

# Wirtschaftsinformatik Zusammenfassung

Informationssysteme

Electronic Business

Entscheidungsunterstützungssysteme

Systementwicklung

Datenmodellierung

Business Intelligence

# Informationssysteme

## Wirtschaftsinformatik

Wissenschaft, die sich mit der **Beschreibung, Erklärung** und **Gestaltung rechnergestützter Informationssysteme** und deren Einsatz in der Wirtschaft befasst.

Informationssysteme als soziotechnische Systeme (menschliche und technische Komponente)

Einsatz hat technische, organisatorische, personelle und soziale Aspekte

Wirtschaftsinformatik als interdisziplinäres Fach zwischen BWL und Informatik

## Bereiche der Wirtschaftsinformatik

### Anwendungen

Branchen: Industrie, Handel, Dienstleistung

Anwendungen: Supply Chain, Enterprise Resource Planning, Customer Relationship Management, ...

### Daten

Modellierung von: Geschäftsprozessen, Daten, Wissen, Schnittstellen, Architekturen, Programmen

### Informationstechnik

Automation (z.B. Computer Integrated Manufacturing), Speicherung (Datenmanagement),

Kommunikation (Netzwerke), Transformation (z.B. neuronale Netze)

### Speicherung

Prozessunterstützung: Projektmanagement, Anforderungsanalyse, Systementwurf, Implementierung, Testen, Integration, Informationsmanagement

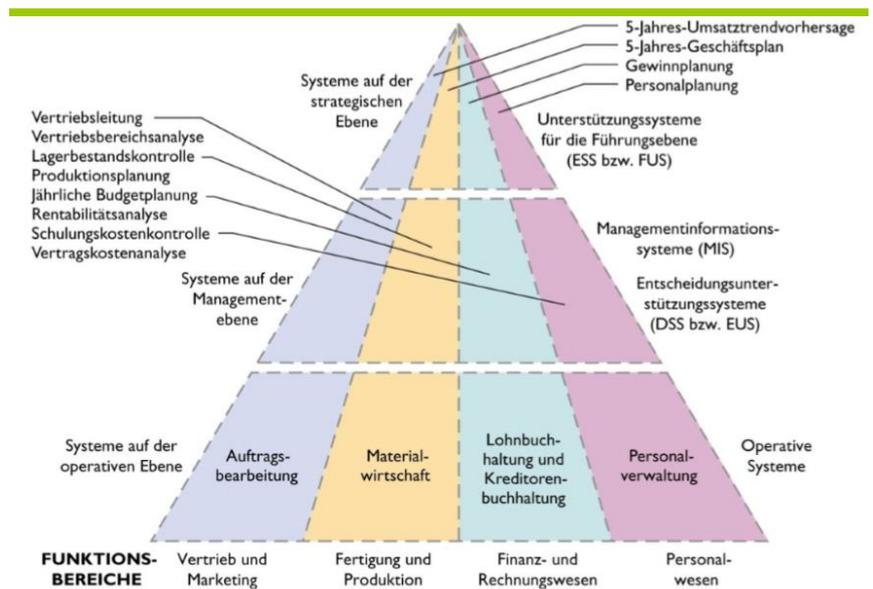
### Berufsfelder

IT-Manager, IT-Berater,

Projektmanager, Mitarbeiter im

Vertrieb, Systemanalytiker

## Informationssysteme



*Spektrum betrieblicher Informationssysteme*

## Rechner/Computer

Funktionseinheit zur Verarbeitung von Daten, wobei als Verarbeitung die Durchführung mathematischer, umformender, übertragender oder speichernder Operationen definiert ist



Mainframe: Großrechner, auf denen mehrere (tausend) Benutzer arbeiten; Verarbeitung großer Datenmengen (Zentralrechner in Banken)

Supercomputer: meist parallele Datenverarbeitung; wissenschaftliche oder militärische Anwendung

### Architektur eines Rechners

Ausgabe: u.a. Monitor, Drucker, Lautsprecher

Eingabe: u.a. Tastatur, Maus, Scanner

Rechnergrundgerät: Hauptplatine, Prozessor, Speicher (intern, extern), Verbindungssystem

Rechnersystem: Hardware (physische Geräte, Mikro-Architektur, Maschinensprache); Systemprogramme (Betriebssystem, Compiler, Editoren, Kommando-Interpreter); Anwendungsprogramme (Bankensysteme, Flugbuchungssysteme, Webbrowser)

### Vier Schichten der Datenübertragung

Anwendungsschicht: regelt Kommunikation zwischen Anwendungsbereichen wie z.B. Webbrowser und Webserver (HTTP Protokoll), Mailclient und Mailserver (POP bzw. SMTP Protokoll) durch entsprechende Protokolle

Transportschicht: regelt Aufteilung in Datenpakete und Transport der Pakete zwischen sendendem und empfangendem System (TCP Protokoll)

Internetschicht: regelt Weiterleitung der Daten von einer Netzwerkkomponente zur nächsten bis hin zum Zielsystem (IP Protokoll)

Netzwerkschicht: ist für physikalische Datenübertragung zuständig (elektrische Impulse, Funk- oder Listwellen)

Internet Protocol (IP): ist eindeutig; besteht aus vier natürlichen Zahlen, zur Lesbarkeit durch Punkt getrennt

Transmission Control Protocol (TCP): zerlegt zu übertragende Daten in Pakete einheitlicher Größe

HTTP Protokoll: Request (Dokumenten-anforderung, Informationen über den Klient), Reponse (angefordertes Dokument, Informationen über Server)

## Informationssystem

**System**, das für die Zwecke eines bestimmten Unternehmens geschaffen bzw. in diesem Betrieb eingesetzt wird; Informationssystem enthält die dafür notwendige Anwendungssoftware und Daten und ist in die Organisations-, Personal- und Technikstrukturen des Unternehmens **eingebettet**

Sozio-technisches System

## Anwendungssystem

**System**, das alle Software beinhaltet, die für ein bestimmtes betriebliches **Aufgabengebiet** entwickelt wurde, inklusive **Technik** (IT-Infrastruktur), auf der das Anwendungssystem läuft und der **Daten**, die vom Anwendungssystem genutzt werden

### Ziele des Einsatzes von Informationssystemen

Beschleunigung Geschäftsprozess  
Verbesserung Qualität und Service  
Kosteneinsparungen

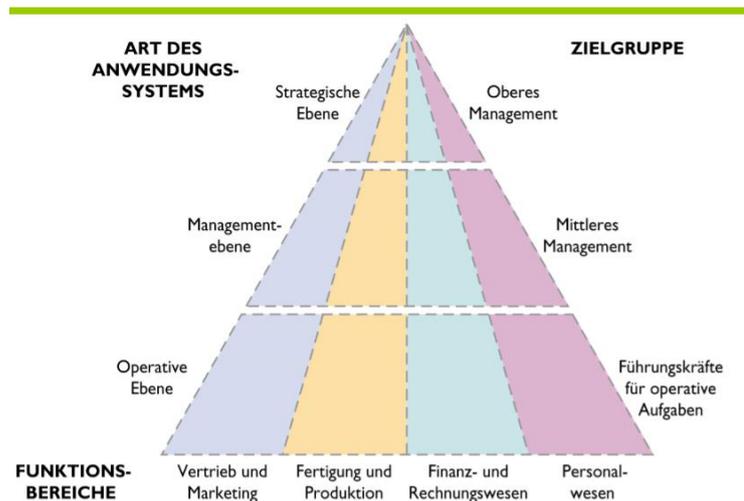
### Wachsender Einfluss

Von einfachen, isolierten Informationssystemen zu Supply Chain Management, Customer Relationship Management und Wissensmanagement

### Gliederungsarten

Nach Unterstützungsebene, Funktionen, Erstellungsart, Integrationsart, Branchen

### Unterstützungsebene vs. Funktionsbereich



## Gliederung nach Unterstützungsebene

### Operative Systeme

Anwendungssysteme, die die **täglichen**, für den Geschäftsbetrieb notwendigen **Routinetransaktionen** ausführen oder aufzeichnen

Unterstützungsebene

**Ausfall** führt zum Erliegen zentraler Geschäftsaktivitäten

### Managementinformationssysteme (MIS)

Anwendungssysteme, die durch Bereitstellung von **Standardübersichtsberichten** sowie **Berichten über Abweichungen** der Planung, Kontrolle und Entscheidungsfindung dienen

Managementebene

### Entscheidungsunterstützungssysteme (EUS)

Anwendungssysteme, die **analytische Modelle** sowie **Datenanalysewerkzeuge bereitstellen**, um schwach strukturierte Entscheidungsfindungsprozesse zu unterstützen

Englisch: Decision Support Systems DSS

Managementebene

### Führungsunterstützungssysteme (FUS)

Anwendungssysteme auf der **strategischen Ebene** des Unternehmens, die **Daten** aus diversen internen und externen Quellen **integrieren** und in einfach verwendbarer, oft graphischer Form **Führungskräften zur Verfügung** stellen

Englisch: Executive Support Systems ESS

strategische Ebene

## Gliederung aus funktionaler Sicht

### Vertriebsunterstützungssysteme

Anwendungssysteme, die dem Unternehmen dabei helfen, Produkte und Dienstleistungen zu **bewerben** und zu **verkaufen** und kontinuierlichen **Service für den Kunden** bereitzustellen

### Fertigungs- und Produktionssysteme

Anwendungssysteme, die zur **Planung, Entwicklung** und **Produktion** von Produkten und Dienstleistungen sowie zur **Überwachung** des Produktionsablaufs dienen

### Finanz- und Buchhaltungssysteme

Anwendungssysteme, die zur **Verwaltung der finanziellen Vermögenswerte** und der **Ein- und Ausgaben** einer Firma dienen

### Anwendungssysteme für das Personalwesen

Anwendungssysteme, mit denen **Personaldaten** geführt und Informationen über Fähigkeiten, Arbeitsleistung und Weiterbildung der Mitarbeiter verzeichnet werden, sowie die Planung des Arbeitsentgelts und der beruflichen Laufbahn der Mitarbeiter unterstützt wird

### Betriebliche Standardsoftware

Software für **allgemeine, unternehmens-neutrale Funktionen** und Problemstellungen

### Individualsoftware

Software, die **für ein einzelnes Unternehmen entwickelt** wurde, um spezielle Anforderungen zu realisieren

Verstärkter Einsatz von Standardsoftware in Unternehmen durch oft höhere Fehlerfreiheit, geringeres Entwicklungsrisiko, günstigere Kosten

## Microsoft Office als Beispiel für Standardsoftware

### Power Point

Programm zum Erstellen von **Präsentationen**

Folienmaster: Schablone, in der festgelegt wird, wie alle Folien der Präsentation aussehen sollen

Einfache Bildbearbeitung in Power Point möglich

Leitlinien für eine gute Präsentation: **KISS** (Keep It Short and Simple)

### Word

Programm zur **Textverarbeitung**

Inhaltsverzeichnis muss manuell aktualisiert werden

Mit **Formatvorlagen** arbeiten, statt manuell zu formatieren

Zitate, Fußnoten

### LaTeX

Für wissenschaftliche Publikationen oft genutzt

Sehr gute Satzqualität = professionelles Erscheinungsbild

Referenzen ohne Pflege der Worddatenbank → Inhaltsverzeichnis und Referenzen immer aktuell

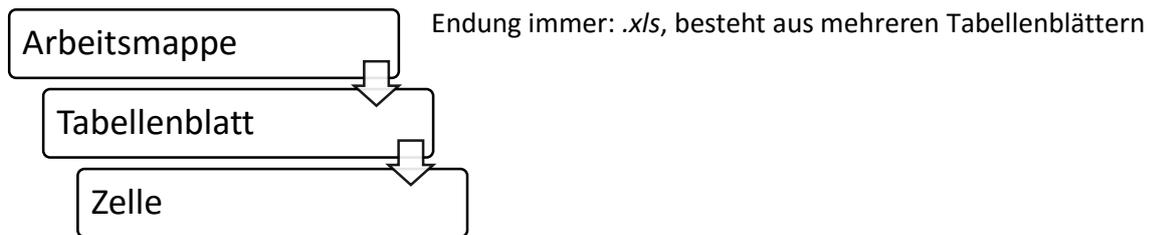
Endung: .tex

LaTeX Datei besteht aus Präambel (welche Art Dokument liegt vor?, welche Ansprüche an Zitation, Grafiken, etc.?) und Textkörper

Referenzen mit Bibtex

## Tabellenkalkulationen (Excel)

Dienen zur **Berechnung und Analyse** von Daten



Zellen können formatiert werden

Formel beginnt immer mit **Gleichheitszeichen =**

Einfache Funktionen: MAX(); MIN(); SUMME(); ANZAHL(); MITTELWERT(); MEDIAN(); WURZEL()

: Bezug auf Zellbereich

; ermöglicht Verbindung mehrerer Bezüge

Relative Adressierung: beim Kopieren ändern sich Zeilen- bzw. Spaltenreferenzen automatisch

Absolute Adressierung: beim Kopieren verändern sich die Zellreferenzen nicht → ‚\$‘-Zeichen

E6 → Verschiebung entlang Spalten und Zeilen

\$E\$6 → Keine Verschiebung entlang Spalten und Zeilen

\$E6 → Keine Verschiebung entlang der Spalten

E\$6 → Keine Verschiebung entlang der Zeilen

WENN-Funktion: =WENN(Bedingung; Wahr-Zweig; Falsch-Zweig)

Auch geschachtelte WENN-Funktionen möglich

ZÄHLENWENN-Funktion: =ZÄHLENWENN(Bereich; „Suchkriterium“)

z.B. =ZÄHLENWENN(B5:B16; „>20000“)

SUMMEWENN-Funktion: =SUMMEWENN(BereichSuche; „Suchkriterium; BereichSumme; WAHR/FALSCH)

SVERWEIS-Funktion: =SVERWEIS(Suchkriterium; Matrix; Spaltenindex)

Findet in Bereich aufgrund eines Suchwertes eine Zeile und gibt anschließend Wert einer zugehörigen Spalte zurück

WAHR: ungefähre Entsprechung

FALSCH: genaue Entsprechung

WVERWEIS analog

**S**VERWEIS: Suche **s**enkrecht (in Spalten)

**W**VERWEIS: Suche **w**aagrecht (in Zeilen)

## Informationssysteme im Wandel

Durch globale Trends: Fortschritt der Informations- und Kommunikationstechnik, Globalisierung der Wirtschaft, Zunahme Bedeutung Dienstleistungen, Wandel Unternehmensstruktur- und –kultur

### Triebkräfte der Globalisierung

Liberalisierung des Welthandels, sinkende Transportkosten, verbesserte Kommunikation

### Unternehmensweite Anwendungssysteme

Wettbewerb zwingt Unternehmen zur Optimierung aller Geschäftsprozesse

Erfordert Denken in Geschäftsprozessen über Funktionsgrenzen hinweg

Geschäftsprozess: Folge logisch zusammenhängender Aktivitäten, die für das Unternehmen einen Beitrag zur Wertschöpfung leisten, einen definierten Anfang und ein definiertes Ende haben, wiederholt durchgeführt werden und sich in der Regel am Kunden orientieren

Enterprise-Resource-Planning-Systeme: Unternehmensweites Anwendungssystem, das zentrale Geschäftsprozesse eines Unternehmen in einem einzigen Softwaresystem integriert und so den reibungslosen und unternehmensweiten Informationsaustausch ermöglicht

Bestehen aus einer zentralen Datenbank und mehreren Softwaremodulen

Beispiel: SAP

Zunehmende Verbreitung von ERP-Systemen

Supply-Chain-Management-Systeme: Anwendungssysteme, die Informationsaustausch zwischen Unternehmen und seinen Lieferanten und Kunden automatisieren, um Planung, Beschaffung, Fertigung und Vertrieb von Produkten und Dienstleistungen zu optimieren

*Supply Chain*: Netzwerk von Unternehmen und Geschäftsprozessen, das zur Materialbeschaffung, zur Umformung der Rohmaterialien in Zwischenprodukte oder fertige Produkte und zur Verteilung fertiger Produkte an Kunden dient

Herausforderungen: Globalisierung, Dynamisierung der Märkte, gestiegene Kundenanforderungen, Konkurrenzdruck

Push- und Pull-Lieferkettenmodell

*Supply-Chain-Management*: Ansätze zur unternehmensübergreifenden Koordination der Material-, Informations- und Zahlungsflüsse über gesamte Lieferkette

*Lieferkettenplanungssysteme:* Anwendungssysteme, die Erstellung von Bedarfsvorhersagen für Produkt ermöglichen und Einkaufs- und Produktionspläne für die gesamte Lieferkette des Produkts zu entwickeln

*Lieferkettensteuerungssysteme:* Systeme zur Verwaltung des Produktlaufs durch Verteilzentren und Lager, um sicherzustellen, dass Produkte auf effiziente Weise an richtige Standorte ausgeliefert werden

*Electronic Data Interchange (EDI):* Austausch standardisierter Geschäftsdokumente zwischen Informationssystem über elektronische Netzwerke

Customer-Relationship-Management-Systeme: Anwendungssysteme, die sämtliche Interaktionen der Firma mit Kunden aufzeichnen und analysieren, um Umsatz, Rentabilität, Kundenzufriedenheit und Kundenbindung zu verbessern

CRM ist kundenorientierte Unternehmensphilosophie, die mithilfe moderner Informations- und Kommunikationstechnologien versucht, auf lange Sicht profitable Kundenbeziehungen durch ganzheitliche, individuelle Marketing-, Vertriebs- und Servicekonzepte aufzubauen und zu festigen

*Analytisches CRM:* analysieren Kundendaten, um Informationen für die Verbesserung von Geschäftsleistung bereitzustellen; Kennzahlen: z.B. Anteil des Budgets pro Kunde, Kosten pro Interessent, Kundenzufriedenheit, CLV

*Operatives CRM:* CRM-Anwendungen, die direkt mit Kunden zu tun haben, z.B. Werkzeuge für Automatisierung von Aufgaben im Verkauf

Wissensmanagement-Systeme: Anwendungssysteme, die Erwerb, Erfassung, Speicherung, Weitergabe und Nutzung firmenbezogenen Wissens und Fachkenntnissen unterstützen

*Wissensmanagement:* Gesamtheit aller Aktivitäten, die darauf abzielen, Fähigkeiten eines Unternehmens zum Erwerb, der Erfassung, Speicherung, Weitergabe und Nutzung von firmenbezogenen Daten zu verbessern; effektives Wissensmanagement beinhaltet 80% Management, Organisation und Unternehmenskultur und 20% Technik

*Wiki-Systeme:* Sammlung von Intranet- oder Webseiten, die von Benutzern nicht nur gelesen, sondern auch direkt online geändert werden kann

*Blogs, Mailinglisten, Diskussionsforen*

*Portale:* Enterprise Information Portals (EIP) erlauben einfachen, personalisierten Zugriff auf alle arbeitsrelevanten Inhalte und Anwendungen über einen Webbrowser

# Electronic Business

Internet als Medium der schnellsten Diffusion

¾ der Deutschen online

Weltweit mehr als 2 Mrd. Surfer

## New Economy

Ökonomisches Handeln in einer Volkswirtschaft mit veränderten Rahmenbedingungen und „neuen“ ökonomischen Regeln

## E-Business

Unterstützung der Leistungserstellung und der horizontalen und vertikalen Koordination auf Märkten durch Informationstechnik

## E-Commerce

1. Nutzung des Internets, der digitalen Kommunikation und der IT-Anwendungen, um den Prozess der **Kaufens** und **Verkaufens** zu ermöglichen
2. Alle oder zumindest wichtige Phasen einer Transaktion zwischen Anbieter(n) und Nachfrager(n) erfolgen via elektronischer Medien

Bezieht sich allein auf Realisierung der Koordination zwischen Käufer und Verkäufer  
Unterstützung und Realisierung wichtiger Phasen des Kaufens und Verkaufens durch die IT  
Vorbereitung, Anbahnung, Realisierung echter Kaufakte

## Einsparungspotentiale und Erlösquellen durch E-Business

Erlösquellen: Transaktionen (online/offline), Werbung, Provision, Datenhandel

Einsparungspotentiale: Senkung von Transaktionskosten, Senkung von Suchkosten, Senkung der Kommunikationskosten, Zeitersparnis

## Beispiele „online-basierter“ Geschäftssysteme

Amazon, Google, Otto, YouTube, Xing, Dell, Spiegel Online, Ebay, T Home, ...

## Geschäftsmodell und Geschäftssystem

Geschäftsmodell: bezeichnet allgemein-typisierend Abbildung des betrieblichen Produktions- und Leistungssystems einer Unternehmung oder Kooperations-/Anbietergemeinschaft

Geschäftssystem: konkrete Ausgestaltung eines allgemeinen Geschäftssystems, z.B. Ebay, Google

## Geschäftsmodelltypen

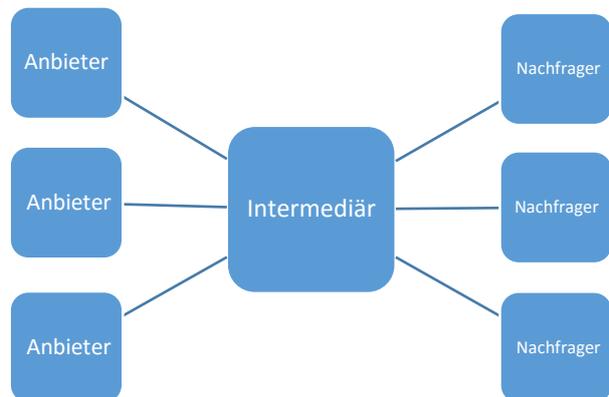
### E-Shop

#### 3<sup>rd</sup> party marketplace

*Zusammenführung von Käufern und Verkäufern:* Bestimmung Leistungsangebote, Suche (von Käufern für Verkäufer, von Verkäufern für Käufer), Unterstützung zentraler Transaktionsphasen, insbesondere Preisbestimmung

*Durchführung der Transaktion:* Vereinbarung Leistung – Gegenleistung, Logistik, Zahlung und Versicherung, Sicherung

*Institutionalisierte Infrastruktur*



### *Systematisierungsansätze von Elektronischen Marktplätzen*

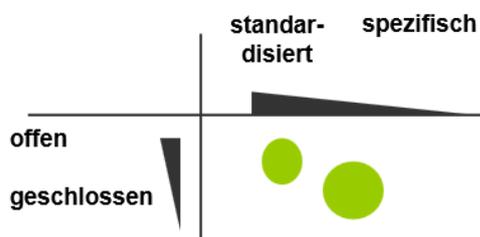
#### Nach dem Leistungsangebot

- **Horizontal** (für eine bestimmte Leistung / Funktion: MRO, Logistik, Finanzen...)
- **Vertikal** (für eine bestimmte Branche: Chemie, Pharma ...)
- **„Meta-Markets“** (vertikal + horizontal)

#### Nach den Betreibern des Marktes

- **„buy-side“** (Nachfrager als Betreiber)
- **„sell-side“** (Anbieter als Betreiber)
- **„3. Party“** (durch unabhängigen Dritten)

#### Nach der Offenheit/Standardisierung



#### Nach den Transaktionsmechanismen

- **Katalog-Modell** (z.B.: Aggregation von Angebotskatalogen)
- **Börsen** (temporärer Ausgleich zwischen Angebot und Nachfrage, z.B. Reste)
- **Auktionen** („vorwärts“ / „rückwärts“)
- **„Group Buying“** (Nachfragebündelung)

collaboration platforms

E-Procurement

Virtual Communities

Information Broker, Trust Provider, Consultancy

E-Auction

Value chain service provider

## E-Mail

### Value chain integrator

## Web 2.0

„The Web as a Platform“

„Harnessing Collective Intelligence“

„Data is the next Intel Inside“

„End of the Software Release Cycle“

„Lightweight Programming Models“

„Software above the Level of Single Device“

„Rich User Experience“

## Facebook-Fallstudie

Schwachstellen der Datenschutzpolitik und –funktionen: Opt-out Probleme, Account Löschung, Fremd-Posting, Digitaler Nachlass, Gesichtserkennung, Account Verlinkung, Klarnamenpolitik

# Entscheidungsunterstützungssysteme

Einordnung: Systeme auf der Managementebene (Mittelteil der Pyramide)

## 1. Mathematische Optimierung

Eine der wichtigsten Methoden zur quantitativen Entscheidungsunterstützung

Abbildung eines Teils der Realität als mathematisches Modell

Anwendungen: Produktionsplanung, Produktmixüberlegungen, Verschnittminimierung, Transportkostenminimierung

Methoden: lineare, ganzzahlige und netzwerkbasierte Optimierung, stochastische, dynamische, quadratische, nichtlineare; Metaheuristiken, naturanaloge Verfahren

## 2. Simulationssysteme

Ausprobieren von verschiedenen Einflüssen, Lösungsideen und Konfigurationsvarianten

Es geht nicht um mathematisch optimale Lösungen, sondern darum, Systemverhalten und Beeinflussungsmöglichkeiten zu verstehen

z.B. Flugsimulator, Simulation Wartschlange im Supermarkt, Ampelschaltungen

## 3. Business Intelligence

Oberbegriff für informationstechnische Instrumente zur Auswertung von unternehmensweit verfügbaren Fakten

Unterschied zur Optimierung und Simulation: nicht primär modellbasiert, sondern Aufbau auf gesammelten Daten

Anwendungsgebiete: Erfahrungswerte, z.B. für Bestellungen, Kaufverhalten von Kunden

## 4. Multicriteria Decision Making

Entscheidung liegen so viele Kriterien zugrunde, dass Mensch nicht in der Lage ist, Abschätzung vorzunehmen

## 5. Strategische Entscheidungsunterstützung

Mithilfe von Optimierung und Simulation können aus Ist-Zustand Verbesserungen für nahe Zukunft abgeleitet werden, während Business Intelligence Daten der Vergangenheit für Entscheidungen der Gegenwart nutzt

Methoden: Szenariomanagement, War Gaming

2.-4.: Vertiefung EUS

## Optimierungssysteme

Produktionsprogramm: gewinnmaximale Kombination der Produktionsmengen verschiedener Typen/Produkte, die in Fertigung gehen sollen

Einsatz in vielen betrieblichen Bereichen

z.B. Produktionsplanung, -steuerung, Logistik, Finanzplanung, Personaleinsatz, Marketing

## Elemente eines Optimierungsproblems

Entscheidungsvariablen: für Ziel relevante Größen, die wir beeinflussen können

Zielfunktion: eine Größe soll maximiert oder minimiert werden

Restriktionen: diese Einflussgrößen (Variablenwerte) können nicht beliebig gewählt werden

## Modellierung von Entscheidungssituationen

Modell: Abstraktion der Realität, die Entscheidungsproblem in seinen wesentlichen Aspekten korrekt abbildet

Modellierung: Bildung des abstrakten Modells aus der verbalen Beschreibung des Problems in einer für eine computerbasierte Darstellung geeigneter Form

Reduzierte Kosten (für eine Variable): durch „Nicht-Produktion“ vermiedener Verlust; Grenzkosten

Schattenpreis (einer Restriktion): Grenzpreis der Begrenzung des Lösungsraumes durch diese Restriktion

Nichtnegativitätsbedingung

## Spezialfälle

Keine optimale Lösung (oft Zeichen für falsche Modellierung)

Mehrdeutige Lösung

Degenerierte Lösung

## Arten von Optimierungsproblemen

Ganzzahlige Variablen: INT

Binäre oder logische Variablen: (nur 0 oder 1) BIN

# Systementwicklung

Beinhaltet die **Gesamtheit der planenden, analysierenden, entwerfenden, ausführenden und prüfenden Tätigkeiten** zur Schaffung eines neuen oder Änderung eines bestehenden Informationssystems

Operative Systeme (TPS)

Management-Informationssysteme (MIS)

Entscheidungsunterstützungssysteme (DSS)

Führungsunterstützungssysteme (ESS)

Standard-Informationssysteme für verschiedene Bereiche:

- SAP (Abwicklung sämtlicher Geschäftsprozesse im Unternehmen)
- Oracle Database (Datenbank-Managementsystem)
- Lotus Notes (Groupware, E-Collaboration)

Es gibt aber nicht nur Standard-Probleme! → Software muss für spezielle Anwendungsfälle selbst entwickelt werden

Gründe für den Einsatz von Individualsoftware: keine Standardsoftware verfügbar, geringere Entwicklungskosten, Datensicherheit, Kontrolle über Quellcode, Erweiterbarkeit, Wettbewerbsvorteil

Trennung von Spezifikation und Konstruktion

Spezifikation: Festlegung, was ein System leisten soll, maßgeblich: Anwender

Konstruktion: Festlegung, wie Anforderungen erfüllt werden sollen, maßgeblich: IT-Fachleute

Jedes Systementwicklungs-Problem hat aus Sicht des Unternehmens **einmaligen Charakter**

Dennoch normalerweise allgemeingültiges **Grundschema**

## Vorgehensmodelle

### Sequentielle Modelle

In 60ern entstanden

Relativ statische Vorgehensweise

Gut geeignet für relativ große Projekte

**Wasserfallmodell**

**V-Modell**

### Agile Vorgehensmodelle

Flexibel, chaotisch

In 90ern entstanden

Nur generelle Leitlinien und Prinzipien

Geeignet für kleine Projekte

**Extreme Programming (XP)**

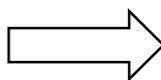
**Crystal**

### Evolutionäre/Inkrementelle Modelle

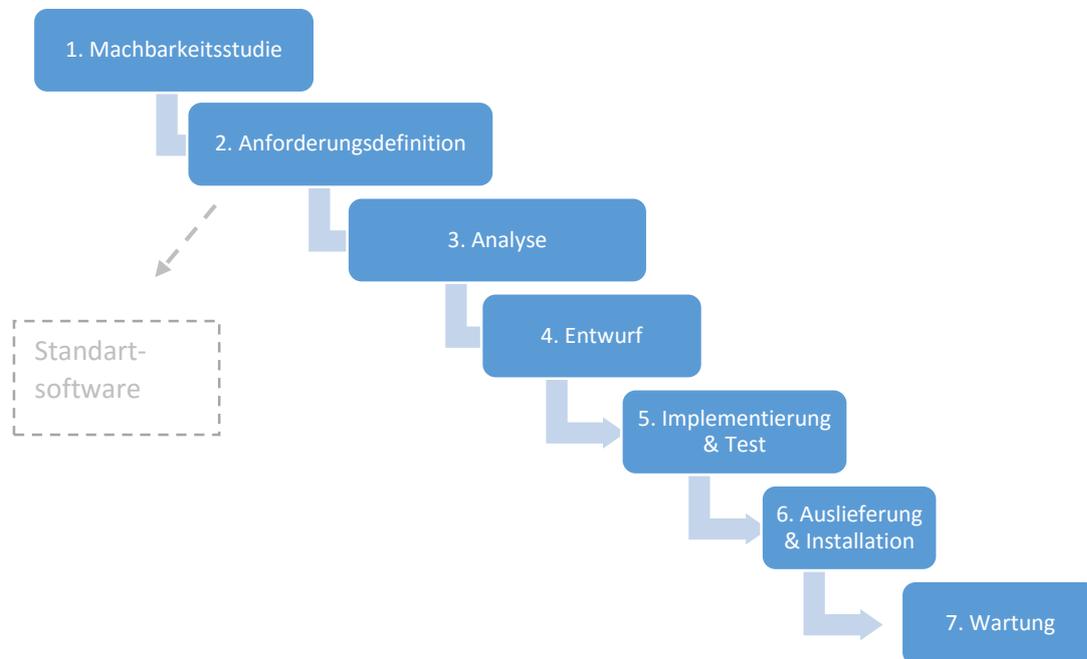
**Überlappendes Phasenmodell**

**Prototyping**

**Spiralmodell**



# Das Wasserfallmodell



## 1. Machbarkeitsstudie

Abschätzen von Kosten und Ertrag geplanter Systementwicklung

Aufgaben: Problem beschreiben, Lösungsansätze erarbeiten, Kosten abschätzen, Angebot erstellen

Ergebnisse: **Lastenheft** (Beschreibung der Anforderungen des Auftraggebers), Kalkulation und Projektplan, Angebot an Auftraggeber

## 2. Anforderungsdefinition (Soll-Konzept)

Exakte Festlegung, **was** System leisten soll

Aufgaben: Ist-Analyse: Beschreibung Abläufe und Begriffe des Problembereichs, Soll Konzept: Festlegung Systemeigenschaften wie Funktionalität und Benutzerschnittstellen im Pflichtenheft

Ergebnisse: **Pflichtenheft** = Anforderungsdefinitions-Dokument, Testplan

## 3. Analyse

Analyse der Anforderungen, weitere Verfeinerung

Aufgaben: Verfeinerung Soll-Konzept, Erstellen grober Architektur, Beschreibung Ablauf und Struktur des zu erstellenden Systems

Ergebnis: große Systemarchitektur (Modul-Definition), Analysedokument

## 4. (System-)Entwurf

Exakte Festlegung, wie die Funktionen der Software zu realisieren sind

Aufgaben: Programmieren-im-Großen = Softwarearchitektur, Grobentwurf, Feinentwurf

Ergebnisse: Ausarbeitung Module, detaillierte Testpläne für Module und Komponenten,  
**Entwurfsdokument**

#### 5. Implementierung und Test

Eigentliche Umsetzungsphase

Realisierung, Test und Integration der Module in System

Aufgaben: **Implementieren** in Programmiersprache, schrittweises Testen und Implementieren, Systemtests ( $\alpha$ -Version)

Ergebnis: Quellcodes und deren Dokumentation, **erstes fertiges System** ( $\alpha$ -Version), Benutzerhandbuch

#### 6. Auslieferung und Installation

Auslieferung und Inbetriebnahme der Software beim Kunden

Aufgaben: Auslieferung an ausgewählte Benutzer ( $\beta$ -Version), Integration in Zielumgebung, Auslieferung an alle Benutzer

Ergebnis: **fertiges System** ( $\beta$ -Version), Akzeptanztestdokument

#### 7. Wartung

Systemwartung ca. 60% der gesamten Kosten

Aufgaben: Fehlerbehebung, Vornehmen von Anpassungen, Einarbeitung Verbesserungen

Ergebnisse: Fehlerbeschreibungen, Veränderungsvorschläge, **neue Versionen** (Releases)

Vorteile Wasserfallmodell: Einteilung in klar abgegrenzte Phasen, Strukturierung Entwicklungsprozess, phasenweise Ergebnisplanung und -kontrolle, Komplexitätsreduktion durch Teilphasen, Einsatz spezifischer Methoden und Werkzeuge

Nachteile: rein sequentielle feste Anforderungen oft unrealistisch, **Qualität Pflichtenheft** kritisch, häufig erst später Fehlererkennung

Zu Beginn des Projekts nur ungenaue Kosten- und Ressourcenabschätzungen möglich

Selbst gutes Pflichtenheft kann nie Umgang mit fertigem System ersetzen

Daher oft erst spätes Feedback

In einigen Fällen vollständige Erstellung eines Pflichtenheftes unmöglich

Reale Softwareentwicklungs-Projekte nie so sequentiell, wie das Wasserfallmodell impliziert

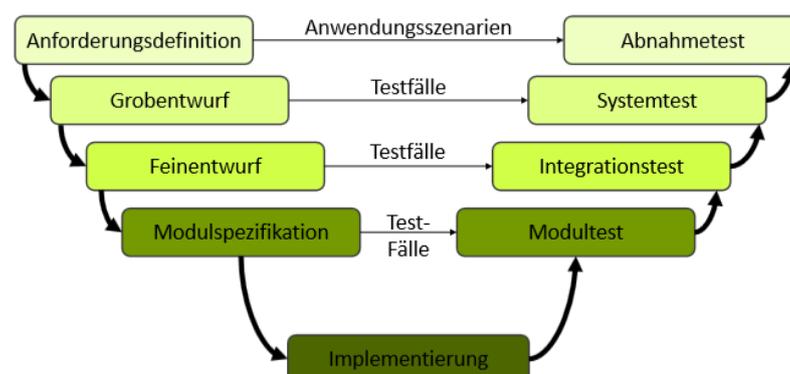
## Alternative Vorgehensweisen

### Das V-Modell

Betonung auf Qualitätssicherung

Erweiterung des Wasserfallmodells

### Alternative Vorgehensmodelle – Das V-Modell



## Prototyping

Ausführbare Vorversion eines Informationssystems, das zur genauen Erfassung der Anforderungen dient

Oft Verwendung von reinen Oberflächen-Prototypen ohne dahinterliegende Funktionalität und Datenhaltung

u.a. keine Berücksichtigung der Aspekte Skalierbarkeit, Qualitätssicherung und Wartbarkeit

## Extreme Programming

„Agile Entwicklung“ mit Betonung auf Programmierbarkeit

**Wenige Regeln**, die laufend nach Projektbedingungen adaptiert werden

Starker Fokus auf Implementierung

Kleine Entwicklungsteams und häufige Iterationen

**Pair Programming** gegen Qualitätsrisiken

Vorgehen in vielen **Open-Source-Projekten**

## Machbarkeitsstudie – Das Lastenheft (Wasserfallmodell)

1. Beschreibung des Ist-Zustandes
2. Beschreibung der Ziele des Projekts
3. Beschreibung der Schnittstellen – mit Benutzern und anderen Systemen?
4. Funktionale Anforderungen – was soll System können?
5. Nichtfunktionale Anforderungen – Benutzbarkeit, Zuverlässigkeit, Effizienz, ...
6. Erste grobe Skizze des Entwicklungszyklus
7. Lieferumfang und Abnahmekriterien

## Anforderungsdefinition (Soll-Konzept) – Das Pflichtenheft (Wasserfallmodell)

Im Lastenheft Beschreibung der Anforderungen

Darauf aufbauend Darlegung von Lösungsansätzen im **Pflichtenheft**

Grundlage jeder Softwareentwicklung

1. Zielbestimmung – als Text
2. Produkteinsatz – Problemumfeld, welche Strukturen und Abläufe
  - 2.1 Beschreibung Problembereich – als Text, Illustrationen
  - 2.2 Glossar – textuelle Nachschlagewerk für Fachbegriffe
  - 2.3 Modell des Problembereichs – grafische Notation der Strukturen (UML) (Klassendiagramm)
  - 2.4 Geschäftsprozesse – grafische Notation der Prozesse und Anwendungsfälle (UML) (Aktivitätendiagramm)
3. Produktfunktionen – grafische Notation der Anwendungsfälle (UML) (Use Case Diagramm und Aktivitätendiagramm, Verteilungsdiagramm)
4. Produktcharakteristiken – textuelle Beschreibung der nicht-funktionalen Anforderungen (Wartbarkeit, Benutzerfreundlichkeit)

Widerspruchsfreies, quantitativ formuliertes Zielsystem

## Formale Sprachen

Vermeidung Mehrdeutigkeit

Informale: definierte Syntax, keine präzise Semantikdefinition, Mehrdeutigkeiten möglich

Formale: definierte Syntax, präzise definierte Semantik, Bsp.: arithmetische Ausdrücke

70er: Datenflussdiagramme und Entity-Relationship Diagramme

90er: mehr als 50 objektorientierte Modellierungssprachen

2000: Standardisierung durch die Object Management Group (OMG)

**UML**: Unified Modeling Language

Standardisierte Notation für Analyse, Entwurf und Dokumentation von Informationssystemen

UML Diagramme zur Visualisierung von Strukturen und Abläufen im Rahmen des Pflichtenheftes

Use Case Diagramme

Aktivitätsdiagramme

Verteilungsdiagramme

## Andere Standards: ARIS

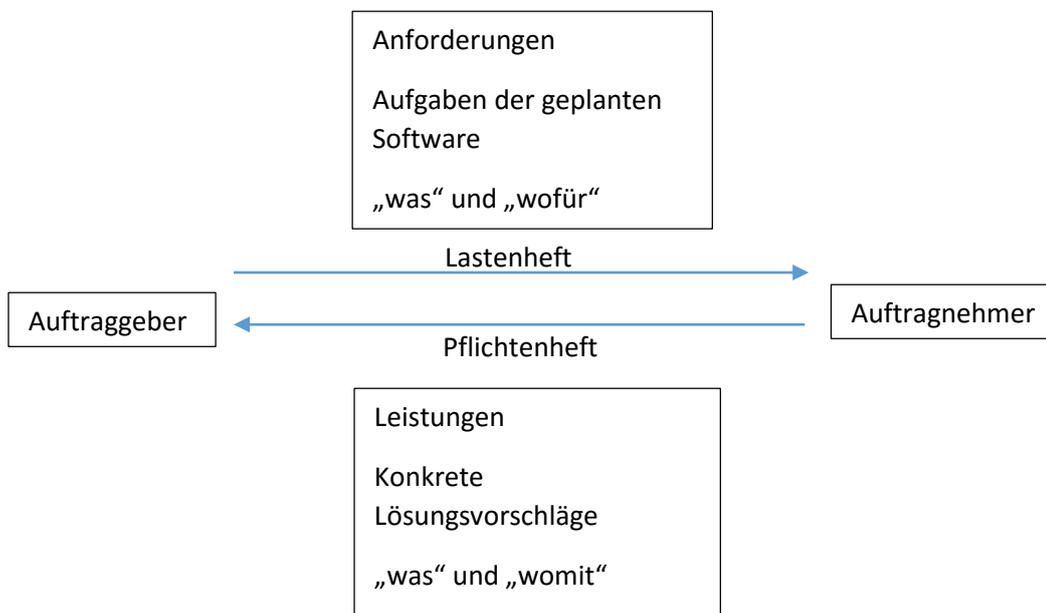
Architektur integrierter Informationssysteme – Industriestandard für Analyse und Entwurf betrieblicher Informationssysteme, der ganzheitliche Betrachtung von Geschäftsprozessen anstrebt

Sichten: Organisationssicht, Funktionssicht, Datensicht, Leistungssicht, Steuerungssicht

Beschreibungsebenen: Fachkonzept, Datenverarbeitungskonzept, Implementierung

## EPK Diagramme

Ereignisgesteuerte Prozessketten (ePK): in Steuerungssicht verwendete Modelle zur Abbildung des Prozessmodells; zentraler Bestandteil von SAP-Referenzmodellen und ARIS



## Standardsoftware

Customizing: Anpassung von Standardsoftware an konkreten Anwendungsfall

Vorteile Standardsoftware gegenüber Individualsoftware:

- geringeres Entwicklungsrisiko
- häufig bessere Softwarequalität (Fehlerfreiheit, Stabilität)
- meist geringere Kosten
- Time-to-market kürzer

Nachteile von Standardsoftware gegenüber Individualsoftware:

- lässt sich oft nicht an spezifische Anforderungen des Unternehmens anpassen
- Abhängigkeit vom Anbieter (z.B. Konkursgefahr bei Start-ups)

## Total Cost of Ownership von IT-Systemen

**Gesamtkosten**, die sich durch Anschaffung, Entwicklung und Betrieb **eines Informationssystems** über die gesamte Nutzungsdauer ergeben

## Software-Architektur

Leitet sich aus (nicht-funktionalen) Zielen ab

Definition von Modulen und deren Interaktionen

### 3-Schichten-Architektur:

- Präsentationsschicht
- Anwendungs-/Logistikschicht
- Datenhaltungsschicht

### Model-View-Controller (MVC-Prinzip):

- Analyzer/DataManager
- Form
- Controller

Serviceorientierte Architektur (SOA)

### Software-Architektur setzt sich zusammen aus:

- Modulen** mit bestimmten Eigenschaften/Verhalten
- Beziehungen** zwischen den Modulen
- Beschreibungen der **erlaubten/verbotenen Interaktionen**

Jeder Architekturstil ist Kompromiss, keiner kann alles

Analyse legt große Strukturen/Architektur fest

## Entwurf

Im Systementwurf wird exakt festgelegt, **wie die Funktionen des Systems zu realisieren** sind, alle wichtigen Implementierungsentscheidungen werden getroffen

Spezifiziert Ausrichtung auf Plattform und Programmiersprache

Grobentwurf

Feinentwurf

## Implementierung

Erstellung eines **lauffähigen**, qualitativ hochwertigen **Softwaresystems** mit zugehöriger **Dokumentation**

Algorithmus: **endliche Folge von Anweisungen**, deren schrittweise Ausführung eine gestellte Aufgabe löst

Programm: Umsetzung eines Algorithmus mit konkreter Programmiersprache

Programmiersprachen:

C++, C#, Java (Universalsprachen, oft plattformunabhängig)

FORTRAN (mathematische Problemstellungen, besonders schnelle Rechenoperationen, z.B. MOPS)

PHP, Perl, Python, Ruby, JavaScript (webbasierte Anwendungen)

Visual Basic for Applications **VBA** (Entwicklung einfacher Tools zur Erweiterung der Funktionalität von **MS Office**)

## Codegenerierung

CASE (Computer Aided Software Engineering) Tools erlauben Modellierung, Reverse-Engineering, teilweise Codegenerierung

Probleme bei der Codegenerierung: oft unvollständige Verhaltensdiagramme, schwer zu standardisieren

Model Driven Architecture (MDA): OMG Initiative zum Model Driven Development (MDD), Modelle als Zentrum der Systemerstellung, Code soll vollständig automatisch generiert werden

## Testen

Systematisches Ausführen eines Programms mit Ziel, Fehler zu finden und Qualität nachzuweisen

## Auslieferung und Installation

Erstellung eines Einführungsplanes

Übergabe der Software inklusive Dokumentation

System-Installation

Datenmigration aus Alt-System

Personelle und organisatorische Vorbereitung des Systembetriebs

Schulung der Benutzer des Systems

Inbetriebnahme des Systems

## Dokumentation

Benutzerhandbücher, Systemdokumentation, Installations- und Administrationsdokumentation

Vollständige, zielgruppenspezifische Dokumentation entscheiden für Akzeptanz durch Benutzer

## Veröffentlichung

Bei webbasierten Systemen, die sich an **nicht klar umrissene Anwendergruppen** richten, wie z.B. Portale oder E-Shops, wird nach der Veröffentlichung eine systematische Site-Promotion-Kampagne durchgeführt

Online-Promotion

Offline-Promotion

## Wartung

Wartung: laufende Fehlerbeseitigung

Pflege: Modifikation des Systems

Risiko- und Sicherheitsmanagement: Einspielen sicherheitskritischer Softwareupdates, regelmäßige Datensicherung

Kapazitätsmanagement: Aufstockung Hardware bei gestiegenen Benutzerzahlen

Ausfallmanagement: Planung von Notfallmaßnahmen bei Systemausfall

### Verteilung des Arbeitsaufwands

Machbarkeitsstudie + Anforderungsdefinition: 10%

Analyse + Entwurf: 20%

Implementierung & Test: 20%

Auslieferung & Installation + Wartung: 50%

### Vor- und Nachteile beim Wasserfallmodell

Vorteile: klarer Ablauf, Prozess beinhaltet in sich abgeschlossene Stufen, leicht identifizierbare Meilensteine und Auslieferung

Nachteile: unflexibel bei sich ändernden Anforderungen, inkonsistente Anforderungen werden mitunter erst bei der Implementierung erkannt, Tests werden erst sehr spät durchgeführt, der Kunde bekommt das Produkt erst bei der Auslieferung zu sehen

### Schätzproblem der Wirtschaftsinformatik

Das Schätzen von Aufwandsgrößen bei langfristigen komplexen Projekten ist a priori oft nur schwer umzusetzen

Regel 1: Trenne die Preiskalkulation von der Schätzung von Aufwandsgrößen

Regel 2: Schätze nie Gesamtgrößen, sondern zerlege das Projekt in Komponenten und Prozesse und schätze den Aufwand getrennt für diese Teile

Regel 3: Trenne die Durchlaufzeit vom erforderlichen Arbeitsaufwand

Regel 4: Schätze immer konkrete Größen

Regel 5: Bei vernetzten Arbeiten steigt die Projektdauer mit zunehmender Mitarbeiterzahl durch den erforderlichen Koordinationsaufwand

### Extreme Programming (XP)

Vorgehen, welches insbesondere für kleine Teams sinnvoll ist, wenn Anforderungen vage sind und sich schnell verändern können

Hauptziel: Änderungskosten gering halten

### Motivation

Anpassung an **natürliche Gegebenheiten** im Software-Entwicklungsprozess

Änderungen müssen zugelassen werden und organisatorisch umsetzbar bleiben

Anforderungen können oft erst durch Einsatz von Prototypen geklärt werden

Frühe Verfügbarkeit der **Grundfunktionalität** oft entscheidend

Funktionen mit zeitintensivem Entwicklungsaufwand und niedrigem Nutzen können interaktiv hinzugefügt werden (80:20 Regel)

[Weitere agile Vorgehensmodell: Scrum oder Kanban, wird nicht weiter vertieft]

## Prinzipien agiler Methoden

### Das agile Manifest:

Individuals and interactions over processes and tools  
Customer collaboration over contract negotiation  
Working software over comprehensive documentation  
Responding to change over following a plan

### Die zentralen XP-Werte:

Kommunikation  
Respekt  
Einfachheit  
Feedback  
Mut

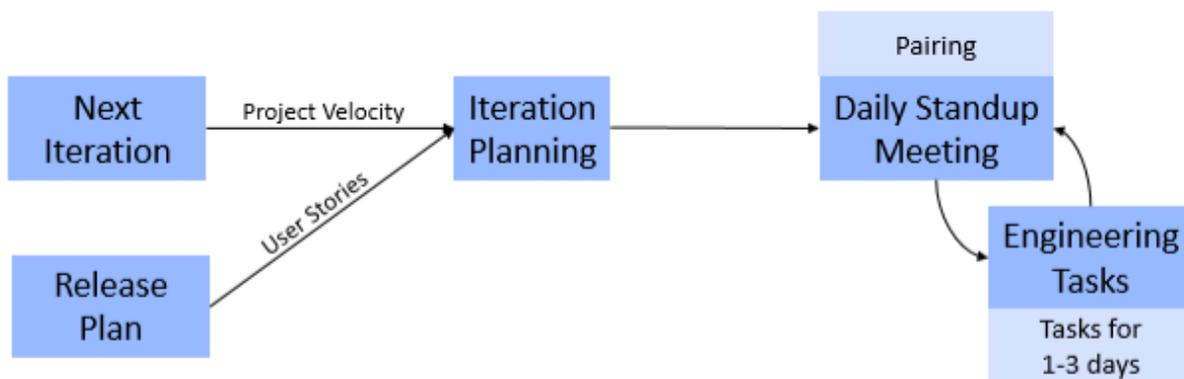
## XP

Starker Fokus auf Implementierung statt vorheriger Planung

Kleine Entwicklungsteams und häufige Iterationen (halten Aufgaben klein und überschaubar)

Reaktion auf immer komplexer werdenden Managementprozess

Der **einfachste Entwurf**, der alle Testfälle besteht, wird implementiert



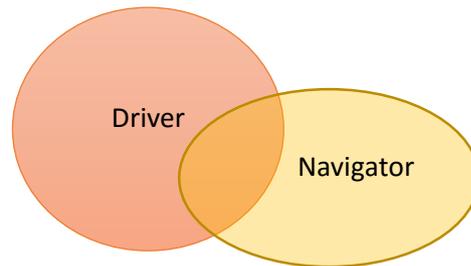
## Pair Programming

### Kein Wissensmonopol:

Know-how einzelner Funktionalitäten liegt nicht bei einzelnen Entwicklern  
Ausscheiden eines Mitarbeiters leichter aufzufangen  
Wissen verbreitet sich im Team  
„Abgreifen“ von externem Expertenwissen

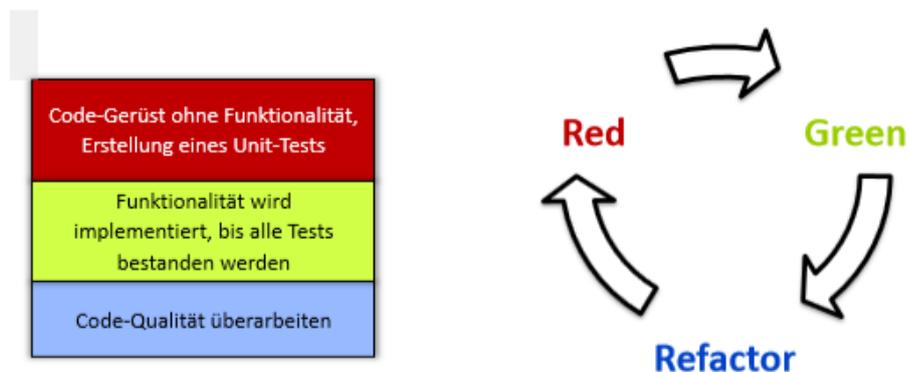
### Effizienz:

Bessere Qualität (gemessen an Anzahl bestandener Testfälle)  
Weniger Zeitaufwand  
Weniger Lines-of-Code  
Größere Zufriedenheit der Mitarbeiter  
Konzentration auf Wesentliches



### Test Driven Development

Szenarien werden als Testfälle vor Implementierung festgelegt  
Integration: nicht neue Funktionalität testen, sondern Gesamtsystem  
Grundregel: je nach Komplexität gibt es pro Funktionalität ein bis zehn Tests  
Module und Funktionen werden nur hinzugefügt, wenn alle Testfälle bestanden werden



### Kundeneinbindung und Releases

Zu Beginn des Projekts werden Anforderungen mit Kunden in Form von User Stories festgelegt

**Projektfortschritt** wird in Anzahl **erfolgreich umgesetzter User Stories** angegeben

Kunde kann anhand **kleinerer Release-Zyklen** zeitig eingreifen, wenn entwickelte Funktionalität nicht seinen Wünschen entspricht

### Vorteile und Nachteile von XP

#### Vorteile:

**Flexibilität** bei sich ändernden Anforderung wird gewahrt  
Dynamische Vorgehensweise mit **wenig Overhead** für Prozess-Management  
**Offene Kommunikation** über Fehler und Ängste

#### Nachteile:

Ungeeignet, wenn es auf beweisbare Programmeigenschaften ankommt  
Komplettpaket, wobei für einzelne Methoden Voraussetzungen nicht passen können  
Agiles Vorgehen wird noch **zu oft als Allheilmittel angesehen**, sobald eine Entwicklung problematisch wird

Agiles Vorgehen kann man nicht „einfach so“ einführen, da es weitreichende Umstrukturierungen, mitunter auch in Denk- und Kommunikationsprozessen nach sich zieht. Oft wird es als Ausrede herangezogen, wenn ein Wasserfall-Projekt dermaßen aus den Fugen gerät, dass es „agil“ wirkt.

### Agile Projekte grundsätzlich besser?

**Grundproblem** der Software-Entwicklung in agilen Projekten dasselbe

Scheitern aus denselben Gründen wie Wasserfallprojekte

Agile Methoden werden oft als Zwischenschritt zu zukünftigen Weiterentwicklungen gesehen



	Wasserfallbasiert	Agil
<b>Erfolgreich</b>	14%	42%
<b>Herausfordernd</b>	57%	49%
<b>gescheitert</b>	29%	9%

Standish Chaos Report (2011)

### Einführung in VBA

Steuerung von Maschinen per Algorithmus

Syntax: Programmiersprachen haben syntaktische Regeln, die Struktur der jeweiligen Programmiersprache vorgeben, keine Fehlertoleranz bei syntaktisch falschen Anweisungen

Damit **Programmiersprache** vom Computer umgesetzt werden kann, muss man sie in **Maschinensprache** übersetzen

Semantik: Absicht des Entwicklers/Anwenders (was soll Programm machen?)

### Variablen und Datentypen

**Variablen** sind Platzhalter, die zum Speichern von Werten verwendet werden. Sie haben **Namen** und **Datentypen**.

Datentyp	Erklärung	Beispiel
<b>String</b>	Zeichenketten	Hallo
<b>Integer, Long</b>	Ganzzahl	12345
<b>Single, Double</b>	Gleitkommazahl	243,657
<b>Boolean</b>	Wahrheitswert	Wahr, Falsch
<b>Variant</b>	Nimmt Daten jedes Typs auf. Wird von VBA standardmäßig verwendet. Vermeiden, da langsame Verarbeitung – und weil ihr es lernen sollt 😊	

# Zusammenfassend – Verzweigungsvarianten

1. Ein Anweisungsblock <b>wird nur ausgeführt</b> , wenn die Bedingung wahr ist.	<b>If</b> Bedingung <b>Then</b> Anweisungsblock <b>End If</b>
2. Es wird <b>entweder</b> Anweisungsblock_1 <b>oder</b> Anweisungsblock_2 ausgeführt abhängig davon, ob die Bedingung wahr oder falsch ist.	<b>If</b> Bedingung <b>Then</b> Anweisungsblock_1 <b>Else</b> Anweisungsblock_2 <b>End If</b>
3. Es wird genau der Anweisungsblock ausgeführt, <b>dessen Bedingung wahr</b> ist. Ist keine der Bedingungen wahr wird auch kein Anweisungsblock ausgeführt	<b>If</b> Bedingung_1 <b>Then</b> Anweisungsblock_1 <b>ElseIf</b> Bedingung_2 <b>Then</b> Anweisungsblock_2 <b>End If</b>
4. Es wird der Anweisungsblock ausgeführt, <b>dessen Bedingung wahr</b> ist. <b>Ist keine der Bedingungen wahr</b> , wird Anweisungsblock_else ausgeführt.	<b>If</b> Bedingung_1 <b>Then</b> Anweisungsblock_1 <b>ElseIf</b> Bedingung_2 <b>Then</b> Anweisungsblock_2 <b>Else</b> Anweisungsblock_else <b>End If</b>

## Datenmodellierung

### Business Intelligence und Datenbanken

Business Intelligence: Technik zur Konsolidierung, Analyse und Bereitstellung von Daten zur Entscheidungsunterstützung

Fußt in der Regel auf umfangreichen, operativen Datenbanken/Datenquellen

### Datenbanksysteme

Dienen

- 1) Speicherung großer Datenbestände
- 2) Auswertung dieser

Teil fast alle Anwendungssysteme (Wissensmanagementsysteme, Supply Chain Management-System, Customer Relationship Management-Systeme, ERP-Systeme)

Architektur ERP-System: zentrale Datenbank, mehrere Softwaremodule → Vermeidung Dateninkonsistenz

### Aufbau eines Datenbanksystems

Datenbanken + Datenbankverwaltungssystem = Datenbanksystem

Datenbank, welche eigentliche Daten enthält

*Datenbankverwaltungssystem (DBMS Data Base Management System):*

**Abfrageinterpreter:** Übersetzt SQL-Anweisungen in Programmaufrufe entsprechender "low level"-Routinen des DBMS

**Abfrageoptimierer:** Sucht für eine gegebene SQL-Abfrage an Hand von Statistik- und Indexdaten einen "guten" Zugriffspfad auf die gewünschten Daten

**Integritätskontrolle:** Stellt sicher, das Daten nur entsprechend vorher definierter Integritätsbedingungen geändert werden können

**Autorisierungskontrolle:** Überprüft, ob der Nutzer auf die Daten zugreifen darf

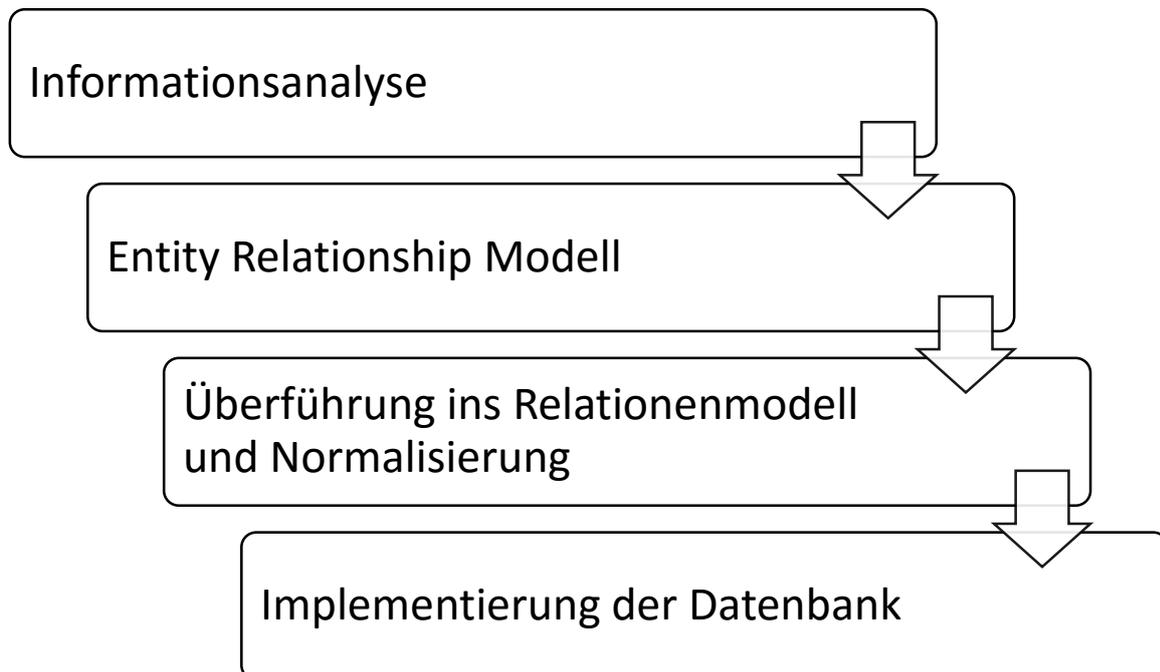
**Kontrolle von Mehrfachzugriffen:** Synchronisiert Mehrfachzugriffe und sperrt temporär Datenzugriffe, die zu Konflikten führen würden

**Datensicherung:** Maßnahmen um Folgen von Systemabstürzen oder anderen Fehlersituationen klein zu halten

**Dateimanagement:** Speicherallokation, Zugriffe auf die Festplatte

## Datenbankmodellierung

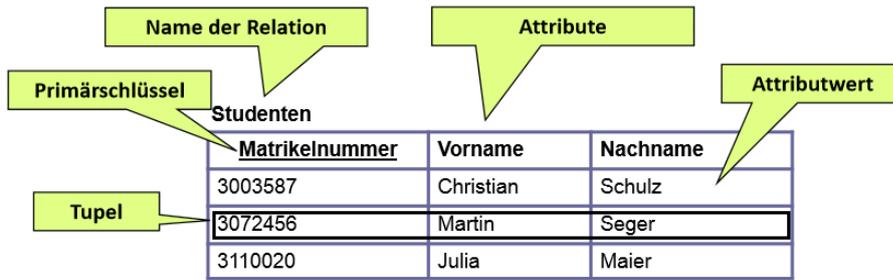
Phasenschemata des Datenbankentwurfs:



1) Das Relationenmodell

**Relationstyp:** abstrakte Beschreibung einer Tabelle (=Relation); Definition von Attributen, Datentyp, Primärschlüssel und Tabellename

Schreibweise: Studenten(Matrikelnummer, Vorname, Nachname)



Kandidatschlüssel:

Jedes Tupel muss durch einen Schlüssel eindeutig identifizierbar sein

Als Schlüssel kann einzelnes Attribut oder Attributgruppe dienen

**Jedes Attribut oder jede minimale Attributgruppe, die Objekt eindeutig und zeitinvariant identifiziert = Kandidatschlüssel**

Es kann in einer Relation mehrere Kandidatschlüssel geben

Primärschlüssel:

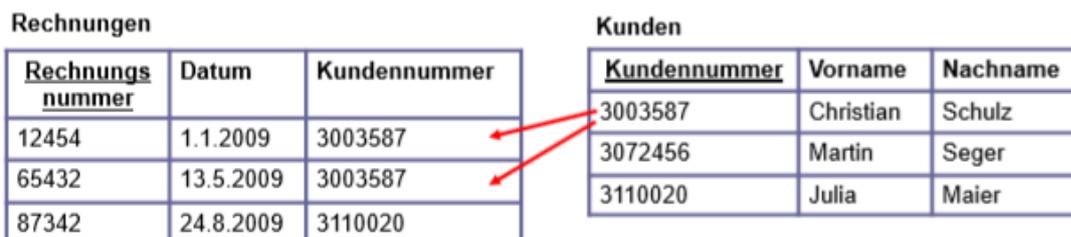
**Der ausgewählte Kandidatschlüssel**

Kennzeichnung durch Unterstreichung

Fremdschlüssel:

Dienen der Verknüpfung mehrerer Relationen

**Attribut oder Attributgruppe in Relation B, die gleichzeitig Primärschlüssel in Relation A ist = Fremdschlüssel**



- Das Attribut Kundennummer ist Primärschlüssel in der Relation *Kunden* und Fremdschlüssel in der Relation *Rechnungen*.

## 2) Structured Query Language (SQL)

Standardisierte Sprache zum Arbeiten mit Datenbanken

Umfasst Teilbereiche:

Data Definition Language (DDL): Erstellen und Verändern von Tabellen, Schlüsseln und Indices

Data Manipulation Language (DML): Daten einfügen, ändern und löschen

Data Control Language (DCL): Vergabe von Zugriffsrechten

Query Language (QL): Abfrage der Daten

Verarbeitung von SQL-Abfragen:

FROM: definiert Ausgabestellen

WHERE: selektiert der Bedingung genügende Tupel

ORDER BY: sortiert Tupel

SELECT: selektiert Attribute

„Strings“ = konstante Zeichenketten, werden mit „“ gekennzeichnet

## 3) Entity-Relationship-Modellierung

Entity-Relationship-Diagramme dienen Modellierung und Visualisierung von Datenstrukturen

Entity:

Objekttypen, die im Anwendungsfall Rolle spielen, z.B. Kategorien, Artikel, Rezensionen, Kunden

Attribut:

Definieren Eigenschaften von Entities

Haben bestimmte Wertebereiche (Domains)

Beispiel: Termin einer Lehrveranstaltung (Wertebereich: Datum)

Relationship:

Beziehungen zwischen Entities, z.B. Kunden tätigen Bestellungen, Artikel gehören zu Kategorien

## Darstellung von Beziehungen

---

- ▶ Es gibt mehrere zulässige Darstellungsformen:

### 1. Darstellung ohne Beziehungs-Bezeichnung



### 2. Darstellung mit Beziehungs-Bezeichnung



### 3. Darstellung mit Raute



- ▶ Die Rautendarstellung muss bei
  - ▶ n:m-Beziehungen und
  - ▶ Beziehungen mit eigenen Attributen benutzt werden.

## Kardinalitäten

---

### ▶ Kardinalitäten

- ▶ Kardinalitäten legen fest, mit wie vielen anderen Objekten ein einzelnes Objekt minimal und maximal in Beziehung stehen kann.

### ▶ Schlageter-Stucky-Notation



- ▶ Eine Abteilung hat einen oder mehrere Mitarbeiter.
- ▶ Ein Mitarbeiter arbeitet in genau einer Abteilung.

### ▶ Kardinalitätenkürzel:

- ▶ k : genau k-mal. Z.B. 3
  - ▶ [n,m] : mindestens n, maximal m. Z.B. [2,5]
  - ▶ \* : Null oder mehr
  - ▶ + : 1 oder mehr
  - ▶ c : Null oder 1
-

## ER-Modellierung: Empfohlenes Vorgehen:

1. Problemrahmen abstecken
2. Festlegen der Objekttypen
3. Festlegen der Beziehungstypen
4. Festlegen der Kardinalitäten
5. Festlegen der Attribute und Schlüssel

### 4) Relationale Modellierung

#### Relationstyp:

Abstrakte Beschreibung einer Tabelle

Definition Attribute, deren Datentyp, der Primärschlüssel und Tabellename

#### Relation:

Konkrete Ausprägung eines Relationstyps

### 5) Normalisierung

Ziel: Verbesserung eines entworfenen Relationenmodells für operative bzw. Transaktionsdatenbanken

#### **Zerlegung von Relationstypen in Relationstypen höherer Normalform = Normalisierung**

Strukturdefizite von Relationstypen sind durch ihre Zerlegung in Relationstypen höherer Normalform vermeidbar

Strukturdefizite von Relationstypen: Datenredundanzen, Löschanomalien, Repräsentanzprobleme

#### 1. Normalform (1NF)

**Ein Relationstyp befindet sich in 1NF, wenn der Wertebereich jedes seiner Attribute atomar ist.**

Atomarität: zusammengesetzte Attribute, Wiederholungsgruppen und Auflistungen nicht zugelassen

#### 2. Normalform (2NF)

**Ein Relationstyp befindet sich in 2NF, wenn er sich in 1NF befindet und jedes Nichtschlüsselattribut von jedem Kandidatenschlüssel vollfunktional abhängt.**

Funktionale Abhängigkeit: **Das Attribut Y ist funktional abhängig vom Attribut oder der Attributgruppe X, wenn es zu jedem Wert von X nur genau einen Wert von Y geben kann.**

Schreibweise funktional abhängig: Matrikel\_Nr → Name

Schreibweise nicht funktional abhängig: Name  $\nrightarrow$  Vorname

Alle Attribute eines Relationstypen hängen per Definition funktional von allen Kandidatenschlüsseln ab

Vollfunktionale Abhängigkeit: **Das Attribut Y ist vollfunktional vom Attribut oder der Attributgruppe X, wenn Y von X funktional abhängig ist und es keine Teilmenge in X gibt, von der Y abhängig ist.**

Eine **funktionale Abhängigkeit**  $X \rightarrow Y$  kann nur dann **keine funktionale Abhängigkeit** sein, wenn X eine **Attributgruppe** ist.

Schreibweise volle funktionale Abhängigkeit: Matrikel\_Nr, Prüfungsfach  $\Rightarrow$  Note

(Das Symbol  $\Rightarrow$  wird nur bei Attributgruppen verwendet, ansonsten  $\rightarrow$ , weil ein einzelnes Attribut immer vollfunktional abhängig ist, da es keine Teilmenge gibt.)

#### Schlüsselattribut:

Jedes Attribut, das Bestandteil eines Kandidatenschlüssels ist, wird als Schlüsselattribut bezeichnet

#### Nichtschlüsselattribut:

Jedes Attribut, das nicht Bestandteil eines Kandidatenschlüssels ist, wird als Nichtschlüsselattribut bezeichnet.

**Existiert kein zusammengesetzter Schlüssel, so befindet sich ein Relationstyp automatisch in der 2NF (Voraussetzung: 1NF).**

#### Vorgehen zur Prüfung eines Relationstypen auf 2NF:

Prüfen auf 1NF

Vollfunktionale Abhängigkeiten bestimmen

Kandidatenschlüssel, Schlüssel- und Nichtschlüsselattribute bestimmen

Prüfen, ob alle Nichtschlüsselattribute vollfunktional von allen Kandidatenschlüsseln abhängen

### 3. Normalform (3NF)

**Ein Relationstyp befindet sich in 3NF, wenn er sich in 2NF befindet und kein Nichtschlüsselattribut transitiv von einem Kandidatenschlüssel abhängt.**

Transitive Abhängigkeit:  $\underline{A} \rightarrow B \rightarrow C$

(Beispiel: **Matrikel\_Nr**, Name, Vorname, **PLZ, Stadt**)

Anders formuliert: **Ein Relationstyp befindet sich in Dritter Normalform, wenn keine funktionalen Abhängigkeiten zwischen Nichtschlüsselattributen existieren.**

Existieren keine Nichtschlüsselattribute, so befindet sich ein Relationstyp automatisch in 3NF (Voraussetzung: 1NF)

Existiert nur ein Nichtschlüsselattribut, so befindet sich ein Relationstyp automatisch in 3NF (Voraussetzung: 2NF)

#### Gesamtverfahren bei der Normalisierung

- 1) Prüfen ob alle Attribute atomar sind (1NF)
- 2) 2) Vollfunktionale Abhängigkeiten bestimmen (eventuell aufzeichnen)
- 3) Kandidatenschlüssel bestimmen
- 4) Schlüssel- und Nichtschlüsselattribute bestimmen
- 5) Alle nicht vollfunktionalen Abhängigkeiten zwischen Schlüsselattributen und Nichtschlüsselattributen durch Zerlegung auflösen (2NF)
- 6) Alle Abhängigkeiten zwischen Nichtschlüsselattributen durch Zerlegung auflösen (3NF)

# Business Intelligence

## Operative Informationssysteme

### Der Dienstleistungsprozess:

Anbahnung, Vereinbarung, Durchführung, Abrechnung

### Online Transactional Processing:

Betriebliches Basissystem bezieht Einsatzgüter aus Umwelt der Unternehmung und transformiert sie in Leistungserstellungsprozess in Produkte

### Enterprise Resource Planning:

ERP-System: Unternehmensweites Anwendungssystem, das alle zentralen Geschäftsprozesse eines Unternehmens in einziges Softwaresystem integriert und so den reibungslosen unternehmensweiten Informationsaustausch ermöglicht

Dienen terminlicher Steuerung von Arbeitskräften, Werkstoffen, Arbeitsmitteln und Arbeitsplätzen

## Analytische Informationssysteme

### Horizontale/Vertikale Integration:

Lenkungs- und Leistungssysteme

Realisierung der vertikalen Integration: Selektion operativer Daten aus Geschäftsvorfälle; zeitliche, räumliche und produktorientierte Aggregation; gegebenenfalls Anreicherung durch extern verfügbare Daten

Data Warehouse: ist eine Datenbank mit Berichts- und Abfragefunktionen, die operative und historische Daten speichert

Daten werden aus verschiedenen betrieblichen Systemen extrahiert und für Managementberichte und Analysen aufbereitet (→OLAP, Data Mining)

multidimensional aufgebaut

Einsatz für Langzeitspeicherung von historischen, gereinigten, validierten, synthetischen, operativen Daten aus internen und externen Quellen

### Arbeiten mit einem Data Warehouse:

1. Auswahl geeigneter Attribute aus operativen Datenbanken
2. Hinzufügen ausgewählter Daten aus externen Quellen
3. Transformation und Laden der Daten
4. Speicherung der geladenen Daten in Form von Dimensionen
5. Administration von Datenbanken (wie bei operativen Datenbanken)
6. Suche und Analyse mittels Berichten oder mit OLAP-Technologie

Operative Datenbanken versus Data Warehouses:

Operative Datenbanken

- unterstützen primär das tägliche (operativ) Geschäft
- zeichnen operative Geschäftsvorfälle auf
- verfolgen redundanzfreie Speicherung von Transaktionen
- sind relativ unübersichtlich
- nicht sehr benutzerfreundlich

Eigenschaften	Operative Datenbank	Data Warehouse
Operative Daten	●	-
Daten vollständig	●	-
Daten detailliert	●	-
Daten redundanzarm	●	-
Daten änderungsintensiv	●	-
Datenmodell komplex	●	-
Strategische Daten	-	●
Historische Daten	-	●
Abgeleitete Daten (z.B. Mittelwerte)	-	●
Verarbeitung abfrageintensiv	-	●
Abfragen oft 'ad hoc'	-	●
Endbenutzerorientierte Schnittstelle	-	●

**CRM-Anwendungen** analysieren Kundendaten, um Informationen für die Verbesserung der Geschäftsleistung bereitzustellen; einheitliche Sicht auf Kunden

**Repräsentation analytischer Daten** in Würfeln oder mit multidimensionalen Datenmodellen

Fakt = aggregierbares, meist numerisches und kontinuierliches Attribut, das betriebliches Erfolgskriterium mehr mehrdimensional misst

Dimension = symbolisches und diskretes Kriterium, das Auswahl, Zusammenfassung und Navigation eines Indikators ermöglicht

Physikalische Datenrepräsentation:

gespeichert auf Basis eines relationalen oder multidimensionalen Datenmodells

	Relationale Datenbank	Multidimensionale Datenbank
Einfache Abfrage	-	●
Effiziente Abfrage	-	●
Effiziente Speicherung	●	-
Skalierbarkeit	●	-
Kompatibilität	●	-

## Komponenten eines Data-Warehouse-Systems:

### 1) Data Marts

Kleine analytische Datenbanken

Spezialisiert auf Anforderungen einer Gruppe im Unternehmen (z.B. Abteilung oder Arbeitsgruppe)

Beziehen Daten von anderen Data Warehouses oder operativen Datenquellen

Sind mit anderen analytischen Datenbanken zu koordinieren

Entwicklung weniger komplex als bei zentralem Data Warehouse

Verteilung analytischer Daten auf verschiedene Data Marts ist kompliziert (intramodulare Verknüpfung innerhalb eines Data Marts sollte hoch sein; intermodulare gering)

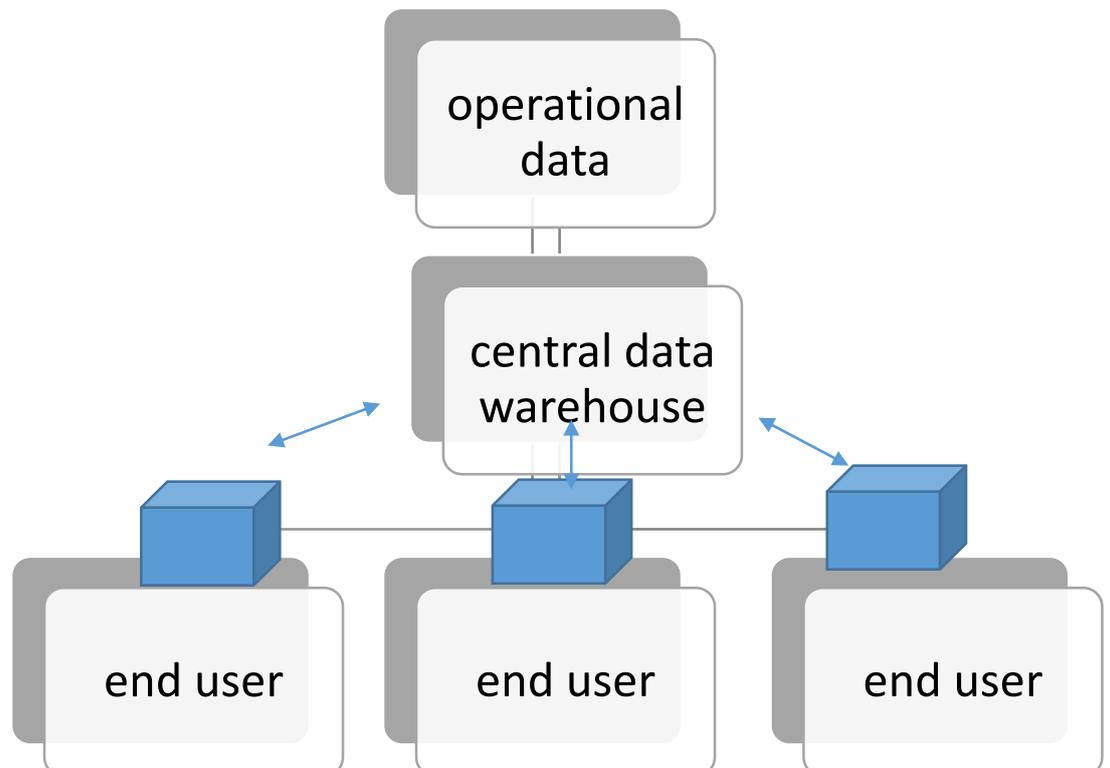
### 2) Zentrales Data Warehouse

Analytische Datenbank, die Daten für lokale Data Marts aufbereitet

Muss nicht unbedingt analytische Daten für gesamtes Unternehmen bereitstellen

### 3) Enterprise Data Warehouse

Bereitstellung analytischer Daten für gesamtes Unternehmen



## Online Analytical Processing

Endbenutzerzugriff auf Data Warehouses

→ Entscheidungsträger benötigen flexiblen und einfachen Zugriff auf analytische Daten, um komplexe Analysen durchführen zu können

- 1) Fest implementierte Berichte
- 2) Abfragesprachen
- 3) OLAP

OnLine Analytical Processing (OLAP): Abfragemethode, die Endbenutzern einen mehrdimensionalen, schnellen Zugriff und eine benutzerfreundliche interaktive Analyse von Daten aus Data Warehouses ermöglicht

OnLine Transactional Processing (OLTP): Verarbeitung von Transaktionsdaten auf Basis operativer Datenbanken

### Standardfunktionen von OLAP:

- 1) Verschiedene Repräsentationsmodi
- 2) Spezielle Würfeloperationen erlauben „Surfen“ durch Daten
- 3) Verschiedene Darstellungsmöglichkeiten

## Case Study Logistikdienstleister

4 r's der Logistik: richtiges Produkt zur richtigen Zeit in richtiger Qualität an richtigem Ort

Angebote von Logistikdienstleistern unterscheiden sich hinsichtlich Art und Gewicht der Sendungen, Laufzeit der Transporte, Preisstruktur

Kurierdienste: individuell begleiteter Transport, Kleinsendungen. Transport in kürzest möglicher Zeit mit hoher Zuverlässigkeit

Expressdienste: Beförderung von Transportgütern ohne Gewichts- und Maßbeschränkungen

Paketdienste: Transport von volumenmäßig beschränkten Kleingütern

## Data Mining

Data-Mining-Tools erlauben automatisierte, komplexe Analyse von Massendaten

Definition: analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner

80% des im Unternehmen relevanten Wissens kann mit konventionellen Tools aus Daten extrahiert werden (Berichte, Abfragesprache (SQL), OLAP und Tabellenkalkulation)

Nachteile konventioneller Tools: Beschränkung auf einfache Fragestellungen, Automatisierung schwierig bzw. unmöglich, Limitierung auf kleine Datenmengen (z.B. Tabellenkalkulation), unzureichende statistische Auswertungsmöglichkeiten, OLAP: fokussiert auf Einzelanfragen mit begrenzter Komplexität

Data-Mining-Methoden beschreiben Beziehungen zwischen zu analysierenden Daten in Form von Modellen und Mustern

Modell: global-gültige Beschreibung des zugehörigen Datensatzes

Muster: beschränkt sich darauf, nur Teilbereich des zugehörigen Datensatzes zu beschreiben

Explorative Datenanalyse (EDA): Ziel ist Analyse den Datensatz ohne fest definierte Strategie

Deskriptive Modellierung: Ziel ist Beschreibung des gesamten Datensatzes

Vorhersagende Modellierung: Ziel ist Bestimmung eines Modells, welchen den Wert einer Variable aus bekannten Werten anderer Variablen vorhersagt

Erkennen von Regeln und Mustern: besondere Phänomene in großen Datensätzen identifizieren

Mustervergleich: Ziel ist, ausgehend von einem gegeben Muster ähnliche Bereiche in einem Datensatz ermitteln

#### Anwendung von Data-Mining-Verfahren:

Definition des Problems und Zuordnung zu einer Data-Mining-Aufgabe, Struktur des Modells und Musters, Definition der Zielfunktion, Such- bzw. Optimierungsmethode, Datenmanagement

#### Clusteranalyse

*“Clustering is the process of grouping data into classes or cluster so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters.”(Han, Kamber 2006)*

Cluster = Sammlung von Datenobjekten, die sich ähnlich sind

Im Data Mining Bestandteil der „Deskriptiven Modellierung“

Hauptzweck: Tool, um Einsicht in Verteilung der Daten zu bekommen; Vorverarbeitung für komplexere Verfahren

Nicht-überwachtes Lernen

Typische Aufgaben:

Marketing: Identifiziere unterschiedliche Kundengruppen und charakterisiere diese anhand von Nachfragemustern

Biologie: Leite Pflanzen-und Tiertaxonomien ab, kategorisiere Gene

(Un)Ähnlichkeitsfunktionen

Spezielle Distanzfunktionen für binär-, nominal- und ordinalskalierte Attribute

Partitionsbasierte Clusteranalyse: Gegeben ein Datensatz aus n Objekten, konstruiert die partitionsbasierte Clusteranalyse k Partitionen der Datenobjekte ( $k \leq n$ )

Jede Partition repräsentiert ein Cluster

Ergebnis: die Datenobjekte sind in k Gruppen klassifiziert

Algorithmus legt zufällige, initiale Partionierung an

## Zusammenfassung Business Intelligence

Techniken zur Konsolidierung, Analyse und Bereitstellung von Daten zur Entscheidungsunterstützung

Datenbanken: Archivierung von Transaktionsdaten

Data Warehouse (OLAP, Data Mining, Abfragen, Berichte): Aufdecken von Mustern und Beziehungen

Business Intelligence: Entscheidungsfindung auf Grundlage der Datenanalysen

Interaktion mit/Reporting für Endnutzer („Manager“)

Analyse mit OLAP, Data Mining-Verfahren, ...

Data Marts als abteilungsspezifischer Auszug des Data Warehouses

Data Warehouses als unternehmensweite analytische Datenbank

Automatisierte Extraktions-, Lade- und Transformationsprozesse

Operative Datenbanken