

Università degli Studi di Salerno

-

Riassunti di

Interazione Uomo Macchina

a cura di

**Simone Avossa
Davide Mastrogiovanni**

-

Anno Accademico 2004/2005

FRAMEWORK DELL'INTERAZIONE

Framework

Sono dei modelli di interazione generali, non sono ristretti ai soli sistemi computerizzati. Essi si concentrano sul punto di vista dell'utente e permettono di comparare facilmente sistemi differenti essendo delle astrazioni.

Ciclo di esecuzione e valutazione di Norman

E' un modello basato su azioni

Secondo questo modello l'utente formula un piano d'azione che poi viene eseguito tramite l'interfaccia del sistema.

Quando il piano è stato realizzato, l'utente osserva l'interfaccia per valutare il nuovo stato del sistema e stabilire ulteriori azioni. Il modello di Norman considera il sistema solo limitatamente all'interfaccia.

Questo ciclo interattivo può essere diviso in due fasi principali: *esecuzione* e *valutazione*, che, a loro volta, possono essere suddivise in altre fasi, sette in tutto:

1. L'utente stabilisce l'obiettivo.
2. Formula l'intenzione. (come fare)
3. Specifica la sequenza di azioni sull' interfaccia.
4. Esegue l'azione.
5. Percepisce il nuovo stato del sistema.
6. Interpreta lo stato del sistema.
7. Valuta lo stato del sistema rispetto all'obiettivo.

Golfo dell'esecuzione

La formulazione delle azioni (le intenzioni) da parte dell'utente può essere diversa dalle azioni permesse dal sistema. Il golfo dell'esecuzione è la differenza che intercorre fra la formulazione delle azioni da parte dell'utente e le azioni consentite dal sistema.

Solo se le azioni previste dal sistema corrispondono a quelle previste dall'utente, l'interazione sarà efficace.

Golfo della valutazione

Le aspettative dell'utente relativamente allo stato del sistema modificato possono differire dalla reale presentazione di questo stato.

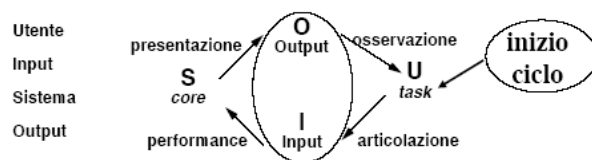
Il golfo della valutazione è la distanza fra la presentazione fisica dello stato del sistema e l'aspettativa dell'utente.

Più sforzo è richiesto da parte dell'utente per interpretare lo stato del sistema, meno efficace sarà l'interazione.

Abowd Beale

Tenta una descrizione più realistica dell'interazione uomo-macchina, includendo esplicitamente il **sistema**. I nodi rappresentano i quattro componenti principali: *Utente*, *Input*, *Sistema* ed *Output*, ognuno dei quali ha il proprio linguaggio e l'interazione consiste in un processo di traduzione da un linguaggio ad un altro. Vi sono quattro fasi nel ciclo interattivo, ciascuna delle quali corrisponde ad una conversione di concetti da un componente ad un altro. Si intercorre in problemi d'interazione quando la traduzione da un linguaggio all'altro è difficile.

Le interazioni dell'utente tradotte in azioni sull'interfaccia, sono tradotte in alterazioni dello stato del sistema, che sono a loro volta riflesse nel display di output, che viene poi interpretato dall'utente.



Ergonomia

Studio delle caratteristiche fisiche dell'interazione.

- Disposizione dei controlli e dei display: raggruppati secondo la funzione, frequenza d'uso, sequenzialità.
- Ambiente circostante: la seduta adattabile ad ogni misura di utente.
- Salute: la posizione fisica, la temperatura, l'illuminazione ed il rumore.
- Uso dei colori: non bisogna usare i colori come un unico tipo di segnalatore, ma unito ad altre informazioni. I colori devono corrispondere a convenzioni comuni (rosso, verde, ecc).

STILI DELL'INTERAZIONE

Stili dell'interazione

L'interazione è un dialogo tra calcolatore e l'utente, ci sono diversi tipi di dialogo.

Interfacce a riga di comando

Si esprimono istruzioni direttamente al computer per mezzo di una combinazione di abbreviazioni e parole intere. Meglio per utenti esperti perché si basa sulla memoria e per aiutare l'utente si usano nomi significativi.

Menù

Insieme di opzioni disposte sullo schermo. Le opzioni sono visibili e richiedono meno memoria. Esse si selezionano con dispositivi di puntamento o con la tastiera. Spesso le opzioni sono raggruppate gerarchicamente.

- Pull-Down: trascinati giù da un titolo al top dello schermo.
- Pop-Up: compaiono quando si clicca su una particolare regione dello schermo.
- Pin-Up: rimangono sullo schermo finché non gli viene chiesto esplicitamente di scomparire.
- Fall-Down: simile a pull-down, ma la barra non è selezionata esplicitamente.

Linguaggio naturale

Riconoscimento parlato o scritto. Vago ed ambiguo. Ristretto ad un sottoinsieme del linguaggio naturale.

Question/Answer e Query

L'interazione è condotta mediante una serie di domande. Adatto per utenti inesperti, ma offrono funzionalità ridotte. I linguaggi d'interrogazione sono usati per ritrovare informazioni nei database, ma è richiesta la comprensione della struttura del database.

Form-Fill e Spreadsheets

I Form-Fill riproducono lo schema di un modulo di cartaceo e per questo richiedono una buona progettazione per porre i dati importanti in posizioni rilevanti.

Gli spreadsheet sono una griglia di celle ciascuna delle quali può contenere un valore o una formula. L'utente può immettere ed alterare i valori delle celle ma la formula rimarrà sempre corretta.

WIMP

Windows Icons Menus Pointers, raggruppa gli stili d'interazione precedenti in un unico stile.

- Windows: sono aree dello schermo che si comportano come terminali indipendenti e possono contenere testo e grafica. Inoltre possono sovrapporsi, ridimensionarsi, oscurarsi e disporsi l'una affianco all'altra. Sono dotate di barre di scorrimento che consentono di scorrere il contenuto della finestra sia orizzontalmente che verticalmente.
- Icons: piccole figure o immagini che rappresentano un oggetto dell'interfaccia. Le finestre possono essere ridotte a questa rappresentazione.
- Menus: vedi menù sopra.
- Pointers: componente su cui si basa lo stile WIMP. Si basa sul puntamento e la selezione di oggetti dell'interfaccia sensibili (Icons). Ottenuto con il mouse o altri sistemi di puntamento, che permettono di far muovere un'ampia varietà di cursori.
- Widgets:
 - Ø Bottoni: regioni individuali ed isolate all'interno di un display, che possono essere selezionate per invocare un'azione.
 - Ø Bottoni radio: insieme di scelte mutuamente esclusive.
 - Ø Check box: insieme di scelte non esclusive.
 - Ø Palette: indicano un insieme di modalità possibili disponibili, oltre a quella corrente. In genere si presentano come una collezione di icone disposte a mattonella.

PARADIGMI DI INTERAZIONE

Paradigmi

Esempi di sistemi interattivi di successo.

La storia del progetto di sistemi interattivi fornisce paradigmi per progetti usabili.

Metafora

Mette la computazione in relazione con altre attività del mondo reale e può essere considerata effettivamente una tecnica di apprendimento.

Per esempio la gestione dei file su una scrivania d'ufficio, l'elaborazione di testi come una macchina da scrivere e la realtà virtuale (l'utente dentro la metafora).

L'interfaccia è progettata in modo da essere simile all'entità fisica, ma ha anche delle caratteristiche proprie. Sfrutta la familiarità degli utenti per aiutarli a comprendere ciò che a loro non è familiare.

Vantaggi:

- Apprendimento più facile
- Possono essere innovative in modo da rendere il mondo dei computer più accessibile.
- Aiutano gli utenti a comprendere il modello concettuale sottostante.

Svantaggi:

- Possono violare le regole convenzionali e culturali (*cestino sul desktop*).
- Conflitti con i principi di progettazione (*forzare troppo la metafora rende il sistema poco chiaro*).
- Si possono utilizzare dei cattivi design esistenti (*utilizzo di cattivi esempi nella metafora*).
- Limitano l'immaginazione del progettista (*limitarsi alla metafora esclude soluzioni innovative*).
- Gli utenti comprendono il sistema solamente in termini della metafora (*si tenta di rapportare la stessa metafora su sistemi diversi*).

Manipolazione Diretta

Metafora del mondo ideale: l'interfaccia non è un mediatore tra utente e sistema, ma dal punto di vista dell'utente essa è il sistema. Le operazioni eseguite sull'interfaccia evitano la necessità di comprenderne il significato ad un livello più profondo del sistema.

Non c'è più una netta distinzione fra linguaggio di Input e linguaggio di Output. Le espressioni di output vengono infatti utilizzate per articolare espressioni di input. E.g. l'icona del documento è l'output, ma viene utilizzata come input per l'operazione di spostamento. I widgets sono così oggetti di output e di input

Correlato alla Manipolazione Diretta è il paradigma "What You See Is What You Get" (WYSIWYG), dove ciò che conta nelle interfacce è la semplicità e l'immediatezza della *corrispondenza* tra *rappresentazione* e *prodotto finale*.

I suoi principi base sono:

- Continua rappresentazione degli oggetti e azioni d'interesse.
- Azioni fisiche e pressione di pulsanti anziché impartire comandi con una sintassi complessa.
- Rapide azioni reversibili con feedback immediato.

Shneiderman definisce la Manipolazione Diretta attribuendogli le seguenti caratteristiche:

- Visibilità degli oggetti che interessano.
- Azione incrementale e feedback rapido su tutte le azioni.
- Reversibilità delle azioni in modo da incoraggiare gli utenti nell'esplorazione.
- Correttezza sintattica di tutte le azioni in modo che ogni operazione sia ammessa.
- Sostituire il linguaggio dei comandi con le azioni per manipolare gli oggetti visibili.

Vantaggi:

- Gli utenti inesperti imparano rapidamente le funzionalità di base ed acquistano sicurezza.
- Gli utenti esperti portano avanti in modo molto veloce più task contemporaneamente.
- Messaggi di errore raramente necessari.

Svantaggi:

- Non tutte le azioni possono essere eseguite direttamente.
- Alcuni task sono svolti in maniera più efficace se si delega il sistema.

Paradigma delle azioni

Le azioni fatte sull'interfaccia fanno superare la necessità di comprendere il significato a qualsiasi livello più profondo. Tuttavia alcune attività sono più difficili da esprimere tramite azioni, se non impossibili. Le interfacce a manipolazione diretta implicano il paradigma delle azioni.

Paradigma del linguaggio

L'interfaccia è un mediatore tra l'utente ed il sistema: l'utente dà all'interfaccia delle istruzioni ed è poi responsabile dell'interfaccia controllare che quelle istruzioni vengano eseguite.

A questo paradigma linguistico si possono associare due interpretazioni significative:

- La prima richiede che l'utente capisca il funzionamento del sistema di base e l'interfaccia come interlocutore non debba eseguire molta traduzione. Questa interpretazione di paradigma linguistico è simile al tipo d'interazione che esisteva prima che fossero introdotte le interfacce di manipolazione diretta.
- La seconda non esige che l'utente comprenda la struttura del sistema di base e prevede che l'interfaccia rivesta un ruolo più attivo, fungendo da interprete tra l'operazione progettata dall'utente e le operazioni possibili del sistema che devono essere attivate per eseguire quel progetto.

La "programmazione per esempio" comprende sia il linguaggio che le azioni: l'utente esegue dei task di routine nel paradigma delle azioni ed il sistema registra ciò come procedura generale. Il sistema interpreta le operazioni dell'utente come uno script linguistico che poi si può eseguire.

Windows Menus Icons Pointers (WIMP)

Gli uomini possono portare avanti più di un lavoro alla volta, usando le finestre per partizionare il dialogo, "cambiando argomento".

Computer Supported Cooperative Work (CSCW)

Con l'arrivo delle reti di computer fu possibile la comunicazione tra macchine diverse. Un risultato di ciò fu l'affermarsi delle tecniche di collaborazione tra singoli via computer, il cosiddetto Computer Support Cooperative Work (CSCW). La principale differenza fra i sistemi CSCW e i sistemi interattivi progettati per un singolo utente è che i progettisti non possono più trascurare l'ambiente sociale all'interno del quale ogni individuo opera. I sistemi CSCW vengono costruiti per consentire l'interazione tra molte persone attraverso il computer e un solo prodotto deve soddisfare le necessità di molti.

Un ottimo esempio di CSCW è l'email (posta elettronica), un'altra metafora per mezzo della quale singoli utenti posti in posizioni fisicamente separate possono comunicare con messaggi elettronici che funzionano in un modo simile ai sistemi postali convenzionali. Gli scambi di comunicazione tra siti di tutto il mondo avvengono nel giro di minuti e non più di settimane.

World Wide Web (WWW)

Il web è costruito sopra Internet ed offre un'interfaccia prevalentemente grafica di facile uso per le informazioni, nascondendo la complessità di base dei protocolli di trasmissione, degli indirizzi e dell'accesso remoto ai dati. Nonostante Internet esistesse dal 1969 non diventò un paradigma di interazione fin quando non divennero disponibili delle interfacce grafiche ben progettate (browser) che consentissero l'accesso a informazioni multimediali

Multimodalità

Combina molte modalità di input simultanee che possono presentare le informazioni in molte modalità di output diverse. Un modo è un canale di comunicazione umano. Particolare enfasi va posta sull'uso simultaneo di più canali di comunicazione per l'input e per l'output.

- Il computer come strumento: la macchina è uno strumento passivo e cerca di comprendere l'utente attraverso tutte le diverse modalità di input che riconosce.
- Il computer come partner di dialogo: le modalità multiple sono usate per aumentare l'antropomorfismo della macchina (riconoscimento parlato). Output come teste parlanti e altre modalità "Human like"

Messaggi di errore

Shneiderman definisce delle linee guida per i messaggi di errore:

- Evitare i termini come FATAL, INVALID, BAD.
- Avvertimenti audio.
- Evitare MAIUSCOLE e codici con lunghi numeri.
- Messaggi precisi.
- Fornire aiuto relativo al contesto.

Antropomorfismo

Attribuire qualità umane ad oggetti inanimati. Noto nella pubblicità e molto sfruttato nell'interazione uomo-macchina per rendere più divertente l'esperienza dell'utente e ridurre la sua ansietà.

Vantaggi:

- Incoraggia l'utente a continuare nell'interazione in caso di feedback positivi.
- Riduce l'ansia degli utenti inesperti.
- Può coinvolgere l'utente

Svantaggi:

- Può far sentire inferiori gli utenti in caso di feedback negativi e provocare ansia.
- Feedback personalizzati possono rendere gli utenti meno responsabili delle proprie azioni.

Legato al paradigma dell'antropomorfismo è quello delle “interfacce basate su agenti” che operano per conto degli utenti all'interno del mondo elettronico in modo da metterli più a loro agio e non farli sentire in difficoltà. A volte invece frustranti

Gli *agenti* di interfaccia si possono classificare sulla base del grado di antropomorfismo che esibiscono in :

- Personaggi sintetici (autonomi in grado di rispondere ad eventi esterni)
- Agenti animati (Giocano un ruolo collaborativi sull'interfaccia)
- Agenti emotivi (Personalità predefinita e un insieme di emozioni che l'utente può cambiare)
- Agenti conversativi emotivi (Caratteristiche fisiche “Human like” e sofisticate intelligenze artificiale per permettere l'interazione)

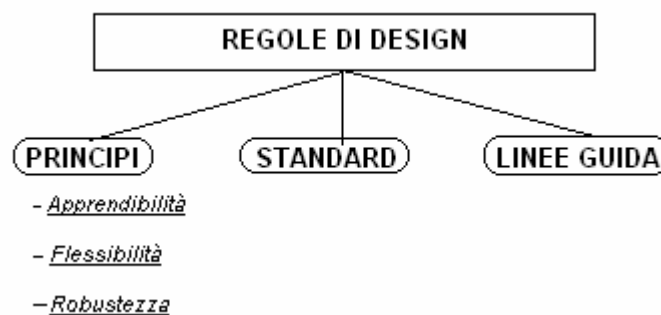
REGOLE DI DESIGN

Regole che un progettista segue per aumentare l'USABILITA' del prodotto software finale. Le si può classificare sulla base di due elementi :

- **AUTORITA'** : indicazione del fatto che la regola debba essere eseguita obbligatoriamente o sia solo un suggerimento
- **GENERALITA'** : se la regola può essere applicata in molte situazioni di progetto o se è valida solo in una situazione più specifica

Le regole di design possono essere suddivise in tre tipi di regole principali:

1. **PRINCIPI DI USABILITA'** : regole di design astratte con alta generalità e bassa autorità.
2. **STANDARD** : regole di design specifiche con alta autorità e limitata applicabilità.
3. **LINEE GUIDA** : tendono ad avere minore autorità ed essere più generali nell'applicabilità.



PRINCIPI DI USABILITA'

I principi di usabilità sono un mezzo più generale per comprendere e promuovere l'usabilità. Guidati teoricamente da conoscenza psicologica, computazionale e sociologica. Vengono raggruppati in tre categorie principali:

- 1. APPRENDIBILITA' O LEARNABILITY** : facilità con cui i nuovi utenti possono iniziare un'interazione efficace e raggiungere le massime prestazioni. Si suddivide in :
 - **Predicibilità**: per determinare l'effetto di azioni future è sufficiente ragionare sulla base della storia delle interazioni precedenti.
 - **Sinteticità**: l'utente deve poter assestare e valutare l'effetto delle operazioni precedenti sullo stato corrente. Quando un'operazione cambia lo stato del sistema è importante che l'utente si accorga della modifica. In alcuni casi questa notifica può arrivare immediatamente non richiedendo alcuna ulteriore interazione dell'utente (*Onestà immediata*). In altri casi può arrivare alla fine dopo esplicite direttive dell'utente (*Onestà ritardata*). (E.G. Confronto del copia incolla in sistemi a Manipolazione diretta e in sistemi a riga di comando.)
 - **Familiarità** : Gli utenti portano con se molta esperienza ottenuta attraverso l'interazione con altri sistemi. Si è interessati al come la conoscenza di sistemi precedenti si applica al nuovo sistema. Il sistema dovrebbe risultare intuitivo
 - **Generalizzabilità** : gli utenti cercano di estendere la conoscenza specifica di una interazione a nuove situazioni che sono simili ma che non hanno mai incontrato prima. Un buon esempio di generalizzazione fra un'applicazione ed un'altra si nota negli editor di testi che tentano di fornire le operazioni di copia e incolla nello stesso modo.
 - **Coerenza**: similitudine nel comportamento input/output che nasce da situazioni o obiettivi simili. Si può avere ad esempio consistenza nel nominare i comandi o nell'invocazione dei comandi.
- 2. FLESSIBILITA'**: si riferisce alla molteplicità di modi in cui l'utente e il sistema si scambiano informazioni. Si suddivide in:
 - **Iniziativa nel dialogo**: il sistema può iniziare il dialogo e l'utente risponde semplicemente alle richieste di informazioni. (dialogo **System pre-emptive**). Il sistema non può essere interrotto. Una finestra modale fa sì che l'utente possa comunicare con il sistema solo tramite la finestra. In alternativa l'utente può essere libero di interrompere il sistema e di intraprendere qualsiasi azione verso di esso (dialogo **User pre-emptive**). In generale si tenta di aumentare la capacità dell'utente di interrompere il sistema e ridurre la possibilità che il sistema interrompa l'utente. Alcune situazioni possono però richiedere un dialogo system pre-emptive (E.G. cancellazione di testo in un editor cooperativo).
 - **Multithreading**: capacità del sistema di supportare l'interazione con l'utente su più task contemporaneamente. Consente quindi al sistema di supportare più di un compito alla volta. Il multithreading può essere **concorrente** quando consente la comunicazione separata di informazioni che riguardano compiti separati o **interfogliato** permette una sovrapposizione temporale tra compiti separati ma in ogni dato momento il dialogo si limita ad un solo compito. Un sistema a finestre supporta un dialogo interfogliato, mentre un sistema multimodale potrebbe supportare un dialogo concorrente.
 - **Migrabilità di un task** : riguarda il passaggio di responsabilità e di controllo fra utente e computer per l'esecuzione di un task. Dovrebbe essere possibile per un utente o un sistema passare il controllo all'altro in modo da ottimizzare alcuni compiti. Un ottimo esempio di migrabilità dei task avviene nel controllo dell'ortografia.
 - **Personalizzazione**: modificabilità dell' interfaccia utente da parte dell'utente o da parte del sistema. Si fa una distinzione tra modifica iniziata dall'utente (**adattabilità**) e modifica iniziata dal sistema (**adattività**). L'adattabilità avviene quando ad esempio l'utente personalizza l'interfaccia sulla base di sue preferenze spostando magari dei pulsanti. L'adattività invece è la personalizzazione automatica dell'interfaccia sulla base dell'osservazione dei compiti utente.
 - **Sostituibilità**: concedere di sostituire fra loro valori di input e di output equivalenti. Ciò contribuisce alla flessibilità permettendo all'utente di scegliere la forma che si adatta meglio. Possiamo avere sostituibilità per l'**input** in modo da immettere valori anche in diverse unità di misura e per l'**output** avendo molteplicità nella rappresentazione degli stati del sistema. A questo punto è legato anche il concetto di **pari opportunità** dove non riscontriamo differenza fra input e output. (E.G. In un pacchetto da disegno tramite manipolazione diretta possiamo disegnare una linea e il sistema calcola la lunghezza, oppure possiamo inserire le coordinate e il sistema disegna la linea)

3. **ROBUSTEZZA**': Il livello di supporto fornito all'utente nel determinare un comportamento di successo rispetto ai suoi goal.
- **Osservabilità**: capacità dell'utente di valutare lo stato interno del sistema tramite la sua rappresentazione percepibile attraverso l'interfaccia. Essa può essere ottenuta tramite:
 - § **Navigabilità** : consente all'utente di esaminare lo stato interno corrente del sistema tramite la visualizzazione fornita sull'interfaccia.
 - § **Default** : i valori predefiniti assistono l'utente tramite un ricordo passivo e riducono il numero di azioni fisiche per immettere un valore
 - § **Raggiungibilità**: si riferisce alla possibilità di spostarsi tra gli stati osservabili del sistema.
 - § **Persistenza**: tratta la durata dell'effetto di un atto comunicativo e la capacità dell'utente di usare quell'effetto.
 - **Recuperabilità**: capacità dell'utente di intraprendere azioni correttive una volta rilevato un errore. Gli utenti spesso commettono errori e vogliono ripristinare lo stato precedente del sistema. La ripristinabilità può avvenire in due direzioni
 - § **forward** (*in avanti*) che implica l'accettazione dello stato corrente e la negoziazione del percorso per raggiungere lo stato desiderato
 - § **backward** (*in indietro*) tentando di annullare gli effetti di una interazione appena eseguita per ritornare a uno stato precedente prima di procedere.
 - § **Principio di sforzo commisurato**: afferma che se è difficile annullare un effetto dato sullo stato dovrebbe essere altrettanto difficile eseguirlo.
 - **Risposta**: come l'utente percepisce la comunicazione con il sistema in termini di tempo che il sistema impiega ad esprimere cambiamenti di stato. Misura la rapidità di comunicazione tra sistema e utente.
 - **Conformità dei task**: il grado con cui i servizi offerti dal sistema supportano tutti i task degli utenti. Poiché lo scopo di un sistema interattivo è consentire a un utente di eseguire vari compiti per raggiungere i suoi obiettivi all'interno di uno specifico dominio, potremmo chiederci se il sistema supporta tutti i compiti di interesse (**completezza dei compiti**) e se lo fa come vuole l'utente. **L'adeguatezza dei compiti** affronta il grado di comprensione che l'utente ha dei compiti.

STANDARD

Stabiliti da organismi nazionali o internazionali per assicurare la conformità da parte di una vasta comunità di progettisti, gli standard richiedono una teoria di base corretta e una tecnologia che cambia lentamente.

LINEE GUIDA

Le linee guida come specificato precedentemente sono più suggestive e generali ed astratte. Le linee guida maggiormente astratte sono applicabili durante le prime attività del ciclo di vita, invece, quelle maggiormente dettagliate sono applicabili durante le attività successive del ciclo di vita.

REGOLE D'ORO ED EURISTICHE

Tutte queste regole descritte in precedenza richiedono al progettista un certo impegno per tener traccia delle linee guida appropriate oppure per interpretare i principi. Molti sostenitori del design centrato sull'utente hanno presentato delle serie di "regole d'oro" o euristiche. Queste sono ovviamente regole imperfette, che possono non essere applicabili in tutte le situazioni, ma forniscono una lista di controllo utile che costituisce una sintesi dei consigli di design disponibili. E' chiaro che un progettista che segue queste norme creerà un sistema migliore di chi le ignora. Le euristiche più famose sono le *dieci di Nielsen*, le *otto regole d'oro di Shneiderman*, e i *sette principi di Norman*.

LE 8 REGOLE D'ORO DI SHNEIDERMAN

Forniscono un riassunto breve ed utile dei principi chiave di design.

1. Preservare la coerenza
 2. Consentire agli utenti abituali di usare comandi rapidi
 3. Offrire feedback informativo
 4. Progettare dialoghi provvisti di chiusura
 5. Offrire una prevenzione e una gestione semplice degli errori
 6. Permettere un'inversione semplice delle azioni
 7. Supportare il controllo interno
 8. Ridurre il carico della memoria a breve termine.
1. **CONSISTENZA - Preservare la coerenza** : sequenze di operazioni simili dovrebbero essere effettuate sempre con lo stesso tipo di azioni. Usare sempre stesse convenzioni e terminologia per i prompt, menù, colori.
 2. **SNELLIMENTI - Consentire agli utenti abituali di usare comandi rapidi**: consentire l'uso di macro ed abbreviazioni per ridurre il numero di interazioni ed abbreviare i tempi di risposta del sistema.
 3. **OFFRIRE FEEDBACK INFORMATIVO**: ad ogni azione del sistema dovrebbe corrispondere un feedback del sistema. Il responso di azioni frequenti dovrebbe essere modesto, mentre azioni occasionali dovrebbero avere responsi più dettagliati. Mostrare i cambiamenti presentando gli oggetti di interesse.
 4. **CHIUSURA – Progettare dialoghi provvisti di chiusura**: organizzare le sequenze di azioni in gruppi, prevedendo feedback informativo alla fine di ciascun gruppo di azioni. Dare all'utente la sensazione di poter scaricare la mente alla fine di una sequenza per dedicarsi interamente al task successivo.
 5. **ERRORI – Offrire prevenzione e gestione semplice degli errori**: progettare l'interfaccia in modo che sia quanto più difficile commettere errori. In presenza di un errore il sistema dovrebbe offrire istruzioni semplici e specifiche per risolverlo. Migliorare i messaggi di errore aumenta la possibilità di usare il sistema con successo. Si possono aiutare gli utenti ad evitare gli errori seguendo tre possibili tecniche:
 - § **Coppie corrispondenti corrette** : la mancata chiusura di una parentesi graffa può essere prevenuta usando un editore "smart". Evita gli errori e la necessità di gestirli ma potrebbe essere preferito un approccio meno rigido. Comunque si cerca di correggere coppie di comandi corrispondenti.
 - § **Sequenze complete**: una fonte di errori è il mancato completamento di una fissata sequenza di passi. Si cerca di evitare questo problema raggruppando sequenze di passi in singole azioni.
 - § **Comandi corretti**: altra causa di errori è l'uso di un linguaggio di comandi che spesso porta errori di battitura, di nomi di comandi etc.. Il completamento automatico può evitare questo tipo di errori riducendo l'uso della tastiera. Non sempre la correzione automatica completa il comando nella maniera voluta.
 6. **REVERSIBILITA' – Permettere un'inversione semplice delle azioni** : rendere le azioni reversibili il più possibile in modo da incoraggiare l'utente a esplorare opzioni sconosciute.
 7. **CONTROLLO – supportare il controllo interno**: dare all'utente la sensazione di essere il responsabile del sistema, che non fa altro che rispondere alle sue azioni. Evitare reazioni sorprendenti.
 8. **MEMORIA – Ridurre il carico della memoria a breve termine**: la memoria umana a breve termine può elaborare un numero limitato di informazioni. (7 ± 2). Mantenere il display semplice e ridurre la frequenza di spostamenti di finestre.

I 7 PRINCIPI DI NORMAN

Sono un riassunto della filosofia di design centrata sull'utente di Norman.

1. **Bisogna usare sia la conoscenza presente nel mondo sia la conoscenza mentale:** le persone lavorano meglio quando le conoscenze di cui hanno bisogno per eseguire un compito sono disponibili all'esterno.
2. **Si deve semplificare la struttura dei compiti:** i compiti devono essere semplici per evitare all'utente una risoluzione complicata dei problemi e un eccessivo carico di memoria.
3. **Si rendano le cose visibili:** bisogna gettare un ponte sul golfo dell'esecuzione e sul golfo della valutazione. L'interfaccia deve mostrare ciò che il sistema può fare e come farlo, permettendo di vedere chiaro l'effetto delle sue azioni
4. **Le corrispondenze vanno chiarite:** le intenzioni dell'utente dovrebbero corrispondere chiaramente ai controlli del sistema e le sue azioni dovrebbero corrispondere chiaramente agli eventi del sistema. Dovrebbe essere chiaro quindi quale controllo fa cosa e in che misura lo fa.
5. **Si sfrutti il potere dei vincoli, sia naturali sia artificiali:** i vincoli sono elementi nel mondo che consentono di eseguire solo l'azione corretta nel modo corretto. Un esempio semplice è il puzzle i cui pezzi si incastrano in un solo modo. I vincoli fisici del design guidano l'utente nel completamento del compito.
6. **Bisogna progettare gli errori:** bisogna anticipare gli errori dell'utente e progettare il recupero del sistema.
7. **Quando tutto il resto non ha successo, si creino degli standard:** se non esistono corrispondenze naturali quelle arbitrarie dovrebbero andare standardizzate. E' questo principio di standardizzazione che permette ad una persona di guidare qualsiasi macchina con pochissime difficoltà.

DESIGN PATTERN

Sono un approccio per riutilizzare le conoscenze sulle soluzioni di progettazione di successo. Un modello è una soluzione parzialmente specificata ad un problema ricorrente all'interno di un contesto specifico.

Si concentrano sulla prassi della progettazione non sulla teoria, raccogliendo le proprietà essenziali comuni ad un buon progetto. Per essere descritti si usa un linguaggio appropriato. Essi sono ripetibili perché astraggono le proprietà generali dei tipi di design.

TECNICHE DI PROTOTIPAZIONE - IMPLEMENTAZIONE

Prototipi si usano prototipi per avere un rapido feedback sul design, per sperimentare design alternativi, per tenere il design centrato sull'utente e per eliminare i problemi legati all'implementazione. Essi simulano le caratteristiche del sistema. Tre tipi di pototipi: Throw-away, Incremental ed Evolutionary

Tecniche di prototipazione:

- **Storyboards:** non sono realizzati necessariamente su supporti digitali, ma possono essere dei "paper-sketches" di interfacce utente. Possono essere animati. Non interagiscono con l'utente, ma si realizzano in poco tempo e con minime capacità tecniche.
Lo-Fi: c'è una bassa fedeltà del prototipo e mancano dettagli rilevanti.
Hi-Fi: assomigliano al prodotto finale.
- **Simulazione di funzionalità:** alcune funzionalità del sistema sono simulate dai progettisti.
Tecnica del Mago di Oz: uno dei designer riceve i comandi in input dall'utente che sta testando il prototipo e li traduce in comandi che il prototipo potrà riconoscere. Ciò consente di intervenire tra l'utente ed il sistema in modo da aumentare le funzionalità e di migliorare il prototipo nelle sue versioni successive.

Architettura dei sistemi a finestre tre possibili architetture software:

- Ogni applicazione gestisce tutti i processi: tutte devono occuparsi della sincronizzazione e così si riduce la portabilità.
- Gestione affidata al kernel del SO: le applicazioni sono vincolate al SO.
- Gestione come applicazione a se stante: massima portabilità.

VALUTAZIONE

Valutazione dell'usabilità

Serve per stimare le funzionalità del sistema, stimare gli effetti dell'interfaccia sull'utente ed identificare specifici problemi. Ne esistono due stili principali:

- **Indagini di laboratorio:** il sistema è studiato all'interno di un laboratorio e la valutazione coinvolge solo designer e valutatori, gli utenti possono essere portati all'interno del laboratorio per effettuare la valutazione. Utile quando la posizione del sistema è impraticabile.

Vantaggi:

- Si può disporre di attrezzature specifiche e l'ambiente è libero da interruzioni.

Svantaggi:

- L'utente può avere una perdita di contesto relativa all'ambiente dove verrà effettivamente usato il sistema e ciò potrebbe causare valutazioni imprecise.
- **Indagini sul campo:** la valutazione è effettuata nell'ambiente di lavoro dell'utente. Utile per osservazioni a lungo termine in cui il contesto del campo è cruciale.

Vantaggi:

- Mantenimento del contesto e studi prolungati nel tempo.

Svantaggi:

- Si può incorrere in distrazioni da parte dell'utente..

Valutazione del design

Viene effettuata in fase di design prima di iniziare l'implementazione per evitare sprechi di risorse. Coinvolge solo gli esperti del settore e può essere applicata su prototipi o su versioni complete del sistema.

Cognitive Walkthrough: valuta quanto il design supporta l'utente nell'apprendimento.

Necessita la *descrizione del prototipo del sistema*, *descrizione di un task*, *lista delle azioni del task*, *indicazioni sugli utenti del prodotto*. I valutatori eseguono una sequenza di azioni sul sistema e per ogni azione rispondono a delle domande:

1. *L'effetto dell'azione è lo stesso dell'obiettivo dell'utente a questo punto?* Ogni azione dell'utente avrà un effetto specifico all'interno del sistema. Questo effetto è lo stesso che l'utente sta cercando di ottenere a questo punto? Per esempio, se l'effetto dell'azione è salvare un documento, è "salvare un documento" ciò che l'utente vuole fare?
2. *Gli utenti si accorgeranno che l'azione è disponibile?* Gli utenti si accorgeranno, per esempio, dell'elemento del menù o del pulsante che è stato usato per produrre l'azione? Questo non vuol dire chiedersi se riconosceranno che il pulsante è quello che vogliono, ma semplicemente se sarà visibile nel momento in cui vorranno usarlo. (Un menù nascosto da un sotto menù potrebbe causare problemi di visibilità).
3. *Una volta che gli utenti hanno trovato l'azione corretta, sapranno che è quella di cui hanno bisogno?* Questo completa la domanda precedente. Una volta che si è reso visibile un pulsante o un elemento del menù, l'utente riconoscerà che è proprio quello che sta cercando per completare il suo compito? (Mentre la domanda precedente riguardava la visibilità dell'azione, questa ha il fine di scoprire se il significato e l'effetto sono chiari).
4. *Dopo avere eseguito un'azione, gli utenti capiranno il feedback che otterranno?* Supponendo che l'utente sia riuscito a compiere l'azione corretta, saprà di esserci riuscito? Il feedback che ottiene sarà una conferma sufficiente di ciò che è effettivamente avvenuto?

Walkthrough pluralistico: è una variazione del cognitive walkthrough, ma il collegio lavora prima separatamente e poi segue una discussione che porta ad una decisione collegiale.

Valutazione euristica: è un metodo per strutturare la critica ad un sistema usando una serie di euristiche generali e relativamente semplici. E' utile in particolare per la valutazione iniziale, ma può anche essere usata su prototipi, storyboard, e sistemi funzionanti.

L'idea che sta alla base è che più lavoratori esaminano autonomamente e criticano un sistema per individuare potenziali problemi d'usabilità. Per aiutare i valutatori a scoprire i problemi d'usabilità viene fornita una serie di 10 euristiche. Ci sono tre fasi principali:

1. Sessione informativa per spiegare agli esperti cosa devono fare.
2. Periodo di valutazione di 1-2 ore in cui:
 - Ogni esperto lavora separatamente.
 - Un primo passaggio per avere un'idea del prodotto.
 - Un secondo passaggio per concentrarsi su aspetti specifici.
3. Sessione in cui gli esperti lavorano insieme per dare priorità sui problemi.

Valutazione basata su revisioni: basa la valutazione su esperimenti analoghi già documentati.

E' fondamentale tener conto delle differenze tra il contesto dell'esperimento e il design che si sta considerando per la valutazione.

Valutazione su modelli: si usano modelli cognitivi e di design per combinare le specifiche e la valutazione del progetto.

Valutazione dell'implementazione

si distingue da quella del design perché richiede un prototipo del sistema o una completa implementazione e coinvolge gli utenti del sistema. Si possono attuare dei metodi empirici, metodi sperimentali e metodi osservazionali che richiedono l'interrogazione degli utenti.

Metodi empirici: si valutano specifici aspetti del comportamento interattivo, mediante la scelta di:

- Ipotesi: predizioni dei risultati dell'esperimento.
- Variabili indipendenti: valori che sono cambiati per produrre condizioni differenti.
- Variabili dipendenti: caratteristiche misurate dell'esperimento.

La scelta dei soggetti (non meno di 10) è effettuata tenendo presente l'utente finale del sistema e di risultati vengono trattati mediante processi statistici.

Metodi sperimentali:

- Within groups design: ogni soggetto esegue gli esperimenti sotto tutte le condizioni. Ci possono essere problemi dovuti al trasferimento di conoscenze, ma possono essere risolti variando l'ordine delle condizioni.
- Between groups design: ogni soggetto esegue gli esperimenti soltanto sotto una condizione. E' richiesto un gran numero di utenti.

Metodi osservazionali:

- Think aloud: l'utente è osservato durante il task e gli è chiesto di descrivere cosa sta facendo ad alta voce.

Vantaggi:

- Semplice da effettuare.
- Fornisce informazioni utili su come il sistema è realmente usato.

Svantaggi:

- I risultati sull'usabilità del sistema sono molto soggettivi.
- **Valutazione cooperativa:** variazione del Think aloud dove il valutatore e l'utente interagiscono durante la sessione, portando l'utente a criticare il prodotto e non solo usarlo. Il valutatore può intervenire nei momenti più critici dell'interazione per indagare sulle difficoltà e verificare proposte alternative.
- **Tecniche di interrogazione:** chiedono all'utente stesso come sia il sistema. Esse sono soggettive e poco costose.
 - **Interviste:** seguono un approccio top-down. Le domande vengono adattate al contesto in modo da esaminare i problemi a fondo. Sono però molto soggettive ed impiegano molto tempo. Esse possono essere strutturate (rigorosamente scritte come un questionario), semistrutturate (guidate da un documento scritto, ma possono essere variate al bisogno), non strutturate (non sono guidate da un documento scritto). Abbiamo delle fasi fondamentali: introduzione, riscaldamento, corpo principale, raffreddamento, chiusura.
 - **Questionari:** le domande possono essere chiuse o aperte, le prime possono essere più facili da analizzare anche al computer. Possono essere somministrate ad ampi fasce della popolazione. Le risposte possono essere basate su scale di valutazione ed è importante definire i punti delle scale stesse al fine di una corretta valutazione. Il più famoso questionario è il SUMI che ha un valore internazionale e richiede circa 5-10 minuti per essere compilato, le sue misure sono basate su efficienza e conoscenza.