

Standard ISO 9126

1 Qualità

Lo standard **ISO 9126** definisce la **qualità** del software in termini di 6 caratteristiche,

- funzionalità
- affidabilità
- usabilità
- efficienza
- manutenibilità
- portabilità

ciascuna delle quali è a sua volta definita in termini di varie sottocaratteristiche.

2 Funzionalità

Definizione: L'insieme di attributi che riguardano l'esistenza di un complesso di funzioni e delle specifiche proprietà. Le funzioni sono quelle che soddisfano i requisiti stabiliti o impliciti.

Questa è la caratteristica primaria del software: riguarda *cosa* deve fare l'applicazione (*requisiti funzionali*), mentre le altre caratteristiche riguardano *come* deve farlo (*requisiti non funzionali*).

Infatti, nella prima versione dello standard, la funzionalità era invece chiamata *correttezza*.

3 Affidabilità

Definizione: L'insieme di attributi che riguardano la capacità del prodotto software di mantenere il livello di prestazioni con condizioni e limiti di tempo fissati.

Questa caratteristica riguarda la frequenza dei malfunzionamenti: nella definizione, la parola "prestazioni" si riferisce al funzionamento corretto (o meno) del software, e non alla sua velocità.

Per valutare l'affidabilità, bisogna specificare la probabilità che si verifichi un errore

- in un dato periodo di tempo (in un periodo più lungo, è più facile che si abbiano malfunzionamenti), e
- usando il software in un certo modo (se si utilizzano più funzionalità, oppure si fa un uso più intenso del software, c'è una maggiore probabilità di errore).

Nel caso delle applicazioni critiche, questo studio viene effettuato mediante appositi modelli matematici.

Osservazione: L'affidabilità del software ha natura diversa rispetto a quella dei sistemi fisici, perché il software non si logora.

4 Usabilità

Definizione: L'insieme di attributi che riguardano lo sforzo necessario all'uso del prodotto e alla valutazione individuale di tale uso relativamente a un insieme di utenti prefissato o implicito.

L'usabilità:

- dipende dall'utente per cui il software è stato previsto;
- riguarda non solo gli utenti finali, ma anche installatori, operatori, ecc.;
- comprende anche lo *sforzo di valutazione*, cioè l'utente deve poter capire se l'applicazione è quella adatta alle proprie esigenze.

5 Efficienza

Definizione: L'insieme di attributi che riguardano la relazione esistente tra il livello delle prestazioni del prodotto software e la quantità di risorse necessarie nell'ambito di condizioni fissate.

Questa caratteristica si riferisce all'utilizzo di risorse, tra cui:

- tempo;

- spazio di memoria;
- occupazione della banda (per i mezzi di comunicazione).

6 Manutenibilità

Definizione: L'insieme degli attributi che riguardano lo sforzo necessario a eseguire modifiche.

La manutenzione del software si suddivide in tre tipi:

- correttiva:** rimozione dei difetti (bug);
- perfettiva:** miglioramento del software mediante l'aggiunta di funzionalità;
- adattativa:** adeguamento del software al cambiamento delle specifiche e/o dell'ambiente circostante.

Le sottocaratteristiche della manutenibilità sono:

1. *analizzabilità:* deve essere possibile identificare le porzioni di software da modificare;
2. *modificabilità:* bisogna poter effettuare facilmente le modifiche;
3. *stabilità:* quando si effettua una modifica a una parte del programma, non devono esserci effetti inattesi che richiedono modifiche anche a altre parti;
4. *testabilità:* il risultato delle modifiche deve poter essere convalidato.

7 Portabilità

Definizione: L'insieme degli attributi che riguardano la capacità del prodotto software di essere trasferito da un ambiente all'altro.

Le differenze tra ambienti possono essere:

- *organizzative* (ad esempio, organizzazioni diverse potrebbero avere flussi di informazioni diversi);
- *software* (ad esempio, sistemi operativi diversi, o anche solo versioni diverse di uno stesso sistema operativo);
- *hardware*.

8 Uso del modello

1. Per ogni sottocaratteristica, si valuta un insieme di misure.
2. Si calcola il punteggio di ciascuna sottocaratteristica, come somma pesata delle misure (stabilendo i pesi in base alle esigenze dell'ambiente d'uso dell'applicazione).
3. Si calcola il punteggio di ciascuna caratteristica, corrispondente a una somma pesata dei punteggi delle sue sottocaratteristiche.
4. Da una somma pesata dei punteggi delle varie caratteristiche si ricava il punteggio complessivo di qualità del software.

In questo modo, è possibile confrontare prodotti diversi e/o stabilire degli obiettivi (valori di soglia) che una determinata applicazione deve raggiungere per essere adeguata a particolari esigenze.