

01 - Human-centered Design Process

Usability (ISO standard definition) DEFINIZIONE IMPORTANTE

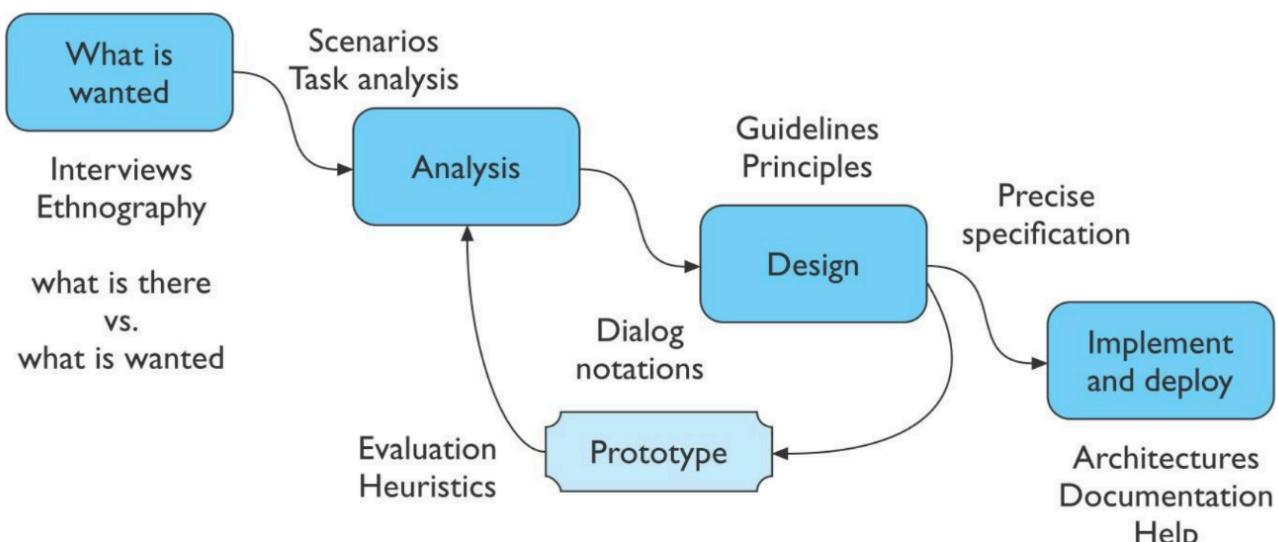
Usabilità: "misura in cui un sistema, un prodotto o un servizio può essere utilizzato da utenti specifici per raggiungere obiettivi specifici con efficacia, efficienza e soddisfazione in un determinato contesto d'uso".

- Gli utenti, gli obiettivi e il contesto di utilizzo "specificati" si riferiscono alla particolare combinazione di utenti, obiettivi e contesto d'uso per cui l'usabilità viene presa in considerazione
- Il termine "usabilità" è usato anche come qualificatore per riferirsi alla progettazione conoscenze, competenze, attività e attributi progettuali che contribuiscono all'usabilità, come ad esempio la **all'usabilità**, come l'**esperienza di usabilità**, l'**ingegneria dell'usabilità**, i **test di usabilità**, ecc.

Un'altra definizione:

Usabilità: la capacità degli utenti di utilizzare le funzionalità del sistema.

- Utilità: fa qualcosa che la gente vuole?
- Imparabilità: è facile da imparare?
- Memorabilità: una volta appreso, è facile da ricordare?
- **Efficacia:** permette di raggiungere l'obiettivo?
- **Efficienza:** una volta appreso, è veloce da usare?
- Visibilità: lo stato del sistema è visibile?
- Errori: gli errori sono pochi e recuperabili?
- **Soddisfazione:** è piacevole da usare?



| Human-Centered Design Process – Passi Principali

| Need Finding - cosa si desidera

- Cosa serve esattamente? In che modo le persone raggiungono attualmente l'obiettivo?
- Osservazione degli utenti, interviste,

| Analisi

- Formalizzare e strutturare le esigenze
- Creare scenari di interazione, storie, compiti
- Confrontare la situazione attuale con la nuova situazione prevista

| Design

- Le scelte principali per dare forma al sistema
- Regole, linee guida, principi di progettazione
- Considerare diversi tipi di utenti
- Modellazione e descrizione dell'interazione
- Layout visivo
- Considerare tutti gli input provenienti da modelli cognitivi, teorie della comunicazione, questioni organizzative

| 02 - Interazione Uomo Macchina

| Interazione Uomo Macchina DEFINIZIONE IMPORTANTE

L'interazione uomo-macchina è una **disciplina multi campo**

Si occupa della progettazione, della valutazione e della sistemi informatici interattivi per l'uso umano e dello studio dei principali fenomeni che li circondano

Coinvolge due entità (l'uomo e il computer) che determinano il comportamento reciproco nel tempo

- inquadrato in termini di obiettivi dell'uomo e relativi compiti/percorsi

| L'interazione Uomo-Macchina è MULTIDISCIPLINARE

- Psicologi
- Ergonomici
- Sociologi
- Informatici
- Economisti
- Grafici
- Esperti di scrittura di manuali e documenti tecnici

Per aiutarci ad applicare le competenze di molti campi diversi:

- Metodi e processi di progettazione processi di progettazione
- Modelli
- Euristica
- Best Practice
- Convenzioni

| I Goals dell'interazione Uomo-Macchina

| Ingredienti

- Gli utenti
- I Computer
- I task da svolgere

| Goals

Il sistema deve supportare il compito dell'utente compito dell'utente, con particolare attenzione alla sua usabilità

- Utile
- Utilizzabile
- Utilizzato

| Modelli di interazione

Cos'è l'interazione?

Concept	View of interaction	Key phenomena and constructs	Good interaction	Example support for evaluation and design
Dialogue	a cyclic process of communication acts and their interpretations	mappings between UI and intentions; feedback from the UI; turn taking	understandable; simple, natural; direct	methods/concepts for guessability, feedback, mapping; walkthroughs
Transmission	a sender sending a message over a noisy channel	messages (bits); sender and receiver; noisy channels	maximum throughput of information	metrics and models of user performance
Tool use	a human that uses tools to manipulate and act in the world	mediation by tools; directness of acting in the world; activity as a unit of analysis	useful and transparent tools; amplification of human capabilities	compatibility in instrumental interaction; break down analysis
Optimal behavior	adapting behavior to goals, task, UI, and capabilities	rationality; constraints; preferences; utility; strategies	improves or reaches maximum or satisfactory utility	models of choice, foraging, and adaptation
Embodiment	acting and being in situations of a material and social world	intentionality; context; coupling	provides resources for and supports fluent participation in the world	studies in the wild; thick description
Experience	an ongoing stream of expectations, feelings, memories	non-utilitarian quality; expectations; emotion	satisfies psychological needs; motivating	metrics of user experience; experience design methods
Control	interactive minimization of error against some reference	feedforward; feedback; reference; system; dynamics	rapid and stable convergence to target state	executable simulations of interactive control tasks

Taken from: Kasper Hornbæk & Antti Oulasvirta, What Is Interaction? In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems

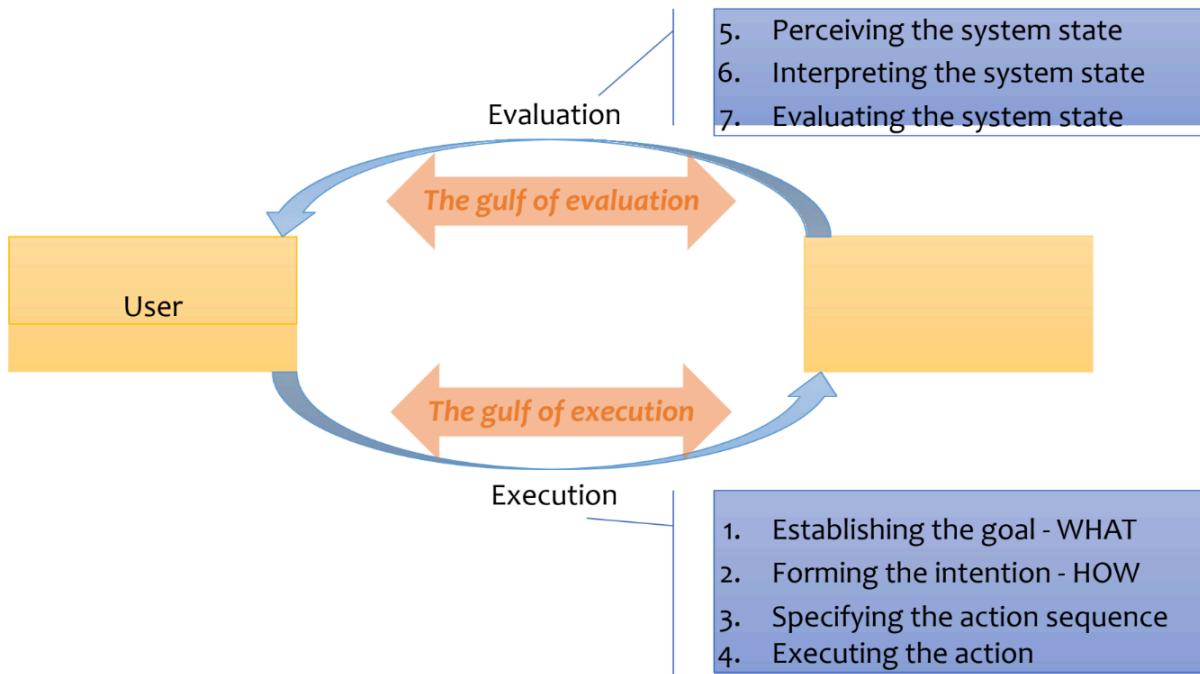
Interazione:

- Non è l'idea promossa e ripetuta nelle nozioni popolari secondo cui un computer e un umano sono impegnati
 - Riguarda due entità - l'uomo e il computer - che si determinano a vicenda nel tempo
 - La loro determinazione reciproca può essere di molti tipi, tra cui statistica, meccanica e strutturale.
- Gli **utenti**, con i loro obiettivi e le loro ricerche, sono la metrica finale dell'interazione.

| Ipotesi

- L'**utente** vuole raggiungere alcuni obiettivi, in uno specifico dominio applicativo.
 - Ogni dominio ha un gergo specifico, un insieme di possibili processi e obiettivi, artefatti e blocchi di costruzione
- I **compiti** sono operazioni per manipolare i concetti di un dominio.
 - L'obiettivo si raggiunge eseguendo uno o più compiti.
- L'interazione studia la relazione tra utente e sistema.
 - Il sistema possiede uno stato e "parla" un linguaggio di base.
 - L'utente possiede uno stato, che comprende la comprensione dello stato del sistema, l'intenzione di eseguire un'attività del sistema, l'intenzione di eseguire un compito e "parla" il linguaggio del compito.

| Modello di Norman per l'interazione uomo macchina



| Gli errori

| Errori Umani

Nel corso dell'esecuzione

| Scivoloni

- Avete formulato l'azione giusta azione giusta, ma non riuscite a eseguirla azione in modo corretto
 - Ad esempio, cliccate sull'icona sbagliata, oppure fare doppio clic troppo lentamente
- Può essere corretto da una migliore interfaccia (spaziatura, layout, evidenziazioni, ...)

| Errori

- Non conoscete bene il sistema e non si può formulare l'obiettivo giusto obiettivo
 - Es fare click su una lente ma non significa zoom ma un'altra cosa
- Il modello mentale dell'utente sullo stato del sistema non è corretto. stato del sistema non è corretto
- Richiede una riprogettazione più radicale o un formazione aggiuntiva

| Informazioni sugli errori umani

- Gli errori umani non devono mai essere considerati come colpe dell'utente.
- Piuttosto, “di solito sono il risultato di una cattiva progettazione” (Norman).
- Gli esseri umani tendono a essere imprecisi, distratti, non consapevoli.
 - La progettazione del sistema dovrebbe anticipare questo comportamento umano
 - Minimizzare la possibilità di azioni inappropriate (valutazione)

- Massimizzare la possibilità di scoprire e riparare un'azione inappropriata (esecuzione) azione inappropriata (esecuzione)
- Consentire agli utenti di comprendere lo stato del sistema e di costruire un modello modello appropriato

| Frameworks: Major UI Styles

- Interfaccia a riga di comando
- Menù
- Linguaggio naturale
- Dialogo domanda/risposta e interrogazione
- Compilazione di moduli e fogli di calcolo
- Windows, icone, menu e puntatori (WIMP)
- Mobile
- Punto e clicca
- Interfacce tridimensionali

| Design Processes and Frameworks

| User-Centered Design (UCD)

- Evitare il rischio di fallimento del progetto software
 - Si stima che il 50% sia influenzato da una cattiva comunicazione sviluppatore < >utente/cliente
- L'UCD tiene conto delle esigenze, dei desideri e dei limiti degli utenti finali reali durante ogni fase del processo di progettazione
 - I problemi di progettazione incentrati sull'utente vengono scoperti durante le fasi iniziali
- **Benefici:** sistemi più facili da apprendere, con prestazioni più rapide, con meno errori umani, incoraggiano gli utenti a scoprire funzionalità avanzate errori umani, incoraggiano gli utenti a scoprire funzionalità avanzate ed evitano di “costruire il sistema sbagliato”.
- **Problemi:** come trovare gli utenti? Quanti? Quanto motivati? Come parlare la loro linguaggio? Come estrarre le esigenze degli utenti, le esigenze aziendali, le implicazioni organizzative?

| Participatory Design

| Agile Interaction Design

| Design Thinking

| Service Design

| 03 - Need Finding

Parto dalle slide "Need Finding III - Altre Query Techniques"

| Interviste specifiche

Sono interviste specifiche svolte a specifici utenti di diverse categorie

| Lead user

utenti che hanno dei need nuovi, prima di tutti gli altri, particolarmente competenti e sofisticati trovano da soli soluzioni ai loro need beneficiano significativamente di tali soluzioni ci indicano i need che molti avranno nel prossimo futuro.

| Extreme User

utenti che spingono un sistema esistente all'estremo trovano problemi altrimenti difficili da individuare needs amplificati e work around più evidenti i loro needs sono i needs anche di altri, ma più visibili

| Interviste a esperti

sfruttare le conoscenze di esperti del settore possono discutere anche di problemi più difficili o più astratti la loro esperienza offre informazioni aggregate attendibili sul comportamento degli utenti

ATTENZIONE

Gli esperti **NON** sono utenti esperti, ma esperti del settore

| QUESTIONARI

Set di domande uguali per tutti gli utenti

Veloci, molte risposte raccolte presto

Possono essere analizzati più rigorosamente

Molti tipi di domande: aperte, chiuse, risposta singola, multipla, scala di valori, ranking, etc.

Meno flessibili delle interviste

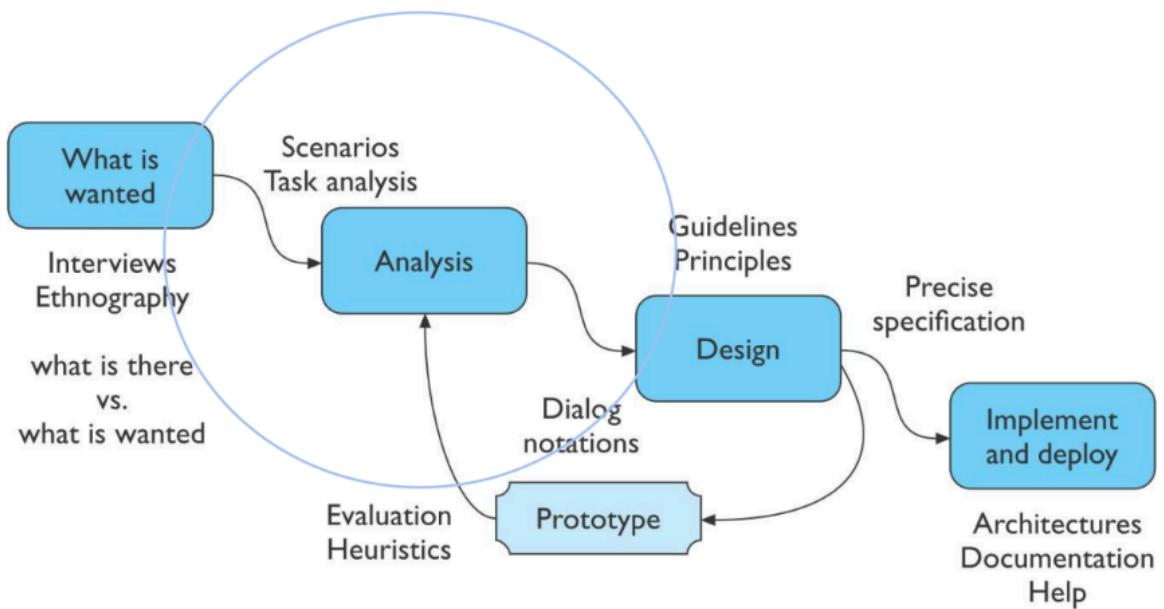
Pensare prima a come saranno analizzate le risposte

Vanno progettati accuratamente e testati su pochi utenti (pilota)

Esistono altri strumenti e tecniche per il needfinding ma

| 04 - Task analysis

Human-Centered Design Process



| Need o Goal?

I **need** guidano i **goal**: i need sono la forza trainante dei goals

- ad esempio: per migliorare il benessere (bisogno), si stabilisce l'obiettivo di fare esercizio fisico regolare, controllare il sonno o mangiare in modo più sano. Regolarmente, monitorare il sonno o mangiare in modo più sano

Più obiettivi possono derivare da un unico bisogno.

Gli obiettivi traducono i bisogni in azioni: gli obiettivi sono una forma concreta di bisogni.

- Il contesto influenza gli obiettivi, ma non i bisogni: gli obiettivi variano a seconda del contesto, delle risorse disponibili, ma anche dei bisogni sottostanti. Contesto, dalle risorse disponibili, ma i bisogni sottostanti rimangono gli stessi
- Ad esempio: il bisogno è “sentirsi sicuri”; l’obiettivo può essere “ottenere una promozione al lavoro” o “installare un sistema di sicurezza”.

Ma facciamo una semplificazione: **Need = Goals**

| Task Analysis

(Need/Goal-Centred)

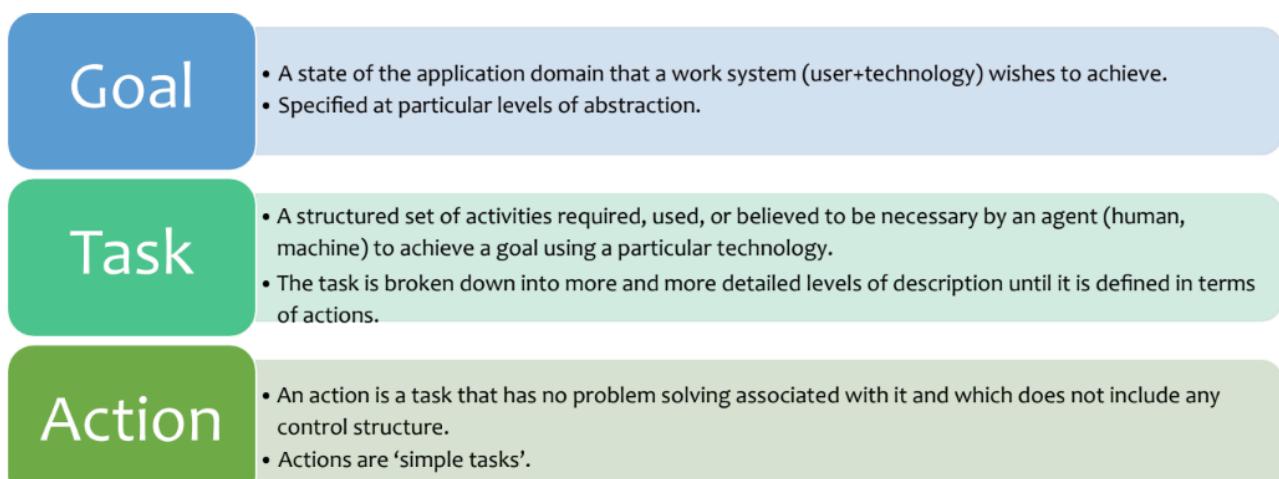
L’analisi dei compiti è lo studio del modo in cui le persone svolgono le loro attività

L’obiettivo è determinare

- cosa fanno (fasi)

- quali cose usano (artefatti)
- quanto riescono a fare (obiettivi)

Cos'è un **Task**



| Cosa impariamo dalla Task Analisys?

- Quali possono essere gli obiettivi dei vostri utenti; cosa stanno cercando di ottenere
- Cosa fanno effettivamente gli utenti per raggiungere questi obiettivi
- Quali sono le esperienze (personal, social e cultural) che gli utenti apportano alle attività
- Come gli utenti sono influenzati dall'ambiente fisico in cui vivono
- Come le conoscenze e le esperienze pregresse degli utenti influenzano:
 - Il modo in cui pensano al loro lavoro
 - il flusso di lavoro che seguono per svolgere i loro compiti
 - I punti dolenti che sperimentano per svolgere le attività

| Perchè è importante?

L'analisi dei compiti è il processo di apprendimento degli utenti ordinari osservandoli in azione per capire nel dettaglio come svolgono i loro compiti e osservandoli in azione per capire in dettaglio come svolgono i loro compiti e come raggiungere gli obiettivi prefissati

L'analisi dei compiti aiuta a:

- identificare i compiti che l'applicazione deve supportare
- affinare o ridefinire la navigazione o la ricerca dell'applicazione
- Raccolta dei requisiti dell'applicazione
- Sviluppare la strategia dei contenuti e la struttura dell'applicazione
- Le fasi iniziali della prototipazione
- Esecuzione di test di usabilità

| Esempi

esempio di esigenze/obiettivi:

- o monitorare la mia salute
- o organizzare meglio il mio tempo

Compiti derivati:

- o inserire i dati sulla salute in un sistema di monitoraggio digitale

- o impostare promemoria per l'esercizio quotidiano

▪ compiti poco definiti:

- o migliorare la dieta (non operativo, troppo vago)

- o utilizzare l'app medica (troppo generico e legato a un'interfaccia specifica)

▪ azioni derivate:

- o misurare i propri parametri di salute, come la frequenza cardiaca o la pressione sanguigna, utilizzando uno strumento o un dispositivo comodo per l'utente

- o registrare i dati in modo da poterli rivedere facilmente in un secondo momento (ad es. in un diario, utilizzando un sistema di tracciamento fisico o digitale)

| Caratteristiche del Task Analisys

L'analisi dei compiti è più facile quando si dispone di flussi di lavoro ben definiti (ad esempio, la pianificazione di un viaggio da viaggio da qualche parte)

- attività ripetute, come ad esempio la programmazione

Sfida:

- Non progettiamo compiti, ma interfacce
- I compiti e gli oggetti non corrispondono a una corrispondenza 1:1
 - Ad esempio, un'applicazione web ha compiti multipli
- Le persone usano la stessa interfaccia e la stessa applicazione per ottenere risultati leggermente diversi o fanno cose diverse l'una dall'altra

| 05 - Personas

Strumento usato nel design di interfacce

- rappresentano modelli astratti di utenti reali, aiutando a comprendere meglio le esigenze (need) e i comportamenti (actions) degli utenti durante il processo di progettazione
- definizione di Personas:

- rappresentazioni semi-fittizie di utenti reali, create sulla base di dati demografici, comportamentali e psicografici
- rappresentano categorie di utenti con caratteristiche, obiettivi, e necessità specifiche

| Obiettivi

- aiutano a creare empatia nei confronti degli utenti, permettendo ai progettisti di mettersi nei loro panni
- guidano le decisioni di progettazione, aiutando a ottimizzare l'esperienza utente
- forniscono un quadro di riferimento per la progettazione, consentendo di identificare le esigenze degli utenti e le possibili sfide

| Elementi

- nome: dà un'identità alla persona
- foto: permette una connessione più umana con la persona
- caratteristiche demografiche: età, sesso, professione, istruzione, ecc.
- obiettivi: cosa cerca di ottenere utilizzando l'interfaccia?
- frustrazioni: quali sono le sfide e i problemi che potrebbe sperimentare?
- comportamenti: come utilizza il sistema? quali sono le sue abitudini?

| Vantaggi

- **creare empatia** - è possibile per i designer acquisire una prospettiva simile a quella dell'utente, avendo interiorizzato gli obiettivi, i bisogni e i desideri della persona
- **sviluppare il focus** - rende più evidente che non è possibile progettare per tutti, altrimenti si rischia di progettare per nessuno

- **comunicare e raggiungere un consenso** - aiutano a comunicare le conclusioni della ricerca alle persone che non hanno potuto partecipare agli incontri con gli utenti
- **prendere decisioni** - quando si vede il mondo dalla prospettiva dell'utente, diventa molto più semplice determinare cosa è utile e cosa è un caso limite
- **misurare l'efficacia** - possono fungere da sostituti degli utenti

| 06 - Storyboards

- “Una rappresentazione grafica dell'aspetto esteriore del sistema previsto. dell'aspetto esteriore del sistema previsto, senza funzionalità del sistema”.
- Un fumetto disegnato a mano che mostra l'esecuzione di un compito (come uno scenario concreto). l'esecuzione di un compito (come uno scenario concreto).
- Con poche tavole (sequenza di schizzi), il fumetto trasmette ciò che una persona può realizzare o include sempre le persone
- Comunicano il flusso, mostrando cosa cosa succede nei momenti chiave
- Non sono richieste abilità artistiche
 - Non si tratta di “belle immagini”
 - Si tratta di comunicare idee

| Cosa deve esserci in una storyboard

- Illustrare un obiettivo (per il compito)
- Come si svolge un compito (persone che interagiscono tra loro e con i dispositivi)
 - Ripetere per tutte le fasi significative
- Alla fine, come si raggiungono gli obiettivi (risultato soddisfacente)

Gli storyboard sono tutti incentrati sui compiti

Esempio di storyboard:



| Gli storyboard devono trasmettere

- **Setting**
 - Persone coinvolte
 - Ambiente
 - Compito da svolgere
- **Sequenza**
 - Quali sono le fasi coinvolte?
 - Non l'interfaccia utente dettagliata
 - Quale ruolo svolge l'interfaccia utente nell'aiutare gli utenti
 - Cosa porta qualcuno a utilizzare il sistema?
 - Il “trigger” per il compito
 - Quale compito viene illustrato?
- **Soddisfazione**
 - Qual è la motivazione per l'utente?
 - Il punto finale da raggiungere dopo tutti i passi
 - Qual è il risultato finale?
 - Quale bisogno state “soddisfacendo”?

| Gestire la dinamicità negli storyboard

- **Storyboard tradizionale**
 - Convenzioni “fumettistiche”: attori, bolle vocali, sfondo
 - Note allegate a ogni scena che spiegano cosa sta accadendo
- **Storyboard a punteggio**
 - Quando l'interfaccia utente è altamente dinamica o contiene elementi mediatici specifici elementi multimediali specifici
 - Aggiungere annotazioni specifiche sul movimento, i colori, i suoni, ...
- **Storyboard di solo testo**
 - Quando il comportamento dell'interazione è troppo complesso per essere compattato in un'illustrazione, utilizzare una descrizione testuale più lunga.

| Perché disegnare a mano?

- **Velocità**
 - Non è necessario dedicare tempo agli strumenti grafici (che “spingono” a concentrarsi sui “spingono” a concentrarsi sui dettagli)
 - Possibilità di sperimentare diversi scenari
- **Imprecisione**
 - Gli utenti si sentono liberi di esprimere ulteriori commenti e suggerimenti in merito. una versione più “curata”.
 - Concentrarsi sul contenuto (la grafica è ovviamente ignorata).
 - Nessuna distrazione da parte di font, colori, icone

| **Benefici dello StoryBoard**

Enfatizzare il modo in cui un'interfaccia svolge un compito.

Concentrare la conversazione e il feedback sui compiti dell'utente.

Mettere tutti d'accordo sugli obiettivi dell'applicazione

Evitare di fare il pignolo sui dettagli dell'interfaccia utente (pulsanti, ecc.).

| 07 - Pototyping

| L'obiettivo

- Envisionment: rendere visibili le idee
 - Generare nuove idee
 - Valutare le nuove idee (all'interno del gruppo di progettazione)
 - Testare le nuove idee (con gli utenti)
- Strumenti e tecniche diversi, a seconda di:
 - La fase di progettazione (iniziale, ..., avanzata, finale)
 - Il pubblico a cui ci si rivolge (progettisti, utenti di prova, clienti, direzione, ...)
- Errore da evitare: concentrarsi sull'interfaccia utente prima di concentrarsi sul compito che l'utente deve svolgere.

| Teniche

- Sketches
- Maps
- Low Fidelity (paper) prototypes
- Video Prototypes
- Medium Fidelity Prototypes
- High Fidelity Prototypes

| Progettare la UI per un prototipo

Passi fondamentali:

- immaginare l'interazione
- disegnare l'interfaccia
- progettare considerando gli “strumenti” disponibili
- dispositivi hardware
- widget
- stili di interazione

| Prototipo

- modello di un sistema interattivo
- simula o anima alcuni aspetti/caratteristiche/funzioni
- ma non tutti

| Proiezione del sistema

- prototipo realistico solo per quegli aspetti
- utilizzato per valutare l'impatto sugli utenti
- condizioni di valutazione devono essere simili a quelle previste per l'interfaccia finale

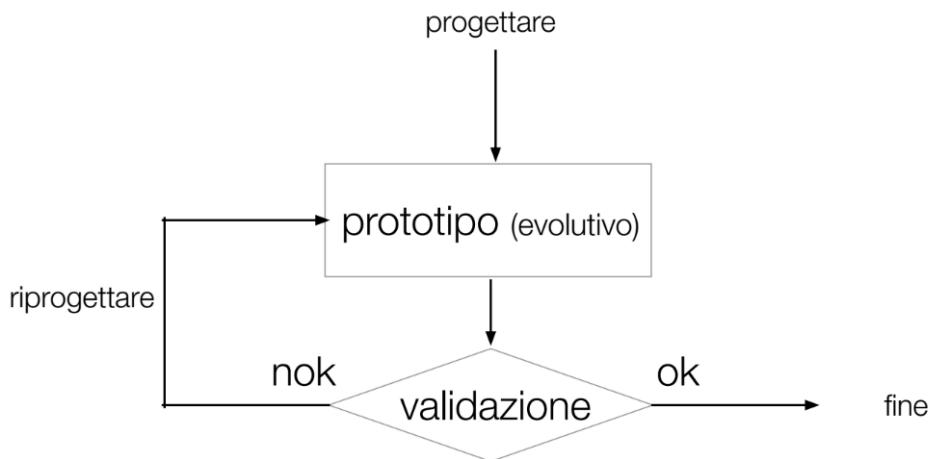
| Uso dei prototipi

- per provare e validare un concetto/idea
- per imparare
- per provare una versione iniziale, intermedia o finale

| Approcci al prototyping

- Throw away
- Incrementale - le varie funzioni vengono aggiunte una alla volta al prototipo
- Evolutivo - un prototipo serve come base per il successivo, che lo migliora

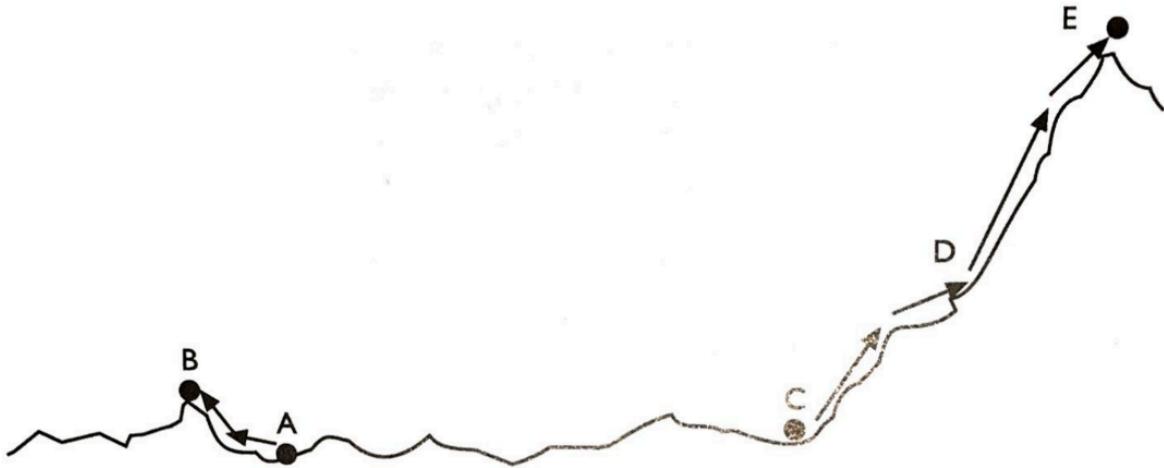
| Ciclo iterativo di prototipazione



| Prototipazione iterativa: best practice con pitfalls

- capire cosa è sbagliato
- ipotizzare come migliorare
- avere un buon punto di partenza

| Hill climbing



| Potenziali problemi

| 1.Tempo

- realizzare prototipi richiede tempo
- se vengono scartati, può sembrare tempo sprecato
- rapid prototyping
- ma non valutazione affrettata

| 2.Pianificazione

- difficile pianificare processo di design con prototyping
- difficile valutarne i costi
- contrattazione tra committente e progettista

| 3. Caratteristiche non funzionali

- sicurezza
- affidabilità
- tempo di risposta
- sacrificate nello sviluppo del prototipo

| Ulteriori Problemi

- inerzia del design — non viene mai riveduta una decisione iniziale sbagliata (vedi hill climbing)
- capire le cause dei problemi — non solo rilevarne i sintomi

| 08 - Tecniche di Prototyping

| 1. Low Fidelity Prototypes

| Paper Prototypes

Un mock-up dell'interfaccia utente disegnato a mano (di solito) su più fogli di carta di dimensioni variabili.

| Punti chiave

- Mockup interattivo su carta
 - Schizzi dell'aspetto dello schermo
 - I pezzi di carta mostrano finestre, menu, finestre di dialogo
- L'interazione è naturale
 - Puntare il dito = fare clic con il mouse
 - Scrivere = digitare
- Una persona simula il funzionamento del computer
 - Posizionando e raccogliendo i pezzi
- Scrivere risposte sullo "schermo"
 - Descrivere effetti che sono difficili da mostrare su carta
- Bassa fedeltà nell'aspetto e nelle sensazioni
- Alta fedeltà in profondità (la persona simula il backend)

| Materiali

- Paper, Transparent paper
- Pens, Markers
- Post-It notes
- Glues, scotch tape, scissors
- Photocopies
- UI Stencils
- Reusable UI components
- Printouts of screenshots

| Perchè usiamo il Paper Prototyping?

- Più veloce da costruire
 - Lo schizzo è più veloce della programmazione
- Più facile da modificare

- È facile apportare modifiche tra un test utente e l'altro, o anche *durante* un test utente
- Nessun investimento in codice - tutto viene buttato via (tranne il progetto)
- Concentra l'attenzione sul quadro generale
 - Il progettista non perde tempo con i dettagli
 - Il cliente dà suggerimenti più creativi, non fa i pignoli
- I non programmati possono aiutare
 - Sono richieste solo competenze di base

Paper Prototypes: Examples



| Come testare un Paper Prototype?

- Il team di progettazione deve coprire i seguenti ruoli:
 - **Attore "Computer"**
 - Simula il prototipo
 - Non fornisce alcun feedback che il computer darebbe
 - **Facilitatore**
 - Presenta l'interfaccia e i compiti all'utente
 - Incoraggia l'utente a "pensare ad alta voce" ponendo domande
 - Evita che il test dell'utente vada fuori strada
 - **Osservatore**
 - Tiene la bocca chiusa
 - Prende copiosi appunti

| Cosa possiamo imparare da un Paper Prototypes

- Modello concettuale
- Gli utenti lo capiscono?

- Funzionalità
 - Fa ciò che serve? Mancano caratteristiche?
- Navigazione e flusso di attività
 - Gli utenti sono in grado di orientarsi?
 - I prerequisiti delle informazioni sono soddisfatti?
- Terminologia
 - Gli utenti comprendono le etichette?
- Contenuti dello schermo
 - Cosa deve apparire sullo schermo?

| 2. Medium Fidelity Prototypes

NOI NON LI USEREMO, SERVONO AI DESIGNERS PER CAPIRE COME SI VEDONO LE COSE A SCHERMO

| Computer Prototypes

- Simulazione software interattiva
 - Riproduce l'interfaccia utente
 - Accetta alcuni input dell'utente
 - Risponde cambiando pagina
- Media fedeltà o Alta fedeltà nell'aspetto e nella sensazione
- Bassa fedeltà in profondità
 - L'operatore umano nella prototipazione cartacea è consapevole degli algoritmi

Conosciuto anche come “Mockup” (o “Interfaccia wireframe”)

-Disegno di una singola schermata o di un insieme di schermate collegate tra loro (seguendo un compito)

- Disegno “ondulato” o “impreciso” (ispirato al disegno a mano)
 - Si vuole trasmettere l'impressione che il disegno sia ancora preliminare
 - Per lo più in scala di grigi (o in bianco e nero)
- Informazioni solitamente statiche (solo pagine definite)
- Può suggerire il dispositivo dell'utente

UI Design Libraries



| Gli svantaggi degli strumenti

- Cliccare, non interagire
 - Nessun inserimento di testo, nessun inserimento di dati, nessuna selezione reale dei dati elencati
 - I widget non sono attivi
- I percorsi sono statici
- Il tester è impegnato in una “caccia all'hotspot”, per trovare i (pochi) unici widget realmente cliccabili
 - Ottimo per testare la comprensione dell'interfaccia utente e del flusso di lavoro
 - Non va bene per testare il comportamento dell'interfaccia utente

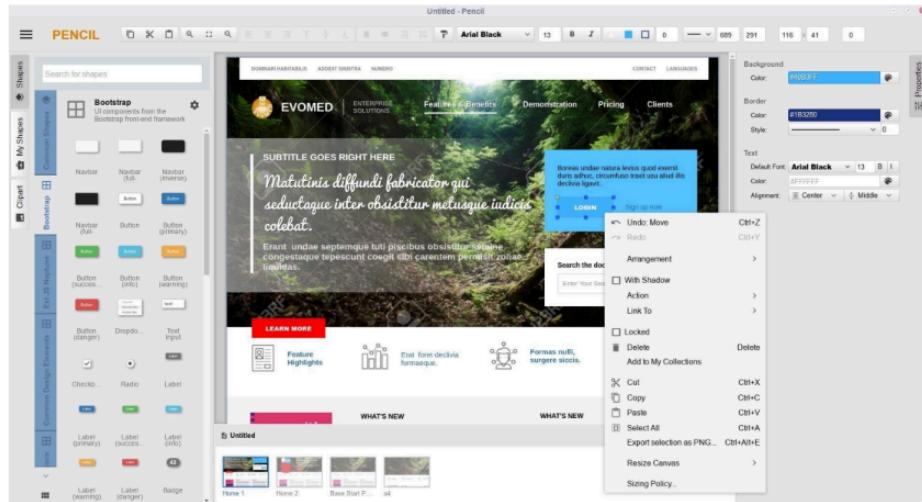
| 3. High Fidelity Prototypes

| Hi-Fi Prototypes

- Applicazione informatica vera e propria, con layout, colori e grafica di aspetto finale
 - Può utilizzare strumenti di prototipazione del design
 - Può utilizzare il codice di un'applicazione reale
- Molto più costoso da costruire
- Si dedica più tempo alla progettazione grafica che all'interazione.
- Quando viene testato, le persone commentano soprattutto i colori, i caratteri, ecc.
 - la rappresentazione comunica “finito”

High-fidelity Computer Prototypes

Semi-interactive



| Cosa possiamo imparare dai prototipi interattivi Hi-Fi?

- Layout dello schermo
 - È chiaro, opprimente, distraente, complicato?
 - Gli utenti riescono a trovare gli elementi importanti?
- Colori, caratteri, icone, altri elementi
 - Sono ben scelti?
- Feedback interattivo
 - Gli utenti notano e rispondono ai messaggi della barra di stato, ai cambiamenti del cursore e ad altri feedback?
- Questioni di efficienza
 - I controlli sono abbastanza grandi? Troppo vicini? L'elenco a scorrimento è troppo lungo?

| 4. Wizard-of-Oz Techniques

| Obiettivi

- Come testare un'applicazione davvero completa...
 - Con l'interfaccia utente finalizzata
 - Con algoritmi finalizzati
 - Con le cose che non siamo ancora in grado di implementare.
- Ma senza scrivere il codice.
 - Tranne che per un prototipo "muto" semi-interattivo

| Wizard of Oz

- Simulazione software con l'aiuto di un umano nel loop
- Il "Mago di Oz" = "l'uomo dietro la tenda".

- Simula il comportamento della macchina con un operatore umano
- Il mago è solitamente, ma non sempre, nascosto
- Spesso viene utilizzato per simulare la tecnologia del futuro.
 - Riconoscimento vocale
 - Apprendimento
- L'assistente può essere nascosto o visibile
 - Deve sempre essere rivelato, almeno alla fine

| Implementare un Wizard-of-Oz Prototype

- Scegliere le attività e gli scenari supportati
- Creare mock-up dell'interfaccia utente
 - Implementare una parte del sistema
 - Lasciare "ganci" per le azioni del Wizard
- Implementare un'interfaccia di back-office per il Wizard
- Definire le "regole di comportamento" per il Wizard
 - Quando deve rispondere
 - Come deve rispondere (l'"algoritmo")

| Benefits

- Più veloce e più economico della maggior parte dei prototipi interattivi
- Più "reale" della prototipazione su carta
- Creare più varianti è facile
- Identifica bug e problemi del progetto attuale
- Può immaginare applicazioni difficili da realizzare
- Giocare con la procedura guidata permette di comprendere meglio i requisiti algoritmici

| 09 - Tecniche di Valutazione con UTENTI

| Valutazione

- richiede un artefatto: simulazione, prototipo, implementazione completa.
- implementazione
- testa l'usabilità e la funzionalità del sistema
- avviene in laboratorio, sul campo e/o in collaborazione con utenti
- valuta sia la progettazione che l'implementazione
- deve essere considerato in tutte le fasi del ciclo di vita della progettazione ciclo di vita della progettazione

| Obiettivi della valutazione

- valutare la portata della funzionalità del sistema
- valutare l'effetto dell'interfaccia sull'utente
- identificare problemi specifici

| Valutazione mediante la partecipazione degli utenti

- partecipanti: rappresentativi degli utenti
- esperienza simile tra loro (omogenea)
- conoscenza di quel tipo di sistemi e di quel dominio
- numerosità dei partecipanti: J. Nielsen suggerisce almeno 3/5 utenti

| Scenario e task

- lo scenario descrive una situazione potenzialmente reale in cui i partecipanti alla valutazione si identificano
- permette di identificare: utenti, azioni, strumenti da utilizzare e contesti
- i compiti sono i singoli compiti che i partecipanti
- è anche possibile condurre valutazioni libere del sistema, senza compiti (ma non nel progetto!). sistema, senza compiti (ma non nel progetto!).

| Stili di valutazione

- In laboratorio
- Sul campo

| Valutazione in laboratorio

- disponibilità di attrezzature specialistiche

- ambiente ininterrotto
- mancanza di contesto
- difficile osservare più utenti che cooperano
- Appropriato se l'ubicazione del sistema è pericolosa o impraticabile

| Valutazione sul campo

- ambiente naturale
- mantenimento del contesto (anche se l'osservazione può alterarlo)
- possibilità di studi longitudinali
- distrazioni
- rumore
- Appropriato quando il contesto è cruciale per gli studi longitudinali

| METODI DI OSSERVAZIONE

- Pensare ad alta voce | **Think About**
- Valutazione cooperativa | **Cooperative evaluation**
- Analisi del protocollo | **Protocol analysis**
- Analisi post-task | **Post-task walkthroughs**

| Think About

- l'utente viene osservato mentre esegue il compito
- all'utente viene chiesto di descrivere ciò che sta facendo e perché, che cosa cosa pensa che stia accadendo, ecc.

| Vantaggi e Svantaggi

- semplicità - richiede poca esperienza
- può fornire informazioni utili
- può mostrare come il sistema viene effettivamente utilizzato
- soggettivo
- selettivo
- l'atto di descrivere può alterare la performance del compito

| Cooperative evaluation

- variante di think aloud
- l'utente collabora alla valutazione
- sia l'utente che il valutatore possono porsi reciprocamente delle domande nel corso della valutazione.

- Ulteriori vantaggi:
 - meno vincoli e più facile da usare
 - l'utente è incoraggiato a criticare il sistema
 - possibilità di chiarimenti

| Protocol analysis

protocollo = registrazione della sessione di valutazione

- carta e matita - economica, limitata alla velocità di scrittura
- audio - buono per le riflessioni ad alta voce, difficile da abbinare ad altri protocolli
- video - accurato e realistico, necessita di attrezzature speciali, invadente
- registrazione al computer - automatica e non invasiva, grandi quantità di dati difficili da analizzare .
- taccuini dell'utente - grossolani e soggettivi, utili approfondimenti, buoni per gli studi longitudinali
- Uso misto nella pratica
- Trascrizione audio/video difficile e richiede abilità <https://archive.mpi.nl/tla/elan>
- Sono disponibili alcuni strumenti di supporto automatico

| Post-task walkthroughs

- trascrizione riproposta al partecipante per un commento
- immediato → fresco di mente
- ritardato → il valutatore ha il tempo di identificare le domande
- utile per identificare le ragioni delle azioni e le alternative considerate
- necessario nei casi in cui non è possibile pensare ad alta voce

| ATTENZIONE

| Questi metodi sono soggettivi

- dipendono dalle conoscenze e capacità del valutatore
- il valutatore deve riconoscere i problemi e capire cosa fa l'utente
- evaluator bias: lo si può attenuare mediante l'uso di più valutatori
- anche la risposta dell'utente è soggettiva

| Misure qualitative

- misure prodotte da questi metodi: non numeriche
- possono rivelare dettagli non determinabili da misure numeriche
- generalmente i metodi soggettivi portano misure qualitative

| Ethical issues

- test anonimi (id anonimi, informazioni aggregate)
- ricompense per i partecipanti

| ESPERIMENTI

- valutazione controllata di aspetti specifici del comportamento interattivo comportamento
- il valutatore sceglie l'ipotesi da testare (determinata misurando il comportamento dei partecipanti)
- si considera un certo numero di condizioni sperimentali che differiscono solo per il valore di qualche variabile controllata.
- i cambiamenti nelle misure comportamentali sono attribuiti alle diverse condizioni

| Fattori

- Soggetti: chi - campione rappresentativo e sufficiente
- Variabili: cose da modificare e misurare
- Ipotesi: ciò che si vorrebbe dimostrare (un'idea provvisoria il cui valore dev'essere accertato)
- Disegno sperimentale: come si intende procedere

| Variabili

- **Variabile Indipendente** (IV):
 - caratteristica modificata per produrre condizioni diverse
 - ad esempio, lo stile dell'interfaccia, il numero di voci del menu
- **Variabile Dipendente** (DV):
 - caratteristiche misurate nell'esperimento
 - ad esempio, il tempo impiegato, il numero di errori.

| IPOTESI

- **ipotesi alternativa**:
 - previsione che una variazione di IV causerà una differenza in la DV
 - Ad esempio, “il tasso di errore aumenterà al diminuire della dimensione dei caratteri”.
- **ipotesi nulla**:
 - afferma che non ci sarà alcuna differenza nella DV
 - L'obiettivo è confutare questa ipotesi
 - Ad esempio, “nessun cambiamento con la dimensione dei caratteri”.

| DESIGN DELL'ESPERIMENTO

- scegliere l'ipotesi

- scegliere le variabili dipendenti e indipendenti (le v.i. devono essere il più “semplici” possibile!)
- scegliere i partecipanti
- scegliere il metodo sperimentale: tra soggetti o all'interno dei soggetti

| Condizioni

- **condizione di controllo**
- **condizione sperimentale:** solo una v.i. modificata rispetto alla condizione di controllo

la condizione di controllo serve ad assicurarsi che è il cambiamento della v.i. il responsabile del cambiamento misurato (eventualmente) nelle v.d.

| Between subjects

- a ogni partecipante viene assegnata una sola delle condizioni, egli quindi svolge il test una volta sola.
- sono richiesti più partecipanti
- variazioni tra gli utenti possono alterare il risultato della valutazione
- non risente del “trasferimento dell'apprendimento”

| Within subjects

- a ogni partecipante vengono assegnate tutte le condizioni, egli quindi esegue il test 2 o più volte
- questo metodo sperimentale richiede meno partecipanti, quindi è meno costoso
- le differenze tra gli utenti influenzano di meno la valutazione
- risente del “trasferimento dell'apprendimento”

| Mitigare il trasferimento dell'apprendimento

- metà dei partecipanti esegue il test dapprima sotto la condizione di controllo, poi sotto la condizione sperimentale
- l'altra metà esegue il test dapprima sotto la condizione sperimentale, poi sotto la condizione di controllo

| ANALISTI STATICHE DEI DATI

- osservare i dati raccolti (DV)
- possibilmente su un grafico
- ricerca di outliers
- salvare i dati originali (potrebbero essere necessari in seguito)

| Tipi di dati

- variabili discrete o continue
- variabili continue positive (> 0). Es. tempo impiegato
- variabili indipendenti tipicamente discrete
- distribuzione normale della variabile dipendente

| Analisi statistica

- se i dati soddisfano la distribuzione normale:
 - test parametrici (tecniche di analisi statistica standard molto robuste)
- altrimenti:
 - test non parametrici, tabella di contingenza,...
- verifiche d'ipotesi:
 - sì/no al rifiuto dell'ipotesi nulla
 - esempio: "possiamo, al 99%, rifiutare l'ipotesi nulla"

| A/B TESTING

- il più semplice esperimento di usabilità
- una sola variabile indipendente con due valori (A e B)
- due condizioni (A e B)
- usato di solito per test online (e.g., sito web): ogni utente riceve a caso una delle due condizioni

| Ipotesi e risultati

- **ipotesi nulla:**
 - non c'è differenza tra A e B
- **ipotesi alternativa** (da dimostrare):
 - c'è una differenza statisticamente significativa tra A e B
- test statistico più comune (distribuzione Gaussiana): t-test

| Pro e contro

PRO:

- semplice da progettare e realizzare
- risultato facile da misurare
- risultato molto chiaro

CONTRO:

- adatto solo al caso di due condizioni diverse
- difficile decidere cosa misurare (A e B) in alcuni casi

ESEMPIO DI VALUTAZIONE DAL PROFESSORE

| **Contesto:**

Vorremmo testare l'usabilità di un pulsante di conferma all'interno di una schermata di un'app
condizione di controllo: il pulsante è posizionato in basso, ha uno sfondo bianco e un bordo blu

| **Dettagli**

in particolare, vorremmo testare:

- la posizione:
 - in alto, oppure
 - in basso nella schermata
- abbiamo la possibilità di reclutare 100 potenziali utenti per svolgere un esperimento quantitativo, in cui valuteremo l'usabilità delle varie versione del pulsante in termini di:
- tempo impiegato a cliccare il pulsante
 - numero di tap effettuati prima di quello sul pulsante

| **Descrizione esercizio**

- descrivere in un paragrafo tutti i possibili tipi di test che si potrebbero svolgere
- in particolare, in termini di:
 - variabili indipendenti
 - variabili dipendenti
 - condizioni (valori delle variabili indipendenti)
 - ipotesi nulla
 - tipo di test between o within subjects e numero di utenti per condizione

| **RISULTATO:**

Variabili indipendenti: posizione del pulsante di conferma (sopra o sotto)

Variabili dipendenti: tempo impiegato a cliccare un pulsante, numero di tap effettuati prima di quello sul pulsante

Condizioni: gli stati del pulsante (pulsante sopra, pulsante sotto)

Ipotesi nulla: non c'è cambiamento nella posizione dei pulsanti

Tipo di Test:

- **between:** A 50 persone viene assegnato il pulsante in alto, mentre alle 50 viene assegnato il pulsante in basso.

- **within**: A 50 persone viene prima assegnato il pulsante in alto e poi quello in basso, per gli altri 50 vengono invertiti.

| 10 - Tecniche di Valutazione con ESPERTI

E' importante valutare spesso durante lo sviluppo più tardi si trova un errore, più è costoso correggerlo coinvolgere gli utenti in tutte le valutazioni è costoso.

| Effettuare le valutazioni senza utenti

uno o più esperti valutano l'impatto di un dato design sull' "utente tipico" per individuare aspetti che possono causare difficoltà:

- perché violano noti principi cognitivi
- perché ignorano risultati empirici accettati

| Vantaggi e svantaggi

- poco costosi
- utilizzabili in tutte le fasi di progetto e sviluppo
- non valutano il vero uso del sistema, ma solo l'aderenza ai principi conosciuti

| Expert based evaluation

- **Cognitive walkthrough**
- **Heuristic evaluation**
- **Model-based evaluation**
- **Review-based evaluation**

| Cognitive walkthrough

- proposto da Polson et al., 1992
- valuta il design in base alla sua capacità di supportare l'utente nell'apprendimento di un compito: quanto è facile imparare
- di solito viene eseguita da un esperto di psicologia cognitiva
- L'esperto "attraversa" il progetto per identificare i potenziali problemi utilizzando principi psicologici

| Per eseguire un cognitive walkthrough servono:

- specifiche del sistema (com'è fatta e come funziona l'interfaccia)
- descrizione di un task
- descrizione dell'utente tipo del sistema
- sequenza dei passi necessari per compiere il task

| Come si fa:

- per ogni passo, l'esperto tenta di rispondere alle seguenti domande:
 - **l'effetto dell'azione è lo stesso dell'obiettivo dell'utente?**
 - **l'utente può vedere che l'azione è disponibile?**
 - **l'utente capirà che quella è l'azione da eseguire?**
 - **dopo averla eseguita, capirà il feedback ricevuto?**
- l'esperto riporta su un form tutti i problemi e il loro grado di severità

VEDERE ESEMPIO SU SLIDE

| Heuristic Evaluation

- Euristica = linea guida, o principio generale, o regola pratica semplice ed efficiente
- per guidare le decisioni di progetto o per criticare decisioni già prese
- J. Nielsen e R. Molich (1994) hanno proposto 10 euristiche per scoprire problemi di usabilità dei sistemi interattivi
- Nella valutazione euristica, il valutatore esamina il progetto per vedere se viola una o più euristiche
- Per ogni violazione, riporta la severità del problema

| Le dieci euristiche di Nielsen

1. visibilità dello stato del sistema
2. corrispondenza tra il sistema e il mondo reale
3. libertà e controllo da parte degli utenti
4. coerenza e standard
5. prevenzione degli errori
6. riconoscere piuttosto che ricordare
7. flessibilità ed efficienza d'uso
8. design minimalista ed estetico
9. aiutare gli utenti a riconoscere, diagnosticare e correggere gli errori
10. aiuto/guida e documentazione

| Model-based evaluation

L'uso di modelli esistenti permette di valutare l'interazione

Esempi:

- GOMS (Goals, Operators, Methods, Selections) — modello di previsione delle performance dell'utente con un'interfaccia
- KLM (Keystroke-Level Model) — modello di previsione dei tempi necessari per compiti fisici di basso livello (uso tastiera e mouse)

- Dialog models — per valutare le sequenze dei dialoghi, stati irraggiungibili, dialoghi circolari, ...

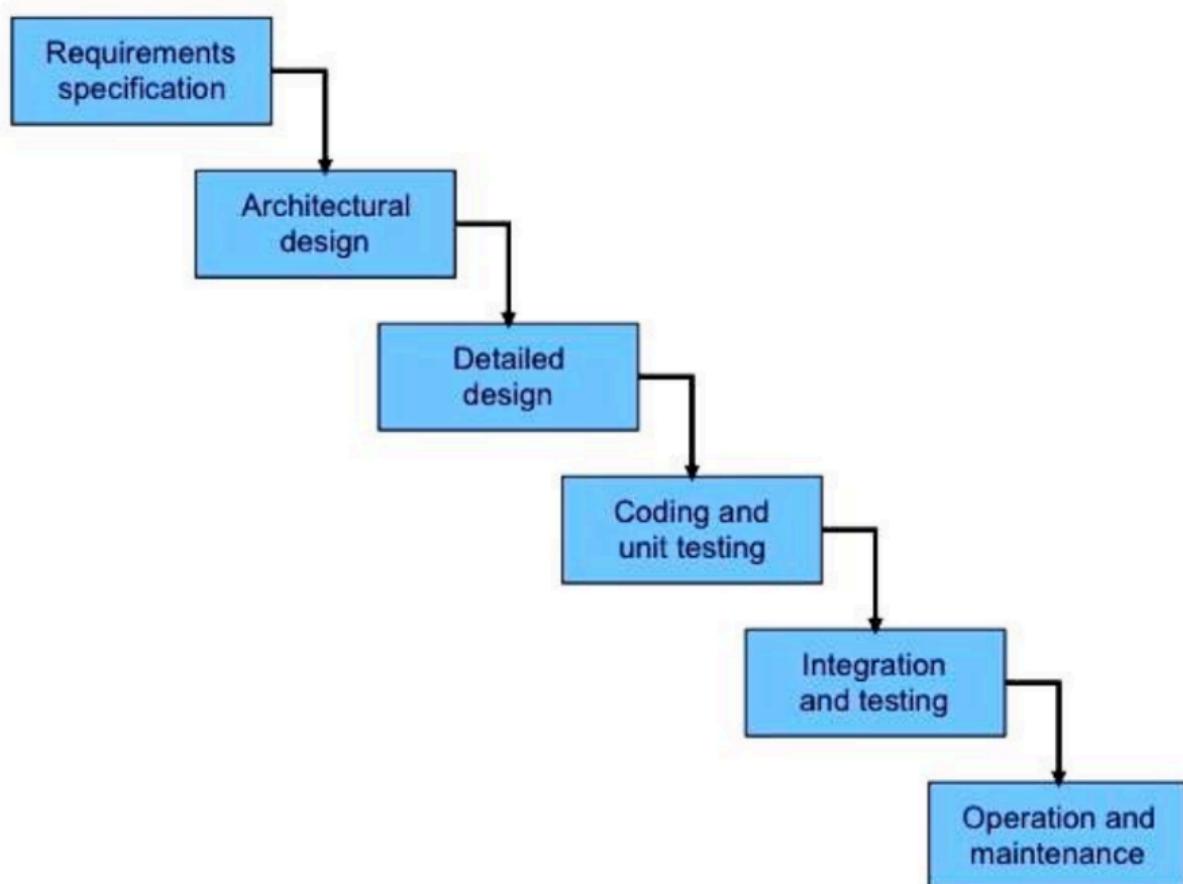
| **Review-based evaluation**

- risultati di studi precedenti
- utilizzati per supportare o rifiutare parti della progettazione
- garantire che i risultati siano trasferibili al nuovo progetto
- è necessario un esperto per garantire che le ipotesi corrette siano state formulate

| 11 - Progetto e Sviluppo di Sistemi Interattivi

| Waterfall e Agile

| Modello Waterfall



AGGIUNGI SLIDE 13 E 14

| 12 - Modi

| TERMINOLOGIA

| content

l'insieme di informazioni che risiedono in un sistema e che hanno significato e utilità per l'utente

| GID

(Graphical Input Device): meccanismo per comunicare al sistema una particolare locazione o la scelta di un oggetto (tipicamente la posizione del cursore)

| GID button

bottone principale del GID

| tap

l'azione di premere e rilasciare un tasto (che automaticamente ritorna nel suo stato originale)

| click

posizionare il GID e fare un tap sul GID button

| double-click

posizionare il GID e fare due click di seguito senza spostare il GID tra i due click

| drag, swipe

premere il GID button, senza rilasciarlo muovere il GID, e poi rilasciarlo (in un'altra locazione)

| Gesture (gesto)

una sequenza di azioni completata automaticamente una volta avviata

- es. scrivere una parola comune (es. "che"), oppure premere il tasto "invio"

| Modi

I modi si manifestano nel modo in cui un'interfaccia risponde ai gesti

- dato un gesto, un'interfaccia è in un modo se l'interpretazione di quel gesto è sempre la stessa
- quando il gesto viene interpretato in maniera diversa, l'interfaccia si trova in un altro modo
 - es. il tasto invio può essere interpretato come "andare a capo" o "confermare l'inserimento" o "seleziona elemento" (di una select)
- la presenza di modi può essere un problema?

| Esempio - torcia

- una torcia che si accende e spegne con un pulsante a pressione
- se ho la torcia in tasca posso dire che succederà se premo il pulsante?
- l'interfaccia dell'esempio presenta 2 modi:
 - non permette di stabilire quale operazione si debba effettuare per raggiungere un dato risultato solamente ispezionando il meccanismo di controllo
 - è necessario verificare separatamente lo stato del sistema

| Esempio - sveglia

- bottone per attivare o disattivare la sveglia (già impostata):
 - se la sveglia è attiva, premendo il bottone la si disattiva
 - se la sveglia è disattivata, premendo il bottone la si attiva
- al buio non si vede il display: non conoscendo lo stato del sistema, non si può stabilire l'effetto della pressione del bottone

| "Luogo dell'attenzione"

oggetto fisico o idea alla quale stiamo pensando attivamente e intenzionalmente
può esserci un solo luogo dell'attenzione alla volta!

quali sono le implicazioni?

se qualcosa accade al di fuori del luogo dell'attenzione dell'utente che succede?
concetto collegato a ma diverso da "focus"

| Esempio - Caps Lock

- il tasto di blocco maiuscole crea un modo
- SPESO SI INIZIA A SCRIVERE E SOLO DOPO ALCUNE PAROLE CI SI ACCORGE DELL'ERRORE
- eppure:
 - un LED verde sul tasto stesso indica chiaramente lo stato del sistema
 - alcune applicazioni hanno un indicatore di caps lock nella barra di stato
- allora perché si commette questo errore?
- si può prevenire in qualche modo?

| Utente esperto e principiante

i modi possono comunque essere utili, aggiungendo flessibilità ma possono indurre in errore l'esperienza non protegge dagli errori dovuti ai modi: l'esperto ha acquisito delle abitudini anche l'inesperienza non protegge

| Minimizzare i 'mode errors'

Mode Errors: "errori che si verificano quando l'utente sbaglia a classificare, o quando l'utente fa un'analisi errata di una situazione" (Norman 1981).

1. non avere modalità
2. assicurarsi che le modalità siano contrassegnate in modo distintivo
3. assicurarsi che i comandi richiesti dalle diverse modalità non siano gli stessi, in modo che un comando impartito nella modalità sbagliata modalità sbagliata non comporti difficoltà (di fatto, il comando non produrrà alcun effetto) (Norman, 1983)

| Interfaccia Modale

Definizione:

un'**interfaccia è modale** rispetto a un gesto quando:

- lo stato corrente dell'interfaccia non è il luogo dell'attenzione dell'utente
- l'interfaccia risponderà al gesto con una tra $N > 1$ possibili risposte, a seconda dello stato del sistema

| Interfaccia non modale

un'interfaccia può essere modale rispetto a uno o più gesti e non modale rispetto a un altro o vari altri gesti

- un'interfaccia è non modale se non è modale rispetto a qualunque possibile gesto
- si può dare una misura precisa di quanto un'interfaccia è modale come probabilità che un gesto non modale venga usato, dove 0 = interfaccia completamente modale, 1 = interfaccia totalmente non modale
- il gesto "cancella carattere" (tasto backspace) è modale? perché?

| Modi particolari

- modo temporaneo: un modo che svanisce dopo l'uso (esempio: pennello di word, attivato con un solo click)
- quasi-modo: un modo ottenuto attivando e mantenendo fisicamente un controllo: es. tasti maiuscole, tasto CTRL

| Interazione noun-verb e verb-noun

- eseguire azioni (verb) su oggetti (noun)
- noun-verb: si seleziona prima l'oggetto
- verb-noun: si seleziona prima l'azione
- anche chiamati object-action e action-object

I Esempi

cambio font: seleziono un testo, scelgo un nuovo font (verb-noun o noun-verb?)

palette di strumenti: seleziono un tool da una palette di strumenti e poi lo uso (verb-noun o noun-verb?)

I Pro e contro

I verb-noun

- si sposta il luogo dell'attenzione sull'azione e la si seleziona
- si sposta il luogo dell'attenzione sull'oggetto
- in caso di distrazione e spostamento imprevisto del luogo dell'attenzione, ci saranno errori di modo

I noun-verb

- si seleziona l'oggetto mentre esso è nel luogo dell'attenzione
- il luogo dell'attenzione si sposta sull'azione e la si attiva (in totale un solo spostamento invece di 2!)
- interrompere l'azione non richiede un'altra azione (cancel o escape): semplicità e reversibilità

I Esempio di passaggio da v-n a n-v

- . contesto: sistema per richiedere prodotti in una multinazionale
- . flusso originale (v-n):
 - . l'utente deve prima selezionare il dipartimento che richiede i prodotti (azione: "dove inviare") e poi scegliere gli articoli da inviare
 - . il sistema setta di default il dipartimento della postazione in uso; questo crea confusione se l'utente utilizza una postazione in un dipartimento diverso dal suo
 - . problema: gli utenti spesso dimenticano di modificare il dipartimento di default, causando errori di invio di prodotti
 - . il flusso non segue il naturale ordine dell'attenzione dell'utente, che si focalizza prima sugli articoli da richiedere e poi su dove inviarli

- . soluzione (passaggio a n-v):

- . spostare la scelta degli articoli in cima, come primo passo dell'ordine
- . aggiungere una descrizione chiara: "a quale dipartimento vuoi che gli articoli siano inviati?"
- . identificare il dipartimento dell'utente tramite login e aggiungere un'opzione "Il mio dipartimento" in cima alla lista

- . benefici della nuova interfaccia:

- . la scelta del dipartimento diventa un'azione finale, che chiude in modo logico il processo
- . riduzione degli errori e miglioramento del flusso naturale di attenzione
- . eliminazione del clic di conferma finale, riducendo le azioni necessarie per completare l'ordine

| 13 - Affordances e significanti

Vedi Slide 17bis

| 14 - Progetto di applicazioni mobili

| Contesto d'uso

| Utente di un dispositivo mobile

- in movimento, non concentrato a lungo
- esegue i compiti nei ritagli di tempo
 - poco tempo a disposizione
 - fretta
- frequentemente interrotto (in modo asincrono) da eventi esterni
- a volte ha necessità di non disturbare altri (ad esempio in un cinema/teatro)

| Attenzione

possibile richiesta di attenzione "contemporanea" da parte di altre attività (es. guida di veicoli)

vista/udito/mani impegnate

condizioni di illuminazione scarse/variabili, rumori

| DIFFERENZE RISPETTO AL DESKTOP

| Schermo dello smartphone

- piccole dimensioni: 5"-7"
- alta risoluzione, es. >1Mpixel
- leggibilità
- affollamento: non c'è spazio per inserire elementi non strettamente necessari nell'interfaccia

| Memoria limitata

- eliminare memory leaks
- ridurre le dimensioni dei file allo stretto indispensabile / formati

| Una schermata alla volta

- non ci sono finestre
- l'utente accede alle varie schermate sequenzialmente
- porting di applicazioni che richiedono più finestre sul desktop/browser:
 - riorganizzarle su schermate sequenziali
 - fare il porting solo di alcuni subtask

| Un'applicazione alla volta

- no multitasking (per app di terze parti)
- quando arriva una telefonata l'applicazione corrente viene interrotta (se rispondi)
- un'applicazione può essere interrotta in modo asincrono, quindi:
 - non deve perdere i dati e lo stato
 - deve poter ripartire rapidamente dallo stato in cui si trovava
- impostazioni fuori dall'applicazione: necessità di variarle interrompe l'applicazione

| Help minimale

- gli utenti di un dispositivo mobile non hanno tempo, voglia, modo di leggere l'help
- l'help non deve togliere spazio sullo schermo
- usare controlli standard (gli utenti già li conoscono)
- cercare sequenze logiche delle schermate dell'applicazione, comprensibili
- dotare le schermate di buttoni back per tornare indietro

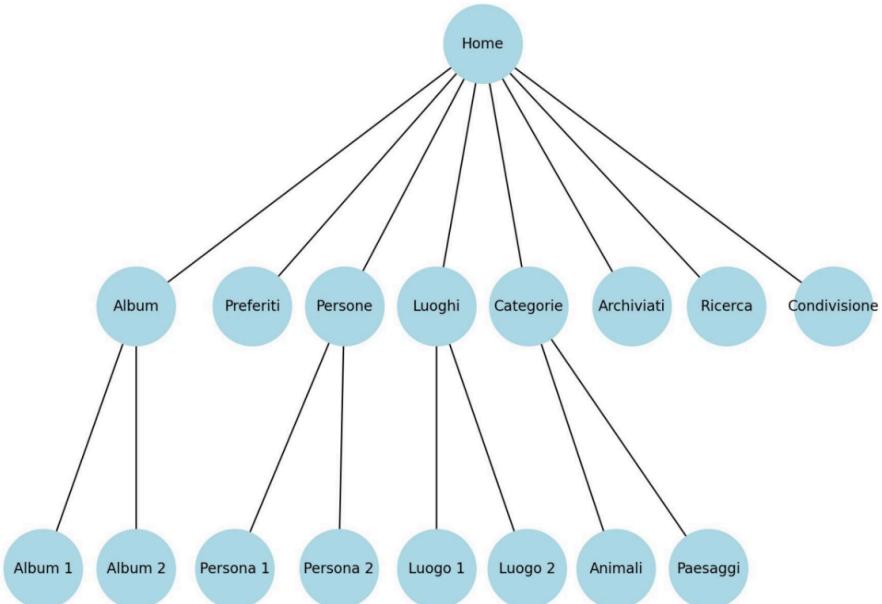
| 14.1 - Stili di applicazioni mobili

| Productivity application

- permette di svolgere compiti basati sulla organizzazione e manipolazione di informazioni dettagliate
- task importanti
 - es: mail
- user experience focalizzata sul task
- "poca" (dettagliata) informazione, molta interazione

Esempio: Google Photo

Struttura Gerarchica di Navigazione in Google Foto



| Dati organizzati gerarchicamente

- interazione tipica:
 - organizzare la lista
 - aggiungere e togliere elementi dalla lista
 - scendere a livelli di dettaglio successivi e poi eseguire operazioni sul livello scelto

| UI per productivity app

- una vista (schermata) per ogni livello
- interfaccia semplice e pulita
- uso di controlli standard
- enfasi sulle informazioni e sul task, non sull'ambiente o sull'esperienza
- impostazione di preferenze per ridurre informazioni e scelte ripetitive

| Utility application

- task semplice che richiede poco input dall'utente
- leggere informazioni essenziali su qualche argomento
- verificare lo stato di qualcosa
- zero struttura gerarchica
- esempio: previsioni meteo
- molta informazione, poca interazione

Esempio di utility application



| Liste nelle utility app

- una o più liste, non gerarchicamente collegate
- esempio: meteo in diverse città
- interazione: lettura delle informazioni, cambio di impostazioni o della sorgente dei dati

| Preferenze nelle utility app

- frequenti cambiamenti alla configurazione
- (es. aggiungere le previsioni del tempo di una città, mostrare o togliere le temperature minime, etc)
- necessità di modificare le preferenze dall'interno dell'applicazione

| Immersive application

- per compiti che:
 - presentano un ambiente particolare
 - non mostrano grandi quantità di testo

- richiedono l'attenzione continuata dell'utente
- ad esempio: videogiochi
- caratteristiche:
 - ambienti ricchi
 - a tutto schermo
 - focalizzate sul contenuto e sull'esperienza di quel contenuto

Esempio: IKEA Place



| Interazione con immersive app

- una immersive app non usa i controlli standard ma ne crea di propri
- cercare i controlli e scoprirne il funzionamento è parte dell'esperienza di una immersive application
- usa spesso grandi quantità di informazione, ma la mostra in un modo particolare legato al contesto

| 15 - Productivity app e modi

| App di produttività

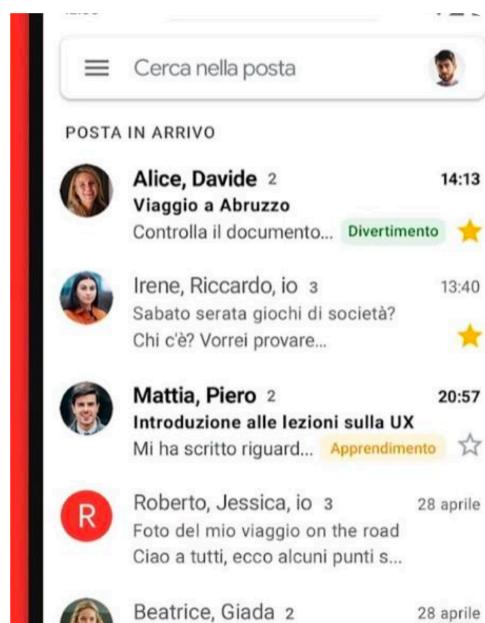
- esperienza focalizzata sull'esecuzione di task
- esempi:
 - mail
 - social
 - home banking
- task tipici: creare e manipolare oggetti come
 - post,
 - messaggi,
 - contatti,
 - chiamate telefoniche • movimenti bancari

| Approccio tipico

- elenco di oggetti
- livelli gerarchici (albero n-ario)
- task da svolgere

gmail (per mobile)

- lista conversazioni
- tap su conversazione per vedere le mail che la compongono in una nuova view (quindi navigazione nell'albero a partire dalla radice scendendo nei figli)
- swipe per effettuare azioni su una conversazione (es. archivia o segna come non letta/non letta)
- tap sul bottone della penna per creare una nuova conversazione



| Approccio tipico: noun-verb

- l'approccio tipico per eseguire operazioni in un'app di produttività è noun-verb (oggetto+azione)
- scelgo un oggetto e poi eseguo l'azione:
 - swipe per archiviare una conversazione

- tap su pulsante per eliminare una conversazione
- tap su conversazione per aprirla
- l'animazione dell'oggetto mi fa capire che l'azione è avvenuta con successo
- **non serve dare un feedback aggiuntivo** (ad esempio con un alert)

| Perché è meglio noun-verb

- l'attenzione dell'utente è ridotta
- l'utente usa l'app nei ritagli di tempo
- può essere interrotto (in modo asincrono)
- non conosce o non ricorda lo stato del sistema, che può non essere nel suo luogo dell'attenzione
- con noun-verb non creo un modo
- quindi non si verificano errori di modo

| Animazione delle view

- la vista figlia entra da destra e si sovrappone alla vista padre
- la vista figlia ha un bottone 'back' in alto a sinistra per tornare alla vista padre
- premendo back, l'animazione mostra la vista figlia scorrere via verso destra, scoprendo di nuovo la vista padre
 - è come se le viste fossero impilate una sull'altra, come se creassero una pila (l'ultima entrata sarà la prima a uscire)

| Aggiunta di un oggetto: verb-noun

- l'aggiunta di un oggetto invece è verb-noun
- esempio: premo 'scrivi' (bottone con icona della matita) e poi posso iniziare a scrivere
- si apre una vista **modale**: infatti sono in un modo, dove posso solo scrivere una mail
- le viste modali 'salgono' dal basso, coprendo (in parte o completamente) la vista precedente
- al termine della scrittura della mail premo enter e la vista modale si chiude, scendendo verso il basso e scoprendo la vista in cui ci trovavamo prima
- nella vista iniziale sarà presente un nuovo elemento nella lista degli oggetti, che comparirà con un'animazione
- questo è già un feedback per l'utente, **non serve dare un feedback aggiuntivo** (ad esempio con un alert)

| Vista di default

- molte app per smartphone iniziano dalla vista radice dell'albero
- questo, insieme all'approccio noun-verb, conferisce all'app:
 - rapidità

- semplicità d'uso
- minor numero di errori

esempi: Gmail, Whatsapp, Telegram, Google Drive, sveglia, rubrica, BancoPosta

| 16 - Design patterns

- I design pattern sono stati utilizzati per la prima volta in architettura
- L'idea è stata introdotta dall'architetto Christopher Alexander, che ha definito un design pattern in questo modo:
 - *“Ogni modello descrive un problema che si ripresenta continuamente nel nostro ambiente, e poi descrive il nucleo della soluzione a quel problema, in modo tale che si possa usare questa soluzione un milione di volte, senza mai farlo due volte nello stesso modo”.*
- Sono un modo per comunicare i **problemi** comuni **di progettazione** e le relative **soluzioni**
 - I design pattern sono soluzioni ricorrenti che risolvono i problemi comuni di progettazione
 - Non sono troppo generici e non sono troppo specifici
 - Utilizzano una soluzione “un milione di volte, senza mai farla allo stesso modo due volte”
 - Sono un linguaggio condiviso
 - I design pattern sono punti di riferimento standard per i progettisti
 - Consentono un dibattito sulle alternative, in cui la semplice menzione del nome di un design pattern ha implicitamente un significato molto più ampio del semplice nomeo
 - Sono leggibili dai non esperti

| UI Design Patterns

- Ogni interfaccia utente è unica e ha i propri obiettivi e dati, ma...o
 - Questo non significa che dobbiamo costringere gli utenti a imparare nuove convenzioni per poterle utilizzare!
- Con i modelli di progettazione dell'interfaccia utente, possiamo accelerare la comprensione dell'interfaccia da parte degli utenti.

UI Design Patterns

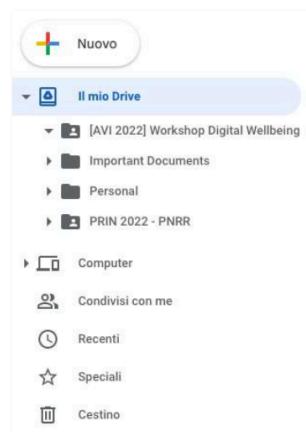
ACCORDION MENU

DROPDOWN MENU

CARDS

BREADCRUMBS

THE HAMBURGER



UI Design Patterns

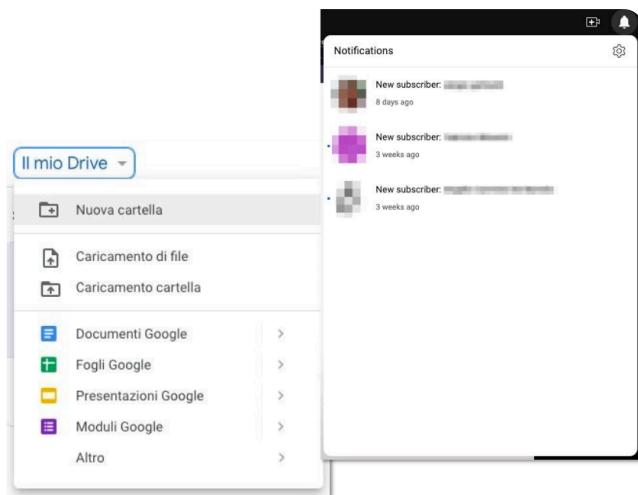
ACCORDION MENU

DROPDOWN MENU

CARDS

BREADCRUMBS

THE HAMBURGER



UI Design Patterns

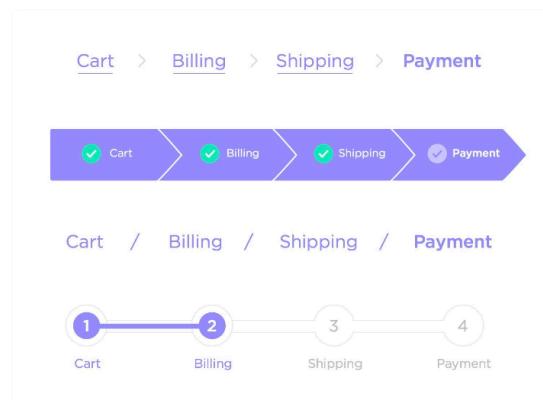
ACCORDION MENU

DROPDOWN MENU

CARDS

BREADCRUMBS

THE HAMBURGER



UI Design Patterns

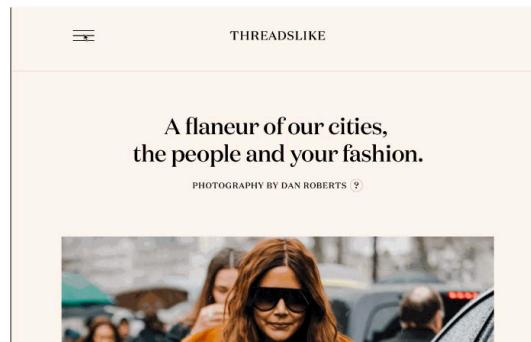
ACCORDION MENU

DROPODOWN MENU

CARDS

BREADCRUMBS

THE HAMBURGER



VEDI DALLE SLIDE DESIGN PATTERS IN ANDOID

| DARK PATTERNS

Il termine “dark pattern” è stato coniato nel 2010 da Harry Brignull, un professionista del design.

L'obiettivo era quello di includere tutti quei disegni che sono deliberatamente adottati per promuovere scelte che non sono nel migliore interesse dell'utente

| Dark Patterns ...or Deceptive Designs

- Molte organizzazioni si stanno allontanando dalla terminologia opprimente tipica dell'informatica. nell'informatica
 - Ad esempio da “master/slave” a “genitore/figlio” e da “lista nera” a “lista di blocco”.
 - Nel caso dei “modelli oscuri”, l'associazione di “oscuro” con il danno è problematica
 - può rinforzare l'euristica razzista che vede le persone con la pelle più scura come scure come malvagie (effetto “il cattivo è nero”).
- Al giorno d'oggi esistono anche nomi alternativi per i disegni scuri, come ad esempio disegni ingannevoli

| Attention-Capture Damaging Patterns

| Attention Economy

- Perché il nostro benessere digitale è minato dalla tecnologia contemporanea?
- La maggior parte delle aziende tecnologiche contemporanee adotta un modello di business che si chiama chiamato “economia dell'attenzione”
 - La nostra attenzione viene trasformata in una moneta
 - “paghiamo” per un servizio con il tempo che vi dedichiamo
 - Le aziende tecnologiche possono vendere la nostra attenzione (e i nostri dati, a volte) a aziende pubblicitarie
- Questo modello di business è conveniente

- Alphabet (la società che possiede Google) ha un valore di 1.000 miliardi di dollari
- Meta (che possiede Facebook, Instagram e WhatsApp) vale circa 700 miliardi di dollari

| Attention-Capture Damaging Patterns: Definizione

Un modello ricorrente nelle interfacce digitali che il designer **utilizza per sfruttare le vulnerabilità psicologiche e catturare l'attenzione.**

Vulnerabilità psicologiche e catturare l'attenzione, portando spesso l'utente a perdere di vista i propri obiettivi, a perdere il senso del tempo e del controllo e a sentirsi a perdere di vista i propri obiettivi, a smarrire il senso del tempo e del controllo e, in seguito, a provare rammarico, rimpianto

L'obiettivo delle ACDP è quello di massimizzare l'uso continuo, le visite giornaliere e le interazioni (ad esempio, le interazioni con i siti web). interazioni (ad esempio, click, condivisioni, like, ecc.).

Rendono gli utenti più propensi a visitare nuovamente un servizio digitale e a cliccare su tipi simili di contenuti gratificanti. di contenuti simili e gratificanti, creando così una “**trappola**” per l'utente che consente di raggiungere l'obiettivo dello stakeholder. **l'obiettivo dello stakeholder**

| Attention-Capture Damaging Patterns: Strategie

In genere, riducono la necessità di prendere decisioni autonome “automatizzando” i processi e le funzioni.

Paradossalmente, possono **migliorare l'usabilità** di una piattaforma.

Sfruttano le vulnerabilità **psicologiche degli utenti**:

- Ricompensa variabile
- gratificazione immediata

Le vulnerabilità psicologiche sfruttate possono essere di vario tipo:

- Le notifiche (con le loro vibrazioni, i LED lampeggianti...) imitano i segnali di pericolo che possiamo sperimentare nella vita “offline”. che possiamo sperimentare nella vita “offline”
- La possibilità di ricevere nuovi commenti o like ci mantiene in uno stato di allerta persistente (ricompensa variabile). stato di allerta (ricompensa variabile)
- Ogni volta che riceviamo un commento o un “mi piace”, il nostro cervello riceve una dose di dopamina, spingendoci a confrontarci con gli altri (influenza sociale).
- In generale, la tecnologia odierna è progettata per tenerci “impegnati”, offrendoci un mix di contenuti vecchi e nuovi, ogni volta
 - un meccanismo che viene utilizzato anche dalle slot machine

| Attention-Capture Damaging Patterns: Impatto

Le ACDP possono influire negativamente sul benessere digitale delle persone

- Promuovono la “dipendenza digitale”
- Minano l'attenzione e la produttività degli utenti
- Minano il senso di agency e di autocontrollo degli utenti
- Provocano un successivo senso di rimpianto

| Attention-Capture Damaging Patterns: Esempi

| GUILTY-PLEASURE RECOMMENDATIONS

Si basano su:

- Interazioni dell'utente precedente (basate sul contenuto)
- Preferenze di utenti simili (Collaborative Filtering)

Le raccomandazioni sono utili se l'obiettivo della piattaforma coincide con gli obiettivi dell'utente (valore-alignment problem).

Possono diventare una “**trappola**” per mantenere l'attenzione dell'utente sulla piattaforma
Sfruttano **tecniche di ricompensa variabile**

| NEVERENDING AUTOPLAY

Un nuovo video viene riprodotto automaticamente al termine di quello in corso.

Non c'è mai un punto in cui l'utente possa **fermarsi a riflettere** e l'opzione per disattivare l'autoplay è nascosta o inesistente

L'autoplay può essere **utile** in alcune circostanze, ad esempio, per ascoltare la musica su YouTube mentre si lavora.

Può **prolungare** le sessioni di utilizzo (“fammi vedere questo nuovo video e poi chiudo l'app”).

Sfrutta le **tecniche di ricompensa variabile**

Riduce l'**autonomia dell'utente**

| CASINO PULL-TO-REFRESH

Quando l'utente passa il dito sul suo smartphone, si verifica un ricaricamento animato della pagina **che può rivelare o meno** un nuovo contenuto

Gli utenti possono essere tentati di aggiornare compulsivamente **la pagina, sperando di ricevere nuovi contenuti**

Sfrutta le tecniche di ricompensa variabile**

Simile alle slot machine

| INFINITE SCROLLING

Man mano che l'utente scorre una pagina verso il basso, **altri contenuti vengono automaticamente e continuamente i contenuti in basso.**

Diminuisce lo **sforzo necessario** per sfogliare i contenuti

Promuove **sessioni** di utilizzo “**infinite**”

Sfrutta le **tecniche** di **ricompensa variabile**

| 17 - Interfacce Android

Vedi tutte le slide +

| ELEMENTI INTERFACCE ANDROID

| Cards

- utilizzare una scheda per visualizzare contenuti e azioni su un singolo argomento
- punto di accesso a livelli più profondi di dettaglio o di navigazione
- le schede possono essere visualizzate insieme in una griglia, un elenco verticale o un carosello
- contenuto: titolo, testo, divisorie, media, icone, immagini, pulsanti
- le schede possono essere un unico grande che attiva una schermata di dettaglio

| Liste

- indici verticali continui di testo e immagini
- aiutano gli utenti a trovare un elemento specifico e agire su di esso
- ordinare in modo logico (come alfabetico o numerico)
- 3 dimensioni: una riga, due righe e tre righe
- mantenere gli elementi brevi e facili da scansionare
- mostrare le icone, il testo e le azioni in un formato coerente
- **allineare le immagini al bordo superiore**

| Bottoni

- comunicare azioni che gli utenti possono compiere
- i contenitori hanno angoli completamente arrotondati angoli completamente arrotondati e sono sufficientemente larghi per testo dell'etichetta
- utilizzati in: finestre di dialogo, finestre modali, moduli, schede, barre degli strumenti
- troppi pulsanti su una schermata interrompe la gerarchia visiva
- le icone (opzionali) devono essere posizionate sul lato iniziale del pulsante, prima del testo dell'etichetta

| Floating Action Buttons (FAB)

- l'azione più comune o importante azione su uno schermo
- promuove un'azione costruttiva: creare, preferire, condividere, avviare un processo
- icona chiara e comprensibile
- persiste durante lo scorrimento dei contenuti

- allineata a sinistra, al centro o a destra
- posizionata sopra la barra di navigazione o annidato all'interno della barra
- visualizzato in un quadrato arrotondato contenitore

| Bottom sheets

- visualizzare contenuti supplementari e azioni
- possono contenere voci di menu (in layout a elenco o griglia), azioni
- 2 tipi:
 - **standard** - visualizza contenuti che completano il contenuto principale dello schermo, come ad esempio dello schermo, come ad esempio un lettore audio in un'applicazione un'applicazione musicale
 - **modale** - un'alternativa ai menu in linea o alle semplici finestre di dialogo o a semplici finestre di dialogo su dispositivi mobili; può essere tirato su tutto lo schermo e scorrere internamente (alto)

| Side sheets

- mostrare il contenuto secondario ancorato ai lati dello schermo
- visualizzare un elenco di azioni che influenzano contenuto primario della schermata, come ad esempio filtri
- il ritorno predittivo consente all'utente di scorrere a sinistra o a destra sul foglio laterale
- 2 tipi:
 - **standard** - visualizza contenuti che integrano il contenuto principale dello schermo dello schermo
 - **modale** - preferibile in finestre compatte dimensioni compatte, come quelle dei cellulari; gli stessi tipi di contenuti come quelli standard

| Tabs

- organizzare i contenuti su diversi schermate e visualizzazioni diverse
- raggruppare i contenuti in utili categorie
- posizionate nella parte superiore del riquadro dei contenuti sotto la barra superiore dell'applicazione
- etichetta + icona (opzionale)
- applicare una sottolineatura e un colore al testo e all'icona della scheda attiva icona
- per navigare tra le schede, gli utenti possono passare il dito a sinistra o a destra

| Bottom app bars

- navigazione sul display e tasti azioni
- fino a quattro pulsanti icona e un FAB

- **non dovrebbero essere utilizzati** per applicazioni con una barra di navigazione o schermate con una o nessuna azione
- non sono barre di navigazione
- quando si esegue un'azione distruttiva azione, dovrebbe attivare una barra di per confermare o annullare l'azione

| Top app bar

- e le azioni relative alla schermata corrente. schermata corrente
- navigazione, titoli della schermata e azioni
- contengono un'icona di navigazione icona di navigazione, 2 forme:
 - un'icona di menu (**hamburger**)
 - **una freccia indietro**
- lo scorrimento verso il basso nasconde la barra delle applicazioni
- lo scorrimento verso l'alto rivela la barra delle applicazioni

| Navigation bars

- 3-5 destinazioni di pari importanza importanza
- un tempo si chiamava fondo navigazione
- icone ed etichette di testo opzionali
- utilizzare per le destinazioni di primo livello che devono essere accessibili da da qualsiasi punto dell'applicazione
- non utilizzare per singole attività, come una singola e-mail, le preferenze dell'utente o le preferenze o impostazioni dell'utente

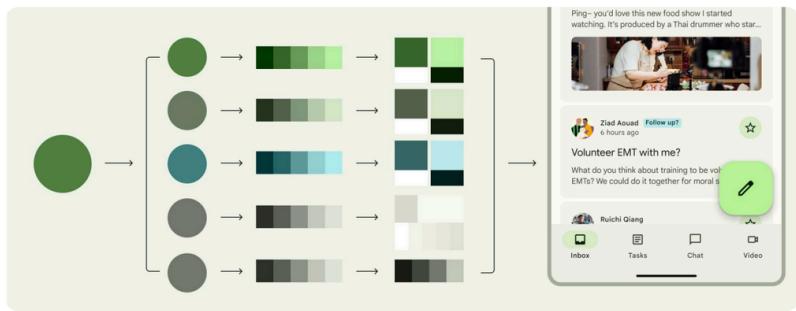
| Transitions

- **avanti e indietro**
 - piattaforma predefinita per la navigazione e indietro (ad esempio, la navigazione gerarchica), navigazione gerarchica)
- **laterale**
 - navigare tra contenuti simili che fanno parte dello stesso insieme, come la navigazione tra le schede di una libreria multimediale

| SISTEMA CROMATICO

- come funziona il sistema cromatico e la creazione di temi statici in Material Design 3
- Material Design 3 definisce, assegna e personalizza i colori per creare interfacce coerenti e accessibili
- elemento centrale: il sistema cromatico, che utilizza algoritmi per generare palette coerenti partendo da un colore di riferimento
- **temi di colore validi anche per iOS!**

- **coerenza visiva**: facilita il branding
- accessibilità integrata: colori e contrasti ottimizzati
- personalizzazione: adattabile a ogni esigenza



| Struttura del Sistema Cromatico

- il modello di colore si basa su tre componenti principali:
 - ruoli dei colori (color roles): etichette che descrivono come e dove usare i colori nell'interfaccia
 - assegnazione dinamica: ogni elemento visivo si collega a un ruolo, che determina il colore assegnato
 - algoritmo della paletta: partendo da un seed color, genera tonalità e livelli di luminosità

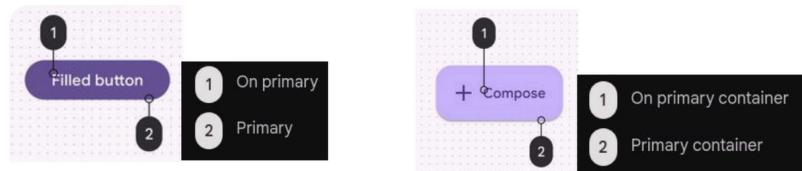
| Cos'è il color role?

- il color role è un'etichetta che associa un elemento visivo a uno scopo specifico
 - un pulsante può essere etichettato come **primary**
 - lo sfondo generale è etichettato come **background**
 - un messaggio di errore usa il ruolo **error**
- il ruolo descrive l'uso dell'elemento, non il colore specifico
- un po' come "colora in base ai numeri"

Primary	Secondary	Tertiary	Error
On Primary	On Secondary	On Tertiary	On Error
Primary Container	Secondary Container	Tertiary Container	Error Container
On Primary Container	On Secondary Container	On Tertiary Container	On Error Container

| Funzionamento dei color roles

- ogni elemento dell'interfaccia viene collegato a un ruolo di colore tramite un'etichetta
- il tema assegna un colore specifico a ciascun ruolo
- il colore finale dipende da
 - modalità del tema (chiara o scura)
 - seed color e tonalità calcolate dall'algoritmo



| Principali ruoli di colore

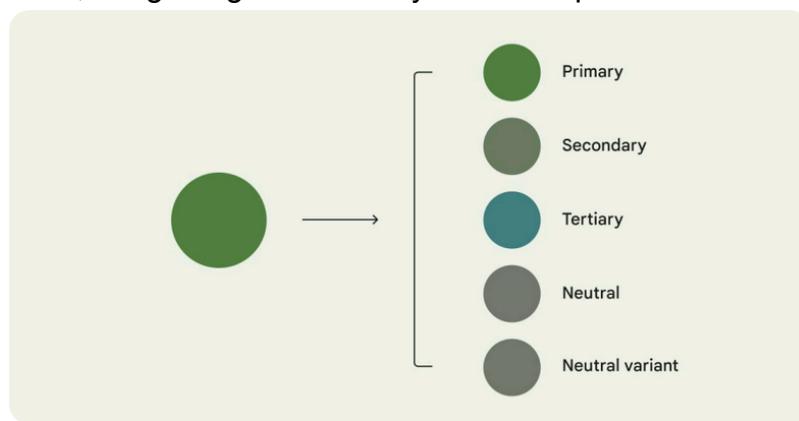
- **primary**: elementi principali interattivi (es. pulsanti)
- **secondary**: accenti visivi e complementi
- **tertiary**: colori opzionali per variazioni stilistiche
- **background**: sfondo dell'interfaccia
- **surface**: superfici come card o finestre
- **error**: stati di errore o avvisi
- **outline**: bordature o contorni di elementi
- **on- roles***: testi o icone sovrapposti a uno sfondo (es. on-primary, on-background)

| Algoritmo per generare temi

- definito in Material Color Utilities (MCU)
- input: seed color (un colore di riferimento)
- passaggi principali
 - generazione di 5 key color complementary
 - generazione di una paletta di tonalità per ogni key color
 - distribuzione dei colori sui ruoli in base alla modalità e al contesto

| 5 key color complementari

A partire dal seed color, vengono generati 5 key color complementari:



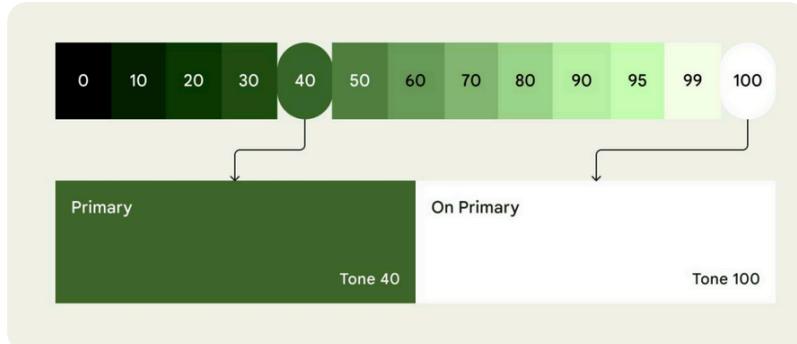
| Palette di tonalità

- per ogni key color viene generata una paletta
 - tonalità chiare: utilizzate per sfondi in modalità chiara
 - tonalità scure: usate per testi in modalità scura
- la paletta contiene 13 livelli di luminosità per ogni colore, distribuiti sui key color



| Assegnazione ai ruoli principali

Usando regole di accessibilità, l'algoritmo assegna gli elementi delle palette ai ruoli:



| Creazione di temi con Figma

- plug-in Material Theme Builder
- semplifica il processo di creazione di temi di colore
- utilizzabile anche per iOS!
- inserire il seed color
- generazione automatica della palette
- assegnazione automatica ai color roles

| Uso con M3 Design Kit in Figma

- creare un prototipo usando componenti di M3 Design Kit (che hanno i color roles assegnati)
- lanciare il Material Theme Builder plugin
- creare un nuovo tema di colore
- scegliere un seed (colore che rappresenta il brand)
- attendere l'aggiornamento del tema
- selezionare i componenti dell'interfaccia
- cliccare su "swap" scegliendo un tema (chiaro, scuso, alto contrasto)