

# Usabilità del software

*Linee guida per realizzare interfacce  
utente intuitive, amichevoli, facili da  
usare*

Ercole Colonese  
Senior Consultant

Versione 1.0 - Aprile 2000

**Versione del documento**

2000  
V1.0

2001

2001

2002

2003

## Indice

Introduzione.....	5
Modello concettuale delle interfacce utente .....	7
Modello dei casi d'uso per le interfacce utente .....	10
Guida alla progettazione delle interfacce utente .....	13
Tabella Oggetti/Azioni .....	15
Validazione dell'usabilità.....	16
Aderenza agli standard (CUA o altri standard) .....	16
Consistenza grafica dei dialoghi.....	17
Validità della messaggistica .....	17
Usabilità del disegno dei dialoghi .....	18
Revisione euristica dell'usabilità.....	20
Test di usabilità .....	22
Metriche di usabilità.....	24
Riferimenti bibliografici.....	25



# Usabilità del software

*Linee guida per realizzare interfacce utente intuitive, amichevoli, facili da usare*

## Introduzione

Una misura del successo di un'applicazione software è sicuramente la sua capacità di permettere agli utenti di completare con successo ed agevolmente le proprie attività di lavoro. In particolare, le applicazioni progettate con un'interfaccia utente rilevante devono essere progettate seguendo criteri, metodi e tecniche appositamente studiati e sperimentati allo scopo.

L'usabilità, appunto, è la caratteristica chiave che le interfacce utente devono possedere per agevolare gli utenti nello svolgimento delle loro attività, per consolidare la loro soddisfazione e per garantire l'accettazione dell'applicazione stessa.

Le metodologie consolidate permettono di progettare e valutare interfacce usabili secondo il seguente schema:

- Analisi e disegno delle interfacce utente utilizzando i seguenti modelli e linee guida:

- Modello concettuale delle interfacce utente
- Modello dei casi d'uso per le interfacce utente
- Linee guida per la progettazione delle interfacce utente
- Validazione dell'usabilità tramite le seguenti tecniche:
- Revisione euristica di usabilità
- Test di usabilità

Segue una descrizione degli elementi elencati.

*Ogni organizzazione ha la propria metodologia per lo sviluppo del software. Le tematiche dell'usabilità sono generalmente trattate e indirizzate da opportune best practices. IBM, in particolare, tratta l'argomento dell'usabilità del software in maniera completa ed approfondita nella propria metodologia: User-Centered Design (UCD). L'autore, pur conoscendo tale metodologia, tratta il tema dell'usabilità in questo documento dal punto di vista pratico, oltre che teorico, fornendo suggerimenti utili per affrontare le problematiche nelle attività quotidiane.*

### Modello concettuale delle interfacce utente

Chiariamo prima alcuni concetti importanti per la comprensione del modello.

**Modello mentale:** E' una struttura cognitiva che le persone utilizzano per capire e rappresentare come il mondo reale è strutturato e come esso funziona. Per esempio, esistono diversi modelli concettuali di come scrivere una lettera: tramite carta e penna, tramite un programma di video-scrittura, tramite la registrazione in un magnetofono e successiva scrittura su carta, ecc. Il modello assicura che tutti capiscano come l'attività può essere completata con successo.

**Oggetti delle interfacce utente:** Sono gli elementi che rappresentano gli oggetti del mondo reale (esempio: documenti, cartelle, cestini, telefoni, ecc.). Le relazioni tra gli oggetti definiscono le modalità (le funzionalità) con cui è possibile operare sugli oggetti stessi (esempio: la relazione tra un file ed il cestino permette la cancellazione del file stesso). Le relazioni, quindi, permettono agli utenti di interpretare come l'applicazione funziona secondo un loro preciso modello mentale.

**Le metafore:** Sono usate per rappresentare gli oggetti e le loro relazioni (esempio: file, cartella, cestino, telefono, ecc.), permettono di comunicare agli utenti come l'applicazione è stata progettata e, quindi, di familiarizzare con le funzionalità offerte.

**Modello concettuale delle interfacce utente:** E' il modello con il quale decidiamo di rappresentare i modelli mentali degli utenti di un'applicazione. E' molto importante ai fini del disegno delle interfacce utente. Solo capendo i modelli mentali degli utenti sarà possibile definire il modello concettuale da proporre e le relative metafore. Utilizzare metafore familiari ed intuitive aiuta la comprensione dell'applicazione, riduce il tempo di formazione e di supporto richiesto, facilita il suo utilizzo, aumenta la soddisfazione degli utenti e favorisce, in ultima analisi, la sua accettazione da parte degli utenti stessi. Così icone appropriate e relazioni intuitive (esempio: foglio di carta, telefono, cestino, ecc.) facilitano mentre icone inadeguate e relazioni poco intuitive (esempio: un rettangolo, un triangolo, un collegamento indiretto, ecc.) confondono e, spesso, irritano. Un esempio eclatante può essere l'utilizzo di una stessa icona con significati diversi in due applicazioni diverse, oppure la necessità di operare non direttamente tra due oggetti ma solo attraverso un terzo oggetto.

**Paradigmi:** E' importante la scelta dei paradigmi visuali e interattivi (look and feel) per rappresentare il modello e le metafore. Essi dipendono molto dalla psicologia degli utenti. Per quanto riguarda la parte visuale (look), ad esempio, in un ambiente di lavoro serio risulta più accettata una rappresentazione realistica della segretaria (fotografia), mentre in un ambiente informale (giovane) risulta più accettato un disegno "bozzetto" (fumettistico). Anche la scelta dei colori può essere determinante. Per quanto riguarda la parte interattiva (feel), occorre scegliere le associazioni tra

oggetti più intuitive. Ad esempio, “cliccando” su un foglio si attiva il programma di videoscrittura, mentre “cliccando” su magnetofono di attiva la funzione di registrazione. Le metafore e le relative icone proposte devono, quindi, essere “intuitive”, “coerenti” e “piacevoli”.

Concludendo, il modello concettuale delle interfacce utente è la base per la progettazione delle interfacce stesse e delle relative funzionalità. La scelta più o meno felice del modello può condizionare l'intera progettazione ed il successo dell'applicazione. Il modello, infatti, contiene la rappresentazione visuale (paradigmi visuali, estetica), l'interazione con l'utente (modalità di utilizzo delle funzionalità) e le metafore (icone, immagini) del modello concettuale degli utenti (gli oggetti reali e le loro relazioni).

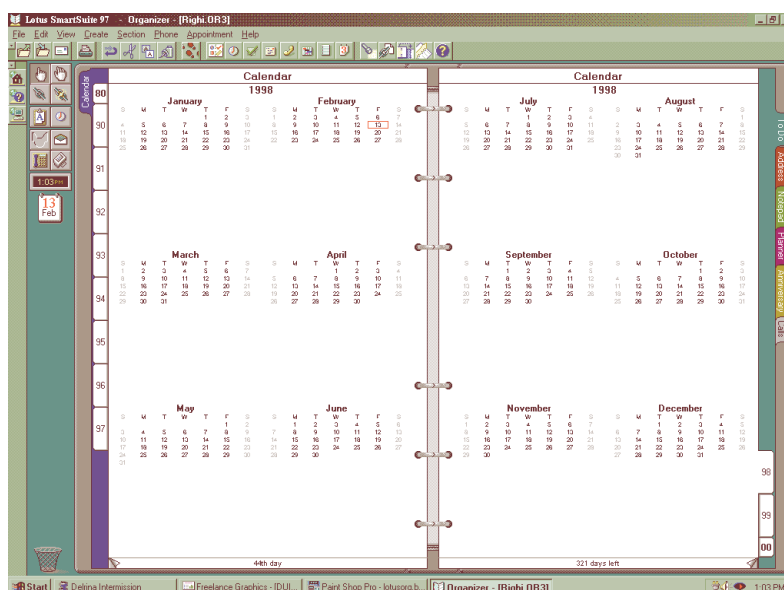


Si intuisce, a questo punto, quanto sia efficace l'utilizzo di prototipi e quanto essi agevolino la valutazione dell'efficacia del modello proposto. Un esempio, in positivo e di enorme successo, di modello concettuale è quello proposto dalle interfacce grafiche dei sistemi operativi dei personal computer (Windows, OS/2, AIX, ecc.) dove la metafora è rappresentata dalla scrivania degli utenti con l'utilizzo di icone parlanti (orologio, documento, cartella, cestino, telefono, ecc.), interazioni intuitive (copia, tagli, incolla, ecc.) e l'utilizzo del mouse al posto della tastiera. Un altro esempio di successo è l'applicazione Organizer che copia il modello concettuale dell'agenda con tanto di agenda giornaliera, rubrica telefonica, ecc.

Praticamente, il modello concettuale contiene le immagini visuali scelte, i testi che descrivono come gli utenti interagiscono con essi per svolgere le proprie attività e la spiegazione di come gli oggetti e le metafore sono state scelte per “mappare” il modello mentale degli utenti.

## Sviluppo di Software Applicativo

---



Segue la descrizione delle attività proposte per definire un modello:

- Raccogliere ed analizzare le informazioni relative a: profilo degli utenti, requisiti di usabilità, casi d'uso (vedi Use Case), soluzioni esistenti.
- Creare i modelli in una sessione di brainstorming permettendo la più libera forma espressiva e la discussione delle diverse soluzioni proposte.
- Scegliere, fra tutte le soluzioni proposte, tre di queste in base a criteri di fattibilità della realizzazione del modello, consistenza con i sistemi esistenti, comparazione con altre soluzioni esistenti, altre considerazioni di disegno.
- Costruire i prototipi a bassa fedeltà (low-fidelity prototyping) delle tre soluzioni scelte.
- Validare i modelli scelti con almeno cinque utenti simulando l'esecuzione dei task principali e verificando se i modelli stessi permettono il completamento delle attività. Dalla validazione si dovrebbe poter scegliere il modello più adatto agli utenti coinvolti. La valutazione riguarderà la capacità di consentire l'esecuzione dei casi d'uso, l'accettazione degli utenti su fattori oggettivi (modalità d'uso) e personali (estetica, altro).
- Redigere un rapporto che spieghi le ragioni delle scelte fatte.

### Modello dei casi d'uso per le interfacce utente

Alla base del modello dei casi d'uso (Use Case) c'è la necessità di comprendere in maniera chiara e completa le attività svolte dagli utenti dell'applicazione, l'ambiente nel quale essi operano, gli eventuali strumenti utilizzati e le modalità con cui le attività stesse sono svolte. Queste informazioni sono preziose per la progettazione (disegno) delle interfacce utente. La tecnica descritta qui di seguito si riferisce alla realizzazione dei casi d'uso che hanno una significativa interazione con gli utenti e, quindi, sono importanti ai fini della soddisfazione ed accettazione dell'applicazione.

La tecnica dei casi d'uso, introdotta inizialmente nello sviluppo ad oggetti per eseguire l'analisi funzionale e la validazione dei requisiti, può essere utilizzata anche nello sviluppo tradizionale (strutturato). In particolare, per la progettazione delle interfacce utente questa tecnica è di grande aiuto. Le attività proposte per realizzare i casi d'uso relativi sono le seguenti:

- Selezionare i casi d'uso, fra tutti quelli definiti per l'applicazione, che hanno un impatto maggiore per gli utenti e raggrupparli per tipo di attore (tipo di utente). L'impatto è considerato in base alla frequenza con cui il task è eseguito ed in base alla sua criticità per il business aziendale. A questo livello i casi d'uso sono identificati e descritti con un codice, un titolo più o meno descrittivo ed il tipo di attore coinvolto.
- Organizzare le sessioni di lavoro dei *focus group* per:
  - Verificare e rivedere la lista ed i casi d'uso in termini di completezza, accuratezza, chiarezza e consistenza delle loro descrizioni.
  - Identificare i requisiti funzionali non indirizzati dagli attuali casi d'uso.
  - Valutare l'impatto dei task descritti dai casi d'uso sulle loro attività in termini di frequenza di esecuzione e criticità per il business aziendale, attribuendo ad ognuno di essi un valore (esempio: da 1 a 5, dove 5 rappresenta la massima frequenza e la massima criticità). Calcolare quindi il valore globale dell'impatto di ogni caso d'uso con un algoritmo adeguato (esempio: somma dei due valori precedenti).
  - Valutare il livello di soddisfazione attuale (valore da 1 a 5, dove 5 rappresenta la massima soddisfazione) nei confronti dei task indirizzati dai casi d'uso in base all'esperienza dell'applicazione attualmente in esercizio.

- Valutare i risultati delle sessioni di lavoro precedenti e risolvere tutte le inconsistenze, gli errori, le mancanze, ecc. riscontrati ed aggiornare i casi d'uso. Per agevolare la valutazione dell'impatto dei casi d'uso si può utilizzare una tabella con i casi d'uso, i valori dati alla frequenza ed alla criticità, ed il valore globale (nell'esempio: la somma dei due valori) come mostrato di seguito.

Casi d'uso		Frequenza	Criticità	Valore globale
Numero	Titolo			
CU 01	Inserimento ordine	4	4	8
CU 02	Lista ordini	1	3	4
CU 03	Stampa ordini	1	3	4
CU 04	Aggiorna ordini	2	4	6

*Tabella. Casi d'uso e valutazione globale*

- Completare la descrizione dei casi d'uso: gli eventi di business (attivazioni) che scatenano il processo analizzato (task); quali attività, in sequenza, l'attore svolge per completare il task; gli input necessari allo svolgimento del task e gli output prodotti; i prerequisiti, se esistono; le condizioni di terminazione; le informazioni aggiuntive relative all'usabilità e quelle utili per meglio capire il caso d'uso. Queste informazioni sono dedotte dalle discussioni con i gruppi di lavoro dove sono presenti gli utenti esperti delle attività considerate. Se l'attività risultasse molto costosa si completino solo i casi d'uso con valore maggiore dell'impatto globale.
- Rivedere i casi d'uso con particolare attenzione alla capacità di garantire il completamento con successo dei task indirizzati. Si richiede agli utenti, a questo punto, di descrivere gli scenari relativi ai casi d'uso. Uno scenario è un caso pratico, reale, completo e significativo del caso d'uso.

Segue un esempio di caso d'uso relativo al task "inserimento di un nuovo ordine".

## Sviluppo di Software Applicativo

Elemento	Descrizione
<b>Codice</b>	CU 01
<b>Titolo</b>	Inserimento nuovo ordine
<b>Area applicativa</b>	Amministrazione
<b>Attori</b>	Agente
<b>Sommario</b>	Inserimento di un nuovo ordine elettronicamente
<b>Impatto</b>	Globale = 8 (Frequenza = 4, Criticità = 4)
<b>Soddisfazione</b>	Attuale = 2 (l'utente inserisce gli ordini manualmente tramite modulo cartaceo)
<b>Flusso normale</b>	<ol style="list-style-type: none"> <li>1. L'utente chiede al sistema di poter inserire un nuovo ordine</li> <li>2. Il sistema presenta la maschera per l'inserimento di un nuovo ordine</li> <li>3. L'utente inserisce i dati dell'ordine e preme INVIO</li> <li>4. Il sistema conferma la correttezza dei dati e richiede la conferma dell'inserimento</li> <li>5. L'utente conferma l'inserimento premendo INVIO</li> <li>6. Il sistema conferma il corretto completamento dell'inserimento dell'ordine</li> <li>7. L'utente termina l'operazione premendo ESC</li> </ol>
<b>Flussi di eccezione</b>	<ol style="list-style-type: none"> <li>4a. Il sistema notifica con un messaggio la presenza di un errore nei dati emessi e ripresenta la maschera con i dati immessi e l'evidenza di quelli errati.</li> <li>4b. L'utente corregge i dati in errore e preme INVIO</li> </ol>
<b>Prerequisiti</b>	Nessuno
<b>Attivazioni</b>	Richiesta inserimento nuovo ordine
<b>Terminazioni</b>	<ul style="list-style-type: none"> <li>• L'utente ha completato con successo l'inserimento del nuovo ordine</li> <li>• L'utente non completa l'inserimento del nuovo ordine perché:</li> <li>• L'utente decide di interrompere l'attività</li> <li>• Il sistema ha rilevato una condizione d'errore</li> </ul>
<b>Input</b>	Dati sul nuovo ordine (agente, cliente, articoli, quantità, modalità di pagamento)
<b>Output</b>	<ul style="list-style-type: none"> <li>• Nuovo ordine inserito nell'archivio degli ordini</li> <li>• Messaggio di conferma inserimento</li> <li>• Messaggio d'errore</li> </ul>
<b>Requisiti indirizzati</b>	Inserimento di un nuovo ordine tramite collegamento al sistema centrale anche remoto da parte degli agenti (REQ-02)
<b>Note di usabilità</b>	L'inserimento di uno o più ordini nuovi avverrà tramite una interfaccia facile ed intuitiva che gli agenti associano, oggi, alla procedura manuale legata ad un modulo di compilazione ed invio via fax

*Esempio di Caso d'uso*

### Guida alla progettazione delle interfacce utente

Questi tipi di linee guida (User Interface Design Guidelines) rappresentano un insieme di standard consistenti per la realizzazione delle interfacce utente. Solitamente esse fanno riferimento a standard accettati dal mercato (esempio: Microsoft Windows User-Interface Guidelines oppure IBM Common User Access Guidelines). Tali standard devono essere integrati con eventuali regolamentazioni e standard specifici dei clienti/enti (esempio: standard ISO). In linea di massima essi si riferiscono alle modalità di presentazione e di inserimento dei dati, tipi di caratteri e loro posizionamento nella tastiera, luminosità e contrasto, ecc.

Le linee guida che definiamo nella nostra azienda per lo sviluppo delle interfacce indicano cosa è obbligatorio e cosa è opzionale, cosa è raccomandato e cosa è da evitare, qual è lo stato dell'arte (best practices) basato principalmente sugli errori più comuni da evitare in azienda ed altrove, ecc. Le linee guida per le interfacce utente si evolvono con l'evoluzione del mercato, delle tecnologie, dell'esperienza acquisita. All'interno di un'azienda, possono esserci più di uno standard (esempio: uno per ogni famiglia di applicazioni).

Obiettivo principale delle linee guida è quello di assicurare la consistenza (look and feel) delle interfacce utente tra le varie applicazioni e garantirne il livello di usabilità.

Nella pratica sono utilizzate nei seguenti momenti del ciclo di vita del software:

- In fase iniziale di disegno delle interfacce utente, sono utilizzate come guida alle decisioni relative alla comprensione dei modelli mentali degli utenti, alla scelta delle metafore da utilizzare, ecc.
- In fase di disegno avanzato, sono utilizzate per la scelta di dettagli relativi alle finestre, i menu, i controlli, i bottoni, i messaggi, ecc.
- In fase di codifica delle interfacce utente, sono utilizzate per la definizione dei dettagli non specificati dal disegno (esempio: utilizzo delle parole mnemoniche per i bottoni, gli elementi dei menu, l'identificazione degli elementi di accelerazione, ecc.)
- In fase di validazione delle interfacce utente, sono utilizzate come elemento di paragone per le verifiche (verifica dell'aderenza agli standard).

Le linee guida possono essere documentati in uno o più documenti (esempio: linee guida valide per tutta l'azienda, valide per un'area applicativa, valide per un progetto). Ai documenti sono spesso associati demo (prototipi con esempi pratici, pagine web con link tra le pagine).

Un tale documento è generalmente composto dai seguenti capitoli:

1. *Aderenza agli standard di industria.* Specifica quali standard esterni devono essere seguiti fornendo dettagli su dove reperirli.
2. *Principi generali sulle interfacce utente.* Fornisce alcuni principi generali applicabili a tutti i tipi di interfacce utente in tutte le piattaforme.
3. *Metodologia di disegno.* Descrive il processo da seguire per la realizzazione delle interfacce utente come la raccolta e l'analisi dei requisiti, il disegno delle interfacce, la realizzazione dei prototipi, la validazione dell'usabilità. Sono anche descritti eventuali tecniche da utilizzare (esempio: revisioni euristiche, test di usabilità), metriche di usabilità (tempo del task, numero di errori).
4. *Linee guida per i componenti.* Descrive la personalizzazione degli standard d'industria per lo specifico progetto. Generalmente è diviso in due parti: la prima descrive quanto è obbligatorio, la seconda quanto è raccomandato, cioè facoltativo ma da seguire a meno di reali impedimenti. Segue un esempio specifico degli elementi descritti.

a. *Utilizzo di componenti chiave:*

- Finestre (windows)
- Menu
- Toolbars
- Controlli (statico del testo, degli input, list box, liste top-down, push button, valore del test, ecc.)

b. *Tecniche di interazione per consentire all'utente di eseguire azioni:*

- Puntatori e cursori
- Tipi di selezione (selezione singola, estesa, multipla)
- Tecniche di selezione (punto di selezione, da-a, random)
- Tecniche di manipolazione diretta ed indiretta (paradigma azione/oggetto oppure oggetto/azione, drag and drop)

c. *Tecniche di interazione per fornire le risposte (feedback) del sistema:*

- Cues (input, suoni, stati selezionati, ecc.)
- Messaggi, Indicatori, Aree informative

## Tabella Oggetti/Azioni

La tabella descrive gli oggetti principali delle interfacce utente, le azioni che gli utenti eseguono su questi oggetti e quali attori (utenti) eseguono tali operazioni. Per esempio sull'oggetto "ordine" si possono eseguire le operazioni "inserimento, cancellazione, modifica, stampa, ecc.". L'Agente è l'utente che esegue le operazioni sopra elencate. La tabella riporta solo gli oggetti più importanti dal punto di vista del business (esempio: una fattura, un ordine di pagamento, ecc.) indipendentemente dal numero delle operazioni che su di esso si possono eseguire.

Scopo di questa tabella è quello di evidenziare gli elementi principali (oggetti, azioni, attori) da utilizzare come base per la creazione del "modello concettuale" da cui partire per la progettazione delle interfacce utente.

Segue un esempio di una tabella di questo tipo.

Funzione	Ordine	Fattura	Prodotto	Cliente
Inserimento	Agente		Ufficio Acquisiti	Agente
Cancellazione	Agente	Amministrazione	Ufficio Acquisiti	Agente
Modifica	Agente	Amministrazione	Ufficio Acquisiti	Agente
Stampa	Agente Amministrazione	Amministrazione	Ufficio Acquisiti Amministrazione	Agente Amministrazione
Invio		Amministrazione		
Creazione		Amministrazione		

*Tabella. Oggetti-Azioni*

La tabella è costruita analizzando, a seconda della tecnica di analisi e disegno adottata (ad oggetti o strutturata), i casi d'uso (*Use Case*) ed estrapolando su quali oggetti gli attori dei casi d'uso agiscono e con quali operazioni. Le informazioni si possono dedurre anche dal "modello logico dei dati" (*Entity-Relationship Diagram*), dal diagramma delle classi (*Class Diagram*) e dal diagramma delle interazioni (*Interaction Diagram*).

### Validazione dell'usabilità

E' fatta tramite due tecniche distinte:

**Revisione euristica di usabilità.** E' eseguita da un gruppo di specialisti di interfacce utente che, applicando linee guida, valuta l'usabilità raggiunta o che potrà essere raggiunta a seconda del livello di sviluppo (disegno, prototipo, codice finale), individua le aree critiche e suggerisce azioni di miglioramento. Possono essere eseguite più revisioni nel corso del disegno per valutare i miglioramenti conseguiti nel tempo.

**Test di usabilità.** Consiste nella valutazione controllata (sistematica e formale) dell'usabilità raggiunta dalle interfacce utente sviluppate tramite un prototipo o direttamente nell'applicazione finale. La valutazione è fatta da un gruppo di utenti presso un ambiente di test particolarmente attrezzato (sala di test protetta da vetri riflettenti, videocamere, registratori, sala di regia, ecc.), detto laboratorio di usabilità o di ergonomia. Sono rilevate misurazioni quantitative e qualitative. Le attività sono generalmente pianificate e documentate tramite un Piano di test di usabilità.

Praticamente, la validazione dell'usabilità consiste nella verifica dei seguenti elementi:

- Aderenza agli standard definiti (CUA o altri standard),
- Consistenza grafica dei dialoghi,
- Validità della messaggistica,
- Usabilità del disegno dei dialoghi.

Le interfacce utente comprendono sia la parte grafica dei dialoghi (Interfaccia Grafica) sia la documentazione prodotta (Manuali e Help on-line).

Segue una breve descrizione delle verifiche effettuate a fronte di ciascuna delle categorie elencate in precedenza.

#### ***Aderenza agli standard (CUA o altri standard)***

Si tratta di verificare che tutti gli oggetti grafici che compongono i dialoghi dell'interfaccia utente siano aderenti agli standard adottati (IBM CUA o altro standard) e definiti per ognuno di essi.

Alcuni esempi di controllo:

**Help.** Il bottone di help è presente in tutte le finestre definite?

**Mnemonic.** La lettera mnemonico è presente in tutti i comandi ed è aderente agli standard richiesti per essi (ad esempio in OK, File, Menu, Cancel, ecc. sono sottolineate rispettivamente le lettere O, F, M, C, ecc.)?

**Bottoni grafici.** I bottoni grafici che implementano le funzioni standard sono aderenti alle linee guida definite per essi (ad esempio alle funzioni Quit, Cancel, Help, ecc. devono corrispondere i bottoni grafici definiti per essi e non altri inventati allo scopo, ecc.)?

**Uso della tastiera.** La navigazione tramite tastiera, in alternativa o in assenza di mouse, è aderente agli standard richiesti per essi (ad esempio la coppia di tasti ALT e F7 è utilizzata per muovere una finestra nello schermo, ecc.)?

Altro. Altri oggetti grafici possono essere verificati oltre a quelli elencati sopra.

### ***Consistenza grafica dei dialoghi***

Consiste nel verificare che i dialoghi siano stati progettati (disegnati) in modo che la navigazione lungo i rami dell'applicazione sia consistente:

- sia nell'aspetto grafico (look) con cui esse vengono proposte all'utente,
- sia nel comportamento (feel), cioè le modalità con cui si possono utilizzare le singole funzioni.

Per esempio, si verifica che la funzione Print non venga proposta a volte tramite l'utilizzo di un bottone grafico e a volte tramite l'immissione di un comando. Si verifica cioè che la funzione venga proposta sempre con le stesse modalità (o sempre tramite lo stesso bottone grafico o sempre tramite lo stesso comando).

### ***Validità della messaggistica***

Una messaggistica chiara, efficace, esaustiva e coerente è molto importante ai fini dell'usabilità.

Si tratta, quindi, di verificare che la messaggistica proposta per l'applicazione sia:

- a) aderente agli standard definiti per essi (IBM CUA o altro standard scelto),
- b) valida nei suoi contenuti.

**a) Aderenza agli standard.** In termini di aderenza agli standard si verifica che:

- la tipologia dei messaggi sia fra quelle previste; i messaggi possono infatti essere:
  - informativi (*Information*)
  - di attenzione (*Warning*)
  - di errore (*Error*)
  - decisionali (*Dialog*)
- ognuna delle tipologie di messaggi sia aderente allo standard definito per essa,
- le caratteristiche delle varie categorie dei messaggi siano omogenee nella grafica rappresentativa (per esempio, il simbolo ? e il simbolo ! rappresentano rispettivamente messaggi decisionali o informativi),
- la grafica ed il testo siano coerenti con la categoria dei messaggi (per esempio un messaggio di attenzione non può avere un testo terroristico).

**b) Validità dei contenuti.** Il testo dei messaggi è controllato per verificare che esso risulti:

- *Chiaro*, cioè che il messaggio sia comprensibile,
- *Leggibile*, che il messaggio descriva esattamente cosa intende comunicare utilizzando frasi brevi,
- *Pertinente*, che il messaggio sia inserito correttamente nel contesto applicativo nel quale viene emesso,
- *Sintatticamente corretto*, che il messaggio non contenga errori sintattici,
- *Omogeneo* con gli altri messaggi della stessa categoria e con l' applicazione stessa.

### ***Usabilità del disegno dei dialoghi***

Consiste nel verificare che la navigazione lungo i rami dei dialoghi proposti sia:

- *facile*,
- *intuitiva*,

- *autoesplicativa*, cioè non si abbia bisogno di ricorrere alla funzione di Help per esercitare correttamente una funzione applicativa proposta (ciò anche se la funzione di Help è comunque richiesta dagli standard CUA per ogni funzione applicativa proposta),

e proponga:

- *cammini brevi, chiari e sintetici*, cioè proponga una navigazione lungo i dialoghi tramite un numero minimo di pannelli prima di giungere a quello desiderato,
- *pannelli con solo le informazioni indispensabili* per la funzione evitando di sovraccargarli con informazioni non strettamente necessarie.

A volte si possono progettare due tipi distinti di dialoghi per una stessa funzionalità:

1. *Base*, per utenti non esperti e, quindi, più ricchi di informazioni,
2. *Avanzato*, per utenti esperti e, quindi, più snello ed essenziale.

*La verifica dell'usabilità del disegno dei dialoghi può essere fatta solo basandosi sull'esperienza maturata nello svolgimento di queste attività. Mancano, infatti, per questo tipo di verifica, criteri oggettivi e misurabili. Per esempio, la misurabilità della chiarezza dei messaggi è prettamente soggettiva, così come pure le metriche sulla leggibilità dei messaggi o la facilità d'uso di un dialogo (vedi **Metriche di usabilità**).*

## Revisione euristica dell'usabilità

E' eseguita da un gruppo di specialisti di interfacce utente che, applicando linee guida, valuta l'usabilità raggiunta o che potrà essere raggiunta a seconda del livello di sviluppo delle interfacce (disegno, prototipo, codice finale). Individua le aree critiche e suggerisce azioni di miglioramento. Praticamente consiste in una riunione formale e può essere eseguita più volte nel corso della fase di disegno per valutare i miglioramenti conseguiti nell'evoluzione del disegno stesso.

Il numero di esperti da coinvolgere può, ovviamente, variare; la raccomandazione è di non scendere sotto le cinque persone. Da analisi statistiche fatte da Nielsen ed altri, infatti, risulta che, coinvolgendo più di cinque esperti, lo sforzo non si traduce in un proporzionale guadagno in termini di errori rilevati. Nella tabella di sotto sono riportati i dati dello studio in termini di numero di esperti coinvolti e numero percentuale di errori di usabilità rimossi.

Numero esperti	1	3	5	10
% errori rilevati	35%	60%	75%	90%

*Tabella. Errori rilevati in funzione del numero di esperti coinvolti*

Una riunione euristica di usabilità si svolge in quattro passi.

1. *Preparazione.* E' necessario spendere un breve tempo (circa una o due ore al massimo) per spiegare ai valutatori l'ambiente, l'applicazione, gli standard adottati, e gli scenari da svolgere (casi d'uso reali). Gli esperti, dal loro canto, concordano la terminologia da adottare, i criteri di valutazione da utilizzare, le modalità per evidenziare i problemi e per presentare i risultati.
2. *Valutazione.* Gli esperti esercitano gli scenari proposti, rilevano gli eventuali errori, valutano l'usabilità e propongono azioni di miglioramento (correttive). I problemi sono classificati in tre categorie a seconda del loro impatto (negativo) sull'esecuzione dello scenario e della possibile insoddisfazione che possono generare sugli utenti. Di seguito è riportata una possibile classificazione della gravità di un problema di usabilità.
  - *Alta.* E' considerato di gravità alta un problema che impedisce di concludere lo scenario e che potrebbe causare grande insoddisfazione (esempio: difficoltà ad utilizzare le funzioni, dialoghi complessi, navigazione confusa, ecc.),
  - *Media:* E' considerato di gravità media un problema che consente di completare il task ma con grande difficoltà e, comunque, con insoddisfazione dell'utente (esempio: necessità di cambiare maschera per reperire

informazioni essenziali, navigazione confusa, utilizzo non standard dei colori, nomenclatura, ecc.),

- *Lieve*. E' considerato di gravità lieve un problema minore che non impedisce il completamento ma può causare insoddisfazione se considerato insieme ad altri problemi (esempio: aspetti grafici estetici, nomenclatura, ecc.).

Le azioni suggerite a fronte dei problemi riscontrati hanno una priorità di esecuzione.

- *Alta*. Con priorità alta, l'azione correttiva sarà eseguita prima di rilasciare l'applicazione per consentire il completamento del task e per evitare una possibile e rilevante insoddisfazione degli utenti,
  - *Media*. Con priorità media, l'azione correttiva sarà eseguita prima di rilasciare l'applicazione per consentire il completamento del task e per evitare comunque l'insoddisfazione degli utenti,
  - *Bassa*. Con priorità bassa, l'azione correttiva sarà eseguita prima di rilasciare l'applicazione per evitare l'insoddisfazione degli utenti.
3. *Sommario*. Consiste in una breve riunione dei valutatori che discutono i problemi rilevati e concordano le azioni da suggerire.
  4. *Rapporto di usabilità*. La riunione si conclude con la stesura di un rapporto dove sono elencati i problemi di usabilità riscontrati, con le relative gravità assegnate, e le azioni correttive raccomandate per risolverli, con le relative priorità.

Questa metodologia è meno costosa e più rapida di un test di usabilità ma è anche, ovviamente, meno oggettiva in quanto si basa sull'esperienza di alcuni soggetti (esperti di usabilità).

### Test di usabilità

Consiste nell' eseguire una serie di prove di validazione con utenti il più possibile rappresentativi di quelli che saranno gli utenti veri destinatari dell'applicazione in esame. L'applicazione validata sarà quella finale oppure un prototipo.

La caratteristica principale dei test di usabilità è il ripetere gli stessi casi di test ogni giorno con un utente diverso. Quello che cambia in ogni iterazione è l'oggetto del test: le interfacce utente. Infatti, al termine di ogni sessione di test si discutono i problemi rilevati con l'utente di turno, si valuta l'usabilità raggiunta, si concordano le azioni correttive e si eseguono le modifiche. L'utente successivo, quindi, testerà gli stessi scenari ma con interfacce sempre più usabili.

Di seguito sono elencate le attività di dettaglio da eseguire.

**Costituire il gruppo di lavoro.** Consiste nell'identificare i partecipanti alle attività di test e di osservazione.

- *Gruppo di osservazione.* Consiste di esperti di usabilità che seguiranno le prove da dietro le quinte rilevando e registrando gli aspetti critici dell'usabilità. Le stesse persone valuteranno l'usabilità raggiunta al termine delle prove e decideranno le azioni correttive da implementare. Il numero suggerito è non meno di tre.
- *Gruppo di test.* Consiste di un campione significativo e rappresentativo di utenti reali con cui eseguire i test. Il numero suggerito è non meno di sei utenti.

**Preparare i casi di test.** Consiste nella preparazione dei casi di test di usabilità da eseguire. Sono realizzati partendo dai casi d'uso (use case) classificati con un alto valore di impatto con gli utenti (con alta frequenza di esecuzione e alta criticità per il business). Generalmente si considera circa il 20% dei casi d'uso con maggiore impatto sugli utenti e si esercitano per l'80% del tempo di esecuzione dei test. La preparazione include un ambiente di test (hardware, software, applicativo ed operativo) simile a quello operativo finale dell'applicazione ed il materiale a supporto necessario (questionario da utilizzare al termine di ogni test, moduli per la rilevazione dei problemi).

**Eseguire i test.** Consiste nell'esecuzione quotidiana dei casi di test preparati con un utente diverso per volta. In ogni ciclo di test si validano le interfacce utente opportunamente modificate secondo i suggerimenti del test precedente. Il test è ripetuto un numero di volte sufficiente a raggiungere il livello di usabilità richiesto (da ciò dipende la pianificazione del numero di test da eseguire e dal reclutamento di utenti da coinvolgere). Durante l'esecuzione il gruppo di esperti osserva il comportamento dell'utente e dell'applicazione e annota i problemi di usabilità rilevati.

**Riunione conclusiva.** Al termine di ogni test, gli esperti discutono con l'utente di turno i problemi rilevati, le sensazioni provate (soddisfazione o insoddisfazione), i punti di vista, ecc. e commentano le loro osservazioni rilevate durante l'esecuzione. Alla riunione partecipano anche i responsabili dell'applicazione (progettisti e programmatori delle interfacce). Si propongono le possibili soluzioni ai problemi evidenziati (che saranno sviluppate prontamente per consentire il successivo test).

**Sommario.** Consiste nella redazione di un Rapporto di Usabilità con i risultati finali (problemi rilevati e soluzioni proposte).

Questa metodologia, sicuramente più costosa di quella precedente (revisione euristica), risulta essere più efficace ai fini dell'implementazione di un alto livello di usabilità in quanto più oggettiva. Maggiori dettagli possono essere trovati nel libro di J. Rubin elencato nel paragrafo Riferimenti.

## Metriche di usabilità

Le metriche relative all'usabilità sono, forse, le meno oggettive in quanto legate alla percezione degli utenti, alle loro modalità operative personali, ecc. Fra le varie metriche predisposte, le norme ISO 9126 indicano che "... Le metriche di usabilità dovrebbero misurare gli attributi del software relative all'operatività dal punto di vista della sua facilità d'uso e dell'adattabilità agli operatori. Inoltre dovrebbero misurare la facilità di imparare il nuovo sistema e capirne il funzionamento".

Di seguito sono riportate solo alcune delle numerose metriche ISO 9126 relative all'usabilità. Una lista completa la si può ottenere direttamente dalle norme.

Caratteristica di usabilità	Metrica
Facilità di comprensione	<ul style="list-style-type: none"><li>• Disponibilità di una Demo</li><li>• Comprensione/Intuizione di quali input sono richiesti e quali output sono prodotti</li></ul>
Facilità di apprendimento	<ul style="list-style-type: none"><li>• Disponibilità di un Tutorial</li><li>• Facilità di imparare</li></ul>
Facilità d'uso (operatività)	<ul style="list-style-type: none"><li>• Interazione attrattiva, piacevole (in USA interfacce "sexy")</li><li>• Facilità di memorizzazione</li><li>• Interazione guidata</li><li>• Presenza di una guida per inesperti</li><li>• Presenza di una guida per tutti gli utenti</li><li>• .....</li><li>• Assenza di errori</li><li>• Tempo richiesto per completare un'operazione (task)</li><li>• Assenza di confusione dovuta a messaggi chiari</li><li>• .....</li><li>• Capacità di recupero dopo un errore utente</li><li>• Numero di tentativi per completare un'operazione (task)</li><li>• Capacità di recupero</li></ul>

*Tabella. Caratteristiche di usabilità del software (tratto da ISO9126)*

## Riferimenti bibliografici

1. Sito esterno IBM: <http://www.ibm.com/ibm/easy>
2. Mandel, T. (1997)  
*Elements of User Interface Design*.  
New York: Wiley.
3. Redmond-Pyle, D. and Moore, A. (1995).  
*Graphical User Interface Design and Evaluation*,  
Prentice Hall.
4. Rubin, J. (1994).  
*Handbook of Usability Testing – How to Plan, Design, and Conduct Effective Tests*,  
John Wiley & Sons.
5. Barman, D.  
*User Interface Style Guide for IBM Global Industries*
6. IBM,  
*Object Oriented Interface Design - IBM Common User Access (CUA) Guidelines*,  
Que Corporation, 1992, ISBN 1-56529-170-0.
7. Isensee, S. & Rudd, J.  
*The Art of Rapid Prototyping*, 1995,  
(<http://www.ibm.com/ibm/hci/designer/docs/proto.html>)
8. ISO, standard ed informazioni disponibili: (<http://www.iso.ch>) e  
(<http://www.ansi.org>)
9. Donald A. Norman (2000).  
*Il computer invisibile*,  
APOGEO

Il documento descrive le problematiche tipiche del software legate all'usabilità e fornisce alcuni elementi di riflessione sul modo di concepire il disegno e lo sviluppo del software. Quanto descritto in questo documento può risultare elementare per gli esperti di usabilità del software, ma risulta ancora non familiare a molta parte del popolo degli sviluppatori, anche professionisti. Elementi più specifici sull'usabilità delle applicazioni e-business sono forniti in un altro documento più specifico "Usabilità dei siti web" dello stesso autore.



Ercole Colonese è Senior Consultant e svolge la sua attività consulenziale essenzialmente nell'area dello sviluppo applicativo. La sua esperienza nel campo dello sviluppo software è maturata in molti anni di lavoro presso i laboratori internazionali IBM dove ha coperto ruoli tecnici e manageriali. Ha realizzato numerosi sistemi qualità aziendali in IBM e presso grandi clienti, certificati iso9000 e secondo modelli riconosciuti (EFQM, etc.). Ha condotto diversi progetti di reingegnerizzazione dei processi di sviluppo secondo il modello *Capability Maturity Model* del *Software Engineering Institute* (SEI-CMM). Come docente, ha tenuto corsi sulla qualità del software ed i processi di sviluppo presso il Learning Center IBM. Ha pubblicato diversi articoli sulle tematiche dello sviluppo software ed il ruolo consulenziale.