

# Interfacedesign

Das **Interfacedesign** („die **Schnittstellengestaltung**“) beschäftigt sich mit der Gestaltung von **Benutzerschnittstellen** zwischen Mensch und Maschine. Dafür werden die Bedingungen, Ziele und Hindernisse dieser Interaktion sowohl von menschlicher als auch von technischer Seite erforscht und später – soweit möglich – auf den Menschen hin optimiert.

Ziel des Interfacedesigns ist eine Anwenderschnittstelle, die so gestaltet ist, dass ein möglichst breiter Kreis von Nutzern eine optimale Wunsch-/Bedürfnis-/Zielerfüllung durch angemessene Handlungsschritte erfährt.

In der Definition des Begriffes begrenzt Jef Raskin die Bedeutung nicht nur auf die Gestaltung grafischer Benutzeroberflächen (**GUI**), sondern nutzt den mit *Interface* bezeichneten Begriff stellvertretend für eine „Schnittstelle zwischen Mensch und Maschine oder Mensch und Computer“. Eine Spracherkennung ist demnach ebenfalls ein Interface.

Die Interaktion mit einer Bedienoberfläche – zumeist ein Bildschirm, aber auch mit Automaten oder etwa Maschinen – soll vom Interaktionswunsch des Nutzers über Ein- und Ausgabe von Daten per Tastatur/Steuerung/gezielte Handlung in angemessener Zeit zu einem abgeschlossenen und sinnvollen Ergebnis führen. Ein wichtiger Aspekt hierbei ist, dass der Nutzer bei der Interaktion ein möglichst positives Anwendungserlebnis **User Experience** erfährt.

Interfacedesign ist somit ein Teilbereich der **Mensch-Computer-Interaktion**.

In der Praxis werden dafür üblicherweise schon während der Entwurfsphase Tests an der jeweiligen Zielgruppe durchgeführt.

Interfacedesign kommt in den unterschiedlichsten Bereichen zur Anwendung.

Einige der wichtigsten Branchen und Anwendungsgebiete sind:

- Automotive
  - Navigationsgeräte
  - Fahrerassistenzsysteme
  - Fahrzeug-Personalisierung
  - Verkehrsleitsysteme
- Consumer
  - Mobile Geräte (Handy, Tablet, Navi, MP3 Player, Apps)
  - Home Entertainment (TV, Smart TV, Spielkonsolen, Media Center)
  - Haus- und Küchengeräte
  - Home Automation (Haussteuerung, vernetztes Wohnen, Hausroboter)
- Enterprise
- Industry
- Medical & Pharma

Insbesondere in der Medizintechnik kommt zur Ästhetik noch die Sicherheit der medizinischen Geräte als weiterer Faktor hinzu. Fehlbedienungen an Medizintechnik zählen zu den häufigsten Ursachen von Unfällen im Gesundheitswesen.

## Geschichte

---

Interfacedesign ist eine junge Disziplin, deren Geburtsstunde mit dem Ende der textgesteuerten Computersteuerung, und dem Beginn von visuellen Anwenderschnittstellen einherging (Arbeiten mit der Maus; grafische Darstellung von Inhalten). Damit entstand auch der Begriff der **Software-Ergonomie**.

Das Internet mit seiner weltweiten Verbreitung ab Ende der 90er Jahre brachte den Begriff der **Usability**, unter dem zunächst Privatpersonen (**Jakob Nielsen**) das neue Medium auf Gebrauchstauglichkeit durch heterogene Benutzergruppen untersuchten. Der fortschreitende weltweite Einsatz und die Akzeptanz von digitalen Angeboten, verbunden mit ihrer hohen Interaktivität, führten zur gegenwärtigen universitären und kommerziellen "Usabilityforschung", die strukturiert Problemfelder in der Mensch-Maschine-Interaktion untersucht.

## Literatur, Forschung, Lehre

---

Wichtige Namen in diesem Zusammenhang sind **Donald Norman**, **Jef Raskin**, **Ben Shneiderman**, **Jakob Nielsen** und **Steve Krug**, die alle einen Schwerpunkt auf den „Common-Sense-Approach“ legen.

- Alan Cooper: *About Face 3.0: The Essentials of Interaction Design*, Wiley & Sons, Auflage: 3. überarb. (30. Mai 2007), **ISBN 978-0-470-08411-3**.
- Steve Krug: *Don't Make Me Think!: A Common Sense Approach to Web Usability*, New Riders, Auflage: 2nd ed. (8. September 2005), **ISBN 978-0-321-34475-5**.
- Jef Raskin: *Das intelligente Interface. Neue Ansätze für die Entwicklung interaktiver Benutzerschnittstellen*, Addison-Wesley; Auflage: 1. Aufl. (15. April 2001), **ISBN 3-8273-1796-7**.
- Ben Shneiderman: *Designing the User Interface*, Addison-Wesley Longman, Amsterdam (Mai 2004), **ISBN 0-321-26978-0**.

# 7 Gesetze des User Interface Designs

Während Websites heutzutage einfache User-Interfaces haben, die vor allem aus Navigation und einem Kontaktformular bestehen, bringen neue Technologien und Standards eine Nachfrage für dynamischere und individuellere Erfahrungen mit sich.

Dies bedeutet zwangsläufig eine intensivere Beschäftigung mit dem Thema Interface. Lasst uns deshalb einen Blick darauf werfen, was Ihr dabei unbedingt beachten solltet:

## 1. Das Gesetz der Klarheit

**Der User wird Interface-Elemente ohne einen klaren Sinn von vornherein nicht beachten.**



*Nutzt Ihr Gmail? Ich tue es. Bis zum letzten Update hatte Gmail eine sehr klare Textnavigation oben auf der Seite — Kalender, Drive, Seiten und andere Google-Services standen sofort zur Verfügung, wenn man auf den entsprechenden Button klickte.*

Dann entschied sich Google, dies alles zu „vereinfachen“ und hinter abstrakten Buttons zu verstecken. Das Ergebnis? Die meisten User

bemerkten die Icons nicht einmal, und Gmail wurde von einer Flut von Support-Anfragen überschüttet.

Für gewöhnlich meiden und ignorieren Menschen Dinge, die sie nicht verstehen — das ist typisch für die menschliche Natur. Vermeidet es, Interface-Elemente zu entwickeln, die bei den Usern die Frage aufwerfen, was sie tun müssen, denn niemand wird sich die Zeit nehmen, es herauszufinden.

## 2. Das Gesetz der bevorzugten Handlung

**Der User wird sich wohler fühlen, wenn er versteht, welcher Schritt als bevorzugte Handlung gedacht ist.**



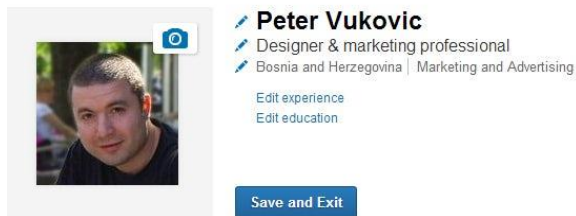
Werft mal einen Blick auf die oben gezeigte Twitter-Seite. Meint Ihr, dass neue User verstehen, was sie tun sollen?

Ganz offensichtlich sollen sie etwas twittern. Allerdings ist der Button mit der Anweisung „Was gibt's Neues?“ nicht besonders eindeutig (siehe das Gesetz der Klarheit), und die Input-Box geht in ihrer optischen Umgebung eher unter. Vom Standpunkt des Designs aus betrachtet erscheint es fast so, als wolle Twitter entweder, dass die User etwas suchen oder von einer der Optionen im Navigationsmenü auf der rechten Seite Gebrauch machen, da diese Interface-Elemente am deutlichsten hervorstechen.

Die User sollten sich niemals fragen, was sie als nächstes tun müssen — der nächste Schritt sollte sich einfach von selbst erklären.

### 3. Das Gesetz des Kontexts

**Der User erwartet die entsprechenden Kontrollelemente in der Nähe des jeweils dazugehörigen Objekts.**



Wie ändert Ihr Euren Namen bei Facebook? Ihr geht auf „Einstellungen“ rechts oben in der Ecke, klickt auf „Kontoeinstellungen“, findet dort Euren Namen und klickt auf „Bearbeiten“. Und wie funktioniert genau der gleiche Vorgang bei LinkedIn? Ihr klickt auf den Bleistift direkt neben Eurem Namen.

Die User erwarten immer, Interface-Elemente direkt im unmittelbaren Kontext der Objekte vorzufinden, die sie kontrollieren möchten. Es ist wie im echten Leben: Wenn Ihr Popcorn machen wollt, geht Ihr zu eurer Mikrowelle und betätigt ganz einfach den Schalter am Gerät. Es wäre nicht besonders praktisch, wenn Ihr von eurer Mikrowelle die Anweisung bekämt, erst mal die Treppen in den Keller hinunterzugehen, die Kellertür aufzuschließen, zum Sicherungskasten hinzugehen und den Schalter mit der Nummer G-35 umzulegen, um das Popcorn-Programm der Mikrowelle zu starten (was dem Beispiel der Namensänderung bei Facebook entsprechen würde).

Macht die Dinge für Eure User so einfach wie möglich – wenn etwas geändert oder auf andere Weise kontrolliert werden kann, platziert die entsprechenden Kontrollelemente in unmittelbarer Nähe.

### 4. Das Gesetz der Standardeinstellungen

**Der User wird bestimmte Standardeinstellungen kaum verändern.**

- Die meisten Menschen haben einen Standard-Hintergrund und Klingelton auf ihrem Handy.
- Die meisten Menschen (auch Ihr) ändern niemals die Werkseinstellungen ihrer Fernsehgeräte.
- Die meisten Menschen ändern niemals die Standardtemperatur ihres Kühlschranks.

Standardeinstellungen sind sinnvoll: Wir nehmen bestimmte Standards kaum zur Kenntnis, und doch beherrschen sie die ganze Welt. Vergewissert Euch deshalb, dass alle Standardwerte so nützlich und praktisch wie möglich sind — Ihr könnt davon ausgehen, dass die meisten User sie niemals ändern werden.

## 5. Das Gesetz der geleiteten Handlung

Der User wird vermutlich nur dann etwas tun, wenn er dazu aufgefordert wird.



Es gibt einen großen Unterschied zwischen der Erwartung, dass User etwas von alleine tun und der spezifischen Anweisung, die entsprechende Handlung auszuführen.



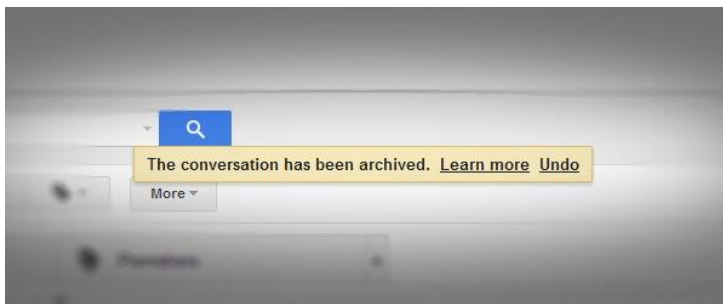
Als zum Beispiel LinkedIn seine „Endorsement“-Funktion einführte, wurde nicht erwartet, dass die User selbst herausfinden müssen, wie diese zu bedienen ist. Stattdessen wurden deutlich sichtbare Call-To-Action-Banner erstellt, die direkt

über den entsprechenden Profileseiten erschienen. Dies sorgte in Verbindung mit der Tatsache, dass Menschen gerne Bestätigungen zu Dingen abgeben dafür, dass diese Funktion zu einem riesigen Erfolg wurde.

Was lernen wir daraus? Wenn ihr möchtet, dass Eure User irgendetwas tun, dann gebt ihnen eine direkte Anweisung.

## 6. Das Gesetz des Feedbacks

Der User wird sich sicherer fühlen, wenn Ihr ihm ein klares und durchgängiges Feedback gebt.



*Das ist einfache Logik — je mehr die User das Gefühl haben, dass Euer Interface ihnen eine Handlung vorgibt, desto sicherer werden sie sich fühlen.*

Gmail ist ein großartiges Beispiel für gutes Feedback. Hier bekommt Ihr für jede ausgeführte Handlung eine klare Benachrichtigung, die auch Links zu weiteren Informationen sowie die Möglichkeit enthält, die entsprechende Handlung rückgängig zu machen. Hierdurch erhalten die User das Gefühl, alles unter Kontrolle zu haben und fühlen sich sicherer im weiteren Umgang mit dem Produkt.

## 7. Das Gesetz der Vereinfachung

Der User wird sich eher dazu bewegen lassen, eine komplexe Handlung durchzuführen, wenn diese in kleine Schritte unterteilt wird.

The image compares two user interfaces for account creation. The left side shows a complex form titled 'Submit An Announcement' with many fields for wedding details. The right side shows a simplified form titled 'Step 1: jQuery Tools Account' with three steps: 1. Enter Your Email Address, 2. Pick a username, and 3. Choose a Password. The right side is much easier to navigate.

**Submit An Announcement**

Click here for guidelines and more information about submitting wedding and celebrant "Vows" video series, to The Times.

**Type of Celebration**

☐ Legal Marriage  
☐ Civil Union  
☐ Commitment Ceremony

**Photographs**

No photos

**Legal Date of Marriage (Not Ceremony)**

**Bride**

Title

First Name (required)  Middle Name

Last Name (required)  Suffix

Date of Birth (required)

Education

Occupation (required)

**Contact Information**

Email Address (required)

**Groom**

Title

First Name (required)  Last Name (required)

Date of Birth (required)

Education

Occupation (required)

**Contact Information**

Email Address (required)

**1. Create Account** **2. Contact Information**

**Step 1: jQuery Tools Account**

Please enter your login information:

**1. Enter Your Email Address \***

Your password will be sent to this address. Your address will not be used for anything else.

**2. Pick a username**

Your preferred username to be used when logging in.

**3. Choose a Password \*** **Verify Password**

Must be at least 8 characters long.

Vergleicht doch mal das Formular auf der linken Seite mit dem auf der rechten Seite. Beide haben eine ähnliche Anzahl von Feldern, und doch ist das rechte viel leichter zu handhaben.

Wir alle hassen es, lange und komplizierte Formulare auszufüllen, da diese uns langweilig, überfordernd und schwierig zu überprüfen erscheinen. Doch wenn Ihr diese Formulare in verschiedene Schritte unterteilt und einen Fortschrittsbalken anzeigt, dann erscheinen die Dinge plötzlich sehr viel einfacher.

Es ist das Gesetz der Vereinfachung — die Menschen werden eher 10 kleine Arbeitsschritte als einen riesigen Arbeitsschritt durchführen. Kleine Aufgaben wirken weniger einschüchternd und geben uns jedes Mal ein Gefühl der Bestätigung, wenn wir sie fertiggestellt haben.

User-Interface-Design ist eine sensible und sehr verantwortungsvolle Aufgabe. Die oben stehenden Gesetze werden Euch dabei helfen, diese Aufgabe besser zu erfüllen, und wenn ihr Euch entscheiden solltet, diese Gesetze zu brechen, dann vergewissert Euch vorher, dass Ihr einen sehr guten Grund dafür habt.

Quelle:

<https://99designs.at/blog/design-kreativitaet/user-interface-design/>



# Grundlegende Richtlinien für gute User Interfaces



Gute Benutzerführung wird nicht nur durch eine Ansammlung grafischer Symbole erreicht. (Bild: gemeinfrei/Pixabay / [CC0](#))

Damit das User Interface, kurz UI, von den Kunden akzeptiert wird, müssen einige zentrale Punkte in Betracht gezogen werden. Aber wie erreicht man das Ziel, wenn ein UI/UX-Experte fehlt?

Themen wie Benutzerfreundlichkeit respektive Benutzererfahrung (User Experience) sind keine Erfindung der Neuzeit. Schön früh erkannten Menschen, dass durch die Erfahrung der Nutzer ein System in seiner Funktion optimiert werden konnte. Ein gutes Beispiel dafür ist die Zeitung. Seit ihrer Erfindung im 16. Jahrhundert durchlief die Zeitung einige Veränderungen und Anpassungen, bis sie so wurde, wie wir sie heute kennen. Besonders im Format, aber auch im Design wurde und wird die Zeitung stetig weiterentwickelt.

Die User-Interface-Entwicklung verläuft sehr ähnlich. Zwei der Personen, die den Entwicklungsprozess massgeblich geprägt haben, sind Ben Shneiderman, der 1986 die «8 goldenen Regeln des Interface-Designs» niederschrieb. Und Jakob Nielsen, der 1994 eine Heuristik zur Überprüfung der Benutzerfreundlichkeit entwickelte.

## Die 8 goldenen Regeln des Interface-Designs

1. **Strebe nach Konsistenz:** In ähnlichen Situationen sollten die Aktionssequenzen immer konsistent sein. Z.B die Position eines Cancel Buttons, Layout, Farbe, Schriftart etc.
2. **Sorge für universelle Bedienbarkeit:** Erkennen der Anforderungen der verschiedenen Benutzer. Die verschiedenen Benutzer können sich in Alter, Erfahrung und allfälligen Behinderungen unterscheiden. Das System respektive Programm soll durch jeden Benutzer schnell und einfach bedient werden können. Zum Beispiel können für Experten Abkürzungen und für ungeübte Benutzer Hilfestellung angeboten werden.
3. **Biete informative Rückmeldungen:** Genügend Feedback über laufende Funktionen oder den Systemstatus muss stets gewährleistet sein.
4. **Entwerfe abgeschlossene Dialoge:** Hiermit soll erreicht werden, dass dem Benutzer bewusst wird, wann eine Aktion, Funktion oder Befehlskette gestartet wird und wann sie abgeschlossen ist. Zum Beispiel führt eine Shop-Website den Benutzer vom Selektieren des Produkts bis zum Zahlen der Ware. Beendet wird der Vorgang mit einer Bestätigungsseite, wonach die Transaktion vollständig und abgeschlossen ist.
5. **Biete einfache Fehlerbehandlung:** Der Benutzer darf nicht verzweifeln, wenn ein Fehler auftaucht. Nach Möglichkeit sollten verständliche Informationen über potentielle Ursachen abgegeben werden. Sehr wichtig ist es, einen Ausweg anzubieten, um in den normalen System- respektive Programmbetrieb zurückzugelangen.
6. **Lass die einfache Umkehrung von Aktionen zu:** Dem Benutzer soll die Möglichkeit gegeben werden, getätigte Aktionen wieder rückgängig zu machen.
7. **Vermittle ein Gefühl der Kontrolle:** Der Nutzer sollte immer das Gefühl haben, Kontrolle über das System respektive Programm zu haben. Kurz gesagt, der Benutzer agiert und das System reagiert und nicht umgekehrt.

8. **Entlaste das Kurzzeitgedächtnis:** Es sollte vermieden werden, dass der Benutzer sich zu viele Informationen merken muss, damit er zum Beispiel nachfolgende Aktionen auf einer nächsten Seite ausführen kann. Als Richtlinie gilt, dass der Mensch sich zwischen fünf bis sieben verschiedene Informationen sicher merken kann.

## Heuristik zur Überprüfung der Benutzerfreundlichkeit

Auch sehr nützlich für die UI-Entwicklung sind die 10 Usability-Heuristiken, die Jakob Nielsen 1994 niederschrieb:

1. **Sichtbarkeit des Systemstatus:** Genügend Feedback über laufende Funktionen oder den Systemstatus muss stets gegeben sein.
2. **Verknüpfung zwischen dem System und der realen Welt:** Es ist hilfreich, entsprechende Konzeptmodelle zu erarbeiten, die auf einer Analogie aus der realen Welt basieren. Dadurch ist die Funktionsweise für den Benutzer viel schneller nachvollziehbar oder sogar vorhersehbar.
3. **Kontrolle und Freiheit des Benutzers:** Der Benutzer sollte immer das Gefühl haben, die Kontrolle auszuüben. Dennoch sollte man dem Benutzer nicht grenzenlose Freiheit einräumen, da er ansonsten überfordert sein und Fehler machen könnte.
4. **Konsistenz und Standards:** Man sollte immer auf entsprechende interne und externe Konsistenz achten. Darüber hinaus sind, ausser in speziellen Ausnahmefällen, immer bestehende Standards zu berücksichtigen.
5. **Fehler-Vorbeugung:** Potentielle Fehlerquellen sollten frühzeitig eliminiert werden. Der Benutzer muss zudem ausreichend Anleitung erhalten, um keine Fehler zu verursachen.
6. **Wiedererkennen vor Überlegen:** Bevor der Benutzer nachdenken muss, wie eine Funktion zu bewerkstelligen oder wo sie zu finden ist, sollte er sie direkt anhand des Interfaces wiedererkennen können.
7. **Flexibilität und Effizienz der Benutzung:** Dies bezieht sich auf die nötige Balance zwischen Abkürzungen (engl. shortcuts) für Experten und ausführlicher Hilfestellung für Anfänger.
8. **Ästhetisches und minimalistisches Design:** Ein Design sollte stets ästhetisch ansprechend, aber dennoch minimalistisch sein, um unnötige Verwirrung und Übersichtsverlust zu vermeiden.
9. **Benutzerhilfe und Support bei Fehlern:** Fehlermeldungen müssen klar und verständlich sein. Sie müssen das Problem und eine mögliche Lösung aufzeigen.
10. **Hilfe und Dokumentation:** Auch wenn es besser ist, ein System ohne Dokumentation betreiben zu können, so ist es manchmal doch nötig, eine Hilfe und eine Dokumentation anzubieten. Diese sollten einfach zu durchsuchen sein und sollten Schritte hin zur Lösung eines Problems aufzeigen.

## Layouts

Früher wurden die Applikationen zumeist mit einem starren Layout fix für eine bestimmte Bildschirmgröße entwickelt. Durch die zunehmende Vielfalt an verschiedenen Displaygrößen und -



aufösungen entstand nach und nach das Problem, dass die Applikation für jeden neu gewählten Bildschirm auch neu übersetzt werden muss.

Um dieses Problem zu beheben, sind neben dem fixen Layout noch zwei Mechanismen zur Gestaltung variabler Layouts gängig.

## Folgende Layouts werden unterschieden:

- **Static/Fixed:** Das Layout ist fix auf eine bestimmte Displaygröße eingestellt. Es ist sehr einfach zu implementieren, benötigt aber auf der anderen Seite eine manuelle Anpassung für jede weitere Displaygröße.
- **Fluid/Liquid:** Das Layout passt sich der Bildschirmgröße in der Breite und Höhe an. Die Elemente werden entweder vergrößert oder verkleinert. Für Bildschirme, die sich in der Größe nur geringfügig unterscheiden, kann diese Art von Design gut eingesetzt werden. Bei deutlichem Bildschirmunterschied werden die Elemente zu sehr gestreckt. Dann muss die Applikation manuell im Code angepasst werden.
- **Responsive:** Das Layout passt sich auch hier der Bildschirmgröße an. Es ändert sich aber nicht nur die Größe der Elemente, sondern auch die Anordnung der Inhalte in den Elementen. Zusätzlich werden Elemente ein- oder ausgeblendet bzw. in ein Menü verschoben. Das Implementieren und das Testen eines solchen Layouts sind deutlich aufwändiger als bei den anderen Arten. Dafür passt sich die Applikation dynamisch laufend an jede gewünschte Bildschirmgröße an.

Immer häufiger wünschen sich Kunden aus dem industriellen Umfeld User Interfaces, wie sie sie von Smartphones her kennen. Diese Interfaces müssen effizient und effektiv zu bedienen sein, damit die Kunden zufrieden sind.

## Literatur- und Quellenverzeichnis

<https://www.cs.umd.edu/users/ben/goldenrules.html><http://www.nngroup.com/articles/ten-usability-heuristics/> Responsive Design Example: <http://www.liquidapsive.com/>

Quelle:

<https://www.embedded-software-engineering.de/grundlegende-richtlinien-fuer-gute-user-interfaces-a-554405/>