

# Corso di Interazione Uomo Macchina a.a. 2012-13

---

Fabio Pittarello

Ca' Foscari  
Università di Venezia  
Dipartimento di Informatica  
Via Torino 155,  
Mestre (Venezia), Italia  
e-mail [pitt@unive.it](mailto:pitt@unive.it)

---

Usabilità

*Nota: il materiale contenuto in questo documento è disponibile solo per uso interno nell'ambito del corso di Interazione Uomo Macchina.*

---

---

## Definire l'usabilità – IEEE & ISO

---

- La facilità con la quale un utente può imparare ad operare, a predisporre l'input e a interpretare l'output di un sistema o di una componente (IEEE, 1990)
  - L'efficacia, l'efficienza e la soddisfazione con la quale determinati utenti raggiungono scopi specifici in un determinato ambiente d'uso (ISO 9241-11, 1998)
    - Efficacia: l'accuratezza e la completezza con la quale determinati utenti raggiungono scopi specifici in determinati ambienti
    - Efficienza: rapporto tra le risorse impiegate e l'accuratezza e la completezza degli scopi raggiunti
    - Soddisfazione: il comfort e l'accettabilità del sistema rispetto agli utenti e ad altri soggetti condizionati dal sistema stesso
- 
-

## Definire l'usabilità

---



La definizione di usabilità non è necessariamente legata a caratteristiche specifiche dei prodotti software: si tratta di una definizione del tutto generale, e come tale applicabile a qualsiasi artefatto dell'uomo.

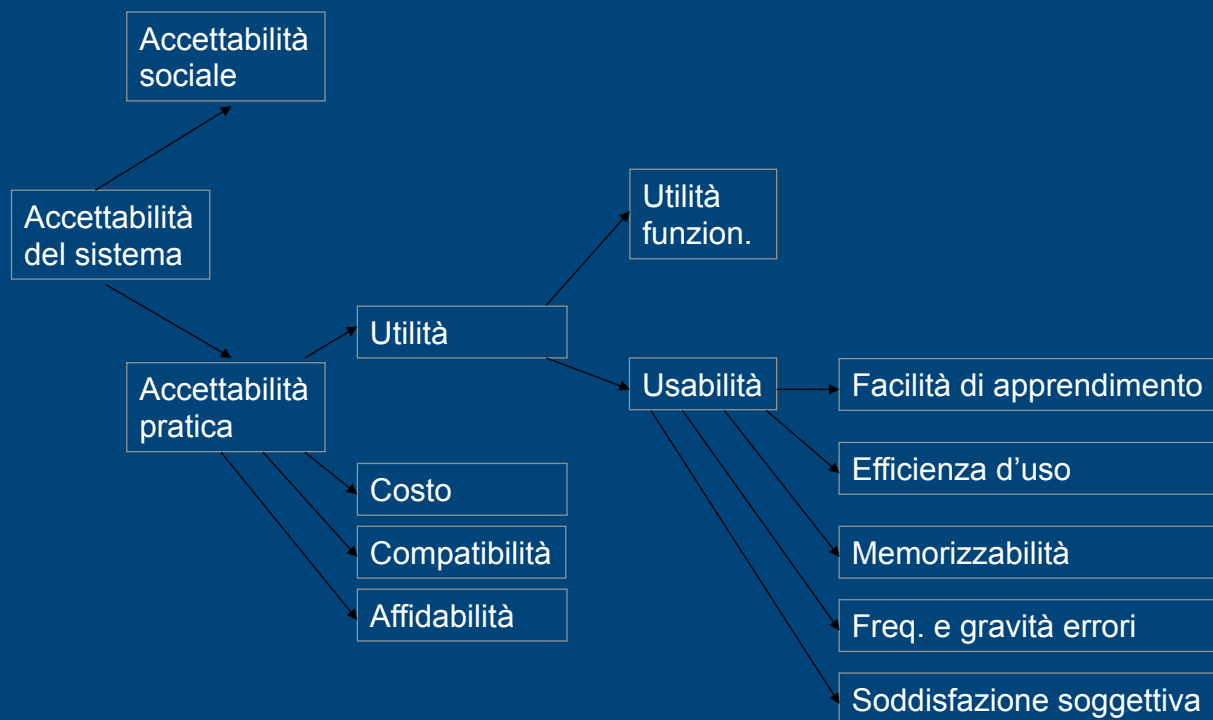
- Esempio: manopola di un rubinetto dell'acqua (rif. Polillo)
    - Per misurarne la usabilità, e quindi valutarne la facilità d'uso, potremmo definire le seguenti metriche:
      - efficacia: numero di volte in cui si riesce a chiudere completamente l'acqua rispetto al numero totale dei tentativi
      - efficienza: numero di giri di manopola necessari, in media, per chiudere completamente l'acqua
      - soddisfazione: numero di volte in cui gli utenti hanno espresso la preferenza per questa manopola, in alternativa a un insieme predefinito di manopole di foggia differente.
- 
- 

## Definire l'usabilità – standard de facto

---

- L' Usabilità è la misura della qualità dell' esperienza dell' utente che interagisce con qualcosa – un sito web, un' applicazione software tradizionale o qualsiasi altro artefatto con il quale l' utente può operare con specifiche modalità.  
[J. Nielsen]
- 
-

## Definire l'usabilità – Nielsen



## Definire l'usabilità – Nielsen

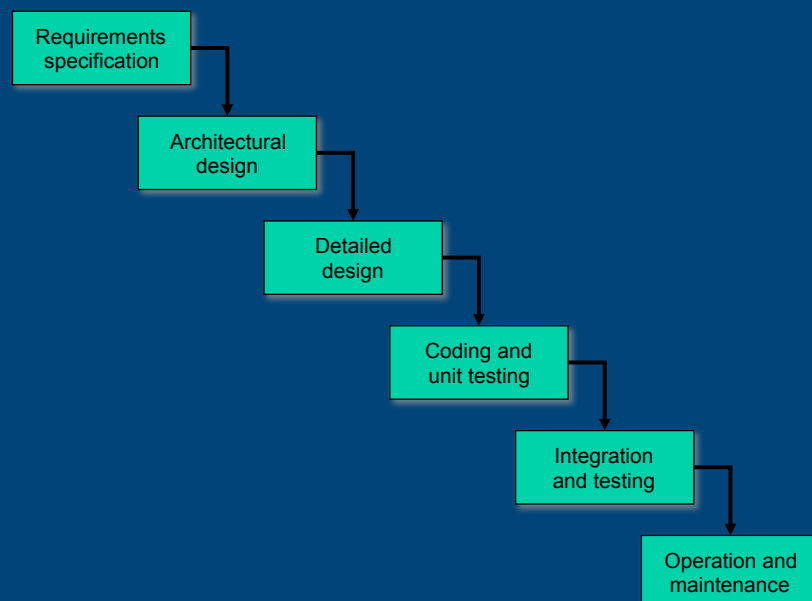
- Facilità di apprendimento
  - Con quale velocità l'utente che non ha mai visto prima l'interfaccia utente può imparare a utilizzarla per svolgere i compiti (task) basilari?
- Efficienza d'uso
  - Una volta che un utente ha acquisito una buona esperienza con il sistema, quanto velocemente può svolgere i propri compiti?
- Memorizzabilità
  - Se un utente ha già utilizzato il sistema, può ricordare in maniera sufficiente le modalità d'uso per utilizzarlo con efficacia anche in un tempo successivo oppure necessita di una nuova fase di training?
- Frequenza e gravità degli errori
  - Con che frequenza gli utenti fanno errori utilizzando il sistema? Quanto gravi sono questi errori? Come riescono gli utenti a recuperare gli errori commessi?
- Soddisfazione soggettiva
  - Quanto *piace* all'utente utilizzare il sistema?

## Usabilità e ciclo di vita del software

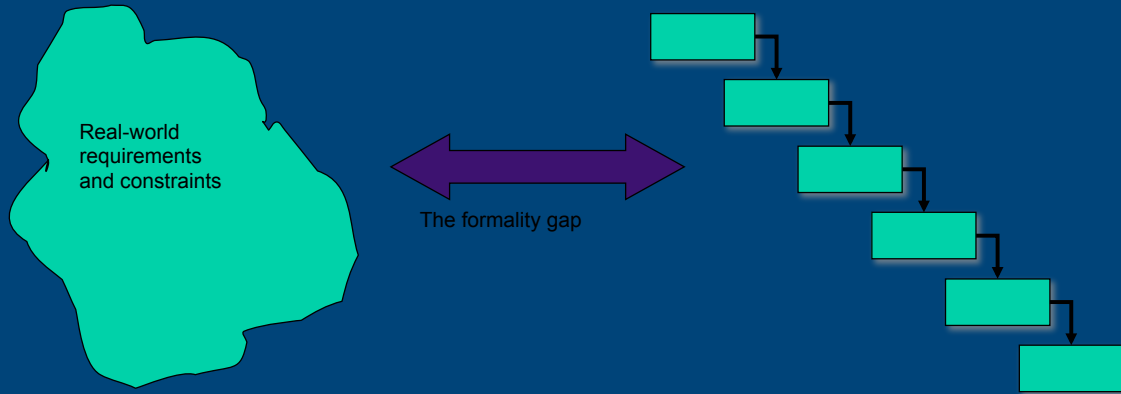
---

- Il ciclo di vita di un software proposto dalla disciplina del Software Engineering (**software engineering life cycle**) rappresenta una mappatura ampiamente accettata del processo di sviluppo del software.
  - L'usabilità non è una caratteristica da aggiungere alla fine del ciclo di sviluppo di un software.
  - Il ciclo di vita proposto dalla disciplina dell'Ingegneria dell'Usabilità (**usability engineering life cycle**) si sviluppa in parallelo con le fasi tradizionali di sviluppo di un software ed è strettamente interconnesso con esse.
- 
- 

## Il modello a cascata

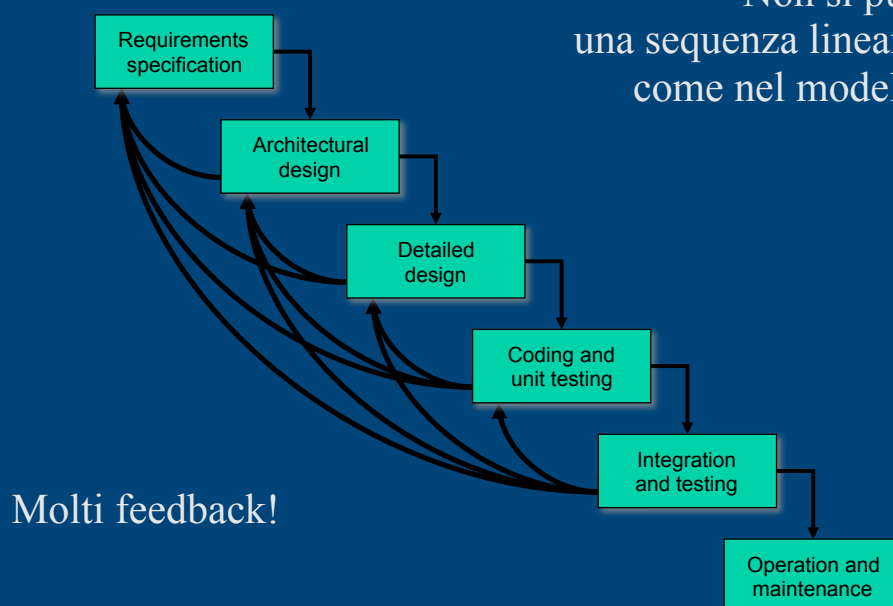


## Verifica e validazione



## Ciclo di vita per sistemi interattivi

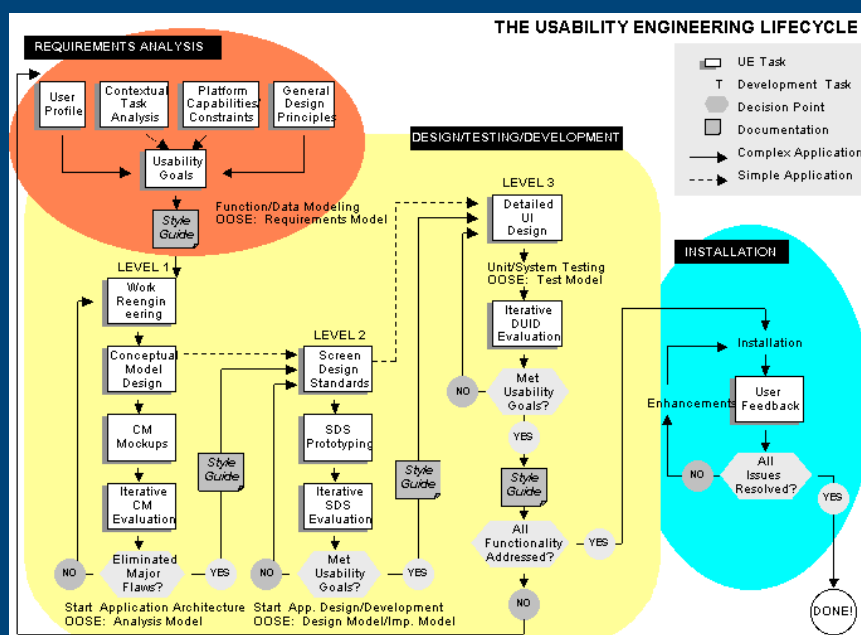
Non si può assumere una sequenza lineare di attività come nel modello waterfall



## Ingegneria dell'usabilità – definizione

- L'ingegneria dell'usabilità è un approccio metodico per raggiungere l'usabilità durante lo sviluppo di un software.
- L'ingegneria dell'usabilità coinvolge molti metodi da applicare nelle diverse fasi dello sviluppo del software.
- Questi metodi coinvolgono la fase di definizione dei requisiti, lo sviluppo e il test di prototipi, la valutazione di design alternativi, la proposta di soluzioni e il test del sito (o dell'interfaccia) con gli utenti.
- La maniera meno costosa per svolgere attività rivolte ad ottenere una buona usabilità di un software è quella di svolgerle quanto prima rispetto al ciclo di vita del software!

## Usability Engineering LifeCycle – D.J. Mayhew



<http://drdeb.vineyard.net/index.html>

## ***Usability Engineering LifeCycle - Nielsen***

---

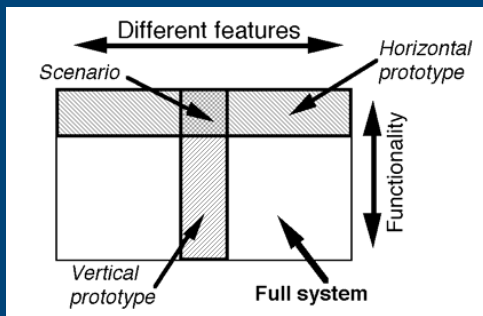
- Analisi degli utenti
    - caratteristiche individuali, attività correnti e desiderate
    - analisi funzionale, analisi evolutiva
  - Analisi competitiva
  - Definizione degli obiettivi di usabilità
  - Design parallelo
  - Design partecipatorio
  - Coordinamento dell' interfaccia complessiva
  - Linee guida e valutazione euristica
  - Prototipazione
  - Valutazione dell' interfaccia
  - Design iterativo
  - Studio e valutazione del sistema installato
- 
- 

## ***Usability Engineering LifeCycle – tipologie di attività***

---

- Il processo di Usability Engineering è composto da diverse attività. Possiamo distinguere queste attività in due macro-aree
    - Attività di pianificazione e di progetto
    - Attività di valutazione
- 
-

## Attività di progetto - prototipazione



- Tra le attività di progettazione un posto rilevante è occupato dalla progettazione di prototipi
- Il prototipo riduce la complessità del sistema eliminandone alcune parti
  - I prototipi orizzontali riducono il livello di funzionalità del sistema e danno luogo a interfacce con features non completamente implementate
  - I prototipi verticali riducono il numero di features; le features selezionate per il prototipo vengono pienamente implementate
  - Gli scenari costituiscono la riduzione estrema del sistema: essi simulano l'interfaccia solo se l'utente segue un percorso precedentemente pianificato.

## Attività di progetto - prototipazione





## Attività di progetto - prototipazione



The search wall. Un esempio di applicazione della tecnica del Mago di OZ. il sistema di interrogazione di database di libri non è effettivamente implementato; tuttavia gli utenti ricevono risposte appropriate alle loro interrogazioni formulando e posizionando oggetti fisici sul pannello grazie ad alcuni tecnici che, non visti, presentano sul pannello stesso i libri che soddisfano i parametri dell'interrogazione.

## Attività di progetto - prototipazione

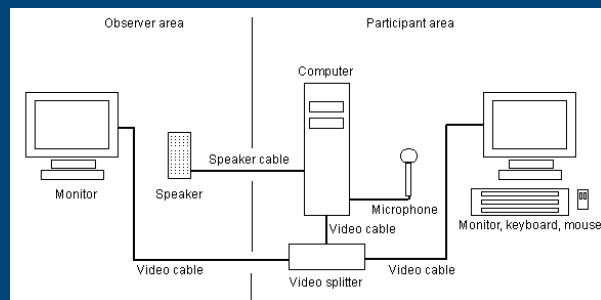
- Oltre a ridurre la porzione di sistema implementata, i prototipi possono essere costruiti più velocemente
  - Attribuendo meno enfasi all'efficienza dell'implementazione
  - Accettando codice meno affidabile o di qualità peggiore
  - Utilizzando algoritmi semplificati che non sono in grado di padroneggiare tutti i casi speciali (es. calendario senza anni bisestili)
  - Utilizzando un umano dietro le scene che si prenda carico delle operazioni troppo difficili da programmare (es. Mago di Oz)
  - Utilizzare una configurazione di computer più potente di quella che verrà utilizzata come target per il rilascio dell'applicazione
  - Usando media a bassa fedeltà (es. fotogrammi fissi anziché stream video)
  - Utilizzando dati falsi (es. stream video non corrispondente a etichetta testuale di selezione)
  - Utilizzando schemi su carta (paper mockup) anziché programmi funzionanti, con un esperto che fa la parte del computer e mostra lo schema successivo quando il testet seleziona un'azione
  - Dando una descrizione verbale dell'interfaccia

## Attività di valutazione dell'usabilità

<http://www.cs.umd.edu/~zzj/UsabilityHome.html>

### • Test

- Utenti rappresentativi operano sul sistema (o sul prototipo) svolgendo i loro compiti. I valutatori registrano quello che l'utente fa o dice mentre svolge il compito. Tipicamente i test vengono condotti con uno o due utenti alla volta. I valutatori utilizzano i risultati del test per verificare come l'interfaccia supporta gli utenti nello svolgimento dei loro compiti.



## Attività di valutazione dell'usabilità

<http://www.cs.umd.edu/~zzj/UsabilityHome.html>

### • Tecniche di ispezione

- Specialisti di usabilità (e a volte sviluppatori del software e utenti) esaminano gli aspetti di usabilità relativi ad una determinata interfaccia utente.

### • Raccolta dati

- I valutatori dell'usabilità ottengono informazioni intervistando gli utenti o chiedendo loro di rispondere in forma scritta ad un insieme di domande.

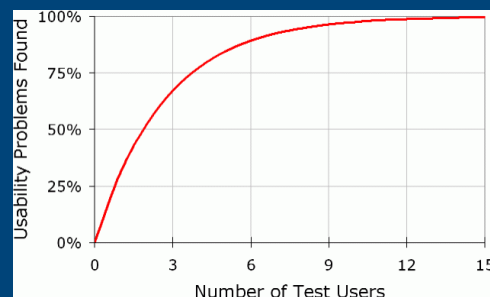


Diagramma tratto da  
**Why You Only Need to Test  
With 5 Users**  
[http://www.useit.com/alertbox/  
20000319.html](http://www.useit.com/alertbox/20000319.html)

## Euristiche di Usabilità

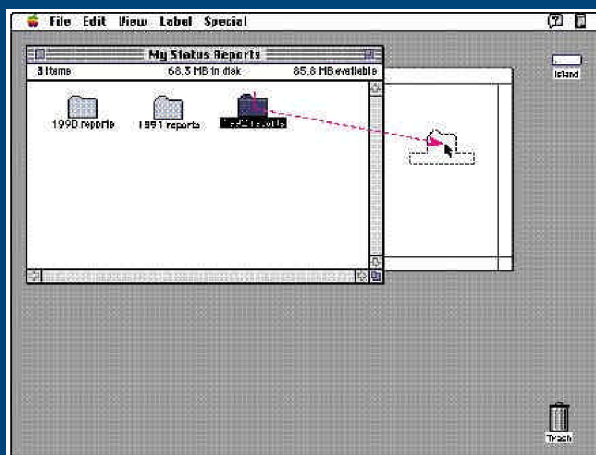
---

- Un insieme di principi di usabilità sufficientemente generali da poter essere applicati ad una grande varietà di interfacce, includendo sia le interfacce basate su caratteri che le interfacce grafiche.
  - Primo elenco di 10 Euristiche [Molich e Nielsen, 1990]
  - Revisione 10 Euristiche [Nielsen 1994]
- E' possibile utilizzare le euristiche sia come supporto all'attività di progettazione, sia come guida per la valutazione di usabilità di un'interfaccia.
- Nel secondo caso si parla di Valutazione Euristica:
  - Tecnica a basso costo, facile e veloce
  - Richiede lo studio dell'interfaccia da parte di un piccolo gruppo di valutatori che verificano l'eventuale violazione delle euristiche stesse

## 10 Euristiche – Nielsen

---

- 1 Visibilità dello stato del sistema
  - Il sistema dovrebbe sempre tenere gli utenti informati riguardo a quello che sta accadendo, con un feedback appropriato e in tempo ragionevole



## 10 Euristiche – Nielsen

---

- 2 - Corrispondenza tra il sistema e il mondo reale
    - Il sistema dovrebbe parlare il linguaggio dell'utente, con parole, frasi e concetti familiari all'utente piuttosto che con termini orientati al sistema. Il sistema dovrebbe seguire le convenzioni del mondo reale, facendo apparire l'informazione secondo un ordine logico e naturale
  - 3 - Controllo dell'utente e libertà
    - Gli utenti selezionano spesso funzionalità del sistema per errore; essi necessitano perciò di uscite di emergenza chiaramente identificate per lasciare lo stato in cui si trovano senza dover passare un'interfaccia sofisticata. Vanno supportate le funzioni di Undo e Redo.
- 
- 

## 10 Euristiche – Nielsen

---

- 4 - Consistenza e standard
    - Gli utenti non dovrebbero preoccuparsi di dover capire se parole, situazioni e azioni diverse significano la stessa cosa. Seguire le convenzioni delle piattaforme su cui si sta lavorando.
  - 5 - Prevenzione degli errori
    - Un design attento che previene un problema è meglio di un buon messaggio di errore!
- 
-



## ***10 Euristiche - Nielsen***

---

- 10 - Aiuto e documentazione
    - Anche se la situazione ideale è quella in cui il sistema può essere utilizzato senza documentazione, può essere necessario fornire aiuto e documentazione. Tale informazione dovrebbe essere facile da cercare, focalizzata sui compiti dell'utente, elencare i passi concreti da fare e non essere troppo estesa.
- 
- 

## ***Regole di design***

Design per ottenere la massima usabilità  
– lo scopo dell'interaction design

- Principi di usabilità
    - Principi generali
  - Standard e linee guida
    - Indicazioni per il design
  - Design patterns
    - Catturare e riutilizzare la conoscenza acquisita nel design
- 
-

# *Principi a supporto dell'usabilità*

## Apprendibilità

La facilità con la quale nuovi utenti possono iniziare un'interazione efficace ed ottenere la massima performance

## Flessibilità

La molteplicità di modi con i quali l'utente ed il sistema scambiano informazioni

## Robustezza

Il livello di supporto che viene fornito all'utente per determinare la positiva conclusione di un'attività diretta ad uno scopo

---

# *Standard*

- Stabiliti da organismi nazionali o internazionali per assicurare conformità da parte di una larga comunità di designers
  - Gli standard sottendono una buona teoria e una tecnologia che cambia lentamente
  - Gli standard hardware sono più comuni degli standard software
  - **Alta autorità e basso livello di dettaglio**
  - ISO 9241 definisce l'usabilità come l'efficacia, l'efficienza e la soddisfazione con la quale gli utenti portano a termini determinati compiti.
-

## *Linee guida*

- Più generali e suggestive
  - Molti libri di testo e report propongono insiemi di linee guida
  - **Linee guida astratte (principi)** applicabili durante le prime attività del ciclo di vita del software
  - **Linee guida dettagliate (style guides)** applicabili durante le attività successive
  - La comprensione delle motivazioni sottesi alle linee guida aiutano nella risoluzione dei conflitti
- 
- 

## *Golden rules ed euristiche*

- Regole di design broad brush
  - Utili check list per un buon design
  - Si ottiene un buon design utilizzando queste regole piuttosto che non utilizzando nessuna regola!
  - Diverse raccolte disponibili, es.
    - 10 euristiche di Nielsen
    - 8 regole d'oro di Shneiderman
    - 7 principi di Norman
- 
-



## **8 Regole d'oro di Shneiderman**

- 1. Combattere per la consistenza*
  - 2. Permettere agli utenti abituali l'utilizzo di scorciatoie*
  - 3. Dare feedback informativo*
  - 4. Progettare dialoghi provvisti di chiusura*
  - 5. Supportare la prevenzione degli errori e una facile gestione degli stessi*
  - 6. Consentire la reversibilità delle azioni*
  - 7. Supportare il controllo interno*
  - 8. Ridurre il carico della memoria a breve termine*
- 
- 

## **7 Principi di Norman**

- 1. Utilizza sia la conoscenza disponibile nel mondo che quella disponibile nella testa.*
  - 2. Semplifica la struttura dei compiti (task).*
  - 3. Rendi le cose visibili: oltrepassa i golfi dell'esecuzione e della valutazione.*
  - 4. Definisci le giuste relazioni (mappings).*
  - 5. Sfrutta il potere dei vincoli, sia naturali che artificiali.*
  - 6. Progetta considerando la possibilità di errore.*
  - 7. Quando tutto il resto fallisce, standardizza.*
- 
-

# Design patterns

- Un approccio nel quale viene riutilizzata la conoscenza relativa a soluzioni di design di successo
- Concetto nato nell'ambito dell'architettura: Alexander
- Un pattern è una soluzione invariante di un problema ricorrente all'interno di un contesto specifico.
- Esempi
  - Illumina ogni stanza da due lati (architettura)
  - Torna indietro verso un posto sicuro (HCI)
- Pattern non esistono come entità isolate, ma sono collegati ad altri pattern in linguaggi in linguaggi che permettono di generare soluzioni di design complete

