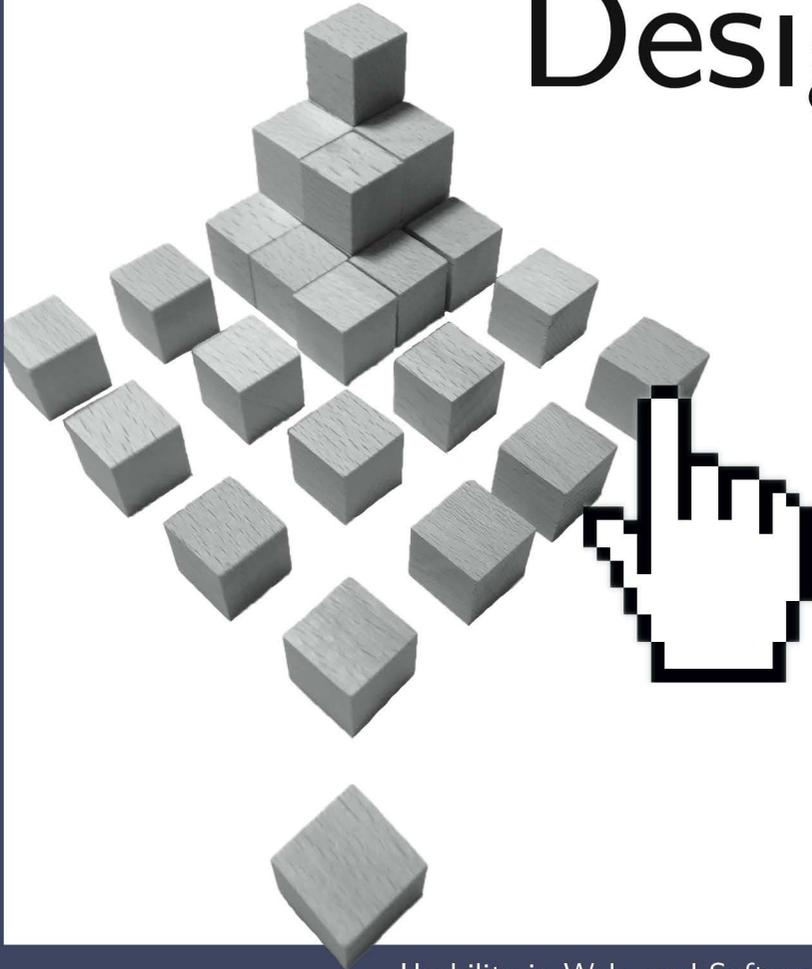


Einstieg und Praxisratgeber

Alexander Florin

# User Interface Design



Usability in Web- und Software-Projekten

[Buch bei Amazon aufrufen »](#)

[Autorenseite Alexander Florin bei Amazon »](#)

[Details zum Buch bei www.axin.de »](#)

[Alexander Florin bei Twitter »](#)

[Alexander Florin bei LinkedIn »](#)

[Alexander Florin bei Xing »](#)

# Inhaltsverzeichnis

<b>1</b>	<b>Häufige Fragen</b>	<b>1</b>
<b>2</b>	<b>Einstimmung</b>	<b>11</b>
	Usability . . . . .	16
	Der Nutzer im Zentrum . . . . .	24
<b>3</b>	<b>Unsere Nutzer</b>	<b>31</b>
	Eine nützliche Software . . . . .	34
	User-Steckbriefe . . . . .	36
	Novizen und Experten . . . . .	37
	Weitere Nutzergruppen . . . . .	47
	Erfassen, was Nutzer wollen . . . . .	54
	Checkliste: Erfassen, was Nutzer wollen . . . . .	64
	Checkliste: Für die Nutzer . . . . .	67
<b>4</b>	<b>Unsere Computer</b>	<b>69</b>
	Eine kurze Geschichte des User-Interface . . . . .	72
	Drei Phasen der Entwicklung . . . . .	101
	Checkliste: Computer . . . . .	105
<b>5</b>	<b>Ziele und Aktionen</b>	<b>107</b>
	Die Sieben Stufen der Aktion . . . . .	108
	Meine Ziele . . . . .	114
	Vereinfachtes Aktionsmodell . . . . .	118
	Klarheit in der Darstellung . . . . .	118
	Vier-Ebenen-Modell . . . . .	120
	GOMS-Modell . . . . .	122
	Widget-Level . . . . .	126
	Object/Action-Interface . . . . .	130
	Checkliste: Ein Modell für mein Projekt . . . . .	133
<b>6</b>	<b>Ordnung auf dem Bildschirm</b>	<b>137</b>
	Die Grenzen des Bildschirms . . . . .	138

Die Mitte im Zentrum . . . . .	150
Der menschliche Blick . . . . .	154
Präsentation der Informationen . . . . .	161
Funktionale und optische Strukturen . . . . .	171
Checkliste: Ordnung schaffen . . . . .	185
<b>7 Design-Nussschale</b>	<b>187</b>
Visuelle Variablen . . . . .	188
Position und Größe . . . . .	193
Farbe, Helligkeit und Textur . . . . .	203
Neigung, Form und Gestalt . . . . .	215
Schrift . . . . .	217
Icons und Symbole . . . . .	226
Technische Design-Hinweise . . . . .	231
<b>8 Standardelemente</b>	<b>243</b>
Funktionselemente . . . . .	245
Checkliste: Funktionselemente . . . . .	277
Strukturelemente . . . . .	279
Checkliste: Strukturelemente . . . . .	286
Informationselemente . . . . .	287
Suchen und Finden . . . . .	290
<b>9 Das eigene Interface</b>	<b>295</b>
Start-up-Modell . . . . .	297
Qualitätsanspruch . . . . .	298
Checkliste für die Anforderungsaufnahme . . . . .	307
Mein Recht auf Usability . . . . .	308
Usability testen . . . . .	324
<b>10 Ganzheitlicher Ansatz</b>	<b>331</b>
<b>11 Glossar &amp; Index</b>	<b>353</b>

# 1 Häufige Fragen



Wer die Antworten bereits meint zu kennen, findet in diesem Buch unterstützende Argumente oder Hinweise. Was er oder sie jedoch kaum finden wird, sind endgültige Weisheiten, wie in einer bestimmten Situation die perfekte Lösung aussieht. Erstens gibt es keine perfekte Lösung, nur besser oder weniger geeignete. Zweitens handelt es sich bei Usability nicht um eine Rezeptsammlung, der man nur zu folgen braucht. Usability ist vielmehr das Verständnis, wie der Geschmacksapparat funktioniert, welche Servier- und Speise-Konventionen bestehen, welche Zutaten und Küchenwerkzeuge zur Verfügung stehen – aus diesen wird ein dem Anlass entsprechendes Mahl zubereitet und würdig serviert.

Die häufigen Fragen werden meist von denen gestellt, die sich bislang nicht eingehender mit Usability beschäftigt haben. Für diese ist Usability meist entweder ein Kostenfaktor oder ein Teil der Umsatzoptimierungsstrategie. Wie alle gern verwendeten Schlagwörter verliert der Begriff „Usability“ zunehmend an Schärfe und Präzision – er wird schlichtweg für alles verwendet, was irgendwie mit der Bedienung von Webseiten oder Software zu tun hat. Für unsere Zwecke verstehen wir das Kunstwort „Usability“ als Mischung aus Gebrauchstauglichkeit, Anwenderfreundlichkeit und einfacher Bedienung.

## Wer sollte dieses Buch lesen?

Alle Personen, die Webseiten oder Software betreiben, verantworten, entwickeln, planen, bewerben, konzipieren, umsetzen, testen oder einfach nur einige Zusammenhänge besser verstehen wollen:

**Unternehmer, Leiter:** die tatsächlichen Nutzer-Interessen verstehen

**Projektmanagement:** Usability in Planung und Projektalltag integrieren

**Anforderungserfassung:** Nutzer verstehen und geeignete Dokumente erstellen

**Umsetzung:** technische Anforderungen berücksichtigen, Design-Grundlagen und Bedienkonzepte kennen, verfügbare Elemente und Werkzeuge anwenden

**Marketing:** als Schnittstelle zwischen Entwicklern und Nutzern passendes Erwartungsmanagement betreiben

Gelegentlich erweitern theoretische oder historische Ausflüge die Perspektive und inspirieren zu anderen Denkbahnen. Als Schnelleinstieg oder zum Nachschlagen dient das Glossar am Buchende, das die häufigsten Begriffe und Konzepte kurz erläutert. Viele Kapitel bzw. Unterkapitel werden durch Checklisten beschlossen. Diese unterstützen bei Entscheidungen und fassen die für die Praxis wichtigsten Punkte zusammen.

Wie beim Lernen eines Instruments benötigt es jedoch mehr als nur die richtige Lektüre. Für gutes Spielen sind mindestens 2.000 Übungsstunden nötig; wer auf Profi-Niveau spielen will, benötigt etwa 10.000 Stunden. Gleichmaßen ist für Usability-Experten die praktische Erfahrung entscheidend, Bücher (wie dieses) können allenfalls Anregungen und Hilfestellung bieten.

Entscheidend ist vor allem, dass Usability nicht als Einzeldisziplin begriffen wird, sondern gut in alle Prozesse integriert ist. Wie beim Qualitätsmanagement oder Controlling kann der Erfolg nur interdisziplinär entstehen. Oft bedarf es einer Person oder kleinen Personengruppe, bei der die Fäden koordiniert zusammenlaufen – aber ohne die gute und kontinuierliche Zusammenarbeit mit anderen Unternehmensbereichen oder Projektbeteiligten verpufft der Effekt oder bleibt kosmetische Fassade. Für solche Usability-Verantwortlichen und alle Personen, die mit diesen zusammenarbeiten, ist dieses Buch gedacht und soll eine **gemeinsame Grundlage** bilden. Auf dieser aufbauend entwickeln die Personen dann gemeinsam ihre Lösungen für ihre Nutzer.

## Wann mit Usability anfangen?

So früh wie möglich. Usability ist nicht der Zuckerguss, der am Ende drübergegossen wird, sondern das Mehl, das alles gut zusammenhält. Auch alle

---

Funktionen und technischen Aspekte haben Auswirkungen auf die Bedienbarkeit. Die Belange der Nutzer bereits am Anfang zu berücksichtigen, zahlt sich insgesamt aus, denn diese bezahlen mit ihrem Geld, ihrer Aufmerksamkeit oder ihrer Anerkennung.

So wie zu einem erfolgreichen Auftritt mehr gehört als die richtige Kleidung – eben auch die korrekten Umgangsformen, Körperhaltung und sprachlichen Codes –, so gehört zu einer guten Software die Usability essenziell dazu. Sonst erhält man den „Proll im Anzug“ oder den „Ritter im Penner-Look“, und beide bieten keinen guten Gesamteindruck.

## Was kostet Usability?

Usability kostet nichts extra, wenn sie von Anfang an einbezogen wird. Gute Usability kann die Summe aller entstehenden Kosten über den gesamten Life-Cycle einer Software oder Webseite verringern. Vor allem nachgelagerte Aufwände – oft versteckte und gern ignorierte Kostenverursacher – sinken beträchtlich. Mittel- und langfristig entstehen Einsparungen:

- ◊ Der interne Schulungs- und externe Beratungsaufwand sinken. Wissen kann von einer Software oder Webseite auf eine andere übertragen werden und muss nicht für jede Anwendung separat erlernt werden.
- ◊ Die Performanz der Anwender steigt. Sie werden rascher zu Experten, können schneller arbeiten und dadurch mehr schaffen.
- ◊ Die Nutzer machen weniger Fehler, und es werden weniger Ressourcen für die Fehlerbehandlung benötigt.

Außerdem steigt die Nutzerzufriedenheit. Das Produkt bietet wenig Frustanlässe und hat eine hohe Empfehlungswahrscheinlichkeit.

Die praktische Erfahrung bestätigt, dass eine gute Planungsstunde zehn schlechte Reparatur-, Schulungs- oder Entwicklungsstunden spart. Das Thema Usability kann einen guten „Vorwand“ bieten, um bereits in der Konzeptions- oder Planungsphase kostenintensive Nutzerprobleme zu identifizieren und zu vermeiden.

Jede realistische Projektkalkulation berücksichtigt die voraussichtlichen direkten und indirekten Gesamtkosten für die gesamte Lebensdauer: Schulung, Wartung, Nachbesserung, Pflege, Erweiterung, Änderungen, verbaute Wege. Als Richtwert können drei Jahre dienen.

## Was sind die schlimmsten Ressourcenfresser?

Häufig entstehen sogenannte technische Schulden in diesen Bereichen:

- ◊ Vernachlässigen von Dokumentation
- ◊ Verzicht auf Versionsverwaltung, Datensicherung, Kontinuierliche Integration
- ◊ Nachlässiges Testen
- ◊ Keine Coding-Standards
- ◊ Folgen von Anti-Mustern, Anti-Pattern
- ◊ Missachtung von Warnungen der Werkzeuge (Compiler, Code-Analyse)
- ◊ Furcht vor der Korrektur von zu großem oder zu komplexem Code

Was in dem jeweiligen Moment Zeit und Ressourcen spart, muss durch zusätzliche Ressourcen in der Zukunft ausgeglichen werden. Ständig muss eigentlich fertige Software korrigiert, angepasst oder erweitert werden. Änderungen sind aufwändig, weil die ursprüngliche Lösung nur auf einen unflexibel-konkreten Fall zugeschnitten wurde. Fehler werden erst im Produktivbetrieb erkannt. Jede Nachbesserung beinhaltet die Folgekosten des erneuten Bereitstellens und Aktualisierens; auch wenn diese Prozesse stark automatisiert ablaufen können, so sind sie dennoch nicht umsonst zu haben. Der Entwickler selber versteht seinen Code nach einigen Monaten bereits selbst kaum noch und benötigt lange, um dessen Funktionsweise nachzuvollziehen und eine Änderung vorzunehmen. Die Auswirkungen auf andere Code-Teile errät er dabei. Ist es nicht sein eigener Code, ist die Wartung ohne Coding-Standards für ihn unanständig aufwändig.

Die Problemliste, die Wikipedia bei „Anti-Pattern“ auflistet, bietet viele Anregungen, was es zu vermeiden gilt.

So lassen sich durch besseres Arbeiten an aktuellen Projekten deren technische Schulden in der Zukunft reduzieren, was Ressourcen freisetzt.

## Wieso genügen Hilfe, Anleitung nicht?

Weil niemand liest. Vor allem die Nutzer nicht. Hand aufs Herz: Wie viele ungelesene Bedienungsanleitungen liegen bei Ihnen zuhause? Bei Ihren Kollegen? Dennoch ist eine gute Anleitung bzw. Dokumentation unverzichtbar; denn einige Nutzer benötigen diese. Die Nutzerdokumentation muss dem aktuellen Stand der Software oder Webseite entsprechen. Wird sie parallel erstellt, hat das oft positive Auswirkungen auf die Konzeption, da durch die Textform einige Aspekte auffallen, die sonst übersehen werden.

---

Für die Dokumentation ist eine geeignete Struktur zu wählen, und auch für die Dokumentation gilt der Usability-Anspruch: gut verständlich, leicht zu nutzen, passend zu den Nutzerzielen strukturiert. Für Webseiten bieten sich häufig die sogenannten FAQ an. Diese „Frequently Asked Questions“ (Häufigen Fragen) beantworten Fragen, die sich die Nutzer stellen oder stellen könnten. Das Format ist etabliert, und wenn es gut angewendet wird, stellt es einen wirklichen Mehrwert dar. Ein Webshop sollte sich beispielsweise folgende Fragen stellen:

- ◊ Wie finde ich das geeignete Produkt?
- ◊ Wie bestelle ich? Wie läuft eine Bestellung ab?
- ◊ Welche Zahlungsoptionen gibt es?
- ◊ Wie kann ich reklamieren oder umtauschen?

Über solche und ähnliche Fragen lässt sich auch ein komplexer Webshop gut dokumentieren. Die Antworten beschränken sich dabei jeweils auf die konkrete Frage, unterstützend sind sie untereinander verlinkt. Die ideale Antwort besteht aus einem Satz, der die Frage kurz und bündig beantwortet. Dies ist oft nicht möglich, daher gibt der erste Absatz eine allgemeine Antwort, und weitere Absätze beschreiben Details oder geben ergänzende Hinweise.

Für Software ist der sehr ähnliche „Wie tue ich“-Ansatz geeignet. Dabei werden die Aufgaben aus Nutzerperspektive benannt und anschließend die nötigen Schritte in der korrekten Reihenfolge beschrieben (und durch Screenshots illustriert): „Bild einfügen“ [die Aufgabe des Nutzers]

1. Cursor an der Stelle platzieren, wo das Bild erscheinen soll. [„erscheinen“ = Ziel des Nutzers]
2. Im Menü „Einfügen“ den Eintrag „Bild“ anklicken. [genaue Angabe des Ortes und der Aktion]
3. Im erscheinenden Dateiauswahlfenster die gewünschte Datei wählen. (Die Dateiauswahl ist ausführlich in XY beschrieben) [ergänzender Hinweis für Novizen]
4. Den Button „Einfügen“ anklicken. [Angabe, wie die Aktion ausgelöst wird – das hört sich schon nach fast fertig an.]
5. Das Bild erscheint an der Cursor-Position. [das Ziel ist erreicht]
6. Durch Rechtsklick auf das Bild erhalten Sie Möglichkeiten zur Formatierung des Bildes, diese sind in Kapitel XY beschrieben. [Hinweise zu wahrscheinlichen Folgeaktionen]

Auf diese Weise wird jede Funktion (ausgehend von der Nutzeraufgabe) auf ein bis maximal zwei Seiten beschrieben.

Zur Software-Dokumentation gehört noch mindestens ein Kapitel, das die Elemente und deren Grundbedienung sowie das Bedienparadigma vorstellt.

Auch wenn viele Nutzer die Dokumentation nicht lesen, wissen sie doch, dass sie da ist. Bereits ihr Vorhandensein und die Möglichkeit, im Bedarfsfall auf sie zugreifen zu können, erhöhen das Sicherheitsgefühl erheblich. Dazu wird sie an geeigneten Stellen im Programm integriert und der Aufruf sowie die Recherche darin sind einfach möglich. Ergänzend ist an möglichst vielen Stellen die entsprechende Hilfe-Seite aufrufbar (oft als „?“-Icon). Auch Kurztexte in der Oberfläche verlinken für detaillierte Informationen auf entsprechende Hilfe-Seiten.

## Wie unterscheiden sich Design und Usability?

Die beiden Begriffe werden gern synonym verwendet, wie auch von Steve Jobs: „Design ist nicht nur, wie es aussieht oder sich anfühlt. Design ist, wie es funktioniert.“ Im praktischen Alltag bietet sich eine Trennung an: Design ist das konkrete Aussehen, Usability umfasst alle Aspekte der Bedienung, auch die abstrakten, esoterischen, metaphorischen oder prozessuralen.

Das Design beschreibt bei Software und Webseiten „lediglich“ die Optik. Die Usability dagegen gibt an, wie gut etwas benutzbar ist, damit ist sie ein funktionales Kriterium, das nicht von persönlichem Geschmack abhängig ist. Die Usability lässt sich messen (z.B. als benötigte Zeit, die für eine Aufgabe benötigt wird).

Das Design ist der sichtbarste Aspekt der Usability, doch besitzen auch viele Programme und Webseiten ohne aufwändige Gestaltung eine hohe Usability. Zugespitzt lässt sich formulieren: „Je schicker, desto schlechter benutzbar“. Nutzern fällt es in Designexzessen schwer, das gewünschte Bedienelement schnell zu finden, oder die Gestaltung verstellt den Blick auf die Funktionalität. Der Umkehrschluss gilt übrigens nicht, eine hässliche Webseite oder Software besitzt nur selten eine gute Usability – dieser Eindruck entsteht auch dadurch, dass schlecht bedienbare Webseiten und Programme schneller als hässlich wahrgenommen werden als gut bedienbare.

Wenn sich der Designer an das FFF-Credo („Form follows function“) hält, dann unterstützt das Design die Usability.

---

# Wieso gibt es Probleme, obwohl alle Standards befolgt werden?

Standards sind lediglich „Denkhilfen“, Guidelines, Orientierungen. Sie wecken Erwartungen bzw. entsprechen diesen nur. Explizite Standards wie Normen beschreiben die Funktion, implizite Standards wie das Corporate Design beziehen sich auf die Optik.

Jeder Fall ist individuell und von zahlreichen Faktoren abhängig:

- ◇ Ist der gewählte Standard überhaupt für die Aufgabe geeignet? Für Industrie und kommerzielle Anwendungen (z.B. Banken, Versicherungen) gelten andere Standards als für Büro, Heim und Entertainment (z.B. E-Mail, Textverarbeitung, Suchmaschinen).
- ◇ Passen Nutzermenge und -spektrum sowie Nutzungsfrequenz und -intensität zum gewählten Standard?
- ◇ Benötigen die Nutzer eine andere Motivation, als sie der Standard ermöglicht?
- ◇ Profitieren Erlernbarkeit und Schulungsaufwand vom Standard?
- ◇ Hilft der gewählte Standard, Fehler-Risiken zu mindern und die Sicherheit zu erhöhen? Für lebenskritische Systeme (z.B. Reaktorsteuerung, Flugsicherheit) sind teilweise Standardabweichungen nötig, um realweltliche Risiken zu vermeiden.
- ◇ Unterstützt der gewählte Standard die subjektive Zufriedenheit und das Vertrauen der Nutzer?
- ◇ Kann der Standard die Erwartungen in Effektivität, Effizienz und Performance erfüllen?
- ◇ Passt der Standard zum Preis des Produkts und den verfügbaren Entwicklungsressourcen?

## Wie erreiche ich eine gute Usability?

Indem von Anfang an mindestens eine Person, die sich mit deren Anforderungen und Umsetzung auskennt, in das Projekt einbezogen wird. Diese hat bei allen Usability-Aspekten die Entscheidungsbefugnis und kann in den anderen Feldern als Moderator fungieren. Eine offene, konstruktive und sachlich-kritikfähige Arbeitsumgebung lässt Usability fast von allein entstehen. Bedingung dafür ist ein fachlich buntes Team. Dieses umfasst mindestens Entwickler, Marketingmitarbeiter, Nutzer-Vertreter, Usability-Erfahrene.

Praktisch ist es von Vorteil, wenn die Beteiligten außerhalb ihrer fachlichen Kompetenz fordernde Hobbys betreiben, beispielsweise ein Instrument spielen, Modelleisenbahnen bauen, Autos reparieren oder kunstgewerbliche Stücke anfertigen. Ein breites Interessenspektrum erweitert den Kompetenzhorizont und befruchtet die Arbeit aufgrund des größeren Erfahrungsraumes.

## Wieso ist gute Usability nicht der Normalfall?

Analog lässt sich auch fragen: Warum essen wir ungesund, obwohl wir wissen, wie wichtig eine gesunde Ernährung ist? Wer Hunger oder Appetit hat, konsultiert nicht die Ernährungspyramide und Nährstofftabellen, sondern das umgebende Nahrungsangebot, seine Brieftasche und die Umstände.

Essen ist situativ und sinnlich. Die Portion Pommes, das Glas Cola, der Burger, das zerkochte Gemüse, das Weißbrot, die Extra-Portion Fleisch, der Fertigpudding, die süßen Corn Flakes – darauf habe ich gerade Appetit, die werden mich schon nicht umbringen, außerdem sind sie günstig, gerade in Reichweite und günstig.

Jeden Tag ein halbes Kilo frisches Obst und Gemüse vorzuhalten und zuzubereiten, ist viel zu aufwändig. Fast nur Wasser und ungesüßten Fruchttete trinken macht ebenfalls keinen Spaß. Gesundes Essen zeigt seine Effekte langfristig. Ungesundes Essen bringt uns nicht um, verringert nur unsere Chancen auf gute und beständige Gesundheit; der Zyniker stirbt sowieso nicht an Ungesundheit, sondern an einem Unfall oder einer neuen Krebsart.

Jeden Tag gesund zu essen erfordert genauso wie das Erreichen guter Usability:

- ◊ (anfangs) stetige Selbstdisziplin – bevor es zur Selbstverständlichkeit und Normalität wird
- ◊ Blick auf mittel- und langfristige statt kurzfristige Effekte – taugt allenfalls als rationales, aber nicht als sinnliches oder situatives Argument
- ◊ Umdeuten der Prioritätsverschiebung nicht als Verzicht, sondern als etwas Positives

Im Gegensatz zu ungesundem Essen äußern sich die Effekte guter Usability eher, unmittelbarer und besser erkennbar. Sie haben betriebswirtschaftliche Auswirkungen, während potenzielle Gesundheitsprobleme eine Hypothek für die eigene Zukunft und die gesellschaftlich finanzierten Gesundheitssysteme aufnehmen.

Wer den Wechsel zu bewusster und gesunder Ernährung geschafft hat, weiß um die Vorteile und das verbesserte Körpergefühl. Genauso wissen jene

---

Teams, die Usability fest in ihre Projekte integriert haben, um die positiven Auswirkungen. Alle anderen leben auch (irgendwie) und verzichten auf gute Chancen für ein besseres Leben oder erfolgreicherer Projektergebnis.

## Was ist die User Experience?

Die User Experience (abgekürzt UX) beschreibt das Gesamt-Nutzererlebnis. Dabei geraten auch Faktoren außerhalb der Software oder Webseite in die Betrachtung. Zur UX eines Webshops gehören beispielsweise auch das Verhalten bei einer telefonischen Support-Hotline, die Verpackung der Lieferung, der Umgang mit Retouren, die Präsentation in Katalogen, Broschüren oder Anzeigen. Bei Software gehören beispielsweise die Verpackung, Präsentation auf der Internetseite oder im Katalog, die Produktbeschreibung und Nutzerdokumentation, der Support, der Installations- und Aktualisierungsprozess zum Gesamterlebnis.

Ergänzend wirken sich die Leistungsfähigkeit des eigenen Geräts sowie die Internetgeschwindigkeit auf die UX aus. Bei der Konzeption sind daher die realen Verhältnisse zu berücksichtigen, die auf Nutzerseite herrschen. Insbesondere bei Mobilgeräten ist das technische Spektrum sehr breit, daher ist die Geräteverteilung unter den Nutzern bei der Entwicklung zu berücksichtigen.

Historisch hat sich die User Experience aus den Ansprüchen des Architekten und Designers Vitruv (1. Jh.v.u.Z.) entwickelt: Firmitat (Festigkeit), Utilitas (Nützlichkeit, Usability) und Venustas (Schönheit). Je nach Objekt (Gebäude, Kleidung, Software, etc.) fällt die Gewichtung der Aspekte unterschiedlich aus. Die ISO 9241-210 (Seite 320) definiert User Experience über die Wahrnehmungen und Reaktionen einer Person, die sich bei der Benutzung oder der erwarteten Verwendung eines Produkts ergeben. Das umfasst die psychologischen und physiologischen Aspekte, die Emotionen, die Erwartungen und das Verhalten.

Um die emotionale Wirkung zu steigern, ist es gelegentlich angebracht, mit Konventionen oder Regeln (und dadurch mit den Erwartungen) zu brechen. So lassen sich gezielt Akzente setzen oder Reibflächen erzeugen, die das Verhältnis zum Produkt intensivieren.

# Was unterscheidet Web-Usability von Software-Usability?

Die Grenzen zwischen Webseiten, Web-Applikationen und Software werden immer weicher. Der Hauptunterschied liegt nicht im Medium „Web oder Software“, sondern in der Nutzung und Funktionalität. Daher illustrieren die Beispiele in diesem Buch jeweils prototypische Anwendungsfälle und stellen keine Dogmen für Web oder Software dar. Tendenziell gilt folgende Orientierung:

Tab. 1.1: Unterschiede zwischen Webseiten und Software

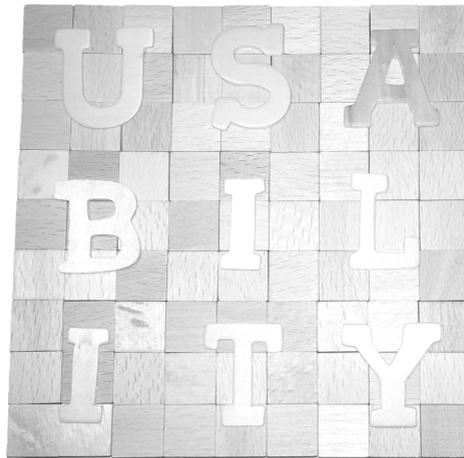
	Prototypische Webseite	Pragmatische Software
Nutzung	gelegentlich	intensiv, dauerhaft
Kontext	Freizeit	Arbeit
Motivation	eigene, selbstbestimmt	zielgerichtet, fremdbestimmt
Ziel	Unterhaltung, Information	Funktionalität, Aufgaben erledigen
Design	eigenständig, ein wenig verspielt	funktional, zurückhaltend
Farbigkeit	meist deutliche Farbigkeit	Farbigkeit nur zur Akzentuierung
Inhalte	Webseiten-Inhalte (vorwiegend Konsum)	Nutzerinhalte (vorwiegend Kreation, Bearbeitung)

Web-Apps wie Dokumentbearbeitung im Browser oder Webmailer-Oberflächen übertragen die Ansprüche von Software in das Web. Spiele oder kleine Softwaretools erfüllen teilweise mehr Web-Ansprüche. In vielen Fällen stellt sich bei neuer Software die Frage, ob sie als Web-App umgesetzt wird. Mobile Apps bilden oft optimierte Oberflächen für den Zugriff auf Webseiten-Funktionen. Daher gibt es kaum Regeln, die software- oder web-exklusiv sind. Es gelten stets die Regeln, die für die Nutzungssituation und Zielgruppe angemessen sind.

Jede Webseite und jede Software sollte in irgendeiner Form Nützlichkeit besitzen; diese kann sich in erfolgreicher Maschinensteuerung, guter Aufgabenbewältigung, effektiver sozialer Interaktion, aber auch in Spielspaß äußern. Diese Nützlichkeit spiegelt sich in der Präsentation wider, daher lohnt sich im ersten Schritt die Orientierung an pragmatischer Software. Im zweiten Schritt erhöhen soziale, ästhetische oder Gamification-Aspekte die Nutzfreude und steigern ergänzend die Motivation.

## 2 Einstimmung

*Das Ganze ist mehr als die Summe seiner Teil. (Aristoteles)*



Usability stellt den Nutzer in das Zentrum – eine Software oder Webseite dient dessen Zielen. Was heute wie ein Allgemeinplatz klingt, ist für viele Entwickler noch keine Selbstverständlichkeit. Projekte werden meist aus der technischen Perspektive vorangetrieben und dabei der Nutzen für die späteren Anwender aus dem Blick verloren.

Gute Usability befriedigt die fünf Grundbedürfnisse der Nutzer:

- ◇ Respekt
- ◇ Vertrauen
- ◇ Unterstützung
- ◇ Verständnis
- ◇ Kommunikation

Bei jedem Element, bei jeder Funktion, bei jeder Entscheidung über die Bedienung werden diese fünf Faktoren erfüllt, mitunter sind diese gegeneinander abzuwägen. Einige historische Beispiele illustrieren, wie unterschiedlich sich diese in der Praxis ausprägen.

Drei große Bereiche geraten in den Blick, wenn man über User-Interface-Design spricht:

**Projekt:** Alles ist heute ein Projekt, von der ersten Idee bis zum fertigen Ergebnis. Ein Projekt beginnt mit seiner Definition: Mit gegebenen Ressourcen (Zeit, Personal, Geld) soll zu einem bestimmten Zeitpunkt das Projektziel erreicht sein. Daraus lassen sich die Anforderungen an Projektmanagement ableiten:

- ◊ Definition des Projektziels in messbaren Einheiten nach validierbaren Kriterien
- ◊ Verwaltung der Ressourcen und Planung des Ressourceneinsatzes
- ◊ Prioritätensetzung bzgl. Teilzielen, Ressourcen, Anspruchsaspekten
- ◊ Koordination und Gewährleistung der Arbeitsabläufe
- ◊ Steuerung des Fortschritts und Kontrolle der Zwischenergebnisse

Einige Kurzausflüge in Bereiche des Projektmanagement zeigen, wie User-Interface-Design in der Praxis integriert werden kann. Im Sinne eines ganzheitlichen Ansatzes wäre die Abspaltung dieses Aspekts eine Degradierung. In allen Projektphasen und Teilprojekten sind die Anforderungen an das User-Interface-Design zu berücksichtigen.

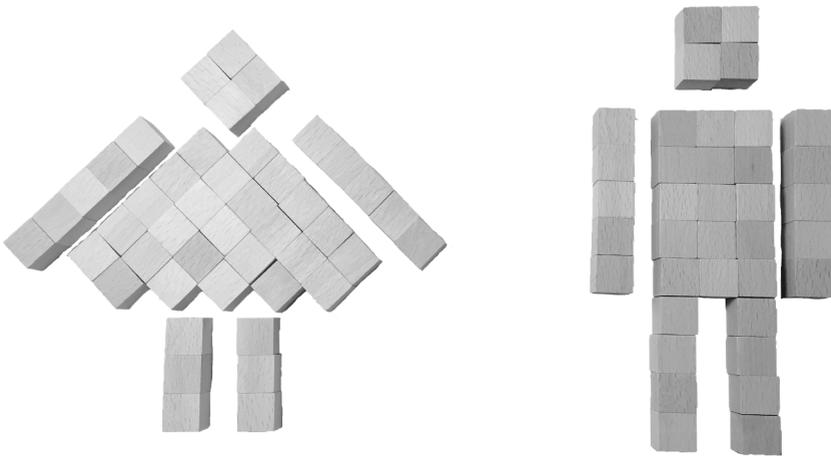
**Software:** Programme und Apps stellen Nutzern bestimmte Funktionalitäten oder Erlebnismöglichkeiten (beispielsweise Spiele) bereit. Die Software läuft auf dem Computer bzw. Smartphone oder Tablet des Nutzers und kommuniziert so direkt mit der Hardware. Die Definition einer Software umfasst mindestens:

- ◊ Systemische Voraussetzungen: Plattform bzw. Betriebssystem, nutzbare Hardware, Ein- und Ausgabemöglichkeiten
- ◊ Funktionale Anforderungen: Was tut die App oder das Programm? Wie werden Daten verarbeitet (Algorithmen)? Die funktionalen Anforderungen sind im Projektziel definiert.
- ◊ Nicht-funktionale Anforderungen: Gestaltung, Benutzerführung

Auch wenn gelegentlich Spiele zur Illustrierung eines Aspekts herangezogen werden, so richten sich die Ausführungen bewusst nicht an Spiele-Designer. Ziel ist vielmehr die Entwicklung von produktiver Software, die mit einem Nutzen beispielsweise in der Arbeit eingesetzt wird.

**Webseite:** Eine Webseite repräsentiert in gewisser Weise das Ergebnis einer Software, die auf dem Webserver läuft; die Eingaben und Ausgaben der Software

# 3 Unsere Nutzer



Um die Nutzer ins Zentrum aller Web- und Software-Projekte zu stellen, muss man sie erst einmal kennen. Dabei helfen etablierte Persönlichkeitstypen und situative Verhaltensmuster. Vor allem die sehr unterschiedlichen Bedürfnisse von Novizen und Experten stellen die Interface-Designer vor große Herausforderungen. Die einen beispielsweise vertrauen auf Tastenkürzel, die die anderen noch nicht einmal kennen.

Für die weitere Arbeit werden die Nutzer und ihre Anforderungen erfasst:

- ◇ Personas: prototypische Nutzer
- ◇ Akteure und Rollen: abstrakte Nutzer
- ◇ Use-Cases und Anwendungsfälle: (Teil-)Aufgaben
- ◇ User-Stories: die Nutzung aus Anwendersicht

Auch innerhalb jedes Projekts werden unterschiedlichen Rollen von verschiedenen Personen besetzt, von der Projektleitung über Design und Entwicklung bis zum Marketing. Diese stellen in einigen Software- und Web-Projekten ebenfalls zu berücksichtigende Anwender für sogenannte Backend-Prozesse und -Aufgaben dar.

Niemand muss Psychologie studieren, um zu wissen, dass Menschen sehr unterschiedlich sein können. Katharine Briggs und Isabel Myers haben den sogenannten Myers-Briggs-Typindikator etabliert, der die Persönlichkeit nach vier Gegensatzpaaren bestimmt:

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

Ausgehend davon lassen sich verschiedene Typen definieren. Je nach Veranlagung werden Aufgaben beispielsweise explorativ, kreativ, kollaborativ, sozio-technisch oder determiniert angegangen.

Für uns mag die Erkenntnis genügen, dass zwei völlig gegensätzliche Menschen gleichermaßen mit unserer Software oder Webseite etwas erreichen wollen. Wir könnten zwei unterschiedliche Schnittstellen entwickeln, die den jeweiligen Charakteristiken gut entsprechen – oder wir entwickeln eine Schnittstelle, die für alle Nutzer geeignet ist. Denn abgesehen von der allgemeinen charakterlichen Prägung unterliegen alle Menschen auch situativen Schwankungen. Ein geduldiger Mann kann in bestimmten Situationen sehr ungehalten reagieren, eine nervöse Frau kann in bestimmten Situationen sehr fokussiert und konzentriert arbeiten.

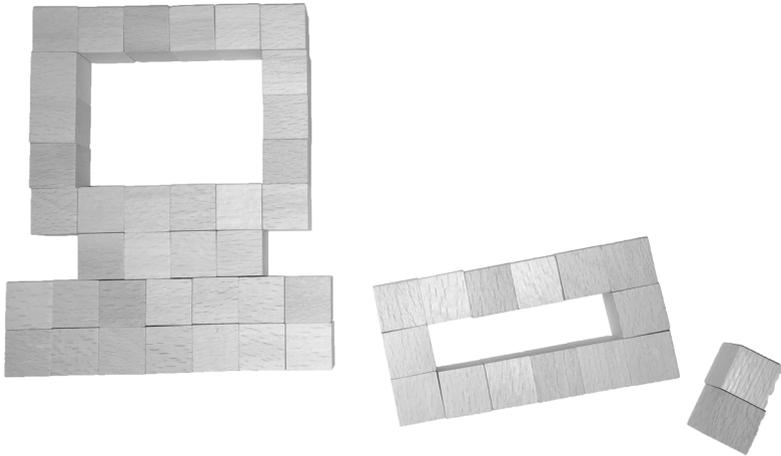
Somit stellt sich die Frage weniger, welche Persönlichkeit vor dem Computer sitzt, sondern ob das Interface der Aufgabe angemessen ist. Wenn es also gelingt, eine Schnittstelle so zu gestalten, dass ein geduldiger Mann seine Geduld und Ruhe behält und eine nervöse Frau ihre Nervosität ablegt, dann ist allen geholfen. Den Menschen ist bewusst, dass sie vor einer Maschine sitzen und nicht mit anderen Menschen interagieren. Insofern braucht man sich lediglich an die fünf Prinzipien der Einleitung zu halten.

Unterstützt man dabei die explorative, kreative, kollaborative, sozio-technische oder determinierte Nutzung, kann sich jeder Nutzer den Weg suchen, der am besten zu seiner Arbeitsweise passt.

Der **Explorative** wird die Webseite oder Software erkunden und sich seine Wege suchen. Also sind alle Funktionen korrekt beschriftet, und es gibt keine Sackgassen.

Der **Kreative** wird versuchen, die Grenzen der Software zu überwinden oder deren Möglichkeiten für neue Zwecke zu nutzen. Daher gestattet die Software alles, was keinen direkten Schaden anrichtet.

## 4 Unsere Computer



Usability entsteht im Wechselspiel aus Menschen und Computern. Wenn erfasst ist, was die eine der beiden Parteien antreibt, fehlt noch der Blick auf die andere Partei: Was ist das Wesen des Computers, wie kann er dem Nutzer überhaupt helfen? Indem man ihn als Werkzeug begreift, das von einer Menschengruppe zur Verwendung durch eine andere gestaltet wurde.

Der Blick in die Historie zeigt wichtige Meilensteine der Entwicklung der Mensch-Maschine-Schnittstelle. Dabei ist gut zu erleben, wie durch neue Interfaces die Nutzung von Computern und computerähnlichen Geräten immer mehr Lebensbereiche erfasste.

Margaret A. Boden formuliert in „Künstliche Intelligenz und Menschenbilder“ (1985) das grundsätzliche Dilemma:

*„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. Schließllich besteht Intelligenz zum großen Teil darin, vernünftige Entscheidungen ohne vollständige Evidenz zu treffen.“*

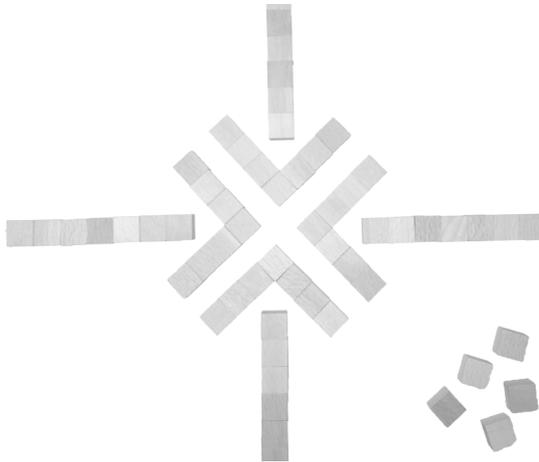
Menschen und Computer sind grundverschieden. Besonders deutlich wird dies, wenn man sich die Stärken beider Spezies vor Augen führt:

Tab. 4.1: Stärken

Menschen	Computer
Schwache Stimuli aufnehmen	Physikalische Größen zählen & messen
Stimuli vor verrauschtem Hintergrund erkennen	Signale außerhalb menschlicher Wahrnehmung aufnehmen
Muster in variierenden Situationen erkennen	Kodierte Informationen korrekt speichern
Unübliche, unerwartete Ereignisse aufnehmen	Vordefinierte Ereignisse überwachen
Prinzipien & Strategien erinnern	Schnelle, konsistente Antworten auf Eingangssignale
Wichtige Details erkennen, auch ohne Wissen über Kontext	Quantitative Größen & detaillierte Informationen akkurat abrufen
<i>Abstraktionsfähigkeit, Muster erkennen, Intuition, Entscheidungen mit wenig Datenmaterial bzw. Evidenz</i>	<i>selbe Eingabe + Algorithmus = immer die selbe Ausgabe; nicht ablenkbar</i>

Ein gutes Interface nutzt die Stärken der Menschen, kaschiert deren Schwächen und gewährt dem Computer die gleiche Behandlung: Stärken unterstützen und betonen sowie Schwächen ausgleichen oder abmindern. Es wäre unfair, von Menschen die exakte Uhrzeit, millisekundengenau, abzufragen, wenn der Computer sich diese von einem Uhrzeitserver besorgen kann. Ebenso wäre es – derzeit – von einem Computer zu viel verlangt, aus einer Vielzahl von Informationen die wichtigste herauszusuchen; allenfalls Vorschläge nach bestimmten Auswahlverfahren sind geeignet. Die Entscheidung, welche tatsächlich die wichtigste ist, trifft der Nutzer: Das ist seine Kompetenz, das bestärkt ihn

## 5 Ziele und Aktionen



Sind die Nutzer und Computer verstanden, gilt es nun, diese an der (eigenen) Schnittstelle zusammenzubringen. Um ein User-Interface zu konzipieren, gibt es verschiedene methodische Ansätze. Ausgehend von den User-Stories und Use-Cases wird ein Ansatz gewählt, der zur Aufgabe passt.

Da sich die vorgestellten Ansätze nicht gegenseitig ausschließen, ist es nützlich, alle zu kennen. So kann der beste gewählt werden, und Erkenntnisse aus den jeweils anderen Herangehensweisen verbessern die Aussagekraft des Konzepts. Oft bietet es sich an, einzelne Bereiche einer Software oder Webseite nach einer anderen Methodik oder zusätzlich in einer zweiten Konzeptualisierung zu erfassen.

# Die Sieben Stufen der Aktion

Der Usability-Berater Donald Norman hat 1988 in „The Design of Everyday Things“ sieben Stufen definiert, die jeder Nutzer durchläuft:

1. Formulierung des Ziels
2. Formulierung des Plans
3. Spezifikation der Aktion
4. Ausführung der Aktion
5. Wahrnehmung des Systemstatus
6. Interpretation des Systemstatus
7. Auswertung des Resultats

## Praxisbeispiel: Einkauf im Webshop

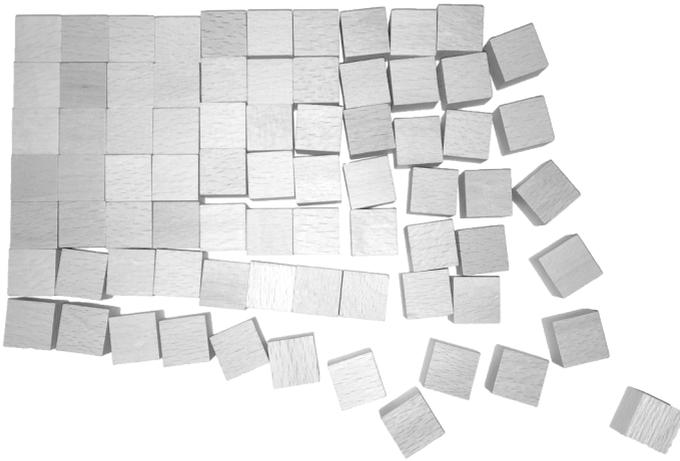
**Erstens: Mein Ziel.** Ich möchte ein neues Fahrrad haben.

**Zweitens: Mein Plan.** Ich werde mir ein neues Fahrrad kaufen. (Man beachte den Unterschied zwischen „haben“ und „kaufen“ – ersteres interessiert den Nutzer tatsächlich, zweites ist nur Mittel zum Zweck, er würde es auch „haben“, wenn er es nicht kaufen würde, sondern geschenkt bekäme, irgendwo stähle oder bei einer Tombola gewänne.)

**Drittens: Meine Aktionen.** Ich definiere die Kriterien, die für mich entscheidend sein sollen (Pflicht-, Kür-, Ablehnungskriterien, auch der Preisrahmen ist ein Kriterium). Ich sichte das Sortiment und wähle mehrere Fahrräder aus, zwischen denen ich mich im nächsten Schritt entscheide. Sollte ich keine finden, nutze ich die Kenntnisse aus der Sortimentsrecherche und überarbeite meine Kriterien und wähle anschließend neu aus. Anschließend kaufe ich das entsprechende Fahrrad.

**Viertens: Die Ausführung.** In Online-Shops suche ich Fahrräder, die zu meinen Kriterien passen, dabei helfen mir die Suche, die Shop-Kategorien sowie Filter für den Preis und bestimmte Funktionen. Innerhalb eines Webshops hilft mir ein Merk- oder Wunschzettel, alle Kandidaten zusammenzutragen. Anschließend beginnt der Detailvergleich, manche Fahrräder fallen aufgrund weicher Kriterien – beispielsweise falsche Farbe, schlechtes Foto, fehlende Angaben über Ausstattung oder Funktionen, unsympathischer Webshop, zu hoher oder zu niedriger Preis – schnell wieder aus der Selektion. Die verbleibenden prüfe ich hinsichtlich der Kriterien und ob sie mir gefallen. Eventuell berate ich mich auch mit einem Bekannten, hole in Foren Informationen und Erfahrungsberichte über ein Modell ein, oder breche die Suche erst einmal ab und schaue mich lieber bei

## 6 Ordnung auf dem Bildschirm



Die bisher vorgestellten konzeptuellen Ansätze beschreiben vorwiegend, wie in der Planung die Anforderungen geeignet erfasst werden. Widmet man sich nun der konkreten Nutzerschnittstelle, landet man auf dem zweidimensionalen Bildschirm als Informations- und Interaktionsfläche für den Nutzer. Die Grenzen können durch verschiedene Darstellungsarten (Single View, Scrolling, Zooming) überwunden werden. Entscheidend ist vor allem, dass alle benötigten Informationen und Funktionen bereitgestellt werden.

Um diese geeignet zu präsentieren, hilft das Verständnis über die menschliche Blickführung und Fokussierung. Daraus ergeben sich fünf Parameter, um die Elemente auf dem Bildschirm anzuordnen:

- ◇ Ausrichtung
- ◇ Dominanz
- ◇ Hierarchien
- ◇ Weißraum
- ◇ Visuelle Balance

In den meisten Fällen wird ein Interface über einen Bildschirm bedient. Sprachsteuerung und andere Eingabemethoden folgen anderen Regeln, aber die zugrunde liegenden Prinzipien sind gleich:

- ◊ begrenzte Ein- und Ausgaberesourcen:
  - **eindimensional**: Schallwelle (Töne und gesprochene Worte) oder Textfluss  $\Rightarrow$  Linearität
  - **zweidimensional**: Fläche (Papier, Monitor)  $\Rightarrow$  Rändergrenzen
  - **dreidimensional**: Raum (Hologramm oder 3D-Simulation)  $\Rightarrow$  Horizont
  - **vierdimensional**: Raum mit Zeit  $\Rightarrow$  Erstellungsaufwand
- ◊ klare Informationspräsentation und Nutzerführung
- ◊ funktionale, strukturelle und optische Strukturierung
- ◊ Trennung zwischen Wichtigem und weniger Wichtigem
- ◊ Erkennbarkeit der Bedienelemente und deren Bedienung
- ◊ unmittelbare Reaktion auf Nutzeraktion

Ein Sprach-Interface eignet sich vorwiegend für einfache Aufgaben ohne wechselnden Fokus. Dabei werden Ein- und Ausgaben linear angeordnet, Hierarchien oder Strukturen der Daten müssen ebenfalls linearisiert werden, das Vergleichen von zwei Informationen ist schwer möglich. Im dreidimensionalen Raum steigt die – im Vergleich zur Fläche – für den Nutzer zu verarbeitende Informationsdichte erheblich. Für eine effiziente Bedienung müssen relevante Informationen besonders betont und Bedienelemente noch deutlicher hervorgehoben werden. Geeignete Eingabegeräte sind nötig, um ein flüssiges Bedienen im Raum zu ermöglichen.

Die hauptsächliche Nutzung von Computer-Software oder Webseiten findet – derzeit noch – in der zweidimensionalen Fläche auf dem Monitor statt. Die dritte Dimension wird allenfalls metaphorisch simuliert, wenn sich Fenster hintereinander befinden oder Schatten werfen. Als Eingabegerät stehen eine Maus für die Navigation in der Fläche und die Tastatur für die lineare Eingabe von Zeichen zur Verfügung. In zahlreichen Anwendungsfällen wird die Maus inzwischen von einem Touch-Interface abgelöst oder ergänzt; die Nutzer agieren also nicht mehr indirekt über eigenständige Eingabegeräte, sondern das Ausgabemedium dient gleichzeitig der Eingabe.

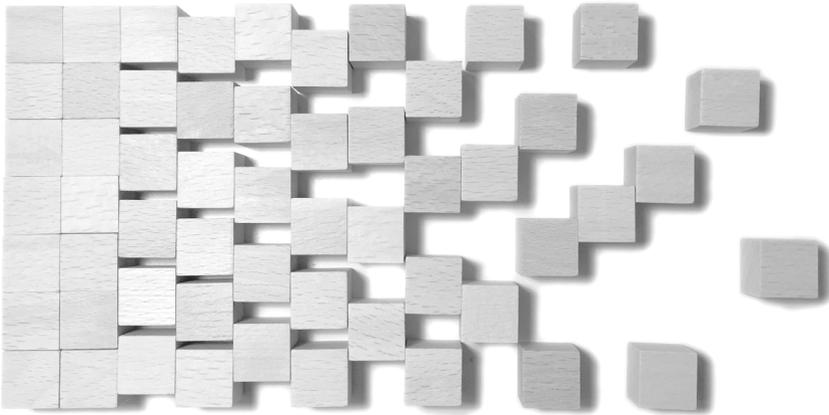
## Die Grenzen des Bildschirms

Für Software und Webseiten gilt die gleiche Einschränkung: Was nicht auf dem Monitor sichtbar ist, kann der Nutzer nicht wahrnehmen. Entweder erfährt er

# 7 Design-Nussschale

*„Je weniger einer braucht, desto mehr nähert er sich den Göttern, die gar nichts brauchen.“ (Sokrates)*

*„Vollkommenheit entsteht offensichtlich nicht dann, wenn man nichts mehr hinzuzufügen hat, sondern wenn man nichts mehr wegnehmen kann.“ (Antoine de Saint-Exupéry)*



Um die abstrakten Grundlagen in eine tatsächliche Gestaltung zu überführen, wird das Design benötigt. Die kurze Vorstellung macht mit den wichtigsten visuellen Parametern vertraut und zeigt deren Einsatz und Wirkung für den Alltag:

- ◇ Position und Größe sowie Aufteilung, Gitter und Raster
- ◇ Farbe, Helligkeit und Texturen
- ◇ Schrift

Hinweise zu Icons und Symbolen sowie zur Berücksichtigung der Zielmonitore, beispielsweise als Responsive Design, führen in die komplexe Praxis. Das Plädoyer, lieber InDesign als Photoshop für Webseiten-Entwürfe zu verwenden, beschließt den Ausflug ins Design.

Mehrere Prinzipien leiten den Designer bei der Gestaltung einer Software-Oberfläche oder einer Webseite:

- ◊ Bescheidenheit, Mäßigung beim Einsatz der Mittel
- ◊ Verständlichkeit der vermittelten Inhalte, „Form follows function“
- ◊ Beschränkung auf die nötigen Mittel, „Weniger ist mehr“
- ◊ Grafische Strukturen leiten interaktive Strukturen
- ◊ Design und Gestaltung sind keine Show, sondern Dienstleistung!

## Visuelle Variablen

In der konkreten Umsetzung stehen sieben Basis-Möglichkeiten zur Verfügung, Elemente voneinander zu unterscheiden:

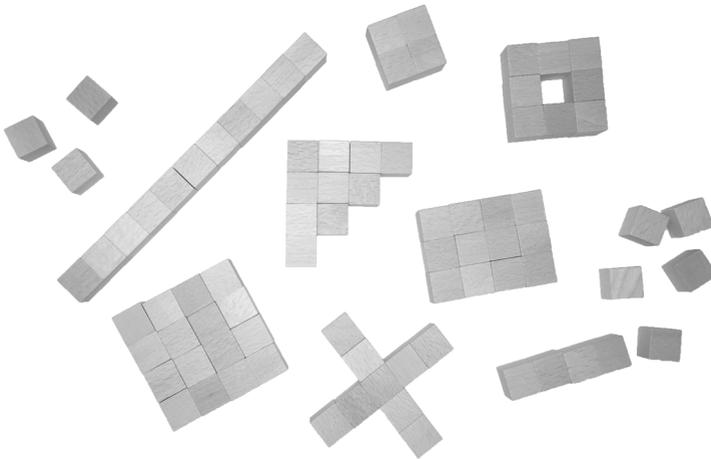
- ◊ Position
- ◊ Fläche, Größe
- ◊ Helligkeit
- ◊ Farbe
- ◊ Textur
- ◊ Neigung
- ◊ Form, Gestalt

Ergänzend können Länge, Farbton, Winkel, Enthaltung, Volumen, Sättigung, Verbindung, Bewegung und Blinken genutzt werden. Insbesondere die letzten beiden Möglichkeiten sind nur mit Vorsicht einzusetzen, denn der Blick wird automatisch auf diese Elemente gelenkt. Bewegung und Blinken wirken stärker aufmerksamkeitsanziehend als alle anderen Gestaltungsmöglichkeiten.

Psychologisch sind Menschen konditioniert, Bewegung als relevant wahrzunehmen. Die nächst höhere Aufmerksamkeitsanziehung besitzen Gesichter, insbesondere Augen, das machen sich Werbeanzeigen zunutze und ziehen durch Blinken, Bewegung oder (attraktive) Gesichter den Blick auf sich. Die menschliche Fähigkeit, Gesichter zu erkennen, beschränkt sich jedoch nicht auf Fotos.

Gesichtsähnliche Strukturen können auf verschiedene Weise evoziert werden. Eine rein pragmatische Anordnung von Bedienelementen kann menschliche Assoziationen hervorrufen, die die funktionale Bedeutung überlagern. Insbesondere Webseiten mit weichen Formen (runde Ecken, sanfte Farbverläufe, sehr harmonische Größenverhältnisse) verschleiern so ihre Technizität und provozieren beim Nutzer eine emotionale Beziehung. Software ist dagegen meist funktionsmächtiger und hat weniger optische Gelegenheiten für humanoide Anspielungen.

## 8 Standardelemente



Für die häufigsten Interaktionen existieren Elemente wie Buttons, Menüs, oder Listen. Für Eingaben gibt es Eingabefelder für alle Sorten von Zeichen, Checkboxes und Radio-Buttons ermöglichen die einfache Auswahl von Optionen. An diese Standardelemente sind die Nutzer gewöhnt. Fehlbedienungen geschehen meist nur aus inhaltlichen Missverständnissen und nicht, weil der Nutzer ein Bedienelement nicht versteht. Daher muss der Interface-Designer die Standardelemente gut kennen und jeweils korrekt einsetzen.

Die wichtigsten werden ausführlich vorgestellt, dabei wird die funktionale Bedeutung erläutert und auf besondere Anwendungsfälle verwiesen. Wo geeignet wird die Verwendung in Webseiten mit Code-Beispielen illustriert.

Ergänzend wird die Standard-Anforderung an die Suchfunktion in Webseiten besprochen und mit Beispielen illustriert. Im strengen Sinn ist sie kein Standardelement, aber als Standardfunktion verdient sie eine Betrachtung.

Jede Webseite und jede Software arbeitet mit den selben Elementen. Diese dienen dem Erfassen von Eingaben oder dem Anzeigen von Ausgaben.

Tab. 8.1: Die Standardelemente

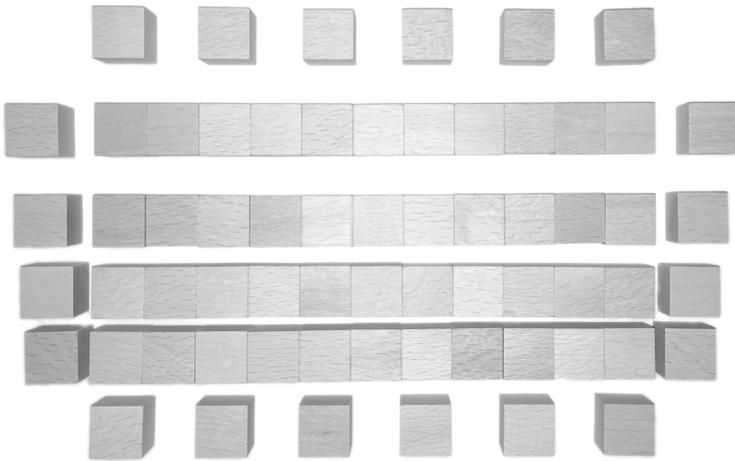
Element	Funktion	Häufigste Probleme
Eingabefeld	Dateneingabe	Fehleingaben
Check-Boxen	X von Y	Beschriftung
Radio-Buttons	1 von X	Beschriftung
Auswahlmenü	1 von X	zu lange Liste
Auswahlliste	X von Y	Mehrfachauswahl
Combo-Box	1 von X + Dateneingabe	Fehleingabe
Slider, Regler	1 von X	präzise Eingabe
Buttons	Aktion	Beschriftung
Tabulator/„Tabs“	gleichartige Bereiche	Trennung
Gruppierung	optische Zusammengehörigkeit	Grenzen übersehen
Ausklappbereiche	(erweiterte) Funktionen platzsparend anbieten	Bedienelement erkennen
Funktionsmenü	(alle) Funktionen bereitstellen	Sortierung
Ribbon	(alle) Funktionen bereitstellen	Funktion finden
Symbolleiste, Palette	Funktionen bereitstellen	Fehldeutung
Hinweisfenster	Information	Verständlichkeit
Dialogfenster	Datenabfrage, Assistent	Verständlichkeit
Statusanzeige	Information	Fehldeutung
Tabellen, Übersichten	Information	Übersichtlichkeit

In Programmen wird idealerweise der Standard des Systems verwendet. Die User Interface Guidelines (Seite 313) helfen beim Einsatz der Standardbedienelemente. Sie illustrieren geeignete und ungeeignete Anwendungen und stellen die technische Integration mit den möglichen Parametern vor.

Auf Webseiten ist die Gestaltungsvielfalt größer, und es bietet sich an, die Elemente mittels CSS anders zu gestalten und beispielsweise an das Corporate Design anzupassen – dabei müssen sie aber weiterhin erkennbar, unterscheidbar und in ihrer Funktion korrekt bleiben.

## 9 Das eigene Interface

*„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)*



Wer für seine eigene Webseite oder Software eine gute Usability anstrebt, findet sich in der Praxis in einem von zwei Extremen wieder. Entweder wird nach dem Start-up-Modell gearbeitet, das zwar schnell funktionierende Ergebnisse liefert, aber Usability weniger als Aufgabe begreift. Oder alle direkt und indirekt Beteiligten werden in die Anforderungserfassung integriert und dabei ein Qualitätsanspruch verfolgt. Im zweiten Fall sind die Aspekte guter Usability quasi institutionalisiert.

Ergänzend zu den eigenen Ansprüchen schreiben zahlreiche Normen vor, wie gute Usability zu erreichen oder zu bestimmen ist. Diese Vorschriften bieten – auch wenn sie für das eigene Projekt nicht zwingend gelten – wertvolle Inspiration und Handlungsempfehlungen. Außerdem sind die Ansprüche des Corporate Design und der Corporate Identity des Unternehmens zu berücksichtigen.

Abschließend werden die wichtigsten Verfahren vorgestellt, um die Usability von Webseiten oder Software zu testen bzw. zu evaluieren.

Egal ob Software oder Webseite, ob ausgefallenes High-End-Programm oder einfacher Webshop, ob Single-Purpose-Tool oder interaktive Web-Lösung – es kommt der Zeitpunkt, an dem Nutzer damit arbeiten.

- ◊ Nutzer wollen Probleme **lösen**.
- ◊ Nutzer wollen Aufgaben **erledigen**.
- ◊ Nutzer wollen dabei in geeigneter Weise **unterstützt werden**.
  
- ◊ Nutzer wollen **keine** Dokumentationen wälzen.
- ◊ Nutzer wollen **nicht** ständig bei den Entwicklern oder einer Hotline nachfragen, was sie wie tun dürfen.
- ◊ Nutzer wollen **nicht** mehr Arbeit haben als vorher.

Usability ist ein gemeinschaftlicher Prozess, bei dem viele verschiedene Menschen mit unterschiedlichen Fachkenntnissen eine gemeinsame Lösung entwickeln. Der diktatorische Ansatz liefert nie annähernd gleichwertige Ergebnisse. Diktatorische Entscheidungen kommen meist zum Einsatz, wenn gefühlt die Erarbeitung der Lösung sich ständig verzögert und keine brauchbare Lösung entstehen will (vorzugsweise bei Projekten, deren Entscheidungen diktatorisch vorge setzt wurden). Statt eines kompletten Moduswechsel vom gemeinschaftlichen Erarbeiten zur Diktatur ist es effektiver, sich auf geeignete Rahmenbedingungen für die kollaborativen Prozesse zu verständigen.

Dafür stehen zwei grundsätzliche Ansätze zur Verfügung:

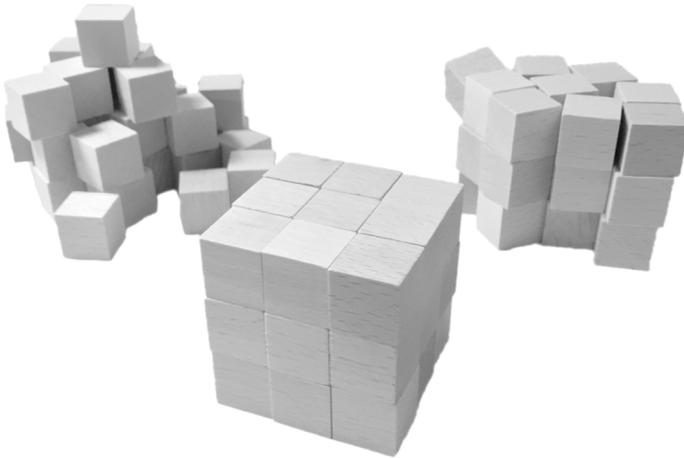
**Start-up-Modell:** Spontanität, Kreativität, Freude, „Hauptsache es läuft – egal wie.“

**Qualitätsanspruch:** Konzept, Planung, Struktur, „Es muss alle Ansprüche erfüllen und langfristig, robust, stabil laufen.“

Diese zwei Pole markieren das Spektrum. In vielen Fällen sind Projekte zwischen den Extremen angesiedelt und versuchen die Kombination beider Welten – und scheitern dabei oder laufen aus dem geplanten Rahmen. Erfolgversprechender ist der Ansatz, sich für eines der Extreme zu entscheiden und als Team gelegentliche Ausflüge in das andere Terrain zu unternehmen:

- ◊ Im Start-up-Modell wird alle zwei Wochen ein „Theorietag“ eingelegt, in dem sich das Team zusammensetzt und nicht mit Code beschäftigt. Stattdessen verständigen sich die Beteiligten auf geeignete Standards, Herangehensweisen und die konzeptionelle Basis. Wenn die Team-Dynamik es zulässt, können zwei bis drei externe Personen an diesen Tagen beteiligt werden. Das Team zoomt aus den konkreten Aufgaben heraus und verschafft sich einen Überblick über das entstehende Ganze.
- ◊ Das Start-up-Modell agiert iterativ. Es werden Zwischenstufen vereinbart, die von Außenstehenden abgenommen werden. Dadurch wird regelmäßig

# 10 Ganzheitlicher Ansatz



Anhand des Würfelmodells werden die 27 wichtigsten Aspekte für gute Usability auf dem abstrakten Level vorgestellt. Die praktische Umsetzung und Marktrealitäten sind dabei ebenso zu berücksichtigen wie idealistische Visionen und verfügbare Ressourcen. Die 27 Aspekte verdichten bereits Gesagtes und weisen auf weitere Bereiche hin, die für jedes Projekt gesondert zu vertiefen sind.

Usability benötigt einen ganzheitlichen Ansatz; in einem Bereich super zu sein und andere zu vernachlässigen führt niemals zu einem guten Nutzer-Erlebnis. Diese Ganzheitlichkeit lässt sich gut vor Augen führen, indem man Usability als kompakten Würfel auffasst, der drei Dimensionen repräsentiert:

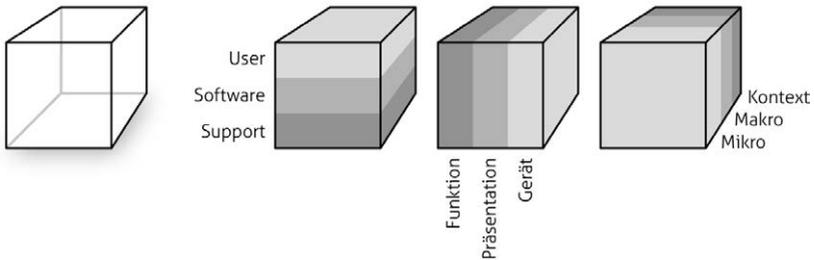


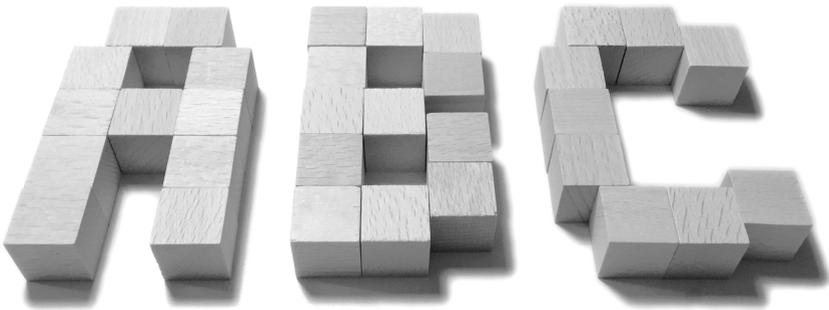
Abb. 10.1: Usability funktioniert nur als großes Ganzes. Die drei Dimensionen verdeutlichen relevante Aspekte.

- ◇ **User:** Nutzer und deren Bedürfnisse
- ◇ **Software:** Programme, Webseiten, Web-Applikationen
- ◇ **Support:** jede Art von (indirekter) Unterstützung
- ◇ **Funktion:** was die Software leistet bzw. welche Funktionalitäten eine Webseite anbietet
- ◇ **Präsentation:** wie die Software oder Webseite ihre Funktionen dem Nutzer anbietet
- ◇ **Gerät:** in welchem (technischen) Kontext die Software funktioniert bzw. die Webseite genutzt wird
- ◇ **Mikro:** konkrete Software, Software-Einheit oder Webseite bzw. Webseienelement
- ◇ **Makro:** Gruppe aller gleichartigen Software- bzw. Web-Angebote
- ◇ **Kontext:** die Welt außerhalb der fokussierten Software bzw. Webseite

Oft steht bei Usability die Beziehung zwischen Nutzer und Software im Vordergrund. Diese zwei Schichten bilden ein Usability-Sandwich, gut für den Hunger zwischendurch, aber keine vollwertige Mahlzeit. Dazu benötigt es noch die nachgelagerten Prozesse, die hier unter Support subsumiert werden.

Die Verbindung aus Funktionalität und Präsentation ist in allen denkbaren Zusammenhängen beliebt, als „Inhalt und Form“, „Ursache und Effekt“ oder „Sein und Schein“. Daneben bestimmt die konkrete Umgebung in Form der Hardware, welche Funktionen möglich sind und welche Regeln für die Präsentation gelten.

# 11 Glossar & Index



Im Folgenden werden die wichtigsten Begriffe kurz erklärt. Die Zahlen in eckigen Klammern verweisen auf die entsprechenden Seiten im Buch.

- Akteur:** Nutzerrolle in einem bestimmten Kontext einer Software oder Webseite, beispielsweise Bearbeiter, Datenerfasser oder Besucher, Kunde. Hilft beim Erfassen der Anforderungen und beim Verfassen der Use-Cases und User-Storys. [56]
- Aktion:** Aus Sicht des User-Interface jede Möglichkeit des Nutzers, auf den Computer einzuwirken, z.B. durch Maus-, Tastatur- oder Spracheingabe. Aus Entwicklersicht jeder in sich abgeschlossene Bearbeitungsschritt der Datenverarbeitung. Aus vielen Einzelaktionen (z.B. Daten lesen, verarbeiten, an ein Gerät ausgeben, Bestätigung der korrekten Übertragung) können komplexe Aktionen bzw. Funktionen (z.B. Datei drucken) zusammengefügt werden.
- Alpha-Release:** Erste nutzbare Version einer Software, meist nur für einen begrenzten (internen) Nutzerkreis. Dient vorrangig dem Finden von Fehlern.
- Android:** Betriebssystem für Mobilgeräte (Smartphones, Tablets), wird von Google entwickelt und Hardware-Herstellern (kostenlos) zur Verfügung gestellt, die dafür Lizenzgebühren an patenthaltende Unternehmen wie Microsoft (und Nokia) entrichten. Die grundsätzliche Bedienweise entspricht der von iOS.
- ARM:** Advanced RISC Machines Ltd., ein britischer Chiphersteller. ARM bezeichnet auch eine Musterarchitektur (Advanced RISC Maschine), die ihre Stärken in Ausführungsgeschwindigkeit und Energieeffizienz hat und bevorzugt in Mobilgeräten (Smartphones, Tablets) verwendet wird.
- Assistent:** Unterstützung des Nutzers durch „automatisches“ Generieren von Ergebnissen anhand der Eingaben. Etabliert sind die Assistenten, in denen der Nutzer Schritt für Schritt Angaben macht, mit „Weiter“ zum nächsten Eingabebereich wechselt und so Einstellungen an seinem Computer vornimmt oder eine Office-Programm-Vorlage nutzt. Assistenten gewährleisten, dass alle relevanten Daten vorliegen, auch bieten diese die Möglichkeit, den Kontext für Eingaben oder Daten darzustellen, der an anderer Stelle oft fehlt. Der Assistent orientiert sich bei seiner Abfrage an den Schritten, die der Nutzer für das Erreichen seines Zieles als geeignet ansieht. Assistenten werden als modale Dialoge umgesetzt, müssen meist vom Nutzer aktiv aufgerufen, können abgebrochen und nach Durchlaufen abgeschlossen werden. Daneben unterstützen Assistenz-Programme (wie Siri, Cortana oder Google Now) allgemein den Nutzer, indem sie beispielsweise aus einer Spracheingabe einen Befehl erkennen und diesen umsetzen, z.B. eine Nachricht versenden, einen Anruf auslösen oder einen Termin im Kalender eintragen. Auch die Office- Assistenten, die sich

---

aus Microsoft Bob heraus entwickelt haben, arbeiten eher allgemein und bieten dem Nutzer geeignet scheinende Aktionen an. [334]

**Auswahlliste:** Bedienelement, um mindestens eine Option aus einer Reihe von gleichartigen Alternativen auszuwählen. Darstellung erfolgt als Liste, in der ggf. gescrollt werden kann, die alle verfügbaren Optionen enthält. Die Auswahl erfolgt über Anklicken eines Eintrags, bei Mehrfachauswahl wird die `[strg]` - bzw. `[cmd]` -Taste beim Anklicken gedrückt. Die gewählten Optionen werden in der Liste markiert. [260]

**Auswahlmenü:** Bedienelement, um (genau) eine Option aus einer Reihe von gleichartigen Alternativen auszuwählen. Das Menü zeigt die gewählte Option an, beim Anklicken wird die Liste aller verfügbaren Optionen angezeigt (z.B. die Wahl eines Landes, einer Anrede). [256]

**Auto-Vervollständigung:** Unterstützende Funktion bei der Eingabe in Formularen oder auf der Kommandozeile. Anhand der eingegebenen Zeichen werden geeignete Vorschläge eingeblendet (z.B. „Berlin“ bei „ber“ oder „Mit freundlichen Grüßen“ bei „mfg“), dies kann automatisch oder durch Drücken einer speziellen Taste erfolgen. In iOS und MacOS ist die Auto-Vervollständigung mit der Auto-Korrektur kombiniert und steht systemweit an allen Stellen mit Texteingabe zur Verfügung, und die Ersetzung erfolgt automatisch. Bekannte Vertreter sind die Auto-Vervollständigung in der Browser-Adresszeile und in der Google-Eingabemaske.

**Beta-Release:** Erste Version einer Software, die auch externen Nutzern zur Verfügung gestellt wird. Dient vorwiegend dem Entdecken von Fehlern. Sollte nicht in Produktivumgebungen verwendet werden, kann aber die Nutzer auf die fertige Software einstimmen und helfen, Probleme im Alltag zu entdecken.

**Box-Modell:** Mit HTML lassen sich Blöcke bilden; meist durch `<div>`-Container, aber viele HTML-Elemente können durch CSS-Anweisungen als Block fungieren. Diese Blöcke werden mittels CSS in der Webseite und zueinander angeordnet. Bis vor einigen Jahren wichen die Box-Modelle der Browser voneinander ab, rechneten beispielsweise einen Innenabstand (Padding) mal der Boxgröße hinzu oder zogen diesen von der definierten Größe ab. Dadurch entstanden unterschiedliche Darstellungen in den verschiedenen Browsern. Seit etwa 2010 verwenden alle Web-Browser das selbe Box-Modell mit Innen- und Außenabständen.

**Button:** (auch „Schaltfläche“) Bedienelement, das bei Anklicken eine Aktion auslöst. Diese Aktion ist durch Beschriftung oder Icon eindeutig auf dem Button angegeben. Üblich sind „Ok“- oder „Abbrechen“-Buttons zur Bestätigung bzw. Abbruch einer zuvor angeforderten Aktion. Auch „Weiter“-

„Zurück“-, „Drucken“- und ähnliche Schaltflächen sind gebräuchlich. Entweder wirken Buttons durch ihr Design wie drückbare Tasten oder sind aufgrund ihrer Platzierung, Beschriftung, Gestaltung als anklickbare Elemente erkennbar. Im Gegensatz zu Links haben sie entweder (z.T. indirekt) bestätigenden, ablehnenden oder funktionalen Charakter. [265]

**Checkbox:** Bedienelement mit zwei Zuständen ( und  bzw. ). Bei allein auftretenden Checkboxes ist das Setzen des Häkchens die Bestätigung der Beschriftung (z.B. „Newsletter abonnieren“, „Helligkeit automatisch anpassen“, „doppelseitig drucken“); das Nicht-Setzen des Häkchens entspricht der Ablehnung der Beschriftung. Sind mehrere Checkboxes miteinander kombiniert (z.B. für verschiedene Schrift-Formate wie fett, kursiv, hochgestellt), so kann jedes einzeln gesetzt werden, oder auch alle. Das Setzen aktiviert die entsprechende Option. [251]

**Checkout-Prozess:** (auch „Bestellvorgang“ oder „Bestelltunnel“) Bezeichnet die Schritte in einem Webshop zwischen Warenkorb-Ansicht und „Kaufen“-Button („zahlungspflichtig bestellen“). Dabei werden Rechnungs- und Lieferadresse, Zahlungsweise und ggf. Bestelloptionen eingetragen oder aus dem Kundenkonto übernommen (wenn vorhanden).

**Codec:** Kunstwort aus Coder und Decoder. Verfahren, um Medieninhalte komprimiert abzuspeichern und wiederzugeben. Für Bildinhalte ist jpg der Quasi-Standard, bei Musik MP3 und AAC, bei Videos mpg und H.264 (oft in Dateien mit der Endung .mp4). Diese Verfahren reduzieren die Dateigröße nach psychologischen Aspekten, entfernen also Inhaltsinformationen, die dem Gehirn kaum auffallen bzw. die von diesem leicht wieder ergänzt werden können. Dabei wird häufig eine Reduktion um den Faktor 10 erreicht.

**Combo-Box:** Bedienelement, das Auswahlmenü mit Eingabefeld kombiniert. Beim Anklicken werden die verfügbaren Optionen angezeigt und können ausgewählt werden. Alternativ kann in das Feld eine Option direkt eingetragen werden, z.B. für die Auswahl einer Schriftgröße. Mit Vorsicht einzusetzen, da manuelle Eingaben fehlerhaft sein können. Ist heute meist durch andere Eingabemöglichkeiten ersetzt, z.B. Eingabefelder mit Auto-Vervollständigung. [261]

**Contextual Inquiry:** Usertest, indem dem Tester Fragen gestellt werden. Der Tester befindet sich dabei in der Nutzungssituation, die Befragung erfolgt parallel zur Nutzung oder direkt im Anschluss.

**CSS:** Abkürzung für Cascading Style Sheets. Bei Webseiten wird zwischen Inhalt und Form getrennt. Die HTML-Datei enthält den Inhalt, die CSS-Datei steuert das Aussehen der in HTML verwendeten Elemente. Durch

- 
- Unterschiede bei der Interpretation des „Box-Modells“ war umfangreiches Testen mit den verschiedenen Browsern nötig. Heute interpretieren alle Browser den Standard CSS2 nahezu identisch. Mit CSS3 sind in modernen Browsern die flexible Gestaltung mit Farbverläufen, Schatten und Animationen möglich.
- Cut & Paste:** Möglichkeit, einen Inhaltsbereich zu entfernen und an anderer Stelle einzufügen. Das gilt z.B. für Textabschnitte, Bildbereiche oder Dateien. Statt entfernen, kann der gewählte Bereich auch kopiert und durch das (mehrfache) Einfügen (an anderen Stellen) vervielfältigt werden.
- Default-Wert:** Standard- oder empfohlener oder wahrscheinlich genutzter Wert, der voreingestellt ist. Nutzer kann ihn bei Bedarf durch eigenen Eintrag überschreiben. Auch Checkboxes, Radio-Buttons oder Auswahlmensüs können Default-Einstellungen aufweisen – das nimmt dem Nutzer Entscheidungen ab bzw. präsentiert eine plausible Auswahl zur Orientierung. In psychologischen Kontexten bezeichnet der „Default-Wert“ die übliche Wahrnehmung, den „unmarkierten Zustand“ bzw. das gängige Interpretationsmuster, das auf etwas angewendet wird.
- Design:** Konkrete optische Gestaltung eines funktionalen oder Inhaltsbereichs in einer Software oder Webseite. Das Design kann eine eigene „Sprache“ beinhalten, die das Aussehen und die Funktionalität von Elementen definiert. Dazu gehören u.a. Farben, Schriftarten, optische Effekte, Stil von Icons und Abbildungen, Linien und geometrische Figuren.
- Dialog:** Kleines Fenster, das mindestens eine bestätigende Schaltfläche besitzt und die Interaktion mit dem eigentlichen Interface unterbricht. In einem Dialogfeld werden Hinweise gegeben, Bestätigungen eingefordert, Daten abgefragt oder Assistenten integriert.[284]
- Direkte Manipulation:** Bedienparadigma, das darauf basiert, dass durch ein Hilfsmittel (wie Maus oder Touchpad) der Bildschirminhalt direkt beeinflusst werden kann. Auf Touch-Geräten (wie Smartphones oder Tablets) erfolgt die Bedienung meist durch Fingerberührung. Technische Bedingung ist, dass die ausgeführte Aktion in „Echtzeit“ – ohne wahrnehmbare Verzögerung – ausgeführt wird; z.B. wirkt das Verschieben einer Webseite, als würde die Seite am Finger kleben und von diesem tatsächlich direkt verschoben. Gibt es Verzögerungen oder stolpert die Reaktion, ist die Illusion der direkten Manipulation gebrochen.
- Drag'n Drop:** Ziehen und Fallenlassen. Bedienparadigma der grafischen Oberfläche, wo der Mauszeiger Objekte aufnimmt (Klick), bewegt (Klick halten) und loslässt (Klick lösen). Dient neben der Möglichkeit der Umplatzierung dazu, Aktionen auszulösen, z.B. eine Datei zum Löschen

auf den Papierkorb zu bewegen. Zieht man z.B. ein Datei-Icon auf ein Programm-Icon, wird die Datei von diesem Programm geöffnet.

**DTP:** Abkürzung für Desktop Publishing. Erstellung von Publikationen auf dem (virtuellen) Schreibtisch. Print-Produkte werden rein digital erzeugt und als Postscript-Datei oder PDF an die Druckerei gegeben. Bleisatz oder andere gestaltende Zwischenstufen sind nicht nötig. [84]

**Eingabefeld:** Bereich, in den Daten per Tastatur eingetragen werden, von Einzelzeichen bis zu Texten. Durch eine Umrandung optisch begrenzt; diese muss nicht synchron zur funktionalen Begrenzung sein (kann schmaler sein als die akzeptierte Textmenge). [247]

**Ergonomie:** Vorfahr der Usability. Unter Ergonomie werden alle physischen Aspekte zusammengefasst, die ein angenehmes Arbeiten ermöglichen (dadurch Reduktion von Unfallgefahren und Fehlern sowie Steigerung der Produktivität): beispielsweise Größe und Platzierung von Hebeln an einer Maschine, Lichtsetzung in Büros, Einstellmöglichkeiten für Büromöbel, Wartungsfreundlichkeit von Maschinen und Geräten. Als Begründer gilt unter anderem Raymond Loewy, der in seiner Biografie „Hässlichkeit verkauft sich schlecht“ Beispiele aus zahlreichen Bereichen seit Anfang des 20. Jahrhunderts beschreibt.

**Farbverlauf:** Innerhalb einer Fläche wechselt die Farbfüllung von einem Farbwert zu einem anderen. Dadurch erhält die Fläche eine farbige Dynamik und eine räumliche Anmutung. Bewährt haben sich dezente Verläufe, die von einem Farbton in eine bis zu zehn Prozent hellere oder dunklere Version des selben Tons übergehen. Mit CSS3 auch direkt in Webseiten möglich.

**Feedback:** Rückmeldung, Begriff aus der Kommunikationstheorie. Feedback bezeichnet jede Reaktion, die ausdrückt, dass der Computer auf Nutzeraktionen reagiert, der sich bewegende Mauszeiger ist das Feedback auf die Mausbewegung, eine Dialogbox kann das Feedback einer Anweisung sein. Feedback sollte affirmativ, bestätigend und motivierend erfolgen.

**Fehlermeldung:** Hinweismeldung, dass eine Eingabe nicht korrekt erfolgt ist oder etwas nicht wie vorgesehen läuft. Je nach Schweregrad des Fehlers wird eine Bestätigung angefordert. Fehlermeldungen sind konstruktiv formuliert, sodass der Nutzer erkennt, worin der Fehler besteht und wie er ihn beheben kann. [287]

**Fenster:** In der WIMP-GUI bildet ein Fenster den Arbeitsbereich, der den Monitor teilweise oder vollständig („Fullscreen“) ausfüllt. Die meisten Programme verwenden Fenster, um Informationen darzustellen oder bearbeiten zu lassen (Dokumente). Fenster verfügen über Titelzeile,

---

Buttons zum Schließen, Vergrößern, gegebenenfalls Scrollbalken, um den sichtbaren (Dokument-) Ausschnitt zu wählen, und den Hauptbereich. Zu unterscheiden sind Programm-, Dokument-, Pop-up- bzw. Dialog- und Palettenfenster. Fenster können auf dem Monitor frei verschoben und ihre Größe (meist) geändert werden.

**Formular:** In Analogie zu Papierformularen enthalten Formulare auf dem Monitor Eingabefelder, die auszufüllen sind. Übliche Daten sind: Zahlen, Texte, Ja/Nein-Angaben mittels Checkboxen oder eine Auswahl aus mehreren Optionen mittels Radio-Buttons. Für Formulare ist kennzeichnend, dass die Eingabefelder in einer bestimmten Weise angeordnet sind, z.B. bestehen Adressformulare aus: ein oder zwei Feldern für die Namenseingabe in der ersten Zeile, einem Feld für Zusätze, ein oder zwei Feldern für Straße und Hausnummer in der dritten Zeile und zwei Feldern für Postleitzahl und Ort in der vierten Zeile. Dieser optische Quasi-Standard erleichtert die Orientierung und unterstützt beim Ausfüllen. Die Formulareingabe wird mit Klick auf einen Button abgeschlossen. Formulare können sich über mehrere Bildschirmseiten erstrecken, die zu erscrollen sind, oder zwischen denen mit Weiter- und Zurück-Tasten gewechselt wird.

**Foveale Wahrnehmung:** Der Sichtbereich, auf den fokussiert wird, der als scharf erscheint. Auf einem Computermonitor meist der Bereich etwas oberhalb der Bildschirmmitte. [155]

**Gesten:** Auf Touch-Geräten bzw. Touch-Eingabegeräten für Computer sind einige Aktionen durch Gesten steuerbar, beispielsweise Zoomen durch Auseinanderspreizen bzw. Zusammenziehen von zwei Fingern oder das Erscheinen einer Befehlsliste durch Bewegen des Fingers vom Bildschirmrand. Gesten müssen erlernt werden, es gibt keine optischen Hinweise, wann welche Gesten wie funktionieren. Auf Spielkonsolen gibt es Gestensteuerung, die durch kameraerfasste Bewegung erfolgt; diese hat sich bislang nicht in der Computerbedienung etabliert.

**Gitter:** Ein unsichtbares Raster (in nur einer Dimension, meist Einteilung in Spalten), das bei der Anordnung der Elemente hilft, vergleichbar dem Satzspiegel im Drucklayout. [196]

**GOMS-Modell:** Verfahren zur Beschreibung eines Interface. Dabei werden Ziele (Goals), Aktionen (Operators), Aktions-Sequenzen (Methods) und Nutzungsregeln (Selection Rules) definiert bzw. beschrieben. [122]

**GUI:** Abkürzung für Graphical User Interface, grafische Benutzeroberfläche. Computerbedienung mit WIMP-Elementen via Point'n Click.

**Hierarchie:** Gibt es mehrere Elemente, können diese in Abhängigkeit zueinander stehen. Eines bildet ein Unterelement zu einem anderen oder

ist in seiner Wirkung von diesem abhängig. Solche Abhängigkeiten bestehen in der funktionalen, optischen bzw. gestalterischen und auf der inhaltlichen (semantischen) Dimension. Alle drei Hierarchie-Dimensionen sind synchron, d.h. eine funktionale Abhängigkeit ist auch optisch und semantisch erkennbar bzw. eine optische Unterordnung entspricht der funktionalen Abhängigkeit. Durch inhaltliche und optische Hierarchisierung wird der Arbeitsraum strukturiert und das Funktionsangebot erschlossen. Mitunter kann es geeignet sein, eine Hierarchieebene zur besseren Übersicht einzufügen, beispielsweise „Verzeichnis einfügen“ und erst auf der Hierarchieebene darunter wird der konkrete Verzeichnistyp (Inhalts-, Abbildungs-, Tabellen- oder anderes Verzeichnis) gewählt. Ziel ist, die Funktionen so in Hierarchien zu strukturieren, dass auf jeder Ebene zwei bis acht Elemente vorhanden sind. [166]

**Hover:** Optische Reaktion eines Elements auf die Berührung durch den Mauszeiger, beispielsweise das Färben eines Links vor dem tatsächlichen Klick. Steht auf Touch-Geräten nicht zur Verfügung, da kein Mauszeiger zum Berühren existiert.

**HTML:** Abkürzung für Hypertext Markup Language. Eine Seiten-Beschreibungssprache, in der gesteuert wird, was eine Internetseite enthält. Guter HTML-Code nutzt semantische Auszeichnungen (z.B. `<header>` `<artice>` `<nav>` `<footer>` `<h1>` `<h2>` `<ul>` `<p>` `<strong>` `<em>` etc.), um den Seiteninhalt zu strukturieren. Die Gestaltung erfolgt über CSS, Dynamisierungen sind mittels Javascript möglich.

**Icon:** Repräsentiert manipulierbare Objekte auf dem Bildschirm. Sinnbildliche Darstellung für Dateien, Programme, Laufwerke oder andere Elemente. Kann mittels Zeigegerät (Maus) aufgenommen und bewegt werden (Drag'n Drop).

**Infinite Scrolling:** Endloses Scrollen auf Webseiten. Eine Liste wird nicht auf mehrere Seiten verteilt, zwischen denen der Nutzer navigiert, sondern beim Erreichen des aktuellen Seitenendes wird automatisch der nächste Teil der Liste nachgeladen. Dadurch wirkt die Liste unendlich, und das Scrollen wird zur Hauptnavigation; eignet sich vorwiegend für Content und nicht für eCommerce. [146]

**Interface:** Schnittstelle. Bezeichnet den Bereich, in dem der Nutzer mit einem Gerät interagieren kann. Bei einem heutigen Computer umfasst dies mindestens: Monitor, Tastatur, Maus bzw. Touchpad. Auf einem Smartphone oder Tablet sind die Interaktionselemente in einem Gerät kombiniert, und die Interaktion erfolgt fast ausschließlich über den Bildschirm.

- 
- iOS:** Betriebssystem für iPhone und iPad. Dieses wird über Touchberührungen und -gesten bedient und stellt mangels Fenstern und Mauszeiger eine Abkehr vom WIMP-Bedienparadigma dar. Wird im Jahrestakt auf eine neue Version („Major-Release“) aktualisiert, das auf die aktuelle Hardware-Version von iPad und iPhone optimiert ist, aber auch mindestens die zwei Vorversionen dieser Geräte unterstützt. Die Basis des Betriebssystems ist die gleiche wie bei MacOS, doch die Nutzeroberfläche ist speziell an Touch-Geräte angepasst und nur für diese verfügbar.
- iPad:** Tablet der Firma Apple, erschien 2010 in der Größe von 9,7 Zoll, seit 2012 auch als „mini“ mit 7,9 Zoll. Wird mit iOS betrieben. [98]
- iPhone:** Smartphone der Firma Apple, erschien 2007 und erhält jährlich Updates der Hardware und Software (iOS). [96]
- Javascript:** Ermöglicht interaktive Funktionen innerhalb einer Webseite. Während Webseiten aus statischem HTML-Code bestehen, können mittels Javascript Teile der Seite nachgeladen, ein- oder ausgeblendet werden. Auch das Einlesen von Daten aus Datenquellen oder das Speichern in diese ist möglich. Das ermöglicht Webseiten, bei denen der Nutzer nur innerhalb einer Webseite bleibt, aber vielfältige Aktionen ausführen kann; meist als sogenannte AJAX-Applikation umgesetzt, „Asynchronous JavaScript and XML“. Zu solchen Web-Applikationen gehören beispielsweise Webmailer, Cloud-Dienste zur Online-Bearbeitung von Dokumenten oder Online-Kalender.
- Kommandozeile:** Tastaturbasierte Eingabemöglichkeit für Computerbefehle. Der Computer wird „in seiner Sprache“ direkt angewiesen, etwas zu tun. Populäre Beispiele sind MS-DOS, Unix/Linux-Shell („Terminal“ oder „Konsole“) und CP/M. [75]
- Kommunikation:** Faktor für gute Usability. Direkte (inhaltliche) und indirekte (gestalterische, behavioristische) Aussagen sind stets synchron und widerspruchsfrei. Der Nutzer erhält stets Feedback; jede Nutzer-Aktion bewirkt eine Computer-Reaktion. Motto: *„Do tell me, what I need to know, and don't hide information from me.“* [19]
- Kommunikation (2):** Gemäß dem Sender-Empfänger-Modell wird eine Nachricht zwischen zwei Entitäten ausgetauscht. Kommunikation entsteht dann, wenn eine Reaktion (in Form von Rück-Kommunikation bzw. Feedback) erfolgt, die in sinnhaftem Zusammenhang mit der ersten Nachricht steht. Auch die Bedienung eines Computers kann nach Kommunikationsregeln erfasst werden, indem Eingabe- und Ausgabedaten als Nachrichten interpretiert werden. Eine Kommunikation weist innere Kohärenz auf, und

es ist „nicht möglich, nicht zu kommunizieren“ (Paul Watzlawik). Jede Ausgabe des Computers wird vom menschlichen Nutzer als Kommunikationsbestandteil wahrgenommen. Die Kommunikation zweier Computer über Netzwerke ist stärker formalisiert, sie basiert auf Protokollen, die Anfang, Übertragung und Ende genau definieren.

**Konsistenz:** Zusammenhalt, Einheitlichkeit. Gleiches führt zu Gleichem, Unterschiedliches zu Unterschiedlichem. Sehen sich zwei Bedienelemente ähnlich, dann ist deren Funktionalität und/oder Verhalten auch ähnlich, sehen sie unterschiedlich aus, unterscheiden sie sich auch hinsichtlich Funktionalität und Verhalten. Konsistenz bezieht sich beispielsweise auf die Darstellung, Platzierung, Gestaltungskontext, Text bzw. Beschriftung, Interaktionsmöglichkeiten. [341]

**Kontextmenü:** Mittels Rechtsklick wird eine Liste geeigneter Funktionen aufgerufen. Diese stammen aus verschiedenen Menüs des Hauptmenüs und sind entsprechend des Umfelds, in dem der Rechtsklick erfolgt, zusammengestellt. Der Rechtsklick auf eine Datei liefert andere Kontextmenü-Einträge als der Rechtsklick auf ein Wort in einer Textverarbeitung. Webseiten verwenden standardmäßig das Kontextmenü des Browsers; zunehmend nutzen Web-Applikationen eigene Kontextmenüs, die besser an die konkreten Aufgaben angepasst sind. Da es keine optischen Hinweise gibt, ob sich das Öffnen des Kontextmenüs lohnt, listet es nur Funktionen auf, die auch an anderer Stelle (z.B. im Menü) vorhanden bzw. auch auf anderem Wege erreichbar sind. Kontextmenüs sind für viele Novizen die ersten Elemente, um sich als fortgeschrittene Nutzer zu fühlen.

**Konversion:** Gibt den Quotienten aus Erfolg und Potenzial an. In einem Webshop wird der Erfolg in Bestellzahlen gemessen und das Potenzial in Besucherzahlen; wenn von 1.000 Besuchern 80 eine Bestellung auslösen, hat der Webshop eine Konversion von 8 Prozent. Die Konversion kann für verschiedene Bereiche gemessen werden, z.B. Newsletteranmeldung, Anlegen eines Kundenkontos, Lesen eines bestimmten Beitrages, das Verfassen eines Kommentars oder einer Kundenrezension, auch das Verweilen auf der Webseite für mindestens x Minuten kann ein Konversionsziel darstellen. In vielen Situationen werden Konversionszahlen für die Erfolgsmessung verwendet, doch ist zu berücksichtigen, dass nur das Ergebnis erfasst wird und keine Antworten auf die Fragen, wie oder warum ein Ergebnis erzielt wurde. Mitunter hat die Welt außerhalb des Computers (z.B. Wetter, News, Epidemien, Politik) eine stärkere Auswirkung auf das Nutzerverhalten als alle Elemente auf einer Webseite oder in einer Software.

---

**Layout:** Anordnungssystem für die optische Organisation in der zweidimensionalen Fläche. Das Layout definiert die Regeln für die Platzierung, Größenverhältnisse und Relationen (z.B. oben/unten, links/rechts) der Funktions- und Inhaltsbereiche. Die potenzielle Gesamtfläche wird dazu in Bereiche unterteilt, z.B. mithilfe eines Rasters oder Gitters. Die Funktionen und Inhalte einer Software oder Webseite werden anschließend den Bereichen zugeordnet, dabei können mehrere Raster- oder Gitter-Bereiche zu einem funktionalen oder Inhaltsbereich zusammengefasst werden. Erst das Design schafft dann die konkrete optische Ausprägung der Elemente.

**Macintosh:** Name der Computer mit grafischer Oberfläche, die 1984 von der Firma Apple erstmals vorgestellt wurden. Lebt als „Mac“ fort, das als Bestandteil der Computernamen seitdem in Apple-Computern vorkommt, vom PowerMac über MacBook bis iMac und MacPro. Das Betriebssystem der Mac-Computer heißt entsprechend MacOS und teilt sich die Basis mit iOS. MacOS folgt – im Gegensatz zu iOS – der WIMP-Bedienung, die durch Übernahme von Touch-Gesten erweitert wurde. [81]

**Makro:** Der Nutzer zeichnet Einzelaktionen auf und kann diese dann als eigene Funktion genau so erneut aufrufen und nutzen. Visual Basic, Scripting-Fähigkeiten einiger Programme oder das MacOS-X-Tool „Automator“ (früher AppleScript) erlauben es so, eigene Funktionen zu definieren, die aus zahlreichen Einzelaktionen bestehen. Dies bewährt sich vor allem bei häufig wiederkehrenden Aufgaben und senkt die Fehlerquote, da kein Einzelschritt vergessen werden kann.

**Maus:** Eingabegerät für GUIs, das über eine Oberfläche bewegt wird. Die (durch eine Kugel oder einen optischen Sensor erfasste) Bewegung vollzieht der Mauszeiger auf dem Bildschirm nach. Die Maus verfügt über mindestens eine Taste, um auf dem Bildschirm einen Klick auszulösen. Die meisten Modelle bieten weitere Tasten für Zusatzfunktionen, beispielsweise den populären „Rechtsklick“, um das Kontextmenü aufzurufen.

**Memento mori:** Lateinisch für Gedenken des Todes. Diskussionsszenario, um die Ursachen für das (unterstellte) Scheitern zu untersuchen. [301]

**Menü:** Hierarchische Gliederung der Befehle innerhalb eines Programms. In einer Zeile sind die Rubriken (mit ihren Menütiteln) aufgelistet, beim Klick darauf öffnet sich das zugehörige Menü und zeigt die möglichen Befehle an, ein Klick auf einen der Befehle führt diesen aus. Somit sind potenziell alle verfügbaren Befehle stets zugänglich und müssen nicht nachgeschlagen werden. [270]

- Microsoft Bob:** Softwarepaket von Microsoft, wurde 1995 veröffentlicht. Eine spielerische Oberfläche bildet die grafische Benutzeroberfläche des Betriebssystems, ein Comic-Assistent unterstützt den Anwender bei der Arbeit. Enthalten waren diverse Einzelprogramme für private Anwender, beispielsweise Finanzberater, Haushaltsmanager, E-Mail-Clients, GeoSafari Quiz, Schreibprogramm, Kalender, Adressbuch. Für PC-unerfahrene Benutzer wurden technische Details unter den Analogien aus der häuslichen Umgebung verborgen. [27]
- Mock-up:** Ein „Als ob“-Entwurf. Die Stufe einer Software oder Webseite zwischen Skizze und fertigem Prototyp. Ein Mock-up verdeutlicht die wesentlichen Funktionen in einer reduzierten Gestaltung, um die Bedienung und Funktionsweise begutachten zu können. Je nach Fokus kann ein Mock-up bereits sehr weit gestaltet sein, z.B. bei Internetseiten, aber die Funktionalität nur andeuten. Oder es simuliert eine Funktionalität und dient dazu, die gestalterischen Entscheidungen zu prüfen und ggf. zu modifizieren.
- Modus:** Eigener Zustand des Interface. Der einfachste Modus ist der ausgeschaltete Computer, denn dann erfolgt keine Reaktion auf Nutzerinteraktionen, allenfalls auf das Drücken der „Ein“-Taste. Innerhalb eines Modus wirken sich die Nutzer-Aktionen anders aus, z.B. führt im Format-Dialog von MS-Word das -Drücken zum Übernehmen getätigter Einstellungen, und der Dialog wird geschlossen; im Textbearbeiten-Modus bewirkt die -Taste ein Absatzende. Jeder Modus hat Anfang und Ende, der Nutzer muss ihn beginnen und beenden. Daher sind modale Dialoge (da sie die Interaktion mit dem eigentlichen Programm verhindern) mit „Ok“ oder „Abbrechen“ zu schließen. Wenn möglich, sind modale Dialoge zu vermeiden, z.B. lassen sich Formate über Paletten oder Symbolleisten direkt zuweisen, ohne einen interaktionsverändernden Dialog aufzurufen.
- MS-DOS:** Microsoft Disk Operating System, erschien 1981 für den IBM-PC. Kommandozeilenbasiertes Betriebssystem, fand in den 1980ern rasch Verbreitung, bildete den Standard für sogenannte IBM-kompatible PCs und diente als Unterbau für Windows 1 bis 3, 95, 98 und Me.
- Multi-Touch:** Eingabemethode, bei der auf einer berührungssensitiven Fläche (z.B. Touch-Display oder Trackpad) mehrere Fingerbewegungen auf einmal (als Geste) erkannt und als eine Eingabe behandelt werden. Z.B. löst das Auseinanderziehen zweier Finger ein Hineinzoomen aus, während das Zusammenschieben ein Herauszoomen bewirkt. Das parallele Bewegen zweier Finger bewirkt ein Scrollen. Daneben sind zahlreiche spezielle Gesten mit mehreren Fingern möglich.

---

**Norton Commander:** Ein Dateimanager (erschieden 1986), der durch sein Layout (zwei Spalten, in denen die Inhalte unterschiedlicher Bereiche von Datenträgern angezeigt wurde) den Standard für Dateiverwaltung in den 1980ern setzte. Für nahezu alle Betriebssysteme gab es Dateimanager, die nach dem gleichen Prinzip funktionierten. Dieses wurde erst obsolet, als sich die Browser-Logik in der Dateiverwaltung durchsetzte. [26]

**Nutzer:** Der konkrete oder idealisierte Anwender einer Software oder Besucher einer Webseite. Wird in der Planung als Persona beschrieben.

**Nutzergruppe:** Mehrere Software-Anwender oder Webseiten-Besucher, die in bestimmten Eigenschaften Gemeinsamkeiten haben. Sinnvoll ist die Gruppenbildung nach Erfahrungslevel, Aufgabenziel, Nutzungskontext oder (für Marketingzwecke) nach sozialen bzw. demografischen Aspekten.

**Objekt:** Element, das in eine Aktion einbezogen bzw. von dieser direkt betroffen ist: Icon, markierter Textabschnitt, Dokumentbereich oder Listenzugriff; manche Funktionen wirken sich auf das gesamte Dokument oder andere Entitäten aus (Druckauftrag, Papierkorbinhalt, Maschinenauftrag, Festplatten- oder Ordnerinhalt).

**Optische Achse:** (Trenn-)Linie, die zwar nicht existiert, aber durch andere Elemente evoziert wird. Die Tabellen in diesem Buch besitzen keine vertikalen Linien, dennoch werden die Spalten durch die Textränder deutlich als optische Achsen markiert. Jeder Bildschirmaufbau sollte möglichst wenige solcher Achsen aufweisen (aber deutliche), idealerweise im rechten Winkel zueinander. [163]

**Palette:** Zusatzfenster in Programmen zur Dokumentbearbeitung. Palettenfenster bzw. -bereiche enthalten (wie Symbolleisten) Funktionen zum schnellen Zugriff. Insbesondere funktionsreiche Programme wie die Adobe Creative Suite beschleunigen so die Arbeit, indem der Nutzer selbst wählt, welche Paletten er verwendet; somit sind diese Funktionen schneller (in einer nicht-modalen Arbeitsweise) erreichbar und unterbrechen die Arbeit nicht.

**Periphere Wahrnehmung:** Bereich auf dem Monitor, der nicht fokussiert wird und daher außerhalb der fovealen Wahrnehmung liegt. [155]

**Persona:** Prototypische (fiktive) Nutzer einer Software oder Webseite. Werden für die Anforderungserfassung (in Form von User-Stories) benötigt. Jede Persona wird mit Steckbrief und biografischen Daten sowie Motivation und Nutzungskontext skizziert. [55]

**Piktogramm:** Oberbegriff für Icon und Symbol, bildet eine Information durch vereinfachte grafische Darstellung ab.

- Point'n Click:** Zeigen und Klicken. Bezeichnet den Aktionsrahmen des Mauszeigers: auf etwas zeigen (sich bewegen) und etwas anklicken. Bedienparadigma der grafischen Oberfläche (WIMP), wo der Mauszeiger auf Befehle bzw. Bedienelemente zeigt und diese durch Klick ausgelöst werden bzw. Elemente mit dem Mauszeiger durch Klick aufgenommen und verschoben werden (Drag'n Drop).
- Prototyp:** Entwicklungsstufe einer Software oder Webseite (als Ganzes oder eines Teils), die bestimmte Funktionen abbildet und der finalen Gestaltung recht nahe ist. Wird vor allem für Machbarkeitsbelege oder Akzeptanztests benötigt. Je nach Fokus kann eine Funktion korrekt implementiert oder simuliert werden. Relevant ist, dass der Prototyp einen definierten Bereich der finalen Software oder Webseite erlebbar umfasst.
- Radio-Buttons:** Bedienelement mit zwei Zuständen (○ und ⊙), markiert die gewählte Option von mindestens zwei. Radio-Buttons sind in Gruppen organisiert, die mehrere alternative (sich gegenseitig ausschließende) Optionen beschreiben (z.B. Hochformat/Querformat, Lautstärke-Level 0 bis 7), von denen genau eine gewählt wird. Wie ein Auswahlménü ermöglicht es ausschließlich die Wahl einer Option von mehreren; es ist nicht vorgesehen, keine Wahl zu treffen. [253]
- Raster:** Ein unsichtbares Gitter (in zwei Dimensionen), das bei der Anordnung der Elemente hilft. Die verfügbare Fläche wird in gleichgroße Rechtecke aufgeteilt, die für benötigte Funktions- und Inhaltsblöcke dann zusammengefasst werden. [196]
- Respekt:** Faktor für gute Usability. Fachliche Kompetenz, angemessenes Verhalten in Bezug auf Nutzerstärken und -schwächen. Motto: „*Don't waste my time.*“ [16]
- Ribbon:** Bedienparadigma der grafischen Benutzeroberfläche, von Microsoft in der Version von MS Office 2007 erstmals vorgestellt. Dabei werden alle Befehle grafisch dargestellt (beispielsweise als Piktogramm oder Formatvorschau) und diese durch Anklicken ausgelöst. Die Befehle sind in mehren Bereichen untergebracht, zwischen denen gewechselt werden kann. Die Ribbons ersetzen in Microsoft-Programmen die vormaligen Text-Menüs. [95, 274]
- Schatten:** Gestalterisches Mittel, um Räumlichkeit zu simulieren. Einander überlagernde Elemente werden durch Schattenwurf hierarchisiert, dabei wird in der zweidimensionalen Arbeitsfläche eine räumliche Tiefe suggeriert und somit ein Arbeitsraum geschaffen. Mit CSS3 sind auch Schattenwürfe in Webseiten unaufwändig möglich.

---

**Schnittstelle:** (auch Interface) Vermittlung zwischen zwei voneinander unabhängigen Bereichen, z.B. zwischen Computer und Drucker oder zwischen Prozessor und Festplatte oder zwischen Mensch und Maschine. Die Schnittstelle definiert den Punkt der Verbindung und die Möglichkeiten, Bedingungen und Regeln der Datenübergabe zwischen beiden Bereichen.

**Schriftart:** Grafische Gestaltung eines Zeichensatzes. Dabei wird u.a. zwischen Serifen- (Times, Cambria oder Garamond) und serifenlosen Schriften (Helvetica, Arial oder Calibri) unterschieden, zwischen proportionalen Schriften (Times oder Helvetica) und jenen mit fester Laufweite (Monaco, Consolas oder Courier). Jede Schriftart liegt als Font-Datei im System vor, meist TrueType (.ttf) oder OpenType (.otf), und kann für die Darstellung von Schrift ausgewählt werden. Für die GUI sind vom System oder Nutzer Standard-Schriften eingestellt. Im Browser sind nur Schriftarten darstellbar, die auch auf dem eigenen Rechner installiert sind; moderne Browser binden zusätzliche Schriften für die Webdarstellung ein. [217]

**Scrollen:** Verschieben des sichtbaren Bereichs. Sind Dokumente, Ansichten oder Webseiten zu groß, um sie auf einmal darzustellen, ist nur ein Ausschnitt sichtbar. Der Nutzer holt mittels Touch-Geste, Mausrad oder Verschieben der Scrollbalken am Fensterrand einen anderen Ausschnitt in den sichtbaren Bereich. Dabei verdeutlichen Scrollbalken die relative Position des Ausschnitts.

**Sieben Stufen der Interaktion:** Von Donald Norman in „The Design of Everyday Things“ (1988) definiert: 1. Formulierung des Ziels. 2. Formulierung des Plans. 3. Spezifikation der Aktion. 4. Ausführung der Aktion. 5. Wahrnehmung des Systemstatus. 6. Interpretation des Systemstatus. 7. Auswertung des Resultats. [108]

**Skeuomorphismus:** Designrichtung, die eine möglichst realistische Abbildung realer Objekte auf dem Monitor anstrebt. Da der Monitor als zweidimensionale Abbildung die dritte Raumdimension vernachlässigen muss und aufgrund seiner Maus- und Tastaturbedienung nach anderen Bedienparadigmen verlangt als die Objekte in der realen Welt, geriet dieser Ansatz in Verruf. Gern wurde dabei auf die Kalender-App von MacOS X verwiesen, die durch hyperrealistische Texturen und unnötigen Realismus die Bedienung eher behinderte als förderte. [210]

**Slider oder Regler:** Bedienelement für das Einstellen eines Zahlenwertes in einem definierten Bereich. Ein Slider ist einem Schieberegler nachgebildet. Ein Element wird entlang einer markierten Achse verschoben und dadurch ein Wert eingestellt, der meist daneben oder darunter angezeigt wird. Werden zwei Schiebe-Elemente auf einem Slider kombiniert, lässt

sich ein Wertebereich „Von–Bis“ einstellen. Problematisch ist die exakte Bedienung, insbesondere bei Touch-Bedienung gestaltet sich das pixelgenaue Einstellen des Wertes schwierig. „Slider“ dient auch als Bezeichnung für einen Bereich auf Webseiten, wo mehrere gleichartige Elemente nebeneinander oder untereinander angezeigt werden und der Nutzer weitere dieser Elemente in dem Bereich anzeigen lassen kann, z.B. durch Scrollen oder Klicken auf Pfeil-Tasten am Bereichsrand. [262]

**Status:** Zustand, der dem Nutzer mitgeteilt wird, dabei gibt es mindestens zwei voneinander klar unterscheidbare Zustände. Im User-Interface erfolgt meist die Bewertung zur Entscheidungsunterstützung, ob es sich um einen guten Zustand („Es hat alles geklappt.“), einen Hinweis („... Trotzdem fortfahren?“) oder ein Problem oder einen Fehler handelt. Ggf. genügt eine sogenannte Statuszeile zur Information des Nutzers.

**Symbol:** Repräsentiert Aktionen in bildhafter Form. Werden beispielsweise in Symbolleisten oder Paletten zusammengefasst.

**Symbolleiste:** Sammlung von Symbolen oder Funktionen, die (meist am oberen Fensterrand) Zugriff auf häufig benötigte Funktionen bieten. [274]

**Tastenkürzel:** (auch Tastaturkürzel) Mehrere gemeinsam gedrückte Tasten lösen die gleiche Aktion aus wie der Klick auf einen Menüeintrag oder eine bestimmte Funktion. Dank Tastenkürzel muss der Nutzer nicht die Hände von der Tastatur heben, bei genügend Übung ist fast die komplette Bedienung über Tastatur möglich. [39]

**Technische Schulden:** In der Architektur-Metapher entspricht dies den Hypotheken, die mit jeder technologischen Entscheidung und schlechter Umsetzung aufgenommen werden. Dazu gehören Aufwände für Änderungen, Erweiterungen, Ersatzlösungen. Technische Schulden können bewusst in Kauf genommen werden, um ein Projekt schnell abzuschließen, oder unbewusst entstehen, wenn beispielsweise die Auswirkungen einer technologischen Entscheidung nicht absehbar waren. [4]

**Touch Screen:** Bildschirminhalte können direkt mittels Fingerberührung „angeklickt“ (also ausgelöst) oder via -bewegung manipuliert werden. Dabei entfällt das „Point“ des „Point’n Click“, was zum Teil neue Bedienkonzepte nötig macht; damit orientiert sich die Touch-GUI nicht mehr am WIMP-Paradigma.

**Trackpad:** (auch „Touchpad“) Eingabemöglichkeit, bei der der Mauszeiger durch Fingerbewegungen über eine sensitive Oberfläche gesteuert wird. Moderne Trackpads erkennen auch Gesten und Multi-Touch-Bedienung.

- 
- Transparenz:** Gestalterisches Mittel, um Räumlichkeit zu simulieren. Bestimmte Elemente erscheinen halb-durchsichtig, sodass das Dahinterliegende sichtbar wird. Meist erfolgt die Kombination mit einem Weichzeichner-Filter. Einander überlagernde Elemente werden durch Transparenz hierarchisiert, dabei wird in der zweidimensionalen Arbeitsfläche eine räumliche Tiefe suggeriert und somit ein Arbeitsraum geschaffen. Mit CSS sind auch Transparenzen in Webseiten unaufwändig möglich. Vorsicht ist geboten, da durch Transparenz auch Bildschirmteile sichtbar werden, die für die eigentliche Aufgabe keine Relevanz besitzen. Transparenz kann helfen, den Kontext für bestimmte Aktionen sichtbar zu halten (z.B. bei transparenten Menüs oder Abdunkeln einer Webseite bei einer sogenannten Light- oder Shadow-Box).
- Unterstützung:** Faktor für gute Usability. Beschreibt das Maß, in dem der Nutzer etwas als nützlich und hilfreich empfindet. Wird der Grat zur Gängelung oder Bevormundung überschritten, schlägt der Faktor in sein Gegenteil um. Motto: „*Do help me doing what needs to be done.*“ [17]
- Usability:** Begriff, der Konzepte zur Erreichung, Messung, Beurteilung der Brauchbarkeit, (Be-)Nutzbarkeit, Bedienbarkeit, Gebrauchstauglichkeit, Software-Ergonomie und Benutzerfreundlichkeit umfasst.
- Use-Case:** Anwendungsfall, der als Anforderung erfasst wird. Sachliche Beschreibung aus Nutzersicht, wie eine Software oder Webseite zur Erreichung eines Zieles benutzt wird („Ich klicke ... und erhalte ... ich wähle ... und bestätige ...“). Die konkreten Funktionen werden dabei als „Black Box“ behandelt und nicht näher beschrieben. [60]
- User Experience:** Beschreibt die Gesamterfahrung aus Gestaltung, Funktionalität und Leistungsmerkmalen eines Produkts. Usability ist davon nur ein Aspekt. [9]
- User-Interface:** Die Schnittstelle zwischen Mensch und Maschine. Besteht aus Ausgabe (meist auf dem Bildschirm), Eingabemöglichkeiten (via Tastatur, Maus, Trackpad u.a.m.) sowie den definierten Interaktionsmöglichkeiten.
- User-Story:** Textliche Beschreibung, was der Nutzer erreichen will und wie er es erreichen wird. Basiert auf den Personas, die mit bestimmtem Wissen und konkreten Zielen eine Software oder Webseite nutzen. [62]
- User-Test:** Versuchsreihe mit 1 bis X Vertretern der Zielgruppe, die eine fertige Software oder Webseite bzw. deren Prototypen in möglichst realistischer Weise anhand von Aufgaben verwenden. Ziel ist, dabei Fehler und Probleme in der Bedienung zu erkennen. Zu den Methoden gehören u.a. Eye-Tracking, „lautes Denken“, Bildschirm- und/oder Videoaufzeichnung,

- Zeitmessungen, Contextual Inquiry. User-Tests zeigen nur, welche Defizite oder Optimierungspotenziale bestehen, konkrete Maßnahmen werden erst im nächsten Schritt aus den Erkenntnissen abgeleitet. [324]
- Verständnis:** Faktor für gute Usability. Die Ziele des Nutzers stehen im Vordergrund, und dessen Schwächen wird aktiv begegnet. Motto: „*Do not pretend you didn't understand me.*“ [18]
- Vertrauen:** Faktor für gute Usability. Basiert auf Zuverlässigkeit und Konsistenz. Motto: „*Do as I expect you to do.*“ [17]
- Weißraum:** Die leere, funktionslose Fläche zwischen Funktions- und Inhaltsblöcken. Hilft bei der Strukturierung und optischen Gliederung der Arbeitsfläche, schafft Kontraste und bietet dem Auge Erholung. [168]
- Widget:** Ein in sich abgeschlossener Funktionsblock, z.B. in der Nutzeraktion oder in der Software selbst. Widgets sind so gestaltet, dass sie an mehreren Stellen verwendet werden können. Beispiele: Die Formateinstellungen für Schrift können an mehreren Stellen in einem Programm aufgerufen werden (z.B. als Palette, als Dialog, als Bestandteil anderer Einstellungen), dabei wird stets das selbe Widget verwendet, d.h. das selbe Aussehen und die selben Funktionen (optische und funktionale Konsistenz). Die Suchlogik kann als Software-Widget programmiert werden, sodass sie an vielen Stellen wiederverwendet wird und je nach Kontext korrekt reagiert.
- Widget-Level:** Methode zur Beschreibung eines User-Interface. Dabei wird das Interface in möglichst kompakte Funktionsblöcke (Widgets) zerlegt. [126]
- Windows:** Name des grafischen Betriebssystems von Microsoft, das ursprünglich auf MS-DOS aufsetzte, später (als Windows NT) eine eigene Basis mitbrachte. Seit Windows Acht verschmelzen alle Gerätetypen (Computer, Laptop, mobile Touch-Geräte) und erhalten das selbe Windows. Das sogenannte „Modern Design“ (mit Kacheln, Ribbons und Charmbar) ist speziell auf die Bedürfnisse von Touch-Geräten abgestimmt. Windows folgt der WIMP-Bedienung, die durch Übernahme von Touch-Gesten erweitert wurde. Microsoft Windows ist nicht an einen bestimmten Hardware-Hersteller gebunden, sondern für nahezu alle Computer verfügbar (auch für Mac-Computer mittels Virtualisierungssoftware oder nativ via „Boot Camp“). Derzeit läuft Windows auf weit über 90 Prozent der Computer weltweit, damit bilden dessen Bedienlogik und -elemente die Basis für die Erwartungshaltung der meisten Nutzer.
- WIMP:** Bezeichnung für die grafische Bedienung, die aus Windows (Fenster), Icons, Menus und Pointer (Mauszeiger) besteht.

---

**Wysiwyg:** What you see is what you get. Der Text sieht auf dem Bildschirm so aus, wie er aus dem Drucker kommt. Ohne Wysiwyg würde man einfach den Text hintereinanderweg schreiben, mit bestimmten Befehlen Formatierungen vornehmen und dann schauen, wie es beim Drucken aussieht. Z.B. Latex als Textsystem basiert nicht auf Wysiwyg, auch die Bearbeitung von HTML-Quelltext ist Nicht-Wysiwyg. Gibt es eindeutige Formatvorlagen oder -vorgaben, die nur mit entsprechendem Inhalt gefüllt werden, ist die Eingabe in einem Nicht-Wysiwyg-Editor effektiv. Software benötigt Wysiwyg-Fähigkeiten, wenn der Nutzer das fertige Aussehen aktiv beeinflussen kann (Layout, Design; Fett- oder Kursivsetzungen sind kein aktives Gestalten) und entsprechend seiner Ziele auch will bzw. muss.

**Ziel:** Das, was der Nutzer erreichen will oder muss. Das Ziel bildet die Antwort zur Kernfrage „Warum soll ich etwas benutzen?“ D.h. während der Nutzung kommuniziert die Software bzw. Webseite, dass sie zur Zielerreichung geeignet ist.