

# Betriebliche Informationssysteme

## L6 – Requirements Engineering



# L6 – Requirements Engineering

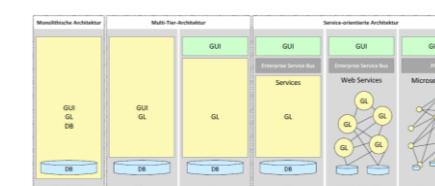
**„Wie kann ich Anforderungen von User festhalten und dokumentieren?“**

- Anforderungen *an das System* im Detail verstehen (vgl. Business Case: generelle Bedürfnisse verstehen)
- Den Begriff „Requirements Engineering“ verstehen
- Die Schritte des Requirements Engineering kennen
- Textuell und modellbasiert Anforderung beschreiben können
- *Weiterführende Literatur*
  - Bühne, Herrmann 2015, *Handbuch Requirements Management nach IREB Standard*
  - Herrmann, Andrea. "Grundlagen der Anforderungsanalyse." (2022).  
<https://link.springer.com/book/10.1007/978-3-658-35460-2>
  - Pohl, Klaus; Rupp, Chris (2021); *Basiswissen Requirements Engineering*.

# Zusammenhang mit Unternehmenskontext & Systemarchitektur

Nachdem wir

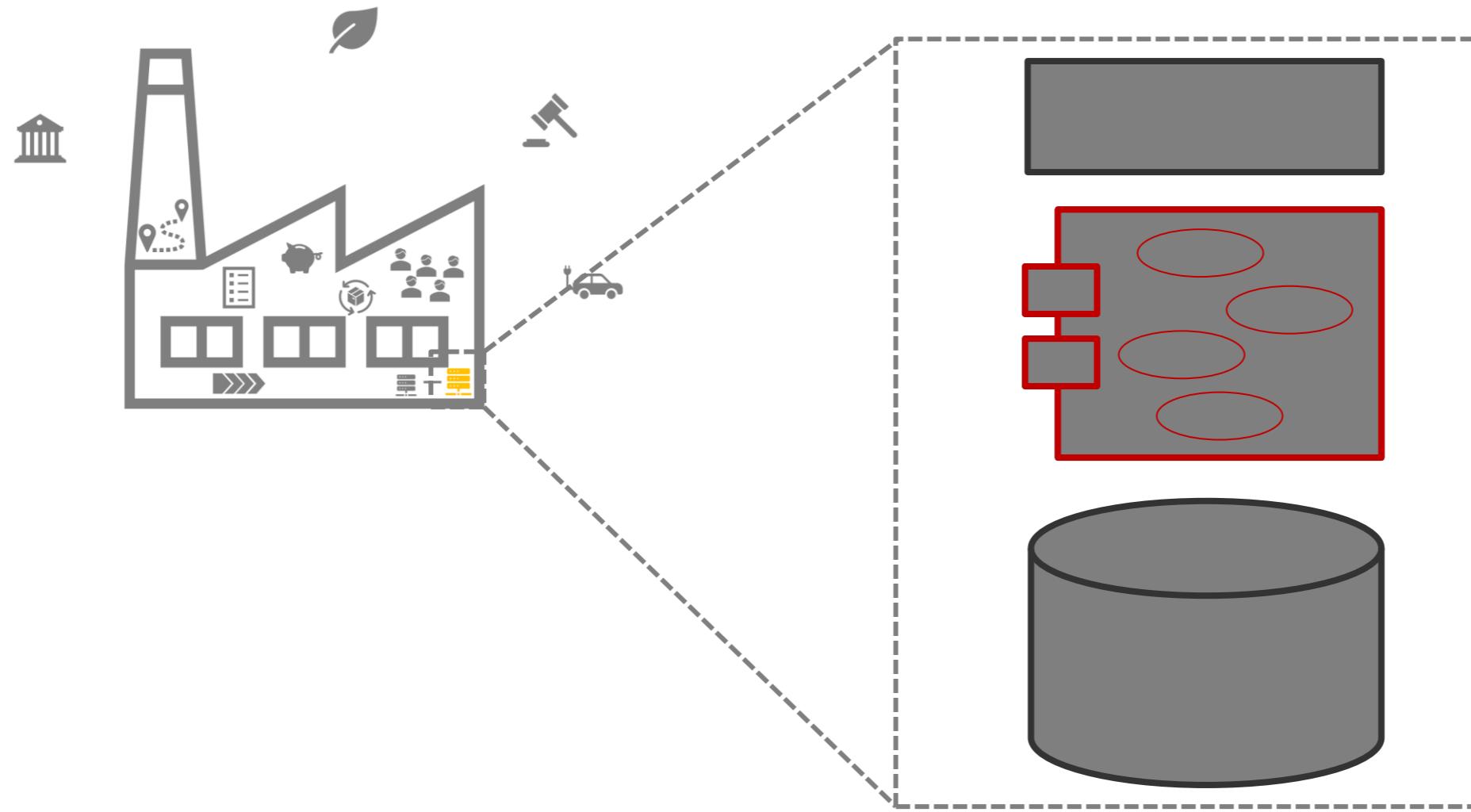
- die grundlegenden Bedürfnisse an das System (welche Funktionalitäten es haben soll) verstanden haben,
- die grundlegenden Vorgaben bekannt sind
- und wir die gewünschte Systemarchitektur gewählt haben,



können wir die detaillierten Anforderungen an das System festhalten.



# Big Picture



# Was ist Requirements Engineering?



Quelle: Dall-E. Prompt «Grafik zu Requirements Engineering».  
Zugriff 30.01.2025

# Definition Requirements Engineering

- *Requirements Engineering* ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen mit den folgenden Zielen:
  1. Die relevanten Anforderungen zu kennen, Konsens unter den Stakeholdern über die Anforderungen herzustellen, die Anforderungen konform zu vorgegebenen Standards zu dokumentieren und die Anforderungen systematisch zu managen.
  2. Die Wünsche und Bedürfnisse der Stakeholder zu verstehen und zu dokumentieren.
  3. Die Anforderungen zu spezifizieren und zu managen, um das Risiko zu minimieren, dass das System nicht den Wünschen und Bedürfnissen der Stakeholder entspricht [Glin2014] und [PoRu2011], Kap. 1.1.2.
- *Requirements Management*: Der Prozess des Verwaltens existierender Anforderungen und anforderungsbezogener Artefakte. Dies schließt insbesondere die Ablage, das Ändern und die Verfolgbarkeit von Anforderungen mit ein [Glin2014]. Auch das Verwalten des RE-Prozesses gehört dazu, d.h. das Planen, Steuern und Kontrollieren des RE-Prozesses.

Synonyme: Anforderungsanalyse, Anforderungsspezifikation

# Funktionale & Nicht-funktionale Anforderungen

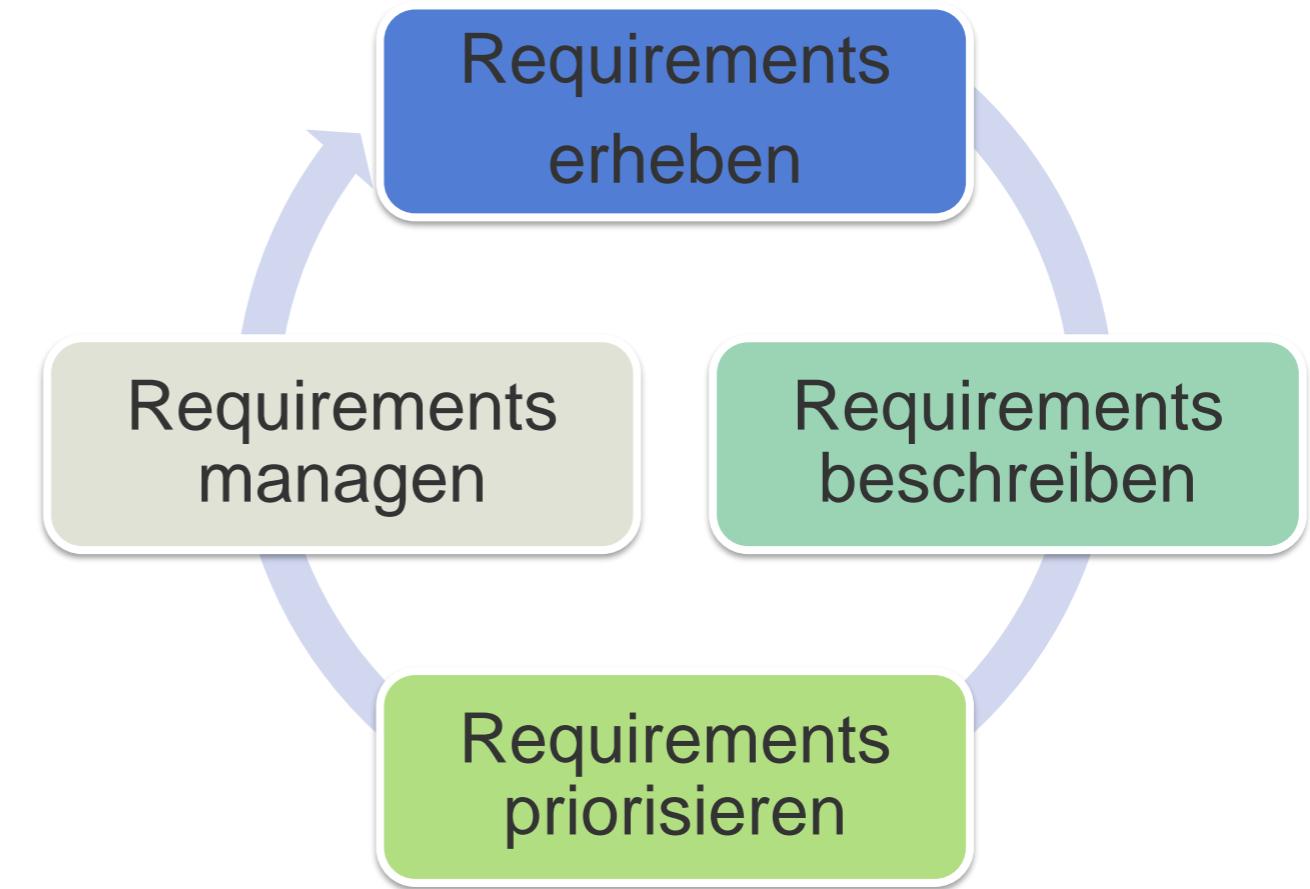
**Funktionale Anforderungen** beschreiben die Funktionalität, die das geplante System bereitstellen soll. *Beispiele:*

- Benutzeranmeldung und -authentifizierung: Das System sollte Benutzern ermöglichen, sich anzumelden und ihre Identität zu überprüfen.
- Datenverarbeitung und -speicherung: Das System muss in der Lage sein, Daten zu speichern, zu verarbeiten und abzurufen.
- Benutzeroberfläche (UI): Das System sollte eine benutzerfreundliche Oberfläche bieten, die es den Benutzern ermöglicht, effizient zu interagieren.
- Berechtigungsverwaltung: Das System muss Zugriffsrechte und Rollen für Benutzer definieren und verwalten können.
- Berichterstattung und Analyse: Das System sollte Berichte generieren und Daten analysieren können.

**Nicht-funktionale Anforderungen** betreffen die Qualitätseigenschaften des Systems. *Beispiele:*

- Leistung (Performance): Das System sollte eine akzeptable Antwortzeit haben, um Benutzerfrustration zu minimieren.
- Sicherheit: Das System muss sicher sein, um unbefugten Zugriff oder Datenverlust zu verhindern.
- Skalierbarkeit: Das System sollte in der Lage sein, mit wachsenden Benutzerzahlen umzugehen.
- Benutzerfreundlichkeit (Usability): Die Benutzeroberfläche sollte intuitiv und leicht verständlich sein.
- Verfügbarkeit: Das System sollte rund um die Uhr verfügbar sein, um Unterbrechungen zu minimieren.

# Der Requirements Engineering Prozess



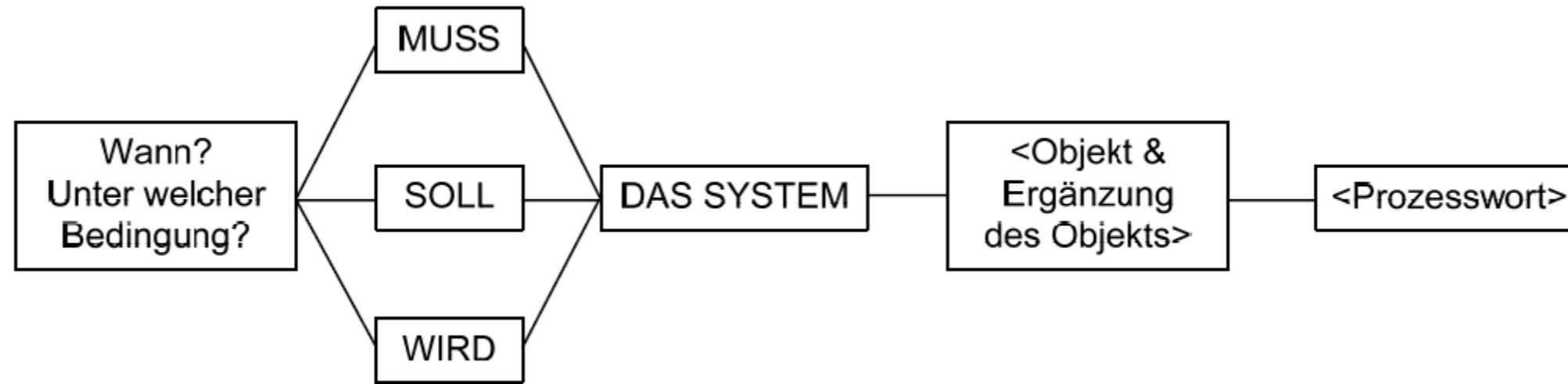
Quelle: Angelehnt an Bühne, Herrmann 2015, Handbuch Requirements Management nach IREB Standard, Kap. 9

# Typische Erhebungstechniken

Requirements  
erheben

Erhebungstechnik	Vorteile	Nachteile
Dokumentenstudium	<ul style="list-style-type: none"><li>• Schnell</li><li>• kostengünstig</li><li>• formal</li></ul>	<ul style="list-style-type: none"><li>• Veraltete Infos</li><li>• fehlender Stakeholder-Austausch</li></ul>
Interviews	<ul style="list-style-type: none"><li>• Detaillierte Infos</li><li>• Verständnis überprüfbar</li></ul>	<ul style="list-style-type: none"><li>• Zeitaufwendig</li><li>• Beschränkte Zahl von Teilnehmern: subjektive Verzerrung</li></ul>
Fragebogen	<ul style="list-style-type: none"><li>• (Theoretisch) grosse Teilnehmerzahl erreichbar</li><li>• (Tendenziell) standardisierte Daten</li></ul>	<ul style="list-style-type: none"><li>• Keine Klärungsmöglichkeit</li><li>• Häufig schlechter Rücklauf</li><li>• Ungeeignet für offene Fragen</li><li>• Erstellung braucht grosses Vorwissen</li></ul>
Beobachtung	<ul style="list-style-type: none"><li>• Zeigt aktuelle Situation</li><li>• keine Verzerrung durch Erinnerung</li><li>• Auch Selbstverständliches wird festgehalten</li></ul>	<ul style="list-style-type: none"><li>• Hoher Zeitaufwand</li><li>• (Ggf.) beeinflusstes Verhalten</li></ul>
Workshops / Gruppenexploration	<ul style="list-style-type: none"><li>• Grössere Zahl von Teilnehmenden</li><li>• Stakeholder-Austausch</li></ul>	<ul style="list-style-type: none"><li>• Hoher Aufwand,</li><li>• Anspruchsvolle Moderation</li><li>• Domination einzelner Teilnehmenden</li></ul>

# Satzschablonen



## Beispiele

- Bei der Tagesendverarbeitung muss das System (die Applikation, etc.) eine Fahrzeugreservationsliste drucken.
- Wenn ein Benutzer bei einer Reservation einen Kunden auswählt, soll das Fahrzeugvermietungs-system dem Benutzer Namen, Adresse und Kontostand des Kunden anzeigen.

# User Stories

Eine **User Story** beschreibt eine Anforderung mit einem definierten Satzbau. Eine User Story hat i.d.R. die folgende Form:

Als <ROLLE>  
möchte ich <FUNKTIONALITÄT>,  
sodass <NUTZEN>.

Beispielsweise:

Als Kunde  
möchte ich Geld von meinem Konto auf ein fremdes Konto überweisen,  
um meine Rechnungen möglichst bald zu bezahlen.

Für jede User Story können auch **Akzeptanzkriterien und Testfälle** genauer spezifiziert werden, i.d.R. in folgender Form:

Unter Voraussetzung dass <Vorbedingung>  
wenn <Trigger>  
dann <Ergebnis>.

Beispielsweise:

Unter der Voraussetzung, dass auf meinem Konto mehr als 100 CHF vorhanden sind,  
wenn ich eine Überweisung über 100 CHF tätige,  
dann werden anschliessend auf meinem Konto 100 CHF weniger angezeigt und auf dem Zielkonto 100 € mehr als zuvor.



# Übung: User Stories

**Es wurde eine Interview durchgeführt, um die Anforderungen an ein generatives KI-Tool zu verstehen.**

Übung User Story Interview\_Transkription.docx

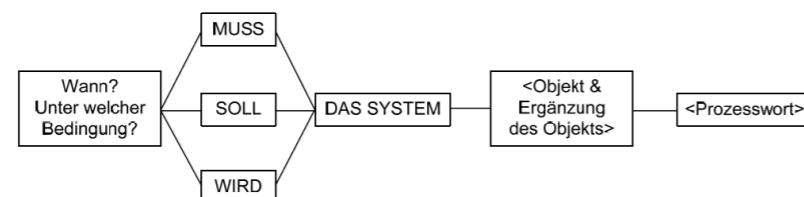
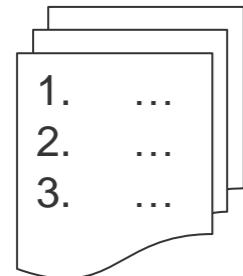
- **Lest die Interview Transkription auf moodle durch.**
- **Leitet davon 4 User Stories ab.**
- **Davon 2 User Stories, welche direkt die interviewte Person betreffen & 2 User Stories, welche andere Personen (User bzw. Rollen) betreffen.**

**2-3er Gruppen / 20 min**

# Anforderungen festhalten

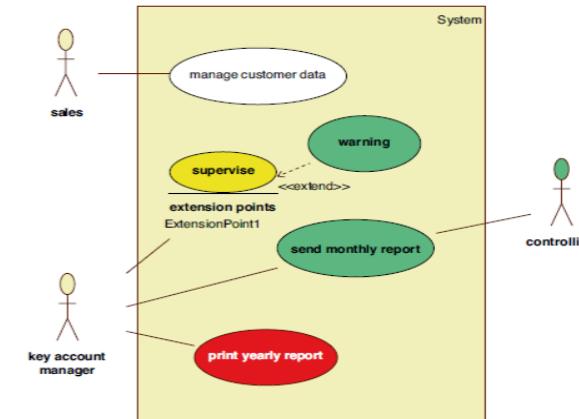
Requirements  
beschreiben

Textuelle Darstellung



Quelle: Pohl, Klaus; Rupp, Chris (2021); Basiswissen Requirements Engineering.

Modellbasierte Darstellung



bb. 5.10 Delta-Anforderungen: Use-Case-Diagramm mit Fischaugensicht und Farbcodierung

Quelle: Herrmann, Andrea. "Grundlagen der Anforderungsanalyse." (2022).

# Modellbasierte Darstellung

## Kontextdiagramm

- Wenn du das System und seine Schnittstellen grob definieren willst.
- Zweck: Zeigt das System als eine Black Box und stellt dar, wie es mit externen Entitäten interagiert.
- Fokus: Abgrenzung des Systems und dessen Umgebung.

## Use Case Diagramm

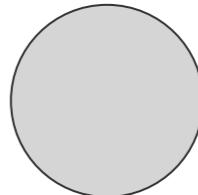
- Wenn du wissen willst, welche Funktionen das System bietet.
- Zweck: Beschreibt die Funktionalität des Systems aus der Sicht der Benutzer.
- Fokus: Welche Anwendungsfälle (Use Cases) ein Akteur mit dem System durchführen kann.

## Aktivitätsdiagramm

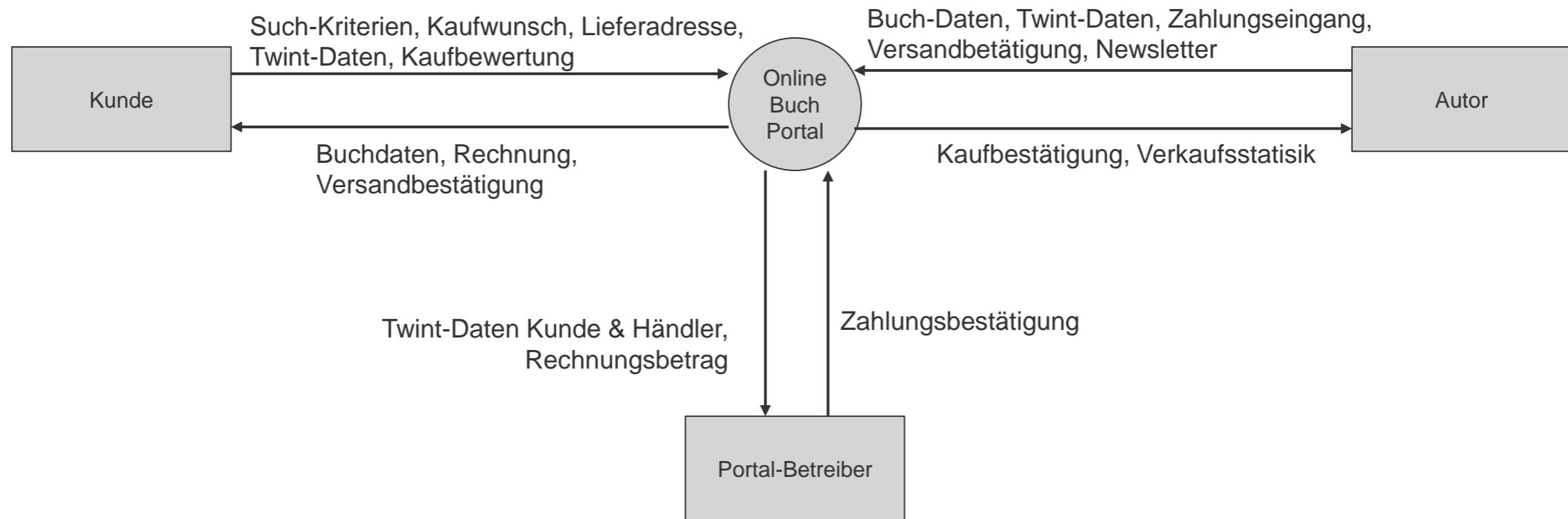
- Wenn du die genauen Abläufe und Prozessschritte darstellen willst
- Zweck: Zeigt den detaillierten Ablauf einer Aktivität oder eines Geschäftsprozesses.
- Fokus: Ablauflogik und Workflows.

# Kontextdiagramm

# Erläuterung Kontextdiagramm

- ✓ Die Verwendung erfolgt **in der frühen Entwurfs- oder Analysephase**.
- ✓ Beim Kontextdiagramm handelt es sich um ein **abstraktes Datenflussdiagramm**, dessen Zweck in der **Abbildung der Schnittstellen des Systems zu dessen Umwelt (Kontext) besteht**.
- ✓ Das Diagramm besteht aus **genau einem Prozess, der als Kreis dargestellt wird**.
- ✓ **Komponenten**, die mit dem System interagieren, werden als Rechtecke dargestellt.
- ✓ **Die Datenflüsse** werden durch Pfeile vom oder zum Prozess repräsentiert.

# Beispiel: Buchportal zum Kauf & Verkauf von Büchern



Quelle: Basierend auf Herrmann, Andrea. "Grundlagen der Anforderungsanalyse." (2022).



# Übung: Erstellung eines Kontextdiagramms

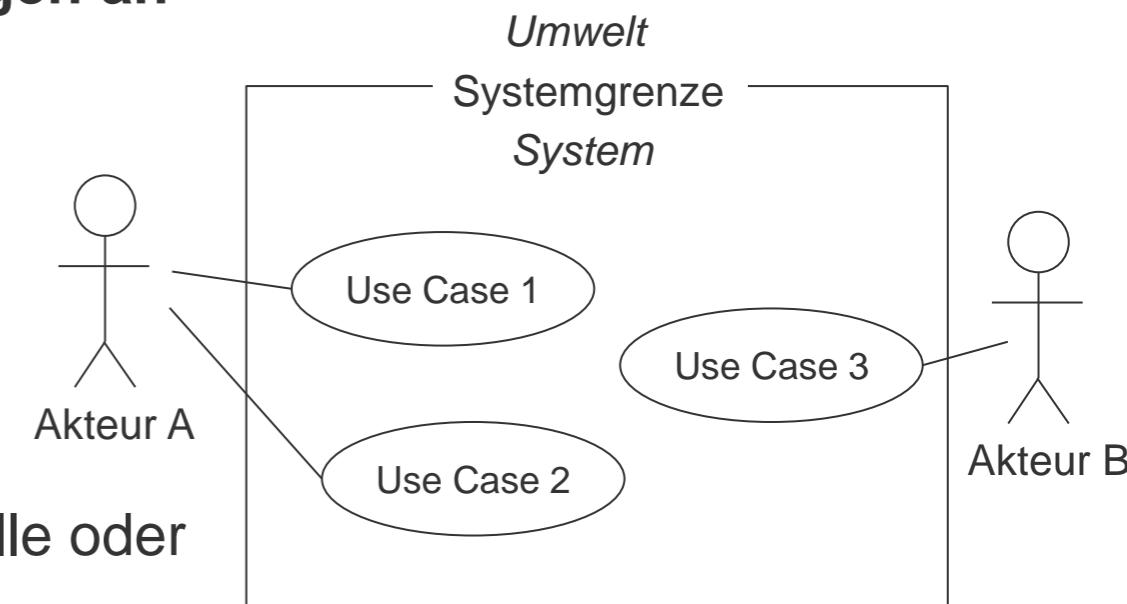
**Erstellt ein Kontextdiagramm für MS Teams Telefonielösung.**

**2-3er Gruppen / 20 min**

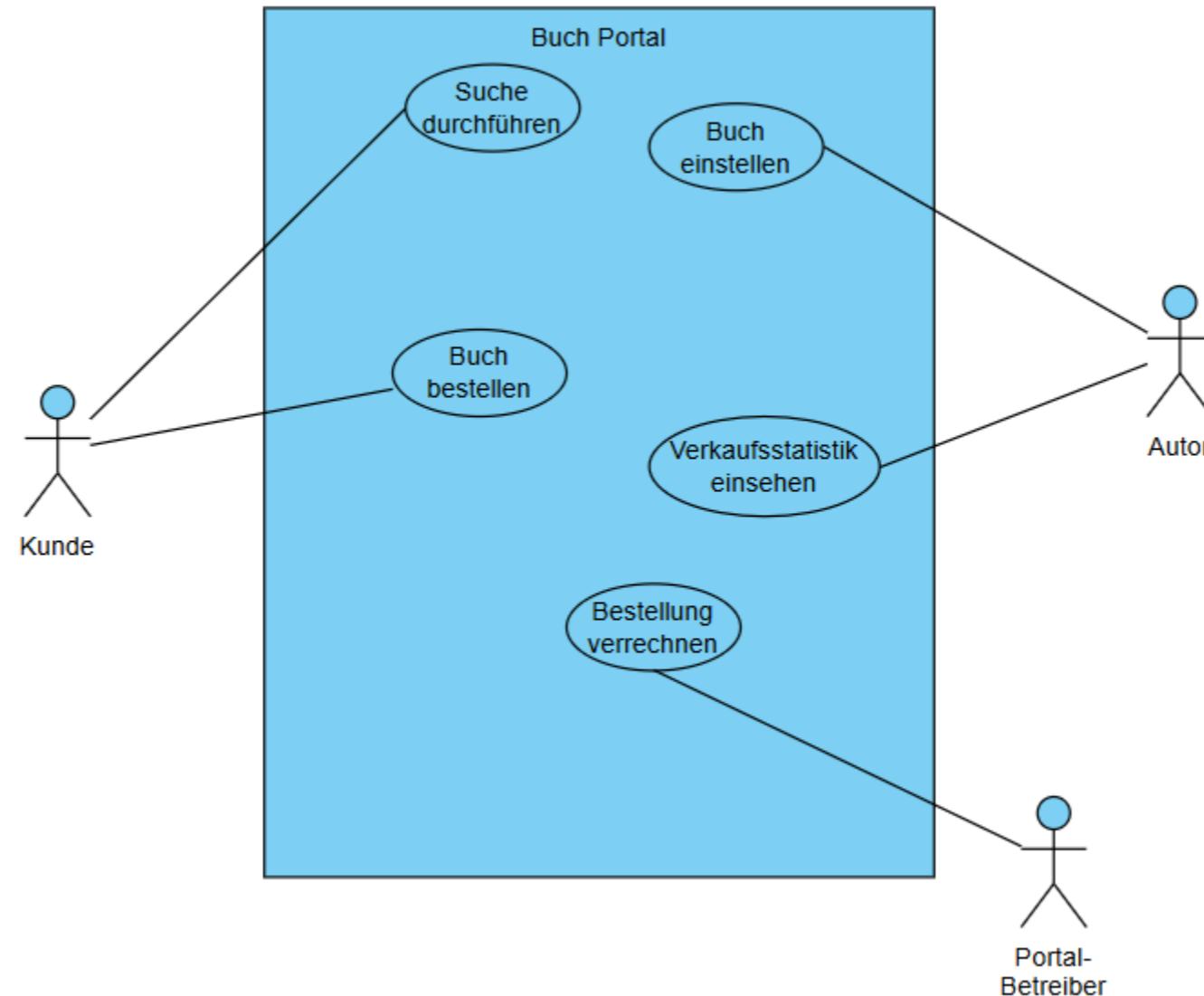
# UML Use Case Diagramm

# Erläuterung Use Case Diagramm

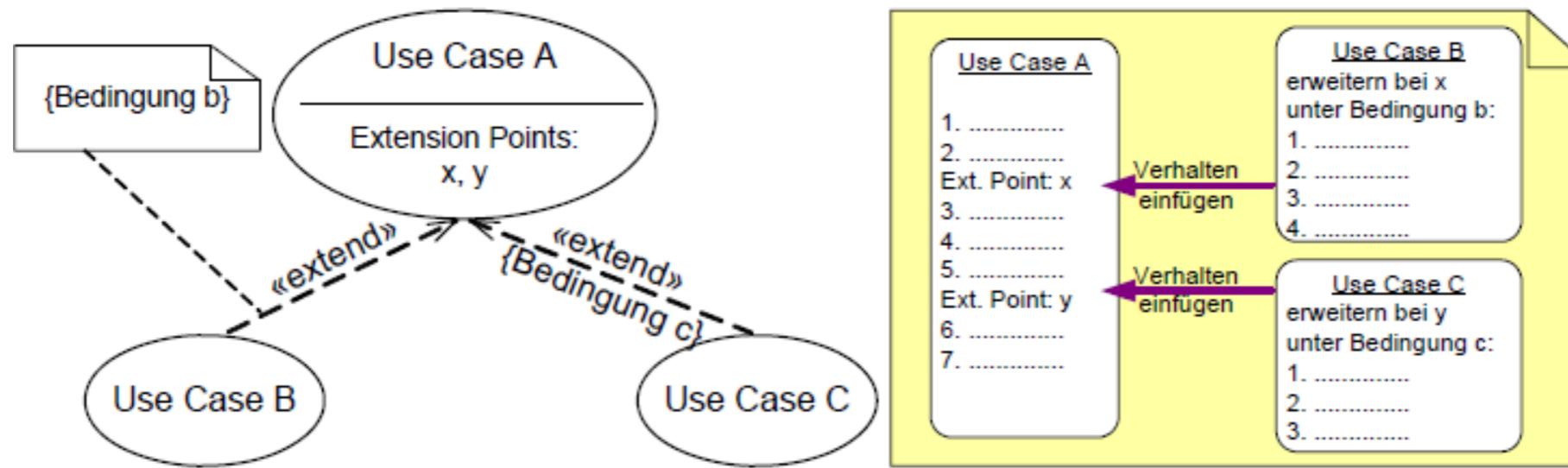
- ✓ Methode zur Definition der **funktionalen Anforderungen an ein System**
- ✓ **Zielsetzungen sind hauptsächlich:**
  - ✓ Eine klare System-Abgrenzung zu erzielen sowie
  - ✓ alle System-Funktionalität erfassen zu können.
  - ✓ Die Hauptaufgaben der einzelnen Akteure festzuhalten.
- ✓ **Der Akteur** verkörpert eine exemplarische Benutzerrolle oder -Kategorie. Dies sind Personen oder Systeme.
- ✓ Ein **Use Case** beschreibt eine typische Form der Anwendung eines IT-Systems zur Erfüllung einer bestimmten Aufgabe & hält Interaktionen zwischen einem Actor System fest



# Beispiel: Online Buch Portal



# Use Case-Erweiterung (extend)

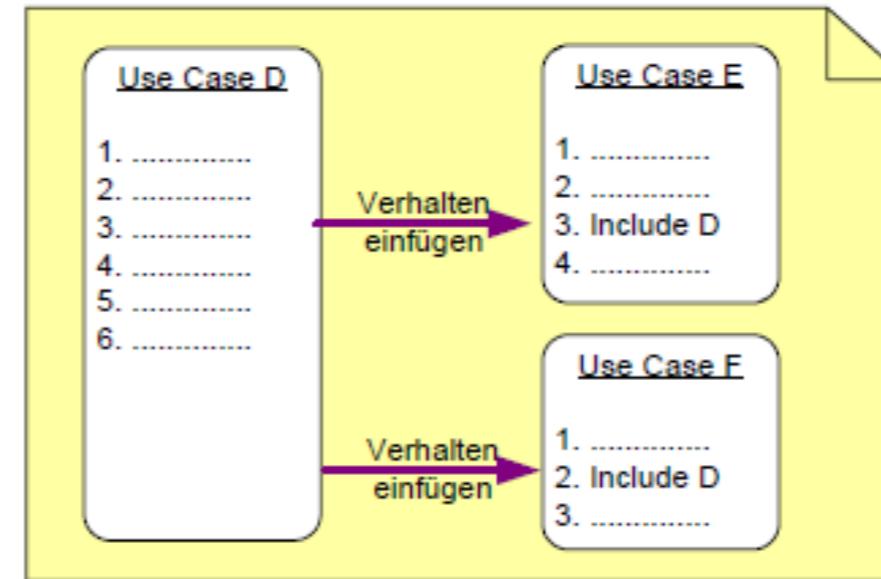
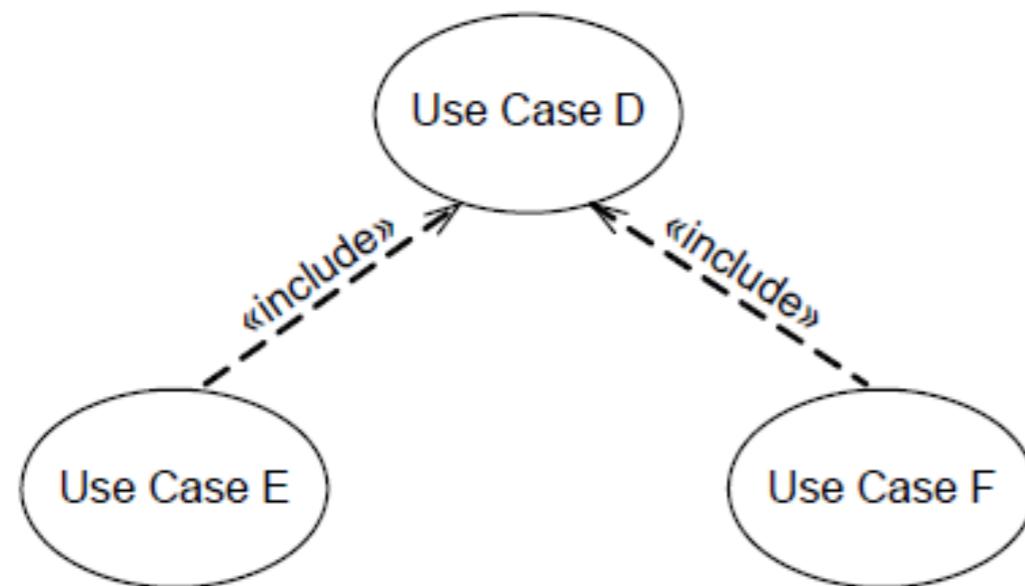


- > Ein Use Case B kann den Use Case A erweitern, aber nur unter bestimmten Bedingungen.
- > Die Erweiterung tritt optional und nur bei bestimmten Szenarien ein.

- ✓ Der Basis-Use-Case (A) kann ohne den erweiterten Use-Case (B) existieren.
- ✓ Der erweiternde Use-Case (B) tritt nur in bestimmten Fällen auf

Quelle: Fuchs, E., Fuchs, K. H., & Hauri, C. H. (2013). Requirements-Engineering in IT effizient und verständlich

# Use Case-Inkorporation (include)

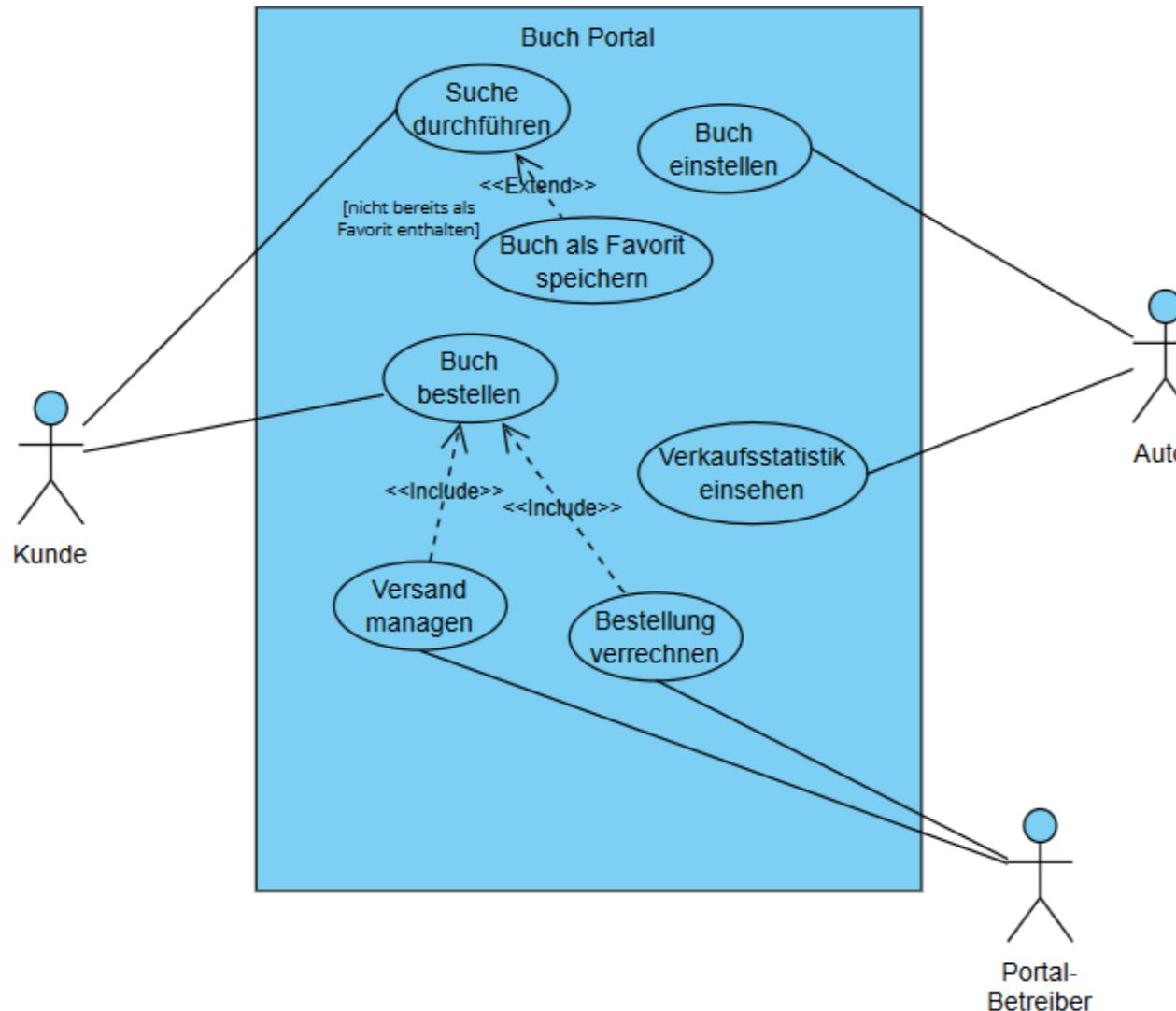


- > Ein Use Case E ruft immer Use Case D auf, weil D eine notwendige Teilfunktion ist.
- > D ist kein optionaler Schritt, sondern wird immer ausgeführt.

- ✓ Der inkludierte Use-Case (D) ist **zwingend erforderlich**.
- ✓ Er wird **mehrfach wiederverwendet** in verschiedenen Haupt-Use-Cases.

Quelle: Fuchs, E., Fuchs, K. H., & Hauri, C. H. (2013). Requirements-Engineering in IT effizient und verständlich

# Beispiel: Online Buch Portal - Erweitert



Alles korrekt?

# Misuse Case – Weitere Anwendungsmöglichkeit

- Am Beispiel Buch Portal
- «**threaten**»: Möglicher Fehler  
Use Cases die auftreten können  
festhalten (schwarz eingefärbt)
- «**mitigate**»:  
**Gegenmassnahmen** (weiss  
eingefärbt)

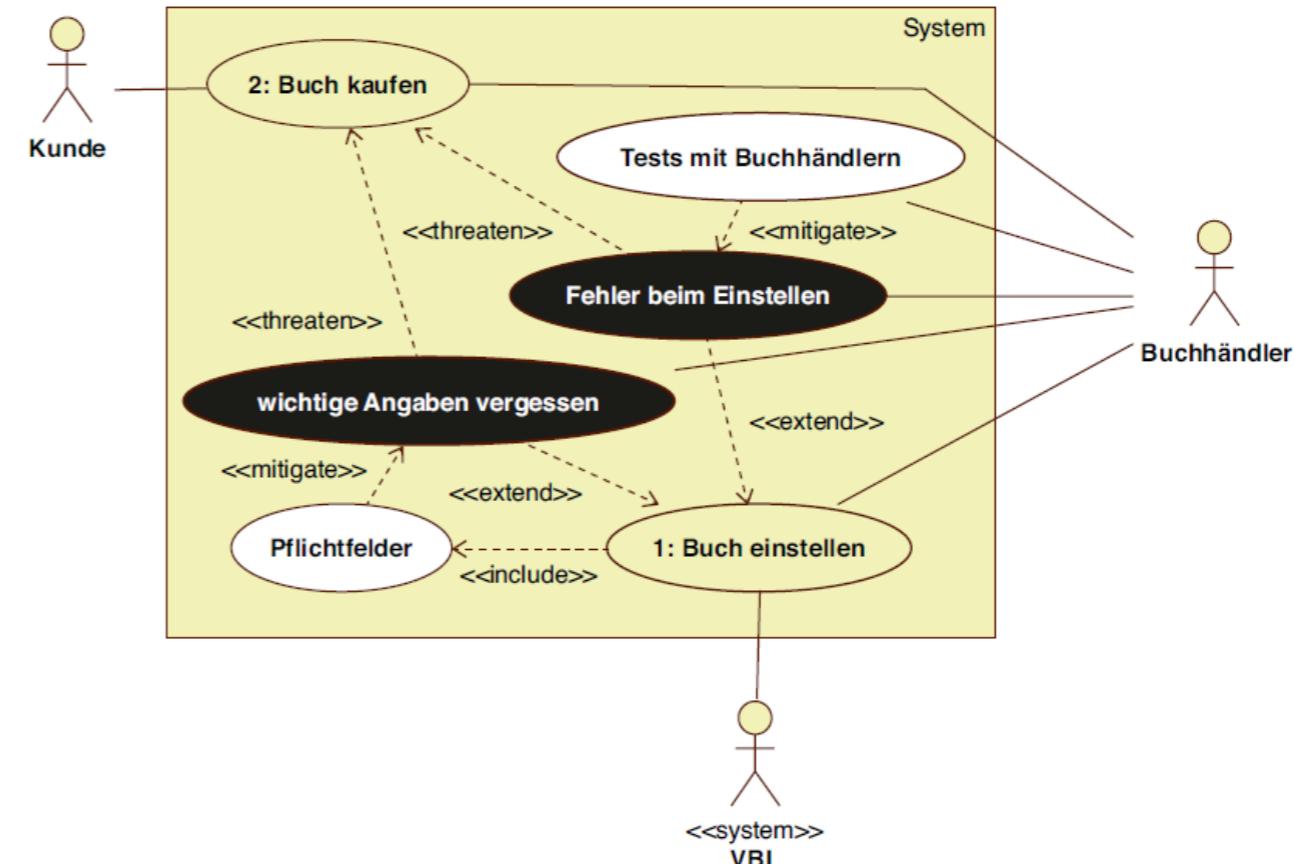


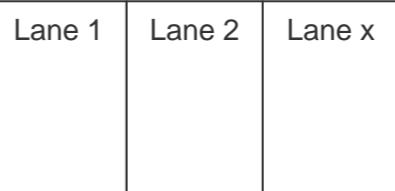
Abb. 4.3 Misuse Cases für das Buchportal

Quelle: Herrmann, Andrea. "Grundlagen der Anforderungsanalyse." (2022).

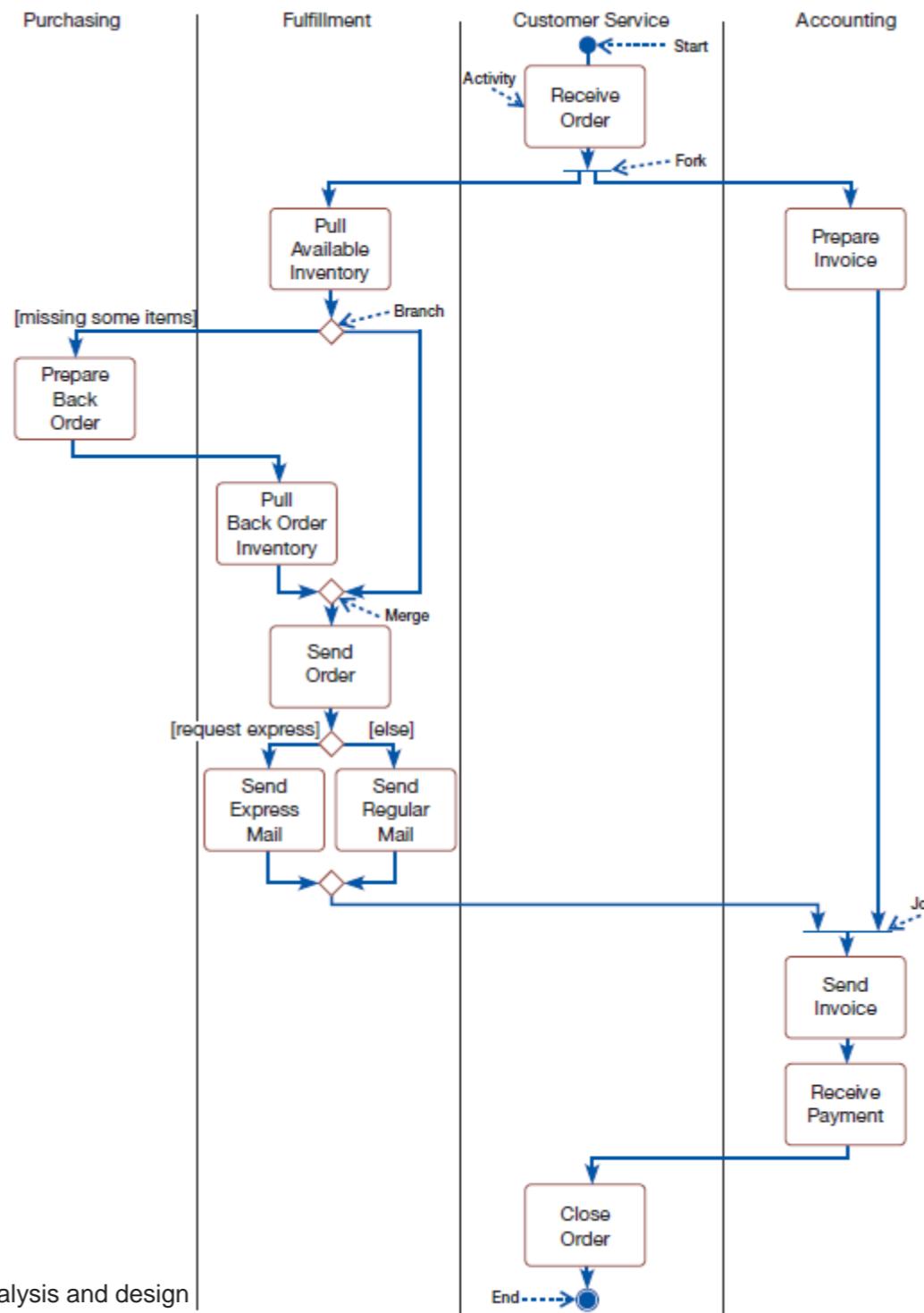
# UML Aktivitätsdiagramm

Quelle: Dall-E. Prompt «Grafik für Material Requirements Planning (MRP) ». Zugriff 16.01.2025

# Erläuterung Aktivitätsdiagramm

- ✓ Methode zur Spezifikation von **Abläufen**, bzw. detaillierten Ablaufbeschreibung von Use Cases. Sie haben eine Ähnlichkeit zu BPMN Modellen.
  
- ✓ Die wichtigsten **Elemente** sind
  - ✓ Die eigentliche Aktivität 
  - ✓ Start und Ende einer Aktivität  
  - ✓ Die Entscheidung (inkl. möglicher Bedingungen)  [Bedingung] und Zusammenführung
  - ✓ Die Parallelisierung (fork) und Synchronisation (join)  
  - ✓ Schwimmbahnen zur Darstellung von Zuständigkeiten 
  - ✓ Der Kontrollfluss 

# Beispiel: Bestellprozess



Quelle: Valacich, J. S., George, J. F., & Valacich, J. S. (2021). Modern systems analysis and design



# Übung: Aktivitätsdiagramm

2-3er Gruppen / 20 min

Erstellt ein Aktivitätsdiagramm für folgende Beschreibung:

Der Kühlschrank überprüft regelmässig die aktuellen Vorräte und analysiert dabei insbesondere, welche Lebensmittel sich dem Verfallsdatum nähern. Zusätzlich werden hinterlegte Allergieinformationen des Nutzers berücksichtigt, um potenziell unverträgliche Zutaten auszuschliessen. In der Folge wird ein Online-Rezeptservice aufgerufen, um passende Rezepte zur Verwertung der bald verderblichen, aber unbedenklichen Artikel zu erhalten. Die ausgewählten Rezeptvorschläge werden anschliessend in einer Nachricht zusammengefasst und auf das Handy des Nutzers gesendet.

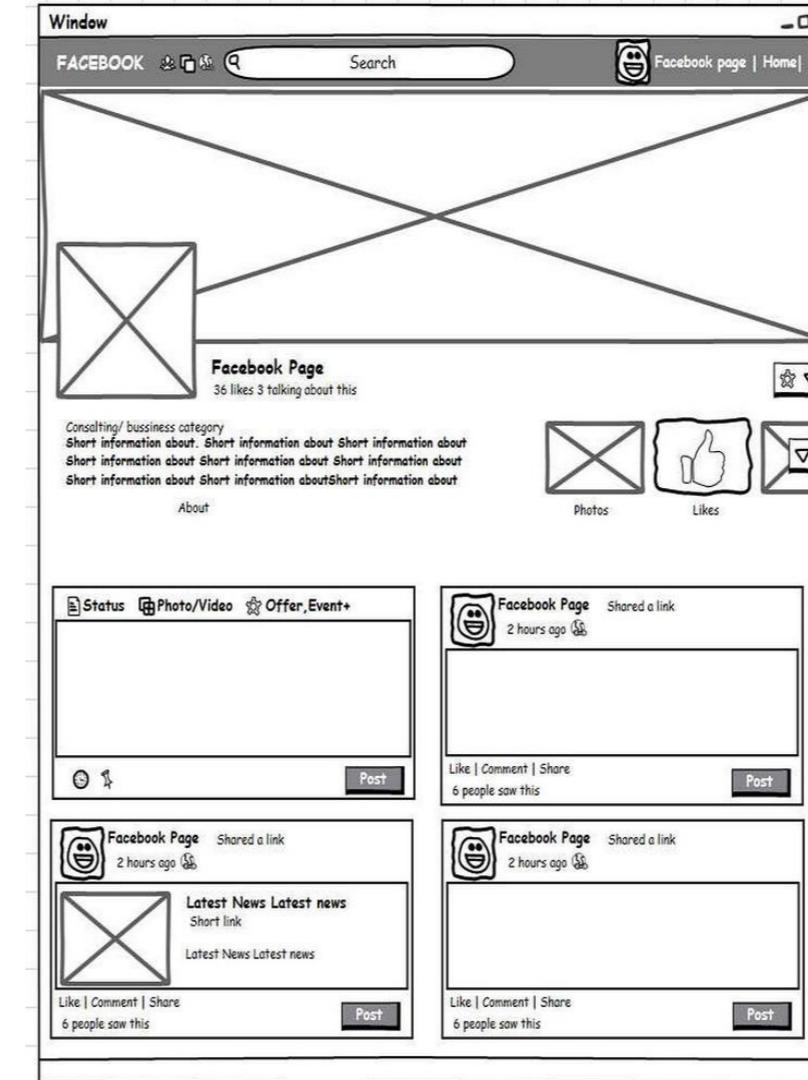
Verwendet [Visual Paradigm Online](#), [draw.io](#), anderen verfügbar Tools oder Pen&Paper.

Vergleiche.: <https://www.siemens-home.bsh-group.com/ch/de/produkte/vernetzte-hausgeraete/kuehlen-und-gefrieren>



Quelle: <https://www.siemens-home.bsh-group.com/ch/de/produkte/vernetzte-hausgeraete/kuehlen-und-gefrieren>

# Weiter Technik: Wireframes (GUI Mockup)



Quelle: <https://www.flickr.com/photos/mockupbuilder/8705902115> | Freeware: <https://wireframe.cc/>

## Requirements priorisieren

# Priorisierung



Quelle: Dall-E. Prompt «Bild zu Priorisierung». Zugriff 30.01.2025

# Ad-hoc vs analytische Priorisierungstechnik

- Bei einer **Ad-hoc-Priorisierungstechnik** weist (erfahrungsbasiert) ein Experte jeder Anforderung einen Wert zu.
- Die **analytische Priorisierungstechnik** geht systematischer vor, z.B. durch paarweisen Vergleich von Anforderungen untereinander oder dadurch, dass verschiedene Experten verschiedene Kriterien bewerten, die dann zusammen die Priorität ergeben.

# Eine Auswahl an Methoden

## „Ad-Hoc“ Priorisierungstechnik

- **Pairwise Comparison** – Anforderungen werden paarweise verglichen und basierend auf der subjektiven Einschätzung priorisiert.
- **Ranking** – Anforderungen werden auf einer Liste nach subjektiver Wichtigkeit von hoch nach niedrig sortiert.
- **Planning Poker** – Stakeholder bewerten Anforderungen mittels verdeckter Zahlenkarten und gleichen ihre Einschätzungen durch Diskussion ab.
- **100-Dollar-Technik** – Beteiligte erhalten fiktive 100 Geldeinheiten und verteilen diese auf Anforderungen entsprechend ihrer Wichtigkeit.
- **MoSCoW-Methode** – Anforderungen werden in Must-have, Should-have, Could-have und Won't-have eingeteilt, um Dringlichkeit und Bedeutung zu verdeutlichen.
- **Dot Voting (Punktbewertung)** – Teilnehmer vergeben Punkte für Anforderungen, wobei die mit den meisten Punkten priorisiert werden.

Quelle: Bühne, Herrmann 2015, Handbuch Requirements Management nach ss Standard

## Analytische Priorisierungstechnik

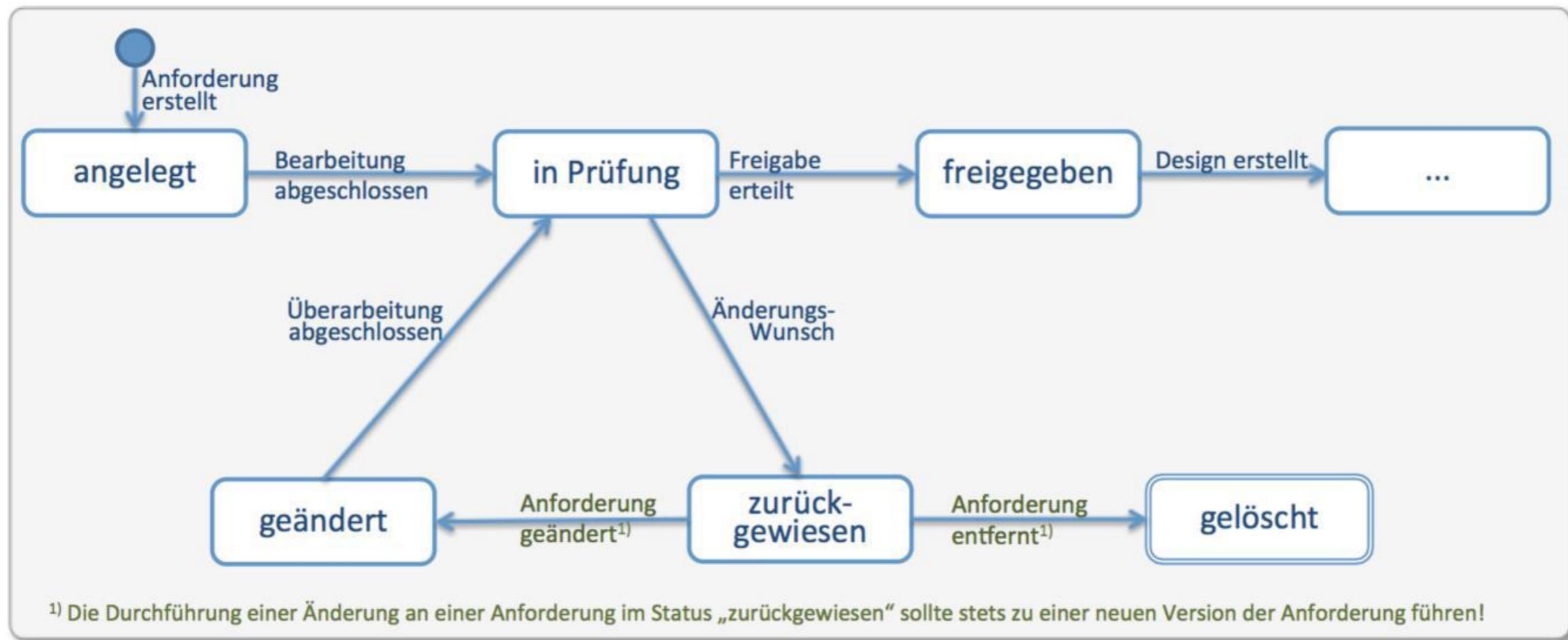
- **Analytical Hierarchy Process (AHP)** – Anforderungen werden paarweise verglichen und gewichtet, um eine mathematisch fundierte Priorisierung zu erhalten.
- **Nutzwertanalyse** – Anforderungen werden anhand vordefinierter Kriterien mit Bewertungen versehen, um eine gewichtete Entscheidung zu ermöglichen.
- **Kostennutzen-Analyse** – Die Priorisierung erfolgt auf Basis des zu erwartenden Nutzens im Verhältnis zu den entstehenden Kosten.
- **WSJF (Weighted Shortest Job First)** – Anforderungen mit dem besten Verhältnis von geschätztem wirtschaftlichem Nutzen zu Implementierungsaufwand werden priorisiert.

COST OF DELAY		
(Actual) BUSINESS VALUE + Business Value * (1-Uncertainty)	TIME CRITICALITY	RISK REDUCTION / OPPORT. ENABLEMENT
Wie hoch ist <b>der intuitive Wert</b> für Kunden oder die Positionierung im Markt (inkl. Strategie-Fit, Case-Potenzial, Leuchtturm)?	Wie wichtig ist es, <b>schnell zu handeln</b> ? Wartet der Markt bzw. die Opportunity auf uns?	Vermindert <b>der Case Risiken</b> , bringt weiteres Skalierungspotenzial und / oder schafft <b>Grundlagen</b> für weitere Cases?
<b>WSJF=</b>		
JOB DURATION		
Personliche Einschätzung, <b>wie lange braucht es</b> , bis der Case einen Grossteil seiner Wirkung entfaltet (ca. 80% des Potenzials)		

Requirements  
managen

# Requirements verwalten

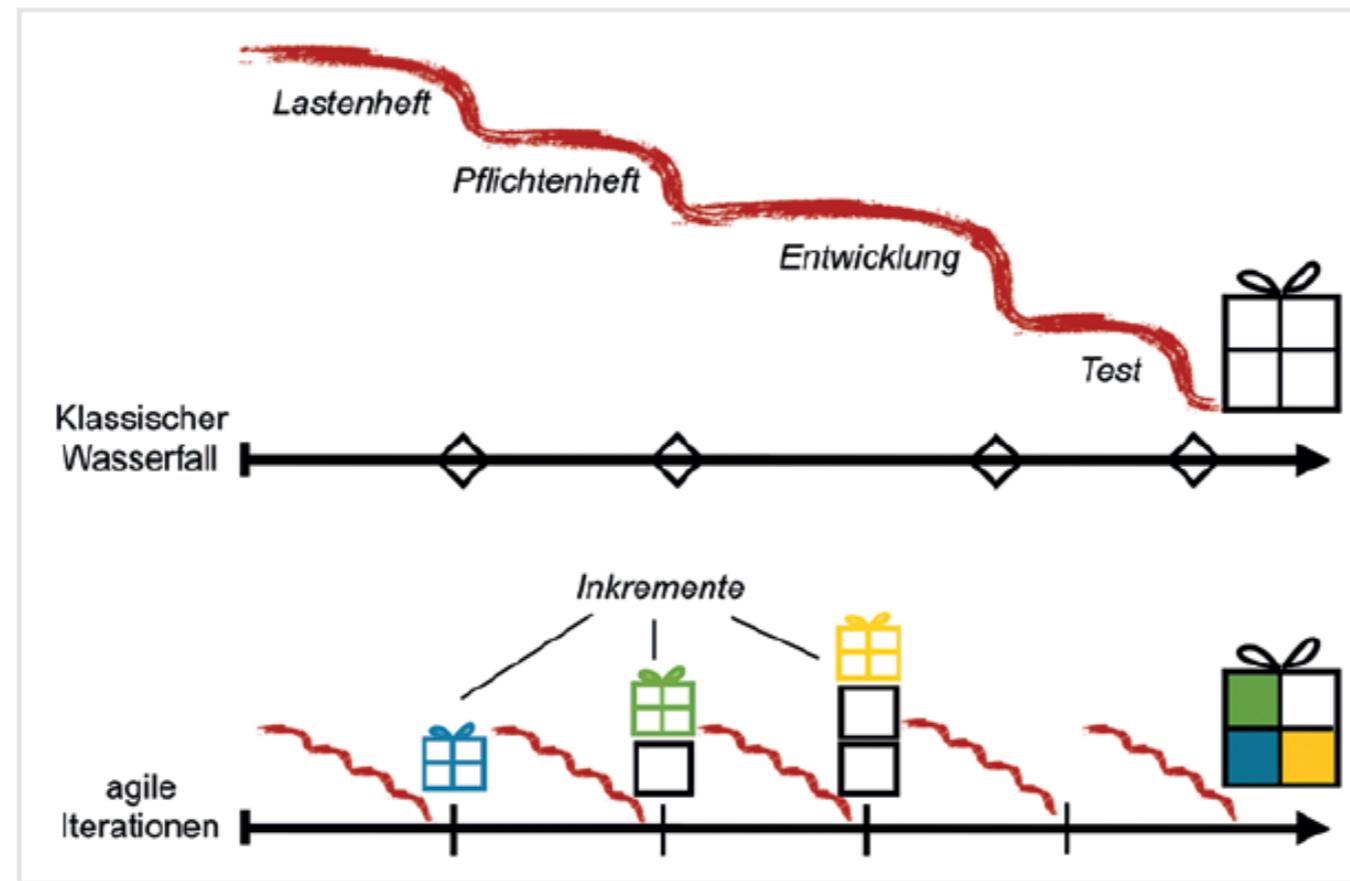
# Versions- und Änderungsmanagement



Quelle: Bühne, Herrmann 2015, Handbuch Requirements Management nach ss Standard

# Ablauf Klassisch vs Agil

Wann werden – je nach Vorgehensmethodik – die Anforderungen definiert?



Quelle: Preussig, J. (2024). Das Agile Manifest: die Basis des Agilen Projektmanagements

# Gruppenarbeit

## Auftrag 6



# Feedback Gruppenarbeit

## Auftrag 5



## Auftrag 5: Klassenfeedback

- «Das System folgt einer service-orientierten Architektur (SOA) zwischen den einzelnen Komponenten (SAP S/4HANA, ChatCRM, Chatbot-Service, Datenbanken). Die Systeme kommunizieren über Schnittstellen und nutzen zentrale Datenbanken als gemeinsame Grundlage.»
- Aufgabe DB: «Unterstützung von Offline-Zugriff durch Zwischenspeicherung»



# Auftrag 5: Feedback pro Gruppe

## Vormittag

Gruppe 2

Gruppe 4

Gruppe 7 (kein Feedback)

## Nachmittag

Gruppe 1

Gruppe 7

Gruppe 2 (kein Feedback)

Gruppe 5 (kein Feedback)

Gruppe 8 (kein Feedback)

Gruppe 10 (kein Feedback)