



# AGILE

## Scrum

Il manifesto per lo sviluppo agile di software è un documento redatto nel febbraio 2001 da diciassette sviluppatori con l'intento di definire i **valori e i principi chiave** dello sviluppo agile.

## Perché:

- tradizionali metodi che troppo spesso hanno fallito clamorosamente
- vecchi processi erano: troppo costosi, troppo lenti e spesso incapaci di portare ai risultati sperati.

## Nascita dell'AGILE Alliance

Dall'11 al 13 febbraio 2001, nello ski resort "The Lodge at Snowbird" nelle montagne Wasatch nello Utah, **diciassette persone** si sono incontrate per **parlare, sciare e rilassarsi insieme**. Tra gli **obiettivi** c'era anche quello di **trovare un'intesa comune**.

Ciò che **emerse** da questo incontro è il **manifesto** per lo sviluppo agile di software, talvolta abbreviato come "**manifesto agile**", sottoscritto da tutti e diciassette.

# Manifesto AGILE



# Cos'è Importante

---

Siamo arrivati a considerare importanti:

- Gli individui e le interazioni più che i processi e gli strumenti
- Il software funzionante più che la documentazione esaustiva
- La collaborazione col cliente più che la negoziazione dei contratti
- Rispondere al cambiamento più che seguire un piano

Ovvero, fermo restando il valore delle voci a destra, consideriamo più importanti le voci a sinistra.

# I 12 Principi

1

- La nostra priorità più alta è soddisfare il cliente attraverso rilasci immediati e continui di software di valore.
- Accetta cambi nei requisiti, anche a sviluppo inoltrato. Il processo agile sfrutta il cambiamento a vantaggio competitivo del cliente.
- Rilascia software funzionante di frequente, da un paio di settimane a un paio di mesi, con una preferenza per tempistiche più brevi.
- La parte di business e gli sviluppatori lavorano insieme e quotidianamente durante tutto il progetto.

# I 12 Principi

## 2

- Costruisci il software intorno a persone motivate. Dà loro il contesto e supporta i loro bisogni, e fidati di loro che il lavoro sarà svolto.
- Il metodo più efficiente e efficace di convogliare le informazioni da e verso il team di sviluppo è la comunicazione faccia a faccia.
- Il software funzionante è la primaria misura di progresso.
- Il processo agile promuove lo sviluppo sostenibile. Sponsor, sviluppatori e utenti devono essere in grado di mantenere un passo costante indefinitamente.

# I 12 Principi

## 3

- Una continua attenzione all'eccellenza tecnica e al buon design migliora l'agilità.
- La semplicità – l'arte di massimizzare la quantità di lavoro non fatto – è essenziale..
- Le migliori architetture, requisiti e design emergono da team auto organizzati.
- A intervalli regolari, il team riflette su come divenire più efficienti, poi regola e aggiusta il proprio comportamento di conseguenza.



# The Map is not the Territory



How the customer explained it



How the project leader understood it



How the analyst designed it



How the programmer wrote it



How the business consultant described it



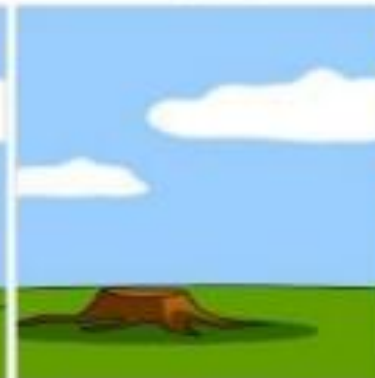
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

# Concetto MVP

## 1

1. MVP è il prodotto con più alto ritorno di investimenti rispetto al rischio (wikipedia).
2. MVP è una metodologia di sviluppo orientata ad avere il prima possibile una versione con funzionalità essenziali per gli *early adopter*
3. Un MVP è un software rilasciabile, che permette di raccogliere immediatamente informazioni sull'utilizzo

# MVP 2

## HOW TO BUILD MINIMUM VIABLE PRODUCT RIGHT

NOT LIKE THIS:



INSTEAD LIKE THIS:



# SCRUM

- framework agile creato da Ken Schwaber e Jeff Sutherland
- framework per sviluppare e sostenere prodotti complessi
- non una tecnica, non un processo
- controllo empirico da cicli di feedback frequenti in opposizione al *command and control*
- facile da capire
- difficile da padroneggiare
- leggero
- basato sulla scrum guide (19 pag.) [scrumguides.org](https://www.scrumguides.org)

# SCRUM

## Pilastri

Scrum è un approccio basato sulla teoria del controllo empirico dei processi: le decisioni vengono prese sulla base dell'esperienza (empirismo).

- **trasparenza:** gli aspetti significativi del processo devono essere visibili ai responsabili del risultato
- **ispezione:** è un aspetto critico dell'empirismo, può essere fatta sia sul prodotto, che sulle persone, che su ogni aspetto di scrum
- **adattamento:** viene dall'ispezione e va applicato il prima possibile per iniziare un nuovo ciclo di (trasparenza) ispezione e adattamento

# SCRUM

## 5 Valori

- impegno
- coraggio
- concentrazione (focus)
- apertura
- rispetto

I valori vanno vissuti per dare vita ai pilastri e creare fiducia.  
Scrum si nutre di e alimenta questi valori

# Scrum come funziona

Come tutte le metodologie Agile, si basa sulla divisione del progetto in più fasi, chiamate Sprint:

nuove funzionalità → fatte valutare al cliente



sistema iterativo che consente di **incrementare** poco alla volta le funzionalità del progetto



verificare costantemente **l'andamento** complessivo e la **soddisfazione** del cliente

# Scrum team

---

Lo scrum team è

- auto organizzato: sceglie come meglio compiere il proprio lavoro
- cross funzionale

Il modello serve per ottimizzare:

- flessibilità
- creatività
- produttività



# Scrum team

- Product Owner (PO)
- Dev Team (DEV)
- Scrum Master (SM)

## Product Owner

---

Il Product Owner ha la responsabilità di massimizzare il valore del prodotto risultante dal lavoro svolto dal Team di Sviluppo.

E' il responsabile reale del prodotto e tutti devono rispettare le sue decisioni. Nessun'altro ha il permesso di dire al dev team su cosa lavorare.

# Product Backlog 1

Il Product Backlog è un elenco ordinato di tutto ciò che è noto essere necessario al prodotto.

Esso è l'unica fonte di requisiti per le modifiche da apportare al prodotto.

Il Product Owner è il responsabile del Product Backlog, compreso il suo contenuto, la sua disponibilità e l'ordinamento dei suoi elementi.

Il productr backlog non è mai completo ed è documentazione di quello che è stato e quello che sarà il prodotto..

## Product Backlog 2

---

L'ordine rappresenta l'importanza e la conoscenza dei requisiti (storie).

Gli elementi ordinati più in alto sono solitamente più chiari e meglio dettagliati rispetto a quelli più in basso.

Il dev-team e il product owner si occupano, durante lo sprint, di fare attività di raffinamento, per migliorare la descrizione dei requisiti e fornirne alcune stime.

# Product Backlog

## 3

Il Product Owner è l'unica persona responsabile della gestione del **Product Backlog**:

- chiarezza degli elementi
- corretto ordinamento degli elementi per raggiungere gli obiettivi
- ottimizzazione del lavoro del devteam
- visibilità, trasparenza e chiarezza del product backlog
- comprensione da parte dei membri del devteam

**Professionisti** che lavorano insieme per consegnare un incremento di prodotto **done** potenzialmente rilasciabile alla fine di ogni **sprint**.

- auto organizzati
- cross funzionali
- non esistono titoli o gerarchie interne
- non esistono sotto-team
- il team nel complesso ha la responsabilità condivisa, non esistono responsabilità per specializzazione / area

TUTTO IL DEV TEAM E' EGUALMENTE RESPONSABILE VERSO IL PRODUCT OWNER.

Dimensione:

2 Pizzas Rule: 3-9 persone

Team troppo piccoli: rischio di mancanza di skill

Team troppo grossi: troppa complessità rispetto a quella gestibile empiricamente

# Scrum Master

---

Responsabile di promuovere e sostenere SCRUM. Aiuta chiunque a comprendere teoria - pratiche - regole - valori.

E' un leader al servizio dello scrum team (**servant leadership**).



## Scrum Master servant per PO

- assicura che gli obiettivi siano chiari a tutti nello scrum team
- cerca tecniche efficaci per la gestione del product backlog
- supporta la comprensione della pianificazione empirica
- aiuta il po a massimizzare il valore
- facilita gli eventi di scrum **come richiesto e necessario**

## Scrum Master servant per dev

- allena il team all'auto organizzazione e alla cross-funzionalità
- aiuta a creare prodotti di alto valore
- elimina gli ostacoli di avanzamento
- facilita gli eventi di scrum come richiesto e necessario
- assiste (e protegge) il dev team in ambienti in cui scrum non è ancora pienamente adottato o compreso

- sprint
- daily scrum
- sprint review
- sprint retrospective
- sprint planning

Creano regolarità quindi consistenza

Minimizzano i meeting non definiti in scrum

Sono **timeboxed** e supportano trasparenza e ispezione

Riducono la complessità grazie alla predittibilità

## EVENTI sprint

Periodo **time-boxed** di [1-4] settimane durante la quale viene creato un **incremento** di prodotto potenzialmente rilasciabile (**done**).

E' di durata costante (e invaribile, in corsa).

Alla fine di uno sprint parte immediatamente un nuovo sprint. Contiene tutti gli altri eventi: lo Sprint Planning, i Daily Scrum, il lavoro di sviluppo (non è un evento), la Sprint Review e la Sprint Retrospective

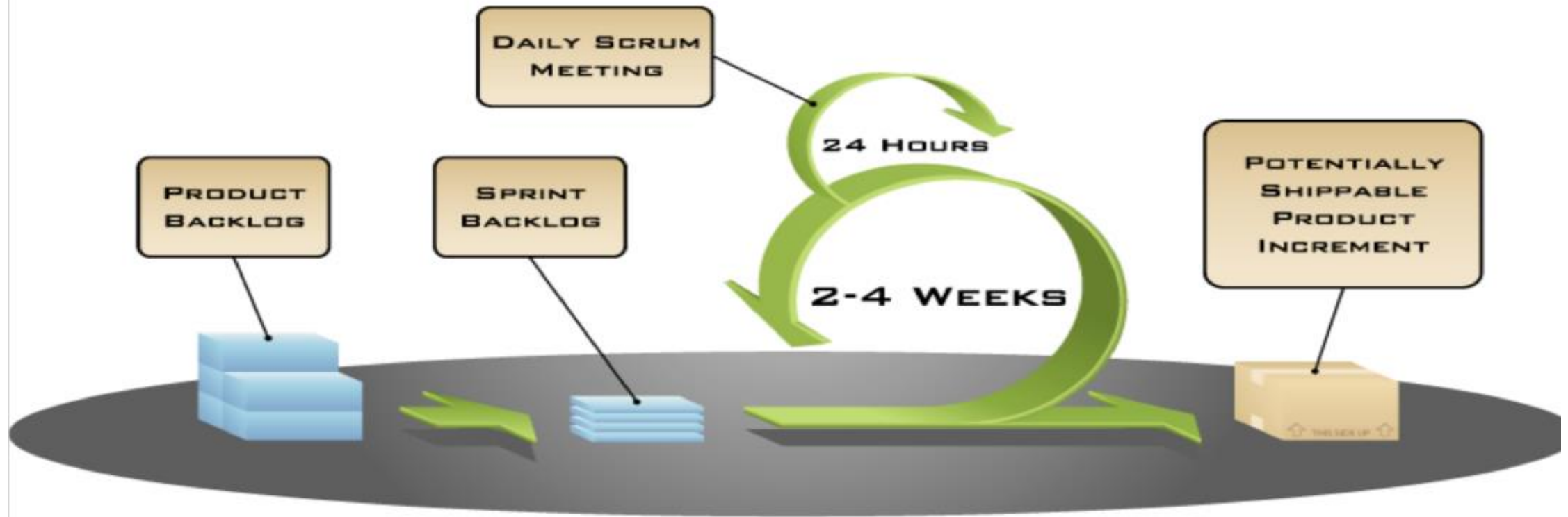
## EVENTI sprint

Ogni sprint ha un **obiettivo** di ciò che si va a costruire e uno **Sprint Goal** (deciso dal PO).

Durante:

- non possono essere fatte modifiche che mettano a rischio il goal
- gli obiettivi di qualità non devono degradarsi
- la visibilità può essere chiarita e rinegoziata tra il Product Owner e il devteam, quando si è appreso di più.

# EVENTI sprint



## EVENTI

### Daily Scrum (standup meeting)

max 15 min - solo per il dev team - tutti i giorni

- cosa ho fatto ieri
- cosa farò oggi
- ci sono eventuali impedimenti che bloccano il mio lavoro

no discorsi tecnici

no discussioni

no risoluzione problemi

lo scrum master presenza (ma non interviene)

## EVENTI

### Sprint Review

Alla fine di ogni sprint - max 4 ore

Il dev team mostra quello che ha fatto al PO e a eventuali stakeholder (*demo*)

Si parla di quello che è andato **bene** e dei problemi incontrati

Il product owner **accetta** (o meno) gli elementi fatti come incremento



## EVENTI

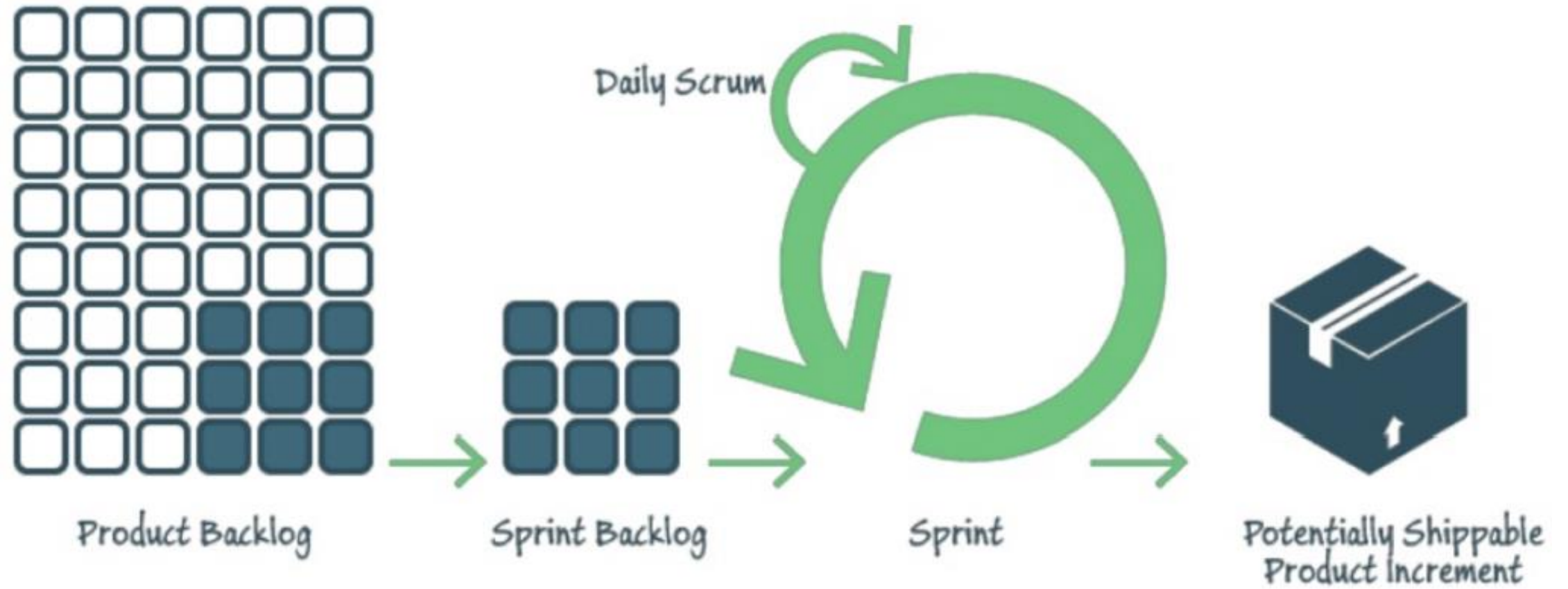
### Sprint Planning

Si pianifica il lavoro da eseguire nello sprint che sta per iniziare, max 8 ore.

- definizione dello **sprint goal**
- decisione dell'incremento da portare al prodotto durante lo sprint

Si prendono gli elementi in testa al **product backlog** e si mettono nello **sprint backlog**. Il devteam è l'**unico** che può decidere quanti elementi prendere

# EVENTI



## EVENTI

### Sprint Backlog (artefatto)

Insieme degli elementi del Product Backlog selezionati per lo Sprint.

Previsione fatta dal Team di Sviluppo riguardo a quale funzionalità sarà presente nel prossimo **Incremento**, e riguardo al lavoro necessario per consegnare tale funzionalità in un Incremento **done**.

Aggiornato durante lo sprint per meglio raggiungere lo sprint goal.

Immagine in tempo reale del lavoro che il Team di Sviluppo prevede di compiere durante lo Sprint  
**di esclusiva appartenenza del Team di Sviluppo**

## Sprint Backlog Incremento(artefatto)

$$I = I(n) + I(n - 1)$$

Somma di tutti gli elementi del product backlog completati durante uno sprint e del valore degli incrementi degli sprint precedenti.

Il nuovo incremento deve risultare **done** per essere considerato tale.

E' un passo verso la visione/obiettivo e ogni incremento deve essere ispezionabile.

Deve essere *potenzialmente rilasciabile*.

## Sprint Backlog Definition of Done

Tutti devono capire quando un elemento del backlog è done.  
DoD è un documento, condiviso e pragmatico che sancisce cosa si intende per done.

La DoD guida lo scrum team nella fase di sprint planning, e può cambiare solo alla fine di uno sprint.

La propone l'organizzazione ma gli scrum team possono approfondirla.

## EVENTI

### Sprint Retrospective

E' un'occasione per lo scrum team per ispezionare se stesso e creare un piano di miglioramento per il prossimo sprint. max 3h

- ispezione ultimo sprint: persone, relazioni, processi, strumenti
- identificare gli elementi che sono andati bene e possibili miglioramenti
- creare un piano di miglioramenti per il team (action points)

Fornisce una opportunità formale per focalizzare l'ispezione e l'adattamento

## EVENTI

### Sprint Planning Meeting

- sprint review
- chiusura dello sprint
- sprint retrospective
- sprint planning
- avvio dello sprint

# Approfondimenti

- [scrumguides](#)
- [scrum.org](#)
- [scrumalliance.org](#)
- [heartofagile.com](#)



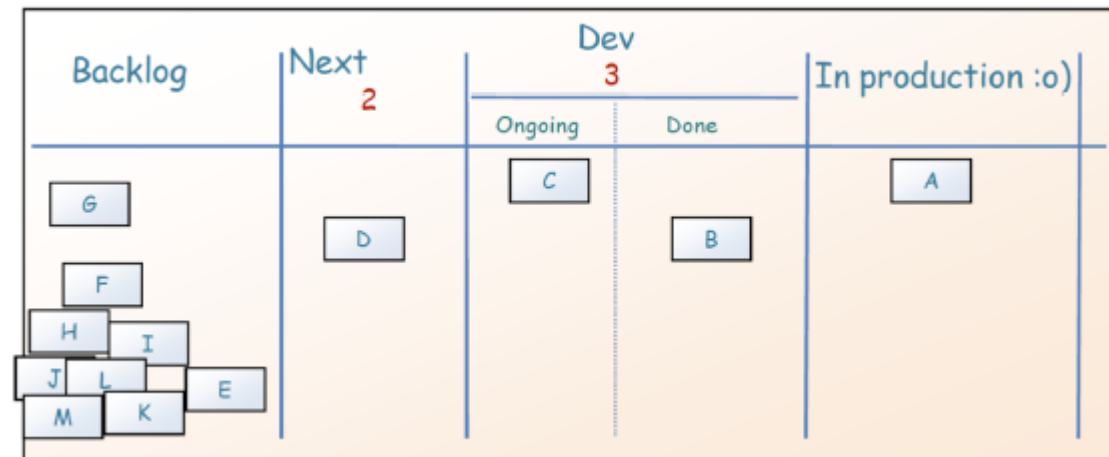
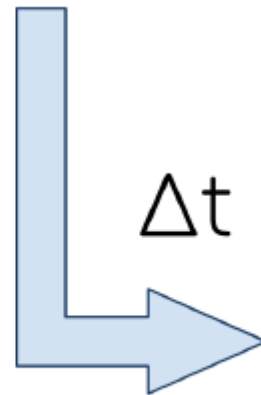
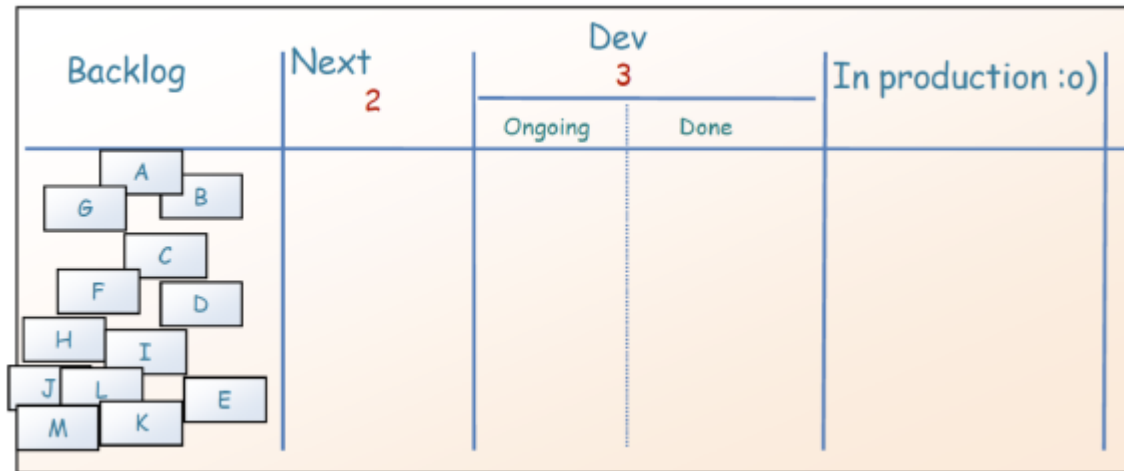
# Kanban

## 1

- utilizzo di board a colonne
- limitare il work in progress
- visualizzare il workflow
- misurare il lead time
- fornire un feedback visuale del lavoro

# Kanban

## 2



tecnica mista che utilizza le board kanban per organizzare il lavoro dello sprint

Strumenti:

- [Trello](#) - Noi useremo questo
- [Jira](#)
- [Clickup](#)
- [Asana](#)

Miglior strumento

Postit

