

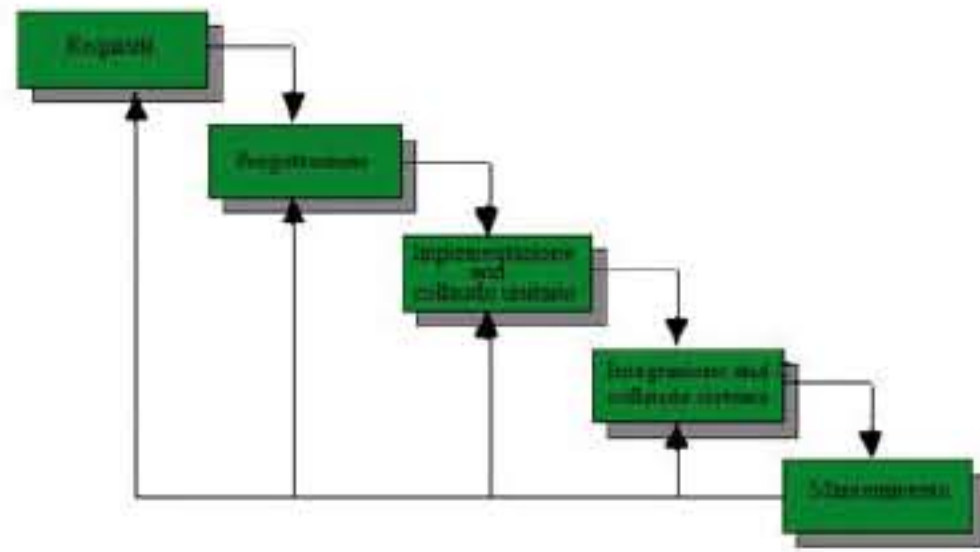
INGEGNERIA DEL SOFTWARE: I METODI AGILI E EXTREME PROGRAMMING

FELICE PESCATORE

In contrapposizione ai metodi tradizionali di sviluppo del software, negli ultimi anni, si sono andati sviluppando diversi metodi alternativi. I metodi **Agili**, in particolare, si caratterizzano per una forte riduzione della documentazione, per una continua reiterazione di rilasci incrementali del software seguiti da aggiornamenti dei requisiti e, in generale, per uno snellimento dei processi.

Il problema della sequenzialità dei metodi classici

Il principale problema legato ai modelli classici di sviluppo del software (dal modello a cascata ed in parte anche quello a spirale) è dato dalla sequenzialità del processo, che porta alla definizione delle specifiche nella fase di avvio del progetto.



il classico modello a cascata

Nella realtà i requisiti sono spesso inizialmente poco chiari sia al team di sviluppo che al committente stesso, rendendo utopistica la pretesa di congelare la definizione in una fase predefinita del processo. Spesso i dubbi si chiariscono solo con il rilascio di una prima release del prodotto, con la conseguenza che gli errori di analisi possono addirittura portare ad una riprogettazione di parte del sistema, allungando i tempi/costi di sviluppo e deludendo le attese del committente.

Anche una rigida pianificazione dei metodi di lavoro può presentare effetti indesiderati, inducendo gli sviluppatori ad assumere una mentalità processiva e burocratica (*check-list mentality*) che non stimola la ricerca di soluzioni innovative ai problemi che si presentano.

Il Manifesto della Programmazione Agile

Le criticità dei metodi tradizionali appena illustrati hanno indotto la nascita di processi di sviluppo alternativi, i così detti metodi *Agili* o *Flessibili*, nei quali viene abbandonata l'idea di un processo monolitico che porti al rilascio del software, mentre vengono considerati approcci di tipo incrementale che adattano lo sviluppo ai requisiti man mano che questi vengono esplicitati.

Alla base dei Metodi Agili vi è un vero e proprio movimento

culturale, l'**Agile Alliance**, volto all'esaltazione dell'individuo, rispetto alle procedure, e del software in quanto tale, rispetto ai documenti, alle negoziazioni commerciali e alle pianificazioni stringenti.

Nel 2001 è stato pubblicato il *Manifesto della Programmazione Agile* che riassume i principi ideologici su cui questi metodi si fondano:

*"We have come to value
Individuals and interactions over process and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan.
That is, while there is value in the items on the right, we value the items on the left more"*

ovvero:

Le persone e le interazioni prima dei processi e degli strumenti, privilegiare le persone e le interazioni tra di esse, cercando di ottenerne il meglio, senza imbrigliarle in complicati processi che ne possano limitare la creatività ed abbassarne il morale;

Software che funziona prima della documentazione comprensiva, costruire un sistema software che funziona, e solo in un secondo tempo pensare alle problematiche di documentazione che non deve sopperire alle mancanze o ai malfunzionamenti dello stesso;

Collaborazione con i committenti al posto della negoziazione del contratto, stabilire un rapporto di collaborazione con il committente, in modo che sia anche lui una forza trainante nello sviluppo del progetto. La collaborazione produce risultati migliori rispetto a pratiche che prevedono di imbrigliare il committente all'interno di contratti limitanti e stringenti;

Risposta ai cambiamenti piuttosto che aderenza alla pianificazione, sapersi adattare alle modifiche necessarie piuttosto che legarsi in modo inscindibile ai requisiti raccolti, in modo da migliorare la risposta ai cambiamenti delle specifiche e dell'ambiente circostante.

Le Metodologie Agili

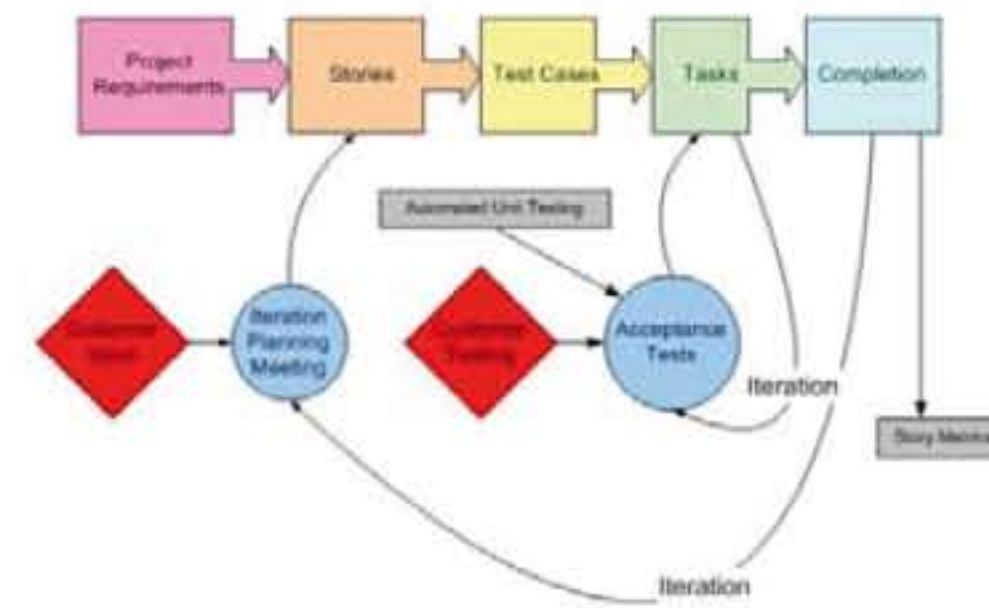
Dal Manifesto Agile sono nate svariate metodologie che interpretando i principi in modo differente, forniscono all'ingegnere del software, nuovi strumenti di cui servirsi per risolvere problematiche specifiche.

Di seguito viene proposta una carrellata delle metodologie più utilizzate che hanno raggiunto un buon grado di maturità e penetrazione nei contesti produttivi reali.

Extreme Programming

La disciplina che ha portato alla ribalta le metodologie agili e sicuramente la Programmazione Estrema (Extreme Programming). Sebbene sia molto nota per l'approccio alla progettazione e codifica, essa è primariamente un metodo di gestione del progetto, aspetto su cui si è concentrata l'evoluzione dell'ultimo periodo. L'XP (nell'edizione 2005) è definita su **5 valori** (*Comunicazione, Semplicità,*

Feedback, Coraggio, Rispetto), **14 principi** (*Umanità, Economia, Mutuo beneficio, Auto-similarità, Miglioramento, Diversità, Riflessione, Flusso, Ridondanza, Fallimento, Qualità, Piccoli passi, Accettare le responsabilità*), **13 pratiche primarie e 11 pratiche corollarie**.

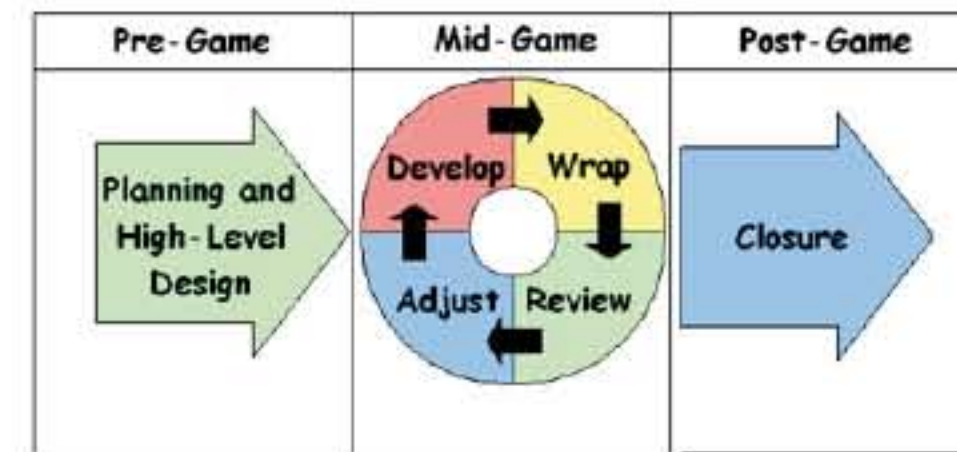


tipico processo extreme programming

Scrum

SCRUM è una disciplina agile di gestione del processo che funge da contenitore di pratiche di **ingegneria** del software già esistenti. Basato su un approccio iterativo ed incrementale, è studiato per sistemi dove i requisiti cambiano velocemente, fornendo dei modi per migliorare la comunicatività e massimizzare la cooperazione.

Il processo si basa sull'elenco di caratteristiche funzionali richieste che vengono assegnate al team di sviluppo che se ne occuperà durante una iterazione di sviluppo composta da massimo 30 giorni e chiamata *sprint*, al termine del quale la nuova funzionalità sarà disponibile al cliente.



tipico processo extreme programming

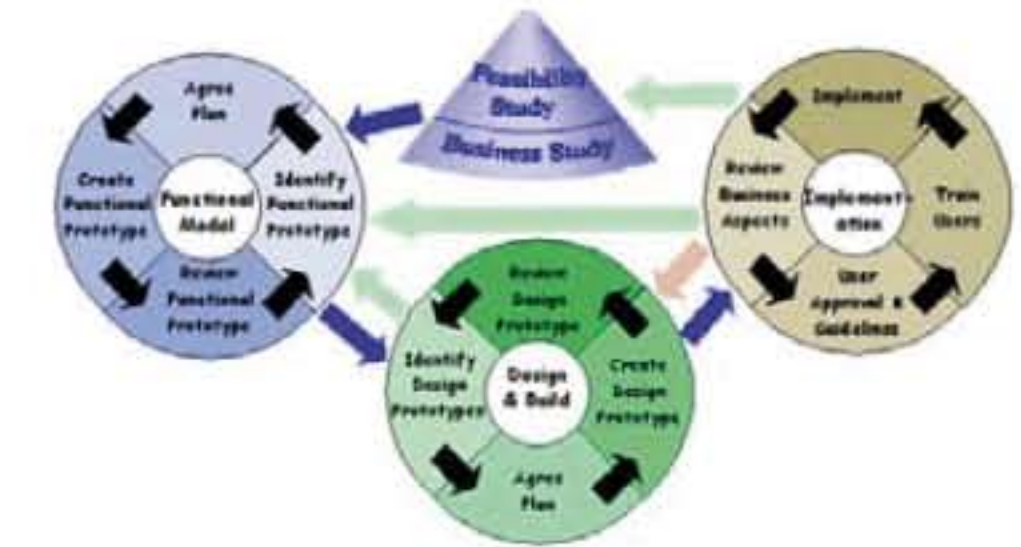
Dynamic Systems Development Method (DSDM)

DSDM è uno standard europeo che sta conoscendo una notevole diffusione, anche se è più un framework che un metodo vero e proprio. Basato anche questo su sviluppo iterativo ed incrementale, prevede però fasi di pre-project, studio di fattibilità e studio funzionale che vengono eseguiti in modo seriale. Un aspetto importante è che, in fase di studio di fattibilità, il metodo prevede di valutare se per il progetto specifico questa disciplina è l'approccio corretto.

Agile Modeling e Agile Data

L'Agile Modeling (AM) e l'Agile Data (AD) sono due metodologie proposte da Scott Ambler per la creazione di modelli agili relativi all'applicazione, alla documentazione e ai dati. Il concetto di base è che un modello agile è un

modello buono quanto basta, ovvero deve essere chiaro, dettagliato, accurato, consistente e comprensibile.



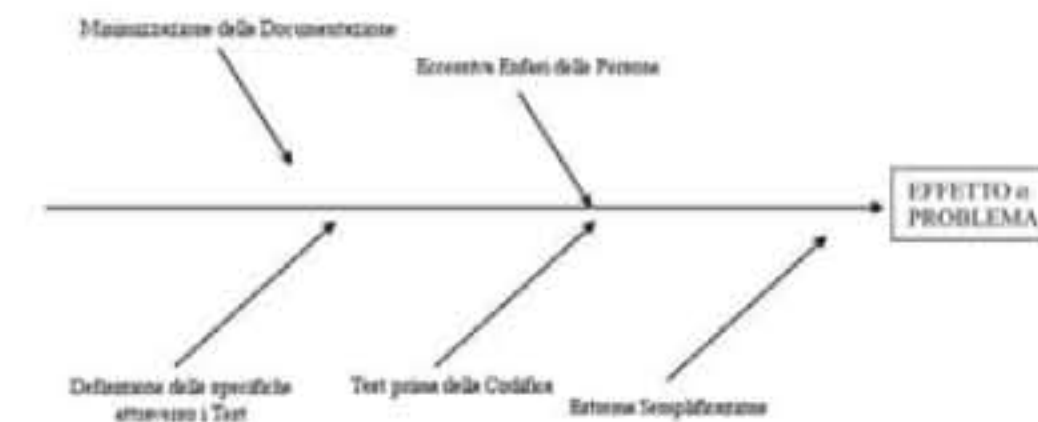
tipico processo DSDM

Feature Driven Programming

Si tratta della programmazione orientata alle funzionalità, basata su una scaletta composta da cinque passi: *sviluppo di un modello complessivo, costruzione di una lista di funzionalità, pianificazione sulla base delle funzionalità, progettazione delle funzionalità, costruzione delle funzionalità*.

Criticità dei Metodi Agili

Come è possibile immaginare, i metodi agili presentano numerose criticità che li rendono fortemente non adatti a certi contesti.



criticità modelli Agili

La minimizzazione della documentazione fa sì che tutta la conoscenza del prodotto in fase di sviluppo sia racchiusa nelle persone che ci lavorano, originando problemi di comunicazione interna con team troppo grandi (più di 20 persone) e creando difficoltà nel caso vengano cambiati i componenti del team stesso.

Un ulteriore problema è la validazione spinta dei requisiti attraverso i test senza il supporto di strumenti specifici per analizzare le informazioni in modo sistematico. I test, per definizione, possono descrivere solamente un campione finito di possibili comportamenti, lasciandone pericolosamente indefiniti altri che, seppur raramente, potrebbero presentarsi. È per questo motivo che i metodi agili si rivelano molto poco efficaci nello sviluppo di applicazioni *safety-critical*, che richiedono una certificazione del prodotto.

Note

1) Agile Alliance Manifesto for Agile Software distribution, www.agilealliance.org