Qualità nel software

Standard nell'informatica e metriche nel software: una panoramica

Antonio Candiello,
PhD Responsabile Qualità Met Sogeda

Premessa

Met Sogeda ha appena completato un articolato percorso verso la certificazione Iso 9001:2000 conosciuta anche come Vision 2000.

Nei primi giorni di dicembre 2003 è stata infatti superata con successo la verifica ispettiva iniziale per la Vision 2000 per quanto riguarda i processi di erogazione e progettazione relativi alle attività tecnologico/sistemistiche e software/applicative per i siti di Padova e Milano.

Il progetto per il passaggio a Vision 2000 è stato concepito nel 2002 e successivamente potenziato e ampliato nel corso del 2003 a comprendere le attività di sviluppo applicativo nel momento in cui è iniziato il processo di convergenza tra realtà sistemistica/tecnologica (Sogeda) e realtà applicativa/software (Met) che ha portato alla costituzione di Met Sogeda.

Va ricordato che l'azienda è certificata Iso 9001:1994 per le attività tecnologico/sistemistiche delle sedi di Milano e Padova sin dal 2001 e che per alcune unità organizzative si risale fino al 1998.

Nell'ambito di questo percorso sono state prese in esame, oltre alle norme di riferimento per la qualità, anche alcune norme o guide di settore applicabili al settore dell'informatica, allo scopo di avere una visione il più possibile completa degli standard allo scopo di garantire una maggiore qualità nei servizi e prodotti offerti ai clienti.

La norma Iso 9001:2000 rappresenta infatti uno standard applicabile nei più diversi settori merceologici, ma per approfondirne le modalità di applicazione in specifici settori, in questo caso nell'informatica, è opportuno fare riferimento a norme e guide specializzate.

Nella prima parte del presente articolo verranno pertanto citate le norme di riferimento emesse da diversi enti internazionali allo scopo di far comprendere il quadro complessivo di riferimento per il settore dell'informatica.

L'ampliamento del campo di applicazione della qualità al settore dei servizi applicativi e dello sviluppo software ci ha suggerito inoltre una ampia analisi delle metriche disponibili per il software, allo scopo di stabilire criteri oggettivi di misurazione della qualità in questo settore. Il processo di miglioramento continuo, che è la più grande innovazione della Iso 9001:2000 nei confronti della precedente edizione, rende necessario introdurre una cultura relativa agli indicatori oggettivi, che possano misurare l'*entità* del miglioramento.

Nel settore dei servizi tale cultura è molto diffusa, e tali indicatori sono utilizzati non solo per misurare il miglioramento, ma anche per definire e confrontare servizi che, in quanto attività immateriali, sono meno facilmente differenziabili.

Le attività relative al software richiedono invece un'analisi più approfondita. La teoria della misura del software è una scienza ricca e complessa, che presenta delle forti differenze nei confronti delle misure nel mondo fisico.

Essendo il software un prodotto - tra i più evoluti - dell'attività umana, ha delle leggi che ne rendono più difficile l'analisi oggettiva.

Nella seconda parte di questo articolo verrà quindi fatta una analisi sulle metriche utilizzabili nel software, sulla loro categorizzazione, e un approfondimento sulle metriche di prodotto.

Norme e guide relative all'informatica

Per affrontare consapevolmente il tema della qualità nel software è necessario valutare le norme di riferimento per i sistemi di gestione per la qualità, ovvero le Iso serie 9000, e le relative specializzazioni per il settore della produzione di software.

La *figura 1* sintetizza le norme di riferimento applicabili all'informatica.

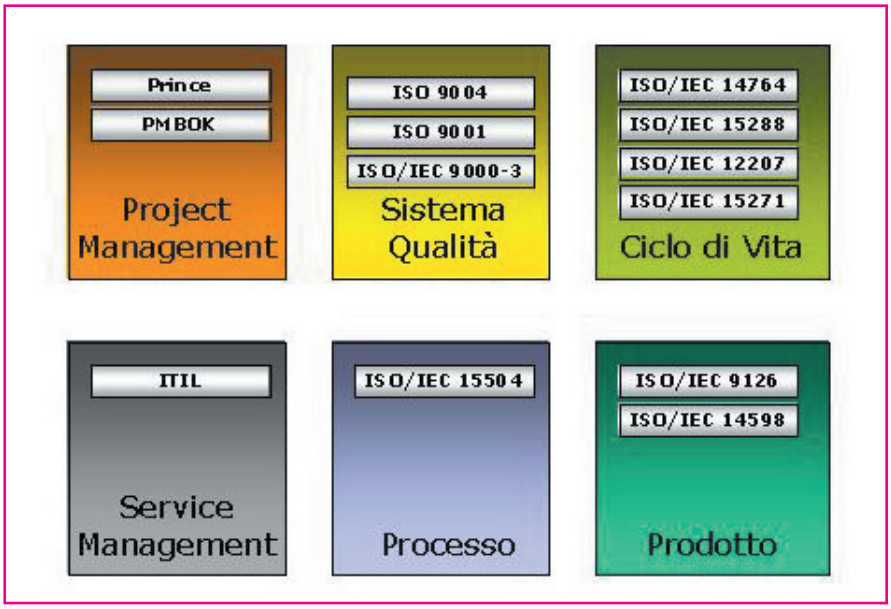


Figura 1 - Quadro di riferimento degli standard relativi all'informatica.

Le norme di riferimento per i sistemi di gestione per la qualità sono le seguenti:

• Uni En Iso 9000:2000 - nella quale sono presenti le definizioni relative ai sistemi di gestione della qualità ed i principi di base;

• Uni En Iso 9001:2000 - che rappresenta la *norma contrattuale* per i sistemi di gestione della qualità con l'indicazione dei requisiti;

• Uni En Iso 9004:2000 - che è la guida all'applicazione della Iso 9001 per la gestione del miglioramento continuo;

• Uni En Iso 19011 - nella quale vengono illustrate le modalità di gestione delle attività di auditing, applicabile ai sistemi di gestione per la qualità come anche per i sistemi di gestione ambientale.

Per quanto riguarda le relative specializzazioni per il settore della produzione di software va segnalata la norma:

• Uni En Iso 9000-3 :1998 - che è una guida per l'applicazione della norma Iso 9001 allo sviluppo, alla fornitura, all'installazione ed alla manutenzione del software per elaboratore.

Tale norma è il punto di riferimento per le aziende certificate che intendano progettare e sviluppare software secondo una metodologia che ne garantisca il risultato e rappresenta un approfondimento della Iso 9001 dove, punto per punto, ne viene suggerita l'applicazione al settore del software. Di tale norma si è in attesa dell'emissione della Uni En Iso 9000-3:2002 aggiornata allo schema Vision 2000.

Di seguito viene fornito, come suggerimento, l'elenco di alcune norme specializzate all'ambito dell'informatica:

• Uni Iso/Iec 14764:2000 "Tecnologia dell'informazione - Manutenzione del software",

• Uni Cei Iso/Iec 12207:2003 "Tecnologia dell'informazione - Processi del ciclo di vita del software",

• Iso/Iec Tr 15271:1998 "Information technology — Guide for ISO/IEC 12207"

• Iso/Iec 15288:2002 "System Engineering - System life cycle processes";

• Iso/Iec 14598:2001 parts 1-6 "Software engineering — Product evaluation - Parts 1-6".

Merita un approfondimento a parte, presente nella seconda parte di questo articolo, la:

• Iso/Iec 9126-1:2003 "Software engineering — Product quality — Parts 1-3", che classifica i requisiti di qualità per un prodotto software.

La progettazione, anche quella relativa all'informatica, può avvalersi degli standard internazionali più autorevoli:

• PRojects IN a Controlled Environment (Prince), emessa dal Central Computing and Telecommunications Agency inglese - ora in carico all'Office of Government Commerce del governo inglese.

• Project Management Body Of Knowledge (PMBOK) emessa dal Project Management Institute americano.

Mentre l'erogazione dei servizi in ambito informatico può convenientemente far uso degli standard emessi dagli organismi governativi inglesi:

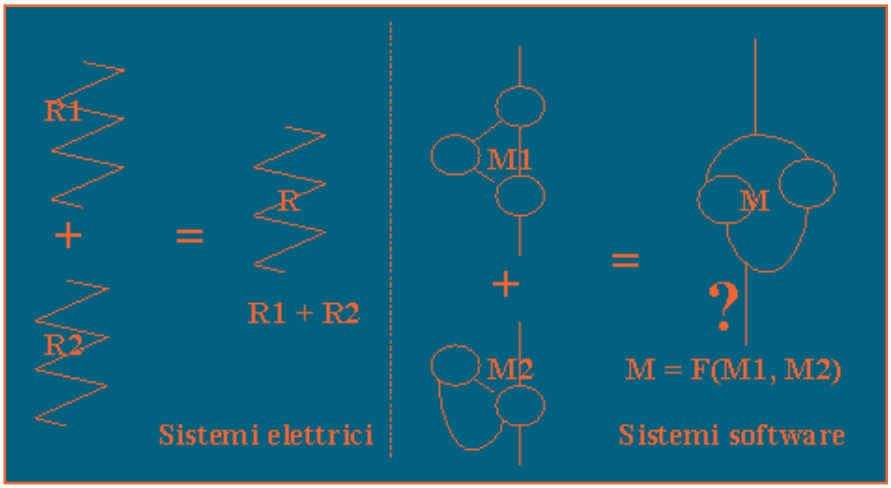
• Information Technology Infrastructure Library (ITIL), emessa dal Central Computing and Telecommunications Agency inglese - ora in carico all'Office of Government Commerce del governo inglese.

Di tale *library* segnaliamo i due testi di riferimento Service Support e Service Delivery, che insieme rappresentano la guida di riferimento denominata *Service Management* per la gestione dei sistemi informativi.

Le metriche nel software

Con il termine *metriche* si intendono parametri di misurazione di diversi aspetti relativi alla progettazione, sviluppo e mantenimento del software. L'introduzione delle metriche ha l'obiettivo di introdurre un modello di oggettività scientifica nelle attività di sviluppo software, in analogia con la scienza della misura nei sistemi fisici. L'ingegneria del software presenta tuttavia

delle caratteristiche che ne rendono più difficile la misurabilità rispetto ad ambiti più noti, come ad esempio i sistemi fisici.
Per presentare un'analogia, mentre in un sistema di reti elettriche è possibile utilizzare delle grandezze, come la resistenza, che si combinano secondo leggi precise aggregando sistemi diversi, evidenziare grandezze dotate di proprietà simili nel software non è scontato.
La difficoltà nel costruire metriche affidabili e di facile utilizzo è legata al fatto che una metrica "ideale" dovrebbe riuscire ad astrarre dal codice la logica dell'algoritmo sottostante eliminando ogni dipendenza dal linguaggio di programmazione e dallo stile di programmazione utilizzato.
Le metriche nel software sono da tempo oggetto di studio da parte sia del mondo accademico, sia delle imprese che hanno la necessità di mantenere controllati i parametri di riferimento dello sviluppo software.
Le metriche possono misurare sia aspetti *quantitativi* per garantire la necessaria *efficienza* dei processi di produzione del software finalizzati all'ottenimento di adeguate *soglie di produttività* e di *scalabilità,* sia aspetti qualitativi per garantire la necessaria *efficacia* dei processi di produzione del software finalizzati al mantenimento di determinate *soglie* di riferimento dei parametri di *qualità.*
L'utilizzo delle metriche permette di dare oggettività ai processi relativi al software.
La misurazione dei parametri quantitativi e qualitativi legati al software permette infatti di:
• stabilire e verificare a intervalli prestabiliti la qualità dei risultati delle attività di sviluppo e manutenzione e porla in relazione alla quantità del codice prodotto;
• definire e controllare le soglie di riferimento per alcuni parametri di qualità del software;
• stabilire in anticipo l'entità di lavoro necessario, utilizzando le metriche dimensionali (come ad esempio i *function points*) e nel contempo chiarire i requisiti;
• valutare oggettivamente la produttività di determinate unità organizzative e il livello qualitativo medio del software da esse prodotto;
• controllare e validare i prodotti software in uscita dai processi manutentivi e di sviluppo;
• fornire ai clienti elementi di garanzia per gli interventi sui loro sistemi.
L'adozione di criteri oggettivi di misurazione del software permette inoltre di attribuire valutazioni economiche relative agli interventi che si intendano progettare sul software e quindi di stabilire la convenienza di tali interventi.



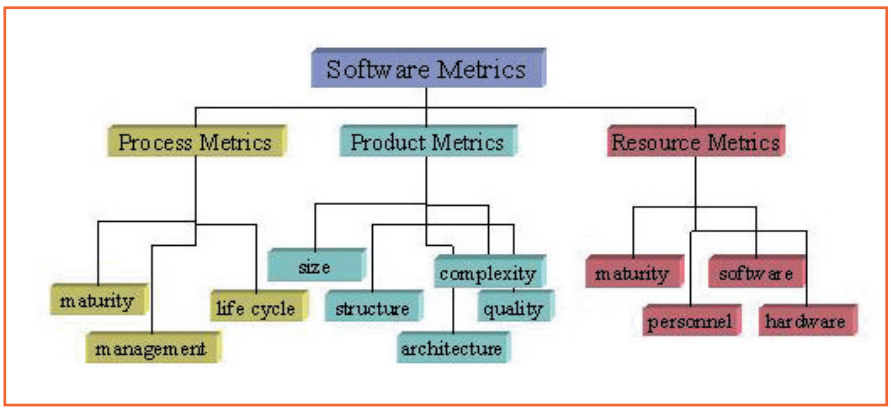


Figura 2 - Differenze tra sistemi fisici e sistemi software.
Figura 3 - Classificazione delle metriche nel software.

Le metriche nel software sono *classificabili* nelle categorie riportate nella *figura 3.*
Le metriche sono suddivise in metriche di *processo,* per la misurazione della comprensione, valutazione e miglioramento dei metodi di sviluppo, in metriche di *prodotto,* per la misurazione delle caratteristiche del prodotto software, ed in metriche relative alle *risorse,* per la misurazione dei supporti allo sviluppo.
Nel seguito approfondiremo il caso più interessante, quello delle *metriche di prodotto.* Con il termine "metriche" si intendono infatti generalmente le sole metriche di prodotto, tra le quali si individuano le

size metrics e le *complexity metrics,* strumenti di misurazione quantitativa del software, le *quality metrics* della norma Iso 9126, ed infine le *architecture metrics* e le *structure metrics.*

Concentrandoci sulle prime tre categorie, e iniziando con le metriche dimensionali e di complessità, va detto che queste rappresentano il primo strumento utilizzato per valutare l'impegno necessario per lo sviluppo.

La metrica più nota in tal senso è rappresentato dal semplice conteggio delle righe di codice - la *LOC,* Lines Of Code.

E' ben noto che tale metrica, nonostante rimanga la più utilizzata in assoluto, presenti delle marcate dipendenze con il linguaggio utilizzato e con lo stile di programmazione (la *verbosità* influisce infatti in maniera sensibile sulle LOC misurate). E' quindi inadatta per stimare l'entità di risorse da impegnare nello sviluppo di un'applicazione e parimenti assolutamente inadeguata a misurare la produttività - non considera la duplicazione, né il riuso del software.

Una metrica introdotta per superare le limitazioni delle LOC ed in grado di misurare la complessità del codice è la *metrica ciclomatica* di McCabe. Tale metrica misura la complessità delle strutture di controllo del codice, calcolando il parametro *V(G)* ottenuto valutando il numero di percorsi indipendenti nel grafo G derivato dal codice. *V(G)* cresce aumentando la nidificazione dei cicli condizionali e aumentando i *loop* e rappresenta un indicatore della difficoltà di effettuare i test di affidabilità.

La metrica *funzionale* dei *function points* è calcolata invece conteggiando i diversi flussi di input, output, i file e altri parametri delle routines. E' una metrica utilizzabile già dopo l'analisi - prima dell'implementazione - indipendente dal linguaggio, e questa caratteristica la rende molto popolare nella comunità informatica. Come aspetti negativi di questa metrica va segnalata la necessità di una complessa elaborazione, prevalentemente manuale.

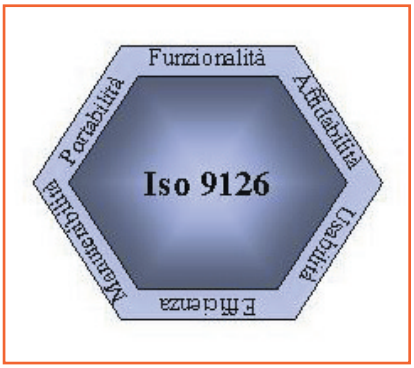


Figura 4 - Le sei caratteristiche della norma Iso 9126 sulla qualità del prodotto software.

L'utilizzo metodologie di sviluppo *orientato agli oggetti* permette l'introduzione di metriche specializzate per valutare aspetti dimensionali e di complessità relativi alle classi, alle funzioni contenute, ai diagrammi di ereditarietà e agli schemi di connessione tra di esse. Le metriche hanno naturalmente come obiettivo quello di astrarsi dal linguaggio e di misurare caratteristiche della programmazione a oggetti indipendente dal linguaggio.

Citiamo di seguito alcuni esempi:

- *DIT Depth Inheritance Tree*, per misurare la profondità media degli alberi di ereditarietà: il suo incremento segnala un potenziale comportamento maggiormente imprevedibile ma nel contempo un riuso elevato;
- *NOC Number Of Children,* per misurare l'aggregazione tra classi, indice di riuso crescente ed astrazione calante;
- CBO/CRC *Classi, Responsabilità e Collaborazioni,* indici che misurano il grado di interrelazione tra classi;
- *CS Class Size*, misura il numero di funzioni & attributi per classe - indicatore di "globalità".

Un uso accorto di tali misuratori consente di istituire un processo di sviluppo controllato stabilendo e verificando periodicamente il desiderato grado delle caratteristiche object-oriented del software in corso di sviluppo e di garantire quindi di conseguenza che siano realmente disponibili i relativi benefici in termini di manutenibilità, riuso e portabilità del codice.

Per quanto riguarda le *quality metrics*, l'obiettivo è la misurazione oggettiva delle caratteristiche di qualità del prodotto software risultante dal processo di sviluppo. Il modello stabilito dalla norma Iso 9126 classifica i requisiti in sei caratteristiche principali che a loro volta vengono strutturate in ulteriori aspetti di secondo livello.

Le caratteristiche principali riguardano:

- *Funzionalità:* misura il grado in cui il prodotto garantisce le funzioni richieste;
- *Affidabilità:* misura la capacità del prodotto di garantire le funzionalità sotto determinate condizioni;
- *Usabilità:* misura la facilità d'utilizzo del prodotto;
- *Efficienza:* misura il rapporto tra prestazioni e utilizzo delle risorse;
- *Manutenibilità:* misura la facilità nella modifica del prodotto;
- *Portabilità:* misura la facilità di trasferire il prodotto su differenti sistemi/ambienti operativi.

Considerazioni conclusive

Come si è visto in queste brevi note, la qualità nell'informatica è un settore già molto articolato e interessante; le novità in questo settore continuano peraltro ad apparire con notevole frequenza, trainato dall'effervescenza tipica dell'informatica, e quindi suggeriamo di seguire con continuità e frequenza il tema, garantendo così un adeguato aggiornamento delle conoscenze e dei processi. ❑