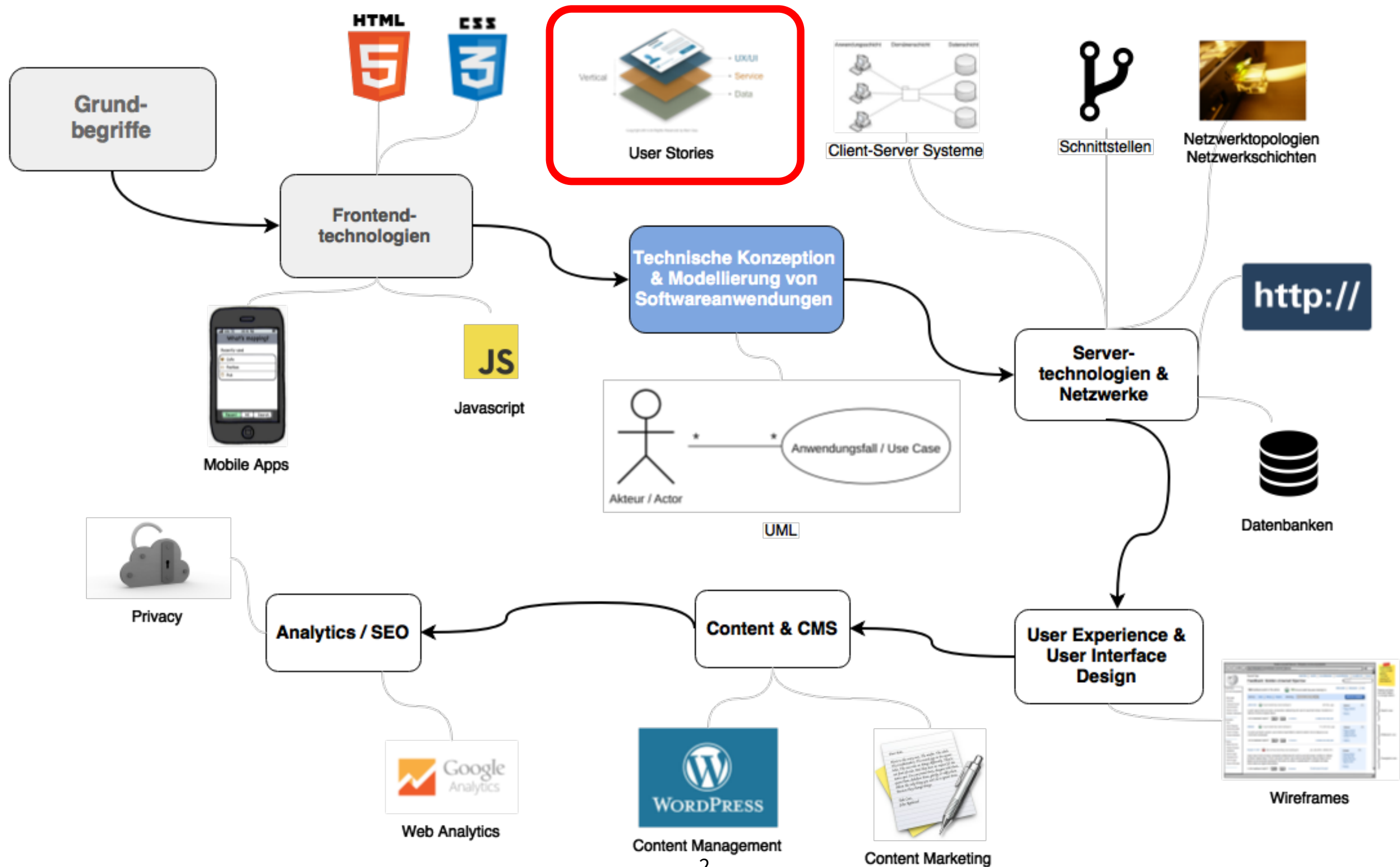


Technische Konzeption

Einführung in Softwaretechnologien

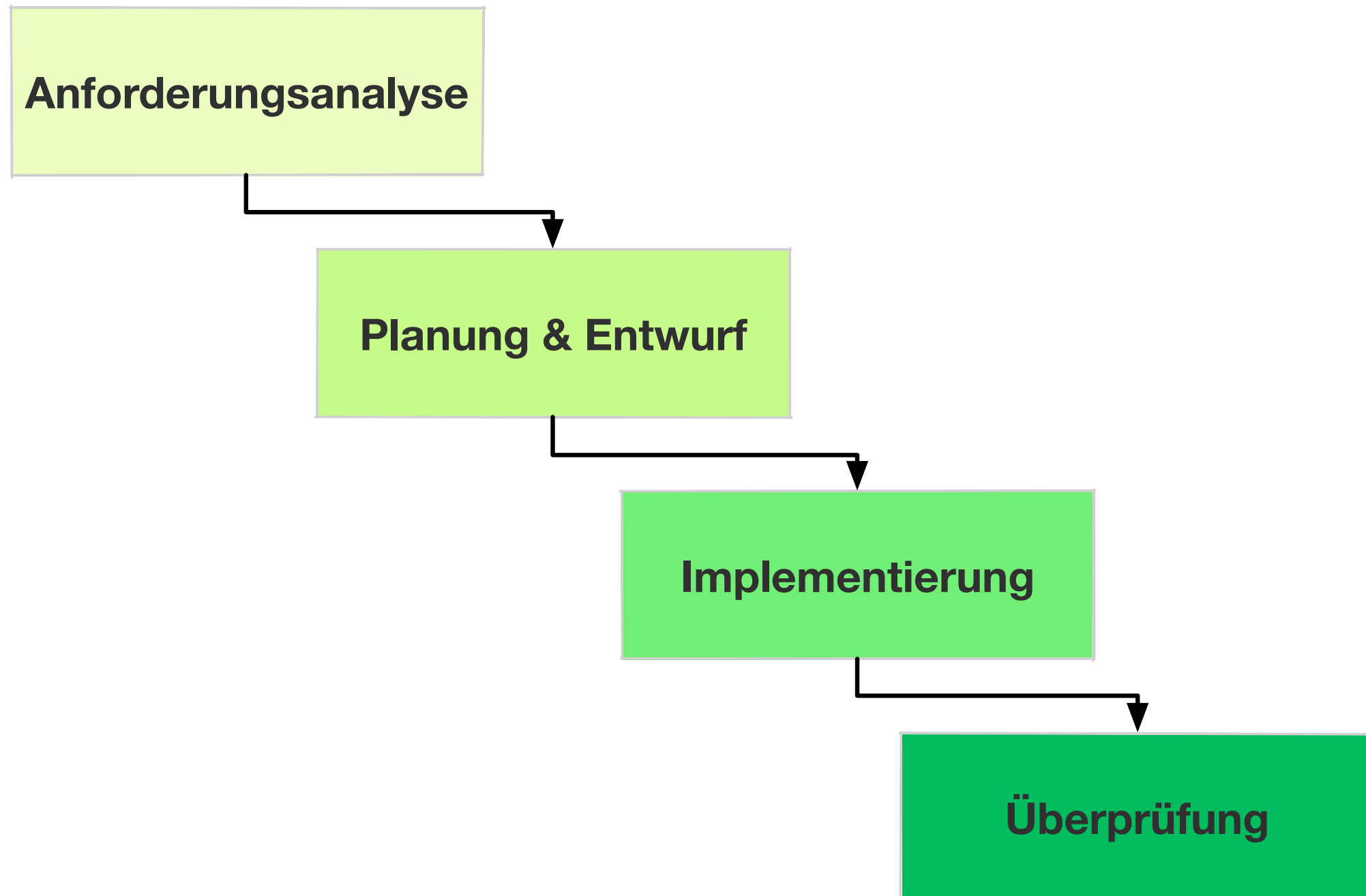
Überblick



Protokolle

- **Strukturierende Elemente** nutzen
 - Listen
 - Überschriften verschiedener Ordnungen
- **Links** auf weiterführende Quellen setzen
- **Schlagworte** vergeben
- Immer die **Kategorie „Protokolle“** wählen

Softwareprojekte in einer perfekten Welt



I. Anforderungsanalyse

- Die Anforderungsanalyse (requirements engineering) dient der **Ermittlung aller Anforderungen des Auftraggebers** an das Projekt
- Die Anforderungen werden **gesammelt** (elicitation), **strukturiert** und **geprüft**
- Dadurch wird ein **gemeinsames Verständnis** hergestellt
- Das Ergebnis wird in einem **Lastenheft** festgehalten.

*DIN 69901-5 beschreibt das **Lastenheft** als die
„vom **Auftraggeber festgelegte** Gesamtheit der
Forderungen an die Lieferungen und
Leistungen eines Auftragnehmers innerhalb eines
Auftrages.“*

2. Planung & Entwurf

- Der **Auftragnehmer** erstellt auf Grundlage des Lastenheftes ein **Pflichtenheft**.
- Es beschreibt wie und womit etwas realisiert werden soll
- Darin werden **jeder Anforderung des Lastenhefts** (Requirement) eine oder mehrere **Leistungen des Pflichtenheftes** (Features) zugeordnet

*DIN 69901-5 beschreibt das **Pflichtenheft** als die
„vom Auftragnehmer erarbeiteten **Realisierungs-**
vorgaben aufgrund der Umsetzung des vom
Auftraggeber vorgegebenen Lastenheftes“*

3. Umsetzung

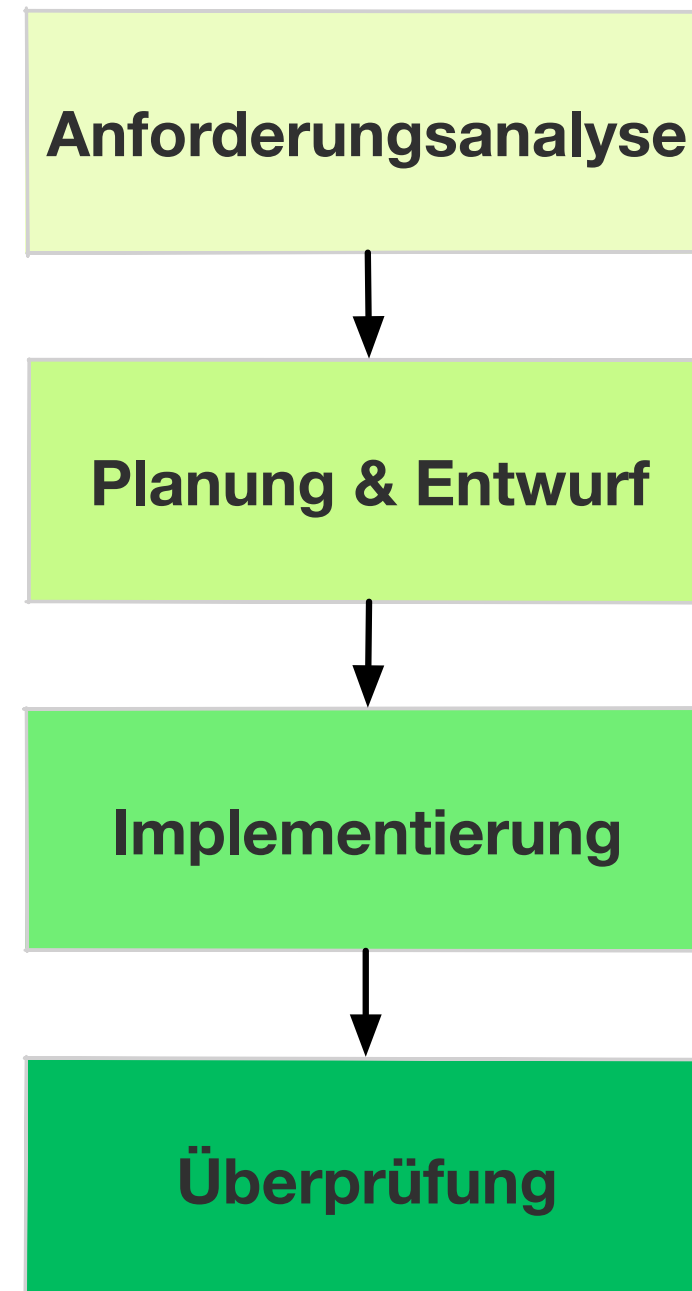
- In der **Implementierungsphase** setzt der Auftragnehmer die Features des Pflichtenheft Schritt für Schritt um,
- **testet** sie im Anschluss auf **Übereinstimmung** mit der Spezifikation und
- **übergibt das Projekt** nach Abschluss an den Kunden

Party!



Wasserfallmodell

- **Aktivitäten** sind in der **vorgegebenen Reihenfolge** durchzuführen
- Der **Entwicklungsablauf** ist streng **sequenziell Top-down**
- Eine **Benutzerbeteiligung** ist **nur in der Anfangsphase** vorgesehen, Entwurf und Implementierung erfolgen ohne Beteiligung Auftraggebers
- **Weitere Änderungen** stellen danach **Neuaufträge** dar



Wasserfallmodell

Vorteile

- **Klare Abgrenzung der Phasen**
- Einfache Möglichkeiten der **Planung und Kontrolle**
- **Sehr effizientes Modell**
 - *bei stabilen Anforderungen und klarer Abschätzung von Kosten und Umfang*

Aber...

- Eine Studie von IAG Consulting* ergab:
 - Bei **68%** der untersuchten **IT-Projekte** war ein **Scheitern wahrscheinlicher als ein Erfolg!**
 - Von diesen Projekten hatten **50%** das **Fertigstellungsdatum zu 180% überschritten** oder **mehr als 160% des Budgets benötigt**
 - um **70%** der Funktionalität umzusetzen

* <http://bit.ly/11SgBLH>

Aber...

- Es zeigte sich, dass die Ursache vor allem **Unklarheiten in der Spezifikationen der Anforderungen** waren. Diese führten zu 60% der Budgetüberschreitungen
- Das Problem ist also **Stille Post und ein starrer Prozess**, der alle Anforderungen schon zu einem sehr frühen Zeitpunkt festschreibt

- Wenn wir davon ausgehen, dass zu einem Projekt ein **Auftraggeber** und ein **Auftragnehmer** gehören, bedarf es:
 - zunächst einer **Vorarbeit auf Seiten des Auftraggebers**, um im Rahmen eines **Briefings** für sich selbst und für den Auftragnehmer **den Projektgegenstand zu beschreiben**
 - Im zweiten Schritt ist der **Auftragnehmer** dann gefordert, **auf Grundlage des Briefings und in Abstimmung mit dem Kunden die Anforderungen des Projekts zu identifizieren** und so zu **strukturieren**, dass das Projekt umgesetzt werden kann

Das Briefing

Briefing

- Im besten Falle entspricht ein Briefing einer etwas **informelleren Variante eines Lastenhefts**
- Es dient in erster Linie dazu, den **Auftragnehmer** mit allen **Informationen zum Projekt** und dessen Rahmen zu versorgen
- Ein Briefing ist jedoch ebenfalls relevant, um **auf Seiten des Auftraggebers ein gemeinsames Verständnis** zum Projekt zu entwickeln

Checkliste für ein Briefing

Briefing

Der Auftraggeber

- Tätigkeit
 - *In welcher Branche ist der Auftraggeber aktiv? Wo erfolgt die Wertschöpfung?*
- KPIs
 - *Wie viele Mitarbeiter hat das Unternehmen, wie groß ist der Umsatz, ...*
- Unternehmensgeschichte
 - *Wie ist das Unternehmen zu dem geworden, was es heute ist?*
- Unternehmensstruktur und -organisation

Briefing

Die Ausschreibung

- Ansprechpartner
- Kurzbeschreibung des Projekts
- Budgetrahmen
- Erwartung an den Auftragnehmer
 - *Nach welchen Kriterien wird der Auftragnehmer ausgewählt?*
- Zeitraumen?

Briefing

Mitbewerber und Marktlage

- Alleinstellungsmerkmal (USP)
 - *Über welche Alleinstellungsmerkmale verfügt das Angebot oder die Firma?*
- Marktposition
 - *Welche Position besetzt der Auftraggeber / das Produkt / die Dienstleistung im Markt? (Nischenplayer, Marktleader, ...)*
- Mitbewerber
 - *Wer sind die engsten Mitbewerber?*

Briefing

Mitbewerber und Marktlage

- Relevante Webauftritte von Mitbewerbern
- Nennenswerte Funktionen/Services/Apps von Mitbewerbern
- Social Media-Aktivitäten von Mitbewerbern
- Benchmarks
 - *Websites, Apps, Funktionen, an denen das Projekt gemessen wird?*

Briefing

Das Projekt

Auftrag

- Das Primärziel
 - *Kurze Beschreibung des hauptsächlichen Ziels für dieses Projekt*
 - *Gibt weitere Ziele, die mit diesem Projekt erreicht werden sollen?*
- Kernzielgruppe
- Zielgruppenbedürfnisse
 - *Welche Anforderungen soll das Projekt vor allem adressieren?*

Briefing

Das Projekt

Anforderungen an Usability

- Unterstützte Browser
- Barrierefreiheit
 - *Soll der Auftritt barrierefrei gestaltet werden?*
- Mehrsprachigkeit
- Bestehende Struktur der Site
- Interaktion
 - *Wie soll die Zielgruppe über das Projekt interagieren können (z.B. etwas kaufen, etwas sharen bzw. liken)?*

Briefing

Projektorganisation

- Termine & Meilensteine
 - *Wie sieht die Roadmap für dieses Projekt aus?*
- Projektverantwortliche & Rollen
 - *Wer ist in die Entscheidungs- und Freigabeprozesse involviert?*
- Ablauf GoLive
 - *Wie sieht der GoLive Prozess für dieses Projekt aus?*

Strukturierung von Anforderungen

User-Stories

- Eine **User-Story** („Anwendererzählung“) ist eine in **Alltagssprache** formulierte Software-Anforderung
- Sie umfasst in der Regel nicht mehr als **zwei Sätze**
- **User-Stories** werden zur **Spezifikation von Anforderungen** eingesetzt
- Dabei wird jede User-Story auf eine **Story-Card** geschrieben

Als <Rolle>

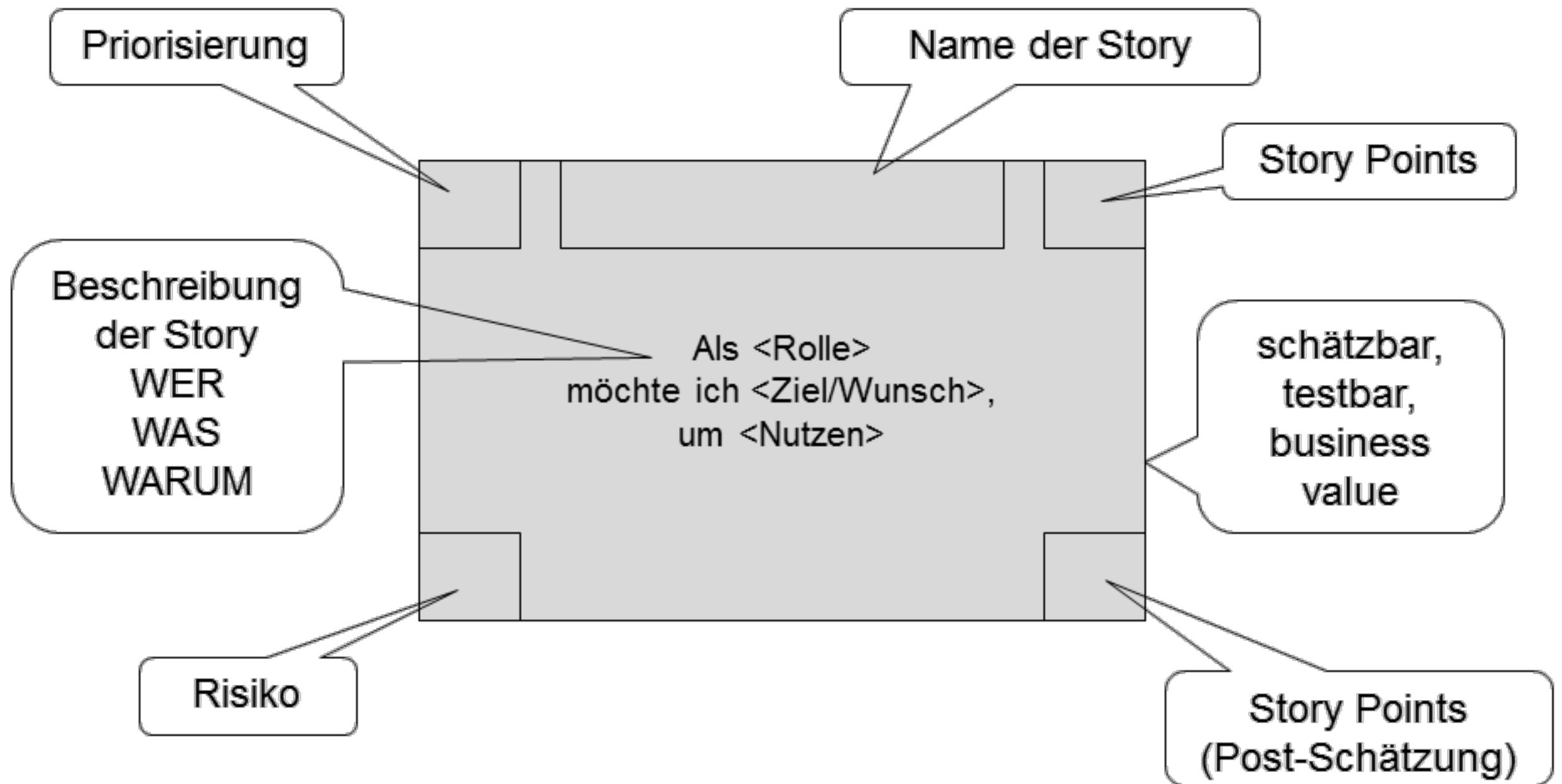
möchte ich <Ziel/Wunsch>

Story-Cards

- Die Story-Cards dienen als **Kommunikationsmittel zwischen Anforderern und Umsetzern**
- Die **Vorderseite** der Story-Card enthält die **Beschreibung der Story**

Story-Card

Vorderseite

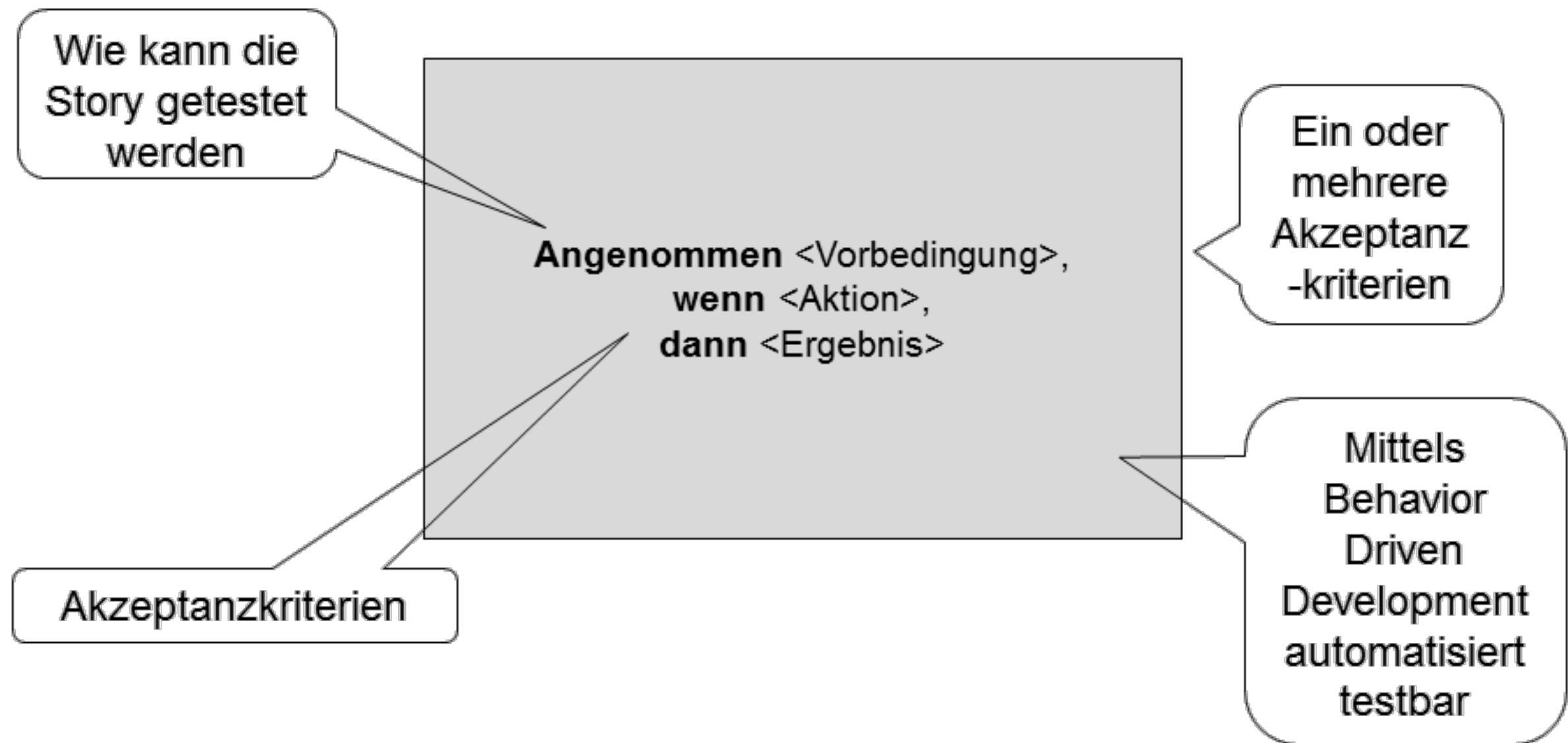


Story-Cards

- Die **Rückseite** enthält die Beschreibung der **Akzeptanzkriterien**
- Der **Anforderer** beschreibt, wie er die **korrekte Umsetzung der User-Story** testen würde.
Dies ist sowohl für das Verständnis, als auch für den Test der User-Story hilfreich

Story-Card

Rückseite



Card, Conversation, Confirmation

Card

- Eine Karte enthält nur so viel Information, wie nötig ist, um die **Anforderung zu identifizieren!**

Conversation

- Die **Anforderung** selbst wird von Kunden und Programmierer **besprochen**. Diese Konversation findet wieder und wieder statt, so lange, bis das Feature abgeschlossen worden ist.

Confirmation

- Nach **Fertigstellung** der Aufgabe zeigt der Programmierer dem Kunden an Hand der vorher entwickelten **Akzeptanzkriterien**, dass die Aufgabe erwartungsgemäß umgesetzt wurde

Story Decomposition

- Ziel ist es, mit User-Stories einen **adäquaten Detaillierungsgrad** zu erreichen!
- Dafür werden zunächst die **Anwender und ihre Ziele identifiziert**
- Diese werden in Form von **Epics** mit einem hohen Abstraktionsgrad **zusammengefasst**
- Anschließend werden die **Ziele in Aktivitäten** unterteilt und diese in **User Stories** herunter gebrochen

Struktur einer Story Map:

User > Goals > Actions > Stories

Story Map

User

Goal

Action

Story

Story

Story

Story Map

Besucher

Produkt
kaufen

Warenkorb

Produkt
auswählen

Produkt in
Warenkorb

Warenkorb
speichern

Checkout

Bezahlen

Story Map



- Story Mapping ist also ein **Top-Down Ansatz** um **Anforderungen** zu sammeln und als **Baum** zu strukturieren



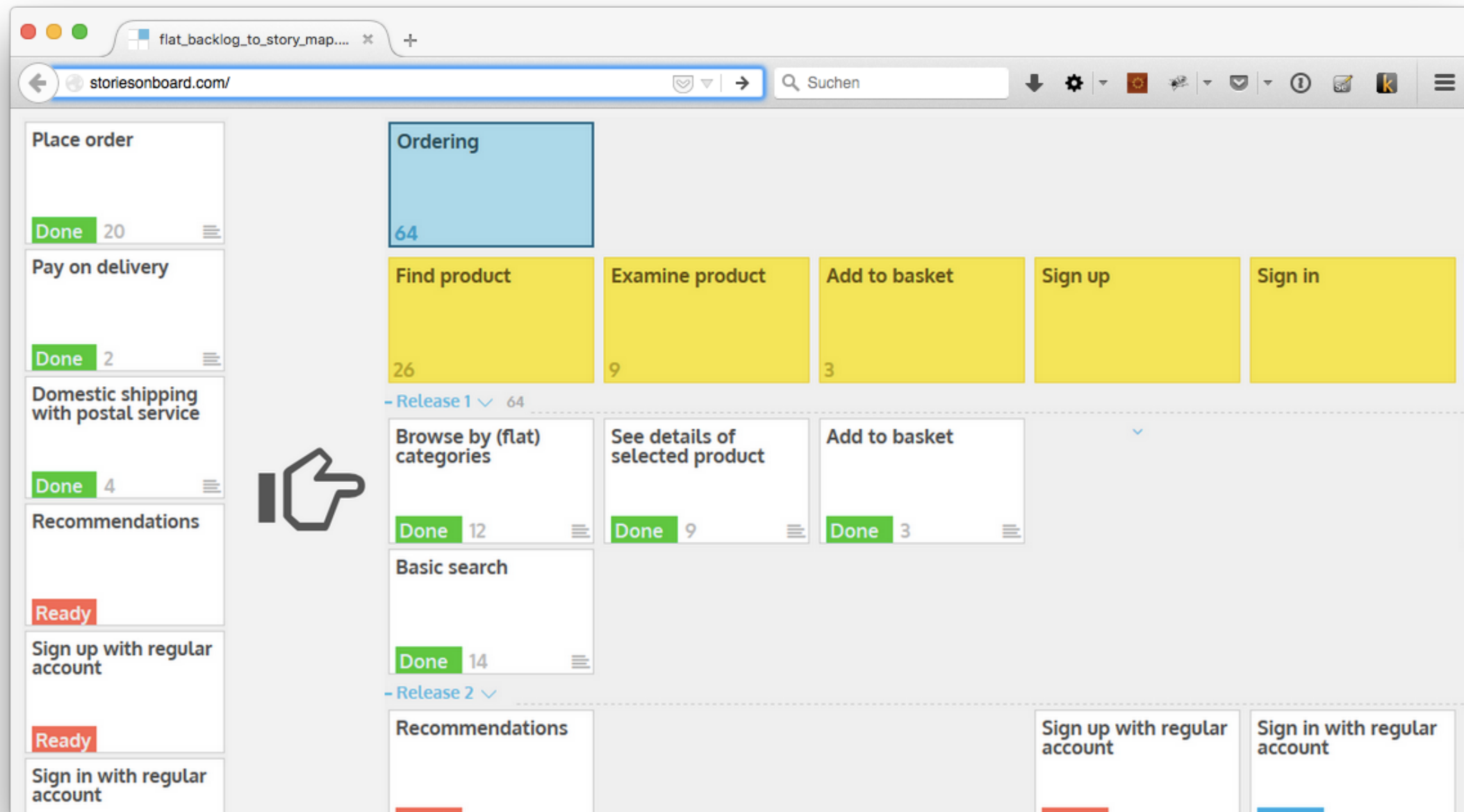
Jetzt sind Sie dran!

Aufgabe

- Tun Sie sich in **4 Gruppen** zusammen
- Entwickeln Sie mit Hilfe verschiedenfarbiger Notizzettel **die Ziele eines Nutzers und für ein Ziel die entsprechenden User Stories**
- Nutzen Sie dafür **das Briefing, das wir gerade gemeinsam entwickelt haben**

Sie haben 10 Minuten Zeit!

storiesonboard.com



<https://b042015.storiesonboard.com/m/mapshop-sample>

Anwendungsfälle (Use Cases)

Anwendungsfälle

- Ein alternativer Ansatz zur **Ermittlung von Systemanforderungen** sind **Anwendungsfälle** (Use Cases)
- Use Cases geben Auskunft darüber, was ein geplantes **System aus Sichtweise der Benutzer** leisten soll und stellen damit die **Menge der Systemfunktionen** dar.
- Diese Funktionen müssen **im weiteren Projektverlauf genauer zu beschrieben** werden.

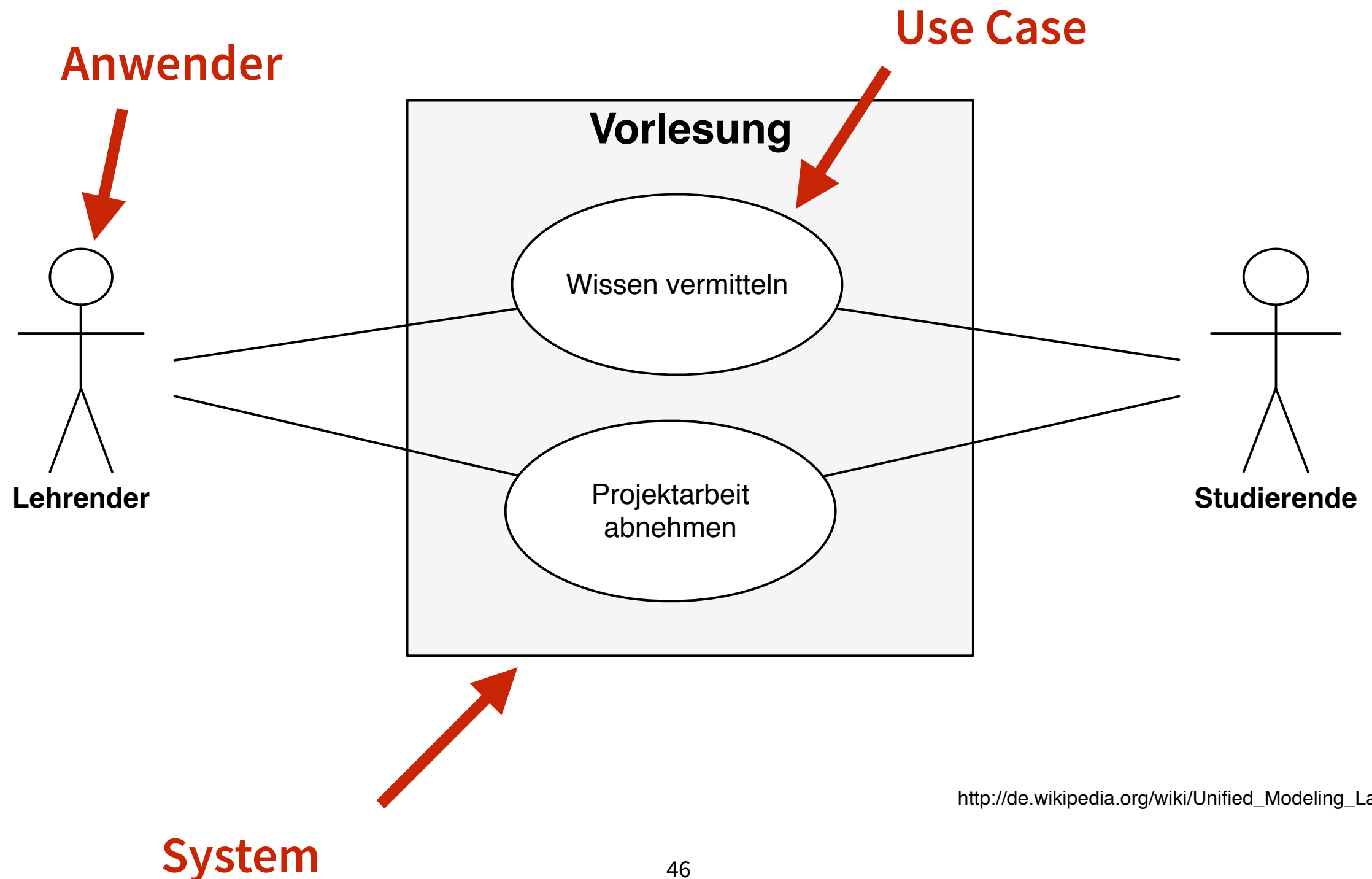
Use Case Identifikation

- Folgende **Fragen** sind zur **Identifikation von Use Cases** hilfreich:
 - Für **welchen Zweck** soll das System eingesetzt werden?
 - **Wofür** will der **spätere Benutzer** es einsetzen?
 - Durch welches **externe Ereignis** wird ein Use Case angestoßen?
 - Welches **Ergebnis** liefert ein Use Case dem späteren Nutzer?

Use Case Diagramm in UML

- **Sammlung und Visualisierung** von Anwendungsfällen erfolgt oft mit Hilfe eines **UML Use Case Diagramms**
- Hierbei werden **Anwendungsfälle** und **Akteure** mit hohem Abstrahierungsgrad **durch Linien einander zugeordnet**

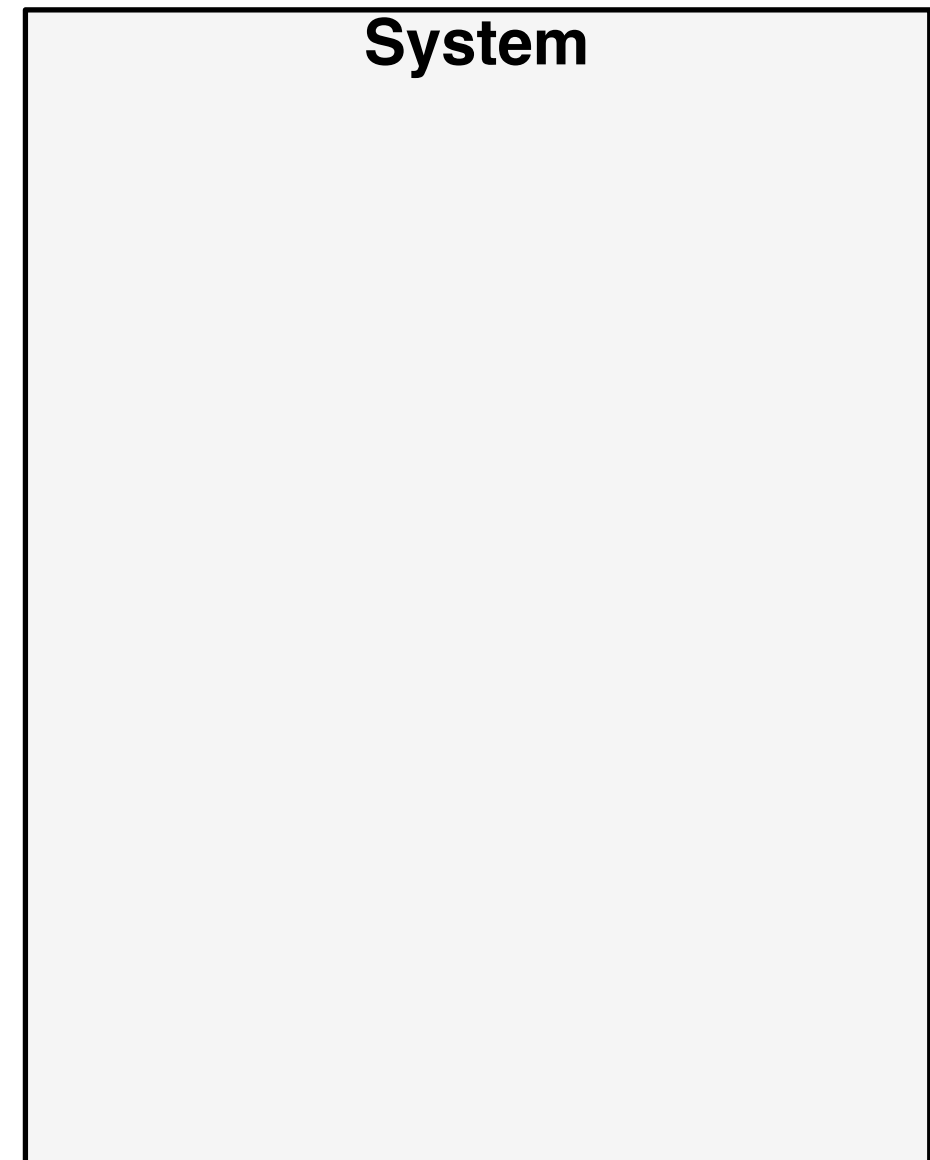
Use Case Diagramm in UML



Use Case Diagramm in UML

System

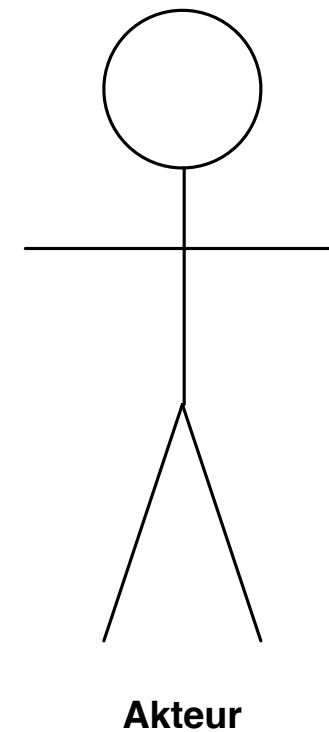
- Das **Rechteck** stellt das **geplante System** dar. Das System erhält einen Namen.
- Ein **Use Case Diagramm** kann auch **mehrere Systeme** enthalten, die ineinander geschachtelt sein dürfen. Dadurch kann ein System in Teilsysteme gegliedert werden.



Use Case Diagramm in UML

Akteur

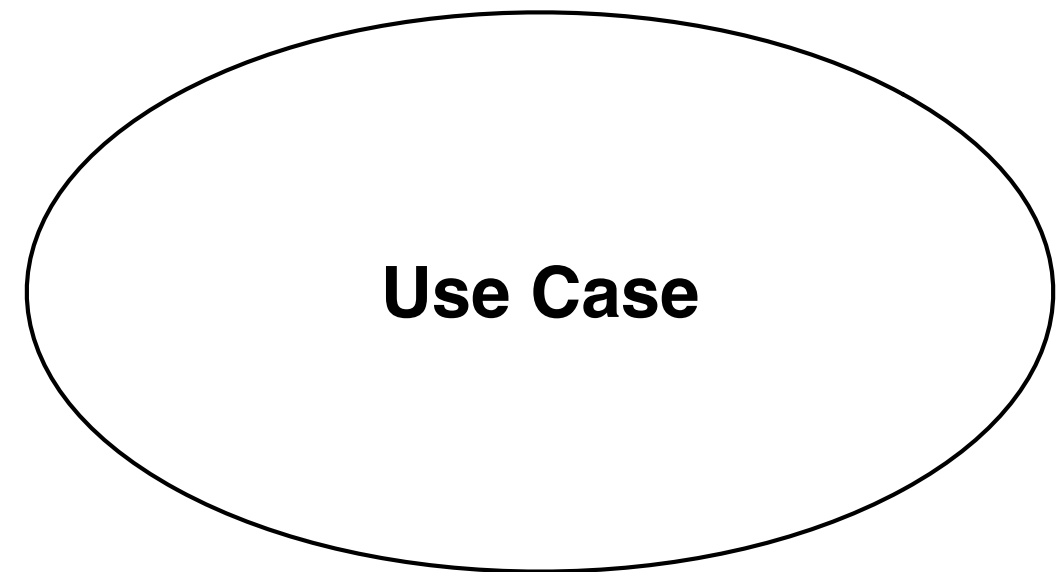
- **Person**, die auf das System zugreift **oder ein anderes System**, das mit dem geplanten System kommuniziert.
- **Gehört nicht zum geplanten System!**
- Der Akteur kann als **Strichmännchen** oder als **Kästchen mit dem Stereotyp <<actor>>** dargestellt werden.



Use Case Diagramm in UML

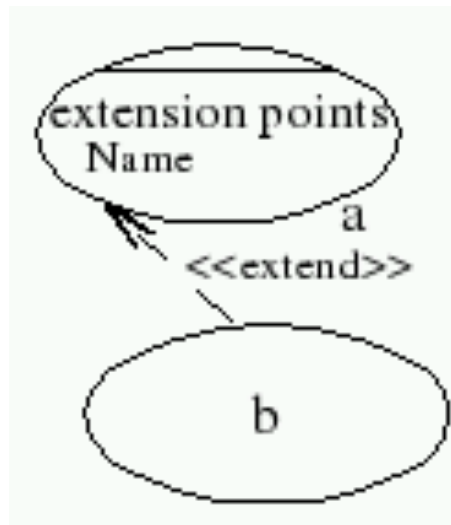
Use Case

- Ein Anwendungsfall ist ein in sich **abgeschlossener Vorgang**, der für einen oder mehrere Akteure ein **beobachtbares Ergebnis** liefert.
- Er beschreibt aus Sicht der Akteure welche Leistungen das System für den Anwender zur Verfügung stellt.
- Kann auch als Rechteck mit einer Ellipse dargestellt werden.

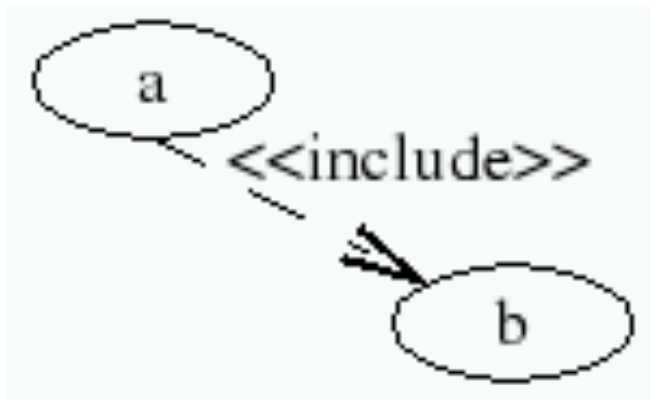


Use Case Diagramm in UML

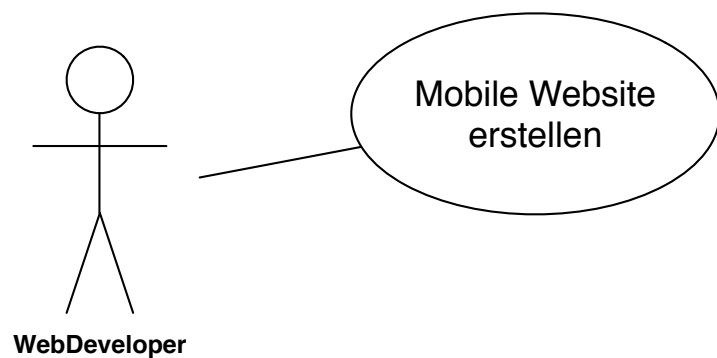
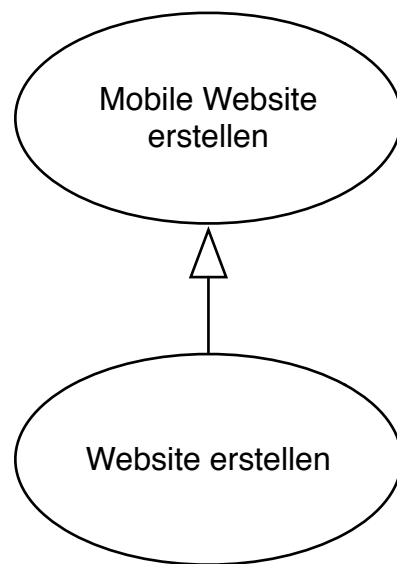
- Durch die **extend Beziehung** beschreibt die Erweiterung des Verhaltens eines Use Cases durch einen anderen.



- Die **include Beziehung** definiert einen Use Case, der die Funktionalität, die ein anderer Use Case zur Verfügung stellt, importiert. Der Use Case a importiert die Funktionalität des Use Case b.



Use Case Diagramm in UML



- Die **Verebungsbeziehung** kann verwendet werden um Verhalten zwischen Use Cases zu vererben oder Vererbungen zwischen Akteuren aufzubauen.
- Eine **Assoziation** zwischen einem Akteur und einem Use Case beschreibt den **Zugriff des Akteurs auf die Funktionalität** des Systems bzw. die **Antwort des Systems an einen Akteur**.

Use Case Spezifikation

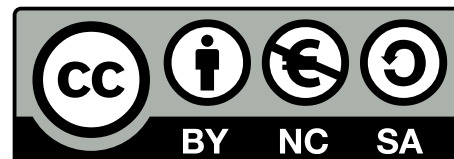
- Die **Textstruktur** eines Use Cases enthält üblicherweise **folgende Abschnitte**:
 - **Name** des Use Case
 - **Auslösendes Ereignis**
 - Verhalten im **Normalfall**
 - Verhalten im **Fehlerfall**
 - **Ergebnis**

Use Case

Textstruktur

- **Name des Use Case:** Wissen vermitteln
- **Auslösendes Ereignis:** Lehrender und Studierende treffen sich im Rahmen einer Vorlesung
- **Verhalten im Normalfall:** Nachdem der Lehrende einen thematischen Kontext hergestellt hat, gibt er Impulse, um die Studierenden zu aktivieren und die Vorlesung zunehmend zu einer Diskussion werden zu lassen.
- **Verhalten im Fehlerfall:** Gelingt es dem Lehrenden nicht mit den Studierenden in einen lebendigen Austausch zu treten, langweilen sich die Studierenden 90 Minuten lang zu Tode und der Lehrende ist frustriert.
- **Ergebnis:** Die Studierenden erwerben neue Kompetenzen und werden zum Denken angeregt.

Vielen Dank.



Dieses Werk ist lizenziert unter einer Creative Commons 4.0 International Lizenz mit folgenden Eigenschaften:

- Namensnennung
- Nicht-kommerzielle Nutzung
- Weitergabe unter gleichen Bedingungen.

Quellen

- Checkliste Website-Briefing
- <http://www.extremeprogramming.org/rules/userstories.html>
- <http://www.mountangoatsoftware.com/agile/user-stories>
- <https://www.thoughtworks.com/insights/blog/story-mapping-visual-way-building-product-backlog>
- Zu Use Cases
https://www.fbi.h-da.de/fileadmin/personal/b.humm/CNAM_SWT_WS0910/Vorlesung/Skript_Softwaretechnik.pdf