

Programmazione d'Interfacce

Federico Matteoni

Indice

1	Introduzione	2
2	Il Corso	2
3	Design	2
3.1	XX Designer	2
3.1.1	UX Designer	3
3.1.2	UI Designer	3
3.2	Front-End Developer	3
4	Interfacce Utente	3
5	Good and Bad Design	4
6	Human Centered Design	5

1 Introduzione

Appunti del corso di **Programmazione d'Interfacce** presi a lezione da **Federico Matteoni**.
More like *design d'interfacce*.

Prof.: **Daniele Mazzei**, mazzei@di.unipi.it
Riferimenti web:

- ?

Esame: compitini/scritto + orale discorsivo dove si discute un software noto.
Possibile proporre un software personale da presentare all'esame orale, spiegando come si è applicati i rudimenti del corso sul software presentato.

Materiale didattico:

- **Google Classroom**, slide presentate a lezione e altro materiale didattico
Codice **c14kiy** con le credenziali d'ateneo.
La suite Google è attivabile a *start.unipi.it/gsuite*
Non è autorizzata la divulgazione
- La Caffettiera del Masochista, Donald A. Norman
Eng: The Design of Everyday Things
- Designing the User Interface, Ben Shneiderman
- *www.usability.gov*
- *interaction-design.org*

Ricevimento: Mercoledì 16-18, Stanza 366

2 Il Corso

Interface Development in 2020 diventa **Interface Design in 2020**

Diviso in due parti

- **UX e UI** con introduzione, UI vs UX, HCI, paradigmi, gamification...
- **Strumenti per lo sviluppo dell'interfaccia utente** presentati da vari ospiti: Unity, Zerynth, Ubidots, Angular, Amazon Lex, ...

Interfaccia è qualsiasi metodo utilizzato da una persona per **interagire** con un dispositivo.

3 Design

Cos'è il design Il design è la **pianificazione o la specifica per la costruzione di un oggetto o sistema** o per l'implementazione di un'attività o processo. Diventa l'esatto opposto della decomposizione del problema in sottopassaggi, cioè del pensiero computazionale. Il design parte dalla base del problema e **identifica soluzioni per la causa del problema**. Si può avere anche il design di una strategia di implementazione.

„Bisognerebbe progettare le applicazioni come se fossero persone che ci piacerebbe frequentare”. Ad esempio Netflix, o il frigorifero.

3.1 XX Designer

Discernere tra Graphic Design, User Experience Design (UX Design) e User Interface Design (UI Design).

UX : come l'utente si sente per interagire e cosa vuole fare. Aspetto più psicologico, guida la UI design in base a statistica fatta su gruppi di utenti. Manda "l'output" a chi fa UI e al marketing.

UI : come l'utente interagisce col prodotto (shortcut, sottomenu...)

3.1.1 UX Designer

Si deve porre il problema di quali approcci usare per risolvere problemi evidenziati da analisi di mercato.

Chi paga non è detto che sia chi usa il servizio. Ad esempio Netflix viene pagato da una persona, ma lo stesso account viene usato anche da altre persone (anzi, in particolare **il 90% del tempo** chi usa l'account non è chi paga). Uno dei metodi usati per fare UX è quello della **definizione delle personas** (cioè un archetipo di utente). Una persona può assumere diverse personas.

User Experience Con User Experience si parla del prodotto e di come si comporta nel mondo reale, che è fatto di *personas*. **Non si può progettare una user experience, si può progettare per la user experience.** La user experience è ciò che fa l'utente, e lo sviluppatore non ha controllo su ciò. L'utente si avvicina al software come gli pare.

3.1.2 UI Designer

Dalla UX si crea lo **sketch** dell'interfaccia. Non viene prodotto subito il wireframe ma bisogna partire da altro, ad esempio dai **casi di studio**. Esistono più casi di studio per ogni personas (casalinga voghera che fa bonifico, casalinga voghera che cambia password, ecc.). Ogni caso di studio è **specifico per personas**, poiché personas diverse hanno capacità diverse (non conoscere alcuni concetti, non saper fare determinate operazioni...).

L'UI design è un procedimento diverso dal front-end developing, quindi possono essere persone separate. Il designer progetta le guideline che istruiscono il developer.

Si può dire che la UI design è sottoarea di UX design.

3.2 Front-End Developer

Esegue il design della UI convertendolo in funzionalità del prodotto.

4 Interfacce Utente

L'interfaccia utente (UI) L'UI di un sistema è **lo spazio dove avviene l'interazione uomo-macchina**: lo schermo, le casse, il mouse e quant'altro.

L'obiettivo dell'interfaccia è far sì che **l'utente possa controllare la macchina, e non il contrario**. L'interfaccia può però influenzare il comportamento dell'utente, ad esempio se voglio guidare l'utente in un particolare modo l'interfaccia deve dare un feedback tale da guidare l'utente.

L'altro obiettivo dell'interfaccia è **rendere fruibile in maniera piacevole le funzionalità che una macchina eroga** verso l'utente. Il termine **user-friendly** non può essere omissivo: tra un'app facile e piacevole da usare e una solo facile da usare, l'utente medio preferirà sempre la prima.

L'interfaccia è strutturata a layer. lo HID (Human Interface Device) è la periferica con cui l'umano interagisce col sistema. Questo serve per usare più HID per interagire con diverse applicazioni.

HMI (Human Machine Interface) è più astratta rispetto a HCI (Human Computer Interface), quindi in HMI è più teorica la cosa.

Diversi tipi di interfacce Abbiamo 5 sensi, quindi diverse **categorie d'interfaccia**: le più comuni sono **grafiche** e **tattili** (**GUI**, Graphical User Interface). Se si aggiunge anche il suono diventano **MUI** (Multimedia User Interface). Il concetto di GUI è stato coniato in un tempo in cui l'audio era raro. Adesso **praticamente tutte le interfacce sono MUI**.

Esempio di MUI riprogettata in GUI: Facebook. I video partivano in automatico con l'audio attivo, mentre ora sono mutati. Poi sono stati aggiunti i sottotitoli automatici: questo è un esempio di tecnica ideata per le utenze disabili e riusata per poter far fruire il prodotto a quelle personas che in quel momento non possono usufruire dell'audio. *Meglio un sottotitolo sbagliato che niente.*

Categorizzare interfacce composite user interfaces di tipo standard (touchscreen, monitor, mouse, tastiera, mobile in utilizzo base). In AR e VR abbiamo composite user interface rispetto aumentata (interfaccia eroga contenuti non puramente digitali ma prendono parte della realtà esterna che circonda l'utente, *questa realtà viene aumentata*) e virtuale (utente schermato dall'interfaccia, no relax diretta col mondo esterno, es. Oculus Rift, *un'altra realtà*). Qualia Interfaces quando si stimolano tutti i sensi

CUI anche catalogate per numero di sensi. Termine 4d scorretto (es. vr è sempre 3d, 4d va bene per marketing). in ui si usa x-sense. es di 3-sense smell-o-vision: vista, suono e olfatto. Se poltrona vibra diventa 4-sense, si aggiunge il tatto.

Le ui sono le stesse di 10 anni fa. Bisogna mettere in discussione i paradigmi attuali. Ho convertito ambiente fisico in digitale prendendo ispirazione dall'abitudine utente per rendere semplice passaggio. Ora l'utente è abituato, bisogna cambiare.

5 Good and Bad Design

Il buon design non esiste, poiché si fa design *per* la user experience **di una determinata personas**. Le due caratteristiche più importanti su cui misurare il buon design sono:

Discoverability: capacità innata di un sistema di veicolare i possibili usi e dire come si usa. Nno è detto che capit o cosa fare si riesca a farlo
per avere buona discoverability si usa tipicamente la visibilità. Un rubinetto con i pomelli bene in vista incrementa la discoverability. Nel software, questo lavoro lo fanno i bottoni.
Altro esempio è la maniglia (ambigua, quindi si scrive push/pull).

Understanding: capire i possibili usi
Il fornello, è in cucina quindi so che si usa per scaldare ecc., il problema è il mapping pomello -> fornello. Si può risolvere con l'icona del fornello corrispondente.
Non sottovalutare il costo mentale dell'utente.

Design of Useful Things **Il paradosso di TripAdvisor**: le gente mangia bene e non recensisce. Mangia male e recensisce.

Quando le cose vanno bene si dà per scontato, si dimenticano subito. In qualche modo l'uomo pensa che vadano bene per definizione. Quando qualcosa va storto l'amigdala crea un ricordo con un peso molto maggiore.

marcatore somatico: ricordo esperienze in base alle sensazioni. Più forte è la sensazione più si cementifica il ricordo. Un incidente ad una curva, la ricordo bene.

stesso discorso software: se non riesco ad usarlo monta frustrazione. Gli umani non informatici tengono a ritenere le macchine come superintelligenti, quindi associano a frustrazione l'incapacità personale: se credo di non essere in grado di usare il software non ci riprovo.

Confrontando IA e intelligenza umana, IIA è strettamente limitata a computazione e risoluzione problemi logici. La mente umana non funziona ad algoritmi, ma procede per deduzione del tipo: ipotesi senza fondamento e me ne auto-convinco.

Le macchine seguono regole semplici: gli algoritmi. Se non comprendo e seguo le regole sotto le macchina non hanno la flessibilità (common sense) tale da assecondare l'utente.

Se chiedo telecomando per aula D2, la signora corregge in D1 e dà telecomando. La macchina dice non esiste D2.

le macchine non hanno buonsenso. La maggiorparte delle regole sotto il software sono note solo agli sviluppatori. Potrebbe andare bene basta renderle discoverable.

Bisogna invertire il paradigma: se qualcosa va storto è colpa dello sviluppatore non dell'utente. Dovere della macchina è essere comprensibile da parte dell'utente.

Non esiste "l'Utente", ma tutta una serie di utenti (personas) diversi fra loro.

Altro problema dello sviluppatore è pensare che la spiegaz logica sia sufficiente, ma la maggiorparte ragiona in maniera empatica.

Bisogna accettare che il comportamento umano è com'è e non come vogliamo che sia.

Three Mile Island accident.

6 Human Centered Design

Alla fine di ogni passo c'è l'umano.
Si tratta di una norma ISO.

Un approccio Lo HCD è un approccio di design specificatamente orientato allo sviluppo di sistemi interattivi. L'obiettivo di un design antropocentrico è fare sistemi utili, altamente usabili e che si focalizzino sull'utente. Il metodo è orientato ad efficienza ed efficacia. Significa che se devo risolvere problema non mi interessa risolvere completamente il problema ma raggiungere il migliori risultato che posso ottenere grazie ad un utente che usa il mio software. Se il 70% degli utenti raggiungono scopo col nostro software allora ha efficiacia 70%. Magari risolvo una parte minore del problema ma sposto efficienza all'80%.

Less is more. Meglio una feature in meno che una in più. Ogni volta che aggiungi una feature devi dimostrare perché e a cosa serve perché va: spiegata, testata, mantenuta oggi e domani (**backward compatibility**).

Prima di tutto metto i bisogni del mio utente. Prima di tutto il processo di sviluppo metto l'assoluzione dei bisogni dell'utente.

Il problema principale delle UI è un problema di comunicazione. Le due di prima -ibility si tradutcono in capacità comunicativa dell'interfaccia.

La comunicazione è ancora più importante quando le cose non vanno bene: strategie di mitigazione dell'errore. focalizzare interazione soprattutto nel comunicare ciò che è andato storto. Quando qualcosa v'è storto devo aiutare l'utente frustrato a risolvere il problema, se lo aiuto a risolvere da solo proverà sensazione positiva di successo per aver capito cosa non funzionava e crea empatia col sistema. Quindi evitare frustrazione, e aiutare a risolvere quando insorge problema e frustrazione.

Capire l'utente, prima tecnica l'osservazione. Non è detto sia sempre possibile, ma ogni volta bisogna chiederci come provare a farla: beta e alfa non devono solo debuggare ma servono anche a capire ciò che fanno gli utenti. Bisogna avere interfacce che danno statistiche sull'utilizzo: quanti click sul bottone, quante volte procedura finisce ecc..

Le specifiche dello HCD non si possono scrivere. Quindi è un paradigma che si sposa bene con la CS perché va avanti per iterazioni: specifica alto livello, implemento parte, testo sull'utente reale. Quando ritengo buono congelo, e passo ad altra parte dell'interfaccia.

tabella da riscrivere

Possiamo progettare per esperienza utente, il design industriale e progettare per l'interazione. lo HCD non è area di focus del processo di design ma è metodo.

Utilizzo l' HCD per progettare tutto il resto.