**PROGETTARE L’INTERAZIONE FRA UOMO E MACCHINA** *Lezione 1*

Negli ultimi anni si parla molto di User Experience *Design*, User Interface *Design*, … ma che cos’è il *design*? Per **design** si intende:

* sia il processo di progettazione (“*è stato avviato il design del prodotto*” cit.);
* sia il risultato di questo processo. (“*questo prodotto ha un bel design*” cit.)

Nel mondo del design ci si approccia alla progettazione con una forma mentis molto diversa da quella “computazionale”, tipica del mondo informatico:

* Un informatico, non appena vede un problema, inizia subito a dividerlo in tanti sottoproblemi, risolvendo ognuno in una forma che sia eseguibile da un calcolatore;
* Un designer, non appena vede un problema, prima cerca di capire perché il problema esiste, e solo una volta appurato che esiste veramente – e cioè, che gli utenti lo percepirebbero – si adopera per cercare di risolverlo (affrontandolo così con una maggiore comprensione del problema nel suo insieme).

Il capo di un designer, dunque, è l’utente. Questo in alcuni casi può essere un bordello, perché costringe a modificare le specifiche di prodotto in favore dell’esperienza utente, sacrificando funzionalità tecniche e qualità dell’implementazione. Occorre dunque trovare un corretto bilanciamento tra esperienza utente, funzionalità tecniche e qualità dell’implementazione.

**INTERACTION DESIGN**

L’**interaction design** è l’attività di progettazione dell’interazione che avviene tra esseri umani e oggetti.

E’ composto da tre fasi:

* Product design;
* UX design (user experience design);
* UI design (user interface design).

**PRODUCT DESIGN**

Il **product design** è il punto di partenza dell’interaction design: in questa fase si progetta un prodotto il cui obiettivo principale è essere utilizzato da quanti più utenti possibili, migliorandone la vita.

Un *product designer*, dunque, è un inventore: si focalizza su un problema e inventa un nuovo prodotto che permetta di risolverlo.

Nella fase di product design può essere utile costruire un *lean canvas*.

Immagine che contiene testo, schermata, Carattere, documento

Descrizione generata automaticamente

**UX DESIGN**

Nella fase di **UX design** viene progettato come l’utente “si deve sentire” durante l’interazione con il prodotto, con l’obiettivo di far vivere all’utente la migliore esperienza possibile (*es.* in termini di facilità d’uso e piacere nell’interazione con il prodotto).

**UI DESIGN**

Nella fase di **UI design** viene progettato lo stile grafico dell’interfaccia utente, affinché l’utente abbia la migliore esperienza possibile progettata in fase di UX design.

Un *UI designer*, dunque, solitamente non è un informatico, e produce un *wireframe*, e cioè una bozza grafica dell’interfaccia con una serie di linee guida che verranno poi seguite dagli sviluppatori per creare l’interfaccia vera e propria.

**HUMAN CENTERED DESIGN** *Lezione 2*

Lo **Human Centered Design** è un modo di pensiero che parte dai bisogni, capacità e comportamenti umani e adatta la progettazione ad essi.

Questo modo di pensiero procede per quattro fasi:

1. **Specificare il contesto d’uso:** si identificano gli utenti che utilizzeranno il prodotto, per cosa lo utilizzeranno e sotto quali condizioni e vincoli;

Ricorda ***CR****isto*(le prime due fasi),le altre fasi sono poi temporalmente successive.

1. **Specificare i requisiti:** si identificano i business requirements e gli obiettivi utente;
2. **Progettare la soluzione**: si crea il prodotto, passando prima per bozze e prototipi;
3. **Testare**: si fa testare il prodotto agli utenti. Si riparte poi dalla prima fase, tenendo conto dei risultati del test, così da migliorare continuamente il prodotto.

**PROGETTARE L’INTERAZIONE** *Lezione 3*

Un buon prodotto, per essere tale, deve avere due proprietà fondamentali:

* **Discoverability:** è la capacità di un prodotto di comunicare i propri possibili usi all’utente.   
  Per avere una buona discoverability si usa tipicamente la visibilità: per esempio, un rubinetto con i pomelli ha una buona discoverability, perché un utente, vedendo i pomelli, capisce subito come va utilizzato. Un rubinetto che si attiva con dei sensori, invece, ha una discoverability inferiore, perché un utente, non vedendo nulla, potrebbe non capire subito come va utilizzato;
* **Understanding:** è la capacità di un prodotto di farsi usare correttamente dall’utente.

La discoverability viene ottenuta applicando sei principi psicologici fondamentali: *affordances*, *signifiers*, *mappings*, *feedback*, *conceptual* *model, constraints*.

**AFFORDANCES E ANTI-AFFORDANCES**

Un’**affordance** è una relazione tra le proprietà che il prodotto possiede e le capacità dell’utente che ci interagisce. Se stabilita, abilita alcuni modi di interazione tra l’utente e il prodotto.

Per esempio, una sedia presenta l’affordance di consentire la seduta ed essere sollevata da una persona adulta normodotata, mentre non presenta tale affordance per un neonato (non è dotato della capacità per farlo).

Un’**anti-affordance,** invece, è una relazione tra le proprietà che il prodotto possiede e le capacità dell’utente che ci interagisce. Se stabilita, disabilita alcuni modi di interazione tra l’utente e il prodotto.

Per esempio, un’insegna con degli “spunzoni” sopra presenta l’anti-affordance di permettere ai piccioni di poggiarvisi sopra.

**SIGNIFICANTI (SIGNIFIERS)**

Le affordances, dunque, indicano “cosa” si può fare nell’interazione con il prodotto. Queste, dunque, devono essere ben visibili; per dare loro visibilità si usano i **significanti:** elementi del prodotto che indicano invece “dove” e, in alcuni casi, “come” interagire con il prodotto.

I significanti possono essere:

* **Intenzionali**. Un esempio è un’etichetta con scritto “spingere” attaccata vicino alla maniglia di una porta: la porta presenta l’affordance di essere aperta da una persona adulta normodotata   
  (“cosa” si può fare), e l’etichetta rende tale affordance visibile (indicando “dove” e, in questo caso, anche “come” aprirla);
* **Non intenzionale**. Un esempio è un sentiero tracciato da persone che camminano attraverso un campo: un campo presenta l’affordance di essere attraversato da una persona adulta normodotata   
  (“cosa” si può fare), e il sentiero rende tale affordance visibile (indicando “dove” attraversarlo).

**MAPPING** *Lezione 4*

Il **mapping** è il posizionamento dei significanti.

Immagine che contiene simbolo, cerchio, elettrodomestico

Descrizione generata automaticamenteIl modo migliore di fare mapping è quello *naturale*, e cioè quello che sfrutta analogie spaziali:

Questo è un esempio di mapping naturale.

Abbiamo di fronte dei fornelli, che hanno l’affordance di essere accesi da una persona adulta normodotata (“cosa” si può fare). Le manopole sono i significanti, in quanto rendono tale affordance visibile (indicando “dove” i fornelli si possono accendere e, in questo caso, anche “come” si possono accendere”), e queste sono posizionate nello stesso ordine in cui sono disposti i fornelli, rendendo così più chiari i possibili usi dei fornelli (discoverability) e come vanno usati correttamente (understanding).

**FEEDBACK**

Un **feedback** è la comunicazione del risultato di un’azione, è una risposta che il prodotto dà all’utente.

Un feedback deve essere:

* **Immediato:** se il feedback viene dato troppo in ritardo, l’utente potrebbe rinunciare all’attività che stava compiendo e passare ad altro o addirittura non riuscire a comprendere l’origine del feedback stesso;
* **Informativo:** deve far capire bene il risultato dell’azione compiuta;
* **Semplice:** non deve essere costituito da troppi segnali, altrimenti l’utente inizierebbe ad ignorarli, rischiando così di ignorare anche i feedback davvero importanti.

**CONCEPTUAL MODEL**

Un **modello concettuale** è una spiegazione altamente semplificata ma utile di come funziona il prodotto, effettuata dal designer del prodotto stesso. Questa esprime come il designer vuole che l’utente percepisca il prodotto: considerando ad esempio il desktop di un computer, le cartelle e le icone costituiscono un modello concettuale, in quanto il designer ha voluto che l’utente percepisca il computer come organizzato in cartelle ed icone, quando invece la realtà è molto più complessa.

Quando un utente interagisce con un prodotto, a sua volta, si costruisce un **modello mentale**, e cioè una propria spiegazione altamente semplificata di come funziona il prodotto. E’ bene che il modello mentale costruito dagli utenti si discosti il meno possibile dal modello concettuale costruito dal designer, così che gli utenti non abbiano troppe difficoltà nell’utilizzo del prodotto.

L’**immagine di sistema** è l’insieme di tutte le informazioni che gli utenti usano per costruirsi un modello mentale del prodotto. Queste informazioni sono costituite dal modello concettuale + tutte le informazioni sul prodotto messe a disposizione dell’utente (l’aspetto del prodotto, ciò che comunicano le pubblicità, ciò che è scritto nel libretto delle istruzioni…).

**VINCOLI (CONSTRAINTS)** *Lezione 5*

I **vincoli** sono degli indizi che limitano l’insieme delle azioni possibili sul prodotto.

I vincoli possono essere di 4 tipi:

* **Fisici:** si affidano a proprietà del mondo fisico. Considerando come esempio di prodotto una motocicletta Lego da assemblare, ritroviamo questo vincolo nel saper determinare come vanno inseriti i pezzi: questi si incastrano solo in un determinato verso, dunque sarà quello il verso corretto;
* **Culturali:** si affidano ad abitudini culturali, comportamenti e convenzioni (che possono anche cambiare nel tempo). Nell’esempio della motocicletta Lego, ritroviamo questo vincolo nel saper determinare la collocazione delle luci: è abitudine che il bianco va messo all’anteriore, mentre il rosso al posteriore;
* **Semantici:** si affidano al significato della situazione corrente. Nell’ esempio della motocicletta Lego, ritroviamo questo vincolo nel saper determinare la collocazione del motociclista: vista la situazione, deve per forza stare seduto guardando in avanti;
* **Logici:** si affidano alla logica umana. Nell’esempio della motocicletta Lego, ritroviamo questo vincolo nel saper determinare dove collocare un pezzo quando è l’unico rimasto: a logica, va messo nel posto in cui si vede che manca un pezzo.

Un buon mapping, ad esempio, può creare dei vincoli logici.

**APPLICAZIONE DEI CONCETTI VISTI: FUNZIONI OBBLIGANTI**

Le **funzioni obbliganti** sono una forma di vincolo fisico: sono situazioni in cui le azioni sono vincolate in modo tale che un mancato passaggio non permetta che il prossimo avvenga.

Sono di tre tipi:

* **Interlock:** obbliga a svolgere una serie di passaggi in un certo ordine per eseguire l’azione desiderata.
  + Nel mondo fisico, un esempio di interlock è un dispositivo di sicurezza che non permette di avviare un frullatore se prima non si inserisce il coperchio:prima bisogna inserire il coperchio, poi bisogna premere un bottone, e così si avvierà il frullatore;
  + Nel mondo software, un esempio di interlock è il sistema di sicurezza *reCAPTCHA* che non permette di registrarsi a un sito se prima non si inseriscono le lettere che compaiono a video: prima bisogna inserire le lettere che compaiono a video, poi bisogna premere un bottone, e così ci si registrerà al sito.

Sono tutte funzioni obbliganti, visto che un mancato passaggio non permette che il prossimo avvenga.

* **Lock-in:** mantiene attiva una funzione impedendo che qualcuno la interrompa prematuramente.
  + Nel mondo software, un esempio di lock-in è la finestra “*salva prima di chiudere*” tipica di molti applicativi: quando si prova a chiudere un’applicazione senza salvare, compare a schermo una finestra che ci chiede se siamo sicuri che vogliamo chiudere l’applicazione senza aver salvato nulla, impedendo così che l’applicazione possa essere chiusa prematuramente. Solo dopo aver cliccato sull’apposito bottone “*Don’t save*”, l’applicazione verrà effettivamente chiusa.
* **Lock-out:** impedisce l’accesso ad uno spazio pericoloso o che succeda qualcosa.
  + Nel mondo software, un esempio di lock-out è la finestra “*VM 18*” tipica di molti siti porno: quando si prova ad accedere ad un sito porno, compare a schermo una finestra che ci chiede se abbiamo 18 anni, impedendoci così di accedere ad uno spazio pericoloso. Solo dopo aver cliccato sull’apposito bottone “*I’m 18*”, entreremo effettivamente nel sito.

**APPLICAZIONE DEI PRINCIPI VISTI: ACTIVITY-CENTERED CONTROL**

In alcuni casi, può essere utile avere un interruttore per ogni attività. Si parla allora di **activity-centered control**: per ogni attività che l’utente potrebbe voler fare, viene progettato un controllo tale che, azionando un semplice interruttore, viene regolata automaticamente tutta una serie di oggetti che consentono di svolgere tale attività, senza che l’utente debba regolare ogni cosa singolarmente. Un esempio è il pulsante “*cinema*” presente sul telecomando di molte TV: premendolo, viene regolata automaticamente la luminosità della TV, il volume, l’equalizzazione… tutta una serie di oggetti che consentono di svolgere al meglio l’attività della visione dei film, senza che l’utente debba regolare ogni cosa singolarmente.

**COME LE PERSONE AGISCONO** *Lezione 6*

Quando un utente interagisce con un prodotto deve superare, nell’ordine, due golfi:

1. **Il golfo dell’esecuzione:** l’utente deve pianificare e specificare le azioni da eseguire sul prodotto.   
   Influiscono sul golfo dell’esecuzione i significanti, il mapping, i vincoli e il modello concettuale;
2. **Il golfo della valutazione:** l’utente deve percepire e interpretare le risposte del prodotto alle sue azioni, confrontandole con quello che era il suo obiettivo. Influiscono sul golfo della valutazione il feedback e il modello concettuale.

In particolare, nell’interazione con un prodotto, un utente passa per i cosiddetti **7 stati dell’azione:**

* **GOAL:** l’utente deve definire l’obiettivo per cui usare il prodotto (*es.* mandare i propri dati ad un’azienda usando un certo sito web);
* **PLAN:** l’utente deve pianificare cosa fare sul prodotto (*es.* riempire un form e assicurarsi di mandarlo all’azienda corretta);

**Golfo dell’esecuzione**

* **SPECIFY:** l’utente deve specificare qual è la sequenza di azioni precisa da eseguire sul prodotto (*es.* riempire uno *specifico* form e assicurarsi di premere uno *specifico* pulsante);



* **PERFORM:** l’utente deve eseguire la sequenza di azioni che ha specificato;
* **PERCEIVE:** l’utente deve percepire le risposte del prodotto alla sua sequenza di azioni;

**Golfo della valutazione**

* **INTERPRET:** l’utente deve interpretare queste percezioni;



* **COMPARE:** l’utente deve confrontare le risposte del prodotto che ha interpretato con quello che era il suo obiettivo, per vedere se il suo obiettivo è stato effettivamente realizzato.

L’informazione necessaria all’utente per andare avanti nel golfo dell’esecuzione è il **feedforward**.

L’informazione necessaria all’utente per andare avanti nel golfo della valutazione è il **feedback**.

**PENSIERO UMANO** *Lezione 7*

La mente umana è suddivisa in:

* **Pensiero subconscio**: veloce, automatico, è usato per attività alle quali si è abituati;
* **Pensiero conscio**:lento, manuale, è usato quando si sta apprendendo qualcosa di nuovo.

Sulla base di questi, è possibile dividere il funzionamento del nostro cervello secondo tre livelli:

* **Livello viscerale**: è il livello più elementare. Permette di rispondere prontamente senza controllo cosciente;

**Basati prevalentemente sul pensiero subconscio**

* **Livello comportamentale**: è il livello delle abilità apprese in circostanze più o meno simili a quella attuale. E’ guidato dalle aspettative, e durante l’attesa di conferma di tali aspettative è guidato dalle emozioni;

**Basato sul pensiero conscio**



* **Livello riflessivo**: è il livello del controllo cosciente. E’ qui che ha luogo la comprensione profonda e il ragionamento.



Veicolare informazioni all’utente mentre egli si trova nel livello riflessivo è estremamente efficace: in questo livello, infatti, l’utente comprende profondamente tali informazioni, e quindi le conseguenti emozioni sono più durature.

**ERRORE UMANO**

La maggior parte degli errori umani non è dovuta agli umani stessi (alla loro volontà, o alla loro incapacità), bensì al cattivo design dei prodotti. Molti prodotti, infatti, per essere utilizzati correttamente, richiedono:

* La memorizzazione di procedure molto complesse;
* Stare attenti per molte ore;
* Dover svolgere diversi compiti in simultanea, costringendo così l’utente a dover ogni volta interrompere un compito per eseguirne un altro.

E tutto questo fa perdere pian piano la concentrazione dell’utente, portandolo all’errore.

**ROOT CAUSE ANALYSIS**

Quando si verifica un errore, la **root cause analysis** è un metodo che consiste nell’indagare l’errore finché non si trova la *singola* causa che ne è l’origine.

Questo metodo ha due difetti:

* La maggior parte degli errori non ha una singola causa, ma più cause. Si dice allora che gli errori seguono il *modello a groviera*:

Immagine che contiene schizzo

Descrizione generata automaticamente

* Spesso, purtroppo, la root cause analysis si ferma non appena viene scoperto l’umano che l’ha commesso, additandolo come causa dell’errore (quando invece, come si è detto prima, la maggior parte degli errori umani non è dovuta agli umani stessi).

**5 WHYS**

Quando si verifica un errore, il **metodo dei 5 perché** consiste nell’indagare l’errore per trovare le cause che ne sono l’origine e non fermarsi non appena se ne trova una, ma continuare a indagare finché non si trovano le vere cause di fondo (non bisogna per forza trovare *5* cause; il numero *5* è indicativo).

**CATEGORIE DI ERRORI** *Lezione 8*

Gli errori si dividono in due categorie:

* **Mistakes (errori cognitivi):** si ha un mistake quando si effettua una pianificazione sbagliata.

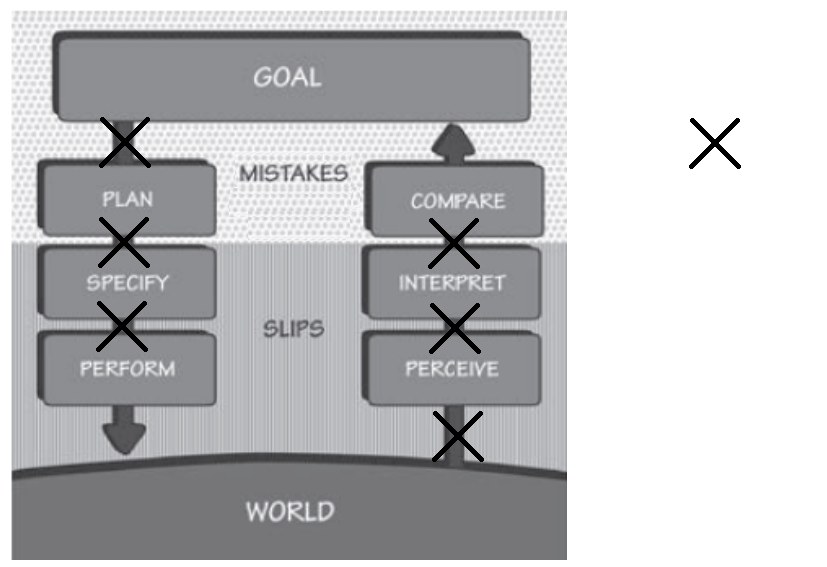
Si dividono in:

* + **Rule-based mistakes:** viene effettuata una diagnosi della situazione giusta, ma poi viene seguita una regola operativa sbagliata, effettuando così una pianificazione sbagliata (*es.* quando un meccanico diagnostica correttamente un difetto nella batteria di una macchina ma decide volontariamente di non sostituirla quando invece sarebbe stato corretto farlo è un rule-based mistake, perché il meccanico ha effettuato una diagnosi della situazione giusta, ma poi ha seguito una regola operativa sbagliata, effettuando così una pianificazione sbagliata);
  + **Knowledge-based mistakes**: viene effettuata una diagnosi della situazione sbagliata, e di conseguenza verrà effettuata anche una pianificazione sbagliata.
* **Action Slips (lapsus):** si ha un action slip quando pianifica di fare qualcosa ma si esegue un’azione sbagliata (*es.* “a forza di fare sempre una stessa cosa, l’ho sbagliata” è un action slip, perché ho pianificato di fare tale cosa, ma poi ho eseguito un’azione sbagliata).

Esistono poi i **memory-lapse,** e cioè dimenticanze che possono portare ad un mistake (se avvengono in fase di pianificazione) o ad un action slip (se avvengono non in fase di pianificazione).

Come si può vedere dalla figura:

* I mistakes avvengono nelle 3 fasi alte dell’azione;
* Gli action slips avvengono nelle 4 fasi basse dell’azione;
* I memory-lapse (segnati dalle *X*) avvengono nelle transizioni tra una fase e l’altra.

****

Tipicamente:

* Un principiante commette mistakes: le cose, infatti, non le sa bene ancora, e quindi tende a fare pianificazioni sbagliate. Commette questi errori dunque perché è al livello riflessivo in preda al pensiero conscio, effettua ragionamenti, ma sono sbagliati;
* Un esperto commette slips: Le cose, infatti, le sa, quindi la pianificazione la sa fare.   
  Commette questi errori dunque perché è al livello viscerale/comportamentale in preda al pensiero subconscio, non effettuando più ragionamenti.

**PREVENZIONE DEGLI ERRORI**

Ci sono vari modi per prevenire gli errori:

* **Vincoli:** si aggiungono dei vincoli, così da limitare le possibili azioni sbagliate sul prodotto;

Sono temporalmente successive (la prima è prevenzione errore, le due seguenti sono “verifica” casomai l’utente sta commettendo un errore, l’ultima è annullamento dell’errore)

* **Messaggi di conferma:** quando un utente cerca di svolgere un’azione che potrebbe distruggere qualcosa di importante, gli viene chiesto se è sicuro di volerla fare. Questi messaggi, tuttavia, non sono molto efficaci a prevenire gli errori, in quanto vengono mostrati all’utente poco dopo che egli ha richiesto di compiere un’azione, e quindi quando è ancora molto convinto di volerla compiere (dunque, quasi sicuramente, l’utente confermerà di voler proseguire con la sua azione);
* **Controlli di sensibilità:** il prodotto verifica che l’azione che l’utente vuole eseguire sia ragionevole   
  (*es.* considerando un’applicazione usata per fare bonifici, se un utente ogni giorno ha sempre effettuato bonifici da 5 euro ed un giorno ne fa uno da 2000 euro, il programma potrebbe vedere tale azione come non ragionevole, bloccandola e allertando l’utente).
* **Undo:** si fornisce un comando che permetta di annullare le azioni svolte in precedenza sul prodotto;

**UX DESIGN** *Lezione 9*

Iniziamo ad analizzare in dettaglio la fase di UX design di un prodotto.

**PERSONAS**

La prima cosa da fare nella fase di UX design è definire le *personas*. Una **persona** è un personaggio che rappresenta una delle possibili tipologie di utenti del prodotto.

Esistono varie tecniche per definire le personas (e dunque, per individuare le possibili tipologie di utenti del prodotto):

* **Task Analysis**: si analizza come gli utenti vorrebbero raggiungere i loro obiettivi;

E, sulla base dei risultati ottenuti, si individuano le possibili tipologie di utenti del prodotto.

* **Feedback**: si effettuano delle interviste e focus groups;
* **Prototipazione**: si creano dei prototipi per vedere la reazione degli utenti.

A seconda dei dati che si hanno per definire le personas, si parla di:

* **Proto-personas:** personas che vengono definite dai designer senza avere alcun dato;
* **Qualitative-personas:** personas che vengono definite a seguito di attività come quelle descritte sopra;
* **Statistical personas:** personas che vengono definite a seguito di un’analisi statistica.

Ogni persona viene descritta con:

* **Dettagli personali:** un nome, una foto, una breve biografia…
* **Dettagli demografici**: l’età, lo stato civile, il reddito…
* **Dettagli attitudinali e/o cognitivi:** informazioni su come si comporta, informazioni su quello che potrebbe essere il suo modello mentale del prodotto…
* **Obiettivi per cui potrebbe usare il prodotto;**

Quante personas andrebbero identificate? Ci si basa sul Principio di Pareto: “*il 20% delle cause provoca l’80% degli effetti*” (e cioè, bisogna identificare il 20% delle personas che utilizzerà il prodotto per l’80% dell’uso complessivo).

**REQUIREMENTS**

Una volta definite le personas, vanno definiti i **requirements**, e cioè caratteristiche di cui il prodotto deve essere dotato per soddisfare le esigenze delle personas.

I requirements si dividono in due categorie:

* **Requirements funzionali:** descrivono quali funzionalità deve avere il prodotto (*cosa fa*);
* **Requirements non funzionali:** descrivono i tratti qualitativi del prodotto (*come lo fa*).

**USER STORIES**

Una volta individuate le personas e definiti i requirements, vanno scritte le user stories. Una **user story** è un requirement espresso dal punto di vista di una persona: è una breve frase in cui viene descritto chi è la persona, che cosa vuole e perché lo vuole.

Tipicamente ci sono una o più user stories per ciascuna persona, ma non il contrario: se si hanno più personas per una user story, molto probabilmente si è descritta la stessa persona usando termini differenti, e quindi si ha una ridondanza.

Esiste uno schema ben preciso per scrivere una user story:

*As a [role], I want [feature] because [reason].*

**SCENARIOS** *Lezione 10*

Una volta scritte le user stories, vanno scritti gli *scenarios*. Uno **scenario** è un’estensione di una user story, che descrive anche come la persona dovrebbe agire durante l’interazione con il prodotto per gestire una specifica situazione (cosa che invece mancava nelle user stories).

**USE CASES**

Una volta scritti gli scenarios, vanno scritti gli *use cases*. Un **use case** è un’estensione di uno scenarios, che descrive anche la completa narrativa delle azioni che la persona dovrebbe compiere per gestire la specifica situazione (devono quindi essere più dettagliati rispetto agli scenarios).

***NB:*** *Alla domanda d’esame orale “come si scrive x” (es. come si scrivono gli scenarios) dì che innanzitutto vanno definite le y cose precedenti (es. personas, requirements, user stories, dando dettagli su come si definiscono), e poi si scrive x.*

**INTERFACCE UTENTE (UI)** *Fine Lezione 19*

Un’**interfaccia utente (UI)** è lo spazio di un prodotto dove avviene l’interazione tra uomo e macchina.   
Il suo principale obiettivo deve essere consentire un utilizzo facile, piacevole e intuitivo del prodotto.

Le interfacce utente si possono classificare in base ai sensi che coinvolgono, considerando tra questi anche l’equilibrio:

* **Visual UI (vista);**
* **Tactile UI (tatto);**
* **Auditory UI (udito);**
* **Olfactory UI (olfatto);**
* **Gustatory UI (gusto);**
* **Equilibrial UI (equilibrio).**

**COMPOSITE USER INTERFACES (CUI)**

Prendono il nome di **Composite User Interfaces (CUI)** le interfacce utente che coinvolgono più di un senso. Alcuni esempi sono:

* **Graphical User Interfaces (GUI)**:interfacce utente che coinvolgono la vista e il tatto (e cioè, sono composte da elementi grafici e tattili);
* **Multimedia User Interfaces (MUI)**: interfacce utente che coinvolgono la vista, il tatto e l’udito (e cioè, sono composte da elementi grafici, tattili e uditivi).

E’ possibile classificare le CUI in tre macrocategorie:

* **Standard:** sfruttano dispositivi standard come tastiere, mouse, monitor. La CUI, dunque, è reale;
* **Virtual:** bloccano all’utente l’interazione con il mondo reale e creano un mondo virtuale.   
  La CUI, dunque, è virtuale;
* **Augmented:** non bloccano all’utente l’interazione con il mondo reale, ma la vanno ad arricchire.   
  La CUI, dunque, è un mix di elementi reali e virtuali.

Immagine che contiene schermata, Software multimediale, testo, Software per la grafica

Descrizione generata automaticamente

E’ possibile inoltre classificare le CUI in base al numero di sensi che coinvolgono (*es.* una CUI che coinvolge   
3 sensi è 3S; una CUI che coinvolge 4 sensi è 4S…). Una **Qualia interface** è una CUI che coinvolge tutti i sensi.

**NATURAL USER INTERFACE (NUI)**

Una **Natural User Interface (NUI)** è un’interfaccia utente dove l’interazione è diretta, intuitiva e consistente con il nostro comportamento “naturale”.

Interfacce del genere permettono:

* Basso sforzo cognitivo.
* Apprendimento progressivo;
* Passaggio da principiante ad esperto più rapido possibile.

**STRUTTURA DI UN’INTERFACCIA** *Lezione 11*

La **struttura di un’interfaccia** riporta come sono organizzati i vari elementi dell'interfaccia e come questi sono collegati fra di loro.

Si distinguono le seguenti strutture:

* **Struttura gerarchica**: l’utente comincia a navigare da un’informazione generale, per poi addentrarsi in informazioni via via sempre più dettagliate;

Immagine che contiene design

Descrizione generata automaticamente con attendibilità bassa

* **Struttura sequenziale:** l’utente naviga tra le informazioni passo per passo, seguendo un percorso prestabilito;

****

* **Struttura a matrice:** l’utente può navigare tra le informazioni seguendo il percorso che vuole, in quanto le informazioni sono collegate tra loro in molteplici modi;

**Immagine che contiene testo, diagramma, Carattere, schermata

Descrizione generata automaticamente**

* **Struttura a database:** si basano su una rappresentazione delle informazioni sotto forma di tabella (*es.* Excel ha una struttura a database).

Immagine che contiene testo, Diagramma, schermata, numero

Descrizione generata automaticamente

**ARCHITETTURA DELL’INFORMAZIONE**

L’**architettura dell’informazione** indica come sono organizzate e rappresentate le informazioni, e come gli utenti le ricercano e navigano tra di esse.

Si compone dunque dei seguenti elementi:

* **Schemi organizzativi:** indicano come sono organizzate le informazioni. Sono di due tipi:
  + **Schemi esatti:** le informazioni sono organizzate in modo oggettivo. Alcuni esempi sono:
    - **Schema alfabetico:** le informazioni sono organizzate in ordine lessicografico;
    - **Schema cronologico:** le informazioni sono organizzate in base alla data;
    - **Schema geografico:** le informazioni sono organizzate in base alla posizione.
  + **Schemi soggettivi:** le informazioni sono organizzate in modo diverso per ogni utente.
* **Sistemi di labeling:** indicano come sono rappresentate le informazioni;
* **Sistemi di ricerca:** indicano come gli utenti possono ricercare le informazioni;
* **Sistemi di navigazione:** indicano come gli utenti possono navigare tra le informazioni.   
  Una buona navigazione si basa su due proprietà:
  + **Findability:** l’utente deve essere in grado di trovare con facilità l’informazione che sta cercando;
  + **Discoverability:** l’utente deve essere in grado di scoprire con facilità nuove informazioni.

**USERS BEHAVIOUR PATTERNS**

Nell’interazione con un’interfaccia, gli utenti presentano alcuni comportamenti comuni. Considerando per esempio l’interazione con un’interfaccia per ricercare informazioni, alcuni comportamenti comuni sono i seguenti:

* **Quit:** l’utente effettua una ricerca, vede i risultati, e termina l’interazione (probabilmente non ha trovato i risultati che gli interessavano);
* **Narrow:** l’utente effettua una ricerca, vede i risultati, e usa delle opzioni per limitare il numero di risultati (*es.* per vedere solo i risultati dell’ultima settimana);
* **Expand:** l’utente effettua una ricerca, vede i risultati, e usa delle opzioni per espandere il numero di risultati (*es.* fa sì che vengano mostrati anche i risultati correlati);
* **Pearl growing:** l’utente effettua una ricerca, vede i risultati, ne apre uno, e apre un link all’interno del risultato (tipico comportamento che si ha quando si naviga su Wikipedia);
* **Pogosticking:** l’utente effettua una ricerca, vede i risultati, e ripetutamente apre un risultato e torna a vedere tutti i risultati;
* **Thrashing:** l’utente effettua una ricerca, vede i risultati, ed effettua una nuova ricerca, raffinando la query precedente (probabilmente i risultati della ricerca effettuata prima non l’avevano soddisfatto)
* **Berry picking:** l’utente effettua una ricerca, vede i risultati, ne apre uno, e dall’informazione che ha appreso effettua una nuova ricerca, raffinando la query precedente (probabilmente ora sa meglio cosa ricercare).

**DESIGN PATTERNS**

Similmente, le interfacce presentano alcune caratteristiche comuni. Considerando per esempio le interfacce che permettono di ricercare informazioni, alcune caratteristiche comuni sono le seguenti:

* **Autocomplete:** l’interfaccia completa la query mentre l’utente la sta ancora scrivendo;
* **Instant Results:** l’interfaccia fornisce dei risultati mentre l’utente sta ancora scrivendo la query;
* **Did you mean?:** dopo che l’utente ha scritto la query, l’interfaccia fornisce suggerimenti su quale sarebbe dovuta essere la query più appropriata (è utile se la query scritta dall’utente presenta errori di battitura);
* **Best First:** dopo che l’utente ha scritto la query, l’interfaccia mostra come primi risultati quelli che sono i migliori secondo una certa metrica (*es.* la rilevanza, la data, il formato…);
* **Advanced Search:** l’interfaccia permette di raffinare la ricerca prima ancora di eseguirla.

**WIREFRAME** *Lezione 12*

Un **wireframe** è una bozza grafica dell’interfaccia che ha lo scopo di mostrare quella che dovrà essere la sua struttura e il suo layout.

Immagine che contiene diagramma, testo, linea, modello

Descrizione generata automaticamente

**RESPONSIVE DESIGN**

Il **responsive design** consiste nel progettare il layout dell’interfaccia in modo che questo si adatti automaticamente alle dimensioni e alla risoluzione del dispositivo utilizzato per interagirci.

Esistono varie tecniche per garantire un responsive design:

* **Flexible grid:** viene usata una griglia che si adatta automaticamente alle dimensioni dello schermo;

**Immagine che contiene testo, Sito Web, software, Pagina Web

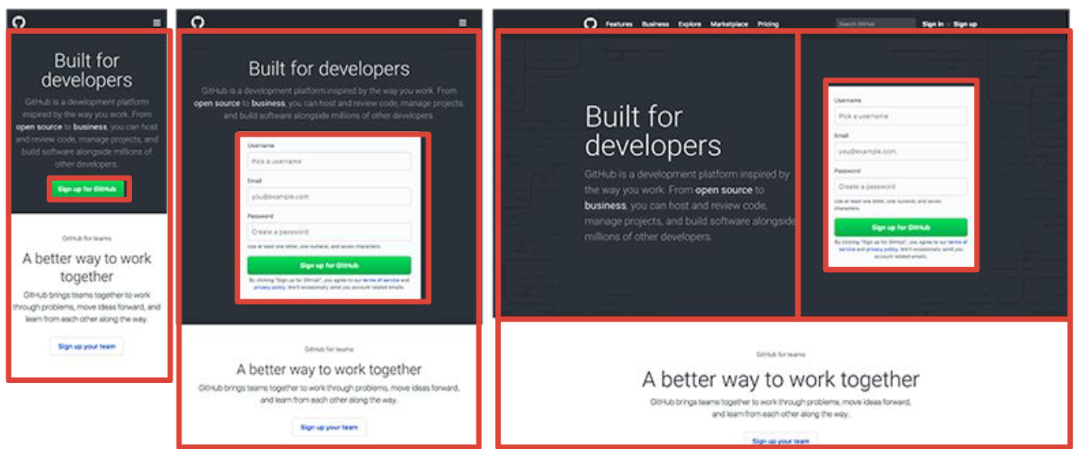
Descrizione generata automaticamente**

* **Flexible images:** viene fatto in modo che le immagini si adattino automaticamente alle dimensioni dello schermo;

**Immagine che contiene testo, Sito Web, Pagina Web, Pubblicità online

Descrizione generata automaticamente**

* **Media queries and breakpoints:** viene fatto in modo che lo stile cambi automaticamente a seconda delle proprietà del dispositivo.



**ADAPTIVE DESIGN**

L’**adaptive design** consiste nel progettare più layout dell’interfaccia, uno per ogni dispositivo utilizzato per interagirci.

**ACCESSIBILITY**

L’**accessibility** consiste nel progettare il layout dell’interfaccia in modo che tutti possano interagire con essa (anche chi ha problemi visivi, motori, uditivi, cognitivi…).

Esistono varie tecniche per garantire l’accessibility:

* Aggiungere testo alternativo alle immagini;
* Aggiungere sottotitoli ai video;
* Rendere il contenuto facilmente leggibile (*es.* scegliendo un font comprensibile);
* …

**DAI WIREFRAME AI WIREFLOW**

Una volta prodotta la bozza grafica dell’interfaccia tramite un wireframe, bisogna produrre gli *user flow*.   
Un **user flow** è una rappresentazione schematica di un possibile percorso di utilizzo dell'interfaccia da parte di un utente.

E’ possibile descrivere un user flow attraverso un *flow chart*:

Immagine che contiene testo, schermata, diagramma, Carattere

Descrizione generata automaticamente

Un **wireflow** è un particolare tipo di flow chart che sfrutta il wireframe:

Immagine che contiene schermata, Blu elettrico, blu, Rettangolo

Descrizione generata automaticamente

**MOCKUP**

Un **mockup** è una bozza grafica dell’interfaccia che include più dettagli rispetto ad un wireframe   
(immagini, colori…). Come un wireframe, ha lo scopo di mostrare quella che dovrà essere la struttura e il layout dell’interfaccia.

**HUMAN INTERFACE DEVICES (HID)** *Lezione 18 - inizio 19*

Un **HID** è un dispositivo fisico tramite il quale l’utente interagisce con il prodotto.

Vediamo alcune categorie di HID.

**DISPOSITIVI “TESTI E CARATTERI”**

**TASTIERE**

La **tastiera** è uno dei dispositivi “testi e caratteri” più comuni.



Ogni tastiera dispone di tre layout.

* **Layout fisico:** il posizionamento fisico dei tasti;
* **Layout visuale:** il posizionamento delle etichette sui tasti;
* **Layout funzionale:** le associazioni tasto – funzione, che cambiano a seconda del software utilizzato (*es.* su Youtube il tasto *spazio* permette di mettere in pausa un video; su Word, invece, lo stesso tasto permette di stampare a video un carattere vuoto).

**LETTORI DI CODICE A BARRE**

Un **lettore di codice a barre** è uno scanner che legge i codici a barre, li decodifica in una serie di caratteri e invia questi caratteri a un computer.



**TAG RFID**

Un **tag RFID** è un piccolo transceiver radio (trasmettitore e ricevitore). Quando riceve delle onde radio da un lettore RFID nelle vicinanze, il tag trasmette dei dati al lettore.



Esistono due tipi di tag RFID:

* **Tag RFID passivi**: sono alimentati dalle onde radio ricevute dai lettori RFID;
* **Tag RFID attivi**: sono alimentati a batteria.

**DISPOSITIVI DI PUNTAMENTO**

Un **dispositivo di puntamento** è un HID che trasferisce input di tipo spaziale verso un computer.

Immagine che contiene schizzo, testo, linea, diagramma

Descrizione generata automaticamenteI dispositivi di puntamento si basano sulla **Legge di Fitts:** una legge che predice il tempo che ci vuole per muovere rapidamente un dispositivo da un punto iniziale a un’area obiettivo che presenta una certa estensione:

Dove:

* ***a*** è il tempo che ci vuole per iniziare a far muovere il dispositivo   
  (varia da dispositivo a dispositivo);
* ***b***è la velocità del dispositivo (varia da dispositivo a dispositivo);
* ***D***è la distanza tra il punto iniziale e il centro dell’area obiettivo;
* ***W***è la dimensione principale dell’area obiettivo.

Esistono vari criteri per classificare i dispositivi di puntamento:

* Sulla base del **tipo di input:**
  + **Diretto:** il puntatore a schermo è nella stessa posizione fisica del dispositivo di puntamento (*es.* il pennino di un tablet);
  + **Indiretto:** il puntatore a schermo non è nella stessa posizione fisica del dispositivo di puntamento (*es.* il mouse di un computer);
* Sulla base del **modo in cui il movimento del dispositivo di puntamento viene mappato nel movimento del puntatore a schermo**:
  + **Assoluto:** tanto muoviamo il dispositivo di puntamento, tanto si muove il puntatore a schermo (*es.* una tavoletta grafica da 14 pollici su uno schermo da 14 pollici);
  + **Relativo:** il puntatore a schermo si muove di più o di meno di quanto muoviamo il dispositivo di puntamento (*es.* una tavoletta grafica da 14 pollici su uno schermo da 28 pollici);
* Sulla base del **modo in cui dobbiamo interagire con il dispositivo di puntamento per far muovere il puntatore a schermo:**
  + **Isotonico:** il dispositivo di puntamento è mobile, e il puntatore a schermo si muove sulla base dello spostamento applicato al dispositivo di puntamento (*es.* un mouse);

*Stessa forza*

* + **Isometrico:** il dispositivo di puntamento è immobile, e il puntatore a schermo si muove in maniera proporzionale alla forza applicata al dispositivo di puntamento (*es.* un trackpoint di un portatile);

*Stessa posizione*

* + **Elastico:** il dispositivo è semi-mobile, e il puntatore a schermo si muove in maniera proporzionale allo spostamento applicato al dispositivo di puntamento (*es.* un joystick a molla).

*Proporzionale a spostamento*

* Sulla base di **quali proprietà del puntatore a schermo possiamo controllare interagendo con il dispositivo di puntamento:**
  + **Position control:** interagendo con il dispositivo di puntamento è possibile controllare la posizione del puntatore a schermo (*es.* un mouse);
  + **Rate control:** interagendo con il dispositivo di puntamento è possibile controllare la velocità e la direzione di spostamento del puntatore a schermo (*es.* un joystick).

**DISPOSITIVI DI PUNTAMENTO INNOVATIVI: EYE TRACKER**

Gli **eye-tracker** sono dei dispositivi di puntamento che considerano, come input di tipo spaziale, la posizione della pupilla.



Esistono tre modi misurare la posizione della pupilla:

* **Bright-pupil**: l’eye-tracker illumina la pupilla con una luce infrarossa coassiale;
* **Dark-pupil**: l’eye-tracker illumina la pupilla con luce infrarossa non coassiale.
* **Passive light**: viene utilizzata la luce ambientale.

Il modo bright-pupil è chiaramente quello che crea la migliore immagine della pupilla (e quindi ha performance migliori), ma richiede un hardware più complesso.

**DISPOSITIVI DI PUNTAMENTO INNOVATIVI: GAZE TRACKER**

I **gaze-tracker** sono dei dispositivi di puntamento che considerano, come input di tipo spaziale, dove il soggetto sta guardando (e quindi, oltre a misurare la posizione della pupilla, hanno bisogno di conoscere anche l’angolo e la posizione della testa del soggetto).

**DISPOSITIVI AUDIO E INTERFACCE VOCALI**

Un **microfono** è un dispositivo audio che converte il suono in segnali elettrici.

Un **array di microfoni** è un dispositivo audio costituito da più microfoni (tipicamente omnidirezionali) che operano in parallelo. Viene utilizzato per riconoscere la direzione di provenienza del suono, per la cancellazione dell’eco…

I dispositivi audio permettono di implementare interfacce vocali; queste sono efficaci, ma solo se applicate in specifici contesti, e soprattutto quando l’interazione fra utente e dispositivo richiede lo scambio di poche informazioni (altrimenti si userebbe troppa banda, si effettuerebbe uno sforzo cognitivo molto alto e si farebbero tanti errori, visto che interagire vocalmente con un computer non è sempre molto intuitivo…).

**SENSORI DI IMMAGINI**

**SCANNER 3D**

Uno **scanner 3D** è un sensore di immagini che permette di raccogliere informazioni sulla dimensione, sulla posizione e/o sul colore di oggetti del mondo reale.

Si distinguono due categorie di scanner 3D:

* Immagine che contiene diagramma, linea, Diagramma, Parallelo

  Descrizione generata automaticamente**Scanner 3D passivi:** non emettono radiazioni elettromagnetiche, affidandosi alla luce ambientale. Alcuni esempi sono:
  + **Basati su camere stereoscopiche:** usano due camere, poste ad una certa distanza l’una dall’altra, che acquisiscono immagini della stessa scena. Calcolando la differenza nelle immagini pixel per pixel, questi scanner riescono a determinare la distanza di ogni oggetto nella scena;
  + **Sistemi fotometrici**: usano una camera che acquisisce più immagini della stessa scena sotto condizioni di luce differenti. In questo modo, questi scanner riescono a determinare dalle immagini la distanza e il colore di ogni oggetto nella scena;

*Foto = luce*

Immagine che contiene Viso umano, schermata, Fronte, uomo

Descrizione generata automaticamente

*Pensa alla silhouette della statua cui sono state fatte acquisizioni da più angoli diversi*

* + **Basati su tecniche silhouette**: usano una camera che acquisisce più immagini della stessa scena da angoli differenti. In questo modo, questi scanner riescono a ricostruire dalle immagini ogni oggetto nella scena;

Immagine che contiene schizzo, arte, origami, diagramma

Descrizione generata automaticamente

* **Scanner 3D attivi:** emettono radiazioni elettromagnetiche.

Alcuni esempi sono:

* + **Basati su tecnologia *Time-of-flight*:** emettono un fascio laser sulla scena, e calcolano il tempo impiegato dal segnale per essere riflesso e tornare ad un ricevitore. In questo modo, questi scanner riescono a determinare la distanza di ogni oggetto nella scena;

**Immagine che contiene testo, Carattere, diagramma, linea

Descrizione generata automaticamente**

* + **A triangolazione:** emettono un fascio laser sulla scena, e sfruttano una camera per vedere dove cade il punto laser. In questo modo, questi scanner riescono a determinare la distanza di ogni oggetto nella scena;

**Immagine che contiene schermata, diagramma, testo, linea

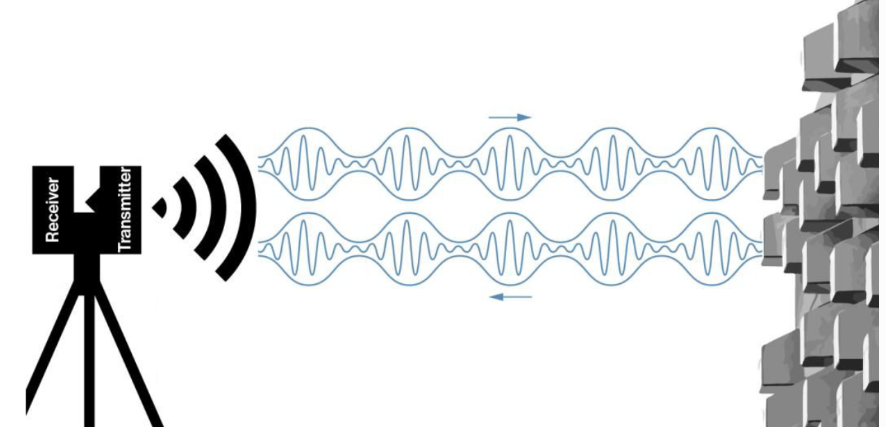
Descrizione generata automaticamente**

* + **A luce strutturata:** emettono un fascio di luce strutturata sulla scena - e cioè, un’immagine che ha una certa struttura geometrica - e sfruttano una camera per valutare come i vari oggetti della scena hanno deformato l’immagine emessa. In questo modo, questi scanner riescono a determinare la distanza di ogni oggetto nella scena;

Immagine che contiene schizzo, sedia, design, illustrazione

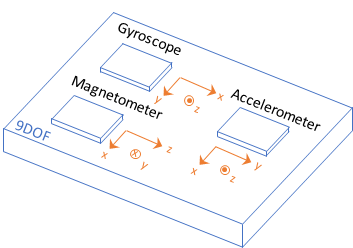
Descrizione generata automaticamente

* + **A luce modulata:** emettono un fascio di luce modulata sulla scena - e cioè, impulsi luminosi a diversa frequenza - e misurano come cambia la frequenza degli impulsi dopo che questi sono stati riflessi e sono tornati ad un ricevitore. In questo modo, questi scanner riescono a determinare la distanza di ogni oggetto nella scena.

****

**INERTIAL MEASUREMENT UNIT (IMU)**

Una ***IMU*** è un HID che misura la forza, la velocità angolare e l’orientamento di un corpo usando un accelerometro, un giroscopio e un magnetometro.



Un IMU è in grado di misurare 9 gradi di libertà tramite un accelerometro a 3 assi, un giroscopio a 3 assi e un magnetometro a 3 assi.

**DISPOSITIVI WEARABLE**

Un **dispositivo wearable** è un HID in cui la parte sensoriale, di computing e di interfaccia sono tutte riunite e vengono indossate dall’utente.

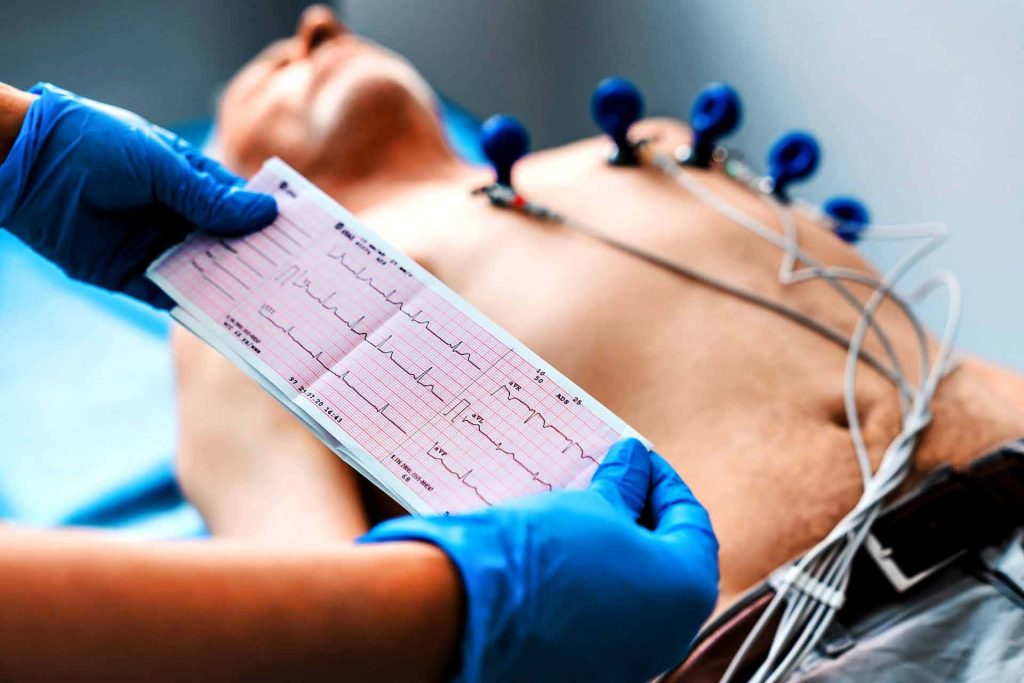
**DISPOSITIVI PER LA MISURA DEL BATTITO CARDIACO TRAMITE PPG (PHOTOPLETHYSMOGRAPHY)**

Un **dispositivo per la misura del battito cardiaco tramite PPG** è un dispositivo wearable che misura il battito cardiaco dell’utente illuminando una zona corporea (*es.* il polso) con luce verde o rossa e analizzando l’immagine riflessa.



**DISPOSITIVI PER LA MISURA DEL BATTITO CARDIACO TRAMITE ECG (ELETTROCARDIOGRAMMA)**

Un **dispositivo per la misura del battito cardiaco tramite ECG** è un dispositivo wearable che misura il battito cardiaco dell’utente usando degli impulsi elettrici per capire quando il muscolo cardiaco si espande e si contrae.



**HEADSET EEG (ELETTROENCEFALOGRAFIA)**

Un **headset EEG** è un dispositivo wearable usato per monitorare gli impulsi elettrici provenienti dal cervello.   
Viene usato come dispositivo di puntamento per utenti disabili.



**METODI PER L’INNOVAZIONE** *Lezione 14*

Si distinguono due tipi di innovazione:

* **Sustaining innovation:** è un’innovazione basata sul creare un prodotto che apporta piccole migliorie a prodotti già esistenti. Un’innovazione del genere è poco rischiosa, in quanto non richiede un costo troppo elevato, e i clienti saranno più o meno gli stessi dei prodotti già esistenti.

*Innovazione “incrementale”*

* **Disruptive innovation:** è un’innovazione basata sul creare un prodotto che si discosta totalmente dai prodotti già esistenti. Un’innovazione del genere è molto rischiosa, in quanto potrebbe richiedere un costo molto elevato, e non è possibile prevedere con certezza quali saranno i possibili clienti.

*Innovazione “dirompente”*

**METODI PER L’ORGANIZZAZIONE DI PROGETTI RELATIVI A PRODOTTI INNOVATIVI**

Introduciamo ora una serie di metodi per l’organizzazione di progetti relativi a prodotti innovativi.

**HUMAN CENTERED DESIGN PROCESS**

Lo **Human Centered Design Process** è un primo metodo (framework) per l’organizzazione di progetti incentrati sull’utente.

E’ composto da tre fasi:

* **Inspiration:** viene osservato l’utente e i suoi bisogni;
* **Ideation:** si discutono di tutte le possibili soluzioni ai bisogni dell’utente individuati nella fase precedente;
* **Implementation:** viene realizzato un prodotto che implementi una delle soluzioni individuate nella fase precedente.

In queste fasi si segue il cosiddetto **modello a doppio diamante**:

* Si diverge inizialmente discutendo di tutte le possibili soluzioni ai bisogni dell’utente, finendo poi per convergere verso una certa soluzione;
* Si diverge nuovamente discutendo di tutti i possibili modi di implementare la soluzione scelta, finendo poi per convergere verso un certo modo di implementarla.

Immagine che contiene testo, linea, Carattere, diagramma

Descrizione generata automaticamente

**DESIGN THINKING**

Il **Design Thinking** è un secondo metodo (framework) per l’organizzazione di progetti incentrati sull’utente.

E’ composto da cinque fasi:

* **Empathize:** viene osservato l’utente;
* **Define:** vengono identificati i bisogni dell’utente;
* **Ideate:** si discutono di tutte le possibili soluzioni ai bisogni dell’utente individuati nella fase precedente;

***EDIPT*** (Piaf)

* **Prototype:** viene realizzato un prototipo che implementi una delle soluzioni individuate nella fase precedente;
* **Test:** si fa testare il prototipo agli utenti. Si riparte poi dalla prima fase, tenendo conto dei risultati di questo test, così da migliorare continuamente il prototipo e arrivare così al prodotto finale.

E’ possibile unire Human Centered Design Process e Design Thinking, ottenendo il **social enterprise process.** La mappatura di un metodo nell’altro è la seguente:

Design Thinking presenta in più la fase di *Test*.   
In Human Centered Design Process questa fase non è esplicitamente indicata, ma è sottointeso che il test va sempre fatto.

* Inspiration: Empathize e Define;
* Ideation: Ideate;
* Implementation: Prototype.

**METODI PER LO SVILUPPO DI PROGETTI RELATIVI A PRODOTTI INNOVATIVI**

Introduciamo ora una serie di metodi per lo sviluppo di progetti relativi a prodotti innovativi.

**WATERFALL**

Il **metodo Waterfall (a cascata)** è il più tradizionale metodo di sviluppo SW.

Prevede una sequenza di fasi:

* Raccolta e analisi dei requisiti;
* Progettazione;
* Sviluppo;
* Collaudo;
* Manutenzione.

**AGILE**

**Agile** è un metodo di sviluppo che si focalizza sul testare il prima possibile.   
Il team di sviluppo definisce dei requirements, sviluppa una prima parte del prodotto che fa testare agli utenti, e sulla base dei risultati del test definisce nuovi requirements per realizzare nuove funzioni o migliorare quelle già presenti (che vengono poi sviluppate, testate, e così via).

**SCRUM**

**Scrum** è un metodo di sviluppo nato da Agile. Adatto per team di sviluppo piccoli (<10 persone), prevede di dividere il progetto in blocchi detti *Sprint*. Ogni giorno vengono fatti dei meeting (*daily scrum*) della durata di 15 minuti per tenere traccia dello sviluppo degli Sprint, e al termine di uno Sprint vengono fatti dei meeting per fare il bilancio dello Sprint (*Sprint retrospective*) e mostrare i risultati ottenuti (*Sprint review*).

**DEVOPS**

**DevOps** è un metodo di sviluppo nato dall’integrazione di Agile con il mondo del Cloud. Si focalizza quindi sul testare il prima possibile, sfruttando i vantaggi del Cloud.

**IOT E INDUSTRIA 4.0** *Lezione 15*

L’**IoT** è un sistema di dispositivi interconnessi, ciascuno dei quali è dotato di un identificatore univoco (detto UID) ed è in grado di trasferire dati sulla rete senza richiedere l’interazione uomo-uomo o uomo-computer.

Ogni prodotto IoT è dotato di:

* **Sensori:** catturano dati dal mondo reale e li convertono in segnali elettrici;
* **Attuatori:** convertono segnali elettrici in azioni fisiche sul mondo reale.

L’**industria 4.0** è la tendenza dell'odierna automazione industriale a integrare le nuove tecnologie, tra cui l’IoT.

Il cuore dell’industria 4.0 è il *Digital Twin*.Un **Digital Twin** è una copia digitale di un bene, macchina o processo reale esistente. La comunicazione tra l’oggetto fisico e il suo Digital Twin è continua e bidirezionale, e dunque il Digital Twin viene usato per controllo remoto, simulazioni e manutenzione.

**DIFFICOLTA’ NELLA PROGETTAZIONE DI UN PRODOTTO IOT**

Progettare un prodotto IoT presenta diverse difficoltà:

* Un prodotto IoT è costituito da più dispositivi, ciascuno dei quali deve essere progettato in maniera precisa, così che possa svolgere lo specifico compito per cui è deputato. Ognuno di questi dispositivi, poi, è dotato di una propria interfaccia, e non è detto che tutte le funzionalità del prodotto siano disponibili su tutte le interfacce. Serve dunque garantire *l’interusabilità*, e cioè una UX univoca e coerente su ciascuna interfaccia.
* Bisogna progettare bene l’abilità di fare da ponte tra il mondo digitale e il mondo reale:
  + Bisogna progettare bene i sensori, in modo che catturino tutti e soli i dati utili;
  + Bisogna progettare bene gli attuatori, tenendo conto che le azioni svolte da questi sul mondo reale potrebbero non essere reversibili.
* Bisogna tenere conto delle problematiche dovute alla rete.
* Bisogna fare in modo che il consumo energetico sia il minore possibile. A tale scopo, i prodotti IoT vengono spesso progettati in modo che si connettano alla rete ad intermittenza, così da avere un grande risparmio energetico.

**PRETOTIPI** *Lezione 16*

Un **pretotipo** è un semplice modello del prodotto che si vorrebbe sviluppare. E’ utile per testare molte idee velocemente, in modo da vedere se sono fallimentari o meno già all’inizio della fase di progettazione.

Gli step per produrre un buon pretotipo sono i seguenti:

1. **Individuare le funzionalità chiave che dovrà avere il prodotto;**

***F****uck*

***TIT***

1. **Scegliere un tipo di pretotipo per testarle;**
2. **Fare ipotesi di mercato:** quante e quali tipi di persone utilizzeranno il pretotipo? Come lo utilizzeranno?
3. **Testare:** si fa testare il pretotipo alle persone. Si riparte poi dalla prima fase, tenendo conto dei risultati di questo test, così da migliorare continuamente il pretotipo.

Vediamo alcuni tipi di pretotipi:

* **Pretotipo *fake door*:** si crea una descrizione in poche parole e con poche immagini del prodotto.

**Immagine che contiene testo

Descrizione generata automaticamente**

Si ha l’idea di un servizio che ti invia i pannolini direttamente a casa, senza dover andare al negozio per comprarli ogni volta. Qualche persona potrebbe essere interessata? Vediamolo con un pretotipo *fake door*: creiamo una pagina Web in cui si descrive l’idea in poche parole e con poche immagini.

* **Pretotipo *mechanical Turk*:** si usano le abilità umane per simulare quello che dovrebbe fare il prodotto.

***Immagine che contiene testo, lettera, design, catalogo

Descrizione generata automaticamente***

Si ha l’idea di un prodotto Text-to-speech. Creare anche un modello base di un prodotto del genere potrebbe essere molto costoso, dunque come possiamo vedere senza spendere troppe risorse se qualche persona potrebbe essere interessata? Utilizziamo un pretotipo *mechanical Turk*: le persone scrivono del testo, e poi facciamo leggere questo testo ad un’altra persona, per simulare quello che dovrebbe fare il prodotto.

* **Pretotipo *Impersonator*:** si crea un modello del prodotto modificando leggermente un altro prodotto già esistente.

Si ha l’idea di una nuova macchina con motore elettrico. Come possiamo farla sperimentare alle persone? Creiamo un pretotipo *Impersonator*: prendiamo un’altra macchina già esistente, sostituiamo il motore con un motore elettrico e la facciamo provare alle persone.

***Immagine che contiene trasporto, veicolo, Veicolo terrestre, ruota

Descrizione generata automaticamente***

* **Pretotipo *Pinocchio*:** si crea un modello chiaramente finto del prodotto, che serve tuttavia a trasmettere dettagli sul prodotto che non si riuscirebbero a spiegare in altri modi (*es.* a voce).

***Immagine che contiene Attrezzature mediche, strumento, medico, assistenza sanitaria

Descrizione generata automaticamente***

Si ha l’idea di un nuovo bisturi con un’impugnatura “a pistola”. Come possiamo trasmettere questo dettaglio sul prodotto? Creiamo un pretotipo *Pinocchio*: costruiamo un modello simile ad una pistola giocattolo, così che le persone possano immaginare la reale esperienza che avranno poi con il prodotto.

* **Pretotipo *One Night Stand*:** si crea un modello del prodotto secondo uno dei modi visti in precedenza e lo si fa testare ad una platea selezionata di persone in una finestra di tempo limitata.
* **Pretotipo *Facade:***si crea un modello del prodotto secondo uno dei modi visti in precedenza affittando o prendendo in prestito dell’attrezzatura e spazi costosi, così da dare un’immagine solida dell’azienda.

Dopo diverse fasi di pretotipazione è possibile costruire un **Minimum Viable Product (MVP),** e cioè una versione minimale del prodotto contenente solo ed esclusivamente le funzionalità che si sono pretotipate nelle fasi precedenti. Non si ha ancora per le mani un prodotto definitivo, ma un qualcosa di vendibile, in modo da ottenere dell’utile che, se sufficiente, permetterebbe la produzione definitiva del prodotto.

**PROTOTIPI** *Lezione 17*

Un **prototipo** è un semplice modello del prodotto che si vorrebbe sviluppare. E’ utile per testare molte idee velocemente, in modo da vedere se sono fallimentari o meno già all’inizio della fase di progettazione.

In base alla *fedeltà*, e cioè quanto i prototipi assomigliano a quello che sarà il prodotto finale,   
è possibile dividere i prototipi in due categorie:

1. **Prototipi a bassa fedeltà** 
   * **Vantaggi:**
     + Sono fatti in poco tempo e in maniera facile (spesso con carta e penna);
     + Sono facili da modificare.
     + Essendo dotati di pochi elementi, gli utenti si concentreranno sui dettagli che sono davvero importanti;
     + Vengono usati per collegare i wireframe tra loro.
   * **Svantaggi:**
     + Potrebbero essere troppo semplici, dunque potrebbe essere difficile per gli utenti dare un feedback;
     + Spesso non offrono alcuna funzionalità funzionante (*es.* considerando un prototipo fatto su carta e penna sfruttando i wireframe, deve essere il designer a simulare l’interazione tra l’utente e il prototipo, ad esempio passando da un wireframe ad un altro quando l’utente preme un certo bottone disegnato), e la mancanza di interattività è un altro elemento che potrebbe rendere difficile per gli utenti dare un feedback.

Tolta la roba dei wireframe, i vantaggi di una categoria sono gli svantaggi dell’altra,   
e viceversa.

1. **Prototipi ad alta fedeltà** 
   * **Vantaggi:**
     + Essendo molto simili a quello che sarà il prodotto finale, è più semplice per gli utenti dare un feedback;
     + Gli utenti possono testare molti elementi (la leggibilità, funzionalità funzionanti offerte…)
   * **Svantaggi:**
     + Ci vuole tempo per farli, e non sono semplici da fare;
     + Non sono facili da modificare;
     + Essendo dotati di molti elementi, potrebbe succedere che gli utenti finiscano col concentrarsi su dettagli superficiali, trascurando quelli più importanti.

Sono 6 gli elementi principali che influiscono sulla fedeltà di un prototipo (detti *gradi di fedeltà*):

1. **Visual realism**: quanto il prototipo assomiglia esteticamente a quello che sarà il prodotto finale;
2. **Scope (portata)**:l'ampiezza e la profondità del design rappresentato nel prototipo. Sono due i possibili scope:
   1. **Horizontal scope:** il prototipo fornisce una visione generale di quello che sarà il prodotto finale, senza scendere troppo in dettaglio sulle singole funzionalità;

***V****isual* ***S****tudio*

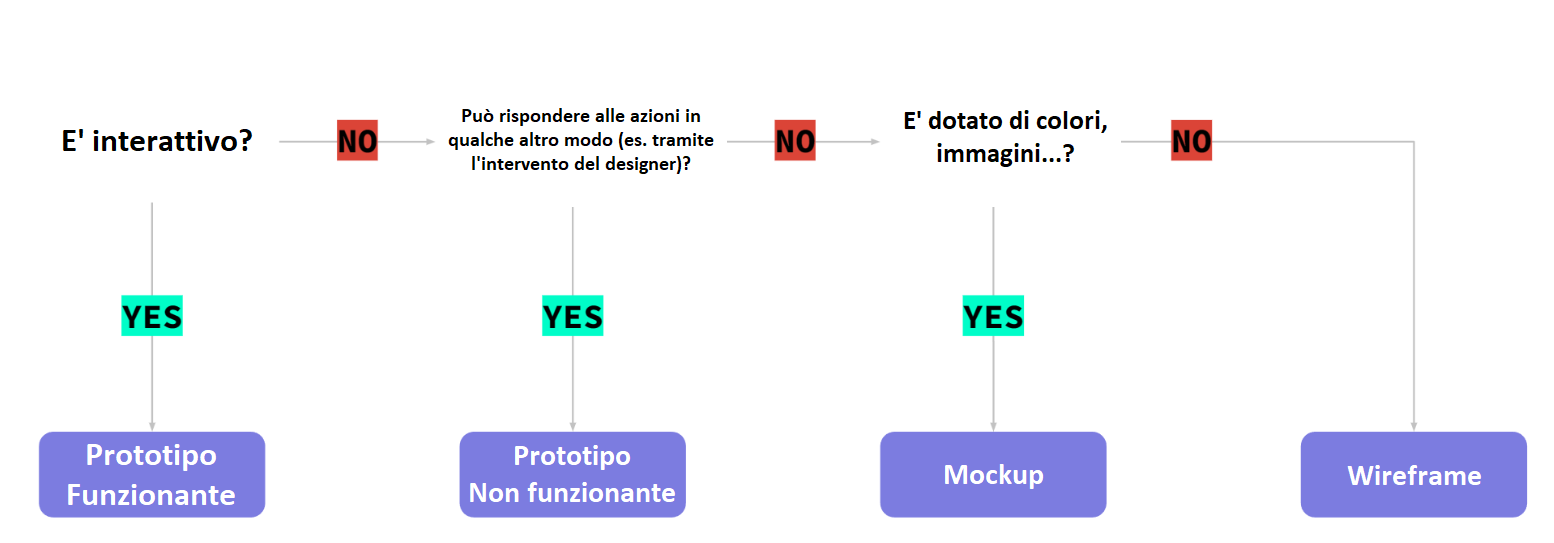
***FA***

***D****io****P****orcare*

* 1. **Vertical scope:** il prototipo fornisce una visione dettagliata di una specifica funzionalità, tralasciando le altre.

1. **Functionality:** cosa funziona effettivamente nel prototipo. Si distinguono a tal proposito due tipi di prototipi:
   1. **Prototipi funzionanti:** presentano qualche funzionalità funzionante;
   2. **Prototipi non funzionanti:** non presentano alcuna funzionalità funzionante.
2. **Autonomy:** se il prototipo è in grado di operare autonomamente o se richiede l’intervento del designer.
3. **Data:** se il prototipo è in grado di operare su dati reali oppure devono essere scelti “ad arte”;
4. **Platform:** se la piattaforma usata per testare il prototipo è provvisoria o è quella finale (*es.* se si testa un’applicazione per cellulare su un computer, questa è una piattaforma provvisoria; se invece tale applicazione si testa su un cellulare, questa è la piattaforma finale).

**DIFFERENZA TRA WIREFRAME, MOCKUP E PROTOTIPI**

Qual è la differenza tra un wireframe, un mockup e un prototipo? Basta seguire questo schema:

**TEST DI USABILITA’** *Lezione 20*

Una volta prodotto un prototipo, bisogna farlo testare alle persone. In questo modo è possibile vedere, tra le varie cose, se il prototipo presenta problemi di *usabilità*.

L’**usabilità** è la capacità di un’interfaccia di essere usata facilmente dall’utente. Sono 5 gli elementi che influiscono sull’usabilità di un’interfaccia:

1. **Learnability:** quanto è facile per gli utenti completare compiti basilari la prima volta che interagiscono con l’interfaccia;
2. **Efficiency:** quanto tempo ci mettono gli utenti a completare dei compiti dopo aver appreso come interagire con l’interfaccia;

***LE***

***MES****se*

1. **Memorability:** quanto è facile per gli utenti riprendere a interagire con l’interfaccia dopo un periodo di non utilizzo;
2. **Errors:** quanti errori compiono gli utenti interagendo con l’interfaccia, quanto sono gravi e quanto è facile per loro rimediare;
3. **Satisfaction:** quando è soddisfacente per gli utenti interagire con l’interfaccia.

In un test dell’usabilità si hanno sempre 2 elementi:

* **Participants:** i partecipanti del test;
* **Facilitators:** guidano i partecipanti nel test, rispondendo alle loro domande, facendogli domande a loro volta e facendogli svolgere dei compiti sul prototipo. E’ importante che i facilitators non influenzino il comportamento dei partecipanti.

In base al tipo di dati che si raccolgono, si distinguono due tipi di test dell’usabilità:

* **Test qualitativi:** si raccolgono dati *qualitativi*, e cioè dati espressi senza usare alcuna metrica.   
  Questi dati possono essere raccolti tramite domande a risposta aperta ai partecipanti, osservazioni, feedback da parte dei partecipanti…
* **Test quantitativi:** si raccolgono dati *quantitativi*, e cioè dati espressi usando una o più metriche. Questi dati possono essere raccolti tramite domande a risposta chiusa ai partecipanti, questionari ai partecipanti…

Per pianificare un test dell’usabilità bisogna tenere conto di diversi elementi:

* **Bisogna definire l’obiettivo:** cosa si vuole testare?
* **Bisogna definire le modalità del test:** si deve effettuare il test in laboratorio o sul campo? Di persona o da remoto?
* **Bisogna definire il numero di partecipanti:** un test qualitativo non richiede molti partecipanti;   
  un test quantitativo, invece, ne richiede di più, in maniera da dare validità statistica alle metriche raccolte;
* **Bisogna scegliere i partecipanti giusti:** dovrebbero rispecchiare le personas definite nella fase di UX design;

Sono temporalmente successive.

* **Bisogna scegliere i compiti giusti:** devono servire a testare effettivamente ciò che si vuole testare;
* **Bisogna scegliere le metriche (in caso di test quantitativo);**
* **Conviene scrivere il piano del test:** una volta deciso come effettuare il test, conviene appuntarsi il tutto su un documento;
* **“Testare” il test:** prima di far provare il test ai partecipanti, è meglio effettuare un test “pilota”   
  (*es.* con poche persone), per vedere se il piano del test va bene (*es.* se modalità del test sono fattibili, i compiti sono chiari…);
* **E’ meglio non fare tutto da solo:** conviene sempre avere più persone che si occupano di svolgere il test, in modo da avere risultati più attendibili (facendo tutto da soli, per esempio, molti dati potrebbero sfuggire).

**VALUTAZIONE DELL’USABILITA’ TRAMITE EURISTICHE DI NIELSEN**

Prima di far testare il prototipo a persone reali, è possibile effettuare una prima valutazione della sua usabilità attraverso le cosiddette **euristiche di Nielsen:**

1. **Design minimalista:** l’interfaccia dovrebbe contenere solo gli elementi strettamente necessari;
2. **Riconoscere invece di ricordare:** bisogna rendere ben visibili tutti gli elementi dell’interfaccia, in modo che l’utente non debba memorizzare troppe informazioni per riuscire ad interagire con essa.
3. **Corrispondenza tra il sistema e il mondo reale:** bisogna usare parole e concetti familiari all’utente, evitando tecnicismi;
4. **Coerenza e standard:** l’utente non deve domandarsi se parole, situazioni o azioni diverse abbiano lo stesso significato. Bisogna dunque creare un’interfaccia coerente;
5. **Flessibilità ed efficienza d’uso:** bisogna creare delle shortcut per gli utenti esperti, in modo da velocizzare le loro interazioni più frequenti;

Sono logicamente successive.

1. **Visibilità dello stato del sistema:** l’utente deve essere informato su cosa sta facendo il sistema attraverso feedback immediati, informativi e semplici;
2. **Prevenzione degli errori:** bisogna ridurre al minimo la possibilità che l’utente possa compiere delle azioni per errore;
3. **Controllo e libertà dell’utente:** poiché gli errori possono comunque verificarsi, bisogna fornire e rendere visibile all’utente una “uscita di sicurezza” che permetta all’utente di annullare subito l’azione compiuta per errore, senza che egli debba passare attraverso una lunga sequenza di passi;
4. **Aiutare gli utenti a diagnosticare e recuperare dagli errori** : ogni messaggio di errore dovrebbe essere espresso in un linguaggio chiaro e semplice, indicando il problema e suggerendo una soluzione in maniera costruttiva;
5. **Aiuto e documentazione:** una buona interfaccia non ha bisogno di alcuna spiegazione per essere usata. Tuttavia, potrebbe essere utile fornire una documentazione per aiutare gli utenti ad interagire con essa.