



Ludwig-  
Maximilians-  
Universität  
München



Lehr- und Forschungseinheit für Programmierung und Softwaretechnik

---

**Vorlesung am 5. Mai 2009**

Serviceorientiertes E-Government

---

## **Zentrale Probleme in komplexen IT-Landschaften**

**Dr. Frank Sarre**

Lehrbeauftragter der LMU München

# Typische Aufgabengebiete in einem Unternehmen

## Einkauf

Lieferanten  
Anfragen  
Bestellung  
Terminmanagement

## Verkauf

Kunden  
Angebote  
Rechnungen  
Terminmanagement

## Lager

Material  
Fertigwaren  
Chargenverwaltung  
Verfallsüberwachung

## Logistik

Flottenplanung  
Kapazitätsplanung  
Terminmanagement  
Lieferoptimierung

## Konstruktion

Produktplanung  
Entwicklung  
Musterbau

## Produktion

Produktionsplanung  
Steuerung  
Einzelfertigung  
Serienfertigung

## Dienstleistung

Personalverwaltung  
Einsatzplanung  
Abrechnung

## Projekt- Management

Zeitplanung  
Ressourcenplanung  
Überwachung

## Buchhaltung

Externes / internes  
Rechnungswesen  
Anlagenbuchhaltung  
G&V / Bilanz

## Controlling

Kostenrechnung  
Planung  
Steuerung  
Managementinfo

## Personal

Lohn/Gehalt  
Personalplanung  
Zeitmanagement

## Management

Planung  
Überwachung  
Steuerung

## Marketing

Kunden  
Produkte  
Aktionen  
Veranstaltungen

## Archivierung

Revisionssichere  
Langzeitspeicherung  
von Dokumenten

## Dokumenten- Management

Aufgabenbezogene  
Speicherung von  
Dokumenten

## Portale

Kunden  
Lieferanten  
Mitarbeiter

...

Viele weitere  
Bereiche

- **Automobilindustrie**
  - Serienfertigung mit Individualisierungen
  - Just-in-time Beschaffung
- **Banken**
  - Management von Finanzprodukten
  - Weltweite Transaktionen
- **Lebensmittelindustrie**
  - Chargenfertigung und Herkunftsnachweise
  - Logistik mit Kühlkette
- **Öffentliche Verwaltung**
  - Überwiegend völlig andere Aufgaben, die es sonst nirgends gibt, z.B. Melderegister, Handelsregister, Grundbuch, etc.



**Selbst Unternehmen der gleichen Branche haben im Detail immer unterschiedliche Anforderungen.**

# Organisatorische Rahmenbedingungen (1)

- Meistens **getrennte Verantwortlichkeiten** der einzelnen Abteilungen, z.B. Vertrieb, Produktion, Logistik, etc.
  - **Häufig unkoordinierte Beschaffung von IT-Systemen**
- Einzelne Funktionsbereiche können auch bei **externen Dienstleistern** liegen, z.B. Personal, Logistik, etc.
  - **Eingeschränkter Einfluss auf externe Rahmenbedingungen**
- Es existieren häufig **keine Gremien**, um übergreifende Anforderungen zu definieren und abzustimmen.
  - **Keine Vereinheitlichung möglich**
  - **Kein zentrales IT-Architekturmanagement möglich**

- Die einzelnen Fachbereiche haben in der Regel unterschiedliche **nicht-funktionale Anforderungen**, u.a. in folgenden Bereichen:
  - Sicherheit (Integrität, Echtheit und Vertraulichkeit von Informationen)
  - Datenschutz (Verarbeitung personenbezogener Daten)
  - Verfügbarkeit (z.B. 7x24-Betrieb oder normale Bürozeiten)
  - Performance (z.B. Realtime-Maschinensteuerung oder Office-Betrieb)
- **Einheitlicher Betrieb und Schnittstellen sind kompliziert**
  
- Aufgrund von einer Zusammenlegung von verschiedenen IT-Anwendungen können einzelne Funktionsbereiche auch mehrfach vorhanden sein.
- **Schwierige Konsolidierung von Anwendungen und Daten**

- **Organisationsformen einer IT-Abteilung**
  - Zentrale Abteilung für alle Organisationseinheiten
  - Komplette oder teilweise Auslagerung (Outsourcing)
  
- **Hauptaufgaben einer IT-Abteilung**
  - Planung und Betrieb der technischen Infrastruktur
  - Beschaffung von IT-Komponenten
  - Benutzer-Support
  - Bereitstellung von fachlichen und technischen Diensten (Services) für die Abteilungen
  - Entwickeln und Anpassen von Anwendungen



**Typischerweise gehört die Organisation von abteilungsübergreifenden Abläufen nicht zu diesen Aufgaben.**

- In „gewachsenen“ IT-Landschaften muss die IT-Abteilung einen ganzen „Zoo“ von unterschiedlichen IT-Systemen beherrschen.
- Eine umfassende **Betriebsdokumentation** über die Verflechtung aller Systeme fehlt in aller Regel.
- **Wartungsfenster** sind teilweise schwer zu finden, da hierfür auch Systeme abgeschaltet werden müssten, mit denen andere Systeme kommunizieren.

→ **Was passiert, wenn man ein System herunterfährt?**

- Fehlerhafte **Updates** an einem System können zu Fehlern im Zusammenspiel mit anderen Systemen führen.

→ **Welche Systeme werden von einem Update betroffen sein?**

- **Batch-Prozesse** vertragen sich oft mit dem Online-Betrieb nicht
- Eine sachgerechte **Datensicherung** aller Systeme ist oft schwierig

- Geschäftsprozess
- Standardsoftware vs. Individualsoftware
- Schnittstellen
- Management von Benutzeridentitäten und Berechtigungen (IAM)
- ERP
- EAI



## Was ist ein Geschäftsprozess?

„Ein **Geschäftsprozess** (business process) ist eine funktions- und stellenübergreifende Folge von Schritten zur Erreichung eines geplanten Arbeitsergebnisses in einem Unternehmen. Diese Schritte heißen **Geschäftsprozessaktivitäten** (business process activities), kurz **Aktivitäten**.

Der Geschäftsprozess dient direkt oder indirekt zur Erzeugung einer Leistung für einen Kunden oder den Markt.“

Quelle: „Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten“

„Ein Geschäftsprozess kann Teil eines anderen Geschäftsprozesses sein oder andere Geschäftsprozesse enthalten bzw. diese anstoßen.“

Quelle: Wikipedia

Beispiel: Verkauf von Individualreisen für Geschäftskunden

- Für fast jede Aktivität oder jeden Geschäftsprozess in einem Unternehmen gibt es eigenständige, optimierte Speziallösungen.

Typische Eigenschaften solcher „Speziallösungen“:

- In der Regel nur für einen **Teilausschnitt** aller Geschäftsprozesse eines Unternehmens entwickelt.
- Die Funktionen eines Aufgabenbereichs werden vollständig durch **eine oder mehrere eigenständige Anwendungen** abgebildet.
- Für die Durchführung der Aktivität oder den Geschäftsprozess ist kein **Datenaustausch** mit anderen Systemen erforderlich.
- Es werden **keine Funktionen anderer Systeme** genutzt.

- Standardsoftware
  - Einrichtung auf die speziellen Bedürfnisse durch Parametrisierung
  - Beispiele:
    - Finanzbuchhaltung
    - PPS-System

Anmerkung:

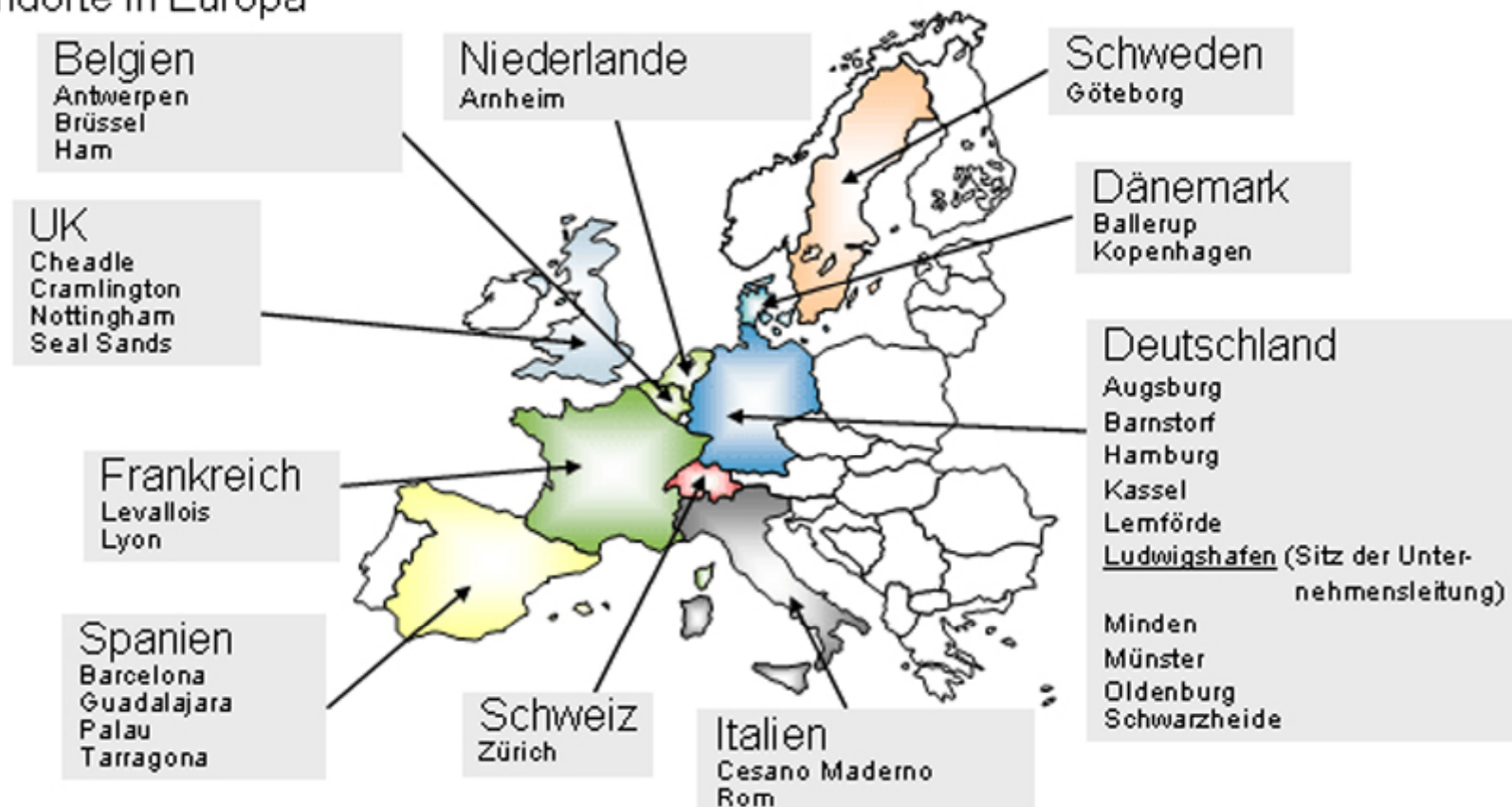
*Die beste Standardsoftware für einen Anwendungsbereich wird auch als „Best of Breed“ bezeichnet*

- Individualentwicklungen
  - Entwurf und Entwicklung einer Anwendung auf Basis der Anforderungen des Anwenders
- Mischformen
  - Standardsoftware mit individuell „hinzuprogrammierten“ Programmteilen

- Server
  - Großrechner, Mainframes
  - Midrange-Systeme, z.B. AS400
  - UNIX / LINUX
  - Windows
  
- Clients
  - Fat-Client
  - Terminal-Client
  - Web-Oberflächen
  
- Netzwerk
  - Ethernet, Token-Ring im LAN
  - Spezielle Maschinensteuerungen, z.B. CAN-Bus
  - Weitverkehrsnetze mit Standleitungen oder Wählverbindungen
  
- Softwaretechnik
  - Alte Sprachen: COBOL, BS2000-Assembler, BASIC, C, 4GL, ...
  - Moderne Sprachen: C++, Java, ...
  - Architekturen: Monolith, Client/Server- oder Multi-Tier-Architektur

## Beispiel: BASF IT-Services Europa

Standorte in Europa



- Daten(austausch)schnittstellen
- Einfache Funktionsaufrufe (API)
- Komplexe Sequenzen von Funktionsaufrufen (→ Protokolle)

- In komplexen Anwendungslandschaften ist der Datenaustausch ein zentrales Thema.

## Beispiele

### a) Einfacher Datentransfer

- Buchungssätze
- Verkaufszahlen
- Adressdaten

### b) Komplexer Datentransfer

- Behandlungsdaten (Gesundheitswesen)
- Strukturierte Objekte (Produktionsauftrag)

## ■ Übertragungsmethoden

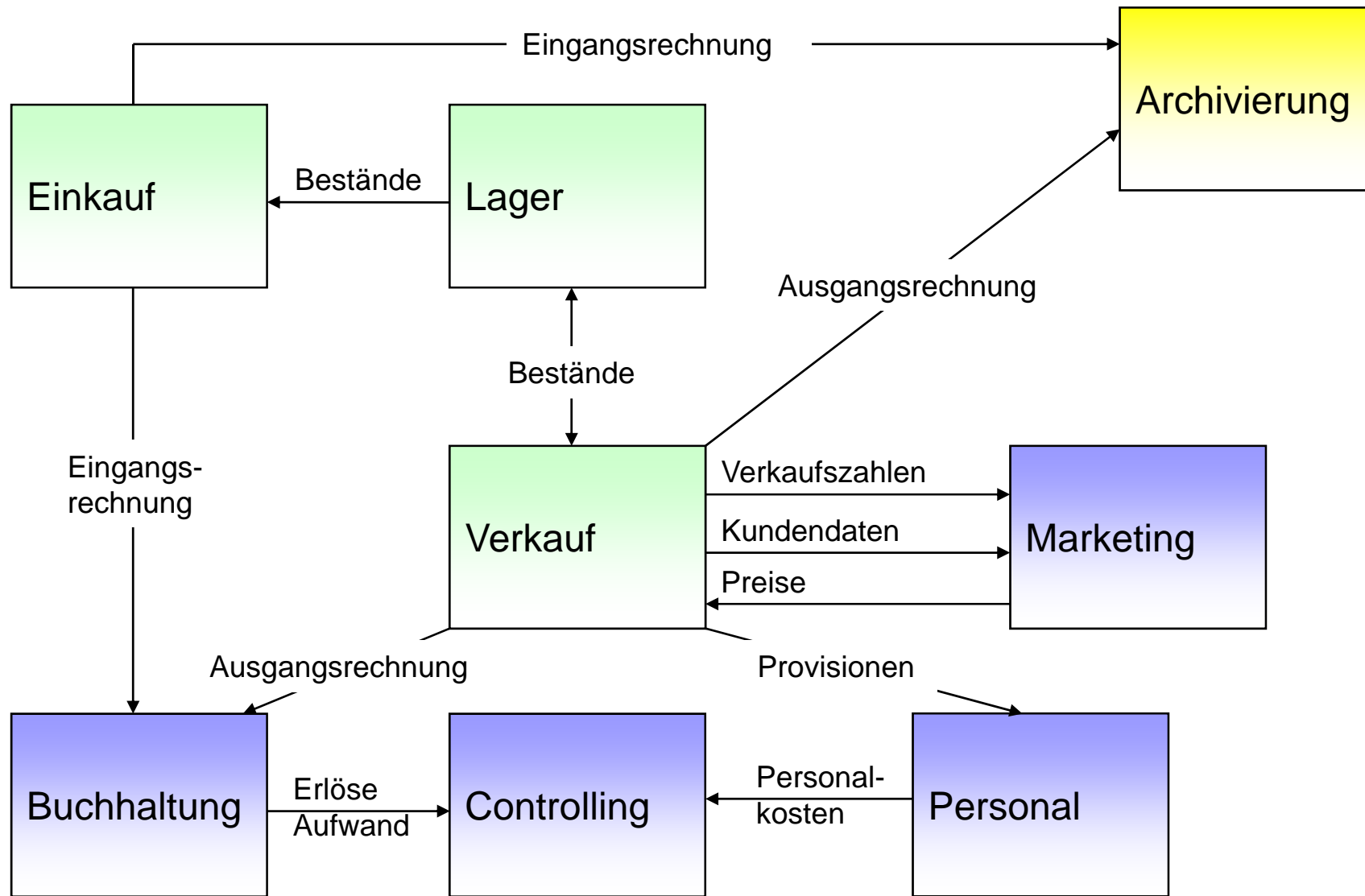
- Manuell (ausdrucken und wieder eingeben)
- File-Transfer (Daten im Quellsystem in eine Datei schreiben und diese im Zielsystem wieder einlesen, manuell oder automatisiert)
- Kommunikationsprotokolle (direkte Kommunikation zwischen Anwendungen oder über eine Zwischenschicht)

# Daten(austausch)schnittstellen (2)

- Typische Eigenschaften von Daten(austausch)schnittstellen
  - Definierter Informationsinhalt
  - Definierter Kommunikationsablauf
  - Transformation von Zeichensätzen
  - Mapping von Dateninhalten
  
- Physische Ausprägung
  - File
  - Kommunikationsprotokoll
  
- Zeitlicher Ablauf
  - Synchron
  - Asynchron

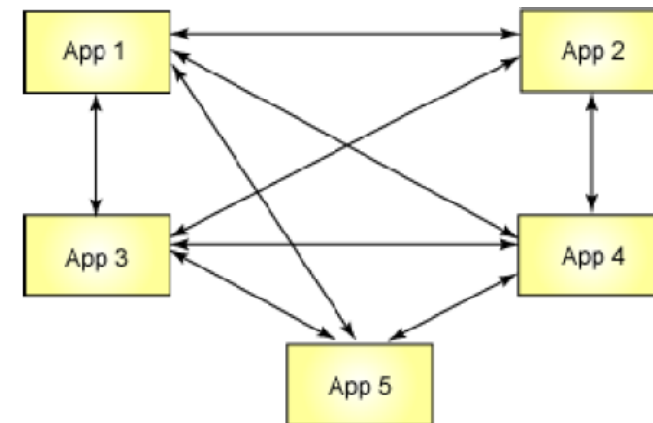


# Einfacher Datenaustausch



# Typische Schnittstellenprobleme

- Punkt-zu-Punkt-Kopplung
- Häufig individuelle, nicht standardisierte Schnittstellen
- Gelegentlich Ad-hoc-Schnittstelle ohne detaillierte Analyse
- Ungenügende Kapselung
- Bei zahlreichen Anwendungen treten viele Schnittstellen auf
- Mit jedem neuen System steigt die Komplexität
- Hohe Fehleranfälligkeit
- Schwer wartbar, da bei Änderungen an einem System evtl. mehrere Systeme adaptiert werden müssen (Release-Festigkeit!)
- Bei Änderung der „Lokation“ eines Systems müssen alle verbundenen Systeme entsprechend angepasst werden
- Pufferung von Daten bei Ausfall eines Systems?



## ▪ **Spezielle Kommunikationsserver**

Beispiele: eGate (JCAPS), Cloverleaf im Gesundheitswesen

- Datentransformationen
- Einfache Routing-Regeln
- Verbesserte Transparenz

## ▪ **Komplexe Middleware-Lösungen**

Beispiele: DCE, CORBA, Websphere MQ

- Datentransformationen
- Komplexe Routing-Regeln
- Hohe Transparenz durch Verzeichnisdienst, Pufferung und Adapter
- Verteilte Transaktionen



**Es handelt sich überwiegend um sehr komplexe Werkzeuge.**

**Die Verwendung proprietärer Standards legt den Anwender auf einen Hersteller / ein Produkt fest.**

# Beispiel: Cloverleaf HL7-Transformation

The screenshot shows a software interface for HL7 transformation. It is divided into three main vertical sections:

- Left Panel (Input Message Format):** Labeled "Eingabedatenstruktur" and "SAP-Nachricht". It shows a tree view of input fields: HEA: HEA.frl, PAT: PAT.frl, { ANG: ANG.frl }, { FAL: FAL.frl }, { FKL: FKL.frl }, { WF: WF.frl }, and END: END.frl.
- Middle Panel (Operation):** Labeled "Transformationsregeln". It contains a table of transformation rules with columns for "Operation", "Pre", "Post", and "Source/Destination".
 

Operation	Pre	Post	Source/Destination
COPY			FALATTPTNAM1 -> {0(0).PV1.0...
COPY			FALATTPTNAM2 -> {0(0).PV1.0...
COPY			FALATTPTHTITL -> {0(0).PV1.00...
COPY			FALATTPTHTADD -> {0(0).PV1.0...
COPY			FALATTPTHTYPREF -> {0(0).PV1.0...
COPY			FALATTPTHTYPID -> {0(0).PV1.001...
COPY			FALATTPTHTYPNAM1 -> {0(0).PV1.0...
COPY			FALATTPTHTYPNAM2 -> {0(0).PV1.0...
COPY			FALATTPTHTYPHTITL -> {0(0).PV1.00...
COPY			FALATTPTHTHTMPPHYNADD -> {0(0).PV1...
COPY			FALATTPTHTHTMPPHYHPREF -> {0(0).PV1.0...
CONCAT			FALADMHOQSNAM1 FALADMHO...
COPY			FALCASENO -> 0(0).PV1.00149
COPY			FALTREATMCAT -> 0(0).PV1.001...
CONCAT			FALADMISSDATE FALADMISSTI...
CONCAT			FALDISMISDATE FALDISMISTI...
COPY			FALDISHOSPID -> 0(0).PV1.00167
COPY			FALADMDIACA1 -> {0(0).DG1(0)...
COPY			FALADMDIAKEY1 -> {0(0).DG1(0)...
COPY			FALADMDIAKIT1 -> {0(0).DG1(0)...
CONCAT			FALADMISSDATE FALADMISSTI...
COPY			=A -> 0(0).DG1(0).00380
COMMENT			VVF - Versicherungsverhältnis...
ITERATE			
IF			WF(%s1).SELPAYFLG ne =X
ELSE			
COPY			=S -> 0(0).PV1.00151
- Right Panel (Output Message Format):** Labeled "Ausgabedatenstruktur" and "HL7-Nachricht". It shows a tree view of output fields: MSH, EVN, PID, [ PD1 ], [[ NK1 ]], PV1, [ PV2 ], [[ DB1 ]], [[ OBX ]], [[ AL1 ]], [[ DG1 ]], [ DRG ], 1 [...], 1.0 [...], { PR1, [[ ROL ]}.

At the bottom, the "Action" dropdown is set to "COMMENT" and the "Error" dropdown is set to "Skip". A text box at the bottom contains the comment: "VVF - Versicherungsverhältnisse Kassenpatient wenn SELFPAYFLAG nicht gesetzt ist, sonst Selbstzahler."

- **Aufruf von Funktionen anderer Systeme**
  - Abfrage von Daten
  - Verifikation / Prüfung von Daten
  - Trigger für das Starten oder Stoppen von Anwendungen
  
- **Typische Umsetzungen**
  - Remote Procedure Calls (RPC)
  - Remote Method Invocation (RMI)

Anmerkung:

*Auch hier ist die Nutzung von Middleware-Schichten möglich, um die (Software-) Verteilung und die technische Basis der aufrufenden und aufgerufenen Systeme zu verbergen.*

Beispiel: Flugbuchung in einer Reisebüroanwendung

- Jeder Systemnutzer muss am Computer, im Netzwerk und in den Anwendungen, die er benutzt, **identifiziert** werden können.
- Betriebssysteme stellen Standardmethoden für die **Authentifizierung** von Benutzern zur Verfügung, z.B. UNIX-Berechtigungssystem, Windows Active Directory.
- Anwendungen haben meistens ihre eigene **Benutzerdatenbank**, um Benutzer zu authentisieren.
- Aufgrund der Heterogenität der Systeme ist eine **einheitliche zentrale Identität des Benutzers** nicht gegeben. Ein und derselbe Benutzer muss auf mehreren Systemen eingerichtet und administriert werden.  
**→ Mehrere Benutzernamen und Passwörter für jeden Benutzer**
- Idealvorstellung wäre ein **Single Sign On (SSO)**, bei dem sich ein Benutzer nur einmal am System anmelden muss und alle Anwendungen entsprechend seiner jeweiligen Berechtigung nutzen kann.

## ▪ Access Management

- Die **Verwaltung der Zugriffsrechte** auf Anwendungen oder bestimmte Daten wird als „Access Management“ bezeichnet.
- Jeder authentifizierte Benutzer hat üblicherweise **definierte Rechte**.
- **Berechtigungen** müssen wie die Benutzer selbst in jeder einzelnen Anwendung administriert werden.

→ **Keine zentrale Übersicht über Berechtigungen eines Benutzers**

## ▪ Tools

- **Verzeichnisdienste** für zentrale Benutzerverwaltung: LDAP & Co.
- Komplette „**Identity Management Suites**“ sollen die zentrale Administration von Identitäten und Zugriffsrechten ermöglichen.



Voraussetzung ist die **Unterstützung durch die Anwendungen** bzw. die Verfügbarkeit von **Adaptern!**

→ Keine vollständig durchgängige Lösung

- Für jede Aufgabe gibt es **eigenständige Lösungen**, die entweder aus historischen Gründen noch vorhanden sind oder aufgrund herausragender Eigenschaften ausgewählt wurden („Best of Breed“).
- Manche **Funktionen und Daten** sind in einzelnen Anwendungen mehrfach und uneinheitlich vorhanden.
- Der Einsatz solcher Lösungen erfordert in der Regel einen **komplexen Datenaustausch**.
- Es gibt zahlreiche **Tools**, die den Datenaustausch unterstützen.
- Es existiert häufig **keine zentrale Steuerung** der Abläufe für einen Geschäftsprozess oder zwischen mehreren Geschäftsprozessen.

## Zwischenfrage:

Wer soll für die Koordination der IT-Lösung hinsichtlich der Geschäftsprozesse und Aktivitäten verantwortlich sein?



- Klassisches, „integriertes“ Softwarepaket
- Versuch, zentrale Geschäftsprozesse mit einem einzigen Anwendungssystem abzudecken
- Anpassung auf die speziellen Bedürfnisse des Anwenders
  - a) Änderungen im Programmcode der Software
  - b) Erweiterungen durch zusätzliche Programmierungen
  - c) Einstellen von (Konfigurations-) Parametern
  - d) Eingabe / Pflege von Stammdaten
- Zentralisiertes IAM innerhalb des ERP-Systems

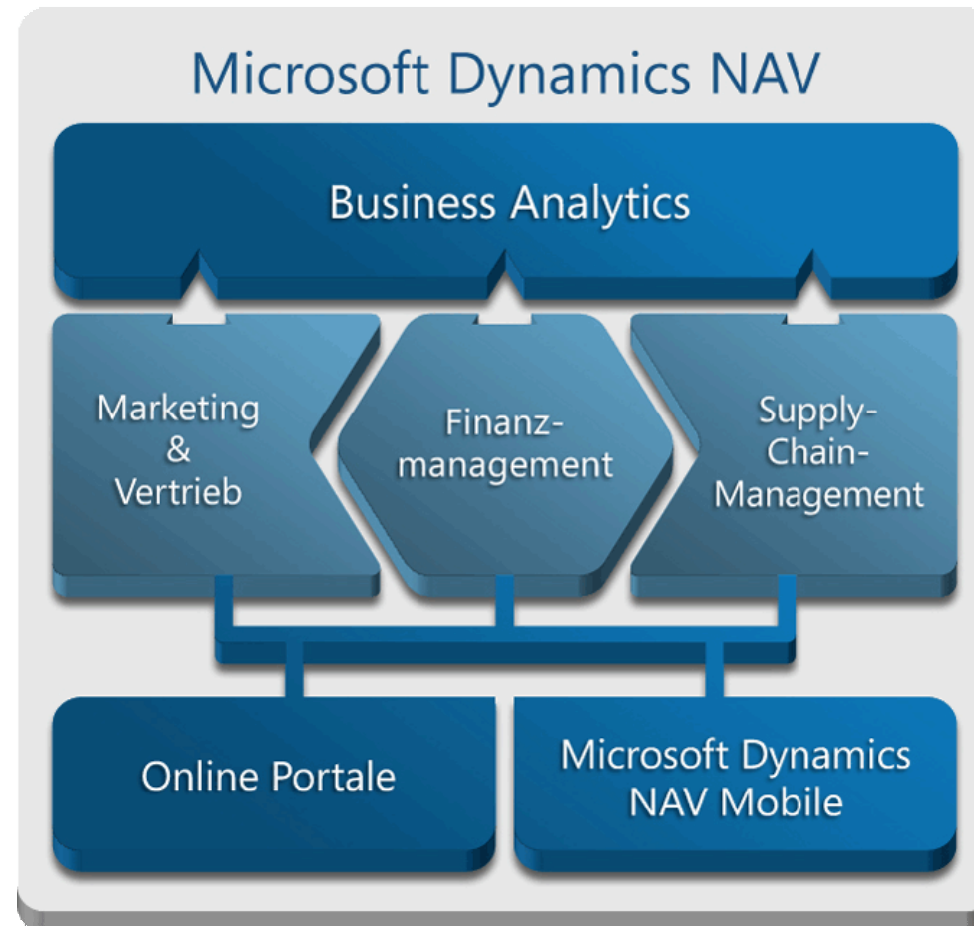
## Beispiele:

SAP R/3, Microsoft Dynamics NAV (Navision), Sage, zahlreiche weitere Systeme mit teilweise spezieller Branchenausrichtung (sog. Branchensoftware oder Branchenpakete)

# Beispiel: SAP-Anwendungsportfolio

Analytics	Financial Analytics		Operations Analytics		Workforce Analytics		
Financials	Financial SCM	Treasury	Financial Accounting	Management Accounting	Corporate Governance		
Human Capital	Talent Management		Workforce Process Management		Workforce Deployment		
Procurement	Procurement	Inventory Warehousemgmt.	In-/Outbound Logistics		Transportation Management		
Manufacturing	Production Planning	Manufacturing Execution	Product Development		Lifecycle Data-management		
Sales / Service	Sales Order Management		Aftermarket Sales and Service		Professional Service Delivery		
Corporate Serv.	Real Estates	Enterprise Assets	Project Portfolio	Travel Mgmt.	Health Safety	Quality Mgmt.	Global Trade

# Beispiel: Microsoft Dynamics NAV



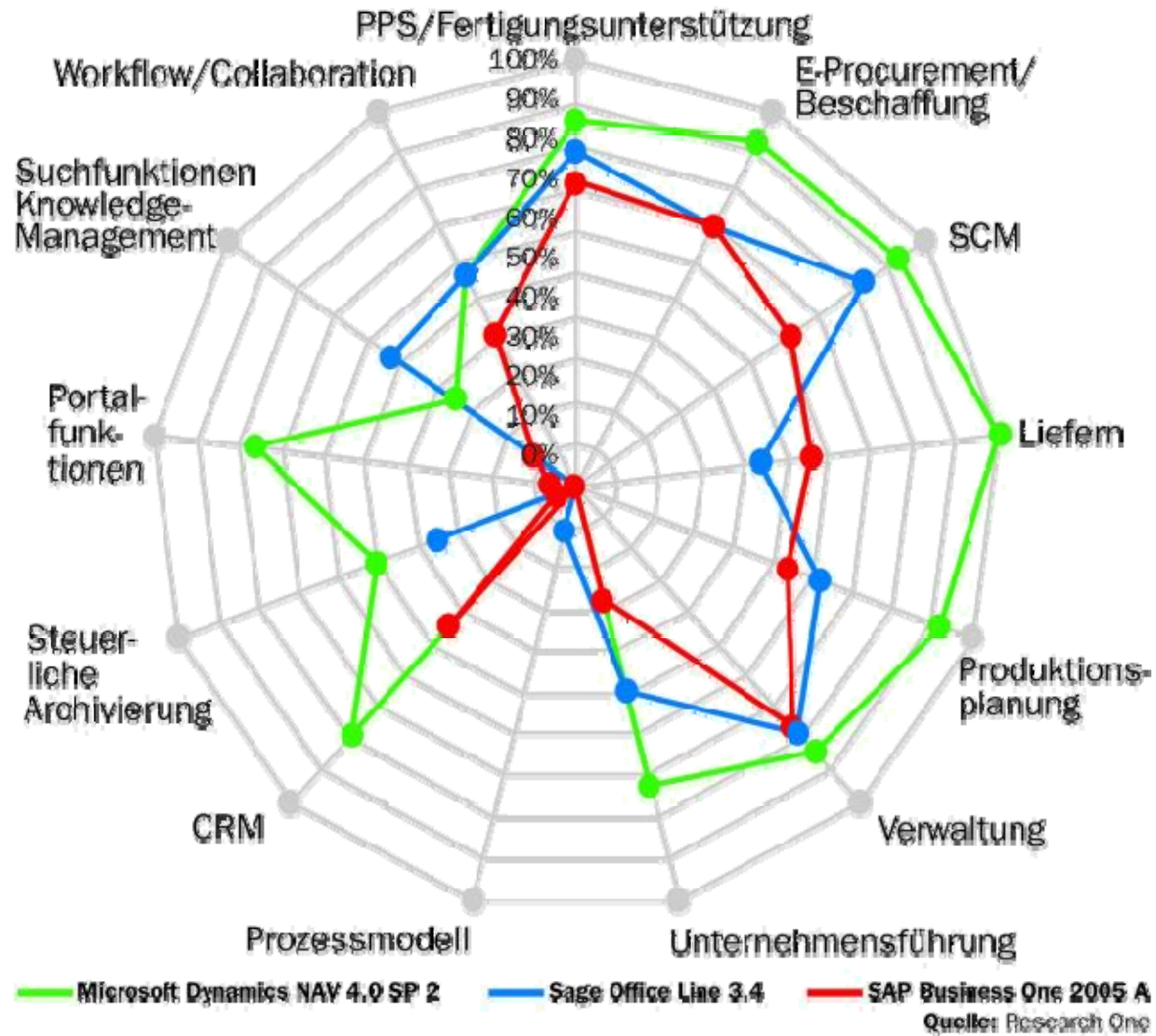
## ■ Vorteile

- Die Datenintegration vermindert die **Anzahl der Schnittstellen**.
- Die **Zusammenfassung von Funktionen** vereinfacht das System.
- Die Vereinheitlichung der **Benutzeroberfläche** erleichtert die Bedienung.
- Es kann eine **Kostenreduzierung** erzielt werden, wenn das System ausreichende Funktionen bietet.

## ■ Nachteile

- Viele Funktionen eines Standard ERP-Systems sind für **spezielle Verwendungszwecke** nicht ausreichend.
- Die Möglichkeiten für **individuelle Anpassungen** sind teilweise begrenzt oder sehr teuer.
- Es sind weiterhin Schnittstellen für die **Anbindung von Spezialem Systemen** erforderlich.

# Beispiel: ERP-Funktionsabdeckung



- Prozessorientierte Integration einzelner Anwendungssysteme auf Basis standardisierter Schnittstellen (meist proprietär)

Beispiele:

- SAP Netweaver
  - Microsoft BizTalk
  - IBM Websphere Business Integration
- Anwendungsübergreifende Prozesssteuerung häufig gegeben
  - EAI ermöglicht die Integration von Einzelanwendungen und unterstützt so den „Best-of-Breed“-Ansatz sehr gut



Adapter, die für die Kopplung von Anwendungen benötigt werden, sind oft non-standard und auf Dauer schwer wartbar

- In großen IT-Systemen sind zahlreiche Aufgaben zu erledigen, die mit bisherigen Mitteln **beträchtliche Aufwände** auslösen.
- Der Überblick über das gesamte System ist unvollständig; die **Dokumentation** ist lückenhaft und schlecht gepflegt.
- Der **Betrieb** ist häufig sehr aufwendig und kostenintensiv.
- Die **Flexibilität** ist aufgrund zahlreicher Abhängigkeiten eingeschränkt.
- Eine **Wiederverwendung** von Funktionen ist aufgrund der Struktur der IT-Anwendungen praktisch nicht möglich.
- **Änderungen** an Geschäftsprozessen können zu aufwendigen Änderungen in der IT-Landschaft führen.
- Es fehlt an einer globalen **Methodik** und an den passenden **Werkzeugen** zur durchgängigen Abbildung von Geschäftsprozessen.