

# Workshop 7

Impulsreferat 4  
Requirements-Analyse und Spezifikation bei  
Individualsoftware

Mensch und Computer 2004  
Paderborn, 7.9.2004

Frieder Strauß

- **Software-Erstellung bei sd&m**
  - Anforderungsermittlung für Individualsoftware
  - Spezifikation der Benutzerschnittstelle

## Mein Erfahrungs-Kontext

- Seit 8 Jahren Mitarbeiter bei sd&m (900 Mitarbeiter)
- sd&m erstellt betriebliche Informationssysteme für die Kernprozesse unserer Kunden
- Individualentwicklungen im Kundenauftrag
- In der Regel datenbanklastige Verwaltungssysteme
  - Auftragsverwaltung, Buchungssysteme,
  - in Bereichen wie Banken, Touristik, Automotive
- Typische Projektgrößen: 4 bis 50 Bearbeiterjahre

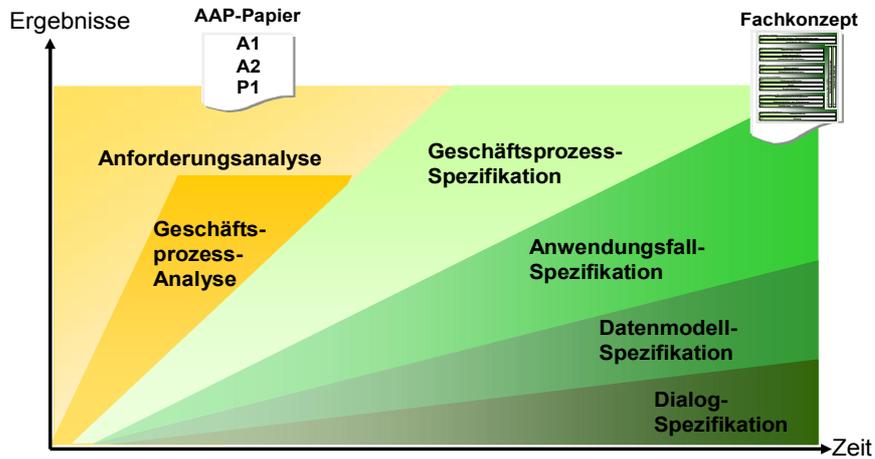
## Volere Snowcards

Eine typische Dokumentations-Technik für Anforderungen

- Jede Anforderung wird einzeln in einer Snowcard erfasst
- Inhalte einer Snowcard:
  - Anforderungsbeschreibung (1 Satz)
  - Begründung,
  - Messkriterium für die Umsetzung,
  - Klassifikation und
  - Anforderer

**Werden alle Anforderungen so erfasst, stehen bis zu Tausend Anforderungen nebeneinander.  
Die Konsistenz ist kaum noch sicherzustellen.**

## Detaillierte Anforderungsanalyse und Spezifikation überschneiden sich in unseren Individualsoftware Projekten



## Schwerpunkte der Teil-Phasen

	Anforderungsanalyse	Spezifikation
<b>Schwerpunkt der Fragestellung:</b>	Was muss die Lösung können?	Wie sieht die Lösung aus?
<b>Schwerpunkt der Tätigkeit:</b>	Anforderungen erheben	Anforderungen gestalten
<b>Eigenschaften:</b>	<ul style="list-style-type: none"> <li>■ definiert Systemgrenzen,</li> <li>■ Ermöglicht eine erste Aufwandsabschätzung und</li> <li>■ hilft, das Fachkonzept auf relevante Dinge einzugrenzen.</li> <li>■ Zu detaillierte Anforderungen sind inkonsistent</li> </ul>	<ul style="list-style-type: none"> <li>■ Macht Abhängigkeiten sichtbar,</li> <li>■ Auf Funktionalität Optimierte Dokumentationstechnik</li> <li>■ Zu früh zu detaillierte Spezifikationen sind häufig instabil</li> </ul>
<b>Dokumentationstechnik:</b>	Anforderungen, Ausgrenzungen, Prämissen	Geschäftsprozess, Anwendungsfall, Funktion, Datenmodell, Dialogbeschreibung

## TP-B Konfiguration - Zulieferung Konfigurator Anforderungen

Beispiel für Anforderungen

- Fachliche Anforderungen
  - Der Konfigurator unterstützt Mehrsprachigkeit, Mehrwährungsfähigkeit und den Wechsel von Preislisten.
- Technische Anforderungen
  - Der Konfigurator (MBOonline oder C2) wird als Java-Bibliothek zugeliefert und bietet die für den VKL II benötigte Funktionalität über seine Schnittstelle vollständig.
  - Der Konfigurator bezieht seine Daten aus einer relationalen Datenbank. Das Datenbankschema des Konfigurators wird zur Verfügung gestellt.
  - Zu Beginn des Projektes wird festgelegt, mit welcher Datenbank der Konfigurator betrieben werden soll.
  - Der Konfigurator benötigt über die Datenbank hinaus keine weitere Infrastruktur (z.B. Applikationsserver).

## TP-D Geschäfte – Zulieferung Fahrzeugfinanzierung/-leasing Prämissen und Ausgrenzungen

Beispiel für Prämissen + Ausgrenzungen

### Prämissen

- Die Berechnung von Leasing- oder Finanzierungsraten aus Preisen erfolgt nach einer von zwei einfachen Formeln (prozentualer Faktor oder lineare Berechnung aus Gesamtpreis und prozentualem Faktor).
- Die in winLEAS mögliche Berechnung von Angebots-Varianten wird im VKL nicht unterstützt. Für die Anzeige von Raten bei der Konfiguration im VKL gibt es jeweils genau eine gültige Variante (in winLEAS festgelegt).

### Ausgrenzung

- Über die Übergabe von Kundendaten beim winLEAS-Aufruf hinaus findet kein Abgleich von Kundendaten zwischen winLEAS und VKL statt.
- Die Frage, wie Kunden in zentralen Systemen der DC-Bank (re-)identifiziert werden, liegt nicht im Scope des VKL-Projekts.
- Der Vertrieb von Versicherungen wird nicht durch den VKL unterstützt.

## 1. These zur Anforderungs-Analyse für Individualsoftware

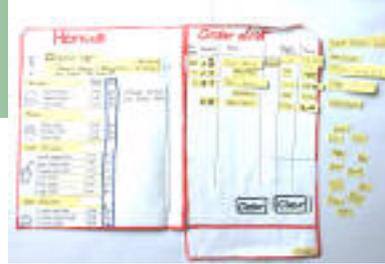
- Detaillierte Anforderungen können nur sinnvoll im Rahmen der Spezifikation ermittelt werden
  - Für konsistente Anforderungen wird eine Spezifikation benötigt, die die Anforderungen im Zusammenhang darstellt.
    - Datenmodell
    - Funktionen
    - (Geschäftsprozess-) Abläufe
  - Die üblichen Anforderungsdokumentations-Techniken sind kein geeignetes Medium um detaillierte Anforderungen zu beschreiben.

## 1. These zur Anforderungs-Analyse für Individualsoftware

### Begründungen:

1. Bei detaillierten Anforderungen ist die Beschreibung der Anforderung und die Beschreibung einer möglichen Lösung häufig schlecht zu trennen und im Aufwand ähnlich groß.
2. Die Anforderungen einer Fachabteilung haben häufig Auswirkungen auf die Arbeit anderer Abteilungen oder führen zu Designkonflikten
  - Bei der Beschreibung einer Spezifikation können die Auswirkungen besser beurteilt werden (vgl. auch ISO 13407)
3. Bei einer Individualentwicklung steht der zukünftige Benutzer/Fachbereich ausreichend zur Verfügung und die zu unterstützenden Aufgaben 'stehen' fest.

## Paper Mockups



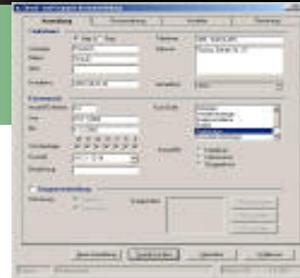
- Paper Mockups basieren auf Nutzung von Papier, Ausdrucken, Post-Ist, Schere usw.
- Eigenschaften:
  - Dialog ist viel schneller änderbar als mit DV-Unterstützung
  - Paper Mockups erlauben schnelle Iterationen des Grob-Designs
  - Low-Fidelity Ansatz unterstützt Konzentration auf das Wesentliche
- Nutzung bei sd&m:
  - Erst mit Vorab-Schulung wird die Technik auch eingesetzt.
  - Hierzu reicht allerdings eine praktische Einführung von 1 Tag.

## User Boards evaluieren Dialogentwürfe

- Walkthrough zur Prüfung von Bedienbarkeit und Funktionalität anhand (ausgedruckter) Dialog-Szenarien
- Eigenschaften:
  - Einfach mit Fachbereich durchzuführen
  - Nur die wichtigsten / kritischen Use-Szenarien sollten durchgespielt werden
  - Ist auch ohne „Usability-Experten“ gut durchführbar
  - Nicht geeignet um detaillierte Darstellungsprobleme bzw. schlechte Benennungen zu finden
- Nutzung bei sd&m:
  - Intensiver Nutzung erst parallel mit Mockup-Technik [wg. Schulung?]

## GUI-Prototypen

- GUI-Prototyp umfasst zumindest die statische Präsentation einzelner Fenster
- Eigenschaften:
  - Ermöglicht detailliertes Layout datenlastiger Oberflächen
  - Durch Nutzung üblicher GUI-Builder stehen alle üblichen Widgets zur Verfügung
  - Dynamik kann exemplarisch programmiert werden.
- Risiken:
  - Sparziel: Wiederverwendung von Code
  - Statt zu spezifizieren werden Details programmiert
  - Aufwändige Prototyperstellung für Reviewzwecke / zur QS
  - Bei Nutzung von Malprogrammen (Powerpoint, Visio...) geht Gefühl für verfügbare „Pixel“ verloren.



## 2. These Anforderungen an die Benutzerschnittstelle können nicht abstrakt ohne eine Lösung diskutiert werden

- Ein Design für die Benutzerschnittstelle kann iterativ erstellt und auf Erfüllung geprüft werden.
- Gute Techniken sind:
  - Paper Mockups für frühe Phasen (**häufig nur im Design-Team**)
  - GUI-Prototypen **für späte Spezifikationsphase**
  - User-Boards für die Beteiligung des Fachbereichs/der Benutzer
- Vergleiche hierzu auch z.B. ISO 13407, die einen Zyklus von
  - Anforderungsermittlung,
  - Spezifikation/Prototyping und
  - Evaluation vorgibt.
- Bei sd&m sind die 'Anforderungen' in der Regel die zu unterstützenden Geschäftsprozess-Schritte bzw. Anwendungsfälle der Fachabteilungen.

## Zusammenfassung

- Thesen zur Anforderungsermittlung:
  - Detaillierte Anforderungen sollten in einem Modell beschrieben werden, anstatt sie abstrakt als Anforderung zu dokumentieren.
  - Nicht-Funktionale Anforderungen insb. zur GUI lassen sich nur sinnvoll in einem Dialogmodell beschreiben.
  - Iteratives Design durch Mockups (sowie GUI-Prototyp) und Evaluation durch User-Boards
- Diese Techniken etablieren sich im Projektalltag aber nur durch entsprechende Schulungen.
- Die Thesen sind dann gültig, wenn
  1. Die späteren Benutzer (und damit die zu unterstützenden Aufgaben) ausreichend eingebunden werden können
  2. Die Software noch „beliebig“ gestaltet werden kann  
(beim Einsatz von vorhandener Standard-Software ist dies ggf problematisch)