

# Engineering IT-basierter Services

Prof. Dr. Klaus-Peter Fährnich

Praxisbeispiel: Engineering multimedialer Online-Services

19.05.2009

## Engineering IT-basierter Dienstleistungen

1. Typologisierung von Dienstleistungen
2. Grundlagen des Service Engineering
3. Vorgehensmodelle
4. Plattformstrategie: Produktmodelle und Modularisierung
5. Methoden und Werkzeuge I
6. Methoden und Werkzeuge II
7. Methoden und Werkzeuge III
8. Methoden und Werkzeuge IV
9. Werkzeuganwendung I
10. Werkzeuganwendung II
11. Zusammenfassung Werkzeuge
12. Service-Technologien
13. Kundenintegration und Kundenmanagement
14. Standardisierung im Dienstleistungsbereich
15. Praxisteil I
- 16. Praxisteil II**

## Einleitung



- die Erstellung semiprofessioneller Videos ist sehr einfach geworden, benötigt werden nur
  - DV-Camcorder
  - PC/Laptop mit Firewire (IEEE 1394)
  - Software
- zunehmende Informationsvielfalt
- Suche/Finden von Informationen wird schwieriger
- strukturierte Informationsablage, Volltextsuche, Schlagwortsuche, usw. sind nicht effektiv genug
- effizientere Möglichkeit:
  - profilbasierte Informationsversorgung

## Einleitung



- Vorarbeiten:
  - Entwicklung einer integrierten Methodik zum Engineering multimedialer Online-Dienste
  - Definition eines geeigneten Entwicklungsprozesses
  - Entwicklung und Evaluation eines entsprechenden Frameworks
- verwendete Technologien/Methoden:
  - UML
  - Service-Engineering nach Shostack und Kingman-Brundage

# Dienstleistungstypologie



<b>Zielgruppe</b>	<b>Business to Business</b>	<p><b>Kundenintegrative Dienstleistungen</b></p> <ul style="list-style-type: none"> <li>• Business TV</li> <li>• Informationsvideos für den Vertrieb</li> <li>• Multimedialer Online-Immobilienmakler</li> </ul>	<p><b>Wissensintensive Dienstleistungen</b></p> <ul style="list-style-type: none"> <li>• Wissensdatenbank im Maschinenbau</li> <li>• Problemlösungsvideos für Call Center/First Level Support</li> </ul>
	<b>Business to Consumer</b>	<p><b>Personalisierte Massendienstleistungen</b></p> <ul style="list-style-type: none"> <li>• Personalisiertes TV</li> <li>• Multimedialer profilbasierter Online-Touristenführer</li> <li>• Multimedialer Museumführer</li> </ul>	<p><b>Produktunterstützungsnetzwerke</b></p> <ul style="list-style-type: none"> <li>• Personalisierte Online-Hilfe in komplexen Software-Produkten</li> <li>• Entwicklernetzwerke</li> <li>• Personalisierte Online-Akademie</li> </ul>
		<b>niedrig</b>	<b>hoch</b>
<b>Variantenvielfalt</b>			

**Serviceprofil**

Produkttyp	➤ Individualprodukt	Baukastenprodukt	Standardprodukt
Haupteinsatzfaktoren	➤ Menschliche Arbeit	Maschinen, Geräte	Informationssysteme
Hauptobjekt der DL	➤ Kunde	Materielle Objekte	Immaterielle Objekte
Produktumfang	➤ Einzelleistung	Leistungsbündel	
Produktart	➤ konsumentenbezogen	unternehmensbezogen	
Erbringungsdauer	➤ kurz (< 1 Tag)	mittel (< 1 Monat)	lang (> 1 Monat)
Interaktionsart	➤ angebotsorientiert	nachfrageorientiert	getrennter Ort
Kundenrolle	➤ Akteur	Zuschauer	ohne direkte Beteiligung

## Anforderungen



- hohe Komplexität und Variantenvielfalt bei multimedialen Services
- daher ist es vorteilhaft, ein Anwendungsframework zu definieren
  - ist ein generisches objektorientiertes System,
  - aus dem kunden- oder produktspezifische Anwendungen erstellt werden können
- die integrierte Entwicklungsmethodik legt fest:
  - Entwicklungsprozess für Anwendungsframework und die zu realisierenden Online-Services
  - Reihenfolge/Simultaneität der Entwicklung von Framework und Service bzw. Pilot-Service
- SW-Unterstützung kann durch vorhandene oder zu erstellende Softwaresysteme gegeben sein

## Anforderungen an das Anwend.-Framework



- Objektorientierung
- wünschenswert: Komponentenorientierung
- Reuse auf hoher Granularitätsebene
- Ausreichende Dokumentation zur Unterstützung von Reuse
- Erweiterbarkeit
- Integrationsfähigkeit in vorhandene heterogene Umgebungen
- iterative Optimierung
- Anpassbarkeit an sich ändernde Anforderungen
- schnelle und kostengünstige Realisierung kunden- und produktspezifischer Anwendungen basierend auf dem Framework
- Erstellung einer langfristig tragfähigen Architektur



## Anforderungen an das Engineering der DL

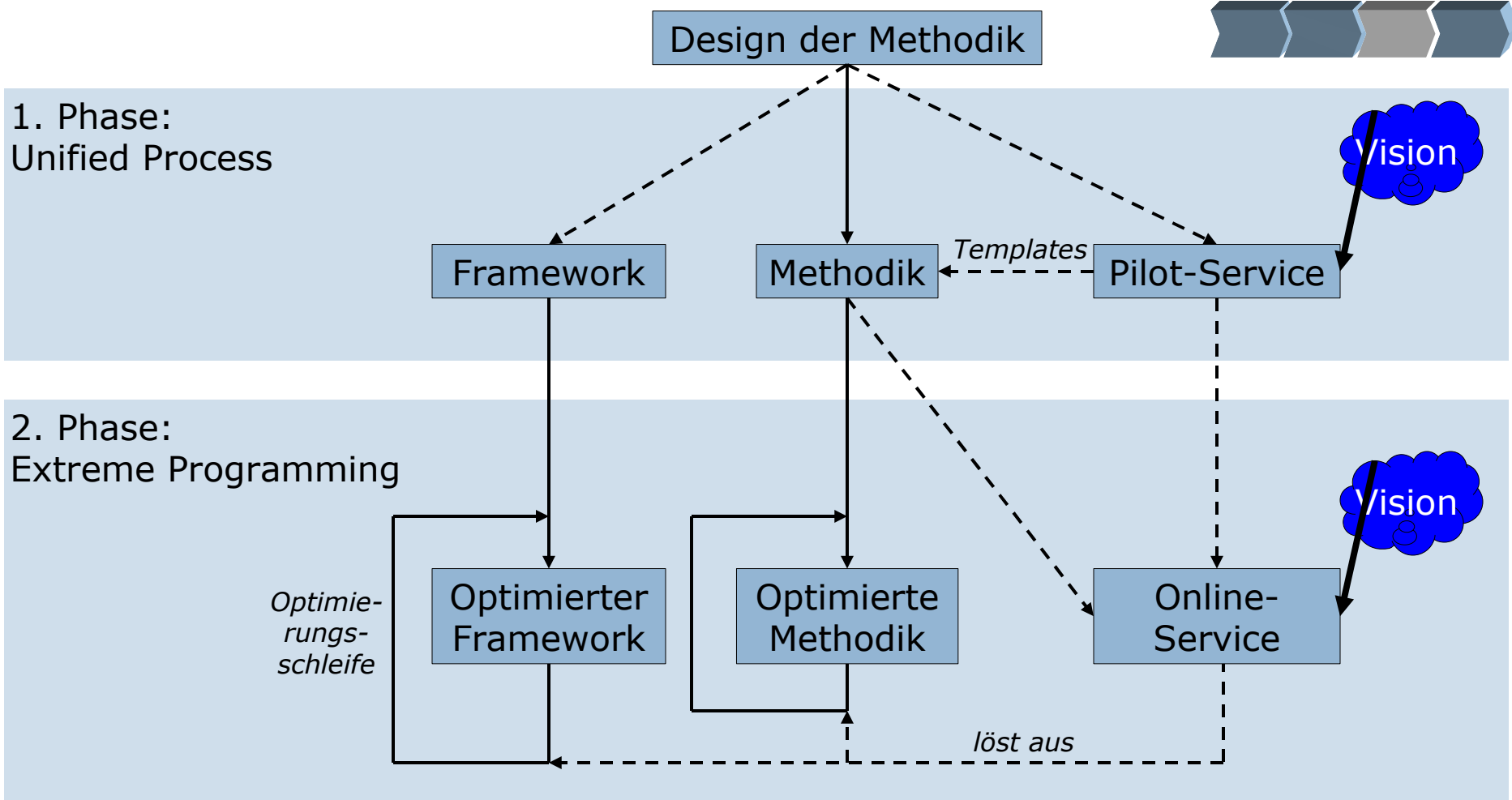


- Nutzung des objektorientierten Paradigmas
- fein granulare Modularität
- minimale Beschreibungskomplexität
- Erweiterbarkeit zur Realisierung neuer und Erweiterung bestehender Produkte
- Weitgehende Nutzung offener Standards, Interoperabilität
- Rapid Prototyping zwecks vorzeitiger Abstimmung mit dem Kunden
- Ermöglichen ganzer Produktlinien durch gemeinsame Spezifikation oder durch Beschreibung mit einheitlichen Anforderungsattributen
- Abdeckung des gesamten Produktlebenszyklusses (Forschung und Entwicklung, Marketing, Produktion und Management)
- Unterstützung unterschiedlicher Produktsichten
- Entwicklungsprozess geringer Komplexität

## Erfüllung der Anforderungen

Legende: ✓ gut/hoch    ○ mittel    ✘ schlecht/gering	Wasserfall- modell	Unified Process	Extreme Programming
Kriterium			
Objektorientierung	○	✓	○
Eignung für Wiederverwendung	○	✓	○
Erstellung einer Entwicklerdokumentation	✓	✓	○
Liefert tragfähige Softwarearchitektur	✓	✓	○
Eignung für große Softwaresysteme	○	✓	✘
Rechtzeitiges Erkennen von Risiken	○	✓	○
Frühzeitiges und häufiges Testen	✘	✓	✓
Frühzeitige Integration	✘	✓	✓
Iterative Vorgehensweise	✘	✓	✓
Eignung zur Entwicklung ganzer Produktlinien	○	✓	○
Zeitliche Planbarkeit	✓	✓	○
Geringe Beschreibungskomplexität	✘	○	✓
Einbindung der späteren Nutzer	○	○	✓
Reaktion auf ändernde Anforderungen	✘	○	✓
Schnelle Verfügbarkeit von Prototypen	✘	○	✓
Werkzeugunterstützung	✘	○	○
Geringer Projektmanagementaufwand	✘	✘	✓

# Neue zweiphasige Entwicklungsmethodik



- 1. Phase: Simultaneous Engineering von Framework und Pilot-Service
- 2. Phase: Framework-basiertes Service-Engineering

## Neue zweiphasige Entwicklungsmethodik (2)



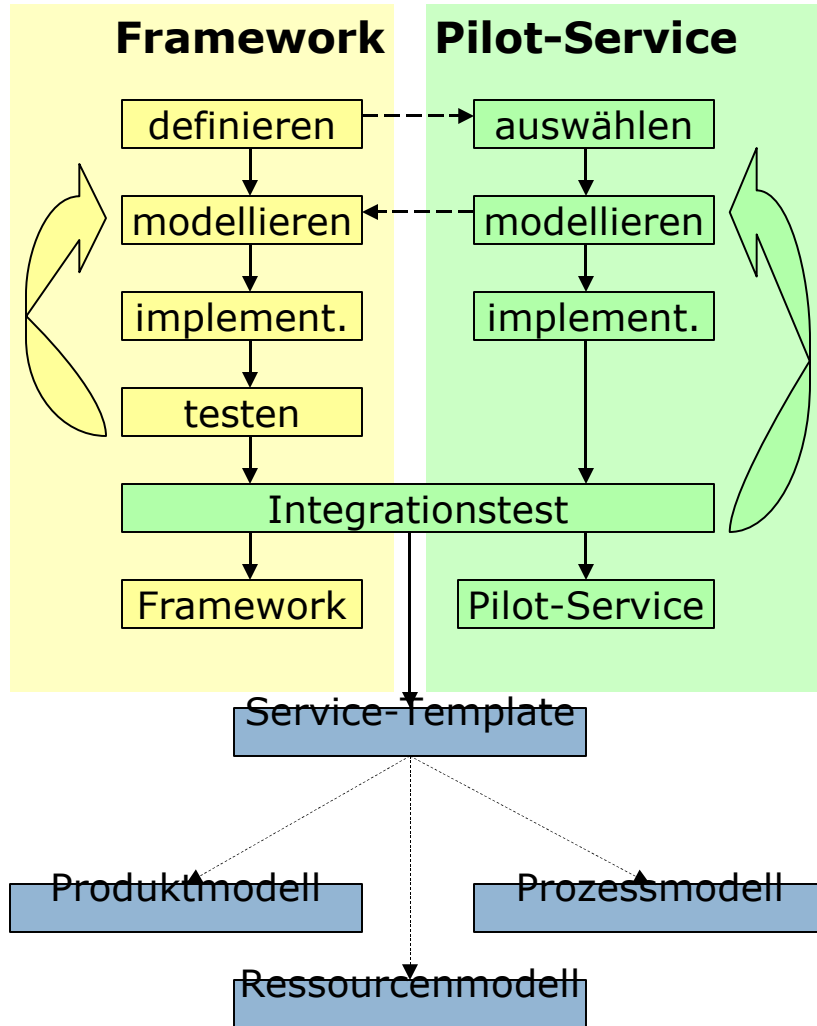
- 1. Phase:
  - Anlehnung an das Simultaneous Engineering aus dem Maschinenbau
  - gleichzeitige Entwicklung von Produkten (Online-Service) und Produktionsstätten (Framework und Service-Templates)
- 2. Phase:
  - Entwicklung weiterer Services
  - Erweiterung und Optimierung des Framework
  - Co-Engineering von Software und Services

## Vorteile dieses Prozesses

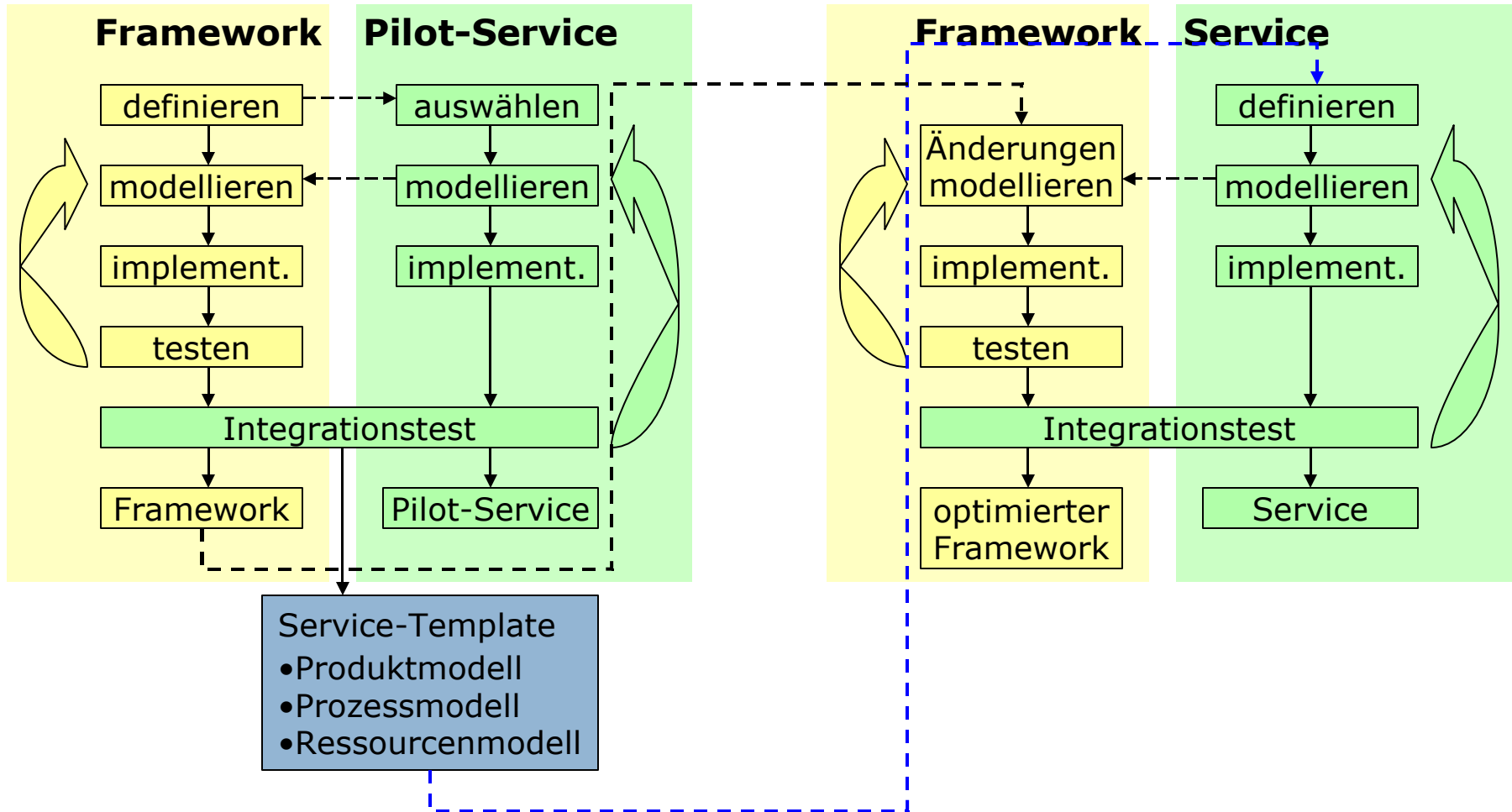


- schnelle Entwicklung des Pilot-Services und weitere Online-Services
  - frühe Verfügbarkeit von Demonstratoren für die Nutzer bzw. Kunden
  - frühes Feedback von den Nutzern bzw. Kunden
  - frühe Verfügbarkeit der fertig gestellten Online-Services
- bessere Abstimmung zwischen den Entwicklern von Framework und Services
  - schnelle Verfügbarkeit erweiterbarer, korrigierter und optimierter Framework-Versionen für die Service-Entwickler
  - Framework-Entwickler erhalten schneller Rückmeldung von den Service-Entwicklern über auftretende Probleme mit dem Framework
  - Verringerung des Änderungsaufwandes an Framework und Online-Services
  - rechtzeitiges Erkennen kritischer Pfade

# Erarbeitung der Modelle



# Erarbeitung der neuen Services



## Implementierung des Pilot-Service



- als Pilot-Service wurde Business-TV ausgewählt
  - traditionell nicht personalisiert und feste Sendeabfolge
- Implementierung als Online-Service
  - Bereitstellung multimedialer Online-Services
  - mit vorkonfigurierten Profilen
  - Filterung relevanter Beiträge für den Interessens- und Aufgabenbereich des jeweiligen Mitarbeiters
- stellt damit eine typische Anwendung für das zu entwickelnde Anwendungsframework dar



## Implementierung des Pilot-Service



### Charakteristika des Business-TV Pilot-Service

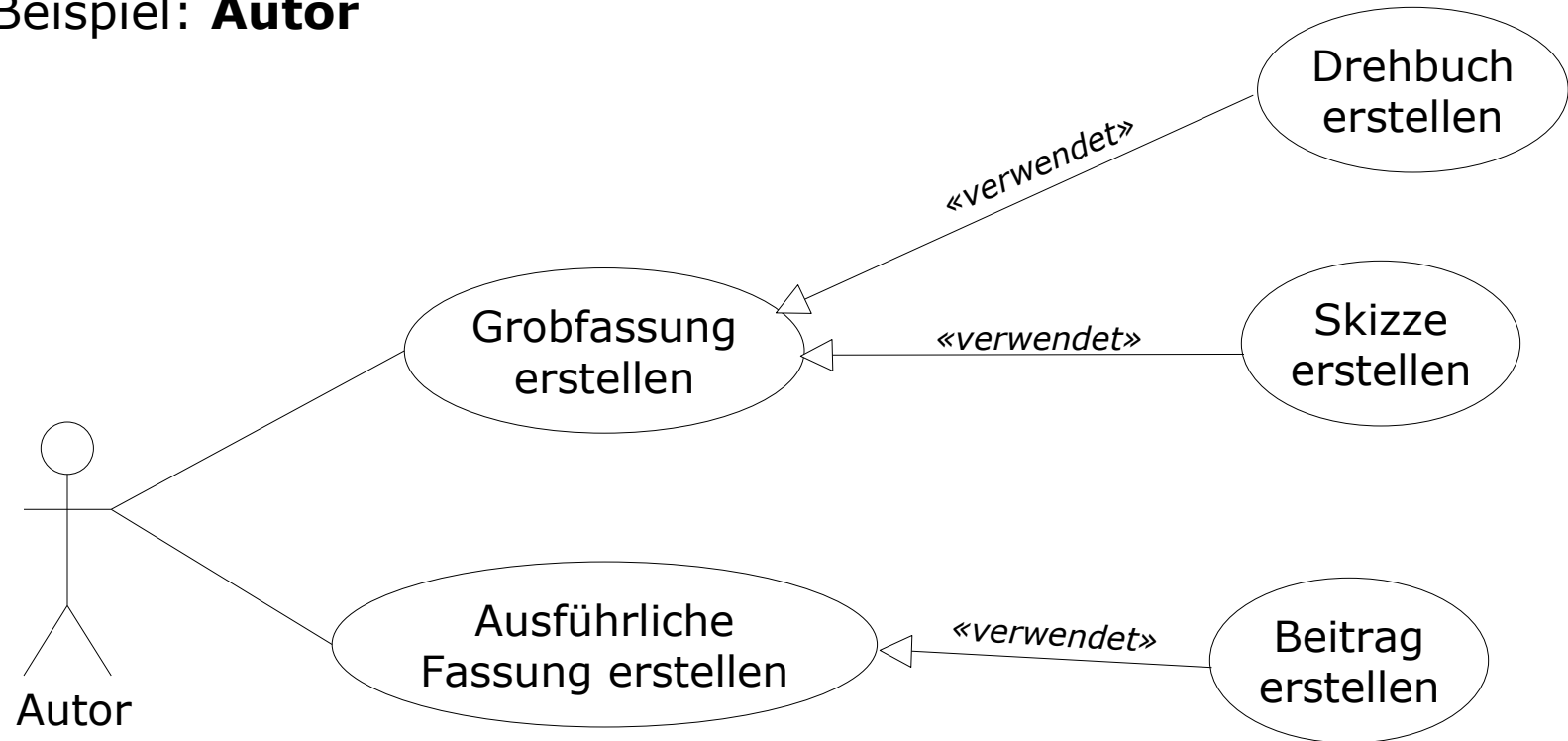
- vorkonfigurierte Aufgabenprofile und persönliche Interessensprofile
- gezielter Abruf von relevanten Videos
- jedes Video wird mit Vorschaubild und Überschrift in einer Übersicht dargestellt
- Einsatz auf Web-Browser auf PCs und PDAs
- dynamische bandbreiten- und ressourcenangepasste Kompression der Videos zur optimalen Darstellung unterwegs und im Büro
- optionale VPN-Verbindung
- Referenzpunkte (Bookmarks) können an beliebiger Stelle innerhalb Videos gesetzt werden
- Feedback zu den Videos möglich
- externe Quellen (z.B. Wirtschaftsnachrichten der Branche) einbindbar

**Implementierung des Pilot-Service**

<b>Akteur</b>	<b>Verantwortlichkeiten</b>
<i>Autor</i>	<ul style="list-style-type: none"> <li>• Grobfassung erstellen (Skizze/Drehbuch)</li> <li>• Ausführliche Fassung erstellen (Texte, Bilder, Audio, Video)</li> </ul>
<i>Redakteur</i>	<ul style="list-style-type: none"> <li>• Texte, Bilder, Audio, Video nachbearbeiten</li> <li>• Sprache, Audio, Video segmentieren (schneiden)</li> <li>• Inhalt kategorisieren und verschlagworten</li> </ul>
<i>Konsument (Arbeitnehmer)</i>	<ul style="list-style-type: none"> <li>• Registrieren, einloggen und ausloggen</li> <li>• Vordefinierte Aufgabenprofile nutzen</li> <li>• Interessenprofile erstellen, optimieren, ändern, auswählen und löschen</li> <li>• Beiträge suchen und abrufen</li> <li>• Benutzerrückmeldung zum Video abgeben</li> <li>• Referenzpunkte erstellen, abrufen und entfernen</li> </ul>
<i>Benutzer-Administrator</i>	<ul style="list-style-type: none"> <li>• Benutzer anlegen, editieren, entfernen und Aufgabenprofilen zuordnen</li> <li>• Aufgabenprofile erstellen, ändern, optimieren und löschen</li> </ul>

## Implementierung des Pilot-Service

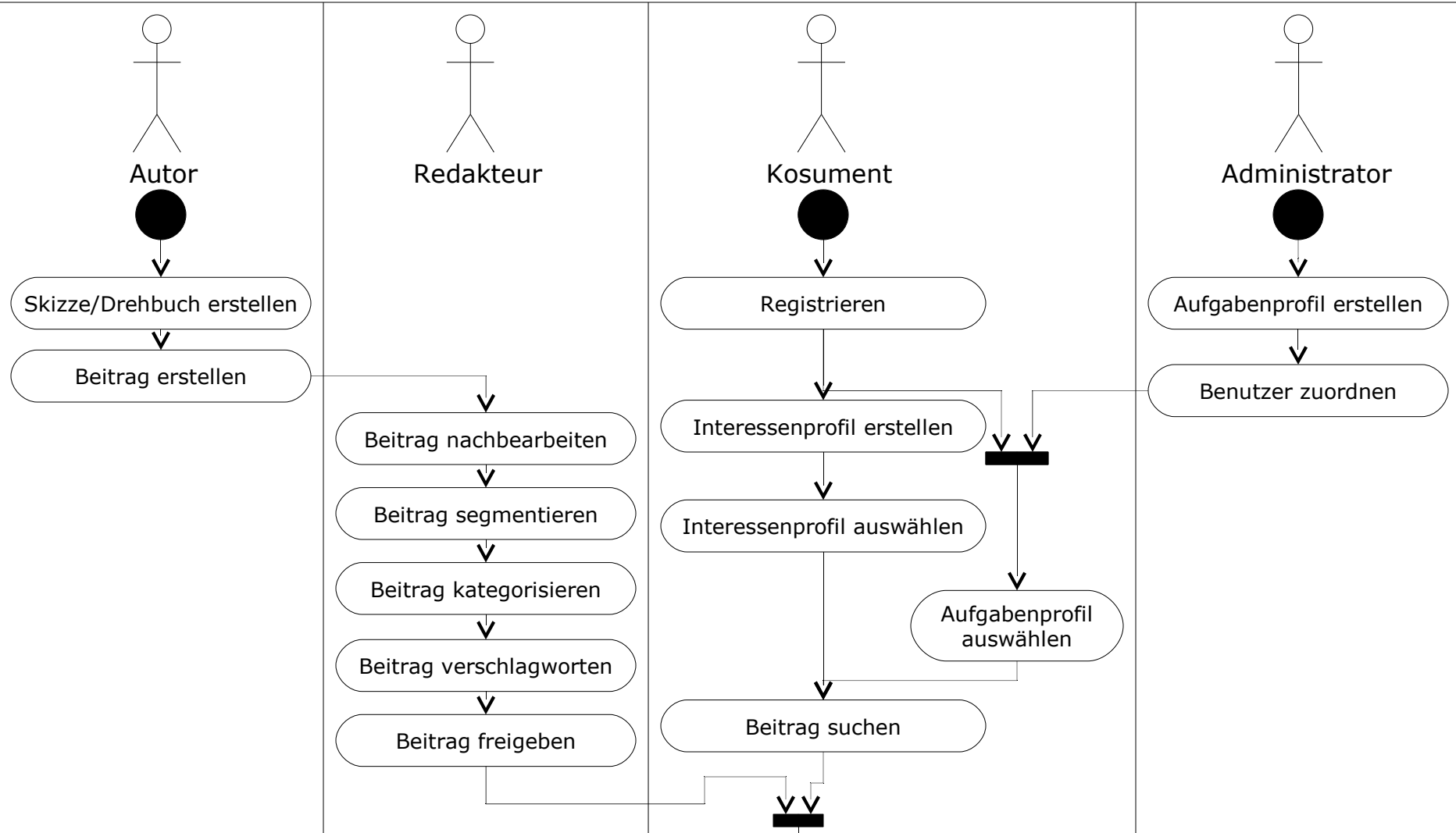
- die Use Cases der Akteure wurden mit UML modelliert
- Beispiel: **Autor**



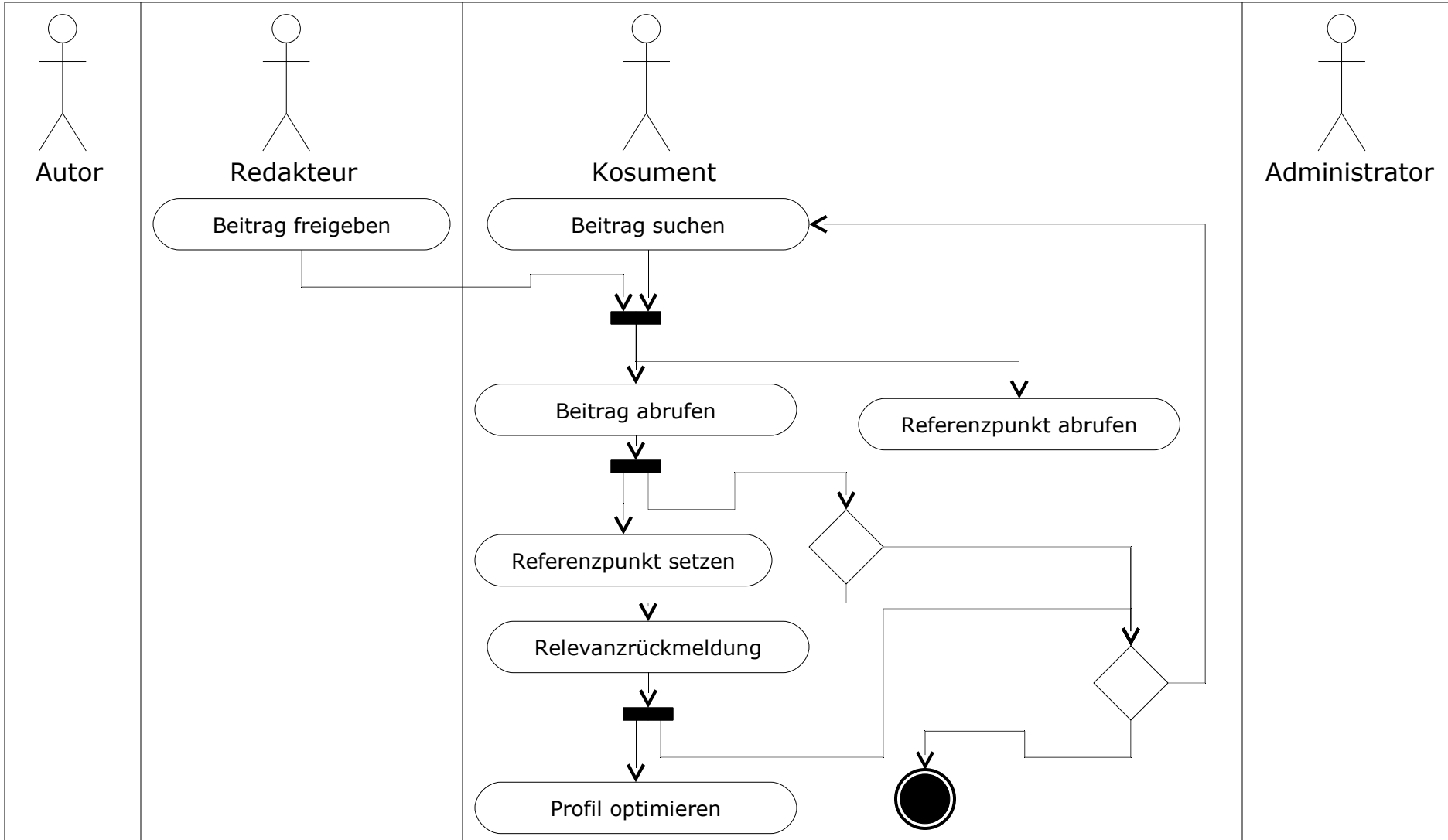
# Implementierung des Pilot-Service



- Prozessmodellierung mit *swimlanes*



# Implementierung des Pilot-Service



## Betreibermodell und Marketing-Konzept




- als Betreiber kommen in erster Linie große Unternehmen beliebiger Branchen in Frage, die damit ihre Mitarbeiter auf dem neuesten Stand halten wollen
- es wird eine eigene Kerngruppe benötigt, die für Inhalt und Technik verantwortlich ist
- Ideen(-skizzen) können von allen Mitarbeitern eingebracht werden
- Zugriff wird auf das Intranet beschränkt
- ein unternehmensübergreifender Dienst wäre denkbar, z.B. über Unternehmerverbände oder Industrie- und Handelskammern oder auch Gewerkschaften

## Entwicklung des Framework



- Framework wird in der 1. Phase zusammen mit dem Pilot-Service entwickelt
- dabei wird auf die Universalität (im Rahmen der generellen Spezifikation) geachtet
  - Abstrahierung bestimmter Begriffe und Funktionen, z.B. ersetze *Arbeitnehmer* durch *Konsument*
  - Erweiterung des Frameworks durch Funktionen, die evtl. in späteren Services auch noch gebraucht werden könnten

## Entwicklung des Framework

- die wesentlichen Funktionalitäten des Frameworks sind
  1. Content Management
    - Verwaltung von multimedialem Inhalt in Ablagehierarchien
    - Zuordnung von Inhaltskategorien an die einzelnen Videos
    - Zuordnung von Kurzbeschreibungen und Vorschau-Bildern zu den einzelnen Videos
  2. Benutzerverwaltung und Aufgabenprofilverwaltung
    - Anlegen, Editieren und Löschen von Benutzern
    - Anlegen, Editieren und Löschen von Aufgabenprofilen
    - Zuordnung von Benutzern zu Aufgabenprofilen
    - Authentifizierung von Benutzern
  3. Konfiguration und Optimierung von Interessensprofilen einzelner Nutzer
    - Anlegen, Editieren und Löschen von Interessensprofilen
    - Optimierung der Interessensprofile durch Benutzerrückmeldung zu den Videos



## Entwicklung des Framework



### 1. Personalisierter und aufgabenspezifischer Content

- Abruf von personalisierten Inhalten zu einem ausgewählten Interessensprofil
  - sequentielles Abspielen von Beiträgen zu einem ausgewählten Interessensprofil
  - Abruf von personalisierten Inhalten zu einem ausgewählten Aufgabenprofil
  - sequentielles Abspielen von Beiträgen zu einem ausgewählten Aufgabenprofil
  - Setzen von Referenzpunkten an den Beginn oder an beliebige Stellen eines Videos
  - Schnelles Wiederabrufen eines Videos durch Anwahl eines Referenzpunktes
- 
- das Redaktionssystem ist nicht Bestandteil des Frameworks, da es hinreichend leistungsfähige kommerzielle Produkte gibt, die diese Aufgabe gut erfüllen

## Generalisierung der Modelle


- die für den Pilot-Service verwendeten Modelle wie
  - Prozessmodell
  - Produktmodell
- können in generalisierter Form auch für die Entwicklung weiterer Services verwendet werden
- für die rollenspezifische Verteilung der Rechte wird eine Matrix entworfen, die jeweils angibt, wofür welcher Akteur create, read, write oder delete-Rechte benötigt



## Verwendete Technologien

- auf der **Client-Seite** werden durch die heterogenen Zugriffsmöglichkeiten verschiedene Technologien benötigt
- offene Internet-Standards werden bevorzugt, da dadurch eine aufwändige Softwareinstallation und -anpassung entfällt
- benötigt werden im groben nur
  - Web-Browser und
  - Video-Player
- Video-Auflösung und -komprimierung werden dynamisch an das jeweilige Endgerät angepasst

## Verwendete Technologien

- 
- auf der **Server-Seite** werden ebenfalls offene Standards bevorzugt eingesetzt, da somit die Anforderungen bzgl. Modularisierung und Interoperabilität am besten eingehalten werden können
  - Komponentenmodell: Enterprise Java Beans (EJB)
  - Präsentationsebene: Java Server Pages (JSP) und Servlets
  - Datenaustauschformat: XML (mit XSLT für Transformationen)
  - externe Schnittstellen: Web Services mit SOAP/WSDL
  - Application-Server: JBoss
  - Datenbanksprache: ANSI SQL92 (an Oracle 9i)

# Fünfschichtenarchitektur

**Clientschicht**

- Web-Browser, PDA
- Client-Software

} Schlanker Client

**Präsentations-  
schicht**

- GUI-Adaption
- Präsentationslogik

**Anwendungs-  
schicht**

- Anwendungsobjekte
- Anwendungslogik (Algorithmen)

} Application-Server,  
Anwendungs-  
Komponenten**Integrations-  
schicht**

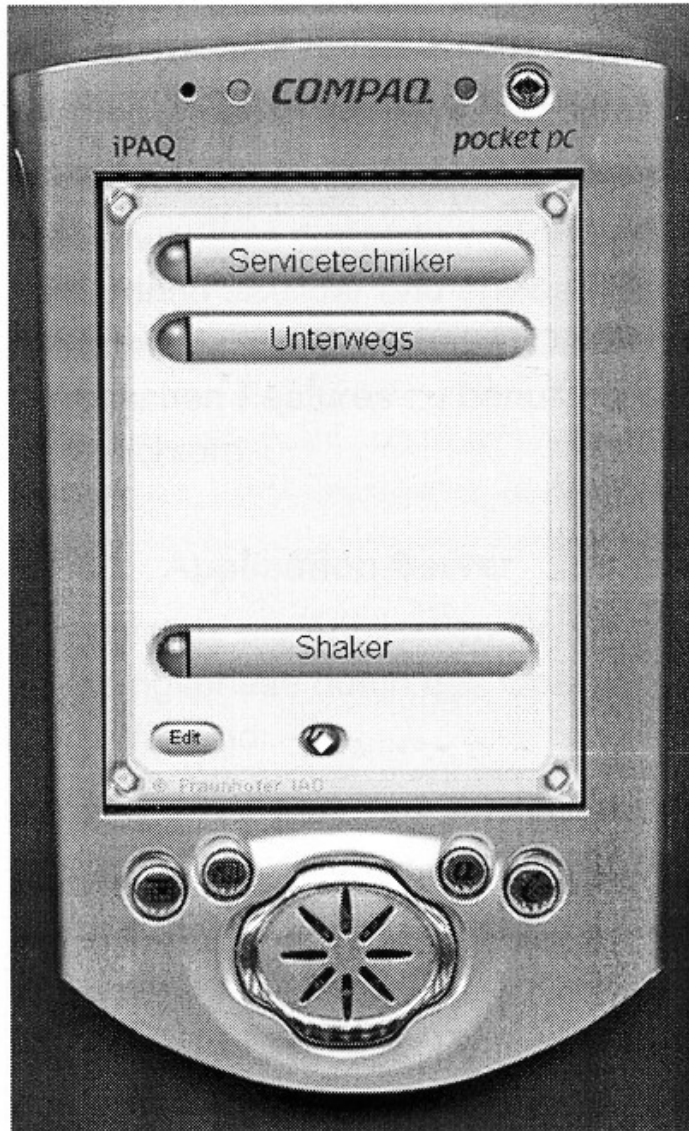
- Ressourcenankopplung  
(Datenbanken, Legacy-Anwend.)

**Ressourcen-  
schicht**

- Datenbanken
- Systemdienste

} Bestehende Software

# Screenshots (PDA-Version)



# Screenshots (PC-Browser)

