

Il testing

Nella fase di testing si svolgono determinate verifiche sul software precedentemente implementato, in particolare si eseguono in tal fase:

- 1) test unitari;
- 2) test di integrazione;
- 3) test funzionali;
- 4) test di performance;
- 5) test di accettazione;
- 6) test d'installazione;

I test unitari sono test che effettua ciascun programmatore sul modulo che ha implementato. I test di integrazione sono test effettuati da una o più persone adibite all'integrazione dei singoli moduli. Questi due tipi di test possono essere eseguiti al momento di una revisione (review) per opera di altri tecnici, o per mezzo di opportuni strumenti automatici.

Nei test funzionali vengono testati i requisiti funzionali, cioè quello che dovrebbe svolgere l'applicazione.

Nei test di performance vengono testati i requisiti non funzionali, osservando come risponde il sistema, portandolo a stress, controllando la velocità di risposta, la sicurezza che fornisce, la sua qualità. Dopo i test di performance si ha la sicurezza che il software sviluppato funzioni.

I test di accettazione comprendono:

- a) gli alpha test, i quali vengono effettuati di solito da un'equipe separata all'interno dell'azienda;
- b) i beta test, effettuati da utenti abilitati all'esterno dell'azienda;
- c) è possibile effettuare i test detti "benchmark", eseguiti dagli utenti, al posto dei precedenti test, nel momento in cui il software da testare sia già conosciuto dagli utenti.

I test d'installazione non sono obbligatori nel caso in cui si effettuino prima i test di accettazione. I test d'installazione comprendono dei test sull'installazione e la configurazione del software su di un parco macchine.

Esistono inoltre dei test chiamati di regressione, effettuabili in qualunque fase.

Nel momento in cui vengono corretti dei bug del software, tali correzioni potrebbero comportare ulteriori problemi e bug, per tale motivo si andrà a ritroso con i test, in modo da verificare se le correzioni effettuate comportino bug. Può anche accadere che un cliente non gradisca certe funzioni, cosicché esse vengono eliminate, anche in questo caso si andrà a ritroso con i test. Questi test a ritroso sono i cosiddetti test di regressione. La fase di testing (o di collaudo) può comportare lo sviluppo di altre applicazioni utilizzate per verificare gli algoritmi del software principale. La fase di testing produce un documento di collaudo, il quale riporterà i test da eseguire e come questi verranno effettuati. Nel caso si debbano verificare norme riguardanti la qualità, potrebbe essere necessario controllare, all'interno del software, il codice e come quest'ultimo sia stato scritto, questo rappresenta un altro tipo di collaudo. Nel documento di collaudo sono esplicitate le funzioni del programma, ottenendo in questo caso il cosiddetto "cross – reference", cioè dei riferimenti al

documento che riguarda i requisiti, in modo tale da verificare se tutti i requisiti sono stati implementati.

Il mantenimento

La fase di mantenimento viene avviata nel momento in cui il software prodotto risulta operativo, cioè nel momento in cui è stato distribuito ai clienti. Dunque il prodotto rilasciato è sempre mantenuto in sviluppo, e, a seconda dei casi, si integrano nuove caratteristiche o anche si correggono dei bug riscontrati. La fase di mantenimento rappresenta qualcosa di diverso rispetto alla possibilità di un cambio di versione maggiore (per esempio da release 1 a 2), anche se la fase di mantenimento comporterà comunque un minimo cambiamento di versione (ad esempio da 1.0 a 1.1). Nel caso di un cosiddetto salto generazionale si potrebbe ripercorrere tutto il ciclo di vita del software, mentre nel caso della fase di mantenimento non accade tutto questo, difatti si risponde di solito alle richieste del cliente mantenute nella fase di gestione della configurazione, dunque la fase di mantenimento non rappresenta una rivoluzione per il software. Si possono classificare diversi tipi di mantenimento:

- 1) il mantenimento di tipo correttivo;
- 2) il mantenimento di tipo adattivo;
- 3) il mantenimento di tipo perfezionativo;
- 4) il mantenimento di tipo preventivo;

Il mantenimento di tipo correttivo prevede la correzione di bug precedentemente riscontrati e segnalati, di solito si occupa di tale mantenimento un'equipe.

Il mantenimento di tipo adattivo potrebbe essere effettuato nel momento in cui la piattaforma su cui veniva eseguito il software subisce dei cambiamenti che comportano il mancato funzionamento del software stesso. In questo caso si cerca di adattare il software alla nuova versione della piattaforma. Si esegue questo tipo di mantenimento anche nel caso in cui si dovesse far eseguire il software su altre piattaforme.

Il mantenimento di tipo perfezionativo viene effettuato quando si cerca di ottimizzare il software, e potrebbe comprendere anche un'ottimizzazione della documentazione a corredo del software.

Il mantenimento di tipo preventivo si esegue nel momento in cui si tenta di proteggere il software da eventuali errori e/o eventi che potrebbero presentarsi.

Il mantenimento è una fase non facilmente effettuabile, in quanto prevede dei costi non indifferenti. Inoltre, quando un software non prevede certe caratteristiche, introducendo queste ultime con il mantenimento si rischia l'introduzione di grossi problemi, per tale motivo è bene essere accorti ed effettuare un salto generazionale nel momento in cui si rende necessario.

Il reverse engineering

Il reverse engineering è una tecnica che permette di svolgere un percorso inverso al ciclo di vita di un software, partendo dal codice a disposizione, giungendo al design e poi ai requisiti. Tale tecnica viene effettuata quando si rende necessario documentare dei vecchi programmi dei quali non sono disponibili, per esempio, la documentazione e codice sorgente. Il reverse engineering potrebbe essere effettuato da un'azienda con lo scopo di capire come vengono fornite delle funzionalità da un software della concorrenza.

Il re – engineering

Per re – engineering si intende il riprocessare tutto il ciclo di vita di un software partendo dal reverse engineering, dunque rappresenta il passo successivo al reverse engineering.