

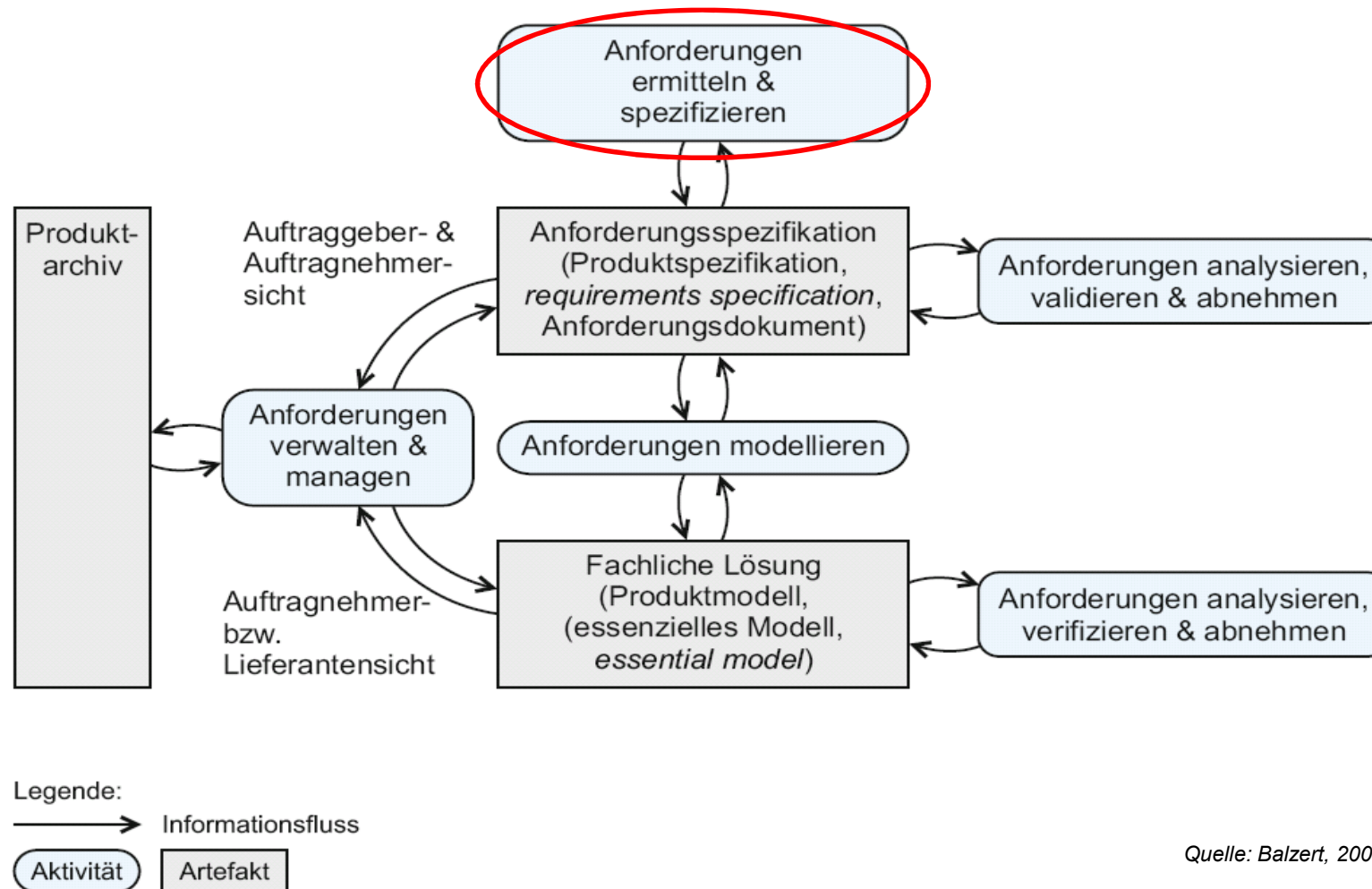
- 01 Einführung
- 02 Prozessmodelle
- 03 Konfigurationsmanagement
- 04 Requirements Engineering
 - 04.1 Aktivitäten, Artefakte und Anforderungsarten
 - 04.2 Methoden und Techniken
- 05 Modellierung
- 06 Qualitätsmanagement

Nach dieser Vorlesungseinheit ...

- ... haben Sie ein detaillierteres **Verständnis** für die **einzelnen durchzuführenden Aktivitäten** bzw. Schritte im Requirements Engineering
- ... haben Sie einen Überblick über **wichtige Methoden und Techniken** des Requirements Engineering

Rückblick – Aktivitäten im RE

04 Requirements Engineering / 04.2 Methoden und Techniken

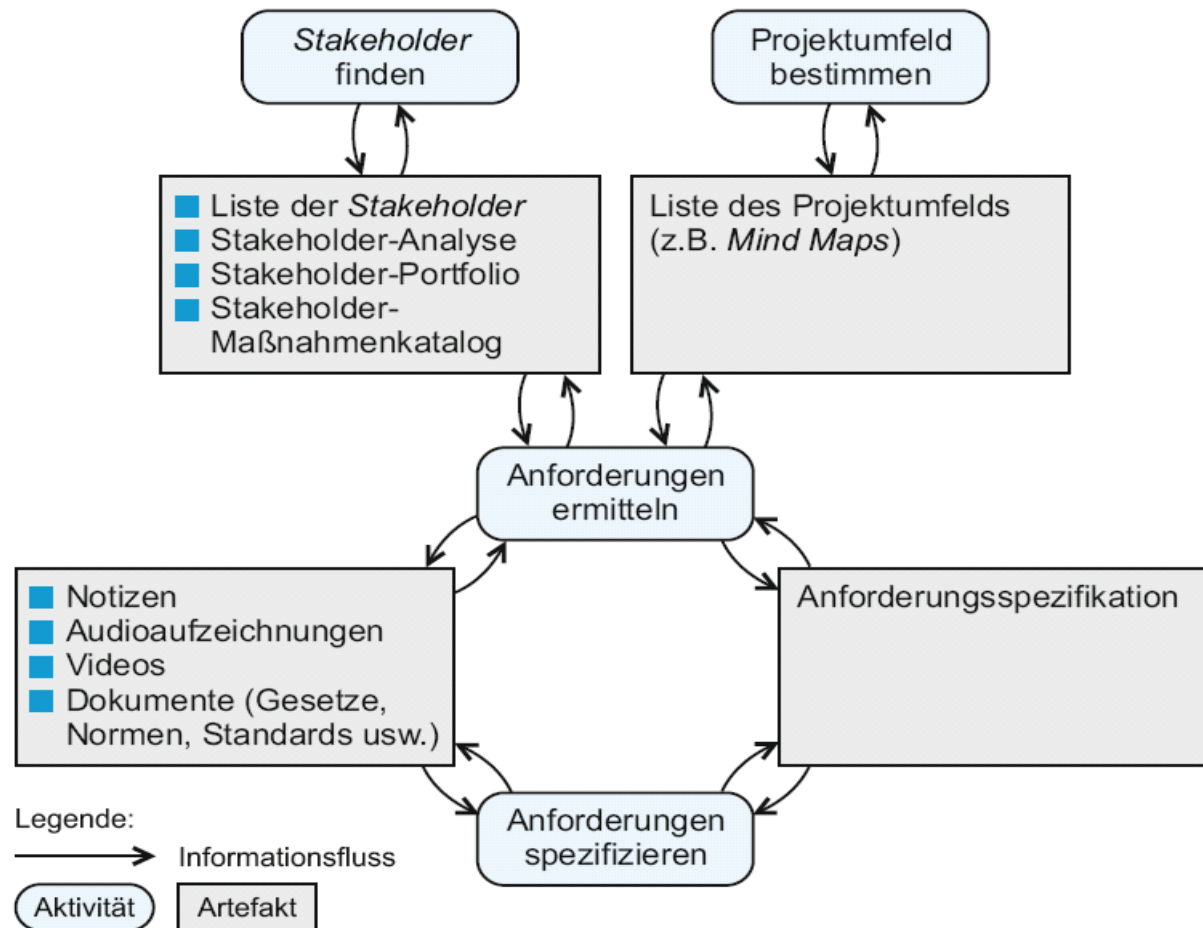


Quelle: Balzert, 2009, S.446

Anforderungen ermitteln und spezifizieren

04 Requirements Engineering / 04.2 Methoden und Techniken

● Vorgehensweise



Quelle: Balzert, 2009, S.503



Stakeholder finden (1)

04 Requirements Engineering / 04.2 Methoden und Techniken

- Beteiligte, Betroffene, Nutzer und Interessierte an der Software identifizieren
- Personen bzw. Personengruppen **beeinflussen** (+/-) sowohl die **Entwicklung** als auch **Einführung** und **Betrieb**
 - Sollten frühzeitig in den RE-Prozess eingebunden werden
 - Liefern Informationen bzgl. Ziele, Rahmenbedingungen, Kontext, Anforderungen als auch Risiken
- **Permanente Aufgabe** – stets überprüfen ob es weitere Stakeholder gibt
 - **Überblick** verschaffen und deren Einstellung und Macht abschätzen



Stakeholder finden (2)

04 Requirements Engineering / 04.2 Methoden und Techniken

● Beispiel: Stakeholder-Analyse

Stakeholder	Erwartung	Klima, Stimmung	Bedeutung, Macht
Geschäftsführer	<i>Als AG und Geschäftsführer großes Interesse am Erfolg des Produkts.</i>	+	5
Veranstaltungsbetreuer	<i>Fürchtet zu viel am Rechner arbeiten zu müssen.</i>	–	2
Seminarsachbearbeiter	<i>Kommt auch ohne die neue SW gut aus.</i>	0	3
Legende		+ ... positiv o ... neutral – ... negativ	5 ... hoch 1 ... gering

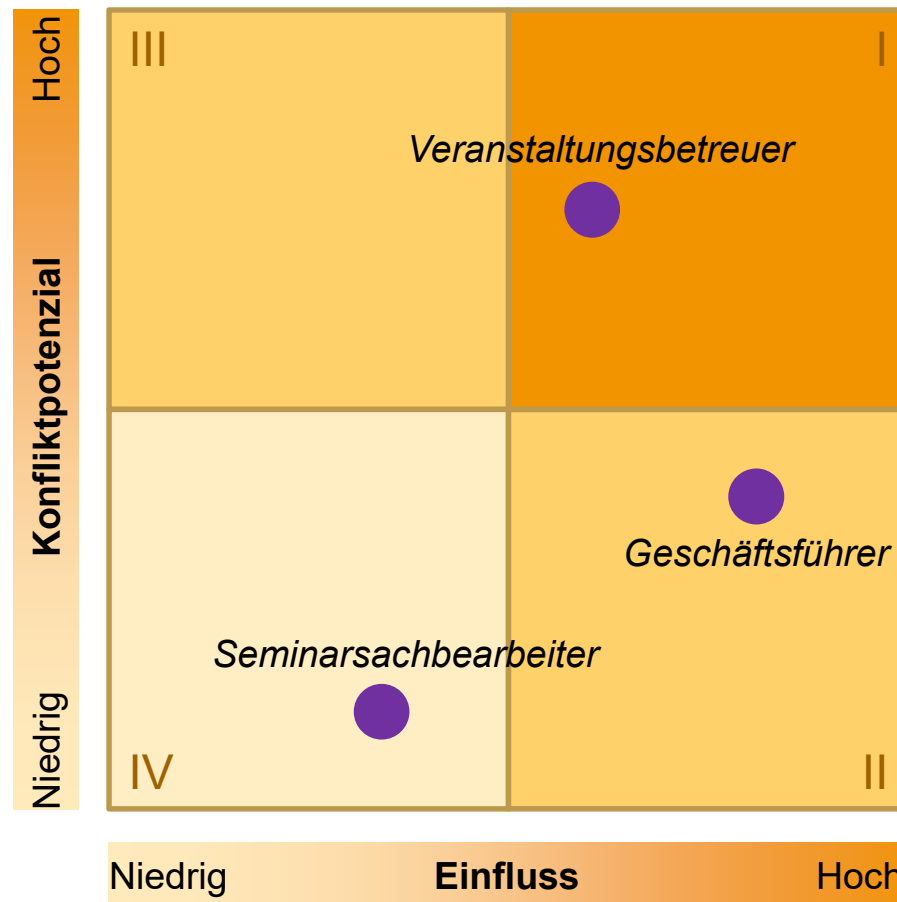
Quelle: Balzert, 2009, S.504



Stakeholder finden (3)

04 Requirements Engineering / 04.2 Methoden und Techniken

● Visualisierung durch Portfoliodarstellung



Quelle: Balzert, 2009, S.505

- Was passiert mit der Einschätzung?
 - Stakeholder im rechten oberen Quadranten müssen ständig im Blickfeld des Projektleiters sein
 - Formulierung von **Maßnahmen** um diese positiv zu beeinflussen
 - Einbinden in das Projekt – aus Betroffenen müssen Beteiligte gemacht werden
 - In die Projektkommunikation einbinden
 - Verantwortung und Aufgaben übertragen
 - Zu wichtigen Projektbesprechungen einladen
 - Sammeln von Informationen zu den Stakeholdern
 - Beziehungen der Stakeholder versuchen zu erfassen (konträre Ziele?)

- Umfeld, in dem ein Projekt entsteht und durchgeführt wird
- Alle Elemente, die ein Projekt wesentlich beeinflussen:
 - Vorgaben, Gesetze, Standards, Normen, ökologische, ökonomische, gesellschaftliche und kulturelle Einflüsse, Gesetzgebung, ...
- Besonders geeignet
 - Anwendung von **Kreativitätstechniken** im Projektteam, z.B. **Brainstorming**
 - Darstellung des Projektumfelds in **Mind Maps**

● Befragungstechniken

- Strukturiertes Interview
(Einzelinterview ↔ Gruppeninterview / Moderator-Rolle)
- Fragebögen
- Selbstaufschreibung (Tätigkeiten, Arbeitsabläufe, Wünsche)
- On Site Customer
(Repräsentant des AG arbeitet vor Ort im Projektteam mit)

● Beobachtung

- Feldbeobachtung – Fokus: Arbeit des Benutzers
- Lehrlingsrolle – Requirements Engineer erlernt unter Anleitung des Benutzers dessen Tätigkeiten



● Vergangenheitorientiert

● Analyse eines vorhandenen Altsystems

- Meist reicht Verstehen des Altsystems aus (black box)
- Ermittlung von Informationen bzgl.
 - Prozessabläufe,
 - Ein- und Ausgabedaten und
 - Funktionenanhand Bedienung und Benutzerhandbuch
- Ggf. Untersuchung des Datenbankmodells (von SQL-Tabellen)
- Evtl. Wrapping von Teilen des Altsystems
 - Verpackung nach außen hin durch Software-Schicht

● Wiederverwendung von Anforderungsartefakten aus anderen Projekten

Anforderungen ermitteln - Techniken (3)

04 Requirements Engineering / 04.2 Methoden und Techniken

● Kreativitätstechniken

- Brainstorming
- Mind Mapping
- Perspektivenwechsel - „Brille anderer Stakeholder“

● Feedbacktechniken

- Simulationsmodelle
- Prototypen
- Anforderungsreview

● Unterstützende Techniken

- Workshops
- Raten von Anforderungen

**Abhängig von Zeit,
Wissen und Motivation
des Kunden**



- Dokumentation der Anforderungen
- Wie aufschreiben?
 - Natürliche Sprache, Tabellen
 - Strukturierte natürliche Sprache
 - Sprachliche Anforderungsschablonen, Formulare
 - (UML-) Modelle

Beispiel: Natürliche Sprache – Story Card

04 Requirements Engineering / 04.2 Methoden und Techniken

Customer Story and Task Card BLW Development / COLA

DATE: 3/19/98 TYPE OF ACTIVITY: NEW: ☒ FIX: ☐ ENHANCE: ☐ FUNC. TEST: ☐

STORY NUMBER: ~~1275~~ 1275 PRIORITY: USER: ☐ TECH: ☐

PRIOR REFERENCE: RISK: TECH ESTIMATE:

TASK DESCRIPTION:
SPLIT COLA: When the COLA rate chgs. in the middle of the BLW Pay Period, we will want to pay the 1ST week of the pay period at the OLD COLA rate and the 2ND week of the Pay Period at the NEW COLA rate. Should occur automatically based on system design.

NOTES:
For the OT, we will run a m/frame program that will pay or calc the COLA on the 2ND week of OT. The plant currently retransmits the hours data for the 2ND week exclusively so that we can calc COLA. This will come into the Model as a "2144" COLA

TASK TRACKING: Gross Pay Adjustment. Create RM Boundary and Place in DEEnt Excess COLA

Date	Status	To Do	Comments	BIN

Quelle: http://ronjeffries.com/xprog/articles/story_and_task_cards/



- Idee stammt aus dem eXtreme Programming
- Kunde dokumentiert Anforderungen als Text
 - Ca. 3 Sätze
 - Sehr informell
 - auf Karteikarte
- Detaillierte Abläufe nur mündlich vermittelbar
- Verwendbar, wenn Kunde vor Ort
 - Unklarheiten können sofort geklärt werden
 - Kunde gibt direktes Feedback auf die Umsetzung
 - Umsetzung zeitnah zur Spezifikation
- Kann gut in Workshops verwendet werden

Beispiel natürliche Sprache: Tabellen

04 Requirements Engineering / 04.2 Methoden und Techniken

Pflicht, Optional,
Vorschlag, Zukunft

Nr.	Priorität	Beschreibung	Quelle	Ab- gestimmt?
A1	Pflicht	Der telefonische Bestellvorgang muss in der Filiale nur über Tastatur oder Touchscreen (ohne Maus) abgewickelt werden können.		
A2	Optional	Der Bestellvorgang im Internet soll sicher sein: Die vom Kunden eingegebene Kreditkartennummer darf nicht von externen Angreifern gelesen werden können.		
A3	Zukunft	Es ist beabsichtigt, eine neue Dienstleistung „Partyservice“ anzubieten.		
...		...		

Eindeutige Nummer

- Beispiel: Auszug Pflichtenheft eines CASE-Werkzeugs zur Bearbeitung von Softwareentwurfsmodellen

2.6 Gitternetzfunktion

Um die Positionierung von Objekten in einem Diagramm zu erleichtern, kann der Benutzer mithilfe einer Option in einem Dialogfeld eine Gitterfunktion aktivieren, die als Einheit entweder Zentimeter oder Zoll verwendet. Am Anfang ist das Gitternetz deaktiviert. Das Gitterfenster kann zu jeder Zeit der Bearbeitungssitzung an- und abgeschaltet werden, und es ist jederzeit möglich, zwischen Zentimetern und Zoll zu wechseln. Das Gitternetz wird auch in der angepassten Ansicht bereitgestellt, aber die Zahl der angezeigten Gitternetzlinien wird reduziert, um zu vermeiden, dass das kleinere Diagramm mit Linien ausgefüllt wird.

Quelle: Sommerville (2007), S. 160



- Regeln für den Einsatz natürlicher Sprache
 - Anforderungen immer **im Aktiv** formulieren
 - **Unvollständige Vergleiche** vermeiden
 - „Das System soll leicht bedienbar sein!“
(Im Vergleich wozu?)
 - Vorsicht bei **Allquantoren** und Verallgemeinerungen (alle, immer, überall, jeder, ...)
 - z.B. „Das System soll immer online sein“
 - Vorsicht bei **unvollständigen Fallunterscheidungen**
 - z.B. „Wenn ein Systemalarm eintritt, ...“
 - **Keine Bandwurmsätze** schreiben!

- Welche **Vorteile** bietet die natürliche Sprache?
 - Verwendung ist **einfach**
(Voraussetzung: alle Stakeholder beherrschen die jeweilige Sprache, z.B. Deutsch)
 - Abstrakte und konkrete Dinge können beschrieben werden -
ist **flexibel**
 - In jeder Anwendungsdomäne einsetzbar – sie ist **universell**

- **Probleme** bei der Verfassung von Anforderungen in natürlicher Sprache
 - **Mangel an Genauigkeit**
 - Keine präzisen und widerspruchsfreien Formulierungen
 - **Verwirrung bei Anforderungen**
 - Funktionale, nicht-funktionale, Systemziele und Entwurfsinformationen können nicht klar unterschieden werden
 - **Verschmelzung von Anforderungen**
 - Zusammenfassung verschiedener Anforderungen in einer
 - **Mehrdeutigkeit natürlicher Sprache**
 - Unterschiedliche Verwendung und Interpretation von Begriffen
→ Missverständnisse

● Beispiele Mehrdeutigkeit

- *„Die letzten 10 Buchungen und die Stornierungen des Kunden werden in einem Fenster angezeigt.“*
(syntaktische Mehrdeutigkeit)
- *„Jeder Sensor ist mit einem Service verbunden.“*
(semantische Mehrdeutigkeit)
- *„Beim Login muss zuerst das Benutzerkennzeichen und dann das Passwort eingegeben werden. Ist dies nicht korrekt, schlägt die Anmeldung fehl.“*
(referentielle Mehrdeutigkeit)
- *„Der Sensor muss neben der Tür angebracht werden.“*
(Vage Begriffe – lexikalische Mehrdeutigkeit)

- Möglichkeiten, um die Probleme der natürlichen Sprache zu reduzieren bzw. zu vermeiden
 - Erstellung Glossar
 - Strukturierte natürliche Sprache
 - Semiformale und formale Beschreibungskonzepte

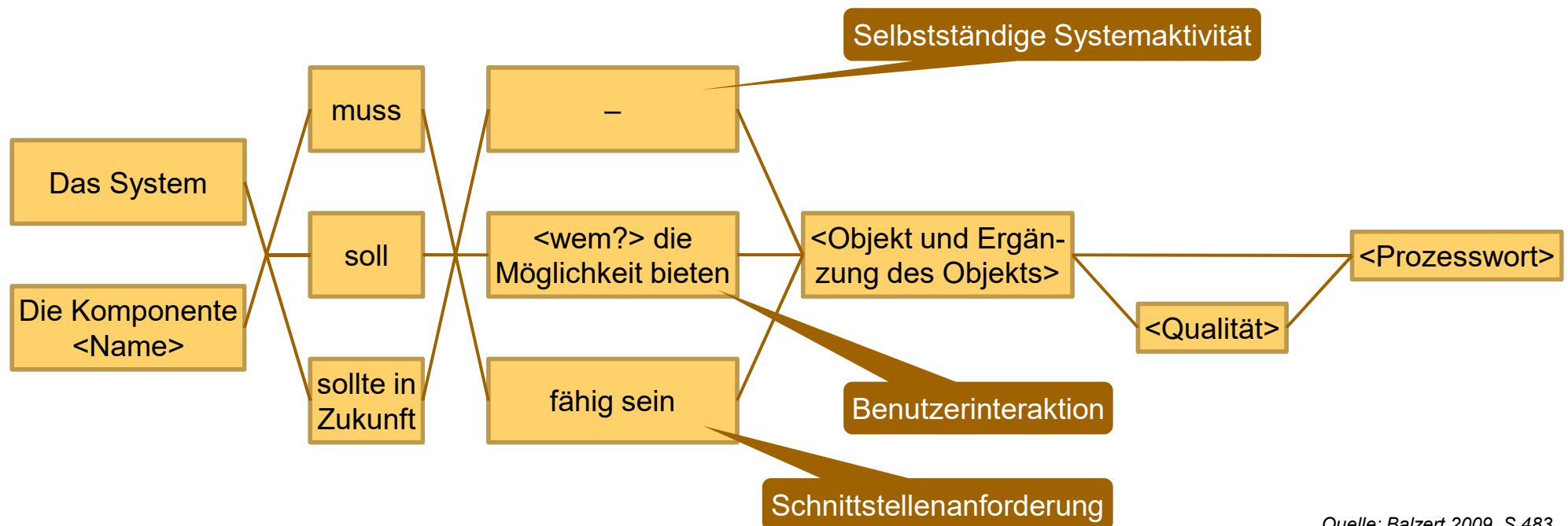
- Zielsetzung: Definition von Begriffen um lexikalische Mehrdeutigkeiten zu vermeiden
- Möglicher Aufbau
 - **Begriff:** zu definierender Begriff
 - **Definition:** Text der Definition
 - **Synonyme:** Begriff mit gleicher Bedeutung
 - **Plural:** Wenn ungewöhnlich, z.B. Algorithmus, Algorithmen
 - **Kurzform:** Wenn vorhanden, z.B. QM für Qualitätsmanagement
 - **Langform:** Wenn der Begriff in Kurzform angegeben wird, z.B. World Wide Web für WWW
 - **Übersetzung:** in eine oder mehrere Sprachen

Strukturierte natürliche Sprache (1)

04 Requirements Engineering / 04.2 Methoden und Techniken

● Idee

- Vorgabe einer Grammatik für die Beschreibung von Anforderungen
- Normsprache → Sprachliche Anforderungsschablonen
- Qualität = i.d.R. nichtfunktionale Anforderung (z.B. permanent, ...)
- Prozesswort = geforderte Funktionalität (z.B. drucken, speichern, ...)



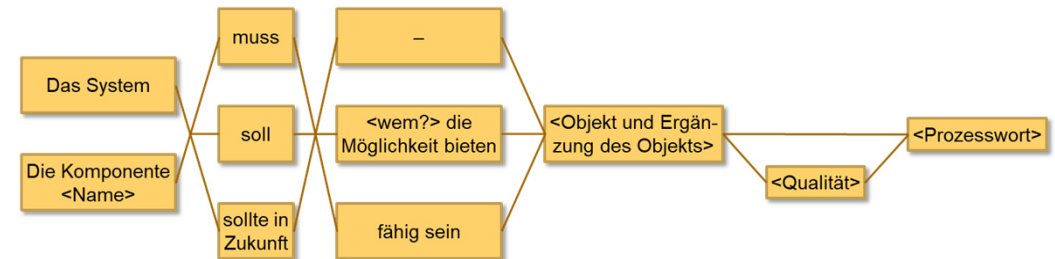
Quelle: Balzert 2009, S.483



Strukturierte natürliche Sprache (2)

04 Requirements Engineering / 04.2 Methoden und Techniken

● Beispiele

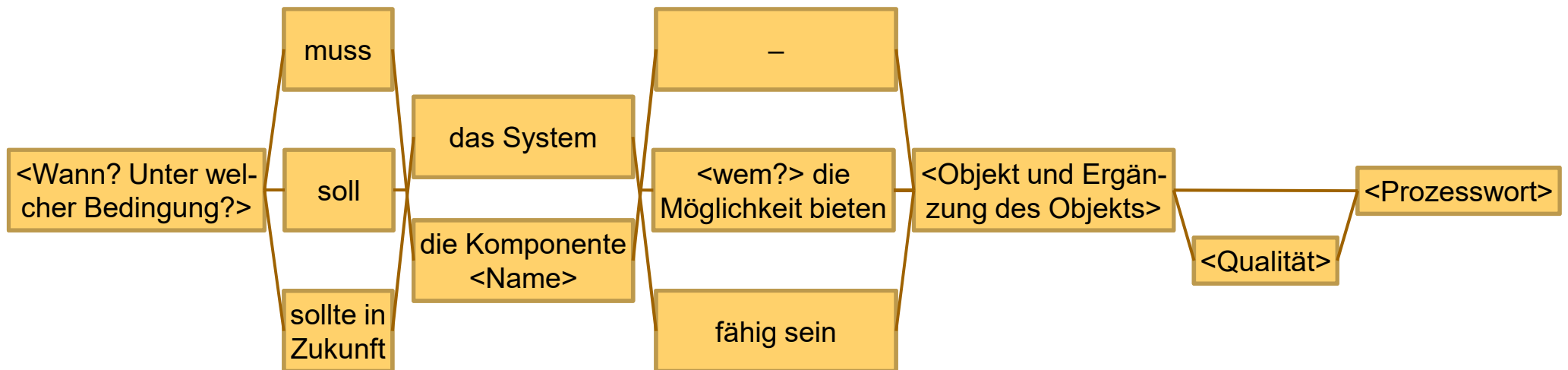


- *Das System muss dem Kunden die Möglichkeit bieten, sich über Seminare und Veranstaltungen zu informieren.*
(Benutzerinteraktion)
- *Das System soll dem Seminarsachbearbeiter die Möglichkeiten bieten, Seminare und Veranstaltungen mit selbst erstellten Suchanfragen auszuwerten.*
(Benutzerinteraktion)
- *Das System muss die Kundendaten permanent speichern.*
(Selbständige Systemaktivität)
- *Das System muss fähig sein, dem Buchhaltungssystem Rechnungsdatensätze mindestens einmal am Tag zur Verfügung zu stellen.*
(Schnittstellenanforderungen)

Strukturierte natürliche Sprache (3)

04 Requirements Engineering / 04.2 Methoden und Techniken

● Aufbau einer Anforderungsschablone mit Bedingung



Quelle: Balzert 2009, S.484

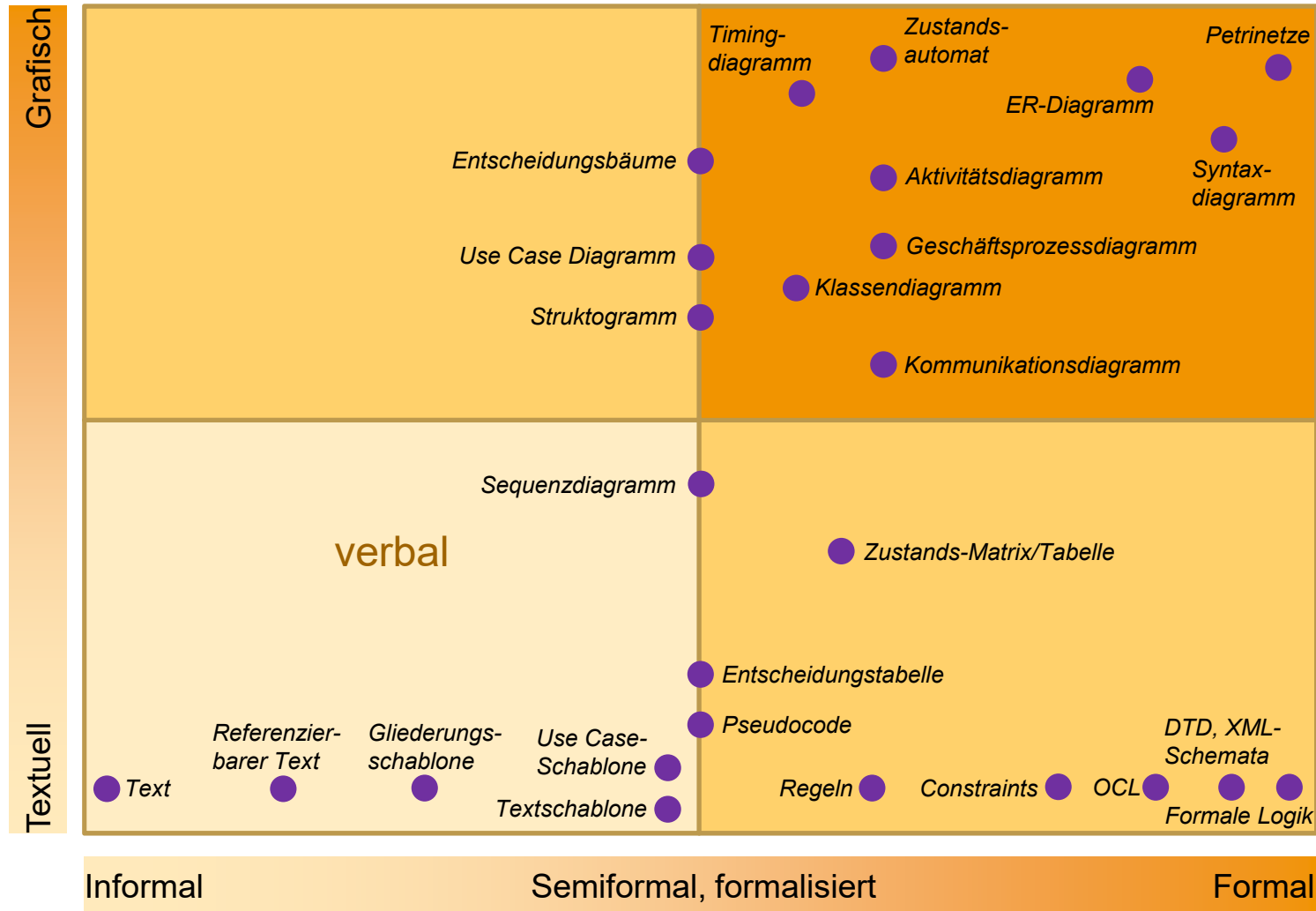
● Beispiel

- *Falls ein Kunde im Zahlungsverzug ist, muss das System eine neue Buchung nicht erlauben.*



Überblick Beschreibungskonzepte

04 Requirements Engineering / 04.2 Methoden und Techniken



Quelle: Balzert, 2009, S.101

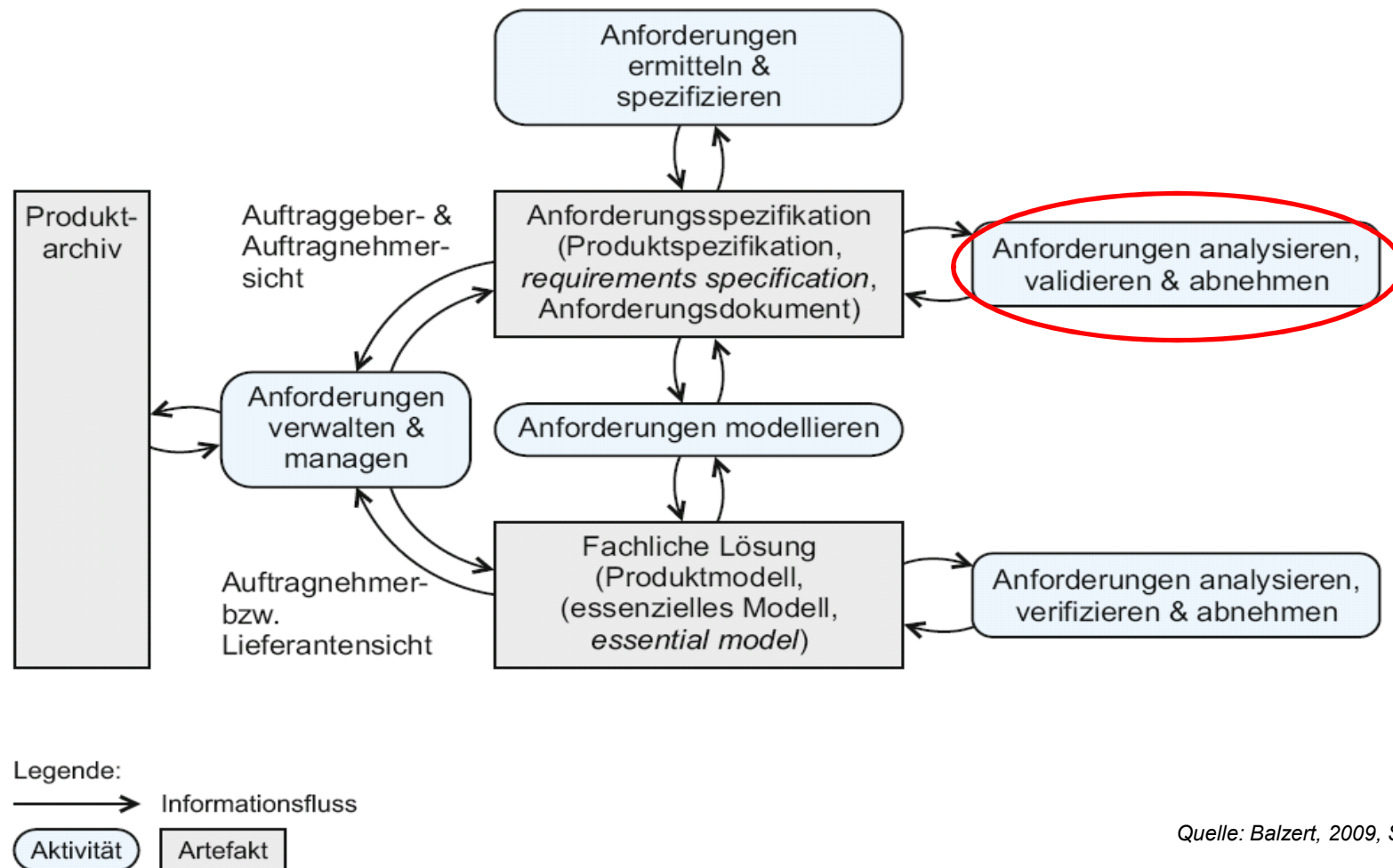


- Modelle, z.B.
 - Anwendungsfälle, Use Cases (+ Aktivitätsdiagramme)
 - Datenmodell, Klassendiagramme
 - Prototyp der grafischen Oberfläche
- = Formalere Beschreibung der Anforderungen
 - Eindeutiger, präziser als natürliche Sprache
 - Simulation, Analyse teilweise möglich
 - Weiterverwendung zur Generierung von Quelltexten
- Erstellung eines Modells der Anforderungen bzw. des geplanten Systems
 - Siehe Kapitel „Modellierung“ dieser Vorlesung

- Ergebnis bzw. resultierendes Artefakt
- Anforderungsspezifikation
 - Evtl. aufgeteilt in Lasten- und/oder Pflichtenheft bzw.
 - User Stories
 - Gliederung entsprechend einer Anforderungsschablone – mit integriertem oder separatem Glossar
- Spezifikation
 - „Ausführliche Beschreibung der Leistungen, die erforderlich sind oder gefordert werden, damit die Ziele des Projekts erreicht werden.“ (DIN 69901)

Rückblick – Aktivitäten im RE

04 Requirements Engineering / 04.2 Methoden und Techniken



Quelle: Balzert, 2009, S.446



- Prinzipiell gilt
 - Jedes Artefakt einer Softwareentwicklung ist einer Qualitätsprüfung zu unterziehen bevor es für weitere Entwicklungsaktivitäten verwendet wird
 - Weitere Details:
s. Vorlesungseinheiten „Qualitätsmanagement /-sicherung“
- Notwendige Schritte
 - Analyse der Anforderungsspezifikation
 - Validierung der Anforderungsspezifikation

- Nach welchen Kriterien können Anforderungen bewertet werden?
 - **Priorität**
(am besten: Reihung, sonst: Pflicht, Optional, Zukunft, ...)
 - **Komplexität**
(hoch / mittel / niedrig)
 - **Stabilität**
(hoch / mittel / niedrig)
 - **Risiko**
 - **Aufwand**

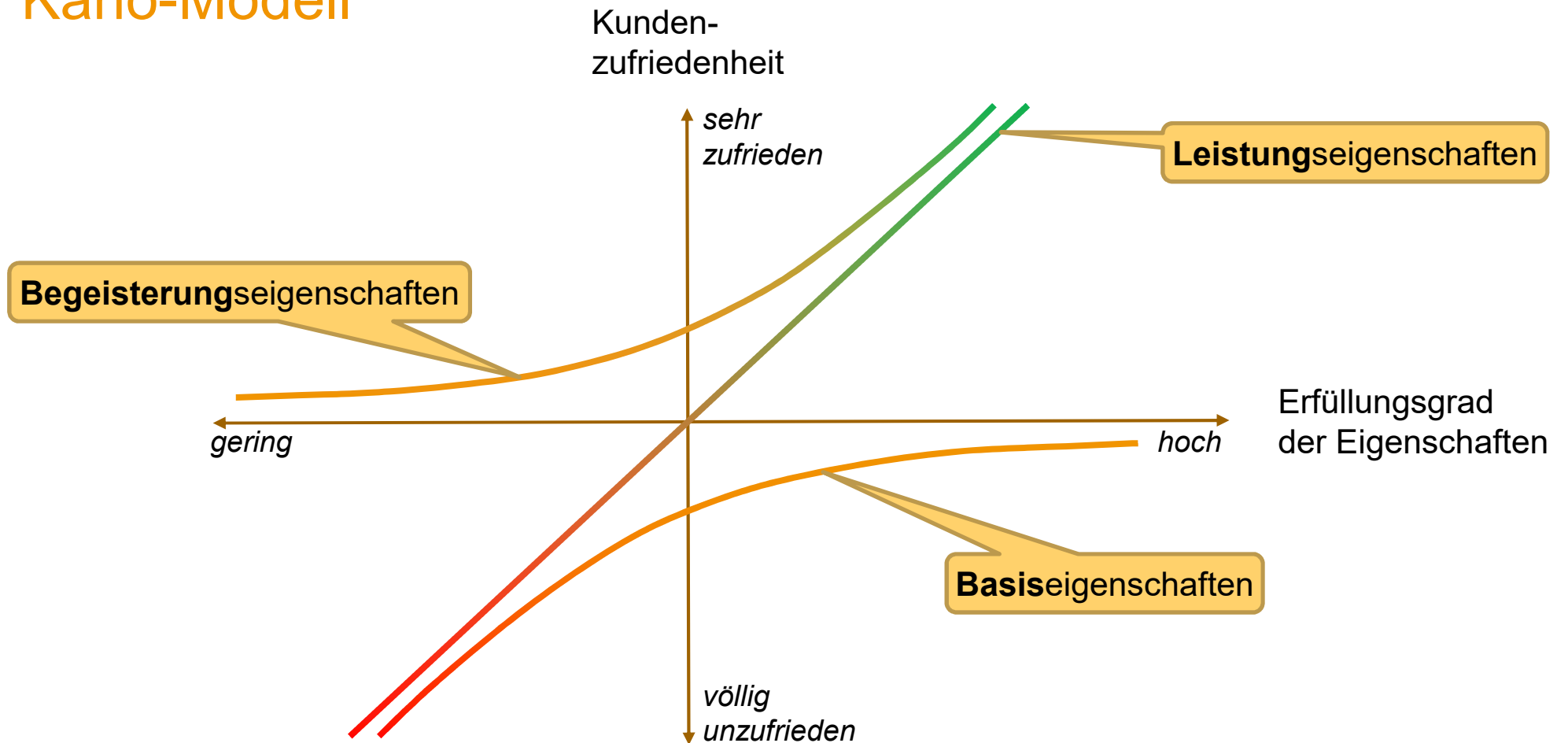
IEEE-Empfehlung

- **Notwendigkeit** (*necessity*)
 - **Essenziell**
 - Software wird nicht akzeptiert, wenn die Anforderung nicht in der geforderten Weise realisiert ist
 - **Bedingt notwendig** (*conditional*)
 - Anforderung wertet die Software auf – wenn sie nicht realisiert wird, wird die Software aber nicht unakzeptabel
 - **Optional** (*optional*)
 - Realisierung der Anforderung kann wertvoll sein, muss aber nicht

Anforderungen priorisieren (2)

04 Requirements Engineering / 04.2 Methoden und Techniken

Kano-Modell



Quelle: Balzert, 2009, S. 544



Kano-Klassifizierung

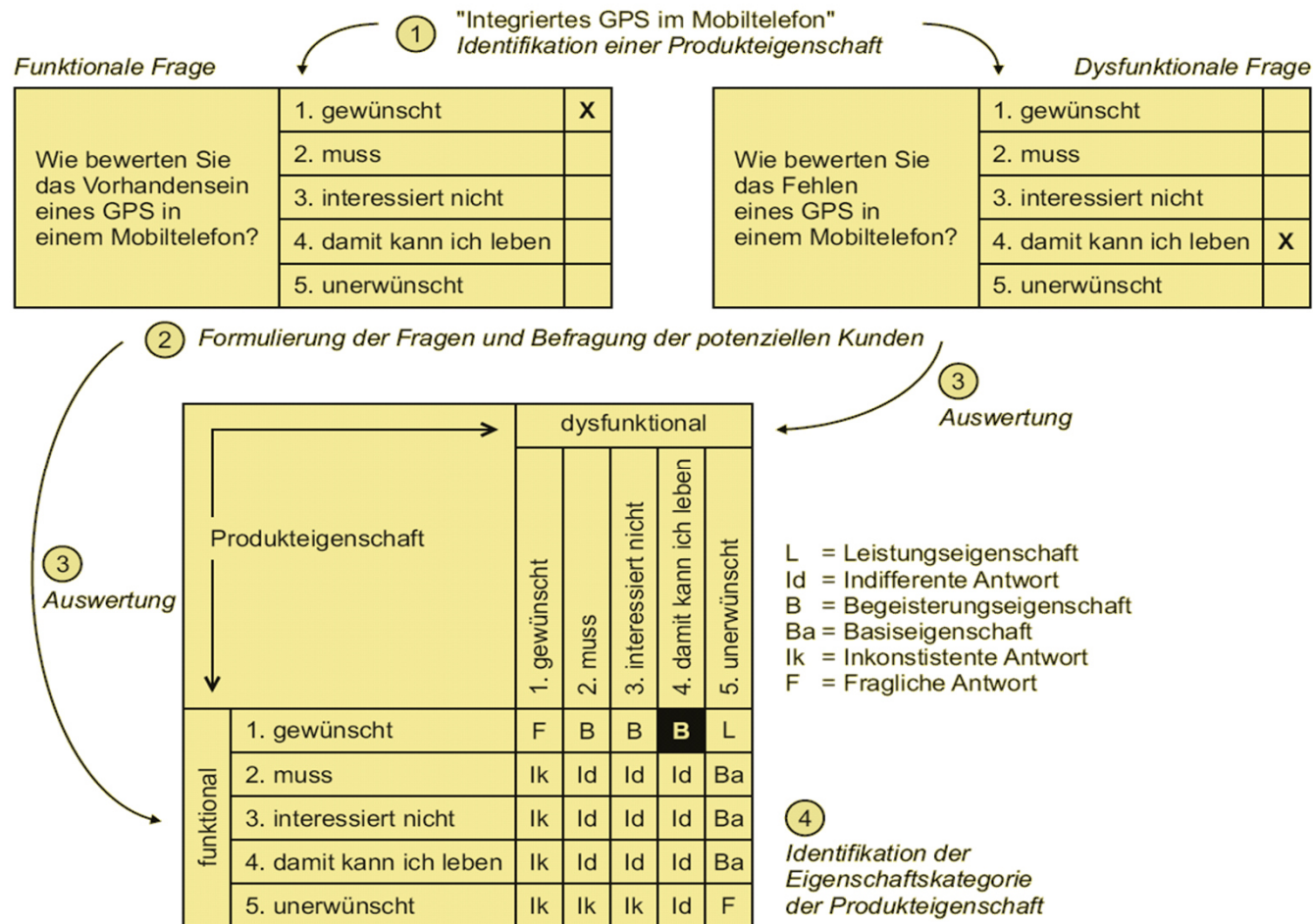
- Einteilung der Produkteigenschaften in Kategorien
 - **Basiseigenschaften**
 - Vom Kunden selbstverständlich vorausgesetzte Eigenschaften (implizite Erwartungen)
 - Fehlen: Unzufriedenheit; Erfüllung: keine Zufriedenheit
 - **Leistungseigenschaften**
 - Vom Kunden bewusst geforderte Eigenschaften (Sonderausstattung)
 - Schaffen Zufriedenheit bzw. beseitigen Unzufriedenheit
 - **Begeisterungseigenschaften**
 - Eigenschaften, die der Kunde nicht erwartet hat
 - Zufriedenheit wächst überproportional



Anforderungen priorisieren (4)

04 Requirements Engineering / 04.2 Methoden und Techniken

● Beispiel: Produkteigenschaftsklassifikation



Quelle: Balzert, 2009, S. 545

Validieren der Anforderungen

04 Requirements Engineering / 04.2 Methoden und Techniken

- **Reviews**

- Strukturierter Prozess zur Qualitätssicherung von Dokumenten, Code, ...
- Menge von Reviewern beurteilt Anforderungen, sucht Fehler, Widersprüche, ...

- **Prototypen**

- Teil der Anforderungen wird prototypisch umgesetzt, etwa als GUI-Prototyp
- Prototyp wird dem Kunden vorgeführt („I know it, when I see it“)

- **Modell Validierung**

- Analyse der (UML-) Modelle

- **Akzeptanztests**

- Aus Anforderungen Testfälle für die Abnahme entwickeln
- Testfälle machen Anforderungen konkreter

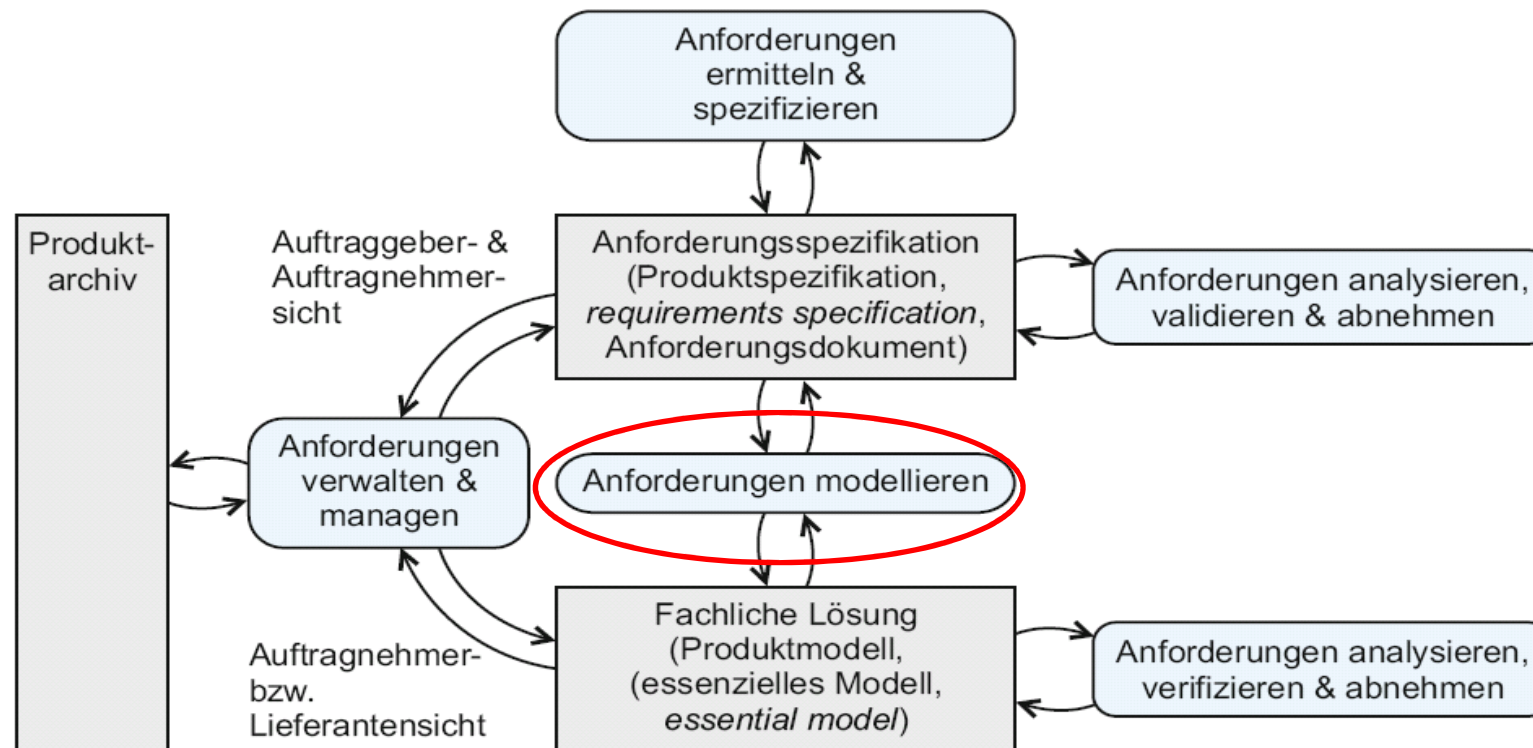
- **Weitere Schritte**

- Erfolgreiche Analyse und Validation sollte durch eine **formelle Abnahme** abgeschlossen werden!



Rückblick – Aktivitäten im RE

04 Requirements Engineering / 04.2 Methoden und Techniken



Quelle: Balzert, 2009, S.446

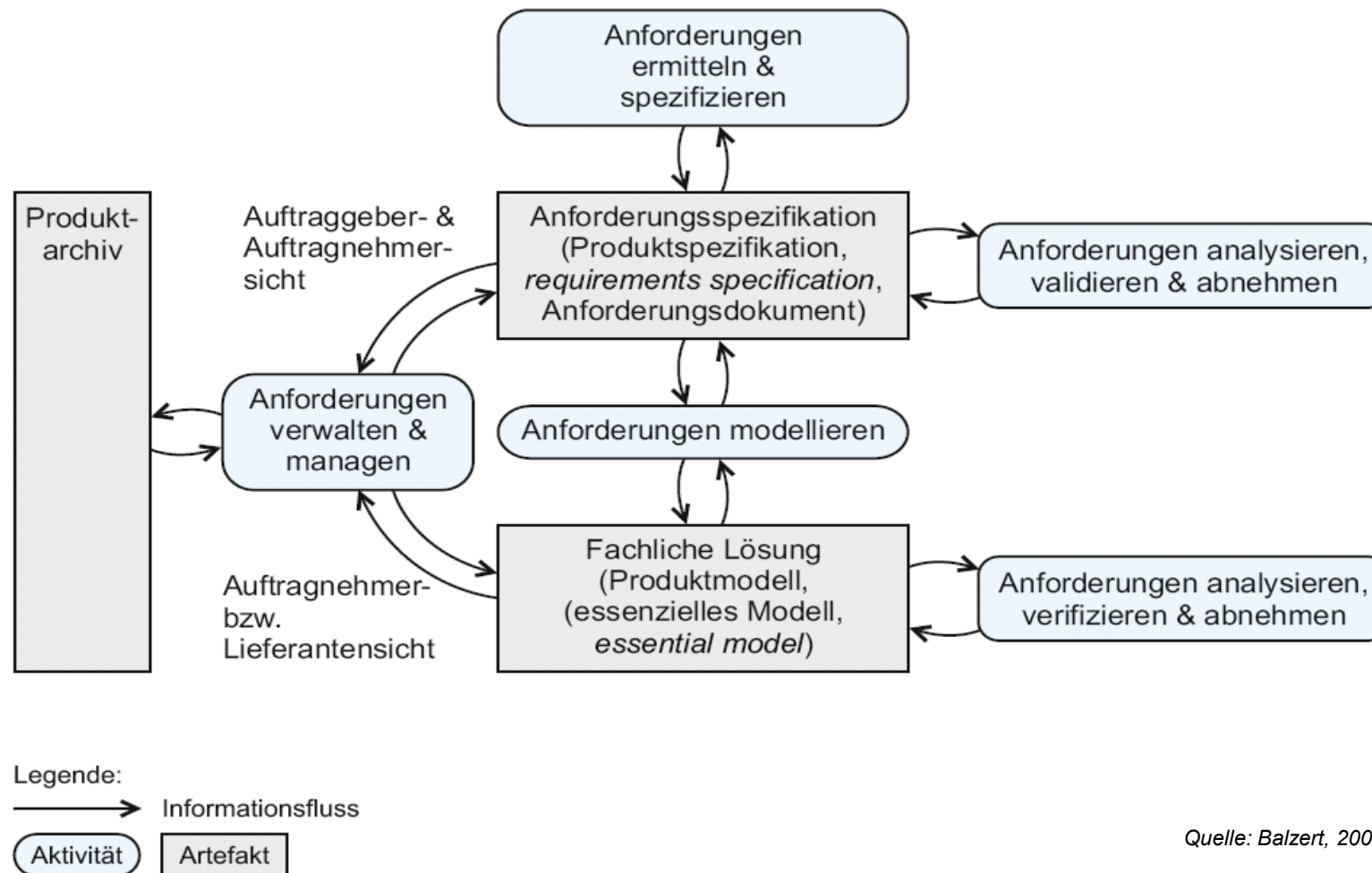


- Ausgangssituation
 - Abgenommene Anforderungsspezifikation
- Erarbeitung einer fachlichen Lösung
 - Synonyme: Produktmodell oder Anwendungsmodell
- Weitverbreitete Methode: OOA (Objektorientierte Analyse)
 - OOA-Methode
 - Vorgehensweise um von den Anforderungen zu einem OOA-Modell zu gelangen
 - OOA-Modell
 - Ergebnis der Analyse (Modellierung der fachlichen Lösung des zu realisierenden Systems)
 - Beschreibungsmittel: UML

- Modellierung der fachlichen Lösung wird i.d.R. durch weitere Artefakte ergänzt:
 - Konzept der Benutzungsoberfläche oder Prototyp der grafischen Oberfläche
 - Ggf. Konzept des Benutzerhandbuchs
- Weitere Details folgen im Kapitel „Modellierung“

Rückblick – Aktivitäten im RE

04 Requirements Engineering / 04.2 Methoden und Techniken



Quelle: Balzert, 2009, S.446