

Interaction Design: progettazione dell'interazione che avviene tra esseri umani e oggetti, il suo obiettivo è quello di rendere le macchine, i servizi e i sistemi usabili dagli utenti per cui sono stati realizzati. Gli sviluppatori devono mettere al centro del processo di sviluppo i bisogni degli utenti arrivando a realizzare un prodotto maggiormente usabile. L'obiettivo dell'interazione uomo-macchina (HMI) e dell'interazione interazione uomo-computer (HCI) è quello di rendere possibile e facilitare al massimo l'uso e l'interazione con i sistemi IT (information technology). L'interaction design è composto da:

- 1) **Design di prodotto:** all'interno di questa disciplina si progettano dei beni/servizi il cui obiettivo è quello di migliorare e rendere più semplice la vita degli utilizzatori. Il nuovo oggetto progettato dal designer permette di svolgere azioni che fino a quel momento non potevano essere svolte o richiedevano un procedimento oneroso (esempio ->spotify)
- 2) **Design dell'esperienza utente:** ha lo scopo di aumentare la soddisfazione e la fedeltà del cliente migliorando l'usabilità dell'oggetto e il piacere fornito nell'interazione tra utente e prodotto. Per quanto possibile lo scopo dello UX designer è quello di far vivere all'utente la migliore esperienza possibili, in realtà è impossibile progettare l'esperienza utente in quanto ogni utente è differente dall'altro, per questa ragione si dice che si progetta PER l'esperienza utente. Per avere una UX ottimale si utilizza la metodologia HCD.
- 3) **Design dell'interfaccia:** il suo scopo è quello di migliorare la comprensione dell'utente di ciò che si può fare con l'oggetto. L'obiettivo dello UI designer è quello di progettare l'aspetto estetico e la struttura dell'interfaccia con il quale l'utente deve interagire.

Human Centered design (HCD): lo human centered design o HCD, è una metodologia di progettazione che adatta la progettazione in base ai bisogni, dalle capacità e dai comportamenti umani. L'obiettivo dell'HCD è quello di produrre sistemi utili, usabili e che si focalizzano sull'utente, infatti mette i bisogni, comportamenti e capacità umane prima di tutto e progetta i sistemi in funzione di esse, per questo motivo il processo HCD parte dall'osservazione dell'utente, per poi arrivare solo dopo all'identificazione della tecnologia necessaria. Nella forma di pensiero HDC le versioni alpha e beta (test) delle applicazioni non servono solo a fare debugging del codice, ma servono anche a capire cosa fanno e come si comportano gli utenti durante l'utilizzo del software (aiuta a capire quali sono le funzionalità più usate e quali non necessitano di essere implementate). Un processo HCD può essere schematizzato come un flusso dei passi seguenti:

- **Specificare il contesto d'uso:** identificare gli utenti che utilizzeranno il prodotto, per cosa lo utilizzeranno e sotto quali condizioni e vincoli.
- **Specificare i Requirements:** Identificare gli obiettivi utente che devono essere raggiunti tramite l'utilizzo del software.
- **Progettare la soluzione:** questa fase può essere a sua volta divisa in sottofasi iterative. Bozza->Prototipo->Soluzione
- **Testare e valutare:** la fase di testing è fondamentale per iniziare il ciclo sulla base dei risultati ottenuti da quest'ultimi e procedere ad un eventuale miglioramento.

Usabilità: secondo la norma ISO per usabilità si intende: grado in cui un prodotto può essere usato da particolari utenti per raggiungere certi obiettivi con efficacia e soddisfazione in uno specifico contesto. Il tema dell'usabilità e del design antropocentrico si applicano, quindi, a tutto il contesto dello sviluppo di un prodotto. Ogni software destinato all'utilizzo da parte di utenti necessita che se ne studi la relativa usabilità e che intorno a questa si progetti il sistema (seguendo quindi la metodologia proposta da HCD).

Progettazione delle interfacce: come definito nel design dell'esperienza utente, la UX non è possibile progettarela in quanto ogni utente sia diverso dall'altro, ma si può progettare un sistema in grado di poter guidare l'utente in modo ottimale. Ci sono due proprietà che risultano essere fondamentali per qualsiasi progetto utilizzato da utenti umani:

- **Discoverability:** è la capacità di un sistema di veicolare e comunicare i propri possibili usi all'utente. Un'applicazione che a prima vista fa capire all'utente quali sono le sue funzionalità ha una buona discoverability. Per garantire questo effetto si usa la visibilità.
- **Understanding:** è la capacità del prodotto di farsi usare correttamente dall'utente.

Discoverability -> quanto bene si capisce cosa si può fare con il prodotto.

Understanding -> quanto bene un prodotto comunica come si usano le funzioni disponibili.

Principi Fondamentali dell'interazione: come detto in precedenza la discoverability è possibile attraverso la visibilità, quest'ultima è in grado di facilitare il modo in cui un utente scopre cosa può fare un prodotto, come funziona e che tipo di azioni è possibile compiere. La visibilità si ottiene attraverso il processo di design dell'interazione ed è il risultato dell'applicazione di cinque concetti:

- 1) **Affordance:** indica la relazione fra le proprietà di un oggetto e le proprietà del suo utilizzatore, le affordance sono quindi relazione prodotto-utente che, se stabilite, abilitano o disabilitano specifiche modalità di interazione fra le parti. Essendo la affordance una relazione dipende sia dalle proprietà dell'oggetto che dell'utente. Oltre alle affordance esistono anche le **anti-affordance**, ovvero delle relazioni che se stabilite vanno a negare alcune proprietà/modi di interazione disponibili fra le due parti. Le affordance e le anti-affordance per abilitare o disabilitare una particolare interazione fra oggetto e utente devono essere percepibili o **discoverable**. Se un'affordance non è percepibile c'è bisogno di aumentarne la visibilità e quindi la percezione da parte dell'utente, per fare questo si usano i significanti.
- 2) **Significanti:** un significante è un modo per indicare dove effettuare un'azione, data una affordance che determina quali azioni sono possibili. I significanti possono essere: **Voluti o intenzionali** (come un'etichetta, una stringa o un'icona) oppure **accidentali o non intenzionali** (come ad esempio un sentiero). In sostanza i significanti comunicano come usare il prodotto. L'associazione tra affordance e significanti avviene tramite convenzioni.
- 3) **Mapping:** consiste nel posizionamento corretto dei significanti, tale disposizione se corretta può comunicare di più circa l'interfaccia e le sue funzionalità. Il modo migliore per fare mapping è quello naturale.
- 4) **Feedback:** il feedback è la comunicazione del risultato di un'azione, è una risposta che l'interfaccia dà all'utente in seguito al compimento di un'azione. Il feedback deve essere immediato, informativo (deve far capire che un'azione è in corso o che è stato prodotto il risultato che ci si aspetta) e semplice/poco invasivo (troppi annunci portano le persone ad ignorarli).
- 5) **Vincoli:** i vincoli limitano l'insieme delle azioni possibili, quest'ultimi permettono alle persone di decidere la giusta azione da compiere anche in situazioni del tutto nuove. Si dividono in 4 categorie:
 - 1) **Vincoli fisici:** si affidano a proprietà fisiche (pezzi che si incastrano solo in un determinato verso).
 - 2) **vincoli culturali:** si affidano alle abitudini culturali, sociali e comportamentali dell'utilizzatore.
 - 3) **vincoli semantici:** si affidano al significato della situazione per limitare l'insieme delle azioni possibili.
 - 4) **vincoli logici:** si affidano alla logica umana (se avanza un solo pezzo, so dove va collocato per esclusione).

I vincoli possono essere usati per veicolare l'utente verso un modello che si avvicini il più possibile a quello desiderato dal progettista. Vincoli e mapping si confondono tra loro: una serie di interruttori mappati in modo opportuno infondono un vincolo logico che permette all'utente di non sbagliare, si intuisce cosa verrà azionato da quell'interruttore posto in quel determinato punto. **Mapping forti possono diventare vincoli logici.**

Esiste poi un sesto principio non legato agli elementi dell'interfaccia, il **modello concettuale del sistema:** un modello concettuale è una descrizione semplificata delle funzionalità di un sistema, non deve essere completa ma utile (le icone a schermo del computer aiutano le persone a crearsi un modello concettuale dei dati in memoria, esse sono solo concettualizzazioni ideate per facilitarne l'uso. Questo esempio cade

quando si parla di memorizzazione tramite cloud). Il modello concettuale esprime come il designer vuole che l'utente percepisca il prodotto (si tratta della progettazione della UX), una volta pensato il modello concettuale si implementa l'interfaccia, in modo che il modello concettuale venga veicolato all'utente tramite affordance, significanti e mapping. Un modello mentale è un modello concettuale nella mente dell'utente che rappresenta il modo in cui, secondo lui, funziona il sistema. Più è grande la differenza tra il modello mentale e quello concettuale, più l'utente farà fatica ad usare il sistema. L'ideale è che l'utente apprenda un modello concettuale giusto direttamente dal dispositivo che utilizza (senza andare a leggere manuali o che venga trasmesso da terzi l'utilizzo).

Immagine di sistema: il modello concettuale degli utilizzatori si forma attraverso tutta l'informazione accessibile: aspetto del dispositivo, esperienze passate, manuali. L'insieme di tutta questa informazione è l'immagine di sistema. L'immagine di sistema è quindi tutto ciò che si percepisce dalla struttura fisica prodotta, completa di documentazione, istruzioni, significanti e ogni informazione accessibile dagli utilizzatori. Il modello concettuale dell'utente deriva dall'immagine di sistema, mediante l'interazione con il prodotto. Il progettista si aspetta che il modello concettuale dell'utente coincida con il suo, ma, non essendoci comunicazione diretta fra lui e l'utente, tutto il peso della comunicazione grava sull'immagine di sistema. In sostanza quello che più conta nella progettazione è la presentazione di un modello concettuale quest'ultima capace di rendere i prodotti comprensibili, di facile uso e gradevoli.

Funzioni Obbliganti: sono una forma di vincolo fisico e consistono in situazioni in cui le azioni sono vincolate in modo che un passaggio mancato impedisca il procedimento successivo. Esempi di funzioni obbliganti sono:

Interlock: obbliga a seguire una serie di operazioni nella sequenza dovuta prima di avviare l'azione richiesta.

lock-in: mantiene attiva una funzione impedendo che qualcuno la interrompa prematuramente (uscita da un'applicazione prevenuta da un alert che chiede la conferma).

lock-out: impedisce l'ingresso in uno spazio pericoloso o impedisce che succeda qualcosa.

Golfi: quando le persone usano un oggetto, si trovano davanti due golfi:

- 1) **Golfo dell'esecuzione:** si cerca di capire come funziona l'oggetto.

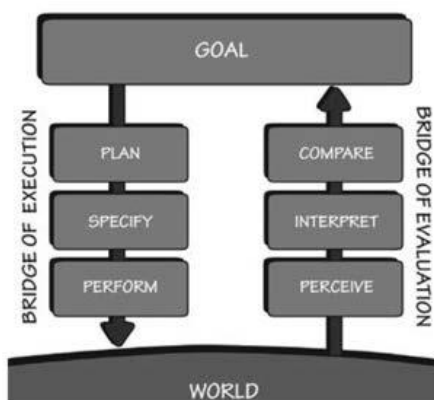
Come si supera -> tramite l'utilizzo dei significanti, vincoli, un buon mapping e un modello concettuale adeguato.

- 2) **Golfo della valutazione:** si cerca di capire cosa sia successo. Riflette lo sforzo necessario per interpretare lo stato fisico del dispositivo, il golfo risulta quindi minimo quando il dispositivo fornisce informazioni sul proprio stato in una forma facile da interpretare.

Come si supera -> utilizzo di feedback e un modello concettuale adeguato.

il compito del progettista è quello di aiutare gli utenti a colmare i due golfi e superarli. Se l'utente trova delle difficoltà nell'interazione con il sistema, tali difficoltà non hanno origine nell'utente ma nel design.

I sette stadi dell'azione:

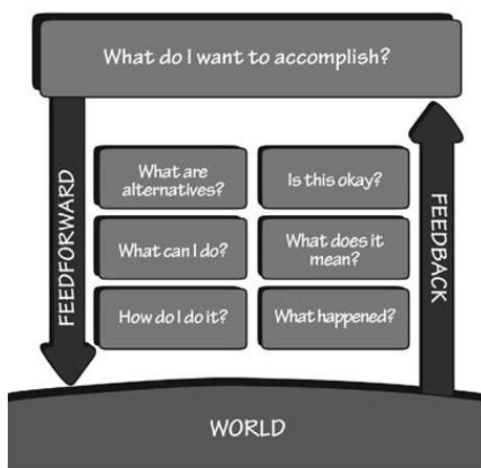


costituiti da un piano per lo scopo, tre per l'esecuzione e tre per la valutazione:

- 1) **Scopo:** definire l'obiettivo.
- 2) **Progettare:** definire l'azione da eseguire.
- 3) **Specificare:** costruire una sequenza d'azione.
- 4) **Eseguire:** eseguire la sequenza specificata.
- 5) **Percepire:** osservare lo stato del mondo.
- 6) **Interpretare:** elaborare la percezione.
- 7) **Confrontare:** rapportare il risultato allo scopo.

La maggior parte delle azioni non richiede di percorrere tutte le fasi in sequenza e la maggior parte delle attività non si risolve in una singola azione. Il modello a sette stadi del ciclo d'azione è uno strumento di progettazione che offre uno schema per sviluppare nuovi prodotti o servizi. Possiamo ricavare sette domande alle quali dovrebbe rispondere chiunque stia usando un determinato prodotto.

- 1) Cosa voglio ottenere?
- 2) Quali sono le sequenze d'azione alternative?
- 3) Quale azione posso fare ora?
- 4) Come faccio questa azione?
- 5) Cosa è successo?
- 6) Cosa significa?
- 7) Ho realizzato il mio scopo?



feed-forward: l'informazione che aiuta a rispondere alle domande sull'esecuzione è definita come feed-forward. Si ottiene mediante l'uso di significanti, vincoli e mapping

feed-back: l'informazione che aiuta a capire cosa è successo. Si ottiene mediante informazioni esplicite sull'impatto dell'azione.

Entrambi i tipi di informazione devono presentarsi in una forma facilmente interpretabile da chi utilizza il sistema e deve essere quindi immediatamente comprensibile.

sette principi fondamentali del design:

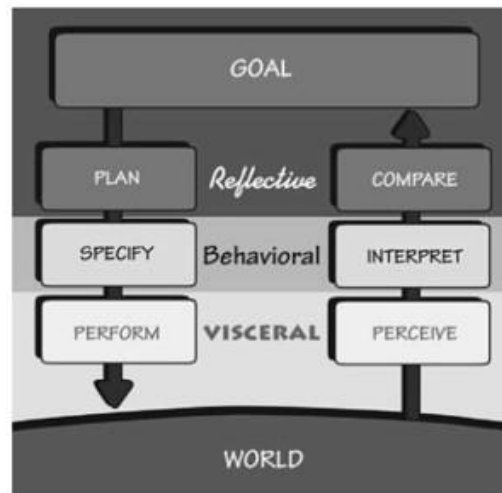
- 1) **Discoverability:** è bene che sia facile scoprire quali azioni sono possibili e qual è lo stato del dispositivo.
- 2) **Feedback:** è opportuno che ci sia un'informazione completa e continua riguardo ai risultati delle azioni.
- 3) **Modello concettuale:** il design dovrebbe fornire tutta l'informazione necessaria per creare un buon modello concettuale del sistema che favorisca la comprensione e la sensazione di controllo da parte dell'utente. Il modello concettuale aiuta la discoverability e la valutazione dei risultati.
- 4) **Affordance:** è bene che le affordance siano impostate in modo da rendere facili le azioni desiderate dal progettista e impossibili quelle indesiderate.
- 5) **Significant:** un uso appropriato dei significanti aumenta la discoverability e garantisce un feedback comprensibile.
- 6) **Mapping:** è necessario che la relazione fra comandi e azioni sia gestita da una buona disposizione spaziale.
- 7) **Vincoli:** è necessario guidare l'azione e facilitarne l'interpretazione attraverso l'uso dei vincoli.

N.B: le azioni opportunistiche sono quelle in cui il comportamento delle circostanze prevale sulla pianificazione. È difficile fare buon design per queste situazioni, anche attenendosi a tutti i principi esposti fino ad ora.

Pensiero subconscio: una volta che le abilità sono state apprese, le prestazioni sembrano senza sforzo ed eseguite automaticamente con poca consapevolezza. Il pensiero subconscio è quindi veloce, automatico e senza sforzo.

Pensiero conscio: l'attenzione cosciente è necessaria per apprendere la maggior parte delle cose. Il pensiero conscio è lento e laborioso ed è utile nel momento di riflessione sulle decisioni.

I tre livelli di processing: una semplificazione dell'elaborazione del cervello si basa sulla divisione di quest'ultimo in tre livelli procedurali:



- 1) **Visceral level (lizard brain):** da rapidi giudizi sull'ambiente (buono o cattivo, sicuro o pericoloso) permette di rispondere inconsciamente, senza consapevolezza o controllo cosciente, quindi in modo rapido. Per i progettisti, la risposta viscerale riguarda la percezione immediata, questo non ha nulla a che fare con quanto sia utilizzabile, efficace o comprensibile il prodotto si tratta di attrazione o repulsione. I designer usano la loro sensibilità estetica per guidare le risposte viscerali.
- 2) **Behavioral level:** è la sede delle abilità apprese, innescate da situazioni che corrispondono ai modelli appropriati. Le azioni e le analisi a questo livello sono in gran parte subconscie. Per i progettisti, l'aspetto più critico del livello comportamentale è che ogni azione è associata a un'aspettativa, le informazioni nel ciclo di feedback confermano o non confermano le aspettative, questo significa che il feedback è fondamentale per la gestione di quest'ultime.
- 3) **Reflective level:** il livello riflessivo è la sede della cognizione cosciente, quindi è qui che si sviluppa la comprensione e dove ha luogo il ragionamento/processo decisionale. I livelli viscerale e comportamentale sono subconsci e, di conseguenza, rispondono rapidamente ma senza analizzare molto. Per il designer, la riflessione è forse il più importante dei livelli di elaborazione perché le emozioni prodotte a questo livello sono durature nel tempo.

Errore umano: il fatto che le persone commettano degli errori è attribuibile alla mal progettazione e al cattivo design. Per evitare di incorrere nuovamente nell'errore, quando quest'ultimo viene commesso, è bene studiarne le causa (Root cause analysis), per poi ridisegnare la procedura o l'intero sistema in modo tale che quest'ultimo non venga commessi nuovamente o, se dovesse ripetersi, che i danni vengano ridotti al minimo.

Se il sistema lascia sbagliare gli utenti -> mal progettato

Se il sistema induce all'errore -> mal progettato

Perché le persone sbagliano -> perché i progettisti si concentrano sui requisiti del sistema e delle macchine e non sui requisiti delle persone

Root cause analysis: consiste nell'indagare sull'incidente finché non si trova la singola causa che ne è l'origine, ovvero quel momento in cui effettivamente qualcuno ha preso decisioni errate o eseguito azioni errate. Lo studio della causa non si ferma a questo, cerca di capire anche da cosa è derivato lo sbaglio, perché è accaduto e cosa si può fare per prevenirlo -> **procedura dei 5 "perché"**: quando si cerca la ragione dello scaturirsi di un evento non ci si deve fermare dopo averne trovata una sola, bisogna indagare finché non si trovano le cause di fondo.

Errori: l'errore è definito come una deviazione del comportamento corretto o generalmente accettato come giusto o adeguato. È possibile dividere gli errori in due classi:

- **Slips (lapsus):** si verifica quando una persona intende fare un'azione e finisce per eseguirne un'altra (l'azione eseguita non è quella voluta). Ci sono due tipi di lapsus: **action-based** e **memory-lapses**. Nel primo caso si esegue un'azione sbagliata, nel secondo caso si dimentica di eseguire l'azione o di valutarne i risultati.

Esempio di action-based: versare il latte nel caffè e poi riporre la tazza in frigo (azione corretta ma applicata all'oggetto sbagliato).

Esempio di memory-lapses: dimenticare di spegnere il fornello dopo aver cucinato (dimentica di eseguire l'azione o di valutarne i risultati).

- **Mistakers (errori cognitivi):** si verifica quando viene stabilito l'obiettivo o lo scopo sbagliato. Da quel momento in poi, anche se le azioni sono corrette, fanno parte dell'errore. In questo tipo di errore l'azione è corretta, ma l'intenzione no. Gli errori cognitivi si dividono in:

regola sbagliata -> la persona ha diagnosticato adeguatamente la situazione, ma poi ha eseguito una linea di condotta errata (si diagnostica un difetto su un pezzo ma si decide di non sostituirlo perché ancora funzionante al 50%).

conoscenza sbagliata -> il problema viene diagnosticato erroneamente a causa di una conoscenza errata (il peso del carburante viene pesato in libbre in vece che in chilogrammi).

Dimenticanza -> si ha quando si dimentica qualche passaggio quando si fissano gli obiettivi (un meccanico non è riuscito a completare la risoluzione dei problemi a causa della distrazione). La differenza con la dimenticanza del memory-lapses è che questa avviene in fase di pianificazione dell'obiettivo.

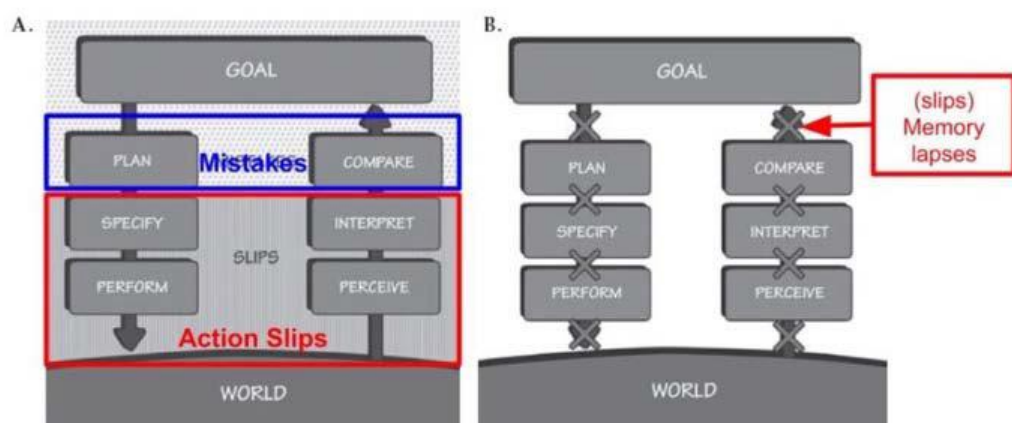


FIGURE 5.2. Where Slips and Mistakes Originate in the Action Cycle. Figure A shows that action slips come from the bottom four stages of the action cycle and mistakes from the top three stages. Memory lapses impact the transitions between stages (shown by the X's in Figure B). Memory lapses at the higher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

I lapsus -> sono tutti legati alla modalità subconscia (gli esperti hanno maggiore probabilità di commettere slips).

Gli errori cognitivi -> sono legati alla modalità conscia (è più probabile che i novizi commettano mistakes). La maggior parte degli errori deriva da una scelta errata dell'obiettivo o da valutazioni errate. Tutto ciò avviene a causa delle scarse informazioni fornite dal sistema.

Interruzioni e feedback: le interruzioni sono una delle principali fonti di errori (soprattutto errori di tipo lapsus), per riprendere il lavoro è infatti necessario ricordarsi precisamente lo stato precedente dell'attività, spesso questo è difficile. Un'altra causa di errori sono i feedback errati (fastidiosi o non necessari): la progettazione dei segnali di allarme è complessa e necessita di tenere conto dei compromessi per le persone che sono interessate. Alternativa ai segnali di allarme -> uso di interfacce vocali: consente di fornire informazioni precise senza bisogno di attenzione visiva, se l'ambiente è rumoroso o ci sono avvisi vocali multipli sono inefficaci.

Prevenzione degli errori: non dovrebbe essere possibile che un semplice errore provochi un danno diffuso. Metodi di prevenzione:

- 1) **Comprendere le cause dell'errore** e progettare per ridurre al minimo tali cause
- 2) Effettuare **Controlli di sensibilità** e chiedersi se le azioni sono "sensitive"
- 3) Rendere possibile **Annullare le operazioni** e rendere più difficili le operazioni che non possono essere annullate
- 4) Rendere più semplice **la scoperta/comprendimento degli errori**
- 5) **Corretto compimento dell'azione:** non trattare l'azione come un errore, ma aiutare l'utente a svolgerla correttamente.

la prevenzione spesso implica l'aggiunta di vincoli specifici alle azioni. All'interno dei dispositivi elettronici possono essere usati un'ampia gamma di metodi per la riduzione dell'errore, tra cui:

- **Segregare i controlli:** i controlli facilmente confondibili si devono trovare lontani l'uno dall'altro.
- **Utilizzo di moduli separati:** qualsiasi controllo non direttamente rilevante per l'operazione corrente non deve essere visibile, ma necessita di uno sforzo extra per essere raggiunto (progettazione di un ambiente modulare).
- **Utilizzo del comando "undo":** annullare le operazioni eseguite dal comando precedente dove possibile.

Molti sistemi cercano di prevenire gli errori richiedendo la conferma prima che un comando venga eseguito, questa scelta non è sempre sensata (i messaggi di avvertimento sono inefficaci), **il punto importante è rendere salienti le implicazioni dell'azione**, quindi cosa può fare il designer? -> rendere più evidente l'oggetto su cui si sta agendo e rendere le operazioni reversibili

I sistemi elettronici posseggono anche il vantaggio di verificare che l'operazione richiesta sia sensata.

Minimizzare i lapsus: i lapsus si verificano quando l'azione che viene eseguita è così ben appresa che può essere eseguita automaticamente, senza attenzione cosciente. La persona non presta quindi sufficiente attenzione all'azione e alle sue conseguenze. **Il modo migliore per mitigare i lapsus è fornire un feedback comprensibile sulla natura dell'azione che viene eseguita o rendere più dissimili possibile le azioni e i relativi controlli (anche collocandoli in spazi diversi).** Oltre all'uso dei feedback risulterebbe utile anche un meccanismo che possa consentire l'annullamento dell'errore. Il modo migliore per minimizzare i lapsus rimane la progettazione dei passaggi iniziali delle azioni in modo che questi siano più dissimili possibile.

Minimizzare gli errori: i modi migliori per la mitigazione degli errori cognitivi sono:

- Aggiunta di più controlli e passaggi di controllo.
- Ridurre il numero di punti critici di sicurezza in cui possono verificarsi gli errori.

Un buon sistema oltre che cercare di ridurre al minimo le possibilità di errore, mitiga anche le sue conseguenze.

Principi chiave di progettazione:

- 1) **Presentare la conoscenza necessaria:** presentare una buona immagine di sistema permette di ridurre tutti gli errori cognitivi legati alla coscienza e alle regole. Un modello concettuale chiaro e semplice, unito a una buona immagine di sistema, permette di facilitare l'apprendimento e minimizzare gli errori.
- 2) **Utilizzo dei vincoli naturali e artificiali**
- 3) **Collegare i due golfi (golfo di esecuzione e golfo di valutazione)** rendendo le cose visibili per la valutazione:
sul lato dell'esecuzione: fornire informazioni feedforward e quindi rendere le opzioni prontamente disponibili.
sul lato della valutazione: fornire informazioni feedback e quindi rendere visibili le i risultati di ciascuna operazione.

Interfacce utente: un'interfaccia utente è sempre composto da due parti. Una di queste parti appartiene alla persona, l'altra allo strumento. Lo strumento è ciò che compie l'azione, l'interfaccia è ciò che serve per permettere all'utente di guidare lo strumento nell'esecuzione dell'operazione. Quando si parla di user interface o UI, stimo parlando dello spazio di un sistema dove avviene l'interazione uomo-macchina, tale interazione deve essere resa più semplice ed efficiente possibile in modo tale da massimizzare la User Experience del prodotto/sistema/oggetto. Le macchine, al contrario dell'uomo, sono puramente deterministiche, l'interfaccia uomo-macchina va quindi ad avere un ruolo fondamentale dal momento che abilita la comunicazione fra due realtà aventi modalità di pensiero diametralmente opposte. Un dispositivo che implementa un HMI (interazione uomo-macchina) è chiamato HID (human Interface device).

Classificazione delle interfacce: gli umani posseggono 5 sensi, questo porta ad identificare cinque categorie di interfacce possibili, più una sesta che è legata al senso dell'orientamento:

- 1) **Tactile UI** (touch, tatto)
- 2) **Visual UI** (sight, vista)
- 3) **Auditory UI** (sound, udito)
- 4) **Olfactory UI** (smell, olfatto)
- 5) **Gustatory UI** (taste, gusto)
- 6) **Equilibrium UI** (balance, equilibrio)

la maggior parte delle interfacce utente utilizza più di un senso, infatti sono dette **CUI (composite User Interface)**.

Esempi di CUI:

- **GUI (Graphical User Interface)**, le quali sono composte da interfacce grafiche (visual) e tattili (tactile).
- Se ad una GUI aggiungiamo il suono otteniamo una **MUI (Multimedia User Interface)**.

Estendere un'interfaccia con più sensi non sempre è una buona idea (Facebook ha disabilitato di default l'audio dei video nella modalità dell'auto-riproduzione).

Categorizzazione delle CUI: le CUI possono essere categorizzate in tre diverse categorie:

- 1) **Standard:** usano dispositivi standard come tastiere, mouse e monitor.
- 2) **Virtual:** creano un mondo virtuale che funge da interfaccia fra utente e macchina.
- 3) **Augmented:** non bloccano la percezione del mondo reale all'utente, ma la vanno ad arricchire (l'interfaccia è un mix di contenuti reali e virtuali).

le CUI possono essere classificate anche tramite il numero di sensi che coinvolgono (smell-o-vision è una CUI standard, che coinvolge 3 sensi: visione, udito e olfatto), quando un'interfaccia utente interagisce con tutti i sensi umani viene chiamata **Qualia Interface**.

Human interface devices: HID è un tipo di dispositivo informatico solitamente utilizzato da esseri umani, riceve in input da quest'ultimi e fornisce in output delle informazioni.

HID software spacification: lo standard HID è stato adottato per consentire l'innovazione dei dispositivi di input per PC e per semplificare il processo di installazione di tali dispositivi.

Le innovazioni hardware hanno reso necessario lo sviluppo di un supporto software universale: tutti i dispositivi definiti da HID forniscono pacchetti auto-descrittivi utili per l'associazione dinamica di dati I/O con le funzionalità dell'applicazione. Il protocollo HID ha i suoi limiti, ma tutti i sistemi operativi moderni riconoscono i dispositivi USB HID standard senza bisogno di un driver specializzato.

Nel protocollo HID, ci sono due entità:

- 1) **Host:** comunica con il dispositivo e riceve i dati di input da quest'ultimo in base alle azioni eseguite dall'utente. I dati di output fluiscono dall'host al dispositivo.
- 2) **Dispositivo:** è l'entità che interagisce direttamente con l'essere umano, come una tastiera o un mouse.

il protocollo HID rende semplice anche l'implementazione dei dispositivi quest'ultimi definiscono i propri pacchetti di dati e presentano all'host un "descrittore HID". Il descrittore HID è un array di byte hardcoded che descrive i pacchetti di dati del dispositivo (quanti pacchetti supporta il dispositivo, la dimensione dei pacchetti, lo scopo di ogni byte e bit nel pacchetto). L'host prima di avviare completamente la comunicazione con il dispositivo, recupera il descrittore per analizzarlo. Grazie all'HID report protocol è possibile eseguire questa operazione.

Essendo che non tutti gli host sono in grado di analizzare i descrittori HID, HID definisce anche un boot protocol, all'interno del quale sono supportati solo dispositivi specifici (tastiera e mouse) che utilizzano formati di pacchetti di dati fissi (il descrittore HID non viene utilizzato in questa modalità), così facendo le funzionalità minime dei dispositivi sono ancora possibili su host che non supportano HID.

Altri protocolli che usano HID: HID attualmente è utilizzato anche in altri bus di comunicazione del computer, questo consente ai dispositivi HID tradizionalmente pensati come USB di essere utilizzati anche su bus alternativi. Il supporto esistente per i dispositivi USB HID può in genere essere adattato molto più velocemente rispetto alla necessità di inventare un protocollo completamente nuovo (motivo per il quale si dice che HID favorisce una rapida innovazione). I bus che utilizzano HID sono:

- Bluetooth HID
- Serial HID: utilizzato nei ricevitori di controllo remoto per PC windows media center
- Dispositivo di input ZigBee
- HID over I²C: utilizzato per dispositivi incorporati in windows 8
- HOGP: utilizzato per i dispositivi HID collegati tramite tecnologia Bluetooth a basso consumo energetico

Dispositivi HID: le periferiche HID possono essere organizzate in due categorie principali:

- 1) **Dispositivi di input:** basati su sensori (il cui scopo è rilevare forme di interazione fisica e convertirli in informazioni elettroniche, analogiche o digitali) **fisico -> elettronico**
- 2) **Dispositivi di output:** basati su attuatori (il cui scopo è convertire segnali elettronici, analogici o digitali in attuazione fisica) **elettronico -> fisico**

gli input e gli output HID sono suddivisi in classi in base al tipo di input/output utilizzato dall'HID (testi e caratteri, posizioni, suono, immagini, parametri ambientali, parametri fisiologici/sanitari/dio), spesso gli HID odierni utilizzano una tecnologia mista.

Keyboards: il più usato dei dispositivi HID. Un layout di tastiera è una disposizione fisica, visiva e funzionale dei tasti, delle legende e delle associazioni di significato dei tasti.

Layout fisico -> posizionamento effettivo dei tasti su una tastiera

Layout visivo -> è la disposizione delle legende (etichette e segni sui tasti)

Layout funzionale -> è la disposizione dell'associazione chiave-significato (mappatura della tastiera), determinata dal software, di tutti i tasti di una tastiera. Questo determina l'effettiva risposta della pressione di un tasto.

Le moderne tastiere sono progettate per mandare uno scan-code all'OS quando viene premuto/rilasciato un tasto, il SO a questo punto converte lo scan-code in uno specifico codice binario di caratteri utilizzando una tabella di conversione (tabella di mappatura della tastiera). Questo significa che una tastiera può essere mappata dinamicamente a qualsiasi layout senza cambiare i componenti hardware.

Multifunctional keyboards: forniscono funzioni aggiuntive alla tastiera standard. Le tastiere multifunzionali possono essere dotate di tastiere personalizzate e funzioni completamente programmabili (tastiere con display touchscreen).

Barcode readers: un lettore di codici a barre è uno scanner ottico in grado di leggere i codici a barre stampati, decodificare i dati contenuti nel codice a barre e inviare i dati a un computer. All'interno del barcode reader viene utilizzato un sensore di luce ottico che traduce gli impulsi ottici in segnali elettrici. I codici a barre sono un metodo per rappresentare i dati in un formato visivo leggibile dalla macchina e sono in grado di variare la rappresentazione dei dati tramite le larghezze e le distanze delle linee. A differenza dei codici a barre i codici QR utilizzano quattro modalità di codifica (numerica, alfanumerica, byte/binaria e kanji)

RFID: un tag RFID, quando viene attivato da un impulso di interrogazione elettromagnetico da un dispositivo lettore RFID, trasmette i dati digitali (di solito un numero di inventario identificativo). Viene generalmente usato per fare l'inventario delle merci. Esistono due tipi di tag RFID:

1) **Tag passivi:** sono alimentati dall'energia delle onde radio che interrogano il lettore RFID.

2) **Tag attivi:** sono alimentati dalla batteria e quindi possono essere letti da una distanza maggiore. a differenza di un codice a barre, il tag non deve trovarsi nella linea visiva del lettore.

NFC: è un insieme di protocolli per la comunicazione bidirezionale tra due dispositivi elettronici vicini. NFC offre una connessione a bassa velocità che può essere utilizzata per avviare connessioni wireless.

Pointing devices: un dispositivo di puntamento è un'interfaccia di input che consiste a un utente di inserire dati spaziali in un computer. I movimenti del dispositivo di puntamento vengono riprodotti sullo schermo dai movimenti del puntatore. Nell'ambito dei pointing devices è fondamentale la **legge di Fitts**: è un modello predittivo del movimento utilizzato nell'interazione uomo-macchina e permette di prevedere il tempo necessario per spostarsi rapidamente in un'area target. All'interno della progettazione questa legge va tenuta in considerazione: grandi obiettivi riducono il tempo e la frustrazione, inoltre i progettisti dovrebbero appianare e ridurre le traiettorie. La legge di Fitts utilizza il rapporto tra la distanza del target e la larghezza di quest'ultimo, versioni più recenti di questa legge prendono in considerazione anche altri parametri come ad esempio:

- Direzione di puntamento
- Peso del dispositivo
- Forma del bersaglio
- Posizione del braccio (riposo o mezz'aria)
- 2D e 3D
- Ambiente a gravità zero.

Classificazione di puntamento:

direct vs. indirect input:

- **Input diretto:** il puntatore sullo schermo si trova nella stessa posizione fisica del dispositivo di puntamento (penna di un tablet, dito sul touch screen). il puntamento diretto è più veloce ma meno preciso di quello indiretto.
- **Input indiretto:** non si trova nella stessa posizione fisica del puntatore, ma ne traduce i movimenti a schermo (mouse, joystick, tavoletta grafica).

absolute vs. relative movement:

- **Movimento assoluto:** fornisce una mappatura coerente tra un punto nello spazio di input (posizione del dispositivo) e un punto nello spazio di output (lo schermo). Un esempio è il dito sul touchscreen.

- **Movimento relativo:** mappa lo spostamento nello spazio di input nello stato di output. Controlla la posizione relativa del cursore rispetto alla sua posizione iniziale (mouse, joystick).

isotonic vs. elastic vs. isometric:

- **Dispositivo isotonic:** è mobile e misura il suo spostamento (mouse)
- **Dispositivo isometric:** è fisso e misura la forza che agisce su di esso (touchscreen a pressione, trackpoint)
- **Dispositivo elastico:** aumenta la sua resistenza alla forza con lo spostamento (joystick)

position control vs. rate control:

- **Dispositivo per il controllo della posizione:** modifica direttamente la posizione assoluta o relativa del puntatore sullo schermo (mouse, dito sul touchscreen).
- **Dispositivo per il controllo della velocità:** cambia la velocità e la direzione del movimento del puntatore sullo schermo (joystick).

N.B: si dovrebbe utilizzare un approccio multimodale (o cursori personalizzabili) per migliorare le prestazioni.

Eye tracking: L'eye tracking è il processo di misurazione del punto di sguardo. Un eye tracker è un dispositivo per misurare la posizione degli occhi e il movimento degli occhi. Esistono diversi metodi per misurare il movimento degli occhi, la variante più popolare utilizza immagini video da cui viene estratta la posizione dell'occhio. La luce, tipicamente infrarossa, viene riflessa dall'occhio e rilevata da una videocamera o da un altro sensore ottico appositamente progettato. Gli eye tracker basati su video utilizzano tipicamente il riflesso corneale e il centro della pupilla come caratteristiche da tenere traccia nel tempo. I design attuali più utilizzati sono gli eye-tracker basati su video. Una telecamera mette a fuoco uno o entrambi gli occhi e registra il movimento degli occhi. La maggior parte dei moderni eye-tracker utilizza il centro della pupilla e la luce non collimata a infrarossi/vicino infrarosso per creare riflessi corneali (CR).

Vengono utilizzati due tipi generali di tecniche di tracciamento oculare a infrarossi/vicino infrarosso:

- 1) **Pupilla luminosa**
- 2) **Pupilla scura**
- 3) **Luce passiva**

La loro differenza di basa sulla posizione della sorgente di illuminazione rispetto all'ottica, la luce (nel caso di fonte coassiale con il percorso ottico) si riflette sulla retina creando un effetto pupilla luminosa. Se la sorgente di illuminazione è sfalsata rispetto al percorso ottico, la pupilla appare scura. Il tracciamento della pupilla luminosa crea un maggiore contrasto iride/pupilla, consentendo un tracciamento oculare più robusto che riduce notevolmente le interferenze causate da altre caratteristiche oscuranti. Questo tipo di monitoraggio è più affidabile ma richiede una configurazione hardware più complessa.

Gli eye-tracker misurano la rotazione dell'occhio rispetto a un quadro di riferimento. Questo aspetto di solito è legato al sistema di misurazione.

- **Se il sistema di misurazione è montato sulla testa,** vengono misurati gli angoli dell'occhio nella testa, in questi casi, la direzione della testa viene aggiunta alla direzione dell'occhio nella testa per determinare la direzione dello sguardo.
- **Se il sistema di misurazione è montato sul tavolo,** gli angolo di sguardo vengono misurati direttamente in coordinate mondiali. Tipicamente, in queste situazioni i movimenti della testa sono proibiti.

Data glove: Un guanto cablato (chiamato anche "dataglove") è un dispositivo di input per l'interazione uomo-computer. Diverse tecnologie di sensori vengono utilizzati all'interno di questi dispositivi per acquisire dati fisici come la flessione delle dita oppure rilevare dati globali come la posizione/rotazione del guanto. Tramite l'utilizzo di tutti questi sensori ogni movimento che viene interpretato dal software del guanto può essere attribuito ad una qualsiasi azione. I gesti possono quindi essere classificati in informazioni utili (riconoscimento del linguaggio dei segni).

Haptic devices: i dispositivi aptici (o interfacce tattili) sono dispositivi meccanici che mediano la comunicazione tra utente e computer, tali dispositivi consentono agli utenti di **toccare, sentire e manipolare oggetti tridimensionali in ambienti virtuali**. Mouse e joystick sono dispositivi di solo input, il che significa che tengono traccia delle manipolazioni fisiche dell'utente, ma non forniscono feedback manuale. **I dispositivi aptici, invece, sono dispositivi di input-output**, il che significa che tracciano le manipolazioni fisiche di un utente (input) e forniscono sensazioni tattili realistiche coordinate con gli eventi mostrati a schermo (output).

Smart paper, whiteboards and similars: questi dispositivi digitalizzano la scrittura dell'utente mediante penne intelligenti tracciate e/o superfici sensorizzate. Questa è un'interfaccia tra categorie in cui il puntamento e le immagini vengono utilizzati come input combinati.

Speech and auditory input interfaces: il suono può essere acquisito tramite l'uso dei microfoni. **Un microfono è un dispositivo/sensore che converte il suono in un segnale elettrico** (i microfoni sono usati anche come dispositivi HID). Oggi vengono utilizzati diversi tipi di microfono, i quali impiegano metodi diversi per convertire le variazioni di pressione dell'aria di un'onda sonora in un segnale elettrico.

Array di microfoni: **è un numero qualsiasi di microfono che operano in contemporanea** sono generalmente usati per:

- Sistemi per estrarre l'input vocale dal rumore ambientale.
- Suono surround.
- Registrazione binaurale.
- Localizzazioni di sorgenti acustiche.
- Registrazioni ad alta fedeltà.
- Monitoraggio del rumore ambientale.

Tipicamente un array è costituito da microfoni omnidirezionali (che lavorano su più direzioni), microfoni direzionali o un mix di entrambi. Tutti i microfoni sono collegati a un computer che registra ed interpreta i risultati in una forma coerente. Gli array possono anche essere formati utilizzando microfoni molto vicini tra loro. **L'elaborazione simultanea dei segnali proveniente da ciascuno dei singoli elementi del vettore di microfoni può creare uno o più microfoni "virtuali".**

Le applicazioni pratiche non consentono una discussione in formato libero con i computer, i principali obiettivi di progettazione sono:

- Basso carico cognitivo
- Bassi tassi di errore
- Esperienza utente naturale

Problemi/limitazioni:

- La larghezza della banda è inferiore rispetto ai display visivi
- Natura effimera del discorso
- Difficoltà nell'analisi/ricerca -> carico computazionale più elevato

Questo tipo di applicazioni hanno successo con vocabolari specializzati:

- Dettatura di rapporti/appunti/lettere
- Pratica delle abilità comunicative (paziente virtuale)
- Recupero/trascrizione automatica di contenuti audio
- Sicurezza/ID utente

Image sensors: **un sensore di immagine è un sensore che rileva e trasmette le informazioni utilizzate per creare un'immagine, lo fa convertendo l'attenuazione variabile delle onde luminose in segnali che trasmettono le informazioni.** Questo tipo di sensori vengono utilizzati nei dispositivi che includono fotocamere digitali.

I due tipi principali di sensori di immagine elettronici sono:

- 1) **Dispositivo ad accoppiamento di carica (CCD):** basati su condensatori MOS
- 2) **Sensore a pixel attivi (sensore CMOS):** basati su amplificatori MOSFET (MOS a effetto di campo transistori).

Entrambi i sensori sono basati sulla tecnologia MOSF (metal-Oxide-Semiconductor)

3D image capture device: la scansione 3D è il processo di analisi di un oggetto o un ambiente del mondo reale per raccogliere dati sulla sua forma e il suo aspetto. I dati raccolti possono anche essere utilizzati per costruire modelli 3D digitali, utilizzati in applicazioni come la realtà aumentata, il motion capture e il riconoscimento dei gesti.

Uno scanner 3D può basarsi su molte tecnologie diverse, ciascuna con i propri limiti, vantaggi e costi. Ancora oggi la scansione 3D presenta delle limitazioni forti, ad esempio sulla digitalizzazione di oggetti lucidi, riflettenti o trasparenti.

La tecnologia di scansione 3D possono suddividersi in due categorie:

- 1) **Passive scanning:** si basano sul rilevamento della radiazione ambientale riflessa, sono dispositivi che possono usare tecnologia come gli infrarossi e possono essere molto economici, nella maggior parte dei casi, infatti, non richiedono hardware particolare ma semplici fotocamere.
- 2) **Active scanning:** a differenza degli scanner passivi, gli scanner attivi emettono un qualche tipo di radiazione/luce e rilevano il suo riflesso o la radiazione che passa attraverso l'oggetto da analizzare.

Esempi di passive scanning:

- **Sistemi stereoscopici:** le telecamere stereoscopiche sono i più comuni sistemi di scansione 3D passiva. Di solito questo tipo di sistemi impiegano due videocamere leggermente separate che guardano la stessa scena, analizzando le differenze tra le immagini di ciascuna telecamera è possibile determinare la distanza di ogni punto delle immagini. Questo sistema funziona nello stesso modo in cui funziona la visione stereoscopica umana.
- **Sistemi fotometrici:** utilizzano una sola fotocamera, ma acquisiscono più immagini in condizioni di luce diverse. Tramite le diverse immagini sarà possibile ricostruire l'oggetto in 3D in base ai riflessi presenti in ognuna di esse.
- **Tecniche di silhouette:** utilizzano contorni creati da una sequenza di fotografie ricavate girando attorno ad un oggetto tridimensionale. Le sagome vengono poi estruse e intersecate per formare l'approssimazione visiva dell'oggetto. Con questo tipo di approcci i dettagli concavi di un oggetto non possono essere rilevati.

Esempi di active scanning:

- **Time-of-flight:** è uno scanner attivo che utilizza la luce laser per sondare il soggetto. Il telemetro laser trova la distanza di una superficie cronometrando il tempo di andata e ritorno di un impulso di luce che viene riflesso dall'oggetto e catturato da un rilevatore. Il telemetro rileva solo la distanza di un punto nella sua direzione di vista, pertanto lo scanner scansiona il suo intero campo visivo un punto alla volta cambiando la direzione del telemetro.
- **Triangolazione:** gli scanner basati sulla triangolazione sono scanner attivi che utilizzano la luce laser per sondare l'ambiente. Rispetto al time-of-flight, il laser di triangolazione fa brillare un laser sul soggetto e utilizza una telecamera per cercare la posizione del punto laser. A seconda della distanza con la quale il laser colpisce la superficie, il punto laser appare in punti diversi del capo visivo della telecamera.
- **Structured-light:** proiettano un modello di luce sul soggetto e osservano la deformazione del modello sul soggetto. Una telecamera, leggermente distante dal proiettore di pattern, osserva la forma del pattern e calcola la distanza di ogni punto del campo visivo. Il vantaggio di questi tipi di dispositivi è la velocità e la precisione (non scansano un punto alla volta ma l'intero campo visivo).
- **Luce modulata:** illuminano il soggetto con una luce che cambia continuamente (con un andamento sinusoidale). Una telecamera rileva la luce riflessa e la quantità di spostamento del pattern determina la distanza percorsa dalla luce. Questo tipo di dispositivo consente di ignorare la luce proveniente da sorgenti diverse da un laser, quindi non ci sono interferenze.

Microsoft Kinect: questo dispositivo incorpora una telecamera RGB, proiettori a infrarossi (luce strutturata) e rilevatori che mappano la profondità attraverso la luce strutturata, oltre a questi dispositivi è presente anche un array di microfoni, e un software di intelligenza artificiale che consentono al dispositivo di eseguire il riconoscimento dei gesti in tempo reale, il riconoscimento vocale e il rilevamento dello scheletro del corpo.

IMU inertial Measurement Unit: una funzionalità essenziale del telecomando Wii è la capacità di rilevare il movimento, ciò consente all'utente di interagire e manipolare gli elementi sullo schermo tramite riconoscimento e puntamento dei gesti, utilizzando l'IMU e la tecnologia dei sensori ottici. Un'unità di misura inerziale (IMU) è un dispositivo elettronico che misura e segnala la forza specifica di un corpo, la velocità angolare e talvolta l'orientamento del corpo, utilizzando una combinazione di accelerometri, giroscopi e magnetometri.

Wearable devices and User interfaces: in un wearable device l'interfaccia è l'unità di calcolo sono unite insieme. I dispositivi indossabili possono essere per uso generale o per scopi specializzati come i fitness tracker quest'ultimi possono incorporare sensori speciali come accelerometri, termometri e cardiofrequenzimetri.

Molti computer indossabili sono sempre attivi, atti a elaborare/registrazione dati in modo continuo tramite dispositivi come:

cardiofrequenzimetro (HRM): ovvero un dispositivo di monitoraggio personale che consente di misurare/visualizzare la frequenza cardiaca in tempo reale o di registrare la frequenza cardiaca per uno studio successivo. Questo tipo di dispositivi usa principi di misurazione elettrica e/o ottica e non può essere utilizzato per scopi medici. Entrambi i tipi di segnali (elettrici/ottici) possono fornire gli stessi dati di base sulla frequenza cardiaca:

- **i sensori ECG:** misurano il bio-potenziale generato dai segnali elettrici che controllano l'espansione e la contrazione del cuore.
- **i sensori PPG:** utilizzano una tecnologia basata sulla luce per misurare il volume del sangue. Alcuni dispositivi che utilizzano questa tecnologia sono in grado di misurare anche la saturazione di ossigeno nel sangue.

luce verde: viene proiettata sulla pelle fino a raggiungere i vasi sanguigni, parte di questa luce viene riflessa e catturata da un photo-detector del dispositivo. L'intensità della luce riflessa permette di misurare la quantità di sangue presente nei vasi sanguigni. Meno accurata e più limitata ma resiliente al movimento del dispositivo.

luce rossa: fondamentalmente funziona in modo analogo alla luce verde, con la differenza che con quest'ultima è possibile eseguire misurazioni a profondità maggiore. Penetrando a una maggiore profondità dei tessuti è più accurata della luce verde non risulta però resiliente al movimento, questo significa che il movimento del dispositivo potrebbe causare inaccuratezze.

Elettroencefalografia (EEG): è un metodo di monitoraggio per registrare l'attività elettrica del cervello. Gli EEG headset indossabili posizionano elettrodi non invasivi e rilevano/registrazione l'attività elettrica nel cervello. I segnali raccolti vengono amplificati e digitalizzati, quindi inviati al computer o dispositivo mobile per l'archiviazione e l'elaborazione dei dati. Gli EEG headset sono utilizzati come dispositivi di puntamento per soggetti con disabilità.

Natural User Interfaces: sono interfacce utente in cui l'interazione è diretta è coerente con il comportamento "naturale" dell'uomo. Sono effettivamente invisibili e rimangono tali poiché l'utente apprende continuamente interazione sempre più complesse. Sebbene l'interfaccia richieda apprendimento, tale apprendimento è facilitato dal design che dà all'utente la sensazione di avere successo immediato e continuo. Pertanto, l'aggettivo "naturale" si riferisce all'interazione che ha l'utente con la tecnologia che avviene quasi come se non ci fosse un reale sforzo. Le NUI sfruttano le abilità che l'uomo ha acquisito durante la vita, questo aspetto permette di ridurre al minimo il carico cognitivo e la distrazione.

N.B: nessuna interfaccia potrà mai essere naturale in tutti i contesti (swipe verso destra/sinistra -> gesto intuitivo, interazione con più dita -> gesto meno intuitivo)

Linee guida per la progettazione di NUI:

- competenza immediata
- apprendimento progressivo
- interazione diretta
- basso carico cognitivo (utilizzo di abilità innate e semplici)

una NUI è un'interfaccia che sembra naturale da usare perché si adatta alle competenze e al contesto dell'utente.

- Una NUI dovrebbe trarre vantaggio dalle competenze e dalle conoscenze di base/esistenti degli utenti
- Una NUI dovrebbe avere un chiaro percorso di apprendimento e consentire a tutte le fasce di utente (esperti e non) di interagire in modo naturale.

Reality based interfaces:

- Una strategia per la progettazione di NUI è l'uso di una "reality Based Interface" (RBI). Un esempio di RBI consiste nell'utilizzare un dispositivo indossabile per fare in modo che chi lo indossa possa fare clic su qualsiasi oggetto quotidiano in modo da farlo funzionare come un collegamento ipertestuale. Le RBI sono spesso confuse con le NUI, quando in realtà sono solo un mezzo per ottenerle.
- Un altro esempio di una strategia per progettare una NUI non basata su RBI consiste in una limitazione di funzionalità e personalizzazione in modo che gli utenti abbiano ben poco da imparare sul funzionamento di un dispositivo. Supponendo che le funzionalità fornite possano di base soddisfare gli obiettivi dell'utente, l'interfaccia è facile da usare.

GUI Design: la progettazione delle GUI si concentra sull'anticipazione di ciò che gli utenti potrebbero dover fare e sull'assicurazione che l'interfaccia abbia elementi di facile accesso/comprendimento/utilizzo per facilitare le operazioni che vuole fare l'utente. La progettazione delle GUI unisce i concetti di design interattivo, design visivo e architettura dell'informazione. Una struttura organizzativa è il modo in cui vengono definite le relazioni tra parti di contenuto di un sistema/oggetto, le tre principali strutture organizzative sono gerarchica, sequenziale e a matrice:

- **Struttura gerarchica:** in queste strutture ad albero è presente un approccio dall'alto verso il basso per la navigazione delle informazioni. Molti utenti hanno familiarità con la struttura delle informazioni in gerarchie perché è un modello che viene sperimentato su base giornaliera.
- **Struttura sequenziale:** le strutture sequenziali richiedono agli utenti di andare passo dopo passo, seguendo un percorso preimpostato di contenuti (esempio: quando si vuole acquistare qualcosa).
- **Struttura a matrice:** una struttura a matrice consente agli utenti di determinare il proprio percorso dato che i contenuti sono collegati tra loro in modi diversi (esempio: un utente può scegliere di navigare i contenuti in base alla data mentre un altro naviga in base all'argomento).
- **Database Model:** adotta un approccio dal basso verso l'alto (al contrario della struttura gerarchica). Il contenuto all'interno di questa struttura si basa sui collegamenti creati tramite i metadati del contenuto. Questo modello facilita un'esperienza più dinamica, consentendo funzionalità di filtro e ricerche avanzate.

l'architettura della GUI di un sito ha un impatto a lungo termine sul sito stesso. I gestori dovrebbero tenere presente quanto segue:

- **Lasciare spazio alla crescita:** creare un sito che possa ospitare l'aggiunta di nuovi contenuti all'interno di una sezione e l'aggiunta di intere nuove sezioni.
- **Evitare strutture troppo superficiali o troppo profonde:** strutture troppo superficiali richiederebbero menù enormi che non presentano raggruppamenti logici i quali renderebbero più semplice il movimento sul sito. Strutture troppo profonde potrebbero imporre all'utente di dover navigare attraverso troppi livelli prima di trovare l'informazione che desidera.

Information architecture: l'architettura dell'informazione (IA) si concentra sull'organizzazione, strutturazione ed etichettatura dei contenuti in modo efficace. L'obiettivo è quello di aiutare gli utenti a trovare le informazioni e completare le attività oltre che a far capire all'utente dove si trovano, cosa hanno trovato e cosa devono aspettarsi. I componenti principali dell'IA sono:

- **Schemi e strutture organizzative:** come categorizzare e strutturare le informazioni.
- **Sistemi di etichettatura:** come rappresentare le informazioni.
- **Sistemi di navigazione:** come gli utenti navigano o si spostano tra le informazioni.
- **Sistemi di ricerca:** come gli utenti cercano le informazioni.

gli schemi organizzativi hanno a che fare con il modo in cui vengono classificati i contenuti e i vari modi in cui vengono create le relazioni tra ogni pezzo del sistema. Gli schemi si suddividono in due tipi:

- 1) **Schemi esatti:** gli schemi esatti dividono OGGETTIVAMENTE le informazioni in sezioni che si escludono a vicenda. Questi sistemi sono comodi per i progettisti ma talvolta possono risultare poco confortevoli per gli utenti, è necessario infatti che l'utente capisca cosa sta cercando e si adatti al modello.
- 2) **Schemi soggettivi:** gli schemi organizzativi soggettivi classificano le informazioni in modo che può essere specificato dall'organizzazione o dal campo. Sono spesso più utili degli schemi organizzativi esatti, l'implementazione di questo tipo di schemi, infatti, mantiene le cose più semplici per l'utente e hanno la capacità di identificare la categorizzazione e formare un modello mentale che può essere compreso rapidamente.

N.B: evitare l'uso di schemi ibridi.

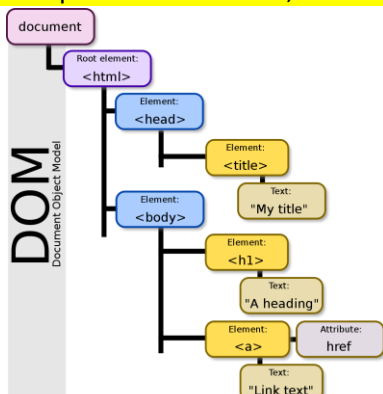
Esempi di schemi esatti:

- **Schemi alfabetici:** fanno uso dell'alfabeto per organizzare i loro contenuti. Affinché questo tipo di schema funzioni l'etichettatura del contenuto deve corrispondere alle parole che gli utenti stanno cercando.
- **Schemi cronologici:** organizza i contenuti per data. Affinché questo schema funzioni, deve esserci accordo sul momento in cui si è svolto l'oggetto del contenuto.
- **Schemi geografici:** organizza i contenuti in base al luogo. A meno che non vi siano controversie sui confini, questo tipo di schema è abbastanza semplice da progettare e utilizzare.

Esempi di schemi soggettivi:

- **Schemi di argomenti:** organizza i contenuti in base all'argomento specifico (genere di film su netflix).
- **Schemi di attività:** organizzano il contenuto considerando le esigenze, le azioni/operazioni, le domande che gli utenti apportano a quel contenuto specifico.
- **Schemi del pubblico:** organizzano i contenuti per segmenti separati di utenti. Gli utenti possono spostarsi da un pubblico all'altro e i contenuti sono categorizzati in base all'utente.
- **Schemi di metafora:** aiutano gli utenti mettendo in relazione il contenuto con concetti familiari. Viene utilizzato nella progettazione di interfacce (cartelle, cestino...).

Document Object Model (DOM): è un'interfaccia che consente a un linguaggio di programmazione di manipolare il contenuto, la struttura e lo stile di un sito web. JavaScript è il linguaggio di scripting lato client



che si connette al DOM di un browser Internet. Quasi ogni volta che un sito Web esegue un'azione (messaggi di errore, presentazione di immagini, ecc..) è il risultato dell'accesso e della manipolazione del DOM da parte di JavaScript. Il DOM rappresenta il documento come un albero logico e i metodi di DOM consentono l'accesso a questo albero, tramite l'utilizzo di quest'ultimi si può cambiare la struttura, lo stile o il contenuto di un documento. Ai nodi inoltre possono essere associati dei gestori di eventi, ogni qual volta un evento si verifica, i gestori vengono eseguiti.

DOM in web browser: per eseguire il rendering di un documento (un file HTML), la maggior parte dei browser Web utilizza un modello interno simile a DOM. i nodi di ogni documento sono organizzati all'interno di

un albero (vedi immagine). Quando una pagina HTML viene visualizzata nei browser quest'ultimo scarica la pagina nella memoria locale per poi visualizzare la pagina a schermo.

DOM in JavaScript: quando viene caricata una pagina Web JavaScript, il browser crea un DOM della pagina, che è la rappresentazione orientata agli oggetti di un documento HTML e che funge da interfaccia tra JavaScript e il documento stesso. Questa procedura consente la creazione di pagine web dinamiche e modificabili.

Componenti di interfaccia: quando si progetta un'interfaccia, bisogna cercare di essere coerenti e prevedibili nella scelta degli elementi dell'interfaccia. Con le scelte giuste gli utenti acquisiscono familiarità con il sistema. Gli elementi dell'interfaccia includono:

- **Controlli di input:** caselle di controllo/riempimento, campi di testo/data, pulsanti di opzione, ecc...
- **Componenti di navigazione:** barra di navigazione, slider, icone, ecc...
- **Componenti informativi:** descrizione dei comandi, finestre di messaggio, notifiche, ecc...
- **Contenitori**

Personas: un personaggio utente è un archetipo che rappresenta un potenziale utente di un sito web/sistema/applicazione. Vengono di gran lunga utilizzati nella fase di ricerca dell'esperienza utente. Chi progetta per la user experience raccoglie dati relativi agli obiettivi/problemi riscontrati dei loro utenti e, quindi, crea personaggi di conseguenza. L'utilizzo di questi archetipi aiuta a:

- comprendere la gamma di utenti possibili nel sistema.
- colmare il divario tra azienda e i suoi utenti permettendo di misurare il vero comportamento di quest'ultimi.
- Guidare le decisioni di progettazione consentendo ai team di ottenere una comprensione più approfondita degli utenti.

La creazione di un utente persona inizia con la ricerca sull'utente: osservando gli utenti, gli UXers possono comprendere il loro comportamento e le loro motivazioni per poi progettare di conseguenza. Alcune tecniche per catturare informazioni sono:

- Analisi delle attività
- Feedback (interviste contestuali)
- Prototipazione (sperimentare idee prima di svilupparle)

Il design della persona dell'utente è basato sul principio di Pareto quest'ultimo afferma che, per molti eventi, circa l'80% degli effetti proviene dal 20% delle cause. Applicato al design della persona dell'utente: bisogna concentrarsi sul 20% della base utenti che utilizzerà/acquisterà l'80% delle funzionalità/prodotti. Dopo aver osservato e suddiviso i dati dei test tutte le informazioni vengono contestualizzate in un modello utente.

Requisiti: sono servizi/funzioni/caratteristiche di cui un utente ha bisogno. I requisiti devono essere presenti per soddisfare le esigenze degli utenti previsti. Definire una serie completa e dettagliata di requisiti troppo presto in un progetto si rivela spesso controproducente, l'ambiente progettuale cambia con il passare del tempo e nuove esigenze/opportunità potrebbero presentarsi. I requisiti vengono appresi man mano che il progetto avanza e il tema capisce di più sulle esigenze aziendali/progettuali.

Requisiti funzionali (FR): esprimono la funzione o la caratteristica e definiscono ciò che è richiesto

Requisiti non funzionali (NFR): descrivono come una soluzione deve comportarsi e descrivono gli attributi come sicurezza, affidabilità, disponibilità e manutenibilità. Gli NFR possono essere:

- 1) **A livello di soluzione (influiscono su un gruppo di requisiti funzionali):** tutte le funzionalità rivolte al cliente devono riportare il logo dell'azienda, tutte le funzionalità rivolte al cliente devono rispondere entro 2 secondi.
- 2) **Relativi a un particolare requisito funzionale:** la sede dell'azienda potrebbe avere NFR di accessibilità/sicurezza/disponibilità.

esempi di requisiti:

- Il Course Manager ha l'obbligo di programmare corsi di formazione e prenotare aule, al fine di rendere visibili i corsi disponibili e per garantire che i corsi vengano eseguiti in modo efficace.

- Il responsabile del centro di formazione ha l'obbligo di tenere traccia della formazione in corso, al fine di garantire un'adeguata assegnazione dei formatori ai corsi.

User stories: una user story è una breve dichiarazione che identifica l'utente e il suo bisogno/obiettivo. Determina chi è l'utente, di cosa ha bisogno e perché ne ha bisogno, aiutano quindi a documentare le diverse esigenze/motivazioni per accedere a un sito Web o un app. Esiste la possibilità di scrivere una user story per coprire grandi quantità di funzionalità, queste stories di alto livello sono note come "epiche". Poiché una epica è troppo di alto livello (troppo generica) per poter essere utilizzata/elaborata può essere suddivisa in più livelli che aggiungono dettaglio. I dettagli possono essere aggiunti alle storie degli utenti in 2 modi:

- 1) **Suddividendo una user story in più user story più piccole:** quando una storia relativamente grande viene divisa in più storie utente più piccole, è naturale che siano stati aggiunti dettagli.
- 2) **Aggiungendo "condizioni di soddisfazione":** sono un test di accettazione di alto livello che sarà vero dopo che la user story sarà completa.

Considera quanto segue come un altro esempio di user story agile: *In qualità di vicepresidente del marketing, desidero selezionare un periodo di vacanze da utilizzare durante la revisione delle prestazioni delle campagne pubblicitarie passate in modo da poter identificare quelle redditizie.*

È possibile aggiungere dettagli a tale esempio di user story aggiungendo le seguenti condizioni di soddisfazione:

- Assicurati che funzioni con le principali festività al dettaglio: Natale, Pasqua, Festa del Presidente, Festa della mamma, Festa del papà, Festa del lavoro, Capodanno.
- Supportare i giorni festivi che si estendono su due anni solari (nessuno su tre).
- Le stagioni delle vacanze possono essere impostate da una vacanza all'altra (come il Ringraziamento a Natale).
- Le stagioni delle vacanze possono essere impostate su un numero di giorni prima della vacanza.

Scenarios: uno scenario è una situazione che cattura il modo in cui gli utenti eseguono le attività sul sito/app e forniscono un contesto all'esperienza utente prevista. Sostanzialmente è uno sviluppo di una user story e può riguardare più utenti target. Gli scenari possono essere anche suddivisi in più casi d'uso che descrivono il flusso di attività che un utente esegue in una determinata funzionalità o percorso. I buoni scenari rispondono alle seguenti domande:

- **Chi è l'utente:** usa i personaggi che sono stati sviluppati per riflettere i reali utenti che attivano sul sito.
- **Perché l'utente accede al sito:** annota cosa motiva l'utente a visitare il sito.
- **Quali obiettivi ha:** attraverso l'analisi delle attività, si può capire meglio cosa vuole l'utente.
- **Alcuni scenari possono rispondere anche a come l'utente può raggiungere i propri obiettivi sul sito.**

Come si scrive uno scenario: tramite l'analisi dell'utente principale definito tramite lo sviluppo della persona, il team di progettazione e gli sviluppatori possono considerare l'attività chiave che l'utente spera di ottenere. Il passaggio successivo consiste nell'eseguire un'analisi dello scenario e nella contestualizzazione degli obiettivi dell'utente. Grazie agli scenari si può determinare:

- I punti più importanti su cui concentrarsi durante la progettazione della UX.
- Quali passaggi richiedono ulteriore assistenza agli utenti.
- Principali esigenze e motivazione dei tuoi utenti.

Gli scenari si basano sulla User Story -> le storie sono brevi affermazioni che descrivono ciò di cui una User Persona ha bisogno e perché, gli scenari portano le user stories al livello successivo aggiungendo alla storia una contestualizzazione e l'interazione con il prodotto.

Use cases: è una descrizione di come gli utenti eseguiranno le attività sull'applicazione. Delinea dal punto di vista dell'utente il comportamento del sistema durante una richiesta.

- **Uno scenario:** può essere composto da più casi d'uso. Implica una situazione che può avere più "attori" che utilizzano una determinata funzionalità per raggiungere il loro obiettivo.
- **Un caso d'uso:** coinvolge un "attore" e il flusso che un particolare "attore" assume in una determinata funzionalità. Il caso d'uso è più granulare dello scenario e aiuta a spiegare come dovrebbe comportarsi il sistema e cosa potrebbe andare storto.

I casi d'uso descrivono una combinazione dei seguenti elementi:

- **Attore:** chi sta usando il sistema.
- **Stakeholder:** qualcuno o qualcosa con interessi nel comportamento del sistema in discussione.
- **Attore principale:** stakeholder che avvia un'interazione con il sistema per raggiungere un obiettivo.
- **Prerequisiti:** cosa vede essere vero prima e dopo l'esecuzione del caso d'uso.
- **Trigger:** evento che causa l'avvio del caso d'uso.
- **Principali scenari di successo ("flusso base"):** caso d'uso in cui nulla va storto.
- **Percorsi alternativi ("flusso alternativo"):** questi percorsi sono una variazione del percorso base. Sono eccezioni che si presentano quando le cose vanno male a livello di sistema.

come scrivere un caso d'uso:

- 1) identificare chi utilizzerà il sito web
- 2) scegliere uno di questi utenti
- 3) definire ciò che l'utente vuole fare sul sito. Ogni cosa sul sito diventa un caso d'uso
- 4) per ogni caso d'uso decidere il normale corso degli eventi
- 5) descrivere il corso base del caso d'uso in termini di ciò che fa l'utente e di ciò che fa il sistema in caso di risposta.
- 6) Quando viene descritto il corso base, considerare anche altri percorsi alternativi e aggiungerli per "estendere" il caso d'uso.
- 7) Cercare i punti in comune tra i casi d'uso e annotarli come "casi d'uso comuni"
- 8) Ripetere i passaggi da 2 a 9 per ogni utente

Takeaway: coinvolgere lo sviluppo delle personas, della user story, dello scenario e del caso d'uso permette di identificare informazioni chiave sugli utenti e creare prodotti validi (tutto ciò che si fa per avvicinarci agli utenti è un passo nella giusta direzione).

Innovation: l'innovazione è comunemente definita come "realizzazione di nuove combinazioni". Un elemento comune alle varie definizioni è l'attenzione alle novità, al miglioramento e alla diffusione. L'innovazione è correlata all'invenzione, ma non è la stessa. Non è possibile innovare senza un approccio HCD.

Sustaining innovation: la maggior parte dei processi di innovazione si basa sull'innovazione incrementale e su piccoli miglioramenti dei prodotti esistenti.

- È un processo graduale.
- A basso rischio.
- Bassa velocità.
- Non sono richieste modifiche all'organizzazione aziendale.
- Non è richiesta alcuna riqualificazione dell'utente.
- Bassa probabilità di cambiare/espandere l'attività.
- Utenti target e settore di mercato stabili.

Disruptive Innovation: un'innovazione dirompente è un'innovazione che crea un nuovo mercato e sconvolge un mercato esistente. Non tutte le innovazioni sono dirompenti, anche se sono rivoluzionarie (automobile fino al modello Ford).

Human Centered Design Process: è un framework per l'organizzazione di progetti incentrati sull'utente. IDEO è una delle aziende di design più innovative e premiate del mondo, secondo loro la chiave per capire cosa vogliono gli utenti risiede nel fare due cose:

- 1) **Osservare il comportamento degli utenti:** cercare di capire le persone osservandole nelle attività.
- 2) **Mettersi nella situazione dell'utente finale:** per capire realmente l'esperienza dell'utente. IDEO utilizza quindi le informazioni che ottiene per alimentare i propri progetti.

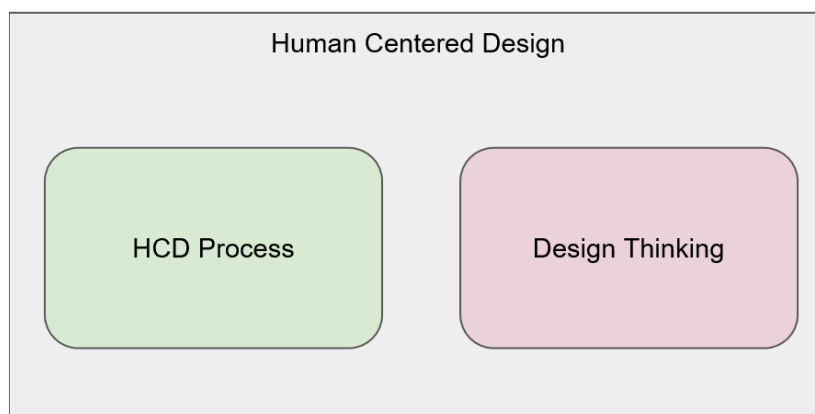
Passi per una buona realizzazione HCD:

- 1) **Ispirazione:** è la fase che consiste nel comprendere le esigenze e le difficoltà degli utenti. In questa fase è necessario abbandonare qualsiasi idea preconcepita per aprirsi ad una più vasta gamma di possibili soluzioni.
- 2) **Ideazione:** questa fase riguarda la visualizzazione, il brainstorming e la discussione di tutte le potenziali soluzioni. Tutto ciò di cui si ha bisogno in questa fase sono alcuni schizzi/modelli utili per attingere alla propria creatività senza la pressione di produrre un prodotto finale lucido
- 3) **Implementazione:** le fasi precedenti avevano lo scopo di gettare le basi per la ricerca di un concetto giusto prima di realizzare un prototipo. In questa fase viene assemblato un prototipo ad alta fedeltà che gli utenti possono provare.

Design thinking: è un processo che si segue per creare soluzioni che saranno effettivamente adottate dalle persone. È un processo iterativo in cui si cerca di comprendere l'utente e ridefinire i problemi nel tentativo di identificare strategie e soluzioni alternative che potrebbero non essere subito evidenti nella fase iniziale di comprensione. Il design thinking è un modo di pensare e di lavorare estremamente utile per affrontare problemi che sono mal definiti o sconosciuti riformulando il problema in modo da incentrarlo sull'uomo. Comprende 5 fasi:

- 1) **Empatia:** studio degli utenti.
- 2) **Definire le esigenze degli utenti, i solo problemi e le tue intuizioni.**
- 3) **Ideare sfidando ipotesi e creando idee per soluzioni innovative.**
- 4) **Prototipo**
- 5) **Test**

L'utilizzo del design thinking unito allo human centered design garantirà il soddisfacimento delle esigenze delle persone.



Waterfall development: la metodologia waterfall è un approccio di gestione del progetto lineare, in cui i requisiti delle parti interessate e dei clienti vengono raccolti all'inizio del progetto, viene poi creato un piano di progetto sequenziale per soddisfare tali requisiti. Questo piano di progettazione viene definito "waterfall" ovvero a cascata perché ogni fase del progetto si sovrappone.

Agile: è la capacità di creare e rispondere al cambiamento. Si tratta di un metodo che permette di pensare a come si può capire cosa sta succedendo nell'ambiente in cui ci si trova, identificare quale incertezza si sta riscontrando e capire come ci si può adattare durante il procedimento. Una cosa che separa agile da altri approcci allo sviluppo software è l'attenzione alle persone che svolgono il lavoro e la loro collaborazione (c'è una grande attenzione nella comunità software agile sulla collaborazione e sui team auto-organizzati). Agile è il miglior metodo di costruzione per HCD e design thinking. Agile è più di un framework come Scrum,

extreme programming o FDD, per “sviluppo agile del software” si intende una serie di strutture e pratiche (esprese nel manifesto dello sviluppo agile).

Scrum: è un framework agile per lo sviluppo software. È progettato per team di dieci o meno membri i quali suddividono il lavoro in obiettivi che possono essere completati entro iterazioni temporizzate chiamate sprint. Uno sprint è l'unità base dello sviluppo scrum la cui durata è fissata in anticipo e varia dal mese alle due settimane. Ogni sprint termina con una revisione dello stesso e una retrospettiva dello sprint, quest'ultima rivede i progressi da mostrare agli stakeholder e identifica i miglioramenti per gli sprint successivi. Uno dei principi chiave dello scrum è il riconoscimento del fatto che i clienti cambieranno idea su ciò che vogliono o di cui hanno bisogno e che ci saranno problemi imprevedibili per i quali un approccio predittivo non è adatto. Lo scrum afferma quindi che il problema non può essere completamente compreso o definito in anticipo, il suo modello punta invece a concentrarsi su come massimizzare la capacità del team di fornire rapidamente soluzioni rispondendo ai requisiti emergenti.

Ruoli nel framework Scrum:

- 1) **Il product owner:** che rappresenta gli stakeholder del prodotto e la voce del cliente, è responsabile della fornitura di buoni risultati di business.
- 2) **Il team di sviluppo**
- 3) **Lo scrum master:** non è un tradizionale team leader o project manager, ma funge da cuscinetto tra il team e qualsiasi influenza che distrae.

DevOps: prevede, attraverso la collaborazione e integrazione tra sviluppatori e addetti alle operation, la gestione più flessibile, affidabile, sicura e controllabile dei rilasci. Il DevOps si ispira alla metodologia Agile, alla necessità di incrementare la frequenza di rilasci in produzione e fa leva sulla disponibilità di infrastrutture virtualizzate e in cloud.

Pretotyping: quando si tratta di leggi di mercati si dovrebbe presumere che un potenziale nuovo prodotto sia sempre un fallimento, almeno finché non si ottengono prove oggettive sufficienti per farci credere il contrario. Per comprendere il fallimento del mercato è utile testare quest'ultimo per le nostre idee in modo obbiettivo, rigoroso e rapido prima di investire per lo sviluppo. La pretotipazione fornisce gli strumenti e le tecniche necessari per convalidare un'idea con risorse minime e tempo breve.

Thoughtland:

- **Lost in translation problem:** nel momento in cui si prova ed esporre la propria idea (ciò che si vede nella nostra mente) ci si imbatte in un impegnativo problema di traduzione, specialmente se l'idea è nuova e diversa da qualsiasi altra cosa.
- **The prediction problem:** anche se la comprensione astratta dell'idea del pubblico è molto simile a quella del progettista, le persone non sono in grado di prevedere se vorrebbero/apprezzerebbero qualcosa che non hanno ancora sperimentato.

I falsi positivi possono indurre a credere che la tua idea sia immune alla legge del fallimento del mercato.

I falsi negativi, d'altra parte, possono spaventare e potrebbero far scartare prematuramente un'idea potenzialmente buona.

L'unico modo per evitare i falsi negativi e positivi è trasportare l'idea da **thoughtland** ad **actionland**:

- **Thoughtland:** si usano idee astratte per porre domande ipotetiche e raccogliere opinioni
- **Actionland:** si utilizzano artefatti per sollecitare azioni e raccogliere dati.

Pretotype: i prototipi possono aiutare a far fallire più velocemente un'idea in modo tale da non investire troppo tempo in un'idea poco fruttuosa. Spesso può capitare di non abbandonare abbastanza velocemente un prototipo, in questo caso si parla di **producttypes** (un prototipo andato troppo oltre). I prototipi consentono di raccogliere dati sull'utilizzo e sul mercato in modo da poter prendere una decisione, sono modelli del prodotto o del servizio previsto.

Prototipo -> può implicare un costo elevato di produzione e di realizzazione a fronte di uno standard di qualità non elevato.

Pretotipo -> è un escamotage che permette di passare dal mondo delle idee a quello dei fatti, aiuta a capire se l'oggetto di cui si fa il pretotipo può avere mercato.

La scienza della prototipazione ha lo scopo di aiutare gli innovatori nei seguenti modi:

- Identifica la caratteristica principale del nuovo potenziale prodotto.
- Aiuta a capire quali funzionalità principali possono/devono essere simulate.
- Utilizza i modelli (prototipi) per testare e raccogliere feedback e dati sull'utilizzo.
- Analizza i dati di utilizzo per determinare il passaggio successivo.

Pretotyping flow:

- 1) **Isolare il presupposto chiave:** l'unico presupposto che, se falso, significa che non è sicuramente quello giusto.
- 2) **Scegliere un tipo di Pretotipo:** ovvero scegliere quale tipo di pretotipo ti consentirà di isolare e testare l'ipotesi chiave.
- 3) **Creare un'ipotesi di coinvolgimento nel mercato:** quante persone faranno cosa con il pretotipo.
- 4) **Provare il prototipo:** mettere il prototipo nel mondo reale e studiare le persone che interagiscono con esso.
- 5) **Imparare e perfezionare:** bisogna valutare i risultati e affinare il pretotipo con i nuovi dati. Se l'ipotesi è sostenuta, bisogna decidere in quali altre simulazioni bisogna testare il pretotipo per ottenere un quadro completo (hypozooming).

Fake Door: può essere utilizzata per pubblicizzare un servizio che non è ancora pronto e per misurare l'interesse degli utenti.

Cosa -> un punto di ingresso marketing per un'idea non ancora sviluppata.

Perché -> una soluzione non è ancora uscita e si vuole catturare una prima indicazione di interesse a un costo molto basso.

Quando -> l'idea può essere descritta e presentata ai clienti dove ti aspetti di trovarli.

Come -> pubblicizzando un nuovo prodotto/funzionalità per poi monitorare il tasso di risposta dei clienti

Dove -> la tecnologia web consente un metodo estremamente affidabile che include: annunci online, pagine di destinazione è un semplice modulo di risposta.

Esempio: una landing page che potrebbe chiedere all'utente di registrarsi/pagare per un prodotto che ancora non esiste (sistemi di crowdfunding).

Mechanical Turk: per la realizzazione di progetti sono necessari algoritmi, prima di trovare la giusta implementazione tramite l'algoritmo è possibile utilizzare un approccio mechanical turk. tramite questo approccio è possibile sperimentare l'algoritmo utilizzando l'input umano.

Cosa -> usa l'umano per simulare la tecnologia che richiederebbe troppi soldi o tempo per essere costruita.

Perché -> rimanda lo sviluppo costoso fino a quando non si è in grado di valutare l'interesse sul mercato.

Quando -> quando il prodotto finale richiede lo sviluppo di una tecnologia costosa e complessa.

Come -> utilizzare un'interfaccia realistica per fornire ai client target un'esperienza della tecnologia proposta. Le funzionalità back-end complesse vengono fornite attraverso l'input umano

Dove -> nella stessa situazione di vita reale in cui verrà utilizzata.

Esempio: IBM speech-to-talk ha verificato se i clienti fossero interessati a una tecnologia speech-to-text senza implementarla, ma utilizzando semplicemente l'aiuto di alcuni dattilogafi.

Impersonator: fornisce un front-end diverso oltre a un'API esistente

Cosa -> utilizzare un prodotto o servizio esistente per proporsi come offerta in prova.

Perché -> risparmiare sui costi di sviluppo mentre l'interesse del mercato non è ancora convalidato.

Quando -> quando è necessario creare un plausibile sostituto per dimensione, forma e caratteristiche della soluzione.

Come -> applicare una maschera ad un prodotto esistente in modo tale che possa essere un buon sostituto.

Dove -> nella stessa situazione di vita reale in cui verrà utilizzata.

Esempio: quando la tesla ha utilizzato il telaio della Lotus elise, rimpiazzando il motore con un motore elettrico per verificare l'indice di gradimento dell'utenza.

Pinocchio: un falso artefatto funge da proxy per l'oggetto reale. Questo modello aiuta le persone a testare il prodotto che si sta costruendo ma anche per esplorare altri modi in cui l'oggetto può essere adattato alle loro esigenze.

Cosa -> un artefatto inanimato che funge da proxy per l'oggetto reale

Perché -> la soluzione ancora non esiste ma si desidera convalidare un parametro di progettazione chiave

Quando -> quando la soluzione richiede un cambiamento significativo o un adattamento da parte dei clienti per sviluppare una nuova abitudine.

Come -> il proxy viene utilizzato per convalidare determinati parametri del prodotto come fattore di forma, funzionalità e usabilità.

Dove -> nella stessa situazione di vita reale in cui verrà utilizzata.

Esempio: la creazione di un modello di palm pilot ricavato da un pezzo di legno e plastica per sperimentare il modo in cui il prodotto poteva essere percepito essendo tenuto in mano

One night stand: l'obiettivo in questo caso è ridurre i costi o indirizzare una classe di utenza ben nota. Se il prototipo one night stand applicato a un target non ha successo, allora sappiamo che stiamo "costruendo" qualcosa di sbagliato.

Cosa -> un'esperienza di servizio completa senza che l'infrastruttura sia permanente.

Perché -> evitare investimenti in infrastrutture complesse e convalidare l'interesse sul mercato.

Quando -> quando la soluzione dipende da un'esperienza di servizio interattiva, quando si devono testare punti di intercettazione del cliente, quando si desidera convalidare un mercato prima di espandersi.

Come -> fornire ai clienti target l'esperienza essenziale in un arco di tempo ristretto.

Dove -> nella stessa situazione di vita reale in cui verrà utilizzata, ma in un arco di tempo limitato.

Esempio: fruit of the loom ha affittato spazi dismessi per brevi periodi di tempo per testare l'interesse dei consumatori in nuovi capi di abbigliamento.

Facade: uguale al metodo di prototyping one night stand con la differenza che in questo caso simula un'infrastruttura più stabile e complessa. Tutti i beni e gli spazi sono presi in prestito o affittati a basso costo.

Minimum Viable Product: l'MVP è la versione di un prodotto con caratteristiche appena sufficienti per essere utilizzabile dai primi clienti. È una strategia mirata ad evitare di costruire prodotti che i clienti non vogliono, che cerca di massimizzare le informazioni apprese sul cliente. Un MVP non è, quindi, un prodotto minimo, ma una strategia ed un processo diretto verso la realizzazione e vendita di un prodotto per determinati clienti.

Cosa -> definisce la transizione tra prototyping a prototyping del prodotto.

Perché -> c'è bisogno di far provare all'utente un prodotto reale in modo tale da fare test veritieri.

Quando -> nel momento in cui si sono ricavati tutti i possibili dati dell'aspetto marketing attraverso i semplici prototipi (fake door, pinocchio, mechanical turk, one night stand, impersonator). Ulteriori informazioni richiedono un'interazione più profonda del cliente con un artefatto funzionante.

Come -> creando un artefatto che offre le "core function" della soluzione.

Dove -> nella stessa situazione di vita reale in cui verrà utilizzata.

Lean canvas: è uno strumento di progettazione e validazione di un'idea di business, permette a chi sta sviluppando un progetto di fare chiarezza sulla propria idea. Ragionare di fronte al lean canvas non solo ci obbliga a non perdere di vista l'insieme dei passaggi logici necessari a rendere l'idea veramente completa, ma aiuta anche a rendere l'idea comunicabile.