

# Metodologie di sviluppo software

## Metodologia Agile

prof. Leonardo Essam Dei Rossi

ITT "M. Buonarroti" - Trento (TN)

Anno scolastico 2025/2026

# Avvertenza

## **Attenzione!**

Queste slide sono state tradotte da quelle originali ([link](#)) e potrebbero contenere errori di traduzione.

# Indice

## 1 Che cos'è la metodologia Agile?

- Punti chiave

## 2 Perché scegliere Agile?

## 3 Le fasi di Agile

- Phase #1: Requirements
- Phase #2: Design
- Phase #3: Development
- Phase #4: Testing
- Phase #5: Deployment
- Phase #6: Review

## 4 Agile vs. Waterfall

# Indice dei contenuti

## 1 Che cos'è la metodologia Agile?

- Punti chiave

## 2 Perché scegliere Agile?

## 3 Le fasi di Agile

- Phase #1: Requirements
- Phase #2: Design
- Phase #3: Development
- Phase #4: Testing
- Phase #5: Deployment
- Phase #6: Review

## 4 Agile vs. Waterfall

## Che cos'è la metodologia Agile? (1)

La metodologia Agile è un approccio che suddivide il lavoro in fasi, ponendo l'accento sulla consegna e sul miglioramento continui. Agile offre vantaggi ai team consentendo una pianificazione adattiva, un'esecuzione rapida e una valutazione costante, che portano a risultati più reattivi e di successo.

L'approccio tradizionale "a cascata" prevede che ogni team lavori a un progetto separatamente, per poi passarlo al gruppo successivo. L'approccio agile, invece, si avvale di team collaborativi e interfunzionali, ponendo l'accento sulla comunicazione aperta, il lavoro di squadra, l'adattabilità e la fiducia.

## Che cos'è la metodologia Agile? (2)

Sebbene il responsabile del progetto o il product owner in genere stabilisca le priorità del lavoro da consegnare, è il team a decidere come il lavoro verrà svolto, auto-organizzandosi attorno a compiti e incarichi specifici. L'approccio Agile non è definito da una serie di cerimonie o da tecniche di sviluppo specifiche.

Al contrario, Agile è un insieme di metodologie che dimostrano un impegno verso cicli di feedback rapidi e il miglioramento continuo. Il Manifesto Agile originale non prescriveva iterazioni di due settimane né una dimensione ideale del team.

## Punti chiave

- Agile è un approccio flessibile e iterativo alla gestione dei progetti che pone l'accento sulla collaborazione, la consegna continua e l'adattabilità;
- Dà più importanza alle persone, al feedback dei clienti e alle soluzioni funzionanti che ai processi rigidi e alla documentazione;
- Teams tailor Agile practices to their needs, blending frameworks like Scrum and Kanban for optimal results;
- Experiment with Agile practices and regularly review their effectiveness to foster continuous improvement and team collaboration.

# Indice dei contenuti

## 1 Che cos'è la metodologia Agile?

- Punti chiave

## 2 Perché scegliere Agile?

## 3 Le fasi di Agile

- Phase #1: Requirements
- Phase #2: Design
- Phase #3: Development
- Phase #4: Testing
- Phase #5: Deployment
- Phase #6: Review

## 4 Agile vs. Waterfall

## Perché scegliere Agile? (1)

I team scelgono metodologie agili per poter rispondere rapidamente ai cambiamenti del mercato o al feedback dei clienti, senza compromettere i piani di un anno intero. Una pianificazione "giusta" e la consegna in piccoli incrementi frequenti consentono al team di raccogliere feedback su ogni modifica e di integrarli nei piani futuri a costi minimi.

But it's not just a numbers game-first and foremost, it's about people. As described by the Agile Manifesto, authentic human interactions are more important than rigid processes.

## Perché scegliere Agile? (2)

Collaborare con clienti e colleghi è più importante degli accordi predefiniti. Fornire una soluzione funzionante al problema del cliente è più importante che fornire una documentazione eccessivamente dettagliata.

Un team agile si unisce sotto una visione condivisa, per poi realizzarla nel modo che ritiene più efficace. Ogni team definisce i propri standard di qualità, usabilità e completezza.

# Indice dei contenuti

## 1 Che cos'è la metodologia Agile?

- Punti chiave

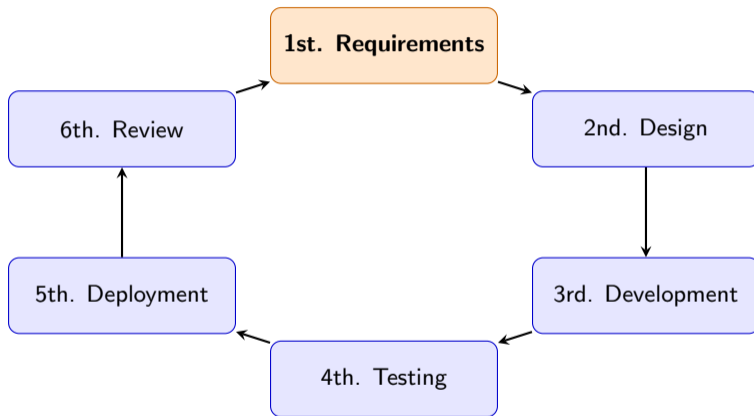
## 2 Perché scegliere Agile?

## 3 Le fasi di Agile

- Phase #1: Requirements
- Phase #2: Design
- Phase #3: Development
- Phase #4: Testing
- Phase #5: Deployment
- Phase #6: Review

## 4 Agile vs. Waterfall

# Le fasi di Agile



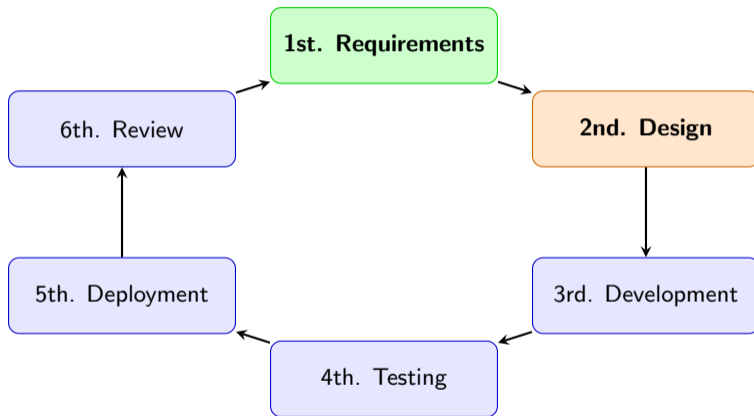
## Phase #1: Requirements

Nel modello Waterfall, i requisiti sono un contratto "firmato e sigillato". Nel modello Agile, sono oggetto di conversazione.

Ci allontaniamo dalla documentazione rigida per orientarci verso un **Product Backlog** – un elenco prioritario delle esigenze degli utenti.

Questa fase si concentra sulla comprensione dell'intento alla base di una funzionalità, piuttosto che sulla semplice verifica delle specifiche tecniche. Concentrandosi sulle "User Story", il team rimane orientato al risultato, consentendo ai requisiti di evolversi in base alle condizioni di mercato o al feedback degli utenti.

# Le fasi di Agile

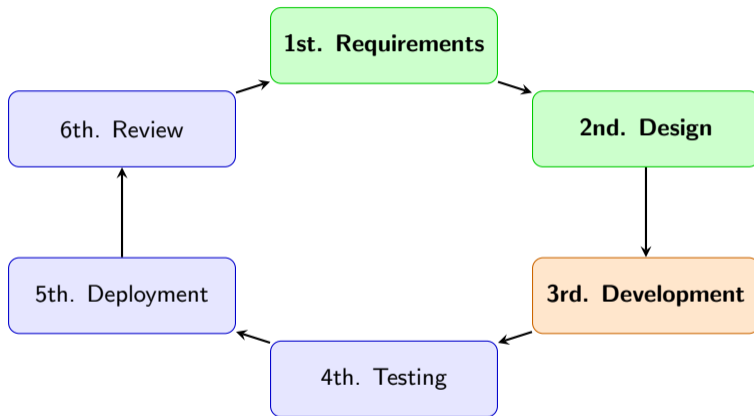


## Phase #2: Design

**La progettazione agile rifiuta la filosofia "Big Design Up Front" (BDUF).**

Al contrario, privilegia un'architettura evolutiva. Il team dedica il tempo necessario per garantire che il sistema sia scalabile e intuitivo, ma non cerca di risolvere ogni possibile problema che potrebbe presentarsi tra tre anni. L'obiettivo è creare un progetto flessibile che consenta modifiche tecniche senza dover scartare mesi di pianificazione.

# Le fasi di Agile



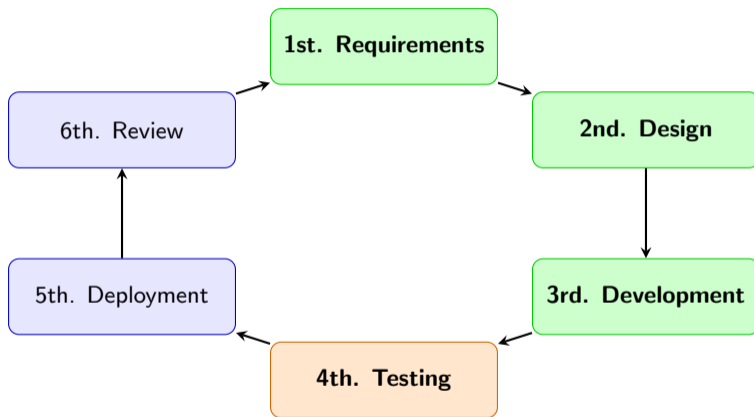
## Phase #3: Development

**Questa è la "sala macchine" del ciclo.**

Lo sviluppo è suddiviso in brevi fasi mirate (sprint). Invece di costruire l'intero sistema in una volta sola, gli sviluppatori si concentrano sulla creazione di un Incremento di **Prodotto Potenzialmente Rilasciabile (PSPI)**. Questo garantisce che, anche se il progetto venisse interrotto domani, il team avrebbe comunque a disposizione un software funzionante da mostrare come frutto del proprio lavoro.

Si tratta di un approccio che privilegia "un software funzionante rispetto a una documentazione esaustiva".

# Le fasi di Agile

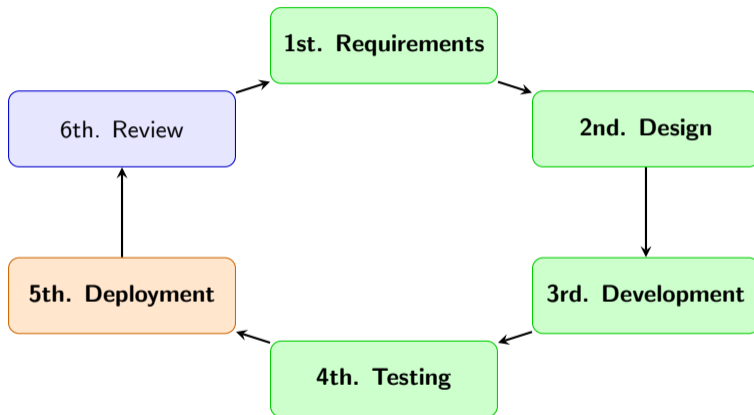


## Phase #4: Testing

**In un contesto Agile, i test non rappresentano una "tappa" finale al progetto, ma sono parte integrante del processo di sviluppo stesso.**

Implementando il *Continuous Testing* [2] e suite automatizzate, il team si assicura che le nuove funzionalità non compromettano quelle esistenti. Questo approccio "shift-left" permette di individuare i bug mentre il codice è ancora fresco nella mente dello sviluppatore, riducendo drasticamente i costi e i tempi di riparazione.

# Le fasi di Agile

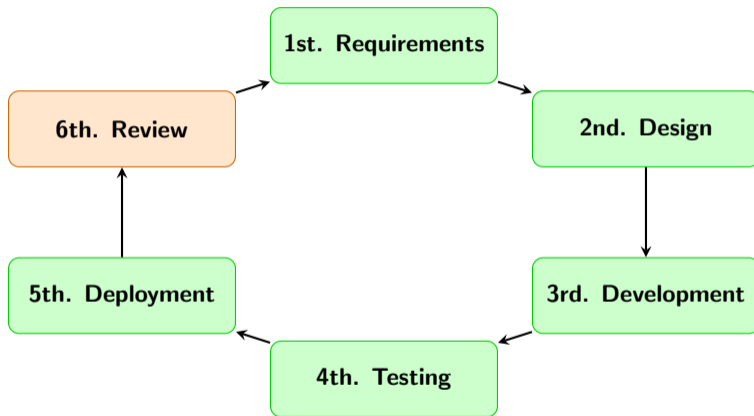


## Phase #5: Deployment

**La fase di implementazione è il momento in cui "la teoria incontra la pratica".**

In un ambiente Agile maturo, questo spesso utilizza pipeline di *Continuous Delivery* [1]. L'obiettivo è quello di mettere il software nelle mani degli utenti reali il più rapidamente e in sicurezza possibile. Ciò consente all'azienda di ottenere valore immediatamente e fornisce al team dati reali su come la funzionalità viene effettivamente utilizzata.

# Le fasi di Agile



## Phase #6: Review

La fase finale è forse la più critica per la filosofia Agile di "ispezione e adattamento":

- **La Sprint Review:**

coinvolge le parti interessate per verificare che il team stia realizzando la cosa giusta;

- **La retrospettiva:**

permette al team di guardarsi dentro per assicurarsi di star costruendo nel modo giusto.

Non si tratta solo di una conclusione; è il ponte verso il ciclo successivo, che garantisce che ogni errore diventi una lezione per l'iterazione successiva.

# Indice dei contenuti

- 1 Che cos'è la metodologia Agile?
  - Punti chiave
- 2 Perché scegliere Agile?
- 3 Le fasi di Agile
  - Phase #1: Requirements
  - Phase #2: Design
  - Phase #3: Development
  - Phase #4: Testing
  - Phase #5: Deployment
  - Phase #6: Review
- 4 Agile vs. Waterfall

## Agile vs. Waterfall (1)

La metodologia Agile è stata inizialmente adottata dai team di sviluppo software, che sono passati dal tradizionale approccio sequenziale a cascata a un metodo che garantisce un feedback costante e la possibilità di apportare modifiche durante tutto il ciclo di vita dello sviluppo.

La gestione agile dei progetti adotta un approccio iterativo allo sviluppo, creando diverse fasi incrementali con intervalli di feedback regolari. Questo favorisce l'adattabilità, poiché il team può apportare modifiche durante tutto il processo di sviluppo del prodotto, anziché essere vincolato a un percorso lineare. Consente inoltre rilasci regolari e di grande impatto, che permettono ai team di ottenere una serie di successi nel tempo.

## Agile vs. Waterfall (2)

Le release iterative offrono a un team molteplici opportunità per:

- Adattarsi alle circostanze mutevoli, dai requisiti scoperti di recente a un lavoro bloccato;
- Raccogli feedback dalle parti interessate durante il processo e apporta modifiche iterative in modo reattivo, senza la pressione di una scadenza di consegna finale;
- Instaurare relazioni e connessioni tra i diversi ruoli, facilitando così la comunicazione e la connessione tra le persone.

La metodologia Agile consente ai team di essere più resilienti ai cambiamenti che inevitabilmente si verificano durante un progetto.

# Riferimenti e approfondimenti

- [1] Wikipedia contributors. Continuous delivery — Wikipedia, The Free Encyclopedia. [Online; accessed 25-February-2026]. 2025. URL: [https://en.wikipedia.org/w/index.php?title=Continuous\\_delivery&oldid=1322843384](https://en.wikipedia.org/w/index.php?title=Continuous_delivery&oldid=1322843384).
- [2] Wikipedia contributors. Continuous testing — Wikipedia, The Free Encyclopedia. [Online; accessed 25-February-2026]. 2025. URL: [https://en.wikipedia.org/w/index.php?title=Continuous\\_testing&oldid=1329392723](https://en.wikipedia.org/w/index.php?title=Continuous_testing&oldid=1329392723).