

# Designprozesse

Jörg Cassens

## Medieninformatik II Contextual Design of Interactive Systems SoSe 2016



### Vorab

#### Orte und Zeiten

- Vorlesung und Übung jeweils am Montag
  - Montag, 13:30-15:00 Uhr (VL)
  - Montag, 15:15-16:00 Uhr (Ü)
- Samelsonplatz A 102
- Keine Veranstaltungen in der Exkursions- und Projektwoche (17.05.2016-20.05.2016)

werden

- Prüfungstermine (voraussichtlich)
  - Freitag, 22.07.2016
  - Mittwoch, 05.10.2016

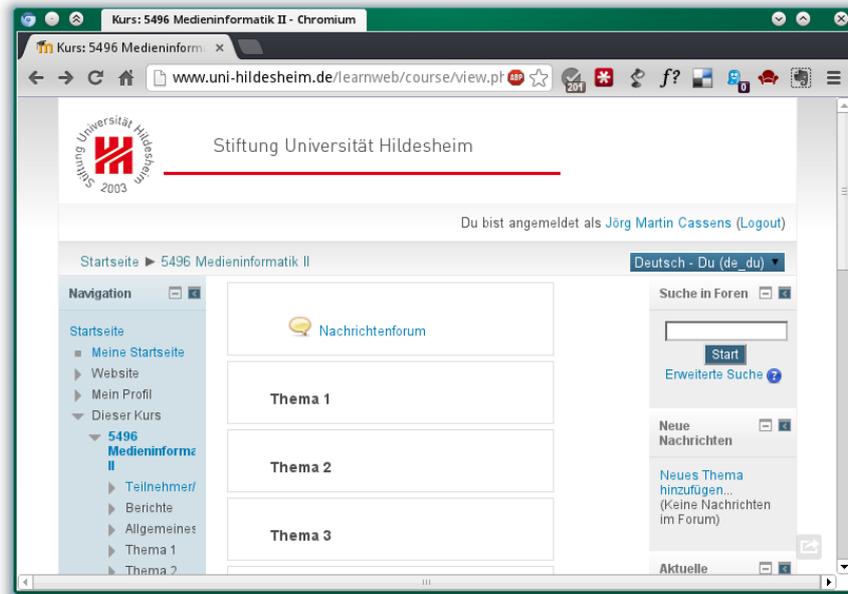
### LSF



The screenshot shows a web browser window with the URL <https://lsf.uni-hildesheim.de/qisserver/rds?state=verpubl>. The page title is 'Medieninformatik I - Einzelansicht'. The main content area displays a table of 'Grunddaten' (Basic Data) for the course.

Veranstaltungsart	Vorlesung mit Übung	Co-finanziert aus Studienbeiträgen	
Veranstaltungsnummer	5495	Kurztext	
Semester	WiSe 2012/13	SWS	5
Erwartete Teilnehmer	20	Max. Teilnehmer	
Rhythmus	i. d. R. jedes 2. Semester	Studienjahr / Zielgruppe	
Credits		Anmeldung	Anmeldepflicht, Vergabe: Manuell (?)

## Moodle



[learnweb.uni-hildesheim.de](http://learnweb.uni-hildesheim.de)

## Dropbox



[v.gd/medieninformatik\\_2](https://www.dropbox.com/v.gd/medieninformatik_2)

## Praktikum Medieninformatik

- Problem-Based Learning
- Ein Projekt soll von der Anforderungsanalyse bis zum Produkt realisiert werden
  - Eine größere Aufgabe, die bis ans Ende der nachfolgenden vorlesungsfreien Zeit beendet werden muß
  - In Gruppen, Gruppengröße nach Art und Umfang des Projekts, in der Regel 3-5 Teilnehmende
- Während der Vorlesungszeit
  - Mittwoch, 16:00-18:00 Uhr (s.t.)
  - Samelsonplatz A 102
- In der vorlesungsfreien Zeit
  - Nach Vereinbarung

## Einleitung

- Entwicklung und Gestaltung digitaler Medien können bei größeren Projekten sehr aufwendig werden
- Bei kleineren Projekten mit einem oder wenigen Entwicklern kann eine ad-hoc Herangehensweise funktionieren

- Umfangreichere Produktionen erfordern Kommunikation und Koordination
- Klassisches Problem der Softwareentwicklung
- Mischung aus unterschiedlichen Bereichen
  - Softwaredesign
  - Mediendesign
  - Interaktionsdesign
- Fokus vor allem auf:
  - Iterativen Modellen
  - Benutzerzentrierten Modellen

### Lernziele

- Einblick in die Notwendigkeit von Designprozessen
- Besondere Herausforderungen bei der Entwicklung von digitalen Medien
- Verschiedene Prozeßmodelle und ihre Vor- und Nachteile
- Herausforderungen unterschiedlich für
  - Interaktive Programme
  - Spiele
  - Digitale Filme
- daher kommt es darauf an, ein für das jeweilige Projekt geeignete Vorgehensmodell zu finden und umzusetzen

## 1 Designprozesse

### Ziele

- Ganz allgemein sollte ein Designprozeß
  - zielgerichtet sein
  - sich an geplantem Nutzen, Material, Kosten und Nutzbarkeit orientieren
  - Kreativität ermöglichen
  - Kosten- und Zielplanung enthalten und
  - nachvollziehbar sein (warum wurde was wie gemacht?)
- Dazu muß es Pläne und Vorgaben geben
  - Wo steht man?
  - Was fehlt noch?
- Flexibilität und alternative Entwürfe sollten möglich sein

### Fallstudie

- Eine Medienagentur soll eine interaktive Präsentation erstellen, die das Produktportfolio einer Firma darstellt
- Sowohl in einfacher Form für das Web wie in aufwendiger Form für eine DVD
- Fragen sind unter anderem:
  - Anhand welcher Kriterien kann man später feststellen, ob das Ziel erreicht wurde und ob das Produkt gut ist?
  - Wen soll das Produkt ansprechen?
  - Welche Technologien und Werkzeuge werden eingesetzt?

- Kann eine geeignete Architektur die Anpassung an zukünftige Entwicklungen erleichtern (neue Produkte)
- Wie kann man Gemeinsamkeiten von Web, DVD ausnutzen?
- Welches graphisches Design wird zugrundegelegt?
- Wie soll die Interaktion mit dem Benutzer stattfinden?
- In welcher Reihenfolge finden die Entwicklung statt?
- Welche Möglichkeiten bieten das geplante Budget und die vorhandenen Ressourcen?

## Medien Engineering

- Wichtig ist, die spezifischen Prozesse in der Medieninformatik mit den allgemeine Entwurfsprozessen der Informatik zu vereinen
- Verschiedene Prozesse und Vorgehensweisen des Softwaredesigns unterschiedlich geeignet für die Einbindung anderer Prozesse
- Medieninformatiker mediieren zwischen den verschiedenen beteiligten Gruppen
  - Wir müssen die Modelle des Software Engineerings kennen und einschätzen lernen
  - Wir müssen die Vorgehensweisen der Designer kennen und unterstützen
  - Das Interaktionsdesign ist ein Hauptgebiet in der Medieninformatik, und wir kennen verschiedene Vorgehensmodelle und Methoden und können diese anwenden

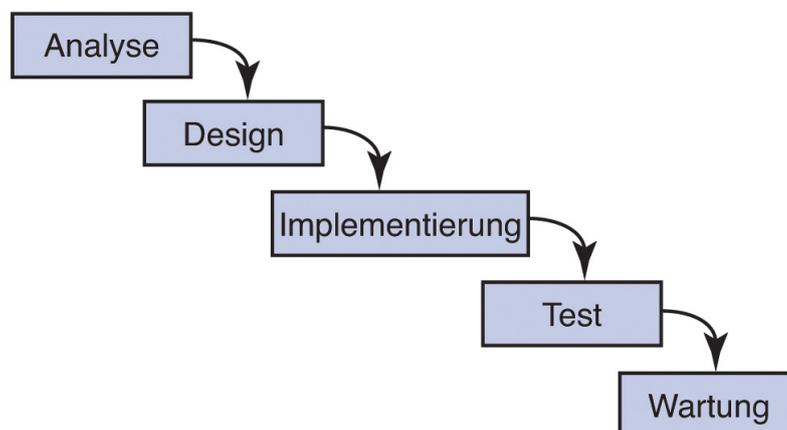
## 2 Klassische Modelle

### 2.1 Wasserfallmodell

#### Wasserfallmodell

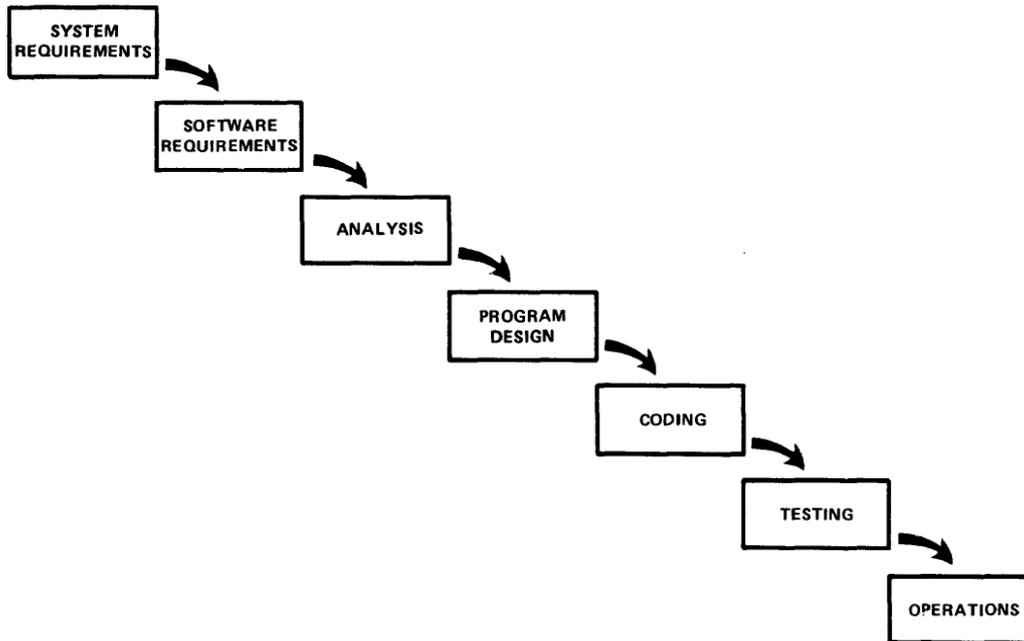
- Klassisches Modell zur Entwicklung von Produkten und zur Beschreibung des Lebenszyklus eines Produktes ist das Wasserfallmodell
- Modell geht von klar definierten Phasen aus, die hintereinander durchlaufen werden
- Man sieht, das Design und Implementierung nur einen Teil der zu erledigenden Arbeiten ausmachen
  - Analyse: Verstehen, was die Aufgabe ist
    - \* Anforderungsanalyse (Requirements Engineering)
  - Prozeß nach der Implementierung nicht abgeschlossen
    - \* Test und Wartung
- Zahlreiche Varianten
  - Oft z.B. Aufteilung in Grobspezifikation und Feinspezifikation

#### Wasserfallmodell Vereinfacht



**Abbildung 12.1:** Klassisches „Wasserfallmodell“

## Wasserfallmodell



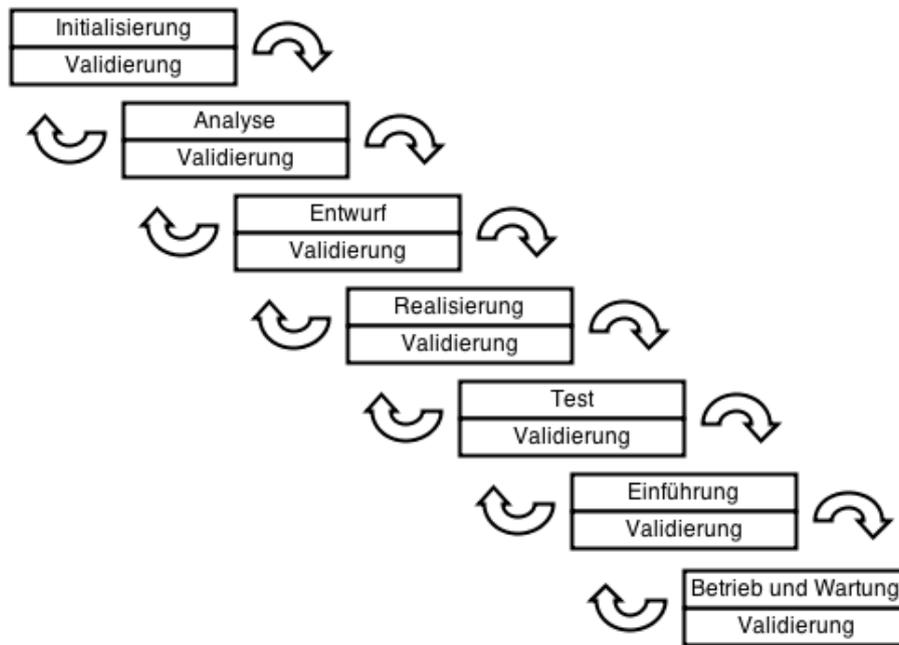
“In my experience, ... the simpler model has never worked on large software development efforts ...” – Royce [1970]

### Vor- und Nachteile

- Vorteile
  - Klare Abgrenzung in einzelne Phasen
  - Einfache Möglichkeit von Planung und Kontrolle
  - Abschätzungen von Kosten und Umfang
- Nachteile
  - Nicht skalierbar
  - Unflexible Anforderungen
  - Relativ späte Umsetzung
  - Späte Erkennung falscher Konzeptionen

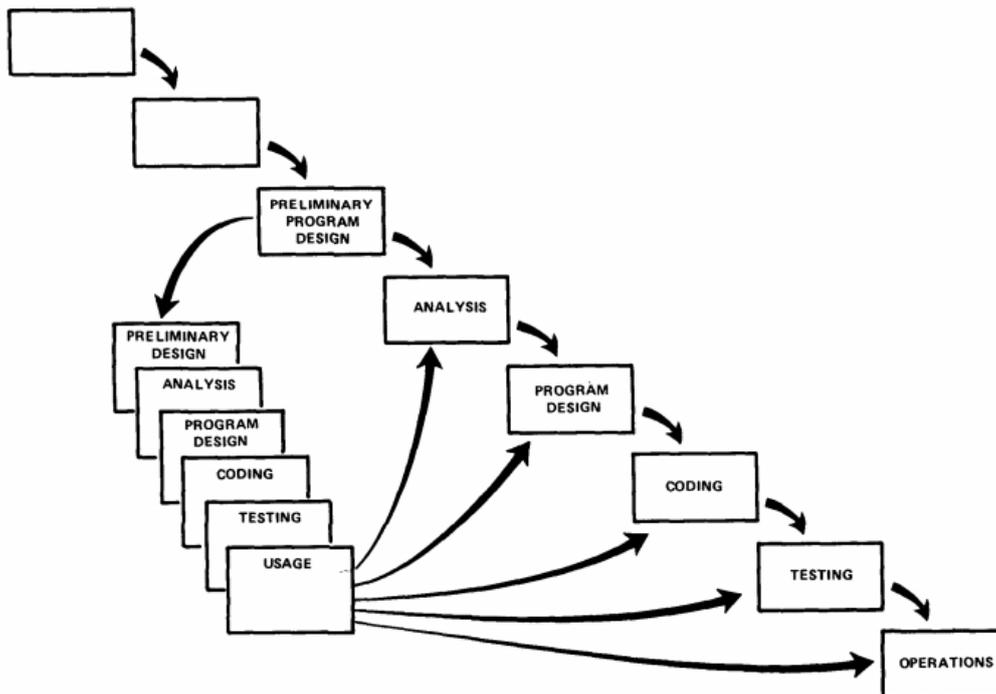
Variante: Wasserfall mit Rückwärtsprüfungen

### Wasserfallmodell – Rücksprung



Wasserfall mit Rücksprung nach Boehm

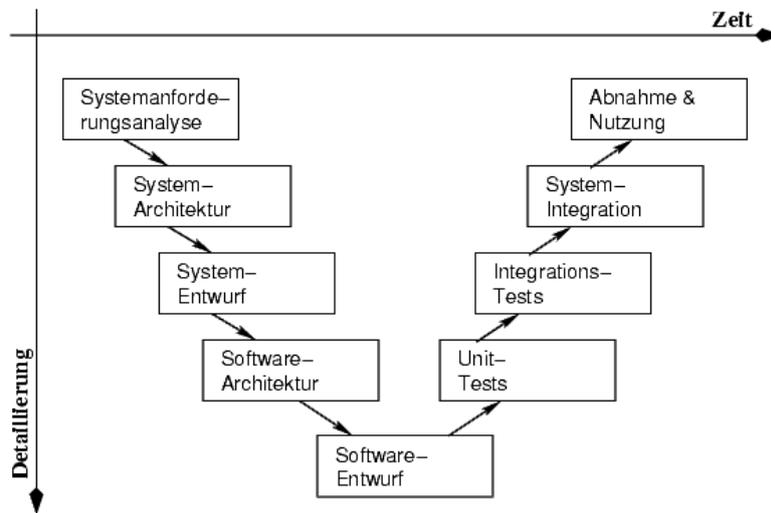
Wasserfallmodell – Do it twice



Royce [1970]

## 2.2 V-Modell

V-Modell



## V-Modell

- Leitfaden zum Planen und Durchführen von Entwicklungsprojekten
- Berücksichtigung des gesamten Systemlebenszyklus
- Definiert Ergebnisse und beschreibt Vorgehensweisen
- "Wer" hat "Wann" "Was" in einem Projekt zu tun

Unter ständiger Weiterentwicklung und unter der Apache Lizenz zur Nutzung freigegeben.

Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung (KBSt)

## Fazit

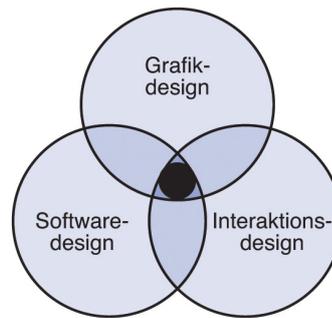
### Fazit

- Warum interessieren uns die klassischen Modelle überhaupt?
- Selbst wenn sie nicht in sequentiellen Phasen auftauchen sind bestimmte Aktivitäten in den meisten Modellen mehr oder weniger offensichtlich wiederzufinden:
  - Analyse – Verstehen der Welt
  - Konzeption – Design einer Lösung
  - Realisierung – Umsetzung dieser Lösung
  - Evaluation – Test dieser Lösung
- Bei der Entwicklung medieninformatischer Lösungen werden uns diese Aktivitäten wieder begegnen

## 3 Ebenen

### Ebenen des Designs

- Softwaredesign
- Graphik- und Audiodesign
- Interaktionsdesign



**Abbildung 12.2:** Überlappende Designaufgaben verschiedener Art

## Herausforderung

- Klassische Softwareprodukte: Schwerpunkt der Entwicklungsarbeit auf dem Softwaredesign
  - Softwarearchitekturen
  - Programmstrukturen
  - Datenmodell
- Bei Projekten aus dem Medienbereich dominieren
  - Graphikdesign
  - Audiodesign
  - Andere medientypische Designentscheidungen (z.B. Drehbuch bei Videos)
- Interaktionsdesign spielt gleichfalls eine größere Rolle
  - Aspekte des Usability Engineering in allen interaktiven Anwendungen
- Erweiterte Komplexität durch parallele, von Spezialisten durchgeführte Entwurfsprozesse

## Komplexität

- Die Entwicklung digitaler Medienprodukte ist komplexer, da zumindest die drei genannten Bereiche miteinander koordiniert werden müssen
- Beispiel
  - Ein Graphikdesigner entwickelt die Farb- und Formensprache der Präsentation
  - Ein Filmteam erstellt Videoclips
  - Ein Interaktionsdesigner erstellt ein Konzept für Systemmetaphern und legt die Menüstruktur fest
  - Ein Softwareentwickler arbeitet an einer modularen und flexiblen Architektur
- Entsprechend werden in den einzelnen Phasen diese Aktivitäten parallel, aber nicht unabhängig voneinander durchgeführt

## Wasserfallmodell mit Ebenen

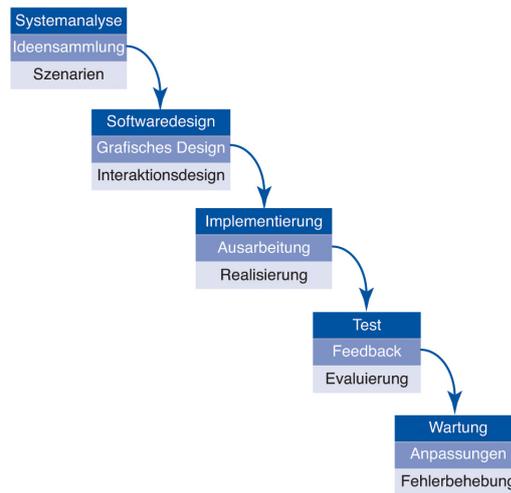


Abbildung 12.3: Wasserfallmodell mit drei parallelen Aktivitätsarten

## Fazit

### Fazit

- Bei medieninformatischen Projekten sorgt die Integration verschiedener, teilweise parallel verlaufender, Entwicklungsprozesse für erhöhte Komplexität
- Gesucht sind Prozesse, die diese Komplexität handhabbar machen
  - Vergleiche unser Anfangsstatement zur Notwendigkeit von Entwicklungsprozessen
- In einem ersten Schritt haben wir die medieninformatischen Anforderungen als parallele Prozesse im Wasserfallmodell skizziert
- Noch nicht betrachtet sind Kernkompetenzen der Medieninformatik
  - Usability (UI)
  - User Experience (UX)

## 4 Iterativ

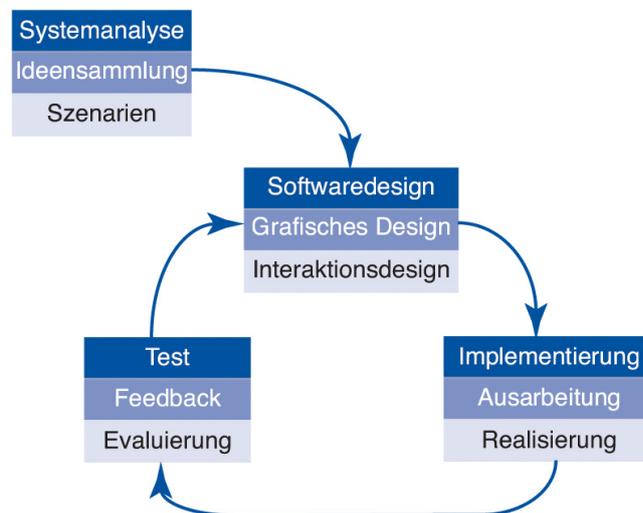
### Praxis

- Entwicklungsprozesse laufen meist nicht so einfach und geradlinig ab, wie in den Modellen beschrieben
- Änderungen in den Anforderungen oder schwer formalisierbare Anforderungen machen ein "wasserfallartiges" Vorgehen schwierig oder unmöglich
- Im schlimmsten Fall wird an den Bedürfnissen der Benutzer vorbei ein Produkt zu Ende entwickelt, welches in der Praxis nicht einsetzbar ist
- Im Software-Engineering werden Zwischenergebnisse entwickelt, die in einem frühen Stadium bereits von der Auftraggebern abgenommen werden
- Viele verschiedenen Modelle:
  - Spiralmodell
  - Unified Process
  - Agile Modelle
- Ganz allgemein für die Softwareentwicklung, ohne daß es Medienprodukte im engeren Sinne sein müssen

## Benutzerzentriertes Design

- Auch bei der Entwicklung von Interaktionskonzepten wurden ähnliche Erfahrung gemacht
- Benutzbarkeit von Schnittstellen am besten sicherzustellen, wenn man sich frühzeitig an den Bedürfnissen und Fähigkeiten der späteren Nutzer orientiert
- User-Centered Design (klassisch: Benutzerzentrierter Entwurf, heute: Menschzentrierter Entwurf)
  - Iterative Variante eines Entwurfsprozesses, welche den Nutzer in das Zentrum stellt
  - Sehr frühe Einbindung der Nutzer
    - \* Feldstudien und Benutzerbefragungen
    - \* (Papier-) Prototypen
    - \* Formative Evaluation von Zwischenergebnissen
- Im kreativen Design sind formale Modelle eher unüblich, aber auch hier sind iterative Modelle üblich, bei denen Entwürfe in Zusammenarbeit mit Kunden weiterentwickelt werden

## Iterative Vorgehensmodelle



**Abbildung 12.4:** Iteratives Modell mit drei parallelen Aktivitätsarten

## Spiralmodell



- Analyse
  - Konzeption
  - Realisierung
  - Evaluation
- Im nächsten Schritt stellen wir uns die Frage, was es heißt, den Menschen in das Zentrum der Betrachtungen zu stellen

## 5 Menschzentrierte Prozesse

### Menschzentrierte Prozesse

Ein menschzentrierter 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.

### Fokus

- Menschzentrierte 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

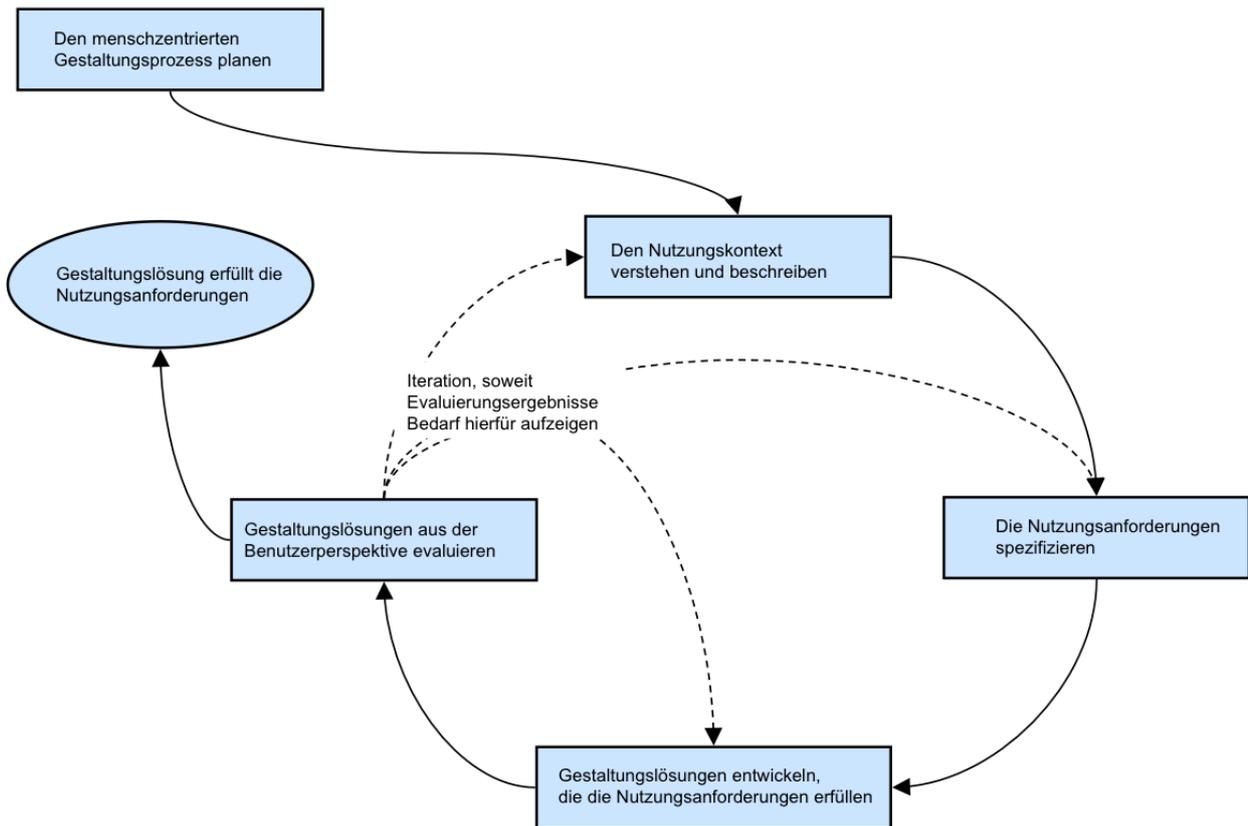


Bild 1 aus DIN EN ISO 9241-210: Wechselseitige Abhängigkeit menschzentrierter Gestaltungsaktivitäten

### Beispiel: Sternmodell

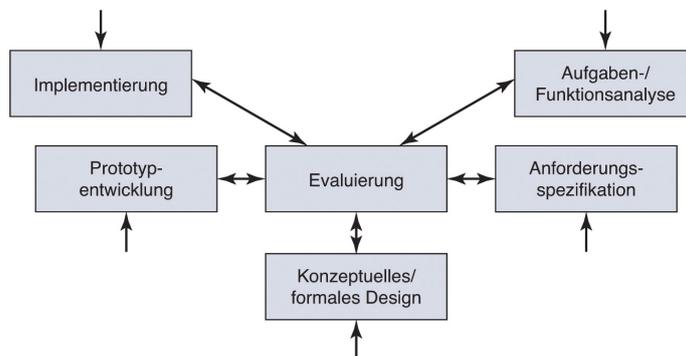


Abbildung 12.5: Beispiel für einen menschenzentrierten Designprozess – Stern-Modell von Hartson und Hix (1989)

### 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

## 5.1 Analyse

### 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

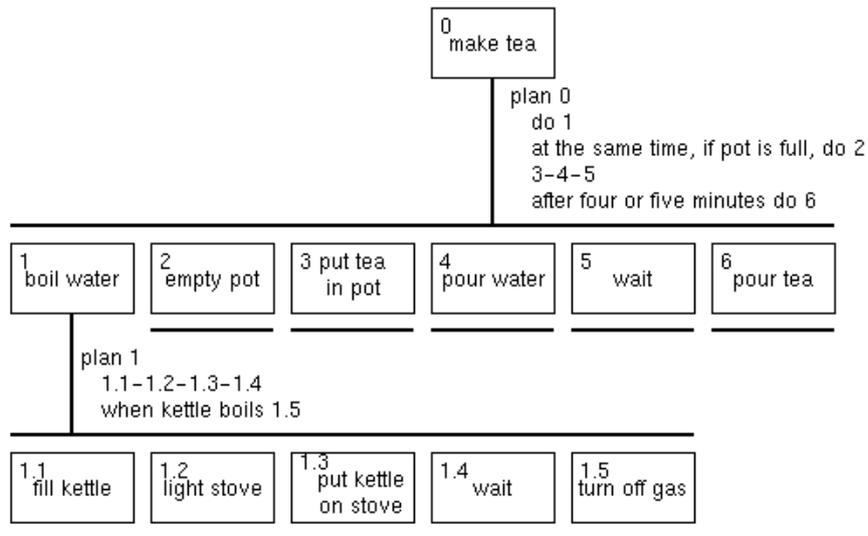
### Benutzeranalyse

- 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)

## Artefakte



cc by-nc-sa Ian Lewis

## 5.2 Konzeption

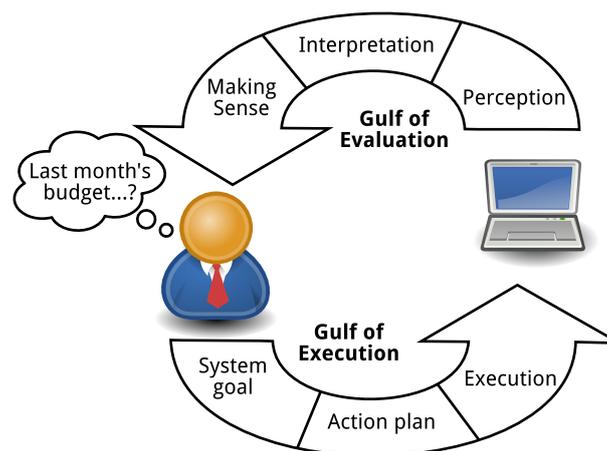
### Aktivitätsdesign

- 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

### Mensch-Computer Interaktion

- Bei der Interaktion von Menschen mit Computern müssen in beide Richtungen verschiedene Schritte durchgeführt werden, damit die Interaktion erfolgreich ist
  - Wir denken an das Ziel unserer Arbeit und müssen dieses erste übersetzen in die Nutzung von Applikationen
  - Wir müssen die vom Computer angezeigten Daten erst in den Kontext unserer Ziele stellen, um sie interpretieren zu können
- Donald Norman spricht in dieser Hinsicht vom "Gulf of Execution" und dem "Gulf of Evaluation" (Ausführungskluft und Evaluationskluft)
- Gesucht werden Prozesse, die dabei helfen, diese Klüfte zu überwinden
  - Das Informationsdesign beschäftigt sich damit, wie die Evaluationskluft überbrückt werden kann
  - Das Interaktionsdesign beschäftigt sich damit, wie die Ausführungskluft überbrückt werden kann

### Klüfte



### Informationsdesign

- 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

### Interaktionsdesign

- 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

## 5.3 Realisierung

### Prototypische Realisierung

- Instead of “Do it right the first time” we will develop iterative prototypes
- A prototype is a concrete but partial implementation of a system design
- Constructed and evaluated to guide redesign and refinement
- Created to explore many questions during system design
  - System reliability
  - Bandwidth consumption
  - Hardware compatibility
- User interface prototype
  - Built to explore usability issues
- User interface prototypes can be built early on in the design process
  - Paper prototype
- Late prototypes will probably be very close to the actual system (depending on process model)

## 5.4 Evaluation

### Evaluation

- Notwendig, um den Fortschritt zu bestimmen
- Aufgabe und Anforderungen schrittweise besser verstehen
- Dadurch Verbesserung der Spezifikation und der Implementierung
- Auch Evaluationen können frühzeitig in den Prozeß einfließen
  - Nicht nur fertige Software kann evaluiert werden
  - Evaluation von User-Interface-Spezifikationen
- Evaluationen helfen, eventuelle Defizite des Designs frühzeitig zu erkennen und zu korrigieren

### Fazit

#### Fazit

- Wir haben gesehen, wie die bereits früh als zentral identifizierten Aktivitäten in einem Menschzentrierten Prozeß eingebunden werden können
- Wir haben erste kurze Beispiele kennengelernt, was in diesen Aktivitäten getan werden kann
- Was uns noch fehlt sind konkrete Vorgehensmodelle, die diese Anforderungen an Menschzentriertes Design auch abdecken können

## 6 Umsetzung

### Umsetzung

- Es stellt sich die Frage, wie solche iterativen Prozesse aussehen können
- Vervielfacht sich durch wiederholtes Prototyping und Evaluation der Aufwand nicht sehr?
  - Fünf Iteration mit jeweils einem voll funktionsfähigen Prototypen, die durch eine große Anzahl von Nutzern evaluiert werden – fünffacher Aufwand?
- Tendentiell höherer Aufwand, zu einem System zu kommen
- Aber: bei nicht menschzentrierten Prozessen ist das erste System häufig nicht das, was die Nutzer wirklich brauchen
- Trotzdem stellt sich die Frage, wie man den Gesamtaufwand so beschränken kann, daß er nicht notwendigerweise größer ist als bei traditionellen Prozessen (bei gegebenem Erfolgskriterien)
- Die folgenden Punkte sind dabei besonders relevant:
  - Wie anfangen?
  - Iterationen
  - Wann aufhören?

### Anfangen und Iterieren

- Wie anfangen?
  - Erste Anforderungen mit Hilfe von Szenarien und Personas modellieren
  - Um zu sehen, wie gut verschiedene Entwürfe zur Aufgabe passen, kann man mit einfachen Systemen beginnen, die Funktionen nur vortäuschen
  - Aufwand kann begrenzt werden, indem man sich auf Teile beschränkt, die wiederverwendet werden
- Iterationen
  - Abhängig vom Fortschritt der Entwicklung, dem Budget und der Art des Systems können verschiedene Techniken angewandt werden
  - Mit einfacheren Iterationen/Feature sets beginnen

## Aufhören

- Wann aufhören?
  - Es müssen Gütekriterien für ein Ende bestimmt werden
  - Iterationen müssen nicht nur zu immer besseren Entwürfen, sondern auch zu einem finalen Design führen
  - Entweder Anzahl der Iterationen vorab festlegen
  - Oder z.B. Kriterien der Gebrauchstauglichkeit verwenden
    - \* “95% der NutzerInnen bewerten das System mit sehr gut”
    - \* “90% der NutzerInnen können eine Aufgabe in 30 Minuten erledigen”
  - Harte Grenzen: Das Budget ist erschöpft
  - Merke: Das Ziel ist nicht, ein perfektes System zu erstellen, sondern eines, das “gut genug” ist
    - \* Perfekte Systems sind unbezahlbar

## 6.1 Szenarien

### Szenarien

- Für jede Entwicklungsarbeit gilt, daß ein genaues Verständnis der zu lösenden Aufgabe Voraussetzung für ein gelungenes Resultat ist
- Unschärf 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

### Inhalte

- Wesentlich, daß
  - typische Form der bisher vorgefundenen Arbeit oder
  - alle Schritte der Benutzung des neuen Produktesaus Benutzersicht dargestellt werden
- Es muß klar werden, wie die Benutzung in das Leben bzw. die Arbeit der Anwenderin integriert ist
  - Eingebettet in den Kontext
  - Was, Wann, Wo, Wie
- Szenarien sind wertvolles Hilfsmittel in allen Phasen der Systemerstellung, z.B. können sie die Evaluation anleiten
- Solcherart textuelle Beschreibungen sind auch für Auftraggeber und Nutzer verständlich
  - Nicht jeder versteht UML-Diagramme
  - Szenarien Kommunikationsgrundlage für Anforderungsdefinitionen

## Beispiel

- “A story about people and their activities” Rosson and Carroll [2002]

*Example 1.* Ein Buchhalter möchte einen Ordner öffnen um ein dort befindliches Memo zu lesen. Dieser Ordner wird aber von einer Spreadsheet Tabelle verdeckt, welche er im Blick behalten möchte während er das Memo liest. Die Tabelle ist sehr groß und nimmt fast den gesamten Bildschirm ein. Der Buchhalter macht eine Pause, verkleinert und verschiebt die Tabelle, öffnet den Ordner und das Memo und plaziert dann die gewünschten Fenster so daß er das Memo und die (bzw. Teile der) Tabelle gleichzeitig lesen kann. (*Beispiel gekürzt.*)

## Szenarien

- Sieht nach Routine aus, zeigt aber einiges:
  - Notwendigkeit der Koordination verschiedener Informationsquellen
  - Vergleich, Kopieren, Integration von Daten
  - Displays sind notwendigerweise zu klein und übertoll
  - Fenster müssen neu positioniert und verändert werden
- Szenarien zeigen auf...
  - welche Ziele von der Anmutung und dem Verhalten des Systems hervorgebracht werden
  - was AnwenderInnen mit dem System machen möchten
  - welche Aktivitäten (nicht) angenommen werden
  - welche Aktivitäten (nicht) erfolgreich durchgeführt werden
  - wie AnwenderInnen das Systemverhalten interpretieren

## Charakteristische Elemente I

- **Setting** – Details über die Situation welche die Ziele, Aktivitäten und Reaktionen der Akteure erklären oder motivieren
  - Buchhaltungsbüro, Stand der Arbeit, Werkzeuge
- **Akteure** – Personen die mit dem System oder anderen Artefakten interagieren nebst relevanter persönlicher Attribute
  - Neuer Buchhalter, das erste Mal mit der Tabellenkalkulation
- **Arbeitsziele** – Motivation der Akteure die Arbeit durchzuführen
  - Vergleich der Zieldaten des Memos mit den Ist-Werten der Tabelle
- **Pläne** – Geistige Tätigkeit welche die Ziele in Verhalten umsetzen
  - Das Öffnen des Memos wird die Zieldaten geben, das Positionieren der Tabelle die Ist-Werte

## Charakteristische Elemente II

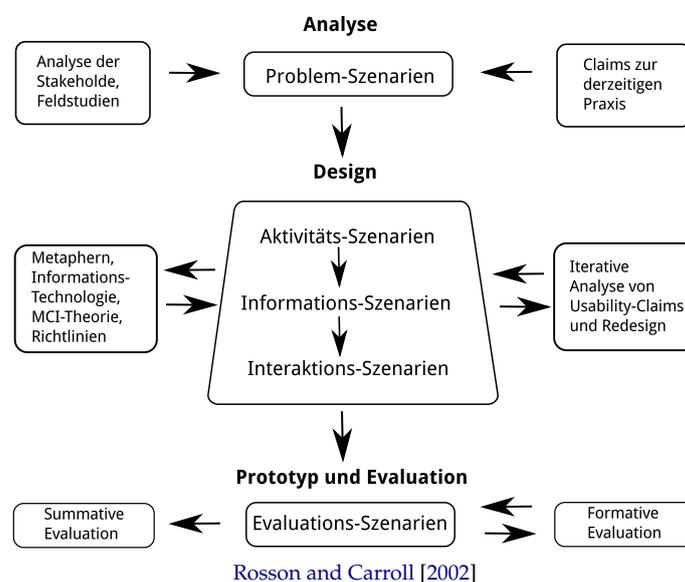
- **Evaluation** – geistige Tätigkeit welche Aspekte der Situation analysiert
  - Zu großes, aktives Fenster
- **Aktionen** - Beobachtbares Verhalten
  - Öffnen des Memos, Verschieben von Fenstern
- **Ereignisse** - externe Aktionen oder Reaktionen des Computers oder anderer Artefakte
  - Feedback aktive Fenster, auditorisches oder taktiles Feedback von Maus und Tastatur

Alles zusammen bildet den Hintergrund für die im Szenario erzählte Geschichte

## 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

## Phasen



## 6.2 Personas

### Problem

- Unscharfe Vorstellungen von potentiellen NutzerInnen des Systems
- Fähigkeiten und Kenntnisse werden oft implizit im Verlauf der Definition angenommen
  - Kann Englisch sprechen
  - Kann Software installieren
- Man spricht hier vom elastischen Benutzer
- Aber: es gibt nicht "den Benutzer"
- Entwickler schlechtes Vorbild für Benutzer
- Um Diskussionen zu konkretisieren und Annahmen zu explizieren können Benutzerklassen oder Personas eingesetzt werden

## Benutzerklassen

- Benutzerklassen sind Mengen "ähnlicher" Benutzer
- "Ähnlichkeit" kann dabei z.B. sein:
  - ähnliche Aufgaben und Rollen
  - ähnlicher Erfahrungsstand
  - ähnliches Milieu
  - ähnliches Marktverhalten
  - ähnliches Alter
- Benutzerklassen sind Abstraktionen von Benutzern
- Problem
  - Benutzerklassen liefern zwar eine Vorstellung von den Benutzern, diese ist aber sehr abstrakt
  - Bei konkreten Fragen zu Konzeption und Gestaltung ist es schwierig, damit zu arbeiten

## Personas

- Vorgeschlagen von Alan Cooper: "The Inmates are Running the Asylum", 1999
- Weiterentwickelt von Cooper in "About Face [2.0 | 3.0]: The Essentials of Interaction Design", [2003 | 2007]
- Personas sind Modellierungen stereotyper Benutzer als Referenzpersonen zu Analyse und Definition von Anforderungen an interaktive Computersysteme
- Personas sind repräsentative, fiktive Individuen einer Benutzerklasse
  - Ähnlich wie Figuren in einem Theaterstück
- Einzelne Designentscheidungen können an Personas direkt geprüft und entschieden werden
- Personas zwingen dazu, über konkrete Benutzer nachzudenken und nicht über Benutzer im Allgemeinen
- Es gibt schließlich keine durchschnittlichen Benutzer
- Eine Systemlösung muß für jede betroffene Persona insgesamt schlüssig sein

## 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

## Maxime

- Bei der Modellierung von Personas gilt:
  - Sei spezifisch
    - \* jeder soll sich die Person genau vorstellen können
  - Bleibe hypothetisch
    - \* echte Personen haben “Macken”, die keine Designgrundlage werden dürfen
  - Vermeide “falsche Genauigkeit”
    - \* genau im Sinne von pointiert
    - \* nicht genau im Sinne von korrekt
  - keine Randperson
- kein Durchschnittsbenutzer (zu unspezifisch)
- Personas müssen glaubhafte NutzerInnen sein
- Personas sind keine Produkte der Phantasie sondern Ergebnisse von Beobachtungen, Interviews, Erhebungen, Workshops etc.
  - alle Eigenschaften müssen auf Untersuchungsdaten zurückgeführt werden können

## 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

## 6.3 Prototypen

### Prototyp-Typen

- 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

## 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

## 6.4 Evaluation

### Warum Evaluieren?

- Eine Usability Evaluation ist jede analytische oder empirische Untersuchung der Gebrauchstauglichkeit eines Prototypen oder Systems
- Das Ziel ist eine Rückmeldung im Software-Entwicklungsprozeß
- Im besonderen sollen iterative Prozesse unterstützt werden
- Aufschlußreiche Analysen und inspirierte Designs können neue Möglichkeiten für Menschen und Organisationen schaffen
- "Best practices" und viel Mühe sind kein Garant für Erfolg
- Mögliche Probleme
  - Nicht nützlich genug
  - Zu schwer zu erlernen oder zu nutzen
  - Nicht befriedigend in der Nutzung
- Die ursprünglichen Zielen können erreicht sein, aber es können die falschen Ziele sein

### Arten

- 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

## 6.5 Contextual Design

### Was ist es?

- Was ist Contextual Design?
  - ein umfassendes integriertes Vorgehensmodell
- Im Einzelnen:
  - Kundenzentrierter front end Designprozess
  - ganzheitliche Methode, die sich am Arbeitsalltag orientiert
  - heterogenes Team mit Mitarbeitern aus allen an der Entwicklung beteiligten Bereichen
  - aus der Praxis in US-amerikanischen Unternehmen entwickelt (Ursprung bei DEC)

### Was ist es? (contd.)

- *Sammlung von Daten über die Arbeit des Kunden dient als Grundlage für Entscheidungen darüber, was das System tun soll und wie es strukturiert ist.*
- Ein Mix aus:
  - bekannten Methoden im neuen Gewand (Contextual Interview)
  - etablierten Techniken, in CD integriert (Paper Prototypes, Scenarios)
  - neuen Modellierungstechniken, um Arbeitspraxis und neues Design darzustellen (Flußmodell, User Environment Diagram)
  - Techniken für Teambildung und gemeinsame Nutzung von Daten (Interpretationssitzung)

## 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

## Fazit

### Fazit

- Methoden, Vorgehensmodelle und Modellierungstechniken kennengelernt, die Menschzentrierte Entwicklungsprozesse unterstützen können
- Einige der hier vorgestellten Ansätze werden wir im Laufe der Vorlesung vertiefen
  - Evaluationsmethoden, Contextual Design
- Die Verbindung mit etablierten Prozessen des Software Engineerings ist notwendig, um sowohl Anforderungen an kosteneffektive Entwicklung funktionaler Software als auch an Gebrauchstauglichkeit zu erfüllen
  - Usability Engineering
- Industrielle und künstlerische Designprozesse werden wir nicht vertiefen
- Fragen “guten Designs” und der User Experience (UX) werden bei Bedarf an passenden Stellen aufgegriffen
- Zunächst Fokus auf Verfahren und Prozessen zur Sicherstellung von Gebrauchstauglichkeit

## 7 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.

### References

## Literatur

Barry W. Boehm. A spiral model of software development and enhancement. *Computer*, 21(5):61–72, May 1988. ISSN 0018-9162. doi: 10.1109/2.59.

Mary Beth Rosson and John Millar Carroll. *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. Morgan Kaufmann/Academic Press, 2002.

Walker W. Royce. Managing the development of large software systems: concepts and techniques. In *Proceedings IEEE WESTCON*, pages 1–9, Los Angeles, 1970. URL <http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>.