

# Contextual Design

## Teil IV: Neugestaltung der Arbeit

Jörg Cassens

Medieninformatik II  
Contextual Design of Interactive Systems  
SoSe 2016

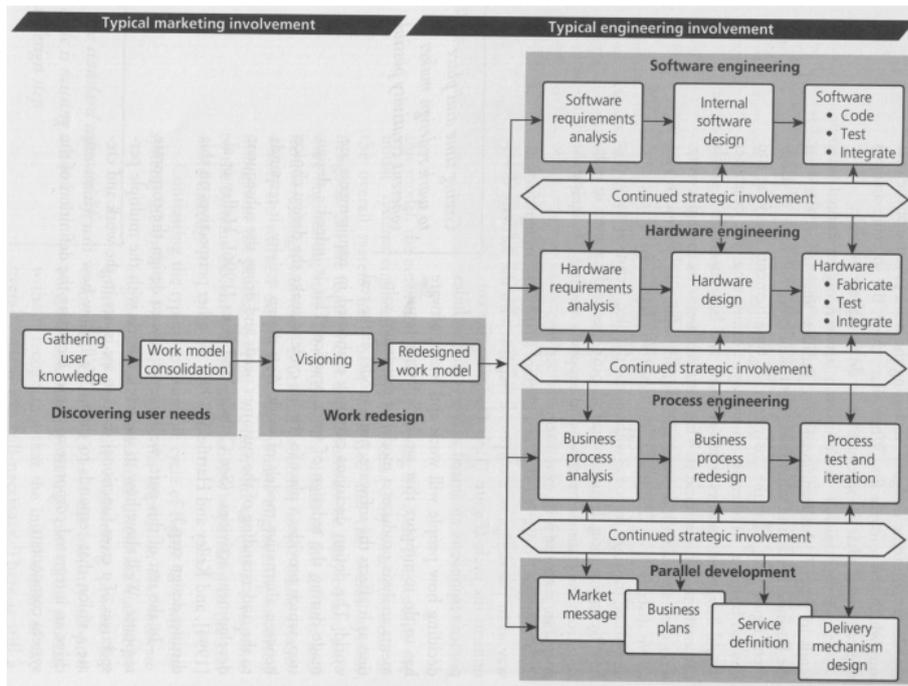


## 1 Überblick

### Schritte

- Phase I: Analyse
  1. Kontext-Erkundung
  2. Interpretationssitzung & Arbeitsmodellierung
  3. Konsolidierung
- Phase II: Gestaltung
  1. Neugestaltung der Arbeit
    - Darstellung des neuen Arbeitsalltags durch eine abstrakte Geschichte (Vision) und mit Detailgeschichten einzelner Arbeitsaufgaben (Storyboards)
  2. Design der Benutzerumgebung
    - Struktur und Funktion des neuen Systems unabhängig von Benutzungsschnittstelle und Implementierung
  3. Paper Prototyping
    - Mit einer Papierattrappe der Benutzungsschnittstelle versuchen die Benutzer "echte" Arbeitsaufgaben zu erfüllen

### Einordnung



## Innovation

- Benutzerdaten plus technische Entwicklung treiben Innovation an, nicht der geniale Erfinder im stillen Kämmerlein
  - VisiCalc – erste Tabellenkalkulation, entstanden als der Entwickler Dan Bricklin einen Buchhaltungskurs machte
  - WordPerfect entstand während die Entwickler Alan Ashton und Bruce Bastian im Stockwerk unter dem Sekretariat arbeiteten
  - Keine Innovation ist ganz unabhängig von existierenden Strukturen und Vorgehensweisen
  - Innovation ist zumeist eher kontinuierliche Evolution denn Revolution
  - *Design: Innovation, die ein Team als Antwort auf Daten gibt*
- Wir suchen einen Prozeß, der diese Art Innovation fördert

## 2 Neugestaltung

Was tun?



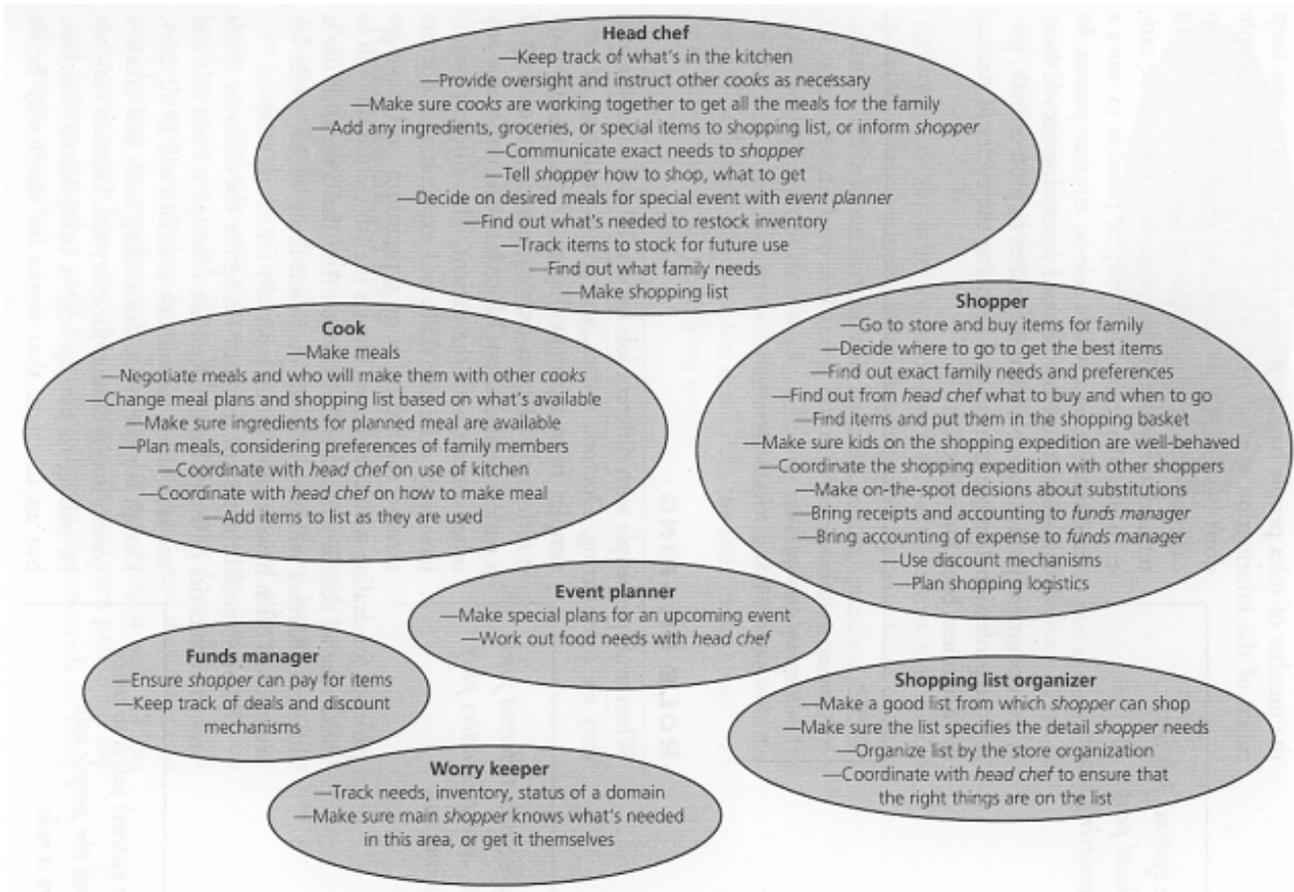
Was machen wir nun mit den vielen Modellen?

## Vorgehen

1. Analyse der konsolidierten Modelle
  - neuer Fokus der Analyse (gestalten/verändern statt verstehen)
2. Entwicklung neuer Arbeitsstrukturen
3. Darstellung des neuen Arbeitsalltags durch eine abstrakte Geschichte (Vision) und mit Detailgeschichten einzelner Arbeitsaufgaben (Storyboards)

## 2.1 Fluß

### Flußmodell



### Rollentausch

- Problem Rollentausch
  - Jeder spielt mehr als eine Rolle
  - Der Übergang von einer Rolle zu einer anderen verursacht Wechselkosten und verursacht die Gefahr, daß etwas "wegrutscht"
- Folgerung: *Das System sollte den Kontext der alten Rolle wiederherstellen können*
- Hinweise:
  - Eliminiere redundante Eingabe von Daten
  - Unterstütze den Wechsel zwischen Rollen
  - Konsistente Interfaces für unterschiedliche Rollen
  - Speichere den Status um Unterbrechungen ausgleichen zu können

## Überlastung

- Problem Überlastung
  - Haben Menschen zu viele Rollen werden sie leicht überlastet
  - Siehe einleitendes Beispiel: Ein privater Haushalt erfordert bereits viele Rollen
- Folgerung: *Das System sollte die Anzahl der Rollen minimieren und verbleibende unterstützen*
- Hinweise:
  - Automatisiere oder eliminiere Rollen
  - Unterstütze und organisiere Rollen
  - Verschiebe Verantwortlichkeiten oder Rollen zu anderen Personen

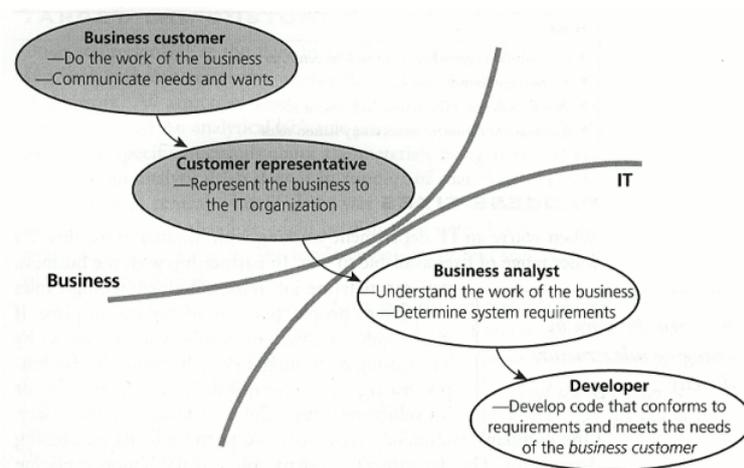
## Rollenteilung

- Problem Rollenteilung
  - Leute mit unterschiedlichen Aufgaben, Fähigkeiten und Toleranzen füllen die gleiche Rolle aus
  - All diese Leute müssen vom System unterstützt werden
- Folgerung: *Design für den anspruchsvollsten Benutzer*
- Hinweise:
  - Passe die Interaktionsmöglichkeiten an die Benutzer an
  - Passe das Informationsdesign an die Benutzer an
  - Teile die Daten intern zwischen den Benutzern, auch wenn sie unterschiedliche System benutzen
  - Passe das System an die anderen Rollen der Benutzer an (Konsistenz)

## Isolierung

- Problem Isolierung
  - Isolierung ist verursacht durch Aufteilung von Rollen auf verschiedene Personen
- Folgerung: *System muß den Kontext mitliefern*
- Hinweise:
  - Kommunizieren den gesamten Kontext zwischen Rollen
  - Unterstütze die Kommunikation zwischen Rollen
  - Präsentiere nur die Informationen die eine Rolle braucht
  - Automatisiere oder eliminiere unnötige Liaison-Rollen

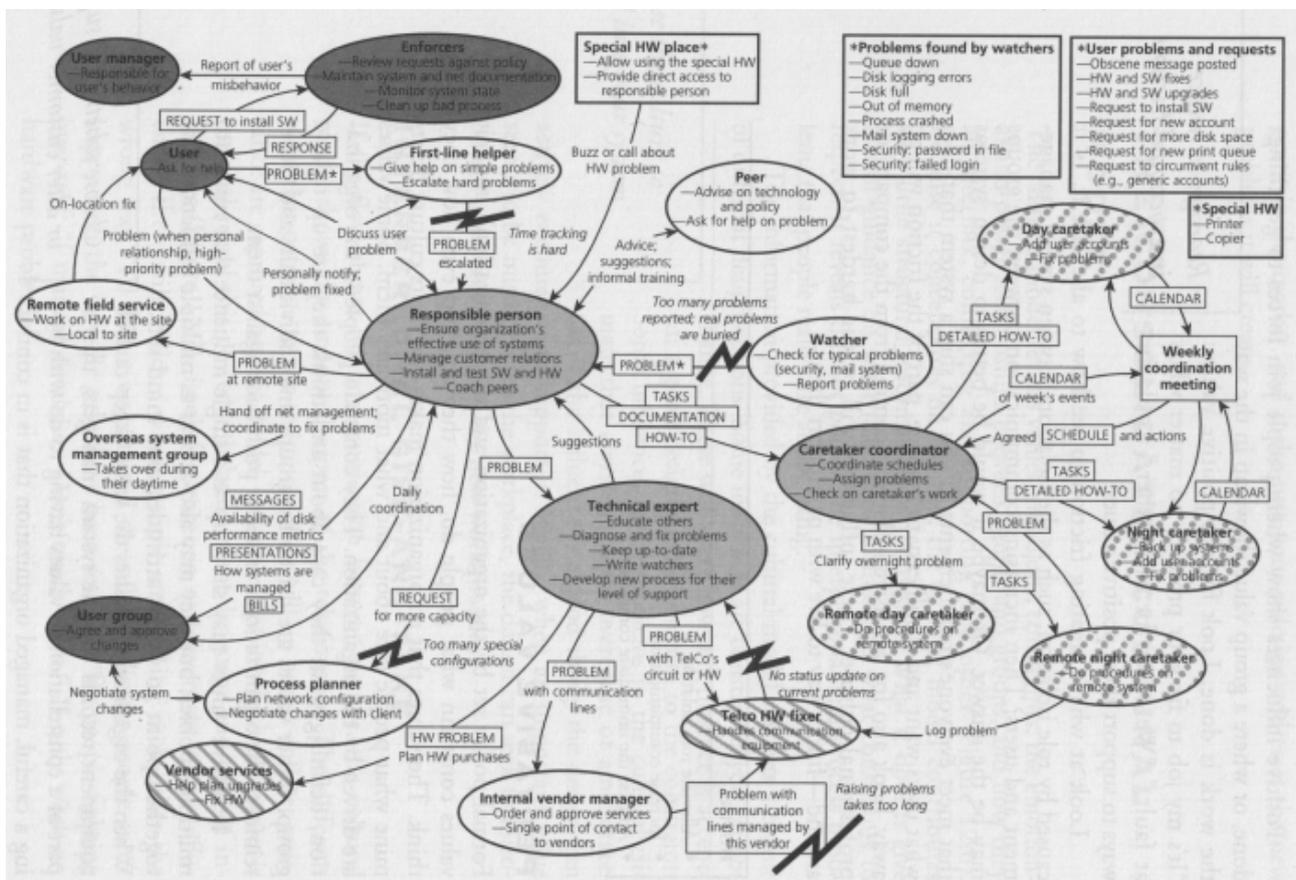
## Rollen: Liaison



## Prozeßmodifikation

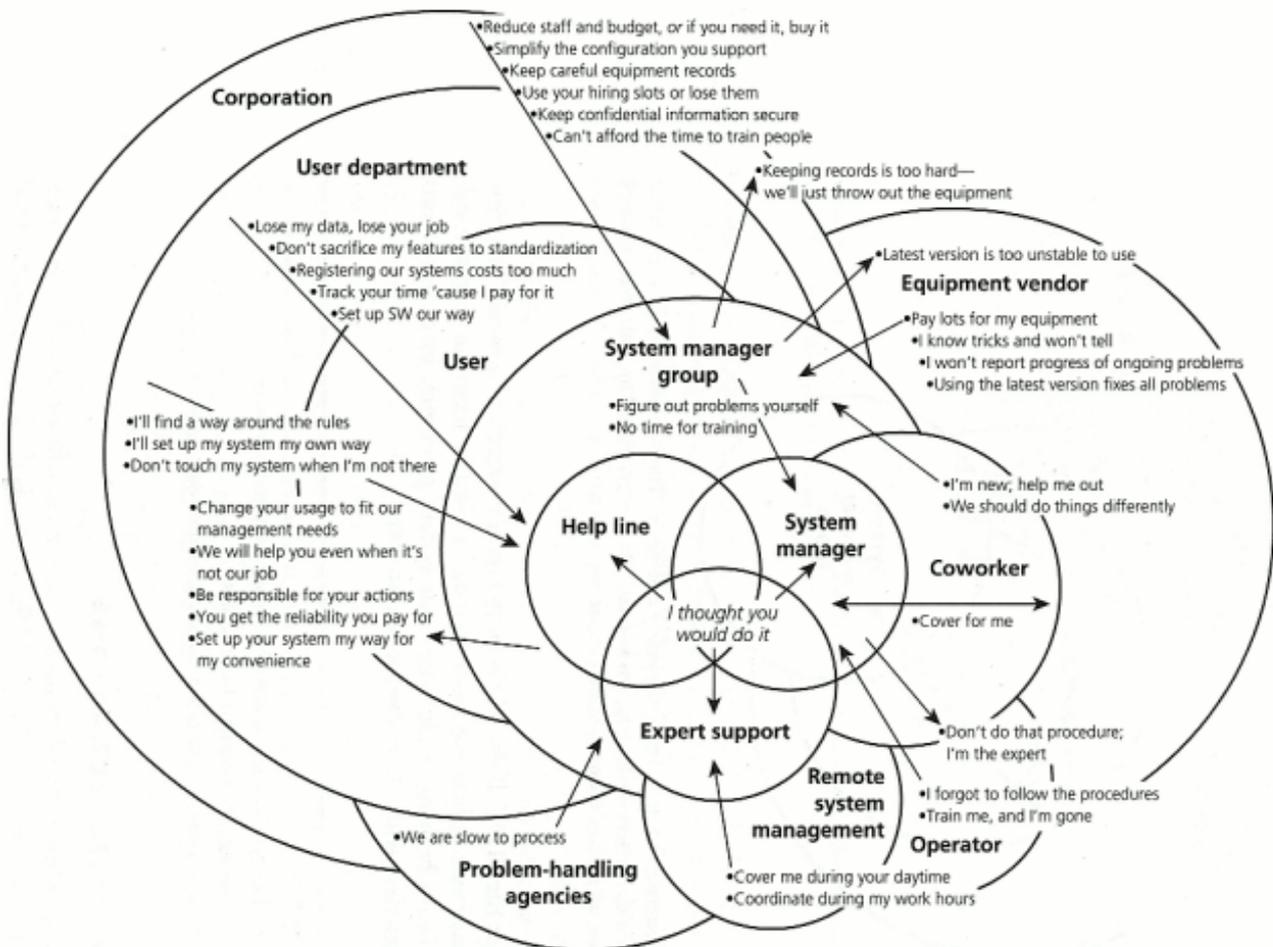
- Verändere die Arbeit durch direkte Änderung der Rollenstruktur
- Business Process Reengineering (BPR)
- Hinweise:
  - Designe die Organisation als Teil des Designs der Arbeit
  - Denke über Änderungen an Prozessen und Prozeduren nach
  - Denke über neue Rollen und Verantwortlichkeiten nach
  - Habe BPR-Leute im Team
- Probleme:
  - Beachte die Organisationskultur
  - Schaffen Deine Änderungen neue Probleme?

## Beispiel: Flußmodell



## 2.2 Einfluß

### Einflußmodell



## Gegenstand

- Information über Werte und Einstellungen → könnte Arbeit mit neuer Technologie verhindern
  - Wo könnten Irritationen auftreten?
  - Wo könnte es Kommunikationsprobleme geben?
  - Wo gibt es grundlegende Wertvorstellungen?
  - Wie verändert die vorgeschlagene Technologie die Verhältnisse zwischen den Beteiligten (z.B. Machtverhältnisse)?
- Beispiele:
  - Banken – Schalterdienst vs Online-Banking
  - Einführung der elektronischen Satzsysteme bei norwegischen Zeitungen

## Ziele

- Reduziere Reibungen zwischen Personen
  - Vermindere Rollenisolierung
  - Erhöhe Kommunikation
  - Adressiere die direkten Störungen
- Werte
  - Vereinfache es, positive Werte und Constraints zu erfüllen
  - Mache es schwieriger, negative Einflüsse zu erzielen
  - Schaffe positive Gegengewichte zu negativen Werten
- Schwierigkeiten
  - Versuche nicht den Kunden in eine Richtung zu drängen die er nicht möchte



## **Bewegung**

- Bewegung und Zugriff
- Separate Orte reflektieren eine Aufteilung der Arbeit – respektiere das
  - Sind Besprechungsräume nah am Eingang oder zwischen den Büros?
  - Gibt es Teeküchen, Raucherecken?
- Hinweise:
  - Treffe oder verbessere den Fluß von Artefakten
  - Halte konzeptuelle Aufteilungen aufrecht
  - Unterstütze die impliziten Intentionen in der Anordnung von Orten

## **Automatisierung**

- Work with paper – it's not going away
- Müssen Dokumente aufgrund von Regularien ausgedruckt werden?
- Hinweise:
  - Befasse Dich mit den Intentionen des Papier-Systems
  - Bilde dieses möglichst komplett im Online-System ab
  - Hilfe dabei Papier und Daten im Einklang zu halten, soweit Papier noch notwendig ist

## **Ziele**

- Aussagen: z.B. viel benutzte Artefakte in Reichweite, selten benutzte hinten im Schrank
- Wie unterstützt oder behindert die Umgebung die Arbeit?
- Ist die Übertragung eines Elements der Arbeitsumgebung, das die Arbeit bereits gut unterstützt, auf den Computer wirklich sinnvoll?
  - z.B.: Multisektorkoordination in der Flugsicherung
- Welche Wirkung haben die vorgeschlagenen Änderungen auf die physische Umgebung?
- Welche Unterbrechungen/Behinderungen können in dem neuen System behoben werden?
- Beispiel: Handgeschriebene Zettel gehen auf einem belegten Schreibtisch verloren

## **2.4 Ablauf**

### **Ablaufmodell**

FIXING A PROBLEM		
Set up to tackle problem	• Set up place and context to tackle problem	Go to the place where problem can be solved
	• Orient self to problem situation	Get more information on problem
	• Anticipate need for help	Look at system to see problem in context
Search for cause	• Identify cause of problem	Search for problem cause by hand
	• Eliminate repetitive tasks	Create and run specialized procedure to search for cause of problem
		Identify cause of problem
Fix problem	• Minimize disruption of users' work	If a disk problem:
		See who will be affected by work on disk
		Warn users of work to be done
		Wait for users to get off disk
		Dismount disk
	Find scratch disk or new disk	
• Move substitute HW into place so users can keep working	Mount new disk for use in fix	

## Intentionen

- Ein System muß alle Intentionen des Benutzers unterstützen
- Das System sollte die Intentionen leichter erreichbar machen
- Teilziele können automatisiert werden (wenn das System Graphiken unterschiedlicher Auflösung anbietet muß nicht mehr kopiert werden)
- Hinweise:
  - Mache die primären Intentionen irrelevant
  - Unterstütze primäre Intentionen auf neue Art und Weise
  - Denke an alle sekundären Intentionen
  - Re-Designe so, daß sekundäre Intentionen erreichbar sind

## Herangehensweisen

- Unterstütze bekannte Strategien
- Wenn Leute an Problemen arbeiten, auch nachdem sie Hilfe geholt haben, hat das seinen Zweck in der Organisation
- Hinweise:
  - Identifiziere und unterstütze Problemlöse-Strategien
  - Mache Strategien nicht überflüssig bevor Du verstanden hast, wann und warum sie eingesetzt werden

## Auslöser

- Gestalte Auslöser so, daß sie Bedürfnisse der Benutzer unterstützen
- Dupliziere vorhandene Auslöser oder ersetze sie durch solche die besser sind – Mails von einem Überwachungssystem oder direkte Notifikationen?
- Hinweise:
  - Biete Auslöser für Arbeitsaufgaben an
  - Passe die Auslöser an den Benutzer an und liefere die "passende" Art von Auslösern

## Ziele

- Zentrale Analyse: Kann die Aktivität automatisiert werden?
- Wenn nicht:
  - Kann die Aktivität durch Teilautomatisierung effizienter werden?
  - Sind alle derzeitigen Schritte nötig?
  - Alle unnötigen Schritte eliminieren!
- Maxime:
  - Neue Variante sollte nicht mehr Arbeit machen

## 2.5 Artefakte

### Artefaktmodell

**Company Name**

Department sent to; **To:** \_\_\_\_\_ Done; date fulfilled; order number

person assigned; **From:** \_\_\_\_\_ Filled in

date **Date:** \_\_\_\_\_

**Subject:** Type of request  
High-level description

**Remarks:** \_\_\_\_\_

**Uses**

- Make SW request
- Make HW request
- Motivate quick and specified action
- Facilitate communication between departments
- Record information about exactly what to order
- Track status
- Track and trace assignees
- Record what was purchased
- Remind what to purchase

**Rationale** for purchase to motivate ("dire need") or justify based on impact on work

**Description** of what is needed not vendor or model numbers assumes knowledge of what already exists and standard configurations

If hardware: **Location** of where to put new equipment

**Rationale** for location

**Supplier** to buy it from

**Estimated costs:**

Item	Cost
_____	_____
_____	_____
<b>Total</b>	_____

**RECEIVED DATE DEPARTMENT**

**Insights**

- Can take up to one year to complete
- Structure is common even when not required
- Dates provide a history of when ordered, approved, and filled

## Hintergrund

- Ein elektronisches Artefakt kann informellen Gebrauch unmöglich machen
  - Sind auf Laufzetteln vielleicht noch Arbeitsanweisungen notiert?
- Hinweise:
  - Unterstütze die primären Intentionen direkter
  - Unterstütze auch die Intentionen die durch informelle Nutzung angezeigt werden
  - Unterstütze alle Intentionen :)

## Aussagen

- Artefakte präsentieren Informationen
  - Bestellzettel ohne Kostenfeld deuten auf wenig Kostenbewußtsein hin
  - Wer braucht welche Informationen? Kommuniziert das Artefakt Informationen? Sind es die selben Informationen für alle Rollen?

- Koordiniert das Artefakt?
- Sie sollten nur die Informationen liefern welche die Leute auch brauchen
- Hinweise:
  - Liefere Daten automatisch
  - Teile Daten zwischen Rollen
  - Unterstütze die Kommunikation die durch die Artefakte impliziert wird

### Aussehen

- Unterteilung und Aussehen
  - Die Struktur eines Artefaktes zeigt Unterscheidungen im System auf
  - Die Präsentation ist wichtig: sie leitet das Auge und hebt Teile hervor
- Hinweise:
  - Benutze die Hauptstruktur des Artefaktes um die Struktur des Systems anzuleiten
  - Behalte Unterscheidungen die wichtig sind für den Benutzer
  - Finde die Intentionen hinter den Details der Präsentation heraus
  - Imitiere die Intention der Präsentation, nicht deren Details

### Ziele

- Zentrale Analysen:
  - Wo funktionieren derzeitige Artefakte nicht so gut?
  - Wie könnte ein Artefakt auf den Rechner gebracht werden?
    - \* sind die zentralen Aspekte des physischen Artefakts auf ein computerbasiertes Artefakt übertragbar?
  - Was gewinnt man dabei ein Artefakt auf den Rechner zu bringen und was verliert man?
    - \* wie findet informelle Kommunikation statt?
    - \* können Elemente automatisch ausgefüllt werden?
- Beispiel:
  - Hotelzimmerschlüssel vs. programmierbare Plastikkarten
    - \* Was sind die Vor- und Nachteile?
  - Elektronische Patientenakte

## 3 Vision

### Details und Überblick

- Vision zeigt Big Picture
- Details werden im Contextual Design beschrieben durch
  - Storyboards
  - Benutzerumgebung
  - Paper Prototypes
- Neues erfinden aus Benutzerdaten?

## Überblick

- Zwei Faktoren:
  - *“Invention is a response to some life or work practice by a designer or technologist who, seeing the need and knowing the technology, imagines a new possibility”*  
(Karen Holtzblatt)
- Ideen generieren:
  - es gibt kein Patentrezept für gute Ideen
  - Ansatz von Beyer/Holtzblatt: intensive Teamarbeit
    - \* Gruppenkreativitätstechniken (Brainstorming etc. )
  - Benyon, Turner und Turner
    - \* Designing Interactive Systems: Modifikation des Ansatzes für Einerteams

## Vision

- abstrakte Sicht auf den neuen Arbeitsalltag
- zeigt, wie manuelle Arbeitstätigkeit, menschliche Interaktionen und bestehende Werkzeuge zusammen mit dem neuen Produkt den Arbeitsalltag unterstützen
- stellt sicher, daß die neue Technologie zum Arbeitskontext passt
- Fokus auf dem sozio-technischen System, nicht auf Details

## 3.1 Visionssitzung

### Ablauf

- *“Walk the data:”* alle Modelle durchgehen, bis man wirklich *“drinsteckt”*
  - Affinitätsdiagramme
  - Konsolidierte Modelle
- Ideen vergleichen und eine gemeinsame Idee bekommen
- *“Priming the brain:”* Auflisten von Möglichkeiten, Brainstorming
  - Technologie
  - Startpunkte
- *“Creating a vision:”* einen Startpunkt ausmachen, anfangen, gemeinsam eine Vision zu zeichnen
  - typische Ausgangspunkte für Designideen: Scheitern oder Probleme des bestehenden Systems

### Rollen

- Zeichner:
  - malt die Vision nach den Vorschlägen der Teammitglieder
  - falls eine Idee nicht in die Vision paßt, schreibt er sie in die *“Hot Ideas”*-Liste
  - trägt nicht zur Geschichte bei
- Moderator:
  - behält den Fokus im Auge
  - erinnert Teilnehmer an fehlende Aspekte
  - achtet darauf, daß jeder zu Wort kommt
  - trägt zur Geschichte bei
- Teilnehmer:
  - Vorschlagen von Designlösungen und Benutzeraktionen

## Regeln

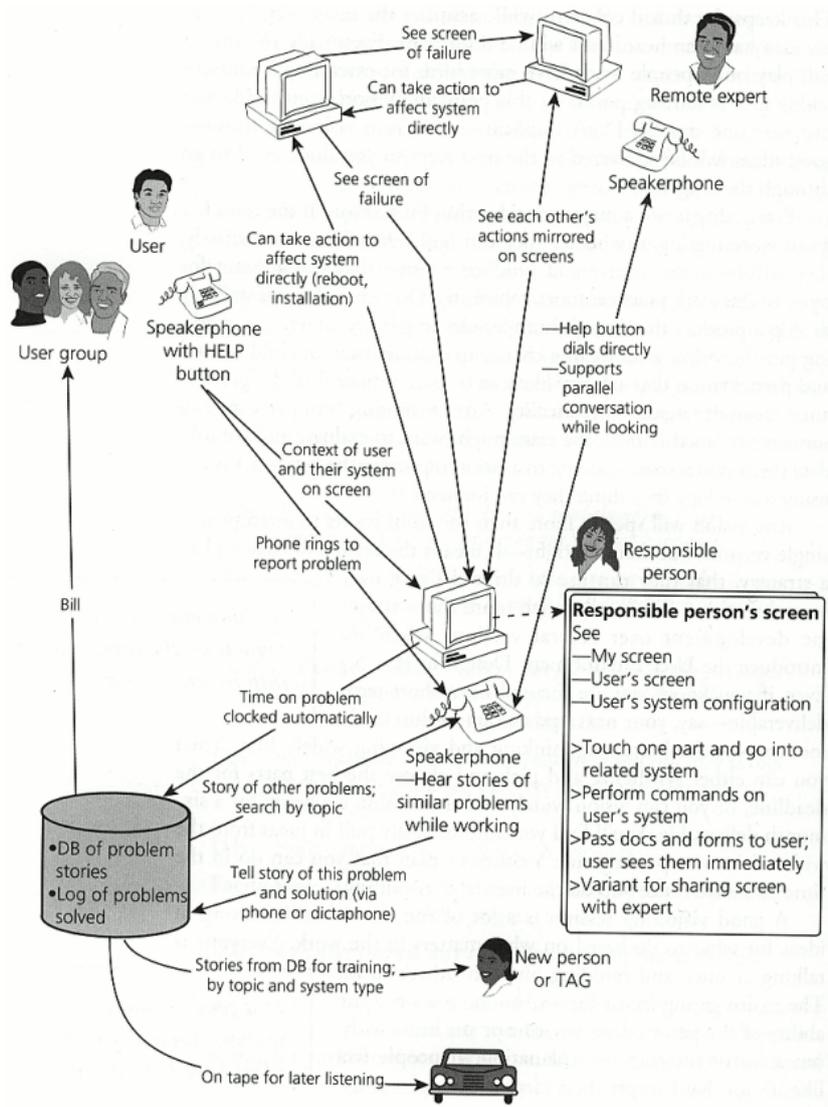
- keine Evaluation während der Entstehung
- jeder macht mit, Ideen sind nicht Eigentum eines Einzelnen
- zum Zeichner sprechen, nicht unter einander
  - “was nicht auf dem Flipchart ist, ist nicht im Design”
- Vorschlagen von Designlösungen und Benutzeraktionen
  - keine Fragen in die Runde werfen
  - keine Diskussion, sondern der Versuch eine Teamgeschichte zu erzählen
- die Geschichte wird aus Sicht des Benutzers erzählt
  - Startpunkt:
    - \* Wer bin ich?
    - \* Was mache ich?

## Anzahl

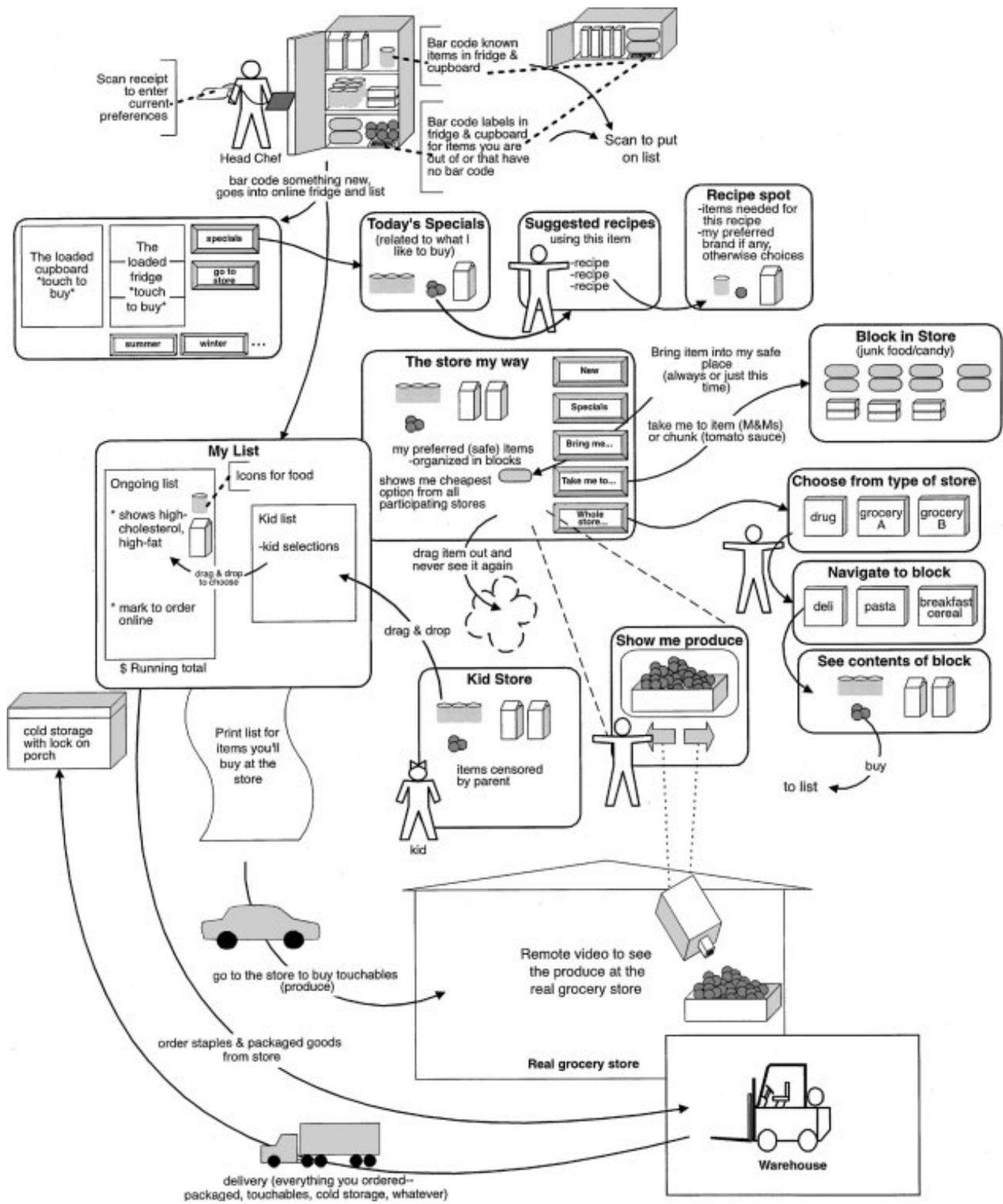
- Verschiedene Visionen mit unterschiedlichen Schwerpunkten erstellen:
  - Wert für den Kunden
  - Arbeitsaufwand
  - technische Möglichkeiten
  - Firmenpolitik

## 3.2 Beispiele

### Beispiel: System Management

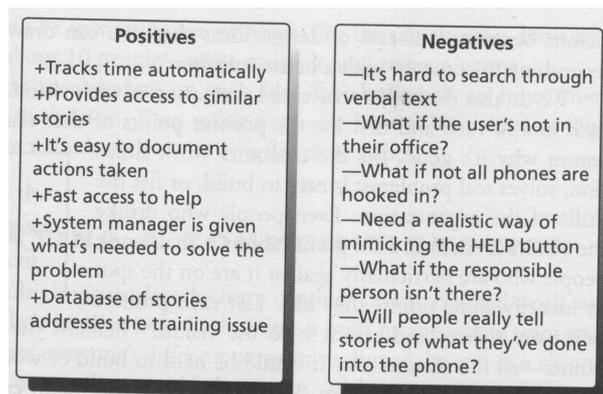


## Beispiel: Shopping System



Synthese

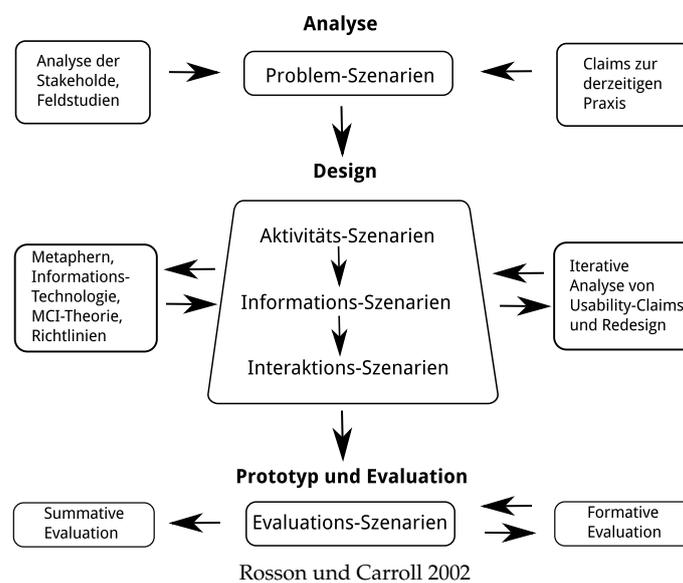
Beispiel System Management  
 Synthese durch Kritik: Claims Analysis



Claims Analysis ist auch zentrales Instrument des Szenarienbasierten Designs

### 3.3 Szenarienbasiertes Design

#### Phasen

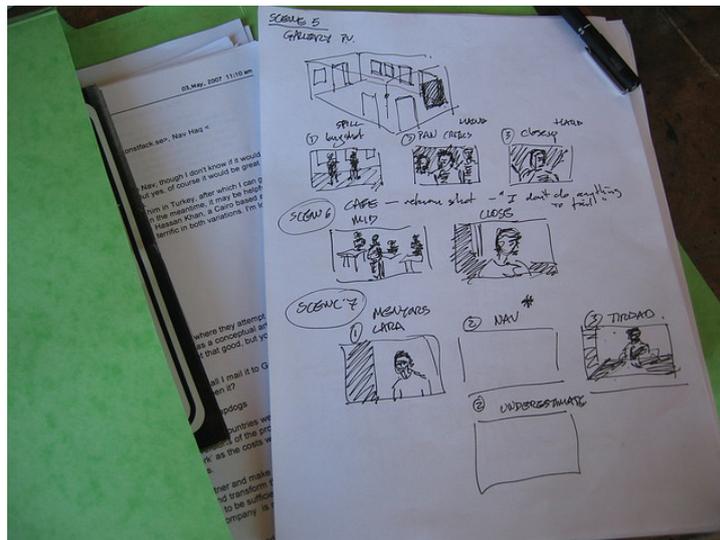


#### Szenarien

- Szenarien können als Vorstufe oder Alternative zu Storyboards dienen
- In Ihnen kann die Vision detailliert ausgestaltet werden
- Die Claims-Analyse spielt eine ähnliche Rolle wie im Contextual Design
  - Minimiere Negatives
  - Verstärke Positives
- Die Szenarien können durch Storyboards um visuelle Elemente und graphische Explikationen von Designs ergänzt werden

## 4 Storyboards

#### Storyboards



cc by-sa Neil Cummings

## Überblick

- zeigen die Details der Vision
- dienen zur Überprüfung der Vision
- beschreiben alle Schritte: Handlungen, System-Aktionen
- enthalten User-Interface-Skizzen
- beruhen auf den konsolidierten Modellen (Ablauf-, Artefakt- und physisches Modell)
  - zentraler Ausgangspunkt: konsolidiertes Ablaufmodell
  - wichtig: erkennen, welche Abläufe bezüglich der Vision relevant sind
- werden meist von 2 Personen angefertigt und verglichen

## Geschichte



1) On arrival at the hotel, the guest goes straight to the Reception desk.



2) At the Reception desk, the receptionist types details of the guest into the computer and checks the guest in.



3) The receptionist asks for a credit card from the guest as a deposit.



4) The receptionist then gives the guest the key to their room.

Storyboards erzählen eine Geschichte: (Quelle: Benyon/Turner)



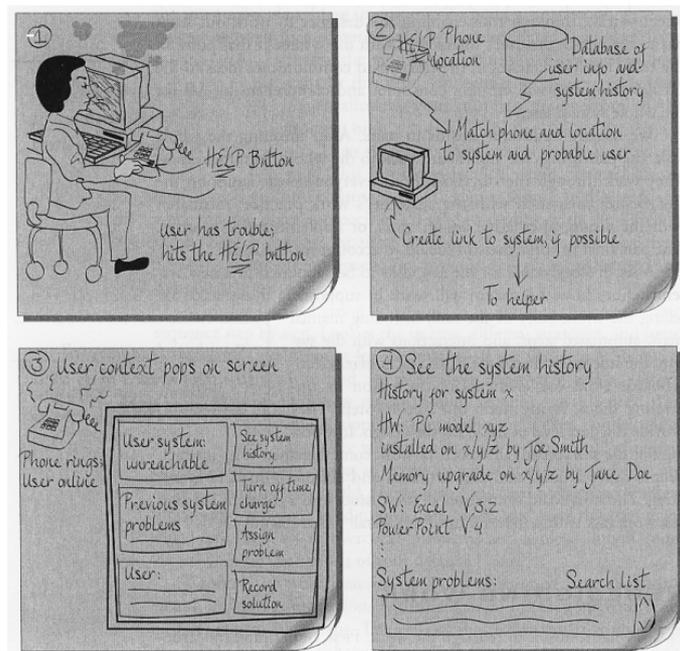
## Methode

- Ausgangsbasis:
  - Affinitätsdiagramm
  - Vision, mit der alle einverstanden sind
  - Ideen aus der Analyse
  - Arbeitsmodelle

## Vorgehen

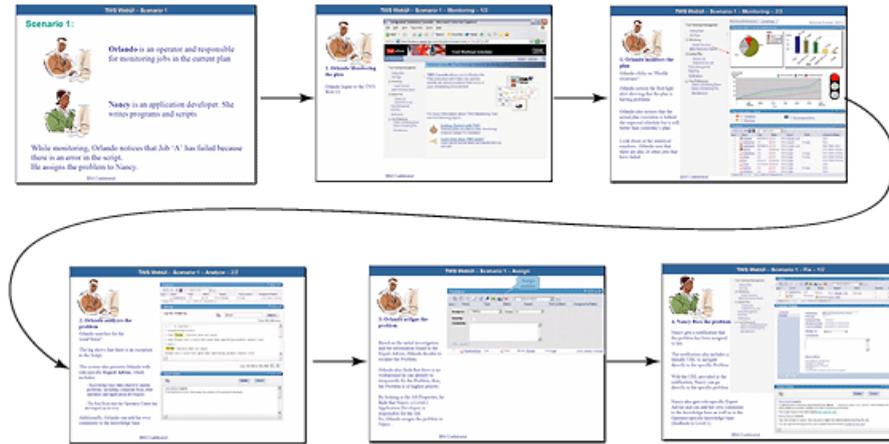
- Schritt 1:
  - identifizieren einer zentralen Aufgabe (use case), die das neue System unterstützen soll
  - (nochmals) via Modelle und Affinitätsdiagramm vergegenwärtigen, was für diese Aufgabe wichtig ist
- Schritt 2:
  - Redesign der Aufgabe (evtl. mit Ablaufmodell)
  - Alternativen vergleichen
- Schritt 3:
  - Redesign mit originalen Ablaufmodell vergleichen
  - Werden die Ziele erreicht?
- Schritt 4:
  - Storyboard skizzieren (ein Frame pro Schritt im Ablaufmodell)
  - Interaktionen zwischen Menschen, zwischen Mensch und System
  - automatische und manuelle Schritte
  - keine Details des Screenesigns
- SOLANGE weitere wichtige Aufgaben vorhanden
  - GEHE\_ZU Schritt 1
- Schritt 5:
  - alle Storyboards zusammen mit dem Rest des Designteams überprüfen

## Beispiel 1



Abarbeitung eines Hilferufs zur Vision System Management

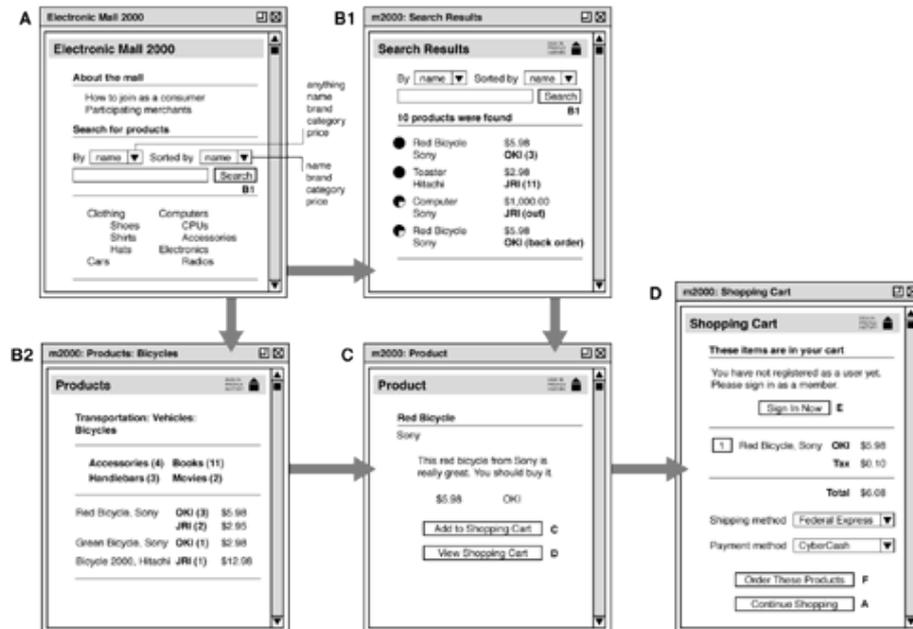
## Beispiel 2



Person, Situation, Screenshot IBM

## Beispiel 3

### Seitai Function Specifications: User Interface Draft (Consumer Side)



Nur noch Screenshots Kevcom

## 5 Bildnachweis

Alle Abbildungen, wenn nicht anders angegeben, aus:  
 Hugh Beyer and Karen Holtzblatt: *Contextual Design: Defining Customer-Centered Systems (Interactive Technologies)*. Morgan Kaufmann, 1997.