

Produzione Editoriale

Paolo Ceravolo

paolo.ceravolo@unimi.it

Editoria Digitale

- Il processo di produzione editoriale è il complesso insieme di attività coinvolte nella ideazione, sviluppo, e distribuzione di materiali editoriali come libri, riviste, giornali, manuali, ...
- La sfida è fare in modo che l'opera passi dal concepimento alla sua disponibilità per il pubblico
- Il processo può variare, tradizionalmente le fasi chiave sono:

Ideazione

Acquisizione dei contenuti

Stesura e Revisione

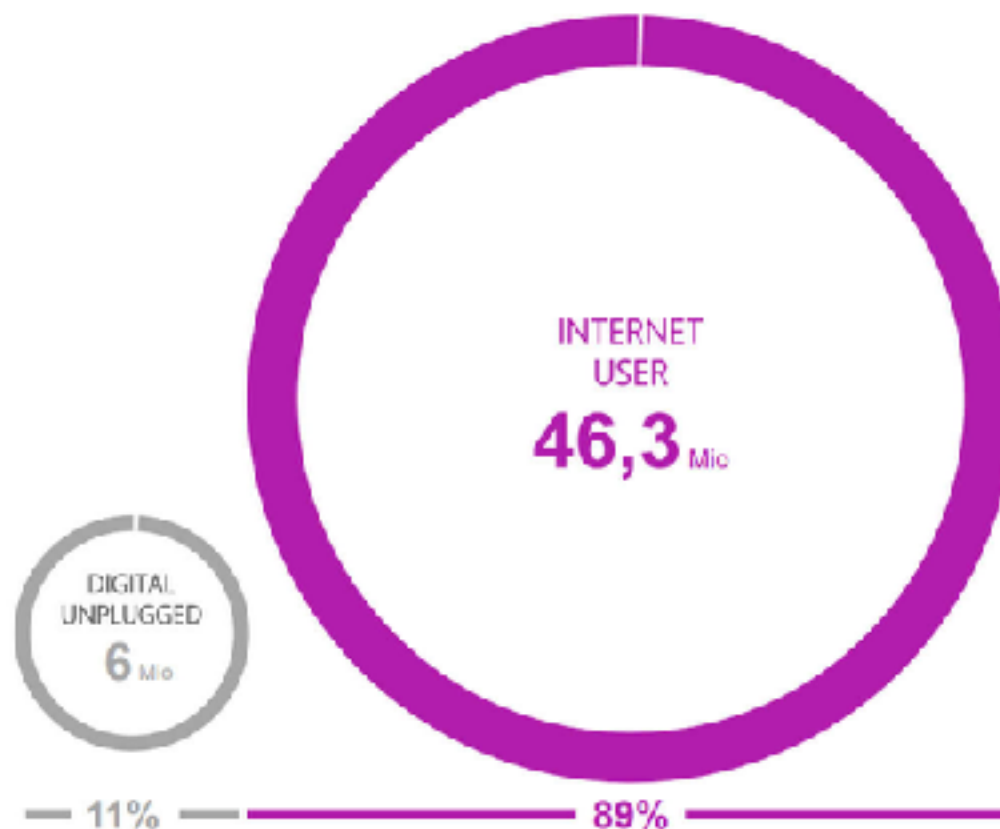
Progettazione grafica

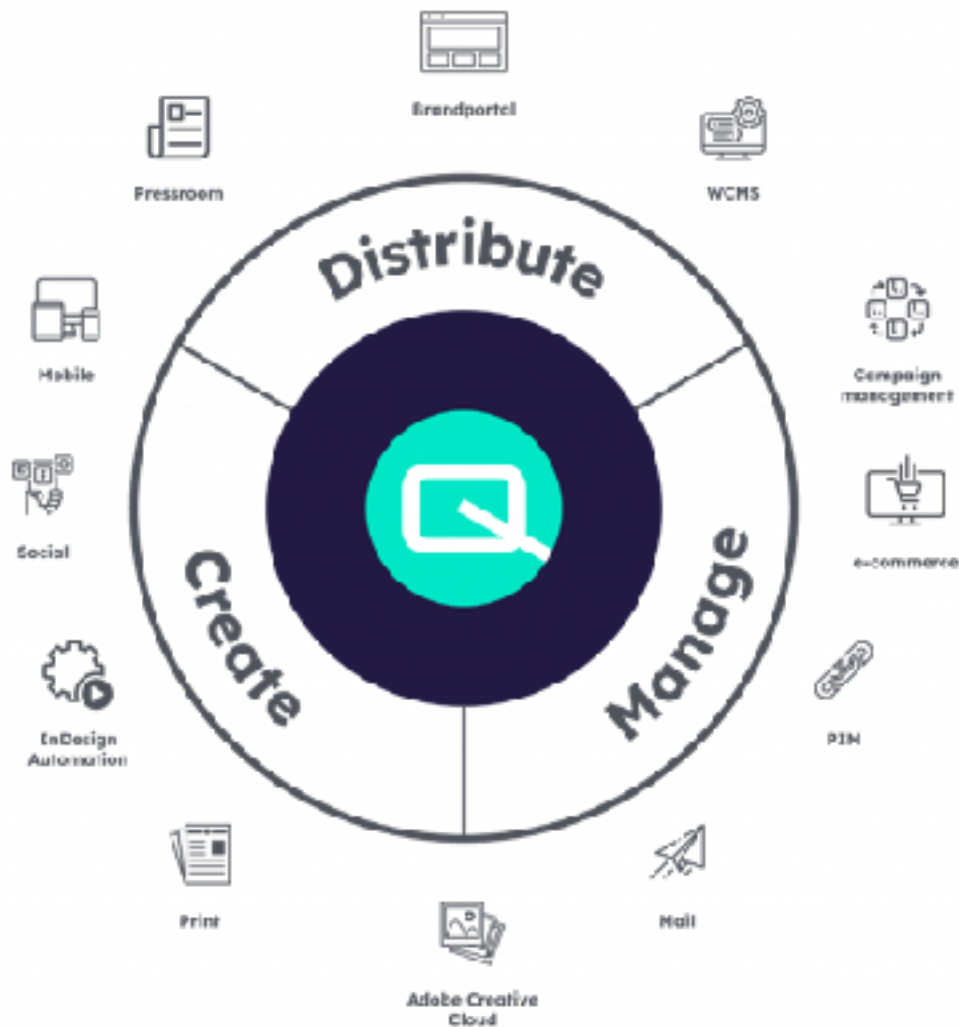
Produzione

Distribuzione

- Nell'attuale contesto economico e tecnologico la multicanalità è un concetto chiave
- Consente di raggiungere diversi segmenti di mercato
 - Il **consumatore** si **differenzia** per bisogni, cultura, predisposizione alla spesa
- Consente di tenere in considerazione le diverse forme di espressione
 - Contenuti diversi hanno diverso grado di **fruibilità** su diversi canali e sulla base del contesto di fruizione

La popolazione italiana e la multicanalità nel 2022



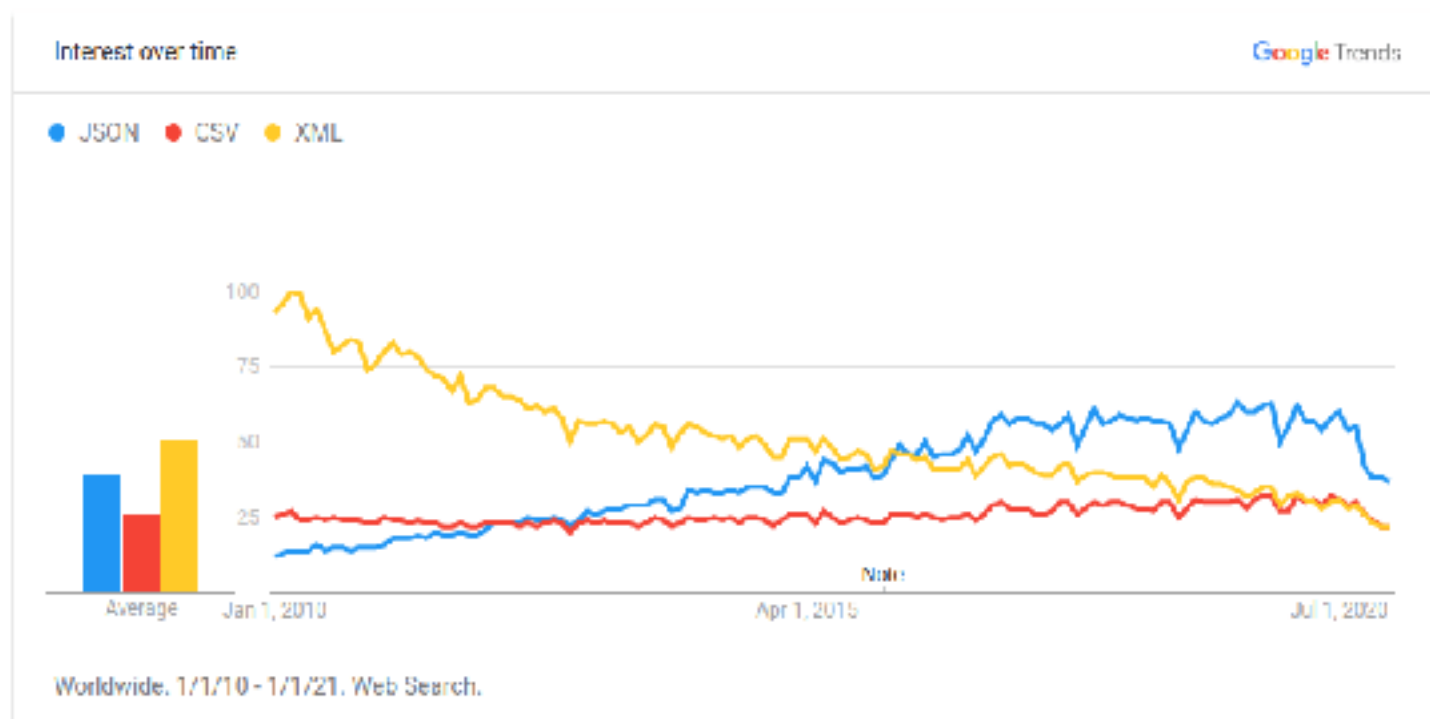


- Un requisito tecnico che facilita la multicanilità e la portabilità dei formati adottati nei contenuti
- La portabilità si riferisce alla capacità di un dispositivo, di un'applicazione software o di dati di essere facilmente spostati o trasferiti da un sistema, piattaforma o posizione all'altra senza significativi problemi di compatibilità o usabilità
 - Questo è importante tanto nella fase di fruizione del contenuto da parte dell'utente quanto nella fase di produzione del contenuto
 - Permette di ottenere un processo di produzione più resiliente e meno dipendente dai lock-in realizzati dagli attori tecnologici
 - Permette di sfruttare al massimo i processi basati sulla esternalità della rete
 - In altre parole riduce i costi di produzione e allarga la capacità di penetrazione del mercato

PORTABILITA'



UNIVERSITÀ
DEGLI STUDI
DI MILANO



UNIVERSITÀ
DEGLI STUDI
DI MILANO

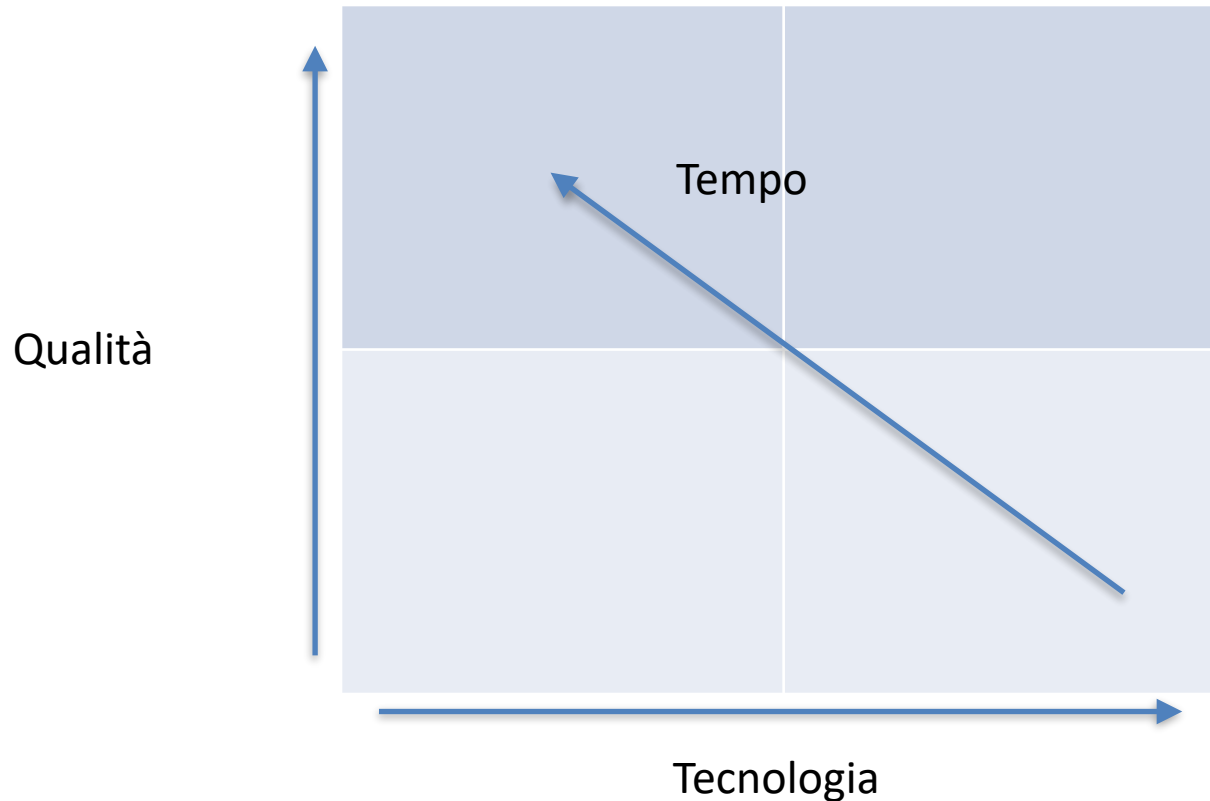
 SESAR LAB

- Per valutare la portabilità di una tecnologia bisogna considerare sia aspetti sincronici che diacronici
- **Aspetti Sincronici**
 - Standard e compatibilità. Gli standard tecnici e la compatibilità con altre tecnologie esistenti sono aspetti critici della portabilità attuale di una tecnologia. Una tecnologia che si adatta facilmente a norme e protocolli esistenti è più portatile
 - Usabilità e risorse. La facilità d'uso influenza l'adozione di una tecnologia. La quantità di risorse necessarie per far funzionare una tecnologia, come hardware, software, potenza di calcolo, connettività, ecc., è rilevante per la sua portabilità
 - Condizioni di mercato. Le condizioni di mercato, comprese le esigenze dei clienti e le tendenze del settore, possono influenzare la portabilità di una tecnologia in un dato momento
 - Normative e leggi. Le leggi e le normative locali e internazionali possono influenzare la portabilità di una tecnologia, ad esempio, per quanto riguarda la privacy dei dati o la sicurezza

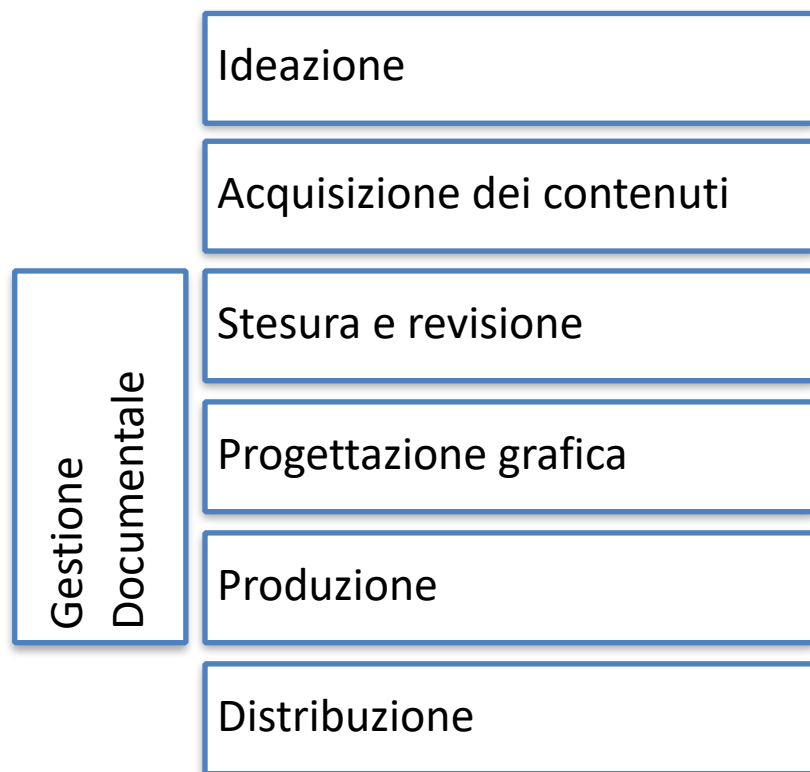
- Per valutare la portabilità di una tecnologia bisogna considerare sia aspetti sincronici che diacronici
- **Aspetti Diacronici**
 - Sviluppo tecnologico. L'evoluzione della tecnologia nel corso del tempo è un aspetto importante. Ad esempio, il passaggio da dispositivi desktop a dispositivi mobili, da rappresentazioni 2D a rappresentazioni 3D degli contenuti visuali
 - Condizioni di mercato. Capire quali attori del mercato possono trarre vantaggi dallo sviluppo di una tecnologia è fondamentale per capire quali sinergie si potranno produrre nel mercato. Ad esempio lo sviluppo dei podcast porta ad un aumento del bisogno di servizi di manipolazione di contenuti audio
 - Cambiamenti culturali e sociali. I cambiamenti nella cultura e nella società influenzano come una tecnologia è percepita e utilizzata. Ad esempio i social inizialmente percepiti come strumenti di arricchimento culturale oggi sono spesso ritenuti fonte di omologazione
 - Evoluzione delle normative. Prevedere le evoluzioni normative può essere molto importante. Ad esempio AI Act della UE potrebbe vietare l'uso di applicazioni AI in certi contesti

- Nella produzione editoriale possono esistere diversi flussi di creazione e gestione dei documenti
- Differiscono in base a
 - Flusso di lavoro
 - Qualità contenuti
 - Qualità formati, es. citazioni
 - Tecnologie
 - Creazione
 - Condivisione
 - Produzione

- Qualità richiesta e tecnologia adottata influiscono in modo inversamente proporzionale sul tempo di gestione documentale



- Il processo può variare, tradizionalmente le fasi chiave sono:



Ognuna di queste fasi può richiedere un flusso di lavoro che coinvolge competenze professionali diverse

Ad esempio i contenuti devono essere (i) di qualità, (ii) impaginati e organizzati graficamente e (iii) valutati per agli aspetti del diritto d'autore

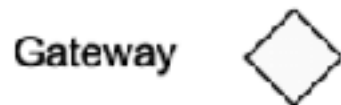
Inoltre un prodotto può essere il risultato di contenuti di diversa natura: testo, audio, grafica, AR ...

Ogni contenuto può richiedere un suo flusso di produzione in parte indipendente ed in parte coordinato con gli altri flussi di produzione

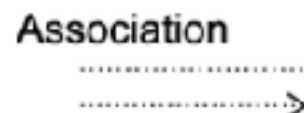
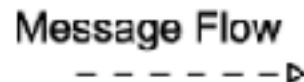
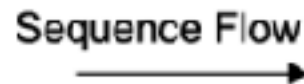


- Lo standard più diffuso per la definizione di flussi di esecuzione è BPMN 2.0 di OMG
- Ad esempio come editor si può usare <https://bpmn.io>

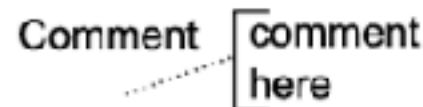
Flow Objects



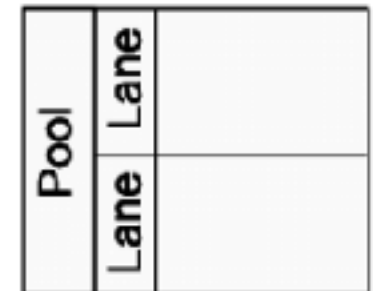
Connecting Objects



Artifacts



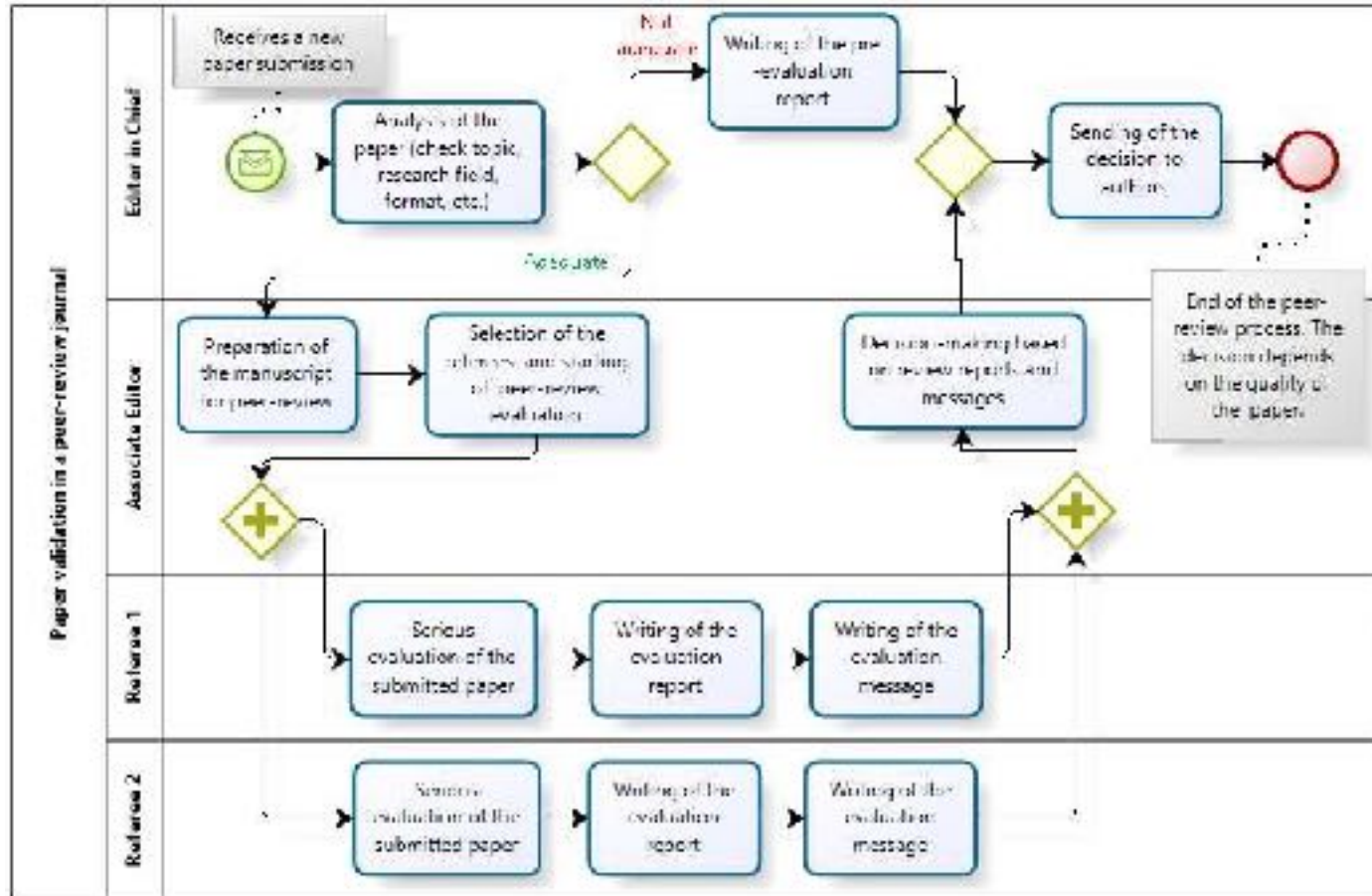
Pools and Lanes



FLUSSO PUBBLICAZIONE SCIENTIFICA



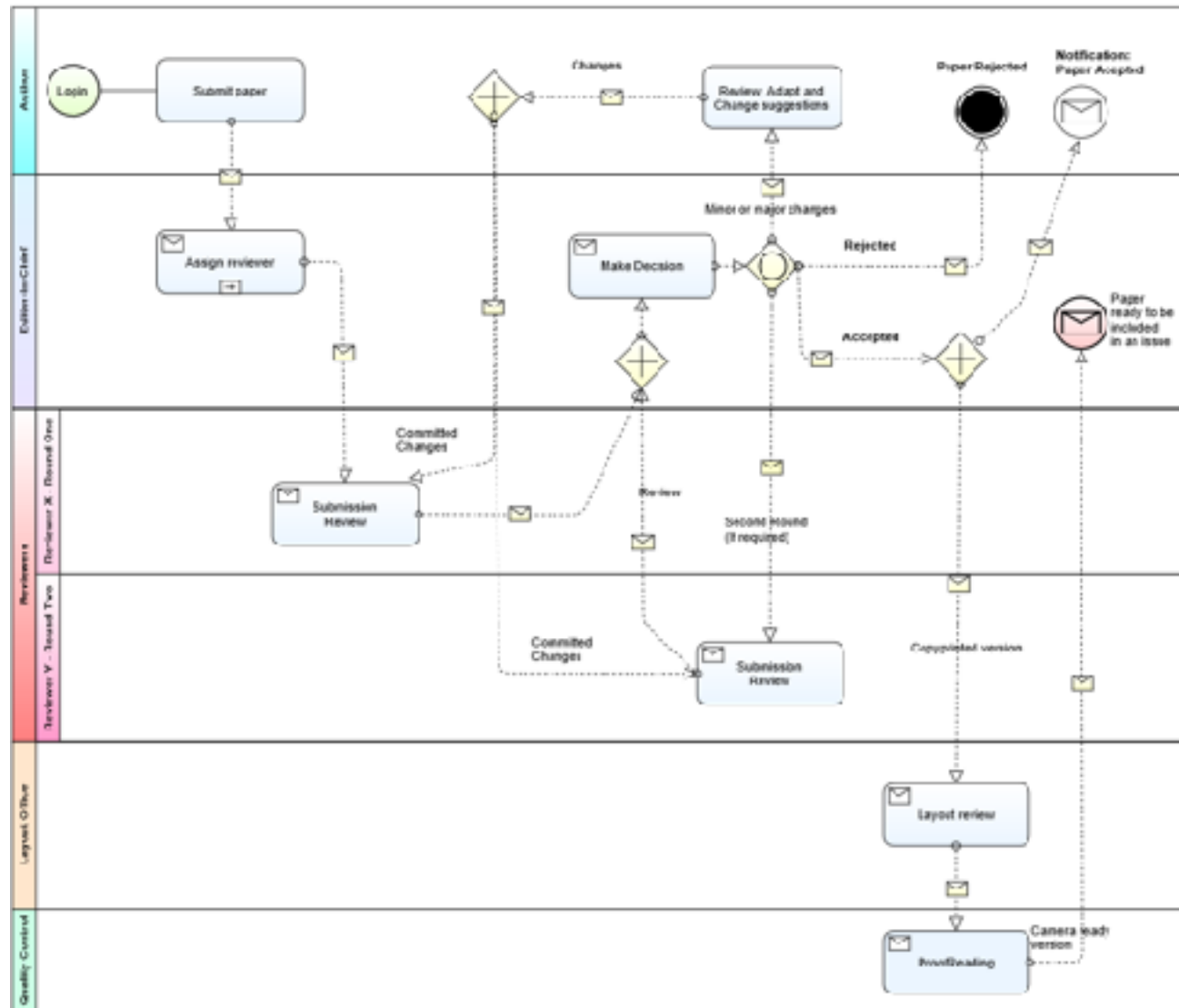
UNIVERSITÀ
DEGLI STUDI
DI MILANO



FLUSSO PUBBLICAZIONE SCIENTIFICA



UNIVERSITÀ
DEGLI STUDI
DI MILANO



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- [Mermaid](#) è una libreria JavaScript per la creazione di diagrammi e grafici basati su testo
- Consente agli sviluppatori e agli utenti di creare facilmente vari tipi di diagrammi, come diagrammi di flusso, diagrammi di sequenza, organigrammi, diagrammi di Gantt e molti altri, utilizzando una sintassi testuale, semplice e portabile

```
flowchart LR
```

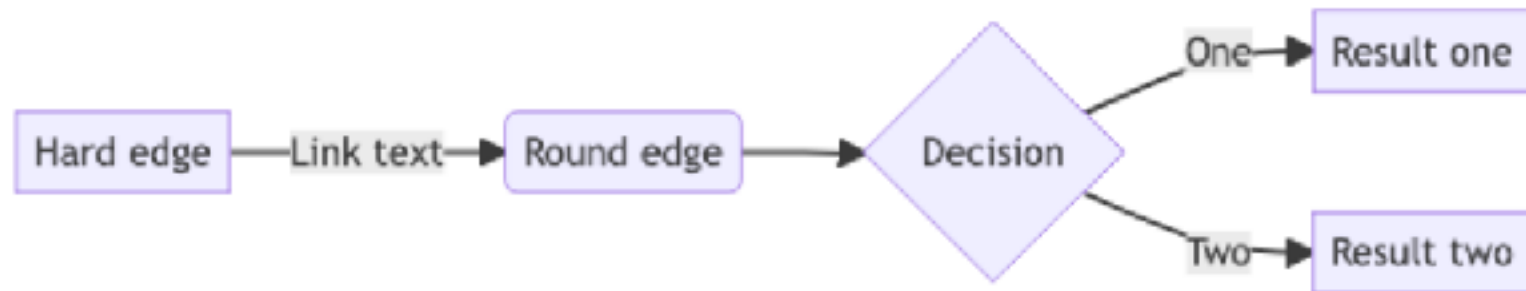
```
A[Hard edge] -->|Link text| B(Round edge)
```

```
B --> C{Decision}
```

```
C -->|One| D[Result one]
```

```
C -->|Two| E[Result two]
```


- [Mermaid](#) è una libreria JavaScript per la creazione di diagrammi e grafici basati su testo
- Consente agli sviluppatori e agli utenti di creare facilmente vari tipi di diagrammi, come diagrammi di flusso, diagrammi di sequenza, organigrammi, diagrammi di Gantt e molti altri, utilizzando una sintassi testuale, semplice e portabile



IDEAZIONE E ACQUISIZIONE DEI CONTENUTI



UNIVERSITÀ
DEGLI STUDI
DI MILANO

Studio e analisi del tema

Studio
competitor

Definizione del
target

Definizione dei
canali e licenze
di distribuzione

Originalità e
obiettivi
comunicativi

Modello di
business

Definizione dei modelli di lettura
Definizione dei formati
Definizione dell'identità visuale

Identificazione
delle fonti

Diritti e aspetti
legali

Cronoprogramma
Suddivisione dei task



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 SESAR LAB

- Per analizzare il tema è utile osservare come questo viene trattato nelle comunità di riferimento
 - Ricerca avanzata
 - https://www.google.com/advanced_search
 - <https://ahrefs.com/blog/it/operatori-ricerca-avanzata-google/>
 - <https://twitter.com/search-advanced>
 - Tool di monitoraggio
 - <https://answerthepublic.com/>
 - <https://socialbearing.com/>
 - <https://www.twitonomy.com/>
 - <https://followerwonk.com/>
 - <https://www.semrush.com/>
- In questo modo possiamo identificare i competitor, il target, le comunità di riferimento e il customer journey

DEFINIZIONE DELLE PERSONAS



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Le personas sono personaggi di fantasia, creati per rappresentare i diversi tipi di utenti che potrebbero utilizzare un prodotto
- La creazione di personas aiuta il progettista a capire le esigenze, le esperienze, i comportamenti e gli obiettivi degli utenti
- La creazione di personas aiuta a identificare le diverse esigenze degli utenti e supporta la progettazione fornendo dei punti di riferimento alle diverse fasi della progettazione
- La creazione di personas può avvenire attraverso diversi strumenti
 - Raccolta di dati di comportamento degli utenti
 - Raccolta di dati attraverso interviste e focus groups
 - Creazione sulla base della sensibilità del progettista



UNIVERSITÀ
DEGLI STUDI
DI MILANO

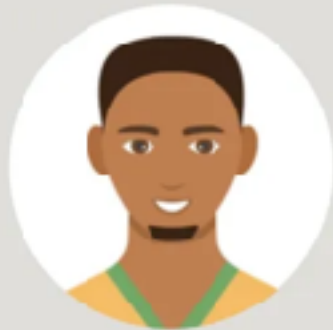
 SESAR LAB



DEFINIZIONE DELLE PERSONAS



UNIVERSITÀ
DEGLI STUDI
DI MILANO



MARK, 22

DEMOGRAPHICS

- Rents a 2 bedroom apartment in uptown Charlotte, NC with two roommates from lacrosse.
- Currently an intern for a large sports management company and a bar tender at a high-end cocktail bar on the weekends.
- Recently single and using dating apps. An only child and rescued a puppy 4 months ago.
- Graduated with a B in Health Sciences from UNCW. Gross \$28,000 in private loans.

BEHAVIORAL IDENTIFIERS

- Hobbies include eating out, lifting weights, playing club lacrosse, drinking draft beer tailgating for UNC, and watching live music.
- Prefers chat systems for customer service and support. Values quick responses and 24-hour availability.
- Wants fast shipping. Values a low price. Prefers online marketing, light colors, and simple websites.
- Shops online between 9pm-11pm. Buys during pay periods at the end of the month and often uses promo codes.
- Owns an Android and shops on mobile with referrals from social media such as Instagram and Facebook.
- Studied abroad in Chile. Can speak Spanish and some Italian. Travels on a budget and prefers hostels to hotels.
- Personal goals to be out of debt by 2024, to be engaged in 4 years and to work in a university sports medicine department.
- Very fit and active but has a bad diet due to eating on a budget and drinking often. Seeks out healthier information daily.



JUSTIN, 31

DEMOGRAPHICS

- Rents a 1 bedroom apartment in uptown Atlanta with his partner. Looking to buy a home next year.
- Works 40 hours a week in a 5-star hotel as a superintendent with a net income of \$45,000.
- Recently engaged to his boyfriend of 3 years with plans to marry in 2024. Wants a son.
- Graduated from a community college with honors and held a job in a university library.

BEHAVIORAL IDENTIFIERS

- Hobbies include organ cooking, yoga, monthly massages, playing the drums, volunteering, drinking cocktails and watching local plays.
- Prefers to call for customer service and support. Values follow up summary emails and easy to read directions.
- Wants free shipping on large orders. Values quality and price. Extremely brand loyal. Prefers email marketing.
- Shops online between 9am-1pm and makes multiple returns. Uses PayPal at checkout.
- Owns an iPhone and tablet. Shops on this tablet with referrals from email marketing and promotions.
- Travels in the USA twice a year to visit family in Maine and California. Prefers long road trips and holiday travel.
- Personal goals to become a general manager within 4 years and to purchase a suburban home after his wedding and honeymoon.
- Cooks very healthy and with local products. Has great self-care and is very active.



MAGGIE, 52

DEMOGRAPHICS

- Owns a 4 bedroom home in coastal NC and 2 high-end cars. Lives in a suburban gated neighborhood.
- Human Resource Director of a small media company for 5 years with a \$48,000 salary.
- Happily married for 12 years with 2 sons and a house hold income of above \$220,000.
- Well-educated with an HR degree from a university. Graduated without debt.

BEHAVIORAL IDENTIFIERS

- Hobbies include reading, playing tennis, networking at the country club being involved in youth groups and visiting the beach.
- Strong need for customer service and support. Values face-to-face communication and rewards systems.
- Wants a simple product. Values quality and brand image. Prefers print marketing and bold graphics.
- Shops in department stores often. Buys in bulk on sale items. Believed by a previous client.
- Owns an iPhone and makes purchases online with a desktop Mac computer. Needs a simple check-out system.
- Well-traveled in Europe. Prefers traveling in off-peak times in chain hotels and loves pre-planned travel-like cruises.
- Personal goals to retire by age 65, to own a vacation home on the Spanish Riviera and for her sons to attend an Ivy League university.
- In very good health but very busy and often stressed. Visits group exercise programs.



DEGLI STUDI
DI MILANO

SESAR LAB

DEFINIZIONE DELLE PERSONAS



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Lo scopo di lavorare con le personas è quello di poter sviluppare soluzioni, prodotti e servizi basati sulle esigenze e sugli obiettivi degli utenti
- Andrebbero descritti includendo dettagli sull'istruzione, lo stile di vita, gli interessi, i valori, gli obiettivi, i bisogni, i limiti, i desideri, gli atteggiamenti e i modelli di comportamento dell'utente.
- Aggiungete alcuni dettagli personali di fantasia per rendere la persona un personaggio realistico
- Date un nome a ciascuna delle vostre personas
- Create 1-2 pagine di descrizioni per ogni persona



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 SESAR LAB



DEFINIZIONE DELLE PERSONAS



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Ad esempio si può provare a rispondere a queste domande
 - Qual è l'età, l'educazione, la professione, l'estrazione sociale del mio utente ideale? Quale la sua cultura di riferimento?
 - Quali sono i suoi valori, quali i suoi obiettivi a medio termine?
 - Dove si informa? Come sceglie i prodotti o i servizi?
 - Che aspetto ha la sua giornata tipo? Quando e dove consuma questi prodotti e servizi?
 - Cosa apprezza di più di un servizio? Cosa lo può allontanare?
- Esistono servizi che aiutano nella creazione di personas
 - [SemRush Persona](#)
 - [Open Design Kit](#)



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 SESAR LAB



- Prepare situazioni o scenari per le personas
- La definizione di uno scenario può aiutare meglio la comprensione delle soluzioni
- A questo scopo, dovrete descrivere una serie di situazioni specifiche che potrebbero innescare l'uso del prodotto o del servizio che state progettando
- Gli scenari di solito iniziano collocando la persona in un contesto specifico con un problema che vuole o deve risolvere

- Esempi di fonti aperte da cui puoi accedere a libri e manuali,
 - ideali per esercitazioni, progetti educativi o per creare contenuti basati su documenti disponibili sotto licenze aperte
- **Progetti Wikimedia**
 - [Wikibooks](#): Una biblioteca digitale di manuali e libri di testo liberamente modificabili. Copre argomenti che spaziano dalla programmazione alle scienze umane. Ad esempio:
 - [Wikisource](#): Raccolta di testi di pubblico dominio, incluse opere classiche e manuali storici.
 - Licenza: Generalmente CC BY-SA o dominio pubblico.
- **OpenStax**
 - [OpenStax](#): Piattaforma che offre libri di testo gratuiti di alta qualità per corsi universitari e liceali.
 - Licenza: CC BY (uso, adattamento e distribuzione permessi con attribuzione).

- Esempi di fonti aperte da cui puoi accedere a libri e manuali,
 - ideali per esercitazioni, progetti educativi o per creare contenuti basati su documenti disponibili sotto licenze aperte
- **MIT OpenCourseWare**
 - [MIT OpenCourseWare](https://ocw.mit.edu/): Materiali educativi gratuiti dai corsi del Massachusetts Institute of Technology.
 - Licenza: CC BY-NC-SA.
- **Project Gutenberg**
 - [Project Gutenberg](https://www.gutenberg.org/): Biblioteca digitale con oltre 60.000 libri gratuiti, inclusi manuali tecnici e opere letterarie.
 - Licenza: Principalmente dominio pubblico.

- Esempi di fonti aperte da cui puoi accedere a libri e manuali,
 - ideali per esercitazioni, progetti educativi o per creare contenuti basati su documenti disponibili sotto licenze aperte
- **LibreTexts**
 - [LibreTexts](#): Risorsa collaborativa che offre libri di testo per scienze, ingegneria, matematica e scienze sociali.
 - Licenza: Generalmente CC BY o CC BY-NC.
- **Open Textbook Library**
 - [Open Textbook Library](#): Raccolta di libri di testo accademici gratuiti, revisionati e utilizzati in contesti universitari.
 - Licenza: CC BY o altre licenze aperte.

- Esempi di fonti aperte da cui puoi accedere a libri e manuali,
 - ideali per esercitazioni, progetti educativi o per creare contenuti basati su documenti disponibili sotto licenze aperte
- **OpenEdition**
 - [OpenEdition Books](#): Piattaforma che offre libri accademici open access, inclusi molti in italiano, su argomenti come scienze sociali, storia, e arte.
- **LiberLiber**
 - [LiberLiber](#): Biblioteca digitale che offre una vasta collezione di libri, saggi e manuali in lingua italiana.
 - Licenza: Principalmente dominio pubblico.

- Esempi di fonti aperte da cui puoi accedere a libri e manuali,
 - ideali per esercitazioni, progetti educativi o per creare contenuti basati su documenti disponibili sotto licenze aperte
- **Book in Progress**
 - [Book in Progress](#): Progetto italiano per la creazione di libri scolastici digitali e collaborativi, scritti da docenti e utilizzabili liberamente nelle scuole.
 - Licenza: Spesso open access per scopi educativi.
- **UniOpen**
 - [Milano University Press](#): Casa editrice digitale e ad accesso aperto dell'Università degli Studi di Milano, nata per promuovere e diffondere i risultati della ricerca scientifica.
 - Licenza: Creative Commons e Open Access

DEFINIZIONE DEL MODELLO DI LETTURA



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Ci sono diversi modelli di lettura che possono influenzare la percezione dell'utente e la fruibilità del contenuto di un testo
 - **Lettura sequenziale.** È il modello di lettura più tradizionale, in cui gli utenti seguono un flusso lineare di contenuti da una pagina o schermata all'altra. Questo modello è tipico per i libri stampati e molti ebook digitali. È adatto soprattutto alla narrativa o alla saggistica divulgativa
 - **Lettura non lineare.** In questo modello, gli utenti possono accedere ai contenuti in modo non sequenziale. Questo si realizza attraverso strumenti di ipertestualità, collegamenti interni o menu di navigazione che consentono agli utenti di saltare direttamente a sezioni specifiche del contenuto. Questo modello è comune in documenti tecnici, manuali online e contenuti educativi interattivi
 - **Lettura visuale.** Questo modello pone un'enfasi particolare sull'aspetto visuale del contenuto. Può coinvolgere l'uso di immagini, grafici, diagrammi e layout visivamente accattivanti per migliorare l'esperienza di lettura. Questo è spesso importante per le riviste, i fumetti, i libri illustrati e i materiali didattici che richiedono una componente visiva significativa
 - **Lettura testuale.** Al contrario, in questo modello, l'attenzione si concentra principalmente sul testo, eliminando altri elementi per ottenere maggiore semplicità e chiarezza. Gli ebook tradizionali e molti documenti di testo online seguono questo modello. La documentazione di una libreria, una API o un software è forse un intermedio tra visuale e testuale, si privilegia la semplicità ma è importante formattare il codice in modo da semplificare la lettura visiva, alcuni diagrammi possono aiutare il lettore a orientarsi



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 SESAR LAB

DEFINIZIONE DEL MODELLO DI LETTURA



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Ci sono diversi modelli di lettura che possono influenzare la percezione dell'utente e la fruibilità del contenuto
- **Lettura collaborativa.** Alcune piattaforme offrono modelli di lettura che incorporano elementi collaborativi. Ad esempio, la possibilità di condividere annotazioni, commenti o discussioni direttamente nel contesto del testo. Questo modello può promuovere l'interazione tra gli utenti e arricchire l'esperienza di lettura
- **Lettura interattiva.** Questi modelli spesso integrano elementi interattivi per coinvolgere gli utenti e migliorare l'esperienza di apprendimento. Piattaforme di apprendimento online, ebook interattivi e risorse educative digitali spesso adottano questi modelli di lettura interattivi per migliorare l'engagement e facilitare l'apprendimento attivo. L'obiettivo è offrire un'esperienza di apprendimento più coinvolgente e personalizzata rispetto alla tradizionale lettura sequenziale
- **Lettura adattiva.** Questo modello coinvolge l'adattamento del contenuto in base al comportamento o alle preferenze dell'utente. Ad esempio, la personalizzazione della formattazione del testo, la dimensione del carattere o la modalità di visualizzazione notturna in base alle preferenze dell'utente
- **Lettura ludica.** Esistono esperienze di lettura che si intersecano con il gioco. Questi libri offrono ai lettori una narrazione interattiva in cui possono prendere decisioni chiave che influenzano il corso della storia. Gli elementi di gioco, come le scelte multiple, i punteggi e i rami narrativi, sono incorporati nel tessuto stesso del libro. Oltre ai tradizionali librogame, i giochi di enigmi e rompicapi oggi possono esistere strumenti basati sulla realtà aumentata o la realtà virtuale



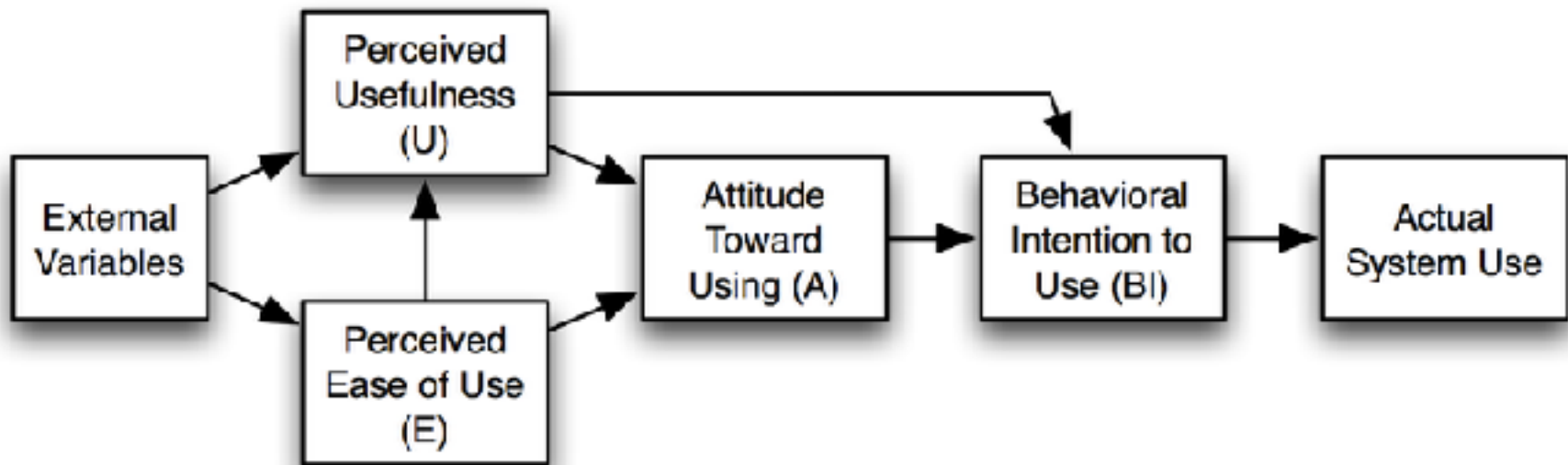
UNIVERSITÀ
DEGLI STUDI
DI MILANO

 SESAR LAB



- Un modello di lettura è anche un modello di fruizione
- Nel valutarlo, è fondamentale considerare il rapporto tra le *funzionalità* offerte e i *costi di fruizione*
 - **Funzionalità.** Il modello consente di accedere rapidamente alle informazioni?
 - **Usabilità.** I costi per gli utenti (tempo, denaro, strumenti) sono proporzionati ai benefici offerti?
 - **Adattabilità.** Le funzionalità si adattano a diversi contesti d'uso o alle capacità degli utenti senza richiedere risorse aggiuntive?
 - **Aggiornabilità.** Quali sono i costi per mantenere il contenuto aggiornato e il servizio funzionante nel tempo?

- In generale può essere utile far riferimento a **modelli di accettazione della tecnologia** che individuano due variabili fondamentali per valutare il grado di accettazione di una tecnologia: *utilità percepita* (guadagno economico o emotivo) e *facilità d'uso percepita* (riduzione dello sforzo cognitivo)



Davis, F. D. (1989), "Perceived usefulness, perceived ease of use, and user acceptance of information technology", *MIS Quarterly*, **13** (3): 319–340, [doi:10.2307/249008](https://doi.org/10.2307/249008), [JSTOR 249008](https://www.jstor.org/stable/249008)

FORMATI DI DESTINAZIONE



UNIVERSITÀ
DEGLI STUDI
DI MILANO

Caratteristica	Libro (cartaceo)	eBook
Formato del contenuto	Testo statico	Testo statico o interattivo
Accessibilità	Supporto fisico	Store online
Leggibilità	Molto alta	Alta (dispositivi e-ink)
Multimedialità	Limitata (immagini)	Limitata (audio, immagini)
Interattività	Limitata (annotazioni a mano)	Moderata (note, segnalibri)
Adattabilità	Fisicamente limitata	Moderata (font, layout)
Aggiornabilità	Moderata (nuove edizioni)	Moderata (aggiornamenti pull)



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 SESAR LAB

FORMATI DI DESTINAZIONE



UNIVERSITÀ
DEGLI STUDI
DI MILANO

Caratteristica	Notebook (tecnico)	Webbook
Formato del contenuto	Codice, testo e grafici	Testo + media ipertestuali
Accessibilità	Alta (WWW)	Alta (WWW)
Leggibilità	Bassa (dipende dallo schermo)	Bassa (dipende dallo schermo)
Multimedialità	Limitata (grafici)	Elevata (audio, video, link)
Interattività	Alta (eseguibile in tempo reale)	Alta (ipertesto, video, script)
Adattabilità	Richiede competenze tecniche	Molto alta (web-based)
Aggiornabilità	Continua (su repository)	Continua (server web)



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 SESAR LAB

FORMATI DI DESTINAZIONE



UNIVERSITÀ
DEGLI STUDI
DI MILANO

Caratteristica	Libro aumentato	Libro Web3
Formato del contenuto	Testo + AR/VR	Testo + media decentralizzati
Accessibilità	Fisico e digitale	Alta (WWW)
Leggibilità	Bassa (dipende dallo schermo)	Bassa (dipende dallo schermo)
Multimedialità	Elevata (media e AR)	Dipende dalla piattaforma
Interattività	Molto alta (con device AR)	Decentrata (tokenizzazione)
Adattabilità	Moderata (AR richiede device)	Moderata (con wallet Web3)
Aggiornabilità	Dinamica (con server based AR)	Continua (smart contract)



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 SESAR LAB

Stesura bozza

Punti da trattare

Sviluppo dei
contenuti

Collegamento
con le fonti

Identificazioni
di elementi
multimediali

Revisione dei contenuti

Assegnazione
delle revisioni

Raccolta

Implementazione
e delle revisioni

Inclusione nuovi
contenuti

Definizione dello stile grafico

Regole di
Impaginazione

Applicazione dello
stile grafico

Definizione di
elementi grafici
avanzati

Verifica dei requisiti
di accessibilità

Creazione del formato di distribuzione

Compilazione
del documento

Revisione
manuale del
documento

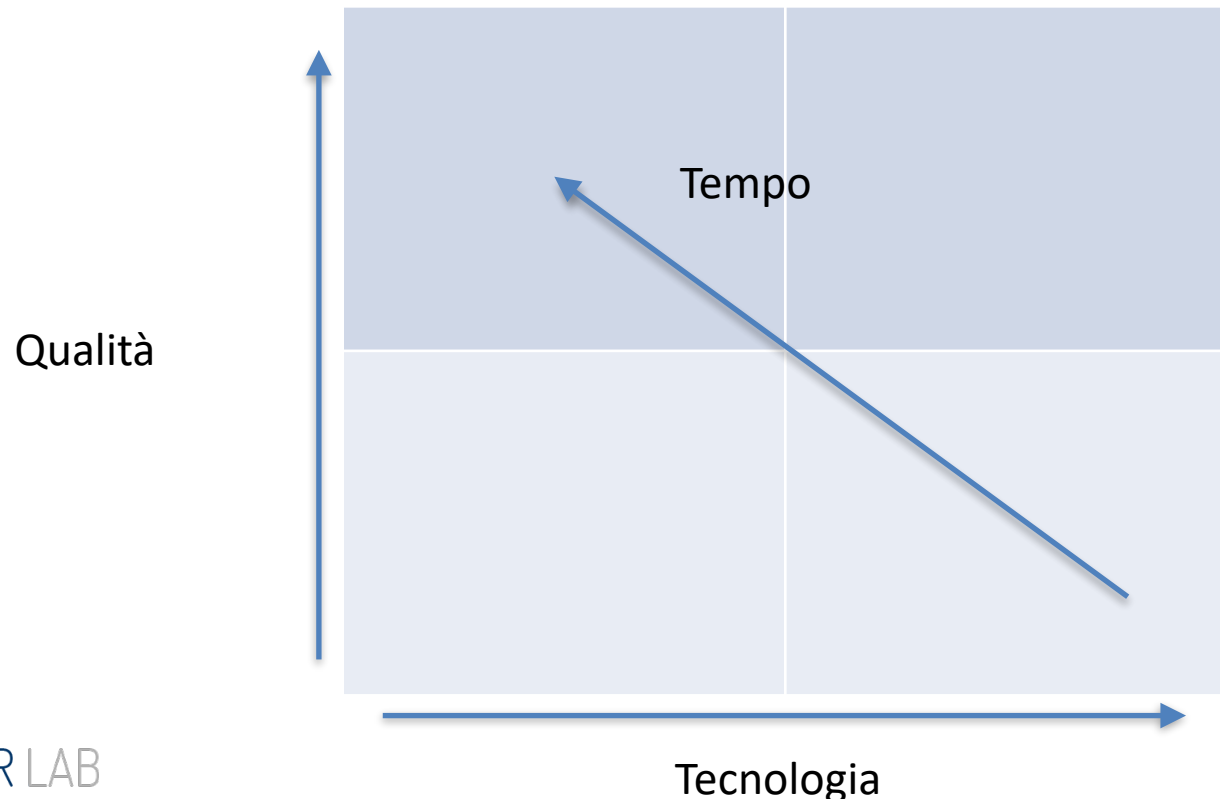
Definizione dei
metadati
descrittivi

Definizione di
documentazio
ne e materiale
di supporto

Dichiarazi
one delle
licenze

- Ogni processo di produzione editoriale richiede di definire un relativo processo di **gestione documentale**
- La creazione del contenuto può richiedere **diverse fasi**
 - Promuovere e verificare la qualità
 - Consentire l'aggiornamento
 - Coinvolgere diverse competenze e professionalità
- Un sistema di gestione documentale più complesso porta ad un aumento della qualità ma anche dei costi
 - La **tecnologia** può abbassare questi costi

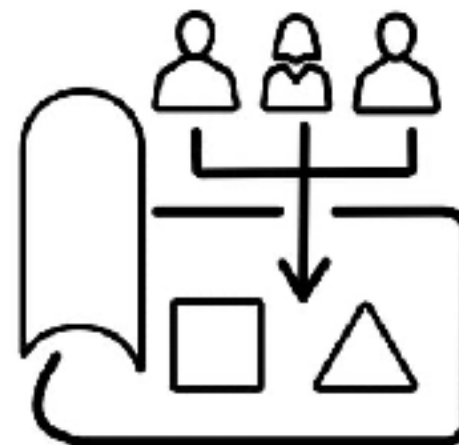
- Un sistema di gestione documentale più complesso porta ad un aumento della **qualità** ma anche dei **costi**
- La **tecnologia** può abbassare questi costi



- Tutti i progetti richiedono una gestione documentale
 - Alcuni soprattutto in fase di creazione e revisione del contenuto
 - Es. Romanzo
 - Altri durante tutto il ciclo di vita del prodotto
 - Es. Documentazione Software
- È importante definire **quale tipo di gestione documentale** sarà richiesta nel progetto

- **Gestire efficacemente i documenti**
 - Utilizzare formati sorgente portabili
 - Definire la struttura prima dello stile
 - Non dipendere da software proprietari
 - Valutare i formati di destinazione e il processo di trasformazione
- **Gestire il versioning**
 - Contenuto: testo, audio, grafica, video, AR
 - Tipografica, stile e identità visuale
 - Bibliografia, metadati, catalogazione
 - Diritti d'autore

- Definire la strategia di collaborazione in base a
 - Complessità del contenuto (specializzazione, multimedialità)
 - Interdipendenza tra i contenuti
 - Frequenza di aggiornamento
 - Numerosità del team



Project

STRATEGIA DI COLLABORAZIONE



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Le strategie di collaborazione si distinguono per diversi aspetti
- **Strutturazione dei contenuti**
 - *Editing sequenziale*
 - Ogni collaboratore lavora in momenti diversi, seguendo un ordine predefinito
 - *Editing a blocchi*
 - Il documento è suddiviso in sezioni o compiti assegnati a diversi collaboratori
- **Frequenza delle iterazioni**
 - *Fasi di iterazione definite*
 - Il contenuto viene rivisto un numero di volte definito a priori
 - *Fasi di iterazione indefinite*
 - Il contenuto viene rivisto ciclicamente fino a raggiungere il consenso del team
- **Livello di sincronia**
 - *Asincrono*
 - In uno stesso momento un'unica persona lavora al contenuto
 - *Sincrono in lettura*
 - Chi non lavora sul contenuto può vedere le sue evoluzioni
 - *Sincrono in scrittura*
 - Più persone lavorano contemporaneamente sullo stesso documento



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 **SESAR LAB**



- È importante definire i **ruoli** e la **numerosità** dei diversi ruoli
- Più persone lavorano al progetto più importante sarà possedere uno strumento di gestione del processo collaborativo
- **Autore**
 - Creare il contenuto e fornire informazioni accurate e complete
 - Collaborare con revisori ed editor per eventuali modifiche
- **Revisore**
 - Analizzare il contenuto per verificarne la correttezza, coerenza e qualità
 - Garantire l'aderenza alle linee guida e agli standard stilistici
- **Redattore (Editor)**
 - Coordinare l'intero processo di creazione e gestione del documento, assegnando ruoli e definire scadenze
 - Assicurarsi che il documento sia conforme alla linea editoriale

- È importante definire i **ruoli** e la **numerosità** dei diversi ruoli
- Più persone lavorano al progetto più importante sarà possedere uno strumento di editing collaborativo
- **Specialista di contenuti multimediali**
 - Aggiungere elementi visivi come immagini, grafici, video o infografiche, AR
 - Garantire la coerenza visiva con il contenuto testuale
- **Amministratore della piattaforma**
 - Configurare e mantenere gli strumenti di gestione documentale (es. Google Workspace, GitHub, o piattaforme di gestione contenuti).
 - Gestire i permessi di accesso e le versioni del documento.
 - Risolvere problemi tecnici legati alla piattaforma.
- **Stakeholder/Committente finale**
 - Fornire l'approvazione finale o richiedere ulteriori modifiche
 - Garantire che il documento soddisfi gli obiettivi strategici o aziendali

Esempio: editoria tradizionale

- Autore: scrive una prima bozza di un testo
- Revisore: verifica l'accuratezza linguistica e propone miglioramenti
- Redattore: rivede il documento per la leggibilità e lo stile
- Specialista di contenuti multimediali: aggiunge immagini e grafica (spesso solo la copertina)
- Stakeholder (Editore): approvazione finale per la distribuzione attraverso i canali aziendali

Esempio: creazione di un rapporto aziendale multimediale

- Raccolta e analisi dei dati: analista dati, autore testi
- Scrittura e revisione contenuti: autore testi, revisore
- Integrazione multimediale: specialista multimediale, designer grafico
- Revisione collaborativa: team aziendale (diverse divisioni), project manager
- Distribuzione: amministratore della piattaforma, marketing

Esempio: sviluppo di documentazione per software open source

- Struttura del documento: project manager
- Scrittura dei contenuti: contributor (molteplici)
- Revisione tecnica e collaborativa: revisori (community open source), project manager
- Integrazione multimediale e pubblicazione: designer grafico, maintainer del progetto
 - Aggiunta di diagrammi, esempi di codice interattivi
- Pubblicazione su piattaforme come GitHub Pages o ReadTheDocs: amministratore della piattaforma, project manager
 - Documentazione pubblicata e aggiornata regolarmente

STRATEGIA DI COLLABORAZIONE



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Date le caratteristiche del contenuto, la frequenza di aggiornamento e la numerosità del gruppo di lavoro definire una strategia di collaborazione
- La tabella riporta i valori massimi supportati

Strategia	Complessità del contenuto	Interdipendenza dei contenuti	Frequenza di aggiornamento	Numerosità del team
Editing Sequenziale	-	Alta	Bassa	Bassa
Editing a blocchi	-	Bassa	Alta	Alta
Iterazioni fisse	Bassa	Bassa	Alta	Alta
Iterazioni cont.	Alta	Alta	Bassa	Bassa
Asincrona	Bassa	Bassa	Bassa	Bassa
Sincrona in lettura	Media	Alta	Media	Alta
Sincrona in scrittura	Alta	Media	Alta	Media



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 SESAR LAB

STRATEGIA DI COLLABORAZIONE



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Ci troviamo di fronte ad una classica situazione di requisiti in conflitto
- Es. Rapporto aziendale: Alta | Alta | Media | Alta
 - Scegliamo editing sequenziale o a blocchi? interazioni basse o elevate?

Strategia	Complessità del contenuto	Interdipendenza dei contenuti	Frequenza di aggiornamento	Numerosità del team
Editing Sequenziale	-	Alta	Bassa	Bassa
Editing a blocchi	-	Bassa	Alta	Alta
Iterazioni fisse	Bassa	Bassa	Alta	Alta
Iterazioni cont.	Alta	Alta	Bassa	Bassa
Asincrona	Bassa	Bassa	Bassa	Bassa
Sincrona in lettura	Media	Alta	Media	Alta
Sincrona in scrittura	Alta	Media	Alta	Media



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 SESAR LAB

STRATEGIA DI COLLABORAZIONE



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Ci troviamo di fronte ad una classica situazione di requisiti in conflitto
- Es. Editoria tradizionale: Bassa | Basa | Bassa | Bassa
 - Scegliamo editing sequenziale o a blocchi? interazioni basse o elevate?

Strategia	Complessità del contenuto	Interdipendenza dei contenuti	Frequenza di aggiornamento	Numerosità del team
Editing Sequenziale	-	Alta	Bassa	Bassa
Editing a blocchi	-	Bassa	Alta	Alta
Iterazioni fisse	Bassa	Bassa	Alta	Alta
Iterazioni cont.	Alta	Alta	Bassa	Bassa
Asincrona	Bassa	Bassa	Bassa	Bassa
Sincrona in lettura	Media	Alta	Media	Alta
Sincrona in scrittura	Alta	Media	Alta	Media



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 SESAR LAB

- I requisiti in conflitto (*conflicting requirements*) si verificano quando due o più **requisiti** di un sistema, progetto o processo **si contraddicono** o rendono difficile soddisfarli contemporaneamente
- *Costo vs. Qualità*
- Creare un prodotto di alta qualità richiede investimenti in materiali, risorse e tempo, ma rispettare un budget ridotto può limitare queste possibilità
- *Tempi di Consegna vs. Funzionalità*
- Rispettare scadenze strette potrebbe richiedere la riduzione delle funzionalità, ma i requisiti del cliente potrebbero includere un set completo di caratteristiche
- Quando un requisito **pone un vincolo** ci si può spostare verso soluzioni che ammettono valori più **alti** ma non più **bassi**

- Strategie per **gestire requisiti in conflitto**
- **Prioritizzazione**
 - Fare un bilancio di costi e benefici per ogni requisito
 - Identificare quali requisiti sono più critici per il successo del progetto e dare loro priorità
- **Compromesso**
 - Cercare un equilibrio tra requisiti contrastanti, accettando compromessi ragionevoli (es. aumentare i tempi per migliorare la qualità dei contenuti)
- **Innovazione tecnologica**
 - Trovare una nuova soluzione tecnologica che elimina il conflitto

- Tradizionalmente, i CMS "monolitici" (es. WordPress o Joomla) presentavano un **conflitto** tra:
 - **Flessibilità nella personalizzazione** del design e funzionalità
 - Gli *sviluppatori* desideravano maggiore controllo sul layout e sull'integrazione con altri sistemi.
 - **Accessibilità e facilità d'uso** per gli editor di contenuti non tecnici
 - Gli *editor* volevano un'interfaccia semplice per creare e modificare contenuti, senza doversi preoccupare delle complessità tecniche.
- Questo conflitto creava compromessi difficili:
 - Più flessibilità per gli sviluppatori implicava interfacce più complesse per gli editor.
 - Più semplicità per gli editor limitava le possibilità di personalizzazione per gli sviluppatori.

- Un headless CMS (Strapi, sanity.io) separa il backend (dove i contenuti sono creati e gestiti) dal frontend (dove i contenuti sono presentati agli utenti). Questa architettura consente di **risolvere il conflitto**
- Flessibilità per gli sviluppatori
 - Possono creare frontend personalizzati utilizzando tecnologie moderne (es. React, Vue.js) e integrare il CMS con altri strumenti tramite API.
 - È possibile utilizzare lo stesso contenuto per molteplici canali (web, mobile, app, dispositivi IoT) senza modificare il backend.
- Semplicità per gli editor
 - Gli editor lavorano su un'interfaccia backend intuitiva, indipendente dal design o dalla logica del frontend. Possono concentrarsi esclusivamente sulla creazione e l'organizzazione del contenuto.

- Una delle soluzioni tecnologiche che ha permesso di conciliare i conflitti legati alla gestione documentale è rappresentata dalle piattaforme **editing collaborativo** che si sono diffuse nell'ultimo decennio
- Queste piattaforme hanno permesso di integrare
 - Editing sequenziale e a blocco > *gestione dei permessi e delle versioni*
 - Iterazione fissa e continua > *revisioni continue ma soggette ad approvazione*
 - Strumenti di sincronia in lettura e in scrittura > *commenti e editing del contenuto*

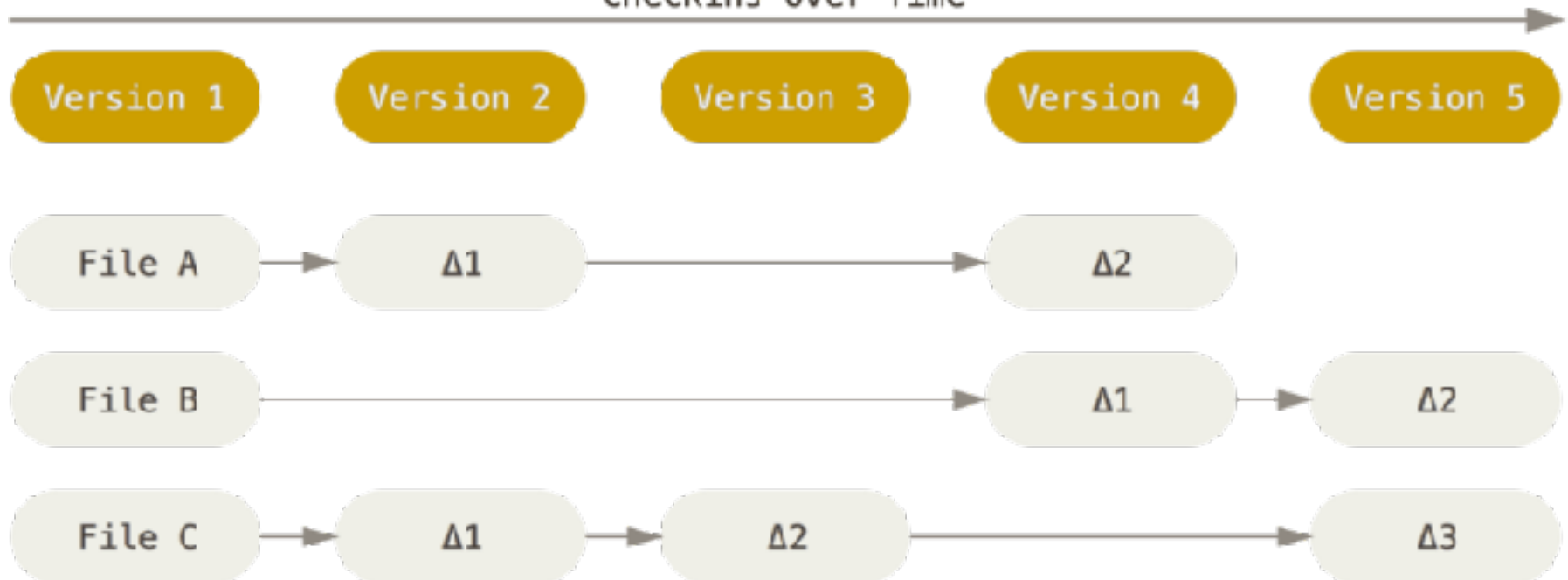
- La piattaforma che ha meglio interpretato le funzioni di gestione del **versioning di documenti** è GitHub
- Lo scopo principale di GitHub.com è quello di facilitare il controllo di versione e il tracciamento di issue nello sviluppo del software
- Per il controllo delle versioni consente di effettuare richieste di pull per proporre modifiche al codice sorgente. Gli utenti che hanno la possibilità di rivedere le modifiche proposte possono vedere una diff delle modifiche richieste e approvarle
- Nella terminologia di Git, l'azione di registrazione delle modifiche apportate a un file è chiamata **commit**, conferma, e una sua istanza è un commit. La cronologia di tutti i commit viene conservata e può essere visualizzata in un secondo momento
- Un manuale completo è [Pro Git Book](#)

GESTIONE DEL VERSIONING



UNIVERSITÀ
DEGLI STUDI
DI MILANO

Checkins Over Time

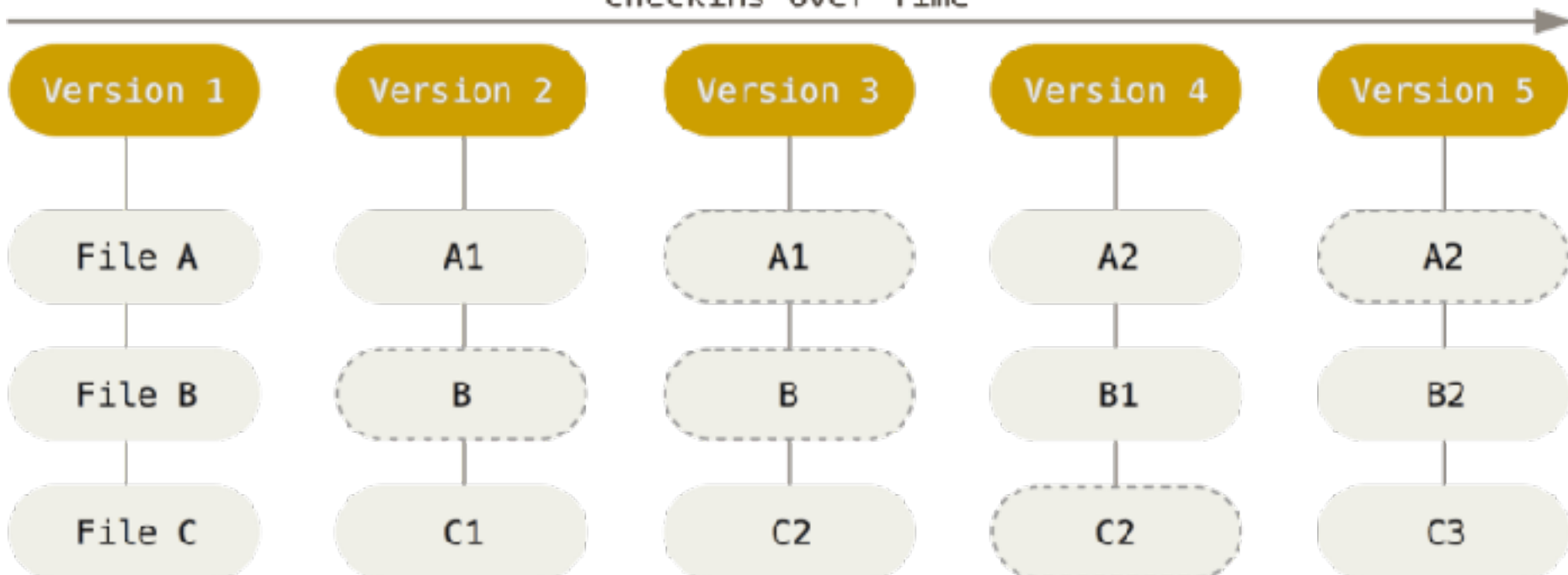


GESTIONE DEL VERSIONING



UNIVERSITÀ
DEGLI STUDI
DI MILANO

Checkins Over Time



UNIVERSITÀ
DEGLI STUDI
DI MILANO

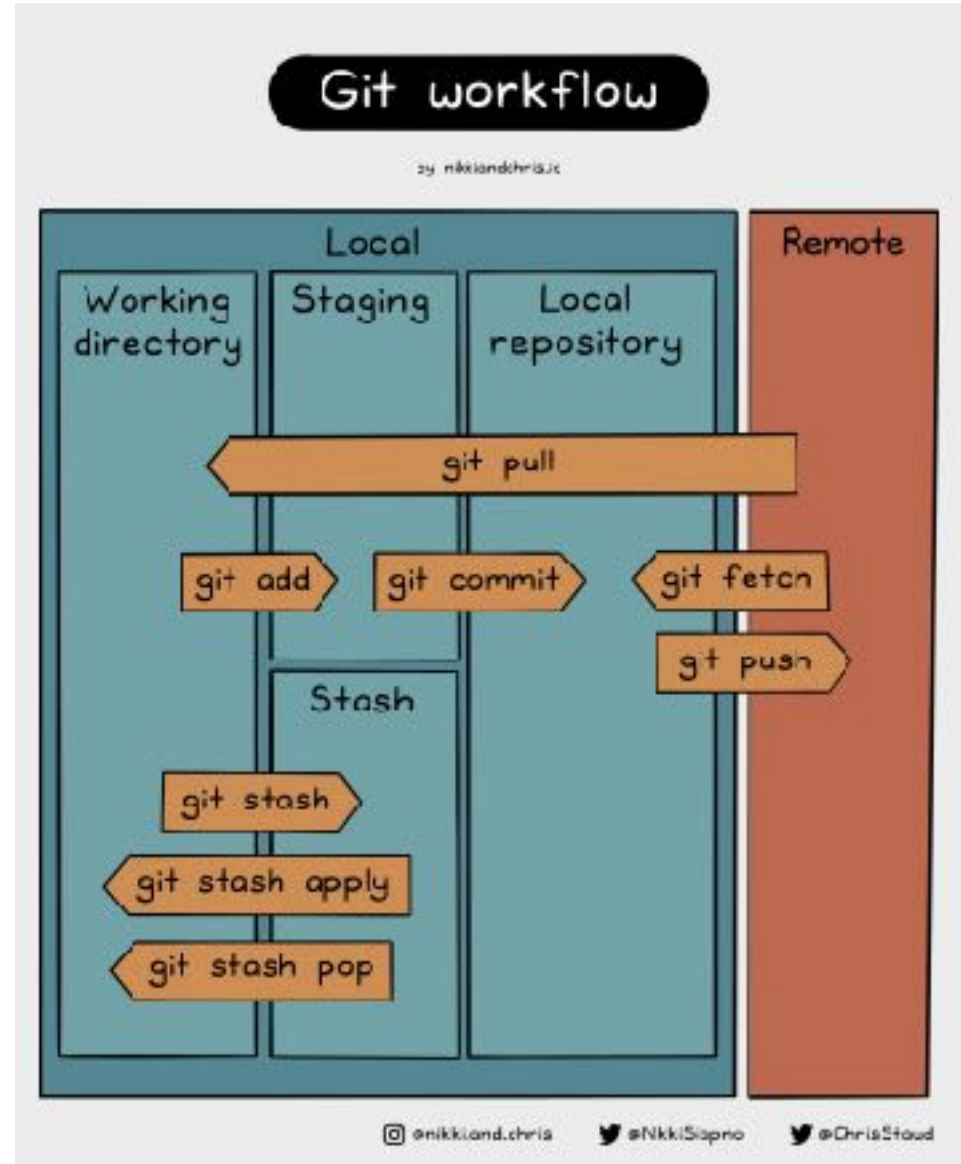
 SESAR LAB

GESTIRE GLI AGGIORNAMENTI



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- **Working directory** o *workspace*: spazio di lavoro locale
- **Staging** o *directory di cache*: spazio che contiene i file pronti per essere committati
- **Stash**: spazio in cui salvare le modifiche mentre si lavora ad altro
- **Local repository**: spazio che contiene tutti i file comitati e le informazioni sui branch. Il main è un repository usato per collegare le diverse branch
- **Remote**: spazio che contiene tutte le versioni del progetto



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 **SESAR LAB**

- Le operazioni essenziali per gestire un progetto di versioning sono:
 - Creare e utilizzare un repository
 - Avviare e gestire un nuovo branch
 - Apportare modifiche a un file e inviarle a GitHub come commit
 - Aprire e unire una richiesta di pull

CREAZIONE DI UN REPOSITORY



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Un repository viene solitamente utilizzato per organizzare un singolo progetto
- I repository possono contenere cartelle e file, immagini, video, fogli di calcolo e insiemi di dati: tutto ciò di cui il progetto ha bisogno
- Spesso, i repository includono un file README, un file con informazioni sul progetto. I file README sono scritti nel linguaggio Markdown
- Creando il progetto è anche possibile impostarlo come progetto pubblico o privato e scegliere la licenza da applicare

The screenshot shows the GitHub 'Create new repository' form. At the top, the 'Owner' is set to 'octocat' and the 'Repository name' is 'hello-world', which is marked as valid with a green checkmark. Below this, a note states: 'Great: repository names are short and memorable. Need inspiration? How about `ubiquitous-system?`'. The 'Description (optional)' field contains the text 'My first repository'. Under the 'Visibility' section, the 'Public' option is selected, with the description 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is also visible. The 'Initialize this repository with:' section includes three checkboxes: 'Add a README file' (selected), 'Add .gitignore' (unchecked), and 'Choose a license' (unchecked). Each option has a link to 'Learn more'. At the bottom, a note says 'This will set `main` as the default branch. Change the default name in your [settings](#).' A green 'Create repository' button is at the bottom right.

COLLEGARE UNA DIRECTORY LOCALE



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- È possibile collegare una directory locale ad un repository

- Una volta nella directory digitare per inicializzarla

```
$ git init -b main
```

- Aggiungere i file della directory in una lista di elementi che potremo salvare attraverso commit

```
$ git add *
```

- Salvare le modifiche nel repository locale

```
$ git commit -m "First commit"
```

- Collegare il repository locale al repository remoto

```
$ git remote add origin <REMOTE_URL>
```

```
# Sets the new remote
```

```
$ git remote -v
```

```
# Verifies the new remote URL
```

- Inviare le modifiche del repository locale al repository remoto

```
$ git push -u origin main
```



UNIVERSITÀ
DEGLI STUDI
DI MILANO

 SESAR LAB

- Il più semplice workflow di gestione delle modifiche si limita a caricare le modifiche locali nel repository remoto

```
$ git status
```

```
$ git add *
```

```
$ git commit -m "my comment"
```

```
$ git push
```

- Per clonare un repository

```
$ git clone <REMOTE_URL>
```

```
$ git init
```

```
$ git remote add origin <REMOTE_URL>
```

- Per ridefinire la struttura dei file, ad esempio quando si fa copia e incolla

```
$ git clean -f
```

CREARE UN BRANCH



- Per lavorare su una nuova versione di un documento e decidere in seguito se integrare le modifiche è possibile creare dei branch
 - Se ci è stato assegnato il ruolo di collaboratore ci si può semplicemente spostare dal master e creare un branch

```
$ git checkout -b revcap1
```

```
$ git pull
```

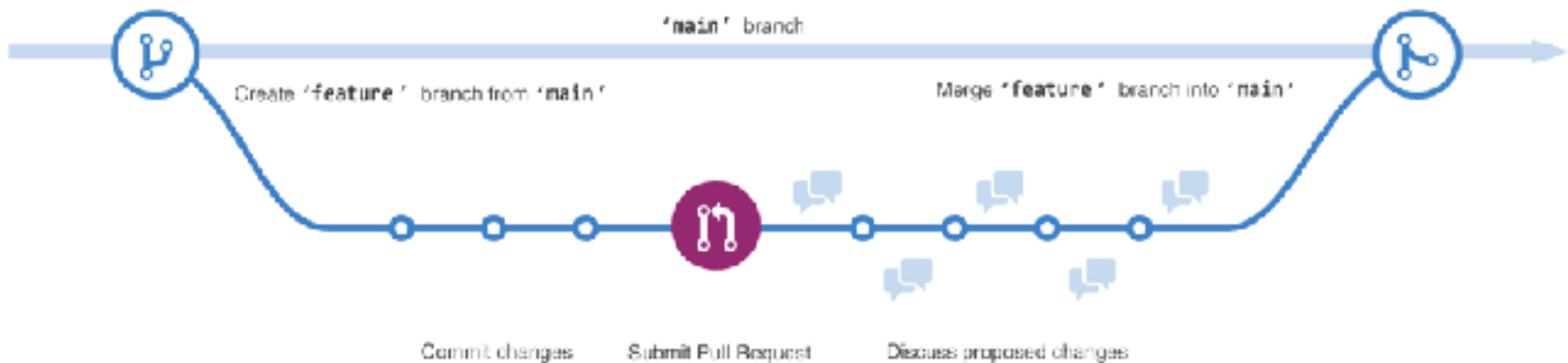
- A questo punto è possibile fare modifiche al documento

```
$ git add .
```

```
$ git commit -a -m 'Revising section 3 of cap1'
```

```
$ git push origin revcap1
```

- A questo punto è possibile fare una pull request
 - Visitare il repository originale su GitHub. Dovrebbe apparire un messaggio che indica che è stato recentemente creato un nuovo ramo. Fare clic sul pulsante "Compare & pull request". Dopo aver creato la richiesta di pull, è possibile rivedere le modifiche e discuterle con i manutentori del progetto. Se tutto sembra a posto, cliccare di nuovo sul pulsante "Create pull request" per inviare la richiesta
- I manutentori del progetto esamineranno la pull request e potranno richiedere modifiche o accettarla



CREARE UN FORK



- Se non si possiede il ruolo di collaboratore è possibile creare un fork del progetto
- Fork del Repository: sulla pagina del repository GitHub a cui si desidera contribuire, cliccare sul pulsante "Fork" nell'angolo in alto a destra della pagina. Questo creerà una copia del repository nel tuo account GitHub.
- A questo punto è possibile clonare il Fork in un repository locale: copiando l'URL del fork dalla barra degli indirizzi del browser, poi attraverso il terminale:

```
$ git clone <URL del repository>
```

- Si può poi creare un nuovo branch per gestire le modifiche:

```
$ git checkout -b feature/duo-nome-feature
```

- Si possono a questo punto effettuare le modifiche:

```
$ git add .
```

```
$ git commit -a -m 'Revising feature'
```

```
$ git push origin feature/tuo-nome-feature
```

- Si può a questo punto creare una pull request:

- Visitare la pagina del fork su GitHub. Dovrebbe apparire un messaggio che indica che è stato aggiornato un branch. Si può ora cliccare sul pulsante "Confronta e crea richiesta di pull". Aggiungere un titolo e una descrizione dettagliata per la tua richiesta di pull. Si può a questo punto inviare la Pull request

- I manutentori del progetto esamineranno la pull request e decideranno se accettarla

- L'editing collaborativo è il processo in cui più persone lavorano insieme per creare, modificare o migliorare un documento, un contenuto multimediale o un progetto editoriale
- **Co-creazione:** tutti i partecipanti contribuiscono attivamente al contenuto
- **Revisione e feedback:** i collaboratori possono commentare e proporre modifiche, spesso in tempo reale
- **Sincronicità e asincronicità:** il lavoro può avvenire simultaneamente (sincrono) o in momenti diversi (asincrono)
- **Qualità dei contenuti:** molte ricerche mostrano che può migliorare la qualità dei contenuti
- **Rapidità dei processi:** molte ricerche mostrano che può ridurre i tempi di produzione
- **Cultura organizzativa:** richiede una buona organizzazione e un atteggiamento di collaborazione

EDITING COLLABORATIVO



UNIVERSITÀ
DEGLI STUDI
DI MILANO



Damiani, E., Ceravolo, P., Frati, F., Bellandi, V., Maier, R., Seeber, I., & Waldhart, G. (2015). Applying recommender systems in collaboration environments. *Computers in Human Behavior*, 51, 1124-1133.



UNIVERSITÀ
DEGLI STUDI
DI MILANO

SESAR LAB

- Nel 1995 nascono, per opera di Ward Cunningham, i wiki, primo esempio di editing collaborativo online
- Collaborazione asincrona: più persone potevano contribuire, ma non simultaneamente
- Esempio iconico: Wikipedia 2001, che ha rivoluzionato la condivisione della conoscenza
- Uno dei primi software che consentiva a più utenti di modificare lo stesso documento in tempo reale è SubEthaEdit, precedentemente noto come Hydra, che è stato introdotto nel 2003 per MacOS

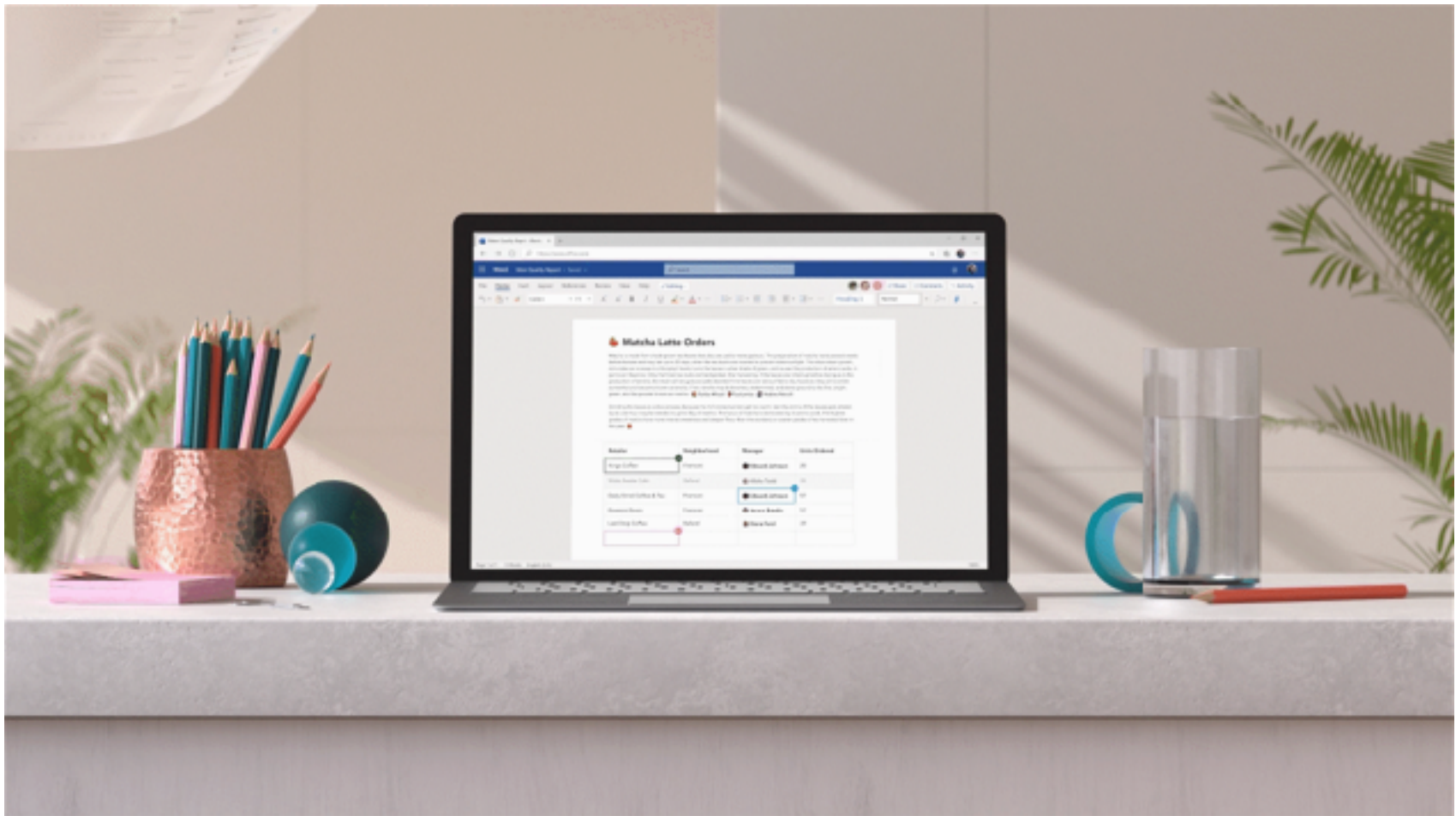
- Nel 2008 Etherpad ha attirato l'attenzione soprattutto dopo che la sua prima versione è stata acquistata nel 2009 da Google e rilasciata con licenza open source
- Nel 2009 Google tolto il progetto Google Docs dallo stato di beta testing, cosentino a tutti gli utenti Google di utilizzarle il servizio. Questo renderà l'editing collaborativo molto popolare
- Nel 2014, Google lancia un'applicazione mobile, per sistemi Android e iOS, dedicata a Docs
- Nel 2020 Microsoft introduce Fluid Framework
 - Supporto per componenti fluidi, unità di contenuto che possono essere integrate in app diverse.
 - Collaborazione cross-app: documenti e dati condivisi in tempo reale attraverso strumenti Microsoft (Word, Teams) e applicazioni di terze parti

- Nel 2008 GitHub combina Git con una piattaforma web per la collaborazione
 - Pull Request: per revisionare e integrare modifiche
 - Issue Tracker: gestione delle attività di progetto
 - Documentazione tecnica collaborativa (es. Markdown nei repository)
- Nel 2014 nasce GitBook, inizialmente come un progetto open-source basato su Git e Markdown
 - Consente a più utenti di lavorare simultaneamente su documenti, con sincronizzazione in tempo reale
 - Struttura i contenuti in capitoli e sezioni, ideale per documentazioni tecniche, manuali e guide
 - Originariamente integrato con Git, ora offre una gestione interna del versioning per tracciare le modifiche ai documenti
 - Documenti pubblicabili su un URL personalizzato o accessibili solo a utenti autorizzati

FLUID FRAMEWORK



UNIVERSITÀ
DEGLI STUDI
DI MILANO



-
- Microsoft ha reso disponibile in forma di preview la prima build di Fluid Framework,
 - Creazione di documenti online interattivi.
 - Web collaboration e dei servizi interattivi.
 - Attualmente rilasciato
 - Dividere il documento in più blocchi collaborativi, per poi analizzarli e modificati da più App in tempo reale
 - Documenti tradotti contemporaneamente in più lingue
 - Fluid potrà interfacciarsi con Cortana o con altri servizi di intelligenza artificiale, così da offrire immagini, video o altri dati basati sui contenuti scritti nel documento

Microsoft Build 2019

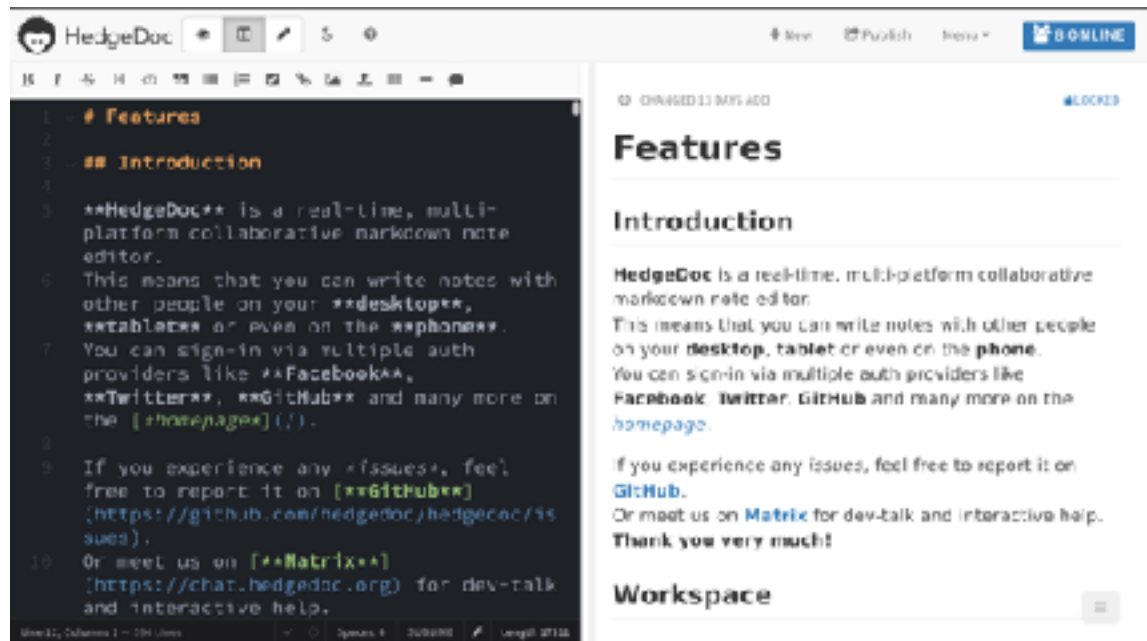
Fluid Framework // Vision Keynote Demo

Speaker
Mike Morton

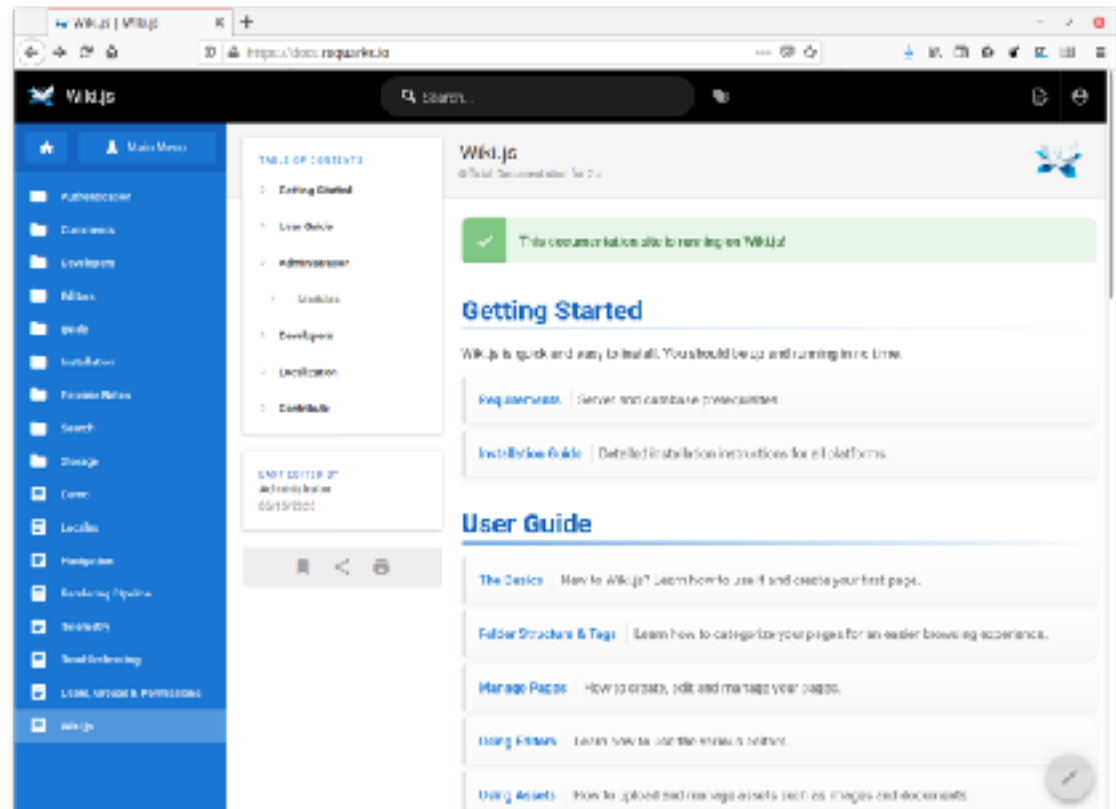


<https://www.youtube.com/watch?v=RMzXmkrlFNg>

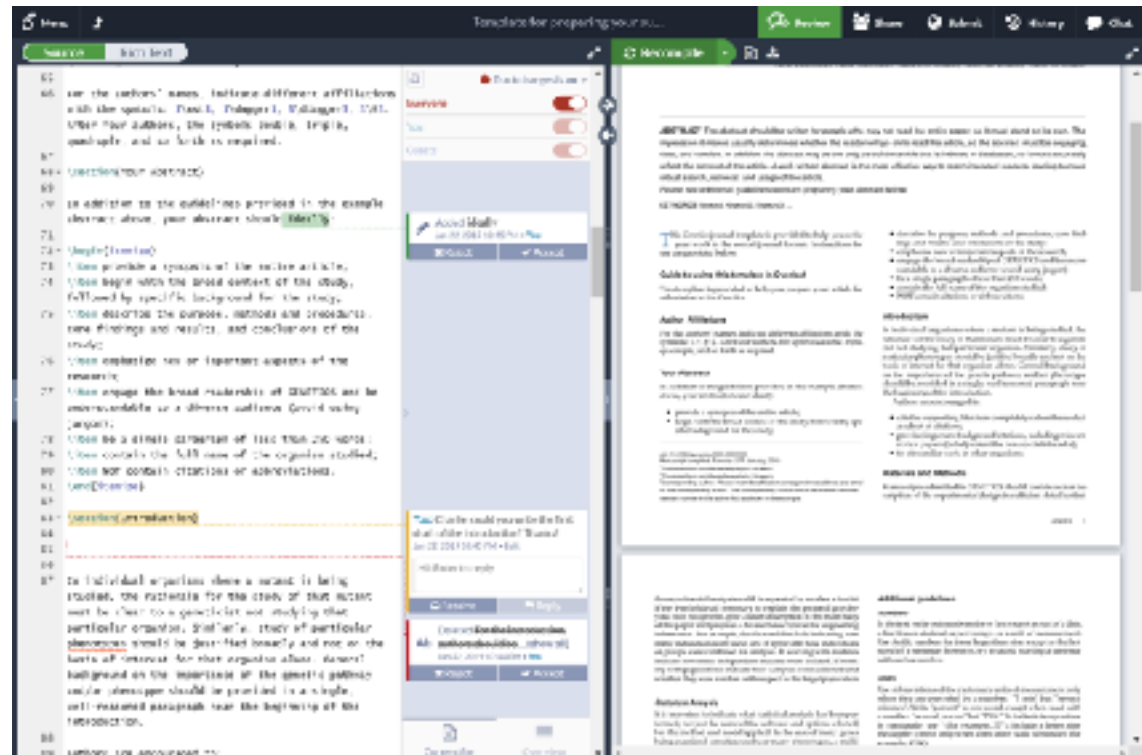
- ▶ HedgeDoc è una piattaforma open-source per l'editing collaborativo di documenti in tempo reale, con supporto nativo per il linguaggio Markdown
- ▶ Si ispira a Hackpad, Etherpad e altri editor collaborativi simili
- ▶ Utilizza Markdown come linguaggio principale per la formattazione del testo
- ▶ Supporta una modalità di commento e discussione direttamente all'interno del documento
- ▶ Ogni documento può essere condiviso tramite un link pubblico o privato
- ▶ Possibilità di esportare i documenti in vari formati, inclusi HTML e PDF



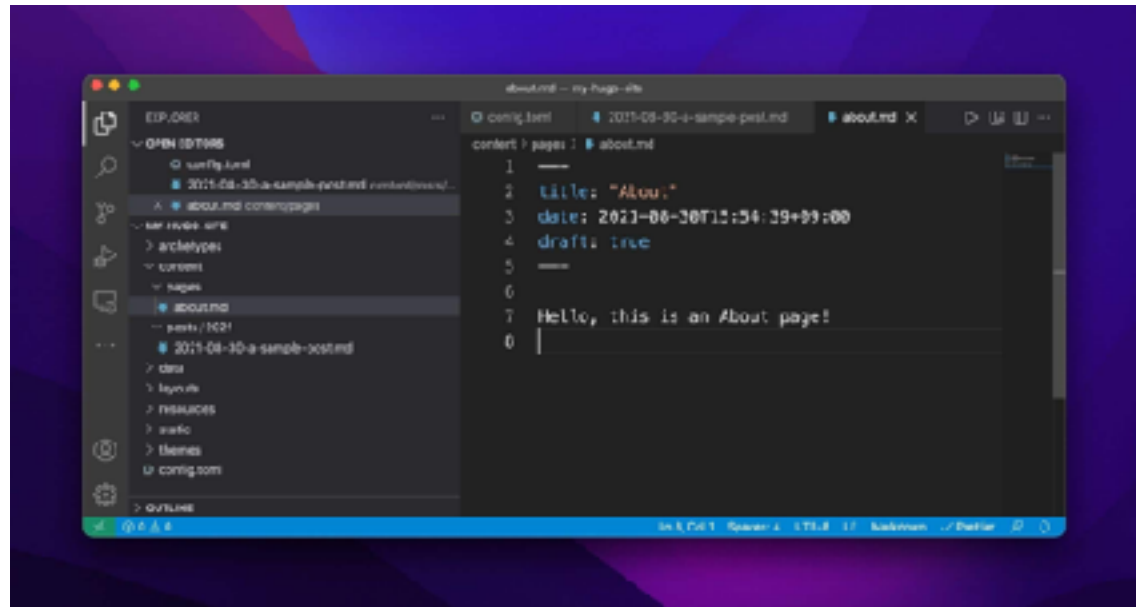
- ▶ Wiki.js è una piattaforma open-source per la gestione di documenti e contenuti, pensata per creare e mantenere wiki in modo semplice e collaborativo
- ▶ Include strumenti per il versioning, la revisione e la gestione dei permessi, favorendo il lavoro in team
- ▶ Supporta la modifica in Markdown, un linguaggio di markup semplice, ma permette anche di utilizzare altri formati per i contenuti
- ▶ Estensibile con moduli per aggiungere funzionalità come autenticazione avanzata, integrazione con servizi esterni (es. Git), e temi personalizzati



- ▶ Overleaf è una piattaforma online per la scrittura e la collaborazione su documenti in LaTeX, un linguaggio di markup ampiamente utilizzato per creare documenti scientifici, accademici e tecnici con una tipografia di alta qualità
- ▶ Permette a più utenti di lavorare simultaneamente su un documento, con sincronizzazione istantanea
- ▶ Aggiorna automaticamente il PDF del documento mentre si lavora sul codice sorgente
- ▶ Non richiede configurazioni locali di LaTeX, rendendo l'uso accessibile a chiunque
- ▶ Possibilità di lasciare commenti inline per discutere modifiche con i collaboratori
- ▶ Controllo delle versioni per ripristinare versioni precedenti del documento



- ▶ Hugo è un framework open-source per la creazione di siti statici, noto per la sua velocità, flessibilità e semplicità
 - ▶ È scritto in Go (Golang) ed è particolarmente popolare tra sviluppatori e creatori di contenuti che cercano un modo rapido ed efficiente per costruire siti web ottimizzati. Hugo è adatto per blog personali, documentazione tecnica, portfolio e siti aziendali
- ▶ Hugo utilizza Markdown per la creazione di contenuti, rendendolo accessibile anche a chi non ha esperienza di sviluppo web
- ▶ Separazione netta tra contenuti e presentazione grazie ai template
- ▶ È possibile archiviare tutto il codice del sito generato da Hugo (inclusi contenuti, temi e configurazioni) in un repository Git su GitHub
- ▶ Attraverso Git si può quindi gestire un processo di editing collaborativo



Promozione

Lancio e
anticipazioni

Presentazione del
prodotto

Attività sui canali

Verifica post produzione

Raccolta dati dai
canali

Verifica del
Target

Verifica dei
Canali

Revisione degli obiettivi
comunicativi

Revisione dei contenuti