

Analisi dei requisiti

L'analisi dei requisiti è la fase più importante del ciclo di vita di un software, difatti non conseguendo a pieno questa fase non si riusciranno a risolvere le fasi successive. Dopo aver compreso ciò che bisogna svolgere, tutto questo verrà formalizzato attraverso l'analisi dei requisiti, con cui verranno definiti i requisiti. A tale proposito è possibile identificare due tipi di requisiti:

- 1) i requisiti funzionali, che rappresentano ciò che deve svolgere il sistema;
- 2) i requisiti non funzionali, che rappresentano le costrizioni, i vincoli del sistema, ad esempio dei vincoli imposti da alcune norme.

In genere, quando si parla dei requisiti, si possono avere dei livelli di definizione:

- 1) un primo livello in cui vengono stilati dei diagrammi e tutto viene espresso su un documento attraverso un linguaggio colloquiale;
- 2) nel secondo livello gli argomenti espressi nel primo livello vengono specificati, utilizzando linguaggi tecnici, definendo sempre requisiti funzionali e non funzionali.

Il primo livello mira all'aspetto manageriale, il secondo livello mira all'equipe di sviluppo e spesso il documento stilato nel secondo livello viene utilizzato come contratto tra le parti.

- 3) Esiste anche un terzo livello, il quale non fa parte dell'analisi dei requisiti, ed è rappresentato dalle specifiche del software da realizzare. Tale livello serve per la progettazione e per l'implementazione.

Dal primo e dal secondo livello viene elaborato un documento chiamato Software Require Specify (SRS). In questo documento viene dunque riportato ciò che deve svolgere il software da sviluppare.

Software Require Specify (SRS)

Per poter produrre un buon documento SRS, i requisiti devono essere chiari, corretti e consistenti (cioè non si devono contraddire). E' molto difficile giungere a questi tre scopi, in quanto non si conosce sempre tutto sin dall'inizio del progetto. L' SRS deve possedere caratteristiche come:

- 1) Deve riportare le costrizioni dell'implementazione;
- 2) Deve essere facile da modificare;
- 3) Deve riportare il ciclo di vita del software, cioè come si procede;
- 4) Deve prevedere delle risposte accettabili per eventi inaspettati, cioè deve prevenire il comportamento del sistema ad azioni inaspettate da parte dell'utente e/o esterni;
- 5) Deve riportare come viene visto all'esterno il sistema.

La tracciabilità

La tracciabilità è rappresentata da un'identificatore per ogni requisito, il quale servirà a risalire, dai successivi documenti, ad un possibile errore.

Modelli per esprimere i requisiti

E' possibile usufruire di diversi modelli per poter esprimere i requisiti di un software, questi sono:

- 1) Dataflow;
- 2) SADT;
- 3) Modello semantico;
- 4) Modello Object;
- 5) Modelli formali;
- 6) Metodo di Jacobson.

- 1) Per esprimere delle funzionalità, non a livello dettagliato, solo per capire il flusso dei dati (come entrano e come escono), si utilizzano dei diagrammi a blocchi, o Dataflow. Esistono diversi modi per esprimere il flusso dei dati:



indica una trasformazione



indica i dati sorgenti o i dati destinatari

Ci si può anche appoggiare ad una serie di Datastore, i quali sono dei registri che contengono informazioni utili per le elaborazioni. Si procede secondo un processo top – down, partendo da un unico diagramma e dettagliando progressivamente il problema, arrivando, ove necessario, a scomporlo in piccoli problemi più comprensibili. Il dataflow fornisce poche informazioni, in quanto non mostra le trasformazioni che avvengono. Esso è un modello semplice, dinamico e immediato, non adatto a sistemi complessi, inoltre risulta difficile il raggruppamento di funzionalità.

- 2) Il SADT (Structure Analysis and Design Techniques) si è sviluppato alla fine degli anni '70, evolvendosi fino all'85. Questo genere di diagramma possiede più regole. Dall'alto giungono i dati per il controllo (cioè se l'utente esiste, se la data immessa risulta esatta, ecc..), dal basso giungono invece i meccanismi, cioè gli strumenti che servono a risolvere le diverse attività. Il SADT ha sempre un approccio di tipo top – down, ma è possibile con esso osservare da più punti di vista. Nel SADT non si elaborano più di sei diagrammi per livello, e, nel caso in cui si debbano elaborare più diagrammi, allora si dividerà il problema in più sottoproblemi, costituendo in tal modo più livelli.

