

I PRINCIPI DI USABILITÀ

NOTE LEGALI

La riproduzione parziale o totale del presente materiale è consentita solo per usi non commerciali purché sia mantenuto invariato il contenuto e i riferimenti all'autore: obbligatoriamente dovrà essere inserito, in posizione visibile, un link al sito <http://www.amicopc.com>

Sui siti diversi da AmicoPc la consultazione di queste pagine non può essere assoggettata a nessun tipo di pagamento o registrazione, nemmeno gratuita, di qualsivoglia tipo di dati personali (nome, cognome, email o altro). Nel caso in cui, per visualizzare queste pagine, abbiate dovuto fornire dati personali o corrispondere un qualsiasi pagamento, siete pregati di comunicarcelo mediante il modulo disponibile all'indirizzo <http://www.amicopc.com/contact.asp> oppure scrivendo all'indirizzo info@amicopc.com

Lo staff di AmicoPc e gli autori della guide nonché i responsabili dei siti che ospitano queste pagine, non forniscono alcuna garanzia, neppure implicita, sulla correttezza delle informazioni qui riportate. In nessun caso possono essere ritenuti responsabili per eventuali malfunzionamenti, perdite di dati, cali di prestazioni o qualsiasi altro danno dovuto all'utilizzo di tale materiale.

I diritti di autore sui testi e/o sull'altro materiale pubblicato rimangono di esclusiva proprietà degli autori

SOMMARIO

INTRODUZIONE	4
PER CHI VUOL SAPERNE DI PIU'	4
1. Realizzare un dialogo semplice e naturale	6
2. Semplificare la struttura dei compiti	7
3. Agevolare il riconoscimento piuttosto che il ricordo	8
4. Fornire feedback in modo da rendere visibile lo stato del sistema.....	10
5. Prevenire gli errori di interazione e facilitarne il recupero	12
6. Essere consistenti	13
7. Parlare il linguaggio dell'utente	14
8. Agevolare la flessibilità d'utilizzo e l'efficienza dell'utente	15
9. Fornire help e manuali.....	16

INTRODUZIONE

I principi di usabilità sono linee-guida generali, indipendenti da specifiche soluzioni tecniche, che descrivono le caratteristiche che una interfaccia deve avere per essere usabile.

Fondati sul modo di ragionare e operare delle persone quando interagiscono con un prodotto software, rappresentano un riferimento importante di cui tenere assolutamente conto sia in fase di progettazione che in fase di valutazione dell'usabilità (Jakob Nielsen ha basato sui cosiddetti 'principi euristici' un metodo di valutazione noto, appunto, come valutazione euristica).

I principi riportati in queste pagine rappresentano una discreta sintesi di quanto si può rintracciare in letteratura su questo argomento:

- ❖ realizzare un dialogo semplice e naturale
- ❖ semplificare la struttura dei compiti
- ❖ agevolare il riconoscimento piuttosto che il ricordo
- ❖ fornire feedback in modo da rendere visibile lo stato del sistema
- ❖ prevenire gli errori di interazione e facilitarne il recupero
- ❖ essere consistenti
- ❖ parlare il linguaggio dell'utente
- ❖ agevolare la flessibilità di utilizzo e l'efficienza dell'utente
- ❖ fornire help e manuali

PER CHI VUOL SAPERNE DI PIU'

I riferimenti sui principi di usabilità sono numerosi.

Per quanto attiene i cosiddetti 'principi euristici', il riferimento più importante è sicuramente Jakob Nielsen. Dalle pagine del suo sito si può consultare l'ultima versione dei principi euristici

(http://www.useit.com/papers/heuristic/heuristic_list.html), revisionati dopo una decennale esperienza nella loro applicazione. Sempre di Nielsen, sui principi euristici, si può consultare la seguente bibliografia:

- Molich R., Nielsen J. (1990), Improving a human-computer dialogue, 'Communications of the ACM', 33, 338-348;
- Nielsen J. (1993), 'Usability Engineering', Academic Press;
- Nielsen J., Mack R. L. (eds.) (1994), 'Usability Inspection Methods', John Wiley & Sons.

I principi euristici di Nielsen sono riportati, in italiano, in un intervento di Federica Goldoni al COM-PA 1999

(http://www.regione.emilia-romagna.it/dg_sit/convegnodue/usabilita.htm)

Sono anche da consultare:

- i first principles di Bruce Tognazzini, l'autore della prima guida per interfacce grafiche redatta alla Apple (<http://www.asktog.com/basics/firstPrinciples.html>);
- i principi di progettazione definiti e adottati dall'IBM (http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/6);
- il celebre e divertente libro di Norman D. (1990), 'La caffettiera del masochista', Giunti, Firenze;
- i principi espressi dallo standard ISO 9241-10 'Ergonomic requirements for office work with visual display terminals (VDTs) - Dialog principles' (<http://web.tiscali.it/userware/standardiso9241.htm#parte%2010%20deII%27ISO%209241>)

1. Realizzare un dialogo semplice e naturale

Il prodotto software deve proporre all'utente un modello di dialogo che sia coerente con il modello dell'attività dell'utente e con il suo sistema di attese e che, in ogni caso, sia facilmente intuibile interagendo con il prodotto stesso.

La progettazione di un dialogo semplice e naturale è agevolata principalmente da due fattori:

- dalla continua considerazione del modo di lavorare dell'utente, dalle sue caratteristiche e dalle sue esigenze;
- dalla scelta di soluzioni di interfaccia che tengano conto soprattutto dei principi sui compiti, sul riconoscimento, sul feedback e sugli errori.

Alcuni modi per rispettare il principio del dialogo semplice e naturale sono:

- tener conto del modo di operare dell'utente e del suo modello dell'attività nella:
 - organizzazione dei contenuti e della struttura del sistema,
 - implementazione della logica dell'interazione,
 - definizione dell'ordine dei menu,
 - definizione dell'ordine di presentazione delle finestre o delle pagine,
 - organizzazione dei contenuti e degli oggetti all'interno delle finestre o delle pagine;
- fornire un buon modello concettuale del sistema e renderne evidenti l'organizzazione, la logica d'interazione, etc... affinché l'utente possa prevedere gli effetti delle proprie azioni: interagire con un prodotto senza avere compreso e sviluppato un modello di come funziona è come agire alla cieca, senza sapere cosa succederà e cosa aspettarsi a ogni azione;
- rendere evidenti le informazioni rilevanti di cui l'utente ha effettivamente bisogno per svolgere l'attività, evitando di presentare informazioni superflue o che si usano raramente: le informazioni irrilevanti possono entrare in 'competizione' quelle fondamentali e togliere a queste ultime la necessaria visibilità;
- anticipare quanto possibile le informazioni che l'utente troverà nella pagina successiva o quello che succederà a seguito di una azione.

2. Semplificare la struttura dei compiti

I compiti o le attività che l'utente deve svolgere in interazione con il sistema devono avere una struttura semplice. Devono essere progettati e implementati nel sistema riducendo al minimo la necessità di elaborazione delle informazioni da parte dell'uomo dovute all'utilizzo dello strumento informatico.

Donald Norman suggerisce quattro approcci per semplificare i compiti:

- mantenere il compito invariato, ma offrire sussidi mentali;
- usare la tecnologia per rendere visibile quello che altrimenti sarebbe invisibile;
- automatizzare, mantenendo il compito sostanzialmente invariato;
- cambiare la natura del compito.

I primi tre approcci prevedono di non cambiare sostanzialmente il compito che l'utente deve fare.

Nel primo e nel secondo caso, il suggerimento è quello di offrire sussidi esterni (che supportano le capacità cognitive dell'utente non costringendolo a dover ricordare tutto a mente) e feedback (che consente all'utente di controllare le componenti non visibili del sistema al fine di verificarne l'adeguato funzionamento). Ad esempio, nella scrittura a videoterminale, la struttura del compito è sostanzialmente identica alla scrittura manuale. In più, però, la scrittura a videoterminale, grazie al sussidio del correttore automatico che segnala gli errori, consente di migliorare la qualità del lavoro.

Nel terzo caso, pur mantenendo invariata la struttura del compiti, alcune sue parti (quelle più rischiose o complicate) vengono eseguite dalla tecnologia e non più dall'utente.

L'ultimo approccio, infine, quando il compito è intrinsecamente complesso per le abilità richieste, prevede una riprogettazione sostanziale del compito: l'obiettivo rimane, ovviamente, lo stesso, ma il modo in cui viene tale obiettivo viene raggiunto è totalmente diverso.

3. Agevolare il riconoscimento piuttosto che il ricordo

Osservando l'interfaccia, l'utente deve poter capire cosa deve fare, come può farlo e, una volta eseguita una azione, deve poter capire cosa è successo e quali sono stati i risultati.

Dal momento che è più facile riconoscere e ricordare una cosa vedendola direttamente, piuttosto che recuperare l'informazione dalla memoria, il modo più semplice per agevolare l'utente è quello di rendergli visibili le cose sull'interfaccia, ovvero fornirgli dei sussidi esterni che gli agevolano il ricordo.

Dire che le cose devono essere visibili sembra banale, ma non lo è! Ecco alcuni esempi di disattenzione di questo principio:

- essere costretti ad imparare a memoria certi comandi o informazioni (o ricorrere alla guida operativa o chiedere suggerimenti a colleghi e amici più esperti);
- non capire, osservandolo, se un elemento (un campo di edit, un link, un pulsante, etc...) è selezionabile/modificabile oppure no;
- non capire se l'azione richiesta è stata eseguita dal sistema;
- non capire perché l'azione richiesta non è stata eseguita dal sistema;
- navigando su un sito internet, non capivamo più dove ci si trova e da dove si era partiti.

Alcuni suggerimenti per agevolare il riconoscimento:

- sfruttare il 'mapping' naturale, ovvero la correlazione naturale che esiste tra due cose, tra causa ed effetto, tra comandi, loro azionamento e risultati (un esempio di 'mapping' naturale è la manipolazione diretta degli oggetti. Con queste tecniche, l'utente non è costretto a ricordare il modo di utilizzare degli oggetti o a descrivere le azioni da eseguire: semplicemente le esegue direttamente sullo schermo, spostando, ad esempio, un documento dalla scrivania virtuale del suo PC al cestino, così come farebbe nel mondo reale);
- fare in modo che le azioni consentite sull'interfaccia siano chiaramente visibili;
- rendere evidente lo stato del sistema ad ogni momento e ad ogni azione dell'utente (risultati dell'azione svolta, contesto nel quale ci si trova, etc...);
- dotare tutte le pagine di un titolo significativo che illustri adeguatamente il tipo di informazione visualizzata o le azioni da svolgere sulla finestra o

sulla pagina;

- utilizzare un linguaggio e una grafica corretta e significativa per l'utente, in modo da non costringerlo a interpretazioni del significato;
- usare liste di selezione che ricordano le scelte ammissibili e il formato consentito;
- fornire delle informazioni di anteprima sugli oggetti selezionati;
- dotare le icone e i simboli grafici di tool tip descrittive della funzionalità associata all'icona stessa;
- abilitare o disabilitare i comandi in base al contesto operativo, in modo da ricordare all'utente l'obbligatorietà di certe azioni o la relazione sequenziale tra certe altre;
- rispettare la consistenza nell'organizzazione dei contenuti e degli oggetti all'interno delle finestre e delle pagine, in modo da non costringere l'utente a continue osservazioni dell'intera finestra o pagina per ritrovare determinati oggetti o gruppi di informazioni.

4. Fornire feedback in modo da rendere visibile lo stato del sistema

Il feedback rappresenta l'informazione di ritorno in risposta all'azione che l'utente ha eseguito sulla interfaccia e ha lo scopo di rendere visibile all'utente lo stato corrente del sistema, in modo da evitare errori, incomprensioni e blocchi durante l'interazione.

Inteso in questo modo, il feedback non si riferisce solo alle azioni sbagliate da parte dell'utente e alla relativa messaggistica di errore, ma coinvolge tutti i modi per comunicare all'utente cosa sta facendo il sistema al momento corrente:

- quale azione ha compiuto o sta compiendo l'utente;
- quali sono gli effetti della propria azione sul prodotto;
- il nuovo stato del prodotto a seguito dell'azione effettuata.

Oltre alla messaggistica, prevalentemente usata per correggere errori di interazione, i modi per fornire feedback sullo stato corrente del prodotto sono estremamente svariati:

- osservare sulla finestra lo scorrimento dell'oggetto selezionato quando l'utente effettua una azione di 'drag & drop' informa che l'esecuzione dell'azione sta avvenendo correttamente;
- il cambio di forma del puntatore del mouse a seguito della selezione di qualche strumento grafico informa che sono possibili solo certe operazioni e non è possibile, ad esempio, digitare un testo;
- la comparsa di un indicatore di avanzamento informa che per eseguire una determinata operazione da parte del prodotto è necessario un certo tempo;
- l'enfasi di non disponibilità di un oggetto indica che quello strumento non è disponibile o quell'azione non è consentita;
- nessun effetto sull'interfaccia abbinato all'emissione di un segnale sonoro indica che si sta cercando di compiere una azione non consentita allo stato corrente. E così via.

Ma entro quanto tempo deve essere fornito il feedback? Jakob Nielsen, nel suo ultimo libro, 'Designing Web Usability' (New Riders Publishing) riporta le seguenti indicazioni:

- **un decimo di secondo** è, approssimativamente, il tempo per dare all'utente la sensazione che il sistema ha reagito istantaneamente;

- **1 secondo** è, approssimativamente, il tempo per mostrare i risultati dell'azione dell'utente senza interrompere il suo flusso di ragionamento, anche se noterà il ritardo nella risposta del sistema;
- **10 secondi** è il tempo massimo per mantenere l'attenzione dell'utente focalizzata sul dialogo. Oltre i 10 secondi, generalmente, l'utente inizia una altra attività mentre il computer sta lavorando: se si vuole mantenere l'attenzione dell'utente sull'applicazione o sul sito la risposta all'azione dell'utente deve essere mostrata assolutamente entro 10 secondi.

Per quanto riguarda specificamente il web, uno studio della Zona Research del 1999 ha dimostrato che la tolleranza all'attesa da parte degli utenti è più bassa di 10 secondi e si attesta invece sugli **8 secondi**: la media di abbandono da parte degli utenti va dal 30% al 50% se i tempi di attesa per il download delle pagine superano gli 8 secondi.

Lo studio evidenzia anche alcune preferenze e comportamenti tipici degli utenti e, soprattutto, pone l'accento sull'impatto economico connesso a lunghi tempi di attesa nel download delle pagine.

5. Prevenire gli errori di interazione e facilitarne il recupero

Commettere errori nell'interazione con un prodotto è naturale. Anzi, qualsiasi errore che possa teoricamente essere commesso, prima o poi accadrà! Ogni azione dell'utente va concepita come un tentativo verso una giusta direzione. L'errore non è altro che una azione specificata in modo incompleto o inesatto. Si tratta di una componente naturale del dialogo utente-sistema che va tollerata, garantendo la giusta flessibilità di utilizzo che consente agli utenti di navigare liberamente senza entrare in vicoli ciechi e in situazioni critiche.

Ci sono alcuni tipi di errori che sono difficilmente eliminabili, come le sviste: inconsapevolmente, viene eseguita un'azione diversa rispetto a quella che ci proponeva nelle intenzioni, a causa di una distrazione o di una interruzione.

Altri tipi di errore, invece, si possono prevenire con una buona progettazione dell'interfaccia: sono gli errori commessi a seguito di una applicazione sbagliata di regole di interazione o per la mancanza di sufficienti e adeguate informazioni e conoscenze. Rientrano tra questi tipi di errore quelli dovuti ad un modello di dialogo che l'utente non capisce o che non incontra le sue aspettative e per questo egli applica regole di interazione sbagliate rispetto a quelle richieste dal prodotto.

Il contributo fondamentale alla prevenzione degli errori d'interazione deriva, quindi, dal rispetto dei principi sul dialogo, sui compiti, sul riconoscimento e sul feedback sulla cui base è consentito agli utenti di individuare, riconoscere e adeguare le proprie azioni alle possibilità offerte attraverso l'interfaccia. Altri modi di prevenire gli errori prevedono l'utilizzo di funzioni bloccanti, che impediscono la continuazione di azioni sbagliate o che possono portare a risultati distruttivi.

Tuttavia, poiché gli errori sono sempre possibili, è importante anche che il sistema sia progettato in modo da diagnosticarli quando occorrono e facilitarne la correzione.

I modi più semplici per raggiungere questo obiettivo sono:

- fornire funzionalità di annullamento delle operazioni, come 'undo' e 'redo' o di ripristino delle condizioni di default;
- fornire una messaggistica efficace (vedi anche feedback);
- evitare di presentare pagine senza opzioni di navigazioni;
- rendere sempre disponibili le funzioni per uscire dal programma o per ritornare alla home page;
- fornire comandi per interrompere operazioni molto lunghe.

Per chi vuole saperne di più:

Esiste una vasta letteratura sugli errori umani in interazione con le tecnologie informatiche. Per iniziare si può leggere il libro di Reason J. (1990), 'L'errore umano', Il Mulino, Bologna.

6. Essere consistenti

La consistenza si riferisce al fatto che la sintassi (linguaggio, campi di input, colori, etc..) e la semantica (comportamenti associati agli oggetti) del dialogo devono essere uniformi e coerenti all'interno di tutto il prodotto software.

La consistenza permette all'utente di trasferire agevolmente la conoscenza da una applicazione all'altra, aumenta la predicibilità delle azioni e dei comportamenti del sistema e ne favorisce l'apprendibilità.

Un problema di consistenza che 'assilla' il mondo web è relativo ai font e ai link.

Molto spesso, all'interno dello stesso sito, si vedono pagine con font diversi per dimensione, stile e colori. Analogamente, i link vengono proposti in svariati formati: standard (sottolineato), in grassetto (ma senza sottolineatura), colorati senza limiti alla fantasia.

Le pagine web sono da immaginare come pagine di un libro. E non si è mai visto un libro 'importante' che usa caratteri diversi per ogni paragrafo, titoli di dimensioni e colori differenti posizionati una volta centrali e una volta a sinistra sulla testata della pagina, rimandi ad approfondimenti ogni volta presentati con uno stile diverso!

L'inconsistenza nei font, nella struttura della pagina, nella grafica genera una situazione di confusione, fornisce l'impressione di una mancanza di cura e di attenzione e, in definitiva, di professionalità.

In sintesi, la consistenza deve essere garantita a diversi livelli:

- **consistenza del linguaggio e nella grafica:** la stessa parola, la stessa icona, lo stesso colore devono identificare lo stesso tipo di informazione o lo stesso tipo di azione entro tutto il prodotto;
- **consistenza degli effetti:** gli stessi comandi, le stesse azioni, gli stessi oggetti devono avere lo stesso comportamento e produrre gli stessi effetti in situazioni equivalenti; non associare agli stessi comandi, azioni e oggetti comportamenti diversi;
- **consistenza nella presentazione:** gli stessi oggetti o lo stesso tipo di informazioni devono essere collocati tendenzialmente nella stessa posizione, avere la stessa forma e lo stesso ordine;
- **consistenza tra ambienti applicativi:** una applicazione, un sito non sono mondi isolati! Gli utenti utilizzano differenti applicazioni e navigano tra siti diversi, e imparano come funzionano certi oggetti di interfaccia. Entrando nella nostra applicazione/sito si aspettano di ritrovare la stessa tipologia di oggetti che si comporta nel modo che hanno imparato. Soluzioni alternative, utilizzo non convenzionale degli oggetti grafici inducono solo incertezza di utilizzo e aprono la porta agli errori di interazione.

7. Parlare il linguaggio dell'utente

Il linguaggio utilizzato a livello di interfaccia deve essere semplice e familiare per l'utente e rispecchiare i concetti e la terminologia a lui noti.

Va evitato il più possibile un linguaggio tecnico e orientato al sistema che utenti non esperti di informatica possono non comprendere e termini stranieri. Vanno evitate, ad esempio, parole come 'default', 'directory', o frasi del tipo 'il documento è disponibile in formato.rtf' o 'documento.zip trasferibile via ftp'.

Rientrano tra gli elementi del linguaggio anche le icone e le metafore, forme per rappresentare concetti in forma grafica e simbolica, che, se ben realizzate, possono agevolare la comprensione in modo più efficace e diretto rispetto alle parole.

La definizione di un linguaggio adeguato e significativo per l'utente, soprattutto se simbolico, è comunque un lavoro ben più difficile di quanto generalmente si pensi poiché comporta conoscere molto bene il mondo degli utenti.

Due consigli:

- verificare con gli utenti la comprensibilità del linguaggio (etichette, istruzioni, elenchi, etc...);
- usare preferibilmente icone e metafore già sperimentate e consolidate; se si vogliono usare simboli originali è sempre meglio sottoporli prima a test di comprensibilità con gli utenti finali del prodotto.

8. Agevolare la flessibilità d'utilizzo e l'efficienza dell'utente

Pensare alla produttività dell'utente!

Nella definizione degli strumenti in grado di agevolare la flessibilità e l'efficienza, va considerato che le esigenze degli utenti variano in relazione al loro livello di esperienza rispetto al compito e alle tecnologie informatiche. Ne consegue che, in relazione a questi due aspetti, il livello di supporto richiesto, gli strumenti utilizzati e le strategie di interazione messe in atto dagli utenti possono essere diverse. Gli utenti non esperti, ad esempio, amano essere guidati passo per passo, mentre gli utenti più esperti preferiscono utilizzare scorciatoie, delle quali anche utenti non esperti man mano che aumenta il loro livello di esperienza possono usufruire.

Si può agevolare la flessibilità e l'efficienza d'uso fornendo, ad esempio:

- 'facilities' d'inserimento (ad esempio, l'anticipazione da parte del sistema nell'inserimento di un termine) e acceleratori (una combinazione di tasti, come ad esempio, CTRL V);
- salti nella navigazione che evitano di passare in punti intermedi;
- funzioni di personalizzazione dell'interfaccia, ovvero la possibilità di modificare alcuni aspetti del sistema in base alle esigenze del compito, alle caratteristiche dell'utente e sue preferenze personali. Da notare che, una volta personalizzato, il sistema deve mantenere, alle successive riaperture, le impostazioni date dall'utente. Alcuni esempi di aspetti dell'interfaccia che dovrebbero essere personalizzabili sono:
 - la lingua,
 - la dimensione dei caratteri,
 - le impostazioni di default,
 - il formato dei dati presentati e livello di dettaglio,
 - la disposizione di alcuni oggetti grafici.

Un altro aspetto dell'efficienza è anche il tempo di risposta del sistema alle azioni dell'utente, problema che nelle applicazioni web è tra i più critici.

I tempi di risposta ad ogni chiamata dell'utente possono dipendere da molte cose (performance dei server e della rete, tipo e velocità di connessione, quantità di informazione che deve essere trasferita, etc...), ma gli utenti non sono interessati a queste motivazioni. Se l'attesa è lunga, gli utenti pensano solo che non viene offerto loro un buon servizio e il livello di fiducia nel fornitore è destinato a diminuire.

Nelle applicazioni e nei siti web, **il tempo massimo di attesa non deve superare gli 8 secondi**, tempo oltre il quale gli utenti non mantengono più l'attenzione sul dialogo e iniziano a fare un'altra cosa (sui tempi di risposta vedi anche feedback).

9. Fornire help e manuali

L'argomento della documentazione (help in linea o manuali utente) è piuttosto controverso per diversi motivi:

- un buon prodotto, teoricamente, non dovrebbe richiedere la consultazione della documentazione;
- la documentazione viene spesso usata per compensare eventuali problemi di usabilità del prodotto;
- nella maggior parte dei casi, gli utenti ignorano questi strumenti di supporto.

Per quanto riguarda l'ultimo punto, infatti, generalmente gli utenti ricorrono all'help in linea o alla documentazione solo come ultimo tentativo, cercando (e non trovando quasi mai) la soluzione al proprio caso specifico. Nella lettura delle informazioni riportate, inoltre, tendono a non approfondire gli argomenti, leggendo rapidamente solo poche righe. Infine, come per tutti i testi scritti, la comprensibilità di help e manuali, se non accuratamente verificata con gli utenti finali, non è sempre garantita.

Considerati questi aspetti, quando la documentazione può essere necessaria, essa va realizzata con l'obiettivo di garantire:

- facilità di consultazione,
- comprensibilità e brevità dei testi,
- orientamento all'attività dell'utente,
- efficacia nella risoluzione del problema.