

Inhaltsverzeichnis

- 01 Einführung
- 02 Prozessmodelle
- 03 Konfigurationsmanagement
- 04 Requirements Engineering
- 05 Modellierung
- 06 Qualitätsmanagement



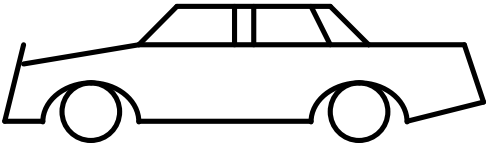
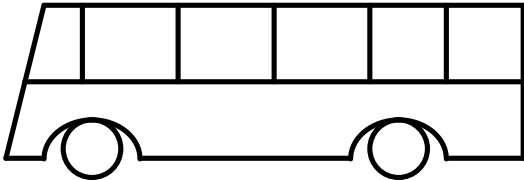
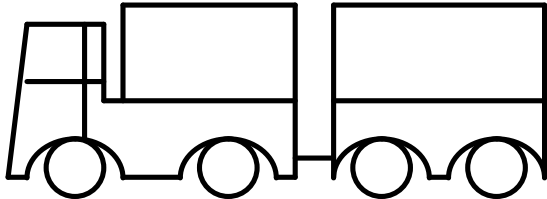
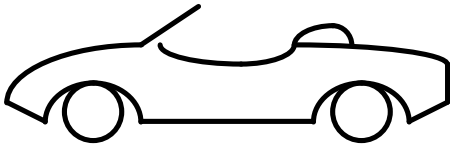
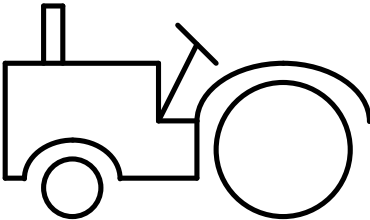
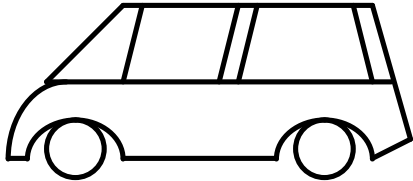
- 01 Einführung
- 02 Prozessmodelle
- 03 Konfigurationsmanagement
- 04 Requirements Engineering
 - 04.1 Aktivitäten, Artefakte und Anforderungsarten
 - 04.2 Methoden und Techniken
- 05 Modellierung
- 06 Qualitätsmanagement

Nach dieser Vorlesungseinheit ...

- ... haben Sie ein Verständnis für den **Sichtwechsel** während einer Softwareentwicklung
- ... können Sie die **Bedeutung des Requirements Engineering (RE)** einschätzen
- ... können Sie die **Aktivitäten** und **Artefakte** des RE beschreiben
- ... können Sie **Anforderungen** und **Anforderungsarten** definieren

Motivation

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

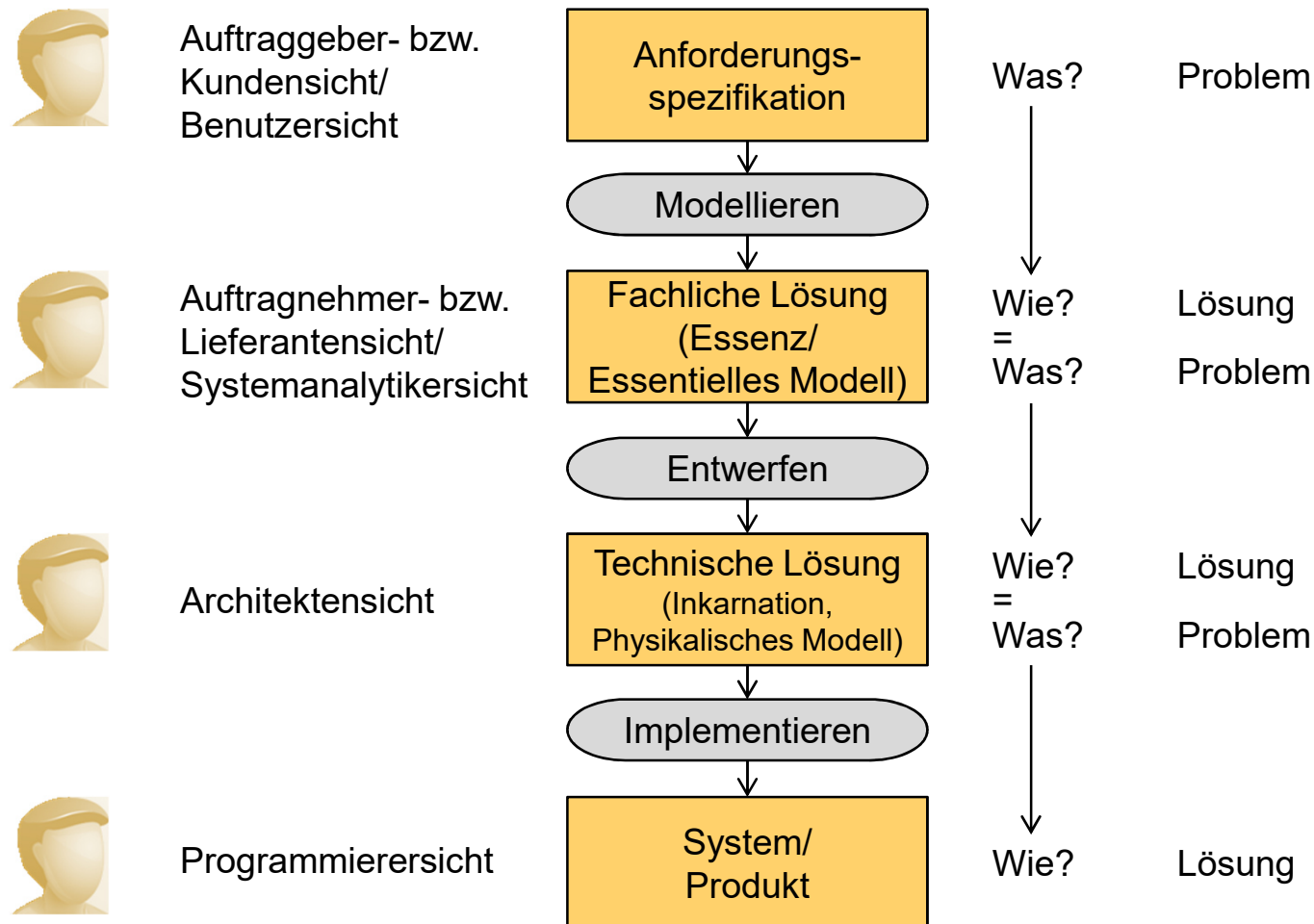
Was der Anwender wollte 	Wie es der Anwender dem Programmierer sagte 	Wie es der Programmierer verstanden hat 
Was der Programmierer bauen wollte 	Was der Programmierer tatsächlich gebaut hat 	Was der Anwender tatsächlich gebraucht hätte 

Quelle: Siedersleben (2002): Softwaretechnik

- Was wird durch Anforderungen festgelegt?
 - Festlegung, **welche Eigenschaften** ein zu entwickelndes Softwaresystem besitzen soll
 - Beschreibung des **Problems**, welches gelöst werden soll

Softwareentwicklung – Sichtwechsel

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten



Wechsel zwischen Problem und Lösung

Legende:

Aktivität

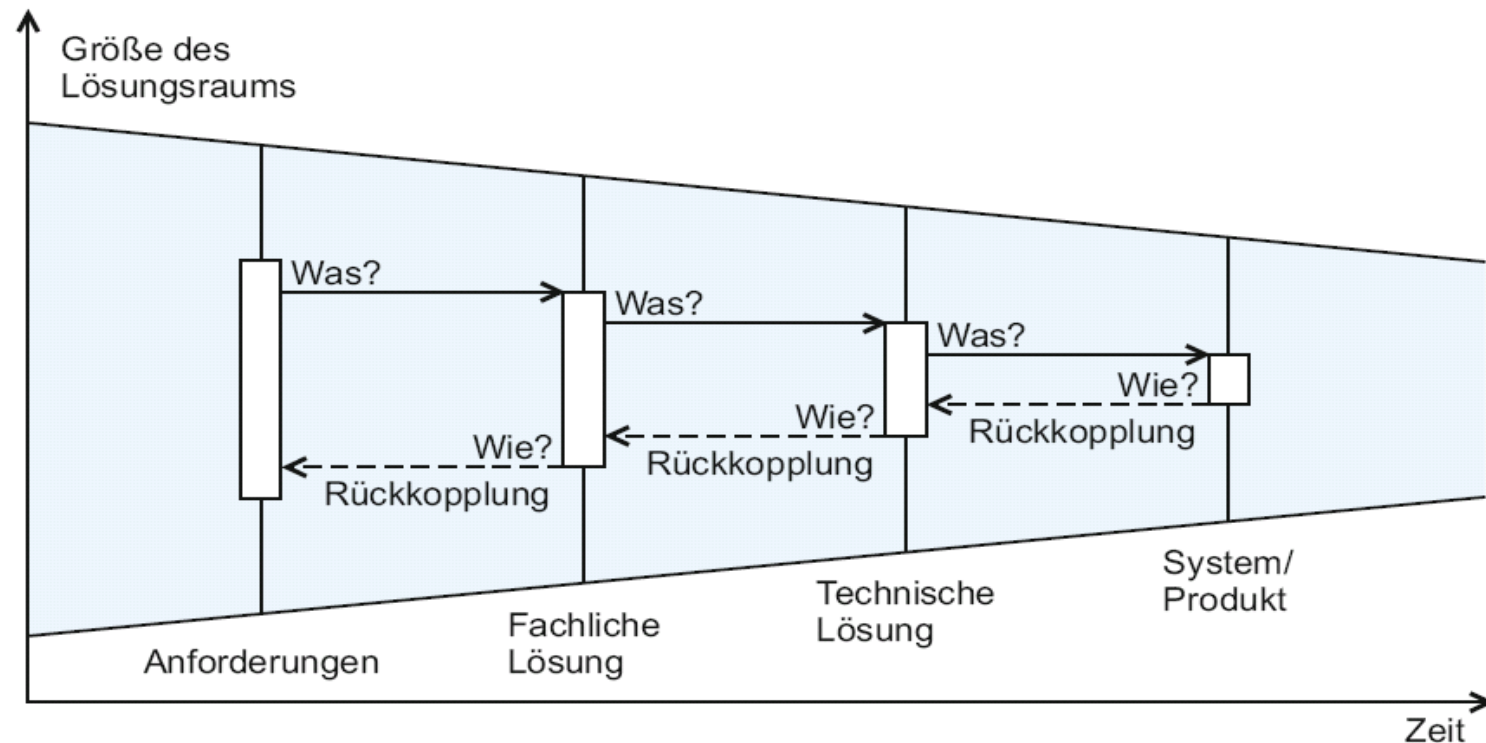
Produkt

Quelle: Balzert, 2009, S.437



Softwareentwicklung – Lösungsraum

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

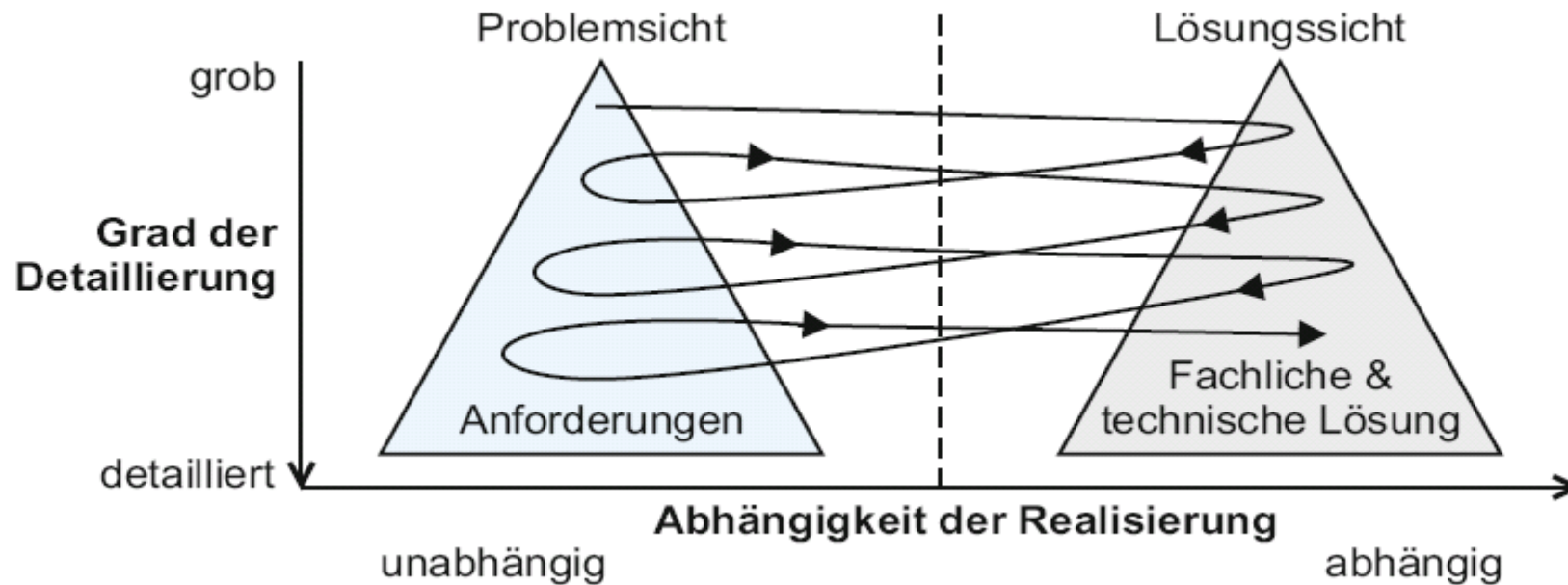


Quelle: Balzert, 2009, S.438

Beachte bei der Aufstellung der Anforderungen:
Möglichst keine Einschränkungen für die nachfolgenden Aktivitäten!

Wechselwirkungen Problemsicht ↔ Lösungssicht

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten



Quelle: Balzert, 2009, S.438

- Möglich: Erkenntnis, dass Problem so nicht gelöst werden kann
➔ Problemmodifizierung oder Abbruch

Bedeutung des RE (1)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

● Warum scheitern Projekte?

- | | |
|---|-------|
| 1. Incomplete Requirements | 13.1% |
| 2. Lack of User Involvement | 12.4% |
| 3. Lack of Resources | 10.6% |
| 4. Unrealistic Expectations | 9.9% |
| 5. Lack of Executive Support | 9.3% |
| 6. Changing Requirements & Specifications | 8.7% |
| 7. Lack of Planning | 8.1% |
| 8. Didn't Need It Any Longer | 7.5% |
| 9. Lack of IT Management | 6.2% |
| 10. Technology Illiteracy | 4.3% |

Quelle: Standish Group Chaos Report 1995



Bedeutung des RE (2)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

„Stolze 43% der im Betrieb festgestellten Fehler in Steuerungs- und Regelungssoftware sind auf Unzulänglichkeiten in der Analysephase oder der Systemspezifikation zurückzuführen. Sie wirken sich viel später aus und kosten eine Menge Geld, wenn man sie beheben will.“

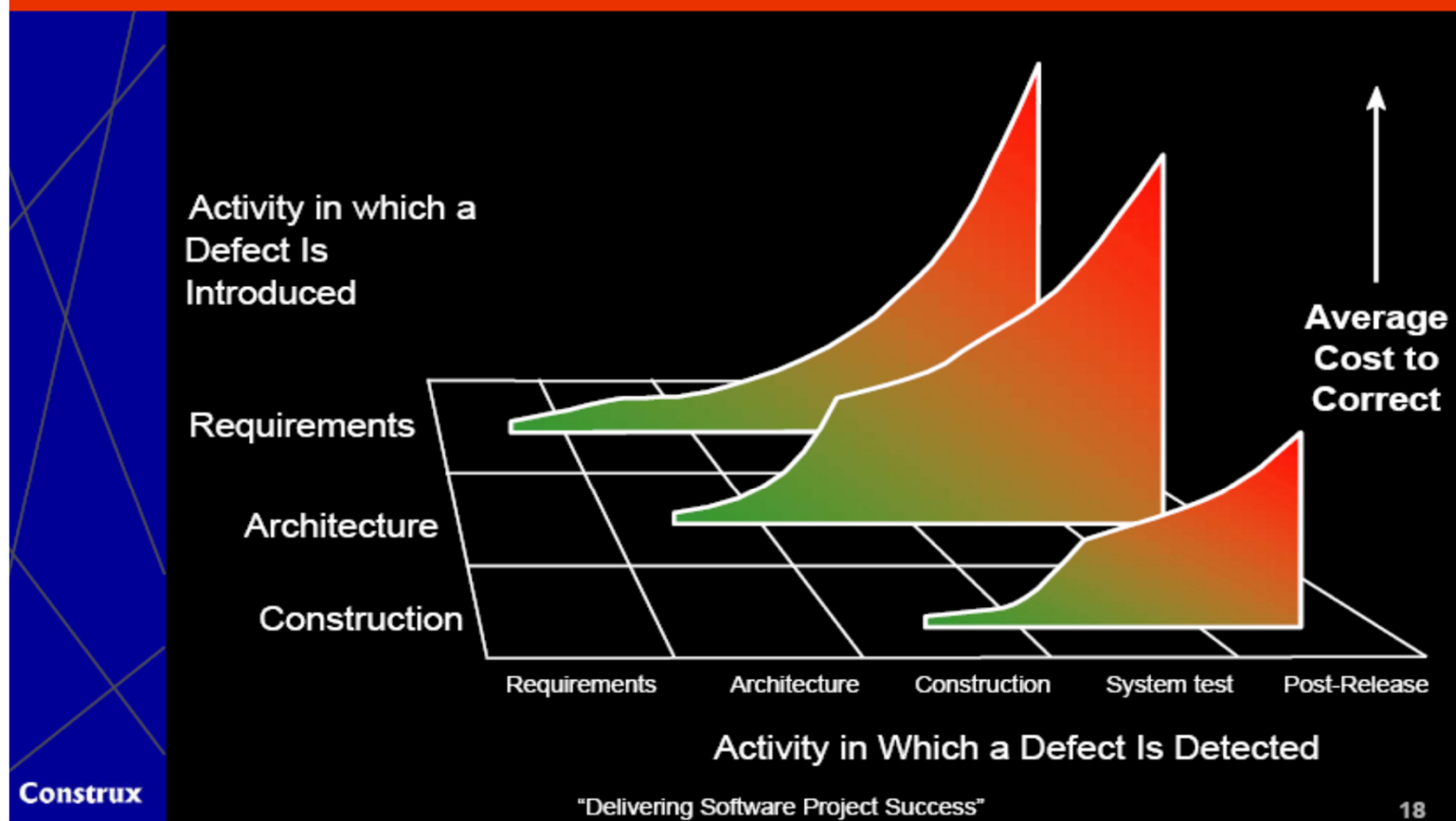
Quelle: Computerzeitung, 23.10.2006



Bedeutung des RE (3)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

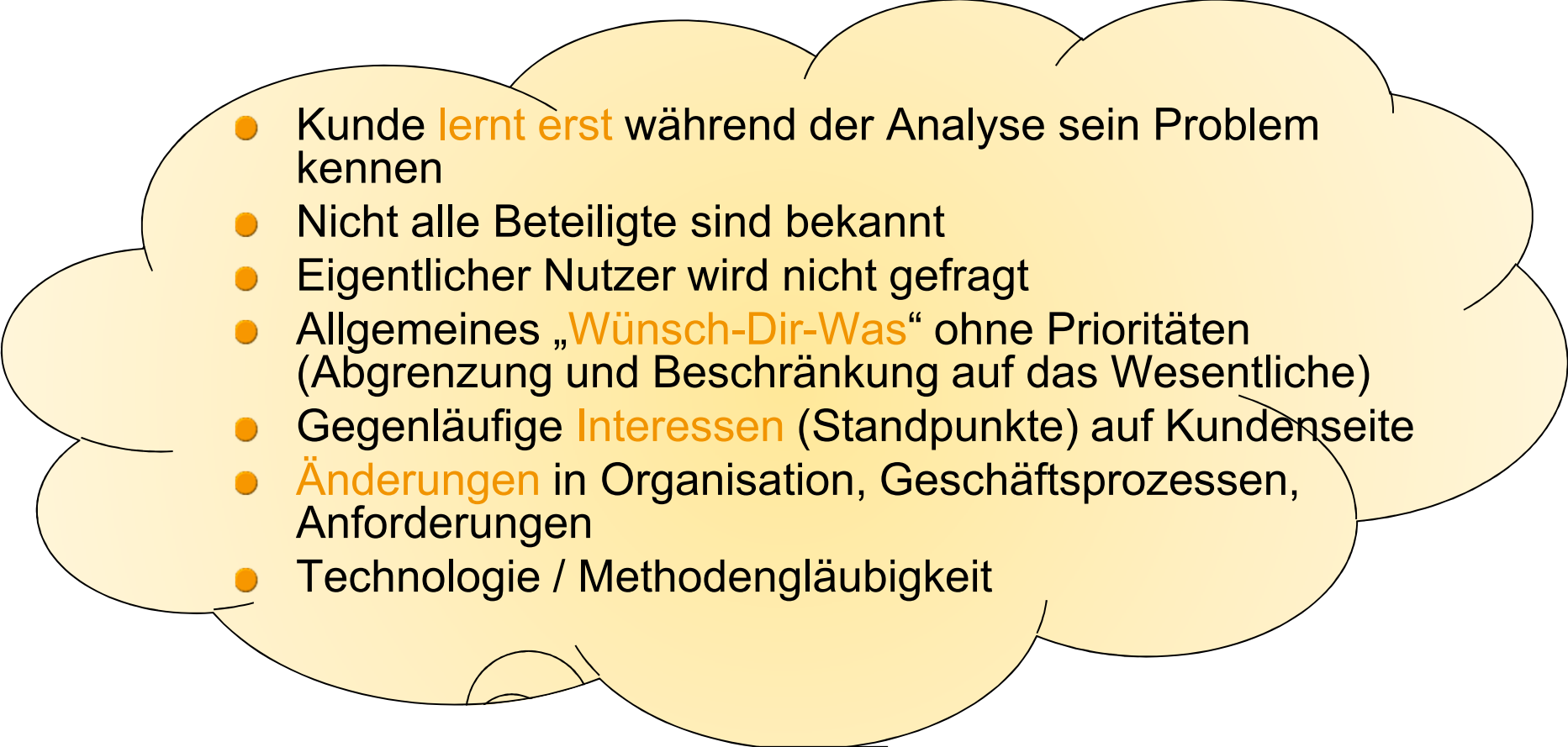
Defect Cost Increase (DCI)



Quelle: Steve McConnell
(Code Complete)

Grundprobleme

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

- 
- Kunde **lernt erst** während der Analyse sein Problem kennen
 - Nicht alle Beteiligte sind bekannt
 - Eigentlicher Nutzer wird nicht gefragt
 - Allgemeines „**Wünsch-Dir-Was**“ ohne Prioritäten (Abgrenzung und Beschränkung auf das Wesentliche)
 - Gegenläufige **Interessen** (Standpunkte) auf Kundenseite
 - **Änderungen** in Organisation, Geschäftsprozessen, Anforderungen
 - Technologie / Methodengläubigkeit



Ziel



Software-System mit klaren Grenzen, Funktionen

Kernproblem: Anforderungen verstehen (1)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

- Beispiel: „Erstellen Sie eine Unterkunft für eine kleine Gruppe Personen zum Schutz vor feindlicher Umgebung“



Lösung 1: Iglu



Lösung 2: Burg



Lösung 3: Raumstation

- Beispiele aus IT-Projekten ...
 - „Berechtigungen sind immer zu prüfen.“
 - „Oberfläche soll leicht konfigurierbar sein.“
 - „Das System soll sicher sein.“

Kernproblem: Anforderungen verstehen (2)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

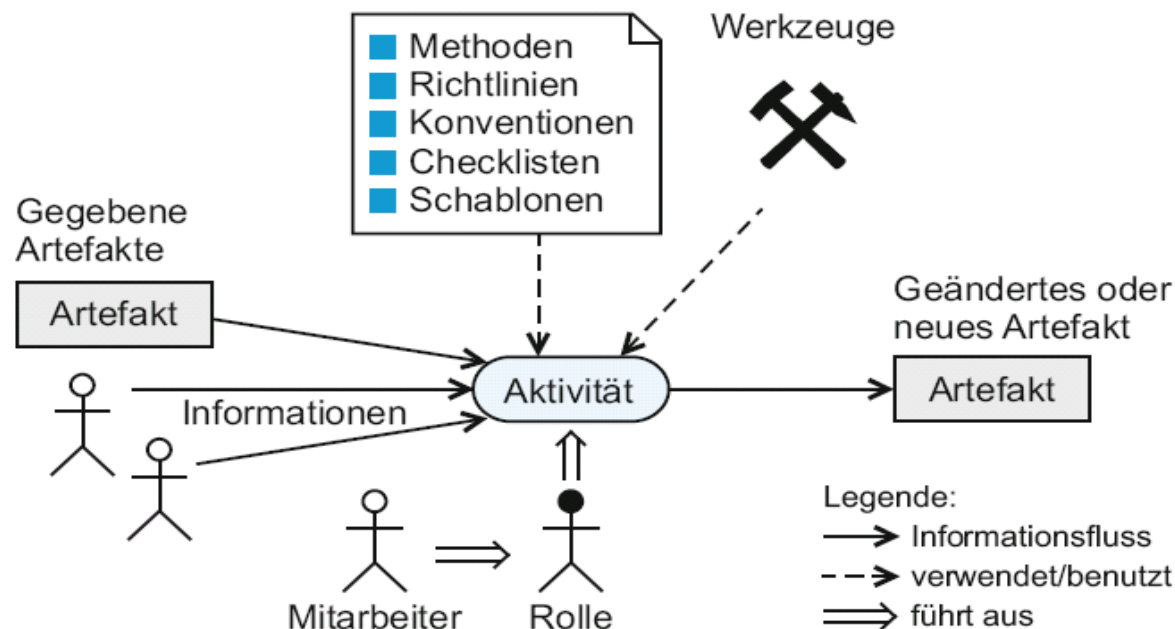
- Beispiel: „Die Software soll intern Währungen umrechnen“
 - **Erste Idee:** Umrechnung wird fest programmiert, feste Kurse
(also: Triviale Anforderung)
 - **Zweite Idee:** Umrechnung ist abhängig von aktuellen Wechselkursen, online Zugriff auf Komponente, welche die aktuellen Kurse kennt
(also: Mittlere Anforderung)
 - **Dritte Idee:** Umrechnungen werden nicht nur online (am aktuellen Tag) benötigt, sondern auch rückwirkend (z.B. zur Bilanzierung, damit muss zu jedem Betrag auch ein Datum gespeichert werden). Mit umgerechneten Kursen soll gerechnet werden, was ist mit Rundungsfehlern? Wird jeder Posten in einer Summe umgerechnet oder nur die Summe? Wie wirkt sich die Umrechnung auf andere Berechnungen aus? Umrechnungsgebühren?



Aktivitäten & Artefakte in der Softwareentwicklung

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

- **Artefakt**
 - = Ergebnis, (Teil-) Produkt *)
- **Softwareentwicklung**
 - = Durchführung (einer Vielzahl) von Aktivitäten

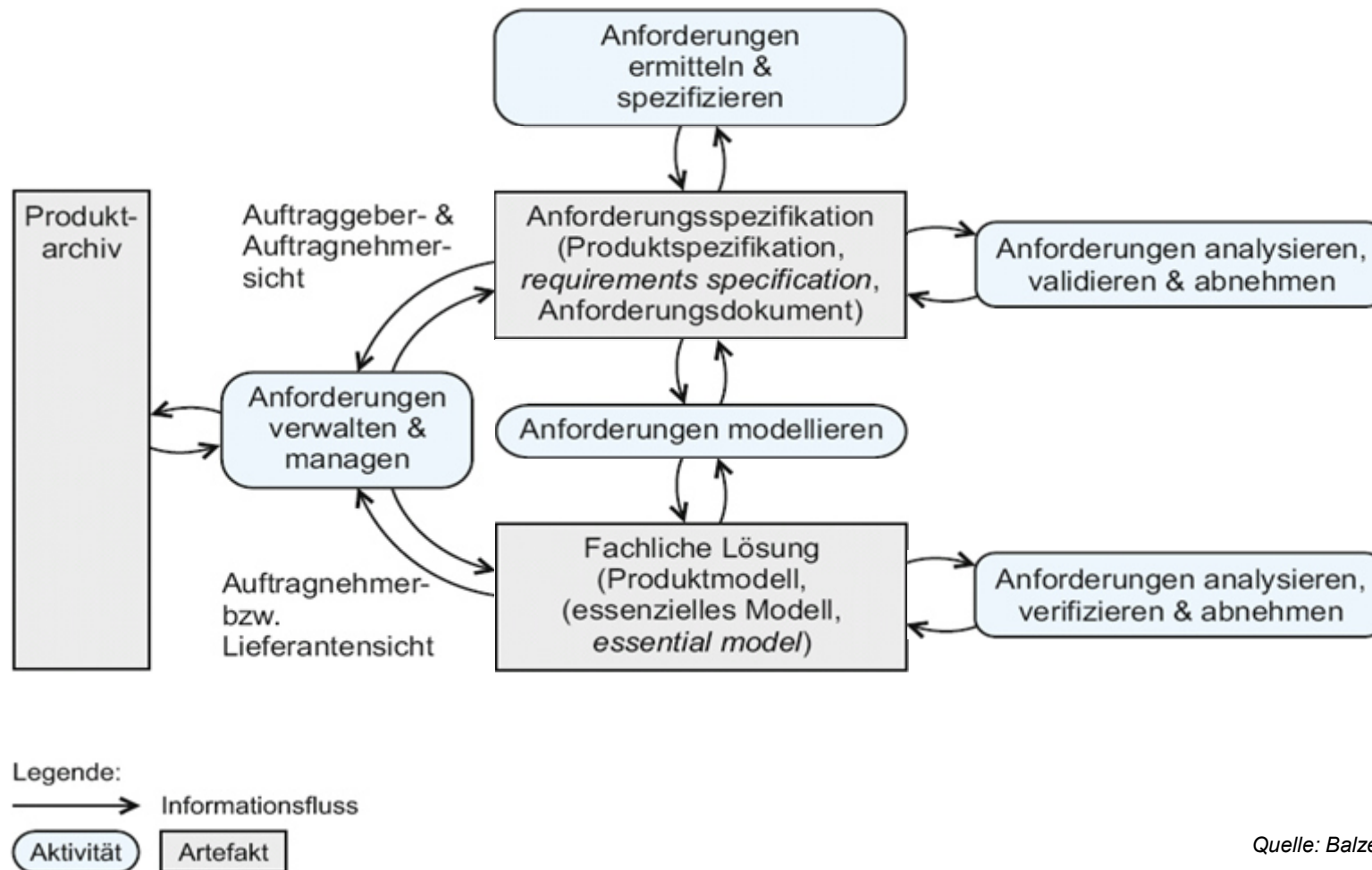


Quelle: Balzert, 2009, S.443

*) Siehe auch V-Modell XT: Vorgehensbaustein

Aktivitäten und Artefakte im RE

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

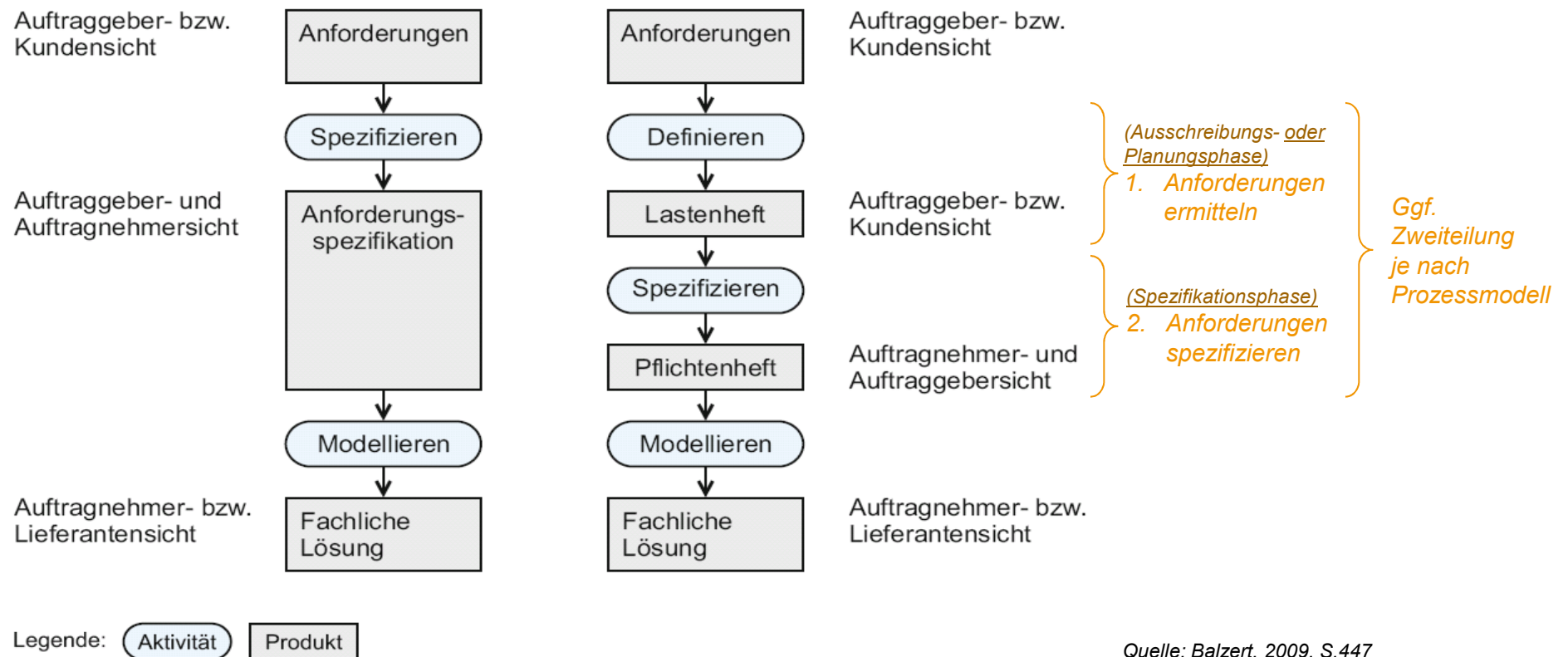


Quelle: Balzert, 2009, S.446

Ein- oder zweistufiges Vorgehen

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

- Umsetzung abhängig von Prozessmodell
 - Anzahl als auch Aufbau von Artefakten unterschiedlich



Quelle: Balzert, 2009, S.447



Begriffsdefinitionen (1)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

● Lastenheft

- Vom Auftraggeber festgelegte Gesamtheit der Forderungen an die Lieferungen und Leistungen eines Auftragnehmers (DIN 69905)
- Erstellt von AG (ggf. in Kooperation mit Beratern)

● Pflichtenheft

- Vom Auftragnehmer erarbeitete Realisierungsvorgaben aufgrund der Umsetzung des vom Auftraggeber vorgegebenen Lastenheftes (DIN 69905)

● Fachliche Lösung

- Besteht in Abhängigkeit von der verwendeten Methode aus verschiedenen Artefakten
- Beispiele: OOA-Modell, GUI-Konzept oder -Prototyp



Begriffsdefinitionen (2)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

- Anforderungen (requirements)
 - Legen fest, was **man** von einem Softwaresystem als **Eigenschaft** erwartet
- Wer ist „man“?
 - Projektstakeholder
 - Alle Personen und Organisationen, die ein Interesse an der Softwareentwicklung haben und von der Softwareentwicklung bzw. dem Einsatz des Softwaresystems betroffen sind
 - Individualsoftware: Auftraggeber (kann komplexes Konstrukt sein)
 - Standardsoftware: Marketingabteilung und Vertrieb



- Welche **Eigenschaften** sollten für ein Softwaresystem festgelegt werden?
 - Visionen und Ziele
 - Rahmenbedingungen/Restriktionen
 - Kontext/Umgebung
 - **Funktionale Anforderungen** (Statik, Dynamik, Logik)
 - **Nichtfunktionale Anforderungen** (Qualitätsanforderungen)
 - Abnahmekriterien

Warum Visionen und Ziele definieren? (1)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

- Funktionale und nichtfunktionale Anforderungen sowie die Rahmenbedingungen können immer gegen diese **abgeglichen** werden
 - „Ist diese Anforderung zielführend, d.h. trägt sie dazu bei, das Ziel zu erreichen?“

- **Vision**

- Beschreibt was erreicht werden soll – Leitgedanke
- Beispiel

V10	<i>Unsere Firma soll durch das System in die Lage versetzt werden, die von uns veranstalteten Seminare sowie Kunden und Dozenten effizient rechnergestützt zu verwalten.</i>
V20	<i>Die Kunden unserer Firma sollen über das Web möglichst viele Vorgänge selbst durchführen können.</i>



Warum Visionen und Ziele definieren? (2)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

● Ziele

- Verfeinerung und Operationalisierung von Visionen

● Beispiel

Z10 *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 unserer Firma von solchen Tätigkeiten entlastet werden.*



- Rahmenbedingung
 - Synonym: Restriktion
 - Engl. **constraint** für das Softwaresystem und/oder Entwicklungsprozess
- Zwei verschiedene Klassen
 - **Organisatorische** Rahmenbedingungen
 - **Technische** Rahmenbedingungen

Rahmenbedingungen (2)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

- **Organisatorische Rahmenbedingungen**
 - Welche **Anwendungsbereiche** sollen abgedeckt werden?
 - Für welche **Zielgruppe** ist das System vorgesehen?
 - Was sind die **Betriebsbedingungen**?
- **Beispiel**

Anwendungsbereich	<i>Kaufmännisch/administrativer Anwendungsbereich.</i>
Zielgruppen	<i>Mitarbeiter unserer Firma gliedern sich in: Kundensachbearbeiter, Seminarsachbearbeiter und Verwaltungsbetreuer. Kunden unserer Firma: Kunden und Firmen können sich per Internetzugang über Seminare und Veranstaltungen informieren und selbst Buchungen durchführen.</i>
Betriebsbedingungen	<i>Büroumgebung.</i>



Rahmenbedingungen (3)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

● Technische Rahmenbedingungen

- **Software:** Welche Softwaresysteme sollen auf der Zielmaschine zur Verfügung stehen?
- **Hardware:** Welche HW-Komponenten sind in minimaler und maximaler Konfiguration für den Systemeinsatz vorgesehen?
- **Orgware:** Welche vorhandene IT-Landschaft muss berücksichtigt werden?
- Anforderungen an die **Entwicklungsumgebung**

● Beispiel

Technische Produktumgebung	<i>Das System ist eine Web-Anwendung.</i>
Software	<i>Server-OS: Linux. Client: Web-Browser (unterstützt werden die drei marktführenden).</i>
Hardware	<i>Server: Standalone Workstation. Client: Browserfähig mit Grafikdisplay.</i>
Orgware	<i>Zugriff des Servers auf die Buchhaltungssoftware.</i>
Anforderungen an Entwicklungsumgebung	<i>Keine Abweichungen von der operativen Umgebung.</i>

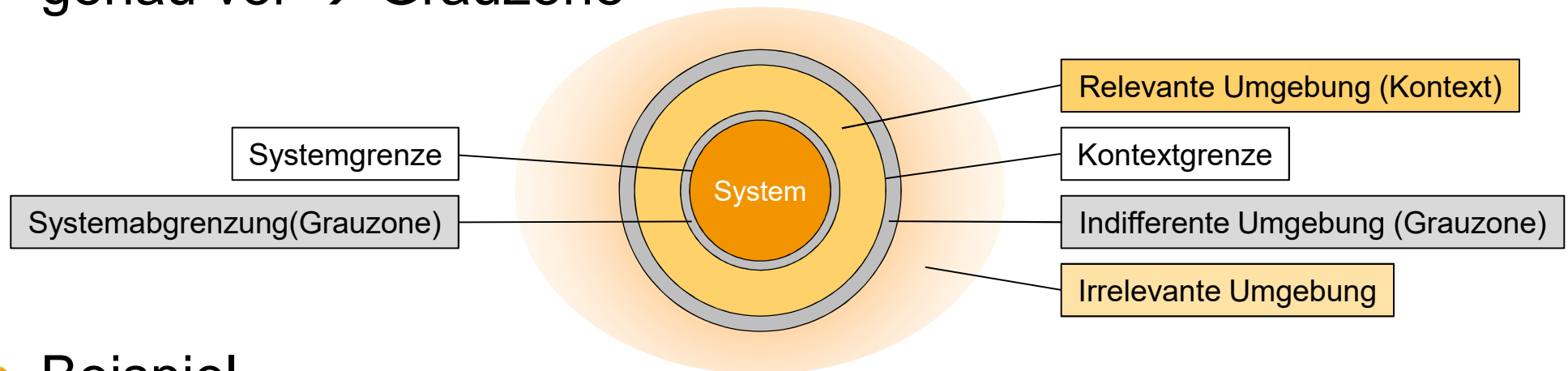


- Um das zu entwickelnde System herum gibt es eine relevante **Umgebung**, die für die Systementwicklung zu beachten ist
- Beispiel
 - **Materielle Umgebung**: Sensoren, Gebäude, Personen, andere technische Systeme, physikalische Kanäle und Übertragungsmedien
 - **Immaterielle Umgebung**: Schnittstellen zu anderen Softwaresystemen, Internet

Kontext und Überblick (2)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

- Am Anfang des RE liegt die Systemgrenze meist noch nicht genau vor → Grauzone



- Beispiel
 - „*Das System SemOrg soll mit der Buchhaltungssoftware Daten austauschen*“
 - Zum Zeitpunkt der Anforderungserstellung steht das Buchhaltungssystem noch nicht fest
→ Schnittstelle kann noch nicht fertig spezifiziert werden

- Festlegung des Kontexts ist wichtig, weil dadurch die Interpretation der Anforderung beeinflusst wird
- Beispiel
 - „*Reaktion auf Benutzereingaben muss in weniger als 2 Sekunden erfolgen*“
 - **Kontext 1:** Anwendung läuft auf einem **Handy** mit Internetanschluss
 - **Kontext 2:** Anwendung läuft auf **PC** mit **DSL**-Anschluss

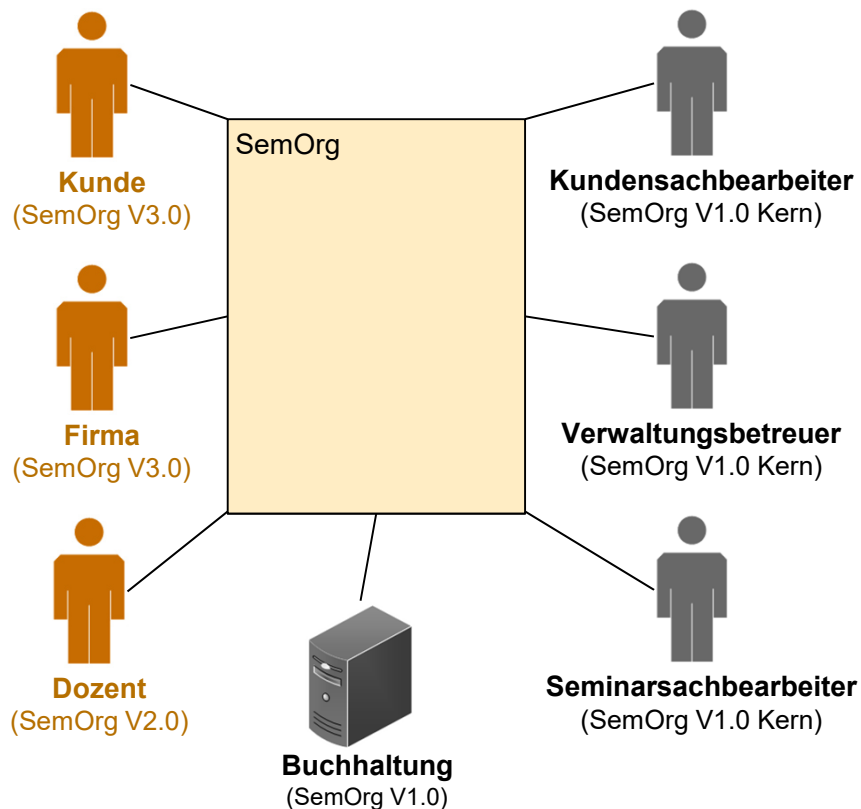
Kontext und Überblick (4)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

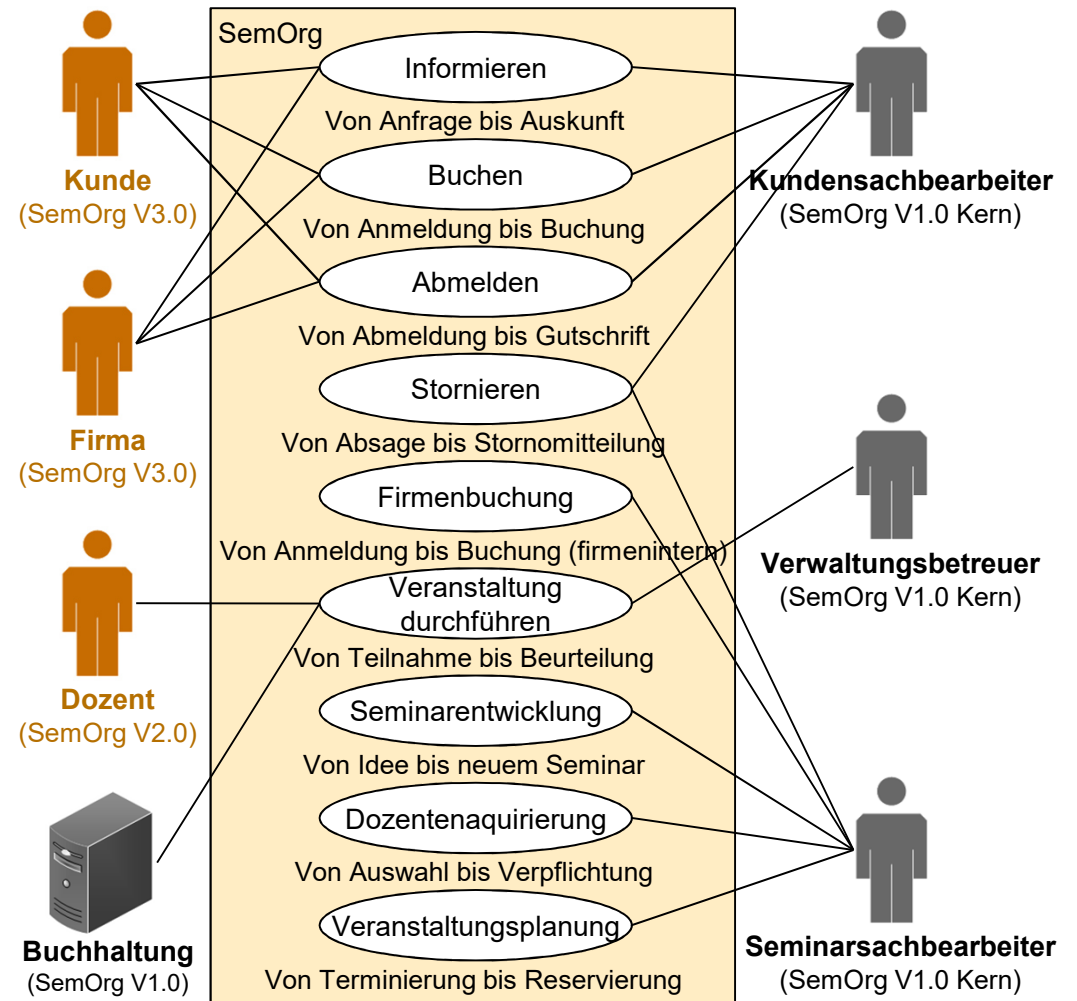
● Notationen für die **Darstellung** des Kontexts

- Natürlich sprachliche Beschreibung
- UML-Diagramme (Anwendungsfall- / Use Case- Diagrammartig)

Umweltdiagramm (Bsp.)



Überblicksdiagramm (Bsp.)



Was sind nichtfunktionale Anforderungen? (1)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

● Unterschiedliche Auffassungen

- Oft alles subsumiert, was nicht zu den funktionalen Anforderungen gehört
- Aspekte, die typischerweise **mehrere oder alle funktionalen Anforderungen betreffen**
 - Genauigkeit, Verfügbarkeit, Konsumierbarkeit, Internationalisierung, Zuverlässigkeit, Sicherheit, Service-Anforderungen, Support,
- „Entweder **Qualitätsanforderungen** oder unterspezifizierte funktionale Anforderungen“ (Pohl)
- Festlegung, welche **Qualitätsmerkmale** relevant sind und in welcher **Qualitätsstufe** sie erreicht werden sollen



Was sind nichtfunktionale Anforderungen? (2)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

- Was sind die Qualitätsmerkmale von Software?
 - ISO/IEC 25000 (vormals ISO/IEC 9126)
 - Funktionalität
 - Zuverlässigkeit
 - Benutzbarkeit
 - Effizienz
 - Wartbarkeit
 - Portabilität
- Beispiel für Zuordnung von Qualitätsstufen
 - Sehr gute Systemqualität
 - Gute Systemqualität
 - Normale Systemqualität



Nichtfunktionale Anforderungen (1)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

- Spezifikation erfolgt meist in **natürlicher Sprache**
- **Allgemeines Problem: schwer zu überprüfen**
 - „*Einfache Bedienung*“
 - „*Fähigkeit, sich von einer Fehlfunktion zu erholen*“
 - „*Schnelle Antwort auf Benutzereingaben*“
- **Ideal**
 - Benutzung von Metriken, die objektiv überprüft werden können (Systemtests)



● Beispiel **Metriken**

● Geschwindigkeit

- Ausgeführte Transaktionen/Sekunde
- Reaktionszeit auf Benutzereingabe oder Ereignis
- Bildschirmauffrischungszeit

● Größe

- Kilobyte oder Anzahl der Speicherbausteine

● Benutzerfreundlichkeit

- Schulungsdauer oder Anzahl der Hilfeseiten

● Stabilität

- Zeit bis zum Neustart nach einer Fehlfunktion
- Anteil der Ereignisse, die zu Fehlfunktionen führen
- Wahrscheinlichkeit der Datenzerstörung nach Fehlfunktion

Abnahmekriterien (1)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

- Angaben, wie die Erfüllung der Anforderung bei der Abnahme überprüft werden soll
 - Z.B. durch Testszenarien
- Welche Vorteile bringt es, wenn mit der Festlegung der Anforderungen bereits die Festlegung von Abnahmekriterien verbunden wird?
 - Transparenz bzgl. der Überprüfung
 - Operationalisierte Festlegung und qualitative Verbesserung der Anforderungen
 - Bessere Veranschaulichung und Verständnis



● Klassifizierung

● Abstraktionsgrad

- Abstrakte Abnahmekriterien (keine konkreten Werte)
- Konkrete Abnahmekriterien (konkrete Werte)

● Reichweite

- Genau eine Anforderung
- Mehrere Anforderungen
- Nur für Teile einer Anforderung

Abnahmekriterien (3) – Beispiel

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

Anforderung

F17 *Wenn ein Kunde oder eine Firma sich von einer bereits gebuchten Veranstaltung später als X Wochen vor der Veranstaltung abmeldet, dann muss das System Stornogebühren in Höhe der Veranstaltungsgebühr berechnen oder nach einem Ersatzteilnehmer fragen.*

Abnahmekriterium (abstrakt)

Ausgangssituation: *Ein Kunde, eine Firma, ein Seminar und eine Veranstaltung sind angelegt. Der Kunde und die Firma haben die Veranstaltung gebucht.*

Ereignis: *Der Kunde und die Firma stornieren die Veranstaltung X Wochen vor Veranstaltungsbeginn (wobei festgelegt sein muss, ab wann Stornogebühren in Höhe der Veranstaltungsgebühr anfallen, X muss in diesen Zeitraum fallen).*

Erwartetes Ergebnis: *Das System fragt, ob der Kunde und die Firma einen Ersatzteilnehmer stellen (Name usw. wird angefordert). Wenn nein, dann wird mitgeteilt, dass der volle Veranstaltungsbetrag fällig wird. Die Stornierung wird bestätigt.*



- Was sind Eigenschaften von „guten“ Anforderungen?
 - Korrekt
 - Eindeutig
 - Vollständig
 - Konsistent
 - Klassifizierbar nach Wichtigkeit
 - Klassifizierbar nach Stabilität
 - Überprüfbar
 - Verfolgbar

Quelle: IEEE 830-1998



● Gute Anforderungen

- „Präsentieren Sie dem Kunden die zehn niedrigsten Flugpreise für eine bestimmte Route“
- „Zeigen Sie mögliche Flüge (mit Verbindungen) zwischen zwei beliebigen Orten an“
- „Zeigen Sie alle Reiserouten, die der Kunde in dem System buchen kann“

● Schlechte Anforderungen

- „Das System soll schnell / effektiv / Ressourcen-schonend sein“
(Wie wird das gemessen?)
- „Das System soll mit EJB gebaut werden“
(Realisierungsdetail)
- „Das System soll robust sein“
(Was bedeutet „robust“?)

- **Prinzipielle Unterscheidung**
 - Spezifikationen **ohne** festgelegte **Regeln**
 - Spezifikationen mit nummerierten oder **markierten Anforderungen**
 - Spezifikationen mit festgelegtem **Gliederungsschema**
 - Verwendung von Schablonen bzw. Muster
 - IEEE Recommended Practice for Software Requirements Specifications (ANSI/IEEE Std 830-1998)
 - Vorlagen im V-Modell XT
 - Schablonen für Lastenheft, Pflichtenheft und Glossar (Balzert 2009)
 - Schablonen für agile Entwicklungen

- Ziel der agilen Softwareentwicklung
 - Außer dem Code möglichst wenig Dokumentation
- Ansätze
 - Integration des Kunden bzw. Kundenrepräsentanten in Entwicklungsteam
 - Spezifische Vorgehensweise (*siehe Prozessmodelle*)
 - Anforderungen werden in Form von sog. **User Stories** (Benutzergeschichten) vom Kunden erhoben

User Story (1)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

- Kunden schreiben Anforderungen selber auf – Entwickler unterstützen durch Fragen
 - Dinge beschreiben, die das System für den Kunden tun soll
 - Umgangssprachliche Sätze in der Geschäftsterminologie des Kunden
- I.d.R. Festhalten auf Karteikarten
 - Eindeutige **Identifizierung** (z.B. Nummer)
 - **Priorität** (vergeben durch Kunden)
 - **Aufwand** zur Implementierung
 - Von Entwicklern geschätzt
 - Gute Erfahrung mit abstrakter Einheit „**Story Points**“
 - In Anlehnung an Fibonacci-Folge: 1, 2, 3, 5, 8, 13, ...
 - Vergleich der Komplexität von User Stories fällt leichter
 - (Aber auch z.B. Stunden im 2er-Team)
 - 1 bis 3 Wochen ideale Entwicklungszeit



User Story (2)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

1P101

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

Priorität: # 4

Aufwand: 20 Stunden

Story 26
05-Oct-21

Edit SP Details Screen

Add "Cancel" - Button to undo changes and return to previous page.

TC: Check that browser returns to correct previous change

Estimated: 2h
Actual:

- Textschablone für User Story

As a [role] I want [something] so that [benefit].

- Aufteilung der Karte in drei Bereiche

- Card Section

- Name, Beschreibung, Referenznummer, geschätzter Aufwand, ...

- Conversation Section

- Weitere Informationen zur Anforderung

- Confirmation Section

- Abnahmetests für die Anforderung (Rückseite)

Quelle: <https://www.101ways.com/writing-good-user-stories/>

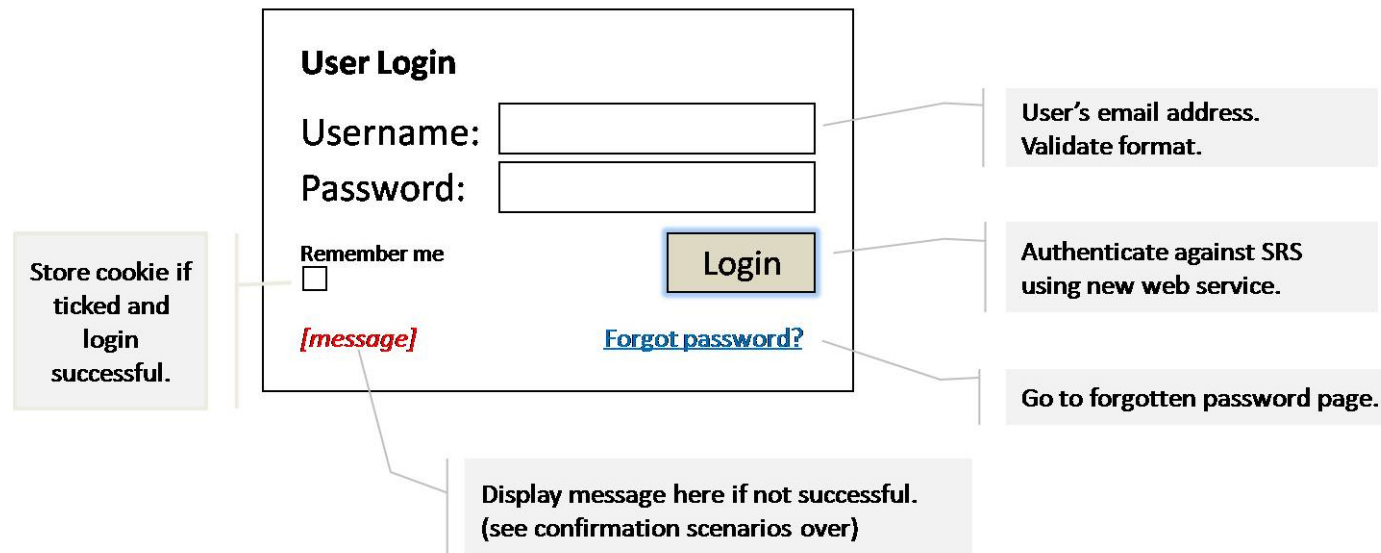


User Story (4)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

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

For new features, annotated wireframe. For bugs, steps to reproduce with screenshot. For non-functional stories, explain scope/standards.



Further information is attached to this story on VSTS Product Backlog.

Quelle: <https://www.101ways.com/2008/01/18/user-story-example/>

User Story (5)

04 Requirements Engineering / 04.1 Aktivitäten, Artefakte und Anforderungsarten

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.

Quelle: <https://www.101ways.com/2008/01/18/user-story-example/>



- Typische Vorgehensweise
 - Am Anfang...
 - ...um Umfang des Systems zu identifizieren
 - Während der Entwicklung...
 - ...um neue Anforderungen aufzustellen
 - Unterteilung vorhandener User Stories
 - Neue Priorisierung
 - Entfernung von User Stories
 - Vor Implementierung...
 - Erstellung einer Bildschirmsskizze oder UML-Aktivitätsdiagramm
 - 60 bis 100 User Stories bilden ca. ein Release