



Von Objekten

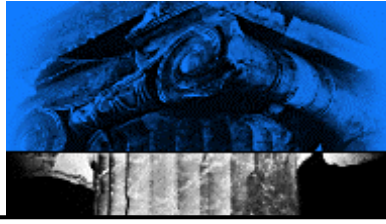
zu Geschäftskomponenten

Prof. Dr. Volker Gruhn

UNIVERSITÄT DORTMUND



adesso
the process company



Gliederung

Abgrenzungen von Begriffen

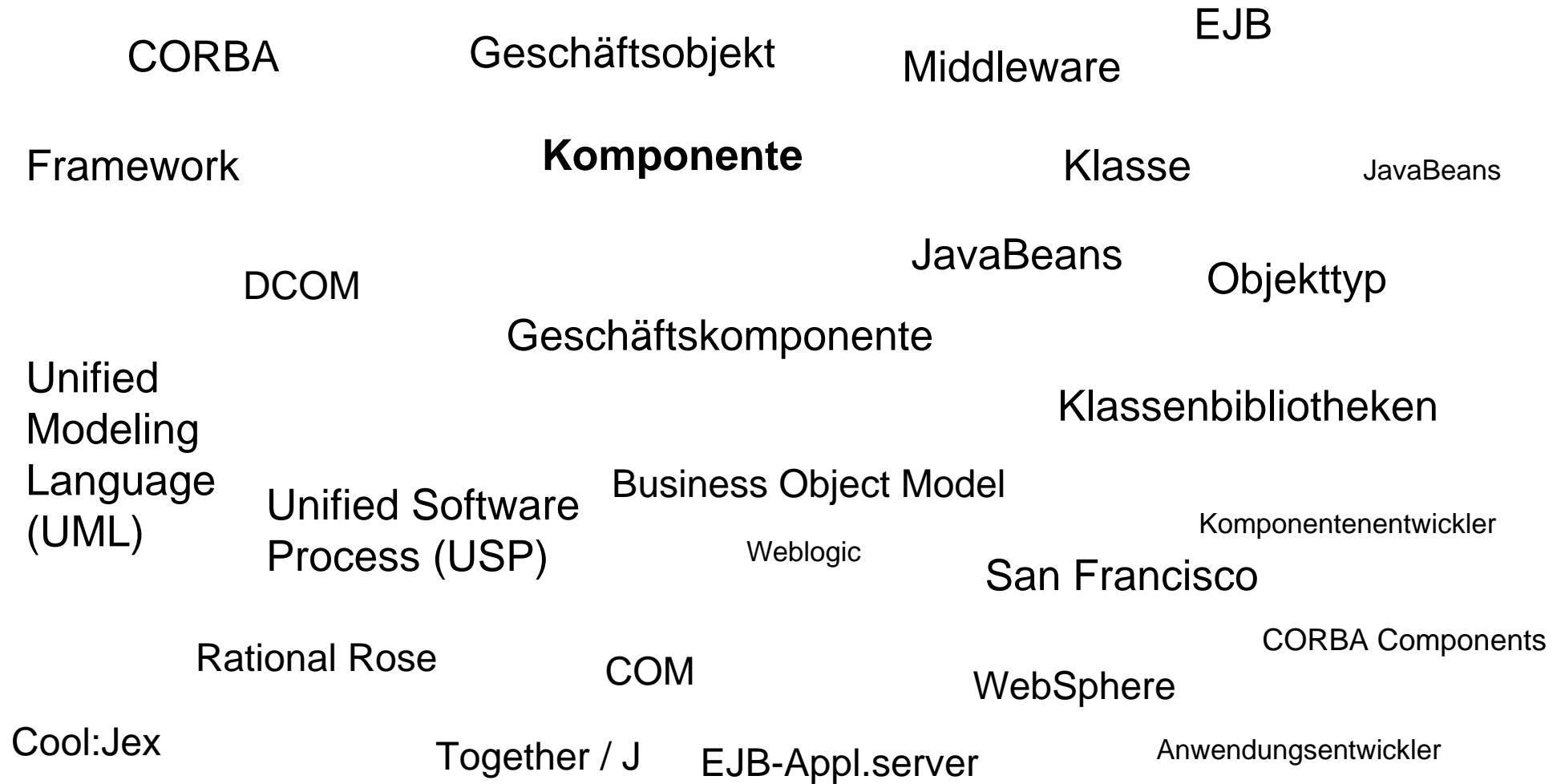
Szenarien: Auf dem Weg zu Geschäftskomponenten

Herausforderungen und Risiken

Ausblick



Worum geht es überhaupt?





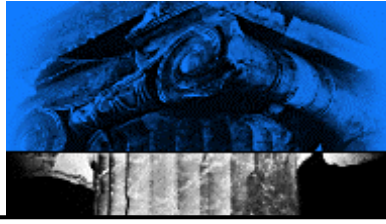
Aber das kennen wir doch schon!

Objektorientierung sorgt für

- Wiederverwendung
- wartbare Software
- austauschbare (weil gekapselte) Algorithmen
- inkrementelle, iterative Entwicklung

Komponenten sorgen für

- Wiederverwendung
- wartbare Software
- leicht austauschbare Einheiten auf Anwendungsniveau
- inkrementelle, iterative Entwicklung
- plattformunabhängige Einheiten
- kontextunabhängige Einheiten



Objekte versus

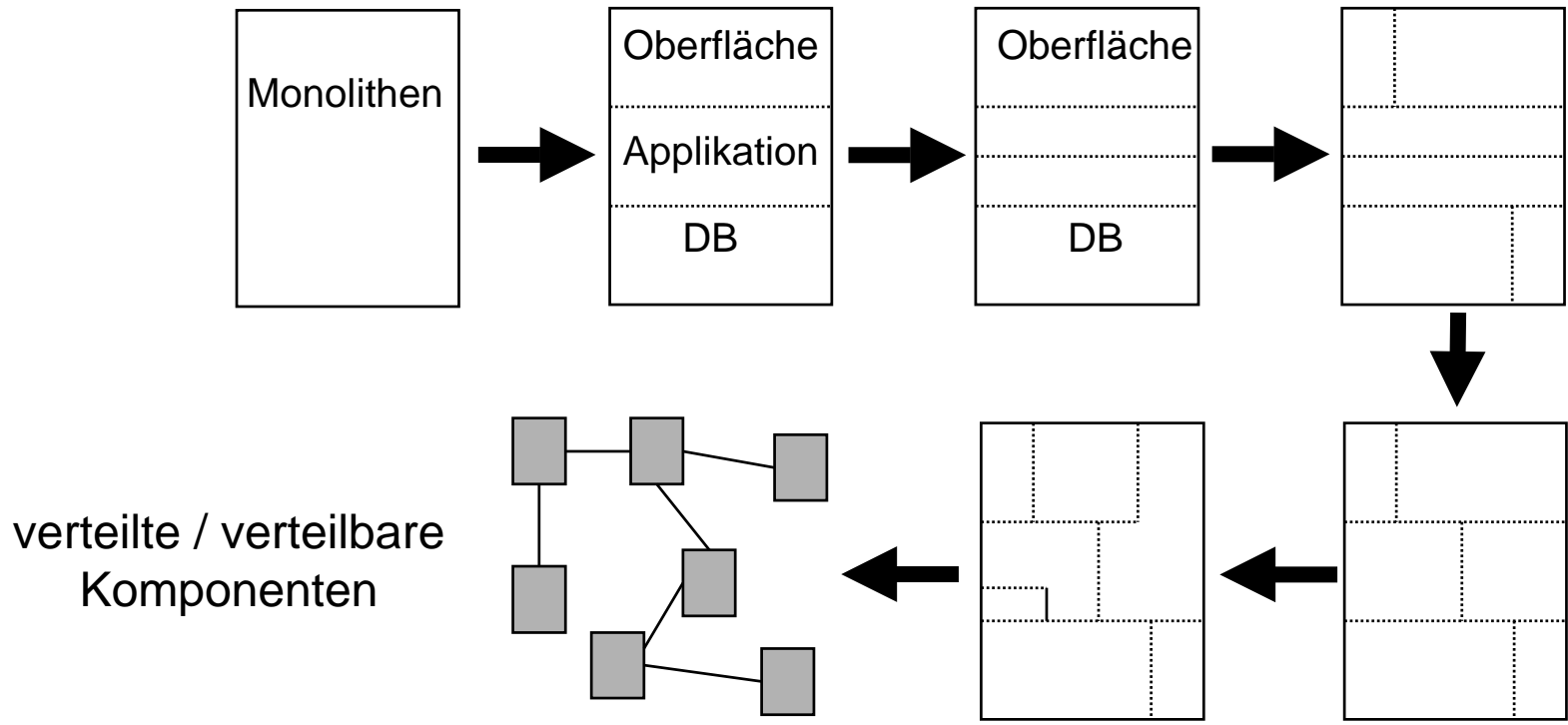
technische Komponenten

	Klasse/Objekt	Komponenten/Instanzen
...besteht aus...	Variablen und Methoden (Zustand und Verhalten)	Schnittstellen und deren Implementierung
programmiersprachen...	...abhängig	...unabhängig
Wiederverwendung	nur als Quellcode und meistens zusammen mit anderen Klassen (white-box reuse)	als eine fertige und ablauffähige Einheit (black-box reuse)
Kontextabhängigkeit	ja	nein
Sichtbarkeit	außerhalb eines Programm- kontextes unsichtbar	verteilt und für alle sichtbar



Trend 1:

Verteilung





Trend 2:

Abstraktion

Geschäftsobjekte
(LV-Antrag, Tarif)

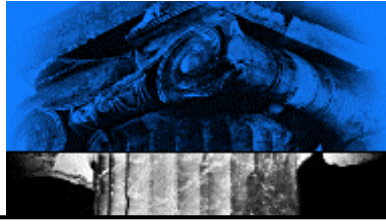
Komponenten (Adresse, Person)

Bibliotheken (Oberflächen, Listen)

prozedurale Konstrukte
(while, case, if)

Assembler und mnemotechnische
Abkürzungen (MV, ADD, GET)

Assembler (001001)



Terminologie

Geschäftsobjekte / Geschäftskomponenten / fachliche Komponenten

- fachlich bestimmte, zusammengehörende, vom Anwendungskontext abhängige Bausteine (Spezifikation)
- noch keine Komponenteneigenschaften wie Plattformunabhängigkeit, Kontextunabhängigkeit

technische Komponenten (oft nur Komponenten)

- Realisierung einer Geschäftskomponente unter Verwendung eines Komponentenmodells
- Komponenteneigenschaften!

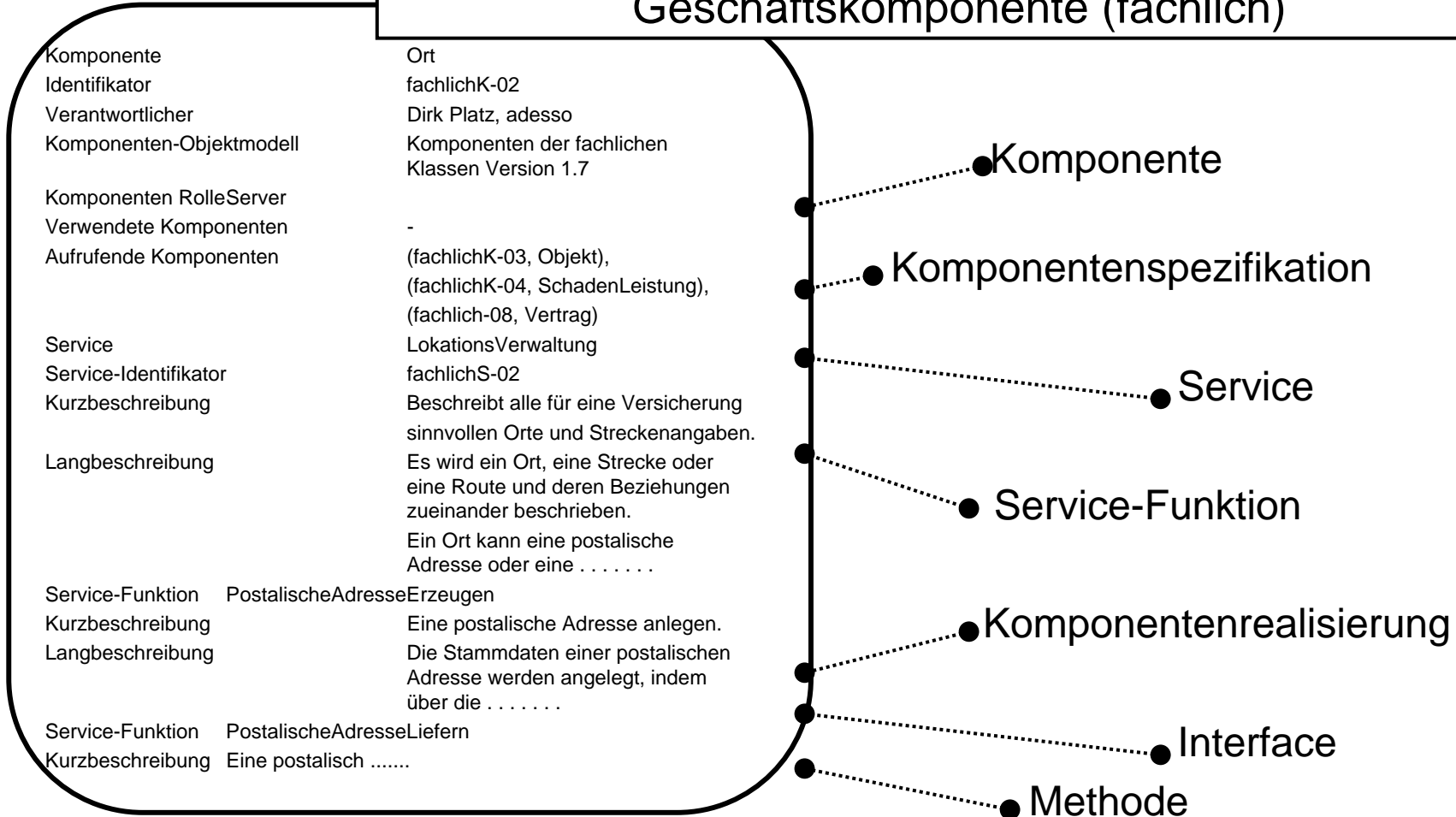
Objekte / Klassen

- durch Methoden gekapselte Daten / deren Schablonen
- Bestandteil von technischen Komponenten



Spezifikation einer

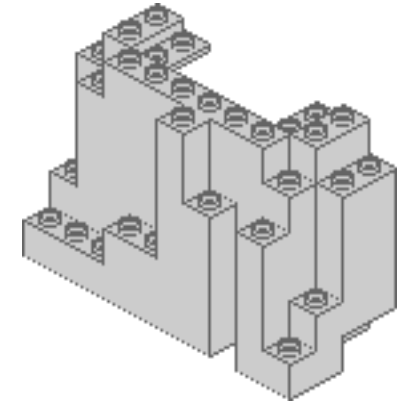
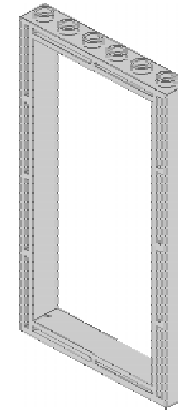
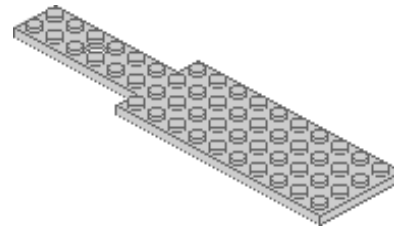
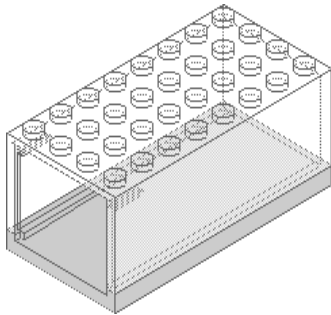
Geschäftskomponente (fachlich)





Der Charme:

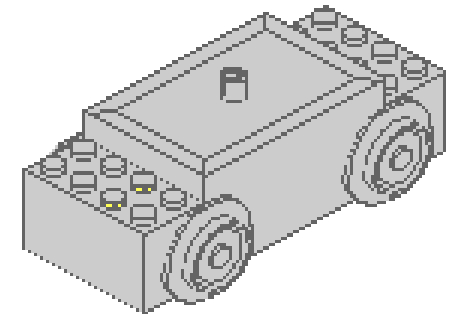
Die Legosteine-Metapher



- die einen bauen Legosteine
- die anderen bauen mit Legosteinen Modelle

oder

- die einen bauen Komponenten
- die anderen bauen mit Komponenten Anwendungen





Der Charme:

Das Ende der proprietären Welt !

- **Make or Buy or Wrap**
- **Write Once - Run Everywhere**
- **Trennung von Oberfläche und Funktionalität**



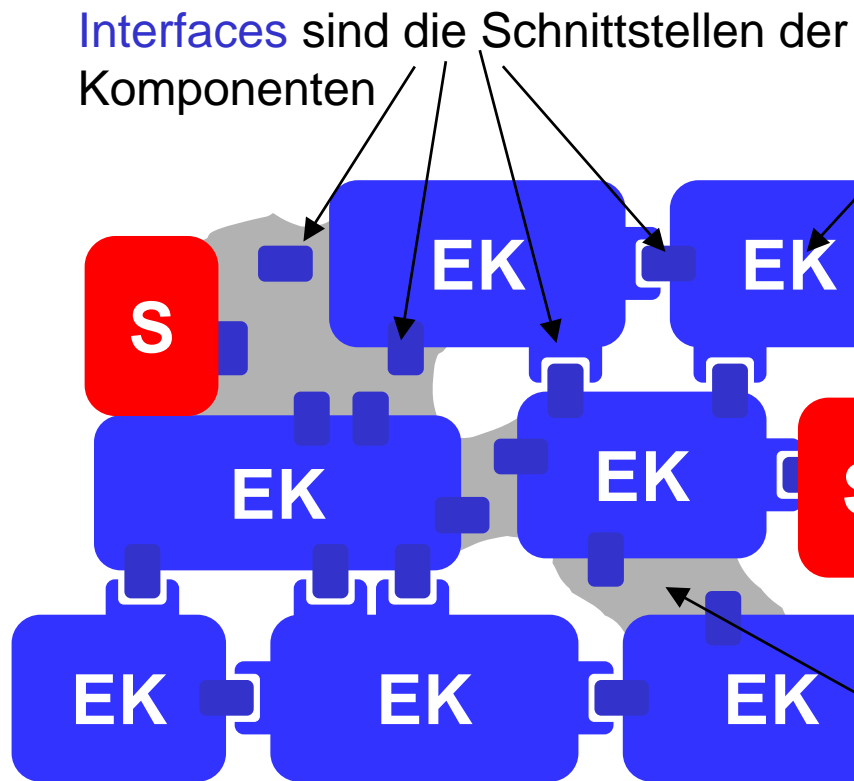
Client-Komponenten

- Oberflächen
- einfache Plausibilitäten
- keine Geschäftslogik
- keine Geschäftskomponenten
- Java Beans / COM

Server-Komponenten

- Geschäftskomponenten
- Abbildung der Geschäftslogik
- Entkopplung von fachlicher Funktionalität und technischem Rahmen
- Enterprise Java Beans /
CORBA-Objekte

Arten von Serverkomponenten



Entity-Komponenten

- repräsentieren die Daten
- kommunizieren mit anderen Komponenten
- werden als Entity Beans realisiert
- z.B: Vertrag, Partner, Schaden, Ereignis, Adresse

Service-Komponenten

- bilden Funktionen ab
- werden als Stateless Session Beans realisiert
- z.B: Drucken, Archivieren

Session-Komponenten

- verdrahten Teile von Prozessen
- werden als Stateful Session Beans implementiert

z.B: komplexe Dialoge, Ablaufsteuerung



Komponentenbasierte Softwareentwicklung

Vier Einstiegsalternativen

Abhängig von:

- der Ausgangssituation des Unternehmens
- der aktuellen Softwarelandschaft
- der Migrationsstrategie

deshalb:

- vier unterschiedliche Einstiege



Komponentenbasierte Softwareentwicklung

Vier Einstiegsalternativen

- **Der evolutionäre Einstieg**
- **Der spartenorientierte Einstieg**
- **Der e-Business - Einstieg**
- **Der architekturorientierte Einstieg**



Vier Einstiegsalternativen

Der evolutionäre Einstieg

- **Technisch: Wrapping der Altsysteme**
- **Komponentenschnittstellen werden auf der Basis „sichtbarer Objekte“ definiert**
- **Schrittweise „Standardisierung“ der Schnittstellen zwischen Systemen**
- **Zukunftssichere Nutzung dieser Schnittstellen für Erweiterungen (z.B. Internet-Anwendungen)**
- **Spätere Ablösung einfach möglich**



Vier Einstiegsalternativen

Der spartenorientierte Einstieg

- **Entwicklung einer einzelnen Anwendung**
- **Komponentenbasierte Entwicklung**
- **Bedarfsgetriebenes Wrapping der Schnittstellen**
- **Technologieerprobung**



Vier Einstiegsalternativen

Der e-Business-Einstieg

- **E-Business- und Internet-Anwendungen für unterschiedliche Benutzergruppen**
- **Integration mit existierenden Anwendungen**
- **Stabile, „wenige“ Komponentenschnittstellen auf existierenden Anwendungen**
- **Viele, stabile Frontends (Programme, Applets, Servlets)**



Vier Einstiegsalternativen

Der e-Business-Einstieg

Typische E-Business-Komponenten:

Content Management:

Präsentation

Commerce:

Shop, Zahlungstransaktion,
Autorisierung,
Authentifizierung

Community Support:

Chatting, Voting, Rating,
Dating, Trading, Tracking

**Customer Relationship
Management:**

Profilmanagement

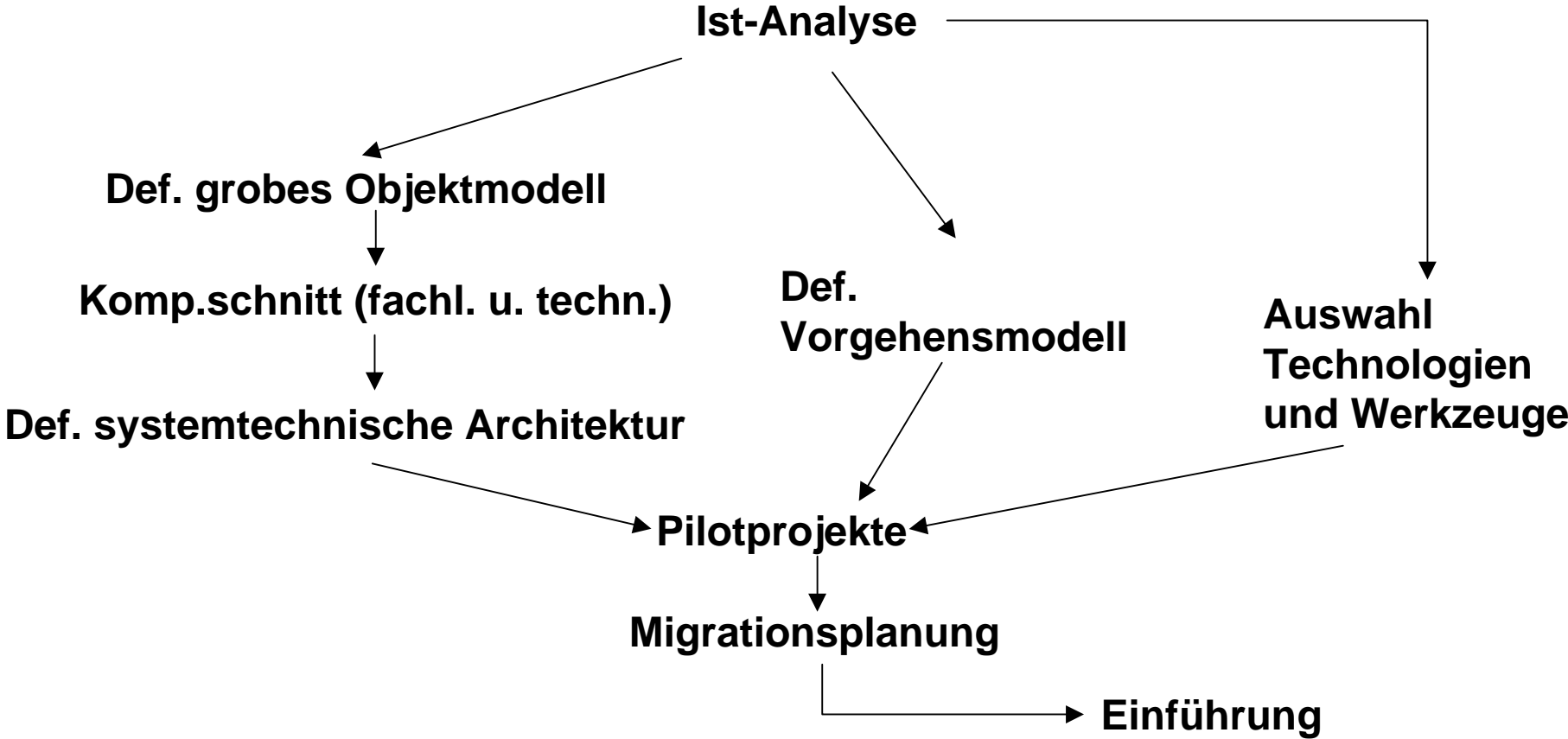
Connectivity:

Middleware, Workflow



Vier Einstiegsalternativen

Der architekturorientierte Einstieg





Komponentenbasierte Softwareentwicklung

Vier Einstiegsalternativen - Bewertung

- **Der evolutionäre Einstieg**
 - geringe Kosten
 - kein unmittelbarer Zusatznutzen
- **Der spartenorientierte Einstieg**
 - Produktivitätssteigerung in Entwicklung
 - Ausbildungsproblem
 - Risiko der falschen Komponenten
- **Der e-Business Einstieg**
 - ohnehin unverzichtbar
 - Erkenntniss „e-Business-Anwendungen sind Anwendungen“ als Voraussetzung
- **Der architekturorientierte Einstieg**
 - Nutzung bisheriger Architekturergebnisse (VAA!)
 - Investitionen in Architektur
 - geringes Risiko
 - bei Verknüpfung mit Pilotprojekten sehr effizient



Und was braucht

man nun alles?

Softwarearchitektur,
z.B. VAA, IAA

Komponentenmodell

sind
Grundlagen
für
Komponenten

sind
Grundlage
für

konkrete
Komponenten

komponentenbasierte
Anwendungen

müssen zueinander
passen

Framework

werden
benutzt
für

werden
betrieben
in

basieren auf

Entwicklungswerkzeuge

Middleware



Und was braucht

man nun alles?

UML-Modell + Schnitt
(orientiert an VAA)

(Enterprise)JavaBeans

sind
Grundlagen
für
Komponenten

sind
Grundlage
für

konkrete
JBs und EJBs

komponentenbasierte
Anwendungen

müssen zueinander
passen

KEIN BO Framework

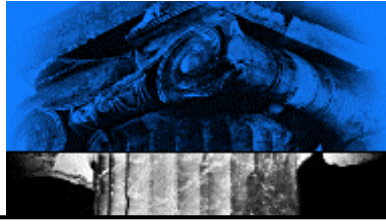
werden
benutzt
für

werden
betrieben
in

UML / Java IDE

EJB Appl.
Server

basieren auf



Und wie passt das alles

zur bekannten Welt?

Anwendungsbaustein:

Komponente

ABER: Komponenten sind plattformunabhängig

Dialogsteuerung / Dialogmanager:

Event-Handling-Mechanismen des Client-Komponentenmodells

ABER: wenn mehr vordefinierte Dialoglogik nötig, dann eigene Komponente „Dialogmanager“

Transaktionsmonitor:

Applikationsserver, Objekttransaktionsmonitor

ABER: zusätzliche, allerdings oftmals noch unreife Services

Datenbankzugriffsschicht:

Objektrelationales Mapping + anwendungsspezifische Ergänzungen



Herausforderungen auf dem Weg zu

Geschäftskomponenten

**Von Cobol / PL1 zu Komponenten bei gleichzeitigem Bewahren der
Betriebssicherheit und der Produktionserfahrung**

**Organisation der Entwicklung in Komponentenentwickler und
Anwendungsentwickler**

**Architekturmanagement
Der schmale Grat zwischen Dogma und Pragmatismus**

Instrumentierung mit best-of-class Werkzeugen



Risiken auf dem Weg zu

Geschäftskomponenten

Nicht-Berücksichtigung von vorliegenden Ergebnissen wie VAA

Unterschätzung der Auswirkungen neuer Technologien

Beispiel Business Object Framework

- Einarbeitung
- Konfigurationsmanagement / Releasewechsel
- Abhängigkeit von Frameworks, schlimmstenfalls noch von undokumentierten, selbstgestrickten

Mangelnder Mut zu Abstraktionen

Fehlende Management-Unterstützung

Unendliche Auswahlprozesse / Werkzeugzentriertheit

Inkompatible Technologien



Ausblick

Geschäftsobjekte und Workflowmanagement

Von komponentenbasierten Softwarelandschaften zu e-business fähigen, komponentenbasierten Softwarelandschaften oder

Wie bekomme ich meine Anwendungen ins WWW?

D.h. aber auch

Wie mache ich meine Anwendungen 7+24H-fähig?

Wie verwende ich die gleichen
Geschäftskomponenten auf dem Weg vom
AO-Vertrieb über den Maklervertrieb zum
Direktvertrieb?



Worum ging es überhaupt?

Begriffe
Geschäftsobjekt / Geschäftskomponente
Komponente
Klasse / Objekttyp

Middleware
CORBA
EJB-Appl.server
WebSphere
Weblogic
DCOM
OTM

Vorgehen
Prozesse
USP
Catalysis
SELECT
A-Entwickler
K-Entwickler

Wiederverwendung
Klassenbib
Framework
Business Object Model
San Francisco

Spezifikation
UML
Rational Rose
Together / J
Cool:Jex

Komp.modell
JB
EJB
COM
CORC



Innovationsbericht

Komponentenbasierte Softwareentwicklung - Von der Vision zur Architektur

a d e s s o
Innovationsberichte

MEMO Nr. 02

Komponentenbasierte
Softwareentwicklung -
Von der Vision zur Architektur

Internes Memorandum der
a d e s s o
Beratungsgesellschaft für
Software-Prozeß-Management mbH
Stockholmer Allee 24

D-44269 Dortmund

ISSN 1437 - 94 81



Danke

Für Ihr Interesse !

Prof. Dr. Volker Gruhn
a d e s s o
Beratungsgesellschaft für
Software-Prozeß-Management mbH
Stockholmer Allee 24
D-44269 Dortmund

Tel.: 0231-9309330

Fax: 0231-9309331

gruhn@adesso.de

www.adesso.de