bZusammenfassung

Interface and Interaction Design
WS 2011/2012

Punkteaufteilung:

- Beispiel 1: 10 Punkte
- Beispiel 2: 30 Punkte
- Beispiel 3: 30 Punkte
- Test: 30 Punkte
- Summe: 100 Punkte

Notenschlüssel:

- Sehr Gut ab 87.5 Punkte
- Gut ab 75.0 Punkte
- Befriedigend ab 62.5 Punkte
- Genügend ab 50.0 Punkte

1.Vorlesung: Grundlagen & Designkonzepte

Grundlagen:

Interface & Interaction Design:
Der Gesamtprozess, mit dem Art und Mittel der Interaktion zwischen Mensch und Maschine gestaltet werden.

User Interface:
Die Gesamtheit aller Mittel mit denen Menschen (die User) mit einer Maschine, einem Gerät, einem Computerprogramm oder einem anderen komplexen Tool (dem System) interagieren.

Interaktion: Input / Output
Mensch→Computer: Bewegung, Sprache, Geräusche, (...)
Computer→Mensch: Sehen, Hören, Fühlen, (...)

Tool vs. Interface:
Tool ist z.B. ein Drucker, Interface dient dazu Tools zu bedienen (sind aber selbst eigentlich auch Tools: Maus, Bildschirm, Tastatur,…)

Funktionalität vs. Bedienbarkeit:

- Hersteller versuchen sämtliche Funktionen zu integrieren
- Darunter leiden auch die elementaren Funktionen

- Trade-off: Man kann so viel damit machen, dass die einfachen Dinge unnötig kompliziert werden (Bsp.: Fernbedienungen, Telefone,…)

Herangehensweisen:

- Accidental:
  Design by Accident, Design ohne Designer

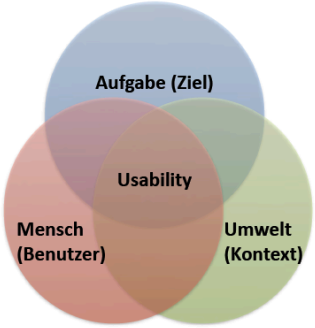
- Aesthetic:
  Ästhetik und angenehmes Erscheinungsbild:
  - Ist das User Interface visuell ansprechend gestaltet?
  - Erweckt das Design einen guten ersten Eindruck?
  - Entspricht das Erscheinungsbild den Erwartungen der Benutzer?
  - Werden gängige Konventionen der visuellen Gestaltung eingehalten?

  Ästhetik und Usability:
  Affekt beeinflusst wie wir Probleme und Tasks lösen.
  „Products and systems that make you feel good are easier to deal with and produce more harmonious results“ (Donald Norman)

  Identität & individueller Charakter:
  Wenn ein Produkt ein ganz anderes Design als Andere hat. Dies kann dazu verwendet werden um Einzigartig zu wirken. (Microsoft Office, Facebook,…)

- Usable:
  Gute Benutzbarkeit des Systems:
  - Erreichen die Benutzer ihre Ziele?
  - Erreichen die Benutzer ihre Ziele schnell und effizient?
  - Ist das System intuitiv bedienbar?
  - Ist das System einfach zu erlernen?
  - Machen die Benutzer wenige Fehler und können sie die Fehler leicht rückgängig machen?
  - Sind die Benutzer mit dem System zufrieden?

  - Usability ist (mehr oder weniger) klar definiert
  - Unterschiedliche Qualitätsmerkmale von Usability lassen sich beobachten und messen:
    - z.B. Effektivität, Effizienz, Zufriedenheit



„Usability eines Produktes ist das Ausmaß, in dem es von einem bestimmten Benutzer verwendet werden kann, um bestimmte Ziele in einem bestimmten Kontext effektiv, effizient und zufriedenstellend zu erreichen.“ (ISO 9241-11)

Kontext:

- Wie passt ein Produkt in eine bestimmte Situation im Alltag und Lebenswandel des Benutzers?
- Umgebung und Situation, in der ein Produkt benutzt wird:
  - Ort
  - Zeit
  - Aktivität
  - Soziales Umfeld
  - ...

Beispiel: Firefox 3 Location Bar
Location Bar in Firefox 3 durchsucht automatisch die Bookmarks und Browsing History, um dem Benutzer Vorschläge zu machen.
Studie: Firefox 2 Updateverhalten

- ~5000 Teilnehmer
- Vorwiegender Grund nicht auf Firefox 3 umzusteigen: Neue Location Bar und damit verbundene Sichtbarkeit der History

Probleme des Firefox 3 Location Bar:
Firefox 3 Location Bar ist grundsätzlich ein nützliches und hilfreiches Feature...
... kann aber unter gewissen Nutzungsbedingungen zu Problemen führen:

- Viele Menschen haben Bookmarks oder Browser-History Einträge, von denen sie nicht wollen, dass andere sie sehen
- Viele Menschen teilen sich einen Computer mit anderen Menschen
- Viele Menschen zeigen anderen Benutzern Dinge auf ihrem Computer oder präsentieren damit in der Öffentlichkeit

→Probleme ergeben sich erst aus einer Kombination von persönlichem Nutzungsverhalten und sozialem Umfeld der Benutzung:

- Arbeitet man ausschließlich alleine vor seinem eigenen Computer, dann gibt es keine Probleme
- Hat man keine Bookmarks, die man geheim halten möchte, dann gibt es keine Probleme
- Lösung des Problems in Firefox 3.5: man kann die Location Bar deaktivieren!

"When people research technology [they often start at] the point at which someone picks up the telephone or starts typing on the keyboard. For me that's already far too down in the process. You want to know; Where does that PC live in someone's home? How did they acquire it? What else is around it? And even one step back further than that: What do people care about? What motivates them? What gets them up in the morning? What do they do when they get up in the morning?" – Genevieve Bell

Wieso scheitern User Interfaces?

"In most people's vocabularies, design means veneer. It's interior decorating. It's the fabric of the curtains of the sofa. But to me, nothing could be further from the meaning of design.

Design is the fundamental soul of a human-made creation that ends up expressing itself in successive outer layers of the product or service." – Steve Jobs, Apple CEO
"Good design is the most important way to differentiate ourselves from our competitors."- Yun Jong Yong, Samsung CEO
"We are the second or the third best design house in the world when it comes to mobile phones. The trick of design is it is not just styling but also great engineering." – Horace Luke, HTC CIO

Designer sind keine Benutzer:

- Designer...
  - sehen Dinge anders als die Benutzer
  - wissen nicht, was die Benutzer wissen
  - wollen nicht, was die Benutzer wollen
  - arbeiten nicht so, wie die Benutzer arbeiten
- Designer treffen Annahmen über Bedürfnisse und Verhaltensweisen der Benutzer
- Diese Annahmen sind nicht immer richtig

Was machen UI Designer falsch?

- Verstehen nicht die Wünsche und Ziele der Benutzer
- Verstehen nicht die physiologischen und kognitiven Grenzen der Benutzer
- Schaffen es nicht die Funktionalität zu kommunizieren
- Bieten dem User kein gutes mentales Modell für die Funktionsweise von Geräten
- Geben kein ausreichendes Feedback
- Beachten nicht den Kontext in dem das Produkt verwendet wird

The Myth of Human Error:

- Die Schuld an Bedienungsfehlern lässt sich meist (zu) einfach auf die Benutzer abwälzen, aber...
- ... spätestens wenn man beobachten kann, dass sehr viele Benutzer dieselben Fehler machen, dann steht fest, dass das Design schlecht ist.
- Benutzer geben sich meist (zu unrecht) selbst die Schuld, wenn etwas nicht wie erwartet funktioniert
- Viele sogenannte „menschliche Fehler" sind in Wirklichkeit Design-Fehler
- Menschen sind unvollkommen und unvorhersehbar
  - Wir haben ein schlechtes Gedächtnis
  - Wir sehen nicht was wirklich vor unseren Augen ist
  - Wir sagen nicht das was wir wirklich meinen
  - Wir werden schnell verwirrt wenn zu viele Dinge auf einmal passieren
  - Wir werden müde, gelangweilt und unaufmerksam
  - Wir lassen uns leicht ablenken
- UI Designer müssen diese Dinge berücksichtigen

Wie sich Benutzer verhalten:

User Behavior Patterns:

1. Safe Exploration
   - Wenn man keine Angst vor negativen Konsequenzen haben muss wird man experimentierfreudiger und lernt mehr
   - Es sollte immer einen einfachen Weg zurück zu einem sicheren, bekannten Ausgangspunkt geben („Undo“, „Redo“)
   - Ermöglicht Lernen durch „Trial & Error“

2. Instant Gratification
   - „Ich will jetzt etwas erreichen, nicht erst später“
   - Benutzer wünschen sich schnelle erste Erfolge
   - Schnelle Erfolge festigen das Selbstvertrauen
   - Benutzer sind eher geneigt, die Software weiterzuverwenden, auch wenn es später schwieriger wird

3. Deferring Choices
   - Benutzer wollen möglichst rasch ihre Arbeit erledigen und ihr Ziel erreichen
   - Sie wollen dabei nicht von (scheinbar) unnötigen Fragen oder Entscheidungen abgehalten werden
   - ... besonders dann nicht, wenn ihnen die Antworten auf die Fragen oder die Konsequenzen ihrer Einstellungen nicht bekannt sind

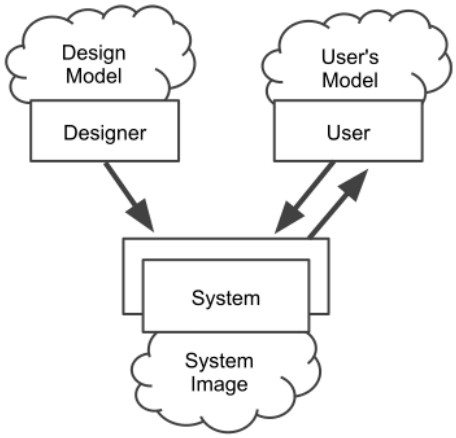
4. Satisficing
   - Benutzer haben oft kein Interesse daran, den „besten“ oder „richtigen“ Weg zu lernen bevor sie handeln
   - Menschen geben sich oft mit „gut genug“ anstatt dem Besten zufrieden, wenn die Alternativen Zeit oder Mühe kosten

5. Changes in Midstream
   - Wir lassen uns leicht ablenken, werden häufig unterbrochen und ändern ständig unsere Meinung
   - Gute User Interfaces müssen es dem Benutzer ermöglichen, seine Arbeit zu unterbrechen und ihn dabei unterstützen, sie später wieder aufzunehmen
   - Geleistete Arbeit darf dabei nicht verloren gehen

6. Habituation
   - Wenn ein Benutzer ein Interface oft verwendet, werden gewisse Operationen zur Gewohnheit
   - Man muss über gewisse Operationen nicht mehr nachdenken
   - Gewöhnung erfordert Konsistenz
   - Schlechtes Bsp: Die Tastenkombination, die bei „normalen“ Computern ein @Zeichen erzeugt, schließt bei einem Mac die gerade offene Anwendung→muss das sein?

7. Spatial Memory
   - Man merkt sich Objekte und Operationen nicht nur anhand ihres Aussehens oder Beschriftung, sondern auch anhand ihrer Position

Designkonzepte



1. Konzeptuelles Modell
   - Die Benutzer haben ein mentales Modell von der Funktionsweise von Dingen
   - Modelle erlauben uns die Operation von einem Gerät mental durchzuspielen
   - Dieses Modell entsteht durch
     - Affordances
     - Constraints
     - Mapping
     - Konsistenz
     - Feedback
   - Schlechtes Design kann zu falschen mentalen Modellen führen

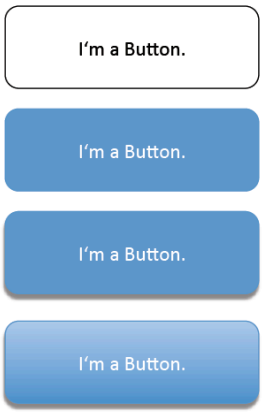
   „If your program model is non trivial, it‘s probably not the user model.”

   “A user interface is well-designed when the program behaves exactly how the user thought it would.” (Joel Spolsky)

2. Affordances
   - „A situation where an object's sensory characteristics intuitively imply its functionality and use.“
   - Der Aufforderungscharakter, den ein Objekt aufweist.
   - Das Aussehen sollte die Verwendung des Objektes und Handlungsmöglichkeiten des Benutzers suggerieren.

   - Komplexe Dinge benötigen oft eine Anleitung
   - Einfache Dinge sollten keine benötigen
   - Brauchen einfache Dinge zur Erklärung ihrer Funktion Labels oder Instruktionen, so ist das Design meist gescheitert

   Visuelle Affordances in Software:
   - Warum sind Visual Affordances für Software UserInterface Design wichtig?
     - Software Interfaces sind virtuell und weisen daher keine physische Affordance auf
     - Die Affordance von solchen Interfaces wird besser beschrieben als „wahrgenommene Affordance“



3. Constraints
   - Grenzen die möglichen Aktionen ein, die an einem Interface ausgeführt werden können.
   - Verhindern die Selektion von ungültigen Optionen.

- Verschiedene Typen:
  - Logical Constraints
  - Cultural Constraints
  - Physical Constraints

Physical Constraints:
Physische Objekte schränken die Möglichkeit an Bewegungen/Aktionen in der realen Welt durch ihre Eigenschaften ein. (verschiedene Formen von Kabelanschlüssen,…)

Poka-Yoke Prinzip:
- = Fehler-sicher (in der Bedienung)
- Verhindern von Fehlern durch Einschränkung der Möglichkeiten, wie eine Operation ausgeführt werden kann.
- Ursprung: Toyota Produktionssystem (Shigeo Shingo)
  - Contact Method
  - Fixed-Value Method
  - Motion-Step Method
- Beispiel PC: 3.5“ Floppy Disk

Prozess-Constraints:
- Nach der Geldausgabe wird das Arbeitsgedächtnis gelöscht
- Bankomat wirft zuerst die Karte aus und wartet, bis sie entnommen wurde, bevor das Geld ausgegeben wird (damit Leute Karte nicht vergessen)

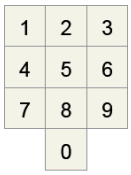
4. <u>Mapping</u>
   Beschreibt die Beziehung zwischen Kontrollelementen und ihrer Bewegung und den Resultaten in der Welt.
   Bsp. Herd: welche Knöpfe sind für welche Platten?

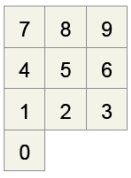
5. <u>Konsistenz</u>
   - Interfaces sollen ähnliche Operationen und ähnliche Elemente für ähnliche Tasks verwenden.
   - z.B.: Shortcuts werden immer mit Control Key plus dem ersten Buchstaben des Kommandos aktiviert - Ctrl+O, Ctrl+S, Ctrl+C, etc.
   - Vorteil: Konsistente Interfaces sind einfacher zu erlernen und benutzen.
     - z.B.: Windows User Experience Guidelines, Apple Human Interface Guidelines

“If cars were designed like computers, every time a new car was introduced car buyers would have to learn how to drive all over again because none of the controls would operate in the same manner as the old car.”

Beispiel für Inkonsistenz:

Telefon, Mobiltelefon, Fernbedienungen	Taschenrechner, Computer Keypad





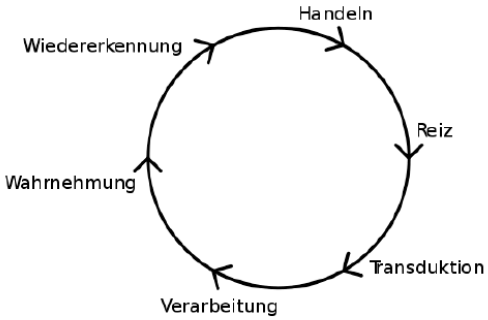
6. Feedback
    - Zurücksenden von Information an den User über
        - das Ergebnis einer Aktion
        - darüber, dass eine Aktion ausgeführt wird/wurde
        - den aktuellen Status und welche Operationen möglich sind (Feedforward)
    - „Tote“ Screens sind frustrierend für den User
    - Mithilfe von: Sound, Highlighting, Animation, etc.

    - Daumenregel:
        - 0,1 Sek: kein spezieller Indikator notwendig
        - 1,0 Sek: User verliert das Gefühl von direkter, unmittelbarer Interaktion
        - > 10 Sek: User verliert Fokus, Status Indikator benutzen
    - Im Endeffekt: Was sich richtig „anfühlt“

    - Feedback muss verständlich sein (schlechtes Bsp.: Unverständliche Windows-Fehlermeldung)
    - Feedback muss zum richtigen Zeitpunkt den richtigen Empfänger erreichen (schlechtes Bsp.: bei Benutzung durch Kunden tritt Windows-Fehlermeldung bei einem Bankomaten auf)

2.Vorlesung: Visuelles Design & Web Design

Gestaltprinzipien



Motivation:

- Menschliche Wahrnehmung als komplexer und individueller Prozess
- Psychologie bietet Ansätze zur Analyse menschlicher Wahrnehmung
- Psychologische Beobachtungen als Motivation für Designguidelines
- Allerdings keine einheitliche Wahrnehmungstheorie
- Keine „Theorie des guten Designs“

Allgemeines:

- Teil der Gestaltpsychologie
  - Beschreibt die menschliche Wahrnehmung als Fähigkeit Ordnung und Strukturen aus Sinneseindrücken abzuleiten
- (Berliner) Schule der Gestaltpsychologie: Max Wertheimer, Kurt Koffka, Wolfgang Kohler, u.a.
- „Das Ganze ist verschieden von der Summe seiner Teile”
- Deskriptive (nicht erklärende!) Theorie

Prinzip der Nähe:

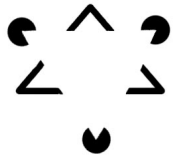
- Näher zusammen liegende Elemente werden als zusammengehörig empfunden
- Eines der wirksamsten Prinzipien
- Eine der einfachsten Möglichkeiten um eine enge Beziehung zwischen Elementen auszudrücken

Prinzip der Ähnlichkeit:

- Gleiche oder fast gleiche Elemente werden als zusammengehörig empfunden
- Farbe gruppiert stärker als Form

Prinzip der Geschlossenheit:

- Die visuelle Wahrnehmung erkennt einfache, mit den vorhandenen Erfahrungen am besten zu vereinbarende Formen
- Geschlossene Formen werden bevorzugt wahrgenommen

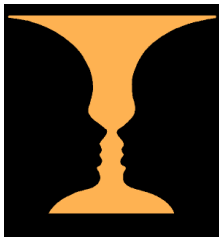


Prinzip der stetigen Fortsetzung:

- Bei mehreren Möglichkeiten wird immer die einfachste und regelmäßigste Variante bevorzugt
- Konturen mit kontinuierlichem Verlauf

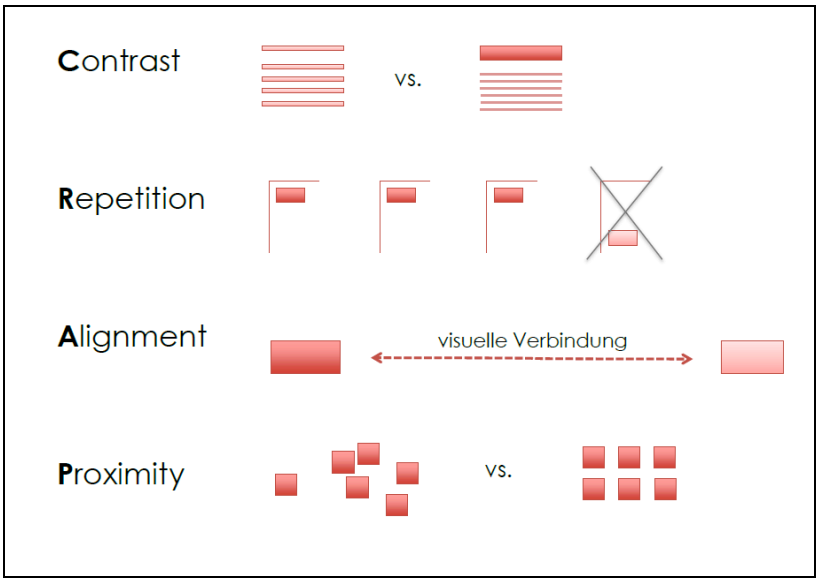
werden gegenüber Konturen mit gebrochenen Verlauf oder abruptem Richtungswechsel stets bevorzugt

Prinzip der Prägnanz:



- Figur vs. Hintergrund
- Manche Objekte treten in der visuellen Wahrnehmung in den Vordergrund
- Üblicherweise kann nur eine „Interpretation“ wahrgenommen werden

Anwendung der Gestaltprinzipien



- Darstellung des Design-Prozesses einer typischenWebseite
- Illustrative Anwendungvon Alignment, Proximity,Contrastund Repetition

Seite inhaltlich vollständig jedoch:

- Wenig Struktur
- Elemente schlecht gruppiert
- Bereiche wirken unzusammenhängend und optisch gleichwertig
- Keine visuelle Hierarchie
- Auge wird nicht geführt

→Gestaltprinzipien

Alignment:

- **Kein Element** der Seite darf **beliebig** platziert werden →Jedes Element muss eine **visuelle Verbindung** mit einem anderen Element besitzen.
- Vertikale und horizontale Ausrichtung der Elemente erzeugen kohäsive **Einheit**

- Ausgerichtete Elemente obwohl physisch voneinander entfernt →verbunden und verwandt
- Ausrichtung an einem Grid

Vorher

Nachher

Proximity:

- Inhaltich zusammen-gehörende Elemente befinden sich in **physischer Nähe**
  - Gruppierung
  - Ordnung
  - Kohäsive Einheit

→**Nähe** impliziert **Zusammengehörigkeit**

- Entfernung zu nicht zusammengehörenden Elementen groß genug und **eindeutig**

Contrast:

Unterscheiden sich Elemente in Farbe, Form, Größe etc. entsteht Kontrast.

- Rund VS. eckig
- Groß VS. klein
- Statisch VS. dynamisch
- Hell & Kalt VS. Dunkel & Warm

→Erweckt **Aufmerksamkeit**
→Schafft **Ordnung** und **Hierarchie**
→Kontrast muss **eindeutig** und **erkennbar** sein!

Repetition:

- Ein grafisches Konstruktoder eine Eigenschaft wird im gesamten Design wiederkehrend verwendet
- Wiederkehrende Elemente sowohl innerhalb einer Seite
  - Aussehen von Links/Navigationselementen
  - Elemente zur Strukturierung von Inhalt (Lightboxen)
- Als auch Seitenübergreifend
  - Navigation
  - Inhaltsbereich
  - Footer

Farbenlehre

Farbe:

- Elektromagnetische Welle/Teilchen
- 400nm-800nm

Lichtfarben:

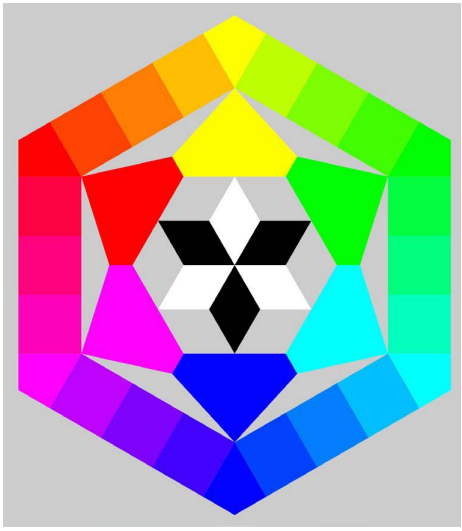
- Additiver Farbmischung
- Grundfarben
  - Rot
  - Grün
  - Blau
- Photosensitive Zellen im Auge
- Alle anderen Farben →Mischung
- Alle drei Farben in gleicher Intensität gemischt →Weiß →Additiv

Pigmentfarben:

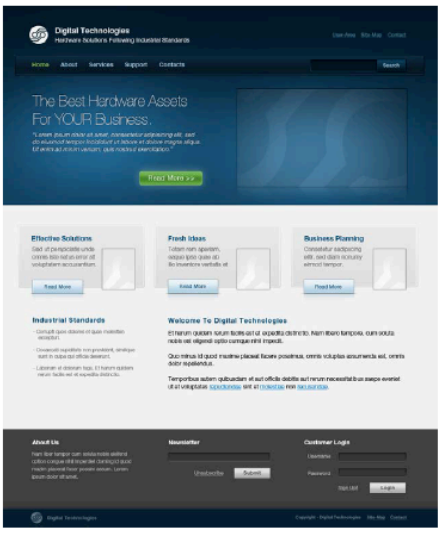
- Reflexion bestimmter Anteile weißen Lichts
- Farben reduzieren Lichtspektrum →Subtraktive Farbmischung
- Grundfarben
  - Cyan
  - Magenta
  - Gelb
- Anwendung im Druck
  - Zusätzlich Schwarz
  - CMY**K**

Farbkreis nach Harald Küppers:

- Komplementär-Farben
- Farben mischen (subtraktiv)
- Farbharmonien
    - Harmonischer Dreiklang
    - Komplementär
    - Analog



Ergebnis nach „Anwendung der Farbtheorie“:

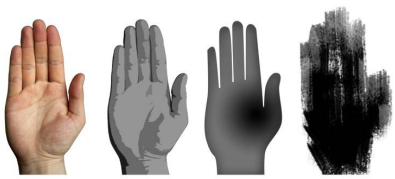


Bilder

- Symbolisch-emotionaler Raum
- Betrachtung ←→physikalische Erfahrungen im Kopf des Betrachters
- Verbindung zu textuell beschriebenen Erfahrungen
- Leichteres Verständnis komplexer Inhalte
- Wirkung des Bildes vom Kontext abhängig →Mediation

Darstellungsmodi:

- Repräsentativ
- Ikonographisch
- Abstrakt



Symbol:

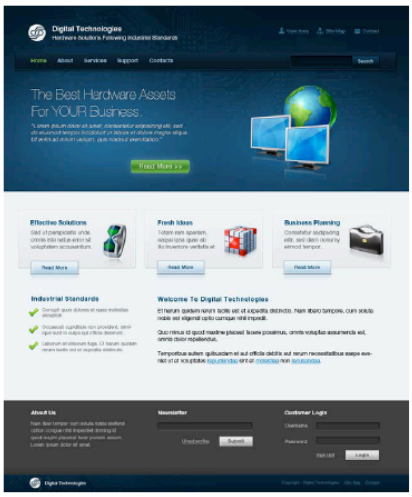
- Abstrahiertes Bild
- Form und physisches Aussehen hängen nicht mit der Idee oder der Bedeutung zusammen

- Sinn von der kulturellen/sozialen Umgebung abhängig – muss gelernt werden

Icon:

- Optisch-abstrahierte Darstellung des Objekts

Ergebnis nach „Anwendung der Bildtheorie“:



Typografie

Geschichte:

- Erste standardisierte Schreibsysteme ca. 3500 v. Chr.
  - Keilschrift der Sumerer
  - Hieroglyphen der Ägypter
- Ideogramm
  - Begriff einer Sache nicht durch Laute sondern durch dessen Abbildung wiedergegeben
  - 1:1 Relation zwischen Idee und Symbol
  - Kein direkter Hinweis auf Aussprache

- Phönizisches Alphabet (1000 v.Chr.): Grundlage für Griechische, Hebräische, Lateinische und Arabische Zeichen
- Symbole stellen keine Objekte oder Aktionen sondern Geräusche dar
- In kombinierter Form →Wörter

Klassifizierung:

- Deutscher Sprachraum: DIN 16518
  - Schriften werden historisch geordnet
  - 11 Gruppen
  - Problem 90% der neuen Schriften in einer Gruppe
  - →Norm wird momentan überarbeitet

- Alternative
  - Schriftklassifikation nach Formprinzip (Indra Kupferschmid)
  - Nicht genormt

Klassifizierung nach Form:

1. Merkmal:
   - Serifen vs. Serifenlos
   - Strichkontrast

2. Merkmal

→Formprinzip:

- Dynamisch
  - Schräge Kontrastachse
  - Organische Form
- Statisch
  - Gerade Kontrastachse
  - Hoher Strichkontrast
  - Versalien ähnlich breit
- Geometrisch
  - Konstruierte Formen
  - und andere Buchstaben sind zirkelrund
  - Wenig Strichkontrast

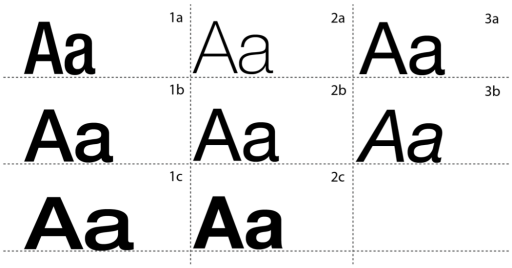
Zusätzlich:

- Schreibschriften
- Freies/dekoratives Formprinzip →Grafische Schriften

Beispiele:

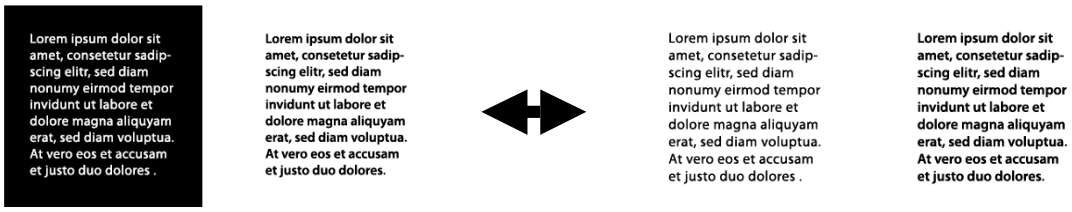
Schriftfamilien:

- Verschiedene „Schnitte“ einer Schrift
- Schriftbreite
- Schriftstärke
- Schriftlage



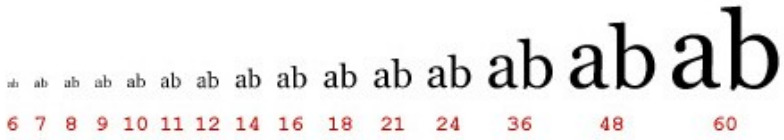
Anwendung:

- Zeilenlänge
  - Zu lange: Auge verliert Zeile
  - Optimal 40-80 Zeichen
- Wichtige Faktoren
  - Schriftart
  - Schriftgröße
  - Schnitt
  - Zeichenabstand
  - X-Höhe
- Zeilenabstand
  - Wichtiges Kriterium für die Lesbarkeit
  - Korrekt angewandt: →Leser kann den Buchstaben folgen und das Gesamtbild des Textes wird verbessert
  - Typografische Farbe
  - Je größer die Schrift desto geringer der Zeilenabstand
  - Für Fließtext im Web 125% oder mehr
- Negativsatz
  - Laufweite erhöhen (Sperren)
  - Schriftstärke verringern



- Schrift niemals verzerren!
  - →Dafür vorgesehenen Schnitt verwenden (Condensed, Extended etc.)
- Korrekte Anführungszeichen
  - Gänsefüßchen: 99 davor und 66 danach „Gänsefüßchen“
  - Französische Anführungszeichen: Guillemets »Guillemets«
- Vertikaler Rhythmus
  - Mithilfe von „Baseline-Grid“
  - Definiert Zeilenhöhe und Abstand zwischen Absätzen
  - konstanter vertikaler typografischer Rhythmus
- Typografische Größenskalierung
  - Für verschiedene Textelemente:Fließtext, Auszeichnungen, Überschriften etc.
  - Bestimmt typografische Hierarchie

    - Harmonie innerhalb der Komposition
- Methoden
    - Traditionell (15. Jahrh.)
    - Fibonacci
    - LeCorbusier



3.Vorlesung: HCI Evolution & Desktop Interfaces

Evolution der Human Computer Interaction

4 Perioden (nach Andries van Dam):

1. Lochkarten:
   Erste Periode: Anfang 1950 - 1970
   Computer mit Lochkarten als Input und Printer als Output → keine direkte Human Interaction

2. Command Line Interfaces
   Zweite Periode: Anfang 1960 bis etwa 1990
   Mainframe und Minicomputer,
   Command Line Interfaces,
   Dialogsystem Metapher

3. Graphische User Interfaces
   Dritte Periode: Anfang 1980 bis heute
   Grafische User Interfaces basierend auf Windows, Icons, Menus und einem Pointing Device (Maus) –
   WIMP GUIs
   Desktop Metapher

4. Post-WIMP Interfaces
   Natural User Interface
   Hier ist noch unklar in welche Richtung die Entwicklung geht, es gibt aber mehrere Ansätze: Multitouch-Displays, Benutzer ist der Controller, …

Recognition vs. Recall:
Recognition: Wir tun uns leichter Dinge wiederzuerkennen/identifizieren als sie uns zu merken → GUI: Dokumente, etc. werden graphisch dargestellt
Recall: Man muss sich an Dinge erinnern, bzw. dem Computer explizit sagen was man will.

Knowledge in the Head:

- Erinnern/Merken von Funktionsabläufen, Dingen oder Ereignissen ist Knowledge in the Head
- Muss erlernt werden, ist aber sehr effizient
- Beispiele:
  - Passwörter
  - Sozialversicherungsnummer
  - Geburtstage
  - Adressen
  - Telefonnummern?
  - E-Mail Adressen?

Knowledge in the World:

- Wissen, das aus der Welt abgeleitet werden kann
- Muss nicht erlernt werden, ist aber weniger effizient, da die externe Information gesucht und interpretiert werden muss
- Beispiele:
    - Terminkalender
    - Knoten im Taschentuch (Signal vs. Message)
    - Labels und Zeichen
    - Natural Mappings (Beispiel: Herdplatten)

Metaphern:

- Vorhandene mentale Modelle nutzen
- *„Indeed, the models that learners spontaneously form are incomplete, inconsistent, unstable in time, and often ripe with superstition.“* (Olson & Carroll)

Probleme mit Metaphern

- Bringen Einschränkungen mit sich
    - ... und können Innovation hemmen
- Werden inkonsistent verwendet
    - ... und sorgen für Verwirrung beim Benutzer
- Sind unvollständig
    - ... und wecken falsche Vorstellungen (Bsp.: Schreibmaschine als Metapher für Microsoft Word)
- Können veralten
    - ... und man wird sie oft nur schwer wieder los (Bsp.: Diskette als Symbol als für speichern)

Inkonsistente Metaphern:

- Papierkorb: Erlaubt das Löschen von Dateien rückgängig zu machen
- Metaphern können problematisch sein, wenn man sie nicht konsequent einhält
- Inkonsistenz-Beispiel: Auswurf von Wechseldatenträgern unter Mac OS

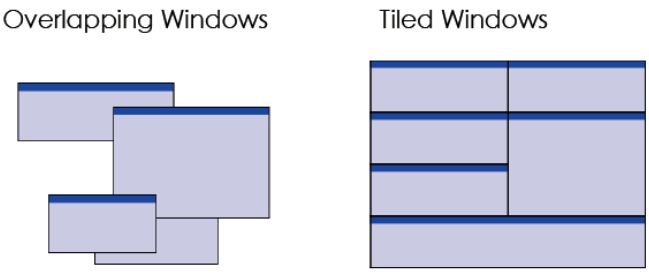
Skeuomorphic UI Design:

Auf dem Computer Dinge nachbilden, die in der Realität so existieren.
Bsp.: Taschenrechner, Bücher

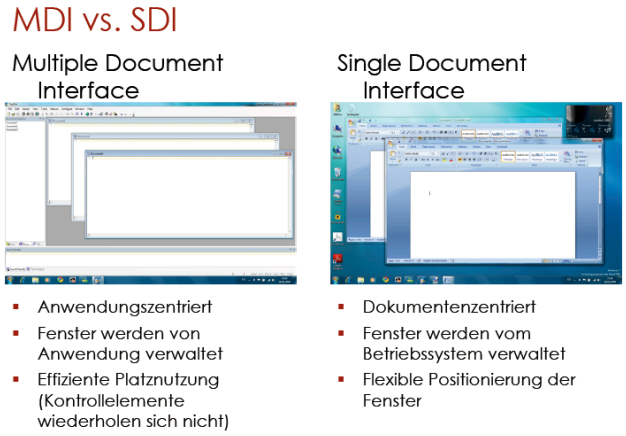
WIMP User Interfaces

- **Windows**

Unterschiedliche Layout Strategien:



MDI vs. SDI:



Probleme von Window Managern:

- Benutzer
  - Benutzer arbeiten meist an mehreren Tasks gleichzeitig
  - Häufige Taskwechsel notwendig
  - Benutzer müssen jedes einzelne Fenster verschieben, vergrößern, verkleinern...
  - Die meisten Fenster werden von anderen überdeckt
- Anwendungen
  - verwenden meist den gesamten Screen
  - verwenden immer mehr Fenster
- „Window Housekeeping“

- **Icons**

  - Symbolische Darstellungen verwenden (keine fotorealistischen Icons)
  - Tool Tips angeben
  - Entweder perspektivische oder flache Icons verwenden (nicht mischen!)
  - Es ist schwierig gute Icons zu designen, daher vorhandene Standard Icons verwenden

- **Menus**

Fitts‘ Law:

- Ein Modell des Bewegungsapparats, welches die Zeit zur Auswahl eines Zielbereichs in Abhängigkeit von Distanz zum Ziel und Größe des Ziels setzt.
- Grob vereinfacht:
  - Je größer ein Ziel ist, desto schneller kann man es auswählen.
  - Je näher ein Ziel liegt, desto schneller kann man es auswählen.

(Ecken sind am besten/einfachsten zu erreichen)

- **Pointing Device**

Hier ist in den Folien nicht wirklich etwas Nützliches/Wichtiges zu finden.

Design Patterns

Richtlinien und Hilfsmittel

- Allgemeine Designkonzepte & Prinzipien
  - Affordances, Constraints, Konsistenz, Mapping, ...
- Design Pattern
  - Web, GUI
- Design Guidelines
  - Windows UX Guidelines, Apple HIG, ...

Konventionen

- Konventionen erleichtern den Benutzern die Orientierung
- Vorwissen von anderen Programmen oder Websites kann übertragen und angewendet werden
- Bruch mit Konventionen ist verwirrend und frustrierend
- Konventionen sind im Allgemeinen sehr hilfreich (ansonsten hätten sie sich nicht durchgesetzt)
- Beispiel für Konventionen im Web:
  - Ein Satz in sehr großer Schrift ist meist eine Überschrift
  - Das Logo links oben ist meist ein Link auf die Startseite
  - Links heben sich durch unterschiedliche Formatierung von Fließtext ab

Design Pattern

- Eine allgemeine Lösung zu einem bekannten (Design) Problem
- Kommen ursprünglich aus der Architektur (Christopher Alexander)
- Anwendung dieser Idee auf Städte, Häuser, Zimmer, Möbel

Struktur eines Pattern nach Alexander

- Name des Pattern mit einem Index (Nummerierung)
- Bewertung des Pattern (0, 1 oder 2 Sterne) – Zuversicht des Autors über die Qualität des Musters
- Bildliche Darstellung des Pattern
- Kontextbeschreibung
- Problembeschreibung
- Gefundene Lösung
- Skizze der Lösung
- Verknüpfung mit anderen Mustern

Pattern in der Softwareentwicklung

- Pattern sind hier bewährte Lösungen, die bei der Entwicklung von Softwaresystemen helfen:
    - Softwareentwicklungspattern – beschreiben gute Implementierungsstrategien
    - User Interface Design Pattern – beschreiben gut benutzbare Komponenten für User Interfaces
    - Web Design Pattern – analog zu UI Pattern beschreiben sie gut benutzbare Komponenten für Websites

Anforderungen an ein Pattern

- Enthält einen konkreten Lösungsweg und die Lösung muss erprobt sein
- Die Lösung muss verallgemeinerbar sein
- Die Lösung darf nicht völlig offensichtlich sein →Sonst ist es kein Pattern
- Es muss so formuliert sein, dass es allgemein verständlich ist
- Umstände, die die Lösung beeinflussen können, müssen dokumentiert sein, um die Lösung ggf. den besonderen Gegebenheiten anpassen zu können

Pattern: Extras On Demand

Zeige den wichtigsten Inhalt zuerst, aber verstecke den Rest. Lass den User diesen Rest mit Hilfe einer einzigen, einfachen Geste erreichen.

Pattern: Responsive Enabling

§ Starting with a UI that's mostly disabled, guide a user through a series of steps by enabling more of the UI as each step is done.

Wozu Design Pattern?

- Ideenlieferant und Grundlage für die Gestaltung eines User Interface bzw. einer Website
- Zum Erlernen von Best Practices
- Zeit und Kostenersparnis, wenn man nicht jedes mal das Rad neu erfinden muss
- Bestehendes Fachwissen verschiedener Designer sammeln und zugänglich machen
- Hilfsmittel zur Kommunikation im Team und als gemeinsamer Ausgangspunkt für Diskussionen
- Konsistenz und Erwartungskonformität

Interface Design Guidelines

Eine Sammlung von Empfehlungen für Designer und Entwickler für die Gestaltung von User Interfaces. (Idee: alle Interfaces einheitlich gestalten, damit User sich leichter zurecht finden)
Guidelines beinhalten in der Regel:

- Allgemeine Design Prinzipien abgeleitet aus wissenschaftlichen Quellen
- Standards (veröffentlicht von großen Softwareherstellern und Plattformbetreibern)
- Style Guides (auf Firmen oder Organisationsebene)

Vorteile von Guidelines

- Sicherstellung von Konsistenz zu anderen Applikationen („visual“ und „behavioral“)
- User können bereits erlernte Fähigkeiten auf neue Interfaces anwenden
- Neue Tasks werden einfacher und schneller erlernt
- Fertige Richtlinien zur Erstellung von User Interfaces
- Spezifische und detaillierte Guidelines lassen wenig Raum für Interpretation und sind deshalb einfach anwendbar

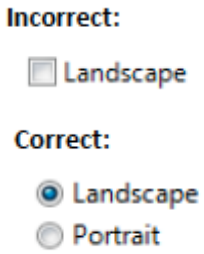
Probleme mit Guidelines

- Können zu allgemein sein
- Können zu spezifisch sein
- Beantworten meist nicht alle offenen Fragen
- Können in Konflikt mit anderen Guidelines stehen
- Konsistenz bedeutet nicht unbedingt gute Usability
- Sind manchmal nicht auf dem aktuellsten Stand
- Guidelines können gebrochen werden, aber man sollte einen guten Grund dafür haben.
- Wenn man Guidelines bricht, dann richtig.

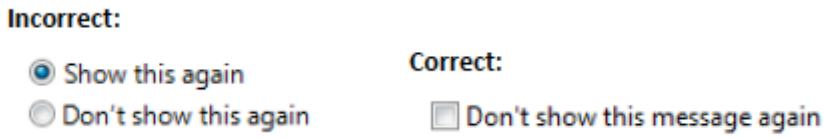
UI Guidelines

- Designprinzipien
- Betriebssystemumgebung
- Visuelles Erscheinungsbild
    - Icons, Schriftarten, Rahmen, Schatten, Perspektiven, Farben, ...
- Menüs, Kontrollelemente & Toolbars
    - Buttons, Formularelemente, Status Indikatoren, ...
- Fenster
- Text & Wording
- User Interaktion und Eingabe
    - Eingabegeräte, Clicks, Shortcuts, ...
- Layout Guidelines
    - Größe, Abstände, Gruppierung, Positionierung, Ausrichtung, ...

Checkboxes und Radiobuttons



- Radiobuttons verwendet man
  - Wenn nur eine Option ausgewählt werden darf
  - Aus einer Menge von mindestens 2 Optionen
  - Genügend Platz für alle Optionen vorhanden ist
  - Die Anzahl der Optionen feststeht und eher gering ist (2-8)
- Alternative: Dropdown-Boxen
- Checkboxen repräsentieren An/Aus-Zustände, die voneinander unabhängig sind



Text und Formulierung

- Formattierung & Wording
- Font und Schriftgröße
- Richtlinien für Sprache, Abkürzungen, Terminologie, Schreibstil
- In der gesamten Applikation eine klare und konsistente Sprache verwenden
  - „Speak the user‘s language“
  - „Present choices in terms of user goals, not technology“

- „Be polite, supportive, and encouraging. The user should never feel condescended to, blamed, or intimidated.“
- Inkonsistente Verwendung von Bezeichnungen:
  - Unterschiedliche Bezeichnung für dasselbe Konzept
  - Selbe Bezeichnung für unterschiedliche Konzepte

Layout und Alignment

- Größe für Formularfelder, Buttons, Statusindikatoren, ...
- Spacing und Positionierung
  - Abstand zu den Rändern
  - Abstand zwischen einzelnen Elementen
- Alignment
  - Entlang eines Grids
  - Relativ zu anderen Elementen
- Reihenfolge von Elementen (links nach rechts & oben nach unten)
- *Center Equalization:* Beide Seiten des Bildschirms sollten ca. gleich „gefüllt“ sein
- *Text Alignment:* Linksbündig damit man besser folgen kann,…
- *Spacing:* Abstände Konsistent halten,…

Dialoge

- Sollten klar verständlich sein.
- Sinnvolle Auswahlmöglichkeiten sollten angeboten werden.
- Ein Button, bei dem Daten verloren gehen, sollte zum „save“-Button mindestens 14px Abstand haben.

Fehlermeldungen

- Sollten klar verständlich sein.
- Möglichst genaue Beschreibung was passiert ist

Keyboard Navigation

- Beim Anordnen der Elemente berücksichtigen, dass der User Maus oder Keyboard zum Navigieren verwendet
    - Fokus setzen
    - Logische Tab-Reihenfolge

Keyboard Shortcuts und Mnemonics

- Verwendung von Keyboard Shortcuts für häufig verwendete Menüpunkte (nicht für alle)
- Verwendung von Standardabkürzungen (z.B. Ctrl für Control)
- Allgemein verwendete Shortcuts beachten und verwenden
    - Ctrl-N, Ctrl-O, Ctrl-S, Ctrl-C, Ctrl-V, etc.
- Mnemonics (Access Keys: ein Buchstabe zur schnellen Auswahl eines Eintrags aus einem Menü)
- Keine Mnemonics für Default oder Cancel Button in Dialogfenstern verwenden (Sondern: Enter und Esc)

4.Vorlesung: Mobile Interaction Design

Mobile Devices

Entwicklung vom Single-Purpose Device zum Multi-Purpose Device.

Damals: Input & Output

Eingabe:

- Tastatur
- Mikrofon (nur zum Telefonieren)

Ausgabe:

- Display
- Beep
- Lautsprecher (der nicht laut ist)

Heute: Input & Output

- Viele Möglichkeiten (Touch-Screen, Kamera,…)
- Rasante Entwicklungen (Bildschirmauflösung wird immer besser,…)
- Input meets Output: Tastatur am Bildschirm während Eingetipptes angezeigt wird

Heute: Input & Sensors

- Touchscreen
- Kamera(s)
- Mikrofon (Sprachsteuerung)
- Accelerometer
- Gyroscope
- Proximity Sensor
- Helligkeitssensor
- Kompass
- GPS
- …

Konnektivität

- Damals:
  - Anruf
  - SMS
- Heute:
  - Breitband Internetverbindung (3G, WiFi)
  - “Always on - Always connected”

Near Field Communication

- Point to Point
- 0,1m Reichtweite
- Automatisches Pairing

Mobile User Experience

User Experience: *How a person feels about using a system.*

Mobile Mindsets

- "I'm microtasking."
  - Kurze Aktivitäten
- "I'm local."
  - Umgebung erkunden
- "I'm bored."
  - Ablenkung / Unterhaltung

Mobiltelefone sind keine Computer!

Computer:

- Relativ vorhersagbare Umgebung
- Großer Bildschirm ermöglicht Multitasking
- Keyboard und Maus für Input

Mobiltelefon:

- Variabler Kontext und Umgebung
- Kleiner Bildschirm und begrenzte Texteingabe
- UI benötigt den ganzen Bildschirm
- Multitasking schwer, leicht zu unterbrechen

Gute Mobile Experiences:

1. **sind einzigartig mobil**

Design Prinzipien:

- Fokus auf die Stärken mobiler Medien
  - „Know the medium you are designing for"
- Fokus auf Bedürfnisse, nicht Lösungen
- Heuristiken und Technologie ändern sich häufig, nicht zu sehr damit aufhalten

Bedürfnis vs. Lösung

- Welchem Bedürfnis entspricht eine Information?
- Beispiel: Adresse eines Restaurants
  - Wie komme ich dorthin?
  - Ist es in der Nähe von ... ?
  - Was sage ich dem Taxifahrer?
  - Kann ich dort parken?

2. **sind kontextbezogen**

"Content is content ... Context is KING." [Michael Kasprow]
→Kontext ist komplex!

Vorteile von Kontext
Steigerung von:

- Usability
  - Information durch weniger Klicks verfügbar
- Relevanz
  - Wo ist der nächste Supermarkt? →räumlicher Kontext
- Interesse
  - Wiederverwenden der Applikation

Design Prinzipien:

- Design für teilweise Aufmerksamkeit und Unterbrechungen
- Reduktion der kognitiven Belastung
- Ideen in der realen Welt sammeln

3. **sind aussagekräftig**

Design Prinzipien:

- Inhalte herunterbrechen - „Bite-Sized" Junks
- An Möglichkeiten, nicht an Tasks denken

Möglichkeiten

- Bei mobilen Applikationen geht es darum, User schnell durch Informationen zu führen
- Es geht darum, Möglichkeiten bereitzustellen, der User entscheidet in welcher Reihenfolge oder zu welcher Zeit.

Mobile IxD Guidelines

"Everything should be made as simple as possible, but no simpler." [Albert Einstein]

Minimize functions

- Auf „Core Use Cases" konzentrieren
- Unnötige Funktionen vermeiden
- Komplexe Aufgaben aufteilen
- Oft besser weniger Funktionen, als die User Experience limitieren

Be consistent

- Konsistenz innerhalb der **Applikation**
- Terminologie, Layout, Element, Icons, …
- Konsistenz innerhalb der **Plattform**

- Erfahrung / mentales Modell des Users
- “Standard Visuals” verwenden
- Tabbar, Toolbar, Dialoge, Icons, …
- Logische Inkonsistenz
- Wenn Konsistenz in einem Fall keinen Sinn macht

Kognitive Last minimieren
- Wieviel Konzentration erfordert die aktuelle Aufgabe?
- Stärken eines Computers nutzen
    - Rechnen
    - Merken
    - Vergleichen
    - Rechtschreibkorrektur

Design for direct access (Dinge auf Touchscreen direkt antippen,...)

Let users “see and point” (Auswahl anzeigen statt Suchfunktion, Kontakte nicht erst anzeigen wenn User Buchstaben eingibt)

Reduce & optimize keyboard input (bei Texteingabe Vorschläge machen, häufig genutzte Buchstaben größer machen,…)

Prevent errors
- Funktionen deaktivieren
- Constraints & Poka-Yoke Prinzip
- Passende Interaktionselemente
- Klare Anweisungen

Detect errors
- Mögliche Fehler erwarten
- Sicherstellen, dass der User weiß:
    - Ich habe getan was ich tun wollte.
    - Was ich getan habe war korrekt.

Offer error recovery
- Klare Anweisungen geben, z.B.
    - „Zurück“
    - „Rückgängig“
    - „Abbrechen“
- Nicht rückgängig zu machendes vorher bestätigen lassen
- Default Werte, die zu weniger Fehlern führen

Verärgere, nerve, belästige den User nicht
- Warnmeldungen reduzieren

    - Wenn App nicht weiterkommt
    - Zur Bestätigung / Zustimmung benötigt
- Nicht für Informationen (oder Rating der App) missbrauchen
- Klare verständliche Sprache
- Dialoge: Default Buttons = Sicherste Option

Use real world metaphors (Schloss „aufziehen“/“sliden“, ON/OFF,…)

Mache dir klare mentale Modelle zu Nutzen
- Physikalische Gesetze
- Animierte Übergänge
- Content vs. Interaktionselemente

Designe für die richtigen Händegrößen/Fingergrößen
- Genug Abstand lassen
- Buttons groß genung machen

Mache Gesten schlau und einfach
Globale Gesten:
- Systemweit verfügbar
- Machen das UI einfach und intuitiv benutzbar

Lokale Gesten:
- Applikationsintern
- Dürfen globalen Gesten nicht widersprechen

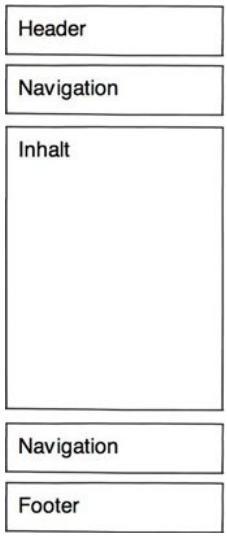
Zusätzliche Gesten:
- Zusätzliche Gestures für Expert-Users verwenden
- Aber: Funktionen müssen auch ohne Gesture durchführbar sein

Gib feedback (visuell, akustisch, haptisch)
- Touchscreen ist nur eine glatte Oberfläche
- Keine Anhaltspunkte/Orientierung
- Benutzer sind oft abgelenkt
- Ohne Feedback muss viel genauer auf das Interface geachtet werden

PRAXIS TIPP
Challenge the client’s thinking
- Konsistenz innerhalb Plattform wichtiger als Konsistenz zwischen Plattformen
- Eigenheiten der Plattformen ausnützen
- Branding (platz)sparend verwenden oder weglassen
- Nicht von Featuritis anstecken lassen

Es gibt auch „**M**obile **U**ser **I**nterface Pattern Libraries“, die das Erstellen von MUIs erleichtern.

Mobile Web



Unterschiede Mobile Web

- Kleiner Bildschirm, wenig Platz
- Hoch & Querformat
- Langsam (Bandbreite, Prozessor)
- Hover?
- Weniger Präzision – kleine “hit targets”
- Gesten möglich
- Nicht immer direkt im Browser
- Java Applets & Flash Support?
- Verbindungsunterbrechungen
- …

Inhalt

- Kleiner Bildschirm, wenig Platz
- Einfach gestalten
- Bandbreite reduziert, wenige Bilder
- Links herausheben

Verschiedene Displays

- Auflösung
- Größe
- DPI
- Orientierung

Browser Features

- Contextual Keyboard („Touchscreen Keyboard“)
- “Context Lite” (Orientierung des Smartphones, Standort, Beschleunigungsmesser,…)

Mobile Stylesheets

1. Serverseitig mit User Agent String
    - Höchste Kompatibilität
    - Spezielle Markups und Content können übertragen werden
    - UA Liste muss aktuell gehalten werden
    - UA Spoofing
2. Clientseitig mit Media Queries
    - Spezielle Markups
    - Verschiedene Stylesheets je nach Medium, Auflösung, etc.
3. User selbst entscheiden lassen
    - Mehr Interaktion erforderlich
    - Sicherster Ansatz
    - Entscheidung merken

Mobile Web - Tipps

- Normale und mobile Version anbieten
  - User entscheidet
  - Mobile-Version kommt per Default
- In mobiler Version "Core Use Cases" gut abbilden, unnötige Funktionen weglassen
  - Top 20% der Funktionalität
- Normale Version für restliche Nutzung

Prototyping

Programme zur Erstellung eines Prototypen:

- Balsamiq Mockups
- Mockflow
- Flowella
- Visio
- Omnigraffle
- Powerpoint
- ...

1. *„Start ugly"*
2. *Improve & Iterate*
3. *Say goodbye to "done"*

Iteration, Evaluierung & Testing
→So schnell wie möglich aufs Gerät bringen!

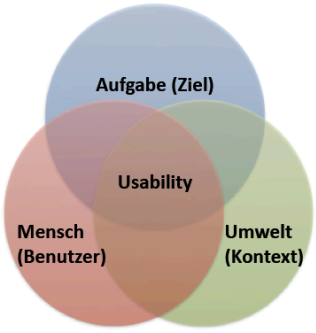
Why work visually?

- Visual ideas get through faster
- Pictures communicate a more complete idea
- Images help groups move faster together
- →Sodass das endgültige Design die Leute und ihre Bedürfnisse reflektiert.

5.Vorlesung: Design Tools & Usability Engineering

Usability Engineering

Usability = Eigenschaft, Ziel
Software Engineering = Prozess, Methodik



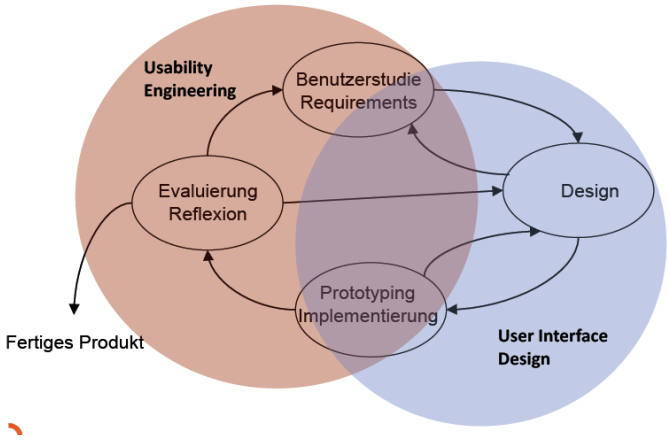
*„Usability eines Produktes ist das Ausmaß, in dem es von einem **bestimmten Benutzer** verwendet werden kann, um **bestimmte Ziele** in einem **bestimmten Kontext** effektiv, effizient und zufriedenstellend zu erreichen.“* (ISO 9241-11)

Usability + Engineering

Ansätze, um Usability in einen Software-Entwicklungsprozess zu integrieren:

1. Usability-Experten zu vorhandenen Teams und Prozessen hinzufügen
2. Usability-Methoden zum vorhandenen Entwicklungsprozess hinzufügen
3. Entwicklungs-Prozess neu gestalten

Prozess:



- *Analyse* – Kontext, Requirements, Konkurrenz, Usage Data, ...
- *Design* – User Interface Design, Interaction Design, User Experience Design, ...
- *Evaluierung* – User Testing, Reviews, Experimente, Feldstudien, ...

Methoden

Für jede Phase im Entwicklungsprozess:
Analyse von Users, Tasks, Kontext, User Interface Design, Prototyping, Evaluierung, Usability Tests

Je nach Zeit und Ressourcen?
Zugriff auf Benutzer?
KnowHow?

User-centered Design

Die Kluft überbrücken

- Zwischen den Wünschen der Benutzer und den Fähigkeiten des Computers.
- Zwischen Designer und Benutzer

Kluft zwischen Designer und Benutzer

- **Level 1:** Der Designer ist der Benutzer. Wenn man etwas entwirft, was man nur selbst verwenden wird.
- **Level 2:** Der Designer versteht das Produkt
  Aber Achtung: Design Team != Typische Benutzer
  →Designer...
    - wissen zu viel über das Produkt
    - sind zu geübt im Umgang mit Computern
    - hängen zu sehr an ihren Entwürfen
- **Level 3:** Design für eine fremde Domäne (z.B. Medizin)
    - Gut ausgebildete User
    - Spezialisierte Aufgaben
    - Experten-Wissen
    - Fremder Nutzungskontext

Du bist nicht der Benutzer ...und auch dein Chef nicht.
→Benutzerverhalten lässt sich nur schwer vorhersehen

Merkmale eines User Centered Design Prozesses:

1. Früher Fokus auf Benutzer und deren Aufgaben
    - Designer müssen die Benutzer verstehen:
    - Bei der Arbeit beobachten, was sind ihre Aufgaben
    - Benutzer in den Design-Prozess einbinden
2. Empirische Beobachtung
    - Reaktionen auf Szenarien, Storyboards beobachten
    - Später: Prototypen, User Testing
    - -> Evaluation!
3. Iteratives Design
    - Probleme tauchen überall im Design-Prozess auf
    - Müssen gelöst werden
    - Deshalb: Iteratives Design

Zusammengefasst

- Designer ≠ Benutzer
- User-centered design:
    - Die Benutzer einbinden
    - Empirische Evaluation mit den Benutzern
    - Iterativer Prozess

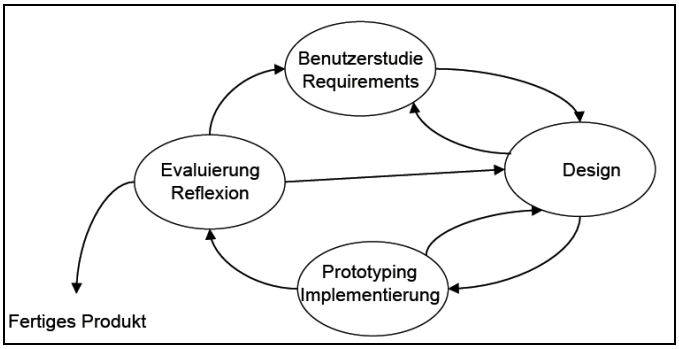
Prozesse

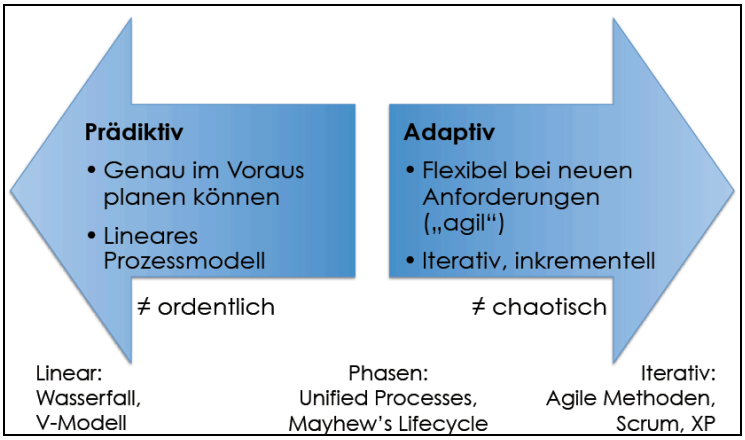
Wasserfallmodell:

1. Reqirements
2. Design
3. Code
4. Testen
5. Ausliefern

→ Das Problem mit linearer Vorgangsweise:
Es gibt „Reaktionen der Benutzer“ und „User Testing“, was Einfluss auf vorhergehende Phasen hat, bzw. man gezwungen ist diese nochmals zu durchlaufen.

Iterative Prozesse





Iterative Prozesse - Methoden in einem systematischen Framework:
Dabei wird der iterative Prozess organisiert/systematisch durchgeführt.

- Zuerst: Analysephase
- Dann: Designen, Testen, Entwickeln →mehrmals durchlaufen und zwar „von grob zu detailliert“
- Zum Schluss: Installieren

Methoden
Methoden für jede Phase im Prozess:

- Analyse
- Design & Prototyping

- Evaluierung

Cost/Benefit von Usability Engineering

- Weniger späte Designänderungen
- Weniger Trainingskosten
- Weniger Aufwand für Support
- Weniger Fehler der Benutzer
- Mehr Produktivität der Benutzer
- Positive Reviews und zufriedene Kunden

Aber: ROI ist schwer formalisierbar

- Return on Investment:
  - ROI bei Usability - Key Performace Indicators: +83% Verbesserung (Nielsen, 2008)
  - ROI, finanziell gemessen "success stories", schwierig abzuschätzen
  - $1 in frühzeitige Usability Tests = spart $10 in der Entwicklung (Nielsen, 1993)

Trotzdem oft hoher Aufwand →" Lieber billige und einfache Methoden verwenden!

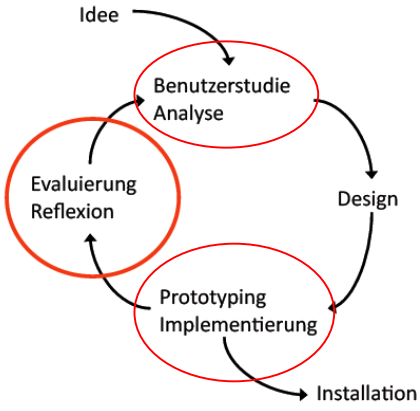
Discount Usability

„Das Beste ist der Feind des Guten"

- Scenarios
- Simplified thinking aloud
- Heuristic evaluation

Zusammengefasst

- Prozess für strukturiertes Vorgehen
- Iterativen Prozess wählen →„Fail early, fail often"
- Discount-Methoden für Zeit- und Kostensparen



Methoden

- **Analyse Methoden**

Wer ist "Der Benutzer"!?

"The elastic user must bend and stretch and adapt to the needs of the moment. However, our goal is to design software that will bend and stretch and adapt to the user's needs."

Ethnographischer Ansatz:

„I don't know how the people of Cushing, Wisconsin, understand their world. That remains to be discovered."

- Technik: Abstrakte Benutzer
- Marktanalyse: Wer kauft vielleicht?
- Design: Das beste System für konkrete Benutzer?

Benutzer haben…

- Wissen und Erfahrung
- Aufgaben
- Psychologische Eigenschaften
- Physische Eigenschaften

Was für Aufgaben?

- Technik: Funktionale Requirements
- Marketing: „Jobs to be done“ " Kaufentscheidung
- Design: Bestmögliche Unterstützung für die (vielen Unter-) Aufgaben

Arbeitsablauf aus Sicht der Benutzer →Vokabular, Abläufe, „Tacit knowledge“, Workarounds

Ziel: Work re-engineering, Optimierung, gute Software

Was für Kontext?

“When people research technology [they often start at] the point at which someone picks up the telephone or starts typing on the keyboard. For me that’s already far too down in the process. You want to know; Where does that PC live in someone’s home? [...] And even one step back further than that: What do people care about?”

- Art der Arbeit
  - Aufgaben und Ziele
  - Abläufe
  - Häufig vs. selten benötigt
- Physischer Arbeitsplatz
  - Hilfsmittel, Artefakte
  - Platzierung von Objekten
- Selbstbild der Benutzer
- Sprache der Benutzer
- Zusammenarbeit
  - Organisation in der Firma
  - Ziele des Unternehmens
  - Kulturelle Einflüsse

<u>Benutzer, Aufgaben, Kontext erforschen:</u>

Interviews

- Unstrukturiert:
  - Keine vordefinierten Pfade
  - Bringt oft neue Aspekte zum Vorschein
- Strukturiert
  - Vordefinierte Fragen oder Themen
  - Damit alles besprochen wird

In der Praxis: Mischformen meist sinnvoll

- Explorativ
  - Offene Struktur, erzählen lassen
  - Offene Fragen stellen
  - →Spontaneität, ungewollte Entdeckungen
- Fokussiert
  - Vorgegebene Struktur, Leitfaden
  - z.T. geschlossene Fragen
  - →Konversation bleibt im Interesse des Beobachters

In der Praxis: Erst explorativ, dann fokussieren

Fragen können bei Interviews sein:

- offen / geschlossen
- immanent / exmanent
- direkt / indirekt / relational / rekursiv
- fakten-bezogen / sozial / zeitlich

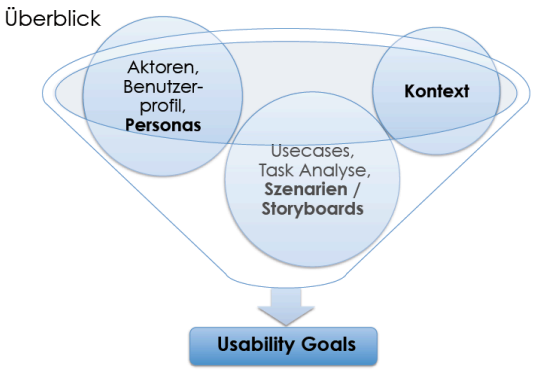
Contextual Inquiry

- Beobachten von realen Benutzern →direkt im Kontext der Arbeit
- Tacit knowledge (Wissen das schwer an Andere zu transferieren ist wie z.B.: Fähigkeit eine neue Sprache zu sprechen, Algebra verwenden können, etc. im Gegensatz dazu steht explizites Wissen das leicht an Andere zu transferieren ist wie z.B.: Wien ist die Hauptstadt von Österreich, etc.)
- Wissen über Arbeitsabläufe und Prozesse sammeln
- Wissen über die Problemdomäne, (Arbeits-) Kultur, physische und soziale Grenzen sammeln
- Vertrauen zu Benutzern aufbauen, die eventuell später das Produkt evaluieren

Weitere Methoden um Benutzer, Aufgaben, Kontext zu erforschen:

- Umfragen, Fragebögen
- Cultural probes
- Stakeholder Meeting
- Fokusgruppen
- Partizipatorisches Design

Benutzer, Aufgaben & Kontext darstellen



- Aktoren: Technischer Ansatz, UML, Objektorientierte Software-Entwicklung
- Zielgruppen, Benutzerprofile: Statistisch-quantitativer Ansatz
- Personas: Zusätzlich noch mit Kontext, Name und Gesicht.

- Usecases, Anwendungfälle: Technischer Ansatz, UML
- Szenarien: Narrativer Ansatz. Berücksichtigt auch den Nutzungskontext.
- Storyboards
    - Geeignet zur gemeinsamen Beschreibung von
        - Benutzer: Personas
        - Tasks: Szenarien
        - Kontext
    - Storyboards sind gut zur Präsentation geeignet
    - Umsetzung auch als Video möglich (Vergleich Übungsbeispiel 3)

Personas
Nicht: Demographische Gruppen: „18-34 jährige verheiratete Akademiker“
Nicht: Job-Beschreibungen: „IT-Manager in Fortune 500 Unternehmen die Kaufentscheidungen treffen können“
Personas sind:

- Archetypische Benutzerbeschreibungen. Stellvertretend für die realen Benutzer (statt der anonymen und pauschalen Größe „Der Benutzer“)
- Anforderungen, Wünsche, Bedürfnisse, Charakter, Denkweise, Meinungen
- Statistisch repräsentativ für Zielgruppe
- Nützlich für die Kommunikation im Team
- Personas sind Stereotype
- Personas sind Designziele, keine Verkaufsziele

Discount-Methoden
In der Analyse-Phase

- Benutzer und Tasks beobachten
- Szenarien formulieren und diskutieren

Analyse-Phase zusammengefasst:

- Benutzer
- Aufgaben
- Kontext

- Interviews machen
- Beobachten gehen
- Personas, Szenarien, Storyboards

- **Design, Prototyping Methoden**

Was ist ein Prototyp?
“A prototype allows stakeholders to interact with an envisioned product, to gain some experience of using it in a realistic setting, and to explore imagined uses.” (Preece et al. 2003)

"Prototypes are representations of a design made before final artifacts exist." (Buchenau et al. 2000)

- Repräsentation eines Systems
- Wird in vielen Gebieten des Designs verwendet
- Unterschiedliche Arten der „fidelity"

Was ist ein Prototyp konkret?

- Software
- Physisches Modell
- Storyboard
- Video
- Slideshow
- Papier-Mockup

→ Prototypen kommunizieren eine Idee

Was kann (soll) man prototypen?

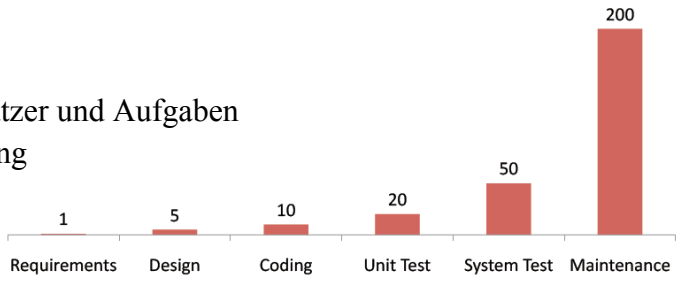
- Technische Aspekte
- Layout am Bildschirm
- Wo werden welche Informationen angezeigt?
- Graphisches Design und „Look and Feel"
- Arbeitsablauf und einzelne Aufgaben
- Inhalte, Bezeichnungen, Abkürzungen
- Kontroversielle und kritische Aspekte

→ Man kann alles prototypen!

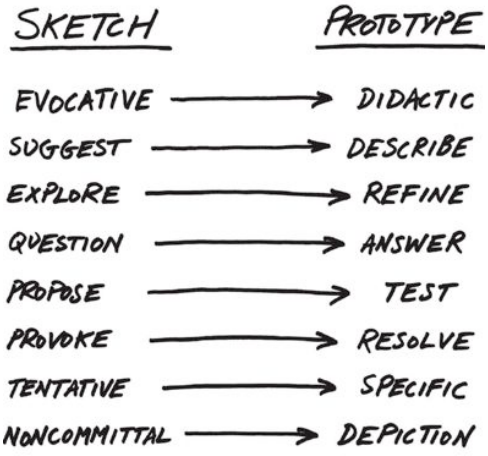
Warum Prototypen?

"Generally the last thing that you should do when beginning to design an interactive system is write code." Bill Buxton

- Ideen finden: Prototypen als Denkwerkzeuge
- Ideen ausprobieren
- Im Team arbeiten
- Zwischen Alternativen wählen
- Konzepte kommunizieren
- Evaluierung und Feedback (Rechtzeitig!)
- Angreifen und Interagieren
- Reflexion und Problemfindung
- Frühe Usability Tests
- User-Centered Design:
  - Früher Fokus auf Benutzer und Aufgaben
  - Empirische Beobachtung
  - Iteratives Design
- Späte Korrekturen sind teuer:

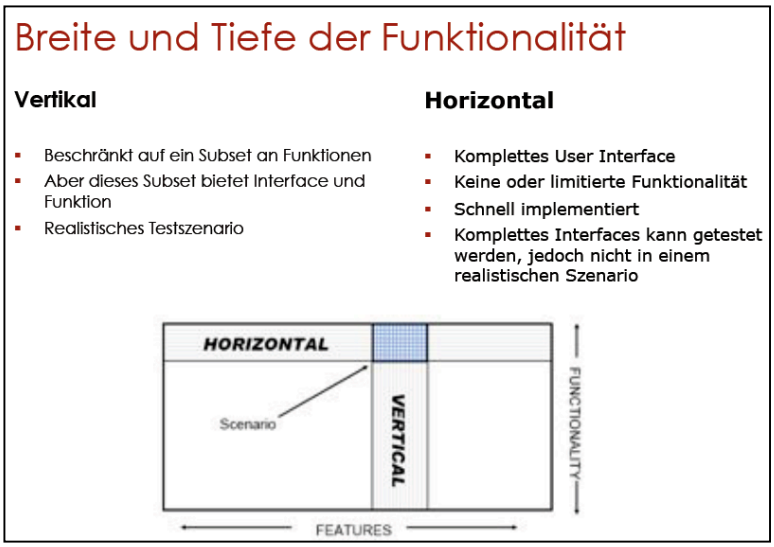


Sketching vs. Prototyping



Fidelity: In Bezug worauf?

1. Breite
2. Tiefe
3. Aussehen: Marke und Identität, Gefühl, Graphikdesign, Schriften und Farben, Styleguide...
4. Interaktion (Achtung: Interaktion≠Tiefe)



Absichtliche Ungenauigkeit

Prototyping Tools: Unterschiedlich detaillierte Formen der Darstellung zum Management von Erwartungen. z.B. kann durch „comicartige“ Mockups der Eindruck vermieden werden, dass das System schon fertig ist.

Discount approach

→Rapid prototyping

- Schnelle und billige Prototypen
- Viele Ideen testen, verbessern, verwerfen
- Oft reichen sehr einfach, billige Papier-Prototypen

Papier Prototypen

- Wichtigste Art von low-fi Prototypen
- Von jedem machbar, „quick and dirty“, Iterationen leicht möglich
- Papier, Klebstoff, Marker, Overhead Folien, Schere, Post-Its (bunt)
- Simulation von dynamischen Interface Elementen

High-fidelity Prototypes

Z.B. MS Office 2007 Ribbon Design

- Über 25.000 Bilder, Mockups
- Hunderte klick-through Prototypen

Protoyping zusammengefasst:

- Low-fi
  - Früh im Design-Prozess
  - Schnell und einfach, damit leichter zu verwerfen
- High-fi
  - Spät im Design-Prozess
  - Detaillierter

- **Evaluierungsmethoden**

Usability ist messbar

- Erlernbarkeit
- Effizienz (!= Effektivität)
- Erinnerbarkeit
- Fehler
- Zufriedenheit
- User-abhängig!

Inspektionen vs. Testing

- Usability Inspektionen: Experten
  - Experten versetzen sich in die Rolle des Benutzers
  - Fehler vorhersagen
  - Beurteilung anhand von Erfahrungen und Richtlinien
  - z.B. Heuristic Evaluation, Guidelines, Cognitive Walkthrough
- Usability Testing: Testpersonen, Endbenutzer
  - Fehler erfassen, die in der Praxis auftreten
  - Repräsentative Benutzer bearbeiten vordefinierte Aufgaben
  - Beobachtung durch Usability Experten
  - Analyse der Ergebnisse aus der Beobachtung zur Verbesserung

Kriterien von Usability Evaluierungen

- Basierend auf Benutzer
  - Usability Test
- Basierend auf Heuristiken und Guidelines
  - Heuristische Evaluierung
  - Cognitive Walkthrough
- Basierend auf kognitiven Modellen
  - Human Information Processor – Modell
    - GOMS (Goals, Operators, Methods, Selection)
    - KLM (Keystroke Level Model, time needed to write text)
    - Fitts law (time needed to hit a button)
  - Psychologische Experimente

- Basierend auf Marktforschung
    - Competitive Analysis

Ziele von Usability Evaluierung

- Exploratives Testen: Sucht qualitative Ergebnisse, eher früh im Prozess, hilft bei Design-Entscheidungen
- Assessment-Testen: Der Benutzer soll eine Aufgabe bewältigen
- Vergleichs-Testen: Welcher Entwurf ist besser?
- Validierungs-Tests: z.B. für die Abnahme eines Produkts durch den Kunden

Usability Test vorbereiten

- Fragestellung, Ziel des Tests festlegen →Welches Leistungsmaß?
- Testpersonen rekrutieren
- Szenarien und Aufgabenstellungen vorbereiten
- Unterlagen und Software / Prototypen vorbereiten
- Durchführung

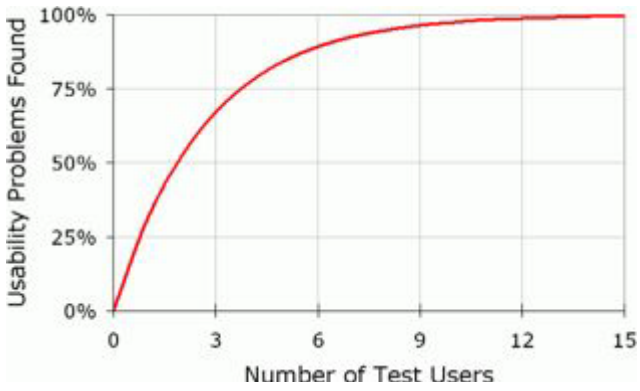
Discount-Approach

Methoden für Discount Usability Evaluation:

- Thinking Aloud (Vorausdenken)
- Heuristische Evaluation

<= 5 Personen aus jeder Zielgruppe

- Drei Testpersonen finden ca. 50% der Fehler
- Vier bis fünf Personen finden 80%
- Zehn Personen finden 90%



Simplified Thinking Aloud

- Informelle Form des User Testing
- Potentielle Benutzer bedienen das Produkt und geben ihre Gedanken laut weiter.
- Entwickler sind anwesend
- Keine Videokamera
- Keine Experten
- Daten werden vor Ort verarbeitet
- Einsatz während der Entwicklung des Produkts

Hallway Usability Testing

„A hallway usability test is where you grab the next person that passes by in the hallway and force them to try to use the code you just wrote.“

Usability Evaluierung zusammengefasst:

- Tests vorbereiten
    - Fragestellung
    - Testpersonen
    - Unterlagen
- Lieber einfach als gar nicht!
    - 5 Benutzer reichen
    - Thinking-aloud, Hallway

Usability Engineering zusammengefasst:

Benutzer-zentriert

- Analyse: Benutzer, Tasks, Kontext
- Design: Sketching, Prototyping
- Evaluierung: Usability Testing &co

→ Iterativ!

6.Vorlesung: Trends & Zukunftsperspektiven

Ubiquitous Computing
„Machines that fit the human environment instead of forcing humans to enter theirs.“
Ubiquitous: (All)Gegenwärtig
Webster: *Existing or being everywhere at the same time*

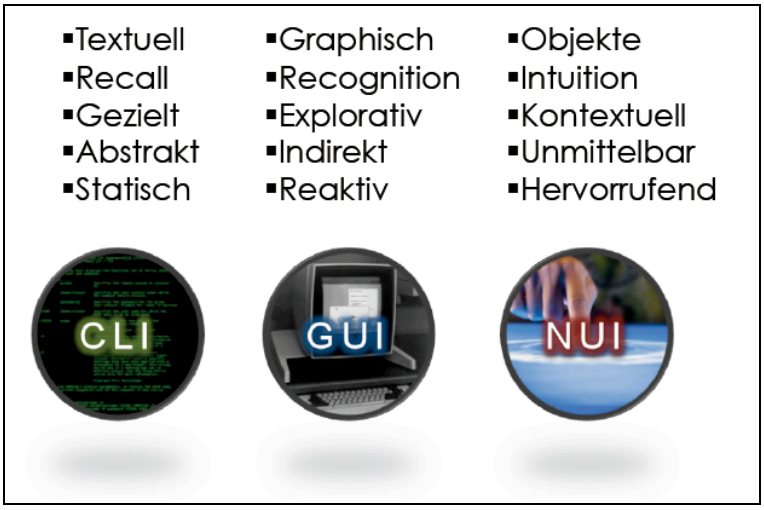
Oft auch: Pervasive Computing, Internet of Things, Everyware, Calm Computing, Continous Computing, Physical Computing, Ambient Intelligence, ...

- Computer im Hintergrund eingebettet
- Integration von Computern in Alltagsgegenständen
- Unaufdringliche User Interfaces
- Nicht-monopolisierend (d.h. nicht von der Umgebung abschottend)
- Periphere Wahrnehmung und selektive Fokussierung
- Fließender Wechsel von einem System zum nächsten

Tabs, Pads und Boards
Mark Weiser beschreibt drei Komponenten für Ubiquitous Computing:

- Tabs: Kleinste Einheit, so groß wie ein Post-It
- Pads: Tablets so groß wie ein Blatt Papier
- Boards: Großformatige Displays

Natural User Interfaces Evolution



Touchscreens & Multitouch Interaction

- *Microsoft Surface:* Gerät, auf dem Mehrere gleichzeitig arbeiten können, sieht aus wie ein Tisch
- *Trackpads:* z.B. bei Laptops: mit 1 Finger Maus bewegen, mit 2 Fingern scrollen,…
- *Microsoft Sphere:* Touchscreen ist eine Kugel
- *Multitoe:* Steuerung mit Füßen, Boden ist Touchscreen

- *Sixth Sense:* Mit Mini-Projektor Bildschirm auf beliebige Oberflächen projezieren, auf Fingern sind Marker angebracht, die vom System erkannt werden
- *OmniTouch:* Mit Mini-Projektor Bildschirm auf beliebige Oberflächen projezieren, funktioniert wie Touchscreen

Touchscreens

- Direkte Interaktion
- Man kann Dinge, mit denen man interagiert, direkt anfassen

Touchpads

- Indirekte Interaktion (vgl. Maus)
- Man kontrolliert einen Zeiger am Bildschirm

Finger oder Stylus

- Aktueller Trend geht weg vom Stylus in Richtung Bedienung mit dem Finger
- Stylus bietet größere Präzision, stellt jedoch ein Problem für Multitouch-Bedienung dar

Multi-Touch Technologien

- Kapazitive Sensorik
- Optische Sensorik
- Resistive Sensorik
- Akkustische Sensorik
- Etc.

Optische Sensorik

- Tracking von Fingern mittels einer Kamera und Blob-Detection Software
- Optisches Tracking in Kombination mit Projektion mittels Infrarotlicht

Technologien:

- Diffused Illumination (DI)
- Frustrated Total Internal Reflection (FTIR)
- Laser Light Pane Illumination (LLP)
- Diffused Surface Illumination (DSI)

Motivation von Multi-Touch

- Kollaboratives Arbeiten (Co-located)
- Keine zusätzlichen Eingabegeräte (Maus, Stylus, ...)
- Direkte Interaktion
- Intuitive Bedienbarkeit
- Ausdrucksstarke Gesten
- Effizienz?

→ Wo soll Multi-Touch eingesetzt werden?
→ Welche Herausforderungen gibt es dabei?

Herausforderungen

Screen-Based

- Affordance of Screens: z.B. "Bitte Bildschirm berühren", Hand die auf Bildschirm abgebildet ist soll eine Aufforderung zum Berühren darstellen,…
- Taktiles User Feedback: Bildschirm ist glatt, d.h. man hat von vornherein kein taktiles Feedback →Vibrieren, Leuchten, akustische Signale

User-Based

- Ergonomie: Buttons/Tasten in der richtigen Größe machen, Touchscreen nicht in 3m Höhe anbringen, etc.
- Individuelle Unterschiede: Jeder Mensch ist anders →unterschiedliche Fingergrößen, etc.
- Accessibility: Ist das Produkt für blinde Personen (oder andere Menschen mit Behinderung) zugänglich/verwendbar?

Input-Based

- Gestures and Patterns: Gesten sollten einheitlich gestaltet sein, bzw. individuell einstellbar sein
- Dateneingabe unterstützen: Vorauswahl anbieten,…
- Multi-User Support: Ist es möglich, dass mehrere Benutzer gleichzeitig am Gerät arbeiten?

Motion Control

Wii, Kinect, PS Move, Kinect Hacking, g-speak,…
→mit Körperbewegungen den Vorgang am Gerät steuern

Tangible Computing

- Physische Artefakte sind direkt mit digitaler Information verbunden.
- Physische Artefakte enthalten und realisieren Mechanismen zur interaktiven Kontrolle.
- Physische Repräsentationen sind wahrnehmbar verbunden mit ihrer digitalen Repräsentation.
- Der Zustand des physischen Objekts verkörpert Schlüsselzustände des digitalen Systems.
- z.B. wird Gerät gesteuert, indem Dinge auf der Screenoberfläche verschoben werden.

„The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it. Such a disappearance is a fundamental consequence not of technology, but of human psychology. Whenever people learn something sufficiently well, they cease to be aware of it. [...] only when things disappear in this way are we freed to use them without thinking and so to focus beyond them on new goals."