

# 28 Aspektorientierte Entwurfsmethoden (Aspektorientierte Zerlegung)

Prof. Dr. U. Aßmann

Technische Universität Dresden

Institut für Software- und  
Multimediatechnik

<http://st.inf.tu-dresden.de>

Version 11-0.2, 28.12.11

- 1) Überblick
- 2) Essentielle Zerlegung in der SA



# Obligatorische Literatur

- ▶ View models (Wikipedia)

# Andere Literatur

- ▶ De Marco, T.: Structured Analysis and System Specification; Yourdon Inc. 1978/1979. Siehe auch Vorlesung ST-2
- ▶ McMenamin, S., Palmer, J.: Strukturierte Systemanalyse; Hanser Verlag 1988

# Ziel

- ▶ Lerne die verschiedenen aspektorientierten Zerlegungsstrategien kennen

# 28.1 Aspektorientierte Zerlegungen



# Entwicklung mit Zerlegungsstrategien

- ▶ Bisher waren die Zerlegungsstrategien der Entwurfsmethoden **uniform**, d.h. Es wurde nach einem einzigen Kriterium zerlegt
- ▶ Ausgehend vom System als Ganzes wird schrittweise top-down-verfeinert:
  - Funktionsorientierte Zerlegung
  - Aktionsorientierte Zerlegung
  - ECA-basierte Zerlegung
  - Datenorientierte Zerlegung
  - Objekt-orientierte Zerlegung
  - Komponentenorientierte Zerlegung
  - Transformative Zerlegung

# Perspektivenmodelle, Viewpoints and Views

- ▶ Ein **Artefaktraum** besteht aus einer Menge von Modell- und Programmfragmenten, -komponenten
- ▶ Ein **Belangraum** besteht aus einer Menge von Belangen (Aspekten, aspects, concerns) und einer algebraischen Struktur, die sie gliedert
- ▶ Ein **Sichtenraum** besteht aus einem gekoppelten Belang- und Artefaktraum
- ▶ **Perspektivenmodelle (Viewpoint models, view models)** sind systematische Modelle für aspektorientierte Zerlegungen
- ▶ Ein **Viewpoint (Perspektive)** besteht aus einer Menge von Aspekten (Belangen, concerns)
  - RM-ODP
  - Zachmann framework
  - MDA

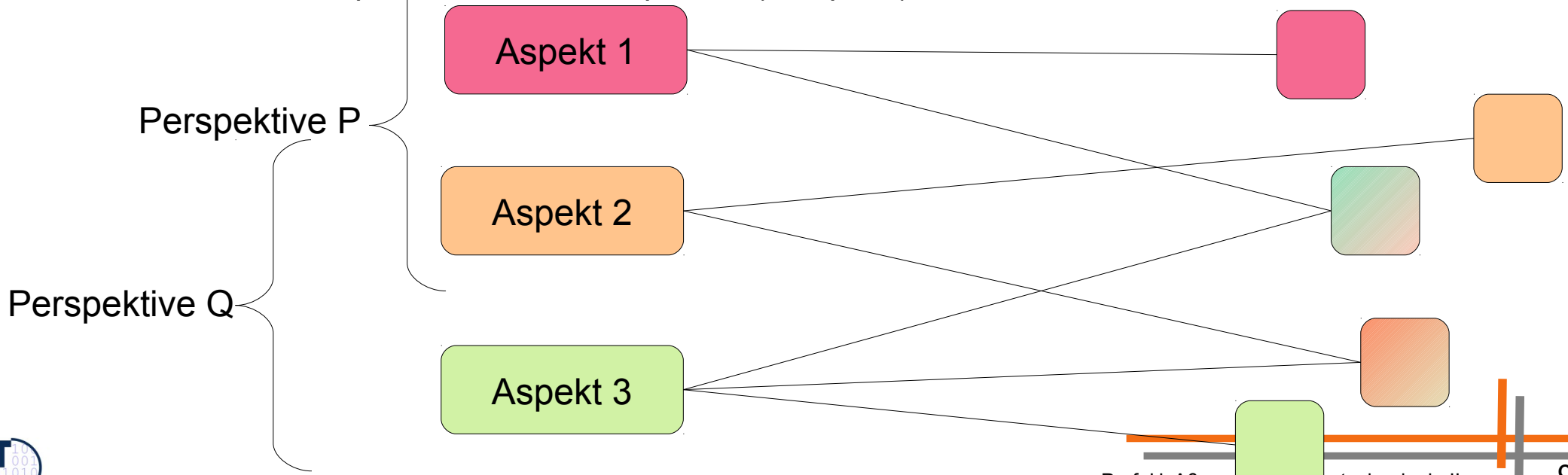
# Aspektorientierte Zerlegungsstrategien

- ▶ In der Literatur sind verschiedene **aspektorientierte** oder **mehrdimensionale Zerlegungsstrategien** bekannt:
- ▶ **Dokument-Zerlegung**
  - Struktur (html, XML)
  - Layout (CSS, FO)
- ▶ **Essentielle Zerlegung** (EAI-Zerlegung) unterscheidet folgenden Aspekte bzw Gesichtspunkte (viewpoints):
  - Essentielle Aktivitäten, Datenstrukturen und ihre Speicher
  - Administrative Aktivitäten (zur Prüfung von Daten)
  - Infrastruktur-Aktivitäten (zur Kommunikation und Anpassung an Plattform)
- ▶ **Architektonische Zerlegung** zerlegt in die Aspekte und Gesichtspunkte:
  - Architektur
  - Anwendung
- ▶ **Blutgruppen-Zerlegung** (Siedersleben)
  - Anwendung
  - Technik
- ▶ **Plattformorientierte Zerlegung** zerlegt in die Aspekte:
  - Essentielle Aktivitäten
  - Plattform 1
  - .. Plattform n



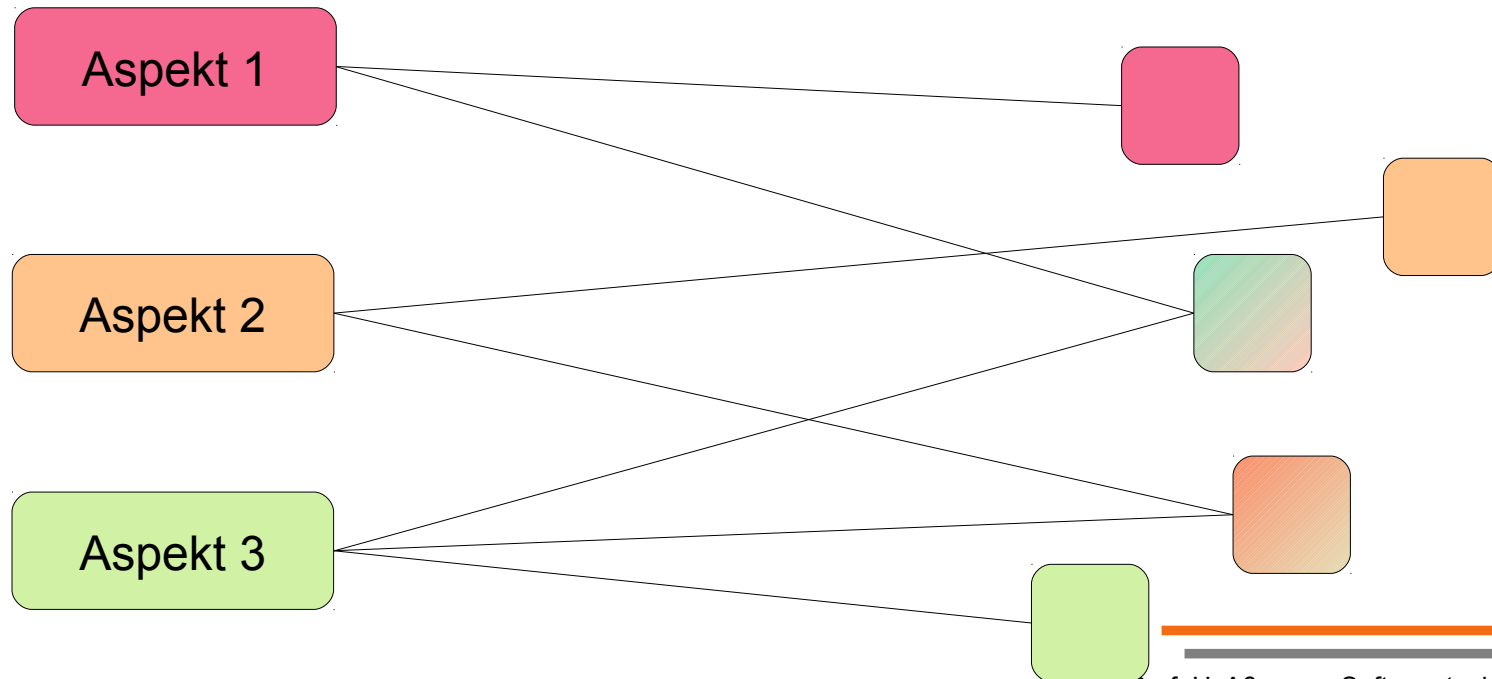
# Aspekte und Blutgruppen

- ▶ Man kann die Aspekte als Farben oder Blutgruppen von Lösungseinheiten (Modulen, Modellen, Modellelementen, -fragmenten) sehen
- ▶ Dann ist die Software gegliedert in einen **Sichtenbildungsraum (SoC space)**
  - Aspektraum (Belangraum, concern space) mit Belangen (Farben), die in Perspektiven (viewpoints) gegliedert sein können
  - Komponentenraum (Lösungsraum) enthält die Komponenten des Softwaresystems
  - Abbildung dazwischen (Concern mapping)
  - Die Abbildung definiert views (Sichten, slices, Scheiben, Schicht), die korreliert sind zu einem Aspekt oder einer Perspektive (viewpoint)



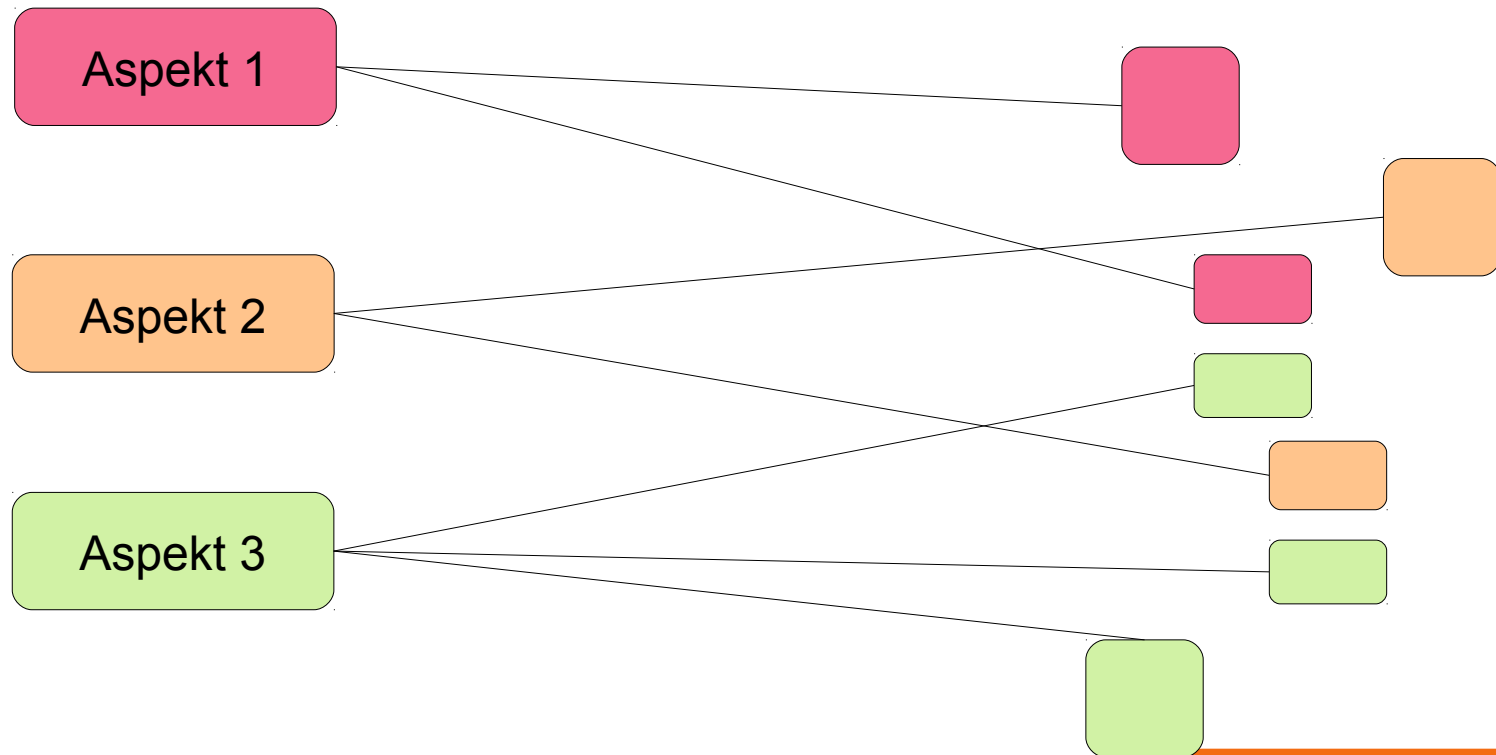
# Aspektmarkierung ist unabhängig von Abstraktionsebene

- ▶ Aspekte für Module:
  - Für alle Module: ordne Aspekte zu
- ▶ Aspekte für Komponenten:
  - Für alle Komponenten: ordne Aspekte zu
- ▶ Aspekte für Aktivitäten, Speicher, Stellen:
  - Für alle Modellelemente: ordne Aspekte zu



# Aspekttrennung

- ▶ Aspekttrennung: Transformiere Einheiten derart, dass jede Einheit nur zu einem Aspekt (Farbe) korreliert ist
  - Spalte Einheiten ggf. auf



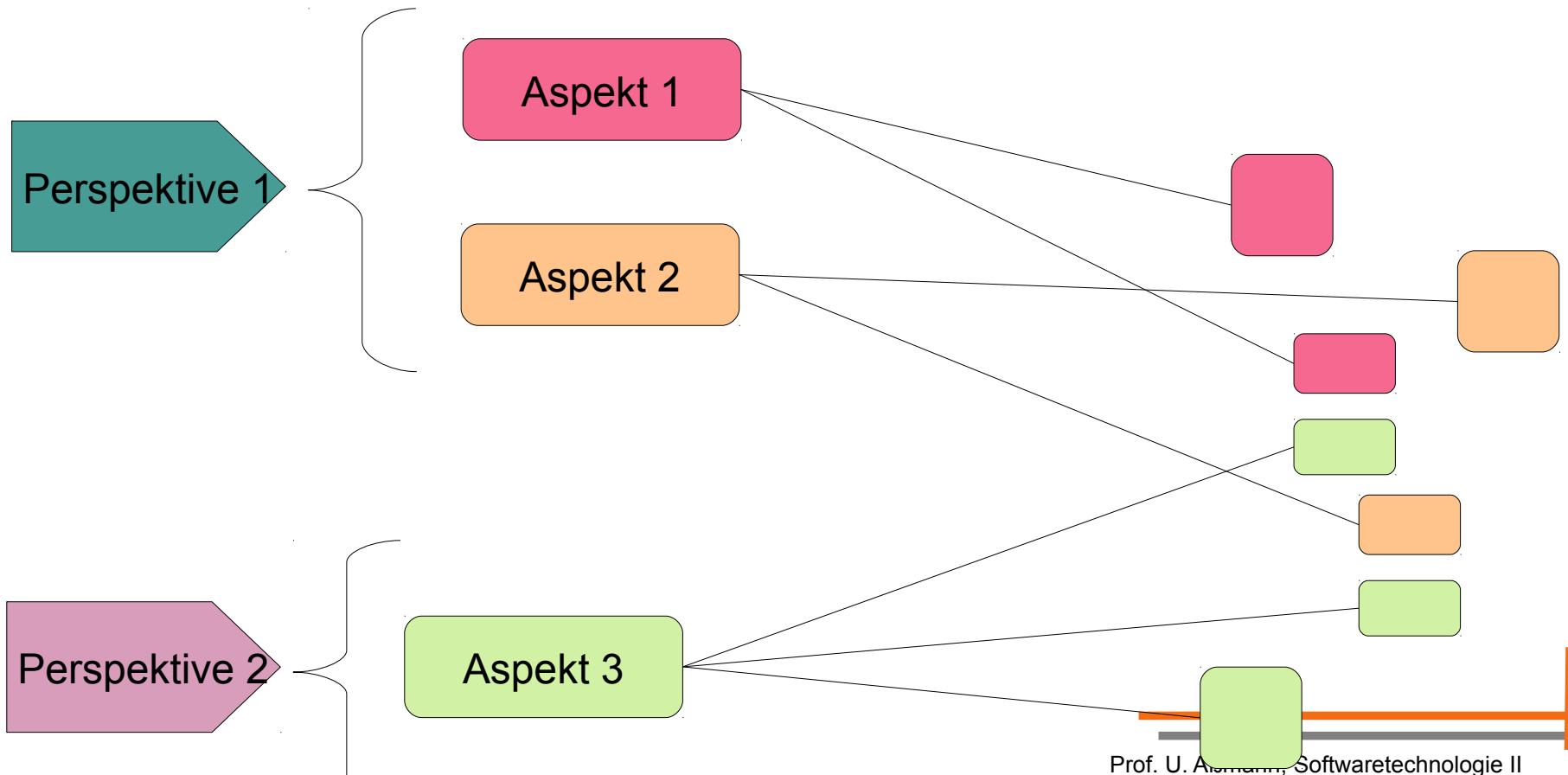
# Gesetz der aspektseparierten Systeme

Ist ein System aspektsepariert, können die zu einem Aspekt korrelierten Komponenten des Lösungsraumes leicht ausgetauscht werden

- ▶ Austausch von Implementierungen
- ▶ Austausch von Fragmenten
- ▶ Austausch von Modellelementen

# Perspektiven (Viewpoints)

- ▶ Eine **Perspektive** besteht aus einer Menge von Aspekten (Belangen)
- ▶ Eine **Sicht** aus einer Perspektive besteht aus allen Sichten des Systems, die den Aspekten der Perspektive zugeordnet sind
- ▶ Eine **einfache Perspektive** besteht aus einem Aspekt



# Perspektivenmodelle (Viewpoint Models)

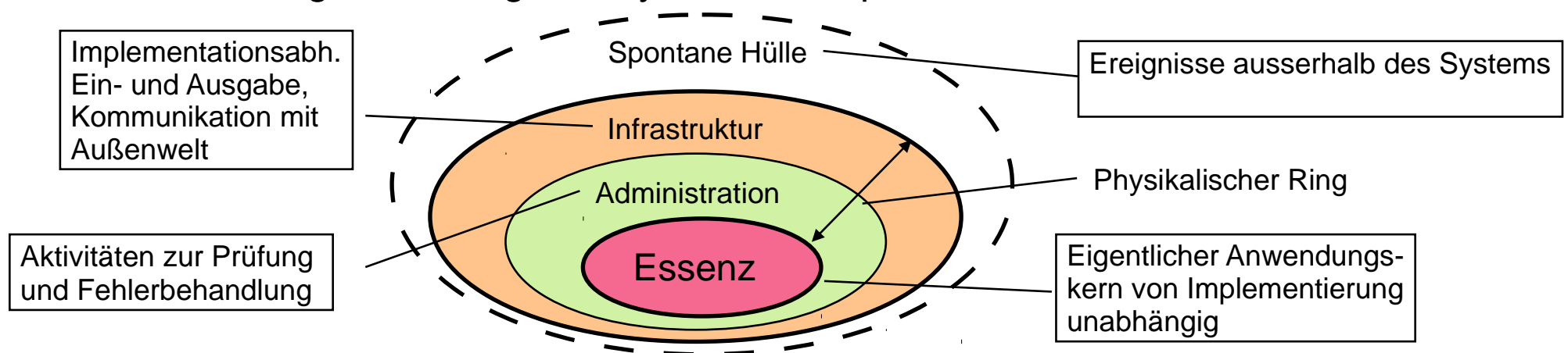
- ▶ Ein **Perspektivenmodell** legt eine Menge von Perspektiven mit ihren Aspekten und Sichten auf das System fest.

## 28.2 Essentielle Zerlegung in der Strukturierten Analyse

Prozessverfeinerung

# Konzept der essentiellen Zerlegung

- ▶ **Essentielle Zerlegung** findet die **Essenz** eines Systems, d.h. derjenigen Systembestandteile, die unabhängig von der Implementierung sind.
  - Essenz nimmt perfekte Technologie an [McMenamen/Palmer]
    - Prozesse ohne Bearbeitungszeit, Speicher stets aktuell mit unendl. Kapazität
- ▶ Essentielle Zerlegung ist ein Sichtenraum mit 3 Aspekten (Essenz, Administration, Infrastruktur)
  - Lösung: Einteilung des Systems in Aspekte:



Bestandteile der Essenz in der SA:

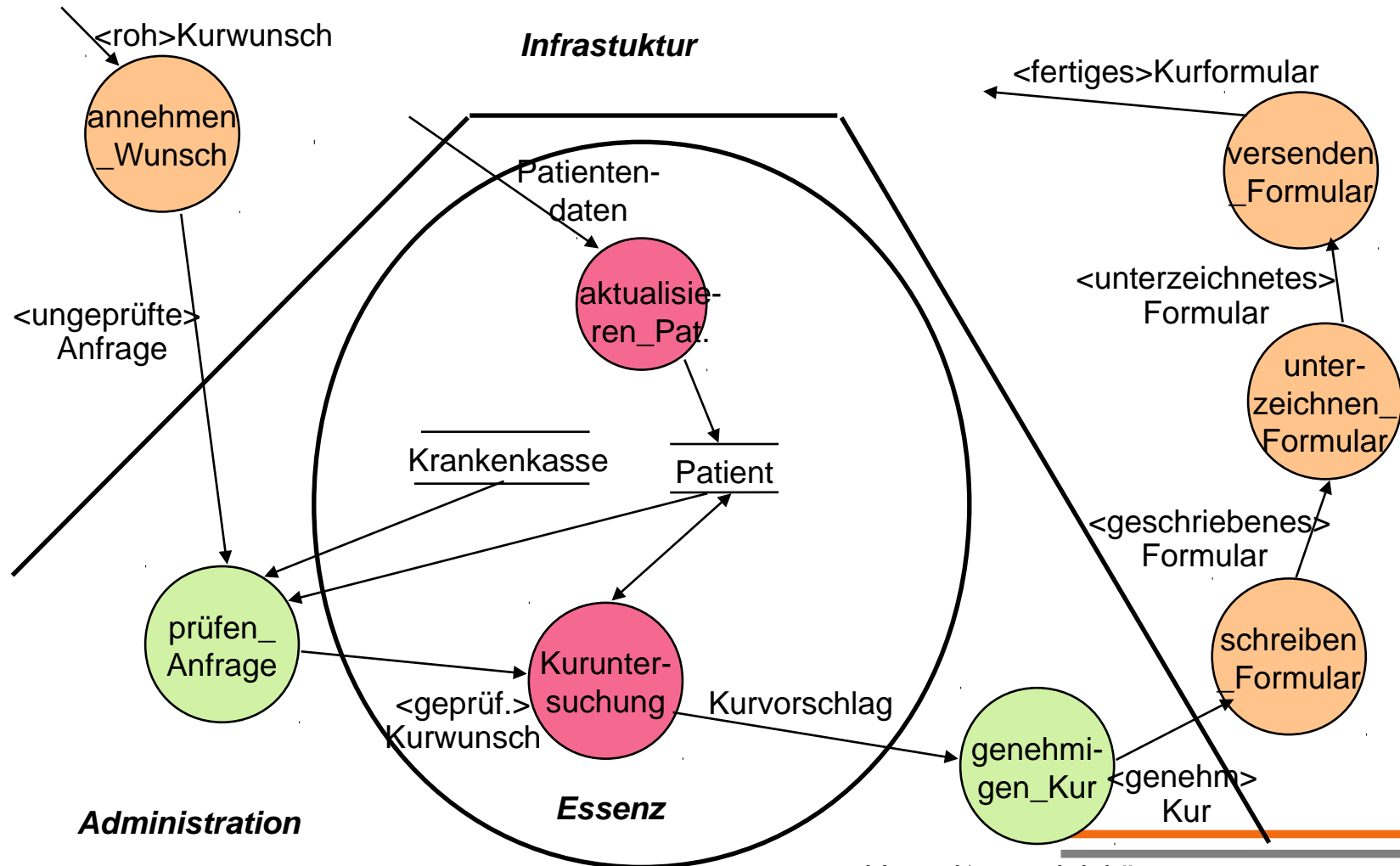
Essentielle Aktivitäten

Essentielle Speicher



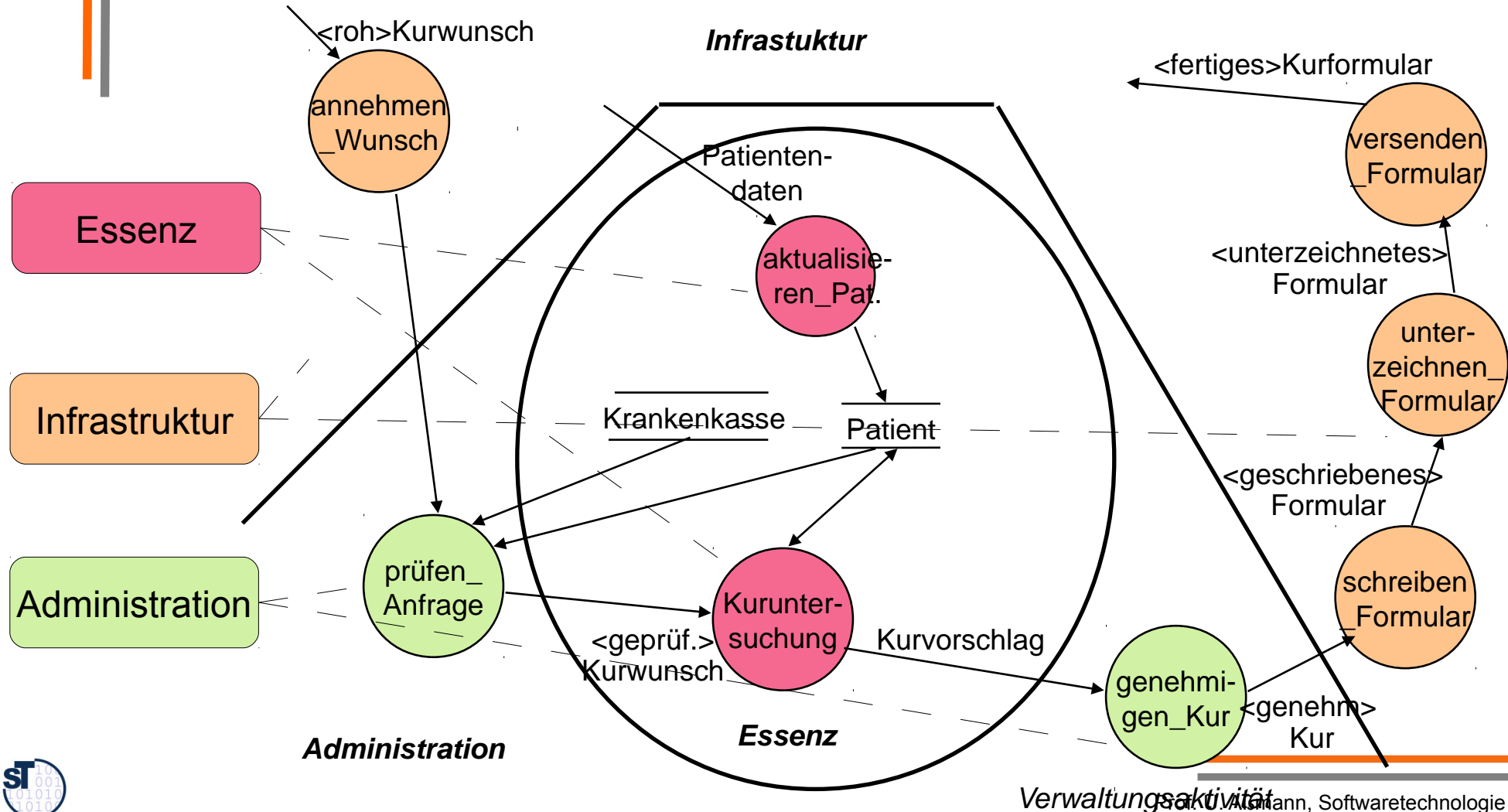
# Beispiel EAI-Zerlegung eines Geschäftsprozesses "Kurantrag"

- Die EAI-Zerlegung eines Geschäftsprozesses (Farbzerlegung mit Essenz, Administration, Infrastruktur)



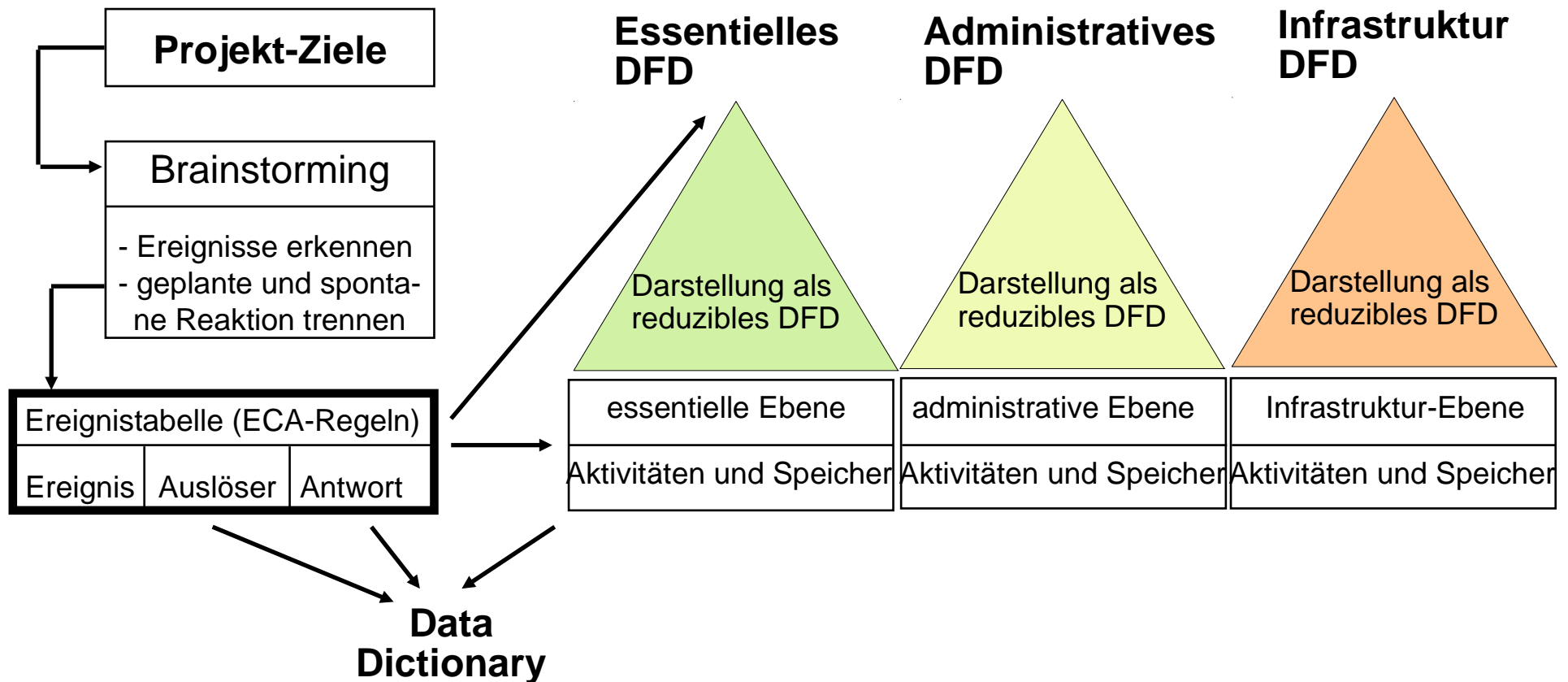
# Beispiel EAI-Zerlegung eines Geschäftsprozesses "Kurantrag"

- Die EAI-Zerlegung eines Geschäftsprozesses (Farbzerlegung mit Essenz, Administration, Infrastruktur)



- 
- ▶ In DFD geht die Einteilung in Aspekte (Essenz, Administration, Infrastruktur) sehr einfach, nämlich durch Graphslicing.

# Vorgehen zur essentiellen strukturierten Analyse



# Methodik der essentiellen strukturierten Analyse

- ① **Ziele des neuen Systems festlegen**, d. h. klare Aufgabenabgrenzung und Bestimmung aller Projektteilziele.
- ② **Grundlegende Aktivitäten finden** über Aufstellen der **Ereignisse** in einer Ereignistabelle, die über **Eingabe-Datenflüsse** möglicherweise **Ausgabe-Datenflüsse** und damit einen **Prozess** initiieren.  
Aus der vollständigen Ereignistabelle ist ein erster Teil der essentiellen Ebene zu erstellen. Nach Erkennen der wahren Terminatoren kann im Falle einer strukturierten Analyse das Kontextdiagramm entworfen werden.
- ③ **Speicher des Systems finden** über Analyse der Ein- und Ausgabewerte von Datenflüssen, ihrer **Spezifikation im DD** und der semantischen Modellierung von Datenstrukturen
- ④ **Verwaltungsaktivitäten finden**, die zur Erstellung und Pflege essentieller Speicher (Laden, Änderung, Löschen,...) notwendig sind. Nur solche Ereignisse, die nicht schon grundlegende Aktivitäten auslösen, definieren Verwaltungsaktivitäten.
- ⑤ **Vorläufiges essentielles Modell erstellen** aus allen Ereignissen, die eine geplante Systemreaktion durchführen und damit die essentiellen Aktivitäten implizieren. Nach Integration eventueller Teilmodelle kann die Vergrößerung zum Kontext erfolgen.
- ⑥ **Wiederholung und Verfeinerung bis zu den MiniSpecs**, wobei die Vorgehensweise nicht starr "top-down", sondern "inside-out-yoyo" ist.

# Erweiterung auf Reengineering von Altsystemen

## Ohne Analyse des Altsystems

A

Ziele  
Anwendergespräche  
Kontextabgrenzung

Ereignistabelle  
Essenz, Kontext (2)  
Datenkatalog 3...6  
MSpecs (Für Elementar\_Aktivit.)

Modellqualität optimieren  
ess. Aktivitätenmodelle integrieren (5)

neue Essenz konzipieren  
neue Inkarnation konstruieren

## Mit Analyse des Altsystems

B

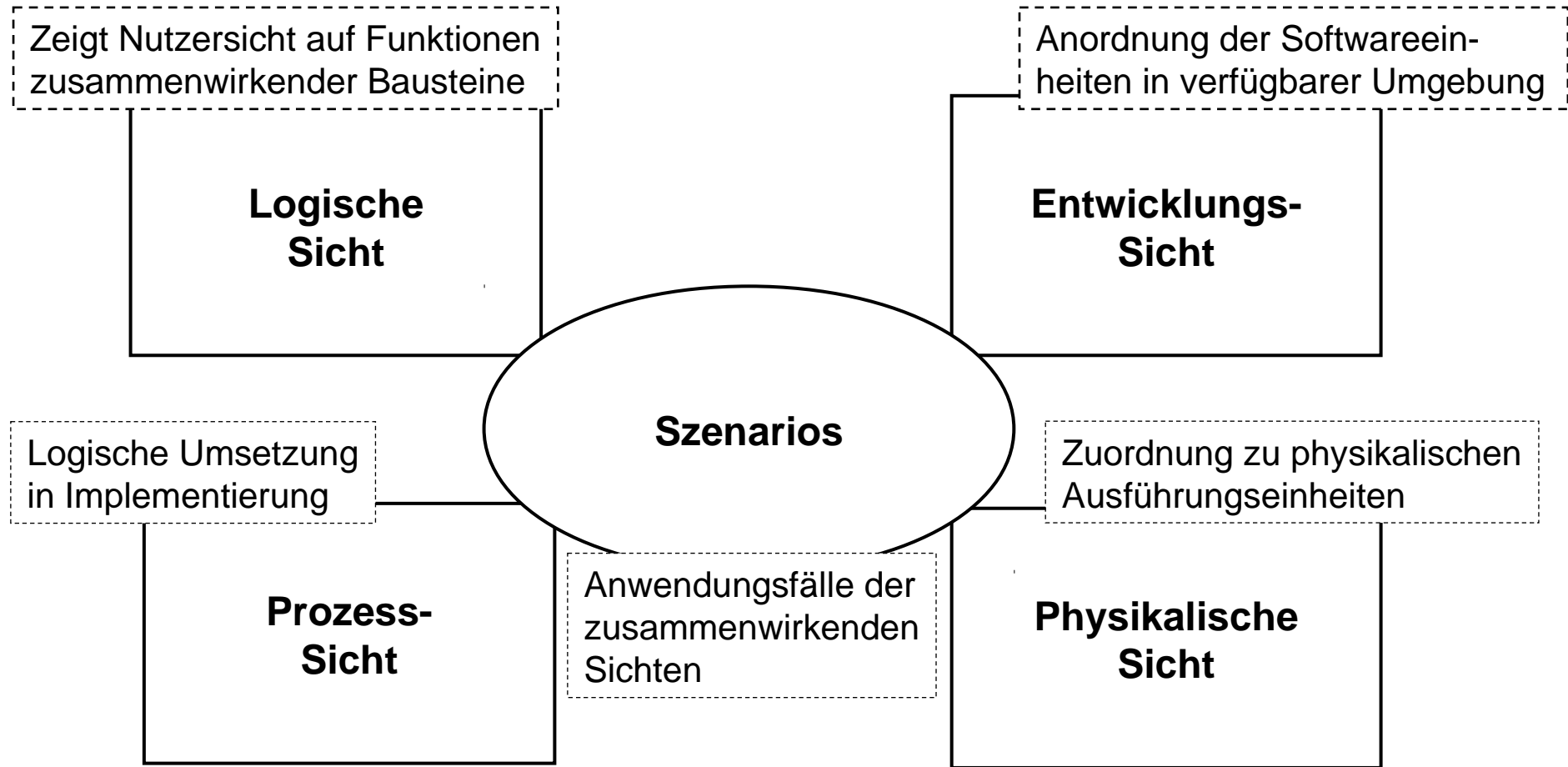
Ist-Zustand erheben  
expandieren  
reduzieren  
essentielle Fragmente klassifizieren  
essent. Aktivitäten ableiten (2)

Modellqualität optimieren  
ess. Aktivitätenmodelle integrieren (5)

neue Essenz konzipieren  
neue Inkarnation konstruieren

# Architektursichten

nach dem 4+1 View Model of Architecture



**Quelle:** Kruchten, P., Vancouver, B., C.: The 4+1 View Model of Architecture; IEEE Software, 12 (6), November 1995, IEEE, S. 42-50

# The End – Was haben wir gelernt?

- ▶ Aspektzerlegung zerlegt eine Software in mehrere Belange
  - Bildet eine Abbildung zu einem “Concern space” (Belang-Raum)
- ▶ Perspektivenmodelle definieren Aspekte, Sichten für Systeme
- ▶ Beispiele:
  - EAI
  - Blutgruppen von Siedersleben