

Collaudo e qualità del software

Introduzione al testing ...

Relatore

Ercole Colonese

Roma, 29 novembre 2010



Introduzione al test e collaudo del software



Temi trattati nel libro

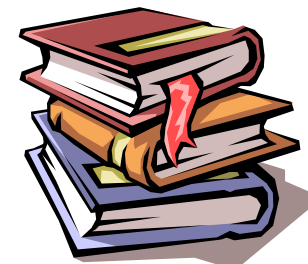
- Il software
 - Che cos'è il software
 - Lo sviluppo del software
 - Quantità di errori immessi e rimossi nel software
 - Propagazione degli errori nel software
 - Costo e rimozione degli errori nel software
- Che cos'è il collaudo del software
 - Psicologia del tester
 - Problematiche relative al test del software
 - Relazione tra test e altre attività
 - Principi generali del testing
- Testing e qualità del software
 - Che cos'è la qualità del software
 - Qualità del prodotto e del processo
 - Assicurazione qualità
 - Standard di verifica e validazione
 - L'approccio alla qualità del software
 - La qualità del prodotto e il ciclo di vita del software



Perché un libro sul test del software ...

Nonostante si sviluppi software ormai da molti decenni ...

- le **best practice** dello sviluppo del software rimangono ancora sconosciute e poco applicate (una di queste è appunto il testing)
- il test è spesso **sottostimato** e relegato in fondo al ciclo di vita del software
- la fase di test subisce le conseguenze negative dei ritardi e dell'aumento dei costi di progetto e viene spesso “**fermato**” prima del tempo o “**ridotto**”
- il test non è quasi mai progettato in base ai **requisiti** funzionali e non funzionali
- il test non è pianificato e progettato come risposta alla strategia di sviluppo (e quindi come **strategia di test**)
- il test viene eseguito quasi esclusivamente “**a mano**” e gli **strumenti** di test sono poco utilizzati
- l'**organizzazione di test** è poco curata e le **competenze** non ricevono la giusta attenzione e sviluppo professionale
- i **metodi** e le **tecniche** di test sono poco conosciute e applicate



Con il nuovo libro sul test del software ...

Si è evoluto fornire un contributo che ...

- presenti un modello concettuale completo
- fornisca esempi pratici tratti dall'esperienza
- rappresenti uno strumento di lavoro (da tenere sulla scrivania) piuttosto che nella libreria
- contribuire a diffondere la cultura della qualità nel mondo dello sviluppo software

ISBN 978-88-6178-517-5

Editrice UNI Service

disponibile (anche in formato PDF) all'indirizzo:
<http://www.uni-service.it/collaudo-e-qualita-del-software.html>

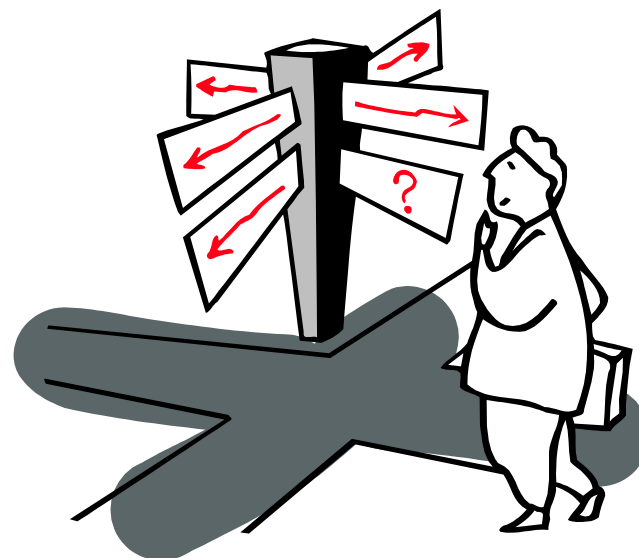


Quali obiettivi ha il testing?

Il test ha due obiettivi principali:

1. Verificare che il software prodotto indirizzi i requisiti stabiliti
2. Scoprire il maggior numero possibile di errori (e permetterne la loro correzione)

Il test rappresenta quindi lo strumento più efficace per il
“**controllo qualità del software**”



Gli errori nel software

Lo **sviluppo del software** è un'attività con una fortissima componente umana e (quindi) fortemente **soggetta ad errori** !

La **scoperta degli errori** viene demandata all'attività di testing

Ma **quali** e **quanti** test si dovrebbero eseguire?

La teoria della **propagazione degli errori** ci aiuta a capire **quali** test eseguire e **quando** eseguirli


Ancora più interessante sarebbe conoscere **quanti** errori siano presenti nel software, di quale **tipo** siano e **dove** si annidino

Il primo bug documentato

Photo # NH 96566-KN First Computer "Bug", 1945

92

9/9

0800 Antan started
 1000 " stopped - antan ✓
 1300 (032) MP - MC ~~1.98264000~~ 1.2700 9.037 847 025
 (033) PRO 2 2.130476415 ~~2.130476415~~ 9.037 846 995 connect
 connect 2.130676415 4.615925059(-2)
 Relays 6-2 in 033 failed special speed test
 in Relay .. 10.00 test.
 Relays changed
 1100 Started Cosine Tape (Sine check)
 1525 Started Multi-Adder Test.
 1545  Relay #70 Panel F
 (moth) in relay.
 First actual case of bug being found.
~~1630~~ 1630 Antan started.
 1700 closed down.

MARK II Project
 G.M. Hopper, 1945

Rimozione degli errori nel software

La **distribuzione** degli errori nelle diverse fasi del ciclo di vita del software segue un andamento “**a campana**” che risulta essere perfettamente simile per ogni

- tipo di software
- tipo di processo
- tipo di organizzazione o cultura

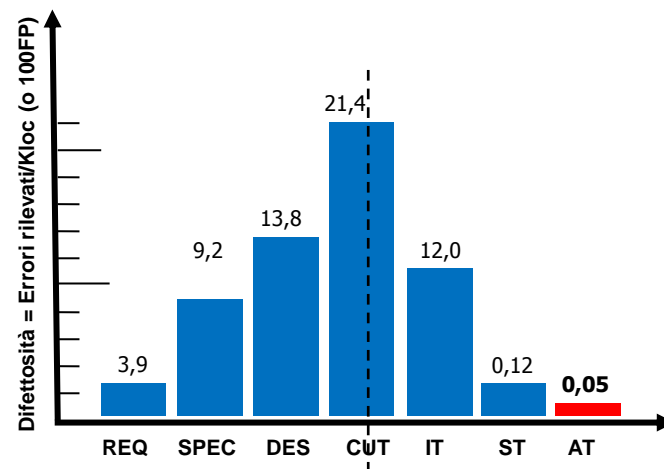
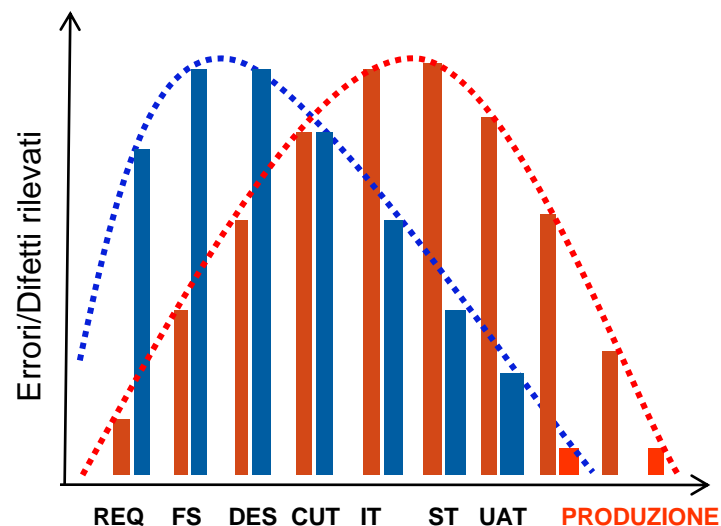


Figura 1. Curva di rimozione degli errori

Curve di rimozione degli errori

A variare è invece la “**simmetria**” della curva di rimozione degli errori in base al processo di testing ...

Anticipare la rimozione degli errori nelle **fasi alte** del ciclo di vita del software produce un enorme **beneficio** calcolato e dimostrato nella pratica sperimentale di molti progetti



Propagazione degli errori nel software

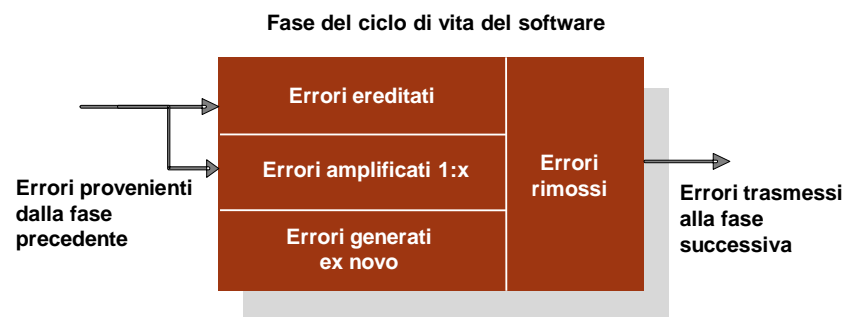


Figura 2. Schema di propagazione degli errori (IBM System Journal, 1981)

Gli errori **ereditati** dalla fase precedente si amplificano con un **fattore moltiplicativo 1:x** che aumenta in ogni fase

Essi si **sommano** a quelli immessi nella fase attuale

Gli errori **rimossi** riducono drasticamente tale numero ma ...

Gli errori **residui** saranno inevitabilmente trasmessi alla fase successiva con l'effetto moltiplicativo appena mostrato

Tipologia di errori immessi nel software

Tipologia di errore	Percentuale
Specifiche errate o incomplete (IES)	22%
Requisiti errati o incompleti (IER)	17%
Errore nella rappresentazione dei dati (EDR)	14%
Test incompleto o errato (IET)	10%
Inconsistenza nell'interfaccia (ICI)	6%
Errore di traduzione del disegno nel codice (PLT)	6%
Errore logico di progettazione (EDL)	5%
Deviazione intenzionale dalle specifiche (IDS)	5%
Documentazione incompleta o errata (IID)	4%
Violazione degli standard di programmazione (VPS)	3%
Interfaccia utente ambigua o inconsistente (HCI)	3%
Altro (MIS)	5%

Fonte: Roger S. Pressman (2007)

Costo di rimozione degli errori nel software

Il costo di rimozione di un errore espresso in “numero di ore” aumenta al passare delle fasi

Anticipare la rimozione degli errori in **ciascuna fase** del ciclo di vita del software produce l'enorme **beneficio** di cui si è parlato prima

La rimozione nelle fasi alte viene effettuata tramite le **revisioni tecniche**, **ispezioni** o **walkthrough**

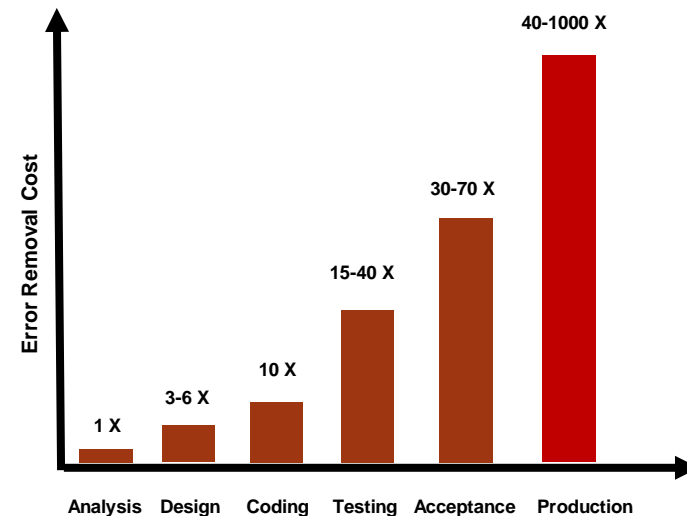


Figura 3. Costo di rimozione degli errori

Fonte: Roger S. Pressman (2007)

Problematiche relative al test del software (1/3)

❑ **Psicologia del tester**

Gli obiettivi del testing condizionano il “tester”

Trovare errori è diverso che verificare che il software funzioni correttamente: sono due atteggiamenti psicologici diversi

“Trovare errori” richiede atteggiamento da “scopritore” (“distruttore”)

“Dimostrare che il software funziona correttamente” presuppone un atteggiamento “costruttivo”

❑ **Selezione dei test**

Quali e quanti test eseguire?

Come scegliere i casi di prova all’interno di un insieme “teoricamente infinito” (*test suite*)?

I limiti di tempo e di costo condizionano la scelta!

La criticità del software e delle diverse funzioni fa da “driver” nella scelta (*Risk Management*)

Problematiche relative al test del software (2/3)

❑ **Efficacia ed obiettivi del testing**

L'esecuzione dei test permette di osservare il comportamento del software in diverse condizioni (utente, ambiente di test, dati di input, sequenza delle operazioni, ecc.)

La scelta delle condizioni determina i risultati e quindi gli obiettivi del test

❑ **Test per scoprire i difetti**

Un test per scoprire errori deve indurre il software a sbagliare

Un test che scopra difetti è quindi diverso da uno che verifichi il corretto comportamento del software

Non solo la psicologia del tester è differente ma anche la progettazione del caso di prova sarà molto diversa

❑ **Il problema dell'oracolo**

L'oracolo determina la correttezza o meno del comportamento del software e dichiara se il test sia OK o No (*Pass or Fail*)

Non sempre è facile prevedere e/o determinare l'esito

Problematiche relative al test del software (3/3)

Limiti teorici e pratici del testing

La teoria mette in guardia circa la fiducia verso i risultati dei test
“Un test può dimostrare la presenza di errori ma non la loro assenza”
(*Dijkstra*)

Il testing deve perciò essere guidato dalla criticità (*Risk Management*)

Il problema dei cammini non percorribili

Sono flussi interni non percorribili da alcun caso di prova e da nessuna combinazione di dati di input al programma

Testabilità del software

Un software è detto testabile se:

- a) possiamo immaginare almeno un caso di prova in grado di esercitarlo
- b) esiste la probabilità che possa fallire almeno durante un caso di prova

Principi generali del testing del software

1. *Il test non dovrebbe mai essere eseguito da chi ha sviluppato il software*
2. *Il test costa e quindi deve essere ottimizzato*
3. *Un caso di prova deve specificare in anticipo i risultati attesi*
4. *Oltre alle condizioni “normali”, il test deve verificare anche quelle “anomale” (limiti, errori, casi particolari)*
5. *Considerare i casi di prova come un investimento (un asset)*
6. *La probabilità di trovare errori durante il test dipende dal numero di errori trovati*

Testing e qualità del software

- ❑ Che cos'è la qualità del software
 - “Capacità di soddisfare i requisiti, espliciti ed impliciti”
 - “Rispondenza alle specifiche applicabili”

- ❑ **ISO 9126** definisce un modello di qualità per il software
 - Qualità interna,
 - Qualità esterna
 - Qualità in uso
 - Definisce caratteristiche e sottocaratteristiche di qualità del software

Funzionalità	<i>Completezza, Accuratezza, Interoperabilità, Sicurezza, Aderenza alla funzionalità</i>
Usabilità	<i>Comprensibilità, Apprendibilità, Operabilità, Attrattività, Aderenza all'usabilità</i>
Efficienza	<i>Comportamento rispetto ai tempi, Utilizzo delle risorse, Aderenza all'efficienza</i>
Affidabilità	<i>Maturità, Tolleranza ai guasti, Recuperabilità, Aderenza all'affidabilità</i>
Manutenibilità	<i>Analizzabilità, Modificabilità, Stabilità, Provabilità, Aderenza alla manutenibilità</i>
Portabilità	<i>Adattabilità, Installabilità, Coesistenza, Sostituibilità, Aderenza alla portabilità</i>

Testing e qualità del software

❑ Qualità di prodotto e di processo

“La qualità del prodotto è direttamente connessa con la qualità del processo con cui è stato realizzato”

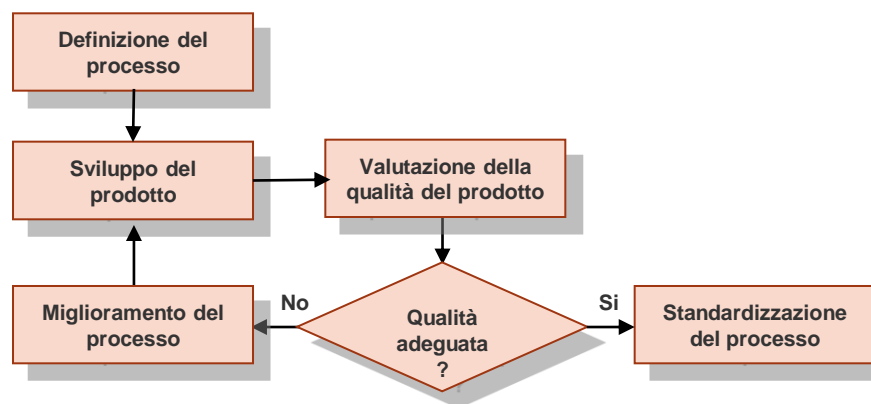


Figura 4. Qualità del prodotto e del processo

Testing e qualità del software

❑ **Assicurazione Qualità (QA)** *Product and Process Quality Assurance (PPQA)*

Standard i processo	Standard di prodotto
Procedura di revisione tecnica dei documenti	Modello di revisione tecnica
Procedura di approvazione dei piani di progetto	Modello dei piani di progetto
Procedura di gestione della configurazione	Modello del profilo della qualità del prodotto
Standard di programmazione	Modello del documento dei requisiti
Procedura di gestione dei difetti	Modello del documento di disegno
Procedura di registrazione dei test	Modello del documento delle specifiche funzionali
Procedura di gestione delle modifiche	Modello di richiesta di modifica
Procedura di rilascio in esercizio	Modello del documento delle specifiche di test

Testing e qualità del software

□ Approccio alla qualità del software

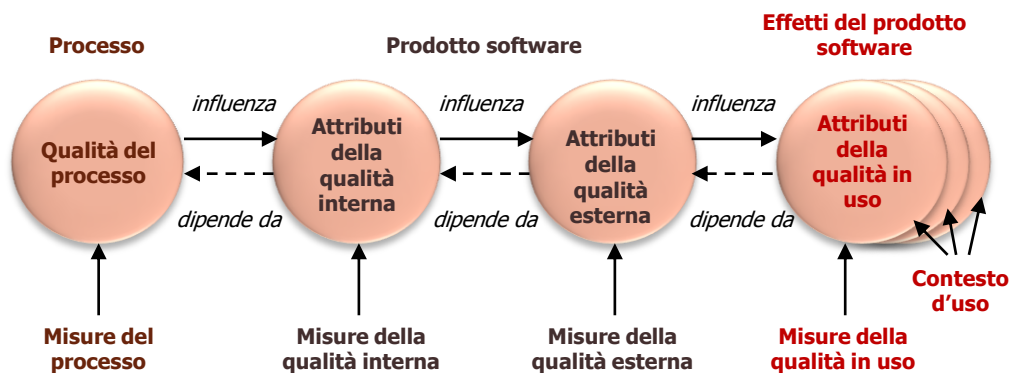


Figura 5. Qualità del prodotto software e CVS

Testing e qualità del software

□ La qualità del software e il ciclo di vita

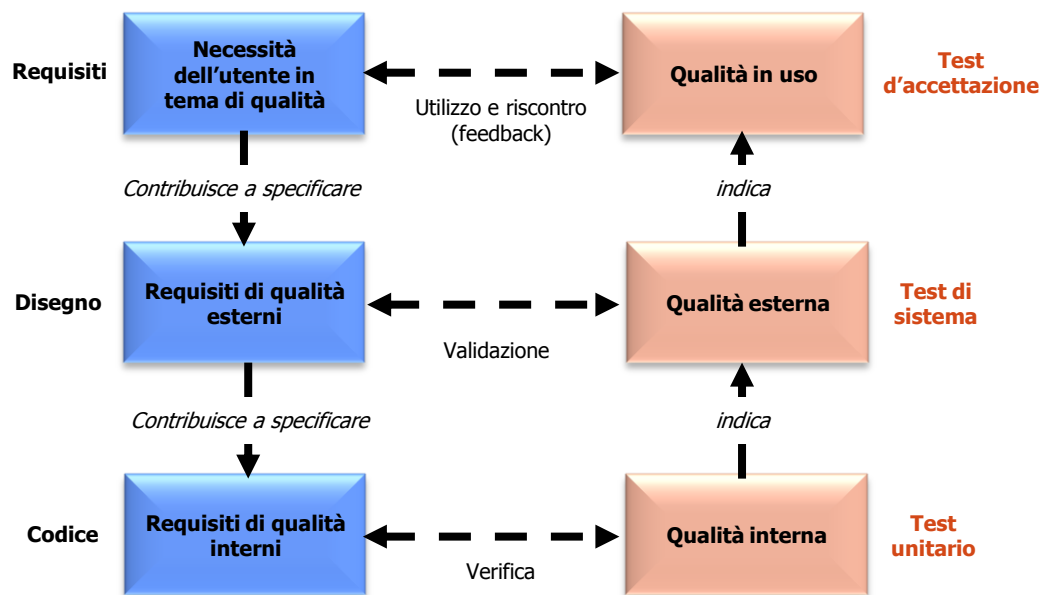


Figura 6. Qualità nel ciclo di vita del software

Grazie per l'attenzione

Ercole Colonese

Consulenza di direzione, organizzazione e IT

ercole@colonese.it

www.colonese.it