

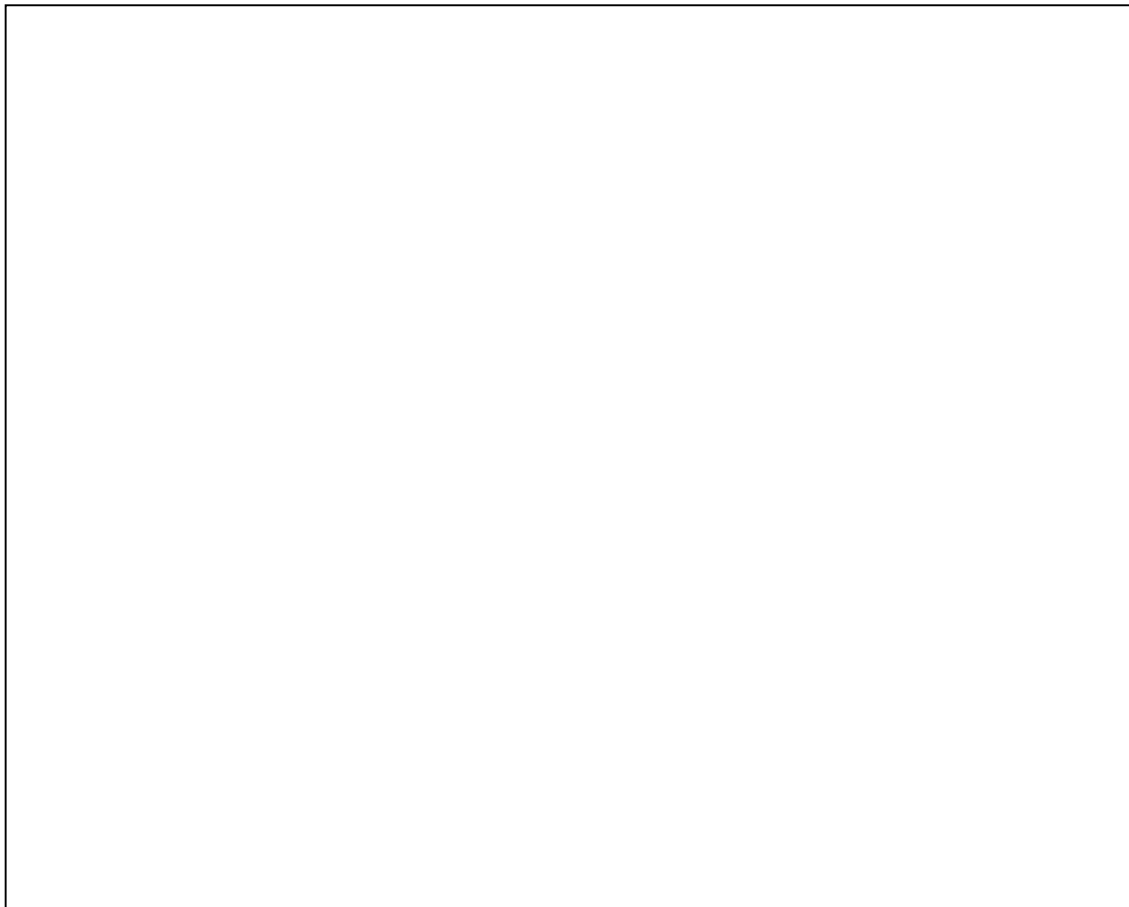


# Produktlinienentwicklung

Scoping, Domain Analysis  
&  
Application Analysis

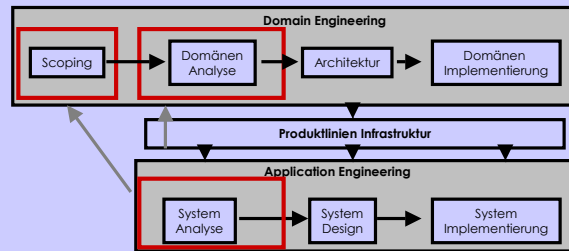
Urte Rietdorf

Cüneyt Egin



## Inhaltsverzeichnis

- Produktlinienentwicklung
- Scoping
- Anforderungsanalyse
- Domänen Analyse
- Applikationsanalyse
- Konkretes Beispiel
- Fazit



Folie 2

Urte Rietdorf, Cüneyt Egin

Im folgenden Vortrag möchten wir auf folgende Punkte eingehen:

Zuerst führen wir kurz die Produktlinienentwicklung im allgemeinen ein, was sie darstellt, worauf es ankommt, und was die Vorteile und Schwierigkeiten sind.

Anschließend geht es um das Thema Scoping, welches zu Beginn des Produktlinienentwickelns durchgeführt wird.

Darauf folgend wird auf die Anforderungsanalyse eingegangen, die sich in der Produktlinienentwicklung in die beiden nächsten Punkte Domänen Analyse und Applikationsanalyse teilt.

Die Domänenanalyse als Anforderungsanalyse des Domänen Engineerings ist dementsprechend auch der nächste Punkt,

Gefolgt von der Applikationsanalyse (auch System Analyse genannt), die im Applikationsengineering stattfindet.

Diese Punkte werden in einem konkreten Beispiel kurz erläutert.

Letztlich werden wir ein kurzes Conclusio über das Thema Produktlinienentwicklung abgeben.



---

# Produktlinienentwicklung

---

Folie 3

Urte Rietdorf, Cüneyt Egin



## Grundidee der Produktlinienentwicklung

- von einer Orientierung auf das einzelne Produkt wegzukommen
  - Softwareentwicklung systematisieren und Methoden bereitstellen
- Fokus auf eine integrierte Entwicklung verschiedener Produkte auf Basis einer gemeinsamen Infrastruktur

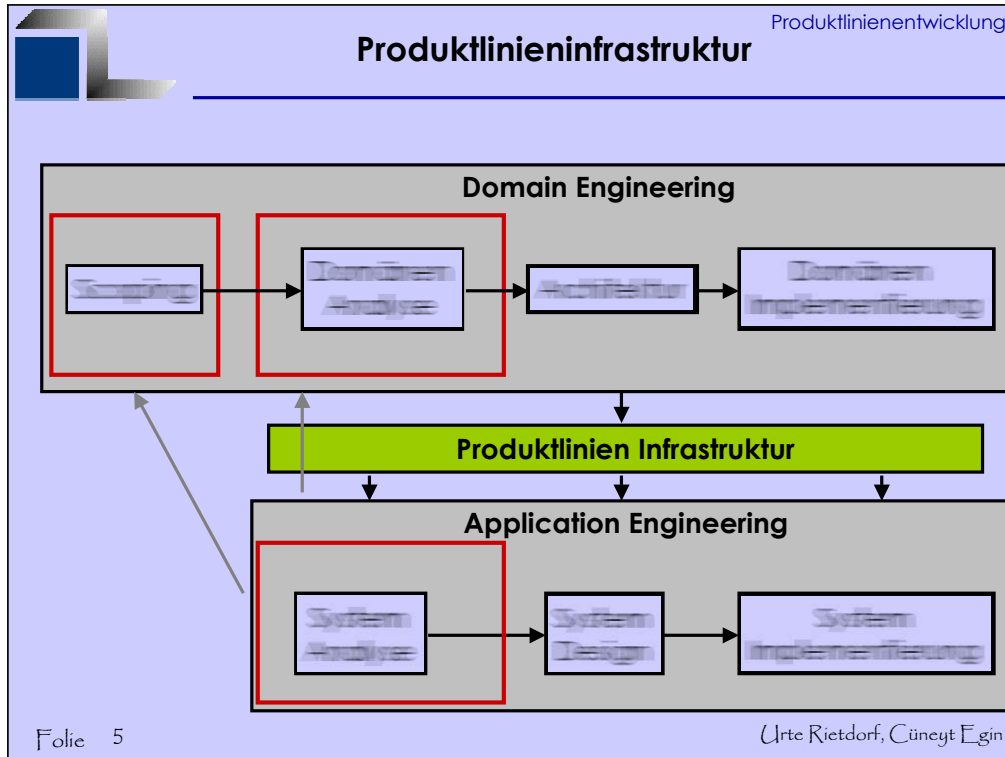
Der Grundgedanke der Produktlinienentwicklung ist es, von einer Entwicklung mit Fokus auf ein einzelnes Produkt wegzukommen, und stattdessen eine integrierte Entwicklung verschiedener Produkte zu fokussieren, die eine gemeinsame Infrastruktur besitzen.

Dabei sollen insbesondere wiederverwendbare Komponenten gefunden und entwickelt werden, um den Prozess der Softwareentwicklung zu systematisieren, und um besser geeignete Methoden für diesen Prozess bereitstellen zu können.

Bei älteren Arten der Wiederverwendung ( z.B. Objektorientierung, Architekturstile, Designmuster und Komponentenentwicklung) wurde eben diese Wiederverwendung meist nur zufällig erreicht, und war nicht geplant. Die Erstellung wiederverwendbarer Komponenten geschah eigentlich nur dann, wenn der Entwickler die Möglichkeiten der Wiederverwendung wahrnehmen und ausnutzen wollte.

Im Gegensatz dazu ist die Wiederverwendung einzelner Komponenten bei Produktlinien strategisch geplant (strategic reuse), und zielt insbesondere auf eine Leistungssteigerung ab, da mit einer Komponente so mehrere Anwendungen realisiert werden können.

Dies hat zur Folge, dass der Entwicklungsprozess die Wiederverwendung explizit vorsieht, und alle Beteiligten an der Entwicklung forciert, sich über Wiederverwendung Gedanken zu machen und die Ergebnisse in die Entwicklung entsprechend zu integrieren.



Die Produktlinieninfrastruktur, auch Plattform der Produktlinie genannt, beinhaltet als wesentlichen Punkt die gemeinsamen und variablen Artefakte, auf die später noch häufiger eingegangen wird.

Die Infrastruktur wird im Domain Engineering entwickelt, und im Application Engineering genutzt, nämlich indem die Variationspunkte der Artefakte eingebunden, und anschließend um produktspezifische Anteile ergänzt werden.

Der Aufbau einer gemeinsamen Infrastruktur (bzw. Plattform) ist jedoch sehr Zeit- und Geldintensiv, was als Hauptkritikpunkte zur Erstellung einer Produktlinie angesehen wird.


Produktlinienentwicklung

## Ursprung

---

- Aus der Hardware-Industrie entlehnt
  - Mit den Zielen
    - Produktionszeit zu verkürzen (Time-to-Market)
    - Qualität der einzelnen Produkte zu steigern
    - Organisationsstrukturen für den Entwicklungsprozess effektiv zu gestalten (niedrige Kosten)
- Durch die Erkenntnis
  - Organisationen von Software-Systemen sind spezialisiert auf ihre
    - Aufgaben der Produkte
    - Verwendeten Methoden

---

Folie 6
Urte Rietdorf, Cüneyt Egin

Der Ursprung der Produktlinienentwicklung stammt aus der Produktindustrie (also z.B. dem Automobilbau), und lehnt sich auch stark an deren Ziele und Methoden an.

Die Ziele der Produktlinien in der Industrie waren es,

- Die Produktionszeit zu verkürzen

Indem ab ungefähr der 5. Wiederverwendung (in der Softwareproduktikon) die Dauer zur Erstellung einzelner Produkte stark sinkt, da viele Komponenten dank der Wiederverwendung bereits vorhanden sind.

- Qualität der einzelnen Produkte zu steigern

Da die einzelnen Komponenten mehrfach eingesetzt werden, und so meist bereits sehr gut getestet und funktionstüchtig sind, und einzelne Fehlerquellen ausgeschlossen werden können.

- Organisationsstrukturen für den Entwicklungsprozess effektiv zu gestalten

Durch den größeren Rahmen einer Produktlinie im Vergleich zu einzelnen Produkten können Vorgänge durch die Produktlinieninfrastruktur effizienter organisiert werden, was zu einer Reduzierung der Kosten führt.

Es wird auf Bereiche mit längeren Erfahrungszeiträumen zurückgegriffen wie z.B. der klassischen Produktindustrie, um die gewonnenen Erkenntnisse auf die Software-Entwicklung zu übertragen.

Die Bestrebungen der Software-Entwicklung und der Quellbereiche (Wo die Erfahrungen gewonnen wurden/ herkommen) sind hierbei weitgehend identisch:

- Ziel: Durch Systematisierung des Herstellungsprozesses Methoden bereitzustellen.

Durch die Erkenntnis, dass die einzelnen Organisationsstrukturen von Software-Herstellern sich auf ihre Aufgaben der Produkte und auf ihre verwendeten Methoden spezialisieren, ist man bestrebt, die Wiederverwendung der Funktionalitäten auf einzelne Komponenten zu verlagern.

## 3 PL Hypothesen

- Redevelopment Hypothesis
  - Entwicklung ist meist Wiederentwicklung (Variationen)
- Oracel Hypothesis
  - Meist vorhersagbar
- Organizational Hypothesis
  - Software und Organisation kann so gestaltet werden, dass Auffangen von Veränderungen ohne große Umstrukturierung möglich ist

Produktlinien beruhen auf den Annahmen:

### **1. Redevelopment Hypothesis:**

- SW-Entwicklung ist meistens eine Wiederentwicklung von SW.
- Es werden meistens nur Variationen von bestehenden SW-Systemen implementiert

### **2. Oracel Hypothesis:**

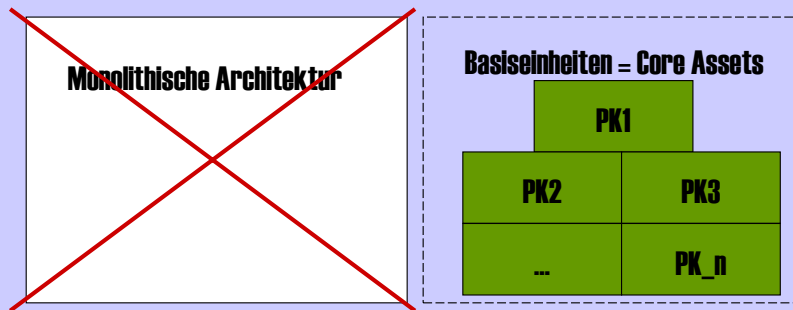
- Es ist vorhersehbar welche Variationen innerhalb des Lebenszyklus eines SW-Systems auftauchen werden

### **3. Organizational Hypothesis:**

- Es ist möglich nicht nur die SW sondern auch die Organisation so zu gestalten, dass das Auffangen von Veränderungen ohne größere Umstrukturierung möglich ist

## Fazit dieser Erkenntnis

- Verlagerung und Aufteilung der Funktionalität auf Produktartefakte



- Artefakte oder auch Assets bestehen aus allen - in einer Produktlinieninfrastruktur enthaltenen - wiederverwendbaren Komponenten, allen Dokumenten und ähnlichem.
- Diese so genannten Basiseinheiten (Core Assets) bilden die ehemals ineinander verwebten Funktionen in -von einander abgekapselten- Bausteinen ab.
- Ziel des Produktlinien Engineerings ist es, ein Design zu entwerfen, das es erlaubt, die geplanten Produkte zu entwickeln und Veränderungen an den Produkten in das Design aufnehmen zu können.
  1. Nämlich sowohl momentan geplante Produkte
  2. Als auch zukünftige Produkte
- Diese Veränderungen sind geplanter und (oftmals in der Realität) ungeplanter Natur, was ein durch die Produktlinie zu realisierendes hohes Maß an FLEXIBILITÄT voraussetzt.

**Also:**

Es werden nicht mehr einzelne Software-  
produkte *reaktiv* nach Markt- oder Kunden-  
bedarf entwickelt,  
sondern der Fokus liegt auf der *proaktiven*  
Gestaltung einer gemeinsamen Plattform für  
eine Vielzahl von jetzigen und zukünftigen  
Produkten für eine bestimmte Domäne

[BKPS 2004]

Das heißt, dass nicht mehr wie bisher üblich auf den Bedarf reagiert wird, in dem anschließend das Produkt entwickelt wird,  
sondern nun wird im Vorfeld überlegt wohin sich der Markt entwickeln könnte, was zukünftig geplant wird, und welche Komponenten auch in zukünftigen Produkten Einsatz finden könnten und sollten.

 **Definition Domäne** Produktlinienentwicklung  
**in der Softwareproduktlinienentwicklung**

- Eine (technische oder Anwendungs-) Domäne ist ein zusammenhängender Teilbereich einer Produktlinie, der eine gut wiederverwendbare Funktionalität für die Produkte der Produktlinie enthält.

Während eine Anwendungsdomäne einen zusammenhängenden Funktionsbereich auf der Problemseite repräsentiert, wird eine technische Domäne meist (größenabhängig) durch eine Komponente auf der Lösungsseite dargestellt. [BKPS 2004]

- „Menge von Konzepten und Technologien in einem Wissensbereich“ [Bass,Clemens, Donohoe]

---

Folie 10 Urte Rietdorf, Cüneyt Egin

Es gibt verschiedene Definitionen des Begriffs „Domäne“. Um diesen abstrakten Begriff ein wenig greifbarer zu machen, wurden hier verschiedene Definitionen ausgewählt.

## Weitere Definitionen & ein Beispiel

- Als Domäne werden Anwendungsgebiete bezeichnet, in denen eine Applikation zum Einsatz kommen kann.

[soko2005]

- Beispiel:  
Domänen für ein Bildbearbeitungssystem können sein:
  - Speicherung,
  - Bildmodi oder
  - Filter

Produktlinienentwicklung

## Vorteile einer Produktlinie

- Potential existierender Ressourcen nutzen
- Produktivität steigern
- Produktqualität erhöhen
- Zeitaufwand verringern
- Erkennen von Gemeinsamkeiten versch. Produkte

▼

Implementierung  
**wiederverwendbarer** Komponenten

Folie 12 Urte Rietdorf, Cüneyt Egin

Die Vorteile einer Produktlinie liegen im:

- Erhöhen der Potentiale existierender Ressourcen

Da diese in einer Produktlinie wesentlich besser genutzt werden können, und – Eingesetzt in vielen Produkten – natürlich den ROI (Return on Invest) steigern

- Steigern der Produktivität

Es können wesentlich mehr Produkte in kürzerer Zeit entwickelt werden, wenn Komponenten gemeinsam genutzt werden.

- Verbessern der Produktqualität


Da Komponenten für den mehrfachen Gebrauch wesentlich sorgfältiger entwickelt, und anschließend besser getestet werden, enthalten sie meist deutlich weniger Fehler

- Verkürzung des Zeitaufwands bis zur Fertigstellung des Produktes

Nachdem die Produktlinie angelaufen ist, können einzelne neue Produkte durch die verbesserte Infrastruktur wesentlich schneller erstellt werden.

⇒ Ermöglicht frühzeitiges Erkennen von Gemeinsamkeiten verschiedener Produkte

⇒ Implementierung wiederverwendbare Komponenten

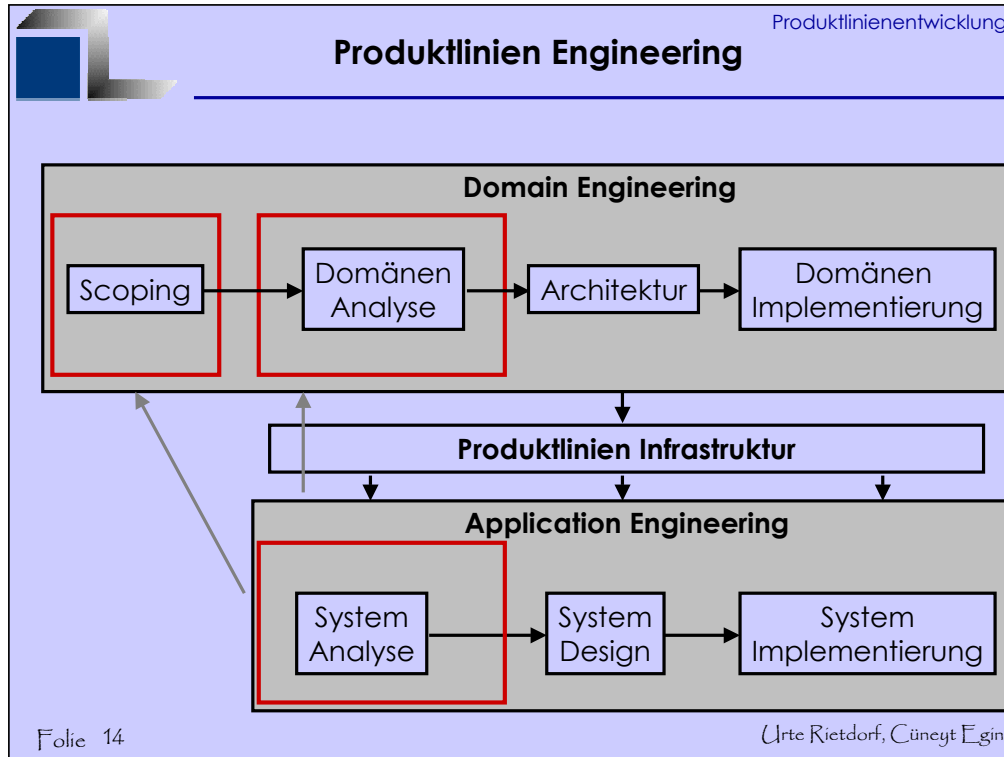


Problem	Lösung
<ul style="list-style-type: none"><li>• Funktionssystemen Komponenten herauszulösen</li><li>• Komponentenbasiert Entwickeln</li></ul> <p>⇒ nicht einfach !</p>	<p>Methodisches und Systematisiertes vorgehen!</p>

Folie 13

Urte Rietdorf, Cüneyt Egin

Das



- Scoping legt die Grenzen für das System und die Domäne fest. Es verbindet die externe Sicht der Produktplanung mit der internen Sicht der Entwicklungseigenschaften
- Produktlinienengineering ist in zwei Phasen unterteilt: Domain Engineering und Applicationengineering
- Domain Engineering besteht aus: Domänen Analyse, Architektur und Domänen-Implementierung
- Aus dem Domain Engineering geht die Produktlinieninfrastruktur bzw. eine Plattform hervor
- Aufbauend auf diesem Ergebnis werden darauf folgend im Applicationengineering die jeweiligen konkreten Produkte der Produktlinie entwickelt
- Applicationengineering besteht aus System Analyse, - Design und -System Implementierung



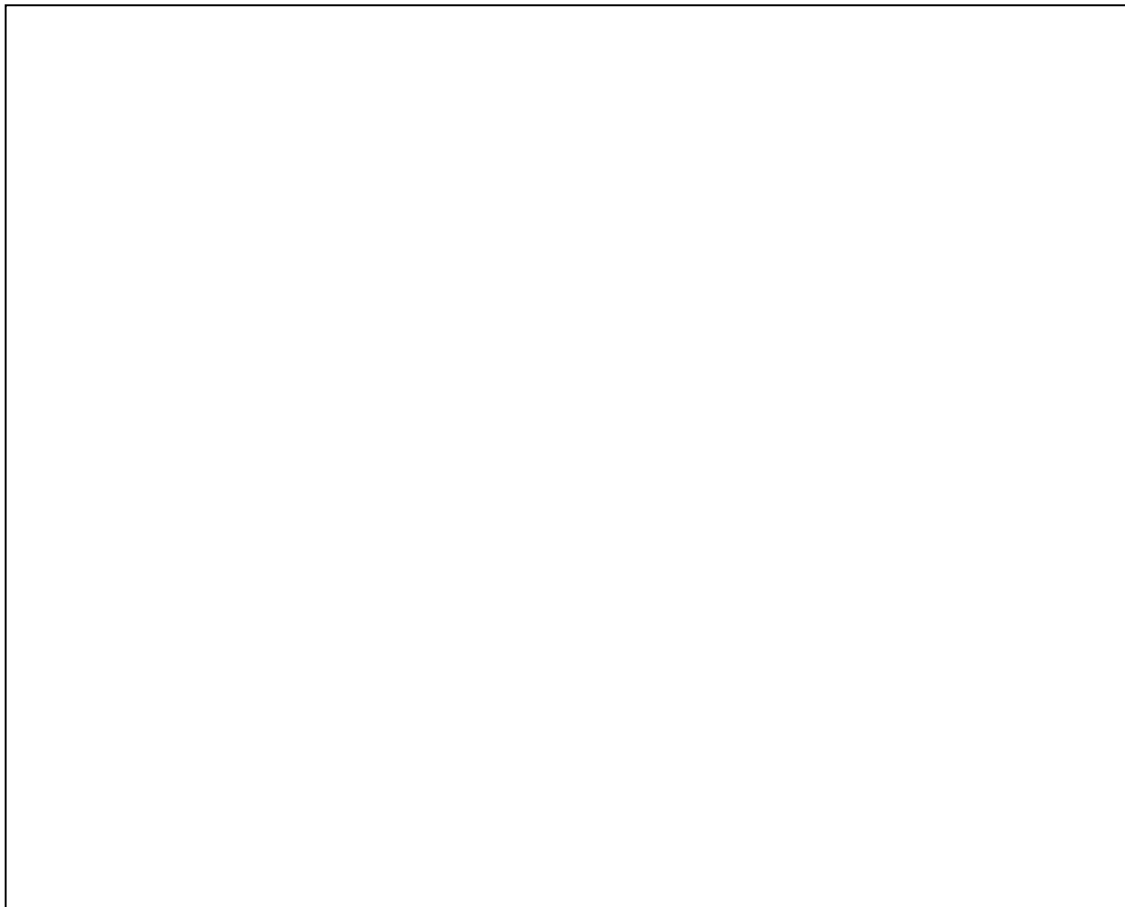
---

# Scoping

---

Folie 15

Urte Rietdorf, Cüneyt Egin



Scoping

## Scoping

- Welche Produkte sollen im Rahmen der Produktlinie erstellt werden
- Welches sind die wiederverwendbaren Komponenten
- Ökonomische Analyse

⇒ Welche Produkte können zu einer Produktlinie zusammengefasst werden

---

Folie 16 Urte Rietdorf, Cüneyt Egin

Scoping hat die Aufgabe, wiederverwendbare Komponenten zu identifizieren und herauszufinden und welche Produkte zu einer Produktlinie zusammengefasst werden können. Zudem ist es sinnvoll, im Rahmen des Scopings die ökonomischen Aspekte der Produktlinie zu analysieren, um die Wirtschaftlichkeit zu eruieren.

**Def. Nach Clements:** Scoping ist eine Aktivität, die ein System oder eine Klasse von Systemen durch Festlegung der Eigenschaften oder Aspekte, die enthalten oder nicht enthalten sein sollen, einschränkt.

**Product Line Scope:** Gesamtheit der zu entwickelnden oder geplanten Produkte und Eigenschaften.



Scoping

## 3 Scopingvarianten

- Product Portfolio Scoping
  - Ziel: Produktportfolio, welches Produkte, die Bestandteil der PL sind beschreibt
- Domain Scoping
  - Ziel: Ableitung & Abgrenzung der Domänen
- Asset Scoping
  - Ziel: Komponentendefinition & Identifikation (legt Grad der wiederverwendbaren Funktionalitäten fest)

Folie 17 Urte Rietdorf, Cüneyt Egin

Im Produktlinienumfeld ist es sinnvoll, statt der eigentlichen Definition des Scopings (die nicht umfangreich genug ist,) drei verschiedene Verfahren zu unterscheiden.

Insbesondere die ersten beiden Begriffe sind Erweiterungen des traditionellen Scoping-Begriffs (Abgrenzung eines Projekts, also welche Projekteigenschaften Eingang in ein Projekt finden sollen, und welche nicht..) ,da eine eindeutige Definition des Projektbegriffs im Produktlinienumfeld nicht mehr möglich ist.

Der dritte Begriff Asset Scoping ergibt sich durch die Übertragung des Scoping-Begriffs auf die Entwicklung der Produktlinienplattform.

Die in der Plattform enthaltenen Komponenten, Dokumente, etc werden im Englischen unter dem Begriff Asset (deutsch = Artefakt) zusammengefasst.

Bei der Festlegung der Produktlinien- und der Domänengrenzen geht es um Festlegungen im Problemraum (Was ist Gegenstand der Entwicklung), bei den Komponentengrenzen geht es um den Lösungsraum (die Festlegung der wiederverwendbaren Realisierung).

Scoping

## Gefahren des Scopings

Zu groß definierter Scope:

- geringen Anteil an Gemeinsamkeiten
- keine Grundlage für effiziente Entwicklung einer Produktlinie
- Wiederverwendungsgrad sinkt auf ein Minimum

Zu kleine Scopes:

- Gefahr nicht flexibel genug zu sein
- Integrierung neuer Produkte fällt durch geringere Variationsmöglichkeiten schwer
- Kurze Lebensdauer der Produktlinie

Folie 18 Urte Rietdorf, Cüneyt Egin

In der Praxis sind zwei Gefahrenbereiche während der Erstellung eines Scopes eingekreist worden

1. Zu groß definierter Scope der Produkte, mit einem geringen Anteil an Gemeinsamkeiten, bietet keine Grundlage für eine Effiziente Entwicklung innerhalb einer Produktlinie. Dies beruht darauf, dass die einzelnen Funktionsbausteine nacheinander entwickelt und der Wiederverwendungsgrad auf ein Minimum sinkt.
2. Zu kleine Scopes laufen Gefahr nicht flexibel genug zu sein. Eine Integrierung neuer Produkte fällt durch die im Vorfeld definierten geringen Variationsmöglichkeiten sehr schwer. Dadurch ist die Lebensdauer einer solchen PL als kurz einzuschätzen.



Scoping

## Ansatz von PuLSE-Eco

- Product Line Mapping
  - Grobbeschreibung der Produktlinie mit Domänen, Features und Beziehungen
- Domain Potential Analysis
  - Bereich mit höchstem Wiederverwendungsnutzen und Risiko nach Kriterientatalog
- Reuse Infrastructure Scoping
  - Welche wiederverwendbaren Komponenten werden entwickelt -> ROI

---

Folie 19 Urte Rietdorf, Cüneyt Egin

Innerhalb des PuLSE-Frameworks wurde der PuLSE-Eco Ansatz entworfen, der seinen Fokus auf dem Domain- und Asset-Scoping hat. Sein Ziel ist in erster Linie eine Bestandsaufnahme, mit Nutzen und Risikoanalyse der wiederverwendbaren Komponenten. Der PuLSE-Eco-Ansatz untergliedert sich in drei Teilbereiche, Product Line Mapping mit einer Produktkarte, Domain Potential Analysis (für das Domain Scoping) welches eine Nutzen-Risiko-Übersicht gibt (Darstellbar als Tabelle mit Pfeilen) , und das Reuse Infrastructure Scoping (für das Asset Scoping), welches die Komponenten auf den Return on Invest (ROI) untersucht.

## Produktkarte nach PuLSE-Eco

	Speicherformat	Existiert	In Entwicklung	Hypothetisch
Profi- Bildbearbeitungs Programm	JPG	X		
	GIF	X		
	BMP	X		
	UBF	X		
	EPS		X	
Digitalkamera Bildbearbeitungs Programm	JPG	X		
	GIF	X		
	BMP		X	
	UBF			
	EPS			
Internet Bildbearbeitungs Programm	JPG			X
	GIF			X
	BMP			
	UBF			
	EPS			
Vektororientiertes Bildbearbeitungs Programm	JPG			X
	GIF			X
	BMP	X		
	UBF	X		
	EPS	X		

Für das Produkt Line Mapping kann die Darstellung in Form einer Produktkarte erfolgen, bei dem die Features katalogisiert aufgelistet, werden, um eine Bestandsaufnahme zu erleichtern.



---

# Anforderungsanalyse

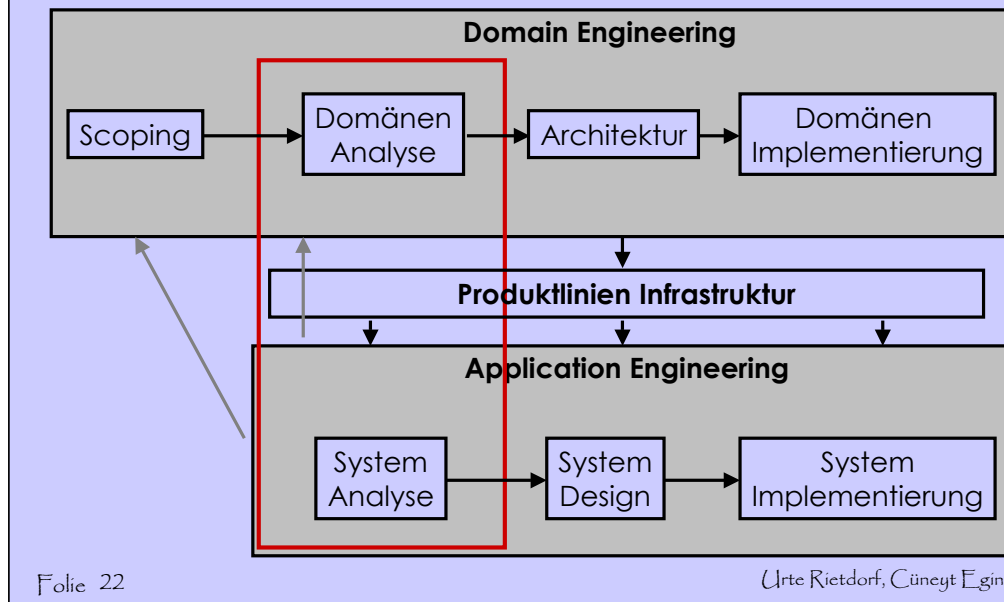
---

Folie 21

Urte Rietdorf, Cüneyt Egin



## Einordnung in Produktlinienentwicklung



In der Produktlinienentwicklung unterteilt sich die klassische Anforderungsanalyse in die Bereiche

Domänenanalyse und

Application Analysis auf, wie in dieser Darstellung gezeigt.

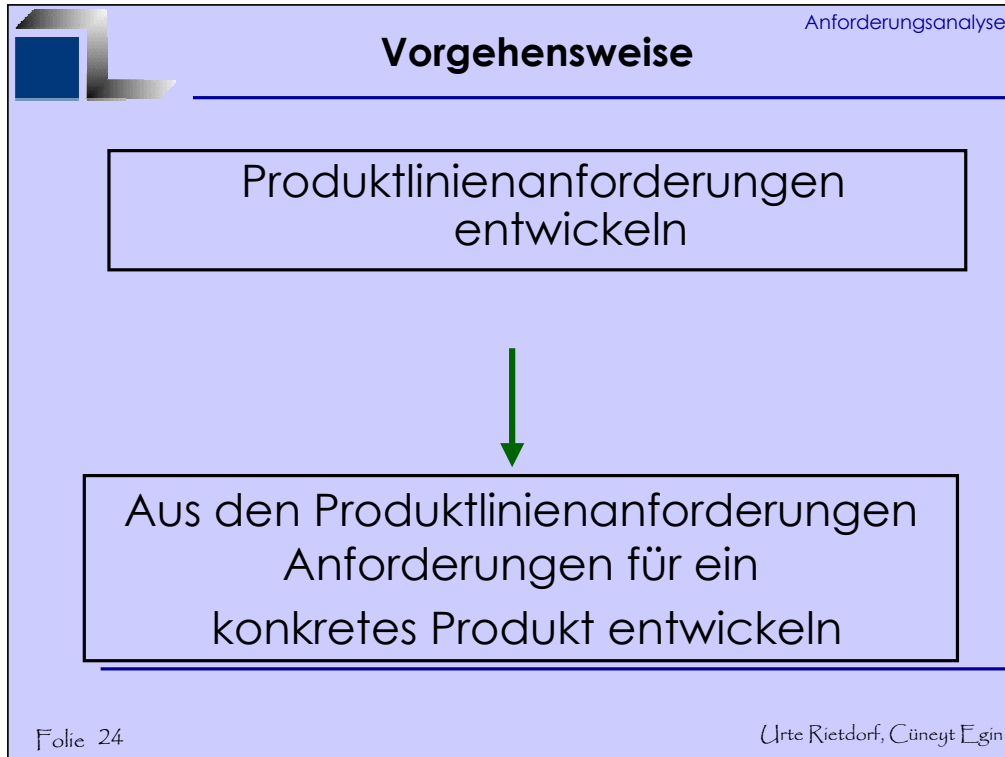
## Anforderungsanalyse

- Es entstehen zwei verschiedene Formen der Anforderungsanalyse
  - Domänen Analyse
    - zielt auf integrierte Analyse der verschiedenen Produkte der Produktlinie ab
    - Führt zu einer integrierten Darstellung der verschiedenen Produkte
    - Liefert als Endergebnis das Domänenmodell, welches in die Produktlinieninfrastruktur einfließt
  - Applikations- (System) Analyse
    - Produkthanforderungsanalyse
    - führt (auf Grund des Domänenmodells) zu einer Analyse des aktuell zu entwickelnden Produkts

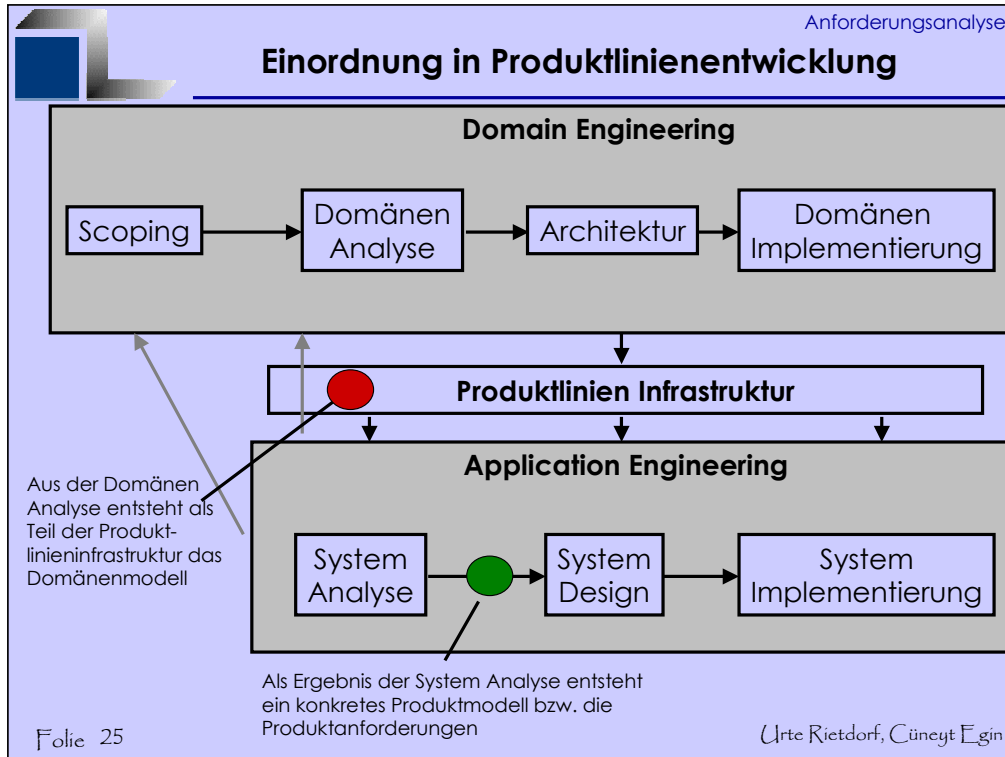
Für die Anforderungsanalyse hat die Produktlinienentwicklung insbesondere zur Folge, dass zwei verschiedene Formen der Anforderungsanalyse entstehen:

Die Domänenanalyse, die auf die Analyse der verschiedenen Produkte der Produktlinie abzielt und so zu einem Domänenmodell für die verschiedenen Produkte führt,

und die Produkthanforderungsanalyse, die basierend auf einer bereits durchgeführten Domänenanalyse zu einer Analyse des aktuell zu entwickelnden Produkts führt.



Also werden in der Domänen Analyse die Produktlinienanforderungen definiert und entwickelt, aus denen später in der Application Analyse die konkreten Anforderungen der Produkte entwickelt werden.



Die beiden Endprodukte der Anforderungsanalyse sind zum einen:  
 Das Domänenmodell (entstanden aus der Domänen Analyse)  
 Zum anderen ein Produktmodell mit den konkreten Produkthanforderungen



---

# Domänenanalyse

---

Folie 26

Urte Rietdorf, Cüneyt Egin




## Charakteristika einer Domäne

Eine Domäne wird charakterisiert

- durch ihren Bereich (Abgrenzung), seine Informationen (Objekte), Features und Verwendungen, sein Verhalten und operationale Charakteristiken [SEI 1999]
- durch gemeinsame Expertise, gemeinsames Design, und gemeinsamen Markt [Mili et al. 1999]

[HG2005]


Domänenanalyse

## Aufgaben der Domänenanalyse

---

- Identifiziert, sammelt, organisiert und repräsentiert relevante Informationen einer Domäne
  - gemeinsame und variable Artefakte der Produktlinie werden definiert!
  - Wiederverwendbare Komponenten identifizieren
  - ⇒ Explizite Repräsentation der Produktlinienvariabilität
- ⇒ **Ziel der Domänenanalyse ist ein Domänenmodell**

---

Folie 28
Urte Rietdorf, Cüneyt Egin

**Def. Domäne:** „Menge von Konzepten und Technologien in einem Wissensbereich“ (Bass,Clemens,Donohoe)

**Def. Domänenanalyse:** „ Domänenanalyse ist ein Prozess zur Identifikation, Akquisition und Evaluation von wiederverwendbaren Informationen, die im Zuge der Entwicklung von Systemen für Klassen von Applikationen oder von Problem domänen wiederverwendbar sind.“ (Arango, SW Definition Methods)

Hier wird der Scope der Produktlinie sowie die variablen und gemeinsamen Bestandteile für die spätere Entwicklung unterschiedlicher Produkte auf der Basis einer Produktlinie definiert

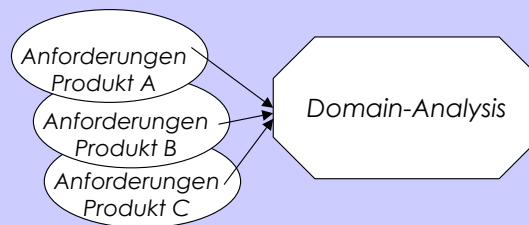
**Also:** Systematische Untersuchung eines Bereichs auf Zusammenhänge und Abläufe ->

**Ziel:**

- Gewinnung der wiederverwendbaren Aspekte
- Erstellen eines Domänenmodells (= i.d. Praxis Dokumente mit extrahiertem Wissen aus den unterschiedlichen Sichten) z.B.
  - Erstellt Begrifflexikon zur Vermeidung von Diskussionen und Schaffung von Klarheit

## Aufgaben der Domänenanalyse II

- Strukturierte Extraktion der Anforderungen an eine Produktlinie
  - Aus Benutzerdokumentation existierender Einzelsysteme
  - Wissen von Domänen-Experten
  - Vorhandenen Theorien und Technologien in der Domäne




Folie 29

Urte Rietdorf, Cüneyt Egin

- ❖ der Analyse existierender Systeme und deren Entwicklungen,
- ❖ Wissen von den Domänen-Experten,
- ❖ zugrunde liegender Theorien und Technologien in einer Domäne

Produktlinien bauen meist auf bereits existierenden Systemen auf, d.h. ein Unternehmen, welches eine Produktlinie erstellen möchte, hat fast immer schon Erfahrung in der Erstellung der Einzelprodukte. Daher sollten für diese existierenden Einzelsysteme auch Benutzerdokumentationen vorhanden sein. Diese Benutzerdokumentationen werden verwendet, um die strukturierte Extraktion der Anforderungen an die Produktlinie erarbeiten zu können, also zum Herausfinden der einzelnen Features, Use Cases und aller anderen Anforderungen; vor allem der Identifikation der gemeinsamen und variablen Anforderungen.

Die Unterscheidung der gemeinsamen und variablen Bestandteile ist eine wesentliche Eigenschaft der Produktlinien!



Domänenanalyse

## Domänenmodellierung

---

- Beschaffenheit der Domäne untersuchen (innere Struktur aus unterschiedlichen Perspektiven betrachten und dokumentieren)
  
- Domänenmodell erstellen
  - Konzepte, um aus der Domäne eine Spezifikation eines Systems zu ermöglichen
  - Methoden zur Transformation der Spezifikation in Source-Code
  
- Domänenmodell dient als
  - Quelle und Referenz
  - Repository (für das gesammelte Wissen über den Anwendungsbereich)
  - Spezifikation für die Implementierung von wiederverwendbaren Komponenten

---

Folie 30 Urte Rietdorf, Cüneyt Egin

### Domänenmodellierung

- Untersuchung der Beschaffenheit einer Domäne. Die innere Struktur wird aus unterschiedlichen Perspektiven betrachtet und dokumentiert
- Diese Phase ist eigentlicher Kern der Domänenanalyse
- Hier wird das eigentliche Domänenmodell erstellt, welches Grundlage für die Strukturierung der Produktlinie dient
- **Grundlage für gutes Domänenmodell = Fachwissen**
  - Aus der Organisation
  - Tech. Literatur
  - Kundengespräche
  - Marktanalysen
  - Kontaktperson des Auftraggebers
  - Berater / Externe Experten

## Probleme der Domänenanalyse

- großer Zeitaufwand
- Benötigt
  - Produktlinieningenieure und
  - Experten aus der Domäne
- Benötigt Kenntnisse des Marktes und der technischen Realisierbarkeit

Ein großes Problem der Domainanalyse besteht darin, dass sie sehr zeitaufwändig ist. Zudem benötigt sie hochqualifiziertes Personal: Zum einen Produktlinieningenieure, welche die Technologiekenntnisse der Produktlinienerstellung beherrschen, zum anderen Experten aus der Domäne, die Erfahrungen mit den zu erstellenden Produkten besitzen. Es müssen also Kenntnisse sowohl über den Markt, als auch über die technische Realisierbarkeit der Produktlinie vorhanden sein.

## Analysen Überblick

- **Feature Oriented Domain Analysis (FODA)**  
wichtigste Technik, modelliert sichtbare Features
- **Scope, Variability and Commonality Analysis (CA)**  
erzeugt Domänenspezifische Sprache mit der PL beschrieben wird
- **Domain Analysis and Design Process (DADPP)**  
OO-Analyse, Unterschiede werden als notwendige Anpassung generischer (Objekte) modelliert
- **Reuse-Driven Software Process (RSP)**  
Das Vorhandene wird genutzt
- **Organisational Domain Modelling (ODM)**  
hierarchisches Prozessmodell

**Feature Oriented Domain Analysis (FODA)** gilt als die wichtigste Technik und modelliert die sichtbaren Features der Anwendungen.

Die **Scope, Variability and Commonality Analysis (SVC)** ist Teil des PL-Ansatzes FAST. Sie erzeugt eine domänenspezifische Sprache, mit der die PL beschrieben wird.

**Domain Analysis and Design Process (DADPP)** beruht auf objektorientierter Analyse. Die Unterschiede werden als notwendige Anpassungen eines (generischen) Objektes modelliert.

Beim Syntheseprozess des **Reuse-Driven Software Process (RSP)** sind keine Besonderheiten angegeben, d.h. ich benutze das vorhandene

Beim Analysis Process of **Organisational Domain Modelling (ODM)** wird ein hierarchisches Prozessmodell definiert, in dem die Domänenanalyse namentlich nicht auftaucht. Es sind aber alle Elemente für eine solche Analyse in dem Prozessmodell vorhanden.

## FODA

- Features Oriented Domain Analysis
- In den 90ern am SEI der CMU entwickelt
- FODA ist einer der meist verbreitetsten Ansätze im Gebiet des PLE's

SEI: Software Engineering Institut

CMU: Carnegie Mellon University

FODA ist einer der meist verbreitetsten und ausgereiftesten Ansätze im PLE. Sie ist 1990 an der SEI am MCU entwickelt worden.

## Was ist FODA?

- Es beschreibt Eigenschaften (Features) von einer Klasse von Anwendungen...
- ... aus der Sicht des Kunden oder des Endverbrauchers
- Sie durchläuft das Scoping, Domain Engineering und das Application Engineering
  - Jede Phase ist unterteilt und erzeugt Dokumente oder Teilmodelle während des Durchlaufs

Es beschreibt „Features“/Eigenschaften von einer Klasse von Anwendungen aus der Sicht des Kunden oder des Endverbrauchers.

Unter Features versteht man dabei ein bestimmtes Verhalten, das durch Anforderungen definiert bzw. spezifiziert wird.

FODA definiert solche Eigenschaften sehr breit. So können dies je nach Zusammenhang funktionale oder auch nicht-funktionale Eigenschaften sein.

Jede der 3 Phasen des PLE (Scoping, Domain Engineering, Application Engineering) wird in der FODA in Unteraktivitäten aufgeteilt. Diese Unteraktivitäten haben Teilmodelle oder Dokumente als Ergebnis ihrer Durchführung.



## Ergebnisse von FODA im Scoping


Exkurs Analysetechniken

Phase	Aktivität	Ergebnis	Darstellung
Scoping	Scoping	Kontextmodell	Datenflussdiagramm
		Strukturmodell	Blockdiagramm

Folie 35

Urte Rietdorf, Cüneyt Egin

Es wird hierbei einerseits die Umgebung einer Domäne festgehalten und dokumentiert und andererseits werden die Datenflüsse und der Informationsaustausch untersucht.

 <b>Ergebnisse von FODA im Domain Engineering</b> <span style="float: right;">Exkurs Analysetechniken</span>			
Phase	Aktivität	Ergebnis	Darstellung
Domain Engineering	Feature Analyse	Feature Modell	UND/ODER-Baum
			Features-Liste
	Informations-Analyse	Informations-Modell	Entity-Relationship Diagramm
	Funktionale Analyse	Datenfluss-modell	Datenfluss-diagramm
Zustands-Übergangs-modell		Zustands-Übergangs-diagramm	

Folie 36

Urte Rietdorf, Cüneyt Egin

Das **ERM (Entity-Relationship Modell)** ist das Ergebnis der sog. Informationsanalyse.

In der **Informationsanalyse** werden die Objekte und Relationen innerhalb einer Domäne beschrieben.

Dies geschieht in der domänenspezifischen Sprache und dient der Veranschaulichung und der Analyse des Problembereichs.

Das **funktionale Modell** entsteht im Laufe der **funktionalen Analyse**, die sowohl die Datenbewegungen innerhalb der Domäne, als auch Zustandsänderungen der Objekte berücksichtigt. Das Funktionale Modell beinhaltet daher sowohl Datenflussdiagramme, als auch Zustandsübergangdiagramme.



## Ergebnisse von FODA im Application Engineering

Exkurs Analysetechniken

Phase	Aktivität	Ergebnis	Darstellung
Application Engineering	Architektur-entwurf	Architektur-modell	Block-diagramm

Folie 37

Urte Rietdorf, Cüneyt Egin



## Das FODA Prinzip

- Pflichtfeatures
  - Features müssen von allen vorgewiesen werden
- Wahlpflichtfeatures
  - Features müssen von best. Prod. vorgewiesen werden
- Wahlfeatures
  - Features können optional von einzelnen vorgewiesen werden

Diese (vorherig erwähnten) Techniken und Aktivitäten sind auch anderen Bereichen der SW-Entwicklung bekannt, jedoch ist die Besonderheit der Features-Oriented Domain Analysis Methode das Featuresmodell.

Es bildet die Gesamtheit der Eigenschaften ab die von einer Klasse von Produkten umfasst werden. Diese Gesamtheit entsteht im Laufe der dafür erforderlichen Featuresanalyse.

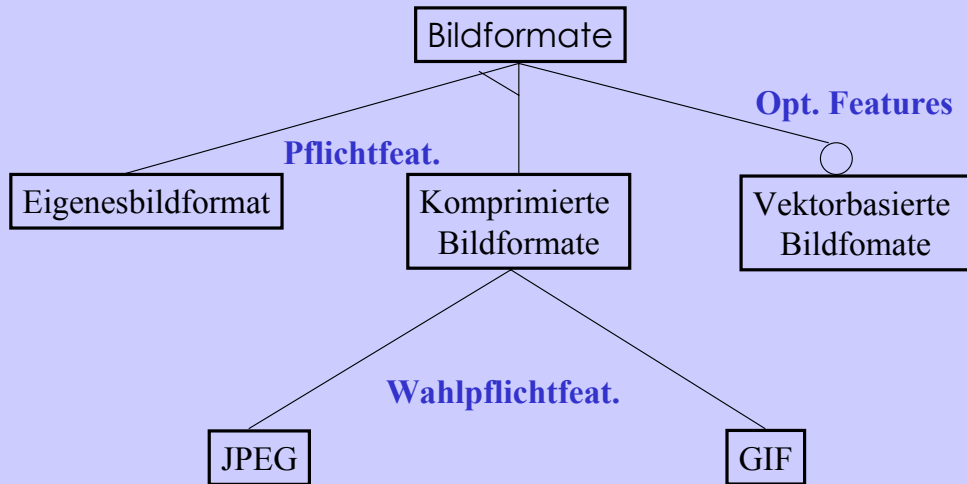
Im folgenden werden unterschiedlichen Arten von Features kurz vorgestellt:

**Pflichtfeatures** – Eine Menge der Eigenschaften, die jedes Produkt der Klasse vorweisen muss.

**Wahlpflichtfeatures** – Eine Menge der Eigenschaften, aus denen das Produkt eine vorweisen muss.

**Wahlfeatures** – Eine Menge der Eigenschaften, die einzelne Produkte optional vorweisen können.

Diese 3 Arten von Features / Eigenschaften werden in einer Featuresliste beschrieben und die Abhängigkeiten untereinander in einem UND/ODER-Baum dokumentiert.

**Beispiel:**

Folie 39

Urte Rietdorf, Cüneyt Egin

**Erläuterung des Beispiels:**

Jedes Produkt aus unserer PL muss unser eigenes Bildformat, komprimierte Bildformate und optional vektorbasierte Formate verarbeiten können. Wobei das komprimierte Bildformat eines bestimmten Produkts entweder JPEG oder GIF sein muss.

**Fazit:**

Ein solcher Feature-Baum zusammen mit einer Liste der Features (ebenfalls Ergebnis der Featureanalyse) ermöglicht eine übersichtliche Beschreibung der Produkteigenschaften und dadurch der Produkte an sich. So lassen sich die Assets in Hinsicht auf die jeweiligen produktspezifischen Eigenschaften einrichten.

**Fähigkeiten dieses Ansatzes:**

Dieser Ansatz ermöglicht es, produktspezifische Eigenschaften durch Verfeinerung bzw. Verästelung in dem Feature-Baum zu definieren und Gemeinsamkeiten der Produkte durch Abstraktion herauszulösen.

FODA liefert so ein umfassendes und detailliertes Domänenmodell als Ergebnis, in dem eine Produktlinie (Klasse von Produkten) beschrieben wird. Gemeinsamkeiten von den jeweiligen Produkten werden so auf funktionaler Ebene dargestellt und können dadurch komponentenbasiert entwickelt werden.

**Nachteil von FODA:**

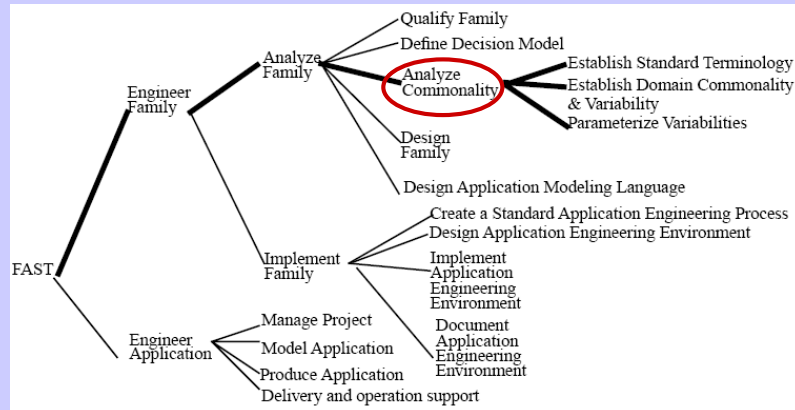
Für die Phase des Application Engineering bietet FODA nur schwache Unterstützung. Der Schwerpunkt von FODA liegt in einer sehr guten und bewährten Methode zur umfassenden Domain Analysis und nicht mehr.

## Commonality Analysis

- Von Lucent Technologies Bell Laboratories
- Untersucht systematisch welche Requirements alle Produktfamilien gemeinsam haben und welche sich unterscheiden
- Teil des FAST-Ansatzes (family oriented abstraction, specification, and translation)

Dieses systematische Vorgehen in dem Gemeinsamkeiten und Unterschiede definiert bzw. herausgearbeitet werden, ist nahezu die Essenz des Produktlinien Konzepts.

# FAST – Aktivitätendiagramm



Wie zusehen ist, wird in der Commonality Analyse eine spezifische Terminologie entwickelt, die Gemeinsamkeiten und Unterschiede der Mitglieder der Produktfamilie ermittelt und die jeweiligen Variationsmöglichkeiten parametrisiert.

Zusätzlich gehen aus dem FAST-Ansatz eine Reihe von Dokumenten hervor aus denen wir die für das CA wichtigen herausgegriffen haben:

### **Commonalities:**

Dies ist eine strukturierte Liste von Annahmen, die für alle Mitglieder Domäne zutreffen

### **Variabilities:**

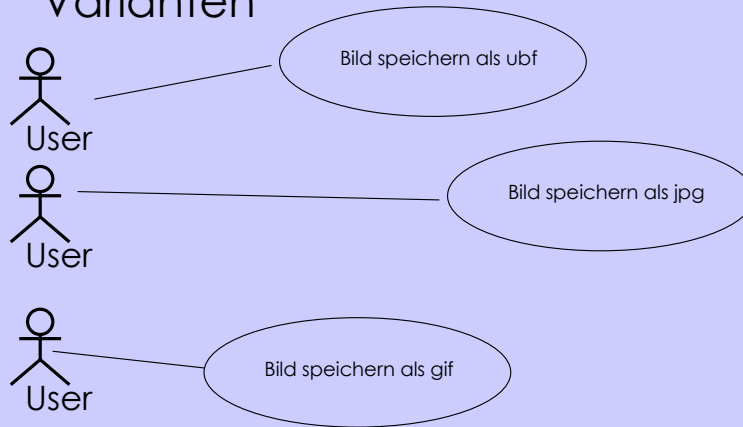
Eine strukturierte Liste von Annahmen, die Auskunft gibt inwieweit die einzelnen Mitglieder der Domäne sich unterscheiden.

### **Parameters of Variabilities:**

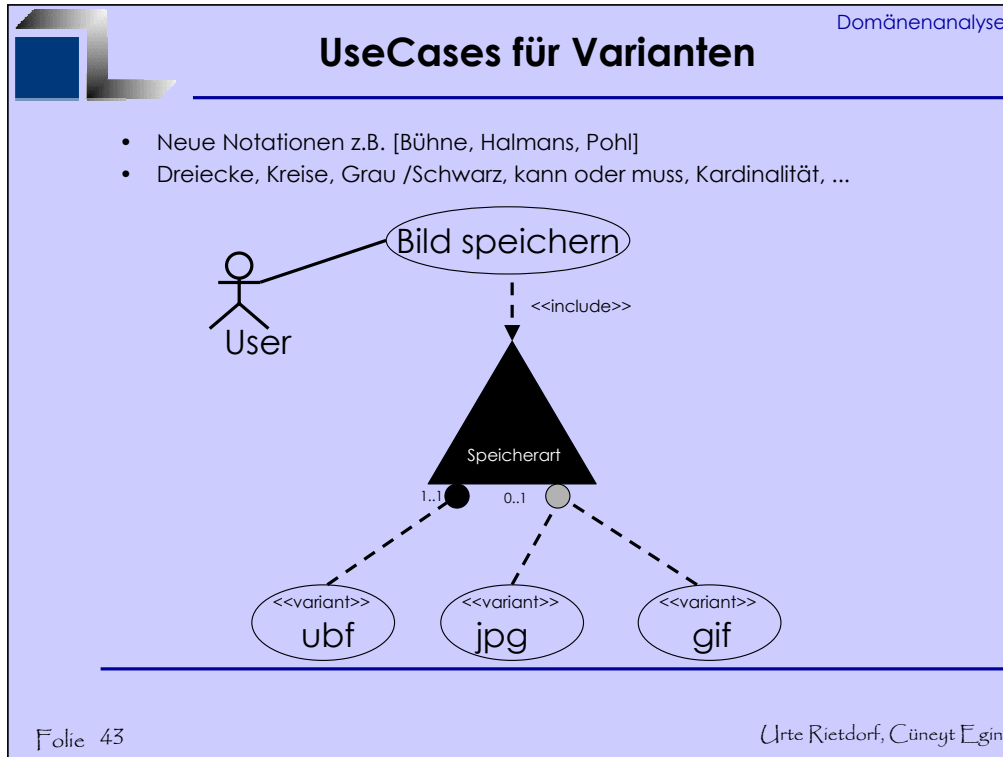
Quantifiziert die Variabilität und spezifiziert das Ausmaß.

## UseCases in der Domänenanalyse

- Normales UseCase kennt keine Varianten



Um die Produktvariabilität zu Dokumentieren werden in der Domänenanalyse unter anderem Use Cases eingesetzt. Ein Problem dabei besteht allerdings darin, dass die herkömmliche UML für Use Cases keine Variationspunkte kennt. So würde der Use Case „Bild Speichern unter“, bei dem es die Möglichkeiten des Speicherns als „jpg“, „gif“ oder „bmp“ gäbe, in drei verschiedenen Use Cases realisiert werden.



Um die Produktlinienvariabilität in UseCases darstellen zu können, können weitere graphische Möglichkeiten der UML genutzt werden:

Dafür wurden verschiedene Ansätze entwickelt. Dieser Ansatz von Bühne, Halmans und Pohl beschriebene Ansatz erweitert die übliche Notation der Use Cases in der UML folgendermaßen:

Ein Dreieck symbolisiert den Variationspunkt;

Dieser ist durch eine gestrichelte Linie an den Use Case angehängt.


Von diesem Variationspunkt gehen die Verschiedenen Variationen ab.

An den Verbindungen sind auf Seite des Variationspunktes Kugeln angebracht, welche anzeigen, ob die Variante optional oder Pflichtbestandteil ist.

Dies wird durch die Farbe gekennzeichnet: grau für optional, schwarz für verpflichtend (mandatory).

Dieselbe Farbkennung gilt für den Untergrund des Variationspunktes, der Angibt ob dieser an sich zwingend erforderlich oder optional ist.

Neben den Kugeln stehen die Kardinalitäten der Variation.



## UseCases für Varianten

Domänenanalyse

---

Name	Authentifizierung	
Ziel:	User speichert Bild ab	
Standardablauf	Schritt	Aktion
		1 User aktiviert das Menü "Speichern unter"
		2 Speichermaske öffnet sich
		3 User gibt Name des Bildes an
		4 User wählt Bildformat "ubf" aus
		5 User klickt den Button "speichern"
Fehlerfälle	Schritt	Aktion
		5a Speicherung nicht erfolgreich, Fehlerfenster erscheint
Variation	Schritt	Aktion
		v1-4 User wählt Bildformat "jpg" aus
		v2-4 User wählt Bildformat "gif" aus

---

Folie 44
Urte Rietdorf, Cüneyt Egin

Beispiel für eine textuelle Beschreibung eines Use Cases mit Variante



# Applikationsanalyse



## Applikationsanalyse

- Gemeinsamkeiten
- Variabilitäten
- Produktspezifika
  
- Produktdefinition & Produktableitung
  - Kunde stellt sich aus den gegebenen Features sein konkretes Produkt zusammen
  - Nutzt Ergebnisse aus der Domain-Analyse, um kundenspezifische Produkte zu definieren
  - Berücksichtigung der Produktlinienvariabilität

Die Application-Analyse nutzt die Ergebnisse aus der Domain-Analyse (gefundene Features, Use Cases, gemeinsame und variable Artefakte), und definiert daraus die kundenspezifischen Produkte. Dabei ist es unbedingt notwendig, dem Kunden die Produktlinienvariabilität zu kommunizieren, damit dieser entscheiden kann, in wieweit seine Anforderungen von der Produktlinie erfüllt werden können.



## Produktlinienvariabilität vermitteln


- Dem Kunden die Möglichkeiten der Produktlinie aufzeigen!!

Dem Kunden müssen die Möglichkeiten der Produktlinie, also die Produktvariabilität unbedingt aufgezeigt werden, damit er sich die Anforderungen für sein konkretes Produkt zusammenstellen kann. Dazu können die Szenarien, Use Cases und Dokumentationen aus der Domänenanalyse herangezogen werden.

## Produktableitung

- 3 Schritte zur Produktableitung:
  1. Abdeckung
    - Szenarien der Produktlinie werden Selektiert
    - Beziehungen zu kundenspezifischen Anforderungen werden dokumentiert
  2. Deltas
    - Anforderungen die nicht (vollständig) erfüllt werden können  
⇒ Trade-Off
  3. Rückfluss
    - Für jedes Delta der Produktszenarien zu den Produktionslinienszenarien:  
entscheiden ob es auch auf andere Produkte der Produktlinie übertragen werden soll

Es erfolgt eine spezifische Produktableitung, indem aus den Möglichkeiten - die sich aus der Domain Analyse für die Produkte der Produktlinie ergeben - die spezifischen kundenorientierten Produkte abgeleitet werden. Dabei kann die Produktableitung in drei Schritte unterteilt werden.



Applikationsanalyse

## Stufen zur Einordnung von Trade-Off-Entscheidungen

1. Vollständige Abdeckung
2. Ausnutzung einer Variante
3. Definition einer neuen Variante
4. Löschen von Funktionalität
5. Hinzufügen von Funktionalität
6. Änderung der Architektur
7. Anforderung kann nicht realisiert werden

[BKPS 2004]

---

Folie 49 Ulte Rietdorf, Cüneyt Egin

Um den Kunden bei seiner Trade-off-Entscheidung zu unterstützen, wurde von Halmans und Pohl [1] eine hierarchische Einstufung der Aufwandsabschätzung beschrieben. Anhand dieser Liste kann der Kunde abschätzen wie einfach die Umsetzung seiner Anforderung zu realisieren wäre.

- Vollständige Abdeckung  
Die Anforderung wird vollständig von der Produktionslinie abgedeckt, und zwar vom den für alle Produkte gemeinsamen Teil
- Ausnutzung einer Variante  
Die Anforderung kann durch eine bereits vorhandene Variante eines Variationspunktes abgedeckt werden
- Definition einer neuen Variante  
Die Anforderung kann abgedeckt werden, wenn an einen realisierten Variationspunkt eine neue Variante entwickelt wird
- Löschen von Funktionalität  
Um die Anforderung erfüllen zu können, muss eine bereits vorhandene Funktionalität ausgeblendet werden, was das Definieren eines neuen Variationspunktes beinhaltet
- Hinzufügen von Funktionalität  
Um die Anforderung realisieren zu können, muss ein neuer Variationspunkt



---


## Konkretes Beispiel

---

Folie 50

Urte Rietdorf, Cüneyt Egin





Konkretes Beispiel

## Unternehmensprofil

- ein großes Unternehmen, welches Software im Bereich Bildverarbeitung erstellt.

Folie 51

Urte Rietdorf, Cüneyt Egin

Unser Unternehmen:

Wir erstellen Softwareprodukte im Bereich Bildverarbeitung

Konkretes Beispiel

## Scoping

- Unsere Produkte aus dem Bereich Bildbearbeitung:
  - Bildbearbeitung Pixelbasiert
  - Digitalkamerabildbearbeitung
  - Internetbildbearbeitung
  - Vektororientierte Bildbearbeitung

Folie 52

Urte Rietdorf, Cüneyt Egin

Dabei haben wir insbesondere im Bildverarbeitungsbereich unterschiedliche Produkte mit ähnlichen Komponenten, bei denen sich die Frage stellt, ob die Zusammenfassung zu einer Produktionslinie sinnvoll erscheint.


Die Bildbearbeitungsserie umfasst dabei

Ein professionelles Bildbearbeitungsprogramm mit extrem vielen Features

Ein Programm welches beim Kauf einer Digitalkamera unserer Firma kostenlos mitgegeben wird, es besitzt nur sehr einfache Features

Ein spezielles Programm zur Bildbearbeitung für die Internetanwendung. Auch dieses Programm hat nur eingeschränkte Features

Ein vektororientiertes Bildbearbeitungsprogramm zur Erstellung von Vektorgrafiken

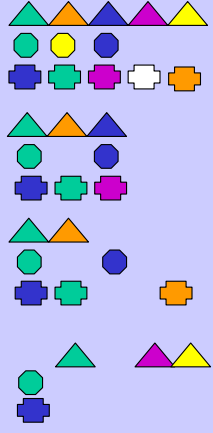


# Domänenanalyse

Konkretes Beispiel

---

1. Bildbearbeitung Pixelbasiert
  1. Speicher: jpg, gif, bmp, ubf, eps
  2. Bildmodi: RGB, CMYK, HTML,
  3. Filter: drehen, spiegeln, retuschieren, Blenden, Verzerren
2. Digitalkamerabildbearbeitung
  1. Speicher: jpg, gif, bmp
  2. Bildmodi: RGB, HTML,
  3. Filter: drehen, spiegeln, retuschieren
3. Internetbildbearbeitung
  1. Speicher: jpg, gif,
  2. Bildmodi: RGB, HTML,
  3. Filter: drehen, spiegeln, verzerren
4. Vektororientierte Bildbearbeitung
  1. Speicher: ubf, vbf, eps
  2. Bildmodi: RGB,
  3. Filter: drehen,



---

Folie 53

Urte Rietdorf, Cüneyt Egin


Um den Rahmen nicht zu sprengen wurden hier nur einige Bereiche betrachtet:  
 Die Speicherformate welche angeboten werden sollen  
 Die Bildmodi in denen gearbeitet werden kann  
 Einige Filter die das Programm besitzen soll:



Konkretes Beispiel

# Domänenanalyse

Unsere Komponenten:



Folie 55

Urte Rietdorf, Cüneyt Egin

Also werden in dieser Produktlinie folgende Komponenten entwickelt:




Speicher: jpg, gif, bmp, ubf, eps





Bildmodi: RGB, CMYK, HTML,




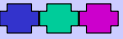
Filter: drehen, spiegeln, retuschieren, Blenden, Verzerren





Konkretes Beispiel




## Applikationsanalyse

Komponenten:   

Produkt 1  +   

Produkt 2  +   

Produkt 3  +   

Neues Produkt  +  

Folie 56 Urte Rietdorf, Cüneyt Egin

In der Applikationsanalyse werden mit diesen Komponenten neue Produkte erstellt, nämlich

Produkt 1, das Profiprogramm




Produkt 2, das Digitalbildbearbeitungsprogramm





Produkt 3, das Internetbildbearbeitungsprogramm




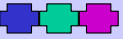
Zusätzlich wollte der Markt ein weiteres Produkt, für das eine neue Komponente erforderlich ist. Hierbei muss geprüft werden ob es sinnvoll wäre, die neue Komponente direkt in die Produktlinie mit aufzunehmen, oder nur in das neue Produkt einzubauen.





Konkretes Beispiel









## Applikationsanalyse

Komponenten:   

Produkt 1  +   

Produkt 2  +   

Produkt 3  +   

Neues Produkt  +       

Folie 57 Urte Rietdorf, Cüneyt Egin

In der Applikationsanalyse werden mit diesen Komponenten neue Produkte erstellt, nämlich

Produkt 1, das Profiprogramm

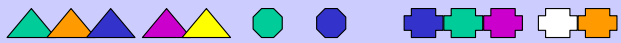
Produkt 2, das Digitalbildbearbeitungsprogramm



Produkt 3, das Internetbildbearbeitungsprogramm



Zusätzlich wollte der Markt ein weiteres Produkt, für das eine neue Komponente erforderlich ist. Hierbei muss geprüft werden ob es sinnvoll wäre, die neue Komponente direkt in die Produktlinie mit aufzunehmen, oder nur in das neue Produkt einzubauen.



Konkretes Beispiel



## Applikationsanalyse

Komponenten: 

Produkt 1  + 

Produkt 2  + 

Produkt 3  + 

Neues Produkt  + 

Folie 58

Urte Rietdorf, Cüneyt Egin

In der Applikationsanalyse werden mit diesen Komponenten neue Produkte erstellt, nämlich

Produkt 1, das Profiprogramm

Produkt 2, das Digitalbildbearbeitungsprogramm

Produkt 3, das Internetbildbearbeitungsprogramm

Zusätzlich wollte der Markt ein weiteres Produkt, für das eine neue Komponente erforderlich ist. Hierbei muss geprüft werden ob es sinnvoll wäre, die neue Komponente direkt in die Produktlinie mit aufzunehmen, oder nur in das neue Produkt einzubauen.



---

**Fazit**

---

Folie 59

Urte Rietdorf, Cüneyt Egin



- Wenig konkretes Material
- Oft Quellen ohne Herkunftsangabe (Internetquellen)
- Oracle Prinzip birgt Gefahren
- Bedarf ein hohes Maß an Bürokratie versus „Woher weiß ich wie es gemacht werden soll“
  - ⇒ könnte abschreckend wirken
- Sehr spannend
- Zukunftsträchtig
- Wenn richtig eingesetzt sehr lohnend



## Quellenverzeichnis

- [BKPS 2004] B.Böckle, P.Knauber, K.Pohl, K.Schmid **Software-Produktlinien Methoden, Einführung und Praxis**, dpunkt.verlag 2004
- [Bühne, Halmans, Pohl] Stan Bühne, Günter Halmans, Klaus Pohl **Modelling Dependencies between Variation Points in Use Case Diagrams**
- [HG2005] Univ.Prof. Dipl.-Ing. Dr. techn. Harald Gall, **Domain Analysis**, Universität Zürich, Technische Universität Wien ( <http://www.infosys.tuwien.ac.at/Teaching/Courses/SWV/slides/reuse05-domain-analysis.pdf>)
- [soko2005] <http://www.software-kompetenz.de>
- David M. Weiss **Defining Families: The Commonality Analysis** Lucent Technologies Bell Laboratories
- Völter **Software System Families and Product Lines**
- Thomas von der Maßen, Horst Lichter **Modellierung von Variabilität mit UML Use Cases** Lehr- und Forschungsgebiet Informatik III RWTH Aachen
- Birgit Geppert, Klaus Schmid (Eds.) **Proceedings International Workshop on Requirements Engineering for Product Lines** September 9, 2002 Co-located with the IEEE Joint International Requirements Engineering Conference (RE02)
- Wadim Schleicher **Domain Analysis and Scoping**
- Periklis Sochos, Matthias Riebisch, Ilka Philippow **Featuregesteuerte Architekturgestaltung zwecks Wartbarkeit und Evolution von Produktlinien**
- Kang, Cohen, Hess, Nowak, Peterson **Feature-Oriented Domain Analysis (FODA) Feasibility Study Technical Report SEI CMU**
- [www.sei.cmu.edu/23-engineering/foda.html](http://www.sei.cmu.edu/23-engineering/foda.html)



Vielen Dank...  
... für die Aufmerksamkeit

Gibt es Fragen?

