

Contextual Design

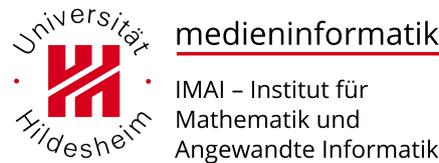
Teil VI: Prototypen

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

Prototyping

- Menschzentrierter Prozeß, aber in letzter Zeit wenig Interaktion mit Benutzern angesprochen
- Wir haben versucht, Prozesse beim Kunden zu verstehen, die Arbeit der Benutzer neu zu gestalten, aber wir haben nicht mit ihnen gesprochen
- Erläuterung des Prozesses ist sequentiell, das erleichtert die Übersicht und das Verständnis, täuscht aber Sequentialität auch dann vor, wenn nicht vorhanden
- Vom Beginn der Konsolidierung bis zum ersten Prototypen sollte aber nicht mehr als ein Monat vergehen
- Mit diesen Prototypen kann das Team weitere Daten sammeln
- Der Benutzer ist die letzte Instanz

Kommunikation

- Ein Design zu kommunizieren ist schwierig
- Demos und Spezifikationen können keine Arbeitspraxis abbilden
 - Der Kunde muß verstehen, wie das Produkt funktioniert und strukturiert ist
 - wie es in die eigene (nicht-artikulierte) Praxis paßt
 - und wie es besser sein könnte
- Herausforderung: die "Experience", die ein neues System anbietet, zu kommunizieren
- Modelle sind eine neue Sprache, die der Kunde erlernen muß
- Ansätze:
 - Szenarien
 - Use-Cases
- Aber: mehr Feedback ist möglich mit Prototypen

Einbeziehen von Benutzern

- Szenarien testen die Reaktion der Benutzer zu einer Geschichte
- Vorher: Lehrmeister
- Jetzt: Co-Designer
- Dazu suchen wir uns neue Benutzer, die nicht Teil des Design-Teams waren
- Startpunkt: Initiales Systemdesign
- Designer und User verbessern dieses iterativ
- Iterationen sind schlecht für revolutionäre Änderungen, der initiale Prototyp sollte deshalb schon sehr gut sein, die richtigen Probleme anpacken

2 Paper Prototyping

Paper Prototyping

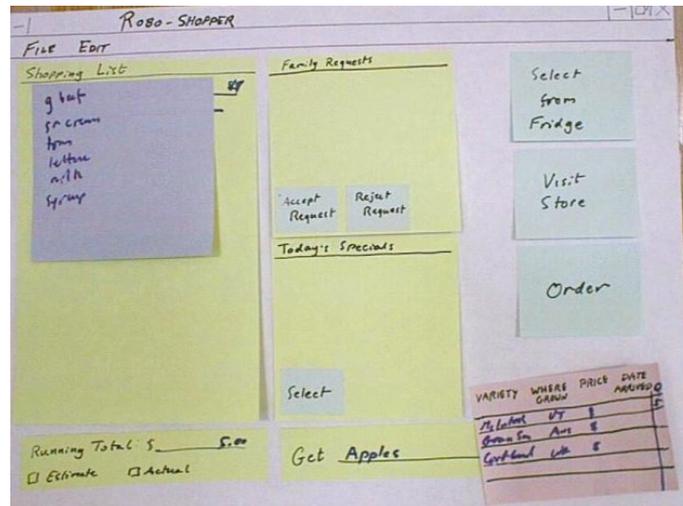


cc by Emre Ayaroglu

Hintergrund

- Herausfinden, ob ich das richtige System designe
- Der Benutzer muß keine Modelle verstehen
- Prototypen sind ein Weg, um zuverlässiges Feedback vom Benutzer zu bekommen
 - Das Design vor der Programmierung überprüfen
 - Redesign gemeinsam mit dem Benutzer

Beispiel



Bilder aus einem Video der Nielsen Norman Group

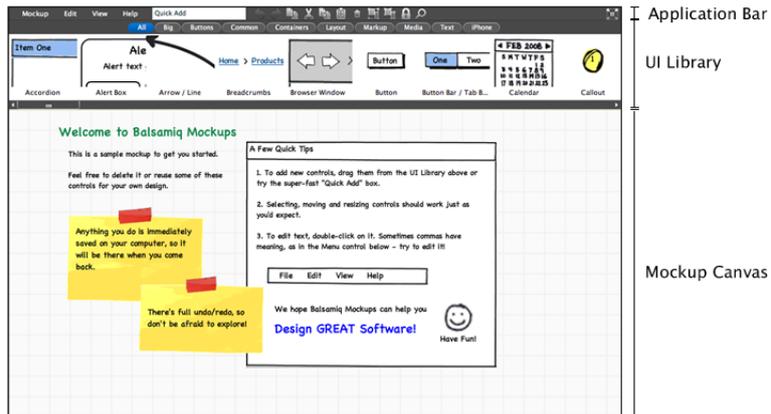
Papier

- Low-fidelity Prototyp
 - Wenn eine UI von Hand gezeichnet ist wird klar, daß das genaue Design, die Icons oder Fensterstruktur, nicht der zentrale Punkt sind
 - Wenn Benutzer mit Papier arbeiten werden sie nicht abgelenkt von den zentralen Strukturelementen, um die es uns hier geht
 - Weg, um "bikeshedding" zu umgehen
 - Papierprototypen laden zu direkten Veränderungen ein
 - "Wenn ich hier klicke, so sollte das passieren"
- Lauffähige, partielle Prototypen sind weniger einfach zu ändern und gaukeln vor, daß viele Entscheidungen schon gefallen seien
- Prototypen machen es möglich, Arbeitsabläufe zu testen, die es derzeit noch nicht gibt

Recap: Prototypen

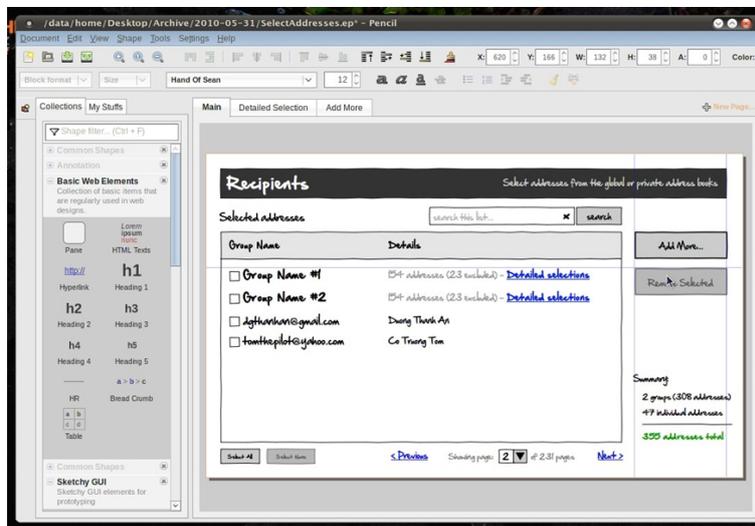
- Designskizze
 - Bewußt "unfertige" Skizzen
- Papierprototypen
 - Die einzelnen Bildschirme werden mit Papier und Stift simuliert
- Wizard of Oz
 - Das Interface ist da, aber die Funktionalität wird durch einen menschlichen Wizard simuliert
- Mock-Up (Look and Feel)
 - In einer anderen als der endgültigen Technologie realisiertes System (HTML, Flash)
- Partially working system
 - Vertikal: Ein paar Prozesse können durchgeführt werden
 - Horizontal: Alle Prozesse werden oberflächlich angerissen

Beispiel 4



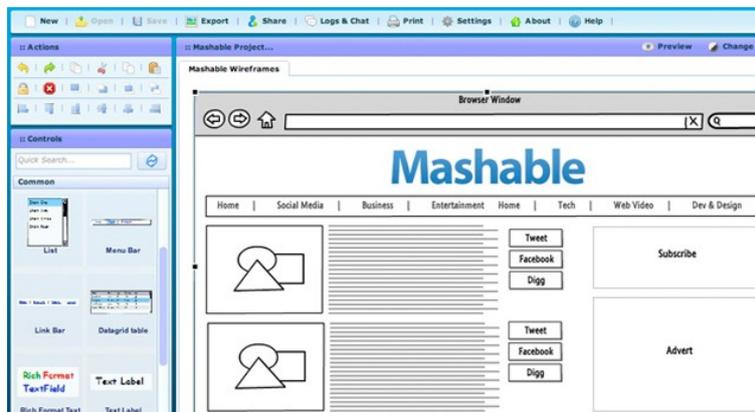
balsamiq

Beispiel 5



pencil

Beispiel 6



Lumzy

Paper Prototypes im Contextual Design Nutzen

- den Benutzer mit den neuen Arbeitsabläufen konfrontieren
- herausfinden, warum der Benutzer bestimmte Aktionen macht und was er dann erwartet

- Notizen über Reaktionen, Vorschläge, Konfusionen
 - heute i.d.R. Video-unterstützt
- Nutzer früh in den Design- und Evaluationsprozess einbinden
- “Nachspielen” der Prototyping Session, um dem Rest des Designteams die Reaktionen zu kommunizieren
 - heute i.d.R. eher Video-unterstützt

Ergebnis:

- Endversion des Papier-Prototypen zusammen mit Benutzerumgebung ist die Spezifikation des neuen Systems
 - Paper Prototype + UE = Systemspezifikation

3 Von der Struktur zum Interface

Überblick

- UED ist die Spezifikation für den User Interface Designer
 - Wie ist die Organisation
 - Welche Funktionen sollen vorhanden sein
 - Wo sollen die Funktionen erreichbar sein
- Es bleibt offen wie das zu realisieren ist
 - UI-Paradigma
 - Interaktionsangebote
 - Hardware-Plattform
 - Aussehen
- Viele Details werden am besten zusammen mit dem Kunden festgelegt

Modelle

Beispielhafte Nutzung

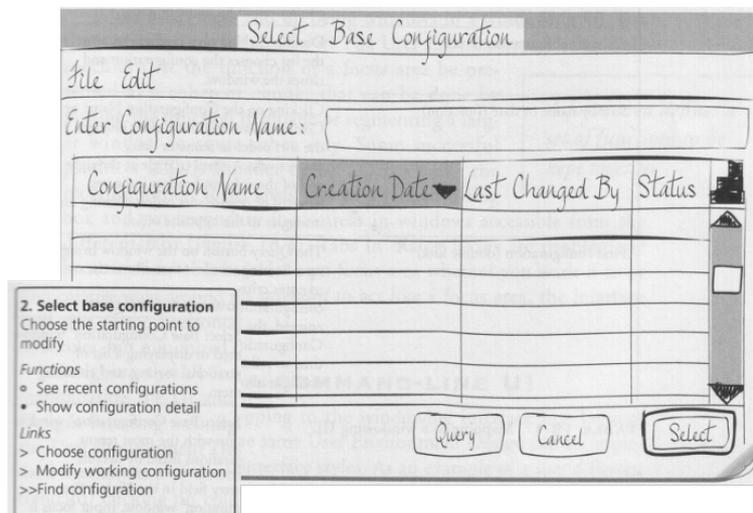
- Affinitätsdiagramme
 - Probleme
- Physisches Modell
 - Wo sind die Dinge?
 - Was ist gut erreichbar, was kann versteckt sein?
- Artefaktmodell
 - Konzeptuelle Modelle und Intentionen
- Storyboards
 - Abfolgen die mit dem UI möglich sein sollen

Erstellung

- Bildschirme
 - Große Karteikarten
- Fenster
 - Post-Its
 - Mit Titelleiste und anderen permanenten Kontrollelementen
 - Menuleiste/Werkzeuggeste, Scrollbalken

- Pull-Down Menus
 - Name auf das Fenster (immer sichtbar)
 - Inhalte auf Post-Its
 - Auflegen wenn auf das Menu geklickt wird
 - Untermenüs iterativ
- Werkzeugpaletten und -leisten
 - Permanent: auf dem Fenster
 - Veränderlich: die Fläche auf dem Fenster markieren
 - Für jedes Werkzeug die veränderlichen Elemente auf Post-Its
 - “Floating Palettes” analog auf eigenen Post-Its
- Radiobuttons, check boxes
 - Auf dem Fenster
- Dialogfenster
 - Kleinere Post-Its/Karteikarten, wie eigenen Fenster
- Fensterinhalte
 - Werden vom Benutzer während des Tests eingetragen
- Spezielle Techniken
 - Drag and Drop auf eigenen Post-Its
 - Overlays (z.B. Notizen) auf transparenter Folie
 - Tabbed Interfaces als eigene Post-Its

Beispiel 1



Mapping Fokusbereich – Windowing UI

USER ENVIRONMENT COMPONENT	WINDOWING EQUIVALENT
Select base configuration (focus area)	The "Select Base Configuration" window is devoted to displaying a list of configurations, sorting, and choosing from them.
View recent configurations (automatic function)	The "Select Base Configuration" window comes up with the most recent configurations listed by default.
Enter and choose configuration name (link)	Text entry field in the "Select Base Configuration" window. Input focus is given to this field when the window first comes up. If the user types a name and presses RETURN, the system will select the configuration (if it exists) and close the window. This supports choosing a configuration by name quickly.
Choose configuration (link)	Double-clicking on a configuration in the list chooses the configuration and closes the window.
Sort by name or date (function)	Clicking on the Configuration Name or Creation Date column header changes the sort order to name or date, respectively. A small triangle at the right end of the column headers indicates forward or reverse sort order; clicking on it toggles to the opposite order.
Find configuration (double link)	The Query button on the window brings up a floating window that allows the user to enter criteria for matching configurations. As new criteria are entered, the list in the "Select Base Configuration" window changes in real time to reflect the matching configurations.

Beispiel 2

```

CM> SELECT BASE CONFIGURATION
1. CONFIG1 8-AUG-96 JOHN SMITH
2. CONFIG2 7-AUG-96 JANE DOE
3. CONFIG3 6-AUG-96 SAM SPENCE
SELECT CONFIGURATION> SELECT 2
CONFIGURATION CONFIG2 SELECTED
CM>

```

Mapping Fokusbereich – Kommandozeile

USER ENVIRONMENT COMPONENT	COMMAND-LINE EQUIVALENT
Select base configuration (focus area)	<code>SELECT BASE CONFIGURATION</code> puts user into a mode that allows searching and specifying a base configuration.
View recent configurations (automatic function)	<code>SELECT BASE CONFIGURATION</code> responds by listing the 10 most recent configurations immediately. This fulfills the requirements for an automatic function, allowing the user to select from the list immediately.
Enter and choose configuration name (link)	<code>SELECT BASE CONFIGURATION <NAME></code> identifies the desired configuration by name. The intent of this function as defined by the User Environment Design is to make it as fast as possible to choose a configuration when the name is known. Command lines excel at this immediate action to function.
Choose configuration (link)	<code>SELECT <N></code> and <code>SELECT <NAME></code> let the user choose a configuration from the current list either by ordinal number in the list or by name.
Sort by name or date (function)	<code>SORT BY [REVERSE] {NAME DATE}</code> allows the user to choose a sort order for the list, sorting in either forward or reverse order, by name or date. The command-line system responds by relisting the current selected configurations in the new order.
Find configuration (double link)	<code>FIND [CONFIGURATIONS] WITH [NAME = <PATTERN>] . . .</code> chooses a set of configurations to view based on criteria

Optionen

- “In your face”
 - Buttons
 - Immer verfügbar, brauchen Platz
 - Leicht zu finden, da sie visuelle Clues anbieten
 - Gut für zentrale Funktionen eines Fokusbereiches
 - Physisches und Artefaktmodell geben weitere Hinweise
- “In your fingers”
 - Shortcuts via Tastatur
 - Schwer zu finden
 - Gut für schnelle Manipulation durch Experten
 - Häufig genutzte Funktionen
- Direkte Manipulation
 - Drag and Drop
 - Schwer zu finden, aber kulturell erlernt
 - Physisches und Artefaktmodell zeigen welche Dinge “mobil” sind
 - Funktioniert am besten wenn es die reale Welt abbildet
- Verfügbar wenn gebraucht
 - Pull-down Menus
 - Weder “in your face” noch versteckt
 - Gut für Funktionieren, die nicht zentral sind, aber komplementär
 - Modelle des Arbeitsplatzes zeigen Dinge, die nicht im direkten Zugriff sein müssen
 - Gruppierung nach Intentionen

- Dialogbasiert
 - Eigene Fenster
 - Dialoge, die keine Fokusbereiche abbilden, stören wahrscheinlich den Ablauf (“Insert page break” in Textverarbeitungen)
 - Zu vermeiden wenn sie keine Fokusbereiche abbilden

Prozeß

- Folge einem definierten Prozeß
 - Kann aufgebaut werden wie eine Vision
 - Fertige Optionen an, evaluiere sie auf positive und negative Aspekte
- Basiere das Design auf den Modellen der Arbeit
 - Die konsolidierten Modelle helfen beim UI-Design
 - Schritte der Ablaufmodelle müssen unterstützt werden
 - Das kulturelle Modell muß getroffen werden (Farben, Stimmungen, Werte)
 - Die Storyboards müssen “ausführbar” sein
- Fokus auf unterschiedliche Ebenen
 - Jeder neue Schritt, jede neue Verfeinerung zeigt Schwächen der vorhergehenden Schritte auf
 - Das UI-Design kann Fehler bei den UED aufzeigen - wo fehlen Funktionen?
 - Die Probleme sollten auf der richtigen Ebene behoben werden - in den Storyboards, in den UED, im UI-Design

4 Interview

Voraussetzungen

- Kontext
 - Arbeite mit “richtigen” Problemen
 - Bearbeite konkret Fälle, mache keine Produkt-Demo
- Partnerschaft
 - Falls der Benutzer Probleme oder Alternativen aufzeigt, modifiziere das Design
 - Der Designer hilft dem Benutzer technische Möglichkeiten zu verstehen
- Interpretation
 - Finde heraus, warum ein Design funktioniert oder nicht
 - Die Ideen der Benutzer sollten diskutiert werden
 - Finde die Motivation heraus – was wollen die User erreichen – nicht die spezifischen Ausformungen
- Fokus
 - Das System soll den Prozeß verbessern – ist das so?
 - Am Anfang geht es um Strukturen, reine UI-Probleme sollten ignoriert werden

Struktur

- Wie bei der Kontext-Erkundung folgt das Interview einer Struktur
- Anstatt die “normale” Arbeit zu machen wird hier mit dem Prototypen gearbeitet
 - Setup
 - Einleitung
 - Transition
 - Interview
 - Wrap-Up

Setup

- Vorab absprechen, damit jeder weiß, was einen erwartet
- Benutzer können neu sein oder bereits befragt worden sein
- Mache Revisionen nach zwei, drei Interviews
- Mache die Interviews mit den Leuten, die relevante Tätigkeiten durchführen
- Die Designer müssen die UED und die Prototypen kennen
- Test sollten vorab diskutiert werden – ausgehend von einer Analyse des Prototypen
- Gerne zwei Tester – einer führt den Test durch, der andere protokolliert

Einleitung & Transition

- Einleitung
 - Sich und den Fokus des Designs vorstellen
 - Suche nach relevanten Tätigkeiten
 - “Laß uns ‘mal so tun, als ob...”
- Transition
 - Greife eine konkrete Aufgabe heraus und spiele diese durch
 - Gib eine Hilfestellung beim “Mapping” des neuen Tools und der alten Arbeit
 - Immer noch: keine Produkt-Demo

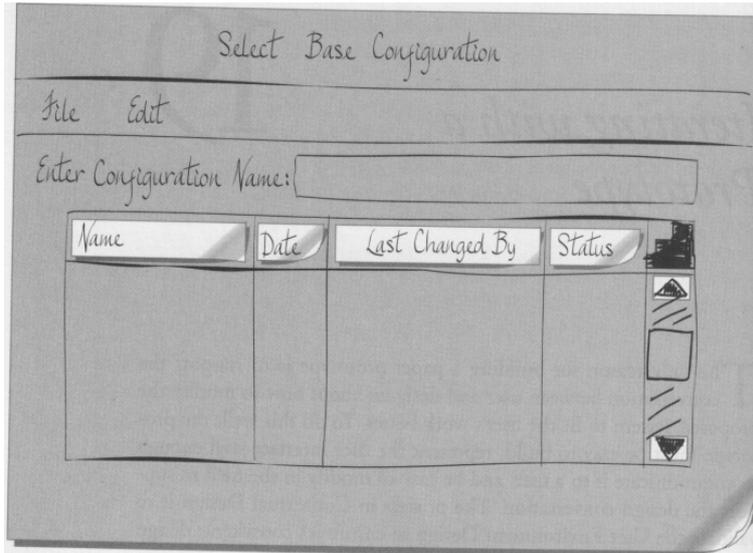
Interview

- “Sei die CPU”
 - Überlasse dem Benutzer die Art, wie er das System erlernen möchte – zielgerichtet zur Tätigkeit, explorativ
 - Ändere das Design um Ideen des Benutzers aufzugreifen
 - “Grounding” – frage was der Benutzer erwartet
- “Sei das Hilfesystem”
 - Wenn der Benutzer Probleme bekommt, frag’ nach, was er erwartet
 - Am Anfang kann sich das System so verhalten, wie er es wünscht
 - Wenn das Design anders gedacht war sollte die intendierte Funktionalität danach geprüft werden
 - Im Laufe der Zeit mehr und mehr Hilfesystem – wenn etwas nicht in zwei, drei Sätzen erklärt werden kann stimmt vielleicht die Modellierung nicht

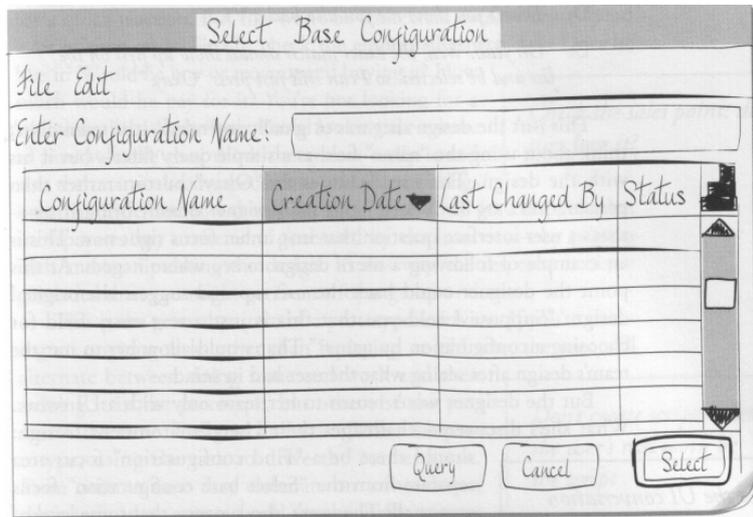
Wrap-Up

- Zusammenfassung der aufgetauchten Punkte
- Wenn etwas offen geblieben ist, kurz ansprechen (Demo des nicht benutzten Teil des Systems)
 - schwierig, da nicht im Kontext
- Frage nach, wie das System gefallen hat
- Würde es gekauft werden?
 - Warum?
 - Warum nicht?
- Nicht einfach ein Skript abarbeiten

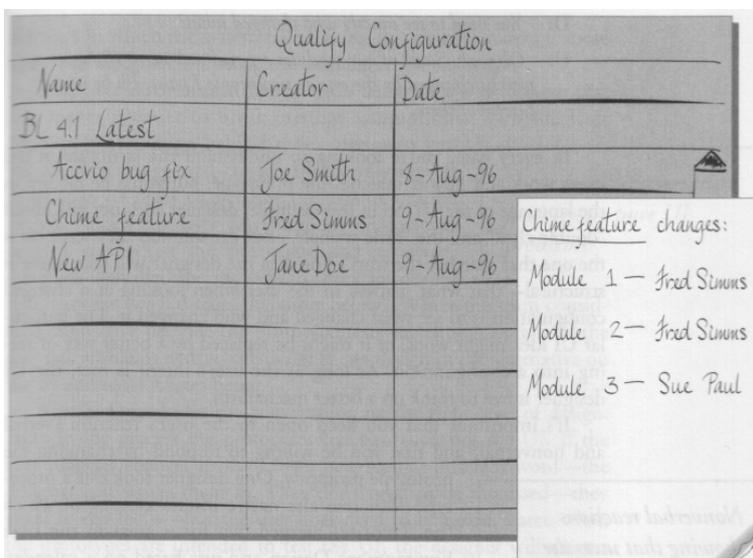
Beispiel 1



Beispiel 2



Beispiel 3



Interpretation

- Alle im Team auf einen gemeinsamen Nenner bringen
- Probleme werden auf Post-Its notiert
- Wenn Probleme im UED offensichtlich werden – mit Post-It markieren
- Die Modelle nicht während der Sitzung neu erstellen

Iteration

- Fehler müssen auf der richtigen Ebene behoben werden
- Wir kamen von den Modellen über das Storyboards zu UED und Implementation
- Auch hier: Herausforderungen so früh wie möglich ansiedeln, in der Reihenfolge abarbeiten:
 - Müssen Arbeitsmodelle revidiert werden?
 - Müssen UED revidiert werden?
 - Muß die Benutzungsschnittstelle angepaßt werden?
- Nach Restrukturierungen muß das Modell neu validiert werden – z.B. durch neue Walkthroughs
- Fragen:
 - Hat die Metapher funktioniert?
 - Waren die Interaktionsmechanismen angemessen?
 - Gibt es Probleme bei der Konsistenz?

5 Rapid CD

Agile Prozesse

- Lassen sich Prinzipien des User Centered Systems Design mit den Prinzipien des Agile Developments vereinen?
- Unterschiedliche Ziele
 - Technisches vs. User-Centered Design
- Unterschiedliche Methoden
 - Stakeholder vs. User
- Organisatorische Herausforderungen
 - Selbstorganisation vs. vorgegebene Struktur
- Unterschiedliches Prozeßmodell
 - Modelliere so wenig wie nötig vs. "Big Design Up Front" (BDUF)

Strategien

- **Lightning Fast:** Zielgruppe und Vision einer Lösung in ein bis vier Wochen
 - 4-12 Benutzer
 - Keine Ablaufmodelle, Konsolidierung, Prototypen
- **Lightning Fast+:** Zielgruppe, Vision einer Lösung, Papier-Mockup und Test in vier bis acht Wochen
 - 6-12 Benutzer
 - Keine Ablaufmodelle, Konsolidierung
 - Test mit 4-9 Benutzern
- **Focused Rapid CD:** Zielgruppe, Sammlung und Konsolidierung von Ablaufmodellen, Vision einer Lösung, Storyboards, Papier-Mockup und Test in sechs bis zehn Wochen
 - 8-12 Benutzer
 - Tests mit 6-12 Benutzern
- Noch agil?

Phasen

- Anforderungssammlung
 - Sammlung von Benutzerdaten (Interviews)
 - Modellierung und Konsolidierung
- Arbeitsneugestaltung
 - Einführung neuer Technologien, um den Arbeitsprozeß zu optimieren
 - Neuorganisation der Arbeit
- User Experience Design
 - Systemunktionalität verfeinern
 - Bestimmung des UI durch iteratives Testen

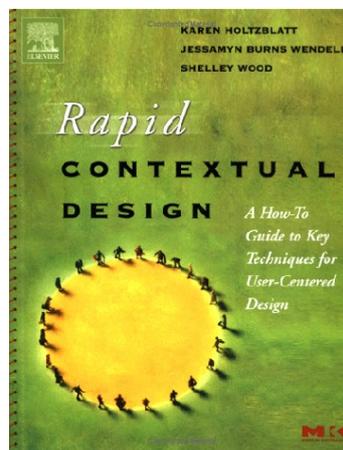
Prozeß

- Projektfokus
 - Komplexität abschätzen
 - Identifikation von 1-2 Schlüsselrollen
 - Interviews planen
- Kontexterkundung
 - Mindestens 3 Leute pro Rolle befragen
 - Eine Woche
- Affinitätsdiagramme und Konsolidierung
 - Affinitätsdiagramme zeigen die Bandbreite der Herausforderungen
 - Ablaufmodelle zeigen wie spezifische Aufgaben erfüllt werden
 - 3-4 Tage
- Einführung der Daten in das Team
 - Die Ergebnisse werden dem Team präsentiert
- Identifikation von Schlüsselproblemen
 - Das Gesamtteam entscheidet welche Probleme angegangen werden
 - Brainstorm für Lösungen zu den drängendsten Problemen aus den Affinitätsdiagrammen
 - 2 Tage
- User Stories als Antwort auf diese Probleme
 - User Stories aus Ablaufmodellen erstellen, Storyboards helfen
- Release Planning auf Grundlage der User Stories
 - Mit Hilfe der Modelle und ersten generellen UI-Mockups entscheiden welche Features in welcher Iteration angegangen werden
 - Stories, die kohärente Arbeitsabläufe schaffen, zusammenfassen
- Priorisieren von Stories
 - Anpassen an Budget- und Ressourcen-Constraints
- UI-Design zur Unterstützung der Stories
 - 1-2 Tage
- Test des Design mit Papierprototypen
 - Teste die ausgewählten User Stories mit einem Papierprototypen mit 3-4 Leuten
 - Wenn Zeit und Ressourcen vorhanden in zwei Runden
 - 2 Wochen

- User Stories und UIs als Grundlage für Entwickler
 - Kostenabschätzung und Testdesign
 - Realisierung
- Zweite Generation von UI-Designs parallel
 - Arbeit an den Stories des zweiten Iterationsschrittes
- Zweiter Satz Stories und UI-Design als Grundlage für Entwickler
 - Die Entwickler gehen in die zweite Runde
- Nächste Generation von UI-Designs parallel
 - Arbeit an den Stories des nächsten Iterationsschrittes
 - Teste die fertigen Iterationen
- Nächster Satz Stories und UI-Design als Grundlage für Entwickler
 - Die Entwickler gehen in die nächste Runde
 - Feedback der Test wird eingearbeitet
- Arbeite Feedback von Benutzern ein
 - Grundlagen für neue Iterationsschritte
 - Überarbeiten realisierter Iterationen
- Wiederhole bis Projektziel erreicht

Agile Methoden

- Voller Contextual Design Prozess nicht adäquat für agile Methoden
- Prozess muß z.B. an kurze Iterationszyklen angepasst werden
- Lösung: Rapid Contextual Design (Holtzblatt, Burns Wendell, Wood, 2005)



6 Wrap Up

Zusammenfassung

umfassende, ausgereifte, menschenzentrierte Designmethode

- pro:
 - wenige Formalismen
 - wird wirklich benutzt; z.B. bei SAP für neue Produkte:
 - * SAP Business Information Warehouse (BW)
 - * BW Administration
 - * R/3 Reporting

- contra:
 - keine expliziten Leitbilder (Was ist “gute Arbeit”?)
 - keine wissenschaftliche Fundierung (Kognition, Arbeitswissenschaften)
 - physische Anwesenheit ist bei den meisten Schritten essentiell
 - Arbeit mit virtuellen Teams im originalen Prozeß nicht möglich
 - Big Design Upfront vs. Agile Development

Intentionen I

STEP	INTENTS		
	CUSTOMER DATA	DESIGN THINKING	TEAM AND ORGANIZATION
Contextual Inquiry	<ul style="list-style-type: none"> • Gather detailed data needed for design • Discover implicit aspects of work that would normally be invisible 	<ul style="list-style-type: none"> • Put technical experts in the customer data • Stimulate the recognition of implications for design 	<ul style="list-style-type: none"> • Build the team through shared experiences • Collect concrete data to resolve conflicts
Interpretation sessions	<ul style="list-style-type: none"> • Use whole team's perspective to see what matters in the work • Capture all aspects of one customer's work efficiently 	<ul style="list-style-type: none"> • Manage the flood of insight from all team members • Capture design ideas as they come • Share preliminary design ideas to start cross-pollination 	<ul style="list-style-type: none"> • Bring multiple perspectives to bear on the data • Teach team members the perspectives of other organizations • Keep everyone engaged in processing the data
Work models	<ul style="list-style-type: none"> • Create a coherent representation of work practice • Record actual user data to check the system • Distinguish between opinions and real data 	<ul style="list-style-type: none"> • Reveal aspects of work that matter for design • Capture elements of work in a tangible form 	<ul style="list-style-type: none"> • Feed market stories, scenarios, and planning • Create a culture in which concrete data is the basis for making decisions
Affinity diagram	<ul style="list-style-type: none"> • Organize data across all customers to reveal scope of issues • Provide a review of the data prior to consolidation and visioning • Identify holes in the data 	<ul style="list-style-type: none"> • Push from point fixes to systemic solutions • Introduce inductive thinking • Allow individuals to develop their response to the data • Share design ideas without evaluation 	<ul style="list-style-type: none"> • Drive consensus about what the data means • Make data easy to share • Make key customer issues stand out • Create the first step toward corporate knowledge of their customer
Work model consolidation	<ul style="list-style-type: none"> • Create one statement of the customer population • Show common structure without losing variation across customers 	<ul style="list-style-type: none"> • Reveal implications for design through dialog with each model 	<ul style="list-style-type: none"> • Create a map of customer population for planning, sharing, and reuse • Make it possible to validate understandings with customers

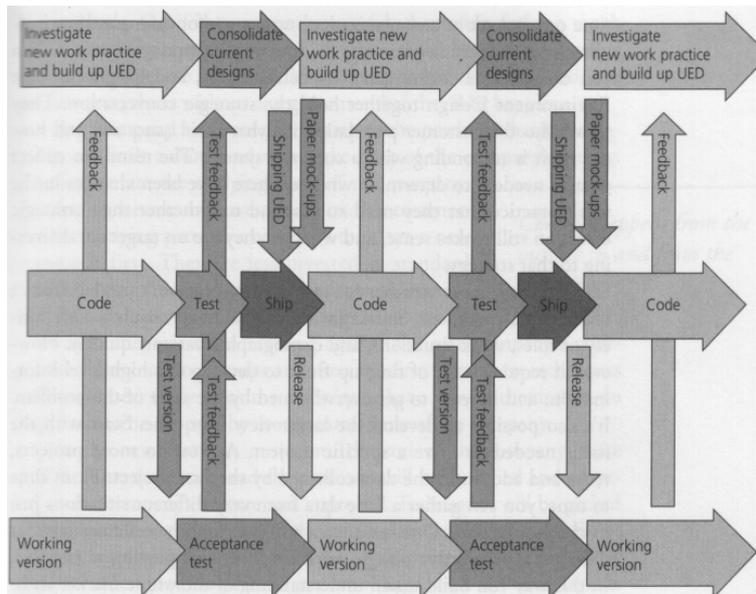
Intentionen II

STEP	INTENTS		TEAM AND ORGANIZATION
	CUSTOMER DATA	DESIGN THINKING	
Vision	<ul style="list-style-type: none"> Respond to the data with new work practice designs Shift the team's focus from tools to work practice 	<ul style="list-style-type: none"> Create a coherent response by reacting to the data rapidly Generate divergent options before deciding on one Separate evaluation from generation of ideas 	<ul style="list-style-type: none"> Develop design ideas together as a team Defuse ownership in ideas
Storyboards	<ul style="list-style-type: none"> Redesign work practice, not technology Ground redesign in consolidated data Ensure redesigned work practice hangs together 	<ul style="list-style-type: none"> Work out details of vision sequentially Let designers think in the UI without committing to it 	<ul style="list-style-type: none"> Create a public representation of a task for sharing and checking Enable parallel design work in small teams
User Environment Design	<ul style="list-style-type: none"> Design the user's experience of the system to be coherent Allow different user scenarios to be checked in the system 	<ul style="list-style-type: none"> Make the system work model explicit Show relationships between parts of the system Find errors in system structure before coding Drive later object modeling Separate out the UI conversation 	<ul style="list-style-type: none"> Make the system structure explicit and sharable Show the relationship between systems Provide a tool for planning and coordinating multiple systems and teams Provide a high-level specification
Paper prototyping	<ul style="list-style-type: none"> Check system structure and user interface with customer Let the customer communicate in their own language Get an additional layer of detailed data about actions within the system Check sales point of potential products 	<ul style="list-style-type: none"> Provide a fast way to check design alternatives Learn to separate UI from structural implications 	<ul style="list-style-type: none"> Create and test ideas quickly to prevent overattachment Ensure a shared understanding of what customers find valuable Share ideas in terms that customers and management can understand

Was brauche ich für mein Projekt?

- Beispiel: Welche 10 Dinge kann ich in meinem derzeitigen Projekt besser machen?
 - mit 6-8 Benutzern über das existierende Produkt und die Alternativen reden
 - Benutzer bei der Benutzung beobachten
 - Interpretationssitzung: nur Notizen, keine Modelle
 - Affinitätsmodell erstellen und 10 Top-Themen herauskristallisieren
 - (bis hierher weniger als zwei Wochen)
 - später eventuell Interviews mit Papier-Prototypen
- Beispiel: Ist mein bereits geplantes Design brauchbar?
 - Re-Design der Benutzerumgebung
 - Struktur der Benutzerumgebung ausbessern
 - neue Papierattrappen mit 8-10 Benutzern iterieren

Prozeß



Fragestellungen

- “Wir haben schon ein System, das wollen wir testen, um die 10 schlimmsten Probleme zu lösen.”
- “Wie können wir ein existierendes Produkt verbessern?”
- “Welches neue Produkt sollen wir erstellen?”
- “Wie können wir unsere bestehenden Produkte integrieren?”

7 Bildnachweis

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