



# Fondamenti e Principi delle Metodologie Agili

*Comprendere l'origine e  
l'applicazione dell'approccio Agile*

# Programma della Presentazione

- Storie di fallimenti e la necessità dell'Agile
- La nascita e l'evoluzione dell'Agile
- Il Manifesto Agile e i suoi valori fondamentali
- Nuove regole e management Agile
- Radici storiche e culturali di Scrum
- I principi Agile in pratica
- Attività di verifica e riflessione personale
- Sintesi e preparazione al modulo successivo





# Chi sono

**Davide Giansoldati**

Creativity Trainer e Agile Coach

Aiuto le aziende nei loro processi di innovazione

Docente a contratto per Università di Verona

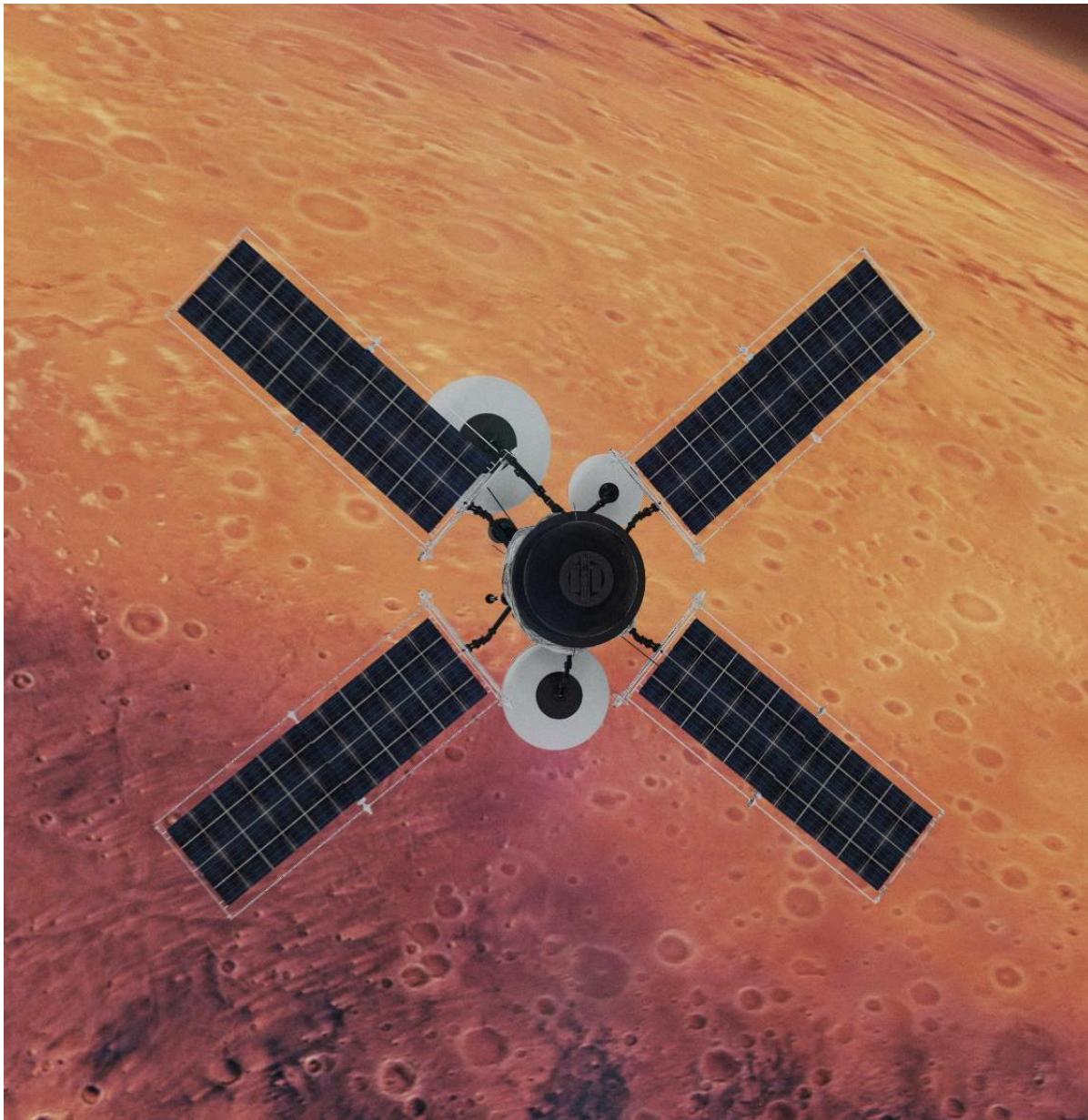
La prima startup nel 1998, l'ultima nel 2021

TEDx Speaker

Autore di «Il libro delle domande potenzianti»

# Storie di fallimenti e la necessità dell'Agile

---



## Due casi emblematici: Mars Climate Orbiter e FBI Sentinel

### Mars Climate Orbiter

Il Mars Climate Orbiter si schiantò per un errore di unità di misura tra team, causando una perdita da 125 milioni di dollari.

### Importanza delle metodologie Agili

Questa storia sottolinea come errori di comunicazione possono causare fallimenti costosi, dimostrando l'importanza delle metodologie Agili.

# Problemi di comunicazione e gestione nei progetti tradizionali



## **Metodi antiquati di gestione dati**

L'FBI utilizzava metodi cartacei obsoleti che causarono isolamento delle informazioni critiche.

## **Informazioni isolate e non condivise**

Le informazioni raccolte da diversi uffici rimasero isolate e non accessibili agli analisti.

## **Conseguenze sulla sicurezza**

La mancanza di condivisione dei dati impedì di unire i puntini per prevenire eventi tragici.

# Storia di un fallimento

**L'11 settembre si poteva evitare!**

Il sistema informativo dell'FBI si rivelò inadeguato.

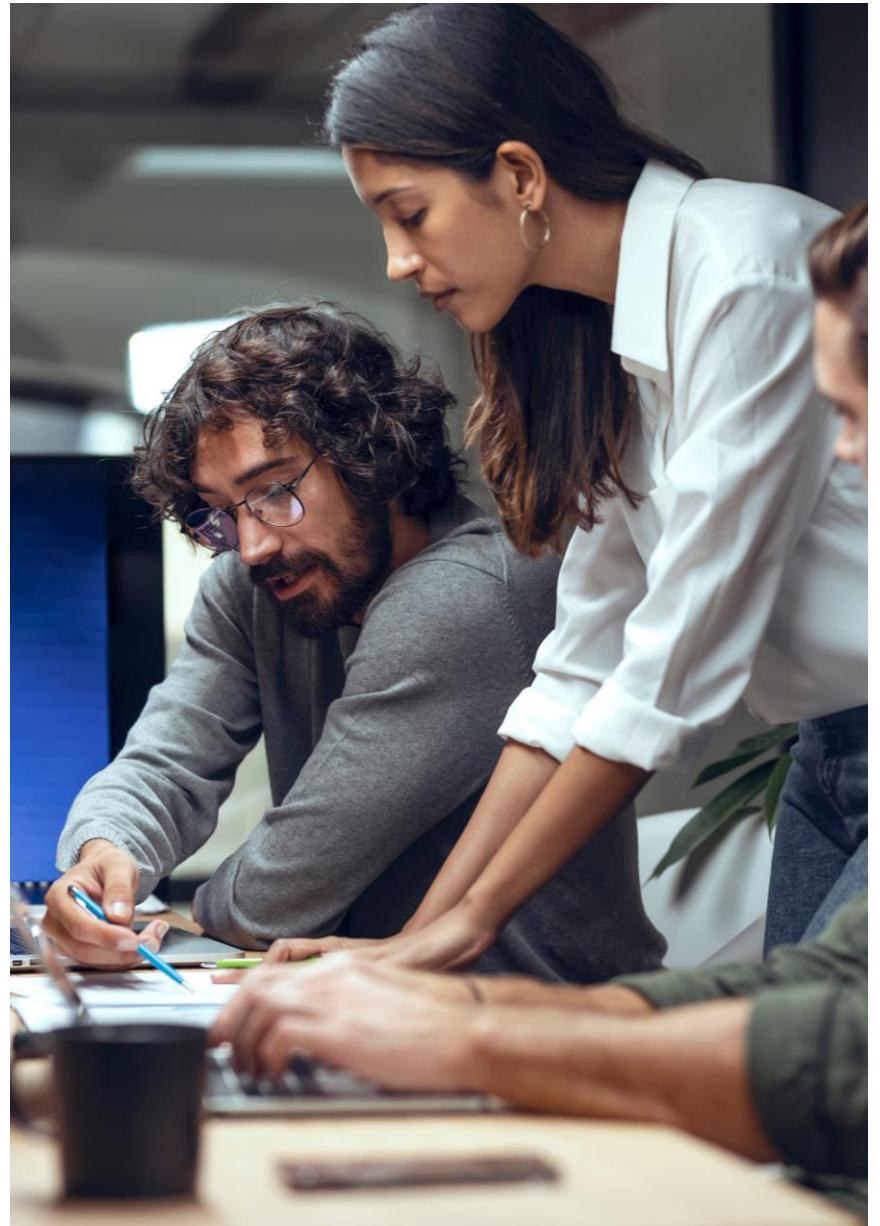
Si progettarono nuovi software per rispondere all'emergenza:

*Virtual case file*: 170 milioni di dollari => mai andato in produzione

*Sentinel*: 541 milioni di dollari => doveva essere pronto nel 2009

Nel 2010 vennero spesi su *Sentinel* 405 milioni, sviluppata più o meno metà delle funzionalità, sarebbero serviti altri 6/8 anni e almeno altri 350 milioni di dollari.

I bug emergevano più velocemente di quanto si riuscissero a risolvere.





## Cosa è successo dopo?

**Jeff Johnson** decise di portare lo sviluppo in casa, in seno al FBI e di adottare la metodologia agile.

Identificò 9 punti chiave e cambiò approccio passando a “ispeziona e adatta”: che significa “fai partire il progetto e controllalo con regolarità, verifica se stai andando nella direzione giusta o rischi di partire per la tangente, verifica se si possono migliorare delle attività.”

**La sfida:** completare Sentinel in un anno nel 2010 e con soli 20 milioni di dollari.

# Storia di un successo

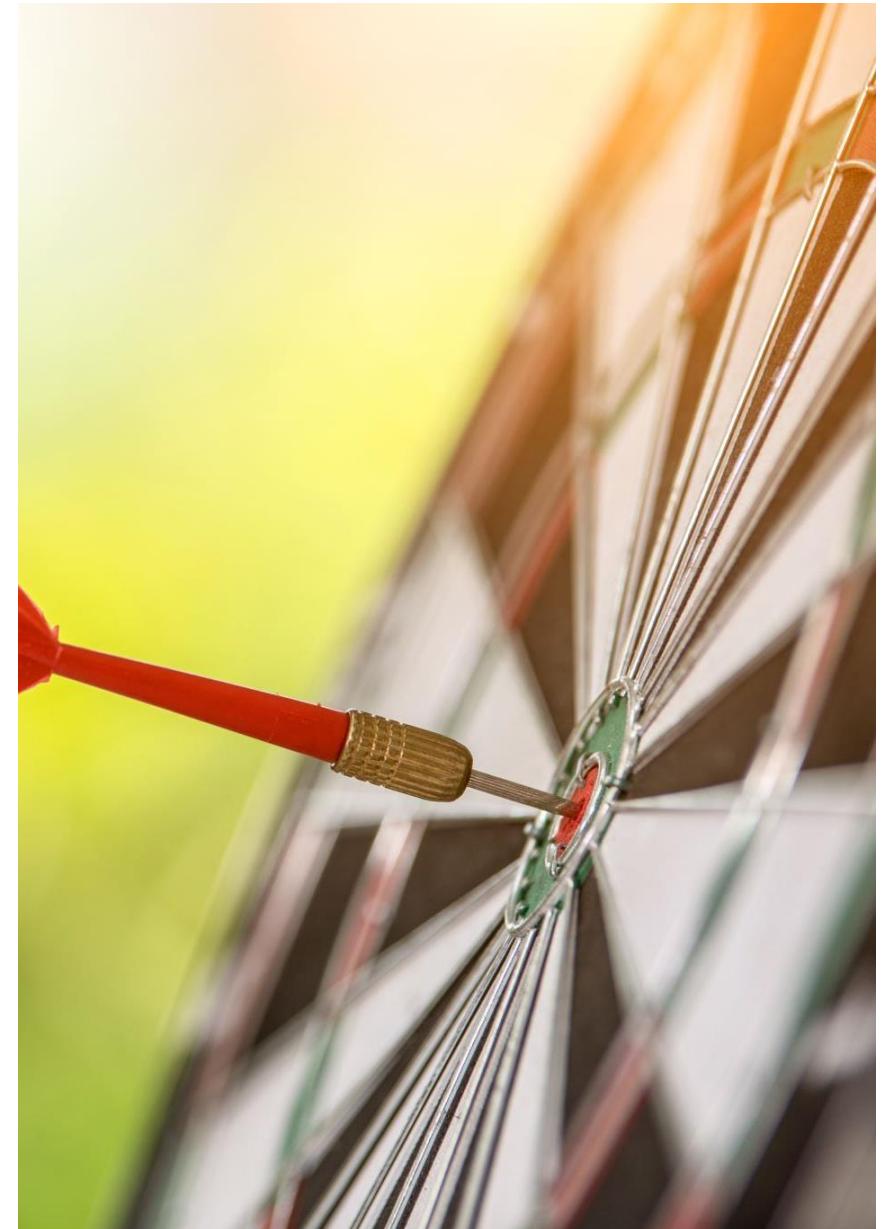
**Jeff Johnson organizza il lavoro in cicli di 2 settimane, chiamati sprint.**

Ad ogni inizio di sprint si stabilisce su cosa lavorare.

Ad ogni fine sprint si ragiona su cosa si può migliorare e sugli ostacoli da “asfaltare”

Risultato: il team di Sentinel ha aumentato la velocità di 3 volte.

Nel 2012 finalmente Sentinel viene lanciato grazie all’approccio agile!





# Tentativi di soluzione e costi dei progetti falliti

## **Virtual Case File costo e fallimento**

Il progetto Virtual Case File ha costato 170 milioni di dollari senza produrre codice operativo utile.

## **Sentinel e ritardi operativi**

Sentinel iniziò nel 2005 con un costo stimato di 541 milioni e doveva essere operativo entro il 2009, ma non lo è stato.

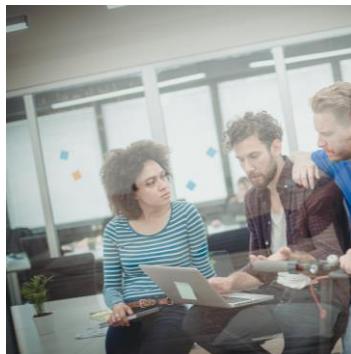
## **Costi e tempi aggiuntivi nel 2010**

Nel 2010, dopo aver speso 405 milioni e sviluppato metà del programma, servivano altri 6-8 anni e 350 milioni.

# La nascita e l'evoluzione dell'Agile

---

# Trasformazione Agile nel progetto Sentinel



## Adozione della metodologia Agile

L'FBI ha adottato l'Agile per completare il progetto Sentinel in meno tempo e con meno risorse, riducendo i costi a 20 milioni di dollari.

## Lancio e impatto di Sentinel

Il lancio di Sentinel nel 2012 ha trasformato significativamente il lavoro dell'FBI, migliorando efficienza e gestione dei dati.

## Identificazione precoce degli errori

L'Agile consente di identificare e correggere rapidamente errori, evitando costose catastrofi finanziarie e operative.

# Origini storiche: dal metodo Waterfall ai limiti del modello tradizionale

## Metodo Waterfall tradizionale

Il modello Waterfall si basa su una sequenza rigida di fasi progettate per essere completate perfettamente una dopo l'altra.

## Origine dei diagrammi di Gantt

I diagrammi di Gantt, inventati nel 1910, furono utilizzati per la prima volta nella logistica militare durante la Prima Guerra Mondiale.

## Limiti del modello tradizionale

L'approccio Waterfall risultò inadatto per progetti complessi, causando fallimenti nonostante la qualità del lavoro.



# Fasi del modello Waterfall

## **Analisi dei requisiti**

Documentare accuratamente tutte le funzionalità e i bisogni che il sistema deve soddisfare.

## **Progettazione dell'architettura**

Definire e progettare l'architettura completa del sistema software prima dello sviluppo.

## **Implementazione del codice**

Scrivere e sviluppare tutto il codice necessario per realizzare il sistema progettato.

## **Testing e rilascio**

Testare l'intero sistema per garantire qualità e rilasciare il prodotto finito agli utenti.

## **Manutenzione continua**

Gestire bug e richieste di cambiamento per mantenere il sistema efficiente e aggiornato.

# Il Manifesto Agile e i suoi valori fondamentali

---

# La creazione del Manifesto Agile e i suoi protagonisti

## Evento Snowbird 2001

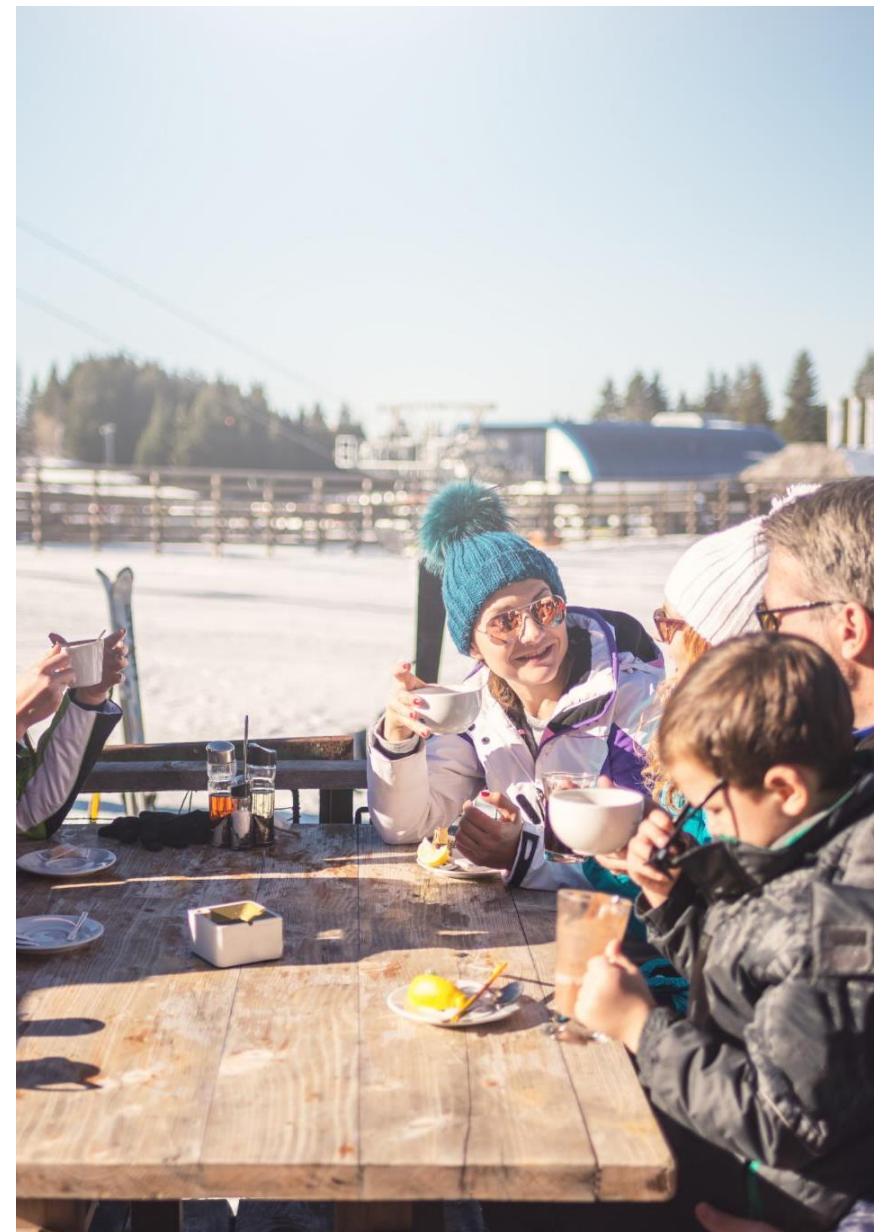
Nel febbraio 2001, 17 sviluppatori si riunirono a Snowbird, Utah per discutere nuove metodologie software.

## Protagonisti principali

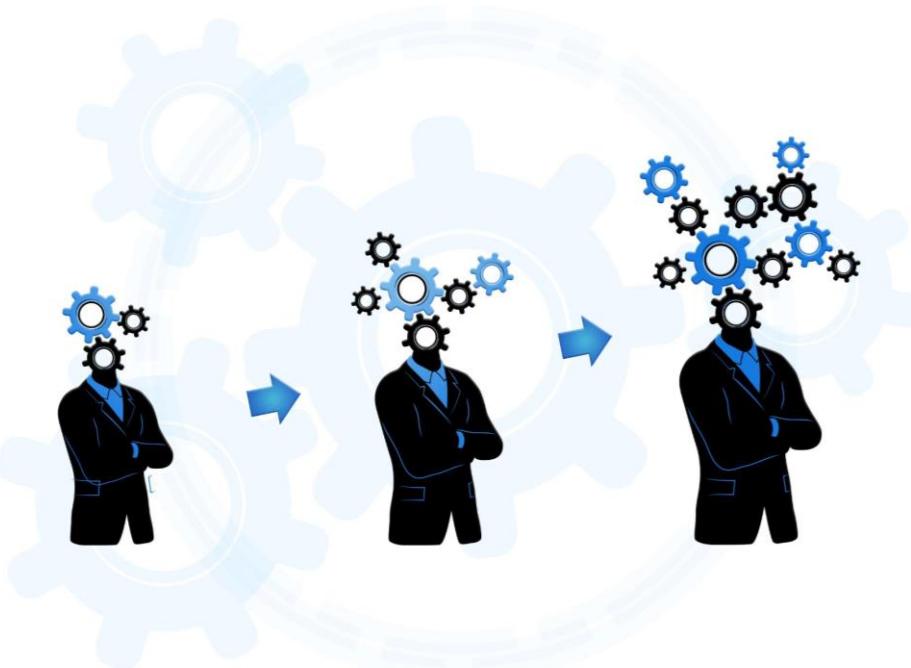
Figure chiave come Kent Beck, Jeff Sutherland e Martin Fowler guidarono la creazione del Manifesto Agile.

## Il Manifesto Agile

In soli tre giorni, venne scritto un documento di 68 parole che rivoluzionò il lavoro nel software.



# I quattro valori del Manifesto Agile: esempi pratici



## **Individui e Interazioni**

Spotify ha rivoluzionato l'organizzazione creando squad autonome di 8-12 persone, incrementando l'agilità e la crescita.

## **Software Funzionante**

Amazon rilascia continuamente piccoli aggiornamenti con test automatizzati per garantire software sempre funzionante.

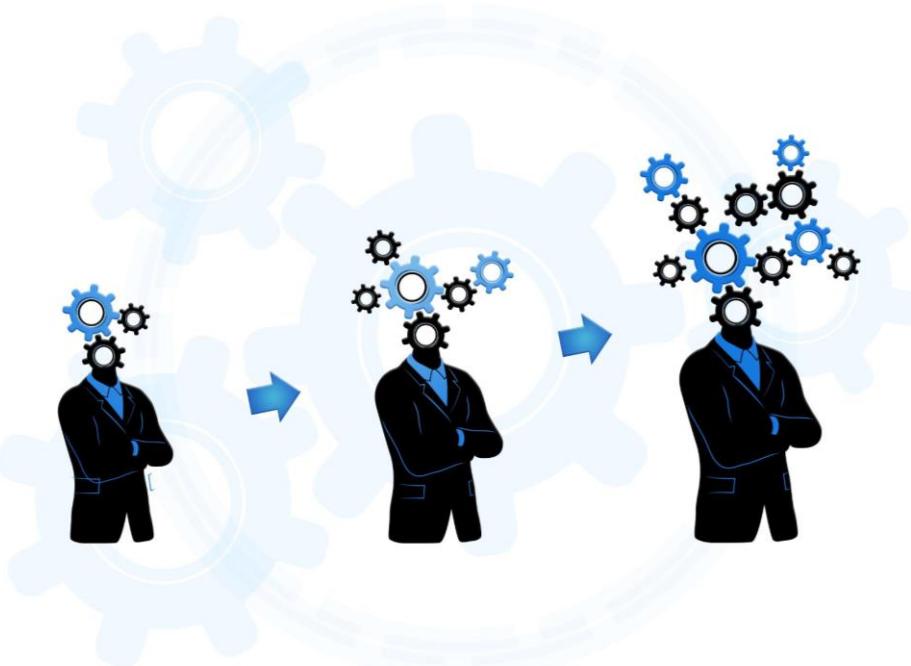
## **Collaborazione col Cliente**

ING Bank ha eliminato gerarchie tradizionali per creare journey team con clienti e IT per accelerare il time-to-market.

## **Rispondere al Cambiamento**

Netflix ha adattato il modello di business passando da DVD a streaming e produzione originale, seguendo il cambiamento.

# I quattro valori del Manifesto Agile: esempi pratici



## **Individui e Interazioni**

Le persone e le loro interazioni *prima* dei processi e degli strumenti

## **Software Funzionante**

Un software funzionante *prima* di una documentazione omnicomprensiva

## **Collaborazione col Cliente**

La collaborazione con il cliente *prima* della negoziazione di un contratto

## **Rispondere al Cambiamento**

La necessità di rispondere al cambiamento *prima* di seguire un piano

Nuove regole e  
management

Agile

---

# Critica al management tradizionale e nuove regole

## **Limiti del management tradizionale**

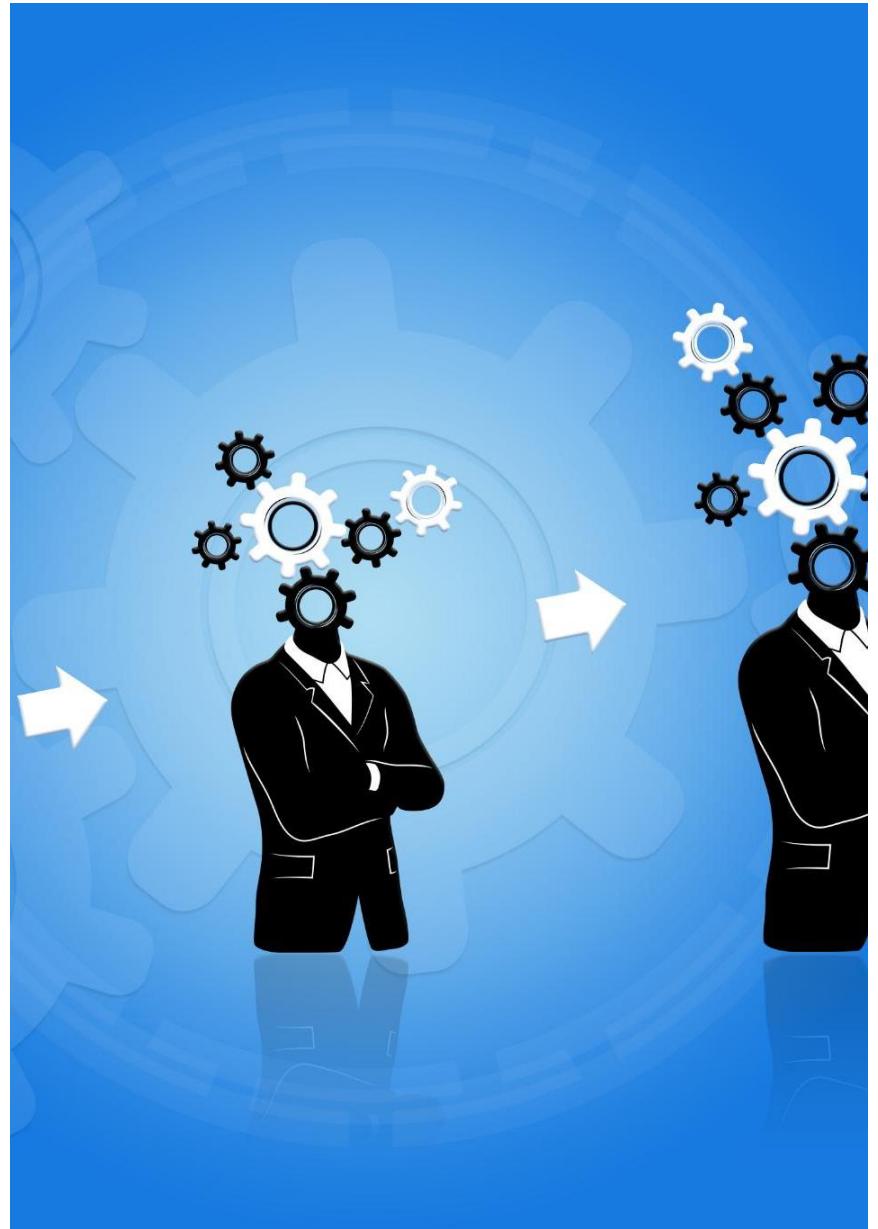
Il management tradizionale enfatizza controllo e prevedibilità, sprecando risorse in pianificazioni rigide e cambiamenti evitati.

## **Sfide del cambiamento rapido**

I cambiamenti rapidi e complessi nel mondo moderno rendono inefficace il management tradizionale nelle attività creative.

## **Principi del management agile**

Il management agile adotta regole semplici che valorizzano adattabilità, collaborazione e risposta rapida al cambiamento.



# Principi chiave del management Agile



## Importanza dell'adattamento

Pianificare è utile, ma è fondamentale ispezionare e adattare continuamente il piano.

## Rifiuto del comando e controllo

Seguire rigidamente il metodo comando e controllo porta al fallimento nei progetti Agile.

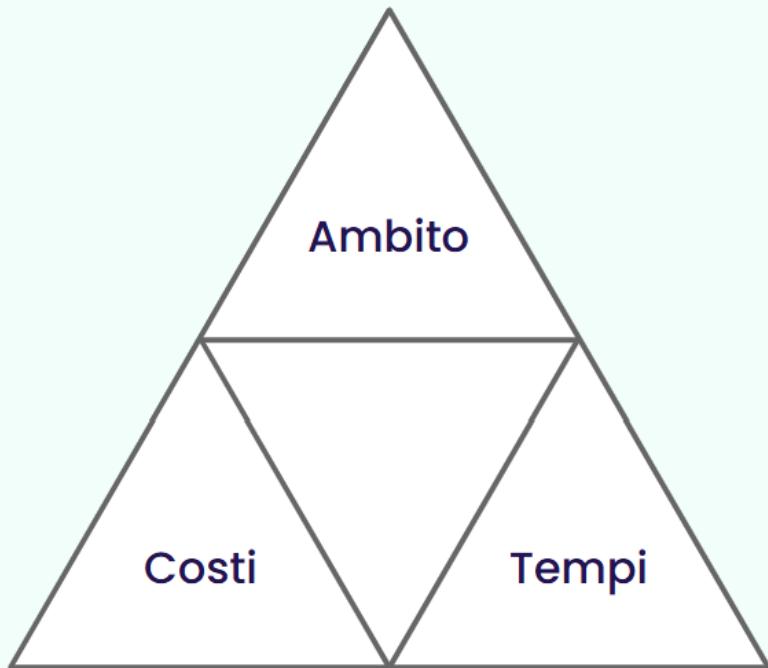
## Fallire rapidamente

Verificare velocemente gli errori aiuta a correggere la direzione verso il successo.

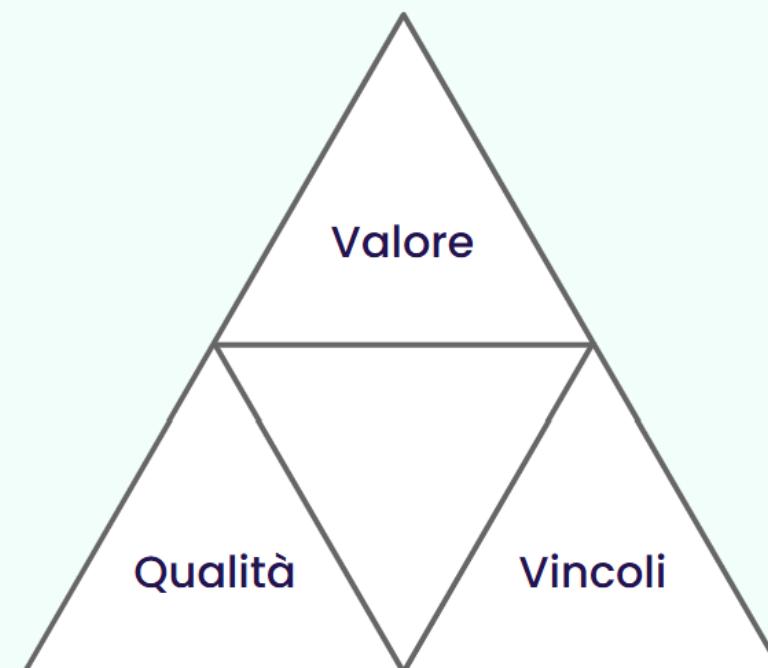
## Eliminazione degli sprechi

Il lavoro che non produce valore reale deve essere identificato ed eliminato.

# Un cambio di prospettiva



**Triangolo "di ferro" tradizionale**



**Triangolo Agile**

# I benefici dell'approccio Agile

Il cambiamento e l'incertezza fanno parte del processo

Le informazioni sono condivise in modo trasversale

Il feedback nel team è costante

I risultati sono migliorati sprint dopo sprint

Il cliente riceve continuamente valore

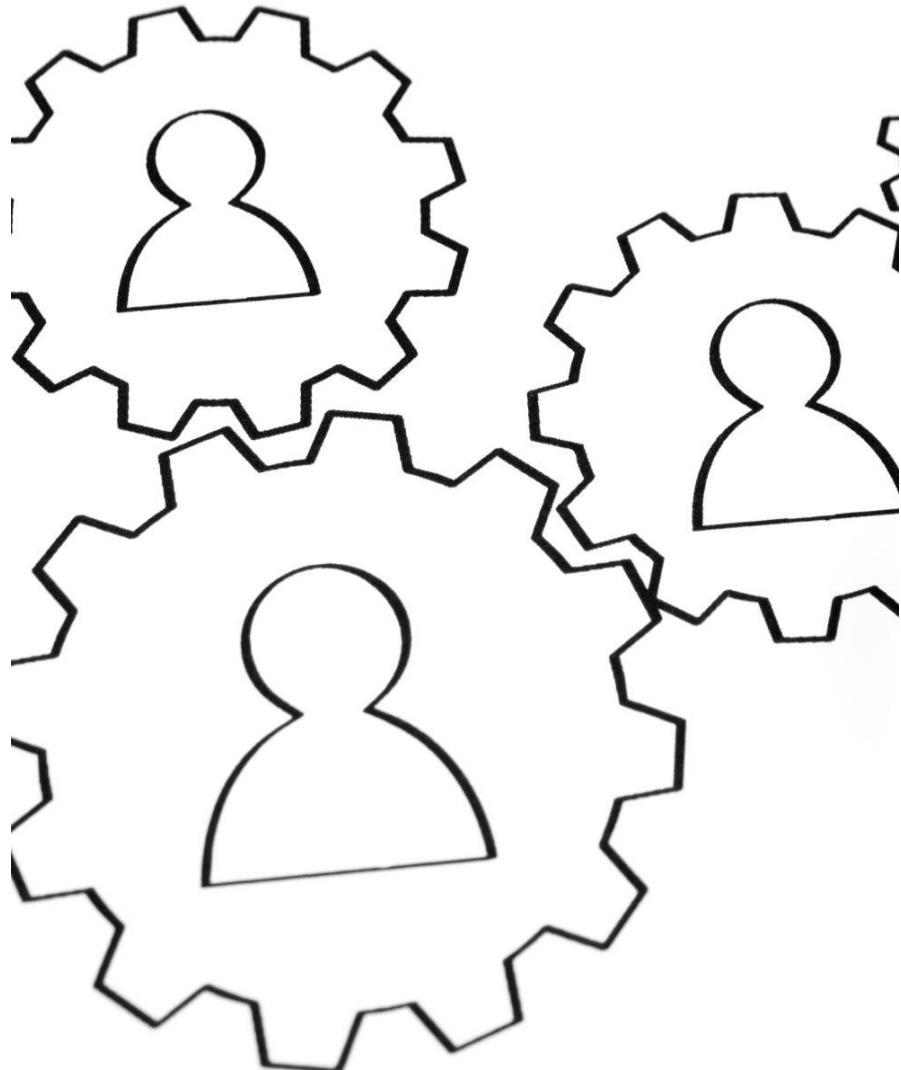
Il team trova il giusto ritmo di lavoro, sostenibile per un tempo indefinito, lavora ai massimi livelli di efficienza

Il cliente riceve le funzionalità di maggior valore per prime

Il team attraverso il confronto è costantemente motivato

Grazie al lavoro di squadra i problemi si risolvono più velocemente

Il cliente è al centro e la responsabilità è collettiva



# I team come sistemi complessi adattivi

## **Team come sistemi adattivi**

I team si comportano come sistemi complessi che reagiscono ai cambiamenti entrando in uno stato temporaneo di caos.

## **Processo di riorganizzazione**

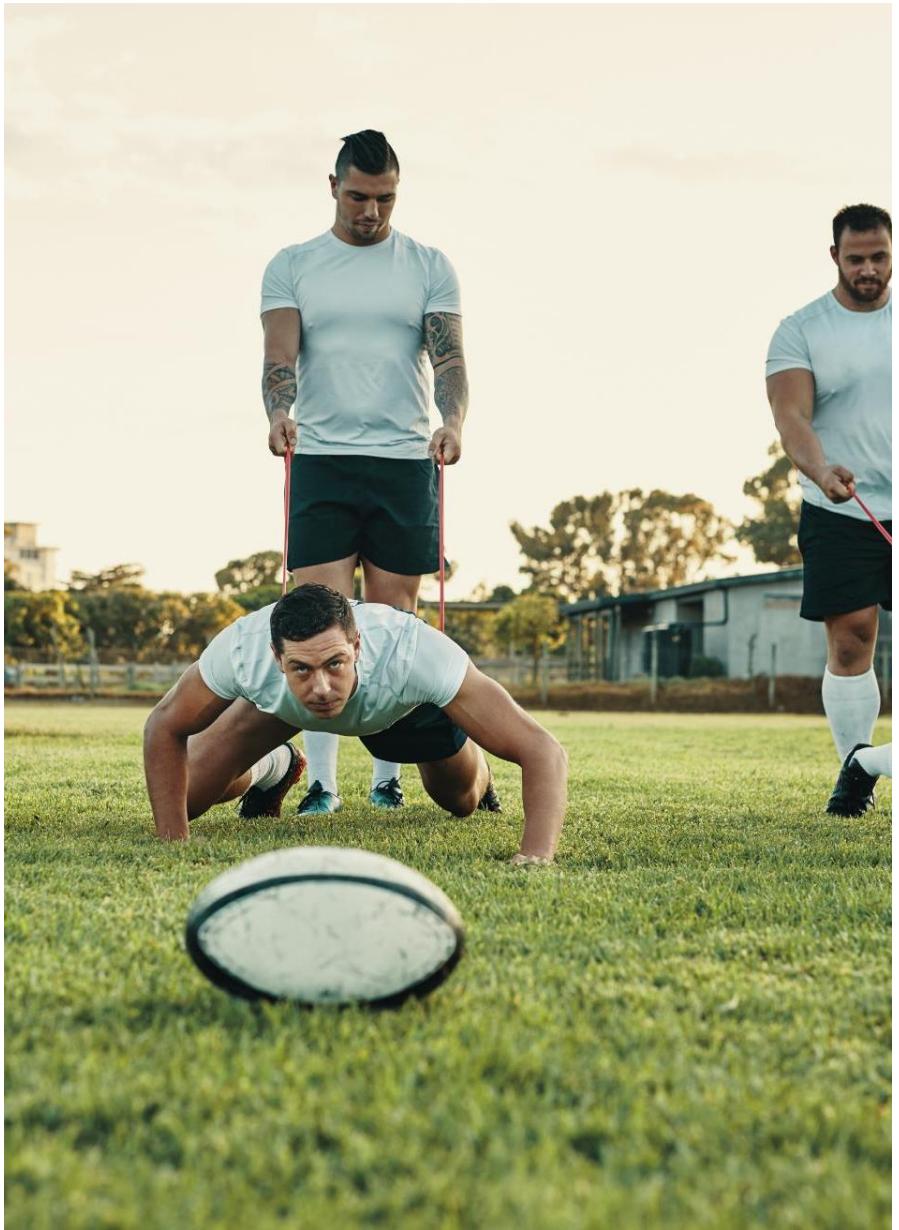
Dopo il caos iniziale, il team si riorganizza in un nuovo assetto più funzionale e migliorato rispetto al precedente.

## **Ruolo di Scrum**

Scrum fornisce regole che guidano i team attraverso i cambiamenti, portando a maggiore produttività e soddisfazione.

# Radici storiche e culturali di Scrum

---



# L'articolo "The New New Product Development Game" e la nascita di Scrum

## Articolo innovativo del 1986

L'articolo presentava studi su aziende innovative con processi di sviluppo flessibili e team autonomi.

## Metafora della squadra di rugby

Il lavoro di squadra veniva paragonato a una squadra di rugby che si muove come un'unica entità sul campo.

## Nascita di Scrum

Da questo articolo nacque Scrum, framework presentato nel 1995 per migliorare lo sviluppo agile dei prodotti.



# Il ciclo PDCA e l'influenza del Toyota Production System

## Il ciclo PDCA di Deming

Il ciclo PDCA significa pianificare, agire, controllare e adattare ed è fondamentale nella gestione della produzione moderna.

## Toyota Production System

Il TPS ha utilizzato il ciclo PDCA come base per diventare un metodo rivoluzionario nella produzione automobilistica.

## Scrum e adattamento continuo

Scrum applica il principio 'ispeziona e adatta' per gestire l'incertezza e incrementare la velocità dei progetti.

# L'influenza della cultura giapponese e il concetto di Shu Ha Ri



## Origini culturali di Scrum

Scrum nasce dalla cultura giapponese, simile all'Aikido, dove apprendimento avviene solo attraverso la pratica costante.

### Fase Shu

In questa fase si apprendono le regole e le forme e l'allievo le ripete senza modificarle per assimilare le basi.

### Fase Ha

Si iniziano a modificare le forme apprese per adattarle, iniziando a personalizzare la propria pratica.

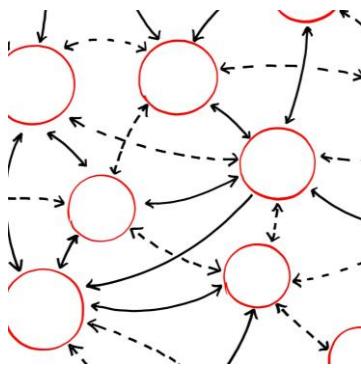
### Fase Ri

Si abbandonano le forme perché padroneggiate completamente, realizzando uno stile personale e fluido.

# I principi Agile in pratica

---

# Principi fondamentali e regola 80/20



## Consegna tempestiva di valore

La priorità è consegnare software di valore frequentemente ogni 2-4 settimane per soddisfare il cliente.

## Accoglienza dei cambiamenti

I cambiamenti anche tardivi sono accettati, riducendo il costo e l'impatto grazie a iterazioni brevi di 2 settimane.

## Regola 80/20 nel software

L'80% del valore nel software risiede nel 20% delle caratteristiche, focalizzandosi su ciò che conta davvero.



# Team auto-organizzati e motivazione

## Motivazione degli individui

Progetti efficaci nascono da individui motivati che ricevono supporto e un ambiente adeguato.

## Team auto-organizzati

I team auto-organizzati favoriscono l'emergere di architetture, requisiti e design di alta qualità.

## Cooperazione VS Collaborazione

Si ha cooperazione quando il lavoro prodotto consiste nella somma degli sforzi di varie persone di un team.

Si ha collaborazione quando un team lavora insieme facendo proprio il lavoro degli altri per produrre qualcosa di più grande.

L'approccio agile si basa sulla piena collaborazione.





## Time boxing

Il tempo rappresenta il vincolo primario.  
A ogni processo/attività va assegnato un  
periodo di tempo prefissato.



# Sviluppo iterativo

Il modello iterativo è molto più flessibile ed efficace dei tradizionali metodi di sviluppo, proprio nel garantire che le richieste di cambiamento da parte del cliente facciano parte del progetto.

Lo sviluppo iterativo in sprint, consente alle persone di correggere il tiro grazie ai feedback del cliente.

I prodotti realizzati si adattano meglio al contesto di business finale.

# La leadership di servizio

- Ascolto
- Empatia
- Capacità di cura
- Consapevolezza
- Persuasione
- Concettualizzazione
- Intuizione
- Stewardship (impegno a servire gli altri)
- Impegno a far crescere gli altri
- Costruzione del team





## Team di generalisti-specialisti

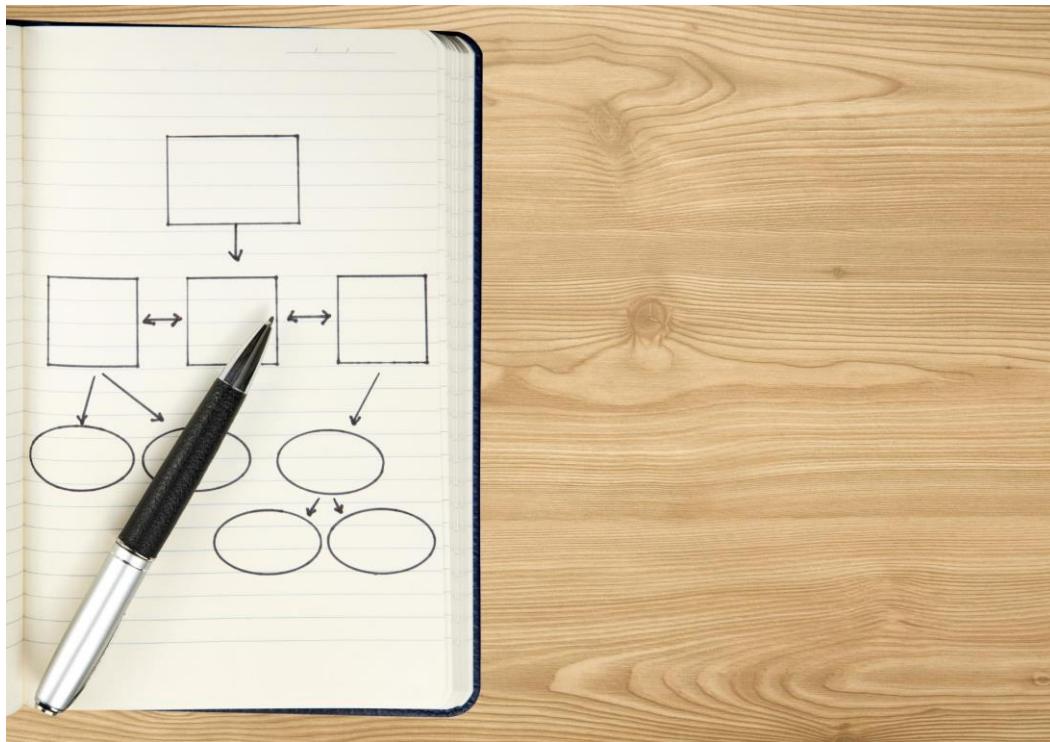
Vantaggi dei Team Interfunzionali:

- Processo decisionale più rapido
- Migliore comunicazione
- Orientamento all'obiettivo
- Responsabilità collettiva
- Innovazione continua

**Attività di  
verifica e  
riflessione  
personale**

---

# Esercizio pratico: analisi del proprio metodo di lavoro



## Pianificazione vs Adattamento

Analizza quanto tempo dedichi alla pianificazione iniziale rispetto all'adattamento continuo durante i progetti.

## Feedback e Comunicazione

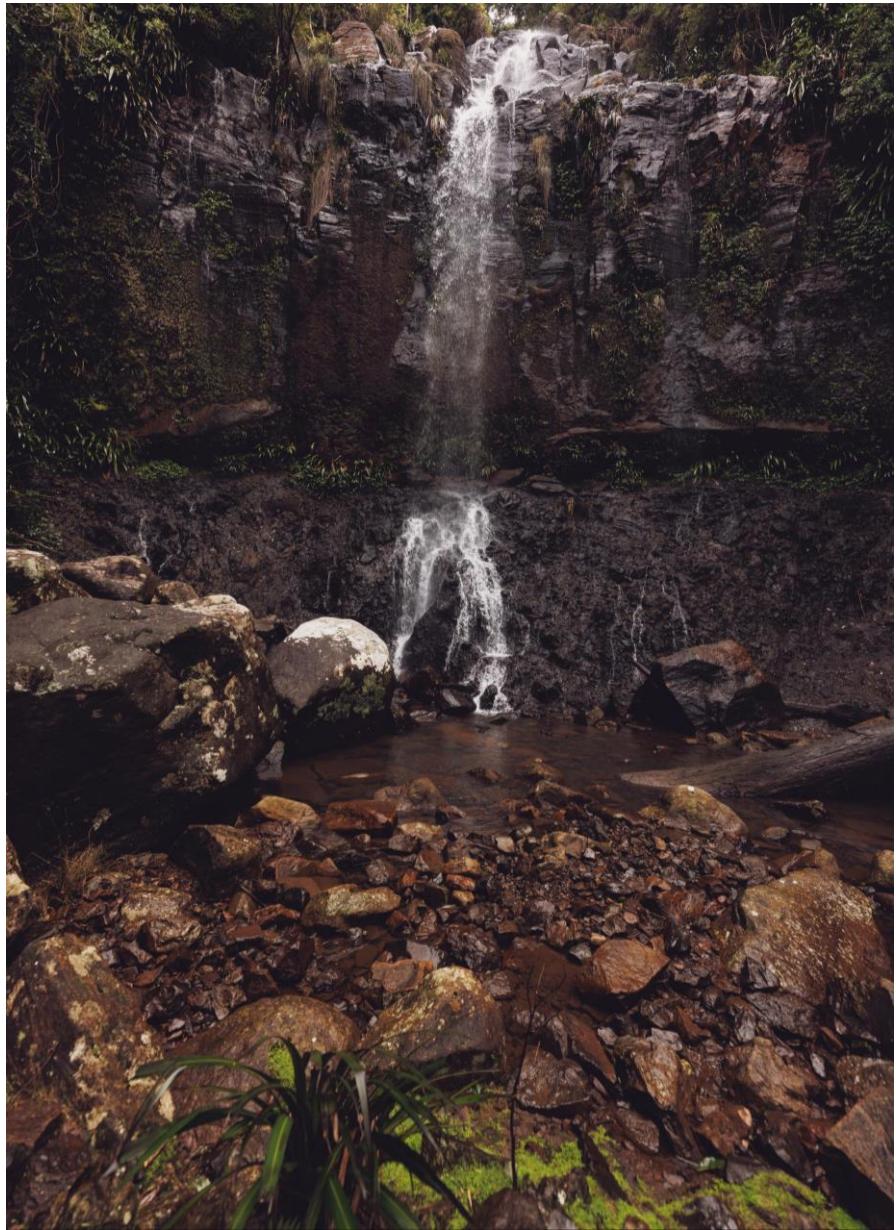
Valuta la frequenza e la natura del feedback ricevuto e la comunicazione con colleghi e stakeholder.

## Valore e Spreco

Identifica attività che non aggiungono valore e valuta il tempo speso in documentazione e processi inutili.

## Controllo vs Fiducia

Rifletti se preferisci il controllo totale o affidarti ai collaboratori per gestire l'incertezza.



# Identificazione di opportunità di miglioramento

## Comportamenti waterfall da cambiare

Individua due pratiche tradizionali in cascata che rallentano il progetto e possono essere migliorate per agilità.

## Principi Agile da applicare

Applica immediatamente due principi Agile per aumentare la flessibilità, la collaborazione e la consegna rapida di valore.

## Cambiamento concreto imminente

Definisci un cambiamento pratico che puoi implementare la prossima settimana per migliorare il flusso di lavoro.

Sintesi e  
preparazione al  
modulo  
successivo

---

# Riepilogo dei punti chiave del modulo



## **Limiti dei metodi tradizionali**

I metodi tradizionali falliscono in progetti complessi e incerti come Sentinel e Mars Orbiter.

## **Origine del Manifesto Agile**

Il Manifesto Agile nasce da necessità pratiche e non da teorie astratte.

## **Quattro valori Agile**

I quattro valori Agile ribaltano l'approccio tradizionale al lavoro favorendo flessibilità e collaborazione.

## **Nuovo management e Scrum**

Il nuovo management abbandona comando e controllo per fiducia, adattamento e miglioramento continuo ispirato alla filosofia giapponese.



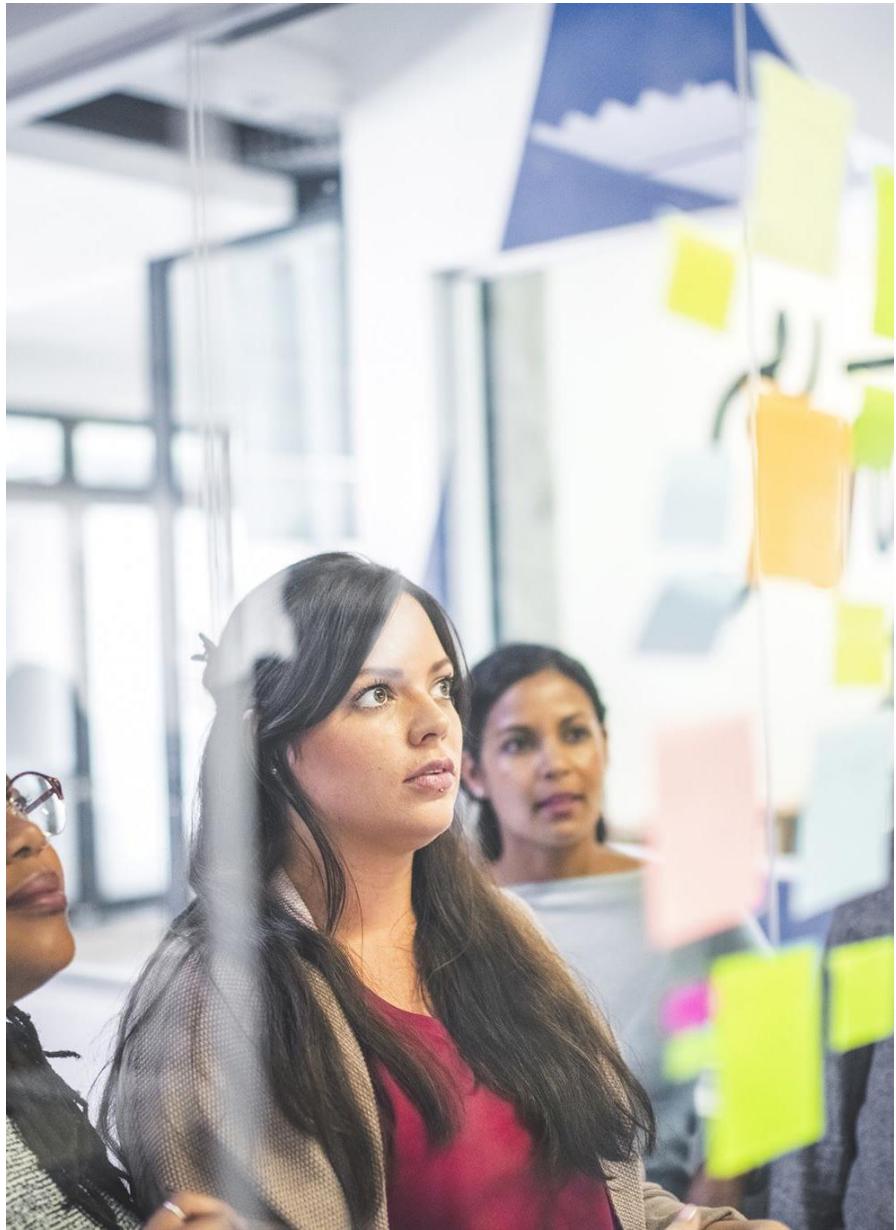
# Il principio del "Fatto a Metà"

## **Importanza del completamento**

In Agile, completare un'attività è più importante che iniziare molte senza terminarle.

## **Valore del lavoro finito**

Il valore reale risiede nel lavoro completato, non in ciò che è solo iniziato o parziale.



# Anticipazione dei contenuti del prossimo modulo

## Ruoli di Scrum

Il modulo spiega i ruoli fondamentali di Scrum: Product Owner, Scrum Master e Development Team.

## Eventi Scrum

Si approfondiscono gli eventi Scrum come Sprint, Daily Scrum e Retrospective per gestire i progetti.

## Efficienza nei progetti

Il concetto di "fare il doppio in metà tempo" viene esplorato come vantaggio principale di Scrum.

# Conclusione

## Fondamenti di Agile

Le metodologie Agile nascono per offrire maggiore flessibilità e adattabilità nei progetti.

## Collaborazione e team

Agile favorisce la collaborazione continua e la comunicazione efficace tra i membri del team.

## Implementazione futura

La comprensione di Agile prepara a un'implementazione di successo nei progetti futuri.