Inhaltsverzeichnis

1 RE * 1
2 Gliederung * 2
3 Lastenheft? * 3
4 Der RE-Prozess * 4
5 Anforderungen? * 6
6 Visionen & Ziele * 7
7 Rahmenbedingungen * 8
8 Kontext & Überblick * 9
9 Funktionale & nichtfunktionale Anforderungen * 10
10 Abnahmekriterien * 12
11 Natürlichsprachliche Anforderungen * 13
12 Anforderungsschablone IEEE 830–1998 * 15
13 Lastenheftschablonen im V-Modell XT * 16**
14 Schablonen für Lastenheft & Glossar ** 17
15 Fallstudie: SemOrg - Lastenheft & Glossar * 18
16 Schablonen für agile Entwicklungen ** 21
17 Anforderungen ermitteln & spezifizieren ** 24
18 Anforderungen priorisieren ** 26
19 Resümee * 27
20 Zum guten Schluss * 28
Literatur 29

1 Requirements Engineering *

Was Auftraggeber falsch machen (können)

Gibt es das perfekte Lastenheft?

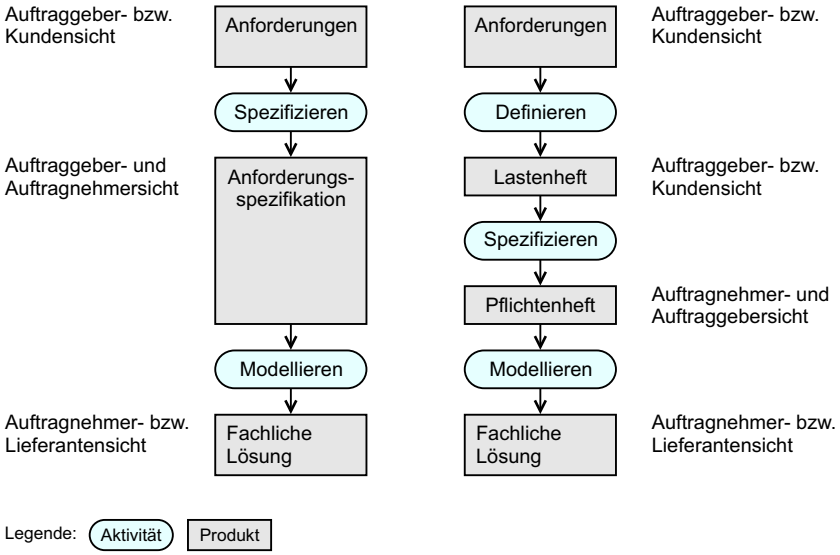
Prof. Dr. Helmut Balzert

2 Gliederung *

- Was ist ein Lastenheft?
- Der Requirements Engineering-Prozess
- Was sind Anforderungen?
- Schablonen für Lastenhefte
- Was kann der Auftraggeber alles tun?
- Denkfallen für den Auftraggeber

3 Was ist ein Lastenheft? *

Die Anforderungsspezifikation



■ Definition 1

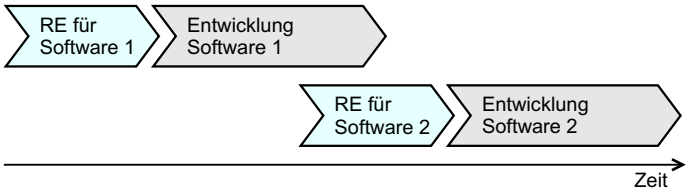
□ **Lastenheft**: »Zusammenstellung aller Anforderungen des Auftraggebers hinsichtlich Liefer- und Leistungsumfang. Im Lastenheft sind die Anforderungen aus Anwendersicht einschließlich aller Randbedingungen zu beschreiben. Diese sollen quantifizierbar und prüfbar sein. Im Lastenheft wird definiert WAS und WOFÜR zu lösen ist. Das Lastenheft wird vom Auftraggeber oder in dessen Auftrag erstellt. Es dient als Ausschreibungs-, Angebots- und/oder Vertragsgrundlage« /VDI2519, S. 2/

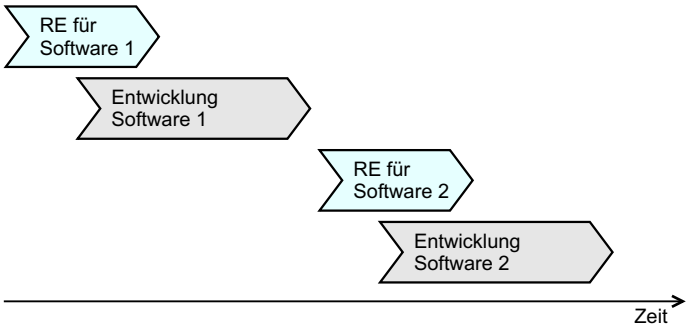
■ Definition 2

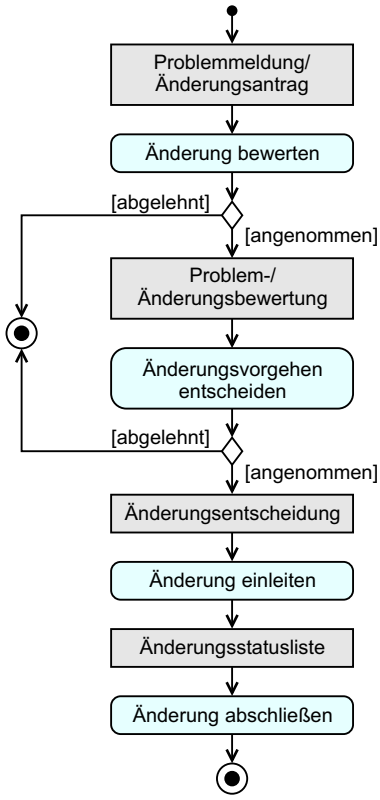
□ **Lastenheft**: »Vom Auftraggeber festgelegte Gesamtheit der Forderungen an die Lieferungen und Leistungen eines Auftragnehmers innerhalb eines Auftrags« /DIN69905, S. 3/

4 Der Requirements Engineering-Prozess *

RE als eigenständige Phase + Änderungsmanagement



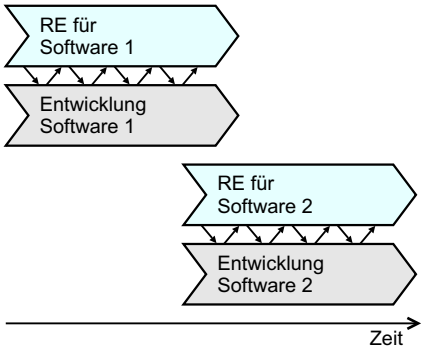




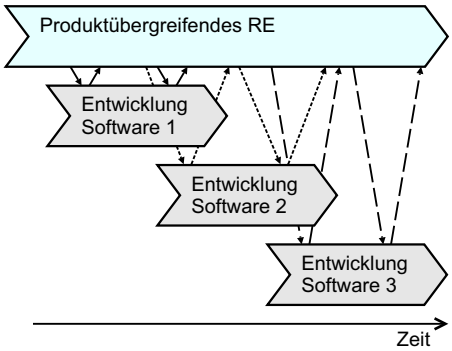
+ Es gibt ein klar abgegrenztes Ende der RE-Phase, zu dessen Zeitpunkt auch der Auftraggeber sich klar für die fachliche Lösung entscheiden muss.

– Spätere Änderungen müssen immer über den Änderungsprozess abgewickelt werden.

RE entwicklungsbegleitend



RE produktübergreifend



+ Es entsteht eine fortlaufende Wissensbasis für die Produktdomäne, auf die bei neuen Produkten sofort zugegriffen werden kann.
+ Die für das RE zuständigen Mitarbeiter können sich ganz auf die Entwicklung und Pflege der RE-Wissensbasis konzentrieren.
– Es entsteht ein erhöhter Koordinierungsaufwand zwischen den Projekten.

5 Was sind Anforderungen? *

■ Definition

□ Anforderungen *(requirements)* legen fest, was **man** von einem Softwaresystem als **Eigenschaften** erwartet.

Wer ist »man«?

Was sind »Eigenschaften«?

6 Visionen und Ziele *

Vision

- Eine **Vision** ist eine realitätsnahe Vorstellung der gewünschten Zukunft. Sie beschreibt, was erreicht werden soll, sagt aber nicht wie.

- Bezogen auf die Softwaretechnik: Eine Vision sollte als Leitgedanke für alle *Stakeholder* dienen.

Ziele

- **Ziele** dienen dazu, die Vision zu verfeinern und zu operationalisieren.

- **Regel 1: Ziele kurz und prägnant formulieren**
- **Regel 2: Aktivformulierungen verwenden**
- **Regel 3: Überprüfbare Ziele formulieren**
- **Regel 4: Ziele, die nicht überprüft werden können, nicht verfeinern**
- **Regel 5: Den Mehrwert eines Ziels hervorheben**
- **Regel 6: Das Ziel sollte begründet werden**
- **Regel 7: Keine Lösungsansätze angeben**
- **Regel 8: Einschränkende Rahmenbedingungen aufführen**
- **Regel 9: Realistische Ziele formulieren**

- Denkfalle des Auftraggebers:
  Visionen & Ziele werden sich im Laufe der Anforderungserstellung schon ergeben.

7 Rahmenbedingungen *

■ Rahmenbedingung: Legt organisatorische und/oder technische Restriktionen für das Softwaresystem und/oder den Entwicklungsprozess fest.

□ Organisatorischen Rahmenbedingungen

○ Anwendungsbereiche

○ Zielgruppen

○ Betriebsbedingungen

□ Technische Rahmenbedingungen

○ Technische Produktumgebung

○ Anforderungen an die Entwicklungsumgebung

■ Rahmenbedingungen können

□ keine der Anforderungen einschränken,

□ mögliche Realisierungen von Anforderungen einschränken,

□ zur Änderung von Anforderungen führen,

□ zu neuen Anforderungen führen oder

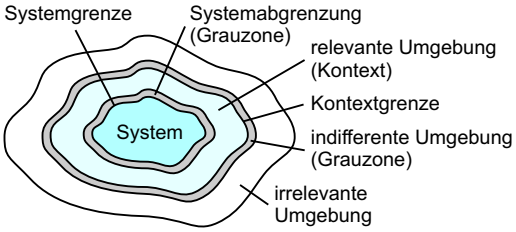
□ zu nicht realisierbaren Anforderungen führen.

■ Denkfalle Auftraggeber:
Rahmenbedingungen legt der Auftragnehmer fest.

8 Kontext und Überblick *

■ Jedes Softwaresystem ist in eine materielle und immaterielle Umgebung eingebettet.

□ Die Systemumgebung hat einen maßgeblichen Einfluss auf die Anforderungen.



■ Auftraggeber:
Kennt Umgebung oft am besten.

9 Funktionale und nichtfunktionale Anforderungen *

■ Definition:
Eine funktionale Anforderung legt eine vom Softwaresystem oder einer seiner Komponenten bereitzustellende Funktion oder bereitzustellenden Service fest.

□ Funktionale Anforderungen lassen sich gliedern in

○ Anforderungen, die die **Statik** des Systems beschreiben,

○ Anforderungen, die die **Dynamik** des Systems beschreiben und

○ Anforderungen, die die **Logik** des Systems beschreiben.

■ Definition:
Nichtfunktionale Anforderungen (*nonfunctional requirements*, kurz NFRs), auch Technische Anforderungen oder *Quality of Service* (QoS) genannt, beschreiben Aspekte, die typischerweise mehrere oder alle funktionale Anforderungen betreffen bzw. überschneiden *(cross-cut)*.

■ Definition:
Eine Qualitätsanforderung legt eine qualitative und/oder quantitative Eigenschaft des Softwaresystems oder einer seiner Komponenten fest.

□ ISO/IEC 9126–1: 6 Q-Merkmale

○ **Funktionalität**

○ **Zuverlässigkeit**

○ **Benutzbarkeit**

○ **Effizienz**

○ **Wartbarkeit**

○ **Portabilität**

□ Qualitätsstufen

Systemqualität	sehr gut	gut	normal	nicht relevant
Funktionalität		X		
Zuverlässigkeit			X	
Benutzbarkeit		X		
Effizienz			X	
Wartbarkeit			X	
Portabilität			X	

Tab. 9.0-1:

□ Allgemein legen funktionale Anforderungen fest, »was« ein System tut bzw. tun soll, während die Qualitätsanforderungen beschreiben, »wie gut« die Funktionen erledigt werden (sollen).

- Denkfalle Auftrageber:
  »Ich will die beste Qualität die möglich ist« oder
  Über Qualität wird nicht nachgedacht.

10 Abnahmekriterien *

- Es ist von vornherein klar, wie überprüft werden kann, ob die Anforderungen korrekt realisiert wurden.
- Es wird sich bei der Abnahme nicht nur auf das realisierte System gestützt.
- Bereits bei der Formulierung der Anforderungen wird darauf geachtet, dass sie auch überprüft werden können, was nach aller Erfahrung zu einer operationalisierten Festlegung und qualitativen Verbesserung der Anforderungen führt.
- Die Formulierung von Abnahmekriterien führt zu besserer Veranschaulichung und Verständlichkeit der meist abstrakt formulierten Anforderungen.

- Denkfalle Auftraggeber:
  Abnahmekriterien müssen erst nach der Fertigstellung des Systems erstellt werden
  oder
  Abnahmekriterien erstellt der Auftragnehmer

11 Natürlichsprachliche Anforderungen *

+ Eine natürliche Sprache ist *einfach*.
+ Sie ist *flexibel* – sie kann abstrakte und konkrete Dinge beschreiben.
+ Sie ist für *jede* Anwendungsdomäne einsetzbar – sie ist *universell*.

■ Natürliche Sprachen sind *mehrdeutig*:
- Lexikalische Mehrdeutigkeit
- Syntaktische Mehrdeutigkeit
- Semantische Mehrdeutigkeit
- Referentielle Mehrdeutigkeit
- Vage Begriffe

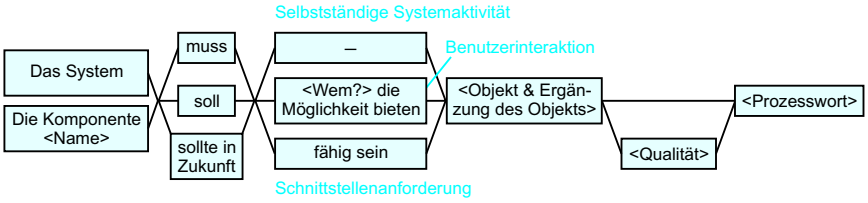
■ 3 Möglichkeiten, um die Probleme zu reduzieren oder zu vermeiden:

□ Erstellung eines Glossars

□ Benutzung sprachlicher Anforderungsschablonen

□ Weitgehende Vermeidung natürlichsprachlicher Anforderungen

Benutzung sprachlicher Anforderungsschablonen



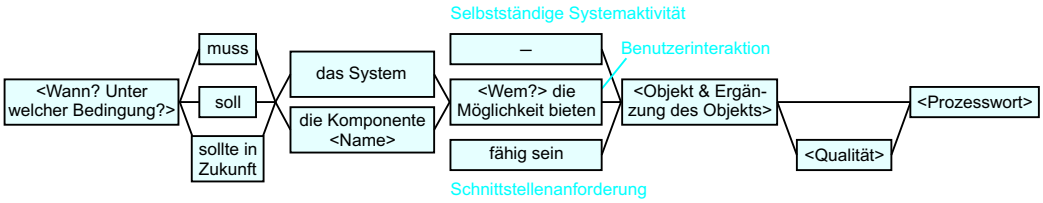
□ Beispiele

○ Benutzerinteraktion:
Das System muss *dem Kunden* die Möglichkeit bieten, sich über *Seminare und Veranstaltungen* zu informieren.

○ Benutzerinteraktion:
Das System soll *dem Seminarsachbearbeiter* die Möglichkeit bieten, *Seminare und Veranstaltungen* mit *selbst erstellten Suchanfragen auszuwerten*.

○ Selbstständige Systemaktivität:
Das System muss die *Kundendaten permanent speichern*.

○ Schnittstellenanforderung:
Das System muss fähig sein, dem Buchhaltungssystem *Rechnungsdatensätze mindestens einmal am Tag* zur Verfügung zu *stellen*.



□ Beispiel

○ Selbständige Systemaktivität:
Falls ein Kunde im Zahlungsverzug ist, muss das System *eine neue Buchung nicht erlauben*.

+ Derer Einsatz einer Schablone führt zu einer Vereinheitlichung der Anforderungsbeschreibungen und zu einer eindeutigeren Semantik.

- Es ist schwierig, alle Anforderungen in Form solcher Sätze zu formulieren – manche Festlegungen sind schwer zu lesen.

- Denkfalle Auftraggeber:
  Anforderungen lasen sich natürlichsprachlich präzise beschreiben, wenn man sich nur Mühe gibt.

12 Anforderungsschablone der IEEE 830–1998 *

1 Einleitung
Gibt einen Überblick über die Anforderungsdefinition
1.1 Zielsetzung
1.2 Produktziele
1.3 Definitionen, Akronyme und Abkürzungen
1.4 Referenzen
1.5 Überblick
2 Übersichtsbeschreibung
Gibt einen Überblick über das Produkt und die allgemeinen Faktoren, die seine Konzeption beeinflussen
2.1 Produkt-Umgebung
2.2 Produkt-Funktionen
2.3 Benutzer-Eigenschaften
2.4 Restriktionen
2.5 Annahmen und Abhängigkeiten
3 Spezifische Anforderungen
Beschreibung aller Details, die für die Erstellung des System-Entwurfs benötigt werden Das am besten geeignete Gliederungsschema dieses Kapitels hängt von der Anwendung und der zu spezifizierenden Software ab.
Die IEEE-Richtlinie enthält dazu acht Vorschläge.

☐ Externe Schnittstellen-Anforderungen

☐ Funktionale Anforderungen

☐ Leistungsanforderungen

☐ Entwurfsrestriktionen

☐ Qualitätsmerkmale

☐ Andere Anforderungen

3 Spezifische Anforderungen (Organisiert nach Objekten)
3.1 **Externe Schnittstellen-Anforderungen**
3.1.1 Benutzungsschnittstelle
3.1.2 Hardware-Schnittstellen
3.1.3 Software-Schnittstelle
3.1.4 Kommunikations-Schnittstellen
3.2 **Klassen/Objekte**
3.2.1 Klasse/Objekt 1
3.2.1.1 Attribute (direkt oder geerbt)
3.2.1.1.1 Attribut 1 bis 3.2.1.1.n Attribut n
3.2.1.2 Funktionen
3.2.1.2.1 Funktionale Anforderung 1.1 bis 3.2.1.2.m Funktionale Anforderung 1.m
3.2.1.3 Botschaften
3.2.2 Klasse/Objekt 2 bis 3.2.p Klasse/Objekt p
3.3 **Leistungsanforderungen**
3.4 **Entwurfsrestriktionen**
3.5 **Qualitätsmerkmale**

13 Lastenheftschablonen im V-Modell XT ***

■ 2 Vorlagen für Lastenhefte

Lastenheft Gesamtprojekt

□ Im »Lastenheft Gesamtprojekt« sollen alle an das zu entwickelnde System verbindlich gestellten Anforderungen vollständig und konsistent beschrieben werden.

□ Es ist Basis für die Aufteilung in Teilprojekte.

Anforderungen (Lastenheft)

1 Einleitung
2 Ausgangssituation und Zielsetzung
3 Funktionale Anforderungen
4 Nicht-Funktionale Anforderungen
5 Skizze des Lebenszyklus und der Gesamtsystemarchitektur
6 Lieferumfang

7 Abnahmekriterien
8 Abkürzungsverzeichnis
9 Literaturverzeichnis
10 Abbildungsverzeichnis

14 Schablonen für Lastenheft und Glossar **

Das Lastenheft

1 Visionen und Ziele
2 Rahmenbedingungen
3 Kontext und Überblick
4 Funktionale Anforderungen
5 Qualitätsanforderungen

☐ Qualitätszielbestimmung

Systemqualität	sehr gut	gut	normal	nicht relevant
Funktionalität		X		
Zuverlässigkeit			X	
Benutzbarkeit		X		
Effizienz			X	
Wartbarkeit			X	
Portabilität			X	

Tab. 14.0-1:

/LQB10/ Qualitätsanforderung zur Benutzbarkeit des Systems
/LQE10/ Qualitätsanforderung zur Effizienz des Systems usw.

Das Glossar

15 Fallstudie: SemOrg - Lastenheft und Glossar *

Lastenheft SemOrg

Version	Autor	Quelle	Status	Datum	Kommentar
0.1	Balzert	Geschäftsführer Teachware	in Bearbeitung	5/09	

Voreinstellungen *(Kursiv dargestellt)*:
Priorität aus Auftraggebersicht = {hoch, *mittel*, niedrig}
Priorität aus Auftragnehmersicht = {hoch, *mittel*, niedrig}
Stabilität = {fest, *gefestigt*, volatil}
Kritikalität = {hoch, mittel, *niedrig*, keine}
Entwicklungsrisiko = {hoch, mittel, *niedrig*}

1 Visionen und Ziele

/LV10/ Die Firma Teachware soll durch das System in die Lage versetzt werden, die von ihr veranstalteten Seminare sowie Kunden und Dozenten effizient rechnerunterstützt zu verwalten.
/LV20/ Die Kunden der Firma Teachware sollen über das Web möglichst viele Vorgänge selbst durchführen können.

/LZ10/ Ein Interessent oder ein Kunde kann mindestens 20 Stunden jeden Tag Seminare und Veranstaltungen über das Web selektieren und eine Veranstaltung online buchen, damit die Mitarbeiter der Fa. Teachware von solchen Tätigkeiten entlastet werden.

2 Rahmenbedingungen

/LR10/ SemOrg ist eine kaufmännisch/administrative Web-Anwendung.
/LR20/ Zielgruppe sind die Mitarbeiter der Fa. Teachware (Kundensachbearbeiter, Seminarsachbearbeiter, Veranstaltungsbetreuer) sowie Interessenten und Kunden.

3 Kontext und Überblick

/LK10/ Das System besitzt eine Softwareschnittstelle zu einem Buchhaltungssystem.
/LK20/ Das System ist mindestens 20 Stunden pro Tag im Intranet der Fa. Teachware und im Internet verfügbar.

4 Funktionale Anforderungen

/LF10/ Das System *muss* Interessenten und Kunden die Möglichkeit bieten, sich über Seminare und Veranstaltungen zu informieren, Veranstaltungen zu buchen und einen Seminarkatalog anzufordern.
/LF20/ Das System *muss* dem Kundensachbearbeiter die Möglichkeit bieten, neue Kunden/Firmen zu erfassen und vorhandene Kunden-/Firmendaten zu aktualisieren und Kunden/Firmen zu löschen.
/LF30/ Das System *muss* dem Kundensachbearbeiter die Möglichkeit bieten, Seminare und Veranstaltungen zu selektieren, Veranstaltungen für Interessenten und Kunden zu buchen und für angeforderte Seminarkataloge Versandpapiere zu erstellen.
/LF40/ Das System *muss* dem Seminarsachbearbeiter die Möglichkeit bieten, neue Dozenten zu erfassen und vorhandene Dozentendaten zu aktualisieren und Dozenten zu löschen.

/LF50/ Das System *muss* dem Seminarsachbearbeiter die Möglichkeit bieten, neue Seminare und Veranstaltungen zu erfassen, vorhandene zu modifizieren oder zu löschen.
/LF60/ Das System *soll* dem Seminarsachbearbeiter die Möglichkeit bieten, für alle Veranstaltungen Termine festzulegen, Hotels auszuwählen und Räume zu reservieren.
/LF70/ Das System *muss* Kunden-, Firmen-, Seminar-, Veranstaltungs- und Dozentendaten permanent speichern.
/LF80/ Das System *muss* fähig sein, dem Buchhaltungssystem Rechnungsdatensätze mindestens einmal am Tag zur Verfügung zu stellen.

5 Qualitätsanforderungen

Systemqualität	sehr gut	gut	normal	nicht relevant
Funktionalität		X		
Zuverlässigkeit			X	
Benutzbarkeit		X		
Effizienz			X	
Wartbarkeit			X	
Portabilität				X

/LQE10/ Alle Reaktionszeiten auf Benutzeraktionen müssen unter 5 Sekunden liegen.

Glossar SemOrg

Version	Autor	Quelle	Status	Datum	Kommentar
0.1	Balzert	Geschäftsführer Teachware	in Bearbeitung	6/09	

Dozent: Führt als freier Mitarbeiter ein oder mehrere angebotene →Veranstaltungen durch. Ist fachlich in der Lage, ein oder mehrere →Seminare abzuhalten.
Firma: Mitarbeiter einer Firma (Ansprechpartner), der für die Aus- und Weiterbildung von Mitarbeitern zuständig ist und sich über Dienstleistungen informiert oder Mitarbeiter zu öffentlichen →Veranstaltungen schickt oder firmeninterne Veranstaltungen bucht.
Interessent: →Kunde, der sich für Dienstleistungen, z. B. Seminarkatalog, interessiert, aber noch an keiner →Veranstaltung teilgenommen hat.
Kunde: Mitarbeiter einer Firma oder Privatperson, der bzw. die an Dienstleistungen interessiert ist, oder ein Seminar bucht und besucht (→Teilnehmer, →Interessent).
Kundensachbearbeiter: Verantwortlich für die Kommunikation mit →Kunden und →Firmen einschließlich der Auskunftserteilung und Buchung.
Seminar: →Seminartyp
Seminarsachbearbeiter: Verantwortlich für die Planung und Terminierung von →Seminaren und →Veranstaltungen. Zuständig für die Kommunikation und Akquirierung von →Dozenten.
Seminartyp: Beschreibt die Gemeinsamkeiten, die eine Menge von →Veranstaltungen besitzen wie Titel, Zielsetzung, Inhalt, Voraussetzungen.
Seminarveranstaltung →Veranstaltung
Teilnehmer: →Kunde, der an einer →Veranstaltung teilnimmt bzw. teilgenommen

hat.
Veranstaltung: →Seminar, das zu einem festgelegten Zeitpunkt, an einem festgelegten Ort von einem oder mehreren →Dozenten durchgeführt wird.
Veranstaltungsbetreuer: Betreut die →Teilnehmer und →Dozenten einer →Veranstaltung.

16 Schablonen für agile Entwicklungen **

☐ *User Story*: Soll Dinge beschreiben, die das System für den Kunden tun soll.

☐ Beispiel: SemOrg

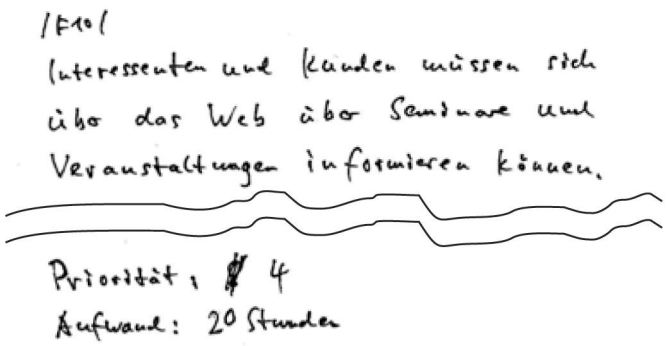
○ Interessenten und Kunden müssen sich über das Web über Seminare und Veranstaltungen informieren können.

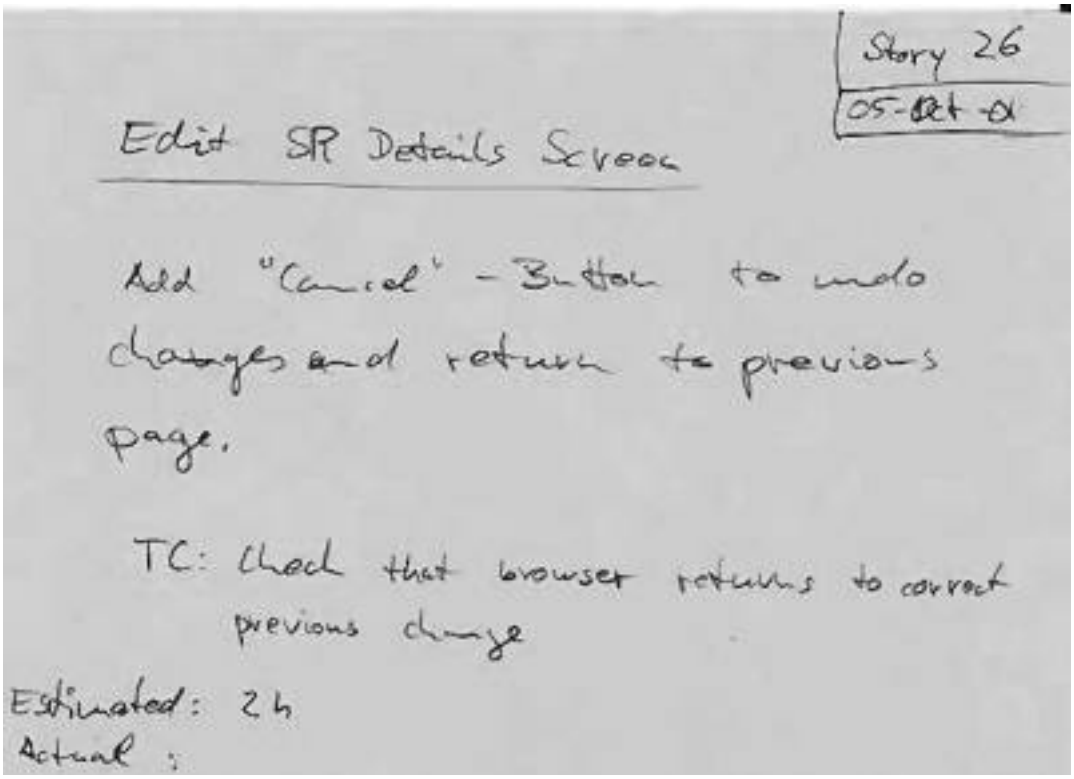
■ Jede *User Story* auf eine Karteikarte

☐ Nr. oder Identifikation der *User Story*.

☐ Priorität (vergeben durch den Kunden).

☐ Aufwand zur Implementierung der Anforderung.





☐ Abnahmetests formulieren

☐ Textschablonen:
»*As a [role] I want [something] so that [benefit].*«

»*As a [user role], I want to [goal], so I can [reason].*«

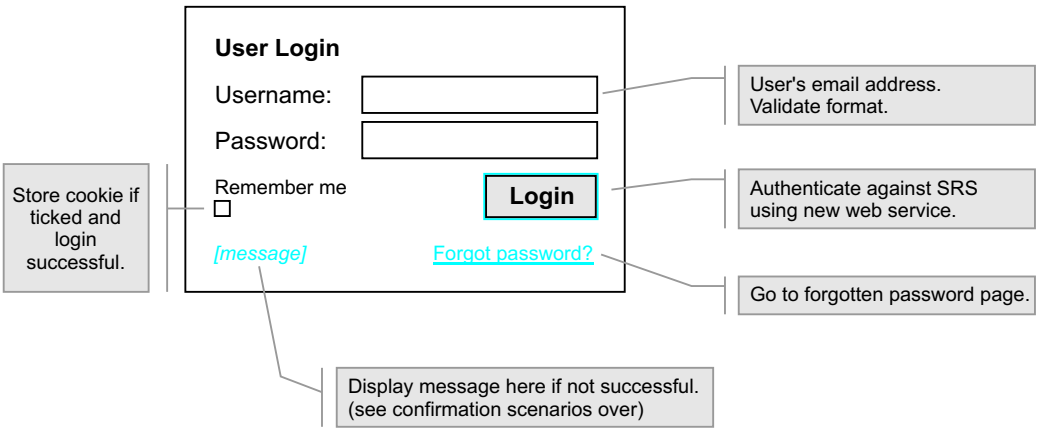
■ Aufteilung einer Karteikarte in 3 Sektionen (3 C's):

☐ *Card section* : Auf der Vorderseite der Karte steht oben der Name und die Beschreibung der Anforderung, irgendeine Referenznummer, der geschätzte Aufwand usw.

☐ *Conversation section* : Enthält mehr Informationen über die Anforderung.

☐ *Confirmation section* : Auf der Rückseite der Karte stehen Abnahmetests für die Anforderung.

#0001	USER LOGIN	Fibonacci Size # 3
As a **[registered user]**, I want to **[log in]**, so I can **[access subscriber content]**.		



Confirmation

1. Success valid user logged in and referred to home page.
   a. 'Remember me' ticked store cookie / automatic login next time.
   b. 'Remember me' not ticked force login next time.

2. Failure display message:
   a) "Email address in wrong format"
   b) "Unrecognised user name, please try again"
   c) "Incorrect password, please try again"
   d) "Service unavailable, please try again"
   e) Account has expired refer to account renewal sales page.

■ Checkliste *User Story*

☐ **I**ndependent

☐ **N**egotiable

☐ **V**aluable

☐ **E**stimatable

☐ **S**mall

☐ **T**estable

■ RE-Prozess

■ Bewertung

✚ Leicht zu erstellen.

+ Enger Kundenkontakt.
– Kunden sind keine Profis für das Schreiben von Anforderungen.
– Überblick über das gesamte zu entwickelnde System ist schwer zu erhalten.
– Starke Abhängigkeit von kompetenten Entwicklern.
– Schätzung des Aufwands schwierig.

17 Anforderungen ermitteln und spezifizieren **

Stakeholder finden

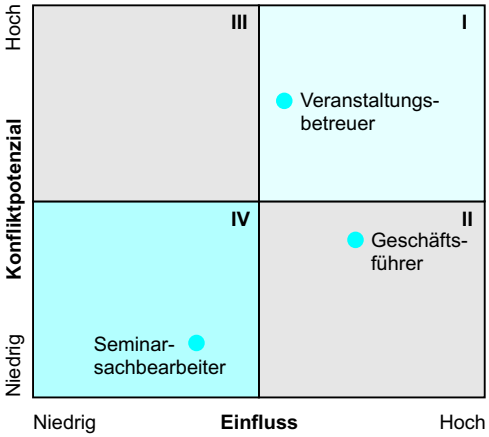
☐ Permanente Aufgabe

☐ *Stakeholder* bewerten

☐ Beispiel: SemOrg

Stakeholder	Erwartung	Klima/Stimmung	Bedeutung/Macht
Geschäftsführer	Hat als Auftraggeber & Geldgeber großes Interesse das Produkt zum Erfolg zu bringen.	+	5
Veranstaltungsbetreuer	Hat Angst zu viel am Computer arbeiten zu müssen.	–	2
Seminarsachbearbeiter	Kommt auch ohne die neue Software gut aus.	o	3
Legende		+ = positiv, o = neutral, – = negativ	5 = hoch, 1 = gering

Tab. 17.0-1:



☐ Persönlichkeitstypen

☐ Risiken

■ Einbindung

■ Ergebnisse

☐ Stakeholder-Analyse

☐ Stakeholder-Portfolio

☐ Stakeholder-Maßnahmenkatalog

Projektumfeld ermitteln

☐ Definition:
Das Projektumfeld ist »das Umfeld, indem ein Projekt entsteht und durchgeführt wird, das das Projekt beeinflusst und von dem es beeinflusst wird« (DIN 69904).

☐ Kreativitätstechniken

☐ Darstellung durch *Mind Maps*

■ Ergebnis

☐ Liste von Elementen des Projektumfelds, alternativ als *Mind Map*.

18 Anforderungen priorisieren **

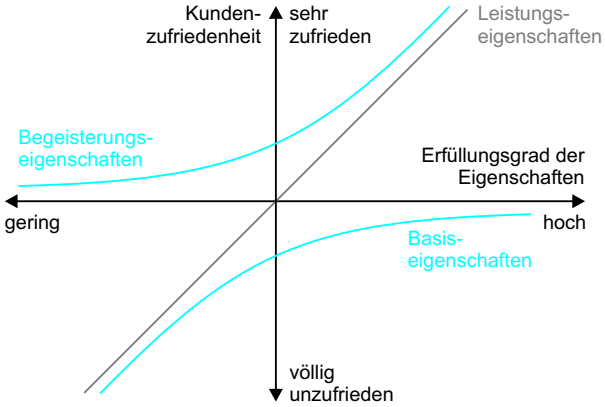
Klassifikation nach einem Kriterium

☐ **Essenziell**

☐ **Bedingt notwendig**

☐ **Optional**

Die Kano-Klassifikation



☐ Beispiel

1 "Integriertes GPS im Mobiltelefon"
Identifikation einer Produkteigenschaft

Funktionale Frage

Wie bewerten Sie das Vorhandensein eines GPS in einem Mobiltelefon?	1. gewünscht	X
	2. muss	
	3. interessiert nicht	
	4. damit kann ich leben	
	5. unerwünscht	

Dysfunktionale Frage

Wie bewerten Sie das Fehlen eines GPS in einem Mobiltelefon?	1. gewünscht	
	2. muss	
	3. interessiert nicht	
	4. damit kann ich leben	X
	5. unerwünscht	

2 *Formulierung der Fragen und Befragung der potenziellen Kunden*

3 *Auswertung*

3 *Auswertung*

Produkteigenschaft		dysfunktional: 1. gewünscht	2. muss	3. interessiert nicht	4. damit kann ich leben	5. unerwünscht
funktional	1. gewünscht	F	B	B	**B**	L
	2. muss	Ik	Id	Id	Id	Ba
	3. interessiert nicht	Ik	Id	Id	Id	Ba
	4. damit kann ich leben	Ik	Id	Id	Id	Ba
	5. unerwünscht	Ik	Ik	Ik	Id	F

L = Leistungseigenschaft
Id = Indifferente Antwort
B = Begeisterungseigenschaft
Ba = Basiseigenschaft
Ik = Inkonstistente Antwort
F = Fragliche Antwort

4 *Identifikation der Eigenschaftskategorie der Produkteigenschaft*

19 Resümee *

Denkfallen des Auftraggebers

☐ Visionen & Ziele werden sich im Laufe der Anforderungserstellung schon ergeben.

☐ Rahmenbedingungen legt der Auftragnehmer fest.

☐ »Ich will die beste Qualität die möglich ist« oder
Über Qualität wird nicht nachgedacht.

☐ Abnahmekriterien müssen erst nach der Fertigstellung des Systems erstellt werden oder
Abnahmekriterien erstellt der Auftragnehmer.

☐ Anforderungen lasen sich natürlichsprachlich präzise beschreiben, wenn man sich nur Mühe gibt.

☐ Man nimmt einen Standard – z. B. IEEE 830 – und ist damit alle Probleme los.

☐ Konflikte werden sich im Laufe der Entwicklung schon auflösen.

☐ Es reicht, funktionale Anforderungen festzulegen.

Was kann der Auftraggeber alles selbst erledigen?

☐ Ermitteln und bewerten aller *Stakeholder*.

☐ Konfliktpotenziale der *Stakeholder* ermitteln und (Ziel-)Konflikte lösen – nicht aufschieben.

☐ Projektumfeld ermitteln.

☐ Anforderungen anhand einer Schablone ermitteln und Textschablonen verwenden.

☐ Anforderungen priorisieren.

☐ Sich durch externe RE-Experten beim Lastenheft unterstützen lassen, wenn kein eigenes Know-how vorhanden ist.

Ziel: Das richtige Produkt spezifizieren, damit es richtig entwickelt werden kann!

Effektivität *und* Effizienz!

20 Zum guten Schluss *

»Ein kluger Mann macht nicht alle Fehler selbst.
Er gibt auch anderen eine Chance.«

»Durch Fehler wird man zwar klug –
aber man muss nicht alle Fehler selber machen!«

Literatur

/Ambl08/
Ambler, Scott W.; *Beyond Functional Requirements On Agile Projects*, in: Dr. Dobb's Journal, October 2008, 2008, S. S. 64–66.

/ASC07/
Azar, Jim; Smith, Randy K.; Cordes, David; *Value-Oriented Requirements Prioritization in a Small Development Organization*, in: IEEE Software, January/February 2007, 2007, S. 32–37.

/Balz08/
Balzert, Helmut; *Lehrbuch der Softwaretechnik – Softwaremanagement, 2. Auflage*, Heidelberg, Spektrum Akademischer Verlag, 2008.

/BlCl08/
Blaine, J. David; Cleland-Huang, Jane; *Software Quality Requirements: How to Balance Competing Priorities*, in: IEEE Software, March/April 2008, 2008, S. S. 22–24.

/Cohn04/
Cohn, Mike; *User Stories Applied: For Agile Software Development*, Amsterdam, Addison-Wesley Longman, 2004.

/DIN69905/
DIN 69905 – Projektabwicklung – Begriffe, Berlin, Beuth Verlag GmbH, 1997.

/Eber08/
Ebert, Christof; *Systematisches Requirements Engineering und Management*, Heidelberg, dpunkt.verlag, 2008.

/FMS08/
Friedenthal, Sanford; Moore, Alan; Steiner, Rick; *OMG Systems Modeling Language (OMG SysML) Tutorial*, 2008, `http://www.omgsysml.org/INCOSE-2008-OMGSysML-Tutorial-Final-revb.pdf`.
Abgerufen am 8.4.2009.

/Glin08/
Glinz, Martin; *A Risk-Based, Value-Oriented Approach to Quality Requirements*, in: IEEE Software, March/April, 2008, S. 34–41.

/GlWi07/
Glinz, Martin; Wieringa, Roel J.; *Stakeholders in Requirements Engineering*, in: IEEE Software, March/April, 2007, S. 18–20.

/HoLe01/
Hofmann, Hubert F.; Lehner, Franz; *Requirements Engineering as a Sucess Factor in Software Projects*, in: IEEE Software, July/August, 2001, S. 58–66.

/IEEE830/
IEEE Standards Board; *IEEE recommended practice for software requirements specifications*, IEEE Press, 1998.

/IREB07/
Lehrplan IREB Certified Professional for Requirements Engineering – Foundation Level Version: 1.3, Hrsg. International Requirements Engineering Board, 2007, `http://certified-re.de/de/files/cpre-fl-lehrplan_20080110.pdf`.
Abgerufen am 2.4.2009.

/JaMa02/
Jarke, Matthias; Mayr, Heinrich C.; *Mediengestütztes Anforderungsmanagement*, in: Informatik-Spektrum, 16. Dezember, 2002, S. 452–464.

/JiLu06/
Jirotka, Marina; Luff, Paul; *Supporting Requirements with Video-Based Analysis*, in: IEEE Software, May/June, 2006, S. 42–44.

/KaRy97/
Karlsson, K.; Ryan, K.; *A Cost-Value Approach for Prioritizing Requirements*, in: IEEE Software, Vol. 14, No. 5, 1997, S. 67–74.

/KMS05/
Kowallik, Petra; Müller-Prove, Matthias; Strauß, Friedrich; *Requirements-Engineering im Spannungsfeld von Individual- und Produktsoftware*, in: i-com, Nr. 3, 2005, S. S. 41–46.

/KTS+84/
Kano, N.; Tsuij, S.; Seraku, N.; Takahashi, F.; *Attractive Quality and Must-be-Quality*, in: Quality – The Journal of the Japanese Society for quality Control, Vol. 14, No. 2, 1984, S. 39–44.

/Lang02/
Benutzergeschichte, Hrsg. Lange, Manfred, 2002, `http://www.xpexchange.net/german/intro/userStory.html`.
Abgerufen am 10.11.2008.

/Mott09/
Motte, Petra; *Moderieren, Präsentieren, Faszinieren*, Herdecke, Witten, W3L-Verlag, 2009.

/NuEa07/
Nui, Nan; Easterbrook, Steve; *So, You Think You Know Others' Goals? A Repertory Grid Study*, in: IEEE Software, March/April 2007, 2007, S. 53–61.

/OBN+08/
Ozkaya, Ipek; Bass, Len; Nord, Robert L.; Sangwan, Raghvinder S.; *Making Practical Use of Quality Attribute Information*, in: IEEE Software, March/April, 2008, S. 25–33.

/Pohl07/
Pohl, Klaus; *Requirements Engineering*, Heidelberg, dpunkt.verlag, 2007.

/Rupp07/
Rupp, Chris; *Requirements-Engineering und -Management*, 4. Auflage, München, Hanser, 2007.

/Schr05/
Schröder, Marion; *Heureka, ich hab's gefunden – Methoden für die kreative Problemlösung, Ideenfindung & Alternativenauswahl*, Herdecke, W3L-Verlag, 2005.

/SLM07/
Straube, R.; Leuschner, H. D.; Müller, P.; *Konfliktmanagement für Projektleiter*, Freiburg, Haufe-Verlag, 2007.

/VDI2519/
Lasten-/Pflichtenheft für den Einsatz von Förder- und Lagersystemen, in: VDI-Handbuch Materialfluss und Fördertechnik, Band 8, Berlin, Beuth Verlag GmbH, 2001.
Obwohl diese VDI-Richtlinie nicht für die Entwicklung von Softwaresystemen vorgesehen ist, definiert sie die Begriffe Lasten- und Pflichtenheft auch für die Softwareentwicklung treffend.

/VPAL08/
V-Modell XT Produktvorlage Anforderungen (Lastenheft), 2008, `http://www.v-modell-xt.de/`.

/VPGP08/
V-Modell XT Produktvorlage Gesamtsystemspezifikation (Pflichtenheft), 2008, http://www.v-modell-xt.de/.

/VXTLG08/
V-Modell XT Produktvorlage Lastenheft Gesamtprojekt, 2008, http://www.v-modell-xt.de/.

/Wate08/
Waters, Kelly; *Introducing User Stories*, http://groups.google.com/group/allaboutagile/files.

/Wate08b/
Waters, Kelly; *Writing Good User Stories*, 2008, http://www.agile-software-development.com/2008/04/writing-good-user-stories.html.
Abgerufen am 10.11.2008.

/Weil06/
Weilliens, Tim; *Requirements-Engineering mit der UML*, in: Objekt-Spektrum, 1, 2006, S. 42–44.

/Wieg99/
Wiegers, K. E.; *Software Requirements*, Redmond, Microsoft Press, 1999.

/WMH07/
Woolridge, Richard W.; McManus, Denise J.; Hale, Joanne E.; *Stakeholder Risk Assessment: An Outcome-Based Approach*, in: IEEE Software, March/April, 2007, S. 36–45.

/WSH07/
Wentzel, Paul-Roux; Seckinger, Otmar; Hindel, Bernd; *Requirements-Engineering aus der Sicht der Persönlichkeitsanalyse*, in: Objekt-Spektrum, 5/2007, 2007, S. 34–41.