

## **Pianificazione (Planning) di un progetto software**

Esistono diverse attività per la pianificazione di un progetto software, lo scopo di una pianificazione è la scomposizione del problema esposto in piccoli problemi più facilmente gestibili. Il capo progetto è responsabile di tutte le attività di pianificazione, per esempio:

- 1) Calendario delle attività da svolgere;
- 2) Organizzazione delle equipe;
- 3) Descrizione tecnica della soluzione proposta attraverso degli strumenti;
- 4) Descrivere quali standard, procedure, tecniche e strumenti utilizzare;
- 5) Svolgere un piano di qualità (una strategia di gestione della qualità), comprendente:
  - la gestione della configurazione
  - la gestione della documentazione
  - la gestione dei dati
  - la gestione delle risorse
- 6) Definire un piano di collaudo comprendente:
  - Training
  - Sicurezza
  - Gestione dei rischi
  - Mantenimento

In un piano di qualità si descrivono gli standard di qualità da utilizzare per il progetto: si può affidare ad una diversa equipe il compito di valutare la qualità di un prodotto e la valutazione si può basare su diversi standard. Per quanto riguarda la gestione della configurazione, essa gestisce il mantenimento delle diverse versioni di uno stesso software da sviluppare, tale attività può essere affidata ad un'equipe, più persone infatti possono lavorare ad un insieme di moduli attraverso degli strumenti. La gestione della documentazione prevede di stabilire il tipo di nomenclatura, il periodo di modifica e come deve avvenire questa modifica sulla documentazione. Questo procedimento è svolto anche quando si parla sia di gestione dei dati, in cui verrà descritto come i dati applicativi verranno manipolati, sia di gestione delle risorse, in cui si descrive come verrà gestito un hardware, se deve essere protetto, ecc...

In un piano di collaudo si attueranno dei collaudi di ordine grafico, del codice, della documentazione, ecc... fornendo in tal modo un ordine per i test da eseguire. Per quanto riguarda il training, esso tenta di gestire l'organizzazione delle persone a cui dover affidare un progetto, in questo caso, ad esempio, si pianifica un periodo di formazione per le persone che lavorano al progetto, oppure procedere al training dei clienti. La sicurezza riguarda i piani di sicurezza che si pianificano con lo scopo di proteggere dei dati riservati: si pianifica come svolgere tale protezione, di solito vi è un responsabile che si occupa di tutto questo. Si potrebbe anche dedicare una parte del piano di collaudo alla gestione dei rischi in cui si potrebbe incorrere. Inoltre si dovrebbe anche discutere del mantenimento del software sviluppato, prevedendo la pianificazione per la correzione dei bug riscontrati, decidendo chi lavorerà a tali correzioni, stando attenti a rispettare il budget.

## **I rischi**

I rischi sono eventi previsti che hanno un impatto negativo sul progetto. Quando si valutano i rischi si considerano tre fattori:

- 1) L'impatto del rischio ( I ): sono i danni e le conseguenze di un evento sul progetto, ad esempio una persona che risulta assente comporterà un ritardo non previsto. Tale fattore è generatore di danni al progetto.
- 2) La possibilità ( P ): si stima la probabilità che accada un rischio.
- 3) L'esposizione ( I \* P ): è il prodotto tra la probabilità e il danno che produrrebbe l'evento.

Quando si valutano i rischi si esegue un'analisi approfondita dei rischi (checkrisk), discutendo sul perché certe soluzioni non sarebbero efficaci. Si elencano i rischi maggiori, e si tenta di ridurre al minimo tali rischi, in quanto non è possibile eliminarli del tutto. Per esempio, nella spirale di Boehm si prevede di seguire un certo tipo di modello per ridurre i rischi. Boehm prevede dieci principali rischi e propone come diminuirli, in modo da acquisire protezione contro di essi.

## **La qualità**

Un processo software, come sappiamo, è composto da una serie di attività, alcune portanti ed altre ausiliarie. La qualità è definita come un'attività ausiliaria, tuttavia ormai è prassi comune considerarla come un'attività fondamentale. Quando si parla di qualità di un software, si include sia la correttezza del software sviluppato, sia il risultato ottenuto, sia lo sviluppo effettuato e l'ambiente di lavoro. Tale attività è stata intrapresa per la prima volta da Edward Denning, successivamente è stata considerata soprattutto in Giappone, dove si è cercato di ridurre i danni agendo sulle cause. Solamente negli anni '70 e '80 anche l'Europa si è dedicata a tale attività. Per evitare i problemi sorti in passato, l'attività di qualità viene integrata in un progetto software, tuttavia, tale attività prevede delle spese che comunque hanno un alto costo, che verranno recuperate in seguito, infatti si spenderà conseguenzialmente di più per rivedere il software e correggerlo, inoltre una correzione ad uno stato avanzato dello sviluppo comporterà degli effetti su molte più parti all'interno del progetto. E' possibile affidare la gestione di questa attività sia al capo progetto, sia ad un'equipe specializzata in qualità, per esempio l'SQA, o anche il TQM (Total Quality Management), le quali rappresentano delle etichette di qualità.

## **Strategia di gestione delle qualità**

E' possibile decidere di educare delle persone introdotte in seguito nel progetto riguardo gli standard di qualità da seguire, per tale motivo si effettuano degli stages in modo da sensibilizzare il personale sulla qualità.

## **Tecnologia (metodi e strumenti)**

Esistono degli strumenti utilizzati per verificare sul software gli standard di qualità, nonché per poter gestire la qualità della documentazione. Esistono inoltre delle procedure di qualità tese a garantire la conformità agli standard per lo sviluppo. A queste si aggiungono alcune metriche con cui poter verificare la qualità di alcuni fattori di un prodotto che di solito sono difficili da misurare. Secondo McCall esistono diversi fattori di qualità.

### **I fattori di qualità secondo McCall**

Correttezza: indica il grado di conformità del programma alle specifiche e agli obiettivi del cliente, cioè quando soddisfa le sue richieste.

Affidabilità: indica il grado in cui un programma svolge la propria funzione con la precisione richiesta. E' possibile che qualcosa di non previsto venga affrontata dal software, il quale dovrà

avere dei criteri che gli permettano di affrontare queste affermazioni. Si stila un range di variabilità all'interno del quale il software saprà affrontare la situazione e sarà quindi affidabile.

Efficienza: indica la quantità di risorse e di codice affinché il programma adempia la propria funzione.

Integrità = sicurezza: indica il grado in cui è possibile controllare l'accesso al software e ai dati da parte di persone non autorizzate.

Usabilità: indica lo sforzo necessario per apprendere ad utilizzare il programma, a prepararne i dati d'ingresso e ad interpretarne i dati d'uscita.

Manutabilità: indica lo sforzo necessario per localizzare e correggere un errore nel programma.

Flessibilità: indica lo sforzo necessario per modificare un programma in esercizio.

Provabilità: indica lo sforzo necessario per stabilire, tramite collaudo, se un programma svolge la funzione prevista.

Portabilità: indica lo sforzo richiesto per trasportare un programma da un ambiente, hardware o software, ad un altro.

Riusabilità: indica il grado in cui un programma può essere utilizzato di nuovo in altre applicazioni; è correlato al modo in cui il programma è assemblato ed è alla portata delle funzioni svolte dal programma.

Interoperabilità = interazione: indica lo sforzo richiesto per far interagire il programma con altri programmi.