

Zusammenfassung


Jörg Cassens

Medieninformatik II
Contextual Design of Interactive Systems
SoSe 2016



1 Einleitung

Klausurtermin

- Erster Prüfungstermin
 - Freitag, 22.07.2016
 - 14:00-16:00 Uhr
 - A 9 (Großer Seminarraum Samelsonplatz)
- Zweiter Prüfungstermin
 - Falls eine Woche vor dem Klausurtermin Anmeldungen im POS vorhanden
 - Letzte Woche im September oder erste Woche im Oktober
 - Voraussichtlich:
 - * Mittwoch, 05.10.2016
 - * 16:00-18:00 Uhr
 - * A 9 (Großer Seminarraum Samelsonplatz)
 - Benachrichtigung über das Learnweb
- Anmeldung im POS bitte bis Dienstag, 19.07.
- Diejenigen, die sich nicht im POS anmelden können, schicken mir bitte bis zum gleichen Termin  eine [Mail](#)

Thema

- Überblicksvortrag
 - Nicht erschöpfend
 - Anleitend
- Fragen...
 - ... können im learnweb diskutiert werden
- Danke...
 - ... vorab für gegenseitige Beantwortung von Fragen

2 Einführung

3 Designprozesse

Iterative Vorgehensmodelle

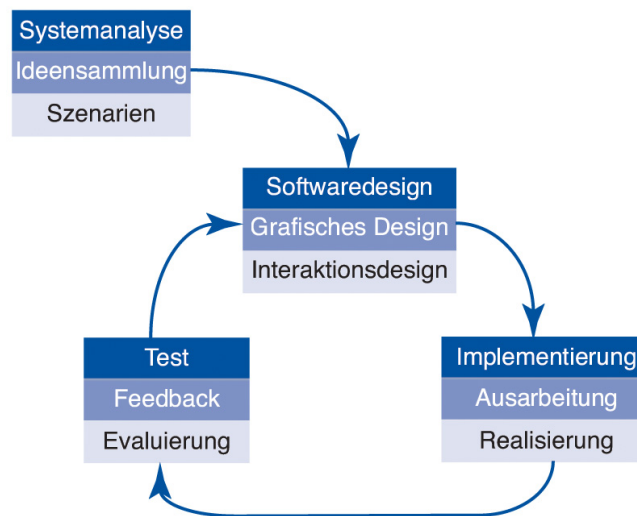


Abbildung 12.4: Iteratives Modell mit drei parallelen Aktivitätsarten

Menschzentrierte Prozesse

Ein menschenzentrierter Prozeß (User-Centred Design Process)

- ist eine Methode, die relevante menschliche Faktoren berücksichtigen hilft,
- erlaubt bewußte und nachvollziehbare Entscheidungen,
- setzt den Fokus auf wichtige Fragestellungen und Anforderungen,
- macht die Evaluation und Überprüfung von Annahmen später möglich

Prozeß und Produkt basieren auf Zielen, Aufgaben, Fähigkeiten, Bedürfnissen und Kontexten der NutzerInnen. Dafür werden diese früh in den Prozeß einbezogen.

Um den Erfolg überprüfen zu können, werden Anforderungen als quantifizierbare und meßbare Kriterien definiert.

UCD: Fokus

- Menschenzentrierte Designprozesse sind spezielle iterative Ansätze für den Entwurf interaktiver Systeme
- Bei ihnen stehen Menschen mit ihren Zielen, Fähigkeiten und Interessen im Vordergrund
- "Klassische" Modelle für die Entwicklung technischer Systeme sind häufig systemzentriert
- Besondere Herausforderung: Anforderungen oft nicht direkt in Spezifikationen umsetzbar
 - Die Anforderung "leichte Bedienbarkeit" bestimmt nicht direkt, wie das Produkt aussehen sollte
- Guidelines und Best Practices können helfen, Entwürfe sinnvoll zu gestalten
- Ohne Einbindung von Nutzern läßt sich vor allem bei neuartigen Produkten nicht sicherstellen, daß die Anforderungen auch erfüllt werden
- Dabei sind Iterationen unausweichlich

Prozeßmodell

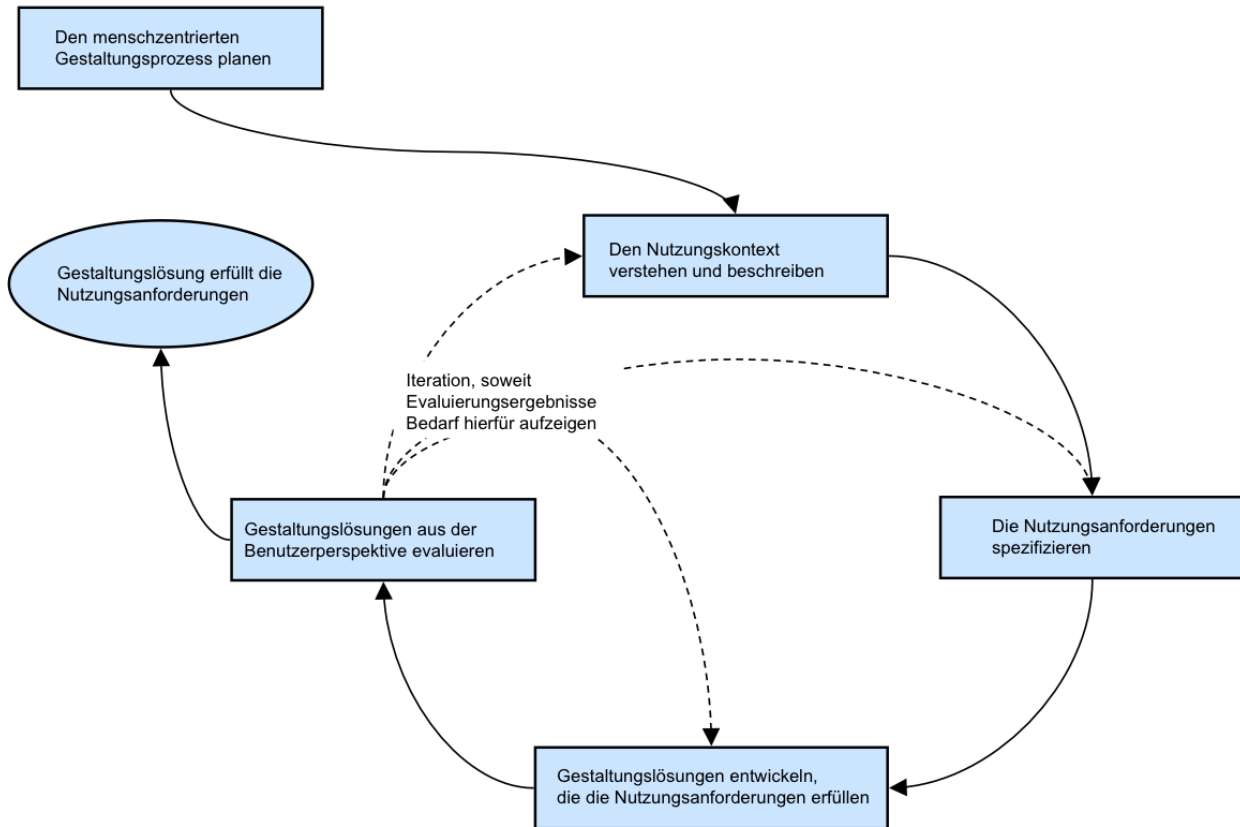


Bild 1 aus DIN EN ISO 9241-210: Wechselseitige Abhängigkeit menschenzentrierter Gestaltungsaktivitäten

Ansätze und Schritte

- Es gibt eine ganze Reihe von Ansätzen, wie Nutzer in den Entwurfsprozeß einbezogen werden können
- Im einzelnen sind folgende Schritte wichtig:
 - Analyse
 - * Definition des Kontextes
 - * Beschreibung der Organisation
 - * Beschreibung der Nutzer
 - * Aufgabenanalyse
 - * Artefaktanalyse
 - Konzeption
 - * Aktivitätsdesign
 - * Informationsdesign
 - * Interaktionsdesign
 - (Prototypische) Realisierung
 - Evaluation
- Hierbei sind nur Teile berücksichtigt, die über das im Software Engineering übliche hinausgehen

Kontext- und Organisationsanalyse

- Zumeist muß erst einmal der Kontext des Produktes beschrieben werden
 - Berufliche, sicherheitskritische, unterhaltende Anwendung
 - Marktanalyse – was erwartet die NutzerInnen?
- Beschreibung des räumlich-zeitlichen Umfeldes für den Einsatz des Systems
 - Wird das System z.B. unter freiem Himmel aufgestellt?
- Beschreibung des organisatorischen Umfeldes für den Einsatz des Systems,
 - Betrieblicher Kontext bei einem System im Arbeitseinsatz

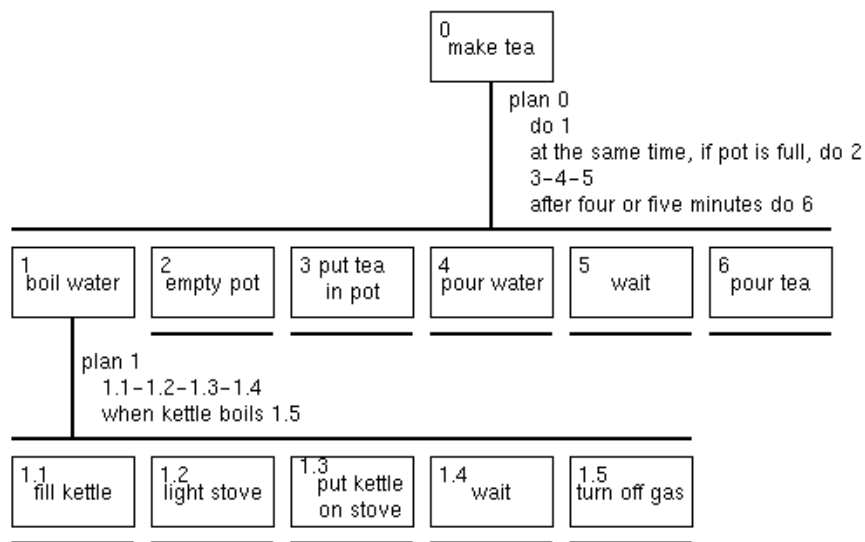
Benutzeranalyse

- Beschreibung der Zielgruppen des Systems
- Physische und kognitive Fähigkeiten
- Kulturelle und soziale Faktoren
- Verschiedene Modellierungen möglich
 - Benutzerklassen – Die späteren NutzerInnen des Systems werden anhand verschiedener Kriterien in Klassen aufgeteilt
 - * ExpertInnen oder GelegenheitsnutzerInnen
 - * Rollen, in denen sie mit dem System zu tun haben
 - Personas – eine konkrete, aber fiktive Beschreibung von Personen, die mit dem System zu tun haben
 - * Primäre, Sekundäre, Negative

Aufgabenanalyse

- Verschiedene Techniken wie Interviews oder (teilnehmende) Beobachtungen
- Wir betrachten zunächst, wie NutzerInnen bisher ihre Aufgaben durchführen
 - Nicht, wie sie glauben (oder ihr Chef denkt), daß sie die Aufgaben ausführen
- Individuelle Aufgaben und Unteraufgaben werden identifiziert und hierarchisch gegliedert
- Dazu gibt es verschieden Modelle, wie die Aufgaben aufgeschrieben werden können
- Beispiel: In einer Bank kann die Aufgabe REVIEW-ACCOUNTS aufgegliedert werden in:
 - RETRIEVE-ACCOUNT-LIST
 - FIND-RECENT-ACTIVITY
 - REVIEW-ACTIVE-ACCOUNTS
- Ein solches Vorgehen nennt sich Hierarchische Aufgabenanalyse (Hierarchical Task Analysis, HTA)

Hierarchische Aufgabenanalyse



© Nick Gibbins

Artefaktanalyse

- Beschreibung der benutzten Artefakte
- Welche Gegenstände und technischen Hilfsmittel werden benutzt?
- Was ist ein Artefakt? Ein kulturell geschaffenes Objekt.
 - Büromöbel
 - Stifte, Locher, Hefter
 - Protokolle
 - Formblätter
 - Aktenordner
 - Computer (Hard- und Software)

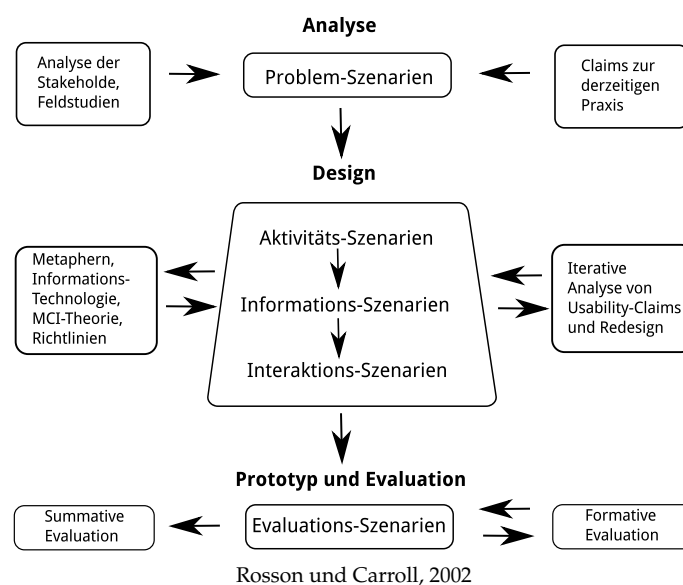
Szenarien

- Für jede Entwicklungsarbeit gilt, daß ein genaues Verständnis der zu lösenden Aufgabe Voraussetzung für ein gelungenes Resultat ist
- Unscharf definierte oder falsch verstandene Aufgabenstellungen führen immer wieder zu Problemen
- Deshalb muß jeder Systementwicklung eine genaue Anforderungsanalyse vorangehen
 - Benutzerbefragungen
 - Benutzerbeobachtungen
 - Analyse von Arbeits- oder Konsumvorgängen
- Die Ergebnisse müssen dann auch in einer geeigneten Form festgehalten werden
- Geeignete Technik: Szenarien
 - Typische Abläufe in der Nutzung des Systems in der Form einer Geschichte
 - Unterschiedliche Granularität
 - * Ein Tag oder eine spezifische Situation

Charakteristische Elemente Zusammenfassung

- **Setting** – Details über die Situation welche die Ziele, Aktivitäten und Reaktionen der Akteure erklären oder motivieren
- **Akteure** – Personen die mit dem System oder anderen Artefakten interagieren nebst relevanter persönlicher Attribute
- **Arbeitsziele** – Motivation der Akteure die Arbeit durchzuführen
- **Pläne** – Geistige Tätigkeit welche die Ziele in Verhalten umsetzen
- **Evaluation** – geistige Tätigkeit welche Aspekte der Situation analysiert
- **Aktionen** - Beobachtbares Verhalten
- **Ereignisse** - externe Aktionen oder Reaktionen des Computers oder anderer Artefakte

Szenarien: Phasen



Personas: Inhalte

- Personas werden verbal (prosaisch) auf 1-2 Seiten hinsichtlich der folgenden Daten und Charakteristika beschrieben
 - Name, Vorname, ggf. Titel
 - Foto
 - Geschlecht
 - Alter
 - Nationalität
 - Familienstand, Kinder
 - Sprachkenntnisse (beruflich, privat)
 - Ausbildung und berufliche Tätigkeiten
 - besondere Fachkenntnisse, die relevant sein könnten
 - Erfahrungen mit Computern und Anwendungen
 - Erwartungen an Computeranwendungen
 - Einstellungen zu Firma, Arbeit, Familie, eigener Kompetenz
- Kurze Zitate der Personas vermitteln Authentizität

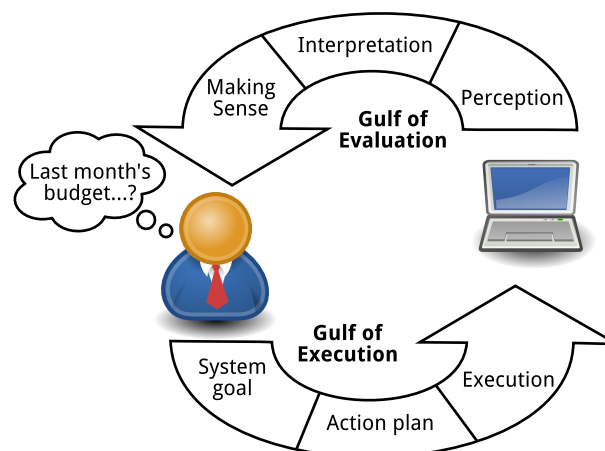
Personatypen

- Personatypen für unterschiedliche Verwendung, u.a:
 - Primary Persona (genau eine)
 - * Repräsentanten der wichtigsten Benutzerklassen
 - * Dienen als Referenz zur Diskussion und Entscheidung über ein Anwendungssystem und seine Benutzungsschnittstelle
 - * soll mit dem System sehr zufrieden sein
 - Secondary Personas (je nach Bedarf)
 - * Personas aus den wichtigsten Benutzerklassen mit wichtigen besonderen Anforderungen
 - * Dienen der Diskussion spezieller Eigenschaften eines Anwendungssystems und seiner Benutzungsschnittstellen
 - * Sind mit einem System ausreichend zufrieden, wenn die Primary Persona zufrieden gestellt ist
 - Negative Personas (je nach Bedarf, mindestens eine)
 - * Personas, die keine Benutzer des Produkts sind
 - * Dienen zur Klärung, für wen das System nicht entwickelt oder optimiert wird
- Weitere Typen: Supplemental, Customer, Served

Aktivitätsdesign

- Erster Designschritt: Welche Aktivitäten soll das neue System unterstützen?
- Im Scenario-Based Development eigener Schritt: Aktivitätsdesigns (Activity Design)
- Probleme und Möglichkeiten der derzeitigen Praxis werden umgesetzt in neues Verhalten
- Ziel ist die Spezifikation der Funktionalität des Systems
 - Welche Informationen werden vorgehalten oder benutzt
 - Welche Operationen sind möglich auf diesen
 - Welche Resultate werden zurückgeliefert
- Die Funktionalität definiert die Möglichkeiten, aber das Erleben wird durch das User Interface, die physische Repräsentation, bestimmt
- Beschreibung z.B. durch Szenarien

Gulfs



Information-Design

- The **objects** and **actions** possible in a system are **represented** and **arranged** in a way that facilitates **perception** and **understanding**
- Includes the design of
 - Application screens
 - Web pages
 - Menus
 - Dialogs
 - Icons
- Other modalities
 - Sound
 - * Speech synthesis
 - Tactile
 - * Force feedback game controls
 - Visual
 - * 3D-displays (geowall)
- Addresses the Gulf of Evaluation

Interaction-Design

- Goal: specify the mechanisms for accessing and manipulating task information
- **Information design** focuses on determining which task objects and actions to show and how to represent them
- **Interaction design** tries to make sure that people can **do the right things at the right time**
- Broad scope:
 - Selecting and opening a spreadsheet
 - Pressing and holding a mouse button while dragging it
 - Specifying a range of cells
- Addresses the Gulf of Execution

Low-Fidelity Prototypen

- Designskizze
 - Frühe Skizzen auf Papier oder Tafel
 - Vermitteln Eindruck von Designideen
 - Diskussion der Skizzen kann gegenseitiges Verständnis von Gestalter und Anwender fördern
 - Fehler können frühzeitig erkannt werden
 - Beispiele wären ein Storyboard oder Comic
- Papierprototypen
 - Bildschirminhalte werden durch Papierstreifen und Post-Its simuliert
 - Austauschen und Manipulation dieser lassen das Verhalten der Benutzungsschnittstelle erkennen
- Diese Low-Fidelity-Prototypen können mit minimalen Kosten erstellt werden
- Der offensichtlich unfertige Charakter lädt zur Kritik und Manipulation ein
- Man kann grundlegende Beurteilungen konzeptioneller Modelle erhalten

Medium-Fidelity Prototypen

- Wizard of Oz
 - Sollen Nutzer einen realistischen Eindruck eines Systems bekommen, das noch nicht implementiert ist, bieten sich Simulationen an
 - Dem Benutzer wird die Benutzungsschnittstelle präsentiert
 - Ein menschlicher Operator (Wizard) beobachtet den Benutzer und seine Umgebung und steuert die Funktionen des Systems
 - Besonders nützlich, wenn vor der Implementierung Interaktionsdaten benötigt werden
 - * Ambiente System
 - * Sprachsteuerung
- Mock-Up
 - Typischerweise mit Rapid Prototyping Werkzeugen erstellt
 - * Webseiten mit Screenshots
 - Simuliert immer noch die funktionalen Teile des Systems
 - Erlaubt typische Interaktionssequenzen

High-Fidelity Prototypen

- Partially working systems
 - Horizontal prototype
 - * all the intended functionality, but only at the top level
 - * Example: initiate a shopping spree, but cannot actually order
 - * Good for testing high level goals and action plans
 - Vertical prototype
 - * only one or two tasks are implemented in detail
 - * Example: shop til you drop, but cannot see shipping information
 - * Good when only few tasks are seen as particularly complex or important
 - Chauffered prototype
 - * Considerable functionality, but little or no error detection
 - * How: A well trained assistant accepts and executes requests on behalf of the actual test user
 - * Orthogonal to vertical and horizontal

Evaluationen

- Evaluationen können zu unterschiedlichen Zeiten im Entwicklungsprozeß durchgeführt werden
- Sie können unterschiedliche Ziele haben
- Sie können unterschiedliche Methoden benutzen
 - Formativ vs. Summativ
 - Qualitativ vs. Quantitativ
 - Analytisch vs. Empirisch
 - Feldstudien vs. Laboruntersuchungen

4 Agile Designprozesse

Challenges Usability

- **Different goals.** Technics-centered vs. human-centered.
- **Different approaches.** User-centered vs. stakeholder-centered.
- **Organizational challenges.** Collaboration of generalists vs. expert culture.
- **Process impedance mismatch.** "Big Design Up Front (BDUF)" vs. models on the go.
- Usability practitioners **struggle to be heard.**

Herausforderungen Medieninformatik

- Vorteile agiler Prozesse
 - Entwicklung digitaler Medienprodukte häufig in kleinen Teams
 - Viele Gestaltungs- und Interaktionsaspekte schlecht abschließend definierbar
- Probleme agiler Prozesse
 - Vorproduktion aufwendig gestalteter Medienelemente (Hintergrundmusik, Produktvideos) steht im Widerspruch zu einem agilen, iterativen Prozeß
 - Automatisiertes Testen funktioniert hervorragend für Programmcode, bei dem Eingabevektoren und Ausgabevektoren gut formalisierbar sind
 - Das automatische Testen interaktiver Programme schwierig (Simulation von Eingabe und Nutzerreaktionen auf Ausgaben)
- Sind Kombinationen möglich, bei denen agile Prozesse in erster Linie für die Codeentwicklung eingesetzt werden, und menschenzentrierte Prozesse für die Mediengestaltung?

5 Usability: Beispiele

6 Usability

Definition

- DIN EN ISO 9241 Teil 110
- Anforderungen an die Gebrauchstauglichkeit – Leitsätze
- “Das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele
 - effektiv,
 - effizient und
 - mit Zufriedenheitzu erreichen.”

Definition (contd.)

- **Nutzungskontext:** “Die Benutzer, Arbeitsaufgaben, Arbeitsmittel (Hardware, Software und Materialien) sowie die physische und soziale Umgebung, in der das Produkt genutzt wird.”
- **Effektivität:** “Die Genauigkeit und Vollständigkeit, mit der Benutzer ein bestimmtes Ziel erreichen.”
- **Effizienz:** “Der im Verhältnis zu Genauigkeit und Vollständigkeit eingesetzte Aufwand, mit dem Benutzer ein bestimmtes Ziel erreichen.”
- **Zufriedenstellung:** “Freiheit von Beeinträchtigungen und positive Einstellungen gegenüber der Nutzung des Produkts.”

Usability nach Ben Shneiderman

- Effektivität mit SW-Engineering-Methoden gut meßbar
- Effizienz und Zufriedenheit sind nicht direkt meßbar
- Kriterien zur praktischen Evaluation, z.B. Usability Measures von Shneiderman (Shneiderman & Plaisant 2005)
 - **Lernzeit:** Wie lange braucht der typische Benutzer, um mit dem System relevante Aufgaben lösen zu können?
 - **Aufgabeneffizienz:** Wie lange dauern bestimmte Aufgaben?
 - **Fehlerrate** der Benutzer: Welche und wie viele Fehler machen die Benutzer?
 - **Erinnerungsvermögen:** Wie gut behalten die Benutzer, was sie gelernt haben?
 - **Zufriedenheit:** Wie gefällt den Benutzern das Interface?

Evaluation: Formativ und Summativ

- Formative Evaluation
 - Während des Prozesses, “wo stehen wir?”
 - Hinweise auf konkrete Mängel bzw. Produktmerkmale, die Mängel verursachen
 - Ausgangspunkt für konstruktive Verbesserungsvorschläge
 - während der Entwicklung von Produkten
 - als Grundlage für Neuentwürfe
- Summative Evaluation
 - Am Ende, “sind die Anforderungen erfüllt?”
 - Abschließende Beurteilung
 - Produktvergleich
 - Überprüfung der Einhaltung von Kriterien
 - Zumutbarkeit von Softwaresystemen
 - Zertifizierung

Evaluation: Analytisch und Empirisch

“If you want to evaluate a tool, say an axe, you might study the design of the bit, the weight distribution, the steel alloy used, the grade of hickory in the handle, etc., or you might study the kind and speed of the cuts it makes in the hands of a good axeman.” Scriven [1967]

- Die Evaluation der Charakteristik der Axt ist eine analytische Evaluation
 - Experten bewerten das System
 - Checklisten, Cognitive Walkthroughs
- Die Evaluation der Benutzung durch den Handwerker ist empirisch
 - Nutzer benutzen das System
 - Benutzertests, Benutzerbefragungen
- Diese Unterscheidung ist orthogonal zur Aufteilung formativ/summativ

Evaluation: Qualitativ und Quantitativ

- Qualitativ
 - Konkretes Feedback, aber nicht in Form von Meßgrößen
 - * Kommentare, Eindrücke, subjektive Bewertungen in Benutzerbefragungen
 - * Detaillierte Ergebnisse einiger weniger Benutzertests
- Quantitativ
 - Erhebung von Meßgrößen in kontrollierten Umgebungen
 - * Messung von Fehlerraten, Dauer der Interaktion, Anzahl der Interaktionsschritte
 - * In erster Linie zur Messung der Effizienz des Systems

8 goldene Regeln von Shneiderman

- Konsistenz
 - Verwende Styleguides und weitere schriftliche Konventionen.
- Berücksichtige unterschiedliche Erfahrungen
 - Jedem sollte eine Benutzungsschnittstelle möglichst eine passende Interaktionsform anbieten.
 - * Anfänger: über Menüs
 - * Abkürzungen für erfahrene Benutzer
- Rückmeldungen auf Aktionen des Benutzers

- Aktion bei der Software angekommen
 - * Insbesondere, wenn die Aktion nicht schnell ein Ergebnis liefert.
 - * Akustisch, visuell, taktil
- Abgeschlossene Operationen
 - Schritte einer Operation im Zusammenhang darstellen
- Fehler verhindern
 - Darstellung eindeutig
 - Auswahlalternativen anbieten
- Einfache Rücksetzmöglichkeiten (undo)
 - Selbstsicherheit des Benutzers steigt stark an
 - Exploratives Lernen
- Benutzer bestimmt den Kontrollfluss
 - Gefühl, die Anwendung steuern, kontrollieren zu können
 - * “I am in control”
- Geringe Belastung des Kurzzeitgedächtnisses
 - Aufbau von Menüs besser breit statt tief

Usability-Heuristiken von Nielsen

- Einfache und natürliche Dialoge
 - dem Lösungsweg der Aufgabe angepaßt
 - an den Erwartungen des Benutzers orientiert
- Ausdrucksweisen des Anwenders
 - Fachsprache des Anwendungsgebiets
- Minimale mentale Belastung des Benutzers
 - “Don’t make me think” (Steve Krug)
- Konsistenz
 - Darstellung
 - Dialoge folgen immer der gleichen Logik
- Rückmeldungen
 - über Annahme der Aktion
 - insbesondere, wenn die Aktion länger dauert
- Klare Auswege
 - bei falscher Navigation
 - bei falscher Aktion
- Abkürzungen
 - Standardwerte, History-Funktionen
 - für geübte Benutzer: shortcuts
- Gute Fehlermeldungen
 - konstruktive Rückmeldungen
- Fehlervermeidung
 - besser als Fehlerbehandlung
- Hilfe und Dokumentation
 - bei Anwendung und Einarbeitung unterstützen.
 - vollständig und übersichtlich
 - korrekt und auf dem aktuellen Stand

Feldstudien und Laboruntersuchungen

- Feldstudien
 - Am Ort des Einsatzes
 - (Teilnehmende) Beobachtung der Nutzung des Systems im tatsächlichen Kontext
- Laboruntersuchungen
 - In speziellen Usability-Labs
 - Unter kontrollierten Bedingungen werden einzelne Arbeitsschritte durchgeführt

Feldstudien sind in der Regel wesentlich aufwendiger in der Durchführung, liefern aber auch realistischere Resultate

Benutzerbefragung

- Voraussetzung
 - Ausreichende Anzahl von ausreichend geübten Benutzern steht zur Verfügung
- Durchführung
 - **Strukturiertes Interview:** Benutzer werden von Experten vor Ort befragt
 - **Fragebögen:** Erfahrene Benutzer beantworten standardisierte Fragebögen zur ergonomischen Qualität selbst
- Auswertung
 - Statistische Analysen
- Ziel
 - Feststellung der Zufriedenstellung der Benutzer (Grobanalyse)
 - eher für die summative Evaluation geeignet

Feldstudien

Was sind Feldstudien?

- Das Feld ist der reale Einsatzort eines Systems.
- Feldstudien sind ein Instrumentarium aus qualitativen und quantitativen Methoden, um Informationen zu sammeln über:
 - Nutzungskontext eines Systems
 - Arbeitsprozesse oder andere Aktivitäten mit einem System
- Feldstudien sind Verhaltensbeobachtungen.
- Sie sollen den Gestaltungs- und Optimierungsprozess für ein interaktives System auf allen Gestaltungsebenen anleiten:
 - konzeptuelle (strukturelle) Ebene
 - logische (semantische) Ebene
 - Interaktionsebene

Feldstudien: Methoden

- grundsätzliche Methoden
 1. naturalistische Beobachtungen
 2. teilnehmende Beobachtungen
 - Befragung im Kontext (Contextual Inquiry)
 3. Artifact Walkthroughs
- Die Methoden unterscheiden sich in
 - Echtzeitinformation oder Retrospektion
 - Umfang der Interaktion mit den Benutzern
 - Fokus auf den Kontext und die Tätigkeit *vs.* Fokus auf die Interaktion mit Technologie

Effekte bei Feldstudien I

- **Hawthorne-Effekt:**
 - Seit den 1920er-Jahren bekanntes Phänomen
 - Die Teilnehmer an einer Studie ändern ihr Verhalten wenn Sie unter Beobachtung stehen
 - * Alle Arbeiter auf der Baustelle tragen ihren Schutzhelm solange der Beobachter dabei ist
- **Novelty-Effekt:**
 - Die stärkste psychologische Antwort auf eine (potentiell gefährliche) Situation gibt es bei den ersten Begegnungen mit der Situation
 - * Wird eine neue Technologie eingeführt so kann sich die Leistung allein aufgrund der Tatsache steigern daß es eine neue Technologie ist

Effekte bei Feldstudien II

- **Observer-Expectancy-Effekt:**
 - Die unbewußte Beeinflussung des Tests durch die Erwartungen des Beobachters
 - * Der "Kluge Hans", ein Pferd welches anscheinend zählen konnte, aber allein auf die Reaktionen des menschlichen Publikums reagierte
- **Subject-Expectancy-Effekt:**
 - Der umgekehrte Fall
 - Das Testsubjekt hat eine Erwartung, die sein Verhalten beeinflusst
 - * Placebo-Effekt

Kognitive Dissonanz

- **Loss Aversion Bias:**
 - Die Tendenz, eher einen Verlust zu vermeiden, als Gewinne zu erzielen (Teil des Status Quo Bias)
- **Negativity Bias:**
 - Negativen Informationen und Erfahrungen wird stärkeres Gewicht beigemessen als positiven
- **Selective Perception Bias:**
 - Erwartungen beeinflussen die Wahrnehmung
- **Confirmation Bias:**
 - Die Tendenz, Informationen so auszuwählen, daß die eigenen Erwartungen bestätigt werden
- **Reactance Bias:**
 - Das Gegenteil von dem tun zu wollen was jemand anders von einem erwartet um das Gefühl zu haben, frei entschieden zu haben

GOMS

GOMS (Goals, Operators, Methods, Selection Rules)

- Reduzieren der Interaktion auf elementare Aktionen (Operators)
- Elemente:
 - **Goals:** Was will der Benutzer erreichen
 - **Operators:** Aktionen die ausgeführt werden um das Ziel zu erreichen
 - **Methods:** Sequenz aus Operationen zur Erreichung des Ziels
 - **Selection Rules:** Auswahl aus Optionen
- Dekomposition in Teilziele
- Motivation:
 - Frühe Entscheidungen
 - Keine teuren Prototypen
 - Klare Metriken

KLM-GOMS

- Vereinfachte Variante
 - Nur Operatoren auf dem Keystroke Niveau
 - Keine Ziele
 - Keine Methoden
 - Keine Auswahlregeln
- Aussagen über die Dauer einer Aufgabe
- Dekomposition in primitive Operatoren:
 - Motorische Operationen
 - Mentale Vorbereitung
 - Systemantworten

7 Contextual Design

Schritte

- Phase I: Analyse
 1. Kontext-Erkundung
 2. Interpretationssitzung & Arbeitsmodellierung
 3. Konsolidierung
- Phase II: Gestaltung
 1. Neugestaltung der Arbeit
 2. Design der Benutzerumgebung
 3. Paper Prototyping

Analyse

- Phase I: Analyse
 1. Kontext-Erkundung
 - Daten sammeln durch Beobachten und Befragen von Benutzern während der Arbeit
 2. Interpretationssitzung & Arbeitsmodellierung
 - Kernpunkte der Arbeitspraxis des *Einzelnen* herausarbeiten, verschiedene Modelle für verschiedene Aspekte
 3. Konsolidierung
 - Konsolidierung der individuellen Modelle, um die Arbeitsstruktur einer *Gruppe* deutlich zu machen, ohne individuelle Variation zu verlieren

Gestaltung

- 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

7.1 Erkundung

Contextual Inquiry

- Kontext-Erkundung (Contextual Inquiry)
- Felddatenerfassung durch Beobachten und Befragen von Benutzern während der Arbeit
- Warum?
 - Designer und Entwickler kennen die Arbeit in der Regel nicht
 - Designer und Entwickler sind in der Regel nicht repräsentativ als Nutzer
 - Fragen, was der Kunde braucht oder der Benutzer möchte, ist nicht genug!

Meister und Lehrling

- Meister und Lehrling als Metapher die von den Leuten verstanden wird
 - Der **Meister** muß kein überragender Lehrer sein, er gibt sein Wissen weiter, indem er seine Arbeit tut und dabei erklärt, was und warum es gemacht wird
 - Der **Lehrling** kann nachfragen, wenn er Dinge nicht versteht
 - Beispiel DTP: *“Ich trage hier die Änderungen aus meiner annotierten Version des Textes ein. Ich arbeite mit einer Vergrößerung von 200% so daß ich sehe, ob der Text an der richtigen Stelle ist. Es macht nichts daß ich keinen Überblick über die ganze Seite habe, das werde ich in einem späteren Arbeitsgang kontrollieren...”*
- Die Interaktion fördert neues Wissen zutage:
 - Was ist wichtig?
 - Was sind die Details?
 - Was ist die Struktur?

Prinzipien

- **Kontext:**
 - Kontext selbst erleben
- **Partnerschaft:**
 - Arbeit gemeinsam verstehen
 - keine vorgefertigten Fragen
 - Benutzer leitet Interview, indem er seine Arbeit verrichtet
- **Interpretation:**
 - Verständnis gemeinsam prüfen
 - Interviewer kommuniziert sein Verständnis:
 - * → Benutzer kann Mißverständnisse ausräumen
- **Fokus:**
 - auf wichtige Aspekte konzentrieren
 - dem Benutzern den Fokus mitteilen (Offenheit)

Interview – Phasen

Traditionelles Interview

- Einleitung (Kennenlernen, Erklärung was passiert, Hinweise zum Datenschutz) – nicht mehr als 15 Minuten
- Transition: “Zugucken & Nachfragen” - 30 Sekunden
- Das Interview
- Wrap-Up – auch 15 Minuten



7.2 Interpretation

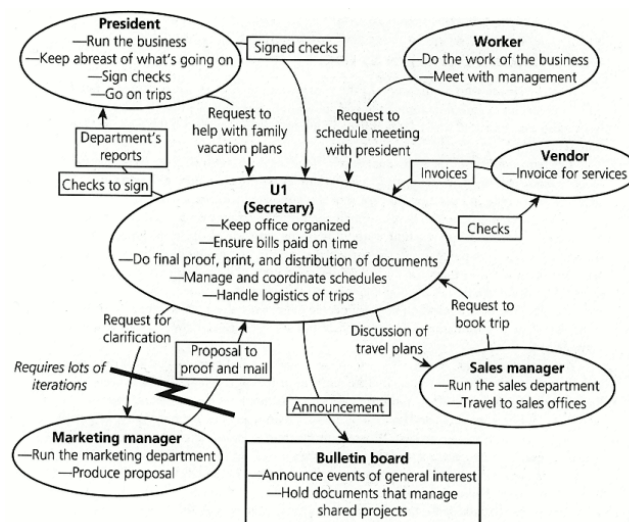
Arbeitsmodellierung

- 5 verschiedene Modelltypen, um die beobachteten und diskutierten Aspekte der Arbeit darzustellen:
 - Flußmodell (flow model)
 - Ablaufmodell (sequence model)
 - Artefaktmodell (artifact model)
 - physisches Modell (physical model)
 - Einflußmodell (cultural model)
- keine formalen Modelle,
- meist ad hoc in der Interpretationssitzung entstanden

Flußmodell (flow model)

- Die **Person** welche die Arbeit tut (in späteren, konsolidierten Modellen: ihre Rolle)
- Die **Verantwortlichkeiten** der Person
- **Gruppen** die gemeinsame Ziele haben oder Aktivitäten durchführen
- Der **Fluß** der Kommunikation um Arbeit zu erledigen
- **Artefakte**, die "Dinge" der Arbeit
- **Kommunikationsthemen oder Aktionen**
- **Orte** zu denen die Personen gehen um die Arbeit zu erledigen
- **Störungen oder Breakdowns** an denen etwas schief läuft

Flußmodell: Beispiel



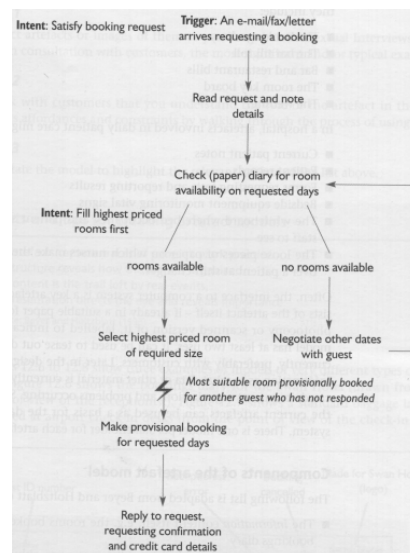
Arbeitsorganisation der Sekretärin

Ablaufmodell (sequence model)

Zeigt die Reihenfolge der Schritte, um eine Arbeitsaufgabe auszuführen

- Die **Intention** die diese Sequenz erfüllen soll
 - Sekundäre Intentionen werden eingeordnet und benannt, wenn sie identifiziert werden
- Der **Auslöser** der diese Sequenz verursacht
 - Die Notifikation die die Person zum Handeln auffordert: Telefonanruf, Höhe des Papierstapels
- Die **Schritte**, Aktionen oder Gedanken
 - Die konkreten Handlungsschritte
- Der **Fluß**, Schleifen oder Verzweigungen
 - Strategische und sich wiederholende Muster von Abläufen
- **Breakdowns** oder **Störungen**
 - Rote Blitze

Ablaufmodell: Beispiel



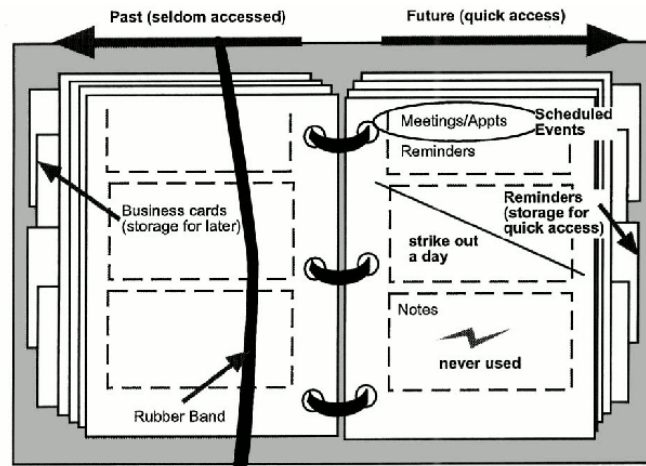
Hotelbeispiel (Benyon, 2010)

Artefaktmodell

Alle Dinge, die die Leute während der Arbeit benutzen (Kalender, Karteikarten, Ablage, Hängeregister, Formulare, Rechnungen...)

- Die repräsentierten **Informationen**
- **Teile** des Objekts mit spezifischer Nutzung
- **Struktur**, explizit oder implizit durch die Nutzung
- **Annotationen**, die den informellen Gebrauch über die Intention hinweg anzeigen
- **Präsentation** des Objekts (Farbe, Form, Layout)
- **Konzeptionelle Auszeichnungen** welche im Artefakt repräsentiert sind und seine Erstellung und Nutzung beeinflussen
- **Gebrauch** des Artefakts
- **Breakdowns** oder **Störungen**

Artefaktmodell: Beispiel



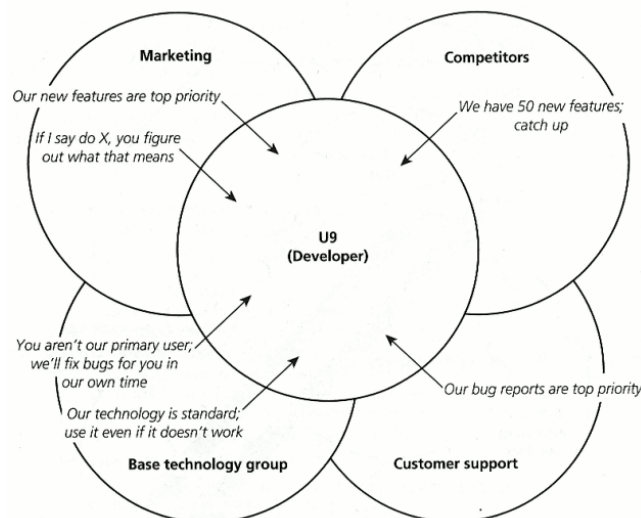
Kalender

Einflußmodell (cultural model)

Zeigt Einflüsse auf eine Person, entweder von außen auf die Firma (Abhängigkeit von einem Großhändler) oder durch interne Firmenpolitik.

- **Beeinflusser** als Blasen
 - Individuen, formelle Gruppen in der Organisation, informelle Gruppen (Management), externe Faktoren
- Das **Maß der Beeinflussung** als Überlappung der Blasen
 - Ist der Einfluß umfassend oder nur auf einen Teil bezogen?
- Der **Einfluß** auf die Arbeit als Pfeile
 - Richtung zeigt wer wen beeinflusst
 - Wie allgegenwärtig die Beeinflussung ist (Individuen, Gruppen, organisatorische Beeinflussungen)
- **Breakdowns** oder **Störungen**
 - Rote Blitze, hier nur die wirklich gravierenden

Einflußmodell: Beispiel



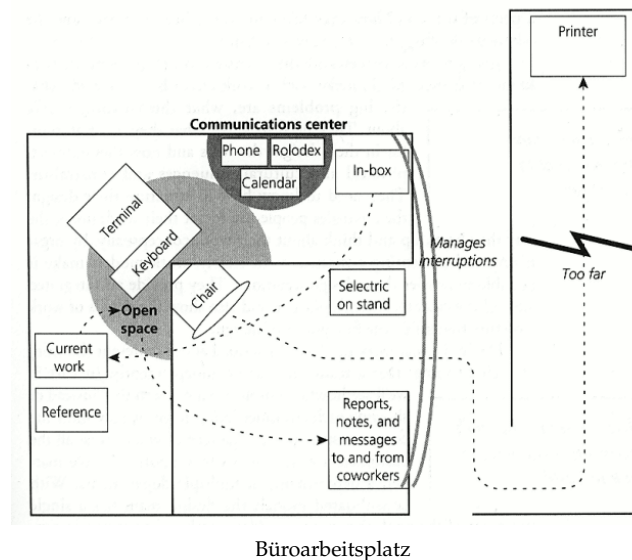
Entwickler

Physisches Modell (physical model)

Zeigt den physischen Aufbau der Arbeitsumgebung

- Welche Einschränkungen folgen für das Design?
- Wie gestalten die Leute ihre Arbeitsumgebung, um ihre Arbeit zu unterstützen?

Physisches Modell: Beispiel



Interpretationssitzung: Ziele

- **Bessere Datengrundlage:** Da der Interviewer von allen gefragt wird erinnert er sich an mehr
- **Schriftliche Dokumentation:** Die Sitzung wird "live" protokolliert, dazu kommen die erstellten Modelle
- **Effektive interdisziplinäre Zusammenarbeit:** Die einzelnen Beteiligten (Designer, Entwickler, Marketing) bringen ihr Wissen in eine klare Aufgabe ein
- **Multiple Perspektiven:** Jedes Teammitglied bringt seine Perspektive ein
- **Entwicklung einer gemeinsamen Perspektive:** Die offenen Diskussionen erleichtern das voneinander lernen
- **Arbeit mit den Daten:** Tätige Auseinandersetzung statt Präsentation
- **Zeit effektiv nutzen:** Besser alle Fragen an einem Ort, zu einer Zeit klären – sonst würden alle später fragen

Interpretationssitzung: Vorgehen

- zeitnah: innerhalb von 48 Stunden nach Interview
- in der Sprache der Benutzer
- Interpretation, keine Präsentation
- alle Teilnehmer stellen Fragen an den Interviewer
- jede interviewte Person bekommt einen Identifier oder einen (fiktiven) Benutzernamen, der in allen Modellen benutzt wird
- jeder denkt laut, keine Beurteilung
 - Brainstorming
- Alle Mitglieder des Teams lernen alle Interviews kennen

Zeitbedarf: ca. 2 Stunden

Ergebnisse: Einsichten, Modelle, erste Design-Ideen

Interpretationssitzung: Protokoll

- Zusätzlich zu den angesprochenen Modellen
- "The recorder keeps notes of the meeting online, displayed so everyone can see them using a monitor or LCD projection panel"
- Every key observation, insight, influence, breakdown is captured
- Captures sequence of the conversation, just collected in an editor

U4 18 Copies of sample cards in shoeboxes; has to keep them for 2 years (regulation)

U4 19 Home office lost cards sent in, had to make copies from one copy and send again

U2 21 Q: is there a defined procedure for storing and disposing cards?

7.3 Konsolidierung

Ziele

- Ziel der Konsolidierung:
 - schlüssige Daten über die Benutzerpopulation gewinnen
 - Diskussionsgrundlage für Designer & Entwickler über Schlüsselanforderungen
 - Designideen anregen
- Problem:
 - die Masse der Daten verführt, Daten zu vereinfachen oder wegzulassen
 - Folge: Komplexität geht verloren
- Lösungsansatz:
 - Zusammenführung der individuellen Modelle, um die Arbeitsstruktur einer Gruppe deutlich zu machen, ohne individuelle Variation zu verlieren

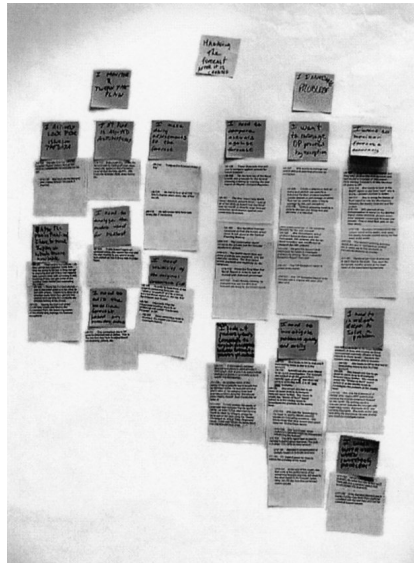
Affinitätsdiagramm

- herausarbeiten der Affinitäten im Affinitätsdiagramm
- wandgroßes, hierarchisches Diagramm
- wandgroßes, hierarchisches Diagramm
- bringt Themen und Einsichten aller Interviewten zusammen:
 - gemeinsame Probleme,
 - Unterschiede,
 - Arbeitsmuster,
 - Bedürfnisse
 - aber auch individuelle Variationen
- Grundlage für "tolle Ideen" direkt aus den Daten

Affinitätsdiagramm: Methode

1. Jede Einzelnotiz (Anforderung/Wunsch/Bedürfnis) aus der Interpretationssitzung auf ein Post-it bringen (bis zu mehreren 100)
2. Eins nach dem anderen an die Tafel bringen
3. Gleichartige gruppieren
4. Wenn die Gruppen zu groß werden (ab ca. 4) unter einem blauen Post-It zusammenfassen
5. Weitere Ebenen in rosa und grün, um Gruppen zu sortieren

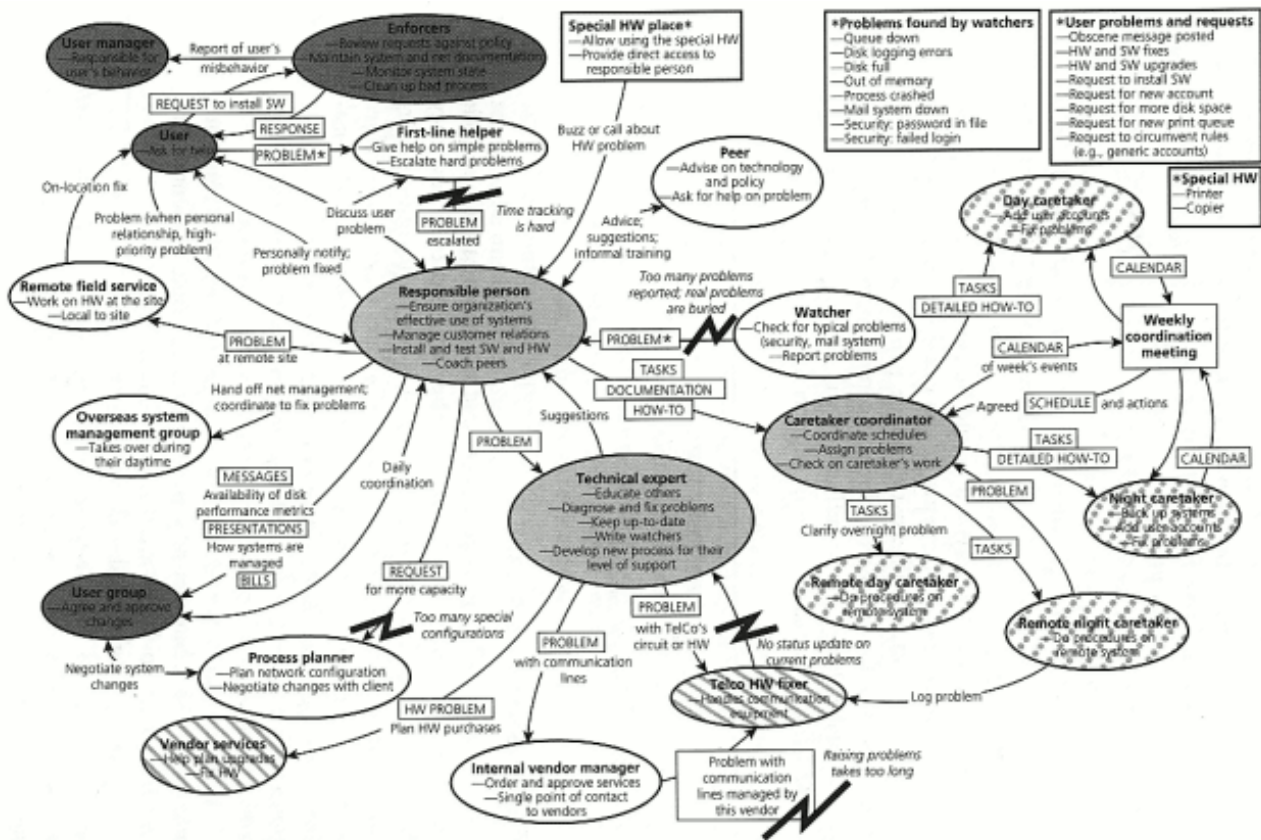
Affinitätsdiagramm: Beispiel



Konsolidiertes Flußmodell – Methode

- Einzelne Flußmodelle herausuchen die komplex und interessant sind sowie zentrale Arbeiten des Anwendungsgebiets abbilden (6 bis 9)
- Verantwortlichkeiten ermitteln und Rollen bestimmen von jeder Person, Gruppe, Ort in den einzelnen Modellen – Rollen benennen
- Ähnliche Rollen in einem konsolidierten Flußmodell zusammenführen
- Artefakte und Kommunikationswege der ursprünglichen Modelle herausuchen und zwischen den entsprechenden Rollen des konsolidierten Modells eintragen
- Jegliche Störungen übertragen
- Restliche Flußmodelle mit dem konsolidierten Modell vergleichen, fehlende Rollen, Verantwortlichkeiten und Flüsse nachtragen

Konsolidiertes Flussmodell: Beispiel



Konsolidiertes Flussmodell (System Management einer größeren Organisation)

Konsolidiertes Ablaufmodell: Methode

- Drei oder vier Abläufe welche die gleiche Aufgabe bearbeiten herausuchen – detaillierte, “passende” Modelle bevorzugen
- Aktivitäten bestimmen und aufteilen
- Passende Trigger herausuchen (diese können an unterschiedlicher Stelle stehen)
- Jeweils gleiche Arbeitsschritte herausuchen, fehlende Schritte evtl. ergänzen
- Aktivität mit abstrakten Schritten beschreiben, Störungen hervorheben
- abstrakte Schritte durch Intentionen ersetzen (z.B. am Anfang oder Ende der Sequenzen)

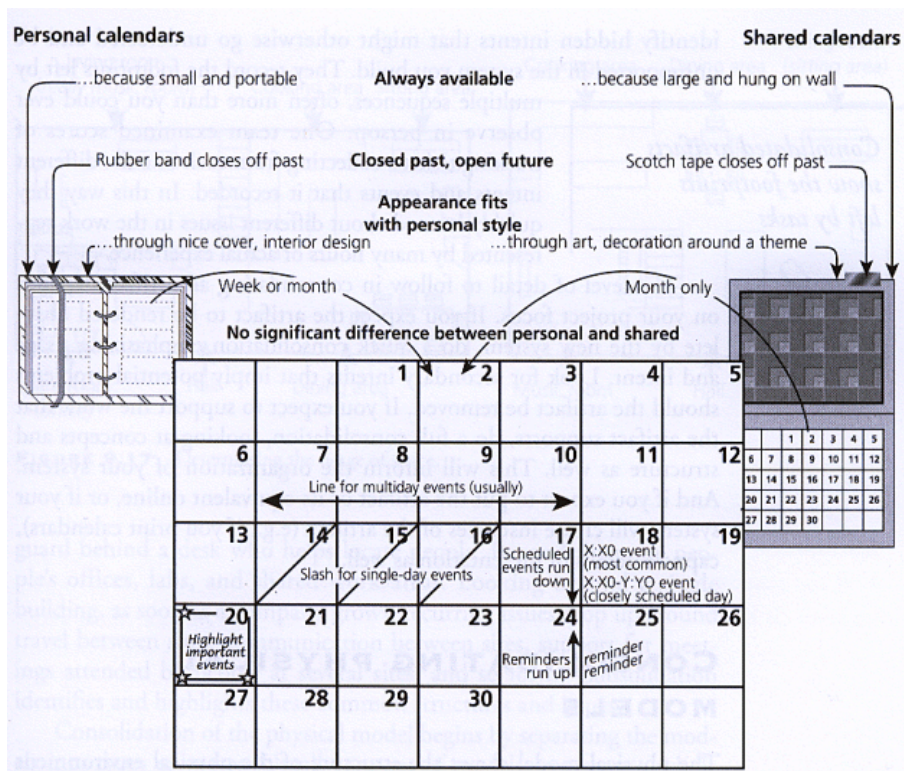
Konsolidiertes Ablaufmodell: Beispiel

ACTIVITY	INTENT	ABSTRACT STEP
Find out about problem	<ul style="list-style-type: none"> • Learn about problems quickly • Discover problems before users do • Provide quick response 	<ul style="list-style-type: none"> • Trigger: Find out about problem <ul style="list-style-type: none"> —Automated procedure —Someone reports problem
Go to problem location	<ul style="list-style-type: none"> • Make it possible to do diagnosis and take action 	<ul style="list-style-type: none"> • Go to the place where the problem can be solved
Diagnose problem	<ul style="list-style-type: none"> • Find cause of problem • Decide who's been affected • Decide if any additional action should be taken to notify people of status • Make sure I don't do things I'm not supposed to 	<ul style="list-style-type: none"> • Execute commands and tests on suspect system to identify anomalous behavior • Determine cause of symptoms • Estimate impact of problem
Fix problem	<ul style="list-style-type: none"> • Fix the problem at once 	<ul style="list-style-type: none"> • Decide whether I can fix the problem • If I decide I can fix it: <ul style="list-style-type: none"> • Attempt fix • See if fix worked • Try to figure out why it didn't work
Call on help	<ul style="list-style-type: none"> • Get the people involved who have the authority or the knowledge to fix the problem • Ensure problem gets fixed, even if not my job 	<ul style="list-style-type: none"> • Decide I can't fix it; call on help

Konsolidiertes Artefaktmodell: Methode

- Artefakte sortieren (welche Rolle spielen sie)
- Gemeinsame Teile erkennen
- Verwendung jedes Teils herausarbeiten
 - die Intention dahinter bestimmen
 - Probleme beim Gebrauch identifizieren
- Gemeinsame Struktur herausarbeiten
 - von einem Artefaktmodell ausgehend in anderen Artefaktmodellen suchen
 - Brüche zwischen den Modellen bestimmen
 - Neues, typisiertes Artefakt skizzieren,
- Konsolidiertes Artefakt repräsentiert gemeinsame Teile
 - zeigt die Variationen
 - visualisiert die Störungen

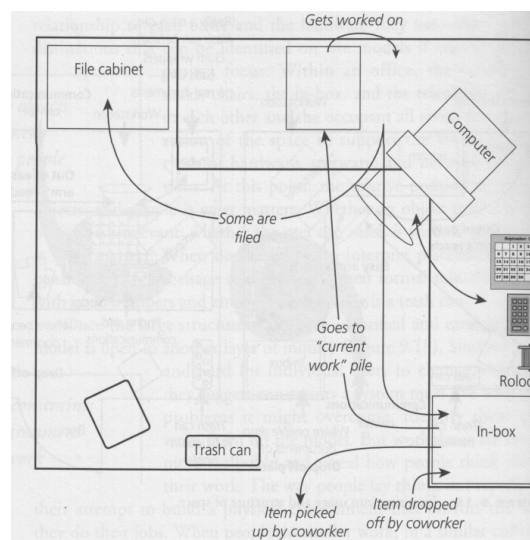
Konsolidiertes Artefaktmodell: Beispiel



Konsolidiertes physisches Modell: Methode

- Physische Modelle nach Typ sortieren
- Die Modelle durcharbeiten ("Walkthrough"), verschiedene Orte des Modells identifizieren und mit Name und Intention bezeichnen
- Für jede Art dieser Orte nach der gemeinsamen Struktur suchen – wo liegen die Artefakte und Werkzeuge?
- In den individuellen Modellen die Bewegungen verfolgen
- Konsolidiertes Modell zeigt alle Teile und ihre Struktur
- Intention, Benutzung und Störungen von individuellen Modellen übertragen

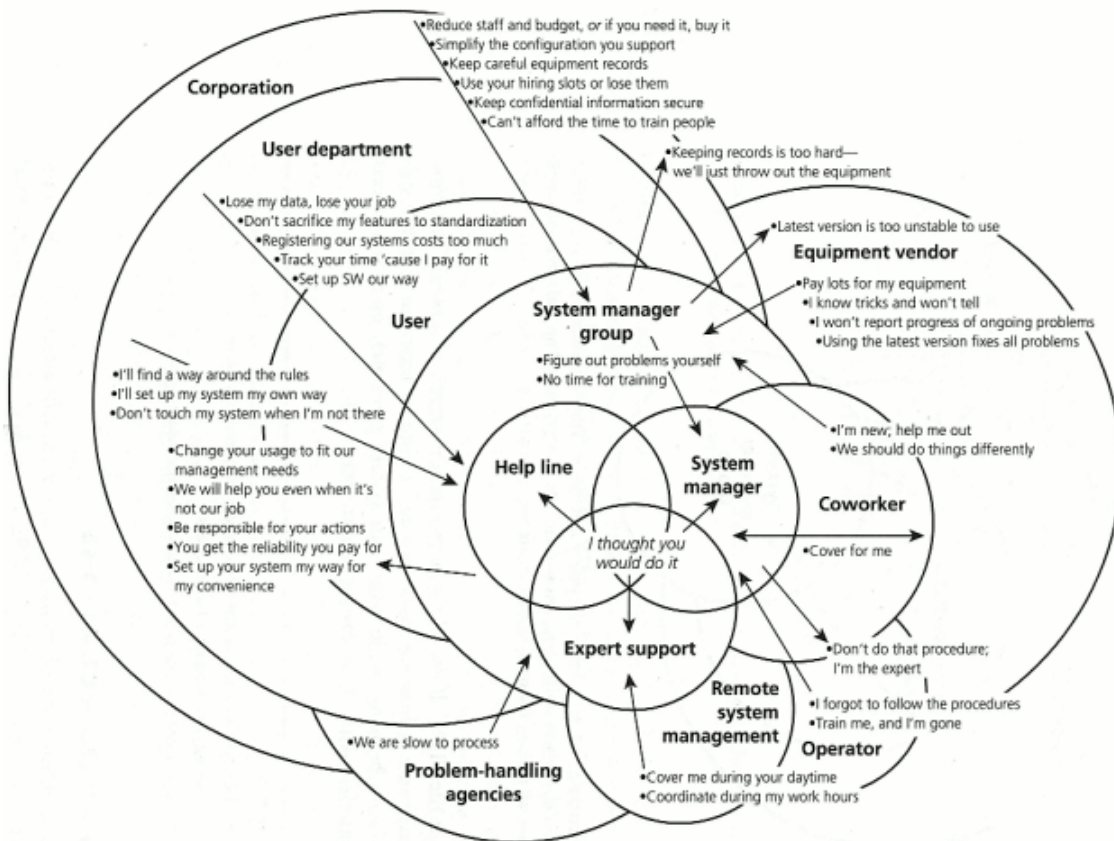
Konsolidiertes physisches Modell: Beispiel



Konsolidiertes Einflußmodell: Methode

- Beeinflusser von individuellen Modellen katalogisieren
- ähnliche Beeinflusser zusammenfassen (diejenigen, die die Arbeit auf vergleichbare Art und Weise beeinträchtigen)
- Beeinflusser der individuellen Modelle sammeln, sortiert nach den Paaren Beeinflusser - Beeinflusster
- Einflüsse den Beeinflussern zuordnen und Duplikate aussortieren
- alles in einem Modell abbilden wobei alle Beeinflussungen und Beeinflusser auftauchen sollten
- Störungen alle übernehmen

Konsolidiertes Einflußmodell: Beispiel



Anzahl der Modelle

- Wie viele Modelle haben wir zum Schluß?
- Am Ende soll in der Regel *eine* Applikation entstehen, daher so viel konsolidieren wie möglich
- Was für individuelle Modelle haben wir?
 - Ablaufmodelle mit verschiedenen Sichten
 - Ablaufmodelle unterschiedlicher Firmen
- Die Konsolidierung ist vielleicht der schwierigste Teil (Systemdesign + Neugestaltung) mit vielen Iterationen
- Konsolidieren was zusammenpaßt - Daumenregel
 - *Ein Flußmodell* - Übersicht über den Informationsfluß
 - *Mehrere Ablaufmodelle* - unterschiedliche zu unterstützende Aufgaben
 - *Mehrere Artefaktmodelle* - zu integrierende Systembestandteile

- *Ein physisches Modell pro Ort* plus Übersicht - was ist zugreifbar, wie stehen die Orte in Zusammenhang
- *Ein Einflußmodell* - sonst häufig Verdoppelung
- Man muß nicht immer konsolidieren

7.4 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)

Flußmodell: Ziele

- Reduziere oder erleichtere Rollentausch
- Automatisiere, eliminiere oder unterstütze Rollen (Reduktion der Überlastung)
- Passe das System an unterschiedliche Rollen an
- Unterstütze isolierte Rollen durch gemeinsamen Kontext

Einflußmodell: 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

Ablaufmodell: 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

Artefaktmodell: 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?

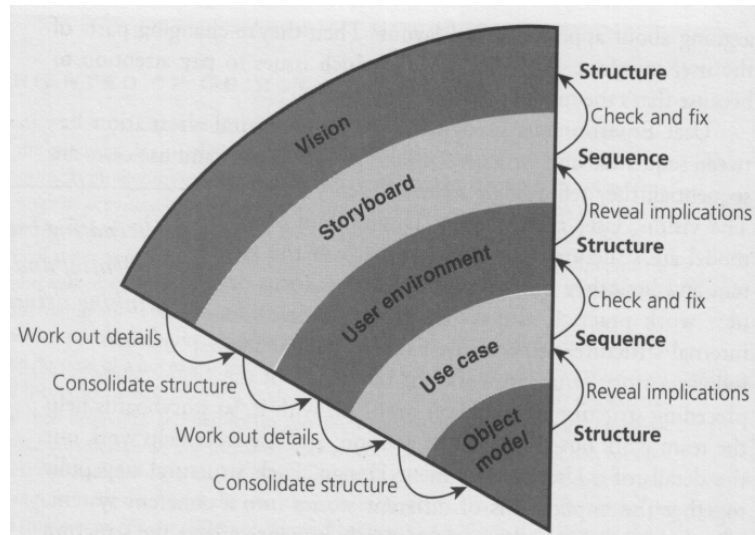
Physisches Modell: Ziele

- 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

Details und Überblick

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

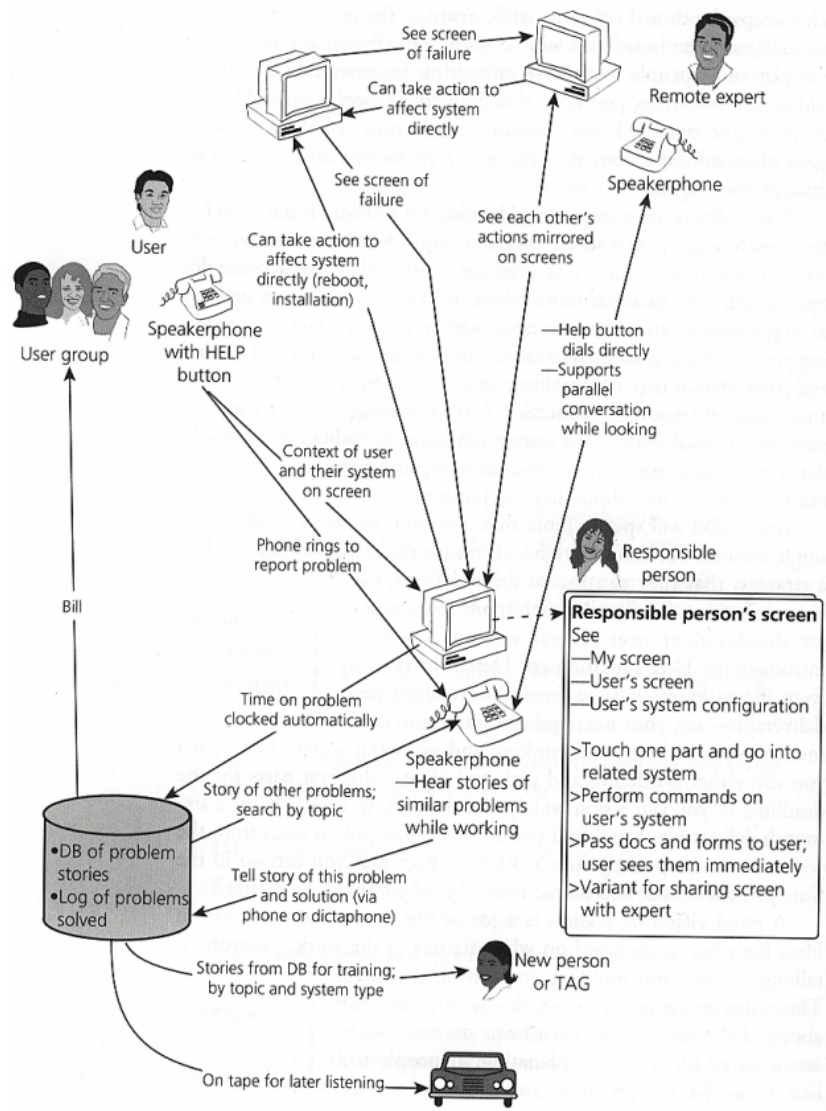
Strukturen und Sequenzen



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

Vission: Beispiel



Storyboards

- 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

Storyboards: Beispiel



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)

Vision: Claims Analysis

Beispiel System Management Synthese durch Kritik: Claims Analysis

Positives	Negatives
<ul style="list-style-type: none"> +Tracks time automatically +Provides access to similar stories +It's easy to document actions taken +Fast access to help +System manager is given what's needed to solve the problem +Database of stories addresses the training issue 	<ul style="list-style-type: none"> —It's hard to search through verbal text —What if the user's not in their office? —What if not all phones are hooked in? —Need a realistic way of mimicking the HELP button —What if the responsible person isn't there? —Will people really tell stories of what they've done into the phone?

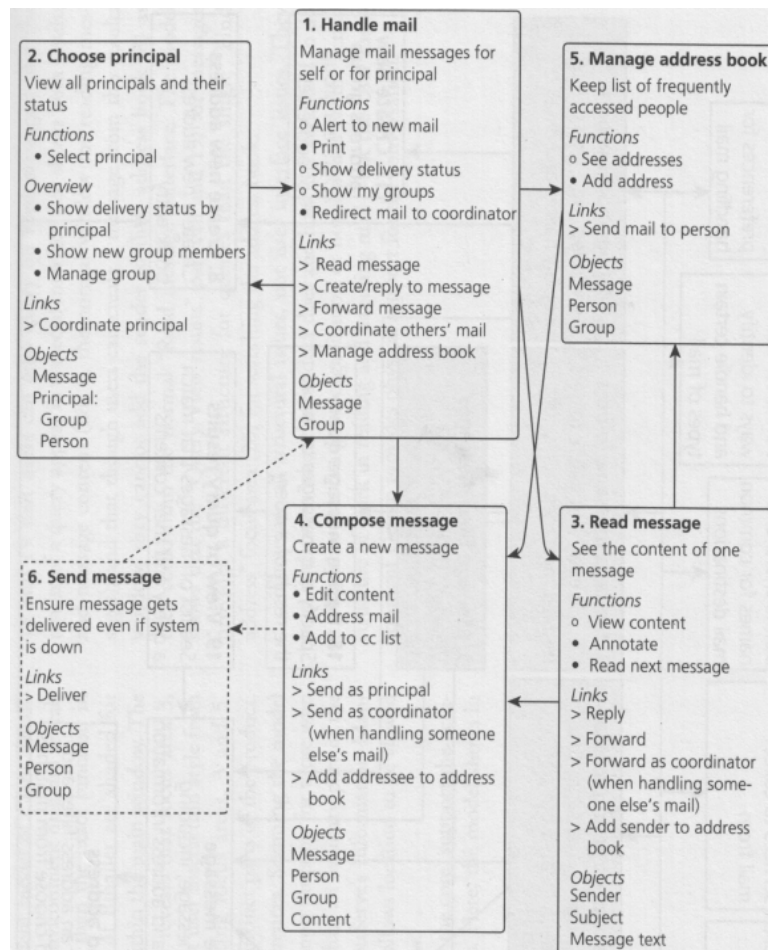
Claims Analysis ist auch zentrales Instrument des Szenarienbasierten Designs

Storyboards: 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

7.5 Benutzerumgebung

Benutzerumgebung



User Environment Design UED

User Environment Design (UED)

- Was?
 - “Raumplan” des Systems
 - * für web-basierte Systeme wie ausführlichere site maps
 - auch Interaktionsstruktur
 - basiert auf Struktur und Ablauf der Arbeit (nicht des technischen Systems)
 - wird vor dem Interface Design entwickelt
- Wozu?
 - Kommunikationsmittel für Entwickler, Designer und Projektmanager
- Merkmale:
 - teilt das System in Fokusbereiche (focus area) für jede Aktivität
 - Struktur und Funktion des neuen Systems unabhängig von Benutzungsschnittstelle und Implementierung
 - * zeigt wie die Teile in Verbindung stehen
 - * Arbeitsstruktur und Arbeitsfluss des Systems
 - konzeptuelles Tätigkeitsmodell: Aufgabenorientierung vor Objektorientierung

Fokusbereiche

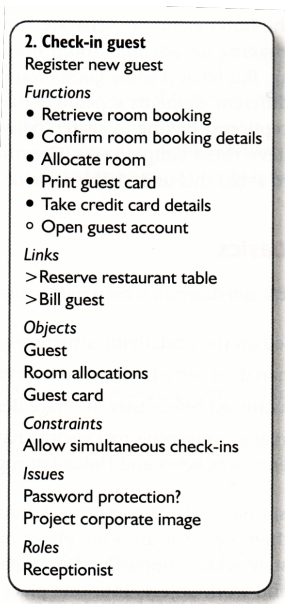
- Fokusbereiche sind Sammlungen von Funktionen und Objekten an zusammenhängenden Orten um bestimmte Arten von Arbeit zu unterstützen
- Die Funktionen sollten notwendig sein um die Arbeit zu machen, nicht um das User-Interface handzuhaben

- Jeder Fokusbereich hat
 - einen einfachen Namen der die Aktivität widerspiegelt
 - eine Nummer zur einfachen Identifikation
- Jeder Fokusbereich zeigt
 - wie er die Arbeitsaufgaben des Benutzers unterstützt
 - welche Funktionen verfügbar sind
 - welche Verbindungen zu anderen Fokusbereichen des Systems bestehen und wie der Benutzer dort hin gelangt

Struktur eines Fokusbereiches

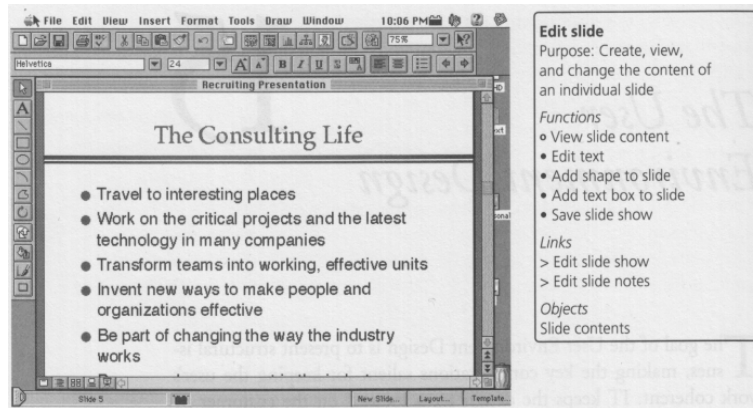
- Name
- Zweck/Ziel
- Funktionen:
 - ● = benutzerinitiiert
 - ○ = systeminitiiert
- Verbindungen (links)
- Arbeitsgegenstände (Objekte)
- Restriktionen (constraints)
- offene Punkte (issues)
- Rollen

Beispiel (Quelle: Benyon/Turner)



UED: Methoden zur Erstellung

- Zwei Varianten (je nach Projekt):
- Variante 1 (Reverse User Environment):
 - Analyse und Verbesserung einer bestehenden Anwendung
 - Benutzerumgebung als Mittel zur Analyse struktureller Probleme der bestehenden Anwendung
 - * direkte Indizien für Verbesserung
- Variante 2 (Neuentwurf)
 - Entwurf einer neuen Benutzungsschnittstelle
 - Ableitung aus Storyboards



Hauptansicht von PowerPoint

7.6 Papierprototypen

Prototypen: Motivation

- Ein Design zu kommunizieren ist schwierig
- Demos und Spezifikationen können keine Arbeitspraxis abbilden
 - Der Benutzer 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 Benutzer 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

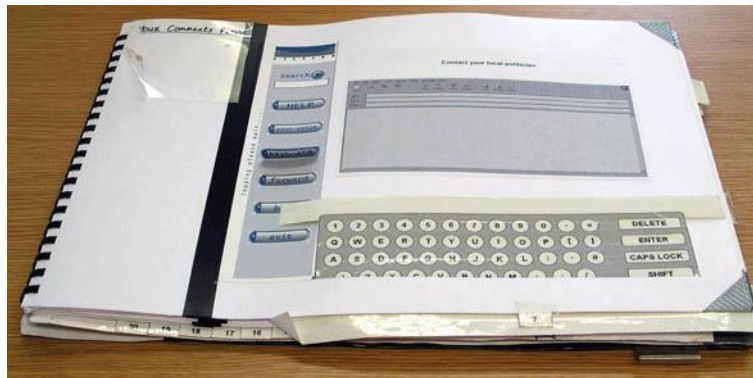
Papier

- 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
 - 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

Methode

- mit einer Papierattrappe der Benutzerschnittstelle versuchen die Benutzer “echte” Arbeitsaufgaben und mit echten Daten zu erfüllen
- Testen und Verändern der Benutzerschnittstelle
 - Benutzer als Co-Designer
- Struktur und Funktion testen, nicht Layout und Icons
- Interview nach den Richtlinien des Contextual Inquiry Interviews
- 2 Personen, einer “spielt” das System, der andere macht Notizen

Beispiel



(Quelle: Benyon/Turner)

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 Designteam 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 Prototye + UE = Systemspezifikation

Prototypen: Interview

- 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

7.7 Einschätzung

Einschätzung

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

8 Models and Metaphors

Models in Human-Computer Interaction

- Model of a system describes how it works
 - its constituent parts and how they work together to do what the system does
- We are here concerned with three models:
 - The **system model** (sometimes called implementation model) is how the system actually works.
 - The **interface model** (or manifest model) is the model that the system presents to the user.
 - The **user model** (or conceptual model) is how the user thinks the system works.
- There are more models
 - The model the developers have about how they think the user model is like
 - The model the system has about the user (inscribed, in terms of Actor Network Theory)

Interaction Styles

- We look at the following Interaction Styles
 - Command language/command line
 - Menus & forms
 - Direct manipulation
 - * Touch and Mouse
- Also interesting, but outside the scope today
 - Other forms of graphical interaction
 - 3D-Gestures
 - Natural Language Interfaces
 - Explicit *vs.* implicit interaction
 - Behavioural Interfaces

Command Line Interface

- User types in commands in an artificial language
 - Unix shell (`ls -l *.java`)
 - Search engine query language (AND, OR)
 - SQL (`SELECT FROM Book WHERE price > 100.`)
 - SPARQL (`SELECT ?name ?email WHERE {?person a foaf:Person. ?person foaf:name ?name. ?person foaf:mbox ?email.}`)
- Command syntax is important
- Powerful tool with a steep learning curve – find all .tex files that mention the word foo in a given subtree and replace those occurrences with bar
- When designing a command language, the key problem is the syntax
- Task analysis drives the choice of commands, the names you give them, the parameters they have, and the syntax for fitting them together

Menus & Forms

- User is prompted to choose from menus and fill in forms
 - web sites “before Web 2.0”
 - dialog boxes
- Navigation structure is important
 - Wizard: linear sequence of forms
- The navigation structure is the important design problem for menu/form interfaces
- Task analysis tells you what choices need to be available, where they should be placed in a menu tree, and what data types or possible responses need to be available in a form

Direct Manipulation

- User interacts with visual representation of data objects (based on Shneiderman, Designing the User Interface, 2004):
 - Continuous visual representation
 - * Verbal or iconic
 - Physical actions or labeled button presses
 - * most direct kind of action, analog to real world interaction

- * not everything can be easily mapped – convert a text to bold – so “command actions” are allowed
- Rapid, incremental, reversible, immediately visible effects
 - * within 100ms (why?)
 - * drag a bit, see the change
 - * physical or logical

Direct Manipulation II

- Examples
 - Files and folders on a desktop
 - Scrollbar
 - Dragging to resize a rectangle
 - Selecting text
- Visual representation and physical interaction are important
- It is powerful since it exploits perceptual and motor skills of the human user
- Some say it depends less on linguistic skills than command or menu/form interfaces
 - Only partly true and for a limited understanding of language

Direct Manipulation Cues

- What is the language in which a system communicates its model to the user?
- What cues rely the users on to learn the model – the parts that make up the interface, and how they work together?
- Donald Norman, *The Design of Everyday Things* (1988), identifies a number of cues
 - Affordances
 - Constraints
 - Natural mapping
 - Visibility
 - Feedback
- Since DM interfaces intend to be a visual metaphor, we look at how these apply to UI

Affordances

- *Perceived* and *actual* properties of a thing that determine how the thing could be used
 - Chair is for sitting
 - Knob is for turning
 - Button is for pushing
 - Listbox is for selection
 - Scrollbar is for continuous scrolling or panning
- Perceived *vs.* actual
 - A paper-mache chair still has a perceived affordance for sitting
 - A pole has no perceived affordance for sitting, but you can sit on it (albeit uncomfortably)
- The DM UI should agree on perceived and actual affordances

Constraints I

- Graphical screen layout relies greatly on conventional interpretations of the symbols and placement
- Different types of constraints:
 - *Physical* constraints are closely related to real affordances
 - * it is not possible to move the cursor outside the screen
 - * Restricting the cursor to exist only in screen locations where its position is meaningful
 - *Logical* constraints use reasoning to determine the alternatives
 - * If we ask the user to click on five locations and only four are immediately visible, the person knows, logically, that there is one location off the screen
 - * It is how the user knows to scroll down and see the rest of the page
 - * Logical constraints go hand-in-hand with a good conceptual model.

Constraints II

- Different types of constraints (contd):
 - *Cultural* constraints are conventions shared by a cultural group
 - * That the graphic on the right-hand side of a display is a “scroll bar” and that one should move the cursor to it, hold down a mouse button, and “drag” it downward in order to see objects located below the current visible set is a cultural, learned convention
 - * The choice of action is arbitrary: there is nothing inherent in the devices or design that requires the system to act in this way
 - * “Arbitrary” does not mean that any random depiction would do equally well: the current choice is an intelligent fit to human cognition, but there are alternative methods that work equally well.

Natural Mapping

- *Physical arrangement* of controls should match *arrangement of function*
- Best mapping is direct, but natural mappings do not have to be direct
 - Light switches
 - * If the switches are arranged in the same fashion as the lights, it is much easier to learn which switch controls which light
 - Stove burners
 - * Most stoves have four plates in a square and four controls in a row
 - Car turn signals
 - * Up and down instead of left and right, but synchronous to turning wheel
 - DJ audio mixer
 - * between turntable
- What is a direct mapping anyway?
 - Rudder of a boat *vs.* steering wheel of a car

Visibility

- *Relevant parts* of system should be *visible*
- If the user cannot see an important control, they would have to
 - guess that it exists, and
 - guess where it is
- Not usually a problem in the real world
 - Look at a bike or a pair of scissors
 - Hiding often takes effort (hidden doors)
 - Design can come in the way
- But takes extra effort in computer interfaces
 - Mouse clicks can be interpreted in arbitrary ways

Feedback

- *Feedback*: what the system does when you perform an action
- When the user successfully makes a part work, it should appear to respond
- Actions should have immediate, visible effects
 - Push buttons depress and release
 - Scrollbars move
 - Drag & drop following the cursor
- Kinds of feedback
 - Visual – see above
 - Audio – clicks made by keyboard (or, artificially, touch screens)
 - Haptic – vibrating touch screens, force feedback 3D-mouse

Modeling Human Error

- Description error
 - Intended action is replaced by another action with many features in common
 - Mitigation: Avoid actions with very similar descriptions
- Capture error
 - A sequence of actions is replaced by another sequence that starts the same way
 - Mitigation: Avoid habitual action sequences with common prefixes
- Mode error
 - User tries to invoke an action that doesn't have the desired effect in the current mode
 - Mitigation: Elimination, Visibility, spring-loaded or temporary modes, disjoint action sets

Metaphors

- Another way to address the model problem
- Advantage: borrowing a conceptual model the user already has experience with.
- Can convey a lot of knowledge about the interface model all at once
- Examples
 - Desktop
 - Trashcan
- Each of these metaphors carries along with it a lot of knowledge about the parts
 - purposes
 - interactions
- The user can draw on these to make guesses about how the interface will work.

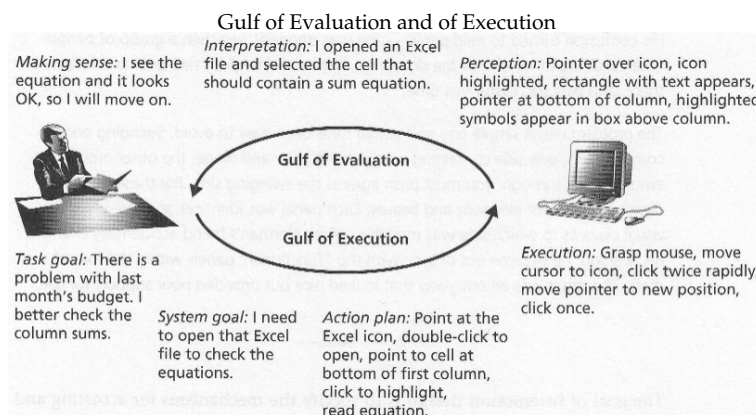
Levels

- Metaphors can be used on different levels
 - A metaphor for how the system works (Activity design in Scenario-Based Development)
 - * A discussion forum like a lecture or like the kitchen at a party
 - A metaphor for how information is displayed (Information Design)
 - * Free space left on hard disk as a partially full bar
 - A metaphor for the interaction offers (Interaction Design)
 - * Dragging a file into the waste paper basket

Stages of Interaction

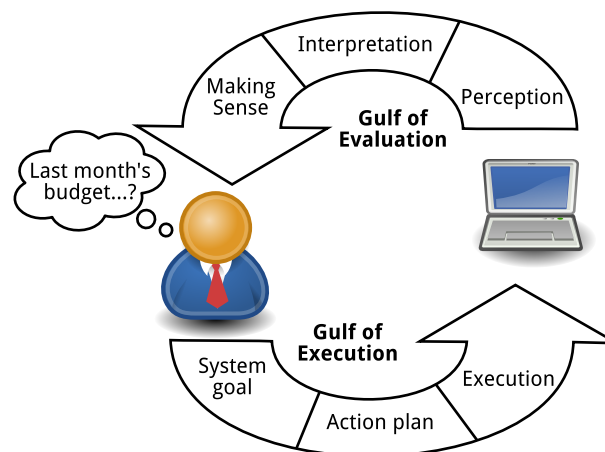
- There are lots of places where interaction between human and machine can go wrong
 - Perception
 - Cognition
 - Action
- Stages of action proposed by David Norman (1986)
- Two gaps
 - **Gulf of Evaluation:** the “cognitive distance” between what is displayed and the user’s mental representation
 - **Gulf of Execution:** distance between the user’s goals and the procedures and actions provided to pursue this goals

Norman’s Gulfs



Source: Rosson & Carroll, 2002

Gulf of Evaluation



9 Videos und Übungen

Visionen



Zum Beispiel: Corning: A Day Made of Glass 2, Where VR in 2025, Microsoft: Productivity Future Vision

Usability Prozesse: Google



Don't Listen to Users

26:11

Don't Listen to Users – Experience Sampling

- A systematic way of having participants provide samples of their ongoing behavior
- Participants record the behavior of interest (e.g., activity, location, mood, thoughts)
- Dependent on either
 - **Signal:** signaled with a beeper, cell phone call, or similar at random times within a fixed time period
 - **Interval:** pre-set intervals for reporting events
 - **Event:** whenever a key event occurs
- Advantages and disadvantages
 - reports are personal and subjective
 - data are similar to those obtained by diaries, but less dependent on memory
 - less intrusive than direct observation
 - data tend to not have the richness of ethnography

Seven deadly sins of user research



David Travis: The 7 deadly sins of user research

35:20

Seven deadly sins of user research (contd.)

- Credulity – Leichtgläubigkeit
 - “Asking the user what he wants and believing the answer”
- Dogmatism – Dogmatismus
 - “Believing that there is one right way to do user research”
- Bias – Befangenheit
 - Sampling bias, method bias, respons bias
- Obscurantism – Verdunkelung
 - “Keeping the finds of user research with UX team”
- Laziness – Faulheit
 - “Recycling old research data”
- Vagueness – Unklarheit
 - “Failing to focus on single research question”
- Hubris – Überheblichkeit
 - “Undue pride in your reports and deliverables”

Übungen


- Der in der Vorlesung beschriebene Prozeß des Contextual Designs soll an einem kleinen Beispiel durchgeführt werden.
- Dies soll in Gruppen von 3-6 Leuten geschehen.
- Aufgabenstellung
 - Suchen Sie sich eine Aufgabe aus, die Sie mittels des Contextual Designs analysieren und neu gestalten möchten.
 - Diese Aufgabe sollte eine genügende Komplexität haben, z.B. die Terminplanung von Gruppen, oder die Publikation von Nachrichten auf einem Webportal.
 - Vorschlag: Der Übungsbetrieb an der Universität, z.B. aus der Sicht der Studierenden, der Übungsgruppenleiter und vielleicht der Lehrenden.

10 Schluß

Veranstaltungen Wintersemester 2016/2017

- Medieninformatik (Medieninformatik 1), Vorlesung und Übung
 - Montag, 13:30-16:00 Uhr
 - Mittwoch, 12:30-14:00 Uhr
 - Beginn: Montag, 17.10.
- Medieninformatik Seminar
 - Vorbesprechung: Dienstag, 18.10., 14:00-16:00 Uhr
 - Bei Interesse gerne ✉ eine Mail
- Terminänderungen sind noch möglich! Bitte das LSF und das Learnweb beachten

Klausurtermin

- Erster Prüfungstermin
 - Freitag, 22.07.2016
 - 14:00-16:00 Uhr
 - A 9 (Großer Seminarraum Samelsonplatz)
- Zweiter Prüfungstermin
 - Falls eine Woche vor dem Klausurtermin Anmeldungen im POS vorhanden
 - Letzte Woche im September oder erste Woche im Oktober
 - Voraussichtlich:
 - * Mittwoch, 05.10.2016
 - * 16:00-18:00 Uhr
 - * A 9 (Großer Seminarraum Samelsonplatz)
 - Benachrichtigung über das Learnweb
- Anmeldung im POS bitte bis Dienstag, 19.07.
- Diejenigen, die sich nicht im POS anmelden können, schicken mir bitte bis zum gleichen Termin  eine **Mail**

11 Bildnachweis

Alle Abbildungen, wenn nicht anders angegeben, aus:
Malaka, Rainer; Butz, Andreas; Hussmann, Heinrich: *Medieninformatik – Eine Einführung*. ISBN 978-3-8273-7353-3, München: Pearson Studium, 2009.