



HOCHSCHULE HEILBRONN

# Software Engineering komplexer Systeme

Vorlesung an der Hochschule Heilbronn 2019

Dr. Andreas Heil

## HEUTIGER INHALT

- Anforderungsmanagement
  - Einführung
  - Klassische Anforderungsanalyse
  - Agiles Anforderungsmanagement
  - Übung



# ANFORDERUNGSMANAGEMENT

## WEITERFÜHRENDE LITERATUR



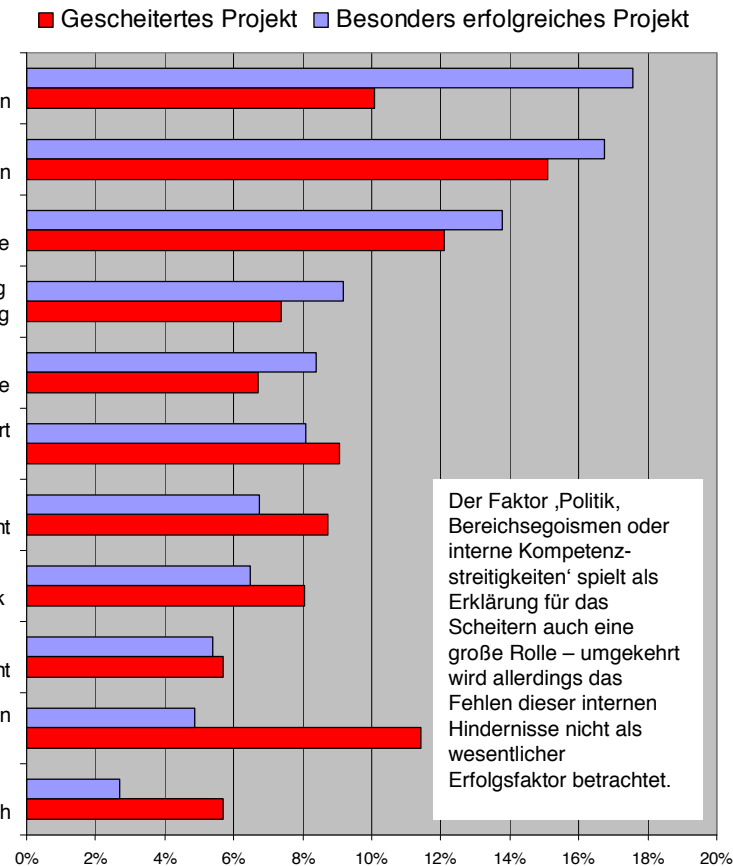
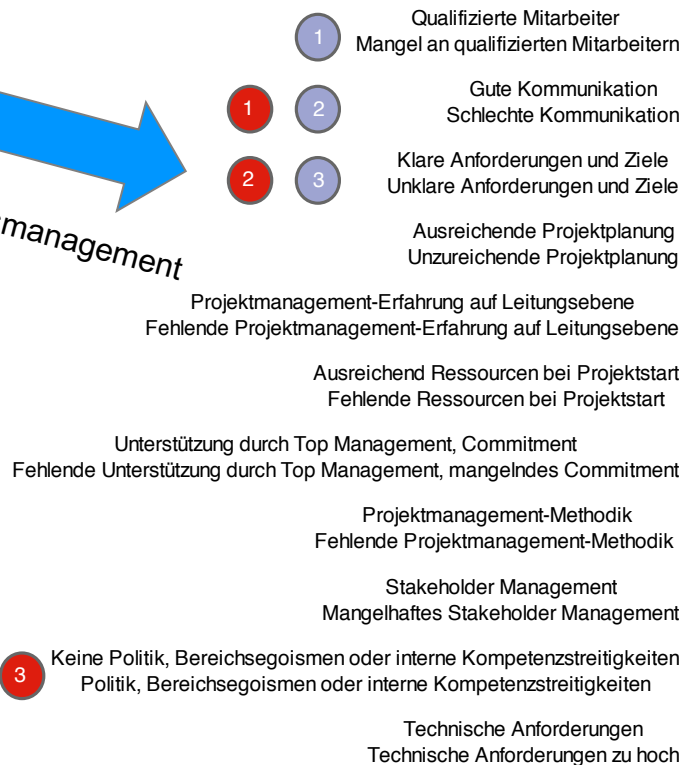
Christine Rupp, die SOPHISTen  
**Requirements-Engineering und -Management**  
Aus der Praxis von klassisch bis agil

ISBN : 978-3-446-43893-4

# WIEDERHOLUNG

## Warum scheitern Projekte?

= schlechtes  
Anforderungsmanagement

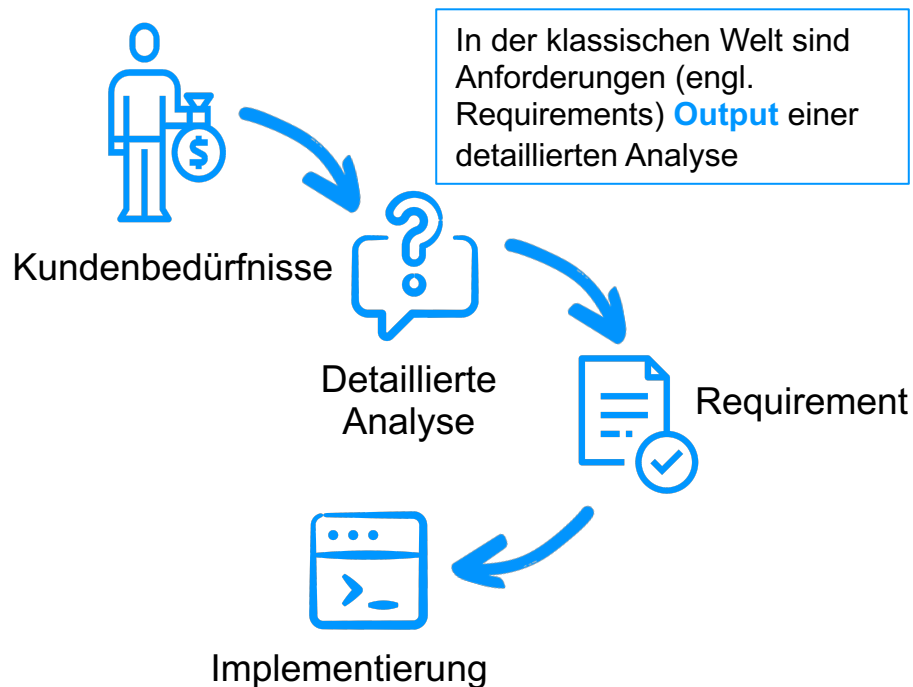


Quelle: Projektmanagement Studie 2008, GPM Deutsche Gesellschaft für Projektmanagement e.V. und PA Consulting Group

# KLASSISCHE VS. AGILE ANFORDERUNGSANALYSE

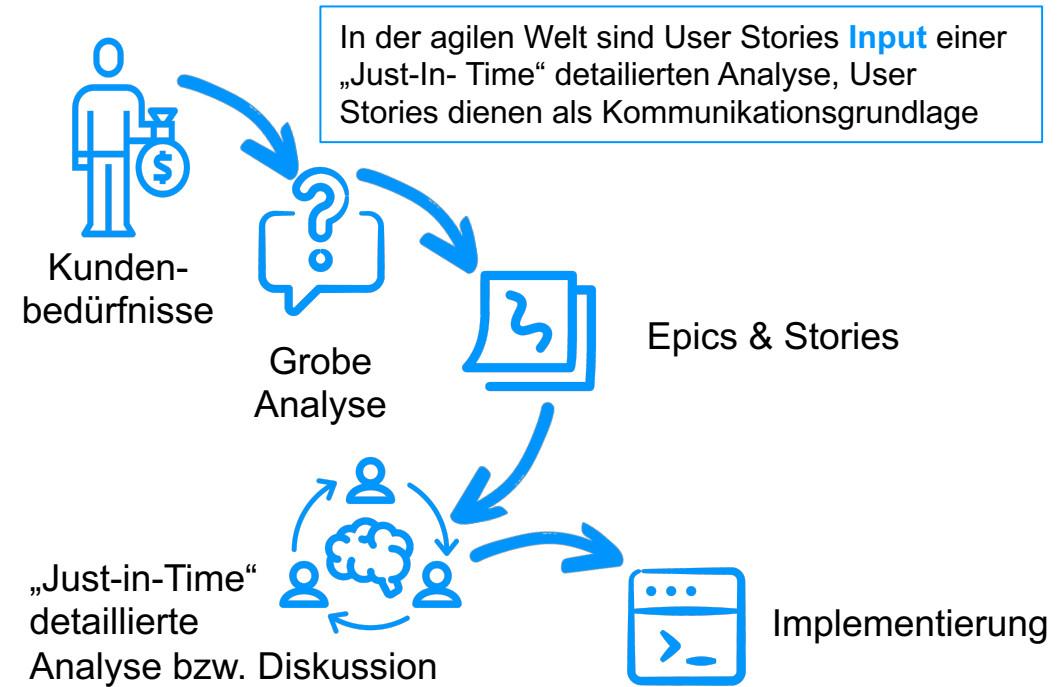
## Klassisch

- Projektplan
- Anwendungsfälle (engl. Use Cases)
- Anforderungen



## Agil

- Regelmäßige Priorisierung (Backlog)
- Anwendungsfälle (engl. Use Cases)
- Epics und Stories

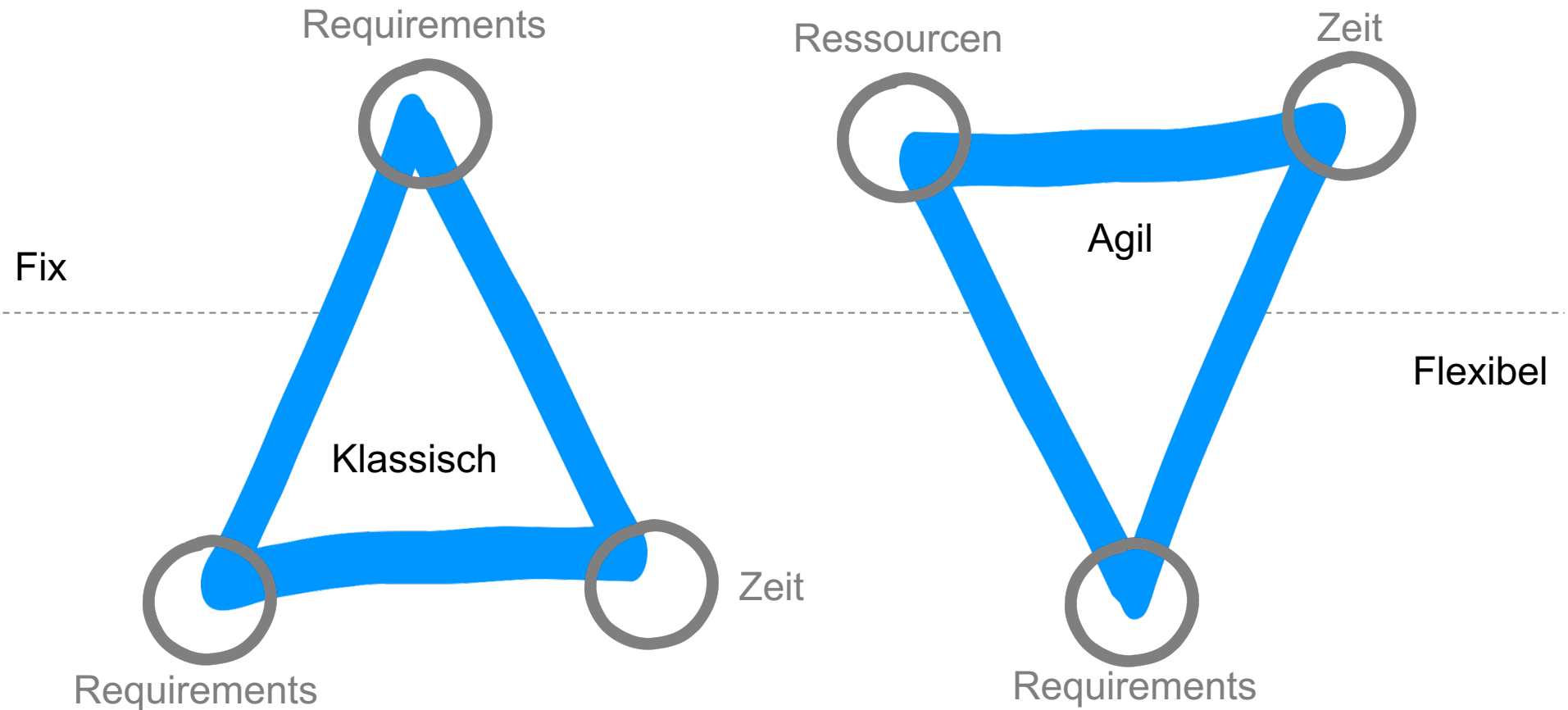


## ÜBER DIE WAND VS. KOLLABORATIVE DISKUSSION



## MAGISCHES DREIECK (WDHL.)

Unterschied zwischen klassischem und agilem Ansatz





## BEISPIEL

### Eine einfache Anforderung

Das System soll dem Kunde die Möglichkeit bieten, zu einem geladenen Patienten alle verfügbaren DICOM Bilder zu laden und anzuzeigen. Die Software soll dem Kunden über ein UI Control „Open DICOM...“ die Möglichkeit bieten, die zugehörigen DICOM Dateien auszuwählen. Nachdem der Benutzer die Dateien ausgewählt hat, soll die SW so schnell wie möglich die DICOM Bilder mittels der Bibliothek DCM laden und mittels der Bibliothek vtk anzeigen. Die SW soll dem Kunden Bildbearbeitungstools nach Stand-der-Technik bieten. Das Laden, Anzeigen und Bearbeiten soll nach dem Model-View- Controller Prinzip implementiert werden.

## BEISPIEL

### Eine einfache Anforderung

Das System soll dem Kunde die Möglichkeit bieten, zu einem geladenen Patienten alle verfügbaren DICOM Bilder zu laden und anzuzeigen.

Was bedeutet das? Kundenanforderung!

Software-Anforderung

Die Software soll dem Kunden über ein UI Control „Open DICOM...“ die Möglichkeit bieten, die zugehörigen DICOM Dateien auszuwählen.

Software-Anforderung

Was heißt das??

Nachdem der Benutzer die Dateien ausgewählt hat, soll die SW so schnell wie möglich die DICOM Bilder mittels der Bibliothek DCM laden und mittels der Bibliothek vtk anzeigen. Die SW soll dem Kunden Bildbearbeitungstools nach Stand-der-Technik bieten. Das Laden, Anzeigen und Bearbeiten soll nach dem Model-View-Controller Prinzip implementiert werden.

Ist hier noch mehr versteckt



## KUNDEN- VS SOFTWARE-ANFORDERUNGEN

### Kundenanforderung

- Kunde ist Anforderer
- Zielsetzung des Kunden
- Was soll mit der Software erreicht werden

Im klassischen Vorgehen  
im **Lastenheft**

### Software-Anforderung

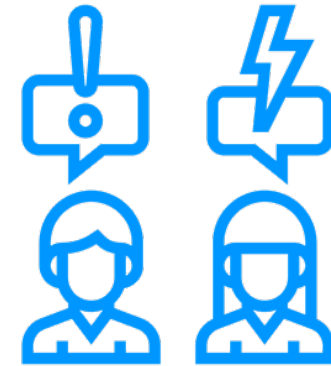
- Anforderung an die Software
- Notwendig um Kundenanforderung zu erfüllen
- Was soll Software bieten um Kundenanforderung zu erfüllen

Im klassischen Vorgehen im  
**Pflichtenheft**



## TYPISCHE PROBLEME BEI DER ANFORDERUNGSANALYSE

- Unklare Zielvorstellung der Nutzer
- Nutzer nicht bekannt
- Hohe Komplexität
  - (Zu) viele Anforderungen
  - Abhängigkeiten zwischen Anforderungen
  - Widersprüchliche Anforderungen
- Sprachbarriere zwischen Nutzer und Analyst / Requirements Engineer / Entwickler
- Schlecht dokumentierte Anforderungen
- Unnötige Anforderungen
- Fehlende Priorisierung der Anforderungen
- Veränderliche Anforderungen
  - Unklare Vorstellungen der Nutzer zu Beginn, Veränderung des Marktes
  - Nicht alle Stakeholder / Anforderer zu Beginn involviert



# QUALITÄTSANFORDERUNG

## Nach IEEE

- Vollständig
- Korrekt
- Konsistent
- Prüfbar
- Eindeutig
- Verfolgbar
- Bewertet (Priorisiert)

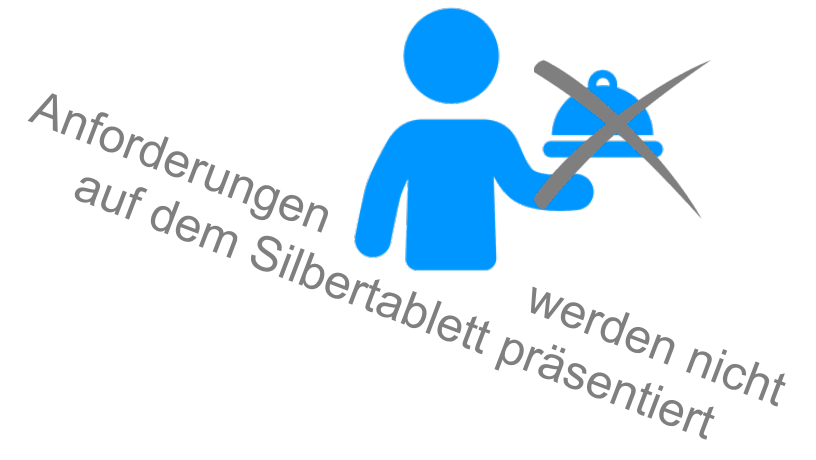
## Schlechte Anforderungen

- Ungenau (z.B. System soll schnell...)
- Redundant
- Widersprüchlich
- Nicht test- bzw. abnehmbar
- Mehrdeutig
- Nicht organisiert
- Unpriorisiert



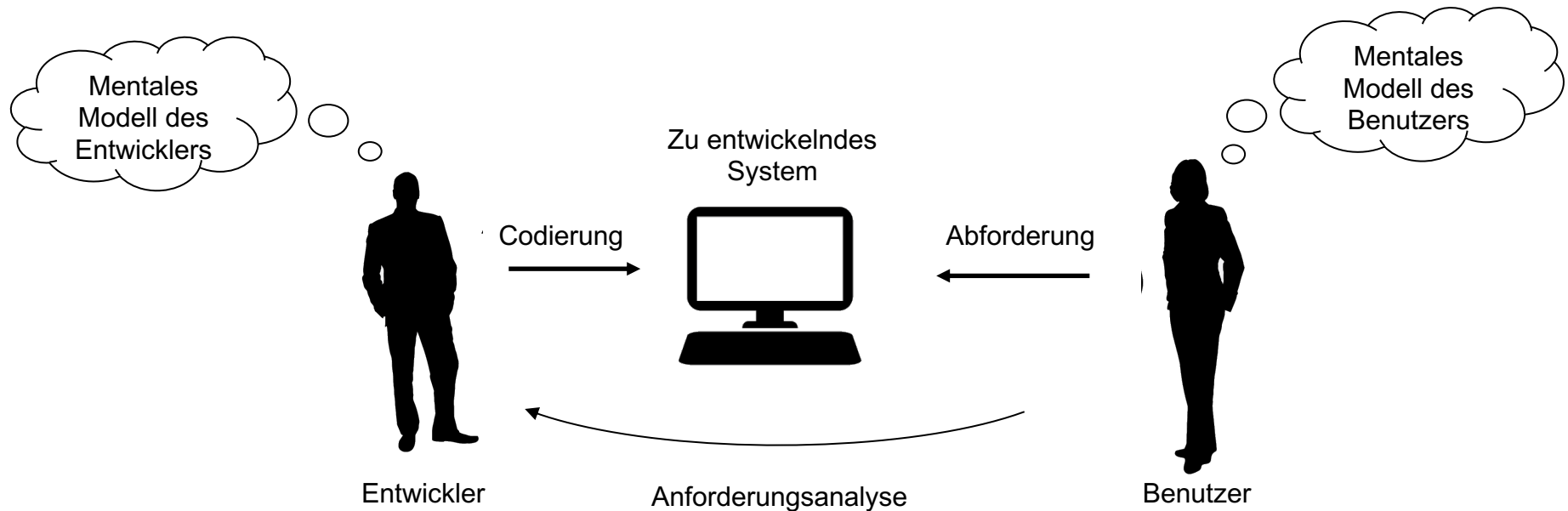
## ERHEBUNG VON ANFORDERUNGEN

- Kreativitätstechniken
  - Brainstorming
  - Methode 6-3-5
  - 6 Hüte Methode
  - Walt Disney Methode
- Beobachtungen
  - Anwender unter realen Bedingungen beobachten
- Befragung
  - Interview
  - Fragebogen
  - Selbstaufschreibung



## SPRACHLICHE BARRIEREN

- Bei der sprachlichen Darstellung von Sachverhalten wird oft unbewusst eine **Transformation** durchgeführt
- Jede Person hat ihr eigenes **mentales Modell**
- Aufgaben von **Analysten** ist es, daraus **objektive** Sachverhalte zu ermitteln



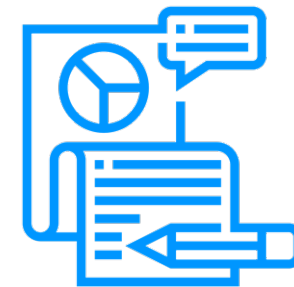
## DOKUMENTATION VON ANFORDERUNGEN

- Diagramme (UML, Use Cases, Prozessdiagramme...)
- Tabellen
- Natürliche Sprache (Prosa, Anforderungsschablone)

→ **Wichtig:** Alle Stakeholder müssen die Anforderungsdokumentation verstehen  
(UML Diagramme sind ggf. nicht jedem bekannt)

→ **Glossar:**

- Definition des fachspezifischen Vokabulars
- Definition von Prozesswörtern (später mehr)
- Definition von Abkürzungen





# KLASSISCHE ANFORDERUNGS- ANALYSE

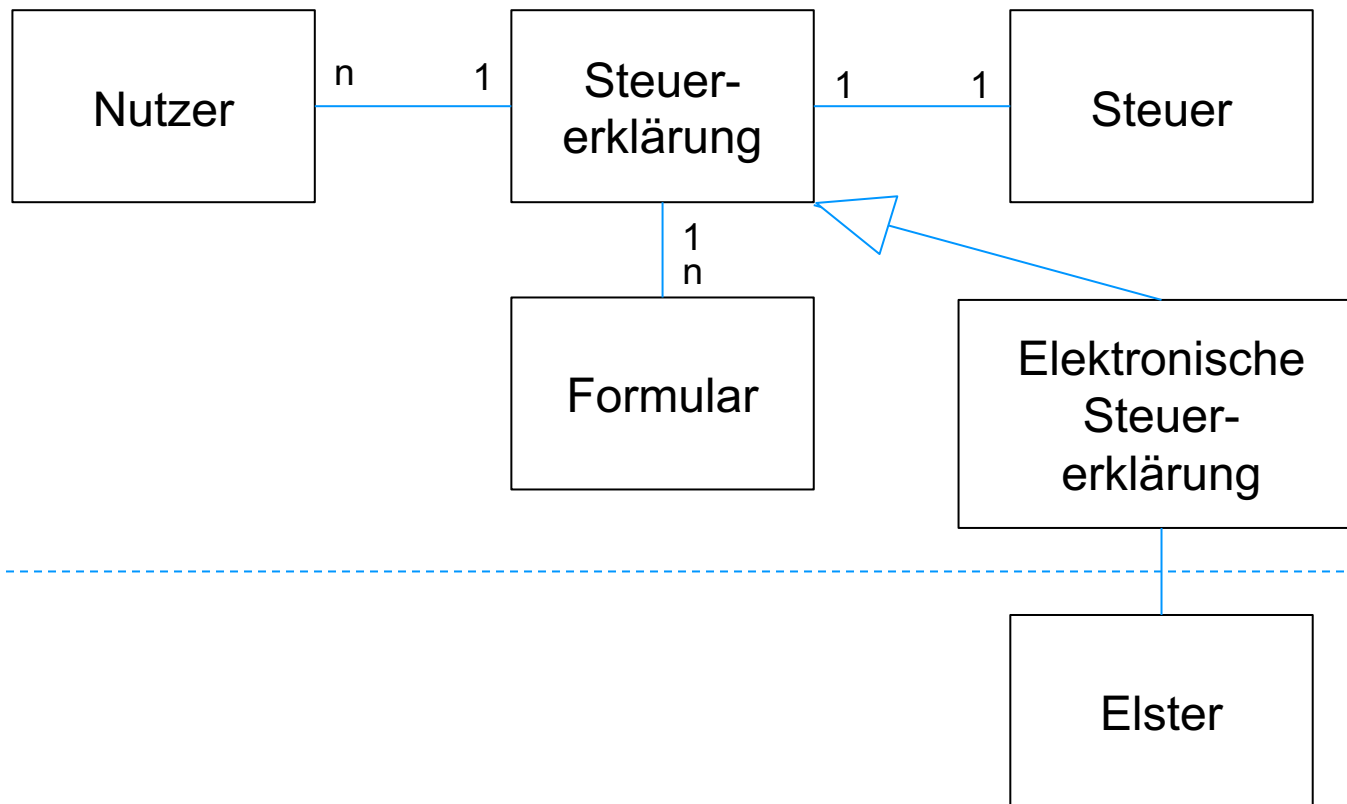
# KLASSISCHE ANFORDERUNGSANALYSE

## Am Beispiel eines Steuerprogramms

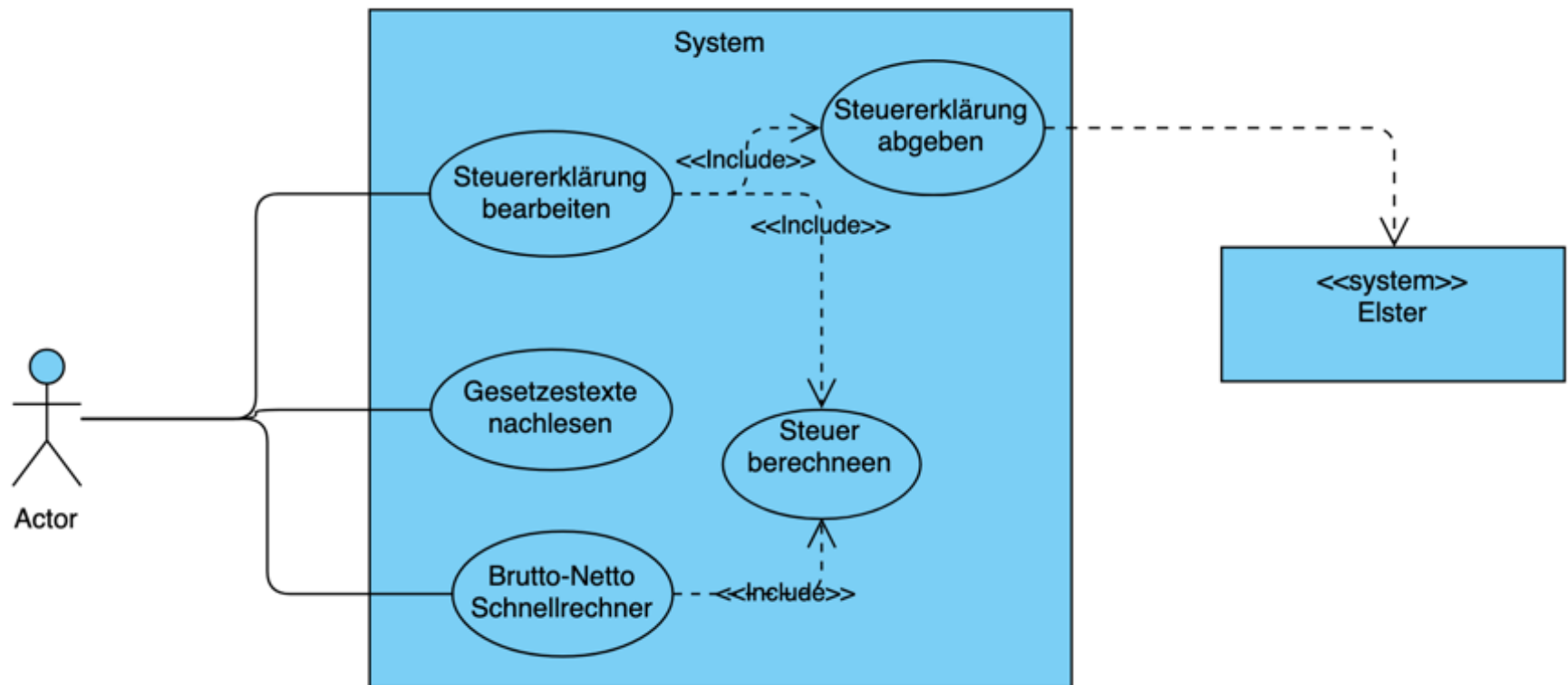
Bereits vorhanden: Lastenheft mit den Kundenanforderungen

1. Fachspezifisches Domänenmodell erstellen
  - Zusammenhänge zwischen realen Objekten
  - Beispielsweise als UML Diagramme
  - Manchmal bereits Teil des Lastenhefts
2. Use Cases Diagramme erstellen
  - Akteure, Systeme und Use Cases erfassen
3. Use Cases verfeinern
  - Use Cases textuell ausformulieren
  - Eventuell als Aktivitätsdiagramme verfeinern
4. Aktivitäten als natürlichsprachliche Anforderungen formulieren
5. Ggf. User Interface definieren

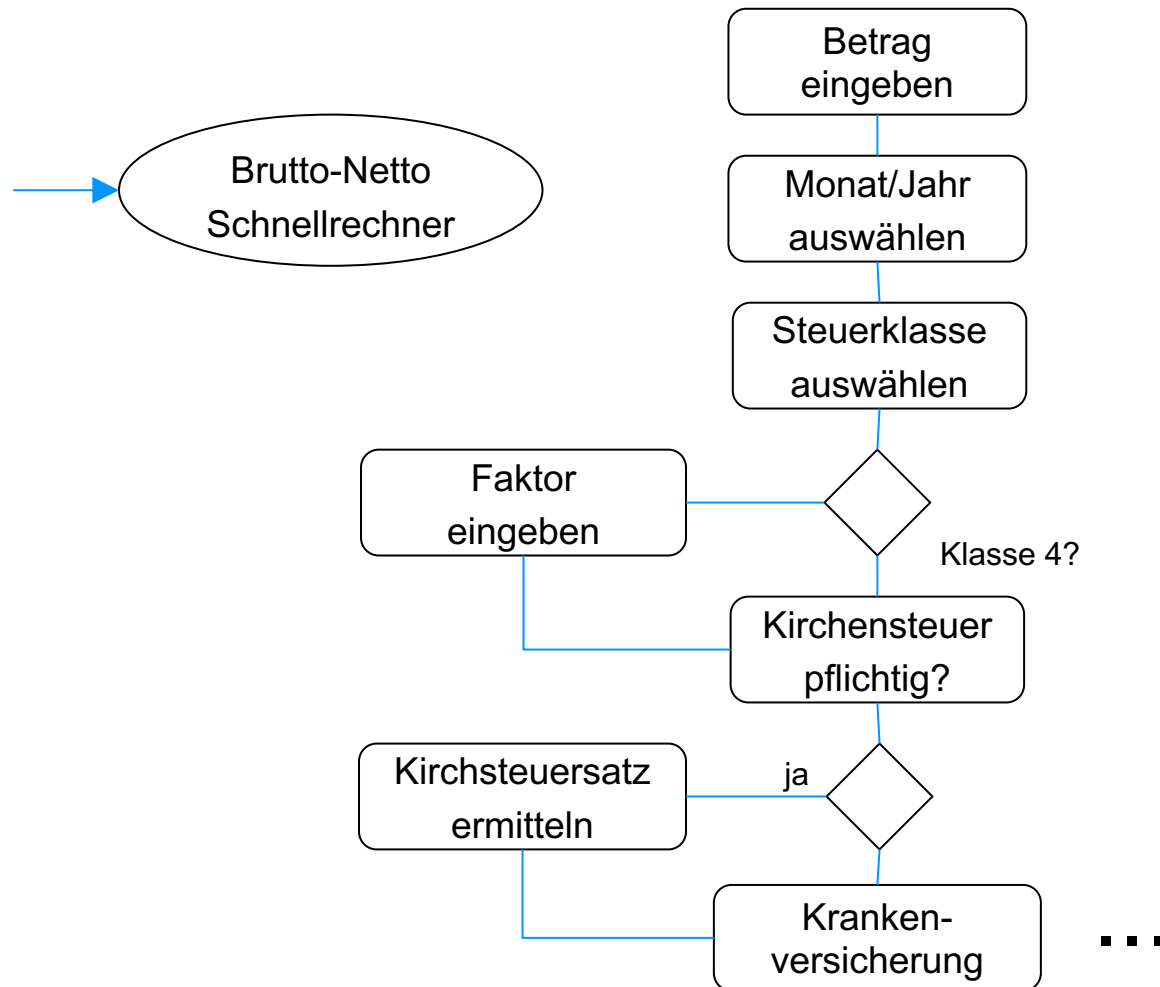
## SCHRITT 1: DOMÄNENMODELL



## SCHRITT 2: USE CASE DIAGRAMM



## SCHRITT 3: USE CASE VERFEINERN (AKTIVITÄTSDIAGRAMM)



## SCHRITT4: NATÜRLICHSPRACHLICHES AUSFORMULIEREN

Anf001: Rechner starten

Sobald der User auf „Starte Brutto-Schnellrechner“ geklickt hat, soll das System Dialog „abc“ anzeigen.

Anf002: Werte eingeben

Sobald Dialog „abc“ angezeigt wird, soll das System dem User die Möglichkeit geben einen Wert in UI Element „xyz“ einzugeben,

Anf003: Auswahl für Monat

Nach der Eingabe von „Beitrag“ soll das System dem User die Möglichkeit bieten „Monat“ im UI Element „...“ auszuwählen“

...

## ANMERKUNGEN

- Use Case Diagramme nicht zu detailliert spezifizieren
  - Änderungen werden dadurch aufwendig und schwierig
  - Es wird immer schwerer Konsistenz zu wahren
  - Später sind meist nur die natürlichsprachlichen Anforderungen relevant
- Nicht zu viele verschiedene Diagrammarten verwenden
- Das beschriebene Verfahren ist nur eines unter vielen zur Erhebung von Anforderungen

# AGILE ANFORDERUNGS- ANALYSE



## USER STORIES (WDHL.)

- In der agilen Welt werden oftmals **Stories** verwendet
- **Nach Mike Cohen (aus User Stories Applied, 2004)**
- |  |  |
|--|--|
| • <i>As a user role</i><br><i>I need functionalitiy</i><br><i>so that I get business value</i> | • <i>Als Anwender mit der Rolle</i><br><i>benötige ich eine Funktionalität</i><br><i>damit ich den Nutzen bekommen</i> |
|--|--|
- Kurze Beschreibung, was das System für den Benutzer tun soll
- Dienen als Diskussionsgrundlage
- Sehr große Stories werden als **Epics** bezeichnet

# EIGENSCHAFTEN VON USER STORIES

## I.N.V.E.S.T

- **I** ndependent
- **N**egotiable
- **V** aluable to the Customer
- **E** stimatable
- **S**mall
- **T**estable

## USER STORIES VS. SOFTWARE-ANFORDERUNG

### Stories

- Sind nicht sehr detailliert
- Dienen als Diskussionsgrundlage mit den Entwicklern
- Werden vom Kunden / ProductOwner erstellt
- Liefern kleine Inkremente an Geschäftswert (vgl. Skateboard-Prinzip)
- Keine Dokumente sondern meist Karten oder Tickets (wenn digital)

### Tasks

- Werden von den Entwicklern im Planning erstellt
- Sind techn. detailliert
- Sind dem Kunden in der Regel unbekannt
- Beschreiben die techn. Maßnahmen, um die Story umzusetzen
- Sind Teile einer Story

## USER STORIES UND USE CASES

Kritik an Stories (nach Alistair Cockburn):

Wenn nur User Stories verwendet werden, geht leicht das „Big Picture“ verloren. Ein genaues Verständnis des Big Pictures (durch Erhebung von Use Cases) fördert das Verständnis der einzelnen Anforderungen und führt zu besseren Abschätzungen.

Erfahrung aus der Praxis:

Stories sind zur Kommunikation mit den Entwicklern und für die Organisieren von Aufgaben geeignet. Der Einsatz von Stories schließt ein vorgelagertes Requirements Engineering nicht aus! Dieser „Fehler“ wird in der Praxis oft begangen!

Agiles Anforderungsmanagement

# ÜBUNG

# Fragen bis hier her?

