

Appunti Interazione Uomo Macchina

Design

Per design si intende sia il processo di progettazione e pianificazione sia il risultato di questo processo. L'obiettivo principale del design di prodotto non è necessariamente quello di trovare una soluzione al problema specifico ma è piuttosto quella di comprendere il problema stesso nel suo insieme. Nel mondo del design il primo passo è sempre quello di capire perché il problema esiste e solo dopo aver appurato che l'origine di un problema non può essere eliminata o mitigata ci si adopera per cercare di risolverlo nello specifico. Nel design di prodotto ci si trova infatti spesso costretti a modificare i requirement e le specifiche di prodotto per andare in contro alle esigenze degli utenti e sacrificando funzionalità tecniche e qualità dell'implementazione software. Trovare il corretto bilanciamento fra esperienza utente, funzionalità e qualità tecnica è la parte più complessa dell'intero processo di sviluppo prodotto.

Il mondo del design è diventato talmente ampio che il termine design da solo ormai non ha quasi più significato. Esistono varie sotto discipline del design e con queste numerose professioni, metodi di lavoro, scuole di pensiero e altrettante immancabili faide e lotte fra fazioni. Tra le varie sotto discipline del design troviamo l'interaction design che è l'attività di progettazione dell'interazione che avviene tra esseri umani e oggetti in generale. L'obiettivo principale dell'interaction design è quello di rendere macchine, servizi e sistemi usabili dagli utenti per cui sono stati pensati e realizzati e non solamente dai propri creatori. Le forze trainanti lo sviluppo di un prodotto dovrebbero esserle quindi gli utenti reali e i loro bisogni e non solo le tecnologie. Ciò che ci interessa quindi è il campo dell'interazione uomo-macchina e uomo-computer cioè Human-Machine Interaction (HMI) e Human-Computer Interaction (HCI). L'interaction design ha a sua volta altre sottodiscipline che spesso si intrecciano tra di loro nei vari settori:

- **Product Design** : è un processo strategico di risoluzione dei problemi che guida l'innovazione e porta a una migliore qualità della vita attraverso prodotti, sistemi, servizi e esperienze innovative. Nel design di prodotto si progettano beni e servizi il cui obiettivo principale è quello di essere utilizzati da quanti più utenti possibili migliorandone la vita
- **User Experience (UX) design** : è il processo volto ad aumentare la soddisfazione e la fedeltà degli utenti migliorando l'usabilità, la facilità d'uso e il piacere fornito nell'interazione tra il cliente e il prodotto. L'UX designer ha il compito di far vivere all'utente la miglior esperienza possibile evitando che l'oggetto induca sensazioni di frustrazione e delusione.
- **User Interface (UI) design** : ha come focus il modo in cui le persone interagiscono con la tecnologia, lo scopo è migliorare la loro comprensione di ciò che si può fare, ciò che succede e ciò che è appena successo, basandosi su principi psicologici, tecnici ed estetici. Dallo studio dell'UX si crea uno schema di interazione che poi viene passato allo UI designer che si occupa di creare l'interfaccia grafica che l'utente vedrà e con cui interagirà. UI designer non costruisce l'interfaccia utente ma progetta l'aspetto estetico e la struttura dell'interfaccia così che questa durante l'utilizzo induca l'utente a seguire l'esperienza che è stata progettata per lui. L'UI designer produce un wireframe per implementare la reale interfaccia del prodotto o servizio.

Il processo di Product Design è un processo strutturato che include varie figure e discipline. L'interfaccia vera e propria viene implementata solo alla fine del percorso di progettazione da figure con profilo tecnico informatico che comprendo le richieste provenienti dalle fasi precedenti e le implementano in un prodotto finito.

Human Centered Design

Lo Human Centered Design (HCD) è una metodologia di progettazione che parte dai bisogni, dalle capacità e dai comportamenti umani, adattando la progettazioni a quei bisogno, quelle capacità e dai comportamenti umani. Lo HCD è un approccio che mette al centro dell'attenzione l'utente e le sue esigenze a differenza di un approccio techno-centrico che non si concentra a cercare tecnologie che possano risolvere i problemi degli utenti ma cerca problemi da risolvere con nuove tecnologie. Per questo lo HCD parte dall'osservazione dell'utente e dai suoi comportamenti e solo identifica la tecnologia necessaria. Progettare interfacce che funzionano fintanto che le cose vanno bene è relativamente facile, ma la comunicazione è ancora più importante quando le cose non vanno bene ed è qui che i progettisti devono concentrare la loro attenzione non a non far presentare errori agli utenti, poiché inevitabile, ma progettare un'interfaccia che guidi l'utente nella risoluzione in modo che l'utente non si senta frustrato. L'obiettivo della forma di pensiero HCD deve dunque essere quello di creare nell'utente empatia verso il sistema.

Possiamo schematizzare un processo HCD come un flusso continuo ed iterativo che attraversa le seguenti fasi:

- **Specificare il contesto d'uso** : identificare gli utenti che utilizzeranno il prodotto, per cosa lo utilizzeranno e sotto quali condizioni e vincoli
- **Specificare i requisiti** : identificare i requisiti e gli obiettivi dell'utente che devono essere raggiunti dal prodotto
- **Progettare la soluzione** : realizzare un prodotto finale attraversando varie fasi di bozze e prototipi
- **Testare e valutare** : testare il prodotto con gli utenti e valutarne l'efficacia e l'efficienza

In tutto questo è necessario anche considerare l'usabilità cioè il grado in cui un prodotto può essere utilizzato da utenti specifici per raggiungere obiettivi specifici con efficacia, efficienza e soddisfazione in un contesto d'uso specifico ed è imprescindibile.

Progettazione delle interfacce

Il problema è che il buon design non è universale, poiché l'esperienza di interazione è soggettiva e quindi dipende più dalla persona che dal prodotto, ma esistono due proprietà fondamentali che qualsiasi progetto deve avere:

- **Discoverability (rilevabilità)** : la capacità di un sistema di veicolare e comunicare i propri usi all'utente a prima vista. Per avere una buona discoverability solitamente si usa la visibilità, cioè se le funzioni dell'oggetto sono visivamente eclatanti è probabile che abbia una buona discoverability
- **Understanding (comprensibilità)** : la capacità del prodotto di farsi usare correttamente dall'utente cioè la misura di quanto bene l'utente capisce come usare le funzioni del prodotto.

design of useful things

Quando le cose vanno bene, si dimenticano subito, quando vanno male non si dimenticano mai. Questo fenomeno è noto a tutti ed ho dovuto perché ormai viviamo in una società in cui le cose devono andare bene per definizione, mentre se qualcosa va storto scattano una serie di reazioni che portano la persona a provare sensazioni e emozioni spiacevoli. Il design deve quindi preoccuparsi di come funzionano le cose, come vengono controllate e dalla natura delle interazioni che questi oggetti e sistemi abilitano con gli utenti. Questo vuol dire che una cattiva esperienza utente induce l'utilizzatore ad abbandonare l'utilizzo del prodotto e etichettare quel prodotto e azienda come negativa e perciò bisogna fare di tutto per evitarlo. Il problema è che le macchine sono concepite, progettate e costruite per funzionare perfettamente ma secondo regole di comportamento rigide a differenza dell'uomo che è aleatorio, versatile, variabile, volubile e intuitivo e data questa differenza bisogna evitare che le macchine obblighino le persone a smettere di comportarsi da umani inducendole a comportarsi da macchine. Quindi quando la macchina fa la cosa sbagliata, tipicamente l'utente si sente incapace e perciò deve essere ribaltata la prospettiva in modo che quando le cose vanno male la colpa non è mai dell'utente ma è sempre della macchina e quindi del progettista, che deve progettare per come le persone sono e non per come vorrebbe che fossero.

L'incidente di Three Mile Island è un eclatante esempio pratico di mal progettazione delle interfacce. Il 28 marzo del 1979 nella contea di Dauphin, in Pennsylvania ci fu il più grave incidente nucleare avvenuto negli Stati Uniti che ha portato al rilascio di piccole quantità di gas radioattivi e di iodio radioattivo nell'ambiente. Il reattore era ad un regime di potenza del 97% e il sistema di raffreddamento secondo si era spento portando un considerevole aumento di pressione al circuito primario causando l'apertura di una valvola PORV di rilascio posta sul pressurizzatore e portò allo SCRAM, l'arresto di emergenza mediante l'inserimento delle barre di controllo. A questo punto la valvola di rilascio non si richiuse e gli operatori non si resero conto del problema, poiché non vi era nella strumentazione l'indicazione della reale posizione della valvola. La strumentazione era infatti legata all'alimentazione del motore della valvola e non alla reale posizione della valvola. Fu così che il

circuito di raffreddamento primario si vuotò parzialmente e il calore residuo del nocciolo del reattore non poté essere smaltito. Gli operatori non poterono diagnosticare correttamente cosa avveniva e reagire in maniera adeguata. La strumentazione carente della sala controllo e l'addestramento inandeguato risultarono essere le cause principali dell'incidente.

Principi fondamentali dell'interazione

esperienza

affordance

significanti

mapping

feedback

modello concettuale

immagine di sistema

Vincoli

classi di vincoli:

- vincoli fisici
- vincoli culturali
- vincoli semantici
- vincoli logici

funzioni obbliganti

- interlock
- lock-in
- lock-out

comandi centrati sulle attività

Come le persone fanno le cose

golfi dell'esecuzione e della valutazione

i 7 stati dell'azione

- scopo

stati d'esecuzione:

- progettare
- specificare
- eseguire

stati di valutazione:

- percepire
- interpretare
- confrontare

i 7 principi fondamentali della progettazione

- cosa voglio ottenere?
- quali sono le sequenze d'azione alternative?
- quale azione posso fare ora?
- cosa è successo?
- cosa significa?
- va bene? ho realizzato il mio scopo?

feedforward e feedback in interaction

i 7 principi fondamentali del design

- visibilità
- feedback
- modello concettuale
- affordance
- significanti
- mapping
- vincoli

pensiero umano

conscio e subconscio

tipi di memoria:

- memoria dichiarativa
- memoria procedurale

emozioni e cognizione

processing del cervello:

- livello viscerale
- livello comportamentale
- livello riflessivo

Errore Umano

root cause analysis

i 5 perché

definizione di errore

- lapsus o slips
 - di azione
 - di memoria
- mistakes
 - rule based
 - knowledge based
 - memory lapse

interruzioni

feedback sbagliati

prevenzione dell'errore

- comprendere le cause dell'errore
- controlli di sensibilità
- annunciare le azioni
- rendere più semplice la scoperta e comprensione degli errori
- aiutare l'utente a compiere correttamente le azioni

per la prevenzione dell'errore utilizzare:

- constraints
- undo
- messaggi di errore e di conferma
- aumentare il numero di controlli
- migliorare il modello concettuale dell'utente
- allertare l'operatore umano quando ci si avvicina a un errore

mitigazione dell'errore usando metafora formaggio svizzero:

- aumentare il numero di controlli (fette di formaggio)
- diminuire la probabilità di errore (buchi nel formaggio)
- allertare l'operatore umano quando ci si avvicina a un errore (buchi si sono allineati)

Le interfacce utente

Un interfaccia è qualcosa che sta fra due facce, un punto di contatto tra due sistemi che cercano di comunicare. Le interfacce possono far comunicare due macchine fra loro oppure possono far comunicare l'uomo con la macchina. Solitamente è sempre composta da due parti: uno strumento appartenente ad una persona che serve per compiere un'azione e l'interfaccia che è ciò che serve per guidare l'utente nell'esecuzione dell'azione.

Quando parliamo di interfaccia utente intendiamo lo spazio di un sistema dove avviene l'interazione tra uomo e macchina e il loro obiettivo è quello di consentire all'utente di controllare e far funzionare il sistema in modo efficiente. Quindi l'interfaccia deve essere progettata in modo da semplificare l'interazione fra l'uomo e la macchina per poi pensare a migliorare la User Experience. Una buona interfaccia massimizza la quantità di informazioni evitando comunque di sovraccaricare l'utente che deve riuscire ad utilizzare il sistema con il minimo sforzo fisico e cognitivo.

livelli di interfacce:

- Human Interface Device (HID)
- Human Machine Interface (HMI)
- Human Computer Interface (HCI)

Le interfacce utente tipicamente sono organizzate sulla base dei sensi che utilizzano per stabilire l'interazione tra uomo e macchina:

- tactile UI
- visual UI
- auditory UI
- olfactory UI
- gustatory UI
- equilibrial UI

La maggior parte delle interfacce però utilizzano più di un senso per stabilire il collegamento e vengono chiamate Composite User Interface (CUI). Tra le più note abbiamo:

- Graphical User Interface (GUI) : composta da interfacce grafiche e tattile
- Multimodal User Interface (MUI) : composta da interfacce che utilizzano più di un senso

Le CUI poi possono essere categorizzate in tre macrocategorie:

- Standard : usano dispositivi standard come tastiera, mouse e monitor
- Virtual : bloccano all'utente l'interazione con il mondo reale e creano un mondo virtuale che funge da interfaccia
- Augmented : non bloccano l'utente dalla percezione del mondo reale ma lo vanno ad arricchire, espandendola con un mix tra contenuti reali e virtuali

Le CUI possono essere categorizzate anche tramite il numero di sensi che utilizzano. Ad esempio, lo Smell-O-Vision è una CUI standard 3S (3 sensi) che utilizza la

vista, l'olfatto e l'udito. Se si aggiungesse un quarto senso (come ad esempio le poltrone mobili del cinema) diventerebbe 4S. Quando un'interfaccia utente interagisce con tutti i sensi umani viene chiamata Qualia Interface.

Human Interface Device (HID)

Un HID è un tipo di dispositivo informatico spesso usato da umani che consente l'interazione input/output tra umani e computer. Con HID indichiamo sia i device fisici, sia il protocollo USB-HID. Gli HID standard sono stati adottati la prima volta principalmente per semplificare il processo di installazione di ogni device. Tutti i dispositivi definiti HID invece inviano pacchetti auto-descrittivi che possono contenere qualsiasi tipo di dato e formato. Un driver HID su computer analizza i dati e consente l'associazione dinamica dell'I/O dei dati con le funzionalità dell'applicazione.

Nel protocollo HID ci sono 2 entità:

- Host : comunica con il device e riceve dati in input dal device in base alle azioni dell'umano e invece gli output attraversano il device e arrivano fino all'umano.
- Device : è un'entità che interagisce direttamente con un umano

Questo protocollo rende l'implementazione di device molto semplice inviando all'host dei pacchetti di dati chiamati "HID descriptor" che descrivono come sono fatti i pacchetti del device e indicano:

- N° di pacchetti che il device supporta
- Dimensione dei pacchetti
- Lo scopo di ogni byte e bit nel pacchetto

Il device tipicamente memorizza HID descriptor in ROM così non ha bisogno di capire o analizzare l'HID descriptor. Ci si aspetta che l'host sia più complesso del device, ed ha bisogno, prima di comunicare con il device, di ricevere e analizzare l'HID descriptor. Dato che non tutti gli host potrebbero essere capaci di interpretare l'HID descriptor, HID descrive anche il boot protocol con pacchetti di formato predefinito

HID è stato esteso ad una serie di protocolli:

- bluetooth HID
- serial HID
- ZigBee input device
- HID over I²C
- HOGP

Le periferiche HID sono organizzate in 2 categorie:

- di input : basati su sensori che convertono la realtà fisica in segnali elettrici.
- di output : basati su attuatori che convertono segnali elettrici in perturbazioni nel mondo reale.

Le varie classi di HID sono:

- testi e caratteri
- posizioni (sistemi di puntamento)
- suoni
- immagini
- parametri ambientali
- posizione
- parametri fisiologici e biologici

HID testi e caratteri

Il primo dispositivo HID di testi e caratteri più comune è la tastiera ma sono disponibili più tipi in base ad ogni particolare necessità e variano a seconda della dimensione, dal formato (ANSI o ISO) e del sistema operativo su cui verranno utilizzati. I layout da prendere in considerazione sono:

- layout fisico : corrisponde al posizionamento dei tasti sulla tastiera
- layout visuale : corrisponde all'arrangiamento dei simboli che appaiono sui tasti
- layout funzionale : corrisponde all'associazione tasto-significato all'interno di un software

Il layout design più utilizzato dalle popolazioni latine è il QWERTY. Esistono inoltre tastiere multifunzione che permettono di estendere le tastiere standard con altri tasti e mappature ma necessitano di driver aggiuntivi per funzionare.

Un altro dispositivo di HID testi e caratteri è il lettore di codice a barre poiché scannerizza è una serializzazione di caratteri e per lo stesso motivo lo è anche il QR code. I tag RFID sono un altro dispositivo di HID testi e caratteri che permettono di identificare un oggetto o una persona tramite un segnale radio che può essere passivo o attivo. Anche NFC è un dispositivo di HID testi e caratteri come i tag RFID ma funziona con distanza minore, è più lento ma la comunicazione è bidirezionale.

HID sistemi di puntamento

Sono dispositivi che trasferiscono input di tipo spaziale verso un computer come movimenti fisici dell'utente attraverso movimenti di punatori o altri cambiamenti visivi. Si definisce in merito la legge di Fitt per il calcolo del tempo di movimento.

legge di Fitt: $MT = a + b * \log_2(2D/W)$

con a definito come il tempo in secondi necessario per iniziare o smettere di muoversi, b la velocità del dispositivo, D la distanza dal punto iniziale al centro dell'obiettivo, W la larghezza dell'obiettivo misurata lungo l'asse su cui ci si sta muovendo.

Si danno i seguenti criteri per la classificazione di dispositivi di puntamento:

- sulla base del tipo di input:
 - diretto : se il puntatore si trova nella stessa posizione fisica dell'utente
 - indiretto : quando traduce il movimento sullo schermo
- sulla base del modo in cui il movimento viene mappato sull'interfaccia:
 - assoluto : quando il mapping tra lo spostamento nel mondo fisico viene replicato così com'è sul dispositivo
 - altrimenti
- sulla base di come i dispositivi producano il segnale sulla base dello spostamento:
 - isotonic : si può muovere nello spazio e misura lo spostamento
 - isometric : è fisso e misura la forza che viene applicata
 - elastico : la forza applicata è proporzionale allo spostamento
- sulla base della velocità con cui si fa avanzare il puntatore:
 - position control : ci controlla la posizione del puntatore

- rate control : ci controlla la velocità e direzione del puntatore

I dispositivi di puntamento innovativi sono l'eye tracker che misurano la posizione della pupilla e i movimenti degli occhi. I vari metodi per estrarre informazioni sono:

- bright-pupil : si illumina con luce infrarossa l'occhio e si misura la posizione della pupilla. La luce è coassiale all'occhio in modo che la pupilla rifletta la luce e illumini di rosso l'occhio.
- dark-pupil : si illumina con luce infrarossa l'occhio e si misura la posizione della pupilla. La luce non è coassiale all'occhio e la pupilla appare nera.
- passive light : si misura la posizione della pupilla con la luce ambientale Tutto ciò è differente dal gaze tracking che si occupa di capire dove il soccetto stia guardando riportando l'angolo della pupilla nello spazio tridimensionale in cui si trova l'utente. Ma per rendere l'eye tracking un sistema di puntamento va affiancato con il gaze tracking. I dispositivi dataglove sono dei guanti che permettono di rilevare i movimenti delle mani e delle dita. I dispositivi aptici sono dispositivi che permettono agli utenti di interagire con ambienti virtuali tramite feedback tattili. Altri dispositivi sono gli smart paper e le lavagne digitali.

HID suoni

In fisica, un suono è una vibrazione che si propaga attraverso una onda acustica, tramite un gas, liquido o solido. Il suono può essere catturato usando microfoni, device con sensori che convertono suoni in segnali elettrici. I microfoni possono essere usati anche in tandem, formando un array di microfoni in modo di estrarre precisamente il suono che vogliamo evitando suoni indesiderati, come il rumore di fondo, e riconoscere la direzione del suono.

HID video

I sensori di immagini sono sensori che rilevano attraverso la radiazione luminosa e convertono immagini in informazioni. Ci sono due tipi di sensori di immagini: i charge-coupled device(CCD) e gli active-pixel sensor(CMOS). Entrambi i sensori sono basati su metal-oxide-semiconductor(MOS). Il 3D scanner consiste nel rappresentare dimensione e posizione nello spazio tridimensionale di oggetti e ambienti. I 3D scanner sono divisi in due categorie:

- Passivi : non emettono radiazioni elettromagnetiche, si affidano all'illuminazione ambientale. Alcuni esempi:
 - camere stereoscopiche : usano due camere poste ad una distanza focale simile a quella dell'occhio umano e, calcolando la differenza nelle immagini pixel per pixel, creano una terza immagine detta depthmap per creare l'illusione di profondità
 - sistemi fotometrici : assumono l'esistenza di una sorgente luminosa controllabile e analizzando gli spostamenti delle ombre al variare dell'incidenza (della sorgente luminosa) vengono calcolati sia la profondità sia il colore.
 - tecniche silhouette : tipicamente messe a disposizione anche da applicazioni per dispositivi mobili, scattando più foto dello stesso oggetto da diversi angoli e usando l'accelerometro e la piattaforma iniziale posizionano i piani immagini nello spazio tridimensionale e ricostruiscono l'oggetto.
- Attivi : emettono radiazioni elettromagnetiche, tipicamente luci, ultrasuoni o raggi x. Alcuni esempi:
 - time-of-flight : inviano un impulso laser e misurano il tempo di ritorno
 - triangolazione : inviano un impulso laser, usando l'angolo di riflessione e il disallineamento ottico tra emettitore e ricevitore, permettono di ricostruire la distanza punto-punto.
 - scanner a luce strutturata : viene proiettata un'immagine geometrica nota su un oggetto tridimensionale e vengono analizzate le deformazioni
 - scanner a luce modulata : viene proiettata una luce modulata su un oggetto tridimensionale e vengono analizzate le deformazioni

Un Inertial Measurement Unit (IMU) è un dispositivo elettronico che misura e riporta l'accelerazione, la velocità angolare e l'orientamento di un corpo rigido. Un IMU è composto da un accelerometro, un giroscopio e a volte un magnetometro.

I dispositivi indossabili sono device che possono essere indossati dall'utente nei quali l'interfaccia e l'unità computazionale sono uniti e spesso sono specializzati nella raccolta di certi dati biometrici o di movimento. Un dispositivo heart rate monitor permette di misurare il battito cardiaco dell'utente utilizzando dei sensori PPG che tipicamente illuminano la pelle e analizzano la luce riflessa. Invece i sensori ECG usano dei segnali elettrici per misurare l'espansione e la contrazione del cuore. Un dispositivo EEG headset permette di monitorare gli impulsi elettrici del cervello posizionando elettrodi non-invasivi sullo scalpo dell'utente.

Natural User Interface (NUI)

La NUI è un'interfaccia utente naturale e semplice da usare con interazione diretta e consistente con il comportamento naturale. Una interfaccia utente naturale è una interfaccia che è effettivamente invisibile, e resta invisibile mentre l'utente impara progressivamente ad avere interazioni via via più complesse. Ci sono dei gesti di tipo touch che oramai fanno parte del background culturale, come il tocco, la lunga pressione, lo scroll, il pinch e lo swipe. Sono diventati intuitivi per gli utenti poiché le interfacce restituiscono il feedback associato molto velocemente. La parola "naturale" viene usata in riferimento al goal di progettazione delle interfacce: si ambisce a offrire un'esperienza che non richieda la comprensione di come una macchina funzioni; si punta a un'interfaccia che si comporti come un oggetto fisico

Poiché una NUI sia considerata tale si richiedono:

- apprendimento progressivo
- expertise istantanea
- interazione diretta
- basso carico cognitivo

una strategia per realizzare NUI è l'uso della Reality User Interface (RUI), anche conosciuta come Reality-Based Interface (RBI). Un esempio di RUI è l'uso di device indossabili per rendere clickabili oggetti del mondo reale. Invece un esempio di NUI non basata su RBI è limitare le funzionalità e le personalizzazioni in modo che gli utenti abbiano ben poco da imparare.

Graphical User Interface (GUI)

struttura dell'interfaccia

- struttura gerarchica
- struttura sequenziale
- struttura matriciale
- struttura a database

architettura dell'informazione e natura degli utenti, contenuti e contesti

componenti principali:

- schemi o strutture organizzative
- sistemi di labelling
- sistemi di navigazione
- sistemi di ricerca

tipi di schemi organizzativi:

- esatti
 - schema alfabetico

- schema cronologico
- schema geografico
- soggettivi
 - topic scheme
 - task scheme
 - audience scheme
 - metaphoric scheme

creare ibridi di schemi organizzativi

navigazione nelle interfacce, principi base di findability e discoverability

modelli di comportamento degli utenti per la ricerca di informazioni:

- quit
- narrow
- expand
- pearl growing
- pogosticking
- trashing
- berry picking

antipattern

design pattern:

- autocomplete
- autosuggest
- instat result
- did you mean
- autocorrect
- best first
- partial matches
- related seraches
- federated search
- faceted navigation
- advanced search
- scoped search
- personalization
- pagination
- actionable result
- comparing result
- unfied discovery

layout di interfacce e componenti Document Object Model (DOM): cross-platform e interfaccia language-independent che permette ai programmi e agli script di accedere e aggiornare il contenuto, la struttura e lo stile dei documenti. Rappresenta il documento come un albero di nodi. Quando una pagina web JavaScript è caricata, viene creato un Document Object Model (DOM) della pagina che è una rappresentazione object oriented del documento HTML che si interfaccia tra JavaScript e il documento dando la possibilità di:

- aggiungere, cambiare e rimuovere elementi HTML
- cambiare stili CSS
- reagire agli eventi
- creare nuovi eventi

gli elementi dell'interfaccia sono:

- elementi di input
- elementi di navigazione
- componenti informativi
- container

UX design

identificare le personas mediante:

- task analysis
- feedback
- prototipazione

dipendentemente dai dati che abbiamo possiamo avere 3 tipi di personas:

- proto-personas
- qualitative personas
- statistical personas

principio di Pareto

tipi di informazioni di una personas:

- demografiche
- personali
- attitudinali e cognitivi
- obiettivi e motivazioni
- comportamentali

personas e archetipi

requirements

- funzionali
- non funzionali

user stories

è una breve dichiarazione o astrazione che identifica l'utente e il suo bisogno/obiettivo. È un requisito espresso dalla prospettiva di un dell'obiettivo dell'utente. aiutano a documentare informazioni pratiche riguardo gli utenti e aiutano gli sviluppatori a tracciare una roadmap. struttura: As a 'role', I want 'feature' because 'reason'. tutti possono scrivere user stories ad ogni livello di dettagli. i dettagli possono essere aggiunti splittando le user stories in multiple user stories o aggiungendo condizioni di soddisfazione.

scenarios

uno scenario è una situazione che cattura come gli utenti interagiscono con un prodotto.

un buon scenario deve rispondere:

- chi è l'utente?
- motivazione e aspettativa dal prodotto?
- qual'è il suo obiettivo?

grazie agli scenarios possiamo determinare:

- i punti importanti durante progettazione per l'UX
- fasi del processo che richiedono ulteriore revisione e attenzione
- le principali esigenze e motivazioni dell'utente

modi per scrivere scenarios:

- goal o task orientati agli scenarios
- elaborated scenarios
- full scale task scenarios

casi d'uso

è una descrizione scritta di come un utente interagisce con un sistema. ogni caso d'uso è rappresentato come una sequenza di passaggi che iniziano con l'obiettivo dell'utente e terminano quando l'obiettivo è raggiunto. un caso d'uso aggiunge valore perché aiuta a spiegare come il sistema dovrebbe comportarsi e forniscono una lista di obiettivi.

scenarios vs casi d'uso

uno scenario richiede una situazione che può avere uno o più attori che intraprendono una determinata funzionalità. un caso d'uso coinvolge un attore e il flusso che un particolare attore prende in una determinata funzionalità o percorso.. la differenza principale è la prospettiva.

includono:

- l'utente
- cosa vuole fare
- il suo scopo
- step necessari per raggiungere lo scopo
- feedback
- trigger
- basic flow
- alternative flow

non includono:

- dettagli implementativi o di scelta tecnologica
- dettagli di UI

i passaggi da seguire per la creazione di un caso d'uso sono:

1. identificare le personas
2. sceglierne una per caso d'uso
3. identificare il suo scopo
4. discenderne i task principali da quelli secondari
5. considerare le sequenze alternative
6. accoppiare punti in comune tra in vari casi d'uso
7. ripetere per tutte le personas

Front-end design Wireframing

wireframe: per trasmettere la tua idea alle altre persone, si creano progetti che aiutano la comunicazione tra designer e sviluppatori attraverso sketch basici che mostrano la struttura generale.

tipi di wireframe:

- low fidelity: semplici sketch che si concentrano sulla struttura
- high fidelity: rappresentazioni dettagliate con una basica interfaccia e icone.

contesto ha un importante ruolo nell'UX design per adattarci ai comportamenti e alle preferenze degli utenti.

adattabilità è la capacità di un sistema di adattarsi a diversi contesti senza cambiare la sua struttura e copre le nozioni di:

- responsive design
- accessibilità

il responsive design è un approccio di progettazione che permette ai siti web di adattarsi a diversi dispositivi e dimensioni dello schermo per avere una buona esperienza coerente con tutti i dispositivi.

tecniche di responsive design:

- flexible grid
- flexible images
- media queries & breakpoints

una filosofia di repesive design è mobile first che consiste nel progettare prima per i dispositivi mobili e poi per i desktop e ciò ci fa concentrare prima sulle funzionalità base. Ciò si oppone alla degradazione in cui si sviluppa prima per i desktop per poi rimuovere alcune funzionalità per adattarsi ai dispositivi mobili perdendo però consistenza dei vari dispositivi.

altre best practise del responsive design:

- keep it simple : design minimal
- priorizzare il contenuto
- progettare per il touch : bottoni grandi
- ottimizzare tempi di caricamento : ridurre componenti pesanti come immagini
- testare su diversi dispositivi

Adaptive design è un'altra alternativa al responsive design che consiste nel creare un'interfaccia che si adatta automaticamente ai vari utenti in modo fluido e flessibile. Solitamente realizzata creando più versioni del sito web per ogni dispositivo.

Adaptive UX ottiene informazione basiliche tra i vari utenti e le usa per adattare l'interfaccia e consigliare ad ogni utente cosa è migliore per lui attraverso il collaborative filtering e il content based filtering.

Nell'UX design è cruciale anche l'accessibilità che consiste nel rendere il prodotto accessibile a tutti gli utenti indipendentemente dalle loro capacità fisiche o cognitive in modo che ogni utente possa accedere a tutte le funzionalità del prodotto. Alcuni problemi di accessibilità sono:

- visuale
- motoria
- uditiva
- convulsioni
- cognitive

L'azienda W3C ha posto degli standard e delle linee guida per l'accessibilità chiamate Web Content Accessibility Guidelines (WCAG) e consigliano:

- alternative testuali per contenuti non testuali come immagini e grafici
- sottotitoli o altre alternative per contenuti multimediali
- diversi modi per usufruire dei contenuti
- contenuti facili da vedere e da ascoltare
- utenti possono usare altre modalità di input oltre la tastiera
- struttura facile da navigare e da capire

Oltre ai wireframes che ci aiutano a comunicare le nostre idee, abbiamo bisogno di uno schema di navigazione che ci aiuti a capire come gli utenti possono navigare nel nostro sito web. Lo facciamo attraverso gli user flows che sono una rappresentazione visuale di come gli utenti possono navigare nel nostro sito web. I più comuni sono:

- flowchart è un diagramma di flusso che mostra le varie possibili scelte che un utente può fare e le conseguenze di queste scelte.
- wireflows esprimono un diagramma di flusso usando wireframe invece di descrizioni astratte

Metodi e strumenti per l'innovazione

Un'innovazione è qualcosa di originale e utile che entra nel mercato e che cambia il modo in cui le persone vivono e lavorano. Spesso è legata al mondo delle invenzioni ma non necessariamente.

sustaining innovation cioè migliorare un prodotto esistente:

- step by step
- basso rischio
- bassa velocità
- non cambia organizzazione aziendale
- non ha bisogno di nuove competenze da parte degli utenti
- basse probabilità di scalare il mercato
- target di mercato stabile

disruptive innovation, usato per creare nuovi mercati cerca di non basarsi sulle tecnologie esistenti per raggiungere utenti che non sono serviti dai prodotti esistenti

human centered design process cerca di sviluppare un sistema che sia utile e utilizzabile concentrandosi sugli utenti cercando di migliorare l'efficacia e l'efficienza.

Il product management si occupa del cosa mentre il product development si occupa del come. Lavorano insieme, il product development team prende le specifiche dal product manager e le implementa in un prodotto funzionante.

Ideo ha sviluppato un processo di innovazione human centered che si basa su 3 fasi:

1. ispirazione : approfondisce i bisogni e le richieste degli utenti per migliorare uno strumento, osservando come viene usato.
2. ideazione : si cerca di interpretare le conoscenze assunte per arrivare a qualcosa di più tangibile tramite la creazione di un semplice prototipo. non definitivo e neanche perfetto ma utile come punto di partenza per continuare a testare e fare considerazioni.
3. implementazione : si cerca di capire come il prodotto può essere implementato nel mondo reale e come può essere scalato.

Ciascuna di queste fasi è svolta dal team con un approccio a fisarmonica, seguendo uno schema detto "Double Diamond" in cui all'inizio si lavora per produrre idee e soluzioni in quantità (divergenza) per poi concentrarsi su quelle più promettenti (convergenza).

Design thinking è un approccio all'innovazione che poggia le sue fondamenta sulla capacità di risolvere problemi complessi utilizzando una visione e una gestione del progetto basata sulla creatività. Si basa su un processo iterativo che cerca di capire l'utente facendo delle assunzioni.

obiettivi:

- avvicinarsi al cliente
- favorire la creatività e generare idee
- sperimentare le idee con prototipi

fasi principali:

1. empatizzare : identificare il problema e l'obiettivo, scrivendo le personas
2. definire : delineare i bisogni degli utenti e le varie categorie di utenti, facendo user stories
3. ideare : ricercare una soluzione al problema, facendo brainstorming
4. prototipare : realizzare i primi prototipi
5. testare : testare i prototipi sul campo e raccogliere feedback dagli utenti

Lo Human centered design è un mindset, un modo di pensare e di approcciarsi allo sviluppo di prodotti, il Design thinking invece è un vero e proprio metodo di lavoro organizzato per fasi che consente di sviluppare prodotti centrati sull'utenza grazie alle tecniche orientate alla stimolazione della creatività e alla produzione di idee. Human centered design e Design thinking sono compatibili e combinabili in un metodo detto Social Enterprise Thinking

Metodi di sviluppo per prodotti innovativi

agile, scrum e devops

modello waterfall è un metodologia lineare per lo sviluppo di progetti in cui ogni fase ricasca sulla successiva ed è strutturato in 5 parti:

1. analisi dei requisiti
2. progettazione
3. sviluppo
4. collaudo
5. manutenzione

Il modello agile è un metodo di sviluppo software che si basa sul creare e rispondere al cambiamento e pone le fondamenta sui 12 principi del manifesto per sviluppo software agile. Una cosa che contraddistingue agile è il fatto di concentrarsi sulle persone che fanno il loro lavoro e come lavorano insieme. Le soluzioni vengono raggiunte attraverso la collaborazione tra team multidisciplinari e auto-organizzati.

Il modello scrum è una sottocategoria agile che utilizza un metodo interattivo ed un approccio incrementale per ottimizzare la prevedibilità e per controllare il rischio. Il modello è particolarmente utile per progetti complessi e con un piccolo numero di persone e funziona sul dividere il progetto in blocchi rapidi per raggiungere piccoli obiettivi alla volta chiamati sprint che non sono più lunghi solitamente di due settimane. Il team traccia i propri progressi in meeting giornalieri di 15 minuti chiamati daily scrum. Un aspetto forte dello scrum è la possibilità di cambiare idea e di adattarsi ai cambiamenti del mercato o alle emergenze poiché il team non si pone il problema di pensare a tutto fin dall'inizio ma si concentra su piccoli obiettivi alla volta. Ogni sprint inizia con una pianificazione degli eventi e stabilendo gli obiettivi e termina con una revisione di ciò che è stato fatto nel periodo prestabilito, adattando il prossimo sprint in base ai risultati ottenuti. Ci sono 3 ruoli principali in scrum:

- product owner : responsabile del prodotto e del suo sviluppo
- scrum master : responsabile del processo e del team
- team : responsabile dello sviluppo del prodotto

modello devops è sottocategoria agile per cloud che ha la necessità di avere una gestione più flessibile, affidabile, sicura e controllabile gestione dei rilasci con cicli di rilascio più brevi e frequenti.

Prototipazione di interfacce

Ogni anno le compagnie lanciano migliaia di nuovi prodotti di tutti i tipi nel mercato, ognuno dei quali seguito da un team che sperano nel suo successo, ma la maggior parte di questi prodotti fallisce, circa l'80% fallisce. Come nella legge criminale, una persona è innocente finché non vengono trovate prove a suo carico, un prodotto è un fallimento finché non viene dimostrato il contrario. L'unico metodo per combattere la legge del fallimento del mercato è quello di testare oggettivamente le idee prima di investire tempo e denaro in un prodotto che potrebbe non avere successo. Il prototyping ci offre strumenti e tecniche di cui abbiamo bisogno per validare un'idea con il minimo delle risorse e in tempi brevi.

Lost in Translation Problem: Un'idea è un'astrazione che immagini nella tua testa. Nel momento in cui provi a comunicarla ad un'altra persona si ha un problema di traduzione, specialmente quando la tua idea è differente da quella dell'altra persona.

Prediction Problem: anche se l'audience capisce la tua idea, non è detto che la capisca allo stesso modo in cui la capisci tu. Ognuno ha la propria esperienza e la propria visione del mondo e questo può portare a fraintendimenti.

I falsi positivi sono un problema perché ti fanno credere che il tuo prodotto sia un successo e ti fanno perdere soldi e tempo mentre i falsi negativi ti fanno credere che il tuo prodotto sia un fallimento e ti fanno perdere un'opportunità. Per minimizzare i falsi positivi e i falsi negativi bisogna collezionare dati oggettivi e l'unico modo per farlo è quello di trasportare le proprie idee da thoughtland ad actionland. Nella thoughtland usi idee astratte su cui ti poni domande e ricevi opinioni. Nella actionland usi artefatti concreti su cui svolgi azioni e collezioni dati.

Un prototipo ti aiuta a fallire più velocemente ma spesso non velocemente abbastanza e comunque con un costo non trascurabile. Tra un'idea astratta e un prototipo c'è un'area grigia in cui si può fare prototyping. I prototipi ci danno la possibilità di collezionare dati per il mercato che ci aiutano a decidere se iniziare ad investire sull'idea. Un prototipo è un mock-up di un prodotto o servizio che può essere costruito in poco tempo.

obiettivi del prototyping sono aiutare a:

- identificare funzionalità chiave
- decidere le funzionalità da implementare nel mockup
- test sui mockup e collezione dei feedback e dati
- analisi dei dati e determinare il prossimo passo

i 7 pilastri del prototyping:

- obbedire alla legge del fallimento del mercato
- assicurarsi di star costruendo il prodotto giusto
- non perdersi in chiacchiere, idee o opinioni
- fidarsi solo dei propri dati
- fare prototyping
- parlare con i numeri e con i fatti
- pensa globalmente, testa localmente

Flusso del prototyping

1. isolare l'assunzione chiave : qual è l'assunzione o funzionalità chiave che se non funziona il prodotto fallisce?
2. scegliere un tipo di prototype : qual è il tipo di prototyping che permette
3. fare ipotesi di mercato (ipotesi XYZ)
4. testare il prototype : mettere il prototipo nel mondo reale e vedere quante persone sono interessate e quante ci interagiscono

5. imparare, rifinire, hyperzoom : valutare i risultati, rifinire il prototipo con i nuovi dati e se l'ipotesi ha retto decidere altre situazioni in cui testare il prototipo per avere una visione completa.

tipi di prototyping

- fake door : un marketing entry point per un prodotto che non esiste ancora e può essere utilizzato per pubblicizzare un servizio non ancora pronto e misurare l'interesse degli utenti. Serve per capire se l'oggetto che si vuole sviluppare può avere successo e risparmiare tempo e denaro. Si può utilizzare questo prototyping quando l'idea può essere descritta in poche e semplici parole, senza possedere nulla di fisico.
- mechanical turk : un servizio che permette di simulare e trasmettere l'esperienza del prodotto finale ad un utente senza che esso sia sviluppato per mostrare la reale esperienza che l'utente avrà con il prodotto.
- impersonator : è un prototipo che riesce a far sperimentare un'esperienza all'utente in modo estremamente economico e con lavoro minimo dietro.
- pinocchio : è un prototipo chiaramente falso che serve per veicolare un messaggio così distante dalla realtà che è faticoso e difficile da spiegare in altri linguaggi naturali, usato spesso per testare l'interesse e il possibile uso di prodotti innovativi non ancora lanciati da nessuno, nemmeno in maniera simile, su mercato.
- one night stand : una tecnica di veicolazione di un prototipo che consiste in un market test che offre solo l'esperienza senza alcun altro tipo di infrastruttura
- facade : è un impersonator ma è usato per dare un'immagine dell'azienda e non del prodotto stesso e viene spesso usato per promuovere servizi.

Dopo aver superato le varie fasi di prototyping ed aver accumulato una sicurezza sufficiente per il successo del prodotto, lo step successivo è quello di produrre il Minimum Viable Product cioè la versione minimale del prodotto contenente solo ed esclusivamente le features che si sono prototipate attraverso la fase precedente. Non si hanno le mani sul prodotto definitivo ma su qualcosa di vendibile, in modo da ottenere del ricavo e dell'utile che se sufficiente permetterebbe la produzione del prodotto finale.

Prototipi e Mock-Ups

Un mock-up è una rappresentazione con alta fedeltà del prodotto che include la maggior parte dei dettagli visibili e gli elementi di design del wireframe. Quindi i mock-ups possono essere usati per rifinire e far notare gli elementi visuali di design ma peccano di funzionalità e perciò non sono adatti a fare testing. I mock-ups vengono usati dai designer per comunicare meglio i loro concetti di design e le funzionalità in modo più efficiente ai responsabili e ai clienti.

Un prototipo è un'implementazione incompleta o approssimativa di un prodotto su cui è possibile interagire prima di rilasciare il prodotto finale. C'è sempre bisogno di un prototipo poiché alla prima prova di prodotto si trovano spesso molti problemi e ciò ci permette di testare e rifinire il prodotto prima di rilasciarlo. I vantaggi di creare un prototipo sono:

- Creare delle solide fondamenta per l'ideazione del prodotto finale e dare al responsabile una chiara idea dei potenziali benefici, rischi e costi.
- Cambiamenti possono essere fatti in modo più economico e veloce
- I feedback degli utenti aiutano ad identificare gli aspetti su cui lavorare
- La prototipazione ci dà la possibilità di sperimentare per le necessità e i problemi degli utenti in modo da migliorare
- I responsabili si legano più facilmente al prodotto
- Time-to-market migliorato grazie alla riduzione degli errori prima del rilascio

Un prototipo assomiglia al prodotto finale in vari gradi di fedeltà che solitamente sono 3:

- Low-Fidelity Prototypes : solitamente fatti con carta e penna, connette i vari low-fidelity wireframes, molto veloce da fare e facile da modificare. Non molto realistico e solitamente sono versioni molto acerbe semplificando molto il prodotto finale e le sue funzioni.
- High-Fidelity Prototypes : prototipi molto più avanzati, solitamente sviluppati attraverso l'uso di software specifici, danno la sensazione di prodotto finito e sono testabili. Hanno bisogno di più tempo e costi per essere creati e modificati.
- Mid-Fidelity Prototypes : sono un compromesso tra i due prototipi precedenti, sono più realistici dei low-fidelity ma più veloci da creare e modificare dei high-fidelity.

Dimensioni di fedeltà di un prototipo:

- Realismo : quanto il prototipo è simile al prodotto finale
- Scope : ampiezza e profondità del prototipo
- Funzionalità : quali e come le funzionalità sono implementate
- dati : se il prototipo funziona su dati reali o simulati
- autonomia : come il prototipo può operare senza il bisogno dell'utente
- piattaforma : se il prototipo è un'implementazione provvisoria o definitiva

In particolare lo scope ti dà informazioni su quanto buono potrà essere il prototipo. Esistono due tipi di scope:

- Orizzontale : ci dà una visione d'insieme del sistema, non andando nei dettagli di ogni feature
- Verticale : ci dà una visione approfondita di una singola feature

Tecniche di prototyping:

- Paper Prototypes : è un low-fidelity prototype, creato su carta e rappresentano il layout base e le funzionalità di un prodotto o servizio, rimanendo una veloce e senza costi via per testare l'interfaccia utente e rifinire prima di investire tempo e risorse in un prototipo più avanzato.
- Wireframes Prototypes : sono prototipi digitali che rappresentano il layout e le funzionalità di un prodotto o servizio, senza molti dettagli visivi. Richiede creare sketch del layout e delle funzionalità base usando semplici forme e simboli che rappresentano i vari elementi dell'UI
- Wizard of Oz Prototypes : sono prototipi che simulano il comportamento dei prodotti o servizi usando input umani invece delle tecnologie e vengono spesso utilizzati quando queste tecnologie non sono ancora sviluppate per capirne la reale utilità. Ciò richiede spesso più persone per simulare che svolgono interazioni dirette.
- Functional Prototypes : sono prototipi che includono funzionalità reali del prodotto e sono usati per rifinire gli aspetti tecnici del prodotto. Richiedono la costruzione di prototipi con codice e hardware o strumenti che danno la possibilità di interagire con gli elementi. Sono particolarmente utili per gli stage più avanzati di sviluppo del prodotto.
- Non-Functional Prototypes : sono prototipi che sembrano esattamente come il prodotto reale ma non hanno funzionalità reali e sono usati per testare il design e l'esperienza utente del prodotto prima di investire ulteriori risorse. Un po' di codice sarà comunque necessario ma senza andare nel dettaglio. Possono essere usati per testare sia l'esperienza dell'utente oppure per testare il design visuale del prodotto.

Wireframes vs Mockups vs Prototypes

Le persone spesso fanno confusione tra i vari concetti. I prototipi sono versioni più avanzate dei mock-ups che includono funzioni di navigazione e funzioni. I designers solitamente usano i mock-ups e i wireframes per creare il prototipo finale. I prototipi sono quindi usati per fare testing prima di muoversi nella fase finale di sviluppo.

Se è clicabile e interattivo oppure risponde alle azioni in altro modo è un prototipo funzionale, altrimenti se ha design, colori e immagini è un mock-up, altrimenti è un semplice wireframe.

Industria 4.0

Quando pensiamo al design di un prodotto connesso, cerchiamo di concentrarci sugli elementi più visibili e tangibili, l'industrial design e la user interface. Ma ci sono maggiori preoccupazioni che impattano la user experience. Possiamo avere una bellissima interfaccia ma l'utente potrebbe avere comunque una cattiva esperienza. Per questo è importante considerare anche l'industrial design, l'interazione tra i dispositivi e l'interazione tra l'utente e il dispositivo.

Con l'internet of things (IoT) i prodotti sono provvisti di proprio identificatore sono connessi tra loro e con il mondo esterno, questo permette di creare nuove esperienze e nuovi servizi.

Negli ultimi secoli l'industria si è sempre evoluta e ha avuto molte fasi:

- Industria 1.0 : meccanizzazione con utilizzo di energia a vapore
- Industria 2.0 : produzione in serie con utilizzo di energia elettrica
- Industria 3.0 : automazione con utilizzo di elettronica e informatica
- Industria 4.0 : smart factory

L'industria 4.0 descrive l'organizzazione di un sistema di produzione intelligente che si basa tecnologie e dispositivi che interagiscono tra loro e una fabbrica intelligente in cui i sistemi computerizzati monitorano processi fisici creando una copia del mondo fisico e prendendo decisioni in autonomia. In futuro, le aziende di successo utilizzeranno l'Internet of Things per ottenere una nuova crescita attraverso tre approcci: incrementare i ricavi aumentando la produzione e creando nuovi modelli di business ibridi, sfruttare le tecnologie intelligenti per alimentare l'innovazione e trasformare la loro forza lavoro.

Un digital twin è una replica digitale di un'entità fisica che vengono collegate con un ponte che gli fa scambiare dati ma esistere simultaneamente.

un prodotto smart è un prodotto fisico che ha un servizio digitale al suo centro.

Fare design per un prodotto delle Internet of Things è interamente più complesso del design di un servizio web.

Fare il design di un prodotto IoT richiede un approccio multidisciplinare dato che ci sono vari aspetti del design da considerare non immediatamente visibili. UX per IoT richiede una natura specializzata, un'abilità di connettere mondo digitale con mondo fisico e il fatto che i prodotti IoT sono distribuiti su più dispositivi diversi. Molte cose dell'IoT sono dispositivi informatici integrati e specializzati, che differentemente dai computer, i loro software e hardware sono ottimizzati e compiono solo specifiche funzioni. La loro UI non è semplicemente a livello software ma spesso è estesa con schermi e bottoni per l'uso di vari controlli e ciò cambia le interazioni che hanno gli utenti con il device che non hanno più il beneficio delle guide che avevano sul web.

nuovi principi del design

- UX per IoT
- device specializzati con differenti capacità
- far interagire i dispositivi tra loro
- controllo remoto
- design per network
- design per risparmio d'energia

Usability Testing

Quando abbiamo una sorta di prototipo, dovremmo fare testing con gli utenti. Questo processo ci permette di valutare il prodotto o il servizio tramite il giudizio diretto degli utenti interessati. In questo modo possiamo individuare varie problematiche da risolvere, capire cosa migliorare e cosa gli utenti preferiscono.

L'usabilità è un attributo qualitativo che ci indica quanto è facile l'interfaccia utente da usare, ed è definita in 5 componenti:

- apprendibilità : quanto è facile per gli utenti completare compiti basilari la prima volta che usano l'interfaccia
- efficienza : velocità con cui gli utenti completano compiti dopo che hanno imparato l'interfaccia
- memorabilità : quanto è facile per gli utenti riprendere a usare l'interfaccia dopo un periodo di non utilizzo
- errori : quanti errori fanno gli utenti, quanto gravi sono e come possono recuperare
- soddisfazione : quanto piace agli utenti usare l'interfaccia

Nell'UX design, un altro concetto chiave è l'utility che si riferisce alle funzionalità e alle caratteristiche di un prodotto o servizio che soddisfano i bisogni degli utenti.

Usabilità e Utility sono due concetti diversi ma correlati e devono essere sempre buoni entrambi poiché importa poco se qualcosa è facile da usare se non è utile e non è buono se un sistema può ipoteticamente fare quello che vuoi ma è difficile da usare. Quindi, per far sì che qualcosa sia utile, c'è bisogno che abbia una buona usabilità e utility. Quindi testare è essenziale per creare prodotti che incontrano le necessità degli utenti e le aspettative del principio dell'HCD. Come metodologia di ricerca dei problemi definiamo un usability-testing session in cui un ricercatore(chiamato facilitatore o moderatore) chiede ad un partecipante di completare delle attività utilizzando il prodotto o servizio in questione mentre gli altri ricercatori osservano e prendono appunti dai feedback.

Valutazione Euristica

Prima di testare con utenti reali, si può fare altro per scoprire i problemi di usabilità. Un metodo usato per valutare le interfacce utente è la valutazione Euristica che si basa su principi stabiliti dell'usabilità. Questa valutazione tipicamente richiede un piccolo team di valutatori, poiché può essere non facile completarla per un singolo individuo e comunque non essere molto preciso. Sebbene la valutazione euristica non fornisca un approccio sistematico per risolvere i problemi di usabilità o valutare le riprogettazioni, può guidare la creazione di progetti rivisti basati sui principi violati di buoni sistemi interattivi in modo efficiente.

I 10 principi di usabilità di Nielsen:

1. Visibilità dello stato del sistema : i designer dovrebbero tenere informati gli utenti su cosa sta succedendo attraverso feedback appropriati entro un tempo ragionevole.
2. Corrispondenza tra il sistema e il mondo reale : il design dovrebbe parlare il linguaggio degli utenti, con parole, frasi e concetti familiari all'utente, piuttosto che termini orientati al sistema.
3. Controllo e libertà dell'utente : gli utenti spesso scelgono le funzioni per errore e hanno bisogno di un "uscita di sicurezza" per uscire dallo stato indesiderato senza dover passare attraverso una lunga sequenza di passi.
4. Consistenza e standard : gli utenti non dovrebbero dover chiedersi se diverse parole, situazioni o azioni significano la stessa cosa, quindi il sistema deve avere una consistenza per tutte le funzioni.
5. Prevenzione degli errori : meglio che fornire un buon messaggio di errore, è meglio progettare un sistema in modo che gli errori non possano verificarsi in primo luogo.
6. Riconoscimento anziché ricordo : minimizzare il carico di cose da memorizzare per l'utente, rendendo gli oggetti, azioni e opzioni visibili.
7. Flessibilità e efficienza di utilizzo : progettare shortcut per gli utenti esperti, consentendo loro di velocizzare le interazioni più frequenti.
8. Design estetico e minimalista : l'interfacce non dovrebbero contenere informazioni irrilevanti ma solo quelle necessarie, poiché ogni elemento aggiunto aumenta la complessità del sistema.
9. Aiuto agli utenti a riconoscere, diagnosticare e recuperare gli errori : ogni messaggio di errore dovrebbe essere espresso in un linguaggio chiaro e semplice, indicando il problema e suggerendo una soluzione costruttiva.
10. Aiuto e documentazione : un buon design non ha bisogno di altre spiegazioni, ma è sempre una buona idea fornire documentazione o aiuto.

Gli elementi per un usability testing

- **Facilitatore** : guida i partecipanti nella sessione di test, spiega cosa sta succedendo e cosa si aspetta dall'utente, fa domande e prende appunti. Si assicura che i test siano di alta qualità e i dati validi.
- **Tasks** : sono attività realistiche che i partecipanti potrebbero fare nella vita reale. Possono essere molto specifici in base alla ricerca che vogliamo fare. Possono essere sia consegnati verbalmente che scritti ma ciò che è importante è la formulazione che deve essere capita dall'utente.
- **Partecipanti** : sono le persone che partecipano al test, idealmente dovrebbero essere selezionate a partire dalle personas che abbiamo creato. Spesso gli viene chiesto dai facilitatori di pensare ad alta voce mentre completano i task in modo da capire quali sono gli step mentali per completare i task.

Ci sono 2 tipi di usability testing in base ai dati che collezioniamo

- **Dati qualitativi** : costituiti da risultati osservativi che identificano le caratteristiche del progetto facili o difficili da usare.
- **Dati quantitativi** : sotto forma di una o più metriche, indicano se le attività erano facili da eseguire.

I test qualitativi tipicamente comportano la raccolta di dati attraverso domande a risposta aperta, osservazioni e feedback per ottenere informazioni di come gli utenti interagiscono con un prodotto. Spesso il test viene svolto negli stage iniziali del progetto per capire le necessità degli utenti, gli obiettivi e i punti deboli.

I test quantitativi è un metodo di ricerca che prevede la raccolta e analisi di dati numerici per misurare il comportamento e l'atteggiamento degli utenti nei confronti di un prodotto o servizio. In genere comporta la raccolta di dati tramite sondaggi, analisi e test a risposta chiusa. L'obiettivo è quello di dare ai designer dei dati statisticamente significativi che possono essere utilizzati per prendere decisioni. Possono essere utilizzate per valutare le decisioni che sono state prese durante il processo di progettazione.

Entrambi i testing sono importanti metodi nell'UX design e si completano l'un l'altro. I test qualitativi sono utili per capire il perché di un problema, mentre i test quantitativi sono utili per capire quanto è grande il problema.

Per pianificare un usability test abbiamo di:

- **Definire gli obiettivi** : cosa vogliamo scoprire? quali sono le nostre ipotesi?
- **Definire le modalità** : decidere la location, moderarli o meno e se farli di persona o in remoto
- **Decidere numero di partecipanti** : quanti partecipanti sono necessari per ottenere risultati significativi? (Nielsen indicava 5 per studi qualitativi e 20 per studi quantitativi)
- **Scegliere i partecipanti giusti** : per avere i risultati più significativi possibili, i partecipanti devono rispettare il target d'utenza del prodotto
- **Studiare i giusti task** : i task devono essere realistici e pertinenti per il prodotto. Possono essere a domanda aperta per capire come gli utenti ragionano o specifici per capire se gli utenti riescono a completare un'azione
- **Provare il test** : prima di farlo con i partecipanti, è importante provare il test per capire se i task sono chiari e se il test è fattibile
- **Decidere le metriche** : i test qualitativi non hanno metriche poiché essendo fatti su poche persone non rappresentano l'intera popolazione mentre i test quantitativi hanno metriche ben definite per misurare l'usabilità.
- **Scrivere il piano** : una volta che abbiamo tutte le informazioni, è importante scrivere un documento per comunicare e registrare gli studi. Il documento dovrebbe contenere:
 - Nome del prodotto
 - Obiettivi dello studio
 - Logistica
 - Profili dei partecipanti
 - Tasks
 - Metriche e questionari
 - Descrizione del sistema
- **Non fare tutto da solo** : gli studi di usabilità possono essere utili a tutti i membri del team di progettazione, quindi è importante coinvolgere tutti i membri del team per avere più punti di vista e più idee.

10 Cose da evitare mentre si testa:

1. Dire agli utenti dove andare
2. Dire agli utenti cosa fare
3. Non aggiornare i dati dei task
4. Fare task troppo semplici
5. Creare uno scenario troppo elaborato
6. Scrivere task usando linguaggio di marketing o tecnico
7. Rischiarare reazioni emotive
8. Cercare di essere divertente
9. Offendere i partecipanti
10. Fare altre domande oltre a quelle dei task

Domande orale Interazione Uomo Macchina

1. differenza tra design thinking e human centered design
2. esempio di mapping
3. cos'è uno scenario? da cosa parto per definirli?
4. cos'è un prototipo? gradi di fedeltà?
5. cos'è la ppg?
6. cos'è lo scanner time-of-flight?
7. user behavior pattern? *
8. come deve essere un feedback?
9. differenza tra mockup e prototipo? *mockup
10. cos'è una imu?
11. cos'è un sistema di eyetracking e quali tecnologie conosciamo?
12. cosa sono le euristiche di Nielsen? *
13. quali sono gli errori umani?
14. prese x persone. quale sarà il numero y che fa le z cose? *
15. cos'è lo human center design process?
16. cos'è il test qualitativo? *

17. scanner 3d a luce strutturata
18. cosa sono lapsus d'azione
19. modello concettuale fa parte del sistema?
20. cos'è il paradigma?
21. cos'è una nui?
22. come si scrivono le user stories?
23. cos'è l'immagine di un sistema
24. quali sono i sette stadi dell'azione?
25. cos'è un wireframe?
26. cos'è un modello a doppio diamante?
27. cos'è un dispositivo aptico?
28. differenza affordance significativa?
29. tipi di test?
30. quali sono i principi fondamentali dell'interazione?
31. cos'è l'HID protocol?
32. differenza tra user stories e requirements?
33. parlami dei vincoli
34. come devono essere i feedback?
35. struttura di un'interfaccia *
36. la regola dei 5 perché?
37. discoverability e understanding
38. differenza tra prototipo e pretotipo
39. livelli mentali dell'essere umano
40. i 7 stadi dell'azione
41. come si dividono i sistemi di puntamento?
42. disruptive innovation
43. 6 caratteristiche principali del prototipo *