



Media Engineering Requirements Engineering



R. Weller

University of Bremen, Germany

cgvr.cs.uni-bremen.de

- Hilfreich ist es, schon während der Analyse ein **Begriffslexikon** anzulegen; dieses wird während der gesamten Software-Entwicklung verwendet und ergänzt.
- In den frühen Phasen wird dieses Dokument **aufgebaut und weiterentwickelt**.
- Das Begriffslexikon enthält solche Begriffe, die
 - **wichtig** sind und
 - von verschiedenen Leuten, v.a. von Klienten, Analytikern und Entwicklern, **unterschiedlich ausgelegt werden** könnten.
- Dazu gehören häufig **Begriffe, die auf den ersten Blick völlig klar zu sein scheinen**.
- Beispiel: Informationssystem für die Prüfungsdaten von Studenten
 - „Student“, „Prüfung“, „Note“, „Erstprüfung“



- a) Begriff und Synonyma (im Sinne der Spezifikation)
 - b) Bedeutung (Definition, Erklärung)
 - c) Abgrenzung (wo ist dieser Begriff nicht anzuwenden?)
 - d) Gültigkeit (zeitlich, räumlich, sonst)
 - e) Fragen der Bezeichnung, Eindeutigkeit u. A.
 - f) Unklarheiten, die noch nicht beseitigt werden konnten
 - g) verwandte Begriffe (Querverweise)
- Die Angaben werden **aus den Gesprächen und Interviews abgeleitet** oder, wenn sie von den Analytikern kommen, **sorgfältig mit den Klienten überprüft.**
 - Typische Quellen sind im genannten Beispiel die Angestellten in der Verwaltung, die Juristen der Uni, die Gesetze und Ordnungen der Universität.

Begriff	Student , synonym Studentin, Studierender, Studierende
Bedeutung	Eine Person, die an der Universität Stuttgart immatrikuliert ist und noch nicht exmatrikuliert wurde, die folglich legal einen Studentenausweis der Universität Stuttgart hat oder haben könnte.
Abgrenzung	Gasthörer und Studierende anderer Hochschulen sind im Sinne dieses Systems keine Studenten.
Gültigkeit	Mit der Immatrikulation an der Universität Stuttgart entsteht ein neuer Student; er existiert bis zur Exmatrikulation, gleichgültig, wie sie zustande kommt. Ein Fachwechsel oder eine Namensänderung implizieren keine Exmatrikulation. Hat sich eine Person im Laufe ihres Lebens mehrfach an der Universität Stuttgart immatrikuliert, so handelt es sich um mehrere, nicht identische Studenten.
Bezeichnung	Ein Student ist durch die Matrikelnummer und einen Zeitpunkt (zu dem die Matrikelnummer gültig war oder ist) eindeutig bestimmt, alle anderen Attribute, insbesondere der Name, können mehrfach vorkommen.
Unklarheiten	Es ist noch ungeklärt, wie Namen aus anderen Schriftsystemen (z. B. Russisch, Arabisch, Chinesisch) dargestellt werden.
Querverweise	Gasthörer, Matrikelnummer, Studentenausweis

Definition (Pohl):

Ein *Szenario* beschreibt ein konkretes Beispiel für die Erfüllung bzw. Nichterfüllung eines oder mehrerer Ziele. Es konkretisiert dadurch eines oder mehrere Ziele. Ein Szenario enthält typischerweise eine Folge von Interaktionsschritten und setzt diese in Bezug zum Systemkontext.

- Szenarientypen
 - Positive, negative, und Missbrauchsszenarien
 - Deskriptive, explorative, oder erklärende Szenarien
 - Systeminterne, Interaktive, Kontextszenarien
 - Haupt-, Alternativ- und Ausnahmeszenarien

- Anwendungsfälle (Use Cases) gruppieren Szenarien

Beispiel „automatisches Bremsmanöver“

- (nach Pohl) Der Fahrer fährt mit konstanter Geschwindigkeit auf der Autobahn. Das vorausfahrende Fahrzeug bremst scharf. Der Fahrer tritt auf das Bremspedal. Das System erkennt eine Unterschreitung des Sicherheitsabstands und gibt eine Warnung aus. Der Abstand zwischen den Fahrzeugen verringert sich weiter. Das System leitet eine automatische Vollbremsung ein und informiert den Fahrer über den Bremsvorgang. Als der Abstand sich nicht mehr verringert, beendet das System die automatische Vollbremsung. Danach stellt es den Mindestabstand zum vorausfahrenden Fahrzeug wieder her und informiert den Fahrer über die Rückgabe der Kontrolle.
- Was wird klar, was bleibt unklar?

Attribute von Szenarien

- **Positiv:** Wie wird das Ziel erfüllt?
- **Negativ:** Wie wird das Ziel nicht erfüllt?
- **Missbrauchsszenarien:** unerwünschte Verwendung

- **Deskriptiv:** verdeutlicht typische Abläufe
- **Explorativ:** verdeutlicht Lösungsraum
- **Erklärend:** verdeutlicht Gründe für Systemverhalten

- **Systemintern:** Handlungsstränge innerhalb des Systems
- **Interaktiv:** Handlungsstränge zwischen Akteuren
- **Kontextszenarien:** weitere Kontextinformationen

- **Hauptszenario:** wesentlicher Funktionsablauf
- **Alternativszenario:** partielle Abwandlung eines Hauptszenarios
- **Ausnahmeszenario:** Reaktion auf Ereignisse, die Hauptszenario verhindern

1. Der Fahrer schaltet das Navigationssystem ein.
2. Das System ermittelt den aktuellen Standort des Fahrzeugs.
3. Das System erfragt den gewünschten Zielort.
4. Der Fahrer gibt den Zielort in das System ein.
5. Das System ermittelt den benötigten Fartenausschnitt.
6. Das System zeigt die Karte des Zielgebietes am Bildschirm an.
7. Das System erfragt die Optionen für die Routenberechnung.
8. Der Fahrer wählt die Optionen für die Routenberechnung.
9. Das System bestimmt die Wegführung.
10. Das System zeigt eine Erfolgsmeldung am Bildschirm.
11. Das System stellt eine Liste von Wegpunkten zusammen.
12. Das System zeigt den nächsten Wegpunkt der Navigation.

Szenario „Navigation zum Zielort“	
Fahrer	Navigationssystem
1. Einschalten des Geräts	
	2. Ermittelt den aktuellen Standort
	3. Erfragt den Zielort
4. Eingabe des Zielorts	
	5. Ermittelt den benötigten Kartenausschnitt
	6. Zeigt am Display Karte vom Zielgebiet
	7. Erfragt die Optionen für die Routenberechnung
8. Eingabe der Optionen für die Routenberechnung	
	9. Ermittelt die Wegführung
	10. Anzeige „Weg wurde berechnet“
	11. Ermittelt die Liste der Wegpunkte
	12. Anzeige des jeweils nächsten Wegpunkts

- Stark vereinfachtes Bild von den Anforderungen in der Praxis:
 - Wenn der Klient eine Anforderung nicht **von selbst genannt** hat, dann **hat** er diese Anforderung **nicht**.
 - Wenn eine Anforderung **funktional** oder **quantifizierbar** ist, wird sie **dokumentiert**.
 - Alle anderen Anforderungen werden **pauschal als nichtfunktionale** Anforderungen bezeichnet und **kaum erwähnt**. Das gilt auch für
 - die Gebrauchsqualität (z.B. der **Bedienbarkeit**)
 - die Wartungsqualität (z.B. der **Erweiterbarkeit**).
- Empfehlung:
 - **Anforderungen nicht „vergessen“, nur weil sie Mühe machen.**



Warum machen wir das Ganze?

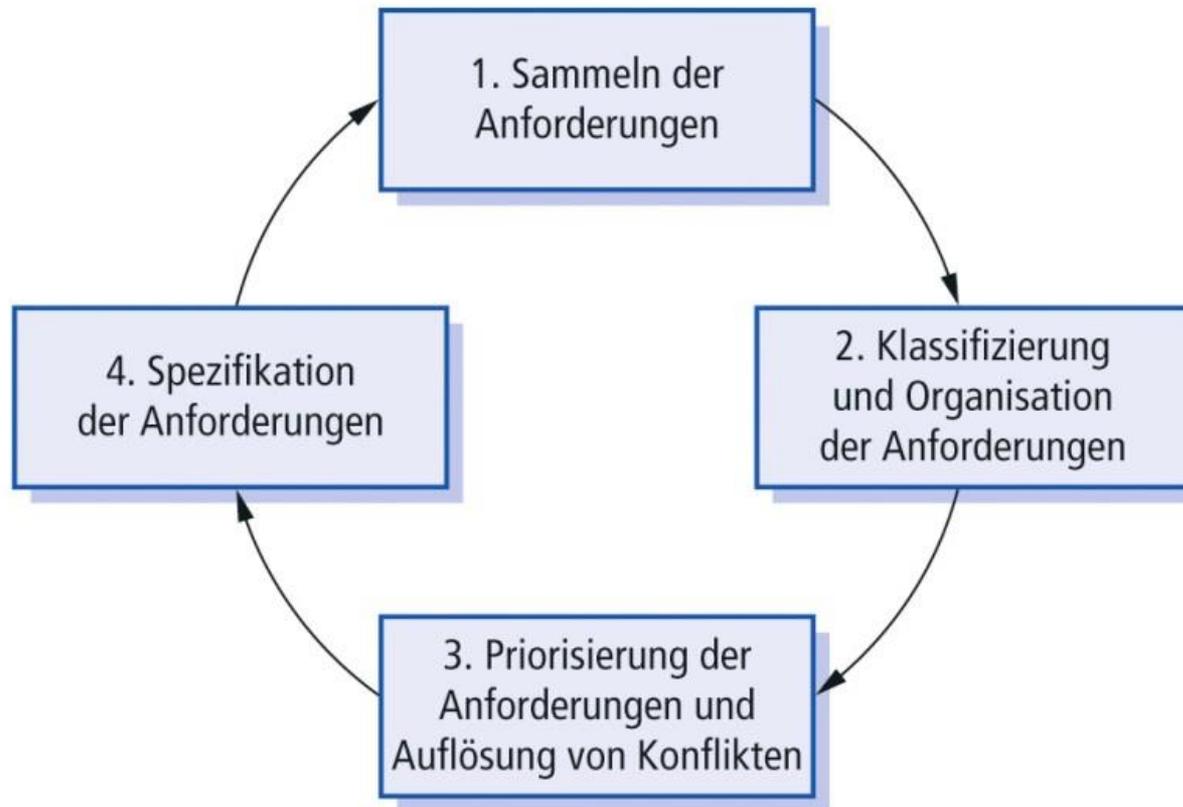


- In der Praxis gibt es viele schlechte Anforderungsanalysen; oft gibt es gar keine.
- Die Anforderungsanalyse ist aber notwendig für
 1. die **Abstimmung** mit dem **Kunden** bzw. mit dem Marketing,
 2. den **Entwurf** und die **Implementierung**,
 3. das **Benutzungshandbuch**,
 4. die **Testvorbereitung**,
 5. die **Abnahme**,
 6. die **Wiederverwendung**,
 7. die **Klärung** späterer Einwände, Regressansprüche usw.,
 8. eine spätere **Re-Implementierung**.

1. Die Anforderungen bleiben **ungeklärt**, sie werden darum auch nicht erfüllt.
2. Den Entwicklern **fehlt die Vorgabe**, darum fragen sie „auf dem kurzen Dienstweg“ Bekannte, die beim Kunden arbeiten, oder sie legen mangels Kontakten die eigenen Erfahrungen und Erwartungen zu Grunde.
3. Die **Basis für das Handbuch fehlt**, es wird darum phänomenologisch, d.h. experimentell, verfasst.
4. Ein gutes Handbuch ist ein umformulierter Auszug aus der Anforderungsanalyse!
5. Ein **systematischer Test** ist ohne Anforderungsanalyse unmöglich, denn es ist nicht definiert, welche Daten das System akzeptieren muss und welche Resultate es liefern soll.

6. Wenn bei der **Abnahme** nicht entschieden werden kann, ob das System richtig arbeitet, wird die Korrektheit zur Glaubensfrage.
 7. Oft zeigen sich **echte oder vermeintliche Mängel** der Software erst nach längerem Gebrauch. Ohne Anforderungen kann diese Unterscheidung aber nicht getroffen werden.
 8. Wer eine Software(-Komponente) **wiederverwenden** will, muss wissen, was sie leistet. Das ist in der Spezifikation dokumentiert.
 9. Wenn ein System ausgemustert und ersetzt wird, ist **Aufwärtskompatibilität** gefordert (vgl. Heninger et al., 1978).
- „**Spezifikation im Kopf**“ gibt es nicht!

- Requirements-Engineering ist kein linearer Prozess, sondern ein iterativer





"Based on our tests, the business stakeholders fall asleep around page 37 of the Functional Requirements Specification. Put the Issues Section on page 40."