

- **Lezione 3 – (14/10/2024)**

Una metrica importante legata all'affidabilità del software è la disponibilità, che rappresenta la percentuale di tempo durante il quale il software è risultato usabile nel corso della sua vita. Questa percentuale è influenzata dal numero di guasti che si verificano e dal tempo necessario per risolverli.

L'affidabilità e la disponibilità sono metriche fondamentali /importanti per quei sistemi in cui l'interruzione del servizio può portare a una perdita di efficienza e sicurezza, con conseguenze economiche e sociali significative. Come ad esempio:

- Sistemi di trasporto
- Sistemi di gestione del traffico aereo
- Sistemi di controllo del volo
- Sistemi di produzione e distribuzione di energia
- Sistemi di comunicazione

Nel corso degli anni, lo sviluppo software ha attraversato diverse fasi:

- Fase di abilità: dove predominano gli aspetti di lavoro individuale e creativo
- Fase artigianale: è quella che ha portato a non usare un singolo programmatore ma a sviluppare piccoli gruppi/team specializzati
- Fase industriale: Qui, lo sviluppo e la manutenzione del software sono pianificati e coordinati, con un crescente supporto da parte di strumenti automatici.

Lo standard IEEE Std. 610.12 (1990) ha fornito una definizione più completa dell'ingegneria del software, comprendente:

- L'applicazione di un approccio sistematico, disciplinato e misurabile nello sviluppo, esercizio e manutenzione del software, utilizzando principi ingegneristici.
- Lo studio di questi approcci delineati nel punto precedente.

Quindi possiamo dire che il software può essere considerato come l'insieme di programmi, documenti e dati multimediali. Realizzati dall'ingegnere del software applicando un processo che conduca a risultati migliori. Questo viene fatto attraverso un approccio ingegneristico. Abbiamo già detto precedentemente che il software non si consuma, è complesso, invisibile, si conforma, si cambia.

I metodi e le tecniche di ingegneria del software mirano a garantire la qualità del prodotto, gestire la crescente domanda rispettando il budget, aggiornare applicazioni legacy, evitare ritardi nelle consegne e implementare con successo nuove tecnologie.

Miti (da sfatare) del Software:

- In caso di ritardo, basta aumentare il numero di programmatori;
- Una descrizione generica è sufficiente a scrivere i programmi. Eventuali modifiche si possono facilmente effettuare in seguito;
- Una volta messo in opera il programma, il lavoro è finito;
- Non c'è modo di valutare la qualità fino a quando non si ha a disposizione il prodotto finale;
- L'ingegneria del software è costosa e rallenta la produzione;

➔ Secondo Blocco

Per realizzare un prodotto software vengono svolte una serie di attività in un determinato tempo, con determinati costi e con determinate caratteristiche, tutto ciò viene descritto con il termine Processo Software.

Questo processo prevede l'uso di metodi, tecniche e strumenti, la creazione di prodotti intermedi e finali, il controllo gestionale, la garanzia della qualità e la gestione delle modifiche.

Precedentemente abbiamo parlato di come il ciclo di vita di un Prodotto Software si pianifica in tre stadi.

Nel primo stadio si possono riconoscere due tipi di fasi:

- Fasi di tipo definizione, si occupa di “cosa” il software deve fornire
- Fasi di tipo produzione, definisce il “come” realizzare quanto ottenuto con le fasi di definizione. Quindi si progetta il software, si codifica, si integra e si rilascia al cliente

Il secondo stadio, ovvero quello di Manutenzione, manutenzione è a supporto del software realizzato e prevede fasi di definizione e/o produzione al suo interno.

Durante ogni fase si procede ad effettuare il testing di quanto prodotto, attraverso tecniche di verifica e validazione (V&V) applicate sia ai prodotti intermedi che al prodotto finale.

Esistono 5 tipologie di Manutenzione:

- Manutenzione Correttiva: che ha lo scopo di eliminare i difetti (fault) che producono guasti (failure) del software
- Manutenzione Adattativa: che ha lo scopo di adattare il software ad eventuali cambiamenti a cui è sottoposto l'ambiente operativo per cui è stato sviluppato
- Manutenzione Perfettiva: che ha lo scopo di estendere il software per accomodare funzionalità aggiuntive
- Manutenzione Preventiva (o software reengineering): che consiste nell'effettuare modifiche che rendano più semplici le correzioni, gli adattamenti e le migliorie

Definizione di Ciclo di Vita secondo lo IEEE Std 610-12:

- Intervallo di tempo che va tra l'istante in cui nasce l'esigenza di costruire un prodotto software e l'istante in cui il prodotto viene dismesso.

- Include le fasi di definizione dei requisiti, specifica, pianificazione, progetto preliminare, progetto dettagliato, codifica, integrazione, testing, uso, manutenzione e dismissione. Queste fasi possono essere eseguite in sequenza, sovrapposte o ripetute più volte

Il modello del ciclo di vita del software specifica la serie di fasi attraverso cui il prodotto software progredisce e l'ordine con cui vanno eseguite, dalla definizione dei requisiti alla dismissione. La scelta del modello dipende dalla natura dell'applicazione, dalla maturità dell'organizzazione, da metodi e tecnologie usate e da eventuali vincoli dettati dal cliente. L'assenza di un modello del ciclo di vita corrisponde ad una modalità di sviluppo detta “build & fix” (o “fix-it-later”), in cui il prodotto software viene sviluppato e successivamente rilavorato fino a soddisfare le necessità del cliente.

