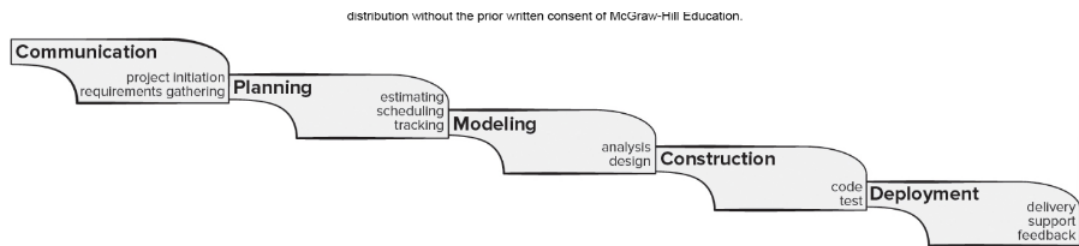


Riassunto

Process Models

Waterfall Process Model

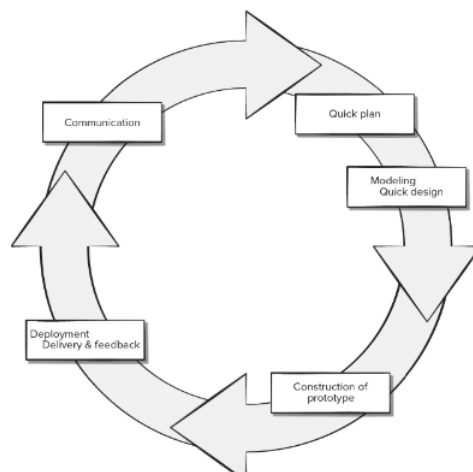


La metodologia Waterfall è un approccio sequenziale allo sviluppo del software, organizzato in fasi distinte e lineari.

Inizia con la definizione chiara dei requisiti attraverso la Comunicazione, prosegue con la Pianificazione, il Modeling, Construction e per finire il Deployment,

Ogni fase è ben definita e completa prima che la successiva abbia inizio, il che implica che i cambiamenti ai requisiti sono difficili da gestire una volta iniziato il processo. Sebbene la metodologia Waterfall abbia la sua efficacia in progetti con requisiti stabili e ben compresi, può essere meno adatta a situazioni in cui l'adattabilità e la risposta ai cambiamenti sono fondamentali, come nei contesti agili.

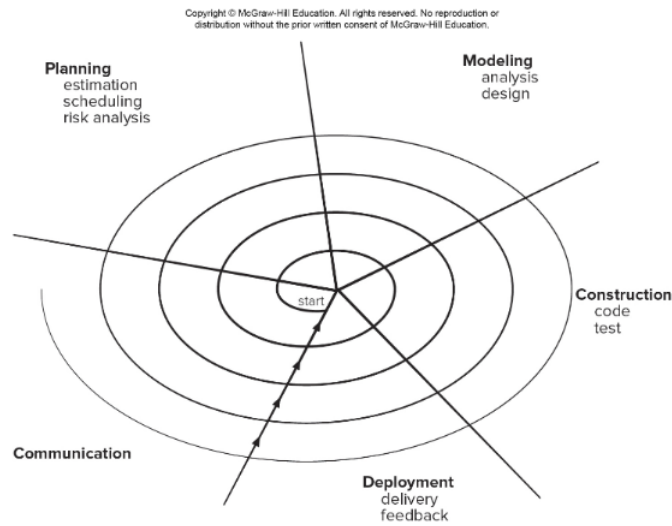
Prototyping Process Model



La metodologia Prototyping si concentra sulla creazione di prototipi del software desiderato prima della sua completa implementazione.

Questo approccio consente agli utenti e agli sviluppatori di interagire con un'anteprima funzionante del prodotto, raccogliendo feedback e apportando modifiche iterativamente.

Spiral Process Model

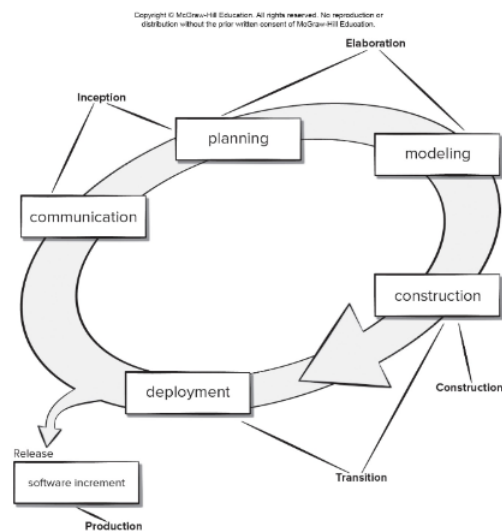


La metodologia Spiral è un approccio iterativo che combina elementi di pianificazione, progettazione, implementazione e valutazione in cicli ripetuti.

Questo modello a spirale prevede la creazione di prototipi in ciascuna iterazione, concentrandosi sull'identificazione e sulla mitigazione dei rischi associati al progetto.

Le fasi successive vengono affrontate solo dopo un'attenta valutazione delle fasi precedenti, consentendo una maggiore flessibilità nel gestire cambiamenti nei requisiti e una miglior gestione dei rischi durante tutto il ciclo di sviluppo del software.

Unified Process Model



L'Unified Process Model è un framework di sviluppo software che integra elementi chiave di altre metodologie, in particolare quelli derivati dall'Object-Oriented Analysis and Design (OOAD) e dalla Unified Modeling Language (UML).

UP si basa su cicli di sviluppo iterativi e incrementali, suddividendo il processo in fasi come l'**Inception**, **Elaboration**, **Construction** e **Transition**. Ogni fase produce output ben definiti, inclusi modelli UML e documentazione.

L'Unified Process enfatizza la modellazione visuale e la validazione continua del sistema attraverso iterazioni, fornendo un approccio flessibile e adattabile alle esigenze del progetto.

Agile

La metodologia Agile è un approccio di project management che mette al centro il cliente, coinvolgendolo attivamente nell'avanzare del progetto, attraverso frequenti rilasci del prodotto in cicli brevi di lavoro.

Qui di seguito riportiamo i quattro valori fondanti dell'Agile:

- 1. Accogliere i cambiamenti nei requisiti, anche a stadi avanzati dello sviluppo.**
 - L'Agile si adatta facilmente ai cambiamenti nelle esigenze del cliente, promuovendo la flessibilità e la risposta rapida ai feedback.
- 2. Consegnare frequentemente software funzionante, con preferenza per intervalli di tempo brevi.**
 - L'approccio Agile mira a consegnare piccole porzioni di valore in modo continuo, consentendo al cliente di ottenere benefici rapidi e fornire input costante.
- 3. Cooperare con il cliente per tutta la durata del progetto.**
 - La collaborazione continua con il cliente è fondamentale per comprendere meglio le sue esigenze e adattarsi di conseguenza.
- 4. Realizzare progetti con individui motivati, fornendo loro l'ambiente e il supporto di cui hanno bisogno**
 - L'Agile valorizza il ruolo degli individui e promuove un ambiente che favorisce la motivazione, la creatività e l'autonomia.

SCRUM

Lo Scrum è un framework agile ampiamente utilizzato per la gestione di progetti complessi nello sviluppo del software.

Scrum si basa su principi di trasparenza, ispezione e continuous integration. Questo approccio favorisce la collaborazione tra i membri del team, ottimizzando la flessibilità e la capacità di risposta alle mutevoli esigenze del progetto, suddividendo il lavoro in iterazioni chiamate "Sprint" e promuovendo la comunicazione costante attraverso ruoli chiave come il Product Owner, lo Scrum Master e il Team di Sviluppo.

Cosa sono gli artefatti di Scrum?

Gli Scrum Team utilizzano strumenti chiamati artefatti di Scrum per la risoluzione di problemi e la gestione di progetti. Gli artefatti di Scrum offrono, sia ai membri del team che alle parti interessate, informazioni essenziali riguardo la pianificazione e le attività. Ci sono tre artefatti principali:

Product Backlog

Il Product Backlog è un elenco dinamico composto da caratteristiche, requisiti, miglioramenti e risoluzioni di problemi che è necessario completare per la riuscita di un progetto. Si tratta, sostanzialmente, dell'elenco dei compiti che il team dovrà svolgere, e le cui priorità vengono riviste e ridefinite costantemente per essere adattato alle trasformazioni del mercato. Il product owner mantiene aggiornato l'elenco, rimuovendo gli elementi irrilevanti e aggiungendo nuove richieste avanzate dai clienti.

Sprint Backlog

Lo Sprint Backlog è l'elenco degli elementi che il team di sviluppo deve completare nel ciclo di Sprint corrente. Prima di ogni Sprint, il team sceglie gli elementi su cui lavorare dal Product Backlog. Uno Sprint Backlog è flessibile e può evolversi durante uno Sprint.

Increment

L'Increment è un passo verso un obiettivo o una visione. È il prodotto finale utilizzabile dopo uno Sprint. I team possono adottare metodi diversi per definire e dimostrare i loro Sprint Goal. Nonostante la flessibilità, lo Sprint Goal fondamentale (ciò che il team vuole ottenere dallo Sprint corrente) non può essere compromesso.

Ad esempio, alcuni team scelgono di rilasciare qualcosa ai clienti alla fine dello Sprint, pertanto lo Sprint Goal sarebbe completato una volta rilasciata la modifica del software. Altri team potrebbero lavorare al completamento di una serie di funzionalità che verranno rilasciate insieme. In questo caso, lo Sprint Goal dovrebbe considerarsi completato nel momento in cui una funzionalità viene testata con successo.

Cosa sono i ruoli Scrum?

Uno Scrum Team necessita di tre ruoli specifici: un Product Owner, uno Scrum Master e un team di sviluppo.

Product Owner

Il Product Owner è attento ad assicurare che il team di sviluppo offra all'azienda il maggior valore possibile. Sono capaci di comprendere e dare priorità alle mutevoli esigenze degli utenti e dei clienti finali. I product owner agiscono in maniera efficace quando:

- forniscono al team una guida chiara sulle funzionalità da fornire successivamente;
- colmano il divario tra ciò che l'azienda vuole e ciò che il team comprende;
- stabiliscono quando e con quale frequenza le nuove versioni saranno rilasciate.

Scrum Master

Gli Scrum Master sono i campioni di Scrum all'interno del team. Sono responsabili dell'efficienza dello Scrum Team. Allenano i team, i Product Owner e l'azienda per migliorare i propri processi Scrum e ottimizzare la consegna. Gli Scrum Master sono anche responsabili di:

- pianificare le risorse necessarie per ogni Sprint;
- facilitare gli altri Sprint e le riunioni del team;
- guidare la trasformazione digitale all'interno del team;
- facilitare la formazione del team quando nuove tecnologie vengono adottate;
- comunicare con gruppi esterni per risolvere eventuali sfide che il team affronterà insieme.

Team di sviluppo di Scrum

Lo Scrum Team consta di tester, progettisti, specialisti UX, ingegneri Ops e sviluppatori. I membri del Team hanno competenze diverse e si formano a vicenda, in modo che nessuna persona provochi un collo di bottiglia nella consegna del lavoro.

Il team di sviluppo Scrum esegue le seguenti operazioni:

- lavora in modo collaborativo per garantire il completamento dello Sprint;
- sostiene le pratiche di sviluppo sostenibile;
- si organizza in maniera autonoma e affronta i progetti con spirito di gruppo;
- guida la pianificazione e la stima della quantità di lavoro che è possibile completare per ogni Sprint.

Cosa sono gli eventi di Scrum?

Gli eventi di Scrum, o le cerimonie Scrum, sono un insieme di riunioni in sequenza che gli Scrum Team svolgono regolarmente. Tra gli eventi di Scrum, rientrano:

Sprint Planning

In questo evento, il team stima il lavoro da completare nello Sprint successivo. I membri definiscono gli Sprint Goal specifici, misurabili e raggiungibili. Al termine della riunione di pianificazione, ogni membro di Scrum sa in che modo ogni Increment può essere consegnato nello Sprint.

Sprint

Uno Sprint è il periodo di tempo effettivo in cui lo Scrum Team lavora insieme per completare un Increment. La durata tipica di uno Sprint è di due settimane, ma può variare a seconda delle esigenze del progetto e del team. Quanto più complesso è il lavoro e quante più sono le incognite, tanto più lo Sprint dovrebbe essere breve.

Daily Scrum o stand-up

Un Daily Scrum è una breve riunione in cui i membri del team si aggiornano e pianificano la giornata. Riferiscono sul lavoro completato e comunicano eventuali difficoltà nel raggiungimento degli Sprint Goals. Si chiama stand-up perché l'obiettivo è far durare la riunione il più breve tempo possibile, come se tutti fossero in piedi.

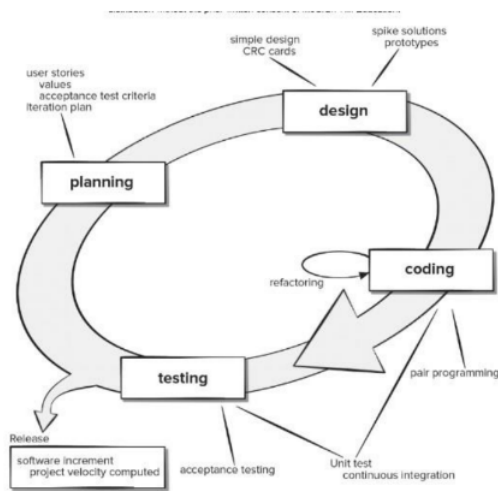
Sprint Review

Alla fine dello Sprint, il team si riunisce per una sessione informale per rivedere il lavoro completato e presentarlo alle parti interessate. Il Product Owner può anche rielaborare il Product Backlog in base allo Sprint corrente.

Sprint Retrospective

Il team si riunisce per documentare e discutere ciò che ha funzionato e ciò che non ha funzionato durante lo Sprint. Le idee maturate vengono utilizzate per migliorare gli Sprint futuri.

Extreme Programming (XP)

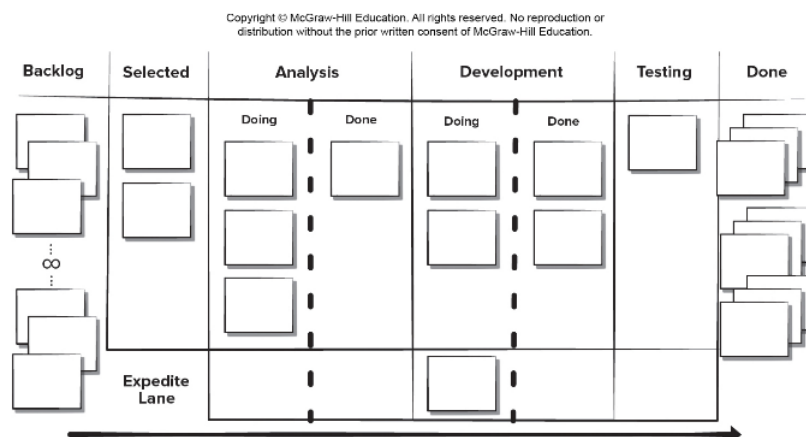


Uno degli aspetti indubbiamente più interessanti dietro XP è l'utilizzo diffuso di **TDD** cioè lo scrivere i test prima di iniziare a sviluppare la parte funzionale. Attraverso questa pratica è possibile testare ogni piccola componente software ancora prima che venga implementata e individuare possibili errori fin dalle prime fasi dello sviluppo.

Una volta sviluppata una determinata funzionalità, il codice prodotto viene trasferito su un build server. A ogni build vengono avviati i test automatici, in modo da assicurare l'integrità del codice sorgente e dare la possibilità al team di accedere in qualsiasi momento a una versione funzionante del software. Questa pratica è chiamata **Continuous Integration**.

Il **Pair Programming** è forse la pratica più conosciuta di XP. L'idea di base è che due sviluppatori lavorino insieme alla scrittura del codice: uno assume il ruolo di "guidatore" e l'altro di "navigatore" o "osservatore".

Kanban

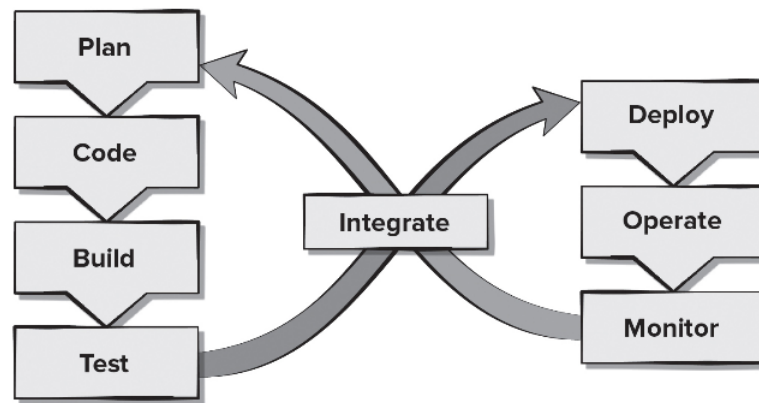


Il lavoro di un team che utilizza il metodo Kanban consiste nel gestire la Kanban Board, uno strumento usato per visualizzare e ottimizzare il lavoro del team.

Una semplice Kanban Board deve avere almeno le seguenti tre colonne:

- To Do
- In Progress
- Done

DevOps



La metodologia DevOps può essere concepita come un ciclo continuo che attraversa fasi di sviluppo, monitoraggio e integrazione.

Inizia con lo sviluppo del software, prosegue con l'automazione del rilascio, e successivamente si estende alla fase di monitoraggio delle prestazioni e del feedback degli utenti. Questa retroazione è cruciale per identificare miglioramenti, risolvere problemi e informare il ciclo successivo di sviluppo.