

## Wozu?

z.B. Kunde kommt mit unklarer Anforderungsbeschreibung und Sie müssen sich Klarheit über dessen eigentliche Vorstellungen und Erwartungen verschaffen auf einer Ebene, die auch für ihn verständlich ist → Use-Case-Diagramm

(Analyse und Verständnis der Funktion in Black-Box-Sicht)

- Sind intuitiv verständlich für die Kommunikation zwischen Entwicklern und Kunden
- Beschreibt was das System macht und nicht wie es funktioniert.
- Stellt die Sichtweise eines externen Beobachters dar
- Use-Case-Diagramme stellen die Anforderung an ein System dar
- Die durch das Diagramm dargestellten Szenarien können auch als Testfälle für die Software agieren

## Grundkomponenten:

- System:
- Rechteck mit Namen
  - auch mehrere ineinander geschachtelte Systeme möglich
  - dadurch Gliederung in Teilsysteme

- Use-Case:
- Sammlung von Aktionen
  - werden von einem Akteur ausgelöst
  - Ergebnis hat einen Nutzen für den auslösenden oder einen anderen Akteur

kein Use-Case:

- eine Teilfunktionalität, deren Zweck nur die Ausführung einer anderen Funktionalität ist

- Akteur:
- etwas, das ein Verhalten zeigt (z.B. Person, anderes System, Sensor, Timer, ...)

- Beziehungen:
- beschreibt den Zugriff des Akteurs auf die Funktionalität bzw. eine Antwort des Systems an einen Akteur

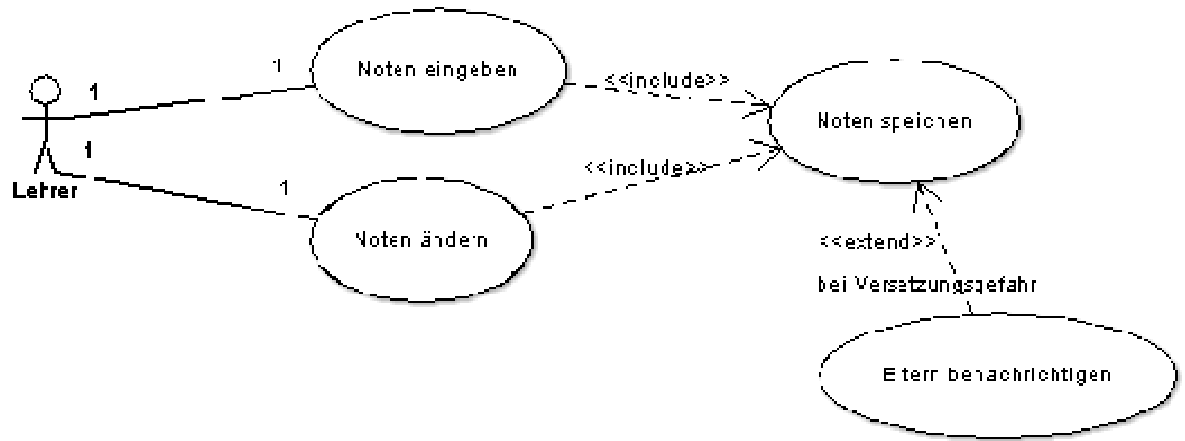
besondere Beziehungen zwischen AW-Fällen:

<<Include>> - Beziehung: Wenn der einbindende Use Case ausgeführt wird, muss auch der eingebundene Use Case ausgeführt werden.

<<extend>> - Beziehung: Use Case B kann Use Case A erweitern, muss dies aber nicht zwingend.

Bsp: Zeugniserstellung

Zeugnis



auch möglich: Generalisierungen zwischen Akteuren und zwischen AW-Fällen

## Problem:

Identifikation von Komponenten aus den Anforderungen

### *Identifikation der Akteure*

- Für wen wird das System entwickelt?
- Wer soll das System benutzen?
- Mit welchen anderen Systemen soll das geplante System interagieren?

### *Identifikation der Use Cases*

- Für welchen Zweck soll das System eingesetzt werden?
- Wofür will der spätere Benutzer es einsetzen?
- Durch welches externe Ereignis wird ein Use Case angestoßen?
- Welches Ergebnis liefert ein Use Case dem späteren Nutzer?

### *Identifikation der Beziehungen zwischen Akteuren und Use Cases*

- Auf welche Use Cases greift ein Akteur zu?
- Welche Antworten erhält er vom System?
- Über welche externen Ereignisse muss ein Use Case informiert werden?

### *Style Guide*

- Akteure und Use Cases sollten niemals in einem Diagramm isoliert sein.
- Ein Use Case Diagramm sollte nicht mehr als ca. 10 Use Cases enthalten. Bei zu vielen Use Cases wird das Diagramm unübersichtlich. Zu viele Use Cases in einem Diagramm sind ein Hinweis darauf, dass die Granularität der Use Cases nicht korrekt gewählt wurde. Sollte das System zu gross sein, ist es sinnvoll es in Teilsysteme zu zerlegen und in mehrere Use Case Diagramme zu verteilen.
- Kreuzungen sind zu vermeiden.

Mögliche Vorgehensweise bei textueller Anforderungsbeschreibung:

1. Rollen identifizieren --> Akteure
2. Vererbungsstruktur zwischen den Akteuren
3. Use Cases identifizieren
4. Include / Extend Beziehungen
5. Beziehung zu Akteuren