

## **Objektorientierter Zuschnitt von Geschäftsprozessen, Geschäftsobjekten und E-Business-Systemen mit der UML**

**Prof. Dr. Alfred Zimmermann**

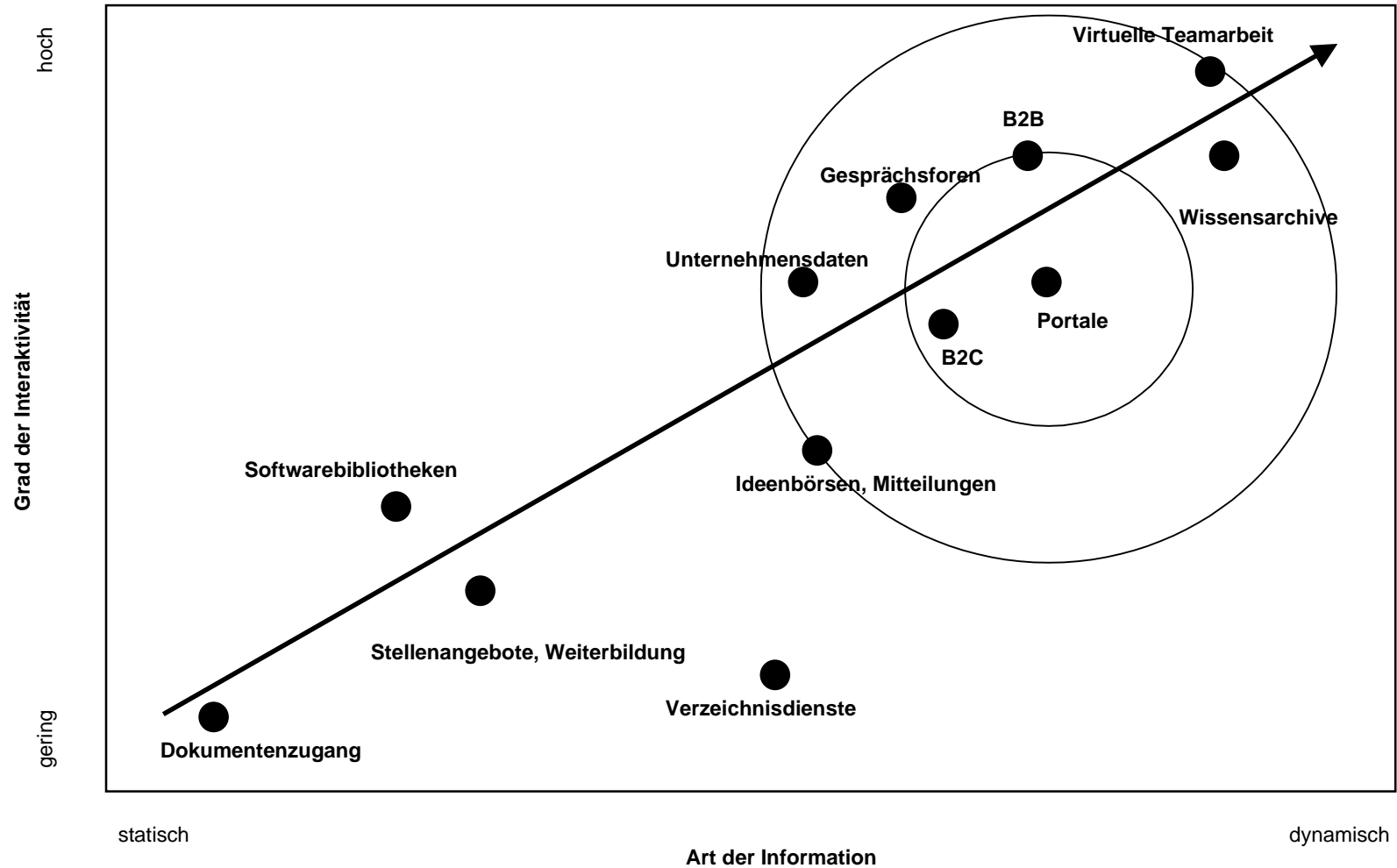
# Gliederung

---

- ➔ **Architekturen für das E-Business**
- **BOE: Methodik und Vorgehensweise**
- **Adaption auf Basismodelle der UML**
- **Erfahrungen aus praktischen Projekten**

# E-Business im Unternehmen

## Integration von B2X-Anwendungen im Internet



# Architekturzentrierter Entwicklungsprozeß

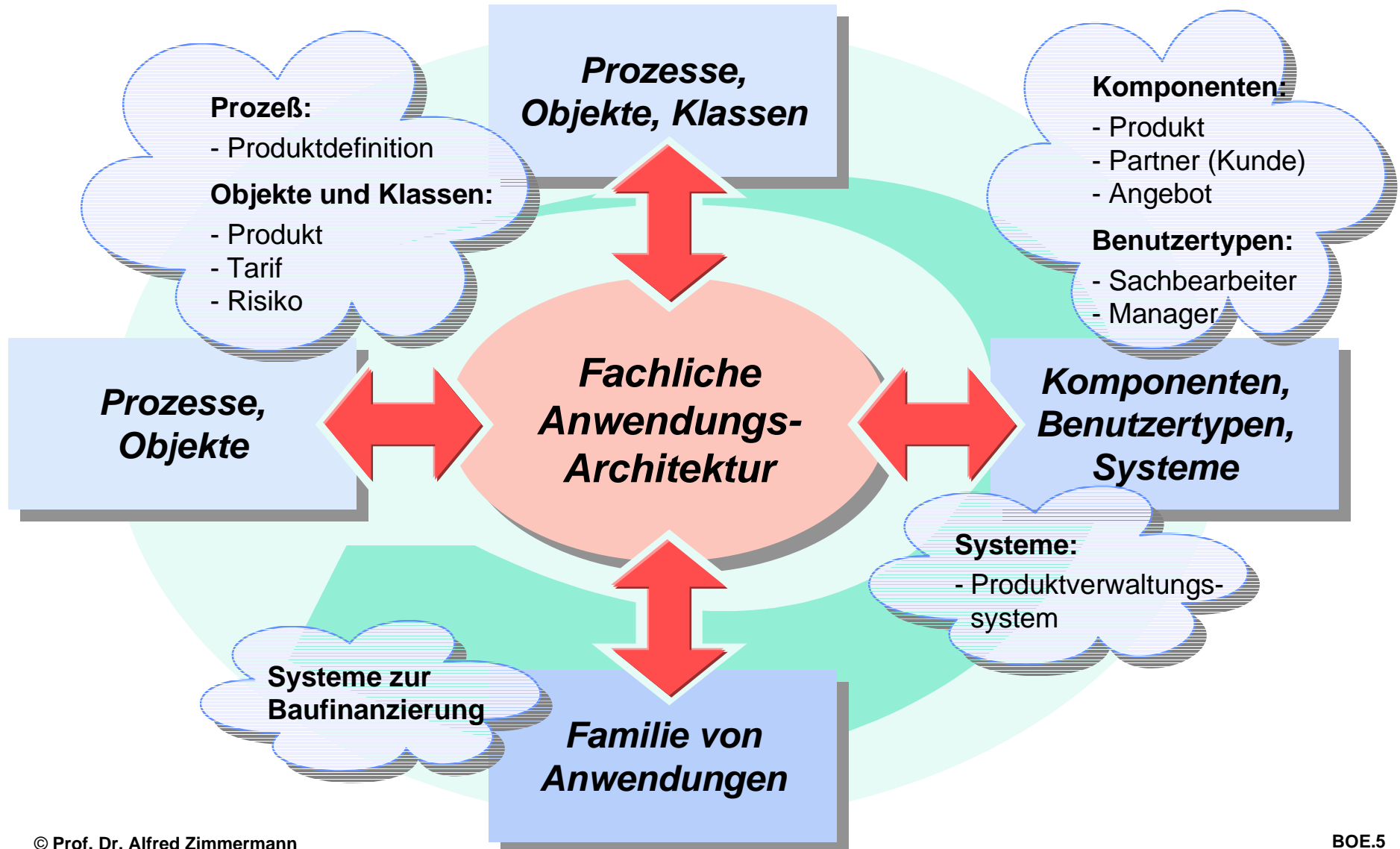
## Sichten objektorientierter Architekturen

---

- Die fachliche Sicht liefert ein statisches Bild der primären Klassen und ihrer Beziehungen sowie eine erste fachliche Gruppierung der Klassen
- Die Entwicklungssicht zeigt die Organisation des Codes in Paketen und Bibliotheken sowie die benutzten Komponenten und integrierten Fremdsysteme
- Die Prozeßsicht zeigt die beteiligten Prozesse und Tasks
- Die physische Sicht stellt den Bezug zu Plattformen, Prozessoren und Geräten her

# Beispiel: Fachliche Anwendungsarchitektur

## Durchgängige objektorientierte Modellbildung

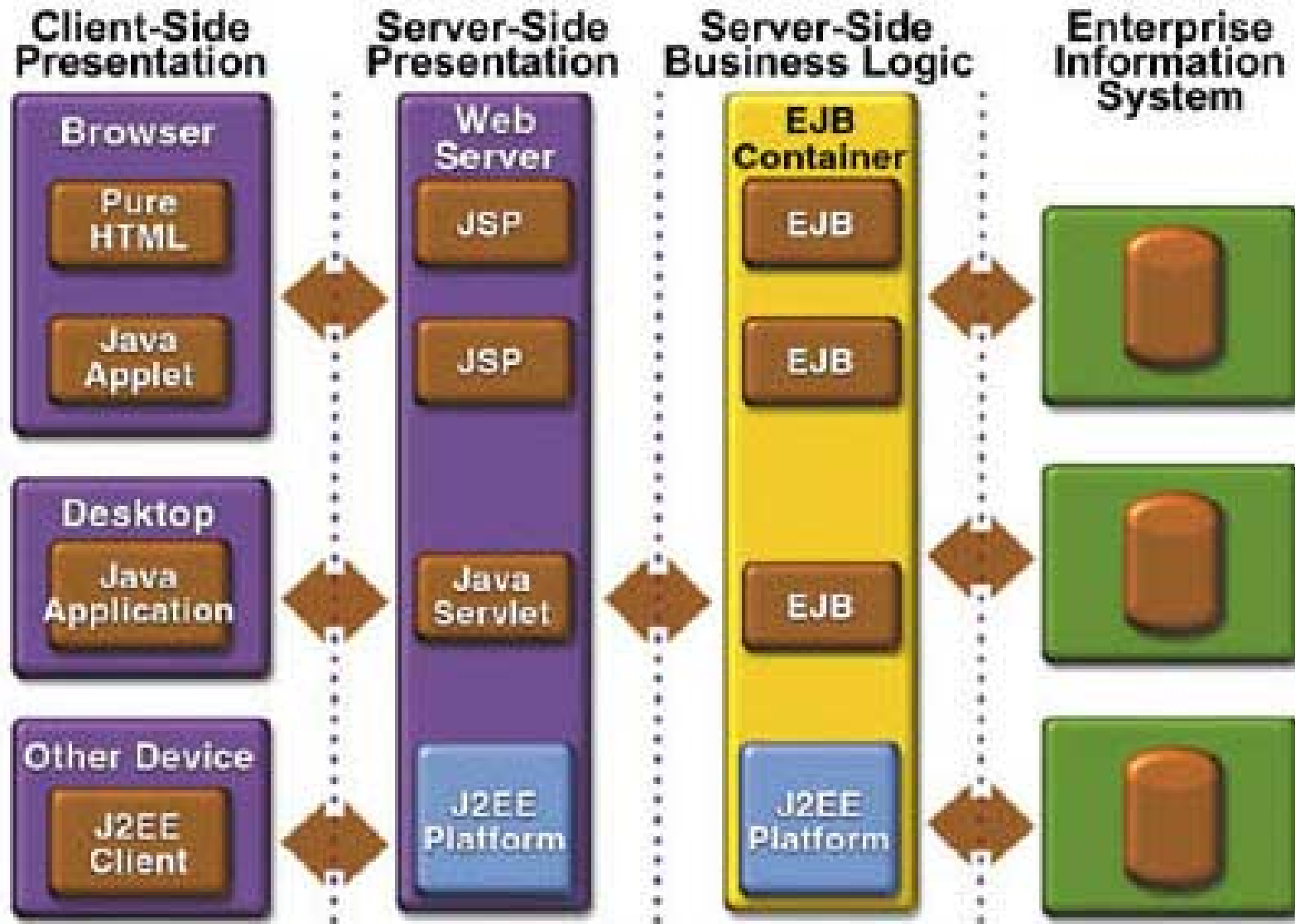


# Fachliche Anwendungsarchitektur: Objektorientierte Elemente

---

- **Objektorientierte Geschäftsprozeßmodelle liefern den Kontext für Systemdienste, die zukünftige Benutzer vom System erwarten**
- **Die fachliche Architektur enthält folgende Elemente:**
  - **Objektorientierte Sicht auf den Geschäftsprozeß**
  - **Geschäftsobjekte, als fachliche Klassen mit ihren Beziehungen**
  - **fachliche Komponenten**
  - **Systemfunktionen**
  - **Systemfamilien, als fachlich integrierte Anwendungssysteme**

# JAVA 2 Enterprise Edition: Application Model (Quelle: SUN)



# Objektorientierte System- und Softwarearchitektur

---

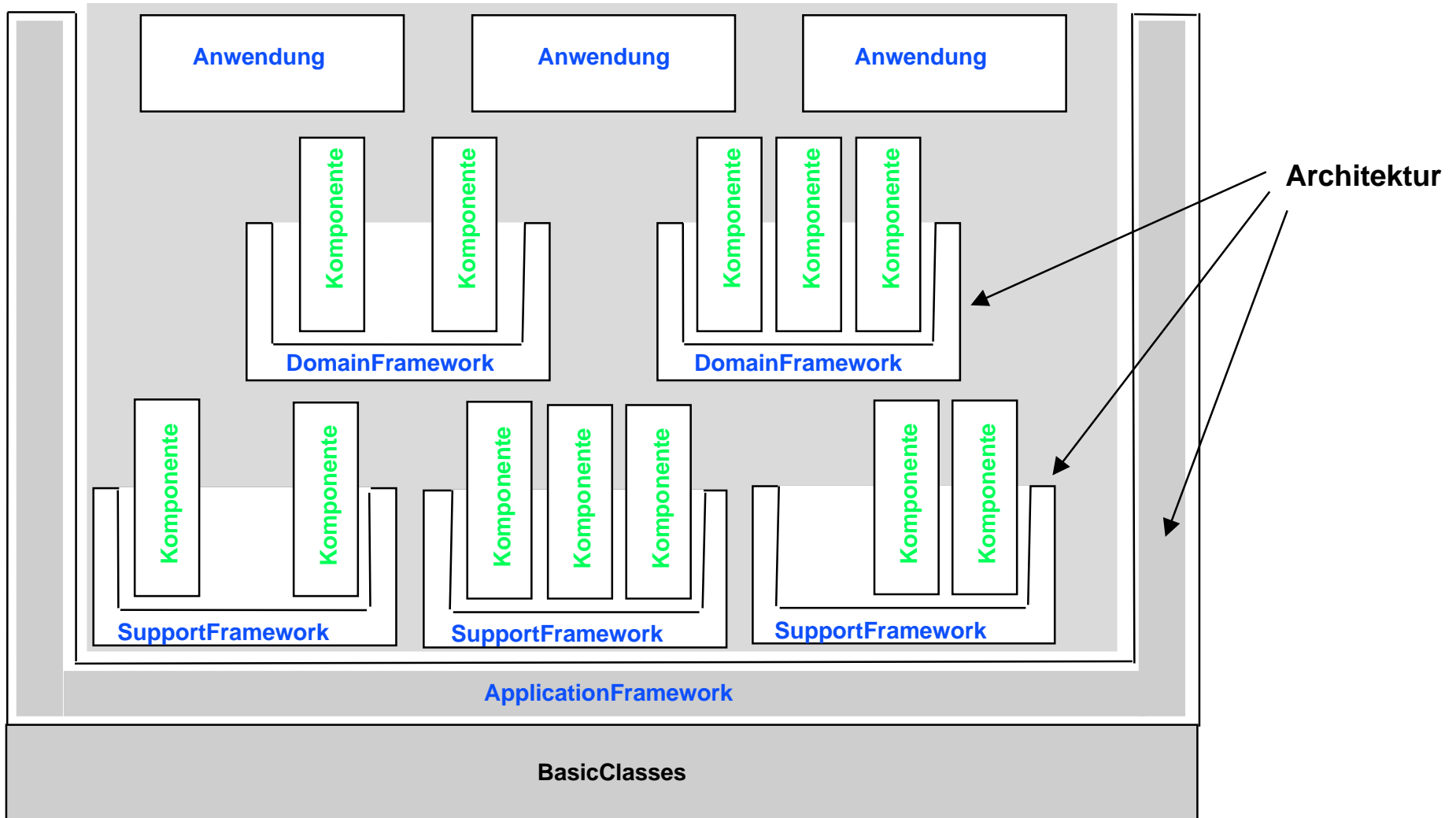
**Eine nach eher statischen Kriterien ausgerichtete Grobstruktur zur Ordnung der Komponenten eines Systems, aufgrund von globalen Konstruktionsprinzipien wie Objektorientierung, Verteilbarkeit, Partitionierbarkeit, Portabilität, Wiederverwendung und Modularität.**

**Die Software-Architektur eines objektorientierten Client/Server-Systems wird durch Schichten, Komponenten (Pakete), Klassen und deren Beziehungen mit Hilfe von Entwurfsmustern (Patterns) für Architekturen ausgedrückt.**

**Die Software-Architektur ist Basis für ein Objekt-Konfigurationsmodell, bei dem zusätzlich Versionszustände und Ladereihenfolgen der Komponenten einer speziellen Systemkonfiguration eine Rolle spielen.**

**Die System-Architektur gruppiert zusätzliche Elemente wie Plattformen, Netze, Systemsoftware, Middleware, Laufzeitumgebungen, Datenbanken, Frameworks, Komponenten und fremde Systeme zu einer Ordnungsstruktur.**

# Architektur komponentenbasierter Anwendungssysteme



# Objektorientierte Komponenten

---

**Eine Komponente ist eine Bündelung von technisch oder fachlich ausgeprägten Grundfunktionalitäten ein oder mehrerer Objekte, um als wiederverwendbare Baugruppe in Anwendungen genutzt zu werden.**

**Black-Box Komponenten können einfach durch ihre äußere Schnittstelle durch Kollaboration, d.h. mittels Botschaften, aufgerufen werden.**

**White-Box Komponenten hingegen erfordern für ihre Benutzung eine detailliertere Betrachtung und Anpassung des internen Framework-Verhaltens.**

**Als ein wichtiger Strukturmechanismus für Komponenten hat sich durch die Programmiersprache JAVA die Komponententechnologie JAVA-Beans und Enterprise JavaBeans (EJB) herauskristallisiert.**

# Gliederung

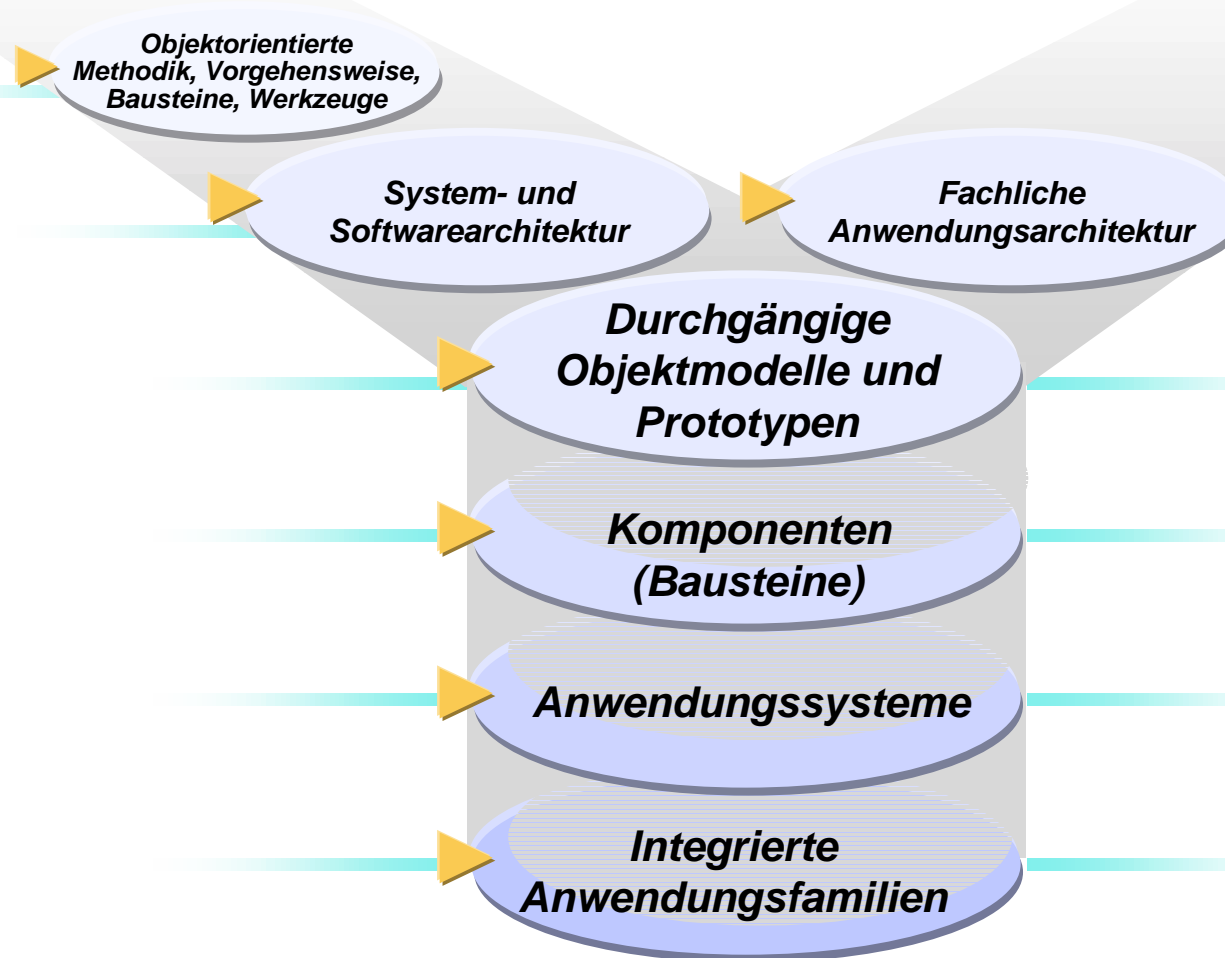
---

- **Architekturen für das E-Business**
- ➔ **BOE: Methodik und Vorgehensweise**
- **Adaption auf Basismodelle der UML**
- **Erfahrungen aus praktischen Projekten**

# BOE (Business ObjectEngineering): Vorgehensweise der objektorientierten Systementwicklung

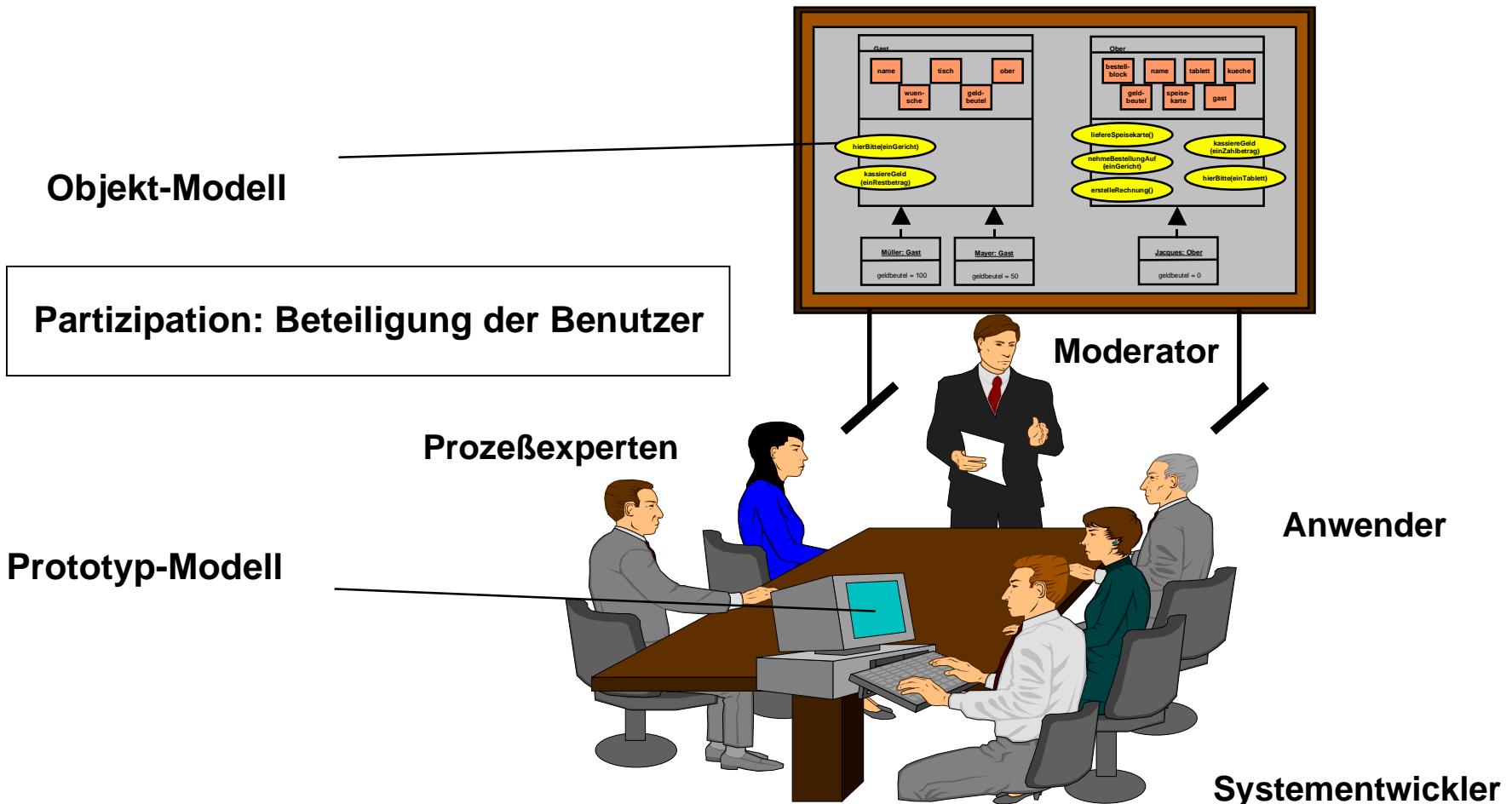
*Technologie*

*Geschäftsprozesse*



# BOE (Business ObjectEngineering): Moderierte Workshops als Kommunikationsinstrument

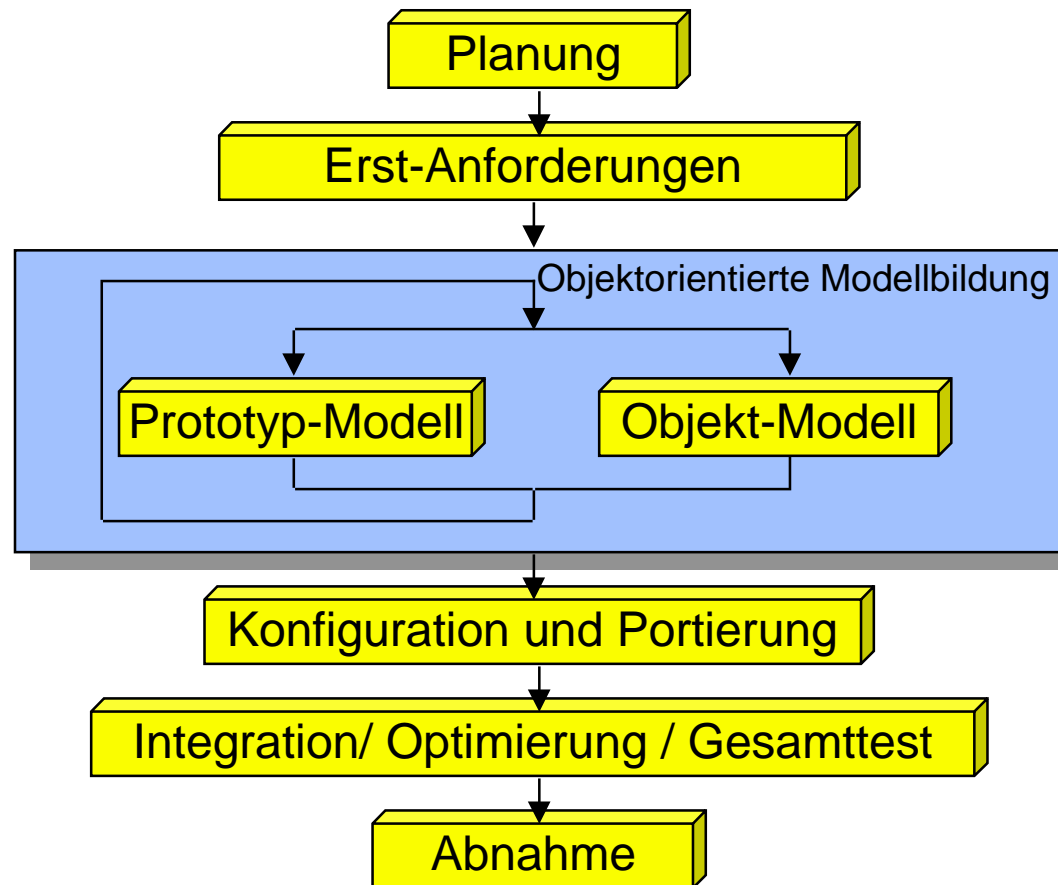
Förderung einer zielorientierten, einheitlichen und offenen Kommunikation  
als Voraussetzung für eine pragmatische und effiziente Projektarbeit im Team



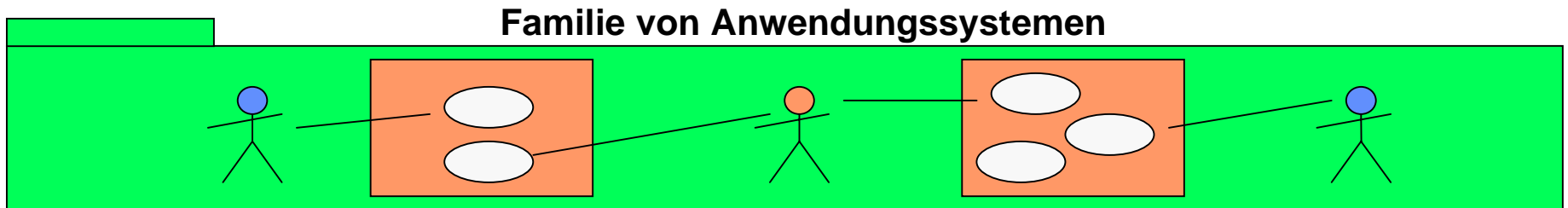
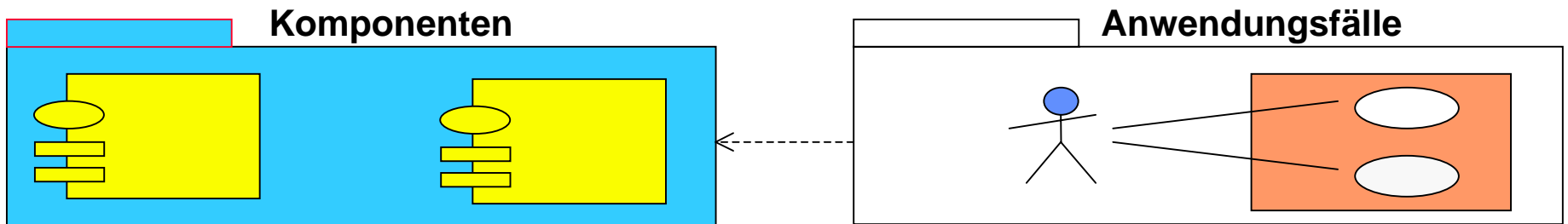
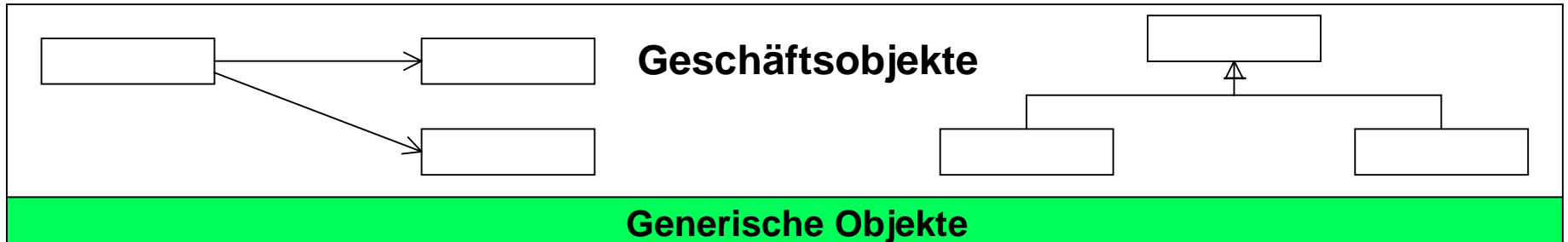
# BOE (Business ObjectEngineering): Objektorientiertes Vorgehensmodell: inkrementell und iterativ

---

Die zyklische Vorgehensweise stellt die frühe Verfügbarkeit von anfaßbaren und bewertbaren Zwischenergebnissen in Form von Prototypen sicher. Diese werden zum Zielsystem ausgebaut.



# BOE (Business ObjectEngineering): Objektorientierte Geschäftsprozesse und Systeme



# **BOE (Business ObjectEngineering): Architekturelemente (1)**

---

- **Geschäftsprozesse:**  
Organisationseinheiten, Partner (Kunden, Lieferanten, Wettbewerber), Geschäftsvorfälle (Auslöser), Prozeßszenarien, Prozesse und Ziele (Kundenziele/Betriebsziele), Prozeßmodell
- **Geschäftsobjekte:**  
Dienstleistungen, Abhängigkeiten, Kollaborationen, Zustände, Attribute, Klassenmodelle
- **Generische (standardisierte) Geschäftsobjekte und Komponenten**
- **integrierte Komponenten**
- **Workflow-Steuerungsregeln**  
(Basis auch für Prozeßsimulation und BusinessEngineering)
- **Komponentendefinition und Komponentenschnittstellen**

# **BOE (Business ObjectEngineering): Architekturelemente (2)**

---

- **Akteure: Benutzer und Fremdsysteme der Komponenten**
- **Systemverhalten (Anwendungsfälle)**
- **Systemfamilien:  
gemeinsam benutzte Geschäftsobjekte und Anwendungskomponenten**
- **Testdaten und Testtreiber für Systemfunktionen und Komponenten**
- **Interaktion (Look & Feel): Navigationsmengen und Ergonomie**
- **Ablaufsteuerungsobjekte (Sitzungsobjekte)**
- **persistente Objekte (Entitätsobjekte) und Abbildung auf Relationen**
- **nichtfunktionale Ziele für:  
System- und Softwarearchitektur, fachliche Komponenten,  
Anwendungssysteme**

# Gliederung

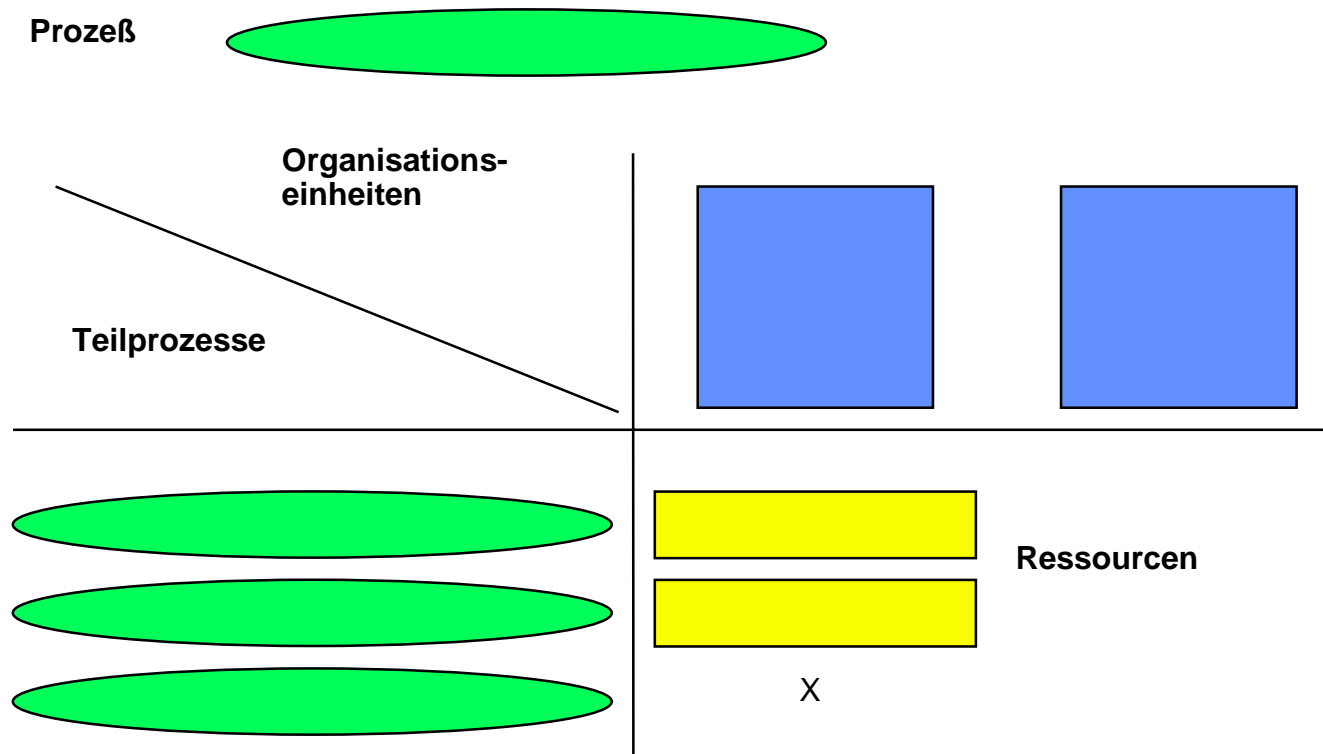
---

- **Architekturen für das E-Business**
- **BOE: Methodik und Vorgehensweise**
- ➔ **Adaption auf Basismodelle der UML**
- **Erfahrungen aus praktischen Projekten**

# BOE (Business ObjectEngineering): Prozeßmodell

---

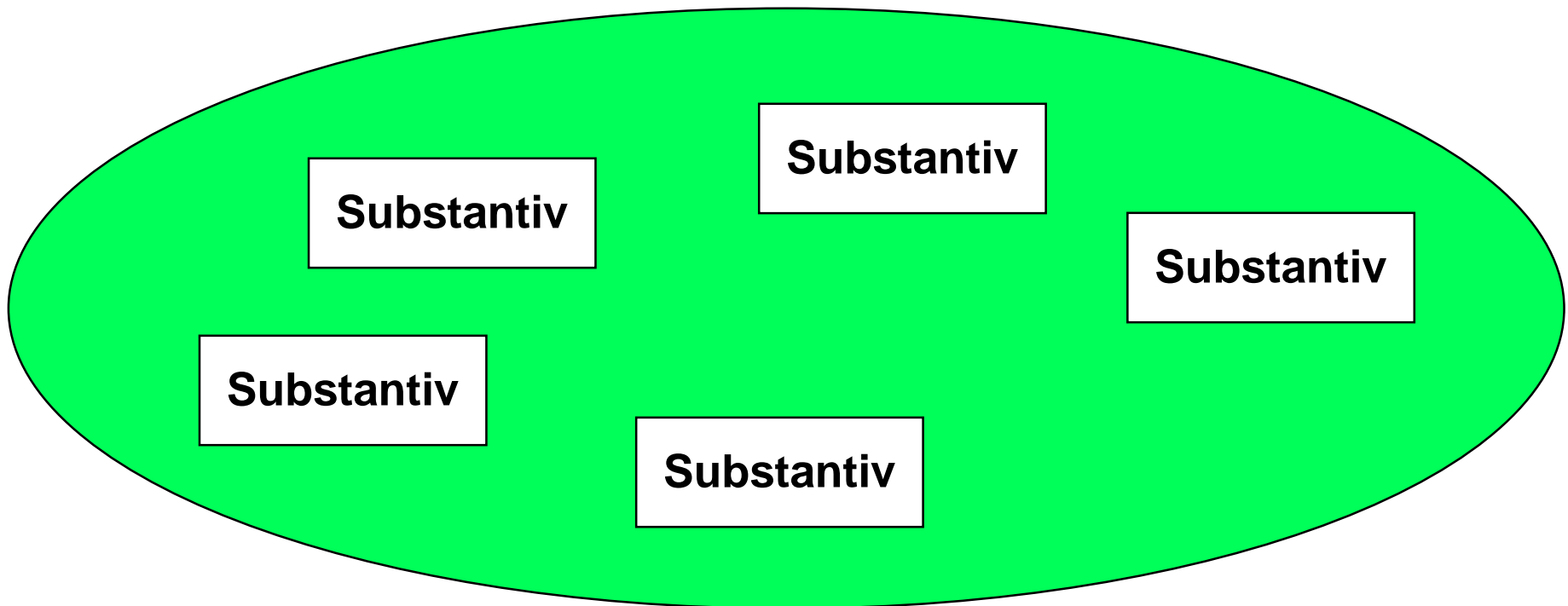
- Welche Ressourcen werden in den zugehörigen Organisations-  
einheiten eingesetzt, verbraucht oder erzeugt, um den betrachteten  
Teilprozeß zu ermöglichen?



# BOE (Business ObjectEngineering): Geschäftsobjekte

---

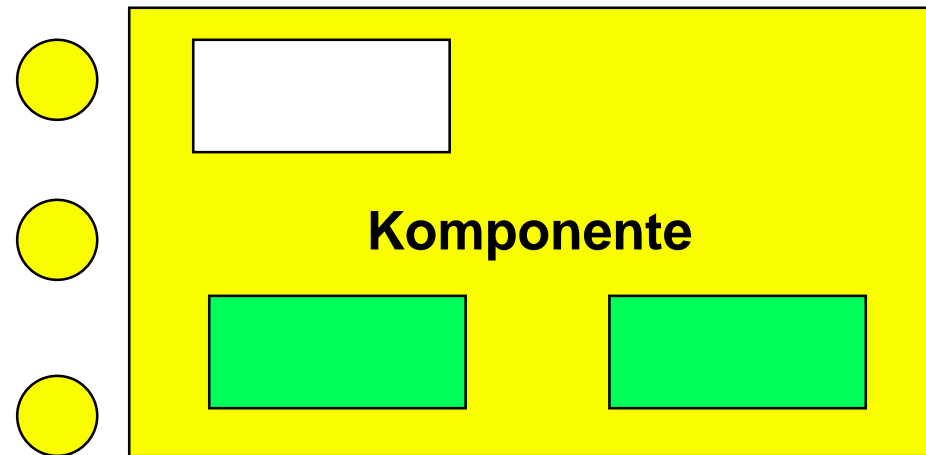
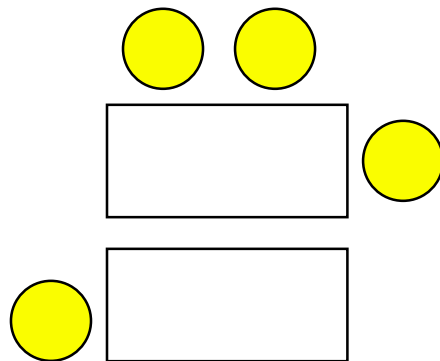
- Welche Informationen, Instrumente, Dokumente, Ordner, Mappen, Karteikästen, Gegenstände, Dinge, Begriffe, Konzepte, Systeme, Einrichtungen, Ausrüstungsgüter, Apparate, Hilfsmittel, Materialien, Ressourcen, Personen, Rollen, Organisationseinheiten, Kunden, Lieferanten, Finanzen, Gebäude, Orte und Ereignisse sind am betrachteten Prozeß beteiligt?



# BOE (Business ObjectEngineering): Komponenten und Schnittstellen

---

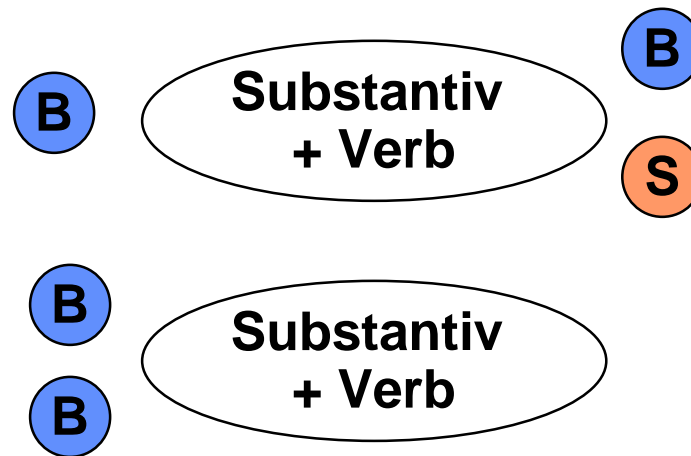
- Wie lassen sich Geschäftsobjekte nach ähnlicher oder zusammengehöriger Fachlichkeit und Kollaborationsdichte zu Komponenten (Paketen) gruppieren?
- Welche Schnittstellen spezifizieren einen Zugang zu Objekten und Komponenten?
- Welche Dienstleistungen (Methoden-Signaturen) definiert jede Schnittstelle (Interface)?



# BOE (Business ObjectEngineering): Akteure und Systemverhalten

---

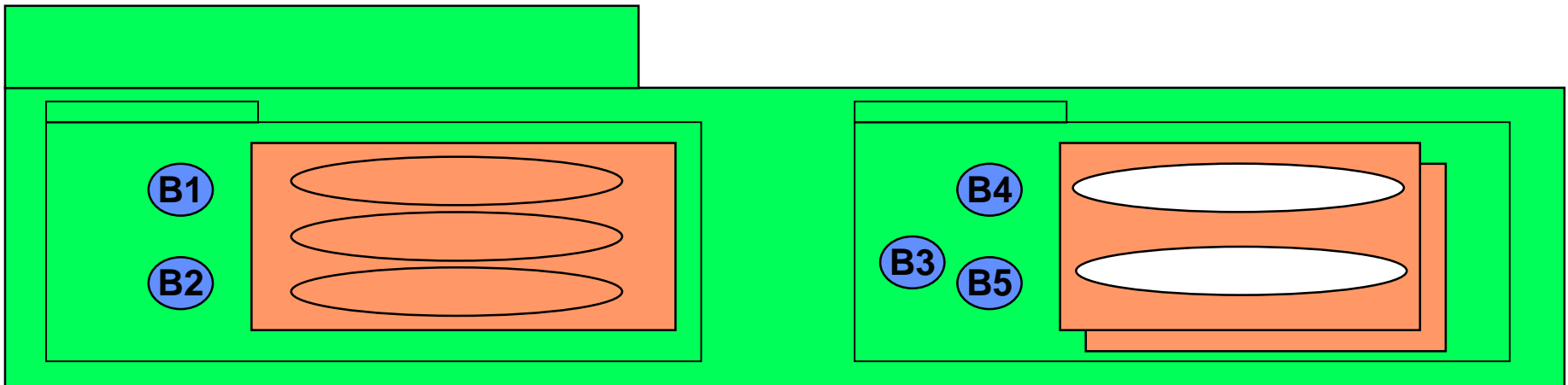
- Welche Benutzer und Fremdsysteme nutzen direkt oder indirekt die Dienstleistungen der betrachteten Komponente?
- Welche beispielhaften Situationen kennzeichnen die Nutzung des Systems durch Benutzer und Fremdsysteme?
- Welche Anwendungsfälle beschreiben den äußeren Zugang zur Benutzung von Systemfunktionen durch Akteure (Benutzer oder Fremdsysteme), um den Zweck der betrachteten Geschäftsprozesse zu ermöglichen?



# BOE (Business ObjectEngineering): Systeme und Systemfamilien

---

- Welche Systeme entstehen durch die fachliche Gruppierung von Systemfunktionen?
- Wie lassen sich Systeme für einzelne Benutzertypen gruppieren?
- Welche Referenzsysteme können aus dieser konkreten Situation heraus verallgemeinert werden?
- Welche Familie(n) von Anwendungssystemen entstehen durch das Zusammenspiel der gemeinsam benutzten Geschäftsobjekte und Anwendungskomponenten?



# BOE (Business ObjectEngineering): Beschreibung der Systemfunktionen

---

- Wie lautet die Kurzbeschreibung der Systemfunktion?
- Welche Start-Ereignisse lösen den Ablauf der Systemfunktion aus?
- Welche Start-Daten (Parameter) sind die für den Ablauf notwendigen Daten?
- Wie lautet die Ablaufbeschreibung?
- Welche Ende-Ereignisse entstehen am Ende des Ablaufs?
- Welche Ergebnis-Daten liefert der Ablauf?
- Welche Ausnahmen beschreiben fachliche Besonderheiten des Ablaufs?

**Systemfunktionsbeschreibung : strukturierter Text**

# BOE (Business ObjectEngineering): Wesentliche Adaptionen für die UML

---

<b>BOE</b>	<b>UML</b>
<b>Prozeßmodell</b>	<b>Activity Diagram (Business) Use Case Diagram</b>
<b>Geschäftsobjekte</b>	<b>Collaboration / Sequence Diagram Class Diagram</b>
<b>Komponenten und Schnittstellen</b>	<b>Packages / Component Diagram Interfaces</b>
<b>Akteure und Systemverhalten</b>	<b>(System) Use Case Diagram</b>
<b>Systeme und Systemfamilien</b>	<b>(System) Use Case Diagram Packages</b>
<b>Systemfunktionsbeschreibung</b>	<b>(System) Use Case Description</b>

# Gliederung

---

- **Architekturen für das E-Business**
- **BOE: Methodik und Vorgehensweise**
- **Adaption auf Basismodelle der UML**
- ➔ **Erfahrungen aus praktischen Projekten**

# **Objekttechnologie aus meiner Praxis: Seit 1985 mehr als 35 erfolgreiche objektorientierte Projekte**

---

- **Grafik-Datenbanken**
- **Diagnose- und Therapiesysteme**
- **Verwaltung von Verbesserungsvorschlägen**
- **Finanz- und Controlling-Systeme**
- **Vertriebs- und Garantiesysteme**
- **Logistiksysteme**
- **Versicherungssysteme**
- **Banksysteme**
- **Intelligente Objektorientierte Systeme (Wissensmanagement)**

# Informationssysteme als Ressourcen von Geschäftsprozessen

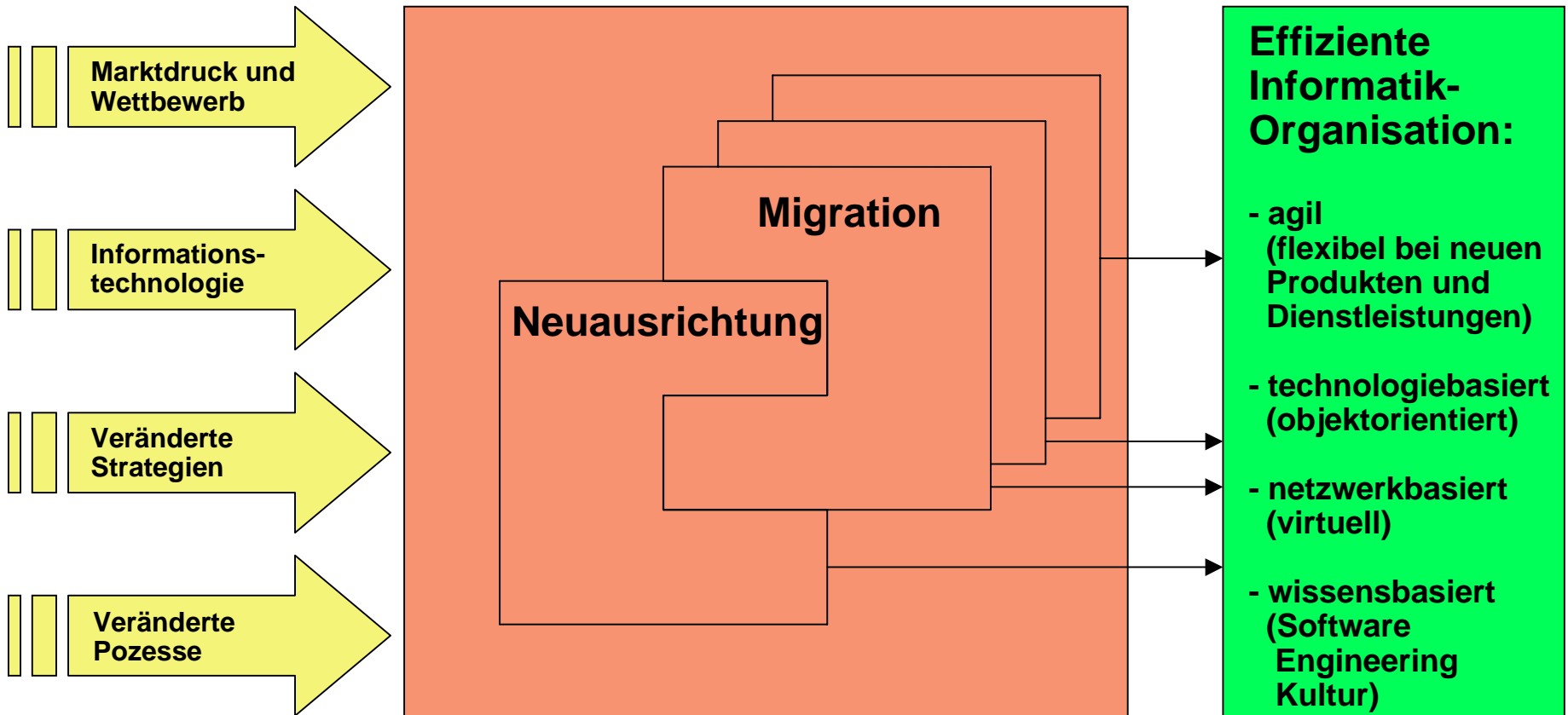
---

- **Abkehr vom nur systembezogenen Denken**
- **konsequente Ausrichtung der Systeme nach Geschäftsprozessen**
- **Optimierung der Prozeßkette, nicht des einzelnen Systems**
- **Reduktion der Komplexität der Schnittstellen durch fachlich integrierte Systemfunktionen und Familien von Anwendungssystemen**
- **Prozeßparameter „Time to Market“ und „Target Costing“ als Basis der Implementation von Fachfunktionalitäten**
- **Kostenoptimierung durch Standardsoftware und Kaufkomponenten (künftig auch durch Geschäftskomponenten)**
- **Differenzierung zum Wettbewerb durch Individualsoftware**
- **Flexible auf Prozeßveränderungen parametrierbare Systeme**

# Veränderungsprozesse bei der Neuausrichtung und Migration der gesamten Informatik-Organisation

Die Kombination zweier verzahnter Veränderungsprozesse ist notwendig:

- Neuausrichtung als strukturelle Erneuerung ermöglicht einen Quantensprung
- Migration als kontinuierliche Verbesserung reduziert Risiken



# ObjectEngineering Technologien: Was können wir gemeinsam tun?

---

- Auswahl, Anpassung und Einführung objektorientierter Vorgehensmodelle, Methoden, Richtlinien, Standards und Werkzeuge für zukunftsweisende Projekte
- Konzeption und Einführung objektorientierter Architekturen für integrierte Systeme
- Durchgängige objektorientierte Neuordnung, Migration und Optimierung von Geschäftsprozessen, zugehörigen Systemen und Informatik-Organisationen
- Gutachten, Analysen, Konzeptionen für innovative Systeme und Informatik-Strategien
- Objektorientierte Analyse und Entwurf (Entwurfsmuster) von komponentenbasierten Informationssystemen mit der UML (Unified Modelling Language)
- Software-Ergonomie: Gestaltung und Entwurf von interaktiven Systemen
- Training und Coaching, Projektmanagement, Qualitätssicherung, Wissensmanagement
- Wissenschaftliche Beratung und Begleitung in praktischen Informatik-Projekten

**Prof. Dr. Alfred Zimmermann**  
Object-Oriented Software Engineering

**Reutlingen University of Applied Sciences**  
Business Information Science  
Alteburgstr. 150  
72762 Reutlingen  
Tel. 07121-271-652; Fax. -683  
[alfred.zimmermann@fh-reutlingen.de](mailto:alfred.zimmermann@fh-reutlingen.de)