

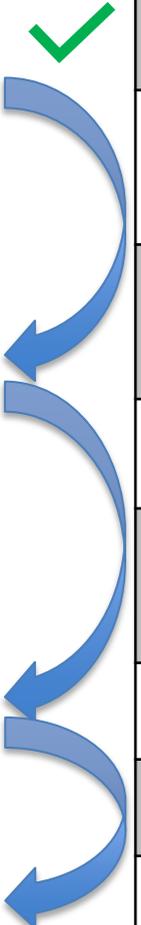
Duale Hochschule Baden-Württemberg  
Heilbronn

---

## **Systemanalyse und -entwurf**

Einheit 2 – Anwendungsfalldiagramm

# Agenda Systemanalyse und -entwurf



|   | Thema                           | Inhalte   | (Geplante(r) Termin |
|---|---------------------------------|---|---------------------|
| 1 | Einführung                      | Einführung in Veranstaltung. Modulhandbuch, Lernziele und Überblick, empfohlene Literatur. Requirements Game. Überblick Anforderungsmanagement und UML. Aufteilung in Teams und Themen.   | 31.03 NM            |
| 2 | Systemanalyse und Entwurf       | Das <b>UML-Anwendungsfalldiagramm</b> (Use Cases). Zweck und Bestandteile, erläutert mit Beispielen. Umsetzung eines Beispiels in Kleingruppen.<br>Das <b>UML-Aktivitätsdiagramm</b> . Zweck und Bestandteile, erläutert mit Beispielen. Umsetzung eines Beispiels in Kleingruppen                        | 02.04. NM           |
| 3 | Systemanalyse und Entwurf       | Das <b>UML-Aktivitätsdiagramm</b> . Zweck und Bestandteile, erläutert mit Beispielen. Umsetzung eines Beispiels in Kleingruppen<br>Das <b>UML-Klassendiagramm</b> . Zweck und Bestandteile, erläutert mit Beispielen. Umsetzung eines Beispiels in Kleingruppen<br>Vorbereitung zur Anforderungssammlung. | 03.04. VM           |
| 4 | Action!<br>Anforderungssammlung | <b>Anforderungssammlung</b> für Projekte (13:30-15:00) mit Kunden oder Brainstorming.<br>Danach (zusammen): Reflexion, Arbeit in Teams (Tooling und Modellierung), Puffer   | 15.04. NM           |
| 5 | Guest Lecture                   | Guest Lecture „Moderne Software-Entwicklung in großen Unternehmen: von der Idee zum Produkt am Beispiel der SAP Internen Development Plattform (IDP) Hyperspace“.<br>Erwartungen an Präsentationen, Puffer  | 17.04. VM           |
| 6 | Präsentationen                  | <b>Präsentationen</b> und Integration der Projekte, Reflexion   | 14.05 NM            |
| 7 | Integration                     | <b>Erstellen von Beispielsaufgaben</b> inkl. Musterlösungen in Teams.   | 21.05 NM            |
| 8 | Integration                     | <b>Messe der Diagramme, Reflexion</b><br><b>Verankern</b> des erlernten Vorgehens mit UML an weiteren Beispielen. Entweder einzeln oder in Kleingruppen.  | 03.06 VM            |
| 9 | Abschluss                       | Fragerunde, Wiederholungen, Vorbesprechung <b>Klausur</b>   | 03.06. NM           |

# Willkommen im Sprint 2

---

## ▼ Sprint 2 - Wichtigsten Grundlagen und Konzepte Für Teilnehmer/innen verborgen

Für Teilnehmer/innen verborgen

### Backlog

- Als Professorin möchte ich Studierenden vermitteln, was Use Case Diagramme sind, damit sie sie einsetzen können.
- Als Professorin möchte ich Studierenden vermitteln, wie Use Case Diagramme zu modellieren sind, damit sie sie Anforderungen in Use Cases übersetzen können.
- Als Professorin möchte ich Studierenden vermitteln, was Klassendiagramme sind, damit sie sie einsetzen können.
- Als Professorin möchte ich Studierenden vermitteln, wie Klassendiagramme zu modellieren sind, damit sie sie Anforderungen in Use Cases übersetzen können.
- Als Professorin möchte ich Studierenden vermitteln, was Aktivitätsdiagramme sind, damit sie sie einsetzen können.
- Als Professorin möchte ich Studierenden vermitteln, wie Aktivitätsdiagramme zu modellieren sind, damit sie sie Anforderungen in Use Cases übersetzen können.
- Als Student:in möchte ich verstehen, wie Anforderungen erhoben werden, damit ich Kunden aufmerksam zuhören und richtige Fragen stellen kann
- Als Student:in möchte ich Notation der Diagramme kennen und richtig einsetzen, damit ich mich auf die Inhalte konzentrieren kann
- Als Student:in möchte ich klausurrelevante Inhalte möglichst früh kennenlernen, damit ich sie besser für die Klausur verinnerlichen kann

Ansicht der Veranstaltung im [Moodle](#)

# Software Engineering in a nutshell

---

## Definition Software Engineering

*Zusammenfassend kann man Software-Engineering als die Wissenschaft der systematischen Entwicklung von Software, beginnend bei den Anforderungen bis zur Abnahme des fertigen Produkts und der anschließenden Wartungsphase definieren. Es werden etablierte Lösungsansätze für Teilaufgaben vorgeschlagen, die, häufig kombiniert mit neuen Technologien, vor Ihrer Umsetzung auf ihre Anwendbarkeit geprüft werden. Das zentrale Mittel zur Dokumentation von Software-Engineering-Ergebnissen sind UML-Diagramme.*

Stephan Kleuker: Grundkurs Software-Engineering mit UML, S. 4

# Anwendungsfalldiagramm

---

## Agenda

- Einführung
- Akteure
- Anwendungsfälle
- Beziehungen
  - zwischen Anwendungsfällen und Akteuren
  - zwischen Anwendungsfällen
  - zwischen Akteuren
- Beschreibung eines Anwendungsfalls
- Modellierung
- Typische Modellierungsfehler
- Beispiel: Informationssystem der Studienabteilung
- Zusammenfassung Elemente / Rückblick

Die Einheit basiert mit freundlicher Genehmigung von Frau Prof. Dr. Seidl auf Inhalten der Veranstaltung „Objektorientierte Modellierung“ an der TU Wien

# Anwendungsfalldiagramm

---

## Agenda

- **Einführung**
- Akteure
- Anwendungsfälle
- Beziehungen
  - zwischen Anwendungsfällen und Akteuren
  - zwischen Anwendungsfällen
  - zwischen Akteuren
- Beschreibung eines Anwendungsfalls
- Modellierung
- Typische Modellierungsfehler
- Beispiel: Informationssystem der Studienabteilung
- Zusammenfassung Elemente / Rückblick

# Einführung (1/2)

---

- Use Cases (= Anwendungsfälle) sind Ausgangspunkt vieler objekt- orientierter Entwicklungsmethoden.
- Zusätzlich dienen sie oft auch als Basiskonzept, das sich über den kompletten Analyse- und Designprozess hinweg spannt.
- Ausgangsfragen für den Einsatz von Anwendungsfällen:
  - Warum verwendet man Anwendungsfälle?
  - Wie sehen Anwendungsfälle aus?
  - Was macht man mit Anwendungsfällen, wenn man sie einmal definiert hat?
- Anwendungsfälle konzentrieren sich auf das fundamentale Problem bei der Entwicklung eines Systems:  
der **Entwicklung einer Lösung für den Kunden bzw. Anwender, die der Kunde bzw. der Anwender auch gewünscht hat.**

## Einführung (2/2)

---

„Ein Anwendungsfall ist eine Sequenz von Transaktionen innerhalb eines Systems, deren Aufgabe es ist, einen für den einzelnen Akteur (Anwender) identifizierbaren Nutzen zu erzeugen.“ [Ivar Jacobson]

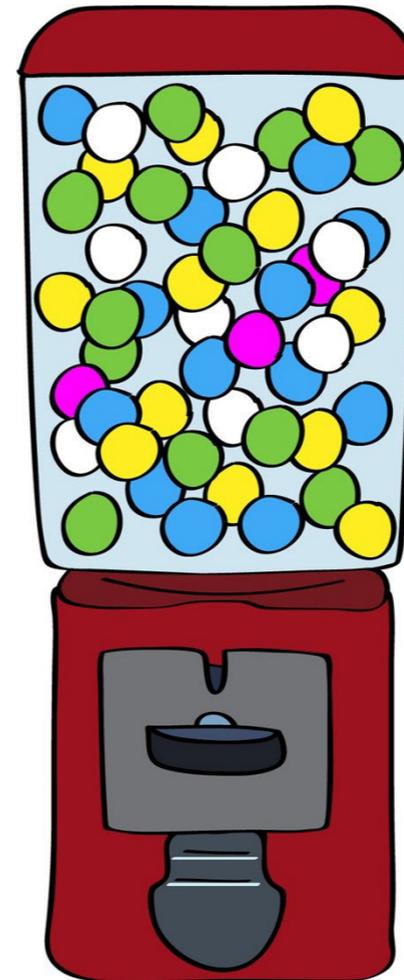
- Anwendungsfälle repräsentieren die Anforderungen der Kunden
- Akteure interagieren mit dem System im Kontext der Anwendungsfälle
- Akteur:
  - Rolle, die jemand oder etwas einnimmt und die in Beziehung zum Geschäftsbereich steht, oder
  - Alles, das mit dem System interagiert
- Transaktionen innerhalb eines Systems implizieren, dass dem Akteur eine Reihe von Möglichkeiten geboten wird um mit dem System zu kommunizieren und dass durch sie ein messbarer Nutzen erzeugt wird.
- Ein messbarer Nutzen impliziert, dass die Ausführung einer Transaktion eine sichtbare, quantifizierbare und/oder qualifizierbare Auswirkung auf jene Dinge hat, die außerhalb des Systems liegen, im speziellen auf den Akteur.

# Beispiele Anwendungsfalldiagramme

---

## Kaugummiautomat

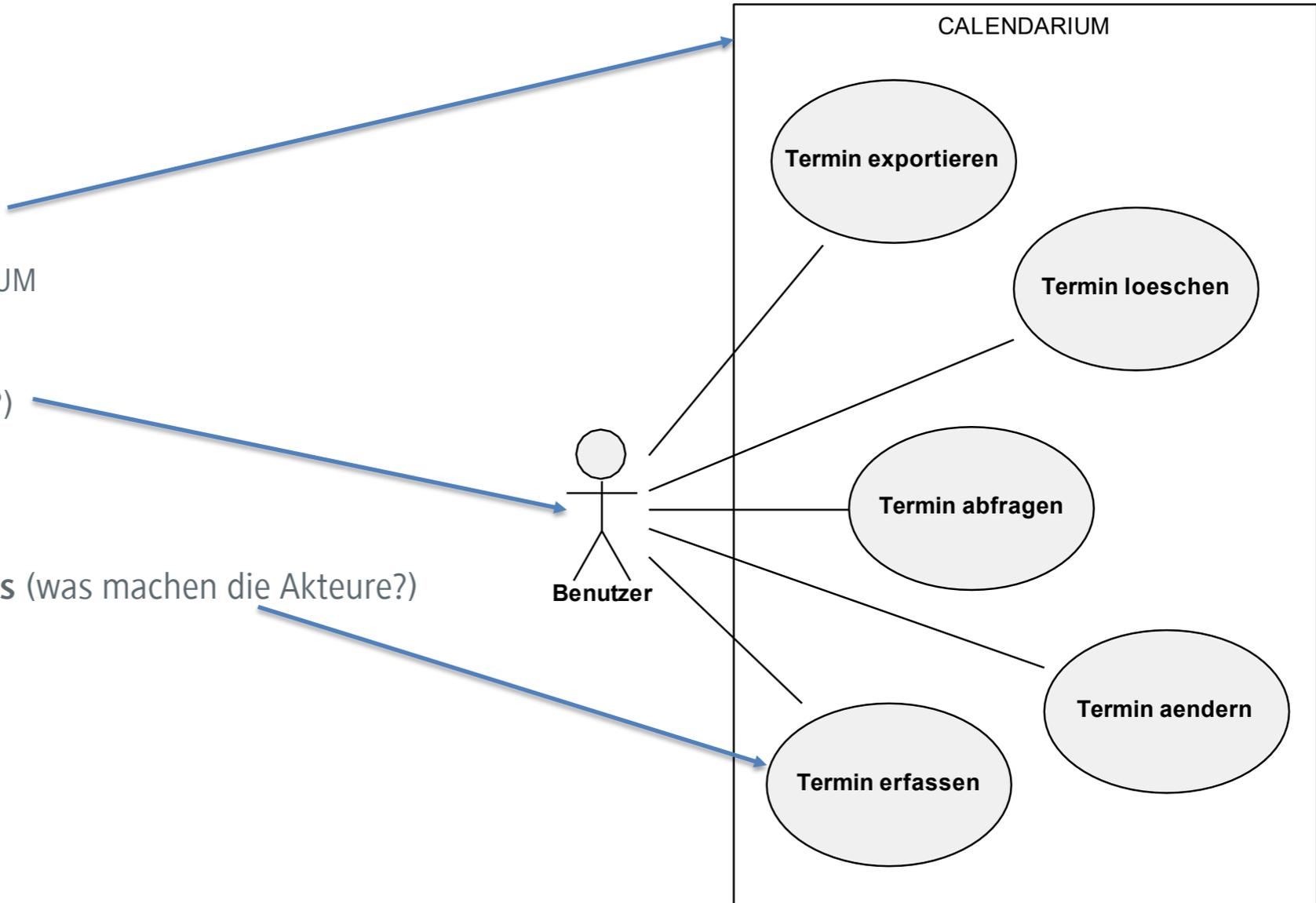
- Anwendungsfall / Use Case Kaugummi kaufen
- Standardablauf der Benutzerinteraktion
  - Münze einwerfen
  - Hebel drehen
  - Klappe öffnen
  - Kaugummi entnehmen



# Beispiele Anwendungsfalldiagramme

## CALENDARIUM

- **System** (was wird beschrieben?)
  - der Onlinekalender CALENDARIUM
  
- **Akteur** (wer benutzt das System?)
  - Benutzer des Kalenders
  
- **Anwendungsfälle des Benutzers** (was machen die Akteure?)
  - Abfragen von Terminen
  - Exportieren von Terminen
  - Löschen von Terminen
  - Ändern von Terminen
  - Erfassen von Terminen



# Anwendungsfalldiagramm

---

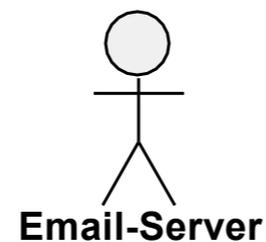
## Agenda

- Einführung
- **Akteure**
- **Anwendungsfälle**
- Beziehungen
  - zwischen Anwendungsfällen und Akteuren
  - zwischen Anwendungsfällen
  - zwischen Akteuren
- Beschreibung eines Anwendungsfalls
- Modellierung
- Typische Modellierungsfehler
- Beispiel: Informationssystem der Studienabteilung
- Zusammenfassung Elemente / Rückblick und Ausblick

# Akteur

---

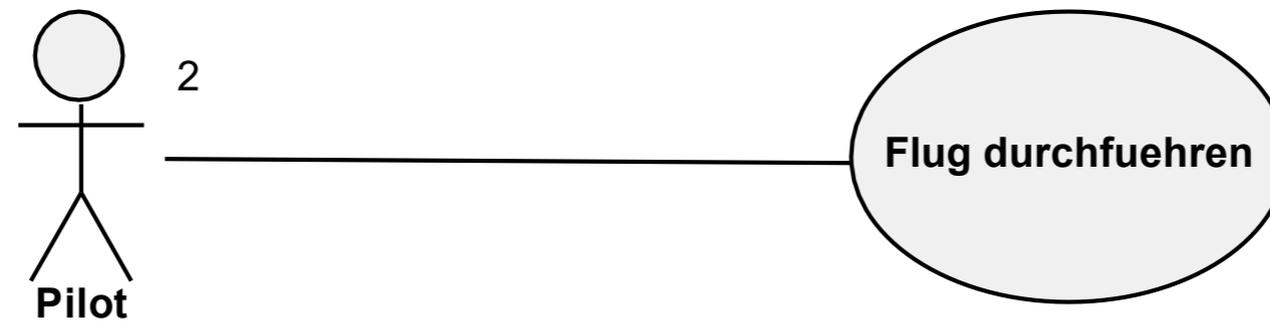
- Akteure **interagieren** mit dem System...
  - indem sie **das System benutzen**, d.h. die Ausführung von Anwendungsfällen initiieren
  - indem sie **vom System benutzt werden**, d.h. Funktionalität zur Realisierung von Anwendungsfällen zur Verfügung stellen
- Akteur wird durch Assoziationen mit Anwendungsfällen verbunden, d.h. er »kommuniziert« mit dem System
- Jeder Akteur muss mit mindestens einem Anwendungsfall kommunizieren
- Die Assoziation ist binär und kann Multiplizitäten aufweisen
  
- Notationsvarianten:



# Akteur

---

## Beispiel für Multiplizitäten: Pilot



# Akteur

---

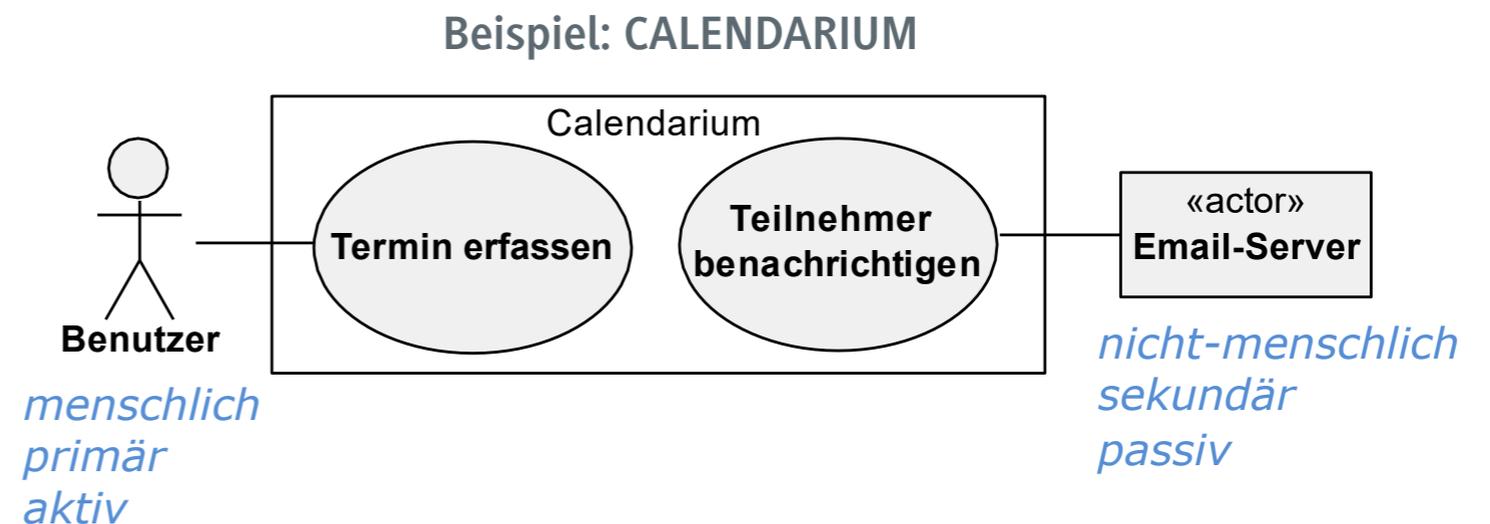
## Eigenschaften

- Akteure repräsentieren **Rollen der Benutzer**
  - Konkrete Benutzer können gleichzeitig mehrere Rollen spielen, annehmen und ablegen
- Akteure befinden sich **klar außerhalb** der Systemgrenzen
- Üblicherweise werden Benutzerdaten auch innerhalb des Systems verwaltet. Diese werden als Objekte bzw. Klassen innerhalb des Systems modelliert.
- **Beispiel: Kassierer**
  - Als Akteur am Kassenterminal
    - Die Rolle, in der die Person mit dem Kassensystem interagiert
  - Die **Klasse Kassierer** umfasst Objekte, welche die Benutzerdaten beinhalten (Name, MitarbeiterNr, ...)

# Akteur

## Klassifikation

- **Menschlich**
  - z.B. Anfänger, geübter Benutzer, Admin
- **Nicht-menschlich**
  - z.B. Fax-System, E-Mail-System
- **Primär:** Hauptnutznießer der Anwendung
- **Sekundär:** notwendig für das Funktionieren des Systems
- **Aktiv:** stößt selbst Anwendungsfälle an
- **Passiv:** stößt selbst keine Anwendungsfälle an



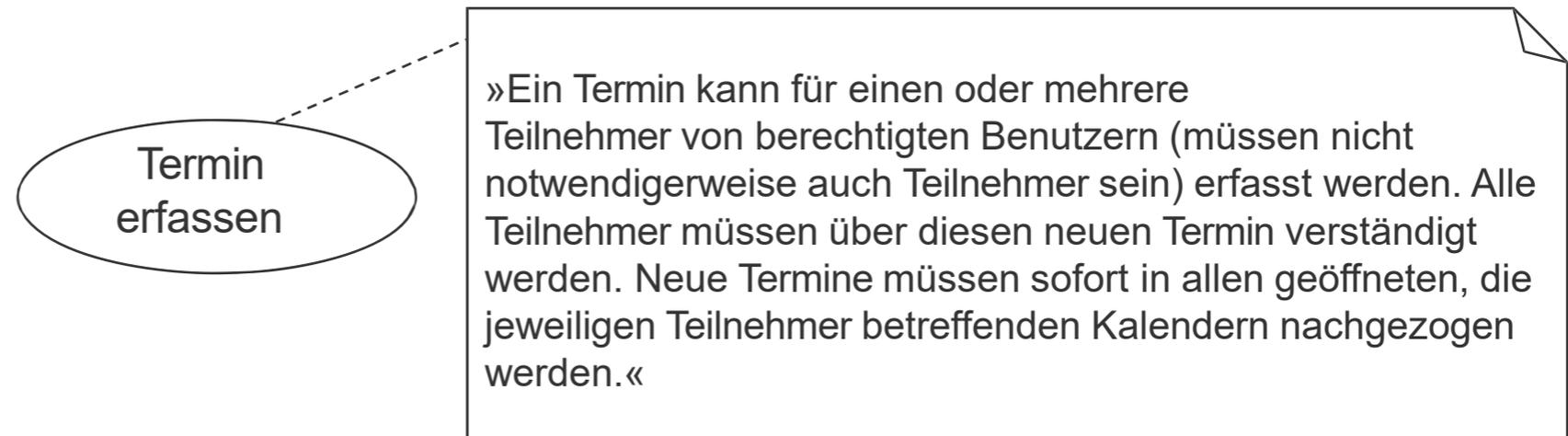
# Anwendungsfall

- Anwendungsfälle (Use Cases) beschreiben das **Verhalten**, das von dem zu entwickelnden System **erwartet** wird
- Identifizierung durch Sammeln von Kundenwünschen und Analyse der textuellen Problemstellung
- Anwendungsfälle sind mit mindestens einem Akteur verbunden

- Notationsvarianten:



- Kurzbeschreibung als Notiz:



# Anwendungsfalldiagramm

---

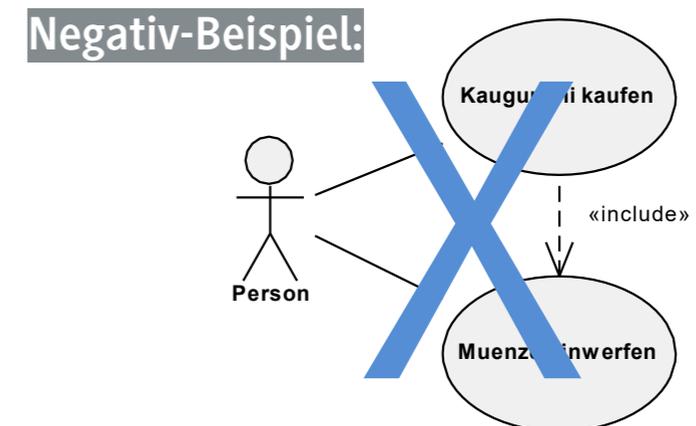
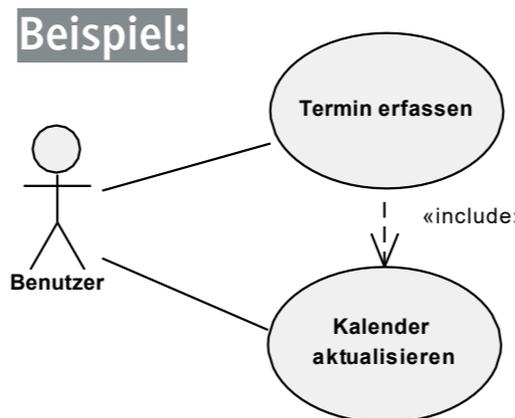
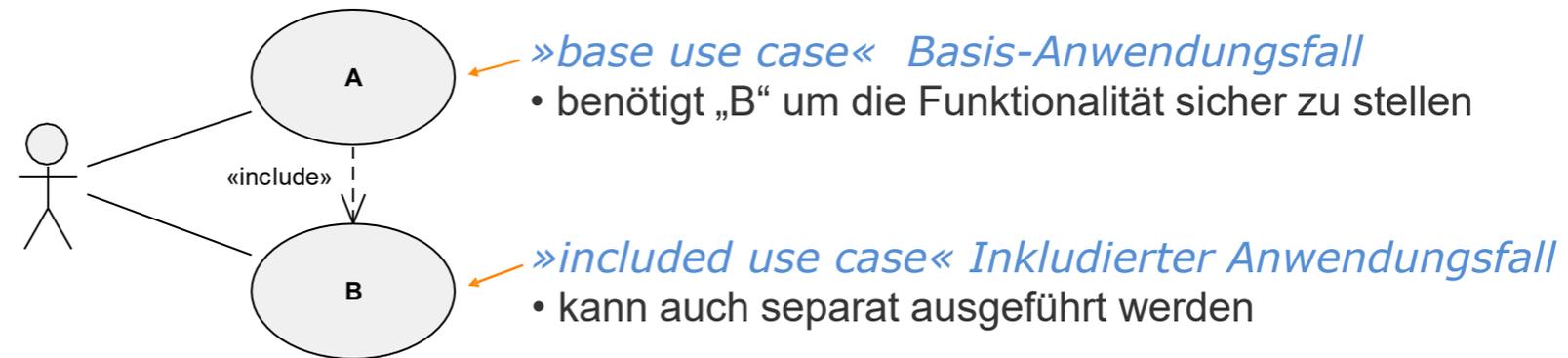
## Agenda

- Einführung
- Akteure
- Anwendungsfälle
- **Beziehungen**
  - **zwischen Anwendungsfällen und Akteuren**
  - **zwischen Anwendungsfällen**
  - **zwischen Akteuren**
- Beschreibung eines Anwendungsfalls
- Modellierung
- Typische Modellierungsfehler
- Beispiel: Informationssystem der Studienabteilung
- Zusammenfassung Elemente / Rückblick

# Anwendungsfall – Beziehungen

## «include» - Beziehung

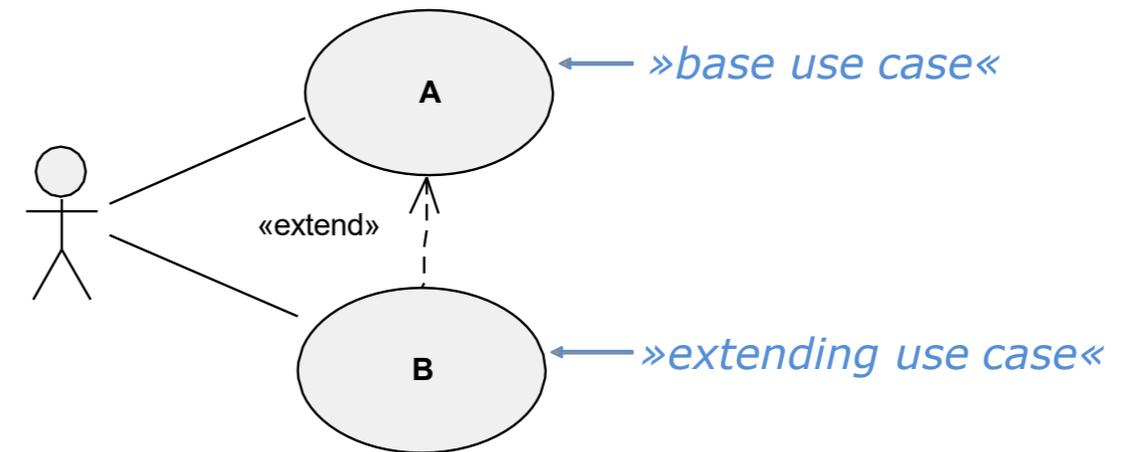
- Das **Verhalten** des benutzten Anwendungsfalls (inkludierter Anwendungsfall) wird in den benutzenden Anwendungsfall (Basis-Anwendungsfall) **eingebunden**
- B ist unbedingt notwendig, um die Funktionalität von A sicher zu stellen
- B kann separat ausgeführt werden



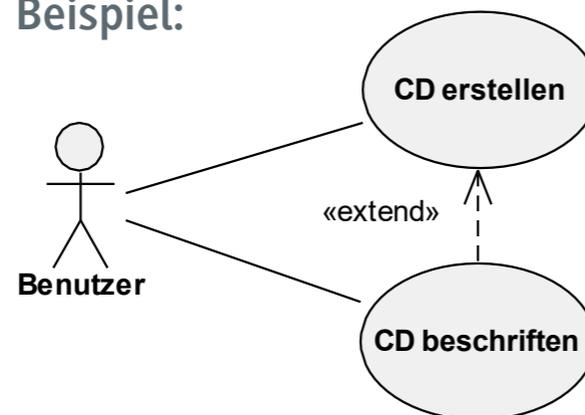
# Anwendungsfall – Beziehungen

## «extend» - Beziehung

- Das Verhalten von B **kann** in A **inkludiert werden**
  - Somit entscheidet A, ob B ausgeführt wird
- B kann von A aktiviert werden, **muss aber nicht**
  - A bzw. B können auch separat ausgeführt werden
- Angabe des »Wo« durch Erweiterungsstellen in A
- Angabe des »Wann« durch Bedingung in A bzw. als Teil der «extend»-Beziehung



### Beispiel:

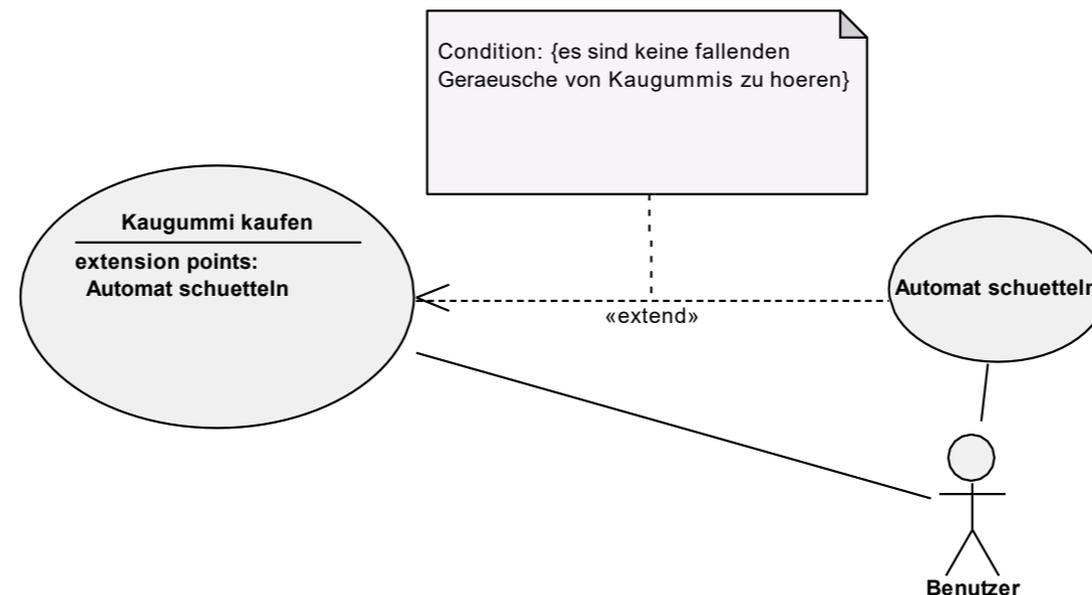


# Anwendungsfall – Beziehungen

## «extend» - Beziehung: Erweiterungsstellen

- Mehrere Erweiterungsstellen (extension points) je Anwendungsfall möglich
- Namen von Erweiterungsstellen
  - müssen eindeutig sein
  - müssen nicht mit den Namen der erweiternden Anwendungsfälle übereinstimmen

### • Beispiel: Kaugummiautomat



# Anwendungsfall – Beziehungen. Beispiele

---

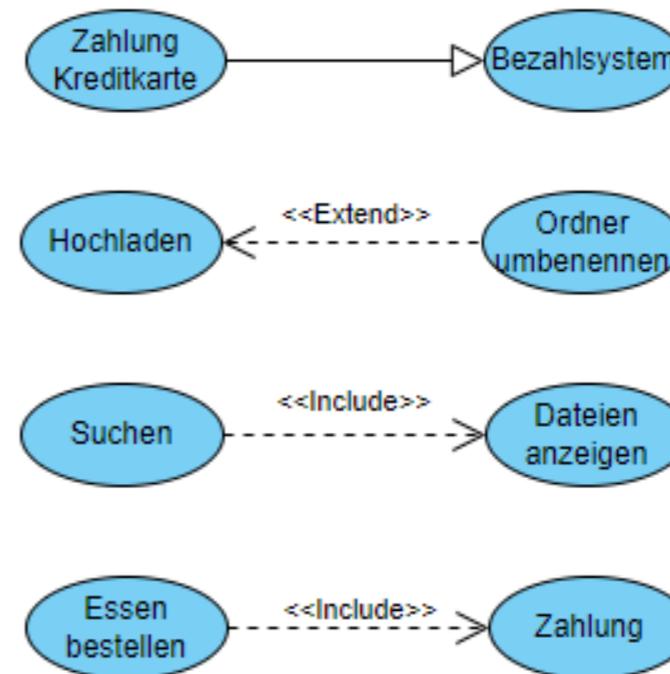
## Um welche Beziehung handelt es sich?

- Zahlen mit Kreditkarte erweitert das Online Bezahlssystem.
- Beim Hochladen von Dateien können die Ordner umbenannt werden.
- Beim Suchen nach Dateien werden die gefundenen Dateien angezeigt.
- Kunden bestellen Essen. Dabei wird das Essen gezahlt.

# Anwendungsfall – Beziehungen. Beispiele

## Um welche Beziehung handelt es sich?

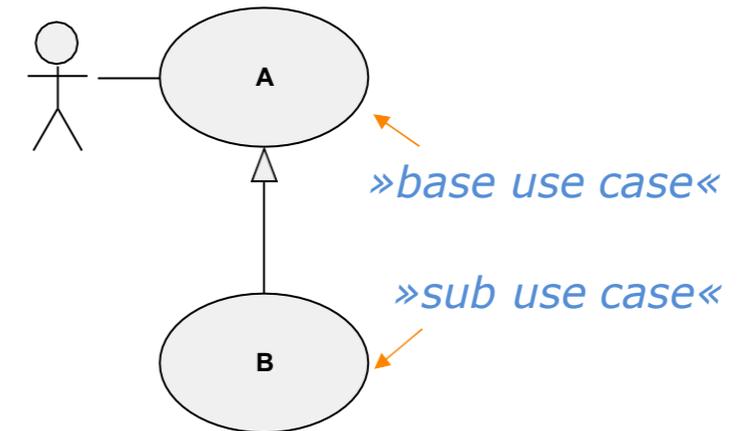
- Zahlen mit Kreditkarte erweitert das Online Bezahlungssystem.
- Beim Hochladen von Dateien können die Ordner umbenannt werden.
- Beim Suchen nach Dateien werden die gefundenen Dateien angezeigt.
- Kunden bestellen Essen. Dabei wird das Essen gezahlt.



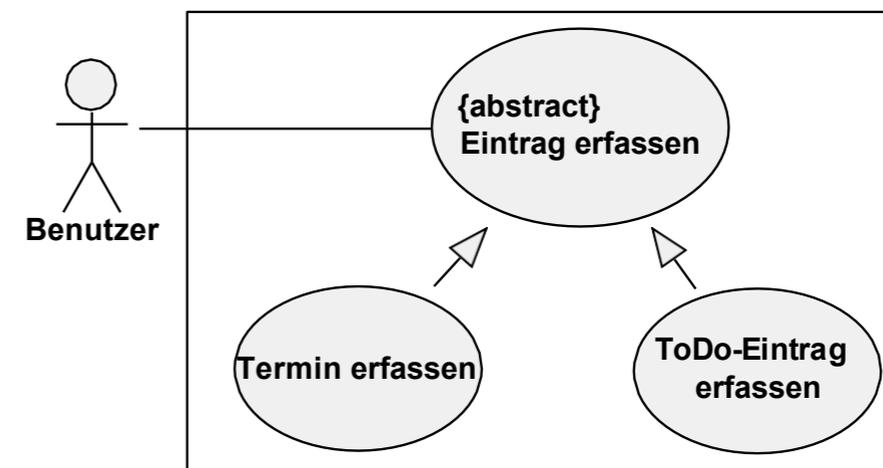
# Generalisierung

## Bei Anwendungsfällen

- B erbt das Verhalten von A und kann dieses überschreiben oder ergänzen
  - B erbt alle Beziehungen von A
  - B benötigt A (übernimmt Grundfunktionalität von A)
  - B entscheidet, was von A ausgeführt bzw. geändert wird
  - Modellierung abstrakter Anwendungsfälle möglich: {abstract}
- abstrakte Anwendungsfälle sind nicht ausführbar!



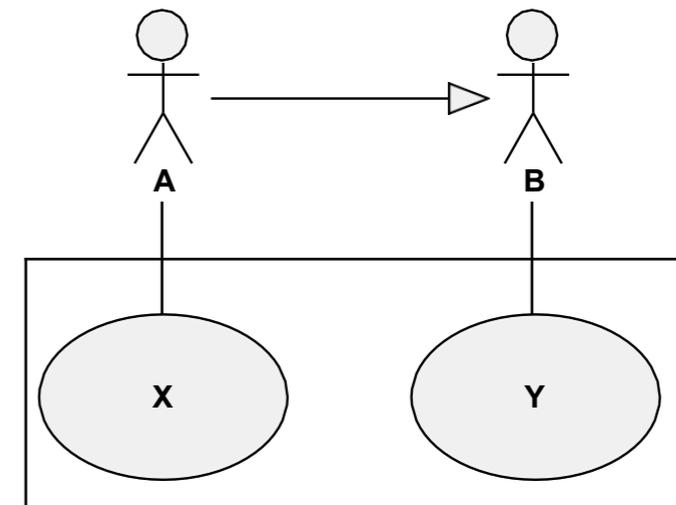
## Beispiel:



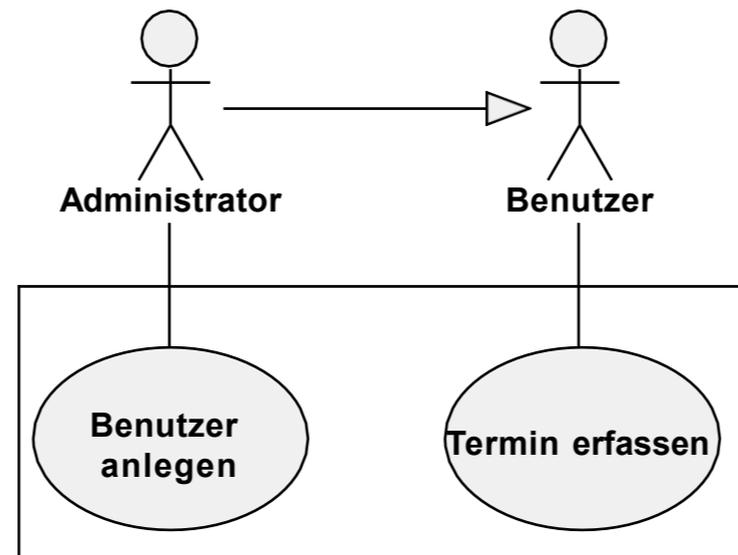
# Generalisierung

## Bei Akteuren

- Akteur A erbt von Akteur B
- A kann mit den Anwendungsfällen X und Y kommunizieren
- B kann nur mit Y kommunizieren
- Mehrfachvererbung ist erlaubt



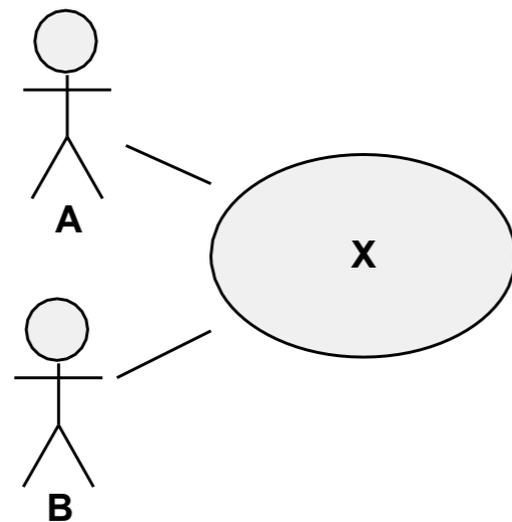
Beispiel:



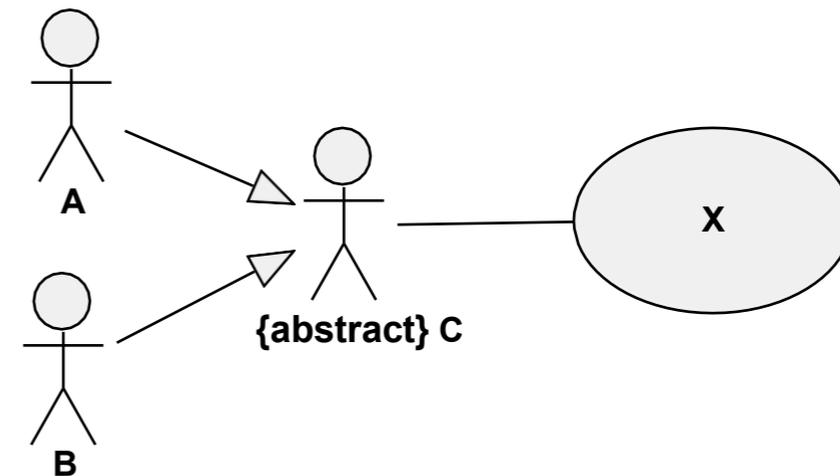
# Generalisierung

## Bei Akteuren

- Unterscheidung, ob mehrere Akteure gemeinsam mit einem Anwendungsfall kommunizieren können oder müssen.



**A und B** kommunizieren mit X

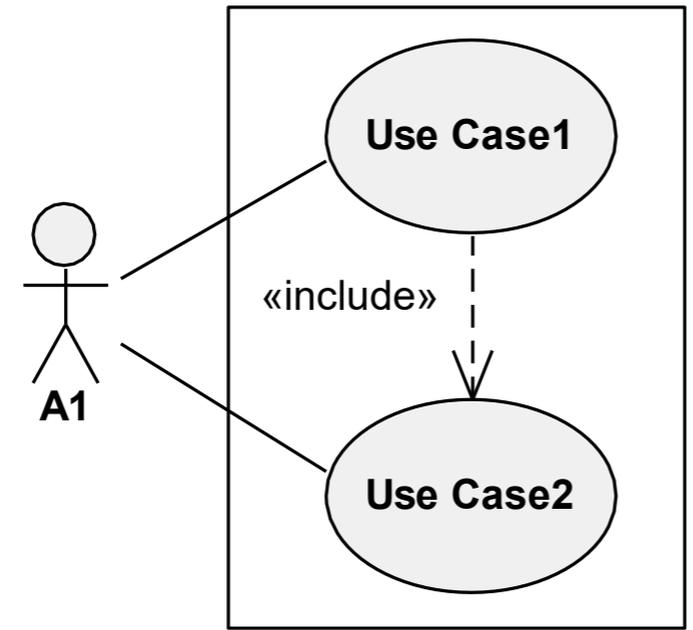


**A oder B** kommuniziert mit X

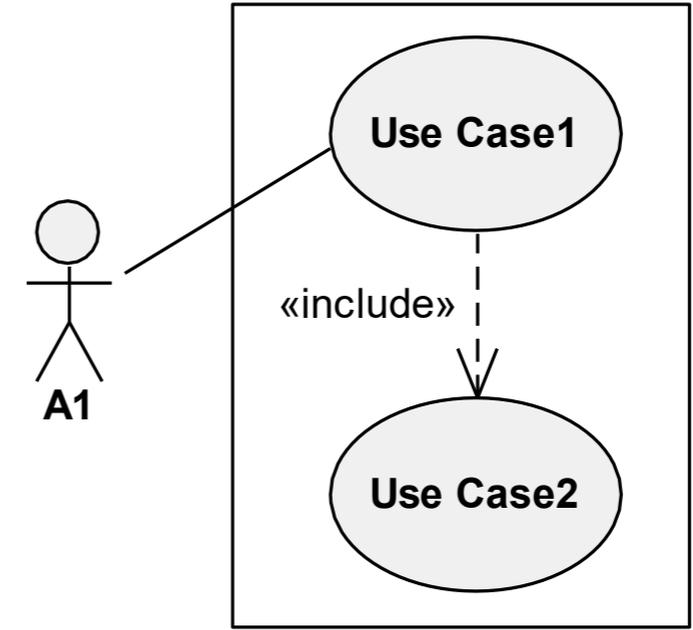
# UML Standard vs. Best Practice

<<include>>

*Standard*



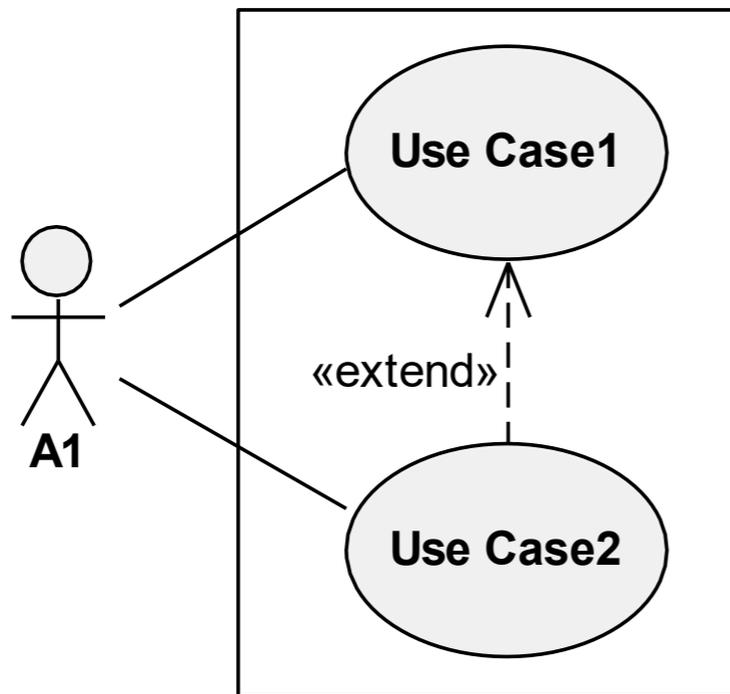
*Best Practice*



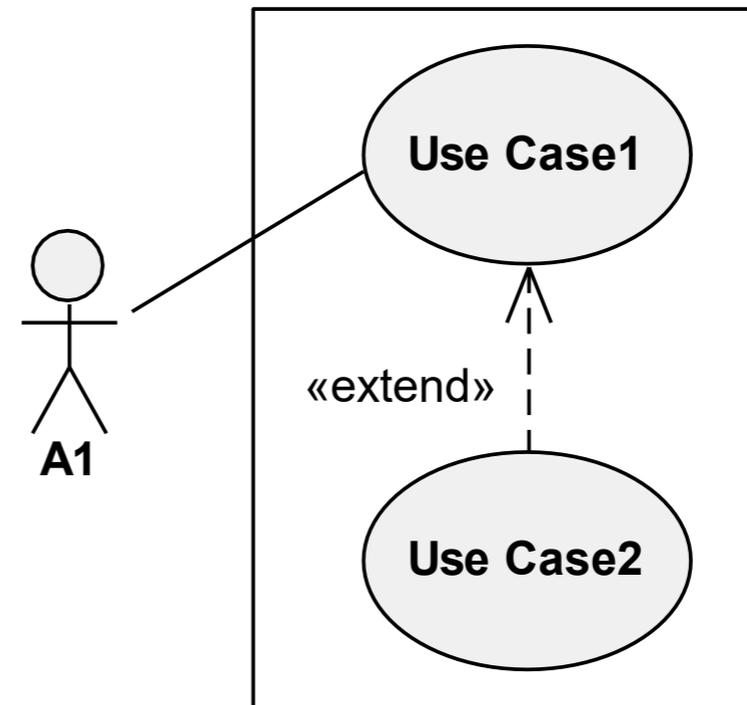
# UML Standard vs. Best Practice

<<extend>>

*Standard*



*Best Practice*



# Gemeinsame Beispiele

---

## Beichten

- Modellieren Sie den Anwendungsfall der Beichte:
  - Der Sünder kommt zum Pfarrer zur Beichte in die Kirche
  - Der Sünder beichtet dem Pfarrer
  - Der Pfarrer weist den Sünder an, nach der Beichte zu beten
  - Berücksichtigen Sie: auch Pfarrer können Sünder sein
- Erledigen Sie die Aufgabe in Kleingruppen
  
- Sie haben 20 Minuten Zeit (da erstes eigenständiges Beispiel)
- Gemeinsame Ergebnisdiskussion



**20 min**

# Anwendungsfalldiagramm

---

## Agenda

- Einführung
- Akteure
- Anwendungsfälle
- Beziehungen
  - zwischen Anwendungsfällen und Akteuren
  - zwischen Anwendungsfällen
  - zwischen Akteuren
- **Beschreibung eines Anwendungsfalls**
- Modellierung
- Typische Modellierungsfehler
- Beispiel: Informationssystem der Studienabteilung
- Zusammenfassung Elemente / Rückblick

# Anwendungsfallbeschreibung

- **Aufbau einer Anwendungsfallbeschreibung – strukturierter Text**

- Name
- Kurzbeschreibung
- Vorbedingung: Voraussetzung für erfolgreiche Ausführung
- Nachbedingung: Systemzustand nach erfolgreicher Ausführung
- Fehlersituationen: nur problembereichsrelevante Fehler
- Systemzustand im Fehlerfall
- Akteure, die mit dem Anwendungsfall kommunizieren
- Trigger: auslösende Ereignisse für den Anwendungsfall
- Standardablauf: einzelne Schritte / andere Anwendungsfälle
- Alternativabläufe: Abweichungen vom Standardablauf

[nach A. Cockburn: Goals and Use Cases. Journal of Object-Oriented Programming]

Beispiel Anwendungsfallbeschreibung:

| <b>Name:</b>                        | <b>Hoersaal reservieren</b>   |
|-------------------------------------|---|
| <b>Kurzbeschreibung:</b>            | Ein Mitarbeiter reserviert für eine Veranstaltung einen Hörsaal der Universität.  |
| <b>Vorbedingung:</b>                | Der Mitarbeiter ist berechtigt, Hörsäle zu reservieren.   |
| <b>Nachbedingung:</b>               | Ein Hörsaal ist reserviert.   |
| <b>Fehlersituationen:</b>           | Kein Hörsaal ist frei.  |
| <b>Systemzustand im Fehlerfall:</b> | Der Mitarbeiter hat keinen Hörsaal reserviert.  |
| <b>Akteure:</b>                     | Mitarbeiter   |
| <b>Trigger:</b>                     | Mitarbeiter benötigt einen Hörsaal.   |
| <b>Standardablauf:</b>              | (1) Mitarbeiter loggt sich im System ein.<br>(2) Mitarbeiter wählt Hörsaal aus.<br>(3) Mitarbeiter wählt den Termin aus.<br>(4) System bestätigt, dass der Hörsaal frei ist.<br>(5) Mitarbeiter bestätigt Reservierung. |
| <b>Alternativabläufe:</b>           | (4') Hörsaal ist nicht frei.<br>(5') System schlägt alternative Hörsäle vor.<br>(6') Mitarbeiter wählt Hörsaal aus und bestätigt Reservierung.  |

# Anwendungsfalldiagramm

---

## Agenda

- Einführung
- Akteure
- Anwendungsfälle
- Beziehungen
  - zwischen Anwendungsfällen und Akteuren
  - zwischen Anwendungsfällen
  - zwischen Akteuren
- Beschreibung eines Anwendungsfalls
- **Modellierung**
- Typische Modellierungsfehler
- Beispiel: Informationssystem der Studienabteilung
- Zusammenfassung Elemente / Rückblick

# Identifikation von Akteuren

---

- Wer **benutzt** die wesentlichen Anwendungsfälle?
- Wer braucht Systemunterstützung für die **tägliche Arbeit**?
- Wer ist für die **Systemadministration** zuständig?
- Mit welchen externen **Geräten / (Software-)Systemen** muss das System kommunizieren können?
- Wer oder was interessiert sich für die **Ergebnisse** des Systems?

# Identifikation von Anwendungsfällen

---

- **Nach** der Identifikation der Akteure
- Trigger für Anwendungsfälle suchen
  - Trigger = Ereignisse, die eintreten müssen, damit das System veranlasst wird ein Ergebnis zu produzieren.
- Der Aufruf des Systems erfolgt oft durch einen Akteur, der damit Akteur des Anwendungsfalles wird.
- Es werden folgende Trigger unterschieden: interne, externe und zeitliche Trigger.

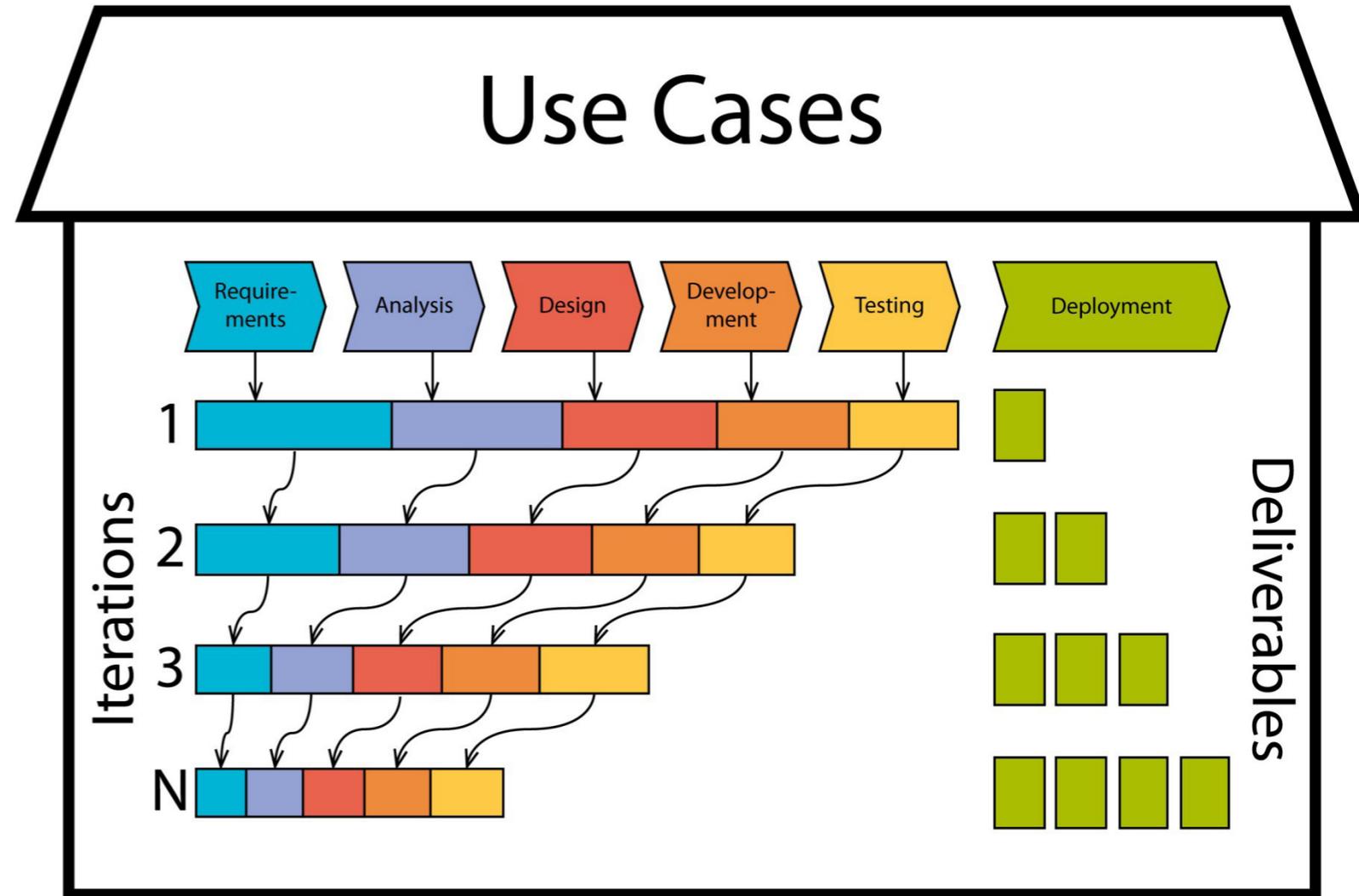
# Regeln zur Anwendungsfallmodellierung

---

- Die **wichtigsten funktionalen Anforderungen** müssen in den Anwendungsfällen **festgehalten** werden.
- Ein Anwendungsfall....
  - beschreibt eine Transaktion für die der Auftraggeber bezahlt.
  - beschreibt einen typischen Fall, ein System zu verwenden und nicht mehr.
  - ist wie ein Theaterstück. Die Anwendungsfallbeschreibung enthält die Choreographie.
  - hat eine Einleitung, einen Hauptteil und einen Schluss.
  - soll so einfach wie möglich beschrieben sein.
  - muss präzise definiert sein.
  - ist dann fertig beschrieben, wenn der Kunde, die Anwender und die Softwareentwickler ihn akzeptieren.
  - stellt die Grundlage für einen Systemtest dar.
  - sollte mit maximal zwei Seiten beschrieben werden.

# Iteratives Vorgehen

- Anwendungsfälle entstehen während der Anforderungserhebung
- Bilden Grundlage für Folgephasen im Softwareentwicklungsprozess
- Im Laufe der Entwicklung können Unklarheiten/Ungenauigkeiten/etc. Anpassungen an Anwendungsfällen oder gar neue Anwendungsfälle erfordern



# Anwendungsfalldiagramm

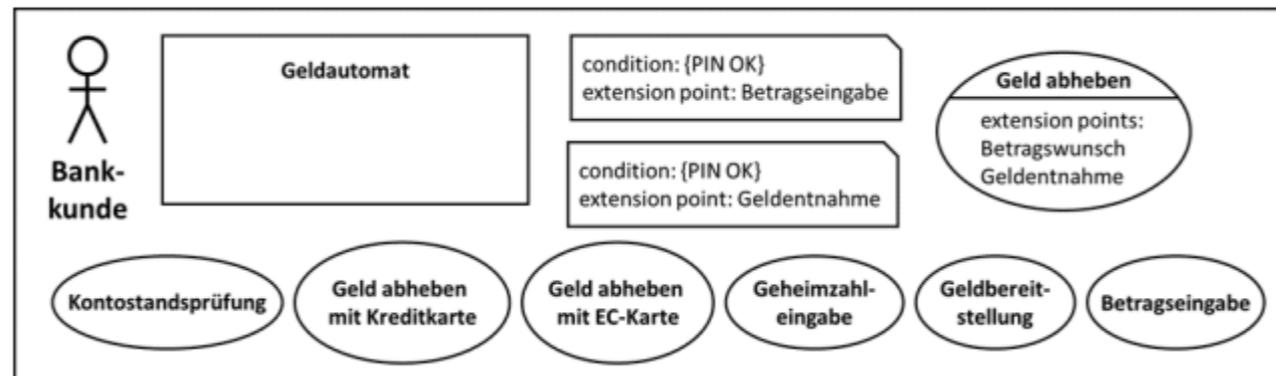
---

## Agenda

- Einführung
- Akteure
- Anwendungsfälle
- Beziehungen
  - zwischen Anwendungsfällen und Akteuren
  - zwischen Anwendungsfällen
  - zwischen Akteuren
- Beschreibung eines Anwendungsfalls
- Modellierung
- **Typische Modellierungsfehler**
- Beispiel: Informationssystem der Studienabteilung
- Zusammenfassung Elemente / Rückblick und Ausblick

# Aufgabe: Use Case Puzzle

- Zeichnen Sie ein UML Anwendungsfall Diagramm für den Prozess einer ATM einer Bank.
- Hierbei soll der User mit EC (bzw. Bank-) oder Kreditkarte Geld abheben können oder seinen Kontostand anzeigen.
- Folgende Elemente existieren:



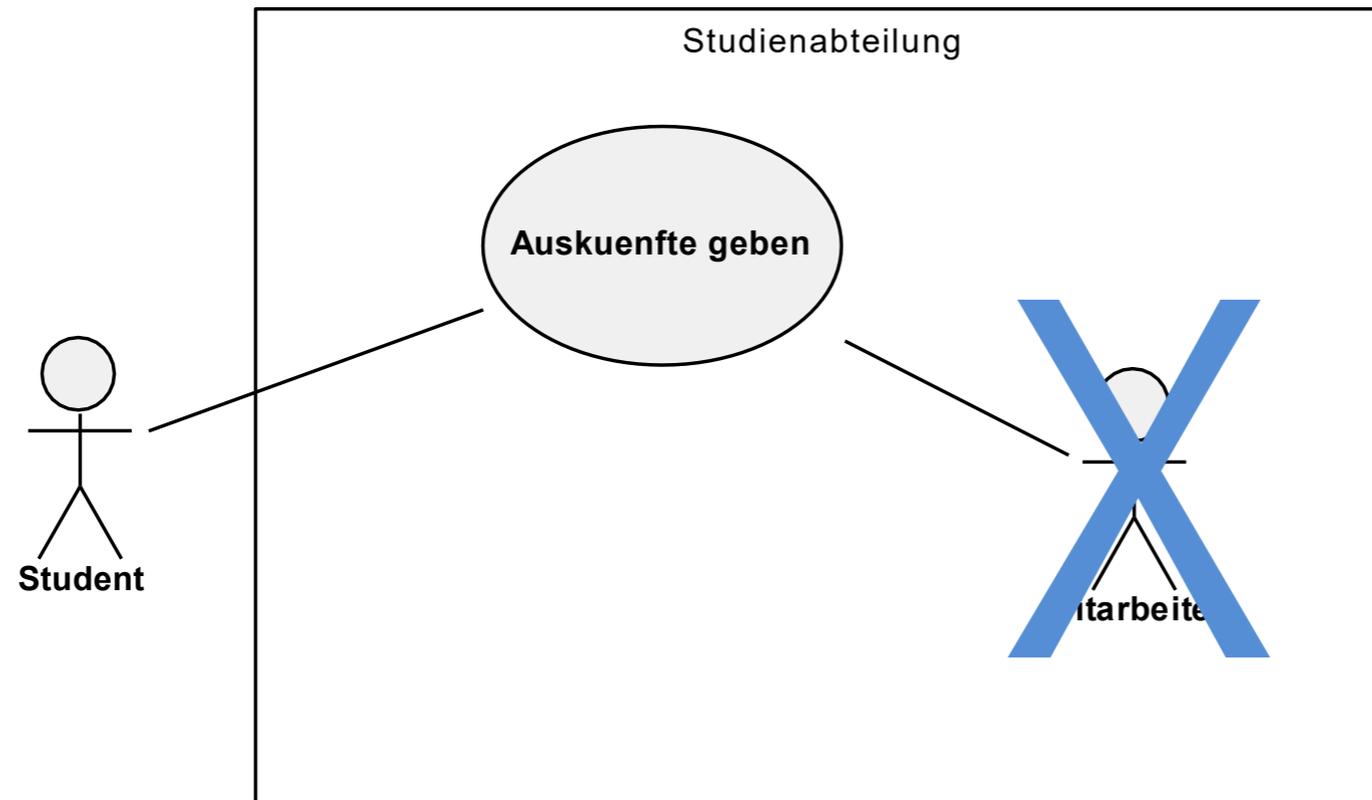
Sie ergänzen dazu die Assoziationen, Erweiterungs- und Include Beziehungen und die Generalisierungen



15 min

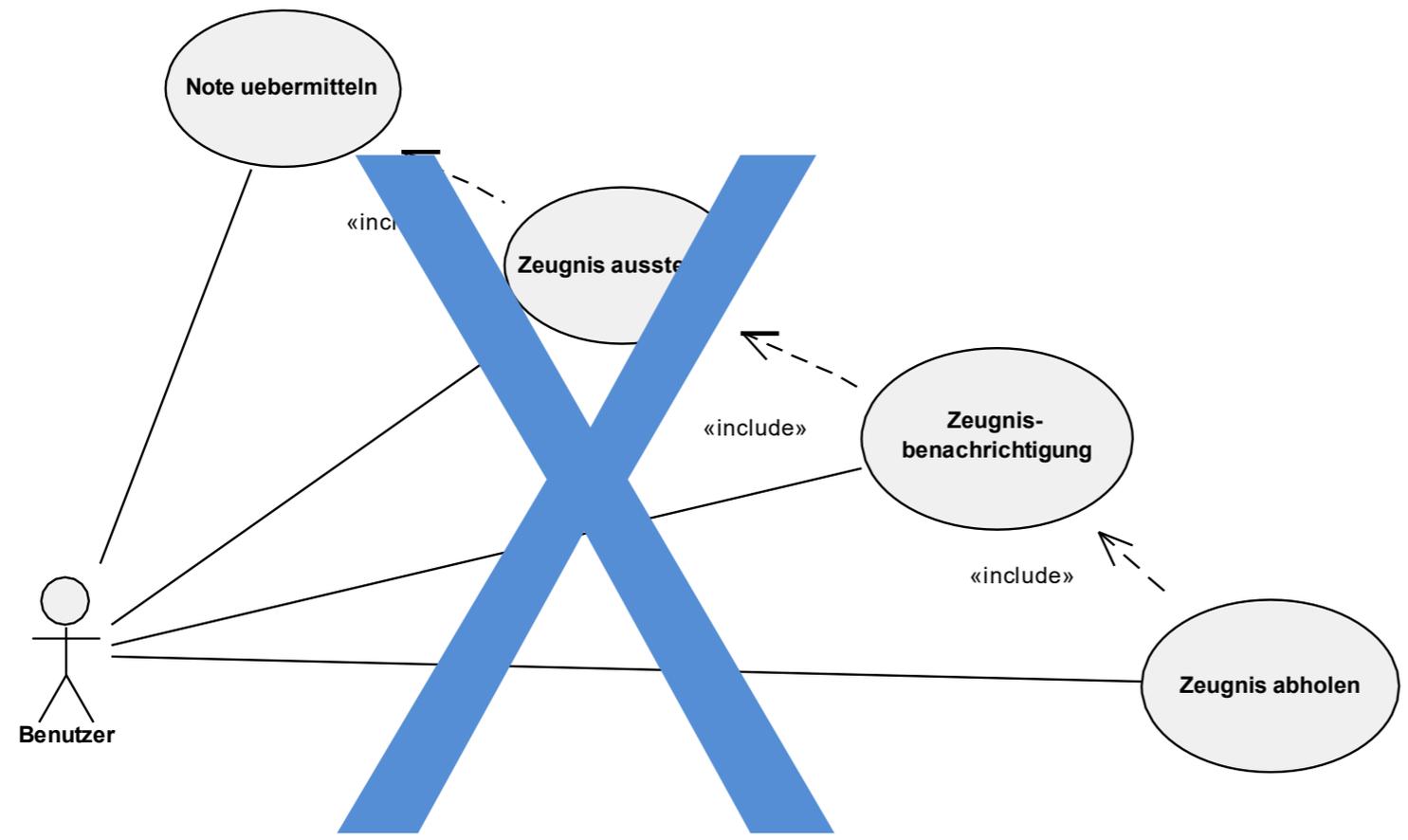
# Typische Modellierungsfehler (1/6)

Akteure werden innerhalb der Systemgrenze modelliert



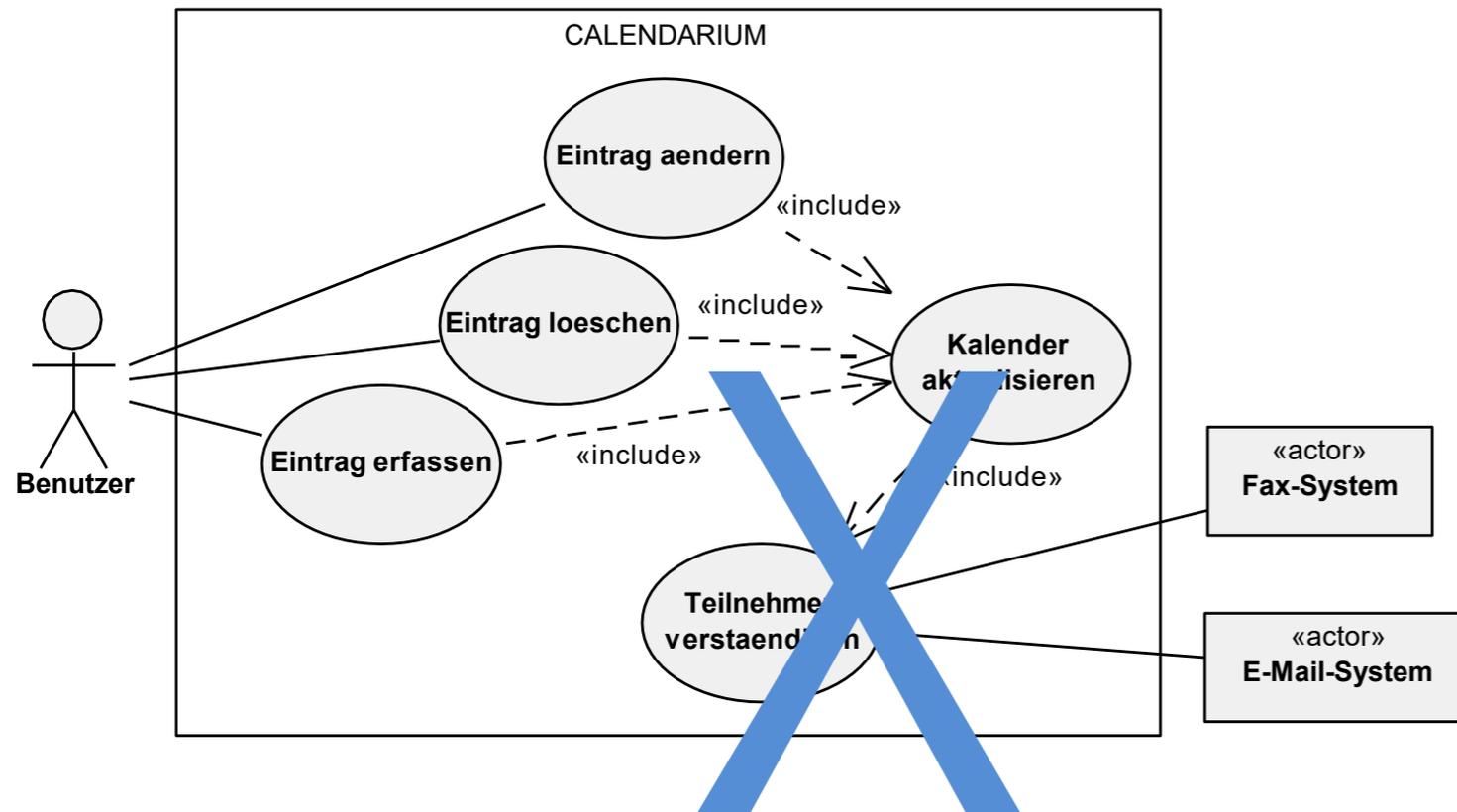
# Typische Modellierungsfehler (2/6)

Es werden Abläufe modelliert



# Typische Modellierungsfehler (3/6)

Es werden Abläufe modelliert



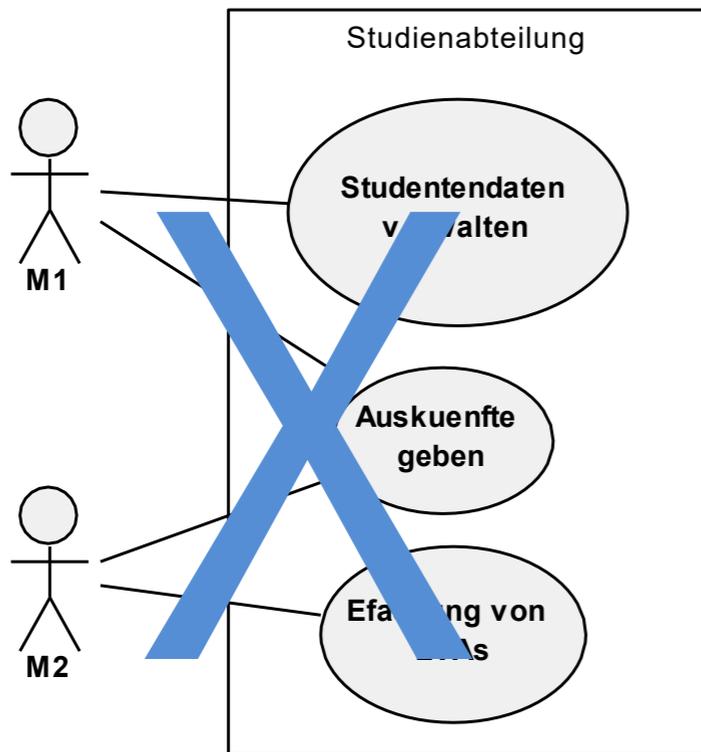
Obwohl der *Ablauf* so ist:

- 1) Eintrag erfassen
- 2) Kalender aktualisieren
- 3) Teilnehmer verständigen

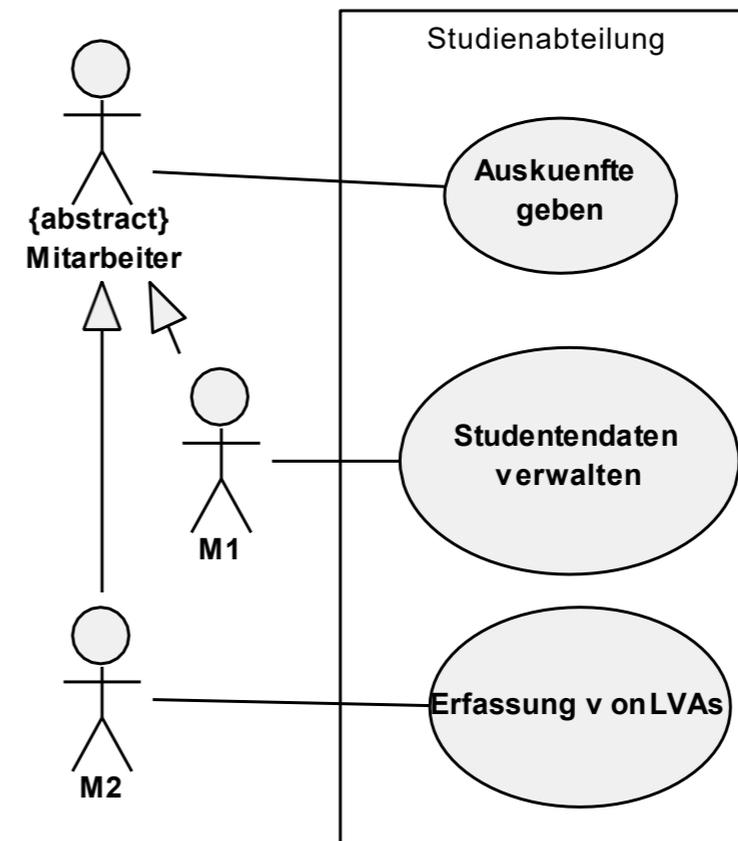
ist dieses Modell **falsch**, denn Teilnehmer verständigen ist nicht *Teil* von Kalender aktualisieren.

# Typische Modellierungsfehler (4/6)

Assoziationen der Akteure (und/oder) werden falsch modelliert

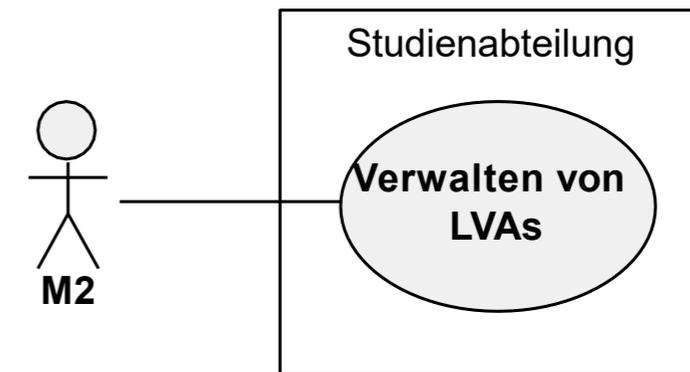
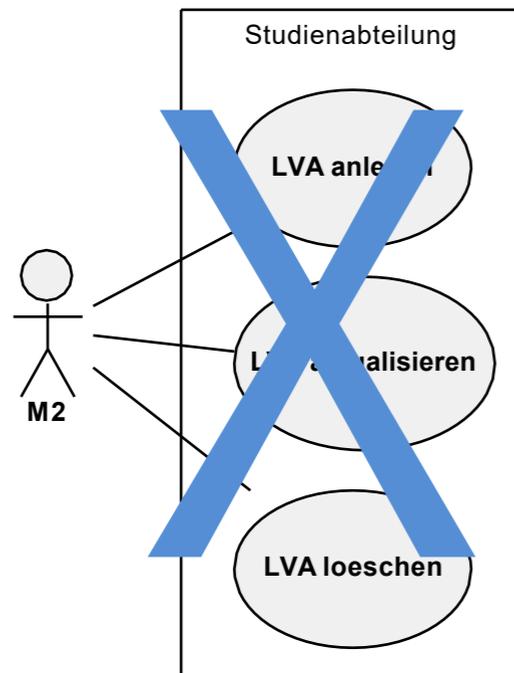


Richtig:



# Typische Modellierungsfehler (5/6)

Die Anwendungsfälle sind zu detailliert



„Richtiger“: Viele kleine Anwendungsfälle werden zu einem Anwendungsfall zusammengefasst, der die unterschiedlichen Möglichkeiten enthält



# Anwendungsfalldiagramm

---

## Agenda

- Einführung
- Akteure
- Anwendungsfälle
- Beziehungen
  - zwischen Anwendungsfällen und Akteuren
  - zwischen Anwendungsfällen
  - zwischen Akteuren
- Beschreibung eines Anwendungsfalls
- Modellierung
- Typische Modellierungsfehler
- **Beispiel: Informationssystem der Studienabteilung**
- Zusammenfassung Elemente / Rückblick

# Beispiel: Informationssystem der Studienabteilung

---

- **Ziel der Übung:** vereinfachte Darstellung des Informationssystems der Studienabteilung einer Universität als Anwendungsfalldiagramm auf Basis der verbalen Beschreibung
  - Erledigen Sie die Aufgabe in Kleingruppen
  - Sie haben 25 Minuten Zeit
  - Gemeinsame Ergebnisdiskussion
  - Verfassen Sie für einen Anwendungsfall Ihrer Wahl eine „ausführliche“ Anwendungsfallbeschreibung (Treffen Sie für diese Beschreibung Annahmen für in der verbalen Beschreibung fehlende Informationen)
- **Vorgehensweise – stellen Sie sich z.B. folgende Fragen:**
  - Wer sind die Benutzer/Akteure?
  - Welche Funktionen/Anwendungsfälle bietet das Informationssystem (auf Basis der verbalen Beschreibung)?
  - Welche Anwendungsfälle können zusammengefasst werden?
  - Welche Anwendungsfälle sind Bestandteil eines anderen Anwendungsfalls?
  - Welche Anwendungsfälle sind optional Bestandteile eines anderen?
  - Welche Akteure benutzen welche Anwendungsfälle?



**25 min**

# Beispiel: Informationssystem der Studienabteilung

---

- **Ziel der Übung:** vereinfachte Darstellung des Informationssystems der Studienabteilung einer Universität als Anwendungsfalldiagramm auf Basis der verbalen Beschreibung
- **Verbale Beschreibung des Beispiels**
  - Viele wichtige Verwaltungstätigkeiten einer Universität werden über die Studienabteilung abgewickelt. Studenten können über die Mitarbeiter der Studienabteilung immatrikulieren und inskribieren, sowie sich aber auch wieder vom Studium abmelden. Die Immatrikulation, d.h. das Einschreiben an einer Universität, beinhaltet das Inskribieren, d.h. das Anmelden zu einem Studium.
  - Studenten erhalten von der Studienabteilung ihre Zeugnisse, die von einem Mitarbeiter ausgedruckt werden. Zeugnisrelevante Daten werden durch Lehrende an die Studienabteilung übermittelt. Die Studenten werden automatisch durch ein Benachrichtigungssystem darüber informiert.
  - Es wird zwischen 2 Arten von Mitarbeitern unterschieden: a) solche, die sich ausschließlich mit der Verwaltung von Studentendaten befassen (StudentMA) und b) jene, die alle restlichen Aufgaben erfüllen (VerwaltungsMA), wobei aber alle Mitarbeiter (StudentMA und VerwaltungsMA) Auskünfte geben können.
  - VerwaltungsMA stellen Zeugnisse aus, sobald der/die Studierende diese abholt.
  - VerwaltungsMA können zudem Lehrveranstaltungen anlegen. Bei der Erfassung einer LVA kann ein Hörsaal reserviert werden.
  - Für ihre Tätigkeiten nutzen die Mitarbeiter der Studienabteilung ein Informationssystem: „SuperStudyOrg“

# Lösung: Informationssystem der Studienabteilung

- Beispiel „ausführliche“ Anwendungsfallbeschreibung

| Name:                        | Zeugnis drucken   |
|------------------------------|---|
| Kurzbeschreibung:            | Ein Mitarbeiter druckt auf Verlangen eines Studierenden sein Zeugnis für eine Lehrveranstaltung auf Papier aus.   |
| Vorbedingung:                | Zeugnisrelevante Daten wurden übermittelt.  |
| Nachbedingung:               | Zeugnis liegt dem Studenten in gedruckter Form vor.   |
| Fehlersituationen:           | Drucker ist defekt.   |
| Systemzustand im Fehlerfall: | Zeugnis wurde nicht gedruckt.   |
| Akteure:                     | VerwaltungsMA   |
| Trigger:                     | Student fordert ausgedrucktes Zeugnis an.   |
| Standardablauf:              | (1) Student kommt in Studienabteilung und fordert Zeugnis an.<br>(2) VerwaltungsMA gibt Matrikelnummer des Studenten ein.<br>(3) VerwaltungsMA wählt Zeugnis aus.<br>(4) VerwaltungsMA gibt Druckbefehl ein.<br>(5) System bestätigt, dass Zeugnis gedruckt wurde.<br>(6) Zeugnis wird Studenten übergeben. |
| Alternativabläufe:           | (1') Student fordert Zeugnis per E-Mail an.<br>(2-5) wie oben<br>(6') Zeugnis wird per Post versendet.  |

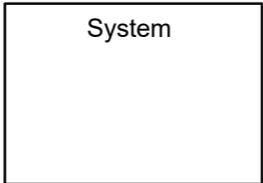
# Anwendungsfalldiagramm

---

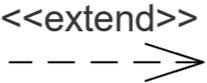
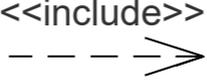
## Agenda

- Einführung
- Akteure
- Anwendungsfälle
- Beziehungen
  - zwischen Anwendungsfällen und Akteuren
  - zwischen Anwendungsfällen
  - zwischen Akteuren
- Beschreibung eines Anwendungsfalls
- Modellierung
- Typische Modellierungsfehler
- Beispiel: Informationssystem der Studienabteilung
- **Zusammenfassung Elemente / Rückblick**

# Anwendungsfalldiagramm - Elemente (1/2)

| <b>Name</b>    | <b>Syntax</b>  | <b>Beschreibung</b>  |
|----------------|--|--|
| Systemgrenze   |    | Grenze zw. dem eigentlichen System und den Benutzern des Systems |
| Anwendungsfall |   | vom System erwartetes Verhalten                                  |
| Akteur         |  | Rolle der Systembenutzer   |

# Anwendungsfalldiagramm - Elemente (2/2)

| Name            | Syntax   | Beschreibung  |
|-----------------|--|---|
| Assoziation     |    | Beziehung zwischen Anwendungsfällen und Akteuren                                  |
| Generalisierung |    | Vererbungsbeziehung von Anwendungsfällen und Akteuren                             |
| extend          |  | <i>A extends B</i> : opt. Verwenden von Anwendungsfall A durch Anwendungsfall B   |
| include         |  | <i>A includes B</i> : notw. Verwenden von Anwendungsfall B durch Anwendungsfall A |

# Willkommen im Sprint 2

## ✓ Sprint 2 - Wichtigsten Grundlagen und Konzepte Für Teilnehmer/innen verborgen

Für Teilnehmer/innen verborgen

### Backlog

- ✓ Als Professorin möchte ich Studierenden vermitteln, was Use Case Diagramme sind, damit sie sie einsetzen können.
- ✓ Als Professorin möchte ich Studierenden vermitteln, wie Use Case Diagramme zu modellieren sind, damit sie sie Anforderungen in Use Cases übersetzen können.
  - Als Professorin möchte ich Studierenden vermitteln, was Klassendiagramme sind, damit sie sie einsetzen können.
  - Als Professorin möchte ich Studierenden vermitteln, wie Klassendiagramme zu modellieren sind, damit sie sie Anforderungen in Use Cases übersetzen können.
  - Als Professorin möchte ich Studierenden vermitteln, was Aktivitätsdiagramme sind, damit sie sie einsetzen können.
  - Als Professorin möchte ich Studierenden vermitteln, wie Aktivitätsdiagramme zu modellieren sind, damit sie sie Anforderungen in Use Cases übersetzen können.
- ✓ Als Student:in möchte ich verstehen, wie Anforderungen erhoben werden, damit ich Kunden aufmerksam zuhören und richtige Fragen stellen kann
  - Als Student:in möchte ich Notation der Diagramme kennen und richtig einsetzen, damit ich mich auf die Inhalte konzentrieren kann
  - Als Student:in möchte ich klausurrelevante Inhalte möglichst früh kennenlernen, damit ich sie besser für die Klausur verinnerlichen kann

### Wichtig!

- Notation der Diagramme kennen und Diagramme verstehen
- Typischen Fehler erkennen und vermeiden
- Unterschied zwischen Include-, Extend-, Generalisierung Beziehungen verstehen und einsetzen
- Use Case Diagramme zeichnen (kommt)

Ansicht der Veranstaltung im [Moodle](#)