

Software Engineering

2. Anforderungsspezifikation

Ruth Breu

Übersicht

2.1 Anforderungsspezifikation mit Use Cases

2.2 Pflichten- und Lastenheft

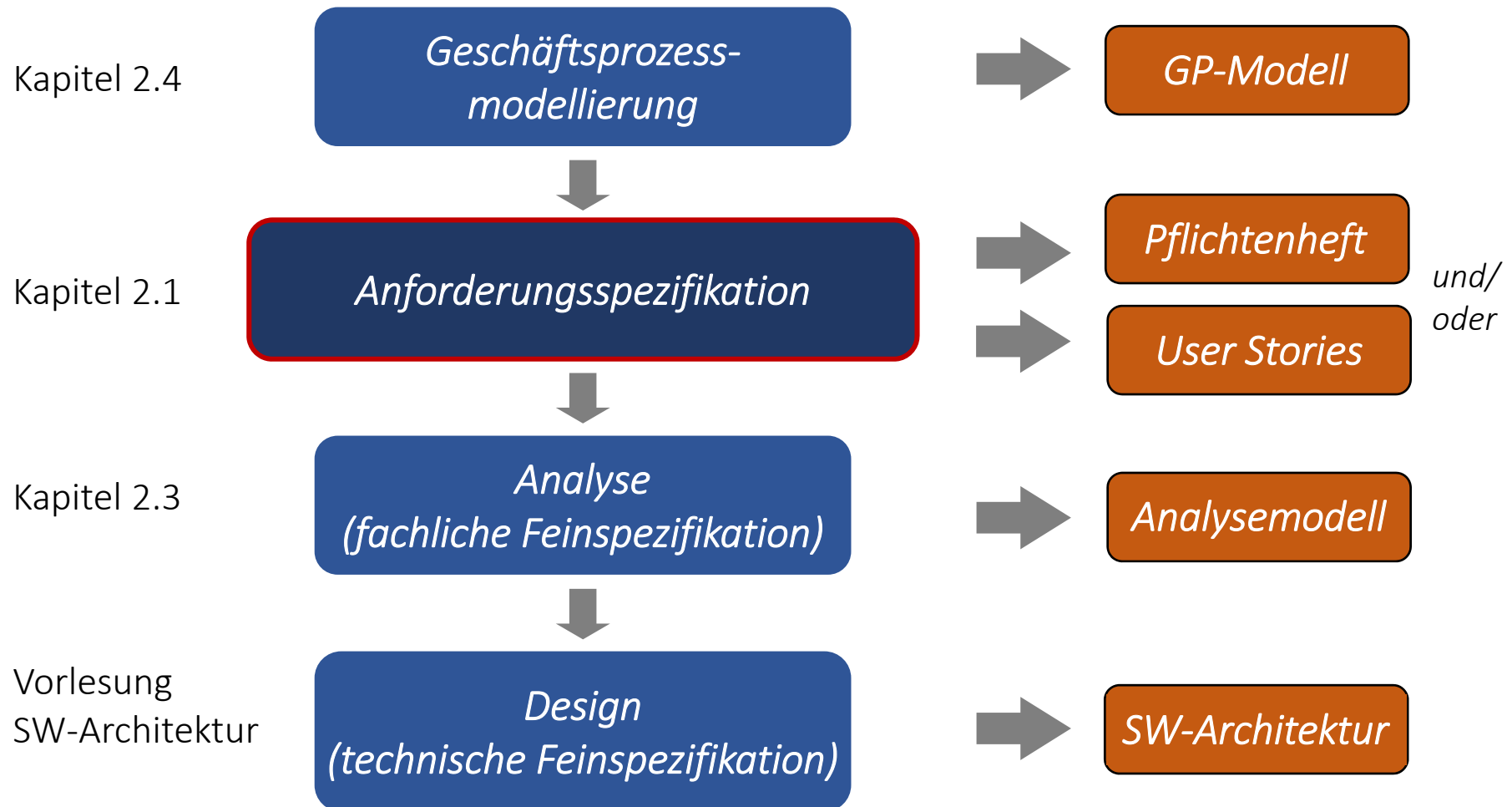
2.3 Analyse

2.4 Geschäftsprozessmodellierung

Worum geht's?

- Erfassen von Anforderungen an ein System
 - Während des Entwurfs (was wünscht sich der Kunde?)
 - Zur Dokumentation (Welche Funktionen sind in Version X implementiert?)
- Vorgestellt wird eine Entwurfsmethodik, die den Entwickler anleitet,
 - in welchen Schritten er vorgehen soll
 - welche Aspekte er modellieren soll
 - welche Beschreibungstechniken er dazu verwenden soll
- Im folgenden: Entwurfsmethodik der Anwendungsfälle (Use Cases)

Einordnung



2.1 Anforderungsanalyse mit Use Cases

- In diesem Teil werden wir vor allem die Entwurfsschritte der Anforderungsspezifikation besprechen
- Die Anforderungsspezifikation wird meist von Entwicklern und Kunden/Anwendern/Fachbereich gemeinsam durchgeführt
- Hauptziel ist die Beschreibung von fachlichen Anforderungen an das System
- Die Ergebnisse der Anforderungsspezifikation werden im **Pflichtenheft** festgehalten

Was ist ein Use Case (Anwendungsfall)?

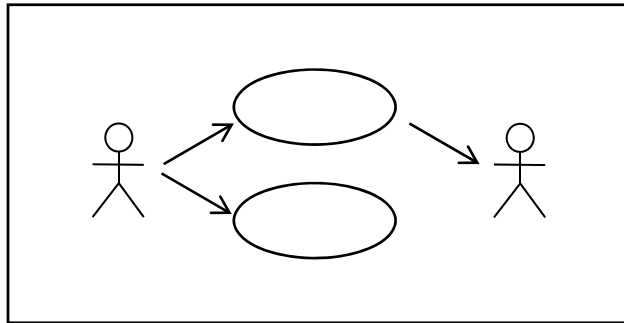
- Ein Anwendungsfall ist ein Dienst, den das zu entwickelnde System nach außen zur Verfügung stellt
 - Motivation: Entwickler (und Kunden) denken in frühen Phasen des Entwurfs nicht nur in „Objekten“ (Kunde, Vertrag, ...), sondern auch in Abläufen, Tätigkeiten, Aufgaben (Bestellung entgegennehmen, Vertrag abschließen, ...)
- Die Identifikation von Anwendungsfällen steuert die weiteren Entwicklungsschritte
 - von groben fachlichen Eigenschaften zu detaillierten fachlichen und technischen Aspekten der Implementierung
 - Grundlage für Aufwandsschätzungen und Projektplanung
 - Grundlage für die Risikoanalyse
 - Grundlage für Tests

Ziele der Anforderungsanalyse

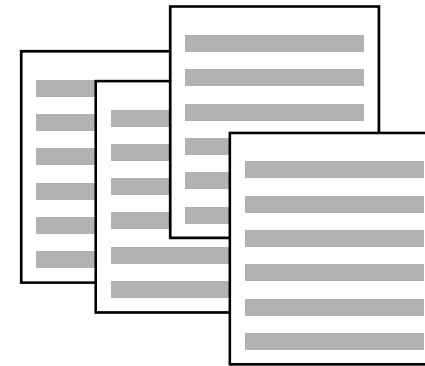
- Gemeinsames Verständnis mit dem Auftraggeber, was das System leisten soll
- Identifikation, wer mit dem System arbeitet, und welche Dienste das System dabei leistet
- Identifikation der Schnittstellen des Systems
- Definition von Systemgrenzen
 - Was leistet das System und was nicht?
 - Mit welchen anderen Systemen kommuniziert das System?

Ergebnisse der Anforderungsanalyse

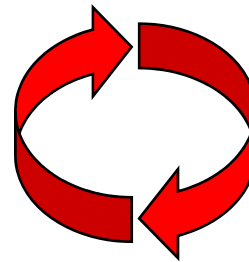
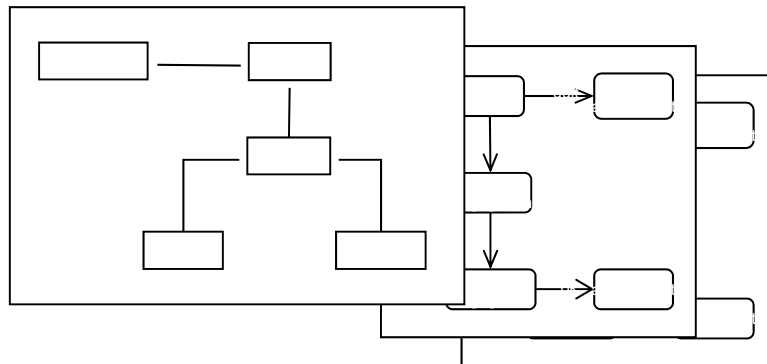
Anwendungsfalldiagramm



Anwendungsfallbeschreibungen

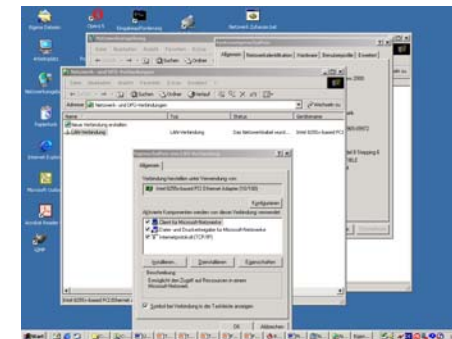


*Klassendiagramm +
Zustandsdiagramme*



Entwicklung verzahnt

Oberflächenprototyp



Übersicht

2.1.1 Anwendungsfalldiagramme

2.1.2 Textuelle Beschreibung von Anwendungsfällen

2.1.3 Modellierung des Anwendungskerns

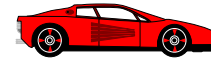
Fallstudie - Die Autovermietung CarRent



Reservierungsstation



Abholstation



Rückgabe

Begriffe

Station

Kalender

Reservierung

Auto

Kunde

Abholung

Autokategorie

Vertrag

Rückgabe

2.1.1 Anwendungsfalldiagramme

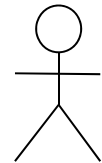
- **Grundkonzepte**

- System

- Mit „System“ bezeichnen wir im folgenden das zu entwickelnde IT-System

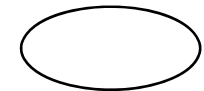
- Akteur

- Ein Akteur repräsentiert alles, was mit dem System interagiert



- Anwendungsfall

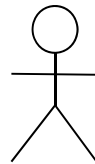
- Ein Anwendungsfall ist eine Abfolge von Aufgaben, die durch das System ausgeführt werden, und die ein Ergebnis von messbarem Wert für einen bestimmten Akteur hervorbringen



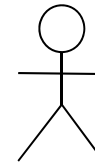
Was ist ein Akteur?

- Akteure stellen Rollen eines Benutzers dar, die dieser gegenüber dem System einnehmen kann
 - Beispiel: Kreditsachbearbeiter, Firmenkundenbetreuer
- Eine Person kann mehrere Rollen spielen
- Ein Akteur kann ein Mensch, eine Maschine oder ein anderes System sein
- Ein Akteur kann Informationen mit dem System austauschen
 - Beispiel: Bankkunde am Bankomaten
- Ein Akteur kann auch nur ein passiver Empfänger von Information sein
 - Beispiel: Fluggast bzgl. der Anzeigetafel der Flüge
- Ein Akteur kann auch nur ein Informationsgeber sein
 - Beispiel: Sensoren

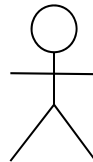
Beispiel 1 - Autovermietung



Angestellter

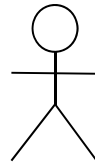


Sachbearbeiter
Zentrale

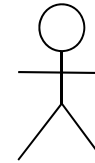


Administrator

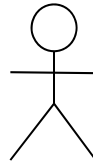
Beispiel 2 - Bankomat



Bankkunde



Bedienungspersonal



Zentrale Konten Kernanwendung

Akteure definieren – Hilfreiche Fragen

- Wer nutzt einen bestimmten Dienst des Systems?
- Wo in der Organisation wird das System eingesetzt?
- Wer beliefert das System mit Informationen, nutzt die Informationen, verändert die Informationen?
- Wer unterstützt und wartet das System?
- Nutzt das System externe Dienste?

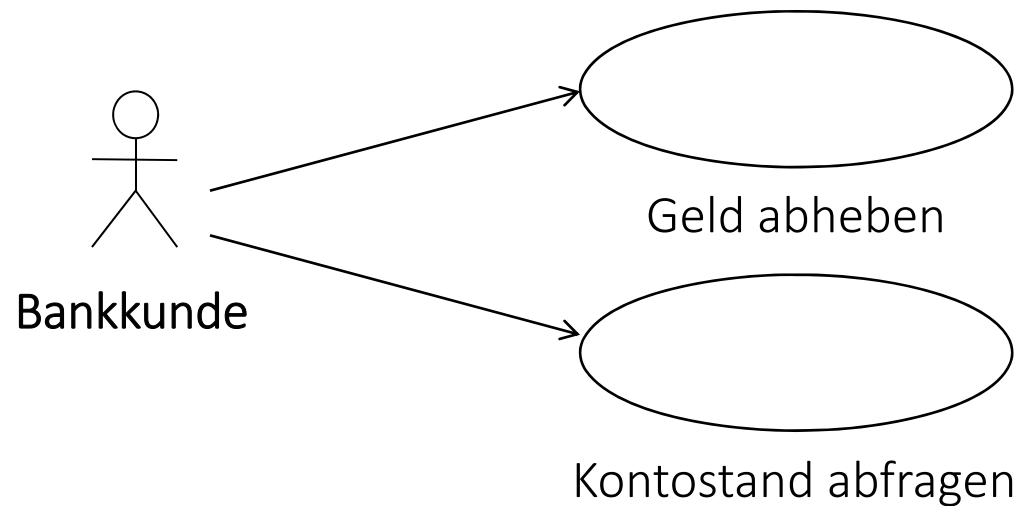
Was ist ein Anwendungsfall?

- Ein Anwendungsfall beschreibt die Interaktion zwischen einem oder mehreren Akteuren und dem System
- Ein Anwendungsfall stellt einen vollständigen und sinnvollen Ablauf von Ereignissen dar
- Ein Anwendungsfall beschreibt das System aus der Sicht der Akteure
- Zusammengekommen bilden die Anwendungsfälle alle möglichen Wege zur Nutzung des Systems ab

Anwendungsfälle definieren – Hilfreiche Fragen

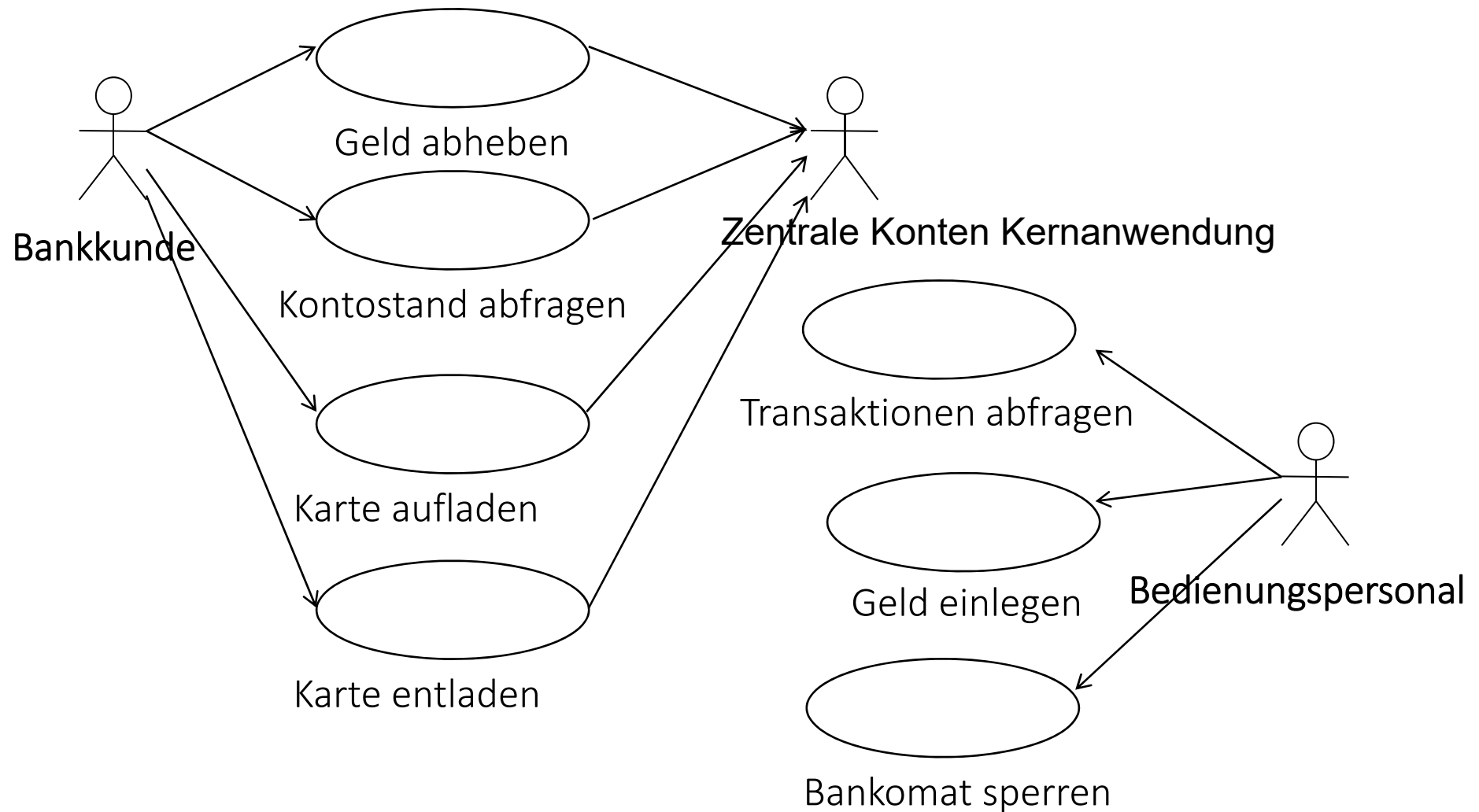
- Was sind die Aufgaben eines Akteurs?
- Will ein Akteur im System Informationen erzeugen, speichern, ändern, entfernen oder lesen?
- Welcher Anwendungsfall soll diese Informationen erzeugen, speichern, ändern, entfernen oder lesen?
- Welche Anwendungsfälle unterstützen bzw. warten das System?

Interaktion mit dem System



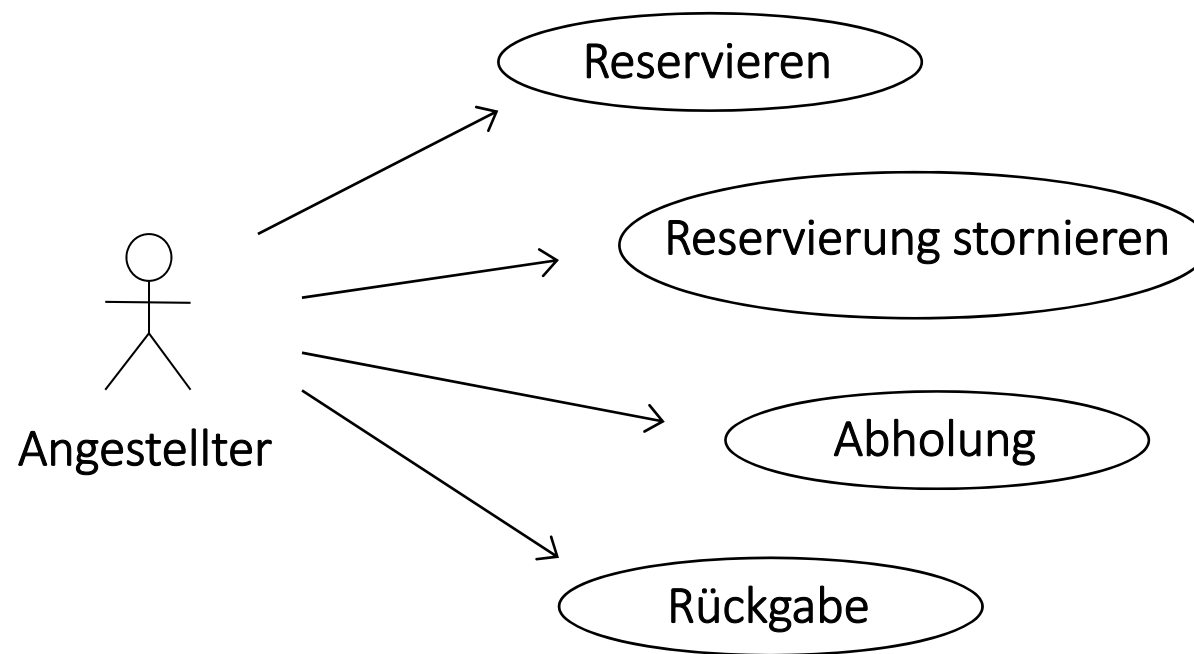
- Eine Kommunikationsverbindung zwischen einem Akteur und einem Anwendungsfall zeigt an, dass diese interagieren
- Die Richtung der Beziehung zeigt an, wer die Kommunikation initiiert (optional)

Beispiel Bankomat Anwendungsfalldiagramm



Beispiel Autovermietung

Anwendungsfalldiagramm



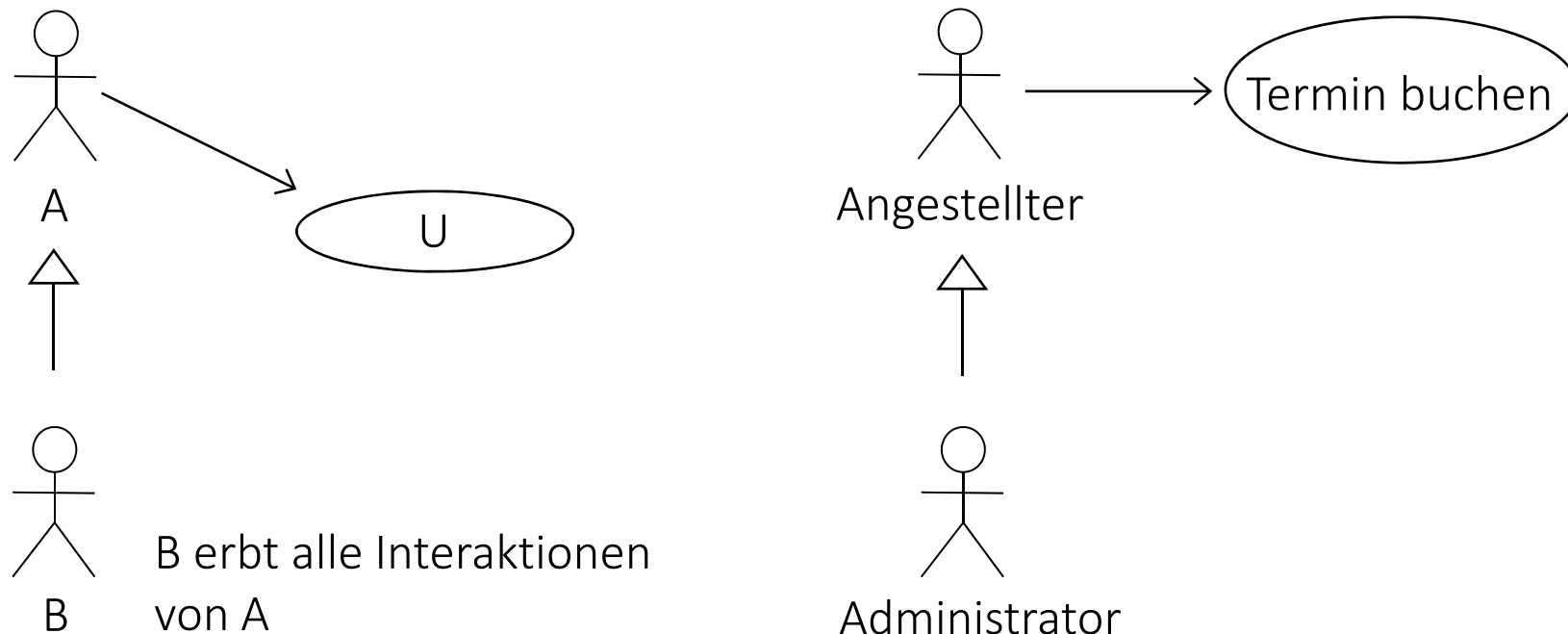
Gefahren und Missverständnisse beim Definieren von Anwendungsfällen

- Die Granularität des Anwendungsfalls ist zu fein
 - Beispiel: Anwendungsfall „PIN eingeben“
 - spätere Zusammenfassung von Anwendungsfällen birgt die Gefahr von Inkonsistenzen
- Die Granularität des Anwendungsfalls ist zu grob
 - Beispiel: Anwendungsfall „Bankomat bedienen“
 - zu wenig differenzierte Basis für den weiteren Entwurf
- Das Anwendungsfalldiagramm soll nichts über den zeitlichen Ablauf von Anwendungsfällen aussagen
- Anwendungsfälle werden nicht in Anwendungsfälle kleinerer Granularität zerlegt

Erweiterte Beschreibungstechniken

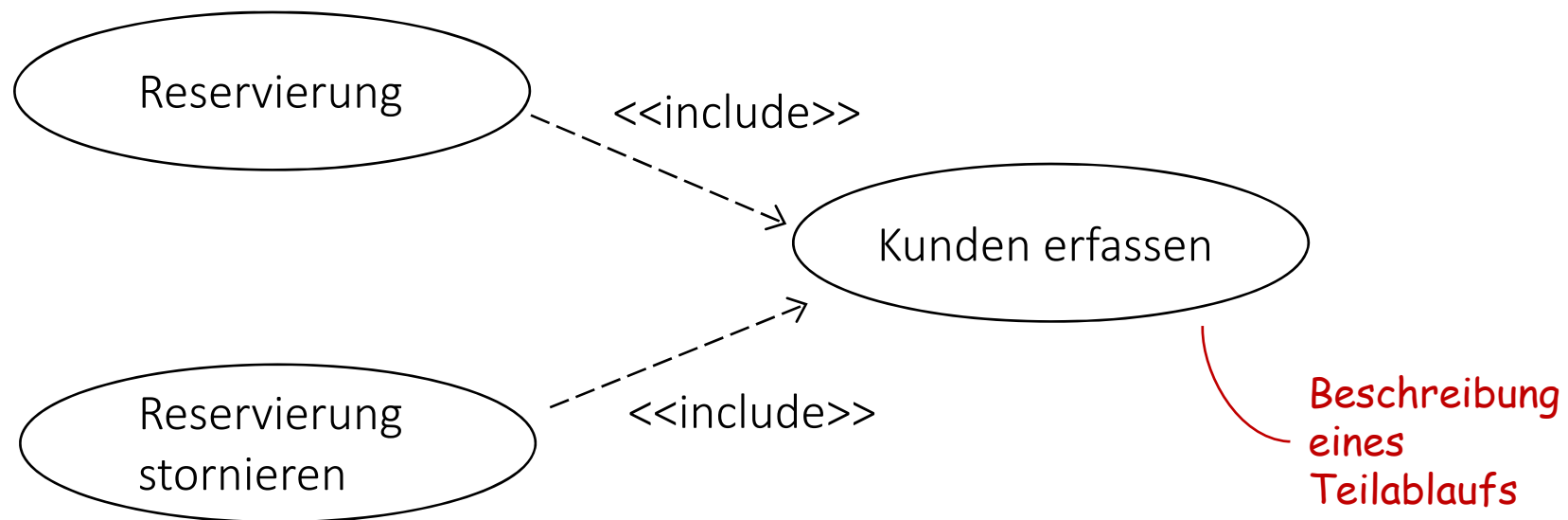
1. Vererbungsbeziehung zwischen Akteuren

Beispiel:

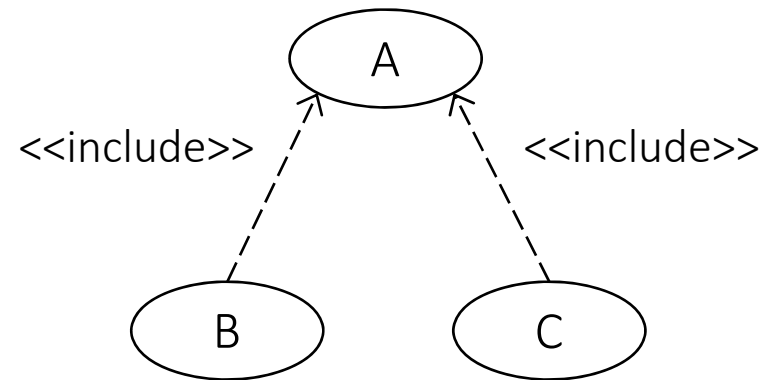


Erweiterte Beschreibungstechniken

2. Include-Beziehung zwischen Use Cases



Die Include-Beziehung



- Der (Teil-)Anwendungsfall A
 - muss nicht in sich abgeschlossen sein
 - kennt nicht die Anwendungsfälle, in die er eingefügt wird
- Der einfügende Anwendungsfall B
 - weiß, welche Anwendungsfälle er einfügt
 - muss (zusammen mit A) vollständig sein

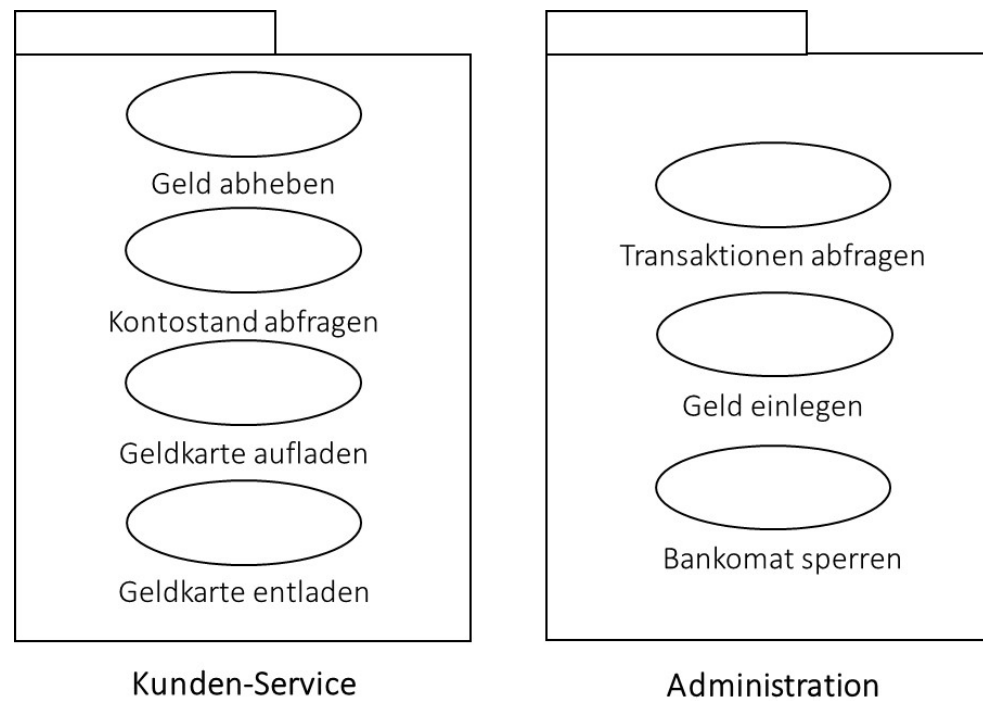
Ein Anwendungsfall kann mehrere andere Anwendungsfälle einbeziehen

Warum benutzt man Include-Beziehungen?

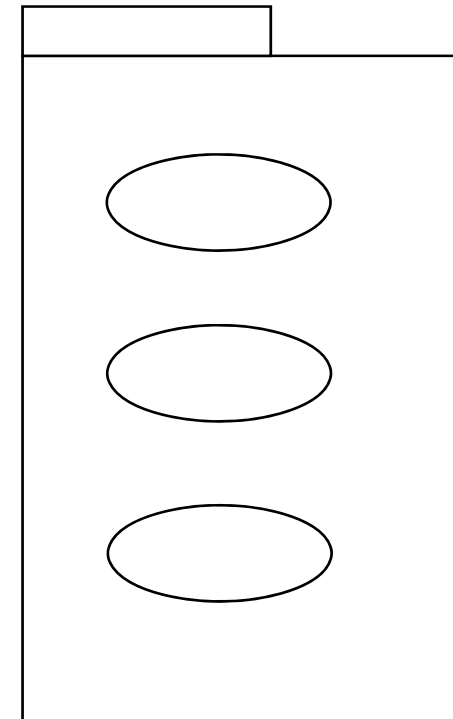
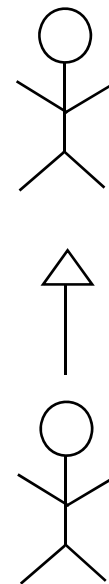
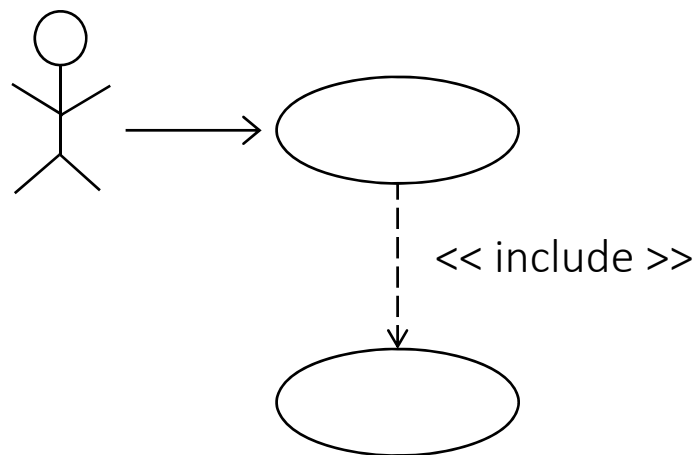
- Um zu zeigen, dass Teile des Verhaltens in verschiedenen Anwendungsfällen gemeinsam sind
- Um zu vermeiden, dass dasselbe Verhalten mehrere Male und in unterschiedlicher Art und Weise beschrieben wird

Gruppieren von Anwendungsfällen

- Typische Anzahl von Use Cases in realen Anwendungen: 100 – 150
 - ⇒ Use Case Diagramme werden unübersichtlich
 - ⇒ Gruppierung notwendig



Zusammenfassung Use Case Diagramme



2.1.2 Textuelle Beschreibung der Anwendungsfälle

Die Beschreibung eines Anwendungsfalls beinhaltet folgende Punkte:

- Kurzbeschreibung
- Detailinformationen:
 - beteiligte Akteure, Auslöser des Anwendungsfalls
 - Vorbedingung: unter welcher Bedingung kann der Anwendungsfall ausgelöst werden?
 - Nachbedingung: welche Bedingung gilt nach Ablauf des Anwendungsfalls?
 - Basisablauf
 - alternative Abläufe, Ausnahmen
 - Welche Klassen des fachlichen Klassendiagramms sind in den Anwendungsfall involviert?

Beispiel (1)

Anwendungsfall: Kontostand abfragen

Initiator: Bankkunde

Vorbedingung: Bankomat betriebsbereit

Basisablauf:

1. Der Anwendungsfall beginnt, indem ein Kunde seine Karte in den Kartenleser einführt.
2. Es erscheint ein Begrüßungsbild auf dem Display des Bankomaten.
3. Nach einer kurzen Pause (1-3 Sek.) erscheint die Aufforderung zur Eingabe der PIN auf dem Display.
4. Der Kunde gibt seine 4-stellige PIN ein und bestätigt seine Eingabe. Das System prüft die PIN.
5. Falls die PIN korrekt war, erscheint das Menü zur Auswahl der möglichen Transaktionen.
6. Der Kunde wählt „Kontostand abfragen“.
7. Bei einem Kunden der Bank fordert das System den Kontostand vom Zentralrechner an und gibt ihn am Display aus. Die Kontostandsabfrage wird protokolliert.

...

Beispiel (2)

Nachbedingung: Karte wurde ausgegeben und entnommen

Alternativen:

4a. *Der Kunde bricht die Eingabe ab.*

Die Karte des Kunden wird ausgegeben.

5a. *Die eingegebene PIN war nicht korrekt.*

...

7a. *Der Kunde ist Kunde einer anderen Bank.*

Es wird die Meldung ausgegeben, dass der Kontostand nicht angezeigt werden kann.

Involvierte Klassen:

Bankomat, Kontostandabfrage (c), Karte, Konto, Bank

(c) = es werden Objekte dieser Klasse erzeugt

(m) = es werden Objekte dieser Klasse verändert

(r) = es werden Objekte dieser Klasse gelesen

Keine Angabe = unbestimmt

Ablaufbeschreibung

- Beschreibung des Ablaufs aus der Sicht des Benutzers
 - Wie wird der Anwendungsfall gestartet?
 - Welche Informationen werden zwischen System und Benutzer ausgetauscht?
 - Nicht auf die Details der Eingabe eingehen: „Geschlecht eingeben“ statt „Knopf M oder Knopf W drücken“
 - Wie reagiert das System auf die erhaltenen Informationen?
 - In welcher Reihenfolge laufen die Aktionen ab?

2.1.3 Modellierung des Anwendungskerns

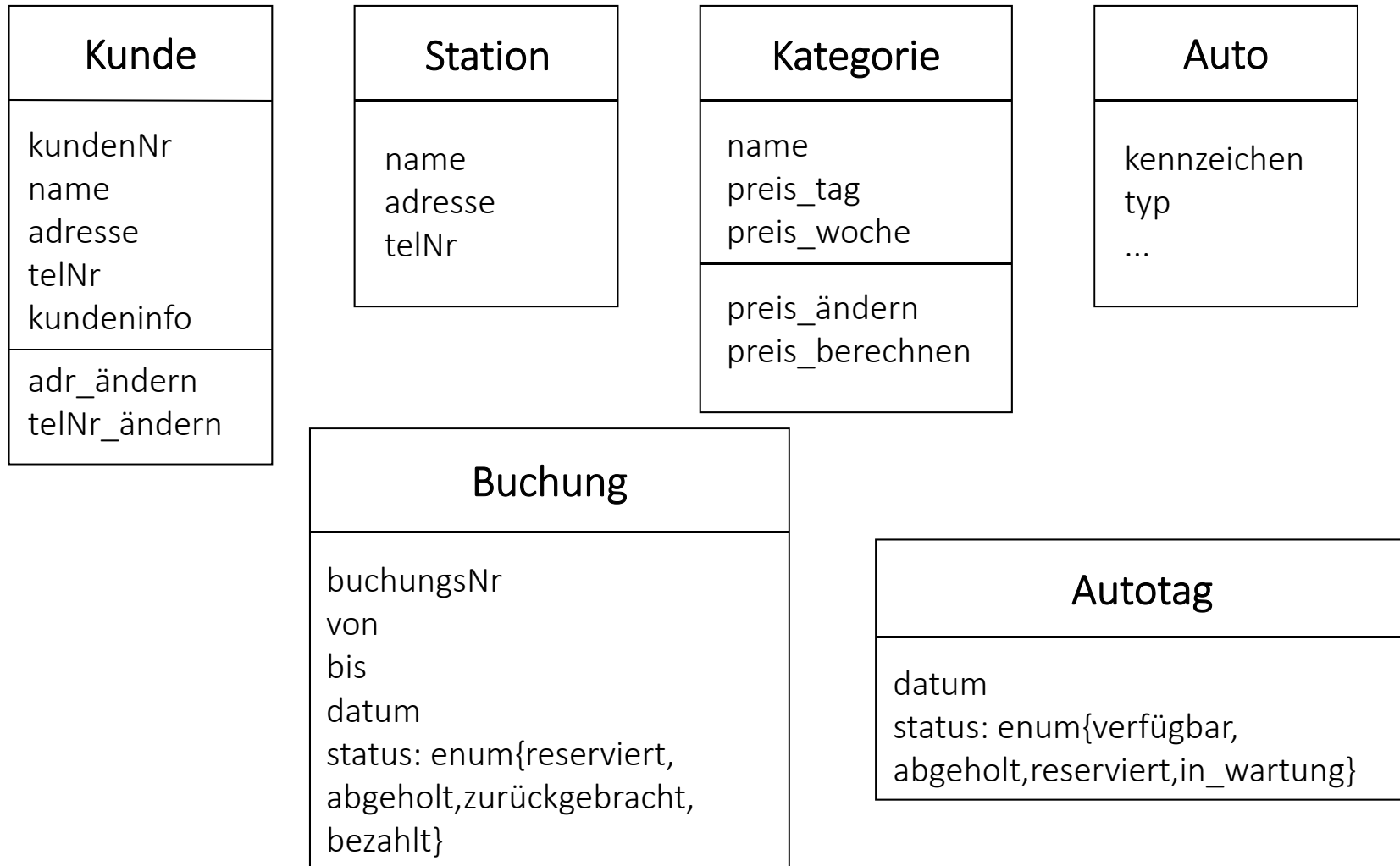
- Die statischen Konzepte des Anwendungsbereichs werden im **fachlichen Klassendiagramm** modelliert
 - Die Klassen werden im folgenden **Fachklassen** (*entity classes, business classes*) genannt
- Fachliche Regeln werden durch **Invarianten** beschrieben
- Zusätzlich werden Klassen mit signifikanten Zuständen durch **Zustandsdiagramme** beschrieben

Die Stabilität des Anwendungskerns ist essentiell für die gesamte Entwicklung.

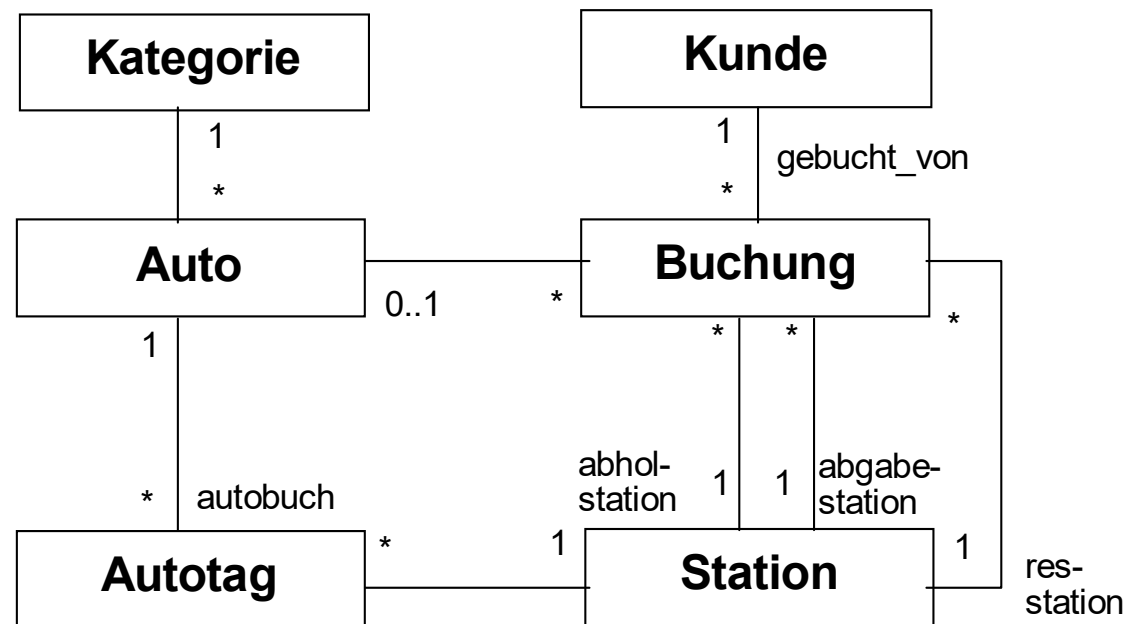
Arbeitsschritte Klassendiagramm

- Finden von Klassen
 - Modellierung von statischen fachlichen Konzepten, die in den Anwendungsfällen eine Rolle spielen (Kunde, Konto, Vertrag, ...)
- Den Klassen Attribute zuordnen, Klassen durch Assoziationen verbinden
- Generalisierungsbeziehungen herausarbeiten
- Operationen und Attributtypen dürfen fehlen!

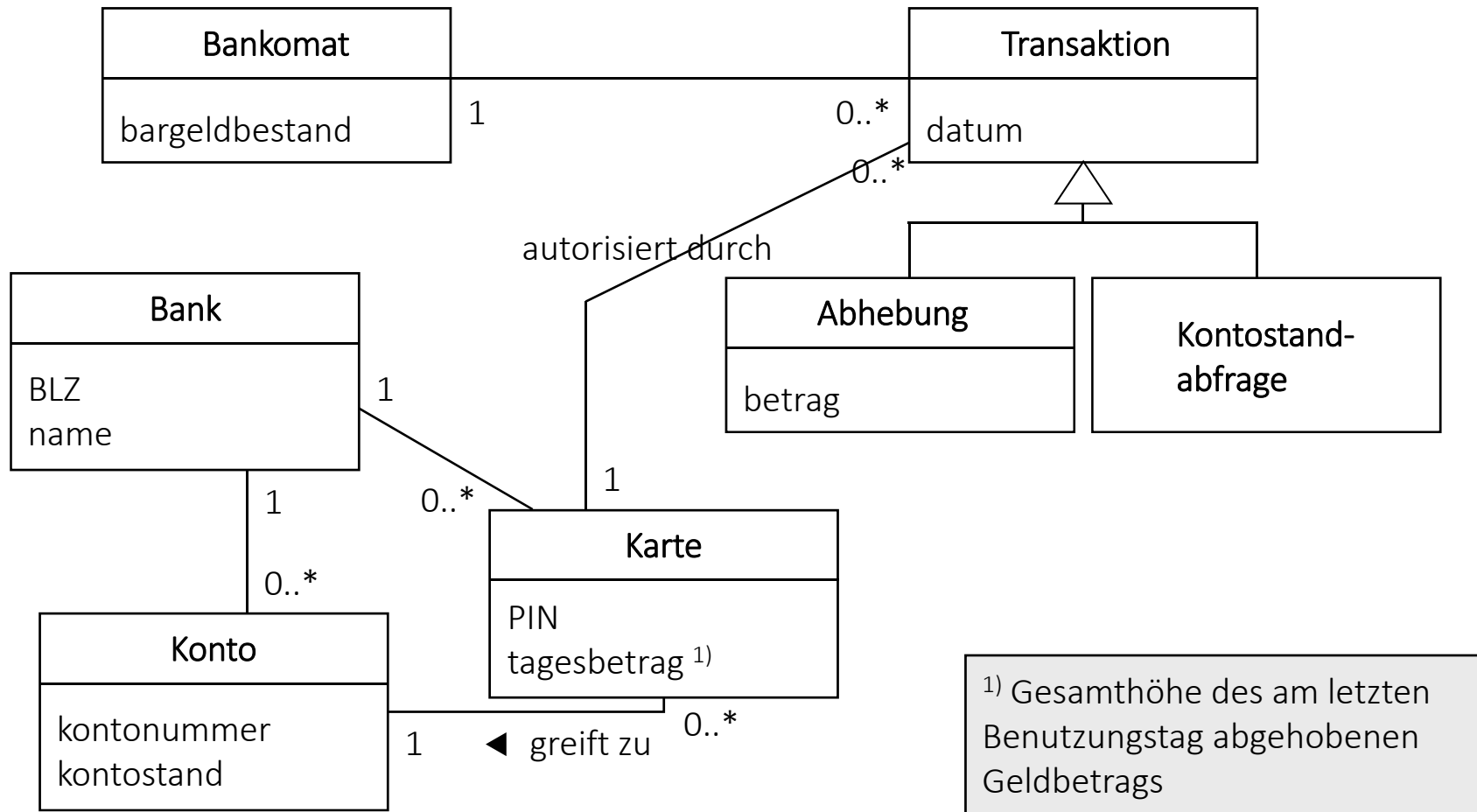
Fachklassen – Beispiel Autovermietung



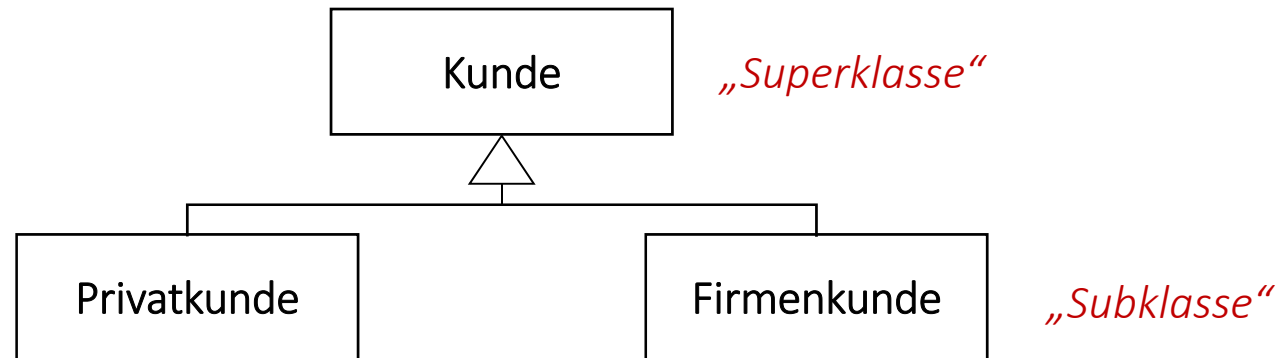
Klassendiagramm - Autovermietung



Beispiel – Klassendiagramm Bankomat



Modellieren von Generalisierung



Gründe und Bedingungen für die Einführung einer Vererbungshierarchie

- gemeinsame Attribute und Assoziationen
- Superklasse muss eine fachliche Entsprechung besitzen
- Erweiterbarkeit der Generalisierungshierarchie
- Super- und Subklassen werden in gemeinsamen Kontexten auf der Basis von Polymorphie verwendet
- Die Objekte in der Klassenhierarchie haben gemeinsame Operationen (d.h. ähnliches Verhalten)
- Es gibt mehrere Subklassen

Techniken zum Identifizieren von Klassen

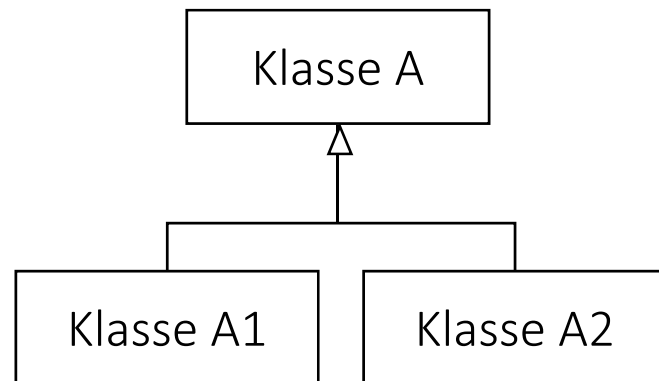
- Informationsquellen: Interviews mit Anwendern, Sammlung von Materialien und Gegenständen (Verträge, Kundenakten, Rechnungen, Werkstattprotokoll, Schlüssel, usw.)
- Mind-Maps
- Glossar

Muster („Analysemuster“)

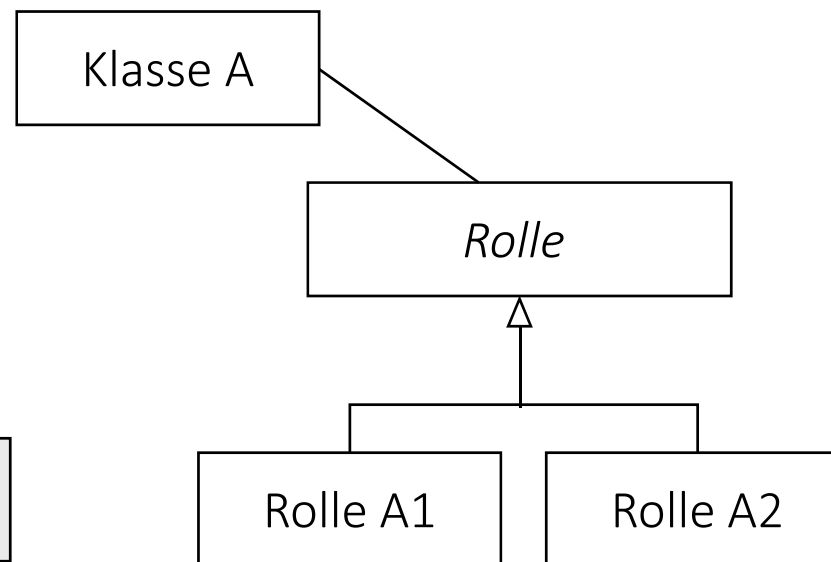
Analysemuster

„Wechselnde Rollen“

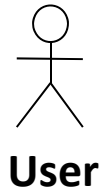
- **Problem:** Ein Objekt einer Unterklasse A1 einer Oberklasse A kann in eine andere Unterklasse A2 von A wechseln.
- **Lösung:** Einführung einer abstrakten Rollenklasse.



Literatur:
M. Fowler: Analysis Patterns, Addison-Wesley



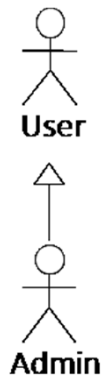
Akteure – Klassen – Rollen



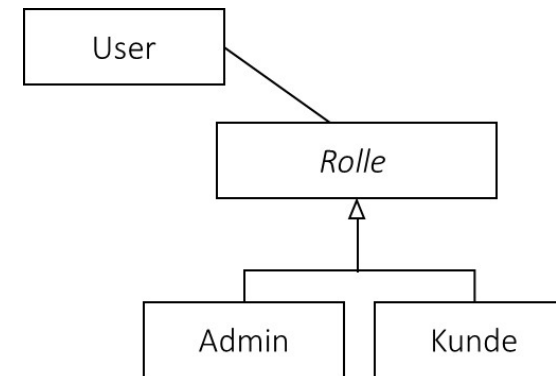
Akteur als Rolle
interagiert mit dem System



Akteur als Klasse
„statisches Wissen“ in System
über den Akteur

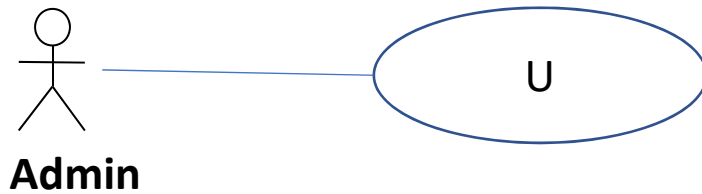


Teilmengenbeziehung
zwischen zugeordneten
Use Cases

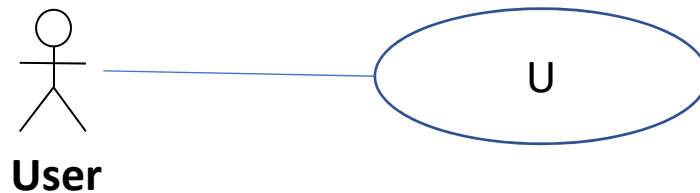


Basis für ein statisches Rollen- und
Rechtesystem

Use Cases, Rollen und Rechte

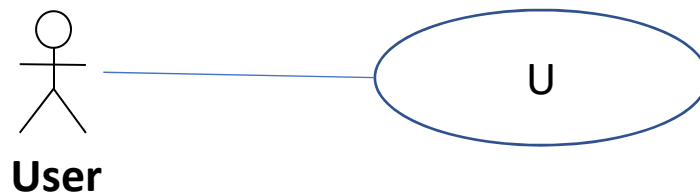


versus



Vorbedingung von U:
User hat die Rolle „Admin“

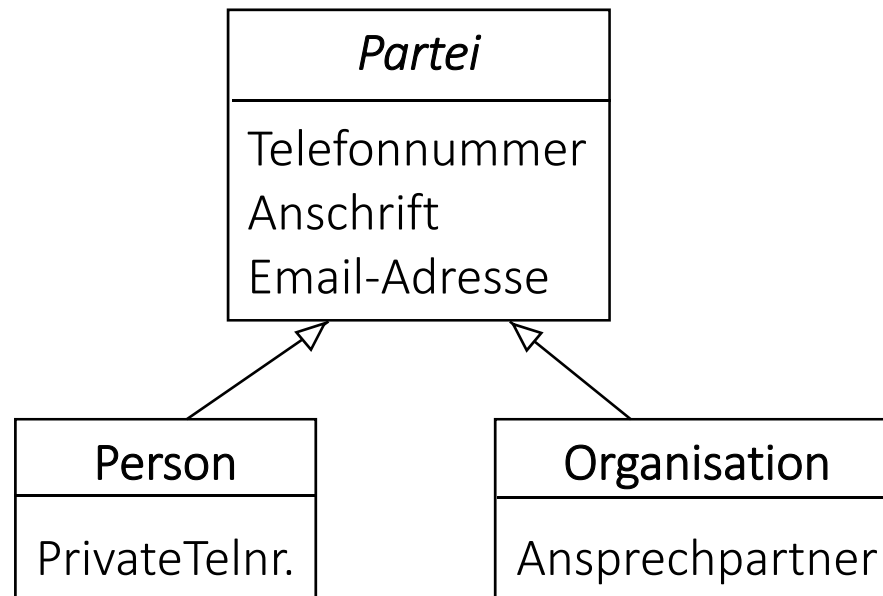
versus



Vorbedingung von U:
User ist im Zugriffskontrollsystem zu U
berechtigt

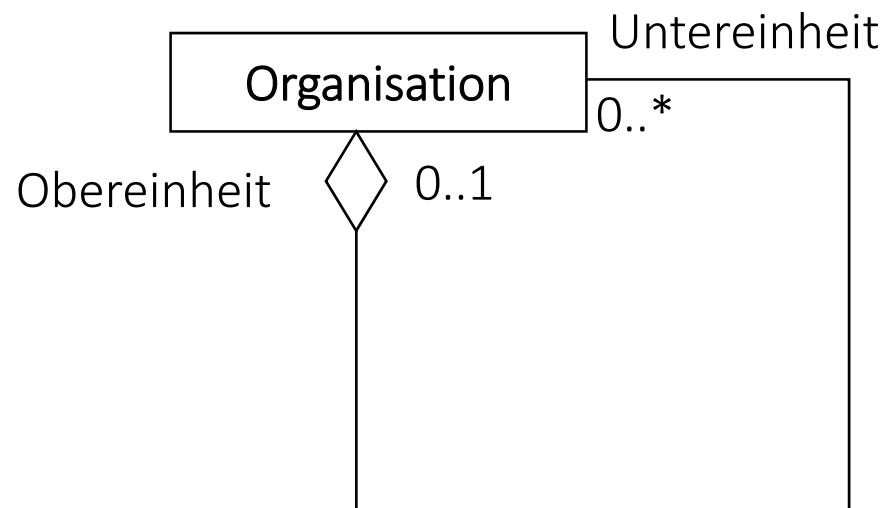
Analysemuster: „Partei“

- **Problem:** Organisationen und Personen spielen ähnliche Rollen gegenüber dem System
- **Lösung:** Abstrakte Oberklasse „Partei“.
- **Regel:** Überprüfe bei Personen und Organisationen, ob es sich nicht um allgemeinere „Parteien“ handelt.



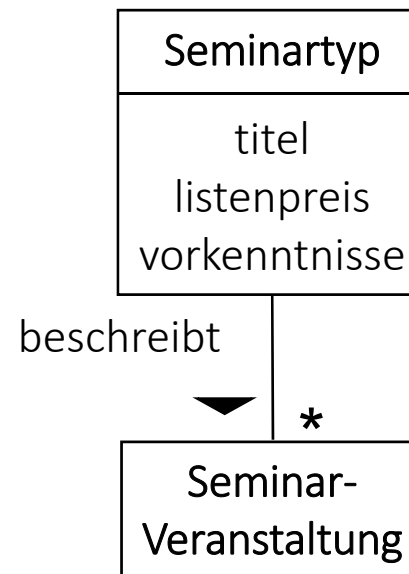
Analysemuster: „Organisations-Hierarchie“

- **Engl. Name:** Organization Hierarchies
- **Problem:** Eine Organisation ist hierarchisch in Unterorganisationen gegliedert
- **Lösung:** Flexible Ordnungsassoziation
- Spezialfall des Composite-Patterns



Analysemuster: „Powertype“

- James Odell: „A *power type* ist an object type whose instances are subtypes of another object type.“
- „Powertype“
 - Bedeutet Einführung einer „Meta-Ebene“
 - Objekte auf Meta-Ebenen beschreiben Eigenschaften anderer Objekte
 - Beispiel:



Invarianten

- Fachliche Invarianten
 - formulieren fachliche Regeln des Anwendungsbereichs
 - Beispiel: Plausibilitätsregeln für Buchungsdaten, Zeitraum zwischen Wartungen, Preise,...
- Modell-Invarianten
induziert durch die Art der Modellierung
 - Der Status einer Buchung und des dazugehörigen Autotags sind konsistent
 - Ist eine Buchung im Status „abgeholt“, muss die Buchung mit einem Auto verbunden sein
- Die Invarianten werden meist textuell formuliert
- Formale Modellierung, siehe Kapitel 5

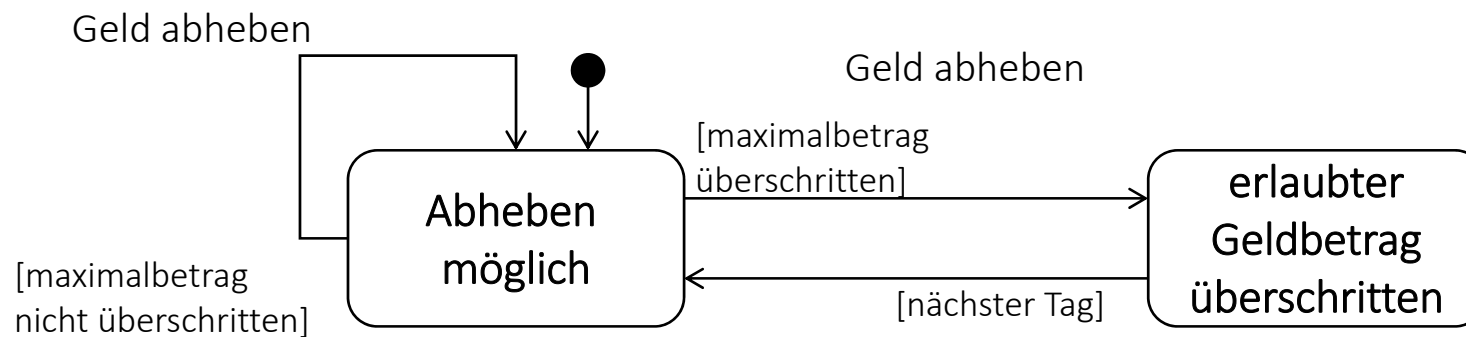
Zustandsdiagramme

- Von den Klassen im Klassendiagramm, die signifikante Zustände und Zustandswechsel haben, werden Zustandsdiagramme erstellt
- Beispiele:
 - Vertrag (Zustände: Vertrag angelegt, Vertrag abgeschlossen, Vertrag gekündigt)
 - Bestellung (Zustände: Bestellung eingegangen, Bestellung in Bearbeitung, ...)
- Oft haben diese Klassen ein Attribut *status* mit einem Enumerationstyp, der die Zustände beschreibt

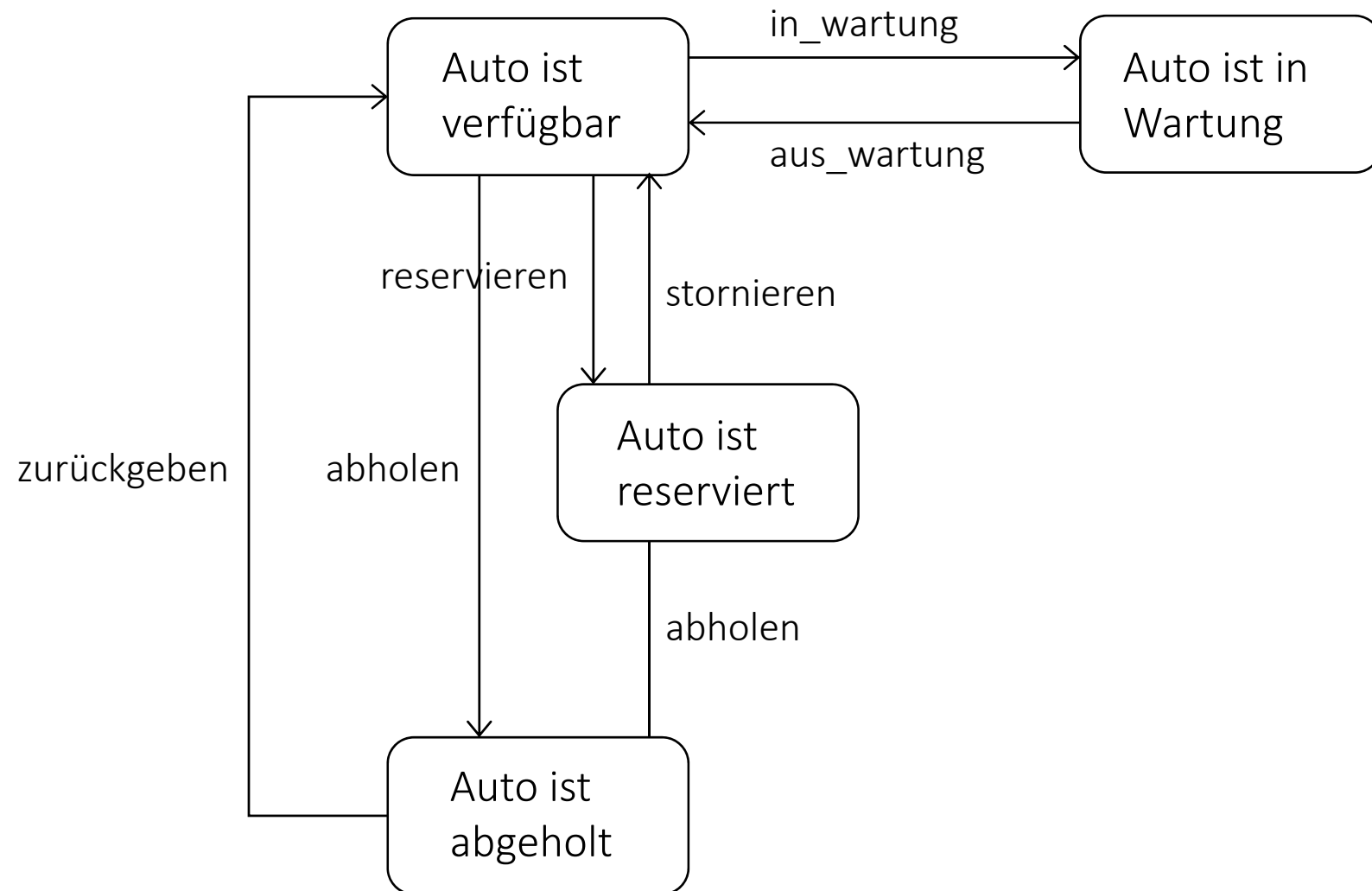
Zustandswechsel

- Meist werden die Zustandswechsel durch die Ausführung von Anwendungsfällen ausgelöst
- Somit beschreibt das Zustandsdiagramm einen Querschnitt durch die Anwendungsfälle
- Mögliche Zustandswechsel führen oft zur Definition weiterer Anwendungsfälle

Beispiel: Klasse Bankomat-Karte



Beispiel: Klasse Autotag



Fragen

- Welche Zustände haben Objekte der Klasse?
- Für jeden Zustand: Welche Zustandsübergänge sind sinnvoll?
- Für jeden Zustandsübergang: In welchen Use Cases findet dieser Zustandsübergang statt?
- Besondere Prüfung von Zuständen, die keine ausgehenden Übergänge besitzen

2.2 Pflichten- und Lastenheft

- Das Pflichtenheft ist Ergebnis der Anforderungsspezifikation
- Das Pflichtenheft enthält u.a.
 - Das Anwendungsfalldiagramm
 - Die Kurzbeschreibung der Anwendungsfälle
 - Die Detailbeschreibung der Anwendungsfälle
 - Das fachliche Klassendiagramm
 - Evtl. (Screenshots des) Oberflächenprototyps
- Das Pflichtenheft wird während des Entwurfs laufend verfeinert und aktualisiert
 - ➔ inkrementeller Entwurf

Pflichtenheft nach DIN 69905

- „Das Pflichtenheft umfasst die vom Auftragnehmer erarbeiteten Realisierungsvorhaben aufgrund der Umsetzung des vom Auftraggeber vorgegebenen Lastenhefts.“
 - Vertraglich bindende, detaillierte Beschreibung einer zu erfüllenden Leistung
 - Die Inhalte des zuvor erstellten Lastenhefts sind präzisiert, vollständig, nachvollziehbar sowie mit technischen Festlegungen der Betriebs- und Wartungsumgebung verknüpft
- Nach Lieferung der Software wird häufig über einen Akzeptanztest festgestellt, ob die Software die Forderungen des Pflichtenhefts erfüllt.

Rolle des Pflichtenhefts im Entwurf

- Das Pflichtenheft ist Basis für
 - Aufwandsschätzungen und Projektplanung
 - die Risikoanalyse
 - die Angebotserstellung
 - den Einkauf externer Komponenten bzw. die Vergabe von Teilprojekten an Dritte
 - die weitere fachliche und technische Spezifikation

Pflichten- und Lastenheft

- Lastenheft
 - wird vom Auftraggeber erstellt
 - Beschreibung eines groben Konzepts aus Anwendersicht
 - Umfasst alle Anforderungen für den Auftragnehmer
- Pflichtenheft
 - Wird von Auftraggeber und Auftragnehmer gemeinsam erstellt
 - Baut auf den Vorgaben des Lastenhefts auf

Gliederung des Pflichtenhefts

Project Drivers

1. The Purpose of the Project
2. The Stakeholders

Project Constraints

3. Mandated Constraints
4. Naming Conventions and Terminology
5. Relevant Facts and Assumptions

Functional Requirements

6. The Scope of the Work
7. Business Data Model & Data Dictionary
8. The Scope of the Product
9. Functional Requirements

Non-functional Requirements

10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements
13. Operational and Environmental Requirements
14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural Requirements
17. Legal Requirements

Project Issues

18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Tasks
22. Migration to the New Product
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

Quelle:

Volere Template

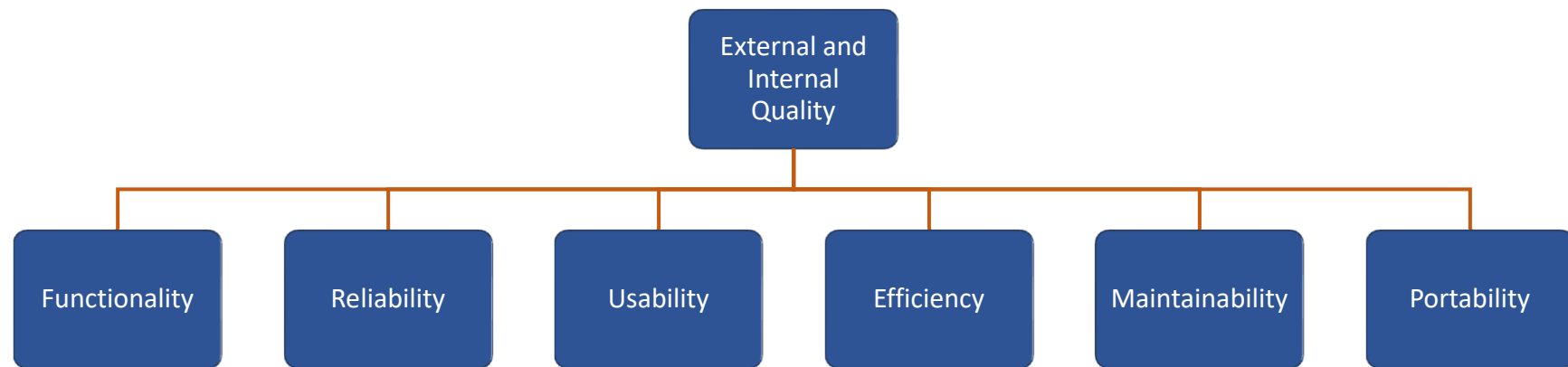
<http://www.volere.co.uk/template.htm>

„Funktionale“ Anforderungen

- Im Rahmen der Pflichtenhefterstellung wird von „funktionalen“ und „nichtfunktionalen“ Anforderungen gesprochen
- Funktionale Anforderungen legen fest, was das System tun soll und werden z.B. in Form von Klassen, Use Cases oder Geschäftsprozessen dokumentiert

Nichtfunktionale Anforderungen

- Nichtfunktionale Anforderungen legen die Qualitätseigenschaften des Systems fest



Qualitätskriterien nach ISO 25010

Spezifikation nicht-funktionaler Anforderungen

- Interpretation der sechs Qualitätskriterien im Kontext des zu beschreibenden Systems, bezogen auf z.B.
 - das System als Ganzes
 - bestimmte Use Cases
 - das fachliche Klassendiagramm
- D.h. nichtfunktionale Anforderungen sind nicht isoliert, sondern beziehen sich auf funktionale Anforderungen

Beispiele /1

- Nichtfunktionale Anforderungen des Systems als Ganzes
 - *The system shall not emit noxious gases that damage people's health.*
 - *The product shall comply with corporate branding standards.*
 - *The product shall achieve 99 percent uptime.*
 - *The product shall provide 10 minutes of emergency operation should it become disconnected from the electricity source.*
 - *The product shall work on the last four releases of the five most popular browsers.*
 - *The new release of the system must be able to access data from the previous five releases.*

Quelle: Volere Template

Beispiele /2

- Anforderungen eines bestimmten Use Cases
 - *The sensor X should be polled every 10 seconds.*
 - *The product shall download the new status parameters within 5 minutes of a change.*
 - *Use Case X shall have a maximum response time of 2 seconds.*

Beispiele /3

- Zugriffskontrollmatrix

Business Classes

↙

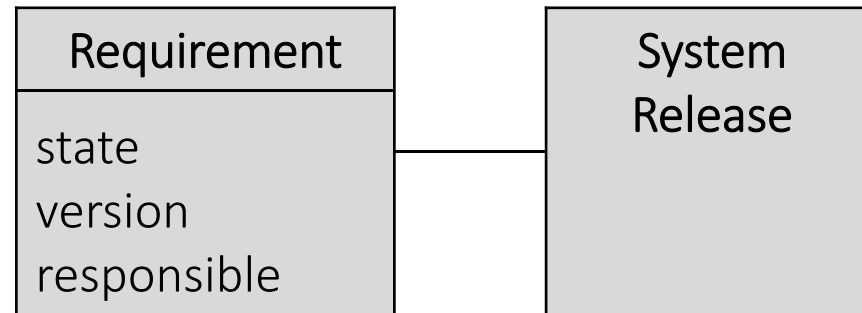
	Health Record	Patient	Access Log
Head of Department	r all w all c v	r all w - c -	r all w - c -
Practitioner	r own department w own patient c v	r own department w - c -	r - w - c -
System Administrator	r all w - c v	r all w - c -	r all w - c -
Patient Administrator	r all w - c -	r all w all c v	r all w - c -

Actors

↗

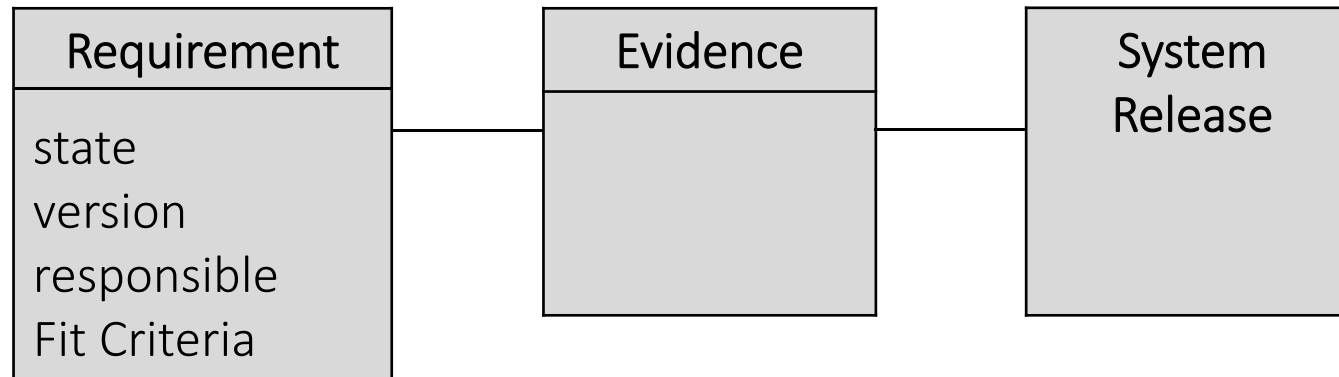
r: read
w: write
c: create

Requirement Lifecycle



- Auch Requirements ändern sich und besitzen einen Lifecycle.
- Requirements können als Objekte betrachtet werden, besitzen z.B.
 - eine Versionsnummer
 - einen Status (*erhoben, implementiert, verifiziert ...*)
 - und beziehen sich auf ein System(-Release), in der sie implementiert sind

Fit Criteria und Evidenz



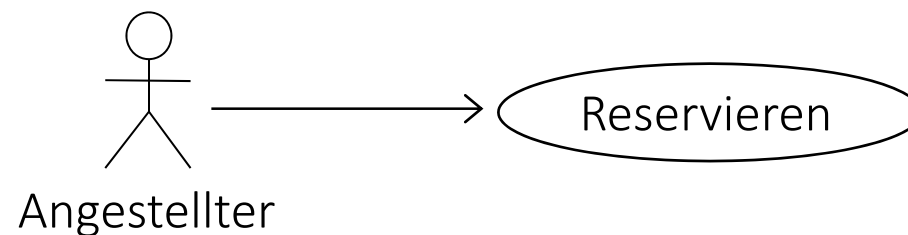
- Fit Criteria definieren, wann eine Anforderung als erfüllt zu bezeichnen ist (z.B. Testabdeckungskriterium)
- Die Evidenz definiert bzw. belegt, mit welchem Argument ein Requirement im System erfüllt ist
 - Beispiel: Requirement ist ein Use Case, die Evidenz sind die Testfälle dieses Use Cases
- Die Verbindung von Requirement, Fit Criteria, Evidenz und System(-komponenten) wird auch Nachverfolgbarkeit (Traceability) genannt

Nachverfolgbarkeit

- Ein wichtiges Ziel für den Entwurfsprozess ist die Nachverfolgbarkeit (Traceability) von Anforderungen
 - In welchen Komponenten werden welche Anforderungen umgesetzt?
 - Welche Testfälle testen welche Anforderungen?
- Nachverfolgbare Anforderungen erleichtern die Wartbarkeit und Qualität des Systems, sind aber kostenintensiv
- Bei sicherheitskritischen Systemen (z.B. in der Medizin- oder Verkehrstechnik) muss die Nachverfolgbarkeit nachgewiesen werden

Beispiel

- Use Cases werden mit (System-)tests verbunden

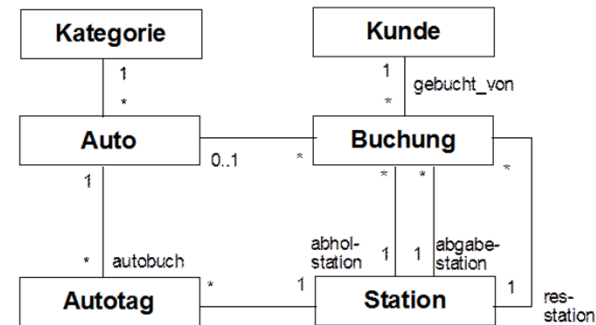
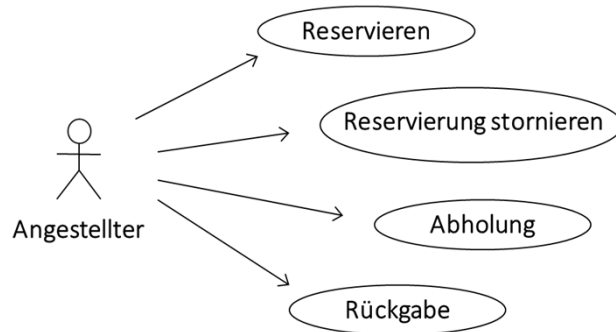


Kunde mit Kundennummer 1234 reserviert ein Auto der Kategorie A vom 1.3.2015 bis 21.3.2015 mit Abhol- und Rückgabestation Innsbruck
Neukunde Max Mustermann reserviert ein Auto der Kategorie A vom 1.3.2015 bis 21.3.2015 mit Abhol- und Rückgabestation Innsbruck
Neukunde Max Mustermann reserviert ein Auto der Kategorie A vom 1.3.2015 bis 21.3.2015 mit Abholstation Innsbruck und Rückgabestation Salzburg.
...

Use Cases und User Stories

- User Stories werden im Rahmen von agilen Methoden verwendet
- User Story: Beschreibung einer Systemfunktionalität *aus Nutzersicht* in wenigen Sätzen
 - Geschrieben in Kundenterminologie
 - In ein bis drei Wochen implementierbar
 - werden priorisiert
 - Typischerweise auf Post-Its/Whiteboards oder in den Tickets der Projektmanagementwerkzeuge gehalten

Beispiel



+ textuelle Beschreibungen

versus

Sachbearbeiter möchte eine Reservierung für ein Auto anlegen können

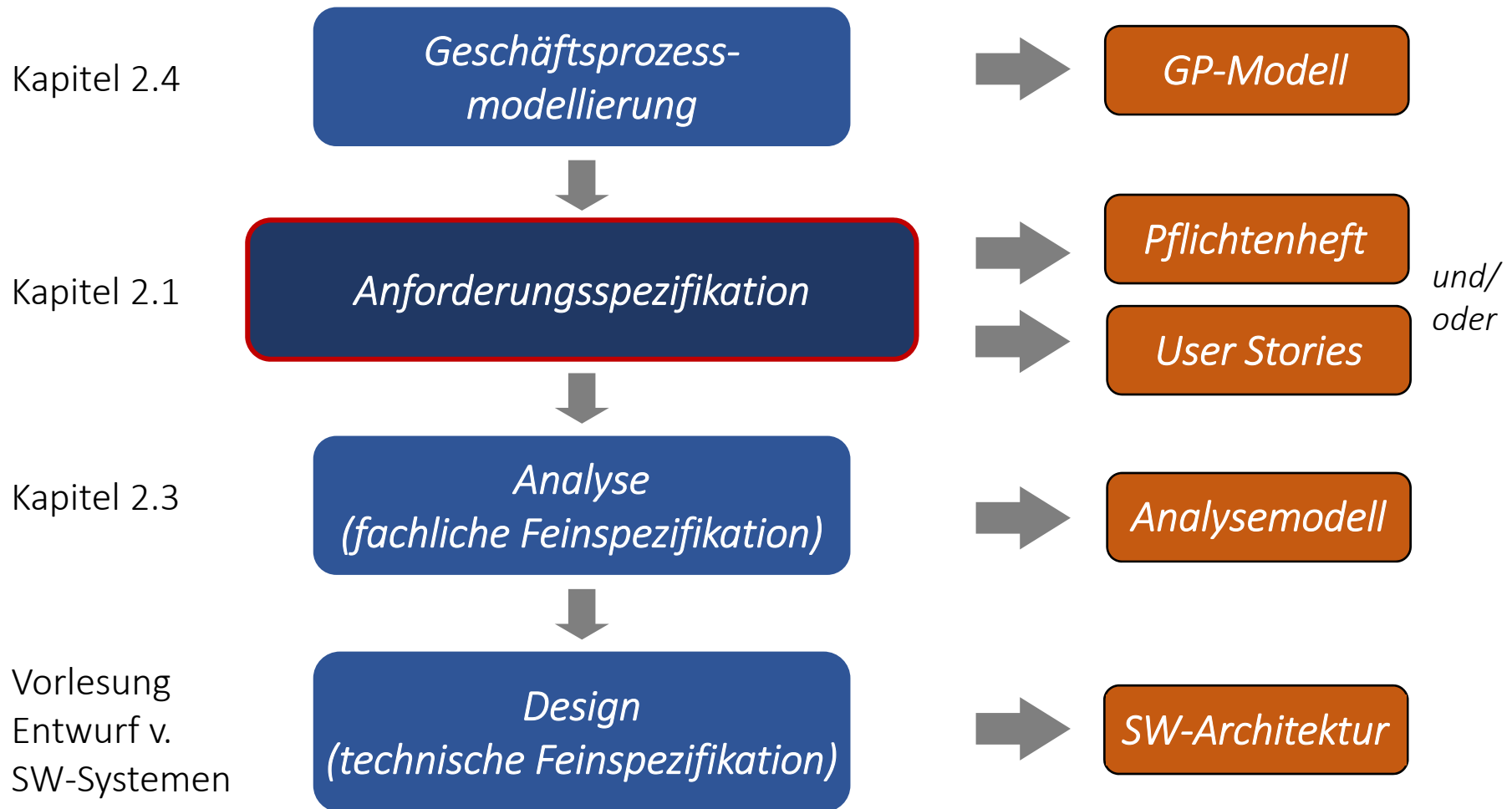
Kunde soll eine Reservierung bis eine Woche vor dem Leihtermin kostenlos stornieren können

Der Sachbearbeiter soll die Rückgabe des Autos mit dem Statusbericht des Autos ins System eintragen können.

Use Cases und User Stories

- User Stories werden als flexibles Planungsinstrument im Rahmen des Entwurfs eingesetzt, eignen sich jedoch nicht zur Dokumentation
 - Zu wenig allgemein
 - Zu personenzentriert
 - Nicht auf Vollständigkeit fokussiert
- Use Case-Beschreibungen werden langfristig gepflegt
 - Für die Dokumentation langjähriger, qualitativ hochwertiger Produkte geeignet
 - Dienen als Vertragsgrundlage

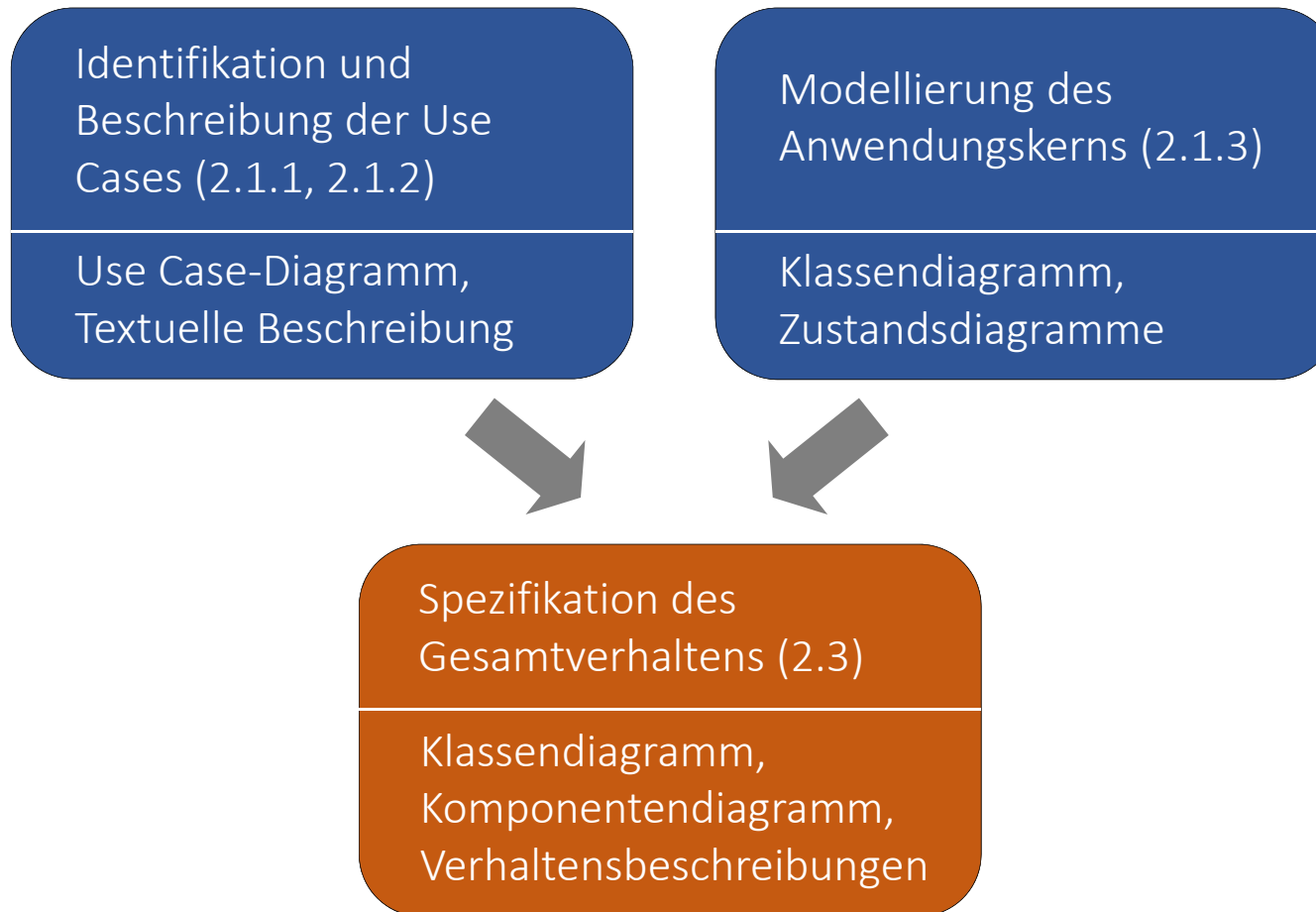
Einordnung



2.3 Analyse

funktionale Sicht

objektorientierte Sicht



Weitere Entwurfsschritte - Analyse

- Analyse (fachliche Feinspezifikation)
 - textuelle Beschreibungen der Anwendungsfälle werden mit Hilfe von Sequenzdiagrammen als Nachrichtenflüsse modelliert („Szenarios“)
 - das Klassendiagramm wird erweitert und verfeinert
 - Komponenten werden gebildet

Szenarios (1)

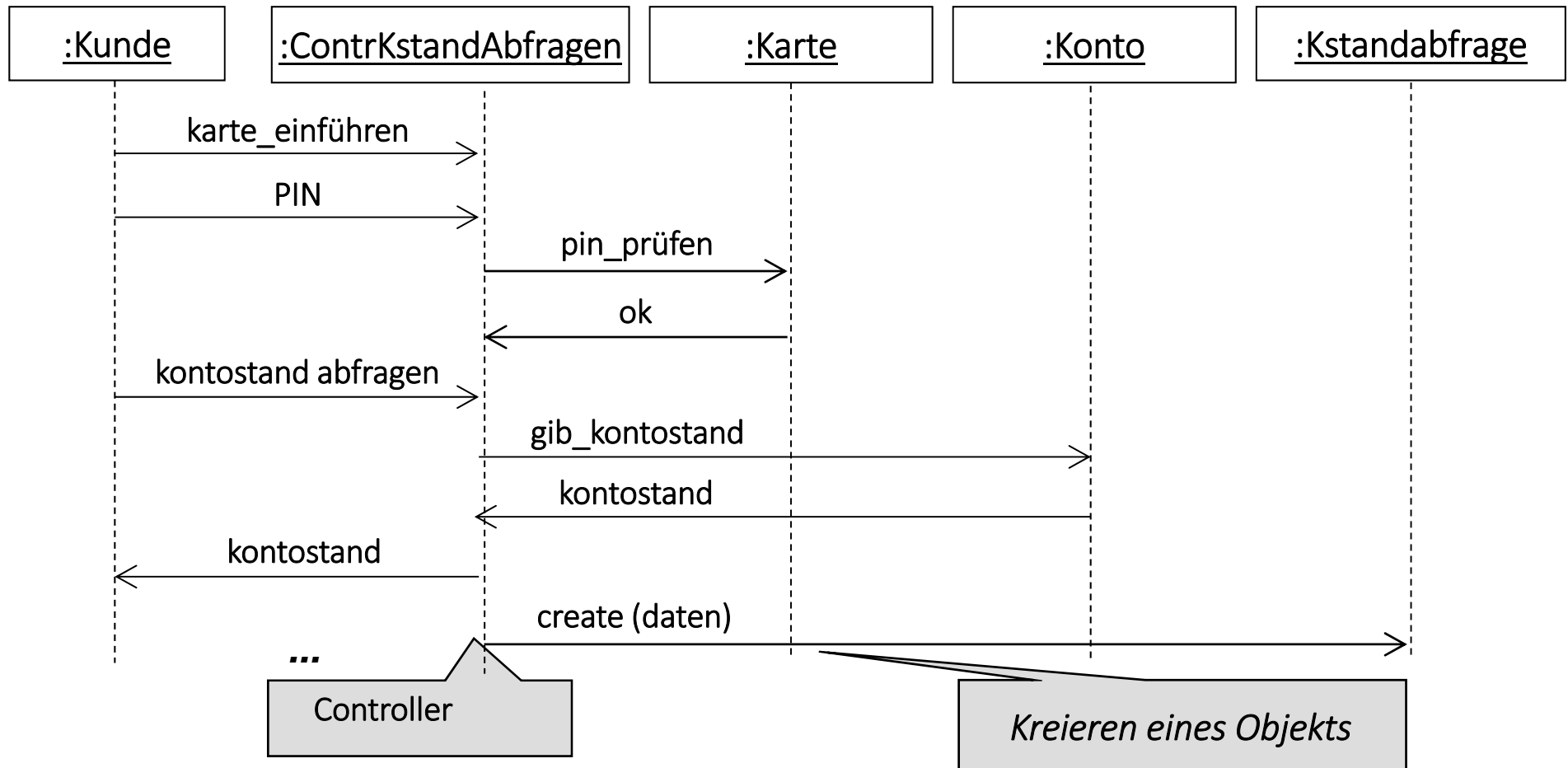
- In den Szenarios wird der textuell beschriebene Ablauf eines Anwendungsfalls als Nachrichtenfluss zwischen Objekten dargestellt (in einem Interaktionsdiagramm, z.B. Sequenzdiagramm)
- Ein Sequenzdiagramm beschreibt einen möglichen Ablauf, d.h. ein charakteristisches Beispiel eines Ablaufs
- „Primäres Szenario“: Beschreibung des Basisablaufs
- „Sekundäre Szenarios“: Beschreibung von Varianten
- Pro Anwendungsfall wird also mindestens ein Sequenzdiagramm entwickelt

Szenarios (2)

- Ziel der Entwicklung von Szenarios: der informell beschriebene Ablauf des Anwendungsfalls wird in die objektorientierte Sicht überführt
 - Übergehen der Sichtweise von Anwendungsfällen auf die Sichtweise von Objekten, die durch Nachrichtenaustausch kooperieren
 - Definition weiterer Klassen
 - Definition von Operationen im Klassendiagramm
 - Assoziationen im Klassendiagramm werden gerichtet

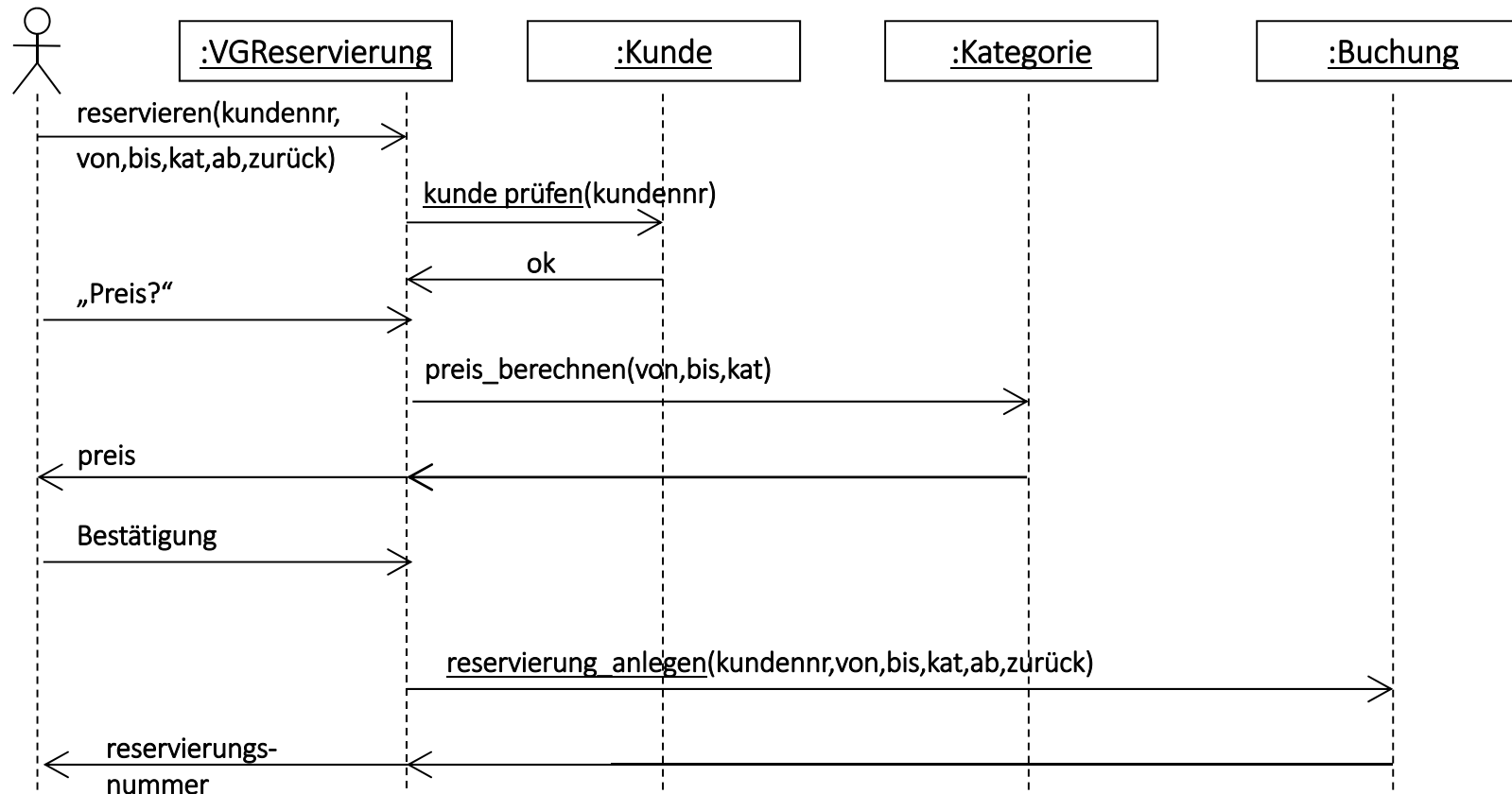
Beispiel – Anwendungsfall Kontostand abfragen

Szenario „Kontostand abfragen“



Beispiel: Use Case Reservierung (1)

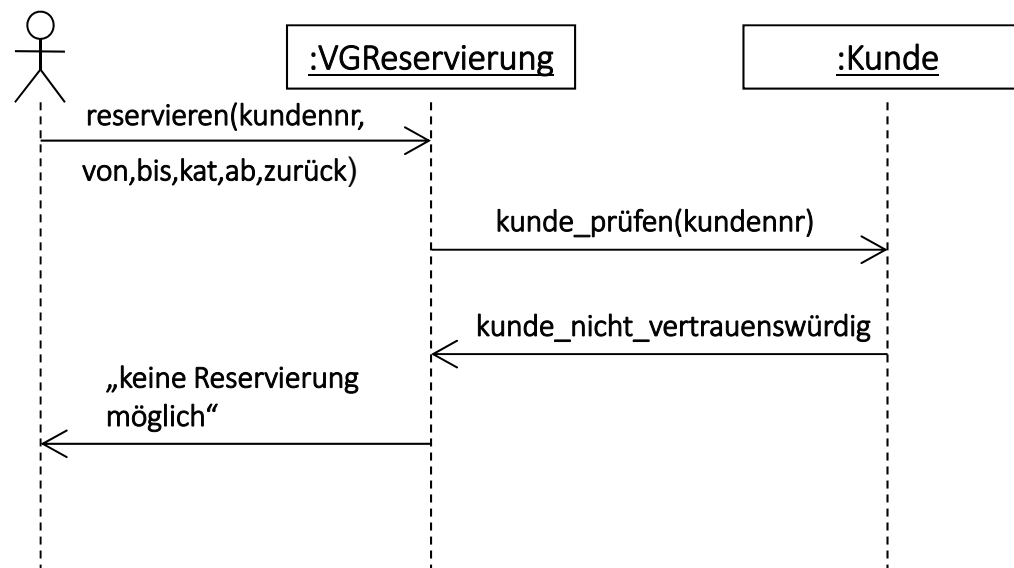
Primäres Szenario



Bemerkung: Die Verwendung von Klassenoperationen (unterstrichen) vereinfacht das Diagramm (extensive Nutzung nicht ratsam).

Beispiel: Use Case Reservierung (2)

Sekundäres Szenario



2.4 Geschäftsprozessmodellierung

Problem

- Nicht immer lassen sich die Anwendungsfälle des zu entwickelnden Systems sofort bestimmen:
- Die organisatorischen Abläufe, die das System unterstützen soll,
 - sind komplex,
 - müssen neu organisiert werden
- Beispiele
 - Reisekostenabrechnung wird auf Internet umgestellt
 - Einreichen des Einkommenssteuerbescheids wird auf Internet umgestellt
 - Einführung eines automatischen Bestellsystems in einem Supermarkt

Lösung

- Können die Anwendungsfälle eines Systems nicht identifiziert werden, muss der Anforderungsspezifikation eine **Geschäftsprozessmodellierung** (GP-Modellierung) vorangehen
- Ziel der GP-Modellierung ist es, die organisatorischen Abläufe, in die das zu entwickelnde IT-System eingebettet werden soll, abzubilden und mit dem Kunden abzustimmen bzw. neu zu entwickeln

Ziele von GP-Modellierung im Allgemeinen

- Analyse bestehender Abläufe
 - Kosten, Dauer, benötigte Ressourcen
 - Verantwortlichkeiten
 - Abhängigkeiten
- Optimierung von Abläufen im Unternehmen
- Analyse des organisatorischen Umfelds von IT-Systemen

NB: Die GP-Modellierung, wie sie im Folgenden besprochen wird, orientiert sich nur an letzterem Ziel!

Definition Geschäftsprozess

Ein Geschäftsprozess fasst die zur Erfüllung einer Unternehmensaufgabe notwendigen Verrichtungen in ihrem sachlichen, räumlichen und zeitlichen Zusammenhang zusammen.

Beispiele:

- Einkommenssteuererklärung
- Mitarbeiter einstellen
- Abschluss einer Lebensversicherung
- Kreditnahme
- Dienstreise

Zerlegung von Geschäftsprozessen

- Arbeitsschritt
 - fachlich nicht unterbrechbar
 - wird im Rahmen der GP-Modellierung nicht weiter zerlegt
- Teilprozess
 - besteht aus mehreren Teilprozessen oder Arbeitsschritten
 - kann mehreren Geschäftsprozessen zugeordnet sein
- Beispiel: GP „Dienstreise“
 - Arbeitsschritte: „Dienstreiseantrag stellen“, Reiseantrag genehmigen“, „Reise organisieren“, „Abrechnung einreichen“
 - Teilprozesse: „Dienstreiseabrechnung bearbeiten“

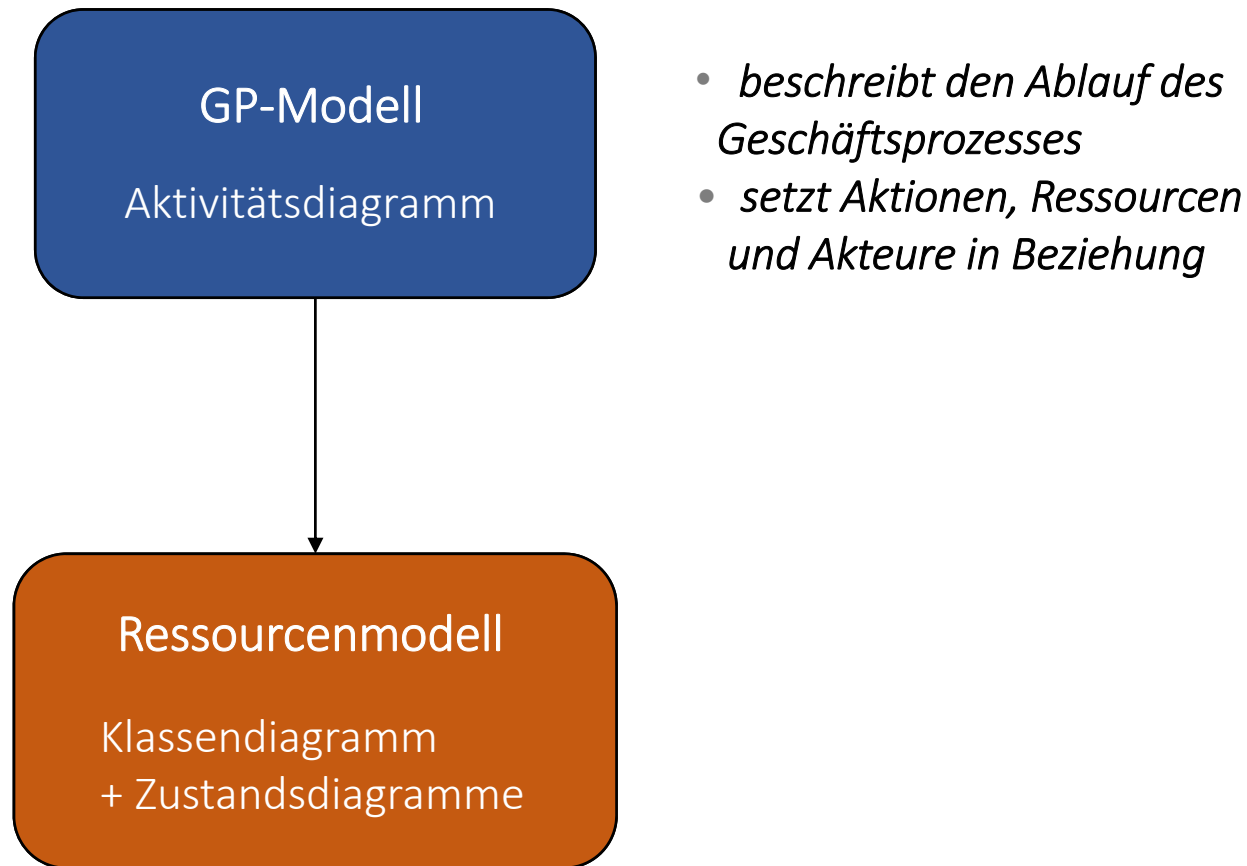
Grundkonzepte

- Ressourcen
 - alle von einem Geschäftsprozess benutzten, verbrauchten oder produzierten Objekte
 - z.B. Vertrag, Rechnung, Schraube, ...
- Akteure
 - alle an einem Geschäftsprozess beteiligten Personen mit ihren Rollen
 - z.B. Sachbearbeiter, Kreditnehmer, Kunde, ...
- Arbeitsschritte oder Aktionen
 - alle in einem Geschäftsprozess durchgeführten atomaren Tätigkeiten
 - z.B. Auftrag annehmen, Rechnung stellen, ...

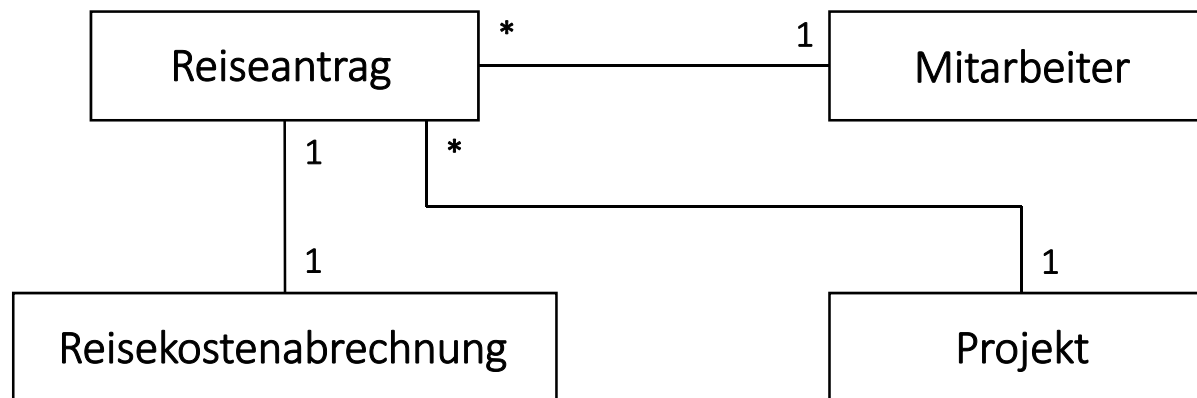
Beziehungen zwischen den Grundkonzepten

- Der Geschäftsprozess besteht aus einer Folge von Aktionen
 - Beschreibung dieser Folge von Aktionen in einem oder mehreren **Aktivitätsdiagrammen**
- Aktionen werden von einem oder mehreren Akteuren durchgeführt
- Ressourcen können für eine Aktion benötigt werden (die Ressource ist Input)
- Ressourcen können in einer Aktion erzeugt oder verändert werden (die Ressource ist Output)

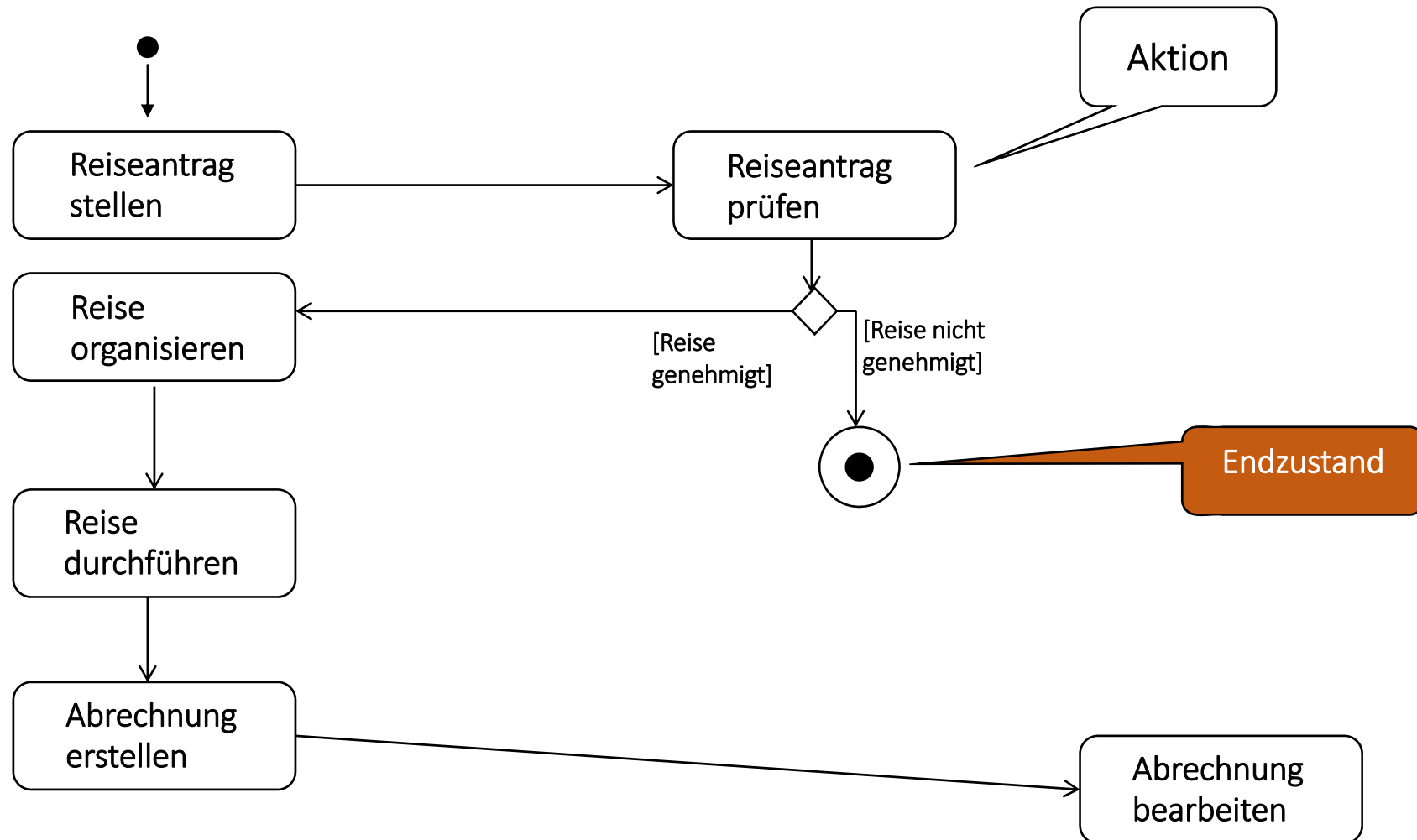
Geschäftsprozessmodell



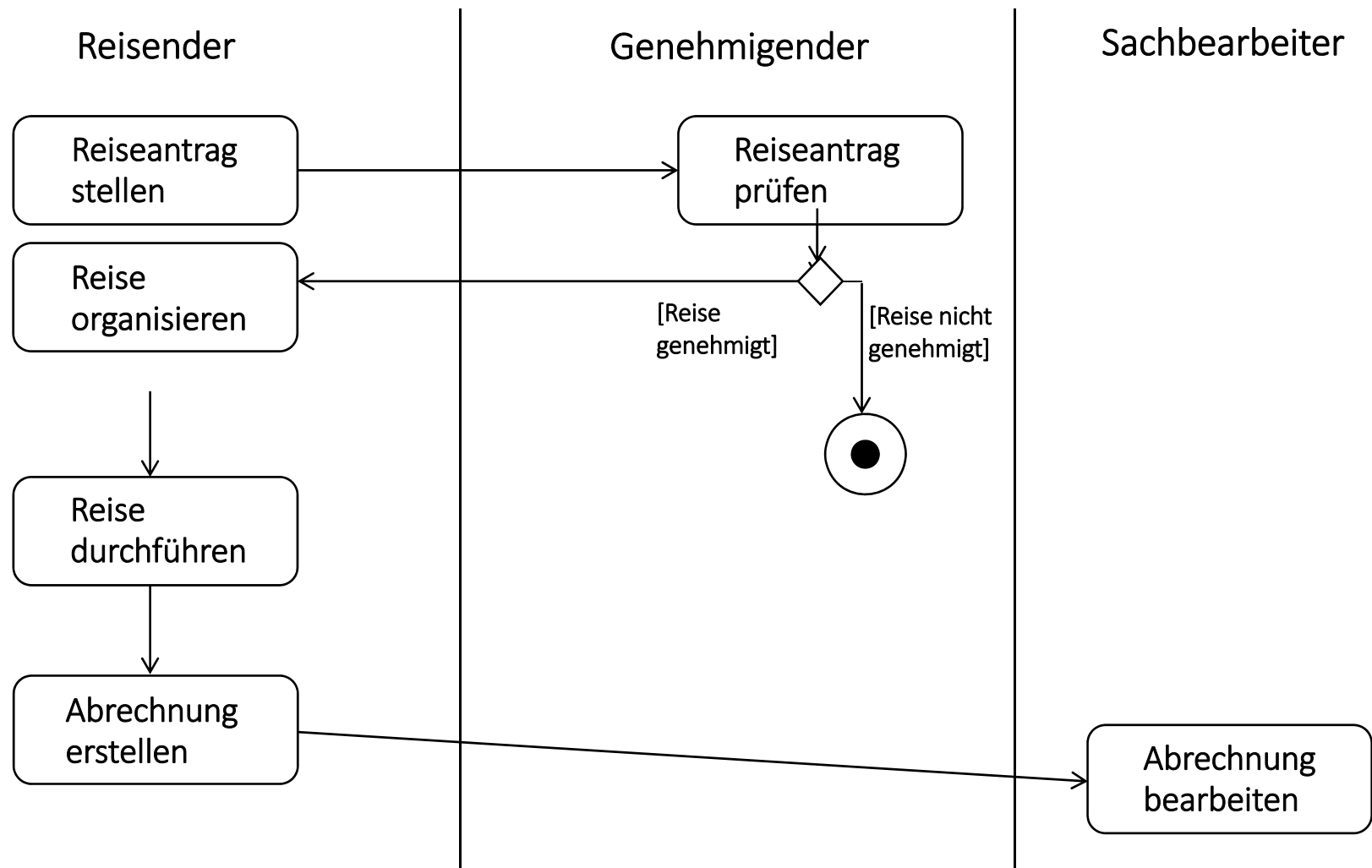
Beispiel - Ressourcenmodell



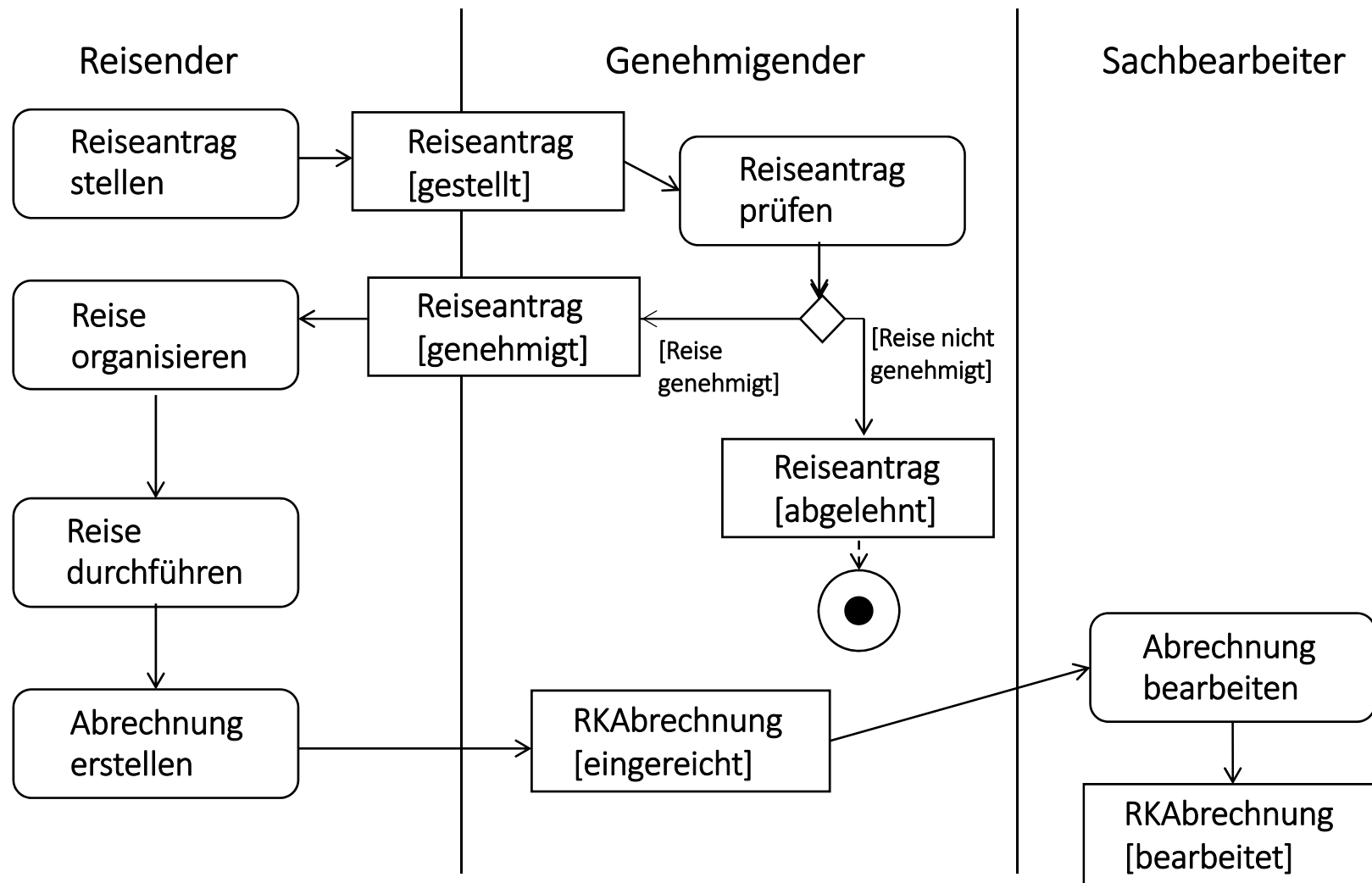
Beispiel - GP-Modell



Erweiterung 1: Zuordnung von Akteuren

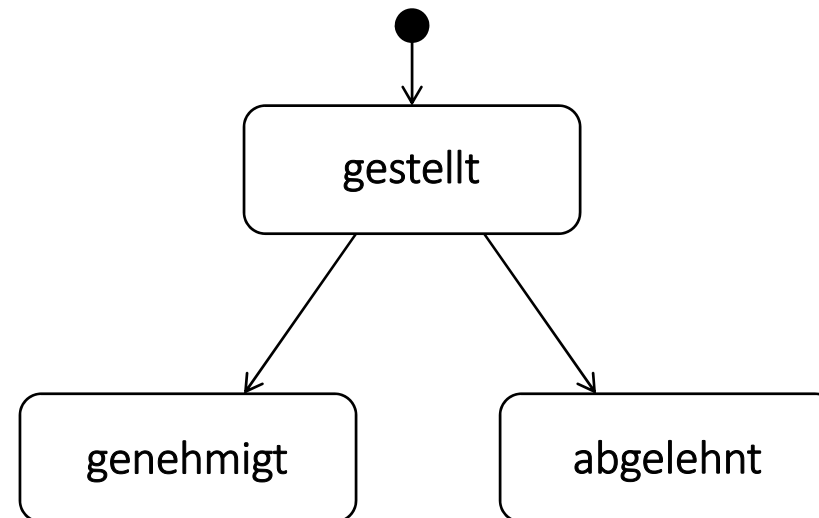


Erweiterung 2: Zuordnung von Ressourcenobjekten



Zugehöriges Zustandsdiagramm

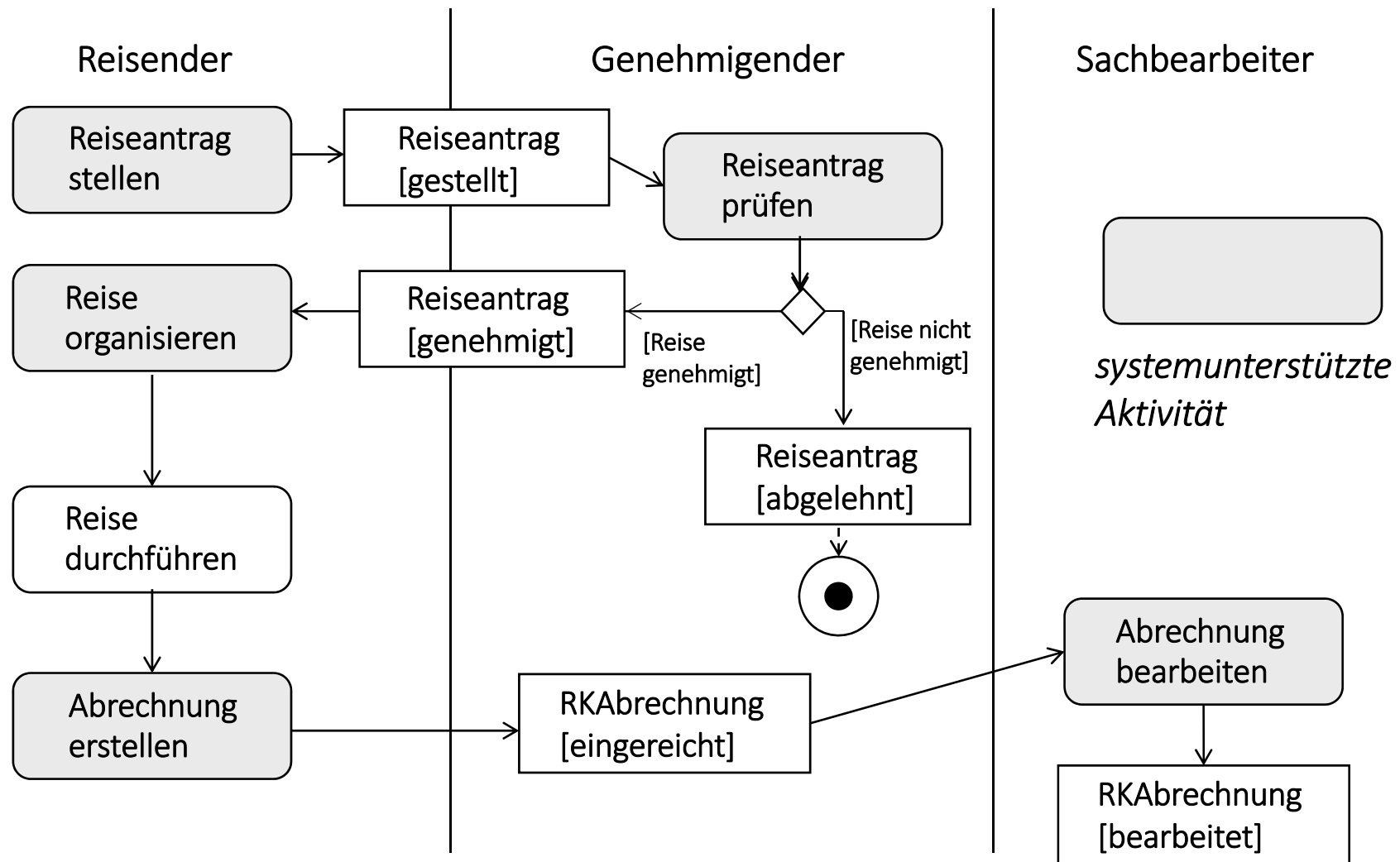
Zustandsdiagramm für Klasse Reiseantrag



Übergang von Geschäftsprozessen zu Anwendungsfällen

- **1. Schritt:** Identifizieren von systemunterstützten Arbeitsschritten
 - welche Arbeitsschritte in den Geschäftsprozessen werden vom zu entwickelnden System automatisch durchgeführt?
 - welche Arbeitsschritte werden von den Akteuren in Interaktion mit dem System durchgeführt?
 - welche Arbeitsschritte werden ohne Interaktion mit dem System bearbeitet?
- **2. Schritt:** Definition von Anwendungsfällen

Beispiel – 1. Schritt



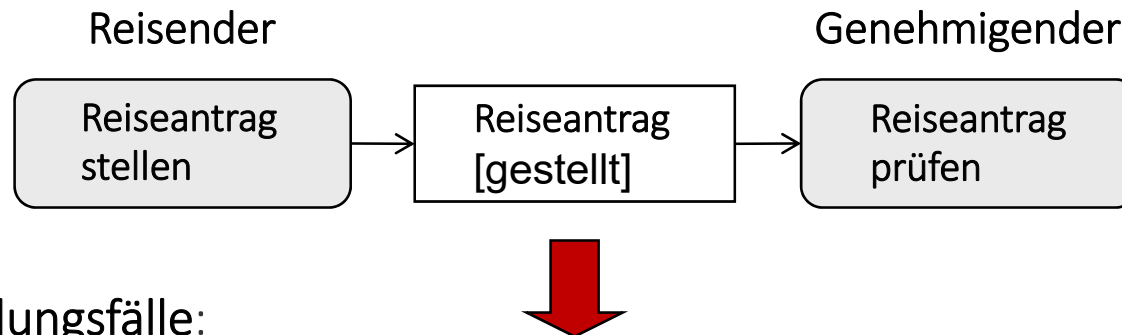
Definition von Anwendungsfällen

- Beim Übergang von Geschäftsprozessen zu Anwendungsfällen wird die Sichtweise gewechselt:
 - Geschäftsprozess: Wie ist der Ablauf im Unternehmen organisiert?
 - Anwendungsfall: Welchen Dienst bietet das IT-System an?

Hilfreiche Fragen bei der Definition von Anwendungsfällen

- Für jeden systemunterstützten Arbeitsschritt:
 - Durch welche Dienste des Systems wird dieser Arbeitsschritt unterstützt?
 - Welche Informationen muss der Akteur zur Ausführung des Arbeitsschritts zur Verfügung haben und wie werden diese vom System zur Verfügung gestellt?
 - Wie fließen die Informationen zwischen den Akteuren (als Papier, in elektronischer Form?) und wie wird dies durch das System unterstützt?
 - Wie werden die Anwendungsfälle gestartet?

Beispiel



Anwendungsfälle:

- *Reiseantrag stellen*
 - Reisender füllt ein Formular im Intranet aus und schickt es ab
- *Reiseantrag prüfen*
 - Vorgesetzter genehmigt oder lehnt Antrag in einem Formular im Intranet ab; der Reisende erhält als Bestätigung eine E-Mail
- *Anzeige Projektbudget*
 - Der Genehmigende kann sich die Projektbudgets anzeigen lassen
- *Anzeige Dienstreisen*
 - Der Genehmigende kann sich frühere Dienstreisen von Mitarbeitern anzeigen lassen

Granularität von Arbeitsschritten und Anwendungsfällen

- Wichtig ist, die richtige Granularität von Arbeitsschritten und Anwendungsfällen zu finden
 - Arbeitsschritte und Anwendungsfälle sollten ungefähr die gleiche Granularität haben
 - Geschäftsprozessmodellierung nicht zu detailliert gestalten
 - nur Geschäftsprozesse „rund um das IT-System“ modellieren
 - nur komplexe Geschäftsprozesse modellieren

Zuordnung von Anwendungsfällen zu Arbeitsschritten (1)

- Ein Arbeitsschritt kann durch mehrere Anwendungsfälle unterstützt werden
 - jeder systemunterstützte Arbeitsschritt wird durch mindestens einen Anwendungsfall unterstützt
- Ein Anwendungsfall kann mehreren Arbeitsschritten zugeordnet werden
 - Beispiel: Anzeige Projektbudget unterstützt Arbeitsschritt „Reiseantrag genehmigen“ und „Abrechnung bearbeiten“
- Es gibt Arbeitsschritte, die keinen Anwendungsfall zugeordnet haben
 - manuelle Arbeitsschritte, z.B. „Reise“

Zuordnung von Anwendungsfällen zu Arbeitsschritten (2)

- Es gibt Anwendungsfälle, die keinem Arbeitsschritt zugeordnet werden können
 - Anwendungsfälle, die die Wartung oder Administration des Systems zum Inhalt haben
 - Anwendungsfälle im Rahmen von Geschäftsprozessen, die nicht modelliert werden
 - Anwendungsfälle, die aus Gründen der Zugriffssicherheit des Systems definiert werden (Beispiel: „Anmelden im System“)

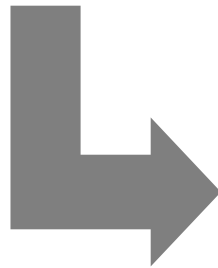
Übersicht

GP-Modell

Aktivitätsdiagramm

fachliches Klassendiagramm

(Ressourcenmodell)



Pflichtenheft

Anwendungsfalldiagramm

fachliches Klassendiagramm

Anwendungsfallbeschreibungen

Oberflächenprototyp

Weitere Ansätze zur GP-Modellierung

- Business Process Modeling Notation (BPMN)
 - Standard der OMG
 - www.bpmn.org
 - Gute Toolunterstützung
- ARIS-Ansatz
 - Anbindung an SAP
 - Konzipiert v.a. für betriebswirtschaftliche Anwendungen (Analyse und Konzeption betrieblicher Abläufe)
 - funktionale Systemsicht
 - ähnliche Grundkonzepte, andere Notationen
 - reichhaltige Syntax
 - Gute Toolunterstützung