



Riassunti PIU - Riassunto Interazione Uomo - Macchina

Interazione Uomo - Macchina (Università degli Studi di Bari)

Modulo 1

Per **interfaccia utente** intendiamo l'insieme di tutti i componenti di un sistema interattivo (software o hardware) che forniscono all'utente informazioni e comandi per permettergli di effettuare specifici compiti attraverso il sistema.

Con il termine compito o **task** si intende qualsiasi insieme di attività richieste per raggiungere un risultato.

Interaction Desing.

È l'attività di progettazione dell'interazione che avviene tra esseri umani e sistemi meccanici o informatici. Il suo obiettivo è sviluppare prodotti facili da apprendere, efficaci da usare e che riescono a fornire una esperienza piacevole.

Divario digitale.

Il divario digitale separa chi può accedere alle tecnologie da chi non può farlo. Gli anziani o tutti coloro i quali non sono nativi digitali, hanno difficoltà ad avvicinarsi alla tecnologia. Occorre inoltre considerare tutte le persone affette da disabilità, quali ipovisione, cecità, ecc...

Il ruolo dell'interfaccia utente.

Bisogna assicurarsi che tutte le persone svantaggiate non siano escluse per mancanza di alfabetizzazione digitale o accesso a internet. Se la tecnologia pone delle difficoltà, abbiamo due strade:

1. operare sugli utenti, istruendoli e avvicinandoli in ogni modo possibile;
2. modificare la tecnologia, promuovendo una cultura della semplicità, che consideri la facilità d'uso come un prerequisito indispensabile;

La memoria umana.

La memoria umana è divisa in tre parti:

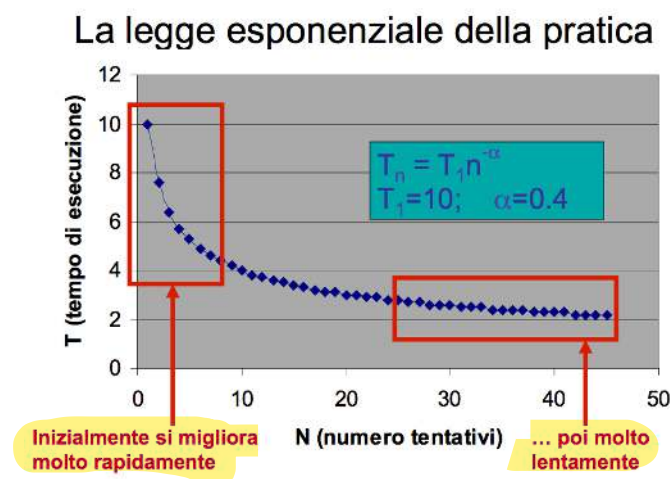
1. Memoria sensoriale;
2. Short-term memory, capace di immagazzinare massimo 7 ± 2 elementi;
3. Long-term memory;

Feedback

In operazioni che richiedono apprendimento motorio, fornire sempre un feedback all'utente.

Legge esponenziale della pratica

Il tempo necessario per effettuare un compito diminuisce con la pratica. In particolare, il tempo T_n per effettuare un compito alla n-esima prova è dato da: $T_n = T_1 n^{-\alpha}$ dove $\alpha = 0.4$ circa.

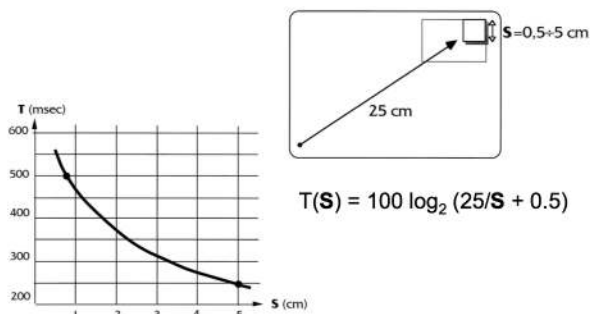


Legge di Fitts

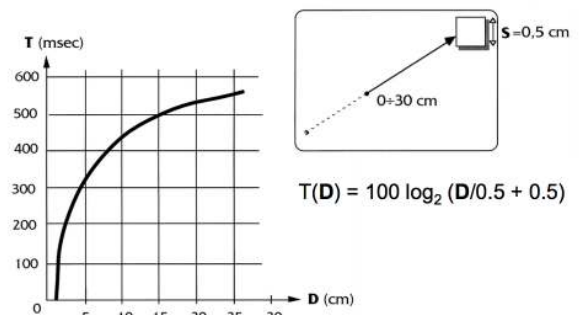
Il tempo T necessario per muovere la mano su un bersaglio di dimensioni S a distanza D , dipende dalla precisione relativa richiesta (rapporto D/S). Più un oggetto è piccolo e lontano, più tempo ci vuole per raggiungerlo.

$$T = 100 \log_2 \left(\frac{D}{S} + 0.5 \right)$$

Esempio



Esempio



Dunque, i bersagli lontani devono essere grandi, i bersagli piccoli devono essere vicini.

Studiare da qui Normann

Modulo 2

Modello di Normann. L'interazione tra l'utente e un sistema viene concettualizzato per la prima volta da Norman. . Suddivise l'operare sugli oggetti in 7 passi principali:

Modello approssimativo che può essere applicato a qualsiasi tipo di azione.

1. Formare lo scopo: decidiamo quale scopo vogliamo raggiungere;
2. Formare l'intenzione: decidiamo che cosa intendiamo fare per raggiungere lo scopo prefissato;
3. Specificare un'azione: pianifichiamo nel dettaglio le azioni specifiche da compiere;
4. Eseguire l'azione: eseguiamo effettivamente le azioni pianificate;
5. Percepire lo stato del mondo: osserviamo come sono cambiati il sistema e il mondo circostante dopo le nostre azioni;
6. Interpretare lo stato del mondo: elaboriamo ciò che abbiamo osservato, per dargli un senso;
7. Valutare il risultato: decidiamo se lo scopo iniziale è stato raggiunto;

ESEMPIO:



Figura 44. La manopola del rubinetto della doccia dell'esempio

- | | |
|-------------------------------------|--|
| 1. Formare lo scopo: | desidero aprire il getto d'acqua per fare la doccia; |
| 2. Formare l'intenzione: | a questo scopo, intendo operare sul rubinetto in figura... |
| 3. Specificare un'azione: | ... ruotandolo con la mano destra verso sinistra, fino in fondo; |
| 4. Eseguire l'azione: | eseguo quanto sopra; |
| 5. Percepire lo stato del mondo: | sento che il rubinetto non può ruotare ulteriormente verso sinistra, e vedo un consistente flusso di acqua uscire dalla doccia; sento che l'acqua è calda; |
| 6. Interpretare lo stato del mondo: | comprendo che il rubinetto è arrivato a fine corsa, e che il flusso dell'acqua calda è conseguenza della mia azione sul rubinetto; |
| 7. Valutare il risultato: | stabilisco che ho raggiunto lo scopo che mi ero prefisso. |

Questo modello, nella sua semplicità, può essere applicato a qualsiasi tipo di azione. Ovviamente, azioni complesse dovranno essere decomposte in azioni sufficientemente semplici, ciascuna delle quali

comporterà il passaggio attraverso i sette stadi. Nel percorrere i sette stadi dell'azione, infatti, è possibile che s'incontrino delle difficoltà nel passare da uno stadio all'altro o, come dice Norman, nell'attraversare i golfi che li separano. In particolare, ci sono due golfi che possono essere particolarmente difficili da superare:

- **il golfo della esecuzione**, che separa lo stadio delle intenzioni da quello delle azioni;
- **il golfo della valutazione**, che separa lo stadio della percezione dello stato del mondo da quello della valutazione dei risultati;

In altre parole, **il golfo dell'esecuzione** è quello che separa le intenzioni dalle azioni che permettono di realizzarle: per superarlo, dovrò identificare, fra le azioni che è possibile eseguire con il sistema, quelle che mi permetteranno di raggiungere lo scopo. Nel caso dell'esempio, non ci sono state difficoltà: il rubinetto è facilmente riconoscibile dal bollino rosso che lo identifica come rubinetto dell'acqua calda, e il suo comportamento è identico a tutti gli altri rubinetti – per aprire l'acqua occorre ruotarlo in senso antiorario. Il golfo dell'esecuzione, perciò, è facile da attraversare. Nel caso, invece, in cui mancasse il bollino rosso, l'utente sarebbe costretto a effettuare varie prove per identificare il rubinetto giusto, e il golfo dell'esecuzione sarebbe, nella metafora di Norman, più difficile da attraversare.

Il golfo della valutazione, invece, è legato alle difficoltà che l'utente deve superare per interpretare lo stato fisico del sistema dopo le azioni effettuate, e comprendere se ha raggiunto o meno lo scopo prefisso.

A questo proposito, nel nostro esempio, si possono verificare varie situazioni. Che cosa pensare se, per esempio, il flusso d'acqua iniziale fosse freddo e restasse tale per parecchi secondi? L'utente non sarebbe in grado di valutare immediatamente se le sue azioni hanno raggiunto lo scopo desiderato, e dovrebbe attendere per un certo periodo - a priori non determinato - con il dubbio che lo scaldabagno sia spento. Il golfo della valutazione sarebbe, in questo caso, più difficile da attraversare. La situazione sarebbe ancora peggiore se il rubinetto dell'acqua fredda e dell'acqua calda non fossero fra loro distinguibili. In questo caso, come tutti noi abbiamo sperimentato almeno una volta, dovremmo procedere per tentativi per identificare il rubinetto giusto, e il semplice compito di aprire il flusso d'acqua calda potrebbe richiedere anche diversi minuti.

Modulo 3

Usabilità

1. Lo standard ISO 9126 definisce l'usabilità come la capacità del software di essere compreso, **appreso, usato e gradito** dall'utente quando usato in determinate condizioni;
2. Mentre per standard ISO 9241 la definizione è: la **efficacia, efficienza e soddisfazione** con cui determinati utenti possono raggiungere determinati obiettivi in determinati ambienti d'uso.
3. Secondo Nielsen: l'usabilità è la misura della qualità dell'esperienza dell'utente in interazione con qualcosa, sia esso un sito web o un'applicazione software tradizionale o qualsiasi altro strumento con il quale l'utente può operare.

In sintesi un prodotto per essere usabile deve:

- essere adeguato ai bisogni e alle aspettative degli specifici utenti finali che lo usano in determinate condizioni;
- risultare facile da capire, da imparare, da usare, ed essere gradevole;
- consentire di eseguire le specifiche attività lavorative in modo corretto, veloce e con soddisfazione.

Affordance.

Con il termine, intraducibile in italiano, di **affordance**, si denota la proprietà di un oggetto di influenzare, attraverso la sua apparenza visiva, il modo in cui viene usato. Un oggetto che possiede una buona **affordance** "invita" chi lo guarda a utilizzarlo nel modo corretto, cioè nel modo per cui è stato concepito.

Feedback.

Per ridurre l'ampiezza del golfo della valutazione gli oggetti dovranno fornire un feedback facilmente interpretabile, cioè un segnale che indichi chiaramente all'utente quali modifiche le sue azioni abbiano prodotto sullo stato del sistema. Il feedback deve essere ben comprensibile e specifico: l'utente deve essere in grado di interpretarlo senza fatica. Importante è la sua tempestività: solo così l'utente lo può porre facilmente in relazione con l'azione cui si riferisce. Se la distanza temporale fra azione e feedback è significativa, essi possono essere interpretati come eventi tra loro indipendenti: a volte bastano pochi secondi di ritardo per disaccoppiare, nella percezione dell'utente, i due eventi.

Conoscere l'utente

Per progettare bene un sistema, dobbiamo conoscere:

- i suoi bisogni e le sue aspettative in relazione al sistema, per realizzare un prodotto che li soddisfi;
- le sue caratteristiche socio-culturali (linguaggio, scolarità, mestiere, esperienza), per fornire un prodotto facile da capire e da imparare;
- i "meccanismi di base" dell'utente (percettivi, motori, cognitivi), per permettere un uso e una iterazione confortevole;

Mettere l'utente al centro del processo di progettazione.

Note

L'usabilità va progettata dall'inizio, e tenuta costantemente sotto controllo nel processo di progettazione e sviluppo. Progettare sistemi usabili è difficile e costoso. Richiede competenze e professionalità specifiche. La valutazione di usabilità di un sistema non può prescindere dall'utente, dunque, bisogna provare. Utenti campione usano il sistema eseguendo compiti tipici in un ambiente controllato, sotto osservazione da parte di esperti di usabilità che raccolgono dati, li analizzano e suggeriscono miglioramenti.

Modulo 4

Progettazione dell'interfaccia

- Analisi dell'utenza potenziale: a quale utente è destinato il nostro prodotto?
- Analisi dei bisogni: quali sono le necessità di tale utente?
- Analisi del contesto: quale sarà il contesto d'uso del prodotto?
- Analisi della concorrenza: perché gli altri prodotti sono inadeguati
- Analisi del compito (task analysis): quali compiti dovrà svolgere con il prodotto?

Metodi applicabili: interviste, questionari, osservazione del loro lavoro, invito a 'pensare a voce alta'.

Tecnica	Serve per	Vantaggi	Svantaggi
Questionari	Rispondere a domande specifiche.	Si possono raggiungere molte persone con poco sforzo.	Vanno progettati con grande accuratezza, in caso contrario le risposte potrebbero risultare poco informative. Il tasso di risposta può essere basso.
Interviste individuali	Esplorare determinati aspetti del problema e determinati punti di vista.	L'intervistatore può controllare il corso dell'intervista, orientandola verso quei temi sui quali l'intervistato è in grado di fornire i contributi più utili.	Richiedono molto tempo. Gli intervistati potrebbero evitare di esprimersi con franchezza su alcuni aspetti delicati.
Focus group	Mettere a fuoco un determinato argomento, sul quale possono esserci diversi punti di vista.	Fanno emergere le aree di consenso e di conflitto. Possono far emergere soluzioni condivise dal gruppo.	La loro conduzione richiede esperienza. Possono emergere figure dominanti che monopolizzano la discussione.
Osservazioni sul campo	Comprendere il contesto delle attività dell'utente.	Permettono di ottenere una consapevolezza sull'uso reale del prodotto che le altre tecniche non danno.	Possono essere difficili da effettuare e richiedere molto tempo e risorse.
Suggerimenti spontanei degli utenti	Individuare specifiche necessità di miglioramento di un prodotto.	Hanno bassi costi di raccolta. Possono essere molto specifici.	Hanno normalmente carattere episodico.
Analisi della concorrenza e delle best practices	Individuare le soluzioni migliori adottate nel settore di interesse.	Evitare di "reinventare la ruota" e ottenere vantaggio competitivo.	L'analisi di solito è costosa (tempo e risorse)

Interviste

La tecnica normalmente più usata è quella delle **interviste individuali** con il committente e i principali stakeholder del prodotto, perché permette di analizzare i singoli problemi in profondità. Gli intervistatori formulano le loro domande in colloqui individuali (faccia a faccia o telefonici) con ciascuno stakeholder, e raccolgono le risposte, annotando esigenze, suggerimenti, desideri e lamentele. Per ottenere la massima sincerità, di solito si garantisce agli intervistati che le loro opinioni verranno riportate solo in forma anonima.

La scelta di chi intervistare va fatta con cura. Occorre prevedere un numero di interviste compatibile con le risorse e il tempo disponibili, ma senza tralasciare nessuna persona che possa avere qualcosa d'importante da dire sul prodotto in progettazione. Dovranno pertanto essere intervistati rappresentanti di ciascuna categoria di stakeholder. Poiché il committente è il referente principale del progetto, le sue indicazioni dovranno avere la massima attenzione. Sarà lui che stabilirà gli obiettivi principali, i tempi di realizzazione e il budget. Sarà lui che indicherà le persone da intervistare e sarà lui che revisionerà e approverà il documento dei requisiti finale. In caso di conflitto fra proposte alternative, sarà lui a decidere quale dovrà essere preferita. Le interviste individuali possono essere più o meno strutturate. Le interviste **non strutturate** sono di carattere esplorativo, e assomigliano a delle conversazioni sugli argomenti d'interesse. L'intervistatore pone domande aperte lasciando all'interlocutore la decisione se

rispondere in modo breve o approfondito. È utile effettuare queste interviste sulla base di un canovaccio preparato in anticipo, in modo da essere sicuri di non tralasciare alcun aspetto rilevante. L'intervistatore potrà comunque orientare il colloquio diversamente da quanto pianificato, per esplorare eventuali aspetti non previsti inizialmente che emergessero nella conversazione.

Le interviste **strutturate** prevedono, invece, un insieme di domande predefinite, come avviene nei questionari di cui tratteremo più oltre. A differenza dei questionari, esse sono comunque realizzate da un intervistatore, in colloqui individuali con gli intervistati. Le interviste strutturate sono utili soprattutto quando gli obiettivi del colloquio siano stati bene identificati, e sia possibile definire un insieme di domande molto specifiche, che richiedono risposte precise. Di solito queste domande sono poste in forma identica a tutti gli intervistati; in questo modo, le risposte possono essere sottoposte ad analisi statistiche. Le interviste semi- strutturate contengono sia domande libere, con carattere esplorativo, sia domande specifiche.

Condurre bene un'intervista può non essere facile e richiede **esperienza**. Occorre evitare di influenzare l'intervistato, formulando le domande in modo che non contengano implicitamente già la risposta. È necessario, inoltre, concentrarsi sui problemi e non sulle soluzioni: si dovrà sempre ricordare che l'obiettivo è quello di identificare i requisiti, e non di effettuare scelte di progetto. Queste dovranno essere fatte in seguito, a fronte di un quadro completo e organico dei vincoli esistenti.

In ogni caso, l'intervistatore dovrà **evitare di usare termini tecnici**, cercando di parlare nel linguaggio dell'intervistato. In molti casi ci si accorgerà ben presto che è necessario chiarire bene il significato di alcuni termini, che possono essere usati dagli intervistati con accezioni particolari. Ogni organizzazione sviluppa col tempo un proprio gergo, che può creare fraintendimenti con interlocutori esterni. Può essere quindi conveniente approfittare delle interviste per definire un sintetico **glossario**. Cioè una lista dei termini più importanti utilizzati nel progetto, con le loro definizioni in relazione allo specifico contesto. Questo glossario, allegato ai requisiti, permette di stabilire una base di conoscenza comune fra gli stakeholder del prodotto e il gruppo di progetto.

Questionari

I **questionari** permettono di raccogliere informazioni in forma strutturata, elaborabili con metodi statistici. Essi possono essere distribuiti ai destinatari in vari modi. Per esempio, si possono predisporre dei questionari compilabili online, generando delle pagine web contenenti le domande del questionario. È così possibile raggiungere una popolazione potenzialmente molto ampia di utenti, anche se, di solito, il tasso di risposta (redemption) è piuttosto basso. Esistono numerosi strumenti software (alcuni anche gratuiti, reperibili in rete), che permettono, da un lato, di costruire facilmente il questionario e, dall'altro, di elaborare i risultati e produrne una visione di sintesi attraverso grafici e diagrammi.

Una tecnica molto usata nei questionari destinati a raccogliere le opinioni degli utenti è la cosiddetta **scala di Likert**. Il questionario è composto da una serie di affermazioni, collegate alle opinioni su cui si vuole indagare, per ciascuna delle quali sono possibili cinque risposte: completamente d'accordo, d'accordo, incerto, in disaccordo, in completo disaccordo. A ciascuna risposta è associato un numero compreso fra 1 e 5. Con questi valori si potrà calcolare la media delle risposte a ciascun gruppo di affermazioni correlate a uno stesso argomento.

Task Analysis

È un passo fondamentale del processo di progettazione dell'interfaccia. L'osservazione dei task svolti da ogni Utente o categoria di Utenti, dell'ordine con in cui vengono eseguiti e dei passi in cui ciascuno di essi può essere decomposto permette di pianificare la dinamica del dialogo e di definire i singoli layout.

Pianificare la dinamica del dialogo:

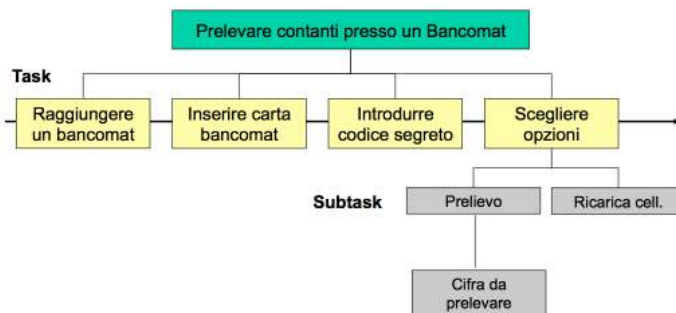
- quali task devono essere resi eseguibili, in ogni fase dell'interazione, a scelta dell'Utente
- quali task devono essere 'bloccati', in ogni fase, per quali categorie di utente

- quale sequenza di comandi elementari permette di realizzare un task complesso
- quando uno stesso task può essere eseguito in modi diversi, e come
- come l'esecuzione di ogni comando elementare modifica il layout dell'interfaccia

Scomposizione dei Task

I compiti che i diversi utenti svolgono possono essere organizzati in una struttura gerarchica, che rappresenta come ciascuno di essi si decompone in diversi compiti più semplici, fino ad arrivare a 'task primitivi', o elementari, non più decomponibili in quanto possono essere eseguiti mediante un solo comando.

Esempio di task analysis: il bancomat



16/10/16

91

Esempio testuale 1

1. Ottenere una tazzina da caffè

- 1.1 Andare al distributore delle bevande
- 1.2 Inserire i soldi
- 1.3 Scegliere il caffè
- 1.4 Aspettare che sia pronto
- 1.5 Rimuovere la tazzina dal distributore

Limiti della Rappresentazione Gerarchica dei Task

Non permette di rappresentare:

- Caratteristiche dei task: i task possono essere connotati come 'frequenti', 'urgenti', o 'infrequenti', ecc: questo influisce sul modo di realizzare l'interfaccia;
- Task comuni: alcuni task possono essere eseguiti in diverse fasi dell'interazione (esempio tipico, il 'salvataggio' di dati, la 'correzione di errori', ecc): i task comuni vanno evidenziati, allo scopo di evitare problemi di 'non consistenza';
- Scomposizioni alternative: un task può essere eseguito in modi diversi.
- Indipendenza d'ordine, concorrenza e interrompibilità: la sequenza di esecuzione dei subtask può non essere rigidamente definita: l'Utente può avere, in alcune fasi del dialogo, la libertà di scegliere quale task eseguire fra diverse alternative; diversi task possono essere concorrentemente attivi; un task può temporaneamente interrompere l'esecuzione di un altro, ecc;

Scenario

È una descrizione, in linguaggio naturale e per grandi linee, di come una applicazione (o un sottoinsieme dei suoi task) verrà utilizzata da uno o più dei suoi utenti potenziali. Permette di discutere il contesto e le modalità secondo cui l'applicazione verrà usata e quindi le esigenze degli utenti. Non descrive invece in modo esplicito il layout dell'applicazione.

Storyboard

È una bozza di prototipo che viene utilizzata, in genere, in combinazione con uno Scenario. Consiste di una serie di schizzi che mostrano come un utente potrebbe eseguire le diverse fasi di ogni task utilizzando l'applicazione. Può consistere in una serie di disegni che mostrano bozze dell'interfaccia nelle varie fasi, sotto forma di disegni oppure, in versione più raffinata, in una serie di schermate realizzate con un tool di sviluppo di interfacce.

Modulo 5 [2seriesdinumeriphppp02,vedere questo pdf e integrare](#)

Prototipi

- Molto utili per prime fasi progetto, per esplorare e valutare diverse soluzioni possibili, a costi contenuti.
- Spesso di tipo “usa e getta”, realizzati anche a costi molto bassi.
- Varie tecnologie usate, di solito molto semplici – Es.: carta, HTML, PowerPoint

Paper Prototype

VANTAGGI:

- Velocità e basso costo di realizzazione
- Permettono di provare interazione in modo semplice
- Basso costo delle modifiche

SVANTAGGI:

- Interazione lenta e quindi innaturale, simulata
- Fedeltà molto bassa per aspetti di layout grafico

Prototipi PowerPoint

VANTAGGI:

- Facili da realizzare e da modificare
- Interattività non richiede mago di Oz
- Aspetti grafici definibili “abbastanza” bene

SVANTAGGI:

- Modello di interazione limitato (“point & click”) - Limiti pratici a complessità ipertesto
- poco gestibile oltre un centinaio di slide

Modulo 6 – Valutazione dell’usabilità

Si possono dividere in due grandi categorie, ossia, valutazioni attraverso l’analisi di esperti di usabilità, oppure, valutazione con la partecipazione degli utenti.

L’aggettivo **euristico**: procedimento non rigoroso che consente di prevedere o rendere plausibile un determinato risultato, che in un secondo tempo dovrà essere controllato e convalidato con metodi rigorosi. Nell’ingegneria dell’usabilità, si chiamano euristiche quelle valutazioni di usabilità effettuate da esperti, analizzando sistematicamente il comportamento di un sistema e verificandone la conformità a specifiche “regole d’oro” (chiamate, appunto, euristiche), derivanti da principi o linee guida generalmente accettati.

Cognitive Walkthrough

Il **Cognitive Walkthrough** è un metodo di ispezione per la valutazione euristica.

Il valutatore esamina in dettaglio (passo dopo passo) le azioni elementari che l'utente del sistema deve compiere per portare a buon fine ogni specifico compito, cercando anche di ricostruire i processi mentali richiesti agli utenti nello svolgimento delle varie azioni, per individuare eventuali problemi di usabilità. Richiede molto tempo ed esperienza. Utile soprattutto per valutare la apprendibilità.

Come si fa:

Per ogni azione nella sequenza di azioni che l'utente deve effettuare per realizzare un compito, il valutatore cerca di rispondere alle seguenti 4 domande:

1. L'utente cercherà di ottenere l'effetto corretto?
2. L'utente noterà che l'azione corretta è disponibile?
3. L'utente assocerà l'azione corretta con l'effetto che sta cercando di ottenere?
4. Se l'azione corretta è stata eseguita, l'utente vedrà che c'è un progresso verso la soluzione del compito?

Valutazioni basate su euristiche Fatta dal riassunto di antonio nel capitolo principi e linee guida, in caso di domanda rispondere scrivendo quella sezione

Il sistema viene esaminato verificandone la conformità a specifiche "regole d'oro" (dette, appunto, "euristiche"), derivanti da principi e linee guida generalmente accettati. Le euristiche possono essere diverse, più o meno dettagliate. Si preferiscono euristiche costituite da pochi principi molto generali, piuttosto che linee guida dettagliate, di difficile utilizzo.



1. **Principi:** Principi generali per la progettazione di interfacce utente usabili, basati su evidenza scientifica o sul generale consenso. Derivano dalla conoscenza degli aspetti psicologici, computazionali e sociali e sono indipendenti dalla tecnologia. Sono espressi spesso in forma molto generale.
2. **Linee guida:** Insieme di raccomandazioni per il progetto dell'interfaccia utente per una particolare classe di sistemi, espresse in modo generale ma meno astratte degli standard, con esempi e motivazioni. Non sono vincolanti, sta al progettista decidere sulla opportunità di applicarle caso per caso.
3. **Standard:** Insieme di regole da applicare nel progetto di una classe di sistemi. Sono vincolanti: i progettisti devono applicarli. Sono di norma emesse da un Ente di standardizzazione. La conformità allo standard deve essere valutabile in modo preciso.
4. **Regole di progetto:** Insieme di regole da applicare nel progetto di un particolare sistema. Sono vincolanti.

Standard

L'ente principale responsabile della preparazione degli standard è l'**ISO**, un'associazione non governativa di enti nazionali di standardizzazione di oltre 160 paesi. I prodotti principali dell'ISO sono dei documenti chiamati "International Standard" (IS), che vengono redatti da appositi comitati tecnici (Technical Committee, TC) rappresentativi degli interessi delle parti coinvolte (produttori, venditori, utenti, organizzazioni dei consumatori, laboratori di prova, governi, professionisti, organizzazioni di ricerca, ecc.)

Principi del dialogo secondo ISO 9241

1. Adeguatezza al compito:

Un sistema interattivo è adeguato al compito se supporta l'utente nel completamento del compito, cioè quando la funzionalità del sistema e il dialogo sono basati sulle caratteristiche del compito, piuttosto che sulla tecnologia scelta per effettuarlo.

2. Autodescrizione:

Un dialogo è auto-descrittivo se agli utenti risulta evidente, in ogni momento, in che dialogo si trovano, a che punto si trovano all'interno del dialogo, quali azioni possono compiere e come queste possono essere effettuate.

3. Conformità alle aspettative dell'utente:

Un dialogo è conforme alle aspettative dell'utente se corrisponde alle necessità dell'utente, prevedibili in base al contesto e a convenzioni comunemente accettate.

4. Adeguatezza all'apprendimento:

Un dialogo è adeguato all'apprendimento se supporta e guida l'utente nell'apprendimento del sistema.

5. Controllabilità:

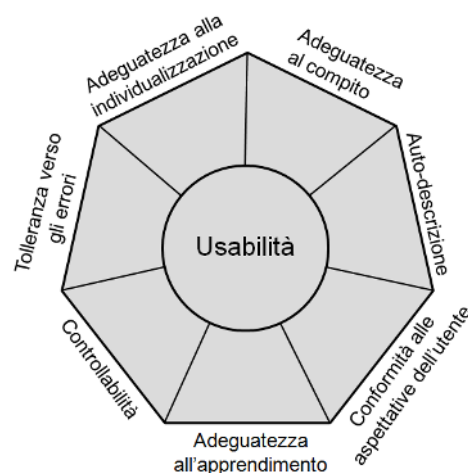
Un dialogo è controllabile se l'utente è in grado di iniziare e tenere sotto controllo la direzione e i tempi dell'interazione fino al raggiungimento dell'obiettivo.

6. Tolleranza verso gli errori:

Un dialogo tollera gli errori se, nonostante evidenti errori negli input, i risultati desiderati possono essere ottenuti senza o con minime azioni correttive.

7. Adeguatezza alla individualizzazione:

Un dialogo è adeguato alla individualizzazione se l'utente può modificare l'interazione e la presentazione dell'informazione per adattarle alle proprie necessità e capacità individuali.



Si preferisce utilizzare euristiche costituite da pochi linee guida e principi generali. Fra queste quelle di Nielsen, costituite da 10 regole che permettono di inquadrare i problemi rilevanti in categorie ben individuate:

Le dieci euristiche di Nielsen

1. **Visibilità dello stato del sistema:** Il sistema dovrebbe sempre informare gli utenti su ciò che sta accadendo, mediante feedback appropriati in un tempo ragionevole.
2. **Corrispondenza fra il mondo reale e il sistema:** Il sistema dovrebbe parlare il linguaggio dell'utente, con parole, frasi e concetti familiari all'utente, piuttosto che termini orientati al sistema. Seguire le convenzioni del mondo reale, facendo apparire le informazioni secondo un ordine logico e naturale.
3. **Libertà e controllo da parte degli utenti:** Gli utenti spesso selezionano delle funzioni del sistema per errore e hanno bisogno di una "uscita di emergenza" segnalata con chiarezza per uscire da uno stato non desiderato senza dover passare attraverso un lungo dialogo. Fornire all'utente le funzioni di undo e redo.
annullo ripristino
4. **Consistenza e standard:** Gli utenti non dovrebbero aver bisogno di chiedersi se parole, situazioni o azioni differenti hanno lo stesso significato. Seguire le convenzioni della piattaforma di calcolo utilizzata.
5. **Prevenzione degli errori:** Ancora meglio di buoni messaggi di errore è un'attenta progettazione che eviti innanzitutto l'insorgere del problema. Eliminare le situazioni che possono provocare errori da parte dell'utente, e chiedergli conferma prima di eseguire le azioni richieste.
6. **Riconoscere piuttosto che ricordare:** Minimizzare il ricorso alla memoria dell'utente, rendendo visibili gli oggetti, le azioni e le opzioni. L'utente non dovrebbe aver bisogno di ricordare delle informazioni, nel passare da una fase del dialogo a un'altra. Le istruzioni per l'uso del sistema dovrebbero essere visibili o facilmente recuperabili quando servono.
7. **Flessibilità ed efficienza d'uso:** Acceleratori – invisibili all'utente novizio – possono spesso rendere veloce l'interazione dell'utente esperto, in modo che il sistema possa soddisfare sia l'utente esperto sia quello inesperto. Permettere all'utente di personalizzare le azioni frequenti.
8. **Design minimalista ed estetico:** I dialoghi non dovrebbero contenere informazioni irrilevanti o necessarie solo di rado. Ogni informazione aggiuntiva in un dialogo compete con le unità di informazione rilevanti e diminuisce la loro visibilità relativa.
9. **Aiutare gli utenti a riconoscere gli errori, diagnosticarli e correggerli:** I messaggi di errore dovrebbero essere espressi in linguaggio semplice (senza codici), indicare il problema con precisione e suggerire una soluzione in modo costruttivo.
10. **Guida e documentazione:** Anche se è preferibile che il sistema sia utilizzabile senza documentazione, può essere necessario fornire aiuto e documentazione. Ogni tale informazione dovrebbe essere facilmente raggiungibile, focalizzata sul compito dell'utente, e dovrebbe elencare i passi concreti da fare, senza essere troppo ampia.

Test di usabilità

Un test di usabilità consiste nel far eseguire a un gruppo di utenti dei compiti tipici di utilizzo del sistema in un ambiente controllato. Si sceglie un campione di utenti che sia rappresentativo della categoria di utenti cui il sistema si rivolge, e si chiede a tutti di svolgere, separatamente, gli stessi compiti. Chi conduce il test osserva e analizza il loro comportamento per comprendere se, dove e perché essi hanno incontrato delle difficoltà.

Il test coinvolge, oltre all'utente che prova il sistema, almeno due altre persone: un **facilitatore**, che ha il compito di gestire la "regia" della prova, e uno o più **osservatori**, che assistono al test, annotando i comportamenti dell'utente che ritengono significativi.

Un test di usabilità ha lo scopo di ricavare indicazioni concrete per il miglioramento del sistema. Chi lo conduce dovrà esaminare in dettaglio le operazioni svolte dagli utenti per capire dove nascono le difficoltà, da che cosa sono causate e in quale modo possano essere rimosse. Per questo, è molto utile la cosiddetta tecnica del **pensare ad alta voce** (think aloud), che consiste nel chiedere all'utente di esprimere a voce alta ciò che pensa mentre compie le varie operazioni. Questo è molto utile, perché permette agli osservatori di raccogliere informazioni sulle strategie messe in atto dall'utente nell'esecuzione dei compiti, e sulle difficoltà che egli incontra durante il test.

L'analisi del comportamento degli utenti non può essere condotta in tempo reale durante lo svolgimento del test, ma deve essere compiuta dopo, con la necessaria tranquillità. Anche se l'utente esprime ad alta voce, durante la prova, le difficoltà che sperimenta, le cause di queste difficoltà possono non essere evidenti. Per identificarle, e comprendere come possano essere rimosse, sarà necessario **riesaminare** con attenzione la sequenza di azioni eseguite dall'utente. Durante il test, l'osservatore dovrà quindi prendere appunti, registrando le situazioni in cui l'utente manifesta incertezza o commette degli errori. Questi appunti saranno riesaminati in seguito, per individuare le cause del problema e studiare le correzioni più opportune. Se possibile, è preferibile eseguire una **registrazione** (audio e video) della sessione di test, per rivedere in seguito tutto ciò che è avvenuto durante la prova.

La tecnica più comune consiste nel **riprendere con una telecamera il viso** dell'utente mentre esegue il test, registrando contemporaneamente le sue parole e, con un'altra telecamera, il sistema. Le due registrazioni dovranno poi essere viste in modo sincronizzato: spesso le espressioni del viso dell'utente sono altrettanto rivelatrici delle sue parole e delle sue azioni.

Nel caso di sistemi eseguiti al computer, tutto questo si può fare, molto semplicemente, con una sola webcam che riprenda il viso dell'utente, un microfono e un programma in esecuzione sullo stesso computer usato per il test, che registri le immagini che appaiono sul video, in modo sincronizzato con le registrazioni audio e video. Tali programmi (ne esistono anche di gratuiti) permettono poi di esaminare in dettaglio, per così dire alla moviola, le azioni effettuate dall'utente e metterle in corrispondenza con le espressioni del viso e le parole pronunciate.

Tipi di test

Test formativi: utilizzati durante il ciclo iterativo di progettazione, per sottoporre i vari prototipi a prove d'uso con gli utenti, allo scopo di identificarne i difetti e migliorarne l'usabilità. Si chiamano formativi perché, appunto, contribuiscono a "dare forma" al prodotto: il loro scopo è individuare il maggior numero possibile di problemi (utili nelle fasi iniziali della progettazione- quick & dirty). Questo perché nelle fasi iniziali del progetto i test mettono in luce rapidamente i difetti macroscopici, che richiedono una parziale (o totale) riprogettazione dell'interfaccia. Quindi si prova in fretta, si modifica rapidamente il prototipo eliminando i difetti più evidenti, e si prova ancora, e così via. Per il primo test di un prototipo iniziale di carta, 2-3 utenti sono in genere sufficienti.

Test sommativi: dal verbo "sommare", indicano una valutazione più complessiva del prodotto, al di fuori – o al termine – del processo di progettazione e sviluppo. Sono test più completi di quelli formativi, che non

hanno lo scopo di fornire indicazioni ai progettisti, ma di valutare in modo sistematico pregi e difetti del prodotto, o sue particolari caratteristiche. Sono di solito condotti quando il sistema è completamente funzionante, per esempio per indicarne i punti deboli e valutare l'opportunità di un redesign migliorativo. Oppure per confrontarne le caratteristiche con quelle di sistemi concorrenti. Un test di tipo sommativo coinvolge di solito un numero maggiore di utenti, per esempio 10-15, o anche di più.