



Vorlesung Softwaretechnik I (SS 24)

5. Analyse

Prof. Dr. Jens Grabowski

Tel. 39 172022

grabowski@informatik.uni-goettingen.de

Dieses Kapitel basiert auf:

- Hauptquelle:

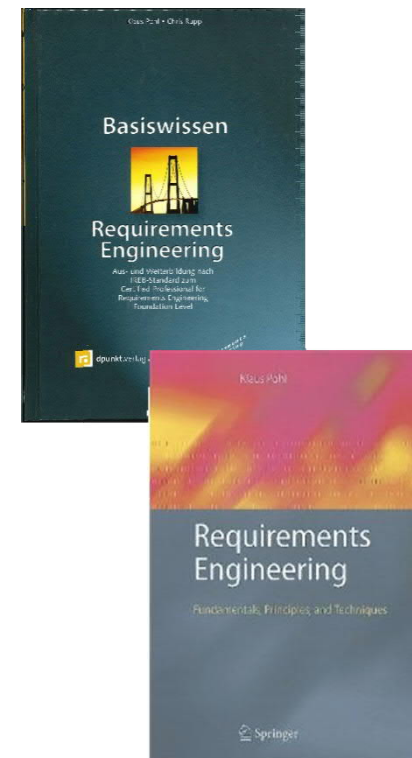
- ☐ Autoren: Klaus Pohl, Chris Rupp
- ☐ Titel: Basiswissen Requirements Engineering
- ☐ Verlag: dpunkt.verlag

- Version auf Englisch:

- ☐ Autor: Klaus Pohl
- ☐ Titel: Requirements Engineering: Fundamentals, Principles, and Techniques
- ☐ Verlag: Springer

- Vorlesung unregelmäßig:

- ☐ Dozent: Dr. Philip Makedonski
- ☐ Titel: Requirements Engineering





Inhalt

- Einführung
- Anforderungen (Requirements)
- Anforderungsdefinition (Requirements Engineering)
- Requirements Engineering Prozess
- Produktdatenmodellierung
- Analyseprototyp
- Lernziele

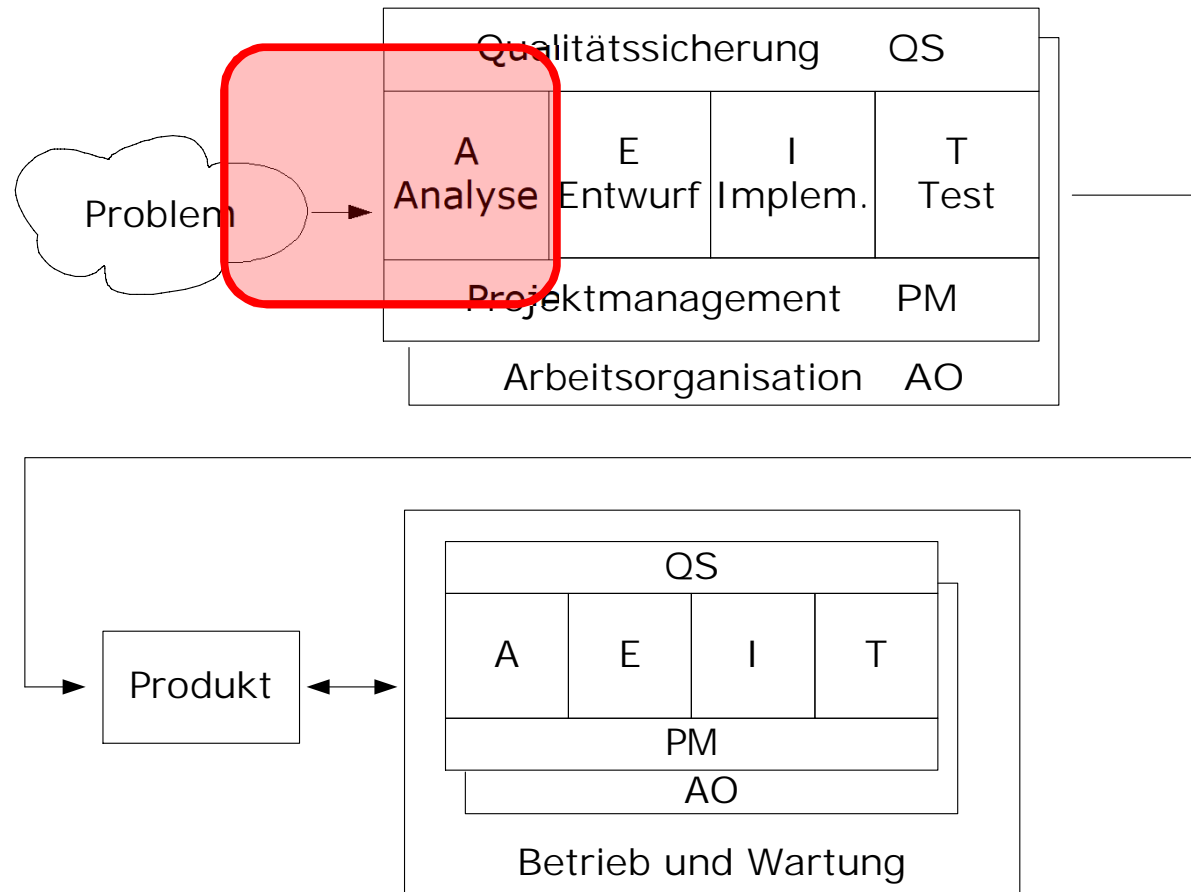


Einführung – Kaufen, Ändern, Bauen?

	Kauf	Änderung	Neuerstellung
Entspricht Anforderungen	Ungefähr (oft ungenutzte oder fehlende Funktionen)	Fast (oft durch technische Grenzen beschränkt)	Genau
Änderbarkeit	Schwierig (techn. Details oft nicht oder schlecht dokumentiert)	Ausgangsprodukt a) selbst hergestellt: <u>leicht änderbar</u> b) von anderen hergestellt: <u>schwer änderbar</u>	Gut (Know-how in der Firma vorhanden)
Preis	Abhängig von Anforderungen und Verbreitung.	Durchführung - selbst: <u>billig</u> - durch andere: <u>teuer</u>	Teuer (Wartung?)

- Für die Entscheidung müssen die Anforderungen bekannt sein!

Einführung – Worum geht es in diesem Kapitel?





Einführung – Analyse besteht aus ...

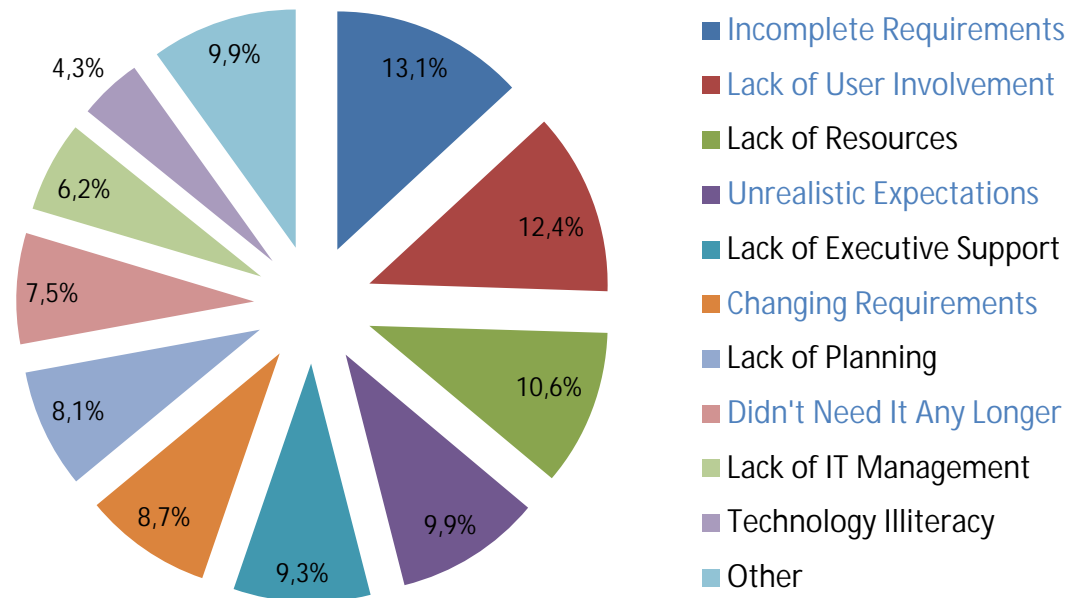
- Requirements Engineering
 - Identifikation der Anforderungen an das Verhalten des Systems.
- Produktdatenmodellierung
 - Erstellung eines **konzeptionellen Datenmodells** (**Domänenmodell**) des Anwendungsbereichs (Domäne).
- Analyseprototyp



Inhalt

- Einführung
- Anforderungen (Requirements)
- Anforderungsdefinition (Requirements Engineering)
- Requirements Engineering Prozess
- Produktdatenmodellierung
- Analyseprototyp
- Lernziele

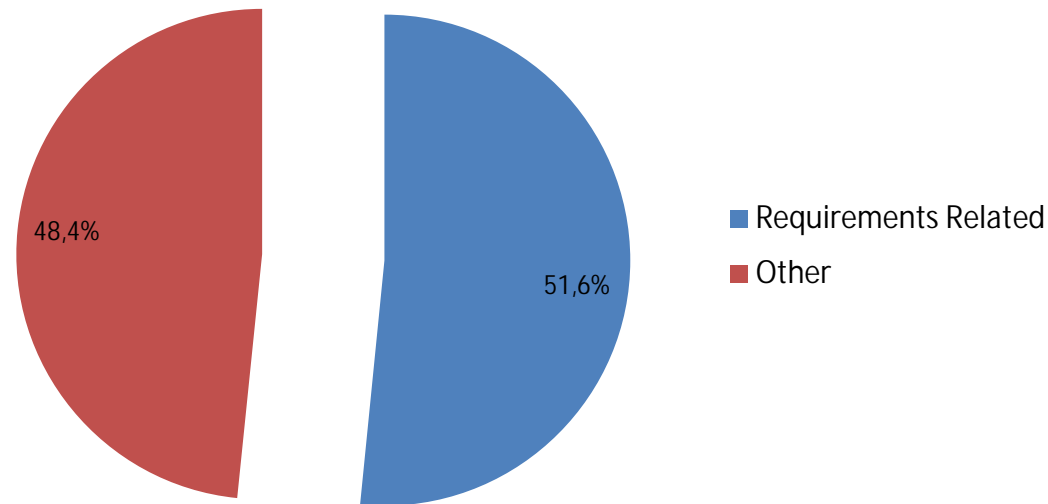
Anforderungen sind zentral für den Projekterfolg!



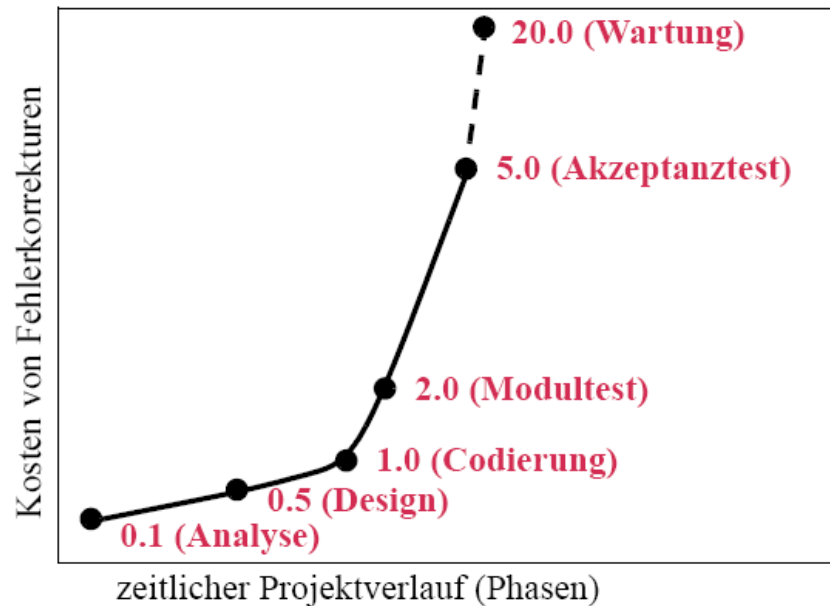
Source: Standish Group 1995 (www.standishgroup.com)

Anforderungen sind zentral für den Projekterfolg!

- Fast 50% der gescheiterten Projekte scheitern aufgrund von schlechten/fehlenden Anforderungen!



Anzahl und Auswirkungen von Fehlern in den Anforderungen



In Phase X beseitigte Fehler können aus Phase X stammen oder Folgefehler von Fehlern in einer früheren Phase sein.

Fazit:

Fast 1/4 aller Anforderungsfehler bleibt bis zur Auslieferung unentdeckt. Die Beseitigung dieser Fehler bei der Wartung (statt bei der Analyse) ist 200 Mal so teuer ...

Fehlerursprung	relative Fehlerzahl	Eliminationsrate	ausgelieferte Fehler
Analyse	1.00	77%	0.23
Design	1.25	85%	0.19
Codierung	1.75	95%	0.09
Sonstige	1.00	86%	0.24
Insgesamt	5.00	85%	0.75

Analyse

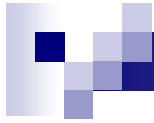


Definition: Anforderung

Eine Anforderung ist:

- (1) Eine Bedingung oder Fähigkeit, die von einem Benutzer (Person oder System) zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.
- (2) Eine Bedingung oder Fähigkeit, die ein System oder Teilsystem erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.
- (3) Eine dokumentierte Repräsentation einer Bedingung oder Eigenschaft gemäß (1) oder (2).

Übersetzt aus [IEEE Std. 610.12-1990]



Arten von Anforderungen

- Drei Klassen von Anforderungen sind zu unterscheiden:
 - Funktionale Anforderungen
 - Nicht-funktionale Anforderungen
 - Constraints (Einschränkungen)

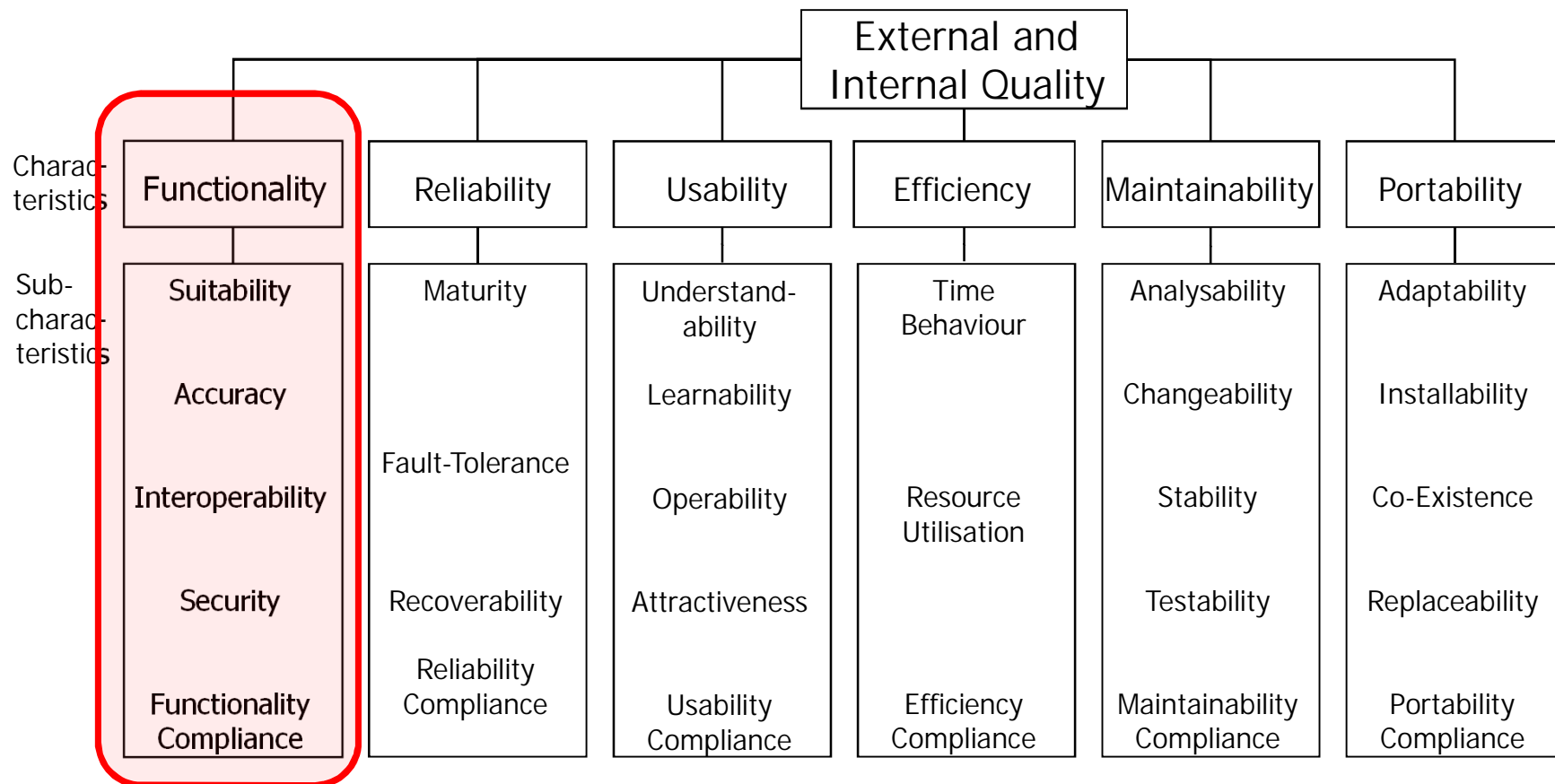


Arten von Anforderungen: Funktionale Anforderungen

- Funktionale Anforderungen spezifizieren das Verhalten, welches das System oder Systemteile erbringen müssen. Sie beschreiben »**was**« das (Teil)System leisten soll. Ihre Umsetzung ist Voraussetzung dafür, dass das System überhaupt einsetzbar ist.
- Merkmale der Funktionalität nach dem ISO Standard 9126 sind: **Angemessenheit, Richtigkeit, Interoperabilität, Ordnungsmäßigkeit, Sicherheit.**

Einschub zur Information:

ISO 9126 Qualitätsmodell



Compliance: If any standards apply to that characteristics: degree of compliance to them.



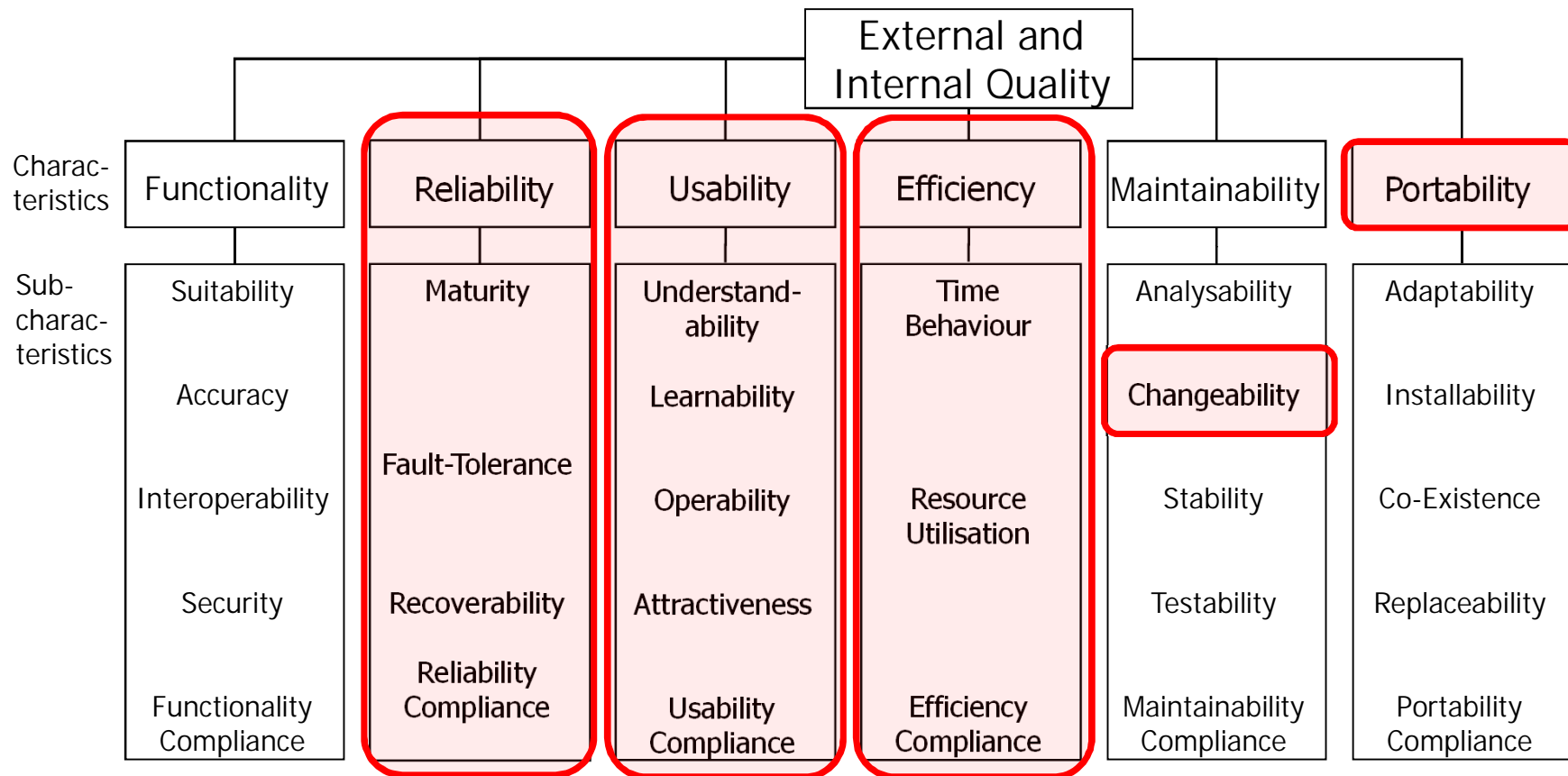
Arten von Anforderungen:

Nicht-funktionale Anforderungen

- Nicht funktionale Anforderungen beschreiben Attribute des funktionalen Verhaltens, also »wie gut« bzw. mit welcher Qualität das (Teil)System seine Funktion erbringen soll. Ihre Umsetzung beeinflusst stark, wie zufrieden der Kunde bzw. Anwender mit dem Produkt ist und wie gerne er es einsetzt.
- Merkmale nach ISO Standard 9126 sind z.B.:
 - Zuverlässigkeit,
 - Benutzbarkeit,
 - Effizienz.
- Indirekt haben auch die
 - Änderbarkeit und
 - Übertragbarkeit Einfluss auf die Kundenzufriedenheit.

Einschub zur Information:

ISO 9126 Qualitätsmodell



Compliance: If any standards apply to that characteristics: degree of compliance to them.



Arten von Anforderungen: Constraints (Einschränkungen)

- Constraints sind **Anforderungen die den Lösungsraum einschränken**.
 - Die Einschränkungen sind dabei nicht notwendig um die funktionalen und nicht-funktionalen Anforderungen zu erfüllen.
 - Constraints
 - können in der Regel nicht vom Entwicklerteam beeinflusst werden.
 - werden nicht implementiert, sondern nur eingehalten, d.h. Constraints sind nicht Teil einer Lösung, sondern limitieren die Lösung.
- Beispiele:
 - Das Produkt muss auf Hardware implementiert werden, die doppelt genauer Gleitkomma Arithmetik (engl. double-precision floating point operations) Arithmetik unterstützt.
 - Das Produkt muss spätestens im Juli 2020 verfügbar sein.



Aufgaben von Anforderungen

■ Anforderungen

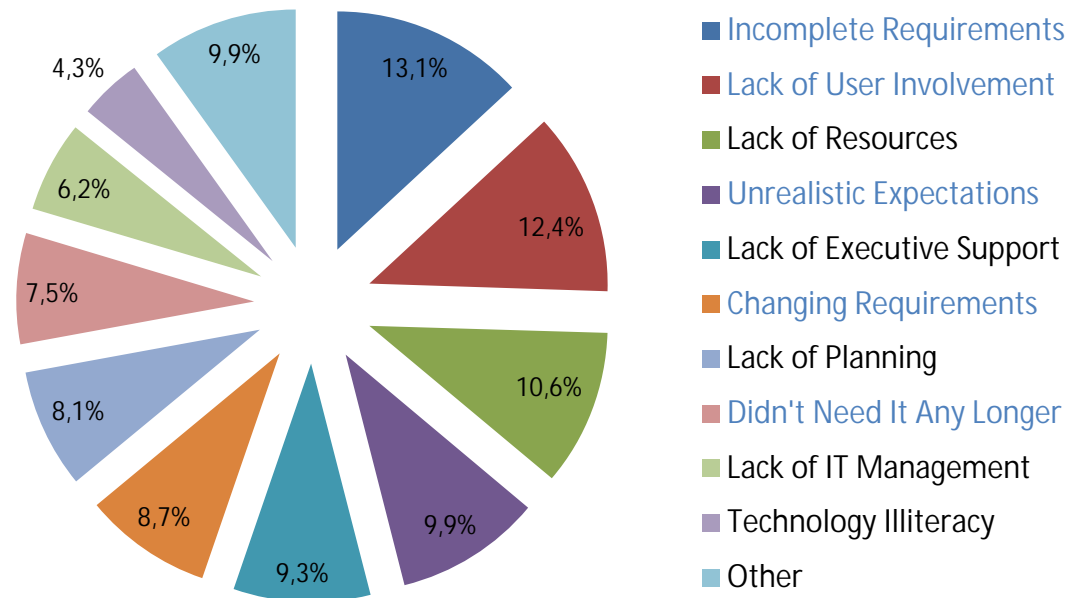
- ☐ beschreiben die von Stakeholdern **gewünschten Ergebnisse**.
 - Ein Stakeholder eines Systems ist eine Person oder eine Organisation, die (direkt oder indirekt) Einfluss auf die Anforderungen des betrachteten Systems hat.
- ☐ repräsentieren verschiedene **Blickwinkel** auf ein System.
- ☐ beschreiben **präzise Akzeptanzkriterien** für Produkte.
- ☐ dienen der **Kommunikation** zwischen Stakeholdern und Entwicklern.
- ☐ sorgen für ein **gemeinsames Verständnis** für das gewünschte Produkt.



Inhalt

- Einführung
- Anforderungen (Requirements)
- **Anforderungsdefinition (Requirements Engineering)**
- Requirements Engineering Prozess
- Produktdatenmodellierung
- Analyseprototyp
- Lernziele

Anforderungen sind zentral für den Projekterfolg!



Source: Standish Group 1995 (www.standishgroup.com)



Herausforderungen für die Anforderungsdefinition 1(2)

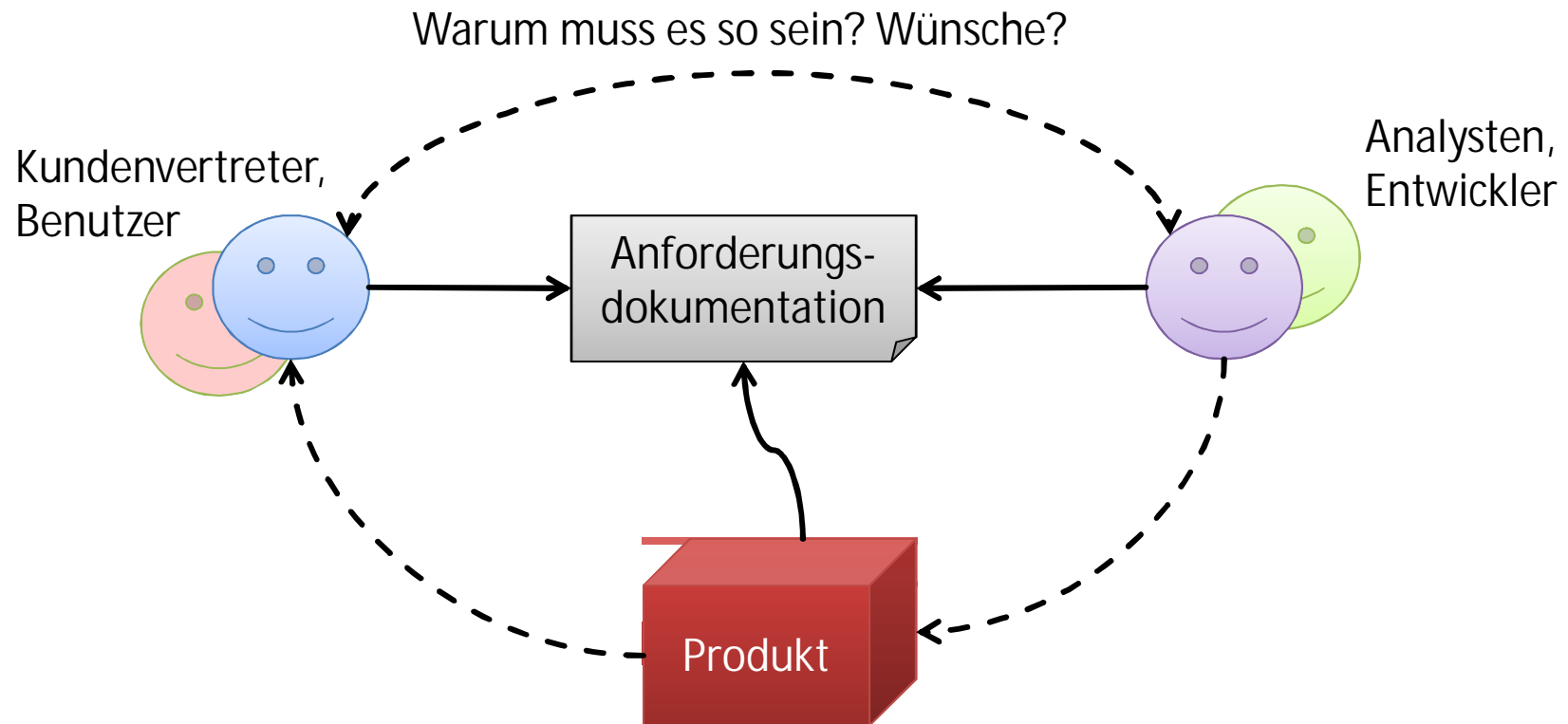
- **Unklare Zielsetzungen**, u.a. wegen
 - vieler Stakeholder,
 - schlechter Abstimmung zwischen den Stakeholdern,
 - geringes Vorstellungsvermögen (wie etwas aussehen könnte).
- **Hohe Komplexität**, u.a. weil
 - Kein Beteiligter alle Details des gewünschten Produkts überblickt,
 - Komplexe Geschäftsprozesse, -abgrenzungen, -regeln und –wünsche unterstützt werden sollen.
- **Sprachbarrieren**
 - Muttersprache vs. Fremdsprache.
 - Geschäftssprache vs. Informatik Jargon.



Herausforderungen für die Anforderungsdefinition 2(2)

- Sich (ständig) **ändernde Anforderungen**
 - ☐ Vage Anforderungen werden erst während der Entwicklung präzisiert.
 - ☐ Geschäftsprozesse ändern sich.
- **Schlechte Anforderungsqualität**
 - ☐ Unpräzise, mehrdeutig und/oder inkonsistent.
- **„Vergolden“ des Systems**
 - ☐ Anforderungen beschreiben nicht (unbedingt) benötigte Features
- **Unpräzise Planung**
 - ☐ Resultat aller aufgeführten Probleme

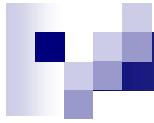
Anforderungen adressieren unterschiedlich Stakeholder



Anforderungsdefinition erfordert Ausgewogenheit



- Kunde und Entwickler müssen beteiligt sein.
- Typischerweise dominiert eine Seite.
 - Besser der Kunde dominiert.



Inhalt

- Einführung
- Anforderungen (Requirements)
- Anforderungsdefinition (Requirements Engineering)
- Requirements Engineering Prozess
- Produktdatenmodellierung
- Analyseprototyp
- Lernziele



Requirements Engineering Prozess

Das Requirements Engineering ist ein kooperativer, iterativer und inkrementeller Prozess, dessen Ziel es ist zu gewährleisten, dass:

- (1) alle relevanten Anforderungen bekannt und in dem erforderlichen Detaillierungsgrad verstanden sind,
- (2) die involvierten Stakeholder eine ausreichende Übereinstimmung über die bekannten Anforderungen erzielen,
- (3) Alle Anforderungen konform zu den Dokumentationsvorschriften dokumentiert bzw. konform zu den Spezifikationsvorschriften spezifiziert sind.

Requirements Engineering – Genereller Prozess

Requirements Engineering						
Requirements Analysis					Requirements Management	
Elicitation	Interpretation	Negotiation	Documentation	Validation	Change Management	Tracing
<ul style="list-style-type: none"> ➤ Identification of stakeholders and other sources of requirements (e.g., old existing systems) ➤ Gathering of raw requirements ➤ Target analysis 	<ul style="list-style-type: none"> ➤ Identification of requirements ➤ Structuring of requirements (merging, grouping, classification) ➤ Refinement (to satisfiable criteria) 	<ul style="list-style-type: none"> ➤ Identification of dependencies ➤ Identification of inconsistencies ➤ Resolution of inconsistencies ➤ Prioritization 	<ul style="list-style-type: none"> ➤ Specification of requirements (incl. intermediate results and assumptions) 	<ul style="list-style-type: none"> ➤ Contentual validation (correctness, completeness, consistency) ➤ Formal verification 	<ul style="list-style-type: none"> ➤ Handling of change requests ➤ Management of different versions of the requirements (history) ➤ Propagation of changes 	<ul style="list-style-type: none"> ➤ Recording of assumptions and requirement sources ➤ Recording of how requirements are implemented



Requirements Engineering – Genereller Prozess – Elicitation

- Projektstart
 - Zielsetzung überschlägig bekannt
- Ziele
 - Identifikation der **Stakeholder**
 - Identifikation von zusätzlichen **Quellen für Anforderungen**
 - Zum Beispiel: existierende Systeme, Standards, externe Experten ...
 - Sammeln der **Grob-Anforderungen**
 - Grob-Anforderungen = Hauptanforderungen die weiter verfeinert werden müssen.
- Vorgehensweise
 - Kundenbesuche und **Kundeninterviews**
 - Identifikation der wichtigen Dokumente und Stakeholder.
 - Auswertung der Interviews und **Identifikation von offenen Fragen**.
 - **Beantwortung offener Fragen** durch den Kunden und andere Stakeholder.
 - Möglicherweise in Form von Workshops.



Requirements Elicitation – Allgemeine Methoden

- Auf die Schwierigkeiten bei der Ermittlung (aller ?) Anforderungen an ein Software-Produkt wurde mehrfach hingewiesen. Wie kann man jedoch Anforderungen ermitteln (requirements elicitation)?
 - Well & Widdrig schlagen folgende Methoden vor:
 - Ermittlung aller Interessengruppen (stakeholders) am Software-Produkt und Durchführung von Interviews mit diesen Interessenten.
 - Ermittlung von Anforderungen aus verschiedenen Sichten (viewpoints). Dabei legen die Interessengruppen u.a. die Sichten auf ein Software-Produkt fest.
 - Durchführung von Workshops mit Brainstorming-Prozess und/oder Storyboard-Techniken, bei denen alle Interessentengruppen vertreten sind.
 - Ermittlung von Anwendungsfällen (typische Funktionsabläufe, engl. use cases) und ggf. Durchspielen dieser Fälle mit verteilten Rollen.
- D.L. Well, D. Widdrig: *Managing Software Requirements – A Unified Approach*, Addison Wesley (2000)



Requirements Elicitation – CRC-Kartenmethode (1)

■ Problem:

- Auftraggeber (**Domain Experts**) haben das Wissen über den Anwendungsbereich, kennen aber nicht Konzepte und Sprachen der Softwaretechnik.
- Auftragnehmer (**Software Experts**) sind zwar in der Erstellung von Modellen geschult, ihnen fehlt aber das Wissen über die Anwendung.

■ Lösung:


- **Gruppensitzung** (Workshop) mit etwa 5 Teilnehmern (sowohl „Domain Experts“ als auch „Software Experts“).
- Beim Durchspielen konkreter **Anwendungsfälle** werden Klassen (oder Entitäten) mit ihren typischen Aufgaben identifiziert.
- Für jede identifizierte Klasse wird eine „echte“ **Karteikarte** angelegt, die eine bestimmte Person verwaltet (und ihre Rolle spielt).

Requirements Elicitation – CRC-Kartenmethode (2)

- CRC-Karte = **C**lass-**R**esponsibility-**C**ollaboration-Card

<i><Class Name></i>	<i><Collaboration></i>
<i><Responsibilities></i>	

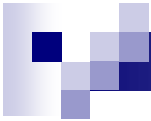
- **Class Name:** ein prägnanter Name für die Klasse/Entität.
- **Responsibilities:** die Aufgaben der Klasse/Entität, die beim Durchspielen von Anwendungsfällen identifiziert wurden.
- **Collaboration:** eine Liste anderer Klassen/Entitäten, die für die Erfüllung der Aufgaben der betrachteten Klassen gebraucht werden.
- **Achtung:**
 - In diesem Stadium keine Attribute, sondern Verantwortlichkeiten von Entitäten suchen! Die Bestimmung von Assoziationen und Attributen folgt später.



Requirements Engineering – Genereller Prozess – Interpretation

■ Ziele

- Bestimmung der **konkreten Anforderungen**.
- **Strukturierung** der Anforderungen.
 - Identifizierung von **Abhängigkeiten** zwischen Anforderungen
 - **Klassifikation** von Anforderungen (z.B. funktional, nicht-funktional, notwendig, wünschenswert, usw.)
 - **Zusammenführung** ähnlicher Anforderungen.
 - **Gruppieren** von Anforderungen, basierend z.B. auf
 - Abhängigkeiten
 - Klassifikation
 - **Verfeinerung** von Anforderungen
 - Verfeinerte Anforderungen können als Basis für Akzeptanztest dienen.



Requirements Engineering – Genereller Prozess – Interpretation

■ Vorgehensweise

- Analyse der Interview- und Meeting- Aufzeichnungen und anderen Materialien.
- Rückfragen oder zusätzliche Interviews.
 - Zurück zur Elicitation!
- Auflösung von Mehrdeutigkeiten
 - Wo möglich off-line, z.B. Rückfragen per Telefon.
 - Konfliktlösung durch Verhandlungen
- Parallel: Schreiben der Anforderungsdefinition
 - Arbeit an „Draft“-Version solange Interpretation noch nicht abgeschlossen ist.



Requirements Engineering – Genereller Prozess – Negotiation

■ Ziele

- Identifizierung aller **Abhängigkeiten**.
- Ermittlung und Auflösung aller **Inkonsistenzen**.
- Priorisierung von Anforderungen
 - Zum Beispiel in notwendige und optionale Features.

■ Vorgehensweise

- **Verteilung der Draft-Spezifikation**
- **Workshop** mit allen „wichtigen“ Personen (insb. „Entscheider“)
 - Präsentation der identifizierten Anforderungen (Folien, Mock-Ups)
 - Interaktive Priorisierung und Konkretisierung
 - Vermittlung/Schlichtung bei Konflikten



Requirements Engineering – Genereller Prozess – Dokumentation

■ Ziele

- Spezifikation der Anforderungen.
 - Anforderungen werden in der Dokumentation fixiert (u.a. durch eindeutigen Bezeichner)
- Dokumentation von **Zwischenergebnissen und Annahmen**.
- **Hintergrundinformation** zu Anforderungen.
 - Gründe für die Wichtigkeit/Unwichtigkeit von Anforderungen.
 - Diskutierte Realisierungsalternativen
- Zuweisung von **Attributen** zu Anforderungen
 - Priorisierung, Wichtigkeit, Abhängigkeiten, Realisierung in Version in, usw.



Requirements Engineering – Genereller Prozess – Dokumentation

■ Vorgehensweise

- Erstellung wird **durch die Mitarbeiter getrieben**.
 - Normalerweise ein **iterativer und inkrementeller Prozess** mit vielen aufeinanderfolgenden immer detaillierter werdenden Drafts.
 - Anforderungsspezifikation basiert auf den Resultaten der Workshops und Interviews.
 - Anforderungsspezifikation ist meist ein **Mix aus natürlicher Sprache, Tabellen, Anwendungsfällen und (UML) Diagrammen**, die häufig auch viele und lange **Anhänge** hat.

■ Anforderungsspezifikation V1.0

- ist meist erheblich länger als die Draft-Versionen.
- ist formal und technisch geschrieben.
- enthält kaum noch Hintergrundinformation zu Anforderungen.



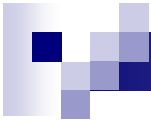
Requirements Engineering – Genereller Prozess – Validation

■ Ziele

- Validierung oder, falls möglich/notwendig, formale Verifikation der Anforderungsspezifikation.
 - Validierung oder Verifikation erfolgt basierend auf vorher existierenden Dokumenten (z.B. Ausschreibungen, Dokumentation von „Legacy Systems“) und den Kundenwünschen.

■ Vorgehen

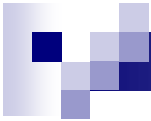
- Reviews durch Experten
 - Formaler Rahmen wird mittels Checklisten, Fragebögen, Richtlinien oder Standards geprüft.
 - Prüfung der Kundenwünsche erfolgt
 - basierend auf den Ergebnissen der Workshops und Interviews.
 - speziell unter Berücksichtigung der als besonders wichtig identifizierten Kundenwünsche.



Requirements Engineering – Genereller Prozess – Change Management

■ Ziele

- Management von **Change Requests**
(Änderungswünsche) für Anforderungen
- Management **verschiedener Versionen** von
Anforderungen
 - Jede Änderung führt meist zu einer neuen Version der
Anforderungsspezifikation.
- Organisation der **definierten Weitergabe** von
Anforderungsänderungen



Requirements Engineering – Genereller Prozess – Change Management

■ Vorgehen

- Typischer Ablauf bei transparentem Change Management
 - (1) Change request
 - (2) **Change Control Board**
 - (3) Entscheidung (des Change Control Boards)
 - (4) Zuweisung des Änderungsauftrags (kostet Geld)

- Empfehlung: Nutzung eines Requirement Engineering-Werkzeugs
 - Werkzeug übernimmt Change Management und Versionskontrolle.
 - Change Requests und Änderungen werden nachverfolgbare Objekte.
 - Weitergabe von Änderungsinformationen erfolgt organisationsspezifisch.

- Change Management ohne Werkzeug
 - Change Management erfolgt manuell.
 - Nachverfolgbarkeit kann problematisch sein.



Requirements Engineering – Genereller Prozess – Tracing

■ Aufgaben

- ☐ Annahmen protokollieren.
- ☐ Entscheidungen protokollieren.
- ☐ Protokoll führen darüber, **welche Annahmen zu welchen Entscheidungen geführt haben** und wie sich Anforderungen hierdurch verändert haben.

■ Vorgehen

- ☐ Effektives Tracing erfordert die Nutzung eines Requirement Engineering-Werkzeugs
- ☐ Manuelles Tracing basierend auf natürlich-sprachlichen Dokumenten ist meist sehr aufwendig und problematisch.



Einführung – Analyse besteht aus ...

- Requirements Engineering
 - Identifikation der Anforderungen an das Verhalten des Systems.
- Produktdatenmodellierung
 - Erstellung eines **konzeptionellen Datenmodells** (**Domänenmodell**) des Anwendungsbereichs (Domäne).
- Analyseprototyp



Inhalt

- Einführung
- Anforderungen (Requirements)
- Anforderungsdefinition (Requirements Engineering)
- Requirements Engineering Prozess
- **Produktdatenmodellierung**
- Analyseprototyp
- Lernziele



Produktdatenmodellierung

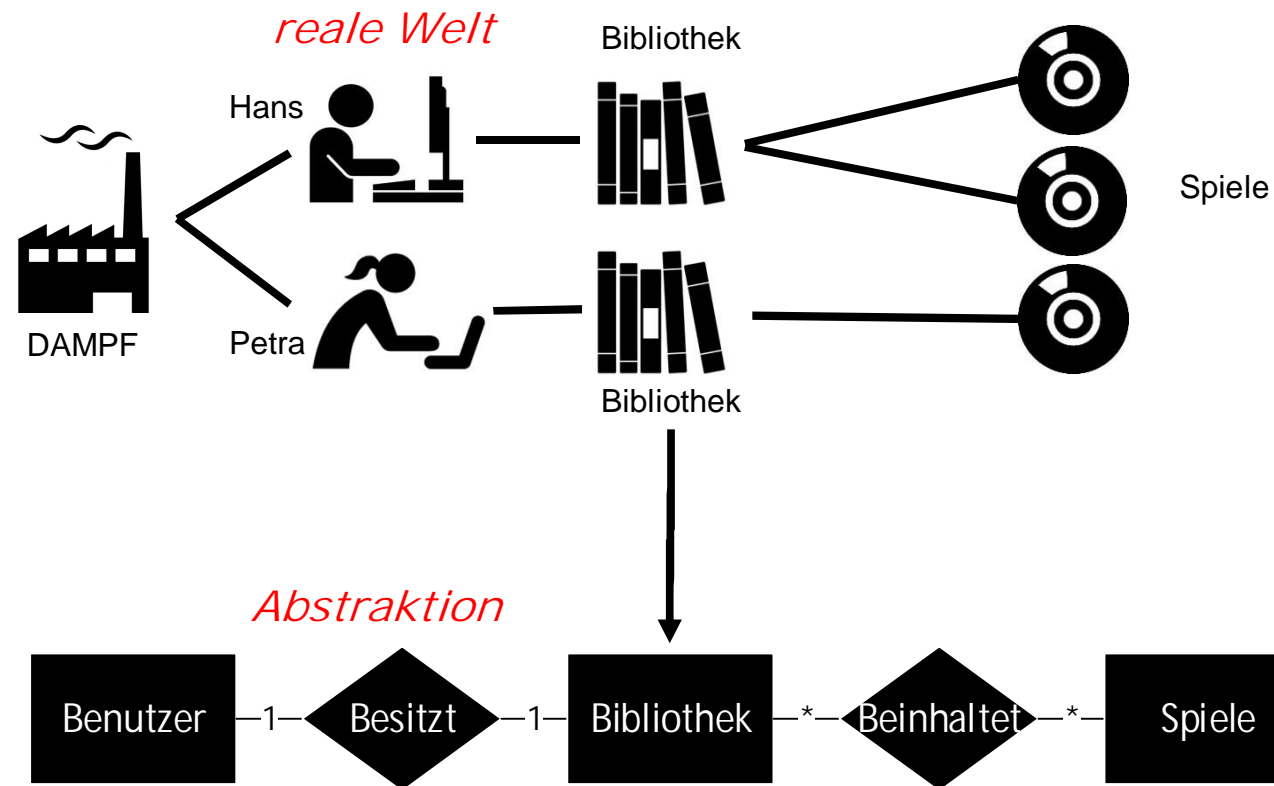
- Aufgabe:

- ☐ Aufbauend auf den Ergebnissen des Requirement Engineerings wird ein **konzeptuelles Datenmodell (Domänenmodell)** des Anwendungsbereiches erstellt. Es handelt sich dabei nicht um das interne Datenmodell des später realisierten Softwareproduktes!

- Vorgehensweise:

- ☐ Bestimmung von **Entitätskandidaten** mit verschiedenen Methoden (Analyse Use Cases, CRC-Karten, usw.).
- ☐ Bestimmung wichtiger **Assoziationen** (Beziehungen) und **Attribute** zu den identifizierten Entitäten.

Produktdatenmodellierung – Von der realen Welt zum Domänenmodell





Produktdatenmodellierung – Von der realen Welt zum Domänenmodell

■ Vorgehen

- ☐ Alle bereits vorgestellten Verfahren zur Ermittlung von funktionalen und nicht-funktionalen Anforderungen (siehe Folien zur „Requirements Elicitation“).

■ Kandidaten für Entitäten:

- ☐ alle physischen, berührbaren Gegenstände (wie Auto, Vertrag)
- ☐ Personenrollen (wie Benutzer, Administrator)
- ☐ Institutionen (wie Firma)
- ☐ abstrakte Begriffe (wie Adresse)
- ☐ durchzuführende Transaktionen (wie Ausleihe, Bezahlung)
- ☐ eintretende Ereignisse (wie Downloadfehler)
- ☐ ...

■ Man beachte:

- ☐ Nicht nur passive Datenbestände werden als Entitäten modelliert! Auch Operationen, Transaktionen, Ereignisse, Prozesse, ... können sinnvolle Entitätskandidaten sein.



Produktdatenmodellierung – Identifikation von Attributen

- Attribute „speichern“ **primitive Eigenschaften** von Entitäten.
- Attribute halten Informationen darüber was für eine Entität in einem bestimmten Kontext von Interesse ist
- Beispiele:
 - Name, Vorname, Kreditkartennummer einer Person
 - Beschreibungen, Mindestalter für Spiele



Produktdatenmodellierung – Identifikation von Assoziationen

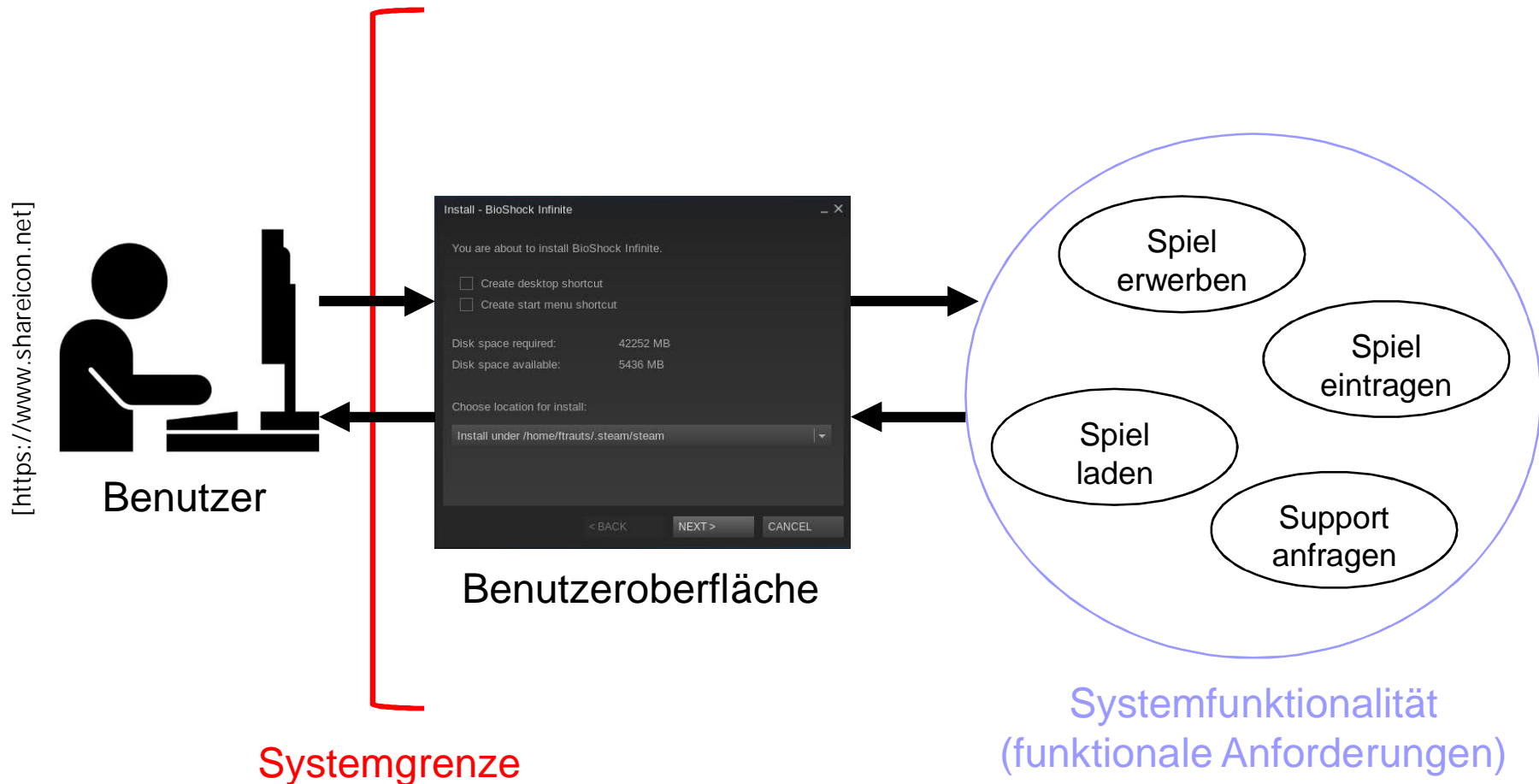
- Assoziationen sind (in aller Regel zweistellige) Relationen zwischen Entitäten. Ihre Instanzen sind Links, die Instanzen der vorgegebenen Entität verbinden.
- In erster Linie sind solche Assoziationen aufzuführen, die nicht nur temporär während der Abarbeitung einer Operation (Systemfunktion) existieren.
- Kandidaten für Assoziationen:
 - A ist ein logischer/physikalischer Teil von B (Benutzer - Unternehmen)
 - A überwacht/besitzt B (Unternehmen - Spiel)
 - A benutzt B (Benutzer - Spiel)
 - A verweist auf B (Ausleihvorgang - Leihvertrag)
 - A kommuniziert mit B (Benutzer - Support)



Inhalt

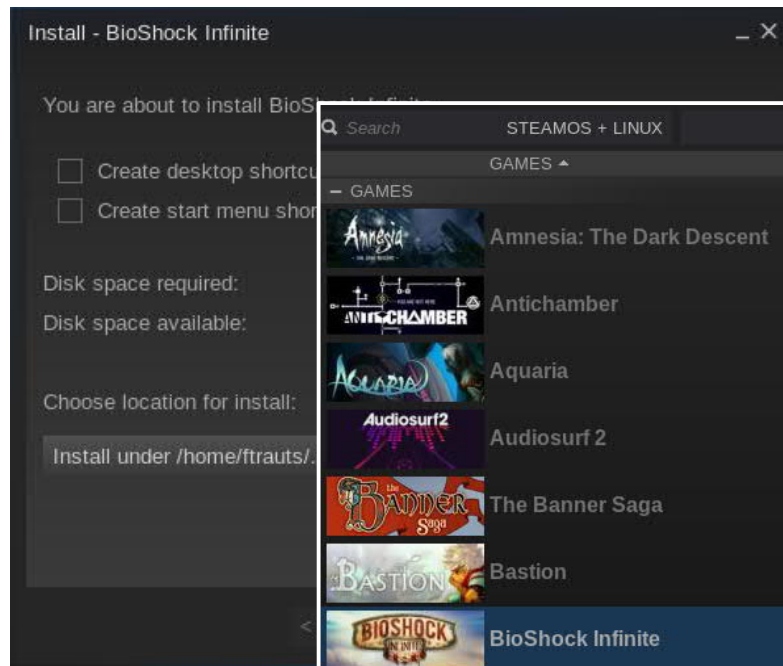
- Einführung
- Anforderungen (Requirements)
- Anforderungsdefinition (Requirements Engineering)
- Requirements Engineering Prozess
- Produktdatenmodellierung
- **Analyseprototyp**
- Lernziele

Analyseprototyp – Systemgrenze, Anwendungsfälle, Ablaufmodellierung

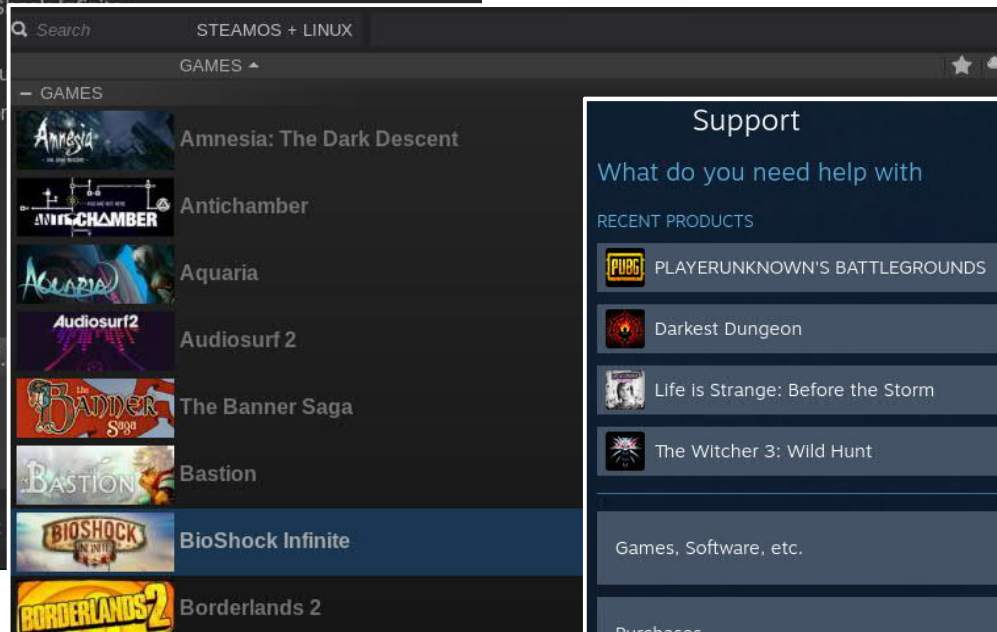


Analyseprototyp: Benutzeroberfläche für Anwendungsfälle + Ablaufsteuerung

Spiel-Download Fenster



Spielebibliothek Fenster



Support Fenster

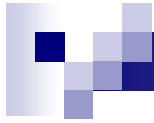


- Erstellung von Benutzeroberfläche mit einem GUI-Tool und einfacher Ablaufsimulation wird häufig als Analyseprototyp (Rapid Prototype) verwendet.



Inhalt

- Einführung
- Anforderungen (Requirements)
- Anforderungsdefinition (Requirements Engineering)
- Requirements Engineering Prozess
- Produktdatenmodellierung
- Analyseprototyp
- **Lernziele**



Lernziele

- Anforderungen
 - Was sind Anforderungen?
 - Welche Arten von Anforderungen gibt es?
- Requirements Engineering
 - Was ist Requirements Engineering?
 - Welche Probleme gibt es beim Requirements Engineering?
- Requirements Engineering Prozess
 - Welche Phasen umfasst der Requirements Engineering Prozess?
 - Was sind die Ziele und Vorgehensweisen der Phasen?
- Was ist Produktdatenmodellierung?
 - Domänenmodell
- Wie kann man einfache Analyseprototypen erstellen?