



Software Engineering Anforderungsanalyse



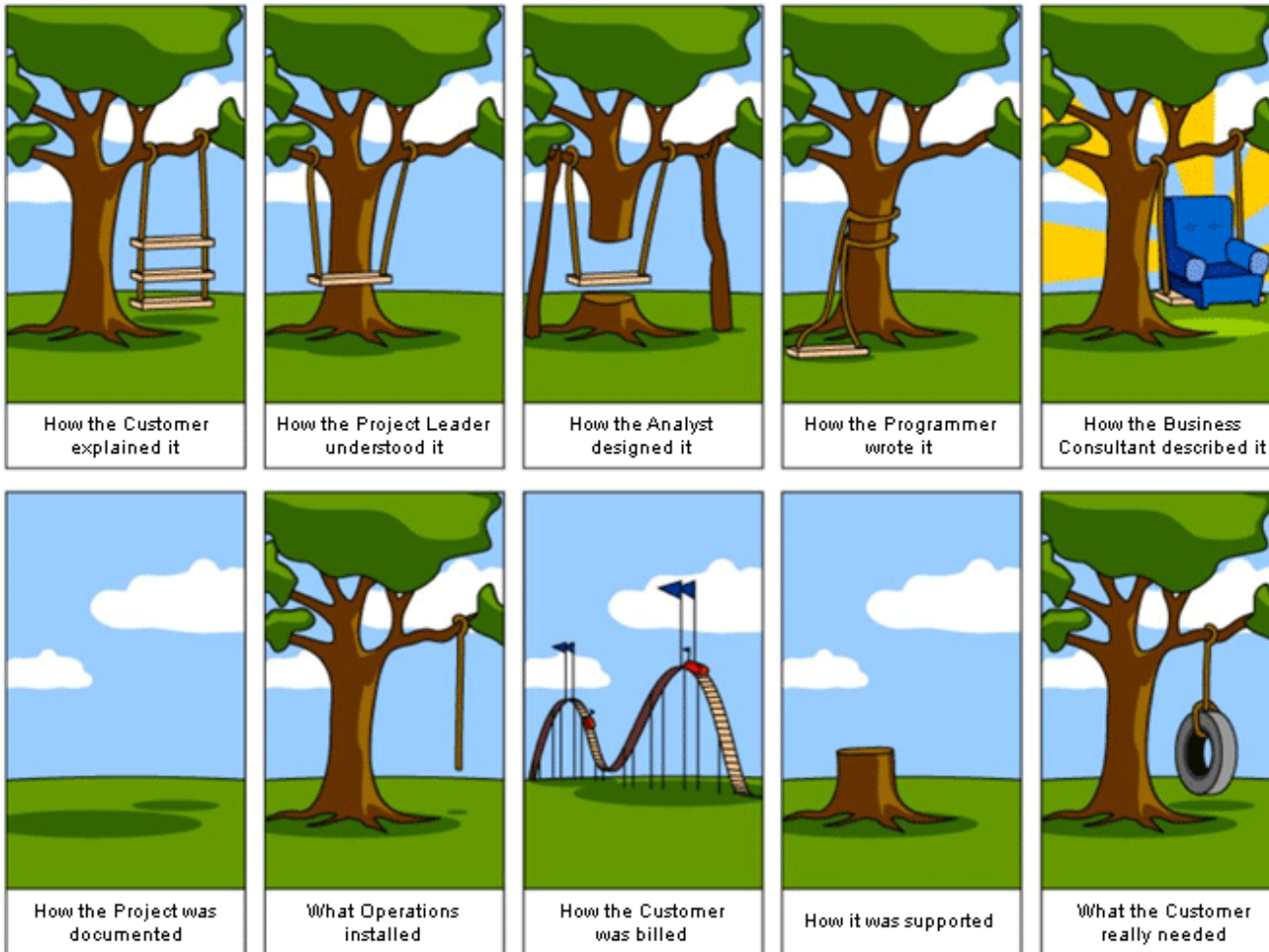
Klaus Ostermann

Agenda

- ▶ Funktionale und nichtfunktionale Anforderungen
- ▶ Benutzeranforderungen
- ▶ Systemanforderungen
- ▶ Schnittstellenspezifikation
- ▶ Das Pflichtenheft

Warum Anforderungsanalyse?

Bauen wir das Richtige?



Beispiel

Zeitschema

Kommission: _____

Bitte ankreuzen wo Sie keinesfalls mitmachen können, und senden Sie das ausgefüllte Formular bis _____ ans Dekanat zurück.

	Jan 2002																					Feb 2002												
	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	
8:00 - 9:00																																		
9:00 - 10:00																																		
10:00 - 11:00																																		
11:00 - 12:00																																		
12:00 - 13:00																																		
13:00 - 14:00																																		
14:00 - 15:00																																		
15:00 - 16:00																																		
16:00 - 17:00																																		
17:00 - 18:00																																		

Bemerkungen: _____

Unterschrift: _____

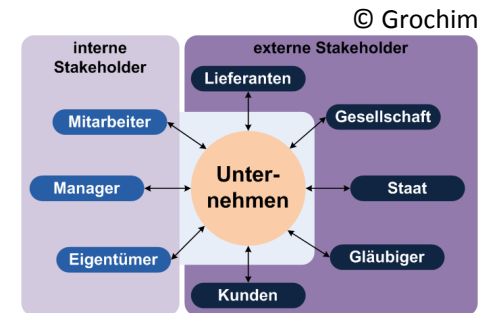
Beispiel (Fortsetzung)

- ▶ “Naja, eigentlich brauchen wir ein Formular das wir per Email verteilen können und einen Ort (HTML) wo wir die ausgefüllten Formulare ablegen können und einen Algorithmus der uns ausrechnet wann wir uns Treffen können. Achja, am besten wäre es wenn, falls nicht alle können, irgendwie berücksichtigt wird dass alle wichtigen Personen dabei sind. Es wäre auch toll, wenn es eine Möglichkeit gäbe ob jeder so ein Formular ausgefüllt hat. Wenn nicht sollte es am besten eine Liste der übrigen Personen geben. Und wenn's irgendwie möglich ist sollte auch eine automatische Einladung an alle beteiligten Personen geschickt werden können.

Wie kommt man hier zu klaren Anforderungen?

Probleme

- ▶ Kunden wissen nicht was sie wirklich wollen
- ▶ Kunden benutzen ihre eigene Fachsprache
- ▶ Verschiedene Stakeholder* können widersprüchliche Anforderungen haben
- ▶ Politische und Organisatorische Faktoren können Anforderungen beeinflussen
- ▶ Anforderungen ändern sich während der Analyse und Entwicklung
- ▶ Neue Stakeholder mischen sich ein



* Stakeholder = Anspruchsträger / Interessierte / Betroffene / Projektbeteiligte / Anspruchsgruppen

Wann sind wir fertig?

- ▶ Szenario: Neue Firma, 3 Mitarbeiter (Gehalt 45.000 EUR)
- ▶ Bekommen Auftrag von Firma \$BigCooperation
 - ▶ geschätzter Aufwand 9 Bearbeiterjahre
 - ▶ Festpreis 500.000 EUR, 250.000 EUR sofort, Rest nach Abnahme
- ▶ Fertigstellung nach 3 Jahren
- ▶ Firma verweigert Abnahme, fordert Nacharbeiten
- ▶ -> Schuldenfalle

Anforderungen

Definition: Anforderung

- ▶ **Benutzeranforderungen** sind Aussagen in **natürlicher Sprache** sowie Diagramme zur Beschreibung der **Dienste**, die das System leisten soll, und der **Randbedingungen**, unter denen es betrieben wird.
- ▶ **Systemanforderungen** legen die Funktionen, Dienste und Beschränkungen detailliert fest. Das **Pflichtenheft** (auch **funktionale Spezifikation** genannt) sollte präzise sein. Es muss genau definieren, was implementiert werden soll. Es kann als **Teil des Vertrages** zwischen dem Käufer und dem Softwareentwickler dienen.

Beispiel

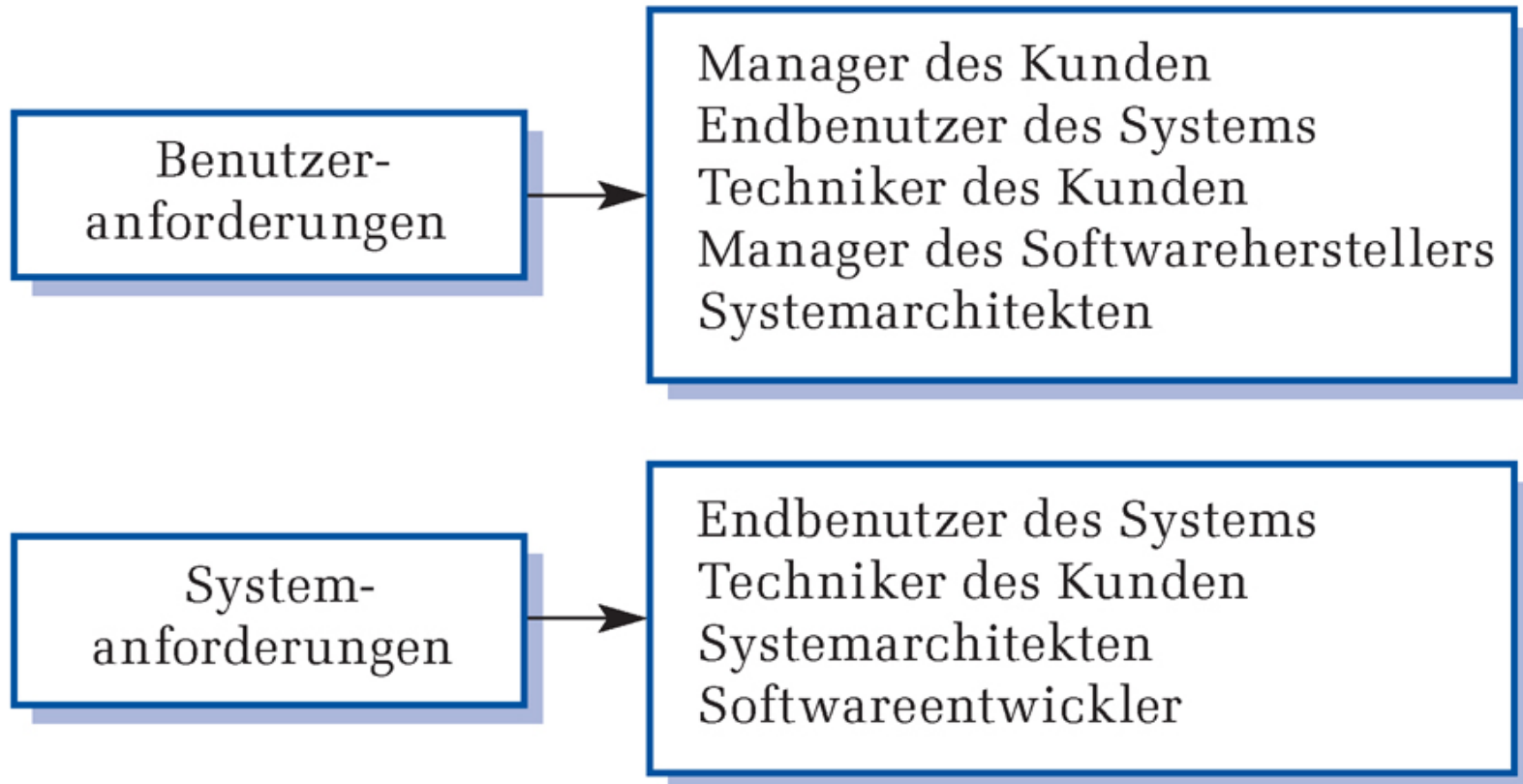
Definition einer Benutzeranforderung

1. LIBSYS soll alle Daten nachverfolgen, die von den Lizenzagenturen in Großbritannien und anderswo benötigt werden.

Spezifikation der Systemanforderungen

- 1.1 Bei der Anforderung eines Dokuments von LIBSYS soll dem Antragsteller ein Formular angezeigt werden, das Einzelheiten über den Benutzer und die Anforderung aufnimmt.
- 1.2 LIBSYS-Anforderungsformulare sollen vom Datum der Anforderung an fünf Jahre lang auf dem System gespeichert werden.
- 1.3 Alle LIBSYS-Anforderungsformulare müssen nach dem Benutzer, dem Titel des angeforderten Materials und dem Lieferanten indiziert werden.
- 1.4 LIBSYS soll ein Protokoll aller Anforderungen unterhalten, die an das System gestellt wurden.
- 1.5 Bei Material, das Leihrechten unterliegt, müssen die Angaben der Ausleihvorgänge monatlich an die in LIBSYS registrierten Lizenzagenturen gesendet werden.

Zielgruppen



Festlegung der Anforderungen: Schritte

- ▶ **Anforderungsermittlung:** Sammeln von (funktionalen und nicht-funktionalen) Anforderungen
 - ▶ z.B. durch Anwendergespräche, Dokumenten-Studium
- ▶ **Anforderungsanalyse:** Klassifizierung, Bewertung, Vergleich und Prüfung
 - ▶ z.B. Kosten-/Nutzen-Aspekte, Konsistenz, Vollständigkeit
- ▶ **Anforderungsbeschreibung:** Beschreibung in einheitlicher Form (z.B. als Anwendungsfälle)
- ▶ **Anforderungsrevision:** Erneute Prüfung/Änderung von Anforderungen
 - ▶ ggf. nur in formellem Änderungsverfahrens

Inhalt der Anforderungsbeschreibung

- ▶ Zielsetzung
- ▶ allgemeine Beschreibung
- ▶ Definitionen und Abkürzungen
- ▶ Produktumfeld
- ▶ funktionale Anforderungen
- ▶ nicht-funktionale Anforderungen
- ▶ Abnahmekriterien
- ▶ Glossar, Index, Referenzen

IEEE 830-98
Standard für
Anforderungs-
dokumente

Funktionale vs. nichtfunktionale Anforderungen

Funktionale Anforderungen

- ▶ **Was** soll das System leisten?
- ▶ Welche Dienste soll es anbieten
- ▶ Eingaben, Verarbeitungen, Ausgaben
- ▶ Verhalten in bestimmten Situationen, ggf. was soll es explizit nicht tun

Nichtfunktionale Anforder.

- ▶ **Wie** soll das System/ einzelne Funktionen arbeiten?
- ▶ Qualitätsanforderungen wie Performanz und Zuverlässigkeit
- ▶ Anforderungen an die Benutzbarkeit des Systems

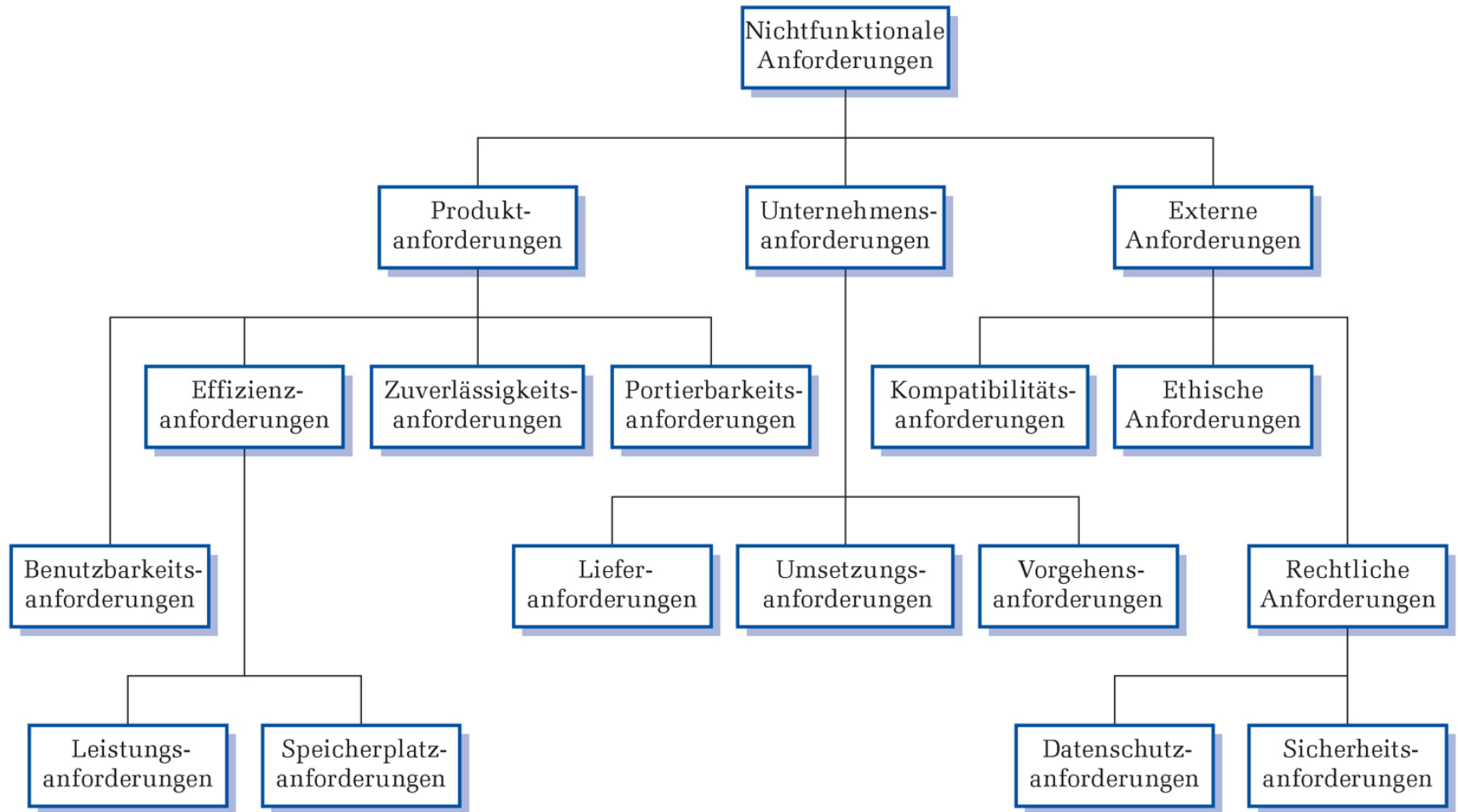
Beispiel: Funktionale Anforderungen

- ▶ Funktion: Vorlesung eintragen
- ▶ Eingaben: Raum, Zeit und Titel einer Vorlesung.
- ▶ Verarbeitungsschritte:
 - ▶ Prüfe, ob der Vorlesungstitel schon vergeben ist
 - ▶ Prüfe, ob der Raum zur angegebenen Zeit schon vergeben ist
 - ▶ Wenn nicht, wird die neue Vorlesung eingetragen und die Daten der Vorlesung werden angezeigt.
 - ▶ Falls vergeben, wird die Vorlesung nicht eingetragen und eine entsprechende Fehlermeldung wird angezeigt.
- ▶ Ausgaben: Die Vorlesung wird angezeigt oder ein Fehler wird gemeldet.

Beispiele: Funktionale Anforderungen

- ▶ Aktionen, die vom System ausgeführt werden sollen
 - ▶ Bsp.: Das System muss Ausleihgegenstände in den Bestand aufnehmen können.
- ▶ Systeminteraktionen, die dem Nutzer ermöglicht werden
 - ▶ Bsp.: Das System muss es dem Administrator bei der Aufnahme eines Ausleihgegenstandes in den Bestand ermöglichen, den Autor, den Titel und die ISBN einzugeben.
- ▶ allg. funkt. Vereinbarungen u. Einschränkungen
 - ▶ Bsp.: Der Client ist für den Kommunikationsaufbau zuständig.

Nicht-funktionale Anforderungen



Beispiele: Nicht-funktionale Anforderungen

- ▶ **technische Anforderungen**
 - ▶ Das System muss mit Java entwickelt werden und muss in der Sun Java VM 1.5 laufen
- ▶ **ergonomische Anforderungen**
 - ▶ Das System muss die gespeicherten Objekte formatiert ausgeben können (Formatvorgabe).
 - ▶ Die Benutzerführung erfolgt in deutsch
- ▶ **Anforderungen an die Dienstqualität**
 - ▶ Das System muss jede Anfrage des Benutzers innerhalb von 30 Sekunden ausführen (auf System XY).
 - ▶ Der Speicherbedarf darf 512mb nicht übersteigen

Beispiele: Nicht-funktionale Anforderungen (2)

- ▶ **Zuverlässigkeit**
 - ▶ Die Verfügbarkeit des Systems muss bei 99.999% liegen
- ▶ **Anforderungen an den Entwicklungsprozess**
 - ▶ Der Entwickler muss mit dem Kunden monatliche Reviews der zu erstellenden Dokumente durchführen.
- ▶ **rechtlich-vertragliche Anforderungen**
 - ▶ Der Kunde leistet für jeden abgenommenen Meilenstein ein Drittel der vertraglich vereinbarten Summe für die Entwicklung des Systems.
 - ▶ Die deutsche Datenschutzrichtlinie muss erfüllt sein
- ▶ **Interoperabilität**
 - ▶ Das System muss Daten mit MS Excel austauschen können

Überprüfbarkeit

- ▶ Alle Anforderungen müssen überprüfbar sein
- ▶ Ein objektiver Dritter muss entscheiden können
 - ▶ Notfalls Abnahme vor Gericht durchsetzen

Ein Systemziel

Das System sollte für erfahrene Benutzer einfach zu bedienen und so aufgebaut sein, dass Fehler durch den Benutzer minimiert werden.

Eine verifizierbare nichtfunktionale Anforderung

Es soll erfahrenen Benutzern möglich sein, nach einer Schulung von insgesamt zwei Stunden alle Systemfunktionen zu verwenden. Nach dieser Schulung sollte der durchschnittliche Wert gemachter Fehler bei erfahrenen Benutzern nicht höher als zwei pro Tag sein.

Messbarkeit

► Beste Überprüfbarkeit bei messbaren Metriken

Eigenschaft	Maßeinheit
Geschwindigkeit	Ausgeführte Transaktionen/Sekunde
	Reaktionszeit auf Benutzereingabe oder Ereignis
	Bildschirmauffrischungszeit
Größe	Kilobyte
	Anzahl der Speicherbausteine
Benutzerfreundlichkeit	Schulungsdauer
	Anzahl der Hilfeseiten
Zuverlässigkeit	Durchschnittliche Zeit bis zu einer Fehlfunktion
	Wahrscheinlichkeit der Nichtverfügbarkeit
	Quote für das Auftreten von Fehlern
	Verfügbarkeit
Stabilität	Zeit bis zum Neustart nach einer Fehlfunktion
	Anteil der Ereignisse, die zu Fehlfunktionen führen
	Wahrscheinlichkeit für Datenzerstörung bei Fehlfunktion
Portierbarkeit	Anteil der plattformabhängigen Anweisungen
	Anzahl der Zielsysteme

Messbarkeit (Beispiele)

- ▶ Das System soll zügig reagieren
 - ▶ 80 % aller Anfragen sollen unter 0.1 Sekunde beantwortet werden, 99.99 % aller Anfragen unter 2 Sekunden (Test-Hardware spezifizieren!)
- ▶ Das System soll auch im Mehrbenutzerbetrieb schnell reagieren
 - ▶ Auf XY-Server Verarbeitung von 200 Anfragen pro Sekunde von 10 unterschiedlichen Systemen mit 100mbit Ethernet-Anbindung
- ▶ Das System soll stabil und ausfallsicher sein
 - ▶ Verfügbarkeit von 99.99 %, Neustart innerhalb von 30 Sekunden, das System darf nicht aufgrund falscher Eingaben abstürzen

Messbarkeit (Beispiele 2)

- ▶ Das System soll gut getestet werden
 - ▶ Automatisierte Unit-Tests erzielen eine Testabdeckung von insgesamt 95% und 100% in Modul X
 - ▶ Durchführung eines Benutzbarkeittests mit zwei nicht-eingewiesenen Nicht-Informatik-Studenten, diese können Aufgabe X ohne Erklärung durchführen
 - ▶ Das System wird mit Simulator X für 2 Wochen fehlerfrei im Dauerbetrieb ausgeführt

Belastungstest:

- mit Linuxprogramm "ab"
- virtueller Server bei webvariants
- 1000 Anfragen, 10 gleichzeitig:
 - Forenübersicht
 - ∅ 11.68 Anfragen pro Sekunde
 - 428 Sekunden insg.
 - max. 6.5 Sekunden

Integrationstests:

- 28 Tests, 27 erfolgreich
- 1 schlägt wegen Rechte fehl
- Gesamtdauer 30 Sekunden

Live-Test:

- 23.09.2009 bis 07.10.2009
- leider FeedBack erst ab 01.10.2009

Nachvollziehbarkeit

- ▶ Dokumentieren woher Anforderungen kam
- ▶ Wer hat sie aufgestellt, für wen ist sie wichtig

- ▶ Vermeidet Phantome
- ▶ Strukturiert Diskussionen
- ▶ Erlaubt gezielte Rückfragen/Prüfen ob erfüllt

Abgrenzungskriterien

- ▶ Was soll nicht gemacht werden?
- ▶ Manchmal sinnvoll dies expliziert zu klären
- ▶ Dem Kunden klar machen worauf er verzichtet
- ▶ Teils Schutz vor Änderungsanträgen

- ▶ Beispiele
 - ▶ Kein Fokus auf intuitive Bedienung, sondern Spezialsoftware für geschulte Benutzer
 - ▶ Schützt nicht vor Fehlern des Benutzers
 - ▶ Festplattenspeicher wird als nicht-limitiert betrachtet
 - ▶ Multilinguale Benutzeroberfläche nicht geplant

Prüfung von Anforderungen

- ▶ Wird der Bedarf des Kunden **vollständig** abgedeckt?
- ▶ **Verständlich** formuliert?
- ▶ **Konsistent** mit den anderen Anforderungen?
- ▶ **Realistisch** mit Budget und Technologie?
- ▶ Anforderung **prüfbar**?
- ▶ **Änderbar** ohne Einfluss auf andere Anforderungen?

- ▶ Regelmäßige Reviews
- ▶ Kunden und potentielle Benutzer in Anforderungsanalyse mit einbeziehen

Anwendungsfälle

Anwendungsfälle

- ▶ Grundidee: Gliederung der Funktionalität in logisch zusammengehörige und handliche funktionale Einheiten, Beschreibung in standardisierter Form
- ▶ Anwendungsfall: abgeschlossene, zusammenhängende Einheit; Teil der Funktionalität des Systems
- ▶ Anwendungsfallbeschreibung:
 - ▶ Titel:
 - ▶ Kurzbeschreibung:
 - ▶ Akteure:
 - ▶ Vorbedingungen:
 - ▶ Beschreibung des Ablaufs:
 - ▶ Auswirkungen:
 - ▶ Anmerkungen:
- ▶ Die Gliederung ist nicht fix und sollte je nach Bedarf erweitert werden (z.B. Hinzufügen von Prioritäten, Stakeholdern ...)

Beispiel: Anwendungsfall

- ▶ Titel: Vorlesung eintragen
- ▶ Kurzbeschreibung: Dozent gibt Raum, Zeit und Titel einer Vorlesung ein und legt einen Vorlesungseintrag an.
- ▶ Akteur: Dozent
- ▶ Vorbedingungen: Eine Vorlesung mit diesem Titel gibt es noch nicht.
- ▶ Beschreibung des Ablaufs:
 - ▶ Prüfe, ob der Raum zur angegebenen Zeit schon vergeben ist
 - ▶ Wenn nicht, wird die neue Vorlesung eingetragen und die Daten der Vorlesung werden angezeigt.
 - ▶ Falls vergeben, wird die Vorlesung nicht eingetragen und eine entsprechende Fehlermeldung wird angezeigt.
- ▶ Auswirkungen: Die Vorlesung wird angezeigt oder ein Fehler wird gemeldet.
- ▶ Anmerkungen:

Aktoren

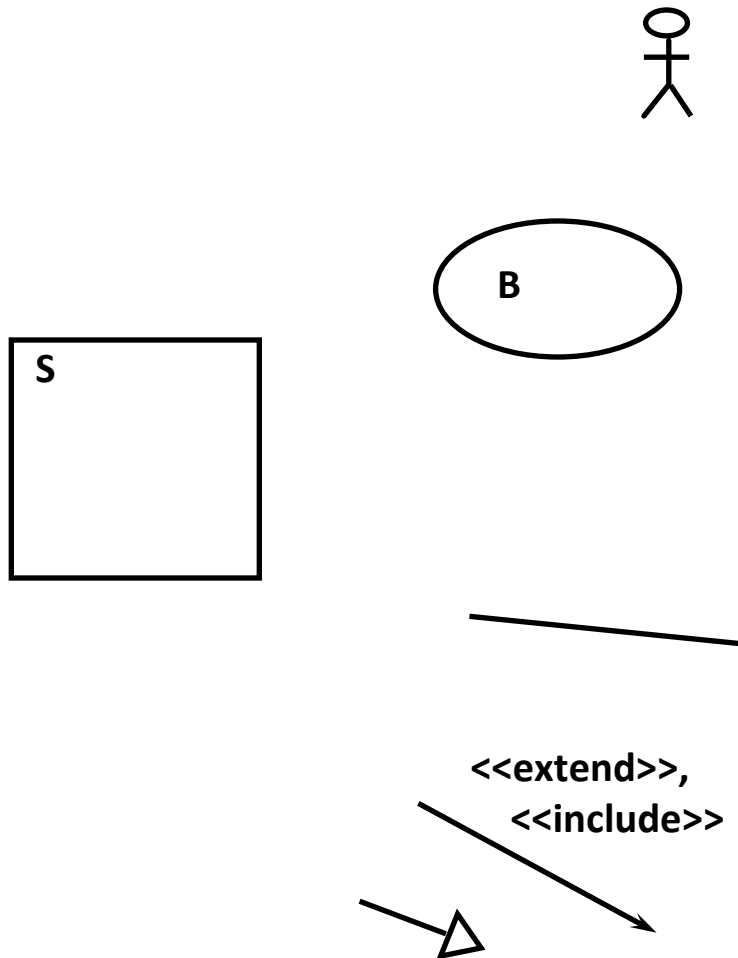


Aktor: Objekt der Systemumgebung, das mit dem System interagiert (und einen oder mehrere Anwendungsfälle auslösen kann).
Aktoren können Personen (System-Nutzer), externe Geräte oder mit dem System verbundene Nachbarsysteme sein.

Aktoren tauschen mit dem System Nachrichten aus und können als Sender und/oder Empfänger von Nachrichten auftreten.

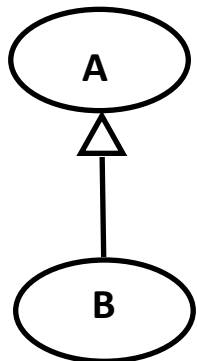
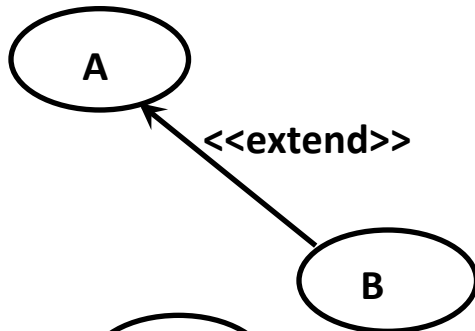
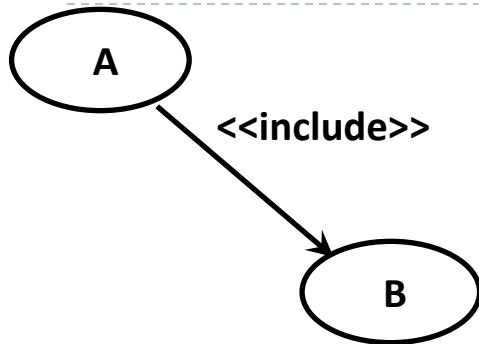
Aktoren sind i.a. selbst nicht Bestandteile des Systems. Oft müssen Daten über sie jedoch (z.B. zur Regelung der Zugangsberechtigung) mit verwaltet werden.

Elemente von Anwendungsfalldiagrammen



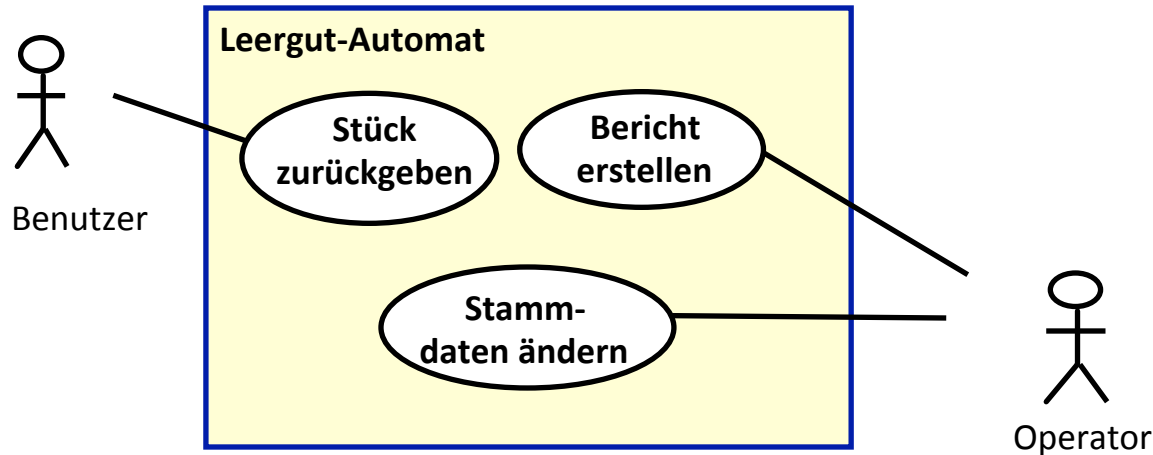
- ▶ Aktor,
- ▶ Anwendungsfall mit Bezeichner B,
- ▶ Systemgrenze (mit Systembezeichner S),
- ▶ Kommunikationsbeziehung (zwischen Aktoren und Anwendungsfällen)
- ▶ Beziehung zwischen Anwendungsfällen

Beziehungen zwischen Anwendungsfällen



- ▶ Ausführung von Anwendungsfall A schließt die Ausführung von Anwendungsfall B ein.
 - ▶ Bsp: "Auftrag bearbeiten" schließt "Zahlung veranlassen" ein.
- ▶ Ausführung von Anwendungsfall A kann (unter bestimmten Bedingungen) erweitert werden durch eine Ausführung von Anwendungsfall B.
 - ▶ Bsp: Beim "Auftrag bearbeiten" wird - falls eine Anfrage dafür vorlag - noch ein Katalog versandt.
- ▶ Anwendungsfall A generalisiert Anwendungsfall B, d.h. B steht für eine Menge von Spezialfällen von A.
 - ▶ Bsp: "Auftrag bearbeiten" generalisiert "Kaufauftrag bearbeiten" und "Verkaufsauftrag bearbeiten"

Beispiel: ein einfaches Anwendungsfalldiagramm

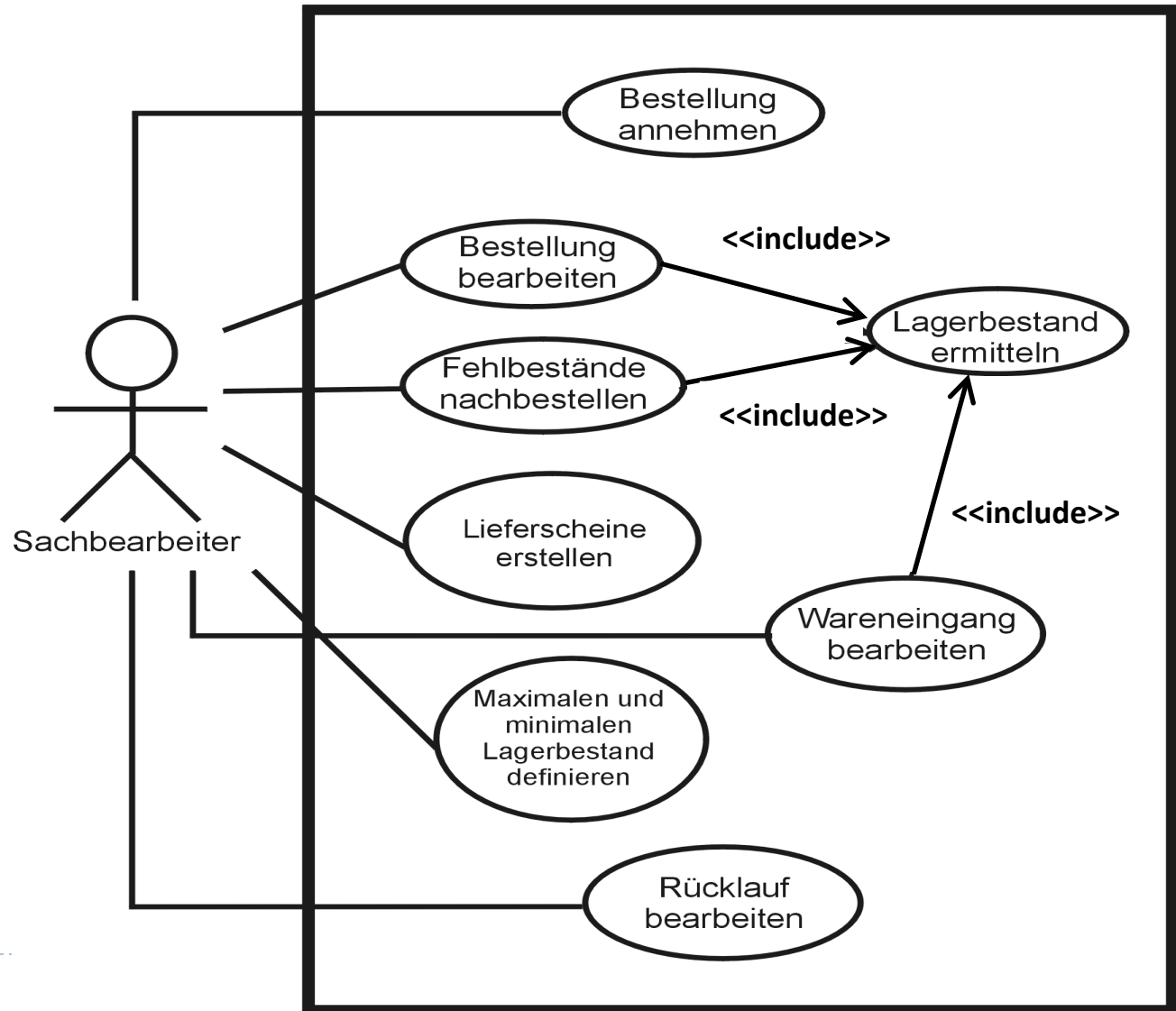


- ▶ *Stück zurückgeben* wird durch den Kunden (= Automatenbenutzer) ausgelöst, der eine Menge von Dosen, Flaschen oder Kisten zurückgeben will.

Beschreibung des Ablaufs:

- ▶ Mit jedem zurückgegebenen Stück erhöht das System die Anzahl der registrierten Stücke sowie die Tagesgesamtanzahl für die betreffende Kategorie.
- ▶ Wenn der Kunde alle Stücke abgeliefert hat, drückt er einen Quittungsknopf und erhält eine Quittung über alle abgelieferten Stücke und deren Gesamtanzahl.

Weiteres Beispiel: Anwendungsfalldiagramm



Anwendungsfälle

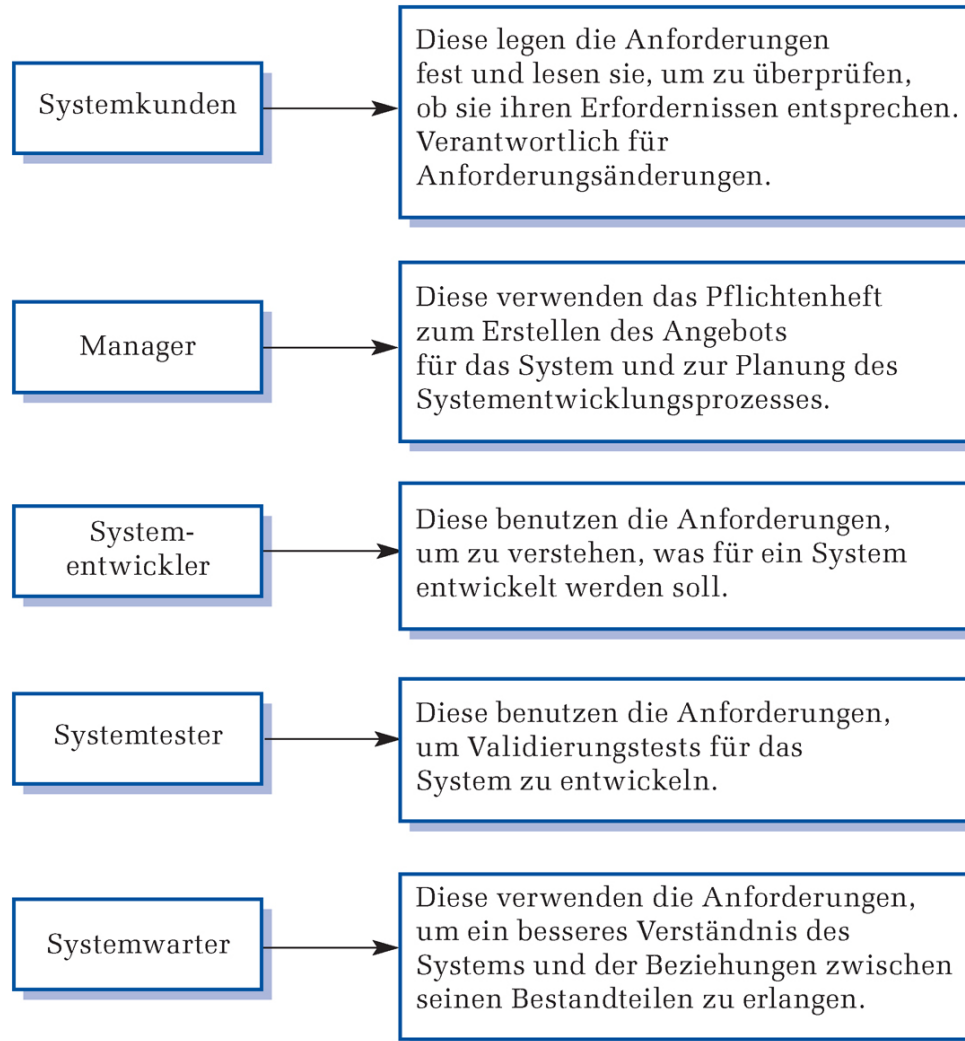
- ▶ Erstellen von Anwendungsfällen ist Textarbeit
- ▶ Anwendungsfalldiagramme sind nur zur Übersicht, ersetzen aber nicht die textuelle Beschreibung

Pflichtenheft

Pflichtenheft

- ▶ Offizielle Aufstellung was von Entwicklern erwartet wird
- ▶ Enthält Benutzer- und Systemanforderungen
- ▶ Hat bei externen Kunden Vertragscharakter

Benutzer des Pflichtenhefts



Aufbau des Pflichtenhefts nach IEEE Standard

- ▶ **Einleitung (introduction)**
 - ▶ Ziel (purpose)
 - ▶ Anwendungsbereich (scope)
 - ▶ Definitionen, Akronyme, Referenzen
- ▶ **Allgemeine Beschreibung (description)**
 - ▶ Produktperspektiven
 - ▶ Produktfunktionen
 - ▶ Benutzercharacteristika
 - ▶ Allgemeine Beschränkungen
 - ▶ Voraussetzungen und Abhängigkeiten
- ▶ **Spezifische Anforderungen (requirements)**
 - ▶ funktionale und nicht-funktionale Eigenschaften (umfangreichster Teil)
- ▶ **Anhänge**

Struktur eines Pflichtenhefts (nach Sommerville)

- ▶ Vorwort
- ▶ Einleitung
- ▶ Glossar
- ▶ Definition der Benutzeranforderungen
- ▶ Systemarchitektur
- ▶ Spezifikation der Systemanforderungen
- ▶ Systemmodell
- ▶ Systementwicklung
- ▶ Anhänge, Index

Evolution eines Pflichtenhefts

- ▶ **Pflichtenhefte sollten “lebendige Dokumente” sein**
 - ▶ Spagat zwischen PH als Vertrag und PH als Dokumentation des aktuellen Projektverständnisses
 - ▶ Zögern Sie nicht, das PH zu ändern wenn
 - ▶ Sich ihr Verständnis des Projekts geändert hat und das PH dies noch nicht reflektiert
 - ▶ Sie im Einverständnis mit dem Kunden Dinge ändern möchten
- ▶ **Dokumentieren Sie die Entwicklung des PH**
 - ▶ Wer hat was wann geschrieben?
 - ▶ Wieso wurde etwas geändert?
 - ▶ Kann manuell dokumentiert werden
 - ▶ Empfehlung: PH als Textdokument in die Versionsverwaltung aufnehmen

Pflichtenheft und “The Mytical Man-Month”



Fred Brooks: “Adding manpower to a late software project makes it later”

Grund: Kommunikationsaufwand wächst quadratisch zur Entwickleranzahl

Ein gutes Pflichtenheft kann helfen, den Kommunikationsaufwand wieder linear zu machen.

Sprache im Pflichtenheft

- ▶ Ein Pflichtenheft ist ein technisches Dokument
 - ▶ Fokus auf Präzision und Verständlichkeit
 - ▶ Keine langen, verschachtelten Sätze, keine unnötigen Fremdworte, keine überflüssigen Informationen
 - ▶ Sollte für alle Projektbeteiligten verständlich sein
 - ▶ Definition von wichtigen Worten im Glossar
- ▶ Vorsicht bei der Nutzung von Templates
 - ▶ Ist die Struktur des Templates wirklich adäquat für das Projekt?
 - ▶ Für jedes Projekt sollte die Struktur des PH an das Projekt angepasst werden

Diskussion Pflichtenheft

- ▶ Pflichtenheft kann sehr umfangreich und aufwendig werden...
- ▶ Vertragscharakter oder Richtlinie?
- ▶ Aufteilung in Muss-/Soll-/Kann-/Abgrenzungskriterien?
- ▶ Veraltet bevor fertig gestellt?
- ▶ Hilft inkrementelles Vorgehen?
- ▶ Kein übermäßiger Einsatz informeller grafischer Notation
 - ▶ UML, Use Case Diagramme etc.
 - ▶ Diese Diagramme dienen zur Illustration aber sind alleine nicht präzise genug

Zusammenfassung

- ▶ Anforderungsanalyse zentral für Projekterfolg
- ▶ Kunde und Entwickler müssen sich auf eine Anforderungsbeschreibung einigen. Hier treten viele Probleme auf.
- ▶ Es werden funktionale und nichtfunktionale Anforderungen an eine Anwendung unterschieden.
- ▶ Einen Überblick über die funktionalen Anforderungen geben Anwendungsfalldiagramme.

Diskussion: Foto-Editor

- ▶ Was wären denkbare Anforderungen und Anwendungsfälle bei einem Foto-Editor?