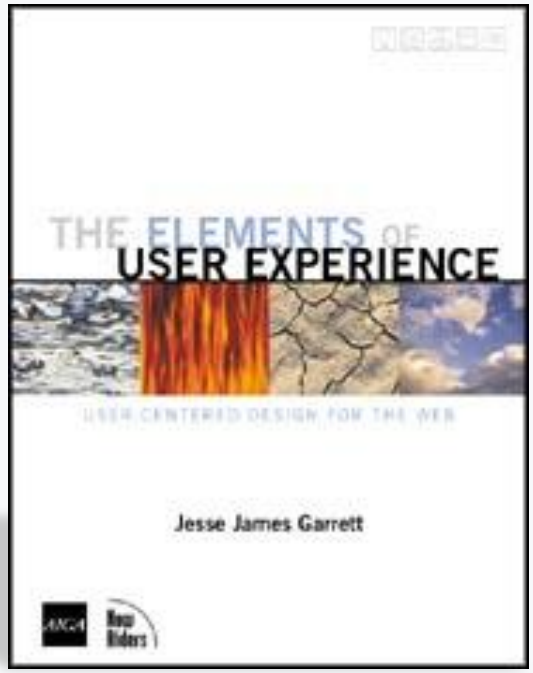
WEB DESIGN

User Experience (UX) - 2

Prof. ssa Romina Eramo
Dipartimento di Scienze della Comunicazione
Università degli Studi di Teramo
reramo@unite.it

Riferimenti

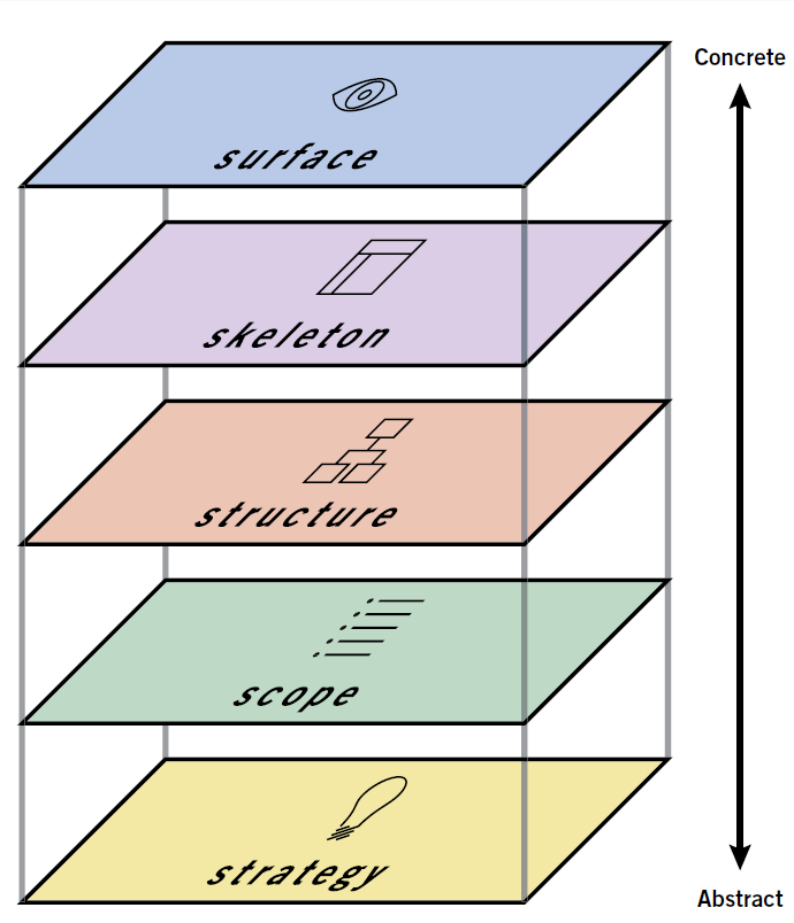


http://jjg.net/ia/visvocab

User Experience 5 piani (livelli)

design

concept



Strategy



- Pianificare gli obiettivi e le finalità del progetto
- Obiettivi del prodotto: *cosa vogliamo ottenere da questo prodotto?*
  - Busines goal
  - Panoramica del prodotto (lista di caratteristiche del prodotto)
  - Competitors
- Esigenze dell'utente: *cosa vogliono ottenere i nostri utenti?*
  - Ricerca utente
  - Bisogni e obiettivi
  - Segmentazione
  - Contesto
  - PERSONAS

Definire la Strategy

- La maggior parte dei siti **non fallisce** per la tecnologia o per la UX…
- …falliscono perché nessuno ha risposto chiaramente a due domande:
  - **Cosa vogliamo ottenere da questo prodotto?**
  - **Cosa vogliono ottenere i nostri utenti?**
- Da queste due domande nascono:
  - **Product Objectives** (interni, dall'organizzazione)
  - **User Needs** (esterni, dagli utenti)
- Insieme definiscono la **Strategy Plane**, la base di tutte le decisioni di UX

Product Objectives

- Spesso esistono solo come **intesa implicita** → fraintendimenti, obiettivi diversi
- Obiettivo: renderli **espliciti e condivisi**
- Le generalizzazioni tipiche:
  - “Far guadagnare l’azienda”
  - “Far risparmiare l’azienda”
- Troppo generico ≠ troppo specifico:
  - Troppo generico → non guida le scelte
  - Troppo specifico → descrive la soluzione, non il problema
- Serve definire **condizioni di successo**, non il percorso per arrivarci

Brand Identity

- **Brand ≠ solo logo e colori**

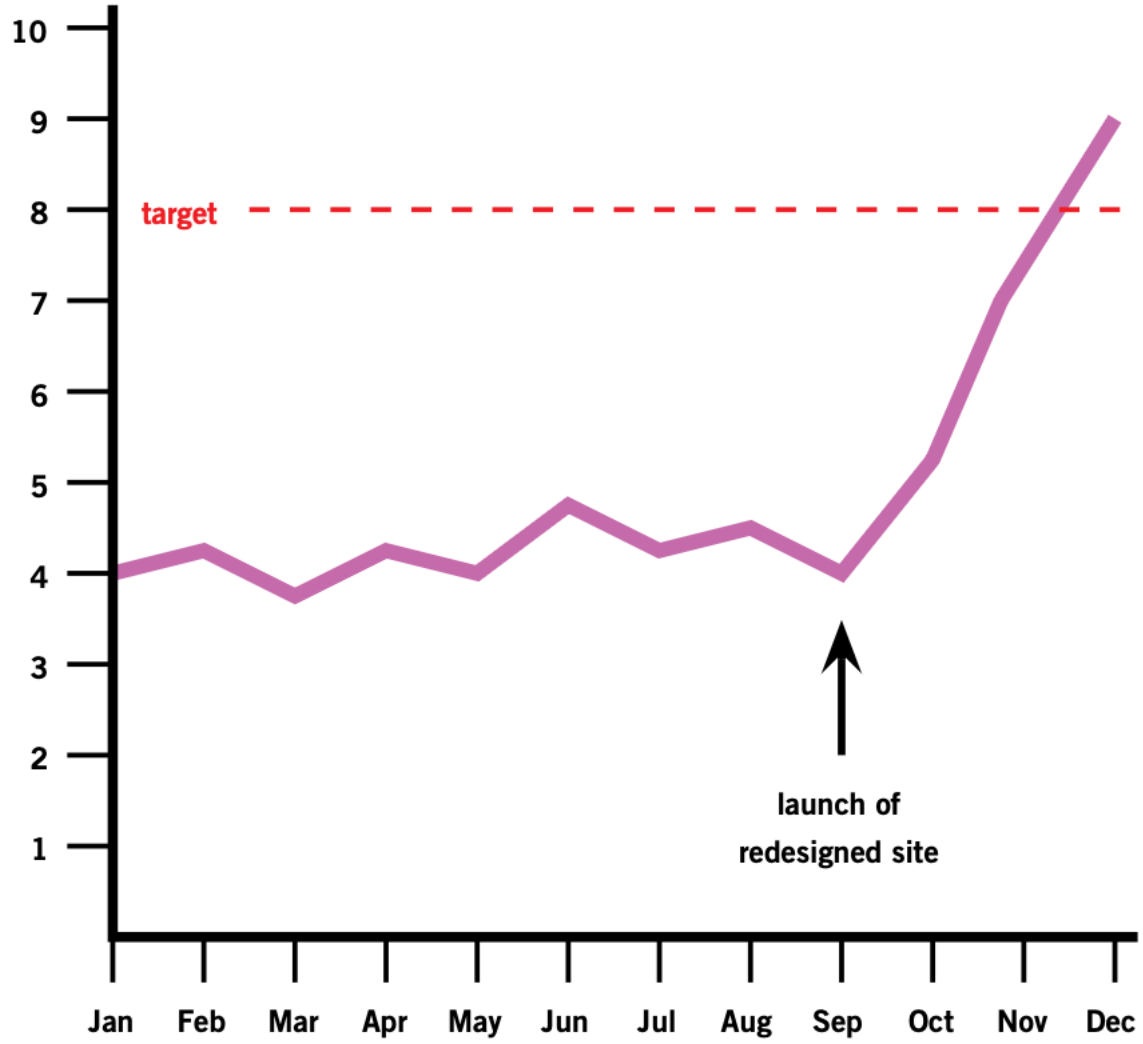
→ è un insieme di **associazioni concettuali ed emozionali**

- Ogni interazione con il prodotto crea un'impressione sul brand:
  - Che tu lo voglia o no
- Obiettivo strategico tipico:
  - Allineare l'esperienza d'uso con l'identità di brand desiderata
- Vale per:
  - Aziende commerciali
  - PA, non-profit, individui
- Codificare la brand identity come **product objective esplicito** → più controllo sull'impressione percepita

Success Metrics

- Domanda chiave: **Come sappiamo se abbiamo avuto successo?**
- **Success metrics** = indicatori concreti per misurare:
  - Se il prodotto raggiunge i **product objectives**
  - Se risponde alle **user needs**
- Esempi:
  - Tempo medio per visita (↑ esplorazione / ↓ frizione info)
  - Visite per utente registrato (engagement)
  - Ad impressions (per siti adv-based)
  - Riduzione chiamate al customer care
  - Riduzione tempo di chiusura vendita (intranet / tool interni)
- Attenzione:
  - Le metriche devono essere **collegate a comportamenti** influenzabili dal design
  - Nessuna metrica va letta **isolata** (es. può intervenire marketing, concorrenza, PR…)

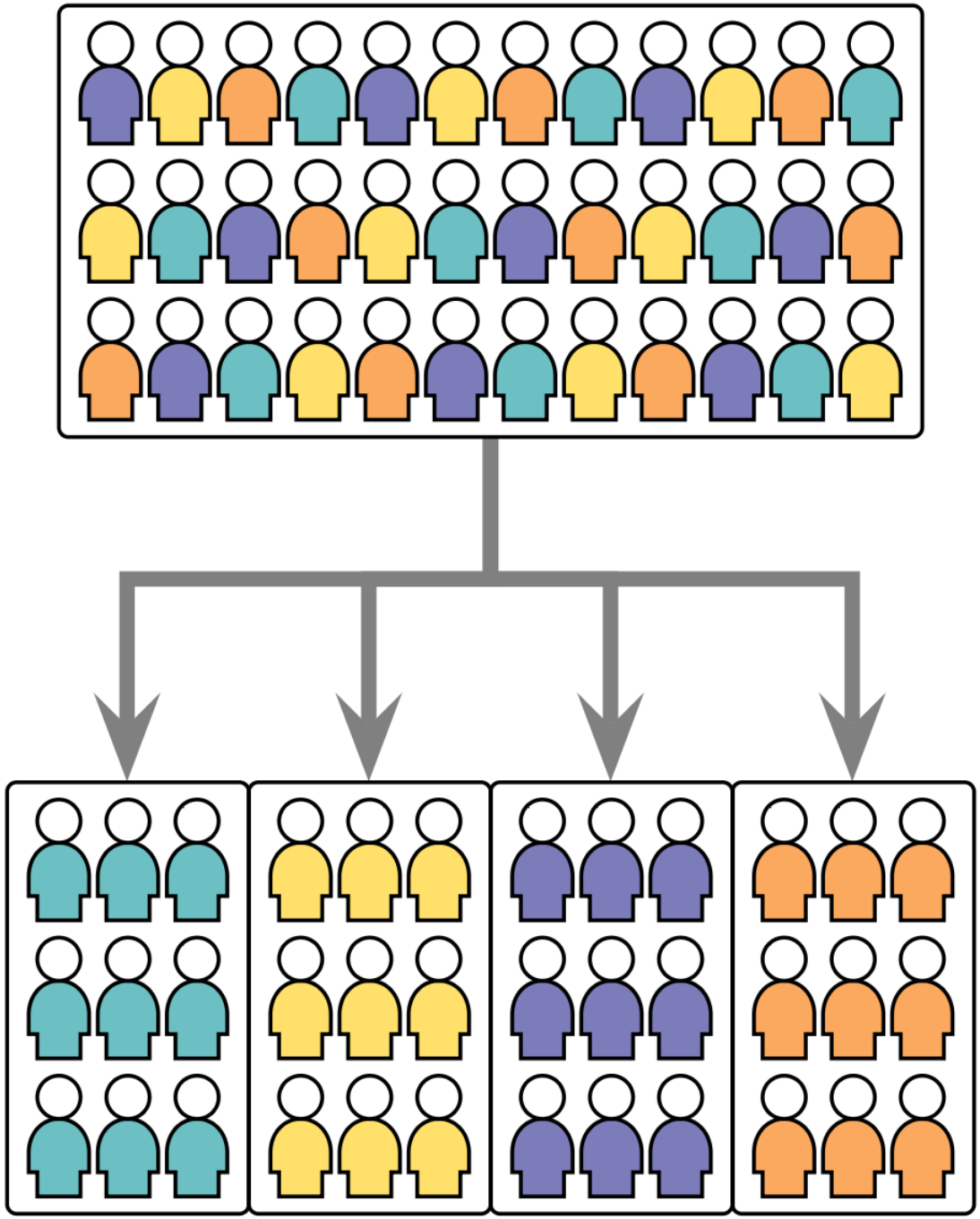
Success metrics are concrete indicators of how effectively the user experience is meeting strategic goals. In this example, measuring the number of visits per registered user per month indicates how valuable the site is to its core audience.



visits per month (registered users only)

User Needs & User Segmentation

- Non stai progettando per **te stesso**, ma per **altri**
- Serve capire:
  - Chi sono gli utenti
  - Cosa fanno
  - Di cosa hanno bisogno
- **User segmentation**:
  - Divide il pubblico in gruppi con bisogni simili
- Tipi di segmentazione:
  - **Demografica**: età, genere, reddito, istruzione…
  - **Psicografica**: atteggiamenti, valori, stile di vita
  - **Rapporto con tecnologia**: technophobe vs power user
  - **Expertise sul dominio**: principianti vs esperti
  - **Ruolo**: genitore vs studente, venditore vs manager…
- I segmenti possono:
  - Essere raffinati (split) o uniti (merge) dopo la ricerca
  - Avere bisogni anche **in conflitto** → decisioni strategiche (chi privilegiare? due percorsi diversi?)



User segmentation helps us understand user needs better by dividing the entire audience into smaller groups of people with shared needs.

User Research & Usability

- **User Research** = capire chi sono gli utenti e come si comportano
- Metodi principali:
  - **Surveys, interviste, focus group** → atteggiamenti, percezioni
  - **Contextual inquiry** → osservare gli utenti nel loro contesto reale
  - **Task analysis** → scomporre i compiti in passi concreti
  - **User testing** (prototipi o siti live) → far testare agli utenti ciò che hai progettato
  - **Card sorting** → capire come gli utenti organizzano l'informazione
- Obiettivo della **usability**:
  - Rendere il prodotto **facile da usare**
  - Valutare sia task concreti (task success) sia aspetti più astratti (es. messaggio di brand)

Laboratorio

- Per progettare la strategia per un sito web per trovare case in affitto o in vendita

-

Personas

- Problema: i dati numerici non “hanno un volto”
- Soluzione: creare **Personas**:
  - Personaggi **fittizi**, basati su dati reali
  - Rappresentano gruppi di utenti con bisogni specifici
- Ogni persona include:
  - Età, professione, contesto familiare
  - Rapporto con tecnologia
  - Obiettivi, frustrazioni, citazioni tipiche
- Uso pratico:
  - Tenere le personas **visibili** (poster, schede)
  - Farsi domande: “Funzionerebbe per Janet? Confonderebbe Frank?”

→ Aiuta a mantenere il focus sugli utenti in ogni decisione

Personas

- Inventa personaggi di fantasia con una propria storia

Frank



"This stuff is all new to me. I want a site that will explain everything."

Frank is interested in learning how he can turn his hobby of making furniture into a business.

Age: 37
Occupation: School bus driver
Family: Married, one child
Household income: $60,000/year

Technical profile: Somewhat uncomfortable with technology; Apple iMac (about two years old); DSL Internet connection; 8-10 hours/week online
Internet use: 100% at home; entertainment, shopping

Favorite sites:

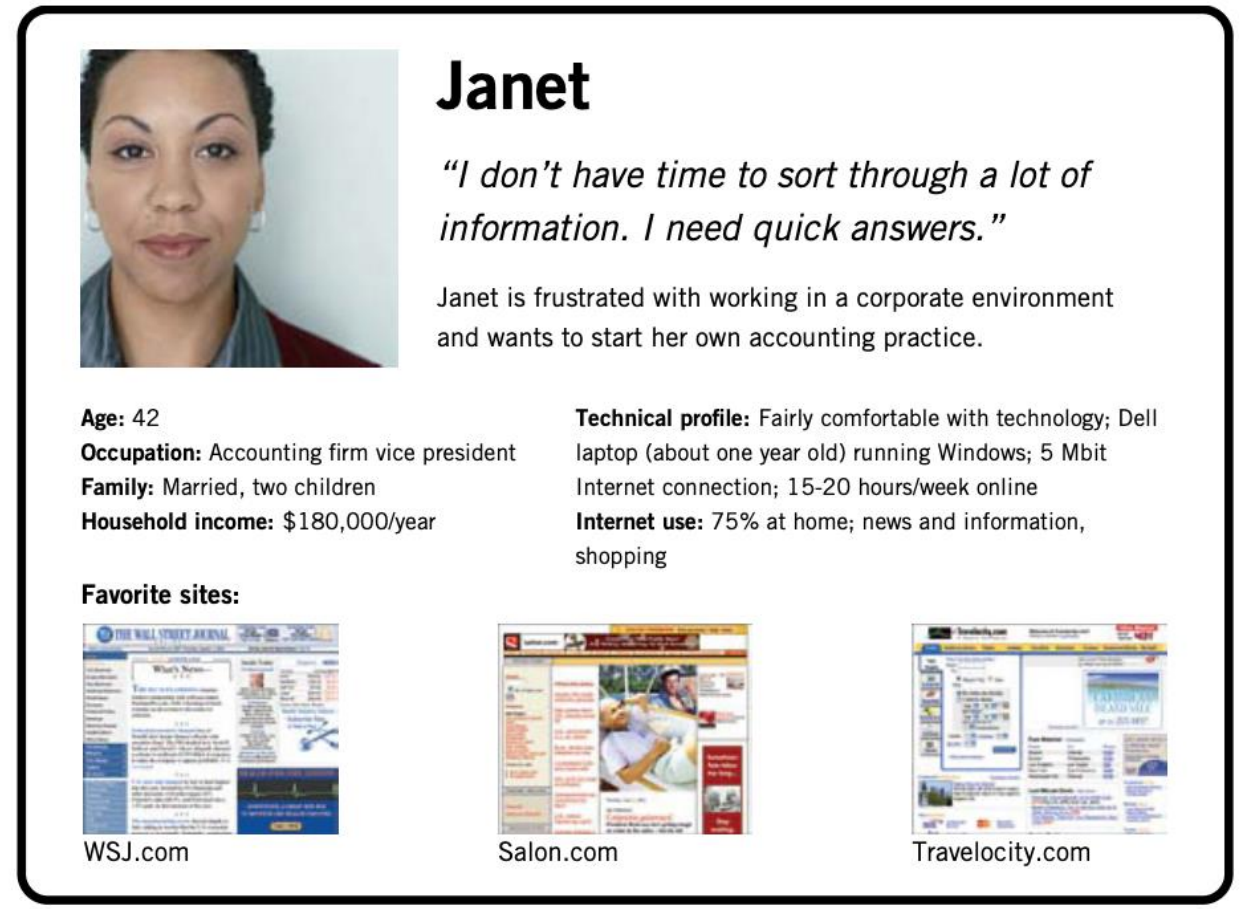

ESPN.com


moviefone.com


eBay.com

Personas

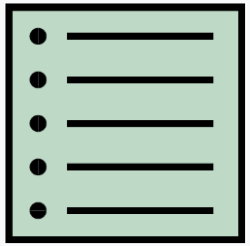
- Inventa personaggi di fantasia con una propria storia



Team Roles, Strategy Document & Process

- La **strategia** riguarda tutti: designer, dev, PM, stakeholder
- Attori chiave:
  - **Stakeholder**: decisori senior interessati dall'esito del prodotto
  - **Rank & file**: chi lavora sul campo, parla coi clienti, conosce i problemi reali
- Output principale:
  - **Strategy / Vision document**:
    - Product objectives + User needs
    - Relazioni tra obiettivi, contesto, citazioni da stakeholder e utenti
- Buone pratiche:
  - Documento **sintetico**, leggibile, non un tomo infinito
  - **Condividerlo** con tutto il team (non chiuderlo in un cassetto)
  - Considerardo un documento **vivo**, che può essere raffinato nel tempo

Scope



- Definizione dei requisiti, specifiche funzionali, fonti dati, scenari
- Caratteristiche dell'app
  - Cosa fa?
- Requisiti prioritari
  - Vincoli, regole, ecc.
- Tipologia di contenuto gestito
  - Come gestisce testo, video, audio
  - Provenienza dei dati (API esterna, servizio web, DB …)
- Scenari (utilizzando Personas)
  - Descrivi come le persone possono interagire con l'app

Definire lo Scope

- Definire lo **scope** è:
  - un **processo** utile → fa emergere problemi e conflitti in anticipo
  - un **prodotto** utile → documento di riferimento condiviso
- Serve a:
  - chiarire **che cosa includiamo** nel progetto
  - chiarire **che cosa NON includiamo** (adesso)
- Obiettivo: **ridurre ambiguità** nel processo di design

Cosa succede senza scope

- Prodotto in **beta eterna**: sempre “quasi pronto”, mai finito
- Sintomi:
  - features che cambiano continuamente
  - nessuna lista chiara di requisiti
  - nessuna roadmap, nessuna milestone, nessun “fine progetto”
- Se nessuno sa **qual è lo scope**, nessuno può sapere **quando abbiamo finito**

Perché definire i requisiti (Reason #1)

1. Sapere cosa stai costruendo

- Obiettivi e funzionalità diventano **concreti e condivisi**
- Tutti sanno:
  - che cosa il prodotto deve fare
  - quando un requisito è soddisfatto
- Evita l'effetto "telefono senza fili":
  - ogni membro del team ha una versione diversa del prodotto in testa
- Permette di:
  - assegnare meglio responsabilità
  - vedere connessioni tra requisiti (es. documentazione e schede prodotto)

Perché definire i requisiti (Reason #2)

2. Sapere cosa NON stai costruendo (ora)

- Tante idee possono sembrare buone, ma:
  - non tutte sono allineate alla **strategia**
  - non tutte stanno nel **tempo** e nel **budget**
- I requisiti chiari aiutano a:
  - valutare nuove idee lungo la strada
  - decidere cosa entra ora e cosa passa a **versioni future**
- Obiettivo: evitare la **scope creep** (il “pallone di neve” che cresce e travolge tempi e costi)

Funzionalità e Contenuti

- Sul piano dello scope ci chiediamo:
- **Che cosa andremo a costruire?**
- Due grandi dimensioni:
  - **Product as functionality** → feature, interazioni, strumenti
  - **Product as information** → contenuti, testi, immagini, media
- Documenti chiave:
  - **Functional Specifications** → requisiti funzionali
  - **Content Requirements** → requisiti di contenuto
- Anche se diversi, funzionalità e contenuto si definiscono in modo simile:

→ entrambi sono **“feature”** da progettare

Definire i requisiti

- Alcuni requisiti sono **globali**:
  - brand, browser support, sistemi operativi…
- Altri sono **specifici di una feature**
- Fonti dei requisiti:
  - **Utenti** (ideale)
  - **Stakeholder interni** (più frequente)
- Tipi di requisiti che emergono:
  1. Cose che **le persone chiedono esplicitamente**
  2. Cose che **credono di volere**, ma che nascondono un bisogno più profondo
  3. Cose che **non sanno ancora di volere**, ma emergono da brainstorming e visione

Come scrivere buoni requisiti

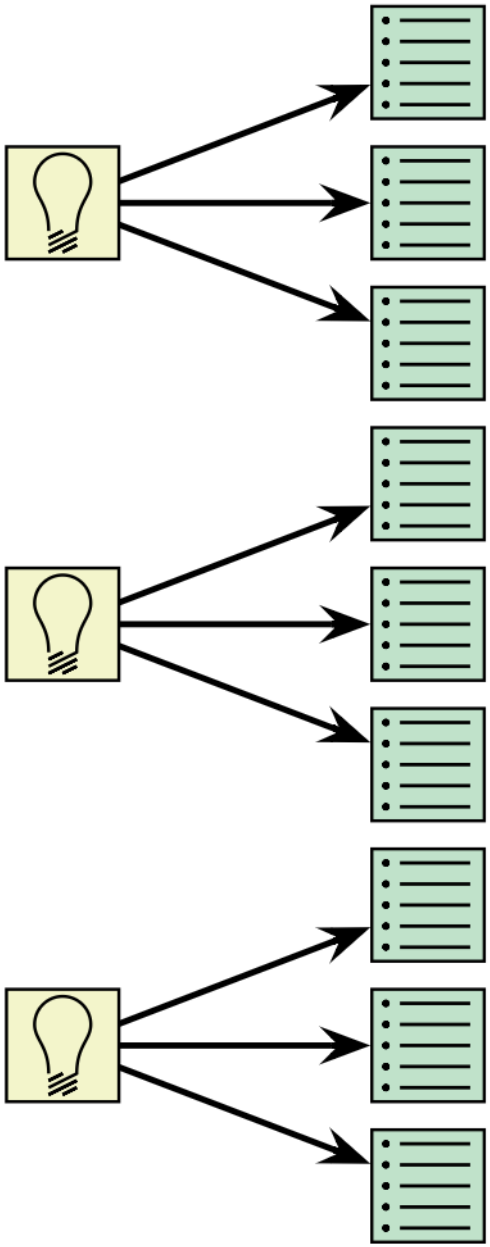
- **Essere positivi**
  - Non “il sistema NON deve…”, ma “il sistema FA X per evitare Y”
- **Essere specifici**
  - Evitare formule vaghe tipo “i contenuti popolari saranno evidenziati”
  - Definire criteri chiari: “Video con più visualizzazioni negli ultimi 7 giorni in cima alla lista”
- **Evitare linguaggio soggettivo**
  - “Stile hip e flashy” è impossibile da verificare
  - Meglio riferirsi a **linee guida di brand** o criteri misurabili
- **Rendere i requisiti falsificabili**
  - Es.: “Supporta 1.000 utenti simultanei” → verificabile
  - Non: “Sistema ad alte prestazioni”

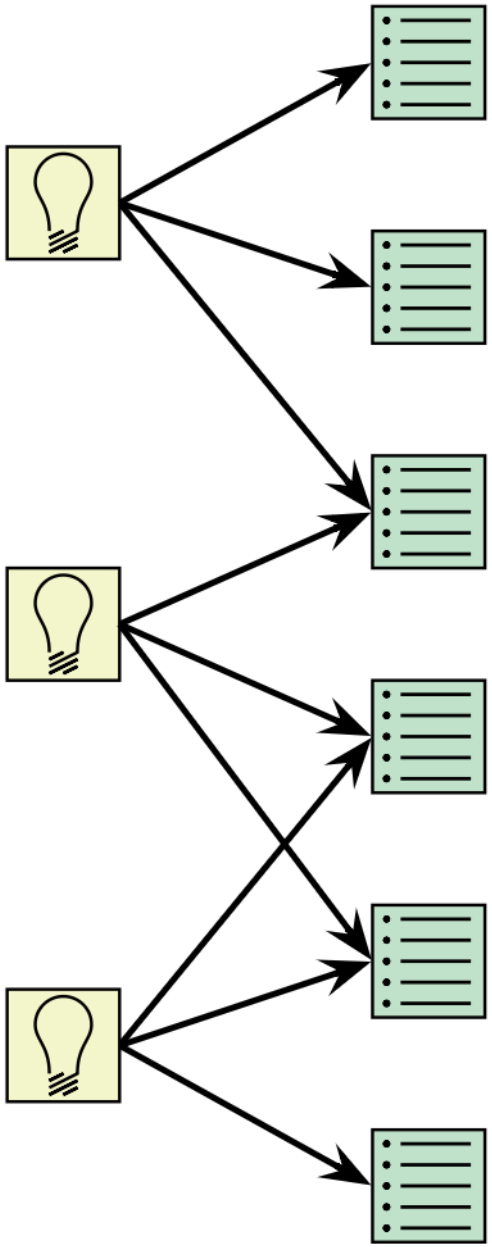
Requisiti di contenuto

- Contenuto ≠ solo testo:
  - testo, immagini, audio, video, PDF, ecc.
- Attenzione a **scopo vs formato**:
  - “FAQ” è un formato, non uno scopo
  - lo scopo è “fornire risposte rapide alle domande più frequenti”
- Ogni contenuto dovrebbe avere:
  - stima di **dimensione** (parole, pixel, MB)
  - chi è **responsabile** di crearlo e aggiornarlo
  - **frequenza di aggiornamento** (giornaliera, mensile, occasionale)
- Contenuto efficace richiede **manutenzione continua**, non “lo pubblico e me ne dimentico”

Prioritizzare i requisiti

- Non è difficile **raccogliere idee** di feature
- Difficile è decidere:
  - cosa entra nello scope **ora**
  - cosa va in una **release futura**
- Criteri di priorità:
  - Allineamento con **product objectives**
  - Risposta ai **bisogni degli utenti**
  - **Fattibilità** (tecnica, economica, di tempo)
- Vincoli tipici:
  - Impossibile tecnicamente
  - Troppo costoso in risorse
  - Non compatibile con tempi di rilascio
- Lo scope è costruito sulla **strategia**:
  - Se una feature non è allineata alla strategia → è **fuori scope** (a meno di rivedere la strategia)





Sometimes a strategic objective will result in multiple requirements (left). In other cases, one requirement can serve multiple strategic objectives (right).

Conflitti, politica e strategia

- Le feature raramente esistono da sole:
  - possono entrare in conflitto tra loro
  - richiedono compromessi (es. semplicità vs integrazione con sistemi legacy)
- La definizione dello scope è anche:
  - **negoziazione** tra stakeholder con esigenze diverse
- Strumento chiave per risolvere i conflitti:
  - tornare sempre alla **strategia**
  - parlare di **obiettivi**, non di feature preferite
- Importante:
  - mostrare **empatia** verso le esigenze dei vari ruoli
  - usare la strategia come “arbitro” e non come arma