

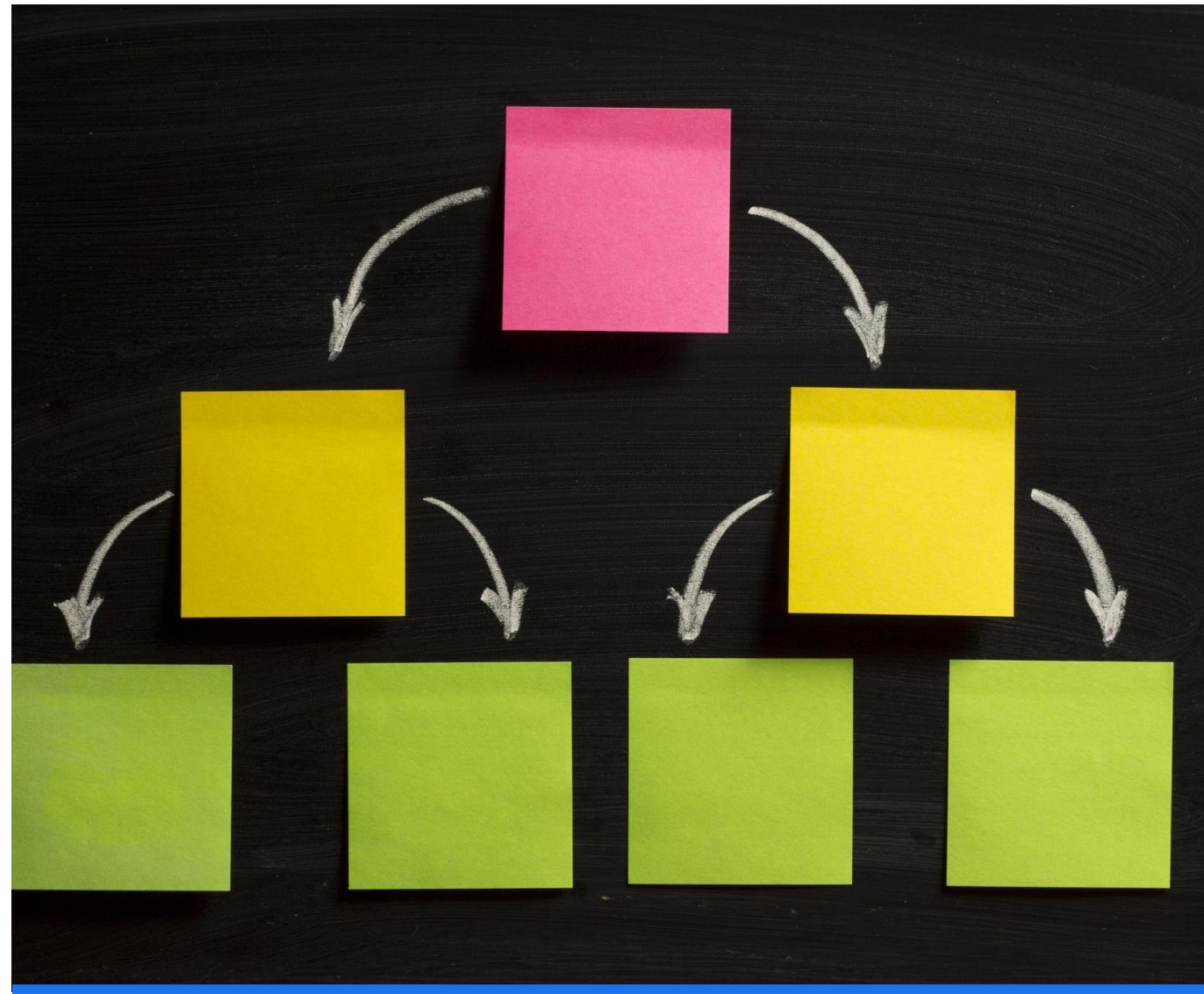


# Scrum in pratica: principi, ruoli, eventi e gestione del valore

APPROCCIO AGILE PER  
TEAM EFFICACI E  
COLLABORATIVI

# Percorso della Presentazione

- Origini e principi fondamentali di Scrum
- Ruoli chiave nel framework Scrum
- Caratteristiche e dimensione ottimale dei team Scrum
- Eventi Scrum: struttura e obiettivi
- Artefatti Scrum e gestione del valore
- Gestione dello spreco e focalizzazione
- Stima, pianificazione e approccio story-driven
- Esercizio pratico: progettare il primo Sprint



# Origini e principi fondamentali di Scrum

---

# Dalla guerra del Vietnam al concetto di Sprint

## Origine del concetto Sprint

L'idea dello Sprint nasce dalle missioni aeree durante la guerra del Vietnam, caratterizzate da cicli brevi e intensi di azione.

## Importanza del feedback immediato

Le missioni richiedevano feedback immediato per adattarsi rapidamente a situazioni in rapido mutamento.

## Ciclo OODA nell'aviazione

Osservare, orientarsi, decidere e agire: il ciclo OODA è fondamentale per la rapidità decisionale in combattimento.



# Il ciclo di ispezione e adattamento continuo

## Importanza del debriefing

Il debriefing dopo una missione aiuta a identificare successi, errori e lezioni apprese.

## Ciclo di ispezione

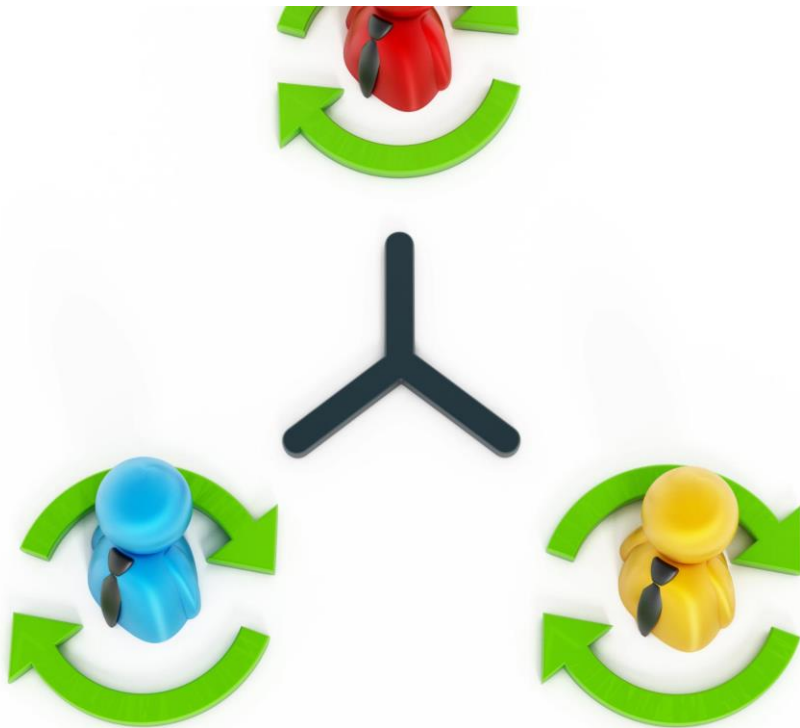
L'ispezione continua consente di valutare cosa funziona e cosa migliorare in ogni iterazione.

## Adattamento continuo

Adattare costantemente le azioni migliora i processi e guida il successo a lungo termine.

## Fondamento di Scrum

Questo ciclo costituisce il cuore di Scrum, che mira a doppiare la produttività in metà tempo.



# Ruoli chiave nel framework Scrum

---

# Product Owner: responsabilità e esempio LinkedIn



## Definizione della visione del prodotto

Il Product Owner definisce cosa deve fare il prodotto e perché, mantenendo la visione chiara e condivisa.

## Gestione del Product Backlog

Il Product Owner gestisce la lista delle funzionalità prioritarie per garantire il valore massimo al cliente e all'azienda.

## Interfaccia con stakeholder

Il Product Owner comunica con stakeholder e clienti per allineare le aspettative e le priorità del prodotto.

## Esempio di leadership: Reid Hoffman

Reid Hoffman ha incarnato il ruolo del Product Owner con una chiara visione: costruire la prima rete professionale online.

# Scrum Master: servant leadership e aneddoti storici

## Ruolo del Scrum Master

Il Scrum Master è un servant leader che facilita eventi Scrum e supporta il team rimuovendo impedimenti.

## Servant Leadership Toyota

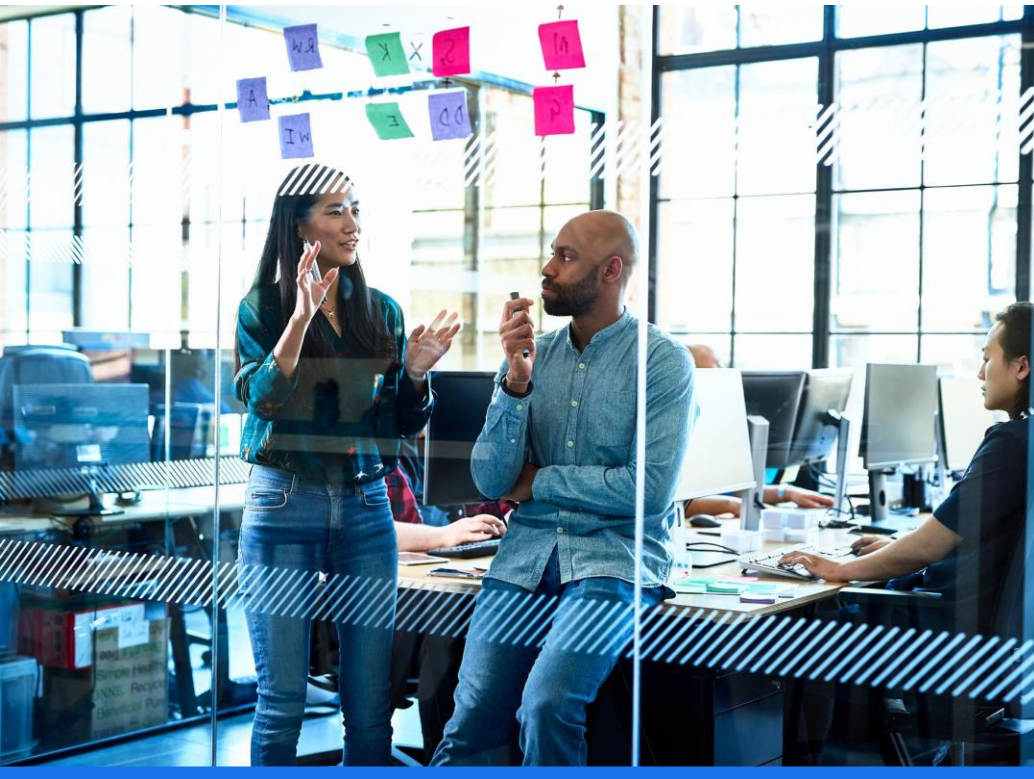
Il concetto di servant leadership viene dalla filosofia Toyota per rimuovere ostacoli nel flusso produttivo.

## Aneddoto di Jeff Johnson

Jeff Johnson organizzò il lavoro in cicli brevi e si concentrò sui problemi principali rimuovendo ostacoli senza dare ordini.



# Development Team: caratteristiche e regola delle "due pizze"



## Dimensione ottimale del team

Il team ideale ha tra 3 e 9 membri seguendo la regola delle "due pizze" di Jeff Bezos per efficienza e comunicazione.

## Team cross-funzionale

I membri possiedono competenze diverse e complementari necessarie per completare il lavoro senza dipendenze esterne.

## Auto-organizzazione

Il team decide autonomamente come svolgere il lavoro, migliorando flessibilità e responsabilità individuale.

## Collaborazione e responsabilità condivisa

I membri collaborano quotidianamente e condividono la responsabilità del successo del progetto.

# Caratteristiche e dimensione ottimale dei team Scrum

---



# Team trascendenti, autonomi e cross-funzionali

## Team trascendenti

I team trascendenti hanno un senso della finalità superiore e rifiutano la mediocrità, puntando alla grandezza.

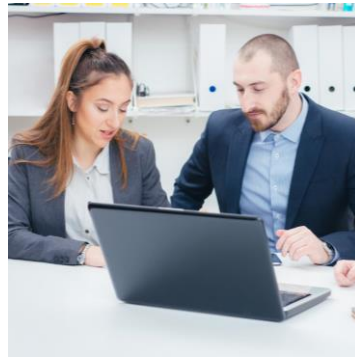
## Team autonomi

I team autonomi sono auto-organizzati e prendono decisioni senza bisogno di micromanagement.

## Team cross-funzionali

Team cross-funzionali possiedono competenze diverse che si rafforzano e alimentano a vicenda.

# Dimensione ottimale e canali di comunicazione



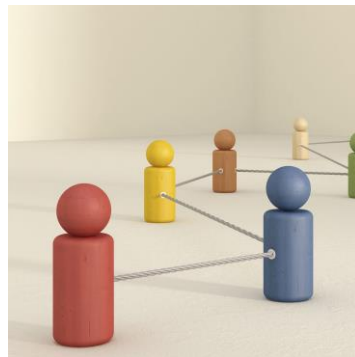
## Dimensione ottimale del team

La dimensione ideale del team è tra 5 e 9 membri per massimizzare efficienza e comunicazione.



## Legge di Brooks

Aggiungere membri a un progetto in ritardo può aumentare ulteriormente il ritardo complessivo.



## Canali di comunicazione

I canali di comunicazione crescono rapidamente con l'aumentare del numero di persone nel team, complicando lo scambio di informazioni.

# Eventi Scrum: struttura e obiettivi

---

# Sprint, Sprint Planning e Daily Scrum



## Concetto di Sprint

Lo Sprint è un periodo di tempo fisso in cui il team crea un incremento di prodotto rilasciabile e mantiene un ritmo costante.



## Sprint Planning Collaborativo

All'inizio dello Sprint si pianificano le attività dal Product Backlog, mostrando le priorità su un muro visibile a tutti.



## Daily Scrum in 15 Minuti

Ogni mattina il team si riunisce per rispondere a tre domande chiave e mantenere l'allineamento in meno di 15 minuti.

# Sprint Review, Demo e Retrospective



## Sprint Demo e Verifica

La Demo permette al team di mostrare il lavoro svolto e verificare il progresso verso gli obiettivi dello Sprint.

## Caso Sentinel e Successo

La Demo ha dimostrato il successo di Sentinel con solo il 5% del budget in 20 mesi, sfidando le aspettative.

## Sprint Retrospective

La Retrospectiva è cruciale per identificare errori e migliorare continuamente il processo di lavoro del team.

## Errore Fondamentale di Attribuzione

Durante la Retrospectiva si evita il bias di colpevolizzare le persone, focalizzandosi sui sistemi disfunzionali.

# Artefatti Scrum e gestione del valore

---

# La pianificazione delle attività

- Prima di tutto si fa un elenco di attività da svolgere, quelle immediate, quelle future, quelle prossime, quelle possibili, quelle che chissà forse un domani... quelle improbabili
- Tutte queste attività finiscono in uno "scatolone" chiamato product backlog.
- Da lì si scelgono le attività da implementare nel prossimo sprint e si parte.
- Le attività dello sprint sono mostrate visivamente sulla agile board.



# Product Backlog: struttura e user stories



## Definizione del Product Backlog

Il Product Backlog è una lista organizzata e dinamica di funzionalità necessarie per il prodotto, gestita dal Product Owner.

## Struttura delle User Stories

Le User Stories seguono la formula 'Come [utente], voglio [funzionalità] per [beneficio]' per definire i requisiti dal punto di vista dell'utente.

## Priorità e stime

Le priorità sono stabilite in base al valore per il cliente, mentre il team assegna stime di complessità per la pianificazione.

## Esempio di User Story

Un esempio pratico mostra un utente che vuole accedere alle ultime partite per continuare a giocare e sfidare gli avversari.

# Sprint Backlog e Increment: impegno e valore tangibile

## Sprint Backlog Definizione

Il Sprint Backlog contiene gli elementi selezionati dal Product Backlog per lo Sprint corrente e il piano per completarli.

## Incremento del Prodotto

L'incremento è la versione potenzialmente rilasciabile del prodotto che deve essere utilizzabile e aggiungere valore concreto.



# Agile Board / Kanban Board

Immagina un foglio excel  
con 3 colonne:

- To do
- Doing
- Done

In To do: l'elenco di voci (o post it) da fare

In Doing: l'attività che sto facendo

In Done: le attività concluse

Vantaggi:

A colpo d'occhio tutte  
le persone del team vedono  
le attività da fare, quello  
su cui i colleghi stanno lavorando  
e le attività concluse.

La comunicazione fluisce in modo veloce e lineare.

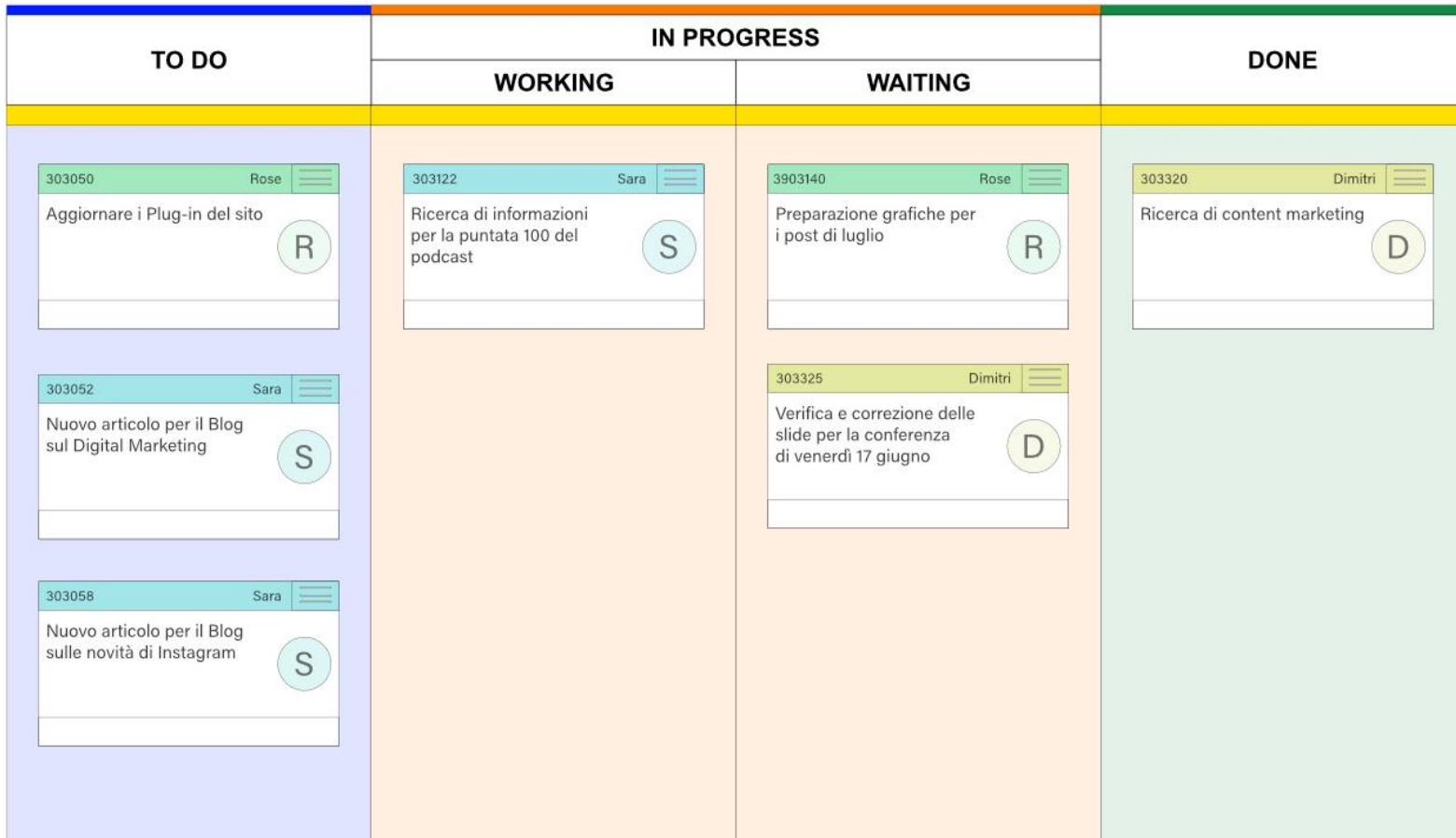
---

# Agile Board / Kanban Board

- Esistono versioni più evolute di questa board:

To do

- In progress
- divisa a sua volta in:
  - working
  - waiting
- Done



# Agile Board / Kanban Board

## Nell'approccio Agile

- Le board si svuotano sempre alla fine dello sprint
- Quello che non è stato concluso torna nel backlog
- Il nuovo sprint partirà da zero con la lavagna sempre vuota

## Nell'approccio Kanban

- Le board non si svuotano mai, non esiste la fine dello sprint
- Si va avanti a ciclo e flusso continuo

# Gestione dello spreco e focalizzazione

---

# I tre tipi di spreco giapponesi: Muda, Mura, Muri



## MUDA - Spreco di risultati

Muda rappresenta sprechi evidenti che non aggiungono valore al cliente, come lavoro inutile o riunioni improduttive.

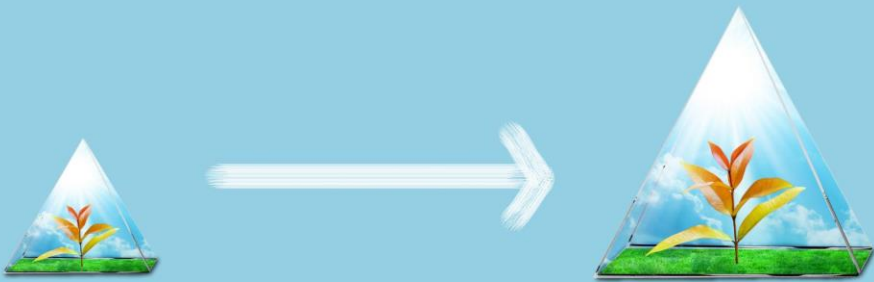
## MURA - Spreco di incoerenza

Mura è lo spreco nascosto causato da variabilità irregolare nei processi, come flussi di lavoro inconsistenti e qualità variabile.

## MURI - Spreco di irragionevolezza

Muri indica sovraccarico distruttivo di persone o sistemi oltre la capacità, causando burnout e errori frequenti.

# Collegamento tra sprechi e approcci Toyota vs Occidentale



## Interconnessione dei tre sprechi

MURI, MURA e MUDA formano un circolo vizioso che peggiora l'efficienza del team di lavoro.

## Esempio pratico di sprechi

Un team sovraccarico produce lavoro irregolare e spreco visibile, aumentando i tempi di rework.

## Approccio Occidentale tradizionale

Si concentra solo sul MUDA, riducendo i costi evidenti con ottimizzazione locale limitata.

## Approccio Toyota sistemico

Elimina MURI e MURA prima, facendo sparire MUDA e ottimizzando tutto il sistema.

# Applicazione pratica in Scrum e riduzione degli sprechi



## Eliminazione di MURI

Sprint di lunghezza sostenibile e Definition of Done realistica prevengono il sovraccarico del team.

## Eliminazione di MURA

Sprint di durata fissa e processi standardizzati assicurano una velocità di lavoro stabile nel tempo.

## Eliminazione di MUDA

Retrospective e Definition of Done aiutano a identificare sprechi e ridurre il lavoro inutile.

## Esempio pratico di progetto software

Ridurre il team a 7 persone e fissare sprint di 2 settimane diminuisce sprechi e aumenta valore.

# Stima, pianificazione e approccio story- driven

---

# Stima dimensionale relativa e serie di Fibonacci



## Stima dimensionale relativa in Scrum

La stima dimensionale relativa confronta due compiti per valutarne la dimensione in modo efficace nel tempo.

## Metodo della taglia dei cani

Utilizza la metafora delle taglie di cani per assegnare punti di complessità ai compiti da stimare.

## Serie di Fibonacci per stime

I valori di stima seguono la serie di Fibonacci, facilitando la percezione intuitiva delle differenze tra le attività.

## Differenze nette nelle stime

La mente umana percepisce meglio differenze nette come 5 o 8 rispetto a incrementi marginali come 5 o 6.

# Comprendere la storia e il ruolo del Product Owner



## Approccio Story-Driven

Scrum parte dalla comprensione della storia dietro il progetto, focalizzandosi sul perchè e per chi si lavora.

## Ruolo del Product Owner

Il Product Owner crea il backlog trasformando specifiche in storie che descrivono bisogni e problemi da risolvere.

## Collaborazione del Team

Il team scompone le storie in attività specifiche per diverse competenze, dal design allo sviluppo.

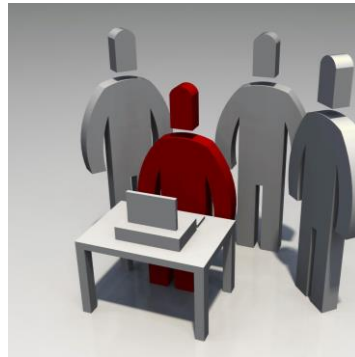
## Valutazione dei Risultati

Alla fine dello sprint, il Product Owner valuta se gli obiettivi delle storie sono stati raggiunti e valore consegnato.

# Esercizio pratico: progettare il primo Sprint

---

# Definizione dei ruoli e creazione del Product Backlog



## Ruoli chiave nel team Scrum

Product Owner definisce la visione e le priorità. Scrum Master facilita il processo e rimuove ostacoli. Development Team esegue il lavoro.



## Creazione del Product Backlog

Il Product Backlog contiene almeno 5 User Stories, descritte con formato chiaro e orientate ai benefici per l'utente finale.



## Esempi di User Stories

User Stories esemplificano bisogni degli utenti come ordinare online o gestire liste invitati per migliorare processi quotidiani.



# Pianificazione dello Sprint e identificazione degli impedimenti

## Durata e Obiettivo dello Sprint

Lo sprint dura 2 settimane con un obiettivo chiaro da dimostrare o consegnare al termine.

## Selezione e Stima User Stories

Si selezionano 2-3 User Stories dal backlog e si stima la loro complessità usando la scala 1,2,3,5,8,13.

## Identificazione degli Impedimenti

Individuare almeno tre ostacoli: tecnico, organizzativo e di comunicazione, e pianificare soluzioni efficaci.

# Conclusione

---

## Framework Scrum Potente

Scrum fornisce una struttura chiara con principi, ruoli e pratiche per guidare team agili verso il successo.

## Creazione di valore efficiente

Scrum aiuta i team a creare valore in modo efficiente e collaborativo, migliorando il prodotto continuamente.

## Miglioramento continuo

Applicare correttamente Scrum favorisce il miglioramento continuo e il successo dei prodotti.