



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN  
University of Applied Sciences

# Agiles Requirements Engineering - *Kontext, Stakeholder, Ziele* -

Master Technische Informatik – Embedded Systems–

Prof. Dr.-Ing. Hartmut Schirmacher

Version 2.0  
(SU vom 16.10.2017)

# Inhalt

- Systemkontext, Systemgrenze, Kontextgrenze
- Stakeholder und Personas
- Abgrenzung Anforderung vs. Ziel
- Qualitätskriterien für Ziele
- Attribute zum Dokumentieren von Zielen
- Visualisierung von Stakeholdern und Zielen



# Systemkontext



# Systemkontext

## Definition \*: Systemkontext

[...] der Teil der Umgebung eines Systems, der für die Definition und das Verständnis der Anforderungen des betrachteten Systems relevant ist.

- Wie integriert sich das System in die Realität?
- Betrachtetes System von Umgebung abgrenzen
- Beziehungen System – Umgebung identifizieren
- Materielles und immaterielles beachten



# Was kann zum Kontext gehören?

- Personen (Stakeholder)
- Systeme und Schnittstellen (andere technische Systeme)
- Prozesse (technische oder Geschäftsprozesse)
- Ereignisse (technische oder Geschäftsergebnisse)
- Dokumente (Gesetze, Standards, Dokumentation)



# Beispiel: Personen

Personen, die zum Verständnis der Anforderungen für einen **Bankautomaten** nützlich sein können:

- **Betreiber**, welche Kaufentscheidung für Automaten trifft
- Verschiedene **Benutzer**, die Geld abheben/einzahlen
- **Geldfirmenpersonal**, welches den Automaten befüllt / leert
- **Service-Techniker**, der Störungen beseitigt
- **Werbekunden**, die Werbeeinhalte auf Automat darstellen möchten
- ...



# Beispiel: Prozesse

Prozesse, die zum Verständnis der Anforderungen für einen **Bankautomaten** betrachtet werden könnten:

- Geld **abheben/einzahlen** (privater Kunde)
- Geld abheben/einzahlen (Geschäftskunde)
- **Geldtransport**
- **Abrechnung und Prüfung** der Geldflüsse
- **Einzug und Sperrung** von Karten
- Störungsmeldung und –behebung
- ...



# Beispiel: Schnittstellen

Schnittstellen, die ein Bankautomat haben könnte:

- Anschluss an Stromversorgung + Datenleitung
- **Benutzerschnittstelle** für Endkunden
- Benutzerschnittstelle / **Kommunikationsprotokoll** für Servicetechniker
- Anschluss an **Alarmsystem**
- Benachrichtigung der Geldfirma bzgl. **Füllstand**
- Benachrichtigung der Bank bei **Missbrauch** / **Karteneinzug**
- Upload+Abrechnung von **Werbematerial**,  
Analyse der Werbenutzung
- ...





# Systemgrenze (Scope) ermitteln

## Definition \*: Systemgrenze

[...] separiert das geplante System von seiner Umgebung. Sie grenzt den im Rahmen des Entwicklungsprozesses gestaltbaren und veränderbaren Teil der Realität von den Aspekten in der Umgebung ab, die durch den Entwicklungsprozess nicht verändert werden können.

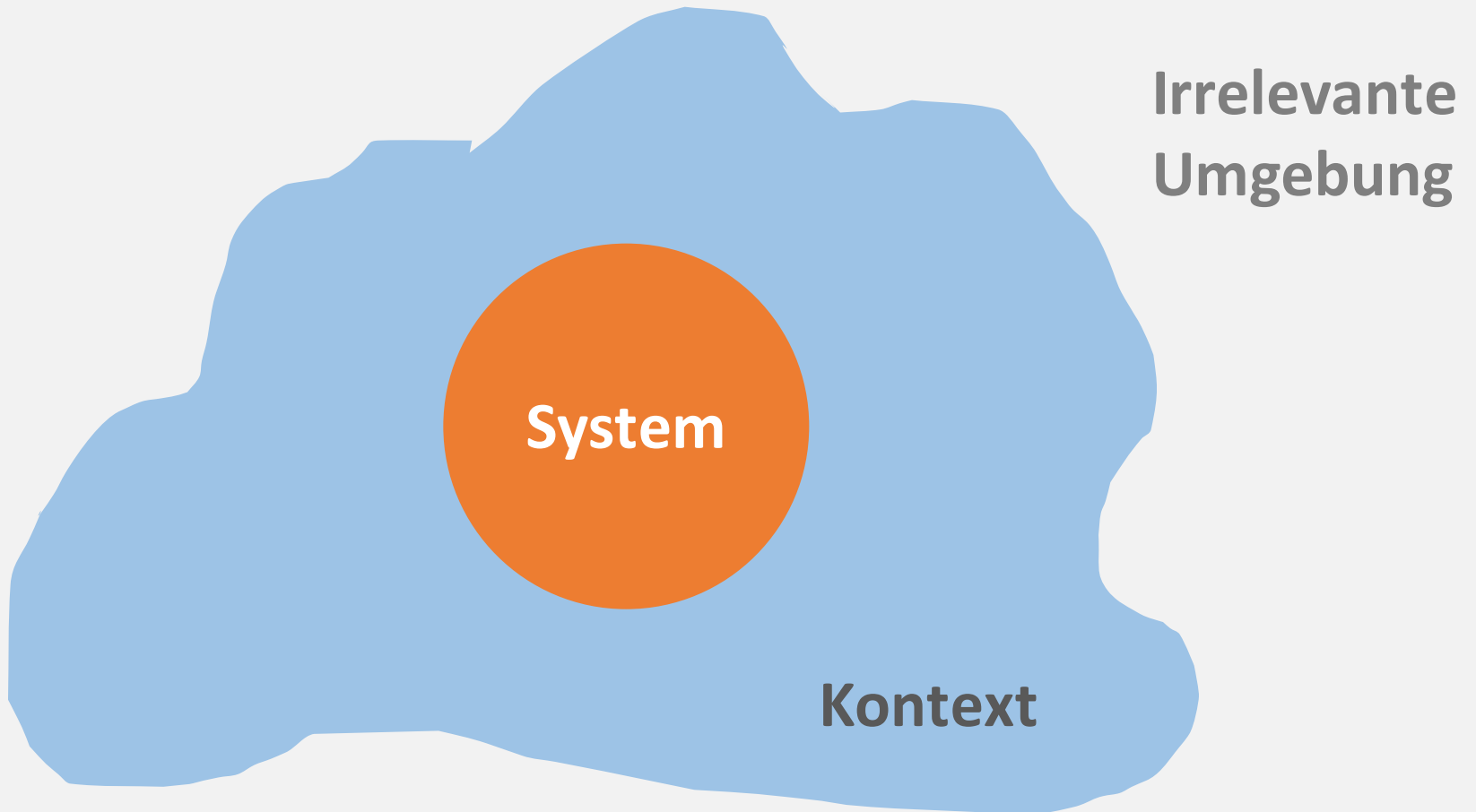
- Alles, was potentiell verändert werden kann/soll
- Auch z.B. Geschäftsprozesse und Infrastruktur

Wie und wann finde ich die Systemgrenze?

- Quellen und Senken betrachten → Schnittstellen des Systems nach außen
- Kann sich im Lauf der Priorisierung und Entwicklung verschieben (neue Kenntnisse, Entscheidungen, ...)



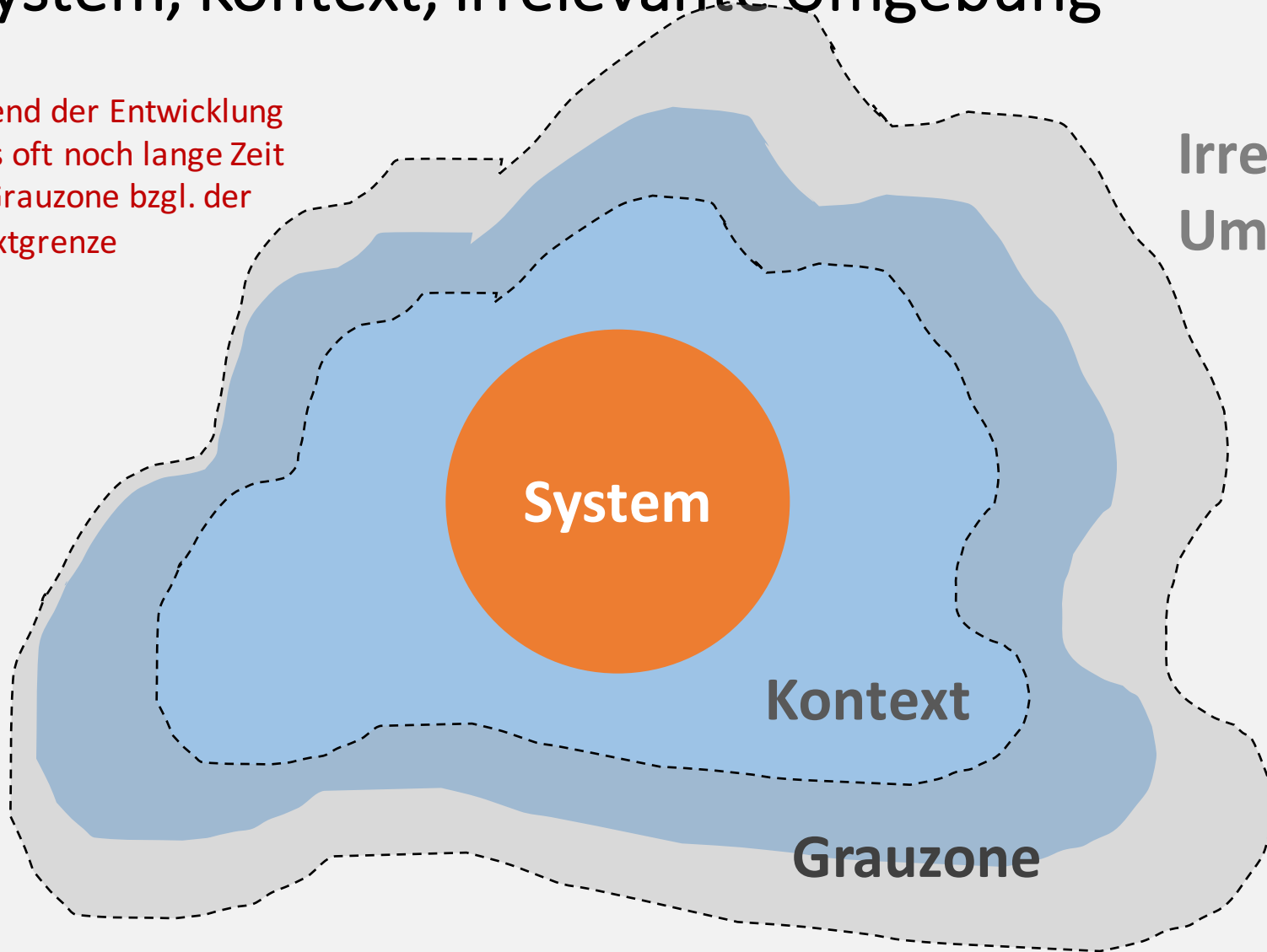
# System, Kontext, irrelevante Umgebung



# System, Kontext, irrelevante Umgebung

Während der Entwicklung  
gibt es oft noch lange Zeit  
eine Grauzone bzgl. der  
Kontextgrenze

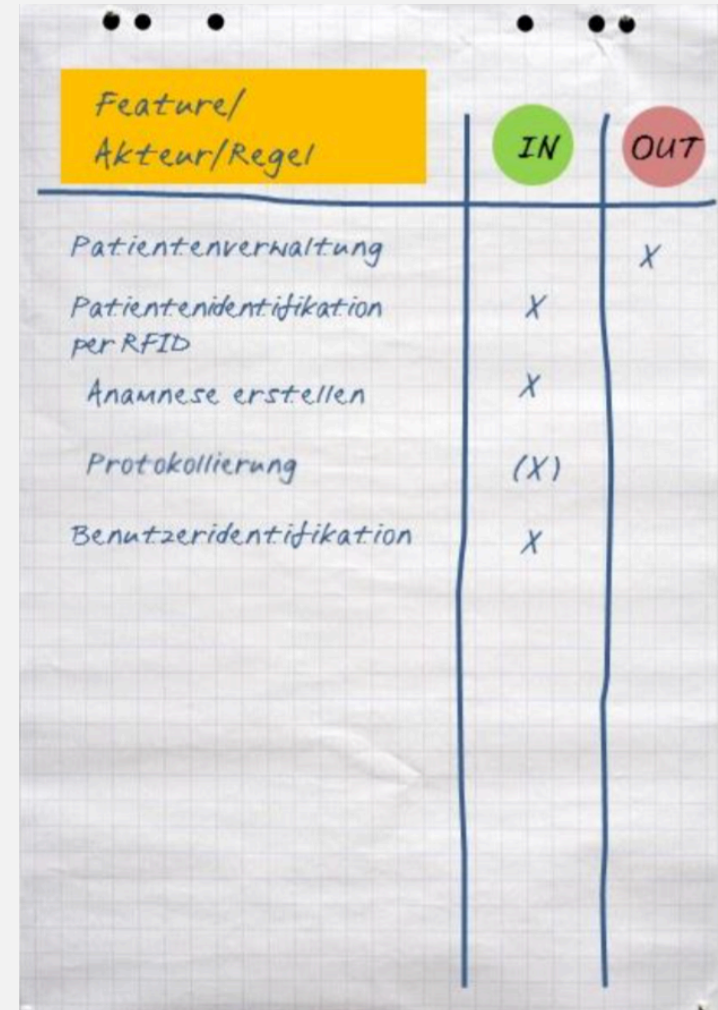
Irrelevante  
Umgebung



# Ermittlung des Systemkontextes

Einmal durch den gesamten Lebenszyklus des Produkts durchdenken, von Inbetriebnahme bis Außerbetriebnahme.

- Stakeholder identifizieren
- Schnittstellen identifizieren
  - zu den angrenzenden Systemen
  - zu den Benutzern
- Informationen zu vergleichbaren Produkten sichten
- Normen und Standards recherchieren
- Informelle In-/Out Liste anlegen
- ... und immer wieder kritisch betrachten und ggf. verändern



Feature/ Akteur/Regel	IN	OUT
Patientenverwaltung		X
Patientenidentifikation per RFID	X	
Anamnese erstellen	X	
Protokollierung	(X)	
Benutzeridentifikation	X	



# Dokumentation des Systemkontextes

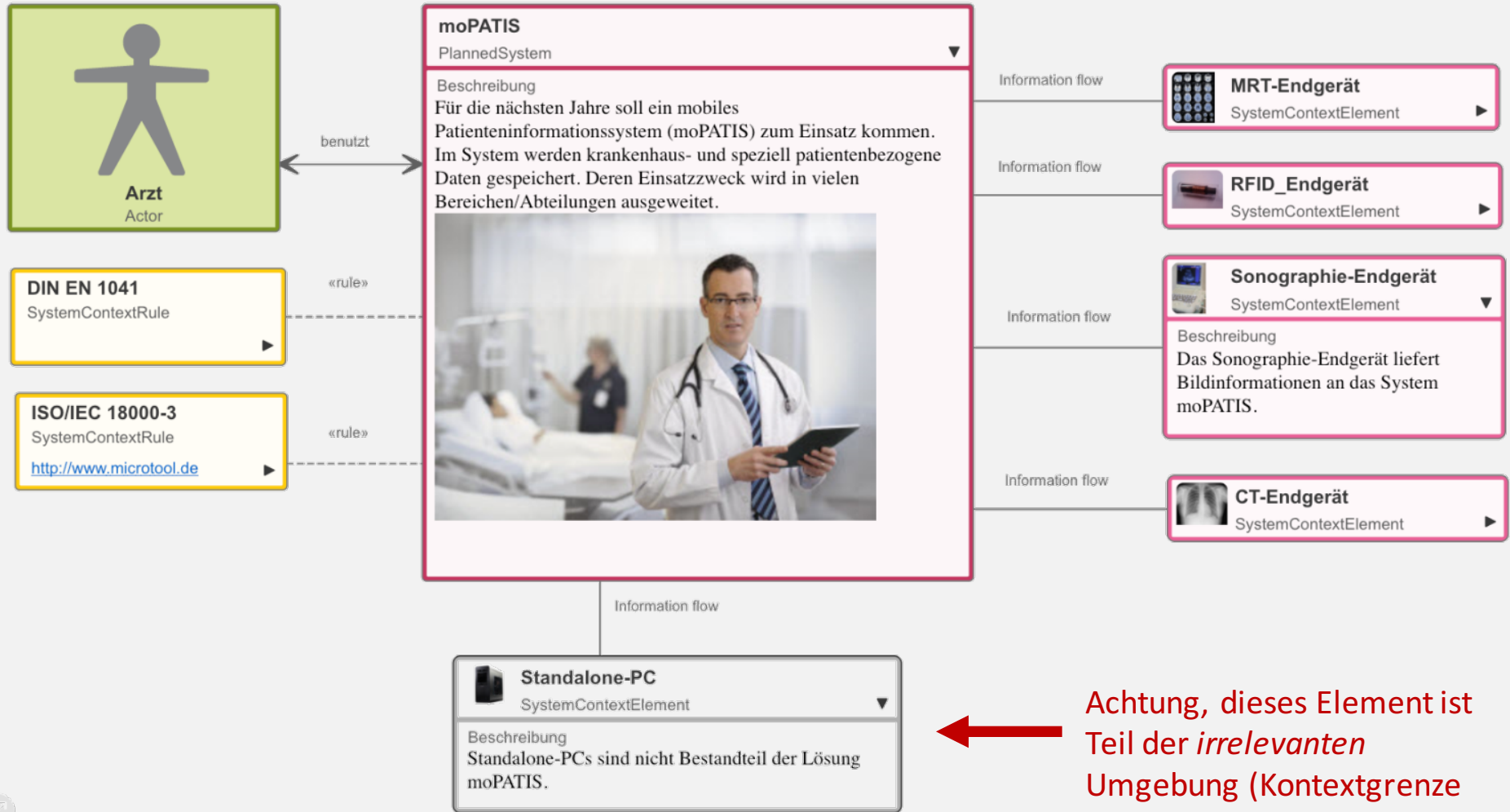
- Use Case Diagramm
  - Akteure und Prozesse
  - System- und Kontextgrenzen
- Datenflussdiagramm
  - Quellen und Senken in der Umgebung des Systems
- Mischformen / freie Diagramme

## Hauptsache:

- sich trauen, es zu dokumentieren
- klare Eingrenzung schaffen



# Systemkontextdiagramm in objectiF RPM



**Stakeholder**



# Stakeholder

## Definition \*: Stakeholder

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

Zum Beispiel

- Angedachte Nutzer des Systems
- Kunden / Käufer, Betreiber
- Internes Management
- Dienstleister, Zulieferer
- Entwickler und Tester





# Identifikation der Stakeholder

Stakeholder sind i.d.R. die wichtigste Quelle für Anforderungen

- Aufdecken von Widersprüchen (Zielkonflikte)
- Alle Stakeholder müssen in den Anforderungen adäquat vertreten sein (auch die Entwicklung!)

Wie finde ich sie?

- Fachexperten fragen, Stakeholder fragen
- Informationen zu vergleichbaren Produkte anschauen
- Sich den Einsatz des Produktes konkret vorstellen
- Sich das Entwickeln und Testen des Produkts konkret vorstellen



# Dokumentation der Stakeholder

- Zentrale Stakeholder-Liste
- Mindestanforderungen / sinnvoll:
  - Name
  - Funktion / Rolle
  - Wissensgebiet
  - Relevanz
  - Kontaktdaten
  - Verfügbarkeit (zeitlich und räumlich innerhalb des Projekts)
- Dann erweitern um die spezifischen Ziele / Interessen



# Rolle des Stakeholders

- Führt den RE-Eng. in sein Fachgebiet ein
  - Zeitaufwand, Kommunikationsfähigkeit
- Lieferung von Anforderungen
  - Gewissenhaft, zielgerecht
- Prüfung von Anforderungen
  - Dokumentiert oder undokumentiert
- Rolle in Entscheidungen
  - Entscheider
  - Unterstützer
- Respektieren von Einschätzungen des RE-Eng.
  - z.B. bzgl. Kosten und Machbarkeit



Stakeholder nicht  
verfügbar?



# Personas

- Persona = Methode aus dem Usability Engineering
- Schaffung fiktiver Anwender / Stakeholder
- Stellvertreter für verschiedene Typen
- Plastisch, Team muss sich hineindenken können
  - Mit Namen, Foto, Hintergrundbeschreibung...
- Entwurf anhand von Merkmalen, die bei Befragungen etc. aufgetaucht sind
- So viele wie nötig, um die Anforderungen zu beschreiben



# Dokumentation von Personas

- Vor- und Nachname (fiktiv)
- Foto (um die Persona noch vorstellbarer zu gestalten)

Typische relevante Attribute:

- Tätigkeit/Arbeitsrolle
- Alter
- Ziele / Wünsche / Erwartungen, Ängste
- Ausbildung/Wissen, Computer-Kenntnisse
- Einstellung zu Produkt und seiner Technologie
- Einschränkungen
- ...



# Persona vs. Stakeholder

Wenn eine Gelegenheit da ist, Annahmen über Persona anhand konkreter Stakeholder prüfen!

Achtung: nicht abstrahieren, sondern konkret bleiben!

Abstraktion findet dann auf der Ebene von Zielen und Anforderungen statt.



**Ziele**





# Ziele

- Anforderungen beziehen sich (bereits) auf das System
- Doch am Anfang: volle Konzentration auf den Stakeholder
- Was will Stakeholder X eigentlich (wirklich) erreichen?  
(unabhängig davon, wie die Lösung aussieht)
- **Ziele = Schlüssel zu erwartungskonformer Lösung**

Aber Vorsicht, nicht alles ist ein brauchbares Ziel:

- *Ich möchte damit die Weltherrschaft erringen...*
- *Es muss einfach besser funktionieren als das jetzige System!*



# Ziele im Projektmanagement: SMART \*

- **Spezifisch** (specific)
- **Messbar** (measurable)
- **Ansprechend** (accepted)
- **Realistisch** (realistic)
- **Terminiert** (time bound)

SMART wurde für das Projektmanagement entwickelt, allerdings für die Formulierung von Zielen für das Projektteam. Die Ziele sollten u.a. motivierend für die Mitarbeiter sein.

Diese Kriterien werden oft verwendet, eignen sich allerdings nicht 100% für Stakeholder-Ziele im RE.

\*) G.T. Doran 1981, siehe [https://de.wikipedia.org/wiki/SMART\\_\(Projektmanagement\)](https://de.wikipedia.org/wiki/SMART_(Projektmanagement))



# Stakeholder-Ziele: **ENDPUNKT**\*

- **Erreichbar** (Rahmenbedingungen, Budget)
- **Nachweisbar** (Akzeptanzkriterien)
- **Detailliert** (anschaulich formulierten)
- **Positiv** (nicht durch Verneinung formulieren)
- **Umwelttauglich** (Auswirkung auf Umfeld beachten)
- **Nützlich** (für den Stakeholder)
- **Korrekt** (Wortlaut, Stakeholder muss prüfen)
- **Toll** 😊 (motivierend für den Stakeholder)

Gute Qualitätskriterien für Prüfung von Stakeholder-Zielen im RE.



# Beispiele?



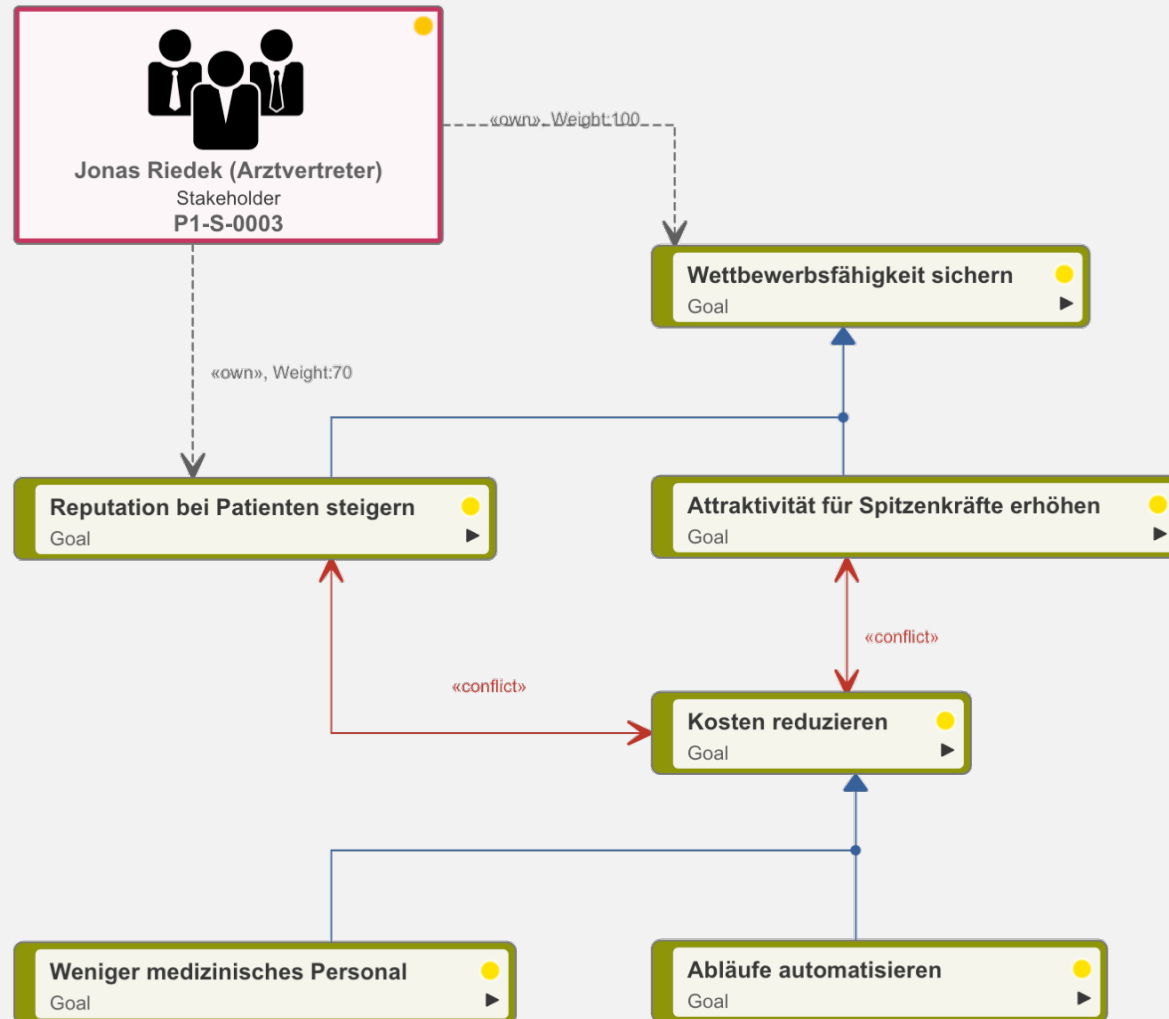
# Wichtige Attribute von Zielen

- **primäre Quelle**, aus der das Ziel stammt,
- alle **Stakeholder**, für die das Ziel von Bedeutung ist
- **Priorität** für dieses Ziel (nach Analyse)
- **Kritikalität** (Bedeutung für Erfolg des Systems)
- **Effekt** auf die Stakeholder bei Erreichen/Nicht-Erreichen

Es ist empfehlenswert, diese Attribute für jede Ziel zu erfassen.



# Zieldiagramm eines Stakeholders in objectiF RPM



# Zielkonflikte

- Sich derer bewusst sein
  - Konflikte sind wichtig und häufig
  - OK, solange sich die Ziele nicht komplett ausschließen
  - Gewichte für die Priorisierung kennen
- Wenn möglich auflösen
  - Notwendige Entscheidungen frühzeitig treffen
  - I.d.R. Management / Stakeholder dazu notwendig

→ Vermeidung enormer Probleme in der späteren Entwicklung



# Ziele – und dann?

- Aus den Ziele die tatsächlichen Anforderungen entwickeln
- Jede äußere Anforderung möglichst auf ein Ziel zurückführen  
→ Tracability, Validierung
- Anforderungen (und Ziele) verfeinern, bis sie nützliche Artefakte im Sinne der Kommunikation und Produktentwicklung sind.





Fragen?

