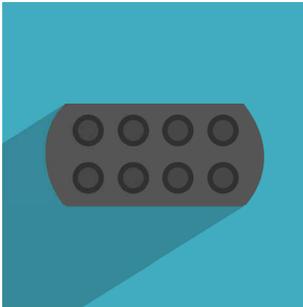


Usability

Konzepte
Regeln
Hinweise



Acht Goldene Regeln

1. Strebe nach Konsistenz
2. Erlaube Abkürzungen
3. Gib informatives Feedback
4. Entwirf abgeschlossene Dialoge
5. Erlaube einfache Fehlerbehandlung
6. Ermögliche Rückschritte („Undo“)
7. Gib dem Benutzer das Gefühl der Kontrolle
8. Reduziere Belastung des Kurzzeitgedächtnisses

Ben Shneiderman (1992)

Prinzipien für gutes Design

Systemstatus und jeweils verfügbare Aktionen sind immer sichtbar

System bietet ein gutes konzeptionelles Modell und ein konsistentes Erscheinungsbild

Interface weist gute Abbildungen auf, welche die Beziehungen zwischen Stadien darstellen (z.B. „fliegende Icons“ in Windows, vergrößernde/verkleinernde Rechtecke beim Öffnen/Schließen)

Anwender erhält kontinuierliches Feedback

abgeleitet aus 7-Stages-Model von Norman (1988)

Ordnung auf dem Bildschirm

Konsistenz der Datendarstellung

Effiziente Informationsaufnahme

Minimale Gedächtnisbelastung

Kompatibilität Datendarstellung – Datenwert

Flexible Anwenderkontrolle bzgl. Datendarstellung

Sidney L. Smith & Jane N. Mosier (1986)

„Das Weibchen sucht sich nicht das Männchen,
das ihm am besten gefällt, sondern das,
das es am wenigsten abstoßend findet.“

Charles Darwin (1809 – 1882)

Anforderungsaufnahme

Die Anwender wollen: schneller, besser, billiger.

Was wollen oder benötigen sie tatsächlich?

Lerne die benötigten Aktionen kennen.

Definiere Ist- und Soll-Zustand & umgebende Prozesse.

Identifiziere häufige und gelegentliche Aufgaben.

Funktionalität muss Anwenderbedürfnissen entsprechen.

Basisanforderungen

Auswirkungen von Aktionen entsprechen spezifiziertem Verhalten.

Dargestellte Daten entsprechen tatsächlichen Daten.

Das System ist verfügbar, fällt nicht aus.

Das System führt keine neuen Fehler ein.

Alle Daten sind geschützt (vor Missbrauch, unautorisiertem Zugriff, unbeabsichtigtem Löschen).

Anwender begegnen jedem neuen System mit Misstrauen.

Checkliste für Aufwände

Nutzermenge & -spektrum

Nutzungsfrequenz & -intensität

Motivation der Nutzer

Erlernbarkeit und Schulungsaufwand

Fehler-Risiken & Sicherheit

Subjektive Zufriedenheit & Vertrauen

Effektivität, Effizienz, Performance

Preis & Ressourcen

„Diejenigen, die immer nur das Mögliche fordern, erreichen gar nichts.
Diejenigen, die aber das Unmögliche fordern, erreichen wenigstens das Mögliche.“

Michail Bakunin (1814 – 1876)

User Storys oder Use-Cases

Personas

- mind. 3 Steckbriefe von potenziellen oder realen Nutzern
- Name, Alter, persönliche Situation
- Aufgaben, Ziele
- Ausbildung, Interessen, Kenntnisse
- „Pain“ und „Gain“
- Zitate (aus Interviews)

Geschichte erzählen

- Hauptdarsteller
- Konflikt
- Handlung
- Auflösung

Happy End:

- Der Anwender hat SEIN Ziel erreicht.
- Es war wenig Aufwand nötig.
- Jede Entscheidung fühlte sich zielführend an.
- Es gab keine unnötigen Umwege.
- Alle Fragen o. Vorschläge passten zum Ziel.

Kontexte erschaffen über Persona, die etwas in einer bestimmten Situation verwendet

Nicht die Software bestimmt den Weg und die Möglichkeiten, sondern diese leiten sich aus den Zielen des Anwenders ab.

Nutzertypen

Novize ./ Experte

Gelegenheitsnutzer ./

Power-User

Jäger ./ Sammler

Zielgerichteter ./ Stöberer

Abenteurer ./ Schutzbedürftiger

Meinungsbesitzer ./ Schwamm

Interakteur ./ Konsument

Genießer ./ Pragmatiker

4 Typen nach Myers-Briggs „Type Indicator“:

- extrovertiert ./ introvertiert
- empfindend ./ intuitiv
- perzeptiv ./ bewertend
- fühlend ./ denkend

4 Arbeitstypen:

- explorativ
- kreativ
- kollaborativ
- sozio-technisch

Nutzervorgehen

1. Bedürfnis => Ziel (Ergebniserwartung)
2. Idee => Plan/Konzept (mentales Modell & Weg)
3. Aktion/en
 - ggf. Koordination, Organisation
 - Intention (Was)
 - Operation (Wie)
4. Ergebnis => Prüfung

Menschliche kognitive Prozesse

Langzeitgedächtnis & semantisches Gedächtnis

Kurzzeit- & Arbeitsgedächtnis (7±2 Daten im Kurzzeitgedächtnis)

Problemlösen & logisches Denken

Entscheiden & Risikobewertung

Sprachliche Kommunikation & Verstehen

Such-, bildliches & sensorisches Gedächtnis

Lernen, Skill Development, Wissenserwerb & Erwerb von Konzepten

Persönliche Faktoren

Alter, Geschlecht, physische & kognitive Verfassung

Bildung, Kenntnisse, Wissen, Trainingsstand

Glaube, Überzeugungen, Furcht, Sorgen

Motivation, Streben, Ziele

kultureller, ethnischer Hintergrund

Woher & Wohin

Training/Gewöhnung macht für bestimmte Dinge blind oder sensibel

Situative Einflussfaktoren

Emotionen, Befindlichkeit, Laune

Erschöpfung, Müdigkeit, Stress

Drogen, Medikamente, Unwohlsein

„Die meiste Zeit verbringen User auf anderen Webseiten.“

Jakob Nielsen

Entwickler sind keine „Nutzer“!

Fünf Formen der Interaktion

Direkte Manipulation: GUI, (Reale-Welt-)Metaphern, bildliche Repräsentation, sofortiges Feedback, schnell aktivierbare, reversible, inkrementelle Aktionen; ermuntert zur Exploration, einfache Erlernbarkeit, gute Beibehaltung der Benutzbarkeit (räumliches, motorisches Gedächtnis), Entwicklungsaufwand (zur Aufwandsreduktion Standards des Systems nutzen); Bsp.: iPad (Zoom- und Wischgeste), Icon in Papierkorb legen

Menüauswahl: Auswahl aus gleichwertigen/gleichartigen Optionen; reduzierter Lernaufwand, strukturierte Entscheidungsprozesse ./ Flexibilität, Error-Handling, Gefahr „Menü-Dschungel“ (z.B. Länderauswahl), wenig effektiv für erfahrene Anwender; Bsp.: DOS-Menü, Programm-Menü, Ribbon

Formularsprache, -eingabe: einfache Dateneingabe, wenig Einarbeitung, Unterstützung des Anwenders durch Strukturierung der Eingabe, unübersichtlich und kompliziert bei komplexen Eingaben

Kommandosprache: Flexibel, geeignet für „Power User“, mächtig, Skripting, schlechtes Fehlerhandling, hoher Trainingsbedarf, Gedächtnisleistung!

Natürliche Sprache: Geringer Lernaufwand, klärender, einführender Dialog nötig, Kontext schwer darzustellen, bei Texteingabe viele Tastatureingaben nötig, Dialoge nicht vorhersehbar, technischer Aufwand

Fünf Prinzipien jeder Interaktion

Respekt: Von mir werden nur Angaben erwartet, die für die Aufgabe nötig sind und die ich zu leisten in der Lage bin. Dafür wird mir der geringstmögliche Aufwand abverlangt.

Vertrauen: Meine Daten werden nicht missbraucht, sondern ich erhalte am Ende das versprochene Ergebnis. Gleiche Daten ergeben das gleiche Ergebnis, das Sprechen über die Vorgänge bzw. deren Darstellung entspricht dem tatsächlichen Umgang mit den Daten, und gleichartige Darstellung bedeutet gleichartige Funktionalität.

Unterstützung: Ich erhalte Hilfe dabei, eine bestimmte Aufgabe zu lösen oder ein Anliegen zu erledigen. Nicht mehr, aber auch nicht weniger. Ich werde weder gegängelt noch bevormundet oder zu Aktionen gezwungen, die nicht meiner Aufgabe oder meinem Anliegen entsprechen.

Verständnis: Mein Helfer kennt mein Anliegen und den Weg, wie ich es erledige, bzw. den Weg zur Lösung meiner Aufgabe. Gibt es für meine Anliegen oder Aufgaben keine vorgefertigten Wege, so werden mir verschiedene Möglichkeiten so bereitgestellt, dass ich mir selber einen Weg erarbeiten kann.

Kommunikation: Ich erhalte Feedback, wenn etwas nicht wie beabsichtigt läuft oder über den erfolgreichen Abschluss einer Aktion. Ich werde in einem angemessenen Tonfall und geeigneter Darstellung geführt. Direkte und indirekte Kommunikation widersprechen sich dabei nicht.

Stärken

Menschen	Computer
Schwache Stimuli aufnehmen	Signale außerhalb menschl. Wahrnehmung aufnehmen
Stimuli vor verrauschtem Hintergrund erkennen	Physikalische Größen zählen & messen
Muster in variierenden Situationen erkennen	Kodierte Informationen korrekt speichern
Unübliche, unerwartete Ereignisse aufnehmen	Vordefinierte Ereignisse überwachen
Prinzipien und Strategien erinnern	Schnelle, konsistente Antworten auf Eingangssignale
Wichtige Details erkennen, auch ohne Wissen über Kontext	Quantitative Größen & detaillierte Informationen akkurat abrufen

Abstraktionsfähigkeit, Muster erkennen, Intuition, Entscheidungen mit wenig Datenmaterial bzw. Evidenz

selbe Eingabe + Algorithmus = immer die selbe Ausgabe; nicht ablenkbar

„Für das Verständnis des Computers [...] ist entscheidend, dass die Intelligenz beim Übergang in die Maschine ihre Einbettung in andere menschliche Fähigkeiten verliert. Aus der wahrnehmenden Intelligenz wird eine **blinde** Intelligenz, aus der fühlenden Intelligenz wird eine **kalte** Intelligenz, aus der sinngeliteten Intelligenz wird eine **dumme** Intelligenz, und aus der verantwortenden Intelligenz wird eine **gleichgültige** Intelligenz.“

*Heinz Buddemeier
„Von der Keilschrift zum Cyberspace“ (2001)*

Schwächen

Menschen	Computer
Unaufmerksamkeit durch Monotonie oder Ablenkung	„Tertium non datur“, benötigt eindeutige Eingaben
Langsame Verarbeitung von Datenmengen	Abbruch bei unerwarteten Eingaben
„Schusselfehler“, Fehlentscheidungen	Ergebnis nur bei vollständiger Eingabe
Niedrige, unzuverlässige Gedächtnisleistung	Begrenzte Ressourcen (Speicher, Anzeige, Eingabe)
Sensorische Kompetenz nur „gut genug“ zum Überleben	Blind/taub gegenüber „Zwischentönen“
Entscheidungswege situationsabhängig, nicht wiederholbar	Gleichgültig, neutral

Irrationalität, Unzuverlässigkeit, (falsche) Sinn- und Bedeutungszuschreibungen

stupide Regelbefolgung; keine Rücksicht auf Kontext und Wirkung

Computer können nicht nach humanen Maßstäben bewertet werden – sie sind nur Werkzeuge!

Die Verantwortung liegt daher wieder bei Menschen, die dem Computer beibringen, sich auf andere Menschen einzulassen: in der zeitlichen, räumlichen, kulturellen Distanz.

Interface kaschiert

- Blindheit
- Kälte
- Dummheit
- Gleichgültigkeit

„Obgleich Computer im Prinzip alles können, was wir können, ist die tatsächliche Maschinenintelligenz heute und in vorhersehbarer Zukunft von der unsrigen weit entfernt. Dies wird deshalb oft übersehen, weil wir so schlecht in dem sind, was Programme gut können. [...] Was jedoch meistens vergessen wird, ist der Umstand, dass Computer nahezu vollständig an Dingen scheitern, die wir alle gut beherrschen. Schließlich besteht Intelligenz zum großen Teil darin, vernünftige Entscheidungen ohne vollständige Evidenz zu treffen.“ *Margaret A. Boden, „Künstliche Intelligenz und Menschenbilder“ (1985)*

Number-Crunching ./. Muster, Anomalien entdecken

Geduld ./. Ungeduld

Da-Sein ./. Bewusstsein

Algorithmus ./. Zielstrebigkeit ./. geistige Flexibilität

0/1-Rationalität ./. Emotionalität

„Der meiste Schaden, den der Computer potenziell zur Folge haben könnte, hängt weniger davon ab, was der Computer tatsächlich kann oder nicht kann, als vielmehr von den Eigenschaften, die das Publikum dem Computer zuschreibt.“
Joseph Weizenbaum (1923 – 2008)

Studie der TU-Berlin (2014)

„Gefühle entscheiden über Akzeptanz, Nutzung und die Bewertung technischer Produkte.“ *Nils Backhaus, Psychologe*
Nutzer würdigen nicht, dass Technik funktioniert, sondern setzen dies einfach voraus.

Die Benutzung eines technischen Gerätes muss positiv erlebt werden.

„Positive Emotionen – das hat die Forschung nachweisen können – verbessern die Informationsaufnahme.“
Stefan Brandenburg, Studienleiter

Uncanny Valley

Akzeptanz von künstlichen Figuren bzw. als anthropomorph empfundenen fällt mit steigender Simulationsqualität; erst ab einem sehr hohen Level steigt auch die Akzeptanz wieder.

=> Disneysierung/
Vermenschlichung der Maschine vermeiden

=> Avatare sind selten eine gute Idee

Anthropomorphismus

„Disneysierung“, Vermenschlichung von Dingen, Tieren, Maschinen. Zuschreibung von menschlichen Eigenschaften

=> Erwartungshaltung, Emotionen, Unsachlichkeit, Enttäuschung.

=> keine „Ich“-Aussagen

Skeuomorphismus

Nachahmung der Form, Gestalt eines konkreten Dings in einem anderen Medium, ohne dass funktionale Gleichheit herrscht.

=> das Wesen, die Funktion erfassen, nicht die Erscheinung

Farbe

Farbe statt Buntheit
 Farbe als (zusätzlicher)
 Bedeutungsträger
 gutes Design funktioniert
 in Schwarz/Weiß
 sparsam einsetzen!

Aufmerksamkeit lenken –
 nicht ablenken!

Elemente gruppieren,
 markieren

Visualisierung von Daten
 (Wetterkarten, Wärmebil-
 der, Hirnscans, „Equalizier“)

Fehlfarbsichtigkeiten
 bedenken

Schriften

Serifen: Cambria,
 Constantia, Garamond,
 Georgia, Times New Roman

Serifenlos: Arial, Calibri,
 Candara, Corbel, Helveti-
 ca, Tahoma, Trebuchet,
 Verdana

Monospace: Consolas,
 Courier New, Lucida
 Console, Monaco

Sonstige: Comic Sans,
 Strutura, Zapfino,
 Webdings, Wingdings

Achtung: 12 pt ≠ 12 px

*Lesetempo bei serifem &
 serifenlosem Fließtext gleich
 => beeinflussen „nur“ Äs-
 thetik, Erwartungshaltung*

*Aber: Schriftgröße, Zeilenab-
 stände, Zeilenlänge beein-
 flussen Lesetempo stark*

Wir nutzen

Ausrichtung um Ordnung zu schaffen

Dominanz um Wichtiges zu markieren

Hierarchien für Wichtigkeitsabstufung

Weißraum für Übersicht

Visuelle Balance für Ausgewogenheit

Basis, Ruhe

Orientierung

Struktur

Kontrast

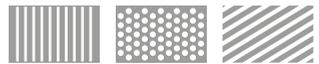
Harmonie

Visuelle Hilfsmittel

- Position



- Textur



- Fläche, Größe



- Neigung, Orientierung



- Helligkeit



- Form, Gestalt



- Farbe



außerdem: Länge, Farbton, Winkel, Enthaltung, Volumen,
 Sättigung, Verbindung, Bewegung, Blinken

Optische Achsen: Satzketten, Bildschirmränder; so we-
 nig wie möglich; wenige Winkel

Wirkung entsteht u.a. durch Verhältnis zu anderen Ele-
 menten, zum Hintergrund, zur Umgebung/zum opti-
 schen Kontext, zur Erwartung, zur „Standard“-Nutzung
 => **Kontrast** als Meta-Hilfsmittel

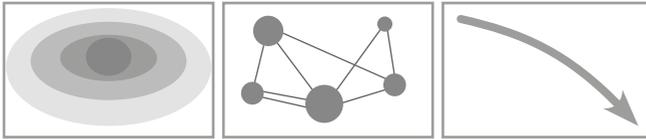
*„Je weniger einer braucht, desto mehr nähert
 er sich den Göttern, die gar nichts brauchen.“
 Sokrates (469 v.u.Z. – 399 v.u.Z.)*

Der wandernde Blick

- Vom Zentrum zum Rand
- Vom „lautesten“ Element umher
- Von links oben nach rechts unten

Foveale (im Fokus) & periphere („unscharfe“) Wahrnehmung

Optische Mitte: etwas oberhalb d. geometrischen Mitte



Hauptfokus

„per default“ zentraler Bereich über Mitte fokussiert
Ablenkung durch „laute“ Elemente, Gewohnheit, situative Erwartung



Hauptfokus abhängig von Position vor Monitor (üblich: Bildschirmoberkante in Augenhöhe), variiert bei Monitorgrößen & Gerätetyp (Laptop, Monitor)

Tablets & Smartphones haben keinen klaren Hauptfokus, da der gesamte Bildschirm in den Fokus passt

Ampel (Farbe mit funktionaler Aussage kombinieren)

schlecht:						Abbrechen	Nein
Hinweis:						Achtung	Prüfen
gut:						OK	Ja

schlecht:						Abbrechen	Nein
Hinweis:						Achtung	Prüfen
gut:						OK	Ja

Farbfläche allein ist keine Ampel. Erst Kontext sowie Inhalt (Text, Form) ergeben die entsprechende Aussage. Ampeln nur für kritische Status einsetzen; wenn mehrere auf einmal auf Bildschirm, dann nur gleichartige.

Visuelle Prinzipien

... der Nähe: Nahes gehört zusammen

... der Ähnlichkeit: Ähnliches gehört zusammen (Farbe wirkt stärker als Form)

... der stetigen Fortsetzung: Kontinuität, es werden einfache, regelmäßige, glatte Formen interpretiert, keine abrupten Änderungen

... der Geschlossenheit (Konvexität):
Nein, hier ist kein Dreieck abgebildet:



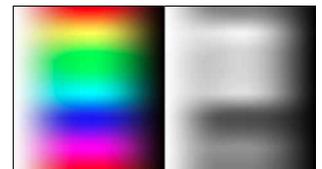
... der Prägnanz, des gemeinsamen Schicksals etc.
Übrigens: Menschen interpretieren immer, stellen Sinnzuschreibungen oder Bedeutung her.

Farbwirkung

Farben wirken unterschiedlich intensiv

Rot und Blau sind „lauter“ als Grün oder Gelb

Kontraste beachten (zueinander & zum Hintergrund): mind. 50% Differenz



Grundlagen

analog Satzspiegel (Print)

Raster: vertikal o. horizontal

Gitter = vertikales + horizontales Raster kombiniert

Raster gibt Basis-Einteilung vor

- feste Hilfslinien; entstehende Flächen zu Bereichen kombinierbar
- fixe Vorgabe für Bereiche (absolut, relativ oder usergesteuert)

übliche Rastereinteilungen

- einheitliche Breite
- 12er-Raster (für 2, 3, 4, 6 Spalten)
- „Goldener Schnitt“
- z.B. 1:2, 1:3, 2:3, 2:5

wichtiges Werkzeug des Informationsmanagement

führt Nutzer: Orientierung, Abhängigkeiten von Funktionen & Inhalten

Platzierung (in Bereichen) gibt Elementen Bedeutung & strukturellen Gehalt

Zugrundeliegende Modelle, Metaphern, Strukturen, Logiken werden sichtbar

Das 960-Grid

960 px breit

12 Spalten á 60 px

11 Zwischenraum á 20 px

Abstand außen: 2 x 10 px

im Web häufig zu finden, auch in Abwandlungen

Aufteilung nutzen

plausibler Anlass, Grund für Aufteilung (z.B. Hierarchie, Reihenfolge)

Markierung als eigene Bereiche (z.B. durch Hintergrund, eigene Gestaltung)

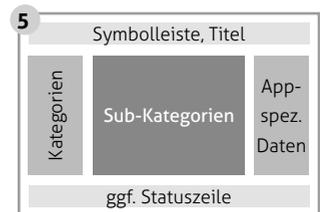
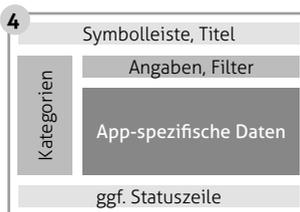
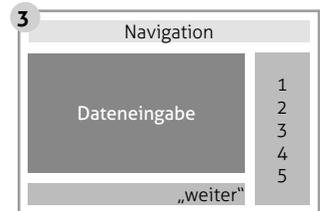
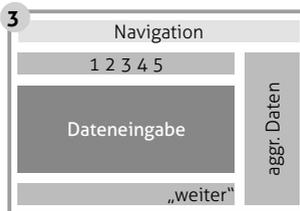
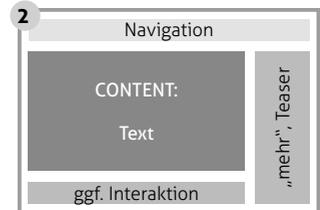
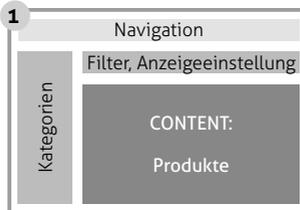
Abtrennung zu anderen Bereichen (z.B. durch Farbflächengrenzen, Linien, Weißraum)

Verwechslung der Funktionsbereiche ausschließen
einheitliche Aufteilung in allen Bereichen erleichtert Orientierung

Aufbau, Gliederung weckt Erwartungen

(durch Vorwissen, Erfahrungen des Nutzers)

1. Webshop
2. News- oder Content-Webseite
3. Assistent oder Bestellvorgang im Webshop
4. App zur Erfassung, Verwaltung, Auswertung: Musik, Bilder, Steuerdaten, Nachschlagewerke
5. Dateiverwaltung: Windows Explorer, Mac Finder



„The grid system is an aid, not a guarantee. It permits a number of possible uses and each designer can look for a solution appropriate to his personal style. But one must learn how to use the grid; it is an art that requires practice.“
Josef Muller-Brockmann (1914 – 1996)

Übliche Platzierungen

links:

- hierarchisch übergeordnete Elemente: Kategorien, Alben, Postfächer
- zeitlich zuerst folgende Schritte

oben:

- Steuerung der Elemente darunter oder Beeinflussung dieser
- Filter, Darstellungsoptionen
- Navigation, Menü

rechts:

- „mehr“: Inhalte (Teaser), Funktionen (Paletten, Unterstützungsangebote, Toolbar)

unten:

- Statusinformationen
- Bestätigung o. Abschluss der Arbeit mit dem aktuellen Bildschirminhalt: „weiter“, „speichern“, „teilen“
- Teil-Navigation (Seitenwechsel), sofern nachrangig und zwischen gleichartigen Ansichten
- Steuerung der Bildschirmdarstellung (Zoom)

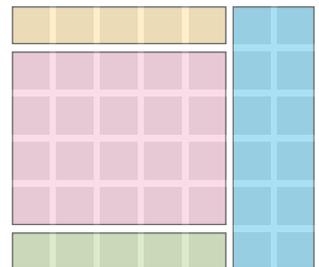
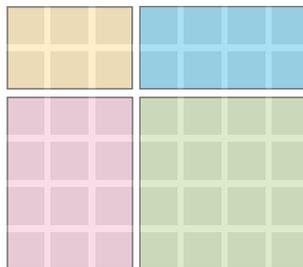
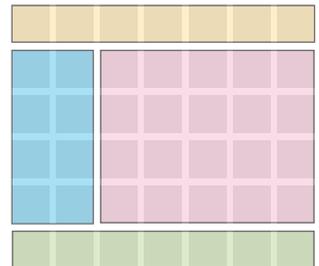
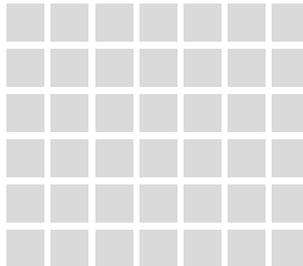
Gitter nutzen

Raster/Gitter helfen, den Bildschirm harmonisch zu strukturieren und Funktionsbereiche zu definieren

Funktionsbereiche und Aufteilung in allen Ansichten einheitlich => Orientierung, Vertrauen

Je nach Kontext

- fixe oder flexible Rasterbreiten
- Nutzer Bereichsgrenzen flexibel verschieben lassen
- Mindest/Maximalbreiten für Bereiche definieren
- Weißraum nutzen



Element	Funktion	Häufigste Probleme
Eingabefeld	Dateieingabe	Fehleingaben
Radio-Buttons	1 von X	Beschriftung
Auswahlmenü	1 von X	zu lange Liste
Slider, Regler	1 von X	präzise Eingabe
Combo-Box	1 von X + Dateneingabe	Fehleingabe
Checkboxen	X von Y	Beschriftung
Auswahlliste	X von Y	Mehrfachauswahl
Buttons	Aktion	Beschriftung

*„Don't give me choices.“
Ein User*

Eingabefelder

einzeliger Bereich, um Tastatureingaben entgegenzunehmen

Typ-spezifisch: Wort/e, Zahl, Passwort, Text

Feldgröße = Indikator für Eintragemenge (Nutzer schreiben selten über Feldgrenze hinaus => PLZ-Feld schmal, Suchfeld breit)

mehrere Felder ausrichten und sinnvoll gruppieren

optische Ähnlichkeit zu Papierformular => Layout, Platzierung weckt Assoziationen

eindeutig beschriften

ggf. Format spezifizieren (z.B: „TT.MM.JJJJ“, „mind 8 Zeichen“)

Fehleingaben behandeln & konstruktives Feedback

Checkboxen

Mehrere Optionen sind unabhängig voneinander wählbar.

Alle, einige oder keine können gewählt werden.

Einzelcheckbox aktiviert eine Option; nicht gewähltes Gegenteil ist eindeutig.

Ihre Hobbys:

- Romane lesen
- Filme schauen
- Musik hören
- Spaziergehen
- Freunde treffen

- Ich stimme zu.

- Ich nutze ein Smartphone.

- Nichtraucher

Radio-Buttons

Aus mehreren Optionen **muss** genau eine gewählt werden. Es kann nicht keine oder mehr als eine gewählt werden.

Wählen einer Option entfernt die Setzung einer anderen aus dieser Radio-Button-Gruppe.

Ihre Lieblingsfarbe:

- Rot
- Gelb
- Blau
- Grün

Ihre Position

- Praktikant
- Auszubildender
- Angestellter
- Leitung/Management
- Selbständig

weitere Elemente	Funktion	Häufigste Probleme
Tabulator/„Tabs“	zwischen gleichartigen/gleichwertigen Ansichten wechseln	Erkennen, dass mehr Ansichten verfügbar sind
Gruppierung	mehrere Elemente optisch zusammenfassen	Beschriftung
Ausklappbereiche	Inhalt erst nach Klick sichtbar	Relevantes nicht sichtbar
Funktionsmenü	Funktionen, Befehle anbieten	Befehl, Funktion finden
Ribbon	Funktionen, Befehle anbieten	missverständlich, uneindeutig
Symbolleiste, Palette	häufige Befehle anbieten	Icongestaltung
Dialog	Abfrage, Assistent	unerwartet; modal

Menüs

Aus mehreren Optionen kann genau eine gewählt werden (abhängig von Menü & Kontext)

Liste gleichartiger Elemente (Aktionen oder Objekte)

alphabetisch = verinnerlichtes Prinzip (alphabetisch : kategorisiert : zufällig = 1: 1,5 : 4)

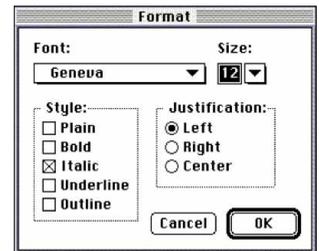
Hierarchie: max.(!) 4 Ebenen

Anzahl pro Ebene: ideal < 8

sind Hilfe, bieten Tipps für Gelegenheitsnutzer

Tastaturkürzel & schnelle Bedienung für Power User

Reihenfolge und räumliche Platzierung wird gelernt => keine adaptiven Menüs



Hinweise

Elemente nach Zusammengehörigkeit gruppieren und optische Blöcke/Einheiten bilden

Vor Beginn des nächsten Schrittes (Wechsel zum nächsten Bildschirm) auf Fehlerfreiheit bzw. Plausibilität prüfen

Beschriftung klar und einheitlich

kurze Begriffe, die der Nutzerwelt entstammen

konsistente Bezeichnung (Grammatik, Layout, Terminologie)

kongruente Begriff(spaar)e, Oppositionen (open/close)

ungewohnte Begriffe erhöhen Bearbeitungszeit

sinnvolle Vorauswahl/Default-Belegung anbieten

Klick auf „Label“ aktiviert ein Element

„natürliche“ Reihenfolge der Dateneingabe

wenn verschiedene Elemente möglich sind – Testen!

Tauglichkeit bewerten

Theorien & Modelle:
Richtlinien & Prozesse

Algorithmen & Prototypen:
UI Software-Tools

Kontrollierte Experimente:
Experten-Reviews & Usability-Testing

Grad der Arbeitsteilung und Höhe des Risikos (Budget, Zeitrahmen, Ressourcen) bestimmen
Formalisierungsgrad

Kalkulation

Projektmanagement +
Konzeption = 30%

Produktion + Test = 70%

+ Gebühren, Lizenzen,
Packaging, Infrastruktur

aber Kunde erhöht immer den Aufwand (Meetings, zusätzliche Präsentationen, Korrekturen, Anpassungen, Vorabversionen, umständliche Freigaben, stockender Informationsfluss, nachträgliche Anforderungsänderungen) => meist Mehraufwand von 30%

Vor der Software

Text: Papier, Textverarbeitung, Latex

grafisch (Mock-Ups): Papier, Paint, PowerPoint, Photoshop, InDesign, OmniGraffle, MS Visio, etc.

geringe Funktionalität (Prototyping): Flash, HTML (z.B. DreamWeaver), JavaScript, FileMaker, Oracle Form Wizard, MS Access

potenziell vollständige Funktionalität: Apple XCode, MS Visual Studio, Java, NetBeans

Das beste Tool ist das, das Sie am schnellsten, sichersten beherrschen! Abhängigkeit von Auftraggeber, Partnern o.a. Faktoren berücksichtigen!

Notationsformen

natürliche Sprache: meist der erste Ansatz, oft ergänzt um grafische Skizzen => länglich („Prosa“), vage, mehrdeutig => Korrektheit, Konsistenz, Vollständigkeit

Storybook: Briefing, (Exposé,) Screenplay (Aussehen, Funktionalität, Ablauf; Anforderungskatalog/Pflichtenheft), Zielgruppe, visuelle Ausprägung

Narrativ: Storyboarding, ggf. „Character Sheets“ für Icons und grafische Elemente

Drei Akte:

1. Exposition: Hauptdarsteller, Ort, Ausgangssituation
2. Entwicklung, Klimax: Konfrontation
3. Ende: Resolution, Ergebnisse, Folgen

Formalisiert: Flussdiagramme, Navigationsstruktur (Ein-
stiege, Ausstiege, Sprünge berücksichtigen!)

Formale Sprachen: Grammatiken (z.B. Backus-Naur-
Form, XML), Menüauswahl-Bäume

Transitionsdiagramme: z.B. UML State Transition Dia-
gram; Angabe von Übergangswahrscheinlichkeiten;
Entdecken von unbeachteten Übergängen, Sprüngen

Interface-Builder: Entwicklung von UI und Funktionalität
entkoppelt => parallele Entwicklung, Aufgabenteilung
(z.B. Software-Architekten), besseres Software-Design,
motiviert Nutzung von Standards

Rapid Prototyping: frühes Testen und Einbeziehung
anderer, Diskussion von Designs

Modelle für Interface-Design

Vier-Ebenen-Modell (Foley, 1982)

- **Konzeptuell:** Mentales Modell d, Anwenders
- **Semantisch:** Bedeutung der Ein- & Ausgaben
- **Syntaktisch:** Inhaltstragende Elemente
- **Lexikalisch:** Gerätespezifika & Mechanismen

GOMS-Modell (Card, Moran, Newell, 1983)

- **Goals:** Ziele und Teilziele des Anwenders
- **Operators:** Grundlegende verfügbare Aktionen
- **Methods:** Sequenzen von Operatoren
- **Selection Rules:** Regeln zur Methodennutzung

Widget-Level-Theorie

- **Definition** der Interface-Elemente
- **Reduktion** des Interface auf benötigte Widgets, Building Blocks
- **Standards** nutzen: Menüs, Buttons, Dialoge, Tabs, etc.
- Prototyping mit **Mock-up-Tools** oder Skizzen

Object/Action-Interface-Modell (Shneiderman, 2001)

- Verstehe die **Aufgabe:** Objekte (aus der realen Welt) + Aktionen (anwendbar auf die Objekte)
- **Metaphorische Repräsentation** der Objekte & Aktionen
- **Visualisierung** der Ausführung
- **Hierarchie** der Objekte und Aktionen

Automation & Human Control

Bewusst entscheiden, was der Computer kann und was der Mensch tun soll. Für Anwender gilt:

- Routine vermeiden
- ermüdende Aufgaben vermeiden
- fehleranfällige Aufgaben vermeiden
- + wichtige Entscheidungen fällen oder bestätigen
- + unerwartete Situationen behandeln
- + künftige Aktionen planen

High-Level-Tasks zerlegen

Häufigkeit, Risikoanfälligkeit von Aufgaben bewerten

Interviews führen &

Nutzer beobachten

Die 7 Stufen der Aktion

1. Ziel formulieren
 2. Plan formulieren
 3. Aktion spezifizieren
 4. Aktion/en ausführen
 5. Systemstatus wahrnehmen
 6. Systemstatus interpretieren
 7. Resultat/e auswerten
- Donald Norman (1988)*

Problempunkte

Kluft der Ausführung: Plan des Anwenders ≠ verfügbare Aktionen

Kluft der Auswertung: Systemstatus bzw. dessen Repräsentation ≠ Erwartungen des Nutzers

Nutzer verfolgt inadäquates Ziel

Interface-Objekt nicht findbar

Unwissen des Nutzers, wie Aktion spezifiziert/ ausgelöst wird

unangemessenes oder irreführendes Feedback

Regel 0: Sei ein Team

Keine Disziplin allein kann ein gutes Interface schaffen. Kenne deine Grenzen, hole dir Hilfe und teste so oft wie möglich!

Regel 1: Denk an den Nutzer

Der User hat ...	Der User ist ...
... Erwartungen.	... ungeduldig.
... keine Zeit.	... unaufmerksam.
... immer Recht.	... faul.

Usability ist eine Dienstleistung, ein Service (keine Selbstbefriedigung, kein Selbstzweck)

Respekt vor dem (Vor)Wissen des Anwenders und dessen Erfahrungen

Der User wird gebraucht (wenns automatisch ablaufen würde, benötigen wir den User nicht)

Der User tritt mit Erwartungen an die Software (Aufwand, Dauer, Ziel, Ergebnis)

Regel 2: Schaffe Einheitlichkeit

Konsistenz in Erscheinung und Funktion

Gleichheit => Gleichheit

Ähnlichkeit => Ähnlichkeit

Verschiedenheit => Verschiedenheit

Orientierung an Standards, Best Practice

gleiches Aussehen bedeutet gleiche

Funktionalität => Vorhersagbarkeit

Standards nutzen! (statt neue erfinden)

Probleme bei Inkonsistenzen:

- Lernaufwand, schwieriger zu merken
- produzieren Fehler
- bremsen den Anwender

=> Grammatiken charakterisieren ein komplettes Set von Tasks => Entwicklung einer Syntax (High Level) für GUI-Aktionen, Tastenkommandos

Regel 3: „Rede mit mir“

neutraler Tonfall, keine „Ich“-Aussagen
 klare, sachliche, eindeutige An- & Aussagen
 informatives & konstruktives Feedback
 aktueller & korrekter Status
 länger als 2s keine Reaktion = kaputt

Regel 4: Eins. Zwei. Drei.

Erst kommt erst, und dann kommt dann.
 Jede Handlungssequenz hat vom Nutzer ausgelösten
 Anfang und erkennbares Ende.

Regel 5: Erlaube keine Fehler

Umkehrbarkeit aller Aktionen
 alle (!) Eingaben verifizieren
 Exception Handling (sodass bei Absturz kein
 Datenverlust entsteht)
 Aus Fehlern lernen UI-Entwickler (hoffentlich) und
 User (und entwickeln Erwartungen bzw. Sorgen)
 geeignete Vorbelegungen, Default-Werte

Regel 6: Wer hat die Kontrolle

User hat stets Kontrolle => Abbruch-Möglichkeit (für alles)
 Unterstützung für verantwortungsvolle Entscheidungen
 Wahlfreiheit bzgl. Vorgehen
 Rückgängig („Undo“) bzw. Schritt zurück
 Vorbelegungen/Default-Werte änderbar
 Nutzer löst Aktionen aus und reagiert nicht nur

Regel 7: Bedenke die Wirkung

Software ist hilfreich, positiv, unterstützend, erleichternd – und so wird die Nutzung auch empfunden.

Fehler vermeiden

Fehlermeldungen:

- spezifisch
- positiv im Tonfall
- konstruktiv

Unterscheidung:

- Slip, Schusselfehler, Versehen
- Mistake, systemischer Fehler, falscher Weg

Förderung korrekter
 Aktionen:

- unangemessene Aktionen deaktivieren
- Auswahl statt Eingabe
- Auto-Vervollständigen

Komplette Aktionssequenzen

- einzelne abstrakte Kommandos statt Sequenzen langer Kommandos
- Makros & Unterprogramme

Testen

„1 Usertest ist besser als 0 Tests“ (Jakob Nielsen)

Teste schnell und teste oft (Steve Krug)

Je Test die drei schlimmsten Fehler beheben

Lösungsideen ignorieren

Fragen, Probleme, Hindernisse ernstnehmen

Die Kern-Norm für User Interfaces (**DIN EN ISO 9241**) liegt seit April 2006 in überarbeiteter Form vor. Dabei wurden der Geltungsbereich auf interaktive Systeme aller Art erweitert und inhaltlich zur ISO 9241 gehörige Normen integriert sowie ein neues Nummerierungsschema eingeführt. Die sieben Grundsätze der Dialoggestaltung der alten Norm ISO 9241-10 wurden in der neuen ISO 9241-110 grundsätzlich beibehalten aber durch eine Präzisierung weiter verbessert.

9241-11

„Gebrauchstauglichkeit bezeichnet das Ausmaß, in dem ein Produkt, System oder ein Dienst

durch bestimmte **Benutzer**

in einem bestimmten **Anwendungskontext** genutzt werden kann,

um bestimmte **Ziele**

effektiv, effizient und zufriedenstellend zu erreichen.“

- Effektivität zur Lösung einer Aufgabe
- Effizienz der Handhabung des Systems
- Zufriedenheit der Nutzer einer Software

9241-110

„Grundsätze der Dialoggestaltung“

Aufgabenangemessenheit: geeignete Funktionalität, Interaktionen

Selbstbeschreibungsfähigkeit:

Hilfen, Rückmeldungen

Lernförderlichkeit:

Anleitung d. Benutzers; geeignete Metaphern

Steuerbarkeit:

Steuerung des Dialogs durch den Benutzer

Erwartungskonformität:

Konsistenz, Anpassung an Benutzermodell

Individualisierbarkeit:

Anpassbarkeit an Bedürfnisse & Kenntnisse

Fehlertoleranz:

klare Meldungen, unaufwändige Korrekturen

9241-210

„Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme“

Das Design basiert auf einem klaren Verständnis der Benutzer, Aufgaben und Umgebung.

Benutzer werden aktiv in Design und Entwicklung einbezogen.

Das Design folgt einer benutzerzentrierten Bewertung.

Der Prozess ist iterativ.

Das Design zielt auf eine vollständige „User Experience“.

Das Design-Team enthält multidisziplinäre Fähigkeiten und Perspektiven.

„Die Grundsätze der Ergonomie sind insbesondere auf die Verarbeitung von Informationen durch den Menschen anzuwenden.“

(Bildschirmarbeitsverordnung, Absatz 20)

DIN EN ISO 14915

Eignung für das **Kommunikationsziel**: Präsentation der Informationen ist für die Zielerreichung geeignet.

Eignung für **Wahrnehmung und Verständnis**: Übermittelte Information sind leicht verständlich und leicht erfassbar.

Eignung für die **Exploration**: Es sind keine Vorkenntnisse oder Erfahrungen nötig, gewünschte Informationen zu finden oder beabsichtigte Aufgaben zu erledigen.

Eignung für die **Benutzungsmotivation**: Ausrichtung auf Bedürfnisse der Besucher sowie ansprechende Präsentation und zielgerichtete Führung motivieren User.

„Multimedia-Norm“ von 2000 in Ergänzung zur Normenreihe 9241. Messung der Kriterienerfüllung durch experten- oder nutzerorientierte Verfahren, Tests, z.B. Cognitive Walkthrough.

„80 Prozent der Ergebnisse werden in 20 Prozent der Gesamtzeit eines Projekts erreicht. Die verbleibenden 20 Prozent der Ergebnisse benötigen 80 Prozent der Gesamtzeit und verursachen die meiste Arbeit.“

Vilfredo Pareto (1848 – 1923)

ISO 13407

1. **Nutzungskontext verstehen**: dokumentierte Beschreibung d. relevanten Benutzer, ihrer Arbeitsaufgaben & Umgebung
2. **Anforderungen spezifizieren**: Zielgrößen auf Kompromissebene ableiten, Teilung der Aufgaben für Menschen & Technik
3. **Lösungen produzieren**: Prototyping o.a. iterativer Prozess
4. **Lösungen bewerten**: Prüfung mit z.B. Experten-Reviews, Usability-Tests, Befragungen o.ä.
5. => nächste Iteration
11/2000 als DIN-Norm veröffentlicht; seit 01/2011 durch EN ISO 9241-210 ersetzt. Die Norm beschreibt einen prototyp-orientierten Entwicklungsprozess.

DIN 66272

Funktionalität: Implementierung & Ausführbarkeit aller geforderten Funktionen

Zuverlässigkeit: Verfügbarkeit & Korrektheit der Software

Benutzbarkeit: Bedienbarkeit, Erlernbarkeit, Reaktion auf Fehleingaben

Effizienz: zeitliches Verhalten bei Anfragen und Bearbeitungen, Ressourcenverbrauch

Änderbarkeit: Aufwand für Verbesserungen, Fehlerbeseitigungen oder Anpassung an Umgebungsänderungen.

Übertragbarkeit: Einsetzbarkeit auf anderen Systemen (Hard- & Software)

2006 ersatzlos zurückgezogen. Definition von Kriterien, die für eine qualitative Software zu erfüllen und zu testen sind.

ISO IEC 12119

(„Software-Erzeugnisse – Qualitätsanforderungen und Prüfbestimmungen“) beschreibt, was an Software-Produkte zu prüfen ist und wie dies geschehen soll. Sie beschränkt sich dabei aber fast vollständig auf technische und funktionale Qualitätsmerkmale.

Wann mit Usability anfangen?

So früh wie möglich. Jede gute Stunde Planung, Konzept spart zehn schlechte Stunden Fixes, Schulung, Nacharbeiten und v.a. Unzufriedenheit und Wahrnehmung.

Was kostet Usability?

Nichts extra. Dafür ergeben sich Einsparungen bei Schulung und Fehlerbehandlung sowie Steigerungen bei der Bedienungsgeschwindigkeit (Leistungsfähigkeit) der Anwender.

Wieso genügen Hilfe, Anleitung nicht?

Weil niemand liest. Es ist nahezu unmöglich, Software und Dokumentation synchron zu halten (Stichwort: Updates). Die Struktur und Gebrauchstauglichkeit der Dokumentation ist eine weitere Baustelle, um die man sich zusätzlich kümmern muss. Eine Dokumentation kann nur fachlich, inhaltlich unterstützen.

Was ist der Unterschied zwischen Design und Usability?

Vereinfacht gesagt bezieht sich „Design“ auf die konkrete Optik, die möglichst schick sein soll. Usability dagegen bezieht sich auf die Funktionalität und Benutzbarkeit. In den meisten Fällen gilt: Je schicker, desto schlechter benutzbar (der Umkehrschluss stimmt allerdings nicht).

FFF: form follows function. Das Design unterstützt die Usability – andersherum geht nicht.

Wieso gibt es Probleme, obwohl man sich an Standards hält?

Standards sind nur „Denkhilfen“, Guidelines, Orientierungen. Beim Nutzer wecken sie Erwartungen, die der konkrete Fall evtl. nicht erfüllen kann oder von denen er abweicht. Ein Standard deckt nicht alle situativen Anforderungen der konkreten Nutzung ab.

Explizite Standards (DIN) beschreiben die Funktion, implizite Standards (Aussehen anderer) beziehen sich dagegen auf die Optik. Jeder Fall ist individuell.

Wie teste ich?

Kollegen oder potenzielle Anwender an die Software oder Webseite setzen, eine klare Aufgabe geben und zuhören! Keine Lösungen oder Meinungen sammeln, sondern die Fragen, Probleme, Unverständlichkeiten, Hindernisse, Unsicherheiten, Fehler.

Wie überzeuge ich meine Chefs?

Mit Mathematik. Anzahl der Vorgänge pro Jahr mal voraussichtliche Ersparnis (Zeit, Kosten, Fehlerbehebung) = Summe X. Erreichen der Ersparnis kostet Arbeitsaufwand Y. Wenn $X > Y$, dann ist der Chef zufrieden (es sei denn, es stehen andere wichtigere Aufgaben an).

Wie fange ich am besten an?

Mit kleinen Schritten. Bei jedem Element, das die Nutzer betrifft, z.B. Button- oder Feldbeschriftungen, Hinweismeldungen, Anordnung der Elemente. Die Nutzer nach häufigen Fehlern oder Unverständlichkeiten fragen und deren Ursachen beseitigen.

Welche Online-Tools helfen?

www.balsamiq.com (Mock-up-Tool; kostenloser Demo-Modus genügt)
www.rapidusertests.com (User tests online durchführen; kostenpflichtig)
kuler.adobe.com (Farb-Mischer)
www.typetester.org (Schriften live vergleichen)

„So if you make the Mac boot ten seconds faster,
you've saved a dozen lives.“

*Steve Jobs (1955 – 2011), 1983 an die Mac-Entwickler
10 Sekunden x 5 Mio User x 200 Arbeitstage/Jahr
= 10.000.000.000 Sekunden = 317,1 Jahre*

Gibt es Lese-Tipps?

www.nngroup.com/articles/ (engl.)
www.usability.ch/bn/news/alertbox.html
www.fit-fuer-usability.de
www.kommdesign.de
www.contentmanager.de/usability/
www.usabilityblog.de
Steve Krug „Don't Make me Think“,
„Rocket Surgery Made Easy“
Ben Shneiderman „User Interface Design“ (engl.)
Theo Mandel „User Interface Design“ (engl.)
Jakob Nielsen „Web Usability“
Jef Raskin „Das intelligente Interface“
Macintosh User Interface Guidelines (ask Google)
Windows Interface Guidelines (ask Google)

- 1833 – Charles Babbage beginnt Arbeit an der Analytical Engine
- 1846 – Der Morse-Telegraf-Service benötigt keine Morse-Code-Übersetzer mehr
- 1884 – Patent für Lochkarten von Herman Hollerith
- 1960er – Kommandozeile
- 1960er – Datenbanken
- 1967 – Jef Raskin: „A Hardware-Independent Computer Drawing System Using List-Structured Modeling: The Quick-Draw Graphics System“
- 1968 – „Die Mutter aller Demos“ von Douglas Engelbart:
<http://sloan.stanford.edu/MouseSite/1968Demo.html>
- 1970 – Der erste mit Textverarbeitung verfasste Roman: „Bomber“ von L. Deighton
- 1973 – Die erste GUI im Einsatz: Xerox Alto (WIMP-Basis)
- 1976 – Die erste PC-Textverarbeitung: Electric Pencil
- 1979 – Die erste Tabellenkalkulation: VisiCalc
- 1981 – Der erste mit PC-Textverarbeitung verfasste Roman: „Oath of Fealty“
von Jerry Pournelle und Larry Niven
- 1984 – Der erste Computer mit GUI für den Massenmarkt: Apple Macintosh
- 1984 – PostScript von Adobe: flexible Seitenbeschreibung => Büro-Revolution
- 1985 – Die Desktop-Publishing-Revolution beginnt (PageMaker, Quark XPress)
- 1989 – Tim Berners-Lee schlägt ein WWW vor (HTML, Webserver)
- 1991 – Multimedia wird wahr: QuickTime (Framework, Player, Dateiformat)
- 1991 – Der wichtigste Lehrmeister: Microsoft Windows 3 (mit Solitaire)
- 1993 – Der erste PDA: Newton
- 1994/95 – Das erste Smartphone: „Simon“ von BellSouth und IBM
- 1995 – Microsoft Windows 95: GUI im Mainstream
- 2000 – MacOS X (GUI als Schicht ausgelagert)
- 2001 – iTunes (Verwaltung großer Datenmengen für alle)
- 2004 – iTunes Music Store (Akzeptanz für den Erwerb virtueller Güter)
- 2007 – Grafische Menüs: Ribbons
- 2007 – Das erste Gerät mit überzeugender Touch-Bedienung: iPhone
- 2010 – Das erste Tablet: iPad
- 2012 – Microsoft Windows 8
- 2012 – Computer direkt im Alltag: Google Glass

Usability

	Paradigma	Interface	Usability-Ära	Fokus
1970/ 80er	Operator bedient Maschine	Kommandozeile: MS-DOS, CP/M, Shell, Terminal	Ergonomie: Mensch-Maschine-Interaktion	Praxis, Bedienung, physische Effizienz
1980/ 90er	User benutzt Computer als Werkzeug	grafische Oberfläche: Mac, Windows, Workbench; WIMP, Wysiwyg	Psychologie: mentale Vorgänge	Theorie, Informationsverarbeitung, psychische Effizienz
2000/ 2010er	Manager delegiert an Assistenten	Attentive User Interface: Assistenten, Google, Siri	Phänomenologie: absichtsvolles Agieren	Diskurse, Kontext, Unterstützung

„Here's the new application.
It's got one window.
You drag your video into the window.
Then you click the button that says burn.
That's it.“

*Steve Jobs (1955 – 2011),
2000 über das User Interface für iDVD*

Abstraktionsgrade

1. Gefühl => Sprache (Semantik, Lexik, Syntax, Grammatik): Es wäre schön, wenn XX bei meiner Party wäre => „Möchtest du zu meiner Party kommen?“
2. Sprache => Text (Alphabet, Notationskonventionen)
3. Text => Eingabedaten (mechanisch)
4. Eingabedaten => kodierte Informationen (digitale Konvertierung)
5. kodierte Information => Ausgabedaten: Anzeige auf Bildschirm oder Übertragung an andere Geräte
6. Ausgabedaten => Validitätsprüfung (z.B. Rechtschreibkontrolle, Spamfilter, Übertragungsfehler)



Icon-Illustrationen auf Umschlag: <http://openclipart.org/detail/192554>