

Contextual Design

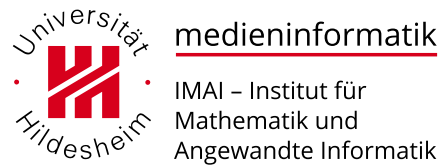
Teil V: Design der Benutzerumgebung

Jörg Cassens

Medieninformatik II

Contextual Design of Interactive Systems

SoSe 2016



1 Vorab

Übungen

- Wer nicht im Learnweb angemeldet ist soll das nachholen
 - Wichtig für die Eintragung der Bewertung zu den Übungen
- Abgabe von zwei der drei Übungen für die Zulassung zur Teilnahme an der Klausur notwendig
 - Zulassung in einem früheren Semester erworben:
 - * Klausur geschrieben $\implies$ automatisch zugelassen
 - * Klausur **nicht** geschrieben $\implies$ Mail an mich mit Angabe, wann Zulassung erworben wurde
- Abgaben in der Regel als PDF-Dateien
 - txt akzeptabel für reinen Text
 - SVG, PNG und JPG akzeptabel für Bilder/Graphiken
 - mp3, wav und ogg akzeptabel für Audio
 - mp4 akzeptabel für Video
 - Im Besonderen keine Dateien von Textverarbeitungen oder Graphikprogrammen (z.B. doc, docx, odt)
 - Abgaben in anderen Formaten werden nicht gewertet
 - Zusammenfassen in zip, gz, bz2 oder rar möglich

2 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

Probleme

- Technische Systeme sind immer auch eine Kristallisation von Arbeitsprozessen
- Verschiedene theoretische Modelle
 - Kulturell-historische Bedingung der Arbeitsgegenstände in der Tätigkeitstheorie
 - Einschreibung von Intentionen in der Aktor-Netzwerk-Theorie
- Herausforderung: Zusammenhängende Arbeitsprozesse durch zusammenhängende Artefakte unterstützen
- Zusammenhang ist nicht nur die Konsistenz der Benutzungsschnittstelle, sondern die Unterstützung des Ablaufs der Arbeit
 - Muß ich bei einer Präsentationssoftware zum Einfügen von Notizen eine neue Ansicht öffnen?

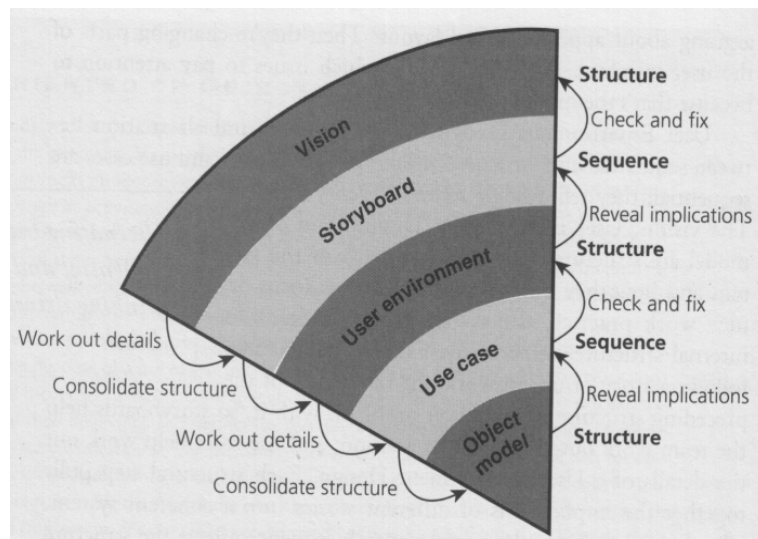
Probleme (contd.)

- Schwierig genug bei einzelnen Schritten, noch schwieriger bei komplexen Abläufen
- Bei fehlendem Zusammenhang muß der Benutzer die Teile zusammenleimen, das führt zu einer erhöhten Belastung
- Das Design von einfachen Speziallösungen ohne Zusammenhang schafft Probleme für den Benutzer
- Ein Fokus auf einfachen Speziallösungen schafft Probleme für die Entwickler (kein Überblick)

Aufteilung und Kohärenz

- Systematische, ingenieurmäßige Herangehensweise: Aufteilung des Problems
- Wir brauchen dann Werkzeuge um den Überblick zu behalten
 - Methoden in der Softwareentwicklung existieren
- Eine externe Repräsentation erhält den Fokus der einzelnen Entwickler
- Systeme schaffen ihren eigenen Zusammenhang, es ist besser diesen explizit zu machen
- Einzelprozesse müssen unterstützt werden → Wechsel zwischen Strukturmodellen und Arbeitssequenzen

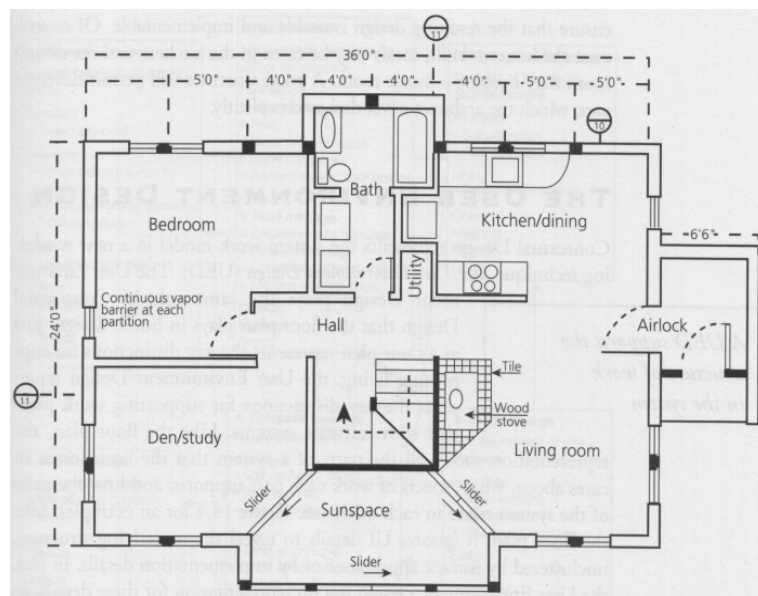
Strukturen und Sequenzen



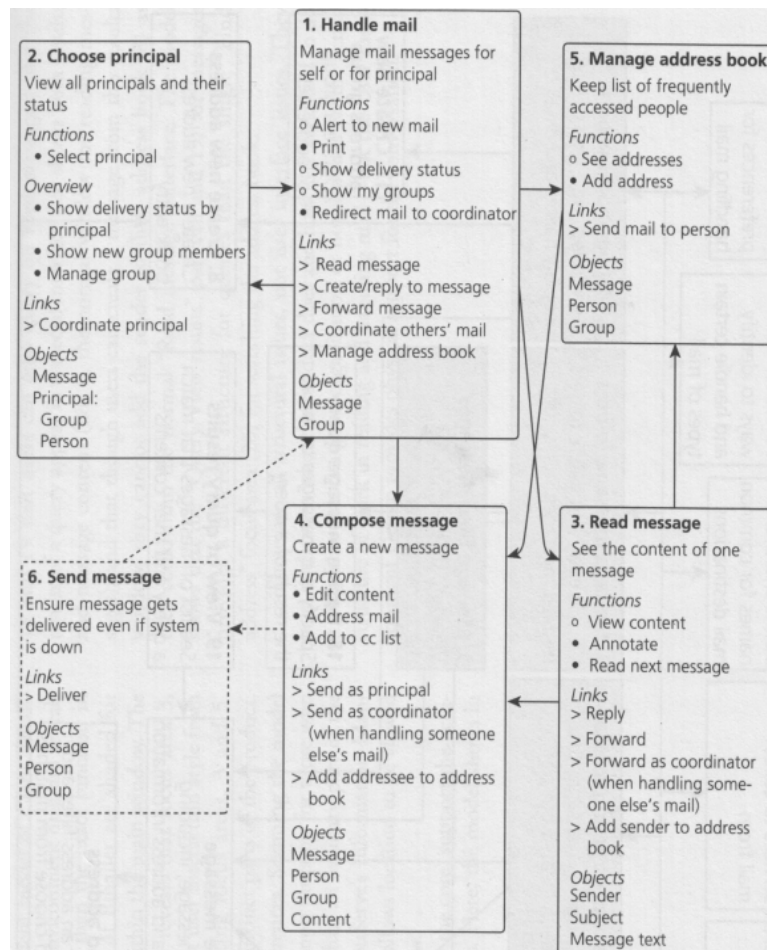
Aufteilung und Kohärenz (contd.)

- Die Struktur entscheidet wie gut der gesamte Arbeitsprozeß unterstützt wird
 - Orte an denen zusammenhängende Arbeiten erledigt werden können
 - Beachten, daß einzelne Änderungen das gesamte System betreffen können
- Das Design der Struktur kommt vor dem Design der Benutzungsschnittstelle
 - Activity Design im Scenario-Based Design
 - Hier: Benutzerumgebungen
 - * Metapher: “Raumplan” des Systems

Bauzeichnung



Benutzerumgebung



User Environment Design UED

3 Benutzerumgebung

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

3.1 Fokusbereiche

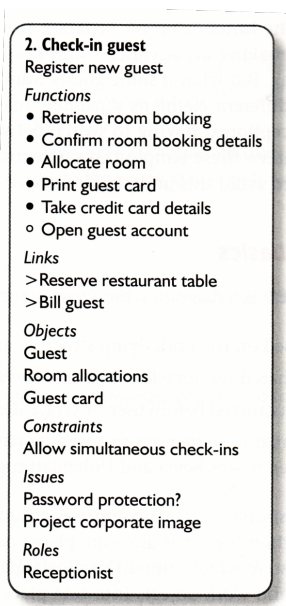
Fokusbereiche

- 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)



Zweck & Funktionen

- Der Zweck (Purpose) ist eine kurze textliche Beschreibung
- Die Funktionen (Functions)
 - haben eine kurze textliche Beschreibung
 - “●” – Funktionen des Benutzers
 - “○” – Funktionen des Systems (der Benutzer weiß, das sie existieren, ruft sie aber nicht explizit auf)
 - (Name) - Ein Name in Klammern für Funktionen die in mehreren Fokusbereichen auftauchen (solche Funktionen sind separat definiert)

Links

Links und doppelte Links weisen zu anderen Fokusbereichen

- “>” – Funktionen, welche Links zwischen Fokusbereichen unterstützen
 - Ein Pfeil repräsentiert den Link
 - Die Funktion muß nicht den gleichen Namen haben wie das Ziel des Links
 - * Dann sollte die Nummer des Zielbereiches in Klammern angegeben werden
- “>>” – Funktionen, welche gegenseitige Links zwischen Fokusbereichen symbolisieren
 - Eine doppelte Linie repräsentiert diese Art von Links

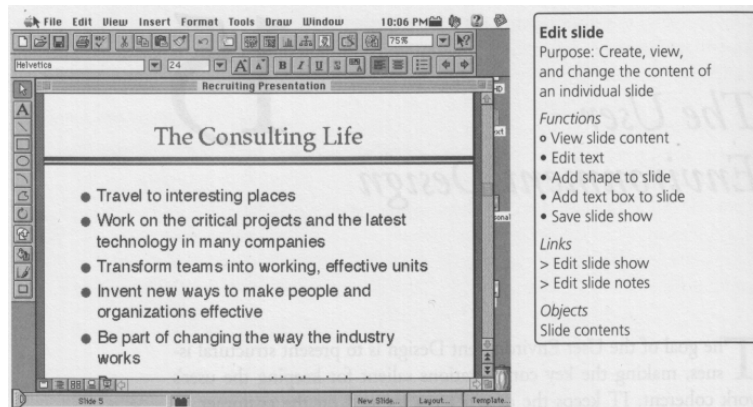
Weiteres

- Arbeitsobjekte (Work Objects)
 - Dinge, die der Benutzer in diesem Fokusbereich sehen und manipulieren kann
- Nebenbedingungen (Constraints)
 - Bedingungen an einen Fokusbereich
 - * Zeitverhalten
 - * Zuverlässigkeit
 - * Verfügbarkeit
 - * Form-Factor (Hardware)
 - * ...
- Probleme (Issues)
 - Offene Probleme im Zusammenhang mit diesem Fokusbereich
 - * Interface-Ideen
 - * Implementationsdetails
 - * Qualitätsanforderungen

Versteckt & Extern

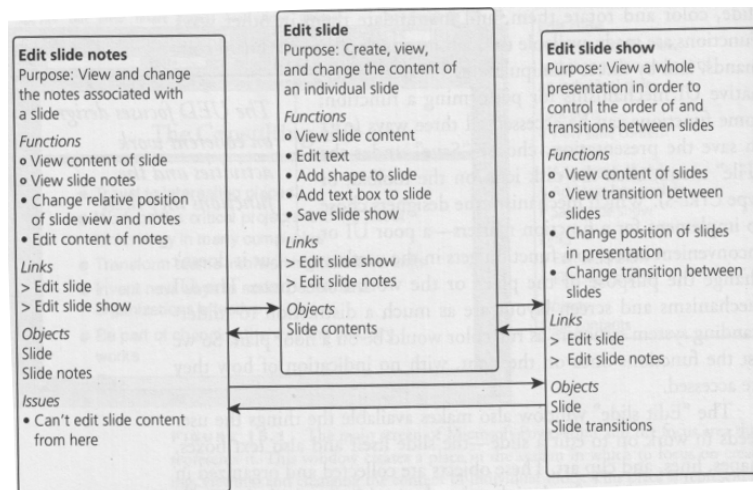
- Versteckte Fokusbereiche (Hidden focus areas)
 - Konzeptuelle Einheiten des Systems, die der Benutzer kennt und die wichtig sind, aber mit denen er nicht interagieren muß
 - Beispielsweise automatisierte Teilaufgaben, die vorher von einer anderen Person erledigt wurden
 - Repräsentiert als Kästen mit gestrichelter Umrandung
 - Mit anderen Fokusbereichen durch gestrichelte Linien verbunden
- Externe Fokusbereiche
 - Konzeptuelle Einheiten, die von anderen Teams geliefert werden
 - Zeigen wie das eigene System mit anderen zusammenarbeitet, um eine zusammenhängende Umgebung für den Benutzer zu schaffen

Beispiel 1



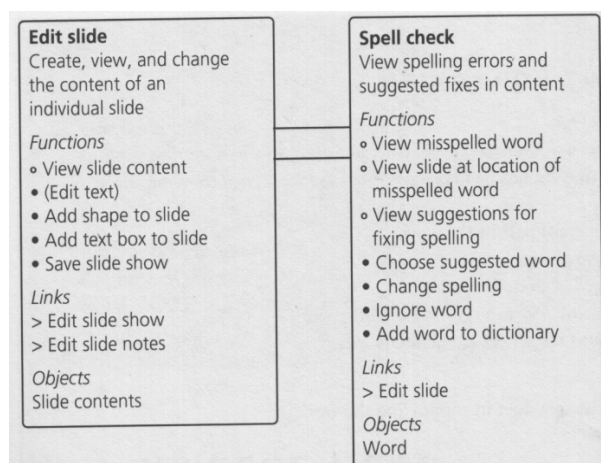
Hauptansicht von PowerPoint

Beispiel 2



Verlinkung

Beispiel 3



Doppelte Verlinkung – Abhängigkeiten

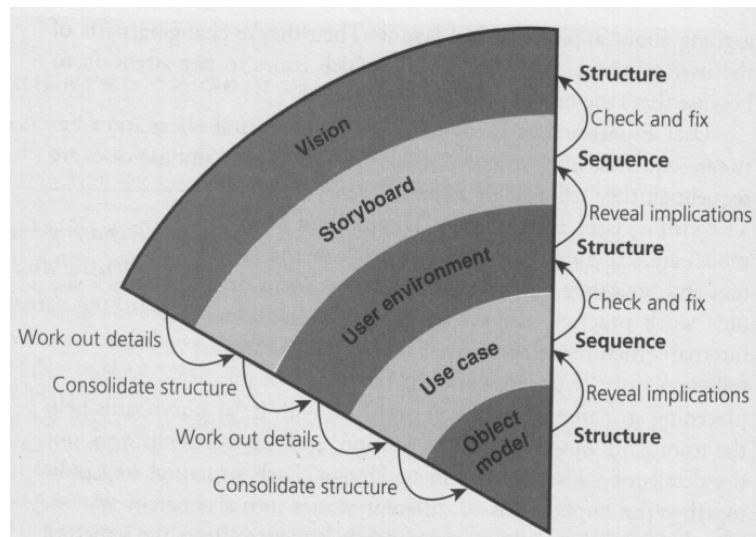
3.2 Methoden

Methoden

- 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

Iterationen



Storyboards

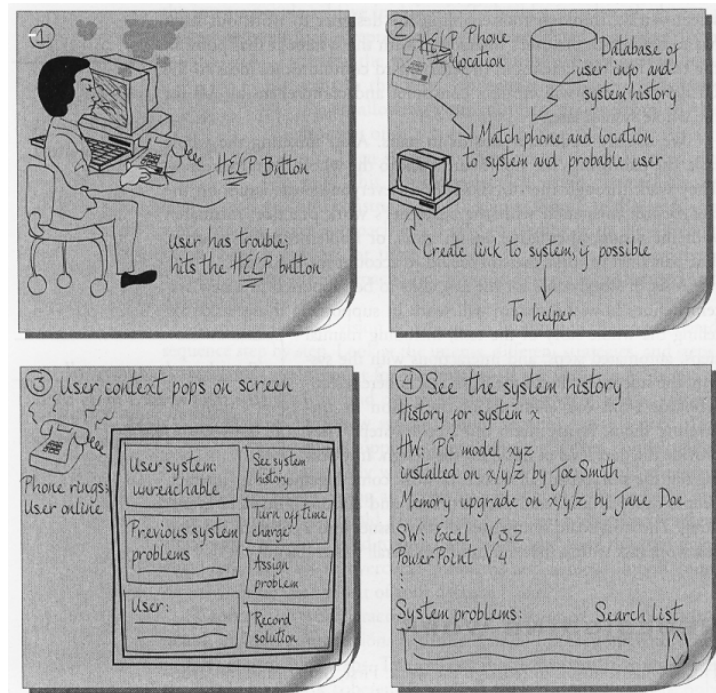
Vorgehen für Ableitung aus Storyboards:

- für jedes Storyboard wird eine Gruppe gebildet
- Gruppe spielt das Storyboard Schritt für Schritt durch
- für jeden Schritt, der eine Aktion mit dem System darstellt
 - ÜBERPRÜFE "Gibt es schon einen passenden Fokusbereich?"
 - WENN_JA
 - * Zweck und die Funktionen des Fokusbereichs überprüfen und gegebenenfalls erweitern
 - ANDERNFALLS
 - * Neuen Fokusbereich entwerfen und der Interaktionsstruktur hinzufügen
- Ergebnis: Repräsentation aller Storyboards in einer Struktur

Ziele

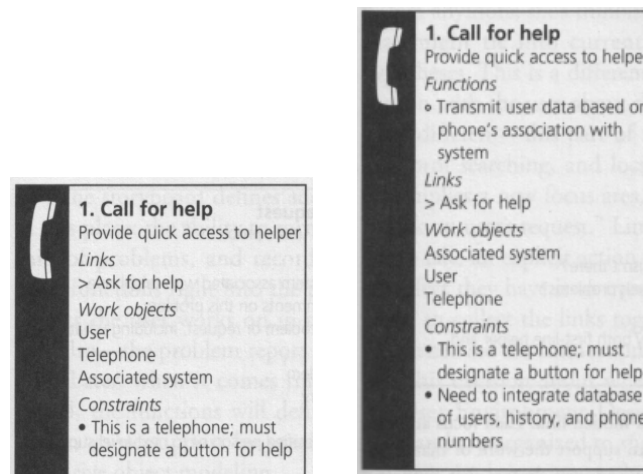
- Trennung von Storyboards und UED hilft, den Fokus zu behalten
 - Storyboards: "Ich sortiere meine Mail. Wie fange ich damit an? Was mache ich?"
 - UED: "Was passiert an diesem Ort wirklich? Unterstütze ich eine einzelne, zusammenhängende Tätigkeit? Wird alles angeboten, was der Benutzer hier braucht?"
- Storyboards implizieren die neue Systemstruktur
- Iteration zwischen beiden kann den Entwicklern helfen, Strukturen zu sehen, ohne den Arbeitsablauf aus den Augen zu verlieren
- UED kann auch Hardware repräsentieren, mit der der Benutzer interagiert

Beispiel 1



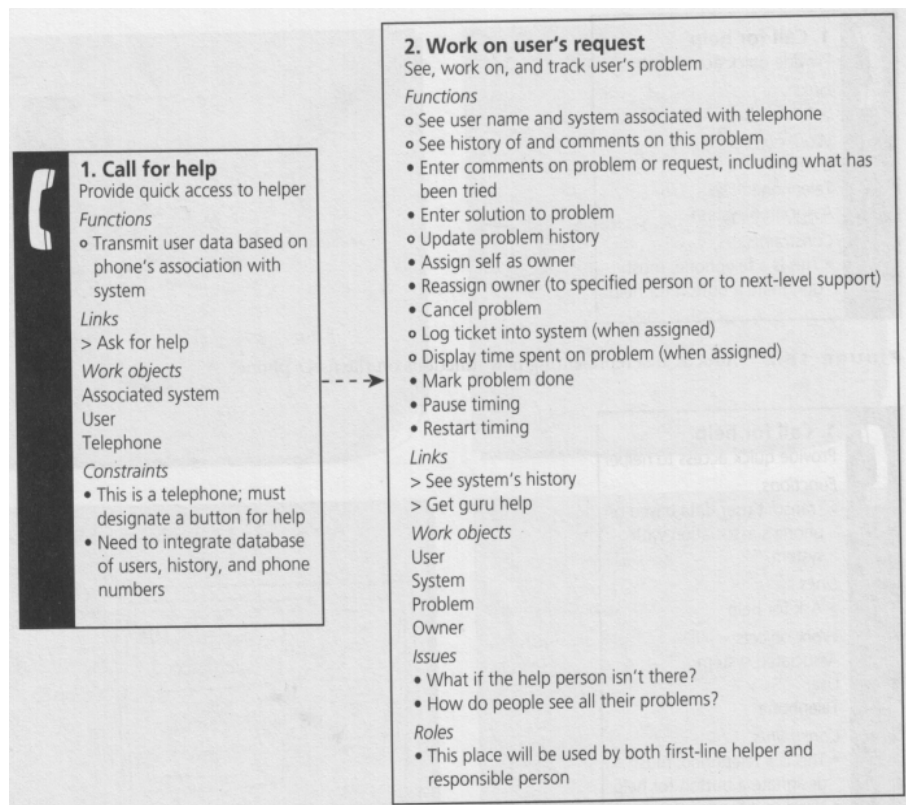
Abarbeitung eines Hilferufs zur Vision System Management

Beispiel 2

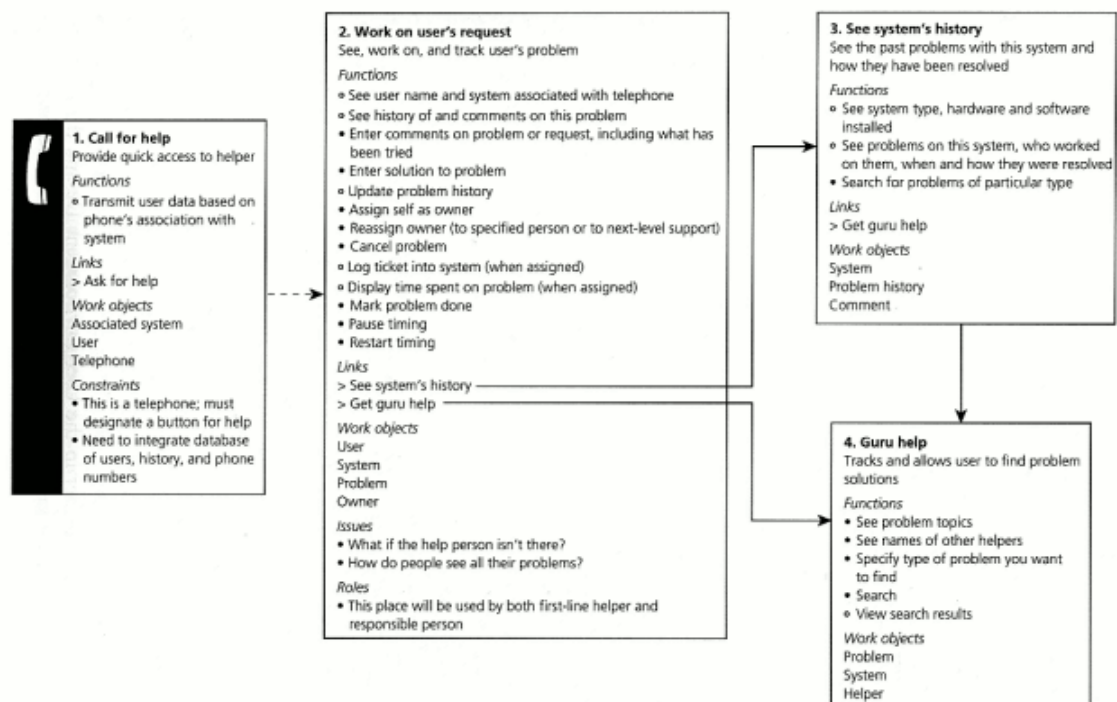


Erweiterung eines Fokusbereichs

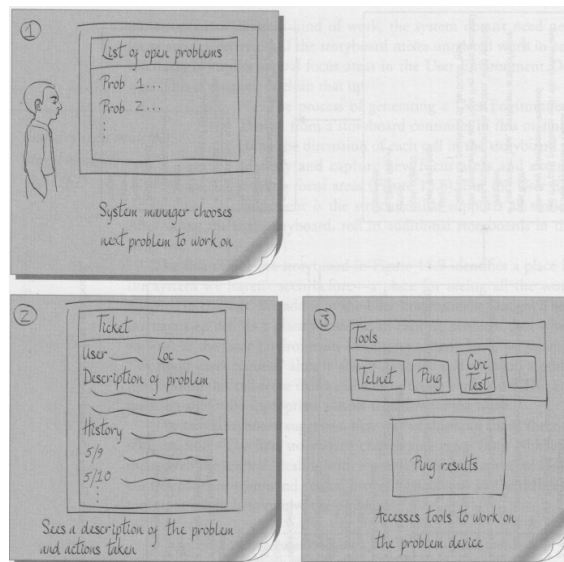
Versteckt



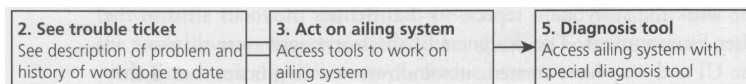
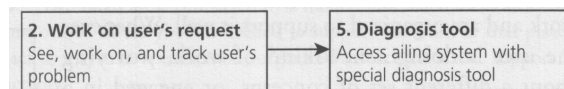
Kombination



Alternativen



Zusammenführung



UED Walkthrough

- Letzter Schritt: Fokusbereiche aufräumen
- Überprüfung mit Walkthroughs (im Team)
 - identifizieren von überlappenden Fokusbereichen
 - identifizieren von unnötigen Funktionen
 - identifizieren von Fokusbereichen, die nur Links beinhalten (hallways)
 - vereinfachen von Fokusbereichen mit mehr als einer Aufgabe
- Abschließende Tests im UED Walkthrough:
 - unterstützen Fokusbereiche reale Arbeitsaufgaben?
 - unterstützen Fokusbereiche alle Anwender?
- Hintergrund:
 - Einigung auf gemeinsames Big Picture
 - Benutzerumgebung (UE) als Grundlage für UI etablieren

3.3 Resultat

Ergebnisse und Nutzen des UED

- Das Team kann
 - *überprüfen*, ob die Struktur für die Benutzer angemessen ist
 - *planen*, in welcher Reihenfolge Features implementiert werden sollen (Priorisierung)
 - die Aufgaben des Projekts entsprechend der Struktur auf Entwicklungsteams *aufteilen*

- das “Was” (Funktionen und Arbeitsobjekte) vom “Wie” (Interaktionsstil, Darstellung von Informationen, Interaktionsangebote) *trennen*.
 - * d.h. das Design-Team kann sich beim Design der Benutzungsschnittstelle auf das “Wie” konzentrieren.

Benutzerumgebung & User Interface Design

- Benutzerumgebung liegt vor... what's next?
- Auswahl **Plattform-Typ** (Standalone, Client/Server, Webservice, ...) und **Interaktionsstil** (Maus, Tastatur, Spracheingabe, ...)
 - werden primär aus dem physischen Modell abgeleitet
- Zuordnung der angemessenen Interaktionselemente
 - werden aus Fokusbereichen der Benutzerumgebung bestimmt
 - Rückgriff auf User Interface Standards (Styleguides) und Usability-Prinzipien (z.B. ISO 9241) der gewählten Plattform
- Entwerfen der Benutzungsschnittstelle in Gruppen:
 - jede Gruppe entwirft einen Teil der Interaktionsstruktur (z.B. einen Fokusbereich)
 - skizziert in einem Brainstorming mehrere alternative Entwürfe
 - besten Entwurf ermitteln und Papier-Prototyp bauen
- Abschließend im Gesamtteam:
 - Diskussion der Papier-Prototypen
 - Beseitigen von Inkonsistenzen bzw. Problemen

4 Projektplanung und Strategie

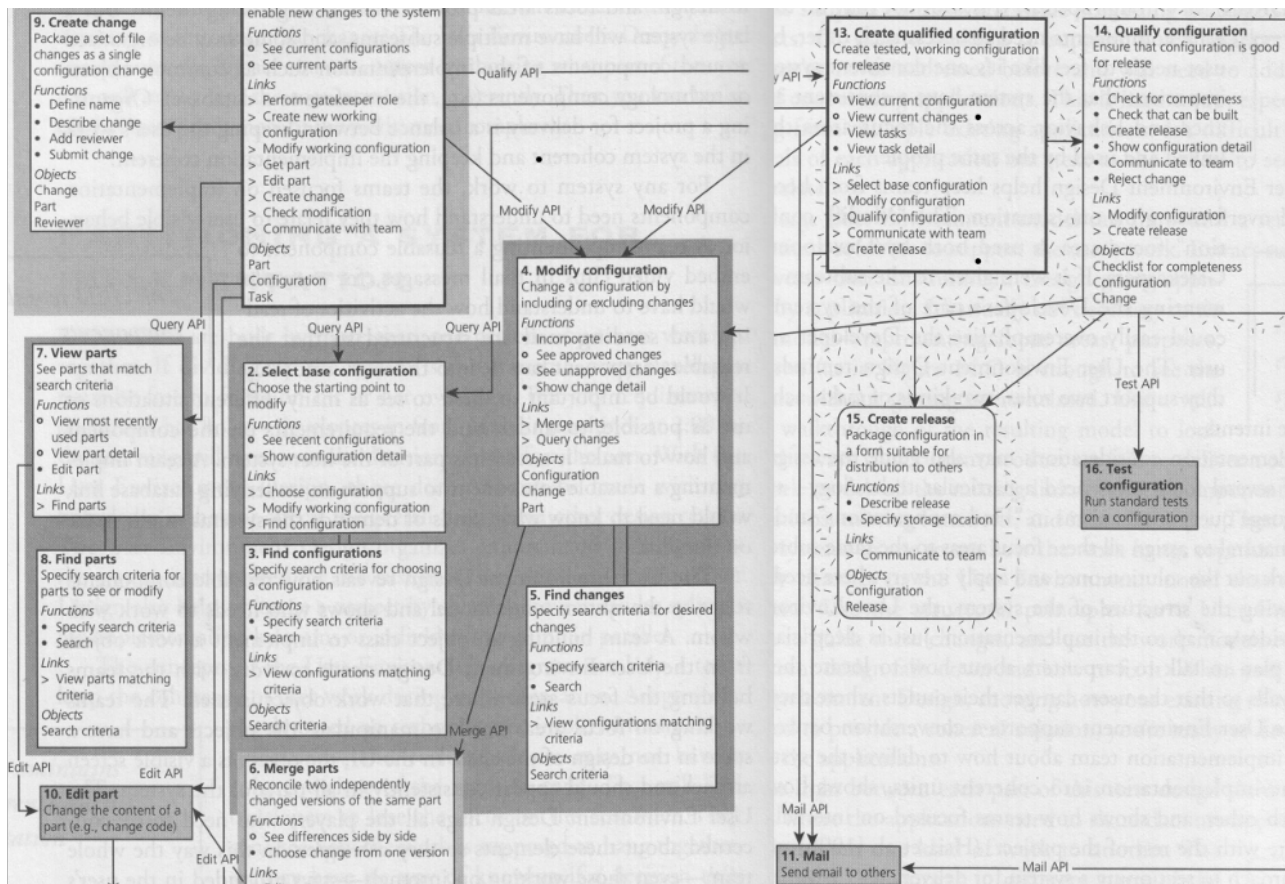
Überblick

- UED zeigen alle Teile des Systems in Beziehung
- Daher nicht nur Hilfe für Design, auch für Planung/Management
- Große Systeme
 - Mehrere Releases notwendig zur Lieferung der gesamten Funktionalität
 - Müssen in der Regel mit anderen Systemen zusammenarbeiten
- Releaseplanung: Kohärent (für den Benutzer) und realisierbar (mit den Ressourcen der Entwickler)
- UED die das Gesamtsystem zeigen helfen bei der Unterteilung

Auswahlkriterien

- Fokusbereiche, die zusammen einen Arbeitsprozeß unterstützen, gehören in einen Release
 - Entweder zusammen ausliefern oder keines
- Rollen, Verantwortlichkeiten, Aufgaben
 - Die Hauptaufgabe einer Rolle unterstützen, von den anderen Bereichen so viel, wie dafür notwendig ist
- Nach der Entscheidung was zum Release gehört:
 - Das “shipping User Environment Design” zeigt nur die Fokusbereiche und Funktionen, die Teil des Releases sein sollen

Beispiel



Partitionierung

- UED helfen den Entwicklerteams einen gemeinsamen Fokus zu behalten
- “Raumplan” – bei einzelnen Entscheidungen kann der Entwickler immer die Übersicht ansehen und einschätzen wie seine Entscheidungen in den großen Rahmen passen
- UED helfen wiederverwendbare Teile zu identifizieren
 - UED zeigen wie die wiederverwendbaren Teil im Gesamtsystem angeordnet sind, welche Aufgaben damit zusammen hängen
- Links zwischen Fokusbereichen welche verschiedenen Teams zugeordnet sind zeigen Integrationsziele
 - Nutzbar für die Definition von APIs

Produktstrategie

- Reverse UED zeigen wie existierende, nicht-integrierte Systeme zusammen gebracht werden können
- Walkthroughs durch das Model helfen dabei problematische Bereiche zu identifizieren
- UED können auch bei der Entscheidung helfen welche externen Komponenten eingekauft werden sollten
- UED zeigen wo die Grenzen des eigenen Bereiches liegen, sie könne dabei helfen die Notwendigkeit von Brücken zu exemplifizieren

Koordination

- Konzeption
 - UED repräsentieren ein Modell des Systems und können zur Planung von Geschäftsprozessen dienen
 - Neue Prozesse können in Walkthroughs getestet werden

- Implementation
 - Separation zwischen UI-spezifischem und generellem Code
- Dokumentation
 - Die Fokusbereiche strukturieren die Dokumentation und geben Hilfestellung bei der Frage, für wen die Doku zu schreiben ist
 - Motivation der einzelnen behandelten Teile
 - Zeigt weitere Optionen auf – Training, Helpdesk
- Tests
 - Storyboards + UED = Startpunkt für Testpläne
 - * Storyboards – wie soll es funktionieren
 - * UED – welche Funktionen werden angeboten

5 Bildnachweis

Alle Abbildungen, wenn nicht anders angegeben, aus:

Hugh Beyer and Karen Holtzblatt: *Contextual Design: Defining Customer-Centered Systems (Interactive Technologies)*. Morgan Kaufmann, 1997.