

Qualità del Software

La qualità è relativa

Prospettive sulla qualità	Osservatore
<i>Trascendentale</i> : la qualità è qualcosa che possiamo riconoscere ma non definire (ideale verso cui tendere)	filosofo
<i>Utente (qualità esterna)</i> : la qualità è appropriatezza rispetto allo scopo	clienti, operatori commerciali
<i>Produzione</i> : la qualità è conformità alle specifiche (del processo costruttivo)	sviluppatori
<i>Prodotto (qualità interna)</i> : la qualità deriva dalle caratteristiche intrinseche del prodotto (metriche del sw)	ricercatori
<i>Valore</i> : la qualità dipende da quanto il cliente è disposto a pagarla	imprenditore

Qualità del sw: una duplice classificazione

Il nocciolo dell'ingegneria del software è l'attenzione alla qualità (Pressman, 2004)

Primo criterio:

- Qualità di prodotto
- Qualità di processo (influenza quella di prodotto)

Secondo criterio:

Qualità esterne: percepibili da un osservatore esterno che esamina una black-box (sono quelle di interesse per SE)

Qualità interne: percepibili esaminando la struttura interna di una white-box (sono quelle che permettono di realizzare le qualità esterne)

Fattori di qualità

Di prodotto

- Correttezza
- Affidabilità
- Robustezza
- Sicurezza
- Innocuità
- Prestazioni
- Usabilità
- Portabilità
- Interoperabilità

Sia di prodotto sia di processo

- Verificabilità
- Manutenibilità
- Riutilizzabilità
- Comprensibilità

Di processo

- Produttività
- Visibilità
- Tempestività

Correttezza

- Il sw è corretto se soddisfa le specifiche dei requisiti funzionali (caratteristica oggettiva)
- Se tali specifiche sono formali, la correttezza può essere provata formalmente (mediante una dimostrazione di teorema) dal momento che i programmi sono oggetti formali, oppure smentita mediante controesempi (attività di testing)

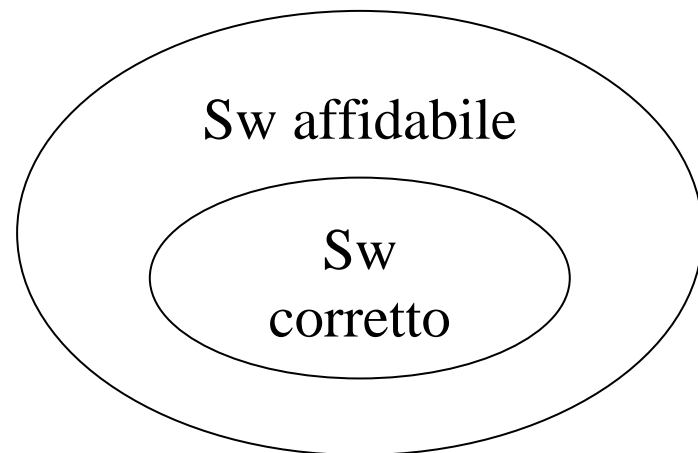
Limiti:

- È una qualità assoluta (sì/no), non esiste alcun concetto di “grado di correttezza” né di gravità dell’infrazione
- E se le specifiche fossero sbagliate (magari a causa di requisiti inattendibili o di errori nella conoscenza di dominio)?

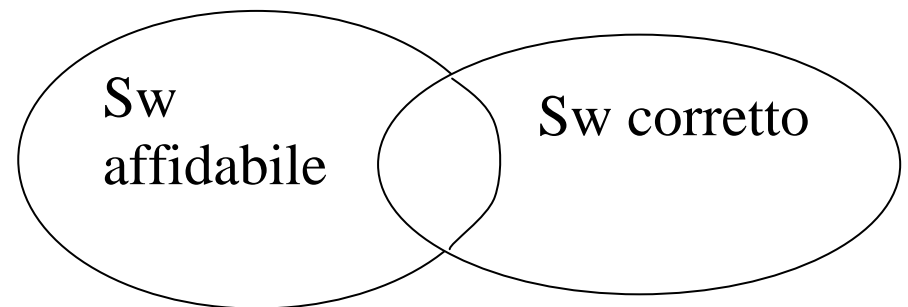
Affidabilità

- Informalmente significa che l'utente si può fidare del prodotto sw in questione
- I risultati delle elaborazioni sono quelli voluti o presentano disturbi tollerabili (concetto soggettivo e variabile da sistema a sistema)
- Matematicamente è definibile come la “probabilità di assenza di malfunzionamenti nell'unità di tempo”
- Se le specifiche sono corrette (situazione ideale), un sistema sw corretto è anche affidabile ma non viceversa

Situazione ideale



Situazione reale



Robustezza

Il prodotto sw si comporta in modo ragionevole anche in circostanze non previste dai requisiti (valori d'ingresso non validi, malfunzionamenti hw, ecc.)

Sicurezza (*security*)

Capacità del sistema sw in esecuzione di impedire l'accesso a info private

Innocuità (*safety*)

- Assenza di pericolosità (o tollerabilità della pericolosità) dell'elaborazione
- Capacità di operare senza malfunzionamenti catastrofici

Prestazioni

- Tempi di elaborazione, occupazione di memoria, traffico in rete
- Dipendono da un uso efficiente delle risorse (memoria, tempo di elaborazione, canali di comunicazione)
- Influenzano l'usabilità e la "scalabilità" di un'applicazione (una soluzione che funziona su una LAN magari non funziona su una WAN)
- Possono essere verificate mediante
 - ✓ Analisi di complessità (insegnamento di "Algoritmi e Strutture Dati")
 - ✓ Misure durante esecuzioni-campione
 - ✓ Simulazione di un modello
 - ✓ Analisi di un modello (es. analisi probabilistica secondo la teoria delle code)

Usabilità

- Intuitività, naturalezza ed ergonomia dell'interazione col sistema sw dal punto di vista delle diverse categorie di utenti dello stesso (espressione vecchia: user-friendliness)
- È influenzata dalla tipologia di interfaccia utente (ad es. testuale o grafica)
- Dipende da coerenza, prevedibilità, gradevolezza ed esplicatività dell'interfaccia
- L'utente non deve provare una sensazione di smarrimento in alcuna occasione

Portabilità

- Un prodotto sw è portabile se può essere eseguito su più piattaforme hw e/o sw
- È una proprietà importante anche quando vengono introdotte nuove piattaforme e nuovi ambienti o quando la rete aziendale è eterogenea

Interoperabilità

Capacità di un sistema sw di coesistere e cooperare con altri sistemi o con servizi web (attraverso middleware)

Verificabilità

- Facilità di verifica di altre proprietà del prodotto (ad es. correttezza o prestazioni)
- È prevalentemente una qualità interna ma può essere anche esterna (ad es. verificabilità della “sicurezza”)
- Lo stato di avanzamento di un progetto deve essere controllabile, unitamente al grado di soddisfacimento dei vincoli

Manutenibilità

Facilità e rapidità con cui è eseguibile la manutenzione

Manutenzione = attività post-rilascio (50-70% dei costi complessivi); si articola in:

- Correttiva ($\cong 20\%$ dei costi di manutenzione): correzione dei difetti residui
- Adattativa (20-25%): adattamento a cambiamenti nell'ambiente
- Perfettiva (50%, è conseguenza della malleabilità del sw): aggiunta/rimozione di funzionalità, miglioramento di alcune caratteristiche di qualità → differenza del concetto di manutenzione rispetto all'ingegneria tradizionale

Situazione di retroazione: l'organizzazione del lavoro fa nascere esigenze di automazione mediante sw → introduzione del sw e conseguente modifica dell'organizzazione del lavoro → necessità di modifiche del sw

Manutenzione del processo produttivo: modifiche ai piani inizialmente previsti

Riusabilità

- Facilità e rapidità con cui prodotti/componenti/processi esistenti possono essere riusati, dopo lievi modifiche, per costruire altri prodotti
- Il riuso di parti standardizzate è una misura della maturità di un settore ingegneristico

Comprensibilità

- Facilità di comprensione di un sistema sw (per modificare un sistema sw è innanzi tutto necessaria la comprensione dello stesso)
ATTENZIONE: non si intende facilità di comprensione del solo codice ma anche di altri artefatti e delle loro relazioni
- Facilità di comprensione di un processo

Produttività

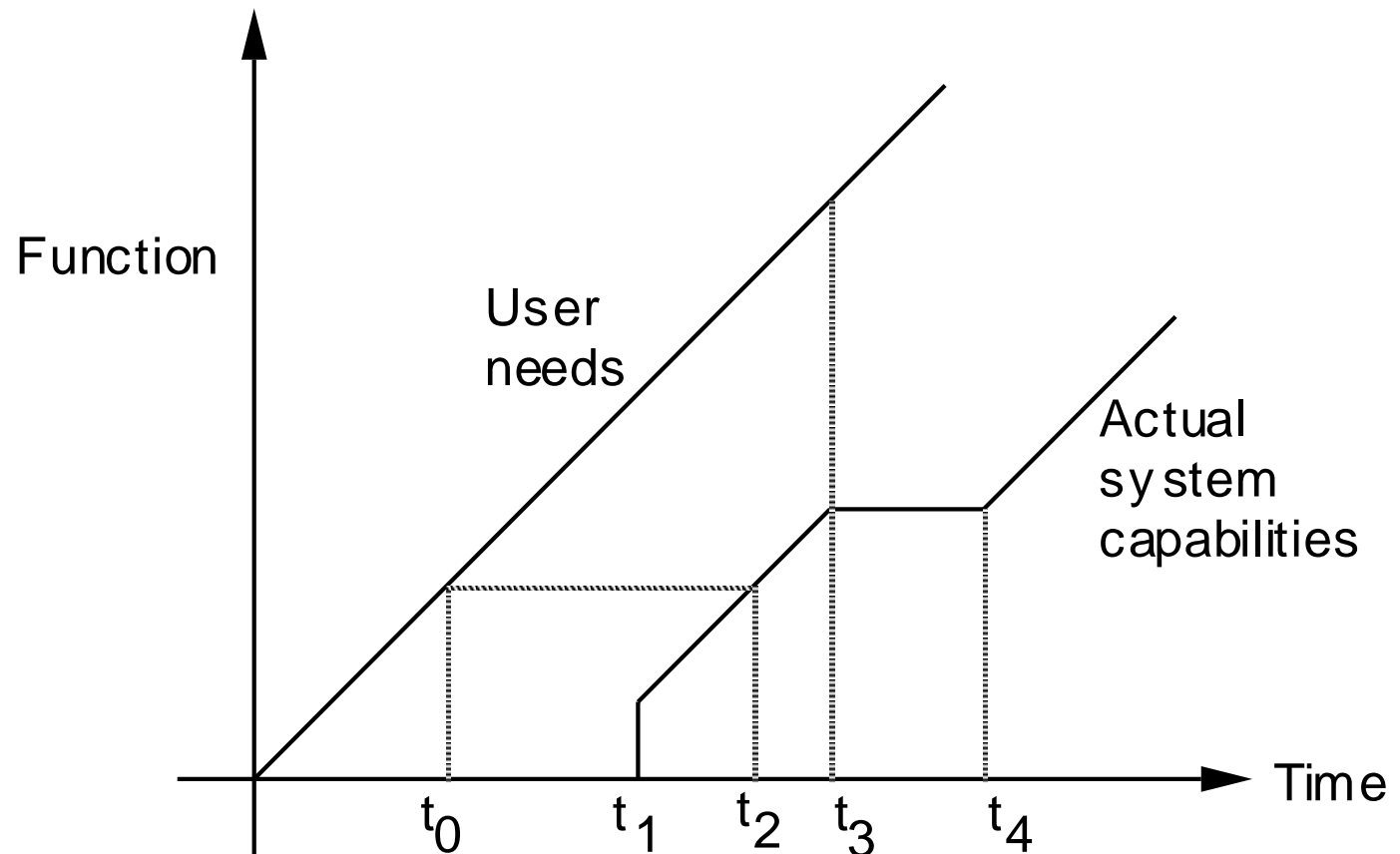
- Capacità di un processo di incrementare il gettito produttivo (ad es. mettendo a disposizione CASE tool)
- È difficile da caratterizzare

Visibilità

- Ogni passo del processo è chiaramente documentato
- In ogni istante è noto lo stato corrente

Tempestività

- Capacità di consegnare un prodotto sw con puntualità (o di presentarlo al momento giusto sul mercato)
- Spesso il processo produttivo non segue l'evoluzione dei requisiti degli utenti, sussiste bensì uno scostamento (mismatch) fra i requisiti degli utenti e lo stato del prodotto



Aree applicative del sw

Classificazione per insiemi omogenei di caratteristiche di qualità da quantificare all'avvio di un progetto

- Sistemi informativi
- Sistemi in tempo reale
- Sistemi distribuiti
- Sistemi embedded

Molti sistemi hanno caratteristiche comuni a più aree (es. sistema informativo distribuito e in tempo reale, sistema embedded in tempo reale)

Sistemi informativi

- Si interfacciano a una base di dati
- Sono orientati ai dati
- Molti forniscono una GUI web
- Molti consentono personalizzazioni (ad es. definizione e generazione di nuovi rapporti)

Es.: sistemi bancari, bibliotecari, di gestione del personale

Requisiti:

- Integrità dei dati
- Security
- Disponibilità dei dati
- Prestazioni relative alle transazioni

Sistemi in tempo reale

- Devono rispondere a determinati eventi in ingresso entro un intervallo di tempo predefinito e spesso molto limitato (quindi sussistono vincoli quantitativi relativi ai tempi di risposta)
- Spesso fanno parte di sistemi complessi (di automazione di fabbrica, sorveglianza, ecc.)
- Sono orientati al controllo

Es.: sistemi di monitoraggio di pazienti/impianti, sistemi di controllo del volo di un aereo, sistemi di difesa, sistema di gestione del mouse

N.B. Definizione sbagliata di sistema in tempo reale: sistema che richiede tempi di risposta veloci. Infatti una risposta troppo veloce può essere scorretta quanto una risposta tardiva

Sistemi distribuiti

Distribuzione su computer diversi, collegati da una rete di TLC, di

- dati e/o
- componenti sw eseguibili

Requisiti da definire:

- Livello di distribuzione
- Possibilità di tollerare il partizionamento (della rete in sottoreti disgiunte)
- Tolleranza per l'indisponibilità di uno o più computer

Sistemi distribuiti: nuove strade per ottenere la qualità

L'eventuale replica dei dati su più macchine aumenta:

- affidabilità
- prestazioni

Java definisce un linguaggio intermedio (*bytecode*) che può essere interpretato su ogni computer → i componenti possono essere caricati in rete in maniera dinamica quando ciò risulta necessario (mobilità del codice)

L'eventuale trasferimento dinamico del codice al nodo che memorizza i dati sui quali il codice deve operare (es. applet Java) migliora le prestazioni

Sistemi embedded

- Il sw è solo uno dei componenti di questi sistemi, quello che controlla gli altri, interfacciandosi con essi
- I requisiti del sw devono essere bilanciati con quelli delle altre parti
- Spesso privi di interfacce rivolte all'utente finale

Es. sw di controllo di aerei, robot, elettrodomestici, cruscotto dell'automobile, telefoni cellulari, macchine distributrici, sistemi di commutazione telefonica