

Utilità dell'ingegneria del software

Nel numero di febbraio del 1980 la rivista scientifica "The Computer Journal" semplificava, con le illustrazioni qui riprodotte, gli equivoci che si verificano nel corso di un progetto, dalla nascita dell'idea alla sua realizzazione.

L'esempio dell'altalena mostra bene come in un processo più che la potenza delle tecnologie adottate conti la comunicazione corretta tra i vari protagonisti. L'incomprensione tra i committenti di un sistema e il team di sviluppo può generare grossi guai, specie quando si tratta di costruire qualcosa di poco familiare (l'80% della spesa per riparare gli errori di un software è dovuto alle incomprensioni degli analisti nella raccolta dei requisiti).

Un committente affida ad esperti la realizzazione di una sua idea. Ha bisogno di un finanziatore, di un progettista di sistemi, di un programmatore. Il progetto è naturalmente più complicato dell'altalena appesa ad un albero rappresentata nei disegni, ma il riferimento a questa struttura semplice evidenzia le incomprensioni e le deviazioni dall'idea originale.

L'idea dell'utente è di un'altalena costituita da un copertone appeso al ramo di un albero con una corda (figura 1.)

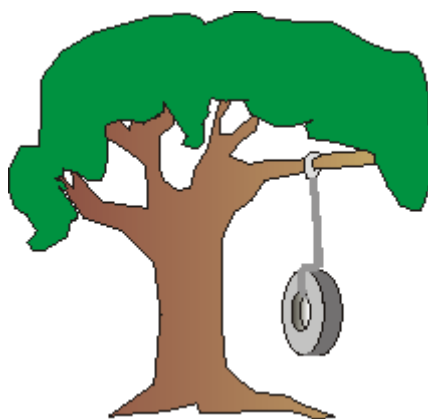


Figura 1 - Cosa vuole veramente l'utente

Nell'illustrare la sua idea, probabilmente perché nella sua testa sta prevalendo il modello più tradizionale di altalena, si capisce che lui desidera un copertone appeso con due funi (figura 2)



Figura 2: cosa ha detto di volere l'utente

Il colloquio con il finanziatore del progetto, deforma ulteriormente l'idea, probabilmente per far apparire il lavoro più importante ed ottenere un finanziamento più sostanzioso. Il finanziatore capisce che l'oggetto avrà la forma mostrata in figura 3.



Figura 3: **come l'idea è stata proposta al finanziatore**

In realtà analizzando obiettivamente la descrizione fatta al finanziatore del progetto, l'opera dovrebbe avere l'aspetto mostrato dalla figura 4



Figura 4: **come è stata descritta**

Il progettista del sistema raccogliendo le varie descrizioni realizza un'opera che ha l'aspetto della figura 5.



Figura 5: **come è stata progettata dal progettista**

Il programmatore che collabora con il progettista, comprende che una tavola che si appoggia per terra non servirebbe a nulla. Si accorge comunque che con il software può rimediare. Ma commette un errore e l'altalena realizzata non può dondolare come mostra la figura 6.



Figura 6: come è stata implementata dal programmatore

Scoprire il guaio software risulta difficile. Occorrerebbe rifare il progetto. L'operazione appare lunga. Il programmatore però scopre un trucco che permetterà il dondolarsi dell'altalena. È un trucco ignobile che porta ad una struttura assurda. Ma la disperazione di non poter rispettare i tempi è più forte. La realizzazione definitiva ha allora l'aspetto della figura 7: in fondo ora basta convincere il committente che, se desidera dondolarsi, deve stare attento ed abbassare la testa.



Figura 7: come è stata realizzata presso l'utente.

Conclusione

Sicuramente le vignette fanno sorridere ma, contemporaneamente, rappresentano un monito per studiare sempre bene in partenza il problema, imparando a descriverlo con precisione. Infatti la possibilità che su un progetto software avvenga ciò che si è appena descritto è quasi una certezza, ed è molto difficile rendersene conto, se non alla fine del lavoro.

Ciò che succede in un progetto software è che le funzioni, che si desidera vengano svolte da un programma, vengono generalmente descritte in modo superficiale, poiché il cliente ha le idee poco chiare. Nella realizzazione pratica vengono spesso dimenticate alcune parti e vengono male interpretate certe informazioni, mentre nella fase di verifica gli errori trovati vengono risolti il più delle volte in modo approssimativo e affrettato senza controllare l'impatto che certe modifiche potrebbero apportare ad altre parti del programma, con il rischio quindi di introdurre nuovi errori.

La conseguenza è la realizzazione di un programma che non rispetta le richieste del cliente, o meglio non fa quello che il cliente pensava di aver chiesto.

Per evitare questi problemi è nata l'ingegneria del software (**software engineering**), cioè lo studio della razionalizzazione dell'attività di programmazione attraverso metodi, tecniche e strumenti il più razionali e rigorosi possibili

Un'altra versione del classico esempio appena descritto

