

I modelli di qualità del software.

Le norme ISO/IEC 9126,
25010, 12119, 9241.

Gianluigi Raiss
Area Indirizzo, Supporto e Verifica PAC
Ufficio Pianificazione e Razionalizzazione
raiss@cnipa.it
<http://www.cnipa.gov.it>

I modelli di qualità del software

Definizione di software

- Cosa si intende per prodotto software?
 - ◆ *Una serie di istruzioni fornite ad un computer per fargli realizzare determinate operazioni rispettando dei requisiti dati da un cliente.*
 - ◆ I requisiti possono riguardare sia le funzioni che il software deve svolgere, sia aspetti non funzionali (prestazioni che deve assicurare, capacità di essere *user friendly* grazie ad una grafica particolare, accessibilità da parte dei disabili, tecnologia da utilizzare, sicurezza etc...).
 - ◆ Dalla ISO 9126: prodotto software è *“l’insieme di programmi, procedure, regole, documenti, pertinenti all’utilizzo di un sistema informatico”*.

Struttura dei modelli di qualità

- I modelli di qualità del software pubblicati in letteratura sono tipicamente a n livelli.
 - ◆ Il primo livello descrive un insieme di caratteristiche (*proprietà*), che, nel loro complesso, rappresentano la qualità del prodotto software, eventualmente secondo diversi punti di vista.
 - ◆ Le proprietà (in genere qualitative, astratte) sono precisate attraverso delle sottocaratteristiche misurabili, quantitative (in genere da una combinazione di sottocaratteristiche).
 - ◆ Il grado di possesso che il software ha di questi attributi può essere valutato su una scala di riferimento, facendo ricorso ad opportune *metriche* ed a meccanismi di *rating*.

I modelli di Mc Call e Boehm

- I primi modelli di qualità del software sono stati sviluppati negli anni '70 da McCall (*Factor-Criteria Model*, 1977) e da B. Boehm (1978). Hanno un'architettura a 3 livelli:
 - ◆ *Fattori*, che descrivono il software da un punto di vista esterno, quello degli utenti; i fattori corrispondono a dei requisiti che sono specificati dal cliente.
 - ◆ *Criteri*, che descrivono gli elementi su cui agiscono gli sviluppatori per corrispondere ai requisiti del cliente.
 - ◆ *Metriche*, che servono a controllare che i criteri sviluppati corrispondano ai fattori specificati. Vengono utilizzate dagli auditors e/o dagli addetti alle verifiche.

Struttura del modello di Mc Call



11 Fattori e 23 criteri

Product operation: è l'insieme delle caratteristiche del software evidenti nella fase in cui esso è in servizio;

Correttezza: se il software raggiunge il suo scopo;

Affidabilità: se il software svolge sempre correttamente il suo compito;

Efficienza: se il software lavora efficientemente in qualsiasi contesto;

Integrità: se il software conserva in sicurezza i dati e la loro consistenza;

Usabilità: quanto il software è usabile;

Product revision: è l'insieme delle caratteristiche evidenti quando si vanno ad attuare delle modifiche sul software;

Manutenibilità: se è semplice mantenere il software operativo;

Flessibilità: se il software è facilmente modificabile per renderlo capace di svolgere nuovi compiti;

Testabilità: se il software è facilmente testabile;

Product transition: l'insieme delle caratteristiche evidenti quando il software viene fatto operare su un nuovo dominio tecnologico (ad esempio una nuova piattaforma hardware);

Portabilità: se il software è facilmente portabile verso varie piattaforme;

Riusabilità: se il software o parte di esso può essere facilmente riusato;

Interoperabilità: se il software è facilmente interfacciabile con altri

Fattori di qualità come funzioni di criteri

Correttezza

La correttezza del software è definibile come il suo grado di adesione agli standard definiti sia nel processo produttivo che nel suo dominio di applicazione.

Correttezza = | (tracciabilità, coerenza, completezza)

Tracciabilità: il grado di reperibilità delle varie specifiche richieste dal software all'interno del codice o in fase di esercizio;

Coerenza: quegli attributi che si riferiscono all'uniformità delle tecniche utilizzate dai requisiti al software;

Completezza: se il software soddisfa tutti i requisiti per cui è stato sviluppato.

Fattori di qualità come funzioni di criteri

Affidabilità

L'affidabilità è definibile come la capacità del software di svolgere correttamente il suo compito nel tempo.

Affidabilità = | (tolleranza all'errore, coerenza, accuratezza, semplicità)

Tolleranza all'errore: il grado di affidabilità dei risultati ottenuti dal software in presenza di condizioni non standard;

Accuratezza: ossia quanto sono precisi i risultati in output;

Semplicità: se il software implementa le sue funzioni in maniera chiara e comprensibile.

Fattori di qualità come funzioni di criteri

Efficienza

Quanto il software svolge il suo compito con minor spreco di risorse (tempi di elaborazione e di comunicazione, spazio di memoria, ecc.);

Efficienza = \downarrow (efficienza di esecuzione, efficienza di memorizzazione)

Efficienza di esecuzione: il tempo impiegato per svolgere il compito richiesto;

Efficienza di memorizzazione: la quantità di spazio occupato in memoria dai dati.

Fattori di qualità come funzioni di criteri

Manutenibilità

E' definita come lo sforzo necessario per trovare e correggere un errore all'interno del codice dopo il rilascio in esercizio al cliente;

Manutenibilità = \downarrow (coerenza, semplicità, concisione, modularità, auto documentazione)

Concisione: la quantità di codice necessaria per adempire ad una certa funzione;

Modularità: il grado di indipendenza dei vari moduli operanti all'interno del software;

Auto documentazione: la capacità del codice di spiegare le funzioni implementate.

Fattori di qualità come funzioni di criteri

Testabilità

E' la facilità con cui è possibile effettuare un testing sull'applicazione;

Testabilità = | (semplicità, modularità, strumentazione, auto documentazione)

Strumentazione

E' la facilità con cui è possibile monitorare il funzionamento del software e quindi verificarne possibili errori.

Fattori di qualità come funzioni di criteri

Portabilità

E' la capacità di adattamento del software ad operare su nuovi ambienti di lavoro (nuovo hardware, software, configurazioni)

Portabilità = | (modularità, auto documentazione, indipendenza dalla macchina, indipendenza dal software)

Indipendenza dalla macchina: gli attributi che definiscono l'indipendenza del software dalla piattaforma hardware.

Indipendenza dal software: gli attributi che definiscono il grado di indipendenza del software dall'ambiente software in cui opera, quindi indipendenza dal sistema operativo, dai drivers, utility, ecc..

Collisioni e relazioni tra criteri e fattori

Alcuni fattori di qualità operano in sinergia, mentre altri sono in conflitto fra loro. Sinergie e conflitti fra alcuni fattori di qualità sono sintetizzati nella seguente tabella.

	C	A	E	I	U	M	T	F	P	R	IO
Correttezza (C)	C										
Affidabilità (A)	S	A									
Efficienza (E)			E								
Integrità (I)				C	I						
Usabilità (U)	S	S	C	S	U						
Manutenibilità (M)	S	S	C		S	M					
Testabilità (T)	S	S	C		S	S	T				
Flessibilità (F)	S	C	C	C	S	S	S	F			
Portabilità (P)			C			S	S		P		
Riusabilità (R)		C	C	C		S	S	S	S	R	
Interoperabilità (IO)			C	C					S		IO

C = conflitto, S = sinergia

Le viste logiche nel modello di McCall

VISTA UTENTE

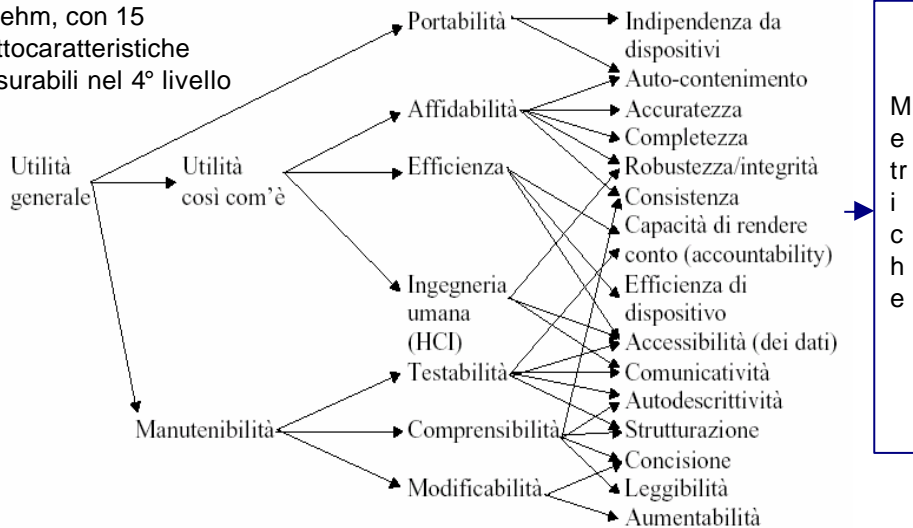
Correttezza
Affidabilità
Efficienza
Integrità
Usabilità
Manutenibilità
Testabilità
Flessibilità
Portabilità
Riusabilità
Interoperabilità

VISTA SVILUPPATORE

Tracciabilità
Completezza
Consistenza
Accuratezza
Tolleranza agli errori
Efficienza di esecuzione
Efficienza di memorizzazione (storage)
Controllo dell'accesso
Audit dell'accesso
Operabilità
Addestramento
Comunicatività
Semplicità
Concisione
Strumentazione
Autodescrittività
Espandibilità
Generalità
Modularità
Indipendenza dal sw di sistema
Indipendenza dalla macchina
Uso di comunicazioni comuni (common)
Uso di dati comuni (common)

I livelli nel modello di Boehm

5 livelli nel modello di Boehm, con 15 sottocaratteristiche misurabili nel 4° livello



Limiti dei modelli di McCall e Boehm

- I modelli di McCall e Boehm hanno dei limiti:
 - ◆ è difficile che le caratteristiche e sottocaratteristiche siano sempre tra loro perfettamente indipendenti,
 - ◆ manca in ogni caso il legame esplicito tra il modello qualitativo e “come” fare poi del buon software,
 - ◆ le caratteristiche sono in genere proprietà astratte misurabili solo attraverso indicatori e metriche. Non sempre l'andamento di queste grandezze è in correlazione perfettamente lineare con le caratteristiche che devono stimare.

I modelli ISO per la qualità del software

- La ISO ha emesso alcune norme che definiscono il modello di qualità del software, in vari contesti di utilizzo.
 - ◆ ISO/IEC 9126-1 - Definisce 6 caratteristiche di qualità principali (astratte) e 27 sottocaratteristiche misurabili attraverso delle metriche, fornite in 3 *technical reports* (ISO/IEC 9126-2, 3 e 4).
 - ◆ ISO/IEC 9241 - Definisce le caratteristiche di qualità di un software “usabile”.
 - ◆ ISO 12119 - Definisce le caratteristiche di qualità di un software “*Commercial off the shelf*” (COTS).
- La ISO ha anche pubblicato una norma, la 14598, che guida alla valutazione della qualità del software, definita secondo i criteri 9126.



La norma ISO/IEC 9126

La norma ISO/IEC 9126

- La norma ISO/IEC 9126 *Software engineering - Product quality* contiene il modello di riferimento per definire le caratteristiche di qualità del software e le metriche per la valutazione della qualità che il software possiede.
 - ◆ E' stata pubblicata la prima volta nel 1991, l'ultima tra il 2001 ed il 2004. E' in corso di ulteriore revisione.
 - ◆ Si può utilizzare per definire i requisiti qualitativi del software e per definire le misure che vanno rilevate per valutare la qualità di un software.
 - ◆ Quanto definito dalla 9126 è applicabile a qualsiasi tipo di software (Custom o COTS).
 - ◆ Non sono previste certificazioni di qualità del prodotto software rispetto a questa norma.

Struttura della norma ISO/IEC 9126

- La norma si compone di queste parti:
 - 1) il modello delle caratteristiche e sottocaratteristiche di qualità del software (ISO/IEC 9126-1 *Software Engineering. Product Quality - Part 1: Quality model, 2001*)
 - 2) le metriche per la misura della qualità esterna (ISO/IEC TR 9126-2, 2003)
 - 3) le metriche per la misura della qualità interna (ISO/IEC TR 9126-3, 2003)
 - 4) le metriche per la misura della qualità in uso (ISO/IEC TR 9126-4, 2004).

Utilizzo della ISO/IEC 9126

- La 9126 può essere utilizzata da acquirenti, sviluppatori, auditors, addetti all'assicurazione qualità, secondo diverse prospettive, per vari scopi, tra cui:
 - ◆ definire i requisiti di qualità del software (*quality goals*) nei capitolati tecnici e/o nelle offerte tecniche che vengono prodotte in risposta ai capitolati,
 - ◆ validare la completezza di un documento di specifica dei requisiti e/o progettazione del software,
 - ◆ identificare obiettivi che vanno soddisfatti dal disegno tecnico del software,
 - ◆ identificare criteri di assicurazione della qualità (v. ISO 90003),
 - ◆ identificare criteri di accettazione, test e collaudo del software.

La ISO WD 25010 (2006)

- La norma ISO/IEC 9126 è in corso di revisione.
- Il modello di qualità del software confluirà nel sistema di norma ISO 25000, nella norma 25010, senza modifiche.
- Le metriche saranno definite nella sezione ISO 2502n, composta da più documenti che supporteranno la misurazione del software secondo i diversi punti di vista della qualità interna, esterna ed in uso, fornendo:
 - ◆ un modello di riferimento per rilevare le misure,
 - ◆ definizioni matematiche delle varie misure, primitive di misura, misure derivate,
 - ◆ una guida pratica alla misurazione.

Struttura del modello 9126

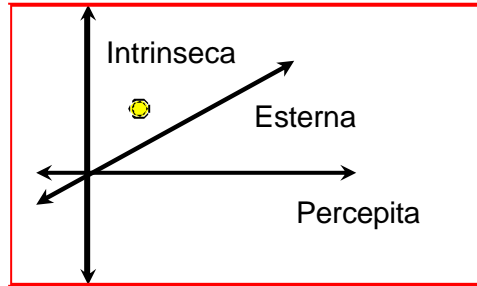
- La 9126-1 definisce un modello a 4 livelli:
 - ◆ tre “punti di vista” sulla qualità del software (*esterno, interno ed in uso*) che qualsiasi progetto di sviluppo deve prendere in considerazione e soddisfare,
 - ◆ i principali attributi (definiti come requisiti qualitativi) che qualificano un software, secondo i 3 punti di vista,
 - ◆ per ogni attributo, le sottocaratteristiche (requisiti quantitativi, misurabili) che la rappresentano e che dovranno essere misurate per valutare il livello di qualità effettivamente raggiunto nel software,
 - ◆ le metriche per effettuare le misure.
- Il processo di valutazione è definito nella ISO 14598.

ISO 9126 - glossario

- Qualità del software:
 - ◆ *“l’insieme delle caratteristiche che incidono sulla capacità del prodotto di soddisfare requisiti espliciti od impliciti”.*
- Prodotto software:
 - ◆ *“l’insieme di programmi, procedure, regole, documenti, pertinenti all’utilizzo di un sistema informatico”.*
- Contesto d’uso:
 - ◆ *“l’insieme di utenti, procedure, risorse hardware, software o daltra natura, l’ambiente fisico e sociale nel quale il prodotto è utilizzato”.*

I punti di vista sulla qualità

A determinare la qualità complessiva di un software concorrono 3 punti di vista.



PERCEPITA (in uso)

Esprime l'efficacia ed efficienza con cui il software serve le esigenze dell'utente, ed è correlata alla percezione diretta dell'utente.

INTERNA (intrinseca)

Esprime la misura in cui il codice software possiede una serie di attributi statici, indipendentemente dall'ambiente di utilizzo e dall'utente.

ESTERNA

Esprime il comportamento dinamico del software, nell'ambiente d'uso.

La qualità esterna

- La **qualità esterna** è quella rappresentata dalle prestazioni del prodotto e dalle funzionalità che offre (il prodotto è visto come una *black box* da testare).
 - ◆ In sostanza, riguarda il comportamento “dinamico” del software, in un dato ambiente operativo e dipende strettamente dai requisiti del cliente.
 - ◆ Va ricordato che il software non “funziona” mai da solo, ma è sempre parte di un ambiente (*environment*) che può contenere hardware, persone, processi etc...
 - ◆ Le caratteristiche di qualità esterne del software lo qualificano in relazione a questo ambiente e permettono di osservarne il comportamento mentre è utilizzato operativamente.

La qualità interna

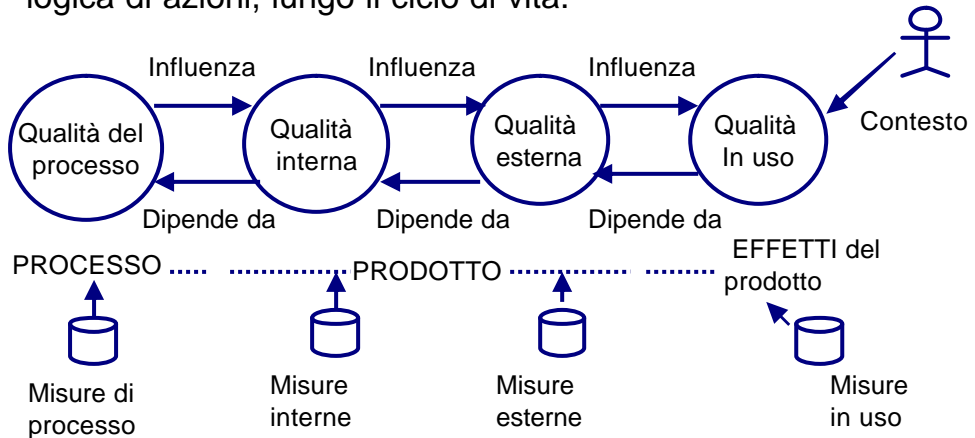
- La **qualità interna** rappresenta le proprietà intrinseche del prodotto (quelle misurabili direttamente sul codice sorgente, sul suo flusso di controllo). Si realizza a partire da:
 - ◆ i requisiti dell'utente (**External Quality Requirements**), che rappresentano le specifiche di qualità così come le dà l'utente, fornendo il primo input alla progettazione,
 - ◆ le specifiche tecniche (**Internal Quality Requirements**), che rappresentano la qualità richiesta dall'utente tradotta dallo sviluppatore nell'architettura del software, nella struttura del programma, nelle interfacce del software verso l'utente.

La qualità in uso

- La **qualità in uso** riguarda il livello con cui il prodotto si dimostra utile all'utente nel suo effettivo contesto d'utilizzo, in particolare la capacità del prodotto di dare efficacia ed efficienza al lavoro dell'utente, a fronte di una sicurezza di utilizzo e di una soddisfazione nel far uso del prodotto.
- In sostanza, è una misura della interazione tra utente e prodotto, in un determinato contesto d'uso.
- ◆ I tre punti di vista sulla qualità si influenzano a vicenda: è chiaro che non può esservi qualità percepita positivamente dall'utente senza che vi sia una buona qualità intrinseca al codice e buone prestazioni!

Qualità del software nel ciclo di vita

- La qualità del software, come percepita dall'utente, si determina progressivamente attraverso una sequenza logica di azioni, lungo il ciclo di vita.



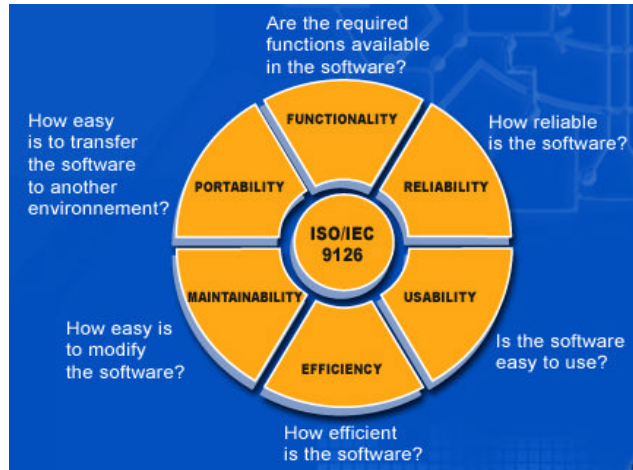
La qualità nel ciclo di vita

- Il focus sulla qualità cambia durante il ciclo di vita.
- ◆ All'inizio, durante la raccolta ed analisi dei requisiti, la qualità è specificata dai requisiti utente, soprattutto da un punto di vista "esterno" e funzionale.
- ◆ Nella fase di progettazione, la qualità esterna si traduce in un disegno tecnico, confrontandosi con il punto di vista degli sviluppatori sulla qualità interna del software e completandosi con i requisiti "impliciti" che il software deve rispettare.
- ◆ La qualità finale (quella in uso) deve essere quella appropriata per gli utenti ed il contesto d'uso. Non esiste una qualità perfetta ed "assoluta". Esiste quella *necessaria* e *sufficiente* per un dato contesto.

Caratteristiche esterne ed interne

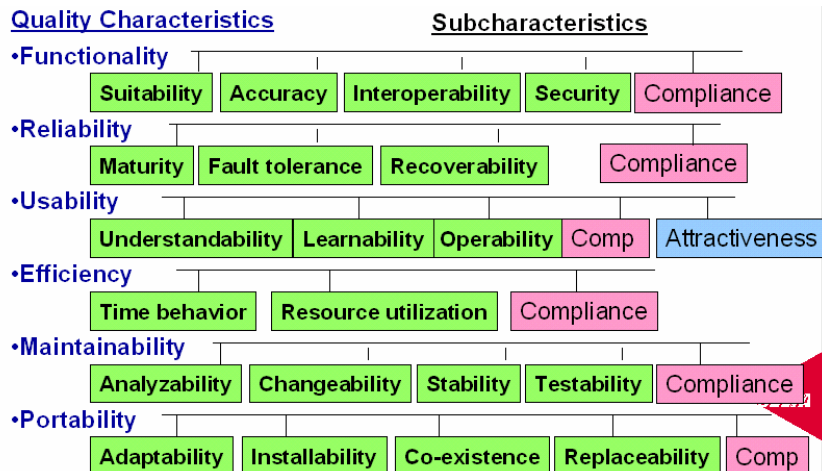
Le caratteristiche che rappresentano la qualità esterna ed interna di un prodotto software sono 6.

- ◆ FUNZIONALITÀ
- ◆ AFFIDABILITÀ
- ◆ USABILITÀ
- ◆ EFFICIENZA
- ◆ MANUTENIBILITÀ
- ◆ PORTABILITÀ



Le sottocaratteristiche

Le 6 caratteristiche principali sono ulteriormente specificate da 27 sotto-caratteristiche.



Le caratteristiche interne/esterne...

1) **Funzionalità**: capacità di fornire servizi tali da soddisfare, in determinate condizioni, requisiti funzionali espliciti o impliciti (il software fa ciò per fare il quale è stato acquistato). Le sottocaratteristiche correlate sono:

- ◆ **Adeguatezza**: presenza di funzioni appropriate per compiti specifici che supportano obiettivi dell'utente.
- ◆ **Accuratezza**: capacità di fornire risultati corretti in accordo con i requisiti dati dall'utente.
- ◆ **Interoperabilità**: capacità di interagire con altri sistemi.
- ◆ **Sicurezza**: capacità di proteggere programmi e dati da accessi non autorizzati e di consentire quelli autorizzati.

....le caratteristiche 9126-1...

2) **Affidabilità**: capacità di mantenere le prestazioni stabilite nelle condizioni e nei tempi fissati (il software reagisce bene a variazioni esterne); le sottocaratteristiche correlate sono:

- ◆ **Maturità (robustezza)**: capacità di evitare fermi della applicazione (*failure*) a seguito di malfunzioni (*fault*).
- ◆ **Tolleranza errori**: capacità di mantenere determinati livelli di prestazione in caso di malfunzioni (non degrado).
- ◆ **Recuperabilità**: capacità e velocità, in caso di malfunzioni, di ripristinare dei livelli di prestazione predeterminati e di recuperare i dati.

....le caratteristiche 9126-1...

3) **Usabilità**: capacità di essere compreso, appreso, usato con soddisfazione dall'utente in determinate condizioni d'uso (il software gestisce bene l'interazione con gli utenti); le sottocaratteristiche correlate sono:

- ◆ **Comprensibilità**: capacità di ridurre l'impegno richiesto agli utenti per capirne il funzionamento e le modalità di utilizzo.
- ◆ **Apprendibilità**: capacità di ridurre l'impegno richiesto agli utenti per impararlo ad usare.
- ◆ **Operabilità**: capacità di mettere in condizione gli utenti di farne uso per i propri scopi e controllarne l'uso.
- ◆ **Attrattività/Piacevolezza**: capacità di essere piacevole per l'utente che ne fa uso.

....le caratteristiche 9126-1...

4) **Efficienza**: rapporto fra prestazioni e quantità di risorse utilizzate, in condizioni definite di funzionamento (il software usa bene le risorse disponibili); le sottocaratteristiche correlate sono:

- ◆ **Comportamento rispetto al tempo**: adeguati tempi di risposta, tempi di elaborazione e throughput rates per eseguire le funzioni richieste, sotto determinate condizioni.
- ◆ **Uso di risorse**: utilizzo di una quantità e di una tipologia di risorse adeguate per eseguire le funzioni richieste, sotto determinate condizioni.

...le caratteristiche 9126-1...

5) **Manutenibilità**: capacità di essere modificato con un impegno contenuto (per evoluzioni e/o correzioni o adeguamenti); le sottocaratteristiche correlate sono:

- ◆ **Analizzabilità**: capacità di limitare impegno richiesto per diagnosticare carenze o cause di malfunzionamenti, o per identificare parti da modificare.
- ◆ **Modificabilità**: capacità di limitare l'impegno richiesto per modificare, rimuovere errori o sostituire componenti.
- ◆ **Stabilità**: capacità di ridurre il rischio di comportamenti inaspettati a seguito della effettuazione di modifiche.
- ◆ **Testabilità**: capacità di essere facilmente testato per validare le modifiche apportate.

....le caratteristiche 9126-1

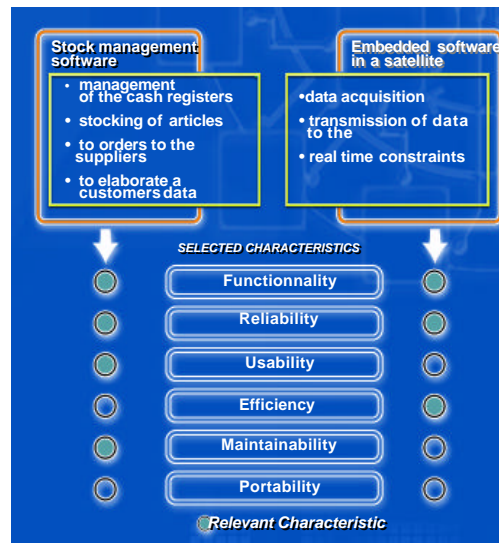
6) **Portabilità**: facilità con cui il software può essere trasferito da un ambiente operativo ad un altro; le caratteristiche correlate sono:

- ◆ **Adattabilità**: capacità di adattarsi a nuovi ambienti operativi limitando la necessità di apportare modifiche.
- ◆ **Installabilità**: capacità di ridurre l'impegno richiesto per installarlo in un particolare ambiente operativo.
- ◆ **Coesistenza**: capacità di coesistere con altri software nel medesimo ambiente, condividendo risorse.
- ◆ **Sostituibilità**: capacità di essere utilizzato al posto di un altro software per svolgere gli stessi compiti nello stesso ambiente.

Ogni software ha proprie caratteristiche

Le caratteristiche di qualità che un software deve possedere sono in genere diverse da prodotto a prodotto.

Un software gestionale, che organizza le attività di un magazzino, ed un software di gestione di un satellite, hanno esigenze di qualità ben diverse.



Qualità in uso

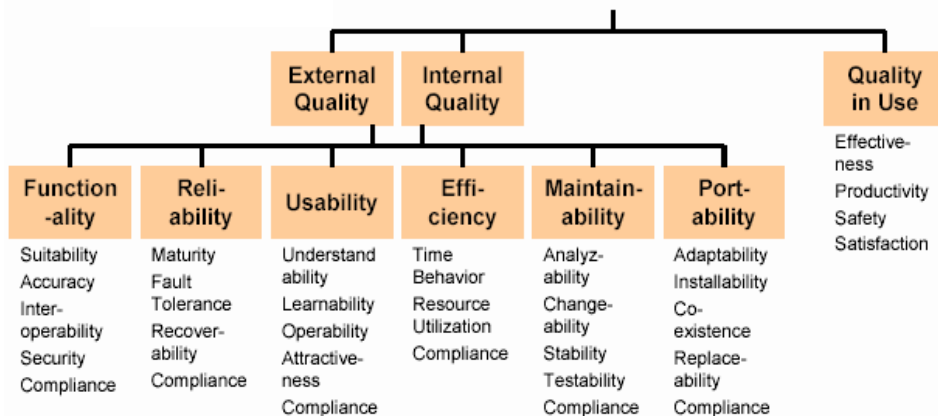
La **qualità in uso** è rappresentata da 4 caratteristiche, che rappresentano il punto di vista dell'utente sul software. Rappresenta la capacità del software di supportare specifici utenti a raggiungere determinati obiettivi, con efficacia, produttività soddisfazione e sicurezza personale, in determinati contesti d'uso.



Caratteristiche della qualità in uso

- ◆ **Efficacia**, la capacità di supportare un utente nel raggiungere i suoi obiettivi con accuratezza e completezza in un dato contesto.
- ◆ **Produttività**, la capacità di supportare un utente nello spendere l'appropriata quantità di risorse in relazione all'efficacia dei risultati da raggiungere.
- ◆ **Soddisfazione**, la capacità di soddisfare un utente in un dato contesto d'uso.
- ◆ **Sicurezza**, la capacità di raggiungere accettabili livelli di rischio per le persone, l'ambiente di utilizzo, le attività dell'utilizzatore, in un dato contesto d'uso.

Riepilogo modello 9126



Caratteristiche e punti di vista

Rilevanza delle caratteristiche per i vari punti di vista

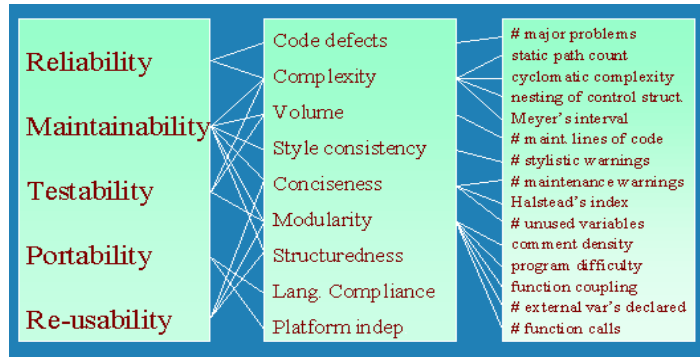
Vista Caratteristica	Utente	Sviluppatore	Gestore	Committente
Funzionalità	Alta	Bassa	Bassa	Media
Affidabilità	Media	Media	Alta	Media
Usabilità	Alta	Bassa	Media	Media
Efficienza	Media	Media	Alta	Media
Manutenibilità	Bassa	Alta	Media	Media
Portabilità	Bassa	Alta	Alta	Media

Interdipendenza tra le caratteristiche

- Nella 9126 non è possibile separare nettamente le varie caratteristiche, rispetto ai loro effetti sul software.
- Diverse caratteristiche e sottocaratteristiche interne ed esterne (ad es: la completezza, accuratezza, la *fault tolerance*) influenzano la qualità in uso.
- ◆ Un problema rilevato nell'uso (ad es. il non poter completare una operazione utente) può essere riferito a caratteristiche di qualità esterne (ad es. la affidabilità o l'operabilità) e/o a caratteristiche interne (un errore nelle strutture di decisione del codice).

Metriche ISO 9126

- La ISO/IEC 9126 fornisce 3 insiemi di metriche, per misurare rispettivamente le caratteristiche esterne (nella 9126-2), quelle interne (nella 9126-3), quelle in uso (nella 9126-4).



Descrizione delle metriche nella 9126

- Ogni metrica è individuata da questi elementi:

Nome della metrica	Scopo di utilizzo	Metodo con il quale va utilizzata	Formula ed elementi di calcolo	Interpretazione delle misure	Scala	Tipo di misura	Fonte degli input alla misura	Riferimento a processi ISO/IEC 12207	Beneficiari della misura
--------------------	-------------------	-----------------------------------	--------------------------------	------------------------------	-------	----------------	-------------------------------	--------------------------------------	--------------------------

Formula ed elementi di calcolo	$X = 1 - (A / B)$ A = n° di funzionalità mancanti scoperte nella valutazione B = n° di funzionalità descritte nei requisiti
Scopo di utilizzo	Misurare la completezza delle funzionalità offerte dal software
Metodo di misura	Utilizzare un test di tipo black box
Interpretazione misure	0 ≤ X ≤ 1 Il valore migliore è 1
Scala	Assoluta
Tipo di misure	A = numero (occorrenza) B = numero (occorrenza) X = numero (occorrenza) / numero (occorrenza)
Fonti	Requirement specification; Evaluation report

Esempio per una metrica di funzionalità, fonte adattamento ISO 9126

Utilizzo delle metriche esterne

- Scopo delle metriche per la misura della qualità esterna:
 - ◆ rappresentare la qualità di un prodotto software rispetto alle caratteristiche e sottocaratteristiche del modello 9126, durante le fasi di test,
 - ◆ validare l'aderenza del software rispetto ai requisiti di qualità esterna,
 - ◆ predire il livello di qualità "in uso" del prodotto,
 - ◆ descrivere il grado di rispondenza del prodotto rispetto ai requisiti espliciti ed impliciti dell'utenza.

Utilizzo delle metriche interne....

- Scopo delle metriche per la misura della qualità interna:
 - ◆ rappresentare la qualità di un prodotto software, nei suoi stati di lavorazione intermedi e finale "non eseguibili", rispetto alle caratteristiche e sottocaratteristiche del modello 9126,
 - ◆ predire il livello di qualità "esterno" del prodotto,
 - ◆ prevenire problemi nel prodotto in uso, scoprendo in anticipo i potenziali difetti.

....utilizzo delle metriche interne

- Attraverso le misure effettuate con metriche interne si dovrebbero prevenire i problemi nel prodotto eseguibile e nel comportamento del prodotto in uso, scoprendo le cause di possibili difetti il più presto possibile.
- ◆ Le metriche interne sono in genere combinazione di metriche elementari applicate al codice sorgente, a diagrammi UML o DFD, o grafi etc... (misurate con strumenti di “analisi statica” o con ispezioni e “code reading”).

Utilizzo delle metriche “in uso”

- Scopo delle metriche per la misura della qualità “in uso”:
 - ◆ verificare la capacità di un prodotto di soddisfare le esigenze dell’utente in un dato scenario d’uso, in relazione a specifici obiettivi.
 - ◆ Le metriche per la qualità in uso sono in genere combinazione di metriche elementari applicate alla interazione tra utente e sistema software, misurate con field test, ispezioni, walkthrough etc..

Uso integrato delle metriche....

- Esistono diverse relazioni ed interdipendenze tra le tre classi di metriche.
- Molte delle caratteristiche 9126 possono essere misurate contemporaneamente da metriche interne ed esterne.
 - ◆ Ad es. la "affidabilità" può essere misurata *esternamente* rilevando il numero di errori durante l'esecuzione del prodotto in un dato periodo di osservazione ed *internamente* ispezionando il codice sorgente per verificare il livello di tolleranza agli errori che possiede.

...uso integrato delle metriche

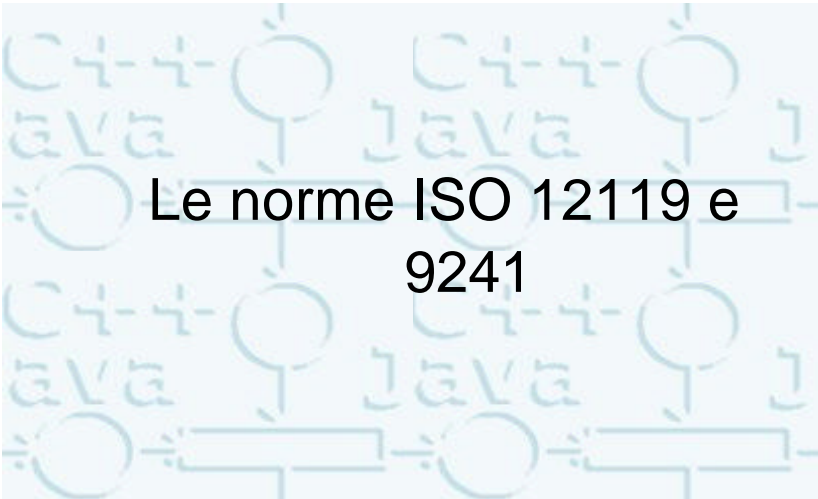
- ◆ Un problema rilevato nell'uso di un prodotto (ad es. l'impossibilità per un utente di completare correttamente una operazione) può essere riferito sia a caratteristiche di qualità esterne (ad es. la affidabilità o l'operabilità, sottocaratteristica della usabilità) che a caratteristiche interne (un errore nelle strutture di decisione del codice).
- La qualità va quindi misurata come combinazione delle misure riferite alle tre qualità, in modo da coprire i diversi punti di vista sulla qualità.

Le metriche nel ciclo di vita del software

- Utilizzo metriche nel ciclo di vita:
 - ◆ Un committente può valutare la convenienza a scegliere un determinato prodotto usando metriche per la valutazione della affidabilità.
 - ◆ Uno sviluppatore dovrà disporre di metriche di funzionalità per verificare la corretta implementazione nei vari semilavorati prodotti.
 - ◆ Un manutentore può valutare lo sforzo ed il rischio di modificare un programma utilizzando metriche di manutenibilità.
 - ◆ Chi dovrà valutare la capacità di migrazione di un prodotto in altri ambienti dovrà disporre di misure di portabilità.
 - ◆ Gli utenti dovranno poter misurare la usabilità e la efficacia del prodotto che è stato loro fornito.

Applicabilità della ISO 9126

- Il modello 9126 può essere utilizzato per definire i requisiti di ogni tipo di software, Custom, COTS, incluso il "firmware".
- La valutazione della qualità secondo la 9126 è possibile in ogni momento del ciclo di vita, dalla acquisizione, allo sviluppo (progettazione, codifica), alla manutenzione.
- Destinatari sono quindi tutti gli attori del ciclo di vita del software, utenti, sviluppatori, gestori ed addetti alla manutenzione, managers, auditors, ognuno secondo le proprie esigenze nel SLC.



Le norme ISO 12119 e 9241

I prodotti COTS

- Una quota importante di mercato è coperta dai prodotti COTS, *Commercial Off The Shelf*.
 - ◆ Questi prodotti sono sviluppati non su requisiti specifici di un dato utente, ma pensando ad un uso diffuso da parte di un'utenza non predefinita al 100%.
 - ◆ A supporto della valutazione della qualità di questi prodotti la ISO ha emesso la norma **12119**, *Software product evaluation, requirements for COTS products and instruction for testing* (1994), in fase di revisione e ripubblicazione nel 2006.

I prodotti COTS - Caratteristiche

- Solitamente, i prodotti COTS sono venduti inseriti in una confezione (package) che tipicamente evidenzia all'acquirente le caratteristiche del prodotto.
 - ◆ Spesso, questa è la principale fonte con la quale il produttore (o chi commercializza il prodotto) comunica con l'acquirente/utente per informarlo circa le caratteristiche di ciò che sta per acquisire e utilizzare.
 - ◆ E' perciò importante che queste informazioni mettano in grado chi le legge di capire se il prodotto soddisfa le proprie esigenze.

I prodotti COTS - Caratteristiche

- La qualità dei prodotti COTS è molto importante per questi motivi:
 - ◆ non sono tipicamente sviluppati seguendo il ciclo iterativo 9126 che vede la qualità finale determinarsi progressivamente dall'interazione utente-sviluppatori;
 - ◆ sono sviluppati di solito in tempi brevi (time to market);
 - ◆ hanno una vita utile sul mercato relativamente breve, e quindi non sufficiente a permettere di scoprire con l'uso molti errori e poi correggerli, o di apportarvi significativi miglioramenti traendo indicazioni dal loro utilizzo.
- ◆ Per ovviare a questi problemi, si ricorre spesso a certificazioni di prodotto che attestano la qualità dei COTS, rispetto a diverse esigenze degli utenti.

Contenuti della ISO 12119

- La norma ISO 12119 definisce:
 - ◆ i requisiti di qualità per i prodotti COTS,
 - ◆ i requisiti per la documentazione dei test effettuati sui prodotti COTS prima del rilascio sul mercato, inclusi i requisiti per effettuare i test, per la definizione dei casi di test e per la documentazione dei risultati dei test,
 - ◆ indicazioni su come valutare la conformità dei prodotti COTS rispetto a dei requisiti di qualità.
- Riguarda quindi solo la valutazione che un acquirente può dare di un prodotto COTS esaminandolo dopo l'immissione sul mercato.
- Non riguarda direttamente aspetti del processo produttivo.

Finalità della ISO 12119

- La norma ISO 12119 può essere usata:
 - ◆ Dai fornitori di prodotti COTS per:
 - ✓ specificare i requisiti di un prodotto,
 - ✓ pubblicizzare le caratteristiche di un prodotto,
 - ✓ valutare le caratteristiche di un prodotto,
 - ✓ emettere dichiarazioni di conformità del prodotto,
 - ✓ applicare marchi di conformità del prodotto.
 - ◆ Da laboratori per:
 - ✓ effettuare test finalizzati al rilascio di marchi e certificati
 - ◆ Dai certificatori di prodotti COTS per:
 - ✓ definire uno schema di certificazione di terza parte
 - ◆ Dagli acquirenti di prodotti COTS per:
 - ✓ verificare se un prodotto può rispondere a delle esigenze

I prodotti conformi alla ISO 12119

- Un prodotto COTS è conforme alla norma 12119 se:
 - ◆ rispetta i requisiti di qualità definiti nella norma,
 - ◆ la descrizione sul package è conforme a quanto richiesto dalla norma,
 - ◆ la documentazione allegata (di installazione, uso, manutenzione etc..) è conforme alla norma,
 - ◆ prima del rilascio sul mercato è stato “testato” secondo quanto definito nella norma, producendo e rendendo disponibile la documentazione di test definita nella norma,
 - ◆ le anomalie rilevate durante i test sono state risolte prima del rilascio.

Requisiti del package (1/2)

- La 12119 definisce questi elementi come rilevanti per la qualità dei prodotti COTS:
 - 1) requisiti della descrizione del prodotto sul package:
 - ? tra questi requisiti la norma include l'identificazione univoca del prodotto, del fornitore, della versione (o della data di rilascio), la descrizione del possibile utilizzo del prodotto, le caratteristiche del sistema hardware e software richiesto per l'uso, indicazioni sulle modalità di ottenere supporto e manutenzione sul prodotto;

Requisiti del package (2/3)

- ◆ Il package deve contenere affermazioni sulle caratteristiche del prodotto, nello schema 9126:
 - ? funzionalità offerte dal prodotto, sicurezza ed interoperabilità; se vi sono solo alcuni valori di input ammissibili per l'utente, devono essere chiaramente identificati; vanno descritte le condizioni che limitano le funzionalità e il modo di limitare utilizzi indesiderati;
 - ? affidabilità del prodotto, in particolare le procedure per il recupero dei dati dopo le malfunzioni e sulla capacità di gestire le malfunzioni (fault tolerance);; accorgimenti che si possono mettere in atto per limitare i problemi;

Requisiti del package (3/3)

- ? Usabilità: i tipi di interfacce offerte all'utente (menu, *web browser*, funzioni di help; le conoscenze base richieste all'utente per l'uso del prodotto; se effettuabili, devono essere descritte le modalità di personalizzazione e parametrizzazione del prodotto e allegati gli strumenti necessari a farle;
- ? Efficienza: configurazione necessaria per l'ambiente operativo di utilizzo; consumo di risorse;
- ? Manutenibilità: le modalità per modificare il software, a fronte di problemi o nuove esigenze;
- ? Portabilità; il sistema hardware/software necessario e sufficiente per farlo funzionare.

Requisiti per la documentazione

2) requisiti per la documentazione allegata al prodotto:

- ? **Completezza:** deve includere la descrizione della procedura di installazione, se è previsto che l'utente possa effettuarla; se la documentazione è in più documenti, va fornita una guida ed un indice;
- ? **Correttezza:** non devono esserci ambiguità od errori;
- ? **Consistenza:** non vi devono essere contraddizioni nella documentazione; ogni termine usato deve avere lo stesso significato in ogni documento allegato;
- ? **Comprensibilità:** deve essere facilmente comprensibile;
- ? **Apprendibilità:** deve facilitare l'apprendimento dell'uso del software.

Requisiti di qualità del prodotto (1/3)

3) requisiti di qualità del prodotto:

- ? **Funzionalità:** tutte le funzionalità descritte nel package devono essere richiamabili; il software deve funzionare come descritto nella documentazione allegata e le condizioni di attivazione delle funzioni devono corrispondere a quanto descritto nel manuale d'uso;
- ? **Affidabilità:** il prodotto non deve consentire l'immissione di inputs dannosi o incorretti; non deve danneggiare o perdere dati;
- ? **Efficienza:** l'utente deve essere avvisato di possibili tempi di risposta lunghi per operazioni che esegue il prodotto a fronte di sue richieste;

Requisiti di qualità del prodotto (2/3)

- ? *Usabilità*; domande, messaggi del prodotto devono essere facilmente comprensibili dall'utente; messaggi di errore devono aiutare a correggere il problema, con l'ausilio della opportuna documentazione; la tipologia dei messaggi (errori, *warnings*, domande, risposte etc..) deve essere chiaramente individuabile; i formati dei campi di input e dei reports devono essere facilmente comprensibili dall'utente; l'esecuzione di una operazione deve essere annullabile (deve essere reversibile la modifica di dati); un utente deve poter apprendere come chiedere servizi al prodotto leggendo la documentazione e le funzioni di help;

Requisiti di qualità del prodotto (3/3)

- ? *Manutenibilità*; ciò che deve essere fatto per mantenere il prodotto deve essere conforme alle descrizioni sul package e nella documentazione allegata;
- ? *Portabilità*; l'utente deve poter effettuare l'installazione seguendo le istruzioni allagate; gli ambienti hardware e software descritti nella documentazione devono essere sufficienti a completare l'installazione; il prodotto deve poter essere rimovibile dal computer in modo automatico.

Requisiti per la documentazione dei test

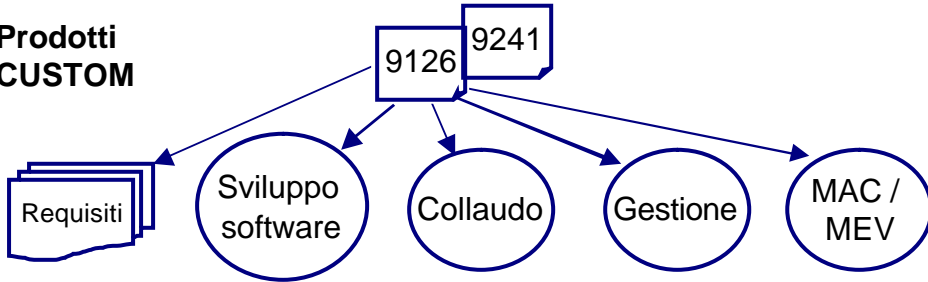
- 4) requisiti per la documentazione dei test pre-rilascio:
- ? Scopo del test: deve essere descritta la finalità dei vari test svolti;
 - ? Consistenza: la documentazione non deve contenere ambiguità o differenze semantiche nelle sue varie parti;
 - ? Completezza: la documentazione deve contenere piano dei test, caratteristiche dell'ambiente di test, casi di test, tracciabilità tra test, risultati dei test;
 - ? Identificabilità: ogni documento deve essere univocamente identificato;
 - ? Tutte le funzioni e le caratteristiche del prodotto (come descritte sul package e nei manuali allegati) devono essere state testate, incluse le procedure di installazione.

Caratteristiche di un software usabile

- La **ISO 9241-10** *Ergonomic requirements for office work with visual display terminals (VDTs): Dialogues principles* (1996), identifica queste caratteristiche di qualità per un software usabile:
 - 1) idoneità al compito,
 - 2) autodescrittività,
 - 3) controllabilità,
 - 4) conformità alle aspettative,
 - 5) tolleranza agli errori,
 - 6) idoneità alla personalizzazione,
 - 7) idoneità all'apprendimento.

Sinottico uso norme

**Prodotti
CUSTOM**



**Prodotti
COTS**

