



Objektorientierte Analyse und Design

Anforderungsanalyse

Prof. Dr.-Ing. Michael Uelschen
Hochschule Osnabrück
Sommersemester 2021

Objektorientierte Analyse und Design

Anforderungsanalyse



— 00 Organisatorisches

— 01 Einführung¹

— 02 Anforderungsanalyse³

— 03 Design⁴

— 04 Entwurfsmuster³

— 05 Sonstiges¹

— Anforderungen

- Ermitteln,
- Formulieren,
- Prüfen und Bewerten,
- Verwalten

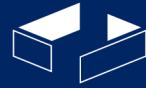
— Requirements Engineering
als eigenständige Disziplin im
Systems / Software
Engineering!



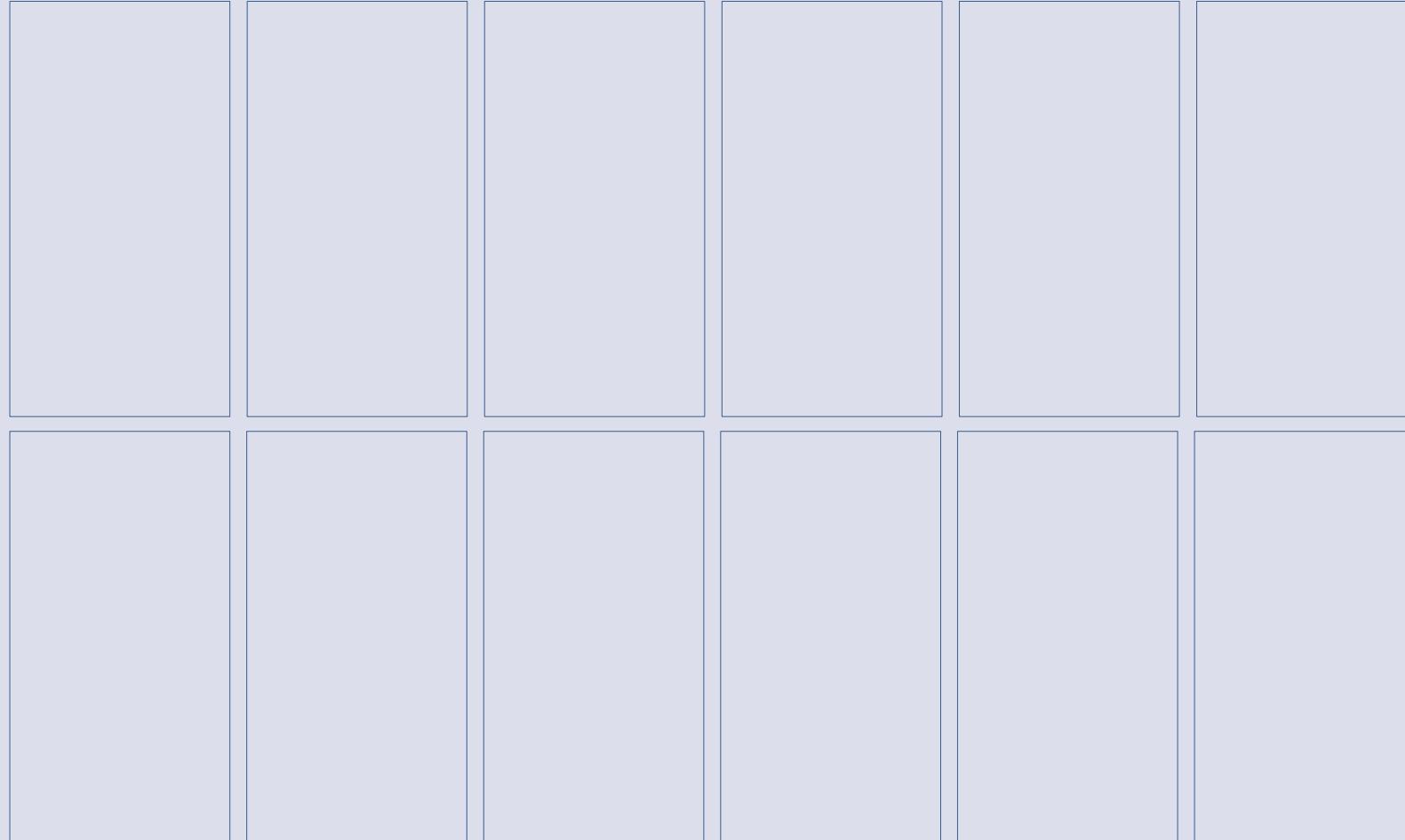
„Die Grenzen meiner Sprache sind die Grenzen meiner Welt.“

Ludwig Wittgenstein, Philosoph (1889-1951)

Objektorientierte Analyse und Design



HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES





Objektorientierte Analyse und Design

ANFORDERUNGEN

Objektorientierte Analyse und Design

Was ist eine Anforderungsanalyse?



- Systematischer Ansatz zur Spezifikation und Verwaltung von Anforderungen, um u.a. die folgenden Ziele zu erreichen:
 - **Relevante** Anforderungen kennen,
 - **Konsens** unter den Stakeholder zu erzielen,
 - Anforderungen nach Standards zu **dokumentieren**.
- Anforderungen an die Software (System) müssen
 - vollständig,
 - notwendig („**WAS statt WIE**“),
 - eindeutig,
 - richtig ("abgestimmt als Teil einer Zielhierarchie") sein.
- Anm.: Anforderungen müssen nicht notwendig vor Beginn des Entwurfs vollständig bestimmt werden.

Objektorientierte Analyse und Design

Ziele der Anforderungsanalyse



- Kennenlernen des Anwendungskontextes
- Zerlegen in Prozesse - Anwendungsfälle
- Vertiefter Kontakt mit dem Anwender
- Definition des Ziels
- Dokumentieren der ersten Anforderungen
- Vorgehensweise dabei: top down

Objektorientierte Analyse und Design

Was ist eine Anforderung? IEEE610



Eine Anforderung ist:

1. Eine Eigenschaft oder Fähigkeit, die von einem Benutzer (Person oder System) zur **Lösung eines Problems** oder zur **Erreichung eines Ziels** benötigt wird.
2. Eine Eigenschaft oder Fähigkeit, die ein System oder Teilsystem erfüllen oder besitzen muss, um **einen Vertrag**, eine Norm, eine Spezifikation oder andere formell vorgegebene Dokumente zu erfüllen.
3. Eine **dokumentierte Repräsentation** einer Eigenschaft oder Fähigkeit gemäß (1) oder (2).

Objektorientierte Analyse und Design

Einteilung von Anforderungen



Funktionale Anforderung

...ist eine Anforderung bezüglich des Ergebnisses eines **Verhaltens**, das von einer **Funktion** des Systems bereitgestellt werden soll.

Nicht-funktionale Anforderung

- Technologische Anforderungen
- Qualitätsanforderungen
- Anforderungen an die Benutzeroberfläche
- Anforderungen an sonstige Lieferbestandteile
- Anforderungen an durchzuführende Tätigkeiten
- Rechtlich-vertragliche Anforderungen

Objektorientierte Analyse und Design

Dokumentation von Anforderungen



- Projektleiter: „Das Wissen befindet sich doch in den Köpfen der Projektbeteiligten.“
- Entwickler: „Soll ich die Funktion programmieren oder soll ich jetzt statt dessen Dokumente schreiben?“
- Firmenleitung: „Dokumente mit mehr als 2 Seiten sind unnötig.“

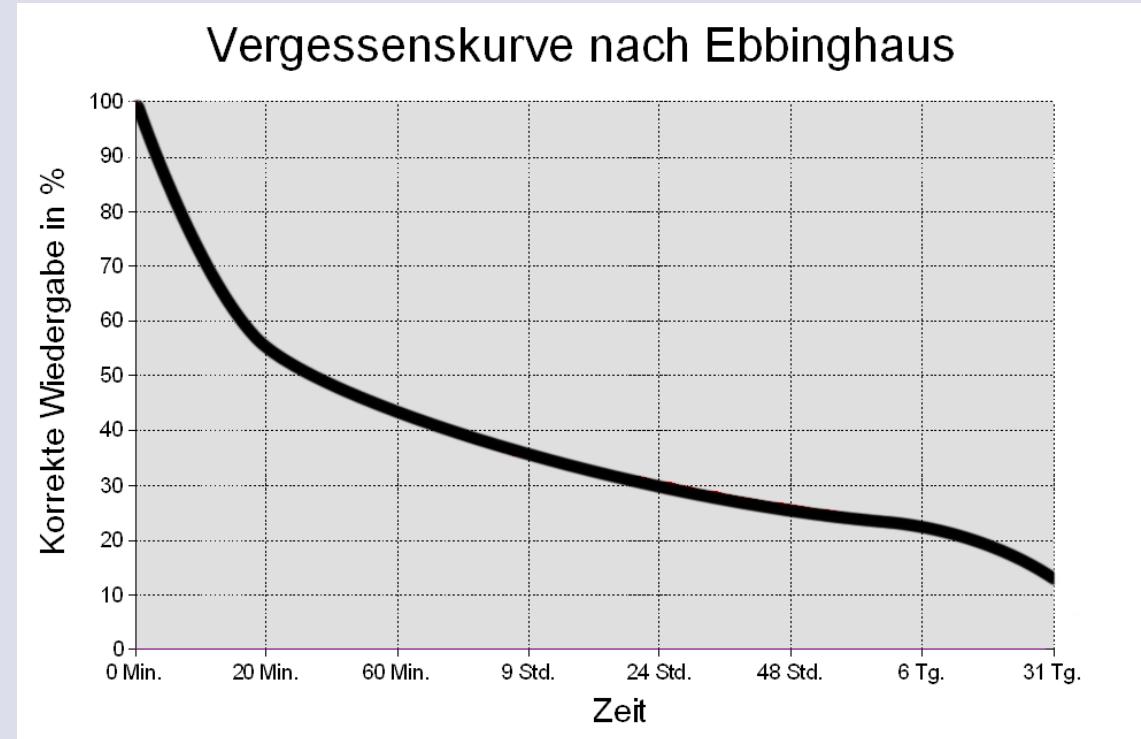
- Gründe für Dokumentation:
 - Wissen verfällt oder diffundiert.
 - Detailtiefe und Verständnis fehlt.
 - Verlust des Gesamtüberblicks.
 - Missverständnisse entstehen und bleiben.
 - Abweichende Informationen verteilen sich.

Objektorientierte Analyse und Design

Vergessenskurve nach Ebbinghaus 1885



- Professor für Psychologie
- Versuch
 - Auswendig lernen von Silben
 - Wiedergabe:
 - nach 20 min noch 60%,
 - nach 60 min noch 45%,
 - nach einem Tag noch 34%,
 - Kritik: Lernen von zusammenhangslosen Silben.
- Repetitio est mater studiorum!



<https://www.neuronation.de/gedaechtnistraining/vergessenskurve>

Objektorientierte Analyse und Design

Qualitätskriterien von Anforderungen



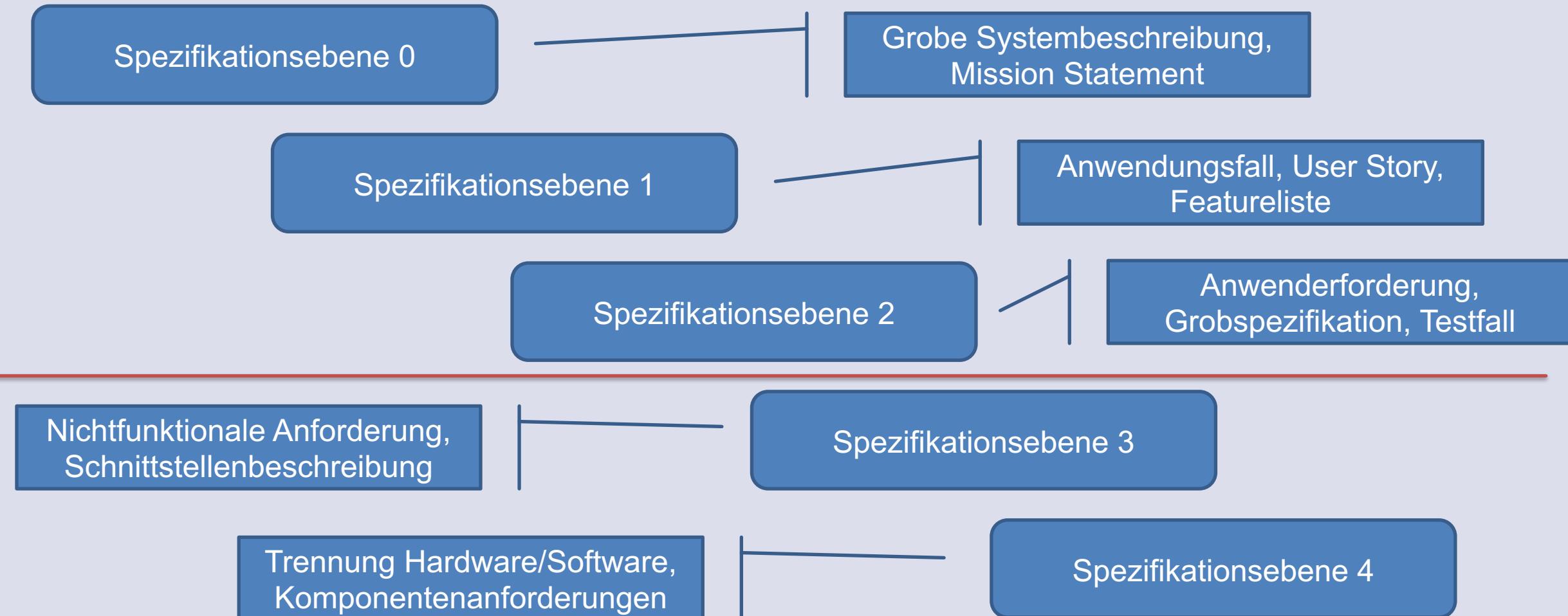
— Jede einzelne Anforderung muss den Qualitätskriterien genügen:

- **vollständig**: Funktion muss umfassend beschrieben sein (messbar!)
- **notwendig**: Funktion muss zur Zielerfüllung erforderlich sein.
- **atomar**: Jede Anforderung erhält ein Prozesswort (keine Konjunktionen!)
- **verfolgbar**: Die Anforderung ist auf seine Quelle zurückführbar.
- **technisch lösungsneutral**: Die Anforderung beschreibt das **Was**.
- **realisierbar**: Es muss eine Lösungsvariante für die Funktion existieren.
- **konsistent**: Anforderungen müssen widerspruchsfrei sein.
- **eindeutig**: Die Verständlichkeit der Anforderung ist nicht interpretierbar.
- **prüfbar**: Die Funktion muss testbar sein.



Objektorientierte Analyse und Design

Detaillierungsebenen



Objektorientierte Analyse und Design

Probleme in der Anforderungsanalyse



- Auftraggeber, Nutzer, Betreiber etc. sind häufig verschiedene Personen:
 - unterschiedliche Personen haben teilweise widersprüchliche Anforderungen.
- Die Effekte des angestrebten Systems sind schwer vorhersehbar.
- Die Anforderungen ändern sich im Laufe der Entwicklungszeit.
- Großer Umfang der Anforderungen.
- Komplexe Interaktion mit anderen Systemen.
 - Erste Aufgabe: Ermittlung der **Stakeholder**
 - Deutsch: Teilhaber, Interessengruppe
 - Definition: Jemand der Einfluss auf die Anforderungen hat, da er vom System betroffen ist (Systembetroffener).



Objektorientierte Analyse und Design

ERMITTeln

Objektorientierte Analyse und Design

Ermitteln – Systemanalyse



Wie kommt man zu den Anforderungen?

Schritte in der Systemanalyse

- Wer sind die **Personen**, die direkt oder indirekt mit dem System betroffen sind?
- Was ist das **Ziel**, welches das System erfüllen soll?
- Was ist der genaue **Umfang** sowie die **Umgebung** des Systems?

Objektorientierte Analyse und Design

Stakeholder und Persona



— Ein **Stakeholder** ist eine Person oder Organisation, welche (direkt oder indirekt) **Einfluss** auf die Anforderungen des betrachteten Systems hat:

- Nutzer und Betreiber,
- Entwickler und Architekten,
- Auftraggeber und Tester,
- ...

— **Fiktive** Personen werden im Usability-Engineering als **Persona** bezeichnet.
— Jede Persona hat eine eigene **Identität** und ein eigenes **Verhalten**.
— Persona können **zusätzliche Stakeholder** in der Anforderungsanalyse sein.

Objektorientierte Analyse und Design

Checkliste: Finden von Stakeholder 1



Endanwender

- Die größte und wichtigste Gruppe, liefert Großteil der fachlichen Ziele.
- Durchdachtes Auswahlverfahren für die Anwenderrepräsentanten nötig (Vertrauensbasis der gesamten Anwendergruppe berücksichtigen!)

Management des Auftragnehmers

- Gewährleisten die Konformität mit Unternehmenszielen und Strategien, sowie der Unternehmensphilosophie.
- Sind die Sponsoren!

Käufer des Systems

- Wer ist für die Kaufentscheidung verantwortlich?
- Liefer-Vertrags-Zahlungskonditionen?

Prüfer, Auditoren

- sind für Prüfung, Freigabe und Abnahme notwendig.

Entwickler

- Entwickler nennen die technologiespezifischen Ziele.

Objektorientierte Analyse und Design

Checkliste: Finden von Stakeholder 2



Wartungs- und Servicepersonal

- Wartung und Service muss unkompliziert und zügig durchzuführen sein
- Wichtig bei hohen Stückzahlen

Produktbeseitiger

- Wichtig, wenn ausgeliefertes Produkt nicht nur Software umfasst, Frage der Beseitigung (z.B. Umweltschutz), kann enormen Einfluss auf die Zielsetzung einer Produktentwicklung haben.

Schulungs- und Trainingspersonal

- Liefern konkrete Anforderungen zur Bedienbarkeit, Vermittelbarkeit, Hilfesystem, Dokumentation, Erlernbarkeit.

Marketing und Vertriebsabteilung

- Marketing und Vertrieb als interne Repräsentanten der externen Kundenwünsche und der Marktentwicklung.

Objektorientierte Analyse und Design

Checkliste: Finden von Stakeholder 3



Systemschützer

- Stellen Anforderungen zum Schutz vor Fehlverhalten von Stakeholdern

Standards und Gesetze

- vorhandene und zukünftige Standards/Gesetze berücksichtigen

Projekt-und Produktgegner

- Die Klasse der Projekt- und Produktgegner -vor allem zu Beginn des Projekts wenn möglich mit einbeziehen, sonst drohen Konflikte.

Kulturkreis

- setzt Rahmenbedingungen, z.B. verwendete Symbolik, Begriffe, ...
- Meinungsführer und die öffentliche Meinung.
- beeinflussen oder schreiben Ziele vor, Zielmärkte berücksichtigen.

Objektorientierte Analyse und Design

Ziele dokumentieren



- Ziele sind **abstrakte** Top-Level **Anforderungen** (Spezifikationsebene 0).
- Fragestellung: **Was ist der Nutzen für den Stakeholder?**
- Alle Anforderungen müssen Zielen zugeordnet werden!
- Verwende Zielschablone.

Ziel	Was soll erreicht werden?
Stakeholder	Welche Stakeholder sind in das Ziel involviert?
Auswirkungen für Stakeholder	Welche Veränderungen werden für die Stakeholder erwartet?
Randbedingungen	Welche unveränderlichen Randbedingungen müssen bei der Zielerreichung beachtet werden?
Abhängigkeiten	Ist die Zielverknüpfung mit anderen Zielen unmittelbar verknüpft?
Sonstiges	Was muss organisatorisch beachtet werden?

Objektorientierte Analyse und Design

Beispiel MäDn



Unterschiedliche Aspekte an einem Beispiel skizzieren.

Vorteil:

- Einfach und vollständig

Nachteil:

- Detailierungsgrad abweichend von realen Projekten, die im ersten Schritt weniger detailliert werden.

Praktikum:

- Einzelne Schritte an anderem Brettspielen erproben und vertiefen.

Objektorientierte Analyse und Design

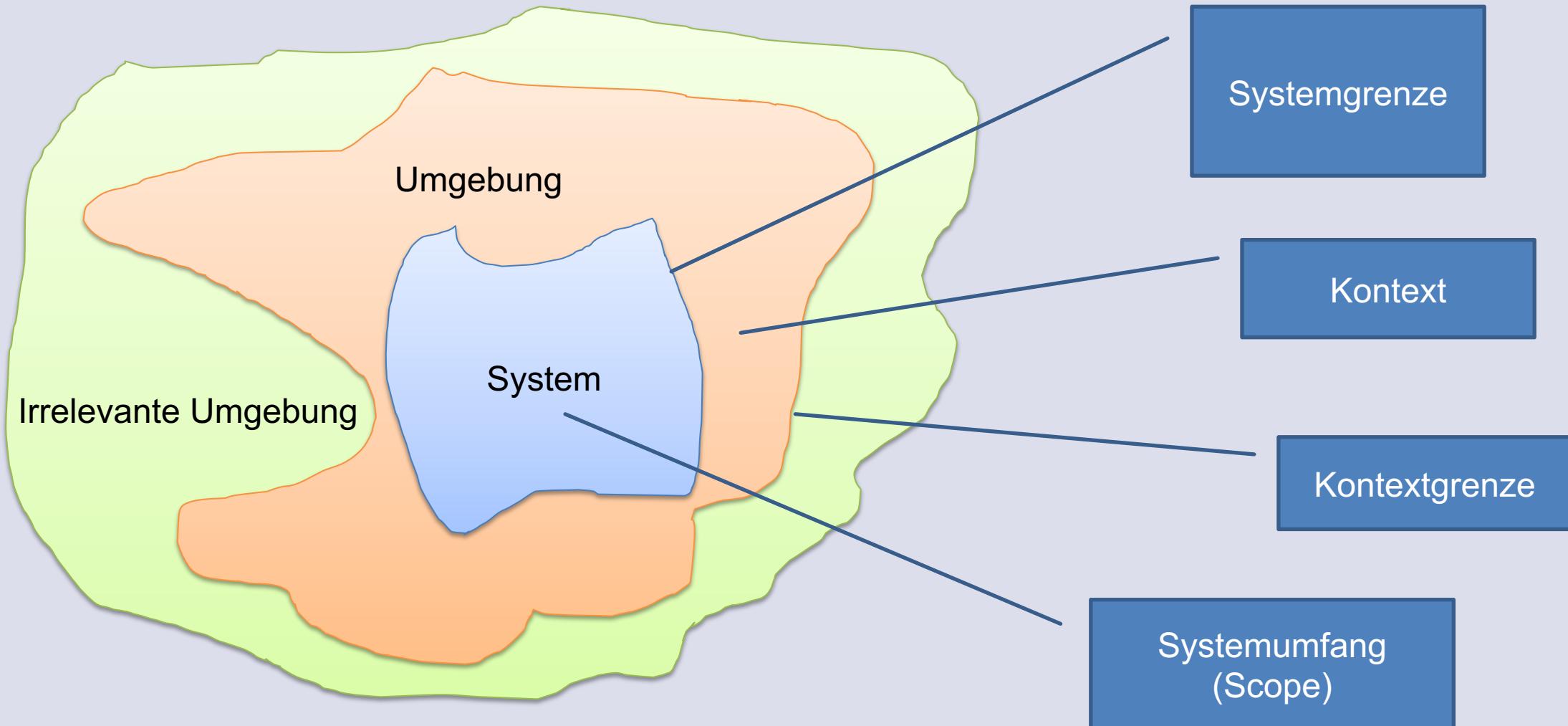
Systemgrenze festlegen



- Es sind die Aspekte zu ermitteln, die das System zu realisieren hat:
Systemgrenze.
- **Was nicht gefordert ist, wird nicht entwickelt?**
- Betrachte zusätzlich den **Kontext**, in dem das System zur Anwendung kommt.
- Kontext definiert Objekte, die mit dem System in Wechselwirkung treten.
- **Kontextgrenze** trennt die relevante von der irrelevanten Umgebung des Systems.
- Unterscheidung zwischen logischen und physikalischen Kontext (Embedded Systems).

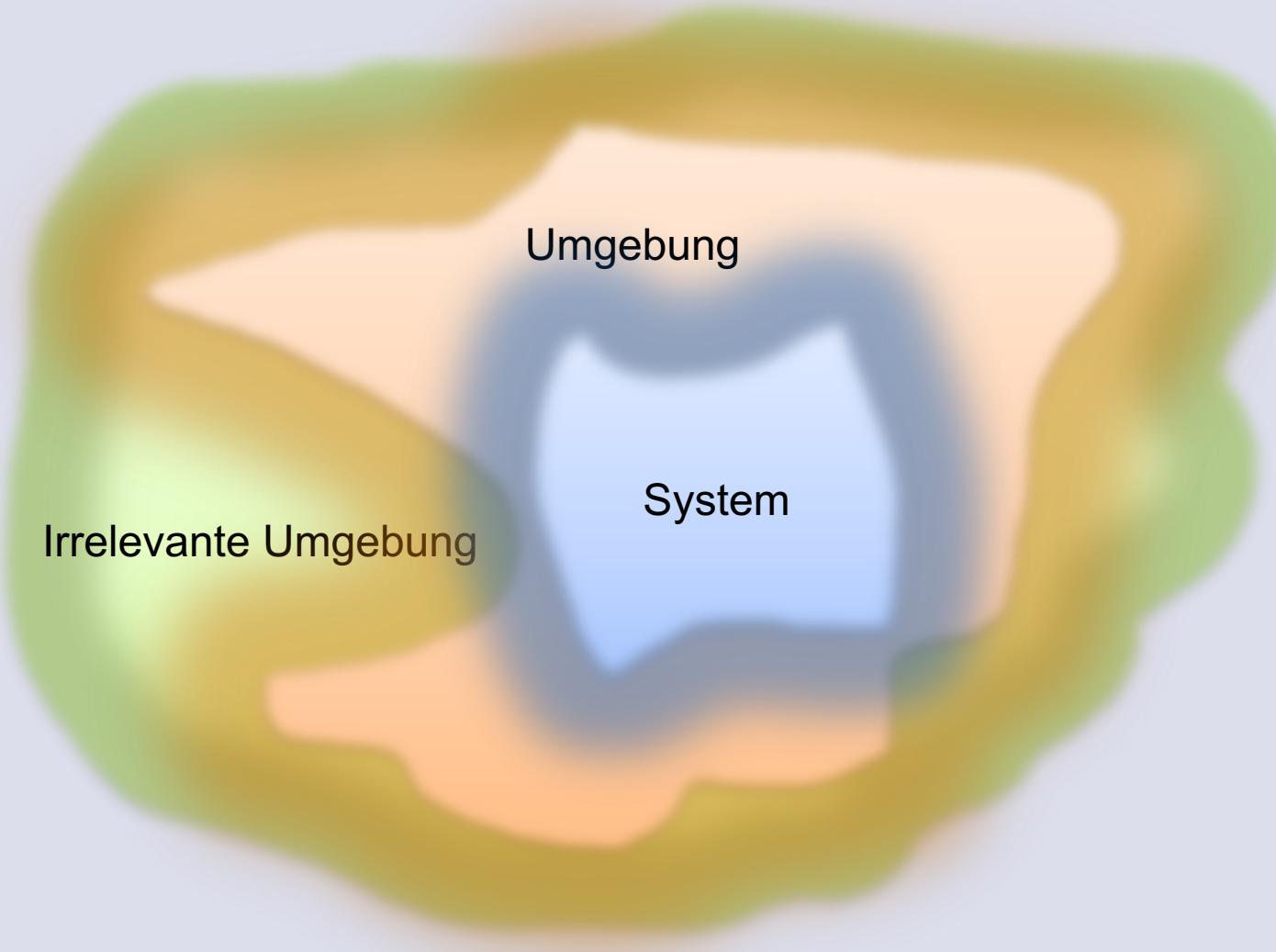
Objektorientierte Analyse und Design

Umfang, Kontext, Systemgrenze



Objektorientierte Analyse und Design

Unschärfe eliminieren



- **Exakte Grenzen** sind erst zum Ende Analyse bekannt.
- **Annahmen** explizit kennzeichnen (tbd.).
- Aufstellung einer **Negativliste**: Was gehört definitiv nicht zum System?

Objektorientierte Analyse und Design

Ermittlungstechniken



- Anforderungen liegen in den wenigsten Fällen umfassend, widerspruchsfrei, ... vor.
- Systematische **Anforderungsermittlung** ist erforderlich und entscheidend für den **Projekterfolg**.
- **Kommunikation** mit den Stakeholdern erforderlich!
- Möglichkeiten:
 - Kreativitätstechniken: Unbewusstes Wissen
 - Beobachtungstechniken: Arbeitsabläufe dokumentieren
 - Befragungstechniken: Wieso, weshalb, warum...
 - Artefaktebasierte Techniken: Fakten statt Rätselraten
 - Unterstützende Techniken



Brainstorming

- Gruppe bis zu 10 Teilnehmern sammeln Ideen und notieren diese ohne Kommentierung.
- Pro: viele Ideen in kurzer Zeit.
- Contra: Gruppendynamik

Methode 6-3-5

- 6 Teilnehmer schreiben jeweils 3 Ideen auf; nach 5 Minuten Rotation der Ideen mit dem Nachbarn
- Pro: Vereinfacht Gruppendynamik

- Contra: Weniger effektiv

Wechsel der Perspektive

- Unterschiedliche Varianten (u. a. Walt Disney-Methode)
- Teilnehmer setzen symbolisch unterschiedliche Hüte auf.
- Pro: Verlassen ein- oder festgefahren Wege
- Contra: „abgehobene“ Technik, ggfs. Akzeptanzprobleme



Feldbeobachtung

- Beobachtung der Stakeholder und Feststellung der Arbeitsabläufe (“Blick von außen”).
- Pro: Entlastung der Stakeholder, Methode kann sehr effektiv sein.
- Contra: Schwer beobachtbare Abläufe (z. B. Motorsteuerung) ungeeignet; Anwesenheit von Feldbeobachter problematisch.

Apprenticing

- Tätigkeiten werden unter Anleitung/Beobachtung Stakeholder selbstausgeführt.
- Pro: Entlastung Stakeholder; Vermeidung von Beobachtungssituation.
- Contra: Kritisches Arbeitsumfeld (z. B. Leitstand Kraftwerk) ungeeignet; zeit- und kostenintensiv.



Fragebogen

- Liste mit offenen und geschlossenen Fragen.
- Kann in Papierform oder elektronisch (große Gruppe von Stakeholder) erfolgen.
- Pro: Minimierung von Zeit- und Kostenaufwand ..
- Contra: Implizites Wissen kann schwer erfragt werden. Keine Möglichkeit für Rückfragen.

Interview

- Geführtes Gespräch (Fragen) mit Stakeholder und Protokollerstellung.
- Möglichkeit für Rückfragen und Klärung.
- Pro: Gezielte Nachfragen und Folgefragen; viele Details möglich.
- Contra: Hoher Aufwand, ggfs. Unterstützung mit Audio/Video.



Systemarchäologie

- Anforderungen auf Basis Dokumentation (z. B. Benutzerhandbuch) des Altsystems.
- Pro: Anforderungen Altsystem umfassend berücksichtigt.
- Contra: aufwendig, betrachtet nur das bestehende System; Qualität der Artefakte ist entscheidend.

Wiederverwendung

- Wiederverwendung der Anforderungen aus Vorgängerprojekt.
- Erfordert entsprechende Prozesse und Qualitätssicherung.
- Pro: Kosteneffiziente Methode, wenn Qualität Altanforderung hoch.
- Contra: Kontrolle der Anforderungen erforderlich (Vermeidung Übernahme von Fehlern).

Objektorientierte Analyse und Design

Unterstützende Techniken



- **SOPHIST-REgelwerk:** Vermeidung von Lücken und Auslassungen.
- **Workshop:** Direkte Kommunikation und Austausch in Gruppe
- **Mind Mapping:** Dokumentationstechnik zur Strukturierung/Visualisierung
- **Audio-/Videoaufzeichnung**
- **Use-Case-Modellierung:** Abstraktion zur Vermeidung von Detailabläufen.
- **Szenarien:** Modellhafte Interaktion zwischen Mensch und System:
- **Essenzbildung:** Fokussierung auf das Wesentliche.

Objektorientierte Analyse und Design

Grundregeln



- Schreiben Sie Ihre Anforderungen in **ganzen Sätzen**.
- Verwenden Sie die Begriffe konsistent und **vermeiden** Sie **Synonyme** (sinnverwandt) oder **Homonyme** (unterschiedliche Bedeutung).
- Verwenden Sie ein **Glossar** und verwenden Sie die darin definierten Begriffe.
- Formulieren Sie Prozesse durch **Vollverben**.



Objektorientierte Analyse und Design

FORMULIEREN

Objektorientierte Analyse und Design

Transformationsprozesse



Wahrnehmungstransformation



Eingeschränkte Wahrnehmung

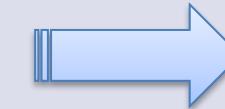


Realität



Lässt sich nicht
kompensieren!

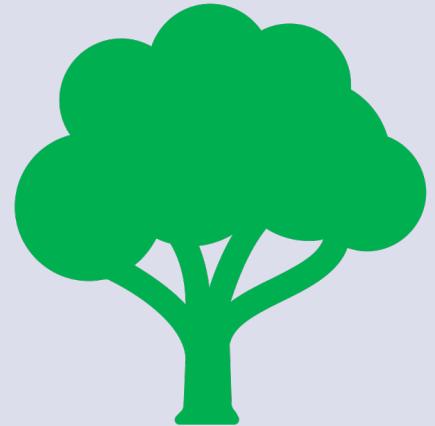
Darstellungstransformation



Persönliches Wissen



Lässt sich
kompensieren!



Sprachliche Darstellung



Tilgung

- Reduktion der Menge von Informationen (Selektion)

Generalisierung

- Übertragung von Erfahrungen auf ähnliche Situationen (Verallgemeinerung).

Verzerrung

- Veränderung von Ausdrücken entgegen der Situation (Ausschmückung oder Trübung).

Ziel: Effekte (er)kennen und Defekte in den Beschreibungen der Anforderungen beheben.

Objektorientierte Analyse und Design

Tilgung



- „Tilgung ist ein Prozess, durch den wir unsere Aufmerksamkeit selektiv bestimmten Dimensionen unserer Erfahrungen zuwenden und andere ausschließen.“
- Beispiel: Die Fähigkeit einer Person in einem Raum voller sprechender Menschen alle anderen Geräusche auszuschließen oder auszufiltern, um der Stimme einer bestimmten Person zuzuhören.
- Problematisch für Anforderungen:
 - implizite Annahmen
 - unvollständige Vergleiche.

Objektorientierte Analyse und Design

Generalisierung



- „Generalisierung ist der Prozess, durch den Elemente oder Teile eines persönlichen Modells von der ursprünglichen Erfahrung abgelöst werden, um dann die gesamte Kategorie, von der diese Erfahrung ein Beispiel darstellt, zu verkörpern.“
- Beispiel: Ein Kind verbrennt sich an einer heißen Herdplatte die Hand. Es sollte für sich die richtige..
 - ..Generalisierung aufstellen, dass es schmerhaft ist auf heiße Herdplatten zu fassen.
 - Problematisch für Anforderungen:
 - Universalquantoren,
 - Unvollständige Bedingungen.

Objektorientierte Analyse und Design

Verzerrung



- „Verzerrung ist der Prozess, der es uns ermöglicht, in unserer Erfahrung sensorischer Einzelheiten eine Umgestaltung vorzunehmen.“
- Die Verzerrung ist der Vorgang, durch den die Realität umgestaltet oder sogar verfälscht wird, so dass beim Betrachter ein Zerrbild entsteht.
- Problematisch für Anforderungen:
 - Korrektheit von Sachverhalten schwer zu erkennen

Objektorientierte Analyse und Design

Dokumentationstechniken



— Aufgabe: Wissen
(Anforderungen) zu Papier
bringen.

— Fragestellungen:

- **Was** sind die Hauptaufgaben
des Systems?
- Wer ist an den Aufgaben
beteiligt?
- Welche Schritte gehören zur
Aufgabenerfüllung?

— **Notation** benennt vereinbarte
symbolische Zeichen

- textuell (nicht/semi-formal)
 - Prosa, MASTeR-Plan
- graphisch (semi-formal)
 - UML, BPMN
- mathematisch (formal)

— Generell: **Nummerierung** der
Anforderungen etc. sinnvoll für
einfache Kommunikation!

Objektorientierte Analyse und Design

Glossar



- Zentrales Hilfsmittel der Anforderungsanalyse:
 - Fachbegriff – Erklärung
- Wichtig: Fachbegriff kann auch Halbsatz sein.
- Kann detaillierte Erklärungen oder Referenzen auf Fachliteratur enthalten.
- Muss von Auftraggebern und Entwicklern verstanden werden.

Objektorientierte Analyse und Design

Unified Modeling Language (UML)



Repräsentation von UML-Elementen:

- **Grafische Notation** (Diagramm)
- XML/XMI
- ...

Unterstützung von **Sichten**:

Ein Modellelement wird in unterschiedlicher Weise mehrfach dargestellt.

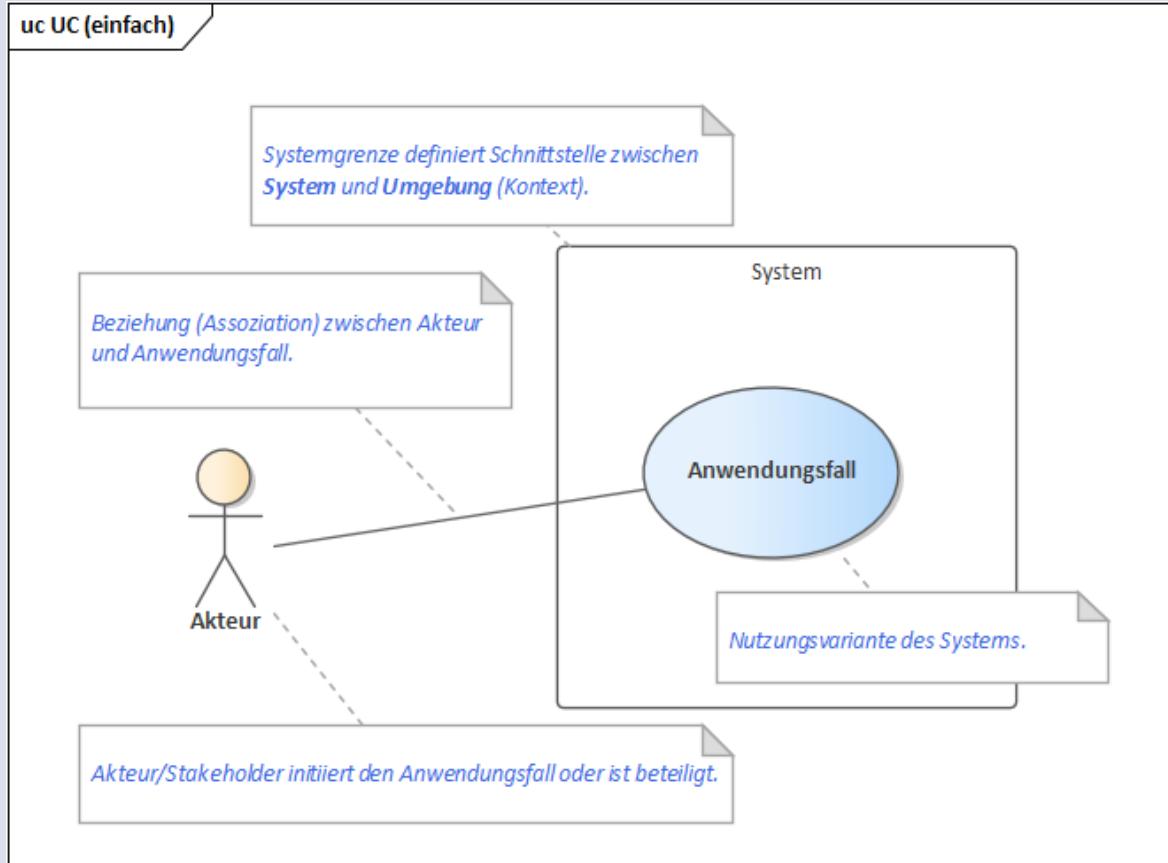
Häufig genutzte Diagramme:

- Anwendungsfalldiagramm
- Aktivitätsdiagramm
- Sequenzdiagramm
- Zustandsdiagramm
- Klassendiagramm

Wichtig: Diagramme können in der **Analyse** (Was?) und im **Design** (Wie?) eingesetzt werden.

Objektorientierte Analyse und Design

Anwendungsfalldiagramm (Use Case)



Akteur...

- ist eine außerhalb des Systems befindliche Einheit.

Anwendungsfall (Use Case)...

- beschreibt eine von mehreren Nutzungsvarianten eines Systems.

Beziehung...

- stellt die Verbindung zwischen Akteur und Anwendungsfall dar. Es enthält Ereignis- und Datenflüsse.

Objektorientierte Analyse und Design

Checkliste Anwendungsfalldiagramm



Frage	Hinweise
Gibt es mindestens einen Akteur?	Akteure sind häufig Rollen (Personen) oder andere, verbundene Systeme.
Ist eine Systemgrenze definiert?	Die Systemgrenze grenzt das zu entwickelnde System ab.
Ist der Anwendungsfall in sich abgeschlossen?	Ein Anwendungsfall ist eine vollständige Nutzungsvariante oder Szenario.
Beschreibt der Anwendungsfall aus der Sicht des Akteurs?	Ein Anwendungsfall ist die Sicht von außen auf das System.
Gibt es eine Beziehung zwischen Anwendungsfall und einem Akteur?	Jeder Anwendungsfall muss mindestens einen Nutzer haben.
Gibt es Akteure, die identische Anwendungsfälle haben?	Zwei Akteure, die identische Anwendungsfälle haben, sind identisch.

Objektorientierte Analyse und Design

Verfeinerung Anwendungsfälle



- Im ersten Schritt beschreiben die Anwendungsfälle nur die Hauptaufgaben des Systems.
- Zur Dokumentation der Anwendungsfälle gehört zunächst nur eine grobe Beschreibung des Inhalts (maximal 5 Sätze).
- Im nächsten Schritt wird dieser Inhalt konkretisiert.
- Dabei ist es sinnvoll, auf eine Dokumentationsschablone zurück zu greifen (oder eine für das Projekt zu entwickeln).
- Im ersten Schritt wird nur der typische Ablauf des Anwendungsfalls ohne danach mit Alternativen beschrieben.

Objektorientierte Analyse und Design

Schablone zur Dokumentation 1/2



<i>Titel des Anwendungsfalls¹</i>	Kurzer prägnanter Name/Beschreibung, meistens bestehend aus Verb und Nomen.
<i>Nummer¹</i>	Eindeutige Nummer zur Verwaltung/Identifikation (sollte von der eingesetzten Entwicklungsumgebung vergeben werden).
<i>Paket²</i>	Bei sehr komplexen Systemen können Anwendungsfälle in Teilaufgabenbereiche zusammengefasst werden, die in der UML als Pakete dargestellt werden.
<i>Autor/in¹</i>	Name des Erstellers/Mitarbeiters dieser Beschreibung.
<i>Version¹</i>	Aktuelle Versionsnummer (inkl. Änderungshistorie).
<i>Kurzbeschreibung¹</i>	Kurze Beschreibung, was mit dem Anwendungsfall auf welchem Weg erreicht werden soll.
<i>Beteiligte Akteure (Stakeholder)¹</i>	Auflistung der beteiligten Akteure (Wer initiiert den Anwendungsfall?)
<i>Fachverantwortliche/r¹</i>	Name des Mitarbeiters, der auf der Fachebene (Auftraggeber) für Rückfragen zur Verfügung steht und Entscheidungen herbeiführen kann.
<i>Referenzen²</i>	Verweis und Nennung aller Informationen (Quellen), die in weiteren Schritten im Entwicklungsprozess herangezogen werden können oder müssen (z.B. Gesetze, Normen oder Dokumentationen existierender Systeme).
<i>Vorbedingungen²</i>	Voraussetzungen und Randbedingungen, die erfüllt sein müssen, damit der Anwendungsfall realisiert werden kann.
<i>Nachbedingungen²</i>	Beschreibung der Ergebnisse, die nach dem Realisieren des Anwendungsfalls entstehen.

Objektorientierte Analyse und Design

Schablone zur Dokumentation 2/2



	weiteren Schritten im Entwicklungsprozess herangezogen werden können oder müssen (z.B. Gesetze, Normen oder Dokumentationen existierender Systeme).
<i>Vorbedingungen</i> ²	Voraussetzungen und Randbedingungen, die erfüllt sein müssen, damit der Anwendungsfall durchgeführt werden kann.
<i>Nachbedingungen</i> ²	Beschreibung des Ergebnisses nach Durchführung des Anwendungsfall. Hierzu sind ggfs. auch alternative Abläufe zu betrachten (z. B. Fehlerfall).
<i>Typischer Ablauf</i> ²	Tabellarische Auflistung der einzelnen Schritte (Aktivitäten, Aktionen), die der Anwendungsfall typischer Weise durchläuft.
<i>Alternative Abläufe</i> ³	Ergänzung des typischen Ablaufes um alternative Schritte.
<i>Kritikalität</i> ³	Festlegung des Anwendungsfall in Bezug auf das Gesamtsystem.
<i>Verknüpfungen</i> ³	Auflistung der Zusammenhänge zu anderen Anwendungsfällen.
<i>Funktionale Anforderungen</i> ⁴	Auflistung der funktionalen Anforderungen, die aus diesem Anwendungsfall abgeleitet werden.
<i>Nicht-funktionale Anforderungen</i> ⁴	Zusätzlich Auflistung der nicht-funktionalen Anforderungen.

Hinweis: Die hochgestellten Ziffern geben die möglichen Iterationen (Reihenfolge) der Bearbeitung der Schablone an.

Objektorientierte Analyse und Design

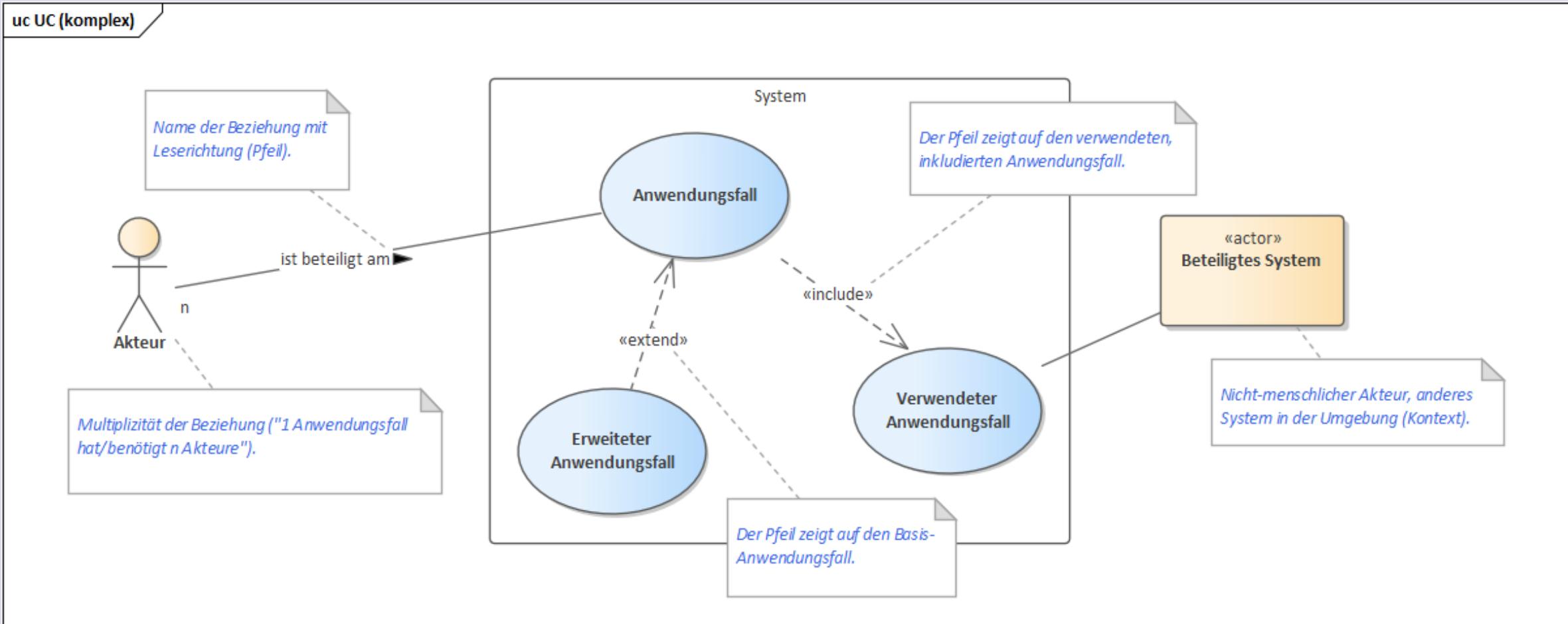
Hinweise zur Dokumentation



- Verwende eine sinnvolle Bezeichnung, die aus einem Substantiv und einem aktiven Verb („Antrag erfassen“) oder dem zugehörigen Gerundium („Antragserfassung“) besteht!
- Definiere zuerst den fachlichen Auslöser und das fachliche Ergebnis, um Anfang und Ende des Anwendungsfalls festzulegen!
- **Formuliere so abstrakt wie möglich und so konkret wie nötig!**
- Standardisiere die Sprache in der Dokumentation! Glossar!
- Anwendungsfälle eignen sich **nicht** zur **funktionalen Zerlegung**, d.h. keine einzelnen Schritte, Operationen oder Transaktionen (bspw. „Vertrag drucken“, „Kunden-Nr. erzeugen“ etc.), sondern zusammenhängende Abläufe (bspw. „Neuen Kunden aufnehmen“). Es wird **keine Ablaufreihenfolge** definiert (andere Ausdrucksmittel).
- Anwendungsfälle belassen das **Sprachmonopol** beim Stakeholder, wodurch diese besser diskutierbar sind.

Objektorientierte Analyse und Design

Erweiterungen in UML



Objektorientierte Analyse und Design

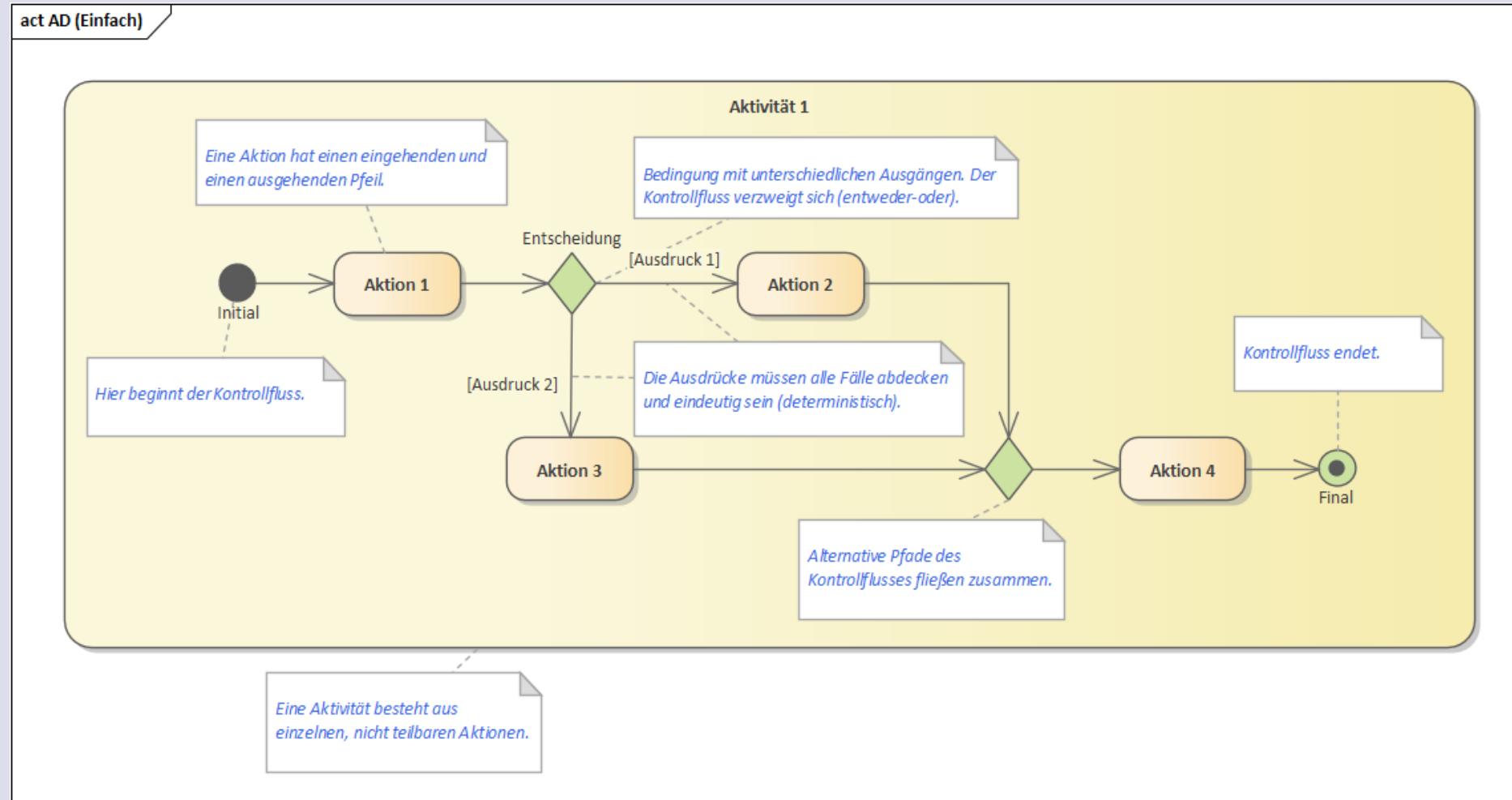
Aktivitätsdiagramm



- Use Case Diagramm zeigt nur **statische Sicht** auf das System.
- Aktivitätsdiagramm erweitert den Anwendungsfall um zeitliche Komponente (**dynamische Sicht**).
- Modelliere zu jedem Use Case genau ein Aktivitätsdiagramm:
Schritte → Aktionen.
- Zerlege die Aktionen mit einem Aktivitätsdiagramm, so dass diese stets genau einen fachlichen Arbeitsschritt repräsentieren.
- Ergänze den Ablauf um alle bekannten fachlichen Ausnahmen, fachlichen Fehler und fachlichen Ablaufvarianten.

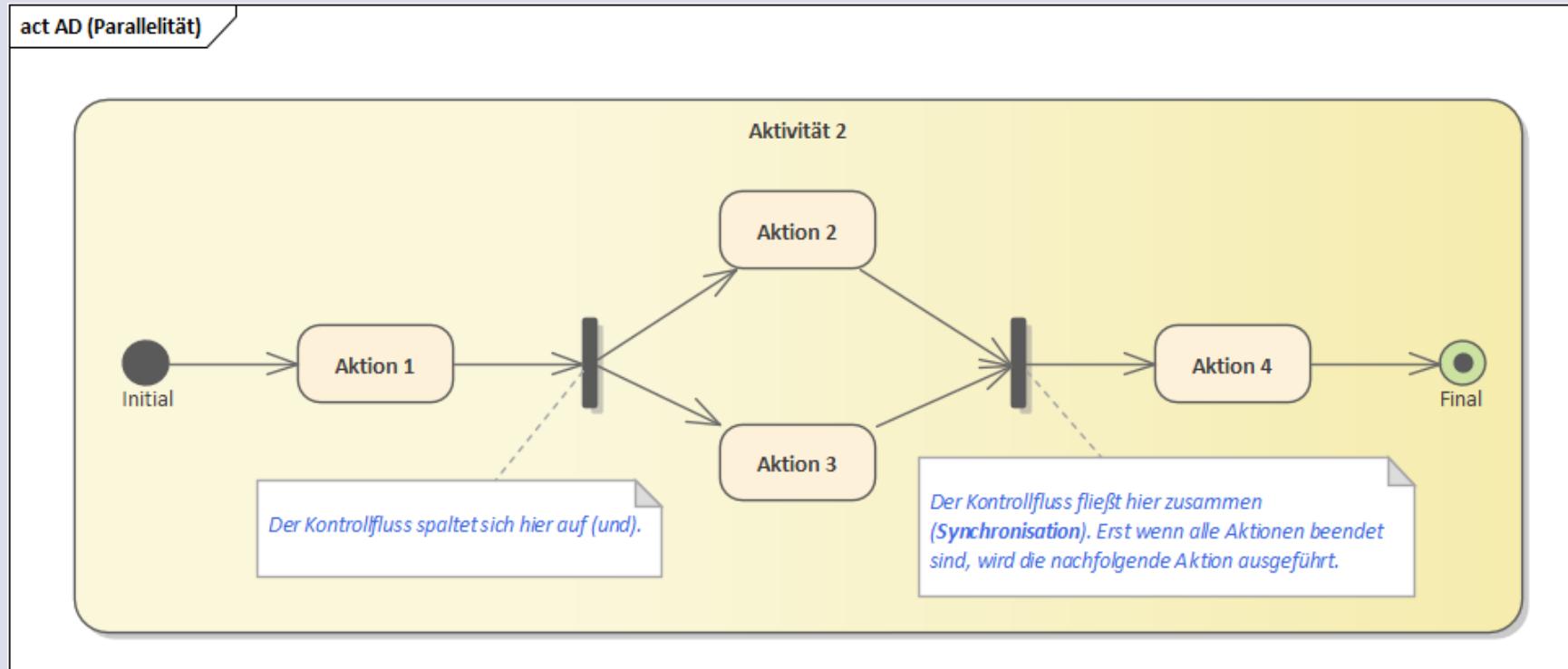
Objektorientierte Analyse und Design

Aktivitätsdiagramm in UML



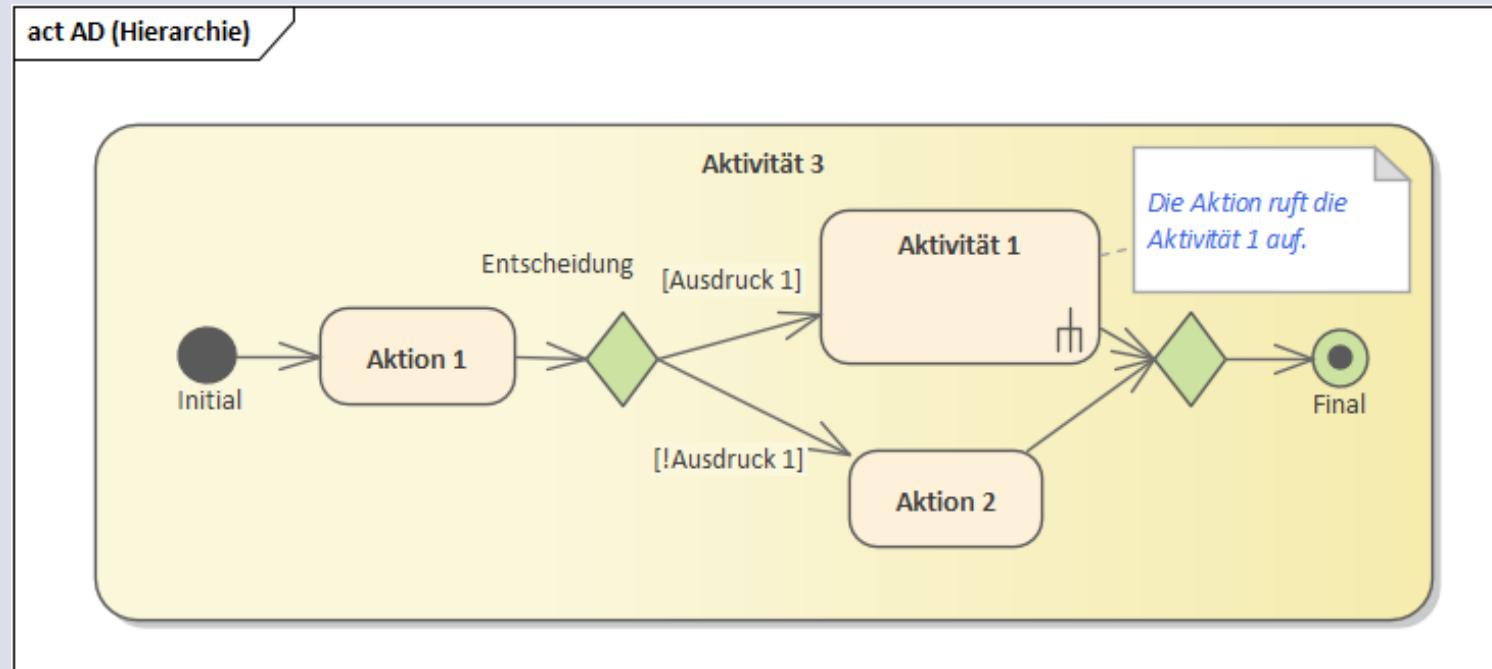
Objektorientierte Analyse und Design

Parallelität



Objektorientierte Analyse und Design

Hierarchie



Objektorientierte Analyse und Design

Checkliste Aktivitätsdiagramm



Frage	Hinweise
Gibt es eine initiale Aktion?	Die Startaktion muss eindeutig festgelegt sein; eine finale Aktion kann fehlen.
Sind die Bedingungen an den Verzweigungen eindeutig definiert?	Jedem alternativen Zweig ist eine zu erfüllende logische Bedingung zugeordnet.
Hat jede Aktion (außer Verzweigungen) eine Folgeaktion?	Außer der finalen hat jede Aktion <u>genau eine eindeutige Aktion</u> zu Folge.
Werden alternative Pfade („Oder“-Pfad) wieder zusammengeführt?	Alternative Pfade müssen zusammengeführt werden (Raute-Symbol).
Werden parallele Pfade („Und“-Pfad) wieder zusammengeführt?	Parallele Pfade müssen zusammengeführt werden (Balken-Symbol)
Sind externe Ereignisse berücksichtigt?	Externe Ereignisse (beispielsweise durch Akteur) sind wartende Aktionen.



Objektorientierte Analyse und Design

STATISCHE SICHT - DOMÄNENMODELL

Objektorientierte Analyse und Design

Statische Sicht: Klassen



- Der Anwendungsbereich stellt sich immer noch unstrukturiert dar! Ziel, Anforderungen, A-fälle sind beschrieben.
- Zu untersuchen ist, welche Klassen, d. h. welche Arten von Objekten im Anwendungsbereich vorkommen.

Schritte

- Klassen identifizieren, d. h. finden
- die entdeckten Klassen im UML-Klassendiagramm aufzeichnen
- dabei zunächst nur die Klassennamen angeben, Methoden, Attribute und Beziehungen werden zunächst zurückgestellt.

Objektorientierte Analyse und Design

Hinweise Domänenmodell



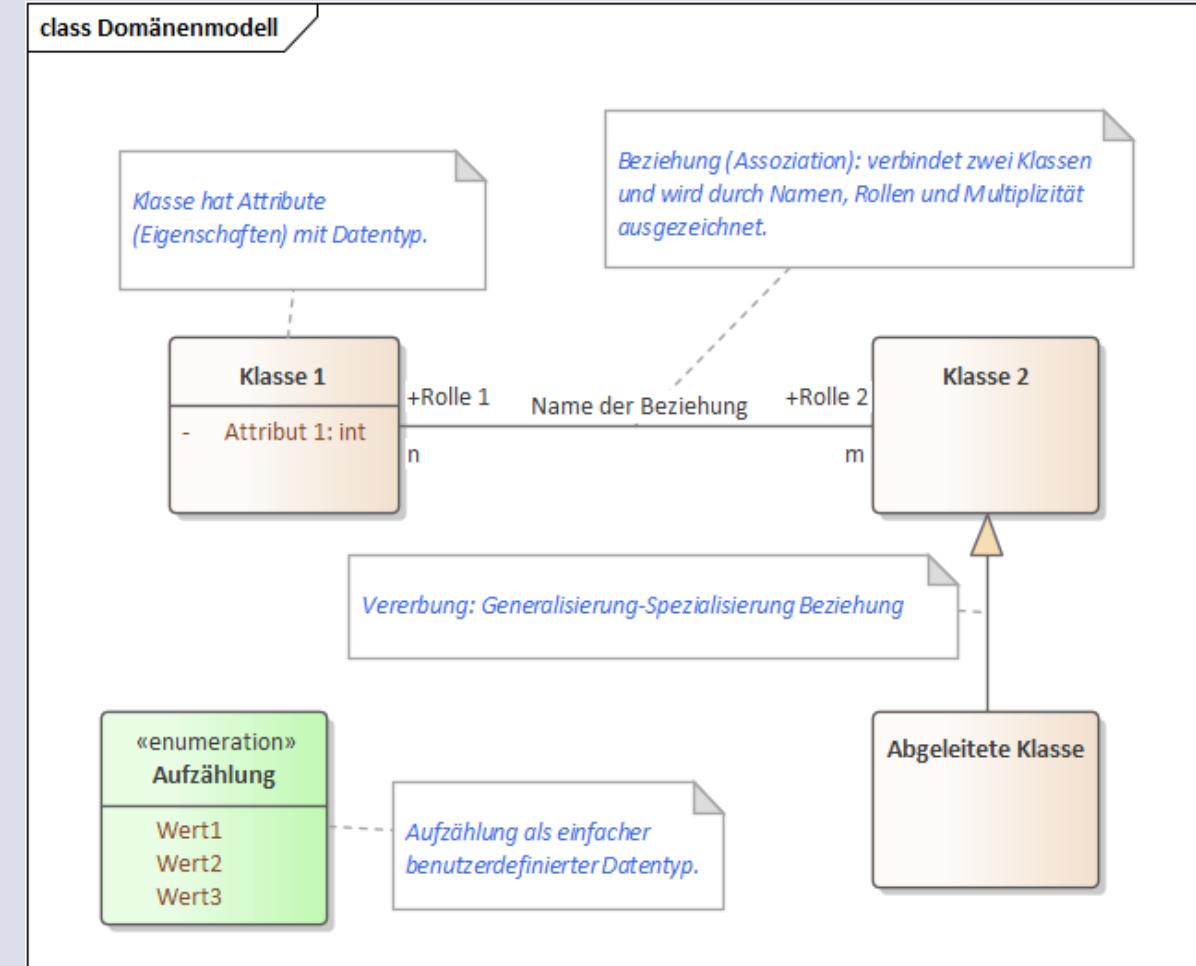
- Ziel: Aufbau eines konzeptionellen, objektorientierten Modells zum Verstehen und Beschreiben des Anwendungsbereichs.
- Vorhanden:
 - Textuelle Beschreibungen
 - Anwendungsfälle, Aktivitätsdiagramme
- Dieses Modell besteht aus:
 - (**Analyse**- bzw. Geschäfts-) **Klassen** ("Konzepte")
 - deren **Beziehungen** untereinander
 - deren **Attribute**
- Man erhält als Analyseergebnis das: **Statische Modell**.
 - Dieses Modell ist nur eine Schätzung. Im Zuge des Gestaltungsschrittes können sich noch zahlreiche Änderungen ergeben.
 - Zusätzliche Aufgabe: Beschreibung des Systemverhaltens, gegeben durch die äußeren Funktionen des Systems.

Objektorientierte Analyse und Design

Klassendiagramm als Begriffsmodell



- Begriffsmodell (**Domänenmodell**, Analysemmodell) als wichtiges Werkzeug, um die Begrifflichkeiten und deren Abhängigkeiten in der Fachdomäne darzustellen.
- Als Darstellung kann das UML-**Klassenmodell** verwendet werden (betrachte nur Ausschnitt).
- Wichtigste Element sind hierbei **Klassen mit Attributen (keine Methoden!)** und **Beziehungen**.



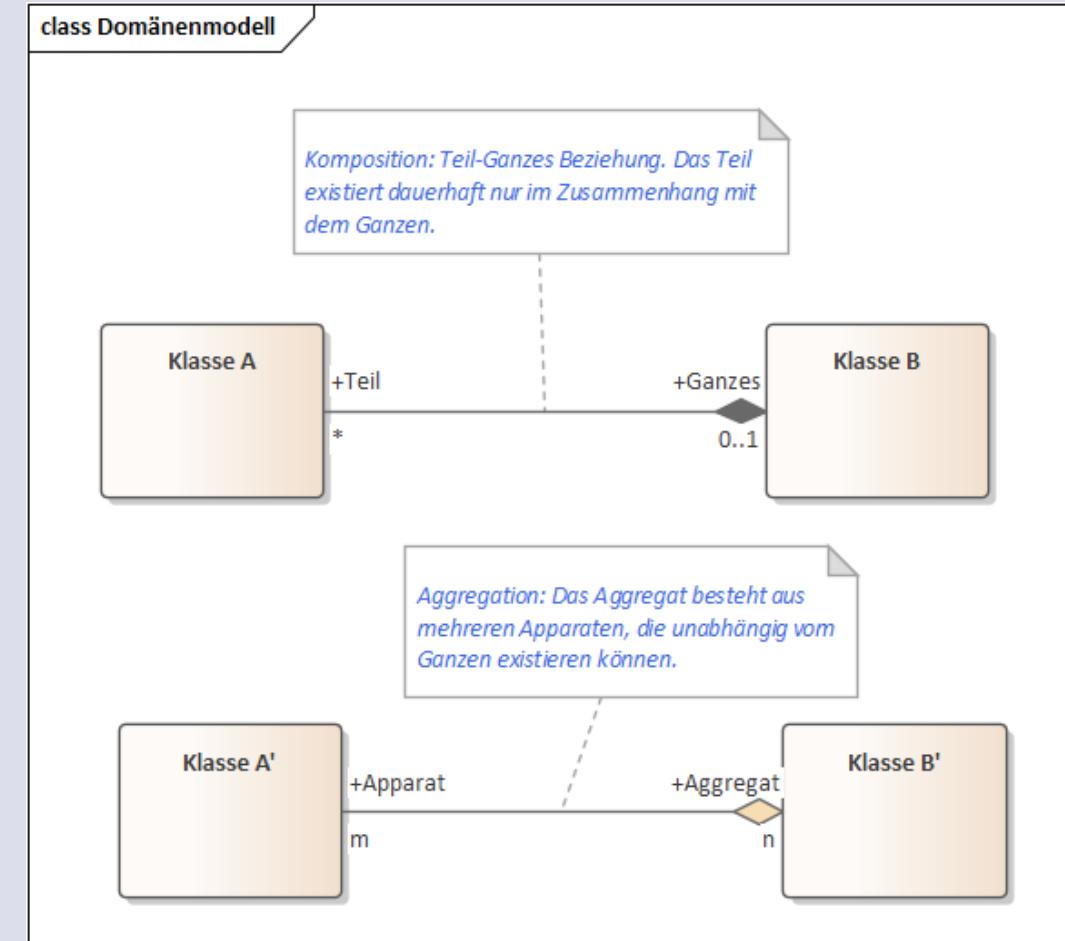
Objektorientierte Analyse und Design

Klassendiagramm in UML



Beziehungen lassen sich unterscheiden:

- Vererbung (geschlossene Spitze)
- Komposition (ausgefüllte Raute)
- Aggregation (offene Raute)



Objektorientierte Analyse und Design

Checkliste Klassendiagramm



Frage	Hinweise
Ist die Klassendefinition vollständig?	Die Klassendefinition umfasst den Namen, die Attribute und die Methoden.
Ist die Vererbungshierarchie vollständig dargestellt?	Eine abgeleitete Klasse erbt die Attribute und Methoden der Basisklasse.
Ist die Navigierbarkeit zwischen den Klassen definiert?	Die Beziehung definiert, dass ein Objekt Methoden eines anderen aufruft.
Sind Aggregation und Komposition von Klassen identifiziert?	Die Komposition zwischen zwei Objekten beschreibt die gleiche Lebensdauer.
Ist die Schnittstelle (öffentliche Methoden) einer Klasse hinreichend?	Durch Vergleich mit dem Sequenzdiagramm wird deutlich, ob alle Aufrufe den Methoden der Klasse korrekt sind.



Objektorientierte Analyse und Design

KLASSEN FINDEN

Objektorientierte Analyse und Design

Finden der Klassen



— Zum Finden der Klassen gibt es drei gängige Methoden:

1. Verwendung einer Begriffsliste
2. Hauptwortmethode
3. „Schwimmbahn“-Methode
4. [Recycling]

Objektorientierte Analyse und Design

Methode 1: Begriffsliste



- Durchgang durch die Anwendung anhand einer **Begriffsliste**.

Objektorientierte Analyse und Design

Beispiel für eine Begriffsliste



Begriff	Beispiel
physische, konkrete Dinge	Flugzeug, Person
Beschreibungen, Entwürfe oder Spezifizierungen	Produkt, Reisebeschreibung
Orte	Flughafen, Vorlesungssaal, Büraum
Transaktionen	Überweisung, Reservierung, Bestellung
Einzelposten einer Transaktion	Einzelposten in einer Bestellung
von Personen eingenommene Rollen	Student, Kunde, Angestellter
Behälter für bestimmte Dinge	Flugzeug, Lager
Teile in einem Behälter	Passagiere, Einzelteile

Begriff	Beispiel
prozessorgesteuerte oder elektromechanische Systeme	Kreditkartenkontrollsysteem, Luftüberwachungssystem
abstrakte Begriffe	Hunger, Angst
Organisationen	Verkaufsabteilung, Fachbereich
Ereignisse	Unfall, Überfall, Flug, Landung
Prozesse	Verkaufsvorgang, Buchungsvorgang
Regeln und Vorgehensweisen	Verfahren zur Immatrikulation/Exmatrikulation
Kataloge	Warenangebot, Studienangebot
Dokumente in rechtlicher oder finanzieller Hinsicht	Quittung, Arbeitsvertrag
finanztechnische Begriffe	Kreditgrenze, Aktie

Objektorientierte Analyse und Design

Methode 2: Hauptwortmethode



- Man betrachtet die Ablaufbeschreibungen der Anwendungsfälle.
- Die in den Sätzen der Ablaufbeschreibungen vorkommenden **Hauptwörter sind Kandidaten für Klassen.**

X bearbeitet Y.

```
graph TD; X((X)) --> A[mögliche Klassen]; Y((Y)) --> A;
```

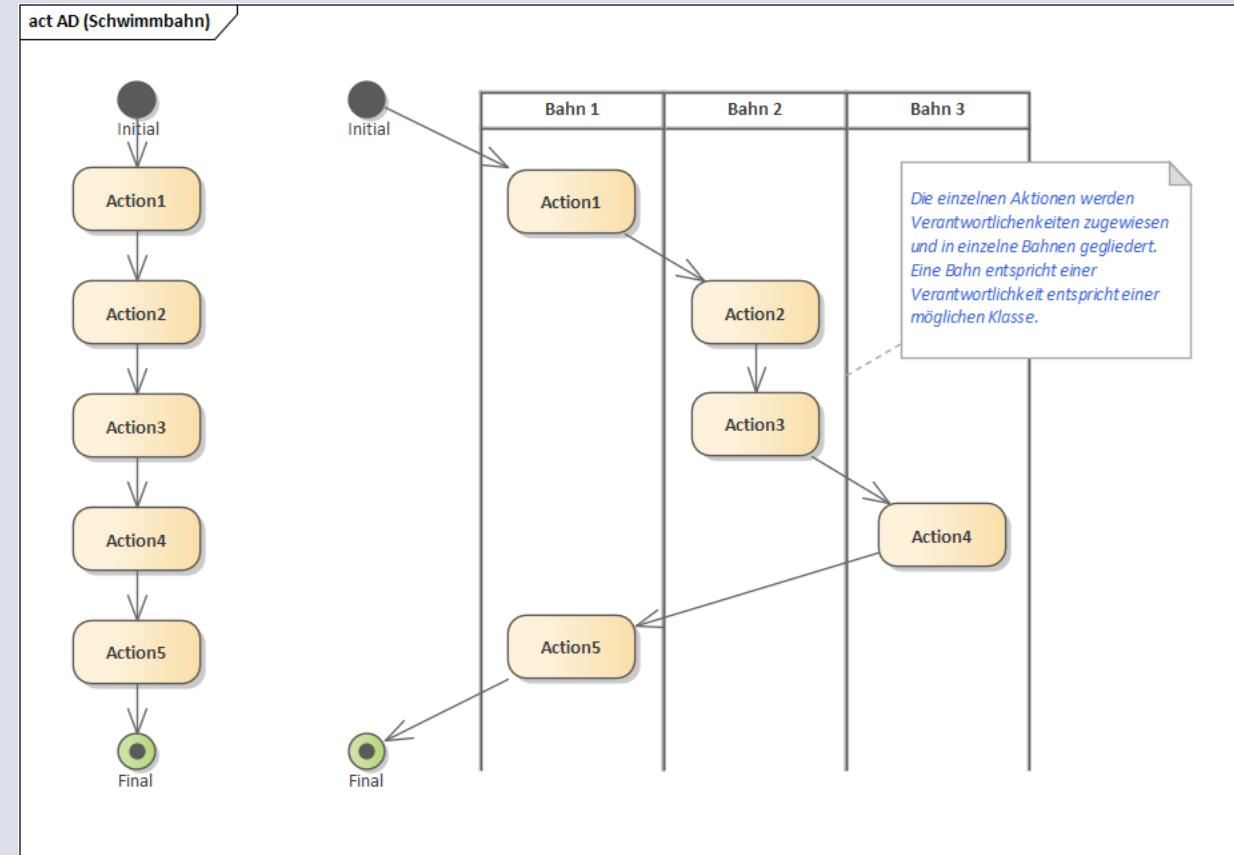
Objektorientierte Analyse und Design

Methode 3: „Schwimmbahn“-Methode



Es handelt sich um eine graphische Variante Methode 2:

- Man geht von einem Aktivitätsdiagramm eines Anwendungsfalles aus.
- Man ordnet die darin vorkommenden Aktionen danach, ob diese denselben Urheber besitzen.
- Aktionen mit demselben Urheber werden in eine Bahn gestellt.
- Die einzelnen Bahnen entsprechen möglichen Klassen.



Objektorientierte Analyse und Design

Methode 4: Recycling



- Verwenden von Klassen eines bereits bestehenden System.
- Insbesondere Verwendung (falls vorhanden)
 - einer Klassenbibliothek
 - eines Rahmenwerkes (framework)

Objektorientierte Analyse und Design

Hinweise zur Klassenfindung



- Die Klassen sollen Dinge oder Vorgänge bezeichnen, die im Anwendungsbereich tatsächlich vorkommen.
 - Keine Programmiergebilde wie „Druckerwarteschlange“, Speicherfeld oder „Fenstersystem“.
- Besser zunächst mehr Klassen als zu wenig! Später vereinfachen.
- Es ist jedoch nicht schlimm, wenn später (bei der Gestaltung oder bei einem folgenden Zyklus) weitere Klassen hinzukommen.
 - Die „Landkartenregeln“:
 - Klassen mit ihren tatsächlichen Namen versehen (d. h. keine Kunstbezeichnungen)
 - Unwichtiges weglassen
 - nichts hinzuerfinden

Objektorientierte Analyse und Design

Finden der Beziehungen



- Zunächst stehen die Klassen noch isoliert nebeneinander!
- Zu untersuchen ist jetzt, welche Beziehungen zwischen den zuvor gefundenen Klassen des Anwendungsbereichs bestehen.
- Zwischen Klassen können Beziehungen der folgenden Arten bestehen:
 - Assoziationen
 - Aggregationen
 - Kompositionen (starke Aggregationen)
 - Vererbungsbeziehungen

Objektorientierte Analyse und Design

Methoden zum Finden der Beziehungen



- Man betrachtet vor allem Beziehungen, die auf **natürliche Weise** im Anwendungsbereich vorkommen.
- Folglich spielen **Vererbungsbeziehungen** zunächst eine **untergeordnete Rolle**. Wichtiger werden sie in der Phase der Gestaltung.
- Man kann zunächst die **Aggregation** und die **Komposition** als einfache Assoziation darstellen.
 - Aggregation und Komposition sind spezielle Assoziationen; die genaue Unterscheidung folgt i. d. R. später.
- Zum Finden der Beziehungen zwischen den Klassen gibt es ähnlich wie bei der Klassenfindung zwei Methoden:
 1. Verwendung einer Begriffsliste
 2. Satzmethode

Objektorientierte Analyse und Design

Methode 1: Beispiel für eine Begriffsliste



Begriff	Beispiel
A ist physikalischer Bestandteil von B.	Flügel -- Flugzeug
A ist logischer Bestandteil von B.	Einelposten in einer Bestellung -- Gesamtbestellung
A ist physikalisch in B enthalten.	Student -- Fachbereich einer Hochschule
A ist logisch in B enthalten .	Vorlesung -- Regelstudienplan einer Semestergruppe
A ist eine Beschreibung für B.	Inhaltsübersicht -- Vorlesung, Reisebeschreibung -- Reise
A ist ein Einelposten der Transaktion oder des Berichts B.	Klausuraufgabe -- Klausur
A wird in B erfasst oder protokolliert.	Anwesende -- Anwesenheitsliste

Begriff	Beispiel
A ist ein Mitglied von B.	Pilot -- Fluggesellschaft
A ist eine Untereinheit von B.	Fachbereich -- Hochschule
A verwendet oder verwaltet bzw. leitet B.	Dekan – Fachbereich, Direktor -- Unternehmensbereich
A kommuniziert mit B.	Kunde -- Verkäufer
A ist mit der Transaktion B verbunden.	Fahrgast -- Fahrt
A ist eine Transaktion und ist mit der weiteren Transaktion B verbunden.	Reservierung -- Stornierung
A ist zu B benachbart oder folgt auf B.	Stadt -- Stadt
A hat B als Eigentümer.	Flugzeug -- Fluggesellschaft

Objektorientierte Analyse und Design

Methode 2: Satzmethode



- Man betrachtet die Ablaufbeschreibungen der Anwendungsfälle.
- Sind in einem **Satz** zwei Klassen (beispielsweise als **Subjekt** oder als **Objekt**) erwähnt, so besteht zwischen ihnen möglicherweise eine Beziehung.

X bearbeitet Y.

X —————→ **Y**

Objektorientierte Analyse und Design

Weitere Regeln für Beziehungen 1



- Gefundene Assoziationen mit **Multiplizitäten** ggfs. mit **Namen** und **Rollen** in das vorhandene Klassendiagramm einzeichnen.
- Auf Übersichtlichkeit achten; nur Assoziationen angeben, die notwendig für das **Verständnis des Begriffsmodells** sind:
 - A ist enthalten in B
 - A ist Bestandteil von B
 - A verwaltet/verwendet/protokolliert B
- Assoziationen, die offenkundig Aggregationen oder Kompositionen sind, als solche kennzeichnen.



Objektorientierte Analyse und Design

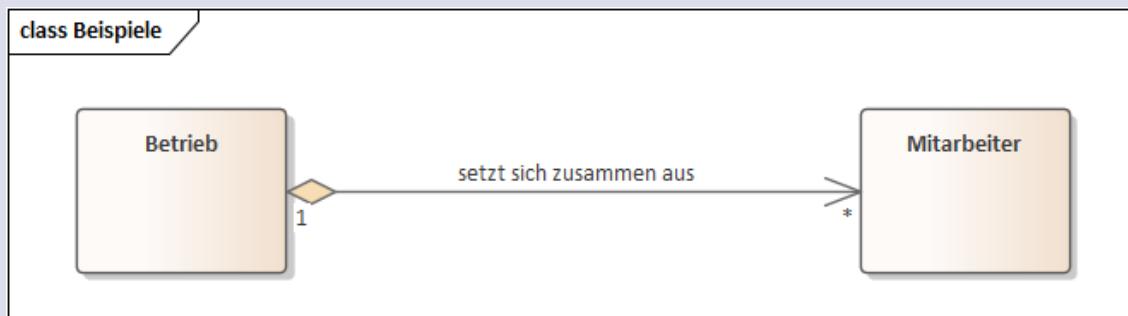
Weitere Regeln für Beziehungen 2



Im Zweifelsfall, ob eine **Aggregation** oder eine einfache **Assoziation** vorliegt: die Beziehung nur als Assoziation ansehen.

Beide Lösungen sind sinnvoll:

- Ein Betrieb setzt sich aus seinen Mitarbeitern zusammen.
- Ein Betrieb ist mit seinen Mitarbeitern durch ein Beschäftigungsverhältnis verbunden.

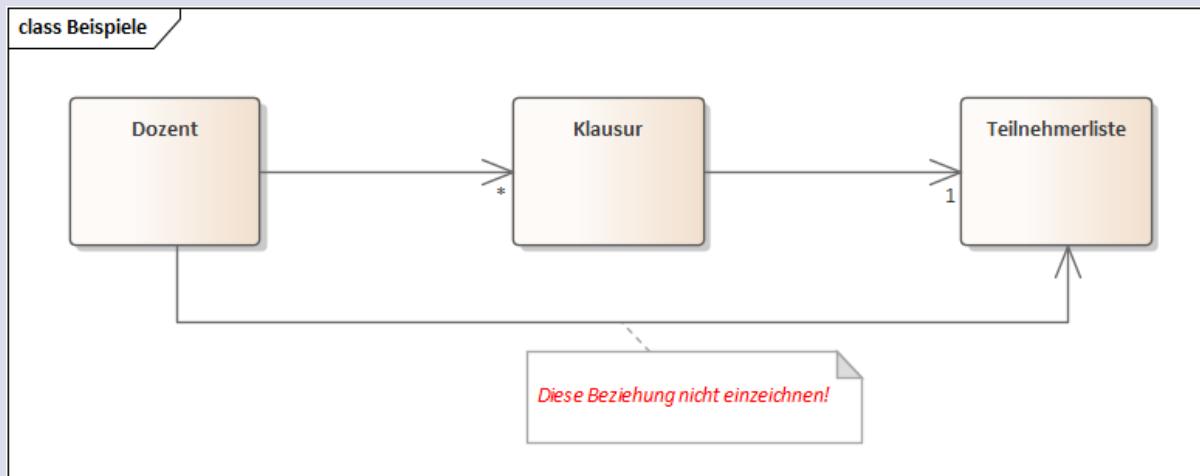


Objektorientierte Analyse und Design

Weitere Regeln für Beziehungen 3



— **Abgeleitete** Assoziationen nach Möglichkeit weglassen.



— Assoziationen auch dann angeben, wenn sie von **begrenzter Dauer** sind.

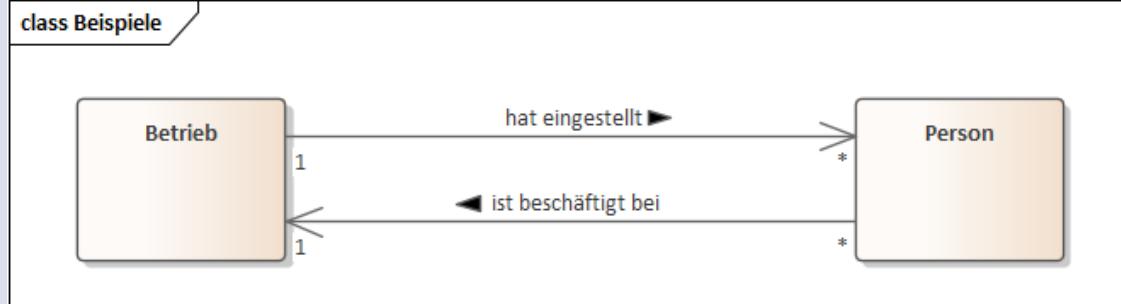


Objektorientierte Analyse und Design

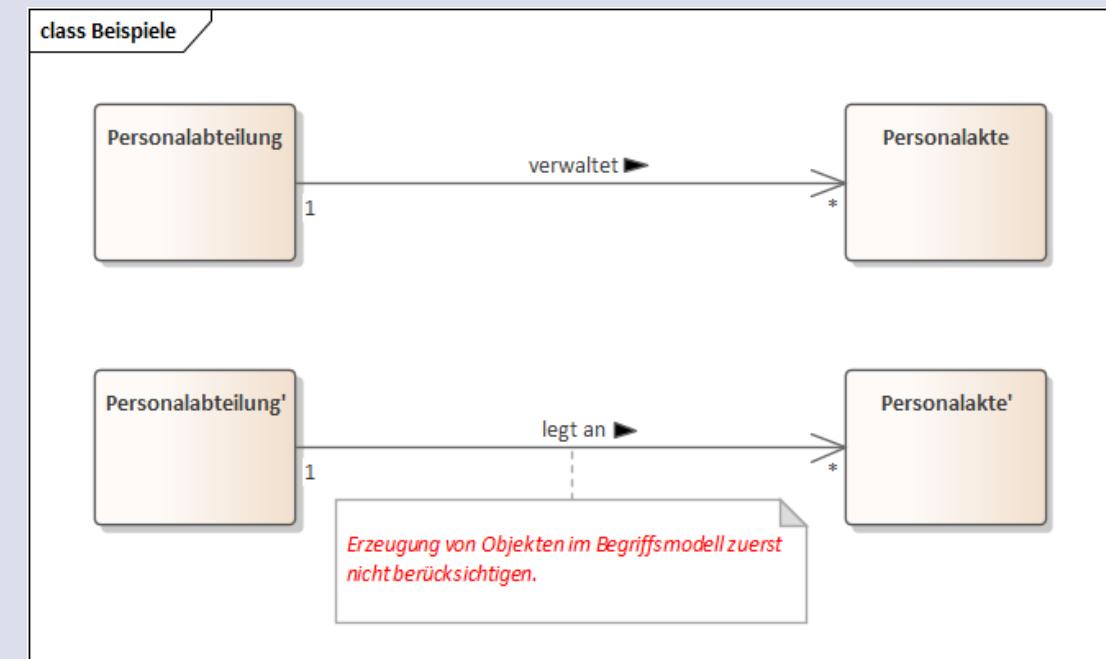
Weitere Regeln für Beziehungen 4



Assoziationen **entgegengesetzter Richtung** (bidirektional) dürfen erscheinen, Beispiel (oder auch Dozent \leftrightarrow Klausur):



Beziehungen mit **Objekterzeugung** werden noch nicht berücksichtigt.

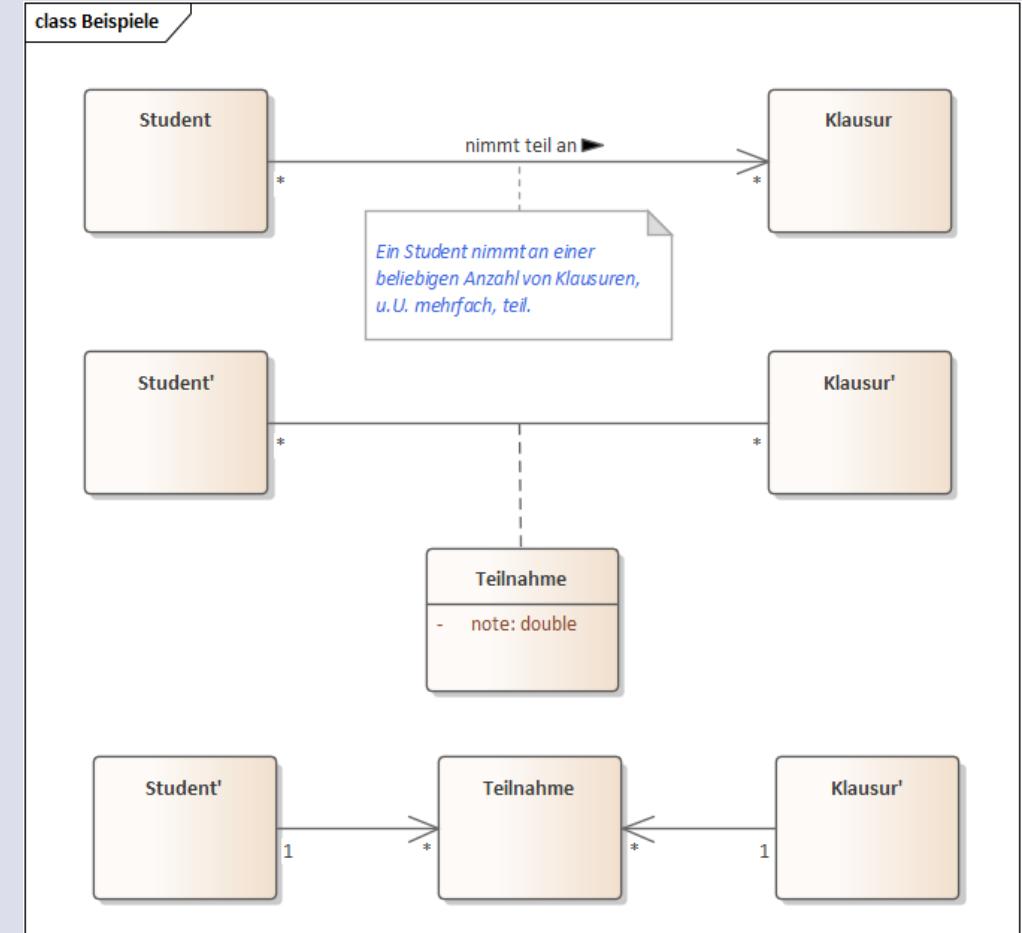
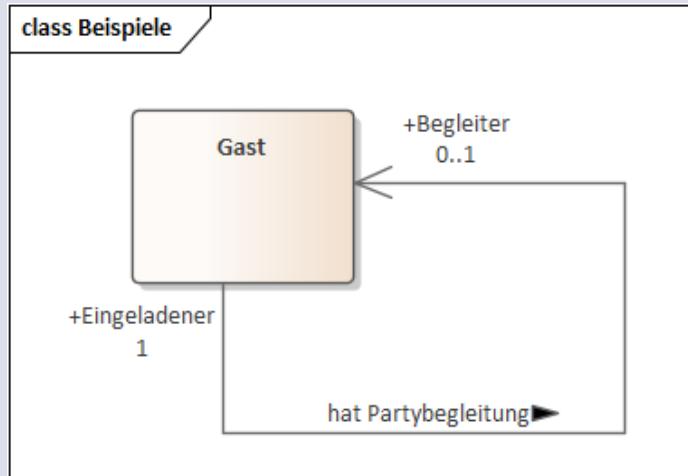


Objektorientierte Analyse und Design

Weitere Regeln für Beziehungen 5



- **Reflexive** Assoziationen sind möglich.
- *-* -Beziehungen können später über Assoziationsklassen aufgelöst (**attributierte Beziehungen**) werden.

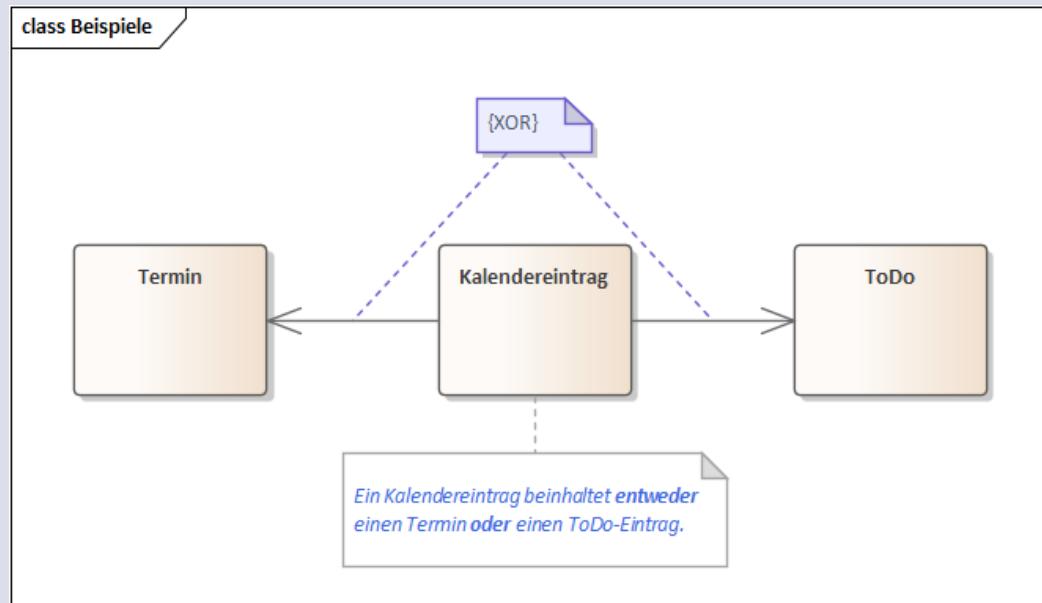


Objektorientierte Analyse und Design

Weitere Regeln für Beziehungen 6



— **XOR-Assoziationen** über Einschränkungen zu definieren.



— Ganz selten kommen **n-äre Beziehungen** vor. Auch diese können i.d.R. über zusätzliche Klassen aufgelöst werden.

Objektorientierte Analyse und Design

Empfehlungen für große Diagramme



— eine Klasse auswählen, die

- viele Beziehungen besitzt oder
- von zentraler Bedeutung bei einem Anwendungsfall ist
- diese Klasse in ein eigenes Unterdiagramm/Layer (desselben Paketes) einfügen
- durch Beziehungen verbundene Klassen dort ebenfalls einfügen
- Klassen und Beziehungen (und Attribute) dort bearbeiten

Objektorientierte Analyse und Design

Klassen mit Attributen versehen



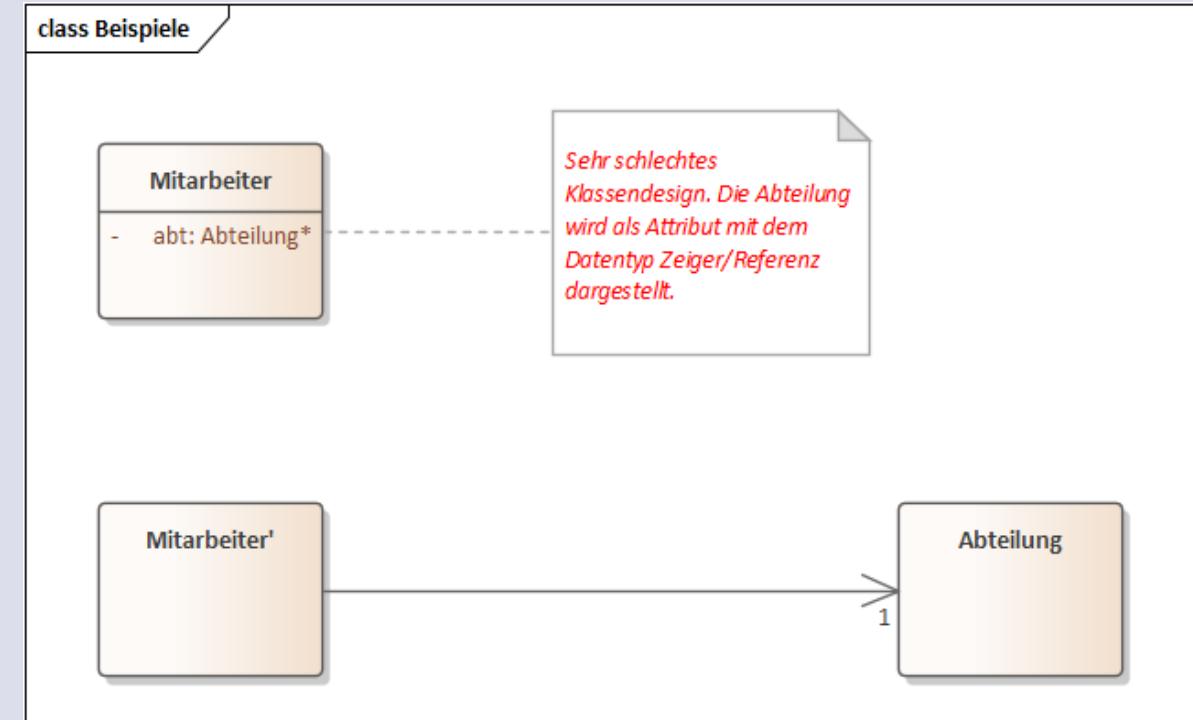
- Ausgangspunkt: Klassen mit Beziehungen.
- Ziel: Analyseklassen um diejenigen Datenwerte ergänzen, die **zum Verständnis** des Analysemodells erforderlich sind und diese ins UML-Klassendiagramm eintragen.
- Welche **Datentypen** sollten bei Attributen vorkommen?
 - Einfache Datentypen aus C++ oder Java bzw. aus der Standardbibliothek: int, double, string usw.
 - „einfache“ Hilfsklassen: Datum, Farbe (RGB), Punkt (x,y,z) usw.
- Merkmal eines einfachen Datentyps: Ihre Werte stellen **keine größeren Individuen** dar!

Objektorientierte Analyse und Design

Fehlerquelle: Beziehung \neq Attribut !



So nicht: Beziehungen als Attribute der Klasse darstellen.





Objektorientierte Analyse und Design

ANFORDERUNGEN VERFEINERN

Objektorientierte Analyse und Design

Anforderungen verfeinern



- Anforderungsanalyse ist **iterativer Prozess**: Verfeinerung (Konkretisierung, Detaillierung) von grob nach fein.

Objektorientierte Analyse und Design

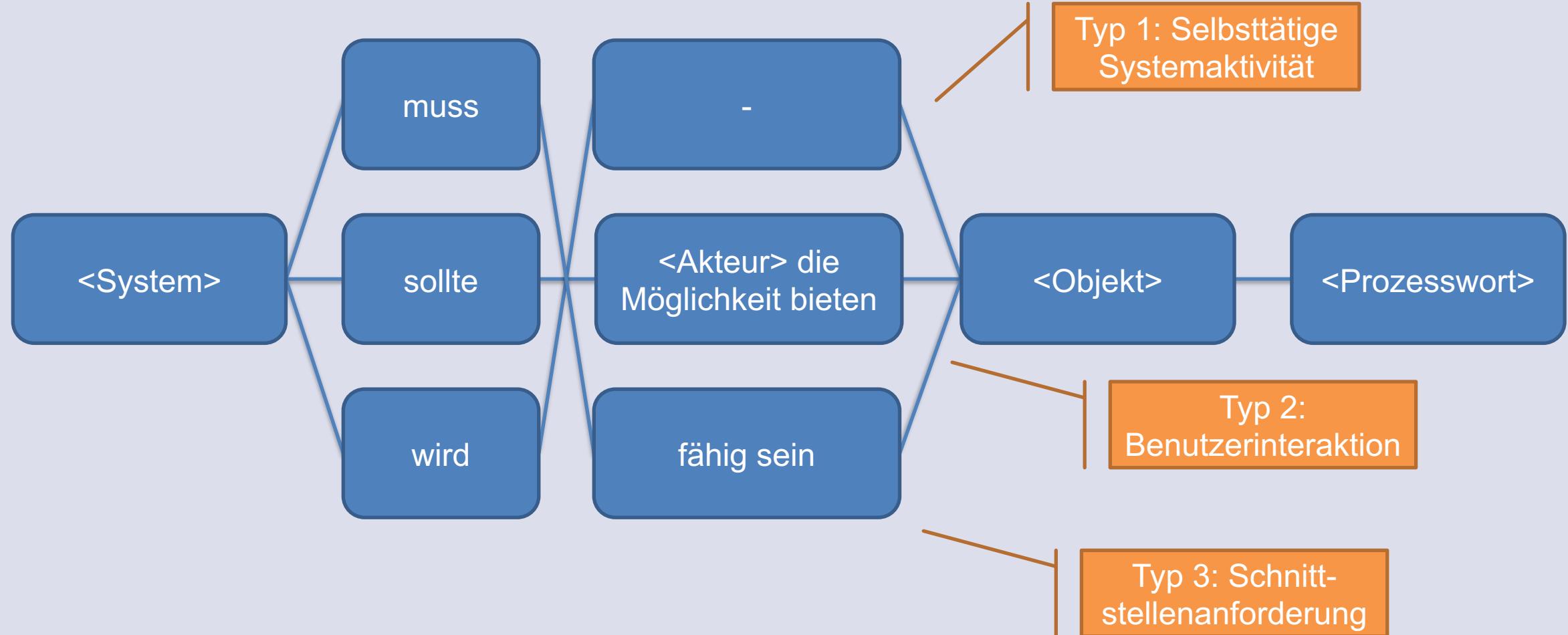
MASTeR-Plan



- Möglicher Weg zu qualitativ hochwertigen Anforderungen: erste Version erstellen und dann schrittweise verbessern.
- Alternative: von Anfang an hochwertige Anforderungen zu schreiben.
 - Dieser Ansatz kann durch Schablonen unterstützt werden, die den Satzbau von Anforderungen vorgeben.
- Es ist zu beachten, dass die erwähnten Ausdrucksprobleme auch in diesem Ansatz noch bestehen.
- Ansatz nach Rupp.

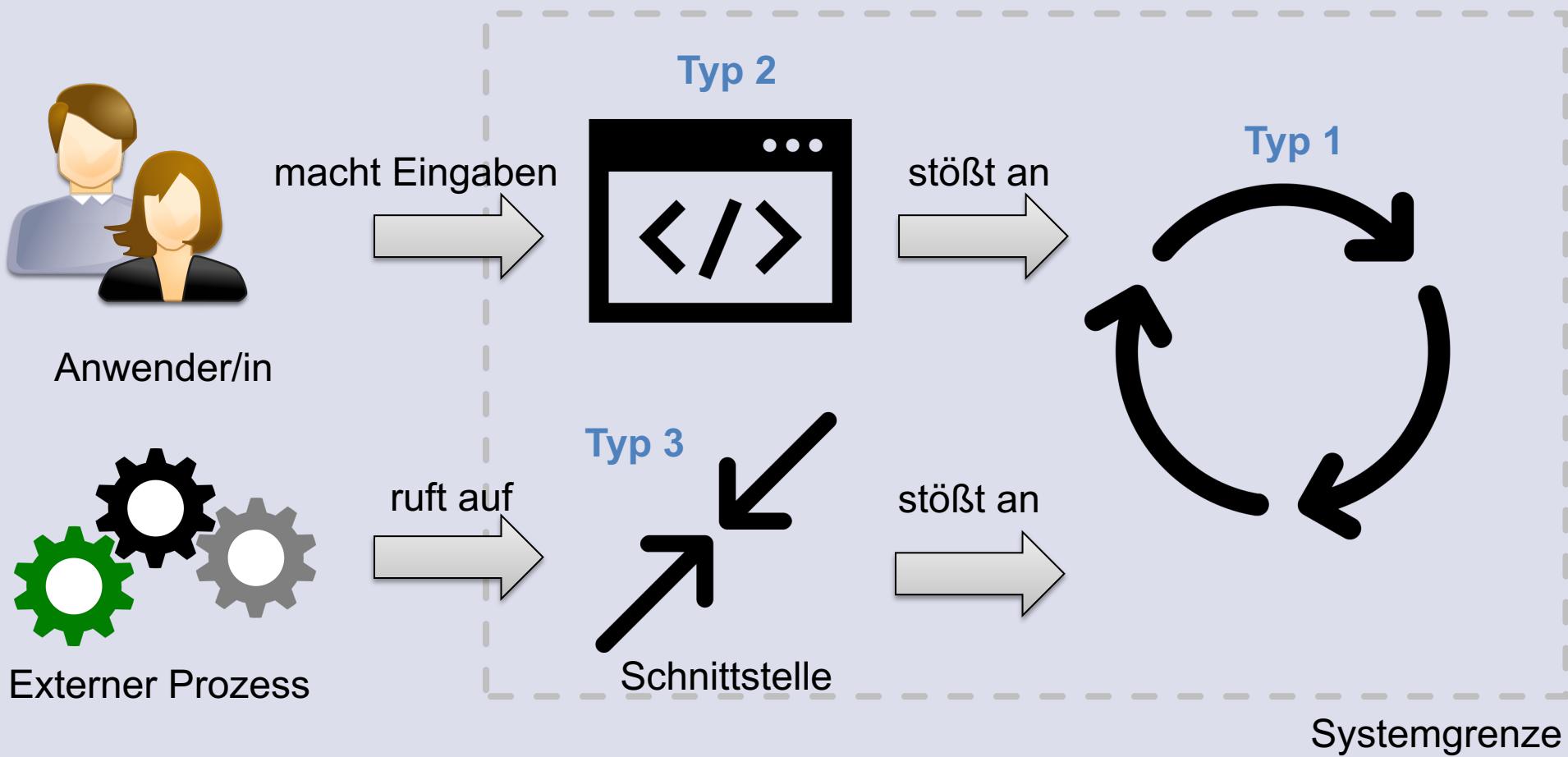
Objektorientierte Analyse und Design

FunktionsMASTeR ohne Bedingung



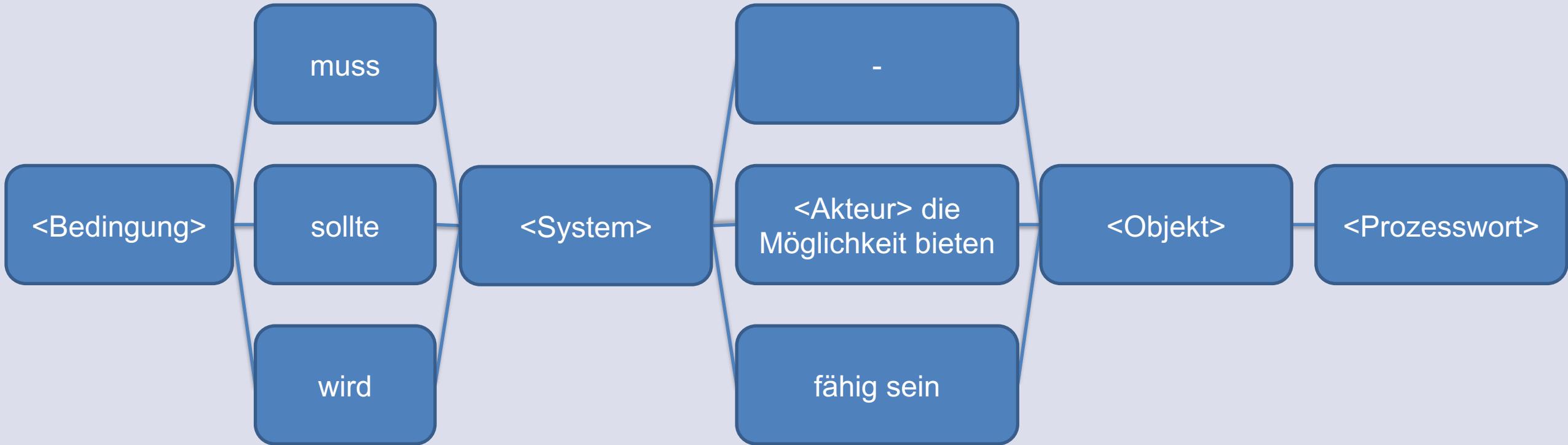
Objektorientierte Analyse und Design

Systemaktivitäten Typ 1 - 3



Objektorientierte Analyse und Design

FunktionsMASTeR mit Bedingung





Objektorientierte Analyse und Design

NICHT-FUNKTIONALE ANFORDERUNGEN (NFA)

Objektorientierte Analyse und Design

Nicht-funktionale Anforderungen



- Bisher im Fokus:
Funktionalität.
- Es gibt **weitere Faktoren**, die für Projekterfolg wichtig und entscheidend sind.
- Keine einheitliche Definition;
Vielzahl unterschiedlicher Aspekte und Randbedingungen.

Objektorientierte Analyse und Design

Nicht-funktionale Anforderungen



- Standards
- Werkzeuge
- Durchführung von Besprechungen

Benutzungs-oberfläche

- Form und Funktion von Ein- und Ausgabe-Geräten
- Ergonomie etc.

Durchzuführende Tätigkeiten

Sonstige Lieferbestandteile

- Zahlungsmeilensteine
- Vertragsstrafen
- Umgang mit Änderungen
- Eskalationspfade

Technologische Anforderungen

- Hardwareanforderungen
- Architekturanforderungen
- Anforderungen an die Programmiersprache

Rechtlich-vertraglich

- Systemhandbücher
- Installationshandbücher

Qualitäts-anforderungen

- Qualität des Systems
- Entwicklungsprozess

Objektorientierte Analyse und Design

NFA für technische Systeme



- Bei technischen Systemen sind häufig Zeitanforderungen einzuhalten: **harte** und **weiche Echtzeitbedingungen**.
- Beispiel für Safety-Anwendung: Reaktionszeit beim Airbag-Steuergerät beträgt 1 ms.
- Anforderungen können textuell und/oder grafisch erfolgen.
- UML bietet hierzu aus der Elektrotechnik bekannte **Timing-Diagramm** an.
 - *(kommt später)*



Objektorientierte Analyse und Design

PRÜFEN UND BEWERTEN

Objektorientierte Analyse und Design

Qualitätssicherung



- Was ist **Qualität**?
- [ISO]: „Qualität ist der Grad, in dem ein Satz inhärenter Merkmale Anforderungen erfüllt.“
- Analyse: Maß, in dem eine Anforderung den objektiv messbaren **Qualitätskriterien** entspricht.
- Ziel: Sicherstellung, dass die Anforderungen dem gewünschten System der Stakeholder entspricht.
- Qualitätssicherung wird im Entwicklungsprozess am Ende der unterschiedlichen Phasen eingeplant.
- Unterschiedliche Methoden.



Objektorientierte Analyse und Design

Prüftechnik: Review



Stellungnahme:

Anforderungsdokument werden Kollegen/in überstellt; diese/r markiert Auffälligkeiten.

- Pro: Einfach durchzuführen, 4 Augen sehen mehr als 2.
- Contra: Ohne Prüfziele, geringere Qualität, abhängig vom Kollegen/in

Walkthrough:

Präsentation der Anforderungen und gemeinsame Durchsprache.

- Pro: Gemeinsames Verständnis; Rückfragen möglich
- Contra: Manipulation durch Moderator möglich, “Wilde Diskussionen”.

Inspektion:

Formales Vorgehen; Stellungnahmen werden gemeinsam diskutiert und bewertet.

- Pro: Hohe Qualität der Ergebnisse.
- Contra: Hoher Zeitaufwand.

Objektorientierte Analyse und Design

Prüftechnik: Prototyp/Simulationsmodell



- Anforderungen werden (teilweise) umgesetzt, um Realisierbarkeit zu überprüfen.
- Häufig wird **Mensch-Maschine-Schnittstelle** betrachtet.
 - Pro: Gute Diskussionsbasis zwischen Stakeholder, Anwendern und Entwicklung.
 - Contra: Hoher Aufwand für “Wegwerf”-Anwendung; falscher Eindruck kann entstehen: „System ist schon fast fertig“.
- Prototypen lassen sich zur **Absicherung** kritischer Anforderungen (z.B. Echtzeitanforderungen) einsetzen, um das Entwicklungsrisiko zu minimieren.

Objektorientierte Analyse und Design

Prüftechnik: Testfälle



- “**Testfall** ist eine Anweisung für den Test bezüglich einer Anforderung, welche die Erfüllung der Anforderung für das zu erstellende System beschreibt.”
 - Pro: Testfälle beliebig einsetzbar, Auffälligkeiten lassen sich gut identifizieren.
 - Contra: Zeitaufwendig; nicht alle Qualitätskriterien lassen sich überprüfen.

- Aufbau:
 - Ausgangssituation: Vorbedingungen
 - Testereignis: auslösendes Ereignis für den Testfall
 - Erwartete Ergebnis
- Wichtig! **Testfallautor/in ≠ Anforderungsautor/in**
 - Natürlichsprachlicher Testfall.
 - Synonym: Abnahmekriterium, Akzeptanzkriterium.

Objektorientierte Analyse und Design

Qualitätsmetriken



- Qualität der Anforderungen bestimmt das Projektrisiko:
 - Hohe Qualität → geringes Risiko.
 - Geringe Qualität → hohes Risiko.
- Qualität muss daher messbar sein: Metrik gibt objektive Aussage.
- Betrachte Strichprobe der Anforderungen.

- Messleitfaden
 - Metriken, die verwendet werden,
 - Berechnungsformel für die Metriken,
 - Beschreibung der Prüfmethode für eine Metrik,
 - Beispiele, um die Prüfmethode zu verdeutlichen.
- Beispiel Metrik: Anzahl der Nominalisierungen.

Objektorientierte Analyse und Design

Konsolidierung



- Anforderungen lassen sich häufig nicht widerspruchsfrei formulieren → **Konflikt**.
- Konflikt zwischen Stakeholder:
 - Sachkonflikt
 - Benennungskonflikt
 - Interessenkonflikt
 - Wertekonflikt
 - Beziehungskonflikt
 - ...

- **Konsolidierung** (Konfliktauflösung)
 - Annäherungsmethoden: Gemeinsame Kommunikation, um Konflikt aufzulösen.
 - Abstimmungs- und Weisungsmethoden: Abgabe der Entscheidung an andere Person(en).
- Ergebnis der Konsolidierung ist zu dokumentieren.



Objektorientierte Analyse und Design

VERWALTEN

Objektorientierte Analyse und Design

Requirements-Management...



- ... umfasst Maßnahmen, welche die Anforderungsanalyse und die weitere **Verwendung der Anforderungen** unterstützen.
- Beobachtung: Anforderungen verändern sich während der Entwicklung. Erforderlich:
 - Objekt-ID
 - Zustand einer Anforderung
 - Versionierung
- Zur Verwaltung von Anforderungen werden häufig in der beruflichen Praxis Büroprogramme (Word+Excel) eingesetzt.
- Besser: **Spezialisierte Werkzeuge** (z. B. Doors, IBM).
- **ReqIF** als standardisiertes Austauschformat für Anforderungen (XML-basiert).

Objektorientierte Analyse und Design

Änderungsmanagement



- Ursachen für Änderungen von Anforderungen:
 - **Bug, Defekt, Innovation oder Tuning**
 - Wunsch von Auftraggeber resp. Auftragnehmer.
- Änderung der Anforderung (**Change Request**) durch Auftraggeber: Möglichkeit für Auftragnehmer zusätzliche Kosten, Terminverzug etc.
- Um die Auswirkungen der Änderungen zu bewerten, muss es möglich sein, jede einzelnen Anforderung über alle Entwicklungsphasen nachzuverfolgen (**Traceability**).



Objektorientierte Analyse und Design

SONSTIGES ZUM ANALYSESCHRITT

Objektorientierte Analyse und Design

Alternative Ansätze und Erweiterungen



- Anforderungsanalyse (engl. Requirements Engineering, RE) ist **umfangreicher und komplexer Vorgang**.
- Hier nur eine Einführung, es existieren eine Reihe weiterer Ansätze (z. B. **zielorientierte Analyse**: GORE).
- **Firmenspezifische** und **branchenabhängige** Ausprägung.
- **Produkthaftung** (Vertrag, Gericht): Nachweis Handlung entsprechend **Stand der Technik**.
- **Standards und Normen**, z. B. Software Requirements Specification (SRS) nach ISO-29148-2018 (ursprünglich IEEE-830).

Objektorientierte Analyse und Design

Lasten- und Pflichtenheft



Lastenheft

- Lastenheft wird vom Auftraggeber (Kunden) geschrieben
- welche Funktionalität ist gewünscht
- welche Randbedingungen (SW/HW) gibt es

Pflichtenheft

- Pflichtenheft wird vom Auftragnehmer (Software-Entwicklung) geschrieben
- welche Funktionalität wird realisiert
- auf welcher Hardware läuft das System
- welche SW-Schnittstellen (Versionen) berücksichtigt



Objektorientierte Analyse und Design

ZUSAMMENFASSUNG

Objektorientierte Analyse und Design

Zusammenfassung



- Die Anforderungsanalyse ist der **wesentliche** Schritt im Entwicklungsprozess, um ein Projekt **erfolgreich** durchzuführen.
- **Qualitativ hochwertige Anforderungen** führen zu einem qualitativ hochwertigen Projektergebnis.
- **Nächste Schritte:**
 - Das „Was“ ist in der Anforderungsanalyse geklärt → das „Wie“ wird im Design (Software-Entwurf) untersucht.
 - Detaillierung durch weitere UML-Diagrammtypen