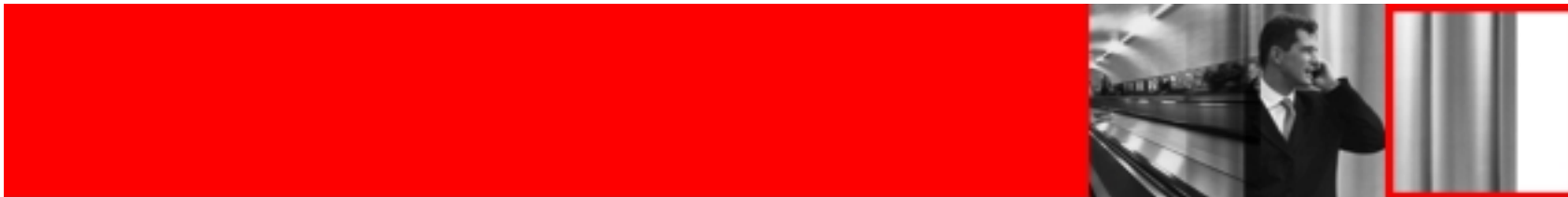


Softwarefamilien und Produktlinien

- systematische Wiederverwendung -



Matthias Clauß
Intershop Research & TU Dresden

Einleitung

Was würden Sie machen, wenn Ihr Auftraggeber oder Chef fordert:

**“Reduzieren Sie ihre Entwicklungszeiten!
...?”**

Nein, es gibt keine Budgeterhöhung.”

Motivation zur Wiederverwendung

- **Produktivität**

- Reduzierung von Entwicklungszeit und Kosten

- **Qualität:**

- bewährte Software-Bausteine

- **Wartung von Altsystemen**

- Lebensdauer (oft 20 Jahre und mehr)
- beständig Anpassung an neue Anforderungen
- Bindung von Entwicklungskapazität
- Softwaresysteme von heute sind die Altlasten von morgen

Objektorientierung als Lösung?

Theorie: Wiederverwendung soll Produktivität und Qualität steigern

Praxis:

- **Klassen als Wiederverwendungseinheiten sind zu klein**
- **Frameworks sind größer, ihre Zusammenarbeit untereinander ist oft mangelhaft; häufig mangelnde Performanz, schlechte Dokumentation und schlechte Verständlichkeit**
- **Entwurfsmuster = wiederverwendbares Wissen, nicht als Code; kombinierter Einsatz verschlechtert Performanz und Verständlichkeit**
- **Ad hoc-Wiederverwendung: keine systematische Entwicklung für und mit Wiederverwendung**

Ursache & Lösung

Eine Ursache:

Die vorherrschenden Verfahren und ihre Werkzeuge sind auf die Entwicklung *einzelner Systeme* ausgerichtet

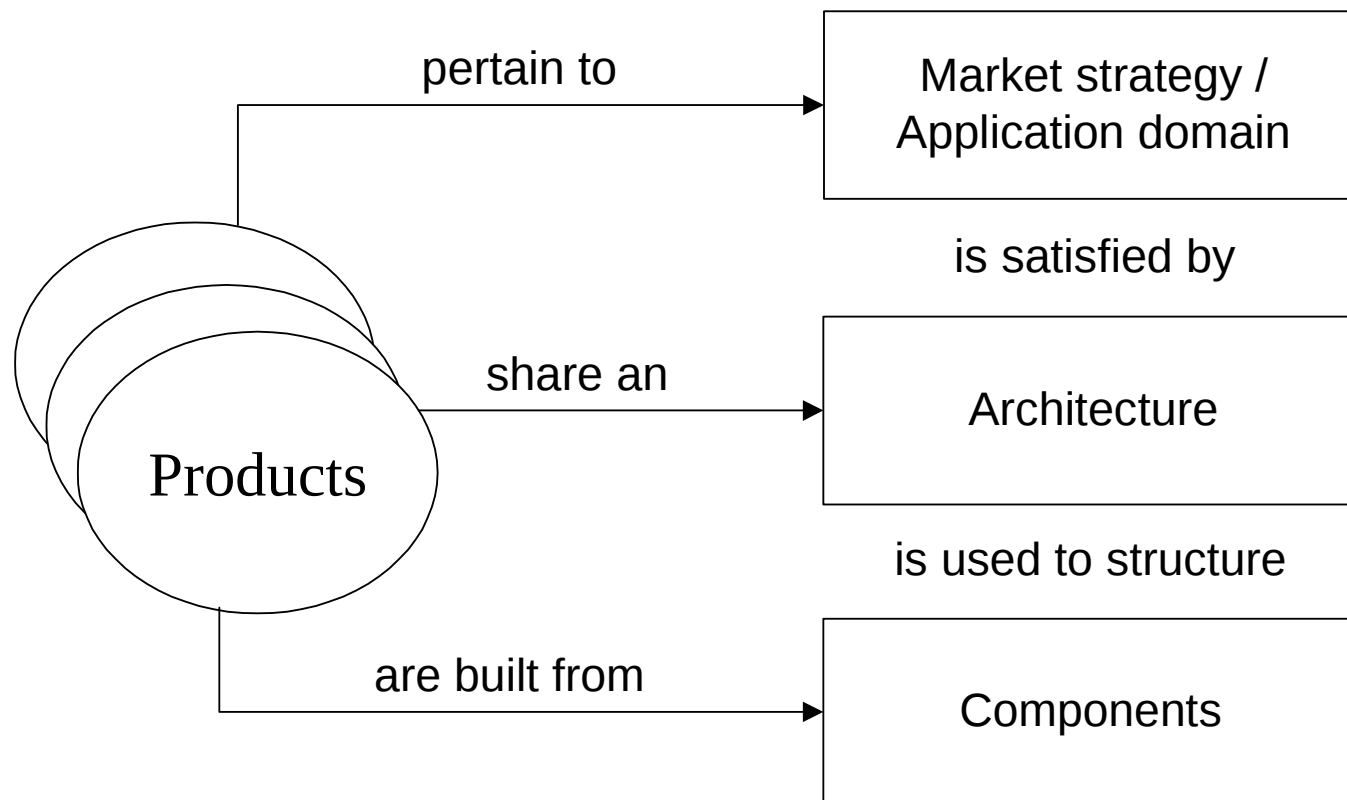
Modularisierung:

- Einzelne Bausteine können leichter gewartet, getauscht und wiederverwendet werden**

Doch nach welchen Kriterien erfolgt die Aufteilung?

Definition: Produktlinien

„A product line is a group of products sharing a common, managed set of features that satisfy specific needs of a selected market or mission.“ [Withey96]



Definition: Produktfamilien & Domänen

„A product family is a group of products that can be built from a common set of assets.“ [Czarnecki00, Withey96]

- **technische Sicht**
- **Entwicklung mit Gemeinsamkeiten & Unterschieden**
- **Core Assets $\Rightarrow$ Plattform**

Domäne = Bereich von Fachwissen, der [Czarnecki00]

1. für spezifische Anforderungen ausgewählt wird,
2. Konzepte und Begriffe beinhaltet, die durch Anwender des Fachbereiches verstanden wird und
3. das Wissen einschließt, wie Softwaresysteme (oder Teile davon) für diesen Fachbereich zu bauen sind.

Beispiele

Produktlinie:

- **verschiedene Modelle / Ausstattungsvarianten eines Automobiles**
 - Käufergruppen
 - Preis-Leistungs-Verhältnis
- **aus Kundensicht, Bestimmung der Funktionalität**

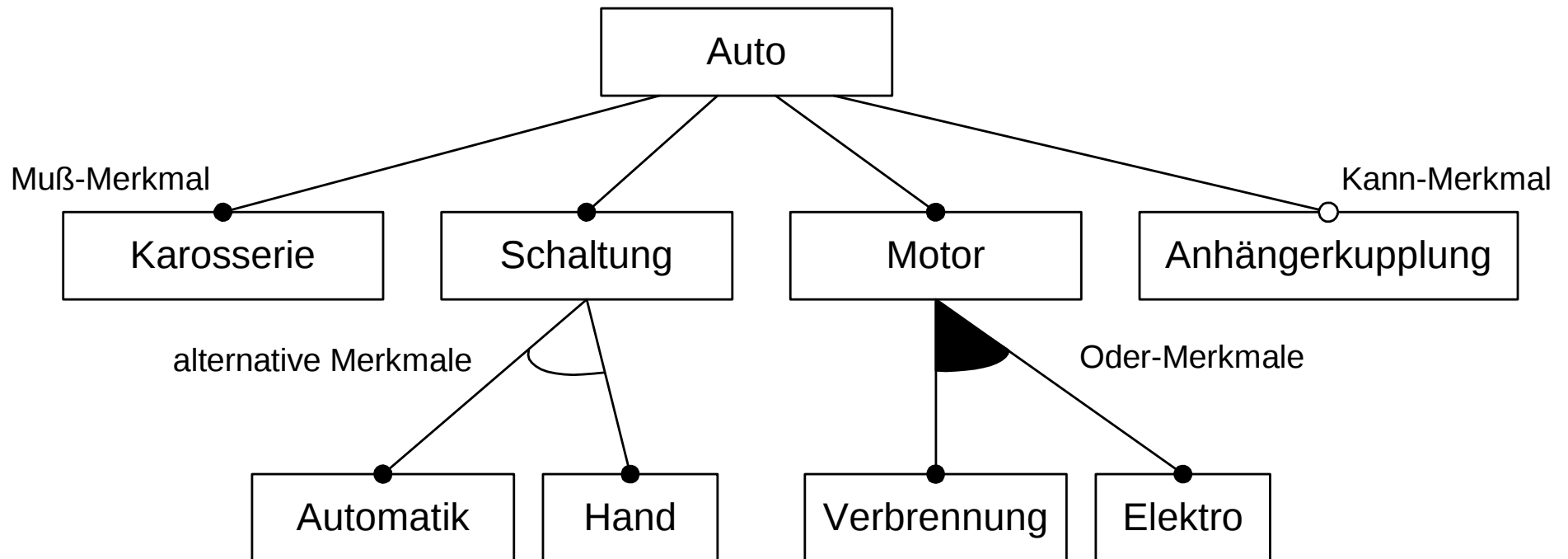
Produktfamilie:

- **Fahrzeugtyp; Antrieb; gemeinsame Karosserie**

Domäne:

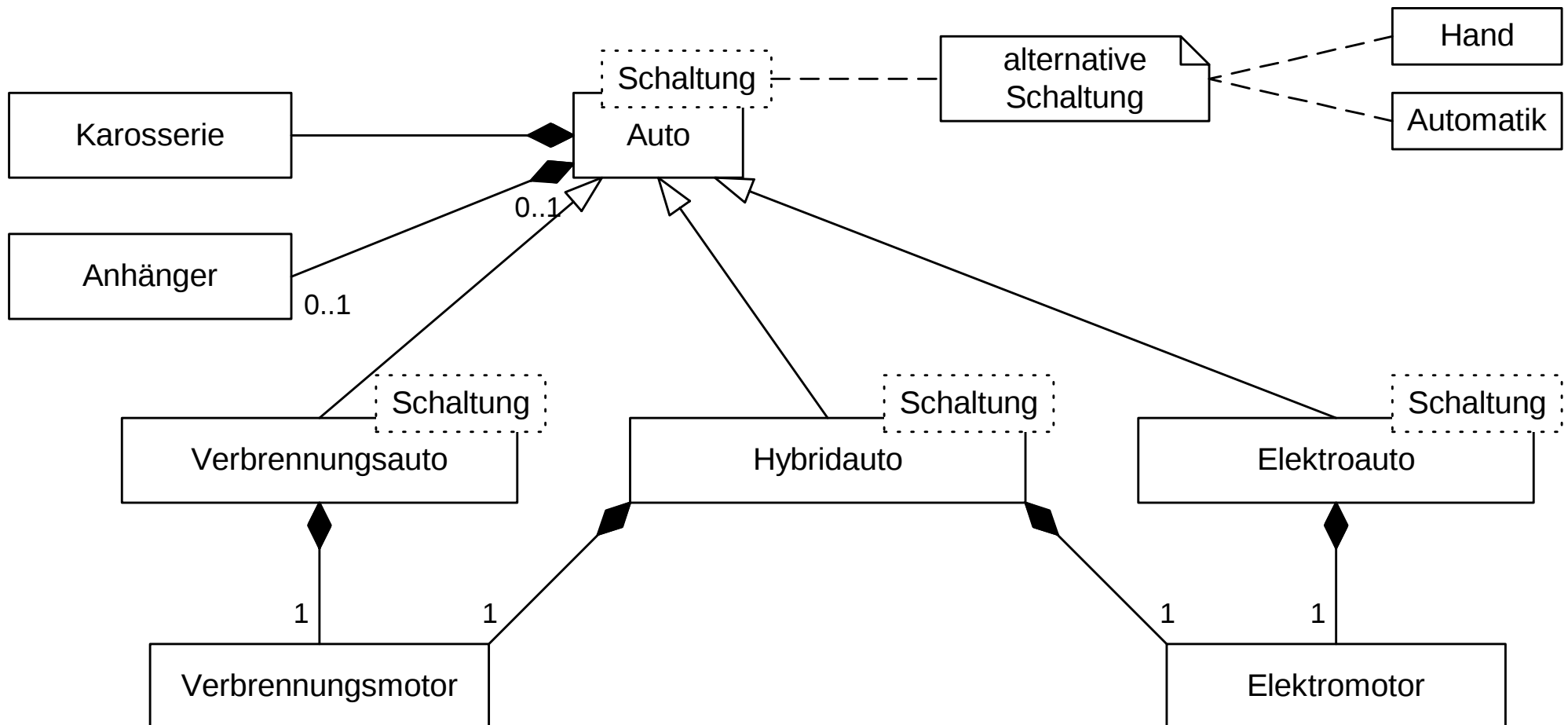
- **Dateiverwaltung, Sicherheit, DBMS, Transaktionen, GUI**

Merkmalsdiagramm



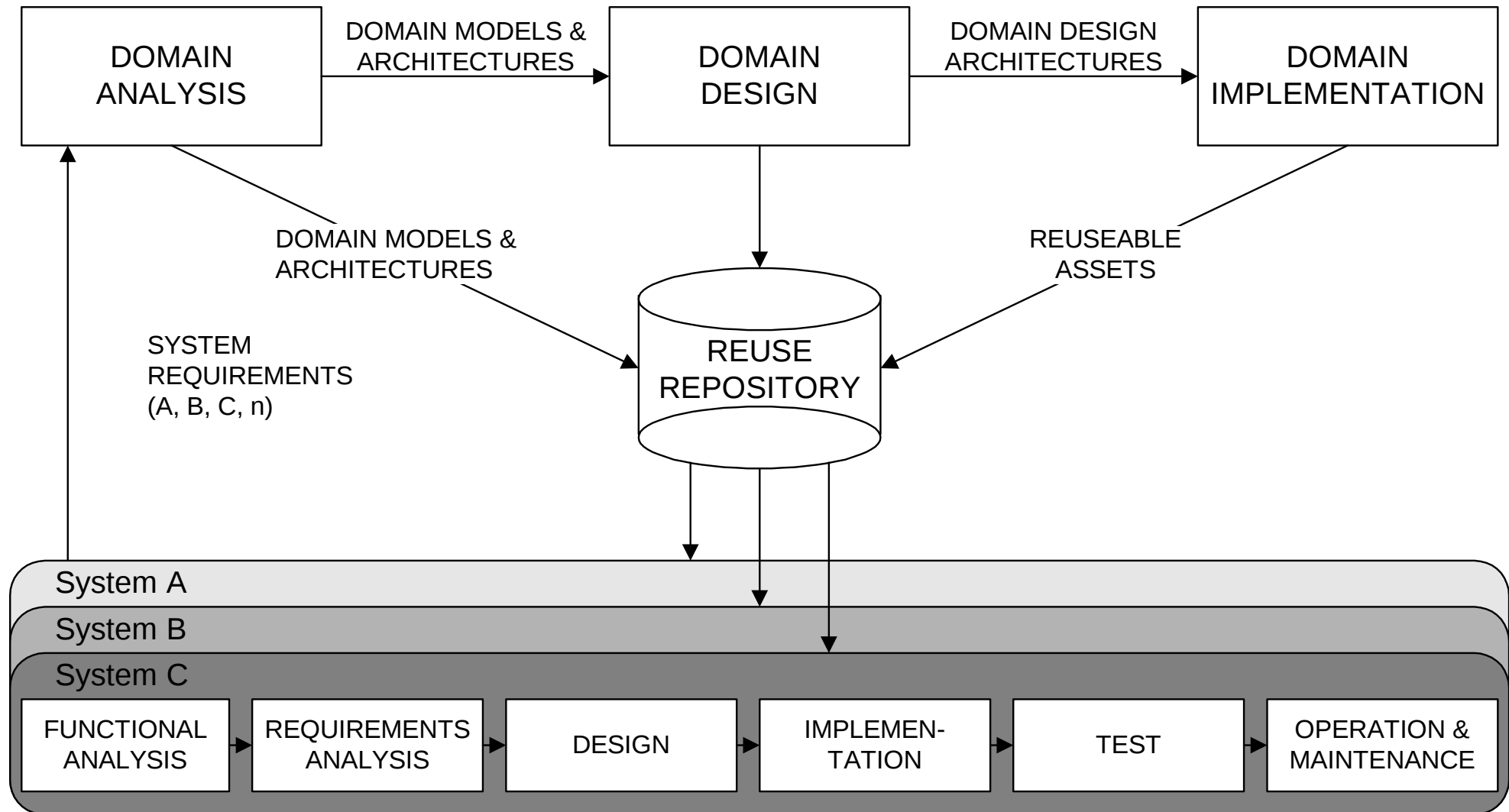
Keine Entscheidung zur Implementierung der Variabilität!

Vergleich: UML-Diagramm

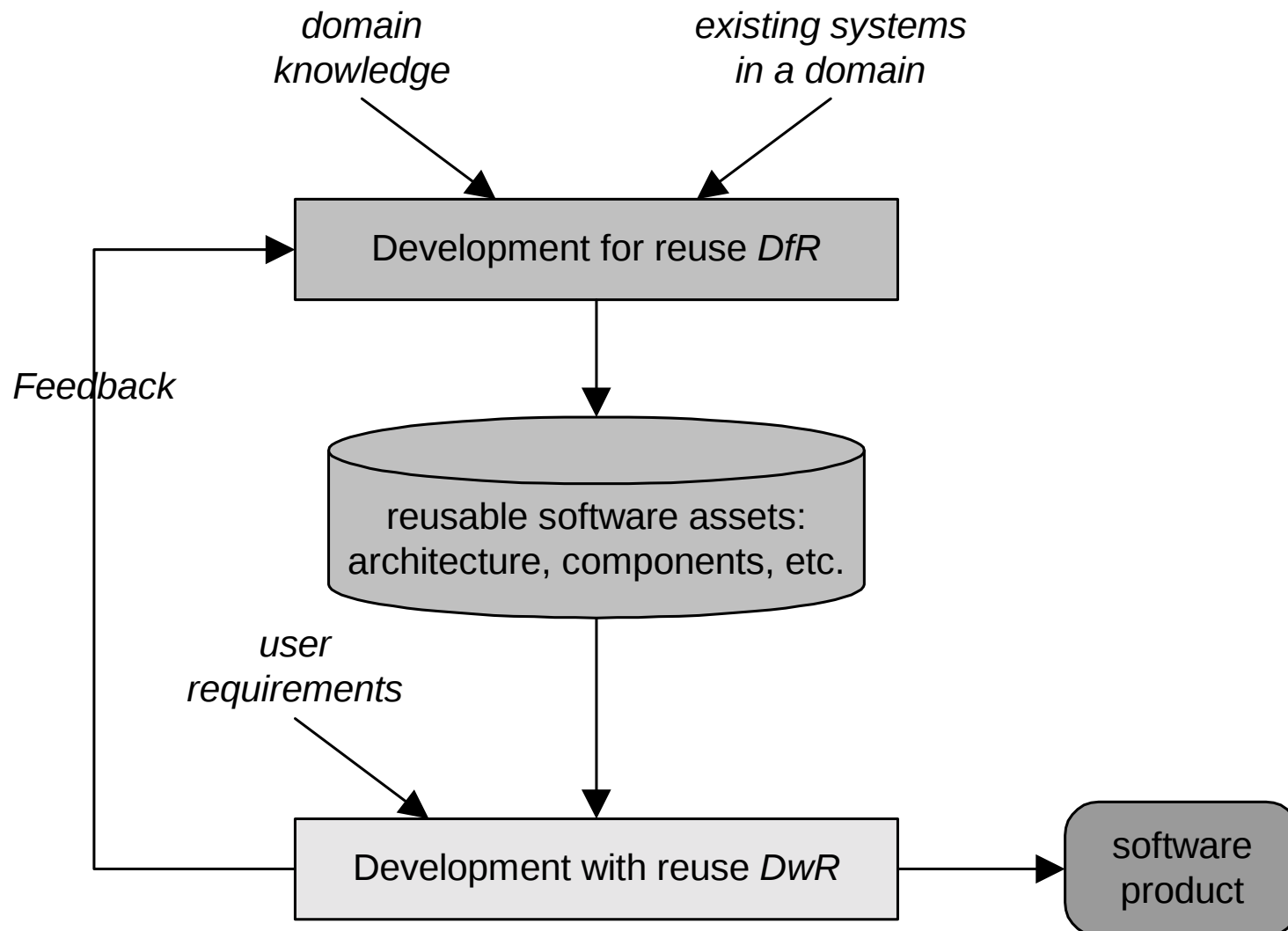


Variabilität konkret implementiert: Vererbung, Assoziation, Templates!

SE mit Domänenentwicklung



Prozeß-Übersicht



Ansätze zur Entwicklung mit Reuse

FODA (1990) - SEI / CMU

- Domänenanalyse, Merkmalsmodellierung

RSEB (1997) - Jacobson, Griss, Jonnson

- Einführung von Variationspunkten

FeatuRSEB (1998) - Griss

- Kombination von OO und Merkmalsmodellierung

SPLIT (2000) - Alcatel, Thomson

- verfeinerte Variationspunkte

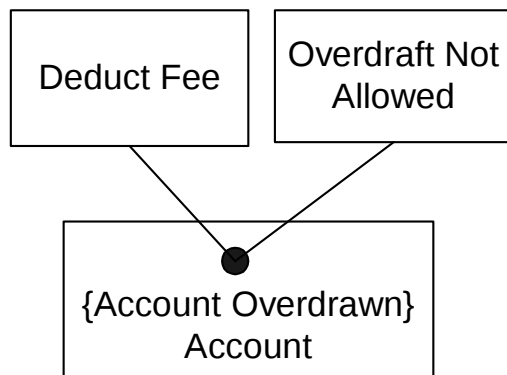
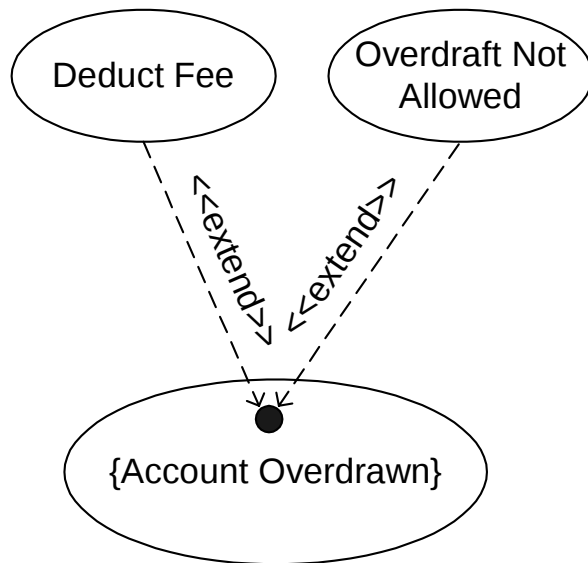
Generative Programming (2000) - Czarnecki, Eisenecker

- Domänenentwicklung und generatives SE

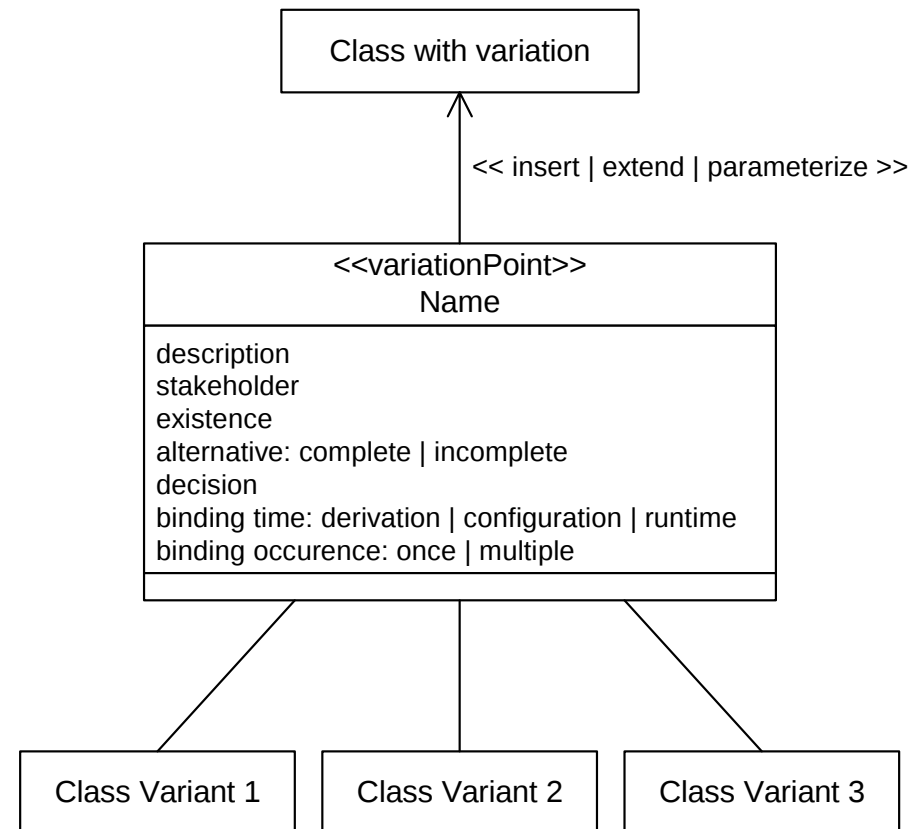
PuLSE / Kobra (1999 / 2000) - IESE / Fraunhofer

- komplexes Vorgehensmodell für PLSE
- komponentenbasiertes SE, kombinierbar mit Produktlinien

Variationspunkte



In RSEB



Nach SPLIT

Das war's für den ersten Teil.

Wo sind Ihre Fragen ;-)

Literaturquellen:

- Generative Programming -- Methods, Tools and Applications; Czarnecki, Eisenecker; 2000
- <http://www.bell-labs.com/user/cope/Mpd/IeeeNov1998/>
- komplette Literaturliste unter <http://www-st.inf.tu-dresden.de/~mc3/varUML>