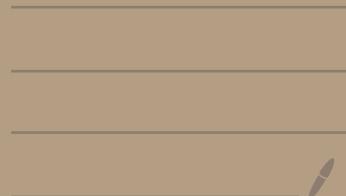


# MMI

---

## SS 20



## HCI Definition

- Interaktion zw. Nutzern und Computern
  - User-Interaktion in Form von **Interfaces** (Software + Hardware)
- Design, Evaluation, Implementierung von Schnittstellen

## HCI Design

- Input + Output → gesamte Interaktion muss entworfen werden

## UI Aspekte

- physikalisch → Knopf drücken
- perzeptuell → was sehe ich?
- konzeptuell → mapping Aktion - Reaktion
- logisch

→ Designprinzip muss Menschen-orientiert sein → was will Nutzer?

## PACT Beispiel - Methode zur Design-Strukturierung

People undertaking  
Activities in  
Contexts using  
Technologies

### Activities

- erzeugen Voraussetzungen / Anforderungen an Technologie
- Requirement Analyse
- Tech-Auswahl



### Technologies

- bieten Möglichkeiten Aktivitäten auf versch. Arten auszuführen
- Technologie beeinflusst Aktivitäten

# People

## äußerliche Unterschiede

- Größe, Gewicht, Seh-, Hörfähigkeit, ...

## psychologische Unterschiede

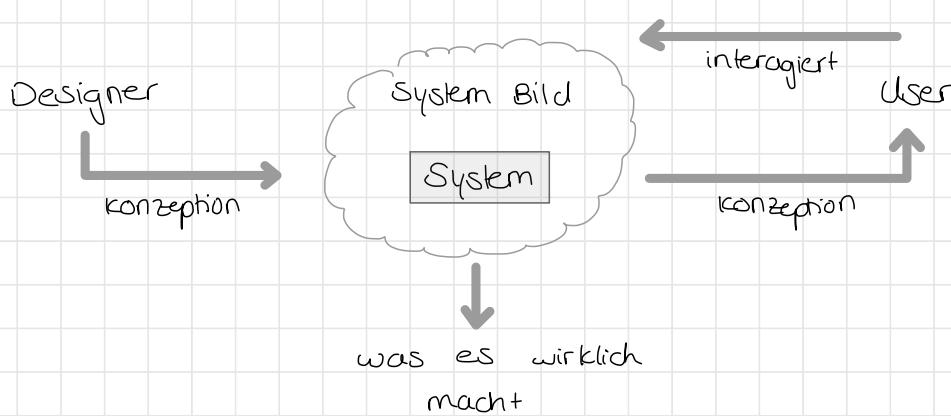
- versch. Arten zu Arbeiten / Lernen / Erkennen / Erinnern / ...
- versch. kognitive Fähigkeiten
- versch. mentale Modelle

## Nutzungsunterschiede

- Experte vs. Erstnutzer
- Verwissen

## Mentale Modelle

- Konzepte = wie denken wir über Dinge?
- Design: Konzept / Modell muss klar gegeben sein
- Sind immer unvollständig → jeder hat "Standard-Skripte" im Kopf die offensichtlich erscheinen



## Activities Charakteristika

### Temporale Aspekte

- wie häufig ist Aktivität
- Aktivität in Stress - oder Ruhesituationen üblich?
- Aktivität als Einheit oder unterbrechbar?
- Antwortzeit System

### Kooperation, Komplexität

- Aktivität alleine oder mit anderen?

### Sicherheitskritisch

- was passiert bei Fehlern?

### Inhalt

- welche Infos, Medien?

### Contexts

#### Physische Umgebung

- Bahnhof oder Bus oder Zuhause?

#### Sozialer Kontext

- Hilfe von anderen?
- Akzeptanz von Designs

#### Organisatorischer Kontext

- Energieversorgung
- Lebensgewohnheiten ändern
- ...

### Technologies

#### Input

#### Output

Kommunikation zw. Person, Technik / zw. Geräten

Content Funktional vs. kontextbasiert

## Szenarien

- Geschichten über Nutzung = PACT - Sätze  
konzeptuell
  - decken Hauptaktivitäten ab "Pete logs onto the comp."
  - konkreter
    - detaillierte Versionen "Pete clicks the log-in icon"

## Personen

- Beispiel person als Beispiel-Endnutzer
- hilfreich: konkrete Personen mit Background-Stories, Alter, ...

## OBSERVATION

→ Studien

### Zeitpunkt der Studie

Formativ = am Anfang / während Prozess um Kontext / Möglichkeiten zu untersuchen

Summativ = am Ende zur Beurteilung

### 4 W - Fragen der Evaluation

Warum (Why) : Hypothese, Fragestellung

Was (What) : was wird untersucht → Prototyp? Verhalten?

Wo (Where) : natürliche / laborumgebung?

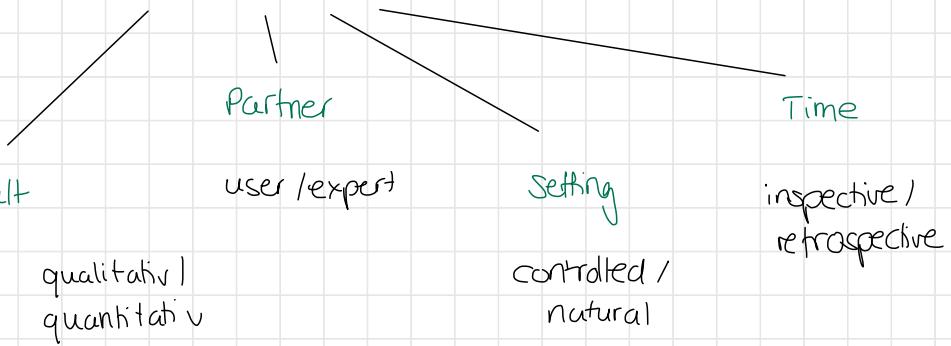
Wann (When) : formativ / summativ?

## Messwerte

### User:

- Effektivität zB % der Aufgaben erfüllt
- Effizienz zB task completion time
- Satisfaktion zB Reuse - Frequency

## Klassen der Evaluation



Result Type abh. von "why"

Subjektiv : Resultate nicht vergl. bar zw. Probanden

Objektiv : Resultate vergl. bar zw. Probanden

Quantitativ : Zahlen

Qualitativ : Text

Evaluation Partner abh. von "what"

### User

- direktes Feedback

### Experte

- best practice Infos

Setting "where"

Controlled = Laborsetting

- kontrollierte, reproduzierbare Umgebung
- protokollieren der nicht kontroll. Bedingungen

Natural

- im "normalen" Umfeld der Nutzung
- nicht kontrollierbar
- nicht alle Bedingungen messbar

Evaluation Time "when"

Inspectiv

- Inspektion + Evaluation während Experiment / Nutzung

Retrospektiv

- Evaluation nach Experiment / Nutzung

## INTERVIEW

Hauptaufgaben

1. Zielsetzung festlegen  
↳ Hypothese + entspr. Methoden / Methodiken
2. Probanden wählen
3. Beziehung zu Probanden  
→ professionell!
4. Triangulation  
→ sammle versch. Arten von Daten! Betrachte Daten aus versch. Perspektiven
5. Pilotstudien  
→ "Probelauf" vor Main Study!

## Interview Typen

### Strukturiert

- vorgefertigte Fragen → strikt der Formulierung folgen
- ⊕ einfach auszuführen
- präzisere Auswertung
- ⊖ limitiert auf Fragenkatalog

### Semi-strukturiert

Strukturierter Teil + "offene" Fragen

### unstrukturiert

- wenn zB wenig Hintergrundinfo bekannt
- minimiert Einfluss des Fragestellers

## Frageformate

### Checkboxes

- Ja/Nein
- mehrere Möglichkeiten
- Multiselect

### Rating Scales

- Likert
- Semantic Scales

### Open-ended

- Freitext

### Likert Skala

quantitativ, kategorisch

- 5 Stufen, Mitte = Neutral
- Resultat: Aufsummieren der Antworten

## Interview Ablauf

### 1. Introduction

- Vorstellung
- Zielsetzung erklären
- Consent

### 0. Vorbereitung

### 2. Warm - Up : einfache Fragen

### 3. Hauptteil : Fragen in logischer Reihenfolge

### 4. Cool - Off: einfache Fragen

### 5. Closure

- Bedanken

- ! · Kontext - Faktoren beachten (Lärm, Temperatur,...)
- ! · Fragesteller muss neutral wirken

## Fragekatalog

häufig genutzt.

- SUS
- TLX
- User Experience Questionnaire
- Computer System Usability Questionnaire

## System Usability Scale (SUS)

retrospektiv, subjektiv

- deckt Effektivität, Effizienz, Satisfaktion ab
- 10 Fragen, abwechselnd positiv, negativ → Skala 0-4
- SUS Punkte aufaddieren → mul mit 2,5
- > 68 gut (immer noch Bias in Richtung positiver Bewertung)



- Likert Skala → einfach

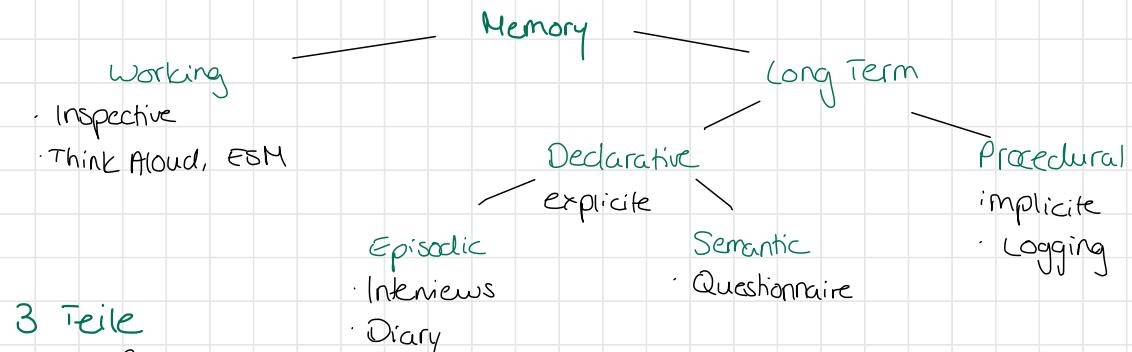
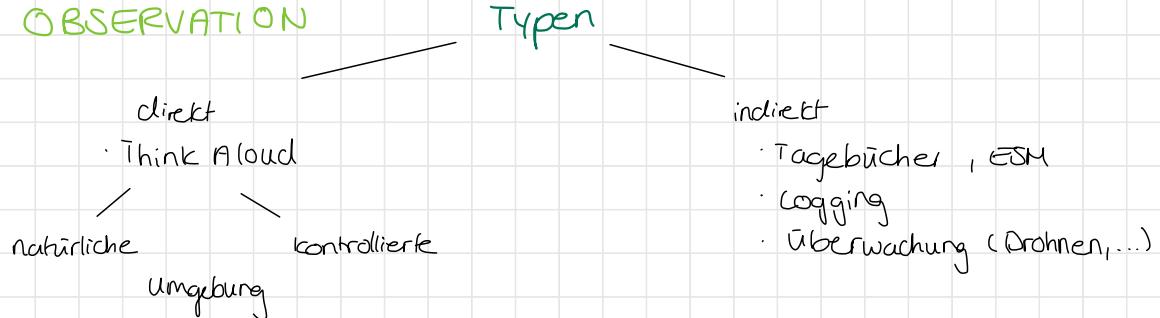
- Generalisierbar, sofern andere Params korrekt (repräsentative Probanden, ...)



- Skala 0-100 ! keine Prozentzahl

- nicht diagnostisch → keine Rückschlüsse auf spezifische Probleme

# OBSERVATION



## 3 Teile

- Who?
- Where?
- What?

**Think Aloud** kontrollierte Umgebung, inspektiv, subjektiv, end-user  
User beschreibt konstant was er denkt während Aufgabe

- Güte abh. von
  - Kandidatenauswahl
  - Setting
    - ↳ so natürlich wie möglich, ruhig, ohne Unterbrechungen
- aufgezeichnet mit Audio / Video / Transkript

## Auswertung

- Interpretation! → Psychologie einbeziehen
- Think Aloud = verbalisierter Teil des Inhalts des Arbeitsgedächtnis
- Generalisierung schwierig

## Feld Studien

### Living Labs

- Nutzer nutzen neue Technologien in t iglichem Leben  
→ Kontext + Situation nicht kontrolliert

⊕ wahrscheinlicher dass neue Einblicke / Erfahrungen

⊖ teurer

### Ubicomp Studien

#### Current Behaviour

- Ergebnis: besseres Verst ndnis f r aktuellen Gebrauch  
→ Implications f r zuk nftige Entwicklung
- Fragen: open-ended  
→ Frage nach aktuellem Gebrauch

#### Proof-of-concept

- Studie: neue Technologie + ggf. Hilfstechnologie verteilen  
↳ nach bestimmten Eigenschaften fragen

#### zB wizard-of-OZ Studien

- Person kontrolliert / simuliert Systemverhalten  
↳ vor Nutzer versteckt

#### Experience

- = Erfahrungen mit neuer Technologie  ber l ngeren Zeitraum

#### Umfragen nach Anclerungen

#### Logging

- Sammle Daten  ber Nutzung automatisch

## Indirekte Methoden

### Logging

- wählt wichtige Daten die geloggt werden sollen  
(Pilotstudie!)
- entsprechend spezifischer Fragestellungen

### ESM Experience Sampling Method

- Nutzer muss Fragebogen ausfüllen (mehrmais pro Tag/Woche)  
↳ bessere Ergebnisse als z.B. am Ende des Tages recap

### Aspekte

- Häufigkeit? nervig vs. präzise
- Umfang? so wenig Fragen wie mögl.

### Tagebuch

- Proband entscheidet selbst wann Erfahrungen aufschreiben

### ESM Speciality

- zeichnet Änderung des Nutzerverhaltens auf

## Design + Evaluation eng verbunden!

### Design

1. Forschungsfrage konkretisieren
2. Beantworten
  - wer nimmt teil?
  - Aufgabe der Probanden?
  - welche Daten erhoben?
  - wann / wie lange ist Studie?

## Daten

- erhebe qualitative = besser verständlich
- + quantitative = besser vergleichbar

Dauer abh. von

- Studienart (Experience (einger als proof-of-concept, ...))
- Neuheit der Technologie
- praktischen Aspekten: Aufwand für Probanden, ...
- Frequenz der Nutzung: höhere Frequenz = kürzere Dauer

## Evaluation

	Kontrollierte Umgebung	natürliche Umgebung	ohne Nutzer
Observation	X	X	
Umfrage	X	X	
Experten-Fragen		X	X
Testing	X		
Modellierung			X

Verlässlichkeit ist Studie reproduzierbar?

Validität beantwortet Studie wirklich Forschungsfrage?

- intern: cause-and-effect Beziehung bekannt
- extern: Ergebnisse generalisierbar

Bias Ergebnisse verfälscht durch Bias?

Scope wie gut generalisieren Ergebnisse?

# PROBANDEN

## Auswahlkriterien

### Repräsentation

- muss Probandenset repräsentativ sein oder reicht nicht-repr. Set?  
→ ! viele statistische Tests brauchen repr. Set!

### Gruppierung

- 1 Gruppe
- > 1 Gruppe:
  - (Selbsteinschätzung) Erfahrung: Experte, Intermediate, Neuling
  - Häufigkeit der Nutzung
  - Erfahrungen in Domäne
  - Demographie (Alter, Geschlecht, ...)
  - versch. Aktivitäten

### Sampling Strategie

- Zufällig
- Systematisch → basierend auf def. Kriterium, z.B. "jeder 10."
- stratifiziert → zusätzlich noch Zielgruppenverteilung beachten, z.B. 50% male/female
- Sample of convenience → freiwillige (oft Freunde / Familie/...)

### Anzahl

Nielson: ~ 5 Probanden finden 75-80% der usability Probleme

Cockton, Woolrych: abh. von Kontext + Task

→ abh. von akzeptablem Fehler: je geringer desto mehr Probanden

### Pre-Test als Abschätzung

- Probanden müssen bekannte Fehler finden
- finden im Schnitt  $p$  Fehler

→  $n$  Probanden der eigentlichen Studie finden  $\sim 1 - (1-p)^n$  Fehler

## Vergleich

### within Subject

- 1 Proband bekommt mehrere Tasks
- Ergebnisse der versch. Tasks vergleichen

! Lerneffekt: Reihenfolge der Tasks beeinflusst ggf. Ergebnis  
=> Reihenfolge je Proband ändern sofern möglich

### Between Subjects

- alle Probanden bekommen gleichen Task
  - vergleiche zw. Probanden(-gruppen)
- (+) kein Lerneffekt  
(-) braucht mehr Probanden

## Ohne Probanden Inspections

### Experten

- nutzen Wissen über Nutzer, Technologien um SW Usability zu bewerten

### Experten Kritiken

formal/informal

## Heuristische Evaluation

- Review basierend auf Heuristiken

### Wichtigste Kriterien

- Klarheit  
minimiere unnötige Komplexität + kognitive Last
- gebe User Kontext
- user experience positiv gestalten

## Ablauf

1. Briefing: Experten die Aufgabe erklären
2. Evaluation: 1-2h
  - jeder Experte arbeitet allein
  - 1. Durchlauf um Gefühl für AW zu bekommen
  - 2. Durchlauf um auf spezifische Features zu achten
3. Debriefing: Experten priorisieren Probleme

## Vorteile

- Experten haben Domänen- und Nutzerwissen

## Nachteile

- schwer + teuer Experten zu finden
- wichtige Probleme werden übersehen → oft besonders triviale
- Experten haben Bias

## Walkthrough

### Cognitive Walkthrough

- Fokus auf Einfachheit der Nutzung + Lernbarkeit
- Experte bekommt Annahmen über Nutzergruppe, Nutzungskontext, task details
- Designer präsentiert Design + Nutzungsszenarien
  - ↳ Experte(n) gehen durch Prototyp mit Szenario
  - angeleitet von Fragen (zB ist Nutzer klar dass korrekte Aktion?)
  - negative Antwort = usability Problem

### Pluralistic Walkthrough

- Cogn. Walkthrough mit Team von Experten
- alle Experten arbeiten anfangs allein
- managed discussion → agreed decisions

## WORKLOAD MESSEN

- mentalen workload messen

↳ User nicht überfordern ↳ schlechte Performance durch Stress  
unterfordern ↳ Langeweile

## Mental Workload Measure

### Kriterien

- Sensitivität: gegenüber Änderung in Schwierigkeitsgrad / Ressourcenverbrauch
- Selektivität: NICHT sensitiv gegenüber Änderungen unabh. Parameter
- Diagnostik: Index muss Ursache der Änderung abbilden
- (Un)Aufdringlichkeit: Index darf primären Task nicht beeinflussen
- Verlässlichkeit / Reproduzierbarkeit
- Bandbreite: Index muss auf schnelle Änderungen des Workloads reagieren

### Ansätze

#### primary task direkt

- messe Performance Metriken

- Zeit
- Geschwindigkeit
- Stärke

→ leite Workload-Metriken ab

- keine absoluten Werte
- Performance untersch. ggt. Workloaduntersch.

#### secondary task indirekt

- rhythmisches Klopfen → variiert mehr je höher workload
- zufällige Zahlen sagen → Zufälligkeit weniger je höher workload
- Reaktionszeit testen → länger
- Zeitschätzung → Überschätzung bei Über- / Unteraufordnung

## Physiologische Messungen

- EKG / EMG / EEG
- Schweißbildung
- Sauerstofflevel
- Augentracking

- kompliziertes Setup
- braucht akkurate Baseline
- schwer vergleichbar zw. Probanden

## Subjective Ratings subjektive Einschätzung vom Proband

### NASA - TLX Skala Beispiel Rating

Messung in 6 Dimensionen

- Mentaler Anspruch
- Physischer Anspruch
- Zeitlicher Anspruch (gehekt?)
- Performanz (wie erfolgreich wurde Aufgabe erfüllt)
- Bemühung
- Frustration

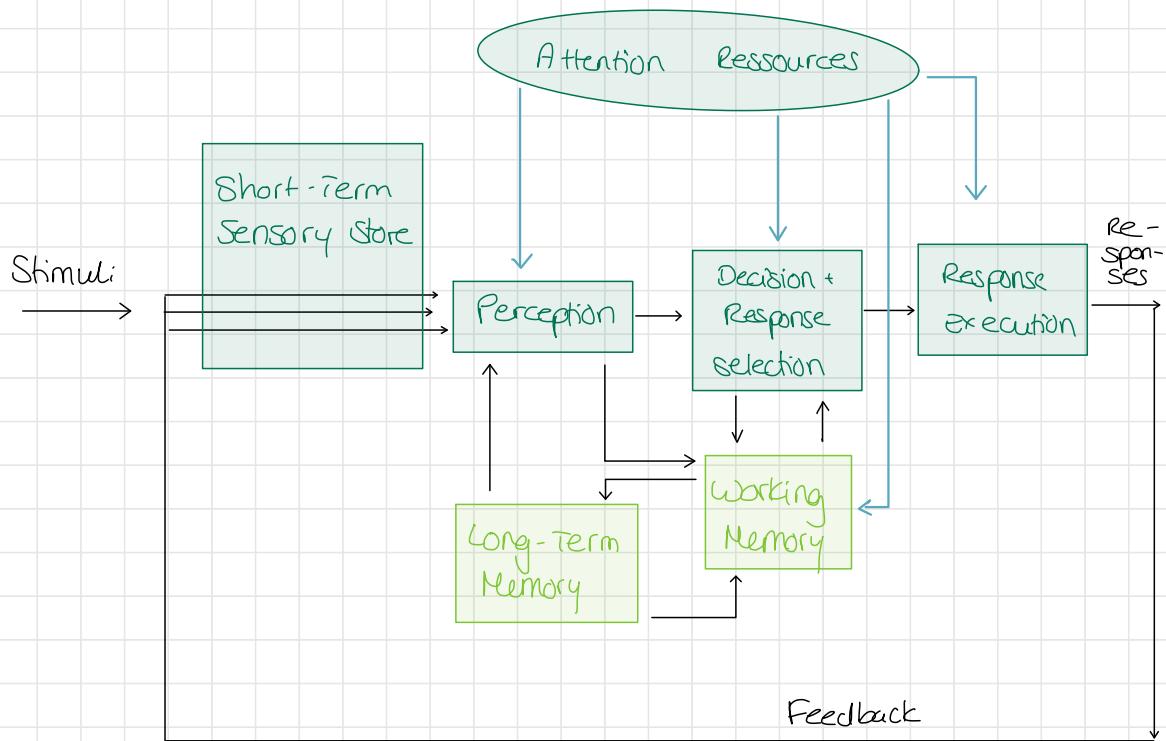
2 Schritte:

1. 6 Dimensionen auf Skala von 1-20 bewerten
2. 15 Paarweise Vergleiche

Scoring:

- pro Dim.
  - Summe wie oft ausgewählt bei 2.
  - durch 15 teilen
  - mult. mit Score aus 1.
- Summe über alle Dim

# INFORMATIONSVERARBEITUNG



## STIMMULI

### Sinne

- 6 externe Sinne: sehen, hören, tasten, riechen, schmecken, gleichgewicht
- interne Sinne: Temp. empfinden, Schmerzempfinden, Tiefensensibilität

↪ Fähigkeiten sind limitiert

## STSS Short-Term Sensory Storage

! = short-term mem

- arbeitet ohne explizite Attention
- Speichert Wahrnehmungen kurz nach Stimulus
- nicht Teil des conscious mem.

echoic memory  
iconic memory

2-10s nach Stimulus zB "was hast du gesagt?"  
0,5-1s nach Stimulus zB Buchstabenfol

# PERCEPTION

## Informationsübertragung

Limitiert!

- wie viel Info übertragen von Stimulus → Reaktion
- Kapazität des Infokanals
- Bandbreite

## Perzeption

- Sensordaten müssen interpretiert werden = Perzeption
- automatischer, schneller Prozess  
↳ wenig attention nötig

bottom-up gegeben von Sensorinfos

top-down gegeben von Langzeitgedächtnis (zB bei unklaren sensorischen Infos)

## ATTENTION

- kognitiver Prozess: Konzentration selektiv auf einen Aspekt der Umwelt ignorieren der restlichen

### Arten

selective attention bewusst Fokus wählen

focused attention auf externe Events reagieren

? eine Stimulusquelle

- zu selektiv = cognitives tunneling

- schwer: leicht ablenkbar

↳ selective: absichtliche Selektion der falschen Quelle

focused: unabsichtliche externe Störquelle

divided attention gleichzeitig auf versch. Events fokussieren

↳ max. Kapazität beachten

## Modelle

### Allocation Modell (Kahnemann)

- limitierte Menge "processing power"

→ Aufgabenausführung abh. von wie viel noch übrig

## controlled and automatic attention (Schneider + Shiffman)

- kontrolliert :
  - hoher Ressourcenverbrauch
  - langsam
  - limitierte Kapazität
- ↳ muss bewusst auf Aufgabe gelenkt werden
- automatisch :
  - braucht keine attention ressourcen
  - schnell
  - keine Kapalimits
  - unumgänglich, schwer zu beeinflussen
- ↳ keine bewusste Lenkung

## WACHSAMKEIT

= erkennen eines selten Ereignisses bei viel Inaktivität / Rauschen

### Paradigmen

#### free-response

- Zielevent tritt zufällig auf
- Nichtevents undefiniert

zB AKW Monitoring

#### inspection

- Events relativ regulär
- manche Events = Zielevents
- meisten = Nichtevents

zB Qualitätskontrolle

#### successive vigilance

- Zielevent geht mit Zielstimulus einher → merken
- sukzessive Events mit Zielstimulus vergleichen

zB erkennen Farbe heller/dunkler

#### Simultaneous vigilance

- alle Events/infos um Events zu untersch.
- gleichzeitig verfügbar

zB Stoffteile mit Standard vergleichen

#### sensory

- Signale ändern auditive/visuelle Intensität

zB Farbwechsel

#### cognitive

- Signale repr. Infos : symbolische / alphanum. Stimuli

## SIGNAL DECISION THEORY

2 Zustände: Signal, Rauschen

2 mögliche Reaktionen: ja, nein

		State	
		Signal	Rauschen
Reaction	Ja	Hit	False Alarm
	Nein	Miss	Correct rejection

## Informationsverarbeitung

Aggregiere sensorische  
Infos über vorhanden-  
Sein eines Signals

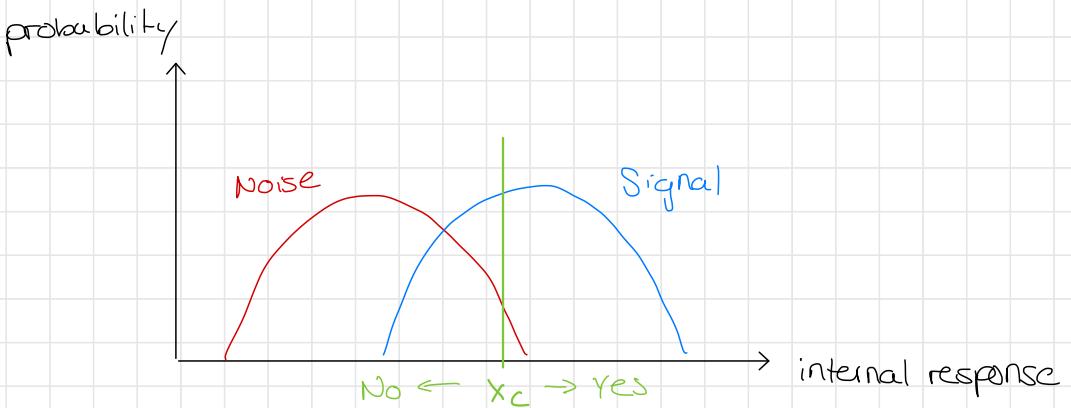


Entscheide ob Evidenz  
für Signal

- externe Stimuli  $\Rightarrow$  neuronale Aktivität: im  $\emptyset$  mehr falls Signal

## Rauschen

- intern = im Hirn, zB Neuronenaktivität variiert bei gleichem Stimulus
- extern = in den Daten
- internal response = im Hirn des Beobachters



$x_c$  = Entscheidungskriterium

$$\Rightarrow \text{response bias } \beta = \frac{P(X_c|S)}{P(X_c|N)}$$

Optimal beta =  $\beta_{\text{opt}} = \frac{P(N)}{P(S)}$ , Wkt für  $x_c$  gleich

Sluggish beta = menschl. Realität

↳ überschätzen von seltenen Events  
unterschätzen häufigen

Sensitivity = Auflösung

Wie weit liegen Rauschen, Signal auf x-Achse entfernt

## MEMORY

2 Hauptkomponenten

- Arbeitsgedächtnis (= Kurzzeitgedächtnis)
- Langzeitgedächtnis

## Arbeitsgedächtnis

- Speicherdauer ~ 80s → danach: vergessen oder umlagern
- schneller Zugriff ~ 70ms
- Speicherkapazität 3-4 chunks
- schnelles überschreiben
- effizientester Reiz: Sound

↓  
ins Langzeitged.  
durch Wiederholen

## Typen

- Verbal : Worte geschrieben + gesprochen
  - visuell : oft als Bilder
- } arbeiten parallel

## Langzeitgedächtnis

- Speicherdauer min - immer
- langsamer Zugriff ~100 ms
- unlimitierte Kapazität
- multi-modal
  - effizientester Reiz : Geruch

im Alter:

## Typen

- episodisch : spezieller Speicher von Events, Erfahrungen
  - prozedural : Wissen wie man Dinge tut, zB laufen
  - semantisch : strukturierte Fakten, Konzepte, Fähigkeiten
- weniger  
gleich  
gleich-  
mehr

## Prozesse

### Vergessen Infos nicht wieder abrufbar

- Decay : langsames Verschwinden der Infos
- Interference :

- retroactive : neue Infos ersetzen alte

- proactive inhibition : alte widersprechen neuen

↳ beeinflusst von Emotionen → unterbewusstes absichtliches Vergessen

## Encoding Speichern von Infos

### Retrieval Abrufen von Infos

- Recall = Erinnerung : aktives Durchsuchen des Gedächtnis nach bestimmter Info
- Recognition = Wiedererkennung : abgleichen Infos aus LTM mit gegebener Info → schneller als Recall

**Rehearsal** kein direkter Link Perzeption  $\rightarrow$  LTM  
- Einlagern Infos in LTM über Wiederholung

## DECISION

**automatic** schnell

- wenig / keine Attention nötig
- = gelernte Reflexe / Verhalten
- $\rightarrow$  LTM procedure  $\approx$  automatic

**controlled** langsam

- attention nötig
- Interaktion working mem  $\hookrightarrow$  LTM

## RESPONSE + FEEDBACK

**feedback - loop** : beobachtete Konsequenzen der Reaktion  
 $\hookrightarrow$  circular model

**CNN - Modell** Card, Moran, Newell

- 3 interagierende Subsysteme
- perceptual cognitive motor
- je mit Processor + Memory
- seriell + parallel Verarbeitung

**perceptual**

- Sensor Input (audio + visuell)
- $\hookrightarrow$  Infos symbolisch codieren
- $\hookrightarrow$  schreiben in WM Buffers (audio + visuell Storage)

## cognitive

- input = Buffer
- LTM access um Aw zu generieren
- Aw in working Mem. schreiben

## motor

- input = Aw aus WM
- ↪ ausführen

## Parameter

- Memories:
  - Kapazität  $\mu$
  - decay time pro Item  $\delta$
  - info code typ  $k \in [\text{physisch, akustisch, visuell, semantisch}]$
- Prozessoren:
  - cycle time  $\tau$

## Power Law of Practice

- nach n-tem Versuch Ausführungszeit

$$T_n = T_1 \cdot n^{-\alpha} \quad \alpha = 0,4$$

↪ gilt für Fähigkeiten, NICHT für Lernen

# PERCEPTION

## SEHEN

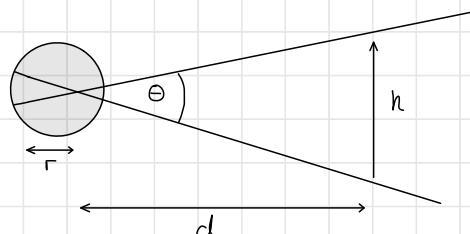
- Schärfer in Mitte des Fokus
- Objekte näher als 25cm schwer zu fokussieren  
↳ dauerhaftes Nah-Fokusieren  $\Rightarrow$  fatigue
- max sampling freq.  $\approx 22$  fps

Perception = Sehr sensitiv ggüüber Bewegungen außerhalb  $30^\circ$  vom Hauptfokus

## Lesen

- optimale Distanz
  - Handy / Tablet 40cm
  - Uhr 50cm
  - Comp. Bildschirm 55 - 60cm

$$h = 2d \cdot \tan\left(\frac{\theta}{2}\right)$$



kognitive perception = Gedächtnis + "Erfahrung" + fokale Perception

## Alterung

- Alterswertsichtigkeit  $\rightarrow$  near-point reading vermeiden
- Kontrastseitivität wird schlechter

## Gestalt Theory

Visual Perception = Bedeutung aus Gesehenem ziehen  
 $\hookrightarrow$  Verbindung zu recognition + understanding

Vision = Farbenerkennen, Formen, Kanten, ...

## Gestalt Laws

### proximity

- Objekte nahe beieinander in Raum / Zeit werden als zusammengehörig wahrgenommen

### Prägnanz

- unbekanntes automatisch zerlegen in bekannte Formen/Muster

### Closure

- Tendenz Objekte als geschlossen wahrzunehmen trotz Lücken  
↳ geschlossene Objekte einfacher wahrzunehmen

### Continuity

- Tendenz glatte, kontinuierliche Patterns wahrzunehmen statt lose, unterbrochene

Common fate: Objekte die sich zusammen "bewegen" scheinen  
zusammengehörig

Parallelism: Chunking durch Alignment

### Similarity

- gleiche Figuren/Formen als zusammengehörig

### Symmetry

- vereinfacht Wahrnehmung

### Part - whole Relationship

- das Ganze größer als Einzelformen

zB

H	H	H	H	H
	H			
H	H	H	H	H

vs.

I	I	I	I
	I		
I	I	I	I

?

H

## Area

- Objekte mit kleinem Bereich werden als großem Einzelobjekte wahrgenommen Hintergrund

## Surroundness

- umschlossener Bereich = Objekt
- umschließender = Grund

## Common Region

- chunking nach Region, nicht Nähe

## Anderer Prinzipien

### Stimulus Intensity

- als Erstes Reaktion auf starke des Stimulus  
↳ dann erst Bedeutung finden

## Proportion

- Hierarchien darstellen zB Header sizes HTML

## Goldener Schnitt

- als ästhetisch wahrgenommen

$$\frac{a+b}{a} \stackrel{!}{=} \frac{a}{b}$$

zB Fibonacci - Reihe

## Screen Complexity (Tullis)

- relative Komplexität  $\sim$  Schwierigkeit Design

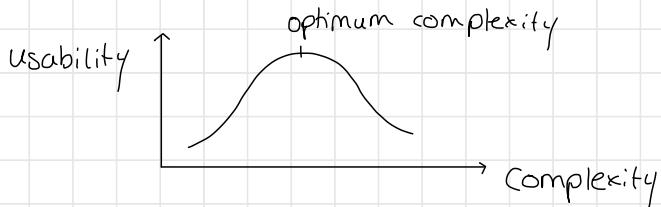
$$C = -N \sum_{n=1}^m p_n \log_2 p_n$$

↑      ↑      ↗  
Systemkompl. total # events      Wert dass n-te Eventklasse auftritt  
in Bits

## Berechnung:

1. um jedes Objekt Rechteck legen
2. Zähle # Objekte + # Spalten = vertical alignment points
3. -u- + # Reihen = horizontal

## Complexity vs Usability



geringere Komplexität:

- bessere Vorhersagbarkeit
- schwerer Objekte abzugrenzen

## Tiefensehen

3D-Sehen

### primary depth cues

- retinal disparity: Augen ca 7cm entfernt
  - ↳ 2 versch. Bilder
  - ↳ Hirn interpretiert das als Distanz
- Stereopsis: versch. Bilder der Augen kombinieren
  - ↳ 3D Bild
- accommodation: Muskelarbeit um scharf zu stellen
  - ↳ Muskelinfos nutzen für Tiefe berechnung
- convergence: bei 2-7m entfernten Objekten: Augenbewegung nach innen um zu fokussieren
  - ↳ nutzte Infos

- Secondary depth cues = Basis für Tiefendarstellung auf 2D Monitor
- Licht und Schatten
  - lineare Perspektive
  - Höhe auf horizontaler Ebene
  - Bewegungsparallaxe (Zug: nahe Objekte bewegen sich schnell entfernte langsam)
  - Überlappung
  - relative Größe
  - Texturgradient entfernter = unschärfer

## Farbsehen

Additiv Licht mischen

für zB Displays

nicht so natürlich wie reflektiertes Licht

Subtraktiv gedrückte Farben

reflektieren Teile des Lichts

## Limits

- blau am Rande des visuellen Spektrums → nicht für kleine Objekte
- Farbuntersch. erkennen abh. von Objektgröße
- Formen werden über Kanten erkannt → schneller als Farbe
- Farbenblindheit

## Design

Unterstützend: Claritying, Relationen, Unterscheidungen, Finden vereinfachen

## Color Coding

- geg. Coding beachten  
↳ Farbe nur zur Unterstützung, ersetzt NICHT Konzept
- untersch. zw. Kulturen

- können nur 10 Farben gleichzeitig untersch. (aber 256 Grautufen)
- nutze Komplementärfarben
- Achtung: Displays nicht kalibriert  
Farbsehen tagsabh. (Lichtefall,...)

→ max 4 Hauptfarben mit je max 4 Varianten

## Farbsysteme

- Hue = Farbton
- Value/luminance/brightness: wie "dunkel" Farbe erscheint
- Sättigung = Graulevel "Schleier"

## HÖREN

### Lautstärke

- in dB = logarithm. Skala
- Flüstern 15dB
- Reclen 60dB
- Hype 110dB
- Hörpektrum 20Hz - 20kHz

### Hörvorgang

#### 1. Transduktion

Übersetzen Geräusch → Nervenimpulse

#### 2. Auditive Gruppierung

- (i) aufteilen in getrennte Streams
- (ii) Integration von Streams  
z.B. nach gleicher Harmonie, Frequenz, Ort, ...

#### 3. Szenen Analyse

#### 4. Interpretation

## Psyacoustic Model führt Lücken mit "Wissen"

### Audio UI

- Sehen + hören gemeinsam
- ↳ Audio sagt Augen wohin schauen
- ↳ ständig Audioinput = Stressfaktor
- Hören schneller als Sehen

### Sprache

- Sprechen schneller als Schreiben  
aber Geschriebenes mehr Redundanz
- Lesen schneller als Hören → effizienter

### Nicht-Sprache

- auditives Feedback  
→ wenige Konzepte initial für Zuordnung Aktion - Sound

### Redundant Coding

- Sounds als Reaktion "Click" sound bei Buttons
- nützlich in Kombi mit Emotionen
- ↳ besseres Recalling von Aktionen

## DESIGN ANALYSE

### Gesetze

- idealisiert!
- helfen um Parameter der UI zu schätzen

### physikalische Gesetze

- Effizienz basierend auf phys. Aspekten des Designs
- berechnen Zeit um Aktionen auszuführen

### Fitts' Law

- abschätzen von Größe + Ort von Objekten

### Index of Difficulty

- Schwierigkeit basierend auf Breite, Distanz

$$ID = \log_2 \left( \frac{A}{w+1} \right) \quad [\text{bits}]$$

A = Distanz vom Startpunkt

w = Größe des Targets entlang Bewegungslinie  
↳ normal kleinerer Wert von Höhe, Breite

### Movement Time

- Zeit bis Task beendet

$$MT = a + b \cdot ID \quad a, b \text{ abh. von experimentellen Bedingungen}$$

a in sec

b in s/bits

- Annahme: lineare Bewegungsgeschw.

- Unterscheidung ziehen / ziehen ← langsamer

## Schlussfolgerungen

- sehr lange Objekte kein Mehrwert
- Objekte verlängert entlang üblichem Trajektorienpfad
- Objekte ohne Offset vom Rand besser

## Hick's Law

- benötigte Zeit proportional zur # möglicher Alternativen

$$T = a + b \cdot H$$

- n Alternativen, alle gleich wkt  $\Rightarrow H = \log_2(n+1)$
- n versch.  $p_i \Rightarrow t_l = \sum p_i \log_2(1/p_i + 1)$

## Power law of Practice

Übung!

$$T_n = T_0 \cdot n^{-\alpha}$$

## Mentale Modelle

- unscharf  $\rightarrow$  reichen ggf. nicht für Systemerklärung aus
- Anhaltspunkt für Menschen was passiert

predictive Vorhersage was vermutlich passiert wenn ich X mache

explanatory Erklärung warum verm. Y passierte als ich X mache

## Eigenschaften

- unwissenschaftlich  $\rightarrow$  Vermutungen
- partiell  $\rightarrow$  nur Teile des Systems beschreiben
- instabil  $\rightarrow$  ändern sich mit Kontext + Erfahrung
- inkonsistent  $\rightarrow$  teilweise inkompatibel
- persönlich / individuell

## Mapping

- Bilden mentale Modelle
- ↳ Lenken durch Constraints in der Oberfläche

## Guidelines

- natürliche Mappings nutzen → konsistent mit Nutzer-Wissen
- gute Mappings:
  - verständlich
  - konsistent
  - wiedererkennbar / schnell lernbar
  - natürlich

## Constraints

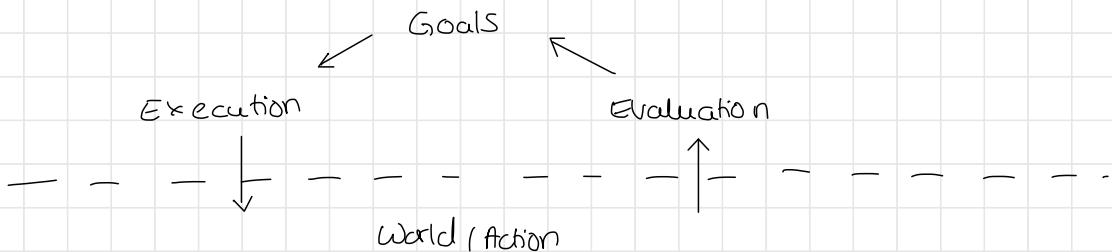
- Hilfe zur Bildung mentaler Modelle
- minim. Fehlerwirk.
- physikalisch, semantisch, kulturell, logisch

## Affordance

Objekte und Umwelt implizieren Nutzung durch ihre Gestalt

- Ziel: wahrgenommene und tatsächliche Eigenschaften matchen
  - nutzbare Features offensichtlich machen
  - natürliche Assoziationen nutzen
  - Feedback geben

## Execution / Evaluation Action Cycle EEFAC



## Stages

- 1. Zielsetzung festlegen } Goals
- 2. Intention formen
- 3. Action spezifizieren
- 4. Action ausführen } Execution
- 5. Weltzustand wahrnehmen
- 6. Wahrnehmung interpretieren } Evaluation
- 7. Interpretation evaluieren

## Gulf of Execution

Mismatch zw. Intentionen und erlaubten Aktionen

## Gulf of Evaluation

Mismatch zw. Systemrepräsentation und Usererwartungen

## Normen Prinzipien für gutes Design

- Zustand + Aktionsalternativen immer sichtbar
- gutes konzeptuelles Modell
- Interface muss Mappings zw. Stages klar machen
- Feedback

## Task Analyse

- wie wird Arbeit über Tasks ausgeführt  
↳ bewerte Performance der Arbeit

Task = Ziel + geordnete Abfolge von Aktionen

Ziel = Anwendungszustand der erreicht werden soll

Action = kleinteilige Task ohne Problemlösung

## Analysemethoden

### Logische Ebene

- Sequenz von Steps

### Kognitive Ebene

- welche kognitiven Prozesse zur Zielerreichung nötig

HTA = hierarchische Task Analyse logische Ebene

- graphische Repr. der Task Struktur

↓  
Sequenz von Tasks

- je aufgeteilt als Hierarchie mit Subtasks
- ggf. mit Auswahloptionen

GOMS Goals, Operators, Methods, Selection Rules Kogn. Ebene

Goals = Tasks

Operators = Aktionen um Task zu erledigen

Methods = Subgoals

Selections = Regeln um korrekte Method zu wählen

### KLM Keystroke Level Model

- physikalische Interaktion die nötig um Task zu erledigen
- 5 basic operations je mit versch. Dauer

K = Knopf drücken

P = Point mit Maus

H = Home

M = Mentale Vorbereitung

R(t) = Systemantwortzeit

# DESIGN

## Szenarien

= Grundlage für Design interaktiver Systeme

→ Verständnis aktueller Nutzung  
Probleme identifizieren  
Dokumentation + Rationale  
Kommunikation von Ideen zw. Stakeholdern

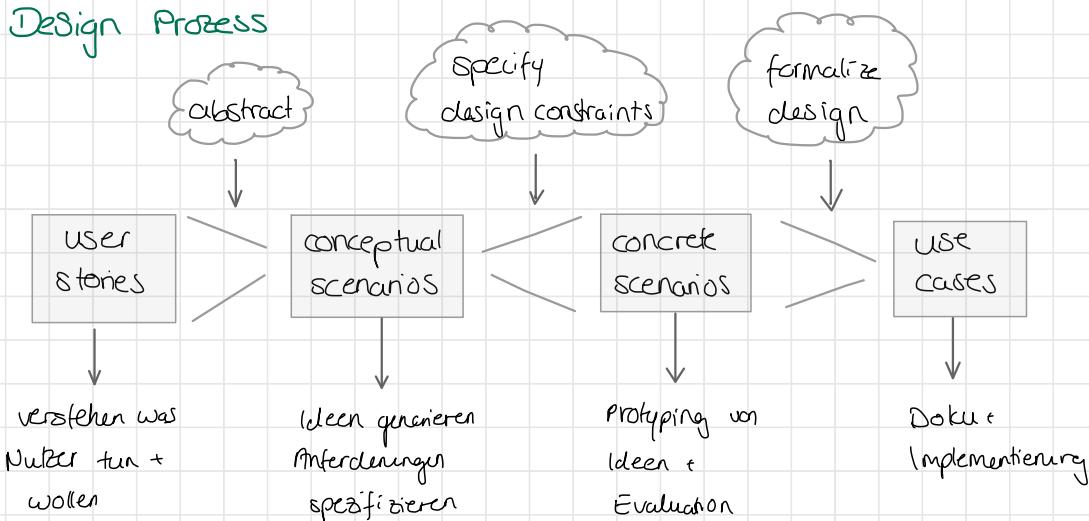
## Szenario-Engineering

1. Requirements definieren
2. Prototyping
3. Envisionment
4. Evaluation
5. konzeptuelles Design
6. physisches Design

## Doku + Rationale

• Gründe für Design Entscheidungen explizit dokumentieren

## Design Prozess



## User Stories

- Problem
  - Stakeholder
  - Constraints
- } identifizieren

## Conceptual Scenarios

- abstrakter als User stories
- kombinierte mehrere user stories
- Abstraktion: Klassifikation + Aggregation
- Nutzen für
  - Ideen generieren
  - Requirements verstehen
- ohne Technologie
- keine Spezifikation des WIE

## Concrete Scenarios

- 1 conceptual sc. => ≥1 concrete scen.
- Vorschläge für konkrete UI Designs
  - ~ Funktionen zw. Nutzer, Geräten
  - ~ Ausgangspunkt für Prototyping + Evaluation

## Use Cases

- beschreiben Interaktion zw. User & Geräten
- 1 use case aus ≥1 concr. scen. → Abstraktion
- brauchte Tasks + Funktionen
- Summe aller Use Cases = System Design

## Szenarienbasiertes Design

- Finde Schwierigkeiten durch Analyse + Abstraktion  
↳ formative Design Prozess
- Finde Requirements
- Priorisiere Requirements

## Umfang

Ziel: Repräsentatives Set von Szenarien → sollen System möglichst gut abdecken

## Conceptual Model / Design

- Objekt / Daten Modell, umfasst
- Szenarien + Analyse
- Hauptobjekte, Attribute, Beziehungen dazu.  
→ sorgt für akkurate mentale Modelle des Systems

## Design Sprache

- Standard Set von Patterns für Interaktionen

## Requirements

Requirements Analyse = verstehen...

- ... Was Nutzer tun
  - ... was Nutzer tun möchten
  - ... der Probleme mit vorhandenen Systemen
  - ... wie Nutzer tun was sie tun
- guter Ausgangspunkt: user stories → für user requirements

## Requirements Specification

formale + präzise Dokumentation der Req.

## Typen

funktional = was muss System tun

nicht-funktional = welche Qualitäten muss System haben  
+ wie operiert die Funktion

! noch keine Technik → nur usage-how-to, kein implementation - h + t

## Priorisierung

· nicht alle Req. umsetzbar → priorisieren + gute Balance finden

## MOSCOW

Must have → sonst System nicht nutzlich

Should have → geht auch ohne, aber wenn noch Zeit gut

Could have → weniger wichtig

Want to have

## Elicitation

Techniken um Req. zu finden:

- Interviews
  - Observations
  - Focus Groups
  - ...
- } user-centered!

## Envisionment

- Ideen sichtbar machen durch materialisieren der Gedanken
- ↳ nutze Szenarien
- nur Darstellung, ohne User-Interaktion

Szenario übertragen → Prototyping → umsetzen Szenario in Figure

· wähle Szenarien für Demo:

- die Häufigen
- die Wichtigen, z.B. sicherheitskritische!
- die Unklaren (Requirements unklar)

## Prototyping

- Design Ideen sichtbar machen
  - User Interaktion simulieren
  - wichtiger Teil des Design Prozesses → schneller + billiger als System neu machen
- ! Prototyp wegwerfen, NICHT als Grundlage für Impl.

### HiFi Prototypen

- look and feel gleich wie antizipiertes System

Nutzen für:

- Wichtige Stages im Design Prozess, zB user acceptance test
- wenn Ideen, Req., Feat. stabilisiert
- detaillierte Evaluation von Design Elementen (zB usability studies)

⊖

- Nutzer glauben Prototyp ist exakt wie finaler System
  - ↪ Details beachten
- Prototyp ≠ Beweis dass System funktioniert
- teuer → größere Hemmungen System neu zu Designen

### LowFi Prototypen

- Fokus auf Ideen, Inhalt, ...
  - ↪ schnelles Evaluieren von Design Ideen

## Papier

- ideale erste Prototypen für screen-based Systems

→ Interaktion: Designer spielt System

- + User guidance = Hilfe um Prototyp zu verstehen

# INTERFACE

## WIMP Interface

Windows

I cons

M enues

P ointers

## Window Managers

2 Arten:

- OS

- User → minim./maxim./resize / access / organisation der windows

- verantwortlich für einheitliches Look-and-feel

## Window Arrangement

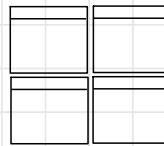
### > Full Size



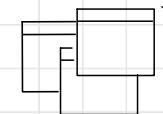
+ weniger komplex

- braucht einfache Navigation zw. Windows

### > Tiled



### > Overlapping



+ effiziente Nutzung des Platzes

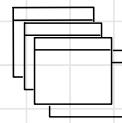
- wird schnell überfordernd

### > Cascade



+ effiziente Platznutzung  
visuelle Organisation

### > Interrupted Cascade



### > Tabbed



# Text

## Lese prozess

### 2 Schritte

1. Buchstaben / Wörter unterscheiden nach Formen
2. Bedeutung assoziieren

## Augenbewegungen

- Saccades: schnelle ruckartige Bewegungen, ~ 8-10 Buchstaben
- Fixations: Pausen in interessanten Bereichen  
↳ vermehrt bei schweren Texten

## Upper-/Lowercase

- Einzelne Buchstaben: Uppercase besser
- Fließtext: Lowercase besser  
↳ besser unterscheidbare Formen

## Papier vs Bildschirm

- Assoziationen mit räuml. Info → wir wissen oft wo etwas steht auf einer Seite  
~ schwierig bei Bildschirmen

## Text in UI

Commentary = Text der informiert = Help-Text

- Contextual help: in Kontext eingebunden, zB Popups
- procedural help: nötige Schritte um Task zu erfüllen
- reference help: (online) reference book
- conceptual help: Hintergrundinfos, Feature Übersicht, ...

Instrumental = Text der "arbeitet"

- Controls: Funktion + deren Label  
zB Buttons, Checkboxes, Icons, Hyperlinks, ...

## Design Fragen

- Lesbarkeit: Alter, Context, Größe, Kontrast, ...
  - ↳ Verständlichkeit abh. von Zeilenlänge, Spacing, Formatierung, ...
- Physische Faktoren: Schriftart / -größe, Kontrast, Scrolling, ...

## Größe

- 9-12 gleich gut lesbar
- abh. von
  - Leserentfernung
  - Screen Resolution
  - Text / Hintergrund Kontrast
  - Sehfähigkeit des Nutzers
  - Art des Lesens (Wort für Wort / Scanning / ...)

## Zeilenlänge

- beeinflusst Performance, nicht Verständnis
- längere Zeilen schneller gelesen
- Medium Length bewirkt

## Margin

- kurze Zeilen + große Margins = bessere Performance

## Zeilenabstand

- double spacing = bessere Performance

## Alignment

- Fließtext: linksbündig / blocksatz

## Kontrast

! altersabh.

Luminanzkontrast wichtiger als Farbkontrast

## Scrolling / Paging

- abh. von Applikation
- Scrolling urteil: kogn. Verankerung als zusammengehörig

## Fonts

- serif E
  - für gedruckten Fließtext
  - ↳ führen Leser
- sans-serif E
  - einfachere Erkennung → für schnelle Erkennung gut zB Praxis
- Cursive
  - handschriftlich
  - erfordert hohe Auflösung
- variable vs. fixed width
  - ↳ beides OK, abh. von Anwendung

## Information

### Information Architecture

- strukturiert Infos die präsentiert werden
- ↳ beeinflusst Konzeptualisierung des Nutzers

Ontologien = Konzept der Information

Taxonomie = Gruppierung / Klassifikation der Dinge

## Strukturierung

### Coarse-grained

- viele Objekte zur Auswahl  
↳ weniger Schritte bis gesuchtes Objekt

### Fine-grained

- multi-step Prozess

## Bildung einer Struktur

### - Lebensdauer

- wähle eine stabile Ontologie

### - Größe

- Scrolling vs. forward - next

### - Conceptual / physical mapping

- anpassen für alle input/output Geräte  
↳ Größe der klickbaren Elemente

### - Topologie

- wie einfach kann ich durch Infos gehen
- Distance = # Klicks nötig zur Navigation
- Direction = nur vorwärts? beide Richtungen?

## Interaktion

### Styles

- CLI
- Menüs
- Form Fill-In
- Q A = Wizards
- Direkte Manipulation / Metaphern
- Web Navigation
- 3D Umgebungen
- Zoombare Interfaces
- Natürliche Sprache

## Command Line

- Info recall statt recognize
- gutes Nutzungskonzept nötig

- ⊕ · steile Lernkurve → besser für Experten  
· einfachere Wiederholbarkeit von Tasks  
· schneller für komplexe Tasks

- ⊖ · hohe Fehlerrate  
· höhere kogn. Last

## Menüs

- Funktionen recognized, Pos./Struktur recalled
- besser für seltene Funktionen
- selbsterklärend
- mögliche Optionen angezeigt → Constraints

## Form Fill In

- zum Sammeln von Infos
- linear
- wichtig:
  - Nutzer muss wissen wie lange Form ist, wo er gerade ist
  - Fehler sind nervig + problematisch

## Wizards = QA

- Forms mit Flow → für Anfänger
- präsentiert nur kleinen Teil der Info auf einmal  
↳ limitierter Control Flow

## Object / Action Modell

- User wählt Objekt → dann Aktion auf Objekt

## Action / Object Model

- User wählt Aktion → dann Objekt auf dem auszuführen

## Direkte Manipulation

- Repr. von möglichen Objekten + Aktionen mit sinnvollen visuellen Metaphern
- Schnelle, inkrementelle, umkehrbare Aktionen mit direkt sichtbarem Effekt

## Touch

- Content ist das Interface













































