

Interazione uomo-macchina

Sommario

Semiotica e storia visiva	4
Origine e aree disciplinari	5
L'utente	5
Ergonomia cognitiva	5
Dispositivi per l'interazione	6
Tematiche principali	6
Sistema socio-tecnico	7
Proprietà emergenti	7
Progettazione	7
Progettazione centrata sull'usabilità	8
Funzionalità e sovraccarico cognitivo.....	9
Blackbox e trasparenza.....	9
Affordance e progettazione intuitiva.....	10
Problemi di terminologia e consistenza	10
Qualità dell'interfaccia	10
Semplificazione	11
Principi e standard	11
Heat maps e neuromarketing	12
Effetti psicologi nell'interazione.....	12
Vincoli	12
Workflow	13
Sentiero del desiderio	13
Key informants	14
Affordance	14
Skeuomorphism.....	14
User-created affordances	15
Mapping	16
Designing by affordance	16
Valutazione dell'usabilità	17

Principi e standard	19
Uso dei principi	19
IXD check-list	20
Principi di ISO 9241-110	20
Euristiche di Nielsen	21
Tecniche.....	26
Procedura.....	26
Osservatore	27
Curva di apprendimento.....	28
Prioritizzazione dei problemi	28
Perché usarla?.....	29
Tabella	29
Errori frequenti	30
Contenuti del rapporto finale	31
Sistema walk-up and use	37
Esercitazione	38
Obsolescenza programmata	38
Dark patterns.....	38
Tipologie	38
Approcci teorici alla teoria delle decisioni	41
Approccio normativo.....	41
Approccio descrittivo	41
Approccio prescrittivo	41
Teoria del doppio processo	41
Euristiche.....	42
Stili decisionali	42
Architettura della scelta.....	42
Nudge.....	43
Bias cognitivi	43
Effetto framing.....	43
Effetto default	43
Effetto posizione.....	44

Effetto social proof.....	44
Usabilità.....	44
Facts of Life.....	44
Euristiche di Nielsen	44
Etica	45
Nella ricerca.....	45
I principi.....	46
Consenso informato	46
Debriefing	46
Auto-valutazione etica	47
Comitato etico	47
Privacy	47
Metodi qualitativi	47
Dati qualitativi	47
Osservazione	47
Focus group	48
Think-aloud protocol.....	48
Intervista.....	49

Riguarda la **progettazione, realizzazione e valutazione di sistemi interattivi** dotati di capacità computazionali destinati all’uso umano, insieme allo **studio dei fenomeni principali che li circondano**.

Questo campo è curato da istituzioni come l’*Association for Computing Machinery* (ACM).

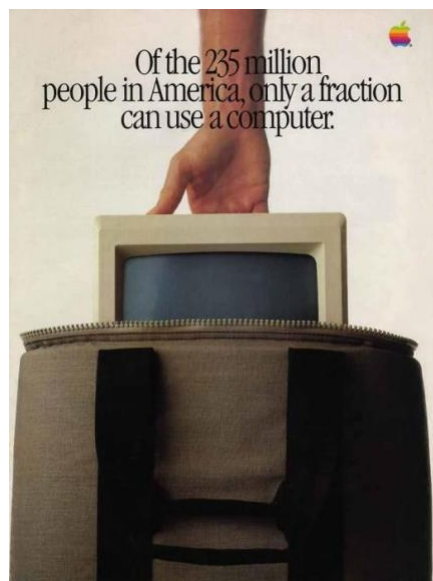
Questa disciplina non si limita alla creazione di sistemi interattivi, ma include anche lo studio delle attività e dei fenomeni che si sviluppano attorno all’uso di tali sistemi. Poiché l’essere umano è una **creatura sociale**, l’interazione uomo-macchina include anche l’interazione **uomo-uomo mediata da una macchina**.

Semiotica e storia visiva

Un aspetto rilevante è l'**analisi semiotica**, un'analisi che studia i segni e i simboli per comprenderne il significato e l'impatto comunicativo.

Andando a vedere le prime immagini di pubblicità per i computer, possiamo notare un paio di cose che vennero adottate fin dall'inizio per la commercializzazione dei computer.

Le fotografie degli anni '60 ritraggono donne al lavoro sui computer; queste immagini suggeriscono che le **macchine siano facili da usare, e accessibili a chiunque**. Esse



rappresentano anche una differenza di ruoli: mentre gli uomini sono spesso rappresentati in piedi, liberi di muoversi, le donne appaiono vincolate alla postazione. Il punto della pubblicità, comunque, era presentare il computer come uno strumento accessibile, e col tempo, l'usabilità in se migliorò: partendo da vere e proprie stanze e armadi dove le persone dovevano entrare per poter cambiare parametri nel computer, si arrivò poi alle schede perforate, e finalmente ai **pulsanti**.

Con l'introduzione del **personal computer** (termine di marketing, ovviamente) da parte di IBM, si diffuse l'idea che non solo il computer passava da essere soltanto per il business, industrie, grande e complicato, ma che poteva essere per chiunque, e utilizzato da tutti.

Le pubblicità presentavano il PC in un contesto di relax e divertimento, enfatizzando la **facilità d'uso**, distanziandosi dalla percezione che fosse solo uno strumento lavorativo.

Nel **1984**, con l'introduzione del Macintosh da parte di Apple, nacque il concetto di **usabilità**, e la presentazione al grande pubblico dell'interfaccia utente grafica a mouse. Il computer divenne ancora più facile da usare. L'innovazione in se, però, nacque dalla Xerox a Palo Alto.

Origine e aree disciplinari

La disciplina dell'IUM nasce formalmente nel 1983, con la prima conferenza annuale dell'ACM, **CHI** (Computer-Human Interaction), pochi anni dopo l'introduzione dell'IBM PC nel 1981, e del Macintosh nel 1984. Le sue radici affondano nell'intersezione di due aree disciplinari principali: l'**ergonomia** (o fattori umani), e l'**informatica**, coinvolgendo anche diversi ruoli come *psicologi, ingegneri, informatici e sociologi*. Si articola in tre aree principali:



- **Scienza del computer:** riguarda la programmazione delle interfacce, i paradigmi di interazione, i modelli di dialogo, e la grafica computazionale.
- **Scienza della progettazione:** include il design dell'interazione, il design industriale, l'ergonomia e la gestione dei progetti.
- **Scienze dell'uomo:** comprendono la psicologia, le scienze cognitive, la linguistica e le scienze della comunicazione.

L'utente

È importante ricordare che **non esiste l'utente in senso astratto**, ma esistono utenti con specifiche intenzioni e obiettivi. L'**ergonomia**, che studia le leggi del lavoro, è essenziale per progettare sistemi che siano effettivamente utili agli utenti.

Un aspetto cruciale della progettazione è che **l'utente finale non è mai un esperto del sistema**, né ha familiarità con esso. Ciò rende complessa la progettazione di interazioni, soprattutto a causa della varietà di sistemi e contesti in cui vengono utilizzati. Tale varietà si può delineare in tre categorie:

- **Sistemi interattivi:** domotica, dispositivi indossabili, dispositivi mobili, sistemi tradizionali (in contesti complessi o sfidanti)
- **Diversità degli utenti:** utenti in contesti virtuali, medici o normali
- **Varietà degli scopi:** usi diversificati a seconda delle necessità degli utenti

Ergonomia cognitiva

È lo studio delle interazioni tra l'uomo e gli strumenti che servono per elaborare l'informazione; quindi, si studiano anche tutti i processi cognitivi coinvolti come percezione, attenzione, memoria, età.

Dispositivi per l'interazione

Gli umani possono sia **dare** sia **ricevere** input utilizzando dispositivi specifici. Ad esempio, il tatto può essere supportato da un guanto, la vista da occhiali, ecc....

In questo modo, anche la postura può diventare un canale di interazione, grazie a sensori di prossimità che rilevano la posizione e i movimenti del corpo.,

Tematiche principali

Le principali tematiche includono:

- **Criteri, metodi e strumenti** per la progettazione dell'interazione tra utenti e sistemi, accompagnati dalla valutazione dell'usabilità, in conformità agli standard ISO.
L'**usabilità** è una componente fondamentale, poiché misura quanto facilmente gli utenti possono utilizzare un sistema in un contesto reale.
- **Progettazione di nuove tecniche di interazione**, mirate a migliorare la qualità dell'interazione tra l'utente e il sistema, rendendola più intuitiva, efficiente e soddisfacente.
- **Valutazione dell'impatto dell'automazione**, che consiste nello studio di come l'introduzione di sistemi automatizzati influenzi il comportamento degli utenti, i processi lavorativi e l'ambiente.

Come terminologia specifica abbiamo:

- **Design** = progettazione
- **Progettare** = ideare qualcosa e studiare il modo di attuarla, così da portare un valore nell'ambito in cui viene usata
- **Sistema interattivo** = sistema che risponde ad utenti umani e reagisce alle loro azioni in maniera percepibile ad un utente umano
- **Valutazione** = processo in cui si considera cosa è di valore per l'utente, e si considera tale valore come criterio per capire la qualità di un sistema.

Un concetto chiave che guida queste tematiche è la **meta-metodologia**, ovvero un approccio che va oltre la semplice applicazione di tecniche e metodi tradizionali.

La meta-metodologia si basa sull'uso integrato e riflessivo di vari metodi per affrontare situazioni complesse, adattandosi ai contesti specifici. Non si tratta solo di applicare procedure predefinite, ma di un **processo dinamico** in cui si valutano continuamente i metodi utilizzati, con l'obiettivo di *ottenere soluzioni innovative e che rispondano efficacemente ai bisogni reali degli utenti*. La verifica sul campo, coinvolgendo direttamente

gli utenti, è centrale per garantire che i risultati siano non solo tecnicamente validi, ma anche in grado di offrire un'esperienza d'uso sorprendente e significativa.

Sistema socio-tecnico

Un sistema socio-tecnico è un **insieme di elementi interconnessi** e potenzialmente dipendenti tra loro che, visti dall'esterno, appaiono come un'entità unitaria ma collettiva.

Questi sistemi possiedono caratteristiche autonome e intenzionali, **orientate al raggiungimento di un obiettivo**. Le componenti umane e tecniche sono strettamente legate tra loro, e dalla interazione *emergono fenomeni* che sono spesso imprevedibili.

Questo tipo di sistema è caratterizzato dall'interconnessione stretta e inestricabile tra la componente tecnica e quella sociale, dove **le due interagiscono costantemente**, e in modo tale da generare fenomeni emergenti imprevedibili.

Il termine **fenomeno emergente** si riferisce a quelle proprietà o comportamenti che non sono riconducibili alle singole parti del sistema, ma che nascono solo quando queste interagiscono tra loro.

Queste caratteristiche emergenti non possono essere previste a priori, e, di solito, si possono osservare e studiare solo dopo che il sistema è stato messo in funzione. Il contesto in cui opera il sistema può influenzare queste proprietà, facendole emergere o meno a seconda delle condizioni ambientali.

Proprietà emergenti

Si distinguono in due categorie principali:

- **Funzionali:** legate al corretto funzionamento del sistema una volta che le sue componenti sono state assemblate, e operano come previsto
 - Ad esempio, il sistema può essere progettato per *garantire l'accesso alle informazioni giuste al momento giusto*
- **Non funzionali:** non riguardano direttamente il funzionamento del sistema, ma piuttosto la qualità con cui questo soddisfa i suoi requisiti in un dato contesto.
 - Ad esempio, affidabilità, sicurezza, performance, usabilità

Progettazione

Una buona progettazione **non può mirare a una soluzione ottimale in senso assoluto**. Il pensiero basato sulla meta-metodologia del **STS thinking** riconosce che la ricerca di una soluzione perfettamente ottimale è irrealistica.

Invece, si deve puntare a **trovare compromessi sub-ottimali che riducano al minimo i problemi di integrazione** tra la componente tecnica, e quella umana. Questo approccio richiede di considerare i bisogni degli utenti, che spesso *sono difficili da prevedere o indirizzare correttamente*, e di evitare di depotenziare l'utente, preservando il capitale umano che, altrimenti, sul lungo termine, potrebbe essere irrimediabilmente compromesso.

Quando si introduce una nuova tecnologia in un contesto reale, si richiede inevitabilmente un cambiamento nella modalità di lavoro. Questo processo di cambiamento deve essere gestito **con attenzione** poiché ogni tecnologia, per quanto ben progettata, può portare a conseguenze inattese.

Un esempio classico è l'**effetto Cobra**: una tecnologia introdotta per migliorare un processo può in realtà provocare comportamenti contrari alle aspettative, generando nuovi problemi.

Una delle conseguenze inattese più rilevanti è la **over-reliance**, ovvero l'eccessiva dipendenza dagli strumenti tecnologici. Questa si può manifestare sotto forma di **overdependence** (perdita di autonomia degli utenti, abuso del sistema, mancanza di un piano alternativo), o **overconfidence** (eccessiva fiducia nel fatto che il sistema non fallirà mai, senza considerare i possibili danni derivanti da eventuali errori).

Complacency e automazione bias

Questi sono i rischi più comuni che possono emergere dall'uso eccessivo della tecnologia. La **complacency** si manifesta quando l'utente, fiducioso nel fatto che il sistema funzionerà sempre come previsto, abbassa la propria attenzione nell'utilizzo del sistema.

L'**automation bias**, invece, si verifica quando gli utenti ripongono eccessiva fiducia nei suggerimenti dati da sistemi di supporto alle decisioni, il che può portare a errori di omissione o azione, specialmente quando il sistema non è perfetto.

Se l'utente fosse in grado di discernere sempre quando una macchina ha ragione o torto, non avrebbe bisogno del sistema di supporto decisionale.

Progettazione centrata sull'usabilità

La progettazione di sistemi usabili implica la **creazioni di strumenti pensati per l'uso**, ma la realtà è che questo uso, nella pratica, dipende dall'interazione tra utente e sistema, e non può essere completamente controllato dal progettista.

Non esistono metodologie fisse per progettare sistemi usabili; la progettazione si apprende attraverso l'esperienza e l'interazione con utenti reali. È necessario essere creativi, ma con moderazione, valutare il sistema in fase di sviluppo coinvolgendo gli utenti in ogni fase, e

assumersi la responsabilità delle conseguenze che il sistema avrà nell'ambiente socio-tecnico in cui verrà inserito.

Un aspetto centrale è, ovviamente, l'**usabilità**, ossia la capacità di un sistema di permettere agli utenti di svolgere i propri compiti in modo *efficiente, efficace e soddisfacente*. I sistemi con bassa usabilità presentano problemi di intuitività che rendono difficoltosa l'interazione, soprattutto per utenti non esperti.

Spesso, chi lavora con questi sistemi *sa bene cosa deve fare*, ma fatica a *capire come farlo attraverso l'interfaccia offerta dal sistema*. Questo accade quando i progettisti non pongono abbastanza attenzione alle esigenze reali dell'utente, finendo per creare software complessi e controintuitivi.

Un esempio tipico sono i **sistemi amministrativi**, che tendono ad essere meno usabili rispetto a piattaforme più "user-friendly". In questi casi, il sistema non comunica chiaramente l'obiettivo dell'utente, perché non dobbiamo dimenticare che **il sistema informatico è solo un mezzo, e non l'obiettivo stesso del lavoro**.

Funzionalità e sovraccarico cognitivo

Molte applicazioni moderne presentano più funzioni di quante ne possa effettivamente usare l'utente medio. Questo fenomeno è noto come **sovraccarico funzionale**, può sembrare inizialmente un vantaggio in termini di *copertura delle esigenze*, ma introduce una complessità eccessiva, con conseguenze sia per l'utente che per il progettista.

Ogni funzione aggiunta comporta un **costo** in termini di *sviluppo, mantenimento e usabilità*. Un sistema ideale dovrebbe offrire solo le funzionalità strettamente necessarie, e un piccolo margine.

Blackbox e trasparenza

Un altro concetto cruciale è quello della **trasparenza** del sistema. Gli utenti devono essere sempre consapevoli di cosa il sistema stia facendo e come. Il cosiddetto **effetto blackbox**, dove il funzionamento del sistema è opaco all'utente, crea un senso di perdita di controllo. Questo problema si è ridotto nel tempo grazie all'aumento delle prestazioni dei sistemi, ma resta ancora rilevante in molti contesti. La mancanza di trasparenza può generare frustrazione, ansia e disorientamento, portando l'utente a errori operativi, che non sono sempre imputabili all'utente stesso, ma spesso a una progettazione inadeguata.

Affordance e progettazione intuitiva

Gli oggetti ben progettati devono offrire all'utente degli **indizi visibili sul loro funzionamento**, rendendoli facilmente interpretabili e utilizzabili senza bisogno di istruzioni dettagliate. Un

esempio classico è quello delle **Norman doors**: porte progettate così male che non è chiaro se debbano essere spinte o tirate. Se un sistema o un oggetto necessita di spiegazioni esterne per essere usato correttamente, allora è stato progettato male.

Altro esempio è la **Floyd toilette**: è un oggetto che funziona, ma che è assurdamente complicato, e rappresenta la tendenza del progettista a **concepire oggetti o sistemi che non invitano all'uso** sulla base di elementi visuali o indichi azioni chiare, affidandosi solo alla memoria, all'esperienza, o all'intuizione, ed incontrando spesso il disorientamento o fraintendimento dell'utente.

Ogni elemento progettato da un essere umano è un **artefatto**, ovvero un prodotto culturale che riflette le intenzioni e la competenza del progettista. Mentre un **manufatto** è ciò che è proprio prodotto da un essere umano.



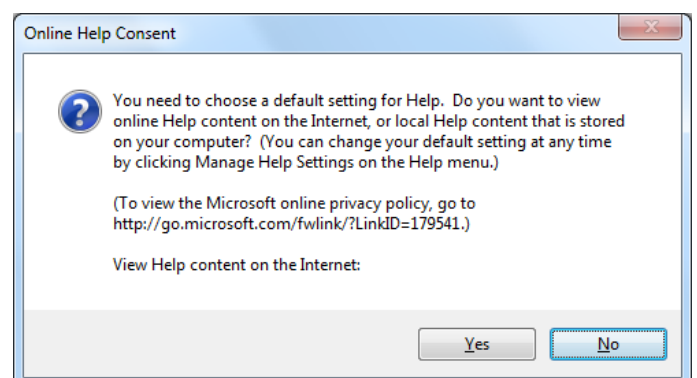
Un esempio

Per acquistare il biglietto è necessario:

1. Selezionare la tariffa (ridotta o intera) premendo il primo pulsante blu
2. Selezionare la zona di viaggio premendo il secondo pulsante blu
3. Inserire le monete (o la carta di credito – vedi passo 3)
4. Confermare premendo il pulsante verde
5. Attendere circa 20 secondi (!!!)
6. Ritirare il biglietto dalla fessura a sinistra.

Problemi di terminologia e consistenza

Un altro ostacolo comune è l'uso di una **terminologia complessa o incoerente**. I sistemi spesso impiegano termini che non risultano chiari agli utenti, o utilizzano la stessa parola per azioni diverse, generando confusione. Questo non solo complica l'interazione, ma riduce la fiducia dell'utente nel sistema.



Qualità dell'interfaccia

Occuparsi della qualità dell'interfaccia è fondamentale perché una buona interfaccia consente all'utente di **svolgere il proprio compito senza rendersi conto di usare un computer**.

La bellezza e l'efficacia percepita di un'interfaccia migliorano l'esperienza, portando a un **effetto alone** in cui l'utente non si rende neanche conto che sta usando un computer, ma il computer diventa semplicemente il mezzo per cui viene eseguito il lavoro. Questo produce effetti misurabili, come un *aumento della produttività, riduzione del turnover e della frustrazione, minori costi operativi e una diminuzione degli errori tecnologici*.

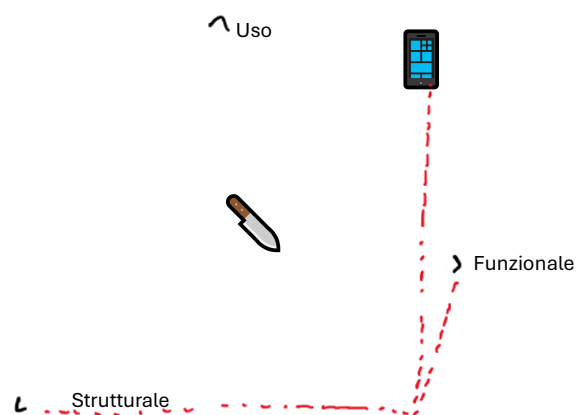
Uno degli errori più frequenti di questo tipo è l'errore di **giustapposizione**, che si verifica quando l'utente *seleziona per errore una voce vicina a quella desiderata nell'interfaccia*, come in un menu a tendina.

Semplificazione

Per ogni oggetto nel nostro mondo, possiamo dividere la complessità di esso in tre aree: **complessità d'uso** (se è difficile da usare), **complessità funzionale** (se esegue un numero alto di funzioni), **complessità strutturale** (se è costituito da tanti componenti).

Sarebbe importante ovviamente avere una bassa complessità d'uso e bassa complessità strutturale, come esempio, ma non sempre migliorando una si migliorano automaticamente anche le altre; talvolta, si tende a migliorarne una, e **peggiore le altre**.

Si vogliono abbassare per avere più utenti ad usare il proprio prodotto, e quindi colmare il *digital divide*, ovvero quel divario dovuto alle competenze digitali.



Ogni oggetto che abbiamo nella vita reale può essere misurato attraverso queste tre complessità: per esempio, un coltello ha bassa complessità d'uso, funzionale e strutturale.

Principi e standard

Secondo la norma **ISO 9241**, un **sistema interattivo** è la combinazione di componenti hardware e software che, ricevendo input dall'utente, forniscono un output utile per completare un compito.

L'**interfaccia**, invece, è definita come l'insieme di componenti, software o hardware, che permette all'utente di interagire con il sistema. Essa deve essere progettata in modo che le informazioni e i comandi siano chiari e accessibili, garantendo un'interazione efficace.

L'utente non vede la parte di back-end, quindi per lui **l'interfaccia è il sistema interattivo**. L'elemento più comune tra i due è il concetto di effettuazione di un compito, con il fine di **raggiungere un obiettivo** ben definito.

Un principio fondamentale è la **suddivisione dei compiti in sottocompiti più gestibili**, ciascuno dei quali deve essere facilmente riconoscibile e realizzabile dall'utente

Heat maps e neuromarketing

Un approccio avanzato nella progettazione dell'interazione è l'uso delle **heat maps**, strumenti che permettono di visualizzare dove gli utenti concentrano maggiormente la loro attenzione all'interno di un'interfaccia. Questa tecnica spesso utilizzata nel **neuromarketing** è impiegata per ottimizzare l'interfaccia affinché catturi e mantenga l'attenzione dell'utente nei punti critici, come nelle pagine di e-commerce, dove l'obiettivo è massimizzare le conversioni.



Effetti psicologici nell'interazione

Nell'interazione uomo-macchina, la **prima impressione** è spesso decisiva. Questo fenomeno, noto come **Halo Effect**, si verifica quando l'utente associa la qualità dell'interfaccia alla qualità dell'intero sistema. Se l'interfaccia è ben progettata, l'utente tenderà a giudicare positivamente anche le altre componenti del sistema.

Altro effetto è il **disorientamento**: se hai offerto agli utenti la stessa funzionalità, nella stessa posizione o stile, per tanti anni, allora il cambiamento può generare confusione e frustrazione, poiché la memoria visiva e abitudini precedenti vengono interrotte.

Vincoli

I vincoli sono **elementi fondamentali per guidare l'utente e limitare le sue azioni a quelle previste dal sistema**.

Si distinguono in:

- **Vincoli strutturali (passivi)**: limitano l'interazione tramite la configurazione fisica o logica del sistema (es. un modulo che prevede solo tre campi per l'inserimento dati impedisce all'utente di inserire informazioni non richieste)

- **Vincoli funzionali (funzionali):** controlli attivi che verificano il corretto svolgimento di un'azione (es. un form che non permette l'invio finché tutti i campi obbligatori non sono compilati correttamente).

Questi vincoli, se progettati bene, migliorano l'usabilità del sistema poiché riducono la possibilità di errori da parte dell'utente, facilitano il flusso di lavoro, e garantiscono che l'interazione avvenga in modo strutturato e corretto.

Workflow

Si riferisce alla **sequenza di funzioni che il sistema informatico abilita per consentire il completamento di un lavoro.**

La definizione ufficiale è:

Qualsiasi azione relativa all'esecuzione di un processo o di un compito, supportati dal sistema informatico, che non è prevista o descritta nei manuali di uso del sistema informatico e/o nei manuali che descrivono tale processo o procedura, e che può bypassare l'uso del sistema, o piegarlo ai propri fini.

A volte si potrebbe definire più accuratamente come *workthrough*, che sottolinea come, quando un workflow è troppo rigido o poco intuitivo, gli utenti possono trovare il modo di *sfondare* le barriere del sistema, cercando strade non previste dai progettisti. Questo evidenzia l'importanza di progettare sistemi che rispondano davvero ai bisogni degli utenti, invece di costringerli ad adattarsi.

Nel contesto **applicativo**, si tratta di flussi di lavoro abilitati direttamente da un'applicazione software. Un workflow troppo complesso, o mal progettato, può diventare un'arma a doppio taglio, portando gli utenti a creare soluzioni alternative, chiamate *workarounds*, per aggirare le limitazioni del sistema.

Sentiero del desiderio

È il percorso che **l'utente preferirebbe seguire per eseguire un compito**, piuttosto che quello predefinito dal progettista. Questo concetto mette in evidenza come gli utenti spesso trovano soluzioni alternative che rispecchiano meglio le loro esigenze rispetto alle procedure ufficiali.

Queste soluzioni informali possono essere azioni *non documentate*, o *non previste* nei manuali, ma che risultano più efficienti, o semplici da adottare.

Key informants

Durante lo sviluppo di un sistema, è fondamentale interagire con i **key informants**; sono esperte dell'ambiente lavorativo in cui verrà implementato il sistema, ed offrono informazioni preziose per adattare il software alle reali esigenze degli utenti e all'ambiente di utilizzo. Includerli aiuta a evitare *disallineamenti tra il sistema e il contesto operativo*.

Affordance

Una qualità o proprietà di un oggetto che definisce i suoi possibili usi, o rende evidente all'utente come utilizzarlo. Ne abbiamo di diversi tipi, e non sono mutualmente esclusive:

- **Cognitive:** aiutano l'utente ad effettuare azioni cognitive (pensare, decidere, imparare, ecc....)
 - o **Es.** un'etichetta che descrive cosa fa un determinato oggetto
- **Fisiche:** aiutano l'utente con le sue azioni fisiche (cliccare, toccare, puntare, ecc...). Associate con la caratteristica dell'operabilità
 - o **Es.** un bottone abbastanza largo in modo che l'utente possa premerlo facilmente
- **Sensoriali:** aiutano l'utente con le azioni sensoriali (vedere, udire, sentire gusti e profumi)
 - o **Es.** il font di un'etichetta abbastanza grande da essere individuata facilmente
- **Funzionali:** aiutano l'utente in lavori reali affinché l'utente compia azioni che hanno effetti nel mondo
 - o **Es.** la capacità interna del sistema di ordinare una serie di numeri
- **Emozionali:** aiutano, supportano e facilitano la comunicazione emotiva. Lo scopo è quello di suscitare emozioni nell'utente
 - o **Es.** le emoji
- **Sociali:** proprietà di un oggetto che invitano e facilitano azioni collaborative, sociali
 - o **Es.** icone che permettono di creare un messaggio vocale ecc.

Skeuomorphism

È un approccio di design in cui gli oggetti digitali **imitano elementi del mondo fisico**. Il motivo dietro questa scelta è *rendere più comprensibili e familiari le nuove tecnologie*, facilitando l'adozione da parte dell'utente grazie all'associazione con oggetti già noti.

Tuttavia, lo skeuomorphismo può anche diventare una limitazione quando l'interfaccia digitale evolve oltre la necessità di imitare oggetti fisici. Molti designer oggi preferiscono un approccio più minimalista e funzionale, come per esempio il *flat design* o il *neumorphism*.



User-created affordances

Sono introdotte dall'utente stesso che si appropria di uno strumento al quale “manca qualcosa” per comprenderne a pieno il funzionamento.

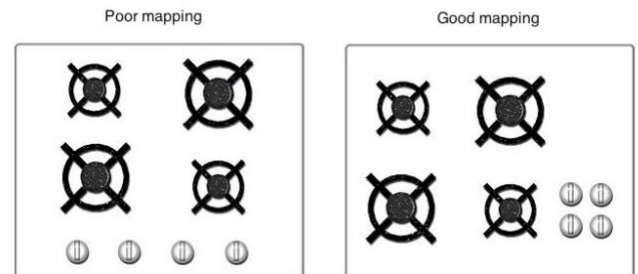
È un segnale al progettista che, secondo l'utente, manca qualche indicazione per l'uso ottimale dello strumento.



Mapping

È una caratteristica dell'affordance, come se fosse un'affordance analogica (lavora per analogia) e **rappresenta la relazione tra un controllo** (e l'azione che esso afforda), e l'effetto che tale azione avrà nel mondo reale/applicativo.

Non è possibile avere un mapping senza un'affordance, ma si può avere un'affordance senza un mapping. Ogni volta che un'etichetta sembra necessaria, sarebbe meglio pensare ad un nuovo design.



Abbiamo vari tipi:

- **Mapping arbitrario:** una relazione del tutto immaginata dal progettista, e che l'utente adotta solo per prove ed errori, e poi **impara a memoria**, senza riferimenti culturali o indicazioni fisiche
- **Mapping convenzionale:** la relazione tra controllo ed effetto è stabile, quasi ovvia, ma legata a **delle convenzioni**.
 - o **Es.** il freno nelle macchine; è basato su una convenzione
- **Mapping naturale:** la relazione è meno immediata che nel caso diretto, ma ci viene appunto naturale immaginare la relazione esistente tra controllo ed effetto (es. il puntatore a forma di mano con l'indice prominente che ci suggerisce di fare click)
- **Mapping diretto:** vi è una relazione fisica, visibile, spesso legata alla posizione tra controllo, che afforda un'azione, e l'effettore controllato che produce un effetto

Designing by affordance

Dobbiamo seguire quattro principi cardine:

1. Seguire le maggiori **convenzioni già stabilite** per immagini e azioni, e adottare il **mapping naturale** ove possibile
2. Se appropriato, utilizzare delle **scritte in aggiunta ad icone e grafiche**, ma non esagerare
3. Utilizzare **metafore riconoscibili**
4. Essere consistenti e coerenti nell'uso del modello concettuale dietro al design

Usabilità

Misura il **grado con cui un sistema consente agli utenti di raggiungere specifici obiettivi** con *efficacia* (la capacità del sistema di aiutare l'utente a raggiungere i propri obiettivi), *efficienza* (l'ottimizzazione delle risorse necessarie per raggiungere un obiettivo), e *soddisfazione* (il piacere e il comfort percepiti durante l'utilizzo del sistema).

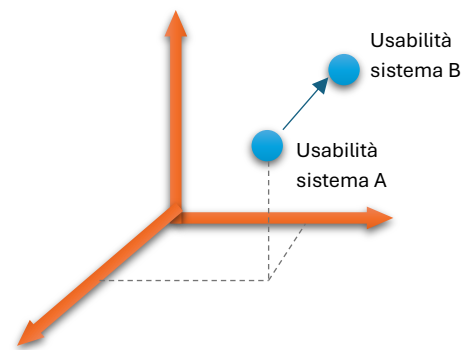
L'esperienza utente, **UX**, amplia il concetto di usabilità, includendo aspetti emozionali, esperienziali e valoriali legati all'interazione con il sistema; considera il **contesto d'uso**, che può variare da situazioni di emergenza a condizioni di distrazione, influenzando la percezione complessiva del sistema.

Metriche e indicatori

La valutazione dell'usabilità si basa su metriche quantitative e qualitative:

- **Metriche:** misurazioni dirette del grado di possesso di un attributo da parte del sistema. Comprende la misurazione quantitativa del grado di possesso di uno specifico attributo da parte di un sistema, e specifica le *procedure*, *modalità*, e *regole di misurazione*.
- **Indicatori:** utilizzati per valutare attributi più complessi e meno tangibili, come la soddisfazione

Una differenza particolarmente significativa è quella che fa differenza.



Estensioni dell'usabilità

Tra le caratteristiche aggiuntive abbiamo la **robustezza** (ISO 9241, il sistema deve indurre un basso tasso di errori in diverse situazioni), **apprendibilità** (deve essere facile da imparare), **ricordabilità** (gli utenti devono essere in grado di ricordare facilmente come usarlo, anche dopo lunghi periodi di inattività).

Ovviamente, è importante bilanciare queste dimensioni in base al contesto d'uso: *per esempio, un sistema per utenti esperti può essere più complesso da imparare, purché garantisca un'elevata soddisfazione a lungo termine*. Questo perché può creare fidelizzazione; mentre altre volte, conviene fare sistemi che piacciono da subito.

Ci può essere anche un trade-off tra robustezza ed efficienza: il desiderio di evitare errori all'utente può portare a progettare un **numero eccessivo di vincoli**, e quindi questi trade-off si risolvono in base al contesto d'uso dell'utente, quali dimensioni di usabilità sono più importanti per un certo sistema.

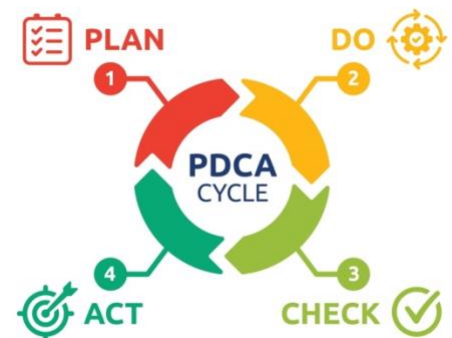
Valutazione dell'usabilità

È il processo attraverso cui si analizza quanto un sistema o un'interfaccia siano efficaci, efficienti e soddisfacenti per gli utenti. L'obiettivo è **identificare problemi o ostacoli che potrebbero compromettere l'esperienza dell'utente** durante l'interazione con il sistema, così da migliorare la facilità d'uso e l'intuitività dell'interfaccia.

Esistono diversi tipi, e tra questi c'è la **valutazione qualitativa**, che si concentra sull'osservazione diretta e sulle opinioni di esperti o utenti reali. L'analisi non si basa su dati statistici, ma su *insight* e *giudizi* ottenuti da utenti esperti nel compito o da esperti di usabilità. Questa metodologia può essere applicata in qualsiasi fase dello sviluppo (persino prototipi o mockup), e il suo scopo è *raccogliere una lista dettagliata di possibili problematiche di usabilità*, simulando l'uso reale del sistema per capire come un utente potrebbe interagirci.

Un concetto importante è il **ciclo PDCA**, una metodologia di miglioramento continuo utilizzata in vari contesti. Si articola in quattro fasi:

- **Plan:** identificazione degli obiettivi e strategie per migliorare un processo o sistema
- **Do:** soluzioni pianificate vengono implementate in una scala ridotta per testare la loro efficacia
- **Check:** fase di analisi dei risultati ottenuti rispetto agli obiettivi prefissati
- **Act:** sulla base delle verifiche, si adattano e standardizzano i cambiamenti i cambiamenti se sono efficaci, o si ritorna alla fase di pianificazione per nuove modifiche.



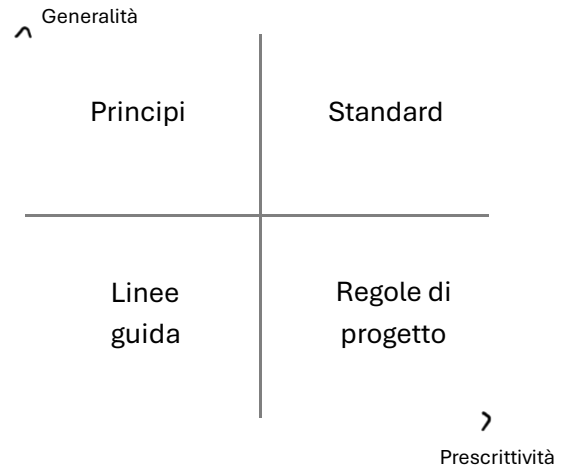
È un processo **empirico, sistematico e concreto**, basato sull'osservazione di utenti reali che utilizzano il sistema in un contesto specifico. Ci sono vari tipi:

Formativa	Supporta lo sviluppo identificando i problemi di usabilità durante la progettazione (euristica)
Riassuntiva	Fornisce un giudizio sul prodotto dopo il completamento dello sviluppo; volta a ottenere benchmark quantitativi o comparare soluzioni alternative
Qualitativa	Basata sulle opinioni degli utenti
Quantitativa	Basata su misurazioni dirette delle prestazioni del sistema
Absoluta	L'usabilità è misurata con una certa percentuale o valore
Comparativa	Si confrontano due soluzioni diverse e si dice qual è quella più usabile

Principi e standard

All'interno della valutazione dell'usabilità, abbiamo a che fare con i seguenti concetti:

- **Principi:** indicazioni **generali** per la progettazione di interfacce utente usabili, basate su evidenza scientifica, o generale consenso.
 - o **Euristica:** sono dei principi; insiemi di concetti, riferimenti, strategie, che sono associate con una buona pratica professionale
 - o **Valutazione:** è svolta alla luce di un determinato insieme di euristiche *ben definite, note e riferite*, per identificare soluzioni di design che si conformano/violano una o più euristiche di tale insieme
- **Standard:** insieme di regole da applicare nel progetto di una classe di sistemi. Possono essere *vincolanti*.
 - o **Due tipi:** de facto (adottati da un vasto pubblico), de iure (rilasciati da una società)
- **Linee guida e design patterns:** insieme di raccomandazioni per il progetto dell'interfaccia utente per una particolare classe di sistemi, espresse in modo generale ma meno astratte, di solito con esempi e motivazioni. **Non sono mai vincolanti**, il progettista decide sull'opportunità di applicarle caso per caso
 - o Sono utili perché si riutilizzano buone pratiche per non dover reinventare cose già esistenti
 - o Diffondono gli standard de facto, di solito partono da linee guida e design patterns
- **Regole di progetto:** insieme di regole da applicare nel progetto, e come tali, sono vincolanti.



Uso dei principi

L'uso dei principi, anziché delle linee guida, è di solito volto a favorire la creatività nella progettazione delle interfacce. I principi sono sempre validi, e rappresentano direttive fondamentali da seguire, mentre le linee guida e i pattern di design che risultato efficaci sono applicati laddove appropriati.

In termini di **autorevolezza**, possiamo considerare diversi livelli nelle affermazioni relative all'usabilità:

- **Supporto empirico:** completamente basate su dati di ricerca e risultati verificati

- **Buone pratiche:** documentate e generalmente accettate nel settore
- **Consenso degli esperti:** supportate da un numero significativo di professionisti, anche se non universalmente accettate
- **Opinione individuale:** frutto dell'esperienza o della prospettiva di singoli professionisti

IXD check-list

La IxD check-list non è una check-list ufficiale fatta da un ente, ma è una check-list per l'Interaction Design fatta da una serie di designer. Si basa su **euristiche pratiche**, ossia tecniche mnemoniche che facilitano la progettazione efficace. Ciascuna sezione è collegata a una specifica euristica (di seguito riporto qualche esempio in corsivo):

- **Affordance**
 - o *I controlli devono corrispondere in modo logico ai risultati che producono.*
 - o *La funzione di ogni controllo deve essere facilmente identificabile a colpo d'occhio, e le icone simboliche devono essere chiare anche per i nuovi utenti*
- **Feedback**
 - o *Gli stati di ogni elemento devono essere mostrati chiaramente.*
 - o *L'utente deve poter comprendere facilmente lo stato attuale del sistema e ricevere un feedback quando un compito è completato*
- **Semplicità**
 - o *Ogni passo in un flusso sequenziale deve essere evidente per l'utente, evitando complessità non necessarie*
- **Struttura**
 - o *La profondità dell'architettura informativa deve essere coerente in tutto il sistema*
- **Consistenza, tolleranza agli errori, accessibilità**

Principi di ISO 9241-110

Lo standard ISO 9241-110 introduce sette principi per il dialogo uomo-macchina, che aiutano a garantire un'interazione efficiente e naturale. Di seguito elencati:

- **Adeguatezza al compito:** l'interfaccia deve essere adatta alle esigenze specifiche del compito
- **Autodescrizione:** il sistema dovrebbe fornire informazioni chiare sul suo utilizzo
- **Conformità alle aspettative dell'utente:** l'interazione deve rispettare le aspettative e i modelli mentali dell'utente
- **Adeguatezza all'apprendimento:** facilita l'apprendimento del suo utilizzo
- **Controllabilità:** l'utente mantiene il controllo dell'interazione
- **Tolleranza verso gli errori:** l'interfaccia consente di prevenire e correggere errori

- **Adeguatezza alla personalizzazione:** l'interfaccia si adatta alle preferenze dell'utente

Euristiche di Nielsen

Si articolano in tre categorie principali:

- **Percezione**
 - Visualizzare lo stato del sistema in modo visibile (*feedback*)
 - Utilizzare un linguaggio comprensibile e vicino alla realtà dell'utente
 - Fornire controllo e libertà, con opzioni di uscita ben visibili
- **Cognizione**
 - Mantenere la coerenza nell'applicazione e nell'ambiente d'uso
 - Favorire il riconoscimento rispetto alla memorizzazione
 - Assicurare flessibilità ed efficienza d'uso (*scorciatoie ed acceleratori*)
 - Visualizzare solo le informazioni essenziali, secondo principi di *design minimalista*
- **Errori**
 - Prevenire gli errori
 - Permettere all'utente di correggere gli errori e non solo di rilevarli
 - Fornire documentazione e supporto quando necessario

Feedback del sistema



Un buon sistema deve informare costantemente l'utente sullo **stato delle operazioni e sull'interpretazione degli input**, permettendogli di capire l'effetto delle azioni passate, e le azioni future disponibili.

Non deve essere limitato ai casi di errore ma fornito in ogni iterazione significativa. È possibile che sia concreto, evitando generalizzazioni. Ogni feedback ha la propria persistenza, che varia in base alla durata dell'operazione:

- **Tempi di risposta brevi <0,1 sec** il feedback non è necessario
- **Operazioni da 1-3 secondi** è consigliato un indicatore di attività
- **Elaborazioni più lunghe** si utilizza una barra di progresso

Adeguamento al mondo reale

Un'interfaccia efficace permette agli utenti di **utilizzare la propria esperienza pregressa nell'interazione con il sistema**. Si adottano notazioni e convenzioni di denominazione familiari agli utenti, e, se possibile, si utilizza la lingua nativa dell'utente.

Anche le icone devono rispettare le convenzioni culturali per garantire una corretta percezione dell'affordance.

Controllo utente e libertà

Il sistema deve offrire all'utente la possibilità di **accedere facilmente** alle sue funzionalità, scegliendo tra diverse alternative e modalità operative. L'utente **non deve sentirsi intrappolato o costretto**, ma deve percepire di avere il controllo del dialogo con il sistema. È essenziale che vengano sempre offerte vie d'uscita chiari e semplici, così da poter uscire agevolmente da situazioni indesiderate.

Un aspetto è la **possibilità di annullare comandi o azioni** precedenti; deve consentire agli utenti di interrompere o cancellare operazioni che richiedono tempi prolungati, evitando di farli sentire bloccati o frustrati.

Altro comando fondamentale è **undo**; deve consentire loro, per quanto possibile, di ripristinare uno stato precedente, poiché gli errori non sempre vengono identificati immediatamente. Ovviamente, abbiamo anche il **ripristina**. Abbiamo due tipi diversi:

- **Flip undo**: consente di annullare l'effetto dell'ultimo undo
- **Multi-step undo**: permette di annullare più operazioni consecutive, tornando indietro di un numero indefinito di passi, o limitato dalla quantità di informazioni disponibili sulla cronologia delle interazioni

Assicurare consistenza

È un principio fondamentale per favorire la familiarità degli utenti con il sistema, incoraggiando l'esplorazione e riducendo il rischio di errori. Quando uno stesso comando o una stessa azione producono sempre lo stesso effetto, gli utenti sviluppano maggiore confidenza nell'interfaccia. Situazioni simili o scopi analoghi devono essere affrontati con approcci simili, per garantire un'esperienza intuitiva e prevedibile.

Layout grafico

La posizione degli elementi visivi e interattivi deve rimanere **coerente** in tutte le schermate, evitando confusione, e rendendo più agevole l'orientamento dell'utente. Un'interfaccia ben strutturata, con una *distribuzione omogenea degli elementi*, aumenta l'efficienza e migliora l'esperienza complessiva.

Messaggi

Anche il linguaggio e il tono dei messaggi devono essere consistenti. I messaggi del sistema *devono attirare l'attenzione dell'utente solo su ciò che è realmente importante*, evitando informazioni superflue o poco rilevanti.

A/B testing

È una tecnica essenziale per **confrontare due soluzioni alternative** e **identificare quella più efficace** in termini di usabilità, attrattività o altri parametri chiave.

Durante il test, due varianti vengono mostrate a gruppi diversi di utenti, e i risultati vengono analizzati per prendere decisioni informate.

Un test binomiale è stato condotto per confrontare il tasso di conversione della soluzione A (*proporzioneA ± intervallo di confidenza*) con quello della soluzione B. I risultati hanno mostrato una differenza non significativa tra i due tassi di conversione (*differenza-proporzione ± intervallo di confidenza*). Un'analisi statistica, ad esempio tramite il test del chi-quadro (χ^2 = valore calcolato), con un p-value pari a p = valore esatto, ha confermato che gli utenti non mostrano una preferenza significativa per una soluzione rispetto all'altra.

Quando il testing A/B identifica una soluzione chiaramente migliore, questa viene adottata e applicata al sistema. Se le differenze tra le soluzioni sono trascurabili però, la scelta può dipendere da altri fattori, come i costi di implementazioni o le preferenze del team di sviluppo.

Riconoscimento piuttosto che ricordo

Quando un sistema richiede all'utente di fornire un dato, è importante che **il formato del dato sia descritto chiaramente**, magari con un esempio esplicativo, o che venga fornito un *valore di default*.

Così, l'utente non deve necessariamente conoscere l'intervallo dei valori accettabili in una casella di input; sarà compito del sistema di fornire tale implementazione in modo trasparente. Mira quindi a **ridurre il carico cognitivo** sull'utente, e a semplificare l'interazione con il sistema.

Assicurare flessibilità ed efficienza d'uso

Un sistema deve garantire che **gli utenti esperti possano eseguire rapidamente le operazioni più frequenti**, riducendo al minimo le azioni necessarie. Questo può essere ottenuto, ad esempio, tramite l'uso di *shortcuts* e la possibilità di eseguire le stesse operazioni in modi diversi, sebbene questo possa sembrare un problema di consistenza.

Strumenti come i *segnalibri*, *breadcrumb trailer*, *cronologia dell'interazione* e la *definizione di template* possono semplificare notevolmente l'esperienza dell'utente.

Visualizzare solo le informazioni necessarie

Ogni informazione presente nell'interfaccia rappresenta una potenziale distrazione per l'utente, specialmente *per chi è meno esperto e non sa subito quali dettagli siano veramente rilevanti*. Infatti, un utente di questo genere, per trovare quelle utili, legge tutte le istruzioni. Per questo motivo, è essenziale che **l'interfaccia sia semplice, chiara e ben strutturata**, mostrando solo le informazioni necessarie nel momento giusto, nel formato giusto e nel posto giusto. Bisogna presentare le informazioni che l'utente vuole, quando la vuole, come e dove la vuole, senza niente non rilevante o fuorviante.

Non bisogna sovraccaricare l'utente con **troppe opzioni**. Ogni funzionalità aggiunta va valutata attentamente, per comprendere se sia veramente utile e necessaria, poiché un'interfaccia che offre troppe opzioni rischia di confondere l'utente, ed aumentare la possibilità di errore, anche perché se ci sono più funzionalità, vuol dire che ci sono anche più cose da ricordare. Gli applicativi per persone poco esperte sono i più semplici.

Il concetto di *low use* suggerisce che l'aggiunta di troppe funzionalità non porta necessariamente a una maggiore soddisfazione dell'utente; al contrario, gli utenti *tendono ad utilizzare solo un numero limitato di funzioni*, quelle che soddisfano le loro necessità immediate.

Per misurare questa tendenza, si può usare l'**Use Index**, che fornisce un'indicazione di quante funzionalità vengano effettivamente utilizzate. Un valore basso di UI ($\leq 0,5$) indica che *l'utente non sta fruttando appieno il potenziale dell'applicazione*.

$$UI = ((\text{numero di funzioni usate spesso} * 1) + (\text{numero di funzioni usate a volte} * 0,5) + (\text{numero di funzioni mai usate} * 0)) / \text{numero di funzioni considerate}.$$

Importanza della grafica, layout e colore

Il design grafico dell'interfaccia deve riflettere la **gerarchia delle informazioni**: le informazioni più importanti devono essere visibili per prime. Elementi lampeggianti, o scritte in maiuscolo, dovrebbero essere utilizzati solo *quando strettamente necessario*, poiché questi elementi rallentano la velocità di lettura, che per una media di lettura si aggira tra le 200 e le 300 parole al minuto.

Per quanto riguarda l'uso del colore, è consigliabile *non superare i 5-7 colori distinti per gli elementi dell'interfaccia* e scegliere accoppiamenti cromatici sensati: è importante evitare colori brillanti per lo sfondo, poiché possono compromettere la leggibilità e l'estetica generale dell'interfaccia.

Prevenire gli errori

La progettazione di un sistema deve focalizzarsi sulla prevenzione degli errori. Le tecniche di prevenzione degli errori comprendono:

- **Diversificazione delle azioni:** evitare modalità diverse che alterano la funzionalità dei bottoni a seconda della modalità attiva dell'interfaccia
- **Amodalità:** non avere modi (e.g. modalità dell'interfaccia dove i bottoni hanno diverse funzionalità in base alla modalità in cui siamo)
- **Funzioni vincolate:** alcune funzioni potrebbero non essere disponibili in determinati contesti, per evitare azioni errate da parte dell'utente. È utile renderle visibili ma non attive
- **Avvisi e conferme:** utilizzare avvisi a vari livelli di criticità per segnalare all'utente la gravità dell'errore. Un errore inoltre deve essere notificato in tre diverse sezioni:
 - o **Alert:** indica che qualcosa non va
 - o **Identify:** specifica qual è l'errore
 - o **Direct:** fornisce indicazioni su cosa fare per risolvere il problema
- **Valori di default sicuri:** i valori predefiniti dovrebbero essere scelti in modo che, anche se l'utente conferma erroneamente, non comportino conseguenze gravi
- **Bypass sicuri:** permette di bypassare alcune funzioni in modo sicuro, garantendo che il processo principale non venga compromesso. Se la modalità principale è inoffensiva, e la modalità alternativa non è tanto visibile, la seconda modalità potrebbe anche permettere di bypassare i vincoli, però questo è pericoloso.

Permettere all'utente di correggere gli errori

Bisogna permettere all'utente di correggere facilmente gli errori. Questo può essere fatto offrendo un'opzione di **annullamento**, e messaggi di errore chiari e utili. I messaggi devono essere formulati in linguaggio semplice, *senza tecnicismi inutili*, e fornire indicazioni precise su come risolvere il problema. È importante che i messaggi siano anche gentili, e non penalizzino l'utente per l'errore commesso. La cortesia e la chiarezza dei messaggi sono fondamentali per ridurre l'ansia dell'utente, ed aiutarlo a rimanere tranquillo. Quindi, è importante anche la **recoverability**.

Per quanto riguarda i messaggi, bisogna **spiegare cosa è successo e perché**, in maniera concisa e comprensibile; **suggerire il prossimo passo**; **mantenere un tono rispettoso**, poiché la colpa non deve cadere sull'utente.

Fornire aiuto e documentazione

Nonostante un sistema ideale non richiederebbe aiuto o documentazione, tale perfezione è raramente raggiungibile. È quindi essenziale progettare strumenti di supporto per gli utenti, considerando diverse forme di aiuto:

- **Suggerimenti contestuali:** tooltip e notifiche leggere sono *altamente apprezzati*, poiché forniscono informazioni senza interrompere il flusso di lavoro
- **Guide visive:** video e animazioni possono spiegare processi in modo chiaro ed efficace
- **Manuali dettagliati,** necessari per gli utenti avanzati che vogliono esplorare funzionalità più approfondite.

Un buon sistema di supporto dovrebbe essere **minimalista**, ovvero offrire solo le informazioni necessarie in quel momento, e **contestuale**, ovvero dovrebbe adattarsi alla posizione e alle azioni dell'utente all'interno del sistema.

Tecniche

Per verificare l'usabilità, possiamo usare diverse tecniche:

- **Valutazione euristica:** tecnica sistematica di ispezione predittiva. Un gruppo di valutatori analizza l'interfaccia identificando problemi di usabilità in relazione a specifiche euristiche. Questo metodo è **predittivo**, in quanto identifica potenziali problemi *prima* dell'uso da parte degli utenti finali
- **Protocol analysis:** l'utente pensa ad alta voce durante l'interazione, fornendo così un resoconto delle proprie impressioni e difficoltà. Le tecniche di protocol analysis possono essere:
 - **Olistiche:** non guidate, basate su una navigazione libera che permette una visione d'insieme del sistema
 - **Orientate ai task:** basate su obiettivi specifici per valutare come l'utente compie determinate azioni

Procedura

Di seguito, un esempio di procedura che dovremmo anche usare nell'esame:

1. **Scelta di un'applicazione:** il primo passo consiste nel selezionare un'applicazione su cui basare la valutazione
2. **Registrazione e consenso:** la procedura dovrà essere registrata, e quindi è necessario far firmare il consenso a colui che sta testando l'applicazione. Inoltre, la registrazione serve per poi andare a rivedere meglio come l'utente ha reagito (l'osservatore passivo, infatti, si segnerà i tempi di ciascuna cosa annotata).
3. **Definizione dei task:** devono essere ideati 2 o 3 task realistici e praticabili, evitando di fornire suggerimenti specifici o descrizioni passo-passo. Di seguito due esempi:
 - **Task realistico:** esplorare prodotti e fare un acquisto
 - **Cattivo esempio:** *acquista un paio di scarpe da corsa Nike arancioni*
 - **Buon esempio:** *compra un paio di scarpe sotto i 40€*

- **Task eseguibili:** cercare un film e gli orari
 - **Cattivo esempio:** vuoi vedere un film domenica pomeriggio; vai su Fandango e indica dove cliccheresti per primo
 - **Buon esempio:** usa Fandango per trovare un film da vedere domenica pomeriggio
 - **Task senza avere un elenco di compiti:** consultare i voti online
 - **Cattivo esempio:** visualizza i voti dell'esame IUM. Vai sul sito del corso, fai il login, mostra dove cliccheresti per vedere il voto
 - **Buon esempio:** cerca il risultato dell'esame IUM sul sito del corso
4. **Esplorazione dell'applicazione:** ogni partecipante esplora l'applicazione, e redige un elenco di problemi riconducendoli alla violazione di specifiche euristiche
 5. **Valutazione con esperti:** l'applicazione viene presentata a **3 utenti esperti**, che esplorano l'interfaccia per **3-4 minuti** con una valutazione olistica. Viene chiesto di **pensare ad alta voce**; uno degli osservatori supporta il partecipante, un altro annota tempi e particolari espressioni verbali
 6. **Presentazione dei task:** un task precedentemente definito viene presentato attraverso una *spiegazione* o un *video*
 7. **Esecuzione del task:** il task viene eseguito dagli utenti in 3-4 minuti, seguito da una valutazione task-based ed analisi di protocollo
 8. **Sintesi dei problemi di usabilità:** collaborativamente, si compila un elenco dei problemi di usabilità identificati
 9. **Unificazione delle liste:** le liste dei problemi vengono consolidate ed organizzate, assegnando una scala di priorità
 10. **Questionario di prioritizzazione:** un questionario aiuta a definire le priorità dei problemi trovati, associandoli ad euristiche specifiche

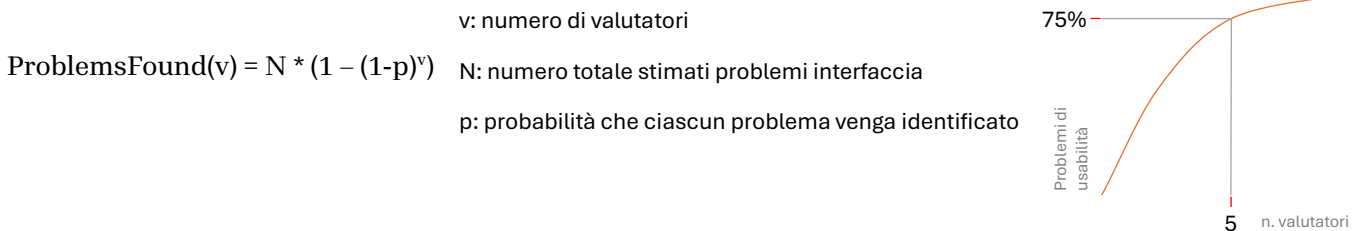
Osservatore

L'osservatore può avere due modalità operative:

1. **Osserva passivamente** mentre i membri del team stesso esplorano l'applicazione, annotando eventuali difficoltà senza intervenire (di solito, espressioni facciali, commenti, e a che tempo).
2. Svolge un **ruolo attivo** quando l'applicazione viene testata da altri utenti. Un osservatore segue da vicino il principiante, aiutandolo nel completamento del task in caso di necessità, mentre un *secondo osservatore* annota dettagli cruciali (reazioni, commenti, etc)

Curva di apprendimento

Con l'aumentare del numero di valutatori, cresce anche il numero di problemi identificati, fino a raggiungere una fase di saturazione. Questo può essere descritto con:



Questo fenomeno dimostra l'importanza di un team di valutatori adeguatamente numeroso, per identificare una gamma più ampia di problemi, pur sottolineando che, oltre un certo numero, i contributi marginali diminuiscono.

Prioritizzazione dei problemi

Non tutti i problemi sono importanti e devono essere inclusi nell'analisi; per valutare quali sono da includere e quali no, è importante attribuire a ciascun problema una priorità, all'interno di una tabella:

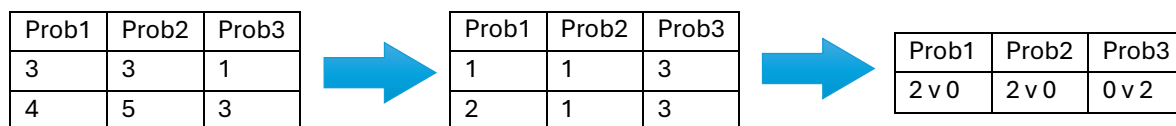
0. Non è considerato un problema
1. Problema **cosmetico**, risolvibile solo se c'è tempo extra
2. Problema **secondario**, bassa priorità
3. Problema **rilevante**, alta priorità
4. **Catastrofe di usabilità**, deve essere risolto prima del rilascio

Nel valutare la severità, si considerano tre fattori: la 🕒 frequenza di occorrenza, l'impatto del problema *in caso di manifestazione* (se è superabile o meno, e le sue conseguenze pratiche), e la **persistenza** (probabilità che lo stesso utente incontri il problema più volte).

Ogni valutatore produce una **tabella** in cui indica la posizione di ciascun problema all'interno della propria classifica, secondo una strategia di **competizione standard**. Una volta ottenute tutte le classifiche, si distingue tra *problemi ad alta priorità* e *problemi di minore rilevanza*, dove le posizioni di alta priorità rappresentano circa il **20% del totale dei problemi**.

Si prendono in considerazione le prime 3 o 6 posizioni di ciascun valutatore, ed infine, per ogni problema, si valuta *quante volte si colloca nella fascia di alta priorità rispetto a quella di bassa priorità*.

È importante includere nella tabella finale la **mediana, o la media dell'indice di severità** per facilitare l'analisi. Sulle righe troviamo i valutatori, sulle colonne i problemi.



1. Ordiniamo quante volte il problema è stato riscontrato (tabella 2)
2. Verifichiamo quante volte un problema è stato riscontrato tante volte
 - a. Es. *problema 1 è stato classificato due volte tra le priorità alte, mentre mai nella fascia inferiore*

Perché usarla?

È una tecnica di ingegneria dell'usabilità *discount* poiché è rapida, efficace, e comporta un rapporto costo-beneficio vantaggioso. È **meno costosa** e **fornisce risultati già interpretati**, ma presenta limitazioni, come *la possibilità di non individuare tutti i problemi* o di produrre falsi positivi, e la *mancaanza di un contesto di un utente reale*

Tabella

Per gestire al meglio i problemi di usabilità, è importante ordinare questi dati all'interno di tabelle e grafici.

- Tabella

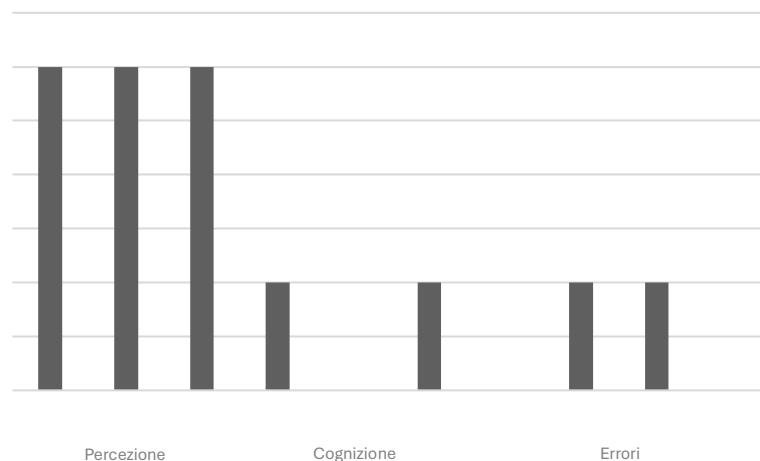
- **ID:** identificatore unico del problema
- **Descrizione breve:** sintesi del problema
- **Descrizione lunga:** contenere dettagli completi
- **Euristiche violate:** numero delle euristiche di usabilità
- **Valutatori:** identificatori che hanno segnalato ciascun problema
- **Severità media:** gravità del problema calcolata come *media/mediana tra le valutazioni dei valutatori*
- **Fascia di priorità:** a che grado di priorità è il problema
- **SD severità:** deviazione standard delle severità, che mostra quanto concordano i valutatori sulla gravità del problema

ID	Descrizione breve	Descrizione lunga	Euristiche violate	Valutatori	Severità media	Fascia di priorità	SD severità

- **Grafico a griglia problema(valutatori):** mostra una matrice in cui ogni problema viene associato ai valutatori che l'hanno identificato. Le caselle riempite indicano *quali valutatori hanno riscontrato un determinato problema*. Un problema segnalato da più valutatori è probabilmente rilevante.

								Valutatore
								6
								5
								4
								3
								2
								1
Problema	7	6	5	4	3	2	1	

- **Frequenza di violazione:** grafico a barre che rappresenta la frequenza di violazione delle euristiche. Mostra quali tipi di problemi di usabilità sono più comuni, aiutando a identificare aree di miglioramento nell'interfaccia. Ad esempio, *se le violazioni cognitive sono alte, potrebbe esserci un problema di comprensibilità del sistema*



Errori frequenti

- **Aggregare i problemi senza analizzarli singolarmente:** ogni problema richiede una soluzione specifica, e una visione generale rischia di omettere dettagli cruciali.

- **Non proporre soluzioni durante la discussione di un problema:** questi devono emergere soltanto alla fine della valutazione. Le soluzioni devono essere formulate in un momento successivo, e non incluse nella fase di analisi euristica
- **Considerare più principi contemporaneamente:** ogni problema deve essere analizzato rispetto a un singolo principio alla volta, identificando le violazioni specifiche
 - o Invece di partire dalle euristiche per verificare se ogni aspetto dell'applicazione le rispetta, è *meglio esaminare ciascun problema* e poi scegliere le euristiche più rilevanti per quel caso specifico.

Contenuti del rapporto finale

Il rapporto finale deve includere:

- **Descrizione del design sperimentale:** specificare **cosa è stato valutato**, il **motivo della scelta**, il numero di partecipanti coinvolti e le caratteristiche delle attività svolte. Includere informazioni rilevanti come il *numero di download dell'applicazione* e i *profili degli utenti*.
- **Elenco dei problemi prioritizzati:** presentare una *tabella dei problemi*, inclusi screenshot dei problemi più gravi, e una matrice di problemi/valutatori. È utile aggiungere un'analisi della distribuzione dei problemi rispetto alle euristiche violate
- **Considerazioni finali:** offrire una *riflessione sul numero totale di problemi rilevati*, sulla loro distribuzione e su eventuali citazioni significative emerse durante le sessioni di valutazione
- **Allegati:** inserire come appendice le *schede di analisi individuali*. Si produce una presentazione PowerPoint, con registrazioni e moduli di consenso raccolti nella cartella del progetto.

Perché effettuare una valutazione?

È uno strumento fondamentale per comprendere se, e **come** la tecnologia migliora la vita delle persone. Permette di verificare **se il sistema è conforme agli standard** e, in alcuni casi, aiuta ad evitare sanzioni. È essenziale per confrontare progetti e sistemi: sia nella fase di *progettazione*, per valutare diverse iterazioni di prototipi, sia per *analizzare la concorrenza*, comprendendo punti di forza e debolezza.

Approcci

Esistono vari approcci per una valutazione:

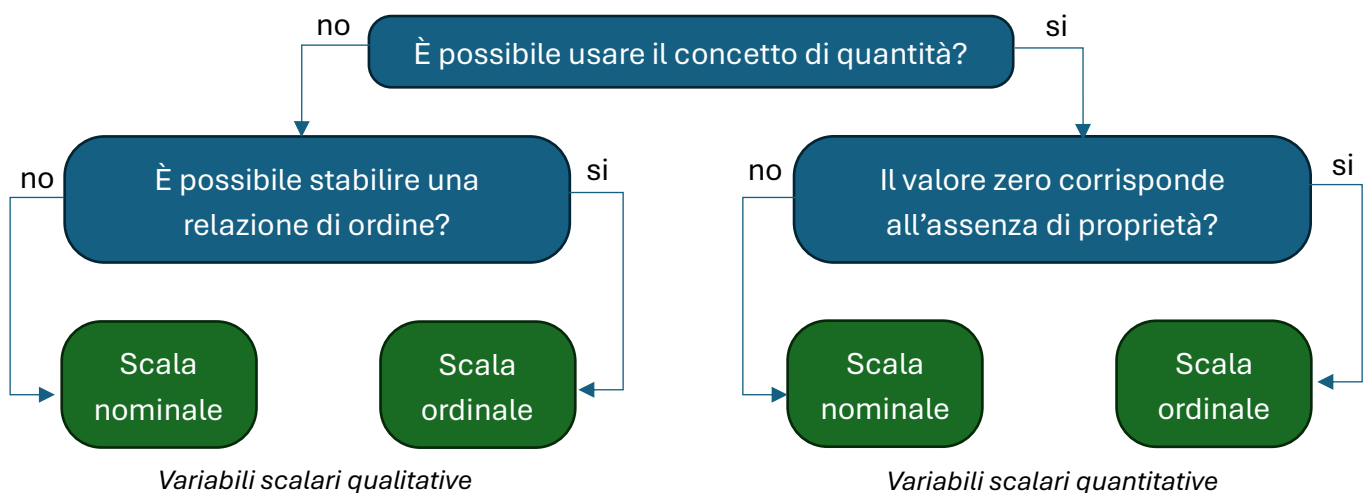
- **Approccio positivista:** basato su metodi quantitativi; l'analisi si concentra su un *campione rappresentativo*, utilizzando tecniche di inferenza statistica inversa. Si predilige l'osservazione di dati concreti, senza speculazioni teoriche astratte.
- **Metodi qualitativi sociologici, antropologici, etnografici ed etnometodologici:** si osservano comportamenti e pratiche di lavoro nell'ambiente naturale dell'utente
 - o **Osservazione e monitoraggio:** tecniche sociologiche utili per analizzare l'interazione degli utenti nel contesto lavorativo
 - o **Raccolta di opinioni:** attraverso metodi strutturati o semi-strutturati, gli utenti sono intervistati su preferenze, abitudini e pensieri
 - o **Approcci interpretativi:** raccolta di dati naturalistici, minimizzando interferenze, per comprendere come il sistema si integri con altre attività dell'utente.

Tipi di variabili

Si parte dai dati: con *empirismo* intendiamo che dobbiamo guardare le cose che accadono concretamente. Questi dati possono essere:

- **Numerici:** possono essere scalari, quali l'*altezza*, *tempo* o *numero di errori*
- **Ordinali:** è possibile stabilire quale categoria o nome viene prima, senza pari merito, quali il *titolo di studio*, *classi di età* o *giudizi di merito*
- **Categoriche:** i valori possibili sono modalità/parole, quali *si/no*.

Ognuna di queste variabili ha una matematica associata.



	Nominale	Ordinale	A intervalli	A rapporti
Tipo di variabile	Qualitativa categorica	Qualitativa rettilinea o ciclica	Quantitativa	Quantitativa
Finalità	Classificazione	Ordinamento	Misura attraverso l'applicazione di una unità di misura, conservando le relazioni tra distanze e adottando zero arbitrario	Misura attraverso applicaizione di una unità di misura, conservando il rapporto tra i valori e adottando uno zero assoluto
Operazioni matematiche di base	=, ≠	=, ≠, <, >	=, ≠, <, >, +, -	=, ≠, <, >, +, -, *, /
Indicatori statistici	Numero di casi (N), moda	Mediana, percentili, quartili, scarto interquartile, correlazione per ranghi di Spearman	Media, varianza, deviazione standard, correlazione di Pearson	Media, varianza, deviazione standard, correlazione di Pearson, concentrazione
Rappresentazione dei dati raccolti	Aerogrammi, istogrammi	Istogrammi, diagramma a scatola e baffi (box and whisker plots)	Diagramma a scatola e baffi; distribuzione continue; grafico di dispersione	Diagramma a scatola e baffi; distribuzione continue; grafico di dispersione
Test di ipotesi inferenziale	Test binomiale, test del chi quadro → numero di errori	Test binomiale, test del chi quadro. <i>Test di Mann-Whitney, test di Kruskall-Wallis → livello di soddisfazione</i>	Test T di Student, ANOVA	Test T di Student, ANOVA → tempo di esecuzione

Statistica

Non possiamo conoscere con precisione i parametri della popolazione, ma *possiamo stimarli basandoci su un campione*. Ogni stima introduce però un errore standard.

Curva di Gauss

La curva di Gauss o curva a campana rappresenta la distribuzione dei valori di una variabile in un campione. La deviazione standard indica quanto i dati si discostano dalla media, influenzando la forma della curva: maggiore è la deviazione, più piatta sarà la curva.

Intervalli di confidenza

Un intervallo di confidenza rappresenta un **range** entro cui, con un certo livello di probabilità, si trova il vero parametro della popolazione.

- **Livello di confidenza:** percentuale di volte in cui il parametro vero è incluso nell'intervallo calcolato (e.g. *95% significa che solo 1 stima su 20 sarà errata*)
- **Dimensione del campione:** maggiore è il campione, più stretto sarà l'intervallo di confidenza, aumentando la precisione della stima.
- **Confronto tra livelli di confidenza:** un intervallo di confidenza al 99% sarà più ampio rispetto a uno al 95%, poiché include una maggiore incertezza per garantire una copertura più alta. Se volessimo ottenere un intervallo di confidenza al 100% dovremmo considerare l'intera popolazione, il che è spesso impraticabile.

P-value

Il p-value rappresenta la **probabilità di raccogliere dati simili, o ancora più estremi, rispetto a quelli osservati, assumendo che l'ipotesi nulla H_0 sia vera**. Questo valore gioca un ruolo cruciale nella decisione di accettare o rigettare H_0 .

Se il p-value è inferiore a una soglia di significatività α , stabilita generalmente attorno al 5%, possiamo considerare i dati **incompatibili con l'ipotesi nulla**, rigettandola.

Un p-value basso implica che è improbabile che una differenza osservata sia dovuta al caso. Tuttavia, non conferma la verità dell'ipotesi alternativa, ma suggerisce che i dati sono più coerenti con essa.

Per confrontare due sistemi A e B, per esempio, utilizziamo un p-value:

- **Two-tail test**, per verificare se A e B sono diversi
- **One-tail test**, per determinare se uno dei due sistemi è migliore dell'altro

Procedura di test statistico

1. **Definizione dell'ipotesi nulla H_0 :** questa ipotesi stabilisce che non ci sia alcuna differenza significativa tra i gruppi o le condizioni in esame

2. **Selezione dello strumento di misura:** cronometri, questionari, etc... con relativa metrica
3. **Raccolta dei dati sperimentali:** Y, i dati raccolti presentano una distribuzione specifica
4. **Scelta del test statistico:** T-test, z-score test, etc..
5. **Confronto del p-value con alpha:**
 - a. $p < \alpha$: rigettiamo H_0 e congetturiamo che le differenze siano reali
 - b. $p \geq \alpha$: non possiamo rigettare H_0 , ma il confronto fornisce indicazioni sulla forza dell'ipotesi alternativa

Un p-value non rappresenta la probabilità che H_0 sia vera, ma indica $P(D|H_0)$, ovvero la probabilità dei dati osservati assumendo H_0 .

Zeta score test

Un z-score test confronta il tasso di errori tra due sistemi, come ad esempio:

- Sistema A: proporzione $A \pm$ intervallo di confidenza
- Sistema B: proporzione $B \pm$ intervallo di confidenza

Se z-score e p-value mostrano differenze non significative, possiamo concludere che **i due sistemi presentano performance comparabili**.

T-test di Student

Il T-test verifica **differenze significative tra i tempi medi di completamento di un task usando sistemi diversi**. Lo facciamo *paired* se sono gli stessi soggetti ma in condizioni diverse (es. Sistema A vs B), o *indipendenti* se sono soggetti diversi per condizioni diverse (es. *maschi vs femmine*).

Come esempio abbiamo che, una differenza significativa per il tempo di completamento ($p < 0.05$) suggerisce che gli utenti impiegano meno tempo con il sistema A rispetto al sistema B.

Alla fine di tale test, avremmo una tabella simile a quella a fianco, e in base a come decidiamo di dividere i dati, potremmo scegliere campioni indipendenti o no. Per esempio, se selezioniamo in base al genere dell'intervistato, avremo campioni indipendenti; se scegliamo in base alle app, avremmo campioni dipendenti.

1	1								
2	1								
1	1								
2	1								
2	2								
1	2								
1	2								

Campioni indipendenti

1	1								
2	1								
1	1								
2	1								
2	2								
1	2								
1	2								

Campioni dipendenti

Formula di Bayer

La probabilità condizionata è calcolata come segue:

$$P(B|A) = \frac{P(A|B) * P(B)}{P(A)}$$

Dove $P(A|B)$ è chiamata **verosimiglianza**, $P(B)$ è chiamata **probabilità a priori di B**, e $P(A)$ è chiamata **evidenza**.

Variabili misurabili

Le metriche possono essere classificate in:

- **Efficacia**: percentuale di compiti completati con successo
- **Efficienza**: tempo medio per completare un compito, o numero di errori per unità di tempo
- **Soddisfazione**: opinioni espresse dagli utenti

Abbiamo i seguenti esempi come variabili misurabili: si dividono in **efficacia**, **efficienza**, **soddisfazione**.

1. Time to complete task
2. Percentage of task completed
3. Percentage of task completed per unit time
4. Ratio of successes to failures
5. Time spent on errors
6. Percentage number of errors
7. Percentage or number of competitors that do this better than the current product
8. Number of commands used
9. Frequency of help or documentation used
10. Time spent using help or documentation
11. Percentage of favorable/unfavorable user comments
12. Number of repetitions of failed commands
13. Number of runs of successes and of failures
14. Number of times the interface misleads the user
15. Number of good and bad features recalled by users
16. Number of available commands not invoked
17. Number of regressive behaviors
18. Number of users preferring your system
19. Number of times users need to work around a problem
20. Number of times the user loses control of the system
21. Number of times the user expresses frustration of satisfaction

Efficienza d'uso

Il sistema deve essere efficiente da usare cosicchè, quando l'utente lo ha appreso, sia possibile un alto livello di produttività.

Uno dei modi per misurare l'efficienza è **calcolare il tempo necessario ad eseguire il compito dato correttamente**; ovviamente, c'è un tempo massimo per eseguire il compito – nel caso l'utente fallisse a fare il compito, o andasse oltre a questo limite, semplicemente si attribuirà come tempo il tempo massimo.

Il tempo è una variabile continua, solitamente con una distribuzione normale, e quindi si può applicare la tecnica nota come il **test T di Student**

Misurazione del successo

Un sistema di successo è definito non solo dalle sue caratteristiche, ma anche dalla sua **accoglienza da parte degli utenti**. Indicatori chiave sono che gli utenti scelgono il sistema, e sono disposti a pagarlo; il sistema genera ricavi crescenti e acquisisce una quota di mercato maggiore; gli utenti lo raccomandano e lo considerano parte della loro routine.

Un sistema di successo dovrebbe essere **utile** (perché soddisfa pienamente le esigenze dell'utente), **usabile** (perché consente di raggiungere gli obiettivi in modo intuitivo riducendo il rischio di errori), e **desiderabile**.

Sistema walk-up and use

Un sistema di questo genere deve offrire un livello di usabilità estremamente elevato, tale da risultare intuitivo e **immediatamente utilizzabile** da qualsiasi utente, anche senza alcuna formazione preventiva. È quindi progettato per essere così autoesplicativo da consentire ad utenti occasionali, o al primo utilizzo, di *operare efficacemente senza necessità di una guida o introduzione*.

Esercitazione

Obsolescenza programmata

L'obsolescenza programmata è una strategia industriale volta a **limitare intenzionalmente la durata dei prodotti**, obbligando i consumatori a sostituirli più spesso. Si realizza mediante opportuni accorgimenti introdotti in fase di produzione, come *l'utilizzo di materiali di scarsa qualità* ecc. È stata iniziata formalmente nel 1924 con il cartello Phoebus, il quale portò le aziende a impostare uno standard per la durata delle lampadine, riducendola da 1500-2000 ore a circa 1000 ore.

L'obsolescenza programmata non si limita alla durata dei prodotti, ma utilizza varie strategie per accelerare la sostituzione:

- **Materiali di bassa qualità:** i prodotti sono realizzati con materiali che si degradano rapidamente
- **Costi di riparazione elevati:** i costi di riparazione spesso superano quelli di acquisto di un nuovo prodotto
- **Innovazioni minime:** i nuovi modelli di prodotto introducono solo cambiamenti superficiali, pubblicizzati come significativi per stimolare la domanda

È **dannosa per l'utente**, che si trova costretto a sostituire frequentemente beni che potrebbero durare più a lungo. Anche se spesso legale, è *eticamente discutibile* poiché sfrutta la dipendenza del consumatore dalle innovazioni tecnologiche.

Dark patterns

Sono tecniche di design ingannevoli che **spingono l'utente a compiere azioni non volute**, e dunque, non sono semplicemente errori di design, ma **scelte deliberate** progettate per favorire le conversioni a scapito della trasparenza.

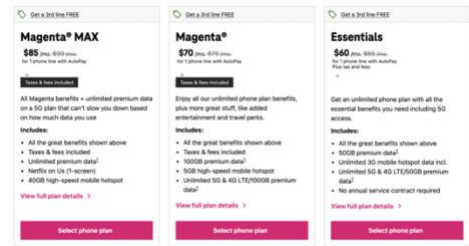
Presentano vantaggi come *l'incremento delle conversioni e delle metriche di engagement* (click-through rate) nei test A/B, ma sono fortemente criticati per le loro implicazioni etiche, e per la perdita di fiducia che possono causare nei consumatori.

Il riconoscimento di essi avviene tramite lo **studio dell'usabilità e dei processi decisionali umani**, ponendo attenzione agli effetti psicologici che queste strategie generano sugli utenti.

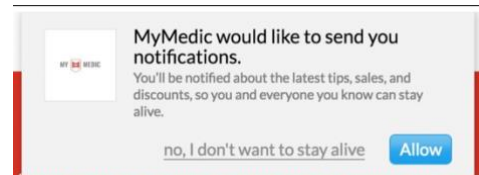
Tipologie

Di seguito, sono illustrate alcune delle tipologie di dark patterns più comuni:

1. **Comparison Prevention:** rende difficile confrontare prodotti o servizi, sfruttando bias come la prova sociale o il default. Se questa attività di valutazione diventa troppo complicata, l'utente potrebbe rinunciare a prendere una decisione affrettata. L'utente diventa più vulnerabile a bias **cognitivi** come la *social proof*, l'*authority bias* o l'*effetto default*.



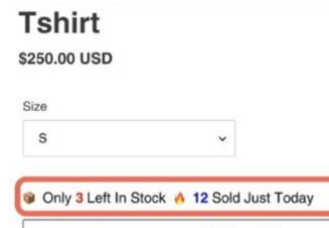
2. **Confirmshaming:** usa sensi di colpa o vergogna per spingere l'utente verso un'azione desiderata. Spesso presentano pulsanti di **opt-out** con etichette formulate in modo denigratorio o sminuente, facendo sentire male l'utente per aver scelto quell'opzione.



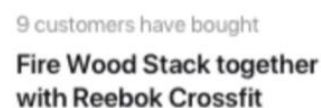
3. **Disguised Ads:** nasconde pubblicità come contenuto ordinario, confondendo l'utente e aumentando le probabilità di click. Sono spesso progettati per **sembrare elementi dell'interfaccia**, articoli correlati o altri contenuti di interesse per gli utenti. In questo modo, si possono generare più entrate dalle impressioni pubblicitarie.



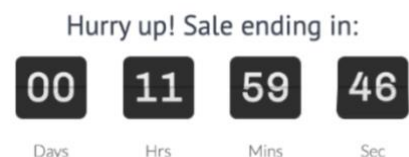
4. **Fake Scarcity:** crea un senso di urgenza ingannevole, suggerendo scorte limitate, creando un senso artificiale di **disponibilità limitata**, spingendo gli utenti ad agire rapidamente per paura di perdere un'opportunità. Viene ottenuto mostrando *messaggi fuorvianti sui bassi livelli di scorte o sull'elevata domanda*



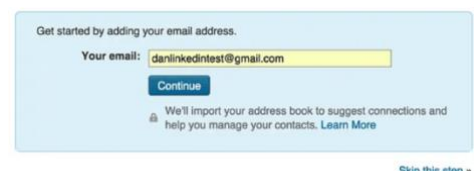
5. **Fake Social Proof:** utilizza recensioni o approvazioni false per sfruttare il bias della prova sociale. Sfrutta il **bias cognitivo della prova sociale**, secondo cui gli individui tendono a conformarsi al comportamento degli altri. È un meccanismo che consente alle persone di evitare il lavoro impegnativo di condurre una valutazione critica autonoma.



6. **Fake Urgency:** pressa l'utente con scadenze imminenti, riducendo il tempo di analisi. I fornitori possono sfruttare questa situazione spingendo l'utente a *completare un'azione che non potrebbe essere del tutto nel suo interesse*.



7. **Forced Action:** richiede un'azione da parte dell'utente per ottenere un servizio o proseguire in un'attività. Può essere combinata con altri dark pattern come *sneaking*, in modo che gli utenti non se ne accorgano, o *trick wording*, per far sembrare l'azione più desiderabile di quanto non sia. A



volte un'azione opzionale viene presentata come obbligatoria, attraverso l'uso di interferenze visive o linguaggio ingannevole.

8. **Hard to Cancel:** rende complesso cancellare abbonamenti, incentivando un uso prolungato del servizio. Questo di solito comporta **nascondere l'opzione di cancellazione**, richiedere gli utenti di chiamare il servizio clienti per annullare, e rendere il processo di cancellazione eccessivamente complesso e dispendioso in termini di tempo.

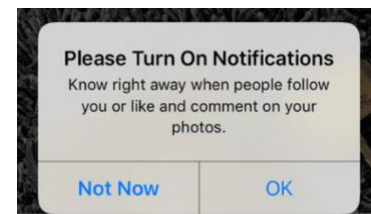


9. **Hidden Costs:** mostra costi aggiuntivi solo in fase avanzata di acquisto, quando l'utente ha già investito tempo. Implicano l'oscuramento o l'omissione di commissioni, addebiti o costi aggiuntivi, fino a quando l'utente non è già avanzato nel processo di acquisto o iscrizione. A quel punto, l'utente *ha già investito tempo ed energia nella transazione*, ed è più incline a proseguire nonostante i costi inattesi.

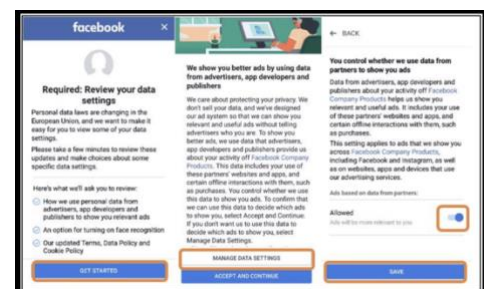
10. **Hidden Subscription:** induce l'utente a sottoscrivere un abbonamento senza esserne consapevole. Gli utenti pensano di portare a termine una determinata azione, ma in realtà c'è una clausola nascosta che li iscrive. Una volta iscritti, *il servizio è discreto* e l'utente non riceve email o notifiche che lo informano che sta pagando su base ricorrente, in modo che i pagamenti continuino il più a lungo possibile. È spesso abbinato all'*hard to cancel*.

Hey @figmadesign, could you please tell people that they're being charged extra money if they submit this form with the "can edit" option? And maybe explain your pricing model INSIDE the app? This is @darkpatterns af

11. **Nagging:** interrompe costantemente l'utente con richieste, aumentando il costo mentale fino a portarlo a cedere. È come una tassa che il fornitore impone agli utenti che non vogliono conformarsi alle richieste del fornitore stesso; sebbene il costo non sia finanziario, si accumula e alla fine diventa significativo.



12. **Obstruction:** crea ostacoli intenzionali per scoraggiare l'utente da azioni non desiderate dalla piattaforma. Viene utilizzata per esaurire gli utenti e indurli a rinunciare, specialmente quando i loro obiettivi sono contrari agli interessi di ricavo o crescita dell'azienda. A volte viene utilizzata per **“preparare” gli utenti a un inganno più grande**.



13. **Preselection:** imposta opzioni predefinite per favorire decisioni convenienti per il fornitore. Un approccio comune è mostrare una **casella già selezionata**. Il contenuto può essere scritto in modo da far sentire l'utente che altre persone come lui accetterebbero l'opzione predefinita, inducendo così il **social proof bias**.
14. **Sneaking:** nasconde informazioni rilevanti per manipolare le scelte dell'utente.
15. **Trick Wording:** utilizza formulazioni ambigue per ingannare l'utente, sfruttando letture veloci degli utenti, che usano questa strategia per gestire enorme quantità di informazioni

con cui si confrontano; quindi, non sempre non leggono e non si soffermano su ogni parola di ogni pagina.

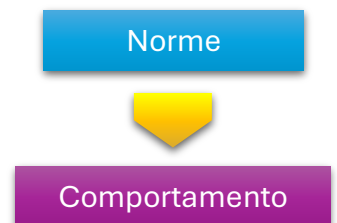
16. **Visual interference**: manipola il design visivo per confondere e nascondere informazioni importanti, utilizzando elementi sovraccarichi o colori che distraggono.

Approcci teorici alla teoria delle decisioni

Approccio normativo

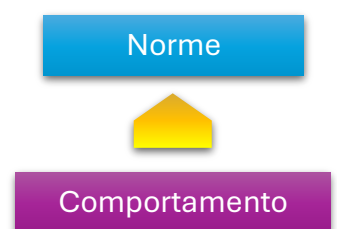
Descrive il **processo decisionale ideale che un individuo seguirebbe se fosse perfettamente razionale**, basandosi su regole standard di razionalità. È un modello **top-down**, che parte dalle norme per arrivare al comportamento.

La razionalità umana, però, è limitata da tre principali fattori: **stupidità** (capacità di calcolo ridotte), **ignoranza** (mancanza di informazioni complete), **passioni** (emozioni che influenzano le scelte).



Approccio descrittivo

Si focalizza su **come le persone prendono effettivamente decisioni**, osservando il **comportamento reale**. È un modello **bottom-up**, che parte dal comportamento per arrivare alle norme, come illustrato nella *teoria del prospetto di Kahneman e Tversky*. Tale teoria evidenzia che le persone spesso violano le regole elementari del calcolo delle probabilità, effettuando giudizi non sempre accurati.



Approccio prescrittivo

Cerca di **combinare elementi dei due approcci** per migliorare la qualità delle decisioni. Si domanda, infatti, come la teoria delle decisioni possa essere applicata per rendere il processo decisionale più efficace.

Teoria del doppio processo

La teoria del doppio processo distingue due sistemi di pensiero coinvolti nelle decisioni di ciascun umano.

- **Sistema 1**: processo rapido, intuitivo e automatico, che utilizza *scorciatoie cognitive ed euristiche* per prendere decisioni veloci

- **Sistema 2:** processo lento e razionale, che richiede uno sforzo mentale maggiore e analizza le informazioni in modo più accurato.

Il Sistema 1, come detto, si avvale di **algoritmi euristici**, che sono strategie cognitive rapide.

Euristiche

Le euristiche, in psicologia, sono scorciatoie cognitive che permettono alle persone di prendere decisioni in maniera rapida e veloce, quando si trovano in condizioni di incertezza e/o non possiedono tutte le informazioni necessarie.

Queste euristiche possono portare a decisioni soddisfacenti, pur non essendo sempre precise. *Herbert Simon* fu uno dei primi a proporre questo concetto, osservando come *le strategie euristiche siano adattive, e non necessariamente negative*. Alcune euristiche comuni sono:

- **Euristica della disponibilità:** giudica la probabilità di un evento in base alla facilità con cui un esempio viene ricordato
- **Euristica della rappresentatività:** valuta un evento in base alla sua somiglianza con un prototipo
- **Fallacia della congiunzione:** quando le persone considerano più probabile una combinazione di eventi piuttosto che gli eventi singoli
- **Euristica dell'ancoraggio:** si basa su un'informazione iniziale per prendere decisioni, anche se tale informazione è irrilevante

Stili decisionali

Gli individui adottano differenti stili decisionali, ciascuno con caratteristiche peculiari:

- **Evitante:** evita decisioni quando possibile
- **Intuitivo:** si affida all'intuito e alle sensazioni per prendere decisioni, spesso focalizzandosi sugli aspetti globali
- **Dipendente:** preferisce ottenere consigli prima di prendere una decisione
- **Razionale:** cerca un'analisi completa e dettagliata delle informazioni disponibili
- **Spontaneo:** è guidato dall'immediatezza e preferisce prendere decisioni rapide basate sul sentimento del momento

Architettura della scelta

Si riferisce **al modo in cui le opzioni vengono presentate agli utenti**, influenzandone le decisioni. Il *choice architect* è chi progetta tale ambiente decisionale, decidendo come

mostrare le opzioni e quali variabili ambientali e di presentazione possono influenzare le scelte dell'utente.

Nudge

Il **nudge** descrive una **spinta gentile che influenza le decisioni degli individui** senza limitare le opzioni. Un nudge deve **poter essere ignorato**, e non deve influire significativamente sugli incentivi economici dell'utente. Weinmann descrive un processo in cinque fasi: *definire l'obiettivo, diagnosticare il processo decisionale e gli effetti psicologici, selezionare i nudge più adatti, implementare le scelte architettoniche, misurare l'efficacia del nudge*.

Sono classificabili in base alla *trasparenza* e alla *visibilità*:

- **Automatico-trasparente**: influenza chiaramente il comportamento, come il *cambio di un'opzione predefinita*
- **Riflessivo-trasparente**: spinge l'utente a riflettere visibilmente, come un promemoria stradale
- **Riflessivo-non trasparente**: influenza le scelte in modo nascosto, come l'aggiunta di opzioni che rendono altre più desiderabili

Bias cognitivi

Effetto framing

Il framing rappresenta il fenomeno per cui **la percezione di una scelta varia a seconda di come è presentata**.

Ad esempio, incorniciare un'opzione in termini di guadagni piuttosto che di perdite influenza positivamente la scelta dell'utente.

Effetto default

Indica la tendenza delle persone ad **accettare l'opzione predefinita**. Anche se razionalmente non dovrebbe influenzare le decisioni, è stato dimostrato che le persone tendono a mantenere impostazioni di default per evitare un'analisi più complessa.

Ad esempio, per la donazione di organi, i Paesi con consenso presunto hanno una percentuale di donatori significativamente più alta.

Effetto posizione

Mostra come la **disposizione delle opinioni influenzi la scelta**. Anche l'ordine dei risultati di ricerca può manipolare le preferenze, influenzando le decisioni degli utenti senza che essi ne siano consapevoli.

Effetto social proof

È un bias psicologico che **porta le persone a imitare il comportamento degli altri** in situazioni di incertezza. Le opinioni e azioni degli altri diventano un punto di riferimento per le decisioni personali, spesso senza un'analisi critica.

Usabilità

Per l'usabilità, le possiamo dividere in due categorie.

Facts of Life

Steve Krug formulò tre “fatti della vita” che spiegano il comportamento degli utenti:

- **Non leggiamo le pagine, le scansioniamo:** gli utenti scorrono rapidamente le pagine cercando informazioni chiave, senza leggere attentamente ogni parola
- **Non facciamo scelte ottimali, ci accontentiamo:** la tendenza è quella di scegliere un'opzione abbastanza buona, piuttosto che cercare la migliore
- **Non cerchiamo di capire a fondo il funzionamento, procediamo a tentoni:** gli utenti spesso navigano per tentativi, senza approfondire il funzionamento di un sistema.

Euristiche di Nielsen



Nielsen ha proposto dieci principi, chiamati anche euristiche di Nielsen, che servono per migliorare l'interazione utente:

1. **Visibilità dello stato del sistema:** il sistema dovrebbe fornire all'utente feedback immediato su cosa sta succedendo
2. **Corrispondenza tra sistema e mondo reale:** dovrebbe usare un linguaggio e simboli familiari all'utente
3. **Controllo e libertà dell'utente:** possibilità di annullare e ripristinare azioni, e di uscire da situazioni indesiderate
4. **Consistenza e standard:** design coerente con standard del settore per facilitare il riconoscimento
5. **Prevenzione dell'errore:** dovrebbe prevenire errori, o fornire conferme per azioni critiche
6. **riconoscimento più che ricordo:** minimizzare il carico di memoria mostrando oggetti e opzioni visibili
7. **Flessibilità ed efficienza:** offrire scorciatoie per utenti esperti, e rendere il sistema accessibile anche ai principianti
8. **Estetica e progettazione minimalista:** evitare elementi grafici o testuali irrilevanti che riducono la chiarezza
9. **Aiuto per riconoscere, diagnosticare ed uscire da errori:** fornire messaggi chiari e soluzioni pratiche per gli errori
10. **Aiuto e documentazione:** offrire documentazione utile e facilmente reperibile, pur preferendo un sistema intuitivo

Etica

Nel contesto filosofico, rappresenta l'**insieme di dottrine e riflessioni speculativi riguardanti il comportamento pratico umano**. Il suo scopo principale è indicare il vero bene e i mezzi per conseguirlo, definendo i doveri morali dell'individuo verso se stesso e gli altri, e offrendo criteri per giudicare la moralità delle azioni.

Secondo una visione comune, quindi, l'etica può essere considerata come uno **strumento per riflettere su come le persone dovrebbero comportarsi** e per emettere giudizi sulle loro azioni, distinguendo tra ciò che è giusto o sbagliato, buono o cattivo.

Nella ricerca

Assume un ruolo centrale, **regolando la conduzione degli studi per garantire il rispetto dei diritti e del benessere** dei partecipanti. I principali punti di riferimento storici sono:

- **1947, codice di Norimberga:** elaborato dopo i processi di Norimberga, questo codice ha stabilito principi fondamentali per la ricerca medica

- **1964, dichiarazione di Helsinki:** redatta dall'Associazione Medica Mondiale, ha rafforzato le linee guida per la ricerca clinica, promuovendo il rispetto per i diritti umani, e il benessere dei soggetti coinvolti
- **1979, rapporto Belmont:** ha introdotto principi fondamentali per la ricerca con soggetti umani, tra cui il rispetto per le persone, la beneficenza e la giustizia.

I principi

I principi fondamentali sono:

- **Integrità:** gli studiosi devono garantire l'onestà e la trasparenza nei loro metodi e risultati
- **Rispetto della dignità delle persone:** tutti i partecipanti devono essere trattati con rispetto, tutelando la loro dignità e privacy
- **Competenza:** i ricercatori devono possedere le conoscenze e competenze adeguate per condurre la ricerca in modo sicuro e rispettoso
- **Responsabilità sociale:** i ricercatori sono responsabili dell'impatto sociale della loro ricerca e devono contribuire al benessere della società
- **Tutela del benessere:** la protezione del benessere fisico e psicologico dei partecipanti è prioritaria

Consenso informato

È una componente cruciale nella ricerca e nel trattamento dei dati, poiché **garantisce che i partecipanti abbiano la piena comprensione delle modalità di utilizzo delle loro informazioni personali**. Perché il consenso sia valido, il titolare del trattamento deve dimostrare che l'interessato *sia consapevole della portata del trattamento e abbia accettato volontariamente*. Il consenso deve essere fornito in un linguaggio chiaro, senza clausole ingannevoli, e il partecipante deve avere la libertà di revocarlo senza conseguenze negative. Il consenso informato implica la **trasparenza sui dati raccolti, l'identità del titolare del trattamento e le finalità specifiche del trattamento stesso**.

Debriefing

Alla fine di uno studio, è il momento in cui i ricercatori incontrano i partecipanti per **fornire una spiegazione completa dell'esperimento**, dei suoi scopi e delle procedure utilizzate. Permette quindi di chiarire eventuali dubbi e rispondere alle domande dei partecipanti, assicurando che *abbiano una piena comprensione dell'esperienza di ricerca*.

Auto-valutazione etica

È un processo fondamentale per i ricercatori che **devono assicurarsi di condurre studi in conformità con i principi etici** e le normative vigenti. Garantisce il rispetto dei diritti e del benessere dei partecipanti, ma promuove anche una ricerca che sia eticamente solida e responsabile.

Comitato etico

Svolge un ruolo chiave nel **controllo della ricerca**, assicurando che gli studi siano condotti in modo rispettoso dei diritti e della sicurezza dei partecipanti. Il comitato esamina i progetti di ricerca per accertarsi che siano etici, conformi agli standard, e promuove pratiche che rispettino la dignità umana.

Privacy

È un altro principio fondamentale, e le loro informazioni devono essere trattate in modo da evitare ogni possibilità di identificazione individuale. I dati devono quindi essere utilizzati solo in forma aggregata e anonima, riducendo il rischio di divulgazione o di uso improprio. Di seguito un paio di esempi:

- **Social media:** per trattare i dati degli utenti sui social media, è necessario ottenere il consenso esplicito e informato di ciascun utente, chiarendo come saranno utilizzate le informazioni personali.
- **Ricerca accademica:** i partecipanti a studi accademici devono essere informati con chiarezza su come i loro dati saranno trattati, e devono avere sempre la possibilità di ritirarsi dallo studio senza penalità.

Metodi qualitativi

Dati qualitativi

Rappresentano informazioni descrittive che riguardano caratteristiche o attributi non numerici, spesso usati per classificare o etichettare variabili in categorie distinte. Sono utili per analizzare aspetti soggettivi e contestuali dell'interazione, senza bisogno di attribuire valori numerici.

Osservazione

È un metodo qualitativo per raccogliere dati osservando direttamente i soggetti in un contesto naturali. Principali tipologie includono:

- **Osservazione naturalistica:** l'osservatore si limita a osservare senza interferire con il contesto

- **Vantaggi:** fornisce un quadro dettagliato e privo di influenze esterne; è utile per *costruire una conoscenza solida sul fenomeno osservato*
- **Svantaggi:** può verificarsi l'*effetto Hawthorne*, dove i soggetti, consapevoli di essere osservati, modificano il proprio comportamento. Richiede tempo e risorse, presenta problemi di oggettività, e non permette interventi diretti per comprendere meglio le azioni osservate
- **Osservazione partecipata:** il ricercatore si inserisce nell'ambiente studiato, partecipando attivamente
 - **Caratteristiche:** richiede un'*immersività prolungata nel contesto sociale*, con tentativi di registrazione accurata e oggettiva delle attività e dei comportamenti. L'osservatore, che può essere esterno al gruppo, cerca di identificare e interpretare azioni rilevanti
 - **Vantaggi:** permette una comprensione interna del fenomeno, sviluppando empatia e raccogliendo punti di vista autentici
 - **Svantaggi:** rischi di soggettività, effetto Hawthorne, **effetto Pigmalione** (le aspettative dell'osservatore possono influenzare i comportamenti osservati).

Focus group

Tecnica di ricerca qualitativa che prevede una discussione in un piccolo gruppo di persone, moderata per raccogliere opinioni e percezioni. È particolarmente utile per **sfruttare la dinamica del gruppo**, permettendo ai partecipanti di stimolarsi reciprocamente, e di ricordare dettagli o aspetti non considerati individualmente.

Può essere **autogestito**, dove i partecipanti, seguendo temi generali, gestiscono autonomamente la discussione; **moderato**, dove il moderatore guida il gruppo secondo una scaletta, cercando di coprire ogni argomento con domande aperte per un'indagine completa.

- **Vantaggi:** fornisce un'ampia gamma di opinioni e prospettive; l'interazione sociale attiva il *ricordo di dettagli*; utile per comprendere dinamiche di gruppo e raccogliere opinioni autentiche
- **Svantaggi:** richiede *tempo e risorse* per organizzazione e moderazione; rischia il *monopolio della conversazione* da parte di figure carismatiche; può emergere il *bias di desiderabilità sociale*, dove i partecipanti esprimono opinioni socialmente desiderabili

Think-aloud protocol

Consiste nell'**esplicitare i propri pensieri durante lo svolgimento di un compito**, una tecnica formalizzata negli anni '40 da *Karl Duncker* per studiare i processi di risoluzione dei problemi.

I partecipanti, guidati da un facilitatore, descrivono i loro pensieri, reazioni ed emozioni, mentre interagiscono con un sistema. Come fasi abbiamo:

1. **Progettazione e preparazione:** definire numero e tipo di partecipanti, istruzioni e compiti
2. **Reclutamento dei partecipanti**
3. **Conduzione della prova**
4. **Analisi dei risultati**

Come **vantaggi** abbiamo un metodo intuitivo, rapido e a basso costo; fornisce insight sui processi decisionali e sul design. Gli **svantaggi** sono una soggettività del partecipante e dell'osservatore, e un rischio di autocensura e dell'effetto Hawthorne.

Intervista

È una tecnica versatile per raccogliere informazioni dirette, con diversi gradi di strutturazione:

- **Strutturata:** domande predefinite e standardizzate, rigide, adatte per risposte specifiche
- **Semi-strutturata:** flessibile, con un elenco minimo di temi che permette approfondimenti
- **Libera:** esplorativa, per raccogliere opinioni e riflessioni spontanee

Deve avere la seguente struttura:

1. **Introduzione:** presentare contesto e tema, ringraziare, stabilire fiducia
2. **Consenso informato:** raccogliere il consenso per l'uso dei dati, con un registro scritto o registrato
3. **Domande di profilazione:** informazioni demografiche (età, esperienza, istruzione)
4. **Domande generali:** creare un ambiente rilassato con domande accessibili e stimolanti
5. **Domande approfondite:** esplorare l'interazione con il sistema, usando esempi ed aneddoti
6. **Domande controverse:** stimolare riflessioni critiche, ad esempio sulle difficoltà incontrate
7. **Spazio per conclusioni del partecipante:** offrire l'opportunità di aggiungere pensieri

Vantaggi	Svantaggi
È un metodo versatile che può essere usato in ogni fase	Richiede tempo e risorse
Permette di sviluppare un dato ottenuto	Può durare a lungo, il che potrebbe stancare tutti i partecipanti
Conoscere e indagare risposte	Problema di identificazione del campione rappresentativo; ci potrebbe essere un'influenza sull'intervistatore

Intervistatore

Cosa fare	Cosa non fare
Conoscere l'argomento di indagine	Usare il gergo degli intervistati, mantenere il tono dell'intervista
Saper ascoltare e comprendere ciò che viene raccontato	Prendere appunti dettagliati, poiché si registra e questo potrebbe distrarre l'intervistatore
Formulare domande	Giudicare l'intervistato
Domande esplorative di carattere generale per aiutare	Porre <i>leading questions</i> per ottenere determinate risposte
Capire quando fermarsi	Domande con negazioni, condizionali, giri di parole, domande chiuse, più di una domanda allo stesso tempo

Analisi qualitativa

L'analisi qualitativa si basa su dati ricchi e dettagliati raccolti da fonti come *documenti*, *osservazioni*, *interviste*, *videoregistrazioni* o *testimonianze storiche*. Questi dati, i quali solitamente sono ampi e complessi, richiedono un'organizzazione strutturata per estrarre il significato e costruire una teoria. Abbiamo una serie di caratteristiche:

- **Volume:** solitamente grande quantità di informazioni
- **Varietà:** i dati possono includere trascrizioni, osservazioni sul campo e video
- **Richiesta di organizzazione:** necessario un processo di codifica per individuare i temi principali, e semplificare l'analisi

Il processo

La codifica

La codifica è il processo che permette di assegnare **etichette** a piccole porzioni di dati; queste servono per identificare temi, raggruppare informazioni e creare categorie di analisi.

I **codici** rappresentano un titolo sintetico di un paragrafo o concetto, e devono catturare l'elemento primario.

Le **categorie** raggruppano codici simili, usati per identificare relazioni tra concetti. Di seguito, i vari metodi.

Coding	Descrizione	Esempio
Descriptive	Assegnare codici descrittivi	<i>As i walked toward the school, there was a 7-11 convenience store next to a Burger King</i> Businesses
In Vivo	Uso delle parole esatte del partecipante come codici	<i>I hated school last year... This year's a lot better</i> Hated school, this year's better

Process	Si concentra su azioni osservabili, con solo gerundi	<i>If you say one thing, two people will tell more, and soon everyone knows</i> Knowing, telling
Emotion	Identifica sentimenti ed emozioni espresse nei dati	<i>I just hated it when he got awarded the honor</i> Delusione, amarezza
Values	Esplora atteggiamenti, valori e sistemi di riferimento	<i>Government regulation of women's health issues has gotten out of hand</i> Misoginia, controllo del governo
Evaluation	Valuta la qualità o il successo di un evento o programma	<i>This residency program was successful this year</i> Residency: successo

Pattern coding

È un livello più alto di codifica, che raggruppa grandi quantità di dati in temi o modelli. Sintetizza i dati complessi, facilita l'analisi durante la raccolta dei dati focalizzando meglio il lavoro, aiuta a sviluppare schemi cognitive per comprendere eventi e interazioni, e prepara il terreno per analisi comparative in studi multicase.

Grounded theory

È un metodo sistematico per **sviluppare teorie a partire dai dati**. È particolarmente utile per *studiare fenomeni poco compresi*, generare teorie laddove quelle esistenti sono inadeguate, ed esaminare processi complessi. Abbiamo vari approcci principali:

- **Glaseriano**, dove si concentra sull'emergere spontaneo dei dati senza preconcetti
- **Straussiano**, che adotta un sistema rigoroso e strutturato di codice (es. *codifica assiale*)
- **Charmaziano**, che combina flessibilità e sensibilità al contesto, enfatizzando la prospettiva del partecipante.

Come fasi abbiamo:

1. **Open coding**: si frammentano i dati in unità minime, e si assegnano codici semplici. L'obiettivo è *individuare concetti iniziali*.
2. **Codifica assiale**: si aggregano i codici intorno a temi principali, identificando relazioni causali, condizioni intervenienti, strategie d'azione e conseguenze
3. **Codifica selettiva**: si individua un concetto centrale che unifica tutte le categorie

Memo writing

Documenta riflessioni e inferenze durante il processo di analisi, come il significato di un'osservazione, reazioni personali ai partecipanti, e dubbi sulla qualità dei dati raccolti.