

- Motivation
- Gliederung des Themas, Begriffe
- Erheben von Anforderungen
- Analysieren und Dokumentieren von Anforderungen
  - Systemkontext
    - ▶ Use Case-Diagramme
  - Aktivitätsdiagramme
  - Use Case-Beschreibungen (Tabellen)
  - User Stories, User Story Mapping
    - ▶ Priorisierung
  - Snow Cards (für Qualitätsanforderungen / NFR)
- Überprüfen von Anforderungen

# Nächster Schritt

Für (fast) jeden Anwendungsfall aus dem *Use Case*-Diagramm beschreiben, wie er abläuft

Es gibt zwei Alternativen:

✓ Beschreibung als Aktivitätsdiagramm

2. Beschreibung in Tabellenform

Anwendungsbeispiel

- Mit seiner Kontonummer und seiner PIN kann ein Kunde auf sein Konto zugreifen: Er kann Beträge in verschiedenen Währungen einzahlen und abheben und er kann in verschiedenen Währungen den Kontostand abfragen.

# Dokumentation in Textform: Use Case vs. Szenario

Use Case	vs.	Szenario
Rolle		Name
Akteur-initiiert, Transaktion		beliebiger (Teil-)Ablauf
vollständige Aufgabe, mit Ausnahmen		genau ein Ablauf, keine Alternativen
Dialog Akteur / System		Geschichte
dient der Dokumentation		dient dem Verstehen und / oder der Analyse

# Beispiel: Use Case „Hebe Geld Ab“ (1)

Anwendungsfallname	Hebe Geld ab
Akteure	Bankkunde
Ziel	Geld abheben
Hauptablauf	<ol style="list-style-type: none"><li>1. Der Bankkunde meldet sich mit seiner Kontonummer und PIN am Geldautomaten an.</li><li>2. Das System überprüft die Identität.</li><li>3. Der Bankkunde gibt einen Betrag ein und wählt eine Währung aus.</li><li>4. Das System überprüft, ob die Währung vorrätig und der Betrag auf dem Konto verfügbar ist; es zeigt den Auszahlungsbetrag sowie die Belastung des Kontos an.</li><li>5. Der Bankkunde ...<ol style="list-style-type: none"><li>5a. ...bestätigt die Angaben.</li><li>6. Das System bucht den Betrag ab und zahlt ihn aus. Weiter mit 8</li><li>5b. ...bricht den Vorgang ab.</li><li>7. Das System bestätigt den Abbruch. Weiter mit 8.</li></ol></li><li>8. Der Bankkunde meldet sich ab.</li><li>9. Das System gibt eine Anmeldungsseite aus.</li></ol>

## Beispiel: Use Case „Hebe Geld Ab“ (2)

Ausnahme-abläufe	<p>Ausnahme in 2.: Kontonummer unbekannt <i>ODER</i> PIN inkorrekt          101. Das System gibt die Fehlermeldung „Kto-Nr. oder PIN falsch“ aus.          Weiter mit 9.</p> <p>Ausnahme in 4.: Währung nicht verfügbar          201. Das System gibt die Fehlermeldung „Währung nicht verfügbar“ aus.          Weiter mit 3.</p> <p>Ausnahme in 4.: Konto hat keine ausreichende Deckung          301. Das System gibt die Fehlermeldung „Betrag nicht verfügbar“ aus.          Weiter mit 3.</p> <p>Ausnahme in 3/5/8: längere Zeit keine Aktion durch Bankkunde (timeout=45s)          401. Das System meldet den Bankkunden ab.          Weiter mit 9.</p>
Anfangs-bedingungen	--
Abschluss-bedingungen	<p>Der Bankkunde hat den gewünschten Betrag erhalten und der Betrag ist vom Konto abgebucht <i>ODER</i>          der Bankkunde bekam eine Meldung, warum die Auszahlung nicht möglich war</p>
Qualitätsanforderungen	Ein Nutzer der häufiger am Automaten Geld abhebt soll von Beginn der Aktion bis zur erfolgreichen Auszahlung des Geldes nicht länger als 90s benötigen.

# Anleitung für Use Cases (1)

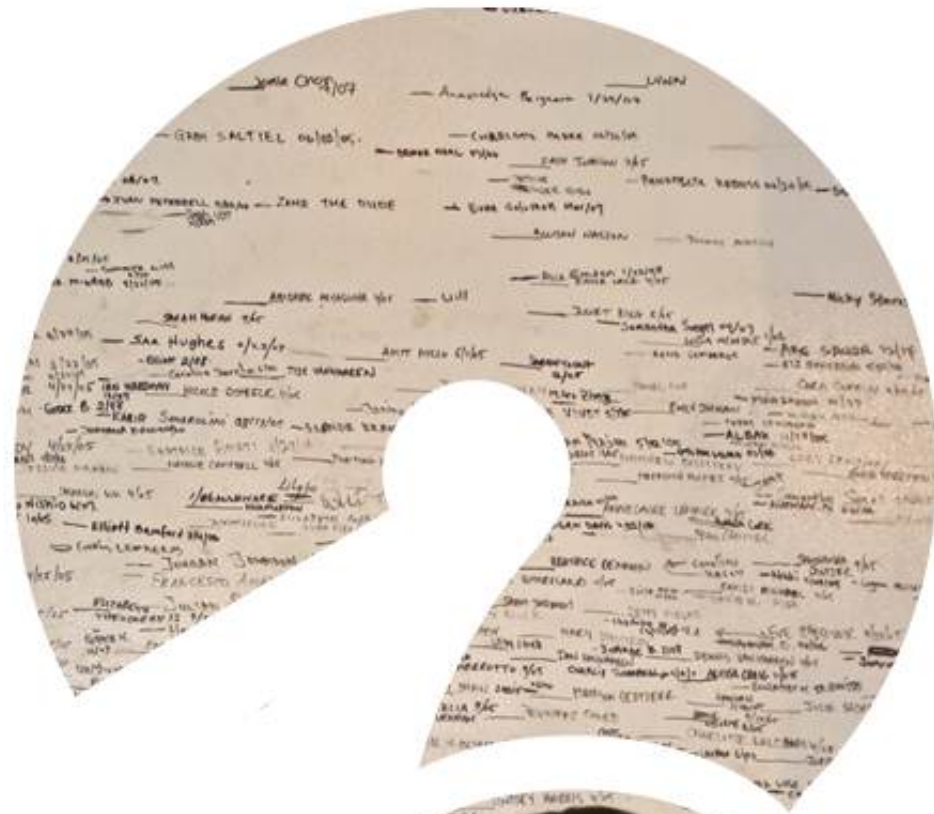
1. *Use Cases* sollen mit *Verbalphrasen* bezeichnet werden. Der Name sollte anzeigen, was der Anwender bewirken möchte.
  - z.B.           Melde Vorfall  
                  Hebe Geld ab
2. *Akteure* sollen mit *Nominalphrasen* benannt werden.
  - z.B.           Außenbeamter  
                  Bankkunde  
                  Zentraler DB-Server
3. Ein Use Case soll eine *vollständige Transaktion* eines Anwenders schildern.
  - z.B.           „Login“ ist *kein Use Case*
4. Ein Anwendungsfall sollte im Idealfall nicht länger als *eine Seite* sein.

## Anleitung für Use Cases (2)

### 5. Bzgl. dem Ereignisfluss:

- Die *Systemgrenzen* sollen klar sein.
  - ▶ Arbeitsschritte des Akteurs und Arbeitsschritte des Systems sollen eindeutig gekennzeichnet sein.  
z.B.    1. Der Bankkunde ....  
          2. Das System ...
- Die einzelnen Schritte im Ereignisfluss sollten im *Aktivstil* geschrieben sein, um deutlich zu machen, wer diese vollzieht
  - ▶ z.B.    3. Der Bankkunde gibt ... ein  
          4. Das System berechnet...
- Die *ursächliche Beziehung* zwischen aufeinander folgenden Schritten soll klar sein.

**Einrücken** der Systemarbeitsschritte



F R A G E N



photography: woodleywonderworks  
<http://www.flickr.com/photos/wwworks/2350106729>  
art work: Peter Kaiser

# User Story Mapping in 30 Seconds

## Idee (1)

- Gemeinsam mit Stakeholdern...
  - seine Ideen, Wünsche, Visionen
  - seine Geschäftsabläufe
  - seine Erwartungen an eine Software-Unterstützung erkunden / festlegen

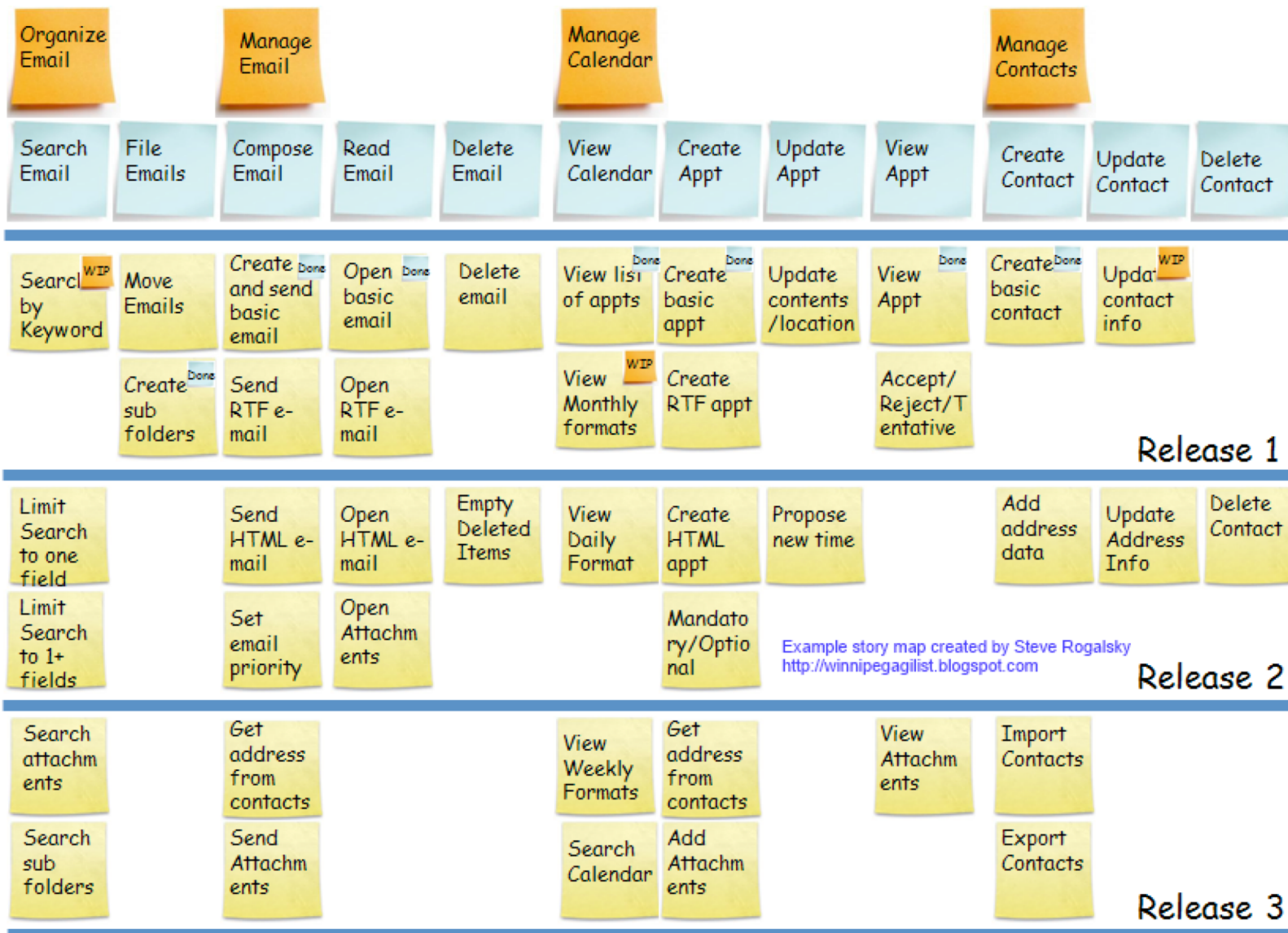
→ *Whole Story*
- Gemeinsam mit Stakeholdern...
  - die Schritte der Whole Story verfeinern und
  - in für Entwickler und Kunde gleich-verstandene Teilaufgaben zerlegen

→ *User Stories*
- Die beiden Tätigkeiten laufen verschränkt ab, nicht sequentiell

- Arbeiten mit Karten / Post-Its
  - Ein Schritt / eine Story pro Post-It
  - Ersetzen, falls
    - ▶ passendere Bezeichnung gefunden werden
    - ▶ Aufteilen / Zusammenfassen von Schritten / Stories erforderlich
  - „Kartentypen“ (verschiedene Farben)
    - ▶ Whole Story
    - ▶ User Stories
    - ▶ Kommentare, Personas, offene Punkte, ...
  - Einfaches Verschieben, um
    - ▶ Schritt / Story einzufügen / zu entfernen
    - ▶ Zu priorisieren
    - ▶ ....



# Priorisierung im User Story Mapping

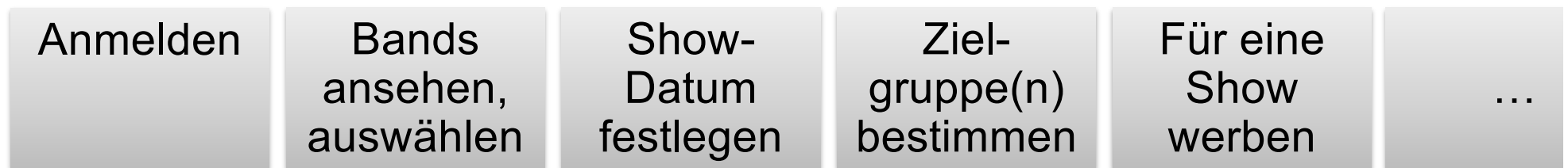


# The Whole Story

- Ablauf aus Anwendersicht beschreiben
  - Anwender formuliert mit!
- “Große” Schritte angeben
- Schritte bei Bedarf löschen / zusammenfügen / verfeinern...
- Hilfreich: Visionsszenarien erstellen / diskutieren

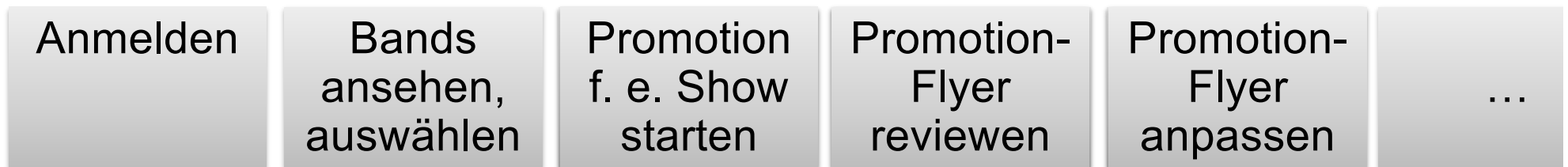
## Beispiel: Whole Story (1)

- Sie sind Auftragnehmer, Gary der Auftraggeber
- Gary möchte ein System, um Auftritte von Musik-Bands zu bewerben und zu managen
  - Schritte formuliert aus Garys Sicht



## Beispiel: Whole Story (2)

- „Für eine Show werben“ ist zu groß? → Aufsplitten



Die Whole Story wird angepasst, wenn dies notwendig ist. Dies ist notwendig, wenn z.B.

- ein Schritt sich als zu klein / zu groß erweist
- der Zuschnitt der Schritte nicht adäquat ist

Dies wird oft erst erkannt, wenn die Schritte mit User Stories detailliert werden

- User Stories sind **Geschichten**
  - Geschichten werden **erzählt**
  - User Stories sind die **Grundlage für die Diskussion** mit dem Kunden
  - Aufgeschriebene User Stories dienen dazu, Ergebnisse aus Diskussionen **merkbar** zu machen
- User Story Templates...
  - ... erleichtern das Schreiben der ersten User Stories
  - ... können / sollen angepasst werden
- Anreichern mit weiterer Informationen, z.B.
  - **Fragen** (sind vor einer Implementierung zu beantworten!)
  - **Personas**
  - **Ablaufkontrolle**, z.B. Auswahl („eine dieser User Stories“)

# User Story-Formulierung

- Template: “*As a role, I want goal / desire so that benefit*”

- Beispiel 1:

As a non-administrative user, I want to modify my own schedules but not the schedules of other users.

- Beispiel 2 (beide Stories gehören zu *einem* Whole Story-Schritt):

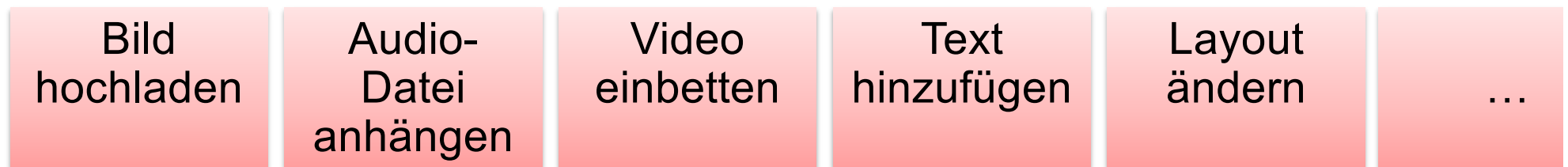
- As a consultant, I want to edit expense forms (including type, description, amount, comments) to get refunded.
- As a consultant, I want to be able to view and edit open expense forms in a list so that I can complete and submit or cancel them.

## Weitere (nicht zusammengehörige) Beispiele für Stories (*As a role, I want goal / desire so that benefit*)

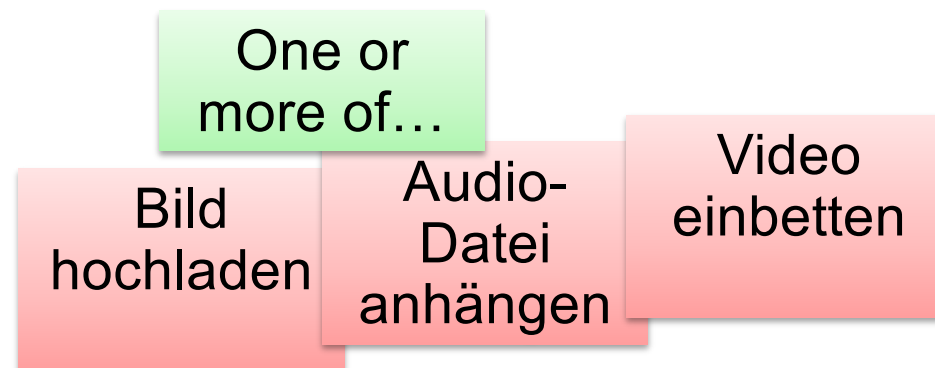
- Als Administrator möchte ich Zugriff auf ein Admin-Dashboard, um Informationen zu Basis-Einstellungen (Benutzer, Passwort-Qualität, Bearbeiten und Löschen von Profilen) zu bekommen.
- Als angemeldeter Benutzer möchte ich mein Profil ändern können, um die Information dort aktuell zu halten.
- Als Benutzer möchte ich die Webseite durchsuchen können, um Inhalte, die ich benötige, einfacher und schneller zu finden.
- Als Administrator möchte ich eine informativere Fehler-Anzeige für unsere Benutzer, damit sie mir weniger böse E-Mails schicken, wenn etwas nicht funktioniert.

# Beispiel: User Stories

- Verfeinerung des Schritts “Promotion-Flyer anpassen”



- User Stories können sequentiell (ohne besondere Markierung), oder – mit Kennzeichen - alternativ oder optional sein



# Gründe für die Priorisierung von Anforderungen

Ist nötig bei ...

- harten Terminen
- harten Preisobergrenzen
- Beschaffung
- Festlegen von Inhalt und Umfang der Inkremente bei inkrementeller Entwicklung
- Release-Planung bei der Weiterentwicklung bestehender Systeme

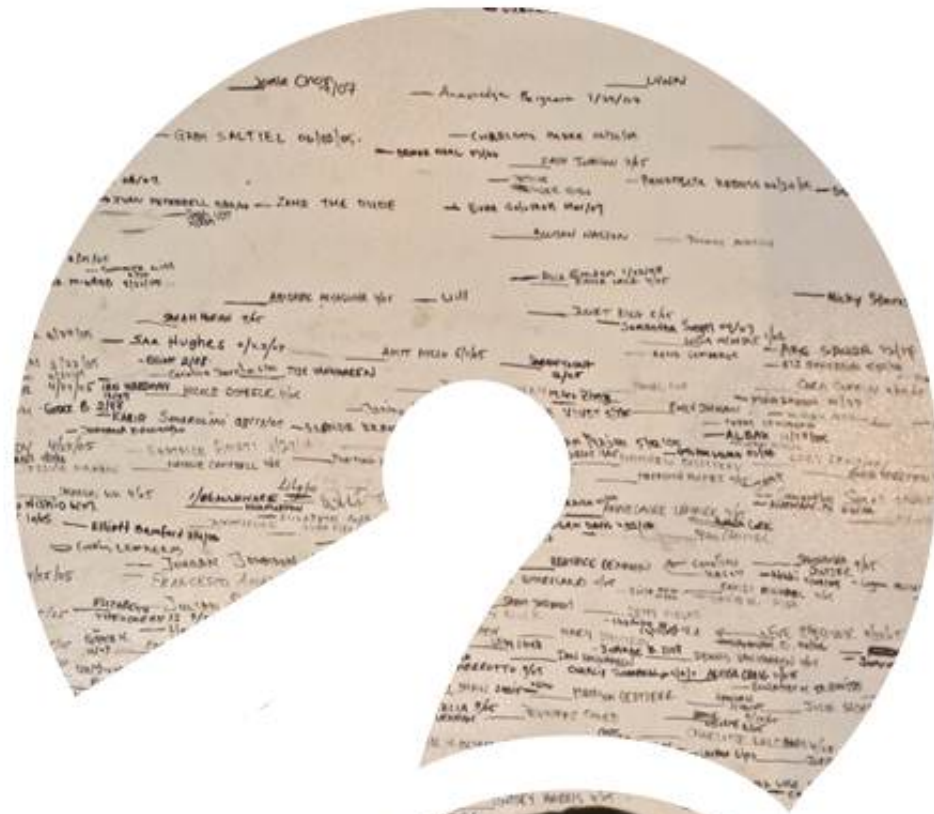
Zeitliche und finanzielle Restriktionen erlauben es meist nicht, alle Anforderungen mit der gleichen Intensität zu beachten.

# Priorisierung (Stufen nach IEEE 830-1998)

- Eine (sinnvolle) Kriterien-Klassifikation
  - **Mandatory, Muss**-Anforderungen
    - ▶ Anforderungen müssen umgesetzt werden, um den Erfolg zu gewährleisten
  - **Optional, Soll**-Anforderungen
    - ▶ Anforderungen, die zwar wichtig sind, deren vereinzelt Fehlen im System den Erfolg aber nicht gefährdet
  - **Nice-to-have, Kann**-Anforderungen
    - ▶ Anforderungen, deren Fehlen im System den Erfolg des Systems nicht gefährdet

Beobachtung: Meist starke Häufung in der Klasse „Mandatory“

Abhilfe: Stakeholder zum **Vergleichen** auffordern



# F R A G E N

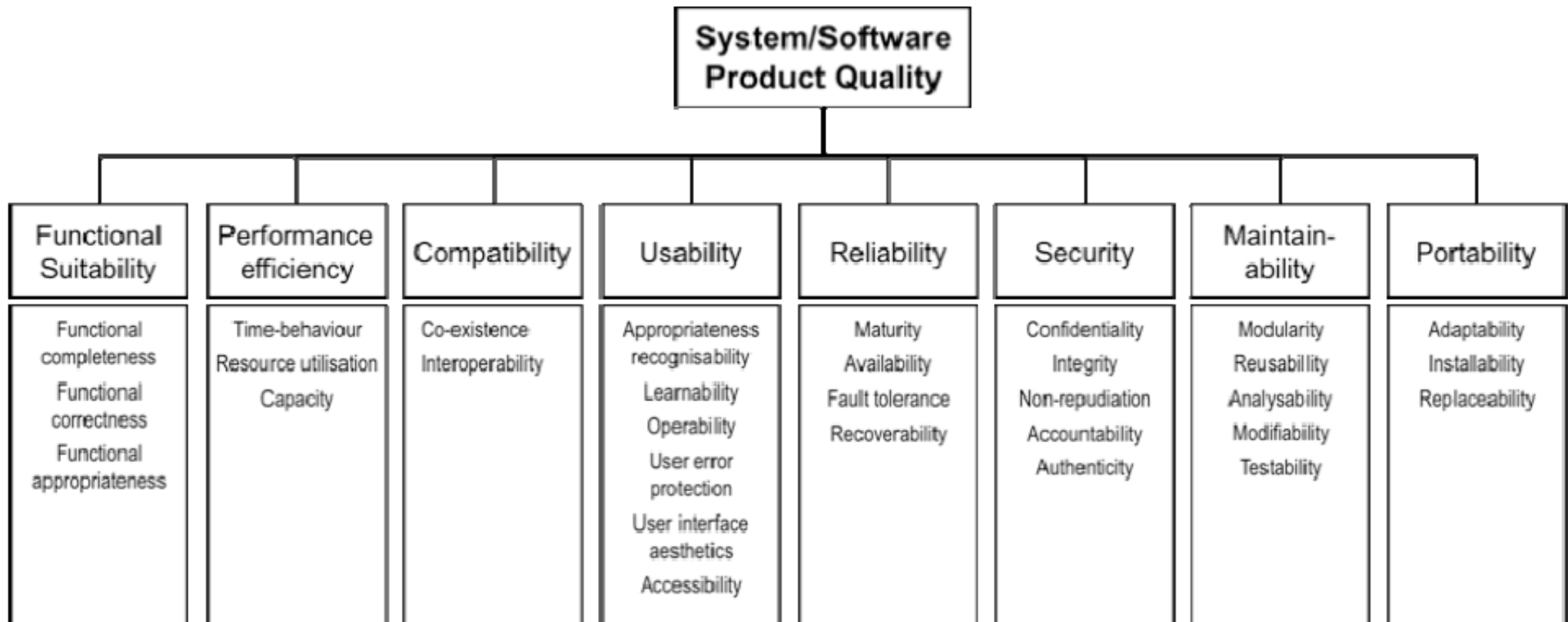


photography: woodleywonderworks  
<http://www.flickr.com/photos/wwworks/2350106729>  
art work: Peter Kaiser

# Beschreibung von Qualitätsanforderungen (NFR)

Typisches Vorgehen:

- Fragen stellen
  - Beispiel: Wie fehlertolerant soll das System sein?
- Orientierung zum Beispiel an der ISO 25010



# Beschreibung von Qualitätsanforderungen (NFR)

## Typisches Vorgehen:

- Fragen stellen
  - Beispiel: Wie fehlertolerant soll das System sein?
- Antworten *quantifizieren* mit Maßen und Abnahmebedingungen
  - direkte Maße:
    - ▶ Die Fehlertoleranz wird in MTTF gemessen und soll größer als 106 Betriebsstunden sein
  - indirekte Maße:
    - ▶ Die Bedienung des System gilt als erlernbar, wenn pro Person nicht mehr als zwei Tage Schulung aufgewendet werden müssen
    - ▶ Für jede Hauptfunktion beträgt der Lernaufwand für ihre erfolgreiche Anwendung im Mittel weniger als eine Stunde

# Regeln zur Formulierung von Qualitätsanforderungen

1. Formulieren Sie Qualitätsanforderungen **kurz und prägnant**
2. Verwenden Sie **Aktiv**formulierungen
3. Formulieren Sie **überprüfbare** Qualitätsanforderungen
4. **Verfeinern** Sie nicht überprüfbare Qualitätsanforderungen
5. Formulieren Sie den **Mehrwert** einer Qualitätsanforderungen
6. Geben Sie eine **Begründung** für die Qualitätsanforderungen an
7. **Vermeiden** Sie Lösungsansätze

## Beispiel: Welche Regel wird verletzt? Finden Sie eine bessere Formulierung!

1. Formulieren Sie Qualitätsanforderungen **kurz und prägnant**
2. Verwenden Sie **Aktiv**formulierungen
3. Formulieren Sie **überprüfbare** Qualitätsanforderungen
4. **Verfeinern** Sie nicht überprüfbare Qualitätsanforderungen
5. Formulieren Sie den **Mehrwert** einer Qualitätsanforderungen
6. Geben Sie eine **Begründung** für die Qualitätsanforderungen an
7. **Vermeiden** Sie Lösungsansätze

Das geplante System soll intuitiv benutzbar sein.

## Beispiel: Welche Regel wird verletzt? Finden Sie eine bessere Formulierung!

1. Formulieren Sie Qualitätsanforderungen **kurz und prägnant**
2. Verwenden Sie **Aktiv**formulierungen
3. Formulieren Sie **überprüfbare** Qualitätsanforderungen
4. **Verfeinern** Sie nicht überprüfbare Qualitätsanforderungen
5. Formulieren Sie den **Mehrwert** einer Qualitätsanforderungen
6. Geben Sie eine **Begründung** für die Qualitätsanforderungen an
7. **Vermeiden** Sie Lösungsansätze

Das geplante System soll besser sein als sein Vorgängersystem.

## Beispiel: Welche Regel wird verletzt? Finden Sie eine bessere Formulierung!

1. Formulieren Sie Qualitätsanforderungen **kurz und prägnant**
2. Verwenden Sie **Aktiv**formulierungen
3. Formulieren Sie **überprüfbare** Qualitätsanforderungen
4. **Verfeinern** Sie nicht überprüfbare Qualitätsanforderungen
5. Formulieren Sie den **Mehrwert** einer Qualitätsanforderungen
6. Geben Sie eine **Begründung** für die Qualitätsanforderungen an
7. **Vermeiden** Sie Lösungsansätze

Die Dauer für die Erstellung von Quartalsberichten, die meist, jedoch nicht immer, im 3-Monats-Rhythmus erstellt werden, soll im Vergleich zum Vorgängersystem halbiert werden.

## Beispiel: Welche Regel wird verletzt? Finden Sie eine bessere Formulierung!

1. Formulieren Sie Qualitätsanforderungen **kurz und prägnant**
2. Verwenden Sie **Aktiv**formulierungen
3. Formulieren Sie **überprüfbare** Qualitätsanforderungen
4. **Verfeinern** Sie nicht überprüfbare Qualitätsanforderungen
5. Formulieren Sie den **Mehrwert** einer Qualitätsanforderungen
6. Geben Sie eine **Begründung** für die Qualitätsanforderungen an
7. **Vermeiden** Sie Lösungsansätze

Durch kompakte Datenübertragung soll das geplante System um 10% kürzere Antwortzeiten aufweisen als sein Vorgängersystem.

# Beispiel: Snow Card

## Anforderungsnummer: 43

- Typ der Anforderung: Qualitätsanforderung
- Beschreibung: Das System ist von 10-jährigen Kindern zu bedienen
- Rational: Das System ist für Kinder vorgesehen, die die Grundschule absolviert haben
- Quelle: Karl-Heinz Müller
- Fit Kriterium: 80% der Testgruppe von 10-jährigen können sich innerhalb 2 Minuten erfolgreich registrieren
- Event/Use Case: UC 1/Registrierung
- Priorität: Hoch
- Konflikte: Keine
- Andere Materialien: keine

Roberston & Robertson, Mastering the Requirements Process, 2008

# Aufgaben der Anforderungsanalyse

## Anforderungen...

- ... erheben
- ... konsistent zusammenstellen, priorisieren
- ... dokumentieren
- ... überprüfen

*Elicitation*



*Analysis*



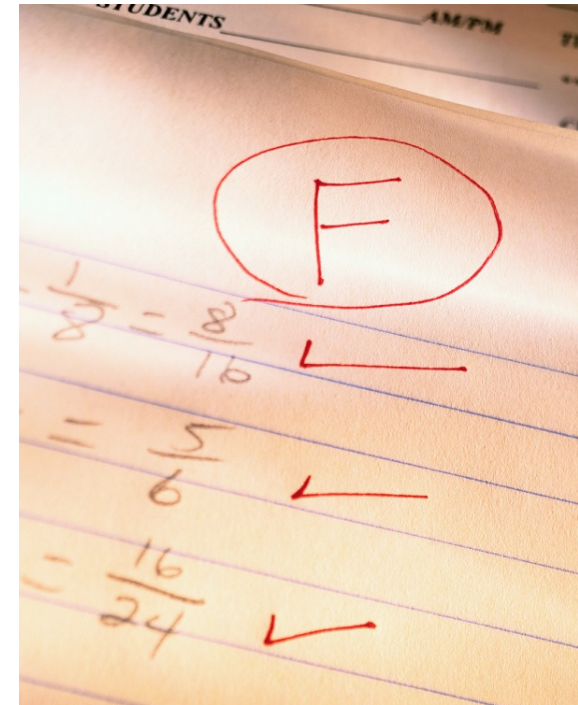
*Specification*



*Validation*

# Ziele der Qualitätssicherung von Anforderungen

- **Abweichungen** von der geforderten Qualität der Spezifikation feststellen
- Möglichst viele Fehler, Lücken, **Unklarheiten, Mehrdeutigkeiten** etc. finden und beheben
- **Konflikte** zwischen den Wünschen / Forderungen verschiedener beteiligter Personen erkennen und lösen
- **Verdeckte** Wünsche / Erwartungen / Befürchtungen erkennen und thematisieren



# Qualitätskriterien für Anforderungsartefakte

- eindeutig / verständlich / „klar“
- vollständig / „komplett“
- konsistent
- korrekt



- nachvollziehbar
- überprüfbar
- bewertet / abgestimmt / priorisiert
- modifizierbar / erweiterbar / verfolgbar

Clarity

Completeness

Consistency

Correctness

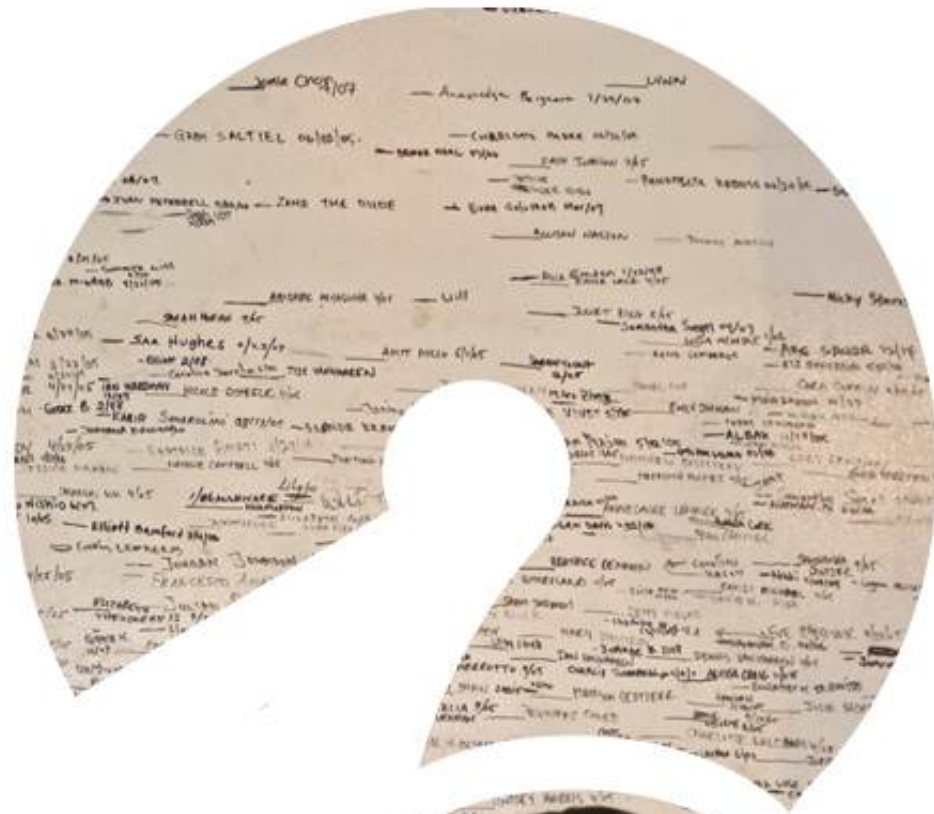
**4C**

Gilt für...

- einzelne Anforderungen
- das ganze Dokument!

# Anforderungsprüfung

- Einbeziehen aller Stakeholder abhängig vom Entwicklungsprozess, zum Beispiel:
  - Plangetriebener Prozess
    - ▶ Explizite Prüfung (Termin mit Kunden)
  - Agiler Prozess
    - ▶ Permanente Prüfung durch im Team mitarbeitenden Kunden
- Review
  - **Stellungnahme:** Expertise einer 3. Person
  - **Walkthrough:** Autor führt durch den Review (= Analyse der Dokumente)
  - **Inspektion:** Gutachter prüfen eigenständig, tragen in Sitzung Befunde zusammen
  - **Autor-Kritiker-Zyklus:** Kunde liest und kritisiert, bespricht Befunde mit Autor



# F R A G E N



photography: woodleywonderworks  
<http://www.flickr.com/photos/wwworks/2350106729>  
art work: Peter Kaiser