Software Product Lines

Concepts, Analysis and Implementation

Feature-Oriented Software Product Lines

Dr. Malte Lochau
Malte.Lochau@es.tu-darmstadt.de

Inhalt

I. Einführung
- Motivation und Grundlagen
- Feature-orientierte Produktlinien

- Grundbegriffe: Produktlinie, Feature, …
- SPL Engineering
- Domain Engineering
- Application Engineering

II. Produktlinien-Engineering
- Feature-Modelle und Produktkonfiguration
- Variabilitätsmodellierung im Lösungsraum
- Programmierparadigmen für Produktlinien

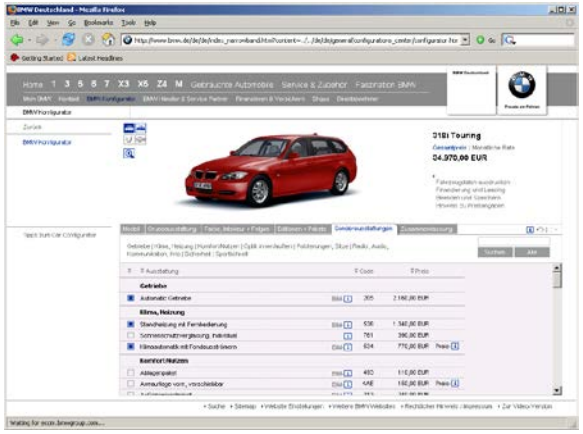
III. Produktlinien-Analyse
- Feature-Interaktion
- Testen von Produktlinien
- Verifikation von Produktlinien

IV. Fallbeispiele und aktuelle Forschungsthemen

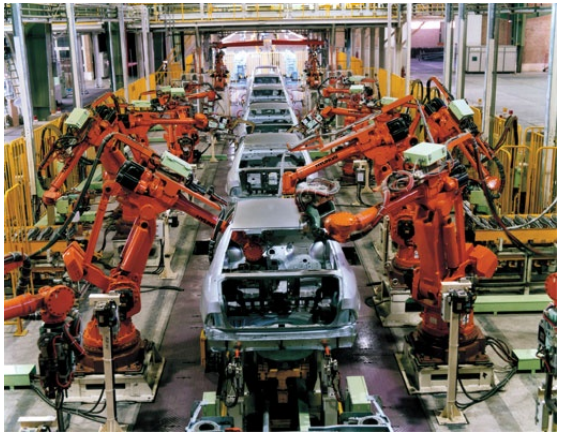
Mass Customization

*[…] producing goods [products] and services to meet **individual** customer‘s needs with near mass production **efficiency**.*

(Davis, 1987)



+



(Software) Produktlinien

*A software product line (SPL) is a set of software-intensive systems that share a common, managed set of **features** satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core **assets** in a prescribed way.*

(Software Engineering Institute
Carnegie Mellon University)

Produktlinien als Paradigma

*A software product line epitomizes **strategic, planned reuse**. More than a new technology, it is a new way of doing business. Organizations developing a portfolio of products as a software product line are experiencing order-of-magnitude improvements in cost, time to market, and productivity.*

(Software Engineering Institute
Carnegie Mellon University)

Produktlinien als Methode

*A software product line is an explicit specification of **commonality** and **variability** between [product] variants in a **family** of similar [software] products by means of **features**.*

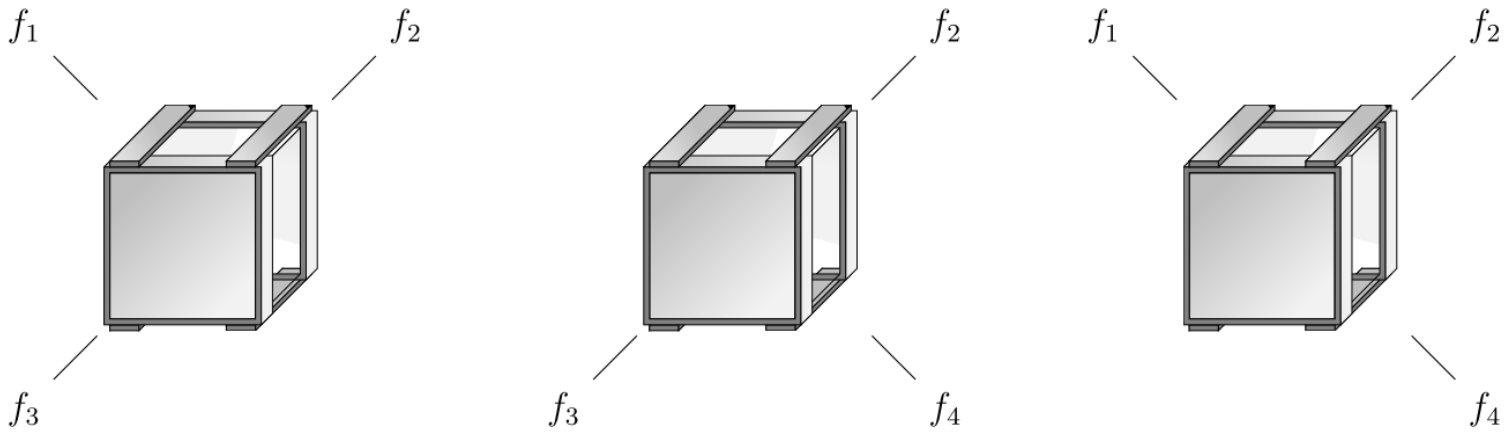
(Pohl et al., 2005)

Software-Produktlinien

Eine Menge von Programmvarianten (Software-Produkten)

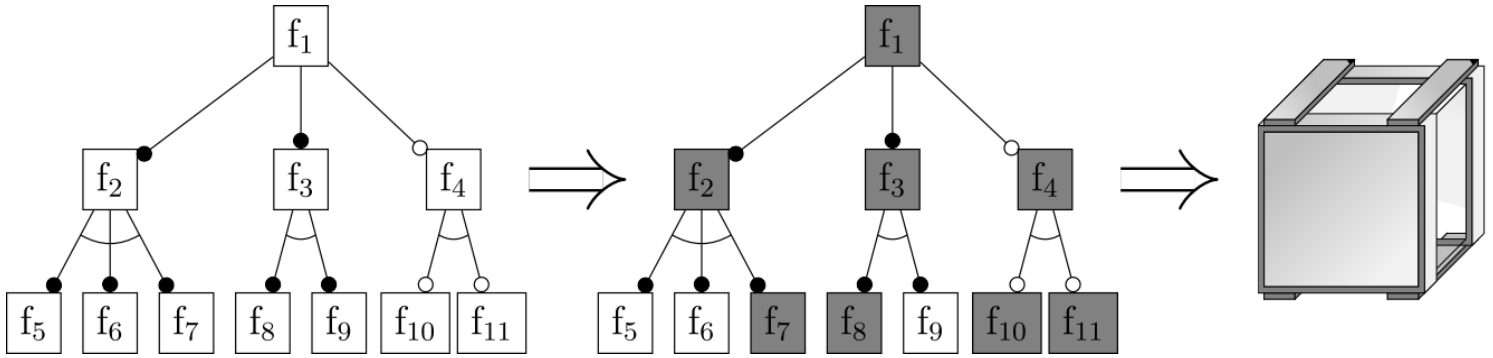
- ... die auf ein gemeinsames Marktsegment (Domäne) zugeschnitten sind
- ... auf Basis einer gemeinsamen Plattform implementiert sind mit dem Ziel der Wiederverwendung von gemeinsamen Software-Artefakten

Produktlinien vs. Produktfamilien



Produktfamilie: Menge gleichartiger Software-Produkte, die unter dem gleichen Produktnamen zusammengefasst sind.

Produktlinien vs. Produktfamilien



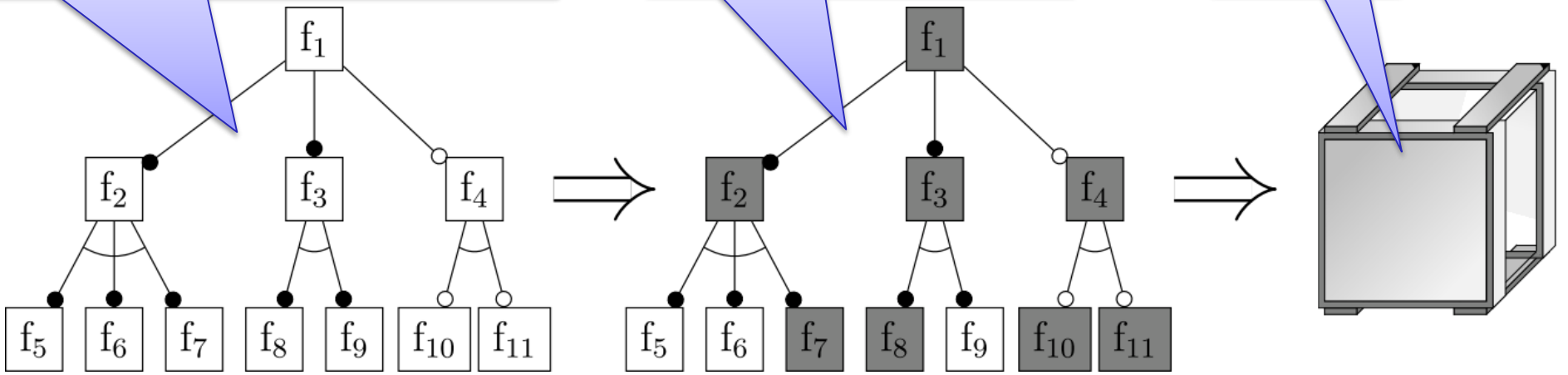
Produktlinie: Erweiterung einer Produktfamilie um eine strukturierte Beschreibung der Gemeinsamkeiten und Unterschiede zwischen den Produktvarianten anhand ihrer Produktkonfiguration.

Produkte, Konfigurationen, Varianten

Konfigurationsraum / -modell

Produktkonfiguration

Produkt



Produktlinien-Implementierung

```
          h(Object o
#if
   , T      ction txn
#endif
   ) {
      if (o==null
#ifdef SYNC
            || txn==null
#endif
            ) return;
#ifdef SYNC
      Lock l=txn.lock(o);
#endif
      elementData[size++] = o;
#ifdef SYNC
      l.unlock();
#endif
      fireStackChanged();
   }
}
```

Produktvariante

```
          ush(Object o
#ifde     NC
   , T      action txn
#endif
   ) {
      if (o==null
#ifdef SYNC
            || txn==null
#endif
            ) return;
#ifdef SYNC
      Lock l=txn.lock(o);
#endif
      elementData[size++] = o;
#ifdef SYNC
      l.unlock();
#endif
      fireStackChanged();
   }
}
```

Produktimplementierung

```
   v          (Object o) {
              =null) return;
         e    tData[size++] = o;
         fire    ackChanged();
   }
}
```

Konfigurationen, Varianten, …

Product Configuration: Collection of Product Parameter Values denoting a customized Product

Product Variant: Collection of Software Engineering Artifacts (Assets) associated with a Product Configuration

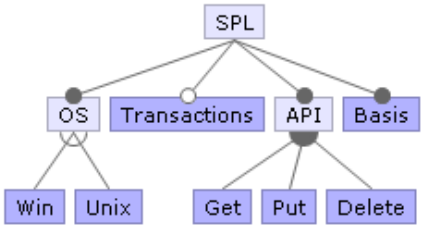
Product: Software Implementation assembled from a Product Variant that corresponds to a Product Configuration

Entwurf und Implementierung einer SPL

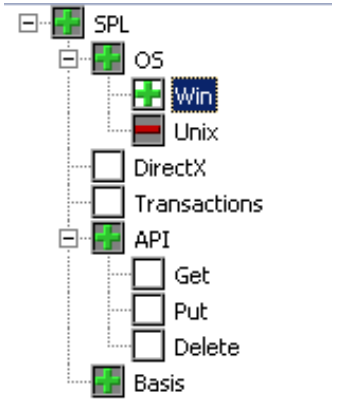
Domain Eng.

Application Eng.

Feature-Modell

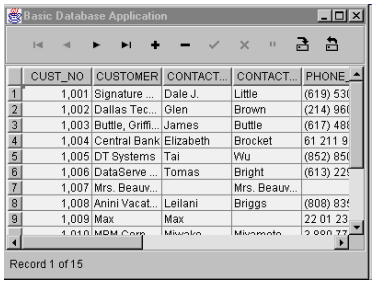


Wiederverwendbare Implementierungs-artefakte



Feature-Auswahl

Generator



Fertiges Program

Domain

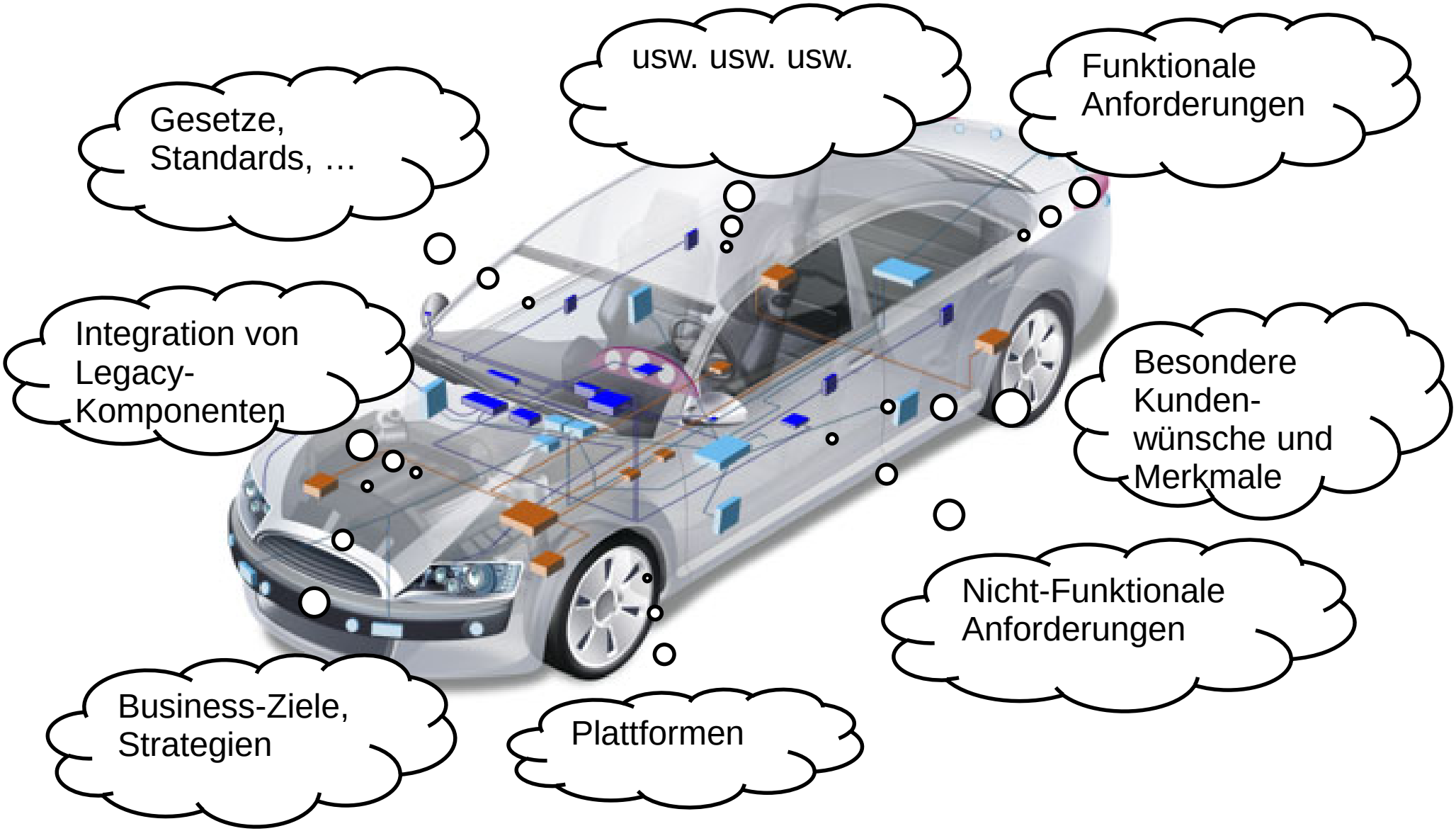
- Die Programme einer Produktlinie sind zugeschnitten auf ein **Anwendungsgebiet**

  Automotive Systems, Automation Systems, Information Systems, Mobile Devices, Operating Systems, …

- Das Anwendungsgebiet wird als **Domäne** bezeichnet

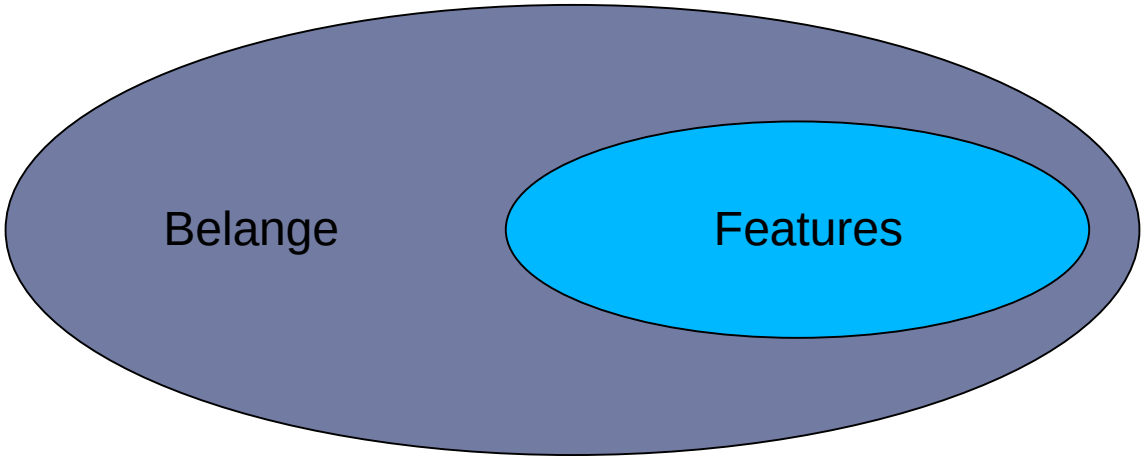
- Innerhalb der Anwendungsdomäne werden relevante **Domain-Features** aus der Menge aller Belange sämtlicher Stakeholder identifiziert

Belange und Stakeholder



Concern vs. Feature

- Concern (deutsch: Belang, siehe spätere Kapitel)
  - Jedwede Problemstellung (Einflussfaktor bei der Produktentwicklung), die von Interesse ist
- Feature
  - Problemstellung, die eine besondere Bedeutung in einer Domäne hat
  - Konfigurationsoption für den Kunden



Was ist ein Feature?

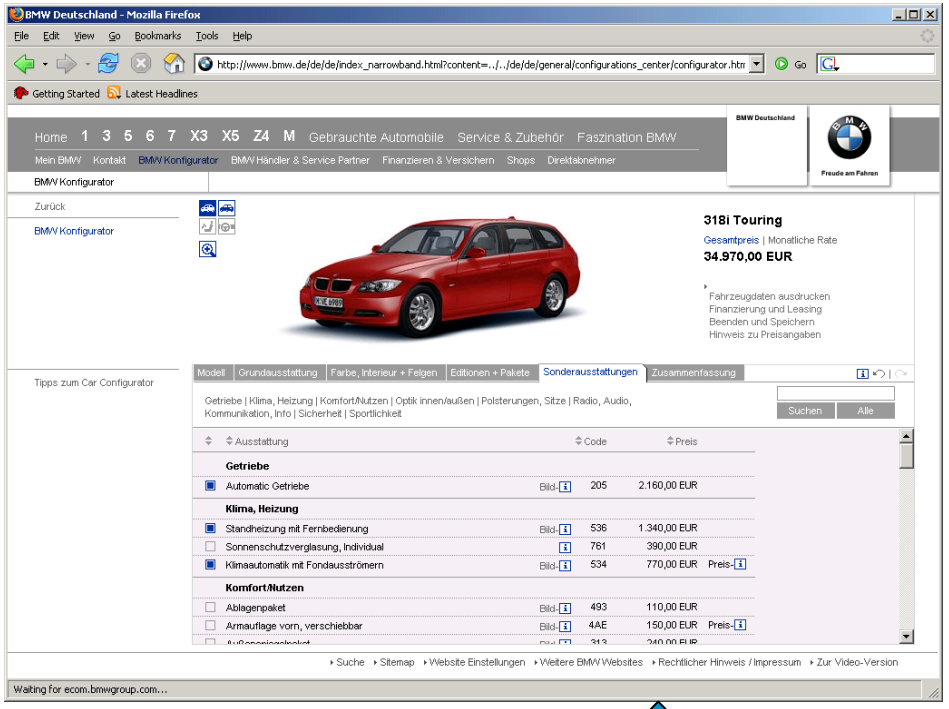
- Features repräsentieren die für die Produktdiversifizierung relevanten Anforderungen, Gemeinsamkeiten / Unterschiede von Produkten einer Domäne

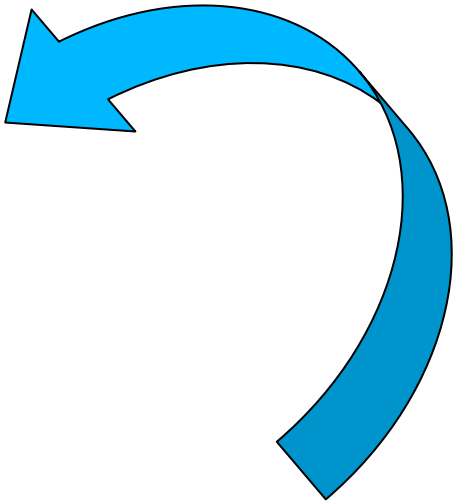
  => Es gibt auch Belange, die keine Features sind

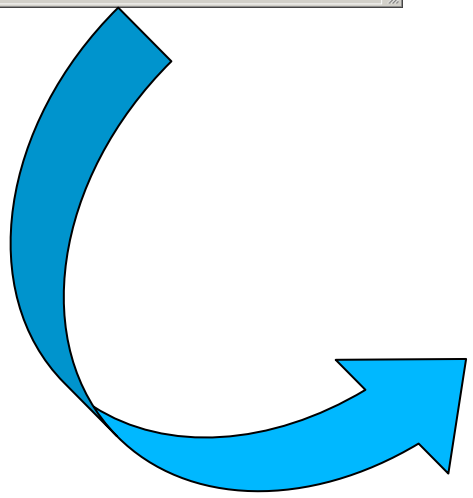
- Ein Feature ist eine Domänen-Abstraktion und dient als Mittel zur Kommunikation zwischen Stakeholdern

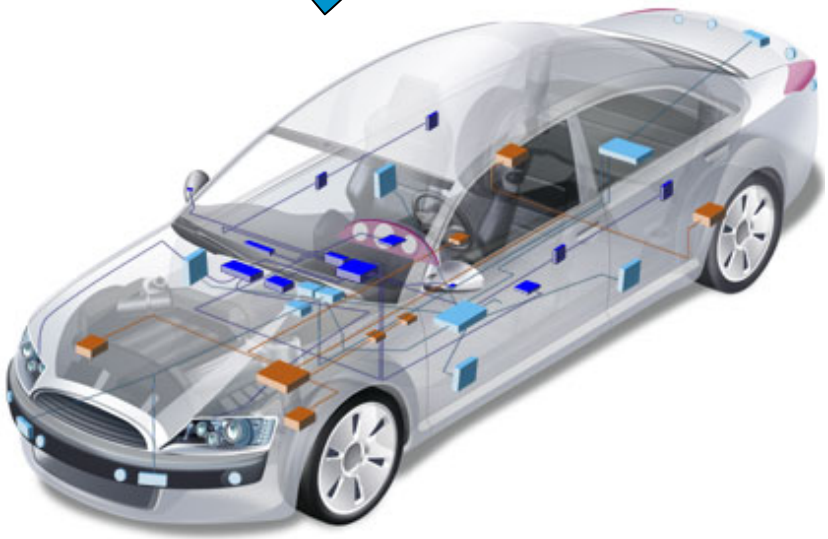
  => Features sind das Ergebnis einer Domänen-Analyse

Beispiel: Automotive Domain









Beispiel: Features in Databases

- Transaktionsverwaltung
- Log & Recovery
- Schreibzugriff
- Persistenz / In-Memory
- Seitenverdrängungsstrategien LRU / LFU / Clock /...
- Sortierverfahren
- Datentypen variabler Länge
- Gruppieren, Aggregation
- Windows / Unix / NutOS / TinyOS / …

SPLE

- Ziel: Entwicklung einer Produktfamilie als Produktlinie statt als einzelne Produkte
- Produktlinie deckt Anforderungen (Features) der ganzen Domäne ab
- Abweichung vom klassischen Entwicklungsprozess und Lebenszyklus, Unterscheidung in
  - Domain Engineering
  - Application Engineering

Feature-Oriented SPLE

Idee: Feature als ganzheitliches Abstraktionskonzept

- Feature als **Domänen-Abstraktion** im Problemraum der Anwendungsdomäne

  => Spezifikation von Produktkonfigurationen

  (Feature-Auswahl als Eingabe für die Produktkonfiguration)

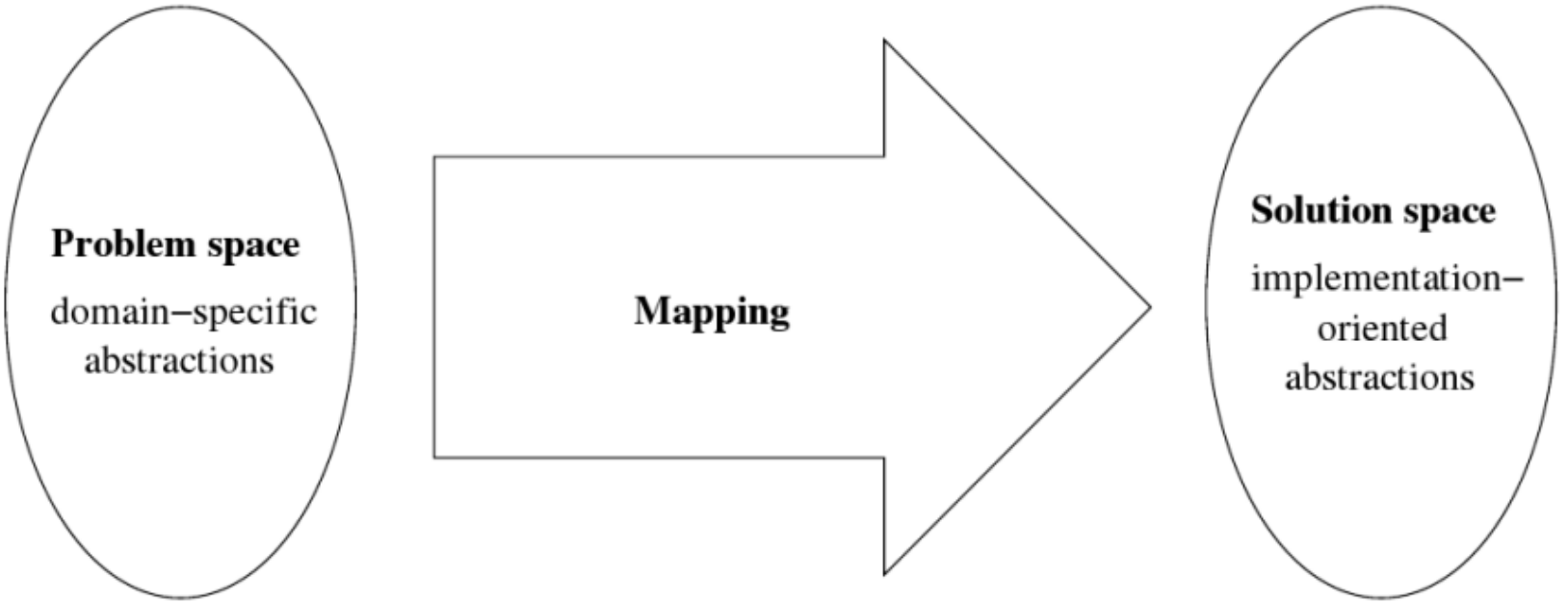
- Feature als **Implementierungskonzept** im Lösungsraum der Produktlinie

  => Spezifikation von Produktvarianten

  (Feature-Auswahl als Eingabe für die Programmgenerierung)

Feature-Oriented Software Development (FOSD)

Problem Space vs. Solution Space



(Apel and Kästner, 2009)

A Feature is…

[…] a prominent or distinctive user-visible aspect, quality, or, characteristic of a software system or systems
(Kang, 1990)

[…] a distinctively identifiable functional abstraction that must be implemented, tested, delivered, and maintained
(Kang, 1998)

[…] a distinguishable characteristic of a concept (e.g., system, component, and so on) that is relevant to some stakeholder of the concept
(Czarnecki and Eisenecker, 2000)

…and it is…

[…] a logical unit of behaviour specified by a set of functional and non-functional requirements

(Bosh, 2000)

[…] a product characteristic from user or customer views, which essentially consists of a cohesive set of individual requirements

(Chen et al., 2005)

[…] a product characteristic that is used in distinguishing programs within a family of related programs

(Batory, 2004)

… and…

[…] a triple f = (R, W, S), where R represents the requirements the feature satisfies, W the assumptions the feature takes about its environment and S its specification

(Classen et al., 2008)

[…] an optional or incremental unit of functionality

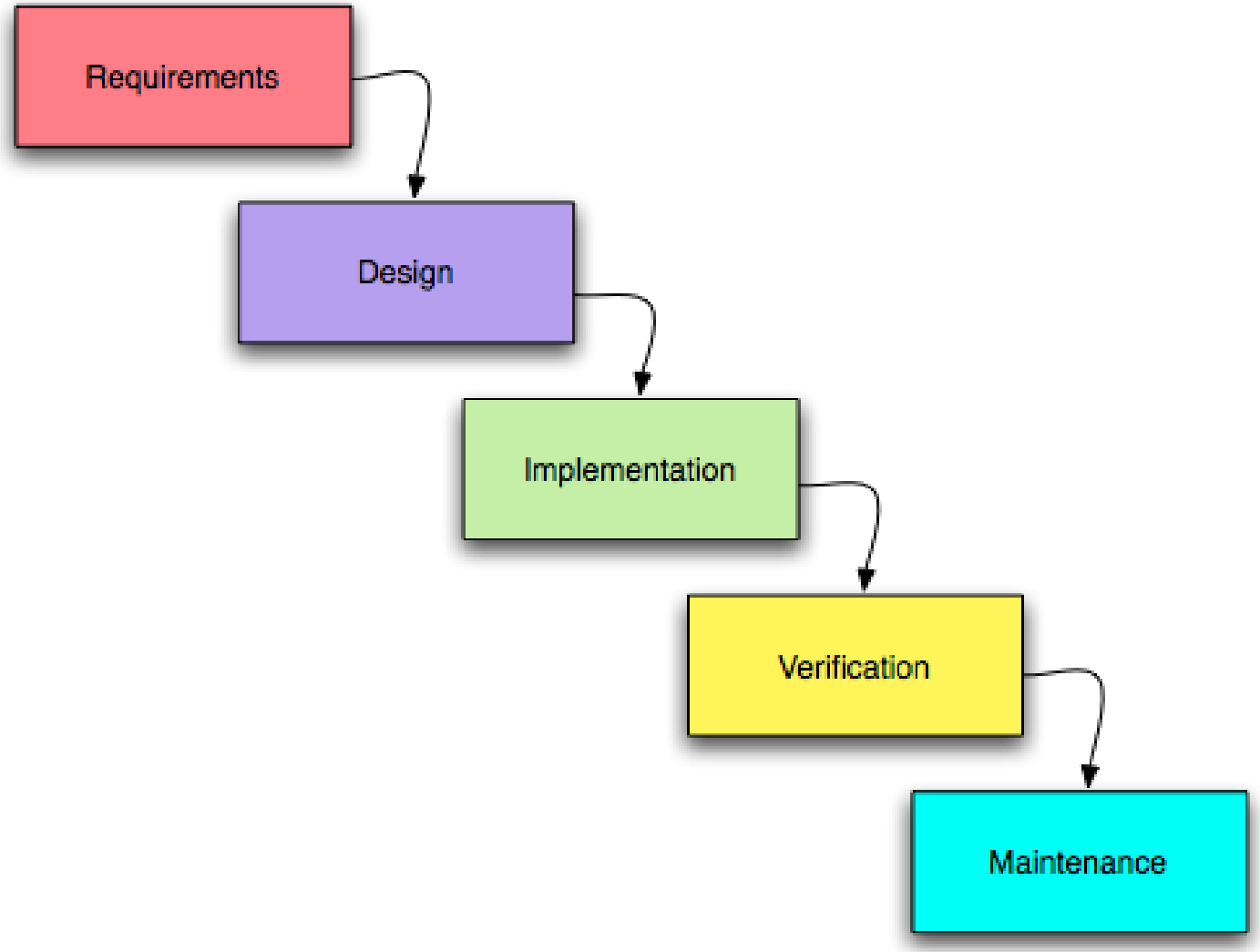
(Zave, 2003)

[…] an increment of program functionality

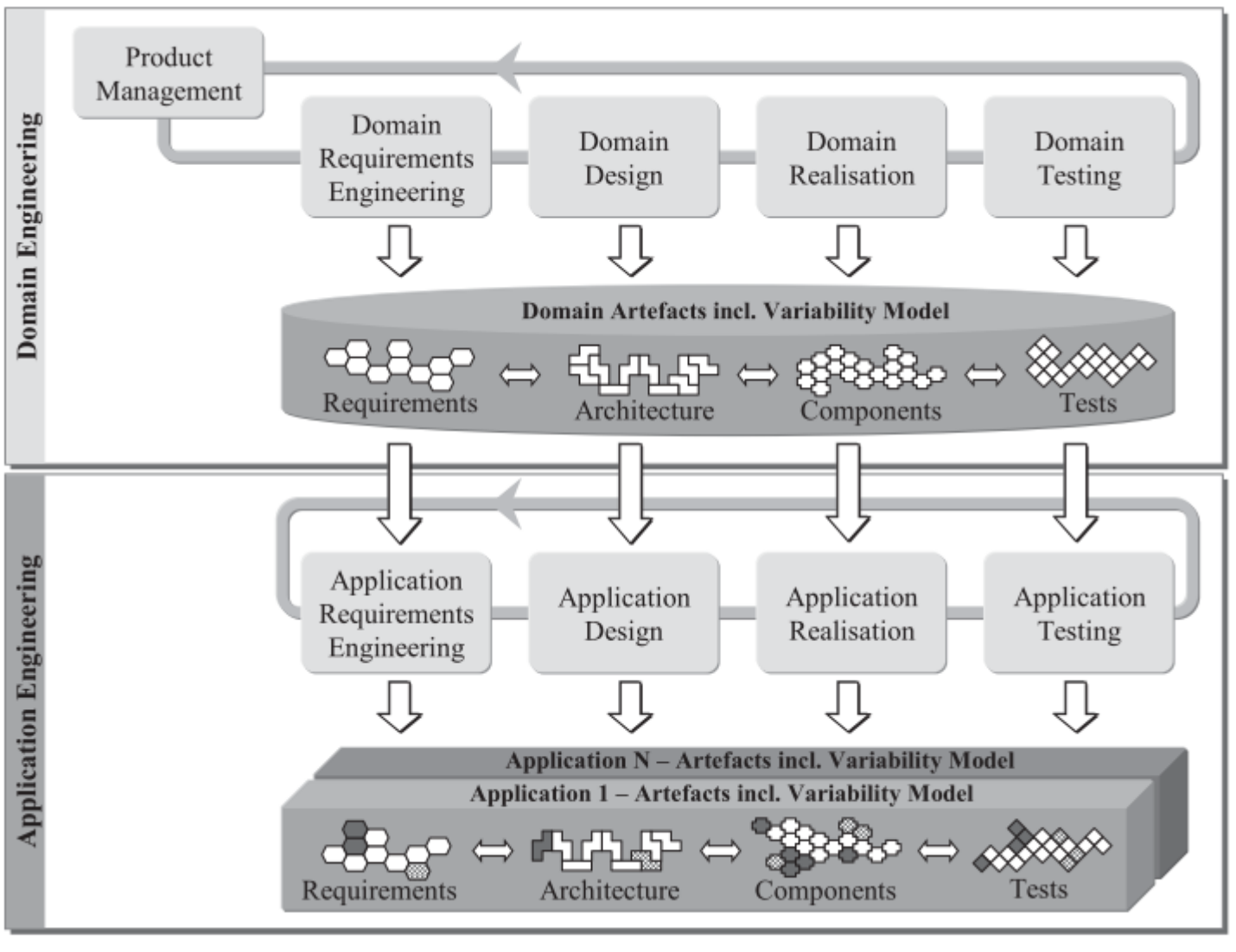
(Batory, 2005)

… to be completed

Software Lebenszyklus – Klassisch



Software Product Line Engineering



Design for Reuse

Design with Reuse

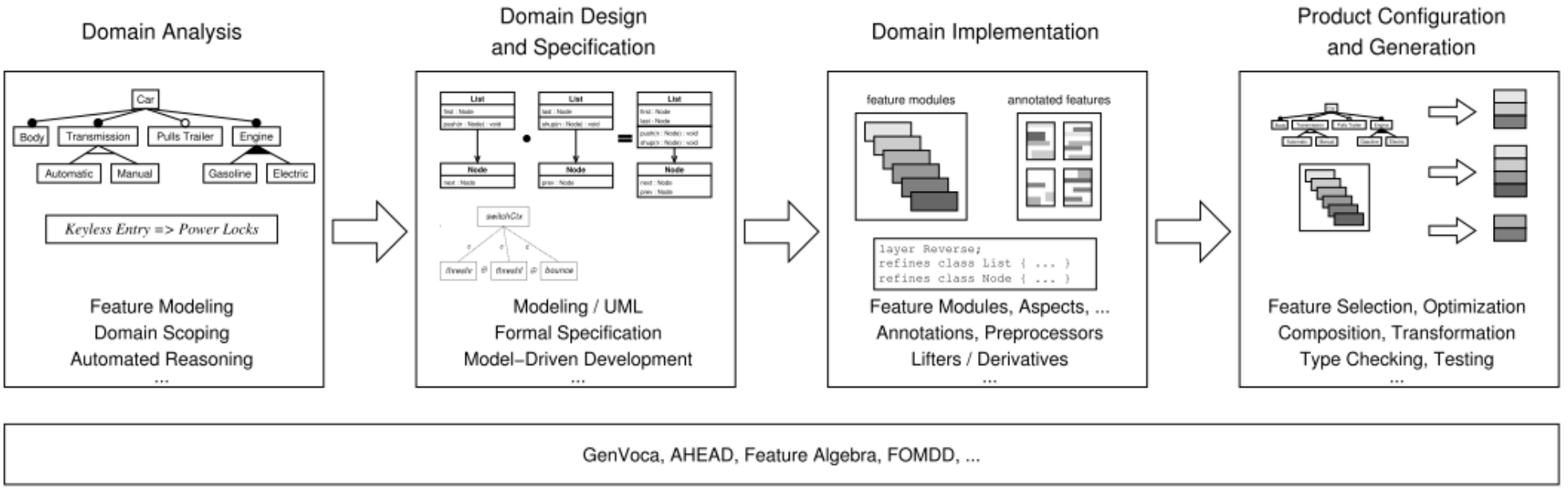
(Pohl et al. 2005)

Domain Engineering

[...] is the activity of collecting, organizing, and storing past experience in building systems [...] in a particular domain in the form of reusable assets [...], as well as providing an adequate means for reusing these assets (i.e., retrieval, qualification, dissemination, adaptation, assembly, and so on) when building new systems.

(K. Czarnecki and U. Eisenecker)

FOSD

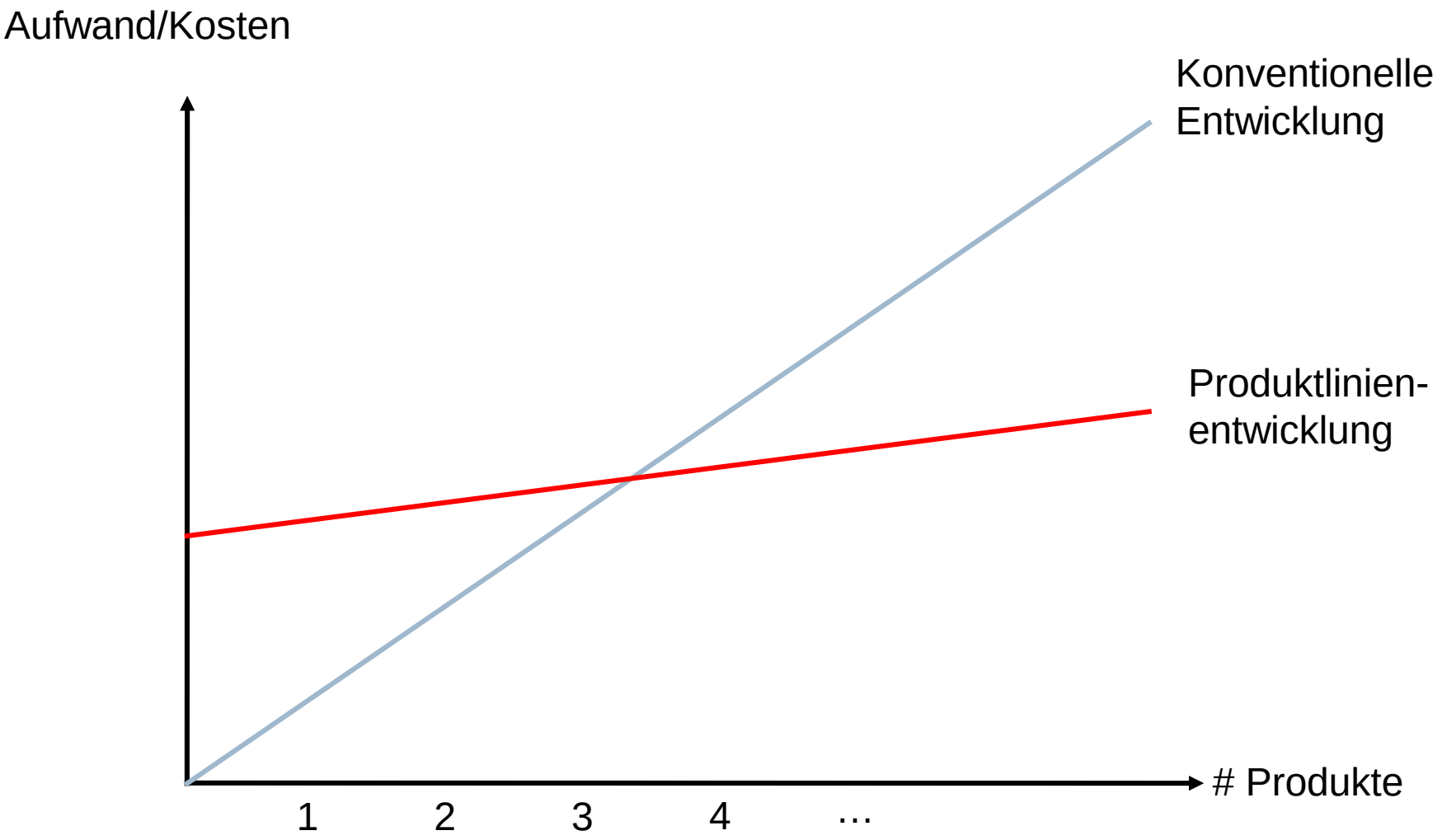


Herausforderungen:

- Features sind crosscutting concerns
- Features interagieren

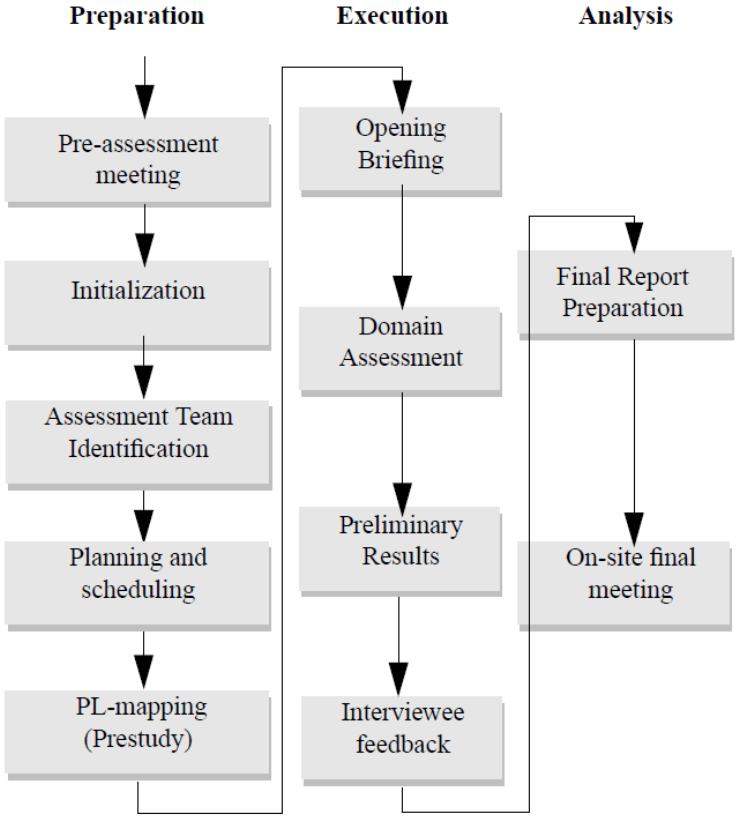
(Apel and Kästner, 2009)

Development Effort



Scoping

- Eingrenzung der Domäne
- Welche Features sind relevant/sollen entwickelt werden
- Oft wirtschaftliche Entscheidung



[Schmid 2002]

			exist.	planned		potent.
			P1	P2	P3	P4
Domain 1	Sub-Domain 1.1	Feature 1.1.1	X	X	X	X
		Feature 1.1.2	—	X	X	X
		Feature 1.1.3	X	X	—	X
	...	...	...	...	...	...
	Sub-Domain 1.n	Feature 1.n.1	X	—	X	X
		...	...	...	...	...
Domain 2	Sub-Domain 2.1	Feature 2.1.1	—	X	X	—
	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	Feature m.1.1	—	X	—	X

Ansätze zur Einführung von Produktlinien

- Es gibt drei gängige Ansätze eine Software-Produktlinie zu erstellen / einzuführen
  - Proaktives Vorgehensmodell
  - Reaktives Vorgehensmodell
  - Extraktives Vorgehensmodell

- Für alle Implementierungsformen

- Auswahl anhand betrieblicher Gesichtspunkte (Kosten, Risiko, Chancen, ...)

Proaktiv vs. Reaktiv vs. Extraktiv

SPL neu entwerfen/implementieren

Später evtl. Umstrukturierungen nötig

Hohe Kosten/Risiko

Gut für schnellen Wechsel zu SPL

Analog Spiralmodell/XP

Eine/mehrere Anwendunge(n) als Basis

Zerlegung einer monolithischen Anwendung

Unterbrechung der normalen Entwicklung

Inkrementelles Zufügen von Varianten

Komplette Domänenanalyse

geringer Ressourcenaufwand/Risiko

geringere initiale Kosten/schnelle Ergebnisse

Referenzen

- *Stanley M. Davis:* **Future Perfect.** Addison-Wesley, Reading, 1987.
- *Paul Clements:* **Software Product Lines: A New Paradigm for the New Century.** In Crosstalk: The Journal of Defense Software Engineering, pages 21–23, 1999.
- *Carnegie Mellon University (CMU), Software Engineering Institute:* **Software Product Lines – Overview.** Website URL https://www.sei.cmu.edu/productlines/, last visited 2014.
- *Paul Clements and Linda Northrop:* **Software Product Lines: Practices and Patterns.** Addison-Wesley Longman Publishing Co., Inc., 2001.
- *Klaus Pohl, Günter Böckle, and Frank van der Linden:* **Software Product Line Engineering: Foundations, Principles and Techniques.** Springer, 2005.
- *Sven Apel and Christian Kästner:* **An Overview of Feature-Oriented Software Development.** Journal of Object Technology, 8(5):49–84, 2009.