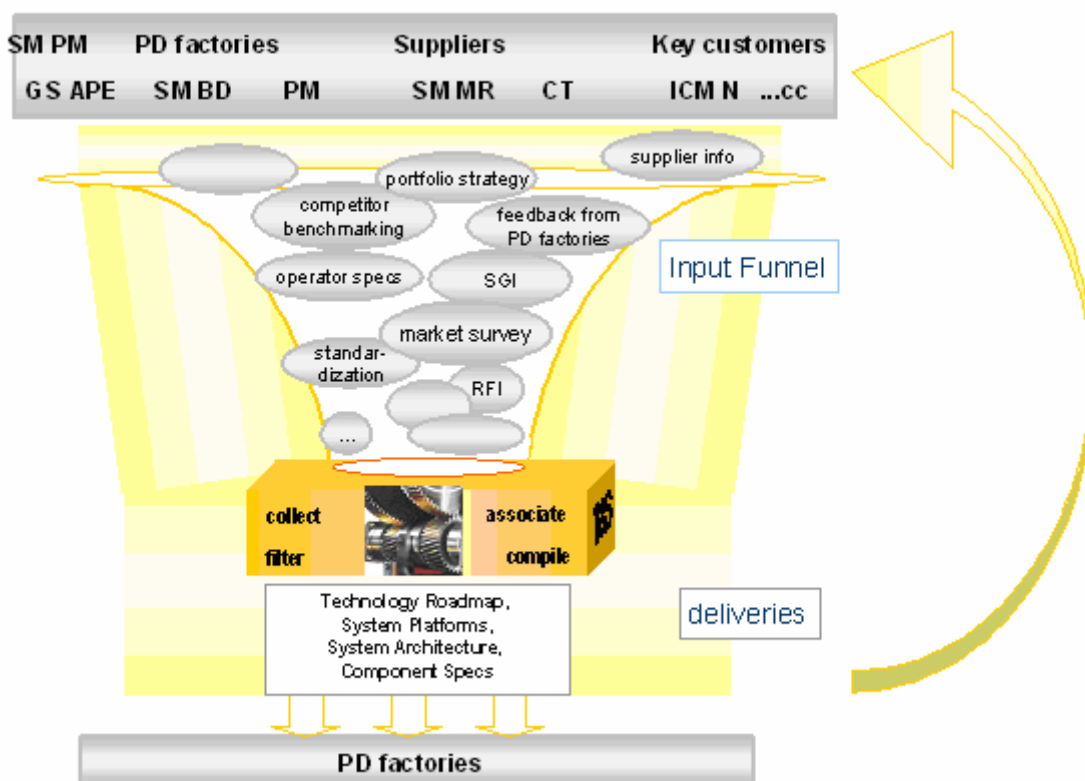


Abbildung 5: T-Programm mit den Aktivitäten collect, associate und compile

Für die Definition einer neuen Mobiltelefongeneration (zum sog. M-alpha Meilenstein) werden die Ergebnisse der kontinuierlich durchgeführten Technologie-Untersuchungen in einer Technology Roadmap zusammengefasst. Diese bildet die Grundlage für die Festlegung der Feature-Liste, mit der die Anforderungen aus den unterschiedlichsten Requirements-Spezifikationen, offenen Anforderungen von Netzwerk Operatoren und aus zurückgestellten Change Requests realisiert werden sollen.

In den UI Principles, auch UI Styleguide genannt, werden generelle Muster für das Interaktionsverhalten und das Screen-Design für eine Produktgeneration festgelegt. Sie müssen ca. 3 Wochen vor dem Beginn der Integration des ersten Produktes einer neuen Mobiltelefongeneration (zum sog. M₀ Meilenstein) gereviewed zur Verfügung stehen. Sie bilden später einen Teil der UI-Specification und sind nicht Bestandteil der SW-Featureliste (siehe auch Abbildung 6).

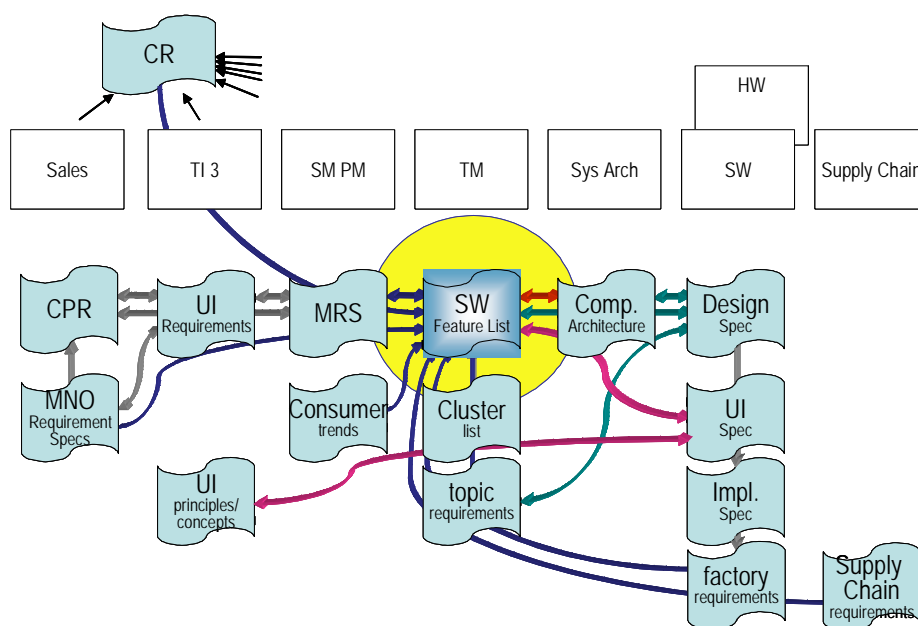


Abbildung 6: Dokumente im Requirements Engineering Prozess.

In weiteren Prozessschritten

- wird ein Produktportfolio festgelegt und bestimmt, welche Features, in welchen Produkten, zur Verfügung gestellt werden sollen.
- werden Systemplattformen so definiert, dass mit so wenig, wie möglich, so viele Produkte, wie möglich, realisiert werden können.
- wird eine Systemarchitektur entworfen, die es ermöglicht, Features so in Komponenten abzubilden, dass eine hohe Wiederverwendbarkeit garantiert werden kann.
- wird der performante und simultane Ablauf parallel laufender Features abgesichert.

Dieser Prozess ermöglicht es, innerhalb einer Produktgeneration ein breites Produktportfolio zu generieren.

Operator Shaping und Compliance Mapping

Die Netzwerk-Operatoren legen vierteljährlich ihre Requirements Spezifikationen vor, in denen sie ihre Anforderungen an die Endgeräte für einen bestimmten Zeitraum im folgenden Kalenderjahr festlegen. Damit diese möglichst mit der zwischen Marketing und Technologie

abgestimmten Featureliste realisiert werden können, wird versucht die Netzwerk-Operatoren sehr früh von den eigenen Technologie- und Featureüberlegungen zu überzeugen.

In einem Compliance Assessment wird festgestellt, inwieweit Features - aus der abgestimmten Featureliste die Anforderungen aus den Requirements Spezifikationen der Netzwerk-Operatoren erfüllen. Je nach Dringlichkeit führen offenen Anforderungen entweder zu Change Requests für die laufende Plattformentwicklung oder werden bei der Festlegung der nächsten berücksichtigt.

Verwaltung der Requirements Listen

Marketing- und Technologie-Strategen, System Architekten, Technology Management, Compliance Assessoren, Komponenten-Architekten und –Entwickler liefern zu unterschiedlichen Zeitpunkten Informationen zu Features ab, bzw. holen solche ein. Statistisch betrachtet, müssen ungefähr 100 Experten etwa 4000 Features bearbeiten Der Einsatz von intelligenten datenbankbasierten Verwaltungswerkzeugen ist hierzu folglich zwingend notwendig. Um eine hohe Flexibilität zu gewährleisten muss das Datenbankschema so schlank wie möglich definiert werden - d.h. nur so viele Querbeziehungen, wie unbedingt notwendig.

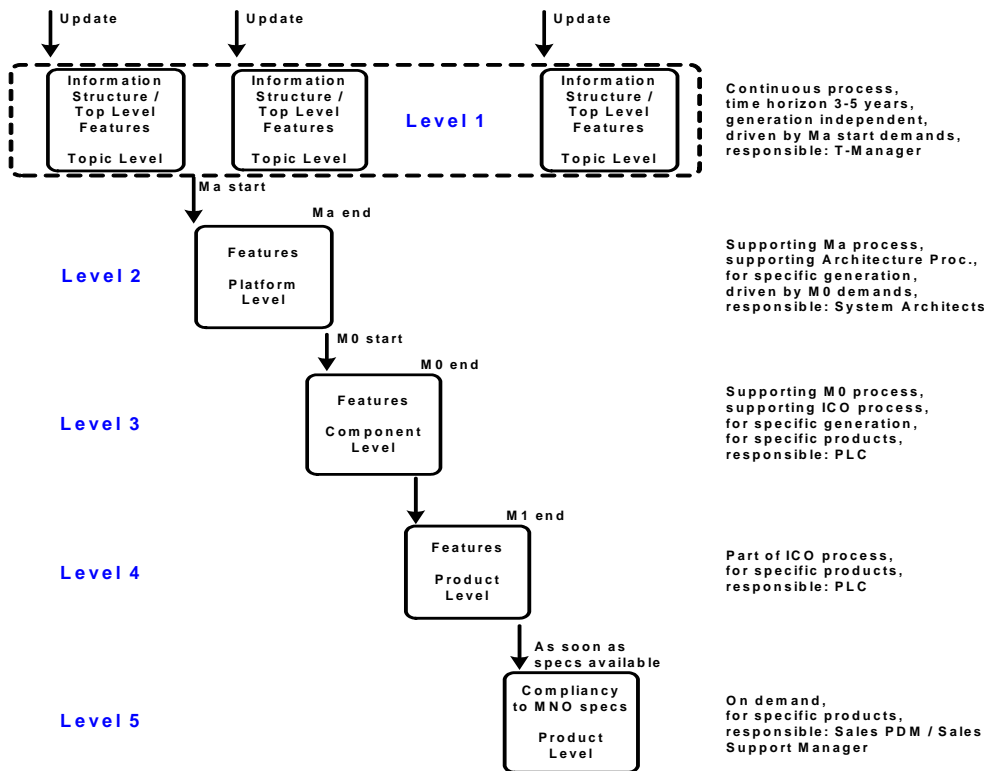


Abbildung 7: Requirements Engineering Ebenen.

Glossar:

GS APE	Global Sourcing ...
HW, SW	Hardware, Doftware factories
ICM N	Information Communication Mobile Networks
ICO Process	Intetgrated Components Process
MNO	Mobile Network Opoerators (Vodafone, T Mobile, ..)
PD	Product Development
PM	Portfolio Management
RFI	Request for Implementation

Sales	Sales area
S GI	Strategy Guidelines Information
SM BD	Strategy & Marketing Business Development
SM MR	Strategy & Marketing Market Research
SM PM	Strategy & Marketing Product Marketing
SYS Arch	Systems Architects
TI	Technology & Innovation Area
TM	Technology Management

Dokumente im Requirements Engineering (vgl. Abbildung 6):

CR	Change Request
CPR	Customer Prioritized Requirement Spec
UI	User Interface
MNO Requirements	Mobile Network Operators Requirements Spec
MRS	Marketing Requirement Spec
SW Feature	SW Feature List
Consumer	Consumer Trends (requirements from endusers)
Cluster List	clustered Technologies
Topic requirements	Requirements from Technology Topics
Comp Architecture	Component Architecture
Design Spec	Design Specification (Component Specs)
UI Spec	User Interface Specification
Impl. Spec	Implementation Specification
Factory requirements	requirements from HW-/SW- factories
Supply Chain requ.	requirements from Supply Chain Mgmt.

3.4 Einsatz von Features zur strategischen Produktplanung (DaimlerChrysler)

In Abschnitt 3.1 wurde der Einsatz von Featurelisten zur Planung und Steuerung der Entwicklung eines einzelnen Steuergeräts beschrieben. Über diesen Einsatzkontext hinaus gibt es im Zuge der Fahrzeugentwicklung ein weiteres zentrales Einsatzgebiet für Featurelisten, nämlich die strategische Produktplanung.

Diese Phase ist vorgelagert und teilweise überlappend zur Definition einzelner Steuergeräte zu sehen (vgl. Abbildung 2). Das Ziel der strategischen Produktplanung ist es, für eine neue Fahrzeugbaureihe festzulegen, welchen Features auf Gesamtfahrzeugebene oder Teilfahrzeugebene (z.B. Innenraum oder Gesamt-E/E-Umfänge) in dieser Baureihe enthalten sein sollen. Basierend auf dieser Planung wird dann die Zuordnung der Gesamtfunktionalitäten auf die einzelnen Produktbestandteile vorbereitet.

Produktdefinition

Bei der Produktdefinition sind drei zentrale Punkte zu berücksichtigen:

- *Konsistente Definition im Kontext der Produktpalette.* Die Entscheidung über die Wahl oder Nicht-Wahl von Features muss natürlich unter Beachtung der Gesamt-Produktstrategie erfolgen. Aus diesem Grunde erfolgt die Definition der Umfänge nicht in einer baureihen-spezifischen Liste, sondern in einer Gesamt-Liste, die verschiedene Pro-

dukte umfasst. So ist ein schneller Quervergleich zwischen bereits existierenden Baureihen und der zu planenden Baureihe leicht möglich.

- *Berücksichtigung von Wettbewerberprodukten.* Ein wesentlicher Treiber bei der Entscheidung über Features ist natürlich auch das Agieren der Wettbewerber. Aus diesem Grund werden häufig Informationen über Wettbewerberfahrzeuge und deren Featureausstattung mit in die Featureliste aufgenommen.
- *Festlegung von Serien- und Sonderausstattung.* Die Definition einer neuen Baureihe ist nicht nur die Definition eines einzelnen neuen Produktes, sondern die Definition einer Vielzahl von Produkten. Die Festlegung, welche Features Serienausstattung oder Sonderausstattung sind, ist hier von zentraler Bedeutung. Diese Festlegung ist typischerweise für mehrere Produktvarianten gesondert zu treffen (Beispiel: Für den C180 ist ein Automatikgetriebe eine Sonderausstattung, während sie für den C320 Serienausstattung ist).

Eine wesentliche Schwierigkeit beim Aufbau einer Featureliste zur Produktdefinition ist die Wahl aller Themenfelder, die in der Featureliste berücksichtigt werden müssen. Insbesondere hat sich gezeigt, dass eine Fokussierung auf funktionale Bestandteile nicht ausreichend ist. Beispielsweise unterscheiden sich die Fensterheber in den verschiedenen Baureihen funktional nur geringfügig. Bezüglich ihrer Umsetzung (und den damit verbundenen Produktkosten) lassen sich jedoch größere Unterschiede beobachten. Die Gründe liegen beispielsweise in den gewählten mechanischen und technischen Lösungen, die eine Auswirkung auf die einzusetzenden Sensoren haben.

Zuordnung auf Produktbestandteile

Basierend auf einer Gesamt-Featureliste gilt es festzulegen, welche Umfänge auf welchen Produktteilen (z.B. Steuergeräten im Falle einer Gesamt-E/E-Featureliste) umzusetzen sind. In der Regel ist es hier nicht ausreichend, einzelne Features zu Produktbestandteilen zuzuordnen. Vielmehr zerteilt sich ein Feature in mehrere Bestandteile, die verschiedenen Produktbestandteilen zugeordnet werden. Nachfolgendes Beispiel soll dies verdeutlichen.

Beispiel: Ein Feature auf Ebene des Gesamtfahrzeugs ist beispielsweise „Außenlicht“. Dies umfasst unter anderem die Ansteuerung der Frontscheinwerfer auf Basis des Lichtschalters. Während der Lichtschalter an ein zentrales Signalmodul angeschlossen ist, werden die Frontscheinwerfer durch ein eigenes Frontmodul angesteuert. Die Betriebsanzeige der Außenbeleuchtung wird im Kombi-Instrument realisiert (siehe auch Abbildung 8).

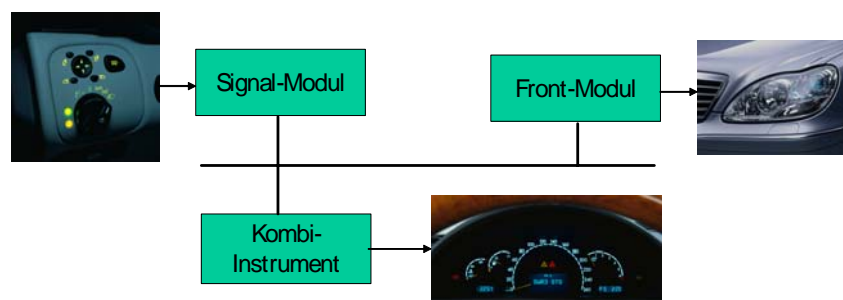


Abbildung 8: Verteilung des Features „Außenlicht“ auf mehrere Steuergeräte.

Featurelisten als Medium zur strategischen Produktplanung sind für den Bereich E/E-Systeme noch relativ neu. Basierend auf produktbezogenen Featurelisten entwickeln sich jedoch zunehmend umfassendere Featurelisten. Wenngleich die Berücksichtigung von Abhängigkeiten hier eine deutlich größere Bedeutung hat als bei produktbezogenen Featurelisten, hat sich bisher eine einfache Listenform als Mittel der Wahl behauptet.

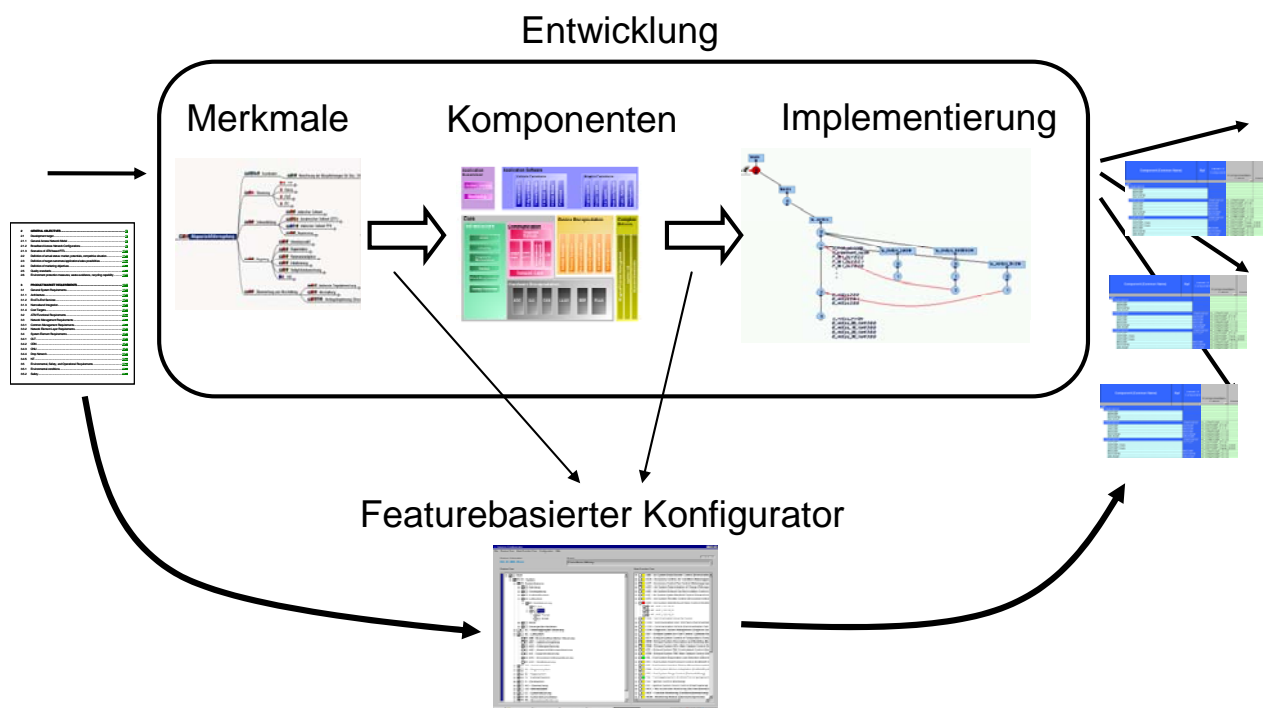
3.5 Einsatz von Features zur Software-Produktkonfiguration

Während in Kapitel 3.2 der Einsatz von Features zur Produktplanung und -verfolgung auf Anforderungsebene beschrieben ist, können Features auch zur Produktkonfiguration auf Anforderungsebene eingesetzt werden. Durch geeignete Abbildung ist dann eine Produktkonfiguration auf Code-Ebene möglich, d.h. zum Zusammenstellen von Konfigurationen.

Dieser Einsatz ist besonders dann interessant, wenn viele Produkte auf der gleichen Plattform basieren. Für eingebettete Systeme stellt die Hardware oft eine Plattform dar, von der aus Rentabilitäts-Gründen große Stückzahlen verkauft werden müssen. Die kundenspezifischen Ausprägungen werden durch Varianten in der Hardware, aber vorrangig in der Software realisiert. Diese Charakteristik ist bei vielen Bosch-Produkten, insbesondere bei Steuergeräten für Kraftfahrzeuge gegeben (Motorsteuerung, Radio, Airbag-Steuerung etc.).

Zum Strukturieren der Merkmale wird eine Feature-Modellierung nach FODA eingesetzt (vgl. Kapitel 2). Die Abbildung verdeutlicht das grundsätzliche Vorgehen, in welchem drei Schritte bearbeitet werden müssen:

1. Strukturieren der Merkmale (Feature-Modellierung)
2. Abbildung von Merkmalen auf Komponenten (Mapping)
3. Konfiguration von Produkten auf Basis von Merkmalen (Merkmal-Stücklisten Konfigurator)



Durch die Anwendung von FODA wird es möglich, die Varianz auf Systemebene und auf Anforderungsebene für jede einzelne Komponente des Systems abzubilden. Diese Varianz hat Einfluss auf die Architektur des Systems und über jedes einzelne Element der Architektur auf die Implementierung der Komponenten. Die Software muss im Code die Varianz abbilden können, die in der Architektur vorgesehen ist.

Liegt eine Umsetzung der gesamten Modellierung in Form von Software-Komponenten vor, ist eine Erstellung von Produkten in folgender Form denkbar:

1. Der Vertrieb stimmt dem Kunden anhand der Merkmalsliste ein System ab, mit dem die Anforderungen abgedeckt werden. Durch die Abbildung von Features auf Komponenten wird sichergestellt, dass Merkmale verfügbar oder geplant sind.
2. Mit Hilfe des Featurebasierten Konfigurators werden die Varianten der Code-Module zusammengestellt. Daraus entsteht eine Liste der Code-Module, die zu einem Produkt integriert werden.

Der Nutzen einer schnellen Bereitstellung von kundenorientierten Software-Lösungen ist unübersehbar. Die Risiken liegen im immensen Aufwand für die Modellierung, die eine hohe Abdeckung aller Möglichkeiten sicherstellen muss. Außerdem ist dadurch nicht gewährleistet, dass die integrierten Software-Module wirklich lauffähig sind. Bei der Variantenvielfalt (ca. 1200 Software-Stände pro Jahr) und Anzahl der Software-Module (ca. 5000 je Ablieferung) muss deshalb ein praktikabler Weg für die Anwendung von Features zur Modellierung gefunden werden. Eine mögliche Vorgehensweise wird in Abschnitt 4.3.3 beschrieben.

4 Auseinandersetzung mit dem Featurebegriff

In diesem Kapitel soll nun auf Basis der in Kapitel 3 skizzierten Anwendungsszenarien, die Einsatzgebiete für Features diskutiert werden. Zudem wird versucht die Verwendung des Featurebegriffs in unterschiedlichen Kontexten zu erläutern und basierend auf den im Arbeitskreis dargestellten Szenarien, Gütekriterien für Featuremodelle zu definieren. Darüber hinaus werden in den Abschnitten 4.3.1 und 4.3.2 theoretische Einsatzszenarien skizziert, die aus wissenschaftlichen Arbeiten hervorgegangen sind.

4.1 Einsatzgebiete für Features

In diesem Abschnitt soll nun von den in Abschnitt 3 vorgestellten Kontexten und den darin skizzierten Einsatzgebieten abstrahiert werden. Es wird versucht die identifizierten Einsatzgebiete kontextunabhängig, detaillierter zu beschreiben und nach Möglichkeit Ebenen zuzuordnen. Weiterhin sollen Beziehungen zwischen den einzelnen Einsatzgebieten aufgezeigt werden. Die Beziehungen drücken aus, wie die in einem Einsatzgebiet erstellen Feature-Modelle in anderen Einsatzgebieten genutzt werden können.

4.1.1 Einsatzgebiete als Dimension zur Charakterisierung von Features

Aufgrund der recht vagen Definition eines Features des FODA-Ansatzes, der unzähligen weiteren Definitionen eines Features, welche in der Literatur gefunden werden können und den unterschiedlichsten Interpretationen der beteiligten Stakeholder, wird eine Kommunikation über Features wesentlich erschwert.

Ein weiteres Problem besteht darin, dass der Feature-Begriff aus dem FODA-Ansatz aus der Domänenanalyse stammt. Da sich das Einsatzgebiet für Features aber nicht mehr auf die Domänenanalyse beschränkt, sondern Features in den unterschiedlichsten Einsatzgebieten verwendet werden, sollte eine Begriffsdefinition unter Berücksichtigung des Einsatzgebietes erfolgen. Somit kann das Einsatzgebiet als eine Dimension zur Charakterisierung eines Features herangezogen werden.

Nach Analyse der in Abschnitt 3 präsentierten Szenarien für die Feature-Modellierung, lassen sich hier insgesamt sechs Einsatzgebiete identifizieren:

- *Portfolioplanung*
 - Erstellung eines Portfolios für ein oder mehrere Produkte
- *Domänenmodellierung*
 - Erstellung eines Domänenmodells
- *Produktinstanziierung* bei gleichzeitiger Plattformentwicklung im Produktlinienkontext

- Wird bei einer Produktlinienentwicklung explizit eine gemeinsam verwendete Plattform entwickelt, auf der alle Mitglieder der Produktlinie (Produkte) basieren, so kann unter Verwendung der Featuremodelle eine Selektion der Features durchgeführt werden, die in einem konkreten Produkt enthalten sein sollen
- *Projektmanagement*
 - Erstellung eines Projektplans auf Basis der zu realisierenden Features
- *Architekturabbildung*
 - Mapping von Features aus der Anforderungsebene auf Architekturkomponenten
- *Auftragsvergabe*
 - Auftragsvergabe an Zulieferer oder Erstellung von Kundenangeboten

Insgesamt lassen sich die oben beschriebenen Einsatzgebiete grob drei Ebenen zuordnen. Tabelle 3 fasst diese zusammen.

Ebene	Einsatzgebiet
Portfoliomanagement	Portfolioplanung und -management
Domänenmodellierung	Domänenanalyse- und modellierung Produktinstanziierung bei Plattformentwicklung
Projektmanagement	Projektmanagement Architekturabbildung Auftragsvergabe

Tabelle 3: Einsatzgebiete für Features.

Auf der Ebene des Portfoliomanagements sind all diejenigen Einsatzgebiete zusammengefasst, in denen Featuremodelle genutzt werden, um Produkteigenschaften festzulegen, Produktabgrenzungen zu definieren und um ein allgemeines Roadmapping abzustimmen.

Aktivitäten zur Erschließung einer oder mehrerer Domänen, welche durch die Produkte adressiert werden, sind in der Ebene Domänenmodellierung zusammengefasst. Dies schließt ebenfalls eine Produktinstanziierung mit ein, falls diese auf Basis des Domänenmodells durchgeführt wird. Bei einer Plattformentwicklung zeichnet sich das Domänenmodell durch gemeinsame und variable Merkmale aus. Gemeinsame Merkmale sind daher typischerweise in der Plattform zu finden, während variable Merkmale produktspezifisch sind und somit nicht Bestandteil aller Produkte sind.

Auf der Ebene des Projektmanagements sind alle Einsatzgebiete zusammengefasst, die sich mit der Projektplanung und -steuerung beschäftigen. Die Festlegung, welche Features auf welche Architekturkomponenten abgebildet werden, werden hier ebenfalls zur Ebene des Projektmanagements gezählt, da hier vor allem das Projektcontrolling und die Traceability eine wichtige Rolle spielen.

Abbildung 10 zeigt, in wie weit die einzelnen Teilgebiete in Beziehung stehen. Ein gerichteter Pfeil bedeutet, dass die Ergebnisse der Featuremodellierung aus einem Einsatzgebiet in einem anderen Einsatzgebiet genutzt werden können.

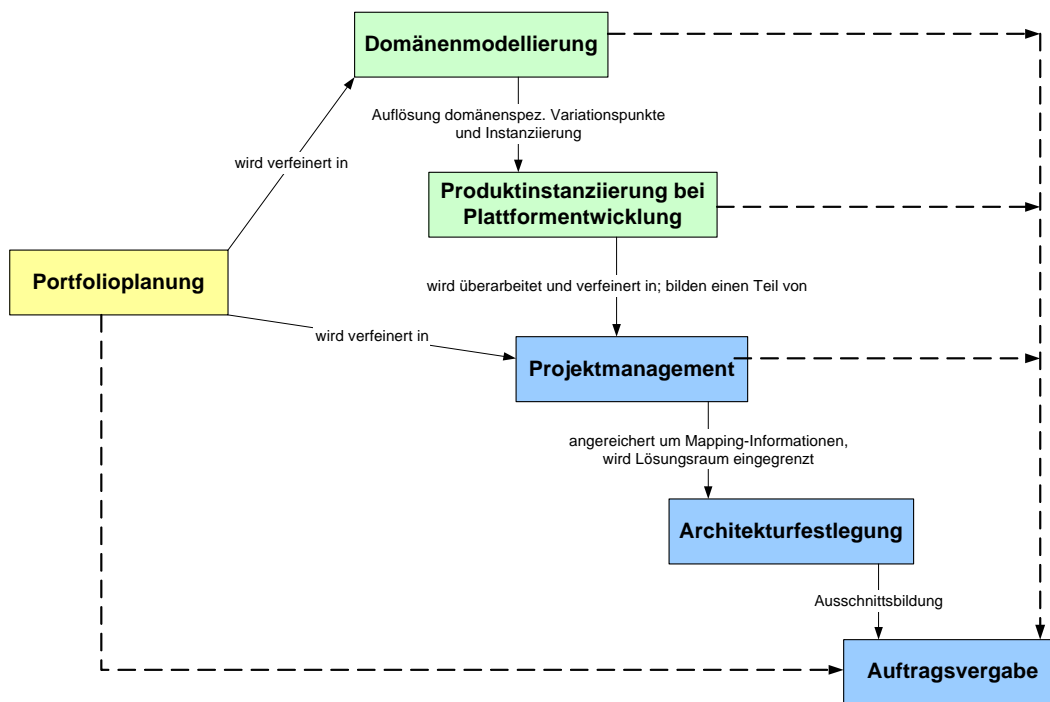


Abbildung 10: Beziehungen zwischen Einsatzgebieten

Dies bedeutet, dass die Modelle, die während der Portfolioplanung erstellt worden sind, als Ausgangspunkt für die Domänenanalyse und das Projektmanagement verwendet werden können. Die Modelle werden dann, um weitere Features und Beziehungen erweitert und existierende verfeinert.

Die Domänenmodelle sollten als Basis für die Produktinstanziierung dienen. Wird eine explizite Plattformentwicklung durchgeführt, so sollten die gemeinsamen Features im Domänenmodell entsprechend modelliert sein, zum Beispiel durch obligatorische Features oder Implikationen. Variable Features werden entsprechend als Optionen, Alternativen oder Oder-Auswahlen modelliert. Während der Produktinstanziierung muss nun festgelegt werden, welche variablen Features zu einem Produkt gehören sollen und welche nicht. Dies geschieht durch Auflösung der Variationspunkte. Variationspunkte kennzeichnen hier eine Stelle im Domänenmodell, an dem eine Auswahl getroffen werden muss, welches Subfeature bzw. welche Subfeatures zu einem konkreten Produkt gehören sollen. Variabilität bezeichnet also hier die unterschiedlichen möglichen Ausprägungen eines Produktes. Dies bedeutet, dass nach Abschluss einer Produktinstanziierung alle Variationspunkte aufgelöst worden sind und somit keine Variationspunkte mehr im Produktmodell vorhanden sind.

Die Modelle der fertig konfigurierten Produkte können dann im Projektmanagement dazu verwendet werden, Ressourcen zu planen und Verantwortlichkeiten festzulegen. Schließlich kann dieses verfeinerte Modell um Mapping-Informationen ergänzt werden, um die Abbildung von Features auf der Anforderungsebene auf Architekturbausteine zu definieren. Schließlich kann auf Basis der Architekturabbildung eine Auftragsvergabe, z.B. an Zulieferer vergeben werden. Die Auftragsvergabe kann hingegen auf Basis aller erstellten Modelle in den unterschiedlichen Einsatzgebieten durchgeführt werden.

Die in Kapitel 3, von den Mitgliedern skizzierten, Szenarien zeigen deutlich, dass nicht in jedem Entwicklungskontext, die Featuremodelle in allen Einsatzgebieten entsprechend genutzt werden. Die oben beschriebenen Beziehungen zeigen hier eine potentiell mögliche Verwendung der Featuremodelle in den entsprechenden Einsatzgebieten auf und skizzieren potentielle Beziehungen zwischen diesen.

Im folgenden Abschnitt sollen nun die identifizierten Einsatzgebiete näher diskutiert werden. Dazu wurde ein Charakterisierungsschema entwickelt, welches benutzt wird, um die Einsatzgebiete zu beschreiben.

4.1.2 Charakterisierung der Einsatzgebiete

Um die identifizierten Einsatzgebiete näher zu beschreiben, wird als gemeinsame Basis ein Charakterisierungsschema verwendet. Das Schema enthält die folgenden Kategorien:

- *Autor*: Wer ist für die Erstellung und Wartung des Modells verantwortlich?
- *Adressat*: Wer ist der Empfänger, bzw. Nutzer des Modells?
- *Zielsetzung*: Was ist der Verwendungszweck des Modells?
- *Artefakt*: Welche Struktur und welche Informationen enthält das Ergebnismodell?

Die einzelnen Einsatzgebiete werden in Tabelle 4 bis Tabelle 9 skizziert:

Einsatzgebiet	Portfolioplanung
Autor	Marketing und Technologiemanager
Adressat	<ul style="list-style-type: none"> • Marketing und Technologiemanager • Top-Management
Zielsetzung	<ul style="list-style-type: none"> • Erstellung eines Produktportfolios und zur Definition der Produktabgrenzung • Kosten- und Nutzenanalyse, insbesondere Marktpotentiale und Benefit für Wiederverwendung zu bestimmen • Featuremodell als Basis für die Machbarkeitsstudie • Roadmapping für aktuelle und zukünftige Produkte festlegen
Artefakt	Strukturierte Featureliste mit Information, welches Feature in welchem Produkt (wann) enthalten ist.

Tabelle 4: Einsatzgebiet Portfolioplanung.

Einsatzgebiet	Domänenmodellierung
Autor	<ul style="list-style-type: none"> • Domänenexperten / -analysten • Plattformmanager bei Plattformentwicklung
Adressat	<ul style="list-style-type: none"> • Asset-Entwickler • Oberes Management • Produktmanager
Zielsetzung	<ul style="list-style-type: none"> • Definition der Plattform (inkl. Kostenabschätzung für generisch und individuell zu entwickelnde Features) • Inkrementelle Entwicklung ermöglichen • Identifikation bereits modellierter Basisfeatures • Ableitung einer konsistenten Produktdefinition ermöglichen
Artefakt	Strukturiertes Featuremodell als Teil des Domänenmodells. Explizite Modellierung von Gemeinsamkeiten und Variabilitäten.

Tabelle 5: Einsatzgebiet Domänenmodellierung.

Einsatzgebiet	Produktinstanziierung
Autor	<ul style="list-style-type: none"> • Produktmanager • Plattformmanager
Adressat	<ul style="list-style-type: none"> • Produktentwickler
Zielsetzung	<ul style="list-style-type: none"> • Erstellung einer Produktdefinition für ein konkretes Produkt • (Semi-)automatische Instanziierung von Code bzw. Dokumentation ermöglichen
Artefakt	Featurelisten, welche die Features des jeweiligen Produkts beschreiben.

Tabelle 6: Einsatzgebiet Produktinstanziierung.

Einsatzgebiet	Projektmanagement
Autor	<ul style="list-style-type: none"> • Projektmanager (Input von Technologieverantwortlichen)
Adressat	<ul style="list-style-type: none"> • Projektkunde (Sponsor, oberes Management) • Projektteam
Zielsetzung	<ul style="list-style-type: none"> • Releaseplanung • Leitfaden für Projekt • Verfolgung des Projektstatus
Artefakt	Verfeinerte, strukturierte Featureliste, ergänzt um Projektmanagementinformationen, wie Releasedaten, Verantwortlichkeiten, Deployment, ...

Tabelle 7: Einsatzgebiet Projektmanagement.

Einsatzgebiet	Architekturabbildung
Autor	<ul style="list-style-type: none"> • Lead-Architekt
Adressat	<ul style="list-style-type: none"> • Komponentenarchitekt / -entwickler • ggf. Zulieferer
Zielsetzung	<ul style="list-style-type: none"> • Festlegung von Designentscheidungen explizit • Umgebung und Schnittstellen der Komponenten festlegen • Machbarkeit Komponentenentwicklung
Artefakt	Verfeinerte, strukturierte Featureliste, ergänzt um Projektmanagementinformationen, wie Releasedaten, Verantwortlichkeiten, Deployment, ...

Tabelle 8: Einsatzgebiet Architekturabbildung.

Einsatzgebiet	Auftragsvergabe
Autor	<ul style="list-style-type: none"> • Produktmanager • Projektmanager (Features auf Systemebene)
Adressat	<ul style="list-style-type: none"> • Lieferanten
Zielsetzung	<ul style="list-style-type: none"> • Abstimmung mit Zulieferer • Change Request Management • Abnahmekriterien • Erfassung von Constraints
Artefakt	Featureliste (einfach oder strukturiert), welche die zu liefernden Features beschreibt.

Tabelle 9: Einsatzgebiet Auftragsvergabe.

4.1.3 Konsequenzen für Features aus den Einsatzgebieten heraus

Je nach dem Einsatzgebiet für Features ergeben sich daraus unterschiedliche Anforderungen an deren Detaillierung und Attributierung. Die folgende Tabelle 10 zeigt eine Reihe typischer Attribute auf und skizziert deren Bedeutung für die verschiedenen Einsatzgebiete. Die relative Bedeutung wird dabei über Punkte dargestellt. Kein Punkt bedeutet ‚Keine Bedeutung‘, drei Punkte bedeutet ‚Hohe Bedeutung‘.

<i>Einsatzgebiet</i>						
<i>Feature-Attribut</i>	Portfolioplanung	Domänenmodellierung	Produktinstanziierung	Projektmanagement	Architekturabbildung	Auftragsvergabe
Benchmark-Aussage zu Wettbewerbern	● ● ●					
Kosten				● ● ●		
Release, in dem das Feature umzusetzen ist				● ● ●		● ●
Argumente für den Kundennutzen						
Priorität						
Bezug zu Vorschriften/Normen/Gesetzen	● ●					

Tabelle 10: Bedeutung von Feature-Attributen in den verschiedenen Einsatzgebieten .

4.1.4 Implikationen aus CMMI für Features beim Einsatzgebiet Projektmanagement

In diesem Abschnitt soll nun aufgezeigt werden, in wie weit die Feature-Modellierung den Anforderungen an den Entwicklungsprozess nach CMMI gerecht wird. Das Capability Maturity Modell Integrated (CMMI) des Software Engineering Institutes (SEI) baut die Projektplanung und –verfolgung sinnvollerweise auf die bekannten Anforderungen an ein Produkt bzw. Projekt auf. Anforderungen bilden die Basis, für das, was zu liefern und damit was zu planen und zu verfolgen ist. Deshalb ist die erste Specific Practice in CMMI Level 2 in der erstgenannten Process Area (PA) Requirements Management (REQM):

REQM SPI.1: Obtain an Understanding of Requirements

Develop an understanding with the requirements providers on the meaning of the requirements [CMMI, V1.1]

Wie bereits beschrieben stellen Features hierfür ein ideales Medium dar, da sie die erlebbaren Eigenschaften eines Produktes darstellen. Zudem reduzieren sie die Anzahl der Anforderungen auf eine beherrschbare Zahl, die als Basis für die Planung von Projekten verwendet werden kann. Bevor in die Planung von Ressourcen eingestiegen werden kann, fordert CMMI die Abschätzung von Größen auf Basis aller relevanten Parameter, insbesondere der Größen, die sich aus den Anforderungen ergeben (CMMI PA Project Planning (PP)).

PP SG 1 Establish Estimates

Estimates of project planning parameters are established and maintained.

... Estimates of planning parameters should have a sound basis to provide confidence that any plans based on these estimates are capable of supporting project objectives.

Factors that are typically considered when estimating these parameters include the following:

- *Project requirements, including the product requirements, the requirements imposed by the organization, the requirements imposed by the customer, and other requirements that impact the project ... [CMMI, V1.1]*

Auch die Verfolgung von Projekten soll mit der Abarbeitung von Arbeitspaketen die Erfüllung von Anforderungen sicher- und damit den Kunden zufrieden stellen. Als festen Bestandteil der Planung wird in CMMI PP SP1.1 eine Work Breakdown Structure (WBS) gefordert. Diese legt die Arbeitsstruktur des Projektes fest, um die inhaltliche und terminliche Abwicklung des Projekts sicherzustellen. Den Beitrag von Features als Werkzeug zur Planung und Steuerung von Projekten ist in den Kapiteln 3.2 und 3.3 ausführlich beschrieben.

In Level 2 wird für das Configuration Management (CM) die Festlegung und Verfolgung von Baselines gefordert, d.h. von integrierenden Ständen, in denen alle Arbeitsprodukte einer Lieferung (Anforderungs-Spezifikation, Entwurf, Produkt, Testlisten, etc.) eindeutig identifizierbar und zu einander zugeordnet sind.

SG 1 Establish Baselines

Baselines of identified work products are established.

SG 2 Track and Control Changes

Changes to the work products under configuration management are tracked and controlled.

SG 3 Establish Integrity

Integrity of baselines is established and maintained

Ein Beispiel für die Anwendung von Features zu diesem Zweck ist in Kapitel 3.3 angerissen. Hier werden Features verwendet, um bereits während der Planung Lieferumfänge festzulegen. Durch gezielte Planung der Baselines je Liefertermin werden basierend auf dem Featureumfang der Lieferung festgelegt, welche weiteren Arbeitsprodukte in welcher Detaillierung vorliegen müssen. Bis zur Lieferung können dann die notwendigen Arbeitsprodukte erzeugt, mit einander in Beziehung gesetzt und als Baseline ausgeliefert bzw. abgelegt werden. Änderungen seitens des Kunden oder innerhalb der Entwicklung beziehen sich dann immer auf eine bestehende Baseline oder eine Baseline-Planung. In anderen Worten: Änderungen an Features haben Auswirkungen auf deren Realisierung. Diese Änderungen an der Realisierung führen zu neuen Schätzungen der Arbeitsumfänge und haben damit Einfluß auf die Liefertermine und – Umfänge. Diese auf Features bezogenen Änderungen werden in der Projekt- und Baseline-Planung eingearbeitet und dann entsprechend der Planungen abgearbeitet.

CMMI Level3 beschäftigt sich intensiver mit dem Engineering, d.h. mit der inhaltlichen Entwicklung von Produkten. Auch hier können Features in den bisher diskutierten Ausprägungen als Hilfsmittel verwendet werden. In der PA Requirements Development wird die Erfüllung folgender Ziele gefordert:

SG 1 Develop Customer Requirements

Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements.

SG 2 Develop Product Requirements

Customer requirements are refined and elaborated to develop product and product-component requirements.

SG 3 Analyze and Validate Requirements

The requirements are analyzed and validated, and a definition of required functionality is developed.

Die Verwendung von Features zur Erfüllung des ersten Ziels ist im gesamten Kapitel 3 umfassend beschrieben. Mit der Arbeit an einem gemeinsamen Verständnis über die Eigenschaften eines Produktes, bieten Features auch ein passendes Werkzeug zur Validierung von An-

forderungen. Nur wenn sichergestellt ist, dass der Kunde auch das bezahlt, was er wirklich haben möchte, hat der Zulieferer einen Wert geschaffen.

Dass die Modellierung und Verwendung von Abhängigkeiten zwischen Features für die Entwicklung von Produkten von Nutzen ist, kommt besonders bei Plattform-Entwicklungen zum tragen. Da in der Produktableitung aus der Plattform immer ähnliche Entwicklungsschritte bei der Zuweisung von Produkt-Anforderungen auf Komponenten-Anforderungen vollzogen werden, können Zuweisungen von Features zu Komponenten und die Abbildung der Abhängigkeiten zwischen Features auf Abhängigkeiten zwischen Komponenten nutzbringend angewendet werden (vgl. 3.4). Ein größerer Nutzen entsteht bei der Integration verschiedener Komponenten und in der Verifikation. Wenn Fehler bei einem Feature auftreten, kann rückverfolgt werden, welche Komponenten an der Umsetzung des Features beteiligt sind. Mit der Festlegung der Features kann bei vollständiger Abbildung auf Komponenten direkt zur Integration geschritten werden. Diesen hohen Grad an Wiederverwendung fordert CMMI nicht. Eine Plattformentwicklung macht nur Sinn, wenn sie zum Markt und zu den Geschäftszielen passt. Generell kann aber gezeigt werden, dass Features die vom CMMI geforderten Prozesselemente unterstützen.

4.2 Versuch einer Begriffsdefinition

In diesem Abschnitt wird nun versucht aus der Betrachtung der in den vorhergehenden Abschnitten aufgezeigten Einsatzgebiete eine Feature-Begriffsdefinition abzuleiten.

Wer sich mit dem Featurebegriff und seiner Verwendung in der Praxis auseinandersetzt, wird frühzeitig feststellen, dass je nach Anwendung etwas Anderes unter dem Begriff verstanden wird. Dies spiegelt sich auch in der entsprechenden Literatur wieder. Zielsetzung dieses Arbeitskreises war damit auch Klarheit über den Begriff „Feature“ zu erhalten. Schon zu Beginn des Arbeitskreises wurde darüber diskutiert ob es sinnvoll bzw. überhaupt möglich ist **eine** Definition für den Feature Begriff festzuschreiben. Wie zu erwarten war, wurden sich die Mitglieder des Arbeitskreises einig darüber, dass es **eine** Definition des Feature Begriffs nicht geben soll, da sich die Definition stark nach dem Zweck, wofür man Features einsetzen möchte richtet. Daher werden im Folgenden die Gemeinsamkeiten und Unterschiede der Features, je nach Verwendungszweck bzw. Einsatzgebiet (siehe Tabelle 2: Einsatzgebiete für Features) herausgestellt.

4.2.1 Gemeinsamkeiten der Features auf verschiedenen Ebenen

Folgende Eigenschaften zeichnen Features unabhängig vom Verwendungszweck aus:

- Features werden einsatzübergreifend als Kommunikationsmedium an Schnittstellen zwischen verschiedenen Aufgaben der Produktentwicklung verwendet. Mit Hilfe von Features werden Charakteristiken von Produkten (Systemen, Komponenten) beschrieben, diskutiert, bewertet und festgelegt.
- Charakteristiken können sein: Funktionalitäten, Qualitätseigenschaften, Architekturen, Entitäten, wie zum Beispiel Daten oder Datenstrukturen und andere anwendungsspezifische Produkteigenschaften. Dies gilt sowohl für Hardware als auch für Software.
- Ein Feature besitzt Atomarität. Es ist vorhanden oder nicht vorhanden.
- Insbesondere dienen Features dem Vergleichen und der variablen Definition von Produkten. So werden anhand von Feature-Modellen verschiedene Produktvariationen (Entwürfe oder Kompositionen) hinsichtlich Nutzen und Wirtschaftlichkeit bewertet und ausgewählt.
- Erstellte Featuredefinitionen (Feature-Modelle) von Produkten können als Grundlage für die Auftragsvergabe zwischen den beteiligten Parteien genutzt werden.

- Ein Feature - eine Produktcharakteristik – wird durch einen abstrakten Oberbegriff (Sammelbegriff) beschrieben. Dieser repräsentiert Konzepte, Modelle oder Sachverhalte die in den Köpfen der beteiligten Personen erarbeitet oder in entsprechenden Dokumenten spezifiziert sind. (Damit hängt der Erfolg der Kommunikation mittels Features von dem Grad der Übereinstimmung der Interpretation der Featurebegriffe ab.)
- Features können Beziehungen zu anderen Features besitzen.
- Werden Features – als Oberbegriffe – explizit (z.B. in einem Dokument) spezifiziert, so sind dadurch Assoziationen und/oder Verfeinerungsbeziehungen zu weiteren Spezifikationselementen – insbesondere Anforderungen - von Produkten definiert. Features, erlauben somit den Zugriff auf Teile der feingranularen Anforderungs- oder Systemspezifikation.

4.2.2 Bestimmung der Featureeigenschaften hinsichtlich der Unterscheidungsdimensionen

Features unterscheiden sich insbesondere in Abhängigkeit von ihrem Verwendungszweck und Einsatzgebiet der Produktentwicklung (Einsatzgebiete des Portfoliomanagements, der Domänenmodellierung und des Projektmanagements – siehe Tabelle 3: Einsatzgebiete für Features.). Je nach Ebene unterscheiden sich Features bezüglich:

- der Stakeholder, die mit den Features kommunizieren und Produktcharakteristika definieren. (Kunden, Marketing, Projektmanager, Systemingenieure, Entwickler, etc.).
- der Einheit, auf die sich das Feature bezieht (Produktlinie, Produkt, Gesamtsystem, Einzelkomponente, etc.).
- dem Grad in dem sie zur Beschreibung von Variabilitäten eingesetzt wird. So werden etwa in der Domänenmodellierung variable Produktkompositionen durch Featurelisten bestimmt, während im Bereich der Architekturabbildung der Aspekt der Variabilität in den Hintergrund tritt. Hier geht es in erster Linie um die Beschreibung und Bewertung von Designentwürfen bzw. Alternativen.
- der Abstraktionsebene im Vergleich zu Anforderungen. Auf der Ebene des Portfoliomanagements bezeichnen Features meist markt-/kundenorientierte Schlüsselanforderungen auf abstrakter Ebene. Sie müssen im Bereich der Domänenmodellierung und des Projektmanagements auf konkrete system- und technologiebezogene Features und Anforderungen abgebildet werden. Insbesondere im Bereich des Projektmanagements fassen Features bestimmte Mengen von detaillierten Systemanforderungen und Eigenschaften zusammen, um sie bezüglich Machbarkeit und Aufwandsschätzung bewertet zu können.

4.3 Theoretische Einsatzszenarien

In den folgenden Abschnitten werden nun exemplarische Einsatzszenarien der Feature-Modellierung skizziert, die im Rahmen des Arbeitskreises diskutiert wurden.

4.3.1 Weitergehendes Szenario: Verwendung von Features bei der Anforderungserhebung mit Rationale

Features ermöglichen wie bereits oben erwähnt, eine kompakte Beschreibung eines Bündels von Anforderungen. Weiterhin ist es möglich Beziehungen zwischen Features zu verwalten. Dies ist insbesondere auch interessant während der Anforderungserhebung, in der man verschiedene Alternativen diskutiert, welche Aufgaben das System wie unterstützen kann. Diese Anforderungserhebung kann als Teil der Einsatzgebiete Portfolioplanung oder Domänenmodellierung erfolgen.

Im Folgenden wird ein hypothetisches Beispiel skizziert. Dabei werden Features als Erweiterung des in [Dutoit, Paech02] skizzierten Ansatzes zur integrierten Erhebung und Spezifikation von Use Cases und Rationale verwendet.

In dem Beispiel wird ausgehend von zu unterstützenden Arbeitsaufgaben (Af) der Benutzer versucht verschiedenen Systemfeatures zu identifizieren. Die Systemfeatures werden auf 2 Ebenen betrachtet: Grobfeatures (GF) beschreiben einen Bereich der Systemunterstützung, Detailfeatures (DF) beschreiben einzelne Funktionen. Details der Funktionen werden durch einzelne Anforderungen (DA) beschrieben. Use Cases (UC) werden verwendet, um die GF besser zu verstehen und DF aus Ihnen abzuleiten. Zur Diskussion von Alternativen werden Qualitätsanforderungen (QA) auf allen Ebenen (Aufgaben GF, DF, QA) verwendet. Abbildung 11 zeigt einen Ausschnitt, eines entsprechenden Graphen für die Diskussion der Parkunterstützung im Auto. Die Pfeile entsprechen „ist-realisiert“-Beziehungen.

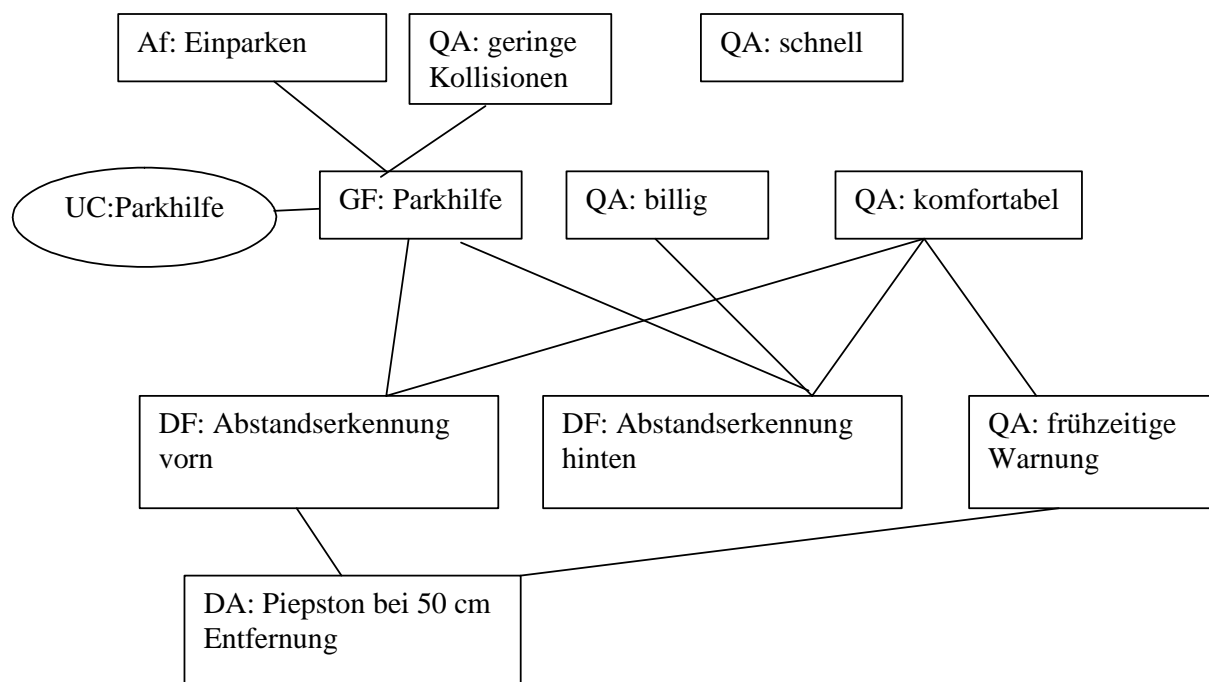


Abbildung 11: Zusammenhang zwischen Use Cases und Features am Beispiel "Parkhilfe"

Ausgangspunkt ist die Af „Einparken“ und die QA an diese Aufgabe: „geringe Kollisionen“. Eine Möglichkeit dies zu erreichen ist das GF „Parkhilfe“. Weiterhin entstehen in der Diskussion QA an das Grobfeature („billig“, „schnell“, „komfortabel“). Der genaue Beitrag des Systems bei der Parkhilfe ist noch nicht klar. Nun wird mit Hilfe von UC beschrieben, welche Detailfeatures mit welchen QA der Nutzer wie verwenden kann (z.B. „Verwendung einer Abstandserkennung mit frühzeitiger Benachrichtigung“). Hier zeigen sich nun mehrere Möglichkeiten, „Abstandserkennung nur hinten“ oder „Abstandserkennung nur vorne“. Weiterhin werden auch die QA verfeinert: „frühzeitige Warnung“ realisiert „komfortabel“. „Abstandserkennung vorn und hinten mit frühzeitiger Warnung“ realisiert eine „komfortable Parkhilfe“. Nur „Abstandserkennung hinten“ wäre eine „billige Parkhilfe“. Diese Beziehungen zwischen QA und GF bzw. DF können natürlich recht komplex werden, wenn verschiedene Features verschiedene QA verschieden gut erfüllen. Die DF sind noch nicht genau genug als Vorgabe für die Entwickler. Deshalb werden sie weiter in DA verfeinert. Im Beispiel verfeinert der „Piepston bei 50 cm Abstand“ die „frühzeitige Abstandserkennung vorn“. Auch hier kann es natürlich wieder viele Alternativen geben.

Der entstehende Graph ist (bis auf die UC) ein Featuremodell. Lässt man die QA weg, so entsteht ein Modell mit „ist realisiert“-Beziehungen zwischen Features. Dieses könnte man natürlich wie in der Domänenmodellierung auch wieder mit Abhängigkeiten zwischen Features erweitern. Im Produktlinienkontext würden hierbei auch Varianten an allen Stellen entstehen (also bei den Af, QA, GF, DF).

4.3.2 Feature zentriertes Requirements Management Informationsmodell

Ein zentraler Aspekt beim Einsatz eines werkzeuggestützten Requirements Engineering in der Kfz-Entwicklung betrifft die richtigen Strukturen und die Vorgehensweise. Basis dafür ist ein Modell der für den Kfz-Hersteller wichtigen Requirementstypen und Entwicklungsebenen auf denen diese abzulegen sind sowie der Beziehungen zwischen diesen. Ein am Arbeitskreis beteiligtes Unternehmen hat aufgrund mehrjähriger Begleitung von Entwicklungsprojekten ein solches Informationsmodell entwickelt, dass von den folgenden drei Requirementstypen ausgeht:

- Zu erreichende Ziele oder Eigenschaften werden mit Goals bzw. Features beschrieben, wie z.B. Umluftschießen.
- Interaktionsszenarien zwischen Funktionalitäten werden durch Use Cases beschrieben, wie z.B. alle möglichen Szenarien beim Umluftschießen. Interaktionsszenarien sind extrem wichtig um die Komplexität der Systeme zu verstehen und zu beherrschen.
- Die präzisen und detaillierten Anforderungen für Fahrzeugkomponenten werden als Produkt Requirements beschrieben, wie z.B.: Ein Innenraum-Steuergerät muss in einem bestimmten Arbeitsmodus innerhalb eines bestimmten Zeitraums, ein bestimmtes Signal erkennen oder schaltet sonst ab. Diese Art von Requirements wird insbesondere in Lastenheftspezifikationen verwendet.

Neben diesen drei Requirementstypen wird gemäß den organisatorischen und technischen Gegebenheiten zwischen vier Systemebenen beim Requirements Engineering (vgl. Abbildung 12) unterschieden:

- Die top-level Fahrzeugebene,
- die System- oder Funktionsgruppenebene
- die Ebene zur Entwicklung einer einzelnen Funktion
- die Komponentenebene, (könnte auch Integrationsebene heißen), wobei sich meistens auf die Abbildung von Funktionalität auf Komponenten und die Spezifikation der (Software) Schnittstellen zwischen diesen Komponenten beschränkt wird.

Aus diesen beiden Dimensionen wurde ein Requirements Management Informationsmodell entwickelt, in dem Szenarios, Features und Produkt Requirements auf allen Systemebenen abgedeckt werden, wie in dem Bild dargestellt ist. Zentrales Navigationselement dabei ist der hierarchisch organisierte Featurebaum, der u. a. zur Steuerung und Kontrolle der Projekte verwendet wird. Dieses Modell ist eine Vereinfachung, um die Grundidee zu illustrieren.

