Engineer

Benutzeroberfläche

der sichtbare und bedienbare Teil eines Programms

Was es ist

UI steht für user interface, auf Deutsch Benutzeroberfläche: die Ansichten, Menüs, Felder, Schaltflächen, Meldungen und Töne, über die eine Person einem Programm sagt, was es tun soll, und erfährt, was es getan hat. Dazu gehören auch die Regeln dahinter: welches Bedienelement von wo aus erreichbar ist, was bei einem Klick geschieht und was das Programm sagt, wenn etwas schiefgeht.

Das NIST-Glossar fasst den Begriff am weitesten und definiert eine Benutzeroberfläche als „die physischen oder logischen Mittel, über die Benutzer mit einem System, Gerät oder Prozess interagieren“. Zur physischen Seite gehören die Tastatur, der Touchscreen und der Drehknopf an einem Gerät. Zur logischen Seite gehört alles, was als Reaktion gezeichnet oder ausgegeben wird: ein Fenster mit Menüs, eine Eingabeaufforderung, die auf einen getippten Befehl wartet, eine Stimme, die nach einer Nummer fragt. Eine grafische Oberfläche, eine Befehlszeile und ein gesprochenes Menü sind alles Benutzeroberflächen, und ein Programm hat oft mehrere davon.

Für das Web benennen die WCAG den Baustein. Eine *Benutzeroberflächenkomponente* ist „ein Teil des Inhalts, den Benutzer als ein einzelnes Bedienelement für eine bestimmte Funktion wahrnehmen“. Laut den Anmerkungen zählt als Komponente, was als ein Bedienelement wahrgenommen wird, unabhängig von der Programmierung. Ein einziges Programmelement kann daher mehrere Komponenten enthalten. Das Beispiel des Standards ist ein Steuerelement, mit dem sich Inhalte zeilenweise, seitenweise oder zu einer beliebigen Stelle durchlaufen lassen: Jede Art der Bewegung braucht einen eigenen Namen und zählt als eigene Komponente. Formularelemente, Links und durch Skripte erzeugte Bedienelemente zählen alle dazu.

Zwei verwandte Begriffe begleiten die UI. *UX*, user experience, ist die umfassendere Frage nach der Nutzung eines Produkts von Anfang bis Ende: ob die Aufgabe gelingt, wie lange sie dauert und wie Fehler behandelt werden. Die UI ist der Teil dieser Erfahrung, auf den man am Bildschirm zeigen kann. Ein *Designsystem* ist ein gemeinsamer Bestand an Komponenten, Tokens und Regeln, aus dem ein Team seine UI baut. So sehen zwei Ansichten, die zwei Personen in zwei verschiedenen Monaten gebaut haben, weiterhin wie ein Produkt aus und verhalten sich entsprechend.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist eine Benutzeroberfläche?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Oberflächen, die schnell laden, auf dem Smartphone funktionieren und sich in allen Browsern Ihrer Kunden gleich verhalten.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. Full-Stack-GIS-Kartenentwicklung geleitet — Full-Stack-Kartenentwicklung von der Datenbank bis zum Browser

Full-Stack-GIS-Kartenentwicklung geleitet Kontakt aufnehmen →

Warum es wichtig ist

Der Nutzen eines Programms erreicht die Menschen über seine Oberfläche oder gar nicht. Eine Funktion, die nur findet, wer bereits weiß, wo sie zu finden ist, ist für die meisten Nutzer nie ausgeliefert worden. Eine solche Lücke zeigt sich selten in einer Testsuite, weil jede Funktion funktioniert; sie zeigt sich in Supportfragen und abgebrochenen Formularen.

Konsistenz wird einmal gelernt und wiederverwendet. Menschen übertragen das auf einer Ansicht Gelernte auf die nächste. Erfolgskriterium 3.2.4 der WCAG 2.2, Konsistente Erkennung, verlangt, dass Komponenten mit derselben Funktion innerhalb einer Gruppe von Webseiten konsistent gekennzeichnet werden. Das zugehörige Understanding-Dokument nennt den Grund klar: Wird dieselbe Funktion an zwei Stellen unterschiedlich beschriftet, nehmen Menschen möglicherweise zwei verschiedene Funktionen an oder verbringen Zeit damit herauszufinden, dass es dieselbe ist.

Fehler gehören zur Oberfläche. Jedes Formular bekommt fehlerhafte Eingaben. Seine Antwort darauf ist ebenso eine Gestaltungsentscheidung wie sein Layout. Erfolgskriterium 3.3.1, Fehlererkennung, verlangt, dass bei einem erkannten Fehler das Feld benannt und das Problem in Textform beschrieben

wird. Kriterium 3.3.3, Fehlerempfehlung, geht weiter: Weiß das Programm, wie sich eine automatisch erkannte fehlerhafte Eingabe korrigieren lässt, macht es Vorschläge, sofern dadurch nicht die Sicherheit oder der Zweck des Inhalts gefährdet würde. Das Beispiel im Understanding-Dokument ist ein Feld, das einen Monatsnamen erwartet: Wer „12“ eingibt, bekommt entweder eine Liste akzeptierter Werte oder die Frage „Meinen Sie Dezember?“ angeboten.

Jedes Bedienelement muss auch für Software erklärbar sein. Kriterium 4.1.2, Name, Rolle, Wert, verlangt, dass Programme für jede Benutzeroberflächenkomponente, einschließlich der durch Skripte erzeugten, Namen und Rolle ermitteln können und über Zustandsänderungen benachrichtigt werden. Ein Bedienelement, das auf dem Bildschirm nur wie eines aussieht, ist für die assistiven Technologien unsichtbar, auf die viele Menschen für den Zugang angewiesen sind.

Wie es in der Praxis aussieht

Das deutlichste Beispiel ist eine maritime Plattform, auf der UI und UX des Produkts durchgängig gestaltet wurden. Sie zeigt Fachkräfte, Unternehmen, Schiffe, Stellen und Bewertungen und musste echte maritime Daten ausreichend dicht darstellen, ohne zur unübersichtlichen Informationswand zu werden. Einige wenige Entscheidungen waren dafür ausschlaggebend. Verzeichnisraster lassen sich zwischen Karten und Tabelle umschalten; die Anwendung merkt sich die gewählte Ansicht. Formulare zeigen ihre Anforderungen oberhalb der Eingabe als laufend aktualisierte Liste. Jede Regel wird grün, sobald sie erfüllt ist, und bleibt sonst bernsteinfarben. So erhalten Menschen während der Eingabe Anleitung statt erst nach dem Absenden Korrekturen. Brotkrumennavigation und ein zweispaltiges Layout kehren auf jeder Entitätsseite wieder und lassen einzelne Ansichten als ein Produkt erscheinen. Fehler führen weiter: eine Weiterleitung statt einer Sackgasse. Die Seite benennt den Zusammenhang mit Vertrauen ausdrücklich: Auf einer Plattform, die Menschen für ihre berufliche Laufbahn nutzen, würde ein nachlässiges Erscheinungsbild ihr Vertrauen in die Daten mindern, und das zu Recht.

Die wirtschaftliche Seite zeigt sich am deutlichsten bei 200 UI- und UX-Verbesserungen für eine GIS-Kartenanwendung. Die Oberfläche war leistungsfähig und kompliziert geworden; gute Funktionen lagen hinter umständlichen Interaktionen. Statt zu raten, begann die Arbeit mit Feedback und Nutzungsdaten. Daraus entstand ein Backlog mit 200 konkreten Änderungen, die nach Wirkung geordnet wurden. Die wichtigsten wurden mit dem Team bis zur

Auslieferung als tatsächliche Funktionen vorangetrieben. Die Oberfläche wurde merklich leichter bedienbar, und die Arbeit trug zu einer Verzehnfachung des Umsatzes bei.

Eine Oberfläche über längere Zeit konsistent zu halten ist ein eigenes Problem, dem ein Designsystem im Stil von iOS 26 „Liquid Glass“ begegnet. Seine visuelle Sprache verwendet durchscheinende Flächen mit unscharfem Hintergrund, geschichtete Schatten, etwas Lichtbrechung und Federanimationen, umgesetzt mit Tailwind-Utilities. Jede Ansicht setzt sich aus einem festen Komponentenbestand zusammen, darunter Card, Label, Button, GlassIconButton und DirectoryGrid. Lint-Regeln weisen selbst gebaute Badges oder Chips zurück und verweisen auf die kanonische Komponente. Die Dokumentation stellt konkreten Fehlern ihre Lösungen gegenüber. Konsistenz hängt damit nicht mehr davon ab, wer die Änderung an diesem Tag prüft, sondern wird durch die Werkzeuge durchgesetzt.

Wo es schiefgeht

- **Die Bedeutung steckt allein in der Farbe.** Ein bei gültiger Eingabe grün und bei ungültiger rot umrandetes Feld sagt jemandem, der diese Farben nicht unterscheiden kann, nichts. Das Understanding-Dokument zu Benutzung von Farbe benennt genau diesen Fall und verlangt ein zweites visuelles Zeichen. Ein Wort, ein Symbol oder eine geänderte Form neben der Farbe erfüllt diesen Zweck.
- **Etwas sieht wie ein Bedienelement aus, ist aber keines.** Ein gestalteter Kasten mit Klick-Handler wirkt für Mausnutzer wie eine Schaltfläche, fehlt aber für alle, die eine Tastatur oder einen Screenreader verwenden. Die eingebauten Bedienelemente der Plattform bringen Rolle, Fokus und Tastaturverhalten mit; eine Nachbildung muss all das Stück für Stück von Hand nachbauen.
- **Jede Ansicht erfindet ihre Komponenten neu.** Jede neue Ansicht erhält eigene Schaltflächen, Badges und Karten, alle etwas anders, bis eine einzige Änderung fünf Versionen derselben Sache betrifft. Ein Komponentenbestand, durch einen Linter statt durch Erinnerung durchgesetzt, verhindert dieses Auseinanderdriften.
- **Fehler erscheinen spät und weit von ihrer Ursache entfernt.** Ein Formular, das erst alles annimmt und dann oben in einem roten Block alle Fehler auflistet, lässt Menschen die Felder suchen. Das Feld zu benennen, das Problem direkt daneben in Worten zu beschreiben und eine Korrektur vorzuschlagen, macht dieselbe Prüfung zur Anleitung.

- **Die Oberfläche ist für den Schreibtisch ihrer Entwickler gebaut.** Großer Bildschirm, Maus, gute Sehkraft und schnelle Verbindung beschreiben die Person, die sie gebaut hat, nicht die Menschen, die sie nutzen werden. Ein mit einer Hand gehaltenes Telefon, eine vergrößerte Seite oder eine Tastatur ohne Maus zeigen jeweils Probleme, die an diesem Schreibtisch nie sichtbar werden.

Weiterführende Lektüre

- NIST CSRC Glossary: user interface — die einzeilige Definition aus NIST SP 800-152, die sowohl physische als auch logische Mittel der Interaktion mit einem System umfasst.
- WCAG 2.2: user interface component — die Definition über das wahrgenommene Bedienelement, mit Anmerkungen zum Verhältnis zwischen Komponenten und den Elementen, die sie umsetzen.
- Understanding SC 3.2.4: Consistent Identification — warum Komponenten mit derselben Funktion auf einer Website gleich benannt werden sollten und für welche wiederkehrenden Komponenten das besonders wichtig ist.
- Understanding SC 3.3.3: Error Suggestion — was ein Formular sagen sollte, wenn es weiß, wie sich eine fehlerhafte Eingabe korrigieren lässt, mit ausgearbeiteten Beispielen.
- Understanding SC 1.4.1: Use of Color — wann Farbe Bedeutung tragen darf und wann ein zweites visuelles Zeichen nötig ist.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist eine Benutzeroberfläche?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/ui/