

# Modellierung der Variabilität einer Software-Produktfamilie<sup>1</sup>

Günter Halmans, Klaus Pohl

Universität Essen, Institut für Informatik, Software Systems Engineering,  
Altendorfer Str. 97-101, 45117 Essen  
Email: {halmans, pohl}@informatik.uni-essen.de

**Abstract:** Variabilität ist das zentrale Konzept in der Produktfamilienentwicklung. Durch Ausnutzung der Produktfamilien-Variabilität können, basierend auf dem Produktfamilienkern, kundenspezifische Produkte realisiert werden. Der Erfolg einer Produktfamilie hängt neben der zu erreichenden Kundenakzeptanz davon ab, möglichst viele kundenspezifische Anforderungen unter Ausnutzung der Variabilität realisieren zu können. Hierfür ist u.a. die Kommunikation der Variabilität zum Kunden eine wesentliche Voraussetzung.

In diesem Beitrag erläutern wir die wesentlichen technischen und fachlichen Aspekte der Produktfamilien-Variabilität für eine Software-Produktfamilie und beschreiben ihre zentralen Eigenschaften. Wir zeigen zudem, wie Teile der fachlichen Variabilität durch Use Cases ausgedrückt und somit leichter zum Kunden kommuniziert werden können.

## 1 Einleitung

Die Entwicklung von Produktfamilien hat in den letzten Jahren immer mehr an Bedeutung gewonnen. Die Idee, Produkte auf einer gemeinsamen Basis zu entwickeln, verspricht Kostenreduktion, Effizienz, schnellere Entwicklungszyklen und damit kürzere Reaktionszeiten auf die Bedürfnisse des Marktes [CN01]. Eine Produktfamilie ermöglicht die Realisierung unterschiedlicher Produkte durch eine entsprechende Nutzung und Kombination der variablen Teile einer Produktfamilie sowie durch entsprechende kundenspezifische Erweiterungen. Die Variabilität einer Produktfamilie bezieht sich hierbei nicht alleine auf die Implementierung, sondern auf alle Phasen der Produktentwicklung.

### 1.1 Produktfamilienentwicklung: Domain und Application Engineering

Die Entwicklung einer Produktfamilie (Produktlinie) ist durch zwei Prozesse gekennzeichnet: Domain Engineering und Application Engineering (Abbildung 1). Im Domain Engineering werden die allgemeingültigen Artefakte und die einzelnen Variationsmöglichkeiten definiert. Im Application Engineering werden die individuellen, kundenspezi-

---

<sup>1</sup> Diese Arbeit wurde gefördert durch das BMBF “ (Förderkennzeichen 01 IS 002 C); Verbundprojekt CAFÉ, „From Concept to Application in System Family Engineering“; Eureka Σ! 2023 Programme, ITEA Projekt ip00004

fischen Produkte idealer Weise durch Wiederverwendung der Artefakte unter Ausnutzung der Variabilität erstellt.

Der Erfolg einer Produktfamilie aus der Sicht des Providers hängt zum einen von der Kundenakzeptanz und den damit zusammenhängend erzielbaren Erlösen ab. Zum anderen wird der Erfolg von dem letztendlich erzielten Wiederverwendungsgrad und den davon abhängigen Kosten der Entwicklung kundenspezifischer Produkte beeinflusst. Gelingt es, einen signifikanten Anteil der kundenspezifischen Anforderungen unter Verwendung der variablen und invariablen Teile der Produktfamilie zu realisieren, so ist der „Return on Investment“ der Produktfamilie hoch. Gelingt dies nicht, so entstehen bei der Entwicklung der produktfamilienbasierten Produkte höhere Kosten; d.h. der „Return on Investment“ sinkt.

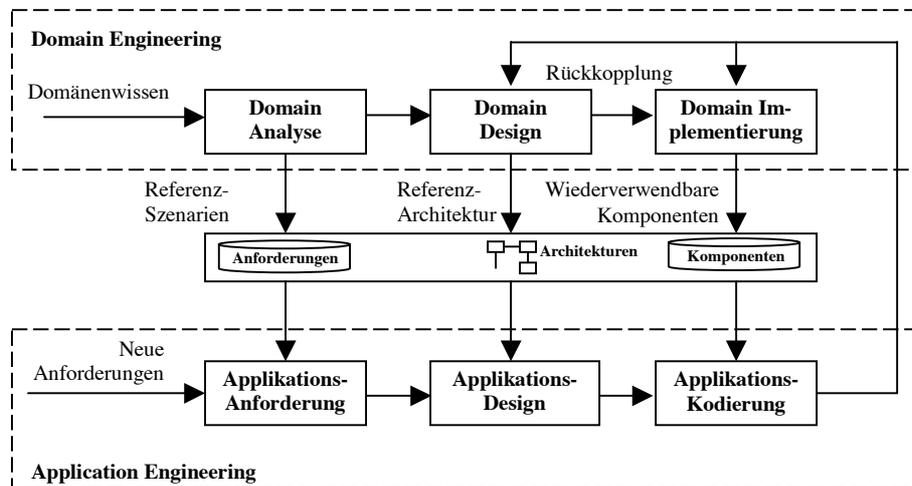


Abbildung 1: Domain- und Application Engineering in der Entwicklung von Produktfamilien [Es99]

## 1.2 Standard vs. Individualprodukte

Bei der Entwicklung von Produkten einer Software-Produktfamilie (im folgenden verstehen wir unter Produktfamilie eine Software-Produktfamilie) unterscheiden wir zwischen Standard- und Individualprodukten. In der Praxis ergeben sich neben diesen beiden klar abgegrenzten Produktarten Mischformen, jedoch lassen sich ihre jeweiligen Eigenschaften auf die beiden Produktarten zurückführen.

**Standardprodukte:** Standardprodukte werden von Anbietern der Produktfamilie definiert. Kundenanforderungen finden nur indirekt, beispielsweise über Marktstudien repräsentativer Kundengruppen oder einiger potenzieller Kunden Eingang in den Definitionsprozess. Eine direkte Integration von Kunden erfolgt in der Regel nicht. Standardprodukte wie beispielsweise Mobiltelefone werden typischerweise für Massenmärkte erstellt. Unterschiedliche Standardprodukte werden nur in einer begrenzten An-

zahl entwickelt; jedes dieser Produkte aber wird von einer großen Anzahl von Kunden eingesetzt.

**Individualprodukte:** Im Gegensatz zu Standardprodukten werden bei der Entwicklung von produktfamilienbasierten Individualprodukten kundenspezifische Anforderungen berücksichtigt. Ziel bei der Entwicklung von Individualprodukten ist es, eine möglichst hohe Abdeckung dieser Anforderungen durch die Artefakte der Produktfamilie zu erzielen. Es hat sich gezeigt, dass die frühzeitige Berücksichtigung der Artefakte in der Phase der Produktdefinition für den Erfolg der Produktfamilie wichtig ist [JGJ97]. Unterschiedliche Individualprodukte wie beispielsweise Radiologische Systeme entstehen in einer größeren Anzahl; jedes dieser Produkte aber wird jeweils nur bei einem kleinen Kreis von Kunden eingesetzt.

### 1.3 Herausforderungen an das Requirements Engineering

Im Rahmen der Anforderungsgewinnung für die Produkte einer Produktfamilie müssen die Möglichkeiten einer Produktfamilie berücksichtigt werden. Dies bedeutet, dass Funktionalität sowie Variabilität der Produktfamilie zum Kunden kommuniziert werden müssen. Hierbei ist insbesondere die Kommunikation der Variabilität der Produktfamilie, d.h. der individuellen, fast kostenneutralen Anpassbarkeit von zentraler Bedeutung. Da eine Produktfamilie typischerweise eine Vielzahl von Variationsmöglichkeiten besitzt, bedingt die Kommunikation dieser Variationsmöglichkeiten eine geeignete Präsentation und Repräsentation der Produktfamilien-Variabilität.

Eine erfolgreiche Kommunikation der Produktfamilien-Variabilität zum Kunden informiert den Kunden nicht nur über die „Auswahlmöglichkeiten“ der vorhandenen Produktfunktionalität und -qualität, sondern unterstützt den Kunden auch in den Trade-Off-Entscheidungen während der Gewinnung und Definition der kundenspezifischen Anforderungen. Werden im Anforderungsgewinnungsprozess zu einem Zeitpunkt mehrere Alternativen bezüglich der Funktionalität oder der Qualität (nicht funktionale Anforderungen) des zu erstellenden kundenspezifischen Produktes betrachtet, so ist die Kenntnis über die Variabilität der Produktfamilie und somit über die zu erwartenden Realisierungskosten für jede der betrachtenden Alternativen, ein wichtiges Entscheidungskriterium.

Szenarien (Use Cases) haben sich in der Einzel-Produktentwicklung mehrfach als geeignetes Mittel zur Kommunikation der Funktionalität eines Softwareproduktes bewährt [Ca95], [PBG01], [PH97], [We98]. Im Gegensatz zur Einzel-Produktentwicklung spielt jedoch in der Produktfamilienentwicklung die Variabilität der Produktfamilie eine zentrale Rolle. Für die Kommunikation dieser Variabilität zum Kunden existieren bisher keine erprobten Ansätze.

### 1.4 Überblick

In diesem Beitrag konzentrieren wir uns auf die Kommunikation der Funktionalität und Variabilität einer Produktfamilie zum Kunden. Wir betrachten hierbei die Entwicklung

von produktfamilienbasierten Individualprodukten. In Kapitel 2 unterscheiden wir zunächst zwischen zwei prinzipiellen Arten der Produktfamilien-Variabilität, und zwar zwischen technischer und fachlicher Variabilität. Für die Kommunikation mit dem Kunden sind insbesondere die fachlichen Variabilitätsaspekte von Interesse, die wir in Kapitel 3 ausführlich betrachten. In Kapitel 4 skizzieren wir, wie fachliche Variabilitäten einer Produktfamilie auf Use Cases und Szenarien abgebildet und dadurch zum Kunden kommuniziert werden können. In Kapitel 5 fassen wir die Ergebnisse des Beitrages zusammen.

## 2 Variabilität einer Produktfamilie

Die Variabilität in der Produktfamilienentwicklung definiert David M. Weiss wie folgt:

*„...An assumption about how members of a family may differ from each other” [WL99].*

Die Variabilität einer Produktfamilie wird durch sogenannte Variationspunkte charakterisiert. Ein Variationspunkt definiert eine Auswahlmöglichkeit zwischen verschiedenen von der Produktfamilie zur Verfügung gestellten (gleichartigen) Funktionalitäten und/oder Qualitäten (bzw. Verfügbarkeit einer Funktion) oder eine optionale Funktionalität bzw. Qualität. Variationspunkte können in allen Entwicklungsphasen definiert und verwaltet werden [Bo01].

Da eine Produktfamilie typischerweise über sehr viele Variationspunkte verfügt, ist die Variabilität einer Produktfamilie oft sehr umfangreich und komplex. Je umfangreicher die Variabilität einer Produktfamilie, umso größer ist die Flexibilität bei der Entwicklung von kundenspezifischen Produkten; d.h. umso mehr unterschiedliche Produkte können durch die Ausnutzung der Variationspunkte entwickelt werden.

Die Betrachtung der Variabilität im Rahmen einer Produktentwicklung auf der Basis einer Produktfamilie führt zu zwei unterschiedlichen Fragestellungen:

**Technische Variabilität:** Unter technischer Variabilität fassen wir alle Aspekte zusammen, die sich mit der Realisierung einer Variabilität, d.h. der technischen Umsetzung der Variabilität beschäftigen. Die technische Variabilität definiert daher WIE und in welchem technischen Umfeld die Variabilität einer Produktfamilie realisiert ist.

**Fachliche Variabilität:** Die fachliche Variabilität einer Produktfamilie umfasst alle Aspekte der Produktfamilien-Variabilität, die sich auf den Einsatz des Produktes beim Kunden beziehen, wie beispielsweise Aspekte der Funktionalität, Systemumgebung oder betriebliche Einbettung. Die fachliche Variabilität definiert daher das WAS, d.h. die Arten der Variabilität, die für den Kunden relevant und sichtbar sind und somit die Variabilität bezüglich des Einsatzes des Produktes sowie dessen „Umgebung“.

Wie in Abbildung 2 dargestellt, steht die technische Variabilität somit dem Informationssystem nahe, wohingegen die fachliche Variabilität die für den Kunden relevante Variabilität charakterisiert. Zwischen der technischen und fachlichen Variabilität bestehen im wesentlichen zwei Beziehungen. Zum einen wird fachliche Variabilität durch

technische Variabilität realisiert. Zum anderen legt die technische Variabilität den Zeitpunkt der „Konkretisierung“ der Auswahlmöglichkeit fest, d.h. ob die gewählte Variante bspw. zur Laufzeit oder bereits während der Übersetzung eingebunden wird (Details siehe [Bo01]).

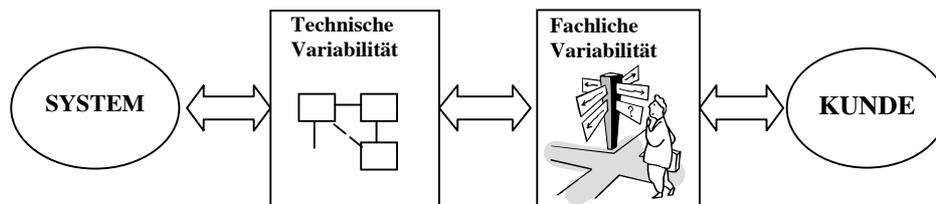


Abbildung 2: Fachliche und technische Produktfamilien-Variabilität

## 2.1 Technische Variabilität

Im Rahmen der technischen Variabilität kann zwischen den Bereichen „IT-Infrastruktur“, „Konkretisierung“ und „Realisierung“ unterschieden werden (siehe Abbildung 3):

**Variabilität im Bereich der IT-Infrastruktur:** Die Einführung eines Produktes erfolgt unter unterschiedlichsten Rahmenbedingungen. Variationspunkte dieser Kategorie definieren die Möglichkeit der Produktfamilie, auf die unterschiedlichen Infrastruktur-Bedingungen der Kunden in geeigneter Weise einzugehen. Beispielsweise können dies Variationen sein, die im Rahmen der Installation konkretisiert werden und etwa die technische Verbindung zum Internet festlegen (56K-Modem oder ISDN-Zugang). In diese Kategorie fallen auch Variationsmöglichkeiten im Hinblick auf die Interoperabilität mit anderen Software- und Hardwaresystemen. Weitere Beispiele dieser Klasse von Variabilität sind: unterstützte Netzwerkumgebung, Lizenzierungsverfahren (gekaufte Software oder gemietete Software), Sicherheitsaspekte oder Variationsmöglichkeiten im Ressourcenbedarf (manche Softwarehersteller unterscheiden ihre Einführungsversionen von den Profi-Versionen dadurch, dass sie z.B. nur eine Obergrenze an Projekten zulassen, die bearbeitet werden können).

**Variabilität und ihr Zeitpunkt der Konkretisierung:** Die Variationspunkte einer Software-Produktfamilie müssen während der Produktentwicklung zu einem Zeitpunkt konkretisiert werden, d.h. es muss zu irgendeinem Zeitpunkt eine Auswahl der letztendlich einzusetzenden Variante getroffen und diese entsprechend festgelegt werden. Das bedeutet, dass jedem Variationspunkt zu einer bestimmten Zeit (genannt „binding time“) eine Variante aus der Menge möglicher Varianten zugewiesen werden muss [Bo01]. Diese Konkretisierung kann beispielsweise schon während der Produktdefinition oder erst später, beispielsweise während der Installation des Produktes, erfolgen. Aus der Sicht der Produktfamilie steigt die Flexibilität mit der Anzahl der Variationspunkte, die zu einem beliebigen Zeitpunkt festgelegt werden können, und somit nicht durch die Art der technischen Realisierung (s.u.) reglementiert sind. Die sich daraus ableitenden Herausforderungen aus technischer Sicht werden in Bosch et al. ausführlich diskutiert [Bo01].

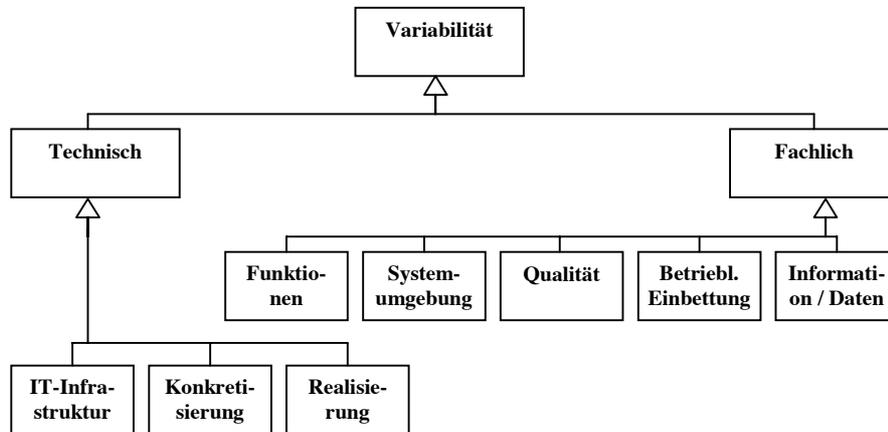


Abbildung 3: Arten der Produktfamilien-Variabilität

**Realisierung:** Viele der bisherigen Betrachtungen von Variabilität fokussieren die technischen Möglichkeiten im Sinne der Modellierung bzw. der Implementierung von Variabilität [AG00], [Bo01], [JGJ97], [Ka96]. Ivar Jacobson beispielsweise beschreibt Mechanismen wie die Nutzung von Parametern, Template Instanzierungen oder die Technik der Spezialisierung, um Variabilität zu modellieren [CN01], [JGJ97]. Aus einer ähnlichen Richtung betrachtet Anastasopoulos Variabilität, indem er sie in Variabilitätstypen wie beispielsweise „Positiv“ und „Negativ“ unterteilt. Mögliche Funktionserweiterungen werden hierbei durch positive Variabilitätstypen definiert. Mögliche Funktionseinschränkungen werden durch negative Variabilitätstypen definiert [AG00].

## 2.2 Fachliche Variabilität

Die fachliche Variabilität einer Produktfamilie unterteilen wir in die Bereiche der „Funktionen“, „Systemumgebung“, „Qualität“, „Betriebliche Einbettung“ und „Informationen/Daten“ (siehe Abbildung 3).

**Variabilität der Funktionen:** Diese Kategorie umfasst Variationspunkte, die Alternativen in der Systemnutzung aufzeigen. Beispielsweise können Funktionalitäten ein- oder ausgeblendet werden. Zusätzlich fallen Variationsmöglichkeiten, welche die Reihenfolge innerhalb eines im System abgebildeten Prozesses verändern, in diese Kategorie. Des Weiteren zählen Variationspunkte in Bezug auf Bedingungen für die Ausführung von Funktionen zu dieser Kategorie.

**Variabilität in der Systemumgebung:** In diese Kategorie fallen Variationsmöglichkeiten, welche die unterschiedlichen Nutzertypen, die mit dem System agieren, und die Verbindungen des Nutzers zum System betreffen. So kann beispielsweise die Anzahl der Nutzer oder aber auch die Art der Verbindung zum System variieren.

**Variabilität in der Qualität:** Hierunter fassen wir Variationsmöglichkeiten aus dem Bereich der nicht funktionalen Anforderungen. Qualitäten wie Ausfallsicherheit, Performance oder Verfügbarkeit können in Abhängigkeit der Domäne, in der das Produkt

eingesetzt werden soll, unterschiedlich sein. Beispielsweise ist die Verfügbarkeit für ein in Europa genutztes Produkt auf eine Kernzeit von z.B. 7:00 Uhr bis 20:00 Uhr begrenzt. Wird das Produkt allerdings weltweit genutzt (mit zentralem Zugriff etwa der Daten), müsste dieselbe Verfügbarkeit auf 24 Stunden ausgedehnt werden.

**Variabilität in der betrieblichen Einbettung:** Variationen in der betrieblichen Einbettung betreffen die Ablauf- und Aufbauorganisation. Hier handelt es sich im Gegensatz zu der geänderten Abfolge von einzelnen Funktionen, die in die erste Kategorie fallen, um Varianten der Geschäftsprozessabfolge.

**Variabilität im Bereich der Informationen/Daten:** Dieser Bereich beinhaltet Variationspunkte, welche die Inhalte des Systems variieren können. Damit sind zum Beispiel Variationsmöglichkeiten in Bezug auf die Detaillierung der Informationen gemeint. Ein weiteres Beispiel umfasst Varianten bezüglich der Datenaktualität.

### 3 Fachliche Variabilität

In Kapitel 2 haben wir zwischen fachlicher und technischer Produktfamilien-Variabilität unterschieden und die fachliche Variabilität in die Bereiche „Funktionen“, „Systemumgebung“, „Qualität“, „Betriebliche Einbettung“ und „Informationen/Daten“ unterteilt.

In diesem Kapitel detaillieren wir die o.g. Bereiche der fachlichen Variabilität (siehe Abbildung 4). Wir konzentrieren uns hierbei auf die wesentlichen Aspekte (Details siehe [HP01]).

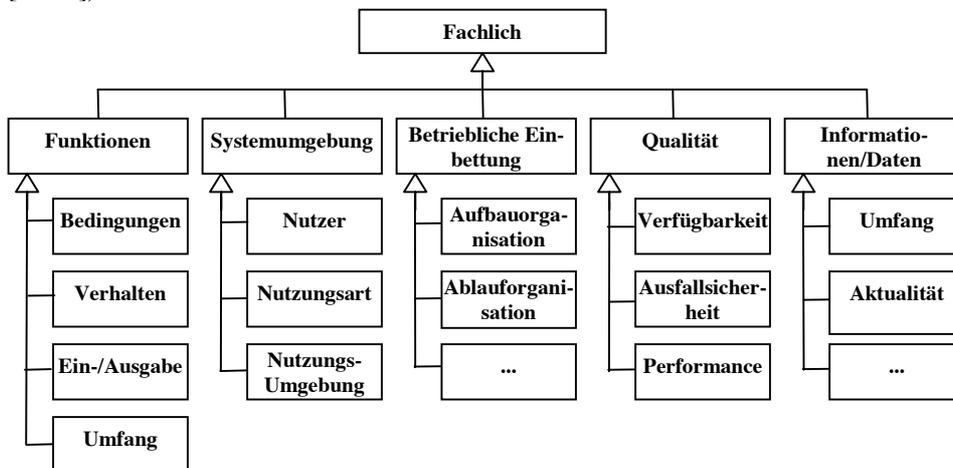


Abbildung 4: Modell der fachlichen Variabilität

**Variabilität bezüglich der Funktionen:** Die Kategorie „Funktionen“ umfasst Variationsmöglichkeiten, welche die Bedingungen bezüglich der Ausführung, das Verhalten des betrachteten Systems, die Ein- und Ausgabe und den Funktionsumfang betreffen. Die Kategorie „Bedingungen“ charakterisiert Bedingungen, die entweder zum Aufruf einer Funktion erfüllt werden (Pre-Bedingung) oder aber auch nach der Durchführung

der Funktion (Post-Bedingung) gegeben sein müssen. Beispiel: In dem einen Produkt ist es notwendig, dass zur Abspeicherung von Auftragsdaten eine Auftragsnummer aus dem Finanzwesen gegeben sein muss, in dem anderen Produkt gibt es eine solche Beschränkung nicht.

Des Weiteren lässt sich Variabilität aus dem Bereich „Verhalten“ noch in die Kategorie „Abhängigkeit“ einteilen, d.h. die Abhängigkeit der einzelnen Arbeitsabläufe wird variiert. Workflow gestützte Systeme werden gerade in diesem Bereich die Abhängigkeit der Funktionsabläufe auf den jeweiligen Kunden abstimmen.

Variationen im Bereich der Ein- und Ausgabe ergeben sich z.B. durch Unterschiede in den Wertebereichen oder in den Fehlermeldungen. Abhängig vom Anwendungsgebiet des Produktes können sehr detaillierte technische Fehlermeldungen oder einfache, für den durchschnittlichen Anwender verständliche Hinweise gefordert sein. Eine weitere Variabilitätsmöglichkeit im Rahmen einer Produktfamilie liegt darin, das Verhalten des Systems unterschiedlich ausprägen zu können. Es kann sinnvoll sein, die Funktionsabfolge A, B, C in die Reihenfolge A, C, B zu ändern. Dabei lässt sich noch unterscheiden, ob die Ablaufänderungen eine Änderung des Ergebnisses hervorrufen oder nicht.

Die Kategorie „Umfang“ beinhaltet Variationen, die aus Hinzufügen neuer Funktionen oder durch Erweitern bestehender Funktionen entstehen. Des Weiteren zählen Variationen, die sich aus der Vereinfachung einer Funktion oder dem Löschen einer gesamten Funktion ergeben, zu dieser Kategorie. Ein Beispiel für Unterschiede im Funktionsumfang ist die geläufige Unterscheidung zwischen Einführungsversion und Profi-Version eines Softwareproduktes, wobei sich die Einführungsversion durch die Reduktion des Funktionsumfangs von der Profiversion unterscheidet, beide aber derselben Produktfamilie entstammen.

**Variabilität in der Systemumgebung:** Die Kategorie „Systemumgebung“ umfasst Variationsmöglichkeiten, welche den „Nutzer“, die „Nutzungsart“ und die „Nutzungsumgebung“ des betrachteten Systems beeinflussen. Variationsmöglichkeiten in Bezug auf den Nutzer sind zum Beispiel die Anzahl der Anwender, die gleichzeitig mit dem System arbeiten können, und welcher Typ von Nutzer mit dem System arbeitet. Für letztere Variationsmöglichkeit sei der Business-to-Business (B2B)-Bereich als Beispiel genannt, in dem zum einen Systeme direkt miteinander kommunizieren, aber gleiche Funktionalitäten wie beispielsweise eine Auftragsvergabe auch von einem Anwender durchgeführt werden muss.

Unter die Kategorie Nutzungsart fallen Variationsmöglichkeiten für die Art und Weise, wie das System genutzt wird oder werden darf. So wird hier zwischen On- und Offline-Betrieb unterschieden. Auch in der Synchronisation des betrachteten Systems mit der „Außenwelt“ ergeben sich Variationsmöglichkeiten wie z.B. ob eine Synchronisation von Daten manuell oder automatisch erfolgt.

Die Nutzung eines Systems kann aus unterschiedlichen Umgebungen erfolgen; so zum Beispiel „fest installiert“ als Stand-alone-Lösung oder als flexible Lösung, die im Intra- oder/und Internet läuft und deshalb erst bei Bedarf „installiert“ wird.

**Variabilität in der Qualität:** Variationsmöglichkeiten im Rahmen der Qualität betreffen nichtfunktionale Anforderungen. Bereiche wie „Verfügbarkeit“, „Ausfallsicherheit“ und „Performance“ fallen in diese Kategorie. Unter Variationen im Bereich „Verfügbarkeit“ verstehen wir die Möglichkeit, die Produkte unter verschiedenen Zugriffsbedingungen

zur Verfügung stellen zu können. Beispielsweise kann ein Produkt nur von einem Anwender gleichzeitig genutzt werden, während ein anderes Produkt der Familie multi-user-fähig ist.

Varianten im Bereich der Ausfallsicherheit können in den Sicherungsmechanismen liegen. Im einfachen Falle werden etwa Daten „nur“ einmal in der Nacht gesichert, während in besonders kritischen Systemen möglicherweise aktuell die Daten immer gespiegelt werden. Die fachliche Anwendung kann dabei davon völlig unberührt sein, d.h. der Nutzer wird keinen Unterschied in der Anwendung erkennen.

Performance-Variationen beschreiben unterschiedliche Anforderungen wie zum Beispiel hinsichtlich des gleichzeitigen Zugriffs von Anwendern oder der Menge der zu erwartenden Daten.

**Variabilität in der betrieblichen Einbettung:** Im Rahmen der betrieblichen Einbettung eines Systems unterscheiden wir zwischen Aufbau- und Ablauforganisations-Aspekten. Als Aufbauorganisation verstehen wir die Struktur des Unternehmens, in dem das System eingesetzt werden soll. Da diese Strukturen auch in einer Domäne sehr unterschiedlich sein können, müssen insbesondere Produktfamilien, deren Funktionalität auf diese Strukturen eingehen, variabel sein.

Die Ablauforganisation beschreibt hingegen welche Geschäftsprozesse in welcher Reihenfolge und Abhängigkeit im Unternehmen abgebildet werden. Diese variieren zwischen den unterschiedlichen Unternehmen und haben somit auch Einfluss auf die kundenspezifischen Produkte.

**Variabilität bezüglich der verarbeiteten Informationen/Daten:** Variationsmöglichkeiten im Rahmen von Informationen und Daten umfassen etwa zum einen ihren „Umfang“ und zum anderen ihre „Aktualität“.

Variationsmöglichkeiten im Bereich des Datenumfangs können aus erweiterten oder reduzierten Informationen resultieren. Möglicherweise wird in einem Produkt einer Produktfamilie, welches die Recherche nach Gebrauchtwagen ermöglicht, nur die Marke, das Baujahr und der Preis dargestellt, während in einem anderen Produkt weitergehende Informationen wie Leistung, Anzahl der Türen, Farbe etc. benötigt werden.

Die Anforderung, wie aktuell Daten in einem System sein müssen, kann zwischen den unterschiedlichen Nutzern der Produkte variieren. Beispielsweise werden in einem Produkt einer Produktfamilie stündlich aktuelle Wetterdaten gefordert, während es in einem anderen Produkt ausreicht, die Wetterdaten einmal täglich zu aktualisieren.

#### **4 Repräsentation von fachlicher Variabilität mit Use Cases**

Die Anwendung von Use Cases zur besseren Integration des Kunden in den Produktentwicklungsprozess und zur Verbesserung von Anforderungsspezifikationen ist in Praxis und Forschung vielfach erprobt [Ja92]. Use Cases setzen die Anforderungen in einen fachlichen Kontext, reduzieren dadurch die Komplexität einer oft sehr großen Liste von Anforderungen und machen sie leichter verständlich [Ca95].

Motiviert durch den erfolgreichen Einsatz von Use Cases in der Einzel-Produkt-Entwicklung verfolgen wir den Ansatz, Use Cases zur Präsentation der Möglichkeiten von Produktfamilien sowie deren Variationsmöglichkeiten (Variationspunkte und Vari-

anten) für den Kunden einzusetzen. Dabei stellt sich die Frage, wie Use Cases mit der Variabilität der Produktfamilie in Beziehung gesetzt werden kann. Hierzu betrachten wir im folgenden Use Case Konstrukte und ihre Eignung zur Abbildung der Variabilität.

Tabelle 1 gibt einen Überblick über die Zuordnung von Use Case Konstrukten und den in Kapitel 3 diskutierten Variabilitätsarten (Details siehe [HP01]).

Use Case Attribute	Fachliche Variabilität													
	Funktionen				System-Umgebung			Qualität			Betrieb. Einb.		Daten	
	edingungen	erhalten	in-/Ausgabe	mfang	utzer	utzungsart	utzungs Umgebung	erfügbarkeit	usfallsicherheit	erformance	ufbauorganisation	blauforganisation	mfang	ktualität
Use Case Level														
Geschäftsziel														
Use Case Diagramm					○	○	○							
Aktivitäts-Diagramm (Haupt-Szenario)	●			○										
Haupt-Akteur					●	○	○				○			
Andere Akteure					●	○	○				○			
Pre-Bedingung	●													
Post-Bedingung	●													
Haupt-Szenario		●		●										
Alternativ-Szenarien		●		●										
Fehler-Szenario		●	●	●										
Unterstützende Funktionalität				●										
Erweiterungen				●										

- Variabilität im entsprechenden Use Case Konstrukt abbildbar
- Variabilität im entsprechenden Use Case Konstrukt teilweise abbildbar

Tabelle 1: Beziehung zwischen Use Case Konstrukt und Variabilität

Die Zeilen in Tabelle 1 stellen die Konstrukte eines Use Case Templates dar. Das Template basiert auf den Use-Case Templates von Cockburn [Co01] und Kulak et al. [KG00]. Die Spalten der Tabelle repräsentieren die in Kapitel 3 beschriebenen fachlichen Variabilitätsarten. Dabei bedeutet ein ausgefüllter Punkt in einer Zelle der Tabelle, dass die entsprechende Variabilität durch das jeweilige Use Case Konstrukt ausgedrückt werden kann. Ein nicht ausgefüllter Punkt bedeutet, dass die entsprechende Variabilität nur teilweise ausgedrückt werden kann und eine leere Zelle bedeutet, dass die entsprechende Variabilität zu dem entsprechenden Use-Case Konstrukt keinen Bezug hat.

Die Darstellung der wesentlichen Variabilitätsarten und ihre Beziehung zu den Use Case Konstrukten zeigt, dass sich einige der Variabilitätsarten gut, einige nur teilweise und einige nicht mit Use Cases und Szenarien ausdrücken lassen.

Am Besten lassen sich, wie erwartet, die Aspekte bzgl. der Variabilität in der Funktionalität der Produktfamilie mit Use Cases darstellen. Beispielsweise können Abweichungen im Funktionsablauf durch unterschiedliche „alternative Szenarien“ dargestellt werden oder verschiedene mögliche Verhalten in Aktivitätsdiagrammen beschrieben werden.

Variabilitätsaspekte bezüglich der Systemumgebung lassen sich, wie in der Tabelle 1 ersichtlich, nur sehr bedingt auf Use Cases abbilden. In Use Case-Diagrammen lassen sich zwar unterschiedliche Akteure (Systembenutzer und andere Systeme), die mit dem betrachteten System interagieren, definieren; eine Abbildung von Variationen im Bereich der Nutzungsumgebung ist dagegen beispielsweise nur sehr schwer möglich.

Variabilitätsaspekte bezüglich der Qualität lassen sich dagegen nicht mit den typischen Use Case Konstrukten ausdrücken. Hinweise über die Ausfallsicherheit oder die Verfügbarkeit eines Softwareproduktes lassen sich in den hier aufgezeigten Use Case Konstrukten nicht abbilden. Ähnlich sieht es bezüglich der Variabilitätsaspekte der Informationen/Daten aus. Auch diese Aspekte können leider nicht mit typischen Use Case Konstrukten dargestellt und somit zum Kunden kommuniziert werden.

Hieraus ergibt sich die Notwendigkeit, neben Use Cases andere Methodiken zur Kommunikation von Produktfamilien-Variabilitäten einzusetzen und diese geeignet mit Use Cases zu „integrieren“. Zudem zeigten unsere ersten Erfahrungen mit der Darstellung von funktionalen Aspekten der Variabilität in Use Cases die Notwendigkeit, Use Cases – in möglichst eingeschränktem Masse – zu erweitern. Diese Erweiterungen beziehen sich insbesondere auf eine geeignete Visualisierung (Präsentation) der vorhandenen Variationspunkte sowie der damit assoziierten Varianten und dienen somit der leichteren Kommunizierbarkeit der von der Produktfamilie zur Verfügung gestellten Variationsmöglichkeiten zum Kunden.

## **5 Zusammenfassung**

Variabilität ist das zentrale Konzept einer Produktfamilie. Kundenspezifische Produkte können auf der Basis des Produktfamilienkerns unter Ausnutzung der Variabilität entwickelt werden. Der Erfolg einer Produktfamilie aus der Sicht des Providers hängt neben der erreichbaren Kundenakzeptanz und damit erzielbaren Erlösen davon ab, möglichst viele der kundenspezifischen Anforderungen durch die konkrete Auswahl der zur Verfügung stehenden Varianten zu erfüllen und damit die Kosten der Entwicklung kundenspezifischer Produkte zu reduzieren.

Die Berücksichtigung von Variabilität als Bestandteil der Produktdefinition von Individualprodukten stellt eine große Herausforderung dar. In diesem Beitrag konzentrierten wir uns auf die Herausforderung der Vermittlung von Software-Produktfamilien-Funktionalität und -Variabilität zum Kunden. Während sich Szenarien mehrfach als geeignetes Mittel zur Kommunikation der Funktionalität eines Softwareproduktes bewährt haben, existieren zur Kommunikation der Variabilität einer Produktfamilie bisher keine Ansätze. Wir haben die Variabilität einer Software-Produktfamilie in fachliche

und technische Variabilität differenziert und die fachliche Variabilität, die bei der Kommunikation zum Kunden eine wesentliche Rolle spielt, auf Use Cases abgebildet.

Die Untersuchung bezüglich der Abbildung der fachlichen Variabilitätsarten auf Use Case Konstrukte ergab, dass sich einige Variabilitätsarten (bspw. Funktionalität) sehr gut auf Use Case Konstrukte abbilden lassen. Andere Variabilitätsarten wie bspw. „Qualität“ können hingegen nicht auf traditionelle Use Case Konstrukte abgebildet werden.

Gegenstand unserer aktuellen Arbeiten ist die Visualisierung der Variabilität von Software-Produktfamilien, die durch Use Cases ausgedrückt werden kann. Die Kommunikation der nicht mit Use Cases darstellbaren Variabilität zum Kunden ist Gegenstand unserer zukünftigen Forschungsaktivitäten.

## Referenzen

- [AG00] M. Anastopoulos; C. Gaceck: Implementing Product Line Variabilities; IESE-Report Nr. 089.00/E, Version 1.0; Fraunhofer Institut Experimentelles Software Engineering, 2000.
- [Bo01] Jan Bosch, Gert Florijn, Danny Greefhorst, Juha Kuusela, Henk Obbink, Klaus Pohl: Variability Issues in Software Product Lines; Fourth International Workshop on Product Family Engineering (PFE-4), Bilbao, Spain, 2001
- [Ca95] J. Carroll: The Scenario Perspective on System Development, Scenario-Based Design: Envisioning Work and Technology in System Development; John Wiley & Sons, 1995
- [CN01] Paul Clements, Linda Northrop: Software Product Lines, Practices and Patterns; SEI Series in Software Engineering; Addison Wesley, 2001
- [Co01] Alistair Cockburn: Writing Effective Use Cases; Addison Wesley, 2001
- [Es99] ESAPS- Engineering Software Architectures, Processes and Platforms for System Families, ITEA Full Project Proposal, June 1999
- [HP01] Günter Halmans, Klaus Pohl: Abbildung von Produktfamilien-Variabilität in Use Cases; Technischer Report; Universität Essen; 2001
- [Ja92] Ivar Jacobson: Object-Oriented Software Engineering: A Use Case Driven Approach; Addison Wesley, 1992
- [JGJ97] Ivar Jacobson, Martin Griss, Patrik Jonsson: Software Reuse, Architecture, Process and Organization for Business Success; Addison Wesley, 1997
- [Ka96] Even André Karlsson: Software Reuse, A Holistic Approach, Wiley Series in Software Based Systems, 1996
- [KG00] Daryl Kulak, Eamonn Guiney: Use Cases, Requirements in Context; Addison Wesley, 2000
- [PBG01] Klaus Pohl; Mathias Brandenburg, Alexander Gülich,; Scenario-based Change Integration in Product Family Development; 2nd ICSE Workshop on Software Product Lines: Economics, Architectures, and Implications, May 2001
- [PH97] Klaus Pohl, Peter Haumer: Modelling Contextual Information about Scenarios; in: Proc. of REFSQ 97, Workshop, 1997
- [We98] Klaus Weidenhaupt, Klaus Pohl, Matthias Jarke, Peter Haumer: Scenario Usage in System Development: A Report on Current Practice; IEEE Software, 15(2): 34-45, March/April 1998
- [WL99] David M. Weiss, Chi Tau Robert Lai: Software Product-Line Engineering, A Family-Based Software Development Process; Addison Wesley, 1999