

Human Centered Design

Approccio che mira a rendere **intuitiva** l'interazione tra l'uomo e l'interfaccia del prodotto.

Le *macchine* hanno un funzionamento logico, rigido, dovuto all'algoritmo con cui sono progettate.

Gli *umani* sono tutt'altro che logici e razionali: siamo intuitivi, flessibili, curiosi.

È sempre stato preteso che gli umani si adattassero alle macchine, questo approccio **centro-tecnico** però rallenta la diffusione e l'accettazione della macchina rendendola meno accessibile a chi non ha le conoscenze richieste: le complete regole di funzionamento dei prodotti spesso sono note solo ai progettisti. Quando queste regole non vengono seguite dall'utente, le cose non vanno come previsto, creando frustrazione.

La macchina è perfetta e non può sbagliare, se le cose vanno male non è colpa dell'utente, ma del progettista!

Nel design **antropocentrico** il paradigma si inverte e l'utente viene messo al centro del processo.

[...] This approach enhances effectiveness and efficiency, improves human well-being, user satisfaction, accessibility and sustainability; and counteracts possible adverse effects of use on human health, safety and performance. - ISO 9241-210:2010(E)

Esempio di sviluppo centro-tecnico:

La **block chain**, che doveva essere la base per l'innovazione radicale di numerosi settori, in verità rimane rilevante solo per lo scambio di criptomonete (che sono nate per trovarle un'applicazione). L'applicazione ad altri settori non sta avendo successo perché non se ne sente il bisogno: stiamo cercando un problema a cui applicarla, e non una tecnologia per risolvere un problema già esistente.

Lo sviluppo **antropocentrico** inizia con una buona concezione del comportamento umano e a quali bisogni vuole far fronte la progettazione. Un buon design richiede buona comunicazione, in particolare dalla macchina all'utente: devono essere chiare le azioni da poter intraprendere, cosa sta accadendo, e cosa accadrà di conseguenza. Soprattutto, deve essere chiaro come affrontare eventuali **problemi e malfunzionamenti: l'utente deve essere guidato nella loro risoluzione.**

L'obiettivo finale è di far insorgere nell'utente **empatia** verso il sistema. **HCD = forma di pensiero.**

Procedura schematizzata per l'HCD:

1. Specificare il contesto d'uso:

a quali utenti mi sto riferendo, per cosa, con quali condizioni.

2. Specificare i requirements:

business requirements e obiettivi dell'utente.

3. Progettare la soluzione:

dalle bozze, ai prototipi, alla soluzione.

4. Testare e valutare:

iniziare il ciclo sulla base dei risultati per uno sviluppo incrementale.

Progettare l'Interazione

Esistono solo due tipi di design: riuscito e fallito. Il buon design non è universale.

Due sono le caratteristiche principali:

Discoverability – COSA?

Capacità di un sistema di comunicare i propri usi all'utente, tipicamente attraverso la visibilità (ma non è detto che una volta capito cos'è si riesca ad usarlo).

Understanding– COME?

Capacità del prodotto di farsi usare correttamente. A volte non basta capire quali sono i controlli, ma è necessario capire come usarli, che funzionamento ha ciascun controllo, e come si combinano.

"We were designing things for people, so we needed to understand both technology and people. But that's a difficult step for many engineers: machines are so logical, so orderly. If we didn't have people, everything would work so much better. Yup, that's how I used to think." – Donald Norman.

Principi Fondamentali dell'Interazione

Sensazione

Un buon design produce un'esperienza gradevole. Parlando di un'auto, ciò che rimane impresso sono le finiture, il controllo nell'accelerazione, la sensazione dei comandi. Le sensazioni sono fondamentali nel determinare un'esperienza positiva o negativa.

Percezione ed emozioni

Quando il prodotto non si comporta come pensiamo, provoca frustrazione e confusione. Quando c'è comprensione c'è sensazione di controllo, di soddisfazione. La percezione dell'oggetto è strettamente intrecciata alle emozioni umane, il designer deve quindi tenere a mente entrambe.

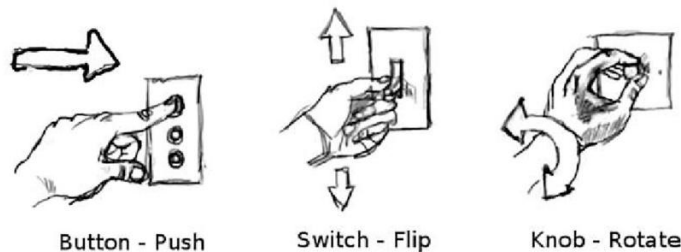
Discoverability

Nell'interazione con un prodotto dobbiamo scoprire come funziona. Ciò avviene correttamente se sono stati applicati i 6 fondamentali principali:

- Affordances
- Signifiers
- Mappings
- Feedback
- Conceptual model
- Constraints

• AFFORDANCES: *potenzialità espressive*

Il termine descrive la relazione tra l'oggetto fisico, le sue proprietà, e la capacità dell'utente di determinare le potenzialità d'uso di esso. Una sedia esprime "supporto", e quindi permette di sedersi.



Le sedie generalmente sono "sollevabili", ma a volte serve una persona particolarmente forzata per farlo: l'utente che non riesce a sollevarla non ha la stessa **relazione** con la sedia = la sedia non ha quella *affordance* per quell'utente.

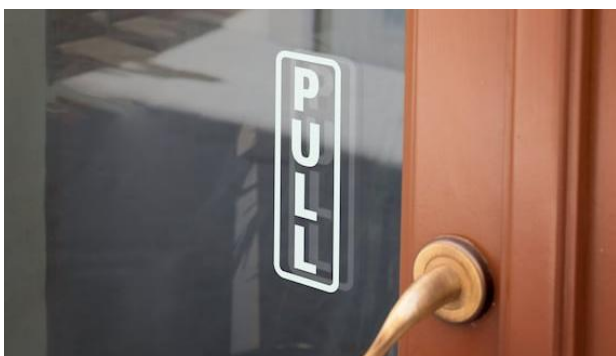
Anti-affordance: prevenzione dell'interazione. Applicare per evitare (prevenire) che l'utente possa utilizzare l'oggetto in maniera incorretta.

Le affordance e le anti-affordance devono essere discoverable and perceivable (*scopribili e percettibili*), inoltre sono interpretabili in maniere diverse a seconda delle convenzioni culturali a cui è soggetto l'utente.

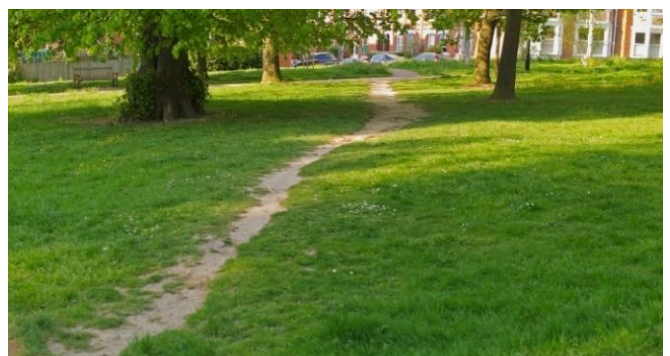
• SIGNIFIERS: *indicazioni, segni*

È l'indicazione grafica, visiva, che suggerisce il corretto utilizzo all'utente. Se le affordances indicano quali azioni sono possibili, i signifiers indicano dove devono avvenire, soprattutto per le interfacce display. Possono essere:

Voluti, intenzionali

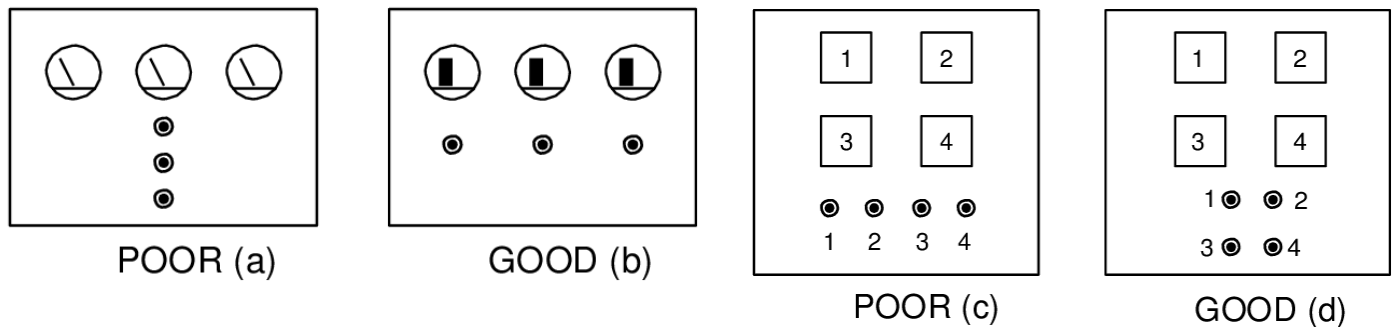


Accidentali, involontari



- **MAPPING:** *rappresentazione, mappatura*

È la relazione tra gli elementi di *due* set di oggetti. Il mapping è un concetto importante nel design e nel layout di comandi di controllo e interfacce display. Quando vengono sfruttate le corrispondenze tra la disposizione del set di comandi e il set dei “comandati”, risulta semplice e immediato operare il prodotto: questo è il **natural mapping**.



Alcune mappature naturali sono **culturali** o **biologiche**, inteso come lo standard universale dove “su” indica “di più” e “giù” indica “di meno”, che rende naturale usare un comando verticale per indicare il controllo di intensità.

- **FEEDBACK:** *riscontro*

È ciò che informa la corretta rilevazione o il risultato del comando effettuato dall’utente. Deve essere **immediato** e **informativo**. Un feedback pensato male può fare più danni di un feedback assente, perché è distraente e fuorviante. Il riscontro è essenziale ma non quando è esagerato e/o distrae dallo scopo principale.

- **CONCEPTUAL MODEL:** *modello concettuale*

È l’illustrazione semplificata del funzionamento di un oggetto. Non ha bisogno di essere completa o perfettamente accurata finché è utile: i file e le cartelle presenti sul desktop aiutano l’utente a creare il modello mentale di “documenti ordinati in cartelle” all’interno del computer, anche se non esistono veramente nella macchina la comprensione è facilitata.

I **modelli mentali** sono la rappresentazione del modello concettuale all’interno della testa degli utenti, persone diverse possono sviluppare modelli mentali differenti per uno stesso modello concettuale, o la stessa persona può gestire più modelli mentali per diversi aspetti del concetto.

L’utente non comunica con il designer, quindi si affida alle informazioni che riesce a reperire: l’aspetto dell’oggetto, l’esperienza costruita sull’utilizzo di strumenti simili, il sito o il manuale del prodotto. La combinazione delle informazioni reperibili è il **system model**. Un modello mentale errato, e quindi l’interpretazione sbagliata del modello del sistema, porterà a un uso incorretto del prodotto.

- **CONSTRAINTS:** *limiti e vincoli*

Forniscono indicazioni importanti limitando l’insieme delle azioni possibili. Esistono 4 classi universali:

- Physical
- Semantic
- Cultural
- Logical

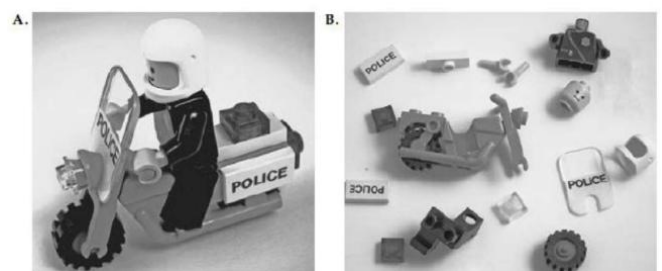


FIGURE 4.1. Lego Motorcycle. The toy Lego motorcycle is shown assembled (A) and in pieces (B). It has fifteen pieces so cleverly constructed that even an adult can put them together. The design exploits constraints to specify just which pieces fit where. Physical constraints limit alternative placements. Cultural and semantic constraints provide the necessary clues for further decisions. For example, cultural constraints dictate the placement of the three lights (red, blue, and yellow) and semantic constraints stop the user from putting the head backward on the body or the pieces labeled “police” upside down.

Le **forcing functions** sono un caso estremo di vincoli fisici che impediscono comportamenti inappropriati: sono situazioni in cui un errore non risolto impedisce il passaggio alla fase successiva.

Interlocks

Forza le operazioni a susseguirsi in un determinato ordine.
Es. l'obbligo di inserire la chiave di accensione prima di poter avviare un'auto.

Locks-Ins

Mantiene attiva un'operazione, impedendo a qualcuno di interromperla prematuramente.
Es. i messaggi che compaiono quando si cerca di chiudere un'applicazione senza aver salvato il lavoro

Lock-Outs

Impedisce l'accesso a un'area pericolosa o il verificarsi di un evento.
Es. i pop-up che richiedono di verificare la maggiore età per accedere a un sito.

NOTE:

1. Le **affordances** e i **signifier** sono estremamente soggetti a fattori culturali. C'è resistenza al cambiamento: le persone tendono a opporsi all'introduzione di nuovi approcci che violano le convenzioni consolidate, perché richiedono un nuovo apprendimento. Se una nuova soluzione è solo leggermente migliore di quella esistente, è preferibile mantenere la coerenza. Se si introduce un cambiamento, è importante che sia adottato da tutti, per evitare sistemi misti e confusione.
2. Come determiniamo un'azione dipende dalla combinazione delle conoscenze universali con quelle mentali: le **conoscenze universali** includono le affordances i signifiers, i mappings, e i vincoli fisici. Le **conoscenze mentali** includono i modelli concettuali, i vincoli culturali, semantici e logici, le analogie tra esperienze passate e attuali.
3. La mancanza di comunicazione tra gli organi di produzione è spesso la causa principale di un design imperfetto e confuso. Un buon design ha bisogno di supporto e occhio dedicato in ogni sua fase.

Activity-Centered Control

È un'estensione del HCD, si concentra sulle azioni che un'utente tende a intraprendere in relazione a un prodotto. È un modello, non un processo, e quindi un'ulteriore prospettiva che può essere presa in considerazione quando si progetta.

Un telecomando universale come [Harmony](#) è un buon esempio di controlli centrati sull'attività: consente di creare "attività" come "Guarda la TV" o "Ascolta la musica", che comunicano direttamente con i diversi dispositivi necessari. Questo approccio semplifica l'utilizzo del sistema, permettendo all'utente di concentrarsi sull'attività che desidera svolgere anziché sulla gestione dei singoli dispositivi.

How People Do Things

Quando le persone si avvicinano ai prodotti e ai sistemi, incontrano due divari:

Gulf of Execution

Rappresenta la difficoltà che una persona incontra nel capire come utilizzare un oggetto o un sistema.

Gulf of Evaluation

Riflette lo sforzo necessario per interpretare lo stato del sistema e capire se le proprie azioni hanno avuto l'effetto desiderato

Il ruolo del designer è aiutare l'utente a navigare questi due divari, tramite:

Signifiers, constraints, mappings, and a conceptual model.

Feedback and a conceptual model.

Le azioni e le loro 7 fasi

Similarmente, ogni azione si compone in due fasi: **esecuzione** e **valutazione**. Entrambe influiscono sullo stato emotivo dell'utente.

1. **Goal** – impostare l'obiettivo.

Execution:

2. **Plan** – pianificare l'azione.
3. **Specify** – definire la sequenza di azioni.
4. **Perform** – eseguire la sequenza di azioni.

Evaluating:

5. **Perceive** – percepire il risultato.
6. **Interpreting** – interpretare il risultato.
7. **Compare** – confrontare il risultato con l'obiettivo.

Domande chiave:

1. Cosa voglio ottenere?
2. Quali sono le possibili azioni?
3. Quali posso intraprendere?
4. Come la eseguo?
5. Cosa è successo?
6. Cosa mi significa?
7. Corrisponde all'obiettivo?

Il modello a 7 fasi è utile alla progettazione, e non è rigido: non è sempre necessario seguire tutte le fasi in sequenza, e una singola attività può richiedere diverse sequenze di azioni estese nel tempo. I golfi dell'esecuzione e della valutazione sono punti di partenza fondamentali per individuare aree di miglioramento. Le 7 fasi suggeriscono 7 quesiti che dovrebbero essere chiari all'utente durante l'interazione.

Feedforward e Feedback

Feedforward: le informazioni che aiutano a capire **come** eseguire un'azione. Si ottiene con l'uso di significanti, vincoli e mappature, e con un buon modello concettuale.

Feedback: le informazioni che aiutano a capire **cosa** è successo dopo un'azione. Si ottiene fornendo informazioni chiare sull'impatto dell'azione e con un buon modello concettuale.

Opportunistic Actions

La maggior parte delle azioni, obbiettivi e intenzioni di tutti i giorni non sono ben specificate: sono **opportunistiche** anziché **pianificate**. Questo tipo di azioni si effettuano per avvantaggiarsi delle circostanze. Sono meno precise di azioni ben pianificate, ma richiedono uno sforzo cognitivo notevolmente inferiore e permettono esperienze più interessanti.

Il Pensiero Umano

TABLE 2.1. Subconscious and Conscious Systems of Cognition	
Subconscious	Conscious
Fast	Slow
Automatic	Controlled
Multiple resources	Limited resources
Controls skilled behavior	Invoked for novel situations: when learning, when in danger, when things go wrong

La maggior parte dei processi mentali avviene a livello subconscio. Questo livello, definito "**riflessivo**", emerge alla coscienza solo in situazioni specifiche, come l'apprendimento di nuove abilità. La pratica costante può portare al "sovra-apprendimento", rendendo l'esecuzione di un'azione automatica e inconsapevole.

Questi compiti riguardano tutti la **memoria a lungo termine**, ne distinguiamo due tipi:

Declarative Memory - dichiarativa
Immagazzina informazioni fattuali, come la capitale del Brasile.

Procedural Memory - procedurale
Immagazzina le procedure per eseguire azioni, come aprire una porta.

Le **emozioni** hanno un ruolo essenziale: influenzano i processi cognitivi attraverso il rilascio di ormoni che modificano l'attività cerebrale. In situazioni di stress, gli ormoni focalizzano l'attenzione su elementi rilevanti dell'ambiente, mentre in situazioni di calma favoriscono l'esplorazione e la creatività. Uno stato emotivo positivo è ideale per il pensiero creativo, ma meno efficace per portare a termine compiti specifici.

Una grezza semplificazione del funzionamento del cervello è basata sulla divisione in **tre livelli procedurali**:

Livelli:

1. **Viscerale** (*lizard brain*): gestisce i meccanismi di protezione di base, generando giudizi rapidi sull'ambiente (buono/cattivo, sicuro/pericoloso). Le risposte viscerali sono veloci, subconscie e restii a essere modificati da nuove nozioni.

2. **Comportamentale**: sede delle abilità apprese, attivate da situazioni che corrispondono a pattern noti. Le azioni e le analisi a questo livello sono in gran parte subconscie, anche se siamo consapevoli delle azioni che compiamo.

3. **Riflessivo**: livello di cognizione cosciente, comprensione profonda, ragionamento e processo decisionale. La riflessione è lenta e spesso avviene dopo che gli eventi si sono verificati. Le emozioni a livello riflessivo sono più durature e includono la colpa, l'orgoglio, la lode e la critica.

Implicazioni per il design:

Contano le sensazioni immediate, come l'aspetto estetico, il suono o la texture di un oggetto. Il design deve suscitare attrazione o repulsione a livello viscerale, indipendentemente dall'usabilità o dalla funzionalità del prodotto.

Ogni azione a livello comportamentale è associata a un'aspettativa di risultato. Il feedback è fondamentale per confermare o smentire le aspettative, positive e negative, generando soddisfazione o frustrazione.

Fondamentale per la creazione di prodotti memorabili. Le emozioni generate a questo livello influenzano la propensione a consigliare o sconsigliare un prodotto. I ricordi riflessivi possono essere più importanti della realtà stessa!

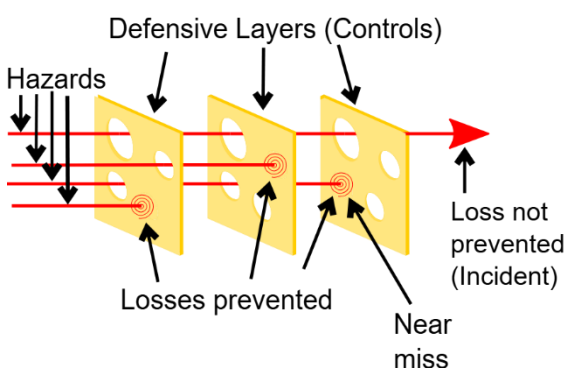
Gli Errori Umani

La maggior parte degli incidenti in ambito industriale (75%-95%) è causato dall'**errore umano**. Questo però non è dovuto al personale incompetente, ma bensì al design. Nella maggior parte dei design, chi lo utilizza deve rimanere vigile per ore in ambienti poco stimolanti, o ricordarsi procedure intricate che vengono usate raramente. Le **interruzioni** sono una causa molto comune di errori, quando si progetta pretendendo attenzione completa: riprendere esattamente dallo stato precedente è ostico.

Quando si verifica un errore, la reazione tipica è quella di cercare un "colpevole" da punire. Questo approccio non risolve il problema, che si ripresenterà ciclicamente. La soluzione, invece, consiste nell'individuare le cause dell'errore e riprogettare il sistema per eliminarle o mitigarne l'impatto.

Riconoscere l'esistenza delle falle, senza scaricare la colpa sugli utenti, è essenziale alla risoluzione. Obbligare gli utenti a inserire informazioni numeriche con precisione assoluta, ad esempio, non tiene conto dei limiti umani. L'uomo eccelle nella creatività e nell'esplorazione, non nella precisione ripetitiva.

Root Cause Analysis



Non limitarsi alla soluzione temporanea, ma ricercare la causa scatenante, non limitandosi a individuare l'azione inappropriata di un individuo. "Swiss cheese model of accidents": la maggior parte dei problemi è il risultato di molteplici fattori concomitanti

1.5 Perché: modello sviluppato dall'azienda Toyota, si basa su farsi molteplici domande sul perché di un errore, fino alla radice del problema. Il numero cinque non è vincolante, ma enfatizza l'importanza di non fermarsi alla prima risposta.

Definizioni e Tipologie di Errore

L'errore umano è definito come **qualsiasi deviazione da un comportamento "appropriato"**, tuttavia, il concetto di appropriatezza è spesso ambiguo e definito a posteriori. Generalmente, quando si parla di **errore** si parla di **azioni sbagliate**.

Slips - lapsus

Una persona intende fare un'azione ma finisce per farne un'altra. Diviso in:

Action-based: si effettua l'azione sbagliata (metto il caffè in frigo invece che il latte).

Memory-based: la memoria fallisce, l'azione prevista non viene effettuata o il risultato non viene valutato (mi dimentico di spegnere il gas).

I principianti sono più inclini a commettere errori cognitivi, mentre gli esperti sono più propensi a commettere lapsus e dimenticanze.

Mistakes – errori cognitivi

Si verificano quando si stabilisce l'obiettivo sbagliato o si elabora un piano errato, anche se le azioni vengono eseguite correttamente. Diviso in:

Rule-based: la situazione è diagnosticata correttamente, ma si sceglie un'azione errata, seguendo la regola sbagliata (un meccanico decide di non sostituire una batteria difettosa perché funziona ancora al 50%).

Knowledge-based: il problema è diagnosticato erroneamente, per conoscenze sbagliate o incomplete (calcolare in libbre invece che kg).

Memory-lapse (dimenticanza): si dimentica una fase in processo decisionale, della pianificazione o della valutazione.

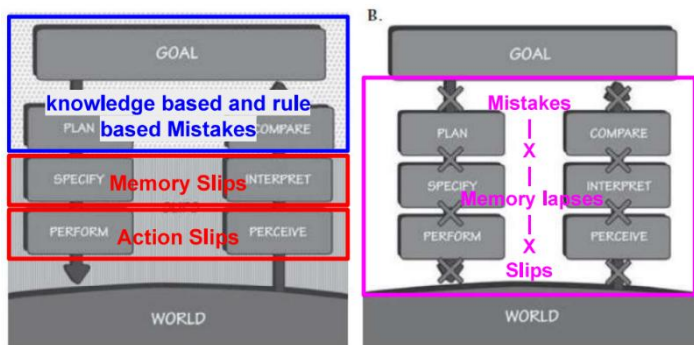


FIGURE 5.2. Where Slips and Mistakes Originate in the Action Cycle. Figure A shows that action slips come from the bottom four stages of the action cycle and mistakes from the top three stages. Memory lapses impact the transitions between stages (shown by the X's in Figure B). Memory lapses at the higher levels lead to mistakes, and lapses at the lower levels lead to slips.

Prevenire gli Errori

Un singolo errore non deve essere in grado di creare danni estesi. È necessario:

- **Comprendere** le cause degli errori e progettare per minimizzarle.
- Effettuare **sensibility checks**: plausibilità e buon senso.
- Rendere le azioni **reversibili**.
- Facilitare l'**individuazione** e la correzione degli errori.
- L'azione non è un errore: **aiutare** l'utente a completare l'azione correttamente, considerare l'errore come approssimazione dell'obiettivo desiderato.

I **vincoli** fisici e logici possono impedire gli errori: moduli e controlli separati, richiedere conferme prima di eseguire azioni irreversibili. Il comando "**Annulla**" è uno strumento potente per mitigare l'impatto degli errori, consentendo di ripristinare lo stato precedente.

Le richieste di conferma sono spesso inefficaci perché gli utenti tendono a ignorarle quando sono sicuri di voler eseguire l'azione. Un metodo migliore consiste nel mostrare chiaramente l'azione e l'oggetto su cui si sta agendo, offrendo la possibilità di annullare o confermare.

Minimizzare i lapsus: accadono quando la mente cosciente è distratta o quando l'azione è così automatizzata da non richiedere attenzione consapevole. È importante:

- **Differenziare** i controlli e le azioni il più possibile.
- Fornire un **feedback** chiaro sull'azione eseguita e sul nuovo stato del sistema.
- Offrire un meccanismo per **annullare** l'errore.

Alcuni errori di progettazione comuni favoriscono l'accadere di lapsus:

- Righe di controlli o valori identici aumentano il rischio di errori di similarità.
- Sistemi che non ammettono interruzioni favoriscono i lapsus di memoria.
- Procedure con step iniziali simili aumentano il rischio di confusione.

Mitigazione degli Errori: Il modello "*Swiss cheese*" suggerisce tre modi per ridurre gli incidenti:

- Aggiungere più fette di formaggio: implementare più livelli di difesa e controlli.
- Ridurre il numero o le dimensioni dei buchi: ridurre i punti critici in cui possono verificarsi errori.
- Avvisare gli operatori quando i buchi si allineano: implementare sistemi di allarme efficaci.

Personas, Requirements, User Stories, Scenarios and Use Cases

Quattro metodi di user research che permettono di creare rappresentazioni credibili e personalizzate degli utenti target, per progettare di conseguenza.

• PERSONAS

Le personas sono rappresentazioni fittizie di utenti reali, basate su dati di ricerca e analisi. Servono a comprendere a fondo i bisogni, le motivazioni e i comportamenti degli utenti, per essere guidati nella progettazione di soluzioni mirate. La creazione parte dalla ricerca, alcune tecniche sono:

- **Task analysis** (card sorting, first click testing)
- **Raccolta di feedback** (interviste contestuali e focus group)
- **Prototipazione** (sperimentazione di idee previo sviluppo)

In base alla quantità di dati disponibili, si possono definire tre tipi di personas:

Proto-personas

Create rapidamente con informazioni limitate, utili per avviare il processo di design.

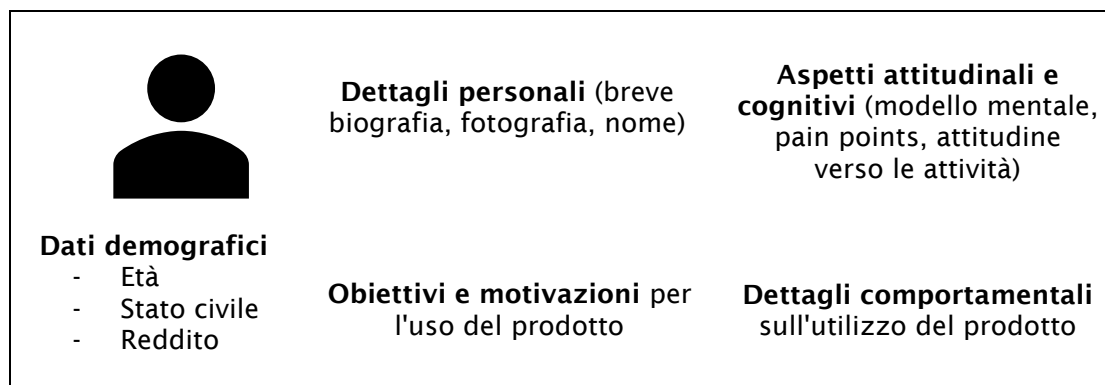
Qualitative personas

Basate su ricerche qualitative approfondite, come interviste a un campione ristretto di utenti (5-30).

Statistical personas

Derivate da analisi statistiche di sondaggi condotti su un ampio campione di utenti.

Qual è una quantità di personas adeguata? Si applica il **principio di Pareto**: focus su quel **20%** di utenti che utilizzerà l'**80%** del tuo prodotto. Informazioni che compongono l'user persona:



Gli **archetipi** sono una forma più astratta di personas, che rappresentano un tipo di utente senza dettagli specifici. Mentre le personas forniscono un volto umano all'utente tipo, gli archetipi si concentrano sui bisogni, i comportamenti e le motivazioni comuni a un gruppo di utenti.

ARCHETYPE

"The unlikely hero with humble beginnings"



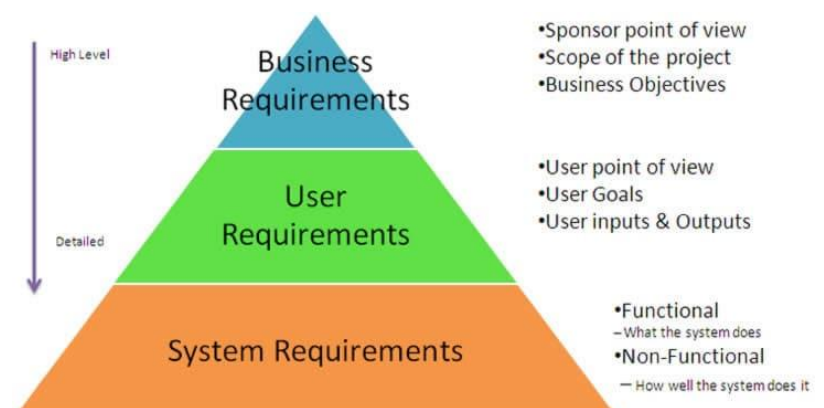
PERSONAS



• REQUIREMENTS

Descrivono le funzionalità, i vincoli, le regole aziendali e altri elementi che un sistema deve possedere per soddisfare i bisogni degli utenti.

Definire requisiti dettagliati troppo presto può essere controproducente e limitante, poiché il contesto aziendale e la comprensione del problema evolvono nel tempo.



Esistono due categorie principali di requisiti:

Functional Requirements (FRs) Esprimono le funzioni o le caratteristiche richieste dal sistema, senza specificare come implementarle.	Non-functional Requirements (NFRs) Definiscono il livello di performance o qualità del sistema, come sicurezza, affidabilità, manutenibilità, disponibilità, prestazioni e tempi di risposta.
<i>Esempi:</i>	
Visitare la sede del cliente: raggiungerla in auto è un'opzione, ma anche con il treno, in aereo, etc.	<ul style="list-style-type: none">- Risponde entro 2 secondi- È disponibile 24/7

• USER STORIES

Brevi descrizioni che identificano l'utente, il suo bisogno e l'obiettivo da raggiungere. Aiutano a comprendere le motivazioni degli utenti e a stimare la roadmap per lo sviluppo del prodotto. La struttura tipica di una user story è: "*Come [ruolo], voglio [funzionalità] perché [motivazione]*."

Possono essere scritte a diversi livelli di dettaglio, user stories che coprono un range molto ampio di casistiche sono chiamate **epics**, e vengono poi suddivise in numerose storie più specifiche.

Es. As a user, I can backup my entire hard drive

↓ *aggiungo dettagli e condizioni* ↓

As a power user, I can specify files or folders to backup based on file size, date created, and date modified.

As a user, I can indicate folder not to backup so that my drive isn't filled up with things I don't need saved.

Le personas e le user stories vengono utilizzate congiuntamente per dare priorità alle funzionalità, contestualizzare l'esperienza utente e guidare il processo di progettazione

• SCENARIOS

Descrivono come gli utenti interagiscono con il sistema per svolgere determinate attività. Forniscono contesto all'esperienza utente, evidenziando motivazioni, obiettivi e possibili ostacoli e aiutano quindi a individuare la soluzione per l'utente. Sono uno sviluppo della user story dove è implementata l'interazione con il prodotto/il servizio, ma possono anche essere scomposti in molteplici **use cases** (*casi d'uso*).

Come scrivere uno scenario:

- **Chi** è l'utente?
 - **Perché** l'utente utilizza il sistema?
 - **Quali** sono i suoi obiettivi?
 - **Come** può l'utente raggiungere i suoi obiettivi?
- 1) **Scenario Mapping:** brainstorming tra team di progettazione, sviluppatori e product owner per definire la strategia e la key task.
 - 2) **Scenario Analysis:** analisi del contesto e dei passaggi che l'utente compie per raggiungere il suo obiettivo.

Gli scenarios permettono di identificare i punti focali del processo di UX design, definire i passaggi che richiedono un supporto aggiuntivo per l'utente, comprendere i bisogni e le motivazioni degli utenti.

• USE CASES

Descrivono in dettaglio il punto di vista degli utenti riguardo sistema. Si focalizzano sul comportamento del sistema in risposta alle richieste dell'utente, scomponendo il processo in una sequenza di passaggi.

La differenza principale tra scenarios e use cases risiede nella **prospettiva**: gli scenarios offrono una visione d'insieme dell'interazione utente, mentre i casi d'uso si concentrano su flussi di azioni specifici, spesso raggruppati per coprire un intero scenario.

I casi d'uso sono utili per definire il comportamento del sistema e anticipare potenziali problemi, stabilire il costo e la complessità del sistema, negoziare le funzionalità che diventeranno requisiti.

What Use Cases Include	What Use Cases Do NOT Include
<ul style="list-style-type: none"> • Who is using the website • What the user want to do • The user's goal • The steps the user takes to accomplish a particular task • How the website should respond to an action 	<ul style="list-style-type: none"> • Implementation-specific language • Details about the user interfaces or screens.

Elementi di uno use case:

- **Attore:** qualcuno o qualcosa che utilizza il sistema.
- **Stakeholder:** chi ha interesse nell'utilizzo del sistema discusso.
- **Attore Primario: Stakeholder** che utilizza il sistema per raggiungere un obiettivo.
- **Precondizioni:** condizioni necessarie affinché avvenga quell'azione.

- **Trigger:** evento che causa l'inizio del caso d'uso.
- **Scenario iniziale:** flusso di azioni senza errori.
- **Flussi alternativi:** deviazione dal flusso principale. Queste eccezioni capitano quando qualcosa va storto nel flusso principale.

Principi fondamentali dell'HCI (*Human-Computer Interaction*)

Gli individui sono complessi: tra loro hanno differenti capacità cognitive e stili di interazione, esiste quindi la necessità di tenerne conto durante il processo di progettazione.

Modelli di HCI

- **Skill-based:** azioni eseguite senza controllo cosciente, es. la digitazione per un dattilografo esperto.
- **Rule-based:** recupero di schemi di comportamento appresi precedentemente, applicati a situazioni familiari.
- **Knowledge-based:** applicarsi a interazioni non familiari che richiedono la formulazione esplicita di obiettivi e piani.

Il sistema progettato deve favorire la transizione da un comportamento knowledge-based a uno rule/skill-based.

Stili di interfaccia

- **Simbolico:** comandi testuali con sintassi complessa, es. sistemi operativi a riga di comando.
- **Atomico:** tasti specializzati con effetti immediati e sintassi semplice, es. *app* menù.
- **Continuo:** dispositivi di input continuo come mouse e joystick, con feedback visivo rapido.

Ogni stile può emulare gli altri due, e un sistema realmente interattivo riesce a combinarli. Ogni stile comporta svantaggi, come carico computazionale e gestione degli errori.

Aspetti linguistici

- **Lessicale:** forme di comandi e di simboli di output.
- **Sintattico:** regole di combinazione dei comandi.
- **Semantico:** significato delle combinazioni di comandi.

Nell'elaborazione dell'input tutti questi livelli sono intrecciati, è importante progettare interfacce con una struttura coerente e comprensibile per l'utente.

Linee guida

Il loro scopo è aumentare la consapevolezza riguardo problematiche che non hanno una risposta generica. Un'indicazione standard è "provvedere feedback informativo", ma è troppo generica. Input e output devono essere compatibili, ma a volte non aiuta, es. l'input della data 97/06/10, è giugno o ottobre? Il sistema deve chiarire la sua interpretazione dei dati.

La **Legge di Fitt** descrive il tempo necessario per raggiungere un bersaglio con il mouse in base alla sua dimensione e distanza, questa legge aiuta a stabilire le dimensioni di oggetti distanti sullo schermo.

Attenzione alla **vaghezza** dei principi di design, che può portare a interpretazioni soggettive e difficoltà nell'applicare le linee guida in modo oggettivo: il **paradosso di Sorites** illustra come un sistema senza comandi non è "facile da usare", ma aggiungere comandi a un sistema già difficile non semplificherà il suo utilizzo, questo perché "facile da usare" è un concetto vago e non è determinato dal numero di comandi.

Il **Principio della minima sorpresa** suggerisce di progettare interfacce che si comportino in modo prevedibile, minimizzando le situazioni inaspettate per l'utente.

What is Good Design?

- 1- Deve essere **task-specific**, e il task deve risultare chiaro.
- 2- Deve essere **prevedibile**.
- 3- Deve essere **iterativo**, da rivedere continuamente in processo di creazione.
- 4- Deve essere **controllato** e non imporsi sull'utente.

GUEPs: Generative User Engineering Principles

Modello introdotto da Thimbery, comprende principi con le seguenti proprietà:

- Sono generici: si applicano a diversi tipi di sistemi.
- Possono essere dichiarati sia in maniera colloquiale che formale.
- Possono essere usati per vincolare e generare il design del sistema.

Modes - Modalità di interazione

Le modalità di interazione sono informazioni nascoste; generalmente ideate per rendere un sistema versatile, finiscono spesso a creare confusione.

Mode, views, *modalità*: variabile all'interno del sistema che influenza il significato della percezione e delle azioni dell'utente all'interno di una stessa applicazione.

Tecnicamente, anche i sistemi apparentemente non modali (modeless, low model) abbiano almeno una modalità (ad esempio, acceso/spento). Tipicamente, il sistema non informa l'utente del suo stato corrente o come c'è arrivato. Se esistono troppe modalità l'utente tende a perdere il conto, diventa difficile capire e ricordarsi in quale modalità si è.

Spatial Mode è l'interazione che avviene nello spazio fisico, ossia il modo con cui l'utente interagisce con l'interfaccia tramite elementi visivi, tattili o spaziali; **Temporal Mode** è l'interazione basata sul tempo, al ritmo con cui l'informazione viene presentata o la sequenza in cui l'utente esegue azioni. Le modalità nascoste generano problemi, come la difficoltà nel fornire informazioni chiare sullo stato del sistema.

Esempio della calcolatrice:

Lo schermo di una calcolatrice non visualizza altro che un numero: la mancanza di feedback chiaro su come si è arrivati a quel numero, o se il pulsante di calcolo premuto è stato registrato correttamente, crea confusione nell'utente, che deve essere in grado di ricordare le azioni precedenti per poter interpretare con successo il comportamento della calcolatrice. Un buon design dovrebbe mostrare tutte le parti del calcolo che sono rilevanti per l'utente.

231	+	4.1
running calculation	operator	current entry

WYSIWYGs: What You See Is What You Get

Nessuna informazione nascosta: è un esempio di GUEP che promuove la trasparenza e la prevedibilità dell'interfaccia. Nel l'applicazione del WYSIWYG ai sistemi di editing testo l'utente ha la possibilità di concentrarsi sul contenuto senza dover imparare comandi complessi di formattazione. Il modello è ottimo per applicazioni semplici, ma limita la libertà in quelle complesse, gli aspetti che sono positivi per un nuovo utente possono risultare tediosi per un esperto. Non sono fornite informazioni sullo storico, ma si è aiutati da funzioni come *undo* e *redo*.

I menù solitamente implementano WW, ma se ci sono più opzioni possibili di quelle visualizzate allora WW non è più da considerarsi applicato: se inseriamo un valore oltre il range indicato, per esempio, e l'azione produce un risultato valido, non è WW.

HCI Compendium

Le interfacce hanno importanza fondamentale nell'utilizzo dei computer, senza le quali sarebbero inutilizzabili. Sorprendentemente, però, il design delle interfacce uomo-macchina non riceve la stessa attenzione e cura dedicata alle interfacce tra computer. La progettazione di interfacce utente richiede un approccio creativo e coinvolge due aspetti fondamentali: quello **informatico**, programmazione di algoritmi per facilitare la comunicazione e il controllo da parte dell'utente; e quello **umano**, comprensione e supporto delle abilità dell'utente.

L'utilizzo di **approcci formali** può essere utile per astrarsi dai dettagli di un'applicazione, specifici a essa, e identificare caratteristiche generali dell'interazione, come l'annullamento di un'azione (*undo*) che può essere applicato in diversi contesti. L'analisi di applicazioni già esistenti può e deve fornire spunti per lo sviluppo di nuove interfacce: emergono **regole generali** del buon design, come il basso carico cognitivo, l'uso di icone intuitive, la possibilità di annullare le azioni e la disponibilità di assistenza online.

Tuttavia, **l'eccessiva formalizzazione** può portare all'alienazione dell'utente, trascurando le sue intenzioni, aspettative e abilità. Un aspetto cruciale del design è la documentazione, che deve descrivere accuratamente l'implementazione del sistema e assicurare la coerenza tra progettazione, implementazione e documentazione stessa.

Progettazione collaborativa

La progettazione di interfacce utente richiede la cooperazione tra **informatici** e **psicologi** per comprendere sia il funzionamento dei computer sia il comportamento umano. Esistono due approcci principali alla progettazione: nel **design formale** gli informatici cercano di creare un'interfaccia ottimale fin dalla prima iterazione, basandosi su regole formali; nel **design iterativo** gli psicologi preferiscono un approccio reiterativo, in cui il sistema viene migliorato gradualmente attraverso il feedback degli utenti, anche coinvolgendo l'utente stesso nella progettazione.

Attenzione alle **composizioni fallaci**, ovvero l'errata convinzione che un'interfaccia composta da buone parti sia necessariamente una buona interfaccia, questo approccio "bottom-up" può portare a funzionalità scollegate ed effetti collaterali imprevisti.

Difficoltà e comportamenti dell'utente nell'interazione

Esistono due tipi di "colli di bottiglia" nell'interazione e riguardano:

- **Trasferimento di informazioni (I/O)**: limitazioni nella velocità e nell'efficienza della comunicazione tra utente e computer.
- **Elaborazione delle informazioni (mentale)**: limitazioni nella capacità dell'utente di elaborare le informazioni presentate dall'interfaccia.

L'utente riscontra inoltre:

- Difficoltà nel mantenere convinzioni nel tempo.
- Mancanza di una notazione adeguata a descrivere i propri modelli mentali.
- Preferenza per le scelte transitive (se A è meglio di B e B è meglio di C, allora A è meglio di C).

La memoria a breve termine (**STM**) ha una capacità limitata (*span*, circa 7 oggetti) e un decadimento rapido (20 secondi), anche se l'informazione può essere mantenuta attiva attraverso la ripetizione. Le risorse coinvolte sono quindi limitate. La memoria a lungo termine (**LTM**) ha una capacità molto maggiore e un decadimento molto lento: la macchina può cancellare dati per sempre, ma l'uomo non dimentica mai veramente.

Priming: fenomeno psicologico in cui l'esposizione a un concetto influenza la risposta a stimoli successivi (es. se esposto al colore giallo, se ti viene chiesto di nominare dei frutti sarai più inclinato a pensare a frutti di colore giallo).

Comportamento stereotipato: quando gli utenti, in situazioni di panico, ricorrono a soluzioni di fuga predefinite che possono non essere le più efficaci.

Dissonanza cognitiva: sensazione di disagio che si prova quando si hanno impressioni contrastanti (es. dopo aver scelto un'opzione, l'utente potrebbe iniziare a percepire l'alternativa scartata come migliore, per ridurre la dissonanza si cerca di tornare indietro, minimizzare l'importanza della scelta o autoconvincersi della decisione presa).

Gli utenti tendono ad adattarsi ai sistemi difettosi o inefficienti piuttosto che affrontare il costo di cambiarli.

Metodi di Innovazione e Frameworks

Novità, miglioramento e diffusione: l'innovazione introduce qualcosa di originale e più efficace nel mercato o nella società. È importante distinguere tra innovazione e invenzione: l'innovazione si concentra sull'implementazione pratica dell'invenzione per avere un impatto significativo.

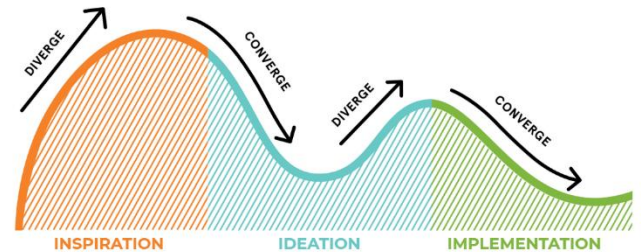
Innovazione di Sostegno

Lento e incrementale miglioramento di un prodotto già esistente: basso rischio, in un mercato presente e consolidato.

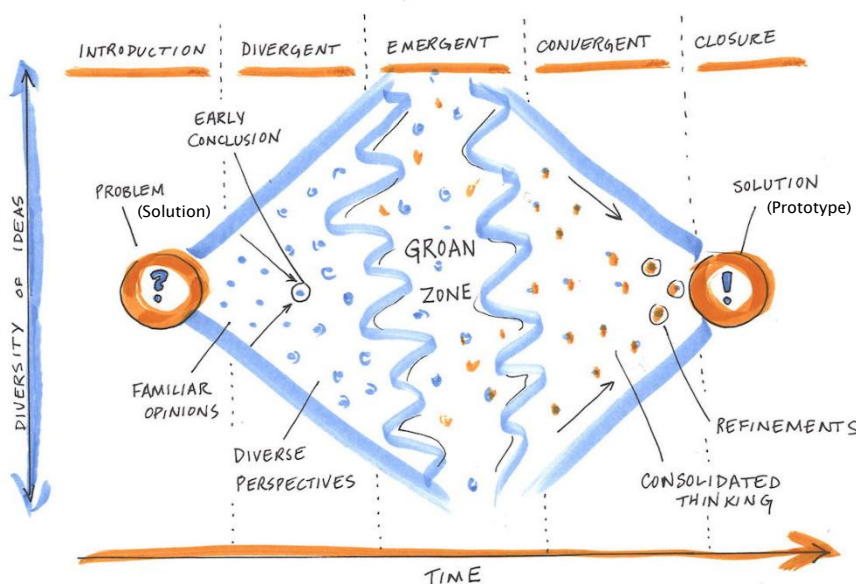
Innovazione Dirompente

Invenzione o miglioramento di un prodotto che porta alla creazione di un nuovo mercato o un drastico cambiamento del mercato attuale.

Per essere innovativo, un p/s deve essere utilizzabile, l'HCD è quindi fondamentale: l'innovazione dirompente deve essere incentrata sull'utente. **IDEO**, una delle aziende di design più innovative al mondo, enfatizza l'empatia per l'utente finale: il loro processo di HCD si basa sull'osservazione del comportamento degli utenti e sulla messa in pratica del punto di vista dell'utente.



SAM KANER'S DIAMOND MODEL OF PARTICIPATION



Ispirazione:

Comprendere i bisogni e le sfide degli utenti, abbandonando preconetti.

Ideazione:

Visualizzare, ridefinire, fare brainstorming e discutere potenziali soluzioni. Reiterare una volta ottenuto feedback.

Implementazione:

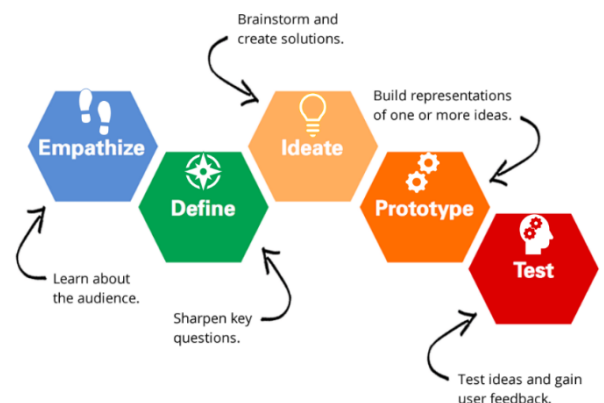
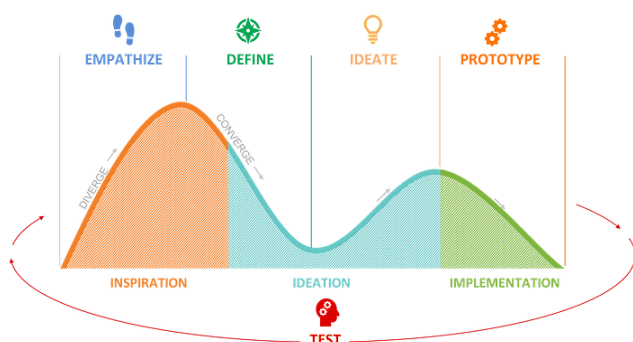
Creare un prototipo ad alta fedeltà, testarlo e preparare il prodotto per l'uso nel mondo reale

Design Thinking

Processo iterativo dove si comprende l'utente, i suoi obiettivi e difficoltà, per ridefinire problemi nel tentativo di trovare nuove strategie alternative e soluzioni che all'inizio non sarebbero venute in mente. Le 5 fasi sono:

- **Empatizzare:** studiare gli utenti.
- **Definire:** i bisogni, problemi, intuizioni degli utenti.
- **Ideare:** generare idee per soluzioni innovative.
- **Prototipare:** rappresentare l'idea.
- **Testare:** presentare l'idea e raccogliere feedback.

HCD + Design Thinking = **Social Enterprise Thinking**



HCD è una **linea di pensiero**. HCD Process e Design Thinking sono **metodi** di progettazione.

Il Design Thinking può essere utilizzato da qualsiasi azienda per creare una soluzione redditizia. L'HCD, applicato al Design Thinking, garantisce che la stessa soluzione soddisfi effettivamente i bisogni degli utenti.

Metodi di Sviluppo per Design Innovativi

Waterfall

Approccio di project management lineare in cui i requisiti vengono raccolti all'inizio e il progetto procede in fasi sequenziali che si susseguono una con l'altra (termine di un task → inizio del prossimo).

Metodo strutturato che persiste da tempo perché funzionale.

Agile

Approccio iterativo e flessibile che si concentra su come comprendere l'ambiente attuale, identificare le incertezze, e capire come adattarsi.

Pensato per team piccoli, ci si focalizza sui dipendenti e come lavorano assieme.

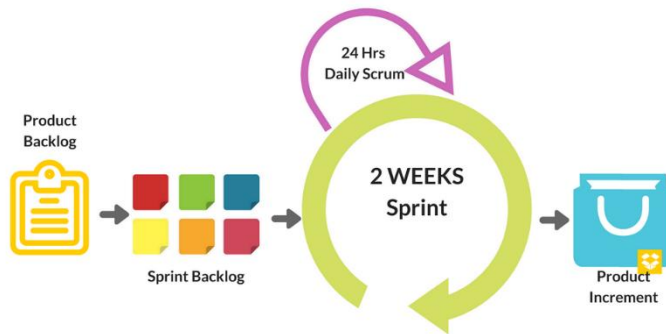
Termine ombrello rappresentato da un manifesto e 12 principi.

Scrum

Framework Agile per lo sviluppo di prodotti complessi, si spezzetta il lavoro in obiettivi da completare in un determinato limite di tempo, detti *sprint*, di durata totale non oltre un mese.

I progressi sono controllati in piccole riunioni di 15 minuti chiamate *Scrum*. La chiave principale è reagire velocemente ai cambiamenti nel contesto.

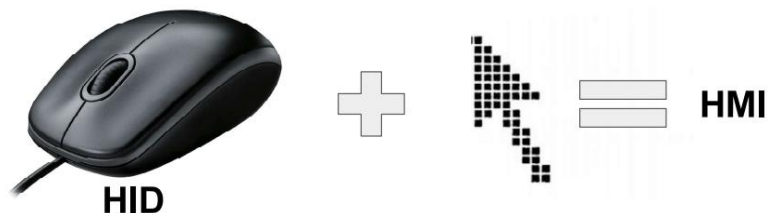
- **Proprietario del prodotto:** clienti o stakeholder, definisce il prodotto in termini *customer-centric*.
- **Team di sviluppo**
- **Scrum Master:** facilitatore del gruppo, garantisce che lo scrum venga compreso e applicato correttamente. Non ha autorità, è un motivatore.



Le Interfacce Utente

Punto di contatto tra due sistemi che cercano di comunicare, in ambito tecnologico l'interfaccia utente (UI) è lo spazio dove avviene l'interazione uomo-macchina. L'obiettivo principale delle UI è consentire all'utente di controllare la macchina in modo efficace: deve essere facile, divertente ed efficiente per massimizzare la user experience. Lo sforzo fisico e cognitivo dell'utente deve essere minimo. La progettazione di una UI è **interdisciplinare**, coinvolge psicologia, neuroscienze, design e fisica.

Le **HMI** (*Human-Machine Interface*) sono interfacce che collegano macchine con hardware di input (tastiere, mouse) e output (monitor, stampanti), un dispositivo che implementa una HMI è un **HID** (*Human Interface Device*).



Le UI sono generalmente classificate in base ai sensi umani utilizzati per l'interazione:

1) Tactile UI (*Tatto*)
Visual UI (*Vista*)

Auditory UI (*Udito*)
Olfactory UI (*Olfatto*)

Gustatory UI (*Gusto*)
Equilibrial UI (*Equilibrio*)

CUI (*Composite User Interface*):
interfacce che utilizzano più di un senso.

→ **GUI** (*Graphical User Interface*):
CUI comuni composte da interfacce visive e tattili.

→ **MUI** (*Multimedia User Interface*):
GUI con l'aggiunta del suono.

CATEGORIE

- **Standard:** utilizzano dispositivi come tastiere, mouse e monitor.
- **Virtuale:** bloccano il mondo reale per creare una realtà virtuale.
- **Aumentata:** non bloccano il mondo reale ma creano una realtà aumentata

CLASSIFICAZIONE

- **3S:** coinvolgono tre sensi, es. Smell-O-Vision: vista, udito, olfatto.
- **4S:** coinvolgono quattro sensi, es. cinema 4D con poltrone mobili.
- **Qualia Interface:** interagiscono con tutti i sensi umani.

Es. Microsoft Reactable è una MUI 3S Aumentata

NOTA: L'aggiunta di più canali sensoriali non sempre migliora l'interfaccia!

Human Interface Devices (HID)

Gli HID ricevono input dagli esseri umani, e forniscono output agli esseri umani. Il termine "HID" si riferisce sia ai dispositivi fisici che alla specifica **USB-HID**. Il termine è stato coniato da Mike Van Flandern di Microsoft, che propose al comitato USB di creare un gruppo di lavoro sulla classe Human Input Device. Il gruppo di lavoro è stato poi rinominato Human Interface Device, poiché lo standard proposto supportava la comunicazione bidirezionale.

Specifica Software HID:

Lo standard è stato adottato principalmente per favorire l'innovazione nei dispositivi di input per PC. Prima dell'introduzione, i dispositivi si conformavano a protocolli rigidi definiti per mouse, tastiere e joystick. Ad esempio, il protocollo standard per il mouse all'epoca supportava solo dati relativi agli assi X e Y e input binari per un massimo di due pulsanti. Qualsiasi innovazione hardware richiedeva il sovraccarico dell'uso dei dati, in un protocollo già esistente, oppure la creazione di driver appositi con nuovi protocolli.

Al contrario, tutti i dispositivi HID forniscono pacchetti autodescrittivi che sostengono un numero qualsiasi di dati e formati. Un singolo driver HID consente l'associazione dinamica dell'I/O con le funzionalità dell'applicazione, il che ha consentito una rapida innovazione e sviluppo e una prolifica diversificazione di nuovi dispositivi di interfaccia umana.

Il protocollo HID ha i suoi limiti, ma tutti i SO moderni riconoscono i dispositivi USB-HID standard senza la necessità di driver specializzati. Quando si installa un nuovo dispositivo, in genere appare sullo schermo un messaggio che dice "*È stato riconosciuto un dispositivo conforme allo standard HID*". Questo messaggio non compare per i dispositivi collegati tramite connettori che precedevano l'USB.

Protocollo HID

Nel protocollo HID sono presenti due entità: l'**host** e il **dispositivo**. Il dispositivo è l'entità che interagisce direttamente con l'essere umano (tastiera, mouse). L'host comunica con il dispositivo e riceve i dati di input in base alle azioni eseguite dall'utente. I dati di output fluiscono dall'host al dispositivo e quindi all'utente.

Il protocollo HID rende l'implementazione dei dispositivi molto semplice: i dispositivi presentano un "descrittore HID" all'host. Il descrittore HID è un array di byte codificato in modo fisso che descrive i pacchetti di dati del dispositivo (quanti pacchetti quanti supporta il dispositivo, la loro dimensione, lo scopo di ogni byte e bit). Ci si aspetta che l'host sia un'entità più complessa del dispositivo: deve recuperare il descrittore HID e analizzarlo prima di poter comunicare pienamente.

L'analisi del descrittore può essere complicata, alcuni SO hanno presentato bug nei driver responsabili dell'analisi anni dopo il rilascio iniziale al pubblico, ma è questa complessità il motivo per cui è possibile una rapida innovazione. Poiché si è capito che non tutti gli host sarebbero stati in grado di analizzare i descrittori HID, HID definisce anche il **boot protocol**. Gli unici dispositivi supportati nel protocollo di avvio sono tastiera e mouse.

Dalla sua definizione originale su USB, HID è ora utilizzato anche in altri bus di comunicazione come Bluetooth HID, Serial HID, ZigBee, etc.

I dispositivi periferici possono essere di:

- **INPUT:** basati su sensori il cui scopo è rilevare eventi o cambiamenti nel mondo fisico e convertirli in segnali analogici o digitali.
- **OUTPUT:** basati su attuatori il cui scopo è convertire segnali elettronici analogici o digitali in eventi fisici volti a modificare il mondo fisico.

Dispositivi di INPUT

- **Tastiere:** L'HID di testo e caratteri più utilizzato. Sono disponibili diversi tipi e ognuna è progettata con un'attenzione particolare a specifiche funzionalità per rispettive esigenze. Oggi, la maggior parte delle tastiere full-size utilizza uno dei tre diversi layout meccanici, generalmente denominati semplicemente ISO (ISO/IEC 9995-2) standard ANSI. Le tastiere standard ANSI hanno tasti che si trovano su centri di tre quarti di pollice (19 mm) e hanno una corsa dei tasti di almeno 3,8 mm. I moderni modelli di tastiera contengono un numero fisso di tasti totali in base al loro standard, descritto come 101, 104, 105, etc.

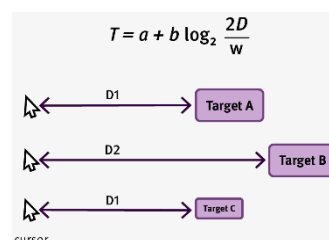
- **Layout di tastiera:** Un layout è la disposizione
 - **fisica** (posizionamento effettivo dei tasti)
 - **visiva** (disposizione delle etichette, segni, incisioni che compaiono sui tasti)
 - **funzionale** (mappatura della tastiera, determinata nel software)

Quelle moderne sono progettate per inviare un codice al SO quando un tasto viene premuto o rilasciato: questo codice riporta solo la riga e la colonna del tasto, non il carattere specifico inciso. Il SO converte il codice nel corrispettivo carattere binario utilizzando una tabella di conversione: ciò significa che una tastiera fisica può essere mappata a qualsiasi layout senza cambiare i componenti hardware.

- **Layout QWERTY:** È il layout più diffuso, non confinato a una particolare area geografica e progettato per alfabeti latini. Il nome deriva dall'ordine delle prime sei lettere in alto a sinistra della tastiera.
- **Tastiere multifunzione:** Forniscono funzioni aggiuntive oltre alla tastiera standard, sono programmabili e configurabili e alcune controllano più PC, workstation etc. solitamente in ambienti di lavoro multischermo. Gli utenti hanno funzioni dei tasti aggiuntive e possono essere dotate di tastierini numerici personalizzati, funzioni programmabili o tasti funzione per macro/preset, lettori biometrici o di smart card, trackball, etc. Quelle di nuova generazione hanno display touchscreen per lo streaming video, controllo dei media audiovisivi, etc. Comuni in luoghi di lavoro complessi e ad alte prestazioni (operatori finanziari, operatori di sale di controllo, servizi di emergenza, sicurezza, gestione del traffico aereo, ...).
- **Lettori di codici a barre:** È uno scanner ottico in grado di leggere codici a barre stampati, decodificare i dati contenuti e inviare i dati a un computer. È costituito da una sorgente luminosa, una lente e un sensore di luce che traduce gli impulsi ottici in segnali elettrici. Un codice a barre è una rappresentazione di dati (stringa codificata, tipicamente numeri) leggibile dalla macchina, quelli standard presentano i dati variando le larghezze e le spaziature delle linee parallele. Un QR è un codice a barre bidimensionale, utilizza quattro modalità di codifica standardizzate (numerica, alfanumerica, byte/binaria e kanji).
- **RFID:** Un tag RFID è costituito da un minuscolo transponder radio, quando attivato dall'impulso elettromagnetico del dispositivo lettore RFID, il tag trasmette dati digitali. I tag **passivi** sono alimentati dall'energia delle onde radio del lettore RFID, i tag **attivi** sono alimentati da una batteria e quindi possono essere letti a una distanza maggiore, fino a centinaia di metri. Non ha bisogno di essere nel campo visivo del lettore: può essere incorporato nell'oggetto tracciato.
- **NFC (Near-Field-Communication):** Un insieme di protocolli di comunicazione bidirezionale tra due dispositivi elettronici a distanza max 4cm. Offre una connessione a bassa velocità con una configurazione semplice, può essere utilizzata per avviare connessioni wireless più potenti.

- **Dispositivi di puntamento:** Consente all'utente di inserire dati spaziali nel computer. I sistemi CAD e le interfacce utente grafiche (GUI) consentono all'utente di controllare e fornire dati al computer utilizzando gesti fisici, muovendo un mouse o un dispositivo simile, usando i pulsanti sul mouse. I movimenti del dispositivo sono riprodotti sullo schermo dal puntatore.

Vedi Legge di Fitt → MT: Movement Time | a: Tempo di partenza e fermata del puntatore | b: Velocità del puntatore | D: Distanza dal puntatore all'obiettivo | W: Larghezza dell'obiettivo.



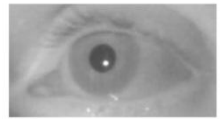
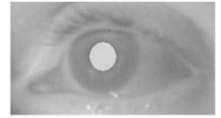
Diretti: il puntatore si trova fisicamente nella stessa posizione del dispositivo (es. dito su touchscreen, stilo su un tablet).		Indiretti: il puntatore traduce il movimento del dispositivo (es. mouse, joystick, stilo su tavoletta grafica).	
Assoluti: fornisce una mappatura costante tra un punto nello spazio di input (posizione/stato) e un punto nello spazio di output (posizione del puntatore sullo schermo).		Relativi: mappa lo spostamento dell'input con lo spostamento dell'output. Il dispositivo si può quindi trovare in un'altra posizione rispetto al puntatore fisico.	
Isotonici: mobile, misura lo spostamento (mouse, penna, braccio).	Elastici: fisso, aumenta la resistenza con lo spostamento (joystick).	Isometrici: fisso, misura la forza che agisce su di esso (trackpoint).	
Di posizione: mobile, modifica direttamente la posizione assoluta o relativa del puntatore sullo schermo		Di velocità (rate): fisso, modifica la velocità e la direzione del movimento del puntatore sullo schermo.	

Dispositivi di puntamento innovativi:

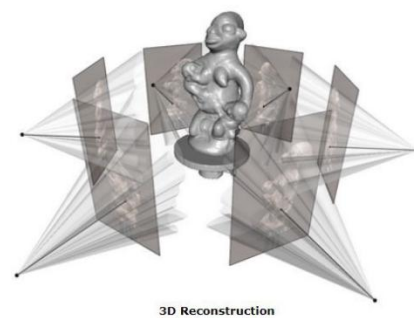
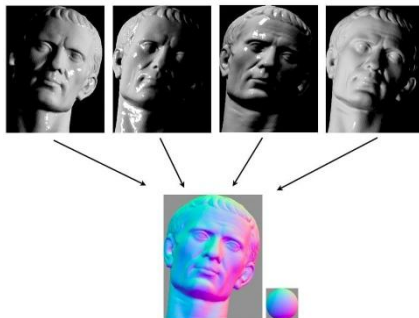
- **Eye Tracker:** l'eye tracking è la misurazione del punto di sguardo o del movimento dell'occhio rispetto alla testa. Permette di misurare la posizione e movimento dell'occhio. La luce infrarossa viene riflessa dall'occhio e rilevata da una videocamera o da qualche altro sensore ottico, le informazioni

vengono quindi analizzate e si estrare la rotazione dell'occhio dai riflessi. Una fotocamera si concentra su uno o entrambi gli occhi e registra il movimento degli occhi. Vengono utilizzate due tecniche di eye tracking a infrarossi (luce attiva):

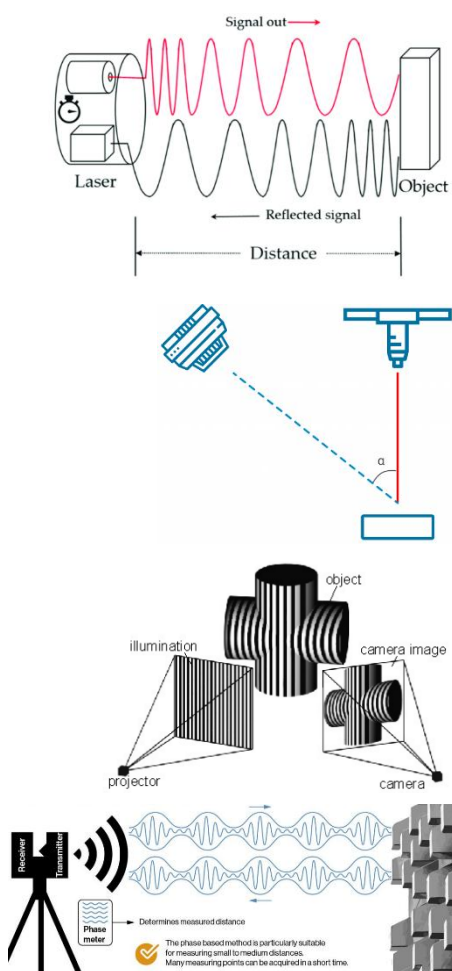
- **Pupilla Lucida:** si crea un contrasto tra la pupilla e l'iride, permette una maggiore precisione e l'eliminazione di interferenze (ciglia), anche in caso di ambienti con scarsa luce. Molto preciso ma richiede un hardware complesso.
- **Pupilla Scura:** non si crea alcun contrasto. Più semplice ed economico ma meno preciso e si possono avere interferenze dovute a ciglia o altri oggetti nell'obiettivo.



- **Dataglove:** invia input tramite un guanto cablato indossabile. Utilizza sensori di movimento, come i magnetometri o IMU per misurare la posizione e rotazione della mano. I gesti possono essere classificati in informazioni, ad es. per riconoscere il linguaggio dei segni o altre funzioni simboliche.
- **Dispositivi Aptici:** dispositivi meccanici che permettono all'utente di toccare e manipolare oggetti 3D virtuali. Restituiscono feedback sottoforma di sensazioni tattili coordinate con gli output.
- **Smart Paper, lavagne e simili:** volti a digitalizzare l'interazione dell'utente con carta e lavagna. La scrittura dell'utente viene digitalizzata tramite smart pens tracciate e/o superfici sensorizzate. Il puntamento e le immagini vengono utilizzati come input combinati.
- **Dispositivi audio:** dispositivi che riescono a catturare e/o emettere onde sonore. Oltre i 20 kHz sono chiamate ultrasuoni e non sono percettibili dall'uomo, sotto i 20Hz sono chiamati infrasuoni. Ricontrano alcune limitazioni come: bandwidth molto inferiore rispetto ai display visivi, natura effimera del parlato (tono, ecc.), complessità di analisi (carico computazionale più elevato).
 - **Microphone Array:** un qualsiasi numero di microfoni che operano in tandem, collegati a un computer che registra e interpreta i risultati. Sono usati per estrarre la voce da ambienti rumorosi, *surround sound* e tecnologie simili, acoustic source localization, registrazioni di alta qualità, etc.
- **Dispositivi a immagini:** convertono le informazioni per creare immagini, catturano le onde luminose e le trasformano in segnali. Due principali tipi di sensori di immagine elettronici sono il dispositivo ad accoppiamento di carica (CCD) e il sensore a pixel attivi (sensore CMOS). Entrambi sono basati sulla tecnologia metallo-ossidosemiconduttore, con i CCD basati su condensatori MOS e i CMOS basati su amplificatori MOSFET.
- **Dispositivi di acquisizione 3D:** effettuano l'analisi di un oggetto o di un ambiente del mondo reale e raccolgono dati sulla sua forma ed aspetto, da cui costruire modelli 3D digitali. Ampiamente utilizzati nell'intrattenimento, inclusa la realtà virtuale, realtà aumentata, motion capture, riconoscimento dei gesti, mappatura robotica, design industriale, ortesi e protesi, reverse engineering e prototipazione, controllo qualità/ispezione e digitalizzazione di manufatti culturali. Uno scanner 3D può essere basato su molte tecnologie diverse, ognuna con i suoi limiti, vantaggi e costi. Sono ancora presenti molte limitazioni nel tipo di oggetti che possono essere digitalizzati, per es. la tecnologia ottica può incontrare molte difficoltà con oggetti lucidi, riflettenti o trasparenti. Le tecnologie di scansione 3D possono essere suddivise in due categorie principali:
 - **Passiva:** non emettono alcun tipo di onda luminosa, ma si basano invece sul rilevamento della luce ambientale riflessa; possono essere utilizzati anche altre emissioni come l'infrarosso. I metodi passivi riescono a essere molto economici, perché spesso non necessitano di hardware particolare ma di semplici fotocamere digitali.
 - **Sistemi stereoscopici:** utilizzano due videocamere, leggermente distanti, che guardano la stessa scena. Analizzando le lievi differenze tra le immagini viste da ciascuna telecamera, è possibile determinare la distanza in ogni punto delle immagini.
 - **Sistemi fotometrici:** utilizzano una singola telecamera, ma acquisiscono più immagini in condizioni di illuminazione variabili. Questa tecnica inverte il modello di formazione dell'immagine per recuperare l'orientamento della superficie in ogni pixel.
 - **Tecniche di Silhouette:** utilizzano outlines generati da una sequenza a 360° di fotografie di un oggetto su uno sfondo ben contrastato. Le sagome vengono estruse e intersecate per andare a creare l'oggetto approssimato. Con questo approccio alcune concavità dell'oggetto (come l'interno di una ciotola) non possono essere rilevate.



- **Attiva:** emettono luce e ne rilevano il riflesso o che attraversa il bersaglio per sondare un oggetto o un ambiente. I possibili tipi di emissioni utilizzate includono luce, ultrasuoni o raggi X.



- **Time-of-Flight:** scanner attivo che utilizza la luce laser per sondare il soggetto. Il cuore di questo scanner è un telemetro laser: esso trova la distanza di una superficie cronometrando il tempo di andata e ritorno di un impulso di luce (*tempo di volo*). Il laser emette un impulso di luce e si misura la quantità di tempo prima che il riflesso venga individuato dal rivelatore. (~3,3 picosecondi = 1 millimetro). Il telemetro laser rileva solo la distanza di un punto; pertanto, viene scansionato l'intero campo visivo un punto alla volta.

- **Triangolazione:** scanner attivi che utilizzano la luce laser per sondare l'ambiente. Rispetto allo scanner time-of-flight, il laser di triangolazione irraggia il soggetto e ricerca la posizione del punto laser con una telecamera. A seconda della distanza a cui il laser colpisce una superficie, il punto laser appare in punti diversi nel campo visivo della telecamera. Si chiama così perché il punto laser, la telecamera e l'emettitore laser formano un triangolo.

- **a Luce Strutturata:** un pattern luminoso viene proiettato sul soggetto e una telecamera leggermente sfalsata rispetto al proiettore osserva la deformazione del pattern sulla superficie colpita e calcola la distanza di ogni punto nel campo visivo. Il vantaggio è la velocità e la precisione: invece di scansionare un punto alla volta, scansionano più punti contemporaneamente o l'intero campo visivo, eliminando il problema di distorsione causato da movimenti accidentali.

- **a Luce Modulata:** il soggetto viene illuminato con una luce che cambia continuamente: la sorgente luminosa varia in ciclo la sua ampiezza (sinusoide). Una telecamera rileva la luce riflessa e le variazioni del pattern, determinando la distanza percorsa dalla luce. Permette di concentrarsi solo sul suo raggio perciò non ci sono interferenze.

Esempi di interfaccia innovativa brida:

Il **Microsoft Kinect** rappresenta un'innovazione significativa: si tratta di un dispositivo di input basato sul rilevamento del movimento che utilizza una combinazione di tecnologie

- **Fotocamere RGB:** catturano immagini a colori dell'ambiente circostante.
- **Proiettori infrarossi:** proiettano una griglia di punti infrarossi invisibili all'occhio umano.
- **Rilevatori infrarossi:** rilevano la distorsione della griglia di punti proiettata per calcolare la profondità.
- **Array di microfoni:** catturano l'audio dell'ambiente.

Queste tecnologie, unite a software e algoritmi di AI, permettono al Kinect di eseguire in tempo reale:

- **Riconoscimento dei gesti:** interpreta i movimenti del corpo umano come comandi.
- **Riconoscimento vocale:** consente agli utenti di interagire con il sistema tramite comandi vocali.
- **Rilevamento scheletrico del corpo:** identifica e traccia la posizione delle articolazioni del corpo umano, permettendo al sistema di comprendere la postura e i movimenti degli utenti.

Kinect funge da interfaccia naturale a mani libere, consentendo un'interazione più ricca e intuitiva.

Un altro esempio di dispositivo motion sensing, meno moderno e complesso, è il **Wii Remote** della Nintendo.

Quest'ultimo utilizza un'**unità di misura inerziale (IMU)**: dispositivo elettronico che misura l'accelerazione, angolo di rotazione e orientamento di un corpo, utilizzando accelerometro, giroscopio e magnetometro. Dispositivo a 9 assi (3 per ogni sensore).

Wearables - dispositivi indossabili

Dispositivo computerizzato che può essere indossato dall'utente. L'interfaccia e l'unità computazionale sono mescolati assieme. Tipicamente questi dispositivi sono utilizzati per la vita quotidiana o nel campo del fitness. Si indossano negli arti, collo o in testa. Essendo molto piccoli, hanno spesso problemi dovuti alla batteria o alla dissipazione del calore.

- **Monitor del battito cardiaco**: dispositivo che riesce a rilevare il battito cardiaco.
 - **ECG**: dispositivo usato nell'ambito medico che cattura i segnali elettrici del corpo che permettono di contrarre ed espandere il cuore.
 - **PPG**: dispositivo basato su sensori luminosi che riesce a misurare il volume dei vasi sanguigni e a calcolare il battito cardiaco. Inoltre, è anche in grado di calcolare l'ossigeno nel sangue.
- **EEG**: dispositivo che monitora gli impulsi elettrici del cervello. Si può usare come dispositivo di input per persone affette da disabilità.

Information Architecture

L'obiettivo dell'IA è organizzare, strutturare ed etichettare i contenuti in modo efficace e sostenibile, per aiutare gli utenti a trovare le informazioni e completare gli obiettivi. Abbiamo 4 strutture principali:

Gerarchica

Struttura "top-down" in cui le informazioni sono organizzate in categorie generali (*parent*) che si suddividono in categorie più specifiche (*child*). Questo tipo di struttura è familiare a molti utenti.

Sequenziale

Gli utenti seguono un percorso specifico passo dopo passo, ad esempio per completare un acquisto o un corso online.

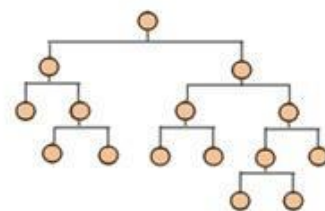
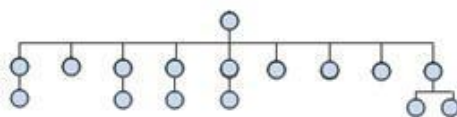
Matrice

Gli utenti possono scegliere il proprio percorso di navigazione, poiché i contenuti sono collegati in vari modi.

Modello di database

Approccio "bottom-up" che utilizza i metadati per collegare i contenuti e consentire funzionalità di ricerca e filtro avanzate, permette un'esperienza dinamica.

È importante progettare strutture che abbiano spazio per adattarsi alla crescita futura e che non siano né troppo superficiali (con menu enormi) né troppo profonde (con informazioni sepolte sotto troppi livelli).



Il compito principale dell'architettura dell'informazioni è far capire all'utente in che **stato si trovano**, che **informazioni hanno già trovato**, che **informazioni ci sono** in quel determinato stato, cosa **aspettarsi** in quello stato. Le componenti dell'architettura sono:

Schemi organizzativi e strutturali: come categorizzare e strutturare le informazioni.

Sistemi di etichettatura: come le informazioni sono rappresentate.

Sistemi di navigazione: come l'utente naviga tra le informazioni.

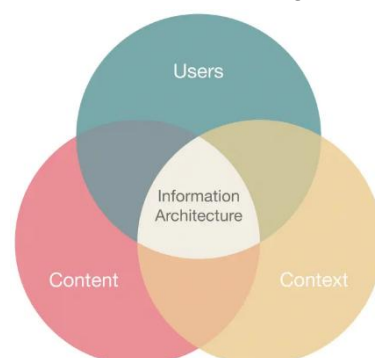
Sistemi di ricerca: come l'utente cerca le informazioni.

Context: obiettivi finanziari, politiche, cultura, tecnologie, risorse e vincoli.

Content: content objectives, docs e data types, strutture, proprietà e gestione.

Users: target, compiti, necessità, attitudine alla ricerca di informazioni, esperienze e impresioni.

Information ecology



• SCHEMI ORGANIZZATIVI

Categorizzazione del contenuto e come le parti si relazionano tra loro. Gli schemi possono essere:

- **Esatti:** le informazioni sono divise in sezioni mutualmente esclusive secondo un ordine preciso. Tipicamente si parla di ordine:
 - Alfabetico *es.* rubrica
 - Cronologico *es.* galleria
 - Geografico *es.* rent bike maps
- **Soggettivi:** le informazioni sono categorizzate in schemi che riflettono i punti di vista, le percezioni o preferenze dell'utente. Difficili da progettare, ma spesso più efficaci. Tipicamente suddivisi in:
 - Topic: contenuti suddivisi in topic che hanno significato per gli utenti *es.* Netflix
 - Task: contenuti suddivisi in base alle necessità, azioni e domande *es.* Google Drive
 - Audience: contenuti organizzati per determinati segmenti di pubblico *es.* account Netflix
 - Metaforici: contenuti divisi e correlati a concetti familiari all'utente *es.* File Explorer

A seconda del contenuto è ammissibile che dei siti combinino più tipi di schemi, ma anche implementarne uno solo ha i suoi vantaggi perché mantiene il tutto più semplice e intuitivo per l'utente.

• SCHEMI DI NAVIGAZIONE

Riguardano l'orientamento dell'utente nell'interfaccia, e il suo scopo è permettere di cercare le informazioni o funzioni desiderate. I principi base di una buona navigazione sono Findability (cosa si aspetta di trovare) e la Discoverability (cosa può trovare).

User Behavior Patterns

Modelli ripetuti di comportamento che gli utenti manifestano quando interagiscono con un prodotto:

Esplorazione autonoma: gli utenti desiderano esplorare l'interfaccia senza perdersi o incontrare problemi.	Gratificazione immediata: gli utenti desiderano vedere risultati immediati dalle loro azioni, e provano gratificazione da un'esperienza riuscita.	"Satisficing": gli utenti si accontentano di soluzioni "abbastanza buone" se le alternative migliori richiedono troppo tempo o sforzo.
Cambiamenti a metà percorso: gli utenti possono cambiare idea su ciò che stanno facendo e desiderano poter cambiare e tornare indietro; possibilità di lasciare un'azione a metà e potervi tornare dopo.	Assuefazione: gli utenti si abituano ad azioni abituali, ma questo può creare problemi se l'azione non funziona come pensato. "Funziona sempre, perché non qui?"	Costruzione incrementale: gli utenti costruiscono le cose gradualmente, apportando modifiche e testando lungo il percorso, è necessario quindi visualizzare il lavoro dopo ogni modifica.
Scelte differite: gli utenti possono voler rimandare alcune decisioni a un secondo momento. "You can always change this later by..."	Microbreaks: gli utenti fanno piccole pause tra le attività e l'interfaccia deve supportare l'interruzione e la ripresa.	Memoria spaziale: gli utenti ricordano la posizione degli oggetti, e non come si chiamano.
Memoria prospettica: gli utenti devono avere la possibilità di creare promemoria per se stessi.	Ripetizione semplificata: le attività ripetitive vanno semplificate.	Keyboard only: fornire shortcut da tastiera per una maggiore efficienza.

Consigli degli altri: gli utenti sono influenzati dai consigli dei loro pari, direttamente o indirettamente. Una community online di supporto è una componente importante.

Start → Data & content → Functionality, Inputs & Outputs → Presentation Layer (UI)

Componenti dell'interfaccia includono (ma non si limitano):

- Comp. di **input**: bottoni, liste, campi di testo o di dati.
- Comp. di **navigazione**: campi di ricerca, paginazione, icone, slide.
- Comp. di **informazione**: indizi, barre di progresso, notifiche, box di messaggi.
- **Containers**: "fisarmoniche", component a scomparsa.

Usability and Evaluation Techniques

Il testing è un passaggio fondamentale dei prototipi, volto a creare prodotti che soddisfino le esigenze e aspettative degli utenti, seguendo i principi del design centrato sull'uomo. Si possono scoprire certi problemi solo con l'uso del prodotto, individuare possibili migliorie e scoprire come l'utente si comporta col prodotto. L'"**usabilità**" è un attributo di qualità che valuta la facilità d'uso delle interfacce utente. Comprende:

- **Learnability**: quanto è semplice per un utente affrontare i task base per la prima volta?
- **Efficienza**: dopo che hanno imparato il design, quanto velocemente svolgono i task?
- **Memorability**: Dopo che l'utente non usa il design per un determinato periodo, in quanto tempo riesce a riprendere confidenza?
- **Errori**: quanti errori effettua l'utente, quanto gravi sono, è facilmente possibile tornare indietro?
- **Satisfaction**: quanto è piacevole usare il design?

Oltre all'usabilità, l'**utilità** è un altro concetto chiave, che si riferisce alla funzionalità del design: fa quello di cui gli utenti hanno bisogno? **Utility + Usability = Useful**.

Le 10 Euristiche di Nielsen

La **valutazione euristica** è un metodo di valutazione delle interfacce basato su dei principi di usabilità denominati "**Euristiche**". Queste valutazioni richiedono un piccolo team di valutatori (3-5 persone), che in base ai risultati e alle violazioni dell'interfaccia vanno a rivedere le corrispettive parti.

1 Visibility of System Status

Designs should keep users informed about what is going on, through appropriate, timely feedback.



Interactive mall maps have to show people where they currently are, to help them understand where to go next.

2 Match between System and the Real World

The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon.



Users can quickly understand which stovetop control maps to each heating element.

3 User Control and Freedom

Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action.



Just like physical spaces, digital spaces need quick "emergency" exits too.

4 Consistency and Standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.



Check-in counters are usually located at the front of hotel which meets expectations.

5 Error Prevention

Good error messages are important, but the best designs carefully prevent problems from occurring in the first place.



Guard rails on curvy mountain roads prevent drivers from falling off cliffs.

6 Recognition Rather Than Recall

Minimize the user's memory load by making elements, actions, and options visible. Avoid making users remember information.



People are likely to correctly answer "Is Lisbon the capital of Portugal?".

7 Flexibility and Efficiency of Use

Shortcuts — hidden from novice users — may speed up the interaction for the expert user.



Regular routes are listed on maps, but locals with more knowledge of the area can take shortcuts.

8 Aesthetic and Minimalist Design

Interfaces should not contain information which is irrelevant. Every extra unit of information in an interface competes with the relevant units of information.



A minimalist three-legged stool is still a place to sit.

9 Recognize, Diagnose, and Recover from Errors

Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.



Wrong-way signs on the road remind drivers that they are heading in the wrong direction.

10 Help and Documentation

It's best if the design doesn't need any additional explanation. However, it may be necessary to provide documentation to help users complete their tasks.



Information kiosks at airports are easily recognizable and solve customers' problems in context and immediately.

1. L'utente deve sempre avere un quadro generale del sistema, questo attraverso feedback veloci e utili.
2. Il sistema deve parlare un linguaggio comprensibile all'utente, è meglio usare termini familiari che tecnicismi.
3. In caso di errori da parte dell'utente, devono essere presenti meccanismi di recupero.
4. Gli utenti non devono mai chiedersi se parole, azioni o situazioni differenti possano avere lo stesso significato.
5. Dei buoni feedback sono importanti, ma è importante prevenire gli errori più che risolverli.
6. Minimizzare il carico cognitivo, ponendo elementi, azioni e opzioni visibili. Si deve evitare di far ricorrere l'utente alla memoria.
7. Implementare funzionalità nascoste per utenti esperti per aumentare l'efficienza col sistema.
8. L'interfaccia non deve mai contenere informazioni superflue o irrilevanti.
9. Un errore deve essere spiegato in modo semplice, deve essere indicato chiaramente e si deve suggerire una possibile soluzione.
10. Non è necessaria, ma è sempre una buona pratica fornire una documentazione all'utente.

Cognitive Walkthrough

Metodo di valutazione basato sull'idea che l'utente scopre l'interfaccia esplorandola con determinati obiettivi in mente, condotto da un esperto che esegue una serie di compiti imitando l'utente.

PREPARAZIONE	PROCESSO	DOMANDE
<ol style="list-style-type: none">1. <u>Descrizione Utente</u> incluso livello di familiarità con il computer ed eventuali ipotesi del progettista.2. <u>Descrizione di Sistema</u> incluse operazioni e prestazioni.3. <u>Descrizione del task</u> specifico che l'esperto deve condurre dal pov dell'utente.4. <u>Sequenza di azioni</u> che descrive il system display e cosa l'utente deve fare per completare il task. 1 system display + 1 user action = 1 step.	<p>Il prototipo, la descrizione utente, di sistema e del task e la sequenza di azioni devono essere preparati dal designer.</p> <p>Il designer consegna i precedenti documenti a un esperto.</p> <p>L'esperto legge le descrizioni e conduce il task seguendo l'elenco di azioni.</p>	<p>L'esperto pone le seguenti domande per ogni step della sequenza di azioni:</p> <ol style="list-style-type: none">1. L'obiettivo successivo è chiaro in questa fase?2. L'azione corretta è ovvia?3. È chiara la conseguenza di quest'azione?4. Quali problemi si riscontrano nell'effettuare l'azione?

Test di Usabilità

Un **test di usabilità** prevede un ricercatore (facilitatore, moderatore) che chiede a un partecipante di eseguire determinati compiti, di solito utilizzando una o più interfacce specifiche. Durante il test il ricercatore osserva le decisioni dell'utente e raccoglie feedback.



Facilitator

Guides the participant through the test process



Tasks

Realistic activities that the participant might actually perform in real life



Participant

Realistic user of the product or service being studied



- **Facilitatore:** colui che controlla il corretto andamento del test e guida i partecipanti. Può dare istruzioni sui tasks, rispondere alle domande dei partecipanti e farne a sua volta.

- **Tasks:** attività che il partecipante effettua attraverso la piattaforma, possono essere date verbalmente o per scritto. La formulazione dei tasks è molto importante, poichè le ambiguità potrebbero confondere.

- **Partecipante:** possibile utilizzatore del prodotto, tipicamente già conosce il prodotto (versioni precedenti) o ha familiarità con prodotti simili, in alcuni casi si prendono partecipanti che non hanno alcuna conoscenza. Andrebbero selezionati in base alle personas. Spesso viene chiesto loro di “pensare ad alta voce” per meglio comprenderne le azioni.

Esistono due tipi di test:

Qualitativi

Ci si concentra sulla qualità dei dati ricavati, viene effettuato attraverso risposte aperte, osservazioni e interviste. Tipicamente effettuato nelle prime fasi del design per capire i bisogni, obiettivi e interessi dell'utente. Usato anche per valutare i prototipi prima del lancio.

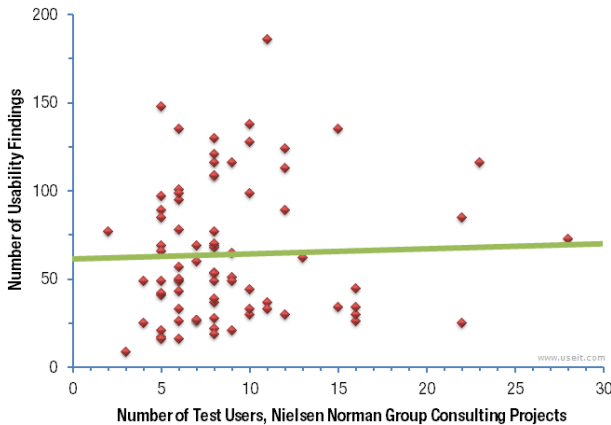
Quantitativi

Permette di ricavare molti più dati rispetto ai test qualitativi, a discapito della qualità. Si ricavano dati numerici in base ai comportamenti e attitudini dell'utente, questo grazie a test statistici. I designer possono misurare il comportamento e le preferenze degli utenti in modo più oggettivo e sistematico, identificando modelli e tendenze.

	Risponde a	Obiettivi	Quando si usa	Risultati	Metodologia
Qualitativa	Perché?	Formativa e sommativa: informare le decisioni e identificare problemi.	In qualsiasi momento.	Basati su impressioni e interpretazioni.	Pochi partecipanti, protocolli flessibili, think-aloud.
Quantitativa	Quanti e quanto?	Sommativa: valutare e tracciare l'usabilità	Con un prodotto funzionante.	Statisticamente significativi.	Molti partecipanti, protocolli rigorosi.

La **pianificazione e conduzione** dei test di usabilità è cruciale. Le fasi chiave includono:

1. **Definire l'obiettivo della ricerca** (cosa testare, lo scopo finale), aiuta a scegliere quale ricerca usare.



2. **Definire il formato:** in laboratorio o sul campo, moderato o non moderato, di persona o remoto.

3. **Stabilire il numero di partecipanti:** Jakob Nielsen suggerisce 5 per studi, almeno 20 per studi quantitativi.

4. **Reclutare i partecipanti giusti:** devono essere rappresentativi del target.

5. **Scrivere i compiti giusti:** devono essere realistici e corrispondere agli obiettivi dello studio. Possono essere *esplorativi* (aperti, non adatti per ricerca quantitativa) o *specifici* (con un obiettivo, adatti per entrambi i tipi).

6. **Fare una prova:** eseguire uno pilot study prima del test vero e proprio per affinare la formulazione dei compiti, l'ordine, la quantità e verificare il reclutamento.

7. **Decidere le metriche** (se presenti): negli studi qualitativi sono meno importanti; ci si concentra sugli insight di design e si pianificano le domande per interviste. Negli studi quantitativi, le metriche sono importanti e includono tempo, livello di soddisfazione, tasso di successo e di errore.

8. **Scrivere il piano:** documentare l'approccio in un piano da condividere. Deve contenere informazioni chiave come nome del prodotto, obiettivi, logistica, profili dei partecipanti, compiti, metriche, questionari, descrizione del sistema.

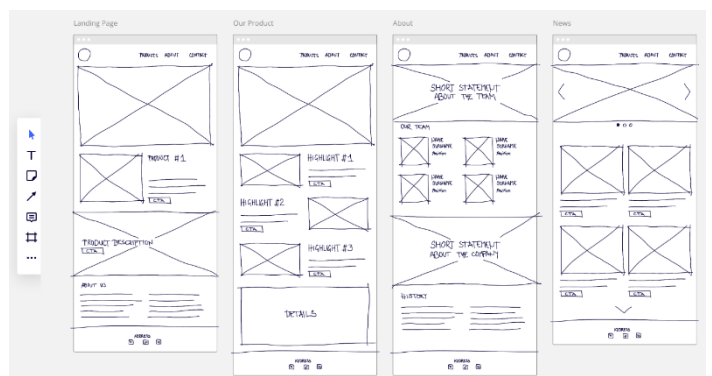
9. **Non farlo da soli:** avere altri tester aiuta a cogliere tutti i dettagli e prendere appunti.

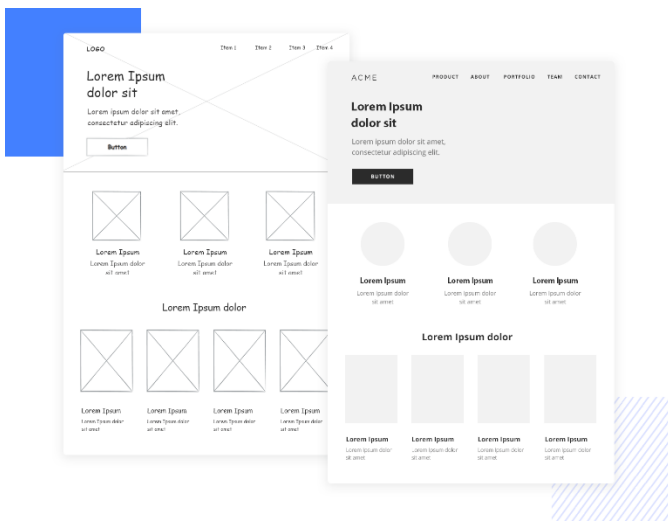
Abbiamo inoltre un'indicazione generale su **10 cose da NON fare durante la fase di test:**

1. **Dire agli utenti dove andare:** evita di suggerire dove cliccare o quale percorso seguire (*priming*).
2. **Dire agli utenti cosa fare:** non anticipare i passaggi necessari per completare un compito.
3. **Usare task obsolete:** assicurati che i compiti si basino su funzionalità o informazioni attuali.
4. **Compiti troppo semplici:** i compiti devono richiedere all'utente di interagire e elaborare informazioni, non solo di localizzare un elemento banale.
5. **Scenari elaborati:** mantieni gli scenari di contesto semplici e pertinenti, evitando dettagli inutili o complessi.
6. **Scrivere annunci invece che compiti:** usa un linguaggio orientato all'utente, evitando frasi di marketing o gergo aziendale.
7. **Rischiare reazioni emotive:** evita dettagli nei compiti che potrebbero toccare corde emotive sensibili per i partecipanti (es. malattie, lutti), per garantire il loro benessere.
8. **Cercare di essere divertente:** evita battute, nomi famosi o tentativi di alleggerire l'atmosfera in modo forzato, perché potrebbero mettere a disagio i partecipanti.
9. **Offendere il partecipante:** evita di includere nei compiti riferimenti a temi sensibili che hanno un alto potenziale offensivo (politica, religione...).
10. **Chiedere – piuttosto osserva:** non chiedere ai partecipanti "come lo faresti...?"; lo scopo del test è osservare cosa fanno realmente quando interagiscono con l'interfaccia e non cosa pensano di dover fare.

Wireframes e Responsive Design

I wireframe sono progetti di base, *blueprints*, che facilitano la comunicazione tra designer e programmatori sulla struttura di un sito web o di un'applicazione. Sono schizzi semplificati che mostrano la struttura e il layout dell'interfaccia utente, senza colori, immagini o caratteri complessi.





Si trattano di *sketches* visivi, perciò privi di alcuna funzionalità. Aiutano a chiarire il processo di pensiero e a trasmettere un'idea in modo efficace ad altre persone, evitando fraintendimenti e assicurando che tutti lavorino nella stessa direzione, ma non illustrano l'aspetto finale dell'interfaccia, e non garantiscono sempre che l'idea originale venga compresa!

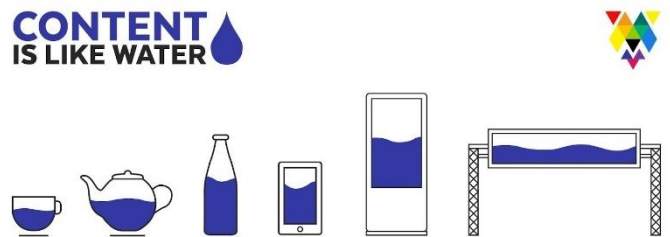
Wireframe a bassa fedeltà: schizzi semplici focalizzati su struttura e layout

Wireframe ad alta fedeltà: rappresentazioni dettagliate con tipografia di base, icone e dettagli di documentazione.

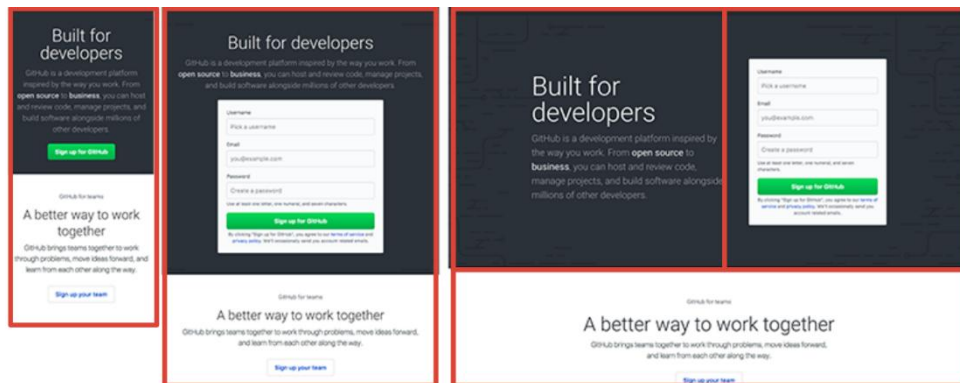
L'**adattabilità** nel design UX si riferisce alla capacità di un progetto di adattarsi e rispondere alle diverse esigenze, contesti, abilità e preferenze degli utenti. Il contesto gioca un ruolo importante nel design UX, influenzando il comportamento e le preferenze degli utenti: un utente su un dispositivo mobile ha esigenze diverse da un utente su un desktop.

Responsive Design

È un metodo di progettazione che comporta la codifica di pagine web e app in maniera che si adattino e permettano un'esperienza coerente su diversi dispositivi. È essenziale per garantire che i siti web siano leggibili e utilizzabili su una vasta gamma di schermi, semplificando anche la manutenzione e l'aggiornamento di essi.



Tecniche di Responsive Design:



Griglia flessibile: gli elementi si adattano alle dimensioni dello schermo.

Immagini flessibili: le immagini si ridimensionano in base allo schermo.

Media query e breakpoint: stili diversi applicati allo stesso elemento in base alle caratteristiche del dispositivo.

Mobile First: una filosofia di progettazione che dà la priorità ai dispositivi mobili rispetto al desktop, garantendo che l'esperienza mobile sia ottimizzata e accessibile prima di progettare per schermi più grandi. Consente un design "*progressive enhancement*": si parte alle funzioni minime e dalle interazioni di base (tipiche degli smartphone) per poi aggiungere interazioni ed effetti più avanzati (accessibili sui desktop).

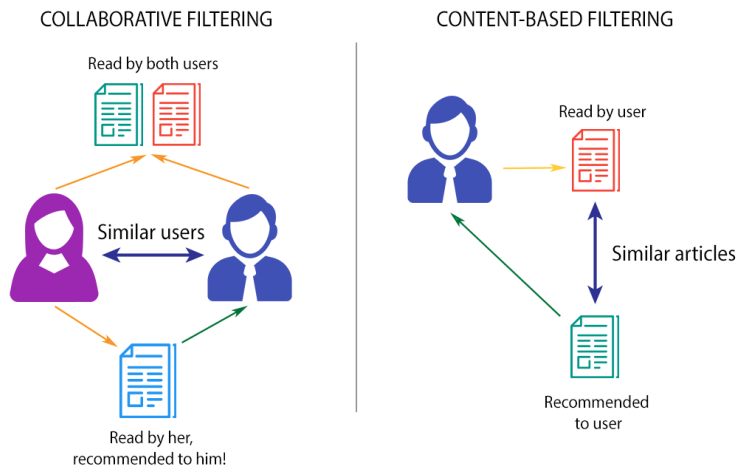
Al contrario, la "*grateful degradation*" parte da un design completo per dispositivi avanzati e riduce le funzionalità per adattarsi a dispositivi più piccoli e meno potenti.

Buone pratiche:

- Mantenere un'**interfaccia semplice**.
- Dare **priorità al contenuto** e alla sua leggibilità.
- Concentrarsi sulla **navigazione touch** (dimensione dei pulsanti, ...).
- **Ottimizzare il tempo di caricamento**.
- Fare **test frequenti**.

Adaptive Design

È un altro modo per rendere l'interfaccia adattabile all'utente, creando layout diversi per ogni dispositivo. È più costoso, ma a volte è preferibile (es. per le grandi aziende che vogliono migliorare il proprio sito per mobile senza dover dedicare tempo all'aggiornamento della versione desktop).



Il criterio di adattamento dipende dal contesto d'uso: può essere il dispositivo in uso, ma anche **l'utente stesso** → **Adaptive UX**.

Un sistema può adattarsi all'utente utilizzando informazioni di base come la posizione GPS, il collaborative filtering o i dati analitici. Il filtraggio può essere:

Collaborativo: suggerisce contenuti in base alle preferenze di utenti simili (user A likes X & Y, user B likes X, so they may be interested in Y).

Content-based: suggerisce contenuti simili a quelli che l'utente ha già apprezzato.

Accessibilità

Abilità essenziale nel design di UXD che consiste nel permettere a chiunque di usare il prodotto/piattaforma a prescindere dalle **abilità** (*fisiche, cognitive, comportamentali*) dell'utente. Alternative testuali per immagini non testuali (foto, video, audio).

Esempi di problemi di accessibilità:

- **Visivo** (es. daltonismo).
- **Motorio/mobilità** (es. problemi degli utenti su sedia a rotelle).
- **Uditivo** (es. difficoltà di udito).
- **Convulsioni** (in particolare epilessia fotosensibile).
- **Apprendimento/cognitivo** (es. dislessia)

Le **barriere** possono sorgere per qualsiasi utente, per fattori **incidentali** (es. privazione del sonno) o **ambientali** (es. utilizzo di un dispositivo mobile in metropolitana).

WCAG (Web Content Accessibility Guidelines): standard di accessibilità riconosciuti a livello internazionale per rendere i contenuti web più accessibili alle persone con disabilità. Alcuni esempi di best practice sono:

- **Testo alternativo** per contenuti non testuali (es. immagini, icone, grafici).
- **Didascalie** e alternative per i contenuti multimediali (es. sottotitoli sui video).
- **Modalità di presentazione del contenuto** (es. layout, ingrandimento, lettura ad alta voce, colori).
- **Contenuto facile da vedere e ascoltare** (es. dimensione del carattere, contrasto).
- **Diverse modalità di input** oltre alla tastiera.
- **Navigazione semplice**.

Non si può pensare all'adattabilità solo alla fine del processo di progettazione, deve essere un'attenzione integrata in tutto il processo, dalla fase di ricerca a quella di test e implementazione.

Recap: come progettare per l'adattabilità?



User Research:
comprendere le esigenze, i comportamenti e le preferenze degli utenti nei diversi contesti.



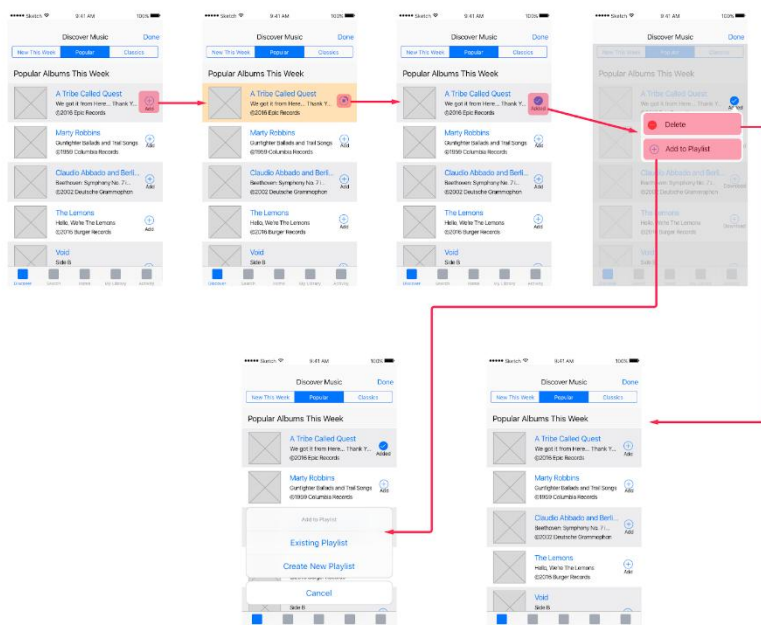
Responsive Design:
creare design che si adattano a diverse dimensioni dello schermo e dispositivi.



Accessibilità:
garantire che il progetto sia fruibile dalle persone con disabilità.



Testing:
convalidare il progetto in diversi contesti e apportare modifiche di conseguenza.



Da Wireframe a Wireflow

Gli **user flows** descrivono le azioni che un utente può compiere sull'interfaccia, mappando tutti i possibili passaggi e movimenti.

Gli user flows possono essere rappresentati con diagrammi di flusso che definiscono come un *personas* utilizza il prodotto. Si parte dalla user *personas*, dal requirement e dallo scenario.

I **wireflow** sono user flows espressi tramite wireframe, offrono una rappresentazione visiva più dettagliata dell'interazione dell'utente con l'interfaccia.

Prototipazione & Mockups

Un **mockup** è una rappresentazione ad alta fedeltà del prodotto che include più dettagli visivi e elementi di design (tipografia, schemi di colori, immagini) rispetto a un wireframe. Assomiglia al prodotto finale ma essendo una rappresentazione statica non ha elementi interattivi. Viene utilizzato perfezionare l'aspetto e comunicare il concept più efficacemente.

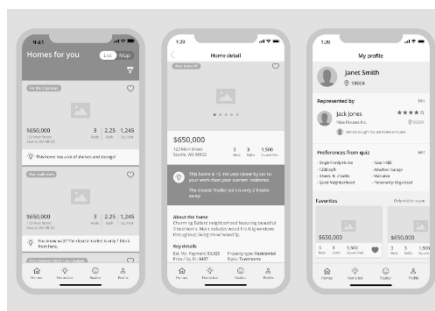
Un **prototipo** è un campione tangibile, sperimentale e precoce di un progetto che consente agli utenti di visualizzarlo e interagire con esso prima che venga sviluppato il prodotto finale. I prototipi consentono ai progettisti di testare le loro idee sugli utenti e di perfezionare i loro progetti in base al feedback prima di impegnarsi nello sviluppo finale.

Esistono dei **livelli di fedeltà**:



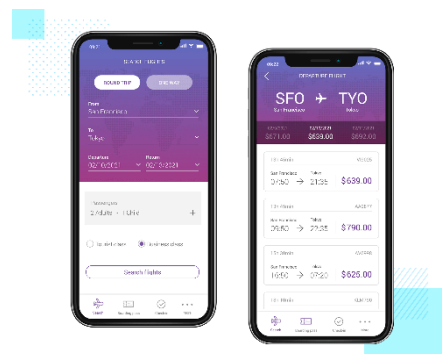
Bassa

Modelli fatti con carta e penna, creati nell'arco di qualche ora. Semplici ed economici, danno solo un'idea visiva e fisica del prodotto, permettendo di utilizzare solo funzionalità molto semplici (o nessuna).



Media

Modello con una maggiore complessità che permette agli utenti di utilizzare le funzionalità base o le più note del prodotto (es. wireframe con interazioni).



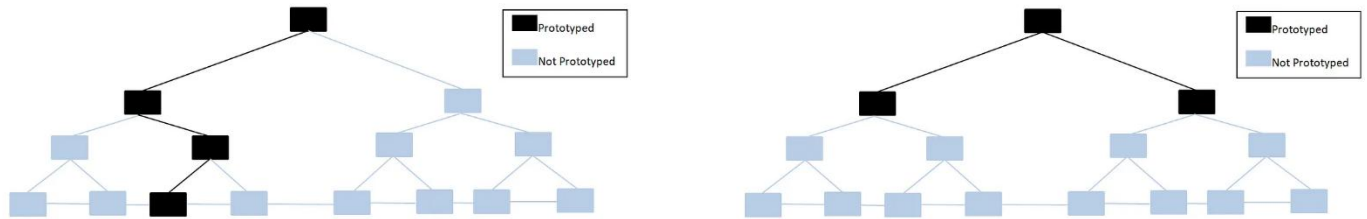
Alta

Modello molto vicino al prodotto finito. Può richiedere parecchio tempo e risorse, perché permette all'utente di avere un utilizzo completo del prodotto. Creato con appositi software.

La fedeltà del prototipo ha 6 dimensioni principali:

- **Visiva/fisica.**
- **Portata:** quali funzionalità o elementi del sistema vengono rappresentati.
- **Funzionalità:** cosa attualmente funziona.
- **Dati:** se opera con dati reali o meno.
- **Autonomia:** se richiede input dall'utente o è in grado di lavorare da sé.
- **Funzionalità:** se è un ver. parziale o finale.

La **portata** inoltre può essere **verticale** o **orizzontale**. Un prototipo verticale rappresenta in grande dettaglio una (o poche di più) caratteristica specifica, mentre un prototipo orizzontale rappresenta una panoramica meno precisa ma dell'intero sistema.



Le **tecniche di prototipazione** includono:

- **Prototipi cartacei:** di basso livello, simula il layout del prodotto/piattaforma e funzionalità semplici. Veloce ed economico, serve per capire se il layout ideato è adatto prima di fare grossi investimenti.
- **Mid-fi Wireframe:** prototipo digitale che rappresenta layout e funzionalità del prodotto, senza scendere nei dettagli. Serve soprattutto per vedere come è distribuita l'informazione e i vari elementi.
- **Wizard of Oz:** prototipo che sfrutta l'*input umano* per simulare una tecnologia che richiederebbe tempo e costi per essere sviluppata. Utilizzato per testare l'esperienza utente con quella tecnologia.
- **Funzionali:** prototipi che includono una funzionalità completa o parziale. Utili nelle fasi avanzate del processo, in cui il designer affina i dettagli tecnici prima di passare allo sviluppo vero e proprio.
- **Non-funzionali:** prototipo che sembra un prodotto vero e proprio ma privo di alcuna funzionalità. Usato per mostrare l'aspetto o l'interfaccia del prodotto senza dover sviluppare i componenti del sistema. Possono essere utilizzati anche nei testing sull'User Experience sul Visual Design.

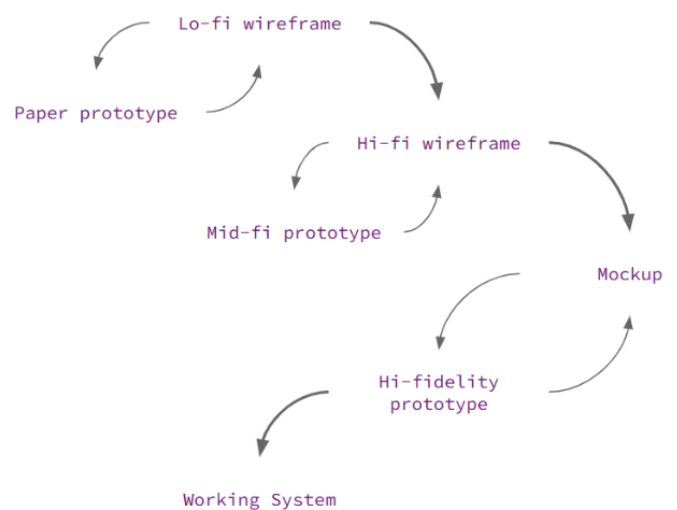
Per prodotti **interattivi** o che riescono a dare risposte in base alle varie azioni, è preferibile il **prototipo funzionale**.

Se invece si vuole testare l'aspetto fisico/visivo della piattaforma, con tipografia, design visuale, colori e/o immagini è meglio il **Mockup**.

Altrimenti come ultima alternativa si usa il **Wireframe**.

early design

late design



Pretotipazione

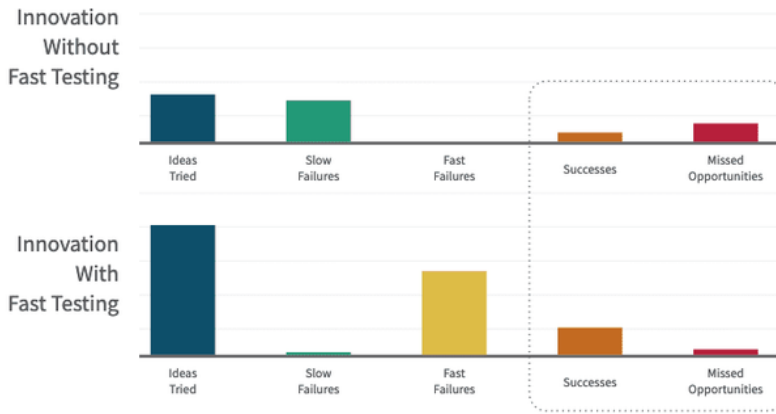
La **legge del fallimento di mercato:** si deve constatare quali idee di prodotto sono interessanti per il mercato prima di investirci troppo tempo e denaro per lo sviluppo completo. I **Pretotipi** sono tecniche e strumenti che permettono di



capire velocemente quali idee sono valide, collezionando dati sull'utilizzo e sul mercato per capire quali decisioni mantenere o lasciare riguardo lo sviluppo del prodotto, investendo meno risorse possibili.

- **80-90%** new products fail in the market [1]
- **4/5** startups lose funder's money [2]
- **90%** mobile apps don't produce money
- **78%** project disappear within 4 years
- **16% to 50%** strategic projects are abandoned [3]
- **47%** companies estimates **50%** of projects fail [3]

capire velocemente quali idee sono valide, collezionando dati sull'utilizzo e sul mercato per capire quali decisioni mantenere o lasciare riguardo lo sviluppo del prodotto, investendo meno risorse possibili.



A volte la pretotipazione produce **falsi positivi** (prodotti apparentemente validi che non hanno ottenuto in realtà il successo sperato) e **falsi negativi** (viceversa). La causa è riconducibile a **due problemi**: il primo è che le idee si “perdono nella traduzione”: quando si comunica un’idea al pubblico, non è detto che *loro* se la stiano immaginando come invece la stai immaginando *te*; il secondo problema è che, anche se la tua idea viene compresa, le persone generalmente non sono in grado di predire correttamente se

qualcosa gli piacerà o lo useranno effettivamente prima di averlo sperimentato. Per minimizzare la possibilità di ricevere falsi positivi o negativi che manderebbero fuori strada il tuo sviluppo, c’è bisogno di raccogliere dati più sostanziali di semplici opinioni.

(Thoughtland) Idee astratte → domande ipotetiche → **opinioni**

▼ *ci spostiamo in ambiente più concreto* ▼

(Actionland) Artefatti → azioni → **dati**

Il Manifesto della Pretotipazione

Innovators beat Ideas

Pretotypes beat Productypes

Building beats Talking

Simplicity beats Features

Now beats Later

Commitment beats Committees

Data beats Opinions

ADDENDUM
Don't finish what you've started
Failure is an option
Scarcity brings clarity
The more the messier
Reinvent the wheel
Play with fire

Pretotyping MEMES

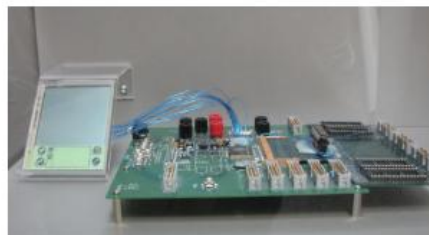


I prototipi possono aiutarti a fallire più velocemente, ma non senza averci speso tempo e denaro: più investi in qualcosa più è difficile da lasciar andare., quando hai un prototipo che funziona, è allettante continuare a lavorarci sopra aggiungendo aspetti che pensi piaceranno alle persone. I prototipi spesso diventano “*productypes*” – un prototipo esagerato, e puoi dire addio al fallimento veloce. La pretotipazione, invece, ti permette di fallire e riprenderti velocemente, con poche risorse, e con la possibilità di esplorare innumerevoli idee finché non arrivi a quella giusta.

Pretotype



Prototype



Product



I 7 Pilastri della Pretotipazione

1. Rispettare la legge del fallimento del mercato.
2. Assicurarsi di essere sulla strada giusta.
3. Non perdersi in “Thoughtland”.
4. Trust Only in Your Data (YODA).
5. Pretotipalo.
6. Dillo con i numeri.
7. Pensa globalmente, testa localmente.

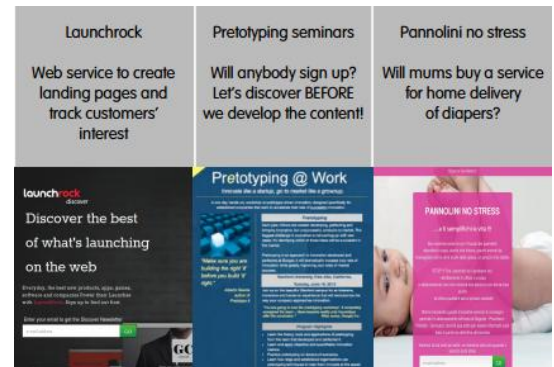
Flusso di Pretotipazione

1. **Isolare l'ipotesi chiave:** qual è l'unica ipotesi sulla tua idea che, se falsa, significa che non è sicuramente quella giusta?
2. **Scegliere un tipo di pretotipo:** quale tipo di pretotipo ti permetterà di isolare e testare la tua ipotesi chiave?
3. **Fare un'ipotesi di mercato:** *il X% di Y farà Z → A.*
4. **Testa il tuo pretotipo:** guarda come le persone interagiscono con il tuo prodotto.
5. **Impara, ridefinisci, hypozoom:** controlla i risultati, ridefinisci il pretotipo in base ai dati, decidi in quali nuove situazioni dovresti testare il tuo pretotipo per avere un quadro generale (*hypozoom*).

Metodi di Pretotipazione

FAKE DOOR

What	A marketing entry point for an as-yet undeveloped idea.
Why	The solution doesn't exist yet and you want to capture an initial indication of interest at next to 0 cost.
When	Your idea can be concisely described and presented to potential customers where you would expect to find them, and you are confident you can manage the expectations of enthusiastic customers by following up within an appropriate time-frame.
How	Advertising a new product or feature, then tracking click-through and customer response rate to see who would be interested in an offering.
Where	Web tech enables a very robust method that includes: online ads + landing pages + simple response forms. Same approach also works with emails, offline posters and other media.



Si vuole provare a catturare l'interesse dell'utenza prima che l'idea venga sviluppata, grazie a pubblicità che riescono a rilevare l'interesse di un utente (*visualizzazioni, click, form*) e a fare analisi o statistiche sulle visite. Molto utilizzate nelle applicazioni web.

MECHANICAL TURK

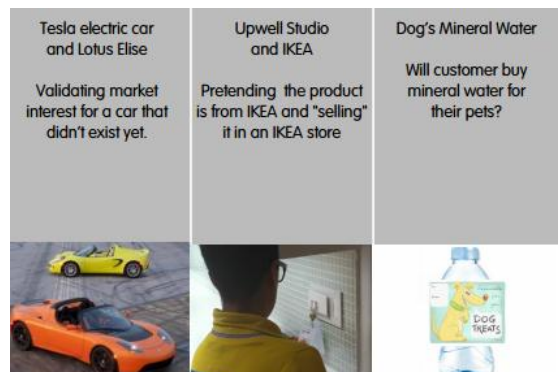
What	Use human power to simulate a technology that would take too much money or time to build
Why	To postpone costly development until market interest is validated.
When	When the final product requires the development of expensive and complex technology, and those actions and outputs could be simulated by humans.
How	Use a realistic interface to deliver target customers the essential experience of a proposed technology, simulating functionality of a complex back-end using human input.
Where	In the same real-life situation where the innovation will be used.



Si utilizza il *lavoro umano* per replicare una tecnologia che richiederebbe tempo e denaro per essere implementata. Utile per ammortizzare costi e tempi sino a quando non vi è un discreto interesse nel mercato.

IMPERSONATOR

What	Use an existing product or service to pose as the new offer under test.
Why	Save on development costs while the market interest is not yet validated.
When	A test of the value of the solution depends on the customers' ability to interact with a full-scale design, and you need to create a plausible stand-in for the size, shape, color, features, etc. of the solution.
How	Apply a new skin to an existing product that can act as a good substitute to validate market interest.
Where	In the same real-life situation where the innovation will be used or accessed.



Viene utilizzato un prodotto o servizio già esistente, si aggiunge qualche piccola modifica e lo si spaccia per nuovo. Permette di evitare investimenti superflui in caso di basso interesse nel mercato.



PINOCCHIO

What	An inanimate (or "dumb") artifact acts as a proxy for the real thing.
Why	The solution doesn't exist yet and you want to validate a key design parameter early on.
When	Your solution requires a significant switching or behavioral adaptation by customers to develop a new habit OR You expect demand to be sensitive to the appearance or form factor of your solution.
How	Use a proxy to validate certain parameters of the product like form factor, features and usability.
Where	In the same real-life situation where the innovation will be used.



Il prodotto o la soluzione al problema non esiste ancora e si crea un finto prodotto che permetta di simulare le funzionalità chiave. Usato per prodotti che richiedono un cambio significativo dal punto di vista pratico/comportamentale del cliente.



ONE NIGHT STAND

What	A complete service experience without the infrastructure required by a permanent solution.
Why	Avoid investment in complex infrastructure and validate market interest and actual use.
When	- The solution is — or depends critically upon — an interactive service experience. - You expect demand for the offer will be sensitive to the choice of channel, and you need to test a number of possible customer interception points. - You want to validate a large homogeneous market before scaling up.
How	Delivering target customers the essential experience within an extremely narrow geo scope and time frame.
Where	In the same real-life situation where the innovation will be used but with limited time and geo scope.



Si offre un servizio completo senza un'infrastruttura perenne. Permette di evitare investimenti su infrastrutture e di focalizzarsi sul prodotto, in questi casi la soluzione è spesso un servizio interattivo. Si prova a studiare in quali regioni sarebbe meglio investire nell'infrastruttura.



FACADE

What	Borrow or rent expensive equipment, space, and assets to simulate a more stable or complex infrastructure underlying your offering.
Why	Avoid investments in expensive equipment, space, and other assets while validating interest.
When	- The solution requires major upfront investment, in equipment, space, or assets. - You expect demand will vary based on customer confidence in your infrastructure.
How	Delivering target customers the essential experience, while communicating stability and complexity.
Where	In the same real-life situation where the innovation will be used, but with all assets and space borrowed or rented cheaply.

L' affitto di locali anche costosi per *simulare* un'infrastruttura. Questo permette di dare al cliente meno informato l'impressione di un'azienda organizzata.

Transizione alla prototipazione: una volta che hai raccolto tutto il possibile con i metodi di pretotipazione, ulteriori approfondimenti richiedono un'interazione più profonda con il cliente → creare un artefatto che fornisca le funzioni principali della soluzione completa, metterlo nelle mani dei clienti per consentire un test nella stessa situazione in cui verrà utilizzato il prodotto.

Strumenti di Pretotipazione

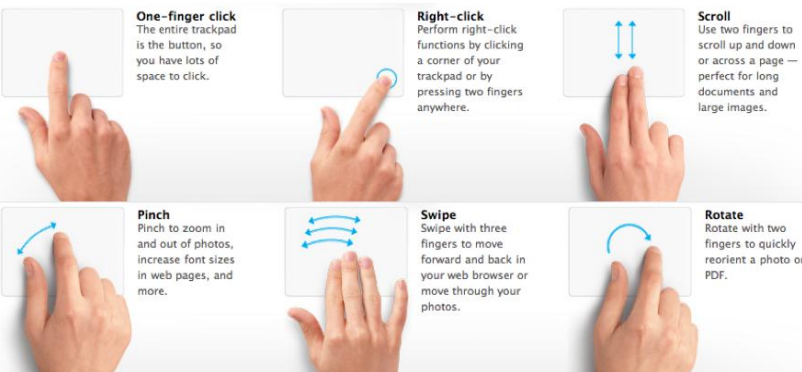
- *Website:* Weebly.com per creare "Fake Door" e testare l'interesse tramite pubblicità online.
- *Software di prototipazione:* Omnigraffle, Appery, Figma, etc. per creare prototipi digitali.
- *Marketplace builder:* Shopify per creare un marketplace online.

Natural User Interfaces

Concetto introdotto da Bill Gates, l'aggettivo *Natural* si riferisce all'obiettivo, che è creare un'esperienza utente in cui l'interazione con la tecnologia avvenga in modo naturale e intuitivo.

Generalmente si riferiscono a interfacce utente percepite come facili da usare, con un'interazione diretta e in linea con il comportamento umano innato: una NUI dovrebbe essere quasi invisibile e rimanere tale man mano che l'utente apprende interazioni sempre più complesse.

È importante distinguere tra un'interfaccia "intuitiva" e una "naturale": un'interfaccia intuitiva può essere utilizzata senza apprendimento preventivo, mentre una NUI richiede comunque un processo di apprendimento, ma questo dovrebbe essere facilitato da un design che renda l'utente immediatamente e continuamente capace di interagire con successo.



Esempio di NUI:

Alcuni gesti sull'iPad, come lo scorrimento con un dito, sono naturali e intuitivi. Questo perché il gesto corrisponde all'azione che si sta eseguendo, simulando il comportamento del mondo reale. Tuttavia, altri gesti, come lo scorrimento con quattro dita, richiedono un apprendimento dedicato perché non sono intuitivi.

Bill Buxton, ricercatore Microsoft, sottolinea che l'utilizzo di voce, gesti e tocco non garantisce automaticamente una NUI. Le NUI efficaci sfruttano le competenze acquisite nel corso della vita, riducendo il carico cognitivo e le distrazioni. Il design dovrebbe sempre considerare il contesto d'uso, poiché nessuna interfaccia può essere naturale per tutti gli utenti in ogni situazione; es. i bambini piccoli potrebbero non essere in grado di eseguire gesti complessi.

Joshua Blake delinea quattro **linee guida per la progettazione NUI**:

- **Competenza immediata:** l'utente deve essere in grado di utilizzare l'interfaccia fin dall'inizio.
- **Apprendimento progressivo:** deve consentire un apprendimento graduale di interazioni più complesse.
- **Interazione diretta:** l'interazione dovrebbe essere intuitiva e corrispondere alle aspettative dell'utente.
- **Basso carico cognitivo:** l'interfaccia deve utilizzare principalmente abilità innate e competenze semplici.

Un approccio alla progettazione di NUI è l'uso di **interfacce basate sulla realtà (RBI)**, che rendono gli oggetti del mondo reale "cliccabili", utilizzando dispositivi indossabili e collegandoli al cyberspazio. Tuttavia, le RBI sono solo un modo per realizzare le NUI, e non vanno confuse con il concetto stesso di NUI.



(a) Users' first-person view in HTC Vive



(b) Third-person view

Un altro approccio **non basato su RBI** è la limitazione di funzionalità e personalizzazione, come nel sistema operativo iOS di Apple. Questa strategia riduce al minimo l'apprendimento richiesto all'utente, rendendo l'interazione con il dispositivo semplice e intuitiva, a patto che le funzionalità predefinite corrispondano agli obiettivi dell'utente.

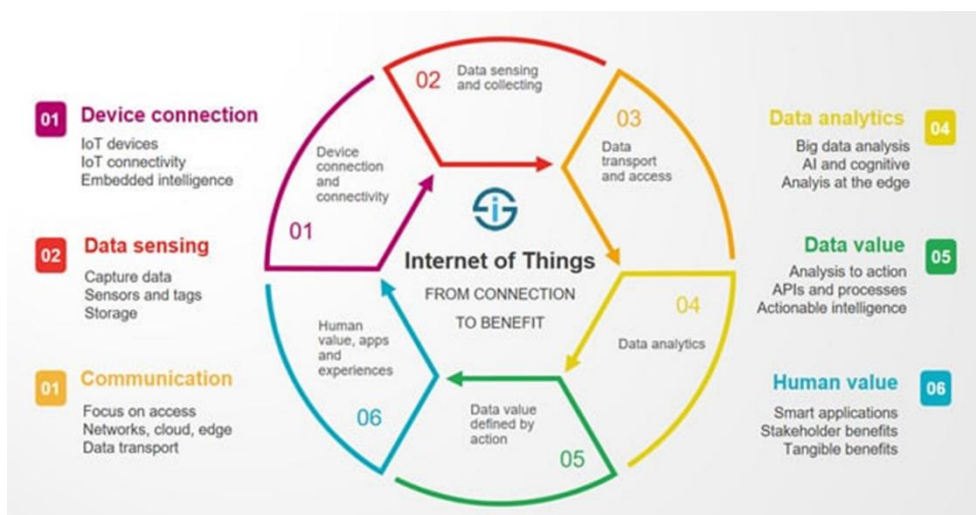
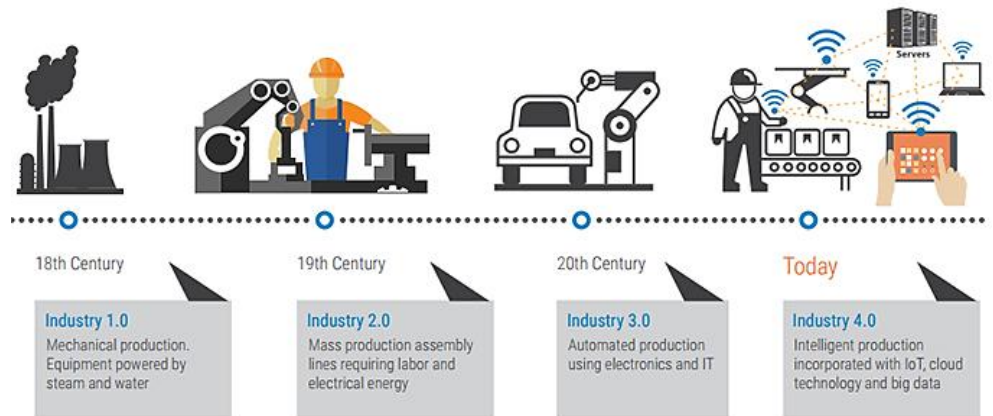
In poche parole, una NUI dovrebbe:

- Riuscire a sfruttare le conoscenze e abilità dell'utente.
- Avere un percorso di apprendimento chiaro, che permette un'interazione naturale a novizi ed esperti.
- L'interazione dovrebbe essere diretta e coincidere col contesto dell'utente.
- Quando possibile, dare priorità alle abilità innate dell'utente.

UX for IoT

L'Internet of Things (IoT) è un sistema di dispositivi interconnessi, oggetti, animali o persone dotati di identificatori univoci (UID) e la capacità di trasferire dati su una rete senza interazione umana diretta. Quando si pensa al design per prodotti connessi, l'attenzione tende a concentrarsi sugli elementi più visibili e tangibili: il design industriale, le interfacce utente (UI) in app mobile e web, e quelle sui dispositivi stessi; ma rappresentano solo una parte del quadro: anche con un'ottima UI e hardware, l'esperienza utente può comunque risultare insoddisfacente.

L'**Industry 4.0** descrive l'organizzazione dei processi produttivi basata su tecnologia e dispositivi che comunicano autonomamente lungo la catena: è un modello di fabbrica "intelligente" del futuro, dove sistemi computerizzati monitorano processi fisici. Questo concetto tiene conto della crescente digitalizzazione delle industrie manifatturiere.



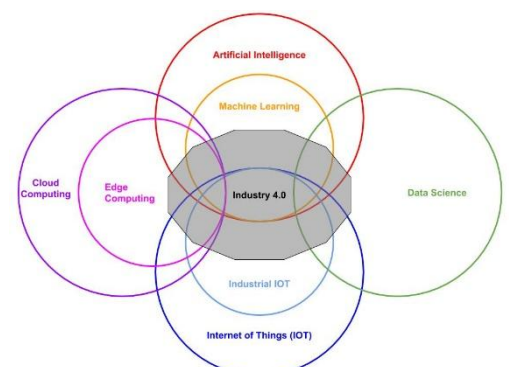
Un concetto correlato è quello di **Digital Twin**, ovvero una replica digitale di un'entità fisica. Questo ponte tra il mondo fisico e quello virtuale consente la trasmissione continua dei dati, permettendo all'entità virtuale di esistere simultaneamente con quella fisica. Può essere una replica digitale di beni fisici, processi, persone, luoghi, sistemi e dispositivi, utilizzabile per vari scopi.

Nell'Era 4.0 stiamo assistendo a un passaggio dai prodotti ai servizi. Si considerano diverse fasi dell'uso del prodotto, che insieme definiscono l'esperienza utente:

1. **BEFORE USE** ("anticipated use"): anticipazione dell'uso del prodotto, senza averlo ancora usato.
2. **DURING USE** ("actual use"): completamento del compito in modo efficiente e soddisfacente. È la fase dove entra in gioco l'usabilità.
3. **AFTER USE** ("digested use"): elaborazione e ritenzione emotiva rispetto al prodotto dopo l'uso.

Vengono considerate anche aree come il servizio clienti, la pubblicità, la ricerca sui clienti, la consegna del prodotto, l'etica e le vendite (prospettiva del cliente sui punti di contatto). Un prodotto smart è un dispositivo fisico con un servizio digitale compreso.

Progettare per l'IoT è intrinsecamente più complesso rispetto alla progettazione di servizi web, ciò è dovuto in parte allo stato attuale della tecnologia, alla nostra ancora immatura comprensione delle proposte di valore convincenti per l'IoT consumer, e al fatto che ci sono più aspetti di design da considerare: affrontarli in modo indipendente porta a un'UX incoerente. È quindi necessario un approccio **olistico**.



Molti degli elementi nell'IoT sono **Dispositivi Specializzati**: hanno hardware e software ottimizzati per funzioni specifiche. Le loro capacità UI possono estendersi oltre schermi e pulsanti per includere controlli fisici, audio, aptici, gesti, interazioni tangibili e altro. Mancano le guide di stile standard. Alcuni dispositivi potrebbero non avere alcuna capacità di input o output utente.

Sensori e attuatori collegano digitale e fisico, permettendo a oggetti nel mondo fisico di essere interconnessi. Le azioni nel mondo reale spesso non possono essere annullate, a differenza dei comandi digitali, il contesto fisico di utilizzo pone ulteriori sfide, come la robustezza, la minimizzazione delle distrazioni (ad esempio in auto), la sicurezza (ad esempio un forno controllato da remoto), i requisiti normativi e il complesso contesto sociale, specialmente in casa.

Molti prodotti sono sistemi di dispositivi e servizi diversi. Le funzionalità possono essere distribuite su più dispositivi con capacità differenti, è necessario progettare UI e interazioni attraverso l'intero sistema, e non trattando i dispositivi come UI a sé stanti, per garantire un'esperienza UX coerente (*interusability*). L'esperienza dell'intero sistema è spesso più importante dell'esperienza di un singolo dispositivo: l'obiettivo è creare un **ambiente unificato** per il sistema IoT, integrando componenti indipendenti in una soluzione "one-stop".

Mentre la progettazione UI tradizionale si basa sul contatto diretto (risultati visibili e reversibili), molte interazioni IoT sono **dislocate nello spazio (controllo remoto)** o **nel tempo (automazione)**. Ciò spezza il legame tra azioni dell'utente e conseguenze visibili e reversibili. I designer abituati al web e mobile danno per scontato una connessione quasi sempre presente, il che non è vero nell'IoT, inoltre i dispositivi spesso si connettono solo a intermittenza per risparmiare energia: i sistemi IoT distribuiti potrebbero non essere sempre sincronizzati, e dispositivi diversi potrebbero riportare informazioni diverse sullo stato del sistema.

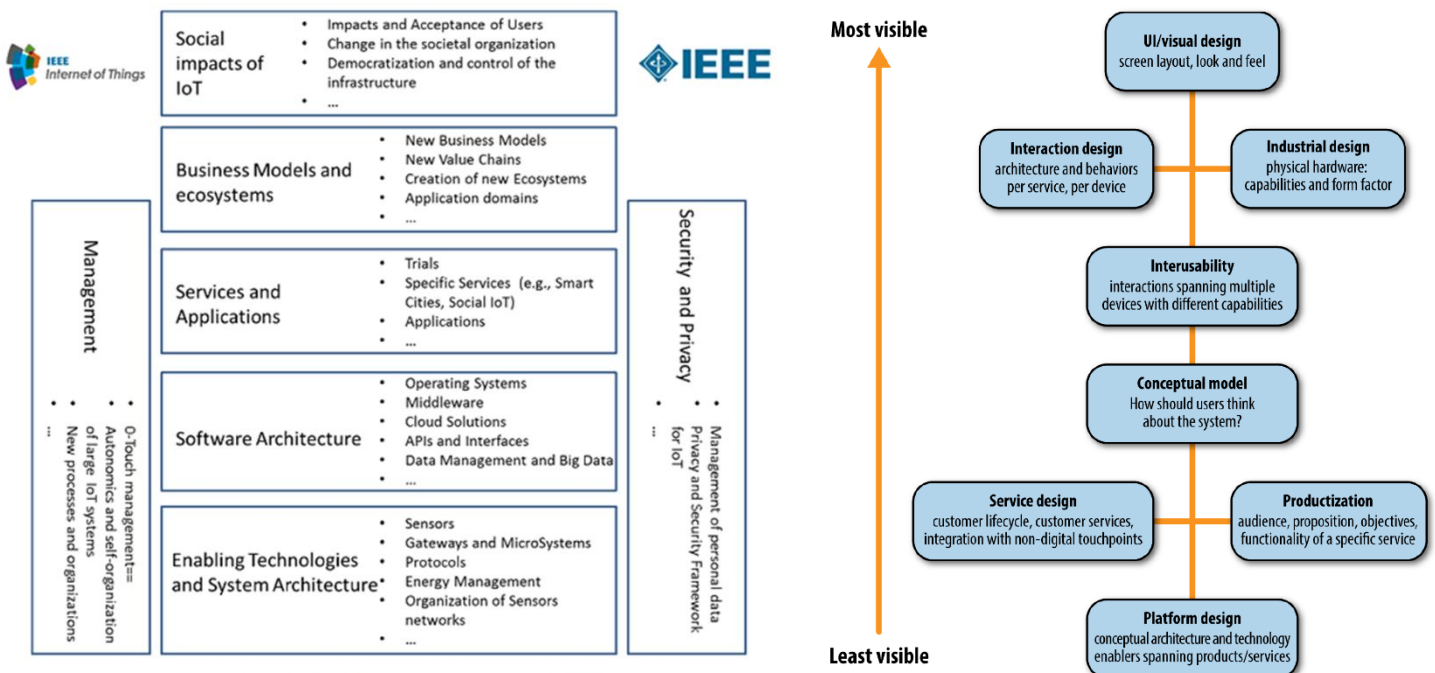
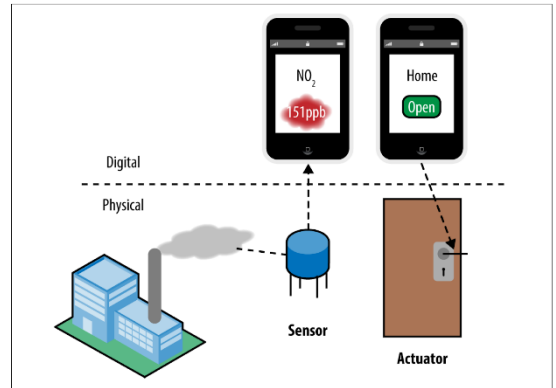


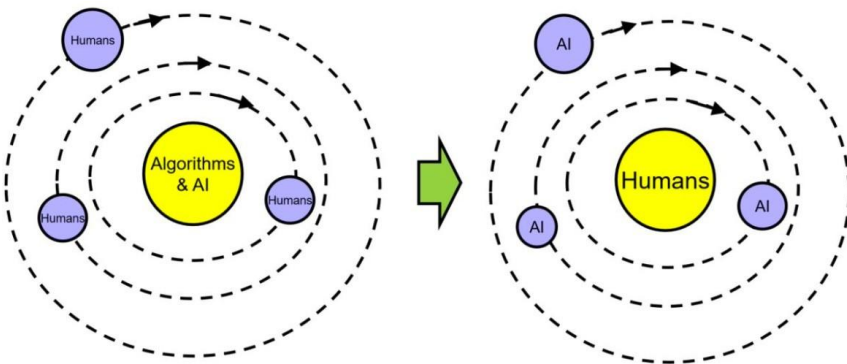
Figure 1. Technological and social aspects related to IoT

Human-AI Interaction

Quando l'AI funziona, è ottima, ma quando fallisce, lo fa in modo disarmante o problematico. Una possibile causa di questi problemi è che l'approccio tipico nello sviluppo AI/ML considera le interfacce utente o gli esseri umani come un pensiero secondario, spesso non includendoli affatto, concentrandosi principalmente su algoritmi e modelli. Eppure, i sistemi AI sono intrinsecamente legati agli esseri umani: sono **progettati da umani**, per risolvere un problema definito da umani, con umani che prendono decisioni

specifiche (es. scelta algoritmo), sono **valutati e testati da umani**, spesso, l'esito è per gli umani, ed è presentato a loro tramite un'interfaccia utente. Pertanto, un'**interfaccia utente adatta è critica** per superare alcune limitazioni. Mantenere le persone coinvolte e considerate fin dall'inizio è fondamentale per risolvere il *cold start problem*, che si presenta nel momento in cui implementiamo un algoritmo: se usi Netflix per la prima volta in assoluto, cosa dovrebbe consigliarti?

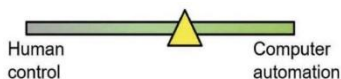
I sistemi basati sull'AI operano tipicamente in condizioni di **incertezza**, producendo spesso falsi positivi e falsi negativi, e possono avere **comportamenti imprevedibili** talvolta dannosi, confusi, offensivi e persino



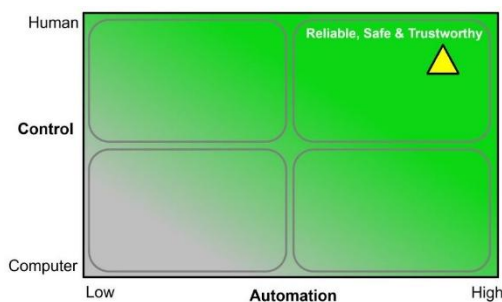
pericolosi per gli utenti. Ciò che contraddistingue i sistemi AI interattivi è il focus sulla **Human-Centered AI**, che mira a migliorare il contributo umano in modo che i sistemi AI diventino affidabili e sicuri: invece di valutare solo le prestazioni dell'algoritmo, vengono prese in considerazione l'esecuzione e la soddisfazione umana con approcci partecipativi. Ciò comporta un **cambiamento paradigmatico**.

Questo porta a chiedersi se si debba progettare tecnologia che **automatizzi** le azioni delle persone e **sostituisca** gli umani (Artificial Intelligence) o che li **aumenti** (Intelligent Augmentation). Invece di contrapporre il controllo umano all'automazione, si può considerare un approccio che **automatizza mentre aumenta**. Misurare l'"intelligenza" dell'AI risulta complicato, è più fattibile misurare la **prestazione congiunta** di un compito svolto da un umano supportato dall'AI: la **collaborazione** è la chiave, insieme possono fare cose per le quali individualmente non sarebbero stati in grado.

È essenziale che le persone si sentano in **controllo** dell'ambiente circostante. Quando si rendono "intelligenti" gli oggetti, le persone dovrebbero sentirsi **a proprio agio** con le azioni del sistema, **capire** perché stanno accadendo certe cose e **fidarsi** del sistema: l'automazione incontra tipicamente resistenza, sebbene possa ridurre il carico di lavoro e consentire compiti pericolosi.



Consideriamo un **framework bidimensionale** per la Human-Centered AI, superando la dicotomia semplice tra controllo umano e automazione vista finora.



Emulation

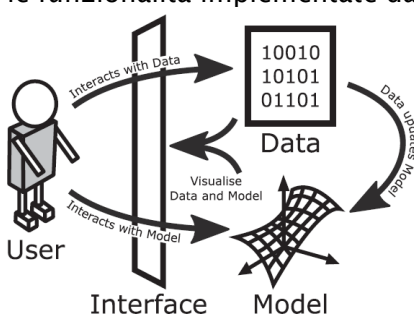
Mira a sistemi autonomi, indipendenti, capaci di stabilire obiettivi e auto-monitoranti, che emulino l'intelligenza umana (es. "compagni", "partner", robot umanoidi). Supporta obiettivi non anticipati e comportamenti emergenti basati su nuovi dati.

Application

Sviluppo di prodotti e servizi che applicano l'AI per raggiungere obiettivi di progettazione, con alti livelli di controllo umano e alti livelli di automazione informatica. Utilizza i metodi dell'HCI (Human-Computer Interaction) per identificare bisogni e obiettivi.

L'**Interactive Human-Centered AI** viene definita come un sistema che consente esplorazione e manipolazione interattiva in tempo reale, progettata con uno scopo chiaro: il **beneficio umano**. L'interazione è fondamentale per la collaborazione, essendo il canale attraverso cui l'AI supporta gli utenti.

Comprendere questi sistemi va oltre la semplice visualizzazione del codice: si tratta di capire gli obiettivi e le funzionalità implementate dall'AI. Le **proprietà** dell'Interactive Human-Centered AI includono:



- Interazione e manipolazione in tempo reale di algoritmi, modelli e dati.
- Osservazione in tempo reale dell'impatto dei cambiamenti.
- Esplorazione interattiva del perché/come vengono prese decisioni.
- Riduzione della velocità nei processi veloci per consentire interazione.
- Trasparenza sui benefici per gli umani.
- Spiegazione dei rischi legati all'utilizzo (individuali e sociali).
- Trasparenza su chi ha il controllo sull'AI (dati, modelli, algoritmi).
- Trasparenza su dati, fonti e informazioni utilizzate per addestrare l'AI.

I sistemi basati sull'AI possono essere visti come:

Smart Tools

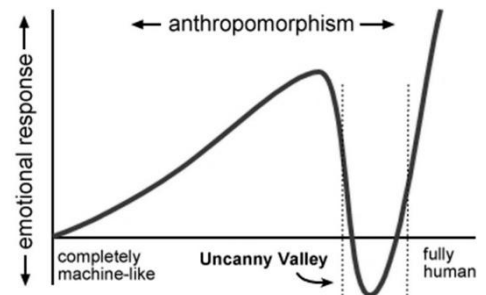
Artefatti cognitivi (oggetti fisici o digitali che migliorano la cognizione umana) potenziati dall'intelligenza artificiale. Assomigliano a GUI standard e mirano ad alleggerire alcuni compiti in autonomia. Gli utenti dovrebbero avere il controllo tramite l'interfaccia, ma il modello probabilistico sottostante può creare confusione tra autonomia e controllo.



Artificial Companions

L'interazione si basa sul modello di quella da umano a umano. Le persone tendono ad antropomorfizzare (attribuire intenzioni umane agli oggetti), queste caratteristiche antropomorfe possono inizialmente aumentare l'esperienza utente (es. fiducia nelle auto automatizzate, efficacia nella collaborazione), tuttavia a lungo termine possono far peggiorare l'esperienza utente: alcuni aspetti anche piccoli possono indurre reazioni negative (es. stereotipi sui chatterbot), ed è importante fare attenzione all'effetto *Uncanny Valley*, dove si prova disagio di fronte a una somiglianza *quasi* umana.

In sintesi: gli Smart Tools sono più "intelligenti" ma meno prevedibili degli oggetti tradizionali, con un modello mentale opaco, richiedono principi di Interaction Design e nuovi principi per gestire l'AI. Gli Artificial Companions sono quasi umani ma non del tutto, incoraggiano l'attribuzione sociale e possono cadere nella Uncanny Valley; richiedono principi diversi.



Rischi, benefici, e User Tolerance

Le **sfide e i rischi** dell'AI includono errori di rilevanza (es. suggerimenti inappropriati rispetto al contesto) e la gestione di input simili da utenti diversi e con diverse intenzioni.

La posta in gioco è variabile: oltre a piccoli errori fastidiosi, i sistemi AI possono anche causare seri problemi, rivelare informazioni private, esporre a contenuti offensivi. Possono anche **violare le linee guida tradizionali** di usabilità, come la consistenza e la prevenzione degli errori: molte componenti AI sono intrinsecamente inconsistenti (risposte diverse allo stesso input nel tempo, comportamento diverso in base all'utente a causa della personalizzazione).

L'Intelligenza Artificiale (AI) fa riferimento a sistemi che dimostrano comportamento intelligente analizzando l'ambiente (RECOGNITION) e compiendo azioni (PREDICTION) - con un certo grado di autonomia - per raggiungere obiettivi specifici.





Da qui definiamo i concetti di precisione e recall: un'elevata **precisione** significa meno false positive ma potenzialmente meno true positive → *evita i falsi allarmi*; mentre un elevato **recall** significa classificare tutti i true positive ma potenzialmente anche qualche false positive → *non perde nessun true positive*. La scelta se ottimizzare una o l'altra dipende dall'applicazione specifica e dal contesto.

Per progettare sistemi AI interattivi:

- Concentrare la valutazione sulle prestazioni e la soddisfazione umana.
- Stabilire se "fare AI o non fare AI" in base a se è essenziale per risolvere i bisogni dell'utente.
- Capire quando automatizzare (sostituire l'utente) e quando aumentare (potenziare le capacità).
- Bilanciare l'incertezza dei sistemi AI con aspettative adeguate e feedback.

La **User Tolerance** verso le funzionalità AI dipende dal loro ruolo:

- **Critico o Complementare**: se il sistema funziona anche senza la feature AI, è complementare. Le feature critiche richiedono risultati accurati e affidabili; le feature complementari possono tollerare risultati meno perfetti.

		Recognition/Prediction	
		Positive	Negative
Reference	Positive	 True Positive	 False Negative
	Negative	 False Positive	 True Negative

- **Proattivo o Reattivo:** proattivo fornisce risultati senza richiesta (potenzialmente fastidioso se di bassa qualità); reattivo aiuta nel compito corrente. Le feature proattive sono utili in piccole quantità, al momento giusto, e se facili da scartare.
- **Visibile o Invisibile:** con feature visibili, l'utente valuta l'affidabilità visualizzando i risultati; per feature invisibili è più difficile comunicare l'affidabilità o ricevere feedback.
- **Dinamico o Statico:** dinamico migliora con l'interazione dell'utente (es. riconoscimento facciale per sblocco); statico migliora con aggiornamenti di sistema (es. riconoscimento facciale in galleria foto). Le feature dinamiche spesso incorporano calibrazione e feedback, le statiche meno: ciò influisce sulla percezione di affidabilità, sicurezza e fiducia.

Scegliere il percorso giusto: People+AI guidellines



By Microsoft Research: guidelines for human-AI interaction