Engineer

UI

🛠 den del av ett program som en person ser och arbetar med

Vad det är

UI är förkortningen av engelskans user interface, användargränssnitt: de skärmar, menyer, fält, knappar, meddelanden och ljud genom vilka en person talar om för ett program vad det ska göra och får veta vad det gjorde. Det är också reglerna bakom dem, t.ex. vilken kontroll som går att nå varifrån, vad som händer vid ett klick och vad programmet säger när något går fel.

Ordlistan från NIST tar det bredaste greppet och definierar ett användargränssnitt som de fysiska eller logiska medel genom vilka användare samverkar med ett system, en enhet eller en process. Den fysiska sidan är tangentbordet, pekskärmen och vredet på en apparat. Den logiska sidan är allt som ritas upp eller skrivs ut som svar: ett fönster med menyer, en prompt som väntar på ett inskrivet kommando, en röst som ber om ett nummer. Ett grafiskt gränssnitt, en kommandorad och en talad meny är alla användargränssnitt, och ett och samma program har ofta mer än ett av dem.

På webben namnger WCAG 2.2 byggstenen. En *user interface component* är enligt standarden en del av innehållet som användarna uppfattar som en enda kontroll för en avgränsad funktion. Anmärkningarna räknar en komponent efter vad användaren uppfattar som en kontroll, inte efter hur den programmerades. Ett enda programmatiskt element kan därför rymma flera komponenter. Standardens eget exempel är en kontroll som flyttar genom innehållet rad för rad, sida för sida eller till valfri punkt, där varje sätt att flytta behöver ett eget namn och räknas som en egen komponent. Formulärelement, länkar och kontroller som byggs med skript räknas alla.

Två närliggande begrepp följer med. *UX*, user experience eller användarupplevelse, är den bredare frågan om hur det är att använda produkten från början till slut: om uppgiften blir gjord, hur lång tid det tar och hur misstag

hanteras. UI är den del av upplevelsen som går att peka på i en skärm. Ett *designsystem* är en gemensam uppsättning komponenter, designvariabler (tokens) och regler som ett team bygger sitt gränssnitt av, så att två skärmar som två personer har byggt under två olika månader ändå ser ut och beter sig som en och samma produkt.

Använd begreppet i ett beslut

Använd ”Vad är UI?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Gränssnitt som laddar snabbt, fungerar i mobilen och beter sig likadant i alla webbläsare dina kunder råkar använda.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Styrde full stack-utvecklingen av GIS-kartor — kartutveckling i full stack, från databas till webbläsare

Styrde full stack-utvecklingen av GIS-kartor Hör av er →

Varför det spelar roll

Ett programs värde når människor genom gränssnittet eller inte alls. En funktion som bara går att hitta för den som redan vet var den finns har, för de flesta användarna, aldrig levererats. Ett sådant glapp syns sällan i en testsvit, eftersom varje funktion fungerar. Det syns i frågor till supporten och i formulär som lämnas halvfärdiga.

Konsekvens lärs in en gång och återanvänds. Människor tar med sig det de lärde sig på en skärm till nästa. Framgångskriterium 3.2.4 i WCAG, Consistent Identification, kräver att komponenter med samma funktion inom en uppsättning webbsidor identifieras på samma sätt. Det förklarande dokumentet (Understanding) för kriteriet ger skälet i klartext: när samma funktion har olika etiketter på två ställen kan människor tro att de två gör olika saker, eller lägga tid på att räkna ut att de inte gör det.

Fel är en del av gränssnittet. Varje formulär möter felaktig indata, och hur det svarar är lika mycket ett designbeslut som dess layout. Framgångskriterium 3.3.1, Error Identification, kräver att ett upptäckt fel pekar ut fältet och beskriver problemet i text. Kriterium 3.3.3, Error Suggestion, går längre: när programmet vet hur automatiskt upptäckt felaktig indata kan rättas ger det förslag, såvida det inte skulle äventyra innehållets säkerhet eller syfte. Exemplet i det förklarande dokumentet är ett fält som väntar sig namnet på en månad, där den som skriver ”12” erbjuds antingen listan över godtagna värden eller frågan om det var december som avsågs.

Varje kontroll måste också gå att förklara för programvara. Kriterium 4.1.2, Name, Role, Value, kräver att namn och roll går att avläsa för program för varje komponent i gränssnittet, även de som byggs med skript, och att ändringar i tillstånd rapporteras till dem. En kontroll som bara ser ut som en kontroll på skärmen är osynlig för de hjälpmedel som många är beroende av för att nå den.

Så ser det ut i praktiken

Det tydligaste fallet är en maritim plattform där produktens UI och UX utformades från början till slut. Den visar yrkesverksamma, företag, fartyg, jobb och recensioner och måste vara tät nog för att rymma verkliga maritima data utan att bli en vägg. Några få beslut bar det. Katalogrutnäten växlar mellan kort och tabell, och appen kommer ihåg vilken vy personen valde. Formulären visar sina krav ovanför fältet som en lista i realtid, där varje regel blir grön när den är uppfylld och gul när den inte är det, så att människor får vägledning medan de skriver i stället för att bli tillrättavisade efter att de har skickat. Brödsmulor och en layout i två kolumner återkommer på varje sida för en entitet, så att separata skärmar läses som en produkt. Felen leder någonstans: en omdirigering i stället för en återvändsgränd. Sidan för fram förtroendeargumentet rakt ut: på en plattform som människor använder för sina karriärer skulle ett slarvigt utseende få dem att lita mindre på uppgifterna, och de skulle ha rätt i det.

Den kommersiella sidan syns tydligast i 200 förbättringar av UI och UX för en kartapplikation inom GIS. Gränssnittet hade blivit kraftfullt och komplicerat, med bra funktionalitet bakom klumpiga interaktioner. I stället för att gissa utgick arbetet från återkopplingen och användningsdatan, tog fram en backlog med 200 konkreta ändringar, rangordnade dem efter effekt och drev de som betydde mest genom teamet tills de levererades som riktiga funktioner. Gränssnittet blev märkbart lättare att använda, och arbetet bidrog till en tiodubbling av intäkterna.

Att hålla ett gränssnitt konsekvent över tid är ett separat problem, och det angrips i ett designsystem i stil med ”Liquid Glass” från iOS 26. Det visuella språket använder genomskinliga paneler med bakgrundsoskärpa, skikt av skuggor, lite ljusbrytning och fjäderanimationer, uppbyggt av verktygsklasser i Tailwind. Varje skärm sätts ihop av en fast förteckning över komponenter, bl.a. Card, Label, Button, GlassIconButton och DirectoryGrid. Lintregler avvisar ett handbyggt märke eller chip och pekar mot den kanoniska komponenten, och dokumentationen ställer konkreta misstag bredvid sina lösningar. Konsekvensen slutar hänga på vem som granskar ändringen just den dagen och upprätthålls i stället av verktygen.

Var det går fel

- **Färgen bär betydelsen på egen hand.** Ett fält med grön kant för giltigt och röd för ogiltigt säger ingenting till den som inte kan skilja de färgerna åt. Det förklarande dokumentet för Use of Color nämner precis det fallet och kräver ett andra visuellt tecken. Ett ord, en ikon eller en ändrad form bredvid färgen gör jobbet.
- **Något som är ritat för att se ut som en kontroll är ingen kontroll.** En formgiven ruta med en klickhanterare ser ut som en knapp för den som använder mus och finns inte alls för den som använder tangentbord eller skärmläsare. Plattformens egna kontroller kommer med en roll, fokus och ett beteende för tangentbordet som en efterapning måste bygga upp för hand, bit för bit.
- **Varje skärm uppfinner sina komponenter på nytt.** Varje ny vy får egna knappar, märken och kort, alla lite olika, tills en enda ändring innebär att fem versioner av samma sak ska redigeras. En enda förteckning över komponenter, upprätthållen av en linter i stället för av minnet, håller glidningen borta.
- **Felen kommer sent och långt från sin orsak.** Ett formulär som tar emot allt och sedan returnerar ett rött block överst med en lista över vad som var fel får människor att leta efter fälten. Att peka ut fältet, beskriva problemet bredvid det i ord och föreslå en rättelse gör samma kontroll till vägledning.
- **Gränssnittet byggs för byggarens skrivbord.** En stor skärm, en mus, skarp syn och en snabb uppkoppling beskriver den som byggde det, inte de människor som ska använda det. En telefon som hålls i en hand, en inzoomad sida eller ett tangentbord utan mus avslöjar var för sig problem som ett skrivbord aldrig visar.

Läs mer

- NIST CSRC Glossary: user interface — definitionen på en rad från NIST SP 800-152, som täcker både de fysiska och de logiska sätten att samverka med ett system.
- WCAG 2.2: user interface component — definitionen utifrån den kontroll som användaren uppfattar, med anmärkningar om hur komponenter förhåller sig till de element som implementerar dem.
- Understanding SC 3.2.4: Consistent Identification — varför komponenter med samma funktion ska ha samma namn på hela webbplatsen, och för vilka återkommande komponenter det betyder mest.
- Understanding SC 3.3.3: Error Suggestion — vad ett formulär ska säga när det vet hur felaktig indata skulle kunna rättas, med genomarbetade exempel.
- Understanding SC 1.4.1: Use of Color — när färg får bära betydelse och när ett andra visuellt tecken krävs.

Ordlista

Välj utifrån era faktiska förutsättningar

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är UI?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

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

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/ui/