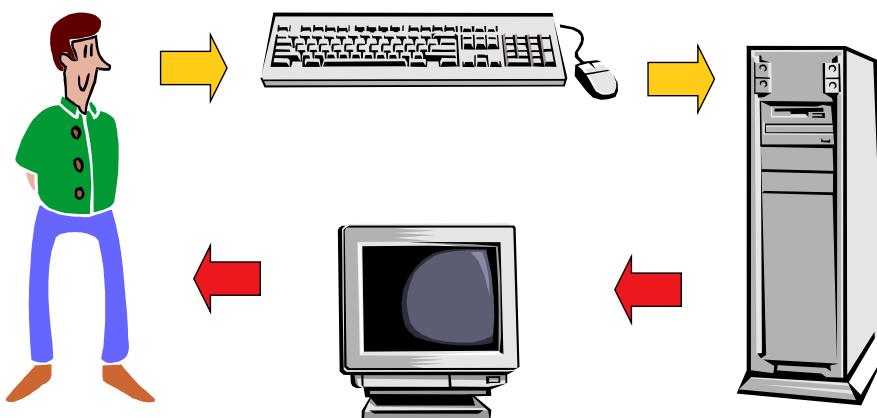
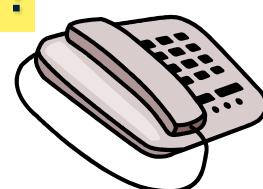


# Progettazione di Interfacce Utente

1

## Cos'è un'interfaccia?

- Nel gergo generale...  
dispositivo fisico/virtuale che  
⇒ *permette il dialogo tra due entità (partner)*
- Nel gergo elettronico...  
dispositivo fisico che  
⇒ *permette il transito di informazione tra due dispositivi o sistemi*



2

# Tecnologia vs. ergonomia

(Aspetti Tecnologici)

- **Sul lato fisico...**
- ⇒ ... tecnologia



- **Sul lato cognitivo** →
- ⇒ ...ergonomia cognitiva

Si concentra su come l'interfaccia e la tecnologia siano progettate per interagire con i sensi e il modo in cui le persone percepiscono ed elaborano informazioni.



3

Evoluzione dei

## I paradigmi di interazione

- Terminale scrivente: **scrivi e leggi** (Gli utenti interagiscono con il sistema scrivendo comandi e leggendo le risposte in formato testuale)
- Terminale video: **scegli e riempi** (Gli utenti selezionano opzioni predefinite e completano campi per fornire input)
- □ Personal Computer: **what if** (Gli utenti possono sperimentare e osservare cosa accade eseguendo azioni diverse)
- Sistemi multimediali: **parla ed ascolta** (L'interazione avviene tramite comandi vocali o risposte audio)
- □ Realtà virtuale: **entra ed agisci** (Gli utenti sono immersi in un ambiente virtuale in cui possono interagire fisicamente e digitalmente)

4

## Dalla parte dell'utente

### Le GUI esaltano le potenzialità del cervello umano:

⇒ riconoscere e associare (Gli utenti identificano rapidamente elementi visivi (icone, simboli) e li associano a funzioni o azioni specifiche)

⇒ generalizzare e dedurre (Le persone apprendono da modelli ripetuti (es. menu simili in diverse applicazioni) e deducono come utilizzare nuove funzionalità senza istruzioni esplicite)

### Come:

⇒ molte informazioni contemporaneamente (Le interfacce grafiche presentano più dati in un unico sguardo, permettendo all'utente di analizzarli rapidamente)

⇒ metafore (Le GUI ricorrono a metafore visive che richiamano oggetti o concetti familiari)

⇒ colore (Il colore è usato per catturare l'attenzione, categorizzare informazioni e comunicare emozioni o stati)

Le GUI efficaci si basano sull'uso equilibrato di queste tecniche per:

Ridurre il carico cognitivo.

Migliorare la navigazione e la comprensione.

Rendere il sistema più intuitivo per una vasta gamma di utenti, inclusi quelli meno esperti.

5

## Riconoscere o ricordare?

- Lista di codici registrazione ordinazioni al ristorante, facili da ricordare e raggruppati per significato.
1. Leggerli per 30 secondi
  2. Chiudere le dispense e cercare di riscriverli correttamente anche se in un qualunque ordine

FISS

SPAG

BRAC

ACQU

CART

RISO

POLL

VINO

SELF

BROD

PESC

LASA

BOLL

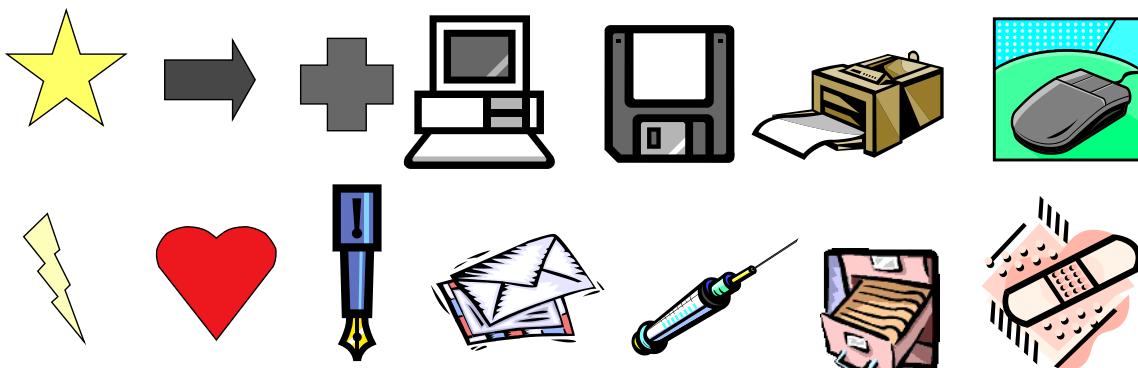
CONI

PUNTEGGIO: \_\_\_ /14

6

## Riconoscere o ricordare?

- Un insieme di icone d'aspetto familiare raggruppate per significato
  1. Osservarle per 30 secondi
  2. Chiudere le dispense e cercare di riscriverne i nomi (secondo la propria interpretazione) in un qualunque ordine



PUNTEGGIO: \_\_\_ /14

7

## Riconoscere o ricordare?

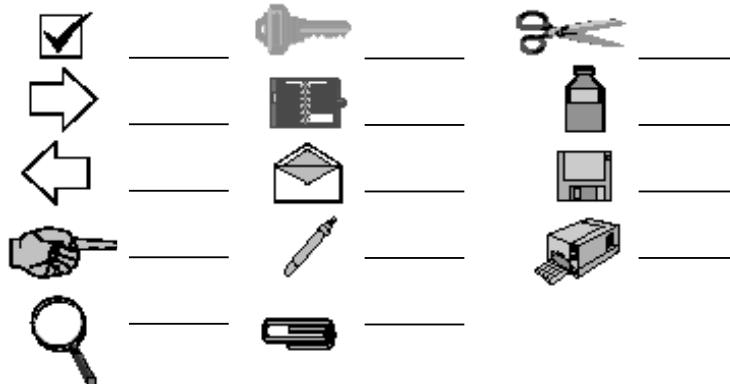
- Un insieme di icone che associano oggetti del mondo reale alle più diffuse funzioni computerizzate
  1. Osservarle per 30 secondi
  2. Voltare pagina e cercare di riscrivere il nome di ciascuna funzione accanto all'immagine dell'icona corrispondente



8

# Riconoscere o ricordare?

- Un insieme di icone che associano oggetti del mondo reale alle più diffuse funzioni computerizzate
  1. Osservarle per 30 secondi
  2. Voltare pagina e cercare di riscrivere il nome di ciascuna funzione accanto all'immagine dell'icona corrispondente



PUNTEGGIO: \_\_\_ /14

9

Riconoscere ed Associare  
é meglio che Ricordare

## Fattori chiave nel progetto dell'interfaccia

La progettazione deve partire dalla comprensione degli utenti finali.

- Competenze tecniche: Gli utenti possono essere esperti, intermedi o principianti.
- Bisogni specifici: Alcuni utenti possono avere requisiti particolari, come accessibilità per persone con disabilità.
- Obiettivi: Che cosa vogliono ottenere utilizzando l'interfaccia?

Chi userà l'interfaccia?

Per cosa l'interfaccia verrà usata?

Definire il contesto d'uso è essenziale.

- Uso professionale: Sistemi aziendali che richiedono efficienza e affidabilità.
- Uso personale: Applicazioni che privilegiano semplicità e intuittività.
- Compiti specifici: L'interfaccia deve supportare al meglio le azioni principali degli utenti.

### OBIETTIVI

### PERSONE



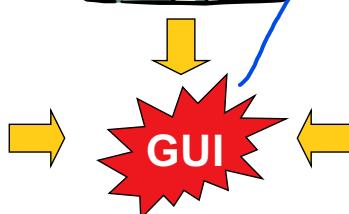
L'interfaccia GUI si trova al centro di tre aspetti fondamentali:

**Persone:**  
Gli utenti sono il fulcro del design. La progettazione deve essere orientata verso l'usabilità e le preferenze umane. Coinvolgono test con utenti reali per assicurarsi che l'interfaccia risponda alle loro esigenze.

**Tecnologia:**  
La tecnologia definisce i vincoli e le possibilità del sistema. È necessario considerare: Compatibilità hardware e software. Prestazioni tecniche.

**Obiettivi:**  
Gli obiettivi delineano ciò che l'interfaccia deve raggiungere, sia per l'utente che per l'organizzazione. L'interfaccia dovrebbe facilitare il raggiungimento di risultati concreti con il minimo sforzo.

### TECNOLOGIA



10

# Scelta della tecnologia per l'interfaccia

## □ ...in funzione degli obiettivi:

- ⇒ **rapidità o efficacia** Determinare se l'obiettivo principale è velocizzare le operazioni o assicurare precisione ed efficacia.
- ⇒ **che cosa è la qualità e quanto è importante** Stabilire il livello di qualità richiesto per soddisfare le aspettative degli utenti e del mercato.
- ⇒ **change management** Considerare come la tecnologia supporta il cambiamento all'interno dell'organizzazione, facilitando la transizione per gli utenti.
- ⇒ **utilizzo di strumenti di produttività individuale** La tecnologia deve migliorare la produttività personale, ad esempio con funzionalità semplici e strumenti automatizzati.
- ⇒ **strategie a lungo o a breve termine (elevato o modesto investimento)**

## □ ...in funzione degli utenti:

- ⇒ **numero d'utenti** Interfacce progettate per un grande pubblico (es. software aziendali) richiedono scalabilità e robustezza.
- ⇒ **esperienza nell'utilizzo della tecnologia** Gli utenti esperti possono gestire interfacce complesse, mentre i principianti necessitano di semplicità e istruzioni.
- ⇒ **età media** Interfacce per utenti più giovani possono includere elementi interattivi, mentre quelle per utenti anziani devono essere chiare e leggibili.
- ⇒ **motivazione o scetticismo** Gli utenti motivati sono più propensi a imparare, mentre gli scettici necessitano di un'interfaccia intuitiva per superare eventuali barriere.
- ⇒ **eterogeneità dei gruppi d'appartenenza** Quando gli utenti provengono da contesti diversi, l'interfaccia deve essere inclusiva e adattabile.
- ⇒ **turnover** Per sistemi con alto cambio di personale, le interfacce devono garantire una curva di apprendimento rapida.
- ⇒ **utilizzatori assidui o saltuari** Gli utenti assidui apprezzano funzioni avanzate, mentre i saltuari richiedono semplicità e immediatezza.
- ⇒ **versioni standard o ad hoc** Decidere se adottare una soluzione universale o personalizzare l'interfaccia per esigenze specifiche.

11

Queste interfacce sono comuni nei sistemi operativi e negli ambienti di lavoro specifici dove è necessario gestire una grande mole di dati o operazioni complesse.

Vari tipi di Interfacce Software

## Interfacce code-based 1

### □ Interazione attraverso comandi (codici)

- ⇒ Ottimale per moli di lavoro elevate che richiedono attenzione in punti lontani dal video (es. Check-In in aeroporto)
- ⇒ Occorre mantenere basso il numero di codici utilizzabili
- ⇒ Nessuna riusabilità delle conoscenze acquisite

Sono efficaci in situazioni in cui è necessario processare rapidamente grandi quantità di dati o automatizzare operazioni.

- Gli operatori possono lavorare senza bisogno di guardare costantemente uno schermo, ma richiedono familiarità con i comandi.
- Mole di lavoro da svolgere: Capacità dell'interfaccia di supportare operazioni complesse o ripetitive in modo efficiente.
  - Qualità: Affidabilità, chiarezza e coerenza del design nell'aiutare l'utente a completare i compiti.
  - Facilità di apprendimento: Tempo e sforzo necessari per capire e utilizzare l'interfaccia.
  - Riuso delle conoscenze: Capacità dell'utente di trasferire le competenze acquisite da una GUI a un'altra con design simile.
  - Soddisfazione: Livello di piacere e comfort nell'utilizzo dell'interfaccia.

Mole di lavoro da svolgere	Qualità	Facilità di apprendimento	Riuso delle conoscenze	Soddisfazione
😊	😐	🙁	🙁	🙁

Comandi Specifici del Software

```
> copy utenti.txt D:  
> print utenti.txt  
> delete utenti.txt
```

Gli acceleratori nelle GUI (Graphical User Interface) sono scorciatoie da tastiera progettate per velocizzare l'esecuzione di operazioni frequenti o comuni, evitando l'uso del mouse o l'esplorazione di menu. Questi permettono agli utenti di interagire con l'interfaccia in modo più rapido ed efficiente.

12

Le interfacce 3270 sono strumenti potenti per contesti aziendali altamente strutturati, dove precisione e standardizzazione sono fondamentali.

## Interfacce 3270 2

### □ Interfaccia a caratteri

Il layout è statico e strutturato, basato su campi fissi per l'inserimento di dati.

⇒ Ottimale per data-entry ed editing di dati altamente strutturati

⇒ Workflow fortemente predefinito (bassa flessibilità)

⇒ Navigazione e tasti funzionali complicano apprendimento e riusabilità delle conoscenze acquisite

Progettata per compiti di inserimento dati veloci e precisi, spesso utilizzata in contesti di lavoro con grandi volumi di informazioni

Le operazioni seguono sequenze rigide di passaggi. Questo garantisce standardizzazione ma riduce la capacità di personalizzazione o adattamento a esigenze non previste.

Mole di lavoro da svolgere	Qualità	Facilità di apprendimento	Riutilizzo conoscenza	Soddisfazione
😊	😊	😐	🙁	🙁

NOME :	_____
COGNOME :	_____
SESSO :	_____
RESIDENZA :	_____

13

Le pseudo-GUI rappresentano un buon compromesso tra interfacce grafiche complesse e interfacce a caratteri, bilanciando flessibilità, usabilità e semplicità. Sono particolarmente adatte per applicazioni aziendali in cui i dati strutturati e standardizzati hanno un ruolo predominante.

## Pseudo-GUI 3

### □ Interfaccia grafica che richiama la strutturazione di un'interfaccia a caratteri

La disposizione degli elementi è ordinata e segue un layout strutturato, simile alle interfacce a caratteri. Sono aggiunti elementi grafici base

⇒ Ottimale per applicazioni che debbano gestire dati fortemente strutturati garantendo una buona flessibilità

⇒ Se standard consente riusabilità delle conoscenze acquisite

Progettate per applicazioni che gestiscono grandi volumi di dati strutturati. Rispetto alle interfacce puramente a caratteri, le pseudo-GUI offrono maggiore adattabilità, permettendo di aggiungere nuovi campi o funzioni senza stravolgere l'interfaccia.

Se il design segue standard comuni, gli utenti possono trasferire facilmente le competenze acquisite a sistemi simili

Mole di lavoro da svolgere	Qualità	Facilità di apprendimento	Riutilizzo conoscenza	Soddisfazione
😊	😊	😐	😊	😐

(più intuitiva delle interfacce a caratteri)

14

Le Standard GUI rappresentano uno standard ampiamente adottato per software moderni grazie alla loro intuitività, flessibilità e capacità di adattarsi a diversi contesti di utilizzo. La loro progettazione enfatizza l'esperienza utente, facilitando il completamento delle attività con strumenti visivi immediati e interattivi.

(nella maggior parte dei Sistemi di OGGI)

## Standard GUI 4

### □ Progettata e sviluppata per un ambiente grafico

⇒ Esaltate le potenzialità di manipolazione diretta (cut & paste, <sup>utilizzo Mouse</sup> drag & drop, etc.)

Le GUI enfatizzano la possibilità di interagire direttamente con gli elementi sullo schermo.

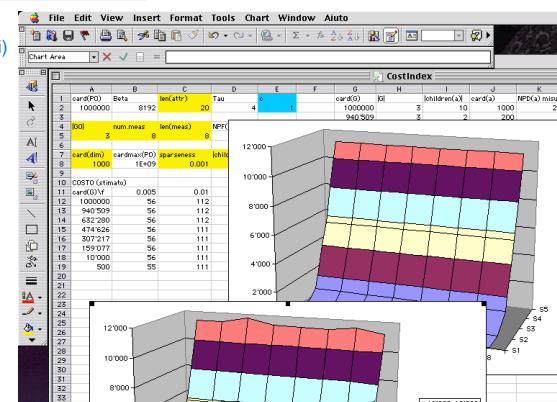
⇒ Ottimale per user-driven applications (flessibilità) →

Mole di lavoro da svolgere	Qualità	Facilità di apprendimento	Riutilizzo conoscenza	Soddisfazione
😊	😊	😊	😊	😊

(intuitiva per la maggior parte degli utenti)

Le GUI standard utilizzano icone, finestre, menu e pulsanti per consentire un'interazione intuitiva tra l'utente e il sistema.

Sono particolarmente adatte per applicazioni in cui l'utente ha il controllo completo sulle operazioni da svolgere. La flessibilità offerta da una GUI permette di affrontare una vasta gamma di compiti, da semplici operazioni a complessi processi analitici.



15

Le Special GUI sono progettate per massimizzare l'accessibilità e l'esperienza visiva, rendendole ideali per utenti non esperti o per contesti dove la semplicità è fondamentale. Tuttavia, il loro utilizzo è spesso limitato a scenari specifici.

## Special GUI 5

### □ Enfasi massima alla presentazione grafica

Gli elementi visivi sono curati per essere accattivanti, intuitivi e adatti a guidare l'utente senza bisogno di istruzioni complesse.

⇒ Obiettivo prioritario è l'autoespliсation (EIS, videogames)

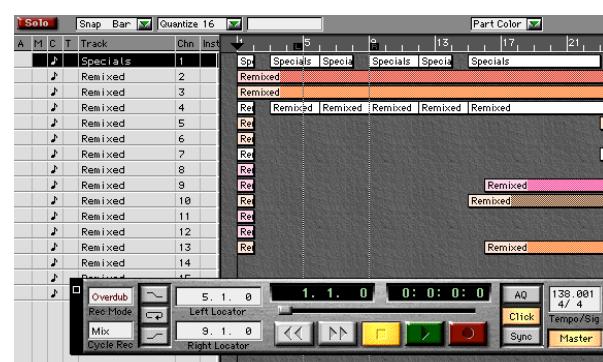
⇒ Il cliente "si serve" da solo... Ideate per consentire all'utente di svolgere operazioni autonomalemente, senza intervento esterno.

⇒ L'utente target potrebbe non avere esperienza sull'utilizzo dei computer

Mole di lavoro da svolgere	Qualità	Facilità di apprendimento	Riutilizzo conoscenza	Soddisfazione
😊	😊	😊	😢	😊

(niente di standard perché molto specifica)

Le interfacce sono progettate per essere immediatamente comprensibili, anche per utenti non esperti.



16

# Strutturazione

- Una struttura “bassa e larga” fornisce all’utente una visione migliore delle possibilità offerte e facilita la navigazione

Alta e Stretta:

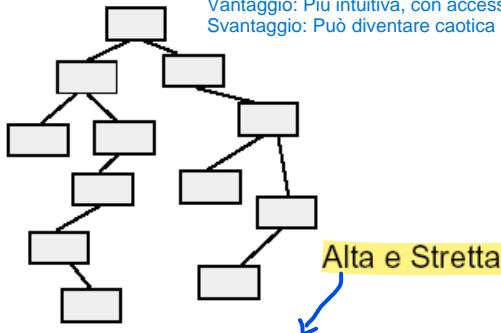
Vantaggio: Più organizzata e sistematica per grandi volumi di informazioni.

Svantaggio: Può essere meno user-friendly per chi cerca una navigazione veloce.

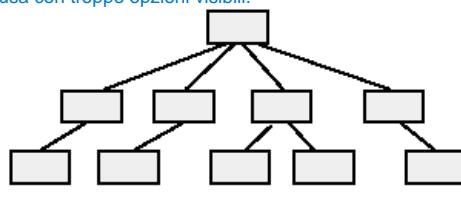
Bassa e Larga:

Vantaggio: Più intuitiva, con accesso immediato a più opzioni.

Svantaggio: Può diventare caotica o confusa con troppe opzioni visibili.



Alta e Stretta



Bassa e Larga

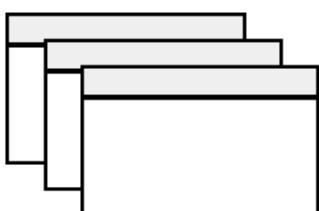
Ha molti livelli gerarchici, ma ogni livello contiene pochi elementi.  
È utile per sistemi complessi che necessitano di suddivisioni dettagliate, ma può risultare più lenta per l’utente, in quanto richiede più passaggi per raggiungere il contenuto desiderato.  
Pro: Ogni step, ho meno scelte/possibilità da fare  
Contro: Poco Ergonomico (Molti Click per arrivare alle foglie)  
Problema: Gli utenti potrebbero perdersi o confondersi nel navigare attraverso troppi livelli.

Ha meno livelli gerarchici, ma ogni livello contiene più elementi.  
Fornisce una visione più chiara e immediata di tutte le opzioni disponibili.  
Pro: Ergonomia (Pochi Click per arrivare alle Foglie)  
Contro: Molte Scelte/Possibilità da fare  
Vantaggio: Migliora la navigazione perché richiede meno passaggi per raggiungere una destinazione.

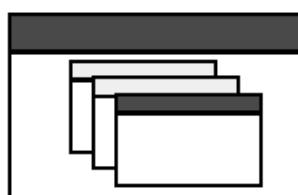
# Strutture di riferimento

Ci sono 3 Tipi di strutture dei sistemi a finestra

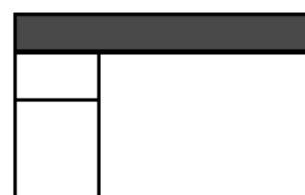
## Multi-Window



## Multi-Document (MDI)



## Multi-Paned



## Vantaggi:

- Flessibilità: Gli utenti possono gestire più attività contemporaneamente.
- Indipendenza: Ogni main window è autonoma.

## Svantaggi:

- Sovraccarico cognitivo: Può risultare difficile tenere traccia di tutte le finestre aperte.
- Disordine visivo: Con molte finestre aperte, lo spazio visivo può essere occupato in modo inefficiente.

# 1

## Modello multi-window

⇒ molte *main window* (ciascuna con un menu)

Ogni finestra principale ha il proprio menu e rappresenta un elemento importante del sistema.

⇒ rapporto 1:1 tra *main window* e *business object*

In relazione 1:1 con un business object, ovvero un'entità funzionale dell'applicazione (ad esempio un documento o una tabella di dati).

⇒ molte *child windows* (senza menu) possibili per ciascuna *main window*

Ogni main window può avere diverse child windows (finestre figlie) associate, che sono più specifiche e subordinate. Non hanno un proprio menu. Le child windows servono per svolgere funzioni specifiche

Pro

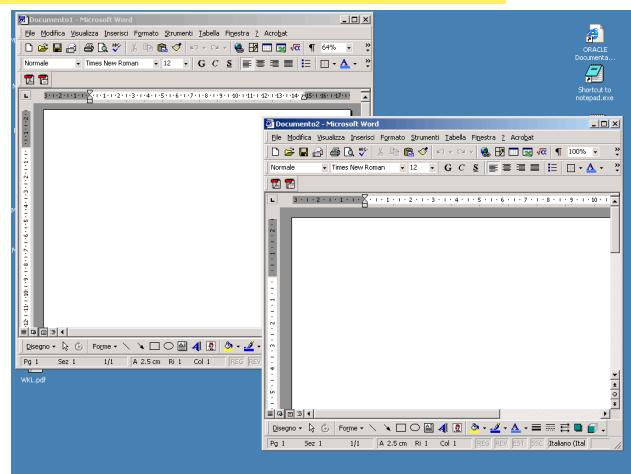
Contro

Più *main window* attivabili contemporaneamente: estrema flessibilità

Navigazione complessa

Uno svantaggio è la complessità di navigazione. L'utente deve passare frequentemente da una finestra all'altra, il che può generare confusione, soprattutto se le finestre si sovrappongono o occupano tutto lo schermo.

In questo modello è possibile aprire e gestire più *main window* in parallelo, garantendo grande flessibilità. Ad esempio, puoi lavorare su più documenti in Word contemporaneamente.



19

# 2

## Modello multi-document

⇒ una sola *top window* con menu

Esiste una sola *top window* (finestra principale) che contiene un menu unico per controllare il sistema. È la finestra principale che guida l'interazione con tutte le altre sottofinestre, denominate *document windows*.

⇒ la *top window* guida una serie di *document window*

⇒ la *top window* deve sempre rimanere aperta

Se la Top Window viene chiusa, anche tutte le document windows si chiudono.

□ Flessibilità inferiore a quella del multi-window model

→ perché tutte le operazioni sono vincolate alla finestra principale.

□ Vi sarà sempre un solo menu attivabile

□ Ottimale anche per utenti inesperti

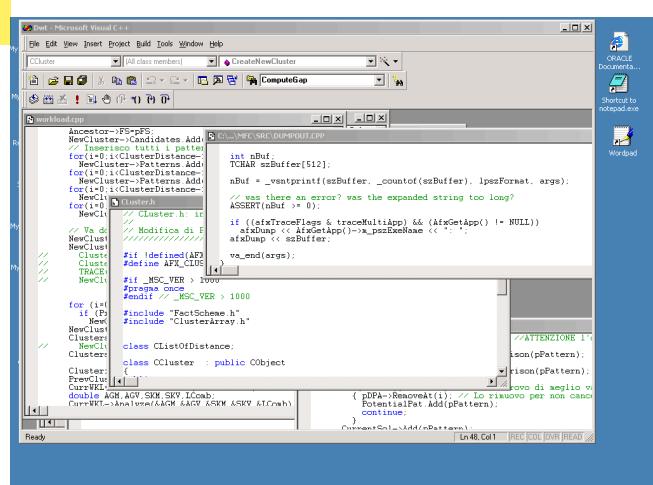
## Vantaggi:

- Semplicità per l'utente: Ottimale per utenti inesperti poiché riduce la confusione data da più finestre indipendenti. La navigazione è più semplice perché tutto è contenuto in un unico ambiente.

- Uniformità del layout: L'utente lavora in un contesto visivo coerente, senza la necessità di gestire più finestre sparse sullo schermo.

## Svantaggi:

- Flessibilità limitata: Non è possibile gestire document windows separatamente dalla top window. L'interazione è vincolata all'ambiente della finestra principale.



20

### 3 Modello multi-paned

Ridimensionabili,  
ma in termini di  
posizionamento  
sono fissi

⇒ una "window" alla volta con o senza menu

Il modello prevede una singola finestra che contiene tutte le funzionalità necessarie. Può avere o meno un menu, a seconda del design specifico dell'applicazione.

⇒ eventuale suddivisioni in aree (pane) monofunzionali e monoposizionali

La finestra è suddivisa in diverse aree chiamate panes, ognuna dedicata a una funzione specifica. Questi pannelli sono monofunzionali (cioè ogni pane serve a uno scopo preciso) e monoposizionali (non possono essere spostati liberamente).

- Assenza di flessibilità
- Per special GUI in applicazioni self-service

Questo modello è rigido: i pannelli sono fissati nella loro posizione, e l'utente non può modificare liberamente il layout. La struttura è pensata per guidare l'utente e ridurre la confusione, ma sacrifica la personalizzazione.



21

I Project Standard sono regole progettuali pensate per migliorare l'esperienza dell'utente, promuovendo coerenza, facilità d'uso e prevedibilità. Creano un'interfaccia uniforme che risponde sia alle aspettative degli utenti che ai requisiti aziendali.

### Project standard

- Definizione degli standard per:

⇒ terminologia Usare termini coerenti e significativi per l'utente.

⇒ metafore, icone Utilizzo di metafore visive comuni che richiamano azioni o oggetti familiari. Le icone devono essere facilmente riconoscibili e coerenti in tutto il progetto.

⇒ caratteristiche delle finestre (menu, buttoni, dimensioni, posizione, ecc.) Uniformità nel design di menu, buttoni, dimensioni, posizione delle finestre.

- Obiettivo prioritario: agevolare l'utilizzo da parte dell'utente (Usabilità)

⇒ consistenza esterna → Allineare il design e la terminologia con i tool già utilizzati dall'utente

⇒ i tool già utilizzati in azienda (standard de facto)

→ Considerare gli strumenti già utilizzati in azienda come standard de facto significa progettare nuovi sistemi e interfacce che rispettino le abitudini consolidate. Ciò migliora l'esperienza utente, minimizza i tempi di adattamento e rende più efficace l'adozione dei nuovi sistemi.

⇒ consistenza interna subordinata all'usabilità

Garantire che tutti gli elementi dell'applicazione seguano le stesse regole e comportamenti.

Vantaggi dell'adozione di standard di progetto

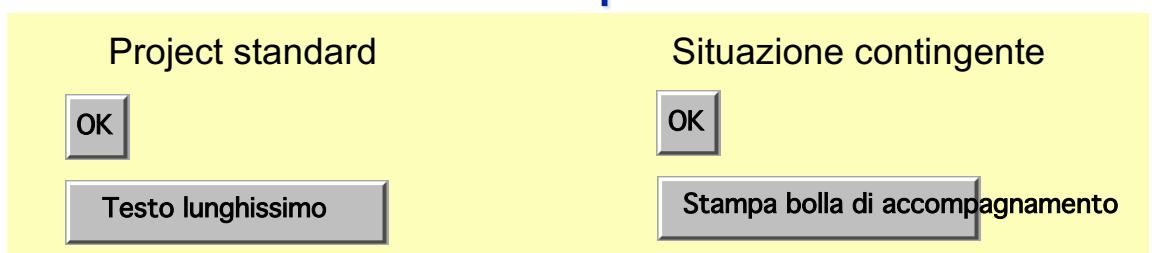
Coerenza visiva e funzionale: Gli utenti si sentono a loro agio poiché non devono imparare meccanismi diversi in diverse parti del sistema.

Riduzione degli errori: Standard prevedibili riducono il rischio di confusione o di utilizzo errato dell'interfaccia.

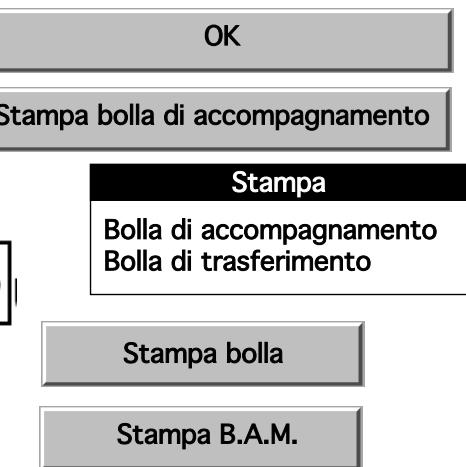
Efficienza nello sviluppo: Designer e sviluppatori possono seguire linee guida definite, evitando ambiguità durante la progettazione.

22

# Esempio



- Allargare tutti i bottoni della window
- Allargare solo il bottone “incriminato”
- Ridisegnare la window ed inserire la scelta nel menù
- Un simbolo al posto del testo 
- Testo più corto compreso ed approvato dall’utente
- Abbreviazione compresa ed approvata dall’utente



23

# Esempio

- Priorità consigliate
  1. Testo più corto compreso ed approvato dall’utente
  2. Abbreviazione compresa ed approvata dall’utente
  3. Allargare tutti i bottoni della window/gruppo
  4. Allargare solo il bottone “incriminato”
  5. Un simbolo al posto del testo
  6. Ridisegnare la window ed inserire la scelta nel menù

24

# Comunicazione visiva nelle GUI

- 1 Si riferisce alla capacità di un oggetto o di un elemento della GUI di suggerire il modo in cui deve essere utilizzato. Per esempio, un pulsante con un design tridimensionale "invita" l'utente a cliccarlo. Nella progettazione, è importante che gli utenti capiscano immediatamente come interagire con un elemento senza necessità di istruzioni aggiuntive.
- 2 **Affordance:** enfatizza gli aspetti di un oggetto che invitano a manipolarlo in un certo modo
- 3 **Metafora:** una parola, una frase o una figura che dipinge un oggetto o un concetto attraverso una somiglianza o un'analogia con un altro oggetto o concetto del mondo reale → Le metafore sono essenziali per rendere intuitiva l'interazione. Collegano il mondo digitale al mondo reale. Usare metafore ben conosciute semplifica la curva di apprendimento dell'utente.
- 4 **Layout:** è determinato dalla posizione del testo, dei disegni e dei controlli all'interno di un'area considerata → Indica come gli elementi della GUI (testi, pulsanti, icone, immagini) sono organizzati nello spazio. Un buon layout garantisce chiarezza e leggibilità.
- 5 **Colori:** utili per focalizzare l'attenzione o per creare associazioni
- 6 **Icone:** disegni piccoli, semplici e metaforici → Devono essere semplici, riconoscibili e spesso basate su metafore. Sono utili per comunicare rapidamente informazioni senza sovraccaricare il design con testo.
- 7 **Font:** leggibilità in relazione al tipo e alle caratteristiche del carattere  
Il font influenza direttamente la leggibilità. Le scelte devono essere basate su:  
- La grandezza (adatta al contesto e leggibile su tutti i dispositivi).  
- Lo stile (evitare font troppo elaborati che riducono la chiarezza).  
- Il contrasto (testo ben visibile rispetto allo sfondo).
- Questi principi contribuiscono a una comunicazione visiva efficace, rendendo l'interfaccia intuitiva, accessibile e piacevole da usare. Un design che segue queste regole consente agli utenti di interagire con il sistema in modo naturale e senza frustrazioni.

25

## 1 Affordance

(Diversi Aspetti per rappresentare Affordance)

**Tridimensionalità** → I controlli con effetto tridimensionale offrono un chiaro segnale visivo che possono essere "premuti" o "cliccati".

**Ombreggiatura** → L'ombreggiatura crea profondità, suggerendo che un elemento è cliccabile o interattivo.

**Puntamento** → Si riferisce a elementi che guidano l'attenzione dell'utente o suggeriscono una direzione d'azione.

Pulsante ON/OFF

Checkbox

Scorri tra le pagine  
Scorri a destra o sinistra con un dito

Scorri tra le app a tutto schermo  
Scorri a destra o sinistra con due dita

Mission Control  
Tocca due volte con due dita

Radio Button

Select Number

1  
 2  
 3  
 4

Dialog Box (Finestra di conferma)

Confirm Save Model and Reload

Select OK to save the data model and continue reload.  
Cancel will continue reload without saving the data model.

OK Cancel

26

## 2 Metafore

Le metafore sono essenziali per rendere l'interazione intuitiva collegando concetti digitali a oggetti o esperienze del mondo reale.

### □ La prima tra le scelte progettuali...

Perché le metafore sono importanti?

- Facilitano la comprensione: Offrono un punto di riferimento familiare all'utente, specialmente per chi non ha esperienza tecnica.
- Riduzione della curva di apprendimento: Aiutano gli utenti a capire immediatamente come utilizzare il software.
- Comunicazione immediata: Un'immagine o un'icona basata su metafore può comunicare più velocemente di un testo.

simbolo di divieto  
+  
evocazione del "fumo"



### Metafore comuni → Associazioni

Documento	File
Cartella	Directory
Schedario	Storage System
Scheda	Record
Lettera	E-mail
Taglia e cuci	Scrivi e leggi da un buffer
Cestino	Cancel
Bottone	Command
Gomma	Undo

27

## 3 Layout →

Il layout definisce come gli elementi sono posizionati all'interno dell'interfaccia, influenzando direttamente l'efficacia comunicativa e l'usabilità.

### □ La posizione degli elementi è un importante strumento di comunicazione

⇒ Le distanze devono essere scelte in relazione al grado di associazione tra gli elementi

Elementi correlati devono essere posizionati vicini per evidenziarne la relazione. Elementi indipendenti richiedono più spazio per sottolineare la loro separazione.

### □ Fra gli standard di progetto...

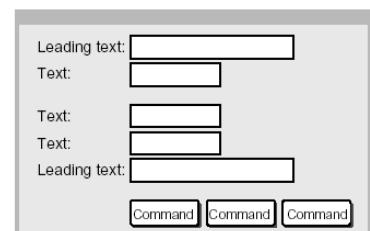
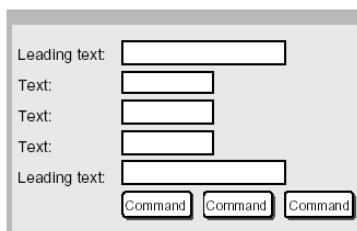
⇒ distanza tra campi correlati → Campi che appartengono allo stesso gruppo (es. Nome, Cognome, Data di nascita) devono essere vicini e ben allineati per facilitare la comprensione.

⇒ distanza tra i gruppi → Gruppi distinti di elementi (es. informazioni personali, dettagli di pagamento) devono essere separati da spazi visivi più ampi per evitare confusione.

⇒ distanza (superiore, laterale, inferiore) tra riquadro ed elementi contenuti → Quando un riquadro contiene elementi (es. una sezione evidenziata), ci deve essere un margine coerente (superiore, inferiore, laterale) per evitare che gli elementi appaiano troppo "compresi".

⇒ distanza (superiore, laterale, inferiore) tra margine dell'area principale ed elementi contenuti

Il contenuto non deve essere troppo vicino ai bordi della finestra o del contenitore principale. Margini chiari rendono l'interfaccia più "leggera" e leggibile.

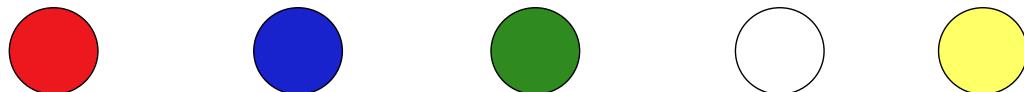


28

Il colore è un potente strumento comunicativo che deve essere usato strategicamente per migliorare l'interazione utente. La tabella dimostra che ogni colore può avere significati diversi in base al contesto culturale, quindi è fondamentale progettarne l'uso con attenzione e consapevolezza del pubblico di riferimento.

## 4 Colori

Cultura	Rosso	Blu	Verde	Bianco	Giallo
USA	Pericolo	Mascolinità	Sicurezza	Purezza	Codardia
Francia	Aristocrazia	Libertà, pace	Criminalità	Neutralità	Temporaneità
Egitto	Morte	Virtù, fede	Fertilità, forza	Gioia	Prosperità
India	Vita, creatività	Gioia, potenza	Prosperità	Purezza	Successo
Giappone	Pericolo	Malvagità	Futuro, energia	Morte	Nobiltà
Cina	Felicità	Paradiso	(Ming) Paradiso	Purezza	Nascita



il colore deve essere usato intenzionalmente per migliorare l'esperienza utente.

dimenticatevi l'estetica, il colore è comunicazione!

(Jan B. White)

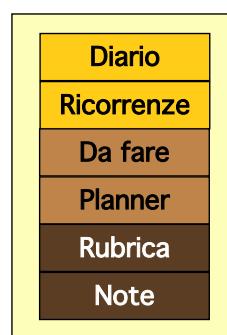
29

## 4 Colori

In che senso il colore è comunicazione?



decorazione



codifica

Decorazione:

I colori vengono usati puramente per abbellire o decorare la GUI, senza aggiungere un significato funzionale specifico. Ogni elemento ha un colore distintivo, ma non vi è una relazione logica tra il colore e il suo significato o scopo.

Codifica:

I colori sono usati intenzionalmente per rappresentare un'informazione o per trasmettere un significato specifico. Qui, i colori sono meno vari, ma ogni tonalità è scelta per suggerire una relazione o un tema comune.

30

## 4 Colori

- Non abusare dei colori in un “ambiente” monocromatico: il risalto è eccessivo → In un’interfaccia monocromatica (cioè dominata da un colore principale), l’uso eccessivo di colori può risultare troppo evidente o “fuori contesto”. Usare colori complementari in modo selettivo, per evidenziare solo le aree più importanti.
- Se il colore è usato come codice: solo 3-5 colori, ricordarsi la semantica (cioè: quando i colori sono utilizzati per trasmettere informazioni)
- Colori vivaci per aree piccole e neutri per aree grandi → I colori vivaci attirano molta attenzione, quindi devono essere usati per piccoli dettagli (es. icone, pulsanti) e non per riempire interi spazi, che diventerebbero visivamente pesanti.
- Ricercare un contrasto efficace tra testo e sfondo → Il contrasto è fondamentale per garantire la leggibilità.
- Sfondo chiaro (bianco, grigio, giallo) è ottimale per testi scuri → Gli sfondi chiari aiutano a mettere in risalto il testo e sono meno stanchi per gli occhi rispetto agli sfondi scuri, specialmente su lunghi blocchi di testo.
- I colori troppo brillanti causano alterazione visiva sui tempi lunghi: sono pertanto sconsigliabili per applicazioni gestionali, mentre risultano ottimali nelle applicazioni self-service

→ I colori brillanti (es. rosso vivo, giallo fluo) affaticano gli occhi e possono risultare disturbanti su applicazioni utilizzate per lunghi periodi.

Sconsigliato per applicazioni gestionali: In software usati per ore, preferire tonalità neutre e sobrie.

Adatto per applicazioni self-service: Nei chioschi o applicazioni rapide, i colori brillanti attirano l’attenzione e possono essere efficaci.

31

## 5 Icone

Le icone sono componenti visivi che rappresentano azioni, oggetti o concetti in modo sintetico e intuitivo.

- Struttura** Le icone sono composte da:
  - ⇒ immagine
  - ⇒ sfondo
  - ⇒ testo (facoltativo)
- Caratteristiche** Per essere efficaci, le icone devono possedere queste qualità:
  - ⇒ Facilmente distinguibili → Ogni icona deve essere chiaramente diversa dalle altre per evitare confusione.
  - ⇒ Elevato valore informativo → Devono comunicare chiaramente il loro significato anche senza testo.
  - ⇒ Presentazione esplicita della metafora → Le icone devono utilizzare metafore riconoscibili, ovvero rappresentazioni grafiche che richiamano concetti del mondo reale.
  - ⇒ Incrementano la velocità e la correttezza della selezione → Dovrebbero essere intuitive al punto da non necessitare di testo aggiuntivo per essere comprese. Tuttavia, il testo può essere utile come supporto.
  - ⇒ Autoesplicative anche se prive di testo → Icone ben progettate aiutano l’utente a trovare e selezionare rapidamente ciò che cerca, riducendo il carico cognitivo.
- Linee guida** per la progettazione delle icone
  - ⇒ Disegni semplici e schematici → Le icone devono essere essenziali e prive di dettagli inutili. La semplicità migliora la leggibilità.
  - ⇒ Colori differenti in icone differenti → Quando possibile, usare colori distintivi e la riconoscibilità, soprattutto su schermi piccoli, per differenziare meglio le icone.
  - ⇒ Il testo è il titolo della finestra collegata → Se un’icona è associata a una funzione o finestra, il testo dovrebbe rappresentare chiaramente tale funzione
  - ⇒ Evitare i puzzle! → Non rendere le icone troppo criptiche o complesse, altrimenti l’utente non capirà facilmente cosa rappresentano.



Piscina



Servizio Elicotteri



Traghetti

32

# 5 Icônes

## 1. Desktop icon

Le desktop icon sono le icone che rappresentano applicazioni o file principali sul desktop o in altre aree simili

⇒ Obiettivo: partenza, riapertura

Partenza: Consentire all'utente di avviare un'applicazione o aprire un file.  
Riapertura: Facilitare l'accesso rapido a programmi o file già usati.

⇒ Per applicazioni collegate per l'utente, icone simili graficamente →

⇒ Se minimize:

Quando un'applicazione ha più finestre o documenti aperti, ma sono minimizzati, il design dell'icona deve seguire alcune regole:

Le finestre aperte di un'applicazione devono avere icone simili per indicare la loro appartenenza alla stessa applicazione.

icone simili per finestre diverse della stessa applicazione

• il testo è fondamentale per icone simili rappresentanti finestre diverse → Quando le icone rappresentano finestre simili, è indispensabile includere testo descrittivo (es. il titolo della finestra o del documento). Questo aiuta l'utente a distinguere facilmente le finestre.

• testo = window title → Il titolo associato all'icona dovrebbe corrispondere al nome della finestra o del file aperto per evitare confusione.



Quando più applicazioni sono collegate (es. Microsoft Office con Word, Excel, PowerPoint), le icone devono essere simili graficamente per suggerire che fanno parte dello stesso pacchetto o sistema.

# 5 Icônes

## 2. Menu icon - Palette Icon

Accompagnano i menu e le toolbar delle applicazioni.

Forniscono un accesso rapido e visivo a funzioni o comandi.

⇒ Sempre visibili accanto ai menu →

Le icone sono posizionate accanto ai menu principali (es. toolbar, ribbon) e restano costantemente visibili per agevolare l'utente.

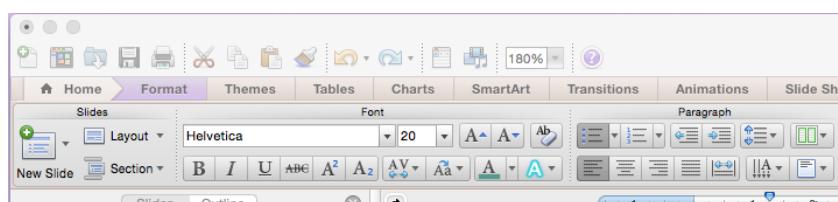
• overview di funzioni sempre attivabili → Mostrano chiaramente le opzioni disponibili senza bisogno di navigare in sottomenu.

• un modo veloce di selezionare → Permettono di attivare un comando con un singolo clic, velocizzando l'interazione rispetto all'utilizzo di menu testuali.

• per comandi esprimibili più facilmente con disegni che con parole →

L'aspetto visivo delle icone incoraggia gli utenti a esplorare nuove funzionalità, soprattutto perché le icone sono spesso accompagnate da tooltip che spiegano il loro utilizzo.

Alcuni comandi sono più intuitivi se rappresentati graficamente piuttosto che con parole.



Utilizzato in combinazione con i pulsanti per migliorare l'esperienza utente e la chiarezza funzionale.

Tre Tipi standard di utilizzo di una Icona

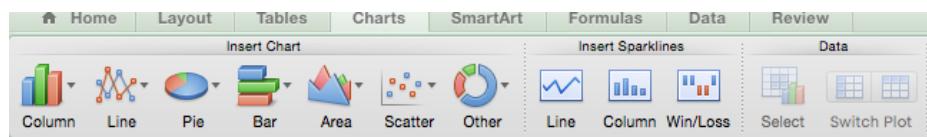
## 5 Icône

3. **Button icon** → Accompagnano il testo all'interno di un pulsante o accanto ad esso.  
Rafforzano graficamente la funzione del pulsante, rendendo più evidente e intuitiva la sua finalità.

⇒ In aggiunta al testo di un bottone → L'Icona non sostituisce il testo ma lo supporta, rendendo più chiara la funzione del pulsante.

### Rafforza graficamente la funzione del bottone

↳ Le icone aiutano a rappresentare visivamente la funzione del pulsante, soprattutto in contesti in cui il testo potrebbe non essere sufficiente o comprensibile a colpo d'occhio.



35

I font giocano un ruolo fondamentale per garantire leggibilità, chiarezza e professionalità.

## 6 Font

Utilizzare Massimo 2/3 Tipi di Font della stessa Famiglia.

- **Linee guida:** → Grazie (decorazioni)

⇒ **Sans Serif per singole righe**

↳ I font Sans Serif (es. Helvetica) sono caratterizzati dall'assenza di decorazioni (serif) sulle lettere.  
Adatti a: Testi brevi o singole righe (es. titoli, etichette, pulsanti). Situazioni dove la leggibilità è fondamentale su schermi.

Questo è il font Helvetica

⇒ **Serif per testi articolati su molte righe**

↳ I font Serif (es. Times New Roman) includono decorazioni alle estremità delle lettere, che guidano lo sguardo durante la lettura.  
Adatti a: Lunghi blocchi di testo (es. paragrafi, articoli, documenti). Stampa tradizionale o contesti in cui è richiesta eleganza.

Questo è il font Times, più adatto per coprire più righe

⇒ **Attenzione al maiuscolo**

↳ Usare lettere maiuscole con moderazione (I testi interamente in maiuscolo possono essere difficili da leggere e trasmettere un senso di "urgenza" o "grido".).

ATTENZIONE

non abusare del maiuscolo

⇒ **Spaziatura proporzionale**

↳ La spaziatura proporzionale migliora la leggibilità, adattando lo spazio tra le lettere in base alla loro forma.

Questo è il font Courier

Spaziatura Non Proporzionale:  
Ogni Carattere occupa stesso pixel

36

Questo significa che un sistema o un'interfaccia deve:  
Permettere agli utenti di raggiungere i loro obiettivi (efficacia).  
Essere facile da usare con un consumo minimo di risorse (efficienza).  
Essere percepito come piacevole e accettabile dagli utenti (soddisfazione).

## Usabilità

L'usabilità è un equilibrio tra efficacia, efficienza e soddisfazione.

L'usabilità è definita come:

**L'efficacia, efficienza e soddisfazione con cui determinati utenti eseguono determinati compiti in particolari ambienti**

⇒ **Efficacia**: in che misura i compiti previsti dal funzionamento vengono eseguiti → Indica la capacità dell'utente di completare i compiti previsti utilizzando il sistema (Se riesco ad effettuare delle Task in maniera corretta).

⇒ **Efficienza**: risorse da impegnare per eseguire i compiti previsti

⇒ **Soddisfazione**: misura dell'accettabilità del funzionamento da parte dell'utente → Valuta la percezione dell'utente sull'esperienza di utilizzo del sistema, ovvero se l'esperienza è stata piacevole o frustrante.

**...ma anche comprensibilità, apprendibilità, operabilità**

(Prossima Slide ho i 6 criteri di Usabilità)

Comprendibilità: Il sistema deve essere intuitivo e facile da capire.

Esempio: Le etichette dei pulsanti e i messaggi di errore devono essere chiari e comprensibili.

Apprendibilità: L'utente deve poter imparare facilmente ad utilizzare il sistema, anche senza formazione.

Esempio: Un'app di onboarding con tutorial interattivi può migliorare l'apprendibilità.

Operabilità: Il sistema deve funzionare in modo fluido e coerente, senza creare difficoltà operative.

Esempio: Un'applicazione che risponde rapidamente ai comandi senza errori tecnici.

37

## Usabilità



i 6 criteri di usabilità

A seconda del contesto,  
un criterio è più  
importante di un altro

38

# 1 Apprendibilità

## Obiettivo

- ⇒ 80% dei nuovi utenti **deve essere** in grado di svolgere compiutamente una singola attività dell'applicazione in 30 minuti

Questo significa che il sistema deve essere progettato in modo:  
Intuitivo: con interfacce facili da navigare.  
Guida-friendly: con tutorial o istruzioni chiare.  
Non troppo complesso: evitando funzioni inutili o processi lunghi.

In contesti dove il tempo e le risorse per la formazione sono limitati, un sistema ben progettato deve essere quasi autoesplicativo.

L'apprendibilità diventa una priorità in contesti specifici

## Quando

- ⇒ Turn-over alto
- ⇒ Utenti saltuari
- ⇒ Riduzione del training
- ⇒ Sistemi solitamente sottoutilizzati per mancanza di training
- ⇒ Breve ciclo di vita dei prodotti

Quando il sistema è utilizzato in ambienti con un frequente cambio di personale.

Se gli utenti non interagiscono frequentemente con il sistema, è importante che possano ricordare o riapprendere facilmente le funzioni principali.

Quando un prodotto ha un ciclo di vita limitato, non c'è tempo per training intensivi. Gli utenti devono poter apprendere rapidamente le funzioni (ES: Sistemi temporanei).

39

# 2 Velocità

(Efficienza)

## Obiettivo

- ⇒ 10 inserimenti ogni 2 minuti

Questo obiettivo stabilisce un benchmark di efficienza per misurare le prestazioni del sistema in contesti operativi ripetitivi.

In contesti lavorativi o produttivi in cui il sistema è utilizzato costantemente, la velocità diventa fondamentale per mantenere alta la produttività.

## Quando

- ⇒ Utilizzo giornaliero e intenso
- ⇒ Attività ripetitiva

Per task che si ripetono frequentemente, anche un piccolo miglioramento in termini di velocità può avere un grande impatto complessivo.

# 3 Soddisfazione

## Obiettivo

- ⇒ 9 su 10 dichiarano che è "bello da usare"

Questo obiettivo si focalizza sull'esperienza soggettiva dell'utente, misurata attraverso feedback e percezioni.

I sistemi self-service, come chioschi o portali web, devono offrire un'esperienza utente positiva per incoraggiare l'uso autonomo e ridurre la necessità di assistenza.

## Quando

- ⇒ Sistema self-service
- ⇒ Business Process Re-engineering incentrato sul nuovo sistema

Durante l'introduzione di un nuovo sistema che modifica processi aziendali, la soddisfazione degli utenti diventa cruciale per garantire una transizione agevole da quello vecchio e il successo del cambiamento.

40

## 4 Facilità di navigazione

### Obiettivo

L'utente può navigare fluidamente nel sistema, eseguendo operazioni correlate senza interruzioni. Si elimina il bisogno di passaggi ridondanti o di tornare continuamente al menu principale.

- ⇒ Possibilità di innescare 6 diverse attività su un singolo oggetto senza ritornare al menu principale

Garantire  
**FLESSIBILITÀ**

Garantire  
**RIGIDITÀ**

### Quando

- ⇒ Il cliente "guida il gioco"
- ⇒ Richiami notevoli tra attività
- ⇒ Si attende una decisione... (ristorante)

Quando

- ⇒ Elevato turn-over
- ⇒ L'importante non è prendere decisioni ma seguire una procedura (mensa)

Le operazioni seguono un flusso predefinito e rigido per evitare errori.

L'utente ha il controllo sul flusso delle operazioni e il sistema deve adattarsi alle sue esigenze.

Quando un'attività è strettamente connessa ad altre e richiede un passaggio fluido tra operazioni.

L'utente deve prendere decisioni frequenti o personalizzate durante il processo.

Gli utenti cambiano frequentemente e non hanno il tempo di imparare un sistema complesso.

## 5 Memorabilità

Si riferisce alla capacità degli utenti di ricordare come utilizzare un sistema o un'applicazione, anche dopo un lungo periodo di inattività (Utente aveva già imparato).

### Obiettivo

- ⇒ Riutilizzo, senza ulteriore training, di una applicazione inattiva da 12 mesi

Il design dell'applicazione deve essere semplice e intuitivo, così che gli utenti non abbiano bisogno di rinfrescare la memoria o cercare aiuto. Le funzioni principali devono essere autoesplicative o richiamare immediatamente la familiarità acquisita in precedenza.

### Quando

- ⇒ Utenti saltuari
- ⇒ Applicazioni per circostanze "eccezionali"
- ⇒ Applicazioni di utilizzo secondario
- ⇒ Applicazioni attivate in date precise (scadenze)

Gli utenti che interagiscono con il sistema solo occasionalmente devono essere in grado di riprendere il controllo senza difficoltà.

Sistemi usati solo in situazioni specifiche o rare.

Sistemi che non sono al centro delle attività quotidiane dell'utente ma vengono utilizzati occasionalmente.

Sistemi utilizzati solo in determinate scadenze o periodi dell'anno.

## 6 Prevenzione degli errori

### Obiettivo

- Riduzione della percentuale degli errori incorreggibili (catastrofici)

Il design deve includere controlli, avvisi e meccanismi per evitare che l'utente compia errori irreversibili.

### Quando

- Risultati/prodotti ottenuti "faticosamente"

Se il risultato di un processo richiede un grande sforzo, sia da parte dell'utente che del sistema, è essenziale evitare errori che possano vanificare il lavoro svolto.

- Risultati correlati a fattori di sicurezza

In contesti dove un errore potrebbe mettere a rischio la sicurezza degli utenti, dei dati o del sistema.

- Risultati immediatamente visibili al cliente esterno

Quando gli errori sono evidenti per il cliente o l'utente finale, possono danneggiare la reputazione del prodotto o del servizio.

Qual è il momento del ciclo di vita in cui entra in gioco l'interfaccia?

## Metodologia di progetto

L'interfaccia viene considerata fin dalle primissime fasi del progetto, per garantire che sia progettata in base ai requisiti di usabilità e agli obiettivi del sistema.

### Prima del termine dello studio di fattibilità

- definire le attività legate alla realizzazione dell'interfaccia
- definire i parametri di riferimento ed i criteri di usabilità
- pianificare le attività di valutazione dell'usabilità
- realizzare il modello concettuale dell'interfaccia

Prevedere come testare l'interfaccia, con quali strumenti e metodi.

Creare una rappresentazione preliminare che descriva come gli utenti interagiranno con il sistema.

### Precocemente nella fase di analisi e progettazione

- Definire e realizzare le strutture base (dialogo, look & feel)
- Stabilire gli standard di progetto per l'interfaccia
- Prototipare le parti ritenute critiche
- Verificare l'allineamento con modello concettuale e standard

L'interfaccia entra in una fase più concreta in cui si sviluppano prototipi e si stabiliscono standard visivi e funzionali.

Progettare il layout e gli elementi visivi principali, come pulsanti, menu e schermate.

Creare linee guida per assicurare coerenza tra tutte le componenti dell'interfaccia.

### Nella fase di sviluppo

Durante la fase di sviluppo, l'interfaccia viene finalizzata e integrata con la logica applicativa.

Confrontare il prototipo con gli obiettivi definiti inizialmente, assicurandosi che sia coerente con i requisiti.

- Ultimare l'interfaccia in dettaglio legandola alla logica applicativa

Collegare l'interfaccia al backend e implementare tutte le funzionalità interattive.

I test con l'utente devono essere integrati in ogni fase del ciclo di sviluppo, dal simulatore iniziale fino al test su prototipi completi. Ogni modalità di test ha uno scopo specifico e contribuisce a garantire un'interfaccia che sia intuitiva, funzionale e soddisfacente per l'utente finale.

Fase essenziale per valutare l'usabilità di un'interfaccia.

# Test con l'utente

Ci sono tre principali modalità di test, ognuna con uno scopo e un approccio specifico:

- **Simulatore** (l'utente è passivo)
- **Dimostratore** (l'utente agisce sulle parti critiche)
- **Prototipo** (l'utente agisce sull'intero sistema in beta-release)

Durante i test con l'utente, è essenziale verificare diversi aspetti chiave per garantire la qualità e l'usabilità dell'interfaccia:

## Elementi da verificare:

- ⇒ Il modello concettuale è sufficientemente rappresentato → Il sistema deve riflettere chiaramente le aspettative dell'utente.
- ⇒ Rispetto al progetto l'interfaccia è adatta e gli standard sono rispettati → Verificare che il design sia coerente con le linee guida progettuali.
- ⇒ Adeguato bilanciamento tra flusso predefinito e flessibilità → Rigidità  
L'interfaccia deve guidare l'utente quando necessario, ma anche offrire libertà di personalizzazione o scelta.
- ⇒ Possibilità d'utilizzo alternativo tra mouse e tastiera → L'interfaccia deve supportare diversi metodi di input (Disponibili gli Acceleratori).
- ⇒ Livello d'integrazione dell'utente con l'interfaccia → Verificare quanto l'utente percepisce l'interfaccia come intuitiva e facile da usare.
- ⇒ E' utilizzata la terminologia utente → Le etichette, i messaggi e i termini usati devono essere comprensibili per l'utente finale.