

Von wegen flache Hierarchieebenen

Wenn man visuelles Design (oder eben in unserem Fall UI-Design) in einem Satz zusammenfassen möchte, so würde der vielleicht in etwa so lauten: „Dinge so gestalten, dass sie gut aussehen“. Schon hat man als Frontend-Code-Jockey keinen Bock mehr drauf. Da sind sie wieder, das angeborene künstlerische Talent und die immerwährende, jederzeit abrufbare Kreativität. Allerdings ist es erstaunlich, dass einer der wichtigsten Faktoren für ein „gutes Aussehen“ überhaupt nichts mit künstlerischen Fähigkeiten und kreativer Gestaltung zu tun hat – nämlich die visuelle Hierarchie.

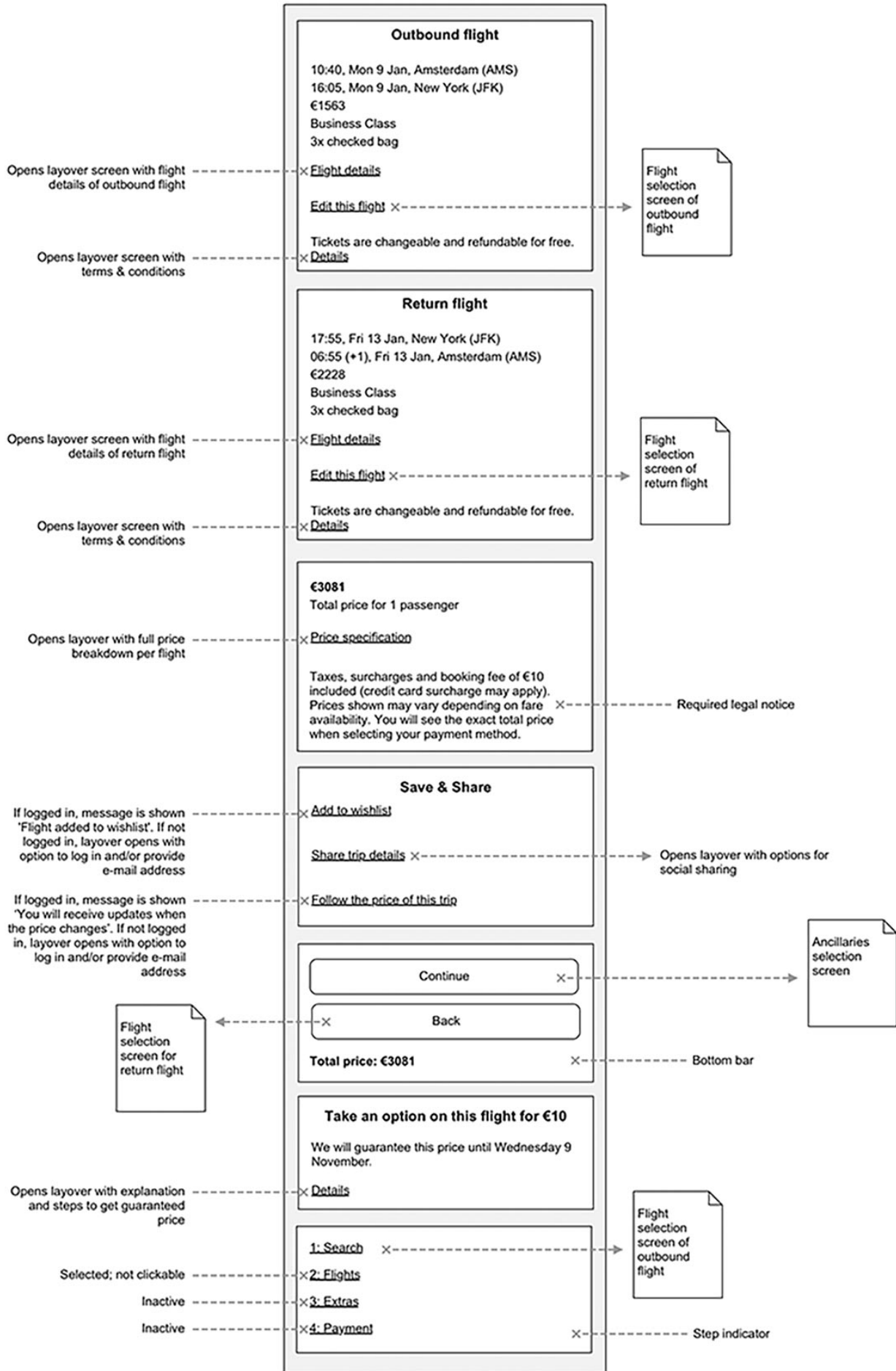
Die visuelle Hierarchie bezieht sich darauf, wie wichtig die Elemente einer Benutzeroberfläche im Verhältnis zueinander erscheinen, und sie ist das effektivste Werkzeug, um etwas „gestaltet“ wirken zu lassen. Wenn alles auf einer Benutzeroberfläche um Aufmerksamkeit konkurriert, wirkt sie laut und chaotisch, in etwa wie ein Platz voller digitaler Werbebotschaften und Marktschreier. Viel Lärm und wenig Signal. Bei all dem (visuellen) Chaos ist es uns nicht klar, was wirklich von Bedeutung ist. Zu dieser Thematik gibt es viele lesenswerte Publikationen, u. a. [1].

Achtung, hier wieder aufmerksam lesen, denn ich verrate euch ein Geheimnis, und zwar das Geheimnis des Designs: Trennt das Primäre vom Sekundären und vom Tertiären oder in anderen Worten, fragt euch, was ist das Wichtigste, was ist das Zweitwichtigste und was das Drittwichtigste. Wenn wir bewusst sekundäre und tertiäre Informationen zurückstellen und uns darum bemühen, die wichtigsten Elemente hervorzuheben, wirkt das Ergebnis sofort ansprechender, auch wenn sich das Farbschema, die Schriftart und das Layout nicht geändert haben.

Ein super Hilfsmittel dafür sind sogenannte Priority Guides. Priority Guides werden als die Content-First-Alternative zu Wireframes gehandelt. Da man bei Wireframes irgendwie auch schon malen können muss, bei Priority Guides aber nicht, macht das einem die Dinge schon gleich sympathisch (übrigens auch dem Projektleiter, da sie in ihrer Erstellung weniger aufwendig als Wireframes sind). Einfach ausgedrückt enthält ein Priority Guide Inhalte und Elemente für einen mobilen Bildschirm (also auch Mobile First), sortiert nach Wichtigkeit von oben nach unten und ohne jegliche Layoutvorgaben. Die Hierarchie basiert auf der Relevanz für die Nutzer, wobei die Inhalte, die für die Befriedigung der Nutzerbedürfnisse und die Unterstützung der Ziele der Nutzer (und des Unternehmens) am wichtigsten sind, weiter oben stehen.

Das Format eines Priority Guides ist nicht festgelegt: Er kann digital sein (Text-, Office-, Figma-Dateien) oder er kann physisch sein, mit Papier und Post-its. Am wichtigsten ist, dass ein Priority Guide von Anfang an automatisch Content First ist und den Nutzer:innen den besten Nutzen bietet. Ein Beispiel sehen Sie in **Abbildung 1** mit einem Priority Guide für die Flugbuchungsseite einer Fluggesellschaft aus „Priority Guides: A Content-First Alternative to Wireframes“ [2].

Book a flight



bb. 1: Ein detaillierter digitaler Priority Guide für eine Flugbuchungsseite

Anmerkungen sind ein wichtiger Bestandteil von Priority Guides, da sie die Funktionen und das Verhalten der Seite erklären, die Komponententypen benennen und die Priority Guides einer Seite mit den Priority Guides anderer Seiten verknüpfen. In diesem Beispiel wird beschrieben, was passiert, wenn ein Benutzer auf eine Schaltfläche oder einen Link klickt.

Bevor wir noch irgendetwas gemalt haben, können wir mit diesem Tool beginnen, das Wichtige vom Unwichtigen zu trennen und uns dem Attribut „gestaltet“ weiter annähern. Bei der Erstellung von Priority Guides konzentriert man sich automatisch darauf, die Probleme von Menschen zu lösen, ihre Bedürfnisse zu befriedigen und ihnen zu helfen, ihre Ziele zu erreichen. Die Benutzeroberfläche ist immer mit Inhalten gefüllt, die eine Botschaft vermitteln oder dem Benutzer weiterhelfen. Indem wir uns auf den Inhalt konzentrieren, entscheiden wir im Sinne unserer Benutzer. Genau dieser Fokus auf den Content bringt uns zum Design Algorithm [3], der uns als Frontend-Schaffende näher an die uns vertraute Denkweise bringt (**Abb. 2**). Der einfachste Weg zu einem „gestalteten“ Design und einem guten Ergebnis führt über vier Schritte:

1. *Der Inhalt*: In diesem Schritt müssen wir erstmal verstehen, mit welcher Art von Inhalten wir arbeiten, welche Texte, Bilder und andere Komponenten es gibt – kurz gesagt: Inventur machen.
2. Es folgt *die Struktur*: das Primäre vom Sekundären zu trennen, zu verstehen, in welche Blöcke der Inhalt aufgeteilt werden kann und welche Beziehungen zwischen den Blöcken bestehen.
3. Jetzt geht es ans *Layout*: Wir erstellen ein Layout, das die Idee widerspiegelt und den Inhalt und seine Struktur am besten darstellt; die Art und Weise, wie das Design

funktionieren soll.

4. Zuletzt folgt *der Stil*: Design, Stil, Farbe und andere visuelle Details, die Art und Weise, wie das Design aussehen soll. Die Farbe kommt erst zum Schluss. Das Großartige an diesem Algorithmus ist, dass das „gestaltete“ Design bereits nach dem dritten Schritt wirklich funktional und zu testen ist. Wie cool ist das denn?

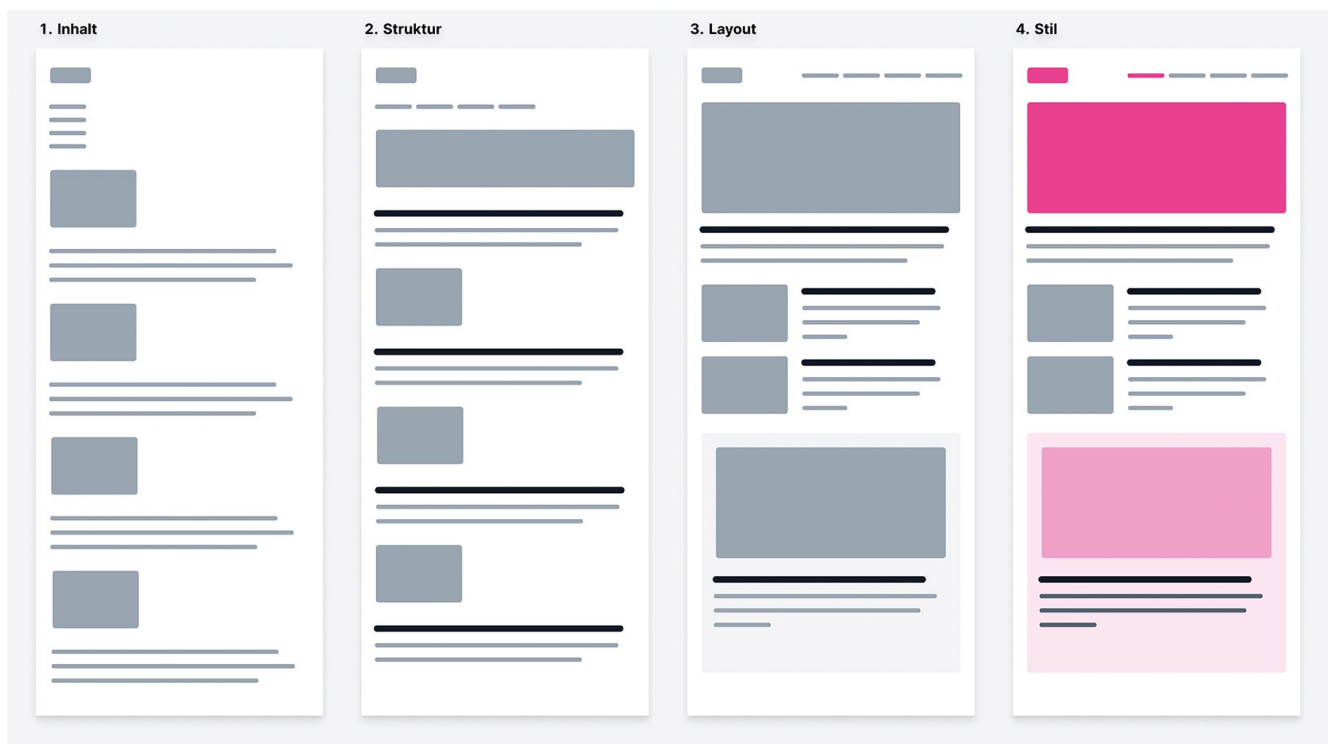


Abb. 2: Der Designalgorithmus eignet sich für jede Art von Projekt

Schauen wir jetzt mal auf das „echte“ UI und versuchen einmal, was man mit Hierarchie so alles retten kann. Beim Separieren von Wichtigem und Unwichtigem kommt einem sofort die Größe in den Sinn, aber Größe ist nicht alles. Es ist ein Fehler, sich nur auf die Schriftgröße zu verlassen, um die Hierarchie zu kontrollieren, da das oft dazu führt, dass der primäre Inhalt zu groß und der sekundäre Inhalt zu klein ist. Effektiverer Stellschrauben sind der Schriftschnitt und die Farbe.

600 ist das neue 24

Ein primäres Element fetter zu machen, erlaubt es weiterhin, eine vernünftige und verhältnismäßige Textgröße zu verwenden. Ebenso macht die Verwendung einer sanfteren Farbe für den sekundären Begleittext anstelle einer kleinen Schriftgröße deutlich, dass der Text zweitrangig ist, während die Lesbarkeit weniger darunter leidet. Farbe und Schriftschnitt statt purer Größe. Man könnte hier auch von einem Algorithmus der Farbe sprechen (**Abb. 3**). Wir beschränken uns auf zwei oder drei Farben:

- eine dunkle Farbe für primäre Inhalte wie z. B. die Überschrift eines Artikels
- ein Grau für sekundäre Inhalte wie das Veröffentlichungsdatum des Artikels
- ein abgestuftes helleres Grau für tertiäre Inhalte, z. B. den Copyrighthinweis in der Fußzeile

In ähnlicher Weise sind zwei Schriftstärken für eine Benutzeroberfläche vollkommen ausreichend:

- ein normaler Schriftschnitt (400 oder 500, je nach Schriftart) für den generellen Text
- ein fetterer Schriftschnitt (600 oder 700) für Text, der in den Vordergrund gerückt werden soll

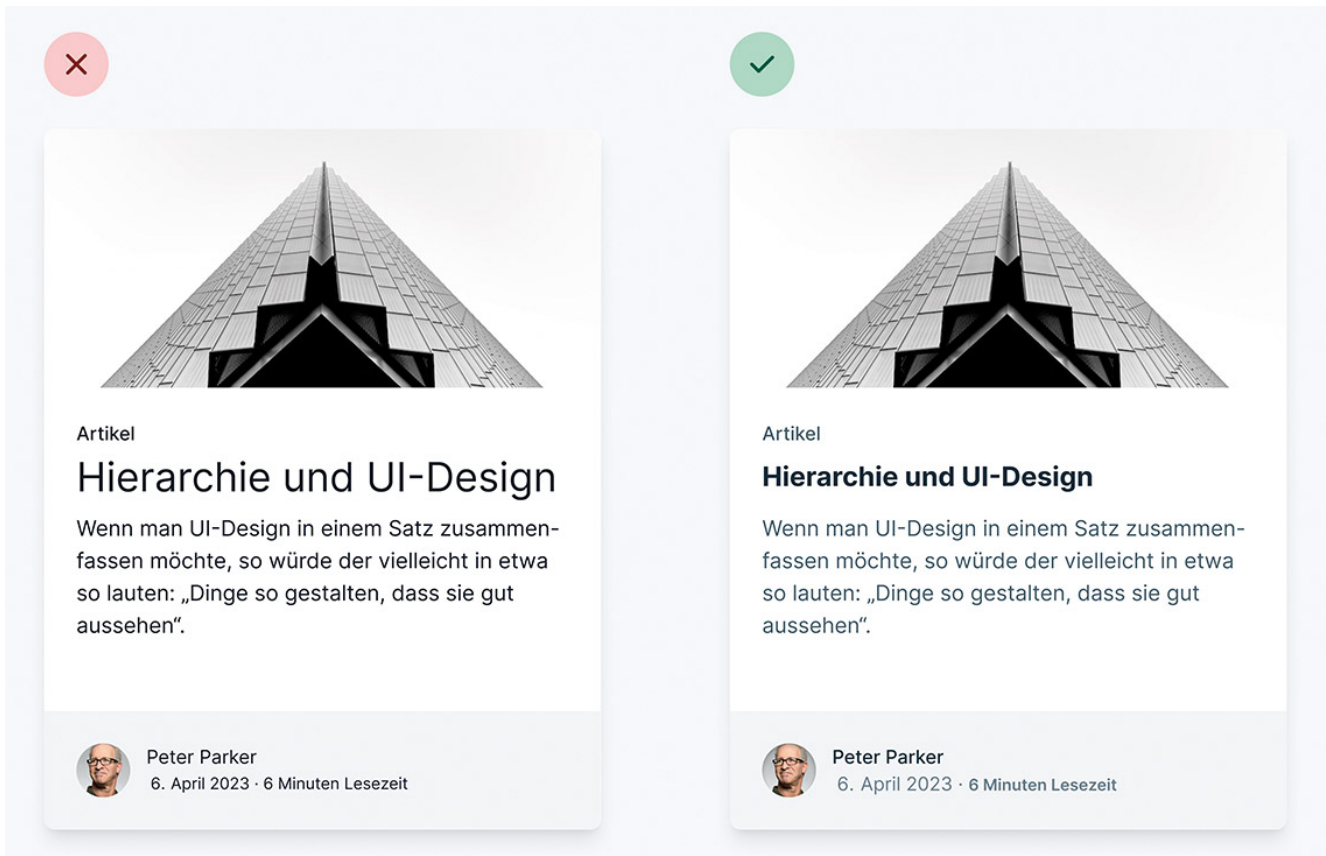


Abb. 3: Gestaltung mit Hilfe der Schriftgröße oder der Farbe

Pro-Tipp: Wir sollten Schriftschnitte unter 400 für UI-Arbeiten tunlichst vermeiden. Sie eignen sich zwar für große Überschriften, sind aber in kleineren Größen zu schwer zu lesen.

Stärkung durch Abschwächung

Manchmal befinden wir uns in einer Situation, in der ein Element einer Benutzeroberfläche gegenüber den umgebenden Elementen nicht ausreichend hervorgehoben ist. Allerdings gibt es auch auf den ersten Blick nichts, was wir noch hinzufügen oder womit wir es „vercoolern“ (Danke, Emilia, für die Leihgabe) könnten, um es stärker hervorzuheben. Nachfolgendes Beispiel zeigt ein Menu, dessen aktives Element bereits durch eine andere Farbe betont ist, aber dennoch kann es sich nicht visuell gegen die inaktiven Elemente durchsetzen.

In solchen Situationen sollten wir nicht versuchen, dieses eine Element, auf das wir die Aufmerksamkeit lenken wollen,

noch mehr hervorzuheben, sondern wir sollten einen Weg finden, die Elemente, die damit in Konkurrenz stehen, weniger auffällig zu machen.

Im folgenden Beispiel wird das dadurch erreicht, dass die inaktiven Elemente eine weichere Farbe als das aktive Element bekommen und damit visuell den Platz für das aktive Element frei machen (**Abb. 4**).

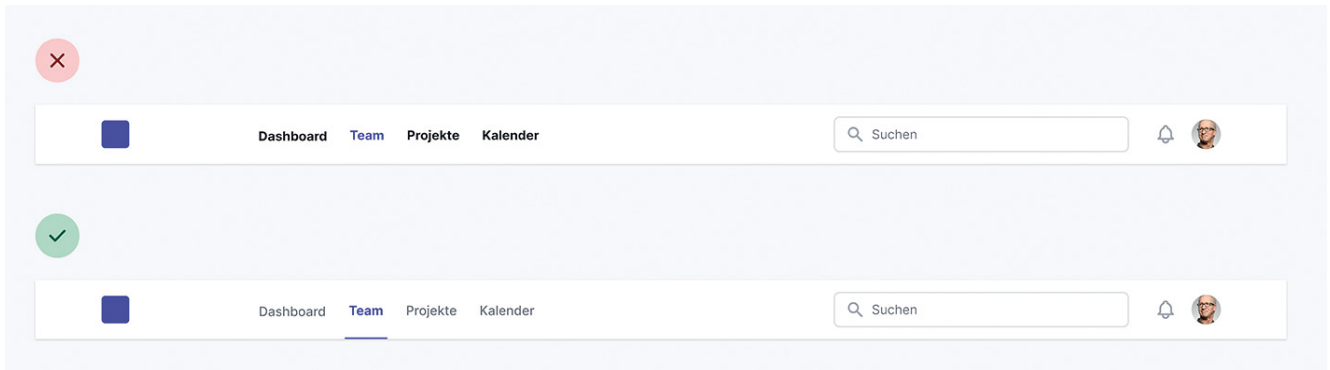


Abb. 4: Wir geben den inaktiven Elementen eine weichere Farbe und senken den Kontrast

Diese Denkweise können wir auch auf größere Teile eines UIs anwenden. Wenn zum Beispiel eine Seitenleiste visuell mit dem Hauptinhalt konkurriert, sollten wir ihr keine Hintergrundfarbe geben, sondern den Inhalt der Seitenleiste direkt auf dem Hintergrund der Webseite platzieren (**Abb. 5**).

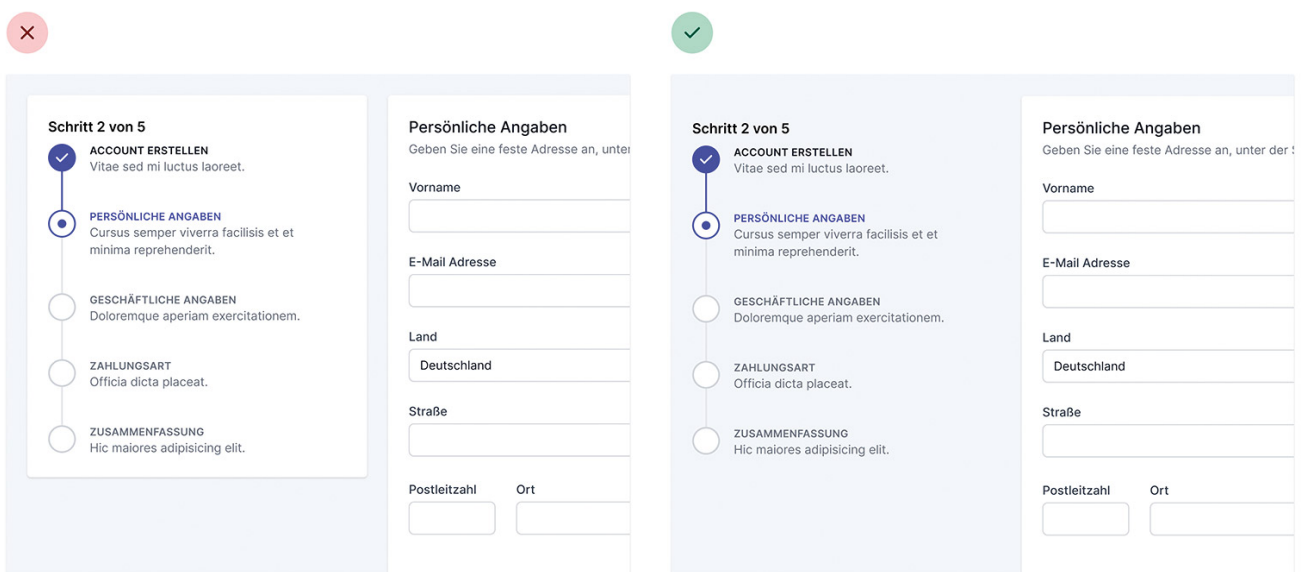


Abb. 5: Das Prinzip der Abschwächung lässt sich auch bei

größeren UI-Organismen einsetzen

Label mich nicht voll

Ich weiß, Labels sind heilige Kühe und da hört bei vielen der Spaß auf. Bei Formularen ist das auch absolut okay, nur in Card Controls, Produktbeschreibungen o. Ä. dürfen wir uns gerne mal Gedanken darüber machen, warum eigentlich meistens der Key einen fetteren Schriftschnitt hat als der Value, der uns eigentlich interessiert.

Es gibt tatsächlich nur eine Situation, die rechtfertigt, dass das Label eine deutlichere Hervorhebung als der eigentliche Wert hat, und zwar wenn wir wissen, dass unsere Betrachter nach der Beschriftung suchen werden. Nur dann ist es sinnvoll, die Beschriftung anstelle der Daten zu betonen. Das ist z. B. bei den technischen Daten eines Produkts der Fall. Wenn ein Mensch die Abmessungen eines Regals sucht, wird meistens nach Begrifflichkeit wie „Länge“, „Breite“ oder „Tiefe“ gesucht, nicht aber nach „41,5 cm“. Auch in solchen Fällen ist der Wert immer noch die primäre Information. Daher ist es ausreichend, eine dunklere Farbe für die Beschriftung und eine etwas hellere Farbe für den Wert zu verwenden.

Daneben halten wir uns bei der Verwendung von Labels fast dogmatisch an das Formatschema Beschriftung, Doppelpunkt, Wert oder für Informatiker:innen Key, Colon, Value. In vielen Situationen können wir jedoch allein anhand des Formats erkennen, worum es sich bei einem Eintrag handelt. Nochmal zur Deutlichkeit: Wir reden nicht von Formularen, das ist eine andere Liga.

Aber wieder zurück zu unseren Datensätzen. Zum Beispiel ist m.muster@beispielwebsite.de eine E-Mail-Adresse, +49 (0)69 630 089 0 eine Telefonnummer und 1,99 € eine Preisangabe. Wenn das Format nicht ausreicht, hilft uns eben der Kontext und der ist bekanntlich König. Ein Eintrag wie „Frontend-Entwickler“ unter dem Namen eines Mitarbeiters auf

der Teamseite ist ein sicheres Indiz dafür, dass es sich um einen Mitarbeitenden handelt, der Frontends für Webapplikationen entwickelt. Ein Label würde hier ziemlich übers Ziel hinausschießen. Deutlich sehen Sie das in den zwei Beispielen aus **Abbildung 6**.

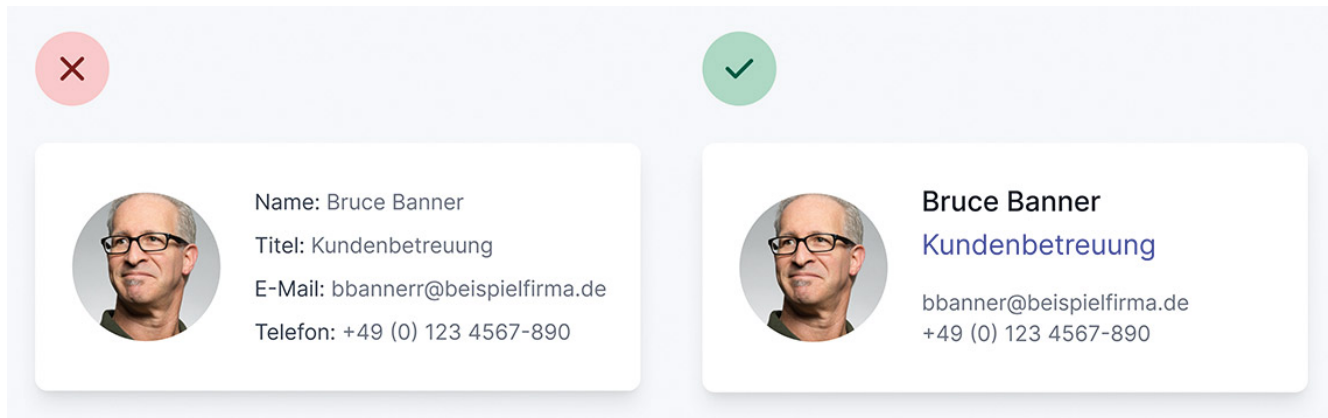


Abb. 6: Labels können oft die schnelle Erfassung von Inhalten beeinträchtigen

Wenn Datensätze ohne Beschriftungen dargestellt werden können, ist es viel einfacher, wichtige oder identifizierende Informationen hervorzuheben, wodurch das UI benutzerfreundlicher und gleichzeitig intuitiver wird. Das zwanghafte Festhalten an der LDW-Struktur (Label Doppelpunkt Wert) führt oftmals auch zu hakeligen Formulierungen auf unserer Oberflächen – „Auf Lager: 15“. Ihr merkt an dieser Stelle selbst, dass es sich komisch liest. Welche Formulierung würden wir erwarten? Ja eben: „Es sind noch 15 auf Lager“. Die Taktik, einen Wert in einen erläuternden Text einzubetten ist durchaus legitim und liest sich flüssiger. In Kombination mit einem hervorhebenden Schriftschnitt für den eigentlichen Wert können zwei Fliegen mit einer Klappe geschlagen werden – also gerne mal das LDW-Dogma mit einem (kurzen) lesbaren Satz aufbrechen (**Abb. 7**).

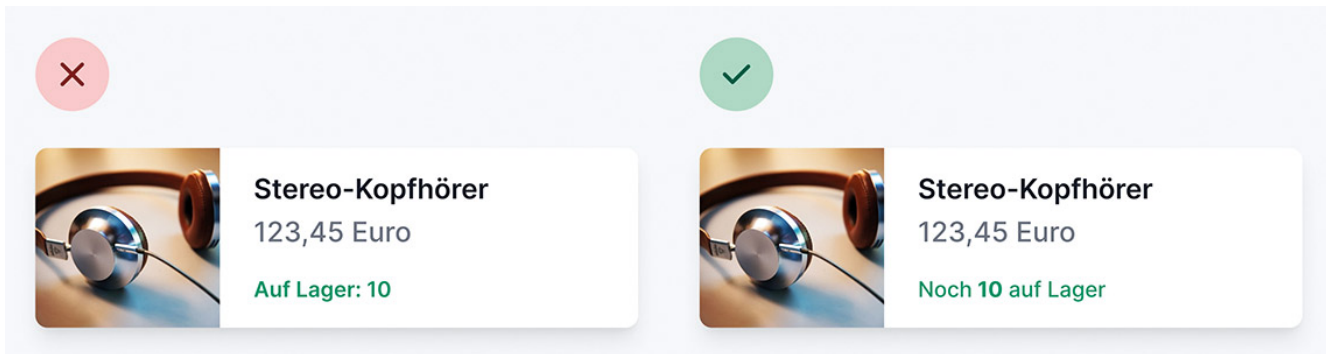


Abb. 7: Besser, so liest es sich nicht wie Kaugummi

Um noch einmal den Bogen zum Beginn unseres Labelexkurses zu schlagen: Labels sind sekundärer Content! Es gibt Situationen oder Darstellungen, in denen wir wirklich eine Beschriftung benötigen, etwa wenn wir mehrere vom Typ ähnliche Daten anzeigen und diese auch auf den ersten Blick leicht ablesbar sein müssen, wie z. B. in einem Dashboard. Dann sollten wir aber das Label auf jeden Fall als sekundären und unterstützenden Content betrachten. Die Daten sind die primäre, wichtige Information, die Beschriftung dient nur der Übersichtlichkeit. In einem solchen Fall verkleinern wir das Label, verringern den Kontrast, oder verwenden einen leichteren Schriftschnitt oder eine Kombination aus allen drei Möglichkeiten (**Abb. 8**).

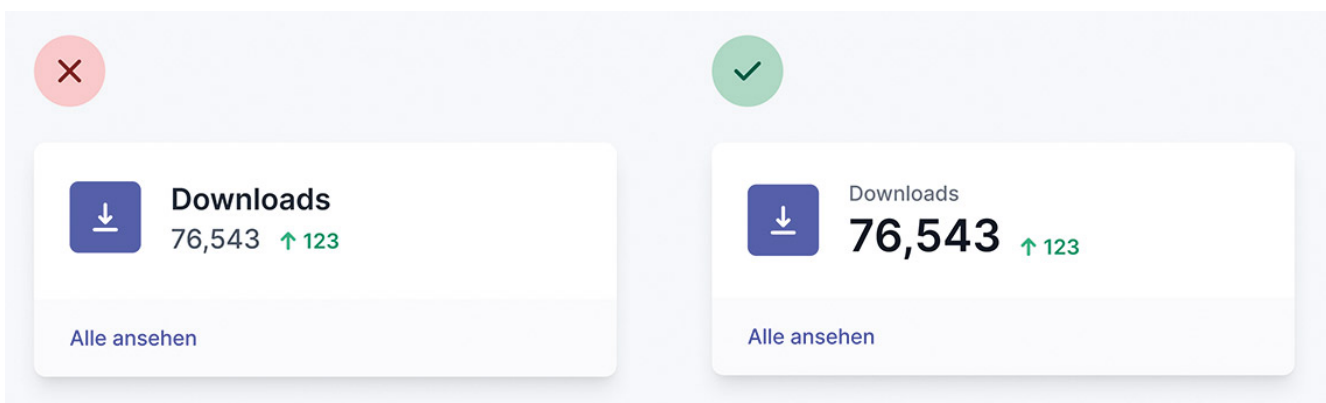


Abb. 8: Labels sind sekundär, gerade in Dashboards

Visuelle Hierarchie ist nicht

gleich Dokumentenhierarchie

Bei der Erstellung von Webinhalten sollten wir immer auf semantisches Markup zurückgreifen. Das bedeutet, dass wir unsere Inhalte strukturieren und z. B. Überschriften mit den geläufigen HTML-Tags wie *h1*, *h2* oder *h3* kennzeichnen. Out of the Box versehen die Webbrowser die Überschriftentags mit entsprechenden visuellen Eigenschaften. So werden mit aufsteigender Zahl hinter dem *h* den Überschriften immer kleinere Schriftgrößen zugewiesen – *h1* groß und *h6* klein. Grundsätzlich ist das schon mal super, aber in einem „gestalteten“ UI kann dieses Verhalten der Browser auch schon mal zu Fehlinterpretationen führen. Ein *h1*-Tag für eine Überschrift wie „Neues Benutzerprofil“ ist semantisch absolut sinnvoll. Weil wir allerdings „evolutionär“ bzw. „erlernt“ den Drang verspüren, *h1*-Überschriften immer groß zu gestalten, kann es leicht dazu kommen, dass wir der Überschrift mehr Prominenz verschaffen als sie visuell eigentlich brauchen würde (**Abb. 9**).

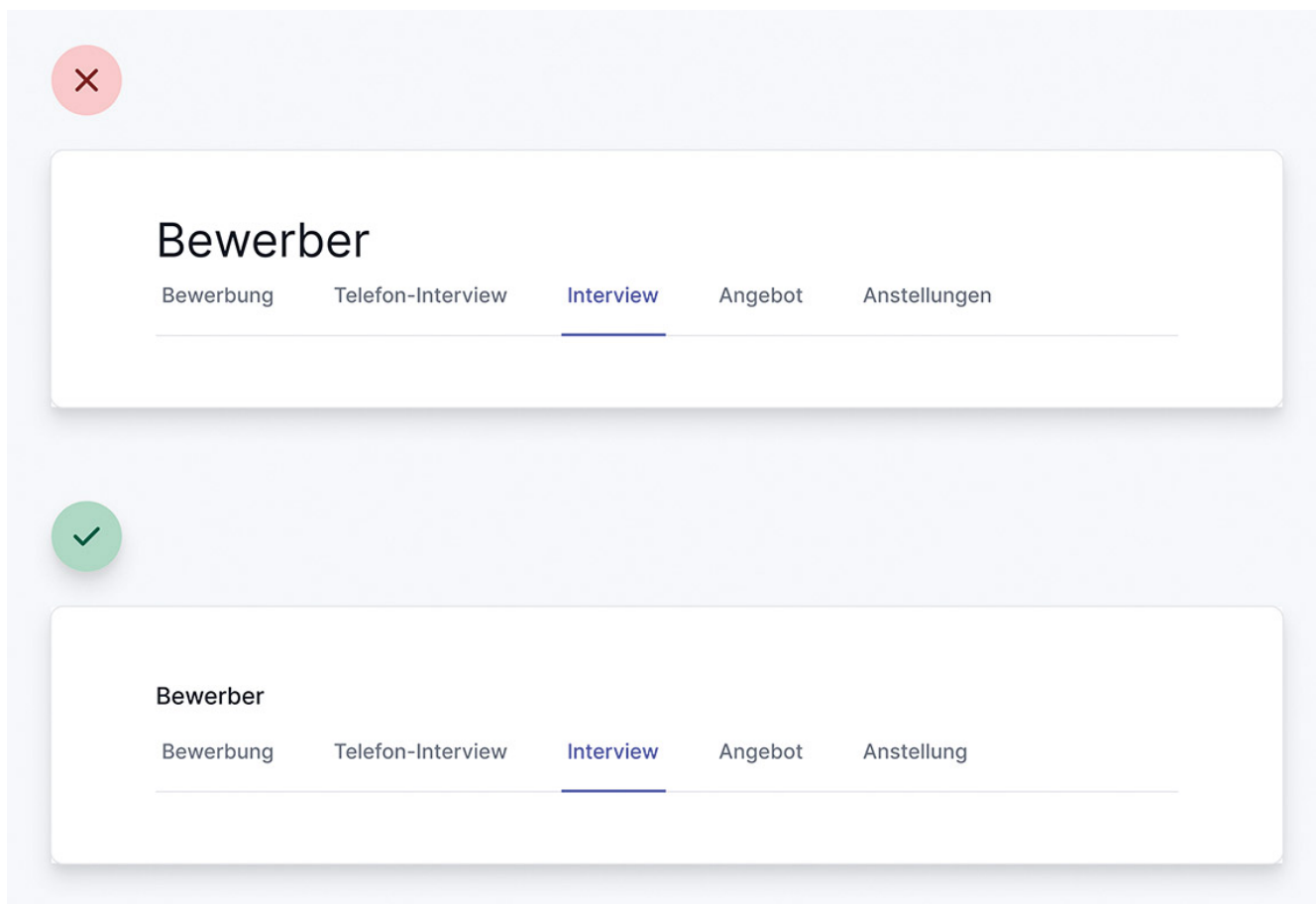


Abb. 9: Abschnittstitel sollten nicht die ganze Aufmerksamkeit auf sich ziehen

Ähnlich wie Labels unterstützen Überschriften den Inhalt und sollten daher nicht die ganze Aufmerksamkeit auf sich ziehen. Normalerweise sollte der Inhalt des Abschnitts im Mittelpunkt stehen, nicht der Titel, also sollten Überschriften in den meisten Fällen verhältnismäßig klein sein. Denn auch dann erfüllen sie noch gut ihre eigentliche gliedernde Funktion. Lassen wir uns beim Designen nicht von den „geerbten“ visuellen Eigenschaften der HTML-Elementen beeinflussen. Wir sollten die Elemente aus semantischen Gründen heraus wählen und sie dann für eine optimale visuelle Hierarchie gestalten.

Dunkel is the new fett

Kontrast eignet sich hervorragend zur Kompensation des visuellen „Gewichts“ eines Elementes. Das Verständnis dieser Beziehung wird unter anderem bei der Arbeit mit Icons wichtig. Genau wie fett gedruckter Text sind auch Symbole (insbesondere Volltonicons) in der Regel ziemlich „schwer“ und nehmen viel Pixelfläche ein. Wenn wir also ein Icon neben einem Text platzieren, wird das Symbol mehr hervorgehoben als der Text. Und jetzt kommt der Haken: Im Gegensatz zum Text kann das „Gewicht“ eines Symbols nicht so einfach verändert werden. Um ein Gleichgewicht herzustellen, muss das Symbol auf andere Weise an Prominenz einbüßen.

Am einfachsten und auch wirksamsten ist es, den Kontrast des Symbols zu reduzieren hin zu einer weicheren Farbausprägung. Diese Technik funktioniert überall dort, wo wir Elemente mit unterschiedlicher visueller Gewichtung an- bzw. ausgleichen müssen. Die Kontrastverringerung lässt schwerere Elemente optisch leichter erscheinen, obwohl sich ihr Gewicht (die eingenommene Pixelfläche) nicht geändert hat (**Abb. 10**).

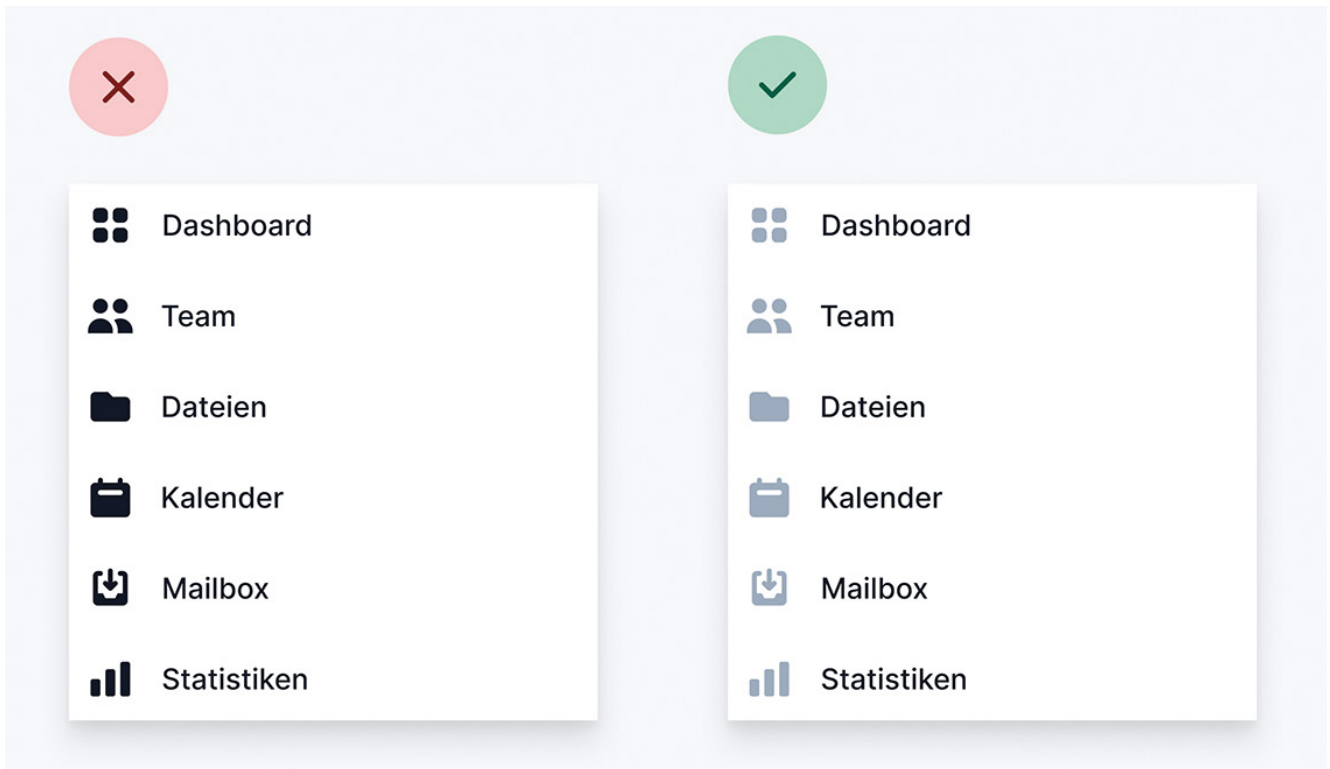


Abb. 10: Verringerung des Kontrasts durch eine weichere Farbe

Genauso wie die Reduzierung des Kontrasts eine Möglichkeit zur Abschwächung visuell gewichtiger Elemente ist, ist die Kontrasterhöhung bzw. die Erhöhung des visuellen Gewichts eine gute Möglichkeit, kontrastärmere Elemente ein wenig zu boosten. Nehmen wir eine ein Pixel dünne Linie mit zu hellem, weichem Grauton. In der einen Version ist die Linie schlecht sichtbar und geht im Design unter, auf der anderen Seite lässt ein dunkler Grauton das Design zu hart erscheinen. Die Lösung ist hier, die Linie ein wenig aufzublähen, indem man ihre Breite vergrößert. Dann können wir unser subtiles Grau behalten und die Linie trotzdem hervorheben, ohne das Design zu sprengen (**Abb. 11**).

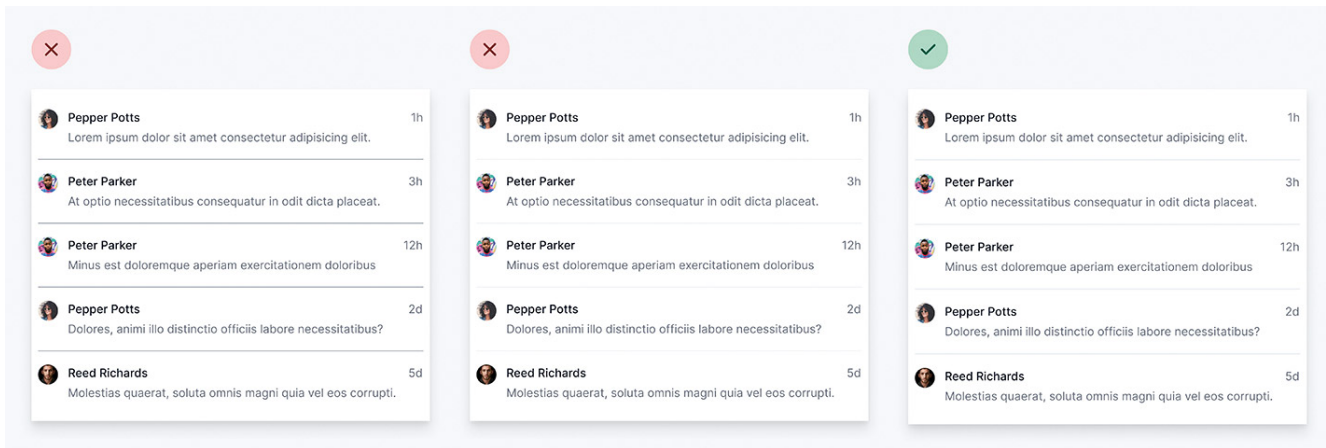


Abb. 11: Etwas dickere Linien helfen, sie zu betonen, ohne dass der subtile Eindruck verloren geht

Semantik spielt die zweite Geige

Semantik kann, wenn es um Benutzerschnittstellen geht, gerne mal ein wenig in die Irre führen, gerade wenn Menschen die Auswahl zwischen mehreren Aktionen (Schaltflächen) haben, die sie in unserem UI ausführen können (und das ist bei UIs bekanntlich meistens der Fall). Hierarchy to the rescue – denken wir an die Priority Guides und stellen uns die Frage, was ist das Wichtigste, das Zweitwichtigste und so weiter. Meistens gibt es nur eine primäre Aktion, mehrere sekundäre und selten ein paar tertiäre Aktionen. Genau an dieser Stelle muss das Design die Wichtigkeit der Aktion widerspiegeln (**Abb. 12**):

- Primäre Aktionen müssen offensichtlich sein, hier auch gerne mal mit Farbe.
- Sekundäre Aktionen sollten deutlich, aber nicht zu auffällig sein. Konturlinien oder Hintergrundfarben mit geringerem Kontrast sind hier eine gute Option.
- Tertiäre Aktionen sollten auffindbar, aber visuell zurückhaltend sein. Die Gestaltung dieser Aktionen als Links ist in der Regel der beste Ansatz.

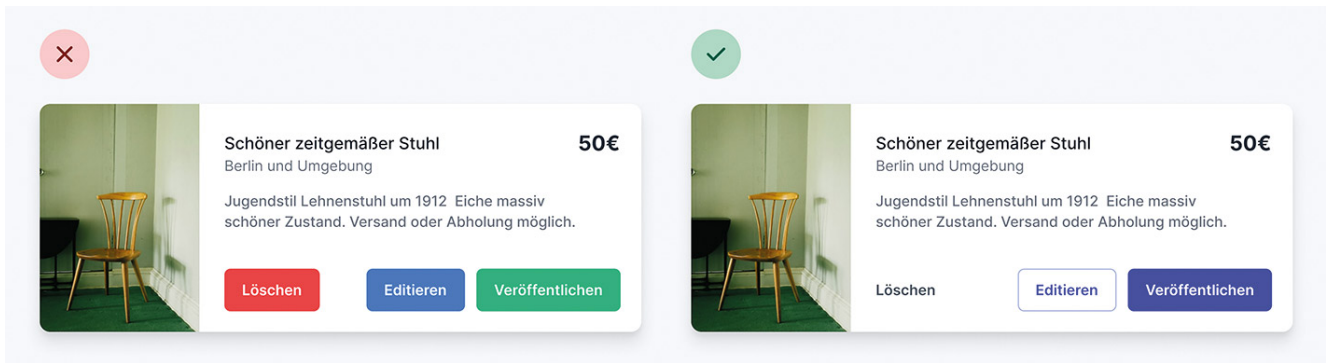


Abb. 12: Wir geben den Benutzenden eine klare Richtung vor

Eine hierarchische Anordnung der Aktionen auf der Seite führt zu einer weniger vollgestopften Benutzeroberfläche, die ihre Aktionen klarer kommuniziert und auch das Papageienoutfit mancher UIs wird vermieden. Eben „don't make me think“ (Abb. 13).

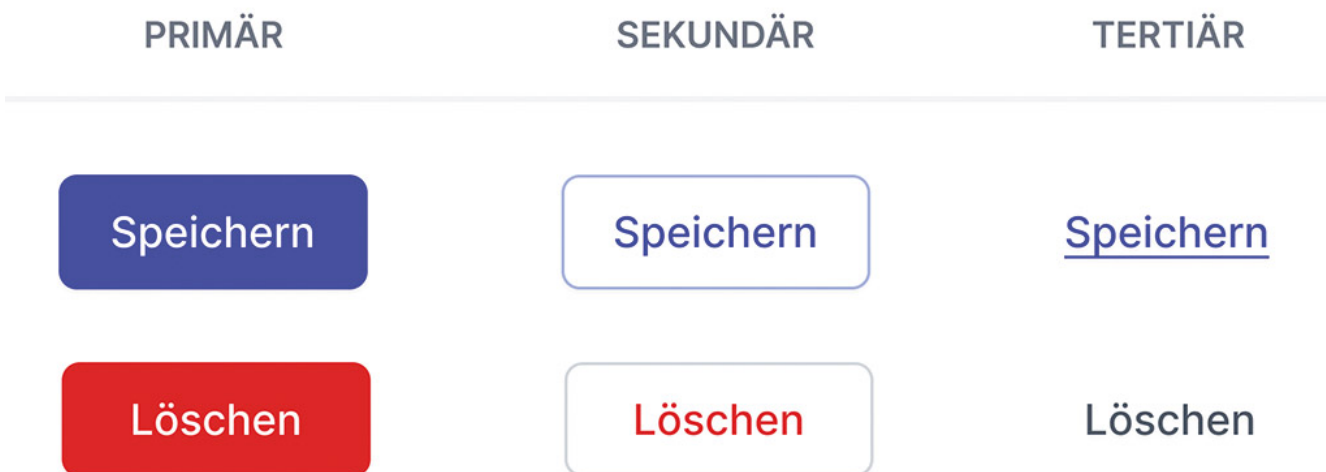


Abb. 13: Aufteilung von Interaktionsmöglichkeiten mit abnehmender visueller Prominenz

Wenn eine Aktion destruktiv ist, z. B. etwas zu löschen, bedeutet das noch nicht automatisch, dass die Schaltfläche gleich groß, rot und fett sein muss. Wenn die destruktive Aktion nicht die primäre Aktion in unserem UI ist, kann es besser sein, ihr eine sekundäre oder tertiäre Ausprägung zuzuweisen. In Kombination mit einer nachfolgenden Bestätigung, bei der die destruktive Aktion die primäre Aktion einnimmt, lässt sich ein stimmiges Design destruktiver Aktionen erzielen und in der Confirm-Message dürfen wir es

auch mit „groß“, „rot“ und „fett“ richtig krachen lassen (Abb. 14).

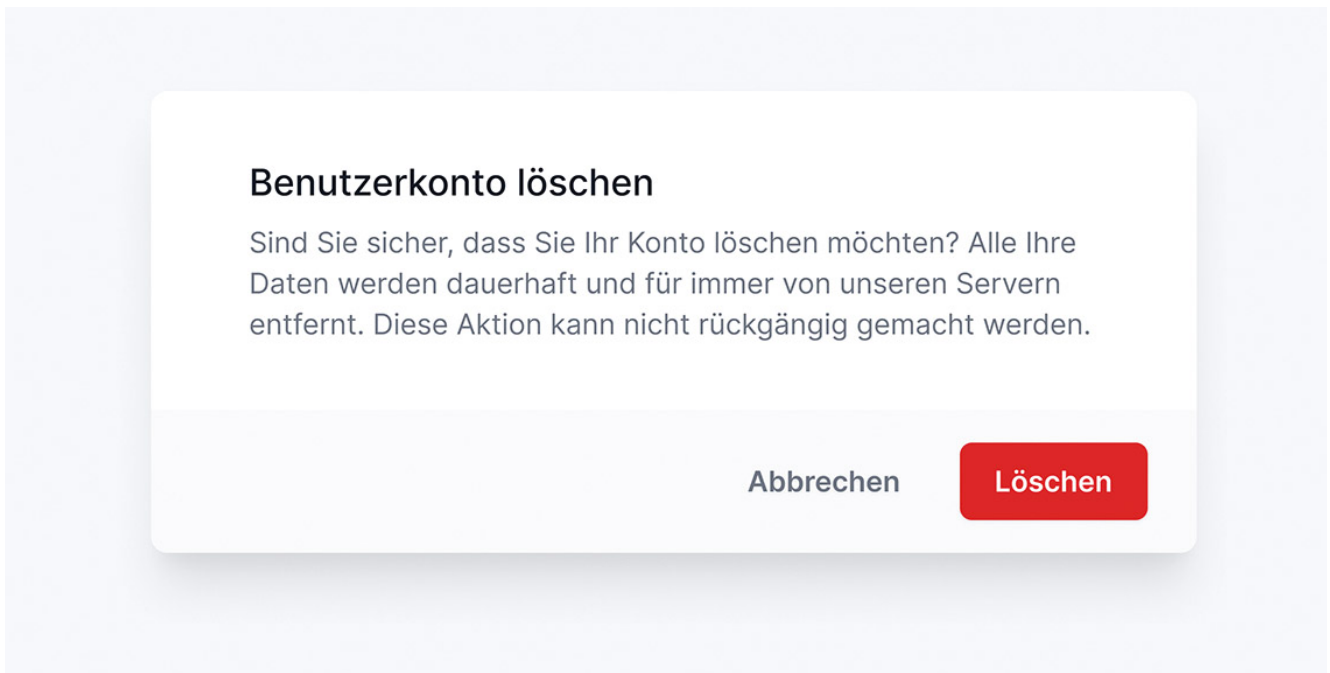


Abb. 14: Bei der Bestätigung darf's auch ordentlich rot sein

Fazit

Seht ihr, (fast) keine Farben, Bilder, Frameworks, nix – nur Gestaltprinzipien und die visuelle Hierarchie. Dennoch sind diese beiden Methoden entscheidend für die Gestaltung einer erfolgreichen und leicht verständlichen Benutzeroberfläche. Wir Webschaffenden im Frontend sollten uns auf jeden Fall mit diesen Konzepten vertraut machen und sie bei der Gestaltung unserer Anwendungen berücksichtigen und gezielt einsetzen, und dann kommt die Farbe ...

Weiterführende Informationen

Es gibt viele Onlineresourcen, die uns dabei helfen können, unsere Skills über Gestaltungsprinzipien und Gestaltungshierarchien zu vertiefen. Nachfolgend einige nützliche Websites und Ressourcen:

- *Smashing Magazine* [4]: Eine Website, die sich auf

Webdesign und -entwicklung spezialisiert hat. Hier finden Sie viele Artikel und Tutorials zu Gestaltungsprinzipien und UI-Hierarchien.

- *UX Planet* [5]: Eine Website, die sich auf UX-Design spezialisiert hat. Hier finden Sie viele Artikel und Tutorials zu UX-Design-Prinzipien und -Techniken.
- *Nielsen Norman Group* [6]: Eine Forschungsgruppe, die sich auf UX-Design spezialisiert hat. Hier finden Sie viele Artikel und Forschungsberichte zu UX-Design-Prinzipien und -Techniken.
- *UX Design* [7]: UX Design ist eine Onlinepublikation, die sich auf UX-Design, UI-Design und Produktstrategie konzentriert. Die Website bietet Artikel, Fallstudien und Ressourcen von Experten aus der Branche. Ein Special: „The Ultimate Guide With All The Secrets You Will Need To Know To Become A Fabulous Design Unicorn“ [8].
- *Awwwards* [9]: Awwwards ist eine Plattform, die das Beste in Webdesign und -entwicklung anerkennt und fördert. Die Website bietet eine Galerie von preisgekrönten Projekten und Fallstudien, die als Inspiration für UI- und UX-Designer dienen können.
- *Dribbble* [10]: Dribbble ist eine Selbstdarstellungsplattform und ein soziales Netzwerk für Designer und digitale Kreative. Es dient als Design-Portfolio-Plattform, Job- und Recruiting-Website und ist eine der größten Plattformen für Designer, um ihre Arbeit online zu teilen. Eine sehr gute Quelle der Inspiration für UI- und UX-Designer.



Henning Fries ist Senior User Interface Architect und beschäftigt sich mit Themen wie Ideation, Accessibility, UX und UI. Mit mehr als fünfzehn Jahren Berufserfahrung arbeitet er als Berater, Trainer, Entwickler, Project Manager und (Web-)Designer in Deutschland und Luxemburg.

Links & Literatur

[1] <https://www.refactoringui.com>

[2]

<https://alistapart.com/article/priority-guides-a-content-first-alternative-to-wireframes/>

[3]

<https://imperavi.com/books/ui-typography/principles/design-algorithm/>

[4] <https://www.smashingmagazine.com>

[5] <https://uxplanet.org>

[6] <https://www.nngroup.com>

[7] <https://uxdesign.cc>

[8] <https://start.uxdesign.cc>

[9] <https://www.awwwards.com>

[10] <https://dribbble.com>