MHP
A PORSCHE COMPANY

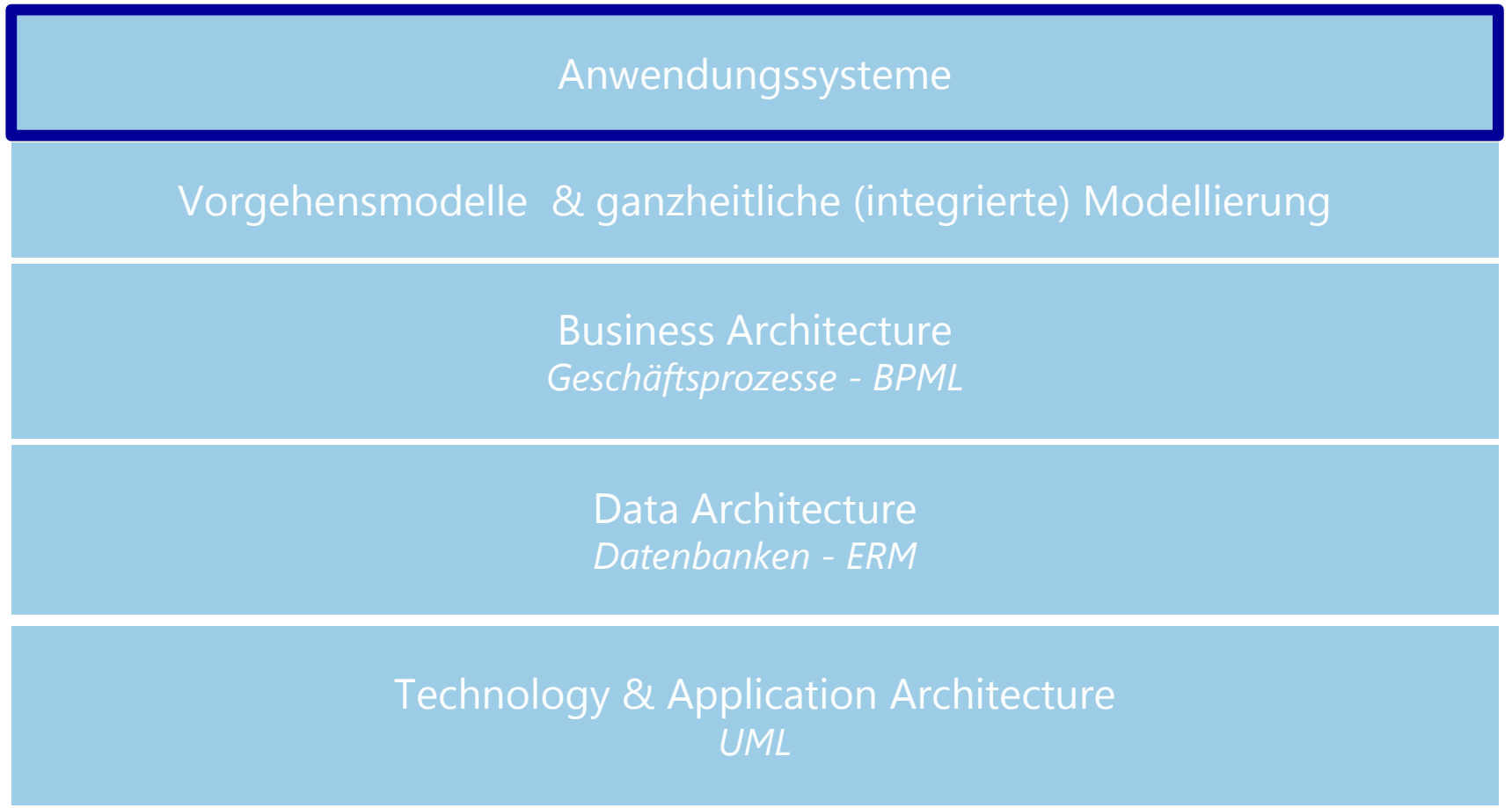
VORLESUNG
INTEGRIERTE MODELLIERUNG KOMPLEXER SYSTEME
Anwendungssysteme

UNIVERSITATEA
BABEŞ-BOLYAI

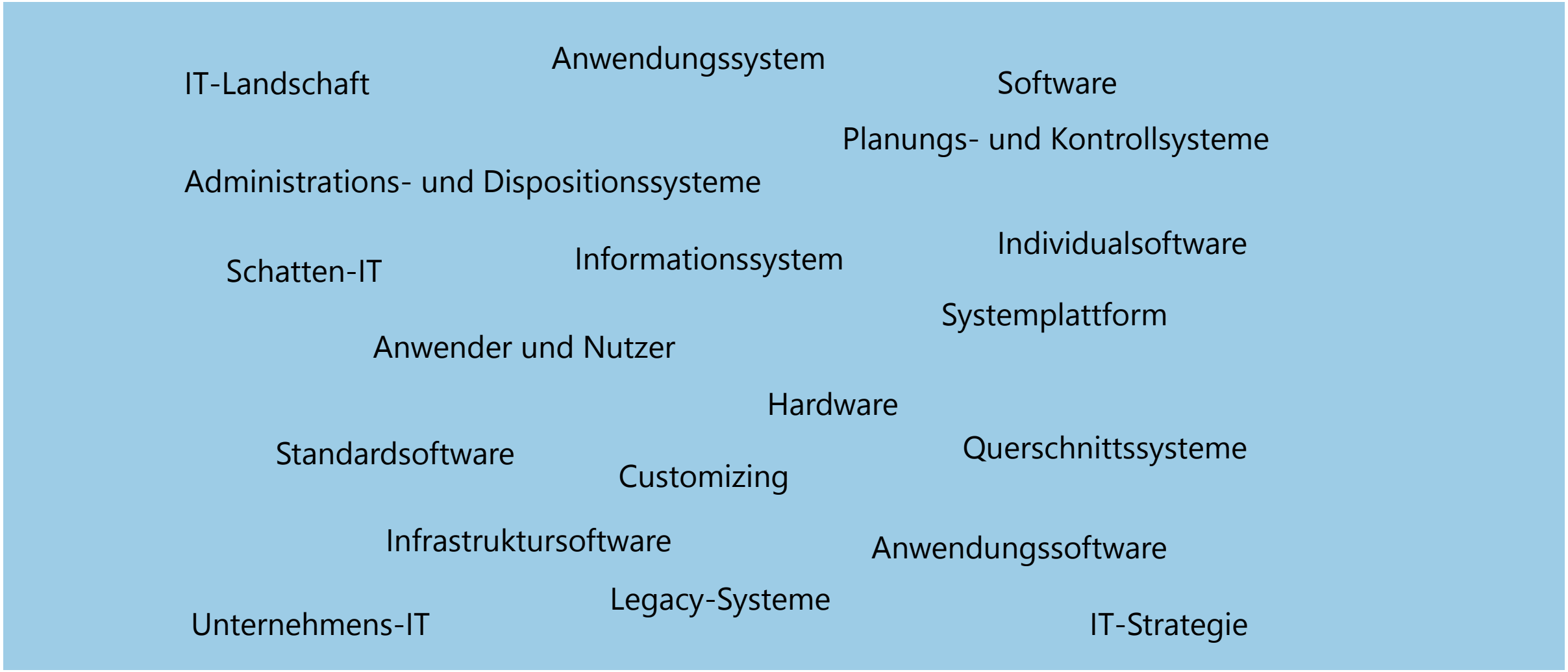
Hugo Colceag | Bachelor Studiengang Informatik

Vorlesungsinhalte und Aufbau

MHP
A PORSCHE COMPANY



Mindmap - Anwendungssysteme



Lernziele

- ✓ Komponenten eines Informationssystems
- ✓ Funktionen eines Informationssystems

- ✓ Unterschied Individualsoftware + Standardsoftware
- ✓ IT Landschaft eines Unternehmens
- ✓ Aufgaben der IT eines Unternehmens

Agenda

MHP
A PORSCHE COMPANY

1. **Informationssysteme**
2. IT im Unternehmen
3. Literatur

Begriff „Informationssystem“

- Informationssysteme (IS) sind soziotechnische Systeme, die **menschliche** und **maschinelle** Komponenten (Teilsysteme) umfassen. Sie unterstützen die **Sammlung**, **Strukturierung**, **Verarbeitung, Bereitstellung, Kommunikation** und **Nutzung** von Daten, Informationen und Wissen sowie deren Transformation.

- IS tragen zur Entscheidungsfindung, Koordination, Steuerung und Kontrolle von Wertschöpfungsprozessen sowie deren Automatisierung, Integration und Virtualisierung unter insbesondere ökonomischen Kriterien bei. IS können Produkt-, Prozess- und Geschäftsmodellinnovationen bewirken.

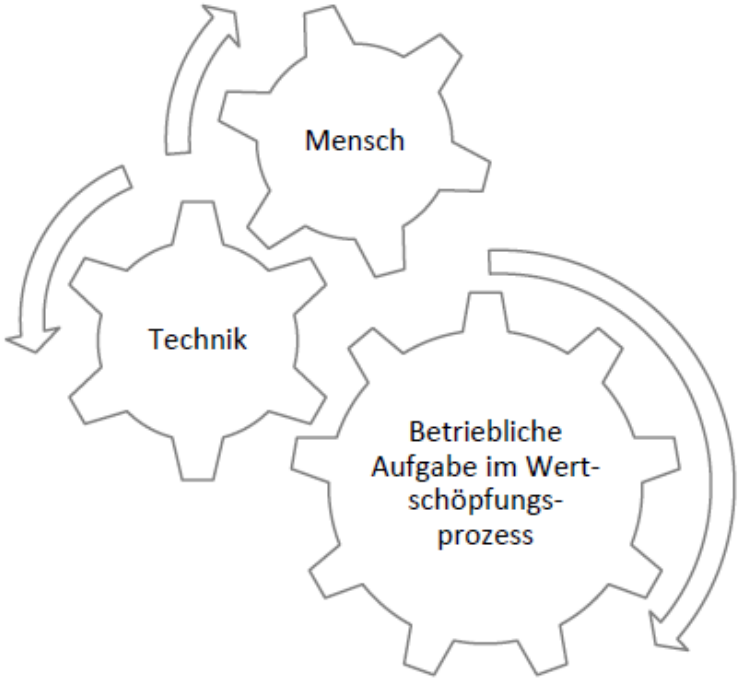


Abbildung 1-1 Komponenten eines Informationssystems

1.1.1 Anwender und Nutzer [Soziale Komponenten im Informationssystem]

- verschiedenen Rollen
- Unterscheidung ist die zwischen
  - **Anwendern**
    - wirtschaftlicher Nutzen aus Informationssystemen
    - Personen oder Organisationseinheiten die ein Informationssystem einrichten
    - verantwortlich, dass und wie ein Informationssystem eingerichtet wird
    - wählen die technischen Komponenten aus, lassen sie entwickeln und führen sie ein
  - **Nutzern/Benutzern** oder *user*
    - arbeiten unmittelbar an den IT-Systemen
    - arbeiten routinemäßig / aktiv mit den technischen Komponenten des Informationssystems
  - **Beispiele**:
    - Endnutzer in den Fachabteilungen
    - IT-Personal wie Entwickler, Systemadministratoren, Programmierer
    - Mitarbeiter im IT-Support und interne
    - externe Berater (*IT consultants*) als Experten für bestimmte Anwendungssoftware

1.1.2 Besondere Rollen von Endnutzern

- Beispiel 1.1 Soziale Komponenten eines Informationssystems

Ein aufstrebendes Unternehmen, die Hoske Gmbh, hat vor 2 Jahren ein ERP-System eingeführt, das viele Änderungen in den Arbeitsabläufen mit sich gebracht hat.

Der Einkäufer Herr Engel erstellt mithilfe dieses **ERP-Systems** Bestellungen und verwaltet damit Angebote von Lieferanten. Die Mitarbeiter im Vertrieb sind ebenfalls **Endbenutzer** des ERP-Systems. Sie bearbeiten damit Kundendaten und Kundenaufträge.

Der Systemadministrator Herr Stöppler ist dafür zuständig, dass das ERP-System stabil läuft und die Daten regelmäßig gesichert werden.

Anwender		Unternehmen Hoske
Benutzer	Endbenutzer	Einkäufer Herr Engel
		Vetriebsmitarbeiter
	IT-Personal	Systemadministrator Herr Stöppler

1.2.1 Hardware und Software [Technische Komponenten im Informationssystem]

Anwendungssystem

- technische Komponenten eines Informationssystems
- besteht aus Hardware und aus Software

Die **Hardware** umfasst

- Rechnern / Computer
- Peripheriegeräten wie Drucker, Massenspeicher, Barcodeleser, Scannerkassen, Zeiterfassungseinrichtungen, und vieles andere mehr;
- die Kommunikationseinrichtungen, insbesondere der Netzwerkanbindung.

Die **Software** umfasst

- Anwendungssoftware
  - dient unmittelbar dazu, Aufgaben im Betrieb zu erfüllen

- Infrastruktursoftware
  - ist erforderlich, um Anwendungssoftware betreiben zu können

1.2.2 Anwendungssoftware und Infrastruktursoftware

- **Anwendungssystem (im engeren Sinn**)

  Ein Anwendungssystem ist eine Anwendungssoftware zusammen mit den damit erstellten und bearbeiteten Daten.

- **Anwendungssystem (im weitergefassten Sinn)**

  Ein Anwendungssystem im engeren Sinn einschließlich aller Hardware und Systemsoftware, die zum Betrieb der Anwendungssoftware nötig sind.

  Beinhaltet weitere Softwarekomponenten, die im Hintergrund laufen und Dienste und Funktionen für die eigentliche Anwendungssoftware bereitstellen

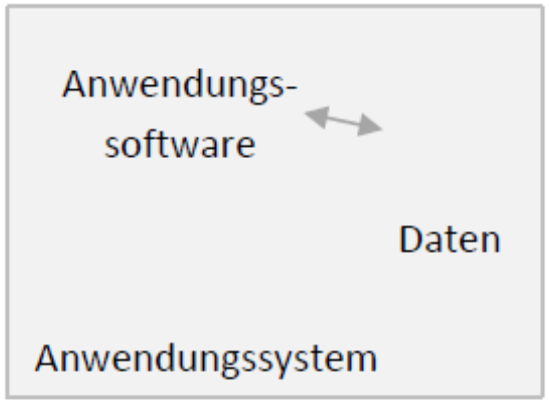


Abbildung 1-2 Anwendungssoftware und Anwendungssystem (im engeren Sinn)

1.2.2 Anwendungssoftware und Infrastruktursoftware

- **Anwendungssoftware**
  - Eine (betriebliche) Anwendungssoftware ist eine Software, mit der Endbenutzer eine bestimmte (betriebliche) Aufgabe oder Aufgaben aus einem (betrieblichen) Anwendungsgebiet bearbeiten können.
  - Besitzt typischerweise eine Benutzeroberfläche
- **Infrastruktursoftware** (Systemsoftware)
  - stellt Funktionen und Dienste bereit, die die eigentliche Anwendungssoftware nutzen kann, die Endnutzer aber nicht routinemäßig direkt bedienen.
  - Ein Datenbankmanagementsystem ist ein typisches (aber nicht das einzige) Beispiel von Infrastruktursoftware.
  - Komponenten der Infrastruktursoftware sind für die Benutzer unsichtbar
  - Zusammenstellung von Infrastruktursoftware eines Anwendungssystems bezeichnet man auch als **stack**. Ein deutscher Begriff dafür, der auch die Hardware mit einschließt, ist „**Systemplattform**“.
  - Infrastruktursoftware, Rechner und sonstige Hardware zusammen bezeichnet man auch als die **Systemplattform**.

1.2.2 Anwendungssoftware und Infrastruktursoftware

MHP
A PORSCHE COMPANY

Beispiel 1.2 Typische Beispiele für Systemplattformen

1. Ein aufstrebendes Unternehmen möchte einen Webshop einrichten. Kunden und Interessenten sollen den Webshop über ihren Browser besuchen und benutzen können. Eine solche Web-Applikation benötigt einige **Infrastrukturkomponenten** im Hintergrund. Dieser Webshop soll auf einem sogenannten LAMP-**Stack** laufen (LAMP = Linux, Apache, MySql, PHP). Auf der **Server** des Unternehmens läuft als **Betriebssystem** Linux, außerdem die **Webserver-Software** Apache, die die Verbindung zum WWW herstellt. Sie empfängt Anfragen aus dem Internet und sendet passende Webseiten an die richtige Adresse zurück. Die **Datenbanksoftware** MySQL verwaltet die Datenbank, in der Artikeldaten, Kundendaten und Daten zu den einzelnen Einkäufen gespeichert werden. Die eigentliche **Webshop-Software** ist in der Sprache PHP programmiert und wird von einem **PHP-Interpreter** ausgeführt. Der PHP-Interpreter baut Inhalte der **Datenbank** dynamisch zu Webseiten zusammen. Der Webserver sendet diese Webseiten an den abrufenden Browser. Alle diese **Softwarekomponenten** sind frei erhältlich.

2. Ein Unternehmen setzt das **ERP-System** MS Navision ein, um damit Aufgaben in Beschaffung, Produktionsplanung, Logistik und Vertrieb auszuführen. Dieses System läuft auf **Betriebssystemen** von Microsoft und greift auf Microsoft SQL **Server** als **Datenbanksoftware** zu.

3. Um die komplexen Abläufe in der Logistik zu steuern, lässt ein Unternehmen eine **Software** in der Programmsprache **Java** individuell entwickeln. Das Software läuft auf einem Rechner mit dem **Betriebssystem** Linux und der **Anwendungsserver-Software** WebSphere der Firma IBM und greift auf eine **DB2-Datenbank** zu. DB2 ist ein **Datenbankmanagementsystem** von IBM.

1.3 Betriebliche Aufgaben

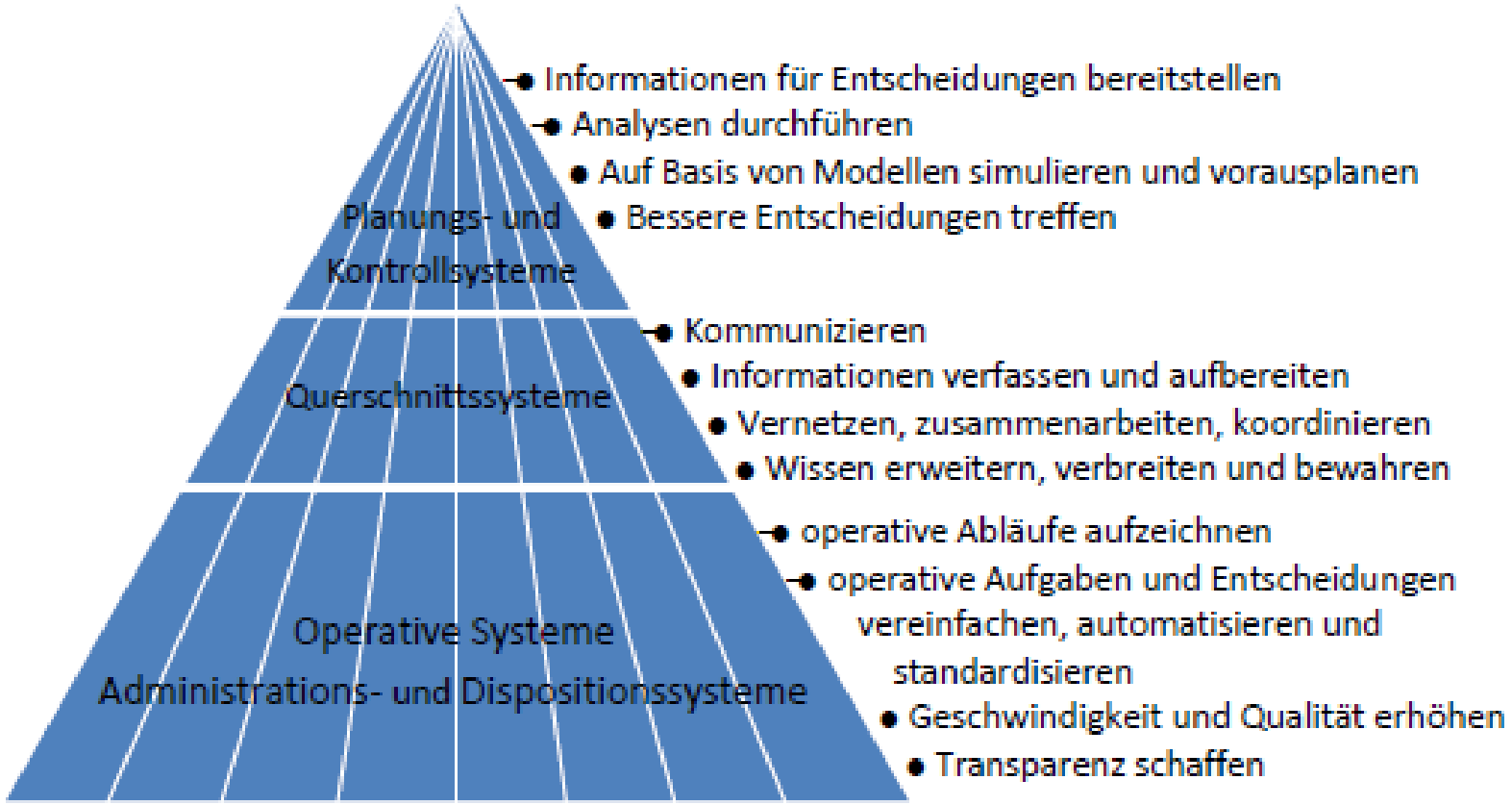


Abbildung 1-3 Überblick über Anwendungssysteme nach Einsatzzweck und Nutzerkreis (ähnlich zum Beispiel in (Mertens, et al., 2012))

1.3.1 Administrations- und Dispositionssysteme

- Abbildung und Abwicklung von betrieblichen Abläufen
- werden auch als operative Systeme bezeichnet
- Unterstützung bei operativen Aufgaben, die häufig standardisiert oder nach festen Vorgaben ablaufen

- Benutzer: unteres Management und Sachbearbeiter

- Dispositionssysteme **bereiten Entscheidungen** durch Menschen auf der operativen Ebene **vor** oder **automatisieren Entscheidungen** auf der operativen Ebene.

Ziele:

- Rationalisierung der Massendatenverarbeitung
- Kosten senken
- Personal von Routineaufgaben zu entlasten
- Abläufe beschleunigen

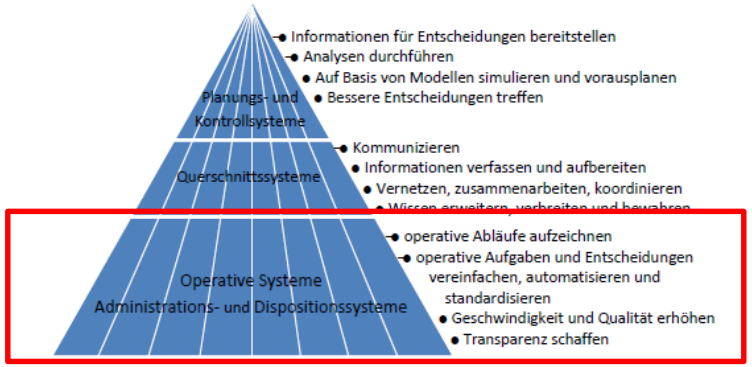


Abbildung 1-3 Überblick über Anwendungssysteme nach Einsatzzweck und Nutzerkreis (ähnlich zum Beispiel in (Mertens, et al., 2012))

1.3.1 Beispiele Administrationssystem

- Beispiel 1.3 Administrationssystem: Auftragsabwicklung real und digital

1.3.1 Beispiel Dispositionssystem

- Beispiel 1.4 Dispositionssystem

  Ein Dispositionssystem unterstützt **routinemäßige Entscheidungen** auf der **operativen** Ebene, zum Beispiel im Bereich der Beschaffung. Es beinhaltet verschiedene Verfahren zur Berechnung von Bestellmenge und Bestellzeitpunkt .

  Im System ist hinterlegt, welches Verfahren für welchen Artikel angewandt werden soll. Das System überwacht laufend Lagerbestände und Planungen und kann auf Basis dieser Informationen mithilfe des passenden Verfahrens Bestellvorschläge erzeugen. Die Mitarbeiter in der Beschaffung rufen diese Vorschläge ab und setzen sie in tatsächliche Bestellungen um.

  Weitere Beispiele zur Vorbereitung von Entscheidungen und automatische Ausführung einfacher Aufgaben:

  - Außendienststeuerung im Vertrieb
  - Tourenplanung
  - Kalkulation in der Kostenrechnung

1.3.2 Querschnittssysteme

- Werden über Hierarchieebenen und Funktionsbereiche oder Abteilungen hinweg von vielen Mitarbeitern im Unternehmen genutzt
- Systeme für die individuelle Büroarbeit, Office-Softwarepaketen mit Textverarbeitung, Tabellenkalkulation und Präsentationssoftware
- Systeme für den Zugang zum Internet, vor allem Email und Webbrowser
- Etablierte Systeme zur Unterstützung der Zusammenarbeit (Dokumentenmanagementsysteme und Workflowsysteme)
- Zunehmend auch Anwendungen aus dem Web2.0 und dem Social Web,

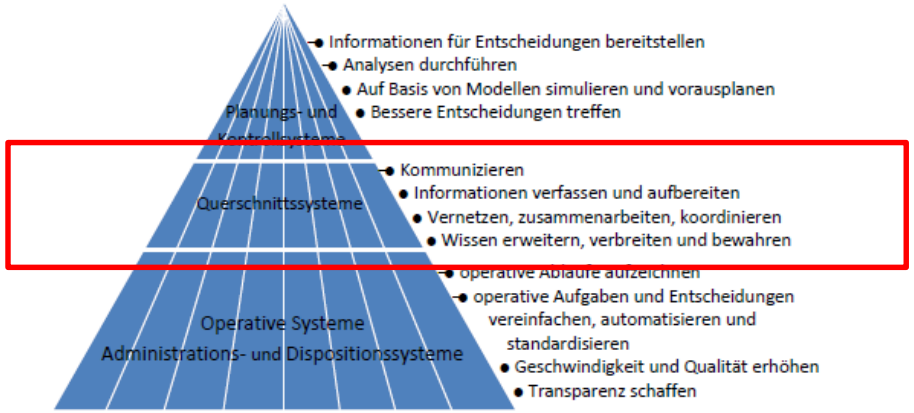


Abbildung 1-3 Überblick über Anwendungssysteme nach Einsatzzweck und Nutzerkreis (ähnlich zum Beispiel in (Mertens, et al., 2012))

1.3.2 Beispiel Querschnittssystem

- Beispiel 1.6 Querschnittssystem: Komax optimiert die globale Zusammenarbeit und das Know-how-Management

  „Als führende Herstellerin innovativer und qualitativ hochstehender Lösungen für die Kabelverarbeitung, für die Fertigung von Modulen in der Photovoltaik sowie für Anlagen zur Herstellung von Anwendungen im Bereich der Selbstmedikation unterstützt Komax wirtschaftliche und sichere Fertigungsabläufe insbesondere bei Automobilzulieferern, Solarpanelherstellern und Pharmaunternehmen. Komax beschäftigt insgesamt 1‘350 Mitarbeitende und erwirtschaftete im Jahr 2012 einen Umsatz von rund 300 Mio. Schweizer Franken.

  Komax wuchs in den vergangenen Jahren sowohl organisch als auch durch Übernahmen. Daraus resultierte eine dezentrale Struktur mit weltweit 24 Standorten. Um die globale Zusammenarbeit und den Informationsaustausch innerhalb der dezentralen Organisation und mit Kunden zu erleichtern, führte Komax zusammen mit dem …-Spezialisten IOZ AG eine umfassende **Kollaborationsplattform** ein. Diese umfasst vordefinierte **Workspaces** für Projekte und Gremien, eine **Dokumenten-Suche**, die auch auf **Netzlaufwerke** zugreift, individuelle **Mitarbeiter-Profile** mit Kompetenzen, eine visuelle Darstellung aller Prozesse und **Workflows** zur Prozessautomatisierung.

1.3.2 Beispiel Querschnittssystem

- Beispiel 1.6 Querschnittssystem: Komax optimiert die globale Zusammenarbeit und das Know-how-Management

  Diese Workspaces erleichtern die team- und standortübergreifende Zusammenarbeit durch eine **gemeinsame Dokumentenablage**, einen **Projektplan** und ein **Diskussionsforum**. Die **Projekt- Workspaces** erlauben zudem, Kunden in die Projekte einzubeziehen, indem sie, mit den entsprechenden Rechten ausgestattet, direkt auf Dokumente und Informationen zugreifen können.

  Kompetenz-Profile

  Damit auch die Erfahrungen aller Mitarbeitenden für Komax nutzbar werden, präsentieren sich alle Mitarbeitenden neu mit einem individuellen Profil. Dieses enthält die Position und Aufgabenbereiche, erworbene Qualifikationen und Erfahrungen sowie diejenigen Projekte, in denen sich der Mitarbeitende engagiert."

  Auszugsweise zitiert aus (IOZ InformationsOrganisationsZentrum AG, Sursee, Schweiz., 2013)

1.3.3 Planungs- und Kontrollsysteme

Unterschiedliche Begriffe

- Management-Unterstützungssysteme (Überbegriff)
- Führungsinformationssysteme (für das Top-Management)
- Managementinformationssysteme
- Entscheidungsunterstützungssysteme

→ **In der Praxis:** englischsprachigen Begriffe aus dem Umfeld der **Business Intelligence**

- Beruhen auf betriebliche Modelle und Methoden
- stellen ausgewählte und zusammengefasste Informationen für das mittlere Management bereit
- unterstützen Entscheidungsprozesse des Managements
- Fokus auf taktische und strategische Ebene

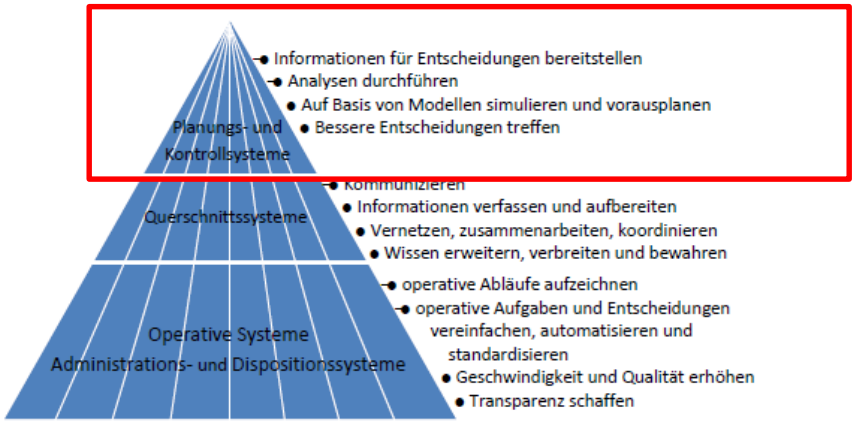


Abbildung 1-3 Überblick über Anwendungssysteme nach Einsatzzweck und Nutzerkreis (ähnlich zum Beispiel in (Mertens, et al., 2012))

1.3.3 Planungs- und Kontrollsysteme

- **Planungssysteme** unterstützen die betriebliche Leitungsebene bei schlecht strukturierten Problemen, z.B. Absatzplanung, globale Unternehmensplanung.

- **Berichtssysteme** decken den Informationsbedarf für operative Entscheidungen in Form periodischer Berichte oder Soll-Ist-Abweichungen.

- **Kontrollsysteme** überwachen die Einhaltung von Vorgaben: Kontrolle des Risikoportfolios einer Versicherung.

- **Entscheidungssysteme** haben die Aufgabe unternehmerische Planungen und Entscheidungen zu fundieren (Höchste Verdichtungsstufe der Unternehmensdaten). Schwerpunkt ist eine kompakte Darstellung der betrieblichen Situation und der kritischen Erfolgsfaktoren.

- **Analysesysteme** verdichten die Daten aus operativen Systemen uns werten sie aus. Auch externe Quellen werden oft einbezogen. Beispiel Nutzung eines Data Warehouse und Business Intelligence Lösungen.

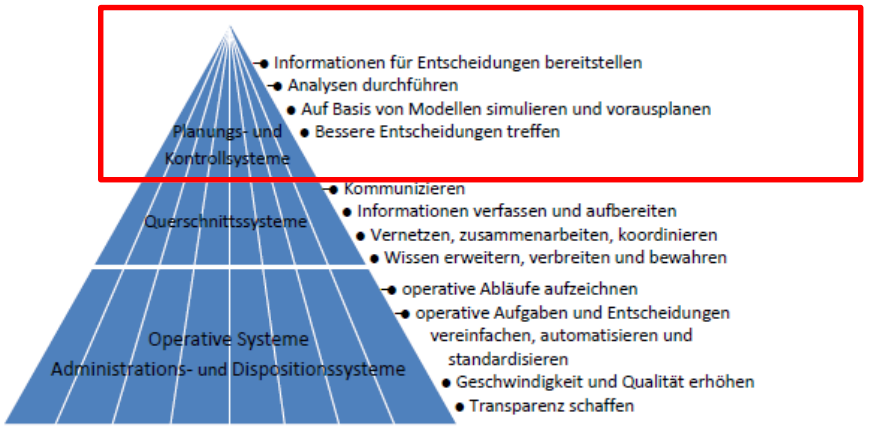


Abbildung 1-3 Überblick über Anwendungssysteme nach Einsatzzweck und Nutzerkreis (ähnlich zum Beispiel in (Mertens, et al., 2012))

1.3.3 Beispiel Planungs- und Kontrollsystem

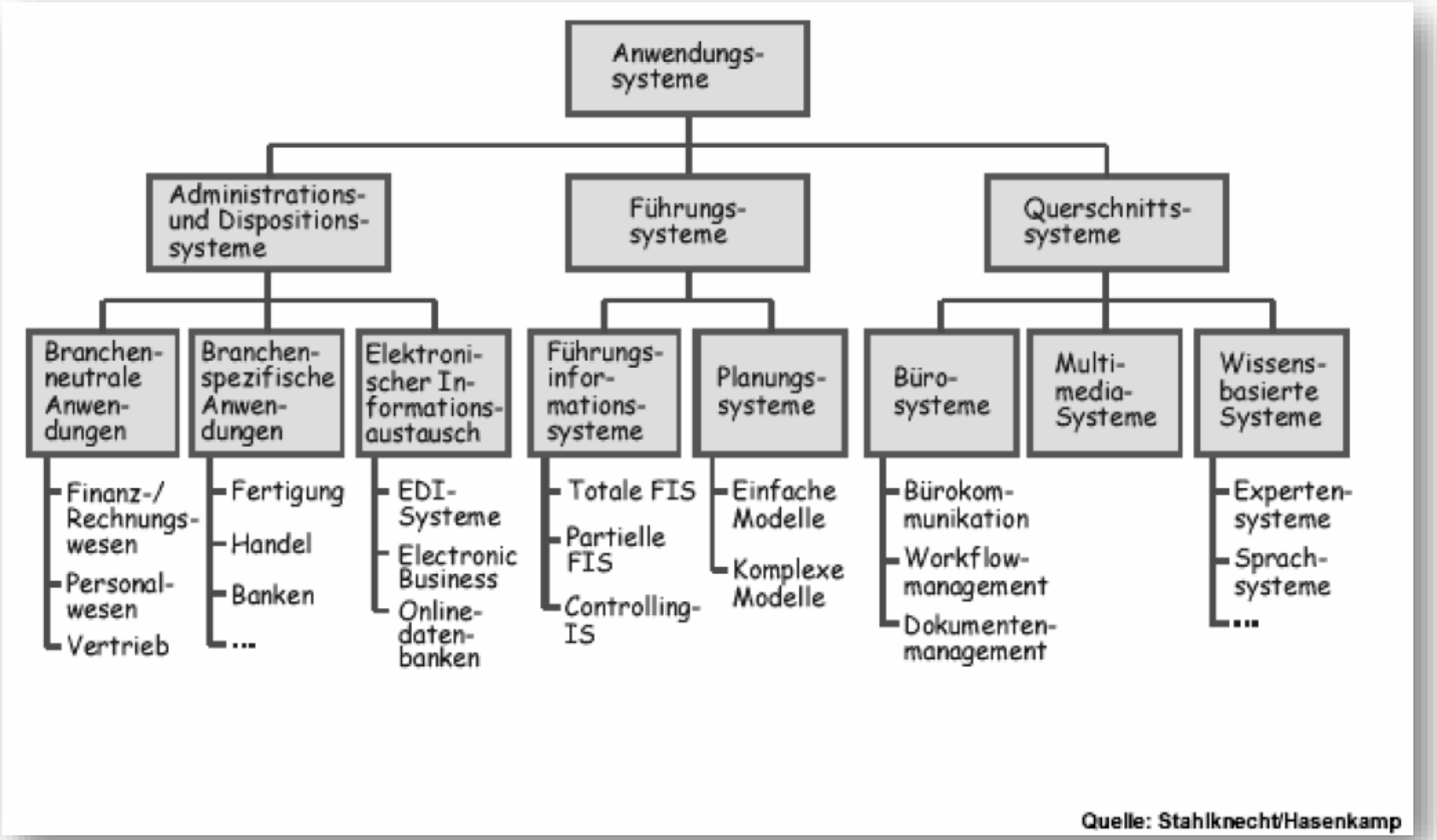
- Beispiel 1.5 Planungs- und Kontrollsystem: Firma Knebel schärft die Geschäftsintelligenz

  Die Firma Knebel GmbH & Co. KG bündelt auf der …-Plattform Informationen aus allen wichtigen Geschäftsbereichen.

  Damit kann das Management nun **Businessdaten** und **Kennzahlen** im Handumdrehen analysieren und das Unternehmen effizienter führen und steuern.

  - Welche Umsätze haben wir im letzten Monat bzw. Quartal mit unseren Top-Ten-Kunden gemacht, und welche Produktgruppen waren hier wie prozentual vertreten?
  - Bei welchen Kunden haben wir im gleichen Zeitraum am meisten Umsatz verloren und in welchen Produktgruppen?
  - Wo liegt derzeit unser Cash-Forecast?

1.3.4 Beispiele übergreifend



1.4 Übungen

MHP
A PORSCHE COMPANY

Aufgabe 1.1 Komponenten von Informationssystemen

In einem Informationssystem wirken soziale und technische Komponenten zusammen, um betriebliche Aufgaben zu erfüllen.

a) Welche Rollen können Menschen in einem Informationssystem innehaben? Stellen Sie die Rollen übersichtlich in einem Baumdiagramm dar.

b) Wie setzen sich die technischen Komponenten eines Informationssystems zusammen? Skizzieren Sie auch diese in einem Baumdiagramm.

1.4 Übungen

MHP
A PORSCHE COMPANY

Aufgabe 1.2 Anwendungsbeispiel

Als aufstrebendes modernes Unternehmen bietet die Hoske GmbH ihren Kunden die Möglichkeit, per Webshop bequem online zu bestellen. Als Webshop-Software wird die Open Source-Software Magento eingesetzt. Magento erzeugt die Webseiten und Bedienoberflächen, auf denen sich die Kunden informieren, Produkte bewerten, Fragen stellen und ihre Bestellungen aufgeben können.

Die Inhalte (Produktbeschreibungen, Abbildungen, Preise, Bewertungen etc.), die auf den Webseiten angezeigt werden sollen, speichert und verwaltet Magento in einer MySQL-Datenbank. Die für den Webshop verantwortliche Mitarbeiterin der Hoske GmbH, Frau Sellski, pflegt regelmäßig aktuelle Werbetexte und Informationen über die Magento-Bedienoberfläche in das System ein.

Eine Webserver-Software (Apache) sorgt dafür, dass der Webshop im Internet aufgefunden werden kann und dass die gewünschten Webseiten angezeigt werden, wenn ein Besucher im World Wide Web auf den Webshop navigiert.

Da es nur ein sehr kleiner Webshop mit wenigen Daten ist, den bisher nur wenige Kunden kennen und besuchen, genügt ein einziger Rechner, um die gesamte Webshop-Software auszuführen. Magento, MySQL und der Apache Server laufen bei der Hoske GmbH auf dem Betriebssystem Linux. Edi Junge aus der IT-Abteilung der Hoske GmbH prüft regelmäßig die Logdateien, in denen die Magento-Software Zugriffe auf den Webshop und andere Ereignisse aus dem laufenden Betrieb protokolliert, um etwaige Störungen oder Probleme frühzeitig erkennen und beheben zu können. Edi ist auch dafür verantwortlich, dass regelmäßig Sicherheitskopien der Datenbank erstellt werden.

1.4 Übungen

MHP
A PORSCHE COMPANY

Aufgabe 1.2 Anwendungsbeispiel

a) Welche sozialen Komponenten des beschriebenen Informationssystems sind in diesem Beispiel genannt? Welche Rolle(n) nehmen diese ein?

b) Welche Software-Produkte werden eingesetzt? Zu welcher Kategorie von Software gehören diese?

c) Welche Ziele verfolgt die Hoske GmbH mit dem Einsatz dieses Webshops?

d) Wie und wodurch der Webshop zum Unternehmenserfolg der Hoske GmbH beitragen?

e) Welche Risiken gibt es dabei?

1.4 Übungen

MHP
A PORSCHE COMPANY

Aufgabe 1.3 Mitwirkung von Wirtschaftsingenieuren

Ihre Mitwirkung an Informationssystemen: Als Wirtschaftsingenieurin oder Wirtschaftsingenieur kommen Sie in Ihrem Berufsleben ebenfalls mit Informationssystemen in Berührung. Dabei können Sie verschiedene Rollen einnehmen: als Benutzer, als Anwender und auch als Entwickler.

Beschreiben Sie je Rolle ein konkretes, plausibles Beispiel für eine Situation oder Aufgabenstellung, in der ein Wirtschaftsingenieur an einem Informationssystemen mitwirkt.

Agenda

MHP
A PORSCHE COMPANY

1. Informationssysteme
2. **IT im Unternehmen**

2.1.1 Standardsoftware und Individualsoftware

Jede Software muss entwickelt werden.

Anwender können entscheiden, ob sie bereits fertig entwickelte Software von Software-Anbietern beziehen oder eigene Software selbst entwickeln oder entwickeln lassen.

Standardsoftware

Fertig zu beziehende Software, die von vielen Anwendern sinnvoll eingesetzt werden kann, bezeichnet man als Standardsoftware.

Individualsoftware

Software, die ein Anwender selbst entwickeln lässt, bezeichnet man als Individualsoftware.

2.1.2 Entscheidung für Individual- oder Standardsoftware

MHP
A PORSCHE COMPANY

Standardsoftware

- Einsatzgebiet: allgemein übliche Aufgaben, die auf übliche Weise bearbeitet werden
- vorgefertigte Lösungen von Hersteller und Anbieter, zu dessen Kernkompetenzen die Entwicklung von Software zählt
- technisch auf aktuellem Stand
- Software erfüllt alle rechtlichen und sonstigen Vorschriften jederzeit
- Für einen breiten Anwenderkreis geeignet
- verfügt meist über einen wesentlich größeren Funktionsumfang als der einzelne Anwender tatsächlich nutzt

Individualsoftware

- sehr spezielle Aufgaben
- keine vorgefertigten Lösungen
- Für spezielle, einzigartige Vorgehensweisen
- passgenaue Software, aber mit einer Fehlern in der Anlaufphase
- alleiniges Recht am Quellcode durch das erstellende Unternehmen

In Unternehmen ist meist ein Mix aus Individual- und Standardsoftware im Einsatz.

2.1.2 Vor- und Nachteile von Standardsoftware

Vorteile Standardsoftware:

- Kostengünstig
- Zeitersparnis in der „Entwicklung“
- Personal- und Knowhow auch außerhalb des Unternehmens vorhanden
- „Bessere“ Qualität als bei Eigenentwicklung durch Erfahrung der Entwickler
- Zukunftssicherheit durch Pflege und Wartung

Nachteile Standardsoftware:

- Features stimmen nur teilweise mit den Anforderungen überein
- Innerbetriebliche Ablauforganisation muss ggf. an die Standardsoftware angepasst werden
- Schnittstellenprobleme zu anderen internen Anwendungen
- Notwendigkeit individueller Anpassungen
- Abhängigkeit von einem externen Software Anbieter

2.1.3 Customizing von Standardsoftware

- Standardsoftware enthält häufig Mechanismen, um sie an individuelle Gegebenheiten und Bedürfnisse anzupassen
- Anwender und Nutzer können Aussehen und Funktion beeinflussen
- Enthalten vorgesehene Optionen und Parameter
- Sind erweiterbar durch fertige Module mit Zusatzfunktionen (*Add-Ins, Plug-Ins*)

- **Customizing**
  Die Anpassung von Standardsoftware an individuelle Anforderungen mit Mitteln, die die Standardsoftware dafür vorsieht, bezeichnet man als Customizing.

  Customizing erfordert auch viel Aufwand und Wissen

- **Erweiterungs- oder Anpassungsprogrammierung**
  Standardsoftware durch individuell entwickelte Programmteile ergänzen beziehungsweise durch eigene Programmierung in den Programmcode der Standardsoftware einzugreifen und diese zu modifizieren.

2.2 Unternehmens-IT

MHP
A PORSCHE COMPANY

Unternehmens-IT

- Bezeichnet einerseits die **Gesamtheit der IT-Systeme** (Hardware und Software) des Unternehmens wie auch die **Organisationseinheit**, die dafür zu sorgen hat, dass diese Systeme zuverlässig bereitstehen und funktionieren.

- Diese Organisationseinheit ist oft als eine zentrale IT-Abteilung im Unternehmen organisiert.

- IT-Leiterin oder IT-Leiter werden auch Chief Information Officer **CIO** bezeichnet.

- Die Gesamtheit der IT-Systeme bezeichnet man auch als **IT-Landschaft**.

2.2.1 IT-Landschaft

- Hardware- und Software-Komponenten.
- Großunternehmen setzen oft mehrere tausend verschiedene Softwaresysteme ein.
- Typischerweise wächst die IT-Landschaft in einem Unternehmen im Laufe der Zeit
- Andere Systeme bleiben dagegen parallel zu den neuen Systemen im Einsatz, weil sie Spezialfunktionen bieten oder nötig sind, um auf die alten Datenbestände zuzugreifen. Solche Alt-Systeme werden auch als „**Legacy-Systeme**“ bezeichnet.

Herausforderungen:

- IT-Landschaft immer unübersichtlicher und schwerer zu warten
- Daten, die eigentlich zusammengehören, werden in mehreren Anwendungssystemen bearbeitet und in mehreren Datenbanken gespeichert
- Die gedoppelten Daten sind anfällig dafür, dass sich Inkonsistenzen einschleichen, und lassen sich nur umständlich auswerten.

IT-Strategie

Legt explizit fest, wie die Unternehmens-IT weiterentwickelt werden soll, um die Geschäftsziele bestmöglich zu unterstützen. Schlagwörter in diesem Zusammenhang sind **Business-IT-Alignment**, **Enterprise Architecture Management**, **IT-Governance** und **IT-Bebauungsplan**.

2.2.2 Aufgaben der Unternehmens-IT

- Beschaffung und Einführung
- Administration
- Support und Anwendungsberatung
- Strategische Entwicklung
- beobachten die technologische Entwicklung
- treiben IT-Innovation im Unternehmen voran

Ziele:

- Wirtschaftlichkeit und Kosteneffizienz bei der Bereitstellung von IT und IT-Diensten
- Business-IT-Alignment
- IT-Sicherheit
- IT-Compliance
- Integrierte Informationsverarbeitung
- Identifikation und Abschaffung von „Insellösungen“

Insellösung

Eine Anwendungssystem, das vom Datenaustausch mit den anderen Systemen in der IT-Landschaft abgekoppelt ist, nennt man eine „Insellösung“.

2.3.1 Schatten-IT: Begriffserklärung

- **Schatten-IT**

  Parallele IT-Strukturen, die Fachabteilungen selbständig einrichten und betreiben, um ihre Aufgaben besser zu unterstützen.

  Nicht in Unternehmens-IT einbezogen.

  Weitere Begriffe: „graue IT“ oder *rogue IT* (ein *rogue* ist ein Schlingel oder Schlawiner)

  Befragung 2014:

  „über 75% von knapp 1000 befragten IT-Entscheidern an, dass es in ihrem Unternehmen Schatten-IT gibt (BT, 2014).“

Gründe:

- Unternehmens-IT kann Anforderungen und Bedürfnisse der Anwender in den Fachabteilungen nicht erfüllen oder nicht schnell genug erfüllen
- Unabhängigkeit der Fachabteilung zeigen und stärken

2.3.2 Formen

- Anwendungen auf Basis von Tabellenkalkulationen (MS Excel) oder Desktop-Datenbanken (MS Access)
- Von Abteilungen selbst entwickelt (komplexer Anwendungen programmiert)
- zunehmend Cloud-Dienste

Beispiele:

- Filesharing-Angebote (zum Beispiel *Dropbox*)
- Kommunikationsdienste wie Chats und Instant Messaging (zum Beispiel *Skype* und *Whatsapp*), Social Media (vor allem *Facebook*)
- Werkzeuge zur Terminvereinbarung (etwa *doodle*) und allgemeine Online-Umfragen(zum Beispiel *surveymonkey* oder *Google Docs Form)*
- Kanäle zur Verbreitung von Informationen in verschiedenen Medienformen (etwa *youtube* oder *vimeo*)

2.3.3 Vorteile

- Zeitgewinn der Fachabteilungen
- Tätigkeiten besser / schneller
- Systeme sehr genau an deren Bedürfnisse und Anforderungen angepasst
- Systeme können schnell und zielgerichtet weiterentwickelt werden
- Kann Schatten-IT die Unternehmens-IT entlasten, Impulse geben und sogar als Innovationstreiber wirken!

2.3.4 Probleme und Risiken durch Schatten-IT

- Keine zentrale Kontrolle und Steuerung durch dafür ausgebildete und verantwortliche Personen
- Mangelnde Informationssicherheit und Compliance
- Bildung einer standardisierter und integrierter IT im Unternehmen verhindet
- schwer mit anderen Systemen integrieren (Insellösung)
- schwache Qualität, Fehler oder unterstützen die Aufgaben unzureichend, so dass Mitarbeiter nicht effizient arbeiten
- Verschiedene Fachabteilungen entwickeln und pflegen parallel ähnliche
- Leistung des Gesamtunternehmens wird geschwächt

MHP
A PORSCHE COMPANY

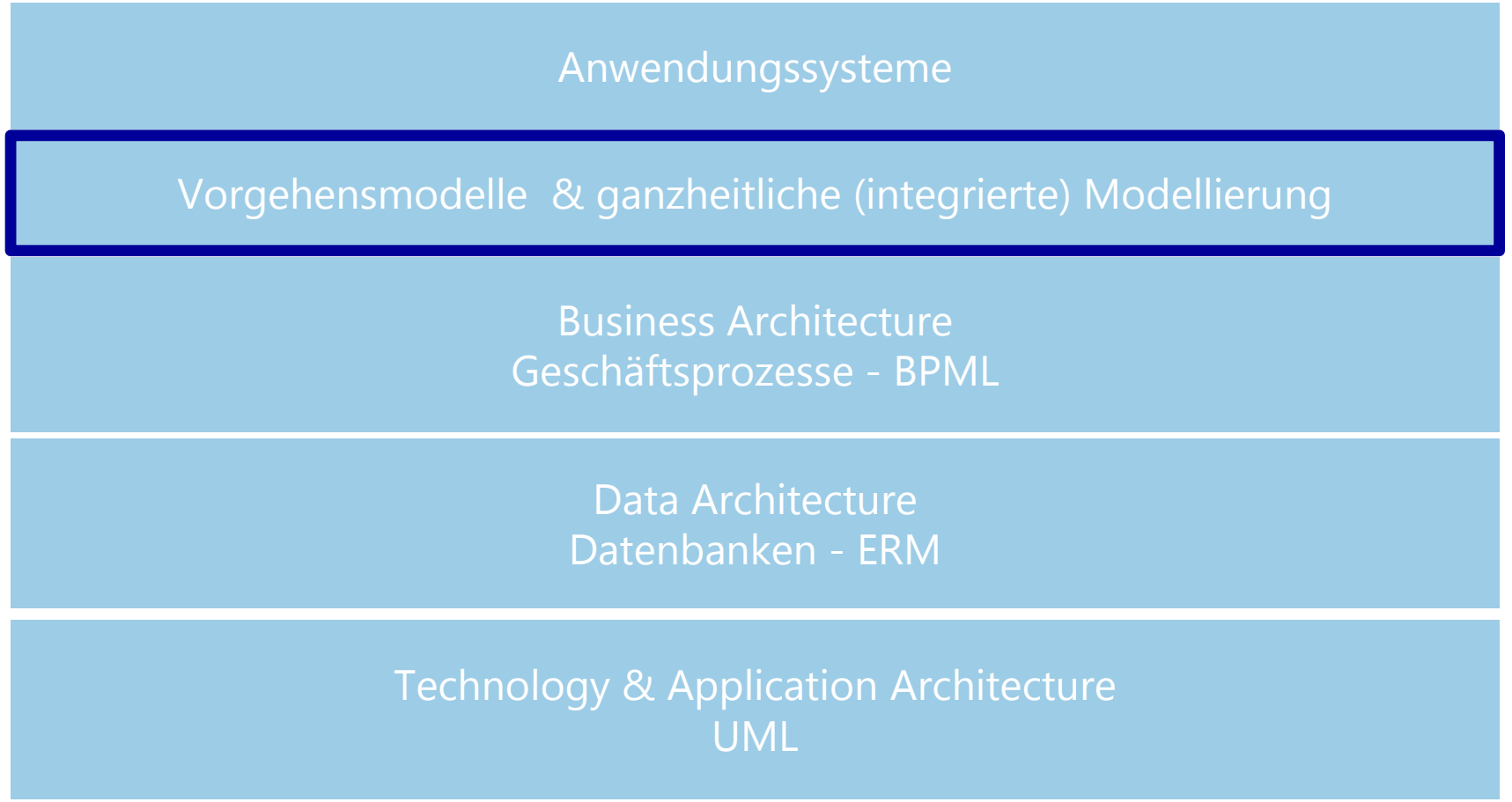
INTEGRIERTE MODELLIERUNG KOMPLEXER SYSTEME

Vorgehensmodelle

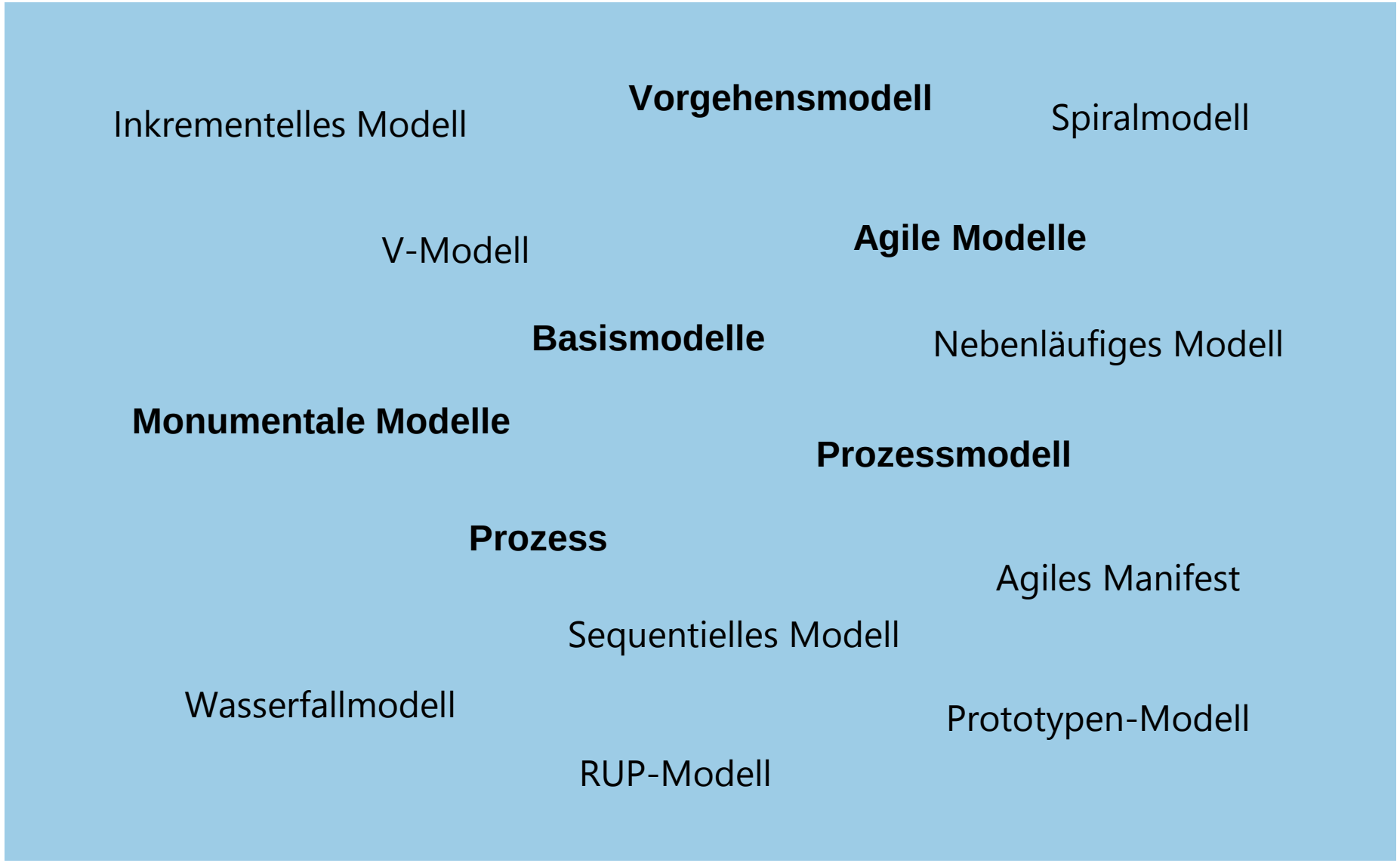
UNIVERSITATEA BABEȘ-BOLYAI

Hugo Colceag I Bachelor Studiengang Informatik

Vorlesungsinhalte und Aufbau



Mindmap - Anwendungssysteme und UML



Lernziele

- ✓ Unterschiedliche Vorgehensmodelle sehen und verstehen
- ✓ Vor-/Nachteile der Modelle sowie unterschiedliche Einsatzmöglichkeiten verstehen

Agenda

MHP
A PORSCHE COMPANY

1. **Begriffe**
2. Basismodelle
3. Monumentale Modelle
4. Agile Modelle
5. Vergleich und Trends

Prozess, Prozessmodell und Vorgehensmodell

- Ein **Prozess** ist ein **Arbeitsablauf zur Erledigung einer Aufgabe**.

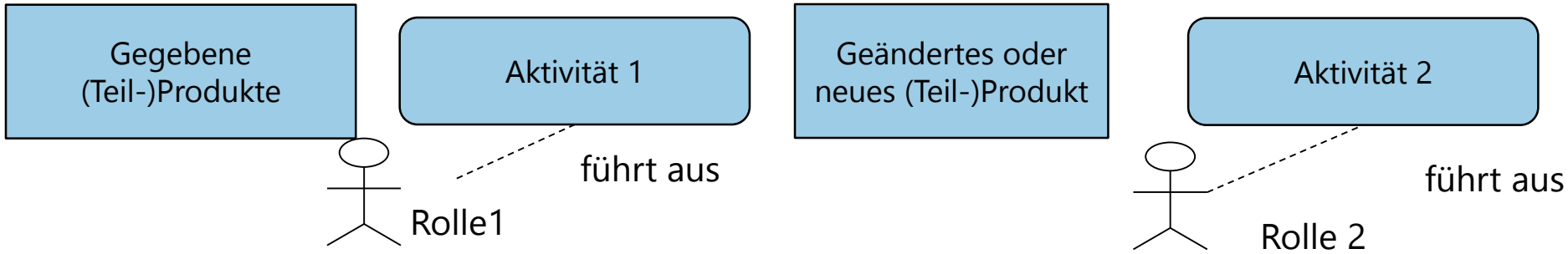
Ein **Prozessmodell** beschreibt in abstrakter und idealisierter Form die **zeitlich-sachlogische Abfolge von Aktivitäten für einen Prozess**.

Ein Prozessmodell, das die **Abfolge von Aktivitäten** für die **SW-Entwicklung** beschreibt, wird auch **Vorgehensmodell** genannt.

- Bei bestimmten Vorhaben (Projekten) haben sich bestimmte Vorgehen als besonders zielführend erwiesen.
- Beispiel Hausbau: Bevor mit dem Ausheben der Baugrube begonnen und die Wände gemauert werden, ist es zielführend, zuerst Pläne anzufertigen, welche das Baufen-ster aus dem Bebauungsplan und die Wünsche des Bauherrn berücksichtigen. Dies stellt sicher, dass das Haus nicht mehrfach abgerissen und neu gebaut werden muss.

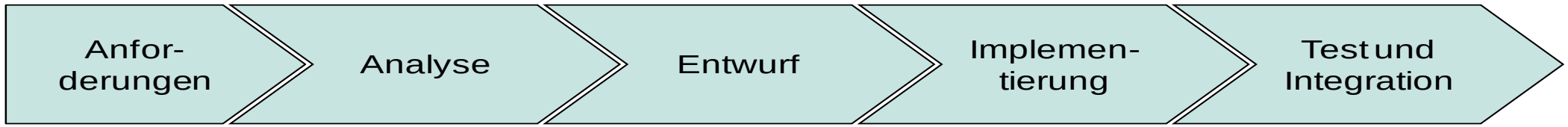
Bewährte **Vorgehensweisen für die Projektabwicklung** werden in **Vorgehensmodellen** dokumentiert.

Notation für Prozess- und Vorgehensmodelle



- Das Basiselement eines Softwareprozesses beschreibt eine **Aktivität**,
  - für die eine **Rolle** (z. B. System-Analytiker, System-Architekt, Projektleiter, Entwickler, Tester, etc., keine konkreten Personen) verantwortlich ist,
  - das Ziel einer Aktivität ist es, ein **Produkt fertig zu stellen**,
  - für die Erstellung der Produkte können **Werkzeuge** zum Einsatz kommen,
  - eine Aktivität wird **innerhalb** einer **Entwicklungsphase** durchgeführt.

- Mehrere Aktivitäten werden in (Entwicklungs-)Phasen zusammengefasst:



Abgrenzungen

- Abgrenzung **Vorgehensmodell** und **Methode**:
  - Im Gegensatz zu Methoden umfassen Prozessmodelle die gesamte Abwicklung einer Softwareerstellung oftmals einschließlich unterstützender Aktivitäten wie Konfigurations-, Änderungs- und Projekt- und Qualitätsmanagement.
  - Methoden beziehen sich in der Regel auf detaillierte Vorgehensweisen in den Kernbereichen der Entwicklung (Beispiel: Objektorientierte Methode für die Aktivitäten Analyse und Entwurf).

- Abgrenzung **Vorgehensmodell** und **Qualitätsmodell**:
  - Produkte eines Vorgehensmodells sind der Qualitätssicherung unterstellt.
  - Zwischen Prozess- und Qualitätsmodellen gibt es starke Abhängigkeiten
    → Qualität muss durch den Entwicklungsprozess sichergestellt werden!

Qualität ist nicht automatisch in einem Softwareprodukt, sondern sie muss in das Produkt hinein entwickelt werden. Wie das zu geschehen hat, wird durch **Qualitätsmodelle** festgelegt.

Agenda

MHP
A PORSCHE COMPANY

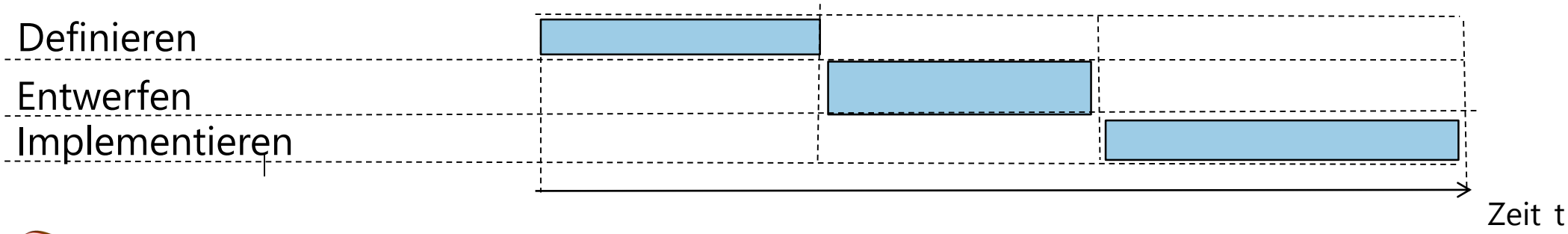
1. Begriffe
2. **Basismodelle**
3. Monumentale Modelle
4. Agile Modelle
5. Vergleich und Trends

Klassifizierung von Vorgehensmodellen

- **Basismodelle**
  - Vorgehensmodelle auf Projektebene
  - Beispiele: Sequentielles Modell, nebenläufiges Modell, V-Modell, Spiralmodell, etc.
- **Monumentale Modelle** (schwergewichtige Prozesse)
  - Detaillierte Beschreibungen, wie Prozess- und Qualitätsziele zu erreichen sind – genaue Beschreibung von Prozessschritten sowie Struktur und Aufbau zu erstellender Dokumente.
  - Beispiele: V-Modell XT, RUP.
- **Agile Modelle** (leichtgewichtige Prozesse)
  - Gegenbewegung zu den immer schwieriger zu durchschauenden und anzuwendenden monumentalen Modellen.
  - Beispiele: Extreme Programming, Scrum.

Sequentielles Modell

- SW-Entwicklung in Phasen (z. B. Definieren, Entwerfen, Implementieren) gliedern.
- Phasen sequentiell nacheinander abarbeiten → serielle Entwicklung.
- Phase beginnt erst dann, wenn die Vorgängerphase vollständig abgeschlossen ist.



Welche Voraussetzungen müssen erfüllt sein, damit das sequentielle Modell erfolgreich eingesetzt werden kann?

Problem: es gibt keine Möglichkeit Änderungen einzubringen → nur geeignet wenn,

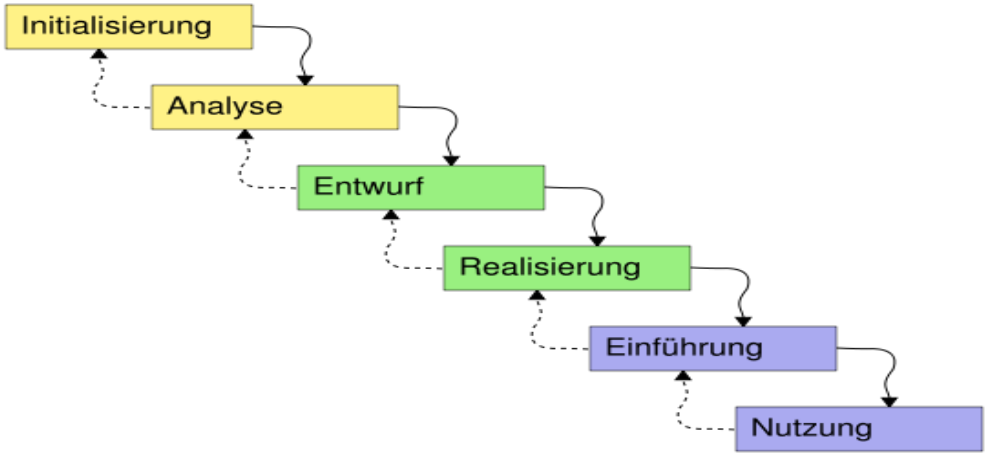
- Teams in der Lage sind, gute, endgültige Definitionen und Entwürfe zu erstellen.
- Auftraggeber in der Lage ist, die Anforderungen vollständig und endgültig vorzugeben.

Wasserfallmodell

- Bekannteste Ausprägung des sequentiellen Modells.
- Erweitert das sequentielle Modell um Rückkopplungsschleifen zur Vorgänger-Phase.

Charakteristika

- Jede Phase ist in der richtigen Reihenfolge und vollständig durchzuführen.
- Am Ende jeder Phase existiert ein Dokument.
- Jede Phase muss beendet sein, bevor die nachfolgende Phase beginnt.
- Kunden/Auftraggeber-Beteiligung nur in der Definitionsphase.

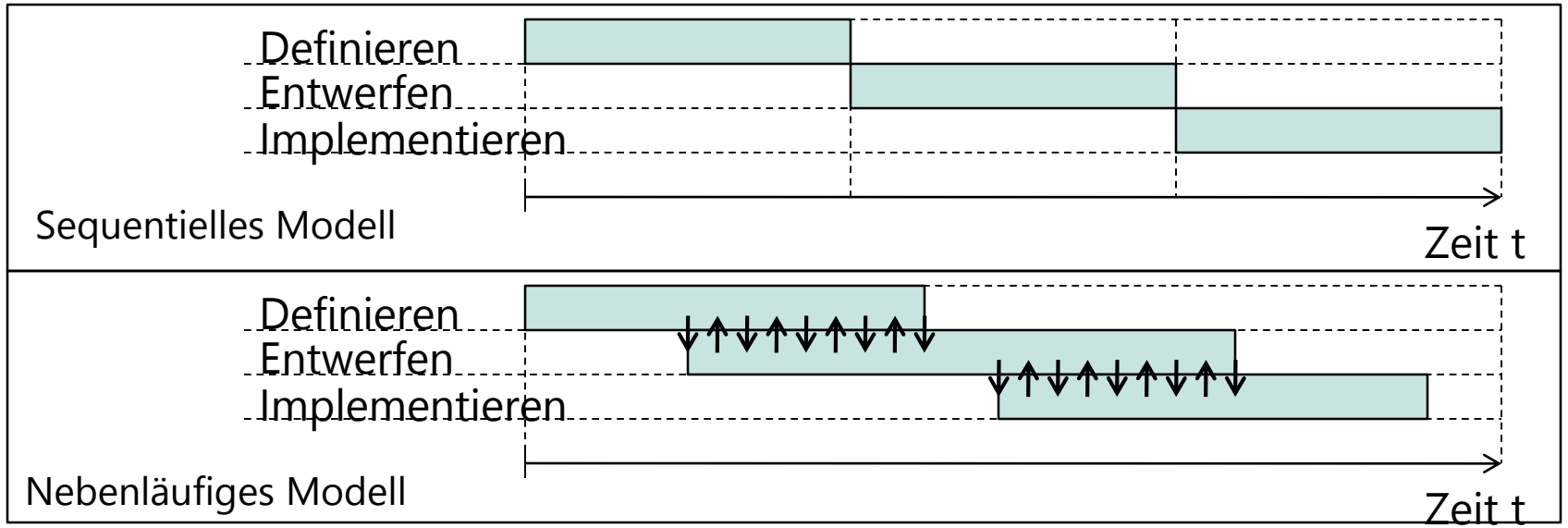


Welche Voraussetzungen müssen erfüllt sein, damit das Wasserfallmodell erfolgreich eingesetzt werden kann?

Problem: Kunde kann keine nachträglichen Änderungen einbringen → nur geeignet wenn, Auftraggeber in der Lage ist, die Anforderungen vollständig und endgültig vorzugeben.

Nebenläufiges Modell

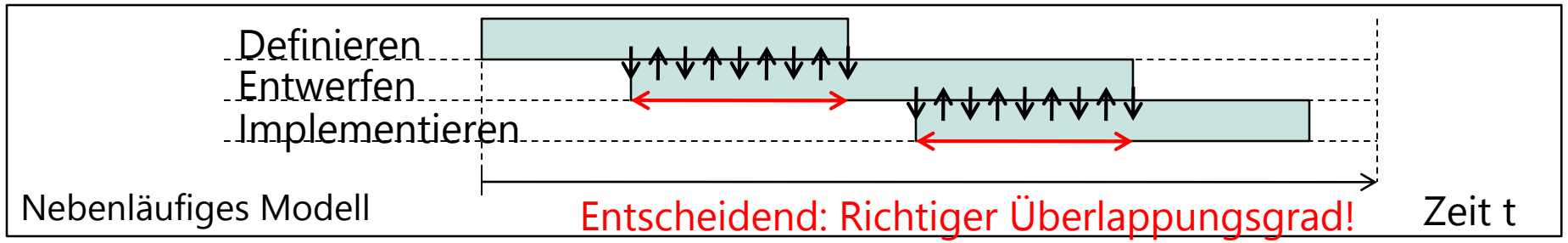
- **Ziel**: gesamte Entwicklungszeit reduzieren.
- Eine Phase muss nicht abgeschlossen sein, um die Nachfolgephase zu bearbeiten.
- Es gibt Rückkopplungen zwischen den Phasen.
- Notwendige Überarbeitungen, welche die Kosten erhöhen, werden in Kauf genommen.



Was ist der entscheidende Erfolgsfaktor beim nebenläufigen Modell?

Nebenläufiges Modell – Erfolgsfaktor und Einsatzbereiche

- Erfolgsfaktor: Den richtigen Überlappungsgrad finden.
  - Wird zu früh mit der Nachfolgephase begonnen, sind die Ergebnisse der Vorgängerphase noch nicht stabil genug → zusätzliche Überarbeitungsschleifen.
  - Wird zu lange mit dem Beginn der Nachfolgephase gewartet, verlängert sich die Zeit für die Fertigstellung des Produktes.



- Einsatzbereiche:
  - Wenn die Anforderungen bei Projektstart noch nicht alle bekannt sind.
  - Wenn die time-to-market-Anforderungen hoch sind.

Inkrementelles Modell

Gemeinsamkeiten sequentielles Modell, Wasserfallmodell und nebenläufiges Modell:

- **Ziel:** Vollständiges SW-Produkt in einem Durchlauf durch die Phasen erstellen.
- **Voraussetzung:** Alle Anforderungen des Kunden können zu Beginn ermittelt werden.
- **Konsequenz:** Kunde erhält das Produkt erst nach Abschluss der Implementierung.
- **Risiko:** Kunde stellt nach Projektende fest, dass er etwas ganz anderes wollte.

Inkrementelles Modell reduziert dieses Risiko durch:

- **Möglichst** vollständige Erfassung der Anforderungen zu Beginn.
- **Zerlegung** des zu erstellenden Produktes in seine Bestandteile.
- Entwerfen, Implementieren und Ausliefern des 1. Bestandteils.
- Entwerfen, Implementieren und Ausliefern des 2. Bestandteils.
- etc.

Welche Zerlegungsmöglichkeiten kennen Sie für ein Produkt?

Prototypen-Modell

Gemeinsamkeiten der bisherigen Modelle:

- (Fast) vollständige Spezifikation der Anforderungen ganz zu Beginn.
  Praxis: Kunde und Endnutzer sind dazu oft nicht in der Lage.
- Entwickler und Endnutzer arbeiten nur zum Aufstellen der Anforderungen zusammen.
  Praxis: Entwickler und Endnutzer müssen kontinuierlich voneinander lernen.
- Einbeziehung des Kunden zur Auswahl verschiedener Lösungen ist nicht vorgesehen.
  Praxis: Für manche Anforderungen müssen die Lösungsmöglichkeiten erst experimentell erprobt und dann mit dem Auftraggeber diskutiert werden.

Prototypen-Modell berücksichtigt diese Praxisanforderungen!

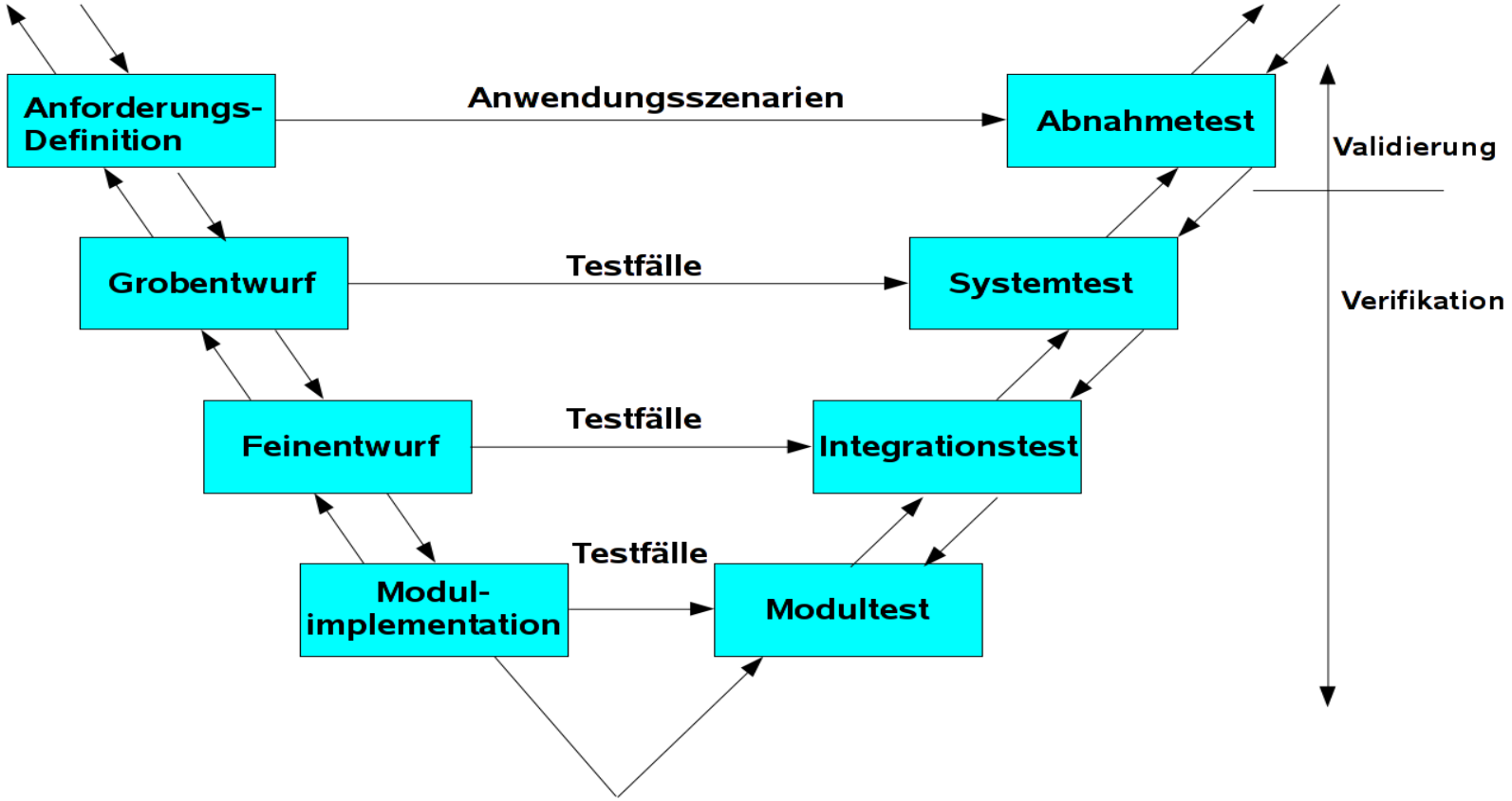
Prototypen werden im Wesentlichen für folgende Zwecke erstellt:

- Relevante Anforderungen/Entwicklungsprobleme klären → **Machbarkeitsanalyse**.
- Diskussionsbasis zur Ermittlung aller Anforderungen → **Wegwerf-Prototyp**.
- Prototypen als Ausgangsbasis für eine inkrementelle Weiterentwicklung.

Kennen Sie den Unterschied zwischen **horizontalen** und **vertikalen Prototypen**?

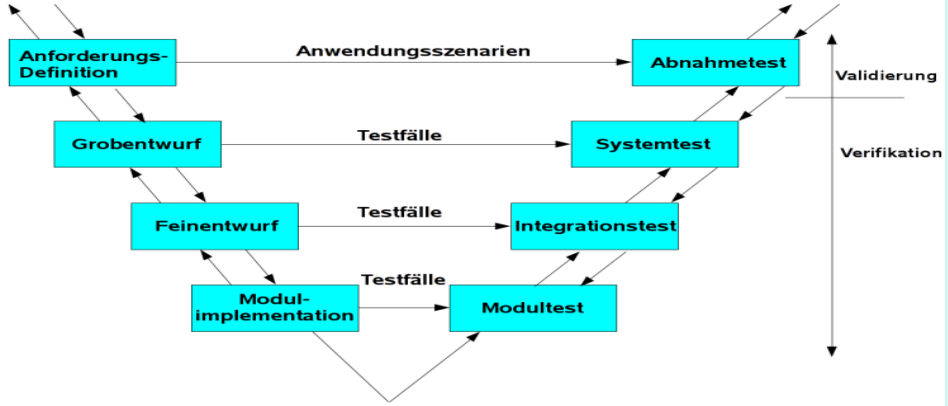
V-Modell

- Erweiterung des Wasserfallmodells → sequentielles Modell.
- Integration von Qualitätssicherungsmaßnahmen.



V-Modell – Validierung und Verifikation

Was ist der Unterschied zwischen Validierung und Verifikation?



Verifikation:

- „Wird ein korrektes Produkt im Sinne der Spezifikationentwickelt?“
- Überprüfung der Übereinstimmung zwischen SW-Produkt und Spezifikation.

Validierung:

- „Wird das richtige Produkt entwickelt?“
- Eignung bzw. der Wert eines Produktes bezogen auf seinen Einsatzzweck.

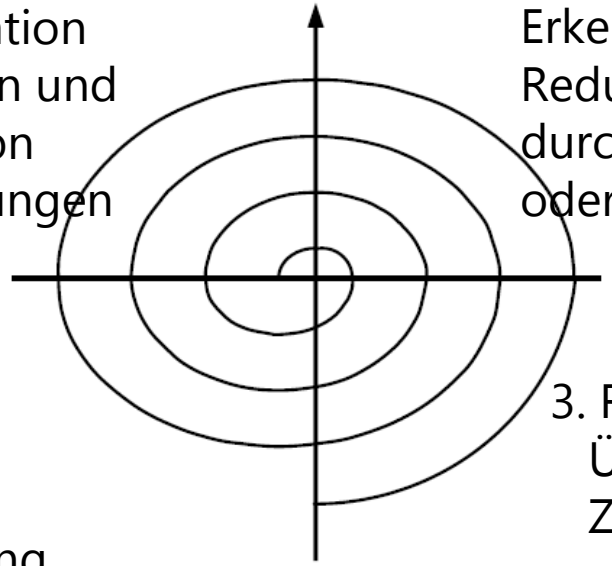
Das V-Modell integriert Maßnahmen zur Qualitätssicherung. Neben den Aktivitäten und zu erstellenden Produkten für die Software-Erstellung wird ferner das Konfigurations- und Projektmanagement geregelt.

Spiralmodell

- Erlaubt das Mischen von verschiedenen Vorgehensmodellen.
- Für jeden Iterationszyklus kann ein anderes Vorgehensmodell eingesetzt werden.
- Der Fokus liegt auf dem Risikomanagement.

1. Festlegung von Zielen, Identifikation von Alter-nativen und Beschreibung von Rahmenbedingungen

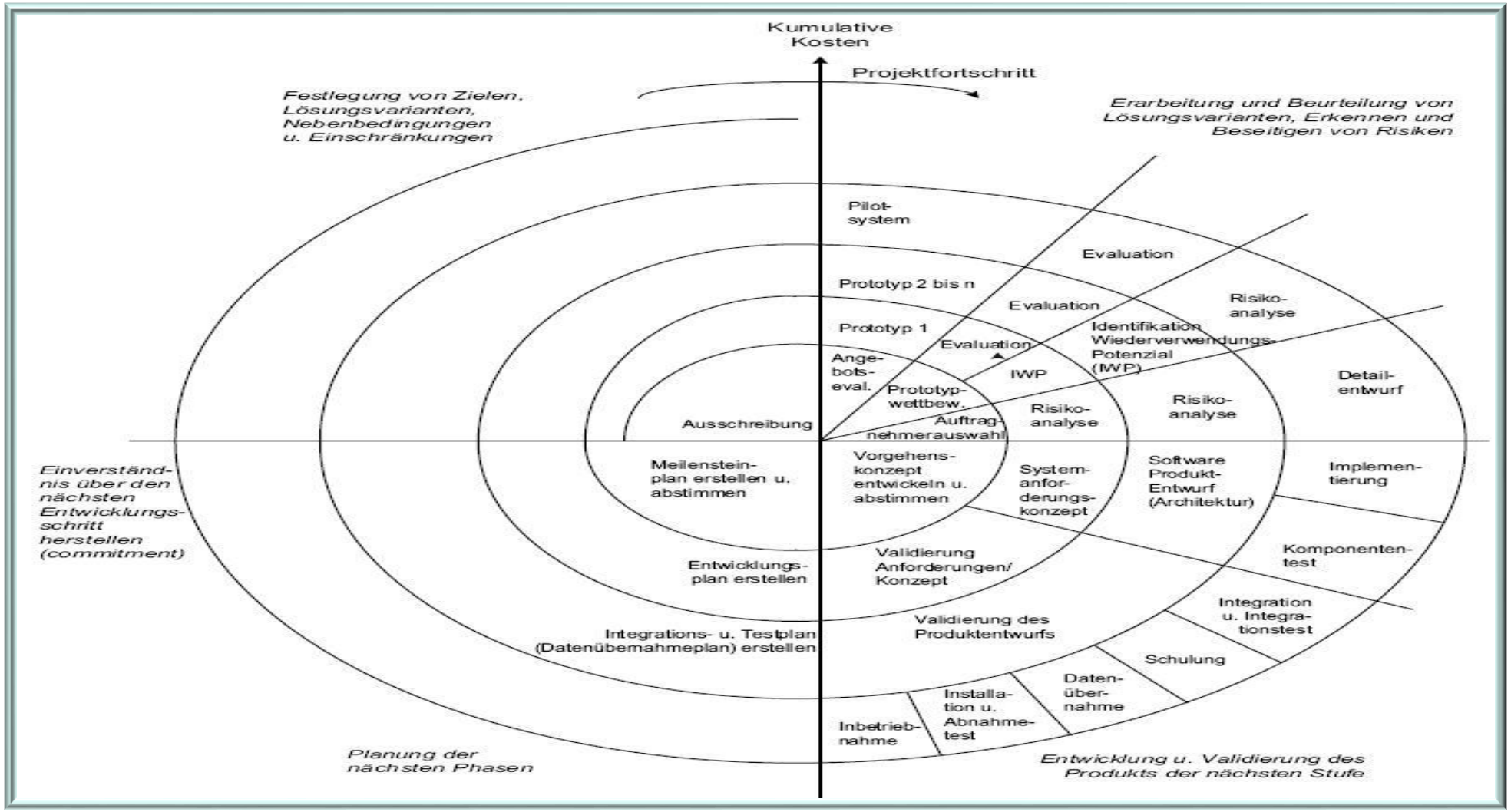
2. Evaluierung der Alternativen und Erkennen, Abschätzen und Reduzieren von Risiken, z. B. durch Analysen, Simulationen oder Prototyping.



3. Realisierung und Überprüfung des Zwischenprodukts.

4. Planung des nächsten Zyklus der Projektfortsetzung

Beispielhafte Ausprägung eines Spiralmodells



Vergleich der vorgestellten Basismodelle

MHP
A PORSCHE COMPANY

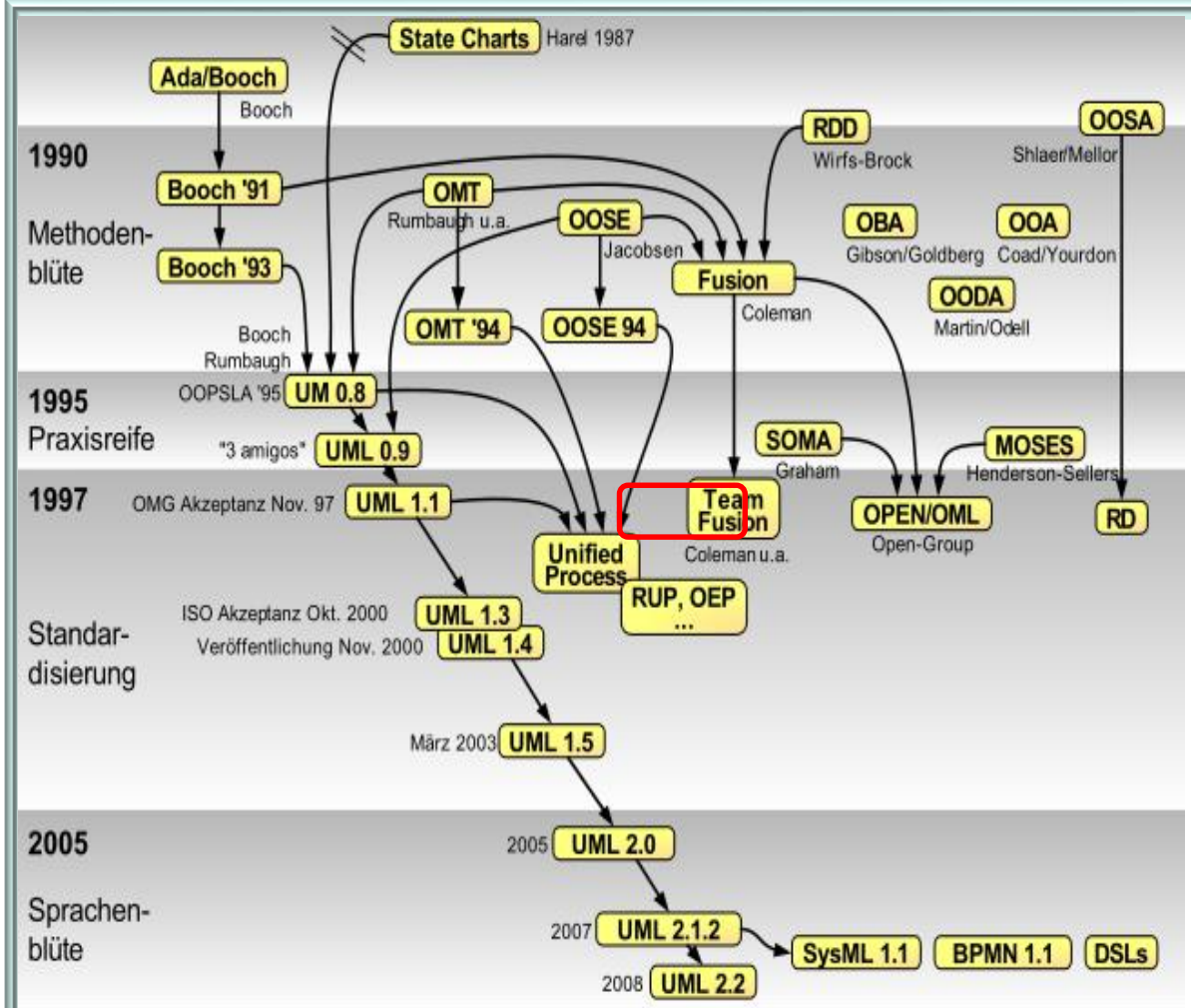
Prozessmodell	Primäres Ziel	Antreibendes Moment	Kunden-einbindung	Charakteristika
Sequentielles Modell	Minimaler Ma-nagementaufwand	Dokumente	gering- nur zu Beginn	sequenziell, volle Breite*
Nebenläufiges Modell	Minimale Entwicklungszeit (fast-to-market)	Zeit	gering - nur zu Beginn	volle Breite*, nebenläufig
Inkrementelles Modell	minimale Entwicklungszeit (fast-to-market) Risikominimierung	Code	mittel – zu Beginn und nach jedem Inkrement	möglichst volle Definition, dann zunächst nur Kernsystem
Prototypen-Modell	Risikominimierung	Code	hoch	nur Teilsysteme (horizontal oder vertikal)
V-Modell	maximale Qualität	Dokumente	gering - nur zu Beginn	sequenziell, volle Breite*, Validation, Verifikation
Spiralmodell	Risikominimierung	Risiko	mittel	Entscheidung pro Zyklus über weiteres Vorgehen

Agenda

MHP
A PORSCHE COMPANY

1. Begriffe
2. Basismodelle
3. **Monumentale Modelle**
4. Agile Modelle
5. Vergleich und Trends

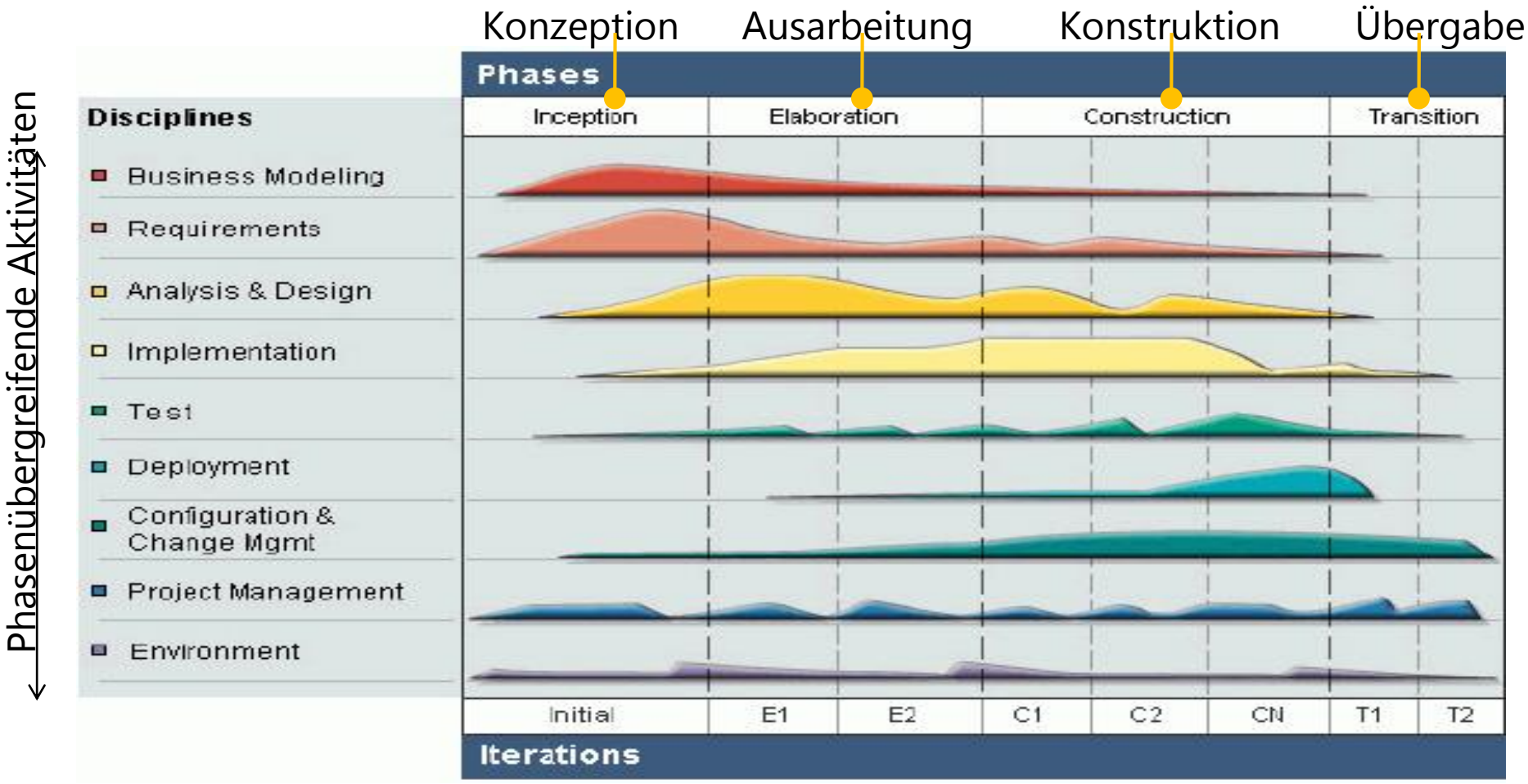
Rational Unified Process – das RUP-Modell



Rational Unified Process (RUP)

- Von der Firma Rational Software, heute IBM
- monumentales Vorgehensmodell für die objektorientierte SW-Entwicklung.
- Entwicklung von RUP ist eng verbunden mit der Entstehung der UML (Unified Modelling Language).
- Grady Booch, Ivar Jacobson und James Rumbaugh (die Ideengeber der UML) standen auch Pate für RUP.
- Der RUP wird durch das Werkzeug „IBM Rational Method Composer“ (integriert in Eclipse) unterstützt.

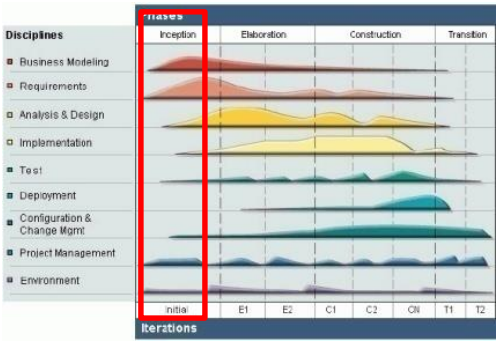
Das Prozessmodell des RUP



Beispiel für die Phase „Konzeption“ im RUP

MHP
A PORSCHE COMPANY

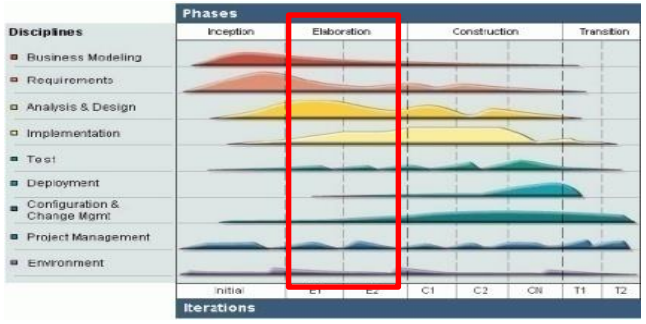
- **Zweck:** Erledigen von Vorarbeiten, um das Projekt planen und starten zu können.
- Aktivitäten und Ergebnisse in der Disziplin Geschäftsprozessmodellierung:
  - Identifizierung der Geschäftsprozesse
  - Identifizierung von Rollen und Verantwortlichkeiten
  - Modellierung der Domäne
  - Hauptergebnisse: Business Use-Case Model, Business Object Model
- Aktivitäten und Ergebnisse in der Disziplin Anforderungsanalyse:
  - Analyse des Problemfeldes
  - Bedürfnisse der Projektbeteiligten (stakeholders) verstehen
  - Grobe Anforderungen vollständig ermitteln (noch kein Pflichtenheft)
  - Erste grobe Planung und Schätzung für die restliche Zeit durchführen
  - Hauptergebnisse: Bedürfnisse der Projektbeteiligten, Use-Case-Modell, etc.
- Parallele Bearbeitung der Disziplinen Konfigurations- und Änderungsmanagement, Projektmanagement und Infrastruktur.
- **Ergebnisse nicht ausreichend** am Ende der Konzeptionsphase → **erneute Iteration**



Ziele und Ergebnisse der Phase „Ausarbeitung“ im RUP

- Use-Case-Modell und Anforderungen sind weitestgehend komplett (ca. 80%),
- für das Projekt notwendige Werkzeuge und Ressourcen sind festgelegt,
- Problembereich ist analysiert,
- stabile Architektur ist entwickelt,
- Hauptrisiken sind identifiziert und Qualitätsattribute sind festgelegt

- Architektur ist beschrieben und liegt in Form eines Architektur-Prototypen vor.
- Der Architektur-Prototyp ist kein „Wegwerf-Prototyp“ sondern die Basis für die weitere Systementwicklung.
- **Falls Ergebnisse nicht ausreichend → weitere Iteration!**



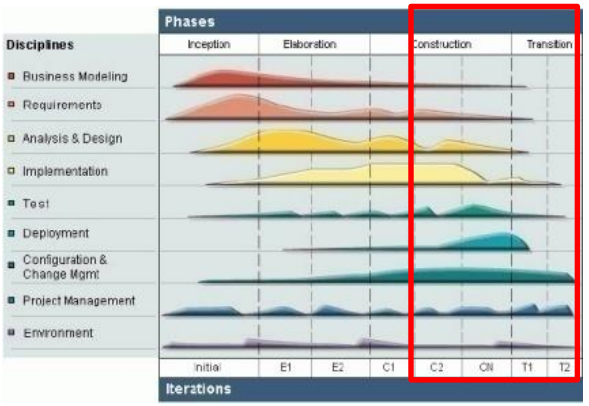
Phasen „Konstruktion“ und „Übergabe“ im RUP

Phase Konstruktion:

- Implementieren und Testen der Systemkomponenten.
- Erstellen der System- und Benutzerdokumentation.
- **Ergebnis:** Lauffähiges und für den Kunden einsetzbares Produkt.

Phase Übergabe:

- Übergang des Systems aus der Entwicklungsumgebung zum Kunden.
- Test des Produktes durch ausgewählte Kunden (Betakunden).
- Beseitigung von Fehlern.
- Schulung der Anwender.
- Fertigstellung von Benutzerhandbüchern.
- Eventuell Konvertierung von Daten aus älteren Systemen.
- **Ergebnis:** Produkt-Release, Projekt ist abgeschlossen und wird übergeben.



Agenda

MHP
A PORSCHE COMPANY

1. Begriffe
2. Basismodelle
3. Monumentale Modelle
4. **Agile Modelle**
5. Vergleich und Trends

Agiles Manifest – Werte

- Gegenbewegung zu den immer mächtiger werdenden monumentalen Modellen
- Agil = flink, wendig, beweglich
- **Agiles Manifest** (Kernideen agiler Modelle, 2001 publiziert, http://www.agilemanifesto.org)

Agiles Manifest – Prinzipien

Grundgerüst, für die Einhaltung der Werte in der Praxis --> Verhaltensregeln:

- Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
- Heisse Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
- Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
- Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
- Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
- Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteam zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.
- … (siehe http://www.agilemanifesto.org)

Agiles Manifest und agile Vorgehensmodelle

MHP
A PORSCHE COMPANY

Praktiken: bewährte praktische (**agile**) **Vorgehensmodelle**, die den Prinzipien gerecht werden:

Scrum	Extreme Programming	Kanban	Crystal	. . .

Prinzipien:

- Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
- Heisse Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
- ...

Werte:

- Individuen und Interaktionen mehr als Prozesse und Werkzeuge
- Funktionierende Software mehr als umfassende Dokumentation
- Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung
- Reagieren auf Veränderung mehr als das Befolgen eines Plans

Agiles Manifest

Agenda

MHP
A PORSCHE COMPANY

1. Begriffe
2. Basismodelle
3. Monumentale Modelle
4. Agile Modelle
5. **Vergleich und Trends**

Vergleich monumentale und agile Modelle

Monumentales Modell	Agiles Modell
Prozess vorhersagbar	Prozess adaptiv
prozessorientiert	menschen- und teamorientiert
formale Kommunikation	informelle Kommunikation
umfangreiche, formale Dokumentation	minimale, informelle Dokumentation
keine festen Zeitraster	oft feste Zeitraster (time box)
mittelmäßig qualifizierte, z. T. unmotivierte Entwickler	verantwortungsvolle & motivierte Entwickler
Kunden, die wenig mit der Entwicklung zu tun haben wollen	Kunden, die in die Entwicklung einbezogen werden wollen
Teamgröße > 50	Teamgröße < 50
Festpreisauftrag	Auftrag nach Aufwand

Viele Prozesse detailliert und formal	Wenige Prozesse
Schwergewichtige Prozesse	Leichtgewichtige Prozesse

Es lassen sich 4 Trends identifizieren

- **Monumentale Modelle werden abgemagert**,
  damit sie auch bei kleinen Projekten und in kleinen Teams wirtschaftlich und mit wenig Einarbeitungsaufwand eingesetzt werden können.
- **Agile Modelle werden erweitert**,
  um den Einsatz mit großen und/oder verteilten Teams zu ermöglichen.
- **Monumentale und agile Modelle koexistieren als hybride Modelle**.
  Innerhalb der monumentalen Modelle gibt es bei der eigentlichen Entwicklung Freiräume, welche die Anwendung agiler Modelle ermöglichen.

- **Monumentale und agile Modelle werden mit Rahmenmodellen kombiniert**.

MHP
A PORSCHE COMPANY

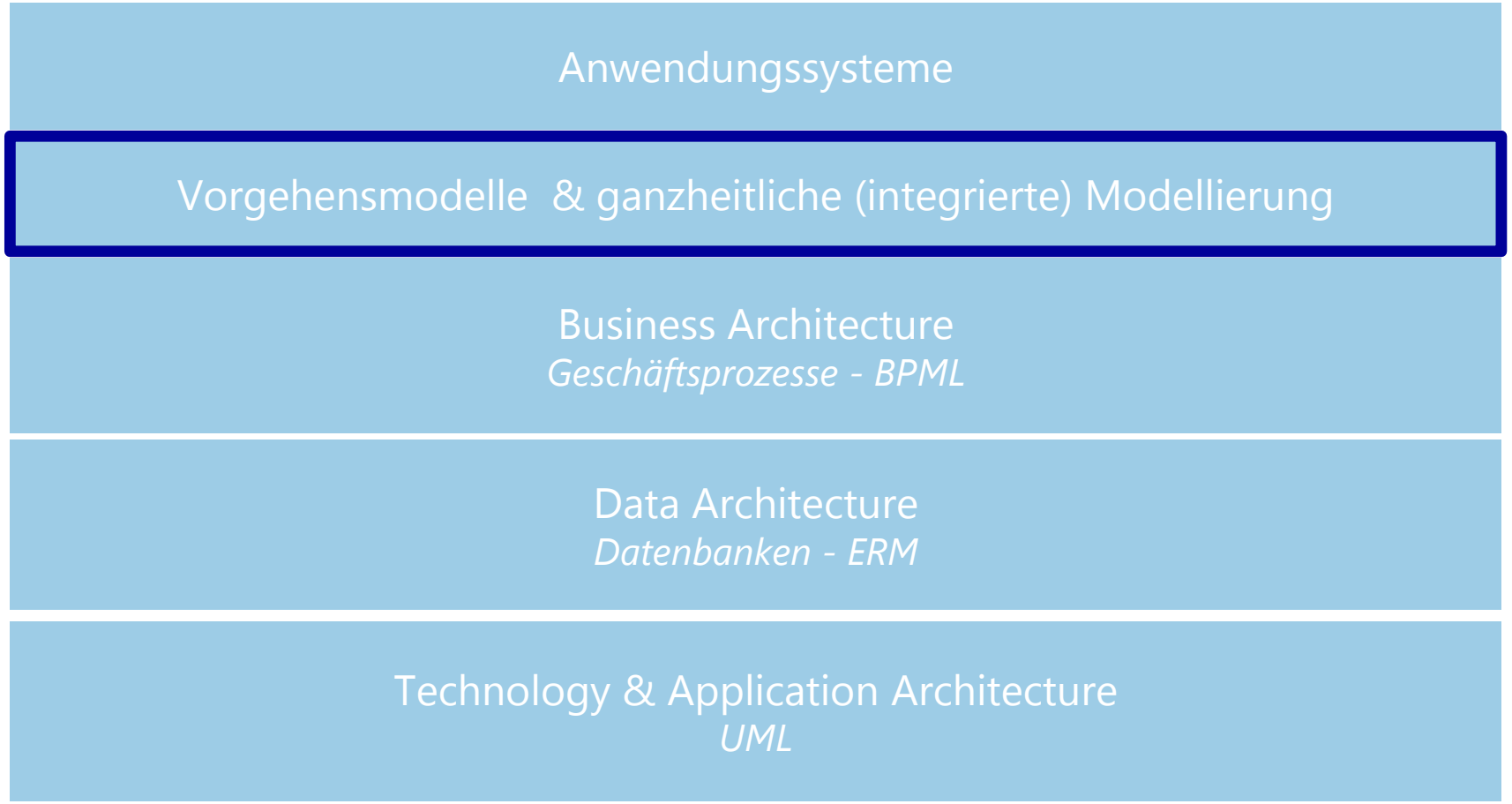
INTEGRIERTE MODELLIERUNG KOMPLEXER SYSTEME

Ganzheitliche Modellierung

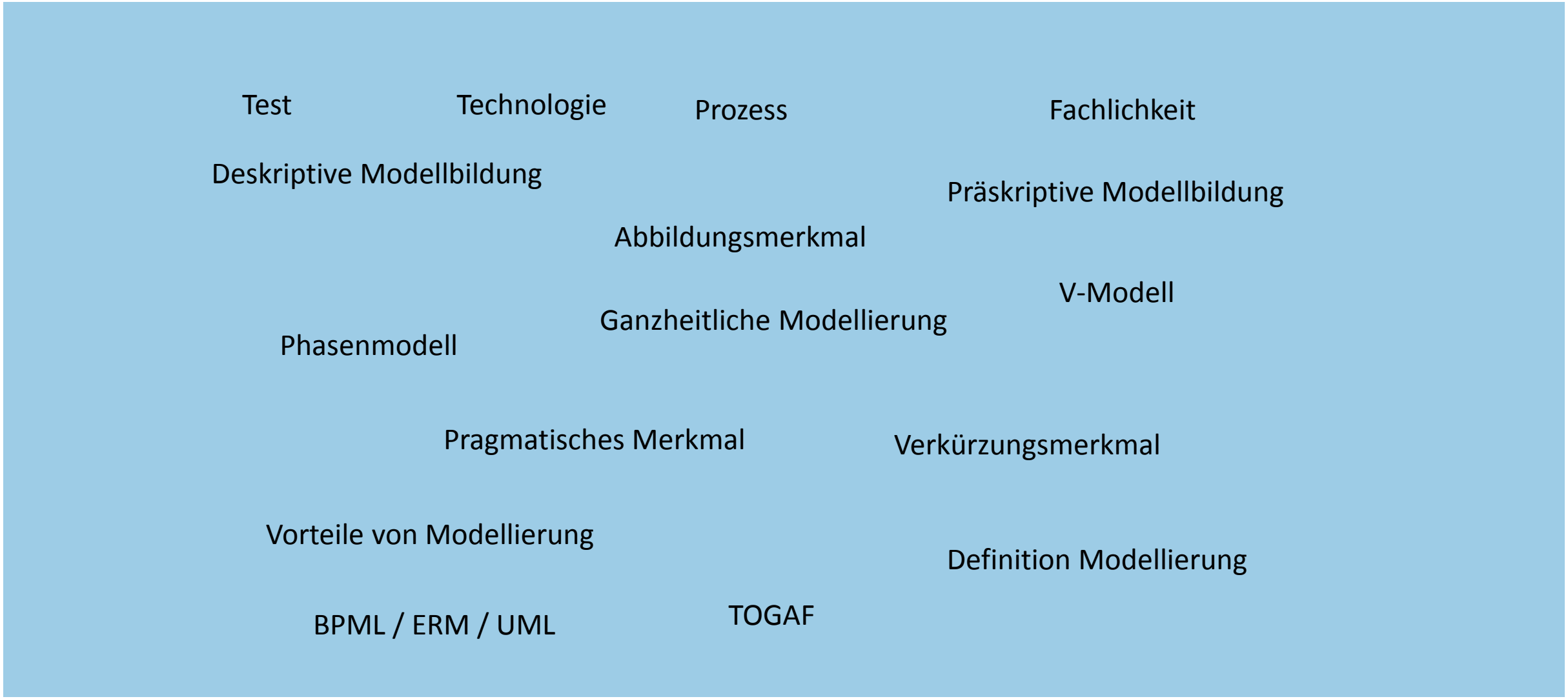
Hugo Colceag(r) l Bachelor Studiengang Informatik

UNIVERSITATEA BABEȘ-BOLYAI

Vorlesungsinhalte und Aufbau



Mindmap - Ganzheitliche Modellierung



Lernziele

- ✓ Grundlagen der Modellierung
- ✓ Vorteile der Modellierung
- ✓ Prinzipien der Modellierung

Agenda

MHP
A PORSCHE COMPANY

1. **Grundlagen der Modellierung**
   - Prinzipien der SW-Technik
   - Prinzip der Abstraktion
   - Prinzip der Bindung und Kopplung
   - Prinzip der Hierarchisierung
   - Geheimnisprinzip
   - (Softwareentwicklungs-) Werkzeuge

2. Einleitung Vorgehensweise ganzheitliche Modellierung

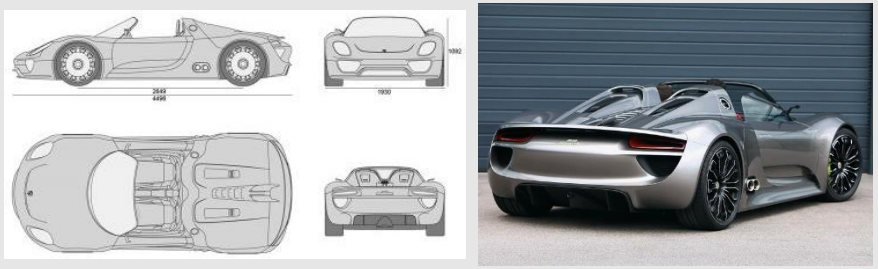
Motivation - Vorteile von Modellierung in Projekten

- ✓ Verständnis der fachlichen Anforderungen, der technischen Umgebung und der Randbedingungen der zu erstellenden Lösung
- ✓ Verbesserung der Kommunikation durch einfache und präzise Dokumentation, die zu jedem Zeitpunkt auf dem aktuellsten Stand ist
- ✓ Schaffung eines Überblicks und Steigerung der Effizienz und Qualität im Entwicklungsprozess
- ✓ Standardisierte Dokumentation
- ✓ Langfristige Kostenersparnis

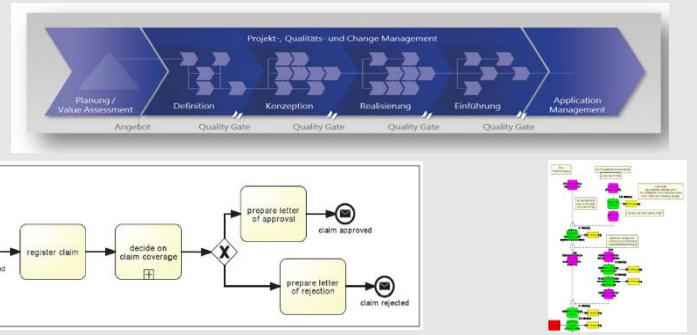
Definition Modellierung

- Modellierung ist die **vereinfachte Abbildung der Realität** mit dem Ziel eine **spezielle Eigenschaft** genauer darzustellen.

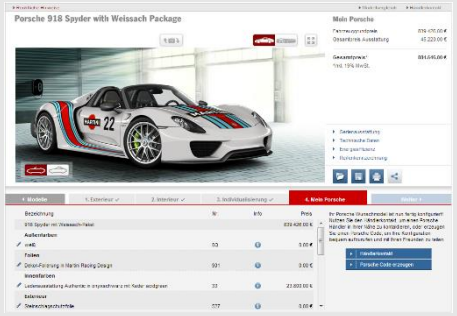
Alltägliche Modelle



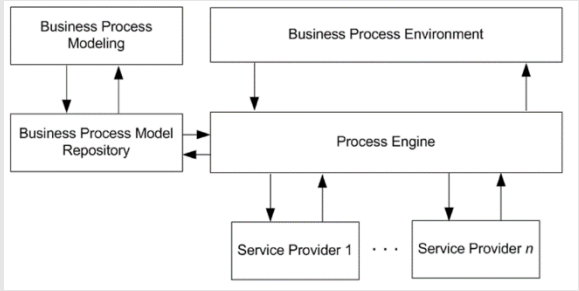
Geschäftsprozessmodelle



Modelle im Web



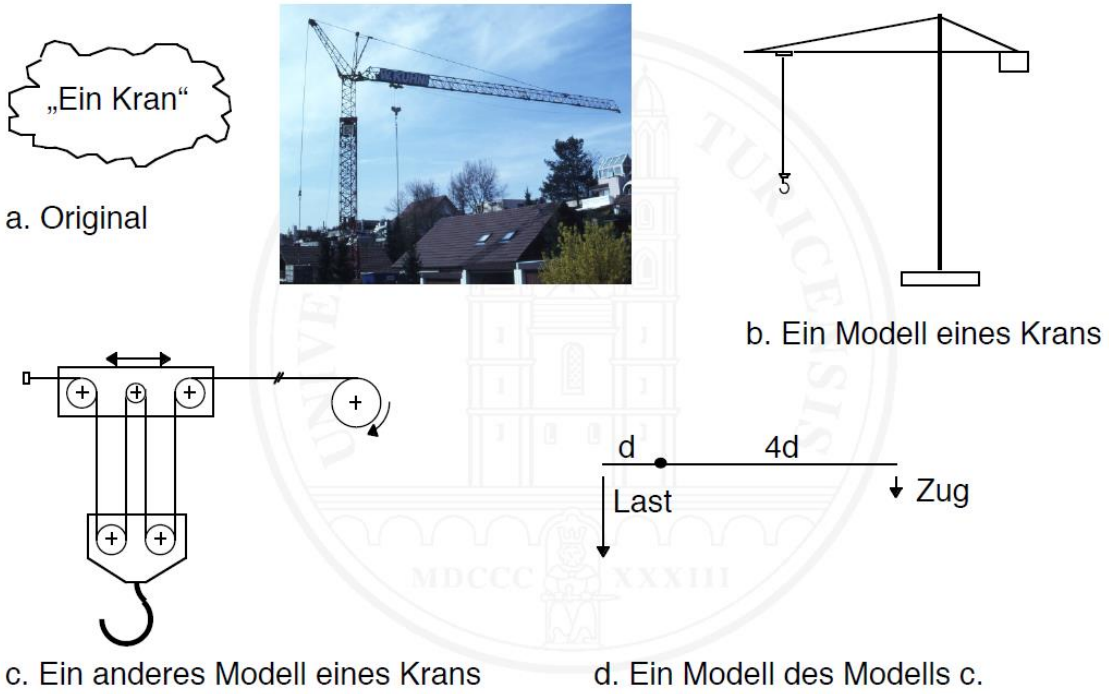
Architekturmodelle



Hauptmerkmale des Modellbegriffs nach H. Stachowiak [1973]

Abbildungsmerkmal

- „Modelle sind stets Modelle von etwas, nämlich Abbildungen, Repräsentationen natürlicher oder künstlicher Originale, die selbst wieder Modelle sein können."
- Jedes Modell ist Abbild oder Vorbild

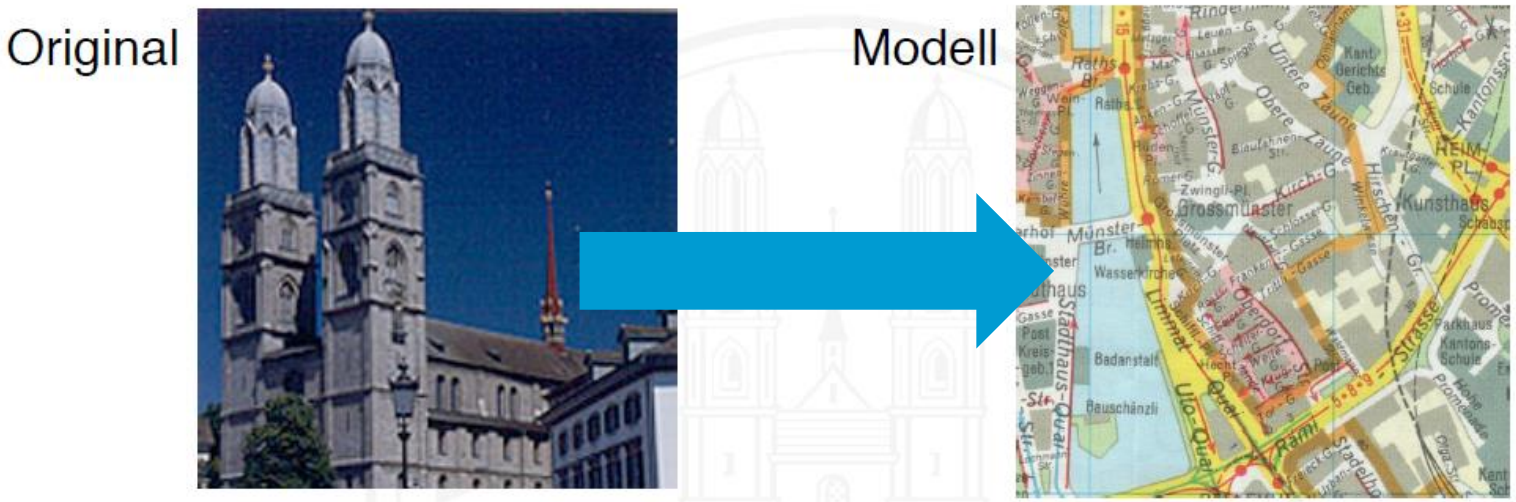


Quelle: M. Glinz 2005

Hauptmerkmale des Modellbegriffs nach H. Stachowiak [1973]

Verkürzungsmerkmal

- „Modelle erfassen im allgemeinen nicht alle Attribute des durch sie repräsentierten Originals, sondern nur solche, die den jeweiligen Modellerschaffern und/oder Modellbenutzern relevant erscheinen."
- Jedes Modell abstrahiert

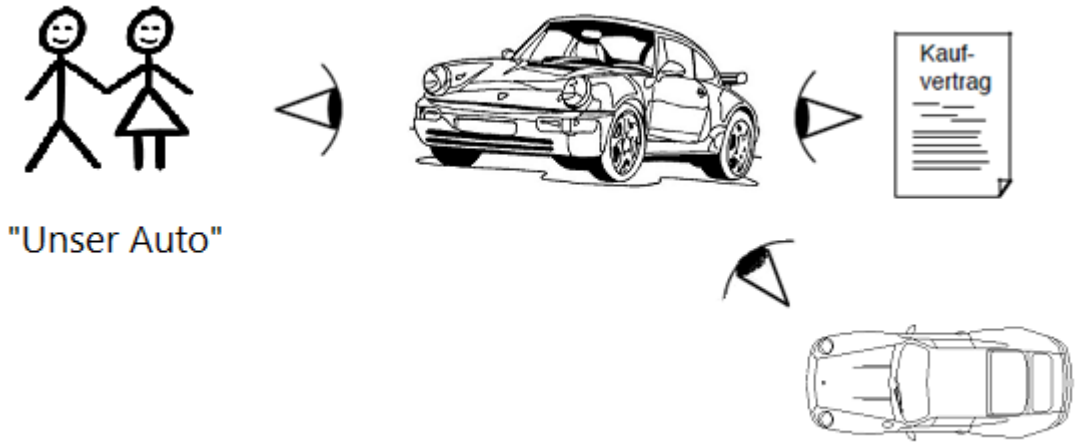


Quelle: M. Glinz 2005

Hauptmerkmale des Modellbegriffs nach H. Stachowiak [1973]

Pragmatisches Merkmal

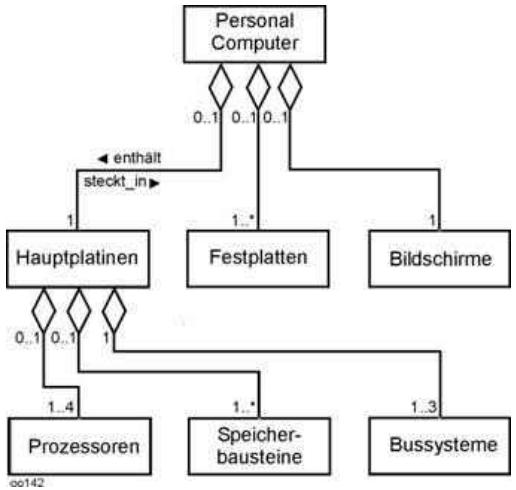
- „Modelle sind ihren Originalen nicht per se eindeutig zugeordnet. Sie erfüllen ihre Ersetzungsfunktion
  - Für bestimmte – erkennende und/oder handelnde, modellbenutzende – Subjekte
  - Innerhalb bestimmter Zeitintervalle und
  - Unter Einschränkung auf bestimmte gedankliche und tatsächliche Operationen."
- Jedes Modell wird im Hinblick auf einen Verwendungszweck geschaffen

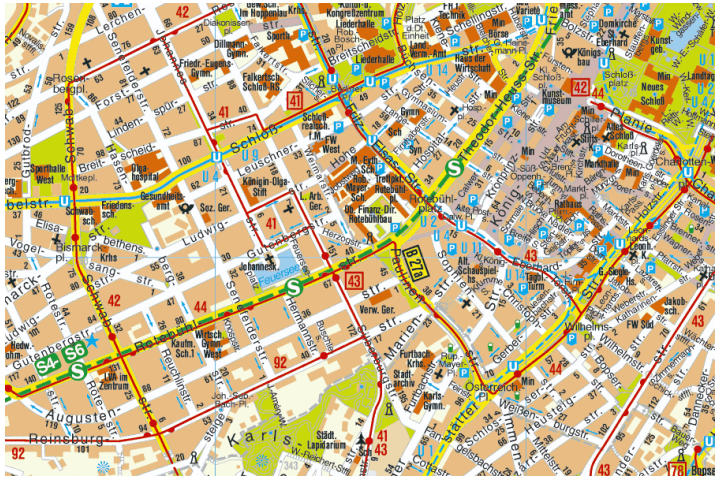


Quelle: M. Glinz 2005

Deskriptive Modellbildung

- Modellierung eines existierenden Originals
- Modellierung eines zukünftigen, aber nicht gestaltbaren Originals
- muss sich streng an der Realität orientieren

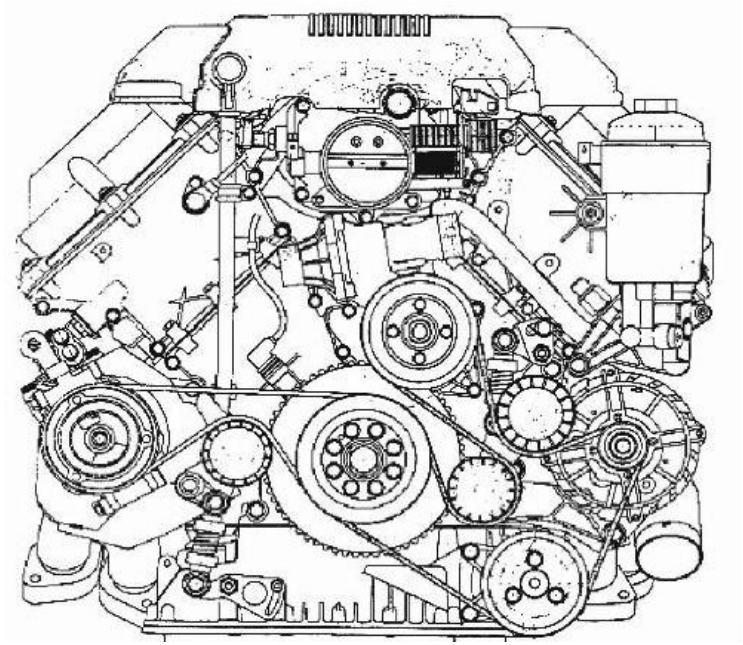




Beispiele: Stadtplan, Wettervorhersage, Komponentenstruktur eines im Einsatz befindlichen Informatiksystems

Präskriptive Modellbildung

- Modellierung eines zu schaffenden, gestaltbaren Originals
- darf zukünftige Realität gestalten



NVH-Merkmale Porsche Cayenne V8 Motor

- Ölwanne als tragendes, dynamisch versteifendes Element
- Abgasanlage dynamisch abgekoppelt
- Oberflächen ohne dominierende Eigenfrequenzen
- Motorfeste Sekundärluftpumpen
- Harmonischer Steifigkeitsverlauf
- Integrierte Motorenlageraufnahmen
- Kurbelgehäuseoberteil in Closed Deck Ausführung
- Abgekoppeltes und strömungsgeräuschoptimiertes Saugmodul
- Akutisch optimierter Steuertrieb mit Steuerkastendeckel
- Schwingungstechnisch optimierte Nebenaggregateankopplung
- Sehr steifer Kubeltrieb mit Kurbelgehäuseventil in Leiterrahmenbauweise

Beispiele: Konstruktionszeichnung, Anforderungsspezifikation für zu entwickelnde Teile

Grundsätze der Modellierung

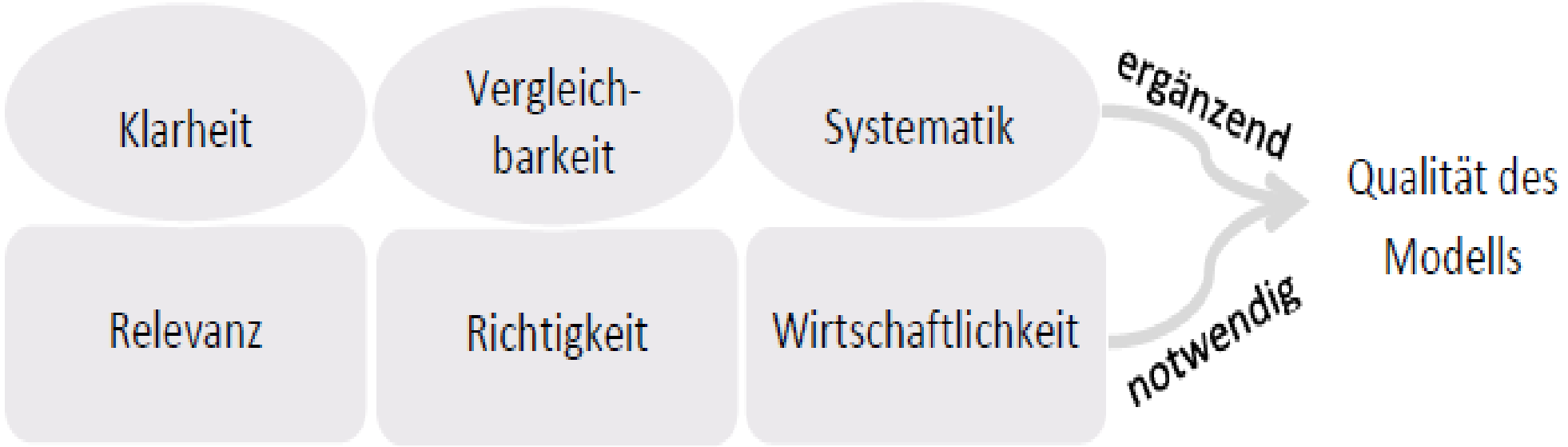


Abbildung 3-23 Grundsätze der Modellierung nach (Koch, 2011)

Grundsatz der Richtigkeit

- Das Modell soll die Wirklichkeit in den **wesentlichen Punkten richtig** wiedergeben.

- Die Richtigkeit der Modelle wird **von den Fachabteilungen geprüft** und ist dann erfüllt, wenn die betroffenen Personen ihre Tätigkeiten und Abläufe in den Modellen unter Verwendung der entsprechenden Fachbezeichnungen **wiedererkennen** und **nachvollziehen** können.

Grundsatz der Relevanz

MHP
A PORSCHE COMPANY

- Ein Modell muss **nicht alle Aspekte** der Wirklichkeit abbilden, doch es muss **alle relevanten** Aspekte der Wirklichkeit abbilden.
- Was im Einzelfall relevant ist, ergibt sich aus dem **Zweck**, den das Modell erfüllen soll.
- Zum Beispiel sind für ein DV-Konzept andere Aspekte der Wirklichkeit relevant als für ein fachliches Modell, das ein schnelles Verständnis eines Geschäftsprozesses vermitteln soll (also ein Swimlane-Diagramm oder ein strategisches Prozessmodell in der Terminologie aus Abschnitt 3.3.1.

Grundsatz der Wirtschaftlichkeit

- Prozessmodelle können, wenn sie **viele Details** enthalten, **sehr komplex** werden, und es ist dann auch oft **aufwändig**, sie zu erstellen.
- Der Grundsatz der Wirtschaftlichkeit fordert, dass ein Modell **nur so detailliert** ausgearbeitet werden soll, **wie** es für den angestrebten Verwendungszweck **erforderlich** ist.
- Das Modell soll nur soweit detailliert ausgearbeitet werden, bis es alle auftretenden **Fragen beantworten** kann.

Grundsatz der Klarheit

- Prozessmodelle sollen
  - leserlich,
  - verständlich und
  - so einfach und
  - anschaulich wie möglich sein.

- Praktisch bedeutet das zum Beispiel:
  - jedes Prozessmodell soll sich **links oben nach rechts unten** lesen lassen,
  - Linien sollen sich möglichst **wenig schneiden**

Grundsatz der Vergleichbarkeit

- Die Prozessmodelle innerhalb eines Projekts sollten miteinander **vergleichbar** sein.

- Praktisch bedeutet das, für alle Modelle auf **derselben Detaillierungsebene** und in **derselben Sicht** einheitliche **Modellierungsverfahren** und einheitliche **Gestaltungsrichtlinien** zu einzusetzen.

Grundsatz des systematischen Aufbaus

- Nach dem Grundsatz des systematischen Aufbaus sollen Modelle in den verschiedenen Sichten und Detaillierungsebenen zueinander **konsistent** sein: wenn Elemente wie Informationsobjekte, Organisationseinheiten, Teilprozesse oder Aktivitäten in Modellen verschiedener Sichten oder Detaillierungsebenen vorkommen, sollen sie überall **gleich heißen**.

- Die ersten drei Grundsätze, Richtigkeit, Relevanz und Wirtschaftlichkeit, muss ein gutes Prozessmodell zwingend erfüllen. Die Grundsätze der Klarheit, Vergleichbarkeit und des systematischen Aufbaus können die Qualität der Modellierung weiter verbessern(Abbildung 3-23).

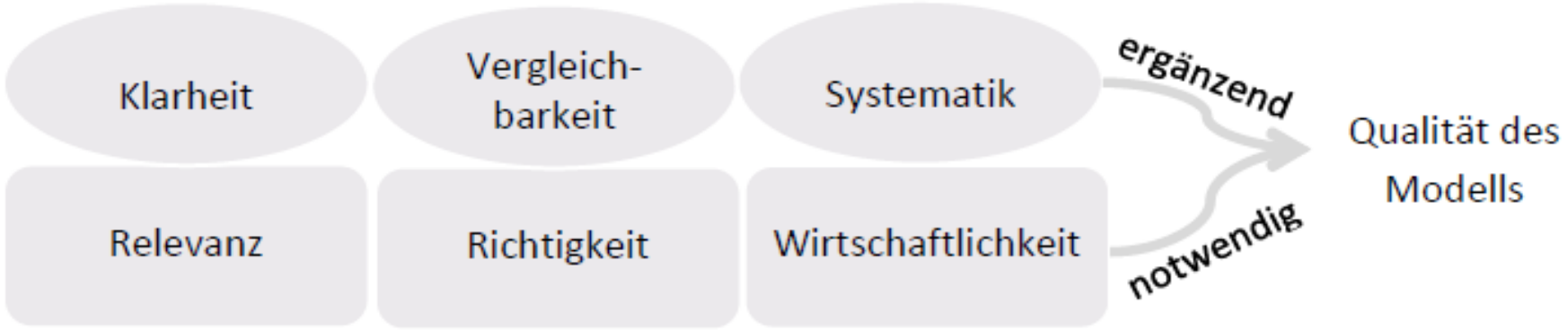


Abbildung 3-23 Grundsätze der Modellierung nach (Koch, 2011)

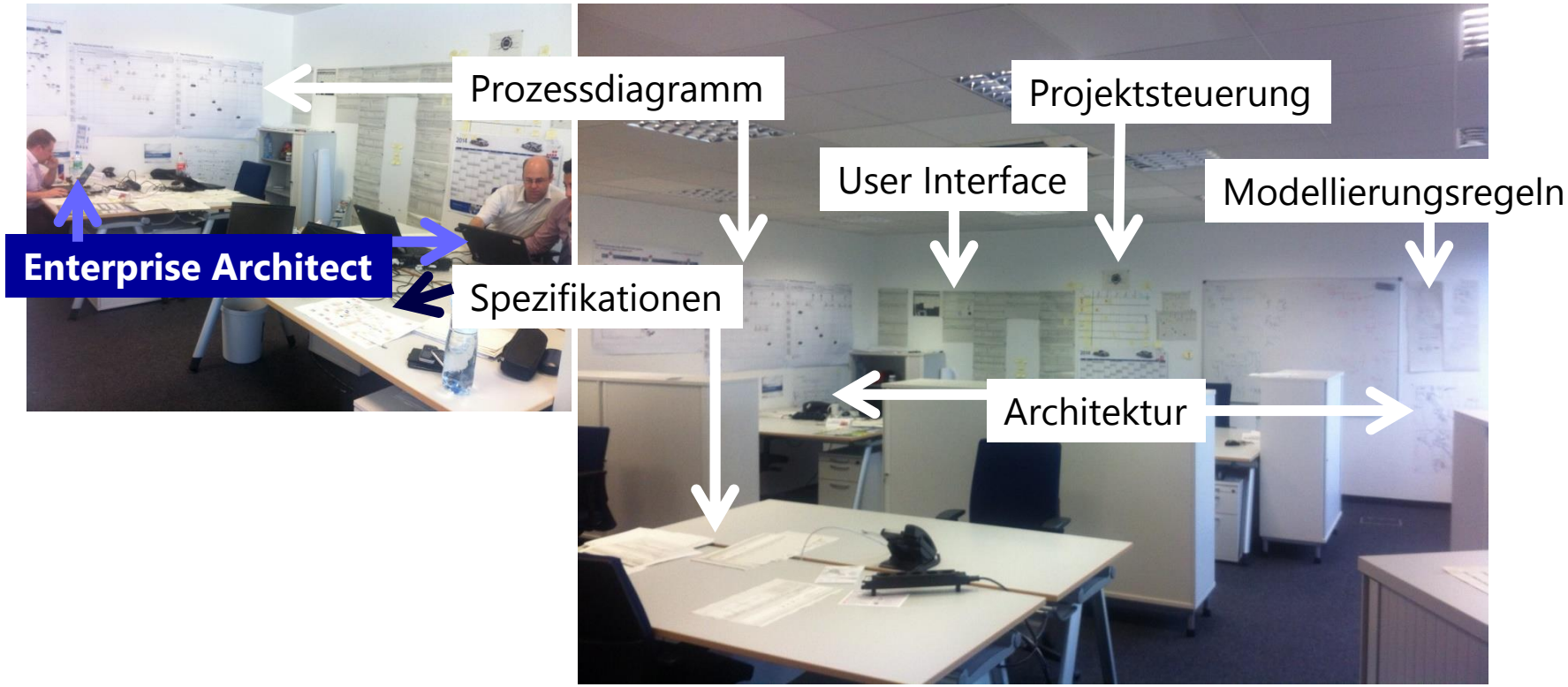
Agenda

MHP
A PORSCHE COMPANY

1. Grundlagen der Modellierung
   - Prinzipien der SW-Technik
   - Prinzip der Abstraktion
   - Prinzip der Bindung und Kopplung
   - Prinzip der Hierarchisierung
   - Geheimnisprinzip
   - (Softwareentwicklungs-) Werkzeuge

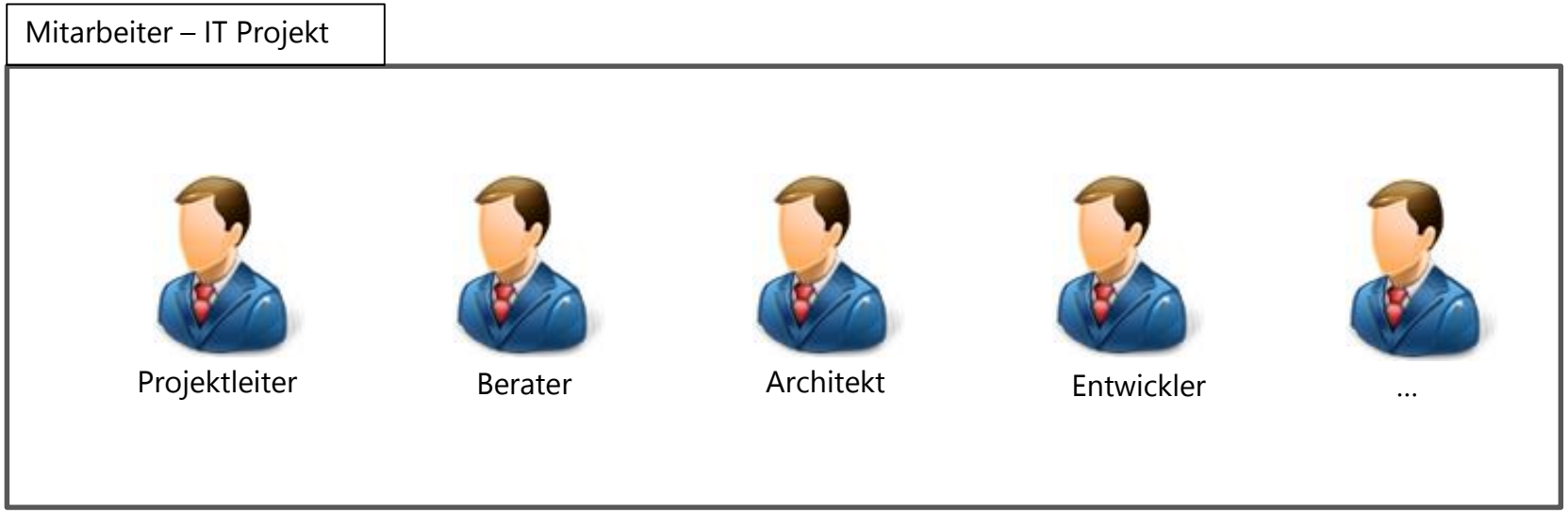
2. **Einleitung Vorgehensweise ganzheitliche Modellierung**

Zielgruppe

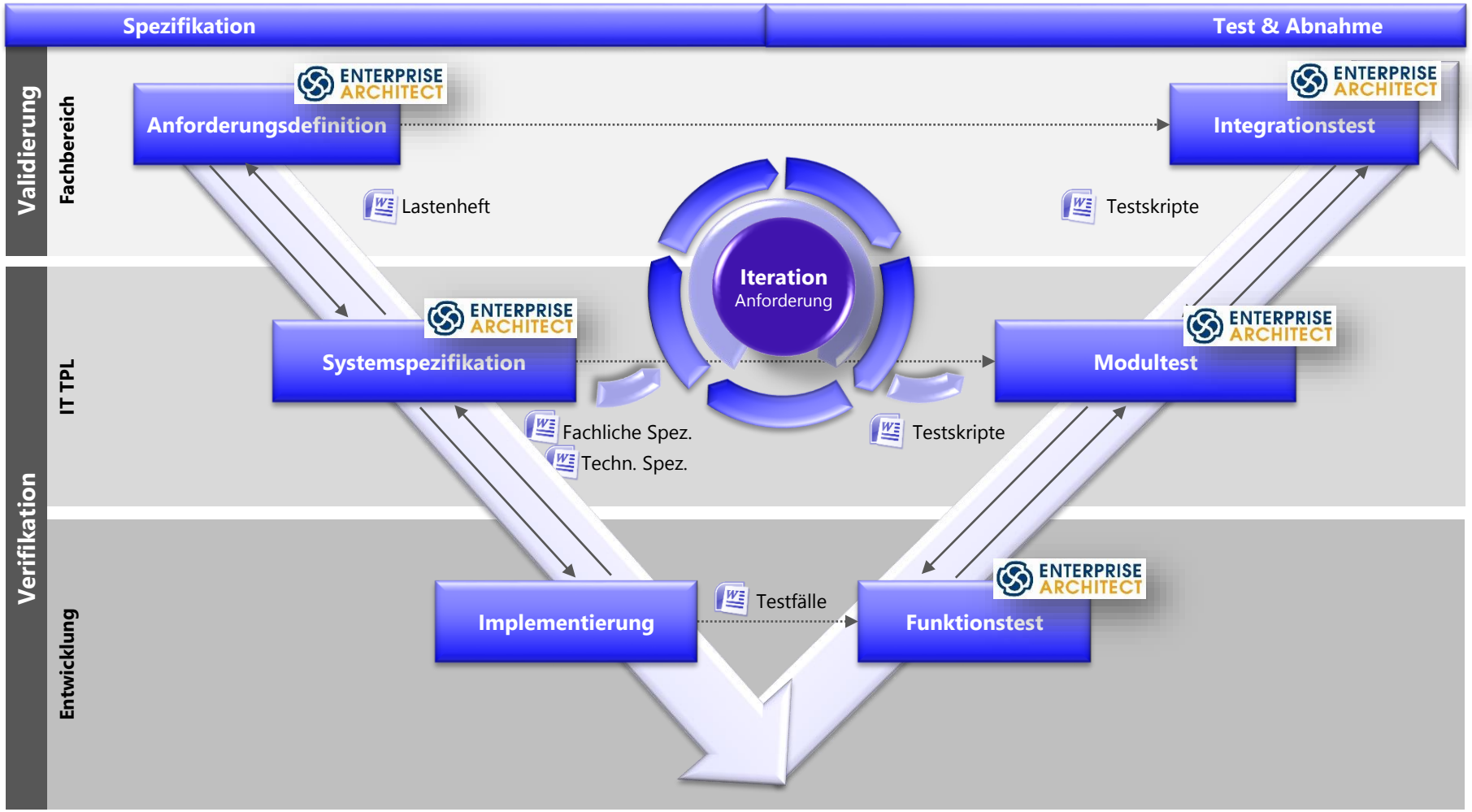


Zielgruppe

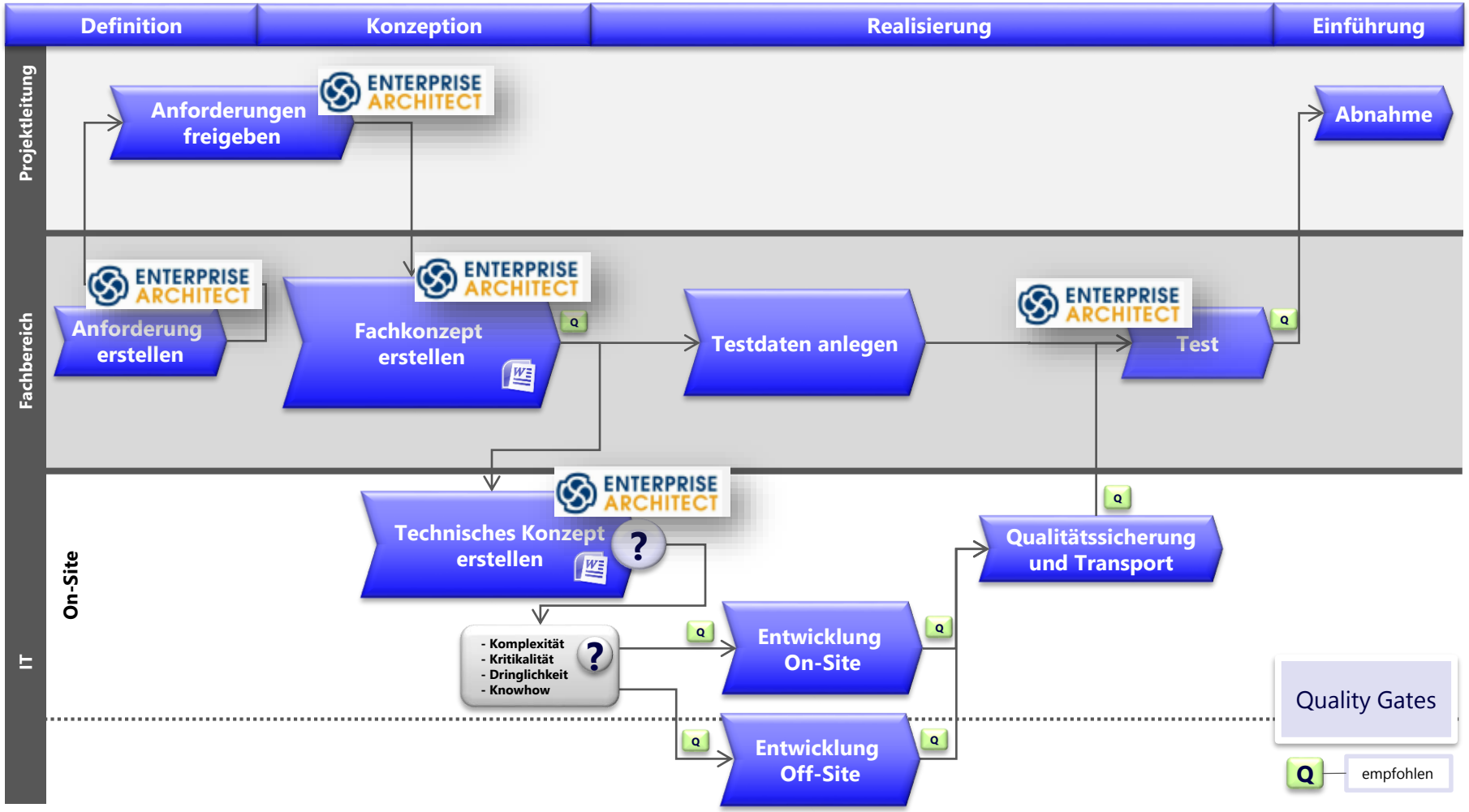
- **Alle Mitarbeiter** die in IT Projekten (Vor allem Individualentwicklung) involviert sind



V-Modell Vorgehensmodell – Individualentwicklung (klassisch und agil)



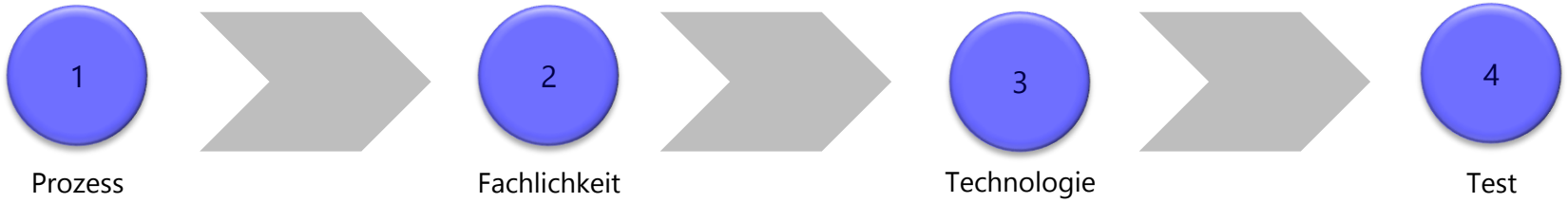
Vorgehensmodell für IT Projekte in Beratungsfirmen (Basismodel - MHP)



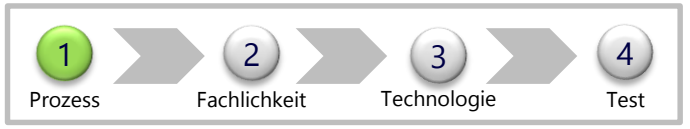
Überführung der Informationen ins Modell

MHP
A PORSCHE COMPANY

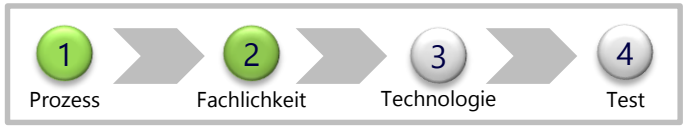
Ganzheitliche Modellierung

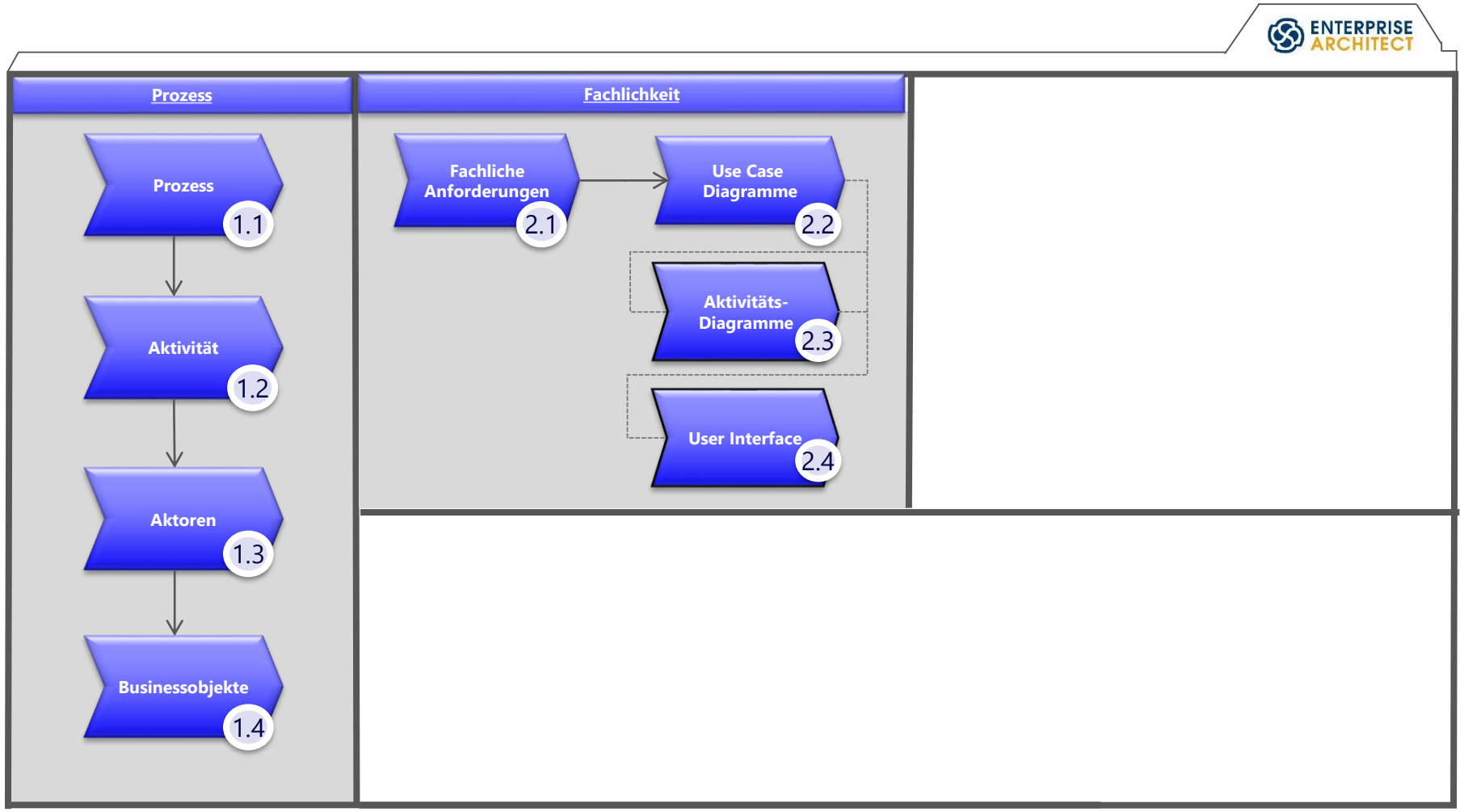


Überführung der Informationen ins Modell

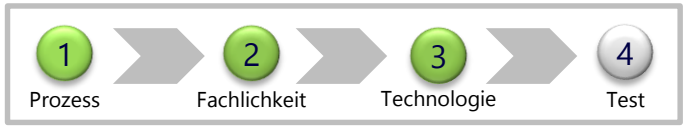


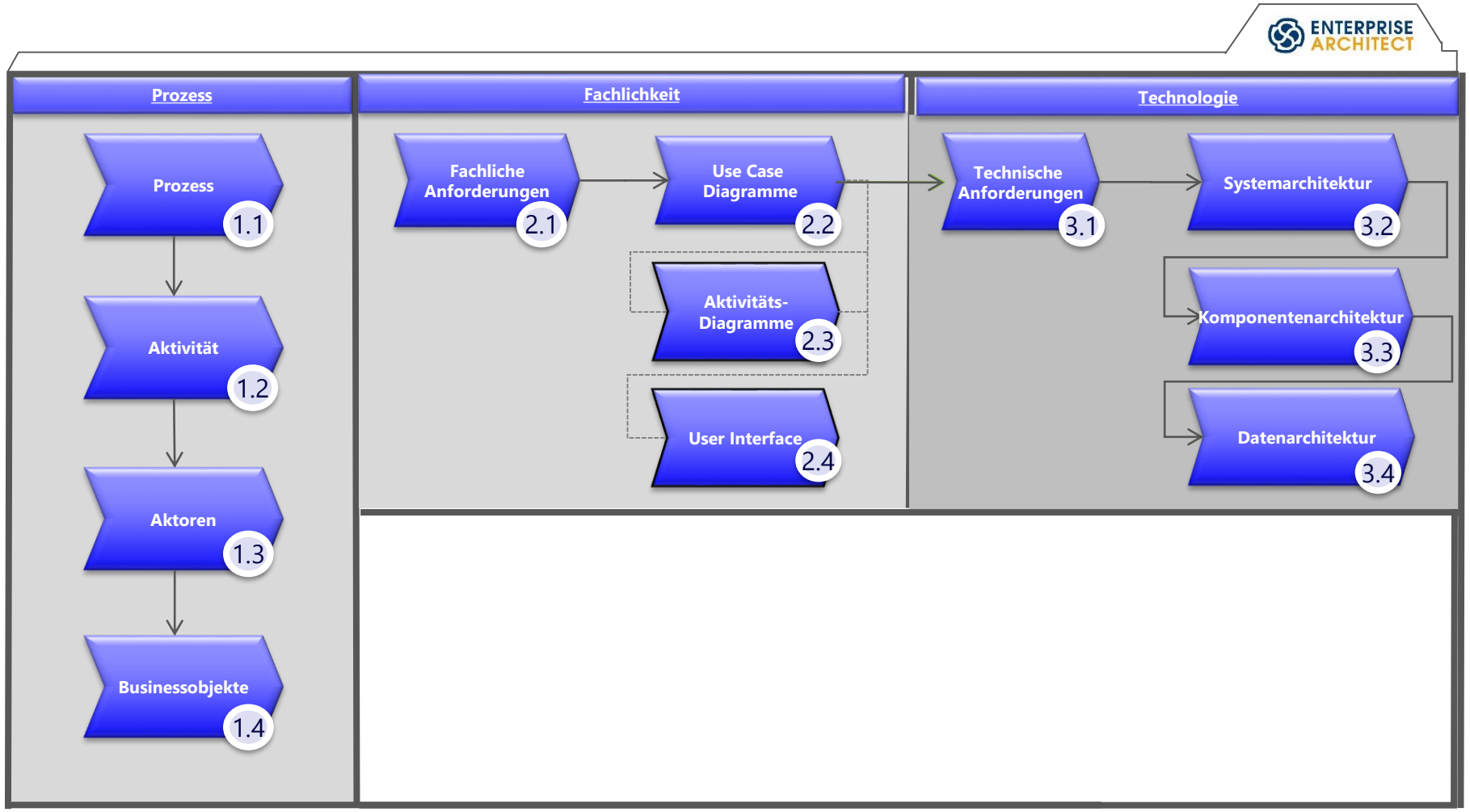
Überführung der Informationen ins Modell



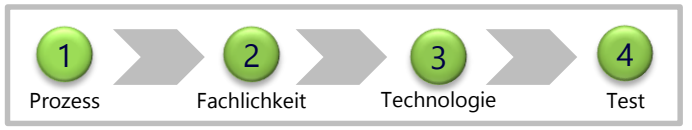


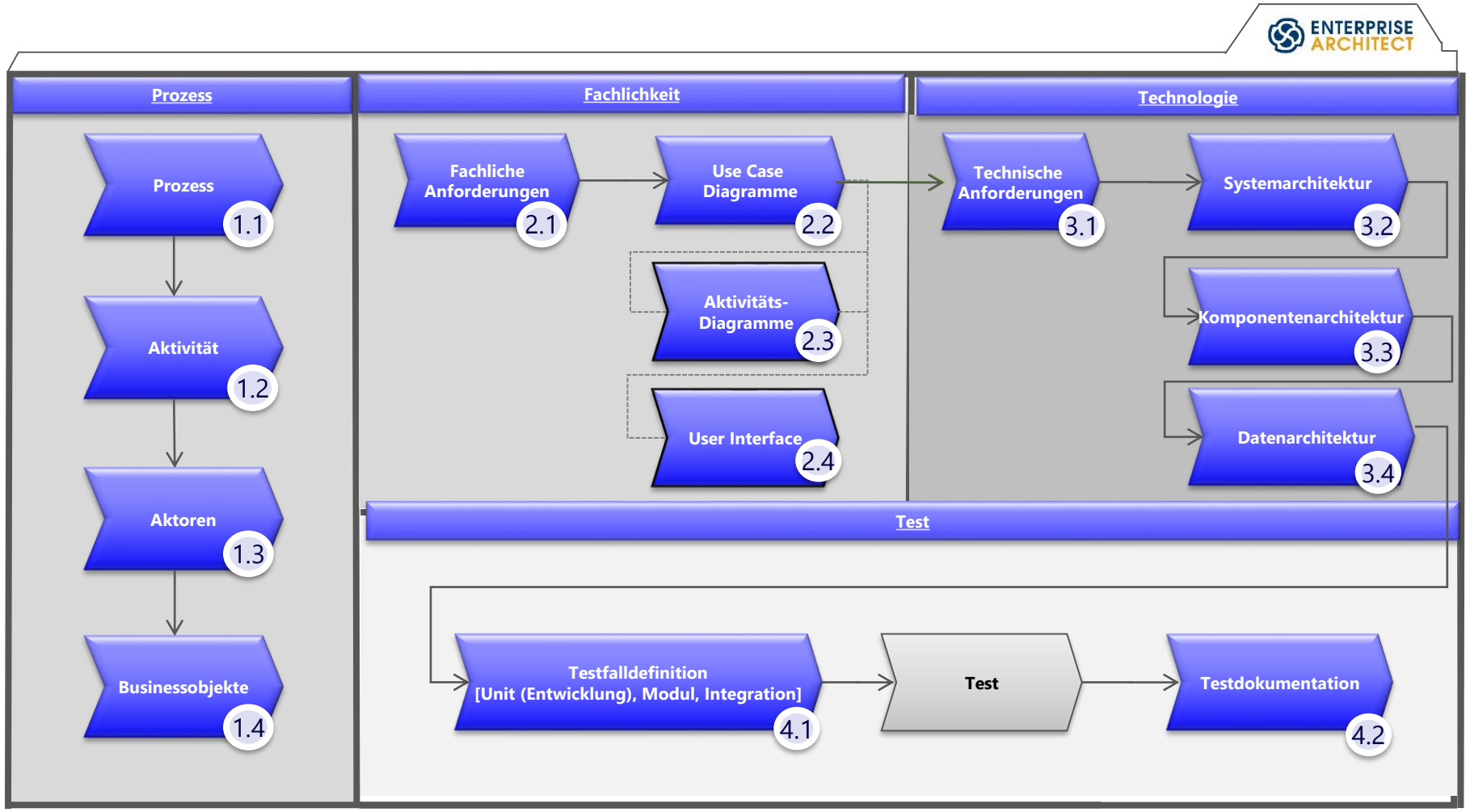
Überführung der Informationen ins Modell





Überführung der Informationen ins Modell





Mapping auf TOGAF

TOGAF:
Business Architecture

BPML / UML

The **Business Architecture** defines the business strategy, governance, organization, and key business processes.

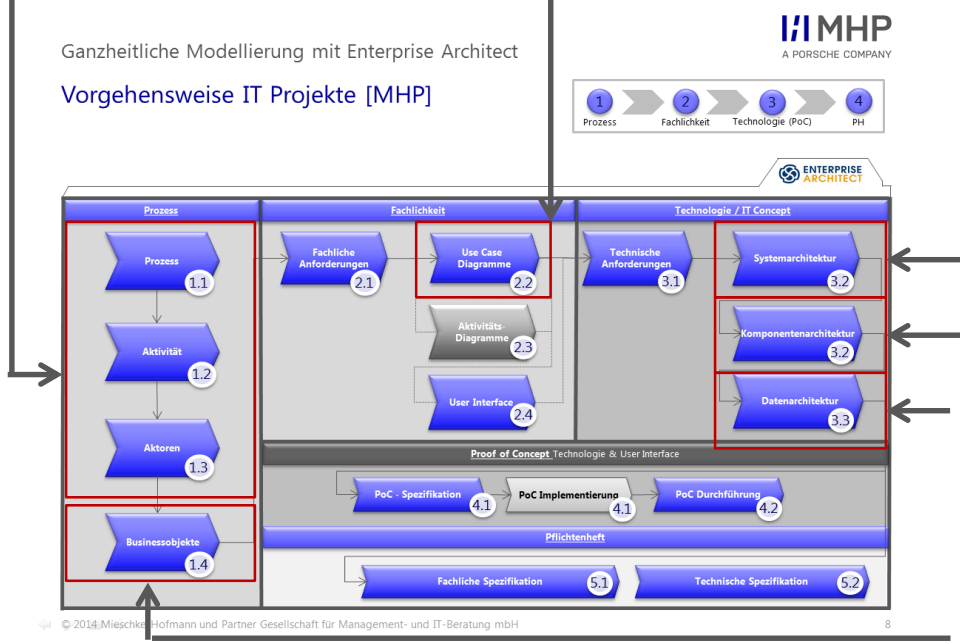
TOGAF:
Technology Architecture

The **Technology Architecture** describes the logical software and hardware capabilities that are required to support the deployment of business, data, and application services. This includes IT infrastructure, middleware, networks, communications, processing, standards, etc.

TOGAF:
Application Architecture

The **Application Architecture** provides a blueprint for the individual applications to be deployed, their interactions, and their relationships to the core business processes of the organization.

UML



TOGAF:
Data Architecture

ERM/ UML

The **Data Architecture** describes the structure of an organization's logical and physical data assets and data management resources.

MHP
A PORSCHE COMPANY

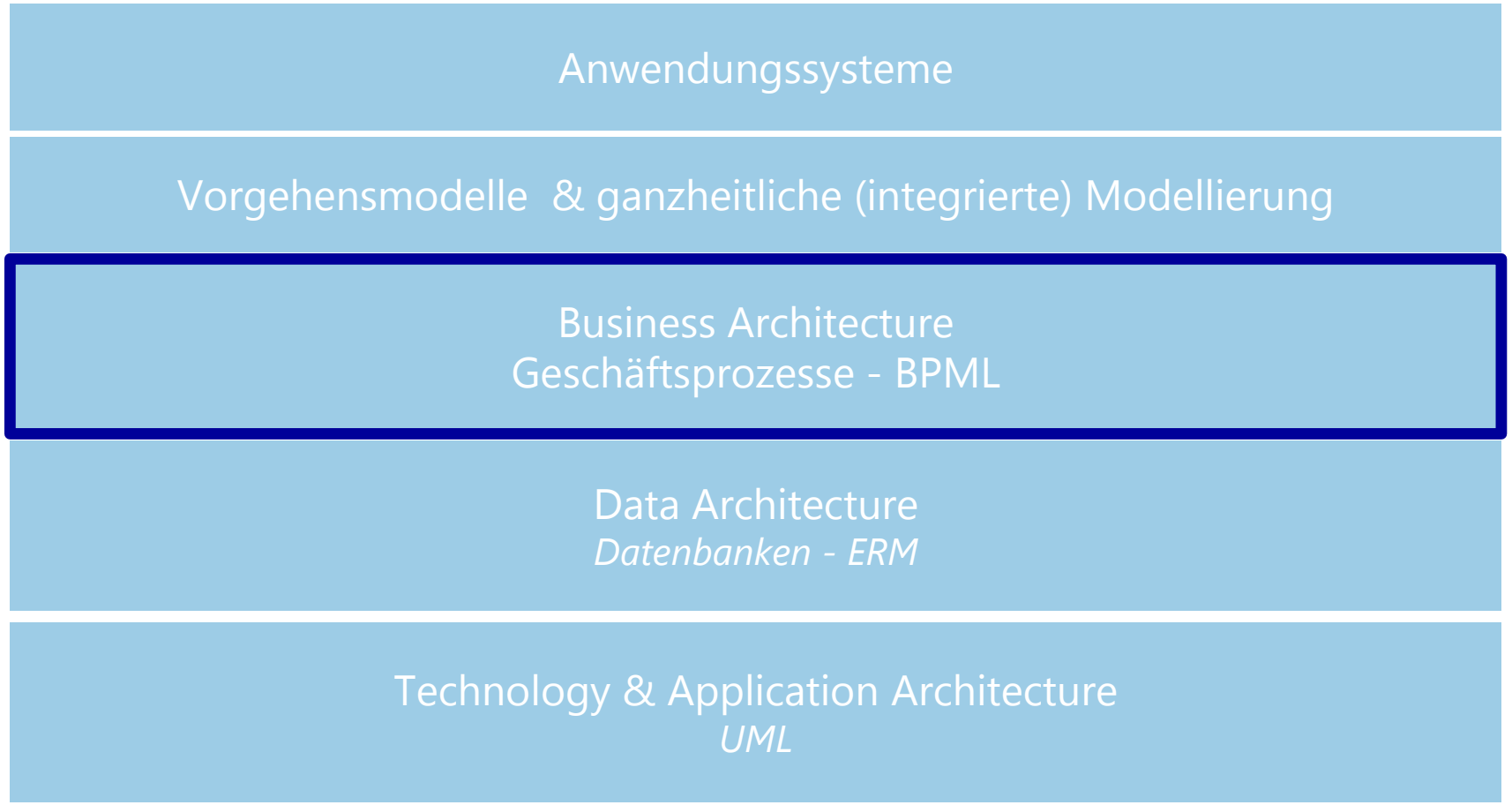
INTEGRIERTE MODELLIERUNG KOMPLEXER SYSTEME

Geschäftsprozesse und Prozessmodellierung

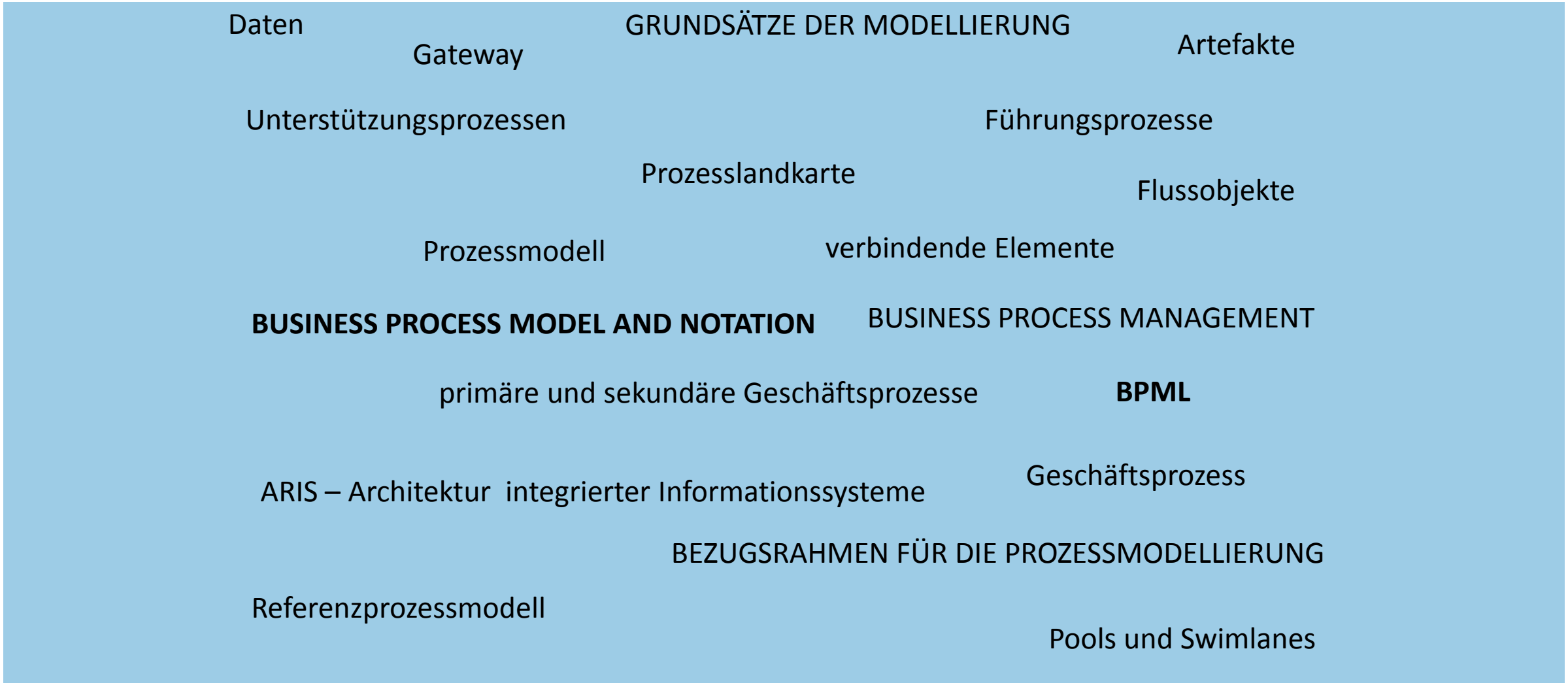
UNIVERSITATEA BABEȘ-BOLYAI

Hugo Colceag I Bachelor Studiengang Informatik

Vorlesungsinhalte und Aufbau



Mindmap - Geschäftsprozesse und Prozessmodellierung



Lernziele

- ✓ Grundsätze der Geschäftsprozess-Modellierung
- ✓ Syntax und Logik BPMN
- ✓ ARIS-Modell

Agenda

MHP
A PORSCHE COMPANY

1. **Modellierung von Geschäftsprozessen**

Motivation

- Geschäftstätigkeit eines Unternehmens betrachtet man heute vor allem aus der Sicht der Geschäftsprozesse
- Zusammenwirken von Menschen innerhalb von Geschäftsprozessen
- Geschäftsprozessen setzen in erheblichem Umfang, Anwendungssysteme und IT ein

- **Prozessmodelle:**
  - zentrales Hilfsmittel im Umgang mit Geschäftsprozessen
  - beschreiben Geschäftsprozesse, die sehr komplex sein können
  - unterschiedliche Abstraktionsebenen
  - graphische Darstellungen als Diagramme

Warum sind Prozessmodelle hier ein Thema?

- **Unterschiedliche Aspekte:**
  - Qualitätsmanagement
  - Geschäftsprozessmanagement (*Business Process Management,* BPM)
  - IT-Projekte

- **Referenzprozessmodell**

  Ein Referenzprozessmodell ist ein Prozessmodell, das eine **bewährte oder empfohlene Vorgehensweise** (eine sogenannte ***Best Practice***) für die Lösung einer betriebswirtschaftlichen Problemstellung beschreibt, die als Vorlage und Ausgangspunkt für die Entwicklung eines individuellen Geschäftsprozesses bei einem Unternehmen dienen kann.

  Anwendungssoftware als Sammlung von Best Practices in Form von Referenz-prozessmodellen

  Ausgangspunkt für Anpassungen

  Verantwortliche Personen sollten daher wissen, welche wichtige Rolle Prozessmodelle im Rahmen von IT- und Software-Einführungsprojekten spielen

Geschäftsprozesse

- Ein **Geschäftsprozess** besteht aus der funktions- und organisationsübergreifenden Verknüpfung wertschöpfender Aktivitäten, die von Kunden erwartete Leistungen erzeugen und aus der Unternehmensstrategie abgeleitete Ziele umsetzen. (Schmelzer & Sesselmann, 2010)

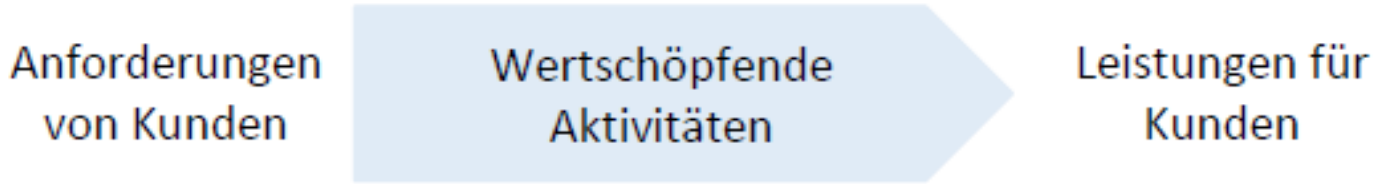


Abbildung 3-1 Geschäftsprozess

- **Beispiele**:
  - Beschaffung
  - Produktion
  - Vertrieb
  - Marketing

Welche Geschäftsprozesse gibt es?

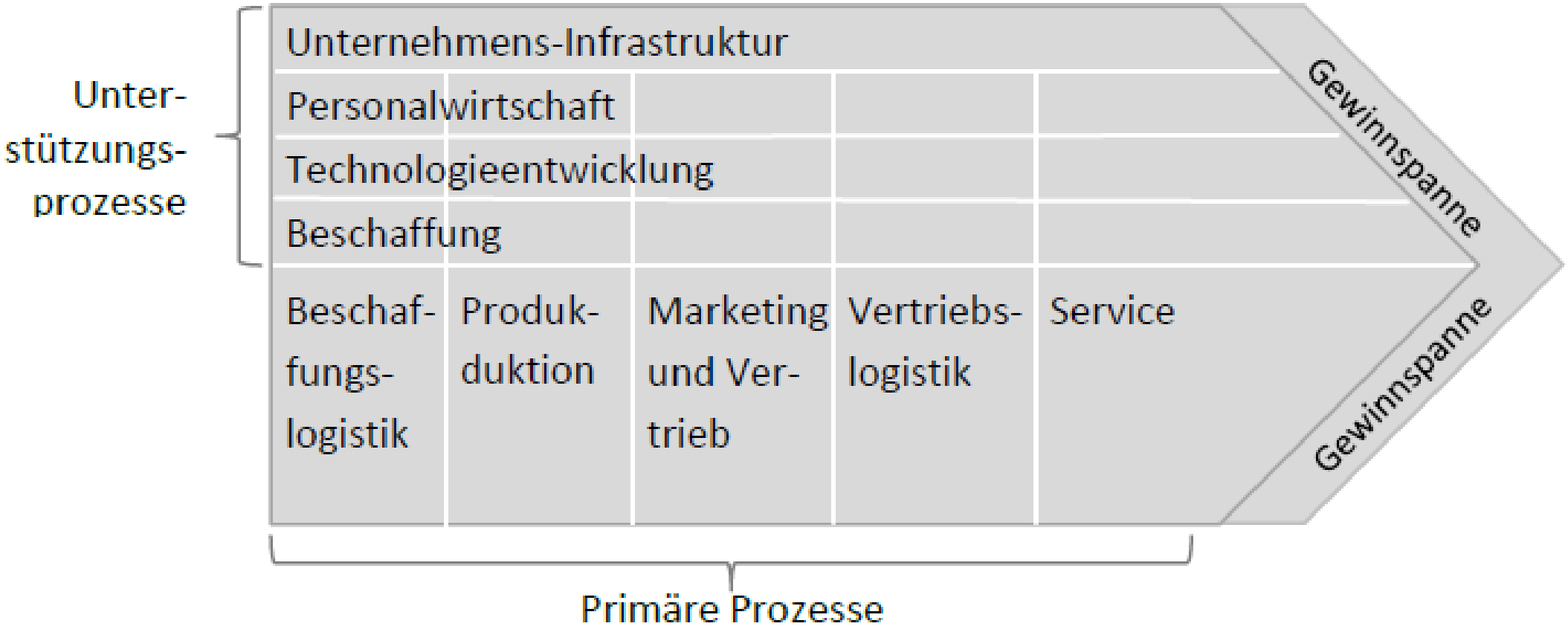


Abbildung 3-2 Die Wertschöpfungskette nach Porter (Porter, 1996)

Welche Geschäftsprozesse gibt es?

MHP
A PORSCHE COMPANY

- **Primärprozesse** (Leistungsprozesse, Kernprozesse, operative Prozesse)
  - wertschöpfende Aktivitäten
  - tragen unmittelbar zur Produktion und Distribution der Produkte oder Dienstleistungen eines Unternehmens bei
  - Beispiele: Eingangslogistik, Produktion, Marketing und Vertrieb
    Vertriebslogistik (Distribution), Service

- **Sekundärprozesse** (Unterstützungsprozesse)
  - unterstützende wertschöpfende Aktivitäten
  - Sind Voraussetzung für die primären wertschöpfenden Aktivitäten eines Unternehmens
  - tragen indirekt dazu bei, Leistungen für den Kunden zu erbringen
  - Beispiele: Unternehmens-Infrastruktur, Personalwirtschaft, Buchhaltung

- **Führungsprozesse**
  - Strategieentwicklung
  - Unternehmensplanung
  - Personalführung oder Controlling

Prozesslandkarte

- Überblick der Geschäftsprozesse von Unternehmen
- Zusammenhänge und Wechselbeziehungen zwischen den Geschäftsprozessen
- Zusammenwirken mit Prozessen externer Partnern (Kunden und Lieferanten
- Die Prozesslandkarte stellt einen Teil der Geschäftsprozesse des Unternehmens dar, untergliedert in Teil-Geschäftsprozesse
- Auswahl der dargestellten Geschäftsprozesse richtet sich nach dem Zweck der Prozesslandkarte
- kann dem Bedarf der Nutzer angepasst werden

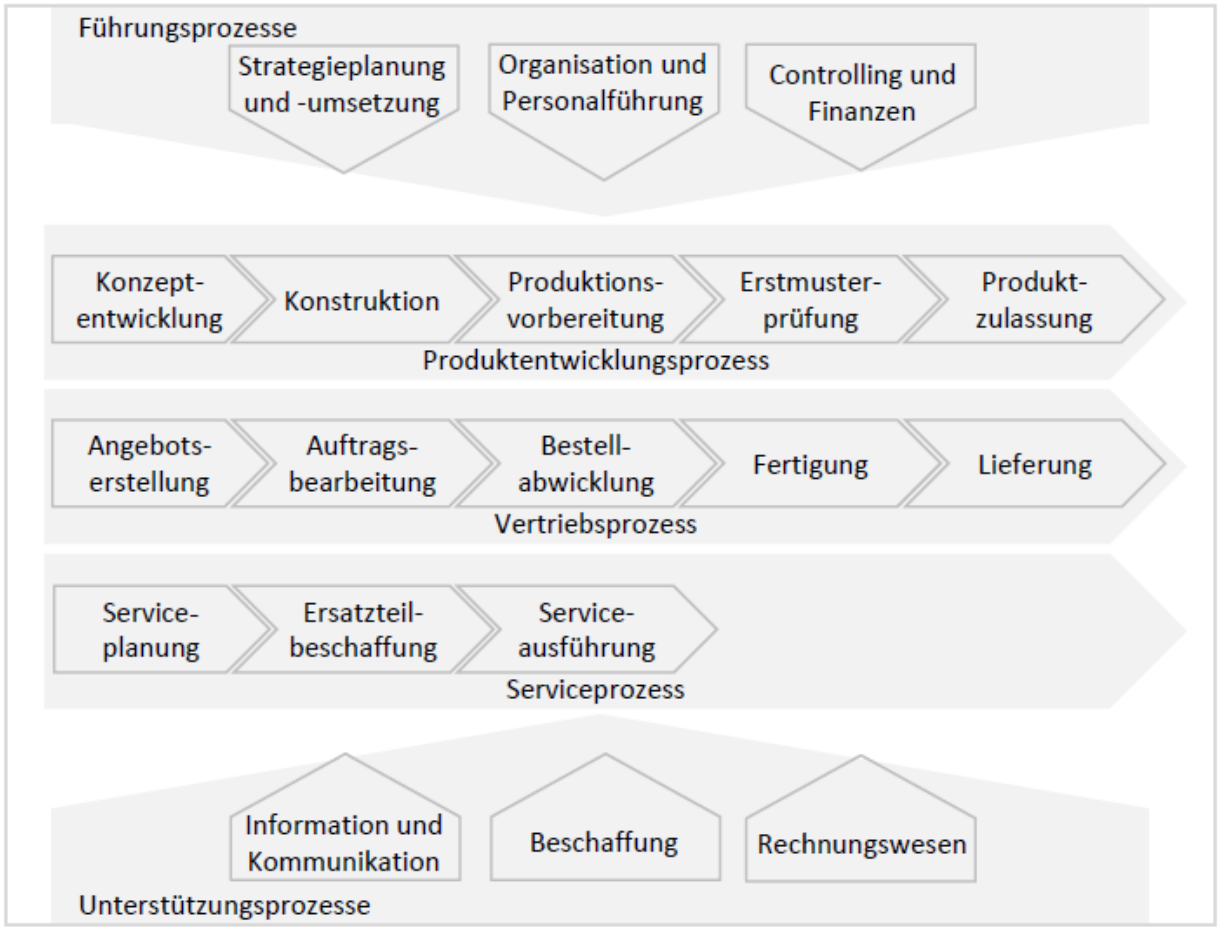


Abbildung 3-3 Prozesslandkarte eines Industriebetriebs

Prozesslandkarte

- keine genauen Vorgaben oder Definitionen, wie eine Prozesslandkarte genau auszusehen hat
- Zweckmäßigkeit im Vordergrund
- Enthalten keine Details (wie einzelne Aktivitäten, Akteure oder konkreten Input oder Output)
- Verwendung der Symbole einer Wertschöpfungskette zur Darstellung
- Führungsprozesse im oberen Bereich
- Unterstützungsprozesse im unteren Bereich
- Kernprozesse im mittleren Bereich dargestellt

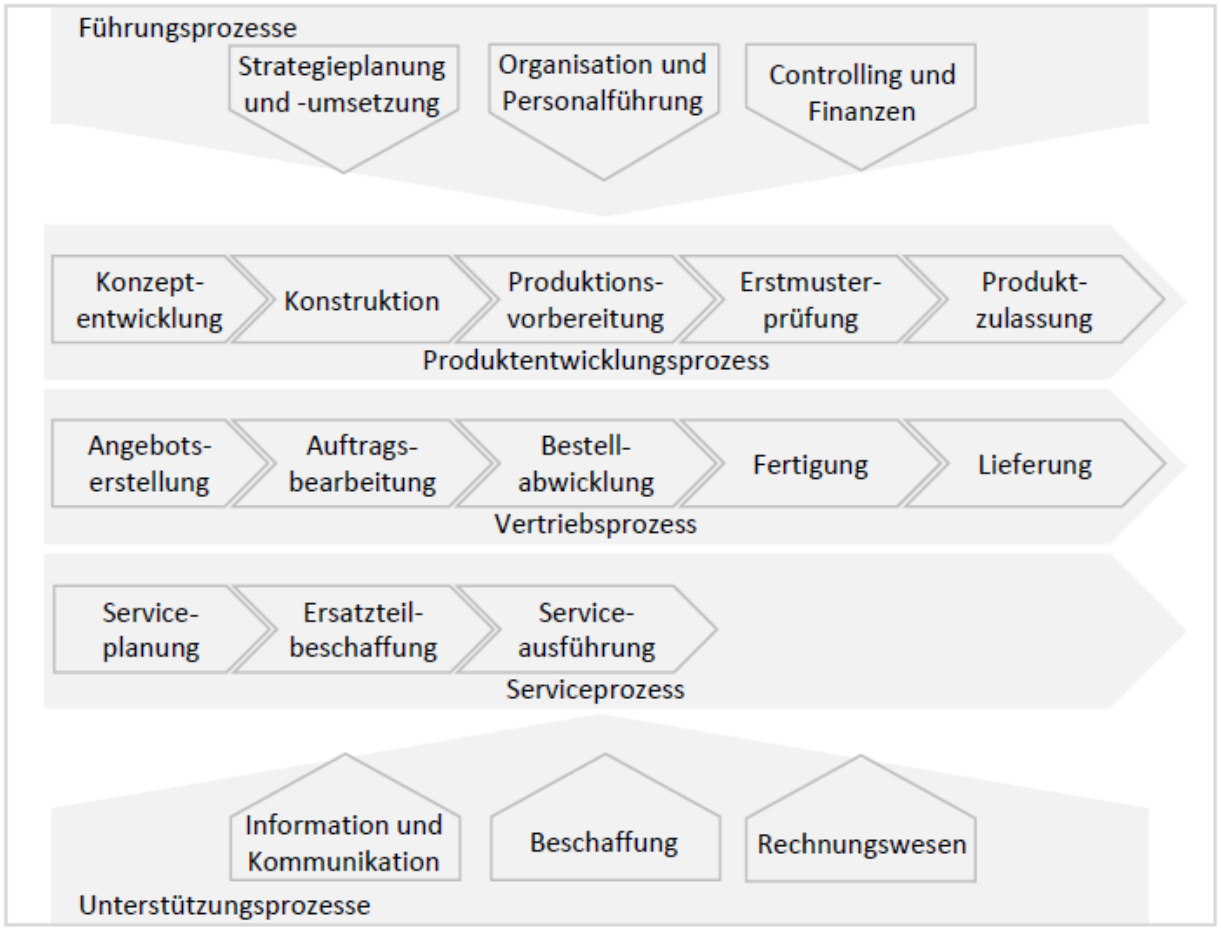


Abbildung 3-3 Prozesslandkarte eines Industriebetriebs

Bezugsrahmen für die Prozessmodellierung

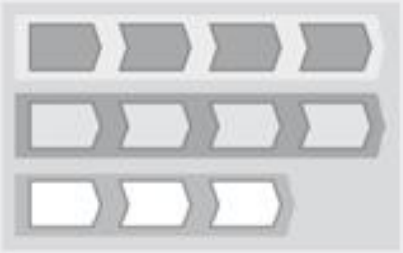
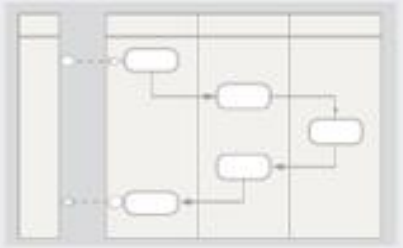
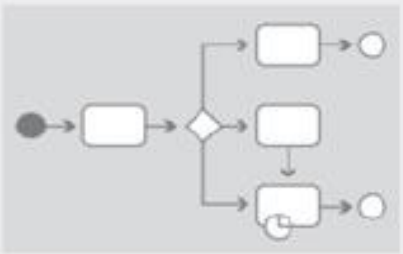
Kontext Prozessmanagement	
Prozesslandkarte beschreibt die Geschäftsprozesse des Unternehmens im Überblick.	
Swimlane-Diagramm dient zur Visualisierung von Zuständigkeiten von Teil-Geschäftsprozessen.	
Ein **Prozessablauf-Diagramm** zeigt den Prozessablauf im Detail. Er beschreibt, welcher Auslöser einen Prozess anstößt, in welcher Reihenfolge und unter welchen Bedingungen Aktivitäten durchgeführt werden und wer eine Aktivität im Prozess ausführt.	

Abbildung 3-4 Darstellungsarten von Geschäftsprozessen im Kontext Prozessmanagement (Hanschke, Giesinger, & Goetze, 2013)

Business process management mit organisatorischem Fokus

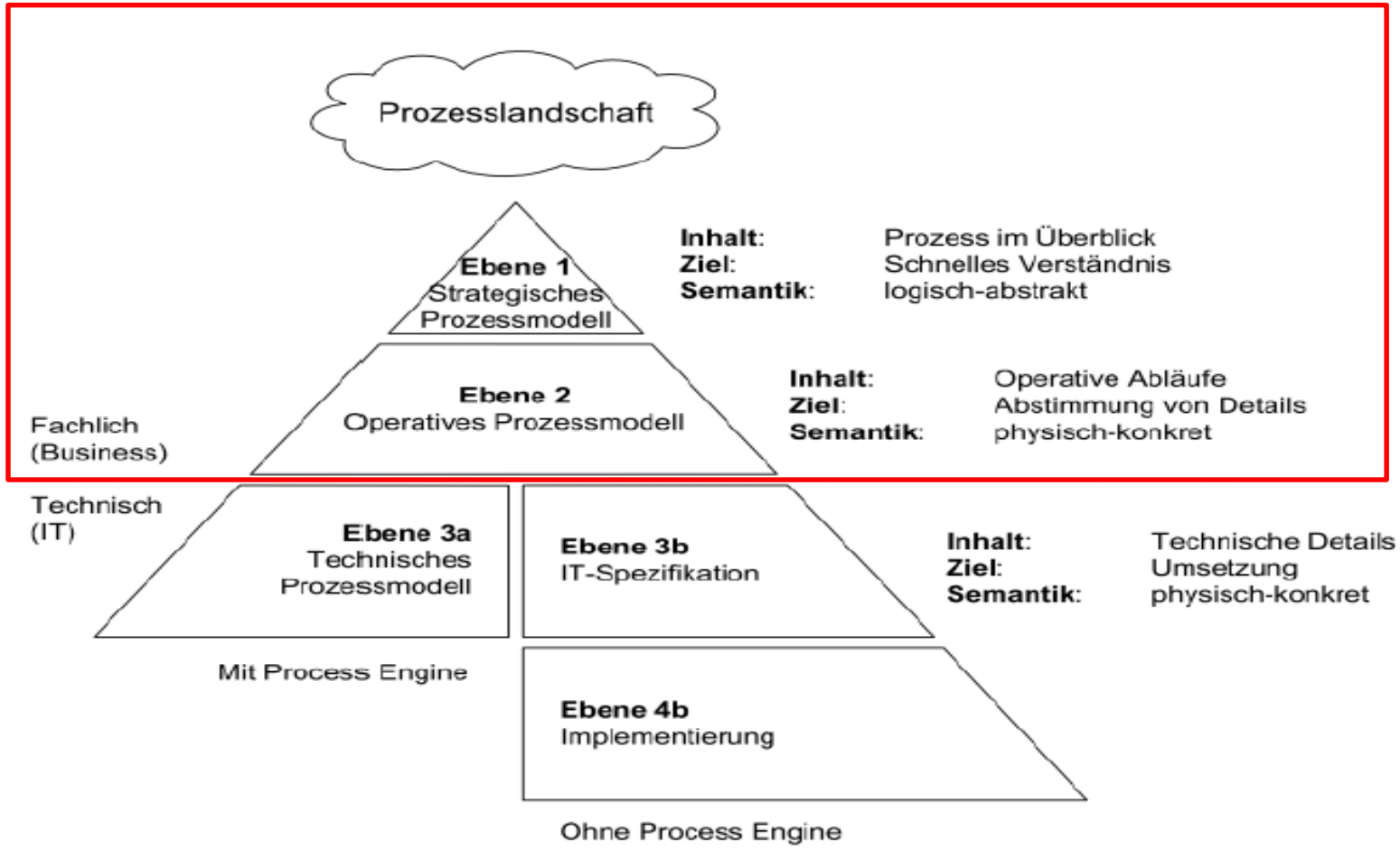


Abbildung 3-5 Das camunda BMNP Framework (Freund & Rücker, 2012, S. 15)

Business process management mit technischem Fokus

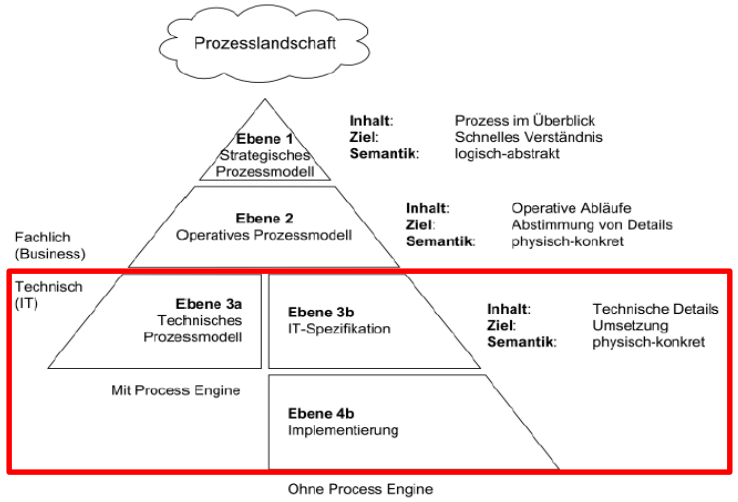


Abbildung 3-5 Das camunda BMNP Framework (Freund & Rücker, 2012, S. 15)

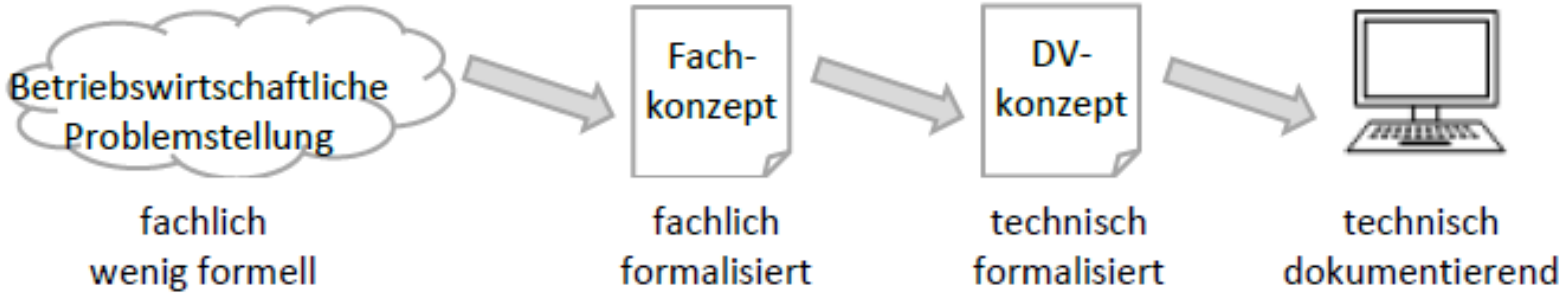


Abbildung 3-6 Beschreibungsebenen in ARIS

ARIS – **A**rchitektur integrierte**R** **I**nformations**S**ysteme

- Architektur integrierter Informationssysteme
- Bezugsrahmen für die Modellierung von Informationssystemen
  - Beinhaltet neben den Geschäftsprozessen auch andere Aspekte eines Informationssystems im Unternehmen

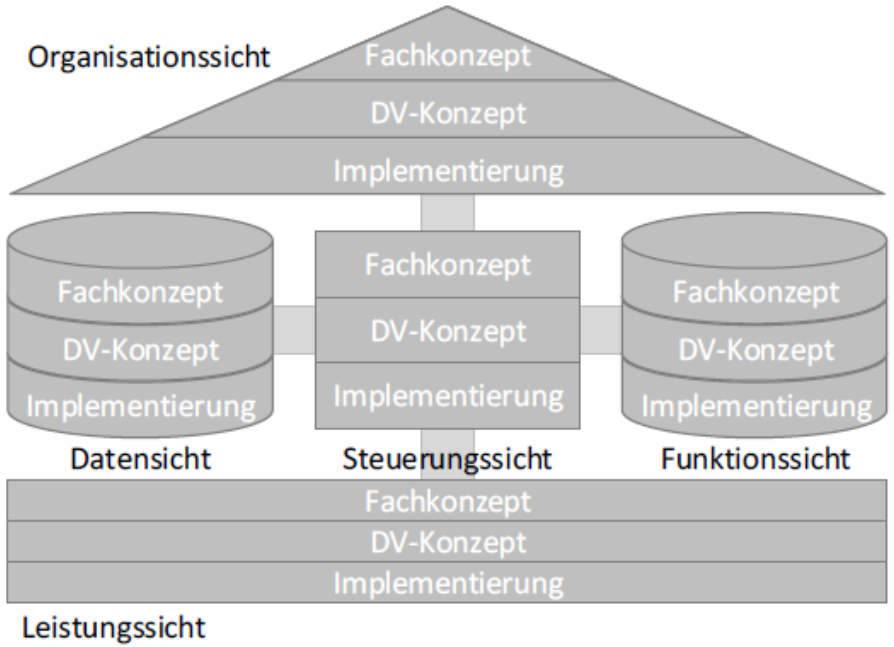


Abbildung 3-7 Das ARIS-Haus (Gadatsch, 2010) nach (Scheer, 1998)

Beschreibungsebenen in ARIS

- **Beschreibungsebenen**
- Fachkonzept
  - Entwurf einer Lösung der betriebswirtschaftlichen Problemstellung durch ein Informationssystem
  - in der Fachsprache der Anwender verfasst
  - stark formalisierte Notation
  - Ziel: Spezifikation für Anwendungssysteme
- DV-Konzept
  - eindeutige formalisierten Notation
  - Verfasst mit Begriffe der IT und mit Bezug auf Anwendungssysteme
  - Ziel: Realisieren und Implementieren
- Implementierung
  - Dokumentation der tatsächlichen technischen Implementierung und Realisierung

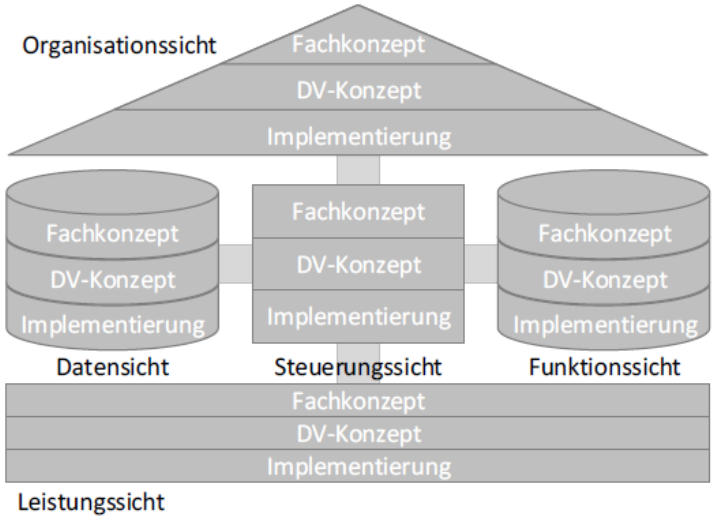
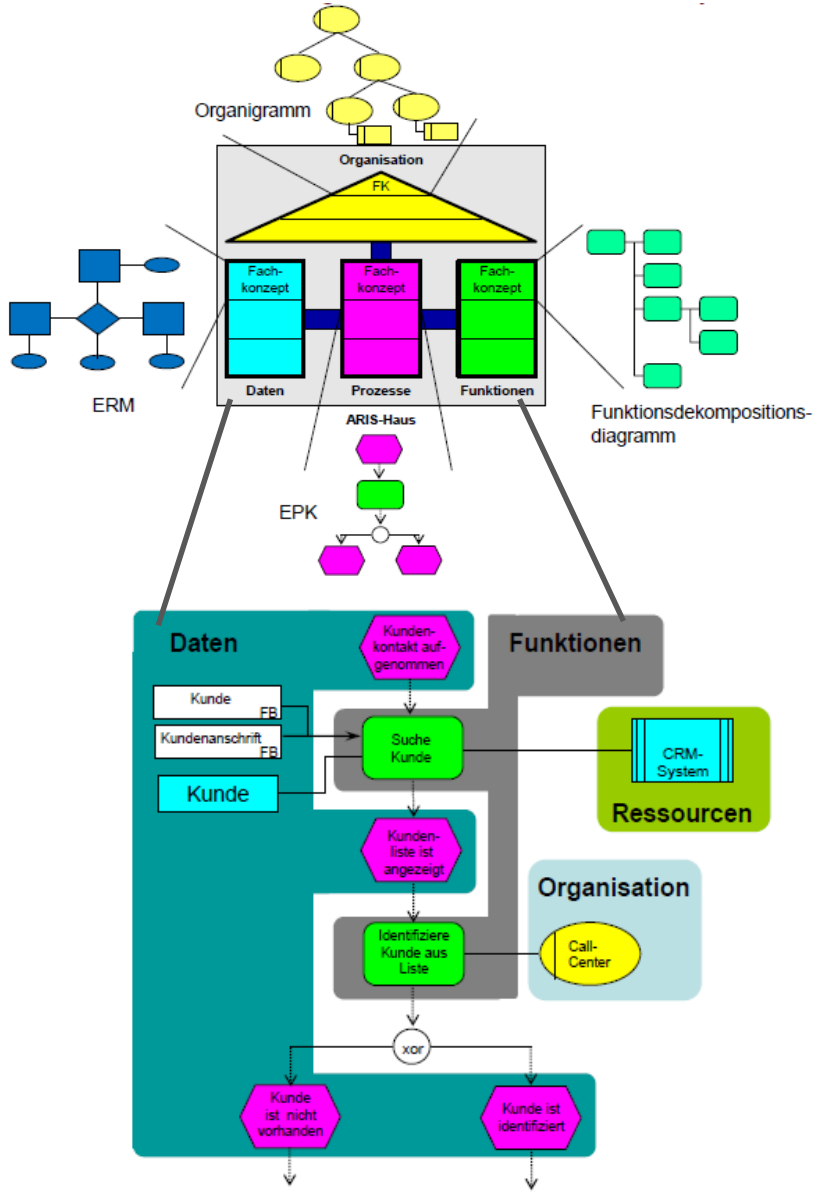


Abbildung 3-7 Das ARIS-Haus (Gadatsch, 2010) nach (Scheer, 1998)

ARIS wurde bereits 1998 entwickelt von A.-W. Scheer (Scheer, 1998), einem einflussreichen und aktiven deutschen Unternehmer und Professor für Wirtschaftsinformatik (Scheer Group GmbH, 2014). Das von A.-W. Scheer gegründete Software- und Beratungsunternehmen IDS Scheer AG (seit 2010 Software AG) setzt ARIS ein und bietet eine Palette von Software-Werkzeugen, mit den Prozess- und andere ARIS-Modelle erstellt und verwaltet werden können.

Sichten in ARIS

- **Organisation**ssicht
  - organisatorische Strukturen im Unternehmen
- **Daten**sicht
  - Informationsobjekte, die ein Informationssystem bearbeitet oder erzeugt
- **Funktion**ssicht
  - Funktionen / Aufgaben oder Tätigkeiten, die von einem Informationssystem ausgeführt werden
- **Steuerung**ssicht
  - Objekte der anderen Sichten zueinander in Beziehung
  - Beschreibt Geschäftsprozesse
- **Leistung**ssicht
  - Leistungen, die das Unternehmen als Ergebnis seiner Geschäftstätigkeit erbringt



Gegenüberstellung

1

Abbildung 3-4 Darstellungsarten von Geschäftsprozessen im Kontext Prozessmanagement (Hanschke, Giesinger, & Goetze, 2013)

2

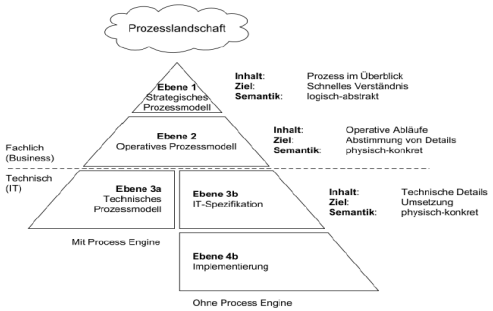


Abbildung 3-5 Das camunda BMNP Framework (Freund & Rücker, 2012, S. 15)

3

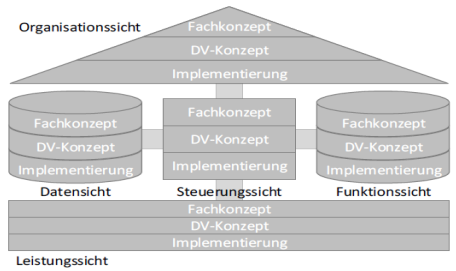
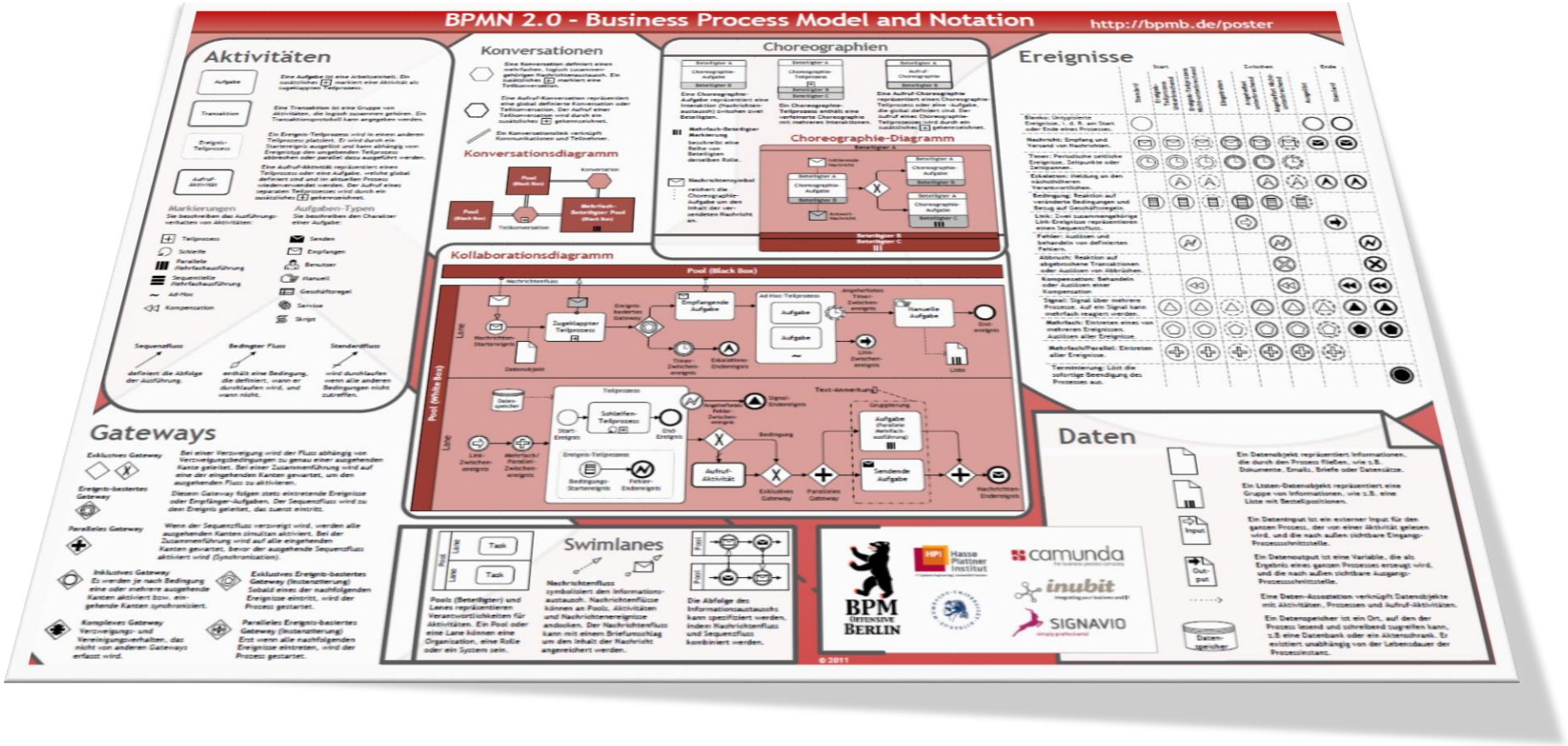


Abbildung 3-7 Das ARIS-Haus (Gadatsch, 2010) nach (Scheer, 1998)

	1	2	3
	(Hanschke, Giesinger, & Goetze, 2013)	(Freund & Rücker, 2012)	(Scheer, 1998) nach (Software AG, 2014)
Wofür wird der Bezugsrahmen schwerpunktmäßig eingesetzt?	Fachliches Business Process Management, Prozessdokumentation und -verbesserung	Business Process Management mit Prozessautomatisierung	Beschreibung von Informationssystemen mit dem Ziel, dieses zu implementieren
Fachliche Ebene	Swimlane-Diagramm zum Verständnis des normalen Ablaufs	Strategisches Prozessmodell zum Verständnis des normalen Ablaufs	Fachkonzept
	Prozessablaufdiagramm mit Details und Varianten	Operatives Prozessmodell mit Details und Varianten	
Technische Ebene	–	IT-Spezifikation oder Technisches Prozessmodell zur Ausführung auf einer Process Engine	DV-Konzept
		Implementierung	Implementierung

Abbildung 3-8 Überblick über Bezugsrahmen zur Prozessmodellierung

Business Process Model and Notation



Basiselemente der BPML

- Kategorien
  - Flussobjekte
    - Beschreiben den Ablauf von Geschäftsprozessen
    - Beispiele: Aktivitäten, Ereignisse und Gateways
  - verbindende Elemente
    - Sequenzfluss (wichtigstes verbindende Element)
    - zeitliche und logischen Reihenfolge
  - Swimlanes
  - Daten
  - Artefakte
- Eine empfehlenswerte Konvention ist es, für Aufgaben Beschriftungen der Form „Objekt- Verb“ und für Teilprozesse substantivierte Verben zu verwenden
-
- Für Ereignisse eignen sich Beschriftungen der Form „**Objekt-Partizip Perfekt**“

Abbildung 3-12 BPMN-Basiselemente der Kategorie Aktivitäten

Abbildung 3-13 BPMN-Basiselemente der Kategorie Ereignis

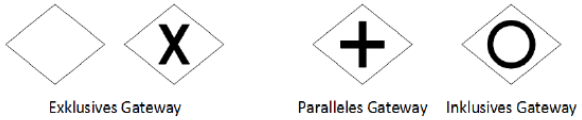


Abbildung 3-15 Gateways

Abbildung 3-21 Ein Pool mit drei Lanes

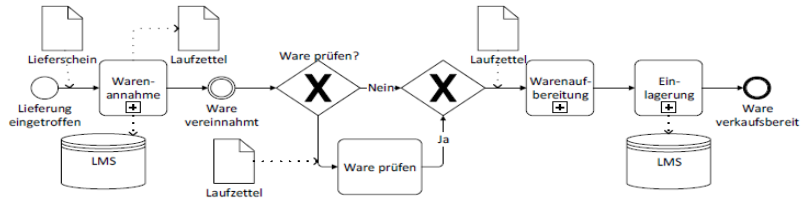


Abbildung 3-20 Wareneingangsprozess mit Datenobjekten

Fallbeispiel: Wareneingang bei der Firma Hoske GmbH

MHP
A PORSCHE COMPANY

- Die fiktive Hoske GmbH ist ein mittelständisches Unternehmen mit ca. 60 Mitarbeitern, das Handel mit Metallwaren im Umfeld B2B (Business-to-Business) betreibt.

- Die Hoske GmbH hat folgendes Geschäftsmodell:

    1. Ware wird eingekauft
    2. kontrolliert
    3. möglicherweise aufbereitet
    4. eingelagert
    5. weiterverkauft

- Die eingekaufte Ware hält das Unternehmen im Lager vorrätig.

- Die Kunden nehmen pro Auftrag typischerweise eher kleine Mengen ab. Das Unternehmen hat einen Standort und ein Lager, das in verschiedene Bereiche unterteilt ist.

- Dieses Fallbeispiel betrifft den **Wareneingang** der Hoske GmbH.

Wareneinnahme

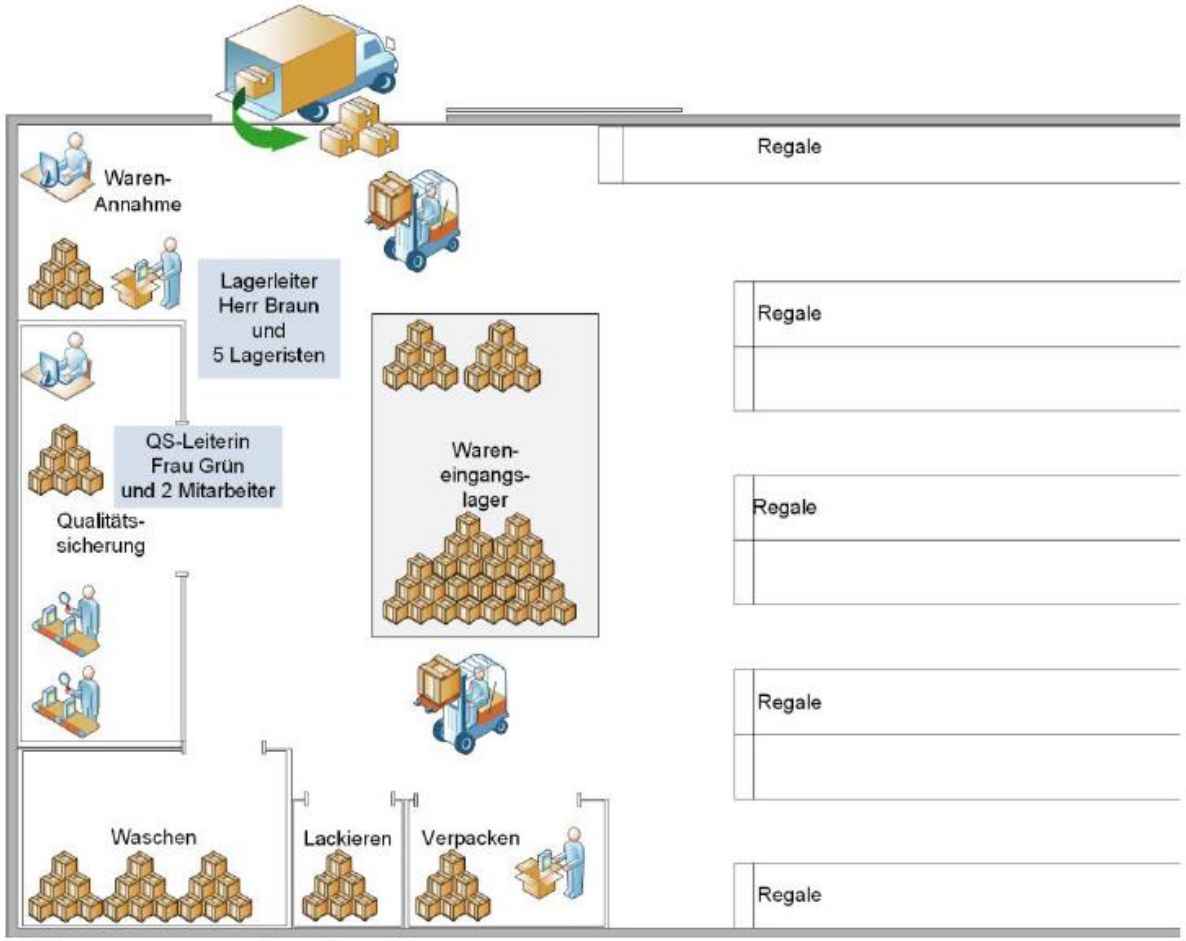


Abbildung 3-9 Das Lager der Hoske GmbH

Abbildung 3-10 Warenannahme- und Wareneingangsbereich (Paul-Georg Meister / pixelio.de)

Qualitätskontrolle

- Ein Teil der Ware kommt dann in die Qualitätssicherung zur Kontrolle
- Für diese Ware liegen Prüfpläne vor.
- Andere Ware kommt bereits geprüft vom Lieferanten und muss bei der Hoske nicht mehr geprüft werden.

Warenaufbereitung

- mehrere Schritte erforderlich
  - Waschen
  - Lackieren
  - Verpacken

Abbildung 3-11 Ware eingelagert im Hauptlager (Paul-Georg Meister / pixelio.de)

Einlagerung

- Ware verkaufsbereit
- Lagerung im Hauptlager
- Einbuchen ins Lagerverwaltungssystem
  - Bezeichnung der Ware,
  - die Anzahl der Verkaufsverpackungen
  - Lagerplatz
  - Die Nummer des Lagerplatzes ist am Regal im Lager angebracht.
  - Status im Lagerverwaltungssystem setzen auf „Freigegeben“.

- Jetzt können die Verkäufer im Lagerverwaltungssystem abrufen, das diese Ware für den Verkauf zur Verfügung steht.

Abbildung 3-11 Ware eingelagert im Hauptlager (Paul-Georg Meister / pixelio.de)

Betriebswirtschaftliche Problemstellung

- Die Verkäufer sehen im Lagerverwaltungssystem **nicht nur die eingelagerte Ware, sondern auch die noch nicht eingelagerte Ware im Wareneingang**.

- Wenn nun ein Verkäufer für einen Kunden dringend einen bestimmten Artikel benötigt, der zwar schon im Unternehmen eingegangen, aber noch nicht eingelagert ist, ruft er den Lagerchef Herrn Braun an und bittet ihn, diese Ware bevorzugt zu bearbeiten.

- Für den Lagerchef beginnt dann eine Suche: steht die Ware noch im Wareneingang, ist bereits in der Kontrolle gelandet, beim Waschen oder beim Lackieren, oder ist sie möglicherweise schon fertig verpackt und nur noch nicht eingelagert?

- Diese Suche kostet viel Zeit und bringt viel Unruhe ins Lager.

- Deshalb möchte die Hoske GmbH diesen **Ablauf verbessern**.

Prozessmodell Wareneingang

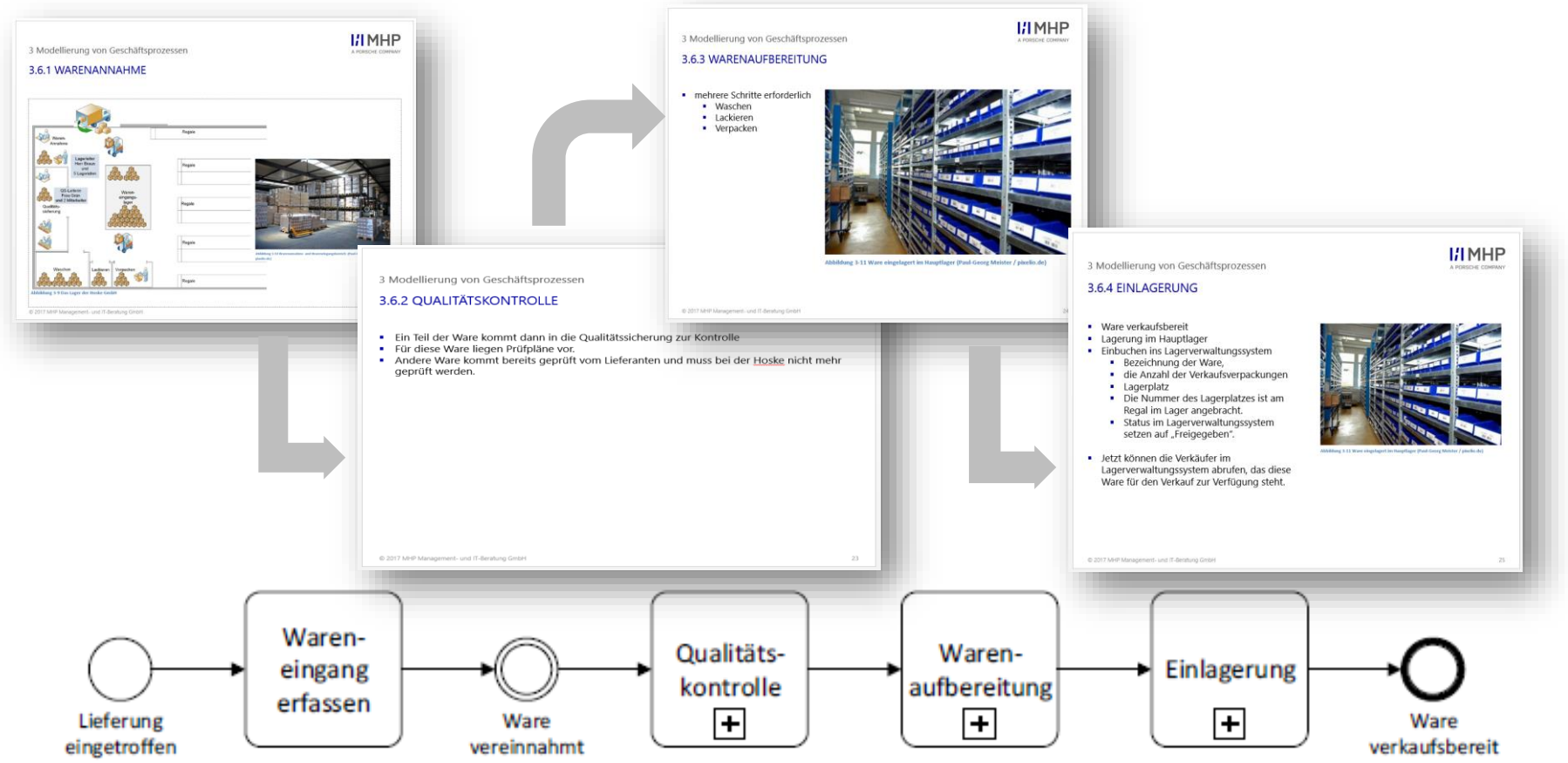


Abbildung 3-14 Beispiel BPMN: Wareneingang bei der Hoske GmbH

Gateways

- **Ziel**:
- Modellierung von Geschäftsprozesse mit
  - Alternativen
  - parallelen
  - optionalen Abläufen

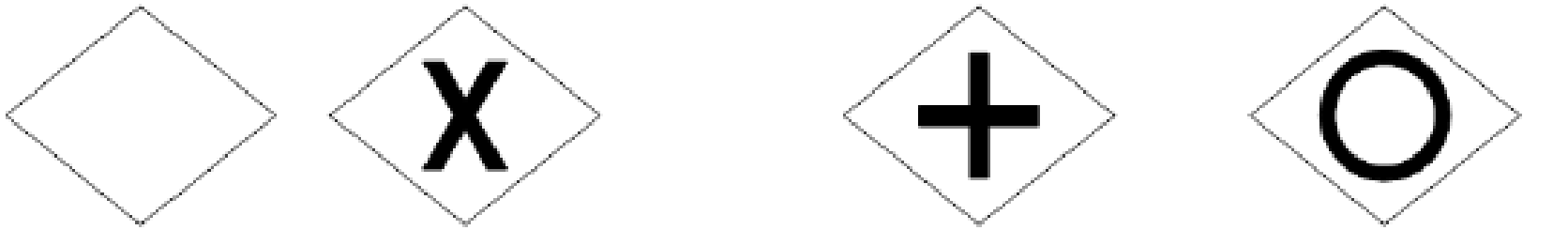


Abbildung 3-15 Gateways

Exklusives datenbasiertes Gateway

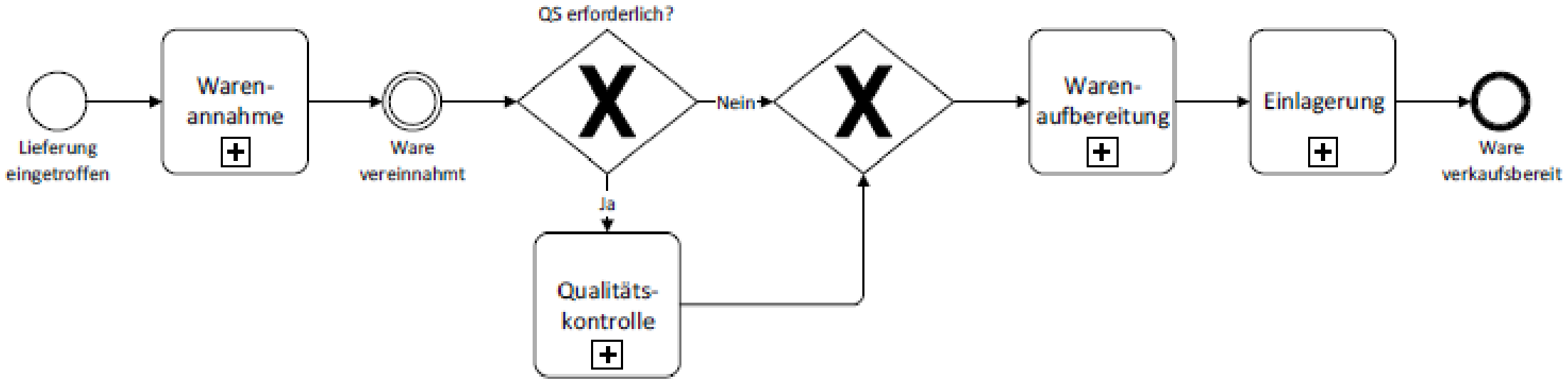


Abbildung 3-16 Wareneingangsprozess mit verzweigendem und zusammenführendem exklusivem datenbasiertem Gateway

Exklusives datenbasiertes Gateway

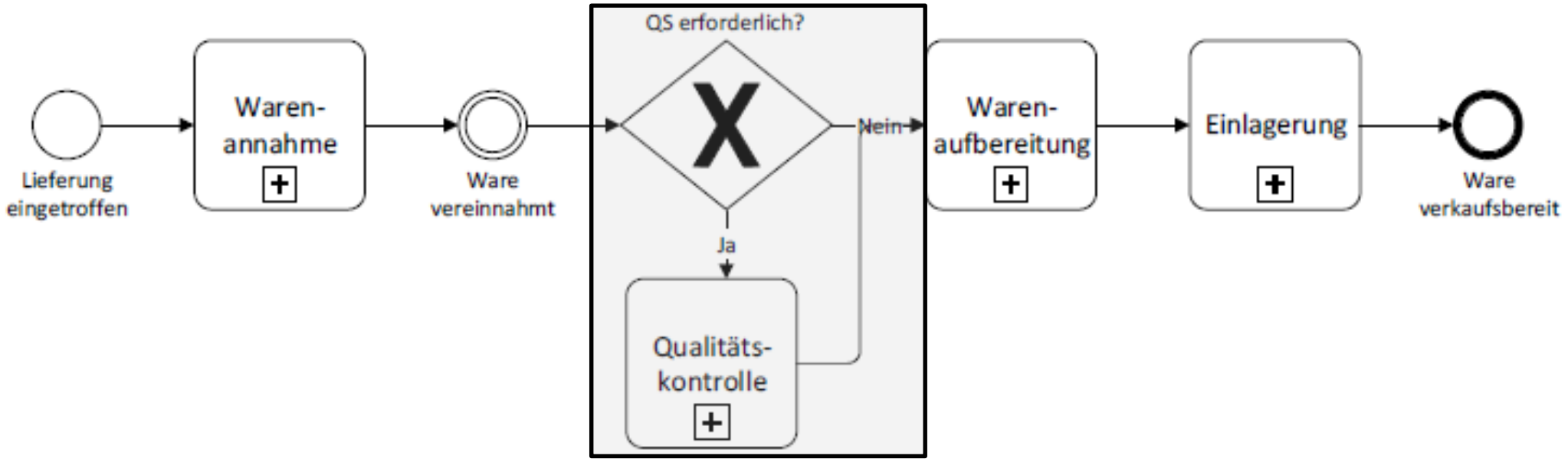


Abbildung 3-17 Wareneingangsprozess mit verzweigendem exklusivem datenbasiertem Gateway

Paralleles Gateway

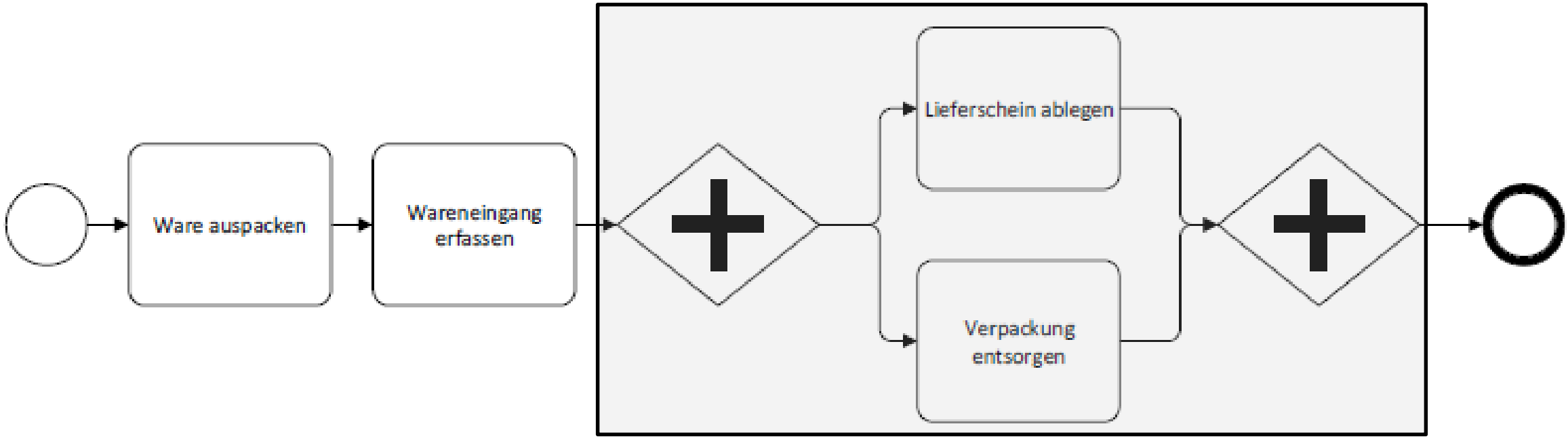


Abbildung 3-18 Prozess mit parallelen Gateways

Inklusives datenbasiertes Gateway [ODER / OR-Gateway

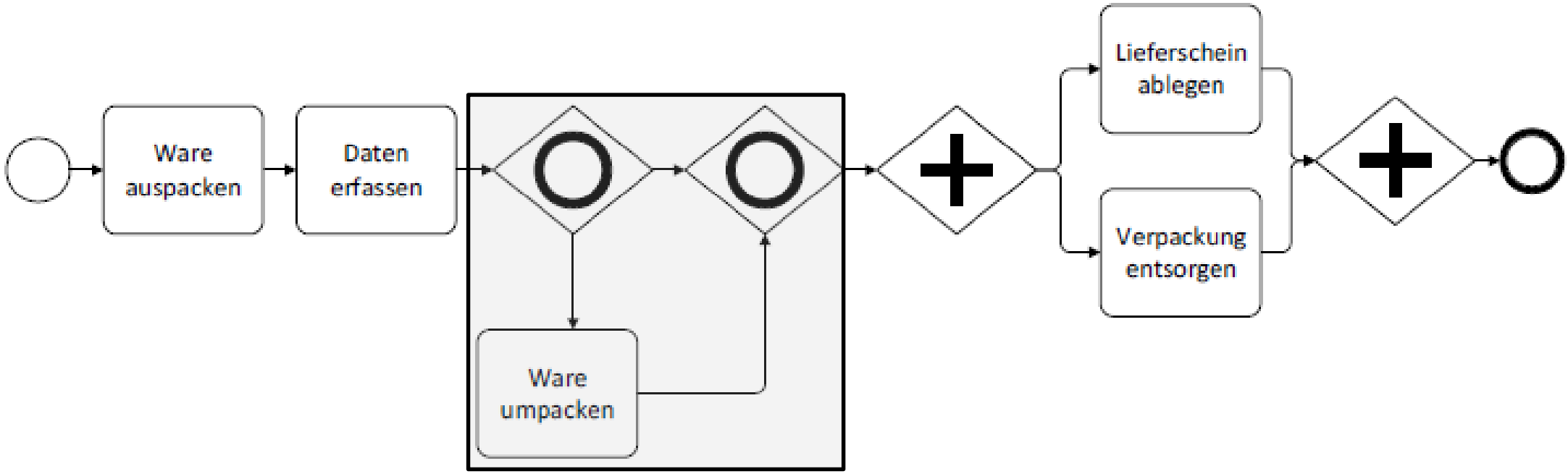


Abbildung 3-19 Prozess mit inklusiven datenbasierten Gateways

Datenobjekte, Datenspeicher und Artefakte

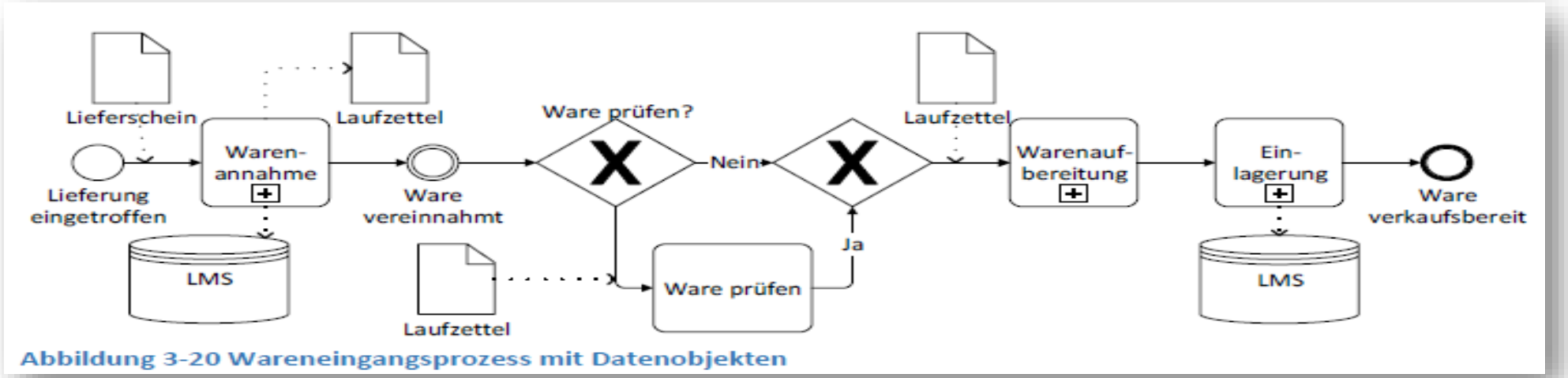
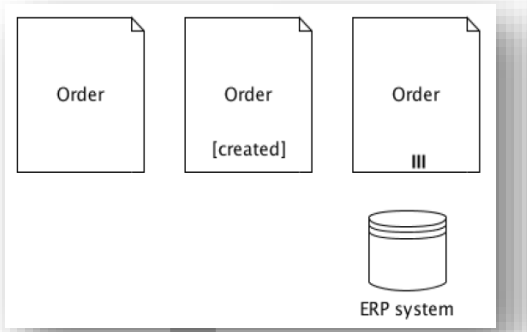
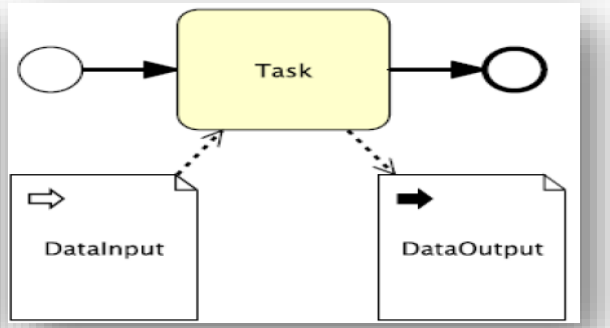
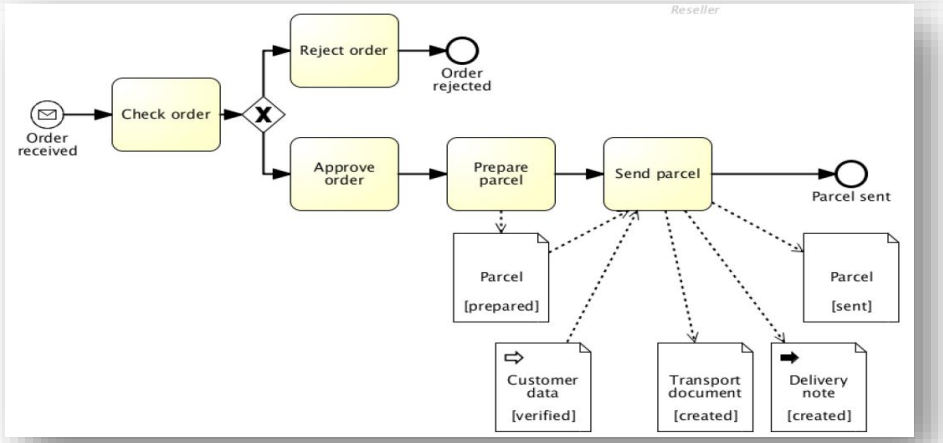


Abbildung 3-20 Wareneingangsprozess mit Datenobjekten







Pools und Swimlanes

- Mittel, um Zuständigkeiten und Zusammenwirken in Prozessszenarien mit mehreren Akteuren zu modellieren

- Grundprinzip:
  - **pro Prozess ein Pool**, der in mehrere Lanes unterteilt sein kann
  - Jeder Prozess gehört genau **einem Teilnehmer**

Abbildung 3-21 Ein Pool mit drei Lanes

Pools und Swimlanes

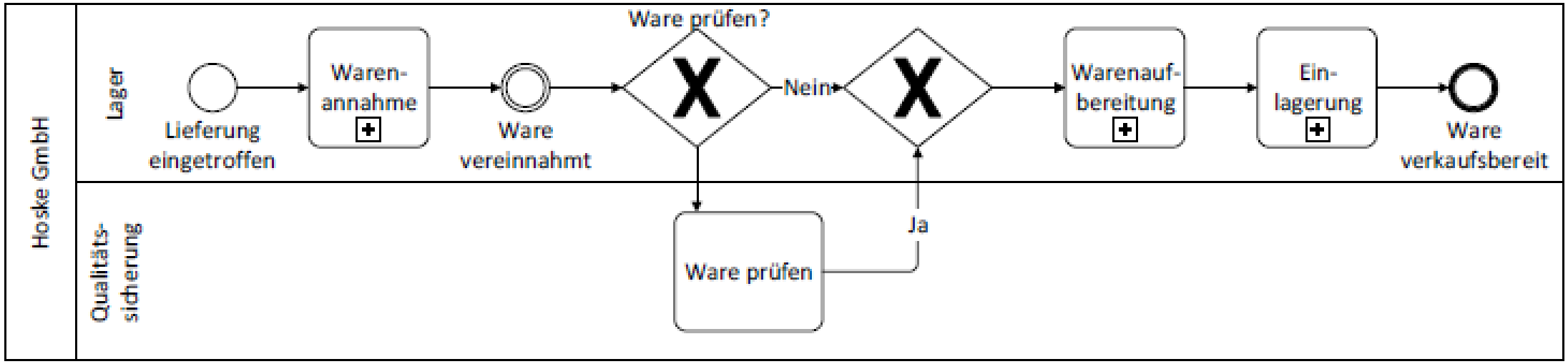
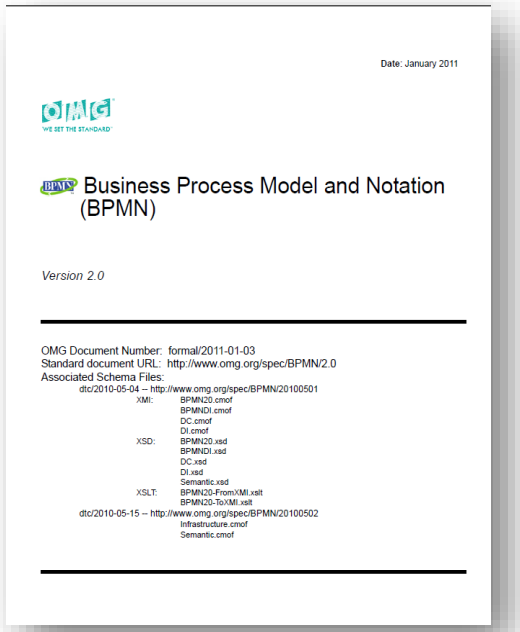


Abbildung 3-22 Wareneingangsprozess mit Pool und Swimlanes

BPML / BPMN @ OMG


Date: January 2011

OMG

Business Process Model and Notation (BPMN)

Version 2.0

OMG Document Number: formal/2011-01-03
Standard document URL: http://www.omg.org/spec/BPMN/2.0
Associated Schema Files:
dtc/2010-05-04 -- http://www.omg.org/spec/BPMN/20100501
XMI: BPMN20.cmof
BPMNDI.cmof
DC.cmof
DI.cmof
XSD: BPMN20.xsd
BPMNDI.xsd
DC.xsd
DI.xsd
Semantic.xsd
XSLT: BPMN20-FromXMI.xslt
BPMN20-ToXMI.xslt
dtc/2010-05-15 -- http://www.omg.org/spec/BPMN/20100502
Infrastructure.cmof
Semantic.cmof

Dokument: http://www.omg.org/spec/BPMN/2.0/

This specification represents the amalgamation of best practices within the business modeling community to define the notation and semantics of **Collaboration** diagrams, **Process** diagrams, and **Choreography** diagrams. The intent of **BPMN** is to standardize a business process model and notation in the face of many different modeling notations and viewpoints. In doing so, **BPMN** will provide a simple means of communicating process information to other business users, process implementers, customers, and suppliers.

Private (interne) & Public (öffentliche) Business Prozesse

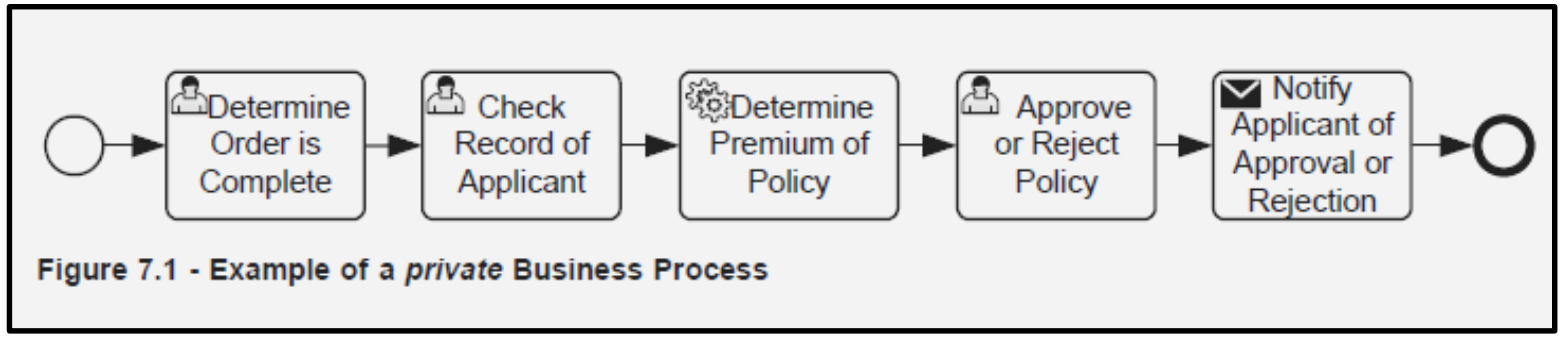


Figure 7.1 - Example of a *private* Business Process

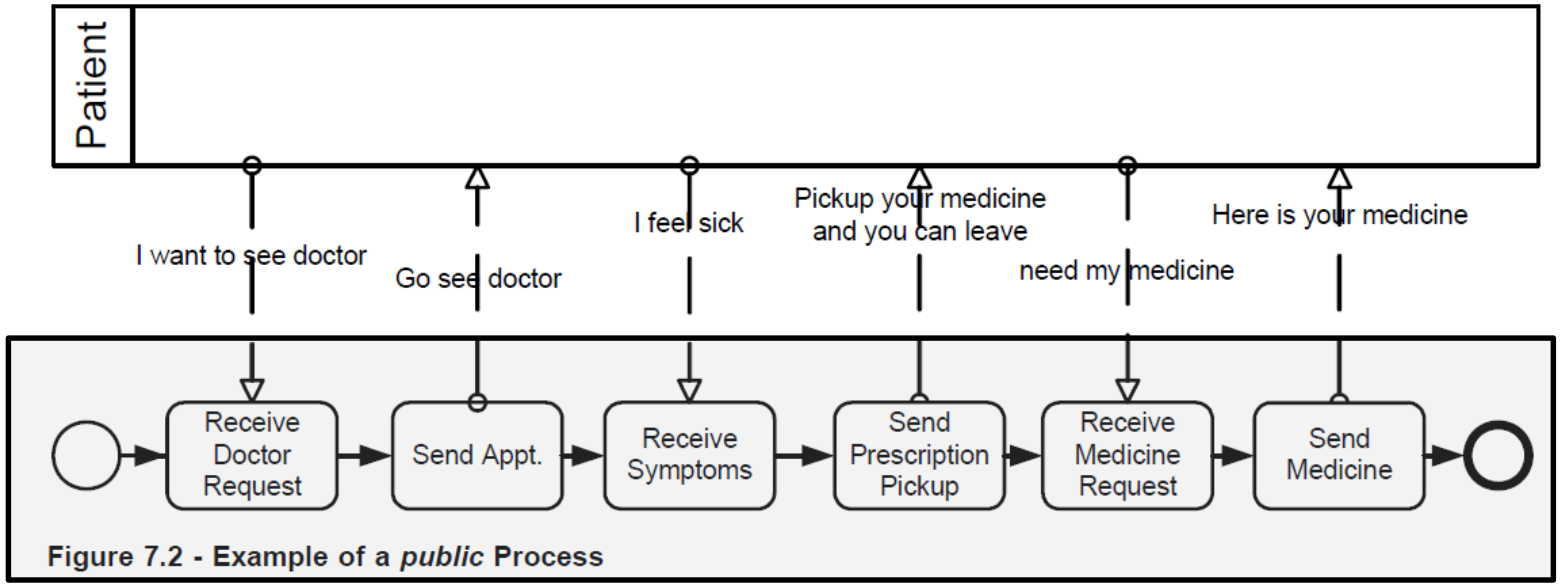


Figure 7.2 - Example of a *public* Process

Kollaborationsdiagramm zur Darstellung des Nachrichtenaustausches

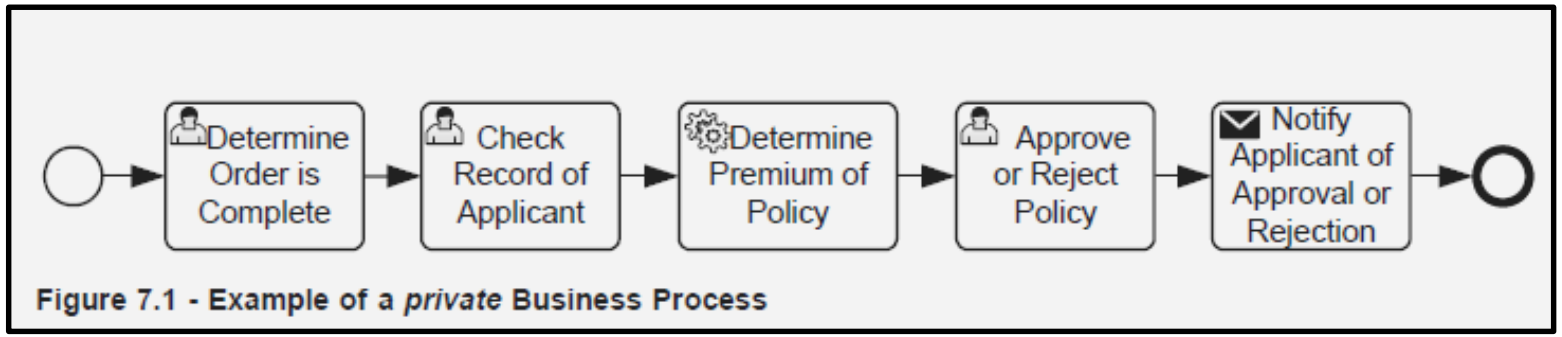


Figure 7.1 - Example of a *private* Business Process

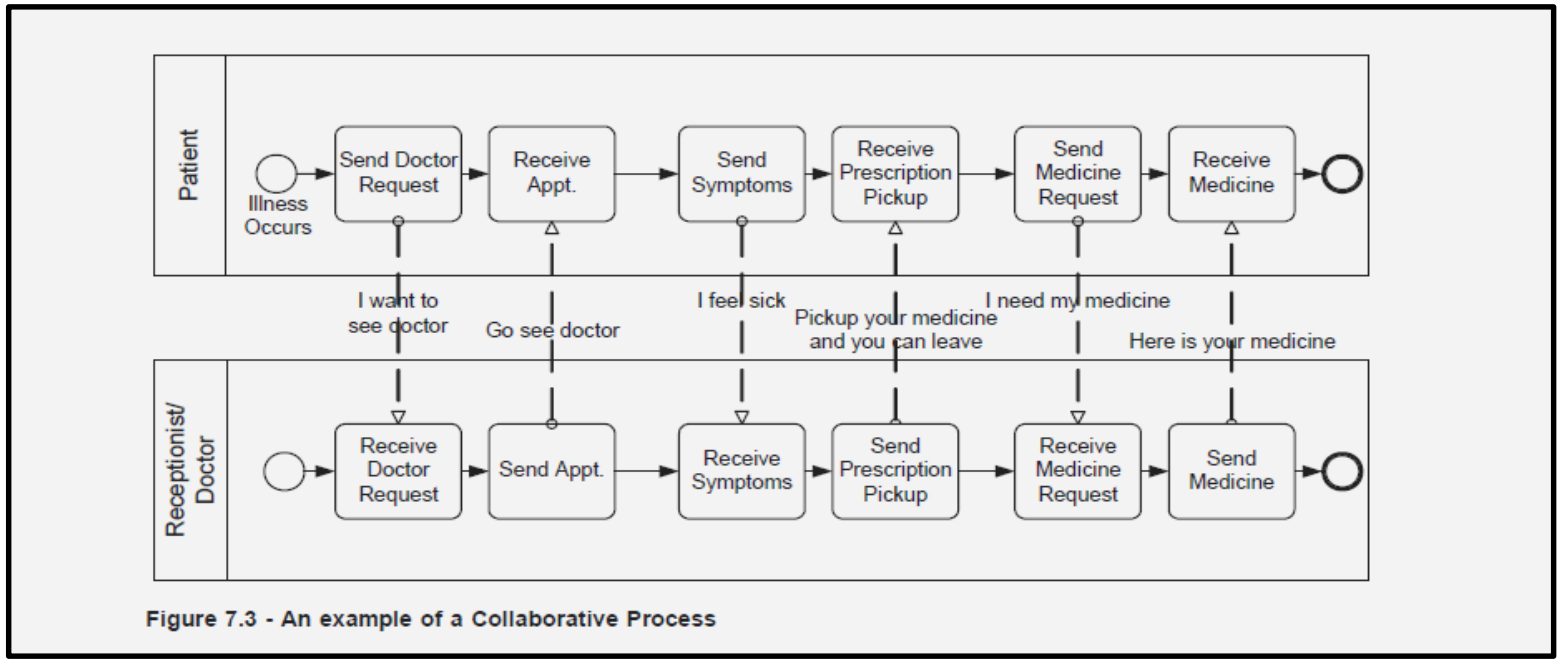


Figure 7.3 - An example of a Collaborative Process

Process Choreography

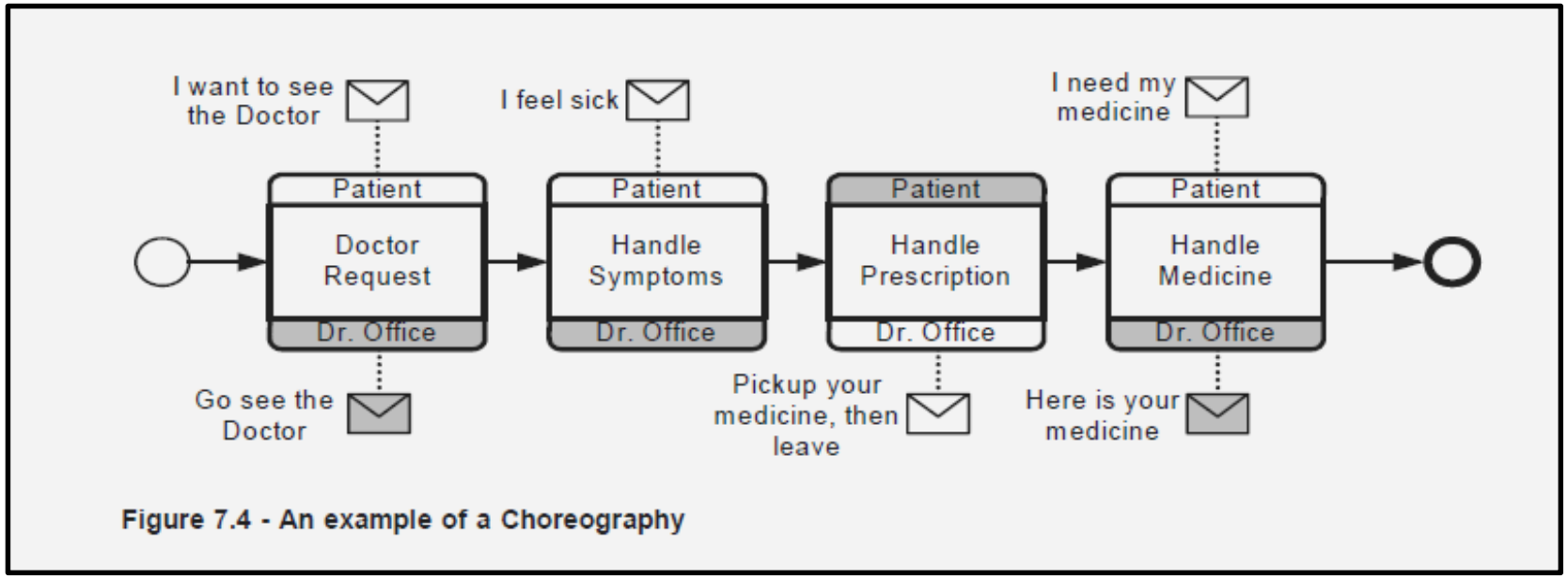


Figure 7.4 - An example of a Choreography

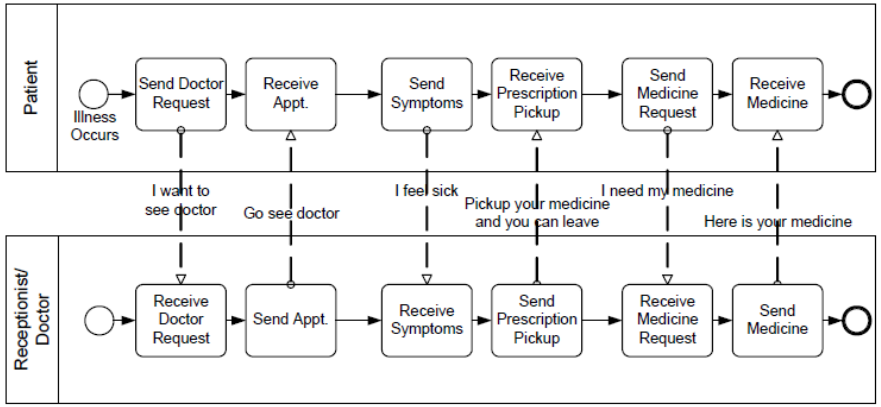


Figure 7.3 - An example of a Collaborative Process

Konversations- / Kommunikationsdiagram (Conversation Diagram)

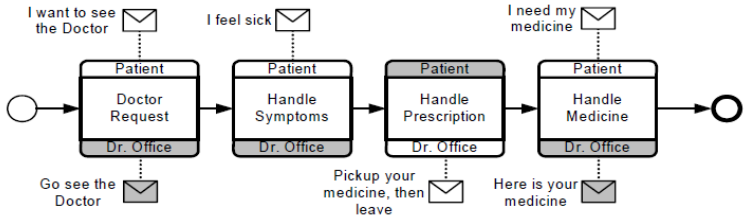


Figure 7.4 - An example of a Choreography

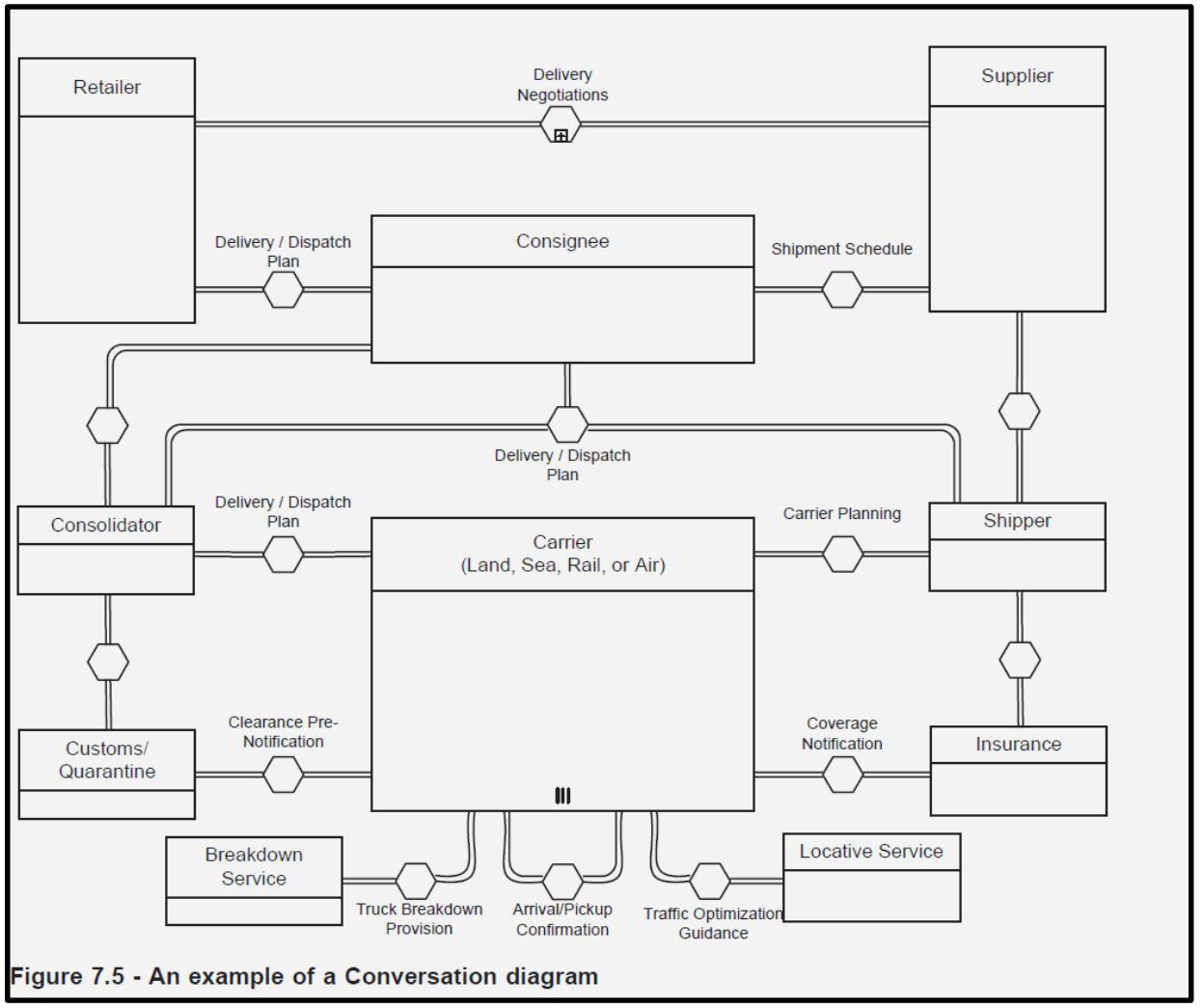


Figure 7.5 - An example of a Conversation diagram

MHP
A PORSCHE COMPANY

INTEGRIERTE MODELLIERUNG KOMPLEXER SYSTEME

Datenbanken und Datenmodellierung

UNIVERSITATEA BABEȘ-BOLYAI

Hugo Colceag I Bachelor Studiengang Informatik

MHP
A PORSCHE COMPANY

Anwendungssysteme

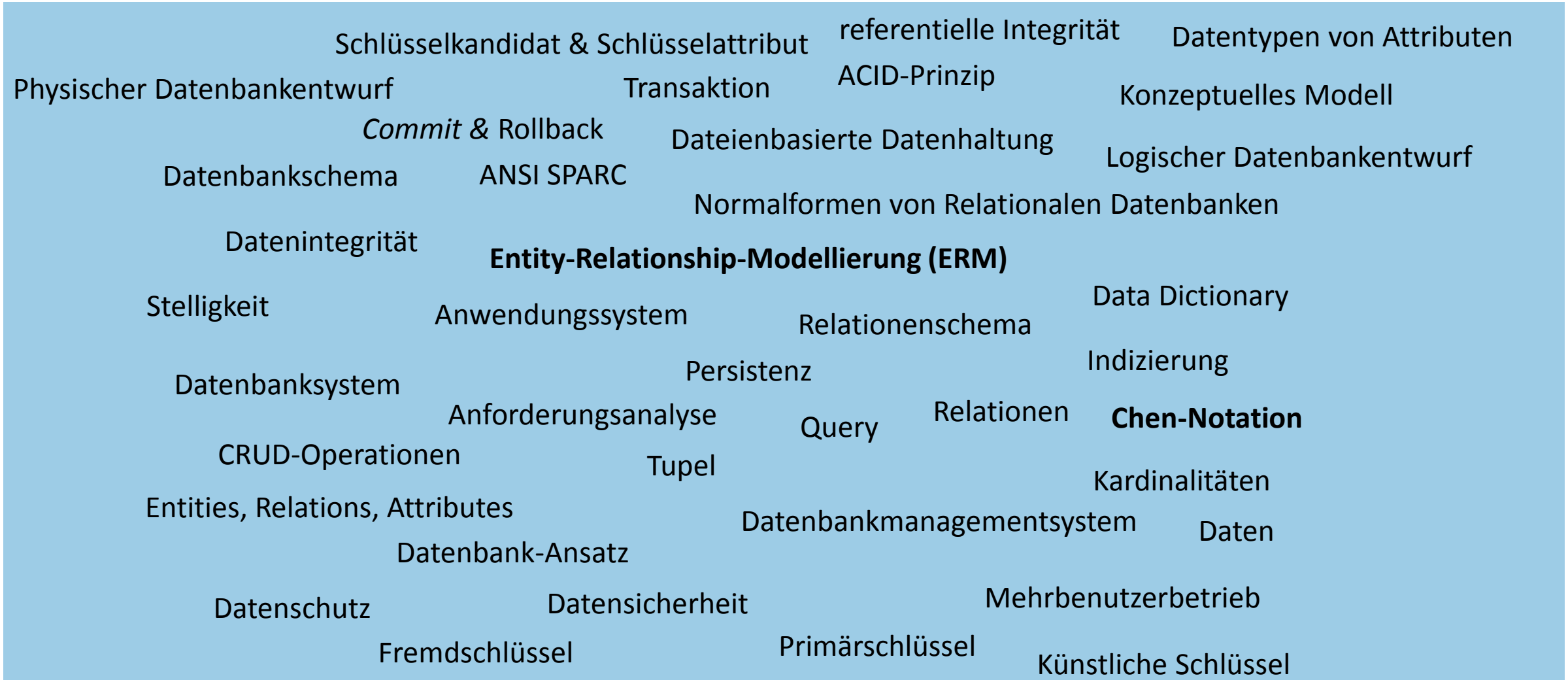
Vorgehensmodelle & ganzheitliche (integrierte) Modellierung

Business Architecture
Geschäftsprozesse - BPML

Data Architecture
Datenbanken - ERM

Technology & Application Architecture
UML

Mindmap - Datenbanken und Datenmodellierung



Lernziele

- ✓ Formen der Datenspeicherung
- ✓ Funktionsweise einer Datenbank und Datenbankmanagementsystem
- ✓ Verständnis relationales Datenmodell
- ✓ Überblick über Entwurf einer Datenbank
- ✓ Entity-relationship-Modellierung

Agenda

MHP
A PORSCHE COMPANY

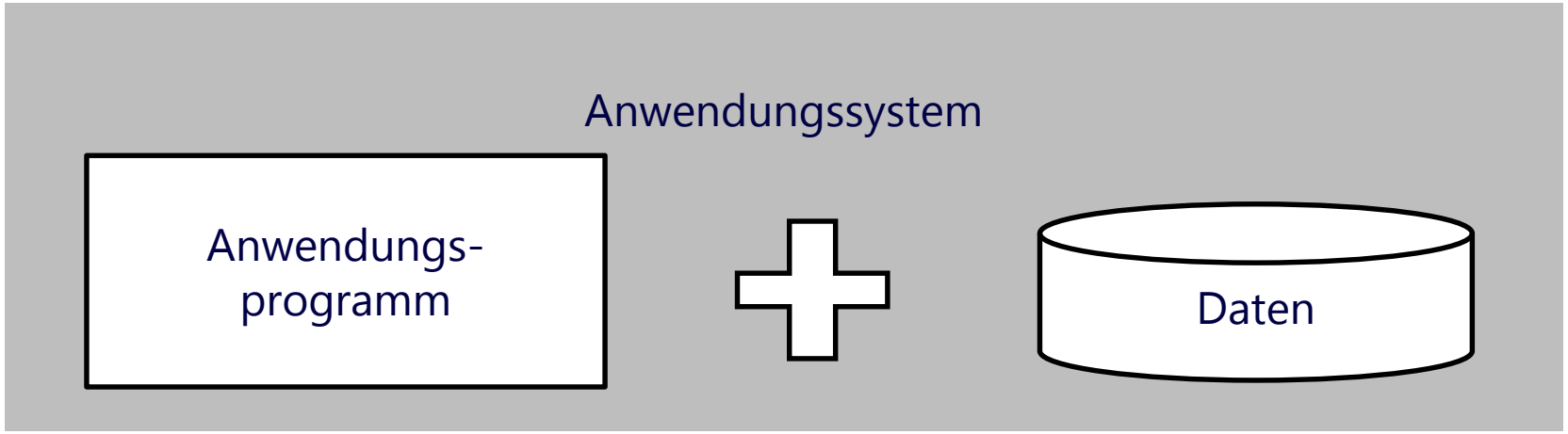
1. **Daten und Datenbanken**
2. Datenbankmanagementsysteme
3. Das relationale Datenmodell
4. Datenbankentwurf

1.1 Daten

- Daten (nach Din 44300) Daten sind Zeichen oder kontinuierliche Funktionen, die aufgrund von bekannten oder unterstellten Abmachungen oder vorrangig zum Zwecke der Verarbeitung Informationen darstellen.

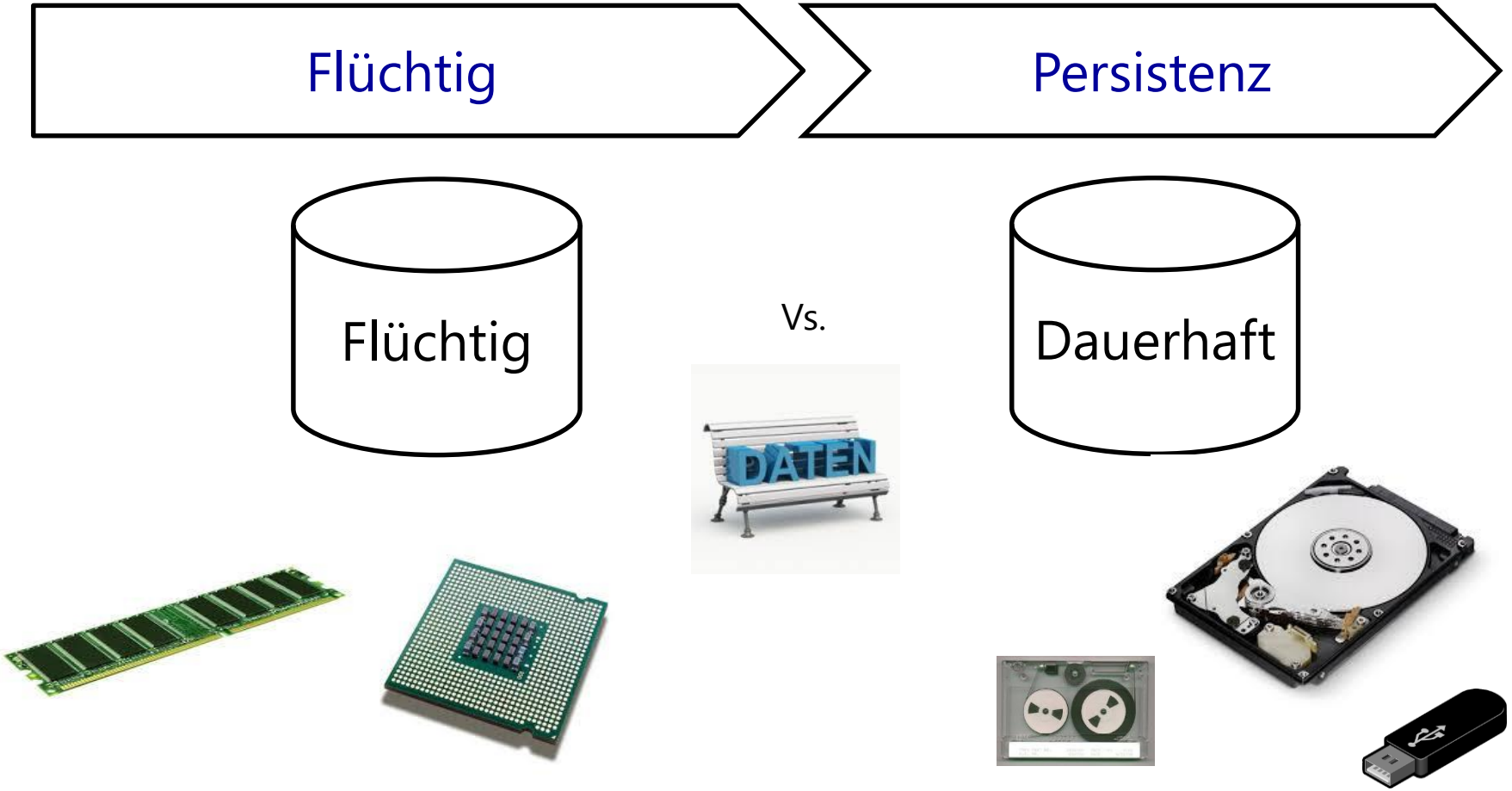
- Daten nach ISO/IEC 2382-1 „a reinterpretable representation of information in a formalized manner, suitable for communication, interpretation or processing“

- Anwendungssystem = Daten plus Anwendungsprogramm, das die Daten verwendet und bearbeitet



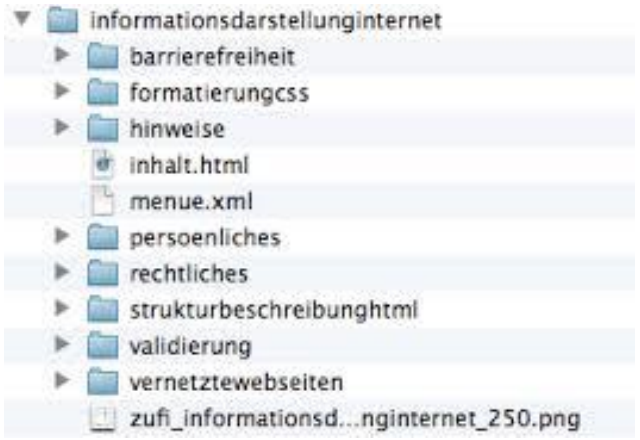
1.2 Dauerhafte Datenspeicherung – Persistenz

- Ablageort in datenverarbeitende Anwendungsprogramme



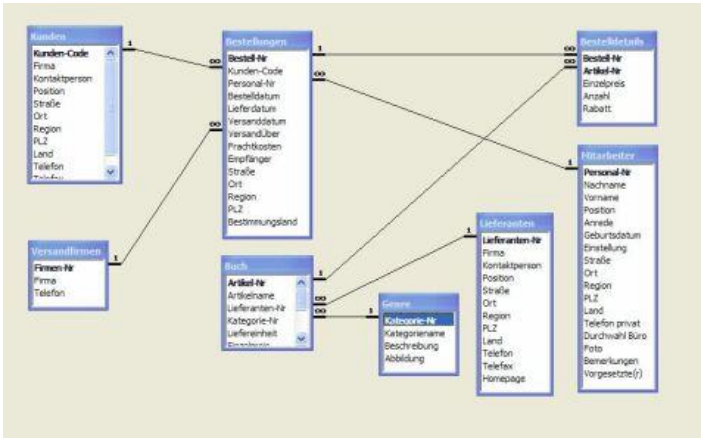
1.2.1 Datenhaltung

- „dateienbasierte Datenhaltung“



Vs.

„Datenbank-Ansatz“



1.2.1 Dateienbasierte Datenhaltung

MHP
A PORSCHE COMPANY

- Beispiel 1.1 Dateienbasierte Datenhaltung für die Schulungsplanung

  Die Hoske GmbH bietet häufig Schulungen zu ihren Produkten in verschiedenen Städten an. Herr Mayer ist dafür zuständig, die Räumlichkeiten und das Catering in den Pausen zu organisieren. Dazu hat er ein Excel-Tool entwickelt, in dem er Listen mit Anbietern von Veranstaltungsräumen und von Catering-Unternehmen führt. Auch die geplanten Schulungstermine sind hinterlegt. Herr Mayer nutzt das Tool, um die Anfragen und Zusagen der Anbieter für die einzelnen Schulungstermine zu verwalten.

  Frau Singers Aufgabe ist es, Dozenten für die Schulungen zu organisieren. Sie hat dafür auch ein Excel-Tool, das Informationen über Dozenten, ihre Kompetenzbereiche und Gagen enthält. Sie nutzt ihr Tool, um die Anfragen und Zusagen der Dozenten für die einzelnen Schulungstermine zu verwalten.

  Linda erfasst die Anmeldungen zu den Schulungen. Eine Schulung findet statt ab 5 Teilnehmern. Pro Termin können maximal 20 Personen teilnehmen.

1.2.2 Der Datenbank-Ansatz zur Datenhaltung

- Datenbank ist eine Sammlung von strukturierten, inhaltlich zusammenhängenden Daten.
- Die Struktur nennt man auch „Datenmodell“.
- Eine Datenbank beschreibt einen wohldefinierten Weltausschnitt (Domain)
- Eine zufällige Datensammlung wird nicht als Datenbank bezeichnet.

- alle Daten, die zu einem bestimmten Anwendungsbereich gehören
  - Zentral
  - Für alle Benutzer und Anwendungen,
  - gemeinsame Datenspeicherung
  - Abruf aus der gemeinsamen Datenspeicherung

- Weltausschnitt vollständig abgebildet
  - alle Daten enthalten, die für den Anwendungsbereich relevant sind

1.3 Übungen

- Beispiel 1.2 Die Miniwelt der Schulungsplanung

  Die Schulungen der Hoske GmbH erfreuen sich reger Nachfrage und sind profitabel. Um dem wachsenden Aufwand gerecht zu werden und um Fehlplanungen zu vermeiden, lässt die Hoske die Planungstools verbessern. Alle Daten, die Schulungen betreffen, wie Räumlichkeiten, Termine, Dozenten, Catering-Anbieter, aber auch Teilnehmer, Schulungsunterlagen, Kugelschreiber, Schreibblöcke und Werbegeschenke und alle Anfragen und Zusagen sind jetzt **in einer gemeinsamen Datenbank** abgelegt.

1.3 Übungen

- Aufgabe 1.1

  Für die Aufgaben der Schulungsplanung, die in Beispiel 1.1 und Beispiel 1.2 beschrieben sind:

  Welche Vorteile bringt der Datenbank-Ansatz gegenüber der dateibasierten Datenhaltung mit sich?

  Geben Sie Beispiele von Vorgängen oder Situationen, die die Mitarbeiter der Hoske dank des datenbankbasierten Ansatzes besser bearbeiten können als mit ihren eigenentwickelten Tools.

  Hat der Datenbank-Ansatz auch Nachteile? Nennen Sie sie.

1.3 Übungen

- Beispiel 1.3 Datenhaltung im Customer Relationship Management

  Dateienbasierte Datenhaltung in der Vertriebsabteilung des Unternehmens Hoske: jeder Vertriebler pflegt seine eigenen Notizen in eigenen Dateien, die er so aufbaut und organisiert, wie es ihm oder ihr am sinnvollsten scheint. Manche nutzen dazu Word-Dokumente mit speziellen Vorlagen, in denen sie die Informationen erfassen, und legen die Dokumente in ihrer eigenen, für sie sinnvollen Verzeichnisstruktur ab. Andere pflegen die Informationen strukturiert in selbst entwickelte Excel-Tabellen ein. Einige mobile Außendienstmitarbeiter verwenden eine Notizbuch-App auf ihrem Smartphone mit Datenspeicher in der Cloud.

  Datenbank-Ansatz: es gibt eine gemeinsame Datenbank, in die jeder Vertriebler die Infos zu seinen Kundenkontakten einstellt. Jeder Vertriebsmitarbeiter hat darauf Zugriff und kann im Falle von Urlaub, Krankheit oder Kundenübernahmen alle Informationen abrufen, die sie oder er benötigt. Speziell entwickelte Software stellt dazu Bedienoberflächen zur Verfügung.

1.3 Übungen

- Aufgabe 1.2

  Im Vertrieb ist es wichtig, seine Kunden genau zu kennen und zu wissen, was sie wollen. Vertriebsmitarbeiter verfassen daher Notizen über ihre Kundenkontakte, sogenannte Besuchszettel, Messeberichte oder einfache Gesprächsnotizen.

  Zu den wichtigen Informationen über einen Kundenkontakt gehören zum Beispiel der Name des Kunden, den Anlass des Kontakts (Messe, Neuauftrag, Reklamation etc.), das Medium (persönlich, telefonisch, Email etc.), den Initiator (Kunde oder Vertriebs- mitarbeiter), den Ort und die Zeit, die Gesprächspartner und die besprochenen Themen und Übereinkünfte.

  Die Vertriebsmitarbeiter des Unternehmens Hoske im Innendienst, insgesamt 5 Personen, lösen die Aufgabe, die Informationen zu Kundenkontakten zu verwalten, mithilfe einer Excel-Arbeitsmappe „Kundenkontakte“.

1.3 Übungen

- Aufgabe 1.2

a) Wie könnte die Excel-Mappe „Kundenkontakte“ konkret aussehen?

b) Welche Daten benötigt und erzeugt die Anwendung?

c) Bitte entwickeln Sie zu zweit oder zu dritt ein Konzept für eine solche Excel-Arbeitsmappe (direkt in Excel oder als Skizze).

d) Beschreiben Sie in Worten detailnachvollziehbar detailliert die Standardprozesse, wie Mitarbeiter Daten in die Mappe einpflegen und wie sie Informationen abrufen.

1.3 Übungen

Aufgabe 1.2

a) Ist die Lösung prinzipiell gut geeignet, die Standardprozesse zu unterstützen?

b) Welche Vorteile und welche Schwächen weist die Lösung auf?

c) Welche Verbesserungs- und Erweiterungsmöglichkeiten gibt es? Beschreiben Sie weitere mögliche Anwendungsfälle für das System, also Ziele, die Mitarbeiter mithilfe eines Systems wie „Kundenkontakte“ möglicherweise erreichen wollen.

d) Schlagen Sie aufbauend auf c) einige zusätzliche Funktionen vor, die für die Vertriebsmitarbeiter nützlich sein könnten. Wie gut ist dieses Excel-basierte System geeignet, um diese zusätzlichen Funktionen umzusetzen?

e) Handelt es sich bei dieser Lösung um eine **dateienbasierten** Datenhaltung oder um den **Datenbank-Ansatz**? Bitte diskutieren Sie.

Agenda

MHP
A PORSCHE COMPANY

1. Daten und Datenbanken
2. **Datenbankmanagementsysteme**
3. Das relationale Datenmodell
4. Datenbankentwurf

2.1 Grundfunktionen

- Ein Datenbankmanagementsystem (DBMS) ist ein allgemein einsetzbares Programmpaket, das die Definition, Erzeugung, Manipulation und gemeinsame Nutzung von Datenbanken durch mehrere Benutzer und Anwendungen unterstützt. (Elmasri & Navathe, Fundamentals of Database Systems, 2011, S. 5)

1. **Allgemein einsetzbar**: prinzipiell unabhängig vom Anwendungsbereich

2. Definition:
   a) Datentypen, Strukturen und Einschränkungen für die Daten festzulegen
   b) Definition und Beschreibung der Datenbank in einem **Datenkatalog** oder ***Data Dictionary***
   c) Informationen: Meta-Daten (das bedeutet „Daten über Daten“, also „Daten, die Daten beschreiben“

3. **Erzeugung**: Daten gemäß der Datendefinition auf einem Speichermedium ablegen, das unter Kontrolle des DBMS steht

4. **Manipulation**: Operationen auf Daten ausführen → *Query*

2.1 Grundfunktionen

4. **Manipulation**: Operationen auf Daten ausführen → *Query*

- *Query:* „Anfragen an die Datenbank stellen" / *querying a database*

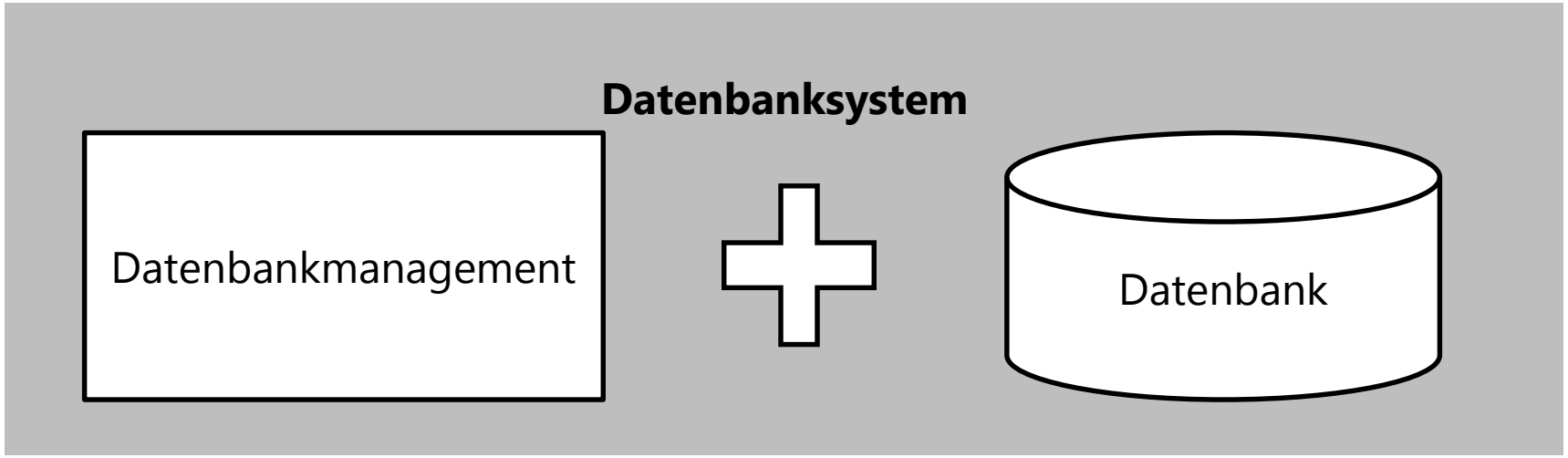
- CRUD-Operationen = Create-Read-Update-Delete:

  - **C**reate: Daten eintragen
  - **R**ead: Daten abrufen
  - **U**pdate: Daten ändern
  - **D**elete: Daten löschen

5. **Gemeinsame Nutzung**: mehrere Benutzer und Softwareanwendungen können Datenbank gleichzeitig nutzen (DBMS koordiniert die Zugriffe)

2.2 Datenbanksystem

- Ein System bestehend aus einem Datenbankmanagement plus einer Datenbank nennt man ein Datenbanksystem.



2.2 Datenbanksystem

- Beispiel 2.1 Excel als Datenbankmanagementsystem?

  Eine Datenbank kann auch in Excel oder einer anderen Tabellenkalkulationssoftware verwaltet werden:

  Die **Definition** der Datenbank besteht darin festzulegen, wie die Daten auf einem oder mehreren Arbeitsblättern in welchen Zellen in welchen Zahlenformaten abgelegt werden sollen

  Erzeugung der Datenbank bedeutet, die Strukturen in der Excel-Arbeitsmappe anzulegen und die Arbeitsmappe dauerhaft zu speichern.

  Die Benutzer können nun Daten in der Datenbank suchen und lesen, neue Daten eintragen und bestehende **Daten ändern** oder löschen.

  Gemeinsame Nutzung, auch durch andere Softwareanwendungen, ist ebenfalls möglich, erfordert aber viel Programmierung oder organisatorischen Aufwand bei der Koordination der Zugriffe. Hier, aber auch an anderen Stellen, stößt eine Tabellenkalkulationssoftware als Datenbankmanagementsystem an Grenzen, wie im Folgenden noch deutlich werden wird.

2.3 Einsatz eines Datenbankmanagementsystem

- **Wie wird ein DBMS eingesetzt?**
- **Wer benutzt und bedient es wie?**
- **Wie wirkt es mit anderen Anwendungen zusammen?**

2.3.1 Zugriff auf die Daten nur über das DBMS

MHP
A PORSCHE COMPANY

- Funktion des Datenbankmanagementsystems [DBMS]
  - Daten
    - dauerhaft speichern (indem es sie in Dateien auf geeignete Speichermedien schreibt)
    - manipulierbar machen (legt die Daten in speziellen Strukturen ab)
    - bereitstellen
  - Die Dateien mit den Daten schreibt und liest deshalb ausschließlich das Datenbankmanagementsystem

2.3.2 Verwaltung von mehreren Datenbanken

Ein Datenbankmanagementsystem kann bei Bedarf auch mehrere Datenbanken verwalten (vergleiche auch 2.3.3).

2.3.3 Datenbanksystem als Dienstprogramm für andere Anwendungen

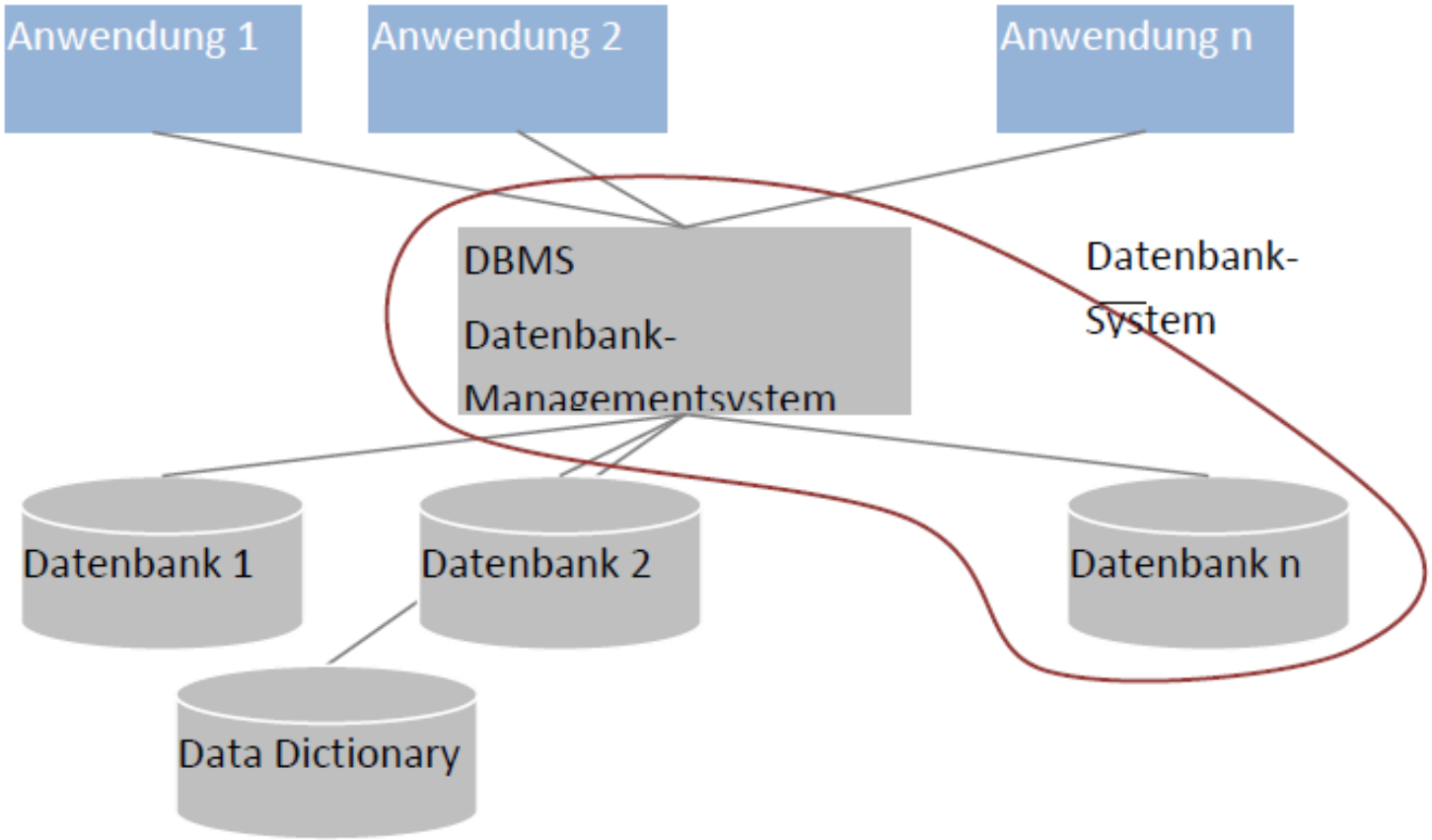


Abbildung 2-1 Aufbau und Einsatz eines Datenbanksystems

2.3.4 Trennung von Daten und Anwendungsprogrammen

- Anwendungsprogramme und Datenbank weitgehend voneinander entkoppelt
- Datenbank: speichert Informationen als **anwendungsunabhängige „Miniwelt“**
- Anwendungssoftware und Datenbank können getrennt voneinander entwickelt werden

Datenbank:

- Aufgabe: erforderliche Informationen bereitstellen
  - Struktur ergibt sich jedoch aus der Logik der abgebildeten Miniwelt
  - !!!! **NICHT** aus den Anforderungen der Anwendungssoftware

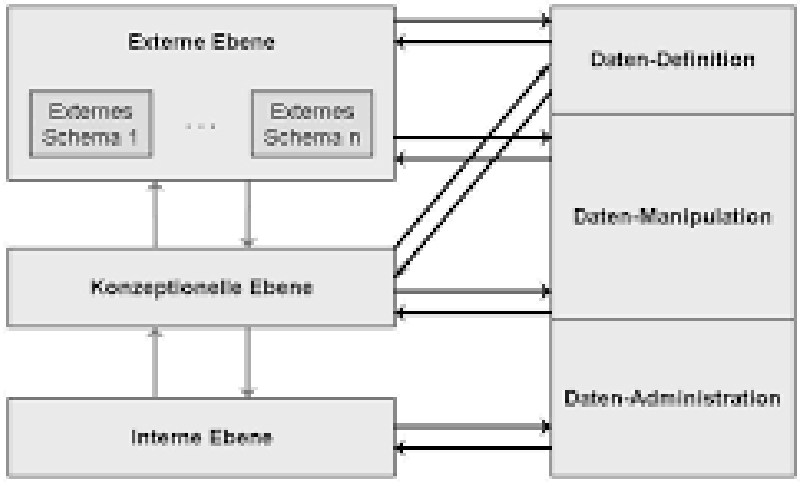
- unabhängig von einer bestimmten Anwendungssoftware
- kann von verschiedenen Anwendungsprogrammen genutzt werden
- liefert auch ohne Anwendungsprogramm verständliche Informationen

2.3.5 3-Ebenen-Architektur nach ANSI SPARC

Die Drei-Ebenen-Architektur wurde 1975 vom Standards Planning and Requirements Committee (**SPARC**) des American National Standards Institute (**ANSI**) entwickelt.

Die **Vorteile** sind:

- Logische Datenunabhängigkeit
- Physikalische Datenunabhängigkeit



Externe Ebene	Benutzersicht
Logische/Konzeptionelle Ebene	Datenmodellierung, Miniwelt
Interne Ebene	Physikalische Speicherung, Indizes

2.4 Weitere Funktionen von Datenbankmanagementsystemen

MHP
A PORSCHE COMPANY

Da Datenbankmanagementsysteme mittlerweile seit einem halben Jahrhundert im Einsatz sind und immer weiterentwickelt wurden, handelt es sich dabei um **hochspezialisierte** und **mächtige Systeme**, ohne die viele Anwendungen gar nicht möglich wären. Man stelle sich zum Beispiel die **Produktkataloge** von Amazon vor, die Menge der darin enthaltenen Daten und die **Anzahl der Zugriffe** voraus der ganzen Welt.

1. Indizierung und Anfrageoptimierung
2. Mehrbenutzerbetrieb / Sperrkonzepte
3. Datenintegrität
4. Transaktionsverwaltung
5. Datensicherheit
6. Datenschutz

2.4.1 Effizienz [Indizierung und Anfrageoptimierung]

Indizierung:

- Verzeichnis, in dem das DBMS nachschlägt, an welcher **Position** im Daten-speicher ein Datensatz steht, von dem nur ein bestimmter Wert bekannt ist
- oft als Baum aufgebaut

- Vorteil: Schneller Zugriff auf den gesuchten Datensatz
- Nachteile: Speicherplatz, Zeit für Erstellung und Aktualisierung

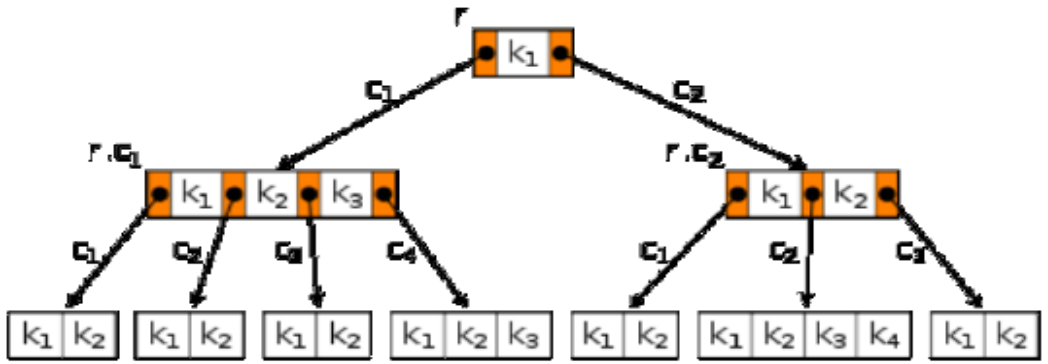


Abbildung 2-3 Beispiel für einen Index

Automatische Anfrageoptimierung:

DBMS analysiert Anfragen und formt sie so um, dass der Zugriff auf die Daten möglichst schnell von statten geht.

2.4.2 Mehrbenutzerbetrieb

MHP
A PORSCHE COMPANY

- mehrere Benutzer oder Anwendungen greifen **gleichzeitig** auf die Daten zu

- **Probleme**:
  - gleichzeitige Änderungen von Daten
  - Gleichzeitige Änderung und Anzeige von Daten
  - Gleichzeitige Anzeige und Löschung von Daten

  - … weitere **gleichzeitige Ausführung von CRUD Operationen**

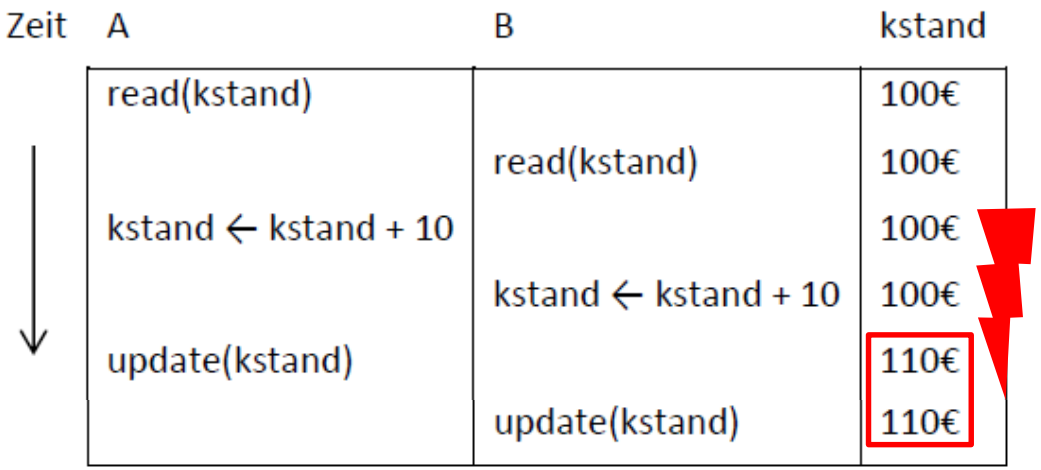
- **Lösung**:
  - Unterschiedliche Sperrkonzepte

2.4.2 Mehrbenutzerbetrieb

- Beispiel 2.2 **Lost Update** durch **konkurrierenden Zugriff mehrerer Benutzer**

  Beim Lost Update verändern zwei Benutzer fast gleichzeitig denselben Datenwert. Das Beispiel zeigt, wie dabei die Änderung eines der Benutzer verloren gehen kann.

  Im Beispiel dient eine Datenbank dazu, Kontostände zu speichern. Auf dem Konto 511 liegen 100€. Der Benutzer möchte 10€ auf das Konto 511 einzahlen, der Benutzer B ebenfalls.



2.4.3 Datenintegrität - Fehler und Inkonsistenzen von vornherein verhindern

MHP
A PORSCHE COMPANY

- Beispiel 2.3 Integritätsbedingungen (*integrity constraint)* in einer Hochschuldatenbank

  I1: Jeder Student muss **eine Matrikelnummer** haben

  I2: Jede Matrikelnummer darf **nur einmal vergeben** werden

  I3: Jeder Student **muss in einem Studiengang** eingeschrieben sein.

  I4: Als Studiengang eines Studenten darf nur ein in der Datenbank **existierender Studiengang** eingegeben werden.

  I5: Der Datensatz eines Studiengangs **darf nicht** aus der Datenbank **gelöscht** werden, solange noch **Studierende im Studiengang eingeschrieben** sind.

Datenbankmanagementsysteme verfügen über Mechanismen und gewährleisten so in gewissen Grenzen Datenintegrität.

Ziel: Vorkehrungen treffen, um Fehler und Inkonsistenzen von **vornherein** zu verhindern!

2.4.4 Transaktionsverwaltung

Transaktion:

Eine Transaktion ist eine **Folge von DB-Operationen**, die als **atomare (unteilbare) Einheit** betrachtet wird.

- zentrales Konzept zur Steuerung des Mehrbenutzerbetriebs
- Erhalt der Datenintegrität
- Transaktion besteht aus mehreren elementaren DB-Operationen
- Elementare DB-Operationen sind die bekannten CRUD-Operationen

- **DBMS** verhält sich so, als wäre die **Transaktion eine unteilbare Einheit**
- **Bei Fehler in einzelnen sequenzielle ausgeführten Operationen** stellt das DBMS den Ausgangszustand wieder her (**Rollback**)
- Persistierung durch DBMS erfolgt nur nach kompletter fehlerfreien Ausführung der Transaktion (***Commit***)

2.4.4 Transaktionsverwaltung – ACID Prinzip

Transaktion:

Eine Transaktion ist eine **Folge von DB-Operationen**, die als **atomare (unteilbare) Einheit** betrachtet wird.

- **ACID-Prinzip**:
  - **A**tomarität: vollständig oder gar nicht
  - **C**onsistenz: Integritätsbedingungen eingehalten
  - **I**solation: jede Transaktion ist unabhängig von anderen Transaktionen
  - **D**auerhaftigkeit: Eine abgeschlossene Transaktion überdauert nachfolgende Fehler

2.4.5 Datensicherheit

MHP
A PORSCHE COMPANY

- DBMS bieten Mechanismen, um Daten vor Verlust zu sichern
- Beispiele:
- Backups: Erstellen von Datensicherungskopien
- Recovery: Wiederherstellen nach Störungen
- Inkrementelle Datensicherung:
  - jede Änderung auf den Daten wird protokolliert
  - Änderungen auf der jeweils letzten Sicherungskopie wiederherstellen
- Replikation:
  - ständig 2 oder mehrere Datenbankmanagementsysteme auf **separater Hardware parallel**
  - Replikation der Operationen des ersten, produktiven Systems auf zweites
  - Ersatzsystem
  - Szenarios bezeichnet man auch als **Replikations-Cluster**, das Prinzip heißt **Failover**

2.4.6 Datenschutz

- Datenschutz ist der Schutz der Daten vor unberechtigten Zugriffen.
- Datenbankmanagementsysteme unterstützen durch Mechanismen
- Definition von Benutzer und Benutzergruppen

- Authentifizierung
  - Anmeldung am Datenbankmanagementsystem

- Autorisierung
  - individuelle Berechtigungen
  - Leserechte oder Schreibrechte auf bestimmten Datenbereichen
  - Rechte Datenbanken oder Strukturen in bestimmten Datenbanken anzulegen oder zu löschen
  - …

2.5 Übungen

- Aufgabe 2.1

  Die Vertriebsmitarbeiter des Unternehmen Hoske sammeln Informationen über Kundenkontakte in einem Tool, das auf Basis des Tabellenkalkulationsprogramms MS Excel realisiert ist. Dieses Tool erfüllt mehrere Funktion:

  Die Mitarbeiter können Informationen zu Kundenkontakten einpflegen und abrufen. Das Tool speichert die Daten dauerhaft. Im Laufe der Zeit sind einige Komfort-Funktionen dazugekommen wie zum Beispiel Auswertungen und Unterstützung bei der Planung von Kundenkontakten.

  Welche der in Abschnitt 2.4 genannten Funktionen eines Datenbankmanagementsystems kann MS Excel in diesem Anwendungsszenario mit wenig Aufwand abdecken?

  a) Welche Funktionen, die ein DBMS bietet, wären in diesem Anwendungsszenario nützlich, aber mit Excel nur mit besonderem Aufwand zu realisieren?

  b) b) Welche Vorteile bietet Excel in diesem Anwendungsszenario gegenüber einem DBMS? (5 Min)

2.5 Übungen

MHP
A PORSCHE COMPANY

Aufgabe 2.2

Ein Unternehmen hat Tausende Kunden weltweit. Täglich kommen viele Anfragen von Kunden per Email und telefonisch und werden im „Kundenkontaktcenter“ bearbeitet. Dort arbeiten bis zu 15 Mitarbeiter gleichzeitig. Auch diese Mitarbeiter erfassen Informationen über die Kundenkontakte in einem CRM-System.

Erläutern Sie, welche Funktionen eines DBMS erforderlich sind, um die Anforderungen dieses Kundenkontaktcenters an das CRM-System abzudecken.

Agenda

MHP
A PORSCHE COMPANY

1. Daten und Datenbanken
2. Datenbankmanagementsysteme
3. **Das relationale Datenmodell**
4. Datenbankentwurf

3.1 Relationen

- Das abstrakte Datenmodell ist die Art, wie Daten betrachtet werden: als Tabellen, als Objekte, also Schlüssel-Wert-Paare und andere mehr.
- derzeit verbreitetste Datenmodellen ist das relationale Datenmodell
- Es wurde in den 1970ger Jahren entwickelt (Codd, 1970), (Date C. , 1975).
- Das relationale Datenmodell repräsentiert Daten als Relationen. Relationen lassen sich darstellen als 2-dimensionale Tabellen

Eine Relation ist eine Menge von Tupeln $(w_1, w_2, ..., w_n) \in D_1 \times D_2 \times ... \times D_n$

Die w_i heißen Werte, die D_i sind Wertebereiche von Attribute.

- Beispiel 3.1 Relation

Student

Matrikelnr	Name	Vorname	Geburtsdatum	Email
1315	Müller	Andi	23.01.1989	mueller8113@yahoo.com
1319	Meyer	Uli	08.08.1988	meyer2221@gmail.de
1321	Braun	Ute	20.05.1990	schlumpf@gmx.de

3.1 Relationen

- Übertragen auf die Darstellung als Tabelle:

  - Die **Attribute** entsprechen den **Spalten**.
  - Die Anzahl der Spalten/Attribute ist fest.
  - In jeder **Zeile** steht ein **Tupel**. Eine Zeile nennt man auch einen **Datensatz**.
  - Die Anzahl der Zeilen ist flexibel.
  - Jedes Tupel/jeder Datensatz kommt **nur einmal in** der **Relation** vor.

- Beispiel 3.1 Relation

Student

MatrikelNr	Name	Vorname	Geburtsdatum	Email
1315	Müller	Andi	23.01.1989	mueller8113@yahoo.com
1319	Meyer	Uli	08.08.1988	meyer2221@gmail.de
1321	Braun	Ute	20.05.1990	schlumpf@gmx.de

3.1 Relationen

- **Relationenschema**: beschreibt den Aufbau einer Relation
- **Stelligkeit** der Relation: die Anzahl ihrer Attribute
- **Datenbankschema**:
  - besteht aus der Menge der Relationenschemata einer Datenbank
  - beschreibt den Aufbau der Datenbank

- Beispiel 3.2 Relation Kunde

Kunde

Kdnr	Suchbegriff	Name	Branche
115	Mayer-Kempten	Mayer GmbH	MV
116	Mueller-Sonthofen	Franz Mueller und Sohn	MB
117	Mueller-Kempten	MPK Mueller Präzision Kempten	MB

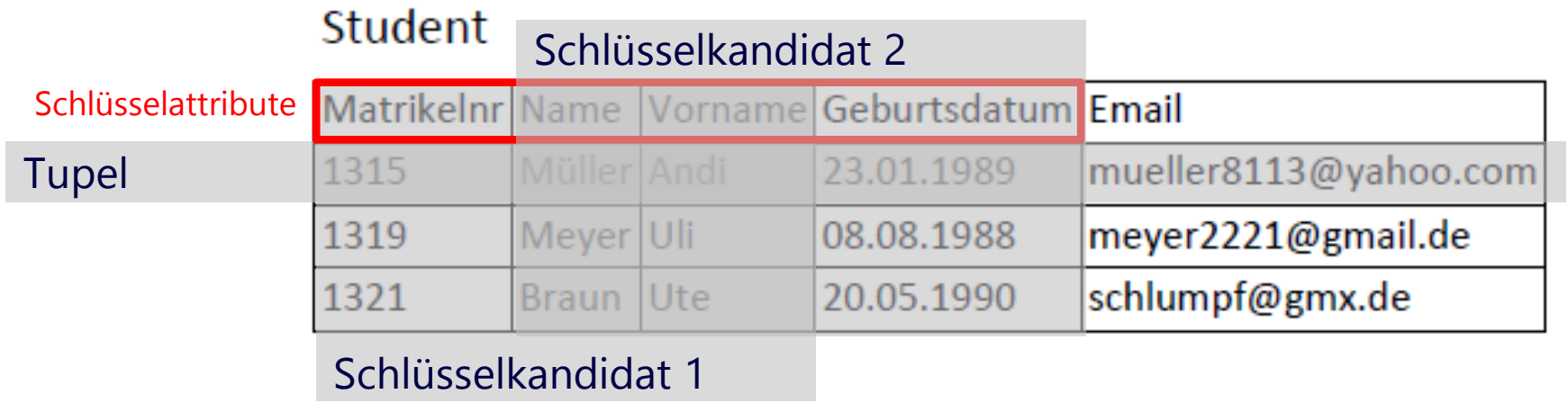
- Relationenschema: **Kunde(Kdnr, Suchbegriff, Name, Branche)**
- Stelligkeit der Relation**: 4**

3.2 Schlüssel

- Ein Schlüssel ist eine **Teilmenge** der Attribute, die jedes **Tupel eindeutig** identifizieren
  - Alle Attribute der Relation bilden zusammen also einen Schlüssel.

- Ein **Schlüsselkandidat** ist ein Schlüssel mit **minimaler** Anzahl Attribute
  - Eine Relation kann mehrere Schlüsselkandidaten haben.

- Ein **Schlüsselattribut** ist ein Attribut, das Teil mindestens eines Schlüsselkandidaten ist.
  - Alle anderen Attribute sind Nicht-Schlüsselattribute.



Matrikelnr	Name	Vorname	Geburtsdatum	Email
1315	Müller	Andi	23.01.1989	mueller8113@yahoo.com
1319	Meyer	Uli	08.08.1988	meyer2221@gmail.de
1321	Braun	Ute	20.05.1990	schlumpf@gmx.de

3.2.1 Primärschlüssel

MHP
A PORSCHE COMPANY

- Ein **Primärschlüssel** (englisch **primary key**) ist ein beliebig ausgewählter Schlüsselkandidat, der zur **eindeutigen Identifizierung** jeder Zeile benutzt wird.

- Ein Primärschlüssel, der aus mehreren Attributen besteht, ist ein **zusammengesetzter Primärschlüssel**

- Primärschlüssel kennzeichnet man in Relationenschemata und Tabellen durch **Unterstreichen der Attributnamen/Spaltenüberschriften**.

Student

MatrikelNr	Name	Vorname	Geburtsdatum	Email
1315	Müller	Andi	23.01.1989	mueller8113@yahoo.com
1319	Meyer	Uli	08.08.1988	meyer2221@gmail.de
1321	Braun	Ute	20.05.1990	schlumpf@gmx.de

3.2.2 Fremdschlüssel und referentielle Integrität

MHP
A PORSCHE COMPANY

- Fremdschlüssel bilden Beziehungen zwischen Daten in **verschiedenen Relationen** ab
- Eine Relation wird bezeichnet als **Elterntabelle**, oder unabhängige Relation, ***Master*-Tabelle**, referenzierte Tabelle.
- Die andere Relation heißt **Kindtabelle**, abhängige Relation, ***Detail*-Tabelle** beziehungsweise **referenzierende Tabelle**
- Ein **Fremdschlüssel** ist ein Attribut in einer Kindtabelle, das sich auf einen Primärschlüssel in der Elterntabelle bezieht
- Ein **Fremdschlüssel** (englisch ***foreign key***) ist ein Attribut in einer Tabelle, das einem Primärschlüssel in einer anderen Tabelle entspricht oder eine Menge von Attributen) in einer Tabelle, die einem zusammengesetzten Primärschlüssel in einer anderen Tabelle entspricht.
- Die **referentielle Integrität** (Integrität der Bezüge) ist eine wichtige **Integritätsbedingung**. Referentielle Integrität ist gegeben, wenn alle Werte, die im Fremdschlüssel in der Kindtabelle vorkommen, auch im Primärschlüssel der Elterntabelle enthalten.

3.2.2 Fremdschlüssel und referentielle Integrität

- Beispiel 3.3 Fremdschlüssel

Student

Matrikelnr	Name	Vorname	Studiengang
1315	Müller	Andi	WI
1319	Meyer	Uli	WI
1321	Braun	Ute	MB

Fremd-schlüssel

Studiengang

Studiengang	Abschluss	Leiter
WI	Bachelor	Prof. Ganz
MB	Master	Prof. Schwer

Primär-schlüssel

- Die **referentielle Integrität ist erfüllt**, denn für alle Studiengänge, in die Studenten laut Kindtabelle eingeschrieben sind, existiert ein entsprechender Eintrag in der Elterntabelle.

3.2.3 Künstliche Schlüssel

- Attribute ein, die an sich keine eigene Information tragen, um sie als Schlüssel zu verwenden nennt man **künstliche Schlüssel**.

Student

Matrikelnr	Name	Vorname	Studiengang
1315	Müller	Andi	WI
1319	Meyer	Uli	WI
1321	Braun	Ute	MB

Künstlicher Schlüssel

Studiengang

Studiengang	Abschluss	Leiter
WI	Bachelor	Prof. Ganz
MB	Master	Prof. Schwer

3.2.3 Künstliche Schlüssel

- Beispiel 3.4 Künstliche Schlüssel

  Beispiele für künstliche Schlüssel sind: Matrikelnummer, Personalnummer, Kundennummer, Auftragsnummer, ISBN, Inventarnummer, Seriennummer.

  Die Kombination (Name, Vorname, Geburtstag, Geburtsort) bildet einen natürlichen Schlüssel für Personen, ist aber unübersichtlich und umständlich einzusetzen.

  Wann und wozu verwendet man künstliche Schlüssel?

- Wenn es keine natürlichen Schlüssel gibt oder wenn sich der natürliche Schlüssel aus sehr vielen Attributen zusammensetzt.

- Weil man den Schlüssel als Verweis in andere Relationen (also als Fremdschlüssel) benutzen möchte.

3.3 Normalformen von Relationalen Datenbanken

MHP
A PORSCHE COMPANY

- Informationen über einen Anwendungsbereich können unterschiedlich in Relationen abgebildet werden

- Normalformen sind „standardisierte“ Arten, die sich als besonders geeignet erwiesen haben
  - Erste Normalform (1NF)
  - Zweite Normalform (2NF)
  - Dritte Normalform (3NF)

3.4.2 Beispiel zum Normalisierungsprozess

- Beispiel 3.7 Normalisierung einer Datenbank mit Prüfungsanmeldungen

In diesem Beispiel wird eine Datenbank bis zur dritten Normalform normalisiert. Anfangs besteht diese Datenbank aus nur einer Tabelle. Diese dient dazu, Anmeldungen von Studierenden zu Prüfungen im laufenden Semester zu verwalten.

Prüfungsanmeldung

Name	Vorname	MatrikelNr	Studiengang	Abschluss	Leiter	Prüfung
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	PM, Physik, BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	BWL, PM
Braun	Ute	1321	MB	Master	Prof. Schwer	PM, Mathe

3.4.2.1 Ausgangsdatenbank

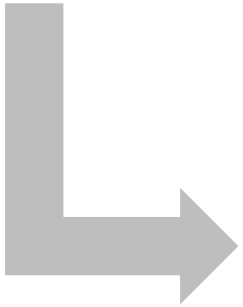
Prüfungsanmeldung

Name	Vorname	MatrikelNr	Studiengang	Abschluss	Leiter	Prüfung
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	PM, Physik, BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	BWL, PM
Braun	Ute	1321	MB	Master	Prof. Schwer	PM, Mathe

3.4.2.2 Datenbank in 1NF

Prüfungsanmeldung

Name	Vorname	Matrikelnr	Studiengang	Abschluss	Leiter	Prüfung
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	PM, Physik, BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	BWL, PM
Braun	Ute	1321	MB	Master	Prof. Schwer	PM, Mathe



Prüfungsanmeldung

Name	Vorname	Matrikelnr	Studiengang	Abschluss	Leiter	Prüfung
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	PM
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	Physik
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	PM
Braun	Ute	1321	MB	Master	Prof. Schwer	PM
Braun	Ute	1321	MB	Master	Prof. Schwer	Mathe

3.4.2.3 Datenbank in 2NF

Prüfungsanmeldung

Matrikelnr	Prüfung
1315	PM
1315	Physik
1315	BWL
1319	BWL
1319	PM
1321	PM
1321	Mathe

Prüfungsanmeldung

Name	Vorname	Matrikelnr	Studiengang	Abschluss	Leiter	Prüfung
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	PM
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	Physik
Müller	Andi	1315	WI	Bachelor	Prof. Ganz	BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	BWL
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz	PM
Braun	Ute	1321	MB	Master	Prof. Schwer	PM
Braun	Ute	1321	MB	Master	Prof. Schwer	Mathe

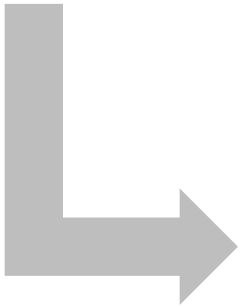
Student

Name	Vorname	Matrikelnr	Studiengang	Abschluss	Leiter
Müller	Andi	1315	WI	Bachelor	Prof. Ganz
Meyer	Uli	1319	WI	Bachelor	Prof. Ganz
Braun	Ute	1321	MB	Master	Prof. Schwer

3.4.2.4 Datenbank in 3NF

Prüfungsanmeldung

Matrikelnr	Prüfung
1315	PM
1315	Physik
1315	BWL
1319	BWL
1319	PM
1321	PM
1321	Mathe



Prüfungsanmeldung

Matrikelnr	Prüfung
1315	PM
1315	Physik
1315	BWL
1319	BWL
1319	PM
1321	PM
1321	Mathe

Studiengang

Studiengang	Abschluss	Leiter	
WI	Bachelor	Prof. Ganz	
MB	Master	Prof. Schwer	

Student

Name	Vorname	Matrikelnr	Studiengang
Müller	Andi	1315	WI
Meyer	Uli	1319	WI
Braun	Ute	1321	MB

3.5 Übungen

- Eine relationale Datenbank enthält Daten zu Schulungen. Abbildung 3-1 zeigt einen Ausschnitt.

- Aufgabe 3.1 Relationale Datenbanken – Begriffe

  a) Welche Relationenschemata gehören zu den gezeigten Relationen?
  b) Was ist die Stelligkeit der Relation Schulung?
  c) Was ist die Stelligkeit der Relation Dozent?
  d) Markieren Sie in den Relationenschemata geeignete Primärschlüssel für die Relationen durch Unterstreichen.
  e) Markieren Sie Fremdschlüssel in den Relationenschemata mit einem Punkt •.
  f) Sind alle Relationen in 3NF? Wenn nein, welche Relation verletzt welche Normalform?

Schulung

Nr	Titel	Dauer
CA1	CAM basic	1 Tag
C22	CAM expert	3 Tage
MT1	Maintenance Expert	2 Tage

Expertise

SchulungNr	DozentNr	Bewertung	Honorar
CA1	1	4	100
CA1	2	4	100
C22	1	4	150
C22	2	4	150
MT1	3	3	100

Dozent

Nr	Name	Titel	Vorname
1	Schneider	Dipl.-Ing.	Franz
2	Sulian	B. Eng.	Sandra
3	Schlieder	M. Eng.	Steffan

Location

Nr	Name	Ort	Adresse
4	Regis Business Center	München	Messe 4
5	Seminaris Hotel	Stuttgart	Bärenstr 12

Termin

Nr	Terminnr	Termin	Ort	Dozent	Location
CA1	41	23.11.2015	München	2	4
CA1	42	23.11.2015	Stuttgart	1	5
C22	30	24.11.2015	Stuttgart	3	5

Abbildung 3-1 Datenbank der Miniwelt Schulungen (Ausschnitt)

3.5 Übungen

Schulung

Nr	Titel	Dauer
CA1	CAM basic	1 Tag
C22	CAM expert	3 Tage
MT1	Maintenance Expert	2 Tage

Expertise

SchulungNr	DozentNr	Bewertung	Honorar
CA1	1	4	100
CA1	2	4	100
C22	1	4	150
C22	2	4	150
MT1	3	3	100

Dozent

Nr	Name	Titel	Vorname
1	Schneider	Dipl.-Ing.	Franz
2	Sulian	B. Eng.	Sandra
3	Schlieder	M. Eng.	Steffan

Location

Nr	Name	Ort	Adresse
4	Regis Business Center	München	Messe 4
5	Seminaris Hotel	Stuttgart	Bärenstr 12

Termin

Nr	Terminnr	Termin	Ort	Dozent	Location
CA1	41	23.11.2015	München	2	4
CA1	42	23.11.2015	Stuttgart	1	5
C22	30	24.11.2015	Stuttgart	3	5

Abbildung 3-1 Datenbank der Miniwelt Schulungen (Ausschnitt)

3.5 Übungen

- Aufgabe 3.2 Autohaus-Daten normalisieren

Ein kleines Autohaus, das seine Daten bisher in Excel erfasst und bearbeitet hat, wird langsam groß und möchte seine Daten jetzt in einer richtigen Datenbank verwalten. Abbildung 3-2 zeigt einen Ausschnitt aus der vorhandenen Excel-Datei.

a) Bringen Sie diesen in 1NF.

b) Bringen Sie diesen in 2NF.

c) Bringen Sie diesen in 3NF.

Name	Adresse	Marke	Typ	Seriennr	Farbe	Verkäufer	VDatum	Eintritts datum
Meier	Platanenweg 7	VW Opel	Golf Astra	123456 345678	blaumetallic silber	Schmid Plüss	23.04.2014 07.08.2015	01.06.2006 15.10.2007
Müller	Altstadt 12	VW	Golf	55567	silber	Frey	12.04.2015	01.06.2006
Steffen	Gartenstr. 2	VW	Bora	3232323	weiss	Schmid	15.07.2015	01.06.2006
Steffen	Gartenstr. 2	Audi	A6	55454545	schwarz	Frey	13.11.2014	01.06.2006

Abbildung 3-2 Tabelle mit Autohaus-Daten vor der Normalisierung

3.5 Übungen

- Aufgabe 3.3 Lieferungsdaten normalisieren

Im Lager der Hoske müssen die eingetroffenen Lieferungen manuell in einem Excel-Arbeitsblatt erfasst werden. Bringen Sie die Daten in 3NF und markieren Sie die Primärschlüssel.

Legende: AgNr: Artikelgruppennummer LiefNr: Nummer der Lieferung

Lieferungen

LiefNr	Lieferant	Land	22.2.2010	Artikel	Bezeichnung	AgNr	Artikel gruppe	Menge	Einheit
15500	Simmler	CH	22.2.2010	123	SchraubeM4	3	Schrauben	1000	Stk
15500	Simmler	CH	15.3.2010	678	SchraubeM8	3	Schrauben	1300	Stk
15300	Simmler	CH	15.3.2010	123	SchraubeM4	3	Schrauben	1000	Stk
15300	Simmler	CH	22.2.2010	678	SchraubeM8	3	Schrauben	1300	Stk
15060	Schauer	D	22.2.2010	123	SchraubeM4	3	Schrauben	4000	Stk
15060	Schauer	D	23.2.2010	690	SchraubeM3	3	Schrauben	2000	Stk
15570	Leclerq	F	23.2.2010	232	Schlauch12	51	Schläuche	10	m
15600	Miller	D	23.2.2010	500	Schlauch14	51	Schläuche	10	m
15600	Miller	D	22.2.2010	232	Schlauch12	51	Schläuche	5	m

Abbildung 3-3 Lieferungsdaten vor der Normalisierung

3.5 Übungen

- Aufgabe 3.4 Daten zum Lagerbestand normalisieren

Auch die Lagerbestände der Hoske werden manuell in einem Excel-Arbeitsblatt erfasst. Bringen Sie die Daten in 3NF und markieren Sie die Primärschlüssel.

Legende: AgNr: Artikelgruppennummer PM: Produktmanager, zuständig für Artikelgruppen

Lagerbestand

Arti-kel	Bezeichnung	Ab-mes-sung	Artikel-gruppe	PM	Einheit	Be-stand	Platz	Wert	Zugang
212	Schraube Me	4x12	Schrauben	Simmler	Stk	150	A-17	1,30	02.10.14
214	Schraube Me	4x14	Schrauben	Simmler	Stk	100	A-18	1,00	03.10.14
214	Schraube Me	4x14	Schrauben	Simmler	Stk	50	A-17	1,00	04.10.14
212	Schraube Me	4x12	Schrauben	Simmler	Stk	200	B-01	1,30	05.10. 14
216	Schraube Me	4x16	Schrauben	Simmler	Stk	80	A-21	1,50	06.10. 14
R15	Rohr Kupfer	15	Rohre	Sepp	m	10	R-11	1,00	07.10. 14
R18	Rohr Kupfer	18	Rohre	Sepp	m	20	R-12	2,00	08.10. 14

Agenda

MHP
A PORSCHE COMPANY

1. Daten und Datenbanken
2. Datenbankmanagementsysteme
3. Das relationale Datenmodell
4. **Datenbankentwurf**

4.1 Phasen des Datenbankentwurfs

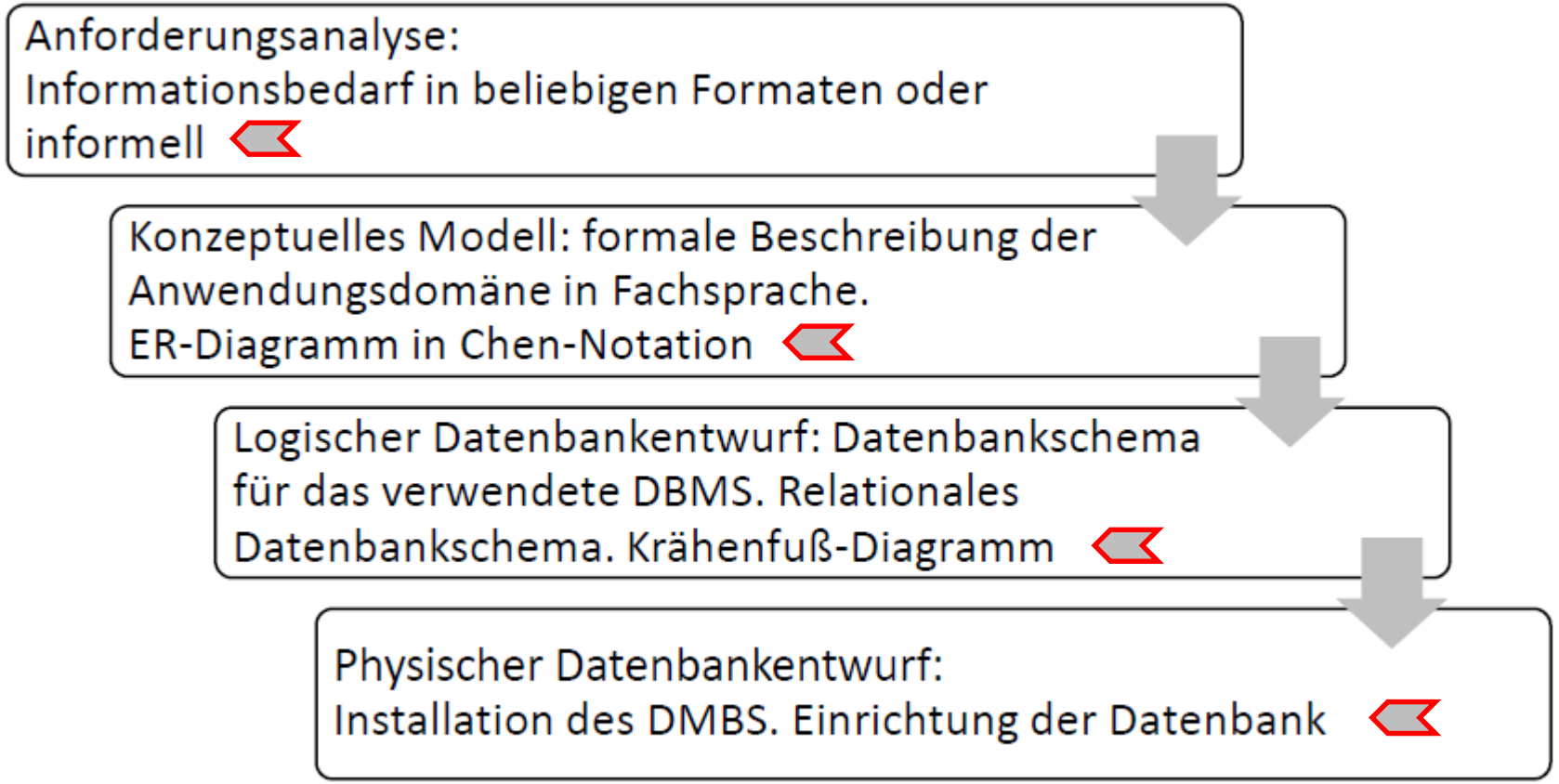
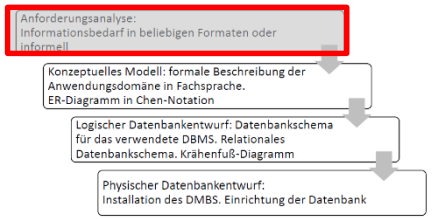


Abbildung 4-1 Überblick über die Phasen des Datenbankentwurfs

4.1.1 Anforderungsanalyse

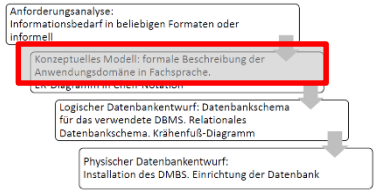


- Datenbanksystem die Rolle eines Daten-Dienstleisters für ein oder mehrere Anwendungsprogramme
- Anwendungsprogramme führen Operationen auf diesen Daten aus und realisieren die eigentliche Funktionalität des Gesamtsystem

  → Anforderungen an Anwendungsprogramme bestimmen welche Informationen die Datenbank speichern und bereitstellen muss

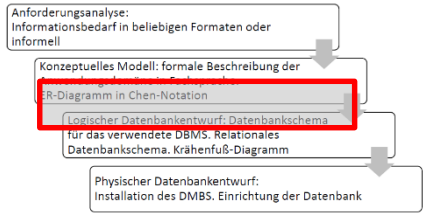
- Die Anforderungen an die Anwendung können in verschiedenen, möglicherweise **informellen** Formaten vorliegen

4.1.2 Konzeptuelles Modell (Miniwelt, Anwendungsdomäne)



- beschreibt die **Anwendungsdomäne** oder **Miniwelt der Datenbank** auf der **Fachebene**
- Nutzung der **Fachsprache** der zukünftigen Fachanwender und Benutzer, jedoch als **formalisierte** Darstellung
- Ziele:
  - für Fachanwender verständlich
  - Diskussionsgrundlage
  - von technischen Details abstrahiert
  - formalisiert, so dass es eindeutig interpretierbar ist
- Beispielnotation:
  - **Entity-Relationship-Diagramme** in der Notation von **Chen**

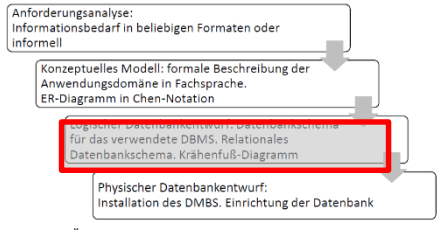
4.1.3 Logischer Datenbankentwurf



- konzeptuelles Modell wir übersetzt in das Datenmodell eines **konkreten** DMBS
  - Beispiel: Übersetzung in ein relationales Modell

- **Eigenschaften**:
  - Detailliert
  - Nahe an der technischen Umsetzung
  - Enthält Schlüssel-Fremdschlüsselbeziehungen, Integritätsbedingungen, Datentypen (physische Umsetzung der Datenbank)

4.1.4 Physischer Datenbankentwurf



- Der physische Entwurf hängt auch ab vom eingesetzten Datenbankmanagementsystem, da die Produkte verschiedener Hersteller hierfür unterschiedliche Optionen bieten.

- **Eigenschaften**:
  - legt technische Details fest
  - Definiert Speicherstrukturen
  - Definiert und richtet Indizes (wie in Abschnitt 2.4.1) ein
  - Auswahl von optimalen Suchalgorithmen

- Für komplexe Anwendungen mit großen Datenmengen braucht es Experten, um die Datenbank auf der Ebene des physischen Entwurfs optimal einzurichten.

- Für kleinere Anwendungen genügen Standardeinstellungen der üblichen DMBS völlig, um ein ausreichend effizientes Datenbanksystem einzurichten.

4.2.1 Entity-Relationship-Modellierung (ERM)

MHP
A PORSCHE COMPANY

Ein Entity-Relationship-Modell beschreibt, wie ein Ausschnitt aus der realen Welt in einer Datenbank abgebildet werden kann.

- Methode zur Modellierung von Daten
- Entwurf von relationalen Datenbanken
- Analyse und Strukturierung der Informationen aus dem Anwendungsbereich
- Fachwissen der zukünftigen Benutzer und Anwender der Datenbank wichtig

- **ER Diagramme**
  - Anschaulich, leicht erweiterbar, abstraktes Niveau
  - für Nicht-Datenbankspezialisten verständlich
  - Objekte und Strukturen der abzubildenden Anwendungsdomäne
  - Arbeitsgrundlage in Team-Diskussionen

- **Erweiterungen und alternative Notationen (Schreibweisen)**
  1. **Chen Notation:** Methode für den ersten manuellen Entwurf eines konzeptuellen Datenbankmodells
  2. Notation nach Barker / **Krähenfuss Notation**: detaillierte Modelle als mit der Chen Notation

4.2.2 Elemente der ER-Modellierung

MHP
A PORSCHE COMPANY

- **Entities**: Gegenstände, Dinge, auch Personen in der betrachteten Miniwelt
- **Relationships**: Beziehungen, die zwischen Paaren von Entities bestehen
- **Attribute**: relevante Eigenschaften der Entities oder der Relationships

- Deutsche Begriffe:
  - Entity → Entität
  - Relationship → Beziehung

4.2.2.1 Entities, Attribute und Entity-Typen

MHP
A PORSCHE COMPANY

- **Entities**:
  - Gegenstände, Dinge, auch Personen in der betrachteten Miniwelt
  - Objekt der realen Welt
  - eigene Identität
  - eindeutig identifizierbar
  - weisen Attribute auf, die einen bestimmten Wert annehmen können

- **Entity-Typ:**
  - Gleichartige Entities
  - dieselben Attribute

- **Identifizierende Attribute**
  - Identifizieren Entitäten eindeutig
  - Entsprechen den Schlüsselattributen in relationalen Datenbanken
  - Beispiel: ISBN-Nummer eines Buches, Inventarnummer eines Bürodruckers

4.2.2.1 Entities, Attribute und Entity-Typen

MHP
A PORSCHE COMPANY

- Beispiel 4.1 Entity und Entity-Typ

  Ein einzelnes von 10.000 Blatt Druckerpapier ist **kein Entity**. Ein Bürodrucker, der im Netzwerk **identifizierbar** ist oder eine **Inventarnummer** besitzt, ist ein **Entity.**

  Alle Bürodrucker sind vom selben Entity-Typ Drucker. Entities des **Entity-Typs Drucker** weisen die **Attribute** Inventarnummer, Art, Anschaffungsdatum und Netzwerkadresse auf.

  Ein Entity dieses Entity-Typs hat dann zum Beispiel im Attribut Inventarnummer den Wert „123321“ und im Attribut Art den Wert „Farblaser“.

  Im allgemeinen Sprachgebrauch wird oft nicht sauber zwischen Entity und Entity-Typ getrennt, so dass mit „Entity“ manchmal genaugenommen „Entity-Typ“ gemeint ist.

4.2.2.1 Entities, Attribute und Entity-Typen

Ein „**schwaches**“ Entity ist ein Entity, das nur in Abhängigkeit von einem anderen Entity existieren kann.

- Beispiel 4.2 Starke und schwache Entities in der Miniwelt „Schulungen“

  Eine **Schulung** ist ein **starkes Entity**. Sie besitzt Attribute wie Schulungsnummer (als identifizierendes Attribut), Titel, Beschreibung, Umfang, usw.

  Zum Beispiel die Schulung „CAM basic“: Diese Schulung findet mehrmals jährlich statt.

  Jeder **Schulungstermin** wird separat geplant. Er hat als Attribute TerminNr, Datum, usw.

  Ein Schulungstermin ist ein **schwaches Entity**, weil er nur in Abhängigkeit zum Entity Schulung „CAM basic“ existieren kann: **ohne Schulung kann es auch keinen Schulungstermin geben**.

4.2.2.2 Relationships und Kardinalitäten von Relationships

MHP
A PORSCHE COMPANY

- Eine Relationship ist eine Beziehung zwischen Entities.
- Die Kardinalität einer Relationship gibt an, wie viele Entities auf jeder Seite maximal an der Relationship beteiligt sind.
- Relationships können Attribute aufweisen.

4.2.2.2 Relationships und Kardinalitäten von Relationships

- Beispiel 4.3 Relationships und Kardinalitäten von Relationships

  „**ist_zugelassen_auf**“ ist eine Relationship zwischen einem Auto einer Person. Diese Relationship hat das **Attribut** „seit“.

  „**fährt**“ ist eine Relationship zwischen einem Auto und einer Person.

  „**gehört_zu**“ ist eine Relationship zwischen einem Auto und einer Zulassungs- bescheinigung.

  1:n-Relationship: ein Auto ist zugelassen auf maximal eine Person. Auf eine Person können mehrere Autos zugelassen sein.

  n:m-Relationship: jede Person kann mehrere Autos fahren. Jedes Auto kann von mehreren Personen gefahren werden.

  1:1-Relationship: zu jedem Auto kann es maximal eine Zulassungsbescheinigung geben. Jede Zulassungsbescheinigung gehört zu maximal einem Auto.

4.3 Entity-Relationship-Modellierung mit Chen-Notation

MHP
A PORSCHE COMPANY

- ER-Modelle sind Diagramme. Die von Peter Chen vorgeschlagene Notation für ER-Diagramme verwendet die Symbole Rechteck für Entities, Raute für Relationships und Ovale für Attribute (siehe Abbildung 4-2). Durchgezogenen Verbindungslinien verbinden die Elemente eines ER-Diagramms. Die identifizierenden Attribute kann man unterstreichen. Abbildung 4-3 zeigt ein Beispiel.

- Kardinalitäten von Relationships werden an den Verbindungslinien angeschrieben wie in Abbildung 4-4.

4.3.1 Entity-Relationship-Modellierung mit Chen-Notation

- ER-Modelle sind Diagramme

- **Peter Chen Notation für ER-Diagramme:**
  - Rechteck für Entities
  - Raute für Relationships
  - Ovale für Attribute
  - Durchgezogenen Verbindungslinien verbinden die Elemente eines ER-Diagramms.
  - Die identifizierenden Attribute kann man unterstreichen. (Abbildung 4-3 zeigt ein Beispiel)

- **Kardinalitäten** von Relationships werden an den Verbindungslinien angeschrieben wie in Abbildung 4-4.

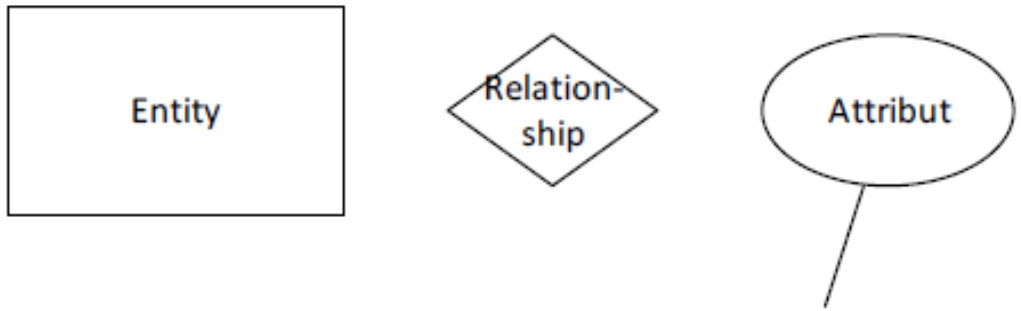


Abbildung 4-2 Symbole der Entity-Relationship-Modellierung

4.3.1 Chen Notation – Beispiel

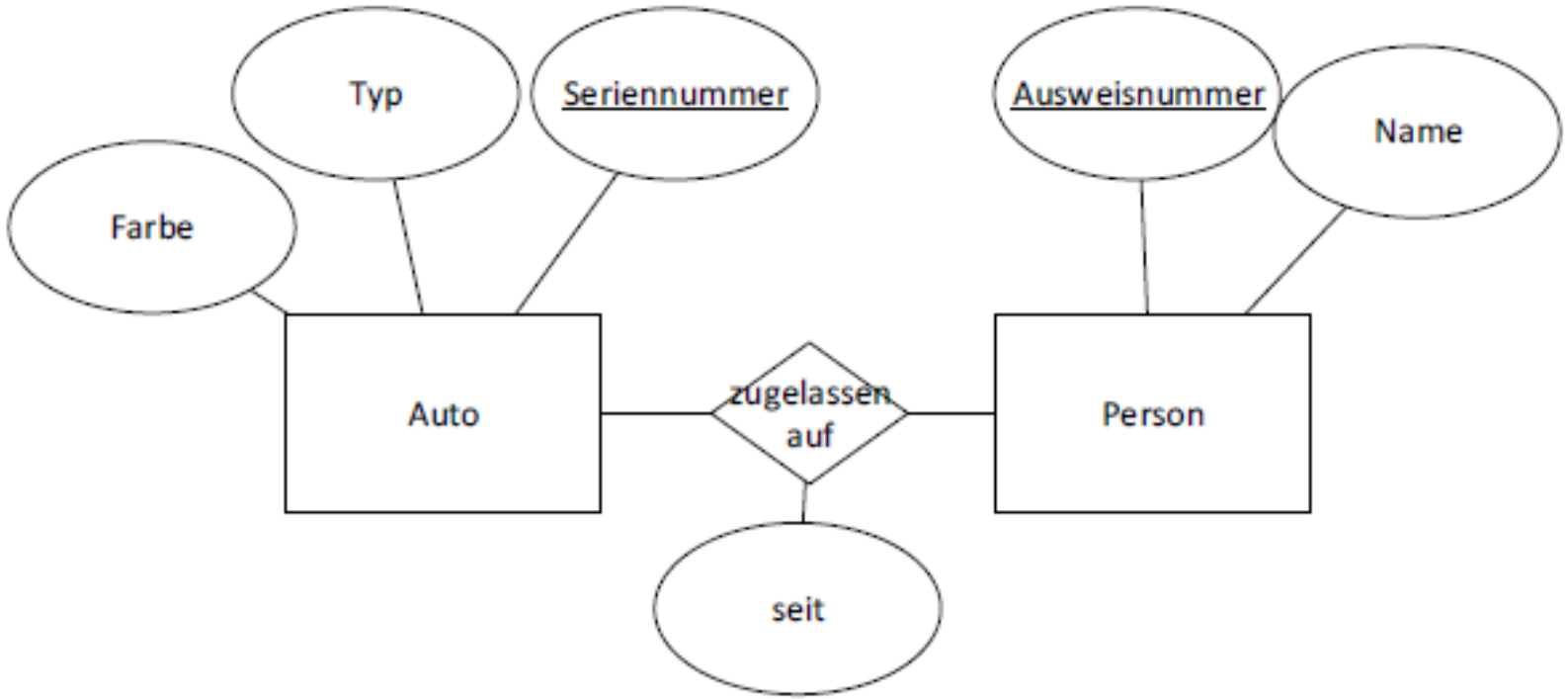


Abbildung 4-3 Beispiel Entity-Relationship-Diagramm

4.3.1 Chen Notation - Kardinalitäten

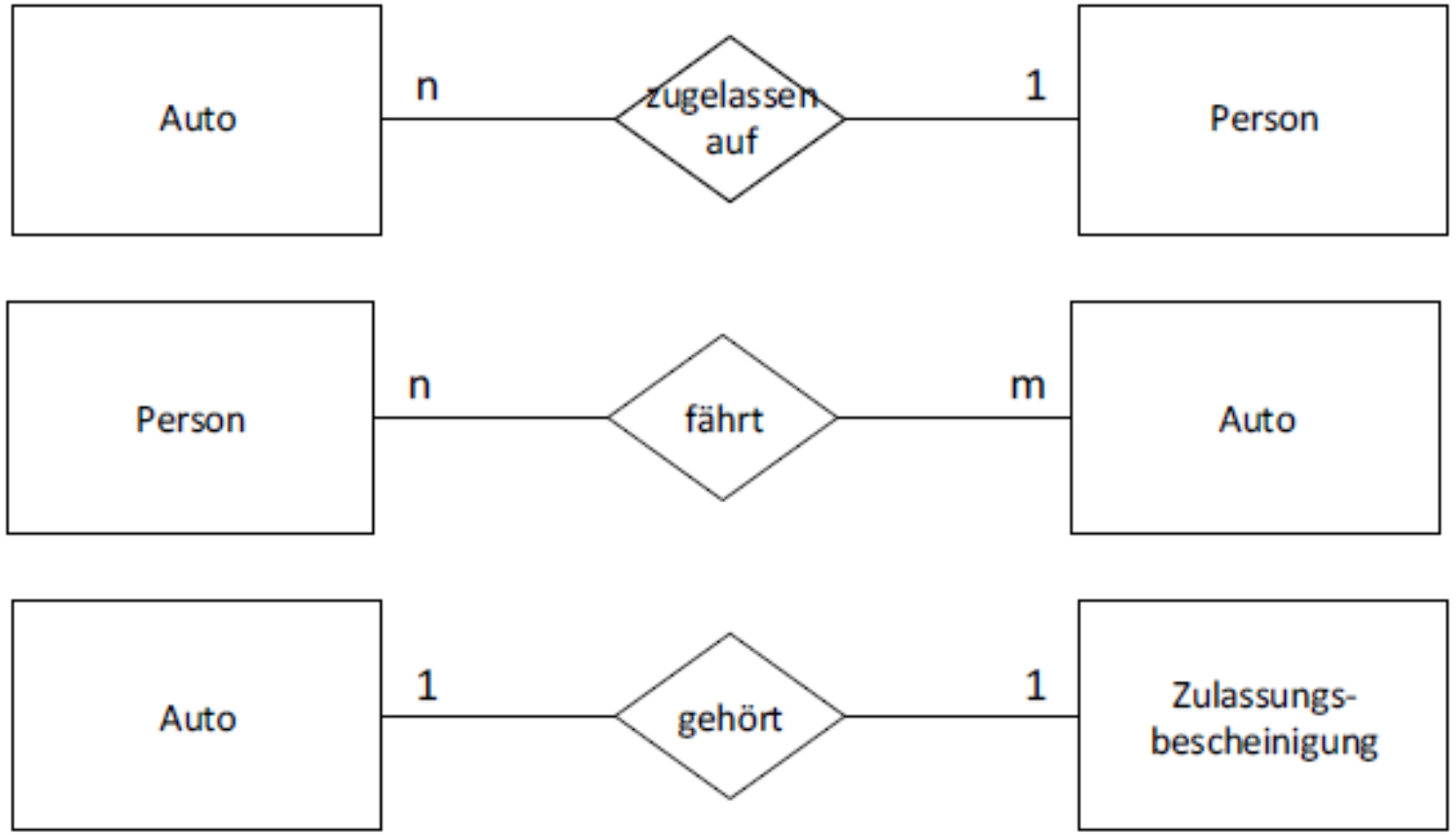


Abbildung 4-4 Kardinalitäten von Relationships

4.3.2 Beispiel

- Beispiel 4.4 ER-Modellierung Hochschule

  Anmerkung: Das Beispiel ist vereinfacht, in einer echten Anwendung wäre die Datenbank wesentlich komplexer!
  Eine Hochschule beginnt, eine Datenbank aufzubauen.
  Fahrplan:

  1. Welche Entities gibt es?
  2. Welche Attribute haben sie? Welche sind identifizierende Attribute?
  3. In welchen Beziehungen (Relationship) stehen sie zueinander?
  4. Welche Attribute haben die Relationships?
  5. Welche Kardinalitäten haben die Relationships?

4.3.2 Beispiel

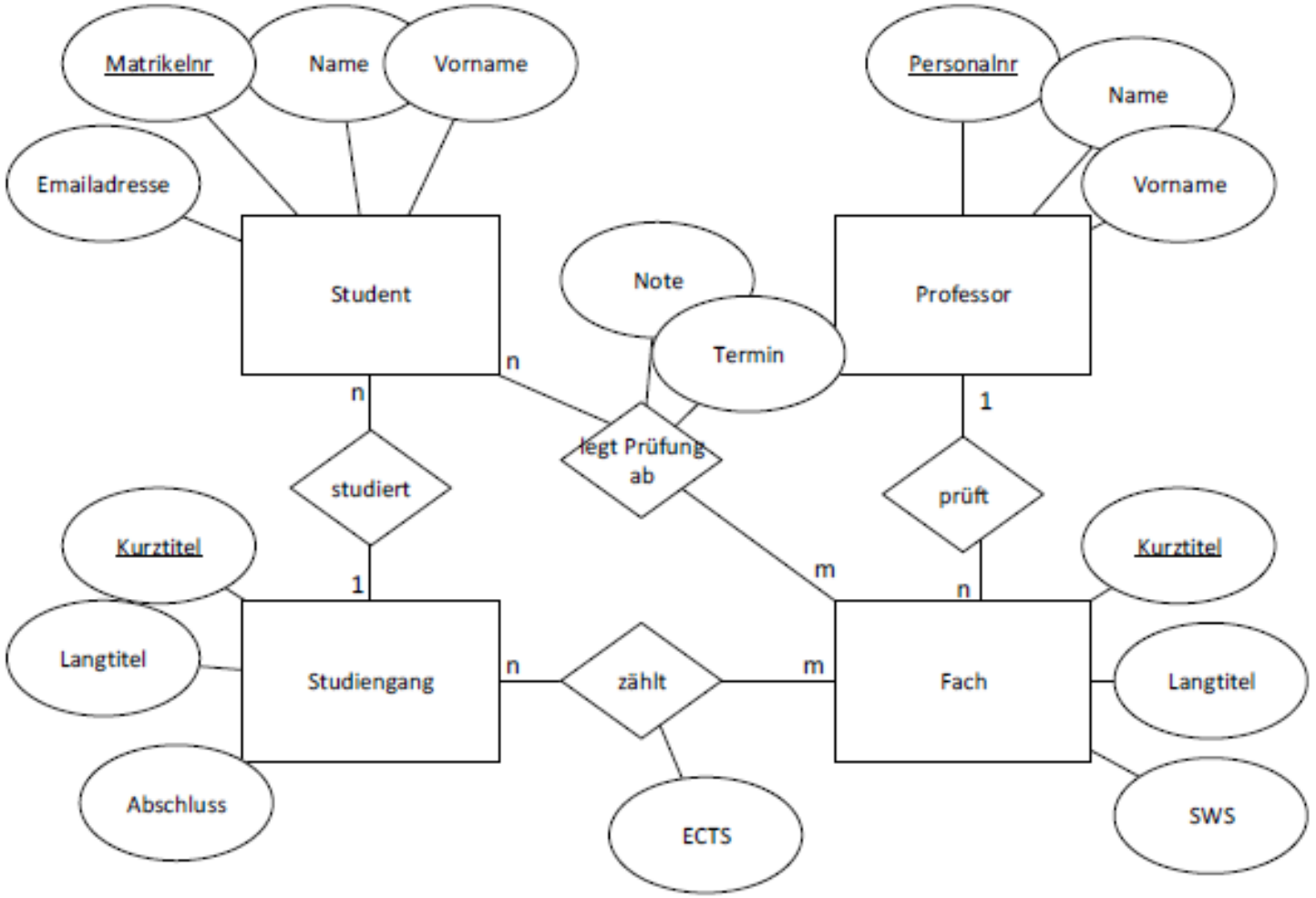


Abbildung 4-5 ER-Diagramm Hochschule in Chen-Notation

4.3.3 Vom Entity Relationship-Diagramm zum relationalen Datenbankschema

MHP
A PORSCHE COMPANY

Ein **Entity-Relationship-Diagramm** kann **eindeutig** in ein **relationales Datenbankschema überführt** werden.

1. **Entities**:
   a) für jedes Entity eine Tabelle erzeugen
   b) jedes Attribut des Entity wird zu einer Spalte der Tabelle
   c) identifizierende Attribute des Entities werden zu Primärschlüsselattributen der Tabelle

2. **1:n-Relationships:**
   a) die identifizierenden beziehungsweise Schlüsselattribute der 1-Seite als Fremdschlüssel in die Tabelle der n-Seite aufnehmen
   b) Wenn die Relationship eigene Attribute hat: Attribute der Relationship als Attribute in die Tabelle der n-Seite aufnehmen

3. **n:m- Relationships**:
   a) für jede n:m-Relationship eine neue Tabelle erzeugen
   b) identifizierende Attribute beider Seiten als Fremdschlüssel in die neue Tabelle aufnehmen
   c) daraus den Primärschlüssel der neuen Tabelle bilden
   d) Wenn die Relationship eigene Attribute hat: Attribute der Relationship als Attribute in die neue Relation aufnehmen

4. Prüfen, ob das Datenbankschema die 3. Normalform erfüllt, und falls nötig **normalisieren**.

4.3.3 Vom Entity Relationship-Diagramm zum relationalen Datenbankschema

MHP
A PORSCHE COMPANY

- Beispiel 4.5 Relationales Datenbankschema zum ER-Modell Hochschule

Student

Matrikelnr	Name	Vorname	Emailadresse	Studiengang_Kurztitel
1315	Müller	Andi	mueller8113@yahoo.com	WI
1319	Meyer	Uli	meyer2221@gmail.de	WI
1321	Braun	Ute	schlumpf@gmx.de	MMB

Professor

Personalnr	Name	Vorname
23	Ganz	Giulia
24	Schwer	Sabine

Studiengang

Kurztitel	Langtitel	Abschluss	Professor_Personalnr
WI	Wirtschaftsingenieurwesen MB	Bachelor	23
MMB	Maschinenbau	Master	24

Fach

Kurztitel	Langtitel	SWS	Professor_Personalnr
PM	Projektmanagement	4	23
Phys	Physik	4	24
BWL	Betriebswirtschaftslehre	4	23
Mathe	Mathematik für Ingenieure	5	24

ECTS

Fach_Kurztitel	Studiengang_Kurztitel	ECTS
PM	WI	5
PM	MBB	4
BWL	WI	6
BWL	MBB	4
Mathe	WI	5
Phys	WI	5

Prüfungsergebnis

Matrikelnr	Fach_Kurztitel	Note	Termin
1315	PM	1	04.02.2015
1315	Phys	1	12.01.2015
1315	BWL	2	13.01.2015
1319	BWL	3	13.01.2015
1319	PM	2	01.02.2014
1321	PM	1	04.02.2015
1321	Mathe	1	11.01.2013

Abbildung 4-6 Hochschuldatenbank zum ER-Diagramm

4.4 Entity-Relationship-Modellierung in Krähenfuß-Notation

- Alternative zur Chen-Notation

- **Primärschlüsselattribute** mit PS
- **Fremdschlüsselattribute** mit FS

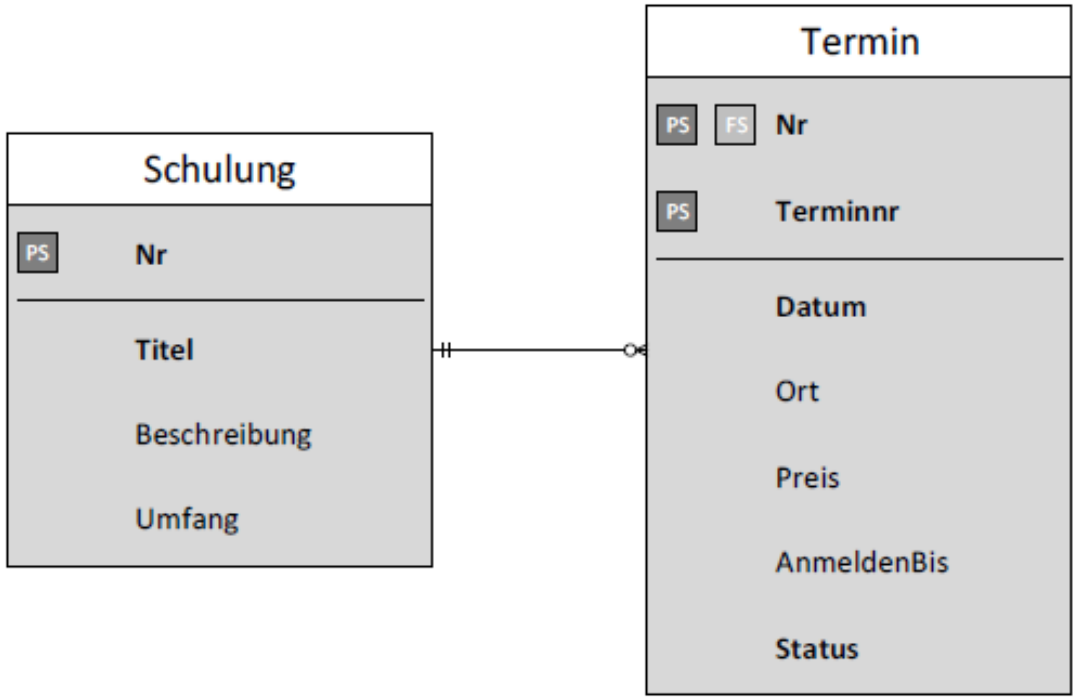


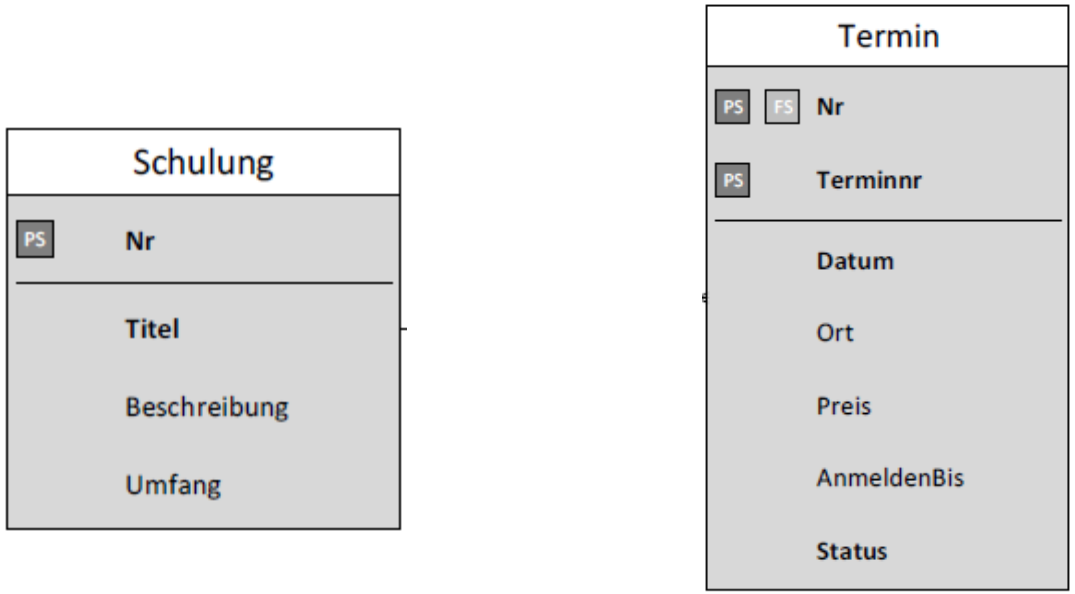
Abbildung 4-7 Beispiel Krähenfuß-Notation in der Miniwelt Schulungen

4.4.1 Entities

- Chen-Notation durch **Rechtecke** symbolisiert

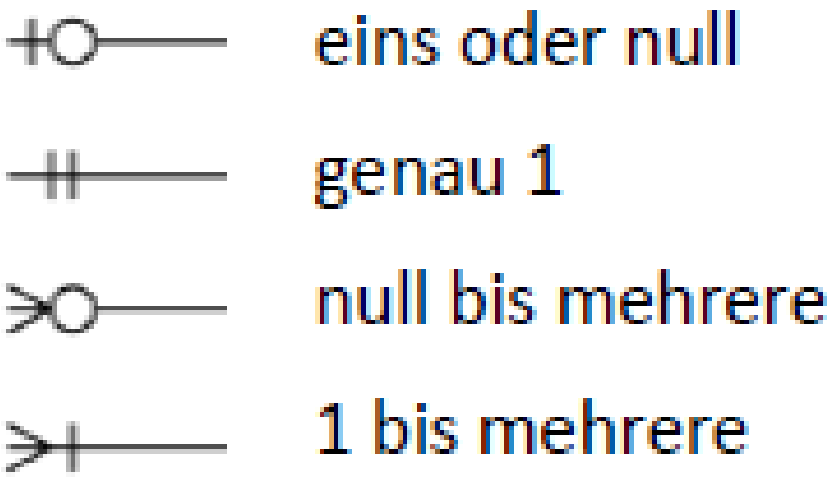
- **Vorteil**: kompakter und „vielleicht“ übersichtlicher
  Modellierung mittels UML Notation möglich

- **Nachteil**: auf Papier nicht so leicht zu erweitern, da das Rechteck nicht mitwachsen kann, wenn weitere Attribute dazukommen



4.4.2 Kardinalitäten

- Krähenfüße geben an, wie viele Entities maximal und mindestens an einer Relationship partizipieren

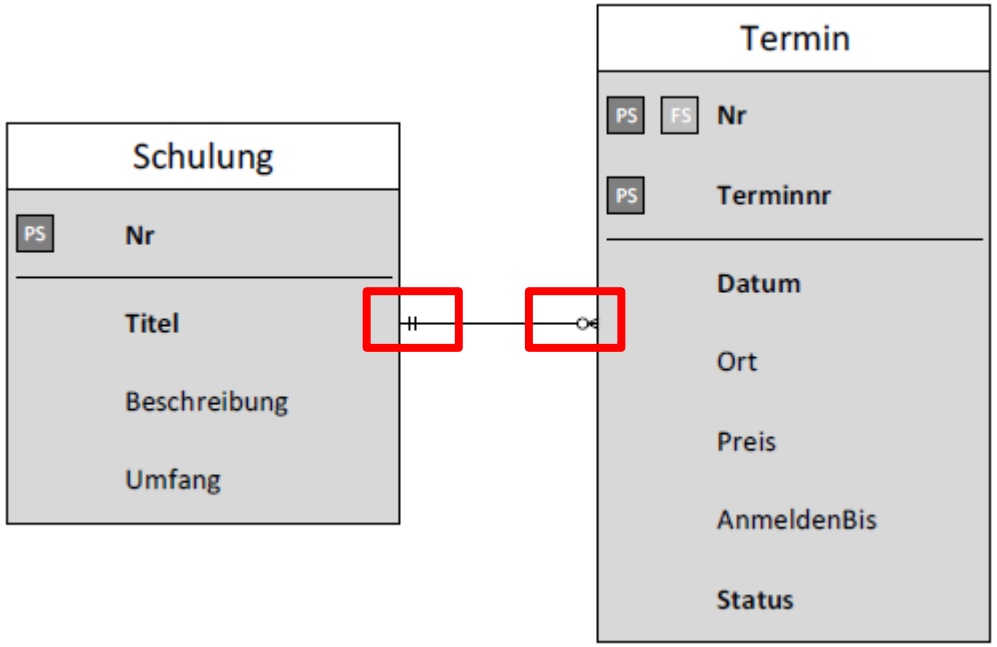


4.4.2 Kardinalitäten

- Beispiel 4.6 Kardinalitäten in Krähenfuß-Notation in der Miniwelt „Schulungen“

  Es kann pro Schulung mehrere Termine geben, möglicherweise ist aber für eine Schulung noch kein Termin geplant. Also pro Schulung kein oder mehrere Termine:

  Zu jedem Schulungstermin gehört genau eine Schulung:



4.4.3 Identifizierende und nicht-identifizierende Relationships

- **Relationship** ist in der Krähenfuß-Notation **nur durch ein Linie** symbolisiert
- zwei Arten von Relationships:
- **Identifizierend**:
  - schwaches Entity auf der n-Seite
  - schwache Entity durch das starke Entity identifiziert
  - Transformation in das relationale Datenmodell:
    - Primärschlüssel des starken Entities einen Teil des Primärschlüssels des schwachen Entities
    - schwache Entity besitzt einen zusammengesetzten Primärschlüssel
    - Primärschlüssel des starken Entities ist Fremdschlüssel im schwachen Entity und gleichzeitig Primärschlüsselattribut

- **Nicht-identifizierend**

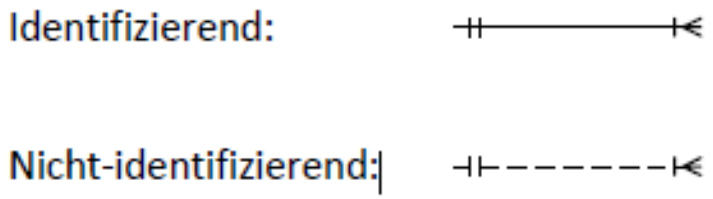
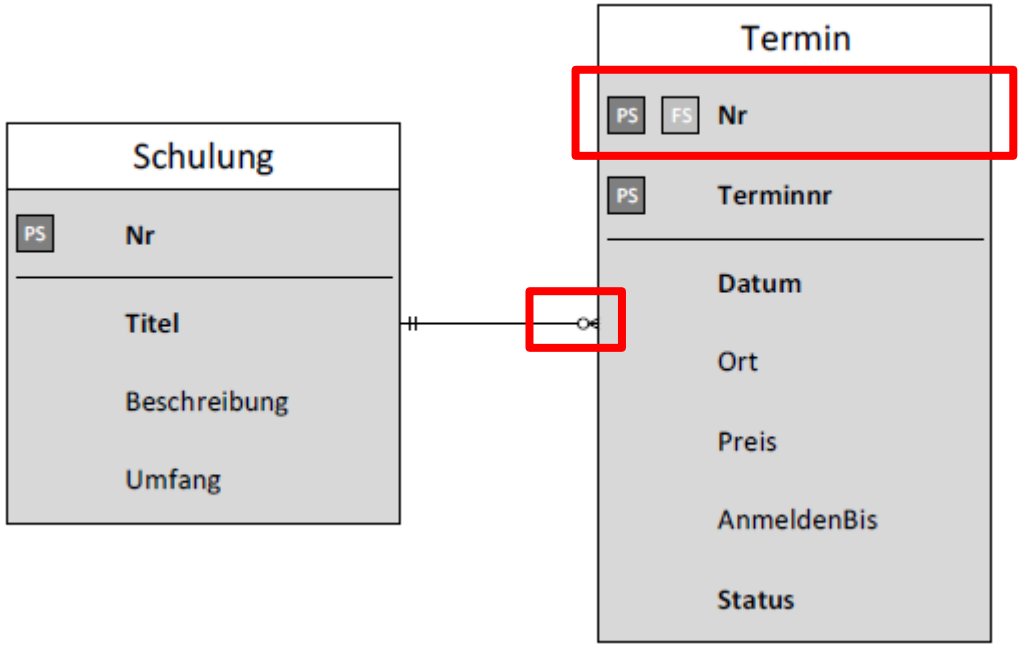


Abbildung 4-9 Krähenfuß-Notation für identifizierende und nicht-identifizierende Relationships

4.4.3 Identifizierende und nicht-identifizierende Relationships

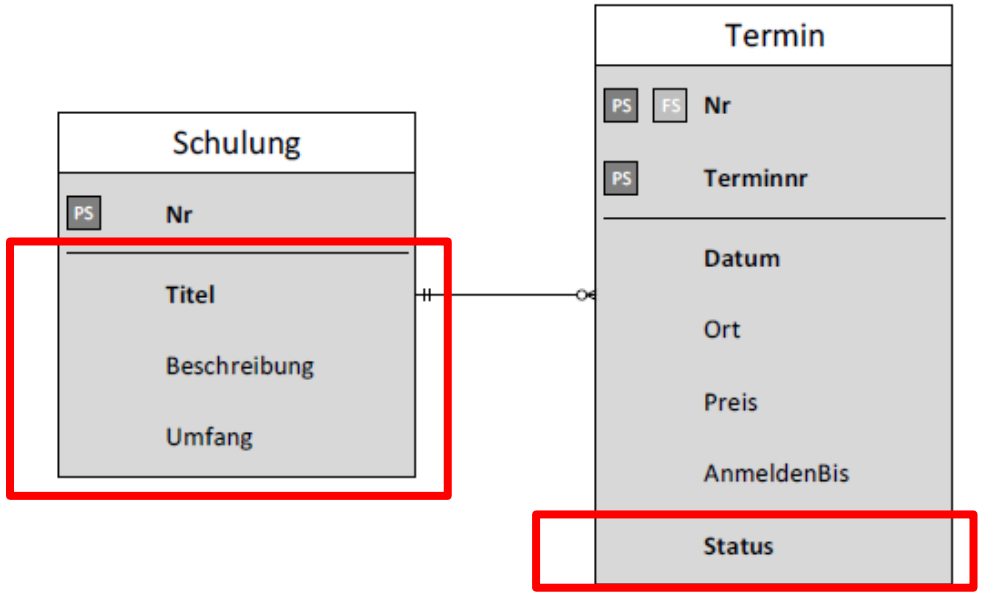
- Beispiel 4.7 Identifizierende Relationship in der Miniwelt „Schulungen“

  In der Anwendungsdomäne „Schulung“ werden die Termine pro Schulung durchnummeriert, daher ist im Krähenfuß-Modell in Abbildung 4-7 eine identifizierende Relationship modelliert.



4.4.4 Notwendige und optionale Attribute

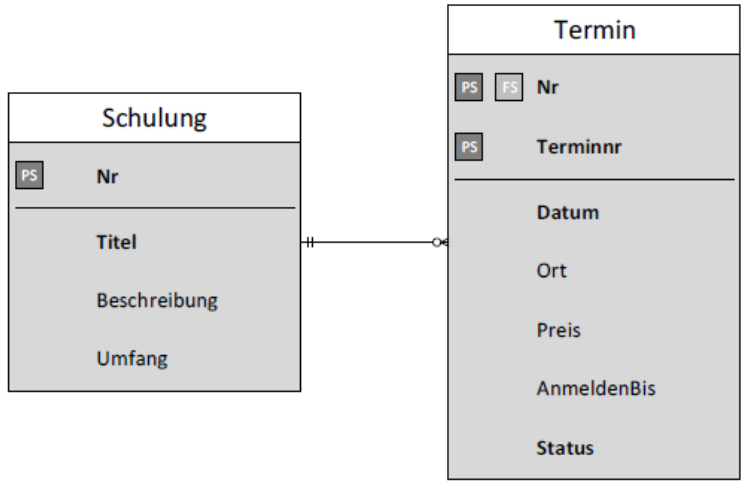
- Attribute eines Entities **zwingend einen Wert** enthalten
  - Attribute **fett drucken** oder durch besondere Symbole
- Attribute auch *Nullwerte* (das heißt keine Werte) zulässig
- Primärschlüsselattribute müssen zwingend einen Wert enthalten.



4.4.4 Notwendige und optionale Attribute

- Beispiel 4.8 Zwingend erforderliche Attribute in der Miniwelt Hochschule

In der Miniwelt der Schulungen in Abbildung 4-7 muss für jede Schulung eine Nummer und ein Titel angegeben sein. die ausführliche Beschreibung kann auch offen bleiben. Für jeden Termin muss die Nr der Schulung angeben sein, die stattfinden soll, sowie die Nummer des Termins und das geplante Datum. Der Status zeigt an, ein Schulungstermin in Planung, offen für Anmeldungen oder bereits vergangen ist. Der Status muss immer angegeben werden, Preis und Anmeldeschluss können auch offen bleiben.



4.4.5 n:m-Relationships

- **n:m-Relationships** sind in der Krähenfuß-Notation **unüblich**
- **Alternative**: zwei 1:n-Relationships und ein neues Entity

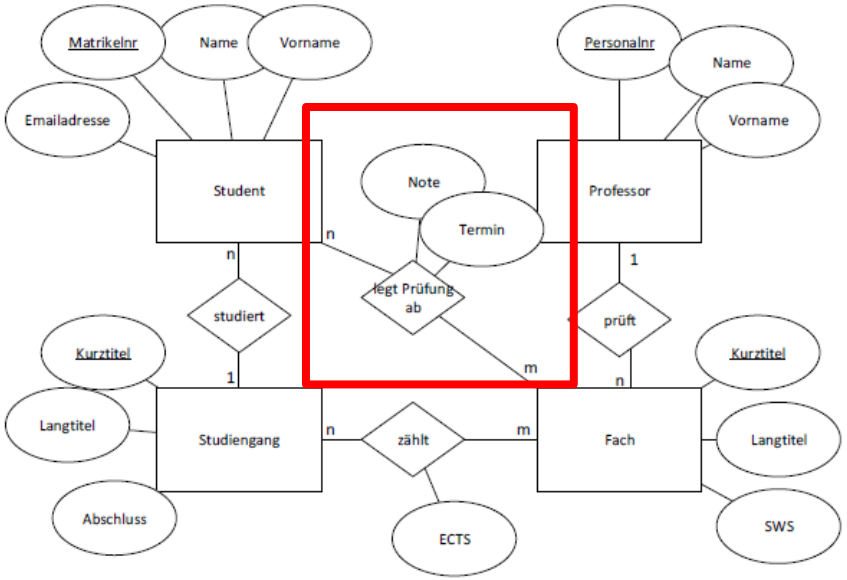


Abbildung 4-5 ER-Diagramm Hochschule in Chen-Notation

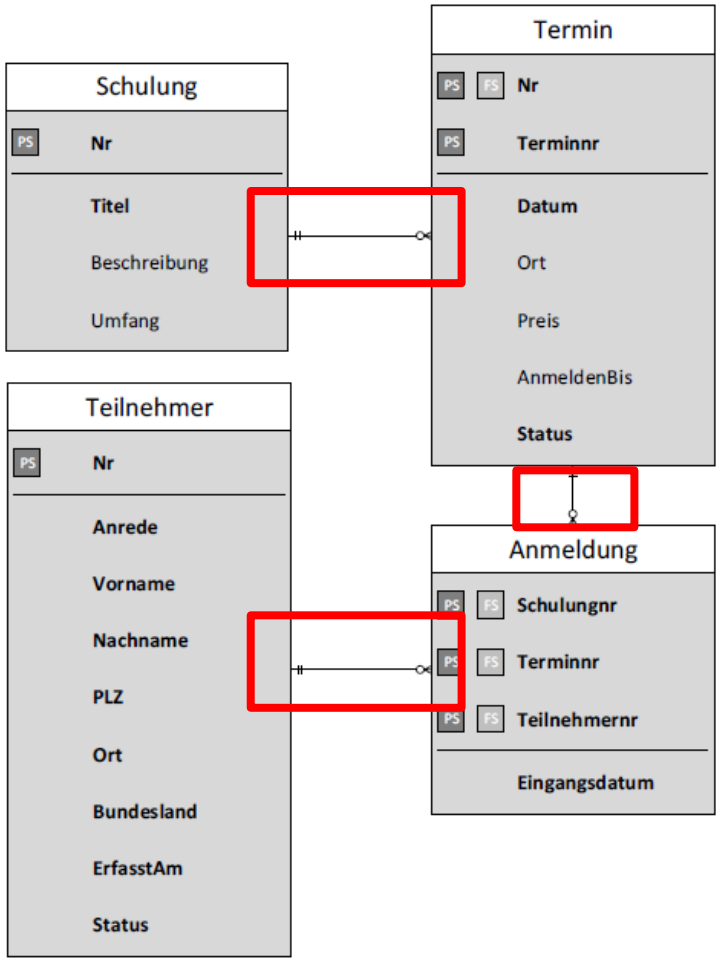


Abbildung 4-10 Krähenfuß-Notation n:m-Relationship aufgelöst in zwei 1:n-Relationships

4.4.6 Datentypen von Attributen

- Werte haben einen bestimmten Datentyp
- logischen Ebene:
  - Datentypen definieren grundsätzliche Einschränkung
- technisch-physischen Ebene:
  - Effizienz (Antwortgeschwindigkeit) und den Speicherplatzverbrauch abhängig
- Standards für relationale Datenbanken:
  - erleichtern Daten zwischen verschiedenen RDBMS (relationalen Datenbankmanagement-systemen) auszutauschen oder das RDBMS zu wechseln
- Basis-Datentypen in Tabelle rechts entspricht standardkonforme RDBMS

Tabelle 4-1 Datentypen in relationalen Datenbanken

Datentyp	Beschreibung	Beispielwerte
INT	Ganzzahlen, ist für künstliche Schlüssel geeignet	6
DEC(m, n)	Dezimalzahlen mit m Stellen insgesamt, davon n Nachkommastellen	DEC(4,2): 12.34
FLOAT DOUBLE	Gleitkommazahl	12.0634, 56663.455669
VARCHAR (n)	Zeichenketten variabler Länge mit bis zu n Zeichen, überzählige Zeichen werden abgeschnitten	VARCHAR(8): 'Ulf','Sybille', 'Matthia'
CHAR(n)	Zeichenkette fester Länge mit genau n Zeichen, fehlende Zeichen mit Leerzeichen aufgefüllt, überzählige abgeschnitten	CHAR(3): 'DE ','CH ','AUS'
DATE	Datum	'2015-10-10'
TIME	Zeit	'12:30:00'
DATETIME	Termin bestehend aus Datum und Uhrzeit	'2015-10-10 12:30:00'
TIMESTAMP	fixer Zeitpunkt unabhängig von Zeitzone, auch mit Millisekunden	'2015-10-10 12:30:00'
BLOB(N)	Binary Large Object („Bitklumpen") zur Speicherung beliebiger Daten, zum Beispiel Bilder, bis maximal n MByte	BLOB(20M): bis 20Mbyte ×ÿ¬Qí¶æ';lÂû1!ýcÑåªã□Ož øwÆ_Žó=;Áï¥¦yÒ□cƒÞrN:'É y¸)lI·v,±ÚhÚ*„çÒ9Á.

4.4.6 Datentypen von Attributen

- Beispiel 4.9 Krähenfuß-Diagramm der Miniwelt Hochschule

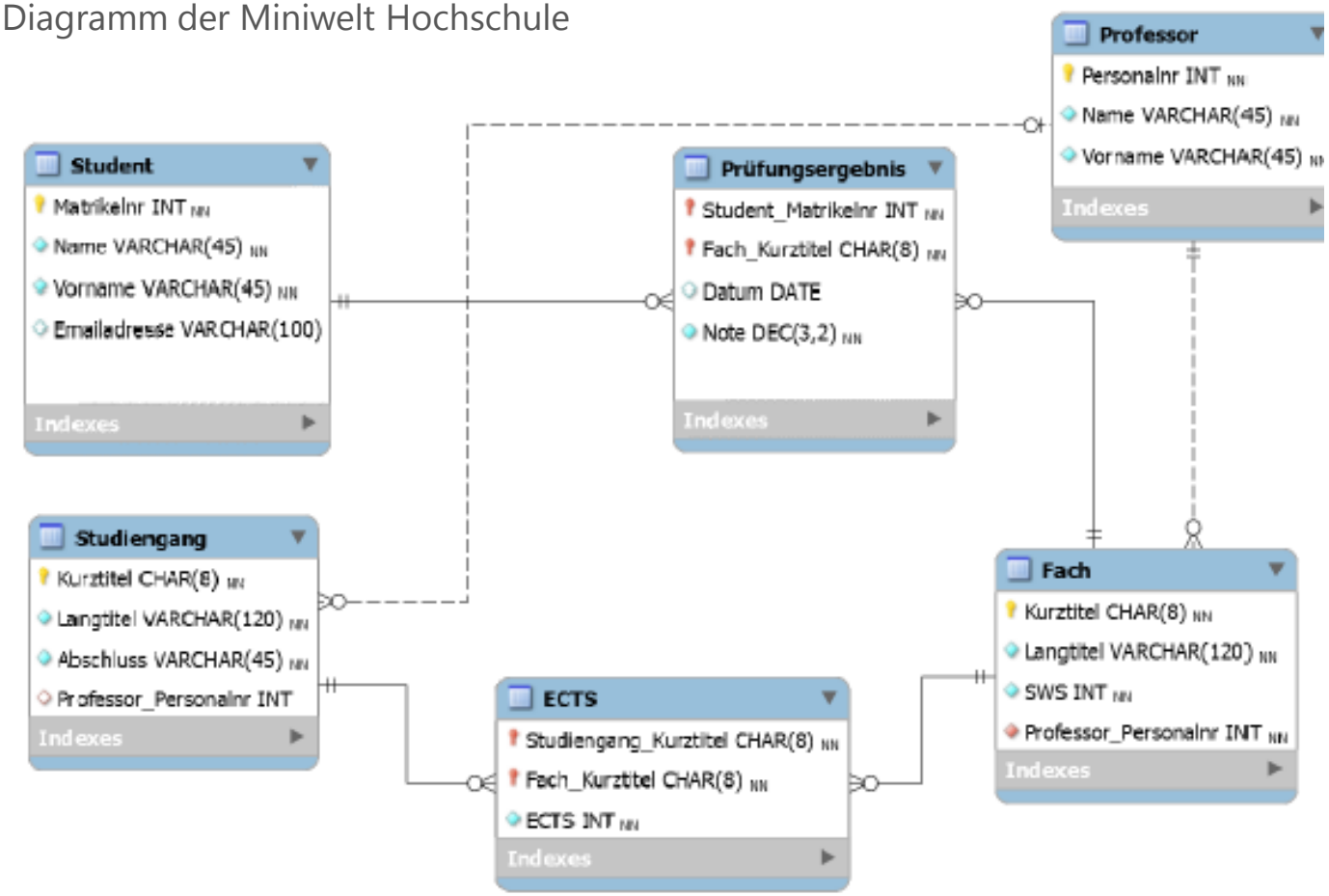


Abbildung 4-11 Krähenfuß-Diagramm der Miniwelt Hochschule mit Datentypen

4.5.1 Visualisierungssoftware zum Zeichnen von Diagrammen: MS Visio

- Die Chen-Notation benötigt nur wenige und einfache Symbole
- sehr zeichenfreundlich
- Diagramme sind leicht von Hand zu erstellen

- Visualisierungsprogramme
  - MS Visio

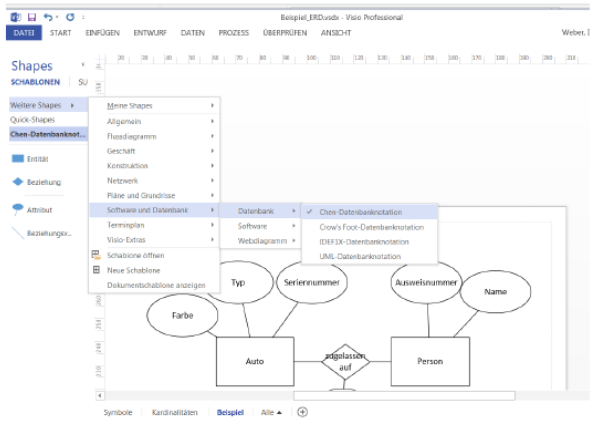
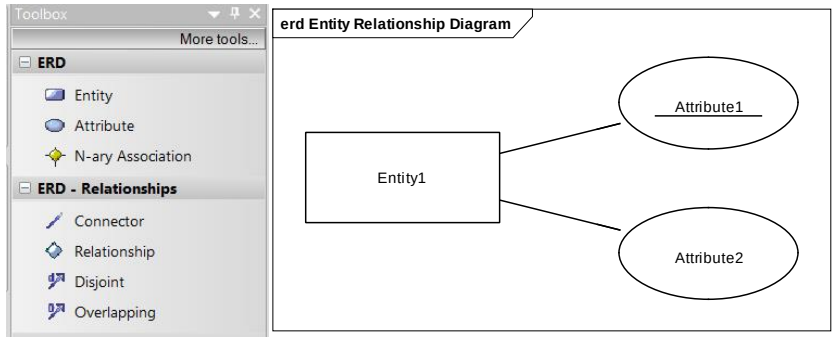
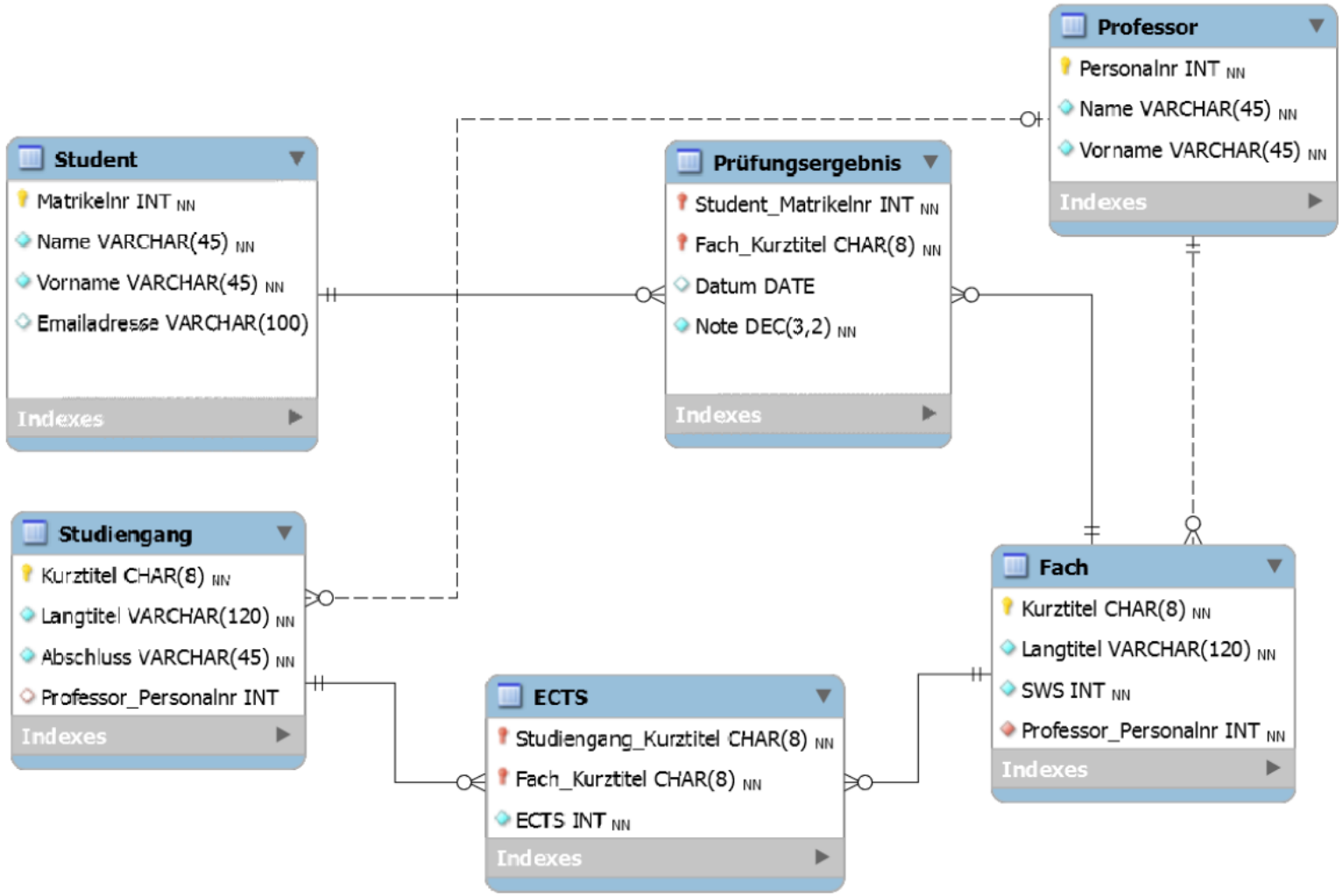


Abbildung 4-12 ER-Diagramme erstellen in MS Visio

  - Sparx Enterprise Architect



4.5.2 Modellierungssoftware zum Erstellen logischer Modelle und physischer Datenbanken durch Forward Engineering



4.6 Übungen

- Aufgabe 4.2 ER-Modellierung Beschaffung

Das Unternehmen Saubermann produziert Reinigungsmittel und bezieht dazu Rohstoffe von verschiedenen Lieferanten. Die Daten zur Beschaffung sollen in einer relationalen Datenbank abgelegt werden. Jeder Rohstoff hat eine eindeutige Nummer, eine Bezeichnung und eine Gefahrenklasse, die als Ganzzahl angegeben wird (1,2,3,4).

Jeder Lieferant hat eine eindeutige Lieferantennummer; außer dieser sind Name und Adresse relevant. Für jeden Lieferanten soll eine Lieferantenbewertung (A, B oder C) hinterlegt werden können.

Lieferanten können mehrere unterschiedliche Rohstoffe anbieten und manche A Rohstoffe sind bei mehreren Lieferanten im Angebot. In der Datenbank soll hinterlegt werden, welcher Lieferant welchen Rohstoffe zu welchem Preis und mit welcher Mindestabnahmemenge anbietet. Für einige der Rohstoffe ist jeweils ein Hauptlieferant festgelegt, bei dem der Rohstoffe normalerweise bestellt wird.

a) Modellieren Sie die Miniwelt Beschaffung als ER-Diagramm in der Chen-Notation.

b) Zeichnen Sie die Tabellen (Relationen), die diesem Diagramm entsprechen und schreiben Sie in jede Tabelle zwei plausible Beispiel-Datensätze.

c) Geben Sie die Relationenschemata zu den Tabellen an.

4.6 Übungen

- Aufgabe 4.2 ER-Modellierung Beschaffung

Das Unternehmen Saubermann produziert Reinigungsmittel und bezieht dazu Rohstoffe von verschiedenen Lieferanten. Die Daten zur Beschaffung sollen in einer relationalen Datenbank abgelegt werden. Jeder Rohstoff hat eine eindeutige Nummer, eine Bezeichnung und eine Gefahrenklasse, die als Ganzzahl angegeben wird (1,2,3,4).

Jeder Lieferant hat eine eindeutige Lieferantennummer; außer dieser sind Name und Adresse relevant. Für jeden Lieferanten soll eine Lieferantenbewertung (A, B oder C) hinterlegt werden können.

Lieferanten können mehrere unterschiedliche Rohstoffe anbieten und manche A Rohstoffe sind bei mehreren Lieferanten im Angebot. In der Datenbank soll hinterlegt werden, welcher Lieferant welchen Rohstoffe zu welchem Preis und mit welcher Mindestabnahmemenge anbietet. Für einige der Rohstoffe ist jeweils ein Hauptlieferant festgelegt, bei dem der Rohstoffe normalerweise bestellt wird.

a) Modellieren Sie die Miniwelt Beschaffung als ER-Diagramm in der Chen-Notation.

b) Zeichnen Sie die Tabellen (Relationen), die diesem Diagramm entsprechen und schreiben Sie in jede Tabelle zwei plausible Beispiel-Datensätze.

c) Geben Sie die Relationenschemata zu den Tabellen an.

4.6 Übungen

- Aufgabe 4.3 ER-Modellierung und Implementation Autohaus

  Für ein aufstrebendes Autohaus sollen Sie ein Datenbank entwickeln. Zu jedem Auto möchte man die Seriennummer, Marke, Typ und Farbe speichern können. Kunden haben eine Kundennummer, Name, Vorname, Postleitzahl, Stadt und Straße. Die Angestellten des Autohauses haben eine Personalnummer, Name und Vorname und ein Eintrittsdatum. Jedem Kunden ist ein Verkäufer zugeteilt, der ihn betreut. Wenn ein Kunde ein Auto kauft, möchte man das Verkaufsdatum, den Preis und den Verkäufer abspeichern.

  a) Modellieren Sie die Miniwelt Autohaus als ER-Diagramm in der Chen-Notation.
  b) Modellieren Sie die Datenbank als Krähenfuß-Diagramm inklusive Datentypen in der MySQL Workbench. Das Modell soll *IhrName*_autohaus heißen.
  c) Erzeugen Sie die dazugehörige Datenbank per Forward Engineering.

MHP
A PORSCHE COMPANY

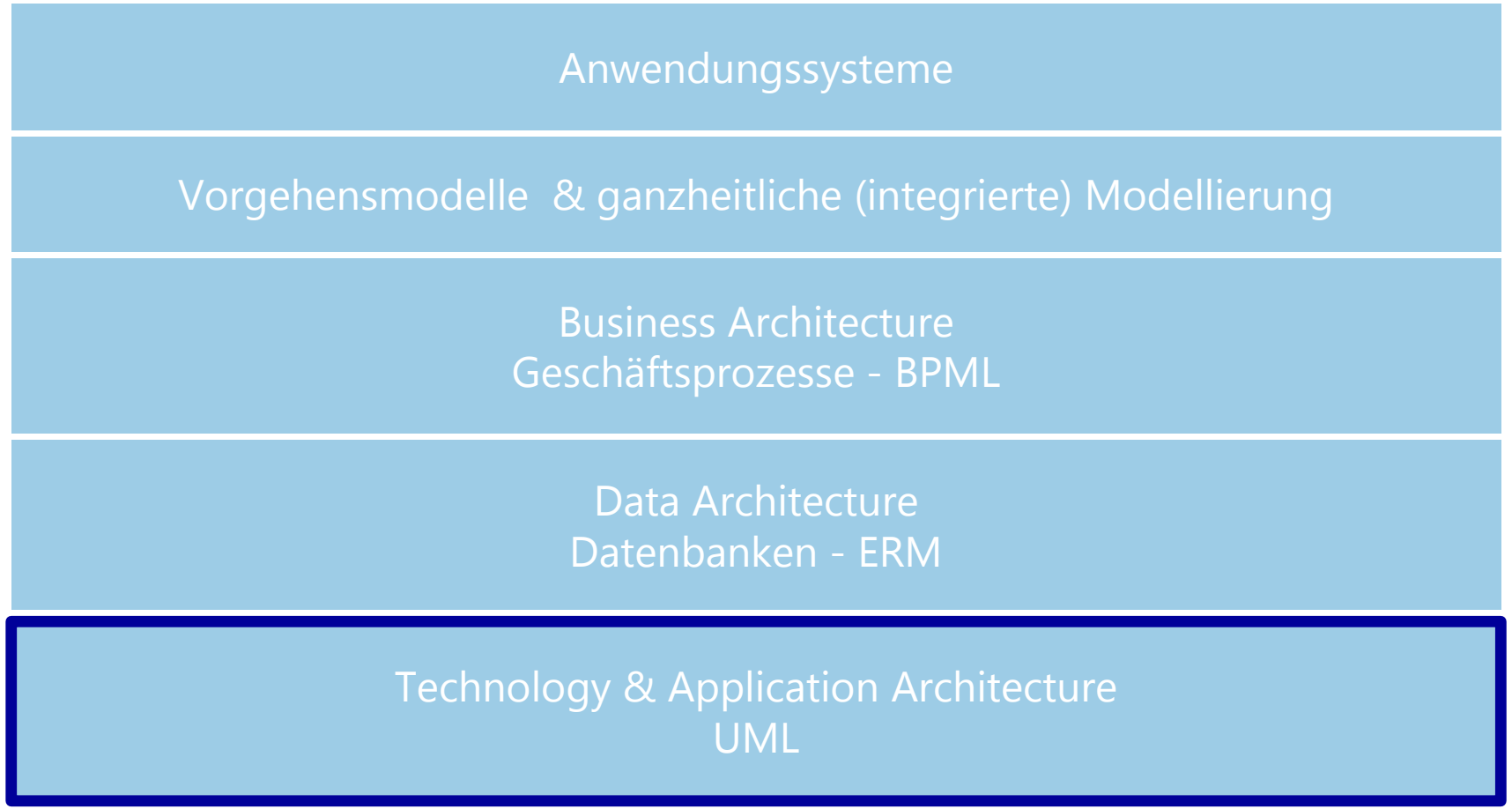
INTEGRIERTE MODELLIERUNG KOMPLEXER SYSTEME

Anwendungssysteme und UML

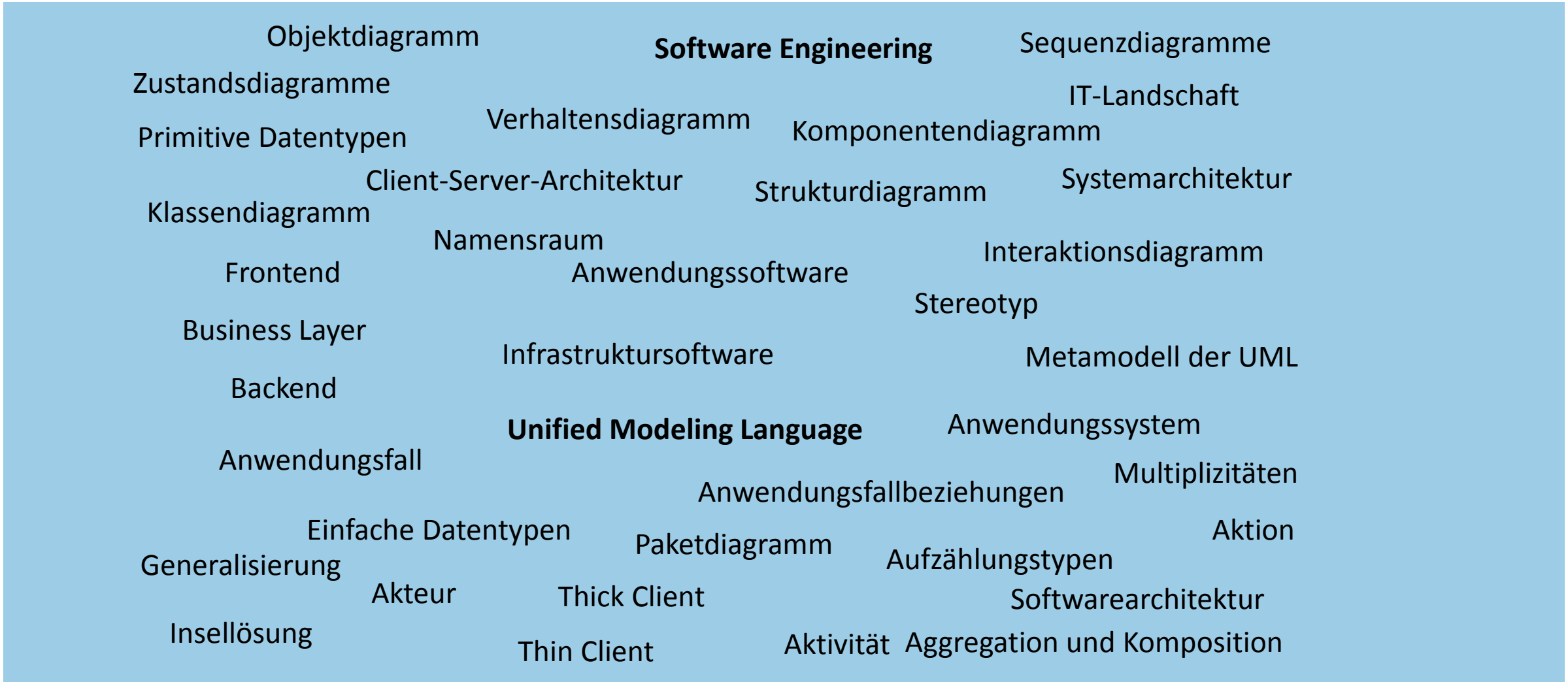
UNIVERSITATEA BABEȘ-BOLYAI

Hugo Colceag I Bachelor Studiengang Informatik

Vorlesungsinhalte und Aufbau



Mindmap - Anwendungssysteme und UML



Lernziele

- ✓ Software-Architekturen sehen und verstehen
- ✓ Was ist UML?
- ✓ UML Diagramme und ihre unterschiedlichen Einsatzbereiche

Agenda

MHP
A PORSCHE COMPANY

1. **Aufbau von Anwendungssystemen**
2. Prinzipien der SW-Technik
3. Methoden der SW Technik & UML

6.1 Anwendungssoftware und Anwendungssystem

- Daten und Informationen sind eine **wichtige** und **zentrale Ressource in Unternehmen**
- Endbenutzer nicht direkt über einen Datenbankserver auf Daten zu sondern über Anwendungsprogramme

- **Anwendungsprogramme**
  - Auf bestimmte fachliche Aufgaben zugeschnitten
  - darauf abgestimmte Bedienoberflächen
  - Nutzen im Hintergrund Dienste eines Datenbankservers

- **Umgang mit Daten**
  - Auswertung
  - mehrfach Nutzung
  - Austausch zwischen verschiedenen Anwendungssystemen

- **Ziele**:
  - Rationalisierung
  - Qualitätssteigerung
  - Services und neu Geschäftschancen

6.1 Anwendungssoftware und Anwendungssystem

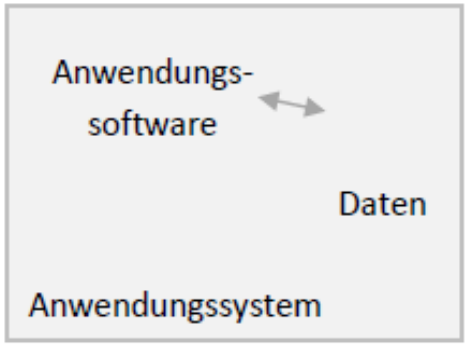
- **Anwendungssoftware**:

  Eine (betriebliche) Anwendungssoftware ist eine Software, mit der Endbenutzer eine bestimmte (betriebliche) Aufgabe oder Aufgaben aus einem (betrieblichen) Anwendungsgebiet bearbeiten können.

- Anwendungssystem (**im engeren Sinn):**

  ein Anwendungssystem ist eine Anwendungssoftware zusammen

  mit den damit erstellten und bearbeiteten Daten



- Anwendungssystem (**im weitergefassten Sinn**):

  ein Anwendungssystem im engeren Sinn **einschließlich** aller

  Hardware und Systemsoftware, die zum Betrieb der Anwendungssoftware nötig sind.

- **Infrastruktursoftware** (Systemsoftware) stellt Funktionen und Dienste bereit, die die eigentliche Anwendungssoftware nutzen kann, die Endnutzer aber nicht routinemäßig direkt bedienen.

6.1 Anwendungssoftware und Anwendungssystem

MHP
A PORSCHE COMPANY

- In Unternehmen sind viele Softwaresysteme im Einsatz

- **IT-Landschaft** oder **Systemlandschaft**: Gesamtheit aller IT-Systeme, die ein Unternehmen verwendet

- **unübersichtliche** „gewachsene" IT-Landschaften, die aufwändig zu warten sind
- häufig bestimmte Funktionen in verschiedenen Systemen **mehrfach** zu Verfügung

- **Ziel IT-Landschaft** :
  - Integrierte Informationsverarbeitung
  - Vernetzung („integriert")
  - durchgängige Informationsflüsse über Systemgrenzen

  - schnelle Informationsverarbeitung
  - widerspruchsfreie Daten
  - Vermeidung von Doppelarbeit

- **Insellösung**: Anwendungssystem, das vom Datenaustausch mit den anderen Systemen in der IT-Landschaft abgekoppelt ist

6.2 Softwarearchitektur: Komponenten von Anwendungssoftware

MHP
A PORSCHE COMPANY

- Moderne Anwendungssoftware:
  - großen Funktionsumfang
  - vielfältige Einstellungsmöglichkeiten
  - flexibel und komfortabel über grafische Bedienoberflächen zu bedienen
  - Mehrbenutzerbetrieb
  - **komplex** in der
    - Erstellung
    - Wartung
    - Weiterentwicklung

- **Software Engineering**: „Die Kunst und das Handwerk, komplexe Software-Systeme zu entwerfen und bauen“

- Maßnahmen:
  - Funktionalität in einzelne Komponenten verteilen
  - Komponenten unabhängig voneinander entwickelbar
  - Komponenten wirken über fest definierte Schnittstellen zusammen
  - Zusammen bilden die Komponenten die Anwendungssoftware

6.2 Softwarearchitektur: Komponenten von Anwendungssoftware

- **Software-Architektur**

  Als Software -Architektur bezeichnet man den Aufbau einer Software aus Komponenten und deren Zusammenspiel.

Sehr verbreitet ist eine Unterteilung der Anwendungssoftware in drei Schichten

- **Frontend** / Präsentationsschicht
  - dient der Darstellung der Daten und Informationen
  - Interaktion mit dem Benutzer
  - meist grafische Bedienoberfläche

- **Business Layer** / Logikschicht / Anwendungsschicht / Geschäftsschicht
  - eigentliche Anwendung
  - Regeln, Prozesse, Workflows, etc. …

- **Backend** / Persistenzschicht
  - Datenspeicherung und Datenbereitstellung

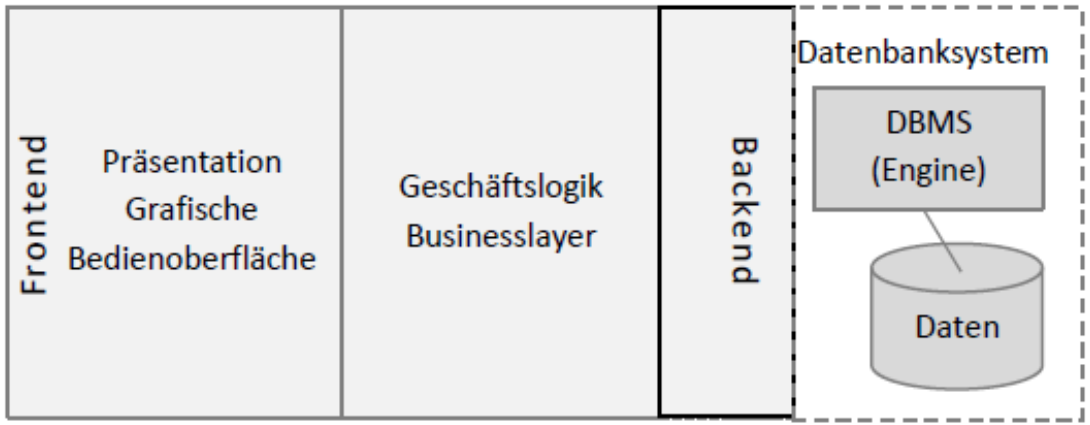


Abbildung 6-2 Aufbau einer Anwendungssoftware: 3-Schicht-Architektur

6.2 Softwarearchitektur: Komponenten von Anwendungssoftware

- Beispiel 6.1 Architektur einer Anwendungssoftware „Autohaus“

  Eine Anwendungssoftware für ein Autohaus erlaubt es unter anderem, Probefahrten zu terminieren und erfassen.

  Das **Frontend** bietet auf einer grafischen Bedienoberfläche eine Eingabemaske, in der der Benutzer Kunde und Auto eingeben oder aus dem Kunden- und Autostamm auswählen kann.

  Das **Backend** ruft Daten zu vorhandenen Kunden und Autos aus dem Datenbankmanagementsystem ab und übergibt eingegebene Daten zu neuen Kunden und zur Probefahrt an das Datenbankmananegemt, das die Daten in die Datenbank persistiert.

  Die **Geschäftslogik-Schicht** enthält unter anderem Funktionen, die Terminüberschneidungen verhindern und dafür sorgen, dass nicht versehentlich ein Rückgabezeitpunkt eingegeben wird, der vor dem Startzeitpunkt liegt.

6.3 Systemarchitektur

- „verteilten System“

  Anwendung ist verteilt auf mehrere Rechnern (Hardware)

- **System-Architektu**r

  Die Art, wie Software-Komponenten auf Hardware-Komponenten verteilt sind und wie sie zusammenwirken, bezeichnet man als System-Architektur.

6.4 Client-Server-Architektur

- **Server**
  - Systemkomponente, die Dienste bereitstellt
  - Die Server-Software läuft ständig
  - wartet auf Anforderung von Dienste

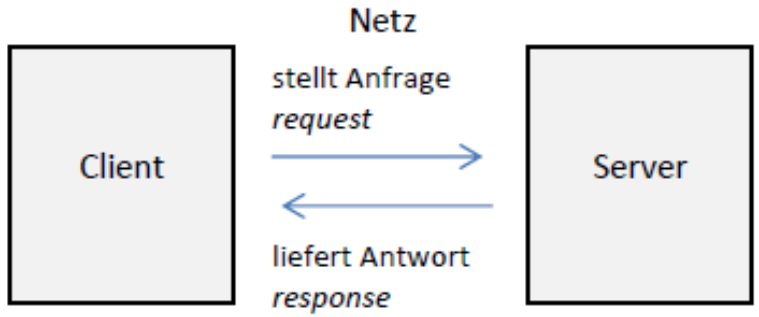


Abbildung 6-3 Client-Server-Architektur

- Kommunikation geht vom Client aus: Anfrage (*request*) an den Server

- Ein Server kann viele Clients bedienen.
- Clients können die Dienste mehrerer Server nutzen

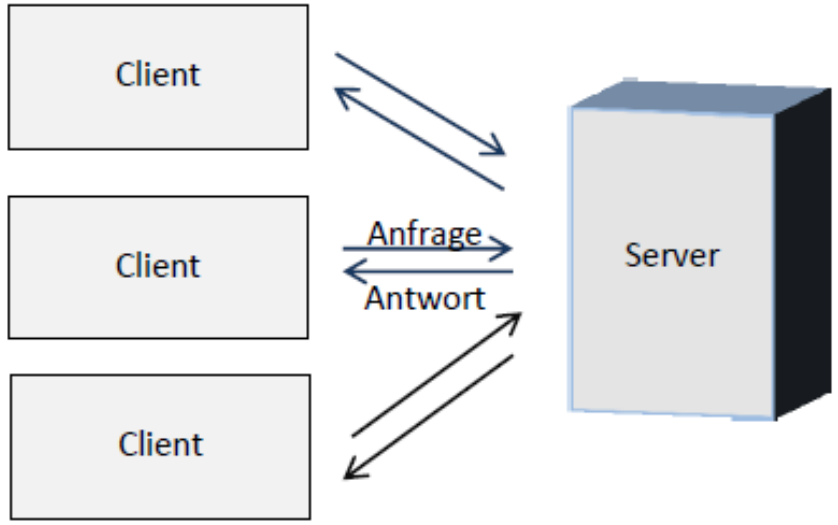


Abbildung 6-4 Ein Server bedient mehrere Clients

6.4 Client-Server-Architektur

- Beispiel 6.2 Beispiel Client-Server-Architektur: MySQL Workbench

  In einem System aus MySQL Workbench und MySQL Datenbankmanagementsystems hat die MySQL Workbench hat die Rolle des **Client** inne: sie sendet Befehle (requests) an den MySQL **Server**, der die Befehle ausführt und eine Antwort zurücksendet, beispielsweise in Form eines Datenmenge. Die MySQL Workbench biete eine grafische Benutzeroberfläche und Anwendungsfunktionen, eta den Entwurf von Modellen und Lesen und Schreiben von SQL Script-Dateien.

  Mehrere MySQL Workbench-Clients können gleichzeitig die Dienste des Datenbankservers nutzen.

6.4.1 Thin Clients und Thick Clients

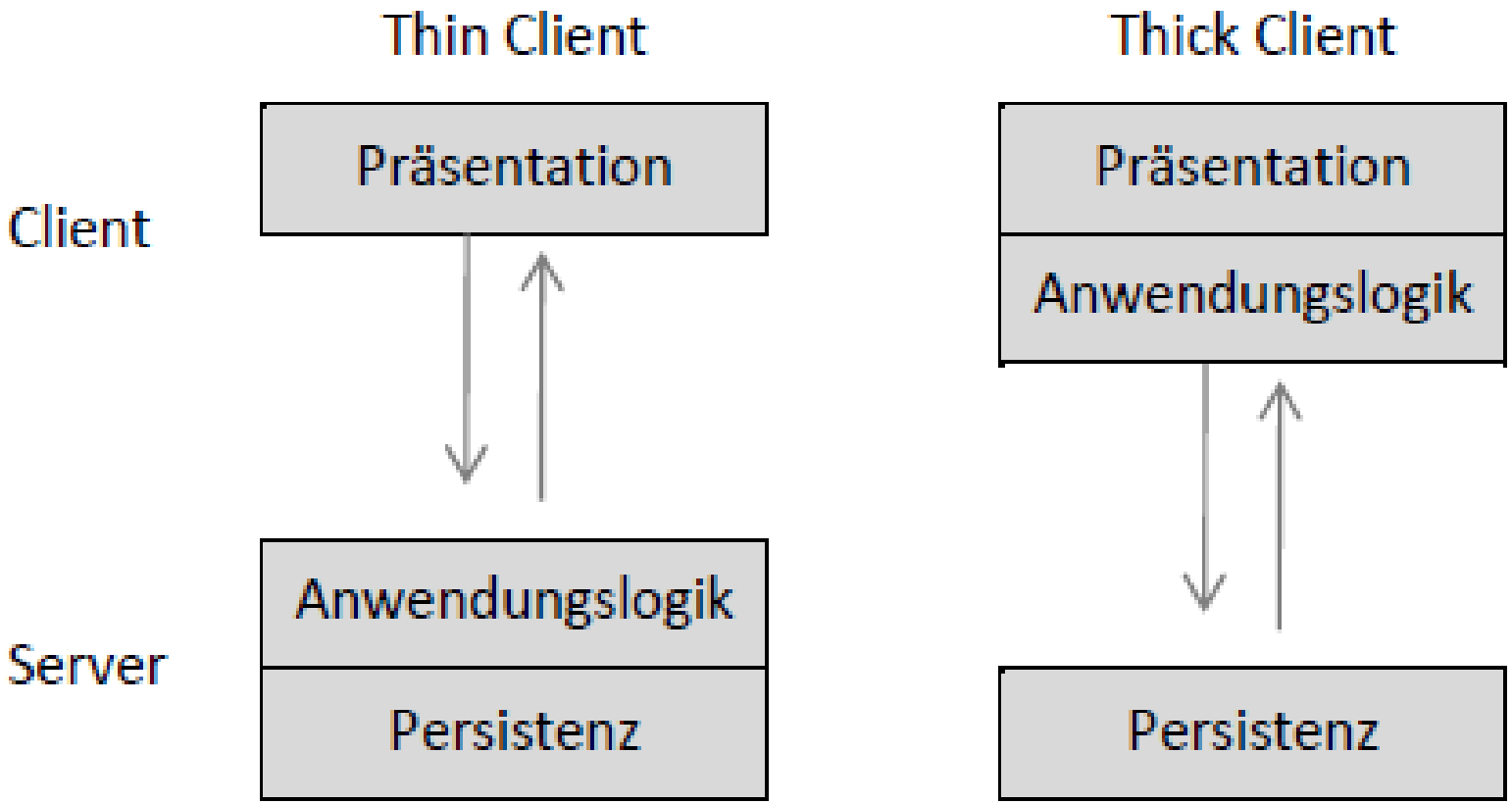


Abbildung 6-5 Thin Client – Thick Client

6.4.1 Thin Clients und Thick Clients

MHP
A PORSCHE COMPANY

- Beispiel 6.3 Beispiel „**Thin Client**“: Wikipedia

  Wikipedia ist ein System, mit dem Artikel erfasst, editiert, angezeigt und kommentiert werden können. Anwendungsfunktionen wie Benutzerverwaltung, Verwaltung von Versionsständen, Freigabe von Artikeln liegen komplett auf dem Server.

- Beispiel 6.4 Beispiel „**Thick Client**“: MySQL Workbench

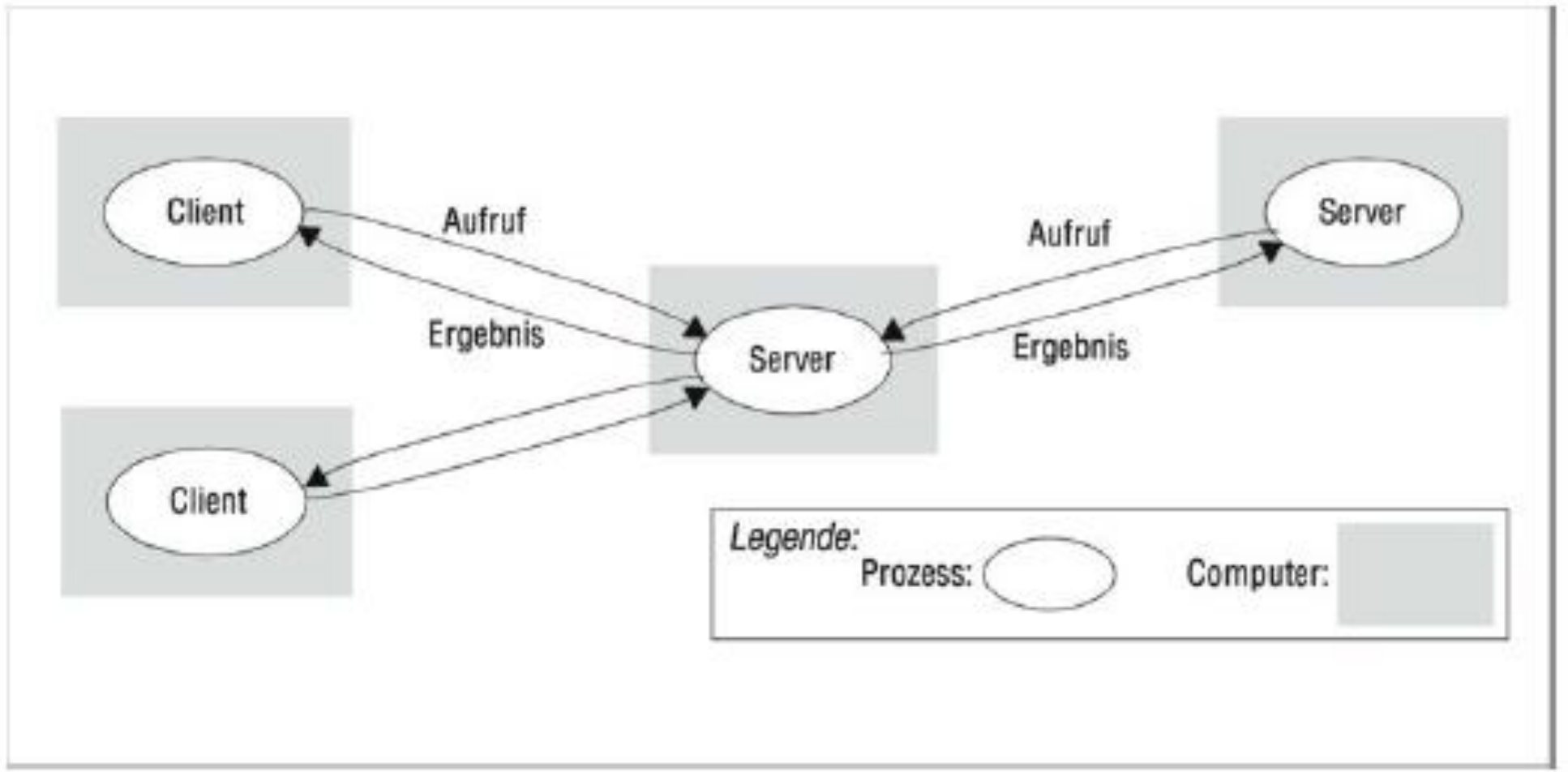
  Die MySQL Workbench ist eine eigenständige Software, die auch ohne Anbindung an einen Backend-Server sinnvoll genutzt werden kann. Die Anwendungsfunktionen (Modelle erstellen, Skripte speichern und laden) stecken im Client. Der Server übernimmt lediglich die Persistenz.

6.4.2 Vorteile und Nachteile von Thin Clients gegenüber Thick Clients

MHP
A PORSCHE COMPANY

- **Thin Client:**
  - belastet die Server-Ressourcen generell stärker
  - Hardware-Anforderungen der Server-Seite sind größer
  - Anforderungen an die Hardware der Client-Seite sind geringer
  - weniger komfortabel zu bedienen
  - „Gefühlt" langsam – Hohe Wartezeiten *(abhängig von Art und Menge der zu übertragenden Daten)*
  - belastet die Netzverbindung stärker und ist abhängig vom Netz.
  - ermöglicht vielen Nutzern gleichzeitig Zugriff auf dieselben Daten
- **Thick Client**
  - bessere Bedienbarkeit für Benutzer
  - offline nutzbar (!!! Datenreplikation und Datensynchronisation)
  - Auf Client-Rechnern muss Software installiert und gewartet werden
  - Client-Software abhängig von Client-Hardware
- **Webbrowser als Thin Client:**
  - Als Webanwendung bezeichnet man eine Anwendung, die auf der Client-Seite komplett im Webbrowser läuft. Die Client-Seite bedient sich eines Standard-Browsers, um eine graphische Benutzeroberfläche anzuzeigen. Sämtliche Anwendungsfunktionen stecken auf der Server-Seite. Hier ist auf der Client-Seite keinerlei Software-Installation und -Wartung nötig.

6.5 3-tier und n-tier-Architekturen



6.6 Datenbankanwendungen mit MS Access

MHP
A PORSCHE COMPANY

- MS Access ist eine (Standard) Software, die es Endnutzer ermöglicht, Datenbanken anzulegen, Daten zu verwalten und Datenbankanwendungen inklusive Bedienoberfläche und Geschäftslogik zu erstellen.
- **Vorteile**:
  - übersichtliche Bedienoberfläche
  - viele Assistenzfunktionen
  - grafische Bedienoberflächen für Datenbanken
  - „Datenbank-Engine“, die auf dem relationalen Datenbankmodell beruht und SQL beherrscht
  - Datenbanken und Anwendungen für andere Systeme kompatibel und zugänglich
  - Zugriff auf externe Datenbanken
  - Erweiterbarkeit durch VBA Programmierung

- **Nachteile**:
  - begrenzte Zahl von Benutzern (maximal 10 bis 20) zur gemeinsamen Benutzung empfohlen
  - Eingeschränkte Funktionen zur Benutzerverwaltung und zum Vergabe von Zugriffsrechten

- MS Access eignet sich daher besonders als Tool für **Rapid Prototyping**, also dazu, schnell und einfach Softwarelösungen zu realisieren und bei Bedarf ganz oder teilweise auf leistungsfähigere Werkzeuge und Technologien umzusteigen.

6.7 Eine typische Web-Anwendung: Wordpress

- freie quelloffene Software, die dynamisch Webseiten erzeugt
- kostenlos nutzbar
- „Software für Blogs / Artikeln, die in der Reihenfolge ihrer Erstellung veröffentlicht werden
- Erweiterbar durch Plugins
- Unterstützung von Themes: Sammlung von Darstellung- und Layoutanweisungen
- In eCommerce-Anwendungen einsetzbar
- leichtgewichtiges“ Content Management-System

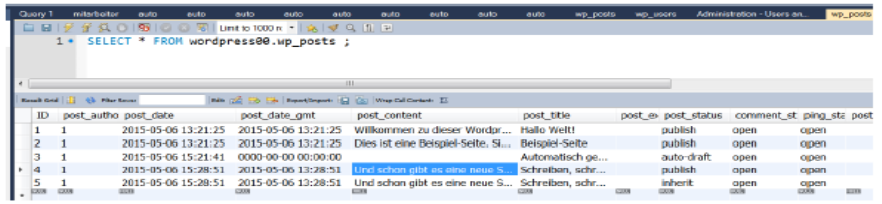


Abbildung 6-7 Wordpress-Datenbanktabelle mit Inhalten

- **Content Management-System**

  Software zur Verwaltung von Webinhalten bezeichnet man Content Management-System

- **Beispiele**
  - http://www.sonymusic.com/
  - http://www.news-sap.com/

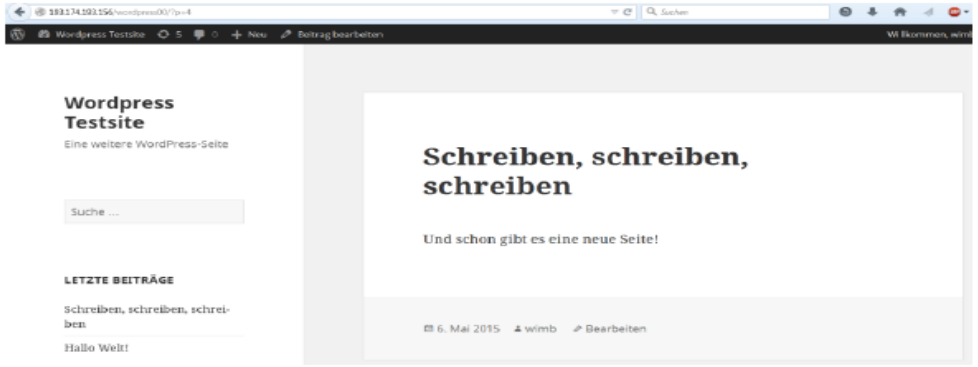


Abbildung 6-6 Die Wordpress-Seite

6.8 Übungen

- Aufgabe 6.2 Beispielanwendung mit Wordpress

  Eine Wordpress-Site hat zwei Ansichten:

  1. die öffentliche Ansicht für Leser ist zugänglich unter (hier) 193.174.193.156/wordpress00/
  2. die Admin-Seite für Blog-Betreiber ist zugänglich unter (hier) 193.174.193.156/wordpress00/wp-admin

  a) Sich mit der Anwendung vertraut zu machen:

  – Am Blog anmelden
  – Theme anpassen
  – 2 Seiten erstellen
  – 2 Blogeinträge

  b) **Anwendungsmöglichkeiten**:

  – Projekt/Projektarbeit: Kann ein Blog die Kommunikation per Email ersetzen? Vorteile und Nachteile?
  – Wofür kann man einen Blog beruflich nutzen? Bitte nennen Sie 3 Anwendungsbeispiele mit Nennung und Erläuterung von Aufgabe, Anwendern und Nutzern

Agenda

MHP
A PORSCHE COMPANY

1. Aufbau von Anwendungssystemen
2. **Prinzipien der SW-Technik**
3. Methoden der SW Technik & UML

Einführung Prinzipien

Was ist ein Prinzip?

Nach welchen Prinzipien handeln Sie?

Welche Prinzipien fallen Ihnen zur Softwaretechnik ein?

Man sollte nicht mit dem Kopf durch die Wand rennen!

Man sollte nicht den Ast absägen, auf dem man sitzt!

Prinzipien der SW-Technik:

- Prinzip der Abstraktion
- Prinzip der Bindung und Kopplung
- Prinzip der Hierarchisierung
- Prinzip der Modularisierung
- Geheimnisprinzip

Prinzipien

- sind Grundsätze, die man seinem Handeln zugrunde legt,
- sind allgemeingültig, abstrakt, allgemeinster Art,
- bilden eine theoretische Grundlage,
- werden aus Erfahrung gewonnen.

Motivation

Warum Prinzipien für die SW-Technik?

Alle Prinzipien der Softwaretechnik haben das Ziel, **Komplexität** zu **reduzieren** und **beherrschbar** zu machen.

Welches Ergebnis erhält man durch die Anwendung der Prinzipien?

Als Ergebnis der Anwendung von Prinzipien entstehen **einfache und verständliche Systemdokumentationen und Entwürfe**, sowie **wartungs-** und **änderungsfreundliche Systeme**.

Agenda

MHP
A PORSCHE COMPANY

1. **Grundlagen der Modellierung**
   - Prinzipien der SW-Technik
   - **Prinzip der Abstraktion**
   - Prinzip der Bindung und Kopplung
   - Prinzip der Hierarchisierung
   - Geheimnisprinzip
   - (Softwareentwicklungs-) Werkzeuge

2. Einleitung Vorgehensweise ganzheitliche Modellierung

Bekanntes Ölgemälde von René Magritte (1898–1967)

Dies ist keine Pfeife.

Was will der Künstler mit diesem Bild sagen?

Was bedeutet Abstraktion

MHP
A PORSCHE COMPANY

- Oft spricht man anstelle von Abstraktion auch von Modellbildung.
- Durch Abstrahieren vom Konkreten erstellt man ein Modell der realen Welt.

Ein Modell ist die Abstraktion von Konkretem.

- Das Gegenteil von Abstrahieren ist Konkretisieren.
- Unter Abstrahieren versteht man
  - das Herausheben des Wesentlichen,
  - das Beiseite lassen von Unwesentlichem,
  - das Erkennen gleicher Merkmale.

Abstraktion bedeutet immer Konzentration auf das Wesentliche.

Die Bedeutung der Abstraktion für das SW-Engineering

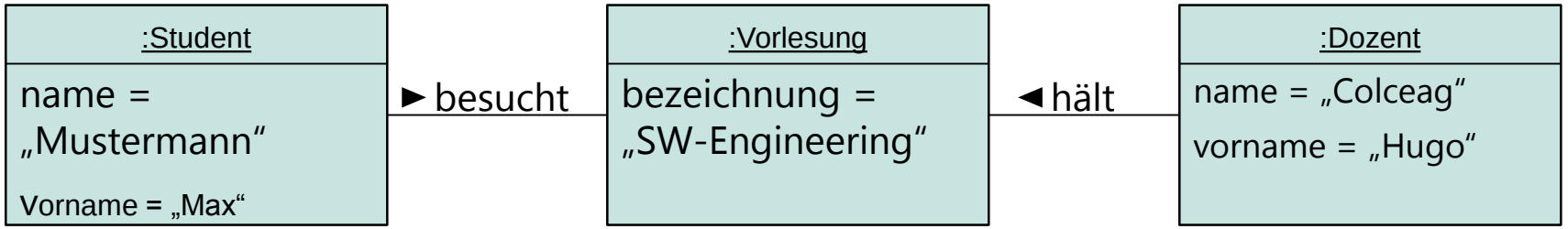
MHP
A PORSCHE COMPANY

- Kennen Sie die Aussagen:
  - „Das ist mir zu abstrakt!“
  - „Das ist mir zu abgehoben!“.
- Beim SW-Engineering geht es darum, die geeigneten Abstraktionen zu finden (für Kunden, für Vorgesetzte, für SW-Entwickler, etc.)
- Bei der Objektorientierten Modellierung werden Modelle bzw. Sichten auf das zu realisierende System erstellt. Es muss also ständig geeignet abstrahiert werden.
  - Abstraktion zur Abgrenzung des Problembereiches.
  - Abstraktion zum Finden der Objekte.
  - Die Klasse ist eine Abstraktion (ein Modell) von konkreten Objekten.
  - Abstraktion zum Erkennen der Datenfelder und Methoden.
  - …

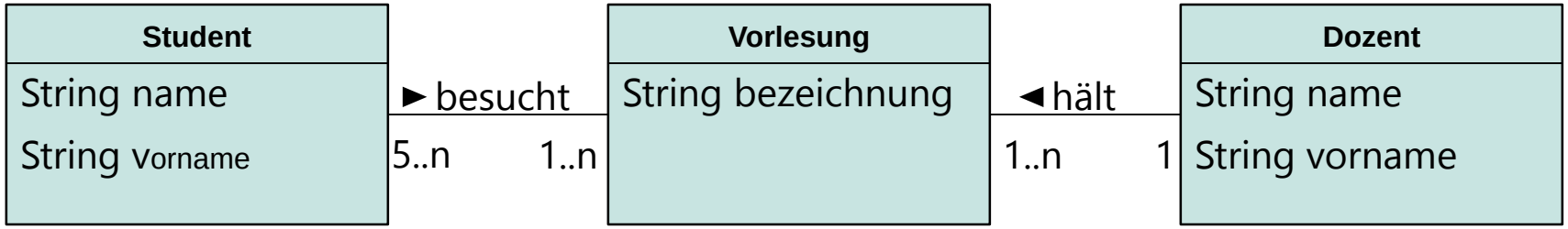
„Abstrahieren ist eine äußerst anspruchsvolle Tätigkeit, denn es ist meist sehr schwierig, geeignete Abstraktionen zu finden und aus vielen konkreten Tatsachen das Wesentliche zu isolieren.“ [Balzert]

Abstraktionsebenen – Exemplare und Typen

- **Exemplar-Ebene (Objekte)**



- **Typ-Ebene (Klassen)**



Exemplar-Ebene entspricht in der Objektorientierung dem Objekt-Diagramm.
Typ-Ebene entspricht in der Objektorientierung dem Klassendiagramm.

Agenda

MHP
A PORSCHE COMPANY

1. Prinzipien der SW-Technik
   - Prinzip der Abstraktion
   - **Prinzip der Bindung und Kopplung**
   - Prinzip der Hierarchisierung
   - Geheimnisprinzip
   - (Softwareentwicklungs-) Werkzeuge

2. Einleitung Vorgehensweise ganzheitliche Modellierung

Bindung und Kopplung in der SW-Technik

- Bindung (cohesion)

  Qualitatives Maß für die Kompaktheit einer Komponente.

  Es wird geprüft, ob die Bestandteile einer Komponente ein hohes Maß an Zusammengehörigkeit aufweisen und wie viele Aufgaben in der Komponente erledigt werden.

  High Cohesion (stark gebundenes System): Jede Komponente ist für eine definierte Aufgabe verantwortlich.

- Kopplung (coupling)

  Gegenstück zur Bindung.

  Qualitatives Maß für die Schnittstellen zwischen den Komponenten.

  Der Kopplungsgrad bestimmt, wie einfach oder schwierig es ist, Änderungen an einem System vorzunehmen.

  Low Coupling (schwach gekoppeltes System): Änderungen beschränken sich optimalerweise nur auf eine Komponente.

Optimierung erfolgt immer nach Low Coupling und High Cohesion.

Praktische Beispiele für Low Coupling und High Cohesion

MHP
A PORSCHE COMPANY

- Client-/Server-System
  - Low Coupling: möglichst wenig Kommunikation zwischen Client und Server.
  - High Cohesion: Berechnungsalgorithmen, die kontinuierlich auf Daten zugreifen, werden auf dem Server implementiert.

- Schichtenmodelle für Informationssysteme
  - High Cohesion: Zusammengehöriges wird in eine Schicht gepackt.
  - Schicht für die grafische Bedienoberfläche, Separierung der Business Logik in einer Schicht, Schicht für den Zugriff auf die Daten.
  - Grund: Die Business Logik ist sehr stabil, während sich die Technologie für die grafische Bedienoberfläche und Datenhaltung häufig ändert

- Klasse
  - High Cohesion nach Innen: Daten und Methoden, die auf den Daten arbeiten, werden in einer Einheit untergebracht.
  - Low Coupling nach außen: Kommunikation zwischen den Klassen wird minimiert.

Low Coupling und High Cohesion bedingen sich meistens gegenseitig.

Praktische Beispiele für Low Coupling und High Cohesion

MHP
A PORSCHE COMPANY

- Client-/Server-System
  - Low Coupling: möglichst wenig Kommunikation zwischen Client und Server.
  - High Cohesion: Berechnungsalgorithmen, die kontinuierlich auf Daten zugreifen, werden auf dem Server implementiert.

- Schichtenmodelle für Informationssysteme
  - High Cohesion: Zusammengehöriges wird in eine Schicht gepackt.
  - Schicht für die grafische Bedienoberfläche, Separierung der Business Logik in einer Schicht, Schicht für den Zugriff auf die Daten.
  - Grund: Die Business Logik ist sehr stabil, während sich die Technologie für die grafische Bedienoberfläche und Datenhaltung häufig ändert

- Klasse
  - High Cohesion nach Innen: Daten und Methoden, die auf den Daten arbeiten, werden in einer Einheit untergebracht.
  - Low Coupling nach außen: Kommunikation zwischen den Klassen wird minimiert.

Low Coupling und High Cohesion bedingen sich meistens gegenseitig.

Agenda

MHP
A PORSCHE COMPANY

1. Prinzipien der SW-Technik
   - Prinzip der Abstraktion
   - Prinzip der Bindung und Kopplung
   - **Prinzip der Hierarchisierung**
   - Geheimnisprinzip
   - (Softwareentwicklungs-) Werkzeuge

Vererbungshierarchie

Ein weiteres Mittel zur Komplexitätsbeherrschung ist die Bildung von Hierarchien.

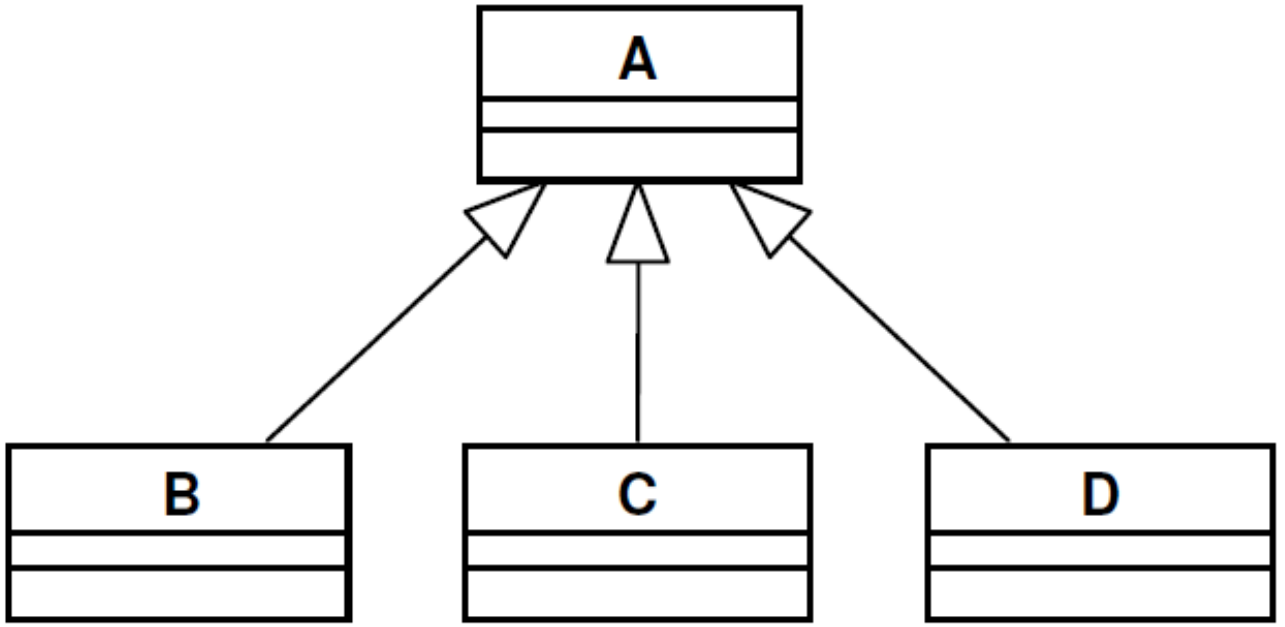


Bild 1-1 Vererbungshierarchie

Vererbungshierarchie

Typisch für solche Vererbungshierarchien sind die Klassifikationsschemen[1] der Tiere und Pflanzen in der Biologie (siehe Bild 1-2).

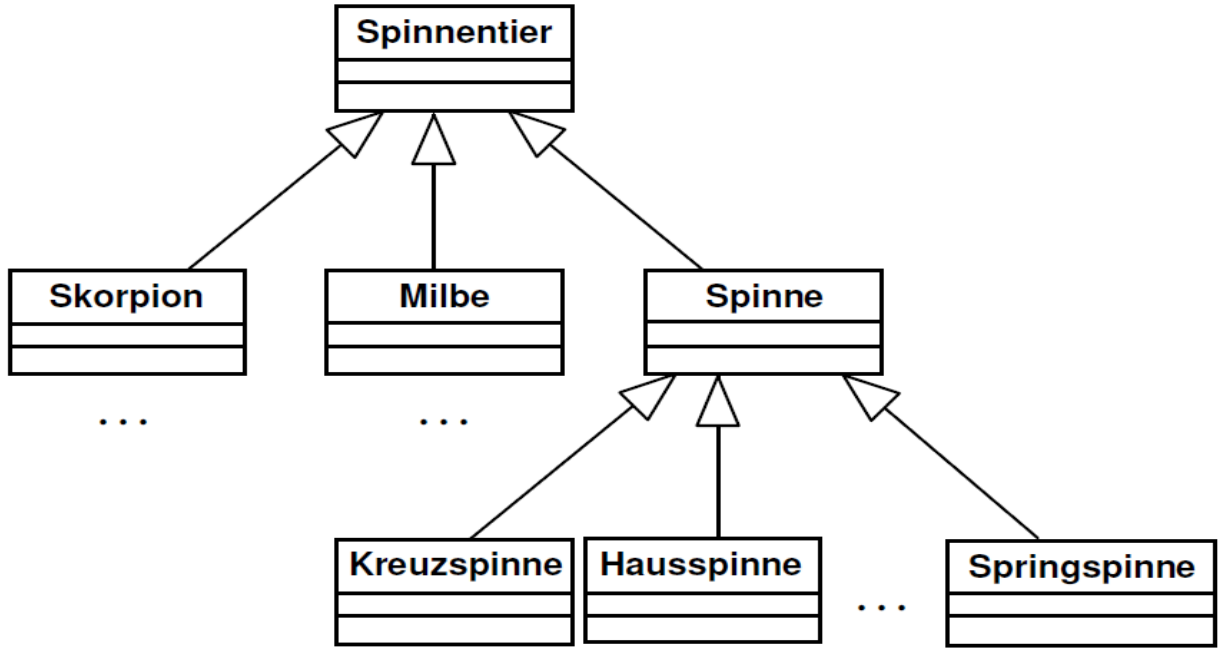


Bild 1-2 Klassifikation von Spinnentieren

[1] Während man in der Objektorientierung nur von Klassen redet, werden in der Biologie die Namen Stamm, Klasse, Unterklasse, Ordnung, Familie, Gattung, Art und Rasse verwendet.

Aggregationshierarchie

Komposition und Aggregation unterscheiden sich bezüglich der **Lebensdauer** des zusammengesetzten Objektes und seiner Komponenten.

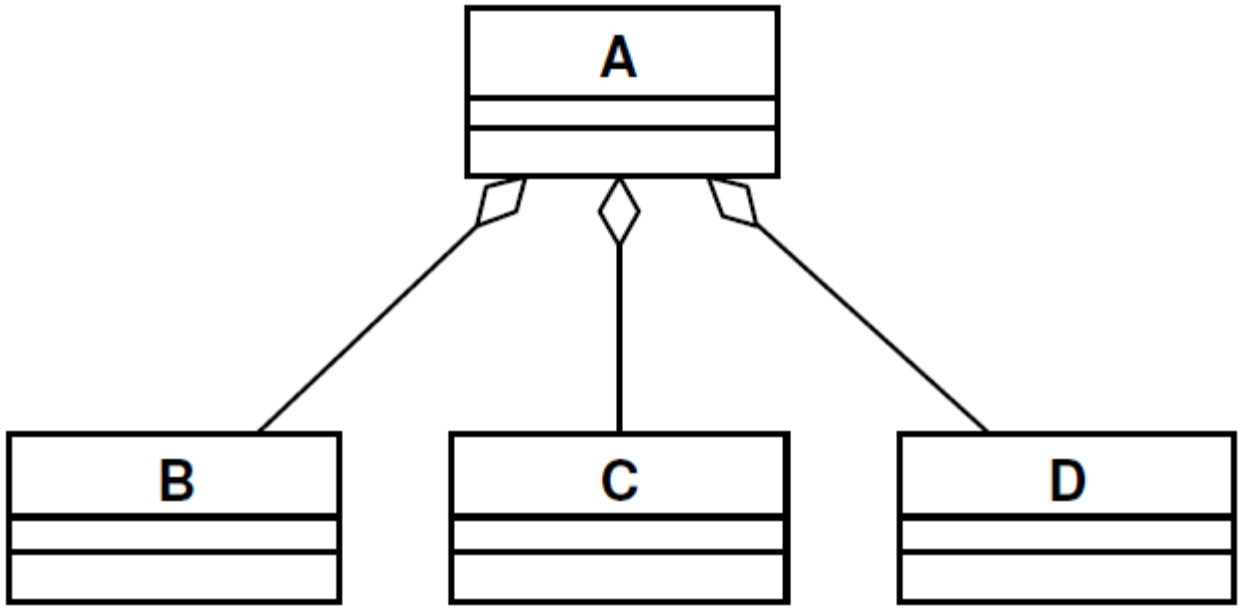


Bild 1-3 Aggregationshierarchie

Agenda

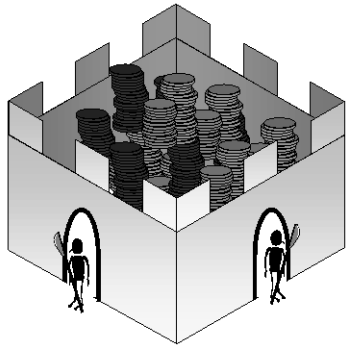
MHP
A PORSCHE COMPANY

1. Prinzipien der SW-Technik
   - Prinzip der Abstraktion
   - Prinzip der Bindung und Kopplung
   - Prinzip der Hierarchisierung
   - **Geheimnisprinzip**
   - (Softwareentwicklungs-) Werkzeuge

Geheimnisprinzip im Alltag und in der SW-Technik

Geheimnisse im Alltag

- zum Überraschen.
- zum Verbergen und Vertuschen (wenn was schief gelaufen ist).
- zum Informationen vorenthalten (Macht ausüben).



Geheimnisprinzip in der SW-Technik

- nur die externen Schnittstellen sind sichtbar.
- Interna werden vollständig verborgen.

Warum ist es so wichtig, die Interna zu verbergen?

Geheimnisprinzip (Information Hiding)

- Geheimnisprinzip ist eine Verschärfung des Prinzips der Modularisierung.
- Anwendung des Geheimnisprinzips auf ein Modul bedeutet, dass nur die Schnittstelle des Moduls nach außen sichtbar ist.

- Vorteile:
  - Die Verwendung eines Moduls wird **zuverlässiger**, da nur über die definierten Schnittstellen kommuniziert werden kann.
  - Der Anwender eines Moduls wird nicht mit unnötigen Informationen belastet.
  - Die **Datenkonsistenz** innerhalb des Moduls kann besser sichergestellt werden, da direkte, unkontrollierbare Manipulationen nicht möglich sind.
  - **Fehlersuche** wird **erheblich vereinfacht**, da alle Methoden, welche die Daten manipulieren, im Modul implementiert sind.

Änderungen im Inneren eines Moduls haben keine Auswirkungen auf andere Module, so lange die **Schnittstellen** stabil sind.

Geheimnisprinzip in der Objektorientierung

MHP
A PORSCHE COMPANY

- Verstecken von Daten und der Implementierung von zugehörigen Methoden in einer Kapsel (Objekt/Klasse).
- Kommunikation mit der Außenwelt nur über wohldefinierte Schnittstellenmethoden.

- Vorteil für Entwickler:
  - Bei der Implementierung der Algorithmen innerhalb der Methoden bestehen hohe Freiheitsgrade.
  - Innerer Aufbau kann optimiert werden, ohne dass sich an den Schnittstellen etwas ändert.

- Vorteil für Benutzer:
  - Muss sich nicht um die internen Details kümmern und muss diese auch nicht verstehen.
  - Kann immer ohne Änderung die neueste Version verwenden, so lange die Schnittstellen identisch bleiben.

Wie unterstützen Objektorientierte Programmiersprachen das Geheimnisprinzip?

Agenda

MHP
A PORSCHE COMPANY

1. Prinzipien der SW-Technik
   - Prinzip der Abstraktion
   - Prinzip der Bindung und Kopplung
   - Prinzip der Hierarchisierung
   - Geheimnisprinzip
   - **(Softwareentwicklungs-) Werkzeuge**

(Softwareentwicklungs-)Werkzeuge

Was ist ein Werkzeug in der Softwaretechnik?

Welche Werkzeuge für welchen Zweck kennen Sie?

- sollen Effizienz und Effektivität von Entwicklern erhöhen
  - **Effektiv:** am Ergebnis/Ziel orientiert**.**
  - **Effizient**: ein Ergebnis/Ziel mit möglichst geringem Mitteleinsatz zu erreichen.
  - Beispiel: Baugrube ausheben.
    Effektiv aber nicht effizient: Baugrube mit der Schaufel ausheben.
    Effektiv und effizient: Baugrube mit dem Bagger ausheben.

- (benötigtes) Hilfsmittel zur SW-Entwicklung
  - Das beste Werkzeug bringt ihnen nichts, wenn Sie keinen talentierten Handwerker haben, der weiß, wie man das Werkzeug richtig einsetzt.
  - Ohne rudimentäre Werkzeuge kann auch der beste Handwerker kein Haus bauen.

(Softwareentwicklungs-)Werkzeuge

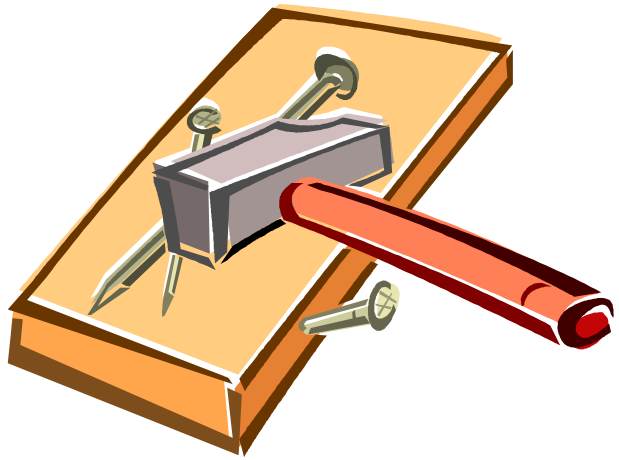
erwarteter Nutzen nur dann, wenn

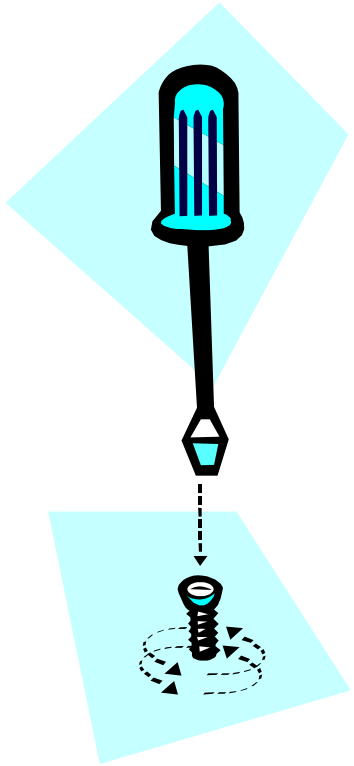
- Anwender die erforderliche Eignung hat:
  Wenn der Arbeiter nur eine Schaufel, aber keinen Bagger bedienen kann, nützt ihm der Bagger nichts!
- das Werkzeug für die Aufgabe geeignet ist:
  Wenn der Handwerker einen Balken absägen will, nützt ihm der Hammer nichts!
- die eingesetzte Methode zum Werkzeug passt:
  Beim Einsatz einer objektorientierten Methode braucht man ein UML-Werkzeug.

Ein Werkzeug muss Teil einer Methode sein, die beschreibt, wann und wie man das Werkzeug sinnvoll einsetzen kann.

Motivation

Aus Erfahrung weiß man, …





… das Werkzeug muss zur Aufgabe passen!

Für die SW-Entwicklung bedeutet dies:

Ist eine geeignete Methode für ein Projekt gefunden, die von den Projektmitarbeitern verstanden und akzeptiert wird,

so können die Arbeitsschritte der Methode durch den Einsatz eines geeigneten Werkzeuges unterstützt werden, um effizienter zu werden!

Was ist Enterprise Architect?

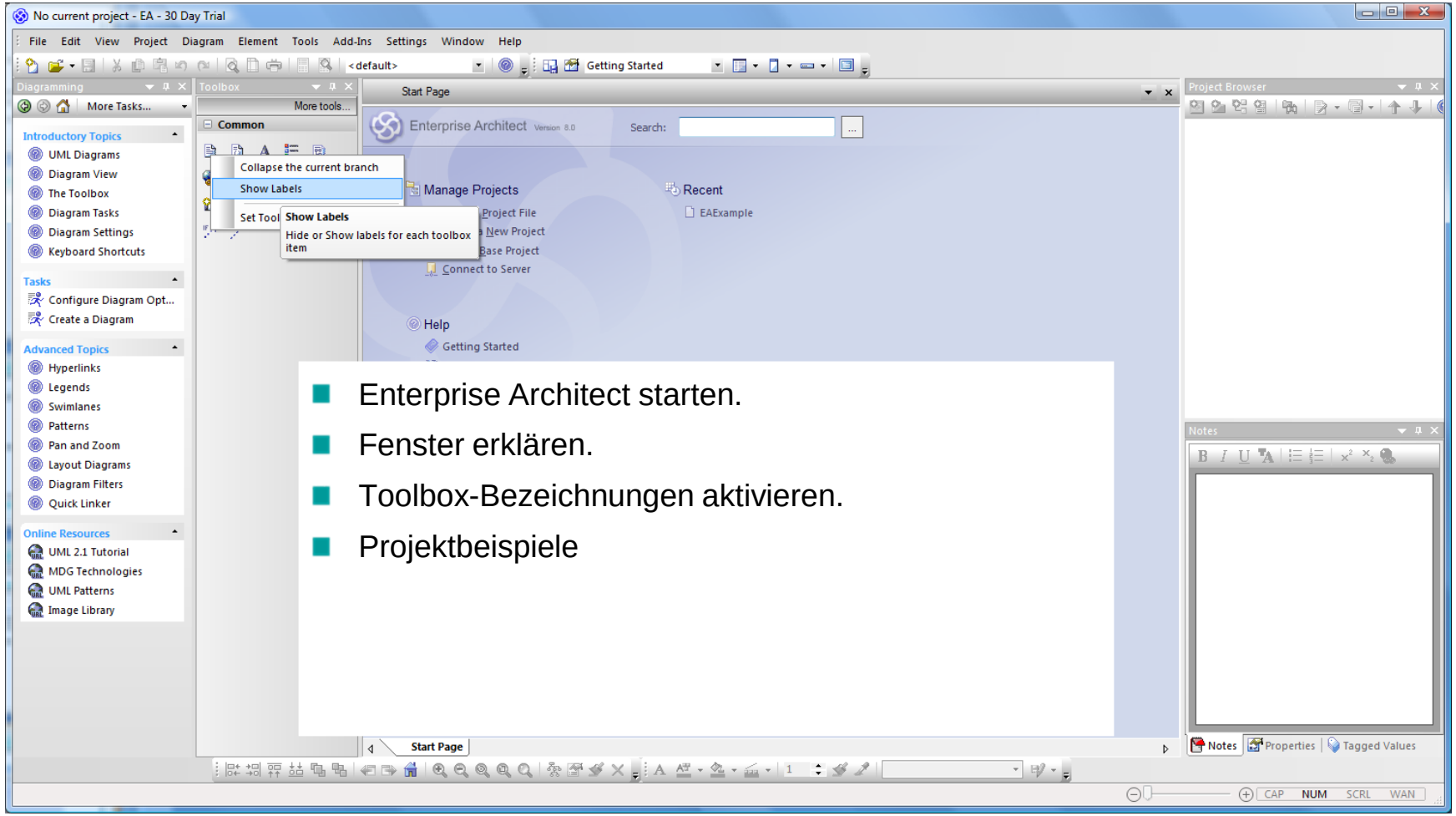
MHP
A PORSCHE COMPANY

- Objektorientiertes Software-Entwicklungswerkzeug für den gesamten Entwicklungsprozess.

- Enterprise Architect unterstützt bei folgenden Tätigkeiten:
  - Geschäftsprozessmodellierung,
  - Anforderungsmanagement,
  - Systemanalyse,
  - Systementwurf,
  - Test
  - und Änderungsmanagement.
- Unterstützung der Modellierung von Systemen mit der UML 2.3.
- Umfassende Informationen unter http://www.sparxsystems.de.

Online Demonstration

MHP
A PORSCHE COMPANY



Agenda

MHP
A PORSCHE COMPANY

1. Aufbau von Anwendungssystemen
2. Prinzipien der SW-Technik
3. **Methoden der SW Technik & UML**
   - Allgemeine Grundlagen UML
   - Anwendungsfall-/ Use Case Diagramme
   - Aktivitätsdiagramme
   - Objekt- und Paketdiagramme
   - Komponentendiagramme
   - Klassendiagramme
   - Sequenzdiagramme
   - Objektorientierte Methode

Motivation

Wir müssen das Rad nicht neu erfinden!

Aber: Wir müssen das passende Rad finden!

Wir müssen keine neuen Methoden erfinden!

Aber: Welche Methode in welcher Ausprägung passt zum Projekt und zu den Projekt

Top-Down-Methode

- Vom Abstrakten zum Konkreten, vom Allgemeinen zum Speziellen.
- Beispiele für die Anwendung:
  - Schichten • Pakete • Komponenten • Klassen und ihre Beziehungen.
  - Modellierung von Zuständen in Zustandsautomaten: Zustand Ein/Aus eines Gerätes (Oberzustand), danach Modellierung der Unterzustände.
  - Identifizieren von Klassen und danach Festlegen der Attribute und Operationen.
  - Spezialisierung: Vom abstrakten Typ zu den konkreten Typen/Klassen.
  - Planung von Phasen und später Detailplanung der Aktivitäten.
- Vorteile und Nachteile
  + Konzentration auf das Wesentliche möglich.
  + Strukturelle Zusammenhänge werden leichter erkannt.
  - Es wird ein hohes Abstraktionsvermögen benötigt.

Wo arbeiten Sie in Ihren Projekten nach der Top-Down-Methode?

Bottom-Up-Methode

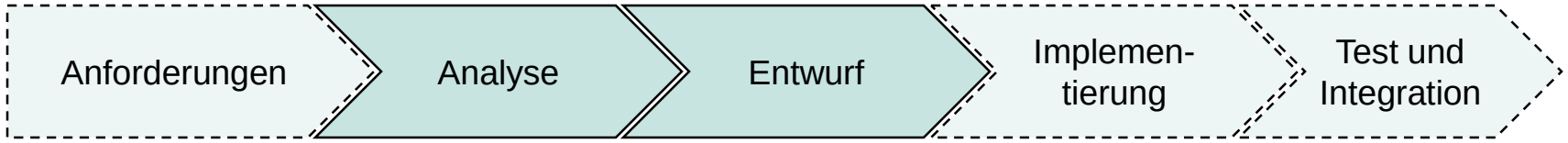
MHP
A PORSCHE COMPANY

- Vom Konkreten zum Abstrakten, vom Speziellen zum Allgemeinen.
- Beispiele:
  - Objektorientierte Modellierung: Identifikation von Klassen und ihren Beziehungen • Zusammenfassung zu Komponenten/Paketen • Bildung von Schichten.
  - Modellierung von Zuständen in Zustandsautomaten: Zuerst Modellierung der Unterzustände danach Zusammenfassen zu Oberzuständen.
  - Identifizieren von Attributen und Operationen danach Identifikation der Klassen.
  - Generalisierung: Vom konkreten Typ/Klasse zum abstrakten Typ.
  - Detailplanung für einzelne Aktivitäten, danach Zusammenfassen zu den Phasen.
- Vorteile und Nachteile
  - \+ Es ist eine konkrete Ausgangsbasis vorhanden.
  - \+ Bei Altsystemen ist dies oftmals die einzige Möglichkeit.
  - \- Übergeordnete Strukturen/Abstraktionen sind schwierig zu erkennen.

Wo arbeiten Sie in Ihren Projekten nach der Bottom-Up-Methode?

Methode

Methoden dienen als Vorschrift für die Durchführung einer oder mehrerer Aktivitäten/Phasen eines Vorgehensmodells.



Konzept:	**Analyse-/Entwurfsmethode:**
datenorientiert	Datenmodellierung mit ERM (Entity Relationship Model)
funktionsorientiert	Strukturierte Analyse (SA) / Strukturiertes Design (SD) — Strukturierte Modellierung
objektorientiert	Objektorientierte Analyse (OOA) / Objektorientiertes Design (OOD) — Objektorientierte Modellierung (OOM)

Auch **klassische Methoden** genannt.

Methoden dienen der **Modellierung** nach einem bestimmten **Konzept**.

Modelle in den unterschiedlichen Methoden

Methoden setzen Modelle ein. Ein Modell ist ein Abbild der Realität. Ein Modell besteht aus mehreren Sichten, die gekoppelt das System beschreiben.

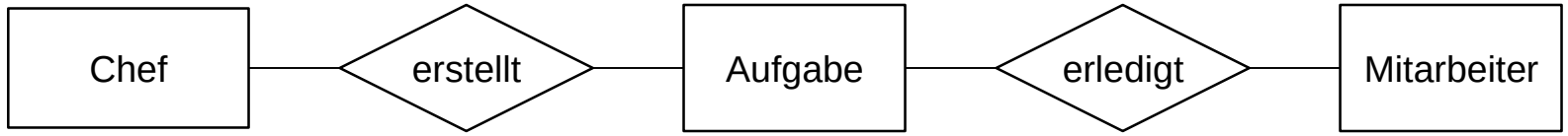
Konzept:	**Analyse-/Entwurfsmethode:**	**Modelle und Sichten:**
datenorientiert	Datenmodellierung mit ERM (Entity Relationship Model)	- ERM, - relationales Datenbankmodell, …
funktionsorientiert	Strukturierte Analyse (SA) / Strukturiertes Design (SD) — Strukturierte Modellierung	-DFD: Datenflussdiagramm - PSPECs: Prozessspezifikationen - CFD: Kontrollflussdiagramm - CSPEC: Kontrollspezifikation - Entscheidungstabelle - Aufrufhierarchie, …
objektorientiert	Objektorientierte Analyse (OOA) / Objektorientiertes Design (OOD) — Objektorientierte Modellierung (OOM)	Unified Modelling Language (UML) - Klassendiagramm - Anwendungsfalldiagramm - Aktivitätsdiagramm - Kommunikationsdiagramm , …

Datenmodellierung mit ERM

MHP
A PORSCHE COMPANY

- ERM = ältestes und allgemeinstes Modell zur strukturierten Modellierung von Daten.
  - definiert 1970 – 1976 von Peter Chen:
  - taiwanischer Informatiker,
  - seit 1983 Professor der Informatik an der Louisiana State University,
  - zählt zu den Pionieren der Informatik.

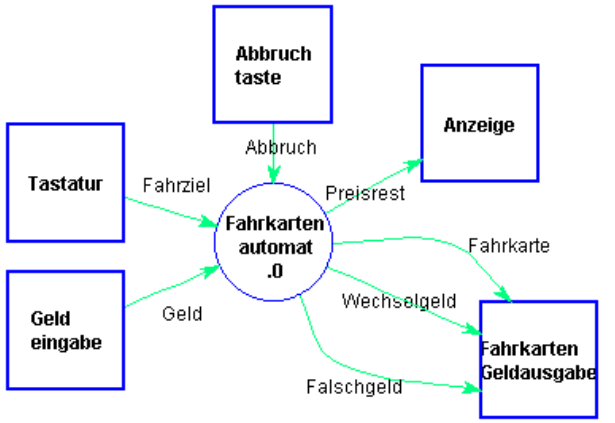
- Beispiel Chen-Notation:

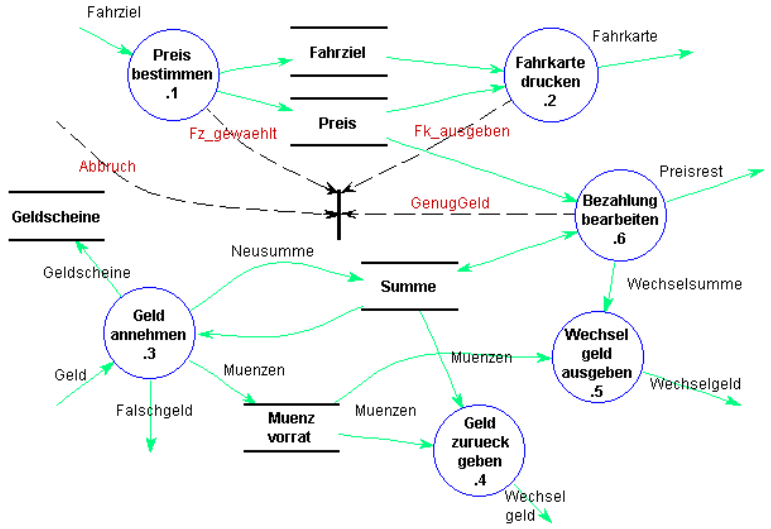


- Bedeutung für nachfolgende Methoden:
  - ERM-Methode wurde nachträglich in die Strukturierte Analyse aufgenommen.
  - Objektorientierte Theorie von Rumbaugh basiert auf ERM.
  - Eine Klasse entspricht einem Entitäts-Typ.
  - ER-Diagramme und UML-Klassendiagramme sind in der Struktur einander ähnlich.

Strukturierte Analyse (SA)

- Strukturierte Analyse (SA) ist eine Top-Down-Methode für die Analyse-Phase einer Software-Entwicklung.
- hauptsächlich von Tom DeMarco von 1970 – 1977 entwickelt:
  - 1940 in Pennsylvania, USA geboren,
  - viele Jahre in der Software-Entwicklung tätig,
  - viele bekannte Bücher (auch „Der Termin“) geschrieben.
- Hierarchische Zerlegung des Systems in immer einfachere Funktionen/Prozesse.
- Gleichzeitig wird eine Datenflussmodellierung durchgeführt.

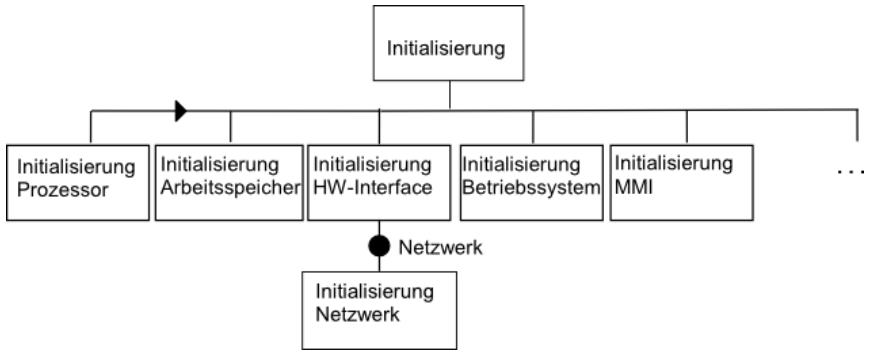


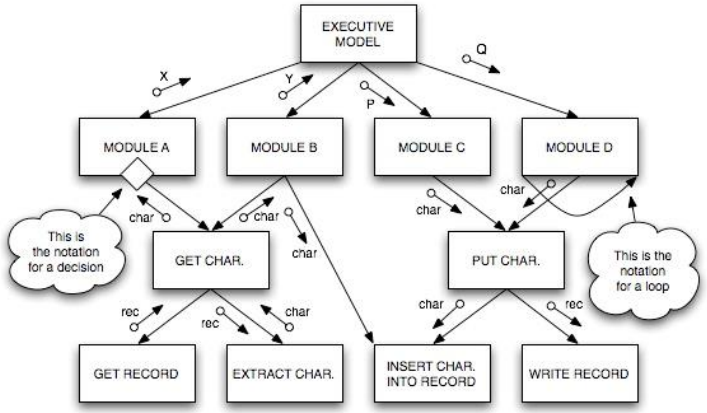


Strukturiertes Design (SD)

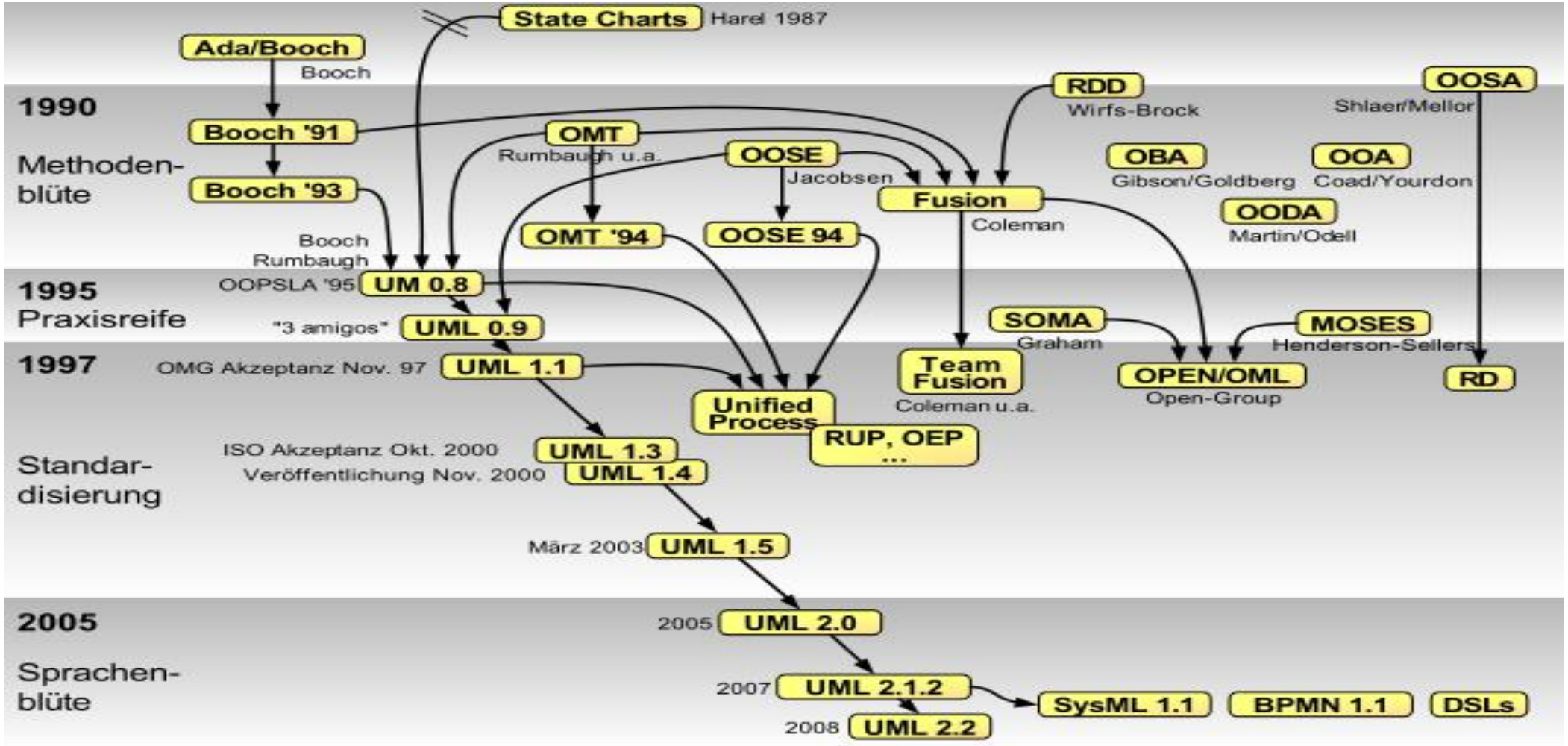
- Schlägt eine Brücke zwischen der technologieneutralen Strukturierten Analyse und der eigentlichen Implementierung.
- entwickelt von
  - Edward Yourdon (geb. 1944)
  - und Larry Constantine (geb. 1943)
  - in den späten 1970ern.

- beschreibt – neben der reinen Funktionshierarchie – den Kontrollfluss in den Kontrollflussdiagrammen sowie die Wechselwirkungen zu übergeordneten Modulen in Structure-Chart-Diagrammen.





Entwicklung von Methoden für die Objektorientierung



Agenda

MHP
A PORSCHE COMPANY

1. Aufbau von Anwendungssystemen
2. **Methoden der SW Technik & UML**
   - **Allgemeine Grundlagen UML**
   - Anwendungsfall-/ Use Case Diagramme
   - Aktivitätsdiagramme
   - Objekt- und Paketdiagramme
   - Komponentendiagramme
   - Klassendiagramme
   - Sequenzdiagramme
   - Objektorientierte Methode

Was ist UML?

MHP
A PORSCHE COMPANY

Definition:

UML ist eine standardisierte Notationssprache zur Beschreibung der objektorientierten Modellierung von Systemen.

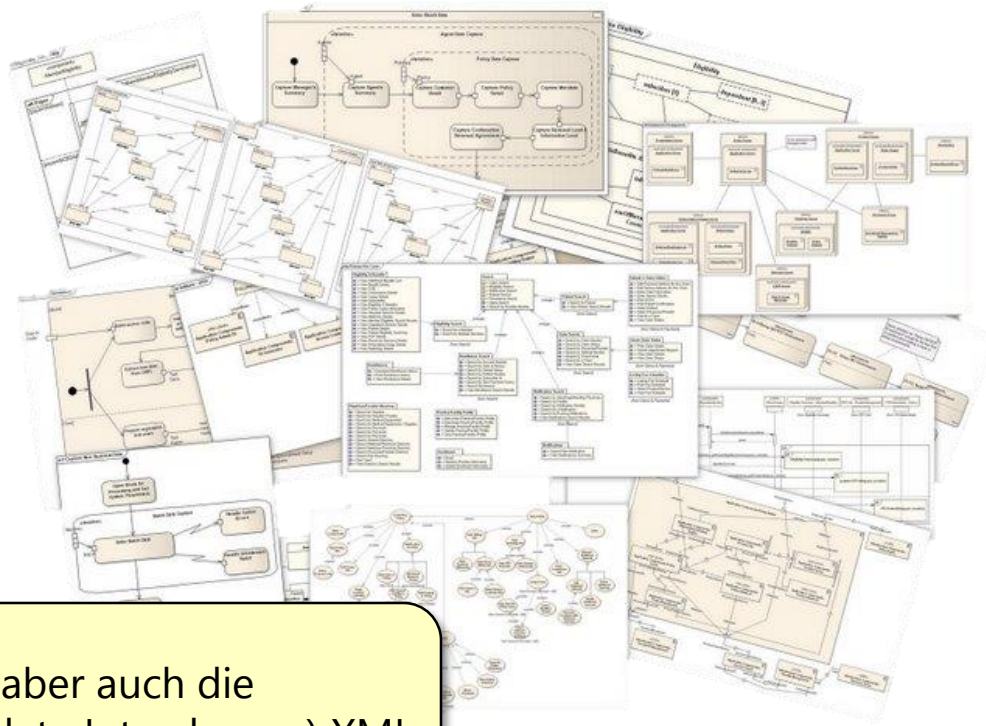
Vorsicht:

Das objektorientierte Konzept und die objektorientierte Methodik beruhen nicht auf UML, sondern können mit Hilfe von UML notiert werden.

Ziel der UML ist es,
die **Analyse** und den **Entwurf** von komplexen, **objektorientierten Software-Systemen standardisiert** und **leicht verständlich zu dokumentieren**.

Was ist die Unified Modeling Language (UML)?

- Sprache und Notation zur Spezifikation, Konstruktion, Visualisierung und Dokumentation von Modellen für Softwaresysteme
- von der *Object Management Group* (OMG) entwickelt und ist sowohl von ihr als auch von der ISO (ISO/IEC 19505 für Version 2.4.1[2]) standardisiert
- Aktuelle Version: UML 2.5
- Offizielle Seite: www.uml.org



“Sichtbar” von **UML** ist die **grafische Notation**, es gibt aber auch die Möglichkeit UML rein textuell zu formulieren (XML Metadata Interchange) XMI.

Wie entstand die UML?

MHP
A PORSCHE COMPANY

- UML entstand in den frühen 90er-Jahren mit dem Ziel die Methodenkriege, wie man objektorientiert modelliert, zu beenden (es gab über 50 Methoden).

- Die Väter der UML – Grady Booch, James Rumbaugh und Ivar Jacobson (auch als die 3 Amigos bekannt) –, die alle nach und nach bei der Firma Rational beschäftigt wurden, arbeiteten gemeinsam daran, eine gemeinsame Methode – die Unified Method – zu entwickeln.

- Da keine Einigung auf eine gemeinsame Methode möglich war, wurde – als bescheideneres, gemeinsames Ziel – eine grafische Modellierungssprache – die Unified Modelling Language – geboren.

Wozu UML?

- Kommunikation – gemeinsame Sprache sprechen
  - zum Auftraggeber
  - in Projektteams (anhand standardisierter Diagramme)
- Abstraktion
  - Code ist zwar präzise, aber zu detailliert
  - UML bietet Abstraktionen und unterschiedliche Sichten auf das zu erstellende System
- Durchgängigkeit
  - UML ermöglicht eine durchgängige und vollständige Systemdokumentation
  - bis hin zur Codegenerierung beim Einsatz von CASE-Tools
- Standard
  - zur Modellierung und Dokumentation objektorientierter Systeme
  - für die Entwicklung von objektorientierten CASE-Tools

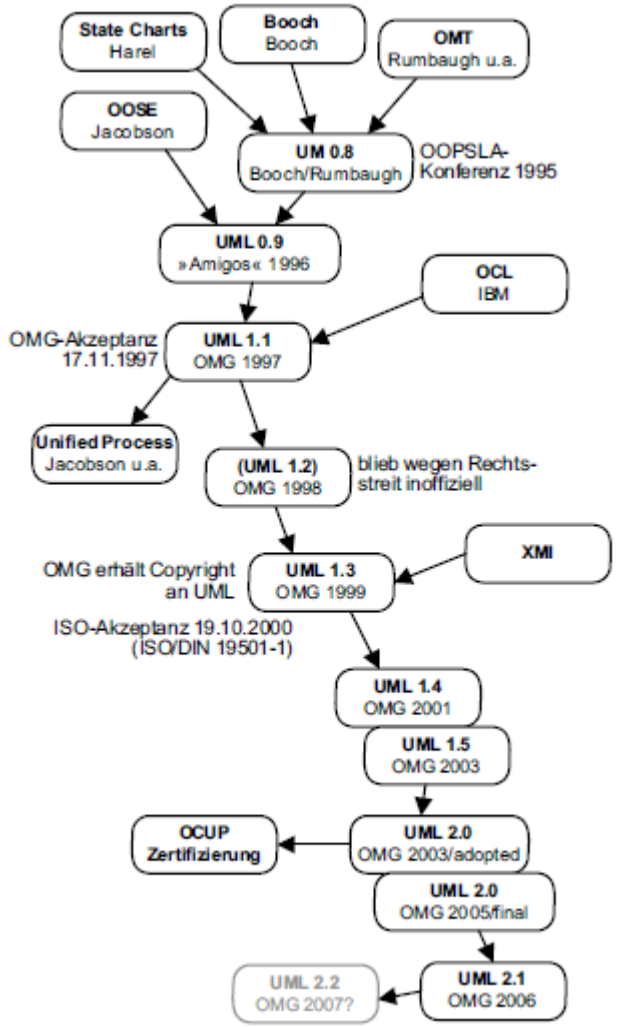


Abb. 1-1: Historische Entwicklung der UML

Standardisierung der UML

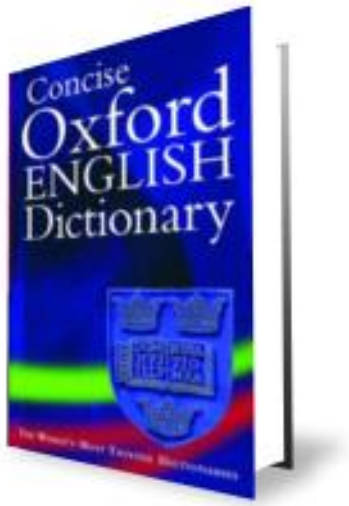
- 1997: Die OMG (Object Management Group) nimmt UML 1.1 als Standard an.
- 1999: Veröffentlichung der Version 1.3.
- 2003: Veröffentlichung der noch heute für die Praxis relevante Version 1.5.

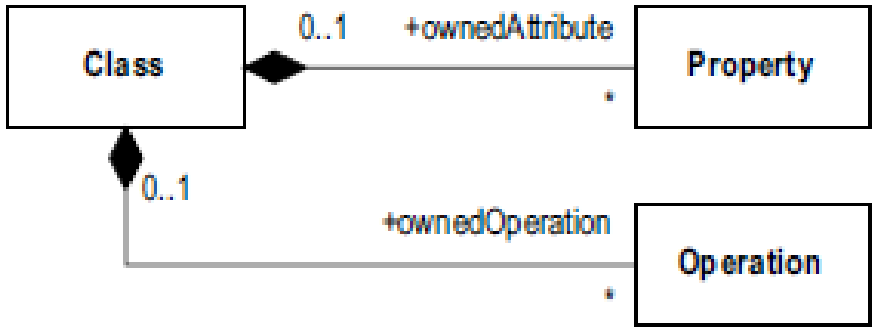
Standardisierungsarbeiten an der UML 2, erhebliche Erweiterungen z. B.:
architektonische Änderungen (Superstructure / Infrastructure)
standardisiertes Format zum Austausch von Modellen und Diagrammen zwischen Werkzeugen

- 2010: Veröffentlichung der Version 2.3

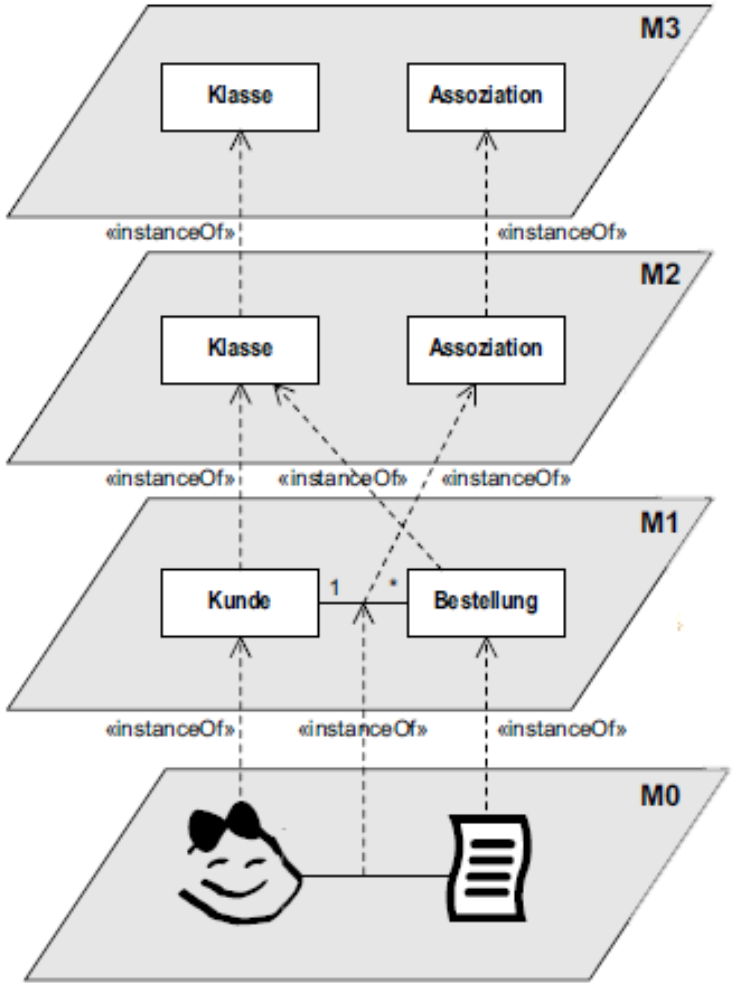
Die UML-Spezifikation sowie weitere Links und Informationen finden Sie unter:
http://www.uml.org

Das Metamodell der UML





Das Metamodell der UML



Meta-Metamodell, MOF
Modellierung z.B. des UML-Metamodells

UML-Metamodell,
gewöhnlich verfügbare UML-Modellierungselemente

UML-Modell,
gewöhnliche UML-Modellierungsebene

Laufzeitmodell
Das Objekt selbst

Datentypen (1/2)

MHP
A PORSCHE COMPANY

UML unterscheidet in:

- Einfache Datentypen (Data Type)
  - wie beispielsweise Geld, Bankverbindung etc.
  - Einfache Datentypen sind ein Oberbegriff für primitive Datentypen und Aufzählungstypen. Ein einfacher Datentyp ist ein Typ, dessen Werte keine Identität besitzen, d.h., zwei Instanzen eines Datentyps mit denselben Attributwerten können nicht voneinander unterschieden werden. Das ist ein wesentlicher Unterschied gegenüber Klassen.

- Aufzählungstypen (*Enumeration*)
  - wie beispielsweise Familienstand, Anrede etc.
  - Aufzählungstypen sind einfache Datentypen, deren Werte aus einer begrenzte Menge von Aufzählungsliteralen (*EnumerationLiteral*) stammen

Datentypen (2/2)

- Primitive Datentypen (PrimitiveType)
- Ein primitiver Datentyp ist ein einfacher Datentyp ohne Strukturen. Die zum Typ gehörende Algebra ist außerhalb der UML beschrieben, z.B. die Addition ganzer Zahlen wird nicht als Operation eines primitiven Datentyps modelliert.
- Die UML definiert selbst folgende primitive Datentypen vor:
  - *Integer*

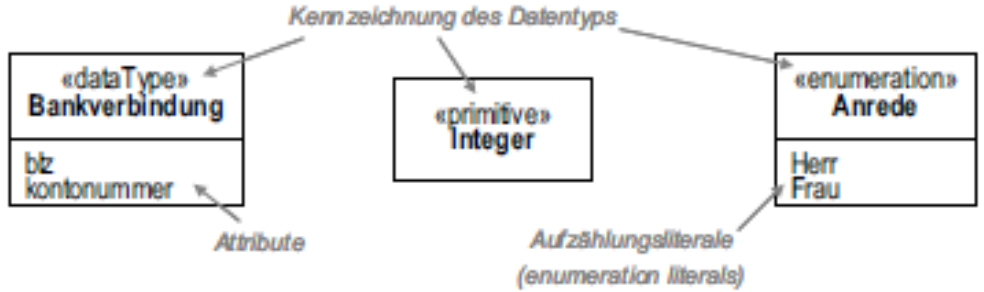
    (Unendliche) Menge ganzer Zahlen: (..., -2, -1, 0, 1, 2, ...)
  - *Boolean*

    true, false
  - *String*

    Beliebige Zeichenkette, auch Umlaute u. Ä. sind zulässig.
  - *UnlimitedNatural*

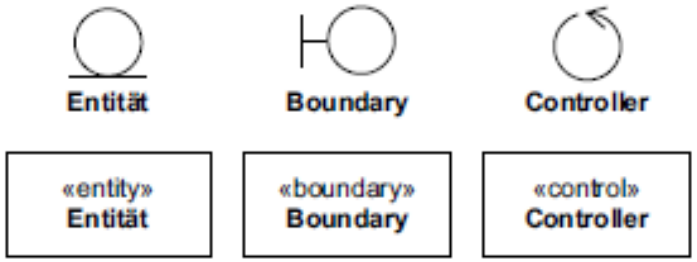
    (Unendliche) Menge natürlicher Zahlen: (0, 1, 2, ...). Das Symbol für unendlich ist „*“. Dieser Datentyp wird im Metamodell beispielsweise für die Definition von Multiplizitäten verwendet.

Zusammenfassend:



Stereotypen

- Formale Erweiterungen vorhandener Modellelemente des UML-Metamodells.
- Entsprechend der mit der Erweiterung definierten Semantik wird das Modellierungselement, auf das es angewendet wird, direkt semantisch beeinflusst.
- Klassifizieren die möglichen Verwendungen eines Modellelementes. Dabei handelt es sich nicht um die Modellierung von Metaklassen, vielmehr werden einem oder mehreren Modellelementen bestimmte gemeinsame Eigenheiten zugeschrieben.
- Visuelle und textuelle Stereotypen

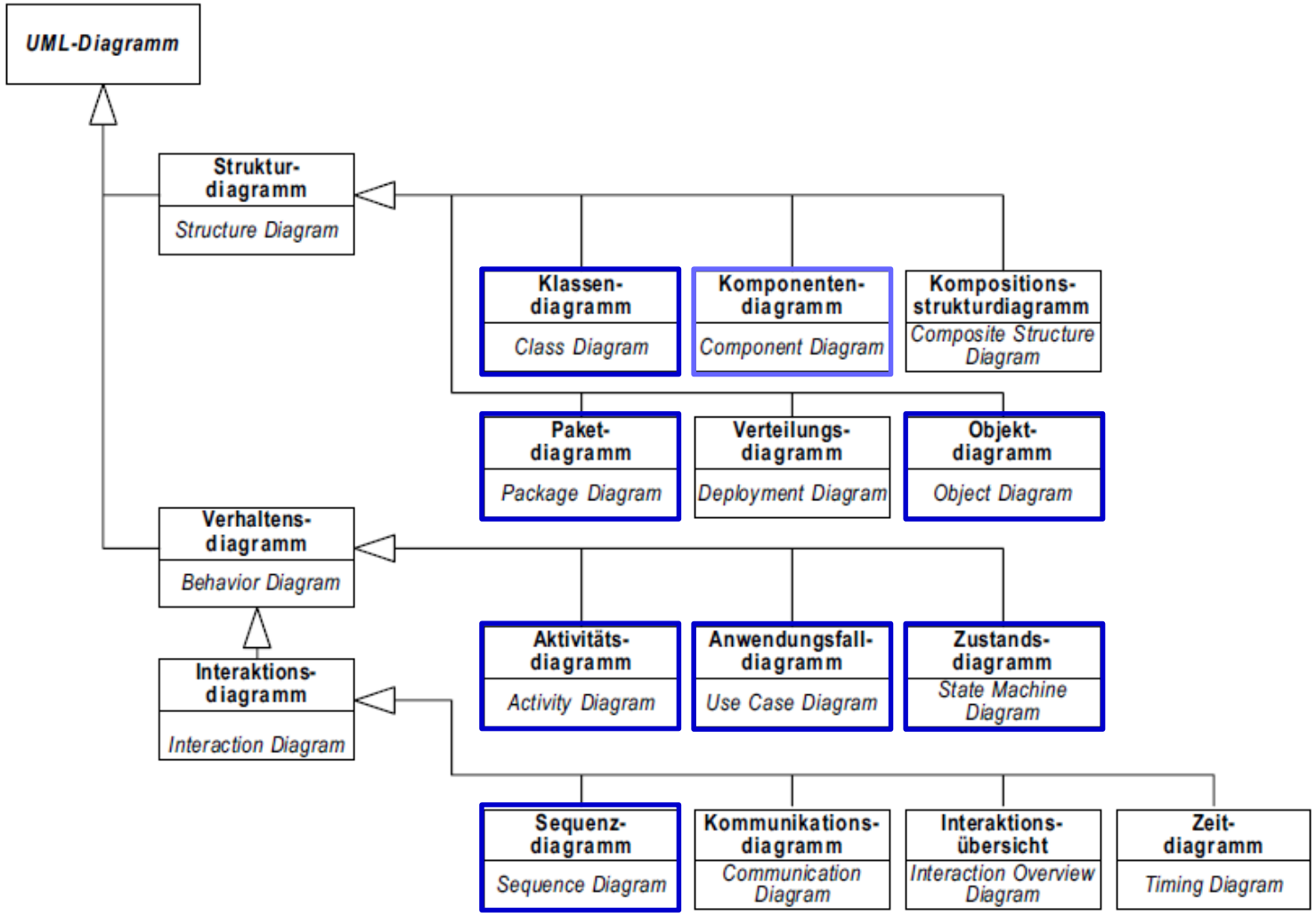


Standard-Stereotypen der UML

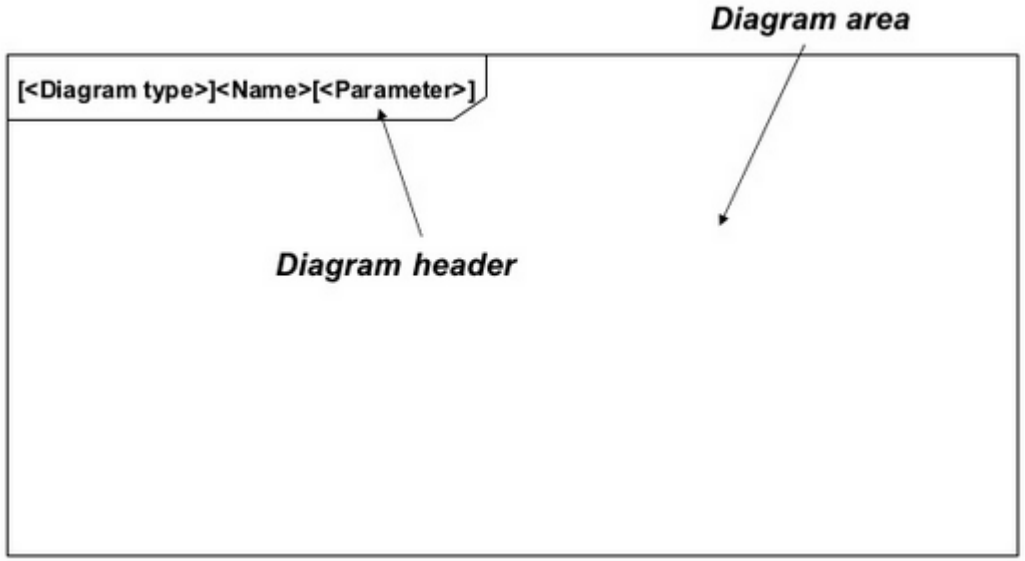
Stereotyp	UML-Element	Beschreibung
«call»	Abhängigkeit (Usage)	Aufrufabhängigkeit zwischen Operationen oder Klassen
«create»	Abhängigkeit (Usage)	Quellelement erstellt Instanzen des Zielelementes.
«instantiate»	Abhängigkeit (Usage)	Quellelement erstellt Instanzen des Zielelementes. (Achtung: Diese Beschreibung ist identisch mit der von «create».)
«responsibility»	Abhängigkeit (Usage)	Quellelement ist verantwortlich für das Zielelement.
«send»	Abhängigkeit (Usage)	Quellelement ist eine Operation, Zielelement ist ein Signal, welches von der Operation gesendet wird.
«derive»	Abstraktion	Quellelement kann bspw. durch eine Berechnung vom Zielelement abgeleitet werden.
«refine»	Abstraktion	Verfeinerungsbeziehung, z.B. zwischen einem Design- und zugehörigem Analyseelement.
«trace»	Abstraktion	zur Verfolgbarkeit von Anforderungen
«script»	Artefakt	Skriptdatei (ausführbar auf einem Computer)
«auxiliary»	Klasse	Klassen, die andere Klassen («focus») unterstützen. Fokus-Klassen enthalten die primäre Logik.
«focus»	Klasse	siehe «auxiliary»
«implementation-Class»	Klasse	Implementierungsklasse speziell für eine Programmier-sprache, wobei ein Objekt nur zu einer Klasse gehören darf.
«metaclass»	Klasse	Klasse, deren Instanzen wiederum Klassen sind.
«type»	Klasse	Typen definieren eine Menge von Operationen und Attributen und sind in der Regel abstrakt.
«utility»	Klasse	Hilfsklassen sind Sammlungen von globalen Variablen und Funktionen, die zu einer Klasse zusammengefasst und dort als Klassenattribute/-operationen definiert sind.
«buildComponent»	Komponente	organisatorisch motivierte Komponente
«implement»	Komponente	Komponente, die keine Spezifikation enthält, sondern nur Implementierung.
«framework»	Paket	Paket, das Framework-Elemente enthält.
«modelLibrary»	Paket	Paket, das Modellelemente enthält, die in anderen Paketen wiederverwendet werden.
«create»	Verhaltenseigenschaft (BehavioralFeature)	Eigenschaft, die Instanzen der Klasse erzeugt, zu der sie gehört (z.B. Konstruktor).
«destroy»	Verhaltenseigenschaft (BehavioralFeature)	Eigenschaft, die Instanzen der Klasse zerstört, zu der sie gehört (z.B. Destruktor).

Abb. 2-7: Standardstereotypen der UML (Auswahl)

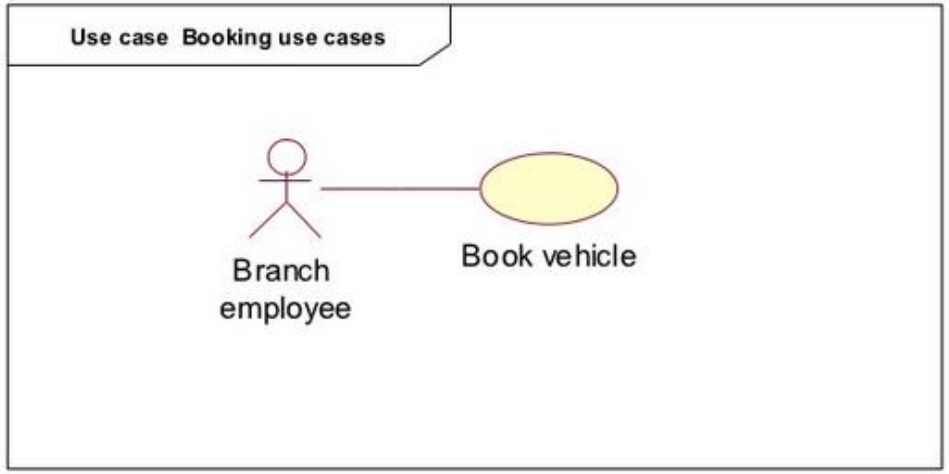
Diagramme im Überblick



Basis Notation von Diagrammen



Beispiel:



Zusammenfassung

MHP
A PORSCHE COMPANY

Checkliste allgemeine Grundlagen:

- Welche Datentypen unterscheidet die UML und wie heißen sie?
- Ist die Kennzeichnung der Datentypen (z.B. «primitive») ein Stereotyp?
- Was unterscheidet ein Datentyp von einer Klasse?

- Was ist ein Stereotyp? Definiert es ein neues Metamodellelement?
- Welche Standardstereotypen kennt die UML und was bedeuten sie?
- Gibt es Schlüsselwörter, die keine Stereotypen sind?

- Wie sieht die grundsätzliche grafische Darstellung eines Diagramms mit Diagrammkopf aus?
- Welche Informationen stehen im Diagrammkopf?

Agenda

MHP
A PORSCHE COMPANY

1. Aufbau von Anwendungssystemen
2. **Methoden der SW Technik & UML**
   - Allgemeine Grundlagen UML
   - **Anwendungsfall-/ Use Case Diagramme**
   - Aktivitätsdiagramme
   - Objekt- und Paketdiagramme
   - Komponentendiagramme
   - Klassendiagramme
   - Sequenzdiagramme
   - Objektorientierte Methode

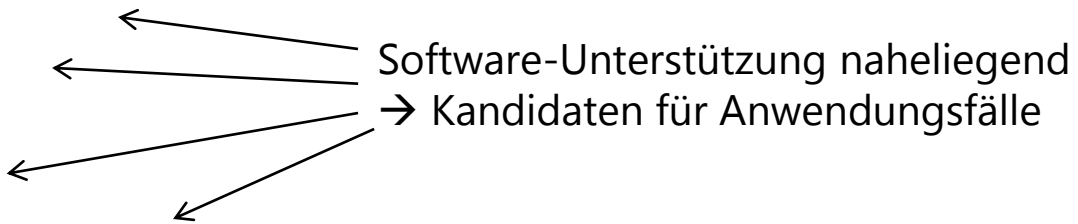
Beispielsystem Bibliothek: Identifikation Anwendungsfälle

MHP
A PORSCHE COMPANY

- Durch Beobachten der Abläufe werden die Geschäftsprozesse identifiziert.
- Für die spätere Programmierung relevant sind die so genannten Anwendungsfälle.

Die Anwendungsfälle werden gefunden, indem man die Geschäftsprozesse genau analysiert und festlegt, welche Teile durch das System automatisiert werden sollen.

- Der Geschäftsprozess „Buch ausleihen“, lässt sich wie folgt zerlegen:
  - Buchrecherche durchführen,
  - Buchverfügbarkeit prüfen,
  - Buch aus Regal holen,
  - Ausleiher identifizieren,
  - Buch für Ausleiher als entliehen buchen.



- Für die Anwendungsfälle ist zu prüfen, welche Objekte mit welchen Methoden hier unterstützen können.

Definition

- Ein *Anwendungsfall (UseCase)* spezifiziert eine Menge von Aktionen, die von einem System (subject) ausgeführt werden und zu einem Ergebnis führen, das üblicherweise von Bedeutung für einen Akteur oder Stakeholder ist.

- Der *Akteur (Actor)* ist definiert als eine Rolle, die sich außerhalb des Systems (subject) des zugehörigen Anwendungsfalles befindet und mit dem System im Kontext des Anwendungsfalles interagiert.

- Ein Anwendungsfalldiagramm beschreibt die Zusammenhänge zwischen einer Menge von Anwendungsfällen und den daran beteiligten Akteuren.

Beispielsystem Bibliothek: Methoden für die Klasse Buch

Wie kann ein Buch-Objekt die Anwendungsfälle unterstützen?

- Anwendungsfall „Buch-Recherche durchführen“
  - Ein einzelnes Buch-Objekt kann keine „Recherche durchführen“.
  - Ein einzelnes Buch-Objekt kann aber seine Daten für die Recherche bereitstellen:
  - getTitel()
  - getISBN()

- Anwendungsfall „Buchverfügbarkeit prüfen“
  - Ein Buch-Objekt muss wissen, wie viele Exemplare in der Bibliothek sind.
    - neues Datenfeld anzahlExemplare
  - Ein Buch-Objekt ermöglicht die Abfrage der vorhandenen Exemplare über eine Methode getAnzahlExemplare()
  - …

Notation und Semantik

- Akteure:

Abb. 2-118: Notation von Akteuren

- Use Case (Anwendungsfall):

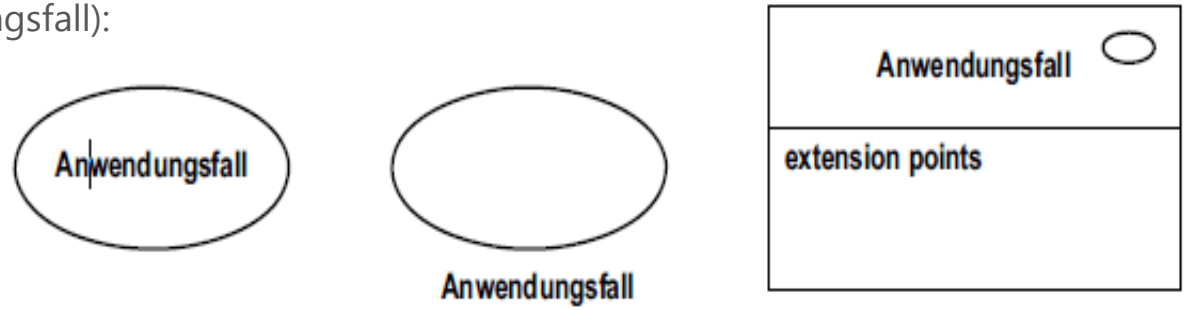
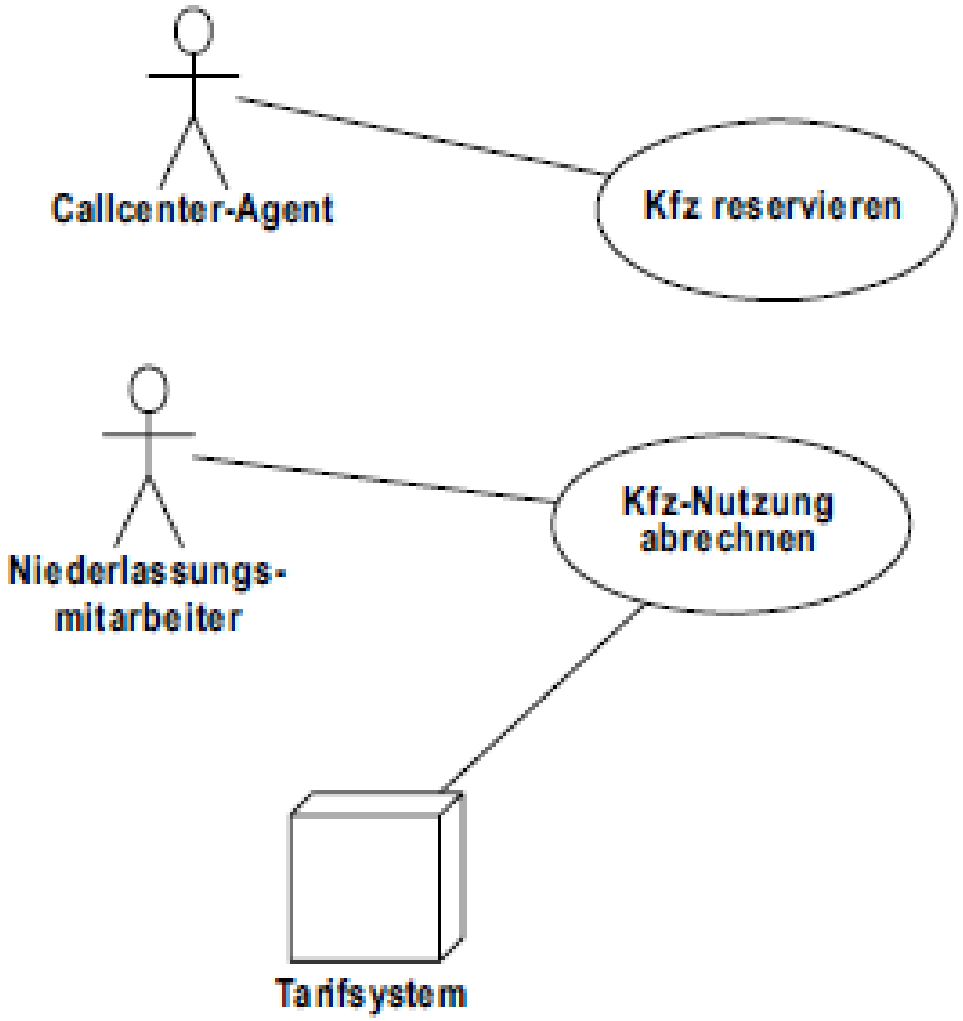


Abb. 2-119: Notation von Anwendungsfällen

Notation und Semantik

- Beispiel einer Anwendung:



Notation und Semantik – Classifier

- Ein Subsystem:

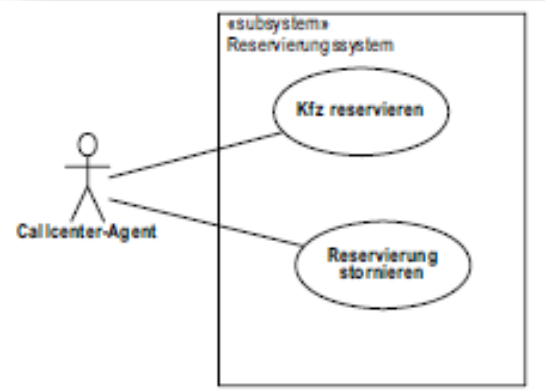


Abb. 2-120: Anwendungsfallkontext

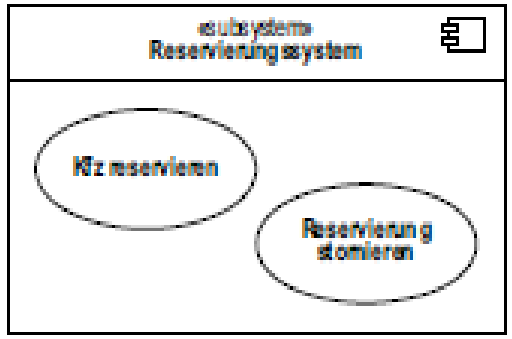


Abb. 2-121: Komponente mit zugehörigen Anwendungsfällen

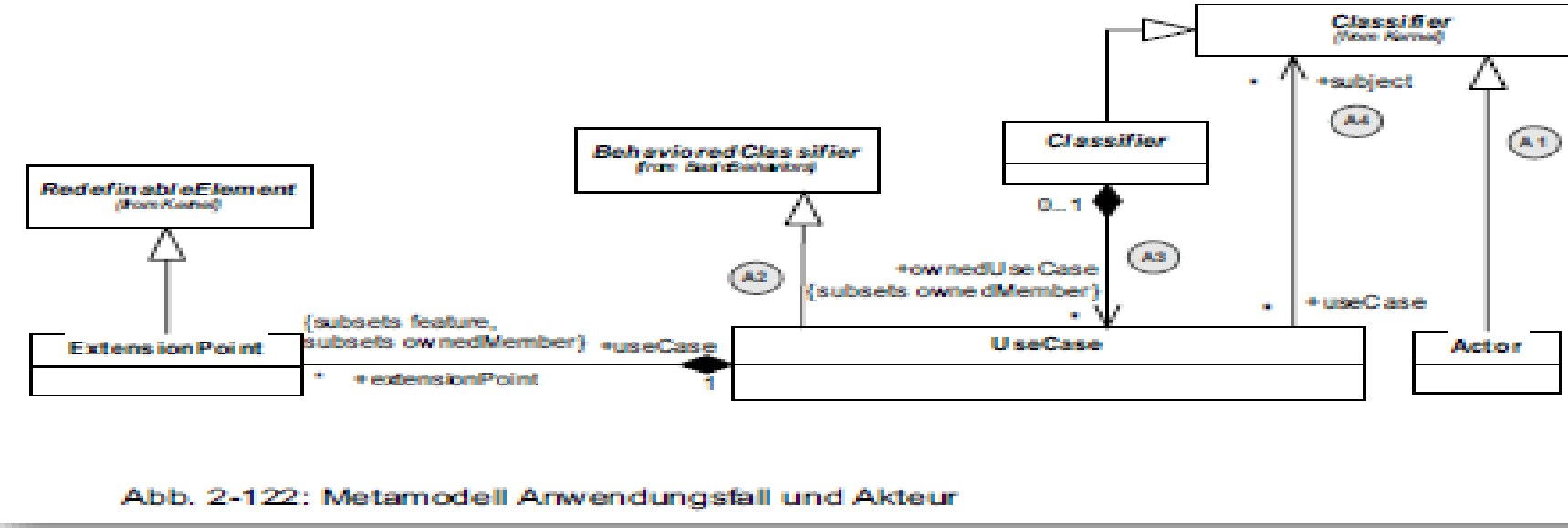


Abb. 2-122: Metamodell Anwendungsfall und Akteur

Anwendungsfallbeziehungen (Include, Extend)

- Mit einer Enthältbeziehung (Include) wird ein Anwendungsfall in einen anderen Anwendungsfall eingebunden und logischer Teil von diesem.

- Mit einer Erweiterungsbeziehung (Extend) hingegen lässt sich ausdrücken, dass ein Anwendungsfall unter bestimmten Umständen und an einer bestimmten Stelle (dem sog. Erweiterungspunkt, engl. ExtensionPoint) durch einen anderen erweitert wird.

Include

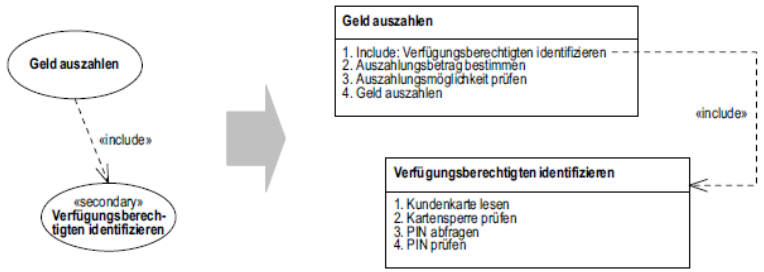


Abb. 2-123: Die Enthältbeziehung im Anwendungsfalldiagramm und ihre Bedeutung in der Anwendungsfallbeschreibung (Hinweis: rechte Seite ist kein UML)

Extend

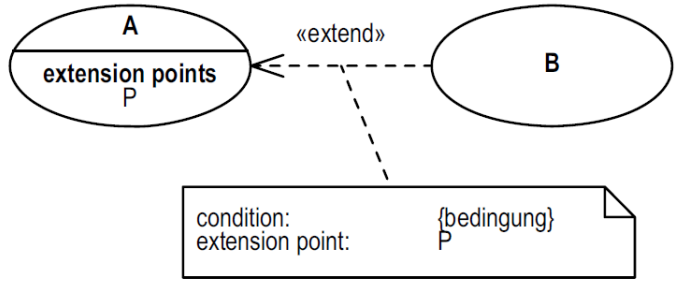
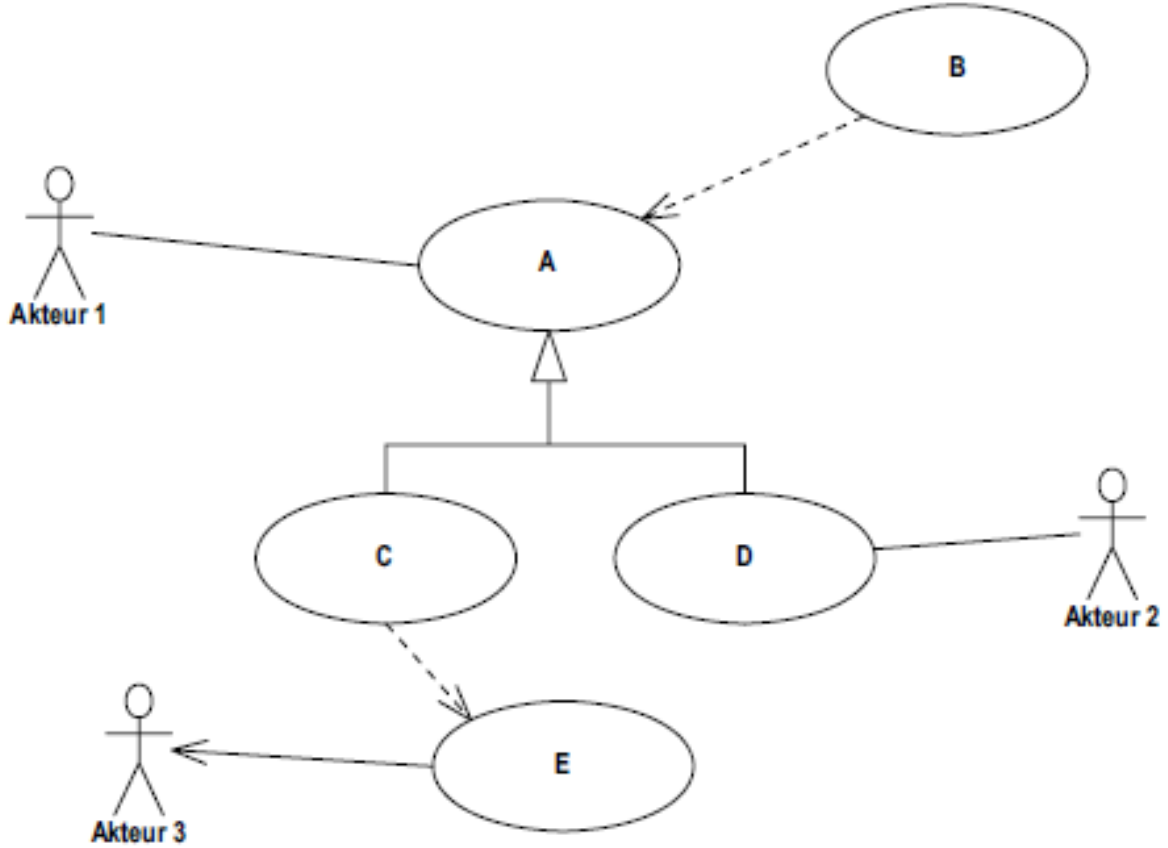


Abb. 2-124: Notation der Erweiterungsbeziehung

Übung 1

- Gegeben: Anwendungsfalldiagramm mit Anwendungsfällen, Akteuren, Generalisierungs-beziehungen und Enthält- und/oder Erweiterungsbeziehungen gezeigt, die ohne Schlüsselwort notiert sind

- Welche Aussagen lassen sich treffen?



Übung 2

MHP
A PORSCHE COMPANY

Kurzbeschreibung

Der Anwendungsfall „Benutzer verwalten“ (Akteur: WiBe-Beauftragter/Admin) beschreibt die Verwaltung von Benutzern des Systems. Er dient dabei der Gruppierung der speziellen Anwendungsfälle und beinhaltet die Anwendungsfälle:

- Kennwort ändern
- Benutzer anlegen,
- Benutzer löschen sowie
- Benutzer bearbeiten.

Das Kennwort ändern können des Weiteren folgende Akteure:

- Mitarbeiter
- Projektleiter
- Controller
- Katalog-Autor

>> Bitte erstellt das dazugehörige Use-Case-Diagramm

Zusammenfassung

MHP
A PORSCHE COMPANY

Checkliste Akteure und Anwendungsfälle:

- Was ist die Oberklasse eines Akteurs?
- Was ist die Oberklasse eines Anwendungsfalles?
- Wer kann der Besitzer eines Anwendungsfalles sein?
- Was ist das Subjekt eines Anwendungsfalles?
- Welche Notationsformen gibt es für einen Anwendungsfall?
- Welche Notationsformen gibt es für einen Akteur?
- Kann nur der menschliche Benutzer ein Akteur sein?

Checkliste Anwendungsfallbeziehungen

- In welche Richtung zeigt die Exclude-Beziehung?
- In welche Richtung zeigt die Include-Beziehung?
- Wie viele Erweiterungspunkte kann ein Anwendungsfall definieren?
- Muss eine Erweiterungsbeziehung eine Bedingung angeben?
- Muss der enthaltene Anwendungsfall einen Akteur haben?
- Muss der erweiternde Anwendungsfall einen Akteur haben?

Agenda

MHP
A PORSCHE COMPANY

1. Aufbau von Anwendungssystemen
2. **Methoden der SW Technik & UML**
   - Allgemeine Grundlagen UML
   - Anwendungsfall-/ Use Case Diagramme
   - **Aktivitätsdiagramme**
   - Objekt- und Paketdiagramme
   - Komponentendiagramme
   - Klassendiagramme
   - Sequenzdiagramme
   - Objektorientierte Methode

Definition

- Die *Aktivität (Activity)* beschreibt auf Basis von Kontroll- und Objektflussmodellen die Ablaufreihenfolge von Aktionen. Eine Aktivität enthält Kanten und Aktivitätsknoten, speziell Aktionen

- Eine *Aktion (Action)* ist im Metamodell eine abstrakte Klasse. Die konkreten Unterklassen werden auch in der UML spezifiziert. Sie sind in den so genannten Action Semantics beschrieben

- Mit Aktivitätsdiagrammen können Abläufe, wie sie beispielsweise in Anwendungsfällen (Use Cases) beschrieben sind, grafisch dargestellt werden. Mit natürlicher Sprache werden gewöhnlich nur sehr einfache Abläufe verständlich beschrieben, mit Aktivitätsdiagrammen hingegen ist es möglich, auch sehr komplexe Abläufe mit vielen Ausnahmen, Varianten, Sprüngen und Wiederholungen noch übersichtlich und verständlich darzustellen.

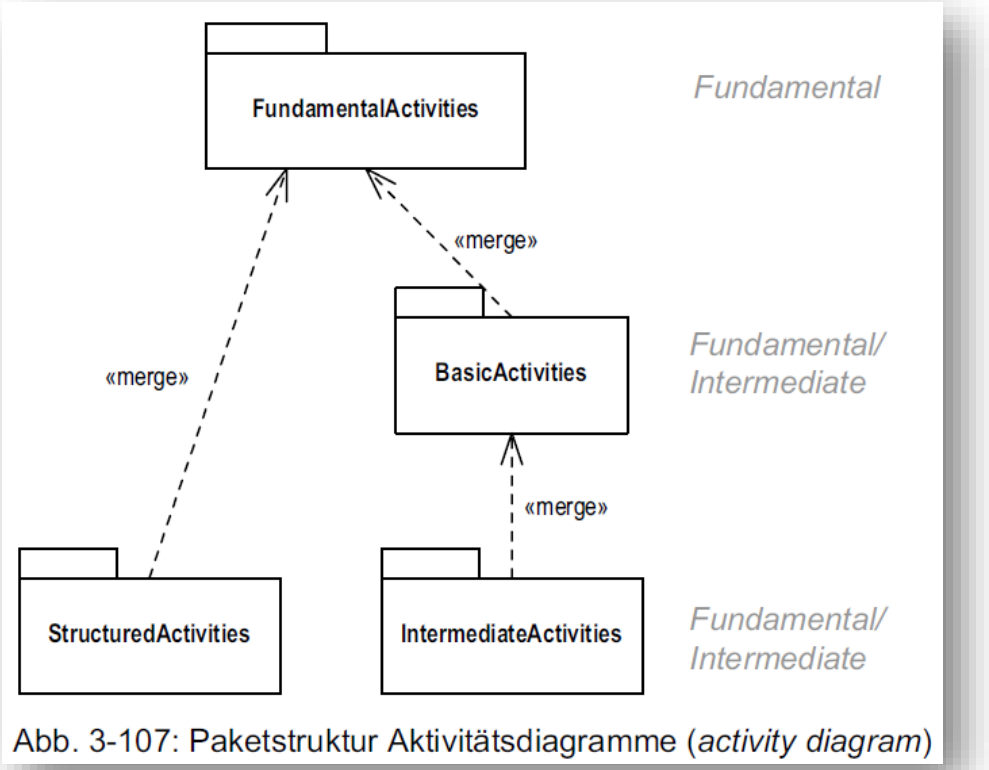
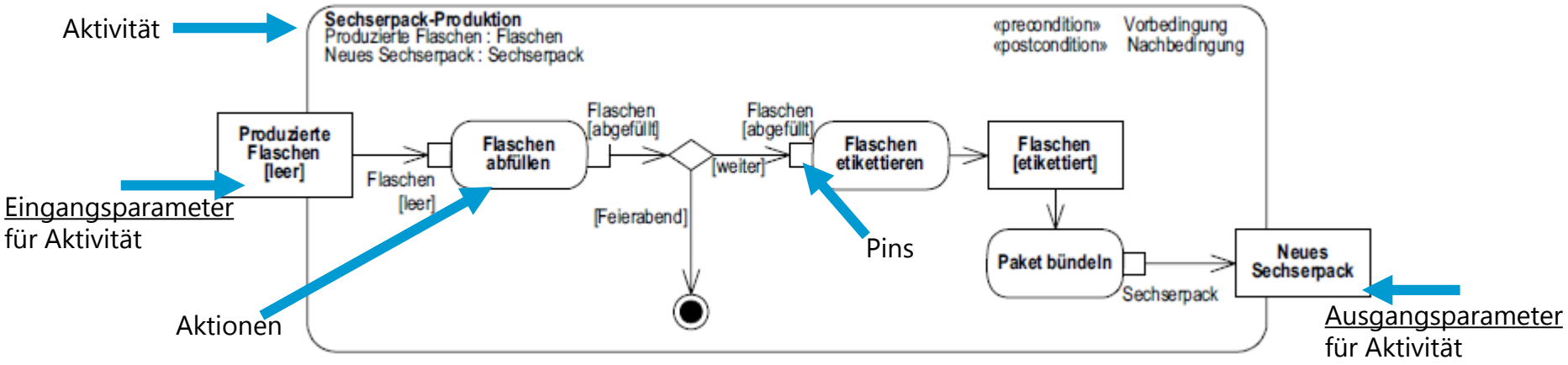


Abb. 3-107: Paketstruktur Aktivitätsdiagramme (*activity diagram*)

Notation und Semantik (1/6)

- Eine Aktivität wird durch ein Rechteck mit abgerundeten Ecken dargestellt. In dem Rechteck befinden sich die Knoten und Kanten der Aktivität.

- Links oben im Rechteck steht der Name der Aktivität.



In der Aktivität befinden sich verschiedene Arten von Knoten und Kanten:

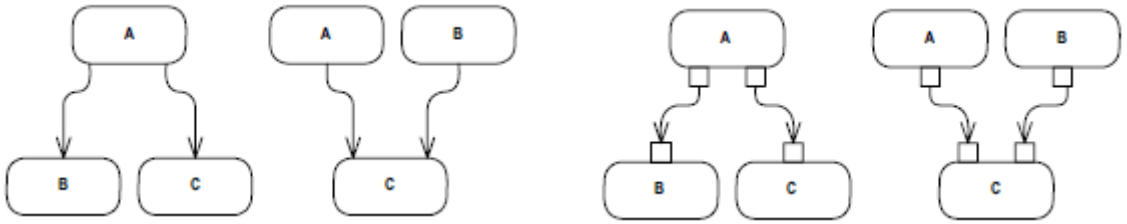
- Die Rechtecke mit den abgerundeten Ecken sind Aktionen.
- Die kleinen Rechtecke an den Aktionen sind so genannte Pins. Sie stellen die Eingangs bzw. Ausgangsparameterwerte für die Aktionen bereit.

Notation und Semantik (2/6)

- Basiert auf dem Token-Fluss. >> Zwischen den Knoten fließen über die Kanten Kontroll- oder Objekt-Token. Dabei gelten folgende elementare Regeln:
  - Eine Aktion kann erst starten, wenn an allen eingehenden Kanten Token zur Verfügung stehen (implizite Synchronisation; Und-Semantik).
  - Wenn eine Aktion terminiert, wird an allen ausgehenden Kanten ein Token zur Verfügung gestellt (implizites Splitting; Und-Semantik).
  - Ein Token fließt von einer Aktion zur nächsten,
    - wenn an der ausgehenden Aktion ein Token zur Verfügung gestellt wird,
    - die hinführende Aktion bereit ist, das Token aufzunehmen, und
    - wenn die Regeln der Kante den Token-Fluss zulassen.

- Es gibt zwei Arten von Kanten: Kontrollfluss- und Objektflusskanten

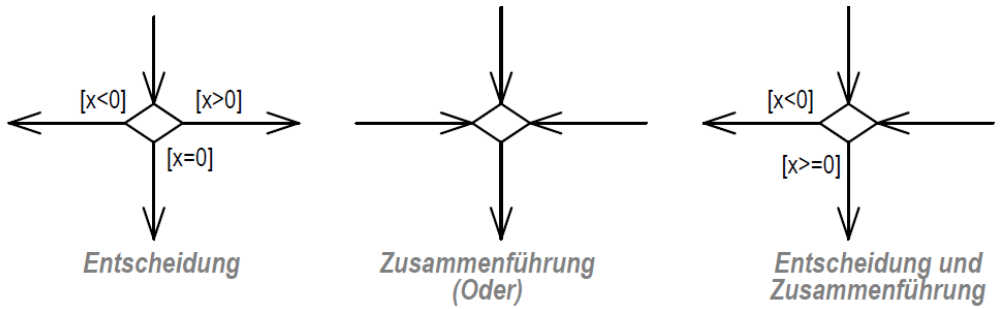
Und-Semantik bei: Kontrollfluss Objektfluss



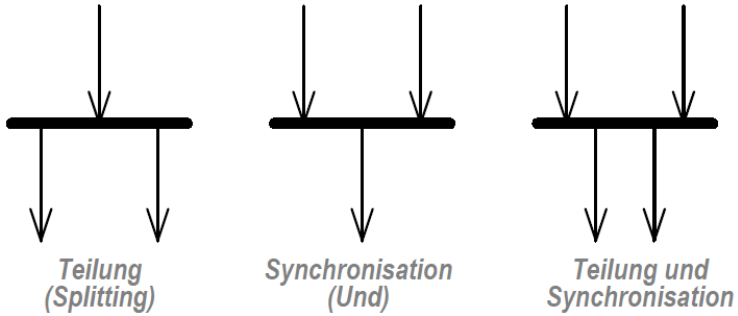
Notation und Semantik (3/6)

- Start-(Initial Node) und Endknoten (ActivityFinalNode)):

- Entscheidungen (MergeNode):

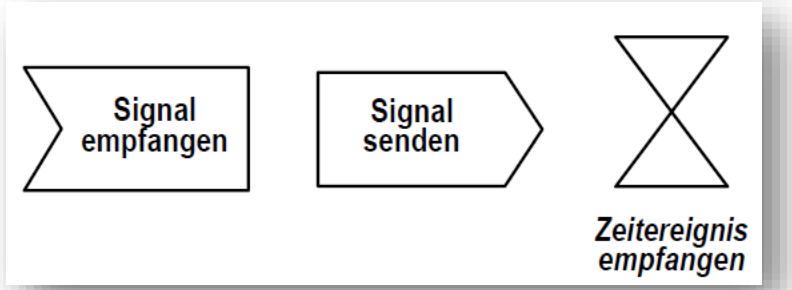


- Synchronisation/Teilung:



Notation und Semantik (4/6)

- Signal-Notation:
  - Signal senden: Sender erstellt aus Eingabedaten ein Signal für den Ereignisempfänger
  - Ereignis empfangen: Gegenstück zum Signal senden; Ablauf läuft erst weiter wenn Ereignis eintrifft



- Partition-Notation:
  - Sind mehrere Personen/Systeme an Aktionen beteiligt, gibt es die sog. Zuständigkeitsbahnen (Partitions)
  - Hierbei wird ersichtlich, in welchen Verantwortlichkeitsbereich die Aktionen fallen.
  - Auch bekannt als Swimlane-Darstellung

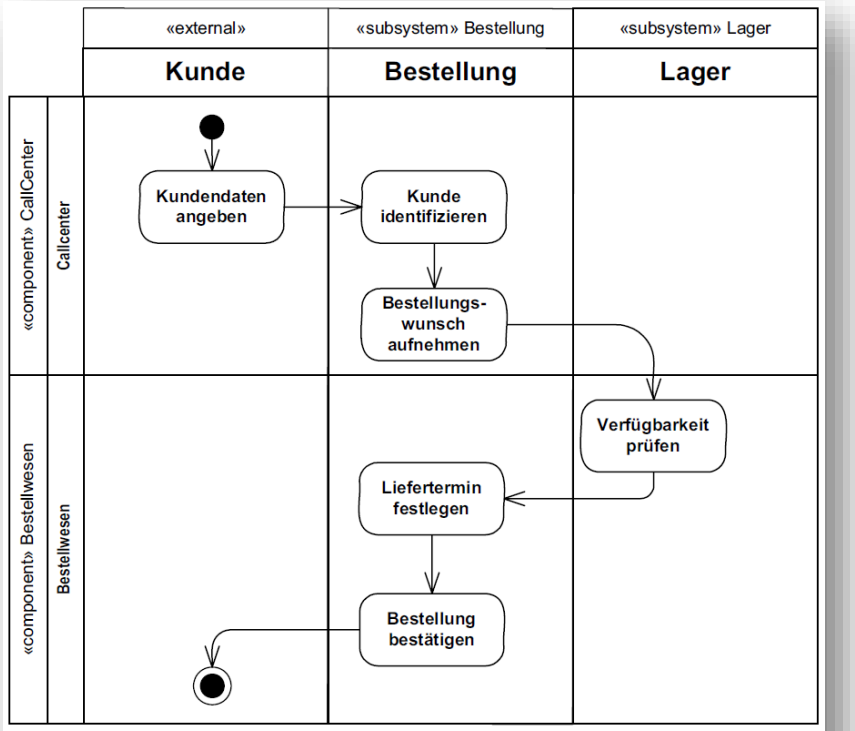
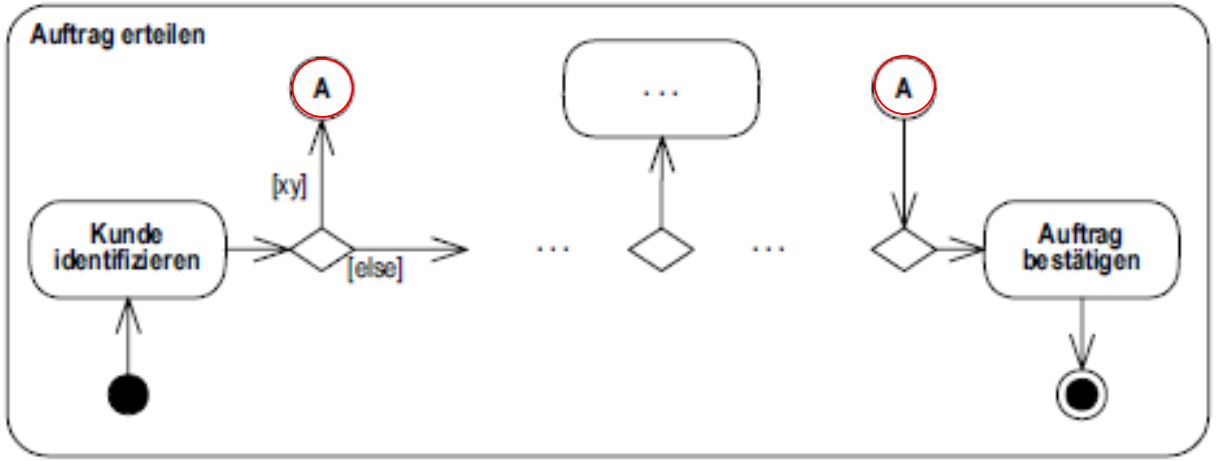


Abb. 3-120: Beispiel Partition

Notation und Semantik (5/6)

- Connector Edges
  - In Aktivitätsdiagrammen kann es vorkommen, dass eine Kante quer durch das ganze Diagramm gezeichnet werden muss. Damit das Diagramm übersichtlich bleibt, können so genannte Konnektoren eingeführt werden. Sie splitten eine Kante auf und haben nur Auswirkungen auf das Diagramm und nicht auf das zugrunde liegende Modell.



Übung

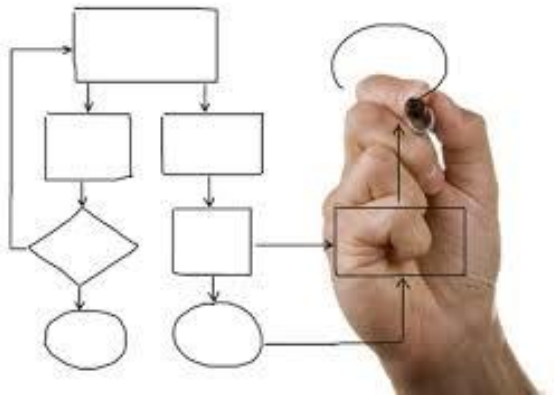
Kurzbeschreibung

Um den prinzipiellen Ablauf der Erstellung einer WiBe zu verstehen und eine Gesprächsbasis zu haben, soll dieser in Form eines Aktivitätsdiagramms dargestellt werden.

Ein Kollege der Fachabteilung versucht den Prozess zu erläutern:

„Bevor wir richtig loslegen können, muss überprüft werden, ob für den vorliegenden Fall bereits ein Kriterienkatalog existiert. Wenn nicht, muss einer vom Mitarbeiter erstellt werden. Wenn dieser erstellt wurde kann zum einen ein WiBe-Projekt vom Projektleiter im System angelegt werden und vom Mitarbeiter die sogenannten Richtwerttabellen befüllt werden. Der Projektleiter muss außerdem nach dem Anlegen Controlling&Reporting durchführen und gleichzeitig die Kriterien auswählen und diese anschließend belegen. Während des Belegens gibt es weiterhin die Möglichkeit weitere Kriterien im System auszuwählen, um sie zu belegen. Das wäre es im Groben und Ganzen."

>> Bitte erstellt das dazugehörige Aktivitätsdiagramm



Zusammenfassung:

MHP
A PORSCHE COMPANY

Checkliste Aktivitätsdiagramme:

- Was stellt eine Aktivität dar?
- Was stellt eine Aktion dar?
- Welche Arten von Aktivitätsknoten gibt es?
- Was sind Pins?
- Welche Arten von Kanten gibt es?

- Welche Voraussetzung muss erfüllt sein, damit eine Aktion ausgeführt werden kann?
- An welchen ausgehenden Kanten werden Token zur Verfügung gestellt, wenn eine Aktion terminiert?

- Was bedeuten mehrere ausgehende Kanten aus einem Startknoten?
- Wie viele eingehende Kanten darf ein Startknoten haben?
- Was bedeuten mehrere eingehende Kanten in einen Endknoten?
- Was bedeutet das Verhalten, das eine Entscheidung spezifizieren kann?
- Wie viele Pins kann eine Aktion besitzen?
- Was geschieht an den Eingangsparametern einer Aktivität, wenn diese aufgerufen wird?

Agenda

MHP
A PORSCHE COMPANY

1. Aufbau von Anwendungssystemen
2. **Methoden der SW Technik & UML**
   - Allgemeine Grundlagen UML
   - Anwendungsfall-/ Use Case Diagramme
   - Aktivitätsdiagramme
   - **Objekt- und Paketdiagramme**
   - Komponentendiagramme
   - Klassendiagramme
   - Sequenzdiagramme
   - Objektorientierte Methode

Das Denken in Objekten und Klassen

- ist von zentraler Bedeutung für das objektorientierte Konzept
- begleitet alle Phasen der SW-Entwicklung
- beginnt mit dem ersten Nachdenken über ein mögliches SW-System

Möchte man Aufgaben, die bisher manuell erledigt wurden, durch den Einsatz eines EDV-Systems unterstützen, so stellt man sich folgende Fragen:

Mit welchen Objekten hat man es beim Erledigen der Aufgaben zu tun?

Welche Eigenschaften dieser Objekte sind im Rahmen der zu erledigenden Aufgaben von Bedeutung?

Was wird mit den Objekten gemacht?

Beispielsystem Bibliothek: Objekte finden durch Beobachten

- viele **„Buch-Objekte“**
- werden ausgeliehen,
- werden zurückgegeben,
- …

- Personen, die Bücher mitnehmen und zurückbringen
- besitzen einen Bibliotheksausweis
- sind in der Bibliothek registriert
- **„Ausleiher-Objekte“**

Neben existenten Gegenständen in der realen Welt (wie z. B. Bücher) können auch Wesen (wie z. B. Ausleiher) oder Konzepte (wie z. B. Versicherungs-verträge) relevante Objekte darstellen.

Beispielsystem Bibliothek: Klassen finden

MHP
A PORSCHE COMPANY

Klassifikation: Gleichartige Objekte werden zu einer Klasse zusammengefasst!

- viele **„Buch-Objekte“**
- Abbildung durch die Klasse `Buch`

- viele **„Ausleiher-Objekte“**
- •Abbildung durch die Klasse `Ausleiher`

Klassen stellen die Baupläne für Objekte dar. Die **Klassen** sind die **Datentypen**, die **Objekte** die **Variablen** (Instanzen) dieser Datentypen. Ein Objekt wird gemäß dem Bauplan einer Klasse erzeugt.

Definition Objektdiagramme

- Ein Objektknoten (*ObjectNode*) gibt an, dass ein Objekt oder eine Menge von Objekten zu einem bestimmten Zeitpunkt in einem Ablauf vorhanden ist.

- Objektknoten können als ein- oder ausgehende Parameter in Aktivitäten (Aktivitätsparameter (*ActivityParameterNode*)) oder als ein- oder ausgehende Parameter von Aktionen (Pin) verwendet werden.

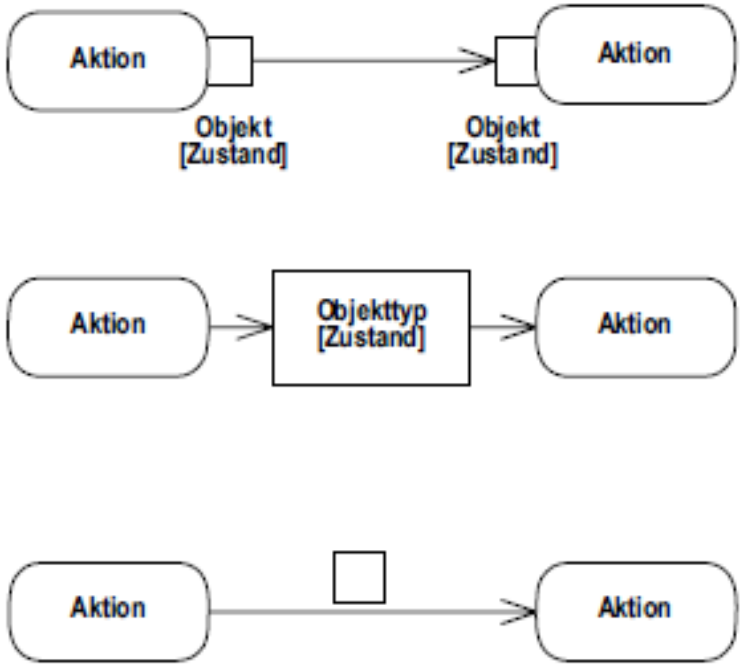
- Ein *Objektfluss* (*ObjectFlow*) ist wie ein Kontrollfluss eine spezielle Kante, bei der jedoch Objekte (*Objekt-Token*) transportiert werden.

Notation und Semantik - Objektdiagramme

- Mit dem Objektfluss wird ausgedrückt, dass die entsprechenden Objekte von den Aktionsknoten vorausgesetzt bzw. erzeugt oder verändert werden. Eine Aktion startet erst dann, wenn die benötigten Objekte vorliegen. Am Ende der Aktion werden die neuen oder geänderten Objekte bereitgestellt.

- Ein Pin ist immer einer Aktion zugeordnet. Es wird zwischen Eingangs- und Ausgabepin unterschieden (*InputPin*, *Output-Pin*). Eine Aktion kann beliebig viele Pins haben

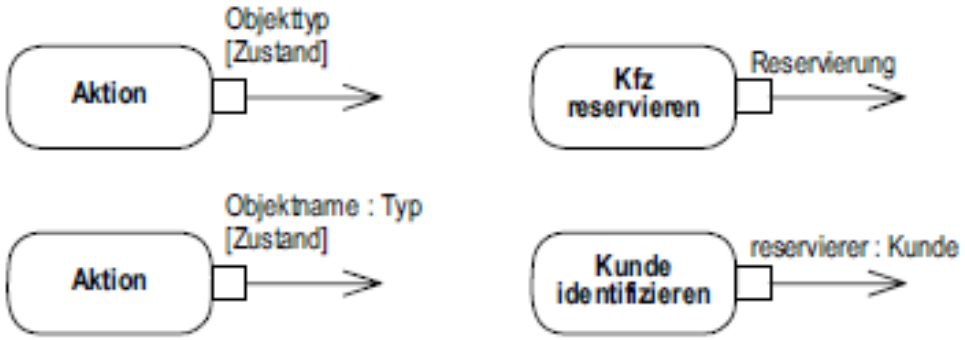
- Verschiedene Notationen:



Notation und Semantik – Beschriftung der Objektknoten

- Ein Objektknoten kann sowohl mit dem Namen des Objekttyps als auch mit dem Objektnamen und der zusätzlichen Angabe des Typs versehen werden.

- Zustände von Objekten müssen nicht modelliert werden; die Notation von Objektzuständen ist lediglich eine Möglichkeit, solche Sachverhalte hervorzuheben, soweit dies von besonderer Bedeutung ist.



- Es ist nicht zwingend erforderlich, dass ein Eingabepin eingehende Kanten und ein Ausgabepin ausgehende Kanten hat. Ein Ausgabepin ohne weiterführende Kanten drückt aus, dass die zugehörige Aktion einen Ausgabeparameter hat, der aber für den weiteren Ablauf der Aktivität keine Rolle spielt.

Definition Paketdiagramme

MHP
A PORSCHE COMPANY

- Ein Paket (*Package*) ist eine Ansammlung von Modellelementen beliebigen Typs, mit denen das Gesamtmodell in kleinere überschaubare Einheiten gegliedert wird.

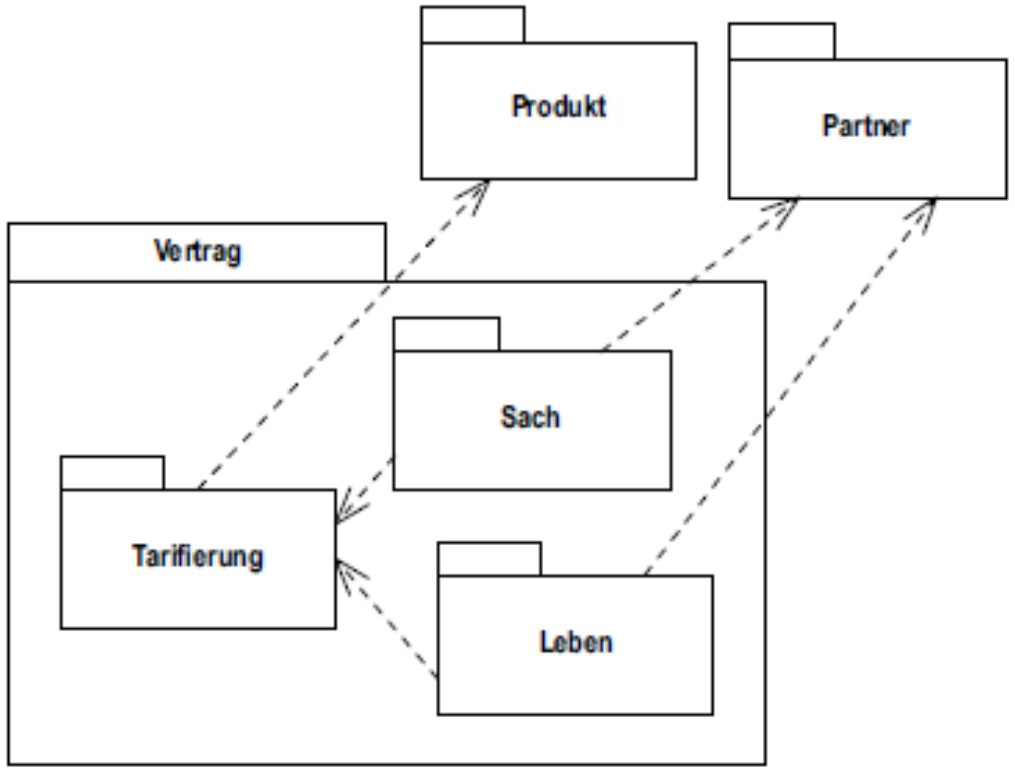
- Ein Paket definiert einen Namensraum, d.h., innerhalb eines Paketes müssen die Namen der enthaltenen Elemente eindeutig sein. Jedes Modellelement kann in anderen Paketen referenziert werden, gehört aber nur zu höchstens einem (Heimat-)Paket. Pakete können wiederum Pakete beinhalten.

- Pakete können verschiedene Modellelemente enthalten, beispielsweise Klassen und Anwendungsfälle. Sie können hierarchisch gegliedert werden, d.h. ihrerseits wieder Pakete enthalten.

Notation und Semantik - Paketdiagramme

- Ein Paket wird in Form eines Aktenregisters dargestellt. Innerhalb dieses Symbols steht der Name des Paketes. Werden innerhalb des Symbols Modellelemente angezeigt, steht der Name auf der Registerlasche, anderenfalls innerhalb des großen Rechteckes.

- Oberhalb des Paketnamens können Stereotypen notiert wer



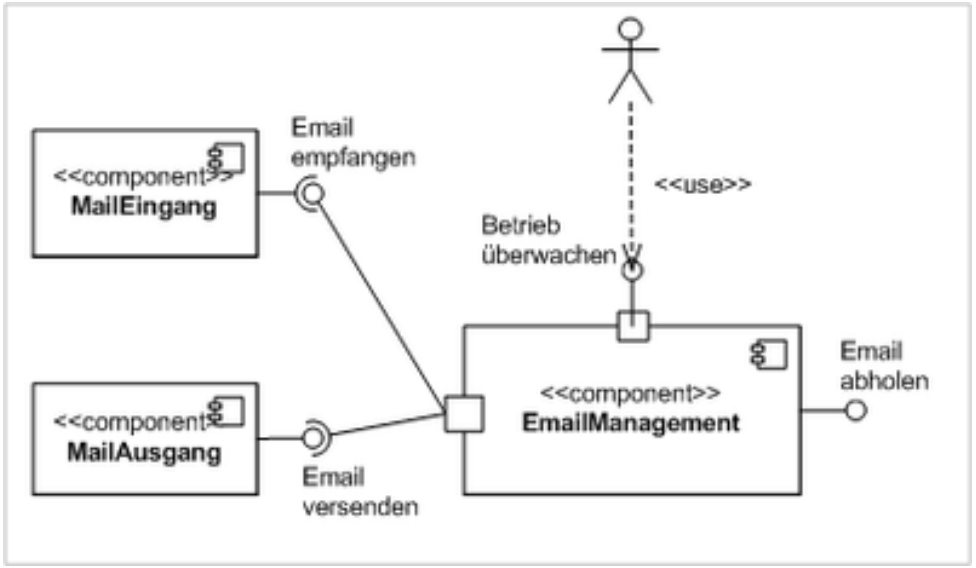
Agenda

MHP
A PORSCHE COMPANY

1. Aufbau von Anwendungssystemen
2. **Methoden der SW Technik & UML**
   - Allgemeine Grundlagen UML
   - Anwendungsfall-/ Use Case Diagramme
   - Aktivitätsdiagramme
   - Objekt- und Paketdiagramme
   - **Komponentendiagramme**
   - Klassendiagramme
   - Sequenzdiagramme
   - Objektorientierte Methode

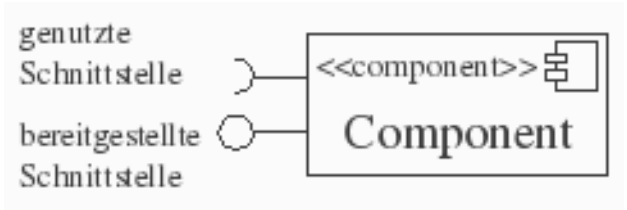
Definition

- Es zeigt eine bestimmte Sicht auf die Struktur des modellierten Systems.
- Die Darstellung umfasst dabei typischerweise Komponenten mit deren Schnittstellen bzw. Ports. Es zeigt auch, wie Komponenten über Abhängigkeitsbeziehungen und Konnektoren miteinander verbunden sind.
- Um das Innere einer Komponente darzustellen, zeigt ein Komponentendiagramm oft Notationselemente, die sonst vor allem in Klassen- oder Kompositionsstrukturdiagrammen angezeigt werden, zum Beispiel Klassen oder Parts.
- Beispiel:

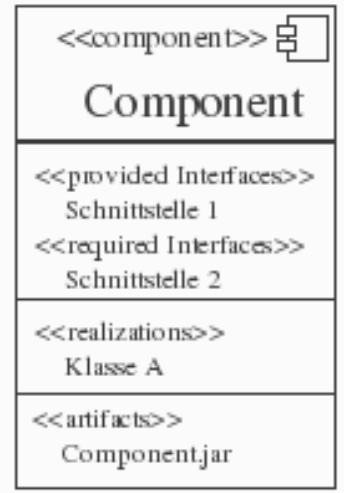


Notation und Semantik

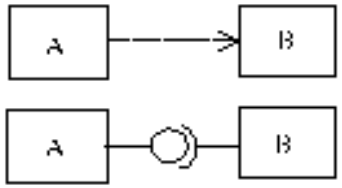
- Eine Komponente ist eine spezielle strukturierte Klasse (Class). Sie kann also auch Attribute und Operationen besitzen.
- Über dem Namen der Komponente steht das Schlüsselwort *component*.
- Notation einer Komponente (externe Sicht):



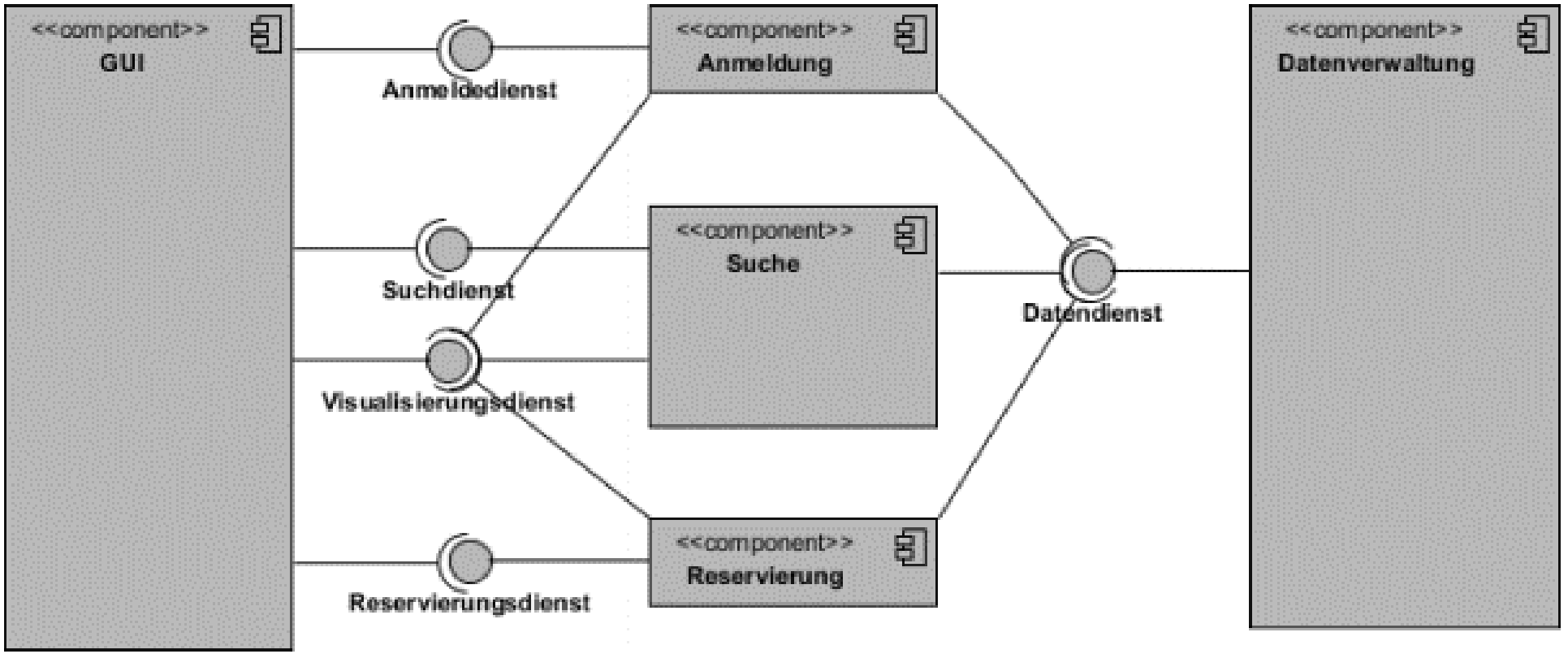
- Notation einer Komponente (interne Sicht):



- Notation von Beziehungen:



Beispiel - Seminarverwaltung



Übung

MHP
A PORSCHE COMPANY

Kurzbeschreibung

Folgende Komponenten sollen in einem Komponentendiagramm dargestellt werden:

- Sicherheit
  - Diese Komponente umfasst alle sicherheitstechnischen Aspekte des Systems. Das System verfügt über eine eigene Benutzerverwaltung und ein einfaches Rollenkonzept.
- Projektverwaltung
  - Die Erstellung einer Wirtschaftlichkeitsbetrachtung erfolgt in Form von Projekten. Hier wird auf einen Kriterienkatalog sowie auf die Benutzerverwaltung zugegriffen.
- Kriterienkatalog
  - Für ähnliche Projekte wird in der Regel derselbe Kriterienkatalog verwendet. Diese Komponente enthält alle notwendigen Funktionen zur Verwaltung der Kriterienkataloge.
- Controlling & Reporting
  - Im Controlling&Reporting können verschiedene Projekte die auf demselben Kriterienkatalog basieren miteinander verglichen und ausgewertet werden.
- Drucken
  - Die Komponente Drucken kapselt alle Funktionen des Druckens. Jede Ansicht des Reportings und Controllings muss dabei druckbar sein.

>> Bitte erstellt das dazugehörige Komponentendiagramm (externe Sicht)

Zusammenfassung:

- Was ist eine Komponente?
- Welche Arten von Schnittstellen können Komponenten haben?

Agenda

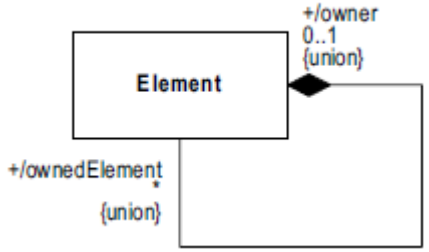
MHP
A PORSCHE COMPANY

1. Aufbau von Anwendungssystemen
2. **Methoden der SW Technik & UML**
   - Allgemeine Grundlagen UML
   - Anwendungsfall-/ Use Case Diagramme
   - Aktivitätsdiagramme
   - Objekt- und Paketdiagramme
   - Komponentendiagramme
   - **Klassendiagramme**
   - Sequenzdiagramme
   - Objektorientierte Methode

Basiskonzepte

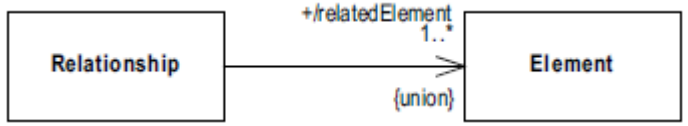
Basisklasse der UML

Die Oberklasse heißt *Element* und hat die Eigenschaft, andere Elemente besitzen zu können.



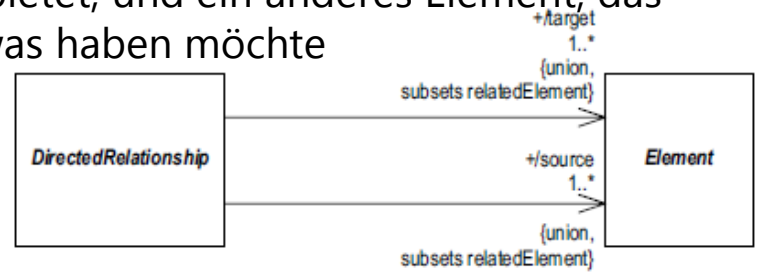
Basisklasse Beziehung

Die Beziehung (Relationship) ist ein abstraktes Konzept, um Elemente in Beziehungen zueinander zu bringen.



Gerichtete Beziehung

Oft gibt es ein Element, das etwas anbietet, und ein anderes Element, das etwas haben möchte



Notation Kommentar

Neben den elementaren abstrakten Konzepten gibt es hier auch ein ganz konkretes Element: den Kommentar (*Comment*). Ein Synonym ist Notiz.

Beispielsystem Bibliothek: Methoden für die Klasse Buch

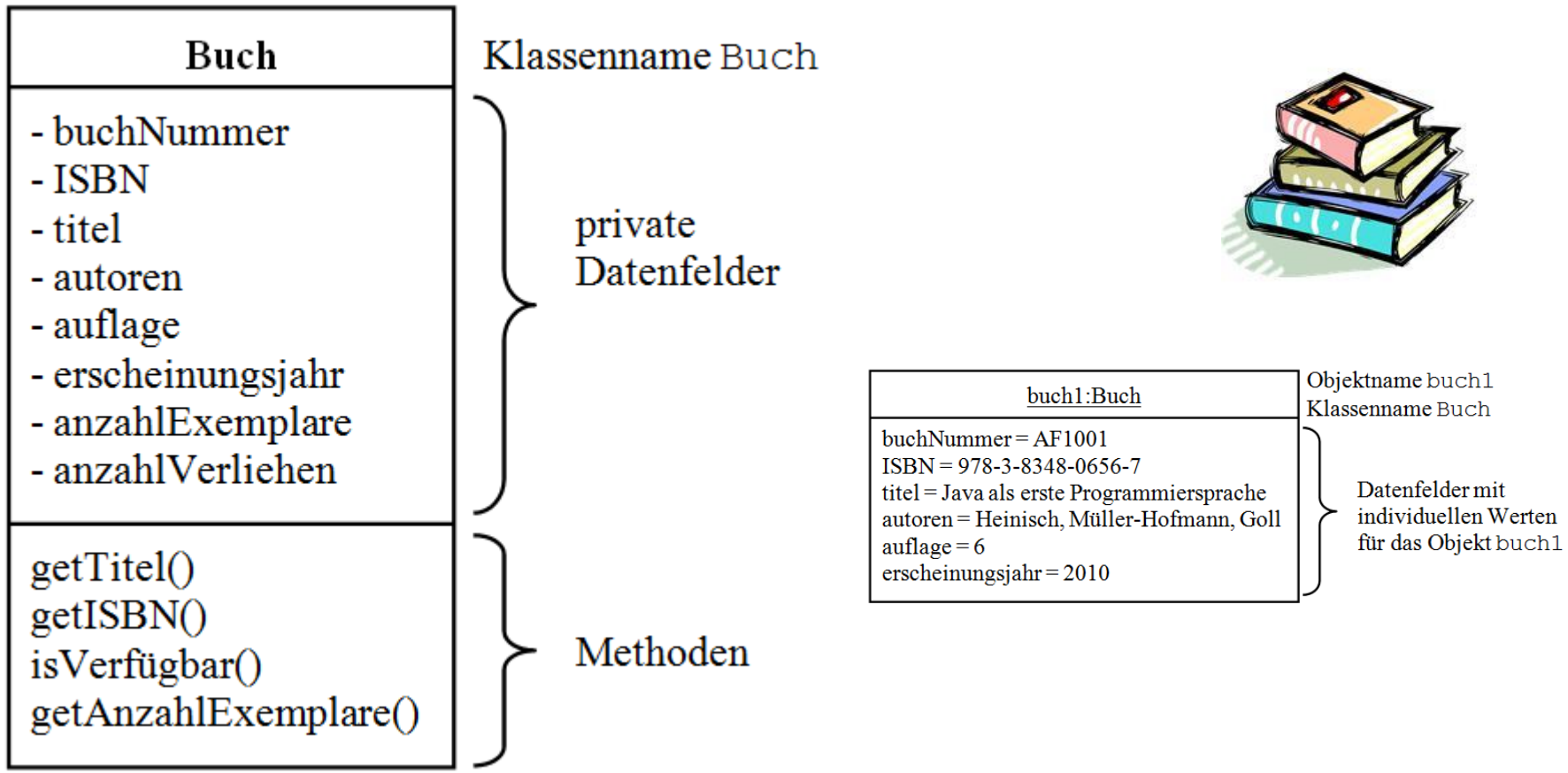
Wie kann ein Buch-Objekt die Anwendungsfälle unterstützen?

- Anwendungsfall „Buch-Recherche durchführen“
  - Ein einzelnes Buch-Objekt kann keine „Recherche durchführen“.
  - Ein einzelnes Buch-Objekt kann aber seine Daten für die Recherche bereitstellen:
  - getTitel()
  - getISBN()

- Anwendungsfall „Buchverfügbarkeit prüfen“
  - Ein Buch-Objekt muss wissen, wie viele Exemplare in der Bibliothek sind.
    → neues Datenfeld anzahlExemplare
  - Ein Buch-Objekt ermöglicht die Abfrage der vorhandenen Exemplare über eine Methode getAnzahlExemplare()
  - …

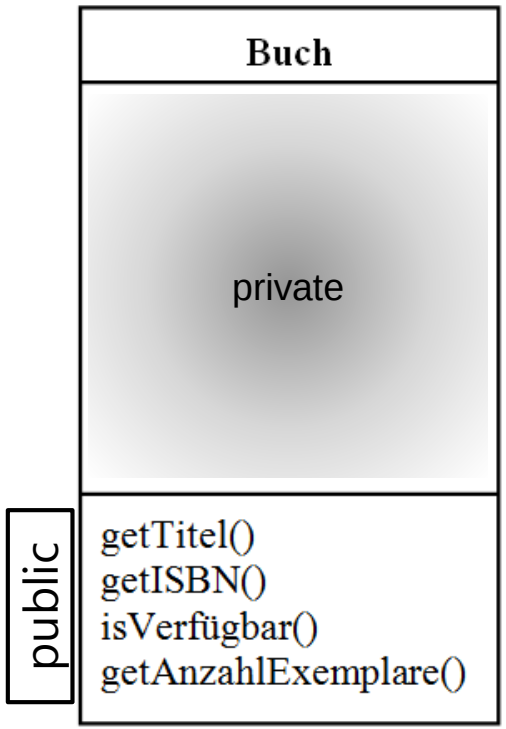
Beispielsystem Bibliothek: UML-Notation der Klasse mit Methoden

UML-Notation der Klasse Buch nach einigen Überlegungen:



Kapselung / Information Hiding / Geheimnisprinzip

- Abschotten der internen Implementierung vor direktem externen Zugriff.
- Zugriff auf Daten und private Methoden nur über explizit definierte Schnittstelle.
- Aufrufer ist unabhängig von Implementierungsdetails.

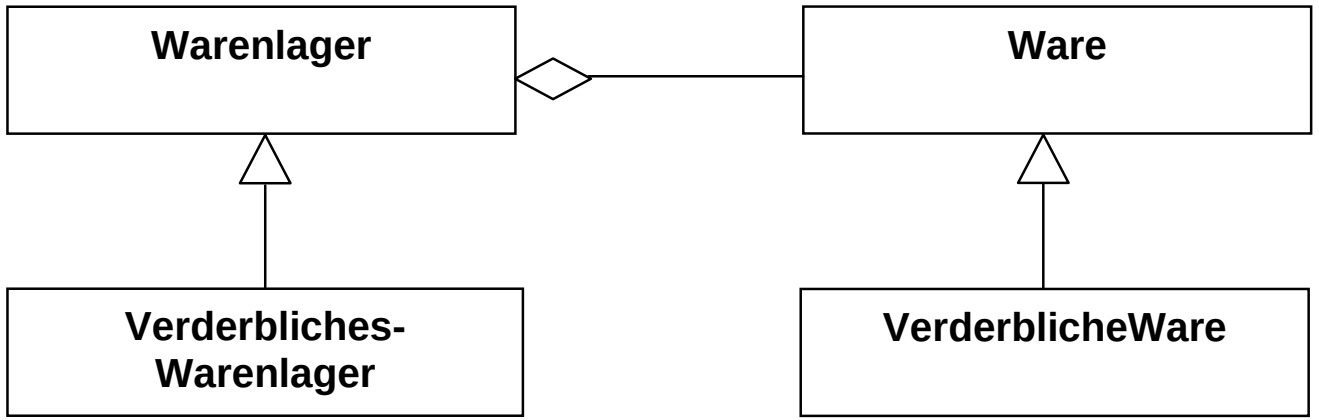


Datenfelder sind verborgen (gekapselt / nach außen nicht sichtbar) und können nur von Methoden der eigenen Klasse manipuliert werden.

Methodenköpfe sind die Schnittstellen. Die Implementierung der Methoden wird ebenfalls verborgen.

Wiederverwendung

- Durch Vererbung:
  Die spezialisierten Klassen erben den Programmcode der Vaterklassen.
- Durch Aggregation:
  Verwendung von fertigen Komponenten, Modulen und Bibliotheken.
- Durch Polymorphie:
  Wiederverwendung gesamter Programmsysteme • Prinzip von Frameworks. Spezifische Eigenschaften werden durch Ableiten und Überschreiben von Methoden hinzugefügt.



Objektorientierung und die Prinzipien der SW-Technik

Wie werden die Prinzipien der SW-Technik in der Objektorientierung abgedeckt?

- Prinzip der Abstraktion:
  Abstrakte Datentypen, Schnittstellen, Klassen und Pakete.
- Prinzip der Kopplung und Bindung (Low Coupling, High Cohesion):
  Anwendung auf Klassen und Pakete.
- Prinzip der Hierarchisierung:
  Vererbungshierarchien und Zerlegungshierarchien.
- Prinzip der Modularisierung:
  Klassen und Pakete.
- Geheimnisprinzip:
  Klassen (Objekte) und Pakete.

Die Prinzipien der SW-Technik lassen sich optimal auf die Objektorientierung anwenden, bzw. sind Bestandteil der Objektorientierung.

Namensräume (Namespaces) (1/2)

- Ein benennbares Element **(*NamedElement*)** ist ein Element, das einen Namen und eine definierte Sichtbarkeit (*public, private, protected, package*) haben kann. Der Name des Elementes und die Sichtbarkeit sind optional.

- Ein Namensraum **(Namespace)** ist ein benennbares Element, das benennbare Elemente enthält, die eindeutig über ihre Namen identifizierbar sind.

- Das Importieren eines Elementes **(*ElementImport*)** ist eine Beziehung zwischen einem Namensraum und einem paketierbaren Element in einem anderen Namensraum. Das referenzierte Element kann dann direkt über seinen (unqualifizierten) Namen angesprochen werden.

- Das Importieren eines Paketes **(*PackageImport*)** ist semantisch äquivalent mit dem Importieren jedes einzelnen Elementes des Paketes. Ein Aliasname ist hier natürlich nicht möglich.

Namensräume (Namespaces) (2/2)

Sichtbarkeit:

„+" public *„#" protected* *„-„ private* *„~" package*

Folgende Importmöglichkeiten bestehen:

- «import»

  Sichtbarkeit ist öffentlich (***public***). In Abb. 2-14 ist beispielsweise Postanschrift in Bestellung sichtbar. Der öffentliche Import ist eine transitive Beziehung.

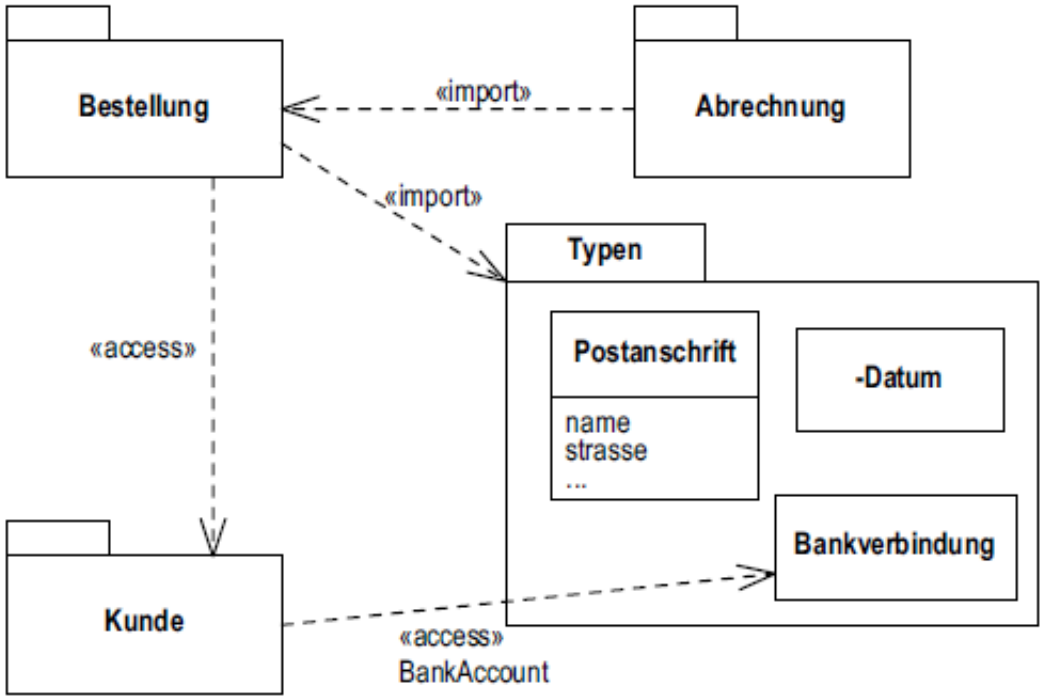
- «access»

  Nichtöffentliche Sichtbarkeit (***private***). In Abb. 2-14 ist beispielsweise Kunde nur in Bestellung sichtbar, aber nicht in Abrechnung. Der private Import ist nicht transitiv.

Namensräume (Namespaces) - Beispiel

Folgende Aussagen können zu diesem Diagramm getroffen werden:

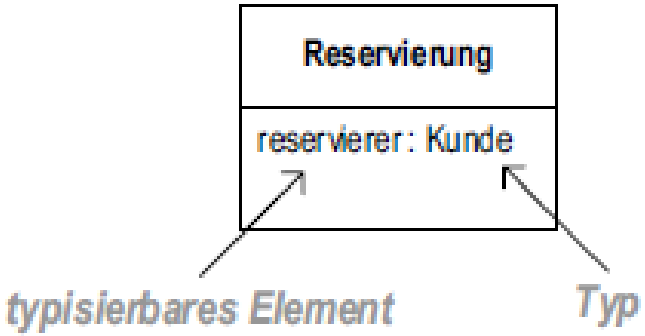
- *Datum* ist ein privates Element im Paket *Typen*.
- *Postanschrift* und *Bankverbindung* sind im Paket *Bestellung* sichtbar (Paketimport). *Datum* ist in *Bestellung* nicht sichtbar, da Elemente mit privater Sichtbarkeit nicht importiert werden.
- Der öffentliche Import ist transitiv. Daher sind *Postanschrift* und *Bankverbindung* auch im Paket *Abrechnung* sichtbar.
- Die Elemente des Paketes *Kunde* sind im Paket *Bestellung* sichtbar (privater Paketimport), aber nicht im Paket *Abrechnung*.
- Im Paket *Kunde* ist die *Bankverbindung* unter dem Namen *BankAccount* sichtbar (Alias).



Typisierbare Elemente (TypedElements)

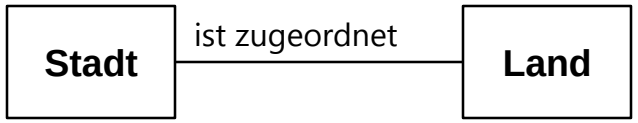
- Ein typisierbares Element **(*TypedElement*)** ist ein benennbares Element (*NamedElement*), das einen Typ haben kann. Zum Beispiel sind Attribute oder Parameter typisierbare Elemente.

- Ein Typ **(*Type*)** spezifiziert eine Menge von Werten eines typisierbaren Elementes. Zum Beispiel sind einfache Datentypen und Klassen Typen.

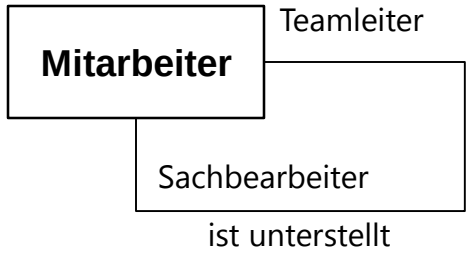


Assoziationen

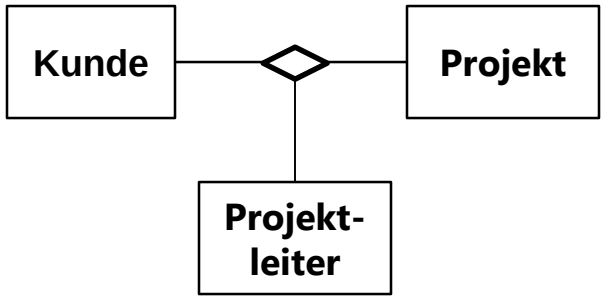
Binäre Assoziation:



Reflexive Assoziation:



Ternäre Assoziation:

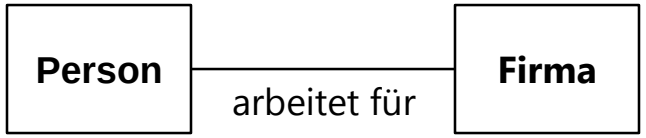


Wichtige Eigenschaften einer Assoziation:

- Name der Assoziation
- Rollen (Endnamen) am Ende der Assoziation
- Multiplizitäten an beiden Enden
- Navigation
- Assoziationsklasse

Assoziationen

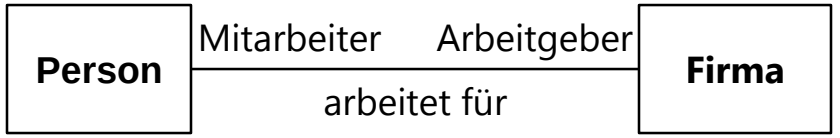
Name einer Assoziation:



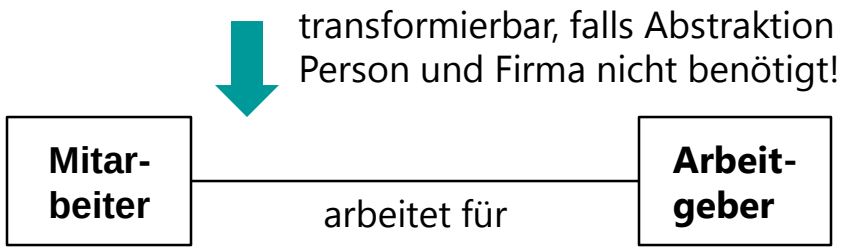
Leserichtung:

- von oben nach unten
- von links nach rechts
- mit Lesepfeil

Endnamen einer Assoziation:



- Klasse ist als Rolle an der Assoziationsbeziehung beteiligt.
- Rollennamen kann am Endpunkt der Assoziation (als Endnamen) notiert werden.
- Bei Verwendung von Rollennamen kann der Name der Assoziation entfallen.



Assoziationen

Navigation:

Erläuterung:

- Eine Assoziation (Strich zwischen zwei Klassen) ist bidirektional oder unspezifiziert.

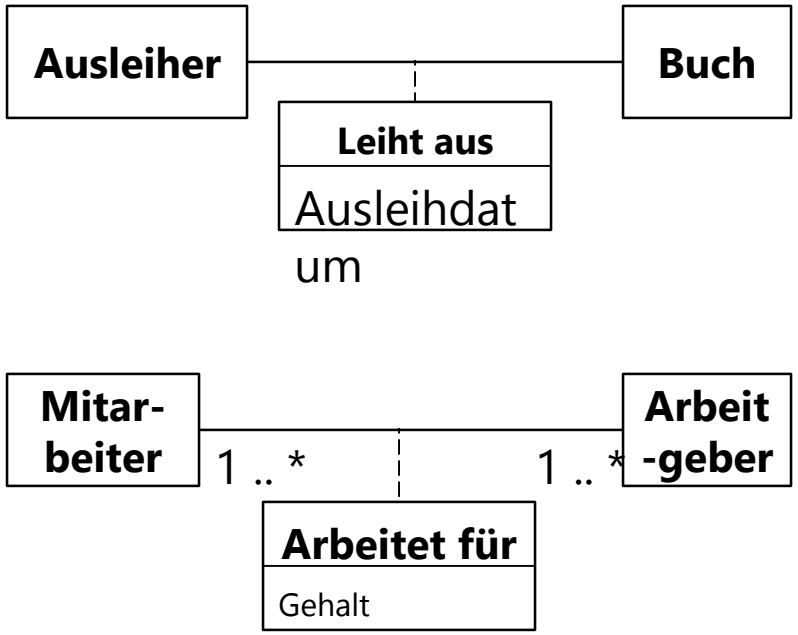
  Beispiel: Ein Ausleiher leiht ein Buch aus und das Buch wird durch den Ausleiher ausgeliehen.

- Die Navigierbarkeit kann auf eine Richtung eingeschränkt werden → unidirektionale Navigierbarkeit.

  Beispiel: Eine Person als Wahlberechtigter kann mit Hilfe eines Stimmzettels wählen, aber vom Stimmzettel aus soll es nicht möglich sein, zurück zur Person zu gelangen.

Assoziationen

Assoziationsklasse:



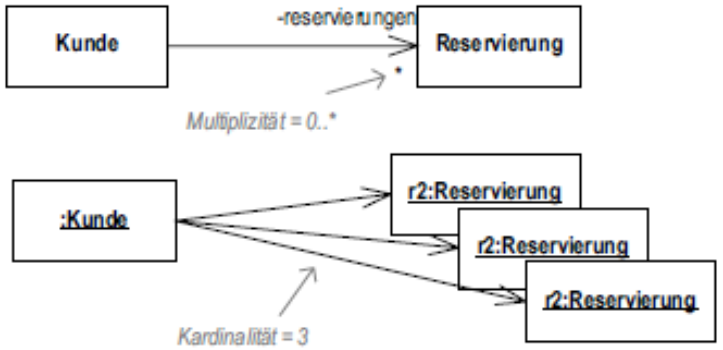
Erläuterung:

- Die Assoziationsklasse fällt weg, wenn die Beziehung gestrichen wird.
- Eine Assoziationsklasse gehört also zur Assoziation und beschreibt diese näher.
- Enthält die Assoziationsklasse keine Operationen, so spricht man auch von einem Assoziationsattribut.
- Der Name der Assoziationsklasse sollte mit dem Namen der Assoziation übereinstimmen bzw. sich daraus direkt ableiten lassen.

Multiplizitäten (Multiplicities)

- Eine Multiplizität (***MultiplicityElement***) ist die Definition eines Intervalls positiver ganzer Zahlen der erlaubten Kardinalitäten (***Cardinality***). Eine Kardinalität ist die Anzahl der Elemente in einer Menge.

- Die Begriffe Multiplizität und Kardinalität werden häufig synonym verwendet. Das ist allerdings verkehrt. Das Beispiel macht den Unterschied deutlich:

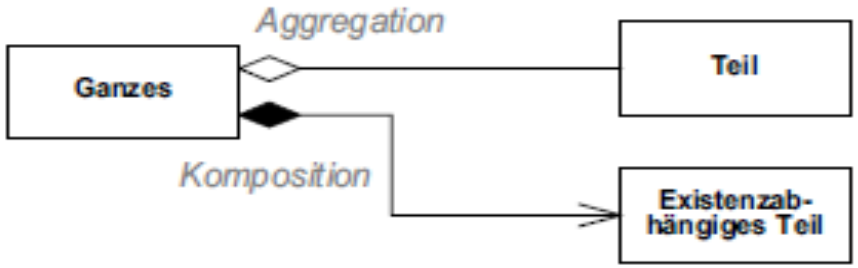


Beispiele für Multiplizitäten:	
0..1	null oder eins
1	genau eins (Kurzschreibweise für 1..1)
*	null bis beliebig viele (Kurzschreibweise für 0..*)
1..*	eins bis beliebig viele
5..3	nicht erlaubt; der untere Wert muss kleiner gleich dem oberen Wert sein
-1..0	nicht erlaubt; die Werte müssen alle positiv sein

Aggregation und Komposition

- Die Komposition wird wie die Aggregation als Linie zwischen zwei Klassen gezeichnet und mit einer kleinen Raute auf der Seite des Ganzen versehen.

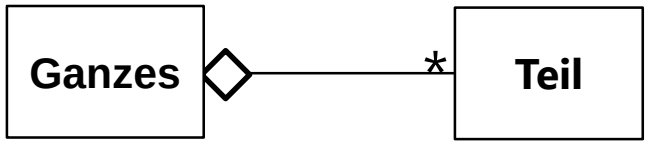
- Im Gegensatz zur Aggregation wird die Raute jedoch ausgefüllt.



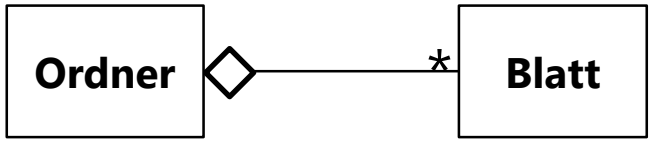
- Aggregation

- Komposition

UML-Grundlagen – Aggregation



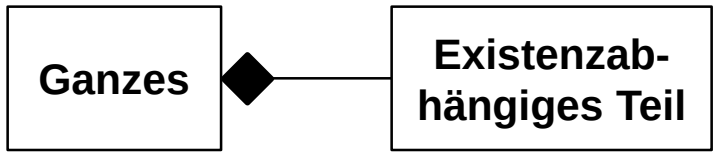
Aggregation (universell wiederverwendbare Teile):



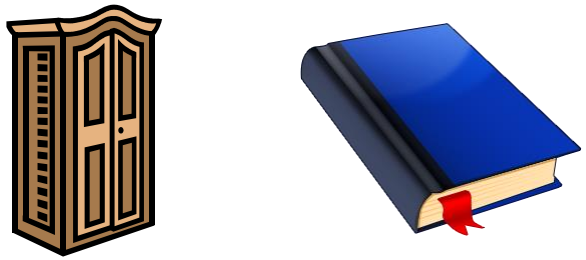
Erläuterung:

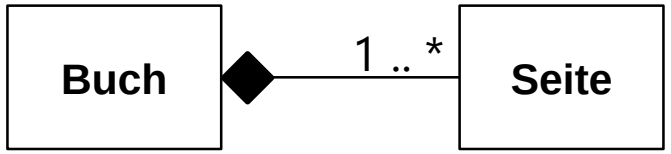
- In der Implementierung wird die Aggregation mit Zeigern bzw. Referenzen umgesetzt.
- Das „Groß“-Objekt (das Ganze) zeigt auf eine beliebige Anzahl von „Klein“-Objekten (das Teil).
- Die Aggregation ist eine Zusammensetzung eines „Groß“-Objektes aus einer Menge von „Klein“-Objekten. Das „Groß“-Objekt nimmt stellvertretend für seine „Klein“-Objekte alle Aufgaben wahr.
- Die Aggregation ist eine stärkere Ausprägung der Assoziation!

UML-Grundlagen – Komposition



Komposition („feste Verbindung“ der Teile):



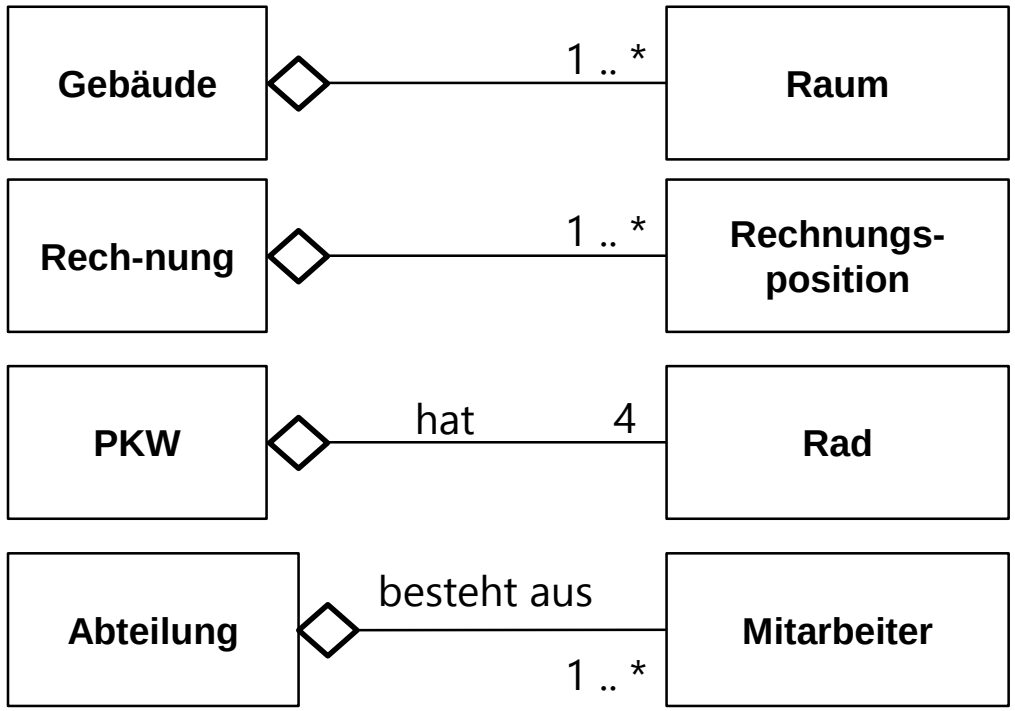


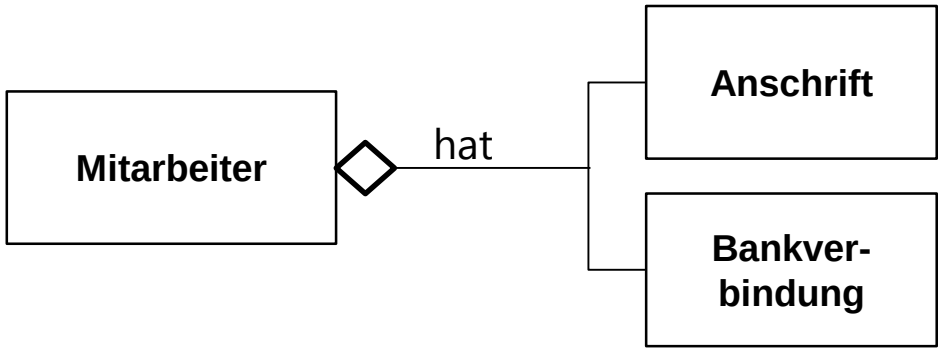
Erläuterung:

- Die Komposition drückt den **exklusiven Besitz** eines „Klein“-Objektes durch ein „Groß“-Objekt aus. Ein „Klein“-Objekt kann nur genau einem „Groß“-Objekt gehören.
- Die Komposition ist eine stärkere Beziehungsform als die Aggregation, bei der die Teile vom Ganzen existenzabhängig sind.
- Wird das „Groß“-Objekt zerstört, so wird auch automatisch das „Klein“-Objekt (existenzabhängiges Teil) zerstört.

Aufgabe: Aggregation und Komposition

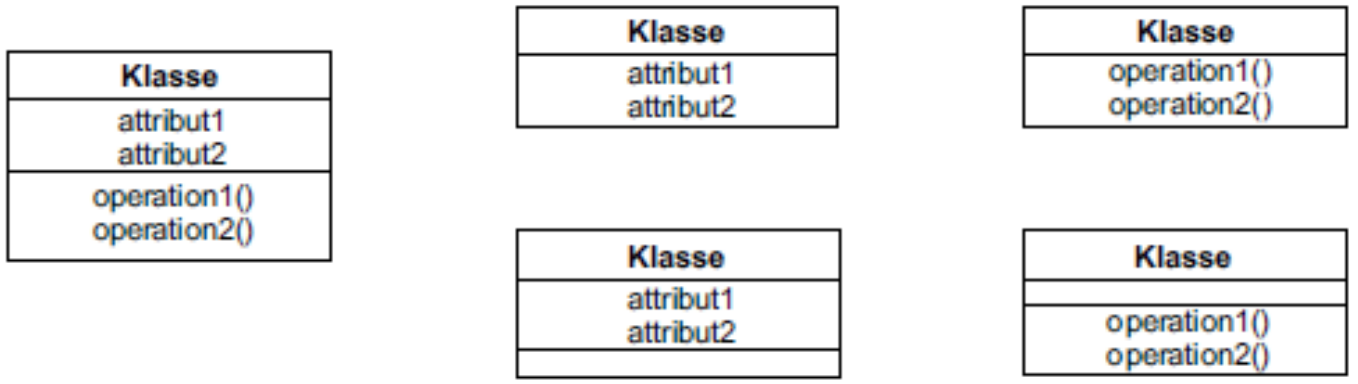
Bestimmen Sie für die einzelnen Beispiele, ob es sich um eine Aggregation oder eine Komposition handelt. Falls es sich um eine Komposition handelt, füllen Sie die entsprechende Raute aus.





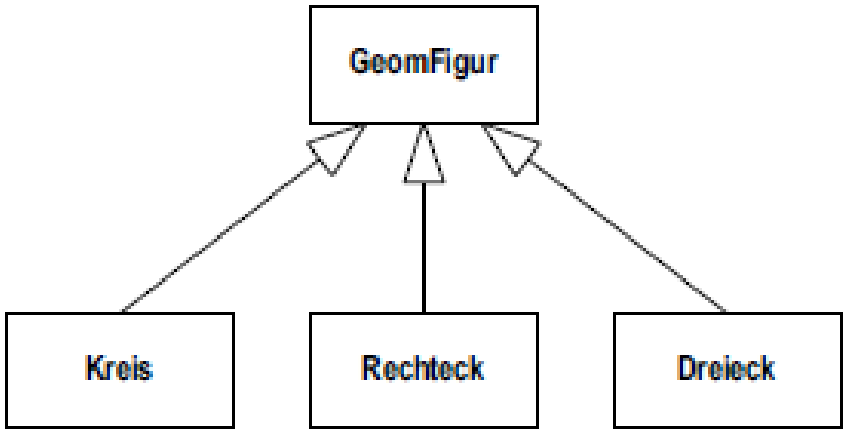
Klasse (Class)

- Eine Klasse **(*Class*)** beschreibt eine Menge von Instanzen, die dieselben Merkmale, Zusicherungen und Semantik haben.
- Klassen werden durch Rechtecke dargestellt, die entweder nur den Namen der Klasse tragen oder zusätzlich auch Attribute und Operationen.
- Verschiedene Notationsvarianten für Attribute und Operationen:



Generalisierung (Generalization)

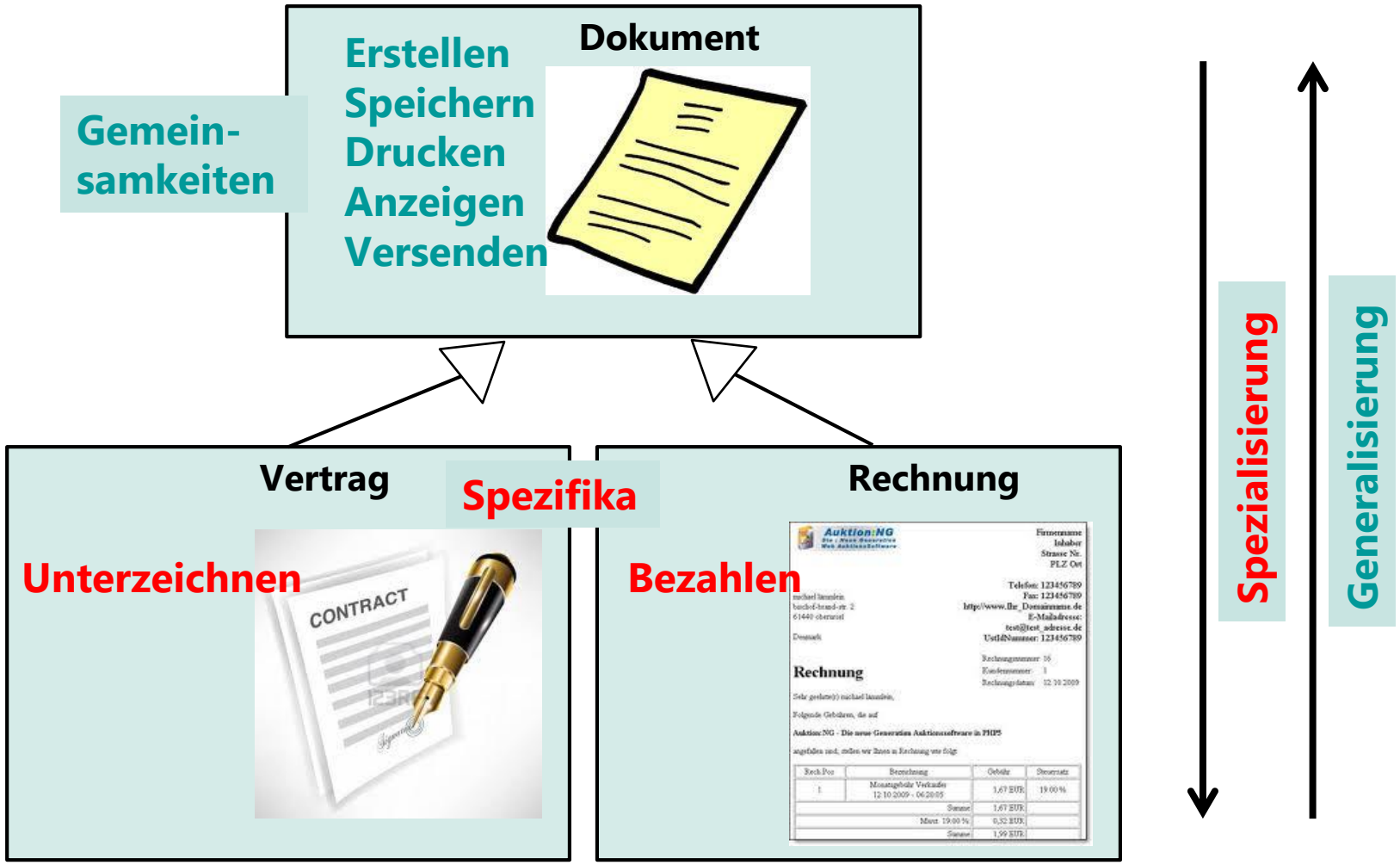
- Die Generalisierung **(*Generalization*)** ist ein Abstraktionsprinzip zur hierarchischen Strukturierung der Semantik eines Modells.
- Eine Generalisierung ist eine Beziehung zwischen einem allgemeinen und einem speziellen Classifier, wobei der speziellere weitere Merkmale hinzufügt und sich kompatibel zum allgemeinen verhält.
- Die Generalisierung wird durch einen speziellen Pfeil notiert, der vom speziellen zum allgemeinen Element zeigt
- Notation:



Vererbung – Ausgangslage: mehrere Klassen mit Gemeinsamkeiten

Problem: Bei separater Implementierung der Gemeinsamkeiten in unterschiedlichen Klassen kommt es zur Code-Duplizierung --> Änderungsaufwand an mehreren Stellen --> erschwerte Wartbarkeit.

Vererbung – Lösung: Vererbung mit Variantenbildung



Übung: Finden von Klassen und geeigneten Abstraktionen

Betrachten Sie die folgenden Entitäten der realen Welt. Finden Sie geeignete Klassen zur Abbildung der Entitäten in einem Klassendiagramm und identifizieren Sie für die gefundenen Klassen weitere Abstraktionen im Sinne einer Generalisierung. Hinweis: Zum Finden der Abstraktionen können Sie auch im Internet recherchieren.

Vorbereitungszeit: 10 Minuten

Ergebnisdiskussion: 10 Minuten

Übung

MHP
A PORSCHE COMPANY

Kurzbeschreibung

- Zu einem Projekt existiert immer mindestens eine Version, welche wiederum Bezug zu mindestens einer oder mehreren Projektalternativen hat. Des Weiteren hat ein Projekt genau einen Projektkopf, welcher mit mindestens einem Projektattribut verknüpft ist. Die Attribute können in einem Projekt mit konkreten Werten belegt sein.

- WiBe unterscheidet verschiedene Kriterien (Monetäre Kriterien, Dringlichkeitskriterien, Qualitätskriterien, Externes Kriterium), wobei ein Monetäres Kriterium mit einem oder mehreren Richtwerttabellen verknüpft ist.

- Verschiedene Kriterien werden in einem Kriterienkatalog miteinander verbunden. Solch ein Kriterienkatalog ist mit einem Projekt über das Projektattribut verbunden.

- Ebenfalls zu beachten ist das Element Notiz, welches die in vielen Teilen der Anwendung verfügbare Notizfunktion abbilden soll.

>> Bitte erstellt für dieses Datenmodell ein Klassendiagramm

Zusammenfassung:

MHP
A PORSCHE COMPANY

Checkliste Klassendiagramm:

- Wie sind gerichtete Beziehungen definiert?
- Können gerichtete Beziehungen mehr als ein Quell- bzw. Zielelement haben?

- Kann für ein Paketimport ein Alias definiert werden?
- Was ist der Unterschied zwischen privatem und öffentlichem Import?
- Welche Art von Elementen können mit dem Elementimport importiert werden?

- Nennen Sie ein Beispiel für einen Typ.
- Nennen Sie ein Beispiel für ein typisierbares Element.

- Welcher Wertebereich wird von einer Multiplizität beschrieben?
- Was ist der Unterschied zwischen Multiplizität und Kardinalität?

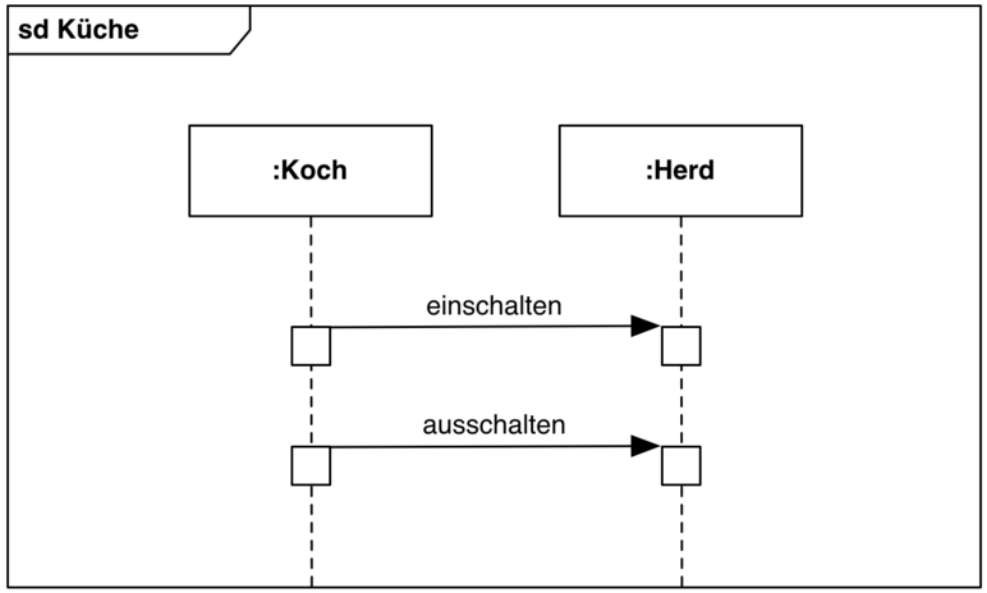
Agenda

MHP
A PORSCHE COMPANY

1. Aufbau von Anwendungssystemen
2. **Methoden der SW Technik & UML**
   - Allgemeine Grundlagen UML
   - Anwendungsfall-/ Use Case Diagramme
   - Aktivitätsdiagramme
   - Objekt- und Paketdiagramme
   - Komponentendiagramme
   - Klassendiagramme
   - **Sequenzdiagramme**
   - Objektorientierte Methode

Definition

- Sequenzdiagramme beschreiben die Kommunikation zwischen Objekten in einer bestimmten Szene.
- Es wird beschrieben welche Objekte an der Szene beteiligt sind, welche Informationen (Nachrichten) sie austauschen und in welcher zeitlichen Reihenfolge der Informationsaustausch stattfindet.
- Sequenzdiagramme enthalten eine implizite Zeitachse. Die Zeit schreitet in einem Diagramm von oben nach unten fort. Die Reihenfolge der Pfeile in einem Sequenzdiagramm gibt die zeitliche Reihenfolge der Nachrichten an.



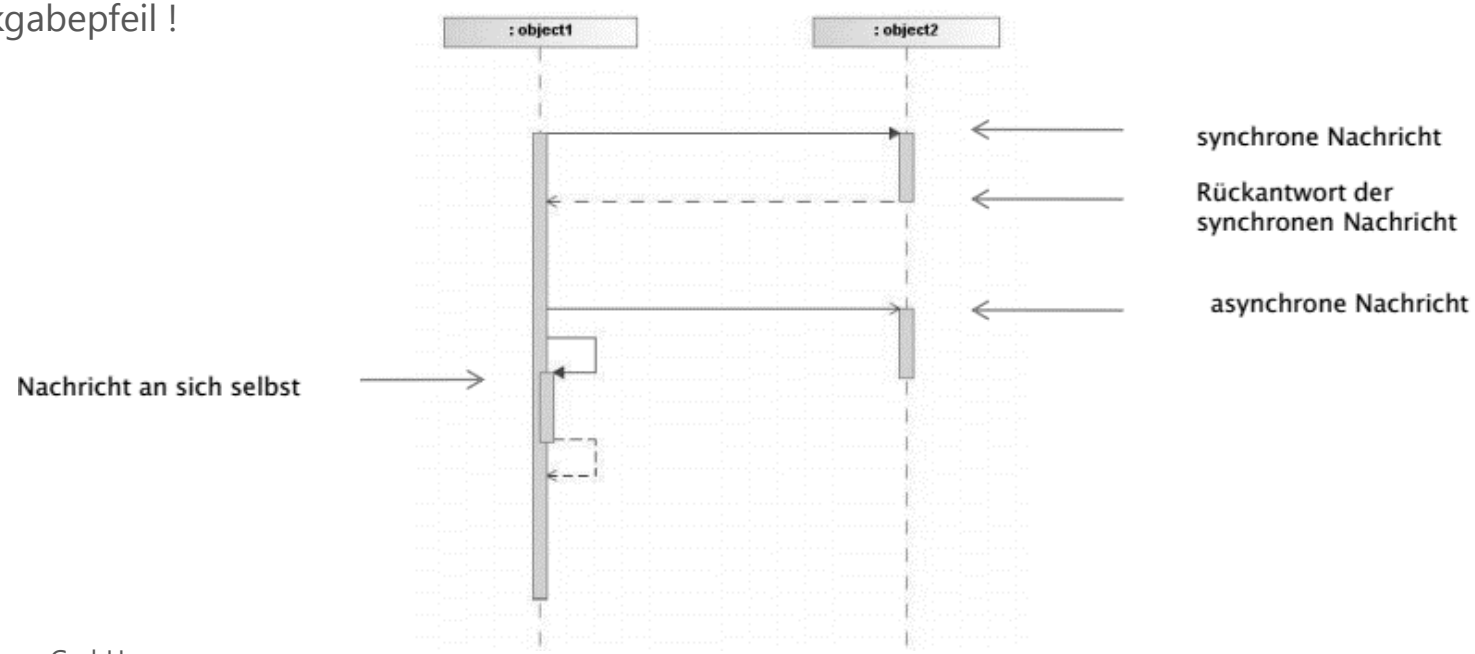
Notation und Semantik

Synchrone Nachricht

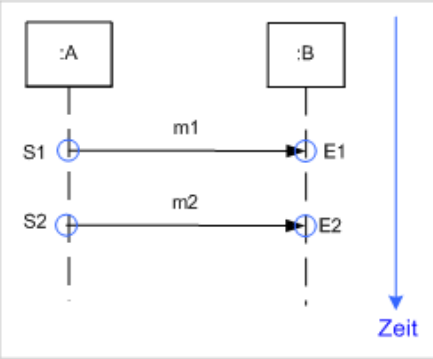
- Sender wartet, bis Empfänger die Nachricht abgearbeitet hat
- Gestrichelter Pfeil für Rücksprung zum Sender

Asynchrone Nachricht

- Sender wartet nicht auf Empfänger und arbeitet unmittelbar weiter
- Sender und Empfänger befinden sich in unterschiedlichen Ausführungsprozessen
- Kein gestrichelter Rückgabepfeil !

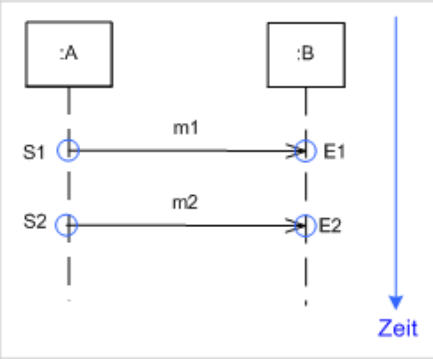


Notation und Semantik – Zeitliche Ordnung



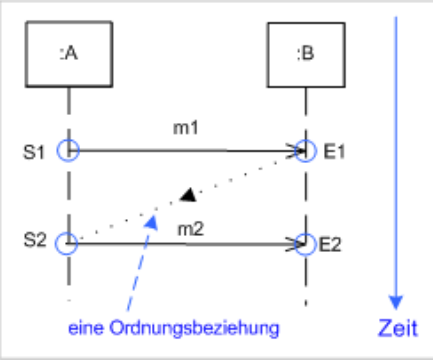
Synchrone Aufrufe

Reihenfolge: <S1, E1, S2, E2>



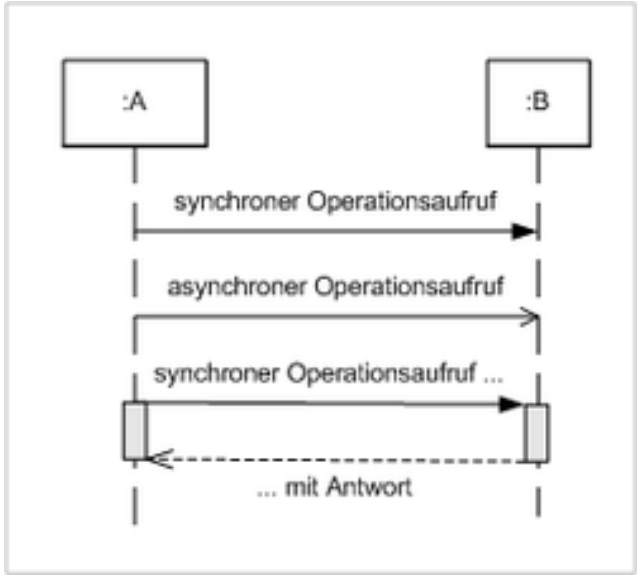
Asynchrone Aufrufe

Reihenfolge: <S1, E1, S2, E2> oder
<S1, S2, E1, E2> oder <S1, S2, E2, E1>



Reihenfolge: <S1, E1, S2, E2>

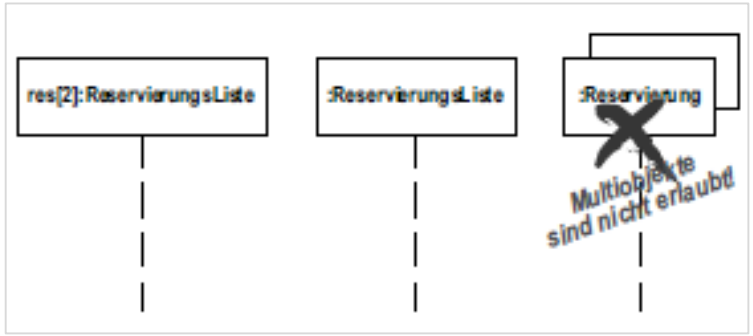
Notation und Semantik



- In der Abbildung: 2 „Lebenslinien“

- Wenn zwischen Lebenslinien Nachrichten ausgetauscht werden, muss auch ein Verhalten in den zugehörigen Elementen ausgeführt werden. Das wird durch die länglichen Rechtecke auf der Lebenslinie dargestellt. Die Rechtecke repräsentieren den so genannten Ausführungsfokus.

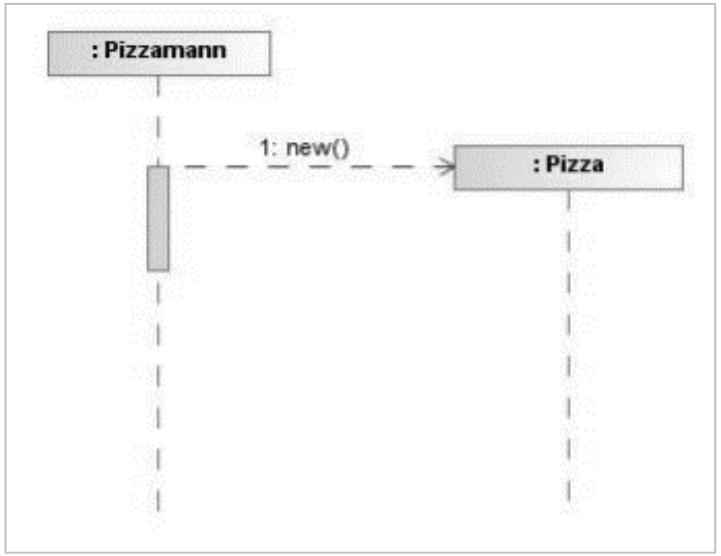
- Multiobjekte sind nicht erlaubt



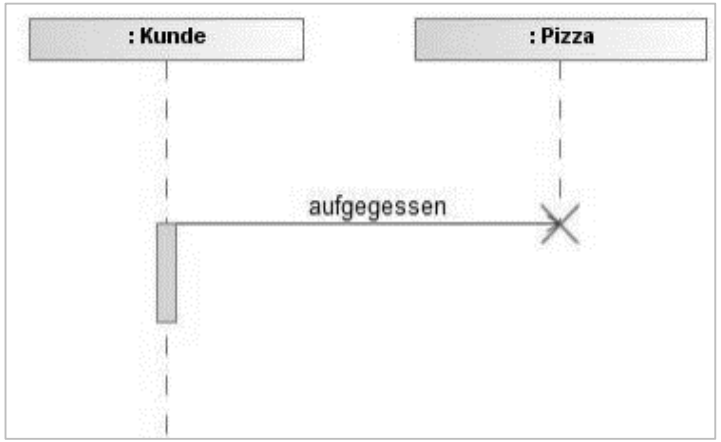
- Im Kopf einer Lebenslinie kann auch das Schlüsselwort *self* stehen. Diem Lebenslinie repräsentiert dann ein Exemplar des Classifiers, zu dem die Interaktion gehört.

Notation und Semantik - Nachrichten

- Erzeugen von neuen Objekten

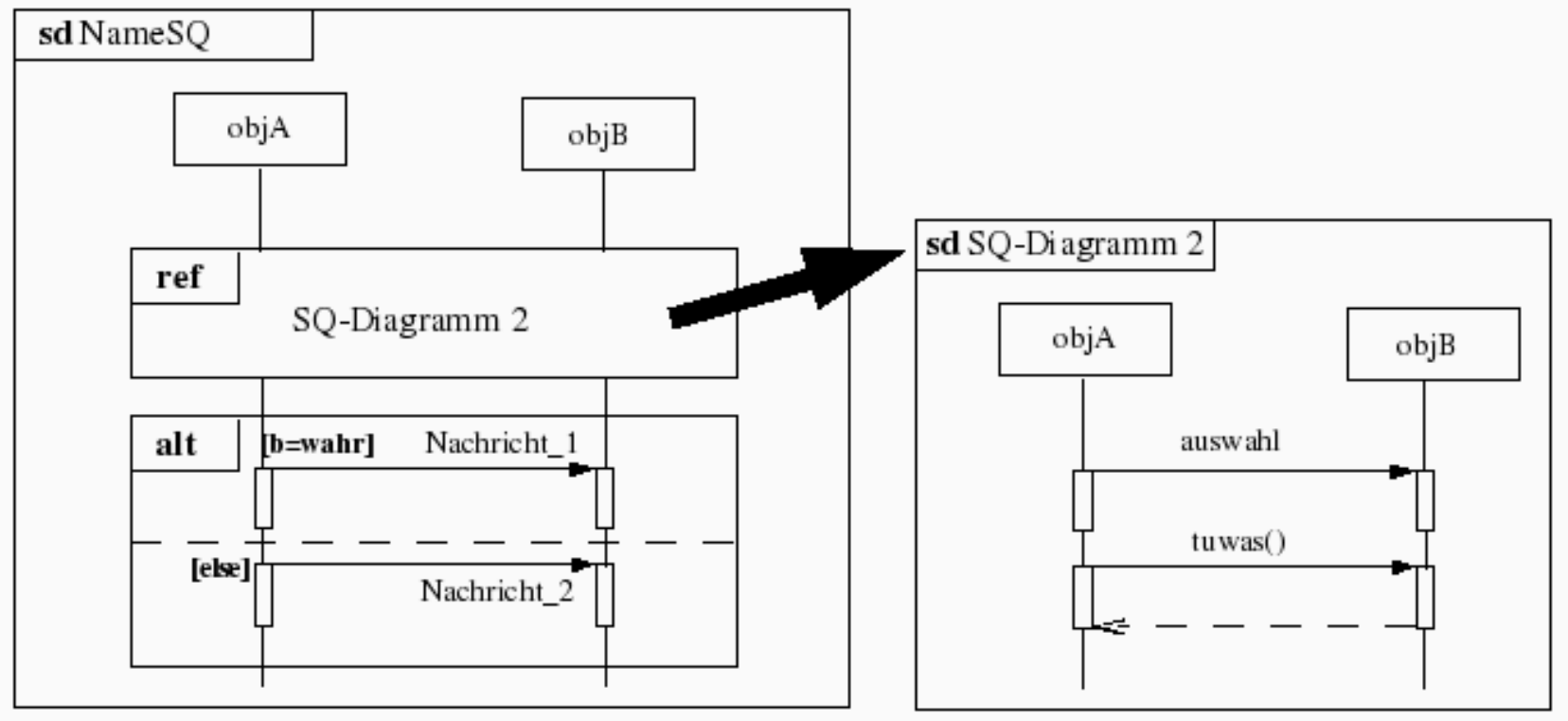


- Zerstören/Löschen von Objekten



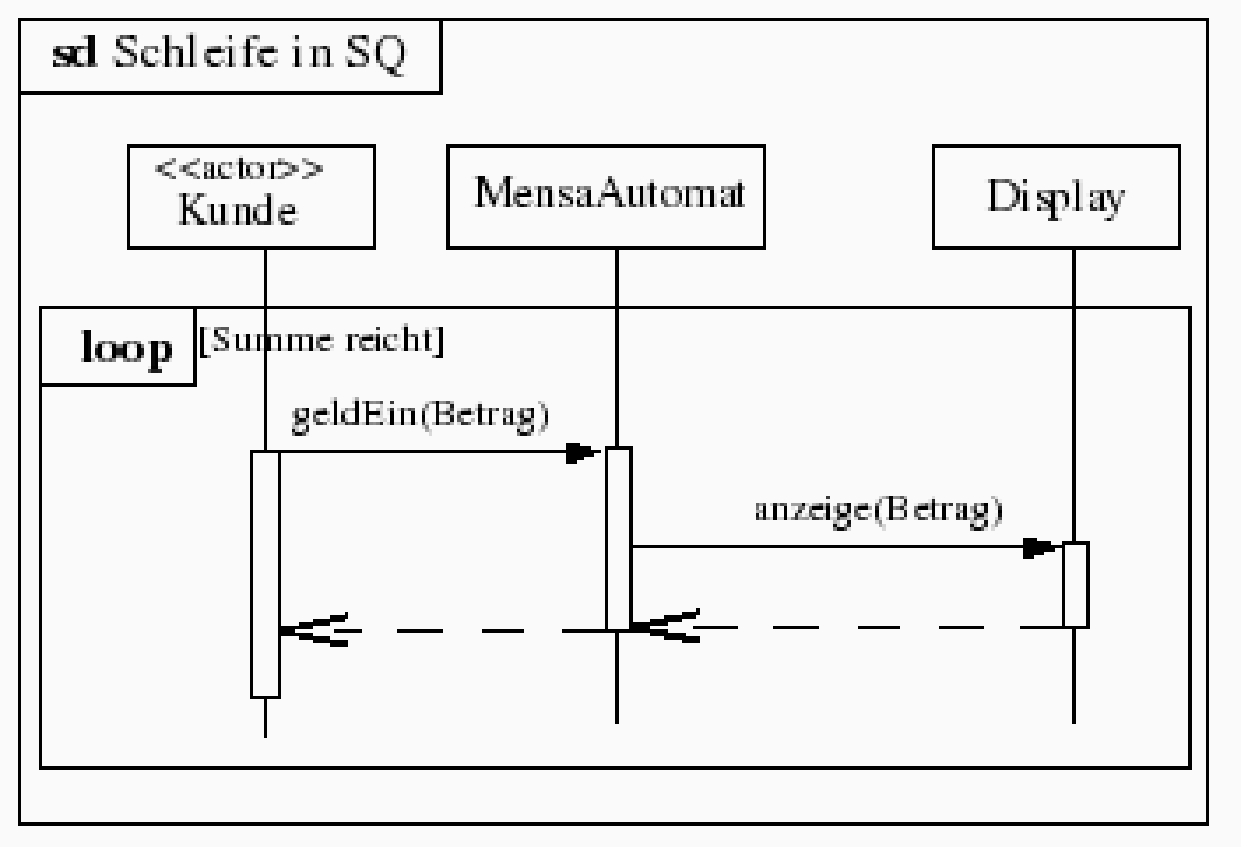
Notation und Semantik – Verweis & Alternative darstellen

- If/Else-Darstellung sowie Referenz abbilden:



Notation und Semantik – Schleife darstellen

- Loop/Schleife abbilden:



Übung

Kurzbeschreibung

Case: Benutzer anlegen

1. Der WiBe-Beauftragte wählt die Operation „Benutzer, Neu …“ aus einem durch die Benutzerschnittstelle zur Verfügung gestellten Menü.
2. Das System präsentiert einen Dialog, in dem die Benutzerdaten (z. B. Name, Email-Adresse, Rolle, Passwort, Passwortwiederholung, etc.) eingegeben werden können.
3. Das System prüft die korrekte Eingabe aller Daten. Dabei wird auf Vorhandensein und Korrektheit bzgl. des Formats der Eingabe und der Verträglichkeit mit den Zieldatentypen getestet.
4. Die erfassten Benutzerdaten werden durch das System dem WiBe-Beauftragten in einer Übersicht zur Kontrolle dargestellt. Findet er einen Fehler, springen wir zu Punkt 2 zurück.
5. Ansonsten legt das System daraufhin entsprechende Datenobjekte und Einträge in der Datenbank ab.

>> Bitte erstellt für diesen Ablauf ein Sequenzdiagramm

Zusammenfassung:

MHP
A PORSCHE COMPANY

Checkliste Sequenzdiagramme:

- Wie sind gerichtete Beziehungen definiert?
- Können gerichtete Beziehungen mehr als ein Quell- bzw. Zielelement haben?

- Kann für ein Paketimport ein Alias definiert werden?
- Was ist der Unterschied zwischen privatem und öffentlichem Import?
- Welche Art von Elementen können mit dem Elementimport importiert werden?

- Nennen Sie ein Beispiel für einen Typ.
- Nennen Sie ein Beispiel für ein typisierbares Element.

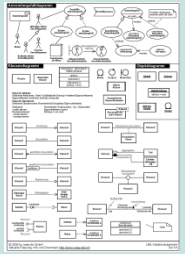
Agenda

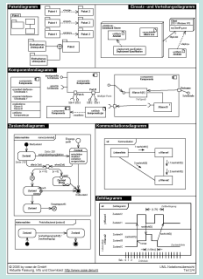
MHP
A PORSCHE COMPANY

1. Aufbau von Anwendungssystemen
2. **Methoden der SW Technik & UML**
   - Allgemeine Grundlagen UML
   - Anwendungsfall-/ Use Case Diagramme
   - Aktivitätsdiagramme
   - Objekt- und Paketdiagramme
   - Komponentendiagramme
   - Klassendiagramme
   - Sequenzdiagramme
   - **Objektorientierte Methode**

Notation, Prozess und Methode

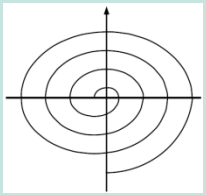
Notation - UML





Prozess (Vorgehensmodell):

- zu erfüllende Aktivitäten,
- deren zeitliche Abhängigkeiten,
- deren Resultate (Dokumente, Code)



Fokus: "Was und Wann"

Objektorientierte Methode

Fokus: "Wie"

bestimmt zusätzlich für die Aktivitäten Analyse und Entwurf eines Prozesses:

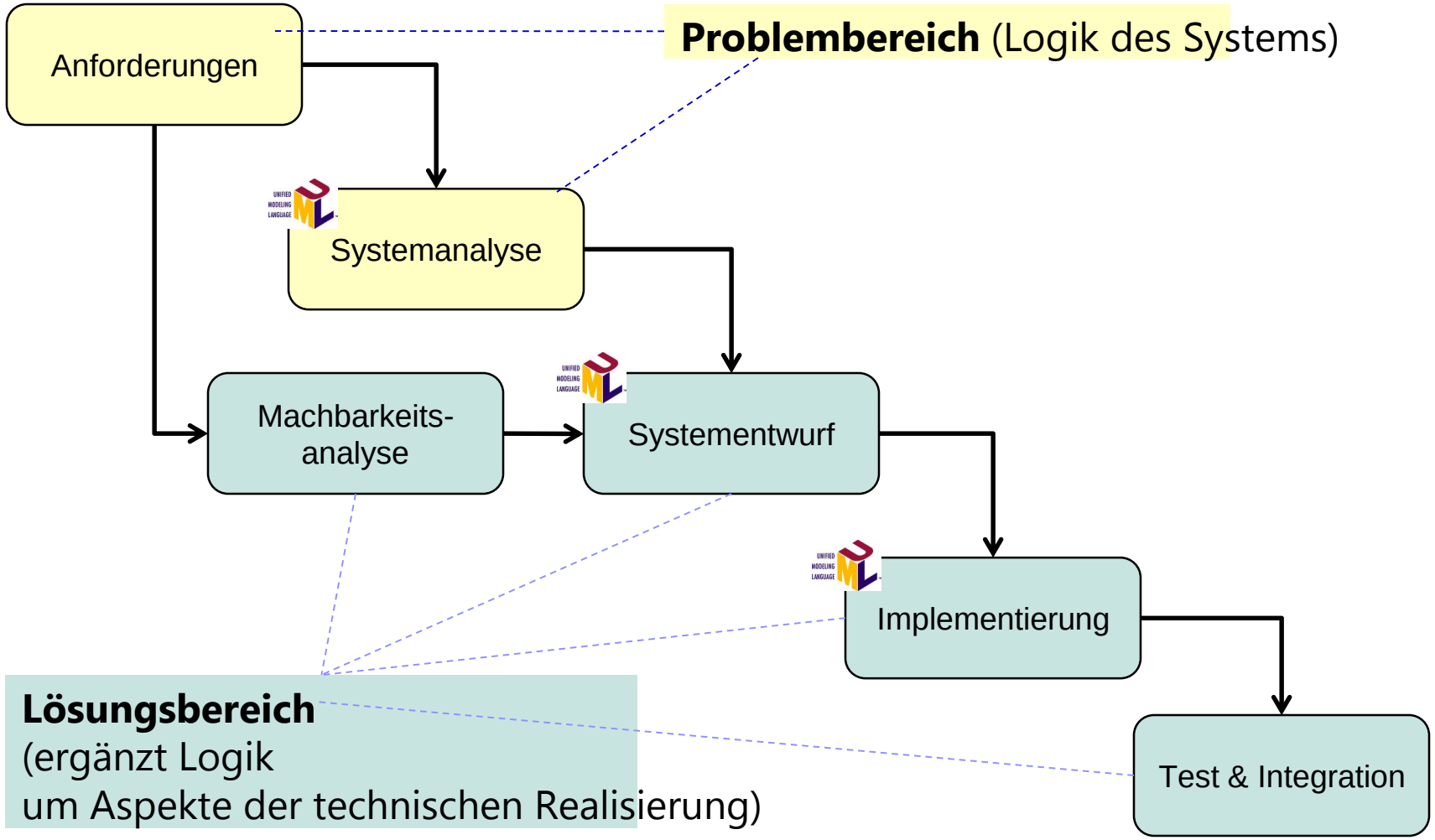
- die einzelnen Schritte, um die Resultate zu erhalten
- die Verwendung der Sprachelemente aus der Notation für jeden Schritt.

Je nach

- Projekt (klein, groß, Informationssystem, Realzeitsystem, ...),
- Anwendungsdomäne (Automotiv, Banken, ...),
- Vorkenntnissen der Projektmitarbeiter

entscheidet der Systemanalytiker/-architekt, welche **Methode** zum Einsatz kommt.

Problembereich und Lösungsbereich



Systemanalyse und Systementwurf

MHP
A PORSCHE COMPANY

Systemanalyse

- befasst sich mit dem **Problembereich**
- d.h. der **fachlichen Logik** der Anwendung
- analysiert werden **Geschäftsprozesse**
- Anwendungsfälle werden identifiziert und ausgearbeitet.

Systementwurf

- befasst sich mit dem **Lösungsbereich**
- d.h. mit der **technischen Realisierung**
- die Logik aus der Systemanalyse wird in ein **lauffähiges System** gegossen

Werden in der Systemanalyse nicht sorgfältig die Geschäftsprozesse analysiert und die **richtigen Anwendungsfälle** identifiziert, läuft man Gefahr, ein System zu bauen, das der Anwender nicht haben wollte, oder nicht gebrauchen kann.

Schritte in der Objektorientierten Systemanalyse

MHP
A PORSCHE COMPANY

1. Überprüfen der Anforderungen
2. Spezifizieren der Geschäftsprozesse
3. Priorisieren der Anforderungen
4. Erstellen des Kontextdiagramms (Systemgrenzen festlegen)
5. Anforderungen neu definieren (Pflichtenheft)
6. Erstellen des Anwendungsfalldiagramms
7. Kurzbeschreibung der Anwendungsfälle
8. Finden von Klassen und Erstellen des Klassendiagramms der konzeptionellen Sicht
9. Langbeschreibung der Anwendungsfälle
10. Erstellen der Kommunikationsdiagramme für jeden Anwendungsfall
11. Erstellen des Klassendiagramms der Verarbeitungssicht
12. Festlegen der Abhängigkeitsbeziehungen
13. Erstellen des Klassendiagramms der finalen Sicht der Systemanalyse

Schritte im Objektorientierten Systementwurf

MHP
A PORSCHE COMPANY

1. Ableiten des Schichtenmodells (inkl. Entwurfsspezifische Schichten)
2. Vervollständigen des dynamischen Verhaltens aus der Systemanalyse
3. Entwurfsspezifische Kollaborationen ergänzen
4. Kollaborationen optimieren und Entwurfsmuster identifizieren
5. Abhängigkeitsbeziehungen ermitteln
6. Klassendiagramm des Entwurfs zeichnen

Alle Schritte aus Systemanalyse und Systementwurf werden anhand der Fallstudie „Bücherverwaltung“ in den nächsten Vorlesungen durchlaufen.

MHP
A PORSCHE COMPANY

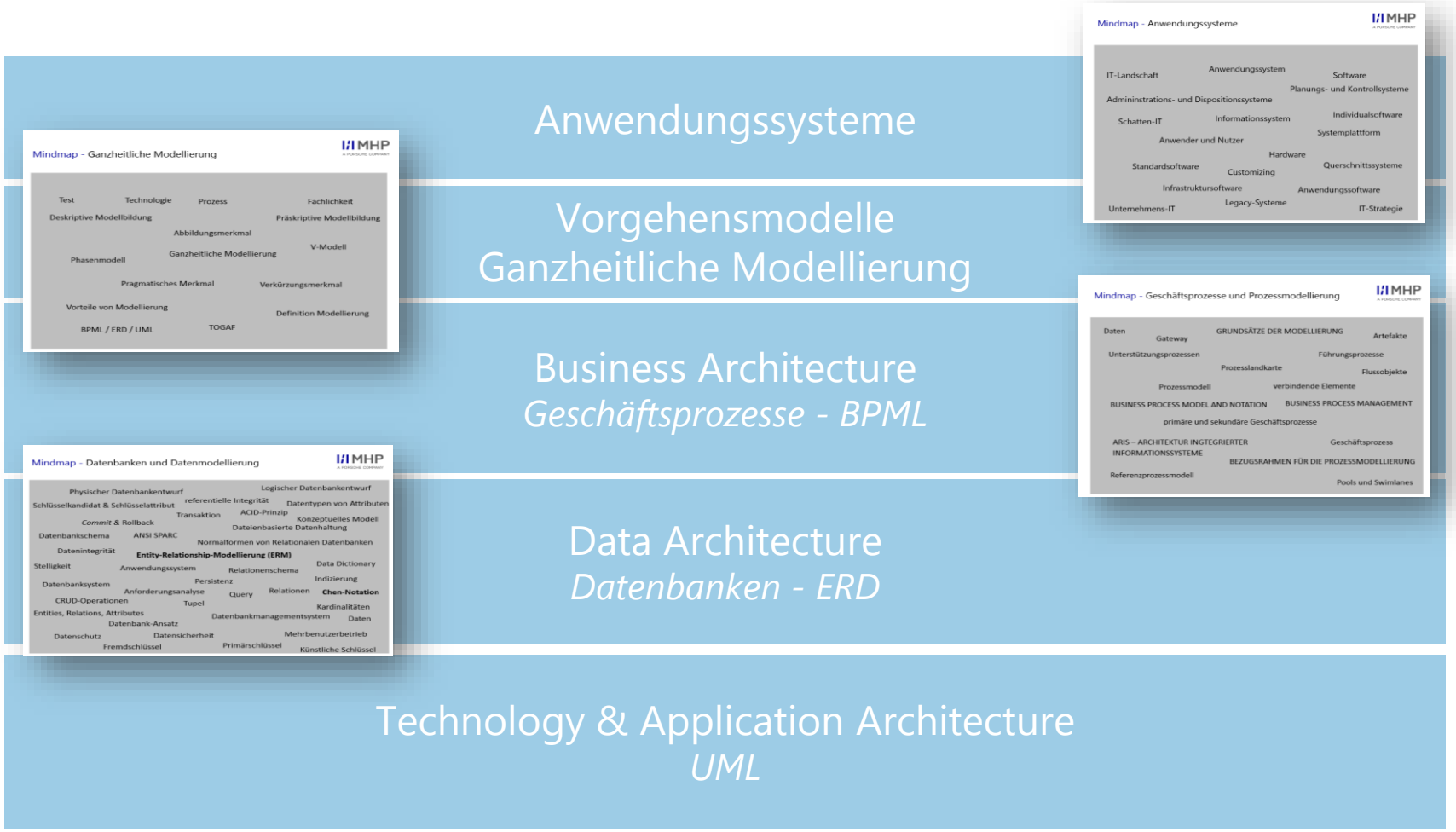
INTEGRIERTE MODELLIERUNG KOMPLEXER SYSTEME

Zusammenfassung und Literatur

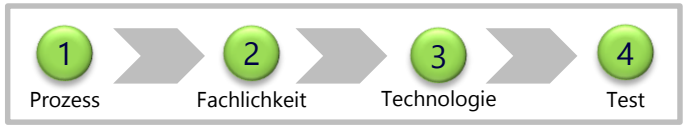
UNIVERSITATEA BABEȘ-BOLYAI

Hugo Colceag I Bachelor Studiengang Informatik

Vorlesungsinhalte und Aufbau



Überführung der Informationen ins Modell



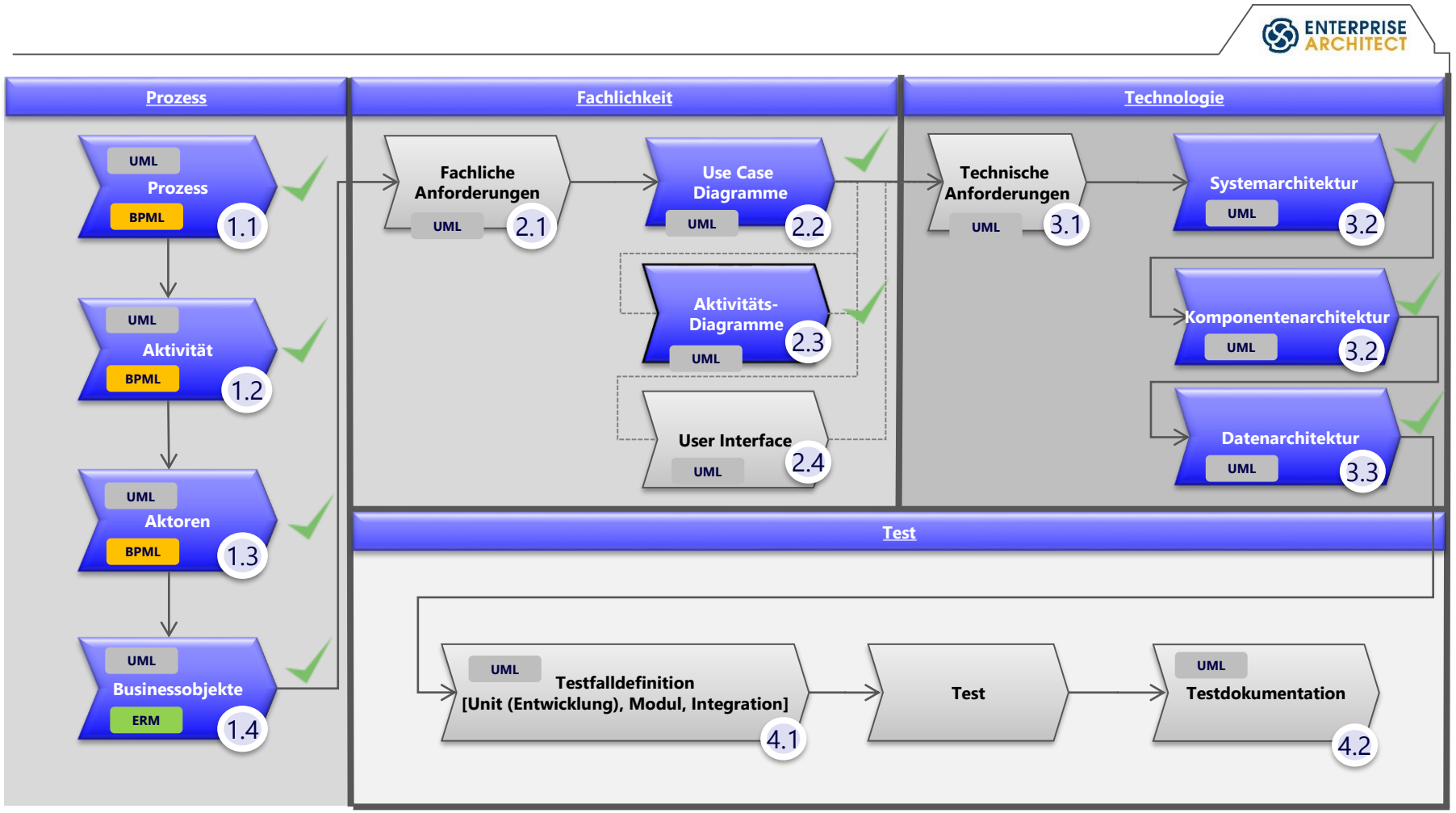
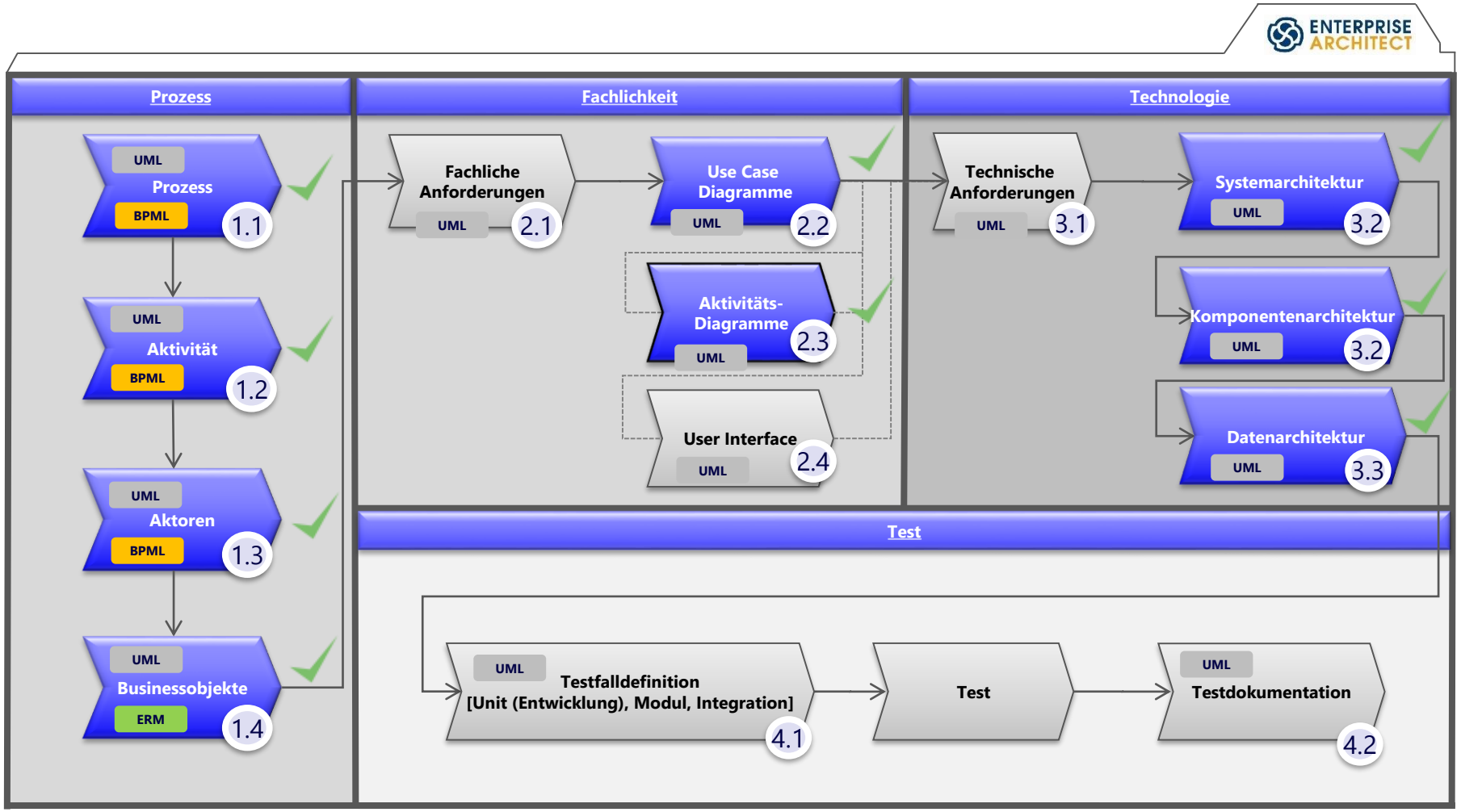
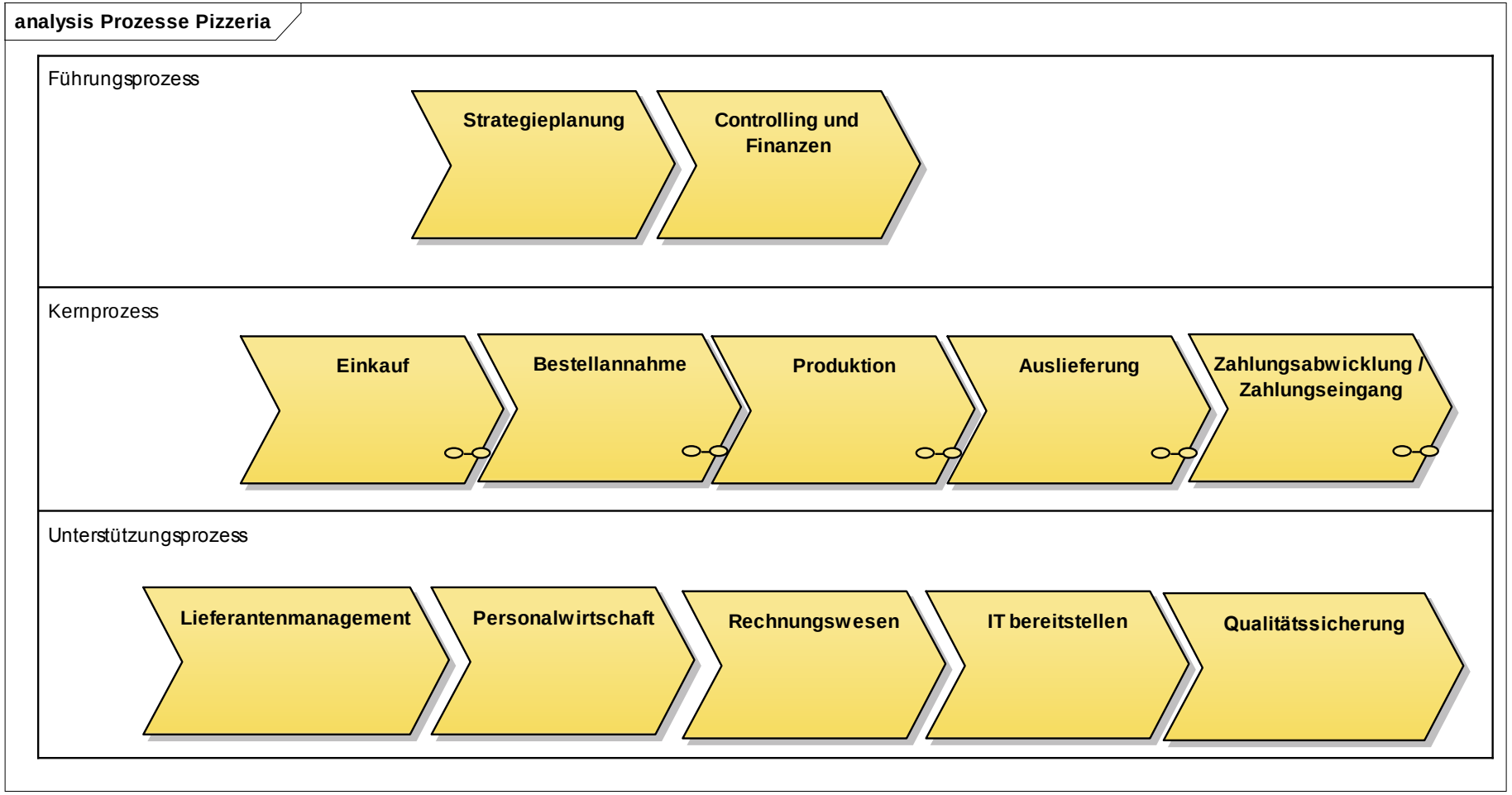


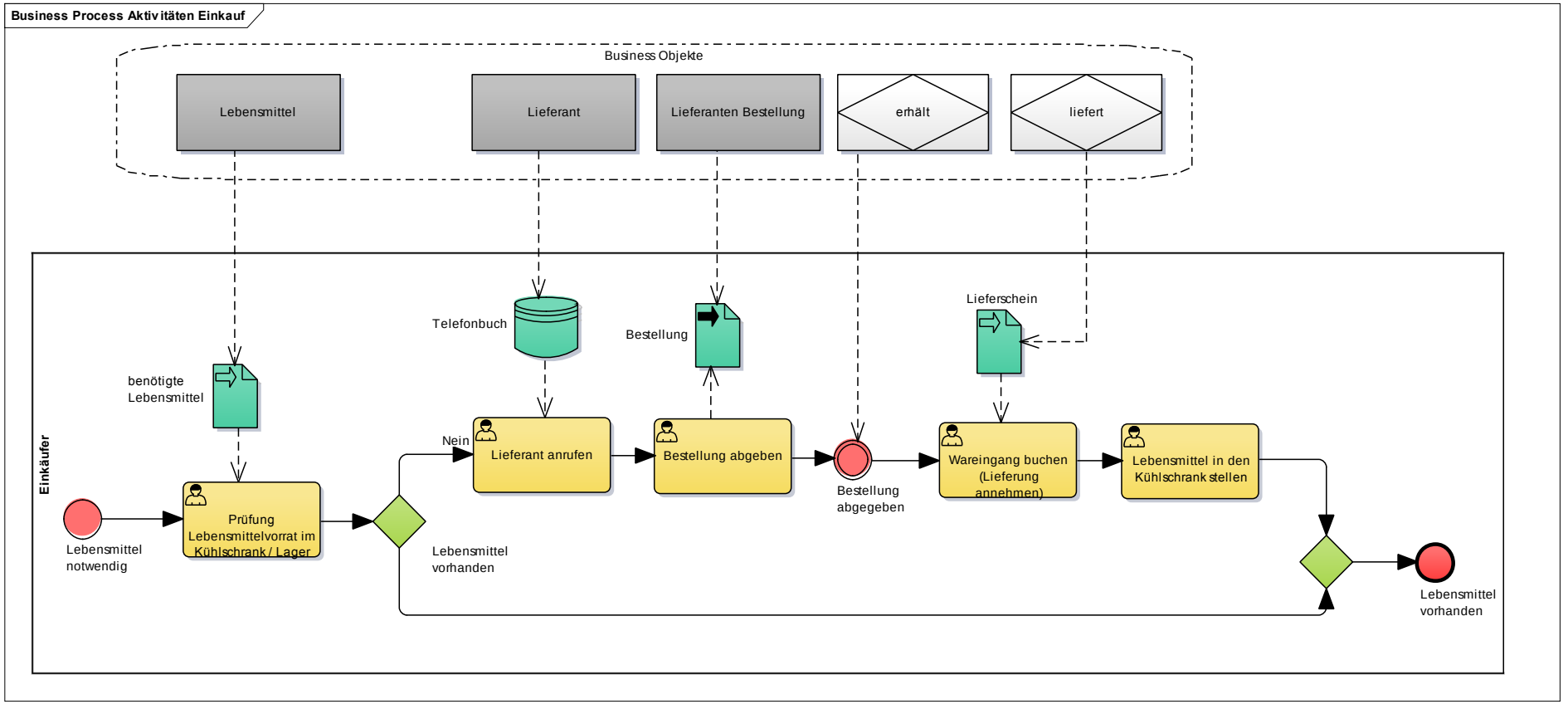
Diagramme aus der Vorlesung



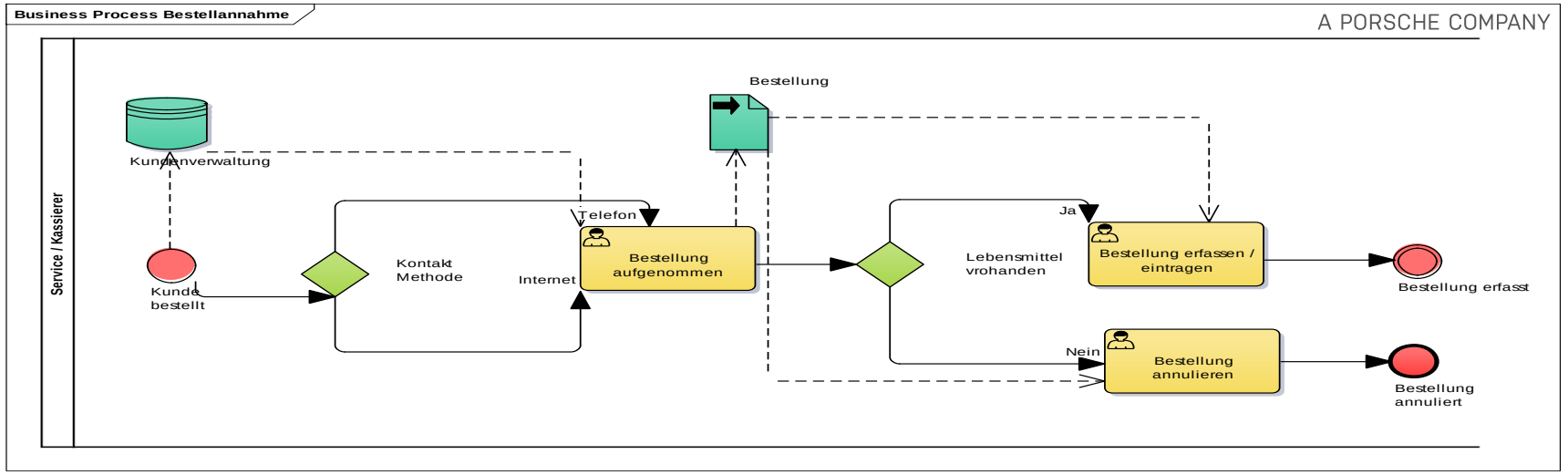
Prozesslandkarte

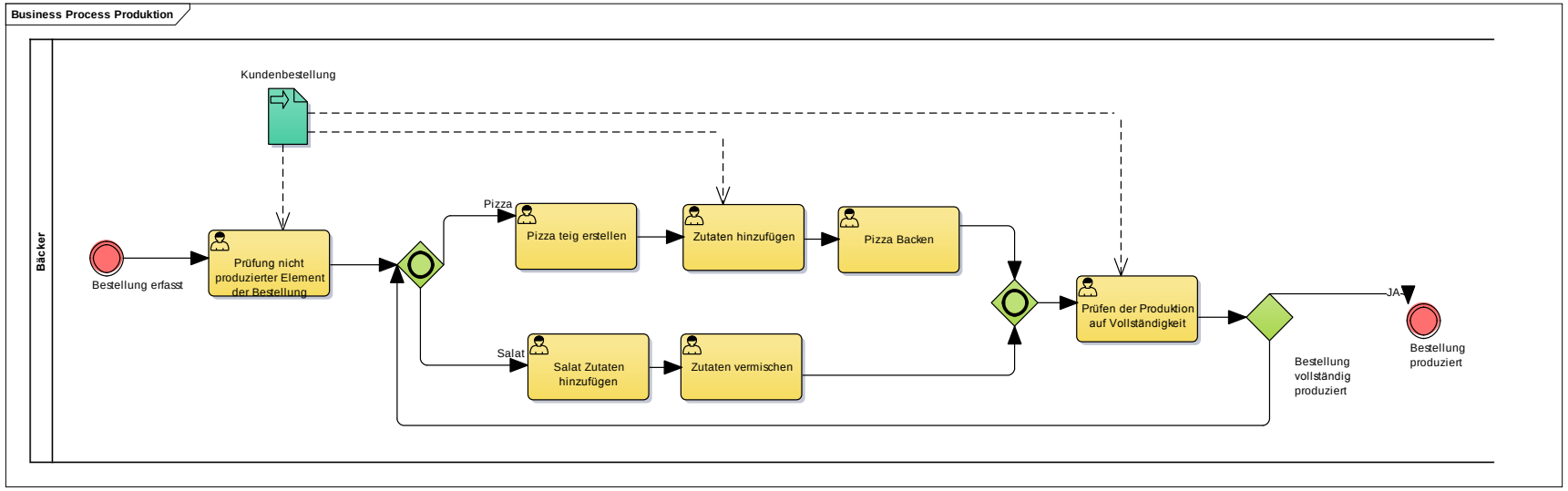


Business Process Modelling

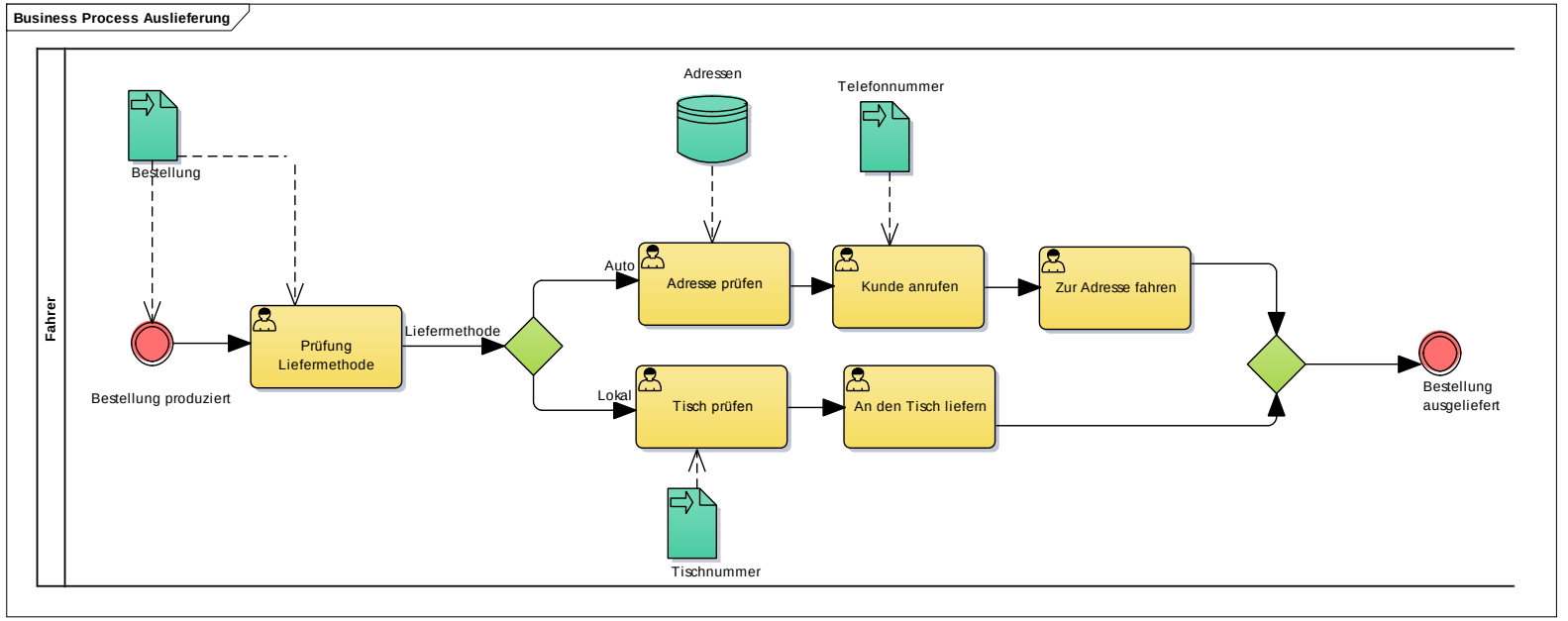


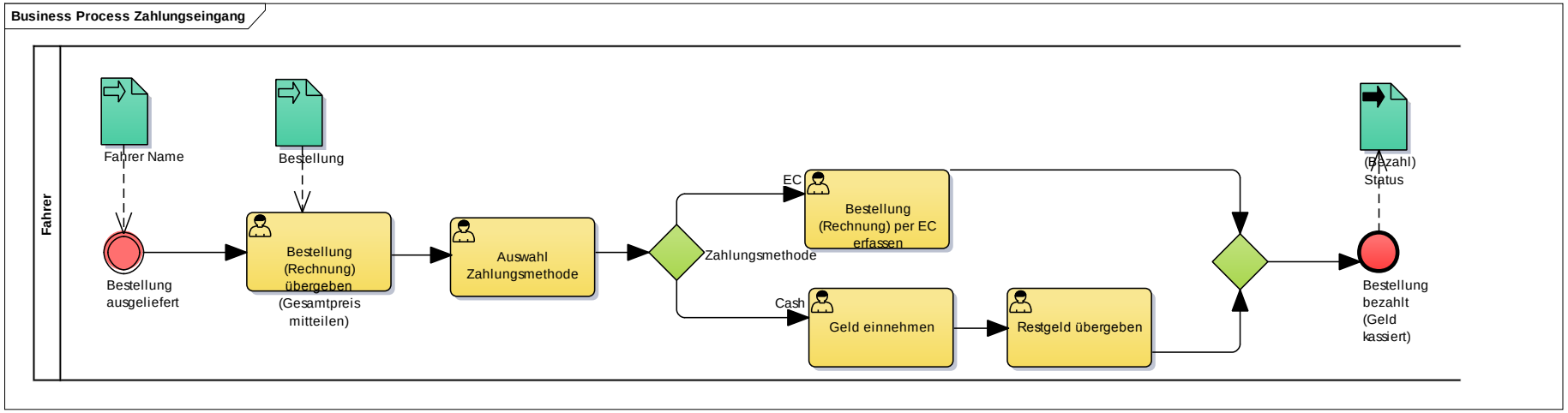
Business Process Modelling



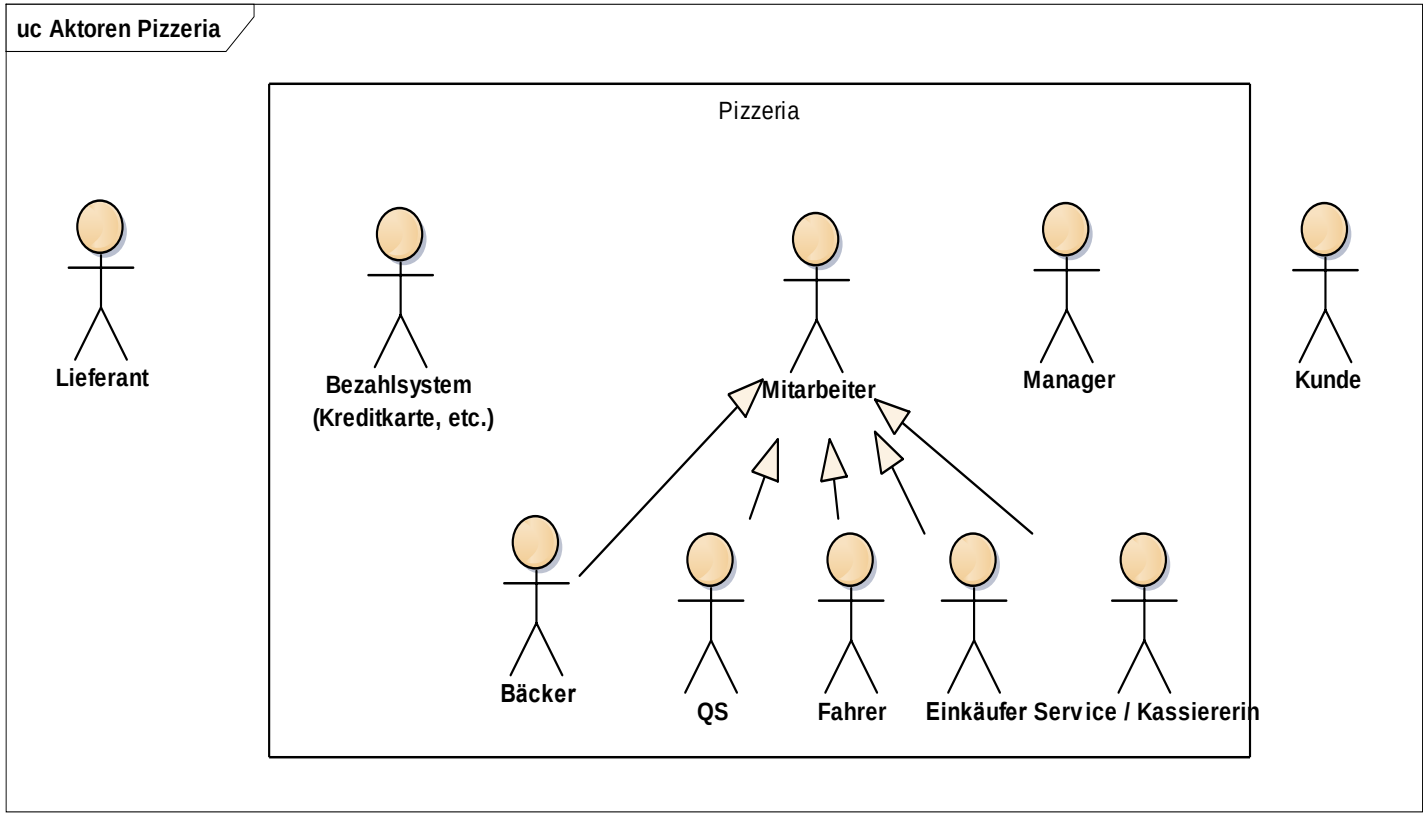


Business Process Modelling

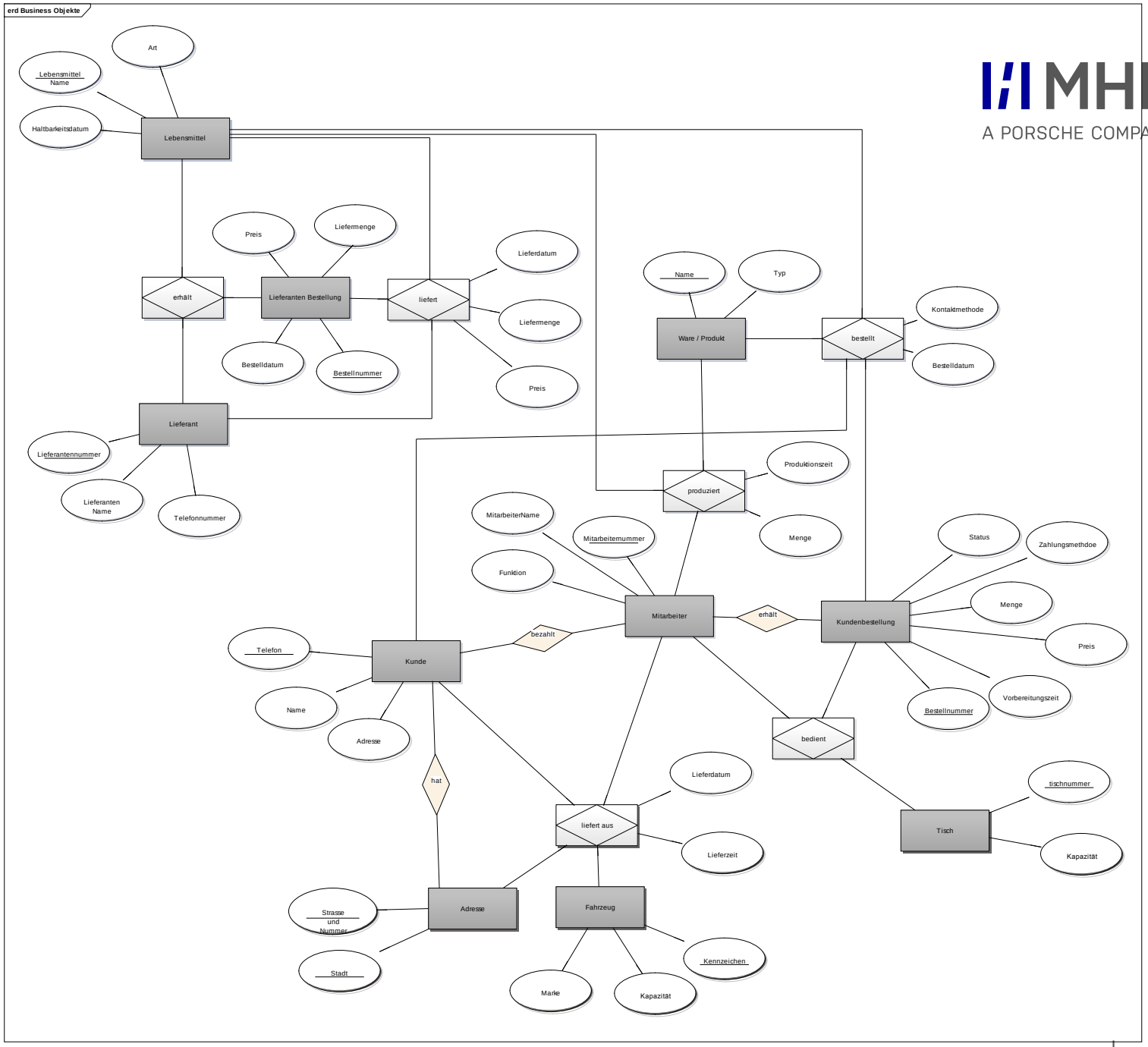




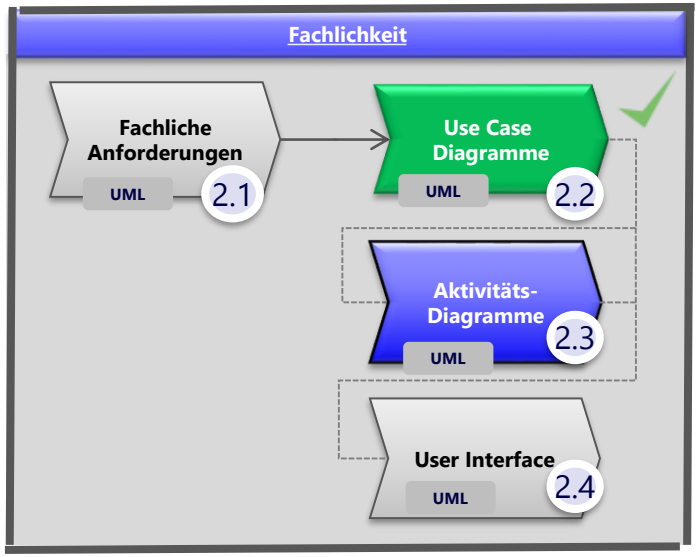
Aktoren Diagramm

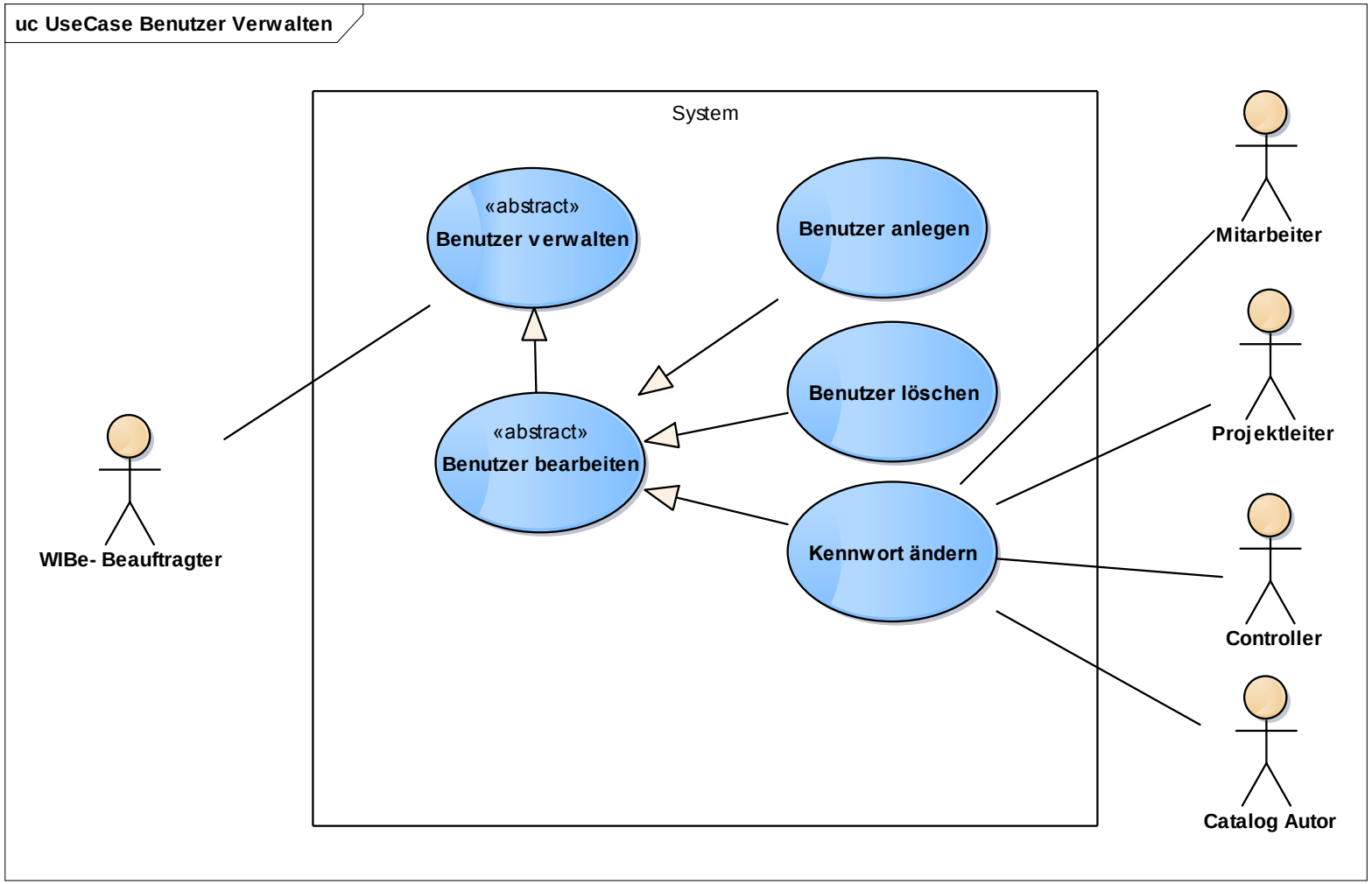


Entity Relationship Model

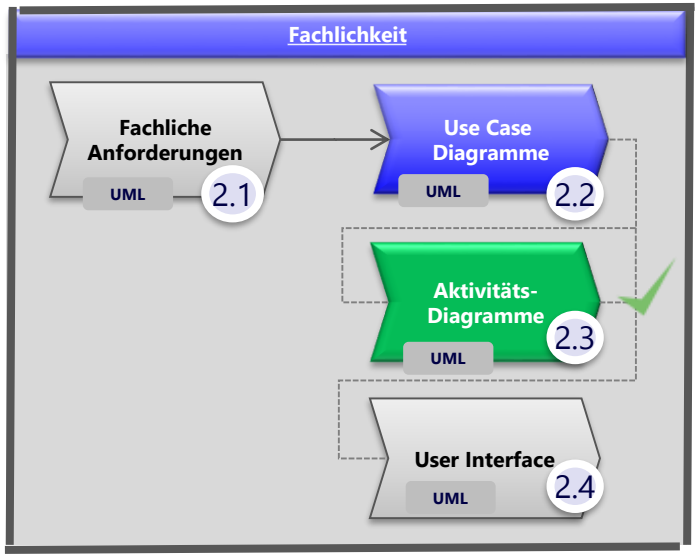


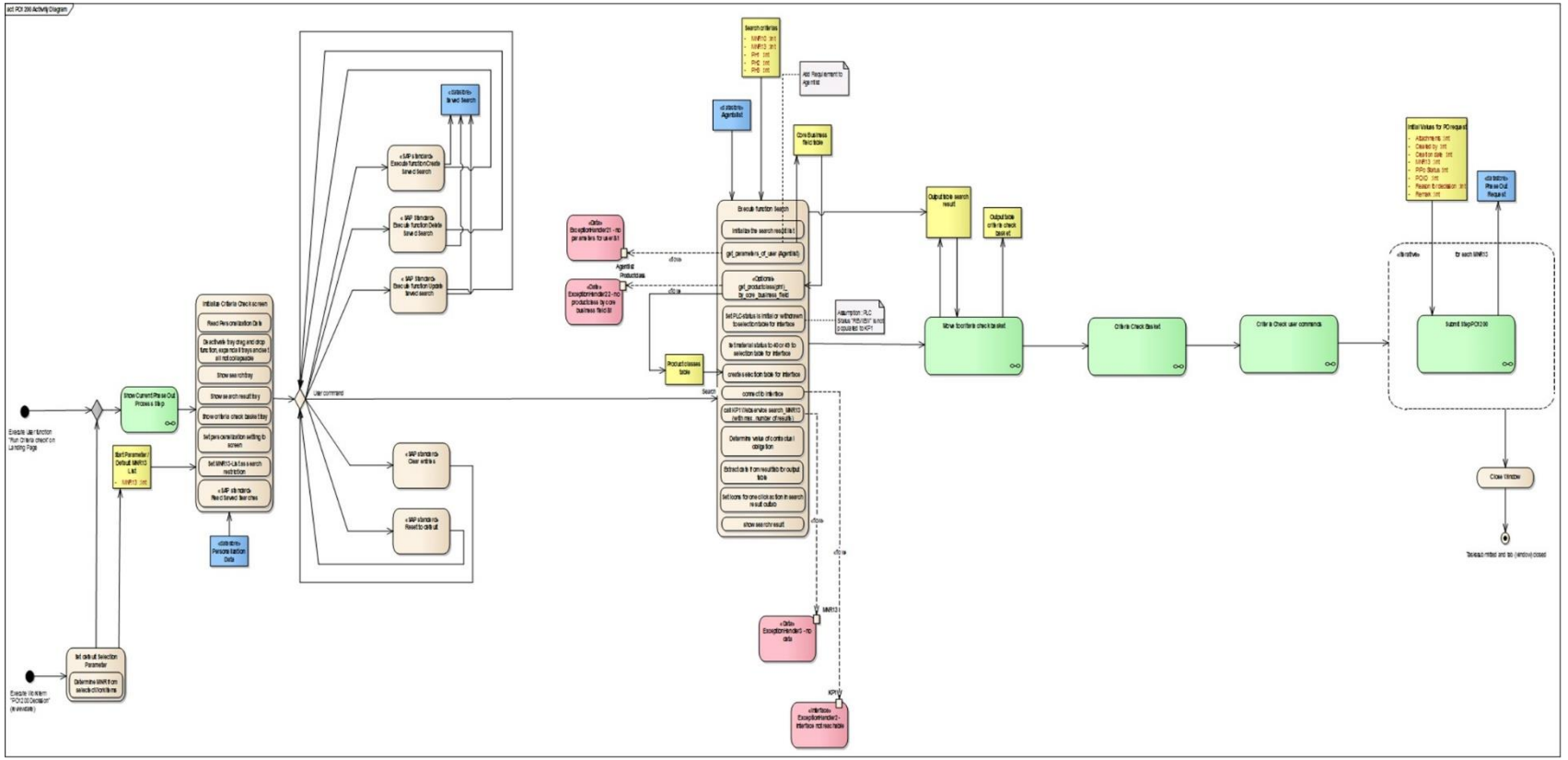
Use Case Diagramm



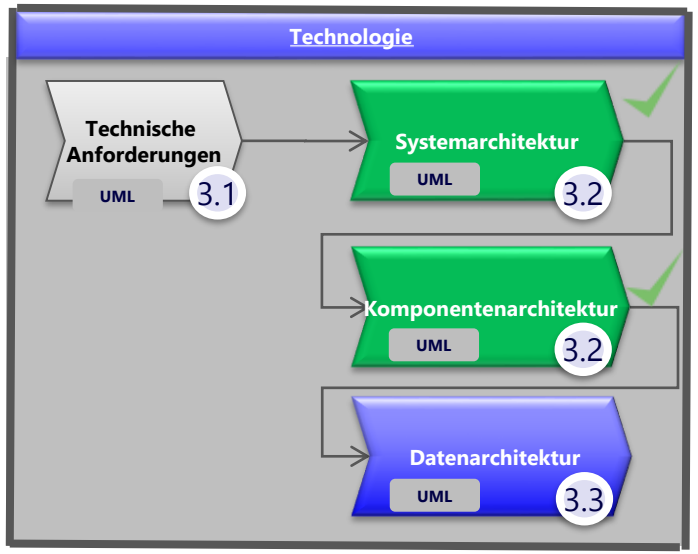


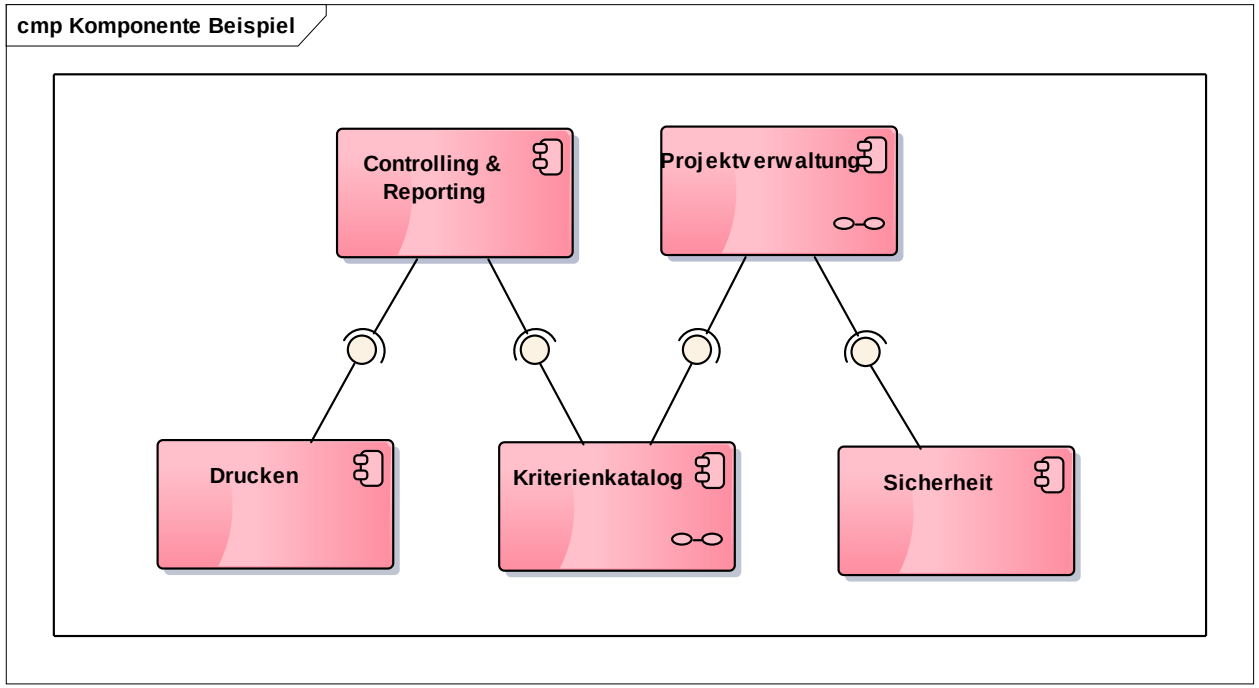
Aktivitätsdiagramm

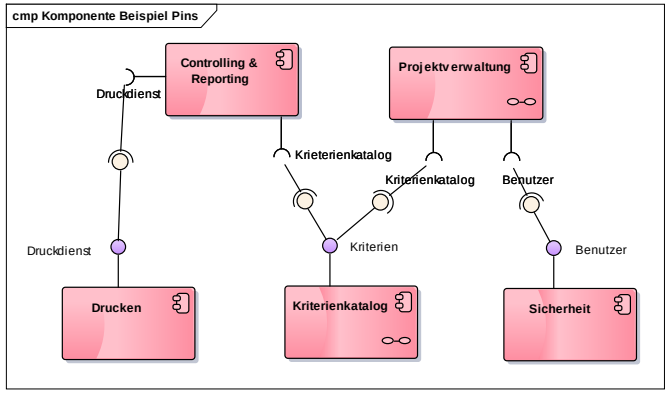




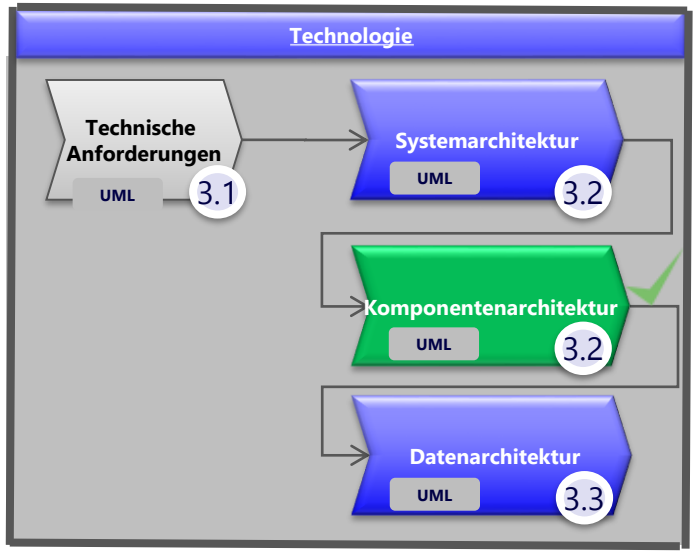
Komponenten Diagramm

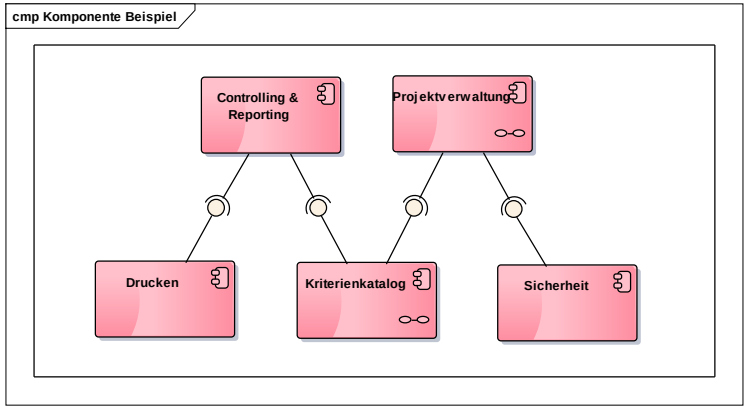


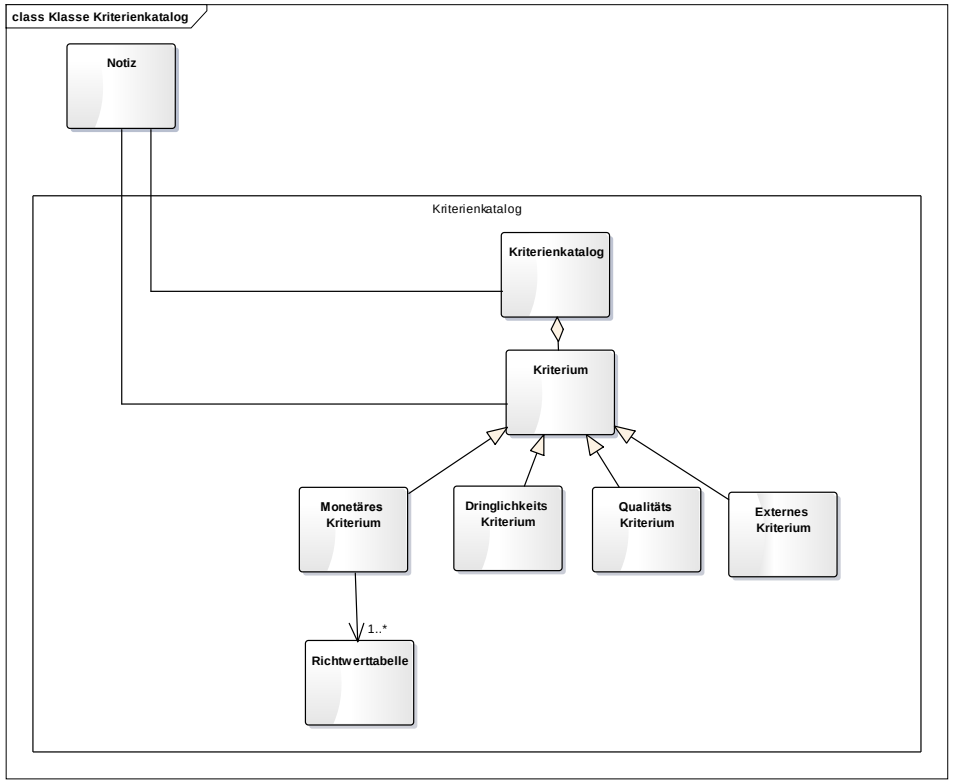


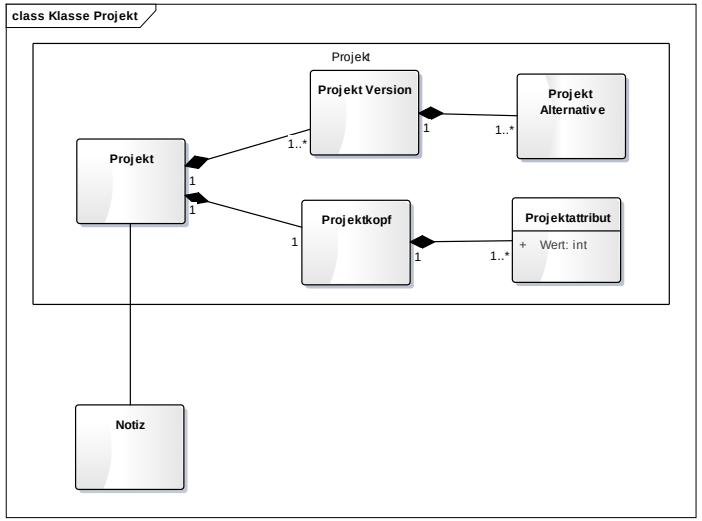


Klassen Diagramm

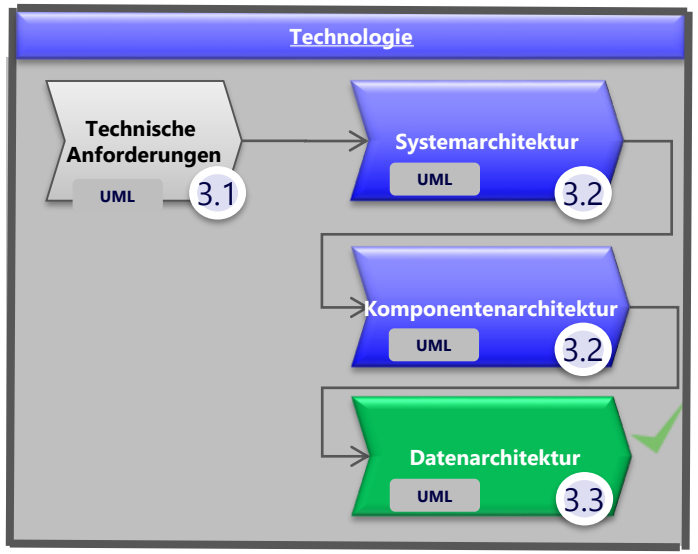


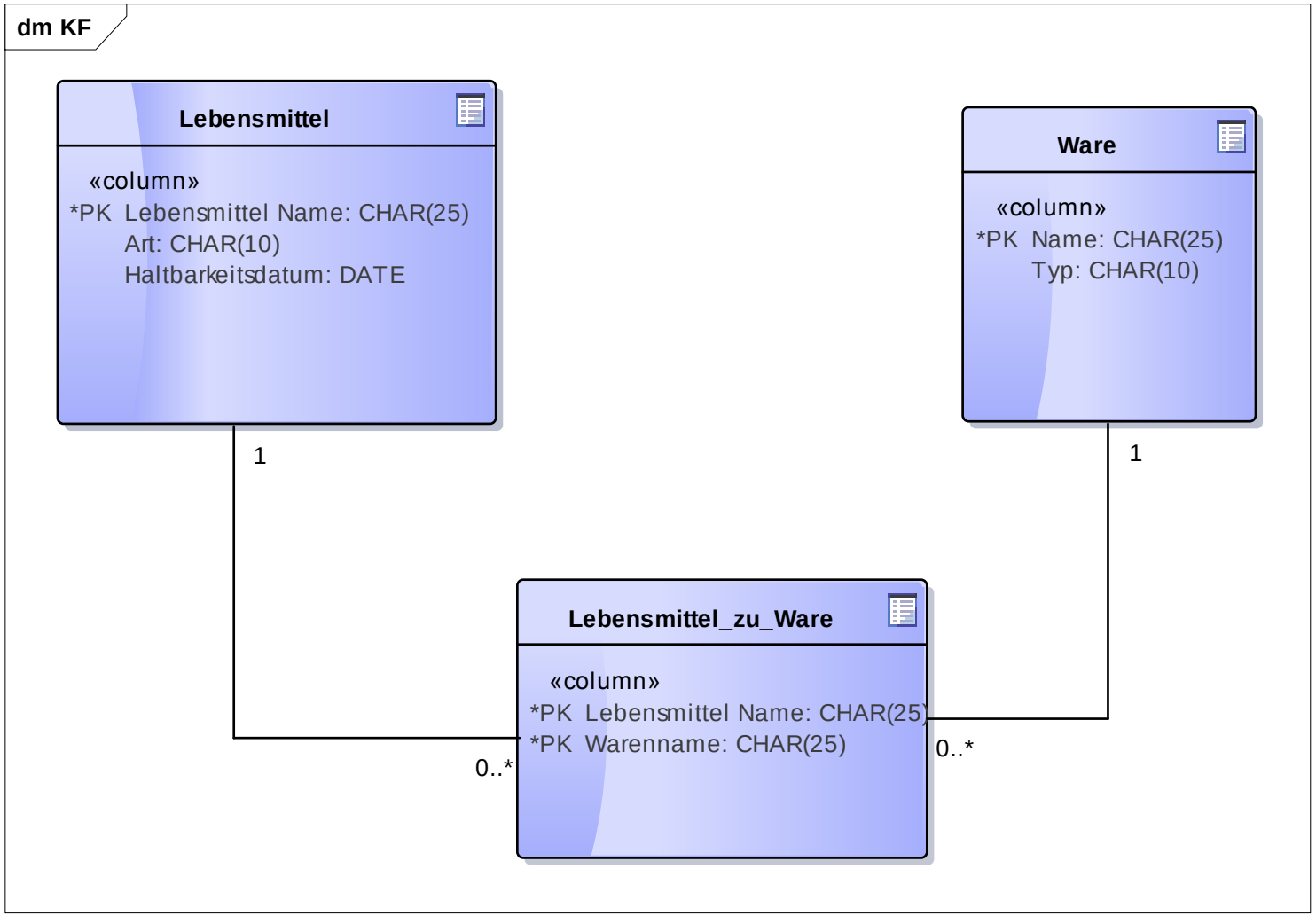






Logisches Datenmodell





Agenda

MHP
A PORSCHE COMPANY

1. Literatur

Literatur

- Abschnitte Anwendungssysteme, Datenbanken sowie Prozessmodellierung sind auf Basis des Skriptes der Fachhochschule Kempten, Studiengang Maschinenbau 2015/ 2016 der Autorin Prof. Dr. Irene Weber erstellt worden

- Abts, D., & Mülder, W. (2011). *Grundkurs Wirtschaftsinformatik: Eine kompakte und praxisorientierte Einführung.* Vieweg.

- Atos Origin. (7. Februar 2011). *Atos Origin sets out its ambition to be a zero email company within three years.* Abgerufen am 2. März 2014 von Atos.net: http://atos.net/en-us/home/we-are/news/press-release/2011/pr-2011_02_07_01.html

- Atos SE. (2014). *Über Atos.* Abgerufen am 2. März 2014 von de.Atos.net: http://de.atos.net/de-de/home/uber-uns.html

- Becker, J., & Rosemann, M. (1997). Die Grundsätze ordnungsmäßiger Modellierung - ein Ordnungsrahmen zur Komplexitätsbeherrschung in Prozeßmodellen. In H.-P. Lipp (Hrsg.), *Proceedings zur Tagung Workflow-Management in Geschäftsprozessen im Trend 2000.*, (S. 18 - 30). Schmalkalden.

Literatur

- BPMN Offensive Berlin. (2011). *BPMN 2.0 - Business Process Model and Notation.* (B. O. Berlin, Hrsg.) Abgerufen am 20. Mai 2014 von www.bpmn.de: http://www.bpmb.de/images/BPMN2_0_Poster_DE.pdf

- BT. (2014). *Art of Connecting: creativity and the modern CIO - Global Results.* (B. T. plc, Hrsg.) Abgerufen am 22. 11 2915 von http://www.globalservices.bt.com: https://www.globalservices.bt.com/static/assets/secure-docs/creativity-modern-cio/Creativityandthemodern-CIO-Global.pdf

- Buchta, D., Eul, M., & Schulte-Croonenberg, H. (2009). *Strategisches IT-Management - Wert steigern, Leistung steuern, Kosten senken* (3. Ausg.). Springer Gabler.

- Bughin, J., & Chui, M. (16. Januar 2013). *Evolution of the networked enterprise: McKinsey Global Survey results.* Abgerufen am 2. März 2014 von McKinsey & Company: Insights & Publications: www.mckinsey.com/insights/business_technology/evolution_of_the_networked_enterprise_mckinsey_global_survey_results

- Bundesgesetzblatt Jahrgang 2009 Teil I Nr. 54. (19. August 2009). *Gesetz zur Stärkung der Sicherheit in der Informationstechnik des Bundes.* Abgerufen am 21. 11 2015 von https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/BSI/bsiges2009_pdf.pdf?__blob=publicationFile&v=1

Literatur

- *Business Process Model and Notation*. (2014). Abgerufen am 23. März 2014 von http://de.wikipedia.org/: http://de.wikipedia.org/wiki/BPMN

- Capgemini. (28. Januar 2014). *IT-Trends-Studie 2014*. Abgerufen am 28. Februar 2014 von http://www.de.capgemini.com: http://www.de.capgemini.com/resource-file-access/resource/pdf/capgemini-it-trends-studie-2014.pdf

- Capgemini. (2015). *Studie IT-Trends 2015*. Von Capgemini IT Trends Magazin: http://mc.capgemini.de/magazin/it-trends/wp-content/uploads/sites/3/2014/01/IT-Trends-Studie-2015.pdf?659b33 abgerufen

- CSCM Forschungsgruppe Kooperationssysteme München. (8. November 2013). Vernetzte Organisation - Die Studie 2013. (A. Richter, & M. Koch, Hrsg.) Abgerufen am 2. März 2014 von vernetzte-organisation.de: http://www.kooperationssysteme.de/docs/pubs/Richter et al. 2013 - Vernetzte Organisation -Die Studie 2013.pdf

- Dittmar, C., Oßendoth, V., & Schulze, K.-D. (Mai 2013). *Business Intelligence - Status Quo in Europa. Business Intelligence Maturity Audit 2012/13*. (S. M. Consulting, Hrsg.) Abgerufen am 26. Februar 2014 von http://www.steria.com: http://www.steria.com/de/bi/bidm-einblicke/europaeische-bima-studie-201213/

Literatur

- Experton Group. (22. August 2013). *Der Markt für „Social Business for Collaboration & Communication" (SB4CC) in Deutschland.* (E. G. AG, Hrsg.) Abgerufen am 2. März 2014 von http://de.atos.net/content/dam/de/documents/Atos-Study-ExpertonGroup-SocialBusiness-2013.pdf

- Fink, A. u. (2005). *Grundlagen der Wirtschaftsinformatik.* Physica-Verlag, Springer.

- Freund, J., & Rücker, B. (2012). *Praxishandbuch BPMN 2.0* (3 Ausg.). München: Hanser.

- Gabriel, R. (30. September 2013). Entscheidungsunterstützungssystem. (7.). (K. Kurbel, J. Becker, N. Gronau, E. Sinz, & L. Suhl, Hrsg.) München: Oldenbourg. Abgerufen am 25. Februar 2014 von Enzyklopädie der Wirtschaftsinformatik - Online-Lexikon: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/uebergreifendes/Kontext-und-Grundlagen/Informationssystem/Entscheidungsunterstutzungssystem

- Gadatsch, A. (2010). *Grundkurs Geschäftsprozess-Management - Methoden und Werkzeuge für die IT-Praxis: eine Einführung für Studenten und Praktiker.* Wiesbaden: Vieweg + Teubner.

- Hanschke, I., & Lorenz, R. (2012). *Strategisches Prozessmanagement – einfach und effektiv.* Carl Hanser Verlag GmbH & Co. KG.

- Hanschke, I., Giesinger, G., & Goetze, D. (2013). *Businessanalyse - einfach und effektiv.* München: Hanser.

Literatur

- Hesseler, M., & Görtz, M. (2007). *Basiswissen ERP-Systeme: Auswahl, Einführung & Einsatz betriebswirtschaftlicher Standardsoftware.* Herdecke: W3L-Verlag.

- Holland, H., Lackes, R., Schewe, G., & Siepermann, M. (kein Datum). Stichwort: Funktion. Springer Gabler Verlag. Abgerufen am 7. März 2014 von Gabler Wirtschaftslexikon: wirtschaftslexikon.gabler.de/Archiv/3234/funktion-v11.html

- *IDS Scheer aktualisiert ARIS auf BPMN 2.0.* (18. Mai 2010). Abgerufen am 23. März 2014 von Heise Developer: http://www.heise.de/developer/meldung/IDS-Scheer-aktualisiert-ARIS-auf-BPMN-2-0-1002198.html

- IOZ InformationsOrganisationsZentrum AG, Sursee, Schweiz. (30. Mai 2013). *Komax optimiert die globale Zusammenarbeit und das Know-how-Management mit Microsoft SharePoint 2013.* Abgerufen am 25. Februar 2014 von IOZ InformationsOrganisationsZentrum: http://www.ioz.ch/loesungen/case-studies/DokumenteCaseStudies/Case-Study_Komax.pdf

- Koch, S. (2011). *Einführung in das Management von Geschäftsprozessen.* Berlin, Heiderlberg: Springer.

Literatur

- Konradin Mediengruppe. (2011). http://www.industrieanzeiger.de/erp. Leinfelden-Echterdingen: Konradin Mediengruppe. Abgerufen am 27. Februar 2014 von Einsatz von ERP-Lösungen in der Industrie. Konradin ERP-Studie 2011: http://www.industrieanzeiger.de/c/document_library/get_file?uuid=9ebf7124-0928-44c0-b63a-33d428242c4e&groupId=32571342

- Krieger, W., & Wischermann, B. (kein Datum). Stichwort Wareneingang. *Gabler Wirtschaftslexikon*. Springer Gabler. Abgerufen am 5. April 2014 von Gabler Wirtschaftslexikon: http://wirtschaftslexikon.gabler.de/Archiv/83509/wareneingang-v7.html

- Kurbel, K. (2011). *Enterprise Resource Planning und Supply Chain Management in der Industrie.* München: Oldenbourg.

- Laudon, K. C., Laudon, J. P., & Schoder, D. (2009). *Wirtschaftsinformatik - Eine Einführung.* Pearson.

- Leimeister, J. M. (2015). *Einführung in die Wirtschaftsinformatik* (12., vollst. neu überarb. u. ak. Ausg.). Springer Gabler.

- Mertens, P. (2009). *Integrierte Informationsverarbeitung 1. Operative Systeme in der Industrie.* Gabler.

Literatur

- Mertens, P., Bodendorf, F., König, W., Picot, A., Schumann, M., & Hess, T. (2012). *Grundzüge der Wirtschaftsinformatik* (11. Ausg.). Springer-Lehrbuch.

- Object Management Group. (Dezember 2013). *Business Process Model and Notation (BPMN) Version 2.0.2*. Von http://www.omg.org/spec/BPMN/2.0.2/ abgerufen

- Oxford University Press. (kein Datum). *Definition of server*. Abgerufen am 17. 3 2015 von Oxford Dictionaries (British & World English): http://www.oxforddictionaries.com/definition/english/server

- Petry, T., & Schreckenbach, F. (Mai 2013). *Enterprise 2.0 – Konsequenzen für die Arbeitswelt von morgen: Status Quo 2013 (Ergebnisbericht)*. (W. B. embrander, Hrsg.) Abgerufen am 2. März 2014 von http://de.slideshare.net/embrander/ergebnisbericht-der-studie-enterprise-20-konsequenzen-fr-die-arbeitswelt-von-morgen-status-quo-2013

- Porter, M. E. (1996). *Wettbewerbsvorteile (Competitive Advantage): Spitzenleistungen erreichen und behaupten*. Campus.

Literatur

- Rautenstrauch, C., & Schulze, T. (2002). *Informatik für Wirtschaftswissenschaftler und Wirtschaftsinformatiker.* Springer-Lehrbuch.

- SAP AG. (2013). *Kundenauftragsabwicklung – Verkauf ab Lager.* Abgerufen am 16. März 2014 von SAP Help Portal - SAP Best Practices Baseline package für Deutschland V3.607: http://help.sap.com/saap/sap_bp/BBLibrary/Documentation/109_ERP607V3_Process_Overview_DE_XX.ppt (Eingeschränkter Zugang)

- SAP AG. (2014a). *Über SAP AG | Unser Unternehmen | 1972-1981: Die Anfangsjahre.* Abgerufen am 2. März 2014 von SAP: http://global.sap.com/corporate-de/our-company/history/1972-1981.epx

- SAP AG. (2014b). *Über SAP AG | Unser Unternehmen | 1982-1991: Die Ära SAP R/2.* Abgerufen am 2. März 2014 von SAP: http://global.sap.com/corporate-de/our-company/history/1982-1991.epx

- Schaffry, A. (23. August 2010). *AKE Knebel schärft die Geschäftsintelligenz.* Abgerufen am 24. Februar 2014 von Computerwoche: http://www.computerwoche.de/a/ake-knebel-schaerft-die-geschaeftsintelligenz,2351464

- Scheer Group GmbH. (2014). *Blog*. Abgerufen am 16. März 2014 von Scheer Group - The Innovation Network: http://www.august-wilhelm-scheer.com

Literatur

- Scheer, A.-W. (1998). *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen*. Springer.
- Schmelzer, H. J., & Sesselmann, W. (2010). *Geschäftsprozessmanagement in der Praxis - Kunden zufrieden stellen - Produktivität steigern - Wert erhöhen*. Hanser Verlag .
- Seidler, L., Mack, M., & Bange, C. (März 2012). *Business Intelligence im Mittelstand 2011/2012*. (B. B. Center, Hrsg.) Abgerufen am 26. Februar 2014 von www.barc.de: http://www.barc.de/content/publications/business-intelligence-im-mittelstand-20112012?f=research
- SoftSelect Gmbh. (2011). *Verbreitung von BI-Lösungen in den Unternehmen*. Abgerufen am 1. März 2014 von http://www.softselect.de/.
- Software AG. (Januar 2014). *ARIS-Methode - Methodenhandbuch Version 9.5 R*. Abgerufen am 14. März 2014 von www.softwareag.de: http://documentation.softwareag.com/aris/aris95d/ARIS-Methode.pdf
- Stahlknecht, P., & Hasenkamp, U. (2005). *Einführung in die Wirtschaftsinformatik*. Springer.
- Wikipedia (Hrsg.). (14. 3 2015). *Server*. Abgerufen am 17. 3 2015 von Wikipedia: http://de.wikipedia.org/w/index.php?oldid=139735825

Literatur

- Wissenschaftliche Kommission Wirtschaftsinformatik (WKWI) im Verband der Hochschullehrer für Betriebswirtschaft e.V. und Fachbereich Wirtschaftsinformatik (GI FB WI) in der Gesellschaft für Informatik e.V. (GI). (18. Februar 2011). *Profil der Wirtschaftsinformatik*. (K. Kurbel, J. Becker, N. Gronau, E. Sinz, & L. Suhl, Hrsg.) Abgerufen am 21. Februar 2014 von Enzyklopädie der Wirtschaftsinformatik – Online-Lexikon: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/uebergreifendes/Kerndisziplinen/Wirtschaftsinformatik/profil-der-wirtschaftsinformatik

- WKWI; GI FBWI. (18. Februar 2011). Profil der Wirtschaftsinformatik (Wissenschaftliche Kommission Wirtschaftsinformatik (WKWI) im Verband der Hochschullehrer für Betriebswirtschaft e.V. und Fachbereich Wirtschaftsinformatik (FB WI) in der Gesellschaft für Informatik e.V. (GI)). (K. Kurbel, J. Becker, N. Gronau, E. Sinz, & L. Suhl, Hrsg.) *Enzyklopädie der Wirtschaftsinformatik – Online-Lexikon*.

- Zimmermann, S., & Rentrop, C. (12 2012). Schatten-IT. *HMD Praxis der Wirtschaftsinformatik, 49*(6).

Literatur

- Chamberlain, D. D., & Boyce, R. F. (1974). SEQUEL: A Structured English Query Language. *SIGFIDET '74 Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) workshop on Data description, access and control*.
- Chen, P. P. (1977). The entity-relationship model: a basis for the enterprise view of data. *National Computer Conference: AFIPS Conference Proceedings. 46*. Dallas, Texas: American Federation of Information Processing Societies AFIPS Press.
- Codd, E. F. (Juni 1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM, 13*(6).
- Cordts, S., Blakowski, G., & Brosius, G. (2011). *Datenbanken für Wirtschaftsinformatiker*. Springer.
- Date, C. (1975). *An Introduction to Database Systems*.
- Date, C. J. (2003). *An Introduction to Database Systems* (8 Ausg.).
- Dustdar, S., Gall, H., & Hauswirth, M. (2003). *Software-Architekturen für verteilte Systeme : Prinzipien, Bausteine und Standardarchitekturen für moderne Software*. Springer.
- Elmasri, R. A., & Navathe, S. B. (2009). *Grundlagen von Datenbanksystemen*. Pearson Studium.
- Elmasri, R. A., & Navathe, S. B. (2011). *Fundamentals of Database Systems* (3. Ausg.). Pearson Studium.

Literatur

- ISO/IEC. (2011). *ISO/IEC 9075-1:2011. Information technology - Database languages - SQL - Part 1: Framework (SQL/Framework).*
- Kemper, A., & Eickler, A. (2015). *Datenbanksysteme. Eine Einführung.* (10., aktualisierte und erweiterte Ausg.). Oldenbourg Verlag.
- Lieder, T. (17. 11 2011). *Die Datenbank vom Weihnachtsmann... .* Abgerufen am 4. 10 2015 von www.youtube.com: https://youtu.be/rkB0wLA6f_U
- MySQL. (2015). *MySQL Workbench Features*. Abgerufen am 6. 10 2015 von MySQL.de. Die populärste Open-Source-Datenbank der Welt: https://www.mysql.de/products/workbench/features.html
- MySQL. (kein Datum). *MySQL CE Downloads.* Von MySQL: http://dev.mysql.com/downloads/. abgerufen
- Unterstein, M., & Matthiesen, G. (2012). *Relationale Datenbanken in Theorie und Praxis.* Springer Vieweg.
- Vossen, G. (2008). *Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme* (5. Ausg.). Oldenbourg Wissenschaftsverlag.
- Prof.Dr. Hermann Gehring, Kurs 825 Anwendungssysteme und Geschäftsprozessmodellierung: https://vu.fernuni-hagen.de/lvuweb/lvu/file/FeU/WiWi/2016WS/00825/oeffentlich/31751_00825.pdf
- Tim Welkins, Bernd Osterreich: UML 2 Zertifizierung – Fundamental, Intermediate und Advanced

MHP
A PORSCHE COMPANY

Herzlichen Dank für Ihre Aufmerksamkeit!

Hugo Colceag

MHP Management- und IT-Beratung GmbH

Film- und Medienzentrum I Königsallee 49 I D-71638 Ludwigsburg
Telefon +49 (0)7141 7856-0 I Fax +49 (0)7141 7856-199
eMail info@mhp.com I Internet www.mhp.com