

# INTERAZIONE UOMO-MACCHINA

## LEZ.1

L'interazione uomo-macchina si occupa della progettazione, implementazione e valutazione di sistemi di calcolo interattivo per uso umano e dello studio di principali fenomeni che li circondano.

---

## INTERFACCE UTENTE MOBILE

Quando si parla di interfacce utente mobile si possono toccare diversi aspetti:

- Approccio tecnico
- Progettazione

Ogni interfaccia utente viene visualizzata su un display, e questo ha specifiche peculiarità:

- Dimensione in pollici
- Risoluzione dei pixel
- Una gamma di colori vario in relazione al dispositivo.

L'interfaccia deve essere unica e deve adattarsi a qualsiasi dispositivo collegato.

## COERENZA COL SISTEMA

- L'utente deve comprenderne il linguaggio visuale, senza doverla studiare da zero.
  - Il primo passo per ottenere ciò è far sì che l'applicazione assomigli al resto del sistema.
  - L'interazione nel sistema si realizza con l'uso di icone comuni e conosciute.
- 

## LE GESTURE

- Quando si usa un dispositivo touch, è possibile associare specifici comandi ad opportune combinazioni di movimenti delle dita sul display.
- Le *gestures* comuni sono note agli utenti che usano tablet e smartphone.

Il progettista deve curare però **l'originalità** aggiungendo elementi innovativi.

Tutti i precedenti aspetti vanno curati in modo attento per poter consentire un'esperienza di interazione positiva. (lo chiede all'esame).

## Lez 2

L'essere umano si è sempre dotato di strumenti che gli permettessero di svolgere compiti che col solo impiego delle sue doti naturali sarebbero stati impossibili.

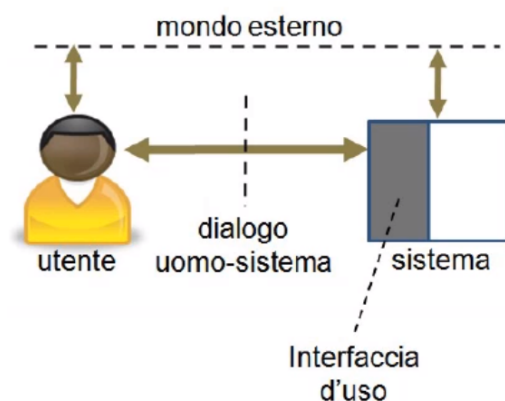
Un esempio sono gli utensili (coltelli, spade, frecce, tamburi...).

Negli ultimi secoli, l'evoluzione della tecnologia e le tecniche per la produzione di energia hanno alimentato progettazione e creazione di strumenti autonomi che svolgono azioni.

Solo da pochi decenni grazie all'informatica questi sistemi non svolgono solo azioni autonome ma anche intelligenti, attraverso dei SW.

---

Nascono così le interfacce che sono un filtro tra l'utente e il sistema.



Le tre componenti principali sono:

1. Sistema interattivo
2. Interfaccia d'uso
3. Dialogo (interazione tra utente e sistema)

Sistema interattivo: qualsiasi combinazione di componenti HW e SW che ricevono input da un utente, che forniscono output allo scopo di supportare l'azione di un task (processo).

Interfaccia d'uso: insieme di tutti i componenti che forniscono all'utente info e comandi per permettergli di effettuare specifici TASKS attraverso il sistema.

Dialogo: è l'interazione tra utente e sistema intesa come un'assequenza di azioni compiute dall'utente (input) e di risposte del sistema (output), allo scopo di raggiungere un obiettivo.

---

## DIALOGO: DISPOSITIVI DI INTERAZIONE

Il dialogo fra utente e sistema interattivo può essere realizzato attraverso diverse interazioni:

### **Dispositivi di Input:**

- Vista: stampanti, occhiali, caschi
- Udito: sintesi vocali
- Tatto: guanto

### **Dispositivi di Output:**

- Mani: tastiera, mouse, joypad, touch...
- Voce: riconoscimento vocale
- Sguardo: eye tracking
- Postura: sensori

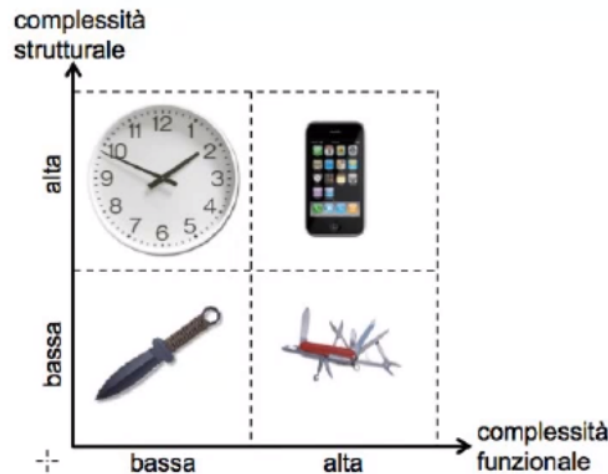
---

## LE DIMENSIONI DELLA COMPLESSITA'

Un sistema può essere considerato complesso per aspetti diversi: perché composto da molti componenti che interagiscono fra loro in modo complicato oppure perché ha molte attività.

Ci sono due misure di complessità:

- Strutturale : quante componenti ha
- Funzionale: quanti task può effettuare



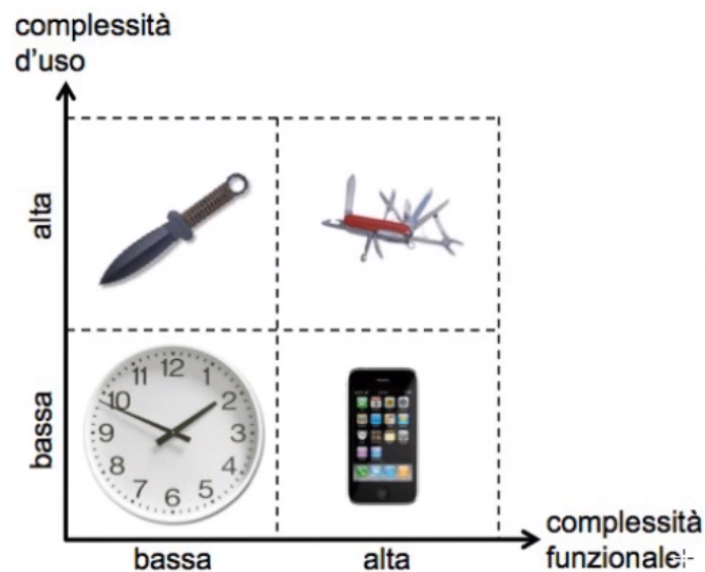
---

## COME MISURARE LA COMPLESSITA'

- La complessità strutturale di un programma può essere misurata contando le linee di codice.
  - La complessità funzionale, si misura attraverso il concetto di **punto funzione**, unità di misura della funzionalità visibili all'utente, indipendente dalla particolare tecnologia usata per realizzarle.
-

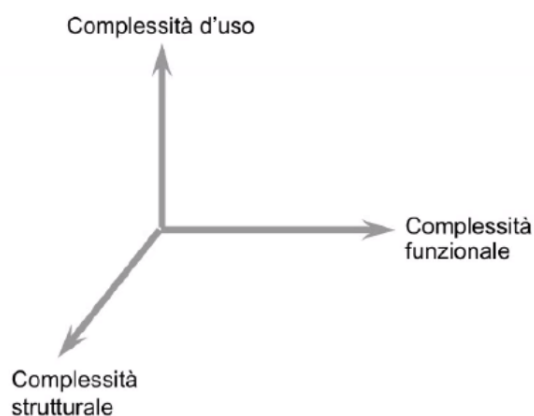
## COMPLESSITA' D'USO

La maggiore o minore facilità con cui siamo in grado di usarlo, se uno strumento è facile da usare la complessità d'uso è bassa.



---

## LE 3 DIMENSIONI DELLA COMPLESSITA'



## LA DIVERSITA' DEGLI UTENTI

Gli utenti sono tutti diversi.

Rispetto alle loro caratteristiche individuali (lingua, cultura, abitudini, preferenze ...).

Rispetto ai rapporti col sistema: ogni utente chiede al sistema cose diverse e si rapporta con esso in modo diverso.

- La diversità degli utenti pone altri problemi di complessità d'uso.

---

## FONDAMENTI DELLA PROGETTAZIONE INTERFACCE

- Comprendere gli utenti e i loro task
  - Progettare avendo in mente l'utente
  - Valutare le interfacce con gli utenti.
-



## LA VELOCITA' DEL CAMBIAMENTO

Accelerazione della complessità:

- questa forte accelerazione del cambiamento è sotto gli occhi di tutti ed è sofferta soprattutto dalle persone più anziane che spesso non sono in grado di adattarsi.
- 

## LE CAUSE DELL'ACCELERAZIONE

- Bisogni dell'utente
  - Ecosistema tecnologico
  - Concorrenza
  - Nuove tecnologie
- 

## COMPLESSITA' D'USO E DIVARIO DIGITALE

Diventa complesso quando iniziano a nascere tecnologie sempre più veloci e si crea un divario digitale tra chi è abile ad usarle e a chi no.

---

## IL RUOLO DELLE INTERFACCE UTENTE

I programmi di e-inclusione promossi dall'UE:

- L'E-Inclusione punta ad assicurare che le persona svantaggiate non siano escluse per mancanza di alfabetizzazione digitale con interfacce intuitive.

La facilità d'uso deve essere considerata non come una caratteristica fra le altre ma è un requisito indispensabile.

Il suo scopo è permettere l'accesso a strumenti che semplificano i compiti quotidiani.

Progettare per tutti significa tener conto delle diversità degli utenti.

Una buona interfaccia: deve filtrare la complessità strutturale interna e ridurre la complessità funzionale, mettendo a disposizione dell'utente funzioni di più alto livello, in grado di effettuare compiti complessi con un automatismo elevato.

---

# EVOLUZIONE DEI PARADIGMI DI INTERAZIONE

5 tecnologie fondamentali che hanno determinato altrettanti paradigmi di interazione fra uomo e computer.

- Terminale scrivente
- Terminale video
- Personal computer
- Browser web
- Mobile

---

Terminale scrivente: il calcolatore attende comandi e restituisce una risposta. È la modalità comunicativa di query e Dos


---

Terminali video: si può interagire con mouse e tastiera una funzionalità del calcolatore. La velocità di una schermata è istantanea facilitando l'interazione con l'utente. La tastiera si arricchisce di shortcuts.

## INTERFACCE FORM FILLING (MODERNE)

- [Home](#)
[Flights](#)
[Hotels](#)
[Cars/Fl](#)
[Vacations](#)
[Cruises](#)
[Last Minute Deals](#)

## Find the best airfare



---

**1. What type of trip is this?**

☒ Round Trip
 ☐ One Way
 ☐ Multiple Destination

☐ Add a hotel and save with TotalTrip™!

---

**2. Where would you like to go?**

From:  To:

☐ Check nearby airports for lower fares

---

**3. When do you prefer to travel?**

☒ Search specific dates
 ☐ Search a date range

Depart: 
 Return:

Oct ▾ 10 ▾ Morning ▾
 Oct ▾ 10 ▾ Evening ▾

---

**4. How many travelers?** (maximum 8)

Booking for children under 12: 
 0 ▾
 Booking for children 13-17: 
 0 ▾
 Booking for adults: 
 0 ▾
 Booking for seniors (65+): 
 0 ▾

---

**5. Other options**

☒ Search all airlines
 ☐ Search specific airlines:

Class of service: 
 Economy ▾

Maximum connections: 
 No Preference ▾

[Search Now](#)

Personal Computer: maggiore potenza di calcolo e possibilità di archiviazione

Nascono i fogli elettronici:

- Griglia di celle con formule o valori
- Le formule possono comprendere valori di altre celle
- L'utente può immettere o manipolare dati

Il mouse introduce possibilità d'interazione completamente nuove rispetto alla tastiera, permettendo di comunicare col calcolatore a gesti:

- Puntare
- Cliccare
- Premere
- Trascinare
- Scorrere con la rotellina

Nasce inoltre la metafora della scrivania:  
l'interfaccia WIMP: (windows, icons, menus, pointers).

**WINDOWS:** aree dello schermo che si comportano come se fossero terminali indipendenti, possono essere spostate, contengono testo o grafica...

**ICONE:** piccole figure o immagini, per rappresentare oggetti dell'interfaccia. Sono stilizzate e di facile comprensione.

**MENU':** scelta offerta sullo schermo di operazioni o servizi che sono offerti da un programma.

**POINTERS:** identificano la posizione del mouse sullo schermo. Ci sono varietà di immagini che si possono usare, di solito una freccetta.

---

## MANIPOLAZIONE DIRETTA SU SCHERMO TATTILE

L'interazione attraverso il mouse o dispositivi quali touchpad è di tipo indiretto.

Con gli schermi touch interagiamo con gli schermi direttamente con le dita, bypassando l'uso del mouse.

---

BROWSER WEB: col mouse possiamo cliccare su link che portano al web. I bottoni si trasformano in testo cliccabile.

Nascono così gli *ipertesti*.

Un ipertesto è un testo costituito di parti chiamate nodi, collegati da link, che associano nodi semanticamente correlati.

Nascono poi gli ipermedia: clicco su testo che aprono degli script o funzioni.

---

Il WWW: la diffusione del paradigma “point and click” avviene però con la diffusione del Web.

Il WWW è un immenso ipertesto, i cui nodi (pagine) non sono contenuti in un archivio locale ma sono distribuiti geograficamente su server connessi alla rete.

L'utente naviga attraverso un browser.

---

Ultimo paradigma: MOBILE

L'iphone può essere considerato il primo device che incarna compiutamente il paradigma mobile.

Il mobile non è un telefono o un computer, è un oggetto del tutto nuovo, pensato per essere

sempre connesso alla rete e destinato a un uso strettamente intimo da parte dell'utente.

Col suo sistema di sensori, il mobile è in grado di raccogliere info e fornire funzioni varie, GPS, NFC, ...

---

## PARADIGMA DEL SOCIAL COMPUTING

Non più interazione fra utenti e sistema ma interazioni fra più utenti mediata da un sistema.

Si pone l'attenzione su sistema con più utenti che interagiscono sulla rete.

---

## PARADIGMA DELL'INTELLIGENZA AMBIENTALE

L'evoluzione della rete procede in diverse direzioni. Nasce l'intelligenza ambientale.

Con questo termine ci si riferisce alla progettazione di ambienti sensibili alla presenza delle persone, che possono interagire con queste in vari modi.

IOT.

Questo paradigma si fonda su diverse tecnologie:

- Embedded: dispositivi sono fra loro connessi



- Context aware: i dispositivi percepiscono info provenienti dall'ambiente
  - Personalizzate
  - Adattive
  - Anticipatorie: predicono i desideri e necessità dell'utente.
- 

## LA ROBOTICA

La nascita dei robot: macchine che possono sostituire l'uomo nell'esecuzione di un compito.

## Lez 3 → Usabilità

### Un modello dell'interazione: CICLO DI FEEDBACK

L'utente per raggiungere il proprio scopo, fornisce un input al sistema e riceve una risposta (feedback) che viene interpretata.

---

#### MODELLO DI DONALD NORMAN

1. Decidiamo quale scopo raggiungere.
  2. Decidiamo cosa vogliamo fare per decire lo scopo prefissato.
  3. Pianifichiamo nel dettaglio le azioni da compiere.
  4. Eseguiamo le azioni pianificate.
  5. Osserviamo come sono cambiati sistema e mondo circostante dopo le nostre azioni.
  6. Elaboriamo ciò che abbiamo fatto.
  7. Valutiamo il risultato.
- 

Esempio:

1. Desidero aprire il getto d'acqua

2. A questo scopo, devo agire sulla manopola dell'acqua
3. Ruoto la maniglia nel verso dell'acqua calda
4. Eseguo l'azione
5. Percepisco che la manopola non gira più e l'acqua è calda
6. Interpreto
7. Valuto e mi accorgo di aver ottenuto il risultato

Se giro una manopola e l'acqua non esce c'è un errore.

Questo modello ci fa comprendere che alcuni sistemi sono più complessi di altri da usare:

- Golfo dell'esecuzione: quantifica il grado con cui il sistema risponde all'azione dell'utente. L'obiettivo è avere un piccolo golfo.
  - Golfo della valutazione: quantifica lo sforzo richiesto a una persona per interpretare il feedback del sistema. Anche qui l'obiettivo è un piccolo golfo
-

## AFFORDANCE E FEEDBACK

- Affordance: la proprietà di un oggetto di influenzare il modo in cui viene usato attraverso la propria forma
  - Ad esempio: le forbici fanno capire come impugnare subito l'utensile.

Una buona affordance riduce il golfo dell'esecuzione. Per ridurre il golfo della valutazione, gli oggetti dovranno fornire un feedback facilmente interpretabile.

Se la distanza fra azione e feedback è significativa, essi possono essere interpretati come eventi tra loro indipendenti. Per ovviare a questi problemi si usano feedback intermedi, come ad esempio un caricamento.

---

## FACILITARE LE AZIONI

Rispecchiare la realtà.

Non bisogna usare gli stessi layout per bottoni e titoli.

## USABILITA'

È l'efficacia, l'efficienza e soddisfazione con cui utenti possono raggiungere determinati obiettivi.

**L'efficacia** è la completezza con cui raggiungo l'obiettivo.

**L'efficienza** sono le risorse spese per ottenere il risultato.

**La soddisfazione** è il grado di accettabilità da parte dell'utente.

Possiamo definire delle metriche per misurare l'usabilità.

---

A seconda di quando si valuta efficacia, efficienza e soddisfazione la valutazione cambia: dipende se l'user è novizio, principiante, competente o esperto...

---

## APPRENDIBILITA' E MEMORABILITA'

L'apprendibilità indica quanto è facile imparare un sistema per utenti novizi.

La memorabilità è la facilità di ricordare il sistema per utenti occasionali.

---

## SUSSIDI DELL'UTENTE

Un sistema interattivo è corredato da una serie di sussidi per aiutare l'utente come l'help desk, le faq, i forum ecc...

Un sistema usabile deve mettere in grado l'utente di usarlo senza usare sussidi.

Una semplice regola: tutte le volte che trovo indicazioni su come usare qualcosa, significa che è progettato male.

---

## ACCESSIBILITA'

Consentire a coloro che hanno disabilità di usare un sistema.

È la capacità dei sistemi informatici di erogare servizi senza discriminazioni anche da parte di

coloro che presentano disabilità e necessitano di configurazioni particolari.

L'accessibilità garantisce la possibilità d'accesso al sistema da parte di tutti.

L'usabilità ne garantisce l'uso efficiente.

## Lez 4 → Conoscere l'utente

### LA VISIONE

Lo sguardo si fissa per un certo tempo su un determinato punto per acquisire l'info visiva.

VISIONE DEL COLORE → la percezione dei colori è affidata ai coni della retina. I coni sono di tre tipi e sono chiamati RGB (red, green, blue).

---

### DALTONISMO: IMPLICAZIONI SUL DESIGN

È necessario che le info rilevanti siano veicolate esclusivamente attraverso il colore. Gli utenti che non possono distinguere i colori usati per comunicare queste info non potrebbero vederle. Non bisogna mai assumere che gli user sappiano distinguere il rosso dal verde.

---

HEAT-MAP: sommando fra loro scanpath (dove l'utente guarda) è possibile costruire le heat-map che mostrano le aree della pagina sulle quali gli sguardi si sono fortemente soffermati.

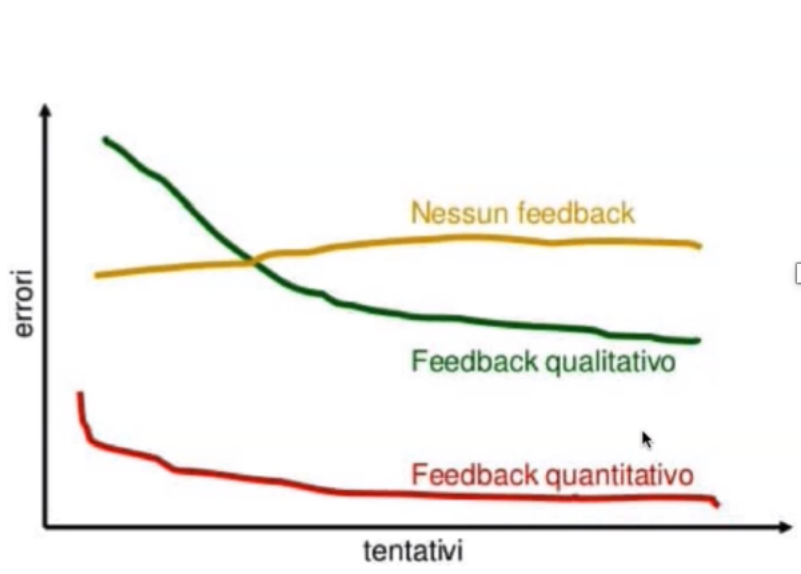


# FEEDBACK:

Il feedback può essere:

- *qualitativo* (es: ingrandire o rimpicciolire una fotografia dello schermo multi-touch dell'iPhone, in corrispondenza del movimento delle nostre dita, l'immagine si modifica in modo continuo. Quando riteniamo che l'immagine abbia raggiunto - *più o meno* - la dimensione desiderata, ci fermiamo.)
- *Quantitativo* (es: impostiamo l'ora della sveglia, ancora sull'iPhone. Selezioniamo l'ora della sveglia strisciando il dito sulla rotella delle ore, che ruota in modo corrispondente al movimento. La rotella ci mostra in ogni istante ora e minuti selezionati. Siamo così in grado di valutare con precisione *quanto* manca al raggiungimento dell'obiettivo, e di graduare il nostro movimento, in maniera più fine.

Quando il feedback è quantitativo, come in questo caso, l'apprendimento è solitamente più preciso, e facciamo meno errori.



## Lez 5

### PROGETTARE VS REALIZZARE

- Progettare: immaginare, ideare e studiare qualcosa per attuarla.
- Realizzare

Quando si progetta si deve avere una visione del futuro: dobbiamo migliorare ciò che esiste oppure andare oltre.

Nella progettazione tradizionale, l'oggetto principale era il sistema. Per progettare un'applicazione centrata sull'essere umano dobbiamo cambiare approccio.

---

### REQUISITI DI PRODOTTO

I requisiti di prodotto sono le priorità richieste al prodotto.

Vengono raccolti per iscritto in un documento strutturato, che fornisce l'input alle attività di progettazione.

---

## I CASI D'USO

È l'insieme d'interazione fra l'utente e il sistema, finalizzato a uno scopo.

Non bisogna confondere i casi d'uso con le funzionalità del sistema. In un caso d'uso il soggetto è l'utente. Una funzione, invece, è una prestazione realizzata dal sistema.

---

## TIPI DI APPROCCIO

- Bottom-up: il progettista è orientato al sistema.
  - Top-down: il progettista è orientato all'utente.
- 

## TIPI DI REQUISITI

- Funzionali: descrivono le funzioni che deve avere il sistema
  - Non funzionali: proprietà che il prodotto deve avere.
-

## PROCESSO DI DEFINIZIONE DEI REQUISITI

La fase di definizione dei requisiti può essere divisa in 3 fasi fondamentali che sono:

- Esplorazione: interviste, analisi, appunti e richieste.
- Organizzazione: stesura dei requisiti
- Revisione: attività di progettazione e sviluppo

## Lez 6

### DAI REQUISITI AL DESIGN CONCEPT

La progettazione non è algoritmica ma è influenzata maggiormente dalla propria creatività.

Cosa determina il passaggio dai requisiti al design concept?

- Inventare: nuove soluzioni ai problemi posti nei requisiti
- Rappresentarle usando notazioni rigorose
- Valutarne la validità e a prove d'uso le soluzioni ipotizzate, per individuare e correggere errori.

Per ideare nuove soluzioni sono state ideate delle tecniche:

1. Mimesi: riprodurre con tecnologie diverse un prodotto già esistente
2. Ibridazione: considerare due o più prodotti esistenti, fonderli e creare un nuovo prodotto

3. Metafora: trasferire nell'ambito del progetto soluzioni adottate in differenti domini applicativi. È la tecnica più usata.
  4. Variazione: progettare un nuovo sistema prendendo le mosse da un modello noto, con varianti
  5. Composizione: collage di soluzioni già note, ma inserite in un contesto nuovo
  6. Creazione pura: il concept del prodotto è completamente nuovo e non prende spunto da prodotti esistenti.
- 

## VARIAZIONE

Versione evolutiva e migliorativa di un prodotto esistente.

L'evoluzione può produrre innovazioni sostanziali.

L'accumularsi di modifiche col tempo può portare a prodotti completamente diversi da quelli di partenza.

---

## Lez 7

### I PROTOTIPI

Si usano diversi strumenti di rappresentazione:

- Formali (Uml).
- Informali (schizzi, diagrammi, storyboard).

L'uso degli storyboard per rappresentare sequenze è meno frequente perché esistono strumenti più potenti come gli *statechart*.

Per rappresentare sulla carta l'interazione con l'utente, servono strumenti più precisi degli storyboard che ci permettono di rappresentare tutte le possibili sequenze d'interazione.

A questo scopo sono state sviluppate nozioni che fanno uso di diagrammi bidimensionali.

Una notazione semplice sono gli *statechart*.

Gli *statechart* sono diagrammi costituiti da due elementi:

- Nodi: rappresenta stato del sistema
- Archi: rappresenta una transizione da uno stato all'altro e viene attivato su uno

scatenarsi di un evento come un click di un bottone.

I vantaggi degli statechart:  
costringe il progettista a esplicitare tutti i percorsi che l'utente segue interagendo con esso. Questo mostra aspetti critici del sistema.

---

## PROTOTIPO

Modello approssimativo del sistema che si vuole realizzare.

Vantaggi:

- Rende le decisioni di progetto esplicite.
- Consente ai progettisti di esplorare design concept alternativi.
- Incorpora feedback degli utenti.
- Migliora la qualità delle specifiche del progetto.
- Riproduce il sistema in forma ridotta.

I prototipi sono validi quando si devono discutere idee con skills diverse.



Servono a:

- Condurre test con utente
- Testare la validità di un'idea
- Chiarire requisiti vaghi.

I prototipi non usano memoria e sono pochi efficienti nei tempi di risposta.

I codici non sono del tutto affidabili e di bassa qualità (sono appunto prototipi).

## COME CLASSIFICARE I PROTOTIPI

In base a:

- Scopo: viene creato per sperimentare il ruolo del prodotto nella vita dell'user, per valutare l'user experience e per sperimentare tecniche e componenti usati nella realizzazione del prodotto.
- Uso:
  - Statico: rappresentazione statica del prodotto.
  - Dinamica: rapp. Dinamica come un video.
  - Interattiva: gli utenti interagiscono.

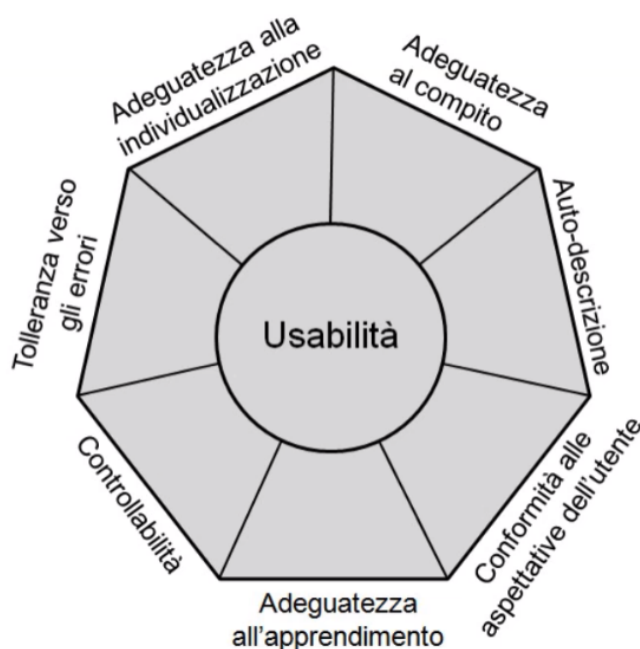
- Fedeltà:
  - Bassa fedeltà: assomiglia alla lontana al prodotto finale (mockup).
  - Alta fedeltà (potrebbe essere un prototipo fisico).
- Completezza funzionale:
  - Prototipo orizzontale: dà idea delle funzionalità del sistema anche se non sono sviluppate
  - “ ” verticale: una funzione è testata esplicitamente e dettagliatamente.
- Ciclo di vita del prodotto:
  - Usa e getta.
  - Evolutivi.
  - Incrementali.

## Lez 9

Fra i numerosi documenti che compongono l'ISO 9241, vogliamo occuparci della parte 110 che descrive 7 principi del dialogo:

1. Adeguatezza al compito
2. Autodescrizione
3. Conformità alle aspettative dell'utente
4. Adeguatezza all'apprendimento
5. Controllabilità
6. Tolleranza verso gli errori
7. Adeguatezza all'individuazione

Questi principi del dialogo, sintetizzando gli elementi chiave che determinano l'usabilità di un sistema interattivo.



La qualità di un'interfaccia è rappresentata con un grafo radar dove si esaminano i principi che avranno un voto da 0-4.

---

## 1. ADEGUATEZZA DEL COMPITO

Questo principio afferma che le funzionalità e le modalità di interazione del sistema devono essere modellate sulle caratteristiche del compito che l'utente deve compiere col supporto del sistema. Il dialogo dovrà essere progettato a partire dai casi d'uso identificati nell'analisi dei requisiti.

Ogni caso d'uso dovrà permettere all'utente di svolgere i compiti e le azioni necessarie nella sequenza più naturale per lui.

### 1.1 DIALOGO ADEGUATO AL COMPITO

i passi del dialogo dovrebbero essere adeguati al compito: dovrebbero essere inclusi tutti i passi necessari ed evitati i non necessari.

### 1.2 INFORMAZIONE ADEGUATA AL COMPITO

Il sistema deve presentare all'utente tutte le info utili per lo svolgimento del compito.

### 1.3 DIALOGO ESSENZIALE

Il sistema dovrebbe evitare di presentare all'utente info ridondanti e la comunicazione dovrebbe essere breve, diretta ed essenziale.

### 1.4 DISPOSITIVI DI INPUT OUTPUT ADEGUATI

La tecnologia mette a disposizione molti dispositivi di I/O per realizzare il dialogo con l'user. Quelli usati dovrebbero essere scelti in funzione del compito specifico.

### 1.5 FORMATI DI I/O ADEGUATI AL COMPITO

I formati dei dati di I/O devono essere adeguati.

Per esempio un programma finanziario dovrebbe accettare valori con più di 2 cifre decimali, quando si parla di Euro.

### 1.6 DEFAULT TIPICI

Una corretta impostazione dei valori di default può semplificare il dialogo. Essi dovrebbero

essere impostati in modo da riflettere le scelte più comuni.

Esempio: in un distributore di biglietti ferroviari, la partenza di default dovrebbe essere quella in cui ci si trova.

## 1.7 COMPATIBILITA' COI DOCUMENTI

Questa indicazione è importante nei sistemi informativi aziendali.

---

## AUTO DESCRIZIONE

Questo principio richiede che il sistema comunichi all'utente che cosa egli possa fare e come, e cosa sta accadendo.

Lo scopo è ridurre il golfo dell'esecuzione e della valutazione.

## 2.1 GUIDA PER L'UTENTE

Ogni passo del dialogo dovrebbe fornire all'utente informazioni utili per proseguire. Le informazioni sono di 3 tipi:

- Istruzioni operative
- Informazioni sullo stato del sistema
- Feedback

## 2.2 INTERAZIONE EVIDENTE

Questa linea è legata al concetto di AFFORDANCE. Raccomanda che i dialoghi siano progettati in modo che l'interazione col sistema risulti evidente.

In molti casi questo si può ottenere adottando modalità operative già note all'utente in altri contesti: ad esempio i player di musica digitale usano un'interfaccia simile a quella reale.

## 2.3 DESCRIZIONE DELL'INPUT ATTESO

Ogni volta che il sistema chiede un input all'user, dovrebbe fornirgli info su ciò che aspetta e sul suo formato.

- all'utente, dovrebbe fornirgli informazioni adeguate su ciò che si aspetta, e sul suo formato. Quindi, sono da evitare dialoghi che contengono campi liberi, come il seguente:

DATA DI NASCITA: \_\_\_\_\_

- che andrebbero sostituiti con input guidati, per esempio:

DATA DI NASCITA: \_\_/\_\_/\_\_\_\_ (gg/mm/aaaa)

---

## 2.4 STATO VISIBILE

L'user dovrebbe essere sempre informato sullo stato del sistema. Se l'utente non conosce lo

stato, non può prevedere come risponderà alle sue azioni. Un sistema “muto” genera incertezza anche quando tutto è ok. La regola del silenzio assenso non è da utilizzare.

## 2.5 FORMATI DESCRITTI

Il sistema deve fornire ogni info su formati e unità di misura usati.

## 2.6 MANUALISTICA MINIMA

Implementare sistemi di help interni senza uscire dall'applicazione.

---

## CONFORMITA' ALLE ASPETTATIVE

Questo principio afferma che il dialogo deve essere conforme a quello che l'utente si aspetta.

## 3.1 LINGUAGGIO FAMILIARE

Il sistema deve usare un linguaggio che l'utente ben conosce. Un suo uso inappropriato può compromettere l'usabilità.



### 3.2 ADERENZA ALLE CONVENZIONI

Il dialogo deve seguire le convenzioni comunemente adottato nel contesto

### 3.3 ORGANIZZAZIONE ABITUALE

La struttura del dialogo e l'organizzazione dei dati dovrebbero permettere all'user di effettuare operazioni secondo le modalità a lui consuete.

### 3.4 DIALOGO CONSISTENTE

I dialoghi realizzati da uno stesso sistema dovrebbero avere aspetto e comportamento consistenti.

### 3.5 FEEDBACK CONFORME ALLE ASPETTATIVE

Il feedback deve essere:

1. Ben comprensibile e specifico
2. TEMPESTIVO: tempi di risposta immediati

### 3.6 TEMPI DI RISPOSTA CONFORMI ALLE ASPETTATIVE

L'user si forma delle aspettative sul tempo di esecuzione delle elaborazioni richieste. Se il

sistema ritarda la risposta o è muto può sorgere il dubbio di presenza di errori.

### 3.7 MESSAGGI ADEGUATI AL CONTESTO

La lunghezza e tipo dei messaggi devono essere adeguati.

Non tutti i messaggi sono adeguati alla situazioni, anche se il contenuto è pertinente.

### 3.8 MESSAGGI IN POSIZIONE APPROPRIATA

I messaggi di feedback fornite all'user dovrebbero apparire dove si trova il focus dell'attenzione dell'user.

### 3.9 INPUT IN POSIZIONE ATTESA

I sistema deve richiedere l'input dell'user nella posizione in cui questi si aspetta di doverlo fornire.

### 3.10 STILE COERENTE DEI MESSAGGI

Lo stile dei messaggi deve essere coerenti, formulati in modo oggettivo ed evitando connotazioni negative o enfatiche.

---

## 4 ADEGUATEZZA ALL' APPRENDIMENTO

Si riferisce alla *learnability*.

Auspica che il dialogo sia organizzato in modo tale da aiutare e guidare l' user nell'apprendimento del sistema.

### 4.1 BASSA SOGLIA DI APPRENDIMENTO

Ogni sistema dovrebbe essere usabile con un livello di apprendimento minimo. L'utente inesperto dovrebbe poter usare le funzioni di base senza addestramento.

### 4.2 AIUTO ALLA FAMILIARIZZAZIONE

Il sistema dovrebbe aiutare l' user a prendere familiarità col dialogo. Un esempio sono i compiti iniziali al primo uso del sistema.

### 4.3 AIUTO ONLINE

È necessario dare all' user la possibilità di aiuto quando lo desidera. Questo è lo scopo di sistemi di help online.

### 4.4 FEEDBACK INTERMEDI

Il dialogo dovrebbe fornire feedback intermedi e finali di un compito. Questa tecnica è usata per le transazioni che avvengono sul Web.

### 4.5 MODELLO CONCETTUALE EVIDENTE

I sistemi deve aiutare l' user a costruirsi un modello concettuale del sistema. Lo scopo è permettere all' user di orientarsi con facilità.

### 4.6 SPERIMENTAZIONE SICURA

Il sistema deve permettere all' user di provare le funzioni, senza che ciò produca conseguenze negative.

### 4.7 RIAPPRENDIMENTO FACILITATO

In tutti i sistemi ci sono funzioni raramente usate.

Può accadere che l'utente dimentichi il funzionamento. Il sistema dovrebbe fornire un aiuto per riapprendere, trattando le funzioni di uso non frequente in modo particolare.