Engineer

UI

den del af et program, man ser og arbejder med

Hvad det er

UI står for user interface, på dansk brugerflade eller brugergrænseflade: de skærme, menuer, felter, knapper, meddelelser og lyde, som en person bruger til at fortælle et program, hvad det skal gøre, og til at se, hvad det gjorde. Begrebet dækker også reglerne bag dem, for eksempel hvilken kontrol der kan nås hvorfra, hvad der sker ved et klik, og hvad programmet siger, når noget går galt.

NIST's ordliste har den bredeste definition: en brugerflade er »the physical or logical means by which users interact with a system, device or process«, altså de fysiske eller logiske midler, brugere interagerer med et system, en enhed eller en proces igennem. Den fysiske side er tastaturet, berøringsskærmen, drejeknappen på et apparat. Den logiske side er alt det, der tegnes eller skrives som svar: et vindue med menuer, en prompt, der venter på en kommando, en stemme, der beder om et nummer. En grafisk brugerflade, en kommandolinje og en talemenu er alle brugerflader, og ét program har ofte flere af dem.

På nettet giver WCAG byggestenen et navn. En *user interface component* er »a part of the content that is perceived by users as a single control for a distinct function«: en del af indholdet, som brugeren opfatter som én kontrol til én bestemt funktion. Noterne til definitionen tæller en komponent ud fra, hvad brugeren oplever som én kontrol, ikke ud fra, hvordan den er programmeret. Ét programmatisk element kan derfor rumme flere komponenter. Standardens eget eksempel er en kontrol, der flytter gennem indholdet linje for linje, side for side eller til et vilkårligt sted, hvor hver måde at flytte på skal have sit eget navn og tæller som sin egen komponent. Formularelementer, links og kontroller bygget af scripts tæller alle med.

To nabobegreber følger med. *UX*, brugeroplevelse, er det bredere spørgsmål om, hvordan det er at bruge produktet fra start til slut: om opgaven bliver løst, hvor lang tid den tager, og hvordan fejl bliver håndteret. Brugerfladen er den del af oplevelsen, der står på skærmen, og som man kan pege på. Et *designsystem* er et fælles sæt komponenter, tokens og regler, som et team bygger sin brugerflade af, så to skærme lavet af to personer i to forskellige måneder stadig ser ud og opfører sig som ét produkt.

Brug begrebet i en beslutning

Brug »Hvad er UI?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Brugerflader, der indlæses hurtigt, virker på en telefon og opfører sig ens i alle de browsere, jeres kunder nu bruger.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har styret full-stack-udviklingen af GIS-kort — full stack-kortløsning, fra database til browser

Har styret full-stack-udviklingen af GIS-kort Kontakt os →

Hvorfor det betyder noget

Et programs værdi når frem til folk gennem brugerfladen eller slet ikke. En funktion, som kun kan findes af den, der allerede ved, hvor den er, er for de fleste brugere aldrig blevet leveret. Den slags huller viser sig sjældent i en testsuite, fordi hver funktion virker. De viser sig i supporthenvendelser og formularer, der bliver forladt halvvejs.

Ensartethed læres én gang og bruges igen. Folk tager det, de lærte på én skærm, med til den næste. Succeskriterium 3.2.4 i WCAG 2.2, Consistent Identification, kræver, at komponenter med samme funktion i et sæt websider identificeres ensartet. Understanding-dokumentet giver grunden i klar tekst: når den samme funktion har forskellige navne to steder, kan folk tro, at de to gør noget forskelligt, eller bruge tid på at finde ud af, at de gør det samme.

Fejl er en del af brugerfladen. Enhver formular møder forkert input, og hvordan den svarer, er lige så meget en designbeslutning som dens layout. Succeskriterium 3.3.1, Error Identification, kræver, at en opdaget fejl peger på feltet og beskriver problemet i tekst. Kriterium 3.3.3, Error Suggestion, går videre: når programmet ved, hvordan automatisk opdaget forkert input kan rettes, giver det forslag, medmindre det ville bringe indholdets sikkerhed eller

formål i fare. Understanding-dokumentets eksempel er et felt, der forventer et månedsnavn, hvor en person, der skriver »12«, får tilbudt enten listen over gyldige værdier eller spørgsmålet, om det var december, der mentes.

Hver kontrol skal også kunne forklares for software. Kriterium 4.1.2, Name, Role, Value, kræver, at navn og rolle kan aflæses af programmer for hver brugerfladekomponent, også dem, scripts har bygget, og at ændringer i tilstand bliver meldt videre. En kontrol, der kun ligner en kontrol på skærmen, er usynlig for den hjælpeteknologi, mange mennesker bruger til at nå den.

Sådan ser det ud i praksis

Det tydeligste eksempel er en maritim platform, hvor produktets UI og UX blev designet fra ende til anden. Den viser professionelle, virksomheder, skibe, jobs og anmeldelser og skulle være tæt nok til at rumme rigtige maritime data uden at blive en mur. En håndfuld beslutninger bar den. Oversigtsgitrene skifter mellem kort og tabel, og appen husker, hvilken visning man valgte. Formularerne viser kravene over inputfeltet som en levende liste, hvor hver regel bliver grøn, når den er opfyldt, og ravgul, så længe den ikke er, så folk bliver vejledt, mens de skriver, i stedet for at blive rettet, efter at de har sendt. Breadcrumbs og et layout i to kolonner går igen på hver entitetsside, så adskilte skærme læses som ét produkt. Fejl fører et sted hen: en omdirigering i stedet for en blindgyde. Siden fremfører selv tillidsargumentet: på en platform, folk bruger til deres karriere, ville et sjusket udseende få dem til at stole mindre på dataene, og de ville have ret.

Den kommercielle side ses tydeligst i 200 UI/UX-forbedringer af en GIS-kortapplikation. Grænsefladen var blevet kraftfuld og kompliceret, med god funktionalitet gemt bag klodsede interaktioner. I stedet for at gætte tog arbejdet udgangspunkt i feedback og brugsdata, gav en backlog med 200 konkrete ændringer, rangerede dem efter effekt og førte de vigtigste gennem teamet, til de blev sat i produktion som rigtige funktioner. Grænsefladen blev mærkbart nemmere at bruge, og arbejdet bidrog til en tidobling af omsætningen.

At holde en brugerflade ensartet over tid er et problem for sig, og det blev løst i et designsystem i iOS 26's »Liquid Glass«-stil. Det visuelle sprog bruger gennemskinnelige paneler med backdrop-blur, lagdelte skygger, en smule refraktion og spring-animationer, bygget af Tailwind-utilities. Hver skærm sammensættes af et fast katalog af komponenter, blandt andre Card, Label,

Button, GlassIconButton og DirectoryGrid. Lintregler afviser et hjemmelavet badge eller en hjemmelavet chip og henviser til den kanoniske komponent, og dokumentationen kobler konkrete fejl til deres løsning. Ensartetheden afhænger dermed ikke længere af, hvem der reviewer ændringen den dag, men holdes af værktøjerne.

Hvor det går galt

- **Farven bærer betydningen alene.** Et felt med grøn kant for gyldigt og rød for ugyldigt siger intet til den, der ikke kan skelne de to farver. Understanding-dokumentet for Use of Color nævner netop det tilfælde og kræver et ekstra visuelt tegn. Et ord, et ikon eller en ændret form ved siden af farven klarer opgaven.
- **Noget, der er tegnet som en kontrol, er ikke en.** En stylet boks med en klikhandler ligner en knap for den, der bruger mus, og er fraværende for den, der bruger tastatur eller skærmlæser. Platformens egne kontroller kommer med rolle, fokus og tastaturadfærd, som en efterligning skal genopbygge i hånden, stykke for stykke.
- **Hver skærm genopfinder sine komponenter.** Hver ny visning får sine egne knapper, badges og kort, hver en smule anderledes, indtil en enkelt ændring betyder, at fem versioner af den samme ting skal rettes. Ét komponentkatalog, håndhævet af en linter i stedet for af hukommelsen, holder afvigelserne ude.
- **Fejl kommer sent og langt fra årsagen.** En formular, der tager imod alt og derefter sender en rød blok øverst på siden med alt det, der var galt, får folk til at lede efter felterne. Når feltet bliver nævnt, problemet beskrevet i ord lige ved siden af og en rettelse foreslået, bliver den samme kontrol til vejledning.
- **Brugerfladen bliver bygget til bygherrens skrivebord.** En stor skærm, en mus, et godt syn og en hurtig forbindelse beskriver den, der lavede den, ikke dem, der skal bruge den. En telefon i én hånd, en forstørret side eller et tastatur uden mus afslører hver især problemer, som skrivebordet aldrig viser.

Læs mere

- NIST CSRC Glossary: user interface — definitionen på én linje fra NIST SP 800-152, der dækker både de fysiske og de logiske midler til at interagere med et system.
- WCAG 2.2: user interface component — definitionen ud fra den oplevede kontrol, med noter om, hvordan komponenter forholder sig til de elementer, der implementerer dem.

- Understanding SC 3.2.4: Consistent Identification — hvorfor komponenter med samme funktion skal have samme navn på tværs af et site, og hvilke gentagne komponenter det betyder mest for.
- Understanding SC 3.3.3: Error Suggestion — hvad en formular skal sige, når den ved, hvordan forkert input kan rettes, med gennemregnede eksempler.
- Understanding SC 1.4.1: Use of Color — hvornår farve må bære betydning, og hvornår et ekstra visuelt tegn er påkrævet.

Ordliste

Vælg ud fra jeres faktiske vilkår

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er UI?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/ui/