Seite 11. UML-Anwendungsfalldiagramm

UML-Anwendungsfalldiagramm

Diese Seite erklärt das **UML-Anwendungsfalldiagramm** so, dass du es schnell für IHK-Aufgaben und Software-Analyse anwenden kannst.

Ein UML-Anwendungsfalldiagramm zeigt:

> Wer benutzt ein System?
>
> Welche Funktionen bietet das System?
>
> Welche Akteure stehen mit welchen Anwendungsfällen in Verbindung?
>
> Welche Anwendungsfälle hängen voneinander ab?

Wichtig:

> Ein Anwendungsfalldiagramm zeigt nicht den inneren Programmablauf.
>
> Es zeigt die Sicht von außen auf das System.

Grafik 1: UML-Anwendungsfalldiagramm – Grundlagen

UML-Anwendungsfalldiagramm – Grundlagen

Diese Grafik zeigt die wichtigsten Bestandteile:

- **Akteur**
- **Systemgrenze**
- **Anwendungsfall / Use Case**
- **Assoziation**
- **«include»**
- **«extend»**
- **Generalisierung**

Wofür braucht man ein UML-Anwendungsfalldiagramm?

Ein UML-Anwendungsfalldiagramm wird verwendet, um die **Anforderungen an ein System aus Benutzersicht** darzustellen.

Es beantwortet Fragen wie:

Wer nutzt das System?

Welche Funktionen braucht der Benutzer?

Welche Systemfunktionen gehören zusammen?

Welche Funktion ist Pflichtbestandteil einer anderen Funktion?

Welche Funktion ist nur eine optionale Erweiterung?

Beispiel:

Ein Kunde kann Artikel suchen.

Ein Kunde kann eine Bestellung aufgeben.

Beim Aufgeben einer Bestellung muss eine Zahlung durchgeführt werden.

Optional kann ein Rabatt geprüft werden.

Daraus entstehen:

Akteur: Kunde

Use Case: Artikel suchen

Use Case: Bestellung aufgeben

Use Case: Zahlung durchführen

Use Case: Rabatt prüfen

Woran erkenne ich in einer Aufgabe, dass ein Anwendungsfalldiagramm gemeint ist?

Typische Signalwörter:

Signalwort / Formulierung	Hinweis
Akteur	externe Rolle oder Person
Benutzer	jemand nutzt das System
System	Systemgrenze ist wahrscheinlich wichtig
Funktion	Anwendungsfall / Use Case
Anwendungsfall	Use Case direkt gemeint
Anforderungen	oft Use-Case-Sicht
Benutzer kann ...	Akteur + Use Case
System soll ... ermöglichen	Use Case
Kunde, Admin, Mitarbeiter	typische Akteure
„stellen Sie die Nutzung des Systems dar“	Anwendungsfalldiagramm

Grundidee

Ein Anwendungsfalldiagramm zeigt die **Außensicht** auf ein System.

Es zeigt nicht:

```
Welche Klasse welche Methode hat.
Wie der Ablauf Schritt für Schritt funktioniert.
Welche Datenbanktabellen entstehen.
```

Sondern es zeigt:

```
Welche Akteure gibt es?
Welche Funktionen stellt das System bereit?
Welche Akteure nutzen welche Funktionen?
```

Merksatz:

```
Anwendungsfalldiagramm = Wer nutzt welche Funktion des Systems?
```

Akteur

Ein Akteur ist eine Rolle außerhalb des Systems.

Beispiele:

```
Kunde
Admin
Mitarbeiter
Lehrer
Schüler
Zahlungsdienst
E-Mail-System
```

Wichtig:

```
Ein Akteur muss nicht immer ein Mensch sein.
Auch ein externes System kann ein Akteur sein.
```

Beispiel:

```
Zahlungsdienst
E-Mail-Server
Versanddienstleister
```

Ein Akteur steht normalerweise **außerhalb der Systemgrenze**.

Systemgrenze

Die Systemgrenze wird als Rechteck dargestellt.

Innerhalb der Systemgrenze stehen die Anwendungsfälle.

Außerhalb der Systemgrenze stehen die Akteure.

Merksatz:

```
Akteure außen.
Use Cases innen.
Systemgrenze als Rechteck.
```

Beispiel:

```
Kunde       [ Shop-System ]
            (Artikel suchen)
            (Bestellung aufgeben)
```

Anwendungsfall / Use Case

Ein Anwendungsfall beschreibt eine Funktion des Systems aus Sicht eines Akteurs.

Darstellung:

```
( Bestellung aufgeben )
```

Also als **Ellipse**.

Gute Use-Case-Namen sind meistens kurze Tätigkeiten:

```
Artikel suchen
Bestellung aufgeben
Zahlung durchführen
Benutzer anmelden
Artikel verwalten
Passwort zurücksetzen
```

Wichtig:

```
Use Cases sollten einen Nutzen für einen Akteur haben.
```

Nicht ideal:

```
Datenbank speichern
Methode ausführen
SQL ausführen
```

Besser:

```
Bestellung speichern
Kundendaten verwalten
Rechnung erstellen
```

Assoziation

Eine Assoziation verbindet einen Akteur mit einem Anwendungsfall.

Darstellung:

```
Kunde ——— (Artikel suchen)
```

Bedeutung:

```
Der Akteur nutzt diesen Anwendungsfall.
```

Wichtig:

Eine Assoziation ist normalerweise eine einfache Linie.

«include»

`«include»` bedeutet:

Ein Anwendungsfall enthält einen anderen Anwendungsfall immer als Pflichtbestandteil.

Beispiel:

Bestellung aufgeben «include» Zahlung durchführen

Bedeutung:

Wenn eine Bestellung aufgegeben wird, muss die Zahlung durchgeführt werden.

Typische Fälle für `«include»`:

Anmelden
Zahlung durchführen
Berechtigung prüfen
Daten validieren

Wichtig:

«include» zeigt auf den eingebundenen Pflicht-Anwendungsfall.

Beispiel:

(Bestellung aufgeben) - - -«include»- - -> (Zahlung durchführen)

«extend»

`«extend»` bedeutet:

Ein Anwendungsfall erweitert einen anderen Anwendungsfall optional.

Beispiel:

```
Rabatt prüfen «extend» Bestellung aufgeben
```

Bedeutung:

```
Rabatt prüfen passiert nur unter bestimmten Bedingungen.
```

Zum Beispiel:

```
wenn ein Rabattcode eingegeben wurde
wenn der Kunde berechtigt ist
wenn eine Sonderaktion aktiv ist
```

Wichtig:

```
«extend» zeigt vom optionalen Erweiterungsfall auf den Basis-Anwendungsfall.
```

Beispiel:

```
(Rabatt prüfen) - - -«extend»- - -> (Bestellung aufgeben)
```

Unterschied «include» und «extend»

Beziehung	Bedeutung	Beispiel	Merksatz
«include»	Pflichtbestandteil	Bestellung aufgeben enthält Zahlung durchführen	muss immer passieren
«extend»	optionale Erweiterung	Rabatt prüfen erweitert Bestellung aufgeben	passiert nur manchmal

Merksatz:

```
include = immer dabei
extend = optional / nur bei Bedingung
```

Generalisierung

Generalisierung bedeutet:

```
Ein spezieller Akteur oder Use Case erbt von einem allgemeineren Akteur oder Use Case.
```

Beispiel bei Akteuren:

```
Admin ist ein spezieller Benutzer.
```

Darstellung:

```
Admin ———▷ Benutzer
```

Wichtig:

```
Das hohle Dreieck zeigt zur allgemeineren Rolle.
```

Also:

```
Admin → Benutzer
```

nicht umgekehrt.

Grafik 2: Beispiel – Shop-System

UML-Anwendungsfalldiagramm – Beispiel Shop-System

Diese Grafik zeigt ein Beispiel für ein Shop-System.

Es enthält:

```
Akteure:
Kunde
Admin

Use Cases:
Anmelden
Artikel suchen
Bestellung aufgeben
```

Zahlung durchführen

Rabatt prüfen

Artikel verwalten

Wichtig:

Kunde und Admin stehen außerhalb der Systemgrenze.

Die Anwendungsfälle stehen innerhalb der Systemgrenze.

Beispiel als Textbeschreibung

Aufgabe:

Ein Kunde kann Artikel suchen und eine Bestellung aufgeben.

Beim Aufgeben einer Bestellung muss eine Zahlung durchgeführt werden.

Vor der Bestellung kann optional ein Rabatt geprüft werden.

Ein Admin kann Artikel verwalten.

Daraus entsteht:

Akteur Kunde

Akteur Admin

Kunde – Artikel suchen

Kunde – Bestellung aufgeben

Admin – Artikel verwalten

Bestellung aufgeben «include» Zahlung durchführen

Rabatt prüfen «extend» Bestellung aufgeben

Akteure im Beispiel

Akteur	Rolle
Kunde	nutzt den Shop, sucht Artikel, gibt Bestellungen auf
Admin	verwaltet Artikel im Shop-System

Wichtig:

```
Akteure sind Rollen.
Nicht jede konkrete Person wird einzeln gezeichnet.
```

Also:

```
Kunde
```

nicht:

```
Max Müller
```

Use Cases im Beispiel

Use Case	Bedeutung
Anmelden	Benutzer meldet sich im System an
Artikel suchen	Kunde sucht Produkte
Bestellung aufgeben	Kunde erstellt eine Bestellung
Zahlung durchführen	Zahlung wird abgewickelt
Rabatt prüfen	Rabattcode oder Rabattbedingung wird geprüft
Artikel verwalten	Admin erstellt, ändert oder löscht Artikel

Warum ist „Zahlung durchführen“ ein «include»?

Weil die Zahlung beim Aufgeben einer Bestellung ein notwendiger Bestandteil ist.

```
Bestellung aufgeben
```

enthält immer:

```
Zahlung durchführen
```

Darum:

```
Bestellung aufgeben «include» Zahlung durchführen
```

Warum ist „Rabatt prüfen“ ein «extend»?

Weil ein Rabatt nicht immer geprüft werden muss.

Nur wenn zum Beispiel ein Rabattcode vorhanden ist, wird dieser zusätzliche Fall relevant.

Darum:

```
Rabatt prüfen «extend» Bestellung aufgeben
```

Merksatz:

```
Rabatt = optional
Zahlung = Pflicht
```

Anwendungsfalldiagramm vs. Aktivitätsdiagramm

Anwendungsfalldiagramm	Aktivitätsdiagramm
zeigt Funktionen aus Benutzersicht	zeigt Ablauf Schritt für Schritt
Akteure außen	Start, Aktionen, Entscheidungen
Use Cases als Ellipsen	Aktionen als abgerundete Rechtecke
keine genaue Reihenfolge	genaue Ablaufreihenfolge
gut für Anforderungen	gut für Prozesslogik

Merksatz:

```
Anwendungsfalldiagramm = Wer nutzt was?
Aktivitätsdiagramm = Was passiert danach?
```

Anwendungsfalldiagramm vs. Klassendiagramm

Anwendungsfalldiagramm	Klassendiagramm
zeigt Systemfunktionen	zeigt Programmstruktur
Akteure und Use Cases	Klassen, Attribute, Methoden
Sicht von außen	technische Struktur
Anforderungen	OOP-Modellierung

Merksatz:

> Use Case = Funktion aus Nutzersicht
>
> Klasse = Bauplan im Programm

Vorgehensweise in der Prüfung

Wenn du ein UML-Anwendungsfalldiagramm erstellen sollst, gehe so vor:

Schritt	Frage
1	Was ist das System?
2	Wo liegt die Systemgrenze?
3	Welche Akteure gibt es?
4	Welche Funktionen nutzt jeder Akteur?
5	Welche Use Cases gehören in das System?
6	Welche Assoziationen gibt es?
7	Gibt es Pflichtbestandteile?
8	Gibt es optionale Erweiterungen?
9	Gibt es allgemeinere und speziellere Akteure?

Typische IHK-Fehler

Fehler	Warum problematisch?
Akteur innerhalb der Systemgrenze	Akteure stehen außerhalb
Use Case außerhalb der Systemgrenze	Use Cases gehören ins System
technische Methoden als Use Cases	Use Cases sollen Nutzersicht zeigen
`include` und `extend` verwechselt	Pflicht und Option werden falsch dargestellt
Pfeilrichtung bei `include` falsch	`include` zeigt auf den eingebundenen Use Case
Pfeilrichtung bei `extend` falsch	`extend` zeigt vom Erweiterungsfall zum Basisfall
Systemgrenze vergessen	unklar, was zum System gehört
Akteure als konkrete Personen benannt	Akteure sind Rollen

Prüfungs-Merksätze

```
Akteur = externe Rolle
Use Case = Funktion des Systems
Systemgrenze = Rechteck um die Use Cases
Akteure stehen außerhalb
Use Cases stehen innerhalb
Use Cases werden als Ellipsen gezeichnet
Assoziation = einfache Linie
include = Pflichtbestandteil
extend = optionale Erweiterung
include zeigt auf den eingebundenen Use Case
extend zeigt auf den Basis-Use-Case
```

Mini-Beispiel 1

Aufgabe:

```
Ein Benutzer kann sich anmelden.
```

Lösung:

```
Akteur: Benutzer
Use Case: Anmelden

Benutzer — (Anmelden)
```

Mini-Beispiel 2

Aufgabe:

```
Ein Kunde kann eine Bestellung aufgeben.
Dabei muss eine Zahlung durchgeführt werden.
```

Lösung:

```
Kunde — (Bestellung aufgeben)

(Bestellung aufgeben) «include» (Zahlung durchführen)
```

Begründung:

```
Zahlung durchführen ist ein Pflichtbestandteil der Bestellung.
```

Mini-Beispiel 3

Aufgabe:

```
Ein Kunde kann bei einer Bestellung optional einen Rabattcode verwenden.
```

Lösung:

```
(Rabatt prüfen) «extend» (Bestellung aufgeben)
```

Begründung:

```
Rabatt prüfen ist optional.
Es passiert nur, wenn ein Rabattcode vorhanden ist.
```

Mini-Testfragen

1. Wofür verwendet man ein UML-Anwendungsfalldiagramm?

Man verwendet es, um darzustellen:

```
Welche Akteure ein System nutzen
und welche Funktionen das System anbietet.
```

Merksatz:

```
Wer nutzt was?
```

2. Wo stehen Akteure im Anwendungsfalldiagramm?

Akteure stehen außerhalb der Systemgrenze.

Sie gehören nicht zum System selbst.

3. Wie werden Use Cases dargestellt?

Use Cases werden als Ellipsen dargestellt.

Beispiel:

```
( Bestellung aufgeben )
```

4. Was zeigt die Systemgrenze?

Die Systemgrenze zeigt, welche Anwendungsfälle zum betrachteten System gehören.

Sie wird als Rechteck um die Use Cases dargestellt.

5. Was bedeutet «include»?

`«include»` bedeutet:

```
Ein Use Case enthält einen anderen Use Case verpflichtend.
```

Beispiel:

```
Bestellung aufgeben «include» Zahlung durchführen
```

6. Was bedeutet «extend»?

`«extend»` bedeutet:

```
Ein Use Case erweitert einen anderen Use Case optional.
```

Beispiel:

```
Rabatt prüfen «extend» Bestellung aufgeben
```

7. Was ist der wichtigste Unterschied zwischen include und extend?

```
include = Pflichtbestandteil
extend = optionale Erweiterung
```

8. In welche Richtung zeigt «include»?

`«include»` zeigt auf den eingebundenen Pflicht-Use-Case.

Beispiel:

```
Bestellung aufgeben → Zahlung durchführen
```

9. In welche Richtung zeigt «extend»?

`«extend»` zeigt vom optionalen Erweiterungsfall zum Basis-Use-Case.

Beispiel:

```
Rabatt prüfen → Bestellung aufgeben
```

10. Was ist ein typischer Fehler bei Anwendungsfalldiagrammen?

Ein typischer Fehler ist, technische Methoden als Use Cases zu zeichnen.

Nicht gut:

```
saveOrder()
```

Besser:

Bestellung aufgeben

Nächste Seite

Danach kommt die eigene Trainer-Seite:

UML-Anwendungsfalldiagramm-Trainer

Dort bauen wir den interaktiven Trainer mit Aufgaben zu:

Akteure erkennen

Use Cases benennen

Systemgrenze verstehen

Assoziationen eintragen

include und extend unterscheiden

Pfeilrichtung prüfen

Revision #1
Created 28 May 2026 11:07:27 by Admin
Updated 28 May 2026 11:10:28 by Admin