

Agile software development

Introduzione

Obiettivi:

- Comprendere il giusto punto di equilibrio tra documentazione e sviluppo

Spesso attività di contorno potrebbero rappresentare processi fuorvianti rispetto alla concreto sviluppo del sistema software, nonostante siano fondamentali per la comprensione dei requisiti funzionali posti. Potrebbero essere visualizzate come *pratiche burocratiche*, ossia l'insieme di elementi modellativi trattati non vengono mai sviluppati per ragioni legate alla costruzione del sistema software. A causa di questo elevato standard processuale non è riconducibile alcuna immediatezza tra la progettazione teorica e l'implementazione software, provocando una perdita della qualità.

L'impegno dovuto a garantire coesione a livello strutturale comprende un vasto insieme di azioni, le quali potrebbero provocare un'elevata perdita di tempo qualora l'analisi adottata non sia ben chiara fin dall'inizio. Da cui ne deriva l'inutilità di una pianificazione perfetta, poichè l'intero contesto è soggetto a dinamicità, soprattutto un campo che riguardi lo sviluppo e la progettazione software.

Non solo il *sistema software* dovrà essere flessibile ai cambiamenti, ma l'intero contesto sviluppato dovrà reagire prontamente a modifiche.

Manifesto per lo sviluppo agile

Il motivo che ha portato alla creazione del **manifesto**, è dovuto al netto spreco di risorse durante fasi di sviluppo software. Il termine correlato, *Agile software development*, non deve essere considerato come un metodo di progettazione, ma rappresenta un insieme di *pratiche* guidate da *principi* e qualità sia *interne* che *esterne*. Attraverso il processo di analisi prodotto occorre valorizzare un insieme di prospettive, quali:

- Interazioni tra progettisti e sviluppatori al di sopra di processi sequenziali e strumenti tecnici
- Sviluppare e adoperare codice piuttosto che prediligere una documentazione esaustiva
- Imbastire una collaborazione con il *costumer*, denigrando una *negoziiazione* conflittuale
- Reagire prontamente a cambiamenti, evitando di sottostare alle linee guida originarie

Il compito del *manifesto* prevede di attribuire maggiore rilevanza all'entità poste alla sinistra dell'elenco, provando a descrivere un approccio che possa portare ad un concreto beneficio per progetti futuri, senza escludere un prossimo utilizzo di tutti gli elementi posti alla destra.

Principi

In relazione all'introduzione precedente, sono formulati di seguito i principi su cui stabilisce il proprio approccio il *manifesto*, suddivisi in:

- La priorità principale richiede che il *customer* sia pienamente soddisfatto del risultato ottenuto, ciò può concretizzarsi solamente se sviluppatori optino per un continuo dialogo con i clienti e fornendo sequenzialmente versioni del sistema software implementato
- Reagire prontamente a modifiche e a variazioni di requisiti funzionali, sfruttando il cambiamento per ottenere vantaggio competitivo
- Valorizzare il *lifecycle* del modello a spirale, il quale impone certe temporalità in cui richiedere confronti e discussioni
- Giornalmente sviluppatori e il *business team* devono rendersi protagonisti nella realizzazione di task scelte
- Predilire il dialogo tra i singoli, poichè permette di diffondere in maniera efficiente ed efficace tutte le informazioni ritenute importanti
- Porre particolare attenzione a tecnologie abili ed eccellenti, relative non solo all'ambiente di sviluppo utilizzato, ma rispetto anche a novità progettuali o metodi differenti di cooperazione, pur di riuscire nell'intento della richiesta
- Adottare un approccio dedicato ad *improvement*, che tendi a migliorare ad ogni passo, piuttosto che formalizzarsi sulla pianificazione dettagliata all'inizio della progettazione

Metodi agili

Esistono molte pratiche che adottano l'insieme dei principi descritti precedentemente, alcune delle quali sono in totale contraddizione, tuttavia proprio questa discordanza dovrebbe dare vita a un processo legato al continuo miglioramento, pur di aumentare la qualità di progettazione e sviluppo. Le metodologie si suddividono in:

- ...