

Appunti UUX

Introduzione

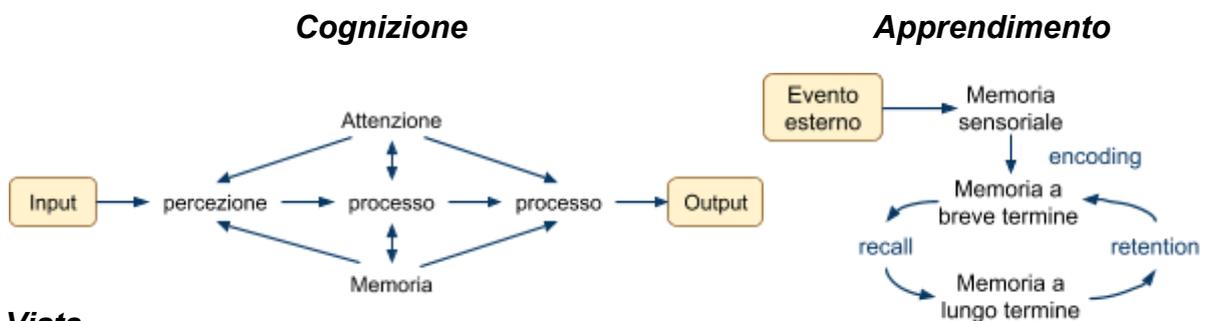
Human performances: il lavoratore è come la macchina e si devono massimizzare le sue performance.

Ergonomics: cercare di creare macchine che sfruttano le caratteristiche fisiche dell'uomo al meglio. Vi sono studi riguardo all'usabilità dei computational devices, man-machine interaction diventa human-computer interaction.

User experience: si pone enfasi sulla soddisfazione dell'utente rispetto alla semplice usabilità. Lo standard ISO 9241-210 definisce l'user experience come le percezioni e le risposte di una persona, che risultano dall'utilizzo di un prodotto, sistema o servizio.

L'usabilità: è la facilità di utilizzo e learnability di un oggetto human-made come uno strumento o un dispositivo. Utility si divide in usability e usefulness, il primo intende la facilità di utilizzo, di apprendimento, i pochi errori possibili; il secondo indica se l'utilizzo ha un fine che serve a migliorare qualche attività.

The human component



Vista

Si hanno 2 stage di percezione:

- La percezione fisica dello stimolo (occhi)
- Un ulteriore processazione dei dati (cervello)

Negli occhi sono presenti i **fotorecettori**, costituiti da *bastoncelli* (diffusi sulla retina, sensibili alla luce ma non ai colori), coni (sensibili ai colori, ma non attivati con poca luce), X-ganglion (identificazione pattern visivi), W+Y-ganglion (identificazione del movimento senza percezione forme).

La percezione della *distanza* invece fa sì che oggetti vicini appaiano più grandi.

La percezione della *profondità* dipende da una diversa percezione tra i due occhi.

La percezione della *brillantezza* dipende dalla quantità di luce emessa da un corpo.

La percezione dei *colori* è data dai coni, si è sensibili a RGB (pochi recettori blu).

L'*elaborazione di immagini* genera concetti interpretati dal cervello, è basata su pattern ed aspettative (le illusioni ottiche sono spesso attivazioni errate di pattern).

Principi di Gestalt

“L'insieme è più della somma delle sue parti”.

- Prossimità (organizzazione in righe/colonne)
- Similarità (gruppi di elementi simili)
- Continuazione (porzioni mancanti non cambiano la visione d'insieme)
- Semplicità (tendenza a preferire la soluzione semplice)
- Background (riconoscere oggetti davanti e sfondo)
- Closure (diverse interpretazioni per varie parti di un'immagine).

Elaborazione del testo

Si hanno 3 fasi:

1. Percezione visuale del testo
2. Decodifica delle parole in base al linguaggio di riferimento
3. Analisi sintattica e semantica del testo

Udito

Vengono fornite informazioni su ciò che ci circonda, distinguendole per intonazione, rumorosità e timbro, tramite la percezione stereofonica delle orecchie possiamo percepire la direzione del suono.

Memoria

Tramite il modello a livelli la memoria può essere divisa in:

- Memoria *sensoriale*: vengono catturate le informazioni sensoriali sull'ambiente per un tempo da ½ secondo fino a 4, per ridurre il carico di stimoli.
- Memoria a *breve termine*: vengono conservate e processate le informazioni rilevanti per l'attività in esecuzione al momento, per un periodo di circa 15 secondi.
- Memoria a *lungo termine*: non è mai una registrazione completa di ciò che si è percepito, ma il risultato di un post processo delle cose rilevanti, filtrate. Può essere divisa in memoria episodica (eventi ed esperienze) e memoria semantica (fatti, concetti ed abilità).
- Memoria *eidetica*: è l'abilità di ricordare un gran numero di dettagli percepiti.

L'organizzazione è gerarchico-associativa (reti semantiche), quindi i concetti disconnessi sono più duri da memorizzare di quelli connessi, lo stesso vale per concetti astratti e concreti.

Dimenticare

Vi sono due teorie:

- *Decadimento*: le informazioni non vengono accedute frequentemente.
 - *Sostituzione*: un nuovo ricordo prende il posto di uno precedente e simile.
- Il design per la memoria a lungo termine prevede la creazione di una narrativa per la sequenza di passaggi da compiere.

Attenzione

Ovvero la selezione di uno o più stimoli sensoriali tra quelli che ci raggiungono. Eventi inaspettati sono fonte di *distrazione*.
Il design per l'attenzione prevede la riduzione del carico cognitivo e mostrare chiaramente lo stato del sistema e il livello di progresso.

Ragionamento

Ovvero come usiamo conoscenze che già abbiamo per generare conclusioni su eventi su cui ci stiamo concentrando. Si hanno 3 tipi di ragionamento:

1. **Deduttivo**: partendo da un'assunzione generale e da un caso concreto, otteniamo una conclusione. [Le persone vive respirano, Matteo respira, allora Matteo è vivo.]
2. **Induttivo**: partendo da casi omogenei produciamo (induciamo) una regola generale (generalizzazione). [Ho visto solo elefanti con la proboscide, quindi tutti gli elefanti del mondo hanno la proboscide.]
3. **Abduttivo**: dato un caso a cui possono essere applicate molte regole, si sceglie la migliore per il caso. [Matteo guida veloce, Matteo guida veloce quando è arrabbiato o quando scappa dalla polizia, non c'è la polizia allora Matteo è arrabbiato.]

Il *problem solving* prevede il trovare una soluzione ad un problema nuovo o non familiare.

La legge di Fitt dice che il tempo necessario per raggiungere un obiettivo su uno schermo con un puntatore è proporzionale alla distanza ad inversamente proporzionale alla dimensione del target.

Psicologia cognitiva

Il corretto design di oggetti deve aiutare la scoperta e la comprensione, prevedendo che le persone facciano errori e pianificare le soluzioni.

Affordance (invito all'uso)

Qualità fisica di un oggetto che suggerisce ad un essere umano le azioni appropriate per manipolarlo. In caso l'affordance non sia abbastanza chiaro, è necessario segnalare l'utilizzo tramite *signifiers* (costrutti che rendono chiara l'affordance).

Modello concettuale

Il designer ha il dovere di fornire il miglior modello concettuale di un oggetto appena lo si osserva.

Mapping

Ovvero la relazione tra i comandi e la loro funzione, alcuni oggetti creano un mapping naturale sfruttando analogie anatomiche, culturali, ...

Feedback

Ovvero ritornare informazioni permettendo all'utente di capire quali azioni sono state eseguite ed il risultato (come la macchina da informazioni sul suo stato).

Conoscienza

Può essere dichiarativa, come le leggi scritte, esplicita e conscia (facile da spiegare, difficile da imparare ed usare). Oppure procedurale, come giocare a tennis, implicita ed automatica (difficile da spiegare, facile da applicare).

7 fasi di un'azione secondo Norman

1. Creare il goal
2. Creare l'intenzione
3. Specificare l'azione
4. Eseguire l'azione
5. Percepire lo stato del mondo
6. Interpretare lo stato del mondo
7. Comparare il risultato con il goal

Il sistema affettivo ha tre livelli:

- Livello *viscerale*: inconscio, protegge dai pericoli del mondo esterno.
- Livello *comportamentale*: competenze imparate.
- Livello *riflessivo*: riflessioni conscie (pensiero, contare, ...).

Golfi di esecuzione e valutazione (Norman)

I *golfi* indicano le distanze tra il modello mentale dell'utente ed il mondo reale degli oggetti su cui l'utente agisce, questi possono generare errori.



Mistakes: l'intenzione dell'azione è proprio sbagliata e causata da incomprensione.
Slips: intenzione è corretta ma l'esecuzione sbagliata e causata da mancanza di abilità e/o attenzione.

Computer

Input e output

L'*interazione* è il processo di trasferimento delle informazioni da un lato all'altro. I dispositivi di I/O devono adattarsi alle capacità di Input ed Output degli umani.

Interaction component

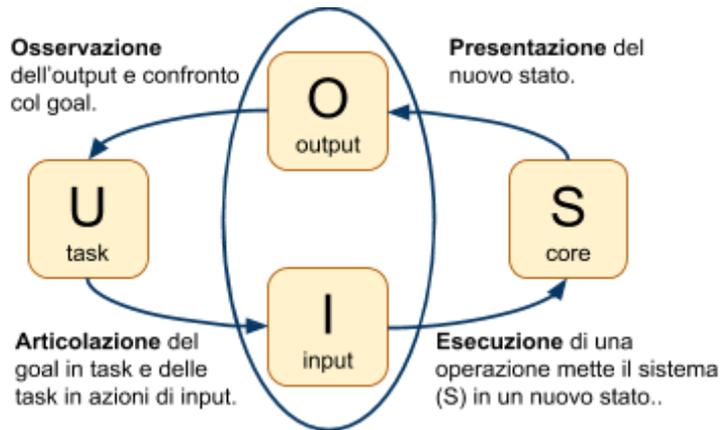
Modelli di interazione

L'*interfaccia* è quel luogo in cui l'interazione tra due sistemi complessi e non omogenei ha luogo; l'interfaccia crea la *traduzione del dialogo tra i due sistemi*, sono però possibili degli errori.

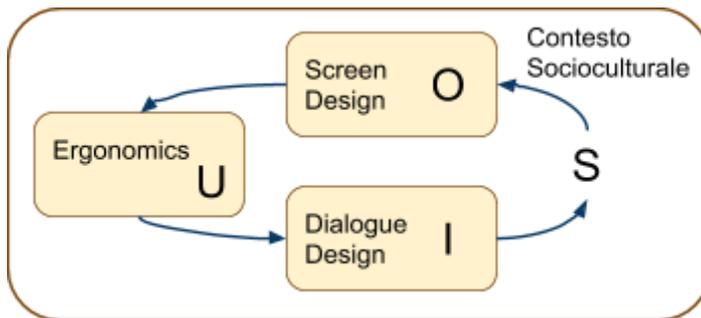
L'uso di modelli permette di evidenziare i problemi di traduzione e di confrontare soluzioni.

- Modello di **Norman**:
Visto in precedenza.
- Modello di **Abowd & Beale**:
Un *sistema interattivo* permette ad un utente di *ottenere un goal* in un dominio di applicazione.
Le *tasks* sono operazioni per *manipolare i concetti* nel dominio, ed il goal è il risultato di queste manipolazioni.
Attraverso task di input e output otteniamo il dialogo che fornisce l'interazione.

Con *evaluation* si intende la traduzione di task umane in task del sistema.



- Modello **ACM** (Association for Computing Machinery):



Per quanto riguarda l'*organizzazione dei controlli* e la *visualizzazione*, il raggruppamento è importante:

- Funzioni correlate → disposte *vicine*
- Azioni in ordine → disposte *sequenzialmente*
- Azioni frequenti → disposte *in vista*

Va prestata attenzione anche alle condizioni dell'ambiente fisico (ad esempio eventuali ostacoli) e agli aspetti legati alla salute e all'uso dei colori.

Progettazione del dialogo

Va progettato il dialogo utente-PC che avviene durante l'interazione. La scelta del tipo di interazione ha effetti sulla natura del dialogo e, di conseguenza, sull'efficacia dell'interazione. Si hanno 6 *tipi di interazione*:

- Command entry: si danno istruzioni al PC direttamente con comandi a parole, è adatto per utenti esperti.
- Menu e navigazione: si hanno comandi disponibili a schermo, adatto sia per task semplici che strutturate.
- Question / Answer: si pone una serie di domande all'utente e lo si guida passo passo nella task, il sistema controlla l'interazione.
- Riempimento form: è adatto solo per l'input di dati.
- Linguaggio naturale: utilizzato per task specifiche, può però generare ambiguità.
- Manipolazione diretta: permette l'interazione fisica immediata con gli oggetti dell'interfaccia, è adatta a tutti i tipi di task e si ha una visibilità dell'oggetto di interesse, azioni veloci, reversibili ed incrementali (es. interfacce point and click).

Sviluppo cognitivo secondo Piaget

La teoria di Piaget prevede 3 stadi nello sviluppo cognitivo:

- Stadio senso-motorio: il bambino impara a muoversi, toccare, muovere gli oggetti e valutarne caratteristiche e robustezza.
- Stadio operatorio-concreto: impara e confronta l'esterno degli oggetti.
- Stadio operatorio-formale: valuta l'uso e il significato degli oggetti.

Elaborazione di Jerome Bruner

Bruner identifica varie *mentalità*, le 3 più importanti sono quelle create dai 3 stadi di Piaget, con 3 modalità di rappresentazione:

- Esecutiva: si elaborano solo informazioni pratiche, immagini concrete di cose o azioni.
- Iconica: si immaginano le cose.
- Simbolica: le azioni e concetti vengono rappresentati con simboli.

L'immediatezza nella manipolazione diretta

Si è parlato prima di un tipo di manipolazione del sistema, la manipolazione diretta. L'*immediatezza* in questo contesto è il collegamento diretto di azioni e comandi dell'utente agli oggetti dell'interfaccia, e può essere:

- *Semantica*: mappatura U-I, cosa voglio fare e cosa l'Input mi permette di fare.
- *Articulatoria*: mappatura I-S, il comando che attiva una funzione del sistema.

Presentare ed inserire informazioni

Lo screen design è il meccanismo di output primario, ha come elementi la ridondanza degli items, il raggruppamento e l'allineamento, i colori, ed altro.

Una regola molto importante è *minimizzare il carico cognitivo* dovuto all'interazione (far ricordare poche cose all'utente).

Le 8 regole del dialogo sono:

1. Consistenza interna (sintassi e semantica) ed esterna
2. Feedback
3. Closure (inizio, centro, fine)
4. Strategie gestione errori
5. Azioni reversibili
6. Scorciatoie per esperti
7. Utente "iniziato" ha senso di controllo
8. Riduzione carico memoria a breve termine (display semplice).

Le 5 regole per la visualizzazione dei dati sono:

1. Standardizzare terminologia, azioni, e formati
2. Informazioni familiari all'utente e connesse alla task
3. Minimizzare il carico di memoria
4. Consistenza della visualizzazione ed dell'inserimento
5. Flessibilità nel controllo della visualizzazione.

Le 5 regole per l'inserimento dei dati sono:

1. Consistenza delle transazioni
2. Minori sono le azioni di input, minori sono le probabilità di errore
3. Minimizzare il carico di memoria
4. Consistenza dei dati inseriti con quelli mostrati
5. I dati inseriti e i formati di dati accettabili devono essere flessibili ma non ambigui.

Modelli di design

- **System-centered**: questo modello risulta comodo per il designer. Ad esempio in sistemi feature-oriented l'interfaccia mostra tutto quello che il sistema può fare (es. un'enorme lista di azioni disponibili).
- **User-centered**: questo modello è basato sulle caratteristiche dell'utente, viste come parametri di design. Ha 5 fasi principali: l'analisi di utenti e tasks, il design, la validazione, il testing e l'iterazione. Può essere diviso in due sotto-modelli:
 - Task-centered: (Standard ISO 9241-210) crea la struttura dell'interfaccia attorno alle user tasks e non alle funzioni di sistema. Questo modello è adatto a sistemi non molto estesi e richiede apprendimento.
 - Goal-centered: (I. I. Garrett) è incentrato sui goal dell'utente, vengono eliminate le task non rilevanti e ci si focalizza sul perché l'utente debba utilizzare il sistema.

ISO 9241-210

Standard per user-centered development process, fornisce guide sull'interazione sistema-uomo, per tutto il ciclo di vita del sistema interattivo.

I *6 principi fondamentali* dello standard:

1. Design basato su user, task e contesto
2. Utenti coinvolti in design e sviluppo
3. Valutazioni user-centered raffinano il design
4. Processo iterativo
5. Il design indirizza l'intera user experience
6. Il team di design include abilità multidisciplinari

Le *6 qualità del software* rilevanti durante lo sviluppo:

1. Funzionalità
2. Affidabilità
3. Usabilità
4. Efficienza
5. Manutenibilità
6. Portabilità

Con **usabilità** si intende la capacità del software di essere compreso, imparato, usato ed attraente per l'utente.

Con **qualità nell'utilizzo** si intende la capacità del software di permettere a specificati utenti, di ottenere specificati goals con efficacia, produttività, sicurezza e soddisfazione in specificati contesti d'uso.

I principi del dialogo secondo lo standard sono 7:

1. Adeguato alla task ed al livello di abilità
2. Autodescrittivo: cosa deve fare dopo l'utente

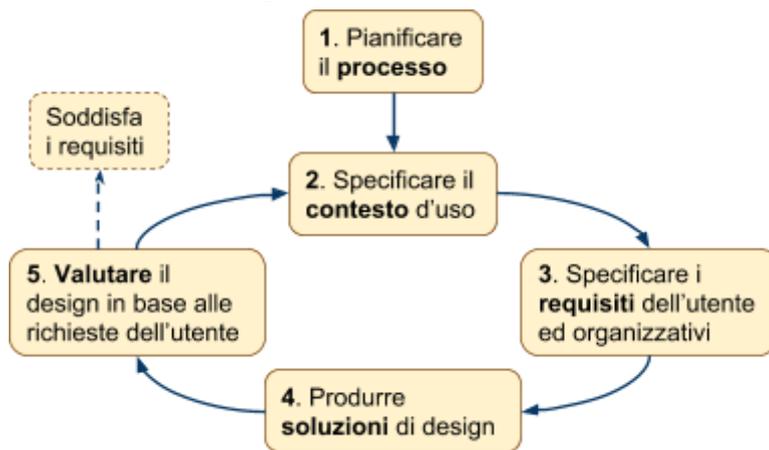
3. Controllabile dall'utente
4. Conforme alle aspettative dell'utente
5. Tollerante agli errori
6. Adeguato all'individualizzazione
7. Adeguato all'apprendimento

Quality in use: qualità che il prodotto dà all'utente al momento dell'utilizzo.

Product quality: qualità del prodotto in se.

Quality of the process: come i team di progettazione e sviluppo portano avanti le attività.

Processo di design human-centered



Ciclo di vita del sistema

Studio di fattibilità	Requisiti	Design	Valutazione	Sviluppo	
Meeting con gli stakeholders	Contesto d'uso	User Experience	Guide di stile	Inspection	Raccolta feedback
	Scenario	Valutazione sistemi esistenti	Prototipi	Testing	

Studio di fattibilità

- Meeting con gli stakeholders: vanno identificati gli *obiettivi*, gli stakeholders e i criteri di successo. Si deve creare una visione condivisa del design e decidere il ruolo dell'usabilità e del contesto e come si relazionano con il business goal.
- Analisi del contesto d'uso: è necessario incontrare il team di sviluppo per definire chi sono gli utenti e le loro task, i vincoli tecnici e dell'ambiente. Inoltre è utile per assicurarsi che tutti i fattori che infettano l'uso del sistema siano stati identificati prima di iniziare la progettazione.

- Scenario: serve a documentare come ci si aspetta che gli utenti portino avanti le task in uno specifico contesto, che è input sia del progetto che del conseguente testing.

Requisiti

- Valutazione di sistemi esistenti: si analizzano le versioni precedenti o la versione di un competitor, verificando e valutando i problemi di usabilità che possano essere usati come metriche qualitative e quantitative nella fase di design e per future valutazioni.
- Analisi dei log: nel caso di un sistema già sviluppato dà molte informazioni (come pattern di utilizzo ed errori più frequenti) senza dover coinvolgere direttamente gli utenti.
- Interviste: vengono effettuate interviste ad un campione rappresentativo degli utenti finali, identificato in collaborazione col cliente.
- Osservazione diretta: si annotano i problemi, le fonti di stress e di distrazione per gli utenti presenti nell'utilizzo dei sistemi esistenti.
- Contesto d'uso: descrizioni documentate dei vincoli tecnici del design.
- Scenari: brevi storie, decomposizione delle user task in azioni ed identificazione delle operazioni compiute dall'utente e dal computer.

Design

- Identificare ed usare linee guida: relative al contesto tecnologico, industriale, e di progetto. Rilevanti per il design delle schermate e del dialogo.
- Prototipi: di bassa fedeltà, per chiarire i requisiti e fornire velocemente risposte a dubbi di implementazione.

Valutazione

- Usability testing: un test reale fatto su un campione rappresentativo del target che esegue le task, misurando i reali risultati e confrontandoli con le aspettative del team. Si hanno 2 tipi di test: *formative test*, effettuato durante la fase di design per identificare velocemente i problemi di usabilità; e *summative test*, per valutare pienamente il sistema nello stato finale a confronto con gli user requirements.

Sviluppo

- Raccolta feedback: usando l'attuale sistema sviluppato si raccolgono pareri da utenti non supervisionati, per identificare quali problemi dovranno essere risolti nelle versioni future.

Usability Metrics

Anche chiamate le 3E, sono metriche basate sui 3 fattori della quality in use specificata nell'ISO 9241-210:

- **Efficacia:** l'*accuratezza* e la *completezza* con cui l'utente conclude la task (# errori).
- **Efficienza:** il rapporto tra l'*accuratezza* e la *completezza* della task e le *risorse utilizzate* per completarla (tempo per task).
- **Emozioni:** la *soddisfazione* (voto dell'utente).

Goal-Oriented Design

Schema di Garrett

Lo schema di Garrett è un processo lineare che va dall'astratto al concreto, è costituito da 5 sezioni:

1. *Superficie* → pagine web reali (testo, grafiche, bottoni, ...).
2. *Scheletro* → elementi posizionati in specifiche location.
3. *Struttura* → scelte organizzative riguardo gruppi di pagine, navigazione e gerarchie.
4. *Scopo* → feature e servizi offerti dal sito.
5. *Strategia* → cosa il proprietario del sito si aspetta di offrire e di ottenere dal sito.

Le decisioni ai piani bassi influenzano quelli alti, ma in alcuni casi l'influenza scende verso il basso. In questi casi le fasi precedenti non vengono chiuse fino a quando quelle successive non sono iniziate da un po' (es. redesign), ovvero si sovrappongono.

Con **dualità del web** si intende che l'*informazione* è il mezzo per la diffusione di contenuti, mentre l'*applicazione* è l'interfaccia per l'accesso a servizi remoti di vario tipo (questo è valido sia da sito a sito, che tra sezioni diverse dello stesso sito).

Piano: Strategia

Gli obiettivi del prodotto sono l'*integrazione con il business goal*, la *promozione dell'identità del brand* e le *metriche per il successo*.

I bisogni dell'utente indicano cosa vogliamo che l'utente ottenga dal prodotto. Per quanto riguarda questo punto è possibile fare riferimento alla gerarchia di Maslow:



Ogni oggetto che scegliamo risponda ad almeno un bisogno di Maslow, lo stesso deve fare il prodotto.

I **livelli di Norman** sono 3 livelli di elaborazione cognitiva che hanno impatto sul design, e sono:

1. *Viscerale*: come mi aspetto di sentirmi quando uso il prodotto (progettare cose belle e desiderabili).
2. *Comportamentale*: come voglio usare il prodotto (progettare prodotti che abbiano un comportamento che si adatti ai pattern mentali/aspettative).
3. *Riflessivo*: cosa voglio pensare su me stesso (progettare un prodotto per una “relazione a lungo termine”, dove ci cambia, ci fa imparare, raggiunge i nostri scopi).

Per quanto riguarda invece gli obiettivi non-utente si parla di ciò che riguarda i vincoli tecnici (es. funzionamento su tutti i browsers) o riguardo il tipo di organizzazione dietro al prodotto (commerciale, pubblica amministrazione, ...).

Le **tecniche di progettazione** goal-oriented prevedono tre elementi:

- Segmentazione degli utenti: si creano sottogruppi del target che siano omogenei riguardo ad alcune caratteristiche (*demografiche*: età, luogo; o *psicologiche*: interessi, motivazioni).
- Ricerche sugli utenti: le ricerche possono essere di *mercato* (fonte indiretta), domande *contestuali* (fonte diretta) o *analisi delle task* che si vuole un utente possa portare a termine.
- Personas: approccio narrativo al design, si hanno personaggi (personas) in un’ambientazione (contesto) seguendo una trama (caso d’uso). Lo scopo è indicare come può il sistema aiutare questi individui a ottenere il proprio goal assecondando le loro abitudini e comportamenti. I tipi di personas sono:
 - protagonista: target del progetto.
 - secondario: ha necessità particolari ed eccezionali, non si effettua redesign.
 - addizionale: né primario né secondario è soddisfatto dal design.
 - non-user: utenti di cui non si è tenuto conto durante il design.

Piano: Scopo

I **requisiti funzionali** sono indicati all’inizio del progetto ed indicano cosa il sistema dovrebbe fare.

Le **specifiche funzionali** sono indicate alla fine del progetto e descrivono cosa fa effettivamente il sistema.

Infine i **requisiti di contesto** indicano quali informazioni debbano essere incluse.

Gli **scenari** sono storie che raccontano in dettaglio come un utente raggiunge il suo goal portando avanti una o più task nel sistema. È importante descrivere le task, non i comandi dell’interfaccia del sistema ma specifica quali path di interfaccia sono influenzati.

Alcune *riflessioni* sulle task:

L'**impegno** è **proporzionato**, la dedizione di un utente sarà proporzionale all'utilità che si aspetta di ricevere dal sistema.

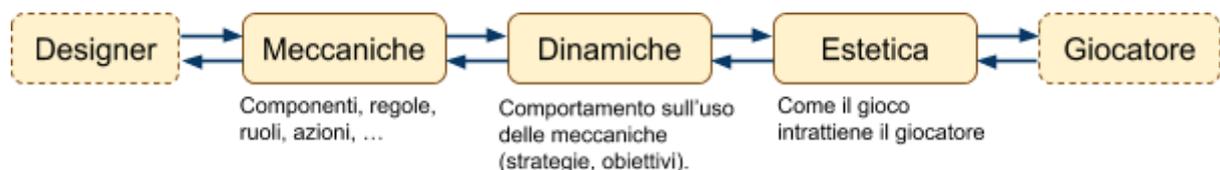
Spunto-Routine-Ricompensa: le attività che diventano routine non richiedono sforzo cognitivo. Nasce il ciclo abitudinario.

Il **ciclo abitudinario** prevede che il completamento di un'azione, o il successo di una scelta, rinforzino nella mente quell'azione/scelta più di altre potenzialmente ugualmente o maggiormente positive. Dopo alcune ripetizioni queste azioni diventano automatiche e le scelte non più richieste. Può risultare pericoloso ma semplifica l'utilizzo.

La **dissonanza cognitiva** avviene quando non si trova la soluzione ad un problema in un tempo ragionevole, questo abbassa la soglia di aspettativa.

Il **flusso** è lo stato d'animo attivato quando ci si concentra su una task, bisogna evitare che l'utente esca dal flusso. Per una migliore interazione l'orchestrazione tra le varie parti dell'interfaccia in una singola interfaccia è il meccanismo fondamentale (questo è chiamato invisibilità dell'interfaccia).

La **gamification** è la disciplina che cerca di migliorare la nostra percezione trasformando attività di routine in giochi, per stimolare l'effetto flusso (aumentando l'attenzione nelle task). Si compone di 3 elementi:



Le **motivazioni** di specifiche azioni possono essere di due tipi: estrinseche, ovvero esterne all'individuo come la vincita di un premio o l'evitare una punizione; intrinseche, ovvero interne all'individuo perché si sente uno stimolo ed una gratificazione nell'atto in sé.

I giocatori possono essere di 6 tipi:

1. *giocatori*: sono in cerca di una ricompensa data dal gioco.
2. *socializzatori*: vogliono interagire.
3. *spiriti liberi*: vogliono esplorare e creare.
4. *achievers*: vogliono diventare bravi ai meccanismi del gioco.
5. *filantropi*: vogliono aiutare altri ad aumentare l'utilità del gioco.
6. *distruttori*: vogliono controllare il gioco e cambiarlo.

Piano: Struttura

Per i modelli *applicativi*, ci si concentra sul design dei servizi interattivi (es. modelli concettuali). Per i modelli *informativi*, ci si concentra sul design della struttura dei dati (catalogare ed organizzare), come l'information architecture.

L'**interaction design** descrive i possibili comportamenti dell'utente e definisce come il sistema risponderà a questi. Il modello è in stile dialogo, non botta e risposta, il si usa un modello concettuale e delle convenzioni con cui l'utente è familiare, importante anche la gestione degli errori.

I **blueprint** sono disegni che definiscono le componenti per l'organizzazione del contenuto e come queste componenti siano connesse tra loro. Di solito si creano diverse versioni per ogni prospettiva (progettisti, clienti, ...), e non si hanno sintassi standard, bensì una leggenda.

La **storyboard** è una tecnica per illustrare attraverso immagini, la struttura degli step dell'esecuzione di una task, mostrando lo stato della schermata durante le fasi dell'azione.

Piano: Scheletro

In questo piano ci si occupa del **design dell'interfaccia** applicativa, con widgets ed altro. Inoltre ci si occupa del **design di navigazione** informativa, ovvero il mezzo per andare da un punto ad un altro del sito e per indicare le relazioni tra gli elementi. Infine il **design dell'informazione** prevede l'utilizzo di *wireframes*, ovvero il disegno delle parti fondamentali della pagina, per ogni schermata (usando strumenti come Balsamiq)

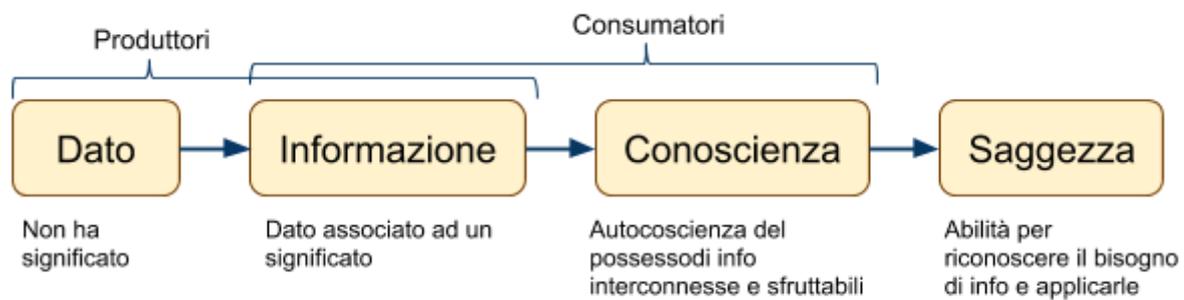
Piano: Superficie

- Estetica
- Contrasto ed uniformità
- Consistenza
- Linee guida

Information Architecture

In questa sezione si discute dell'organizzazione logica e della struttura semantica dell'informazione obiettivo, in modo che sia possibile ottenere le informazioni efficacemente.

Sia l'**information architecture** che l'**information design** hanno a che vedere con la presentazione delle informazioni. Il primo indica come gli oggetti sono legati tra loro, il secondo come viene progettata l'informazione per essere utilizzata al meglio. Il modo in cui organizziamo le cose riflette ed influenza il modo in cui le percepiamo.



Il **modello Cynefin**, ogni comprensione/decisione per un problema appartiene ad uno di questi 5 domini.



L'*organizzazione dell'informazione* può essere fatta attraverso alfabeti, mappe, con una struttura lineare o continua, numerica, per categorie o random (senza organizzazione).

L'**eccellenza grafica** prevede la possibilità di comunicare idee complesse in modo chiaro, accurato ed efficiente (ad esempio dati statistici).

Architettura dell'informazione

- **Strutturare:** determinare il livello di granularità (dettagli) dei dati e decidere come sono collegati tra loro.
- **Classificare:** definire le categorie ed i collegamenti tra queste.
- **Organizzare:** raggruppare componenti di informazioni in categorie specifiche e distinte.
- **Trovabilità:** capacità di una risorsa di essere localizzabile o fruibile (web).
- **Gestibilità:** bilanciamento tra i bisogni dell'utente e business goals.
- **Serendipity:** risultati utili, piacevoli, positivi a cui si arriva per caso e senza aver pianificato, mentre stiamo facendo o cercando altro.

Ecologia dell'informazione

Vi è una stretta relazione tra contesto, contenuto ed utenti.

Contesto: il vocabolario e la struttura di un sito sono condizionati dai fattori contestuali come lo scopo del progetto, le specifiche, la cultura, la mission, ...

Contenuto: il contenuto è influenzato dal controllo (centralizzato o distribuito), dal formato (testuale, audio), dalla struttura (completa o in crescita), dal volume (numero documenti) e dalla dinamicità (soggetto a cambiamenti).

Utenti: le differenze nelle preferenze e nei comportamenti degli utenti nel mondo reale si riflettono in differenti ricerche e necessità di informazione.

Approccio Top-Down e Bottom-Up

Top-Down: viene formulata una struttura generale del sistema senza entrare nel dettaglio delle sue parti. Ogni parte viene poi progressivamente raffinata. Si parte dai goals e da questi si arriva alla strategia più appropriata per raggiungerli.

Bottom-Up: le singole parti del sistema vengono caratterizzate in dettaglio, sono poi unite insieme per formare componenti più grandi, che vengono poi interconnessi in un sistema completo.

Componenti dell'Information Architecture

- *Organizzazione:* per argomento, cronologica, ...
- *Labeling:* per la rappresentazione delle informazioni.
- *Navigazione:* nella pagina e nel contenuto.
- *Ricerca:* query vs navigazione.

Strutture visibili

- Aiuti di navigazione: componenti per trovare le informazioni attraverso la navigazione piuttosto che con le query. Incoraggiare l'esplorazione dei contenuti (serendipity).
- Aiuti di ricerca: permettono all'utente di interrogare ed ottenere un set di risultati. Vanno gestiti problemi come query incorrette, assenza o eccesso di risultati, ...
- Contenuto e tasks: sono il punto d'arrivo per l'utente, cosa esegue effettivamente.
- Componenti invisibili: strutture di informazioni che non sono user-friendly ma possono essere utili ad altri componenti (algoritmi, dizionari, ...).

Processo di design dell'architettura

I requisiti dipendono dal management, dall'analisi degli utenti, dal contenuto e dai metadati.

Il **design** prevede blueprints, wireframes, il riordinamento di card per categorie (tecnica di categorizzazione delle task da parte di utenti/clienti). La **struttura** può essere piatta per siti semplici, o gerarchica per quelli più complessi.

Valutazione interna - Ispezione

L'ispezione avviene internamente al team di sviluppo, mentre il prodotto viene sviluppato. Questo risulta essere uno strumento molto economico ed inaccurato per

la valutazione dell'usabilità di un sistema. Si hanno 3 attività rilevanti: la guida cognitiva, l'analisi delle azioni e l'analisi euristica.

Cognitive Walkthrough

Guida cognitiva è un'esecuzione immaginaria passo passo di una task e la valutazione empirica del gradimento (si immaginano pensieri ed azioni).

Si seleziona una task con una certa interfaccia e si racconta una storia credibile riguardo ogni azione che l'utente deve eseguire per completare la task. Se non si riesce a raccontare c'è un problema di plausibilità dell'interfaccia. Gli aspetti fondamentali da rispettare sono:

- Prototipo dell'interfaccia
- Descrizione di una task
- Lista completa delle azioni per completare la task
- Chiara descrizione dell'utente e delle sue abilità ed aspettative

Non si deve valutare l'interfaccia ma la credibilità della storia che usa l'interfaccia, ciò potrebbe portare a revisionare sia la storia che l'interfaccia.

Analisi delle azioni

È un processo di valutazione che *esamina* la sequenza di azioni da fare per completare una task. Può essere di due tipi:

- **Analisi formale:** fa accurate predizioni del tempo speso da un utente nell'eseguire una task, si stima il tempo di ogni singolo step e vengono sommati. Questa analisi è complicata e lunga, ha lo scopo di predire la performance. Il modello più diffuso per l'analisi di azioni è GOMS:
 - Goal: qualcosa che l'utente vuole ottenere, ha una struttura gerarchica con goals e sub-goals.
 - Operatori: il più basso livello di analisi, azioni associate a precise tempistiche.
 - Metodi: passi per raggiungere il goal, si assume una sequenza conosciuta ed abituale.
 - Regole di selezione: rami delle gerarchie.

L'output è il tempo di esecuzione, verificando che i goal frequenti siano raggiunti in fretta.

- **Analisi informale:** ignora i micro-dettagli, valutando le azioni globalmente (es. scegli l'opzione X dal menu Y). Lo scopo è quello di determinare gli step con peso maggiore e dopo individuare fonti di complessità / perdita di tempo / disorientamento. È più robusta e meno soggetta ad inesattezze. In sostanza verifica che l'esecuzione di una task non richieda uguale tempo al farla a mano, decide se l'aggiunta di una feature complica il resto dell'interfaccia, e decide se aggiungere più modi per portare a termine una task.

Analisi euristica

Usata per verificare l'aderenza del sistema alle linee guida identificate per il progetto e giustificare le deviazioni. Serve anche a valutare l'usabilità di un sistema indipendentemente dalle task per cui è stato progettato (domain-independent), vi è una comparazione dell'applicazione con principi generali, universalmente conosciuti.

Si utilizzano linee guida, possono essere rispettate tre fonti per le guidelines:

- 10 di Nielsen e Molich
- 20 di Weinschenk e Barker
- 9 capitoli di Userfocus

Valutazione esterna - Testing

Questa valutazione è esterna al team, si ha la partecipazione di potenziali utenti, che forniscono considerazioni realistiche, credibili e dimostrabili sul sistema. Per effettuare questa fase si hanno 2 modi: tramite *statistiche* (si hanno molti individui ed un team specializzato che analizzi), tramite il *senso comune* (si hanno pochi individui e nessun team specializzato).

Full Usability Testing (Deluxe)

Di seguito le caratteristiche principali di questo modello:

- *Formalizzato*: protocolli, ruoli, ...
- *Quantitativo*: valutazioni obiettivi e giustificato statisticamente.
- *Parallelo*: tanti utenti insieme, si dà solidità al risultato.
- *Costoso*: sviluppo sospeso.
- Utile *usato alla fine*, come test sommativo.

Discount Usability Testing (Guerrilla)

- *Informale*: chat in una stanza.
- *Intuitivo*: si hanno risultati indicativi.
- *Sequenziale*: ogni test è valutato e risolto prima del successivo.
- *Economico*: pochi utenti, nessun team, nessuna sospensione dello sviluppo.
- Utile *usato come test formativo*, identifica i problemi da risolvere in fretta.

Per progetti a basso budget (come siti web) si sceglie il Discount UT, utilizzando scenari, pensare ad alta voce e valutazioni euristiche.

Gli **scenari** sono prototipi su cui si basa il test (meno costoso). Questi prototipi possono essere verticali, ovvero implementano in modo completo una sola funzionalità (tra tutte le features); od orizzontali, ovvero implementano una piccola parte di ogni feature.

Gli **errori** vengono organizzati per gravità, dividendoli in: catastrofici, non viene completata l'attività; seri, l'attività è terminata con rallentamenti o la qualità dell'output è compromessa; cosmetici, l'utente nota dei rallentamenti nelle performance.

Il problema è che il Deluxe UT non identifica la gravità di un problema, molti troveranno gli stessi errori catastrofici, pochi troveranno i seri e dopo molte iterazioni verranno trovati anche i cosmetici.

Il Discount UT invece, è sequenziale, quindi permette di accelerare il processo di identificazione degli errori perché il primo partecipante trova molti errori catastrofici che vengono risolti, il secondo avrà a che fare con un prodotto migliorato e troverà i rimanenti catastrofici e alcuni seri, e così via.

Classificazione errori

La gravità di un errore dipende da: *frequenza* (quanti utenti lo riscontrano), *impatto* sulla task, *persistenza* (quante task influenza). Per l'impatto si può trattare di errore cosmetico, minore, maggiore, catastrofico o d'implementazione.

La **curva d'urgenza** non è altro che un grafico che mostra l'impatto vs. la frequenza di un errore. Viene decisa una "soglia" d'urgenza sopra la quale l'errore va risolto immediatamente, sotto viene risolto alla prossima release.

Fasi di un test

Le fasi di un test sono indicativamente le seguenti:

- Design: che può essere di tipo formativo o sommativo, può avere portata globale, verticale od orizzontale, la logistica può comprendere assistenti, partecipanti.
- Pilot: un test da svolgere fuori quota (senza statistiche), per verificare che tutto funzioni.
- Utenti: vanno scelti utenti simili al target tenendo conto di problemi etici, privacy e stress.
- Sistema: il sistema su cui fare i test può consistere in: disegni su carta, mock-up su PC, Wizard of Oz (niente funzioni core, le parte non realizzate sono portate avanti da un essere umano), prototipi, sistemi funzionanti.
- Dati: i dati di un test possono essere *di processo*, osservazioni su cosa fanno i partecipanti e perché durante il test, si dà una valutazione qualitativa; o *bottom-line*, si misura il tempo per ogni task, il grado di successo e il numero di errori, si dà una valutazione quantitativa.
- Scopo: lo scopo della valutazione di un test può essere o *formativo*, per suggerire modifiche e migliorie, o *sommativo*, per valutare l'aderenza di un prodotto alle metriche di design adottate ed alle aspettative.

Metriche di usabilità

Le metriche possono essere di due tipi:

- **Quantitative**

- *Successo*: il completamento o meno di una task (si può misurare anche in percentuale).
- *Tempo*: il tempo dall'inizio alla fine dell'esecuzione (si fissa una soglia accettabile).
- *Errori*: se ne calcola il numero.
- *Efficienza*: la relazione tra percentuale di successo e tempo.
- *Apprendimento*: performance degli utenti alla prima esplorazione del sistema confrontate con quelle dopo un certo periodo di utilizzo.
- **Qualitative**
 - *Soddisfazione*: si misura la soddisfazione dell'utente tramite vari strumenti.

Metriche post sessione di testing prevedono:

- Si pone un *questionario* post scenario: 3 domande e likert scale (livello di accordo con la domanda posta).
- Si misura l'usabilità del sistema: 10 frasi positive/negative e likert scale.
- *Net promoter score*: si valuta la lealtà del cliente ad un brand, gli si chiede se lo consiglierebbe [(#promotori - #detrattori) / #partecipanti].
- Valutazione con delle card da associare al design.

Metriche su siti web prevedono:

- *Analytics* che misurano i visitatori e le pagine visitate.
- *Click-through rate* che misura il rapporto tra il numero di visitatori a cui viene mostrato un bottone e quelli che cliccano.
- Drop-off rate è il numero delle persone che iniziano una procedura diviso il numero di quelli che la finiscono.
- *A/B test* è la comparazione tra due possibili ipotesi dello stesso design, l'obiettivo è identificare i cambiamenti delle pagine web che incrementano il risultato interessato.
- *Return on Investment Data* pone l'attenzione sui dati che aumentano il valore del prodotto per la compagnia (più elementi venduti, meno errori, meno costi, ...).

Teoria fondata prevede la costruzione di una teoria quantitativa attraverso la raccolta sistematica e l'analisi di dati qualitativi. I ricercatori rivedono i dati ottenuti, identificano le idee e i concetti e li taggano con codici, i codici vengono poi raggruppati in concetti e successivamente in categorie (che formano le basi della nuova teoria).

CAO=S Design Approach

Concetti, Attori, Operazioni generano Strutture.

Questo modello è utilizzato per adottare un **approccio** di design **goal oriented** in progetti con vincoli economici. QUando non c'è un budget per un'analisi sistematica dell'utente target o per coinvolgere un esperto di usabilità o gli utenti nel design.

Studia il tipo di *informazioni* (Concetti) che l'applicazione deve manipolare per conto dei tipi di *utenti* (Attori) fornendo *comandi* (Operazioni). Una corretta analisi di queste permette di generare tre tipi di strutture, gestite dal modello:

- Viste: per la presentazione, modelli di schermate view+comandi.
- Strutture Dati: per il salvataggio persistente dei dati.
- Navigazione: per i meccanismi di movimento.

Alcune *cause* di *mancaza di usabilità* sono la distanza (linguistica od operativa) tra il modello espresso dal sistema e quello che si aspetta l'utente, eventuali complicazioni procedurali, come il numero eccessivo di operazioni, o l'incompletezza delle viste, quando non vengono mostrate informazioni utili.

Componenti di CAO=S

- Concetti: non sono le strutture dati manipolate dall'applicazione, ma il modo in cui l'utente percepisce e comprende l'informazione.
- Attori: differenziati non per le loro caratteristiche ma per il ruolo che hanno nell'applicazione, ed il sistema proporrà loro diverse interfacce.
- Operazioni: azioni disponibili sull'interfaccia corrispondenti a manipolazioni dei concetti.
- Strutture: modo in cui il designer implementa le idee di operazioni sui concetti dell'attore come widget dell'applicazione.

Analisi dei requisiti

Vengono raccolti bisogni e desideri, analizzati come requisiti e rappresentati come specifiche. Lo scopo è determinare se i requisiti sono non chiari, incompleti, ambigui e da risolvere.

Fondamentalmente vengono registrate le *differenze* tra i concetti degli stakeholder e le strutture concettuali. poi si progettano *strutture* percepibili dall'utente in modo che trasmettano i concetti. Infine si progettano le *operazioni* come manipolazioni dei concetti degli attori e si mappano in funzione del sistema.

C - Concetti

I concetti vengono spesso rappresentati tramite nomi ed aggettivi. L'interfaccia deve indicare operazioni sui concetti, non funzioni sulle strutture dati.

- Problemi di standardizzazione: le scelte lessicali degli attori diretti hanno sempre la priorità (es. abbreviazioni).
- Differenze lessicali: i termini accettati da tutti, anche se preferiti da nessuno.
- Differenze concettuali e polisemie.

Ogni concetto deve avere almeno un attore che ne crea un'istanza, le azioni su un sito non sono concetti ma metafore dei concetti (es. carrello metafora di acquisto, forum metafora di scambio di messaggi).

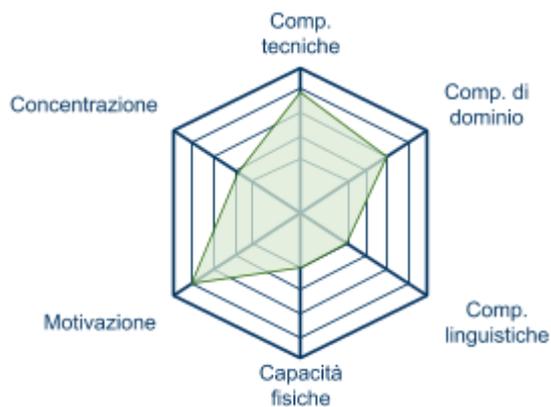
A - Attori

Gli attori possono essere:

- **Diretti**: quelli che useranno personalmente il sistema, organizzati per ruoli.
- **Indiretti**: quelli che possono esprimersi nelle caratteristiche del sistema ma non useranno l'interfaccia: tecnici, clienti, leggi, specifiche dell'architettura, librerie.

Analisi utente: si identificano 6 basic features con un chiaro impatto sull'implementazione e si caratterizza l'utente con quelle. Non ci si chiede "che livello di X ha l'utente?" ma "per che livello di X vale la pena progettare il sistema". Si effettua un'analisi costi-benefici sul costo associato al livello di X nella progettazione.

Per ogni feature si assegna un punteggio numerico e poi si mappa la valutazione dell'utente in uno strategy diagram:



Ogni attore ha un diagramma con un punteggio per ogni categoria.

In generale quando un livello di skill è alto il sistema è più facile da progettare.

Uno **scenario** è associato ad una **persona** per evidenziarne le caratteristiche ed il comportamento. È importante non scegliere personaggi estremi o indefiniti, ma interessanti, peculiari, che siano difficili da gestire ma pienamente nel target.

Ogni azione del diagramma è associata a differenti tipi di azioni per supportare l'utente (ed eventuali conseguenze in caso non siano rispettate):

- concentrazione + motivazione → catturami → abbandono
- capacità fisiche + competenze linguistiche → aiutami → slip
- competenze tecniche + competenze di dominio → spiega → mistake
- altro livello / centro → non preoccuparti → irritazione

Verbosità dell'interfaccia: in molti casi, la soluzione ad un basso valore in una basic feature è l'aggiunta di spiegazioni testuali. Se le competenze di computer sono basse avremo un'interfaccia densa, se sono nella media si possono usare spiegazioni attivate su richiesta.

O - Operazioni

Le operazioni sono essenzialmente le CRUD: Creazione, Vista, Aggiornamento, Cancellazione. Possono essere: semplici, complesse, ripetute, comuni, rare, frequenti, sporadiche, dipendenti/indipendenti dal dominio.

Creazione: generazione di una o più istanze del concetto allo stato iniziale. Specificando tipo, valori di default, persistenza, molteplicità, ...

Vista: visualizzare una o più istanze del concetto in modo comprensibile (full individual view / individual reduced view / multiple view (lista)).

Aggiornamento: modifica di una o più proprietà di una o più istanze. Può essere globale o specifico, multiplo o singolo.

Cancellazione: eliminazione o archiviazione.

S - Struttura

Creazione di un diagramma su come A è in grado di effettuare O su C.