

# BUILDING WORKING SOFTWARE EASY TO CHANGE

**CORSO DI**

**FORMATORE: NICOLA MORETTO**



Contattaci a [info@qmates.tech](mailto:info@qmates.tech)

# DESCRIZIONE

Durante il corso illustreremo e metteremo in pratica le tecniche di programmazione Agile derivanti dalla metodologia eXtreme Programming (XP). Comprenderemo e esploreremo le tecniche dell'XP con una particolare attenzione ai processi collaborativi, alla qualità del codice e alla flessibilità architetturale. Il corso è indipendente dalla tecnologia, quindi concepito per tutti gli sviluppatori, qualsiasi sia la loro specializzazione o il linguaggio che utilizzano.

Le tecniche che insegniamo all'interno di questo corso hanno l'obiettivo di:

- aumentare la collective ownership: un software scritto in modalità collaborativa è più facile da modificare, ottenendo così un processo di sviluppo semplice e a basso tasso di coordinazione tra i membri del team. Diventa, quindi, più facile includere nuovi membri accelerando la comprensione del software e di conseguenza la sua modifica;
- aumentare la cadenza del rilascio di valore: nel tempo, il numero crescente di feature nel prodotto concorre a far aumentare la complessità del software. Grazie alle pratiche di eXtreme Programming è possibile mantenere bassa la complessità ottenendo una cadenza costante di delivery;
- abbassare il rischio di riscrittura del software: con l'inserimento continuativo di nuovo codice e in assenza di un metodo di controllo della complessità, il rischio di dover riscrivere il software per ritornare ad una situazione normale di delivery è sempre più alto. Le tecniche di simple design permettono di tenere sotto controllo la complessità;

# MODALITÀ

Il corso sarà quanto più interattivo possibile, proponendo un esercizio di programmazione per ogni concetto spiegato (circa ogni mezza giornata di corso). Il taglio pratico ha l'obiettivo di mettere subito in pratica quanto imparato. Il corso ha una durata di 2 giorni e sarà svolto online.



Contattaci a [info@qmates.tech](mailto:info@qmates.tech)

# CONTENUTI

- ***Introduzione all'eXtreme Programming***

La prima parte del corso ha come obiettivo l'esposizione delle tecniche di XP al fine di rendere evidente come esse si rafforzino a vicenda. Spiegheremo come l'eXtreme Programming riesce ad ottenere gli obiettivi descritti sopra e nel farlo condivideremo un linguaggio comune. Definiremo pratiche tecniche come il Pair Programming, Test Driven Development e Test Automation, Simple Design, Collective Code Ownership, Continuous Integration e Delivery, Refactoring e Coding standards.

- ***Test Driven Development e Simple Design***

Il test driven development porta all'estremo l'efficienza dei test automatici forzando lo sviluppatore a crearli prima della scrittura del codice di produzione. Questa pratica è il ciclo di feedback più breve in assoluto dell'XP e per questa ragione viene spiegata per prima. Il beneficio principale è che, se applicato bene, il TDD genera un design disaccoppiato, coeso, semplice e quindi facilmente modificabile in futuro.

- ***Acceptance Test Driven Development e Piramide dei test***

L'acceptance test driven development fa parte di un ciclo più ampio di test automation che accompagna lo sviluppatore non solo ad avere feedback sul software design grazie al ciclo del TDD ma anche ad avere feedback sulla correttezza della funzionalità in sviluppo. L'ATDD è un ciclo che avvicina lo sviluppo di una feature al business e acuisce il pensiero iterativo ed incrementale. La definizione di acceptance test ci porterà anche ad esplicitare il concetto di piramide dei test.

- ***Codice pulito***

Riuscire come team ad implementare dei pattern di clean code è un abilitatore alla collaborazione tra sviluppatori. In questo modulo raccontiamo ai partecipanti la



definizione di clean code per identificare le parti del codice da rifattorizzare. Sperimentaremo il refactoring di un codice legacy in modo da apprendere le tecniche utili a pulire le parti di codice non facilmente modificabili. Verranno sperimentate le pratiche tecniche come Refactoring, Good Design, Coding Standard e Collective code ownership.

## OBIETTIVI

L'obiettivo del corso è offrire ai partecipanti tecniche pratiche di sviluppo software per riuscire, nel quotidiano, a:

- ↓ lead time: il tempo necessario a passare da un'idea al suo rilascio in produzione;
- ↑ collective code ownership: quanto è distribuita la conoscenza delle parti di prodotto all'interno del team;
- ↓ curva d'ingresso: il tempo necessario per acquisire conoscenza ed essere efficace come nuovo developer nello sviluppo del software;
- ↑ testabilità: livello di testabilità del codice e quanto il prodotto sia testato automaticamente;
- ↑ flessibilità del team: il tempo necessario al team per cambiare direzione;
- ↓ livello di Work in Progress: il numero di feature al quale il team sta lavorando in contemporanea in un dato momento.

## TARGET

- sviluppatori software indipendentemente dal linguaggio di programmazione e dalla specializzazione;
- engineering manager;
- head of engineering;
- CTO.



Contattaci a [info@qmates.tech](mailto:info@qmates.tech)