Datenbanken und Softwareengineering
–
Wintersemester 2015/16

Organisatorisches

- 15 Termine: ca. 13:45-16:15 Uhr, R. 3.217
- Schriftliche Prüfung (60 min. / keine Hilfsmittel)
  - Alle Themenbereiche, Vorlesung und Praktikum
  - Evtl. weitere Vorbereitungsstunde
- Material steht online zur Verfügung unter: www.malte-wattenberg.de
- Literaturhinweise je Kapitel oder Thema
- Kontakt:
  - malte.wattenberg@t-online.de
  - 0163 – 68 60 877

02.10.2015
09.10.2015
16.10.2015
23.10.2015
30.10.2015
06.11.2015
13.11.2015
20.11.2015
27.11.2015
04.12.2015
11.12.2015
18.12.2015
08.01.2016
15.01.2016
22.01.2016

Ziele der Veranstaltung

- Sie kennen die Grundbegriffe zu dem Thema Datenbanken, können aus Vorgaben Datenbanken konzipieren und einfache Abfragen ausführen
- Sie beherrschen die Nutzwertanalyse
- Sie kennen die gängigen Vorgehensmodelle im Softwareengineering sowie ihre Vor- und Nachteile
- Sie kennen die Business Process Model Notation (BPMN) sowie die gängigsten UML Diagramme und können sie anwenden
- Sie wissen, was (IT-) Projekte sind

Organisatorisches

Inhalt

Teil1: Datenbanken
Übersicht
ER-Modell
Rel. Datenmodel
SQL

Teil2: Softwareengineering
Nutzwertanalyse
Vorgehensmodelle
Geschäftsprozessmanagement
UML Diagramme
IT-Projekte

Datenbanken

Lesen..Nachschlagen..Lernen..

Literatur zu Grundlagen von Datenbanken

- Laudon, Kenneth C.: Wirtschaftsinformatik: eine Einführung
  Pearson Studium, 2010. 2. Aufl.
  Kapitel 6 Datenorganisation und
  Datenmanagement (S. 285-329)

- Hansen, Hans Robert:Wirtschaftsinformatik 2,
  Stuttgart, 2005. 9. Aufl.
  Kapitel 5.3 Datenbanken (S. 435-466)

- Kleuker, Stephan: Grundkurs Datenbankentwicklung,
  Springer/Vieweg, 2013, 3. Aufl.
  kostenlos über Springerlink

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Was sind Datenbanken?

Eine Datenbank ist ein System zur Speicherung von Geschäftsdaten

Kiefer et al.

Eine Datenbank ist ein System zur Beschreibung, Speicherung und Wiedergewinnung von umfangreichen Datenmengen ...

Duden Informatik

Eine Datenbank ist eine möglichst redundanzfreie Sammlung von Daten, die so strukturiert sind, dass sie von mehreren Benutzern oder Anwendungen gleichzeitig genutzt und flexibel ausgewertet oder verknüpft werden können.

Laudon et al.

Daten im Unternehmen:

Was sind Datenbanken?

Eine Datenbank…

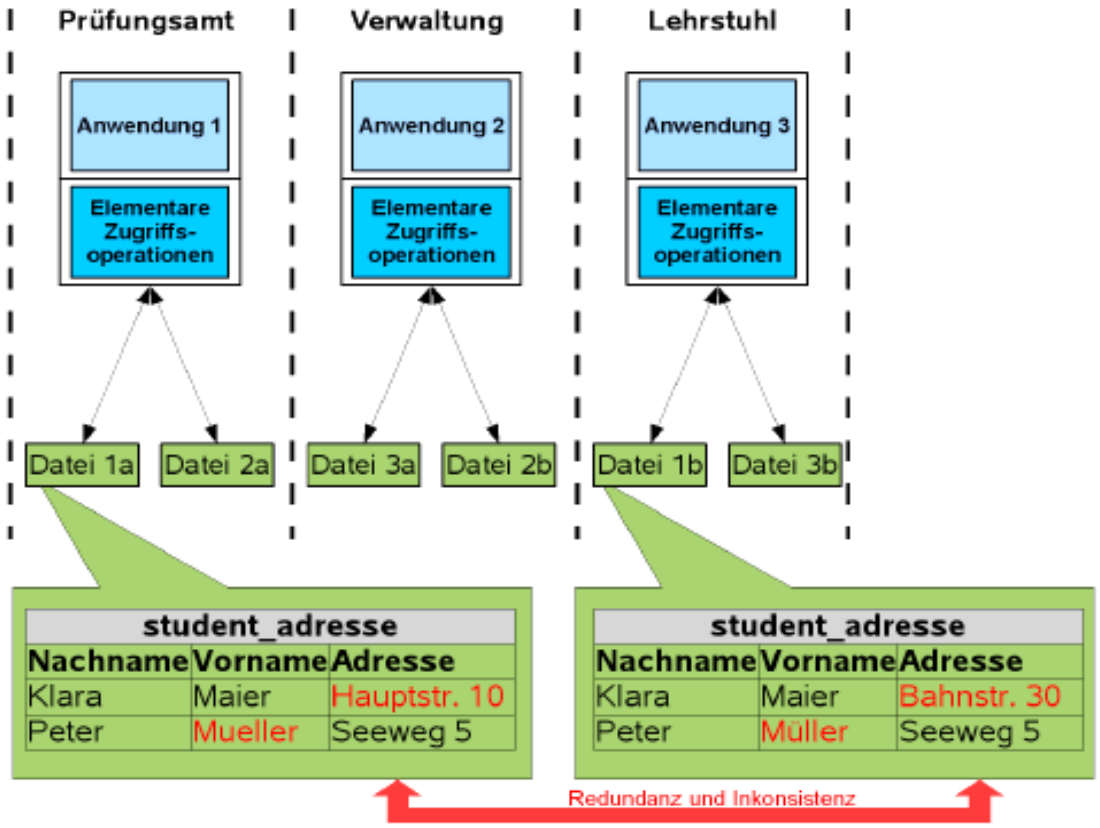
- ist eine logisch zusammengehörende Menge von Daten, denen wir eine Bedeutung beimessen.
- repräsentiert den Zustand eines Teil der realen Welt. Den dargestellten Weltausschnitt bezeichnen wir auch als Anwendungswelt.
- bildet die durch Ereignisse in der Anwendungswelt ausgelösten Zustandsänderungen ab.
- dient als Datenquelle zur Entscheidungsunterstützung.

Herkömmliche Speicherung von Daten

- **Datenunabhängigkeit**: komfortable Anwendung nur für die spezielle Anwendung möglich.
- **Redundanz**: Daten kommen mehrfach vor, müssen an mehreren Stellen geändert werden →Speicherplatzverschwendung
- **Inkonsistenzen**: gleichzeitige Änderung von Daten durch Benutzer oder Programme
- **Datenschutz**: Lesezugriff zum Teil uneingeschränkt möglich

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

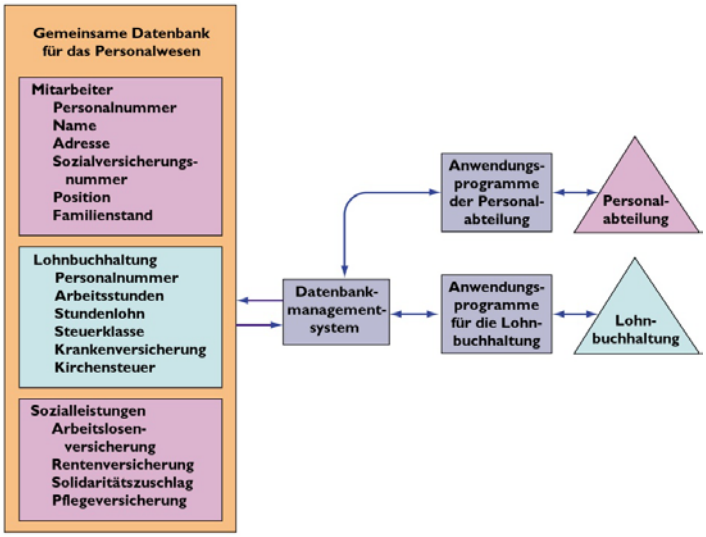
Herkömmliche Speicherung von Daten



Quelle: Zicari, R.: http://www-stud.informatik.uni-frankfurt.de/~prg2/SS2009/folien/zicari/zicari-db_teil1.pdf

Warum Speicherung in Datenbanken?

- Daten werden immer mehr
- Daten wurden als zentrales Betriebsmittel (Ressource) erkannt
- Daten zentral halten und verwalten
- Daten den Mitarbeitern zur Erfüllung des betrieblichen Zwecks zur Verfügung stellen
  → jeder bekommt nur einen Ausschnitt der Daten zu sehen (d.h. einzelne Datenfelder)!



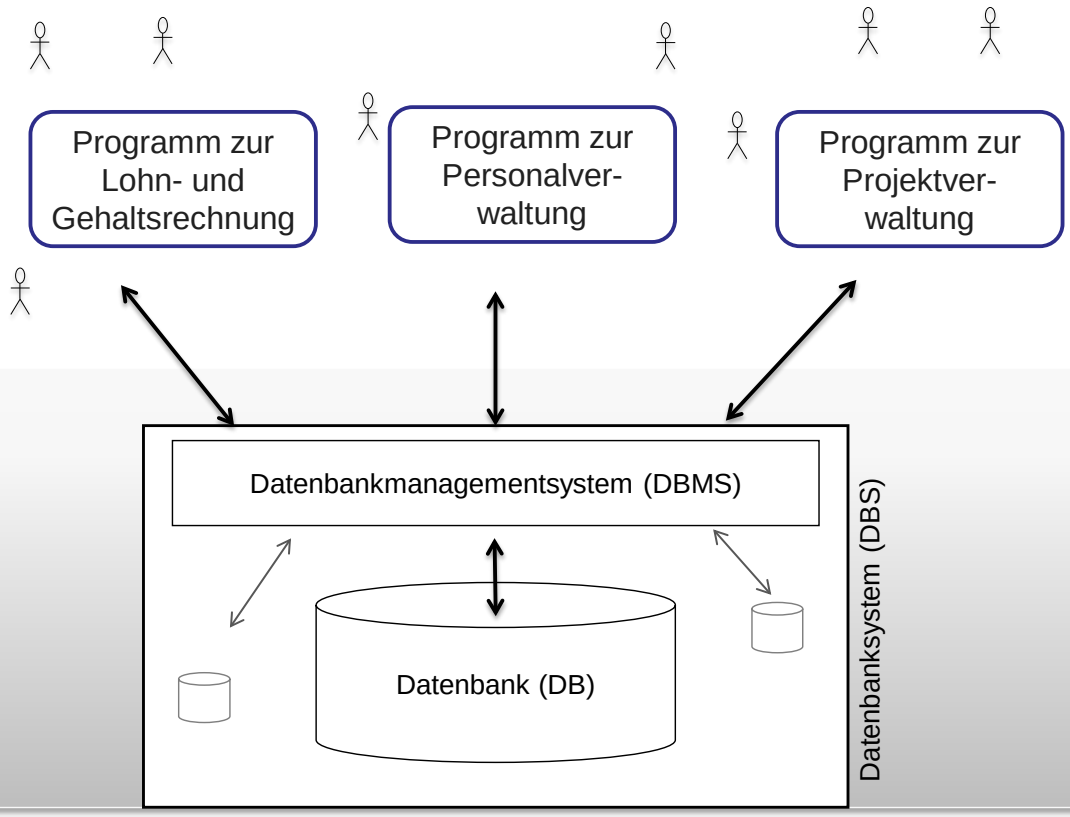
Aufbau eines Datenbanksystems

- **Datenbankmanagementsystem**: Software, die die Anfragen der verschiedenen Anwendungen entgegennimmt

- **Datenbanksystem**: Kombination aus der/den Datenbank/en und dem Datenbankmanagementsystem

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

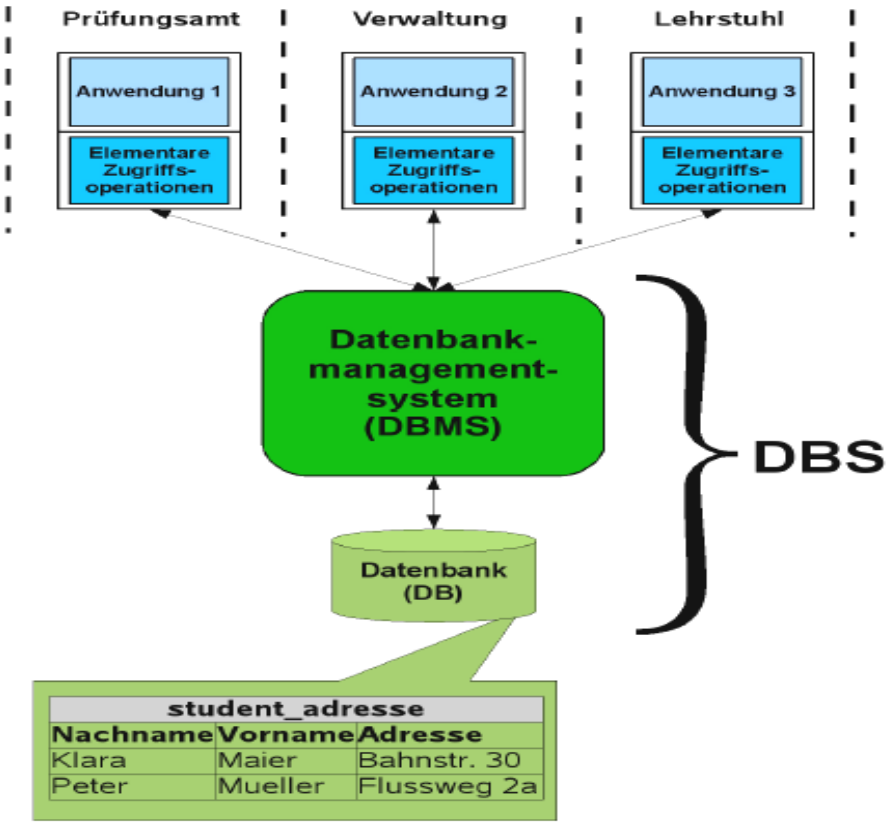
Aufbau eines Datenbanksystems



Eigenschaften von Datenbanksystemen

- **Zugriffskontrolle**: Nur berechtigte Nutzer haben Einblick auf die Daten
- **Synchronisation**: mehrere Anwender können gleichzeitig am gleichen Datenbestand arbeiten
- **Datenschutz**: bestimmte Daten dürfen nur für bestimmte Nutzer zugänglich sein, z.B. Gehälter
- **Integrität**: DBS überwachen die Betriebssicherheit des Systems und der Daten; sie schützen vor D.-verlust und sorgen für widerspruchsfreie Daten
- **Sprache**: DBS stellen eine Abfragesprache zur Verfügung
- **Effizienz**: effiziente Verwaltung großer Datenmengen

Speicherung in Datenbanksystemen



Quelle: Zicari, R.: http://www-stud.informatik.uni-frankfurt.de/~prg2/SS2009/folien/zicari/zicari-db_teil1.pdf

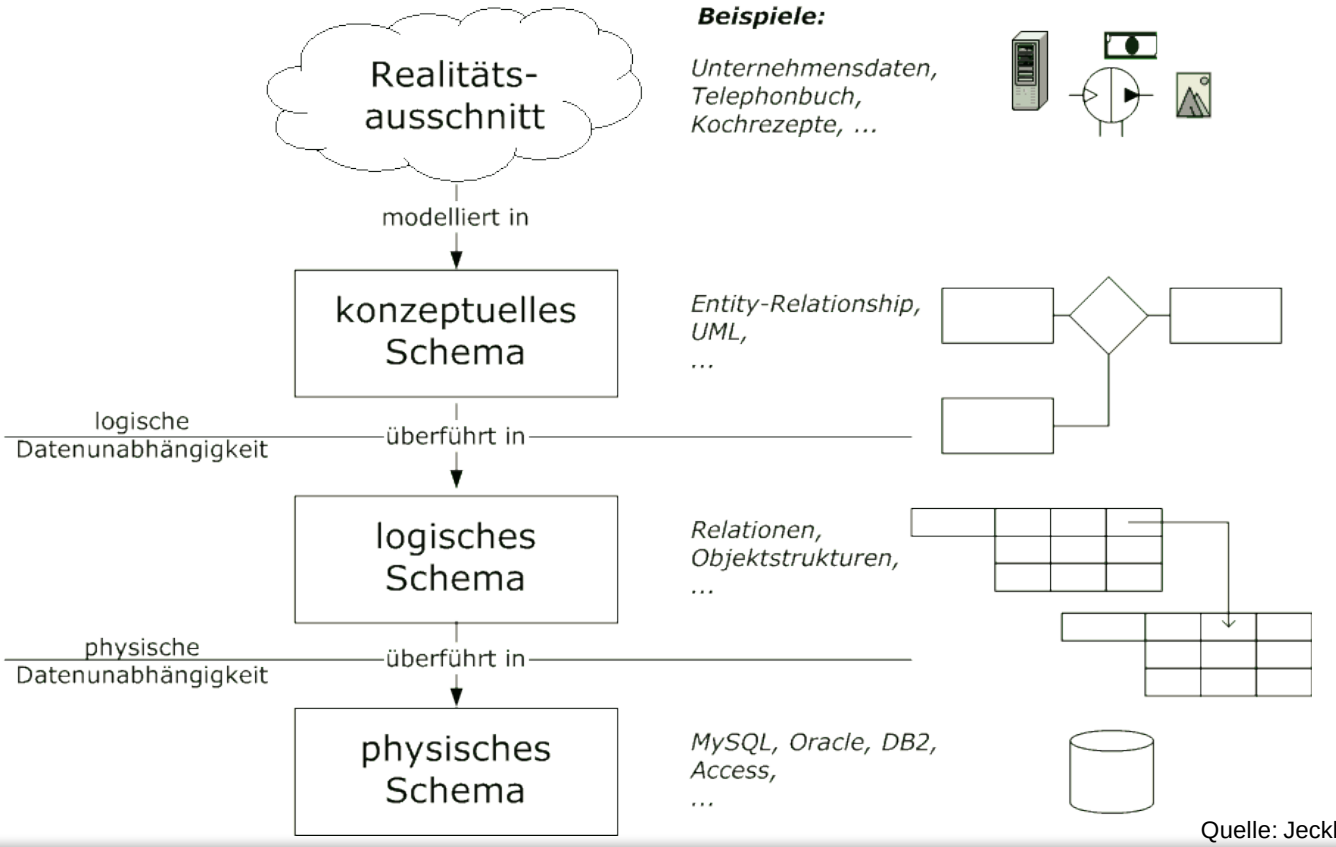
Arten von Datenbanksystemen

- Zentralisierte DBS:
  DB, Datenbanksoftware und Anwendungen laufen auf einem zentralisierten Rechner
- Verteilte DBS
  DB‘s laufen auf versch. Rechnern mit jeweils einem DBMS, welches über Mechanismen zur Zusammenführung der verteilten Datenbank verfügt

- Client-Server DBS
- parallele DBS:
  Läuft auf Parallelrechnern mit/oder Multiprozessorsystemen und verkürzt Datenbankanfragen und Transaktionen

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

3-Ebenen Schemaarchitektur



Quelle: Jeckle.de

Anforderungen an den Datenbankentwurf

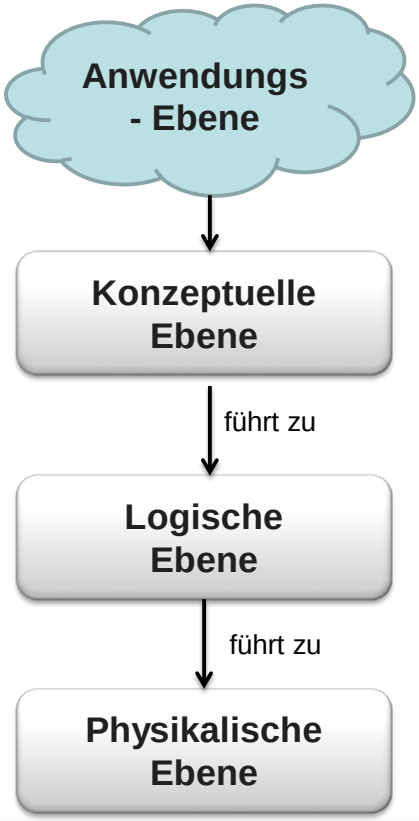
- Das Schema in einer Ebene kann verändert werden, ohne dass Änderungen auf der nächst höheren Ebene erforderlich sind.
- Änderungen in den Anforderungen an die Datenbanken haben keine oder nur geringe Auswirkungen auf bestehende Datenbanknutzer und -anwendungen.
- Die Ebenen in der 3-Ebene-Schema-Architektur können weitestgehend unabhängig voneinander beschrieben und geändert werden.

Zusammenfassende Fragen: Abschnitt 1

- Was ist eine DB, ein DBS, ein DBMS?
- Erläutern Sie die 3-Ebenen Schemaarchitektur.
- Was sind typische Eigenschaften von DBS?
- Was ist ein verteiltes DBS?

Teil 1: Datenbanken

Datenmodellierung



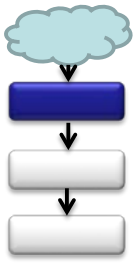
Umgangssprachliche Beschreibung der Systemanforderungen, z.B. mittels Pflichtenheft

Realitätsnahe, graphische Beschreibung der Anwendungswelt / der Informationsstruktur, z.B. mittels **ER-Modell**

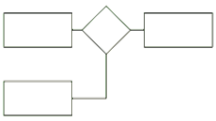
Dient zur Implementierung mittels DBS, z.B. **Relationale Datenmodell**

Abbildung im System durch Datenbanksprache

Grundidee



- Das ER-Modell begreift die Welt als **Objekte**
- Objekte existieren nicht alleine, sondern stehen in **Beziehungen** zu andern Objekten
- Objekte haben charakteristische **Eigenschaften**



Warum ist das ER-Modell ein konzeptionelles Modell?

- Weil es keine Anfragesprache oder Änderungsoperatoren für das Modell gibt → es muss in ein weiteres Modell übersetzt werden

Elemente

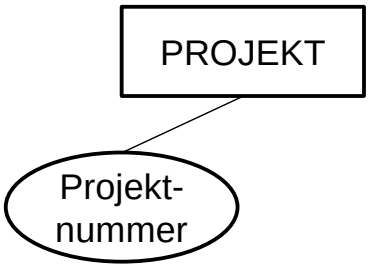
- **Entities**
  (=Entitäten=Objekte, Gegenstände), über die Informationen gespeichert werden soll
  - z.B. Mitarbeiter, Projekt
- **Entity-Set** (Entitätsmenge)
  Sammlung von gleichartigen Entitäten mit gleichen Eigenschaften aber untersch. Eigenschaftswerten
  - Alle Mitarbeiter, Alle Projekte
  - Darstellung durch ein Rechteck
  - Kommt nur einmal vor
  - Bezeichnung durch Substantiv, Großbuchstaben

Elemente

- **Attribute** (Eigenschaften)

  charakterisieren eine Entität oder eine Beziehung. Sie haben einen Namen und einen Wert (value).

  - Alle Entitäten dieses Typs haben diese Eigenschaft.
  - Jede Entität hat mindestens eine Eigenschaft
  - kann mehrmals im Diagramm vorkommen
  - Darstellung durch eine Ellipse,
    Verbunden durch eine Kante mit der Entität



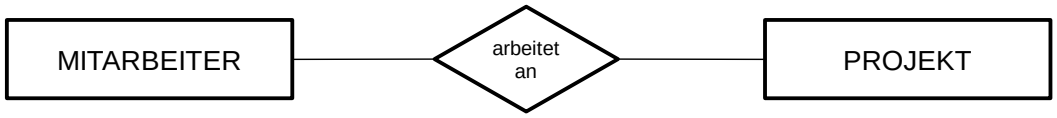
- **Domäne**

  beschreibt den zulässigen Wertebereich einer Eigenschaft. Diese können vorgegeben sein (Januar, Februar, …) oder einen Bereich darstellen (0-999)

Elemente

- **Relationships** (Beziehungen)
  beschreiben die Wechselwirkungen bzw. Abhängigkeiten von mind. 2 Entitäten

  - z.B.: „arbeitet an“ → Mitarbeiter *arbeitet an* Projekt
  - Darstellung durch eine Raute und 2 Kanten zu den entsprechenden Entitäten
  - Die Raute wird durch ein Verb so beschriftet, dass die Art der Beziehung deutlich wird. Dies kann mehrmals im Diagramm vorkommen
  - Die Lesrichtung (bspw. durch Aktiv- oder Passivbenennung ist unerheblich)
  - Die Beziehungen können ihrerseits durch Attribute näher beschrieben werden



Zusammenfassung: Grafische Darstellung der Elemente

Während der Entwicklung wird ein Entwurf häufig überarbeitet und geändert. Eine Planung, so dass sich Kanten nicht überschneiden, ist selbst bei sehr kleinen Diagrammen kaum möglich.

Schlüssel

Problem:

- Wenn Objekte in Entitäten gespeichert werden, dann müssen diese Entitäten wieder eindeutig einzelnen Objekten zugeordnet werden können

- In der Modellierung muss also sichergestellt werden, dass zwei verschiedene Objekte durch zwei unterscheidbare Entitäten dargestellt werden
- Da Entitäten durch ihre verschiedenen Attribute näher beschrieben werden, muss dort eine Unterscheidung stattfinden

Schlüssel

Lösung:

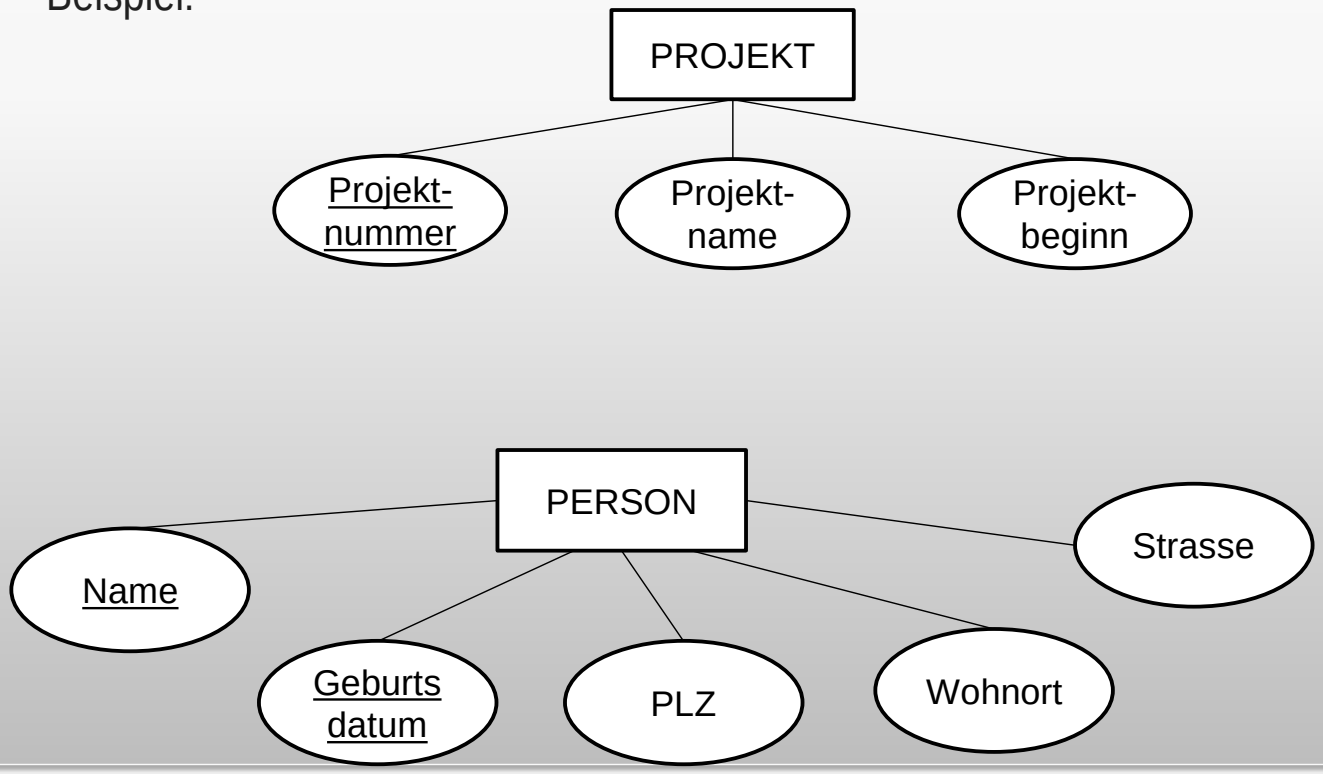
- Ein oder mehrere Attribute, die eine eindeutige Identifikation eines Objektes ermöglichen, werden als **Schlüssel** bezeichnet

  - So kurz/wenig wie möglich, so lange wie nötig
  - Existiert unter den vorhanden Attributen oder Attributkombinationen keine, die als Schlüssel eingesetzt werden kann, wird ein künstliches erzeugt (z.B. „ID“)
  - Darstellung durch <u>Unterstreichen</u>

- Der Wert eines **Primärschlüssels** zeichnet sich dadurch aus, dass er nur ein einziges Mal vorkommt

Schlüssel

- Beispiel:



Kardinalitäten

- Entitäten (Objekte) der Anwendungswelt stehen meist in Verbindung zu einander (Relationship)

  z.B.: Mitarbeiter Schmidt arbeitet an Projekt 4711

- Beziehungen unterliegen häufig Beschränkungen

  - Jedes Projekt steht in Verbindung mit beliebig vielen Mitarbeitern, die an diesem Projekt arbeiten
  - Jeder Mitarbeiter steht mit mehreren Projekten in Verbindung, an denen er arbeitet

Kardinalitäten

- Die **Kardinalität** legt fest, wie viele Entitäten einer Entitätsmenge mit Entitäten einer anderen Entitätsmenge in Verbindung stehen können.

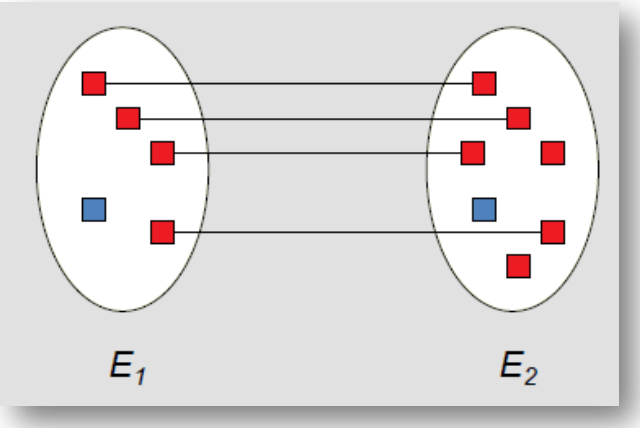
- Kennzeichnung durch:

  - 1 - genau eine Zuordnung
  - n, m - eine oder mehrere Zuordnungen

Kardinalitäten

1 zu 1 Beziehung

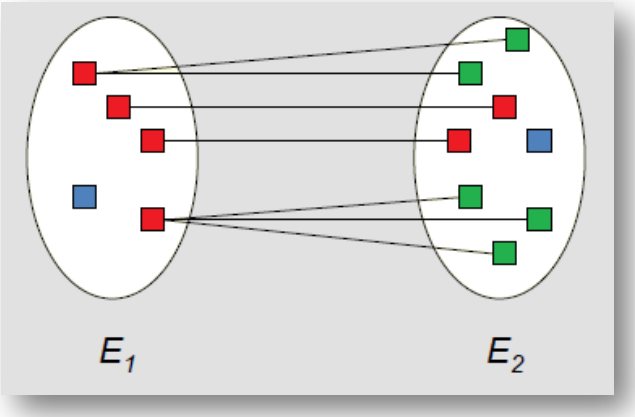
- Jedes Objekt der Entitätsmenge E1 kann mit höchstens einem Objekt von E2 in Verbindung stehen und umgekehrt.



Kardinalitäten

1 zu n Beziehung

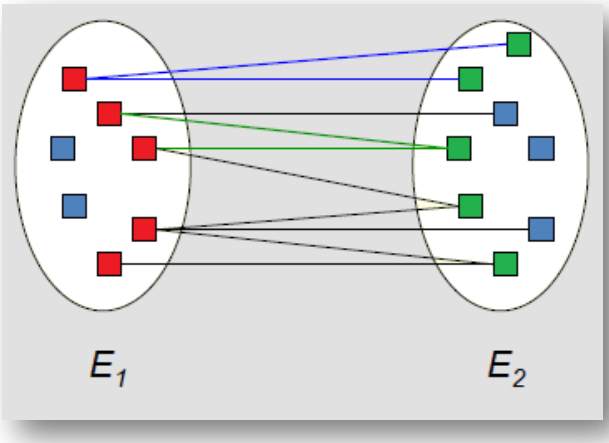
- Jedes Objekt aus E1 kann beliebig viele Verbindungen zu Objekten aus E2 besitzen
- Jedes Objekt aus E2 steht in Verbindung mit max. einem Objekt aus E1



Kardinalitäten

m zu n Beziehung

- Jedes Objekt aus E1 kann beliebig viele Verbindungen zu Objekten aus E2 besitzen und umgekehrt



(n zu n würde bedeuten, dass E1 und E2 die gleiche Anzahl an Objekten besäße)

Kardinalitäten

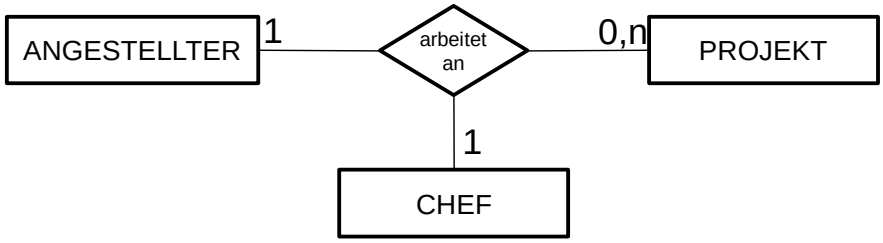
- Wenn die Möglichkeit besteht, dass eine Beziehungen keine Zuordnung hat, kann eine 0 hinzugefügt werden

→ Jeder Mitarbeiter arbeitet an 0,1,2,…. n Projekten

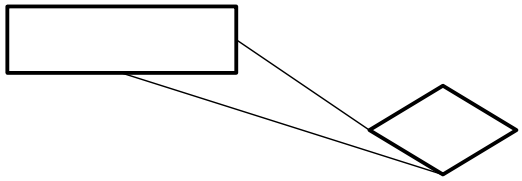
Grad der Beziehung

Anzahl der Entitätstypen, die an einem Beziehungstypen beteiligt sind

- **binär**, wenn zwei Entitäten verbunden sind
- **ternär**, wenn drei Entitäten miteinander verbunden sind



- rekursiv binär, wenn eine Entität mit sich selbst in Verbindung steht



Alternative UML Multiplizität

- Bisher lies sich nicht ableiten, wie viele Entitäten tatsächlich möglich sind. Es gab keine Begrenzung.

- Lösung: Darstellung von Ober- und Untergrenzen

  - 0..1 kein Objekt oder höchstens 1
  - 0..* keine bis beliebig viele
  - 1..1 mindestens 1 und höchstens 1
  - 1..* mindestens 1 bis beliebig viele

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Fallstudie HS-OWL

- Die Hochschule möchte ein Informationssystem zur Verwaltung der anfallenden Daten einführen. Sie erstellen dazu einen Teil des **konzeptionellen** Datenbankentwurfs
  - Zu jedem Fachbereich werden die Fachbereichsnummer, die Bezeichnung und der Ort gespeichert. Die Fachbereichsnummer ist eindeutig.
  - Zu jedem Prof. werden der Name, der Titel und das Lehrgebiet gespeichert.
  - Jeder Prof. arbeitet in einem Fachbereich, jeder Fachbereich hat mehrere Profs.
  - Jeder Prof. unterrichtet mehrere Studenten, diese haben Unterricht bei mehreren Profs.
  - Studenten benötigen eine Matrikelnummer, einen Namen und einen Geburtsdatum
  - Jeder Student muss mehrere Prüfungen schreiben, die der Prof korrigiert. Zur Prüfung wird das Fach und die Art gespeichert.

Programme:

Ein schönes online Programm ist übrigens **https://www.draw.io/**
oder alternativ http://www.gliffy.com/ (mit kostenloser Anmeldung).
Offline eignet sich bspw. **Microsoft Visio**

Fallstudie Bibliothek

- Die Bibliothek möchte ihre Daten in einem Informationssystem speichern Sie erstellen dazu einen Teil des **konzeptionellen** Datenbankentwurfs

  - Studierende identifizieren sich mit ihrer Matrikelnummer und ihrem Namen
  - Die Bücher sind in mehreren Exemplaren vorrätig, maximal jedoch 10
  - Studierende können sich diese Exemplare ausleihen. Jedes Exemplar hat eine Inventarnummer.
  - Bei der Ausleihe wird gespeichert, am welchem Datum die Ausleihe stattfand und wie lange sie Gültigkeit hat
  - Die Bücher ansich können durch die ISBN, den Autor und den Titel beschrieben werden
  - Sofern möglich, wird gespeichert, welche Bücher für welche Module der Hochschule sind. Diese haben eine Modulnummer und einen Namen

Fallstudie Rezept

- Mit Hilfe eines neuen PraxisInformationssystems sollen Rezepte, verwaltet werden. Dabei soll das System folgende Informationen speichern:
  - **Patienten**: Zu jedem Patienten werden der Name und die Anschrift gespeichert. Außerdem hat jeder Patient eine eindeutige Patientennummer.
  - **Rezept**: Zu jedem Rezept sind das Ausstellungsdatum und die Art (Privat/Kasse) zu speichern. Außerdem erhält jedes Rezept eine eindeutige Rezeptnummer.
  - Für jeden Patienten können mehrere Rezepte ausgestellt sein. Jedes Rezept ist natürlich für genau einen Patienten ausgestellt.
  - Jedes Rezept bezieht sich auf ein oder mehrere Medikamente.
  - **Medikament**: Zu jedem Medikament sind die Bezeichnung, der Hersteller und die Packungsgröße zu speichern. Außerdem soll die Medikamenten-nummer gespeichert werden.
  - Jedes Medikament kann auf mehreren Rezepten auftauchen.

Fallstudie Krankenhaus

- Die folgende Fallstudie behandelt einen kleinen Ausschnitt aus einem Krankenhausinformationssystem. Im Detail sind folgende Informationen zu verwalten:
  - Zu jeder Fachabteilung ist die Abteilungsnummer, die Bezeichnung und das Quartalsbudget zu speichern. Die Abteilungsnummern sind natürlich eindeutig.
  - Zu jedem Arzt ist der Name, Personalnummer und der Titel zu speichern.
  - Zu jedem Patienten soll der Name, die Krankheit und die Patientennummer gespeichert werden.
  - Jeder Arzt ist für nur eine Abteilung tätig. Natürlich hat eine Abteilung mehrere Ärzte.
  - Jeder Arzt behandelt mehrere Patienten. Für jeden Patienten ist ein Arzt zuständig, d.h. dieser Arzt behandelt den Patienten.

Zusammenfassende Fragen: Abschnitt 1.2

- Was sind Entitäten, Attribute etc.?
- Wie werden die Elemente des ER-Modells grafisch notiert?
- Was sind Schlüssel?
- Welche Beziehungen gibt es und was bedeuten sie?

Teil 1: Datenbanken

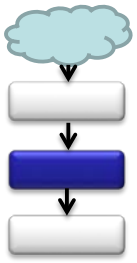
1 Übersicht 

2 ER-Modell 

3 Das relationale Datenmodell

4 SQL

Grundverständnis



- Teilbereich der **logischen** Datenmodellierung
- Standard im Bereich betriebswirtschaftlicher und kommerzieller Anwendungen
- Relationale Datenbanken speichern Daten in Tabellen
- Tabellen=Relationen

Kunde

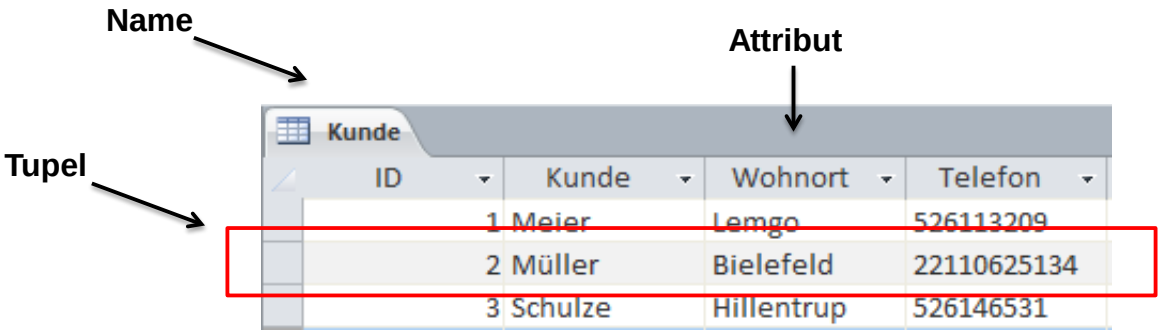
ID	Kunde	Wohnort	Telefon
1	Meier	Lemgo	526113209
2	Müller	Bielefeld	22110625134
3	Schulze	Hillentrup	526146531

Tabellenstruktur

- Jede Tabelle hat einen **Namen**
- Die **Struktur** der Relation wird durch die Anzahl der Spalten und Spaltenüberschriften ausgedrückt
- Name und Struktur bilden das **Schema** der Relation
- Die Spalten werden als **Attribute** der Relation bezeichnet

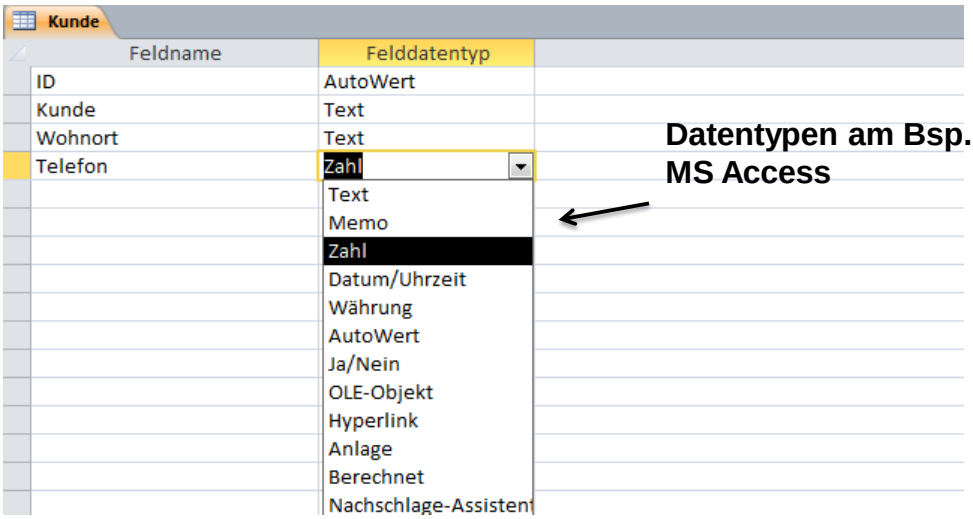
Tabellenstruktur

- Der Datensatz in einer Zeile wird als **Tupel** bezeichnet
- Jeder Datensatz kann zu jedem Attribut nur einen Wert abspeichern
- Es ist unerheblich, in welcher Reihenfolge die Datensätze und die Attribute gespeichert sind. In einer Menge gibt es keine Ordnung im Sinne einer Rangfolge.



Tabellenstruktur

- Für jedes Attribut muss ein **Datentyp** festgelegt werden. Die meisten DBS besitzen zumindest die Typen Text, Zahl, Datum
- Der Datentyp legt fest, welcher Wertebereich für das Attribut zulässig ist



Datentypen am Bsp. MS Access

Beziehungen

- Erinnerung: **Primärschlüssel** ermöglichen die eindeutige Identifikation eines Tupels. Kein Tupel darf doppelt vorkommen, da es in einer Menge keine Duplikate gibt.

- Ein **Fremdschlüssel** ist ein Attribut, welches eine Beziehung zu einem Primärschlüssel einer anderen Relation herstellt. Ein Relationenschema enthält also Schlüsselattribute eines anderen Relationenschemas.

- Fremdschlüssel können öfter vorkommen, und können aus mehreren Attributen bestehen

- Der Name des Fremdschlüssels sollte den gleichen Namen wie der referenzierte Primärschlüssel haben (muss aber nicht)

Beziehungen

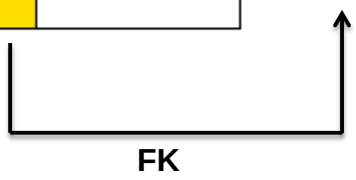
- Die Visualisierung erfolgt durch einen Pfeil (gerichtete Kante)
- Es muss zu jedem Wert der Fremdschlüsselattribute einen Datensatz in der bezogenen Relationen geben, der die gleichen Werte als Primärschlüsselattribut besitzt

Tabelle: Adresse:

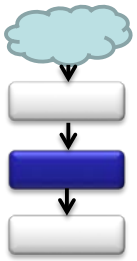
id	nachname	vorname	strasse	plz	telefon
1	Müller	Fritz	Hauptstr. 12	72070	07071-555-12312
2	Simmer	Susi	Herbstallee 1	72074	07071-555-654654
3	Sommer	Susi		72074	07071-555-64444

Tabelle: Plz:

plz	ort
72070	Tübingen
72074	Tübingen

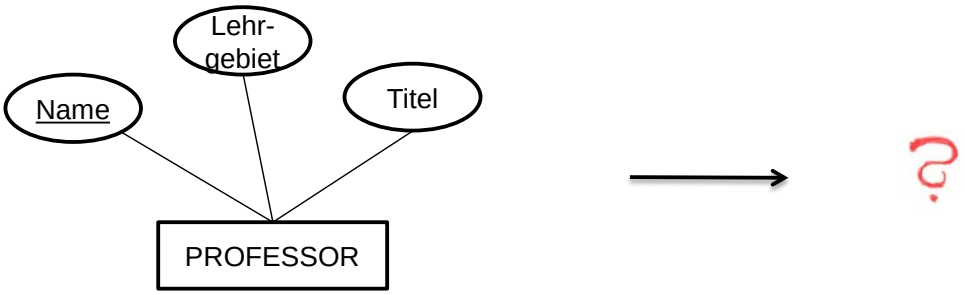


Vom ER-Modell zum Relationalen Datenmodell



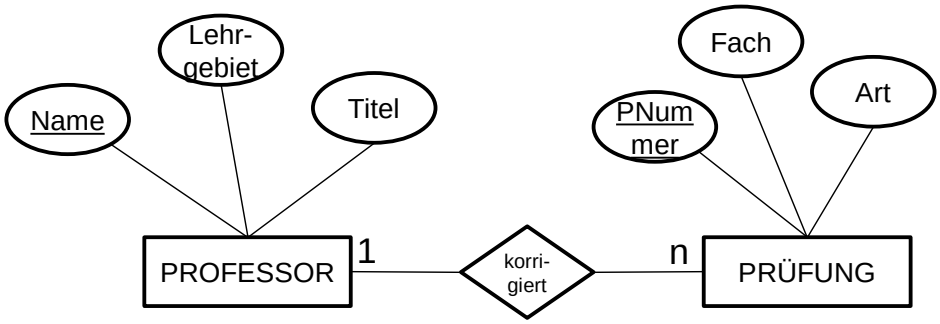
Wie kann ein graphisches ER-Modell in das Relationale Datenmodell überführt werden?

- Jeder Entity-Typ wird als Tabelle mit den entsprechenden Attributen angelegt. Schlüssel werden übertragen.



Vom ER-Modell zum Relationalen Datenmodell

1:1 / 1:n Beziehungen

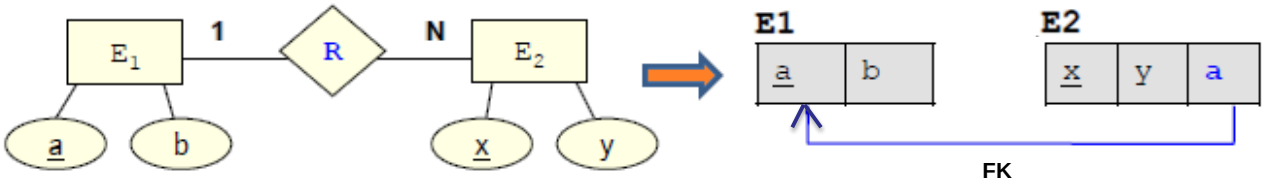


- Für die Entitäten Professor und Prüfung wird **je** eine Relation erzeugt
- Für die Abbildung der Beziehung „korrigiert“ gibt es 2 Möglichkeiten:

Vom ER-Modell zum Relationalen Datenmodell

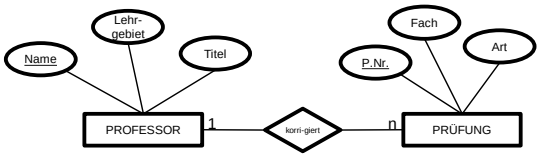
1:1 / 1:n Beziehungen

- Möglichkeit 1:
  Die Relation, wo das n steht (bei 1:1 ist es egal) wird um eine Spalte (Attribut) mit dem Primärschlüssel der in Beziehung stehenden Entität erweitert.
- Sofern die Beziehung Attribute enthält, kommen diese ebenfalls dazu.



Vom ER-Modell zum Relationalen Datenmodell

1:1 / 1:n Beziehungen



Vorteil: keine weitere Relation nötig
Nachteil: Nullwerte möglich

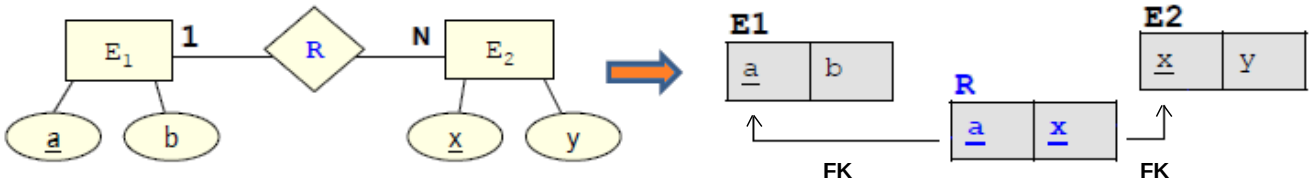
Prüfung

PNummer	Fach	Art	Name
26	Marketing 1	Schrift	Ebert
45	Marketing 1	Schrift	Ebert
23	Marketing 2	Muend	Ebert
24	Orga	Schrift	Kottmann
25	Orga	Schrift	Kottmann
28	Vertrieb	Muend	Kracht

Vom ER-Modell zum Relationalen Datenmodell

1:1 / 1:n Beziehungen

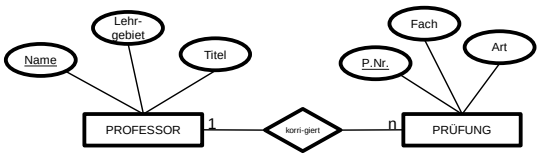
- Möglichkeit 2:
  Anlegen einer neuen Relation mit den Primärschlüsseln der in Beziehung stehenden Entitäten als Attribute
- Der Name der Relation sollte sinnvoll gewählt werden



Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Vom ER-Modell zum Relationalen Datenmodell

1:1 / 1:n Beziehungen



Vorteil: keine Nullwerte möglich
Nachteil: eine weitere Relation nötig

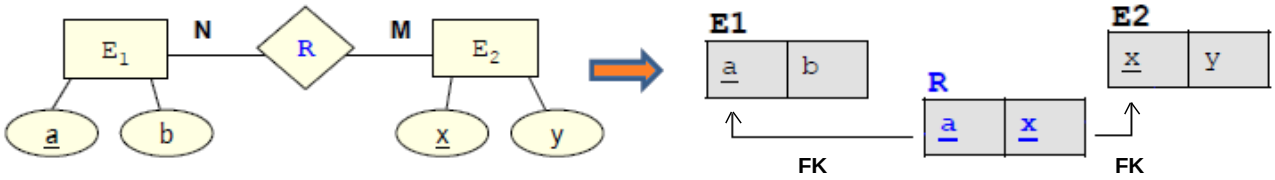
korrigiert

Name	PNummer
Ebert	26
Ebert	45
Ebert	23
Kottmann	24
Kottmann	25
Kracht	28

Vom ER-Modell zum Relationalen Datenmodell

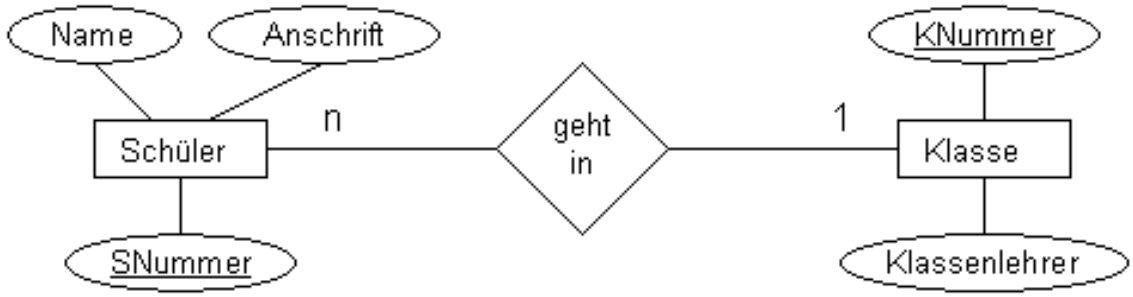
n:m Beziehungen

- Grundsätzlich: Anlegen einer neuen Relation mit den Primärschlüsseln der in Beziehung stehenden Entitäten als Attribute.
- Die neue Relation enthält ebenfalls die Attribute der Beziehung, sofern vorhanden
- Untergrenzen (UML-Notation) können leider nicht abgebildet werden.
- Vorgehensweise ist übertragbar auf ternäre Beziehungen



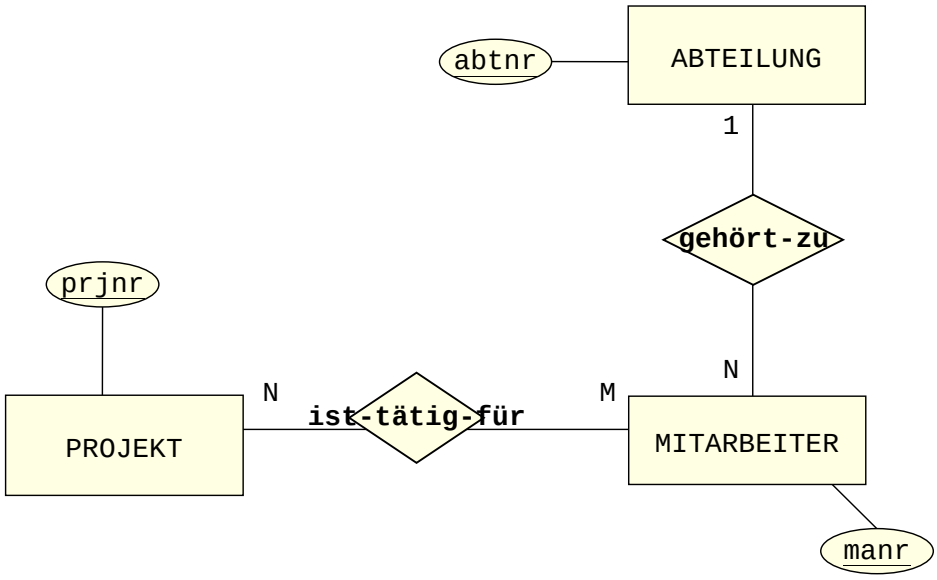
Fallbeispiel

- 1. Erstellen Sie einen logischen Datenbankentwurf für das folgende ER-Diagramm. (Beide Möglichkeiten ☺)



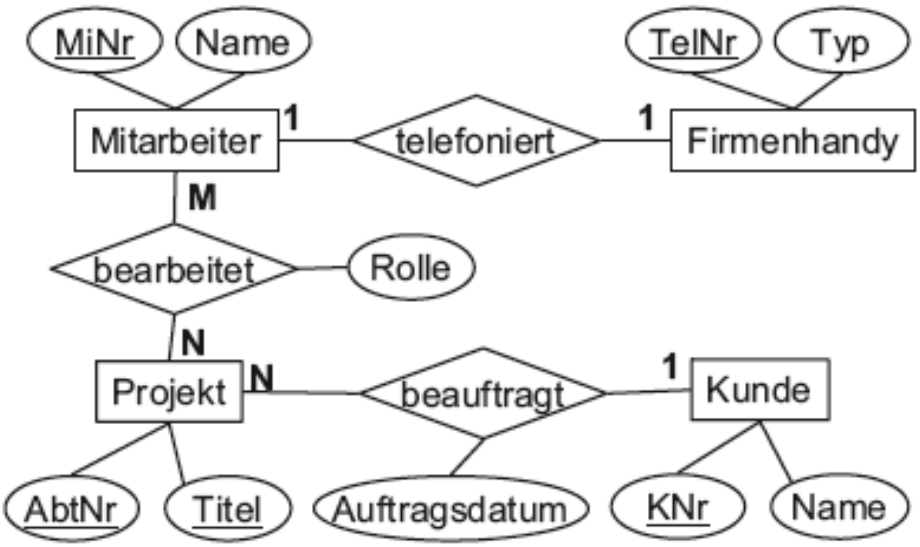
Fallbeispiel

- 2. Erstellen Sie einen logischen Datenbankentwurf für das folgende ER-Diagramm



Fallbeispiel

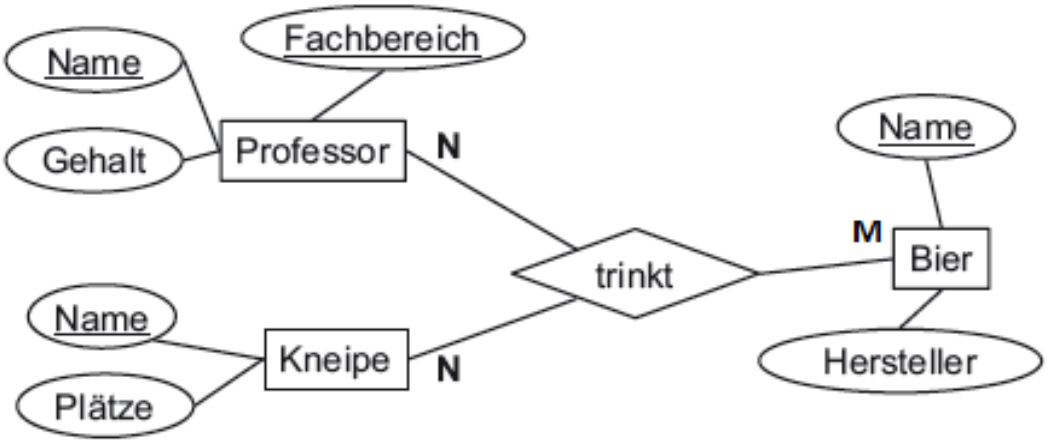
- 3. Erstellen Sie einen logischen Datenbankentwurf für das folgende ER-Diagramm



Aus: Kleuker: Grundkurs Datenbankentwicklung 2. Auflage, 2006

Fallbeispiel

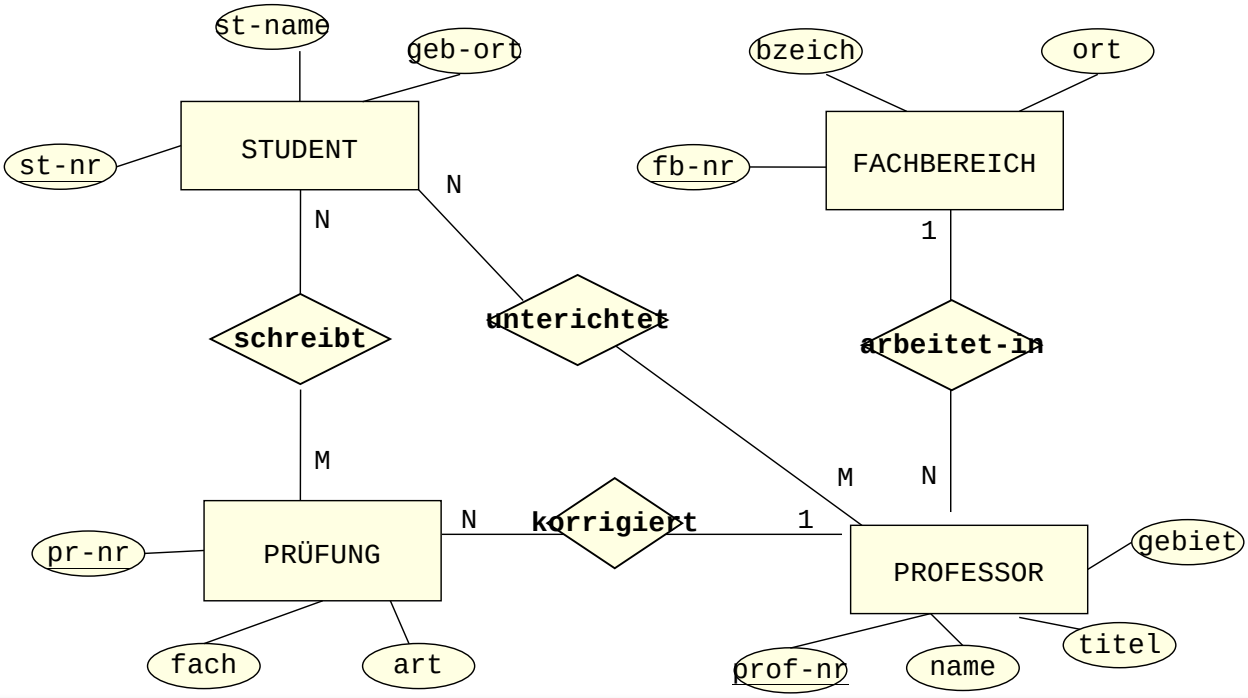
- 4. Erstellen Sie einen logischen Datenbankentwurf für das folgende ER-Diagramm



Aus: Kleuker: Grundkurs Datenbankentwicklung 2. Auflage, 2006

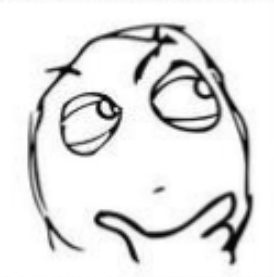
Fallbeispiel

- 5. Erstellen Sie einen logischen Datenbankentwurf zum Fallbeispiel HS-OWL



Zusammenfassende Fragen: Abschnitt 1.3

- Was ist ein Tupel? Was ein Datenbankschema?
- Was sind Fremdschlüssel? Bsp.?
- Was sind Datentypen und welche haben DBS meist?
- Wie werden unterschiedliche Beziehungen abgebildet?
- Wie sehen die Relationen der Fallstudien aus dem 2. Abschnitt aus?



Teil 1: Datenbanken

1 Übersicht ✓

2 ER-Modell ✓

3 Das relationale Datenmodell ✓

4 SQL

Super online Literatur:
http://de.wikibooks.org/wiki/Einf%C3%BChrung_in_SQL:_Inhaltsverzeichnis#Einf.C3.BChrung

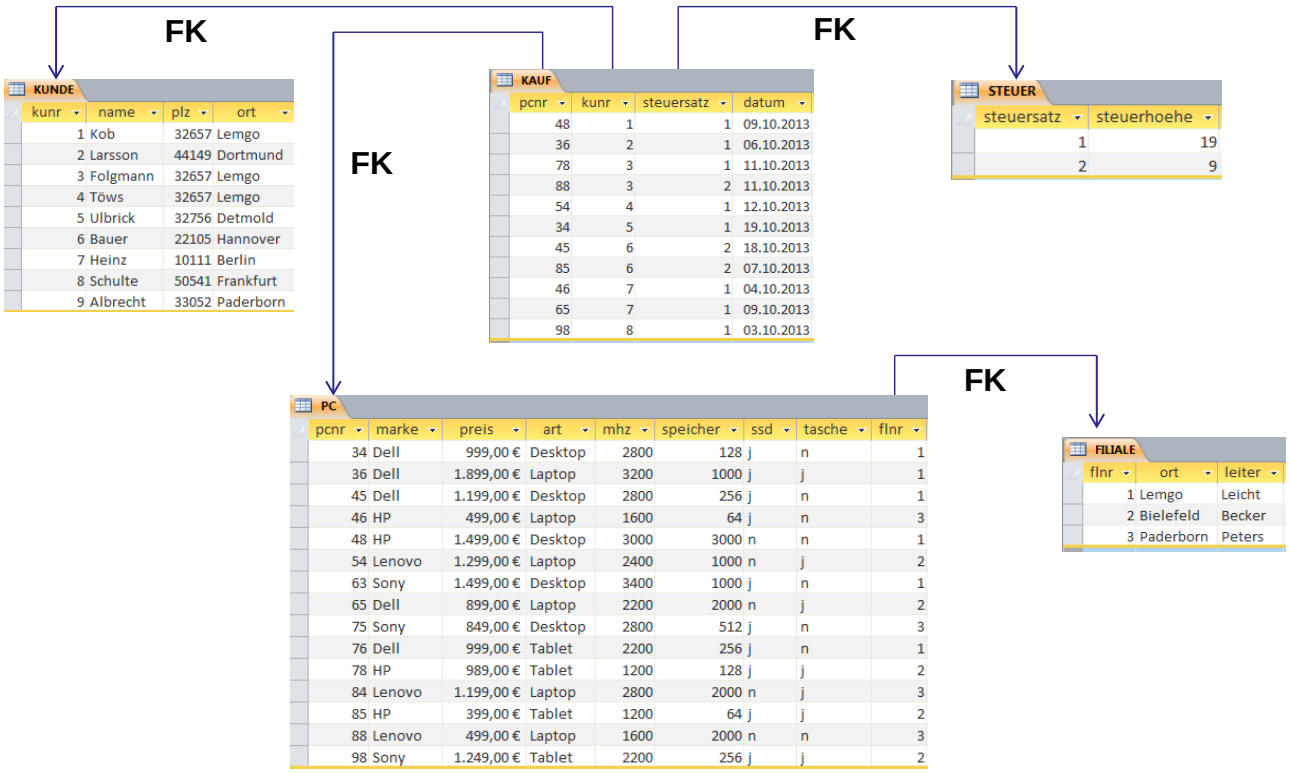
Ausgangslage

Es soll ein Datenbankentwurf für folgende Ausgangslage erstellt werden:

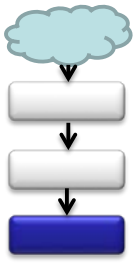
- Ein Computerladen hat mehrere Filialen. Zu den Filialen wird der Ort und der Leiter gespeichert.
- Kunden können dort verschiedene Rechner kaufen. Zu den Rechnern werden die Marke, der Preis, die Art, die Megahertz, der Speicherplatz sowie das Vorhandensein einer SSD oder Tasche gespeichert.
- Zu den Kunden wird der Name, die Postleitzahl und der Ort gespeichert.
- Es gibt verschiedene Steuersätze, die gespeichert werden können.
- Ebenfalls wird vermerkt, wann ein Einkauf stattfand (Datum).

Erstellen Sie einen konzeptionellen Datenbankentwurf!

Ausgangslage



Grundlagen



- SQL (meist: Structured Query Language) ist eine relativ einfache und intuitive Anfragesprache zur Auswertung von Datenbeständen
- Quasi-Standard für relationale Datenbanken
- Auswertung= Abfrage und Manipulation
- SQL kann:

1.	**Erstellen von DB, Tabellen (Relationen) und Indizes**
2.	**Abfragen von Daten**
3.	**Anlegen, Ändern und Löschen von Datensätzen**
4	**Anlegen von Benutzern und Vergabe von Zugriffsrechten**

Einfache Anfrage

Jede Anfrage bietet als Ergebnis eine Tabelle

```
SELECT    art, mhz
FROM      PC
WHERE     marke ="Dell";
```


art	mhz
Desktop	2800
Desktop	2800
Laptop	2200
Tablet	2200
Laptop	3200

Einfache Anfrage

Die Anfrage besteht aus 3 Teilen:

```
SELECT    art, mhz
FROM      PC
WHERE     marke ="Dell";
```

- SELECT

  Nach dem Schlüsselwort SELECT folgt eine Liste an Attributen (=Spalten). * wählt alles aus.

- FROM

  Nach FROM folgt die angesprochene Tabelle. Alle Attribute aus der SELECT Angabe müssen enthalten sein.

- WHERE

  Die optionale WHERE Angabe stellt die Bedingung, nach der gesucht werden soll. Ohne WHERE sind alle Datensätze enthalten.

Einfache Anfrage – Zusatz zu SELECT

SELECT

aufgezählte Attribute entsprechen den Namen der Tabellenspalten. Durch das Schlüsselwort AS kann dieser geändert werden

```
SELECT   art AS Bauform, mhz AS Megahertz
FROM     PC
WHERE    marke ="Dell";
```

→

Bauform	Megahertz
Desktop	2800
Desktop	2800
Laptop	2200
Tablet	2200
Laptop	3200

Die Anfrage enthält alle Datensätze, auf die die Bedingung zutrifft. Damit sind Duplikate möglich.Um Duplikate auszuschließen, verwendet man die Kombination SELECT DISTINCT

```
SELECT   DISTINCT art AS Bauform, mhz AS Megahertz
FROM     PC
WHERE    marke ="Dell";
```

→

Bauform	Megahertz
Desktop	2800
Laptop	2200
Laptop	3200
Tablet	2200

Grau hinterlegte Tabellenzellen sind durch Screenshots entstanden und haben keine weitere Bedeutung

Einfache Anfrage – Zusatz zu SELECT

SELECT

zusätzlich können **Berechnungen** ausgeführt werden

```
SELECT    art AS Bauform, mhz/1000 AS Ghz
FROM      PC
WHERE     marke ="Dell";
```

→

Bauform	Ghz
Desktop	2,8
Desktop	2,8
Laptop	2,2
Tablet	2,2
Laptop	3,2

```
SELECT    3.5 * 4 AS Ergebnis;
```

→

Ergebnis
14

(An dieser Stelle ist kein FROM Teil notwendig)

+	Addition
-	Subtraktion, Negation
*	Multiplikation
/	Division

Einfache Anfrage – Zusatz zu SELECT

SELECT

zusätzlich können **Funktionen** ausgeführt werden
Dabei wird unterschieden in Skalar- und Spaltenfunktionen:

- **Skalarfunktionen**: verarbeiten Werte oder Ausdrücke aus einzelnen Zahlen, Zeichenketten oder Datums- und Zeitwerten

Beispiel: CEILING und FLOOR liefern die nächst größere (kleinere) ganze Zahl.
TRUNCATE schneidet die Dezimalstellen ab.

```
SELECT    CEILING(7.1423)      ergibt: 8
SELECT    TRUNCATE(7.1423)     ergibt: 7
```

Beispiel: ROUND(Zahl, Anzahl_Stellen)

```
SELECT    ROUND(7.1423,1) AS ergebnis
```

→

ergebnis
7,1

CEILING, FLOOR, TRUNCATE kennt MS ACCESS nicht

Einfache Anfrage – Zusatz zu SELECT

- **Spaltenfunktionen:** verarbeiten alle Werte aus einer Spalte.

Beispiel: COUNT() zählt die Anzahl der Ergebnisse.

```
SELECT  COUNT(ort) AS AnzahlOrte
FROM    Filiale;
```

→

AnzahlOrte
3

Beispiel: SUM berechnet die Summe eine Spalte.

```
SELECT  SUM(preis) AS Wert_der_Rechner
FROM    PC;
```

→

Wert_der_Rechner
15.975,00 €

Beispiel: MIN, MAX gibt den höchsten bzw. niedrigsten Wert.

```
SELECT  MAX(mhz) AS Schnellste_Kiste
FROM    PC;
```

→

Schnellste_Kiste
3400

Beispiel: AVG ermittelt den Durchschnittswert.

```
SELECT  AVG(preis) AS Durchschnittspreis
FROM    PC;
```

→

Durchschnittspreis
1.065,00 €

Einfache Anfrage – Zusatz zu WHERE

WHERE
WHERE definiert die Bedingung der Abfrage
Als Vergleichsoperatoren stehen =, <>, <, >, <= und >= zur Verfügung.

- Für Werte sind folgende syntaktische Konventionen zu beachten:
  - Nummerische Werte bestehen nur aus Ziffern. Als Trennzeichen von Vor- und Nachkommastellen dient der Dezimalpunkt. (z.B.: 0.65)
  - Zeichenketten müssen in Anführungszeichen (einfache oder doppelte) eingeschlossen werden

- Beispiel: einfacher Vergleich

```
SELECT   art , mhz
FROM     PC
WHERE    mhz >= 3200
```


art	mhz
Desktop	3400
Laptop	3200

Einfache Anfrage – Zusatz zu WHERE

Manchmal werden mehrere Bedingungen benötigt. Diese können durch logische Operatoren (NOT, AND, OR) miteinander verknüpft werden

- NOT Bedingung

  Nur Datensätze, die die Bedingung nicht erfüllen (siehe Vergleichsoperator <>)

  Beispiel:
  Zeige alle Marken und Rechnerarten an, mit Ausnahme der von Dell

```
SELECT DISTINCT marke, art
FROM            PC
WHERE NOT       marke="Dell"
```

→

marke	art
HP	Desktop
HP	Laptop
HP	Tablet
Lenovo	Laptop
Sony	Desktop
Sony	Tablet

Einfache Anfrage – Zusatz zu WHERE

- Bedingung AND Bedingung:
  Nur Datensätze, die beide Bedingungen erfüllen gehören zum Ergebnis

  Beispiel:
  Zeige alle Rechnerarten und Mhz an, sofern die Marke Dell ist und das Gerät mehr oder gleich 2800 Mhz hat

```
SELECT  DISTINCT marke, art, mhz
FROM            PC
WHERE           marke="Dell"
AND             mhz >=2800
```

→

marke	art	mhz
Dell	Desktop	2800
Dell	Laptop	3200

Einfache Anfrage – Zusatz zu WHERE

- Bedingung OR Bedingung:

  Alle Datensätze, die mindestens eine Bedingung erfüllen, gehören zum Ergebnis

  Beispiel:

  Zeige alle Rechnermarken, Arten und Mhz an, sofern sie von Dell kommen oder mehr oder gleich 2800 Mhz haben.

```
SELECT  DISTINCT marke, art, mhz
FROM            PC
WHERE           marke="Dell"
OR              mhz >=2800
```


marke	art	mhz
Dell	Desktop	2800
Dell	Laptop	2200
Dell	Laptop	3200
Dell	Tablet	2200
HP	Desktop	3000
Lenovo	Laptop	2800
Sony	Desktop	2800
Sony	Desktop	3400

Einfache Anfrage – Zusatz zu WHERE

- Kombination von OR und AND:

  Beispiel:

  Zeige alle Rechnermarken, Arten und Mhz an, sofern sie von Dell kommen oder mehr oder gleich 2800 Mhz haben und ein Laptop sind

```
SELECT  DISTINCT marke, art, mhz
FROM    PC
WHERE   marke="Dell" OR mhz >=2800
AND     art="laptop"
```

→

marke	art	mhz
Dell	Desktop	2800
Dell	Laptop	2200
Dell	Laptop	3200
Dell	Tablet	2200
Lenovo	Laptop	2800

```
SELECT  DISTINCT marke, art, mhz
FROM    PC
WHERE   (marke="Dell" OR mhz >=2800)
AND     art="laptop"
```

→

marke	art	mhz
Dell	Laptop	2200
Dell	Laptop	3200
Lenovo	Laptop	2800

Merke: Die Bindung von AND ist stärker. Abhilfe schafft Klammerung

Was ist denn nun richtig?

Einfache Anfrage – Zusatz zu WHERE

- Mit LIKE können Zellen gegen Textmuster geprüft werden, ohne dass eine vollständige Übereinstimmung notwendig ist.
  - Der Unterstrich (_) fungiert als Platzhalter für ein Zeichen
    In MS Access ist dies das Fragezeichen (?).
  - Das Prozentzeichen (%) schließt 0 bis mehrere Zeichen ein
    In MS Access ist dies das Sternchen (*).

```
SELECT   marke , mhz
FROM     PC
WHERE    marke LIKE "*e*"
```


marke	mhz
Dell	2800
Lenovo	2400
Dell	2800
Lenovo	1600
Dell	2200
Dell	2200
Lenovo	2800
Dell	3200

GROUP BY

- Das optionale GROUP BY fasst alle Zeilen, die in einer oder mehreren Spalten den gleichen Wert enthalten, in jeweils einer Gruppe zusammen.

Beispiel: Zeige die Anzahl aller Marken

```
SELECT      marke, COUNT(*) AS Anzahl
FROM        PC
GROUP BY    marke
```

→

marke	Anzahl
Dell	5
HP	4
Lenovo	3
Sony	3

Beispiel: Zeige die Anzahl aller Arten und Mhz

```
SELECT      art, mhz, COUNT(*) AS Anzahl
FROM        PC
GROUP BY    art, mhz
```

→

art	mhz	Anzahl
Desktop	2800	3
Desktop	3000	1
Desktop	3400	1
Laptop	1600	2
Laptop	2200	1
Laptop	2400	1
Laptop	2800	1
Laptop	3200	1
Tablet	1200	2
Tablet	2200	2

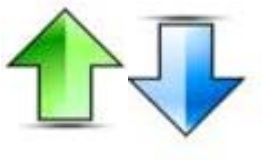
ORDER BY

- Das optionale ORDER BY kann die Ausgabe sortieren.
- Zur Verfügung stehen ASC und DESC.
  Wird keines angegeben, wird automatisch ASC angenommen

```
SELECT      art, mhz
FROM        PC
WHERE       marke ="Dell"
ORDER BY    mhz DESC
```

→

art	mhz
Laptop	3200
Desktop	2800
Desktop	2800
Tablet	2200
Laptop	2200



ASC =

DESC =

LIMIT / TOP

- ORDER BY zeigt alle Datensätze sortiert an. Soll nur eine bestimmte Anzahl angezeigt werden, so kann die Abfrage je nach DBS mit LIMIT oder TOP erweitert werden.

(My)SQL

```
SELECT      art, mhz
FROM        PC
WHERE       marke ="Dell"
ORDER BY    mhz DESC
LIMIT 2
```

MS Access

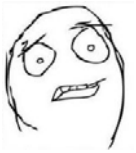
```
SELECT TOP 2  art, mhz
FROM          PC
WHERE         marke ="Dell"
ORDER         BY mhz DESC
```

→

art	mhz
Laptop	3200
Desktop	2800
Desktop	2800

Verbundanfragen

- Bisher wurde nur eine Relation abgefragt. Viele Auswertungswünsche lassen sich aber nur betrachten, wenn mehrere Relationen einbezogen werden.



⟶ Wer hat den Rechner mit der Nummer 45 gekauft?

- Anfragen, bei denen mehrere Tabellen im FROM-Teil verwendet werden, bezeichnet man als **Verbund**- oder **Join**-Anfragen.

Verbundanfragen - Methodik

Methodik ohne Datenbank

- Rechnernummer aus der Tabelle KAUF raussuchen

pc-nr	ku-nr	datum
48	1	09.10.2013
36	2	06.10.2013
78	3	11.10.2013
88	3	11.10.2013
54	4	
34	5	19.10.2013
45	6	18.10.2013
85	6	07.10.2013
46	7	04.10.2013
65	7	09.10.2013
98	8	03.10.2013

- Aus der Tabelle KUNDE den Kunden zur Kundennummer bestimmen.

ku-nr	name	plz	ort
1	Kob	32657	Lemgo
2	Larsson	44149	Dortmund
3	Folgmann	32657	Lemgo
4	Töws	32657	Lemgo
5	Ulbrick	32756	Detmold
6	Bauer	22105	Hannover
7	Heinz	10111	Berlin
8	Schulte	50541	Frankfurt
9	Albrecht	33052	Paderborn

Verbundanfragen - Methodik

Methodik mit Datenbank und SQL

- in dem FROM Teil die Abzufragen Tabellen auflisten (KUNDE und KAUF)
- in dem WHERE Teil festlegen, unter welchen Bedingungen ein Kunden-Datensatz zu einem Kauf-Datensatz gehört

```
SELECT  name, datum
FROM    KUNDE, KAUF
WHERE   KAUF.kunr = KUNDE.kunr
AND     KAUF.pcnr = 45
```

→

name	datum
Bauer	18.10.2013

Verbundanfragen - Methodik

Methodik mit Datenbank und SQL

- Werden zwei Tabellen mit-einander verknüpft, so wird jeder Datensatz der ersten Tabelle mit jedem Datensatz der zweiten Tabelle verknüpft
- Es entsteht das „kartesische Produkt“

```
SELECT  *
FROM    KUNDE, KAUF
```

KUNDE	name	plz	ort	KAUF.I	pcnr	datum
4	Töws	32657	Lemgo	4	54	12.10.2013
1	Kob	32657	Lemgo	4	54	12.10.2013
2	Larsson	44149	Dortmund	4	54	12.10.2013
3	Folgmann	32657	Lemgo	4	54	12.10.2013
5	Ulbrick	32756	Detmold	4	54	12.10.2013
6	Bauer	22105	Hannover	4	54	12.10.2013
7	Heinz	10111	Berlin	4	54	12.10.2013
8	Schulte	50541	Frankfurt	4	54	12.10.2013
9	Albrecht	33052	Paderborn	4	54	12.10.2013
4	Töws	32657	Lemgo	6	45	18.10.2013
1	Kob	32657	Lemgo	6	45	18.10.2013
2	Larsson	44149	Dortmund	6	45	18.10.2013
3	Folgmann	32657	Lemgo	6	45	18.10.2013
5	Ulbrick	32756	Detmold	6	45	18.10.2013
6	Bauer	22105	Hannover	6	45	18.10.2013
7	Heinz	10111	Berlin	6	45	18.10.2013
8	Schulte	50541	Frankfurt	6	45	18.10.2013
9	Albrecht	33052	Paderborn	6	45	18.10.2013
4	Töws	32657	Lemgo	7	46	04.10.2013
1	Kob	32657	Lemgo	7	46	04.10.2013
2	Larsson	44149	Dortmund	7	46	04.10.2013

Verkürzte Darstellung

Verbundanfragen - Methodik

Methodik mit Datenbank und SQL

- Mit dem kartesischen Produkt werden viele sinnlose Kombinationen aufgelistet.
- Um die wenigen sinnvollen zu finden, muss eine Bedingung formuliert werden, bei der jeder Kunden-Datensatz mit dem zugehörigen Kauf-Datensatz verknüpft wird

```
SELECT  *
FROM    KUNDE, KAUF
WHERE   KAUF.kunr = KUNDE.kunr
```

KUNDE	name	plz	ort	KAUF.k	pcnr	datum
4	Töws	32657	Lemgo	4	54	12.10.2013
1	Kob	32657	Lemgo	1	48	09.10.2013
2	Larsson	44149	Dortmund	2	36	06.10.2013
3	Folgmann	32657	Lemgo	3	78	11.10.2013
3	Folgmann	32657	Lemgo	3	88	11.10.2013
5	Ulbrick	32756	Detmold	5	34	19.10.2013
6	Bauer	22105	Hannover	6	45	18.10.2013
6	Bauer	22105	Hannover	6	85	07.10.2013
7	Heinz	10111	Berlin	7	46	04.10.2013
7	Heinz	10111	Berlin	7	65	09.10.2013
8	Schulte	50541	Frankfurt	8	98	03.10.2013

Verbundanfragen - Methodik

Methodik mit Datenbank und SQL

- Aus den so gewonnenen sinnvollen Kombinationen muss noch der gewünschte Datensatz ausgewählt werden…

```
SELECT  *
FROM    KUNDE, KAUF
WHERE   KAUF.kunr = KUNDE.kunr
AND     KAUF.pcnr = 45
```

KUNDE	name	plz	ort	KAUF.k	pcnr	datum
6	Bauer	22105	Hannover	6	45	18.10.2013

- ..und dann die Projektion auf die Ergebnistabelle erstellt werden

```
SELECT  name, datum
FROM    KUNDE, KAUF
WHERE   KAUF.kunr = KUNDE.kunr
AND     KAUF.pcnr = 45
```


name	datum
Bauer	18.10.2013

Verbundanfragen – weitere Beispiele

- In welchen Filialen wurde ein Lenovo Rechner gekauft?

```
SELECT DISTINCT ort
FROM            FILIALE, PC
WHERE           FILIALE.flnr=PC.flnr
AND             marke="Lenovo";
```

→

ort
Bielefeld
Paderborn

- Welche Kunden haben sich für einen Rechner mit SSD entschieden?

```
SELECT  name, ort, art
FROM    KUNDE, PC, KAUF
WHERE   KAUF.pcnr = PC.pcnr
AND     KAUF.kunr = KUNDE.kunr
AND     ssd="j"
```

→

name	ort	art
Ulbrick	Detmold	Desktop
Folgmann	Lemgo	Tablet
Bauer	Hannover	Desktop
Heinz	Berlin	Laptop
Larsson	Dortmund	Laptop
Bauer	Hannover	Tablet
Schulte	Frankfurt	Tablet

Unterabfragen

- werden zur Durchführung einer Abfrage oder eines Befehls Informationen benötigt, die zuerst durch eine eigene Abfrage geholt werden müssen, spricht man von Unterabfragen.

Beispiel: Welche Marke hat der teuerste Rechner?

```
SELECT  marke, preis
FROM    PC
WHERE   preis=
        (SELECT MAX(preis) FROM PC);
```

→

marke	preis
Dell	1.899,00 €

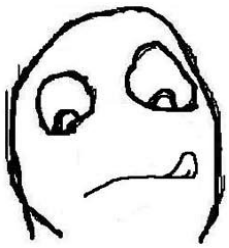
Beispiel: Welcher Kunde hat beim Rechnerkauf am meisten Geld ausgegeben?

→

name
Larsson

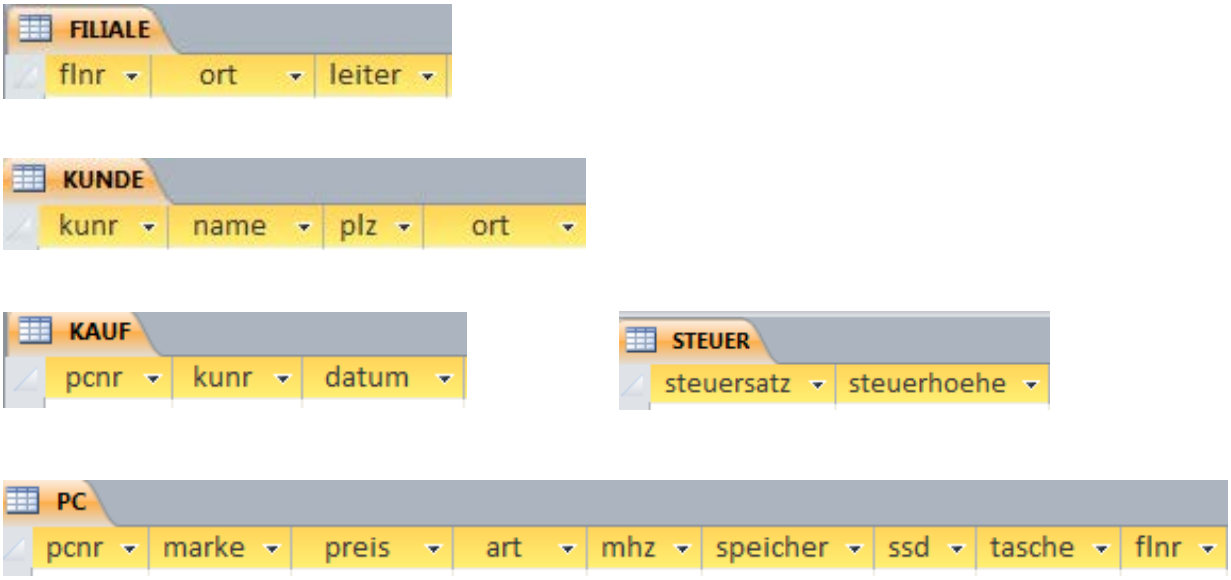
Zusammenfassende Fragen: Abschnitt 1.4

- Wie ist der SELECT Befehl ausgelegt?
- Wie können Bedingungen – auch über mehrere Tabellen – gestellt werden?



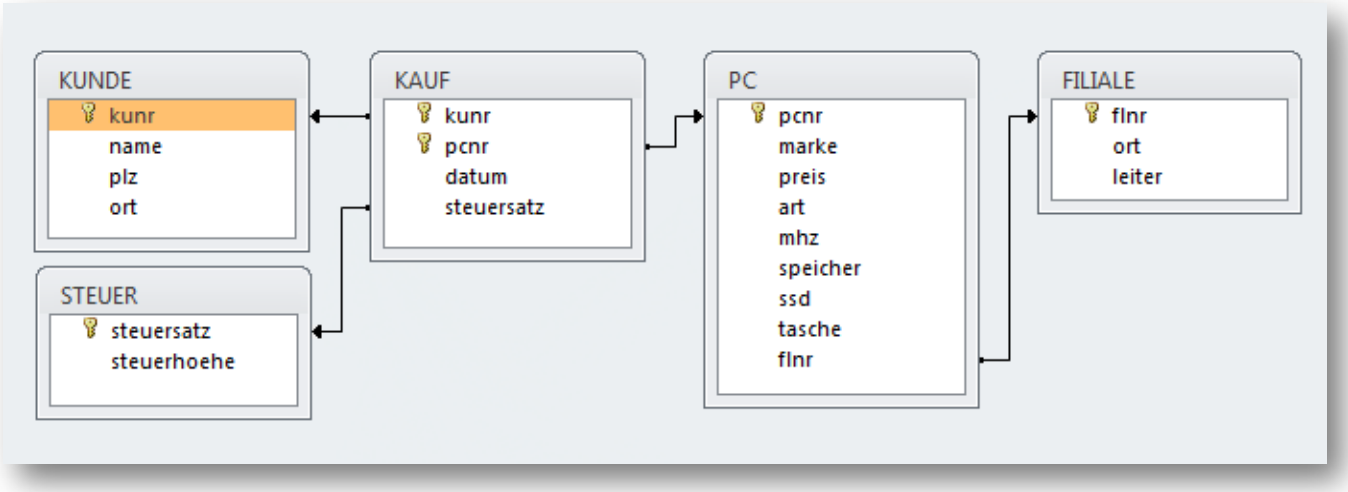
Praktikum Datenbanken

- Bilden Sie die Tabellen (inklusive Datentypen) in MS Access ab.



Praktikum Datenbanken

- Bilden Sie die Datenbeziehungen in MS Access ab
  (unter Tabelle>Beziehungen)



Praktikum Datenbanken

- 1. Erstellen Sie eine Abfrage, die die gesamte Tabelle PC ausgibt

PC

pcnr	marke	preis	art	mhz	speicher	ssd	tasche	flnr
34	Dell	999,00 €	Desktop	2800	128	j	n	1
36	Dell	1.899,00 €	Laptop	3200	1000	j	j	1
45	Dell	1.199,00 €	Desktop	2800	256	j	n	1
46	HP	499,00 €	Laptop	1600	64	j	n	3
48	HP	1.499,00 €	Desktop	3000	3000	n	n	1
54	Lenovo	1.299,00 €	Laptop	2400	1000	n	j	2
63	Sony	1.499,00 €	Desktop	3400	1000	j	n	1
65	Dell	899,00 €	Laptop	2200	2000	n	j	2
75	Sony	849,00 €	Desktop	2800	512	j	n	3
76	Dell	999,00 €	Tablet	2200	256	j	n	1
78	HP	989,00 €	Tablet	1200	128	j	j	2
84	Lenovo	1.199,00 €	Laptop	2800	2000	n	j	3
85	HP	399,00 €	Tablet	1200	64	j	j	2
88	Lenovo	499,00 €	Laptop	1600	2000	n	n	3
98	Sony	1.249,00 €	Tablet	2200	256	j	j	2

- 2. Reduzieren Sie die Abfrage so, dass nur die Marke, die Art und der Preis angezeigt werden

marke	art	preis
Dell	Desktop	999,00 €
Dell	Laptop	1.899,00 €
Dell	Desktop	1.199,00 €
HP	Laptop	499,00 €
HP	Desktop	1.499,00 €
Lenovo	Laptop	1.299,00 €
Sony	Desktop	1.499,00 €
Dell	Laptop	899,00 €
Sony	Desktop	849,00 €
Dell	Tablet	999,00 €
HP	Tablet	989,00 €
Lenovo	Laptop	1.199,00 €
HP	Tablet	399,00 €
Lenovo	Laptop	499,00 €
Sony	Tablet	1.249,00 €

Praktikum Datenbanken

- 3. Benennen Sie die Spalte art in „Bauform“ um.

marke	Bauform	preis
Dell	Desktop	999,00 €
Dell	Laptop	1.899,00 €
Dell	Desktop	1.199,00 €
HP	Laptop	499,00 €
HP	Desktop	1.499,00 €
Lenovo	Laptop	1.299,00 €
Sony	Desktop	1.499,00 €
Dell	Laptop	899,00 €
Sony	Desktop	849,00 €
Dell	Tablet	999,00 €
HP	Tablet	989,00 €
Lenovo	Laptop	1.199,00 €
HP	Tablet	399,00 €
Lenovo	Laptop	499,00 €
Sony	Tablet	1.249,00 €

- 4. Zeigen Sie alle Marken und Arten von Rechnern an. Kein Eintrag ist doppelt.

marke	Bauform
Dell	Desktop
Dell	Laptop
Dell	Tablet
HP	Desktop
HP	Laptop
HP	Tablet
Lenovo	Laptop
Sony	Desktop
Sony	Tablet

Praktikum Datenbanken

- 5. Zeigen Sie alle Marken und Arten sowie Preise von Rechnern an. Der Preis ist allerdings in Dollar umgerechnet (Kurs 1,381)

marke	Bauform	Dollar
Dell	Desktop	1379,619
Dell	Laptop	2622,519
Dell	Desktop	1655,819
HP	Laptop	689,119
HP	Desktop	2070,119
Lenovo	Laptop	1793,919
Sony	Desktop	2070,119
Dell	Laptop	1241,519
Sony	Desktop	1172,469
Dell	Tablet	1379,619
HP	Tablet	1365,809
Lenovo	Laptop	1655,819
HP	Tablet	551,019
Lenovo	Laptop	689,119
Sony	Tablet	1724,869

- 6. Mit der Funktion ROUND(wert, anz_stellen) lassen sich Werte runden. Runden Sie die Dollar Werte

marke	Bauform	Dollar
Dell	Desktop	1379,62
Dell	Laptop	2622,52
Dell	Desktop	1655,82
HP	Laptop	689,12
HP	Desktop	2070,12
Lenovo	Laptop	1793,92
Sony	Desktop	2070,12
Dell	Laptop	1241,52
Sony	Desktop	1172,47
Dell	Tablet	1379,62
HP	Tablet	1365,81
Lenovo	Laptop	1655,82
HP	Tablet	551,02
Lenovo	Laptop	689,12
Sony	Tablet	1724,87

Praktikum Datenbanken

- 7. Was ist der höchste Speicher den ein Rechner hat?

max_Speicher
3000

- 8. Wieviel Megahertz haben die Rechner durchschnittlich?

MHZ_Schnitt
2360

Praktikum Datenbanken

- 9. Geben Sie alle Rechner aus, die mehr als 1000 MB Speicher haben

marke	Bauform	speicher
Lenovo	Laptop	2000
Dell	Laptop	2000
HP	Desktop	3000
Lenovo	Laptop	2000

- 10. Erweitern Sie die Abfrage aus (8) auf zusätzlich alle Dell-Rechner

marke	Bauform	speicher
Dell	Desktop	128
Dell	Desktop	256
Lenovo	Laptop	2000
Dell	Laptop	2000
Dell	Tablet	256
HP	Desktop	3000
Lenovo	Laptop	2000
Dell	Laptop	1000

Praktikum Datenbanken

- 11. Zeigen Sie alle Rechner, die eine SSD haben, unter 1000 Euro liegen und von Dell kommt

marke	Bauform	preis
Dell	Desktop	999,00 €
Dell	Tablet	999,00 €

- 12. Zeigen Sie alle Rechner, die entweder eine SSD haben oder mehr als 2400 mhz . In jedem Fall ist es ein Desktop-Rechner

marke	Bauform	preis
Dell	Desktop	999,00 €
Dell	Desktop	1.199,00 €
Sony	Desktop	1.499,00 €
HP	Desktop	1.499,00 €
Sony	Desktop	849,00 €

Praktikum Datenbanken

- 13. Was ist der Gesamtwert aller Dell Rechner?

Dell_Gesamtwert
5.995,00 €

- 14. Wie viele Laptops mit Tasche gibt es?

Anzahl_Laeppi_Tasche
4

Praktikum Datenbanken

- 15. Sortieren Sie die Abfrage aus (10) nach dem Preis in absteigender Reihenfolge

marke	Bauform	preis
HP	Desktop	1.499,00 €
Sony	Desktop	1.499,00 €
Dell	Desktop	1.199,00 €
Dell	Desktop	999,00 €
Sony	Desktop	849,00 €

- 16. Zeigen Sie alle Rechner, die in Filiale 2 liegen und kein Laptop sind

marke	Bauform	flnr
HP	Tablet	2
HP	Tablet	2
Sony	Tablet	2

Praktikum Datenbanken

- 17. Verbünde: In welcher Filiale kann man Dell-Laptops kaufen?

marke	art	ort
Dell	Laptop	Bielefeld
Dell	Laptop	Lemgo

- 18. Welche Kunden haben in Filiale 2 eingekauft?

name	ort	flnr
Töws	Lemgo	2
Folgmann	Lemgo	2
Heinz	Berlin	2
Bauer	Hannover	2
Schulte	Frankfurt	2

Praktikum Datenbanken

- 19. Welche Kunden konnten von Steuersatz 2 profitieren?

name
Bauer
Folgmann

- 20. In welcher Filiale haben diese Kunden gekauft?

name	Kaufort
Bauer	Bielefeld
Bauer	Lemgo
Folgmann	Paderborn

Praktikum Datenbanken

- 21. Wie viele Rechner hat der Kunde mit der Kundennummer 3 gekauft?

kunr	name	art
3	Folgmann	Laptop
3	Folgmann	Tablet

- 22. Wo hat er sie gekauft?

kunr	name	art	ort
3	Folgmann	Laptop	Paderborn
3	Folgmann	Tablet	Bielefeld

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Praktikum Datenbanken

- 23. Welche Kunden haben an welchem Ort Rechner gekauft, die mehr als 1000 Euro gekostet haben?

kunr	name	art	ort	preis
4	Töws	Laptop	Bielefeld	1.299,00 €
6	Bauer	Desktop	Lemgo	1.199,00 €
1	Kob	Desktop	Lemgo	1.499,00 €
2	Larsson	Laptop	Lemgo	1.899,00 €
8	Schulte	Tablet	Bielefeld	1.249,00 €

- 24. Wer hat Laptops mit Taschen und SSDs gekauft?

kunr	name
2	Larsson

Praktikum Datenbanken

- 25. Was haben alle Rechner, die in Paderborn vertickt wurden, an Umsatz gebracht?

Paderborner_Umsatz
998,00 €

- 26. In welcher Filiale liegt der günstigste Rechner, der mit dem geringsten Steuersatz versteuert wird? Schnäppchenalarm! ☺

ort
Bielefeld

Softwareengineering

1 Nutzwertanalyse

2 IT-Projekte

3 Vorgehensmodelle

4 Geschäftsprozessmanagement

5 UML Diagramme

Problemstellung und Idee

Es müssen Entscheidungen zur Auswahl von verschiedenen Lösungen/Alternativen getroffen werden.

- Unbewusst / bewusst
- Privat / im Unternehmen

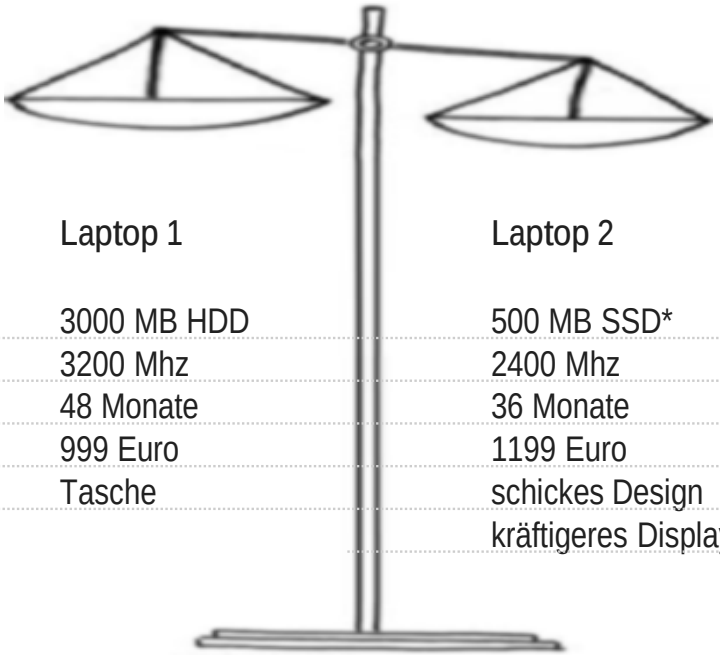
Lösungen können sein:

- Produkte (z.B. Auswahl einer Festplatte, einer Maschine, eines Kredites)
- Systeme (z.B. Auswahl eines ERP-Systems)
- Dienstleistungen (z.B. Beratung, Reinigungsfirma)

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Problemstellung und Idee

Die zentrale Frage lautet:
Welche Lösung ist die „beste“?



	Laptop 1	Laptop 2
Festplatte	3000 MB HDD	500 MB SSD*
Prozessor	3200 Mhz	2400 Mhz
Garantie	48 Monate	36 Monate
Preis	999 Euro	1199 Euro
Weiteres	Tasche	schickes Design kräftigeres Display

HDD =Harddiscdrive: mechanische, aber meist sehr große Festplatte
SSD =SolidStateDisc: bis zu 10mal schnellere Festplatte mit Chip-Technologie

Welche Probleme bestehen bei der Auswahl?

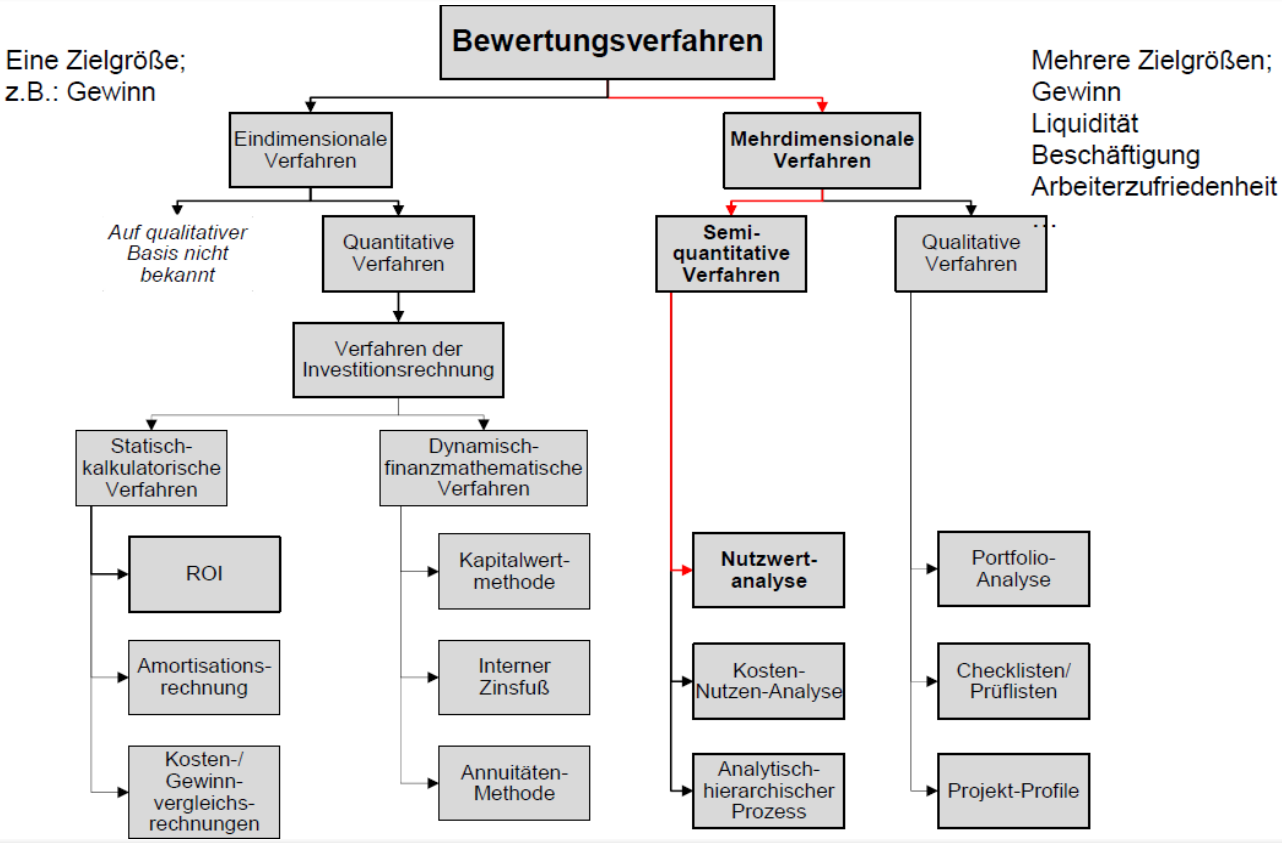
Problemstellung und Idee

Lösung: Einsatz eines Verfahrens

- Die **Nutzwertanalyse** (NWA, auch Scoring- Modell, Punktwertverfahren, Multifaktoren-Methode, Utility Analysis) ist ein Verfahren zur Bewertung von Handlungsalternativen
- Sie ermöglicht die Berücksichtigung mehrerer, beliebiger, komplexerer Ziele.
- Einzelnen Kriterien wird ein Nutzen zugewiesen; die Alternative mit dem höchsten Gesamtnutzen wird gewählt

Definition: *„Analyse einer Menge komplexer Handlungsalternativen mit dem Zweck, die Elemente dieser Menge entsprechend den Präferenzen des Entscheidungsträgers bezüglich eines multidimensionalen Zielsystems zu ordnen“ (C. Zangenmeister, 1976)*

Problemstellung und Idee



Quelle: Endrik Lengwenat , TU München

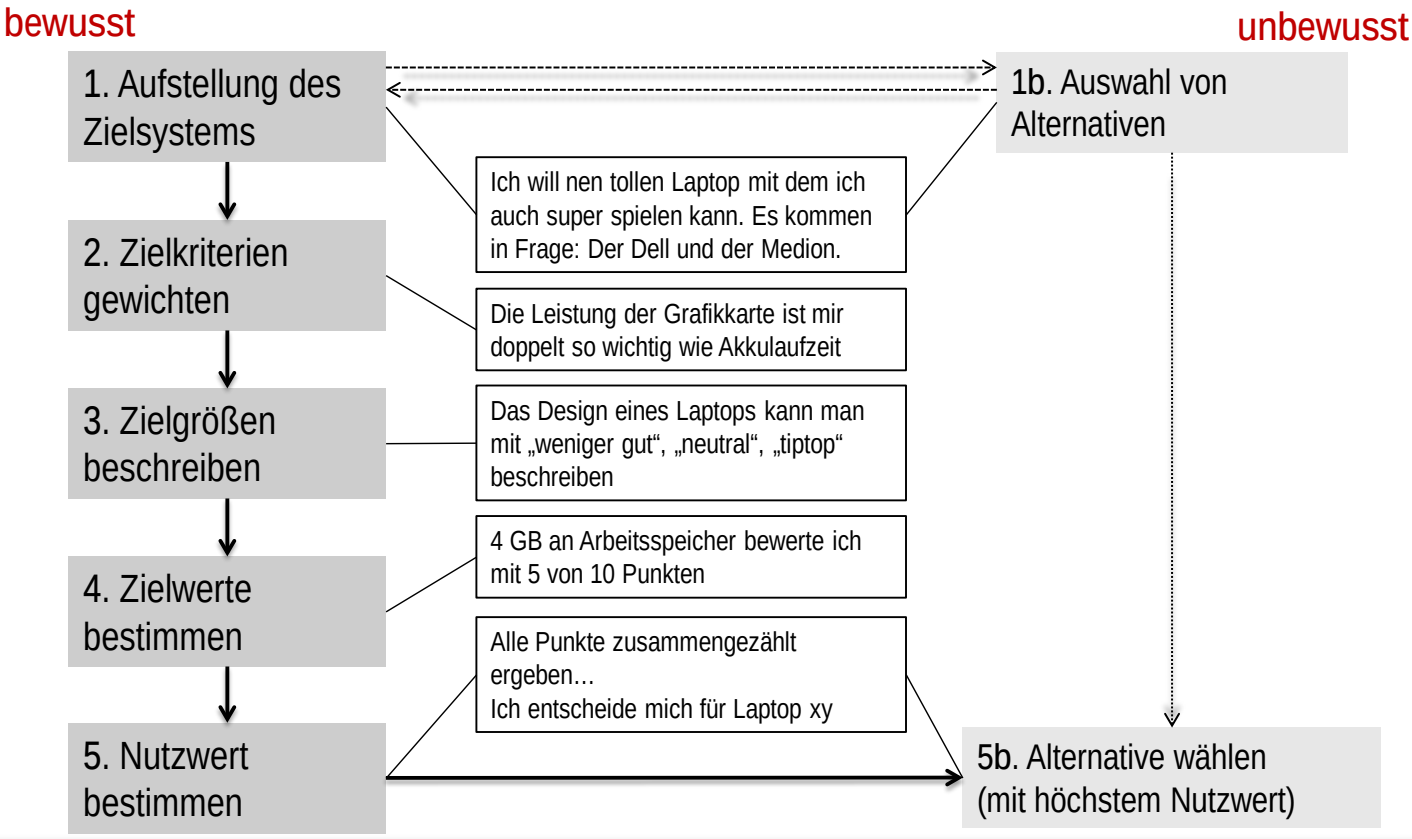
Problemstellung und Idee

- Sie wurde als Erweiterung der Kosten-Nutzen-Analyse entwickelt, da sie nicht nur monetäre Bewertungen sondern auch die persönlichen Präferenzen des Entscheiders berücksichtigt.

- Die Nutzwertanalyse erlaubt bei einer großen Anzahl von Alternativen mit vergleichsweise geringem Aufwand eine erste, grobe Beurteilung von Investitionen oder Projekte.

- Diese Methode kann für eine Vielzahl von Anwendungsgebieten eingesetzt werden, bei denen quantitative Daten fehlen, wie z. B. die Arbeitsplatzbewertung, die Standortwahl, die Erklärung von Käuferverhalten, die Vorauswahl von Investitionsobjekten etc.

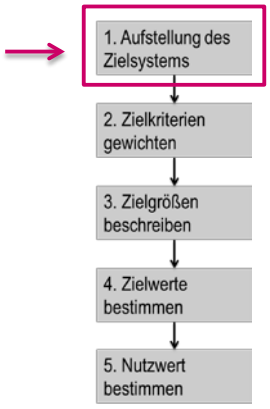
Vorgehen (Übersicht)

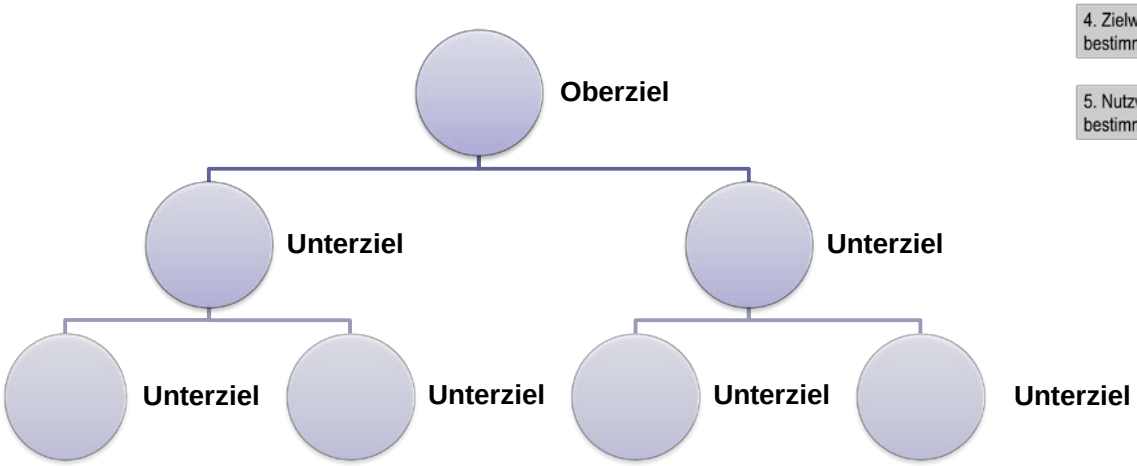


Natürlich kann, bei der bewussten Entscheidung durch eine NWA, die Auswahl von Alternativen an beliebiger Stelle vor Schritt 4 erfolgen. In der Regel handelt es sich um wenige Alternativen. Falls nicht, sollte die Auswahl „mit gesundem Menschenverstand“, z.B. anhand von bekannten KO-Kriterien, reduziert werden.

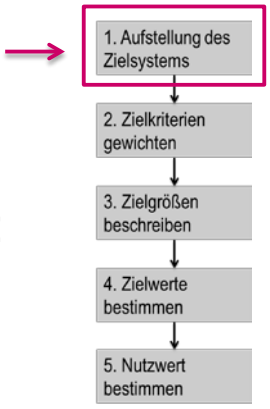
Vorgehen: 1. Zielsystem

Ein Zielsystem kann mit Hilfe eines Zielbaums visualisiert werden.





Vorgehen: 1. Zielsystem



Anforderungen:

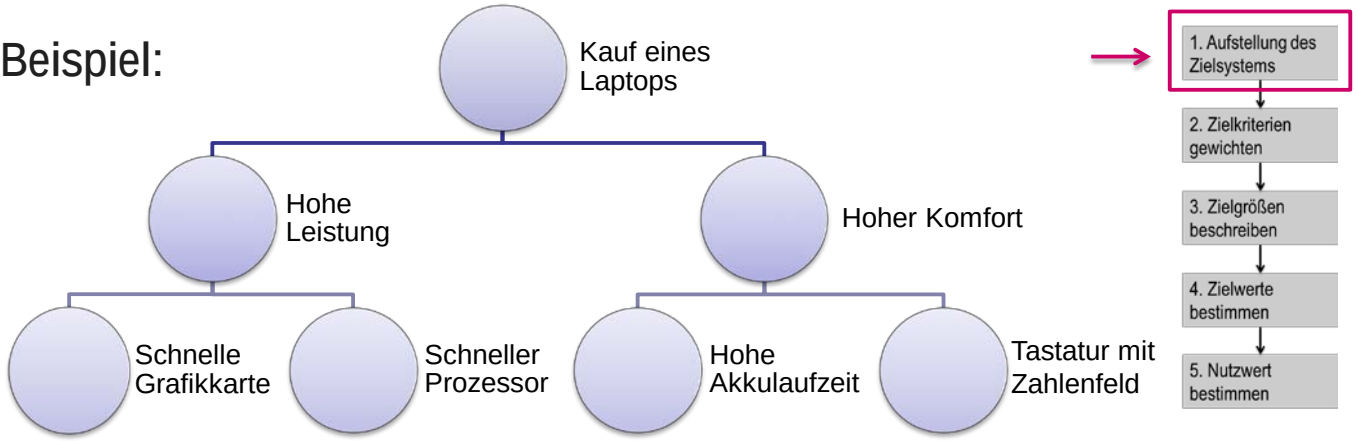
Unabhängigkeit: Die Kriterien dürfen nicht in einem Zusammenhang stehen. Beispiel für fehlende Unabhängigkeit: Fahrtzeit und verkehrsgünstige Lage.

Genaue Formulierung: Beispiel: das Kriterium „Erfolg“ ist mehrdeutig und muss entsprechend genauer beschrieben werden. Dazu eignet sich eine Unterkategorie.

Vollständigkeit: Die Bewertungskriterien sollen unterschiedliche Objekteigenschaften abfragen, um Einseitigkeit in der Beurteilung zu vermeiden.

Vorgehen: 1. Zielsystem

Beispiel:

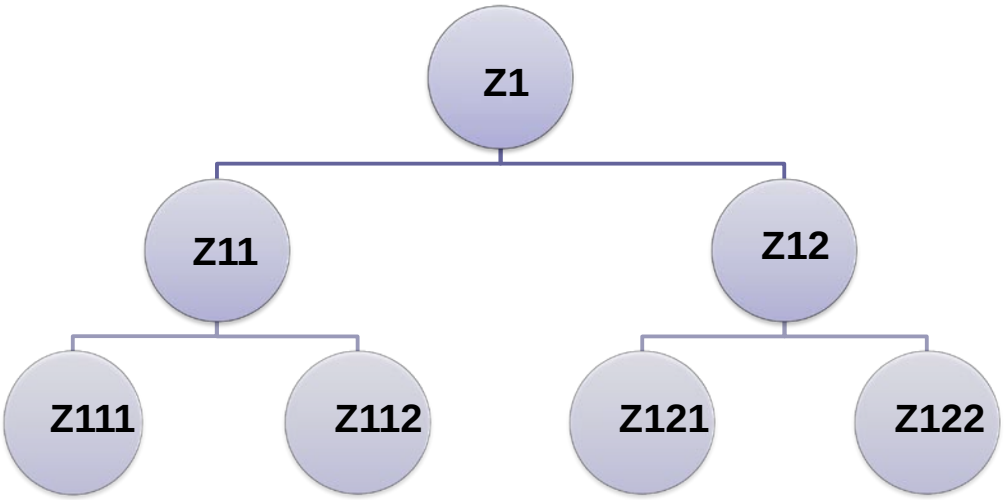


- Die Zielkriterien werden i.d.R. durch mehrere Personen aus unterschiedlichen Unternehmensbereichen festgelegt.
- Bei der Auswahl eines komplexen Systems empfiehlt sich die Erstellung eines Lastenheftes

(Das Lastenheft wird vom Auftraggeber oder in dessen Auftrag erstellt. Hier werden alle Anforderungen, Erwartungen und Wünsche an ein geplantes Produkt (Objekt) in natürlicher Sprache erfasst.)

Vorgehen: 2. Zielkriterien gewichten

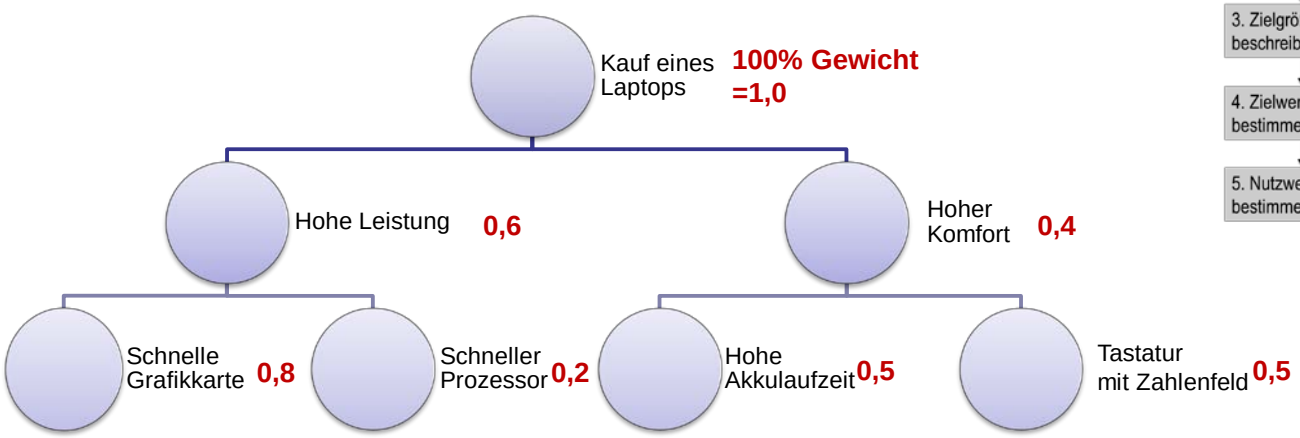
die Zielkriterien werden entsprechend ihrem Einfluss auf den Gesamtnutzen gewichtet. In einer Zielstufe ist die Summe der absoluten Kriteriengewichte = 1.



Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Vorgehen: 2. Zielkriterien gewichten

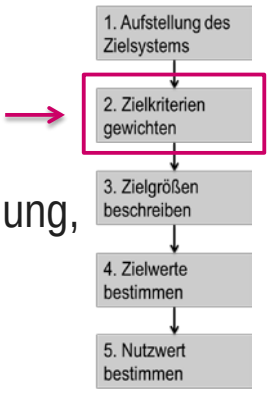
Beispiel:



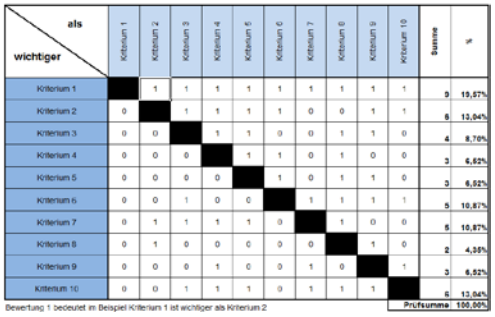
Problem: wie kann die Rangfolge ermittelt werden?

Vorgehen: 2. Zielkriterien gewichten

Lösung: Paarweiser Vergleich



- Vergleich jedes Einzelkriteriums mit jedem anderen und Festlegung einer Wichtigkeit in der direkten Gegenüberstellung, wobei sich die Entscheidungsmenge auf (A wichtiger B), (A gleichwertig B), und (A unwichtiger B) reduziert.
- Soll die Entscheidung einen Zwang zur Reihenfolgenbildung enthalten, kann die Gleichwertigkeit entfallen.
- Die Summe dieser Einzelbewertungen führt auf eine Aussage zur Gesamtwichtigkeit jedes Einzelkriteriums in Form einer Rangziffer.



als / wichtiger	Kriterium 1	Kriterium 2	Kriterium 3	Kriterium 4	Kriterium 5	Kriterium 6	Kriterium 7	Kriterium 8	Kriterium 9	Kriterium 10	Summe	%
Kriterium 1		1	1	1	1	1	1	1	1	1	9	19,57%
Kriterium 2	0		1	1	1	1	0	0	1	1	6	13,04%
Kriterium 3	0	0		1	1	0	0	1	1	0	4	8,70%
Kriterium 4	0	0	0		1	1	0	1	0	0	3	6,52%
Kriterium 5	0	0	0	0		1	0	1	1	0	3	6,52%
Kriterium 6	0	0	1	0	0		1	1	1	1	5	10,87%
Kriterium 7	0	1	1	1	1	0		1	0	0	5	10,87%
Kriterium 8	0	1	0	0	0	0	0		1	0	2	4,35%
Kriterium 9	0	0	0	1	0	0	1	0		1	3	6,52%
Kriterium 10	0	0	1	1	1	0	1	1	1		6	13,04%
										Prüfsumme		100,00%

Bewertung 1 bedeutet im Beispiel Kriterium 1 ist wichtiger als Kriterium 2

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Vorgehen: 2. Zielkriterien gewichten/Paarweiser Vergleich

wichtiger \ als	Kriterium 1	Kriterium 2	Kriterium 3	Kriterium 4	Kriterium 5	Kriterium 6	Kriterium 7	Kriterium 8	Kriterium 9	Kriterium 10	Summe	%
Kriterium 1		1	1	1	1	1	1	1	1	1	9	20,00%
Kriterium 2			1	1	1	1	0	0	1	1	6	13,33%
Kriterium 3				1	1	0	0	1	1	0	4	8,89%
Kriterium 4					1	1	0	1	0	0	3	6,67%
Kriterium 5						1	0	1	1	0	3	6,67%
Kriterium 6							1	1	1	1	5	11,11%
Kriterium 7								1	0	0	5	11,11%
Kriterium 8									1	0	2	4,44%
Kriterium 9										1	3	6,67%
Kriterium 10											5	11,11%
											Prüfsumme	100,00%

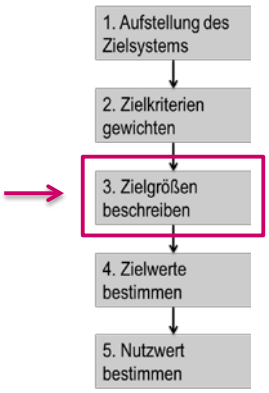
Bewertung 1 bedeutet im Beispiel Kriterium 1 ist wichtiger als Kriterium 2

Quelle: korrigiert von: http://www.sixsigmablackbelt.de/paarweiser-vergleich/

☺ Aufgabe:
Vervollständigen Sie die Tabelle!

Vorgehen: 3. Beschreibung der Zielgrößen

- Anhand der Zielkriterien getätigte Beschreibung der bekannten qualitativen und quantitativen Eigenschaften (sog. Zielgrößen) der Varianten.



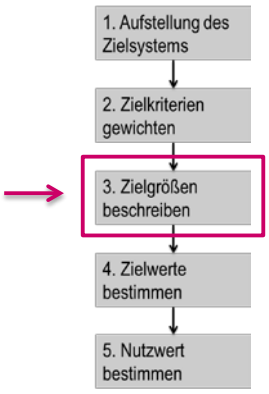
- Punktevergabe (1-10) durch Experte(n)
- Bsp.: Design kann als „schlecht (0 Punkte)“, „mittel“ (5 Punkte) und „super“ (10 Punkte) beschrieben werden
- Messung Anhand von Skalierungen
  - Nominale Skalierungen
  - Ordinale Skalierungen
  - Intervallskalen
  - Verhältnisskalen

Vorgehen: 3. Beschreibung der Zielgrößen

Nominale Skalierung

Einfachste Form der Skalierung

- Die Klassifizierung eines Objektes erfolgt darüber, ob einem Bewertungskriterium entsprochen wird oder nicht.
- Beispiel: „ja/nein“, „gut/schlecht“, „+/-“.



Ordinale Skalierung

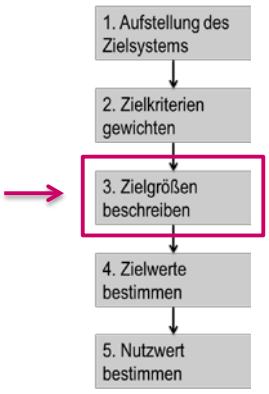
differenzieren stärker die Richtung der Nutzenunterschiede.

- Die Rangordnung wird garantiert.
- Beispiel: „größer“, „kleiner“, „gleich“ etc.

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Vorgehen: 3. Beschreibung der Zielgrößen

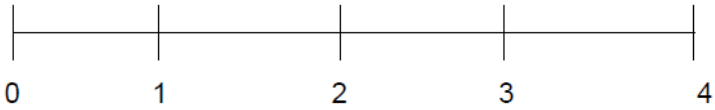
Intervallskala



weisen den Abständen der Rangordnung feste Intervalle zu.

- Die Stufen der Skalen sind immer gleich groß
- Relativer Nullpunkt

- Beispiele: „sehr starke Leistung= 5 Punkte“, „sehr schwache Leistung= 0 Punkte“. Temperaturmessung in °C



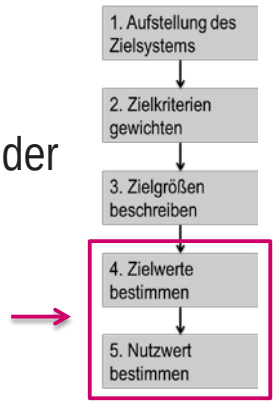
Verhältnisskala

fordern neben den festen Intervallen
auch die exakte Bestimmung eines absoluten Nullpunktes.

- Beispiel: Gewichtsskala.

Vorgehen: 4.+5. Ziel- und Nutzwerte bestimmen

Auch: Wertsynthese



- In einer Zielwertmatrix werden die Alternativen nebeneinander aufgestellt
- Die jeweiligen Kriterien werden untereinander gestellt
- Berechnung

$$Nj = \sum wk * Pjk$$

- Nj: der Nutzwert der Alternative j
- wk: Gewichtung, die dem Kriterium k zugewiesen wird
- Pjk: Punktwert, den eine Alternative j beim Beurteilungskriterium k aufgrund eines Expertenurteils erreicht
- k: das Beurteilungskriterium

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Vorgehen: 4.+5. Ziel- und Nutzwerte bestimmen

		Alternative 1		Alternative 2		Alternative 3		Alternative 4	
	Gewichtung	Bewertung	Wert	Bewertung	Wert	Bewertung	Wert	Bewertung	Wert
Kriterium 1	20,00%	10	2,00	10	2,00	7	1,40	3	0,60
Kriterium 2	13,33%	4	0,53	10	1,33	8	1,07	5	0,67
Kriterium 3	8,89%	3	0,27	3	0,27	3	0,27	3	0,27
Kriterium 4	6,67%	7	0,47	4	0,27	3	0,20	4	0,27
Kriterium 5	6,67%	8	0,53	8	0,53	7	0,47	8	0,53
Kriterium 6	11,11%	4	0,44	9	1,00	9	1,00	4	0,44
Kriterium 7	11,11%	7	0,78	2	0,22	1	0,11	1	0,11
Kriterium 8	4,44%	9	0,40	3	0,13	3	0,13	3	0,13
Kriterium 9	6,67%	8	0,53	1	0,07	0	-	1	0,07
Kriterium 10	11,11%	8	0,89	1	0,11	3	0,33	1	0,11
	Summe		**6,84**		**5,93**		**4,98**		**3,20**

Zielwert für Kriterium 1 der Alternative 4

Nutzwert der Alternative 4

Bewertungszahl von 0 - 10
Bewertungszahl 0 entspricht Alternative erfüllt das Kriterium nicht
Bewertungszahl 10 entspricht Alternative erfüllt das Kriterium vollständig

Was wäre der maximal zu erreichende Wert gewesen?

Wieviel Prozent davon erreicht bspw. Alternative 1?

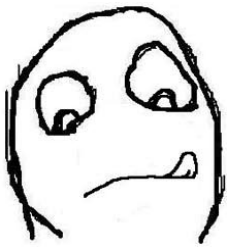
Zusammenfassung und Kritik

- Häufig eingesetztes Instrument als Entscheidungsgrundlage.
- Durch die Gliederung in Teilaspekte gute Übersicht über das zu bewertende Problem.
- Alle Schritte des Verfahrens sind formal transparent
- Berücksichtigung von mehreren Zielen, die auch nicht monetären Inhalt haben können.

- Vorauswahl der der Alternativen sind nicht formalisiert
- Die Auswahl der Kriterien ist subjektiv und die Bewertung erfolgt anhand grob geschätzter Werte.
  → Die NWA kann als Täuschungsinstrument dienen.
- Erheblicher Zeitaufwand, wenn mehrere Personen an der Bewertung beteiligt sind.

Zusammenfassende Fragen: Abschnitt 2.1

- Wozu dient das Verfahren der Nutzwertanalyse?
- Wie lässt sich die NWA einordnen?
- In welchen Schritten erfolgt eine NWA?
- Wie funktioniert ein paarweiser Vergleich?
- Wie wird der Gesamtnutzen ermittelt?
- Welche Kritik gibt es an der Methode? An welchen Stellen?



Aufgaben

Kriterium	Gewichtung (Punkte)	Alternative A	Nutzwert A	Alternative B	Nutzwert B	Alternative C	Nutzwert C
Standort:							
Hauptstraße	7	5		7		3	
Nähe an der Geschäften	9	5		3		10	
Lagernähe	8	10		8		6	
Kosten:							
Steuerliche Belastung	6	4		5		5	
Bau-/Mietkosten	15	6		7		4	
Werbung	10	7		4		8	
Gewinnpotenziale:							
Wenig Konkurrenz	15	3		4		3	
Kunden sind offen für Neues	10	5		5		6	
Arbeitslosen-quote gering	20	7		5		6	
Summe:	100						

Eine Firma möchte eine weitere Filiale in einem anderen Bundesland aufmachen. Ihr wurden drei Angebote aus verschiedenen Bundesländern gemacht und sie muss sich nun für einen Standort entscheiden. Führen Sie eine Nutzwertanalyse anhand obiger Tabelle durch.

Softwareengineering

1 Nutzwertanalyse 

2 IT-Projekte

3 Vorgehensmodelle

4 Geschäftsprozessmanagement

5 UML Diagramme

Lesen..Nachschlagen..Lernen..

Literatur zu Softwareengineering (IT-Projekte, Vorgehensmodelle, UML)

- Brandt-Pook, H.; Kollmeier, R.: Softwareentwicklung kompakt und verständlich; Wiesbaden 2008
  online via Springerlink

Definition eines Projektes

Definition DIN: „Ein Projekt ist ein zeitlich begrenztes Vorhaben zur Schaffung eines einmaligen Produktes, einer Dienstleistung oder eines Ergebnisses.“

DIN

Aus der Definition wird bereits deutlich, dass ein Projekt mit dem Erledigen einer nicht ganz einfachen Aufgabe zu tun hat.

Kriterien zur Definition eines Projektes

Kriterium: Ziele

Ein Projekt hat Ziele.
Anhand der Ziele richten sich Projekte aus.

Bei der Zielfestlegung ist einiges zu beachten:

- Ziele sollten präzise und verständlich formuliert sein.
- Die Ziele sollten vollständig und widerspruchsfrei formuliert sein.
- Ob ein Ziel erreicht wurde, sollte entscheidbar oder messbar sein.
- Sie sollten erreichbar sein.
- Sie sollten ein definiertes Ende haben

→ SMART

Kriterium: Anfang und Ende / Einmaligkeit

Ein Projekt hat einen Anfang und ein Ende.

- Beginn ist ein sog. Kick-Off-Workshop
- Projekte sollten nicht unendlich lange dauern
- Das Projektende ist erreicht, wenn alle Projektziele erreicht wurden.

Ein Projekt ist einmalig.

- Einmaligkeit ist eine wichtige Abgrenzung zum normalen Arbeitsprozess
- An individuellen Kundenwünsche orientiert
- Keine Routinearbeiten

Kriterium: Komplexität

Ein Projekt ist komplex.

- Ist gegeben sobald verschiedene Fachexperten (mit unterschiedlichen Kompetenzen) zur Erreichung eines Ziels benötigt werden.
- kann auch bedeuten, dass eine Fachkraft an einem besonders umfangreichen Projekt arbeitet.

- Anspruchsvolle Projektziele
- Oft hohes Budget
- Lange Projektlaufzeit

Kriterium: Budget / AuftraggeberIn

Ein Budget ist ein Geldbetrag, der dem Projekt zur Verfügung steht.

- Ist ein festgelegter Geldbetrag für ein Projekt,
- kann aber auch als Arbeits- und Personentage festgelegt werden.

Jedes Projekt hat einen Auftraggeber oder Auftraggeberin

- Ist der wichtigste (mächtigste) Bestandteil eines Projektes.
- gibt genaue Angaben (Aufschluss) über die Projektziele.
- Aushandeln von Verträgen ist ihre Aufgabe
- kann auch eine juristische Person sein

Kriterium: Projektorganisation

Jedes Projekt hat eine eigene Projektorganisation

- Einteilung von Rollen
- Zuständigkeiten
- Verantwortlichkeiten

Kriterium: Risiko

Jedes Projekt beinhaltet das Risiko des Scheiterns

Bedeutet nicht, dass kein Projektziel erreicht wurde und nicht das „Aus“ des Projektes, sondern lediglich das „Nicht-Erreichen“ aller Projektziele

Beispiele/Gründe

- Budgetüberschreitung
- Zeitüberschreitung
- Unerfahrene Projektleiter
- Projektziele zu anspruchsvoll
- Mangel an Teamgeist / geringe Identifikation mit den Projektzielen
- Unzureichende Berichtserstattung
- Falsche Projektplanung (Projektteam wurde nicht beteiligt)

Warum machen wir Projekte?

Projekte sind heutzutage in den meisten Unternehmen eine „klassische“ Organisationsform

- Mittel zur Bewältigung des ständigen Wettbewerbs- und Veränderungsdruckes in der Wirtschaft
- Wettbewerbs- und Veränderungsdruckes besonders in der IT außerordentlich hoch

Beispiel:
Moore‘s Law in der IT = 18 Monate vs.
Entwicklungszyklus bei VW Golf V = 5 Jahre

Arten von IT-Projekten

Projektdimensionen im Überblick:

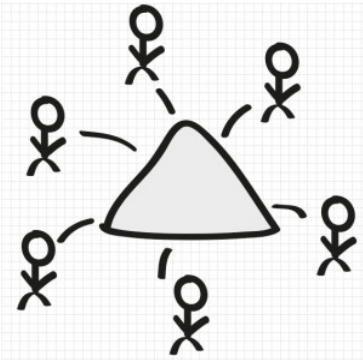
- Projektaufwand – Kleine oder große Projekte
- Projektstandorte – Zentrales oder verteiltes Projektgeschehen
- Technologie – SAP oder Linuxprojekt
- Charakter der Projektziele – Erforschungsprojekt
- Phasenschwerpunkte – Konzeptionsprojekt, Umsetzungsprojekt, Einführungsprojekt, Wartungsprojekt

Rollen in einem Projekt

Wichtige Rollen im Projektgeschehen sind:

- Der Kunde
- Der Projektleiter
- Das Teammitglied
- Der Qualitätsmanager

- Gremium Lenkungsausschuss



http://projekte-leicht-gemacht.de

Rollen in einem Projekt: Kunde

Die wichtigste Rolle im Projekt, da der Kunde das Projekt in Auftrag gibt:

- Was sind meine Ziele? Was will ich erreichen?
- Wie finde ich einen geeigneten Dienstleister?
- Welchen Beitrag kann mein Unternehmen zum Projekt beitragen?
- prüft und priorisiert (Kosten- Nutzenanalyse).
- Welche Projektaktivitäten sind bereits abgeschlossen?
- Welche Projektaktivitäten sind in Bearbeitung?
- Welche Probleme gibt es derzeit?
- Ist der geplante Endtermin zu halten?

Rollen in einem Projekt: ProjektleiterIn

Die Projektleiterin bzw. der Projektleiter leitet das Projekt.
Er/Sie ist verantwortlich für:

- Die Planung eines Projektes
- Die Führung und Steuerung des Projektteams
- Die Kontrolle der erarbeiteten Leistung
- Die Kommunikation mit dem Auftraggeber, dem Lenkungsausschuss und anderen Personen die ein Interesse am Projekt haben → Stakeholder

Rollen in einem Projekt: Teammitglied

Die Rolle Teammitglied ist ein Oberbegriff für alle fachlichen Rollen im Projekt und beinhaltet folgende Aufgaben:

- sollen die ihnen übertragenen Aufgaben erfüllen (Arbeitspakete abarbeiten).
- sollen ihre Kenntnisse auf dem neusten Stand halten.
- beteiligen sich an der Planung und nutzen ihre Fachkenntnisse.
- ergreifen Eigeninitiativen und sollen dennoch in der Gruppe funktionieren.

Exkurs: Teambildungsphase

Forming

- Orientierungsphase
  „Wo ist mein Platz in der Gruppe?“).

Storming

- Konfliktphase
  Unzufriedenheit, weil die eigene Rolle noch nicht gefunden wurde

Norming

- Teambildungsphase
  Organisation: Die eigene Rolle im Team wird klar

Performing

- Performancephase
  Leistung: Alle Beteiligten funktionieren im Team und leisten ihren Beitrag zur Zielerreichung

5-Phasen Modell nach Tuckman (1965)

- Gruppen entwickeln eine Dynamik und durchlaufen Phasen, in denen jeweils andere Themen und Bedürfnisse der Gruppenmitglieder aktuell sind
- Stadien lassen sich nicht immer exakt trennen
- Gruppen können in frühere Phasen zurückfallen
- 5. Phase (Adjourning) bezeichnet das Auseinandergehen einer Gruppe

Rollen in einem Projekt: QualitätsmanagerIn

Die Aufgabe einer QualitätsmanagerIn ist es die Qualität von Prozessen und Produkten sicherzustellen und zu verbessern.

- stellt die Qualität von Prozessen und Produkten sicher und verbessert diese gegebenenfalls.
- gibt Verbesserungsvorschläge zu Vorgehensmodellen und Tests.
- Der Qualitätsmanager ist unabhängig vom Projektleiter
- kann als Rolle auch vom Projektleiter übernommen werden (jeweils abhängig von der Projektgröße).

.

Rollen in einem Projekt: Gremium Lenkungsausschuss

Den Lenkungsausschuss kann man als oberste Instanz im Projektgeschehen Ansehen.

- muss bei größeren Projekten vorhanden sein.
- besteht aus Abteilungsleitern und/oder anderen Entscheidern (Schlüsselpersonen die Interesse am Erfolg des Projektes haben).
- trifft wichtige strategische Entscheidungen.
- hilft in kritischen Projektsituationen.
- wacht über den Projektfortschritt.

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Projektdokumente

Dokumentieren hilft Informationen nachzuvollziehen und Vorgehensschritte zurückzuverfolgen.
Unterscheidung der Dokumente:

vertragliche:	Angebot, Dienstleistungsvertrag, Rechnung, Abnahmeprotokolle.
fachliche:	Testplan, Lastenheft, Pflichtenheft, projektinterne Richtlinien, Benutzerhandbuch, technische Systemdokumentation.
PM-Dokumente:	Projektauftrag, Projektplan, Projektübersicht, Änderungsprotokoll, Projektstatusbericht.

Erinnerung

Lastenheft:
- wird vom Kunden erstellt.
- beinhaltet die Vorstellung über die Lösung aus der Kundensicht.
- je detaillierter der Einbezug von Projektbestandteilen desto geringer sind die Risiken.
- wird verwendet zum Abarbeiten der Anforderungspunkte (Checkliste).

Pflichtenheft:
- wird vom Dienstleister erstellt.
- wird an die Anforderungen des Lastenheftes angepasst.
- beinhaltet alle Anforderungen an die geplante Lösung. (Vollständigkeit)
- hat wohlüberlegte Inhalte (Widerspruchsfreiheit).
- muss im Bezug auf die Anforderungen verständlich für Kunden und Dienstleister sein (Allgemeinverständlichkeit).
- Fachbegriffe werden vermieden oder verdeutlicht (Präzision).
- ist eine Checkliste zur Abnahme des Kunden.

PM-Dokumente: Projektauftrag

Der Projektauftrag beschreibt, um was es bei dem Projekt im Kern geht:

- Projektname, **Ziele**, Rahmenbedingungen, Risikofaktoren, Meilensteine, Aufwand und Zeitplan, Abnahmekriterien, rechtliche Aussagen, Organisation, sämtliche Kosten.
- Abgleich mit der SMART-Regel

PM-Dokumente: Projektübersicht

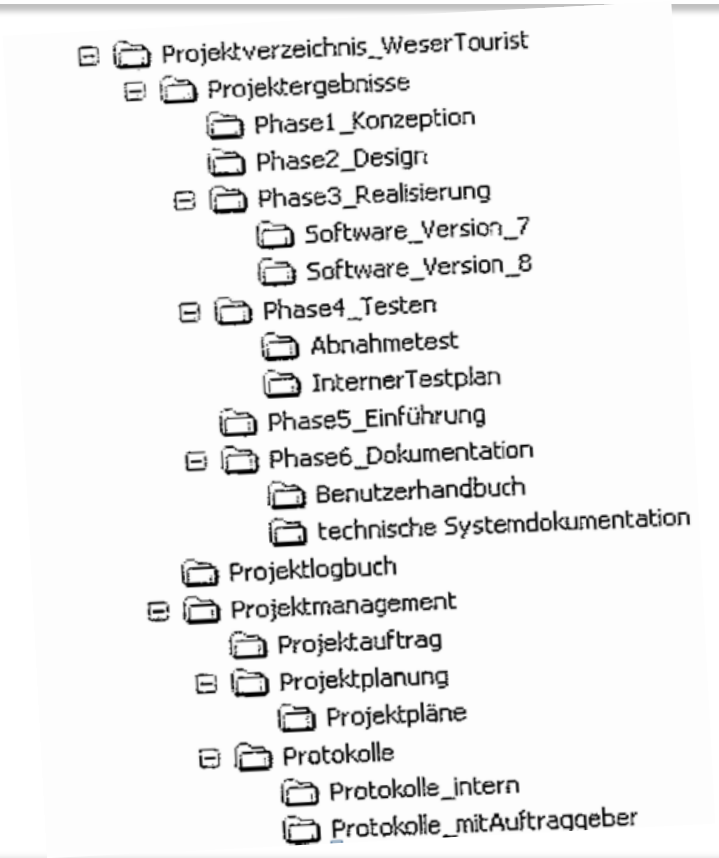
stellt kurz und knapp wesentliche Inhalte aus dem Projektauftrag und dem Projektplan dar.

Projektübersicht	*Projekt: Webshop XXL*
Auftraggeber	*Auftragnehmer*
MBZL GmbH	SWHL GmbH
Projektleiter Auftraggeber	*Projektleiterin Auftragnehmer*
Herr Mommert	Frau Beitinger
Projektziele	- Aufbau Internetpräsenz mit Bestellwesen - Anbindung an bestehendes Kundenverwaltungssystem
Projektauf-wand	Ca. 50 Personentage (PT)
Abrechnungsform	Nach Aufwand

Phase	*Ergebnis*	*Aufwand*	*von*	*bis*	*wer*
Konzeption	Pflichtenheft	8 PT	02.02.	06.02.	Team 1
Design	Systementwurf	7 PT	08.02.	13.02.	Team 2
Realisierung Internetpräsenz	DB läuft	10 PT	16.02.	06.03.	Team 2
Realisierung Schnittstelle	Schnittstelle erprobt	12 PT	08.02.	06.03.	Team 1
Testen	Testplan erledigt; Abnahme	6 PT	09.03.	20.03.	Team 1 + 2

Quelle: Brandt-Pook; Softwareentwicklung kompakt und verständlich

PM-Dokumente: Projektverzeichnis



senkt den Verwaltungsaufwand

Bildung von Hauptkategorien sinnvoll.

Quelle: Brandt-Pook; Softwareentwicklung kompakt und verständlich

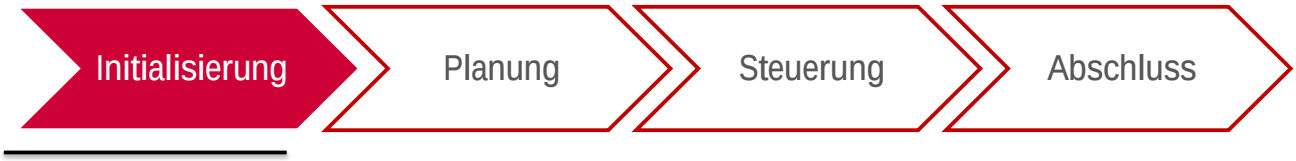
Projektmanagement

Definition: „Projektmanagement bezeichnet die Gesamtheit von Führungsaufgaben, Führungsorganisation, Führungstechniken und Führungsmitteln für die Abwicklung eines Projektes.“

Inhalt:

- planen,
- organisieren und
- kontrollieren von Projekten.

Phasen im Projektmanagement



In jeder Phase werden bestimmte Prozesse und Aktivitäten durchlaufen.

Initialisierung:

- Geburtsstunde des Projekts
- Vorstellung beim Kunden und beim Auftraggeber
- Ziel: Gemeinsames Verständnis von Auftraggeber und Auftragnehmer über die Eckpunkte und wichtigen Rahmenbedingungen des Projektes
- Ergebnis: Das zentrale Ergebnis der Projektinitialisierung ist der Projektauftrag

Phasen im Projektmanagement (2)

Planung:

beinhaltet:

- Die Projektziele
- Angaben zu den wichtigsten Meilensteinen
- Der geplante Aufwand
- Verantwortlichkeiten
- Angaben zum geplanten Vorgehen im Projekt

Phasen im Projektmanagement (3)

Steuerung:

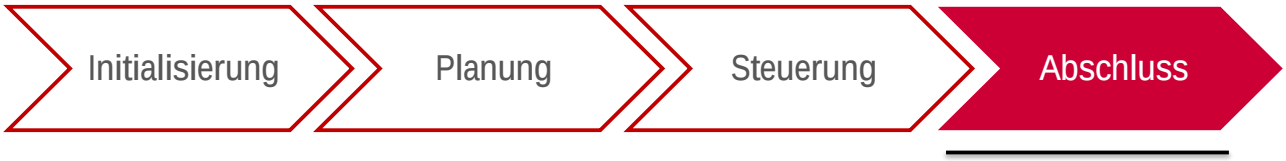
Voraussetzungen:

- Der Projektauftrag und der Projektplan liegen vor.
- Das Projektteam ist zusammengestellt und über das Projekt informiert.
- Die Infrastruktur für das Projektteam ist gegeben

Ziele:

- Geeignete Vorgaben an den Kernbereich des Projekts zu machen
- Messwerte aus dem Kernbereich des Projektes aufzunehmen
- Bedarfsgerecht die Planung anzupassen
- Überarbeitete Vorgaben für den Kernbereich des Projektes zu liefern.

Phasen im Projektmanagement (4)



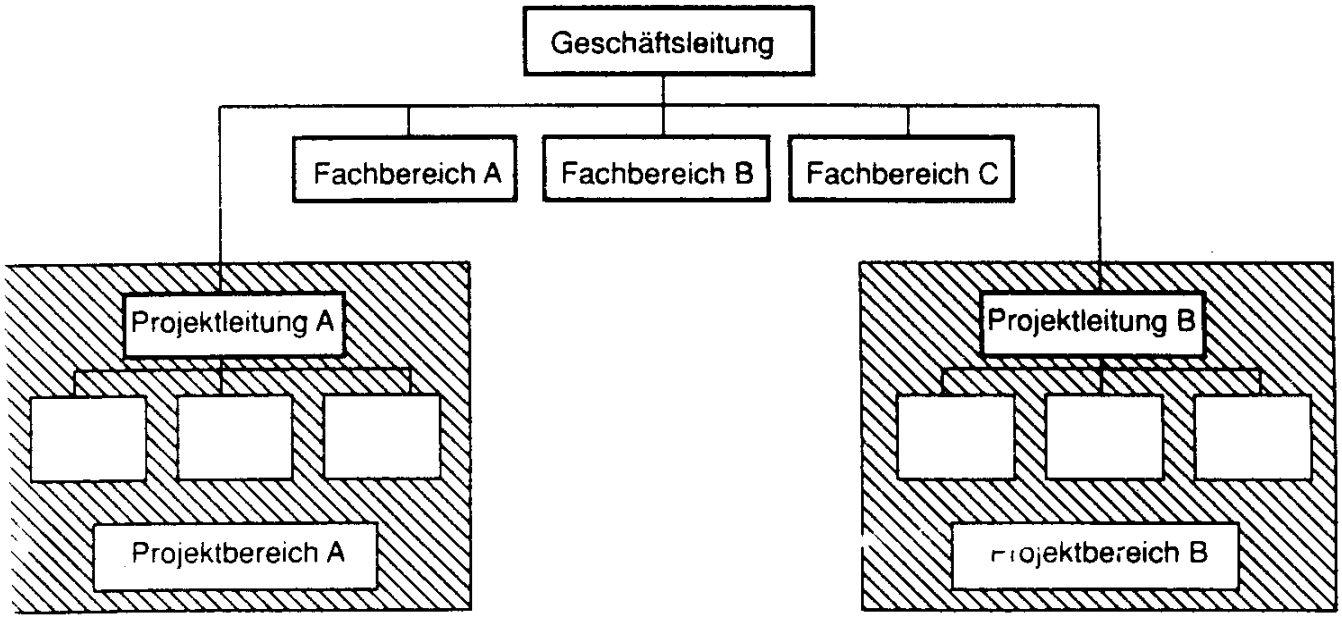
Abschluss:

Die Voraussetzungen für den Projektabschluss sind, dass die Projektziele erreicht wurden und der Kunde die IT-Lösung abgenommen hat.

- Abschlusssitzung
- Archivierung
- Weitere Zusammenarbeit
- Ergebnisse übertragen
- Team auflösen

Projektorganisation

Reine Projektorganisation

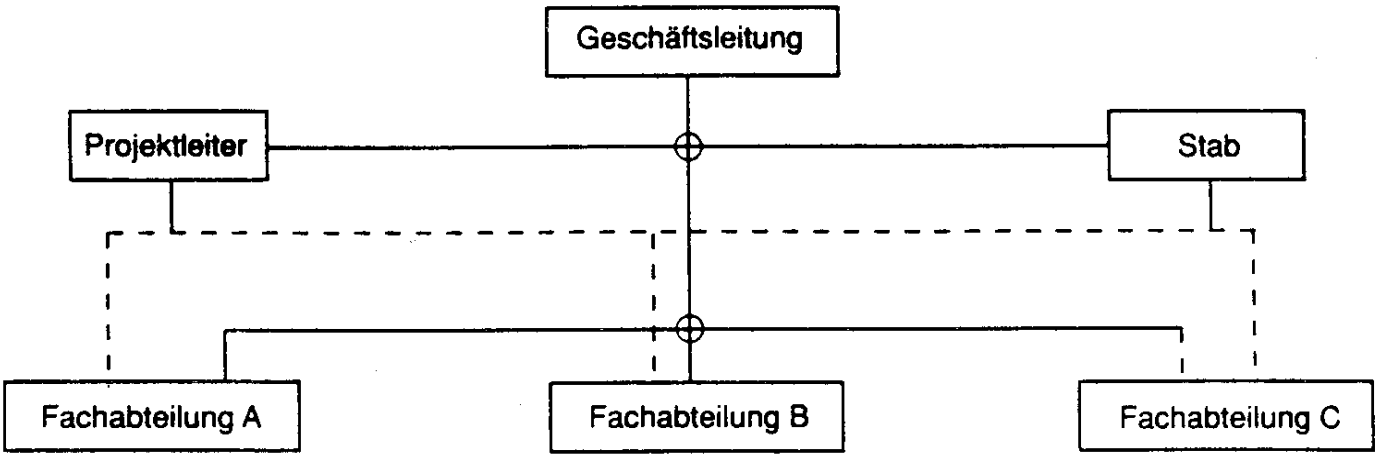


Vorteile:
- volle Konzentration der Beteiligten auf die Projektziele
- gute Identifikation mit dem Projekt
- durch eindeutige Weisungsbefugnis ist die Möglichkeit gegeben zur straffen Führung und zu schneller Reaktion auf Störungen

Nachteile:
- Bereitstellung der erforderlichen Ressourcen
- Auflösung der temporären Organisation am Projektende
- Einsatz von Spezialisten, die nur zeitweise benötigt werden

Projektorganisation (2)

Einfluss - Projektorganisation



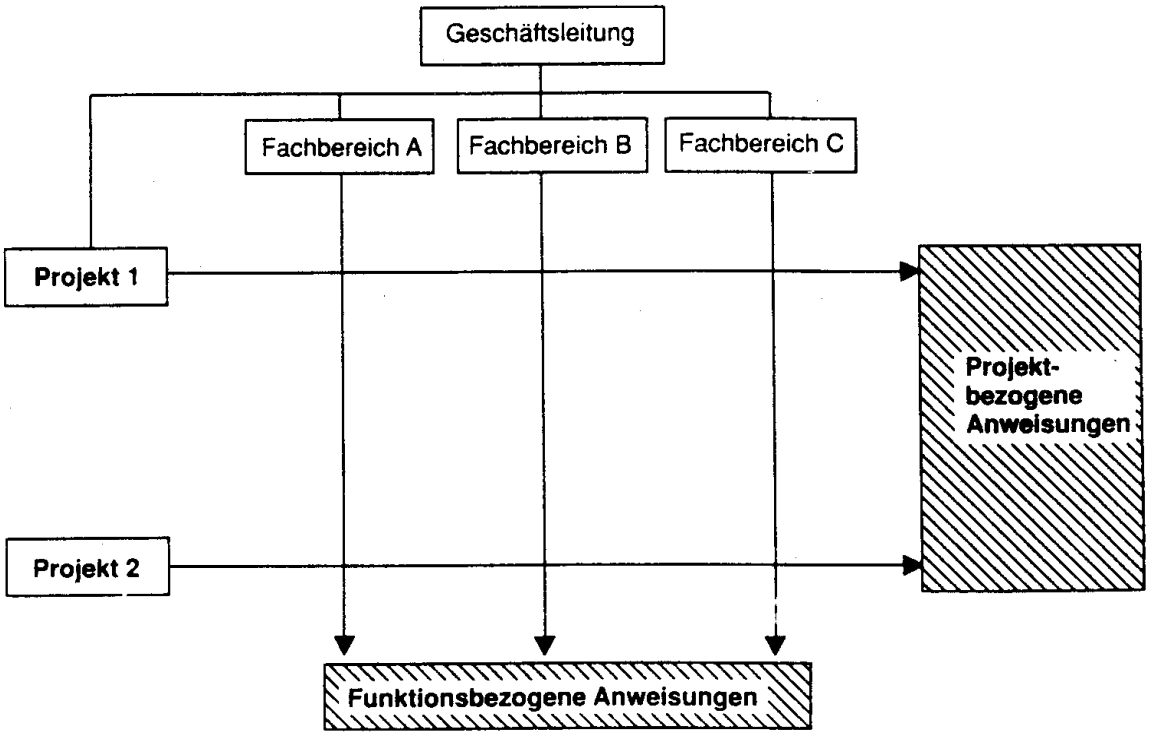
Die Primärorganisation bleibt unverändert bestehen, lediglich ergänzt durch eine Stabstelle.
Der Projektleiter befindet sich im Stab und hat gegenüber den Fachabteilungen nur Informations- und Beratungsbefugnis ohne irgendeine Exekutive

Vorteile:
- hohes Maß an Flexibilität hinsichtlich des Personaleinsatzes
- keine organisatorische Umstellung erforderlich
- Sammlung von Erfahrung und deren Austausch über verschiedene Projekte ist relativ einfach

Nachteile:
- niemand fühlt sich für das Projekt verantwortlich
- Reaktionsgeschwindigkeit bei Störungen ist kleiner
- es bestehen Probleme, Schwierigkeiten über die Abteilungsgrenzen hinweg gemeinsam zu lösen

Projektorganisation (2)

Matrix- Projektorganisation



Jede Organisationseinheit wird zwei Instanzen unterstellt (1. der Fachabteilung, 2. dem Projektleiter)

Vorteile:
- Der Projektleiter fühlt sich für sein Projekt verantwortlich
- Es ist eine zielgerichtete Koordination verschiedener Interessen möglich
- Spezialwissen kann gezielt von einem Projekt zum anderen transferiert werden

Nachteile:
- die aufwendige Organisation
- Gefahr von Kompetenzkonflikten
- der große Personalaufwand für die Projektleitung

Projektmanagement: Hilfsmittel

Sehr häufig wird eine Projektmanagementsoftware eingesetzt, die insbesondere bei hoher Komplexität oder vielen Benutzern unterstützend wirkt.

Bekannte Beispiele sind:

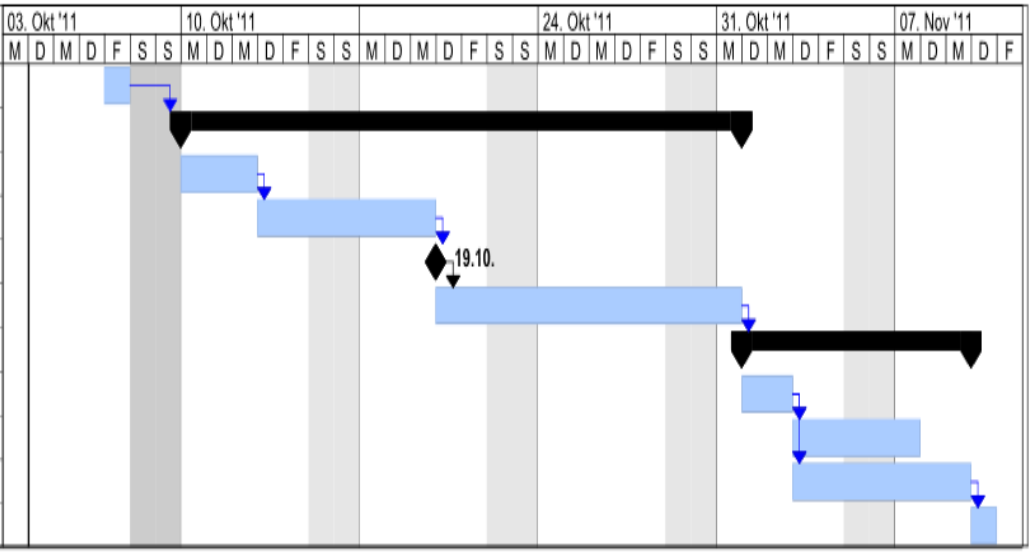
MS-Projekt
Redmine (Webbasiertes Open-Source PMS)
Wikis als Wissensmanagementsoftware

Ein beliebtes grafisches Element zur Unterstützung in Projekten ist der Gantt-Chart

Projektmanagement: Hilfsmittel

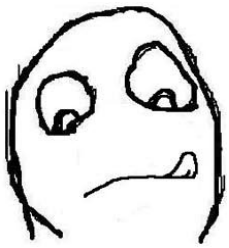
Gantt-Chart

Nr.	Vorgangsname	Dauer
1	Kick off	1 Tag
2	**Phase 1**	**16 Tage**
3	Aktivität 1	3 Tage
4	Aktivität 2	5 Tage
5	Meilenstein	0 Tage
6	Aktivität 3	8 Tage
7	**Phase 2**	**7 Tage**
8	Aktivität 4	2 Tage
9	Aktivität 5	3 Tage
10	Aktivität 6	5 Tage
11	Projektabschluss	1 Tag



Zusammenfassende Fragen: Abschnitt 2.2

- Welche Kriterien kennzeichnen ein Projekt?
- Welche Rollen gibt es in einem Projekt?
- Nennen und beschreiben Sie die Teambildungsphasen.
- Welche Phasen hat der Projektmanagementprozess?
- Welche Organisationsformen gibt es und wie sind sie gekennzeichnet?
- Was ist ein typisches Hilfsmittel im Projektmanagement?



Übungsaufgaben

1) Überlege, ob es sich bei den unten genannten Vorhaben um Projekte handelt. Nutze die Kriterien zur Charakterisierung von Projekten. Wenn es kein Projekt ist dann nenne das fehlende Kriterium.

- Umzug in eine andere Stadt
- Diskobesuch am Samstagabend
- Werbekampagne für ein Handy
- Flughafen BER
- Winterreifen kaufen und montieren
- Planung einer komplett neuen Küche incl. Raumumgestaltung
- Buchung eines Hotelzimmers
- Übernahme von Daten aus dem alten in ein neues IT-System

Übungsaufgaben

2) Was halten Sie von folgender Aussage: „Alle Projekte funktionieren nach denselben Regeln. Ob ein Haus oder ein Schiff gebaut, eine Standardsoftware in einer großen Firma eingeführt oder ein Wahlkampf organisiert wird: Für einen Projektleiter ist dies alles kein Problem“

3) Diskutieren Sie: Welche Regeln innerhalb eines Teams können dem Projekterfolg zuträglich sein?

Softwareengineering

1 Nutzwertanalyse 

2 IT-Projekte 

3 Vorgehensmodelle

4 Geschäftsprozessmanagement

5 UML Diagramme

Quellen u.a:

Brandt-Pook, H.; Kollmeier, R.: Softwareentwicklung kompakt und verständlich; Wiesbaden 2008

Pichler; Scrum; 2007

http://winfwiki.wi-fom.de/index.php/Qualit%C3%A4tssicherung_im_Softwareentwicklungsprozess_der_Software_Life_GmbH

Motivation

- Systeme werden immer komplexer
- Entwicklung wird komplizierter
- Wirtschaftliche Bedeutung von Software steigt
- Wartung wird schwieriger
- Schätzungen werden unübersichtlich
- Überblick geht verloren
- immer kürzere Zeiträume
- immer mehr Anforderungen
- Änderungsmanagement
- Sicherstellung eines hohen Qualitätsniveaus
- Wartung und Weiterentwicklung von Altsystemen
- Guter Programmierstil und Entwicklungswerkzeuge

→ Software entwickeln ist mehr als programmieren

Definition

„**Vorgehensmodelle** in der Systementwicklung und im Softwareengineering beschreiben **Folgen bzw. Bündel von Aktivitäten**, die zur Durchführung eines (IT)-Projekts erforderlich sind. Üblich ist eine **Gliederung in Phasen**, Arbeitsabschnitte und Aktivitätenblöcke, durch die der Entwicklungsprozess in planbare und kontrollierbare Einheiten zerlegt wird“,

wobei man sich stark an den logischen und/oder chronologischen Reihenfolge der Einzelaktivitäten orientiert.

(Quelle: Enzyklopädie der Wirtschaftsinformatik).

Definition (2)

Kennzeichnen:

- *Phasen*: VM legt fest, wie Projekte gleicher Art ablaufen
  - Phasen fassen Tätigkeiten des Gesamtprozesses zusammen
- *Rollen*: VM benennt die Teilnehmer an Projekten und beschreibt ihre Aufgaben
  - Nicht Person XY, sondern die verallg. Benennung und Aufgabenbeschreibung
- *Methoden*: VM stellt Methoden zur Verfügung, die bei der Bewältigung der Aufgaben benutzt werden
  - Methoden sind Verfahren und Anweisungen zu deren Nutzung, um bestimmte Ergebnisse zu erzielen.

(Quelle: Enzyklopädie der Wirtschaftsinformatik).

Arten von Vorgehensmodellen (u.a.)

- Phasenorientierte VM
  - Basismodell
  - Wasserfall-Modell
  - V-Modell
- Iterative VM
  - Spiralmodell
- Agile VM
  - eXtreme Programming
  - SCRUM

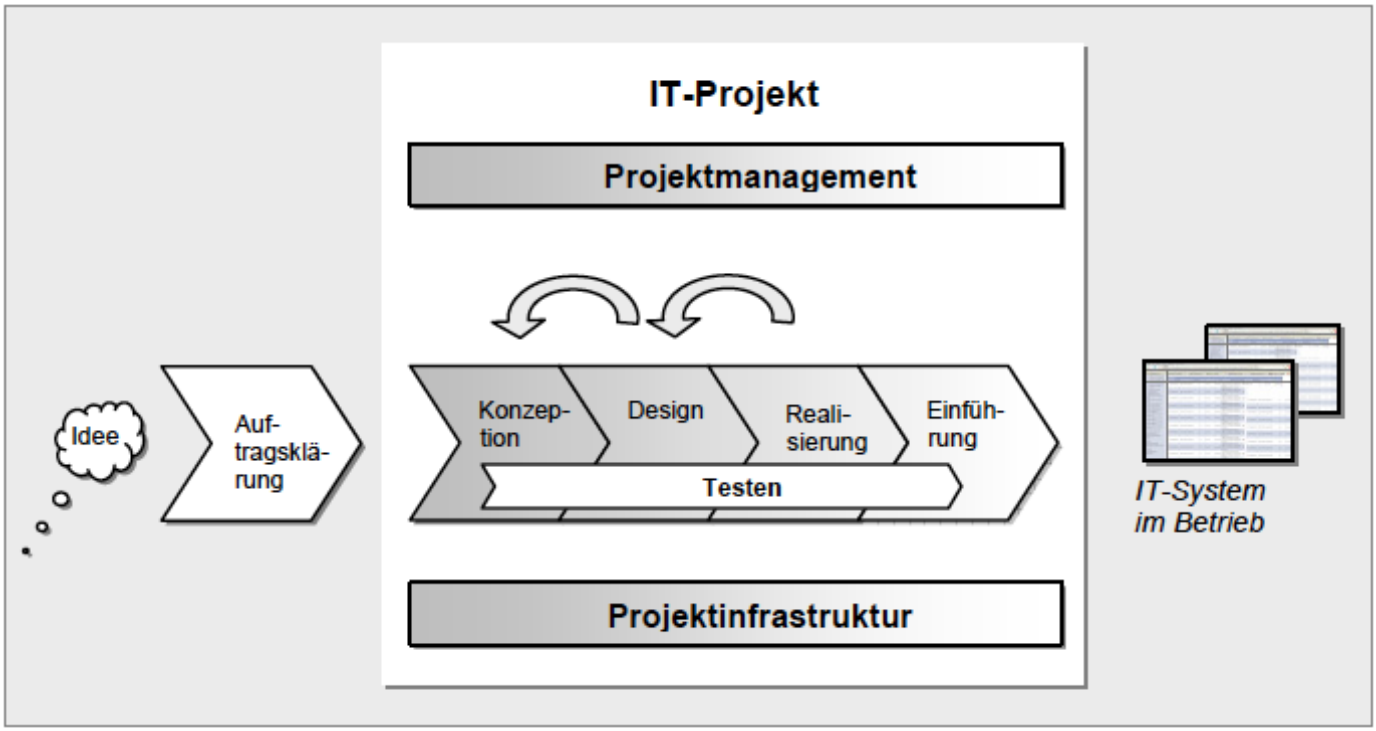
Phasenmodelle

- Merkmal:
  aufeinander folgendes Vorgehen mit klar definierten Phasen und Ergebnissen.

- Vorteile:
  - Einfach durchzuführen
  - schränkt Freiheitsgrade stark ein, daher auch für sehr große Projekte anwendbar
  - sehr effizient bei bekannten und konstanten Anforderungen
  - ist gut zu vermessen (notwendig z.B. für Prozessverbesserung)

- Nachteile:
  - Risiken gesammelt am Schluss („Big Bang“)

Phasenmodelle – Basismodell



Quelle: Brandt-Pook; Softwareentwicklung kompakt und verständlich

Phasenmodelle – Basismodell

1. Phase: Auftragserklärung

- Ziele:
  - Ist die Umsetzung der Idee aus der Sicht des Geschäfts sinnvoll? (Wirtschaftlichkeit; Kosten; Geschäftsanalyse)
  - Welche IT-Projekte zur Umsetzung der Idee anzuwenden sind (Machbarkeit; politische Interessen; Ausarbeitung der Idee).

Phasenmodelle – Basismodell

2. Phase Konzeption

Leistungsumfang des Systems – WAS soll das beabsichtigte IT-System leisten (Eigenschaften und Fähigkeiten des IT-Systems aus Zielen ableiten)

- Dazu 3 Perspektiven:
  - Welche funktionale Anforderungen gibt es an das System?
  - Welche Daten werden in dem System gehalten?
  - Welche wichtigen Kennzahlen und weitere Anforderungen sind zu beachten?
- **Lösungsalternativen** beschreiben und bearbeiten; Nutzwertanalyse (Bsp.: technische Basis der IT-Lösung (Rechner, Betriebssystem usw.)).

Phasenmodelle – Basismodell

3. Phase Design

„WIE soll das IT-System intern aufgebaut und gestaltet sein?“

- Aufgaben:
  - Die Hard- und Softwarearchitektur zu klären
  - Einzelne Komponenten innerhalb der Softwarearchitektur zu spezifizieren
  - Transformation der Komponenten in die geplante IT-Landschaft zu planen.

Phasenmodelle – Basismodell

4. Phase Realisierung (Implementation)

„Build it!!!“

- Ziele:
  - Das IT-System so zu erstellen, dass alle Anforderungen erfüllt sind und das IT-System beim Kunden eingeführt werden kann.
  - Die Konzeption und das Design in einer Programmiersprache zu implementieren.

5. Phase Einführung (Roll-Out)

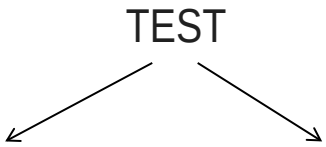
„Go live!!!“

- Aufgaben:
  - Die Abnahme des IT-Systems;
  - Die Überführung in den produktiven Betrieb (Big Bang; Step-by-Step);
  - Die Schulung der Anwender (KeyUser).

Phasenmodelle – Basismodell

6. Phase Testen

- Aufgabe:
  mögliche Fehler in einem IT-System zu finden.
  Wichtig: in jeder Phase zu testen!



Fachlicher Test (Business-Test)

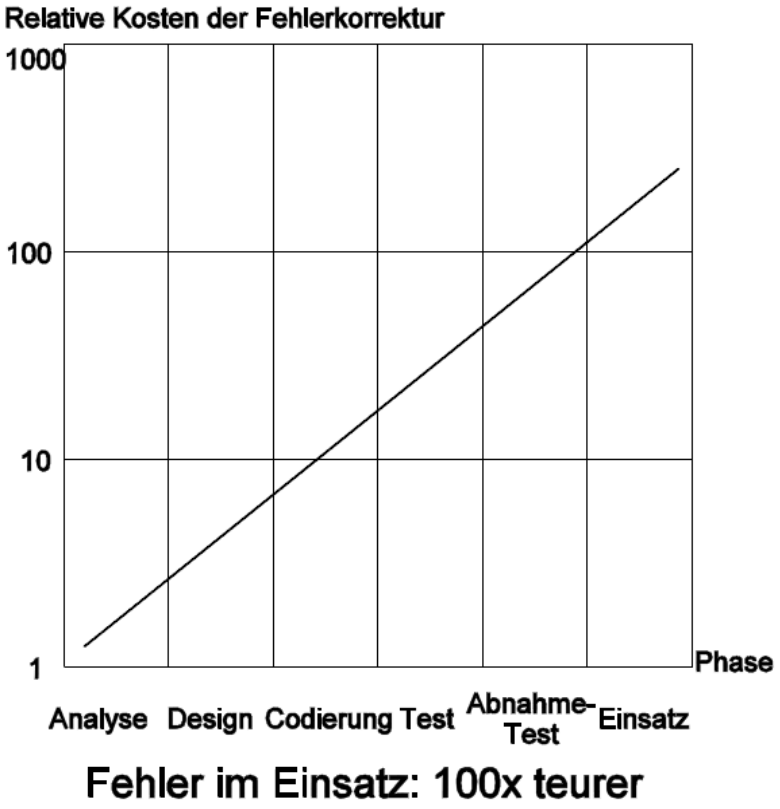
Technischer Test (IT-Systemtest)

- Das Testen des IT-Systems läuft parallel und wird ständig durchgeführt.
- Durch Testen lassen sich Fehler aus dem System „fischen“.
- Ein Nachweis, dass ein System fehlerlos ist, ist nicht möglich.

Phasenmodelle – Wasserfallmodell (Royce 1970)

Phasenmodelle – Wasserfallmodell (Royce 1970)

Korrekturkosten nach Boehm



Fehler im Einsatz: 100x teurer

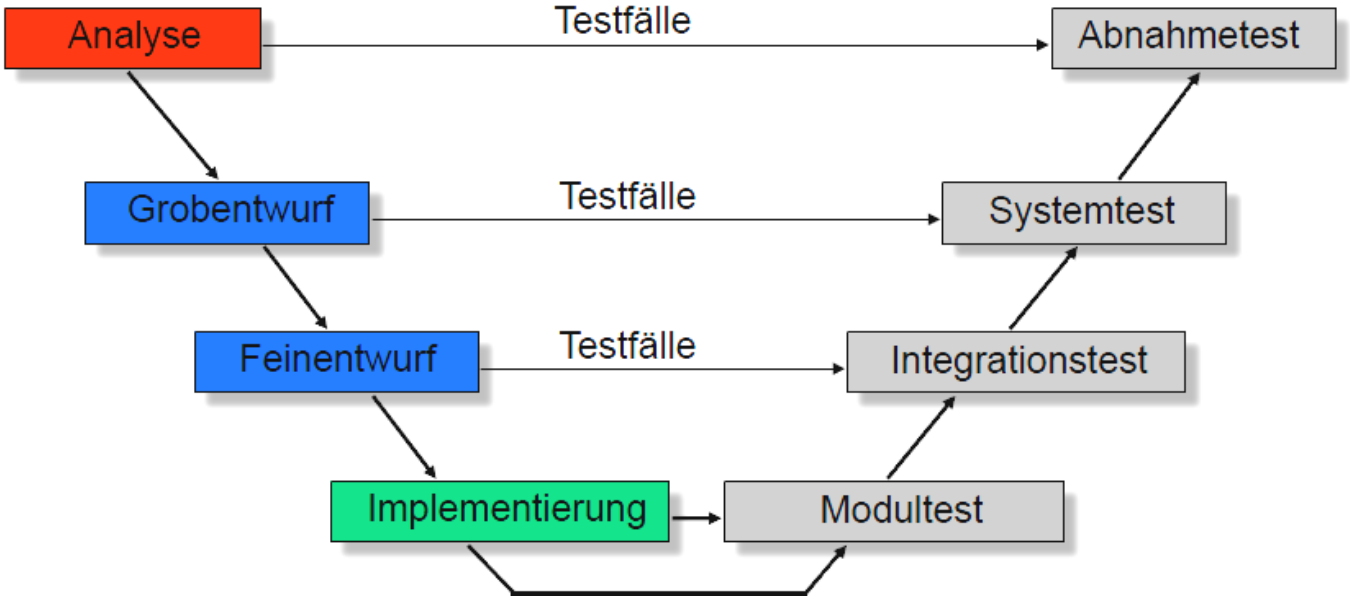
Phasenmodelle – Wasserfallmodell

Merkmale:

- Phasenmodell; die Phasen „Inbetriebnahme“ und „Wartung“ beschreiben das IT-System im produktiven Betrieb
- Die Phasen folgen strikt nacheinander. Eine Phase muss erst vollständig abgeschlossen sein, bevor die nächste Phase beginnt
- Ein einfaches und erfolgreiches Modell. Aber in komplizierteren Projekten ist es schwer anzuwenden
- Behandlung des Kernbereiches (ohne Projektinfrastruktur und Projektmanagement)
- Das Modell sagt nichts über Rollen und Methoden
- Notwendige „Kurskorrekturen“ nicht frühzeitig erkennbar

Phasenmodelle – V-Modell (Boehm 1979)

- Idee: Jede Phase wird überprüft und getestet.
- Erweiterung des Wasserfallmodells



Phasenmodelle – V-Modell (1986)

Merkmale:

- Zu jeder Aktivität sind Ergebnisse (= Produkte) definiert, die erbracht werden sollen
- Die Verantwortung für die Produkte tragen definierte Rollen
- Die Anforderungen an die Produkte sind genau festgelegt
- Für große Projekte geeignet
- „Das Vorgehensmodell des Bundes“

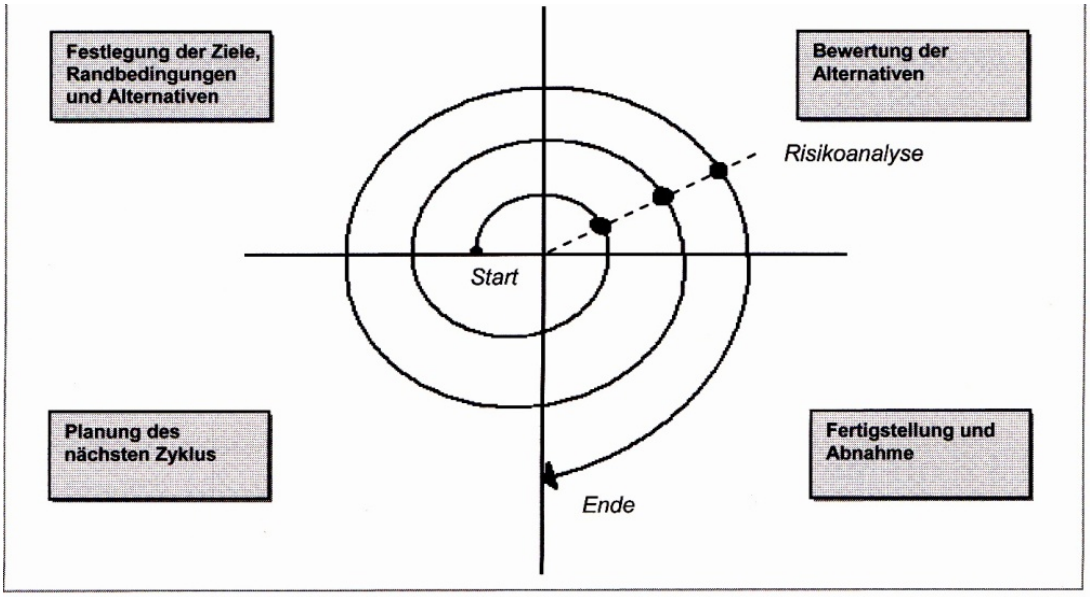
Iterative Modelle

- Merkmal:
  Iterativ heißt, dass sich der Entwicklungsprozess mehrfach wiederholt: statt den „Wasserfall“ einmal zu durchlaufen, werden kleine Wasserfälle hintereinander gesetzt.

- Vorteile:
  - Risiken können früher erkannt werden
  - Schwankende Anforderungen können besser berücksichtigt werden
  - Aufeinander aufbauende Auslieferung wird erleichtert.
- Nachteile:
  - Mehrarbeit
  - komplexeres Projektmanagement
  - schwerer messbar.

Iterative Modelle – Spiralmodell

- Idee: Immer wieder gleichartige Phasen durchlaufen während des Projektfortschritts und in jeder Runde wird eine Risikoanalyse ausgeführt.



Quelle: Brandt-Pook; Softwareentwicklung kompakt und verständlich

Iterative Modelle – Spiralmodell

- Ablauf:
  - Beginn mit der Festlegung der Ziele (grobe Gesamtziele), der Rahmenbedingungen und der Identifikation der möglichen Alternativen
  - Nähere Bewertung und Ausgestaltung der Alternativen; Entscheidung für Alternative
  - Umsetzung, Test und Abnahme bester Alternative
  - Planung nächster Runde

- Vorteil:
  - Risikobetrachtung als festgelegte Aufgabe in jeder Phase
  - Chance zur Korrektur!

Agile Modelle

Agil = beweglich, dynamisch.

- Merkmal:
  - kontinuierliche Anpassung an Änderungen.
- Ziele:
  - Höhere Flexibilität als bei klassischen Modellen
  - Software wird in regelmäßigen, kurzen Abständen dem Kunden präsentiert
  - Flexible Reaktion auf Kundenwünsche
  - Fokussierung auf die zu erreichenden Ziele
  - Angehen von technische **und** sozialen Problemen
  - nicht schwergewichtig und bürokratisch vorgehen
  - Fail fast - fail cheap - fail early

Manifest nach Beck 2001 (Auszug)

- Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge.
- Funktionierende Programme sind wichtiger als ausführliche Dokumentation.
- Die stetige Abstimmung mit dem Kunden ist wichtiger als die ursprüngliche Leistungsbeschreibung in Verträgen.
- Der Mut und die Offenheit für Änderungen stehen über dem Befolgen eines festgelegten Plans.

Agile Modelle (2)

- Vorteile:
  - Gut einsetzbar bei unklaren Zielen und sich ändernden Anforderungen/Umgebung
  - Verspricht besseres Kosten/Nutzen-Verhältnis
  - Bessere Code-Qualität
- Nachteile:
  - Ergebnis ist nicht vorhersagbar
  - Qualitätseigenschaften können nicht garantiert werden
  - Oft nicht nachvollziehbar, wie eine Funktion zustande kommt

Agile Modelle (3)

- Basiswerte:
  - Kommunikation (im IT-Team; zwischen IT-Team und Kunden)
  - Einfachheit
  - Feedback (vom Kunden)
  - Mut (mutig, eigenverantwortlich und umsichtig handeln)
  - → Wichtig ist gute Zusammenarbeit im Team!

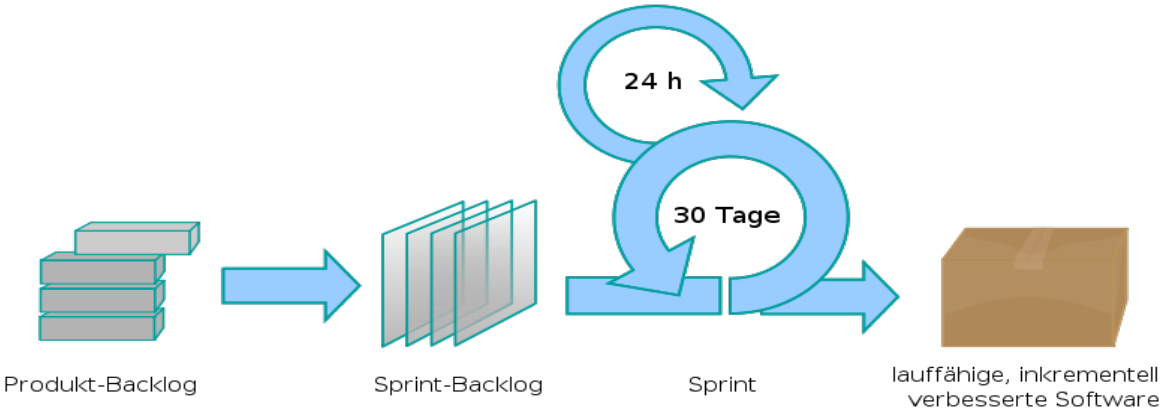
- Methoden und Techniken, u.a.:
  - Pair-Programming
  - Test first (Testen)
  - Kurze Entwicklungszyklen
  - Gemeinsamer Codebesitz u. Programmierstandards
  - Customer on-site (Kundenbeteiligung)

Agile Modelle – eXtreme Programming

Idee: schnelle und flexible Anpassung an Änderungen; Befreiung vom unnötigen Ballast und Konzentration auf die Programmierung.

Agile Modelle – SCRUM

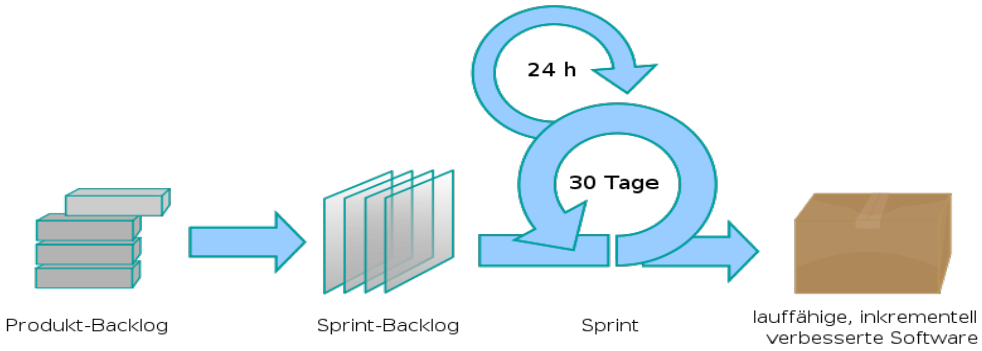
- Idee: die Selbstorganisation von Entwicklerteams und die eigenverantwortliche Auswahl der eingesetzten Mittel. Viele Ähnlichkeiten zu XP, etwas strukturierter



- Scrum-Projekte schreiten in Serien von Sprints iterativ voran
- Ein tägliches Treffen im Team (max. 15 Min.)

Täglich werden jedem Teammitglied folgende Fragen gestellt:
- Bist du gestern mit dem fertig geworden, was du dir vorgenommen hast ?
- Was hast du heute vor ?
- Gibt es ein Problem das dich behindert ?
- Gibt es etwas das wir verbessern können ?

Agile Modelle – SCRUM

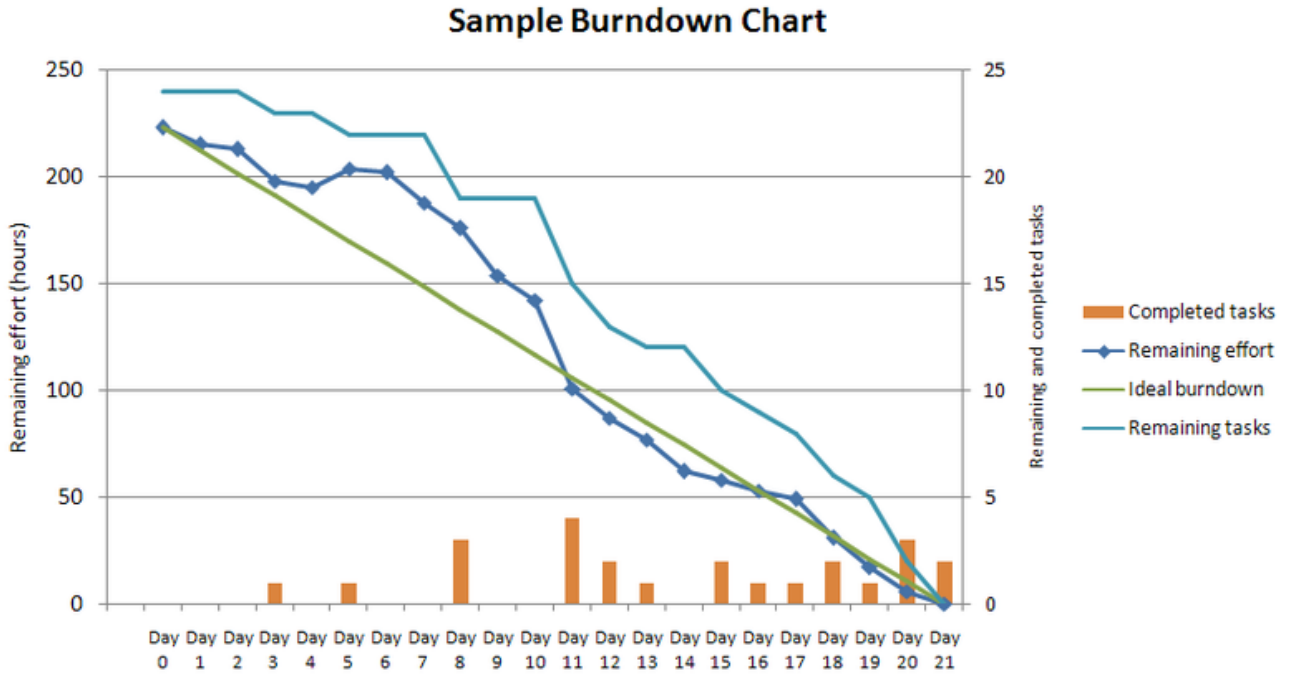


- Produkt-Backlog:
  Eine Liste aller gewünschten funktionalen und nicht-funktionalen Anforderungen

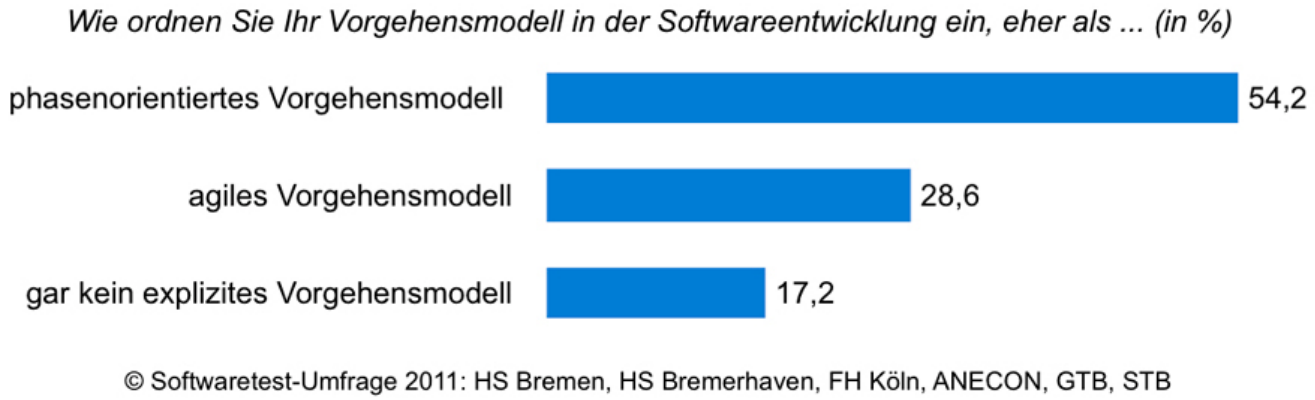
- Sprint-Backlog:
  - Team-Mitglieder wählen Tasks selbst aus, die Arbeit wird nie zugewiesen
  - Der geschätzte Restaufwand wird täglich aktualisiert
  - Jedes Team-Mitglied kann Tasks hinzufügen, löschen oder ändern

Agile Modelle – SCRUM

- Graphische Darstellung des Sprintfortschritts.
- Gegenüberstellung von Restaufwand und geleisteter Arbeit

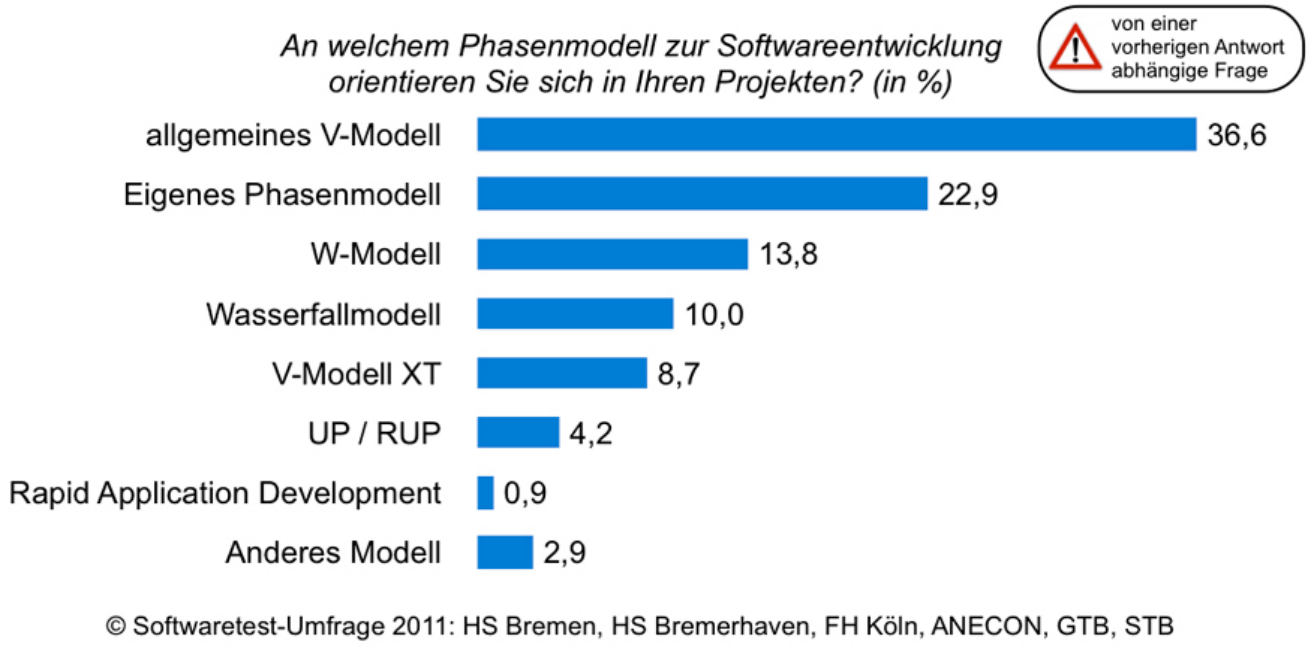


Einsatz im Unternehmen



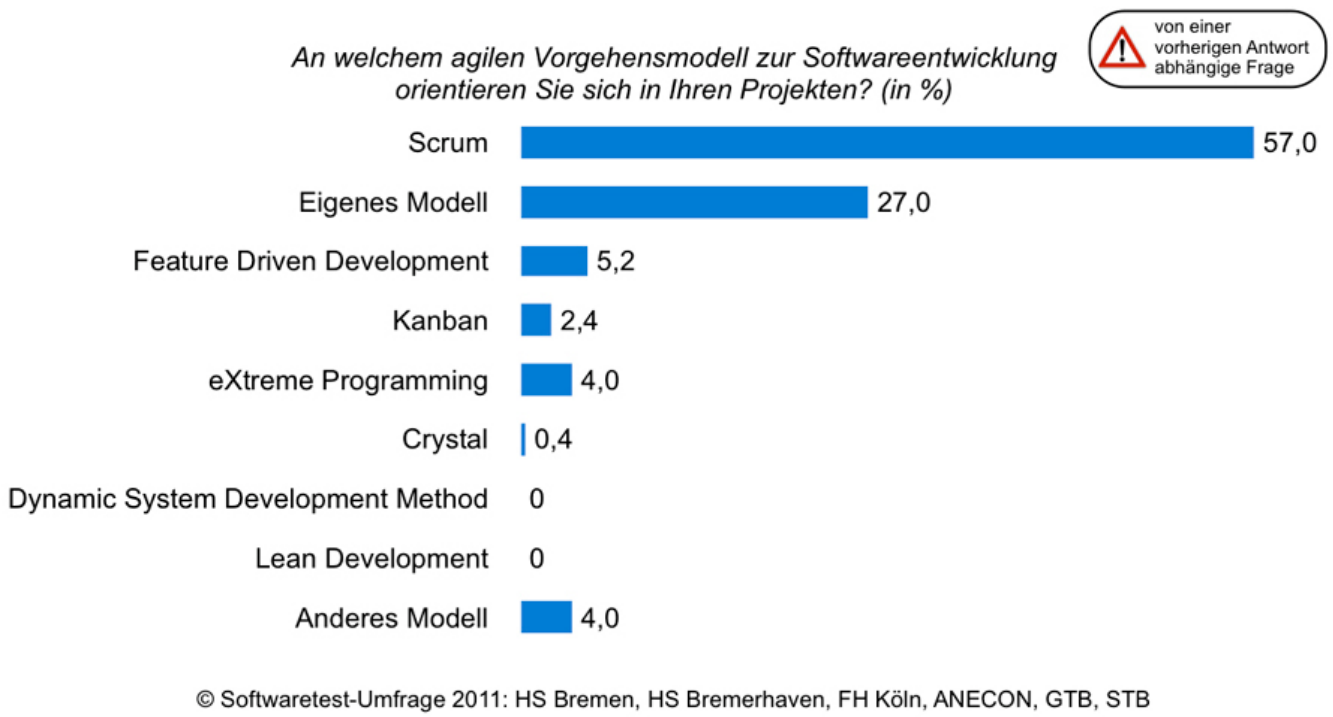
© Softwaretest-Umfrage 2011: HS Bremen, HS Bremerhaven, FH Köln, ANECON, GTB, STB

Einsatz im Unternehmen



© Softwaretest-Umfrage 2011: HS Bremen, HS Bremerhaven, FH Köln, ANECON, GTB, STB

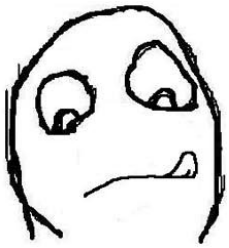
Einsatz im Unternehmen



© Softwaretest-Umfrage 2011: HS Bremen, HS Bremerhaven, FH Köln, ANECON, GTB, STB

Zusammenfassende Fragen: Abschnitt 2.3

- Was ist ein Vorgehensmodell und was ist die Motivation?
- Welche Arten Vorgehensmodellen gibt es?
- Was zeichnet die Arten aus? Merkmale, Vor- und Nachteile?
- Welche Phasen hat das Basismodell? Was zeichnet Phase xy aus?
- Was ist ein Sprint im SCRUM?
- Anhand welcher Kriterien könnte man ein Vorgehensmodell auswählen?



Übungsaufgaben

1) Wählen Sie für die folgenden Projekte jeweils ein Vorgehensmodell aus und begründen Sie Ihre Auswahl!

- a) Entwicklung eines einfachen Editors
  (Teamgröße 2 Entwickler, ca. 30.000 LOC, Lebensdauer 5 Jahre)
- b) Entwicklung eines Kfz-Steuergerätes
  (Teamgröße 5 Entwickler, ca. 30.000 LOC, Lebensdauer 10 Jahre)
- c) Entwicklung eines Buchhaltungssystems für eine Versicherungsgesellschaft
  (Teamgröße 50 Entwickler, ca. 400.000 LOC, Lebensdauer 25 Jahre)

Legende: LOC = Lines of Code

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Übungsaufgaben

	Wasserfallmodell	Spiralmodell	V-Modell	SCRUM
Projektgröße				
Benutzerbeteiligung				
Managementaufwand				
Anpassbarkeit				
zulässiger Änderungszeitpunkt				
Risikominimierung				
Qualität				
Entwicklungszeit				

Füllen Sie die obige Tabelle mit einer in 3 Ausprägungen skalierten Bewertung aus. (klein, mittel, groß / kurz, mittel, lang /.,.,.)
Arbeiten Sie im Team und diskutieren Sie!

Softwareengineering

1 Nutzwertanalyse 

2 IT-Projekte 

3 Vorgehensmodelle 

4 Geschäftsprozessmanagement

5 UML Diagramme

Lesen..Nachschlagen..Lernen..

Literatur zu Geschäftsprozessmanagement

- Freund, J.;
  Praxishandbuch BPMN 2.0;
  3. Aufl.; 2013

- Gadatsch, A..:
  Grundkurs Geschäftsprozessmanagement;
  7. Aufl.; Wiesbaden; 2012
  online via Springerlink

Definition und Motivation

- Def.: Betriebliche Informationssysteme
  „Betriebliche Informationssysteme dienen dazu, die in einem Unternehmen zur Erfüllung des Unternehmenszwecks anfallenden Aufgaben der Informationsverarbeitung zu erfüllen.“ (Dessaules, FH-Bielefeld)

- Motivation
  - In (fast) allen Unternehmen ist Informationstechnologie zu einem wesentlichen Bestandteil für erfolgreiche Geschäftsabwicklung geworden
  - Viele Unternehmen können ohne Informationstechnologie nicht mehr erfolgreich ihre Geschäfte abwickeln
  - Betriebliche Informationssysteme sind zu einem wichtigen Wettbewerbsfaktor von Unternehmen geworden
  - Unternehmen wenden viel Zeit und Kosten auf, um ihre betrieblichen Abläufe und Informationssysteme wettbewerbsfähig aufzustellen

Definition und Motivation

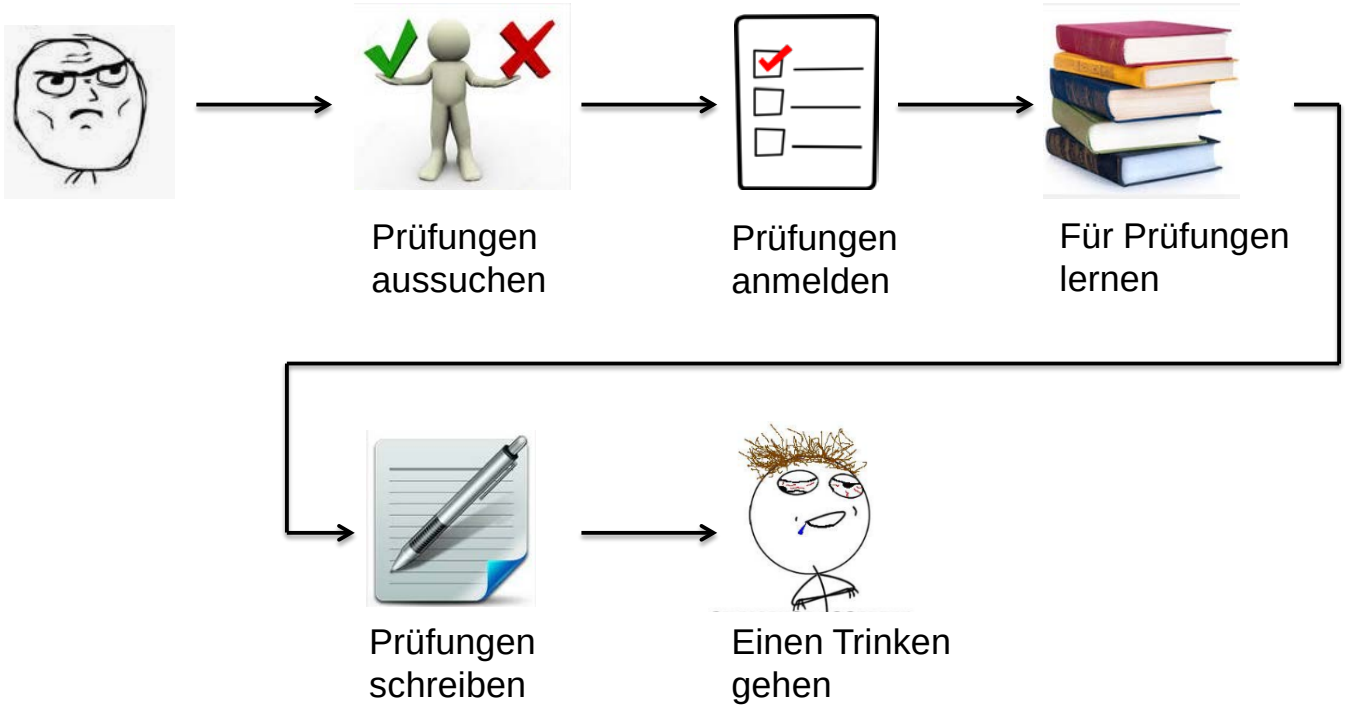
- Def.: Geschäftsprozess
  "Ein Geschäftsprozess ist eine Anordnung von **Aktivitäten** über **Zeit** und **Raum** mit definierten **Anfang** und **Ende** und definierten **Eingaben** und **Ausgaben**". (übersetzt nach Davenport 1993)

- Motivation
  - Viele Beanstandungen, Fehler, Änderungen
  - Hohe Produktkosten und Durchlaufzeiten
  - Unzureichende Liefertreue und -fähigkeit
  - Hohe Bestände
  - Geringe Flexibilität

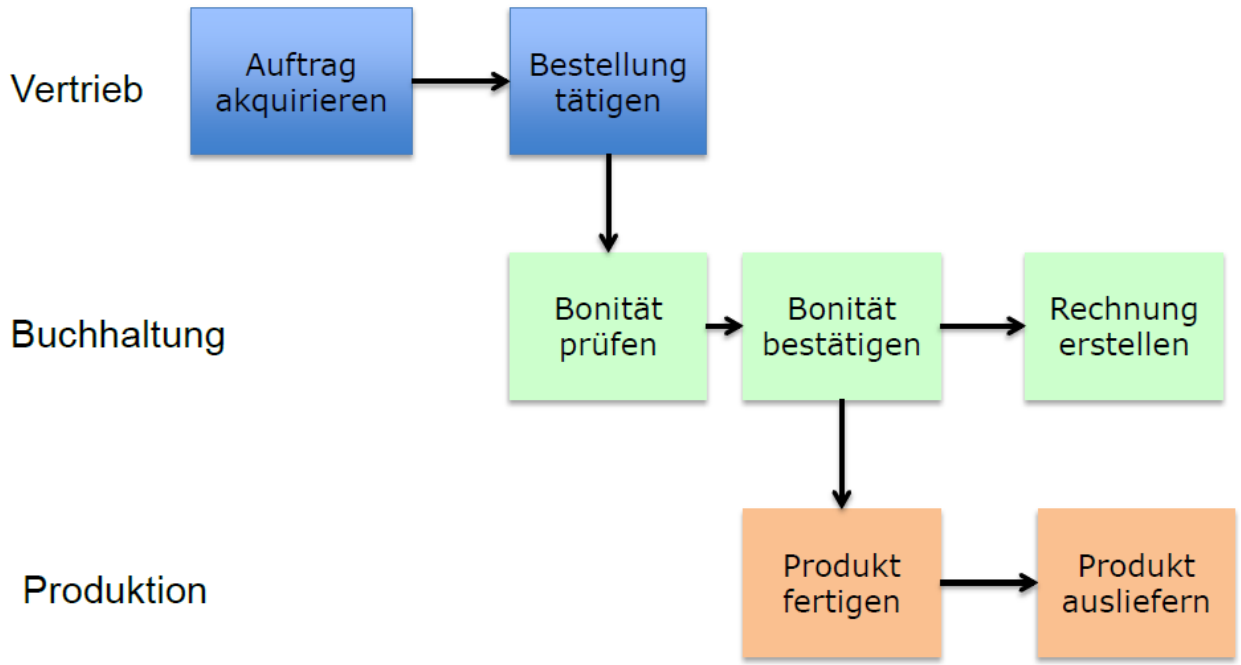
→ Um Prozesse systematisch in Unternehmen zu beherrschen, ist ein systematisches **Geschäftsprozessmanagement** erforderlich

Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Beispielprozess: Prüfung an der HS-OWL



Beispielprozess in einem ERP-System



Quelle: Laudon et al.; S. 97

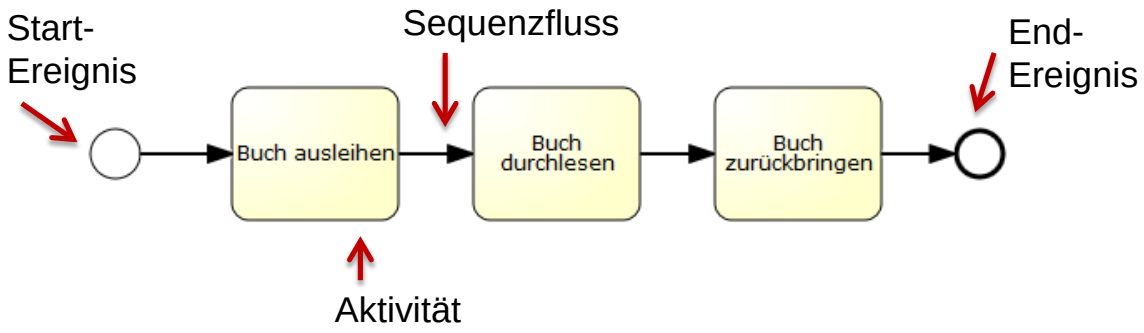
Geschäftsprozessmodellierung mit BPMN

- Motivation:
  - Dokumentation der Prozesse
  - Wiederholte Ausführung der Prozesse an Hand des Prozessmodells
  - Simulation und Analyse von Prozessen
  - Optimierung von Prozessen
  - Automatische und IT-unterstützte Ausführung von Prozessen
  - Wir benötigen eine **eindeutige** Beschreibung der Prozesse

- Ein Prozessmodell beschreibt die **wesentlichen Aspekte** eines Prozesses und lässt **unwesentliche Aspekte** aus

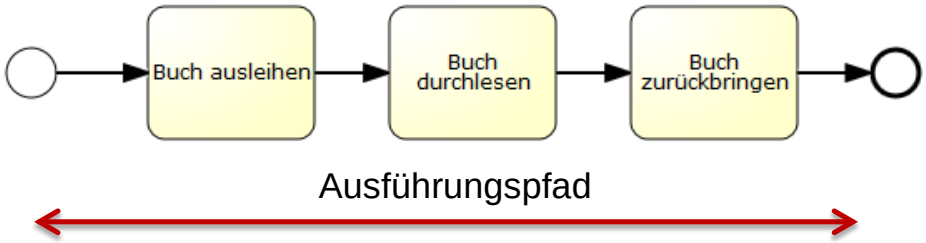
- Wesentliche Aspekte:
  Eingabe, Ausgabe, Aktivitäten, Ablauf, Ressourcen

Notation



- Aktivitäten: beschreiben Aufgaben, die im Rahmen des Prozesses ausgeführt werden
- Ereignisse: beschreiben, dass vor, während oder am Ende des Prozesses etwas passiert
- Sequenzfluss: beschreibt die zeitlich-logische Reihenfolge, in der andere Elemente zueinander stehen

Notation



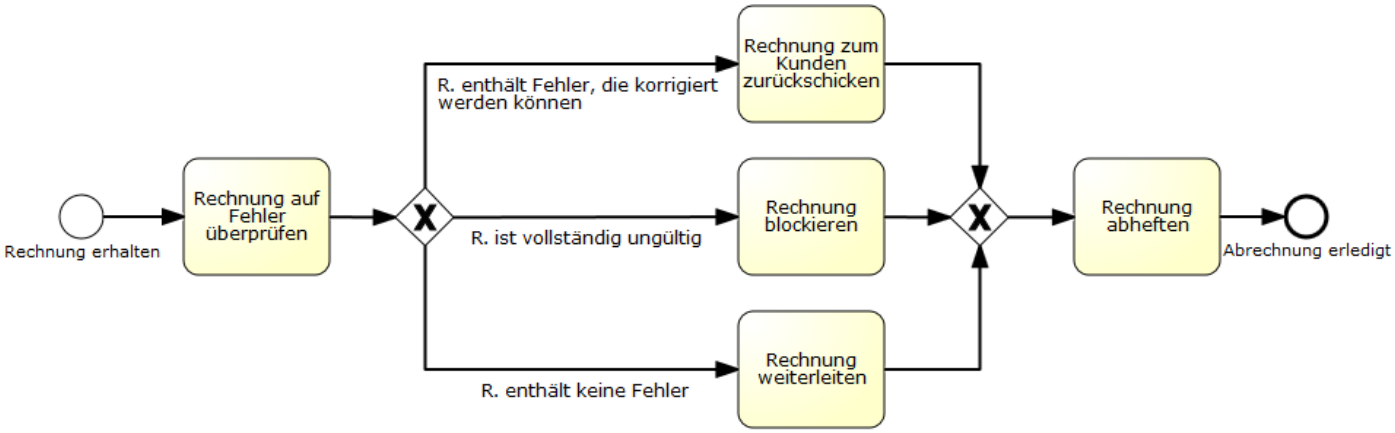
Ausführungspfad

- Ausführungspfade zeigen die Aktivitäten, die in einer bestimmten Prozessinstanz des Prozessmodells durchlaufen worden sind
- Ein Ausführungspfad kann als Sequenz dargestellt werden

  Notation: <A,B,C,D>
  Bsp.: <Buch ausleihen, Buch durchlesen, Buch zurückbringen>

Gateways: XOR-Split und XOR-Join

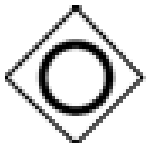
- Eine Entscheidung, bei der der Ausführungspfad in mehrere alleinstehende Ausführungspfade (**nur einer** kann ausgeführt werden) aufgeteilt wird, ist eine **E**xklusive Entscheidung (XOR-Split)

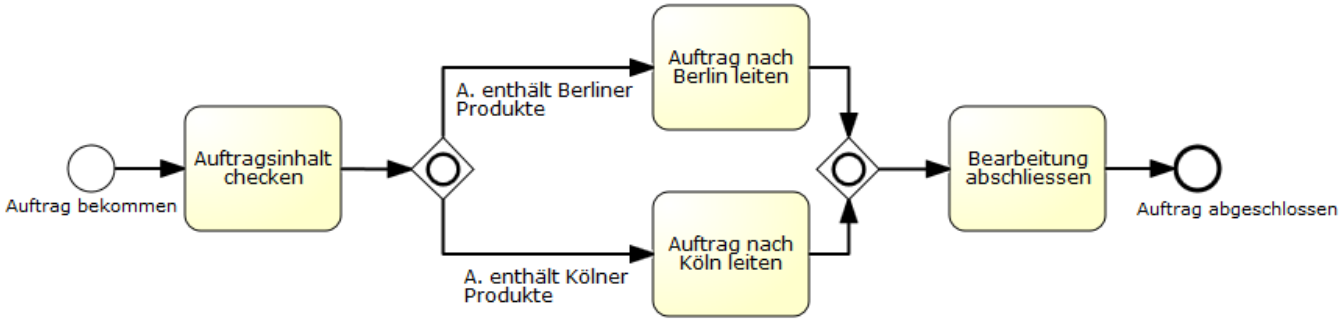


- Wenn in einem Prozess mehrere exklusive Ausführungspfade zusammengeführt werden, benutzt man einen XOR-Join.

Gateways: OR-Split und OR-Join

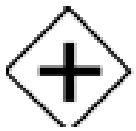
- Eine Entscheidung, bei der **eine oder mehrere** mögliche valide Alternativen **zur gleichen Zeit** ausgeführt werden können, wird als **OR-Split** modelliert.
- Die Zusammenführung wird durch einen **OR-Join** modelliert.



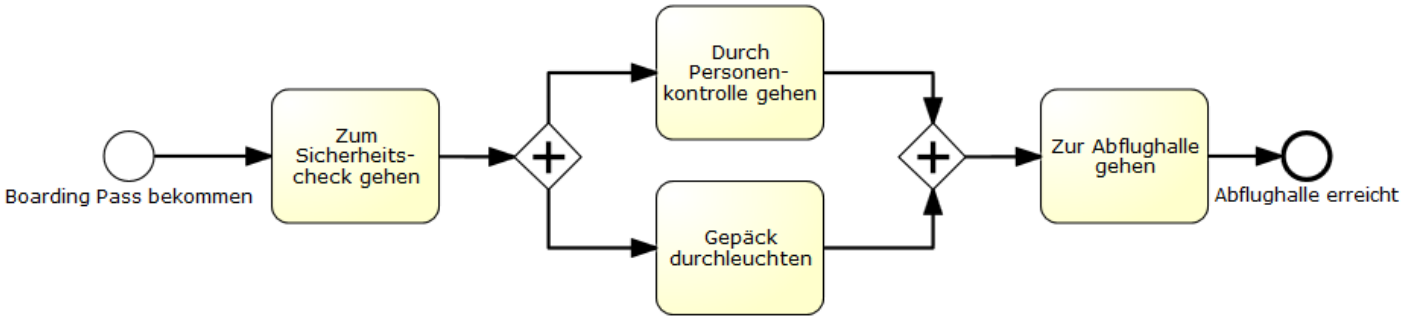


Gateways: AND-Split und AND-Join

- Wenn zwei oder mehr Aktivitäten **keine Abhängigkeiten** haben, können sie **parallel** ausgeführt und ein einzelner Ausführungspfad in mehrere aufgeteilt werden.



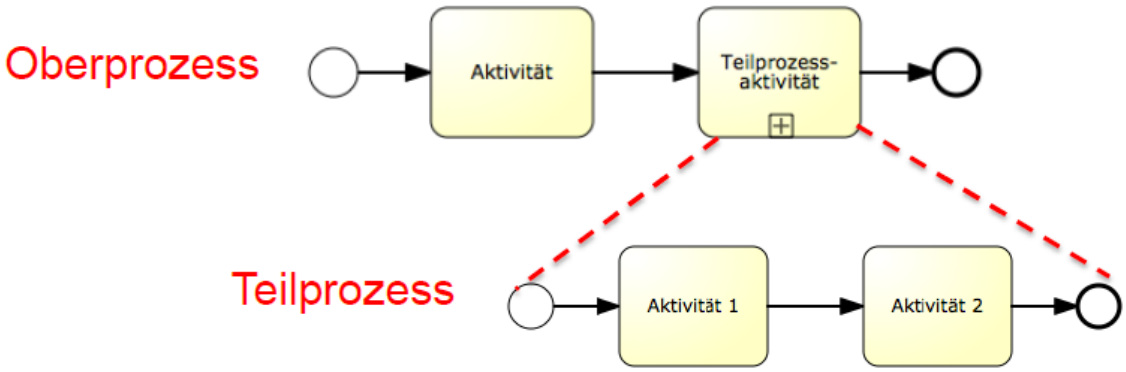
- Ein **AND-Join** wird dazu benutzt, mehrere durch **AND-Split** aufgeteilte Ausführungspfade wieder zusammenzuführen.



Teilprozesse

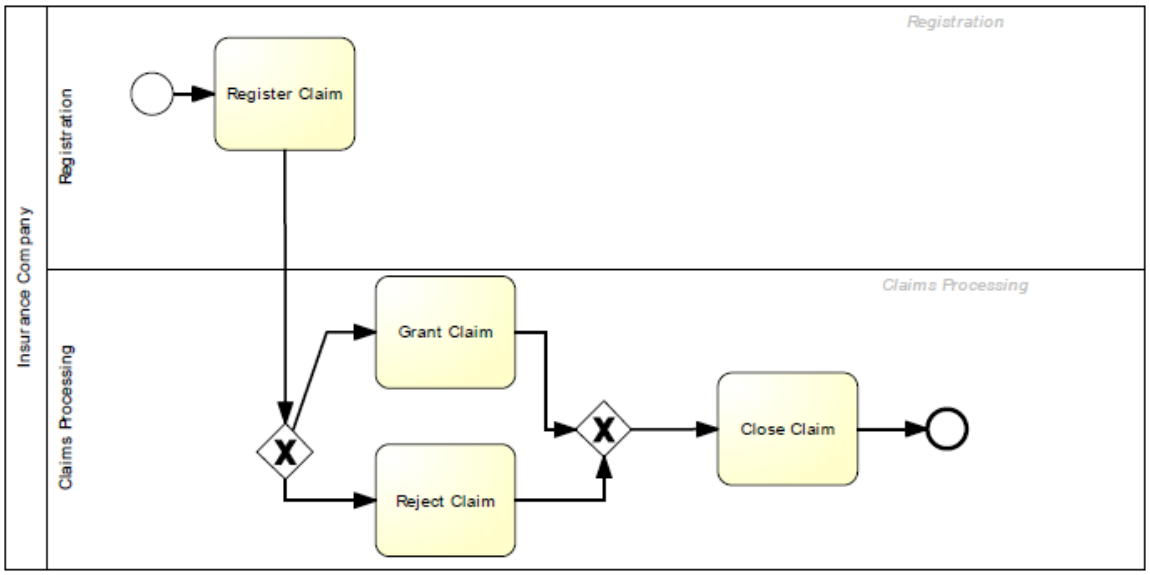
Wenn verwandte Teilaktivitäten das gleiche Ziel erreichen und/oder der Prozess zu groß wird

→ Bildung von Teilprozessen
(Subprozess / Unterprozess)



Lanes

- Eine Lane repräsentiert eine bestimmte Ressource (Person, Softwaresystem oder Hardwarekomponente)
- Lanes dienen dazu, in einem Prozess Aktivitäten dieser Ressource zu zuordnen. Sie können zudem verschachtelt sein



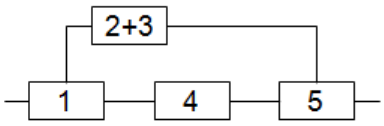
Prozessoptimierung

- Wichtige Einflussfaktoren für Prozessqualität
  - Durchlaufzeiten der Prozesse
  - Kosten, die Prozesse verursachen

- Ein Prozess verursacht bei seiner Ausführung Kosten durch
  - Arbeitseinsatz von Mitarbeitern, die eine Aktivität ausführen
  - IT Systeme, die zur Ausführung benötigt werden
  - Eine Aktivität an sich (z.B. Produktionskosten)

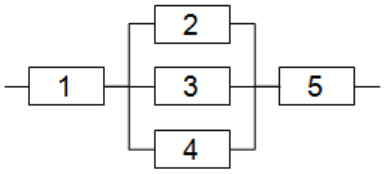
Prozessoptimierung

Zusammen-fassen



- **Zusammenlegung von Aktivitäten**

Paralleli-sieren

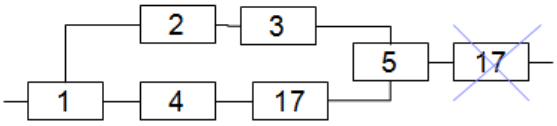


- **Erhöhung der Arbeitsteilung**

Beschleunigen

- **Bereitstellung von Arbeitsmitteln zur effizienten Aufgabenerledigung**
- **Vermeidung von Warte- und Liegezeiten**

Verlagern

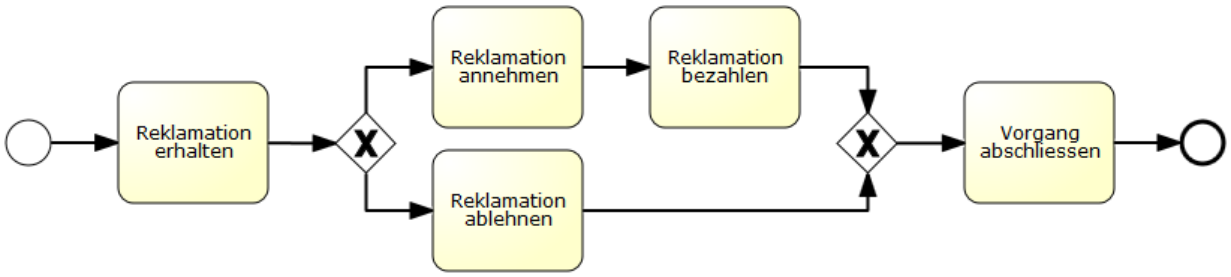


- **Früherer Beginn von bisher nachgelagerten Aktivitäten**

Quelle: Gadatsch, S. 21

Prozesssimulation

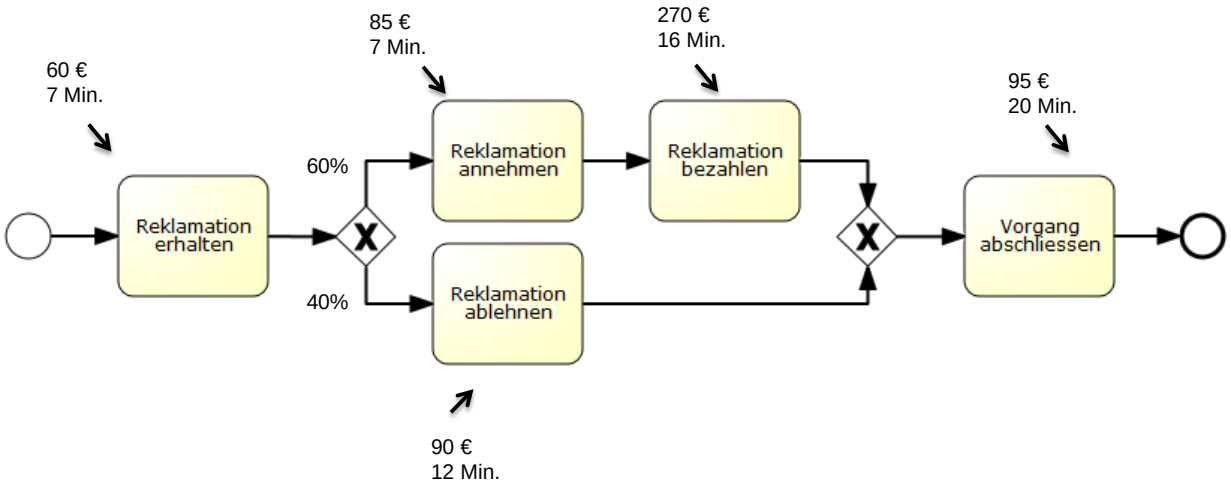
- Die Simulation von Prozessmodellen erlaubt es, Informationen über die Ausführung des Prozesses zu erhalten und zu analysieren
- Typische Fragen:
  - Was kostet die Prozessausführung im Mittel?
  - Wie lange dauert die Prozessausführung im Mittel?



Welche Daten werden von welchen Bestandteilen benötigt?

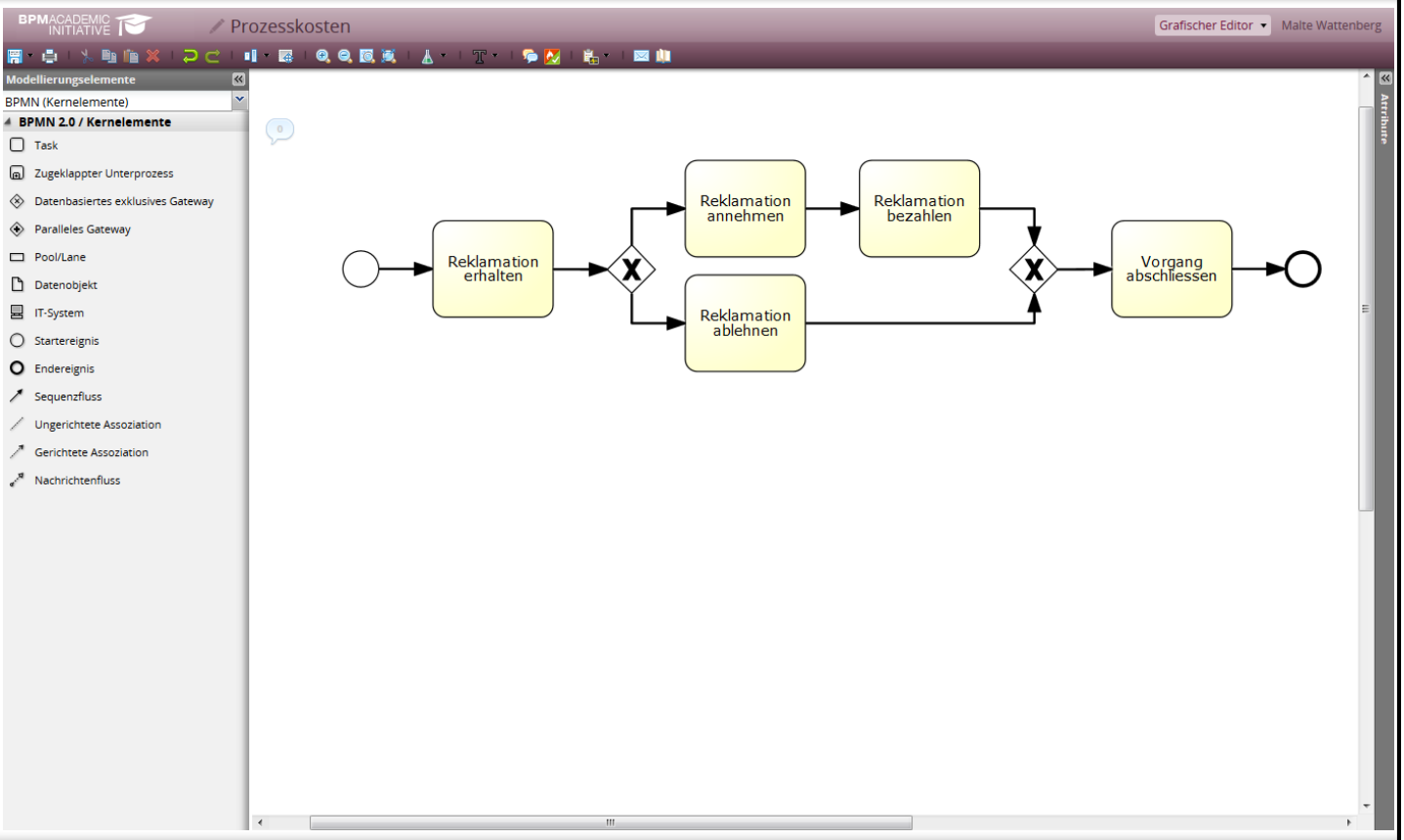
Beispielaufgabe

- Berechnen Sie die mittleren Kosten und Dauer der Prozessausführung!



Hochschule Ostwestfalen-Lippe
University of Applied Sciences

Werkzeuge: Signavio.com



academic.signavio.com
Eine nicht so umfangreiche Alternative ist (ohne Anmeldung) draw.io

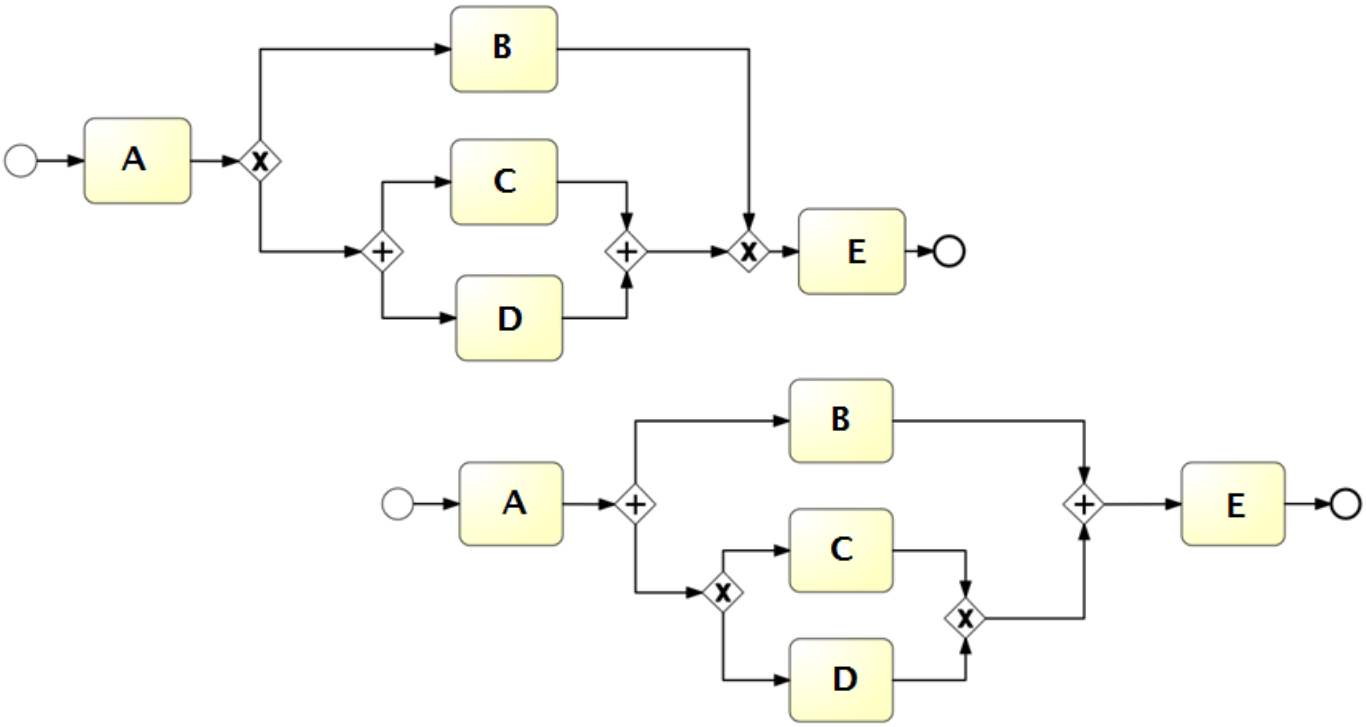
Übungsaufgaben

1. Modellieren Sie folgende Prozesse als BPMN (zunächst per Hand, danach in Signavio)

- Urlaubsantrag: Der Antragssteller füllt ein Formular aus. Der Urlaubsantrag wird an den Weisungsberechtigten weitergeleitet. Er prüft den Antrag. Wenn der Antrag in Ordnung ist wird der Antrag freigegeben, ansonsten zurückgewiesen. Der Antragsteller wird informiert.
- PC-Kauf: Ein Kunde möchte einen PC kaufen. Zunächst wählt er die Marke aus. Danach wählt er die Ausstattung. Anschließend hat er die Möglichkeit, eine Garantie dazu zukaufen. Sofern er das wünscht, kann er die Dauer auswählen. Zuletzt wird das Bargeld für den Kauf übergeben
- Wohnungsabnahme: Zunächst wird eine Mängelliste der Wohnung erstellt. Danach wird die Wohnung gereinigt und parallel dazu gelüftet. Sollten Mängel festgestellt worden sein, werden sie beseitigt. Zuletzt erfolgt die Schlüsselübergabe.

Übungsaufgaben

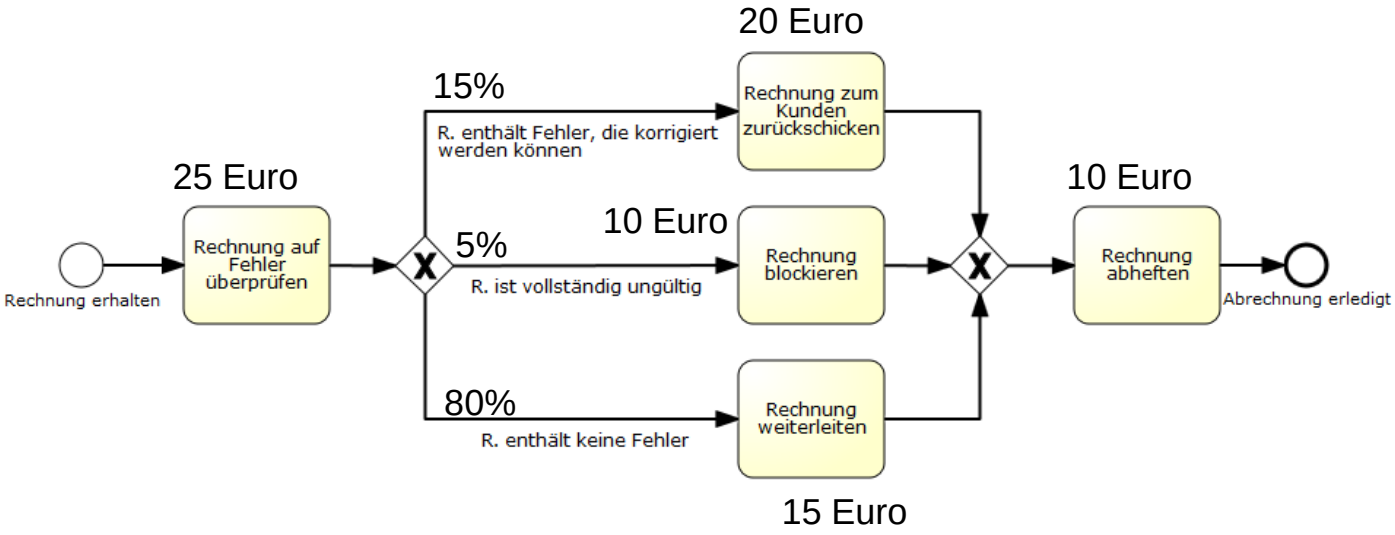
2. Notieren Sie jeweils drei unterschiedliche Ausführungspfade:



Übungsaufgaben

3. Simulieren Sie die Durchschnitte per Hand und mit Signavio

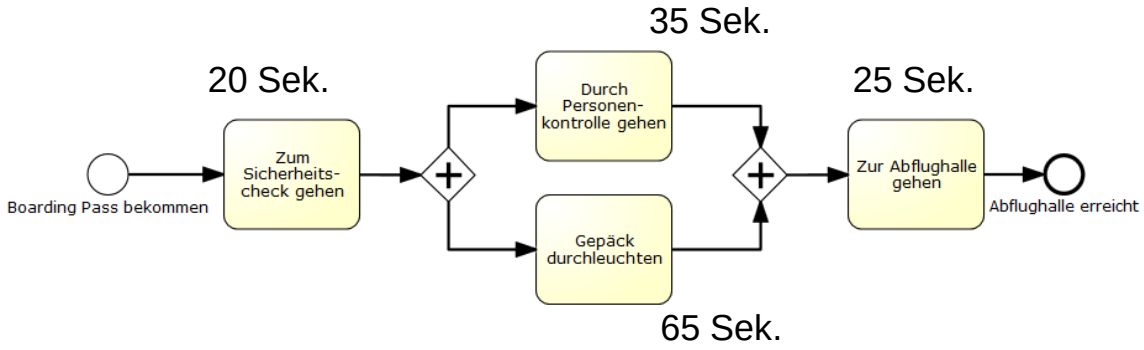
a.



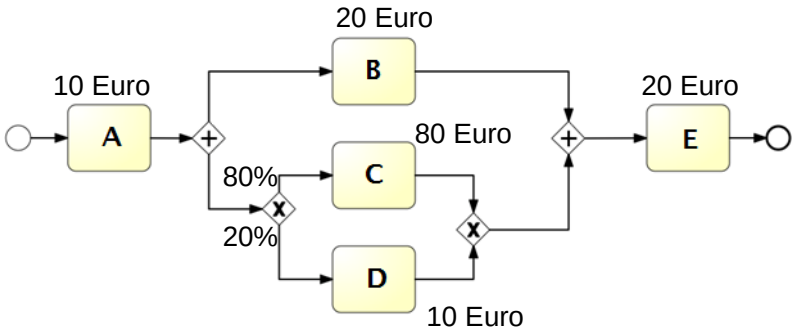
Übungsaufgaben

3. Simulieren Sie die Durchschnitte per Hand und mit Signavio

b.



c.



Softwareengineering

1 Nutzwertanalyse 

2 IT-Projekte 

3 Vorgehensmodelle 

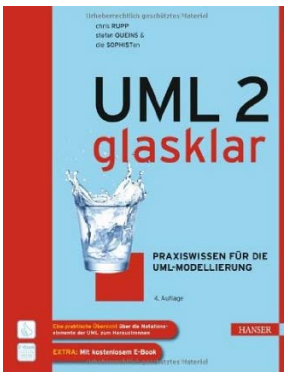
4 Geschäftsprozessmanagement 

5 UML Diagramme

UML Diagramme

Lesen..Nachschlagen..Lernen..

Literatur zu UML u.a.



- Rupp, Christiane et. al.
  UML2 glasklar Praxiswissen für die UML Modellierung
  Hanser, 2007

- Brandt-Pook, H.; Kollmeier, R.:
  Softwareentwicklung kompakt und verständlich;
  Wiesbaden 2008
  online via Springerlink

UML = Unified Modeling Language

Motivation:

- Komplexität verschiedener Systeme ist unterschiedlich groß
- Es gibt unterschiedliche Sichtweisen auf die Systeme

Beispiel

Hundehütte

- kann von einer Person gebaut werden
- Geringe Anforderungen an den Entwurf
- Simpler Prozessablauf
- Einsatz einfacher Hilfsmittel

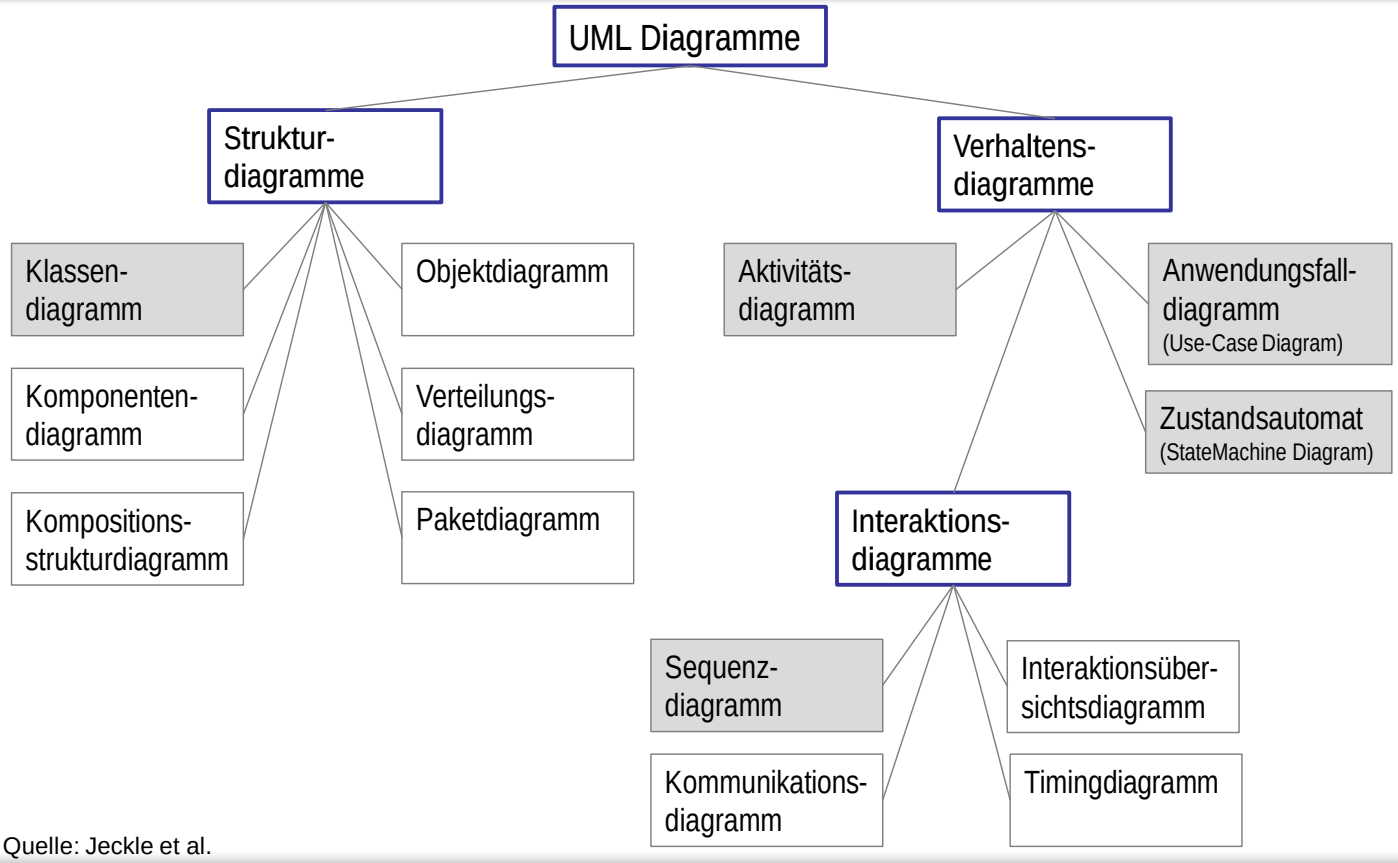
Mehrfamilienhaus

- wird von einem Team erbaut
- Komplexer Entwurf durch Architekten
- Umfassender Prozess mit vielen Rollen und Aufgaben
- Einsatz aufwändiger Werkzeuge

UML – Idee

- UML ist eine Sprache und Notation zur Spezifikation, Konstruktion, Visualisierung und Dokumentation von Modellen für Softwaresysteme.
- UML ist ein Sammelbegriff für bestimmte standardisierte Diagrammtypen, die Softwaresysteme abbilden.
- Ein System wird häufig durch mehrere Diagramme abgebildet, da die verschiedenen Diagramme verschiedene Sichtweisen (z.B. Ablaufmöglichkeiten, Folgen von Zuständen etc.) auf das System liefern.
- Grafisch und objektorientiert
- Weltweit anerkannt

UML – Diagrammarten



Quelle: Jeckle et al.

Strukturdiagramme
» Modellierung Systemstruktur
» Modellierung statischer Aspekte des Systems

Verhaltensdiagramme
» Modellierung von dynamischen Aspekten (Verhalten) eines Systems

UML – Idee (2)

Unterschied zu ER-Modell

- Der wesentliche Unterschied zwischen objektorientierten Konzepten wie UML zu Entity-Relationship Modellen besteht darin, dass Verhalten beim ER-Modell nicht modelliert werden kann. Das ER-Modell enthält keine Operationen oder Nachrichten und es benutzt Schlüssel zur Identifikation von Entitäten.

- Bei UML besitzt jedes Objekt ein einzigartiges Identifikationsmerkmal, welches unabhängig von Attributen ist.

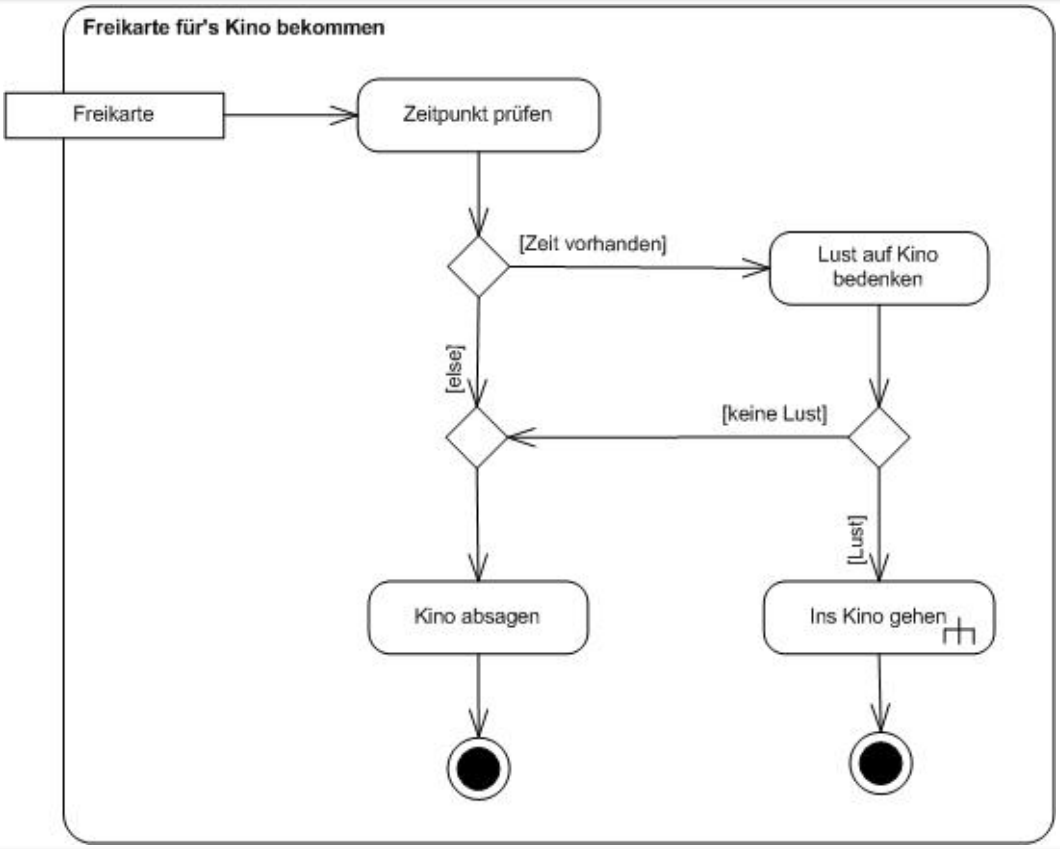
UML Diagramme

Aktivitätsdiagramm

Fragestellung: Wie sieht der Geschäftsablauf im Detail aus?

- Eignen sich zur Darstellung von Abläufen aller Art, vor allem um Geschäftsabläufe in einer frühen Projektphase darzustellen und zu beschreiben.
- Durch die nahezu selbst erklärenden Symbole einfach zu verstehen. Trotzdem lassen sich auch komplizierte Ablaufe darstellen.
  → Einfach, aber ausdrucksstark.
- Das Diagramm wird häufig um einen Text erweitert, der das Diagramm und die einzelnen Funktion erklärt, falls dies nicht selbst erklärend ist.

Aktivitätsdiagramm – Beispiel



Aktivitätsdiagramm – Notation

Element	Name	Bedeutung
	Aktivität	• Alles was man auf dem Diagramm sehen kann. Ein Rahmen für das Diagramm. • Oben links steht der Name der Aktivität
	Aktion	• Beschreibt, was passiert – Ein Prozess oder Funktionalität • Kleinster Einzelschritt innerhalb der Aktivität
	Objektknoten	• An der Aktivität beteiligte Daten oder Objekte, also „anfassbare“ Dinge

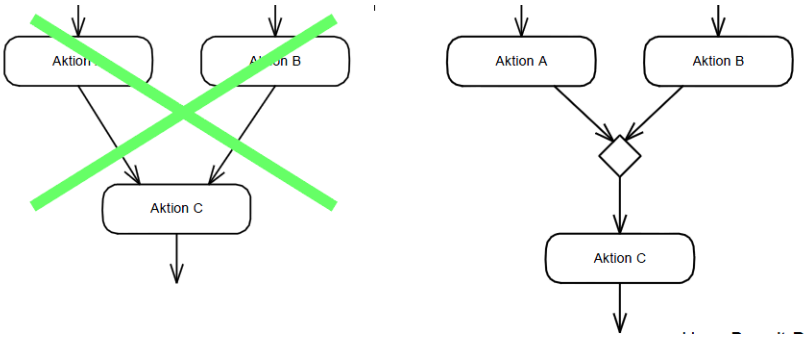
Aktivitätsdiagramm – Notation

Element	Name	Bedeutung
	Verzweigungs-knoten	• Unter bestimmten Bedingungen wird hier derjenige Pfad gewählt, auf den die Bedingung zutrifft • Nur einer der Wege wird beschritten • Der Verzweigungsgrund muss angegeben werden • Exklusive ODER-Verzweigung (XOR)
	Verbindungs-knoten	• Gegenstück zum Verbindungsknoten • Vorher verzweigte Wegewerden nun wieder zusammengeführt und vereinigen sich zu einem einzigem Pfad

Aktivitätsdiagramm – Notation

Element	Name	Bedeutung
	Kante	• Verbindung zwischen Aktionen • Pfeile geben die Richtung an

- Wege, die durch Verzweigungsknoten aufgespalten werden, werden typischerweise* durch Verbindungsknoten wieder zusammengeführt.
- In eine Aktion läuft genau eine Kante ein und es geht genau eine Kante ab: sonst Verzweigungs- bzw. Verbindungskonten nutzen:

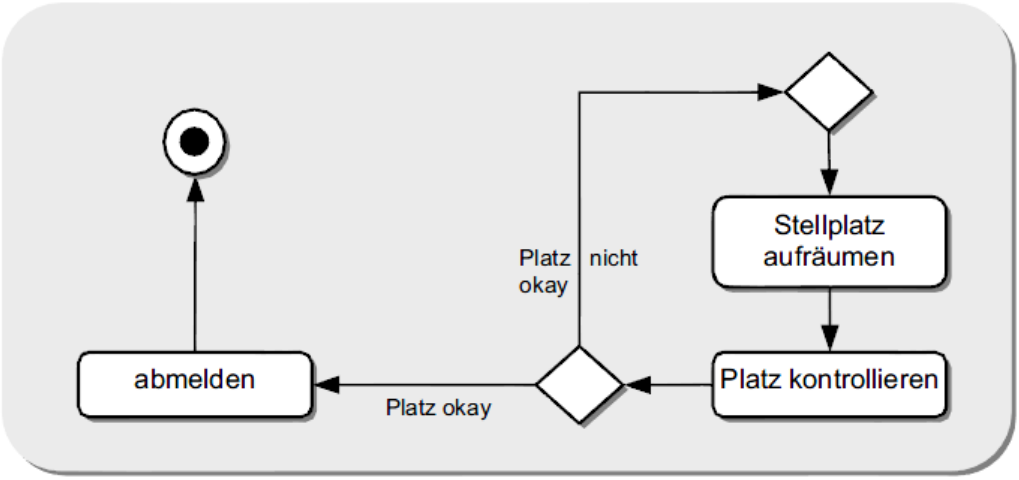


Eine Ausnahme wäre z.B. der Fall, wenn es mehrere Endpunkte im Diagramm gibt.

Aktivitätsdiagramm – Notation

Modellierung von Wiederholungen:

- Typisches Szenario in der Programmierung
- Endlosschleife möglich
- Modellierung durch Verbindungs- und Verzweigungsknoten



Quelle: Brandt-Pook; Softwareentwicklung kompakt und verständlich

Aktivitätsdiagramm – Notation

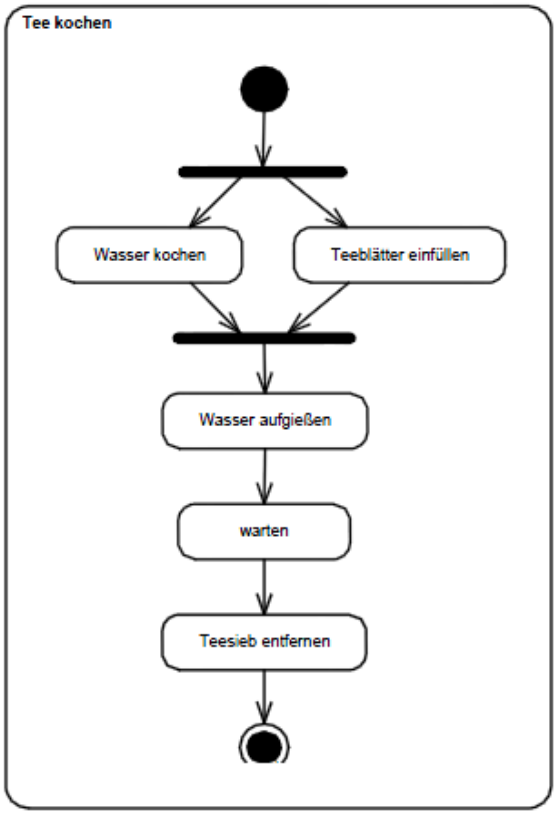
Element	Name	Bedeutung
	Parallelisierung	• Aufspaltung des Pfades. • Ab hier können Aktionen gleichzeitig und unabhängig voneinander erfolgen
	Synchronisation	• Gegenstück zur Parallelisierung. Mehrere Pfade werden hier zusammengeführt, also „synchronisiert“ • Implizite UND-Verbindung der Pfade

Aktivitätsdiagramm – Notation

Beispiel Parallelisierung:

Wasser muss gekocht sein und Teeblätter eingefüllt, bevor Wasser aufgegossen werden kann.

Ob gleichzeitig oder nach-einander ist egal, ebenso dann die Reihenfolge.

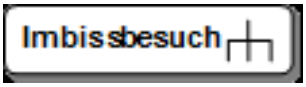


Aktivitätsdiagramm – Notation

Element	Name	Bedeutung
	Startknoten	• Anfang des Diagramms • Objekt statt Startknoten möglich, wenn die Existenz einer konkreten Instanz (bspw. eine Freikarte für‘s Kino) eine Aktivität auslöst
	Endknoten	• Ende des Diagramms • Es kann mehrere Endknoten geben

Aktivitätsdiagramm – Notation

- lasst sich sehr stark verfeinern, wird aber meist vermieden, wenn dies nicht unbedingt erforderlich ist. Das zu verfeinernde Element wird mit einem Gabelsymbol gekennzeichnet und kann in einem Unterdiagramm verfeinert werden.

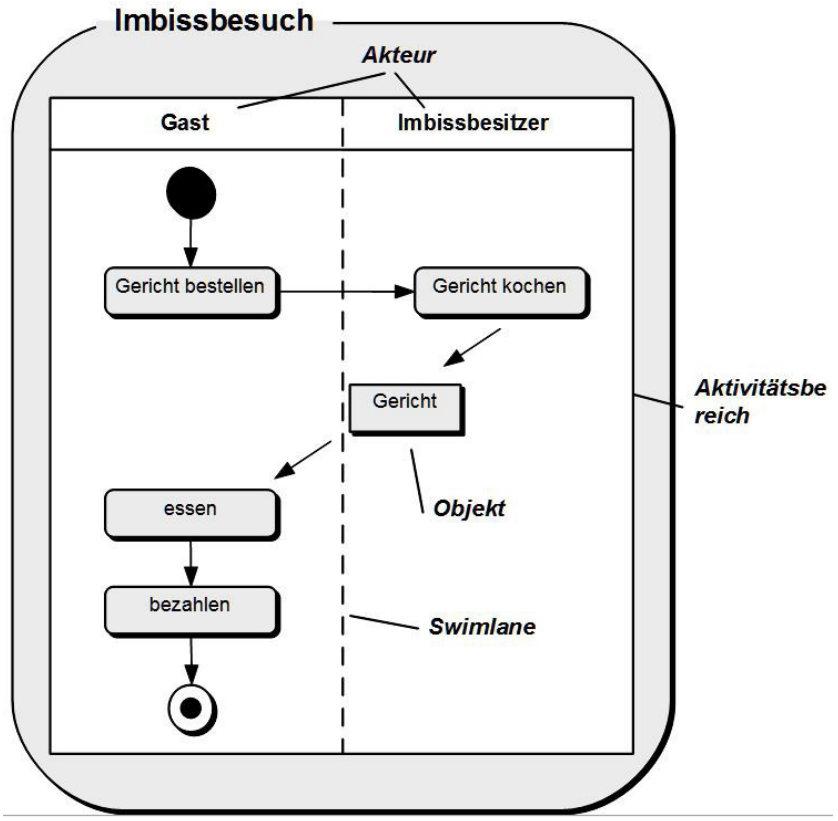


Aktivitätsdiagramm – Beispiel 2

Aufteilung des Diagramms (und der enthalten Aktionen) in unterschiedliche Verantwortungsbereiche.

Welcher Akteur hat welche Aktion?

Trennung der Akteure durch eine „Swimlane“.



Quelle: Brandt-Pook; Softwareentwicklung kompakt und verständlich

Aktivitätsdiagramm

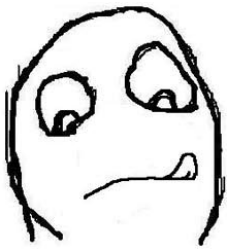
Diese Stärken hat das Diagramm:

- Detaillierte Visualisierung von Abläufen mit Bedingungen, Schleifen und Verzweigungen
- Parallelisierung und Synchronisation ist möglich
- Darstellung von Daten- und Kontrollflüssen
- Nicht auf IT-Projekte beschränkt

- Alternative zum Programmablaufplan PAP (Flussdiagramm) und BPMN

Zusammenfassende Fragen: Abschnitt 2.4.1 / 2.4.2

- Was ist UML?
- Welche Arten von Diagrammen gibt es?
- Wozu eigenen sich Aktivitätsdiagramme?
- Woraus bestehen Aktivitätsdiagramme?
- Wie kann ein Aktivitätsdiagramm verfeinert werden?
- Wie kann eine Schleife modelliert werden?



Übungsaufgaben

Erstellen Sie ein vollständiges (!) Aktivitätsdiagramm unter folgenden Aspekten:

- Inhaltliche Aspekte
  - Es gibt zwei Akteure: einen Käufer und einen Verkäufer
  - Es gibt folgende ungeordnete Aktivitäten: „Kauf tätigen“, „beraten“, „Wunsch äußern“ und „Geschäft verlassen“
  - Es gibt eine Bedingung: für einen Kauf muss die Beratung gut gewesen sein
- Formale Aspekte
  - Ihr Aktivitätsdiagramm nutzt Aktivitätsbereiche

Übungsaufgaben

Erstellen Sie ein Aktivitätsdiagramm unter folgenden Aspekten:

- Inhaltliche Aspekte
  Beschreiben Sie den ganzen Ablauf eines Campingplatzbesuches: Ankunft, Aufenthalt, Abreise, die wie folgt gekennzeichnet sind:
  - Anreise: Anmeldung; Stellplatz aussuchen durch die Besucherin, Stellplatz beziehen
  - Aufenthalt: Zelt aufbauen, Seele baumeln lassen, Wäsche waschen, Essen kochen, essen, Müll wegbringen.
  - Abreise: Abschluss-Check durch Platzwart, Bezahlen, evtl. für's nächste Mal reservieren.

- Der Platzwart hat mehrere Aufgaben, die ebenfalls dargestellt werden könnten, insbesondere Rasenpflege, Müllentsorgung, Abschluss-Check.
  Anmerkung: diese inhaltlichen Aspekte sind nur Anregungen. Falls Sie einen Aufenthalt anders erleben, können Sie das auch gerne anders beschreiben!

Aspekte zu Ihrem UML-Lernprozess
Überdenken Sie, ob Sie alles richtig und geschickt gemacht haben. Vielleicht ist es sinnvoll, eine geschachtelte Aktivitätenstruktur einzuführen?
Nutzen Sie diese Übung, um die verschiedenen Notationselemente des Aktivitätsdiagramms einzusetzen. Die Aufgabenstellung gibt das her!

- Aktionen
- Objektknoten
- Kanten
- Kontrollelemente
- Start- und Endknoten
- Verzweigungs- und Verbindungsknoten
- Parallelisierungs- und Synchronisationsknoten
- Aktivitätsbereiche

UML Diagramme

Use-Case Diagramm (UCD)

Fragestellung: Wer soll was mit dem System tun können?

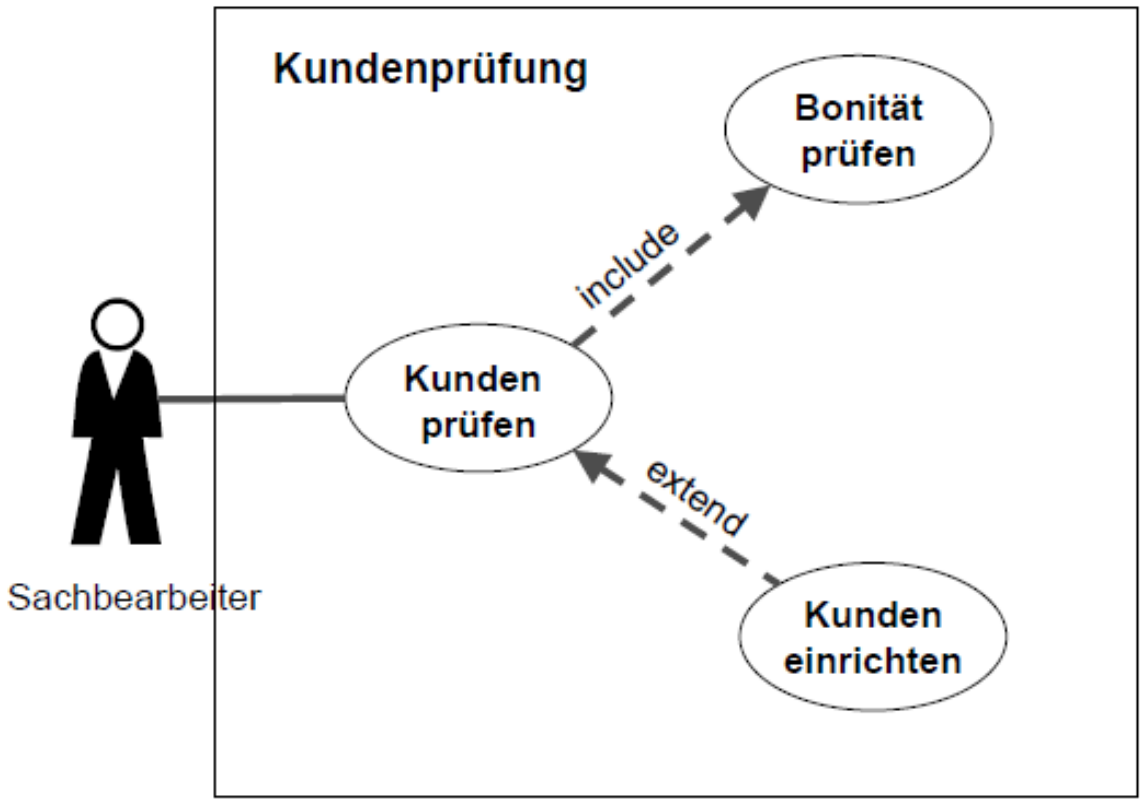
- Beispiele: Datei öffnen, Bild einfügen, Windows herunterfahren
- Ein Use-Case beschreibt
  - in strukturierter Form
  - ein Stück gekapselte* Funktionalität,
  - die ein Akteur mit einem IT-System durchführen kann.

- Use-Cases beschreiben, was man mit einem IT-System machen kann – nicht wie die Anforderungen im System umgesetzt werden!
- Oft gilt: Use-Cases beschreiben die IT-Unterstützung zur Durchführung der elementaren Geschäftsprozesse.

Synonym: Anwendungsfalldiagramm

*Gekapselt bedeutet: sinnvoll und in sich geschlossen; ein Aspekt wird voll und ganz beschrieben. „Alles was man wissen muss (gekapselt)“. Es ist egal, wer den Use-Case aufruft.

Use-Case Diagramm – Beispiel



Nach: Steinweg; Projektkompass Softwareentwicklung

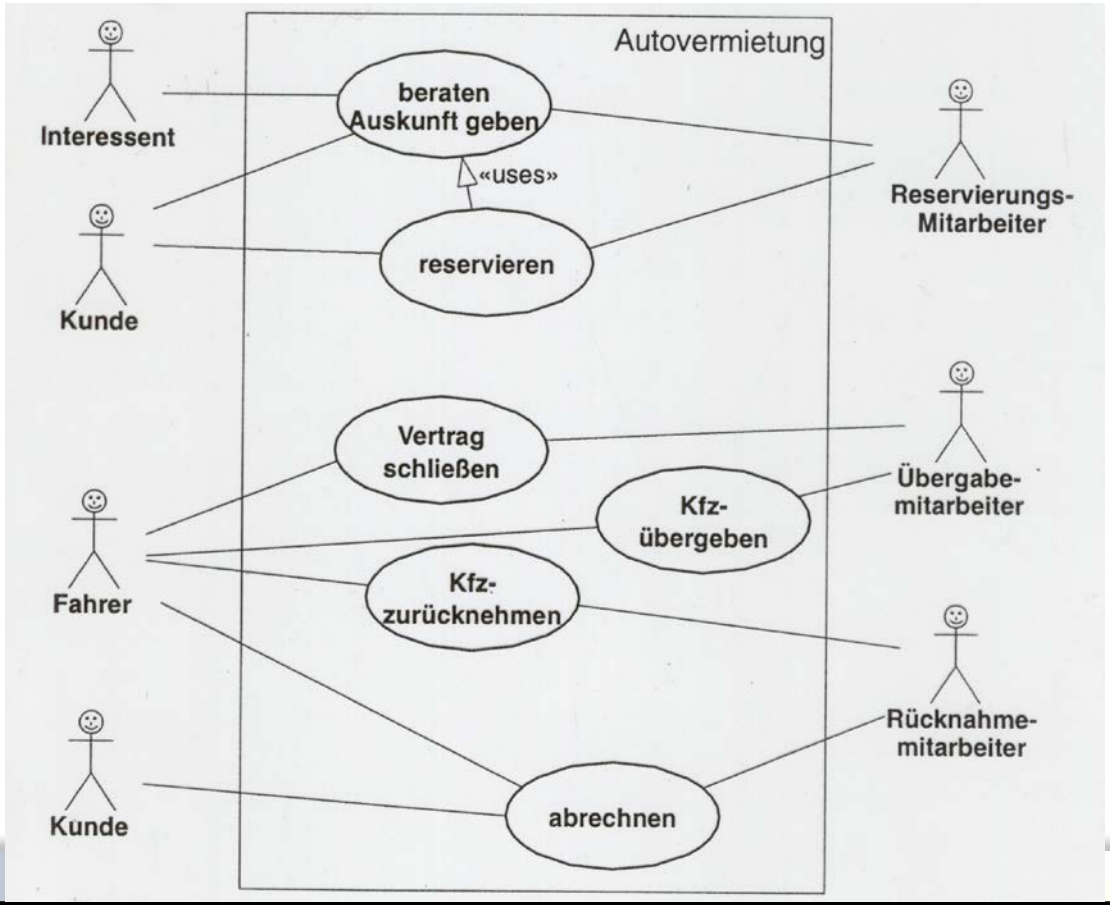
Use-Case Diagramm – Notation

Element	Name	Bedeutung
	Systemgrenze	• Alles außerhalb gehört nicht zu System
	Akteur	• Eine Rolle zur Nutzung des Systems • Kann auch wiederum ein System sein • Mehrere Akteure können einbezogen werden • Benutzt-Beziehung zur Verbindung sagt aus, das UC direkt benutzt werden kann
	Use-Case	• Use-Case ist ein Stück gekapselte Funktionalität • Funktionalität wird immer zusätzlich durch einen Text beschrieben

Use-Case Diagramm – Notation

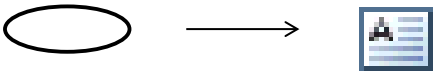
Element	Name	Bedeutung
	Beziehung	• Gestrichelte Pfeile als Verbindung • Beziehungen werden an den Pfeilen vermerkt • Include = Muss-Beziehung • Der inkludierte Use-Case wird stets aufgerufen • Extend = Kann-Beziehung • Der erweiterte Use-Case nutzt unter Umständen den anderen Use-Case

Use-Case Diagramm – Beispiel (2)



„uses“ als synonym zu:

Use-Case Beschreibung

Jeder Use-Case erhält eine Beschreibung: 

Name	• sinnvoll Kombination Substantiv und Verb • Bei großen Softwareprojekten zusätzlich Nummern oder Präfixe
Ziel	Ziel des Use-Case
Ablauf	Darstellung was die Funktionalität beinhaltet und wie sie abläuft
Vorbedingung	• Keine Selbstverständlichkeiten • Was die Vorbedingung abfängt, braucht in den Alternativen nicht beschrieben werden. • um eine Reihenfolge (zeitliche Abfolge) bei der Ausführung zu erreichen, die i.d.R. nicht erkennbar ist
Ergebnis	Wie hat sich das IT-System verändert bei erfolgreicher Ausführung
Alternative Abläufe	• Beschreibung von Sonderfällen • Sammelstellte für vorläufige Fragen

* kein Eintrag ebenfalls möglich

Use-Case Beschreibung (Beispiel)

Name	Kunde prüfen
Ziel	Kundeninformationen sind geprüft und aktuell
Ablauf	1. Der Sachbearbeiter prüft, ob der Kunde vorhanden ist. Falls nicht, wird der Kunde neu angelegt (UC Kunden einrichten) 2. Die Kundeninformationen werden ggf. aktualisiert 3. Die Bonität wird geprüft (UC Bonität prüfen)
Vorbedingung	Kundendaten liegen vor
Ergebnis	• Valide Kundendaten im System gespeichert • Kunde freigeschaltet
Alternative Abläufe	Anfrage übersteigt Kreditlimit

Use-Case Diagramm – Vorgehen

- 1. Use-Cases identifizieren
- 2. Use-Cases systematisch beschreiben
- 3. Use-Cases zu Diagrammen zusammenstellen
  - Beziehungen einzeichnen
  - Akteure identifizieren und eintragen

Wichtig:

- Schritte werden i.d.R. mehrmals durchlaufen!
- Use-Case i.d.R. Ausschnitt der Gesamtfunktionalität
- Use-Case kann in mehreren Diagrammen vorkommen

Ad1)
Wie groß sollte ein UC sein?
Alles was vor geistigem Auge auf einer Bildschirmmaske untergebracht wird

Ad2)
Wie vorgestellt: strukturiert und step-by-step

Ad3)
Keine Restriktionen in der UML-Spezifikation
Möglichkeiten
Akteurorientiert: alle Use-Cases die einem Akteur zur Verfügung stehen Fachexperte kann entscheiden ob Funktionen ausreichen
Prozessorientiert: alle Use-Cases die zur Erledigung einer Aufgabe benötigt werden

Use-Case Diagramm

Einsatz und Zusammenfassung

- Übersicht der Systemfunktionalität und die Einbettung eines Systems in das Geschäft des Auftraggebers
- Anforderungsermittlung aus Sicht der Akteure
- Stellt Anwendungsfälle und Akteure mit ihren jeweiligen Abhängigkeiten und Beziehungen dar
- Fähigkeiten des Systems werden in kleinen Häppchen dargelegt (modularer Aufbau) und stellt in der Regel nur einen Ausschnitt der Gesamtfunktionalität dar

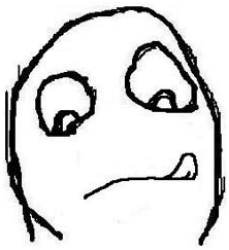
Use-Case Diagramm

Einsatz und Zusammenfassung (2)

- Wird von Auftraggeber und Auftragnehmer verwendet und dient als Kommunikationsgrundlage
- Use Cases sind das Bindeglied zwischen Geschäft und IT
- Use Cases haben viele Querverbindungen zu anderen Diagrammen
- Use Cases begleiten den Entwicklungsprozess bis zur Realisierung

Zusammenfassende Fragen: Abschnitt 2.4.3

- Wozu dient ein Use-Case Diagramm?
- Welche Bestandteile hat ein UCD?
- Was ist der unterschied zwischen einem Use-Case und einem UCD?
- Zeichnen Sie ein Beispiel eines UCD.
- Wie erfolgt die Beschreibung eines Use-Case?



Übungsaufgaben

Diese Übung kommt auf das Campingplatzbeispiel zurück.

1.) Erstellen Sie nun ein Use-Case Diagramm unter folgenden Aspekten:

- Das Diagramm enthält alle notwendigen Bestandteile.
- Das IT-System soll zu Beginn nur den Anreiseprozess unterstützen. Dazu wird die Aktion „Anmeldung“ als zentraler Prozess betrachtet, der abgebildet werden soll.
- Das Diagramm benötigt weiterhin die Use-Cases „Kunden anlegen“, „Platz buchen“, „Platz suchen“.

Dabei gilt:

- Der Use-Case „Anmeldung“ ist direkt mit dem Akteur verbunden.
- Wenn der Kunde bereits im System erfasst ist, muss er nicht neu angelegt werden.
- In jedem Fall kann der Platz nur gebucht werden, wenn die Suche einen freien Platz ergab.

2.) Fertigen Sie eine Beschreibung aller Use-Cases an!

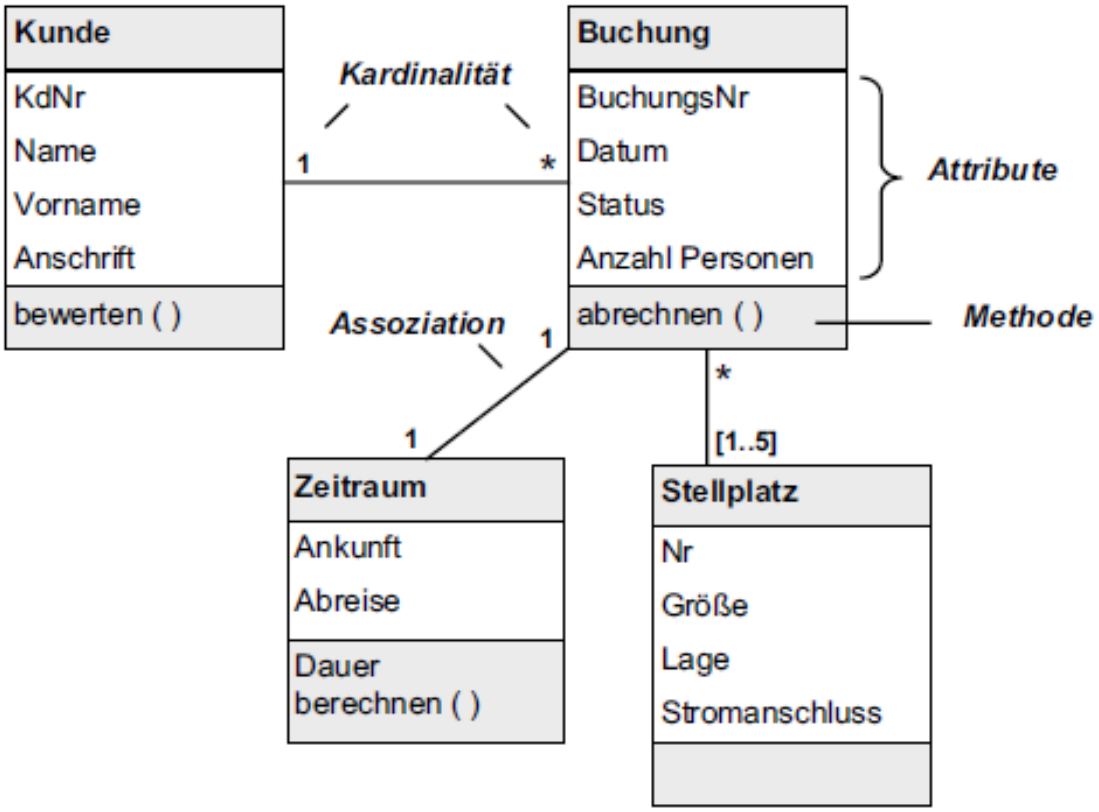
UML Diagramme

Business Object Model (BOM)

Fragestellung: Wie ist die Beziehung zwischen Objekten?

- Objekte mit Eigenschaften und Fähigkeiten sowie ihre Beziehungen untereinander werden beim BOM in einem Diagramm dargestellt.
- Objekte sind Begriffe die sich im IT-System widerspiegeln. Meistens werden Dinge, Rollen oder Ereignisse mit Hilfe von Objekten modelliert. Beispiele für typische Objekte sind: Kunde, Rechnung, Auftrag, Storno und Artikel.
- BOM ist oft erste Grundlage für die Arbeit der Programmierer.
  →Die Objekte finden sich in der Programmierung wieder.
- Es lassen sich Datenbankstrukturen aus dem BOM ableiten.

Business Object Model – Beispiel



Quelle: Brandt-Pook; Softwareentwicklung kompakt und verständlich; S. 84

Business Object Model – Notation

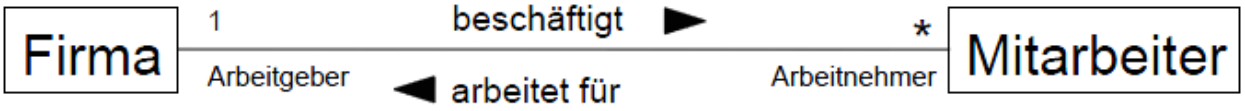
Element	Bedeutung
Objekt	Beschreibt einen Begriff (im Singular). Hier sind die Objekte: Kunde, Buchung, Zeitraum, Stellplatz.
Attribut	Eigenschaften des Begriffs. Beschreiben das Wissen eines Objektes. Attribute müssen vorhanden sein.
Methode	Fähigkeiten des Objektes. Beschreiben das Können eines Objektes. Nicht jedes Objekt benötigt eine Fähigkeit.
Assoziation	Verbindung zwischen zwei Objekten. Es gilt: keine unnötigen Assoziationen einzeichnen.
Kardinalitäten	Geben an, in welcher Häufigkeit die Objekte an der Assoziation beteiligt sind. n → genau n n .. m → eine Zahl zwischen n und m n, m → entweder n oder m * → beliebig viele

Konkret in dem Beispiel heißt das:

- Jede Buchung bezieht sich auf genau einen Kunden. Ein Kunde kann beliebig viele Buchungen tätigen
- Zu jeder Buchung gehört genau ein Zeitraum. Jeder Zeitraum gehört zu genau einer Buchung
- Zu einer Buchung können eins bis fünf Stellplätze gehören (man kann also max.fünf Stellplätze gleichzeitig buchen). Jeder Stellplatz kann in beliebig vielen Buchungen vorkommen.

Business Object Model – Notation (2)

- Assoziationen können Namen haben, die beteiligten Objekte Rollenbezeichner



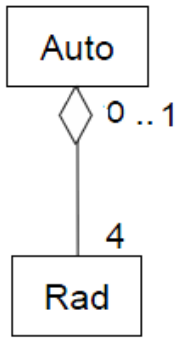
Business Object Model – Notation (3)

Vereinigung

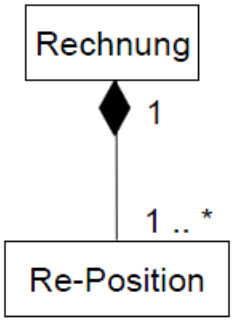
Zusammenstellung

Aggregation und Komposition
Zwei Sonderformen bei den Assoziationen:

- Aggregation: „hat Teil“:
  Darstellung durch offene Raute
  mit der Multiplizität 0 oder 1

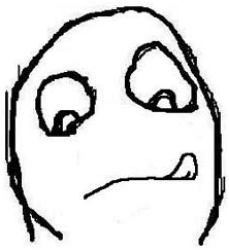


- Komposition:
  „hat existenzabhängiges Teil“:
  Darstellung durch geschlossene Raute
  und der Multiplizität 1



Zusammenfassende Fragen: Abschnitt 2.4.4

- Welche Arten von Kardinalitäten gibt es?
- Welche Arten von Assoziationen gibt es und wie können sie eingesetzt werden?



Übungsaufgabe

Diese Übung kommt auf das Campingplatzbeispiel zurück.

1.) Erstellen Sie nun ein BOM unter folgenden Aspekten:

- Das Diagramm enthält das zentrale Objekt Anmeldung mit den Attributen (Nr., Datum, AnzahlPersonen)
- Weitere Objekte sind Kunde (Nr., Name, Anschrift), Zeitraum (Ankunftsdatum, Abreisedatum), Abrechnung (Nr.), Stellplatz (Nr., Qm, Art, Größe).
- Es handelt sich um jeweils 1-1 Beziehungen, außer dass natürlich Kunden die Möglichkeit haben, mehrere Stellplätze gleichzeitig zu mieten.
- Zeichnen Sie sowenig Assoziationen wie nötig.
- Jedes Objekt hat eine Methode. Was könnten sinnvolle Methoden sein?

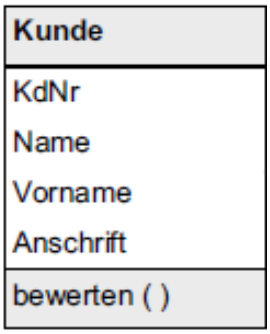
UML Diagramme

Zustandsdiagramm

Fragestellung: Welche Zustände haben die Objekte aus dem BOM?

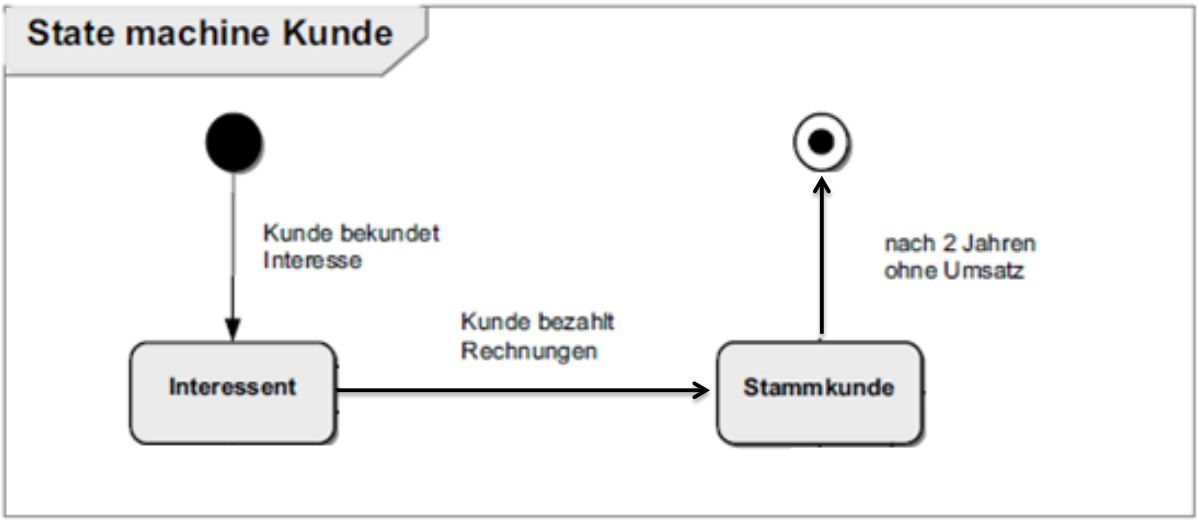
- Zeigt in dem Diagramm die möglichen Zustände eines Objektes auf und dokumentiert wie diese gewechselt werden
- Es zeigt anhand von Zuständen den Lebenslauf eines Objektes
- Zustandsübergänge werden von Ereignissen ausgelöst

- Beispiel:
  Ein Objekt „Rechnung“ kann den Status „Bezahlt“ oder „Offen“ erhalten



Andere Bezeichnungen sind Zustandsautomat bzw. state machine (state=Zustand).

Zustandsdiagramm – Beispiel

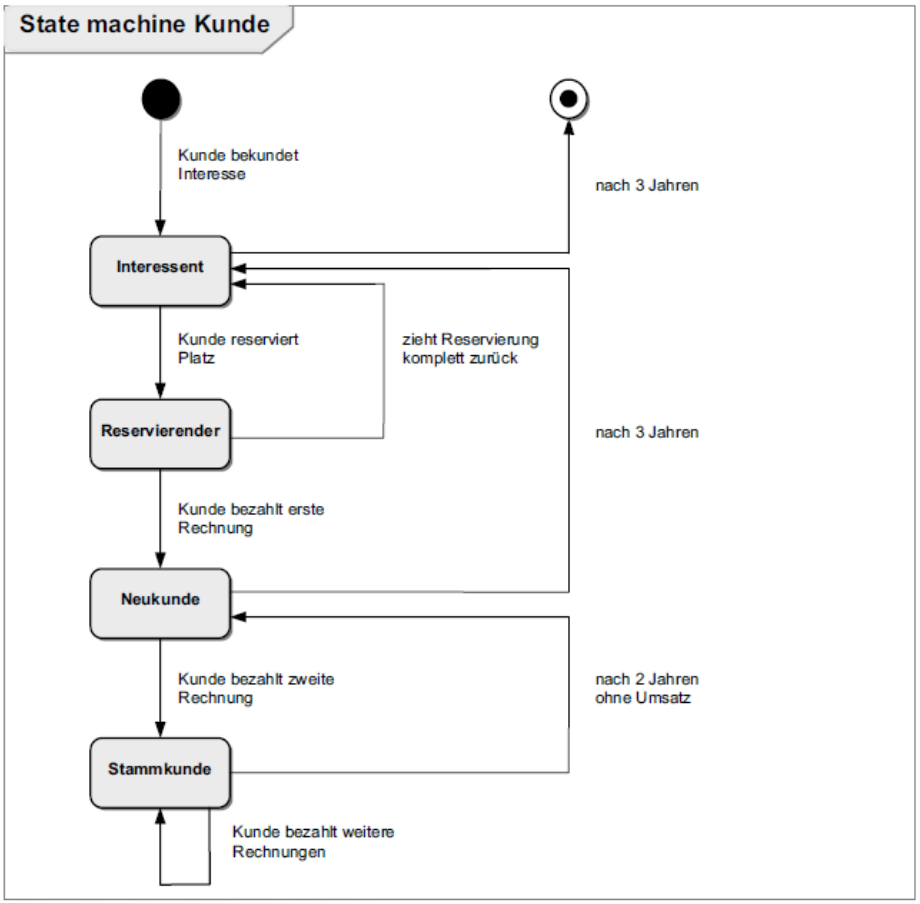


Quelle: Brandt-Pook; Softwareentwicklung kompakt und verständlich; S. 95

Zustandsdiagramm – Notation

Element	Name	Bedeutung
	Zustandsdiagramm	• beinhaltet alle Zustände eines Objektes (z.B. Kunde) • Begriff State maschine bezeichnet die Art des Diagramms
	Zustand	• Zustände, die das Objekt annehmen kann • Beinhaltet verständliche Bezeichnung (z.B. Neukunde)
	Ereignis	• Verbindet die Zustände • Pfeilspitze gibt Folgezustand an • Bezeichnung ist erforderlich
	Startzustand Endzustand	• Anfang und Ende

Zustandsdiagramm – Beispiel (2)



Quelle: Brandt-Pook; Softwareentwicklung kompakt und verständlich; S. 95

UML Diagramme

Objekt Sequenz Diagramm (OSD)

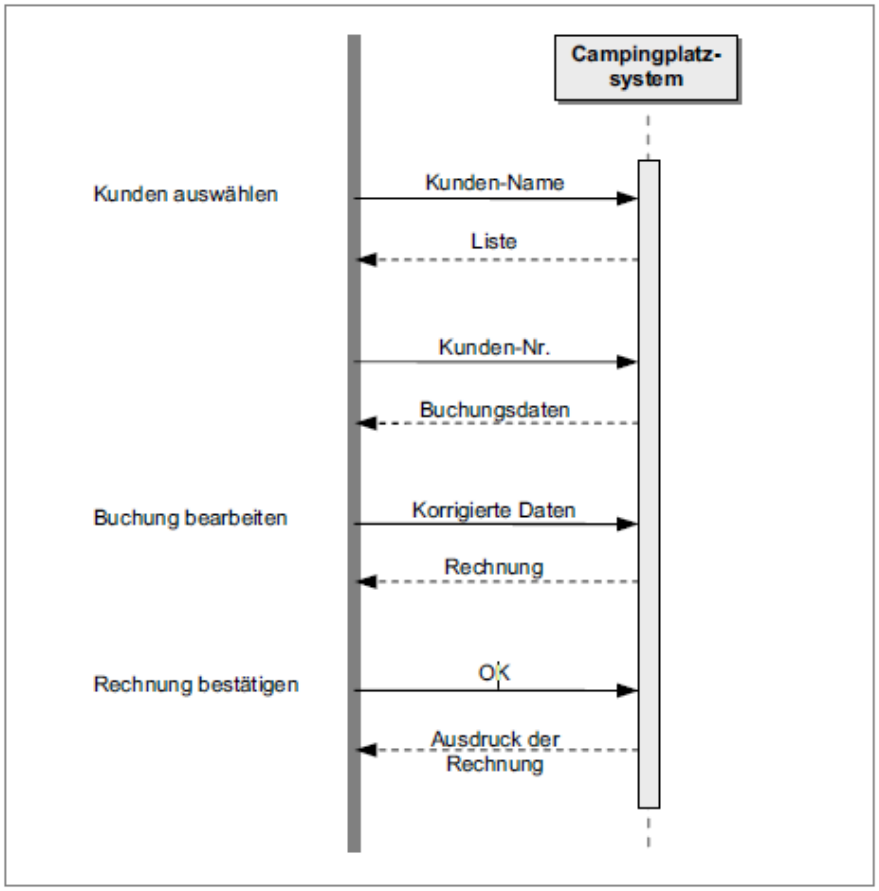
Fragestellung: Wie arbeiten die Objekte aus dem BOM zusammen?

- Ein Objekt Sequenz-Diagramm beschreibt das Zusammenwirken von Objekten bei der Erledigung einer Aufgabe
- zeigt Abläufe entlang der Zeitachse
- zeigt die Wechselbeziehung von Objekten bei der Erledigung einer Aufgabe
- stellt dar, wie die Objekte miteinander sprechen und welche Informationen sie dabei austauschen
- wird häufig im Design und in der Realisierung eines Systems benutzt

- eignet sich dazu, sehr ins Detail zu gehen
  → komplizierte OSD nur für erfahrene UML Anwender*

*Deswegen in der Notation und in Aufgaben nicht weiter vertieft

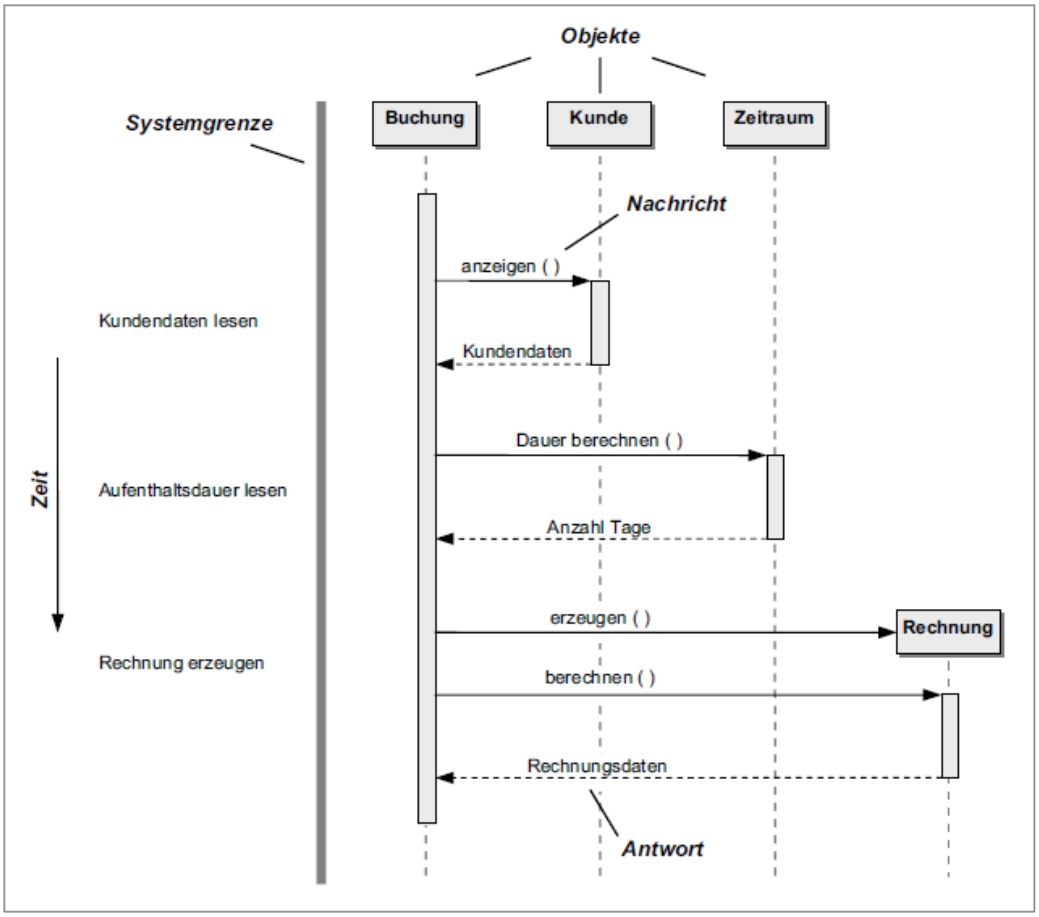
Objekt Sequenz Diagramm – Beispiel



Quelle: Brandt-Pook; Softwareentwicklung kompakt und verständlich; S. 99

In dieser einfachsten Form wird gezeigt, wie ein Benutzer mit dem System arbeitet.

Objekt Sequenz Diagramm – Beispiel (2)

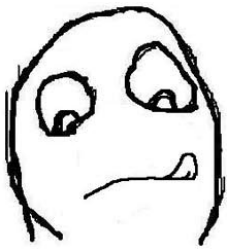


Quelle: Brandt-Pook; Softwareentwicklung kompakt und verständlich; S. 98

Im Regelfall sind dies die Objekte aus dem BOM, aber das ist nicht festgelegt. Das OSD eignet sich ebenfalls dazu, den Ablauf eines Use-Case oder einen Prozess über mehrere Use-Cases hinweg darzustellen.

Zusammenfassende Fragen: Abschnitt 2.4.5 + 2.4.6

- Wozu dienen das Zustandsdiagramm und das OSD?
- Wie können Zustände dargestellt werden?
- Wie werden Zustände verbunden?
- Zeichnen und erläutern Sie in knapper Form ein Beispiel eines Zustands- und OS-Diagramms.



UML Diagramme

4.1	Überblick	✓
4.2	Aktivitätsdiagramm	✓
4.3	Use-Case Diagramm	✓
4.4	Business-Objekt-Modell (i.F.e. Klassendiagramms)	✓
4.5	Zustandsdiagramm	✓
4.6	Objekt-Sequenz Diagramm	✓
4.7	Zusammenspiel	

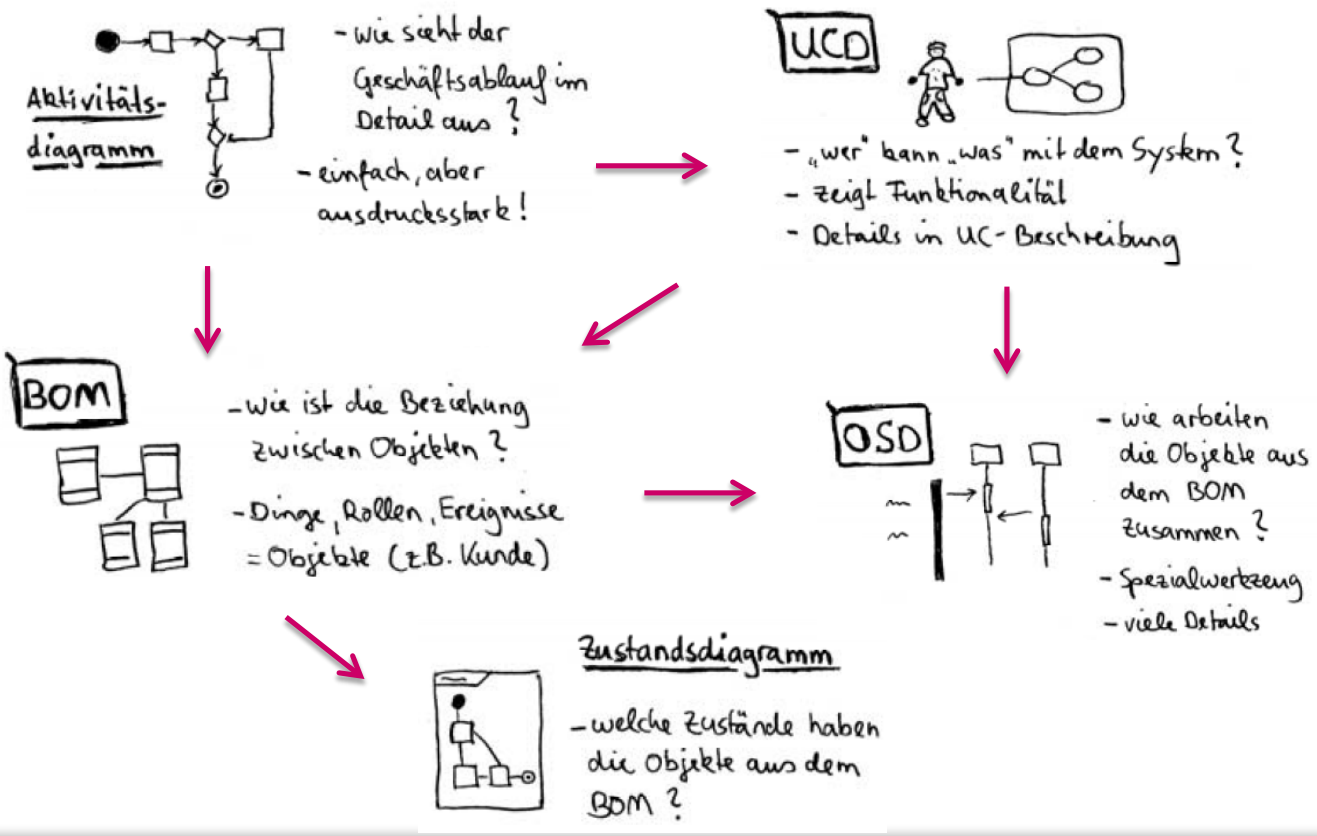
Überblick

- Alle Methoden sind für ihr Einsatzgebiet sinnvoll.
- Methoden greifen in einander, doch gibt es keinen Automatismus.

Häufig aber werden …:

- interessante Prozesse werden in einem Aktivitätsdiagramm dargestellt,
- den Aktionen des Aktivitätsdiagramms Use-Cases zugeordnet und diese zu UCD‘s zusammengefasst,
- die Beschreibungen der Use-Cases zur BOM Erstellung herangezogen
- Use-Cases mit dem OSD in konkrete Abläufe umgesetzt,
- die Objekte des BOM mit Zustandsdiagrammen mit ihrem Status dargestellt.

Überblick (2)



Quelle: Brandt-Pook; Softwareentwicklung kompakt und verständlich; S. 105

Überblick (3)

- Wie spielen die UML-Methoden mit den Phasen des Basismodells zusammen?

Phase:

Methode:

Zusammenfassende Fragen: Abschnitt 2.4

- Was ist UML?
- Welche Diagrammarten kennt UML?
- Wo liegt der Fokus der behandelten Diagramme?
- Wie können die Diagramme ineinander greifen?