

# Übungsgruppen

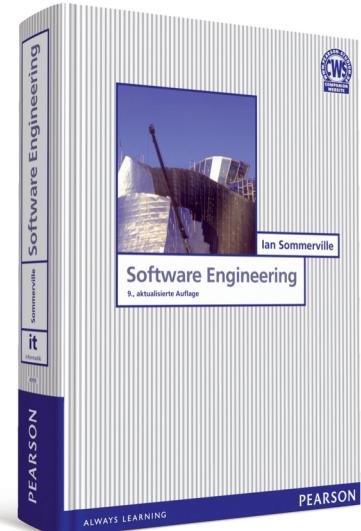
# Übungsgruppen

- Anmeldung:
  - Via AsSESS
  - <http://ess.cs.uni-dortmund.de/ASSESS/index.php?do=lecturelist>
- Platzvergabe:
  - Hat stattgefunden
  - Jeder sollte seinen Platz haben
- Spätere Anmeldungen: *first come first serve*

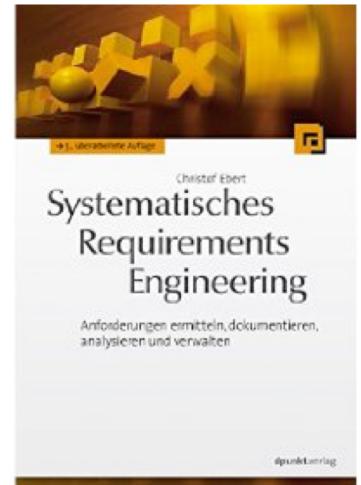
# Teil 1.2:

# Anforderungen

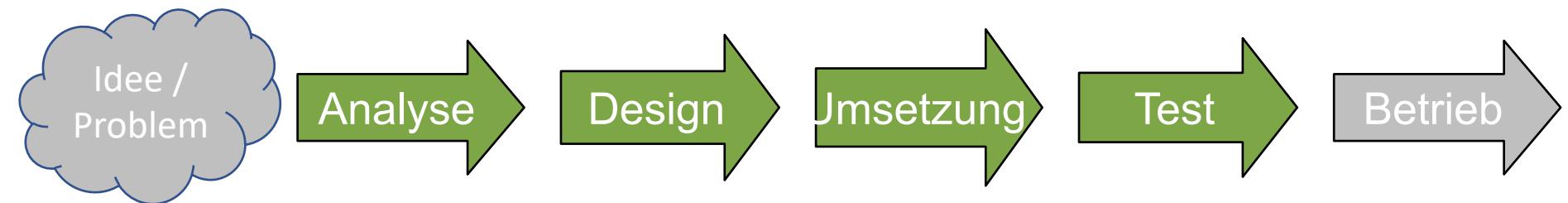
# Literatur



- Ian Sommerville, Software Engineering, 9. überarbeitete Ausgabe, Pearson 2012.
- Systematisches Requirements Engineering: Anforderungen ermitteln, dokumentieren, analysieren und verwalten, 5., überarb. Auflage, dpunkt.verlag GmbH, 2014.



# Einordnung



# Agenda

- Motivation und Beispiel
- Typen und Eigenschaften von Anforderungen
- Spezifikation von Anforderungen
- Erheben und Management von Anforderungen

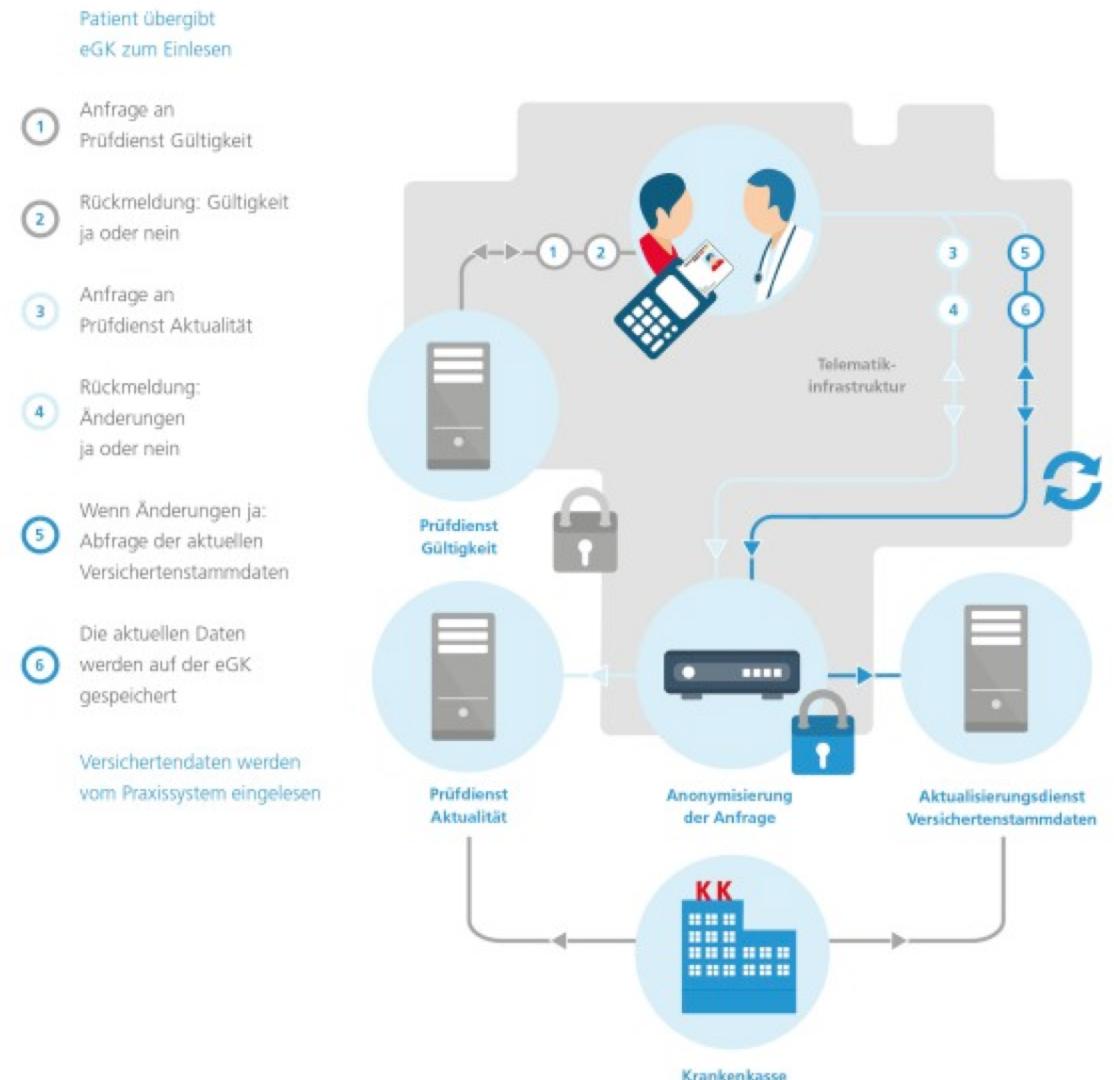
# Beispiel: Elektronische Gesundheitskarte

- Ersetzt seit Januar 2014 Krankenversicherungskarte
- Einführung / Entwicklung seit 2005
- Viele gesetzliche Vorgaben,
- Viele beteiligt am Prozess



- Zentrale Verantwortung: Fa. Gematik GmbH gegründet von Spitzenverbänden der Leistungserbringer und Kostenträger
- Ziel: Die Einführung einer elektronischen Gesundheitskarte (eGK) und der Telematikinfrastruktur zur Modernisierung des Gesundheitswesens, in dem der Patient im Mittelpunkt steht.

# Versichertenstammdatenmanagement (I)



## Versichertenstammdatenmanagement (VSDM).

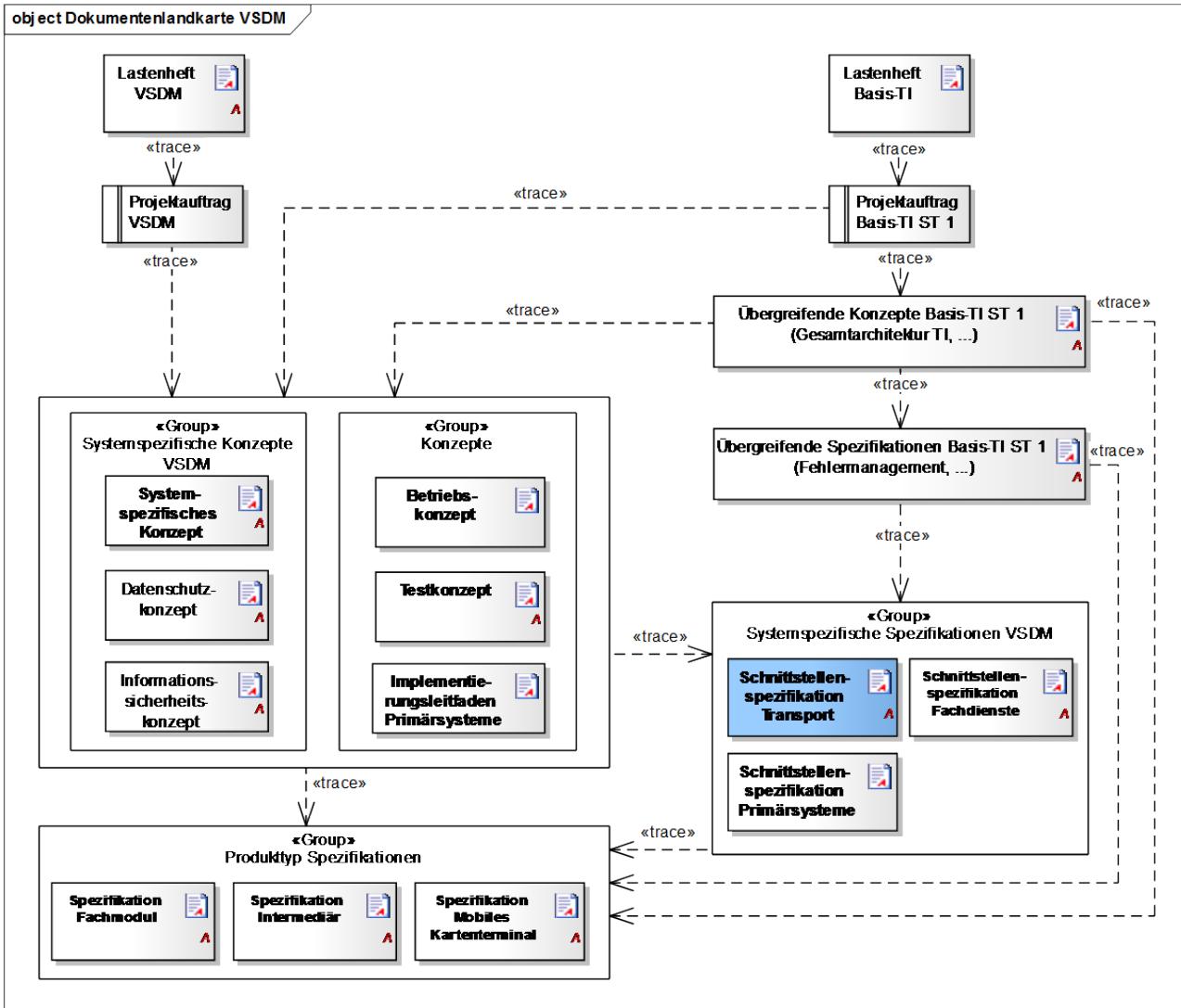
- Verarbeitung und Nutzung der Versichertenstammdaten
- Neues Feature: Stammdaten aktualisierbar aus Arztpraxen.

Bild: [www.gematik.de](http://www.gematik.de)

# Versichertenstammdatenmanagement (II)

**Spezifikation und Informationen bei: [www.gematik.de](http://www.gematik.de)**

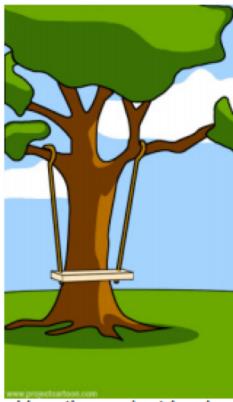
Ausschnitt Dokumentenhierarchie



# Motivation



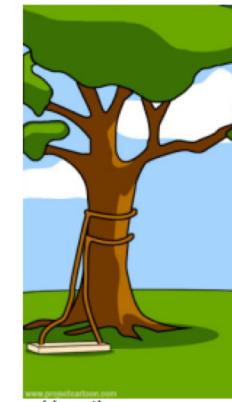
How the customer explained it



How the project leader understood it



How the analyst designed it



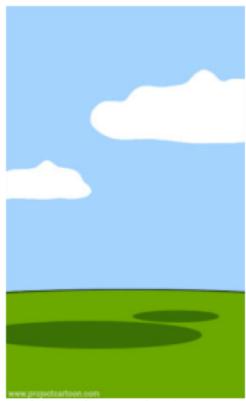
How the programmer wrote it



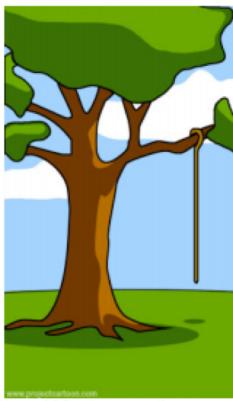
What the beta testers received



How the business consultant described it



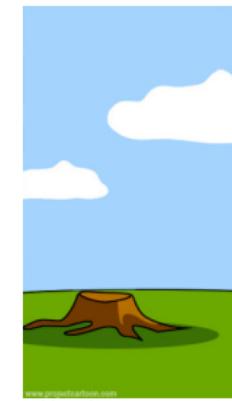
How the project was documented



What operations installed



How the customer was billed



How it was supported

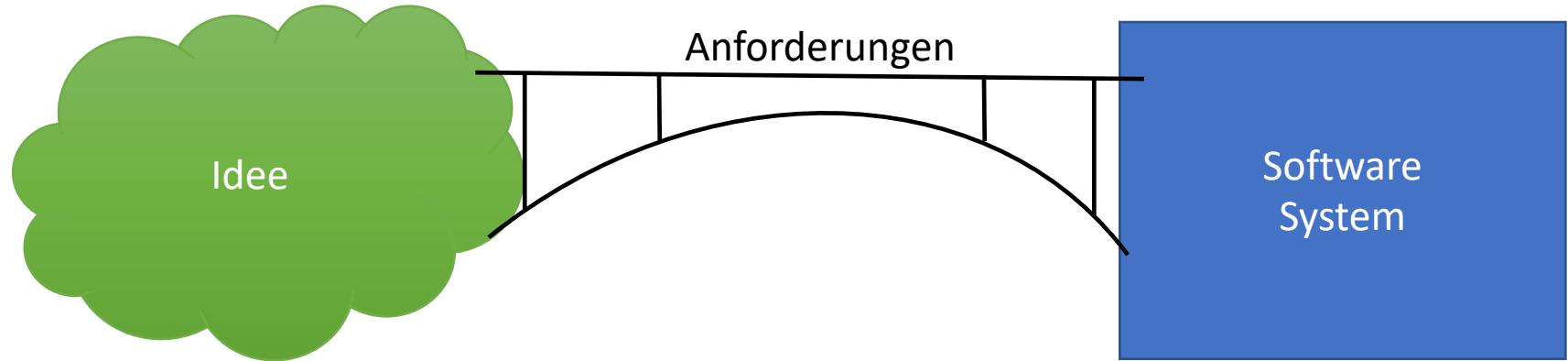


iSwing  
What marketing advertised



What the customer really needed

# Wofür braucht man Anforderungen



Gemeinsames und dokumentiertes Verständnis der Erwartungen aller Betroffener an ein System

# Projektbetroffene (Stakeholder)

Einzelpersonen und Organisationen, die aktiv an einem Projekt beteiligt sind oder deren Interessen als Folge der Projektdurchführung oder des Projektabschlusses positiv oder negativ beeinflusst werden können.

(vgl. PMBOK)

## Bei Softwareprojekten u. A.:

- die Entwickler
- der Kunde
- die Abteilung, die das Produkt später übernehmen und betreiben wird
- das Management

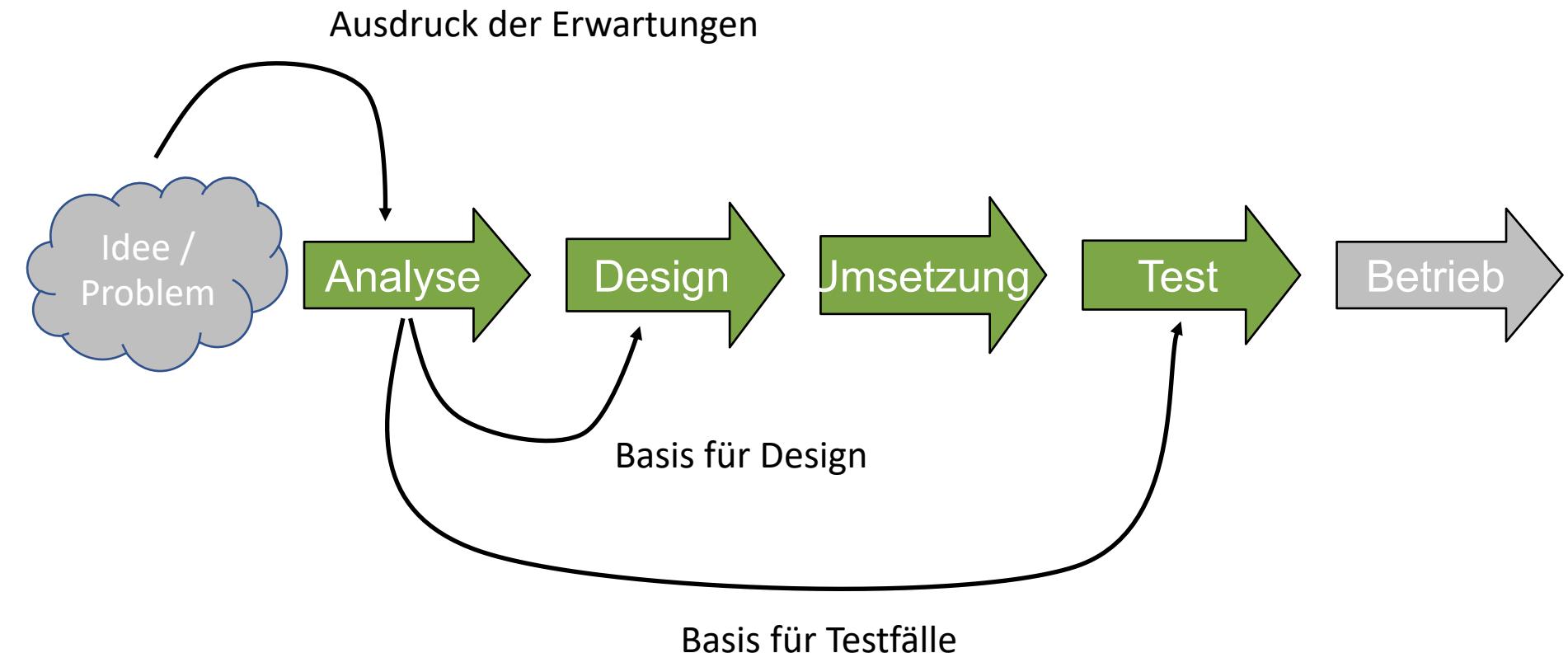
## Beispiel VSDM:

Das Dokument richtet sich neben Personengruppen, die grundsätzlich an den Verfahren des Versichertenstammdatenmanagements interessiert sind, an

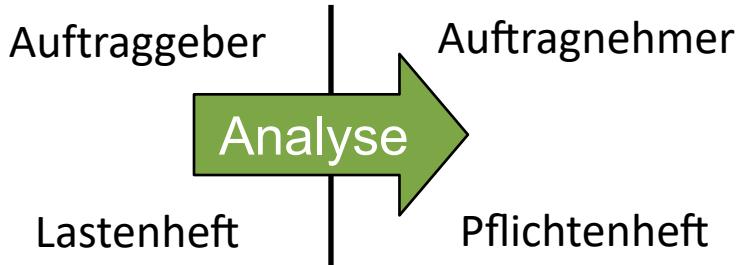
- Konnektorhersteller
- Primärsystemhersteller
- VSDD-Entwickler
- Entwickler von Brokern

Aus: gematik\_VSD\_Facharchitektur\_VSDM.doc

# Anforderungen im Entwicklungsprozess

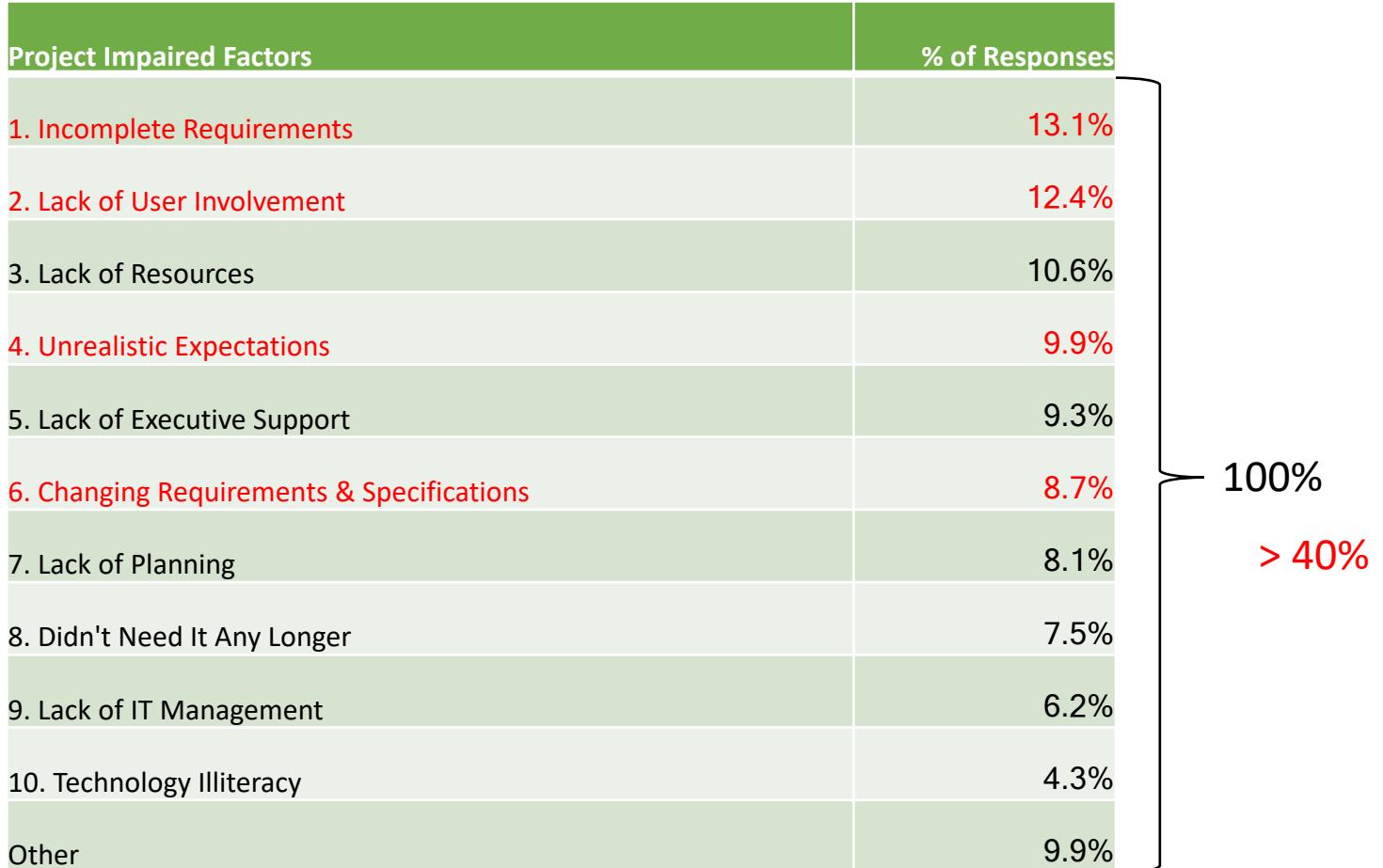


# Anforderungen in großen Projekten



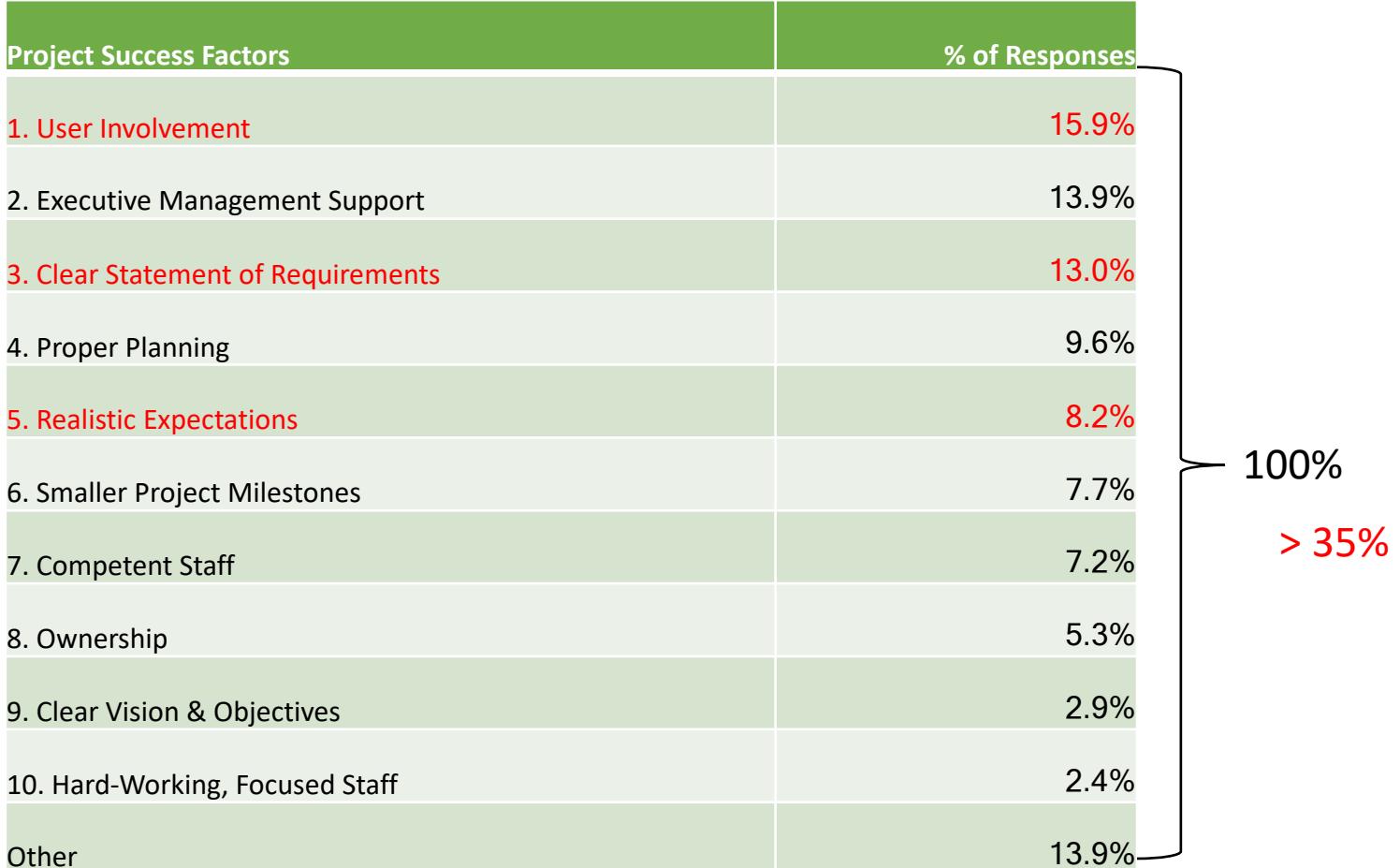
- Lastenheft: Anforderungen des Kunden
- Pflichtenheft: Beschreibung des zu entwickelnden Systems durch potentiellen Auftragnehmer
  - Basis für Verträge über Software
  - Alle Anforderungen müssen überprüfbar sein
  - Ein objektiver Dritter muss entscheiden können
  - Notfalls Abnahme vor Gericht durchsetzen

# Projekte scheitern wegen fehlender / schlechter Anforderungen



The Standish Group, CHAOS Report 2014

# Erfolgsfaktoren Projekte



The Standish Group, CHAOS Report 2014

# Agenda

- Motivation und Beispiel
- Typen und Eigenschaften von Anforderungen
- Spezifikation von Anforderungen
- Erheben und Management von Anforderungen

# Anforderungen

## Definition (IEEE):

- requirement.*** (1) A condition or capability needed by a user to solve a problem or achieve an objective.
- (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.
- (3) A documented representation of a condition or capability as in (1) or (2).

IEEE Standard Glossary of Software Engineering Terminology

IEEE Std 610.12-1990

# Funktionale vs. Nicht-Funktionale Anforderungen

## Funktional

- **Was** soll das System leisten?
- **Benutzer:** Welche Dienste soll es anbieten?
  - Für Dienste
    - Eingaben, Ausgaben
    - Verhalten
  - Was soll es explizit nicht tun?
- **System:** Wie realisiert das System Benutzeranforderungen

## Nicht Funktional

- **Wie** soll das System erstellt werden
- **Unternehmensanforderungen**
  - Anforderungen an Entwicklungsprozess
  - ...
- **Produktanforderungen**
  - Performanz
  - ...
- **Externe Anforderungen**
  - Rechtliche Rahmenbedingungen
  - ...

# Benutzer / System Anforderungen

## Benutzeranforderungen (=> Lastenheft)

Beschreibung der Dienste, die eine Software leisten soll, und der Randbedingungen, unter denen sie betrieben wird (*aber nicht wie Dienste realisiert werden sollen*).

## Systemanforderungen (=> Pflichtenheft)

Definieren, was bzw. wie etwas implementiert werden soll.

Vgl. Ausschreibung  
(Auftraggeber Sicht)

Vgl. Angebot  
(Auftragnehmer Sicht)

# Beispiele: Benutzer / System Anforderungen (I)

## Benutzeranforderung (Außensicht)

VSDM- A\_2146: Die Fachanwendung VSDM MUSS die **Berechtigungen der Akteure** entsprechend der Tabelle Tab\_VSDM\_SysL\_28 "fachliche Berechtigungsmatrix VSDM" umsetzen.

## Systemanforderung (Konkreter Bezug zu System)

VSDM- A\_2971: Die eGK MUSS den **Zugriffsschutz für die Container EF.PD, EF.VD, EF.Prüfungsnachweis und EF.StatusVD** so einstellen, dass sie jederzeit ohne Authentisierung lesbar sind.

# Beispiele: Benutzer / System Anforderungen (II)

## Definition einer Benutzeranforderung

1. Das MHC-PMS soll monatliche Verwaltungsberichte erzeugen, welche die Kosten der verschriebenen Medikamente angeben, die in diesem Monat in jeder Sprechstunde angefallen sind.

## Spezifikation der Systemanforderungen

- 1.1 Am letzten Arbeitstag jeden Monats soll eine Zusammenstellung über alle verschriebenen Medikamente, ihre Kosten und die Sprechstunde, die jeweils die Verschreibung ausgestellt hat, erzeugt werden.
- 1.2 Das System soll automatisch den Bericht um 17.30 Uhr des letzten Arbeitstags im Monat zum Drucken generieren.
- 1.3 Es soll ein Bericht für jede Sprechstunde erzeugt werden, in dem die einzelnen Medikamentennamen, die Gesamtzahl der Verschreibungen, die Anzahl der verschriebenen Dosierungen und die Gesamtkosten der verschriebenen Medikamente aufgelistet werden.
- 1.4 Falls Medikamente in unterschiedlichen Dosierungen (z.B. 10 mg, 20 mg) verfügbar sind, sollen getrennte Berichte für jede Dosierung erzeugt werden.
- 1.5 Der Zugang zu allen Kostenberichten soll auf autorisierte Nutzer beschränkt werden, die in einer Zugangskontrollliste der Verwaltung aufgeführt sind.

# Beispiele: Benutzer / System Anforderungen (III)

- Systeminteraktionen, die dem Nutzer ermöglicht werden:

ID	Titel	Text	Art	Quelle
A_01910	Anzeige komplette VSD incl. manuell angestoßener optionale Aktualisierung - Akteure	Die kompletten VSD MÜSSEN durch folgende Akteure angezeigt und (optional durch Akteur ausgelöst) aktualisiert werden können: - Akteur Arzt - Akteur Mitarbeiter medizinische Institution	F	20071004_BMG KommentareDars tellungZugriffsrec hte_anGem.doc

# Beispiele: Benutzer / System Anforderungen (IV)

- Aktionen, die vom System ausgeführt werden sollen:

Kurzbeschreibung	Auslesen der VSD aus der eGK	
Initiiерender Akteur	Primärsysteme (z. B. Praxisverwaltungs- und Krankenhausinformationssysteme)	
Weitere Akteure	Konnektor, eGK, SMC-A/B / HBA	
Auslöser	Die Versichertenstammdaten sollen von der eGK abgerufen bzw. auf der eGK aktualisiert werden	
Vorbedingungen	Technische Vorbedingungen: Alle lokalen Komponenten sind betriebsbereit und in fehlerfreiem Zustand. Die eGK ist gesteckt und auf ihre technische Funktionsfähigkeit geprüft. Die SMC/HBA ist freigeschaltet.	
Nachbedingungen	VSD liegen dem Primärsystem vor.	
Aufruf	ReadVSD	Aufruf vom Primärsystem an den Konnektor. Als Ergebnis werden entweder die VSD_eGK und optional eine Quittung (nur im Online-Fall bei einer Update-Abfrage) oder eine Fehlermeldung erwartet. Parameter: P1: eGK P2: SMC-A/B-Handle / HBA-Handle P3: Update
Step	Aktion	Kommentar
TUC_KON_038 TUC_KON_018	Fachliche Vorbedingungen auswerten	Durch den Konnektor werden die fachlichen Vorbedingungen geprüft. Ist eine der Bedingungen nicht erfüllt, wird der gesamte Use Case mit einer Fehlermeldung des Konnektors abgebrochen. Mögliche Konnektor-Fehlermeldungen sind unter anderen: 1. Zertifikatsprüfung fehlgeschlagen 2. HCA Container gesperrt Bezuglich der konkreten Fehlermeldungen und Fehlercodes sei auf [gemSpec_Kon] verwiesen. Die vorher ausgeführten Use Cases TUC_KON_038 „eGK-Auth-Zertifikat prüfen und TUC_KON_018 „eGK-Sperrung prüfen“ müssen ergeben, dass die eGK nicht gesperrt ist.

# Nicht-funktionale Anforderungen

- Nicht-funktionale Anforderungen umfassen Qualitätsanforderungen an Produkt und Prozess.



# Beispiele für Nicht-funktionale Anforderungen (I)

- Performance

*“Das System muss jede Anfrage des Benutzers innerhalb von 30 Sekunden ausführen (auf System XY).”*

*“Der Speicherbedarf darf 512MB nicht übersteigen.”*

- Zuverlässigkeit

*“Die Verfügbarkeit des Systems muss bei 99.999 % liegen.”*

- Benutzbarkeit:

*„Das System muss die gespeicherten Objekte formatiert ausgeben können nach Schema XXXX.“*

# Beispiele für Nicht-funktionale Anforderungen (II)

- Anforderungen an den Entwicklungsprozess:

*„Der Entwickler muss mit dem Kunden monatliche Reviews der zu erstellenden Dokumente durchführen.“*

- Umgebungsanforderungen:

*„Das System muss mit Java entwickelt werden und muss in der Sun Java VM Version XY laufen.“*

# Beispiele für Nicht-funktionale Anforderungen (III)

Regulatorische Anforderungen:

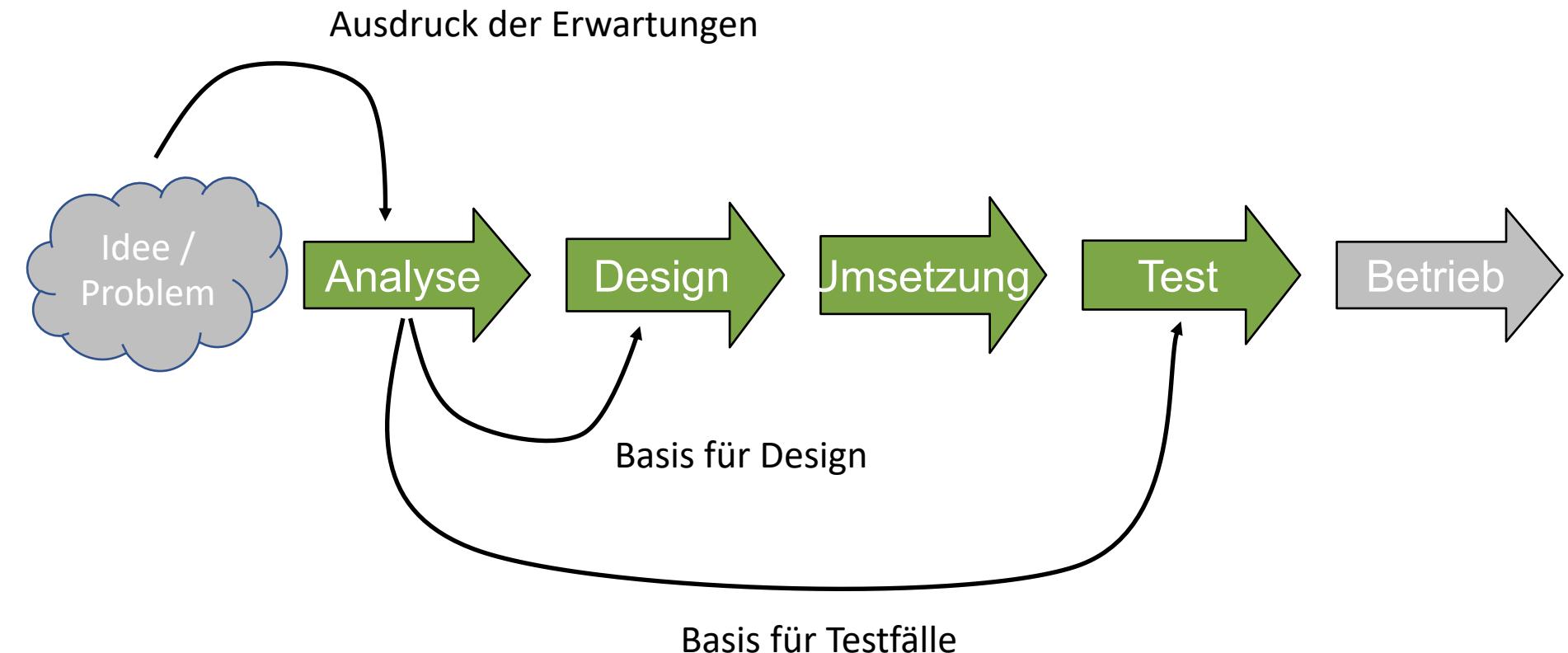
ID	Titel	Text	Art	Quelle
A_00212	RVO: Transparenz für Versicherten	Entsprechend dem jeweiligen Funktionsumfang muss gewährleistet sein, dass die Patientin/der Patient ... die datenverarbeitenden Prozesse in ausreichendem Maße erkennen und nachvollziehen (Transparenz) ... kann.	N	RVO 2006

- RVO 2006: Bundesgesetzblatt Jg. 2006, Seite 2199 ff. (05. Oktober 2006): Neufassung der Verordnung über Testmaßnahmen für die Einführung der elektronischen Gesundheitskarte

# Agenda

- Motivation und Beispiel
- Typen und Eigenschaften von Anforderungen
- Spezifikation von Anforderungen
- Erheben und Management von Anforderungen

# Erinnerung: Anforderungen im Entwicklungsprozess



# Qualität von Anforderungen

## Individuelle Anforderungen

- Korrektheit: Die Anforderung entspricht den Bedürfnissen der Beteiligten / Betroffenen.
- Eindeutigkeit: Die Formulierung der Anforderung erlaubt nur eine gültige Interpretation.
- Prüfbarkeit: Es lassen sich Kriterien oder Tests angeben, um nachzuweisen, ob die Anforderung erfüllt ist oder nicht.
- Umsetzbarkeit: Anforderung sind erfüllbar.
- Notwendigkeit: Jede Anforderung drückt die Erwartung der Betroffenen aus.

## Mengen von Anforderungen

- Minimalität: Es gibt keine doppelten oder überschneidenden Anforderungen.
- Vollständigkeit: Die Anforderungen beschreiben das gewünschte System in allen relevanten Aspekten.
- Konsistenz: Die einzelnen Anforderungen widersprechen sich nicht
- Nachverfolgbarkeit: Der Ursprung der Anforderung und ihre Entwicklung lassen sich feststellen.
- Erweiterbarkeit / Anpassbarkeit: Einzelne Anforderungen lassen sich unabhängig ändern.

# Überprüfbarkeit von Nicht-funktionalen Anforderungen

- Beschreibung von NFR oft unsystematisch und ungenau.
- Darauf achten, messbare Kriterien zu formulieren!

Eigenschaft	Messbare Formulierung
Geschwindigkeit	Ausgeführte Transaktionen / Sekunde Antwortzeit
Speicher	Mbytes Anzahl ROM
Benutzbarkeit	Einarbeitungszeit Umfang der Hilfe
Zuverlässigkeit	Durchschnittliche Zeit bis zu einem Absturz Wahrscheinlichkeit eines Ausfalls Fehlerhäufigkeit Verfügbarkeit
Robustheit	Zeit bis Wiederherstellung nach Ausfall Anteil fehlerhafter Vorgänge

# Dokumentation von Anforderungen

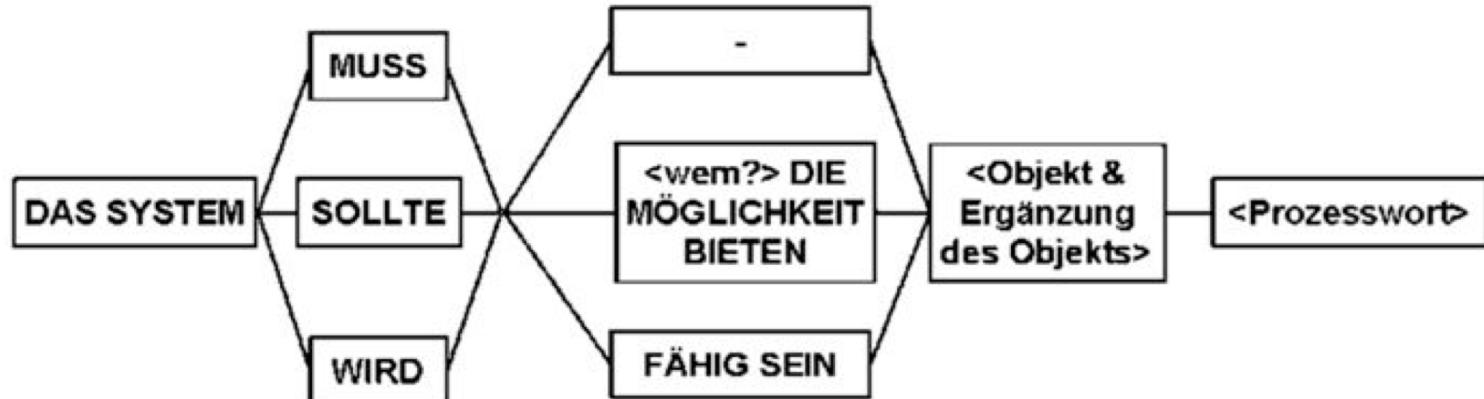
Notation	Eigenschaften
Sätze in natürlicher Sprache	Nummerierte Sätze. Jeder Satz sollte eine Anforderung ausdrücken
Strukturierte natürliche Sprache (z.B. Standardformular oder Satz Schablonen)	Anforderungen sind in natürlicher Sprache verfasst. Sätze werden nach Template gebildet. Jeder Satzteil beschreibt einen Aspekt der Anforderung.
Formale Sprache	Es wird eine formale Sprache verwendet (ähnlich einer Programmiersprache). Beschrieben wird erwartetes Verhalten. (Wird heute selten benutzt)
Grafische Notation	Grafische Modelle, ergänzt um Text Annotationen, beschreiben die funktionalen Anforderungen an das System (z.B. UML Anforderungs- und Aktivitätsdiagramme)
Mathematische Spezifikation	Spezifikation basiert auf mathematischem Formalismus (Automaten oder Mengen). Eindeutig aber oft nicht nutzbar, da Kunden diese nicht verstehen.

Detailliertere Darstellung: Summerville, 2012

# Beispiel: natürliche Sprache

„Um Abhängigkeiten zwischen verschiedenen Requests des Fachmoduls zu vermeiden, darf das Fachmodul in der Kommunikation zum Intermediär kein Pipelining verwenden. Parallelität von Requests vom Fachmodul an den Intermediär VSDM kann ohne Pipelining über mehrere HTTP-connections erreicht werden.“ [VSDM-A\_3066]

# Beispiel: Strukturierte Sätze



Beispielsystem: Sophisten Satzschablonen

Quelle: Basiswissen Requirements Engineering (Pohl, Rupp)

[VSDM-A\_3066]

„(Das Fachmodul VSDM) **DARF NICHT** - (Pipelining [gemäß RFC 2616 für HTTP-Requests zum Intermediär VSDM]) verwenden.“

gemSpec\_SST\_VSDM\_V1.11.0.doc, Version: 1.11.0

# Beispiel: Formale Sprache

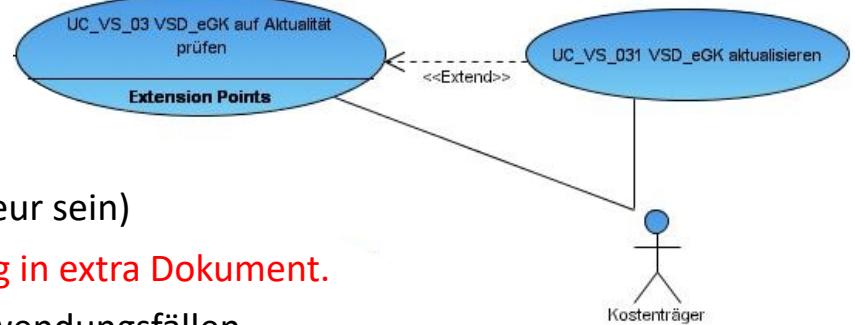
```
PROCESS Ground_level_estimation;  
  
IF (O.OOGROUND EQ $TRUE) THEN  
    G.ZGROUND = G.ZOWN;  
ELSEIF (G.RADAROUT GT P.RADARLOST) THEN  
    G.ZGROUND = -P.ZLARGE;  
ELSEIF (G.RADAROUT GT 0) THEN  
    ...  
END Ground_level_estimation;
```

*“TCAS II requirements were defined by the Minimal Operational Performance Standard (MOPS) document. The MOPS was expressed using a combination of English and approximately 7,000 lines of lowlevel pseudocode.”*

Mats P.E.Heimdahl and Nancy G.Leveson and Jon D. Reese. **Experiences From Specifying the TCAS II Requirements Using RSML**. 17th Digital Avionics Systems Conference, 1998.

# Grafische Notation: UML Anwendungsfalldiagramme

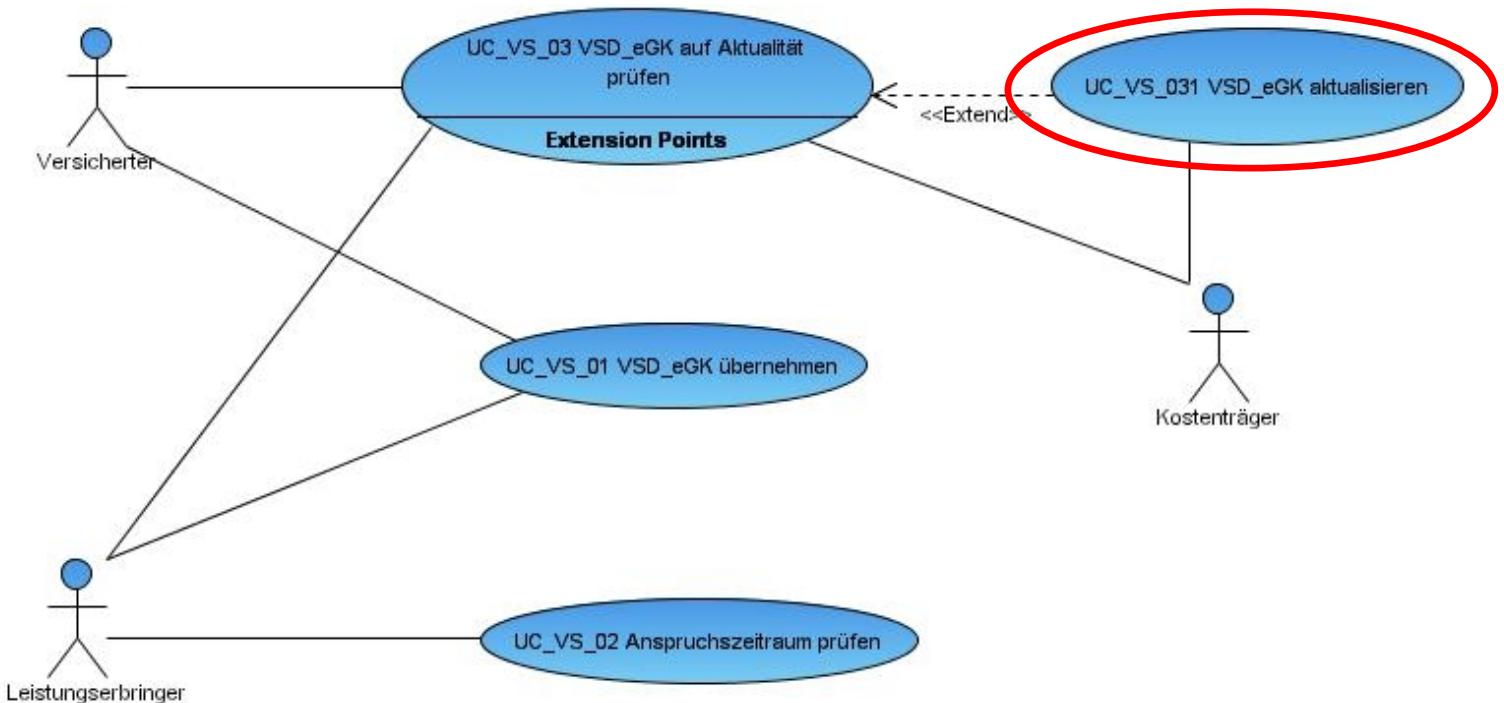
Ein Anwendungsfalldiagramm stellt das *erwartete* Verhalten eines Systems dar.



## Syntax

- **Akteure:** Strichmännchen (System kann auch Akteur sein)
- **Anwendungsfälle:** Ellipsen mit Titel. **Beschreibung in extra Dokument.**
- **Assoziationen:** Linien zwischen Akteuren und Anwendungsfällen.
- **Systemgrenzen:** Rechtecke.
- **include-Beziehungen:** mit <<include>> gekennzeichnete gestrichelte Linie. Pfeil zum inkludierten Anwendungsfall  
Inkludierter für den aufrufenden Anwendungsfall notwendig.
- **extend-Beziehungen** mit <<extend>> gekennzeichnete gestrichelter Linie. Pfeil vom erweiternden Anwendungsfall. Erweiternder kann vom aufrufenden Anwendungsfall aktiviert werden.

# Beispiel: Anwendungsfalldiagramme



gematik\_VSD\_Fachkonzept\_VSDM\_V1.0.0.doc, Version 1.0.0

# Beispiel:

## Beschreibung Anwendungsfall

### 6.2.2 VSD\_eGK aktualisieren

Name :	UC_VS_031 VSD_eGK aktualisieren
Beschreibung :	Der Kostenträger aktualisiert die VSD_eGK über die Telematik-Infrastruktur im Umfeld des Leistungserbringers.
Anwendungsumfeld :	Leistungserbringer
Auslöser :	Leistungsanspruchsprüfung mit Netzzugang vor Inanspruchnahme ärztlicher Leistungen
Vorbedingungen :	<ul style="list-style-type: none"> <li>- eGK liegt vor</li> <li>- Versicherter ist identifiziert als rechtmäßiger Nutzer der eGK</li> <li>- Zugriffsauförderung durch Leistungserbringer ist erfolgt</li> <li>- eGK ist gültig</li> <li>- VSD_eGK ist nicht aktuell</li> </ul>
Eingangsdaten :	- VSD_Kostenträger
Nachbedingungen :	- VSD_eGK in Zustand "aktualisiert"
Ergebnisse :	Aktuelle VSD_eGK stehen zur weiteren Nutzung zur Verfügung
Beteiligte Akteure :	<ul style="list-style-type: none"> <li>- Kostenträger</li> <li>(- Leistungserbringer: stellt eigene Infrastruktur zur Verfügung)</li> <li>(- Versicherter: stellt eGK zur Verfügung)</li> </ul>
Geschäftsobjekte :	VSD_eGK
Standardablauf :	1. Kostenträger: VSD_eGK.* aktualisieren
Varianten :	Keine
Alternative Abläufe :	Keine

Fachliche Fehlermeldungen :	Keine
Referenzen :	Keine
Sicherheits-Anforderungen :	<p>AS_VS_021_01: Die Aktualisierung der VSD_eGK muss über eine vertrauliche Kommunikation zwischen Kostenträger und eGK erfolgen</p> <p>AS_VS_021_02: Das Schreibrecht für die VSD auf der eGK liegt ausschließlich beim Kostenträger (VSDD).</p>
Nichtfunktionale Anforderungen :	<p>AN_VS_031_01: Bei der VSD-Aktualisierung auf der eGK ist der Arztabzug gegenüber dem Kostenträger zu anonymisieren [SV-GB_072005]</p> <p>AN_VS_031_02: Bei der VSD-Aktualisierung auf der eGK handelt es sich um ein Massengeschäft, daher ist eine hohe Anforderung an die Performanz zwingend erforderlich.</p> <p>AN_VS_031_03: Es ist sicher zu stellen, dass der Versicherte und Leistungserbringer über die Aktualisierung informiert wird.</p> <p>AN_VS_031_04: Der Kostenträger muss nachvollziehen können, ob die Aktualisierung der VSD_eGK erfolgreich verlaufen ist.</p> <p>AN_VS_031_05: Verfügbarkeit der VSD_Kostenträger: 24/7</p>
Anmerkungen, Bemerkungen :	Keine

Verweis auf einzelne textuelle Anforderungen

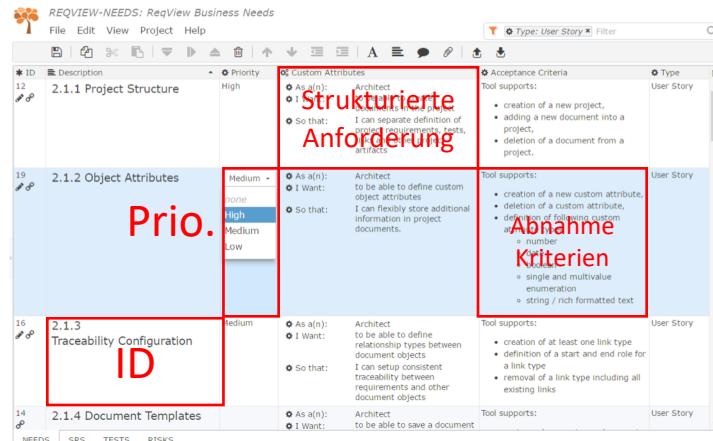
Strukturiert durch Template

# Anforderungen dokumentieren

## Dokumentation

- Woher Anforderungen kam (Autor, Betroffene)
  - Für wen ist sie wichtig
  - Vermeidet Phantome
  - Erlaubt gezielte Rückfragen
- Priorität
- Relation zu anderen Anforderungen
- Abnahme Kriterien

## Werkzeuge



<https://www.reqview.com/>

- IBM Rational Doors
- PTC Integrity
- Vector PREEVision
- MS Excel ☺
- ...

# Agenda

- Motivation und Beispiel
- Typen und Eigenschaften von Anforderungen
- Spezifikation von Anforderungen
- Erheben und Management von Anforderungen

# Herausforderungen

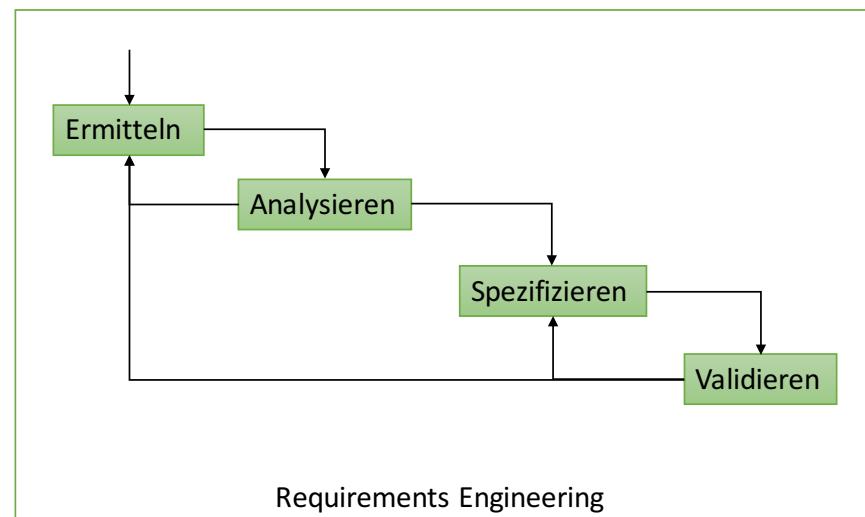
- Veränderung
  - Anforderungen ändern sich während der Analyse und Entwicklung
  - Neue Beteiligte mischen sich ein
- Genauigkeit / Eindeutigkeit
  - Kunden wissen nicht was sie wirklich wollen
  - Kunden benutzen ihre eigene Fachsprache
- Gemeinsame Erwartungen
  - Verschiedene Beteiligte können widersprüchliche Anforderungen haben

# Requirements Engineering

Requirements Engineering umfasst die Tätigkeiten, die mit dem Erheben, Prüfen und Verwalten von Anforderungen (Requirements) verbunden sind:

- Ermitteln
  - Analysieren
  - Spezifizieren
  - Validieren
- } Engineering

- Dokumentieren
  - Fortschreiben
  - Planung / Organisation
- } Management



# Requirements Engineering

1. Ermitteln: Erfassen, Abstimmen, Formulieren der Anforderungen aller Beteiligten / Betroffenen.
2. Analysieren: Gruppieren, Klassifizieren und Priorisieren von Anforderungen.
3. Spezifizieren: Dokumentation von Anforderungen in Standardform.
4. Validieren: Konsistenz, Realisierbarkeit, Vollständigkeit etc. überprüfen.

# Anforderungen ermitteln

## Vorgehen

1. Gruppen von Betroffenen ermitteln
2. Relevante Datenquellen finden (bestehende Software, Gesetze, Normen)
3. Analysemethode festlegen und vorbereiten
4. Anforderungen ermitteln

## Methoden

- Marktanalyse
- Dokumentenanalyse: Normen / Gesetzen
- Beobachtung
- Fragebögen
- Interviews
- Fokusgruppen
- Prototypen

# Anforderungen analysieren

## Gruppieren:

- Zusammengehörigkeit und Überlappungen von Anforderungen in einer großen Menge erkennen (z.B. von unterschiedlichen Autoren)
- Vollständigkeit der Anforderungen prüfen und
- Fehlende Anforderungen erkennen
- Ausschnitte aus dem Anforderungsdokument für bestimmte Rollen erzeugen ("Sichten")

## Klassifizieren / Bewerten:

- Abhängigkeiten, z.B. technische Abhängigkeiten
- Art der Anforderung: Systemfunktionalität, Benutzerschnittstelle, Datenbank, Kommunikation, NFA (z.B. Performanz)
- Betroffene Benutzergruppe: System-Administrator, Arzthelferin, ...
- Priorität

# Priorisierung von Anforderungen

## Wofür?

- Grundlage von technischen und Management-Entscheidungen
- Kompromisse zwischen in Konflikt stehenden Anforderungen finden
- Planung von Releases (zuerst die wichtigen Anforderungen)
- Zertifizierung in kontrollierten Domänen (z.B. eGK)

## Beispiel VSDM

„**MUSS** bedeutet, dass es sich um eine absolut gültige und normative Festlegung bzw. Anforderung handelt.“

„**SOLL** beschreibt eine dringende Empfehlung. Abweichungen zu diesen Festlegungen sind in begründeten Fällen möglich. Wird die Anforderung nicht umgesetzt, müssen die Folgen analysiert und abgewogen werden.“

„**KANN** bedeutet, dass die Eigenschaften fakultativ oder optional sind. Diese Festlegungen haben keinen Normierungs- und keinen allgemeingültigen Empfehlungscharakter.“

# Anforderungen spezifizieren

- Siehe vorheriger Abschnitt

# Anforderungen validieren

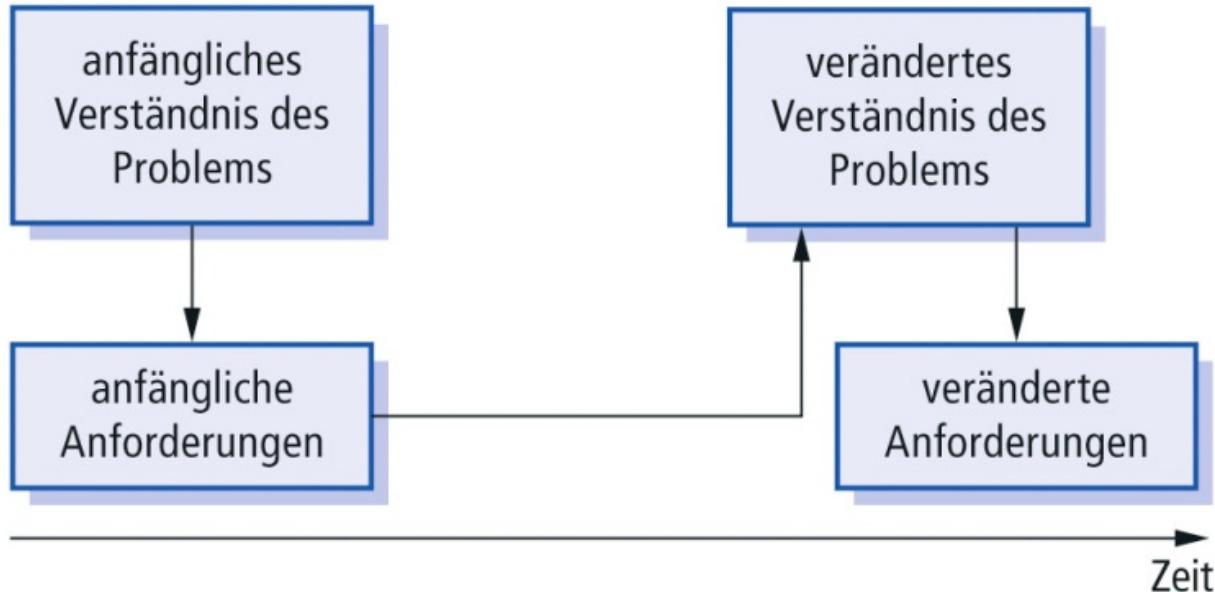
## Fragen

- Wird der Bedarf des Kunden vollständig abgedeckt?
- Verständlich formuliert?
- Konsistent mit anderen Anforderungen?
- Anforderung prüfbar?
- Änderbar ohne Einfluss auf andere Anforderungen?
- Realistisch mit Budget und Technologie?

## Methoden

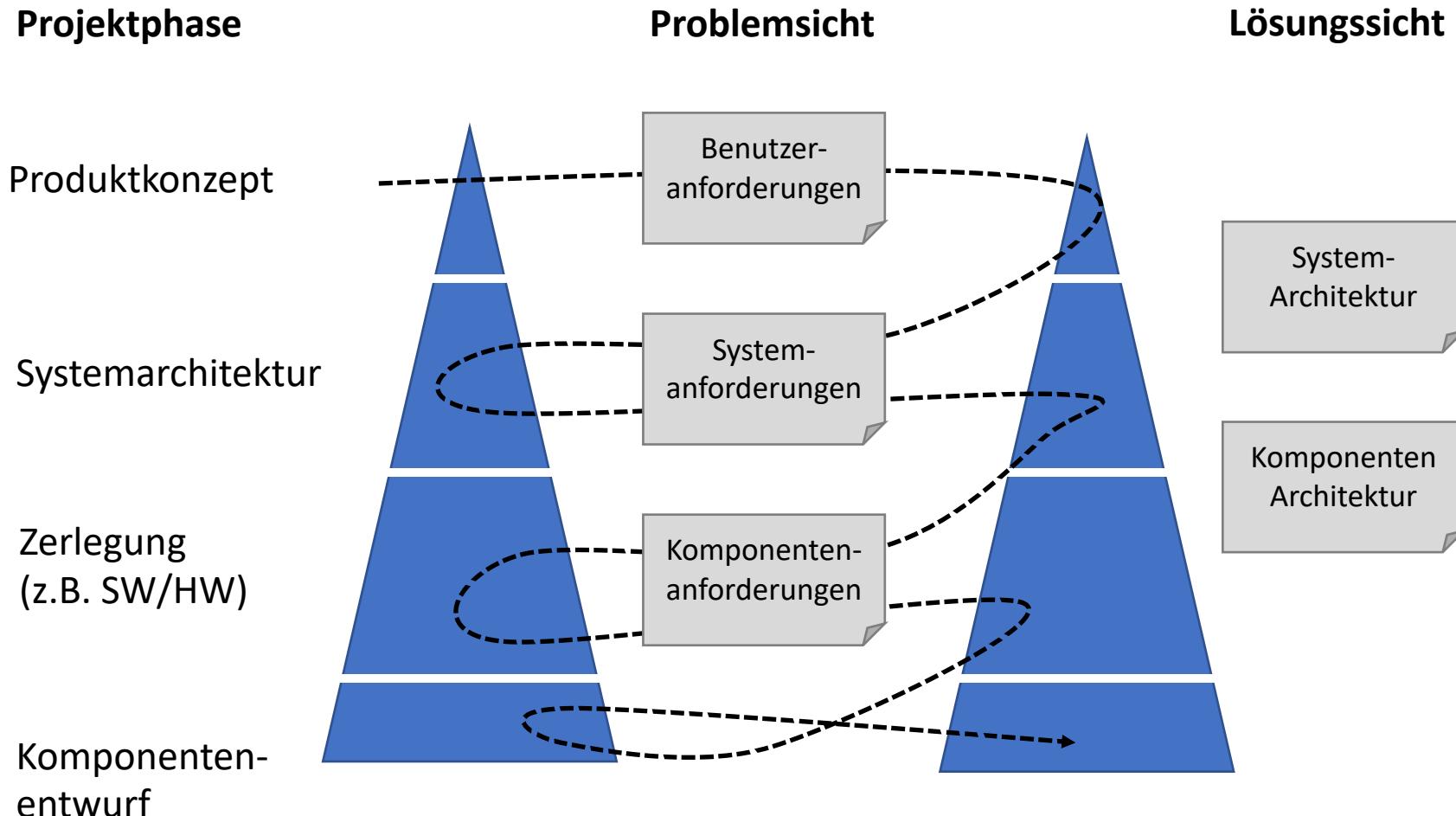
- Reviews
- Prototypen
- Erzeugung von Testfällen

# Anforderungen fortschreiben



- Durch intensive Auseinandersetzung mit einem Problem ändert sich oft auch das Verständnis des Problems  
⇒ Anforderungen ändern sich

# Ko-Entwicklung von Architektur und Anforderungen



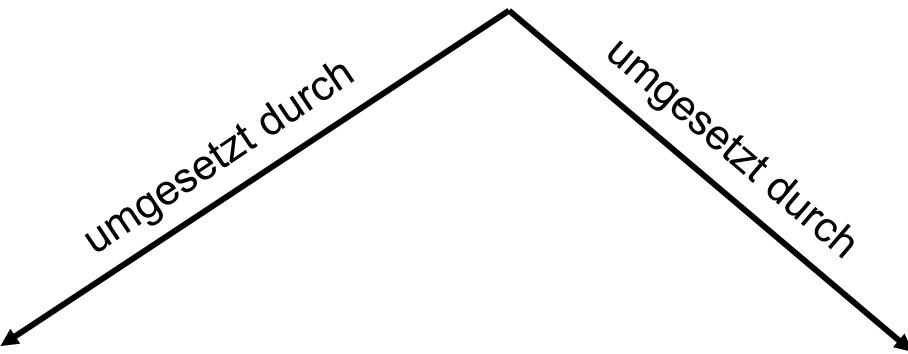
nach Ebert, 2014: Systematisches Requirements Engineering, 5. Auflage

# Beispiel: Elektronische Gesundheitskarte

VSDM- A\_2146: Die Fachanwendung VSDM MUSS die Berechtigungen der Akteure entsprechend der Tabelle Tab\_VSDM\_SysL\_28 "fachliche Berechtigungsmatrix VSDM" umsetzen.

VSDM- A\_2971: Die eGK MUSS den Zugriffsschutz für die Container EF.PD, EF.VD, EF.Prüfungsnachweis und EF.StatusVD so einstellen, dass sie jederzeit ohne Authentisierung lesbar sind.

VSDM- A\_2972: Die eGK G 2 MUSS den Zugriff auf den Container EF.GVD schützen und mittels C2C-Authentisierung nur berechtigten Rollen (alle Rollen außer "Mitarbeiter Rettungsdienst" und "Administrator") erlauben.



# Zusammenfassung

## Anforderungen

- Erwartungen von Beteiligten / Betroffenen an ein System und Entwicklungsprozess
- Basis für spätere Phasen der Entwicklung
- Basis für Verträge über Software Entwicklung
- Erfolgsfaktor in Projekten

## Typen von Anforderungen

- Funktionale Anforderungen
- Nicht-funktionale Anforderungen

## Qualität von Anforderungen

- Korrektheit, Eindeutigkeit, Prüfbarkeit, Umsetzbarkeit, Notwendigkeit
- Minimalität, Vollständigkeit, Konsistenz, Nachverfolgbarkeit, Erweiterbarkeit / Anpassbarkeit

# Zusammenfassung

## Spezifikation von Anforderungen

- Wichtig: Überprüfbarkeit von Anforderungen
- Verschiedene Methoden (von natürlicher Sprache bis zu vollständig formal)
  - Satzschablonen
  - Anwendungsfalldiagramme
- Dokumentation
  - Priorisierung,
  - Gruppierung, Zusammenhänge zwischen Anforderungen
  - Herkunft

## Erheben und Management von Anforderungen

- Betroffene identifizieren
- Vier Schritte: Erhebung, Analyse, Spezifikation, Validierung
- Verschiedene Methoden zur Erhebung: Interviews, Fokusgruppen, Fragebögen, ...
- Analyse: Priorisieren, Gruppieren und Klassifizieren
- Validierung: Qualität, Konsistenz, Korrektheit sicherstellen mit Hilfe von Reviews, Prototypen, ...