

Processi software

Friday, 17 March 2023

14:17

Definisce:

- approccio disciplinato per la creazione di un software
- Ed ogni attore viene definito come:
 - o Cosa: Attività
 - o Chi: i ruoli che ogni attore deve svolgere
 - o Quando
 - o Come: quali metodologie

Ed ogni processo software hanno attività in comune:

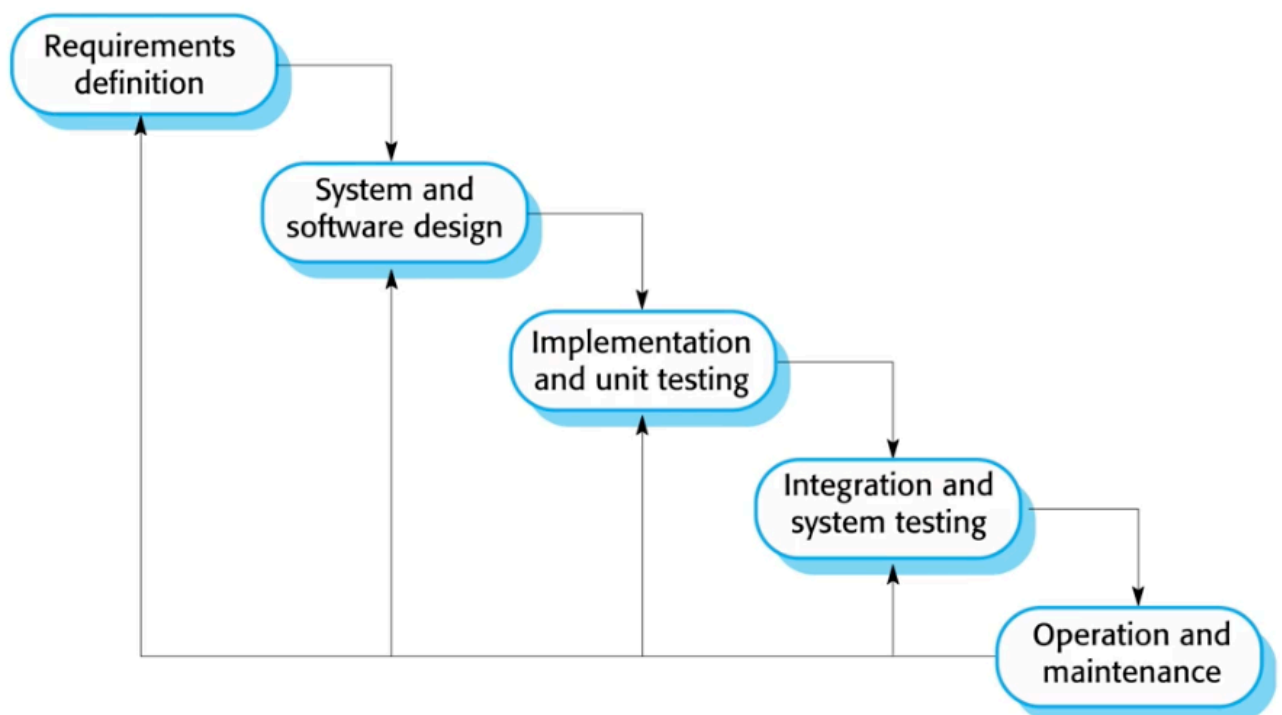
- Requisiti, dove definiamo attività/vincoli
- Progettazione: organizzazione dei componenti software
- Implementazione: da progetto a codice
- Validazione: verifica corretto funzionamento, deve fare ciò che il cliente voleva
- Manutenzione: cambiare sistema a seconda delle esigenze del cliente
- Gestione progetto

Le varie tipologie si differiscono in cosa, chi quando

Ed esistono a differenza del progetto e le esigenze.

Esempi:

- 1) Cascata, dove attività a sequenza



- o Ogni attività crea documenti che dovranno essere revisionati
Ed ogni attività procede da una all'altra ed ogni documento deve essere approvato

Molto spesso le attività vengono svolte da team differenti

- Struttura

- i. Requirement definition: definire i requisiti
- ii. System software design: progettazione
- iii. Implementation and unit testing: nel test di unità testiamo i singoli componenti
- iv. Integratio and system testing: nell'integration controlliamo come ogni componente software si comporta tra di loro, mentre system setting lo controlliamo come sistema quindi come l'unente lo vede
- v. Operation and maintenance: bug fixing

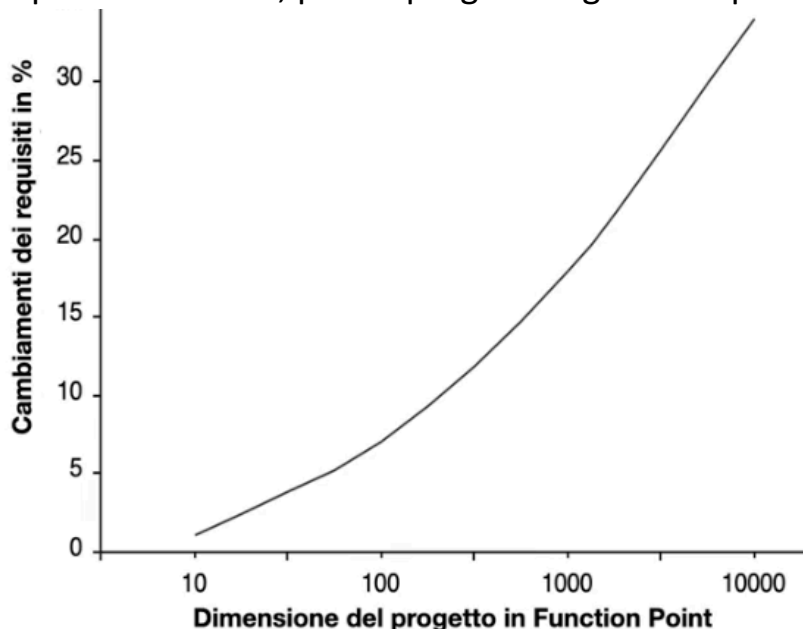
Da qui possiamo andare in qualsiasi attività di sopra

- Fallimento

Questo è soggetto a fallimenti siccome, quando noi non abbiamo in chiaro i requisiti mettiamo una vita a cambiare il tutto siccome dobbiamo rifare tutto il circolo.

Quindi usare questo sistema solamente se tutto è ben chiaro ed abbiamo requisiti chiari

Il problema è che, più un progetto è grande e più i requisiti cambieranno



2) Spirale, dove ogni giro di spirale nuova build

3) Unificato, ciò che noi useremo, ed un esempio di un processo unificato è:

- Sviluppo iterativo, incrementale ed evolutivo



Ter

- Iterativo siccome abbiamo 3 build. Andiamo ad eseguire tutte le build. Che andranno a creare meglio il sistema.
- Incrementale siccome il nostro sistema si va costruendo. Col tempo



- Evolutivo siccome, dopo ogni iterazione, abbiamo un output che il cliente può vedere e da qui avremo un feedback
- Questo riesce ad abbracciare l'intero processo, essere sempre più vicino all'obiettivo

Ciascuna iterazione comporta un piccolo sottoinsieme di requisiti da risolvere (non necessariamente tutti i requisiti sono scelti)

E dopo si va alla progettazione rapida, implementazione e test

Vantaggi:

- (-) probabilità fallimento
- (-) rischi
- (+) progresso visibile

E per via di questo, (+) feedback

Questo feedback:

- ◆ Proviene dai test, devs, cliente, mercato, legale
- ◆ Serve per comprendere l'avanzamento del team per affrontarlo

Questo ci permette un migliore adattamento

- (+) gestione difficoltà -> affrontiamo i problemi spezzandoli, e quindi dividiamo le difficoltà

Come detto prima, questo avviene in 3 settimane:

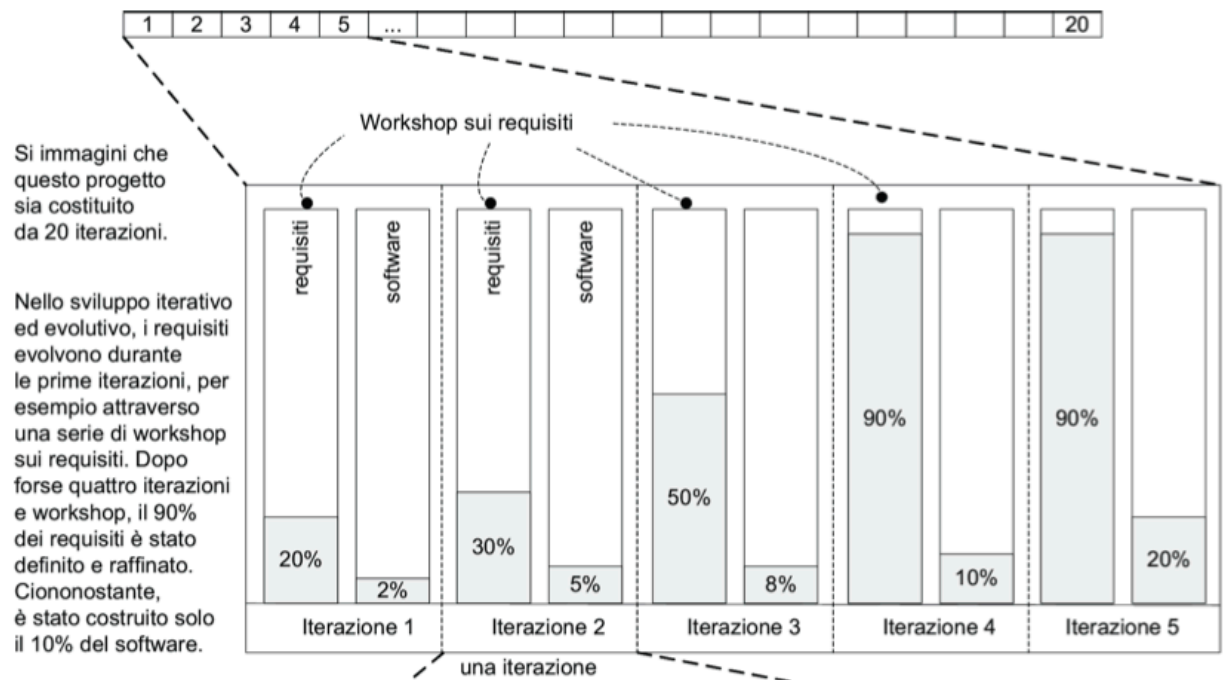
- Varia da 2/6 settimane che viene deciso all'inizio
- **Non** può cambiare, quindi non possiamo allungare, piuttosto diminuire/aumentare i requisiti
- Viene chiamato **timeboxing**: ogni iterazione ha lunghezza fissa
 - (+) lungo aumentiamo il rischio del progetto
 - (-) lungo non consentiamo un lavoro sufficiente a produrre software funzionante e feedback

Errori comuni che si finisce in un progetto a cascata:

- Maggior parte requisiti/casi d'uso prima dello sviluppo
 - Troppe specifiche prima dell'implementazione
- Quindi non deve essere completa, perché? Vogliamo poter cambiare velocità
Quindi UML come abbozzo -> Agile

Nota: il codice deve essere facilmente modificabile e comprensibile

Nota: il codice deve essere facilmente modificabile e comprensibile



Questo è come requisiti/software cresce a seconda delle prime iterazioni

Quindi:

- Mai tentare di pianificare l'intero progetto **dettagliatamente** all'inizio
- Noi dobbiamo promuovere la pianificazione da rischi maggiori e cliente
- Le prime iterazioni si concentrano sulla stabilizzazione del nucleo quindi Facciamo passo per passo, piano, andando prima su ciò che è sicuro a sviluppare. E dove non c'è la certezza fare piccoli passi