



**Università
degli Studi
di Cagliari**

Interazione **U**omo **M**acchina

Autore:
Simone Giuffrida

Revisori:
Matteo Dessì
Leonardo Dessì
Erica Corda

Contents

1	Introduzione alla disciplina	4
2	Storia dell'Interazione Uomo Macchina	4
3	Concetti base	4
3.1	Rappresentazione virtuale delle immagini	5
4	La componente umana	5
4.1	La percezione visiva	6
4.2	Comprensione delle immagini	7
4.2.1	Interpretazione umana di un'immagine	8
4.3	Visione stereoscopica e interpretazione del segnale	9
4.4	Il processo di lettura	10
4.4.1	I font e la loro importanza	11
4.5	Compendio di tipografia digitale	11
4.6	Principi per il raggruppamento	12
4.6.1	F-shape	12
4.7	L'udito	13
4.7.1	Anatomia dell'orecchio	13
4.7.2	Il Suono	14
4.8	Tatto	14
4.9	Movimento e tempo di reazione	14
4.10	Memoria	17
4.10.1	Memoria sensoriale	17
4.10.2	Memoria a breve termine (STM)	18
4.10.3	Memoria a lungo termine (LTM)	19
4.11	Pensiero e ragionamento	21
4.12	Problem solving	21
4.13	Analogia	22
5	Interazione	23
5.1	Il modello di Norman	23
5.2	Errori	24
5.3	Il modello di Abowd e Beale	25
5.4	I principi di usabilità	25
5.4.1	Come si impara ad utilizzare un sistema, l'apprendibilità	26
5.4.2	Modelli concettuali	26
5.4.3	La consistenza	28
5.4.4	Efficienza	29
5.4.5	Sicurezza (safety)	29
5.5	WIMP	30
5.5.1	Le finestre	31
5.5.2	Menu	32
5.5.3	Le icone	33
5.5.4	Widget	33
5.6	Altri stili	34
5.6.1	Interfaccia a linea di comando	34
5.6.2	Interfacce conversazionali	35
5.6.3	Interfacce ad interazione vocale	35
5.6.4	Interfacce tramite realtà virtuale	36

6	Il processo di design dell'interazione	37
6.1	Stabilire i requisiti	38
6.2	Le caratteristiche dell'utente	39
6.2.1	Le personas	39
6.3	Parte laboratorio	39
6.3.1	I requisiti	39
6.3.2	Le personas	40
6.3.3	I scenari	40
6.4	Raccogliere i dati per i requisiti	41
6.5	Analisi dei dati	41
6.6	Creazione di prototipi	41
6.6.1	Prototipo a bassa fedeltà	43
6.6.2	Prototipi ad alta fedeltà	43
6.7	Le 10 euristiche per l'usabilità	44
6.7.1	Visibilità dello stato del sistema	44
6.7.2	Corrispondenza fra sistema e mondo reale	44
6.7.3	Libertà e controllo per l'utente	44
6.7.4	Consistenza e standard	45
6.7.5	Prevenzione degli errori	45
6.7.6	Riconoscimento piuttosto che memoria	45
6.7.7	Flessibilità ed efficienza di utilizzo	46
6.7.8	Estetica e design minimale	46
6.7.9	Aiutare l'utente a riconoscere ed a rimediare agli errori	47
6.7.10	Documentazione e aiuto	47
7	Valutazione	47
7.0.1	Metodi basati su feedback	49
7.0.2	Misure quantitative e informazioni qualitative	50
7.1	Questionari	51
7.1.1	La creazione di un questionario	51
7.1.2	Diversi tipi di domanda	51
7.1.3	Diversi tipi di questionari	54
7.2	Questionario SUS	54
7.3	Questionario SMEQ	56
7.4	Valutazione remota: A/B testing	56
7.5	Between e within subjects	56

1 Introduzione alla disciplina

IUM sta per Human Computer Interaction, è una disciplina che studia la progettazione, valutazione e implementazione di sistemi computazionali interattivi destinati all'utilizzo umano ed i maggiori fenomeni attorno ad essi; lo scopo principale di questa disciplina è di garantire che le tecnologie siano utilizzabili dalle persone, questo però richiede molte competenze informatiche, psicologiche, di ergonomia, di ingegneria e di design, di conseguenza è molto difficile ottenere tutte queste conoscenze da una sola figura professionale.

2 Storia dell'Interazione Uomo Macchina

All'inizio della storia degli elaboratori, essi erano utilizzabili solo tramite riga di comando(i.e. Unix) e gestiti in modo sincrono, utilizzare questi dispositivi richiedeva essere esperti nel settore, in quanto gli elaboratori non erano acquistabili, ma dovevano essere costruiti appositamente.

Si iniziò a parlare di sistemi grafici soltanto nel 1962, quando Ivan Sutherland, creò Sketchpad, una sorta di "penna" che aveva la capacità di scrivere all'interno di uno schermo. Per arrivare ai primi sistemi a finestra si dovette aspettare circa 20 anni. Negli anni 80 infatti si ebbe un boom nell'ambito dell'interazione con i computer, nel 1980 Richard Bolt presentò per primo un paradigma di interazione multimodale(ovvero che utilizza più sensi umani), consisteva in un grande monitor, dove l'utente doveva indicare con la mano un oggetto all'interno del monitor e dire ad alta voce l'azione che si voleva svolgere con quel determinato oggetto.([CLICCA QUI PER IL VIDEO](#)). La grande novità arrivò nel 1981, quando arrivarono i primi sistemi a finestra dove poté essere utilizzato per la prima volta il mouse per interagire con oggetti presenti nello schermo, il primo fu il Xerox Star, sviluppato in circa 10 anni, a seguire arrivarono anche Apple Lisa(1982), Macintosh(1984), Windows(1985), in questo stesso periodo vennero anche progettati i primi portatili. Un'altro grande cambiamento arrivò nel 1990, quando Tim Berners-Lee inventò il web, e nel 1991 vennero inventati i primi palmari. Il primo dispositivo touch a riscuotere successo fu l'iPhone nel 2007, anche grazie alla sua capacità di poter visualizzare le pagine web indipendentemente se fossero progettate per essere viste da cellulare, infatti prima di questa invenzione, le pagine web dovevano essere programmate sia per essere visualizzate su computer, che su cellulare, tramite 2 linguaggi diversi. Attualmente la ricerca prosegue, infatti annualmente si hanno diverse conferenze mondiali (la conferenza della SIGCHI, e INTERACT).

3 Concetti base

Le interfacce sono il mezzo che si pone tra noi e la rappresentazione interna dei dati del computer, funziona come un linguaggio intermedio fra noi e il linguaggio comprensibile al nostro elaboratore. Le informazioni si ottengono tramite periferiche d'ingresso, e i risultati tramite periferiche d'uscita, questa metodologia è chiamata **Trasduzione di segnali**, questa avviene dal mondo analogico al digitale e viceversa. I device di interazione con lo schermo possono essere

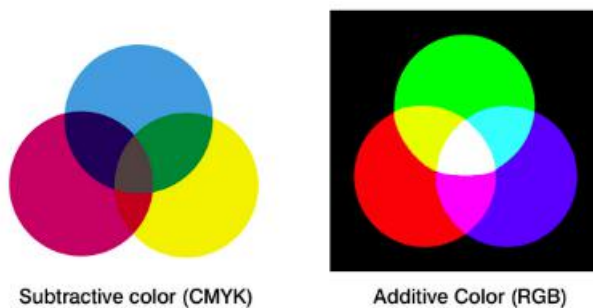
- **device di manipolazione diretti**, come per esempio il touchscreen, infatti quando andiamo ad interagire tramite uno schermo touch, noi interagiamo direttamente nello schermo.
- **device di manipolazione indiretti**, come il mouse, i movimenti di un mouse muovono il cursore sullo schermo, questo cursore è orientato sul piano dello schermo(x,y) mentre noi lo utilizziamo su un piano di coordinate(x,z); per questo motivo il movimento che noi effettuiamo per muoverci col cursore all'interno dello schermo non è ampio come tutto lo schermo, ma è scalato(lo spostamento del mouse di 1cm non sposta anche il cursore di 1cm).

3.1 Rappresentazione virtuale delle immagini

Le immagini in uno schermo possono essere rappresentate in 2 modi:

- **Vettoriale:** ovvero quando l'immagine è costituita da figure geometriche primitive(punti, rette, rettangoli, ecc..), il vantaggio di questa rappresentazione è che l'immagine può essere zoommata all'infinito senza perdere di qualità.
- **Bitmap / raster:** l'immagine non è costruita tramite la logica di figure geometriche, ma è una matrice la quale è riempita di pixel, ogni elemento della matrice è chiamato pixel e contiene rappresenta il colore contenuto in quella cella; a differenza delle immagini vettoriali. Quando vengono zoommate queste immagini perdono di qualità.

Il colore delle immagini invece può essere rappresentato tramite spazi di colore addittivi(RGB, HSV) o sottrattivi(CMYK). Cambia il modo in cui viene generato il colore, con il metodo addittivo si parte dal nero, ovvero l'assenza di colore, e si aggiungono man mano le tonalità dei 3 colori primari(rosso, giallo, blu); mentre con i colori sottrattivi è l'opposto, ovvero si parte dal bianco, ovvero l'unione di tutti i colori



I display attualmente sono utilizzati con un'aspect ratio di 16:9, la risoluzione di uno schermo indica la grandezza della matrice di pixel, la densità invece è il numero di pixel per "unità" di superficie, esiste anche l'angolo di visuale, ovvero l'angolo massimo di visuale, oltre il quale i colori risultano alterati(Avvicinandosi troppo ad uno schermo, si riduce l'angolo di visuale, e quindi si iniziano anche a vedere i diversi pixel che compongono un'immagine).

4 La componente umana

In questa parte si cercherà di capire il funzionamento dell'essere umano in modo da poter fornire dei modelli di ragionamento utilizzati dagli umani, per poterli sfruttare durante il design di interfacce. L'uomo riceve informazioni tramite canali di ingresso e di uscita(visuale, uditiva, tattile, movimento). Come un'elaboratore, anche noi umani possediamo la memoria, questa si divide in tre tipologie

- Sensoriale
- A breve termine
- A lungo termine

Una volta ricevute, queste informazioni vengono trattate ed utilizzate in base al loro scopo.

4.1 La percezione visiva

La retina e il cervello sono i componenti su cui si fonda il sistema visivo umano, il sensore (la retina) passa delle informazioni all'elaboratore delle informazioni(il cervello).

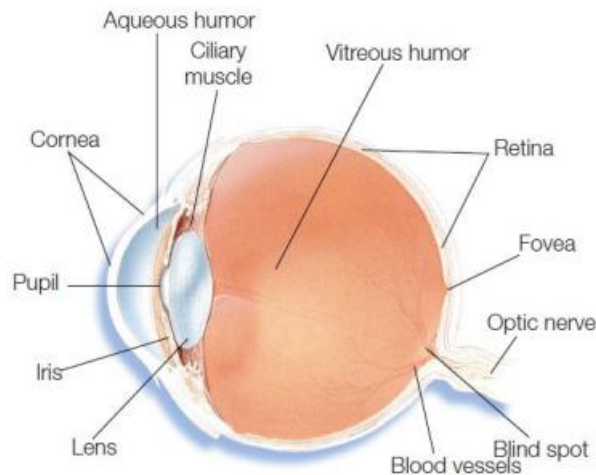


Figure 1: Anatomia dell'occhio

La percezione di un oggetto dipende dalle caratteristiche dell'oggetto stesso, dalla luce, che lo illumina, dal colore dello sfondo, dalla personeache lo guarda, e da tanti altri fattori. L'occhio consente di convertire la luce raccolta in segnali elettrici da trasmettere al cervello, il sensore è costituito da due tipi di cellule:

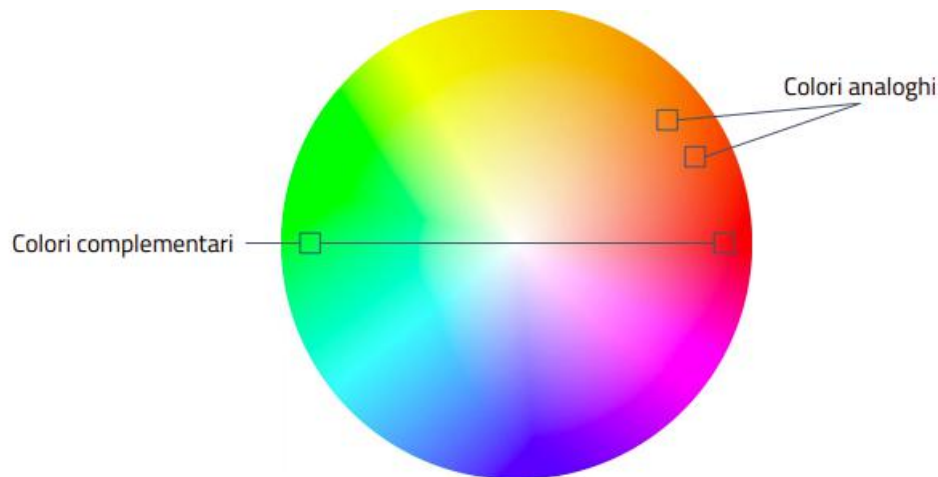
- **I coni.** Sono sensibili alla lunghezza d'onda(**percezione del colore**), fungono da contenitori di composti chimici, detti fotopigmenti la cui attivazione dipende dalla frequenza della luce che li colpisce, i coni possono essere sensibili a diversi colori
 - Blu(onde corte, sono il tipo di coni meno presenti)
 - Verde(onde medie, la cardinalità è circa doppia rispetto a quelli blu)
 - Rosso(onde lunghe, 4 volte più presenti rispetto a quelli blu)

Per questo motivo **si utilizza il rosso per attirare l'attenzione e il blu per gli sfondi.**

- **I bastoncelli.** Sono dei fotorecettori sensibili alla variazione di luminosità che si trovano nel bordo esterno della retina, garantiscono all'umano di poter vedere anche al buio, **la visualizzazione dei movimenti e la visione periferica.**

4.2 Comprensione delle immagini

Il processo di visione umano si basa sulla distinzione di tre colori opposti: Rosso-Verde, Blu-Giallo, Nero-Bianco. La scelta dei colori va effettuata con cautela perché devono essere rispettati dei vincoli.



I colori adiacenti tra loro si dicono **analoghi** (sono utili per la creazione di schemi di colore), mentre quelli opposti si dicono **complementari**, e vanno accostati raramente e con cautela, perché si rischia di incappare nel fenomeno della **color pollution**. Un'altra regola da rispettare è l'utilizzare il rosso, il verde e il blu come sfondo e al contempo come colore di una scritta in quanto avremmo il problema della **cromostereopsi**

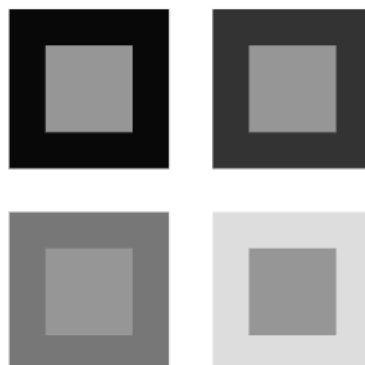


Figure 2: Esempio di color pollution



Figure 3: cromostereopsi

Accostare determinati colori creando un contrasto è uno strumento utilissimo di veicolazione dell'attenzione, infatti uno stesso colore viene percepito diversamente a seconda del colore dello sfondo sul quale è posizionato



4.2.1 Interpretazione umana di un'immagine

Colui che si occupa della comprensione di un'immagine è il cervello umano, il quale individua elementi caratteristici (pattern), inoltre tramite la sola vista si ha la capacità di percepire la dimensione e la profondità di un oggetto, questo fenomeno è detto **acutezza visiva**, ovvero la capacità di percepire i dettagli di tutto ciò che ci circonda. tutte queste informazioni si ottengono attraverso diversi indizi:

Occlusione. Quando un oggetto copre parzialmente un altro capiamo che il primo è più vicino

Altezza relativa. Generalmente gli oggetti che sono più alti nel campo visivo sono più distanti

Grandezza relativa. Quando gli oggetti hanno la stessa dimensione, quello più vicino risulta più grande

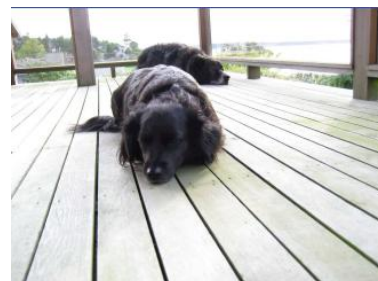
Convergenza prospettica. Le linee parallele convergono in un punto di fuga, basti pensare alle rotaie di un treno, che arrivando fino alla linea dell'orizzonte sembra che si tocchino

Familiarità della dimensione. Informazione sulla distanza data dalla conoscenza della dimensione reale dell'oggetto

Percezione dell'atmosfera. Vediamo meno chiaramente gli oggetti distanti che assumono una tinta bluastra

Texture. Oggetti spazati in modo regolare appaiono più vicini man mano che la distanza aumenta

Parallasse. Oggetti vicini escono subito dal campo visivo, quelli lontani invece sembrano muoversi più lentamente



In tutte le precedenti immagini è possibile accorgersi di più indizi che agiscono sull'immagine, per esempio guardando la quarta immagine, il nostro cervello utilizza il meccanismo delle texture per capire la prospettiva, infatti nell'erba vicina ci accorgiamo dei grandi spazi di sabbia tra le zolle di prato, mentre con l'allontanarsi del terreno assume tutto un verde più uniforme.

In caso si voglia attirare l'attenzione di una persona su un determinato particolare si possono utilizzare diversi canali:

- Forma geometrica
- Colore

- Spessore delle linee
- Curvatura
- Ombreggiatura

Questi canali sono ordinati in base a quello più utile ad attirare l'attenzione, però è importante che questi canali vengano utilizzati con cautela e parsimonia, in quanto **un'uso esagerato potrebbe causare l'effetto contrario**(e.g. rendendo tutte le linee di un testo spesse, non si otterrà più l'effetto di attirare l'attenzione, in quanto diventerà la "normalità" avere le linee di quello spessore). Nell'immagine sottostante si ha l'obiettivo di far saltare all'occhio il quadrato rosso, si nota facilmente come l'uso di diversi canali contemporaneamente possa rendere difficoltoso ottenere l'obiettivo desiderato.



Figure 4: Facile

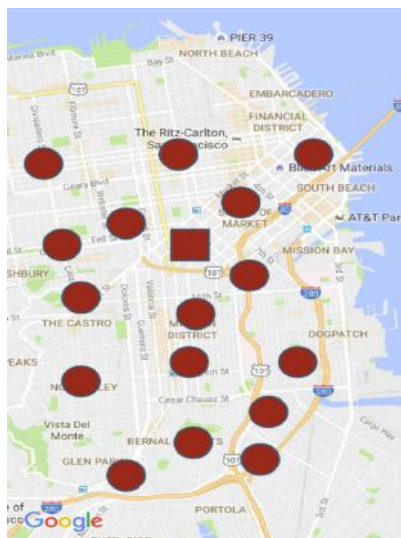


Figure 5: Normale

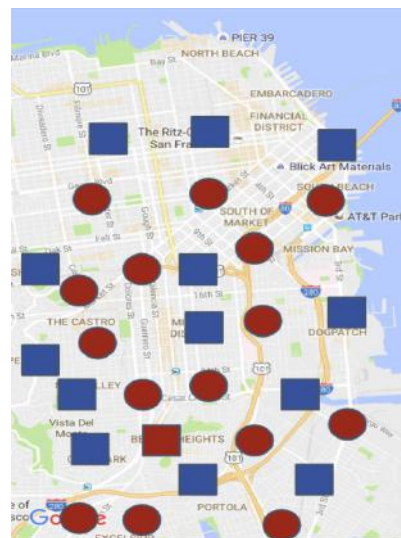


Figure 6: Difficile

4.3 Visione stereoscopica e interpretazione del segnale

Dato che il nostro occhio utilizza 2 immagini, ci permette di percepire meglio la profondità, infatti si possono notare le differenze date dal diverso punto di vista dei due occhi, il nostro cervello unisce queste due immagini in una sola, **utilizzando gli indizi di profondità dati da ogni occhio per creare un'immagine corretta della realtà**, questo processo viene chiamato **fusione binoculare**.

Il nostro sistema visivo è in grado di compensare

- Movimenti degli oggetti che si guardano(Riuscire a seguire un oggetto con gli occhi)
- Modifiche in luminanza(Il nostro cervello riconosce un oggetto anche in caso di riduzione di luminosità)
- Risolvere ambiguità di interpretazione della scena attraverso le informazioni di contesto

Quest'ultimo punto può però creare diverse ambiguità, ed è il fenomeno principale sul quale si basano le illusioni, infatti se per riconoscere un oggetto vengono utilizzate informazioni di contesto, due oggetti uguali in contesti diversi potrebbero risultare diversi(**Fenomeno della sovra-compensazione**).

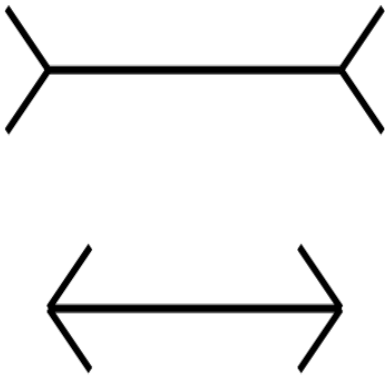


Figure 7: Illusione di Muller-Lyer, due rette di ugual misura si percepiscono diverse

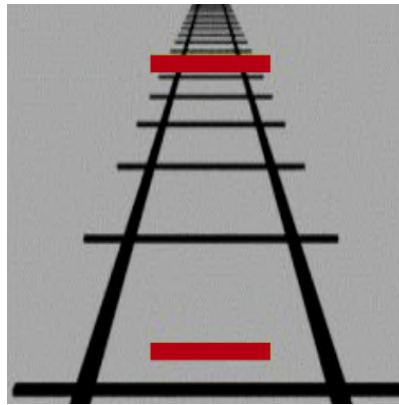


Figure 8: Questa è chiamata illusione di Ponzo, le due linee rosse vengono percepite di lunghezza diversa, in quanto il nostro cervello agisce nello stesso modo con il quale agisce con le rotaie nello sfondo, rendendo l'immagine più piccola allontanandosi

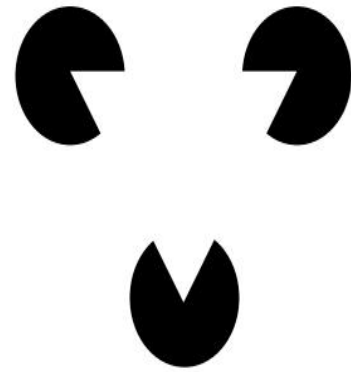


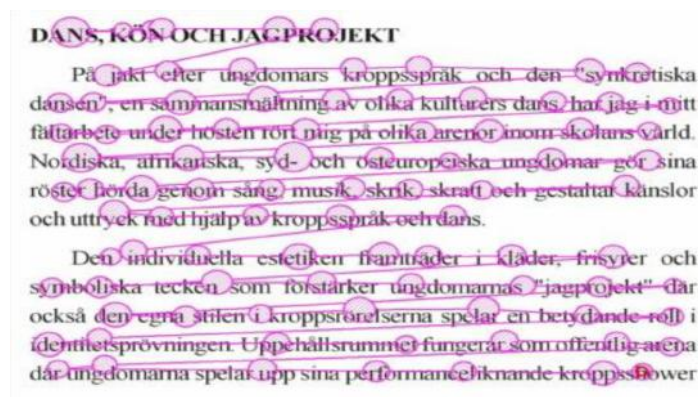
Figure 9: All'interno di un interfaccia **possono esistere immagini non reali** anche tramite un'illusione ottica tramite indizi visivi

4.4 Il processo di lettura

Il processo di lettura è una parte molto importante del processo visivo, la maggior parte delle informazioni all'interno di un interfaccia sono scritte. Il processo di lettura si divide in più stadi

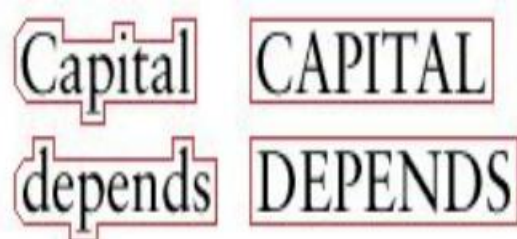
- Percezione del pattern visuale.
- Decodifica utilizzando la rappresentazione interna del linguaggio.
- Interpretazione utilizzando la conoscenza della sintassi, della semantica e della pragmatica di una lingua.

Il processo di lettura è diviso in serie di **movimenti saccadici** e di **momenti di fissazione**. Lo stadio della percezione si ha durante il momento di fissazione (circa il 94% del tempo).

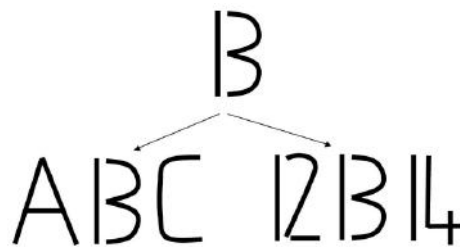


Nel processo di riconoscimento, la forma delle **parole** (e non lettere) è molto importante, infatti esistono metodi per migliorare questo processo, uno fra tutti l'utilizzare scure su sfondo chiaro. In media una persona legge circa 250 parole al minuto, questo non sarebbe possibile se ogni parola venisse scandita carattere per carattere, infatti il nostro cervello, come detto in precedenza,

riconosce le parole dalla loro **forma**. Per questo motivo il testo viene scritto in stampatello minuscolo, in quanto ha una forma molto ben definita rispetto allo stampatello (lo stampatello viene letto lettera per lettera, quindi è utilizzato per sigle o lettere singole), questo perché la leggibilità dipende perlopiù dalla parte superiore delle parole, che nel caso delle minuscole ha dimensione diversa fra le lettere, mentre nello stampatello ogni lettera ha la stessa altezza.



Il riconoscimento della forma di una parola dipende anche dal contesto all'interno del quale si trova. La velocità massima di lettura ottenibile si ha con font di grandezza che varia dai 9 ai 12 punti, e con una lunghezza di riga tra i 58 e i 132mm.



4.4.1 I font e la loro importanza

I font possono essere di diversi tipi

- Serif(spaziatura proporzionale)
- Sans Serif(spaziatura proporzionale)
- Monospaziato(spaziatura fissa)

I caratteri vengono misurati in punti(points,pt), un punto digitale vale circa 0.35mm. Con l'avvento della tecnologia ha avuto l'inizio l'era dei display digitali, nel quale ogni giorno tutte le persone leggono le informazioni, ma c'è una grande differenza fra la lettura su carta e quella su video, infatti il testo a video è di qualità molto inferiore rispetto a quello su stampa, questo perché alcuni caratteri vengono resi male a video a causa del fenomeno del **jagging**(caratteri che vengono composti da pixel semi trasparenti). Nel 1987 venne condotto uno studio nel quale si stabilì che **la velocità di lettura su schermo era del 25% più lenta rispetto a quella su carta** in quanto le fissazioni sono il 15% in più per riga. Con il passare del tempo si sono trovate diverse metodologie per alleviare questa differenza, la più importante è la differenza tra i tipi di font da utilizzare, infatti si è studiato che su carta i caratteri Serif rendono un testo più leggibile rispetto ai caratteri Sans Serif, mentre sul digitale si ha l'effetto opposto, in quanto i caratteri sans Serif alleviano il fenomeno del jagging. Un'altra differenza è l'allineamento del testo, infatti sul web si utilizza l'allineamento a sinistra, mentre su carta si utilizza il giustificato(modalità nella quale in ogni linea le parole vengono suddivise in modo da poter occupare l'intero spazio orizzontale).

4.5 Compendio di tipografia digitale

Occupandosi del design di interfacce è importante non riempire un'interfaccia di testo(wall of text). Le parti piene devono essere bilanciate da parti vuote per dare aria al testo, e al design in

generale, questo viene fatto tramite blocchi uniformi e regolari che favoriscono la leggibilità. Per esempio, le figure a pagina 10, sono un chiaro esempio di come, degli spazi bianchi possano dividere ordinatamente diverse sezioni, creando con gli spazi bianchi fra loro delle sorta di "colonne". Nella scrittura di un testo si devono inoltre evitare 3 casi

- Vedove → Parola unica per riga.
- Orfani → Parola unica che appare in una nuova colonna.
- Fiumi → Spazi troppo larghi fra parole nel testo giustificato.

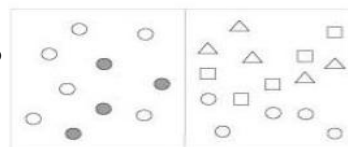
Esistono diversi **metodi per comunicare qualcosa ad un lettore**, il **font** facilita la lettura del corpo del testo, in quanto si può variare per attirare l'attenzione del lettore, inoltre si può utilizzare il **Serif** per titoli o sottotitoli. Il **grassetto** viene utilizzato per evidenziare sostantivi ed aggettivi utili a "riassumere" il concetto racchiuso in un testo, il **corsivo** viene utilizzato per dare enfasi ad un testo, mentre il **sottolineato** invita al click.

4.6 Principi per il raggruppamento

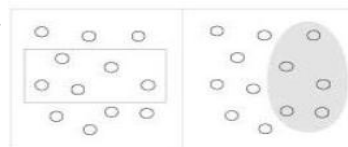
- **Prossimità.** Oggetti vicini vengono percepiti come un gruppo



- **Similarità.** Oggetti che hanno forma simile vengono percepiti come un gruppo



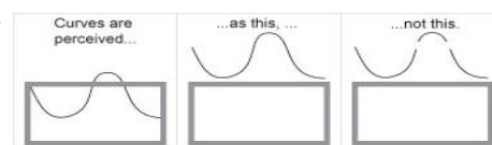
- **Raggruppamento.** Oggetti che hanno un bordo attorno sono percepiti come un gruppo.



- **Chiusura.** Le strutture aperte vengono percepite come chiuse se la loro regolarità lo consente



- **Continuità.** Oggetti allineati o che sembrano continui sono percepiti come un gruppo



4.6.1 F-shape

Il fenomeno della F-shape è la "forma" con la quale viene letta una pagina web, viene chiamata così in quanto viene prima letto con un movimento orizzontale la parte alta del contenuto, per poi ripetere questo stesso movimento scendendo nel documento, questo comportamento poi muterà nello scansionamento della sola parte sinistra con un movimento verticale, quindi formando appunto una sorta di forma ad F. I due motivi principali per il quale questo succede sono **il basso interesse e la necessità di rapidità nell'estrarre l'informazione**. (Quando si entra in una pagina web generalmente si scorre con velocità la pagina per verificare che possa contenere le informazioni desiderate).

Esistono dei metodi però per contrastare questo evento:

- Inserire il contenuto più importante nei **primi due paragrafi della pagina**

- Usare i **titoli** per sezioni e sottosezioni
- Utilizzare **parole rilevanti** per titoli e sottotitoli
- **Raggruppare** contenuti
- Utilizzare il **grassetto** per i contenuti importanti
- Evidenziare i **link**. Utilizzare **parole che anticipino** il contenuto (Non si utilizza per esempio "vai", "clicca", ecc..)
- Utilizzare **elenchi** puntati e numerati per liste e processi
- **eliminare il superfluo**

CONCETTI IMPORTANTI

La leggibilità sul video dipende fortemente dal tipo di font, e dalla sua dimensione. Normalmente si raccomanda sul video di utilizzare font sans serif, evitando comunque per quanto è possibile il corsivo.

Poiché la lettura sul video è faticosa, occorre comunque evitare per quanto è possibile testi lunghi.

Testi lunghi in caratteri maiuscoli vanno comunque sempre evitati.

Usare pochi font contemporaneamente, e usarli in modo consistente per veicolare informazione.

Guidare la lettura su web tramite la tipografia.

4.7 L'udito

L'udito è uno dei sensi più utilizzati nella creazione di interfacce, in quanto consente all'umano di percepire i suoni. Il suono fornisce informazioni sull'ambiente (distanze, direzioni, ecc..), il soggetto della percezione uditiva sono principalmente le **sorgenti primarie**, ovvero gli oggetti che emettono il suono (mentre la percezione visiva utilizza come soggetto le sorgenti secondarie, ovvero gli oggetti che riflettono la luce). L'udito è importante in quanto si tratta di un canale di input che **non richiede attenzione pregressa**, infatti un suono richiama attenzione anche se tu in quel momento non sei "attento" a captarlo.

4.7.1 Anatomia dell'orecchio

L'orecchio si divide in 3 parti:

- **Esterno.** Protegge l'orecchio interno e amplifica il suono.
- **Medio.** Trasmette le onde sonore come vibrazioni all'interno.
- **Interno.** Trasmettitori chimici che vengono rilasciati e che causano impulsi al nervo uditivo.

L'orecchio interno percepisce le accelerazioni nell'apparato vestibolare, per questo si verifica la chinetosi (Insieme di disturbi provocati per viaggi in auto/barca/aereo, caratterizzato da nausea, vomito e vertigini)

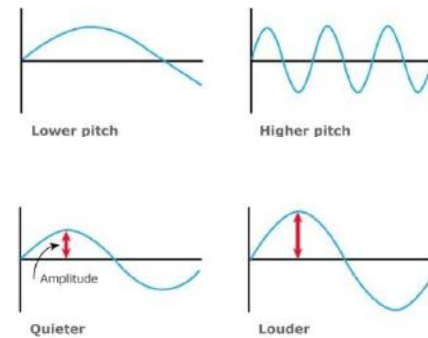
4.7.2 Il Suono

Il suono ha 3 caratteristiche principali.

l'altezza(pitch), che è data dalla frequenza del suono, che può essere grave(bassa frequenza) o acuto(alta frequenza)

Intensità(loudness), ovvero il livello del "volume", questo è indipendente dalla frequenza, infatti con frequenze uguali si possono avere volumi diversi

Timbro è legato al tipo di fonte di produzione del suono utilizzato



L'essere umano può sentire frequenze dai 20 Hz ai 15 KHz, distingue meno accuratamente le alte frequenze rispetto alle basse. Il sistema uditivo **filtra** i suoni, eliminando così il rumore di sottofondo, consentendoci di udire suoni familiari senza prestare attenzione(cocktail party effect); questo meccanismo di eliminazione del suono non entra in atto quando si sente un suono "strano", che non viene filtrato.

4.8 Tatto

Il tatto fornisce importanti feedback circa l'ambiente, utili per fornire informazioni su ciò che ci circonda(temperatura, peso, ecc...)e anche per il coordinamento dei movimenti. Esistono metodi di comunicazione che si effettuano con un'utente tramite il tatto, come il **Feedback aptico**(quello che fa vibrare il dispositivo quando premi un tasto), che utilizza sia il tatto che la cinestesia(Sensazione provocata dal movimento e specialmente dalla contrazione dei muscoli volontari). Il tatto è un senso non localizzato, ovvero è diffuso su tutto il corpo, nonostante ciò sono presenti aree più sensibili di altre, e questo dipende dal loro uso. Gli stimoli sono ricevuti mediante recettori nervosi della pelle, questi sono di tre tipi

- **Termocettori.** sensibili alla temperatura
- **Chemiocettori.** sensibili alla pressione, calore e dolore intensi
- **Meccanocettori.** sensibili alla pressione

Il tatto inoltre assume una funzione ancora più importante se si pensa a persone visivamente disabili, basti pensare alla scrittura Braille.

La cinestesia precedentemente citata, permette di dare consapevolezza della posizione del proprio corpo e degli arti, i recettori che permettono di ottenere queste informazioni si trovano nei muscoli e nei tendini del nostro corpo, questo feedback viene trasmesso poi al sistema motorio.

Il sistema propriocettivo è colui che dà la capacità di riconoscere la posizione e il movimento del proprio corpo nello spazio senza l'ausilio della vista, questo è utilissimo nella vita di tutti i giorni.

4.9 Movimento e tempo di reazione

Per effettuare un movimento è necessario un tempo pari al tempo di **reazione** + il tempo di **movimento**, quest'ultimo dipende da diversi fattori, tra cui l'età, la forma fisica e l'allenamento. In base al **tipo di stimolo** si hanno tempi di reazione diversi:

- **visivo:** ~ 200ms
- **uditivo:** ~ 150ms
- **dolore:** ~ 700ms

Più sarà lungo il tempo di reazione, minore sarà l'accuratezza della reazione, questo però non si verifica in un'utente addestrato. Concentrandosi sul movimento per puntare un oggetto su schermo

si può utilizzare un'equazione, chiamata **legge di Fitts**. Quest'equazione serve a descrivere il tempo impiegato per raggiungere un obiettivo su schermo:

$$Mt = a + b \log_2 \left(\frac{D}{W} + 1 \right) \quad (1)$$

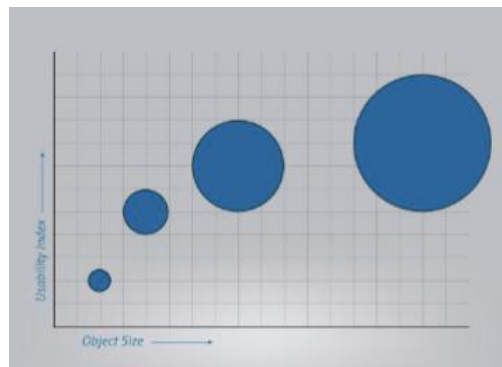
Le grandezze prese in considerazione sono:

- a e b, che sono costanti determinate empiricamente
- Mt, ovvero il tempo di movimento
- D, la distanza da ricoprire
- W, la dimensione dell'obiettivo

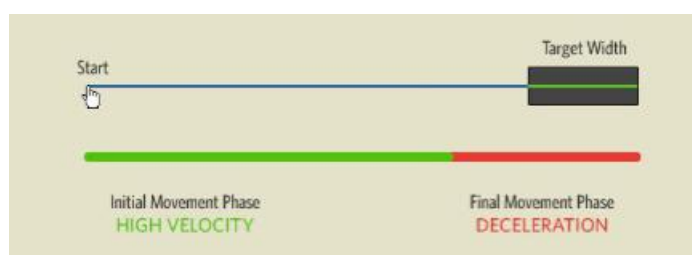
Quindi per diminuire il tempo di movimento si può **aumentare la dimensione** dell'obiettivo, oppure **diminuirne la distanza**.



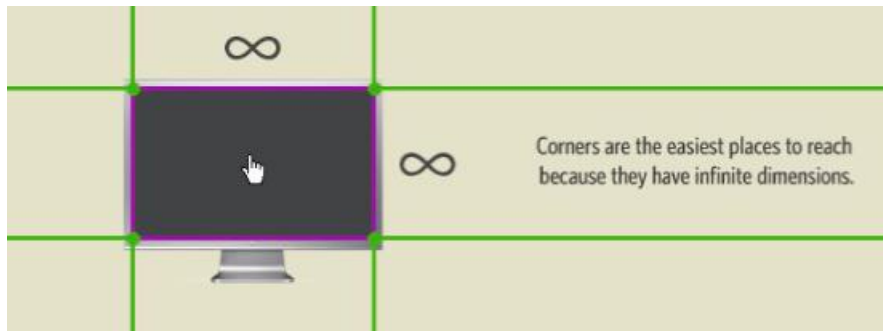
I progettisti di interfacce quindi tendono ad aumentare la dimensione dei controlli che vengono utilizzati più spesso e di concentrarli in una zona facilmente accessibile. Questa legge è **logaritmica**, ciò significa che aumentando la dimensione del 20% di un oggetto piccolo, il tempo di movimento diminuirà sensibilmente, ma se si aumenta del 20% un oggetto già grande, il tempo di movimento non diminuirà di molto.



Il movimento inoltre si divide in due fasi, nella prima fase si effettua un **movimento rapido ma impreciso**, mentre nella seconda fase **cala la velocità in modo da aumentare la precisione**.



La parte più facilmente raggiungibile da un cursore nello schermo sono **gli angoli**, questo perché gli angoli possono essere considerati come **punti a dimensione infinita** dato che il puntatore non può uscire dal box dello schermo.



4.10 Memoria

La memoria è formata da un'insieme di **processi cognitivi** che regolano la capacità di apprendere **dall'esperienza** e di pianificare comportamenti futuri. E' divisa in 3 fasi distinte

- acquisizione: la percezione del materiale da apprendere
- ritenzione: la trasformazione dell'informazione in ricordi
- recupero: la rievocazione del ricordo dalla memoria

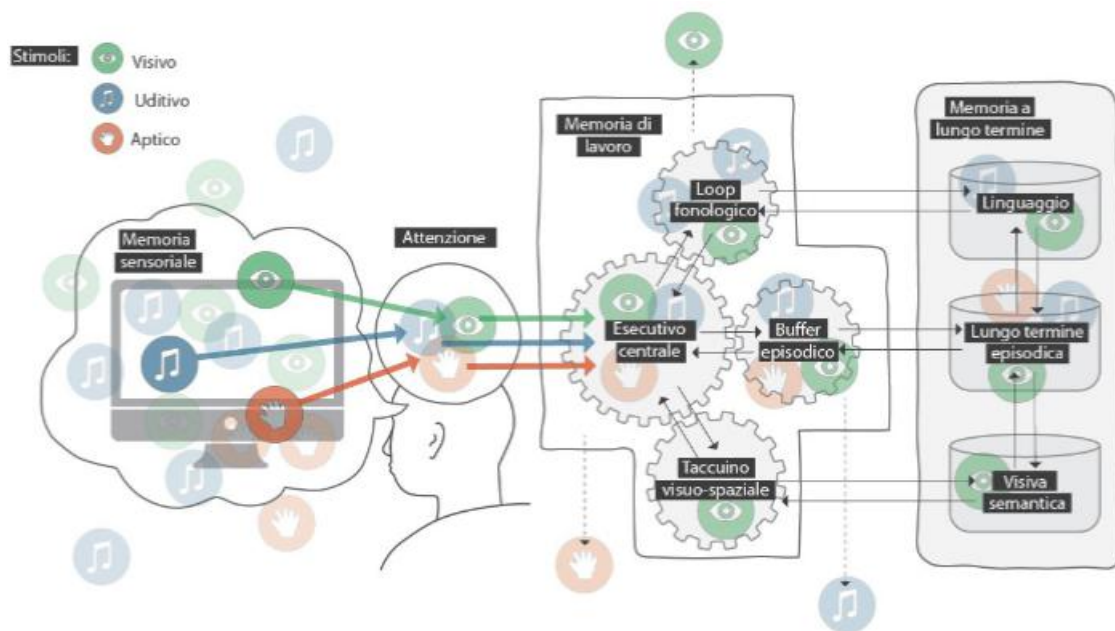


Figure 10: Modello schematico con tre tipologie principali

La memoria si divide in 3 tipi principali, quella **sensoriale (SM)**, ha una capacità limitatissima, che va dai 200 ai 500 ms; poi quella **a breve termine (STM)**, questa memoria viene anche chiamata di lavoro, questa permane da pochi secondi fino ad un minuto, quindi ha una capacità migliore della sensoriale, e poi vi è la memoria **a lungo termine (LTM)**, un'informazione contenuta in questa memoria potrebbe potenzialmente rimanere per tutta la vita.

4.10.1 Memoria sensoriale

E' composta da buffers(memorie tampone) che registrano temporaneamente gli stimoli sensoriali, attraverso 3 tipi di memoria:

- Memoria iconica(stimoli visivi)
- Memoria ecoica(stimoli uditivi)
- Memoria aptica(stimoli tattili)

Un'esempio della memoria iconica possono essere i fuochi d'artificio, mentre della memoria ecoica la direzione del suono. Questi buffer vengono **riscritti continuamente**, infatti si ha una persistenza massima di 2s per la memoria ecoica, mentre per la memoria iconica solo 300ms. Questa memoria consente di estrarre informazioni in modo differenziale, ovvero quando è utile valutare traiettorie e direzioni.

L'attenzione

L'attenzione è un meccanismo che seleziona degli stimoli in base all'interesse, questo perché se gli stimoli non venissero selezionati ne verremmo sovraccaricati, la selezione degli stimoli a cui prestare attenzione è determinata dal livello di interesse o di necessità. L'attenzione è quello strumento tramite il quale le informazioni vengono passate dalla memoria sensoriale alla memoria a breve termine.

4.10.2 Memoria a breve termine(STM)

La memoria temporanea effettua una registrazione momentanea dell'informazione che è oggetto della elaborazione in corso, funziona come una memoria di lavoro(cache), un'esempio di funzionamento della STM è che per comprendere un discorso bisogna ricordarlo tutto dall'inizio. Le sue caratteristiche sono il suo accesso rapido($\sim 70ms$), ma anche il suo decadimento altrettanto rapido($\sim 200ms$). Uno dei fondamenti della STM è stato introdotto da Miller, tramite la sua omonima legge, quest'ultima afferma che il **numero di oggetti(chunks) che una mente umana media può tenere in mente mentre è in funzione è 7 ± 2** . Questi oggetti, chiamati **blocchi di informazione** sono quelli che l'utente deve ricordare. Tramite questi chunks è organizzata la nostra STM, questi sono molto utili per aumentare la nostra capacità di ricordare informazioni, difatti **la nostra mente tende sempre a memorizzare le informazioni a blocchi**, e non come una serie di dati(e.g. generalmente un numero di telefono viene memorizzato come 070 75 31 33, e non come 070753133), la formazione di un singolo blocco è detta **chiusura**. Una volta raggiunto il nostro obiettivo svuotiamo la memoria a breve termine, questo può comportare un'errore chiamato **post completion error**(Scordarsi di ritirare la carta una volta prelevato, non effettuare logout, dimenticarsi di salvare un file, ecc). Durante la fase di mantenimento nella memoria a breve termine, si possono verificare interferenze causate dalla memorizzazione:

- Il ricordo di quanto visto più recente è migliore del ricordo di cose precedenti(**recency effect**)
- L'effetto scomparire se tra presentazione e ricordo il soggetto viene impegnato con un altro compito(Per esempio, se mentre si sta scrivendo un testo e ci si distrae, spesso bisogna rifare un punto della situazione per capire a dove si è arrivati)

Questi problemi di interferenze si hanno quando si provano ad utilizzare gli stessi **componenti della memoria**(teorizzati nel modello di Baddeley) per più attività diverse, quindi l'essere umano ha bisogno di completare senza distrazioni i compiti che impegnano la STM.

I quattro componenti del modello di Baddeley(4.10) sono:

- **Esecutivo centrale.** E' il supervisore che gestisce le risorse limitate di memoria e coordina gli altri sistemi sussidiari
- **Taccuino Visuo-Spaziale.** Si occupa del mantenimento dell'informazione visuo-spaziale e aptica.
- **Loop fonologico.** Mantenimento ed elaborazione dell'informazione verbale e acustica.
- **Buffer episodico.** Integrazione dell'informazione degli altri sistemi sensoriali con le informazioni provenienti dalla memoria a lungo termine.

Per sviluppare un buon programma è nei nostri interessi dare supporto all'utente **riducendo il più possibile l'utilizzo della memoria di lavoro** (e.g. Recap di un acquisto al momento del pagamento), non richiedere mai che l'utente si debba ricordare, anche per breve tempo, gli stati operativi del sistema, dare sempre **indicazione dello stato in cui il processo corrente è arrivato**, aiutare sempre l'utente a individuare chiaramente qual è il **prossimo input** che il sistema si aspetta.

4.10.3 Memoria a lungo termine(LTM)

E' l'area di archivio di tutta la nostra conoscenza, ha un accesso più lento rispetto alle altre memorie, però ha un tempo di decadimento lento se non addirittura nullo. Esistono due tipi di LTM

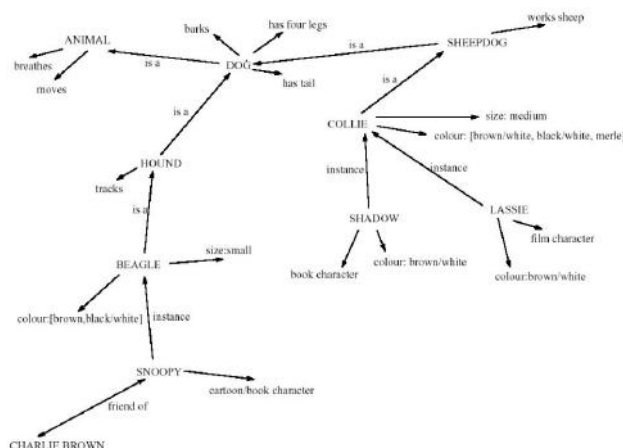
- **Esplicita o Dichiarativa**, raccoglie tutte le memorie che sono disponibili in maniera conscia, si divide a sua volta in
 - **Episodica**. memoria di eventi ed esperienze in forma seriale(Ci si ricorda di aver fatto qualcosa, un viaggio, essere andati a cena in un determinato posto).
 - **Semantica**. memoria strutturate di fatti, concetti e capacità.
- **Implicita o Procedurale**, le informazioni che utilizziamo senza rendercene conto(usare una penna o andare in bicicletta)

L'informazione nella memoria esplicita **semantica deriva da quella episodica**: si imparano fatti o concetti dalle esperienze. La memoria semantica è strutturata in modo da fornire accesso alle informazioni, rappresentare le relazioni tra le informazioni e supportare il ragionamento. La memoria umana è diversa da quella di un computer: quella umana è distribuita in varie parti del cervello che interagiscono tra di loro, quindi non vi è materialmente la divisione all'interno del nostro cervello della memoria.

Per **rappresentare la memoria a lungo termine esistono vari modelli** che sono complementari tra loro e non mutualmente esclusivi

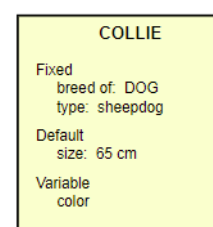
RETE SEMANTICA

E' un modello a rete, rappresentato come un grafo diretto: possiede la caratteristica dell'**ereditarietà**, ovvero i nodi figli ereditano le proprietà dei nodi padri, sono presenti **relazioni esplicite tra elementi informativi**, e supporta la **deduzione** tramite ereditarietà



FRAMES

L'informazione è organizzata in una **struttura dati**, i campi della struttura sono riempiti con valori specifici per un certo esemplare, è utile per rappresentare le **informazioni stereotipate di un determinato soggetto**, le relazioni sono **tipo-sottotipo**



SCRIPTS

Rappresenta l'informazione che ci raggiunge dal mondo esterno, **viene completata** dalla nostra memoria a lungo termine, questo è utile per **ricostruire situazioni usuali e interpretare situazioni**.

Script: Visita veterinaria			
Situazione iniziale	Il cane è ammalato L'ambulatorio è aperto Il proprietario può pagare	Compiti	Il veterinario visita, diagnosi, trattamenti Il proprietario porta il cane in ambulatorio, paga, porta il cane fuori dall'ambulatorio
Risultati	Il cane sta meglio Il proprietario ha meno soldi Il veterinario ha più soldi	Scene	Arrivo alla reception Attesa nella sala d'aspetto Visita Pagamento
Proprietà	Tabella degli esami Medicine Strumenti	Tracce	Il cane ha bisogno di medicine Il cane ha bisogno di un intervento

REGOLE DI PRODUZIONE

Rappresentano la nostra conoscenza procedurale, costruisce delle regole di **condizione-azione**, se la condizione è verificata l'azione conseguente viene svolta

SE il cane muove la coda
ALLORA accarezza il cane

SE il cane ringhia
ALLORA scappa

I **processi della LTM** sono le azioni utilizzate per gestire l'informazione all'interno della memoria. Esistono tre tipi di processi

- **Memorizzare** (archiviare) informazioni
- **Dimenticare** (cancellare) informazioni
- **Recuperare** (accedere alle) informazioni

Il processo di memorizzazione passa dalla memoria a breve termine a quella a lungo termine per ripetizione, **rehearsal** (reiterazione dello stimolo, per studiare un qualcosa si legge più volte). Il materiale presente in memoria può essere **organizzato, creando connessioni e astrazioni**. Per effettuare questo passaggio da una memoria all'altra si è formulata l'ipotesi del tempo totale, ovvero la quantità di informazione memorizzata è proporzionale al tempo di ripetizione(per più tempo si studia, più si ricorda), però per questo bisogna tener conto della **Distribution of practice effect**, ovvero a parità di tempo totale, si hanno migliori risultati quando **questo tempo è distribuito in un più grande lasso di tempo**, l'informazione è più semplice da ricordare quando è presente una struttura, un significato o si tratta di argomenti familiari, ricordare concetti astratti è più difficile rispetto ad oggetti reali(per questo sono utilizzati gli esempi).

La LTM codifica in modo diverso l'informazione in base al canale sensoriale, sono stati ipotizzati due sistemi per descrivere i diversi comportamenti: **verbale**(gestisce informazioni linguistiche), e **visuale**(gestisce informazioni visive), le immagini sono rappresentate da un'unione di entrambi i sistemi. Per questo esiste la **Superiorità dell'immagine**, ovvero le immagini tendono ad essere ricordate più facilmente e per più tempo rispetto ad un'informazione verbale.

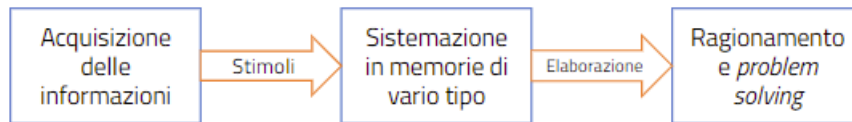
Dalla LTM si può verificare anche una perdita delle informazioni(dimenticate), questo avviene tramite due meccanismi

- **Decadimento** L'informazione viene persa gradualmente ma lentamente
- **Interferenza**, la nuova informazione rimpiazza la vecchia, questo avvenimento è chiamato **interferenza retroattiva**(nuovo numero di telefono rimpiazza il vecchio) può succedere che l'informazione vecchia può interferire con la nuova: questo meccanismo si chiama **inibizione proattiva** (mettere una password vecchia invece di una aggiornata)

Le parole legate ad emozioni si ricordano meglio di quelle legate ad eventi normali, e tra le emozioni si ha che gli eventi positivi si ricordano meglio di quelli negativi. Il recupero dell'informazione

avviene tramite un **ricordo**, tramite questo meccanismo l'informazione viene ripescata dalla memoria, in questo caso il processo può essere agevolato da indizi, oppure tramite il **riconoscimento**, la presentazione di un'informazione provoca la consapevolezza che tale informazione è stata già vista, questo è **meno complesso del ricordo**: l'informazione presentata è un indizio.

4.11 Pensiero e ragionamento



Il pensiero è probabilmente ciò che distingue l'essere umano dagli animali, il pensiero è svolto con due attività: il **ragionamento**, e la **risoluzione di problemi**. Il ragionamento è il processo con cui si usa la conoscenza di cui si dispone per trarre conclusioni o dedurre qualcosa di nuovo sul dominio di interesse, esistono tre tipi di ragionamento:

- **Ragionamento deduttivo**

Si ha quando le **conclusioni sono logicamente derivate dalle premesse** date (Se è Venerdì, lei andrà a lavorare), si possono derivare conclusioni logiche anche se non hanno senso, e vengono escluse per **conoscenza del mondo** (Se piove il terreno è asciutto, falso in quanto non sarebbe una reazione realistica). E' anche facile interpretare male le regole del ragionamento deduttivo (Alcune persone sono bambini, alcuni bambini piangono, alcune persone piangono FALSO).

- **Ragionamento induttivo**

Generalizza da casi visti a casi non visti (Tutti gli elefanti che ho visto hanno la proboscide, quindi tutti gli elefanti hanno la proboscide), non è affidabile perché consente di provare solo falsità di ipotesi e non la loro verità, però è molto utile per costruire modelli. E' difficile utilizzare questo metodo in quanto l'essere umano usa meglio l'evidenza positiva che l'evidenza negativa.

- **Ragionamento abduttivo**

risale dagli eventi alle cause, questo può portare a false spiegazioni, quindi non è affidabile (Marco guida veloce quando ha bevuto, quindi se vedo Marco guidare veloce assumo che abbia bevuto).

4.12 Problem solving

E' importante capire come le persone risolvano i problemi. La soluzione di un problema può essere vista come una modifica della configurazione del sistema che ci circonda, in modo da compiere un obiettivo. Questo può essere fatto utilizzando la nostra conoscenza, e se non si hanno conoscenze di un certo ambito si cerca di utilizzare quelle di un sistema analogo. Esistono diverse teorie che spiegano il funzionamento di questo meccanismo:

- Gestalt (produttivo e riproduttivo)
- Teoria del **problem space**
- Analogia

Gestalt

In questa teoria il problem solving può essere

- Riproduttivo: Riproduzione di soluzioni prese dall'esperienza pregressa (spiegazione comportamentale)
- Produttivo: Cerca di ristrutturare il problema con intuito (insight)

Teoria del problem space

Questa è stata sviluppata negli anni 70', il **problem space** comprende tutti gli stati del problema (la soluzione è un particolare stato), ovvero tutte le descrizioni possibili dello stato del mondo che ci interessa per quel dato problema. Il **problem solving**, comporta la generazione di stati usando operatori legali (mosse), questa teoria è largamente applicata alla soluzione di problemi in contesti ben definiti (giochi, puzzle, ecc...), quindi questo risulta utile anche per capire come l'utente agisca in un'interfaccia per ottenere un suo obiettivo. Nella teoria del problem space si modella il problema tramite 4 componenti:

- Operatori
- Condizioni
- Obiettivi secondari
- risultato

Per selezionare gli operatori si usano euristiche, infatti si utilizza l'analisi *means-ends*, nel quale lo stato corrente viene confrontato con il goal e poi viene scelto l'operatore che più riduce la differenza tra i due. Il problem space opera entro i sistemi umani di elaborazione delle informazioni (Capacità della STM e velocità di recupero dei dati).

Esempio problem space

Se si volesse spostare la scrivania fuori da una stanza, è importante sapere la posizione della scrivania e cosa ci sia sopra, la configurazione obiettivo sarebbe il cambio di posizione della scrivania, non essendo possibile spostare direttamente la scrivania, si devono utilizzare degli stati intermedi, come lo spostamento degli oggetti dalla scrivania. In questo contesto avremmo i 4 componenti di modellazione del problema come: sollevare oggetti, trascinarli, ecc.. sarebbero gli operatori; le condizioni sarebbero per esempio il poter sollevare un oggetto solo se è leggero; un obiettivo secondario sarebbe non poter sollevare la scrivania perché piena di oggetti, e quindi doverla alleggerire; il risultato sarebbero l'insieme delle posizioni intermedie di oggetti e scrivania.

4.13 Analogia

I nuovi problemi vengono risolti per analogia, ovvero utilizzando una soluzione di un diverso dominio simile all'interno di un nuovo problema. Una volta stabilita la corrispondenza, metodi e soluzioni del dominio noto vengono trasferiti in quello nuovo. L'analogia può essere difficile da stabilire, specie se i domini sono semanticamente diversi.

L'analogia è molto importante, in quanto sta alla base del sistema delle **metafore**, che sono utilissime nel mondo delle interfacce (un riproduttore musicale ha gli stessi tasti che avevano i vecchi lettori audio, in quanto gli utenti sapevano utilizzare quelli). Spesso i problemi che dobbiamo affrontare sono ricorrenti, per questo un soggetto diventa esperto (ovvero più abile a risolvere problemi), un utente è considerato esperto quando usa il **chunking** per gestire al meglio la STM, e ha più capacità di raggruppare problemi in base ad **analogie concettuali**. Il comportamento di un esperto diventa automatico nel tempo, e questo può causare errori.

L'emozione è l'espressione conscia dell'esperienza affettiva con l'identificazione delle sue cause e del suo oggetto, ci sono varie teorie che spiegano come funziona l'emozione, sicuramente l'emozione ha a che vedere sia con la risposta cognitiva che fisiologica agli stimoli sensoriali. **L'usabilità** di un

sistema non ha a che fare solo con la sua funzionalità, ma anche con aspetti emozionali: emozioni **positive** possono migliorare l'utilizzo di un interfaccia, facilitare l'apprendimento e l'utilizzo, e rendere più tollerabili i difetti, al contrario emozioni negative amplificano l'effetto dei difetti di usabilità.

Le emozioni provate sono spesso soggettive e si basano sui gusti, questo significa che non tutti gli utenti siano uguali, sia dal punto di vista del **lungo termine** (diversi per sesso, caratteristiche fisiche ed intellettuali) che per **breve termine** (gli utenti variano le loro risposte e i comportamenti sotto effetti di emozioni diverse), si ha anche **variabilità nel corso del tempo** (gli utenti per esempio invecchiano), bisogna sempre chiedersi quindi se delle determinate scelte poste durante la progettazione possano tagliare fuori determinate fette di utenti.

5 Interazione

L'interazione è ogni comunicazione che avviene tra utente e computer, che sia diretta o indiretta. Questa comunicazione avviene attraverso l'interfaccia; affinché la comunicazione possa avere successo, l'interfaccia deve tradurre i concetti dell'utente in quelli del sistema. I termini utilizzati all'interno del contesto dell'attenzione sono:

- L'**utente** è colui che usa il sistema interattivo per raggiungere i suoi obiettivi nell'ambito di un certo dominio applicativo.
- L'**obiettivo** è quello che l'utente vuole raggiungere
- Il **dominio** è l'area applicativa che è oggetto dello studio (descrive concetti, competenze e conoscenze).
- L'**intenzione** è la sequenza di azioni generiche che portano al raggiungimento dell'obiettivo.
- Il **compito o azione** è la traduzione dell'intenzione in azioni specifiche.

5.1 Il modello di Norman

Il modello di Norman afferma che l'attenzione sia incentrata principalmente sulla visione che l'utente ha dell'interazione. Questo modello è basato su un ciclo di **esecuzione/valutazione**.



L'esecuzione si ha quando l'utente ha un obiettivo, formula un piano d'azione e lo esegue mediante l'interfaccia. La valutazione invece si ha quando l'utente osserva l'interfaccia per valutare il risultato dell'esecuzione (rispetto all'obiettivo) e determinare le azioni successive. Quindi gli step del ciclo valutazione/esecuzione sarebbe:

- L'utente stabilisce l'obiettivo

- Formula l'intenzione
- Specifica la sequenza di azioni
- Esegue le azioni
- Percepisce lo stato del sistema
- Interpreta lo stato del sistema
- Valuta lo stato del sistema in relazione all'obiettivo

Il ciclo di Norman si ripete quando si hanno grandi cambiamenti dell'interfaccia che l'utente deve rivalutare. L'interfaccia è interpretata come un traduttore, tra il linguaggio dei compiti e il linguaggio di base. Si hanno due differenze(**Golfi**)

- **Golfo dell'esecuzione.** Differenza tra le azioni formulate dall'utente per raggiungere l'obiettivo e le azioni permesse dal sistema.
- **Golfo della valutazione.** Differenza tra la presentazione dello stato del sistema e le aspettative dell'utente.

5.2 Errori

Gli errori possono essere di quattro tipi all'interno della programmazione di interfacce

- **Svista** (errore non intenzionale), capita quando si ha una corretta intenzione, ma si fallisce nel metterla in atto, una causa di ciò può essere una scarsa abilità fisica, o disattenzione, cambiamenti di contesto nel comportamento esperto possono causare sviste
- **Lapsus** errori dovuti a fallimenti della memoria, oppure di procedure errate.(e.g. una nuova versione di un prodotto viene rilasciata e la procedura nuova si discosta dalla vecchia, quindi gli utenti possono commettere un'errore di tipo Lapsus)
- **Sbaglio (o errore)** questo è il più grave dei quattro da parte di un progettista, e avviene quando si ha un'intenzione sbagliata che viene causata dalla comprensione incorretta del sistema o della situazione. L'uomo crea modelli mentali per spiegare il comportamento delle cose, se il modello è sbagliato, accadono errori(e.g. pensavo che l'interruttore accendesse la luce invece era il campanello)
- **Violazioni** questi sono errori deliberati nei quali il piano e l'azione corrispondono, ma sono in contrasto con procedure definite da un'istituzione.

Ricapitolando si ha che

La Svista è un errore non intenzionale, dove si ha una corretta formulazione dell'obiettivo, corretta formulazione dell'intenzione, ma errata esecuzione delle azioni.

Nel Lapsus si ha una corretta formulazione dell'obiettivo ma una formulazione errata dell'intenzione.

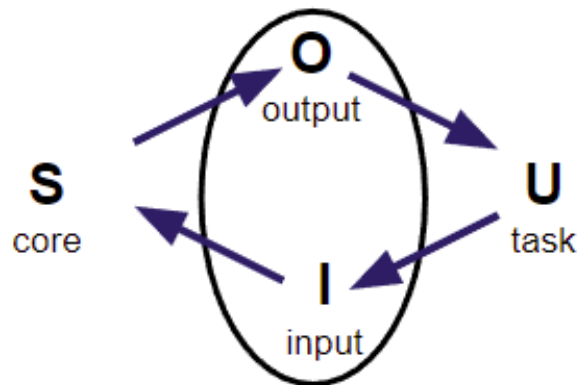
Nello Sbaglio(errore) si ha un obiettivo errato oppure intenzione errata, nel quale entrambe marcano una distanza fra modello del designer e quello dell'utente.

Le Violazioni si hanno quando si ha un obiettivo deliberatamente errato.

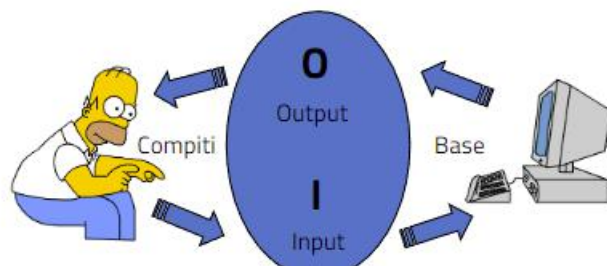
5.3 Il modello di Abowd e Beale

Il modello di Abowd e Beale è un' *Interaction Framework* nel quale si hanno 4 componenti

- Utente(U) linguaggio task
- Sistema(S) linguaggio core
- Input(I) linguaggio input
- Output(O) linguaggio output



In questo modello ogni componente ha il suo linguaggio, l'interazione funge da traduttore tra linguaggi, i problemi di interazione sono modellati come problemi di traduzione tra questi linguaggi, in alcune interfacce che utilizzano questo modello si hanno problemi a distinguere **parti di input** da **parti di output** (per esempio un'interfaccia di gioco)



5.4 I principi di usabilità

L'usabilità è la misura in cui un **prodotto** può essere usato da specifici **utenti** per raggiungere specifici **obiettivi** con efficacia, efficienza e soddisfazione in uno specifico **contesto d'uso**, i parametri che deve avere il raggiungimento di quest'obiettivo hanno significati ben precisi:

- **Efficacia.** Accuratezza e completezza rispetto agli obiettivi
- **Efficienza.** Insieme di risorse spese
- **Soddisfazione.** Comfort e accettabilità

Per dare una definizione breve dell'usabilità quindi possiamo dire che **valuti quanto sia semplice per l'utente utilizzare il sistema**, questo concetto si basa sull'assunzione che tutte le funzionalità del sistema siano implementate correttamente, ed è completamente slegato dalla quantità di funzionalità.

L'usabilità può essere **dimensionata**, attraverso diversi criteri **misurabili**, noi ci focalizzeremo su 3 aspetti fondamentali:

- **Apprendibilità**, ovvero quanto è facile apprendere l'uso del sistema
- **Efficienza**, quante risorse sono necessarie per usare il sistema
- **Sicurezza(safety)**, ovvero quanto sia possibile rimediare agli errori

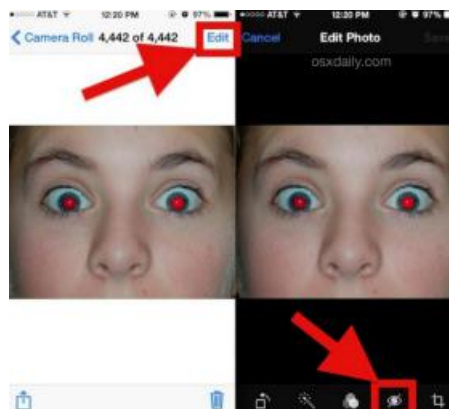
Le dimensioni hanno un'importanza relativa al loro task ed al loro target di utenti, infatti mentre per un utente inesperto sarà molto importante l'apprendibilità, per un esperto sarà importante l'efficienza (qua sorge anche il problema che si ha nel definire un utente esperto o inesperto in

modo uniforme). Come accennato in precedenza le dimensioni hanno importanza relativa anche al task, per esempio lanciare un missile richiede sicurezza, mentre l'ingresso in metropolitana richiede efficienza.



5.4.1 Come si impara ad utilizzare un sistema, l'apprendibilità

In generale per imparare un sistema non si legge un manuale, non si partecipa a corsi dedicati e non si impara utilizzando l'help fornito dal sistema, ma di solito si impara facendo (**learning by doing**), in questo modo l'utente ha un obiettivo, ed esplora l'interfaccia per trovare l'operatore che permetta di raggiungerlo



Un utente in difficoltà, cerca aiuto quando non sa più che fare, quindi formula una domanda **concreta**, e necessita di una **risposta precisa**, quindi è molto più utile una guida/help per task rispetto ad una per panoramiche generali

Punti importanti

E' importante **conoscere gli obiettivi dell'utente**.

Le interfacce **devono comunicare il loro funzionamento** e il modo nel quale devono essere utilizzate.

L'help deve supportare la ricerca ed essere **orientato ad obiettivi**.

Per rimediare ai propri errori e capire cosa non sia chiaro è possibile sfruttare i **big data**, infatti cercando su un motore di ricerca, è molto facile che qualcun altro prima di te abbia avuto il tuo stesso problema, e che sia stato aiutato, in questo modo chiunque può utilizzare la stessa soluzione data precedentemente ad un'altra persona.

5.4.2 Modelli concettuali

I modelli concettuali sono la descrizione del sistema in termini di idee e concetti integrati del suo funzionamento, comportamento e modo di presentarsi.

I modelli concettuali sono tre e **devono coincidere**

- Quello del progettista
- Quello implementato dal sistema
- Quello che si forma l'utente

In caso questi 3 modelli non coincidano si presentano problemi nell'interazione.

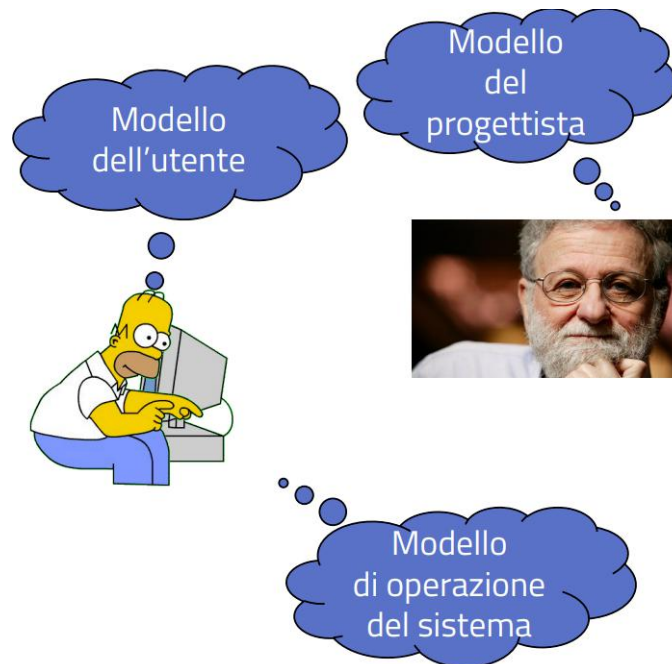


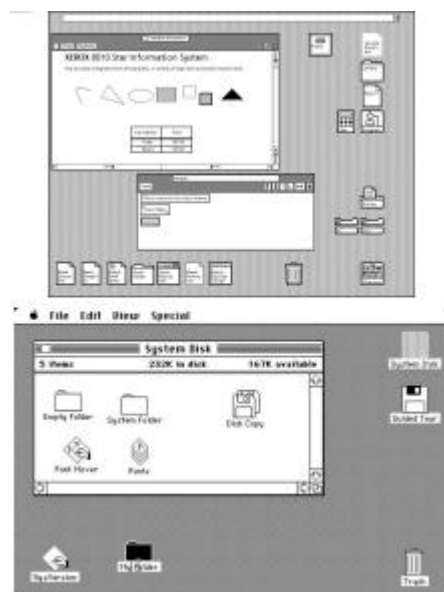
Figure 11: i 3 modelli concettuali

I modelli possono essere basati sull'attività, ovvero fornire istruzioni(comandi)

- Utente fornisce **istruzioni** al sistema su cosa fare(Videoregistratore)
- Il sistema si comporta come un interlocutore di una **conversazione**(Sistemi di prenotazione telefonica)
- **Manipolazione** di oggetti virtuali e **navigazione** in spazi virtuali(in questo caso si ha manipolazione diretta)
- **Esplorare e sfogliare** l'informazione come una rivista o un libro(ipertesto)

Si ha **manipolazione diretta**, quando gli oggetti sono rappresentati nella UI e l'utente li manipola in modo diretto(trascinando il mouse), purché una manipolazione venga considerata diretta devono sussistere 3 principi(inventati da Schneiderman nel 1982)

- **Visibilità** dell'oggetto di interesse
- **Azioni incrementali**, reversibili, con feedback immediato
- Sostituzione di **comandi** dalla sintassi complicata con **azioni**



I vantaggi della manipolazione diretta sono la **facilità di apprendimento**, il fatto che **eviti gli errori**, **incoraggia l'esplorazione** ed inoltre **sostituisce al ricordo il riconoscimento**

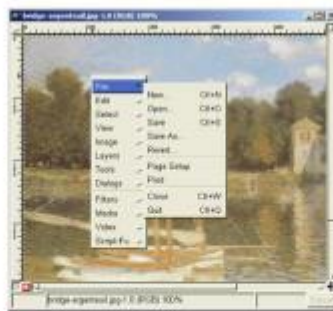
alleggerendo il carico cognitivo, mentre gli svantaggi sono che gli utenti esperti possono preferire metodi più snelli, e non tutte le operazioni si possono rendere i termini di manipolazione di oggetti.

5.4.3 La consistenza

La consistenza è una metodologia a supporto dell'apprendibilità di un sistema, questa afferma che le cose simili dovrebbero apparire e comportarsi allo stesso modo, mentre le cose diverse dovrebbero apparire diverse, questo è chiamato anche principio della "**minima sorpresa**", la coerenza consente all'utente di trasferire facilmente le proprie conoscenze esistenti a una nuova interfaccia utente. La consistenza è di tre tipi



Internal:
with itself



External:
with other UIs



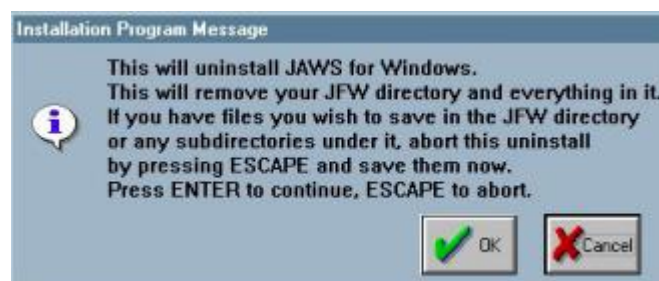
Metaphorical:
with the chosen
interface metaphor

La consistenza interna

Per la consistenza interna occorre utilizzare gli stessi termini per gli stessi concetti in tutta l'interfaccia, se i termini cambiano, l'utente si chiederà se si tratta di cose diverse. Per questo motivo scrivendo testi per UI, occorre:

- Essere chiari e coincisi
- Essere tecnici quando si tratta di argomenti all'interno del dominio applicativo
- non essere tecnici al di fuori dal dominio (specialmente nel gergo informatico)
- essere consistenti con le lingue e non mischiare inglese e italiano

Di seguito un brutto esempio di consistenza interna:



Qua si ha un'errore, in quanto all'interno del testo ci si riferisce ai bottoni "Ok" e "Cancel" con nomi diversi rispetto a ciò che viene riportato sopra al bottone

La consistenza esterna

E' importante utilizzare la **lingua dell'utente**. E' importante chiamare i comandi/concetti simili ad altre applicazioni con gli **stessi termini** (Salva, apri, chiudi, modifica, cronologia, ecc...), e come per la consistenza interna si devono utilizzare termini del dominio dove necessario ed evitare gergo tecnico informatico.



Le metafore

Le metafore sono **analogie** suggerite dal designer, si eseguono basando il modello dell'interfaccia su un corrispettivo reale, e l'utente definisce la soluzione al suo problema pensando agli operatori reali, è molto difficile mappare **tutto il sistema in una metafore unica**, infatti a tratti si rischia di rendere la metafora fuorviante, può succedere quando un utente prende la metafora alla lettera e si crea un modello concettuale sbagliato del sistema.

Una delle metafore più note è il desktop del computer, nel quale il video è la scrivania dell'utente, con i documenti, le cartelle ed il disordine. Le metafore vengono usate in **QUALSIASI** programma, dagli editor di testo, alle applicazioni per disegnare come può essere paint.

Le metafore non hanno solo aspetti positivi, infatti le critiche che vengono fatte a questa metodologia sono

- Comporta sempre delle forzature (Il cestino non è sopra la scrivania)
- Limitano le funzioni che potrebbero, in principio, essere utili nell'interfaccia
- La metafora può entrare in conflitto con principi di progetto
- Incapacità di comprendere il funzionamento del sistema oltre la metafora
- Rischio di ispirarsi ad oggetti fisici progettati male, ovvero poco usabili (e.g. calcolatrice)
- Limita l'immaginazione del progettista nell'escogitare nuovi modelli concettuali

Tutto questo detto riguardava il fattore di apprendibilità di un sistema.

5.4.4 Efficienza

L'efficienza è un altro dei tre aspetti fondamentali utilizzati per calcolare l'usabilità di un programma, programmare un sistema in modo efficiente significa **progettare in modo da far risparmiare risorse all'utente**, che possa essere risparmiare sulle azioni da svolgere, e quindi il **tempo**, le informazioni da tenere in memoria, quindi il **carico cognitivo**, oppure alleggerire lo sforzo di ragionamento.

Per fare questo si devono utilizzare le conoscenze relative alla percezione, alla memoria, al ragionamento ed al movimento che abbiamo affrontato precedentemente nei capitoli passati

5.4.5 Sicurezza (safety)

Ovvero il rimediare agli errori dell'utente, infatti non è possibile che gli errori non si verifichino, qualche svista capita quasi sicuramente, quindi il progettista deve sviluppare dei modi per **rimediare agli errori**. Per fare ciò l'utente deve capire cosa sia successo, poter tornare allo stato precedente, e poter fare molteplici tentativi.

Un bravo progettista deve cercare di evitare le sviste, quindi cercare di non nominare azioni con descrizioni simili, ed allontanare dalle altre operazioni quelle potenzialmente distruttive.

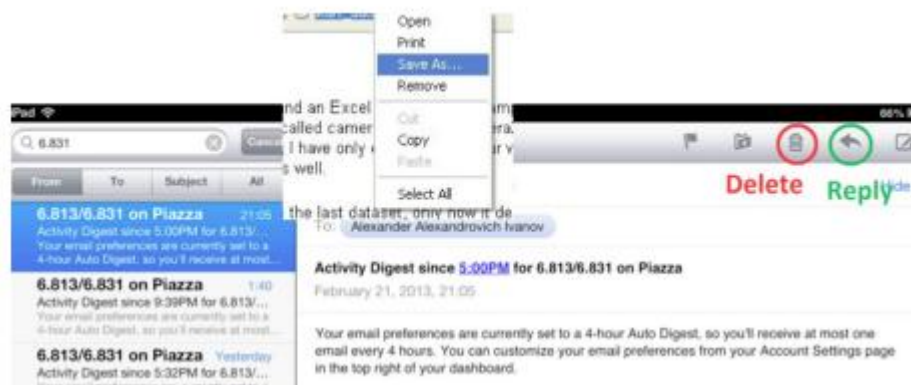
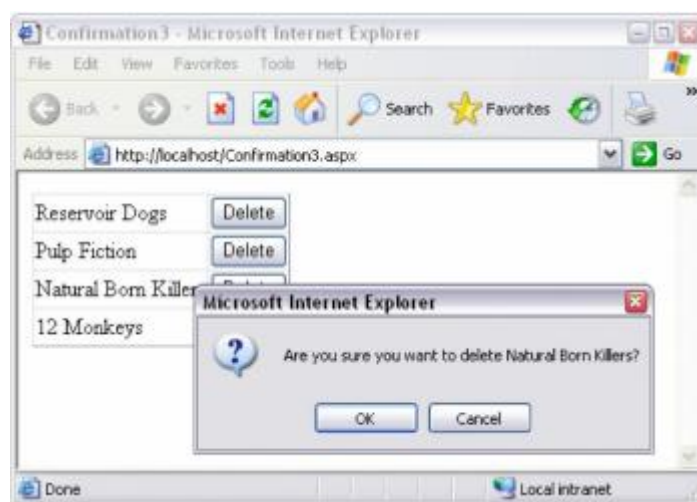


Figure 12: Cattivo esempio di progettazione svolta ad evitare sviste

Ma in caso l'utente sbaglia l'applicazione deve sempre **chiedere conferma per azioni pericolose**.



5.5 WIMP

E' un tipo di interfaccia ad oggi molto utilizzato, **WIMP** sta per

- **Windows.** Perché si possono scorrere, allargare, sovrapporre, aprire, chiudere e muovere sullo schermo delle finestre utilizzando il mouse
- **Icons.** Sono presenti icone rappresentanti applicazioni, oggetti, comandi e funzionalità che possono essere aperte quando vengono cliccate
- **Menus.** Sono presenti menu che offrono una lista di opzioni che l'utente può scorrere e selezionare
- **Pointing device.** Si utilizza un mouse che controlla un cursore, questo offre la possibilità di controllare le finestre, le icone ed i menu sullo schermo

Il WIMP è praticamente lo stile della totalità degli attuali PC, l'interazione avviene mediante **operazioni**(di puntamento, trascinamento, ecc..), che vengono effettuate dall'utente su **oggetti virtuali**, questo è adatto ad utenti con scarse conoscenze del sistema e una discreta conoscenza del compito, questa realizzazione è pratica della **manipolazione diretta**. Questa interpretazione ha diversi pro e contro

- Pro
 - Facile da imparare

- Le operazioni sono quasi sempre reversibili
- Flessibile
- Feedback istantaneo
- Intuitivo
- Contro
 - Può essere inefficiente
 - Occupazione di spazio video
 - Più costoso e complesso da programmare
 - In caso di interazione touch si incappa nel fenomeno dell'occlusione

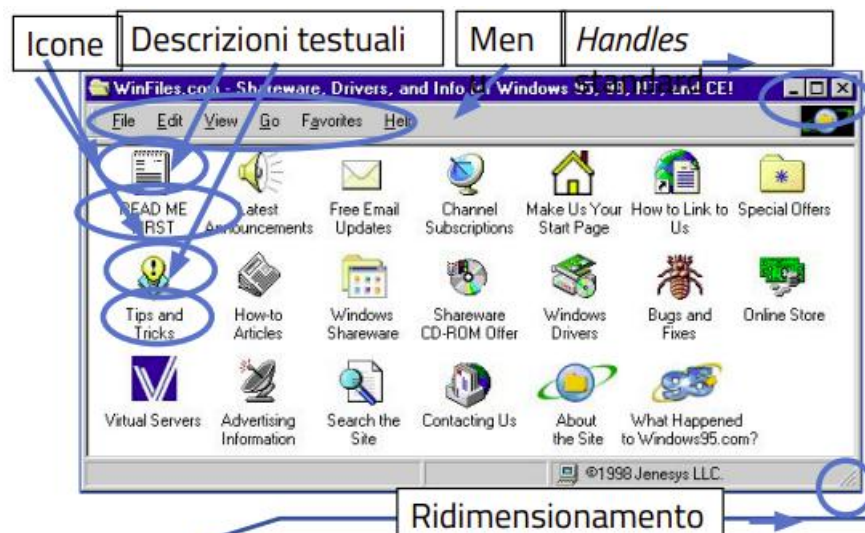
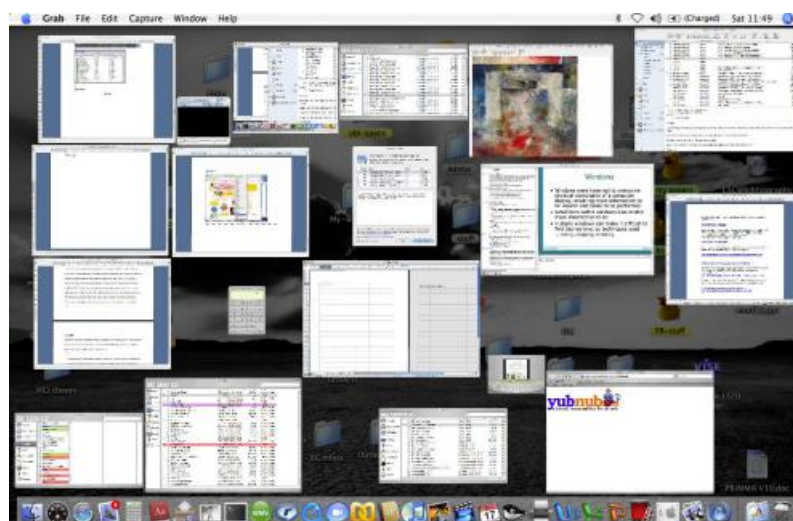


Figure 13: Gli elementi di una GUI si chiamano **widget**

5.5.1 Le finestre

Le finestre sono state inventate per superare i limiti fisici di uno schermo, ovvero per poter permettere di visualizzare più informazioni e di eseguire più task in contemporanea. Le scrollbar permettono di visualizzare più informazioni.



Utilizzando più di una finestra può essere difficoltoso trovare quella da utilizzare, per questo i sistemi a finestre usano delle tecniche per alleviare il problema, queste tecniche sono l'**elenco delle finestre**, la **riduzione ad icona** e la **overview**.

5.5.2 Menu

Un menu può essere di tre tipi **Piatto(flat)**, ad **espansione**, **contestuale**

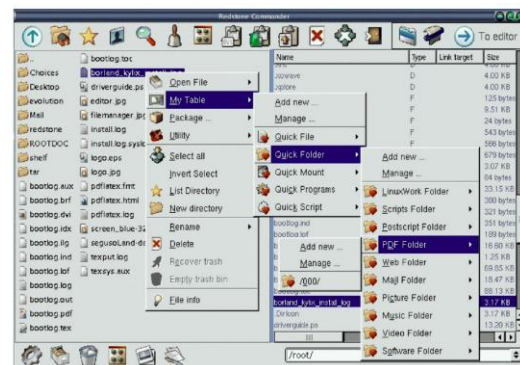
Flat menu

Il flat menu è buono per un piccolo numero di opzioni alla volta, si ha uno schermo di piccole dimensioni e spesso la lista di selezioni è innestata, è disposto su diverse schermate, e si procede tramite esse, per questo motivo tornare indietro può essere noioso



Menu ad espansione

Permettono di **visualizzare più opzioni** in una singola schermata rispetto a quelli flat ed inoltre più opzioni sono selezionabili nella stessa schermata, di conseguenza si ha una navigazione più flessibile. Questo però comporta il problema che la selezione possa risultare più complessa e che sia facile selezionare un'opzione sbagliata al crescere dei livelli.



Menu contestuali

Questi menu permettono di accedere alle **funzionalità utilizzate di frequente**, sono chiamati così perché l'insieme di funzionalità proposte varia a seconda del contesto, sono attivati principalmente in due modi: tramite il tasto destro del mouse, ed in caso di sistema Apple, con il click destro del mouse + CTRL. Questo metodo aiuta a mitigare il problema della navigazione dei menu a cascata, di conseguenza sarebbe meglio **non utilizzare più di uno (o due) livelli**



Scegliere le **etichette** da inserire nei menu sembra un problema banale, ma con una parola si deve descrivere un'intera funzionalità, queste parole inoltre devono essere pensate per l'utente e non per chi sviluppa il sistema. Un attributo molto importante delle etichette è il loro posizionamento nella lista, l'utente deve trovare **presto le funzioni che utilizza di frequente**, e cambiare la

posizione una volta che l'utente si è abituato può essere pericoloso. Per questo esistono moltissimi **standard**(ISO 9241) con linee guida sulla **profondità, struttura e navigazione**

5.5.3 Le icone

Le icone col tempo hanno subito una notevole evoluzione, osservando le icone di un sito web è possibile intuire infatti l'epoca di creazione, in quanto originariamente erano in bianco e nero, mentre adesso sono colorate, con effetti 3D, ombre, ecc..., alcune addirittura sono foto-realistiche. Le icone rendono **l'interfaccia piacevole alla vista**, ed è provato che un'applicazione con una bella grafica dia l'impressione di funzionare meglio.

La relazione fra un'icona ed il suo significato può essere basata su

- **similarità**(una foto per rappresentare un file di tipo immagine)
- **analogia**(un paio di forbici per indicare l'azione del taglio)
- **arbitraria**(l'uso di una X per indicare la cancellazione)

Di questi tre tipi le più efficaci sono quelle che utilizzano la similarità. La maggior parte delle operazioni sono azioni, il che le rende difficili da rappresentare, per questo si deve utilizzare una combinazione di oggetti e simboli per rappresentarle.

Le icone sono talmente diffuse che **difficilmente si parte da zero**, in quanto sono presenti linee guida, stili, icon builders e librerie. Spesso si possono utilizzare delle etichette insieme ad un gruppo di icone di piccole dimensioni per identificarle meglio.

5.5.4 Widget

I widget sono gli elementi individuali presenti all'interno della GUI(checkboxes, menu, buttons, ecc..), di questi elementi sono importanti tre aspetti:

Apparenza

Ovvero quale sia il loro aspetto, possono essere rappresentati tramite **Verbi**, ovvero parole che descrivono azioni(quit,exit,taglia,copia), tramite **Aggettivi**, parole che indicano una descrizione di un elemento o uno stato(bold,italic), oppure **nomi**, tipicamente utilizzati come descrizione(Times New Roman, US Letter)

Interazione Il modo tramite il quale si interagisce con un determinato widget



Semantica

Indipendentemente dal modo nel quale si costruisca l'interfaccia, ogni widget inserito ha un determinato scopo e compie una determinata funzione. La definizione della **semantica** di un oggetto

è di solito a **carico del progettista**, anche se i widget possono essere pre-impostati (collegamenti all'interno del database delle interazioni possibili), ed anche in questo caso il progettista decide a cosa sono collegati. Prima di scegliere il widget adatto per una determinata funzione, si pensa prima alla funzione che deve svolgere, e poi al come farlo.

In base allo scopo di un'azione si utilizzano diversi widget per rappresentarla, per widget che consentono di compiere **azioni**, di solito si usano i menu, i pulsanti o la toolbar. Per modificare **stati** o **selezionare opzioni** generalmente si usano checkbox, radio button e combo-box. Queste regole però non sono univoche.

Un bravo progettista tramite l'interfaccia suggerisce all'utente l'azione corretta da eseguire:



Inoltre in base al tempo che si ha fra l'azione ed il risultato che si vuole ottenere bisogna dare all'utente un Feedback sull'azione:

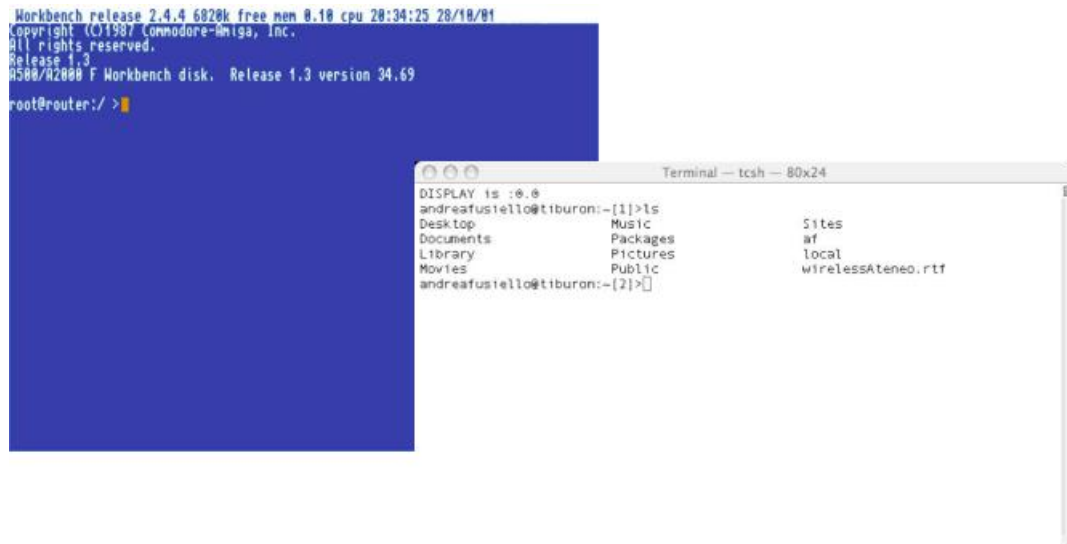
- **latenza < 0.1s** All'utente sembrerà istantaneo
- **0.1s < latenza < 1s** L'utente noterà un ritardo
- **1s < latenza < 5s** Bisogna mostrare all'utente il cursore d'attesa o una progress bar

5.6 Altri stili

Oltre all'interfaccia WIMP esistono altri stili di interfacce ad oggi utilizzati.

5.6.1 Interfaccia a linea di comando

Questa interfaccia è basata su una **serie di comandi**, solitamente abbreviazioni (e.g. ls), questi vengono inseriti tramite una shell di comando, una volta digitato il comando e i suoi argomenti il sistema risponde. Fanno parte di questa categoria anche i tasti funzione della tastiera e gli shortcut per eseguire una funzionalità. Per imparare ad utilizzare l'insieme dei comandi, è necessario molto tempo.



Nella scelta della sintassi dei comandi si deve prestare attenzione al **nome**, alla **forma** e alla **struttura**.

Il principio più importante da seguire è la **consistenza** (e.g. utilizzare nomi sempre senza vocali, utilizzare gli stessi nomi per i flag in diversi comandi, ecc...), questo renderà i comandi più "uniformi" e ne faciliterà la memorizzazione.

I prompt di comando dovrebbero sempre avere una funzione di **autocompletamento** ed una **history dei comandi**

5.6.2 Interfacce conversazionali

In queste interfacce l'utente e il sistema dialogano in linguaggio naturale, questo è utilizzato nelle chatbot e nelle interfacce vocali. Teoricamente non ci sarebbe nulla da imparare, in quanto si tratterebbe di una normale conversazione, ma la tecnologia è per ora limitante, quindi nella pratica si ha che un sistema non può capire tutto, di conseguenza l'utente deve capire cosa può o meno chiedere. Una funzione importante da aggiungere in queste interfacce è un suggerimento sul come riformulare una frase nel caso sia necessario.



5.6.3 Interfacce ad interazione vocale

Si ha quando una persona parla con un sistema che ha un'applicazione vocale (e.g. orario dei treni), questo viene utilizzato per richiedere informazioni molto specifiche, ovvero con un **dominio ristretto**, in quanto ne facilita il riconoscimento (e.g. Orari, numeri, luoghi, ecc..).

Questa funzionalità è utilizzata in modo particolare da **persone con disabilità**, tra gli esempi si hanno

- Word processors con funzione di dettatura
- Riconoscitori di scrittura
- Web e screen readers
- Controlli domotici

Uno dei problemi principali di queste interfacce sono gli errori che commette il sistema nella comprensione del linguaggio naturale umano e all'interpretazione della sua semantica. La vocal killer application è ancora il sistema di risposta automatica al customer care. Alcuni sistemi permettono all'utente di descrivere il problema con parole sue, e poi cercano di capire a quale area rimandare la chiamata, questo non è sempre affidabile. L'introduzione di questo tipo di interfacce sui dispositivi mobili ha sollevato il livello di questi sistemi.

Il sistema può controllare il dialogo vocale richiedendo domande specifiche ed ammettendo specifiche risposte.

sistema: Sceglie la busta uno, la due o la tre?
utente: la quattro!
sistema: Non ho capito. Sceglie la busta uno, la due o la tre?
utente: la tre!
sistema: ha selezionato la busta tre

Alcuni sistemi però sono più flessibili e permettono agli utenti di prendere l'iniziativa.

Le sfide di design di questa metodologia di interfaccia sono tantissime:

- Come fare in modo che il dialogo fra il sistema e l'utente rimanga nei binari previsti
- Aiutare gli utenti a navigare in modo efficiente nei menu vocali, facendo affidamento esclusivamente alla memoria a breve termine
- Aiutare gli utenti a rimediare gli errori
- Guidare chi fa richieste vaghe o ambigue
- Scegliere la tipologia di voce(uomo o donna, neutrali o con inflessioni dialettali), in quanto questa scelta comporta un impatto sulla pazienza degli utenti

5.6.4 Interfacce tramite realtà virtuale

La realtà virtuale può essere definita come una simulazione grafica generata al computer che fornisce l'illusione di una partecipazione in un ambiente sintetico piuttosto che l'osservazione di questo ambiente. Questo fornisce nuovi tipi di esperienza, facendo in modo che l'utente possa interagire con gli oggetti e navigare in uno spazio tridimensionale, questo crea delle esperienze molto coinvolgenti per l'utente.

La rappresentazione può avere un livello di fedeltà superiore a quello che si può raggiungere con il multimedia, introduce un senso di presenza nel quale l'utente si sente completamente immerso nell'esperienza. Il problema di questo metodo è che spesso si utilizza hardware che può risultare fastidioso, infatti gli head-mounter displays devono essere indossati e possono causare disorientamento e mal di mare.

Abbiamo molta ricerca su come creare interazioni di realtà virtuale realistiche e sicure(Flight simulators, applicazioni che aiutano gli utenti a superare le fobie). Si devono seguire diversi criteri nel design di queste interfacce:

- Come scegliere il **paradigma di navigazione nel mondo** (prima persona o terza persona)
- Come **controllare le interazioni ed i movimenti**(Controllers, movimenti della testa e del corpo)

- Come interagire con le informazioni (Utilizzo di tastiere, puntatori)
- Stabilire un livello di realismo per coinvolgere l'utente

La realtà aumentata utilizza rappresentazioni virtuali che sono sovrapposte ad oggetti fisici e dispositivi, nella realtà mista le viste del mondo reale sono combinate con quelle dell'ambiente virtuale. Si hanno innumerevoli applicazioni (medicina, giochi, volo, esplorazione).

6 Il processo di design dell'interazione

I fattori coinvolti nel design dell'interazione sono:

- L'importanza di coinvolgere gli utenti
- Il grado di coinvolgimento degli utenti
- Comprendere il funzionamento di un approccio incentrato sugli utenti
- Stabilire quattro attività base

Il primo fattore è molto importante, infatti bisogna eseguire una **corretta gestione delle aspettative dell'utente** durante la creazione di un programma (aspettative realistiche, nessuna delusione, garantendo un apprendimento tempestivo), infatti è molto importante rendere gli utenti parti attive del progetto, in quanto saranno più accondiscendenti, ed è un aspetto molto importante per il gradimento di un'interfaccia.

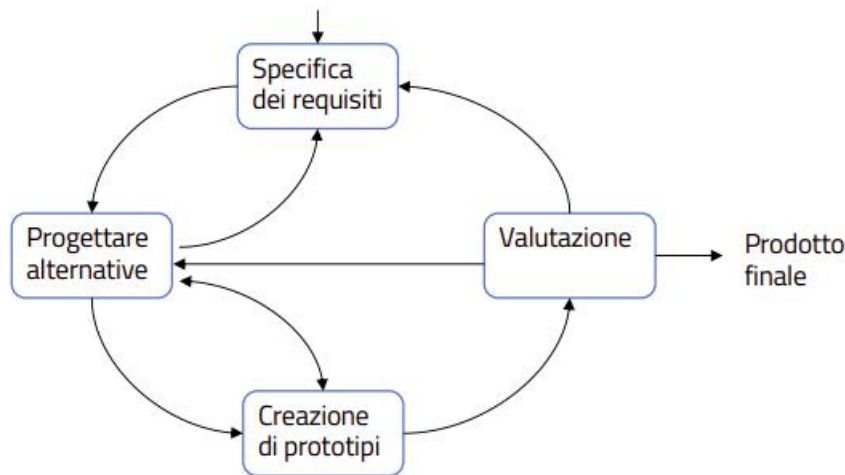
Prima di iniziare a progettare un'interfaccia bisogna stabilire chi sono gli utenti target, quali sono i loro bisogni, e come scegliere tra le alternative di sviluppo. Ma prima di tutto, partiamo dal definire cosa sia **l'interaction design**.

L'interaction design è un processo, un'attività guidata da un obiettivo, influenzata dall'utilizzo che si intende fare dell'applicazione, il dominio applicativo, il materiale, i costi e la fattibilità. Quindi si potrebbe dire che sia un'attività creativa, nella quale sia necessario un bilanciamento di diversi **trade-off** (compromessi).

Viene svolto tramite quattro approcci principali

- **User-centered design.** E' l'utente la guida, il designer progetta una soluzione in base ai suoi bisogni
- **Activity-centered design.** Si focalizza sul comportamento per l'esecuzione di un task
- **System design.** Progettazione rigorosa di un sistema che deve risolvere un problema preciso
- **Genius design.** Utilizza solo l'esperienza del designer

Un approccio user-centered ha la caratteristica di avere una concentrazione sui task delle prime fasi dello sviluppo, nelle quali avviene uno studio delle caratteristiche **cognitive, comportamentali ed attitudinali**. Si fa una misurazione empirica sulle prestazioni degli utenti nei scenari applicativi. L'approccio user-centered abbraccia il concetto di **design iterativo**, ovvero quando qualche problema viene rilevato durante un test utente, viene risolto e si eseguono altri test. Le fasi del processo di creazione sono:



I bisogni dell'utente

Un utente raramente sa cosa sia possibile, e di conseguenza anche cosa non lo sia, quindi non ci possono dire cosa gli sia necessario per raggiungere un certo obiettivo. Possono però concentrarsi sui task esistenti, ed apprendere il loro contesto ed il tipo di informazioni che necessitano. I task previsti dall'applicazione possono essere radicati sul comportamento corrente, o descritti come scenari futuri

6.1 Stabilire i requisiti

Per la fase di stabilimento dei requisiti bisogna seguire tre domande

- **Cosa.** Capire il più possibile sugli utenti, i task ed il contesto, e produrre un insieme stabile di requisiti
- **Come.** Eseguire un'attività svolta alla raccolta di dati e all'analisi di questi, tramite un processo iterativo
- **Perché.** Si tratta della specifica dei requisiti, la fase dove sono più comuni i fallimenti

I requisiti possono essere di differenti tipi

- **Funzionali**, ovvero stabilire cosa il sistema deve fare e storicamente sono il tipo nel quale ci si focalizza di più
- **Non funzionali**, come la dimensione della memoria ed il tempo di risposta
- **Dati**, che stabilisce il tipo di dati che deve venire salvato, ed anche il metodo di salvataggio
- **Ambiente o contesto d'uso** che può essere a sua volta
 - Fisico: polveroso, rumoroso, vibrazioni, luce, caldo, ecc..
 - Sociale: condivisione di file, display, carta, tra lunghe distanze, ecc...
 - Organizzazione: gerarchia, supporto agli utenti, struttura ed infrastruttura dell'organizzazione

Un requisito deve essere **chiaro, non ambiguo e specifico**, un esempio di un buon requisito è: «Il download di qualsiasi pagina del sito deve impiegare un tempo inferiore ai 5 secondi» mentre di uno cattivo: «Le bambine di età inferiore ai 10 anni devono ritenere che il sito sia attraente»

6.2 Le caratteristiche dell'utente

Bisogna stabilire innanzitutto chi sono gli utenti, stabilirne le loro **caratteristiche** (abilità, conoscenze, dimestichezza con i computers), **l'utilizzo del sistema** (inesperto, esperto, casuale, frequente), in caso l'utente sia inesperto le istruzioni vanno date passo-passo e con informazioni chiare. Se l'utente è esperto si ha più flessibilità, e avrà più "potere". Quando un'azione è frequente si devono impostare degli short-cuts per eseguirla, se invece un'istruzione è casuale si devono dare delle istruzioni chiare.

Le persone sono **varie sotto molti aspetti**, la **dimensione delle mani** può per esempio influire sulla posizione e la dimensione dei bottoni, le **abilità motorie** possono influire sulla possibilità di utilizzare certe periferiche di input o di output, **l'altezza** può influire sul design di un chiosco interattivo. Inoltre le persone possono anche soffrire di **disabilità**. Per questo bisogna considerare tutti questi fattori durante la creazione di un'interfaccia



6.2.1 Le personas

Le personas sono degli "esempi di utente reale", che **catturano le caratteristiche dell'utente**, non sono persone reali, ma hanno caratteristiche **sintetizzate** dagli utenti reali, per questo non devono essere **idealizzati** (non deve essere un utente modello). Devono avere un nome, diverse caratteristiche e scopi, conoscenze e attività sociali, per questo si sviluppano diverse personas

Steve McKenna è un Senior Developer che lavora per Accenture. Ha studiato Informatica per le Aziende all'Università di Sidney, in particolare modellazione dei processi e business intelligence.

Ha lavorato per molte compagnie ed ha molta esperienza nell'utilizzo di workflow e tool per la modellazione di processi come BPEL, BPML, BPMN.

Dopo anni di ricerche, crede che l'attività di consulenza sia il lavoro che fa per lui.



Dati personali

Steve sta lavorando ad un nuovo progetto di consulenza con una grossa compagnia di software, che vuole offrire la possibilità per le piccole aziende di usufruire dei servizi offerti dalla compagnia, selezionandoli e componendoli.

I servizi che vuole utilizzare sono già disponibili, deve essere solo offerta la possibilità di comporli e configurarli a seconda delle esigenze dei clienti tipici.



Obiettivi con l'applicazione

6.3 Parte laboratorio

6.3.1 I requisiti

I requisiti possono essere di diversi tipi

- **Funzionali** I requisiti funzionali sono quelli che indicano cosa il sistema deve fare, storicamente sono il tipo su cui si focalizza di più
- **Non funzionali** I requisiti non funzionali indicano requisiti più "tecnici", come la dimensione della memoria e i tempi di risposta
- **Dati** Questi requisiti indicano il tipo dei dati e il modo nel quale questi devono essere salvati
- **Ambiente o contesto d'uso** Questo requisito si divide in altri tre sotto-requisiti
 - **Fisico** Lo stile dell'applicazione (polveroso, rumoroso, vibrazioni, luce, ecc...)
 - **Sociale** Riguarda la parte social dell'applicazione (condivisione di file, display, privacy, lavoro individuale, ecc...)
 - **Organizzazione** La parte che identifica come è "gestita" l'applicazione (gerarchia, supporto agli utenti, struttura ed infrastruttura dell'organizzazione, ecc...)

Immaginando un'applicazione che segnala problematiche presenti nella strada, avremo come requisiti **Funzionali** il fatto che l'applicazione deve permettere la segnalazione con titolo e descrizione di un problema rinvenuto nelle strade cittadine, ecc.. un po' tutto ciò che deve fare l'applicazione e che l'utente vede, i requisiti **Non funzionali** invece possiamo avere per esempio che la dimensione dell'insieme dei file multimediali allegabili non deve superare i 10 MB, per i requisiti dei **Dati** avremmo che i dati salvati dall'applicazione saranno foto e video, e per i requisiti dell'**Ambiente/Contesto d'uso** sarà che l'area di segnalazione è limitata al comune che ha sviluppato l'applicazione, e che quest'app deve poter essere utilizzata in mobilità

6.3.2 Le personas

Le **personas** devono catturare le caratteristiche dell'utente, non sono persone reali, ma hanno caratteristiche **sintetizzate** dagli utenti reali, questi non devono essere idealizzati, ovvero non devono rappresentare un utente che non sbaglia mai. Le personas devono essere "vive", ovvero avere un nome, diverse caratteristiche e scopi, conoscenze e attività sociali. Generalmente si sviluppano diverse personas.

Gianni Zedda - l'autista di bus

Gianni Zedda ha 35 anni, sposato e lavora come autista presso la CTM. Ha conseguito il titolo di perito commerciale nel 2000, presso l'istituto Martini di Cagliari.

Lavorando come autista, ha a che fare ogni giorno con i problemi delle infrastrutture stradali (buche, cartelli divieti etc.).

Gianni è molto attivo sui social network, infatti col suo smartphone dà ogni mattina il buongiorno ai suoi contatti veri (ma non hai falzi!! 1). È un tifoso della Ferrari e la domenica mattina ama consultare CastedduOnline dal suo tablet.

Gianni, dato il suo lavoro, percorre spesso le stesse strade (anche più volte al giorno). Vorrebbe quindi che queste siano efficienti e sicure per lui e per gli automobilisti che come lui devono sopravvivere al traffico. Per questo ritiene che una comunicazione diretta con il comune possa aiutare a migliorare l'efficienza e la sicurezza delle strade cittadine.



Figure 14: Esempio di Personas riguardante l'app

6.3.3 I scenari

Gli scenari devono offrire una descrizione **informale e narrativa**, in modo che da questa descrizione possa essere colto il contesto, le necessità ed i requisiti della persona, non devono per forza prevedere la descrizione del ruolo del software (anche se spesso lo fanno). Una storia consente una più facile **comunicazione** tra le parti, il livello di dettaglio varia a seconda della fase e delle iterazioni di sviluppo raggiunte. Possono anche essere utilizzati per descrivere interazioni nel futuro, per esempio con supporti previsti ma non ancora esistenti.

Gianni, come ogni mattina, sta guidando il suo pullman lungo il percorso del 30. All'altezza di viale Marconi un sobbalzo scuote il bus: Gianni per l'ennesima volta prende in pieno una buca. Ha sempre lasciato correre il fatto, ma questa volta se ne ricorderà. Al momento non può però fermarsi per scattare una foto, tanto meno può incominciare ad utilizzare lo smartphone durante la guida per sfruttare la geolocalizzazione automatica. Gianni attende quindi la fine del turno per inviare la segnalazione, inserisce manualmente la posizione della buca ed invia il tutto con una breve descrizione. In questo modo spera di risolvere il problema il prima possibile in modo tale da non sentirsi più chiamato in causa dai vari pensionati che hanno fatto del "È colpa di Zedda" il proprio cavallo di battaglia.

Figure 15: Esempio scenario

6.4 Raccogliere i dati per i requisiti

Esistono tantissimi modi di acquisizione dei dati tramite gli utenti, ciascuno dei quali utile per motivi diversi:

- **Interviste.** Possono essere utilizzati dei supporti come prototipi o scenari di utilizzo, sono molto buone per indagare su problemi, ma richiedono molto tempo e può risultare impossibile avere un grosso campionamento di dati, in quanto si dovrebbero visitare tantissimi utenti
- **Focus group.** Sono delle interviste di gruppo, utili per raggiungere una visione consensuale o per trovare aree di conflitto, si rischia però che vengano dominate da alcuni individui
- **Questionari.** Sono spesso utilizzati insieme ad altre tecniche, possono fornire dati qualitativi o quantitativi e sono molto buoni per rispondere a domande specifiche utilizzando un buon numero di persone diverse
- **Ricerca prodotti simili.** Molto utile per individuare dei requisiti standard
- **Osservazione diretta.** Si ottengono informazioni sui task che coinvolgono le diverse parti, molto buona per capire la natura ed il contesto dei task, però richiedono tempo e impiego di un membro del team di design, questo inoltre produce un'enorme quantità di dati
- **Osservazione indiretta.** Non è molto utilizzata per i requisiti, ma è buona per tenere traccia dei task correnti
- **Studio di documentazione.** Le procedure e le regole generalmente sono scritte nei manuali, una buona sorgente di dati sui passi coinvolti su un'attività e su tutte le leggi e regole che governano un dato task possono essere estrapolate dalla lettura di questi manuali. Non possono però essere utilizzati come unica fonte di raccoglimento dei dati. Sono buoni per capire la legislazione e per avere informazioni di contesto, e non hanno bisogno del tempo di nessuno degli utenti o dei soggetti coinvolti nell'applicazione, che è un fattore che limita gli altri metodi.

6.5 Analisi dei dati

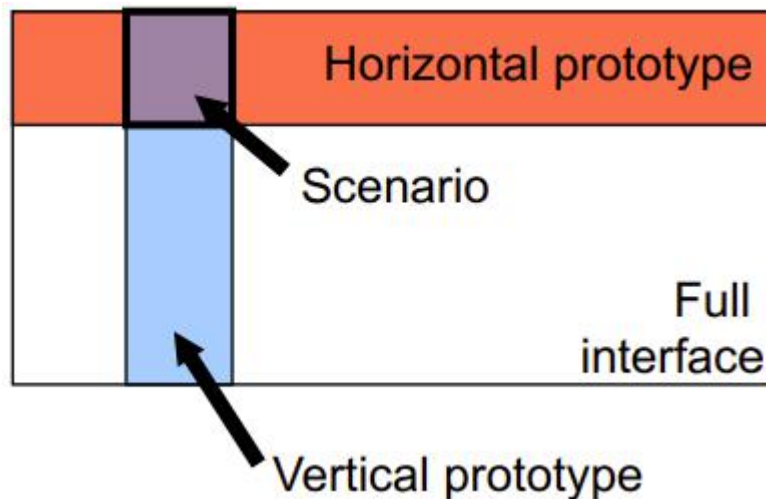
Dopo la raccolta di dati, questi devono essere analizzati, c'è sempre una fase di interpretazione iniziale prima di una analisi più dettagliata. Esistono diversi formati per diversi tipi di requisiti.

6.6 Creazione di prototipi

Un prototipo è una sorta di modello che rappresenta ciò che il prodotto dovrebbe diventare, tralasciando alcuni aspetti della realizzazione (soprattutto dal lato estetico). Questo modello è costruito con strumenti meno costosi e che permettono una veloce realizzazione del progetto, in modo da poter essere anche essere mostrato ad altri. Un prototipo si crea per aiutare il ragionamento, infatti è molto più facile ragionare su qualcosa di pratico rispetto che su qualcosa di astratto; è utile anche per permettere di notare e correggere dei problemi del nostro progetto e per capire i pregi/difetti di una soluzione. I prototipi sono molto importanti per ottenere buone interfacce tramite valutazione e feedback dell'interfaccia stessa, inoltre è utile anche per i programmatori avere un modello di

riferimento per migliorare la comunicazione del team di sviluppo. I prototipi che vedremo noi sono quello a **bassa fedeltà** e quello ad **alta fedeltà**. Nella creazione di un prototipo si utilizzano due compromessi:

- **Verticali.** Fornire un alto livello di dettaglio per poche funzionalità
- **Orizzontali.** Fornire un ampio insieme di funzionalità ma con pochi dettagli



Ora vediamo quali sono i **mezzi per la creazione di prototipi**

- Utilizzare carta e penna per creare un prototipo su carta, questo metodo è molto utilizzato per le interfacce.
- Si possono utilizzare anche le storyboard, ovvero una serie di scene simili a quello di un cartone animato o di un fumetto
- Sono molto utilizzate anche le presentazioni Power Point
- Video che simulano il funzionamento dell'applicazione
- Un software con funzionalità limitata, scritto in qualsiasi linguaggio

In un prototipo bisogna filtrare qualche informazione o particolarità dell'applicazione, in genere si filtrano:

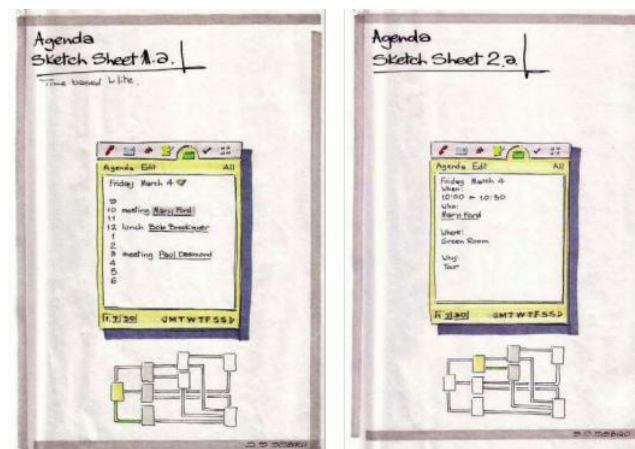
- L'aspetto
- La mole di dati
- La funzionalità
- L'interattività
- La struttura spaziale

Si prototipano in generale le cose che potrebbero essere causa di problemi tecnici, il flusso di lavoro, la sequenza di azioni per completare un task, il layout delle schermate, le informazioni da visualizzare e le aree dove si riscontrano difficoltà o criticità.

6.6.1 Prototipo a bassa fedeltà

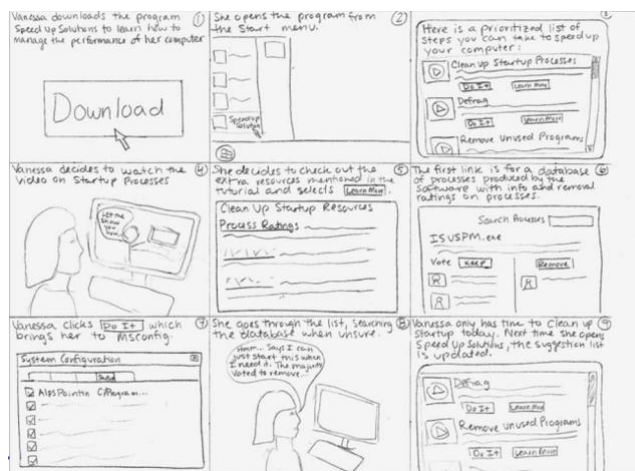
Schizzi delle schermate

Con questa metodologia vengono disegnate le schermate su carta, tralasciando aspetti grafici. I vantaggi di questo metodo è che non è necessaria una grande abilità manuale, e questi schizzi possono essere modificati facilmente, infatti questa modalità è meglio effettuarla su carta, in quanto non si ha nessun limite che altrimenti si potrebbe avere effettuandolo sul digitale tramite un programma. Con questo metodo quindi si rappresenta graficamente un'interazione, tramite la quale si visualizza il contesto, ed è inoltre utile in quanto è possibile annotare e scrivere dei dettagli



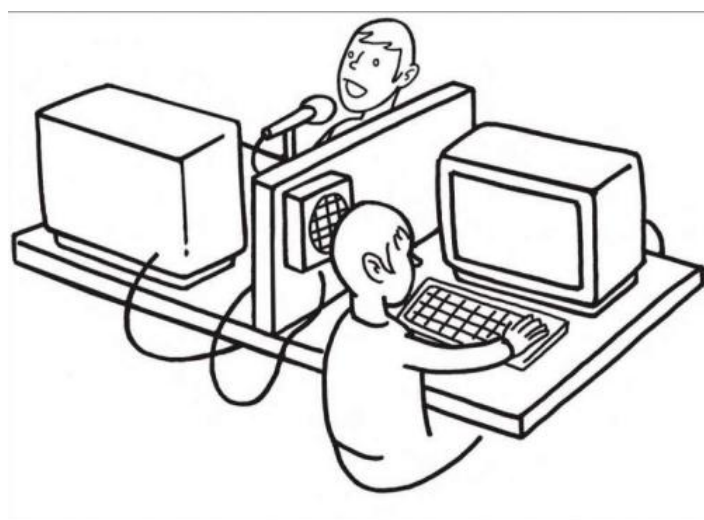
Storyboard

Sono spesso utilizzati insieme agli scenari, introducendo più dettagli e possibilità di interpretare un ruolo preciso, questo viene svolto tramite una serie di schizzi che mostrano come un utente progredisca attraverso un task utilizzando un dispositivo. Sono utilizzati nelle fasi iniziali del design, e rappresentano "stati" dell'interazione. Con questa metodologia non vengono rappresentate le transizioni da uno stato all'altro, e risulta difficile rappresentare situazioni dinamiche.



Mago di Oz

In questa metodologia si finge che il sistema esista e si usa qualche trucco per permettere l'interazione prima dell'implementazione, l'utente pensa di interagire con un sistema reale, mentre in realtà il risultato delle operazioni viene deciso da uno sviluppatore o tramite qualche altro espediente. Solitamente è fatto nelle fasi iniziali per capire cosa l'utente si aspetti dal sistema. Questo può però avere problemi relativi ai tempi di risposta (Per esempio il caso nel quale un umano cerca l'informazione richiesta anziché il sistema)



6.6.2 Prototipi ad alta fedeltà

Il prodotto ad alta fedeltà utilizza materiali che ci si aspetta siano utilizzati nel prodotto finale, questo prototipo assomiglia di più alla versione finale del sistema più che ad una bozza, generalmente

per creare questo tipo di prototipi si utilizza un vero e proprio linguaggio di programmazione. Con questa metodologia si rischia che gli utenti pensino di avere fra le mani un sistema completo. Il prodotto viene costruito dalla consultazione dei prototipi in modo da crearne un tutt'uno, prestando attenzione alla qualità. Quindi il prototipo deve essere ingegnerizzato, ovvero ci dev'essere un'evoluzione dei prototipi, quindi devono essere prototipi "usa e getta"

6.7 Le 10 euristiche per l'usabilità

Sono dieci regole da tenere a mente durante la creazione di un'interfaccia, sono state scritte da Jakob Nielsen, e quelle che vedremo noi utilizzabili sul web sono state riscritte da Theresa Neil

6.7.1 Visibilità dello stato del sistema

Il sistema dovrebbe sempre **tenere l'utente informato su cosa stia succedendo**, con un feedback appropriato in un tempo ragionevole



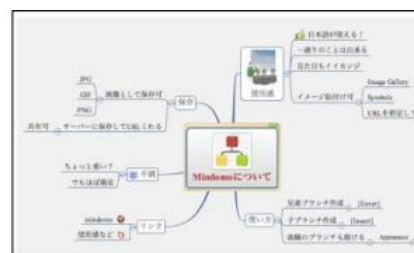
6.7.2 Corrispondenza fra sistema e mondo reale

Il sistema dovrebbe parlare la **lingua dell'utente**, con parole, frasi e concetti che siano familiari per l'utente, più che con parole familiari per il sistema.

Si devono seguire **convenzioni legate al mondo reale**, facendo in modo che le informazioni appaiano in un ordine logico e naturale.



Metafora della libreria dei propri contenuti



Rappresentazione di una mindmap (molto meglio di una lista puntata)

6.7.3 Libertà e controllo per l'utente

Gli utenti spesso scelgono le funzionalità del sistema per errore. Saranno dunque necessarie delle **uscite di sicurezza** per lasciare quello stato non voluto senza dover andare in fondo ad una lunga interazione. Supportare anche l'**annullamento** e la **ripetizione delle azioni**

	A	B	C	D
1	Item	Quantity	Price	Total
2	Tacos	40	\$5.0	=B2*C2
3				

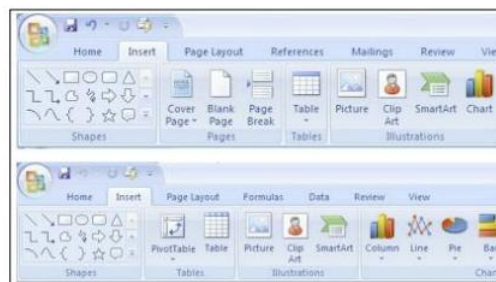
Facile annullare l'editing

6.7.4 Consistenza e standard

Gli utenti non devono stare a pensare se parole differenti, situazioni o azioni significhino la stessa cosa. Seguite le **convenzioni della piattaforma** su cui sviluppate.



GMail ha mantenuto la terminologia degli altri lettori email, anche se è nato dopo



I menu delle diverse applicazioni di office sono consistenti come terminologia per funzioni equivalenti

6.7.5 Prevenzione degli errori

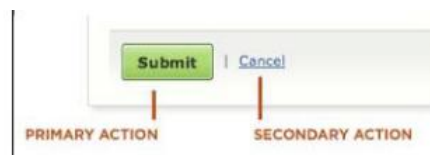
Meglio di un buon messaggio d'errore c'è un buon design che previene il presentarsi di un problema dal primo momento, quindi bisogna **eliminare le condizioni che inducono facilmente in errore** e presentare all'utente una opzione per confermare prima che rendano l'azione definitiva.



Disabilitare il bottone per l'upload mentre l'operazione è in esecuzione



Previene errori di ortografia



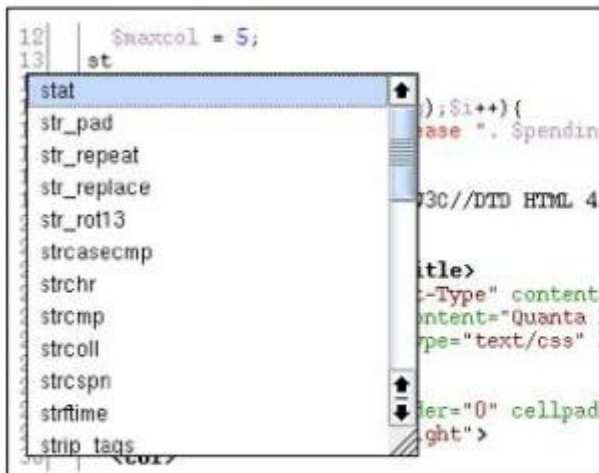
La funzione più frequente è quella accessibile con più facilità



Autofocus per evitare che l'input venga perso

6.7.6 Riconoscimento piuttosto che memoria

Minimizzare il carico della memoria dell'utente facendo in modo che azioni, oggetti ed opzioni siano visibili. L'utente non dovrebbe memorizzare informazioni tra due parti diverse del dialogo con l'interfaccia. Le istruzioni per utilizzare il sistema dovrebbero essere visibili o facilmente individuabili quando appropriato.



Autocompletion per la sintassi di un linguaggio di programmazione



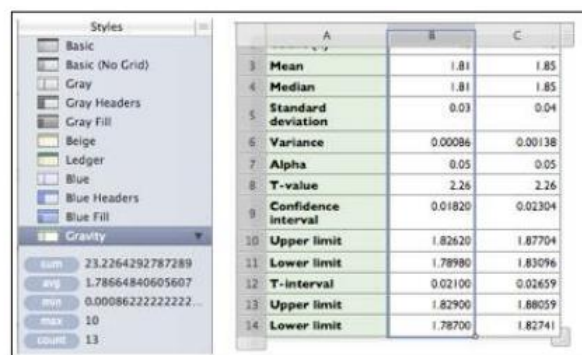
Riconoscimento di un font dalla forma e non dal nome

6.7.7 Flessibilità ed efficienza di utilizzo

Le combinazioni di tasti per i comandi, spesso invisibili all'utente inesperto, possono velocizzare l'interazione per gli esperti ed il sistema può essere utilizzato meglio da entrambi. Quindi bisognerebbe prevedere la possibilità di inserire azioni personalizzate tramite combinazioni di tasti.



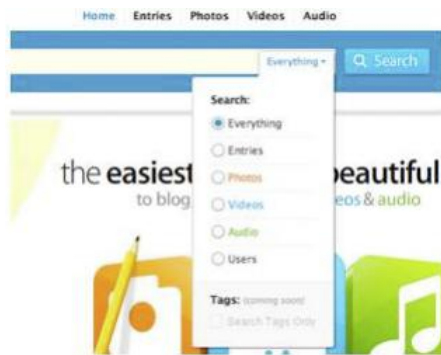
Elenco di shortcuts



Un elenco di azioni frequenti è visualizzato sulla sinistra della colonna selezionata per velocizzare l'interazione

6.7.8 Estetica e design minimale

Le presentazioni non dovrebbero contenere delle informazioni che non sono rilevanti o che sono utilizzate raramente, ogni unità extra di informazione nella presentazione compete con quelle rilevanti e ne diminuisce la visibilità.



Contrasto: le etichette sono visualizzate in grassetto
Ripetizione: I media types hanno colori diversi
Vicinanza: gli elementi sono separati da una linea leggera

Timesheet for Theresa Neil

04 May 2009 - 10 May 2009

	Mon May 04	Tue May 05	Wed May 06	Thu May 07	Fri May 08	Sat May 09	Sun May 10	Total
01					6.00			6.00
02					2.50			2.50
03								
04					4.00			4.00
05					1.00			1.00
06					1.00			1.00
07					4.50			4.50
08					1.00			1.00
09				1.50	1.00			2.50
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
Total	10.00	6.00	7.00	6.00	6.50	2.00		40.50

Una distanza adeguata fra le colonne è sufficiente per visualizzare il cartellino
 Le intestazioni hanno un colore diverso

6.7.9 Aiutare l'utente a riconoscere ed a rimediare agli errori

I messaggi di errore devono essere espressi in **linguaggio comune** (senza codici). Occorre quindi indicare precisamente il problema e suggerire una soluzione.

Or start a new account

Choose a username (no spaces)
 bert

Choose a password

Retype password

Email address (must be real)
 not an email

☒ Send me occasional Digg updates.

▲ bert is already taken. Please choose a different username.

▲ Passwords must be at least 6 characters and can only contain letters and numbers.

▲ The email provided does not appear to be valid.

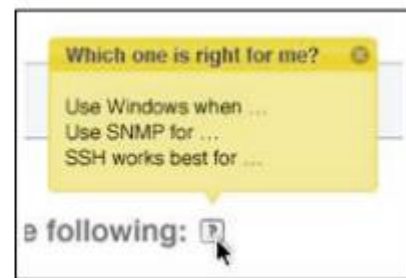
Feedback immediato vicino al campo compilato



Pagina umoristica per una risorsa non trovata. Allo stesso tempo c'è anche l'indicazione di una lista di articoli e la possibilità di navigare il resto del sito

6.7.10 Documentazione e aiuto

Anche se sarebbe meglio che un sistema fosse utilizzabile senza documentazione, può essere necessario mettere a disposizione dei documenti di help. Le informazioni contenute dovrebbero essere **facili da cercare**, localizzare sulle attività dell'utente ed elencare i passi da eseguire senza essere troppo lunghe.



7 Valutazione

Ora non resta da capire come si può valutare l'usabilità.

Come già detto l'usabilità è "la misura in cui un **prodotto** può essere usato da specifici **utenti** per raggiungere specifici **obiettivi** con efficacia, efficienza e soddisfazione, in uno specifico **contesto d'uso**". La valutazione è divisa in vari step :

- **ci si pone un obiettivo.** Si risponde ad una domanda **cruciale** sul prodotto che si vuole valutare.
- **Si raccolgono i dati.** Ponendo delle misure, e individuando un metodo appropriato di raccolta dei dati.
- **Bisogna considerare gli utenti.** Questo può essere fatto in modo **diretto** tramite test con utenti(occorre scegliere un insieme rappresentativo, ed un numero adeguato di utenti), oppure in modo **indiretto** effettuando test senza utenti(tramite leggi, modelli, assunzioni di provata efficacia, o tramite euristiche e linee guida)
- **Analizzare i dati** tramite statistica descrittiva o statistica inferenziale

Gli obiettivi della valutazione principalmente sono 3:

- Raggiungere un determinato obiettivo, ovvero stabilire un' obiettivo, scegliere una metrica per un dato compito, e si cerca di migliorare l'interazione fino al raggiungimento dell'obiettivo(e.g. stabilire l'obiettivo: voglio che il tempo medio per terminare un compito sia di 2 minuti, allora la metrica sarà appunto il tempo, e il miglioramento dell'interazione invece diminuire il tempo).
- Confrontare **diverse soluzioni** di interazione, in modo da poterle confrontare fra loro, tramite test statistici si possono stabilire differenze significative che valgono per la generalità degli utenti.
- **verificare l'esperienza** in modo da ottenere delle informazioni "qualitative".

La valutazione è effettuata riguardo ai problemi principali di usabilità da prevenire, gli aspetti del prodotto che funzionano bene per gli utenti, quali sono gli aspetti frustranti, quali sono gli errori comuni che gli utenti commettono e se sono stati fatti dei miglioramenti tra due iterazioni di design. La valutazione spesso coinvolge degli utenti, ovviamente è impossibile osservarli tutti, quindi se ne analizza soltanto un **campione**. I due aspetti più importanti sono l'**affidabilità**(facendo il test con altri utenti si dovrebbe ottenere lo stesso risultato), e la **validità**(il test deve misurare qualcosa di importante, ed offrire indicazioni su ciò che è di interesse). Le tipologie di valutazione possono essere due:

- **Formative evaluation.** Supporta la progettazione con informazioni utili per il miglioramento, ed è eseguita durante la fase di sviluppo
- **Summative evaluation.** Viene eseguita quando il prototipo è avanzato, è utile per la convalida delle qualità e per la formazione delle strategie di vendita.

Invece i metodi di valutazione sono diversi:

- Basata su modelli: si utilizza una descrizione astratta dei sistemi, analizzando l'applicazione dal punto di vista dell'efficienza ed efficacia(Come la legge di Fitts)
- Basata sull'osservazione: si osservano gli utenti, questo può essere fatto con diverse tipologie(laboratorio, sul campo, da remoto, ecc..)
- Basata sul feedback: Si chiede agli utenti la loro opinione tramite interviste, questionari, focus group e diari
- Basata sull'ispezione: l'analisi viene effettuata da esperti, questa metodologia è effettuata senza la necessità di utenti

7.0.1 Metodi basati su feedback

La prima fase è quella di preparazione, in questa fase si fanno diversi passaggi:

- Si stabilisce l'obiettivo del test
- La durata
- Il supporto HW/SW necessario
- Le condizioni iniziali
- Il tipo di utenti
- Il numero di utenti
- I compiti da svolgere
- I criteri di successo



La seconda fase è quella di introduzione:

- Si stabiliscono quante e quali informazioni fornire agli utenti
- Quali informazioni di tipo demografico chiedere
- Eventuali informazioni di contesto
- A volte può essere necessario un periodo di formazione preliminare



La terza fase è quella di esecuzione:

- Si osservano gli utenti mentre svolgono i compiti
- Si offre supporto al mediatore
- Si effettua la raccolta dati



La quarta ed ultima fase è quella di debriefing:

- E' la fase opzionale è quella in cui si ascoltano gli utenti dopo l'esecuzione del test
- E' utile per evidenziare ragioni e motivazioni per alcuni problemi
- Altrimenti si possono raccogliere suggerimenti o idee



Per stabilire l'ammontare di utenti necessari per effettuare una valutazione bisogna valutare diversi aspetti. Infatti dipende da ciò che si deve fare. Si dice che siano sufficienti 5 utenti per trovare l'**80%**, mentre 20 utenti per il **95%** dei problemi. Generalmente **12 utenti** è considerato un buon numero (mentre 30 è il numero magico dei corsi di statistica). Nella valutazione occorre fare in modo che l'utente sappia che non è lui ad essere valutato ma l'applicazione, e nel caso compia degli errori non deve sentirsi frustrato. I partecipanti hanno diritto a sapere cosa verrà chiesto, e ad andarsene se non ne hanno più voglia, in genere è buona norma offrire una ricompensa.

7.0.2 Misure quantitative e informazioni qualitative

Una **metrica** è un attributo misurabile quantitativamente, solitamente è più semplice raccoglierle in ambiente controllato(ponendo delle regole, dei "paletti"), si può valutare in modo quantitativo anche una dimensione qualitativa, basta infatti proporre una scala(e.g. indicare la propria soddisfazione da 1 a 5: 1 per nulla soddisfatto, 5 totalmente soddisfatto). Tra le misure quantitative tipiche abbiamo:

- **Tempo impiegato** per svolgere un compito
- **Numero di compiti** in un intervallo di tempo
- **Completion Rate**, ovvero il rapporto tra utenti che completano il compito e quelli che non ci riescono
- **Numero di errori**
- **Numero di compiti** svolti dall'utente
- **Numero di compiti** che non sono stati completati dall'utente perché non trova una soluzione
- Il rapporto tra utenti che scelgono la **strategia ottima** e quelli che non la scelgono
- **Tempi di attesa** utilizzati dall'utente per capire il suo compito

Le informazioni qualitative

Nelle informazioni qualitative non si collezionano rappresentazioni numeriche di grandezze, ma bensì si prende nota di difficoltà, piani di soluzione di un problema, interpretazione di messaggi e ragionamenti, con questo metodo si cerca di colmare il divario fra il modello del progettista e quello dell'utente. Si può mescolare la raccolta di dati qualitativi con dati quantitativi ed utilizzarli per spiegare dei fenomeni. Esistono tre metodi per effettuare la valutazione qualitativa

- **Think Aloud.** L'utente dice ad alta voce quello che pensa durante l'interazione, questo viene utilizzato per cercare di capire come l'utente percepisca l'interfaccia, l'utente potrebbe modificare il suo modo di interagire perché deve parlare.

- **Retrospective testing.** Si guarda con l'utente il filmato dell'interazione, e l'utente fornisce una spiegazione della motivazione delle sue azioni.
- **Coaching.** Il valutatore fornisce dei suggerimenti all'utente, in questo modo si aiuta e si capisce ciò di cui necessita.

Le valutazioni potrebbero invece essere basate su feedback, quindi esistono altre quattro metodologie di raccolta dati

- **Focus group.** Sono riunioni di 6-8 persone, con un moderatore, si stimola la discussione per evidenziare necessità e problemi
- **Interviste.** Possono essere **Non strutturate**(conversazione tra valutatore e utente, la discussione non è prestabilita) **Semi-strutturate** (prima un questionario e poi discussione libera) **Strutturate**(Un questionario per guidare la discussione)
- **Diari.** L'utente annota i risultati di diverse attività, questo può essere utile per studi di lungo termine, e possono essere supportati da qualche strumento automatico.
- **Questionario.** Sono il più diffuso metodo di valutazione basato su feedback, si basano su un insieme di domande da somministrare all'utente, possono essere effettuati dopo tutto l'esperimento(post-test) o dopo ogni task(post-task)

7.1 Questionari

Nei questionari le domande vanno numerate univocamente, la somministrazione deve essere accompagnata da istruzioni per la compilazione, è importante alternare le domande con accezione positiva e quelle con accezione negativa(soprattutto per scala Likert). Facendo questo però bisogna sempre lasciare il giudizio positivo sempre come estremo sinistro(o destro) della scala, ma non alternare.

All'interno dei questionari è importante raggruppare le domande che trattano temi o caratteristiche simili, in modo da focalizzare l'attenzione. Possono verificarsi dei diversi risultati in base all'ordine delle domande, per esempio chiedere prima i difetti e poi i giudizi sull'UX può spostare al negativo le risposte, quindi queste due parti devono essere alternate in modo casuale per i partecipanti.

7.1.1 La creazione di un questionario

Per la creazione di un questionario esistono questionari standard che offrono tecniche consolidate per richiedere le informazioni agli utenti, spesso però non sono adatti per ottenere informazioni su particolari peculiari della nostra interfaccia. Per questo motivo ci si pone il problema di come costruire un questionario ex-novo, oppure di "completarlo" per le parti necessarie.

Durante la creazione di un questionario bisogna fare attenzione a diversi fattori. I risultati dipendono da ciò che l'utente ricorda dell'esperienza, quindi è meglio farli subito. Inoltre dipendono dalla capacità dell'utente di **analizzare e descrivere il problema**. Le risposte ai questionari spesso tendono a ricercare il consenso sociale, di conseguenza è possibile che dicano quello che vorremmo sentirci dire, anche se non lo pensano davvero. Quindi è importante il modo nel quale le domande sono poste, spesso è bene alternare tra richieste positive e negative, e spesso è utile utilizzare delle forme ridondanti per essere sicuri(chiedere due volte la stessa cosa ponendo diversamente la domanda).

7.1.2 Diversi tipi di domanda

Le domande categoriali, sono le domande nelle quali la risposta è data scegliendo tra un numero limitato di risposte, a volte è possibile anche avere più risposte per la stessa domanda

Domanda 1: Hai mai utilizzato un foglio di «calcolo»?	SI	NO
-------------------------------------------------------	----	----

Domanda 2: Quali dei seguenti strumenti hai usato nelle ultime due settimane? (È possibile selezionare più di una risposta)	
X	Foglio di calcolo su computer portatile
X	Foglio di calcolo su computer desktop
	Foglio di calcolo su tablet
	Foglio di calcolo su smart phone
	Foglio di calcolo condiviso su rete Internet
	Nessuno tra quelli indicati

La scala di risposta, in questa metodologia di domanda, sono rappresentati diversi livelli pre-definiti di un'unica dimensione (assunta continua), in questo tipo di questionari si deve scegliere se inserire la posizione media di scelta (e.g. 3 su 5) oppure se forzare l'utente a prendere una posizione negativa o positiva. Generalmente si utilizzano scale **Likert** che vanno da 1 a 5 o da 1 a 7.

Domanda 18: Ho trovato piacevole spedire messaggi							
Completamente in disaccordo	1	2	3	4	5	6	Completamente d'accordo

Domanda 21: La funzione di «spedizione messaggi» è								
Indispensabile	1	2	3	4	5	6	7	Superflua
Discreta	1	2	3	4	5	6	7	Invasiva
Curata	1	2	3	4	5	6	7	Trascurata

Le risposte aperte, nascono, in quanto, mentre le domande chiuse hanno il vantaggio di tradurre un giudizio in un formato numerico, e quindi agevolare un'analisi numerica, limitano la libertà dell'utente. Di conseguenza è possibile utilizzare domande a risposta aperta per non limitare questa possibilità. Questo tipo di domanda ha lo svantaggio di non poter essere analizzata con strumenti numerici però. Le risposte chiuse e aperte, possono essere utilizzate in combinazione tra di loro per chiedere maggiori motivazioni, oppure per prevedere altre opzioni meno tipiche che possono essere state trascurate (e.g. il campo *altro*).

Domanda 10: Per quali motivi hai utilizzato il computer nelle ultime settimane?

Ho utilizzato il computer nelle ultime settimane per preparare queste slide...

Domanda 5: Per quali motivi hai utilizzato il computer nelle ultime settimane? (È possibile selezionare più di una risposta)

X	Vedere foto o video
X	Scambiare posta elettronica
	Chattare/istant messaging
	Fare acquisti
X	Scrivere documenti di testo
	Altro (specificare) <i>Guardare la partita del Cagliari in streaming (ovviamente con mySky)</i>

Ranking, questo tipo di domanda richiede all'utente di creare una classifica tra categorie di risposta, ordinandole secondo un certo criterio. Per esempio si può chiedere all'utente di ordinare diversi criteri (di usabilità e non) per una data interfaccia, in questo modo si scopre quali sono quelli prioritari.

Domanda 13: Quali tra le seguenti caratteristiche sono necessarie in un sistema di cloud computing?

Selezionare gli item presentati in ordine di importanza, attribuendo valore 1 al più importante e 6 al meno importante

6	Velocità di accesso
4	Sicurezza
1	Costo
3	Capacità
2	Interfaccia grafica usabile
5	Piacevolezza d'uso

7.1.3 Diversi tipi di questionari

I questionari possono essere creati anche on-line, infatti esistono diversi tool online che facilitano la compilazione ed i capricci per i questionari, il vantaggio di ciò è il basso costo, la velocità fra compilazione e codifica, la facilità di diffusione fra utenti anche molto diversi e lontani, e la possibilità di ricerche cultural-dependent. Gli svantaggi invece sono i Bias(errori sistematici), per il quale si potrebbe avere un campione non rappresentativo, non tiene conto del digital-divide fra le diverse categorie(maschi/femmine, adulti/giovani, ecc..), la difficoltà nel controllare le caratteristiche e la serietà di chi lo compila, e la difficoltà di identificazione di atti di "sabotaggio". Inoltre molto spesso gli utenti non lo compilano potendo scegliere.

Esistono diversi questionari che sono stati creati per poter essere usati in diversi contesti, sono stati oggetto di studi cognitivi che hanno provato l'efficacia, in base al testo e all'ordine delle domande. Le metriche per la valutazione e la comparazione in oltre sono ben definite, di conseguenza sono riconosciuti come metodi validi per la misura dell'usabilità, tuttavia la loro generalità li rende non adatti per valutare parti peculiari di una soluzione. I vantaggi di questi questionari sono:

- **Obbiettività:** permette la verifica da parte di altre persone dei risultati ottenuti
- **Replicabilità:** Nel caso si voglia ripetere lo studio in ambienti diversi, si può fare utilizzando lo stesso test
- **Quantificazione:** Permettono di effettuare confronti numerici con altri studi tramite tecniche statistiche
- **Economia:** Vengono abbattuti i costi della creazione del questionario
- **Comunicazione:** è più facile capire il significato delle misurazioni se si usa una scala comune
- **Generalizzazione:** data la validità del questionario, è possibile generalizzare i risultati

7.2 Questionario SUS

I questionari post-test sono quelli che vengono somministrati alla fine dell'intera sessione di test, ne esistono diversi che hanno vantaggi e svantaggi. Tullis e Stelson hanno dimostrato che nel caso di un limitato numero di utenti, per la misurazione dell'usabilità generale sia da preferire il test **Software Usability Scale(SUS)** creato da Brooke nel 1996.

The System Usability Scale Standard Version		Strongly disagree	1	2	3	4	Strongly agree	5
1	I think that I would like to use this system.		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
2	I found the system unnecessarily complex.		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
3	I thought the system was easy to use.		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
4	I think that I would need the support of a technical person to be able to use this system.		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
5	I found the various functions in the system were well integrated.		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
6	I thought there was too much inconsistency in this system.		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
7	I would imagine that most people would learn to use this system very quickly.		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
8	I found the system very cumbersome to use.		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
9	I felt very confident using the system.		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
10	I needed to learn a lot of things before I could get going with this system.		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

In questo questionario le domande dispari sono positive, mentre quelle pari sono negative. Il punteggio viene calcolato in modo che dagli item dispari vada sottratto uno dalla risposta, e da quelli pari si sottrae la risposta da 5 (5-risposta), i passi precedenti scalano tutte le risposte da 0 a 4. Una volta fatto ciò devono essere sommati tutti i valori, ed il totale deve essere moltiplicato per 2.5, in questo modo il massimo raggiungibile è 100.

Per interpretare il punteggio si utilizza la seguente tabella:

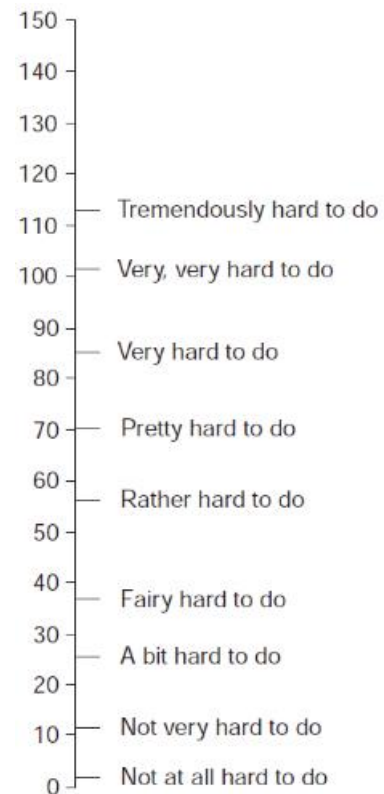
SUS Score Range	Grade	Percentile Range
84.1–100	A+	96–100
80.8–84	A	90–95
78.9–80.7	A–	85–89
77.2–78.8	B+	80–84
74.1–77.1	B	70–79
72.6–74	B–	65–69
71.1–72.5	C+	60–64
65–71	C	41–59
62.7–64.9	C–	35–40
51.7–62.6	D	15–34
0–51.7	F	0–14

Questa tabella non deve essere utilizzata in modo "assoluto" (Se la finestra di login standard ha A come voto, e tu crei una finestra di login che prende come votazione B+, non va bene, mentre se la procedura di stampa unione di word ha C come votazione e voi ne create una con voto B+ è un ottimo risultato)

7.3 Questionario SMEQ

Lo SMEQ è un questionario standard post-task, SMEQ sta per Subjective Mental Effort Question (Zijlstra and van Doorn 1985), in questo questionario si ha una domanda singola, il punteggio varia da 0 a 150, e ad alcuni punteggi sono assegnati delle label esplicative.

L'utente posiziona lo slider tra 0 e 150, le label lo aiutano a scegliere il punto. Scientificamente si ha correlazione fra i punteggi e il tempo di completamento, il numero di numero di compiti completati ed il numero di errori



7.4 Valutazione remota: A/B testing

Nella valutazione remota si possono ricavare delle informazioni utilizzando diversi dati di interazione con applicazioni/siti, si possono utilizzare strumenti diversi in base alle informazioni da ottenere

- **Il logging remoto:** si sfruttano informazioni su pagine accedute e tempistiche per analizzare l'interazione, questo si può effettuare sia lato server che lato client.
- **Google analytics:** fornisce delle metriche da cui si possono estrarre alcuni dati utili per capire quante persone tornano sul sito ecc.
- **Eye tracking:** si analizza il modo di leggere la pagina per individuare le parti critiche.

L'A/B testing consiste nel creare due versioni diverse della visualizzazione, si stabilisce un obiettivo (e.g. click sul pulsante per maggiori informazioni), ed al 50% di utenti si propone la versione del sito A, ed alla restante parte la versione del sito B, dopo un periodo di tempo, stabilisco se ci siano delle differenze rilevanti nei risultati. L'A/B testing è un metodo abbastanza semplice se si ha un insieme di utenti adeguato, però può solo confrontate due alternative, inoltre per poter essere considerato significativo le differenze devono essere minime, altrimenti è difficile capire quale sia la causa. Un lato negativo di questa metodologia è che non fornisce indicazioni sul perché ci sia una differenza.

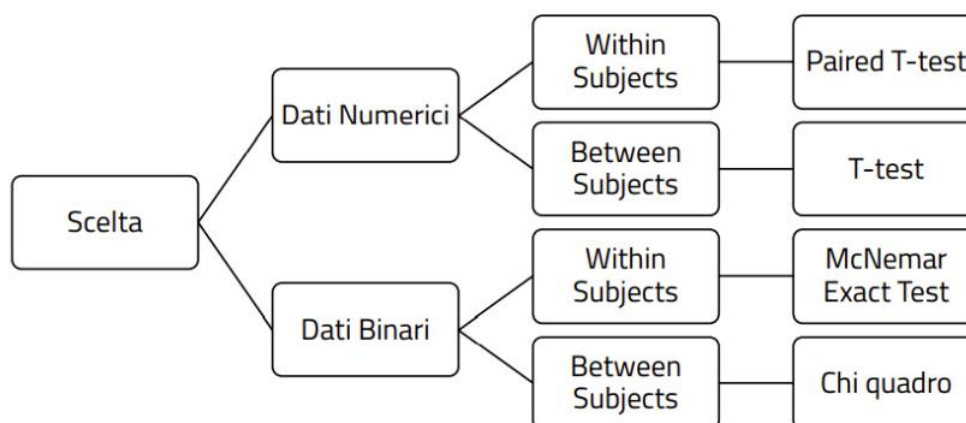
7.5 Between e within subjects

Quando abbiamo diverse (solitamente non più di 2) condizioni da testare per diversi design, è possibile effettuare dei test comparativi con diverse soluzioni di design, con il nostro prodotto

comparato ad un prodotto concorrente, e il nostro design che magari possiede o non possiede una certa funzionalità. Si possono creare due situazioni:

- **Design Between Subjects:** invitiamo un certo numero di utenti, e ad ognuno è assegnato uno solo dei due prototipi su cui effettuare la valutazione, quindi ogni singolo utente viene sottoposto ad una sola condizione di valutazione. Si usano quindi due gruppi diversi di utenti e si introduce più variabilità nel campione
- **Design Within Subjects:** Invitiamo anche qui un numero inferiore di utenti rispetto al Between subjects, perché ad ognuno degli utenti vengono assegnati entrambi i prototipi per farglieli provare. Bisogna prendere degli accorgimenti, perché al ripetere lo stesso compito su diversi supporti, l'utente impara dall'esecuzione del primo design, e si porterà questa esperienza quando dovrà ripetere l'esecuzione con la seconda interfaccia: Dato che nella prima prova l'utente dovrà capire sia cosa fare sia come fare, nella seconda che avviene subito dopo dovrà capire soltanto come svolgere un certo task, e potrebbe sembrargli tutto più semplice. Un accorgimento in merito a questo aspetto potrebbe essere l'inversione dell'ordine in cui far provare i prototipi ai diversi utenti (gli utenti dispari provano prima il prototipo A poi quello B, gli utenti pari il contrario). Se mettessimo per tutti gli utenti il design A come primo, questo sarebbe sempre svantaggiato in quanto potrebbe sembrare più difficile del B per i motivi descritti in precedenza

Per la scelta del metodo di design del test si utilizza la seguente tabella:



Una volta scelto il metodo di comparazione, si deve calcolare il p-value, un valore che **indica la probabilità che non ci sia una differenza tra le due medie**, in linea generale si può dire che se

$$p \leq 0.05$$

allora la differenza è significativa, altrimenti no.

I risultati delle valutazioni quantitative devono essere appurati sfruttando la statistica, a differenti tipologie di valutazioni si associano differenti test, noi vedremo qualche esempio di metodi più semplici:

- Comparazione di due medie, per questo si effettuano valutazioni, oppure si può verificare il tempo di completamento per i task
- Comparazione di risultati binari, per questi risultati si possono verificare il tasso di completamento, il tasso di conversione, oppure effettuare un A/B testing