

Collaudo e qualità del software

Metodi e tecniche del testing

Relatore

Ercole Colonese

Roma, 29 novembre 2010



Metodi e tecniche del testing



Temi trattati nel libro

- Revisione tecnica
- Integrazione del software
- Testing a scatola aperta
- Testing a scatola chiusa
- Testing a scatola grigia
- Validazione dell'usabilità
- La previsione del risultato (*oracolo*)
- Valutazione e controllo dei risultati

Revisione tecnica

Una revisione tecnica è tanto semplice quanto efficace

“Chi inventerà come sostituire le ispezioni diventerà ricco”

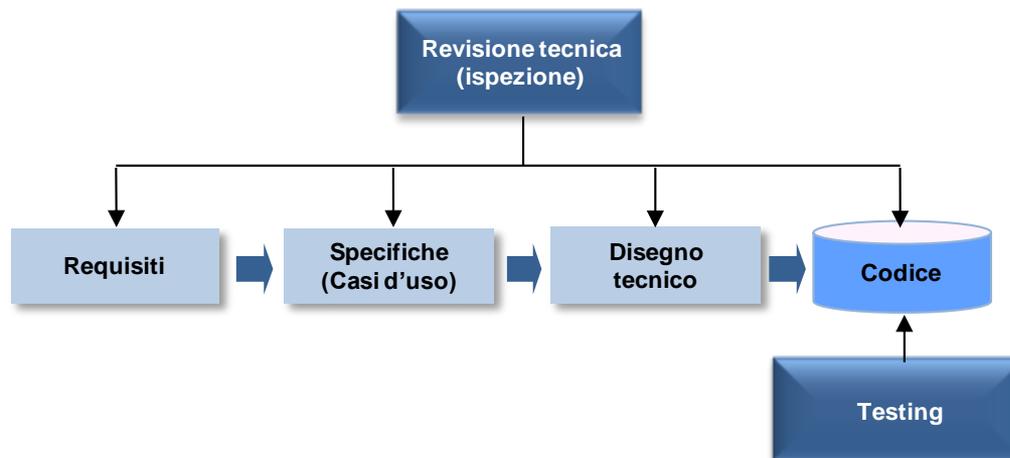


Figura 23. Sinergia tra ispezioni e testing

ISO 9001 a proposito delle verifiche

7.3.5 Verifica della progettazione e sviluppo

La verifica deve essere effettuata in conformità a quanto pianificato per assicurare che gli elementi in uscita dalla progettazione e sviluppo abbiano soddisfatto i requisiti in ingresso alla progettazione e sviluppo. Devono essere mantenute registrazioni dei risultati della verifica e delle eventuali azioni necessarie.”

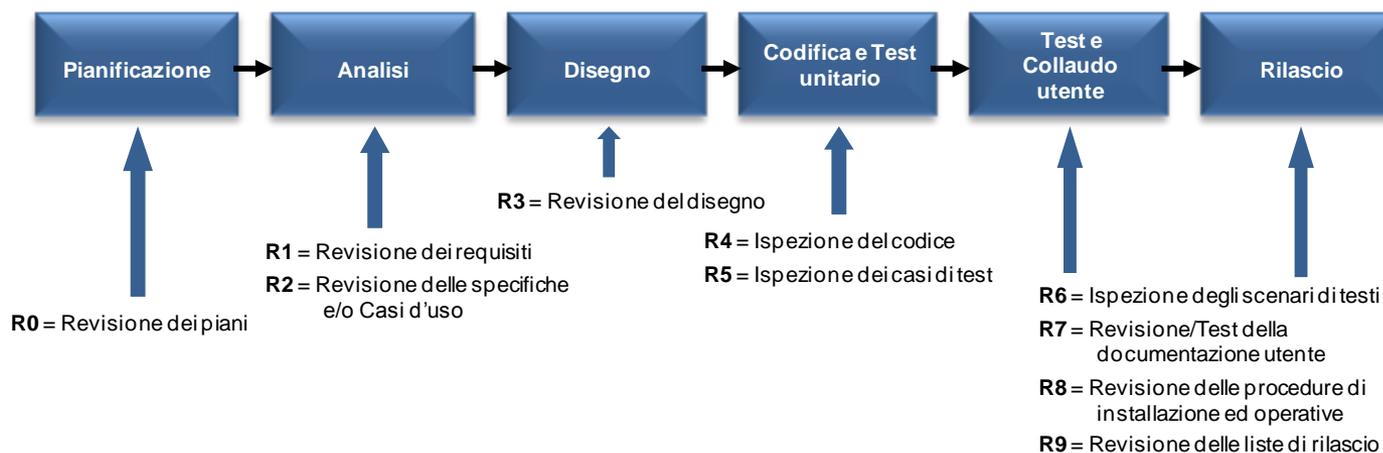


Figura 24. Revisioni tecniche (ispezioni) nel ciclo di vita del software

Ispezioni e walkthrough

Ispezione

- E' la revisione tecnica di un documento o del codice visto come documento
- E' eseguita da una persona tecnica con competenza sul tema trattato
- Si tratta di leggere attentamente il documento e scoprire eventuali errori
- Si esegue su un documento completo
- Gli errori rilevati sono comunicati all'autore in qualsiasi forma
- L'autore provvederà alla correzione nei tempi e nei modi più opportuni
- I ruoli coinvolti sono: autore e ispettore (uno o più)

Walkthrough

- E' una riunione in cui l'autore del documento spiega ai partecipanti (ispettori) i contenuti del documento ripercorrendo il flusso logico dei concetti trattati (simulando, in qualche maniera, l'eventuale percorso del sistema)
- Si applica principalmente al codice, ad un workflow, all'architettura, etc.
- Gli errori rilevati sono registrati (informalmente) durante la riunione stessa
- I ruoli coinvolti sono: autore e ispettore (uno o più)

Revisioni informali e formali

Revisione informale

- Si organizza tra colleghi senza alcun formalismo
- Gli errori sono comunicati in qualsiasi forma (è sempre meglio documentarli)
- Non si produce alcun verbale al termine
- La pianificazione è fatta a livello di progetto prevedendo i documenti da rivedere, il numero di ispezioni da eseguire e l'impegno necessario

Revisione formale

- Si pianifica quali documenti sottoporre a ispezione, chi deve partecipare e l'impegno necessario per l'attività
- Si segue un formalismo (processo strutturato)
- Gli errori sono registrati così come pure la loro correzione
- Al termine si produce un verbale
- I ruoli coinvolti sono: autore, ispettore (uno o più), moderatore (uno)

Ruoli

Ruolo	Descrizione
Autore	E' l'autore del documento. Consegna il documento completo agli ispettori. Recepisce i commenti, le osservazioni e gli errori. Corregge il documento.
Ispettore	Ha competenza sull'argomento trattato. Effettua la revisione tecnica del documento. Prende nota degli errori, osservazioni e commenti. Comunica all'autore i propri commenti, osservazioni e errori.
Moderatore	E' coinvolto solo nelle revisioni "formali". Garantisce il corretto svolgimento del processo ispettivo. Svolge anche il ruolo di ispettore.
Manager	Non è mai coinvolto nel processo per evitare giudizi sulle persone (l'ispezione serve a scoprire gli errori nel prodotto e non a valutare le persone). E' invece compito del manager pianificare le ispezioni e valutare la loro efficacia e efficienza per migliorare il prodotto e il processo.

Processo formale (strutturato)

Fase	Descrizione
Pianificazione	Il manager pianifica le ispezioni formali (documenti da revisionare, partecipanti, stima degli impegni, fasi in cui effettuare le ispezioni)
Distribuzione del materiale	L'autore distribuisce agli ispettori il documento completo e ne illustra brevemente i contenuti ed eventualmente le sezioni da sottoporre a ispezione (*)
Revisione del materiale	Ciascun ispettore effettua la revisione del materiale autonomamente nei tempi previsti. Gli errori, commenti ed osservazioni sono registrati in un modulo.
Riunione congiunta	In una riunione congiunta l'autore scorre il documento e ciascun ispettore illustra le proprie osservazioni ed errori rilevati. L'autore raccoglie i moduli compilati. Gli errori sono classificati secondo la gravità concordata (grave, lieve, osservazione). Il moderatore produce un verbale (report).
Correzione degli errori	L'autore provvede a correggere gli errori rilevati
Verifica delle correzioni	Il moderatore verifica con l'autore che gli errori gravi e lievi siano stati corretti.
Chiusura dell'intervento	Il moderatore chiude il processo quando tutti gli errori siano stati corretti e produce un apposito verbale (report)

(*) Le ispezioni si possono concentrare anche solo su parti del documento ritenute critiche ai fini della qualità del prodotto (efficienza).

Alcune note in merito

Elementi basilari delle ispezioni

- La revisione di un documento si effettua solo quando esso è *completo*
- Se i tempi e le risorse a disposizione non consentono la revisione di tutto il documento, *concentrarsi sulle parti maggiormente critiche* (soluzioni tecniche particolari, introduzione di nuove tecnologie per l'azienda, algoritmi particolari, struttura dei dati, casi d'uso, elementi critici per la qualità, ecc.)
- La revisione è fatta da *personale competente*
- La revisione deve essere *pianificata* per consentire alla persona coinvolta di dedicare il tempo necessario alle attività di revisione (preparazione e riunione di revisione) compatibilmente con i suoi impegni
- La revisione ha lo scopo di *scoprire eventuali problemi* nella progettazione prima di iniziare lo sviluppo e non quello di giudicare il lavoro o la persona
- Gli eventuali problemi riscontrati sono *discussi e corretti* opportunamente
- La revisione è svolta utilizzando opportune *liste di controllo (Checklist)* realizzate per ciascun tipo di documento in base all'esperienza e mantenute aggiornate

Integrazione del software

- ❑ L'integrazione mira a costruire il prodotto finale man mano che si completano i vari componenti, sottocomponenti e singoli moduli
- ❑ L'integrazione si svolge secondo la strategia di sviluppo e di test adoperata (priorità nello sviluppo dei componenti)
- ❑ L'integrazione utilizza la scomposizione del prodotto in singole unità (Product Breakdown Structure - PBS)

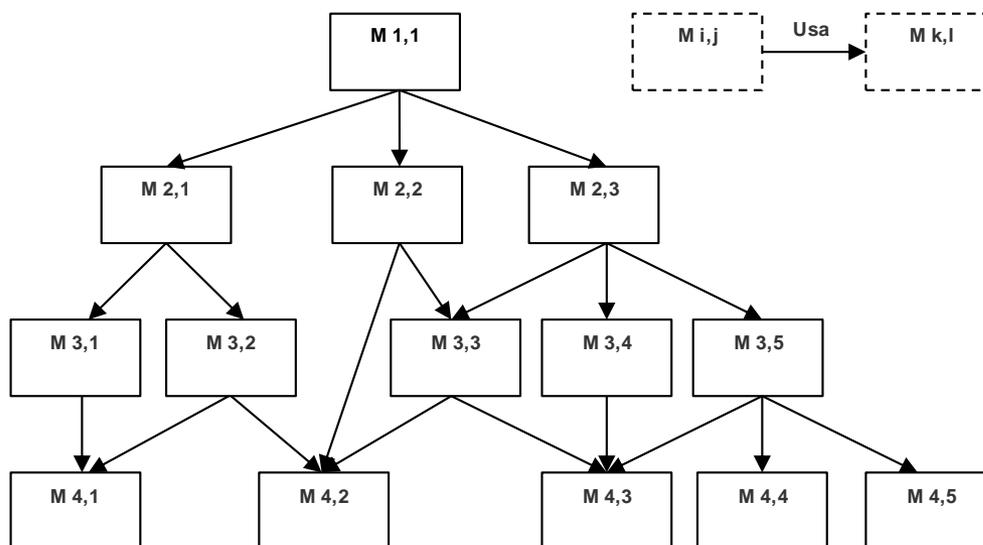


Figura 25. Scomposizione del prodotto software (PBS)

Integrazione del software

- ❑ La pianificazione dell'integrazione richiede un'organizzazione, strumenti e componenti a supporto adeguati
- ❑ L'integrazione può essere svolta in tre modi diversi
 - Integrazione **Top-down**
 - Integrazione **Bottom-up**
 - Integrazione **mista** (entrambe le tecniche precedenti)
- ❑ I componenti non ancora disponibili per l'integrazione e simulati da altri componenti "fantasma" (*scaffolding*) sono di due tipi
 - **Driver**
 - **Stub**

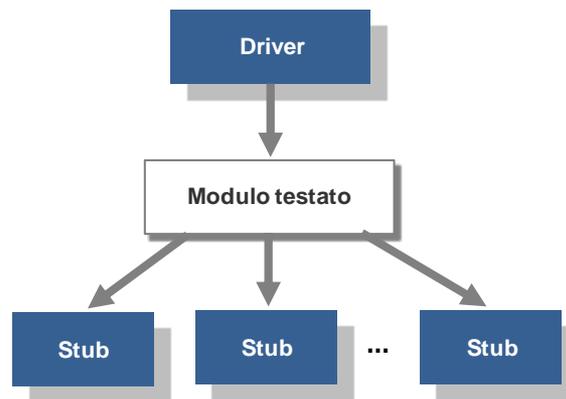
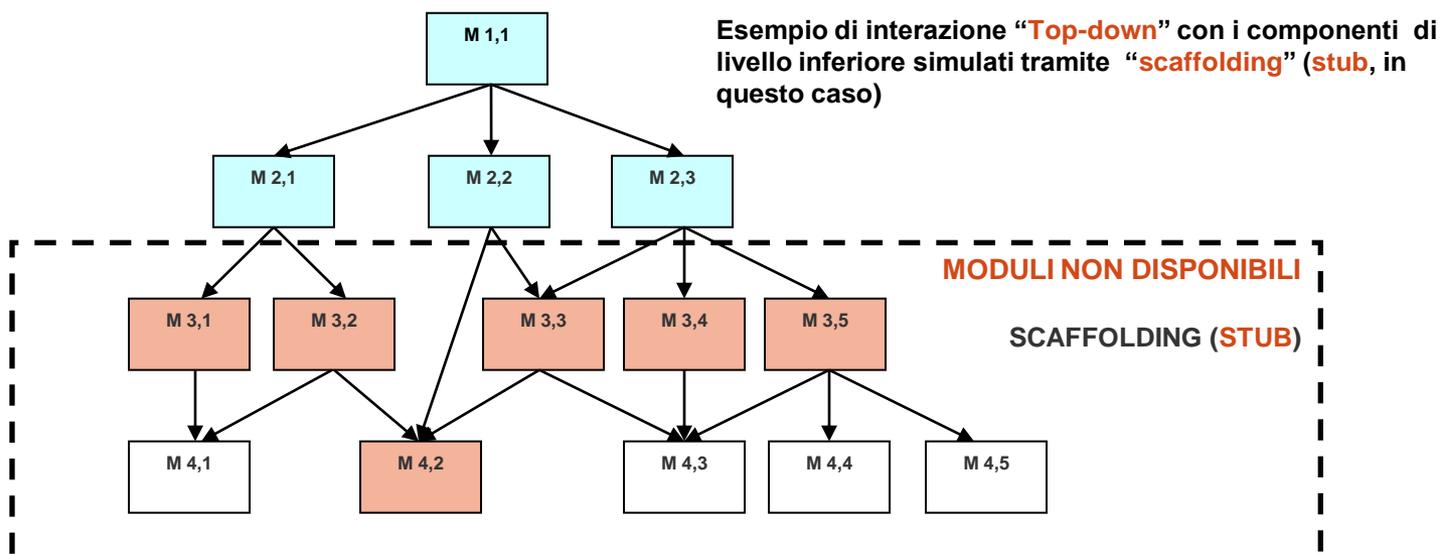


Figura 26. Driver e Stub nel test del software

Integrazione “Top-down”

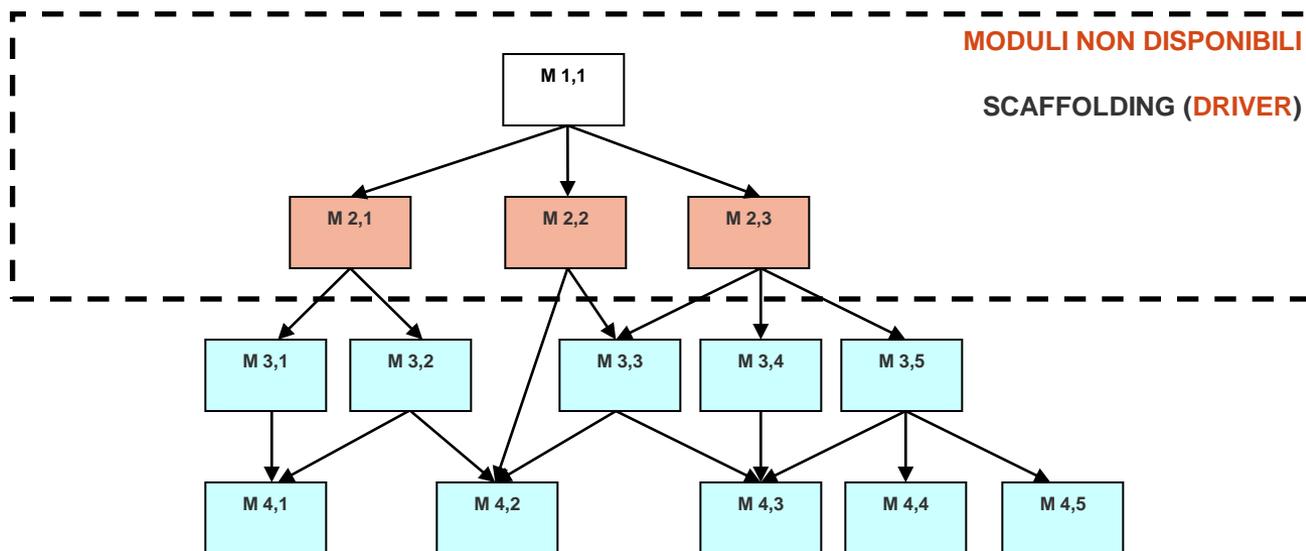
- L'integrazione non è quindi casuale ma viene programmata secondo tecniche diverse
 - L'integrazione **Top-down** richiede che siano simulati i componenti di livello più basso con “**stub**”(nell'esempio in figura)



Integrazione “Bottom-up”

- L'integrazione “**Bottom-up**” parte dal basso e richiede che siano simulati i componenti di più altro livello con dei “**driver**”

Esempio di interazione “**Bottom-up**” con i componenti di livello superiore simulati tramite “**scaffolding**” (**Driver**, in questo caso)



Tecnica di testing “White-box”

- ❑ E' una tecnica di testing focalizzata sulla struttura interna del codice
- ❑ E' un test “strutturato” perché segue la struttura interna del codice
- ❑ E' quindi una tecnica “**logic-driven**” e richiede la conoscenza della logica del programma
- ❑ E' perciò utilizzata dai programmatori che hanno sviluppato il codice
- ❑ Si tratta del “Test unitario” (Unit test)
- ❑ Si testano
 - i percorsi logici del programma
 - i flussi primari e quelli secondari
 - le condizioni di selezione
 - le combinazioni particolari
- ❑ Si utilizzano strumenti appositi di sviluppo (**debugger**) che permettono di agire direttamente sul codice sorgente
 - Eseguire il codice partendo da istruzioni selezionate
 - Fermare l'esecuzione in punti precisi e farla ripartire
 - Visualizzare il contenuto dei campi e modificarlo a piacimento
 - etc.

Testing “White-box”

La tecnica “White-box” permette di eseguire tipologie diverse di verifiche

- Ispezione del codice (*Code Inspection*)
- Valutazione della complessità del codice (*Code Complexity Evaluation*)
- Rimozione degli errori (*Debugging*)
- Copertura del codice (*Code coverage*)
- Test casuale (*Fuzz Testing*)
- Inserimento di errori (*Error Injection*)
- Test di mutazione (*Mutation Testing*)

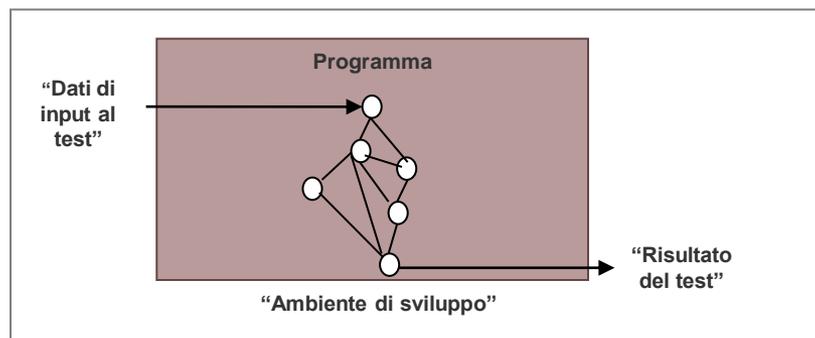


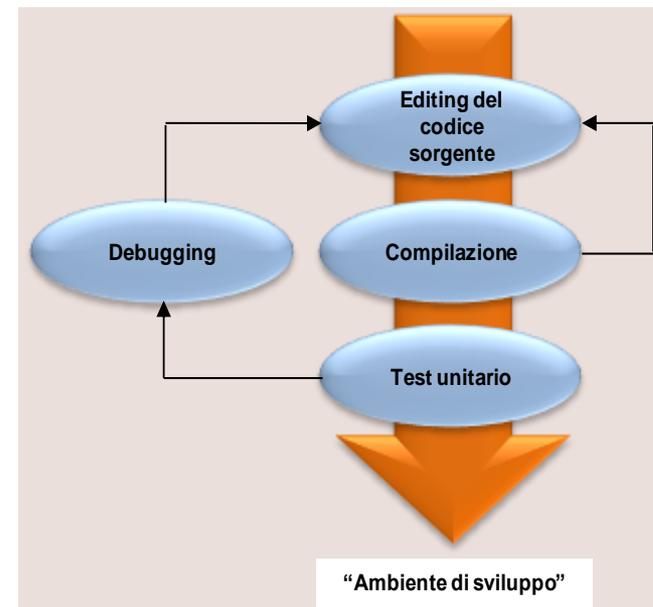
Figura 29. Test a “scatola aperta” (White-box)

“Debugging”

- ❑ Rappresenta la tecnica per eccellenza utilizzata dagli sviluppatori per correggere gli errori (bug)
- ❑ L'attività di debugging può risultare anche molto difficile a seconda della complessità del software, della facilità di comprensione del codice, della possibilità di riprodurre le condizioni di errore, etc

Il processo di debugging prevede:

- Analisi del problema e identificazione della causa del malfunzionamento
- Individuazione del componente in errore che ha generato il malfunzionamento
- Progettazione e implementazione della correzione dell'errore
- Testing della correzione apportata e verifica dell'eliminazione del malfunzionamento



Testing “Black-box”

- ❑ La tecnica “**Black-box**” permette di verificare il comportamento esterno del software
- ❑ E’ governata dall’immissione dei dati in ingresso (**Data-driven**)
- ❑ Il software è esercitato tramite i **casi di prova** progettati in base alle funzionalità richieste (**requisiti funzionali**) e in base alle caratteristiche strutturali richieste (**requisiti non funzionali**) come l’usabilità, le performance, il carico di lavoro capace di supportare, etc

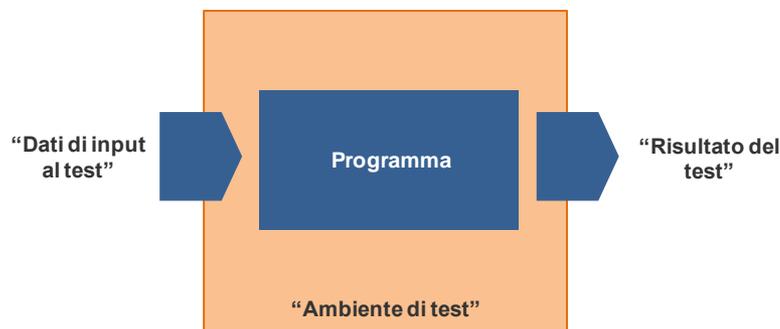
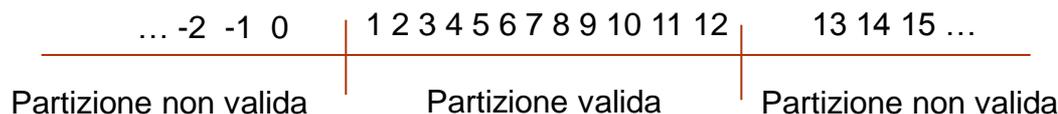


Figura 33. Test a “scatola chiusa” (Black-box)

Testing “Black-box”: metodi e tecniche

L’approccio “Black-box” consente di utilizzare diversi metodi e tecniche

- Partizionamento dei dati in ingresso (*Equivalence Partitioning*)
- Valori al contorno (*Boundary Value Analysis*)
- Previsione degli errori (*Error Guessing*)
- Errori casuali (*Fuzzing*)
- Utilizzo di un oracolo (*Oracle*)



Esempio di partizionamento dei dati in ingresso (caso del valore numerico di un mese dell’anno)

Testing “Gray-box”

- ❑ L’approccio “**Gray-box**” è una tecnica mista delle due tecniche viste prima (White-box e Black-box)
- ❑ Si “entra nel codice” per “modificarlo” e indurre un comportamento “esterno” del software altrimenti difficile o impossibile da creare tramite i dati in ingresso dei casi di prova
- ❑ Le tecniche adoperate sono
 - Iniezione di errore (*Fault Injection*)
 - Mutazione del codice (*Mutation Test*)

Verifica e validazione dell'usabilità

La verifica e validazione dell'usabilità si effettua in due momenti diversi del ciclo di vita e con due tecniche diverse

❑ **Verifica dell'usabilità**

Svolta durante la progettazione delle interfacce

- **Revisione Euristica dell'Usabilità**

❑ **Validazione dell'usabilità**

Svolta durante la fase di System Test

- **Test di usabilità (*Usability Test*)**

Verifica dell'usabilità

❑ **Revisione Euristica dell'Usabilità**

Svolta durante la progettazione delle interfacce

- E' svolta da uno o più esperti di usabilità applicando linee guida, standard e esperienza personale acquisita sul tema
- Valuta il livello di usabilità raggiunto o che può essere raggiunto
- Identifica le aree critiche per l'usabilità e suggerisce azioni di miglioramento
- Si svolge con una riunione e può essere ripetuta più volte per valutare i progressi
- Elementi presi in considerazione ai fini dell'usabilità
 - Aderenza agli standard stabiliti per l'usabilità
 - Consistenza grafica e usabilità dei dialoghi
 - Validità della messaggistica

Verifica dell'usabilità

- Le **metriche** dell'usabilità sono soggettive per definizione

Caratteristica di usabilità	Metrica
Facilità di comprensione	<ul style="list-style-type: none"> Disponibilità di una Demo Comprensione/Intuizione di quali input sono richiesti e quali output sono prodotti
Facilità di apprendimento	<ul style="list-style-type: none"> Disponibilità di un Tutorial Facilità di imparare
Facilità d'uso (operabilità)	<ul style="list-style-type: none"> Interazione attrattiva, piacevole (in USA interfacce "sexy") Facilità di memorizzazione Interazione guidata Presenza di una guida per inesperti Presenza di una guida per tutti gli utenti Assenza di errori Tempo richiesto per completare un'operazione (task) Assenza di confusione dovuta a messaggi chiari Capacità di recupero dopo un errore utente Numero di tentativi per completare un'operazione (task) Capacità di recupero

- Il **processo** seguito è composto dalle seguenti fasi



Figura 34. Processo di revisione euristica del software

Validazione dell'usabilità

□ Test di usabilità (*Usability Test*)

Svolto durante la fase di System Test

- Verifica se il livello di usabilità raggiunto dalla soluzione finale permette agli utenti di utilizzarla agevolmente
- E' svolto da personale specializzato sui temi dell'usabilità del software, sugli standard applicabili, sull'operatività degli utenti finali e sull'accessibilità
- Tali competenze sono generalmente assenti nelle aziende software e devono essere ingaggiate dall'esterno con aggravio sui costi del progetto (generalmente non previsti dalla pianificazione)
- Il test di usabilità è svolto in un ambiente particolare (*Usability Lab*)



Validazione dell'usabilità

□ Test di usabilità (*Usability Test*)

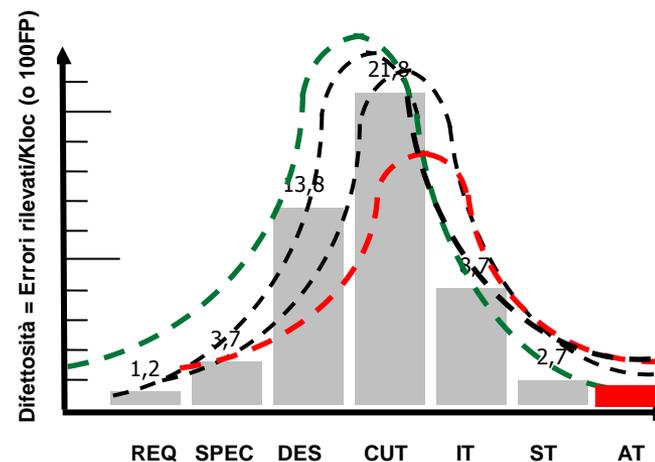
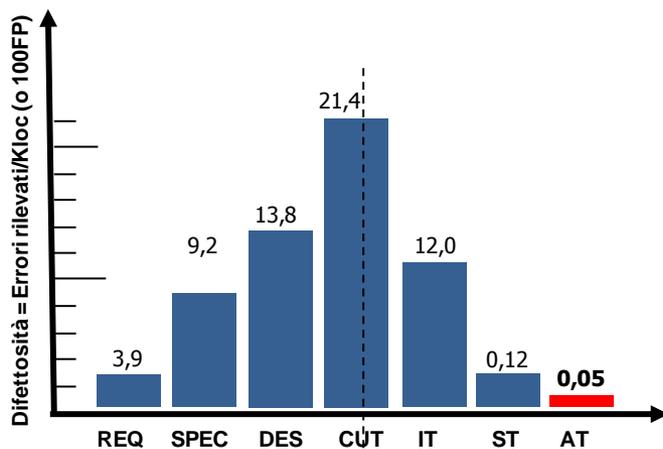
- In sintesi, si tratta di far eseguire i casi di prova selezionati a utenti finali e osservare il loro comportamento durante l'esecuzione delle prove
 - *difficoltà incontrate, reazioni, tempo di completamento, ecc.*
- Le osservazioni portano gli esperti a formulare un giudizio sull'usabilità raggiunta e a suggerire correzioni alle interfacce
- Una volta eseguite le correzioni, il test prosegue con altri utenti per verificare il miglioramento del livello di usabilità
- I test proseguono finché non si raggiunge il livello desiderato oppure finché non si raggiunge un compromesso tra tempi, costi e usabilità

Nota. La difficoltà a eseguire test di usabilità ed i costi relativi suggeriscono le seguenti considerazioni:

- L'usabilità si progetta fin dalle prime fasi (per esempio, realizzando prototipi)
- L'usabilità va presa in seria considerazione quando siano stati definiti espliciti i requisiti in proposito
- Ogni organizzazione software dovrebbe formare personale interno sul tema dell'usabilità e l'accessibilità, definire i processi ed attrezzarsi con strumenti

Controllo e monitoraggio del test

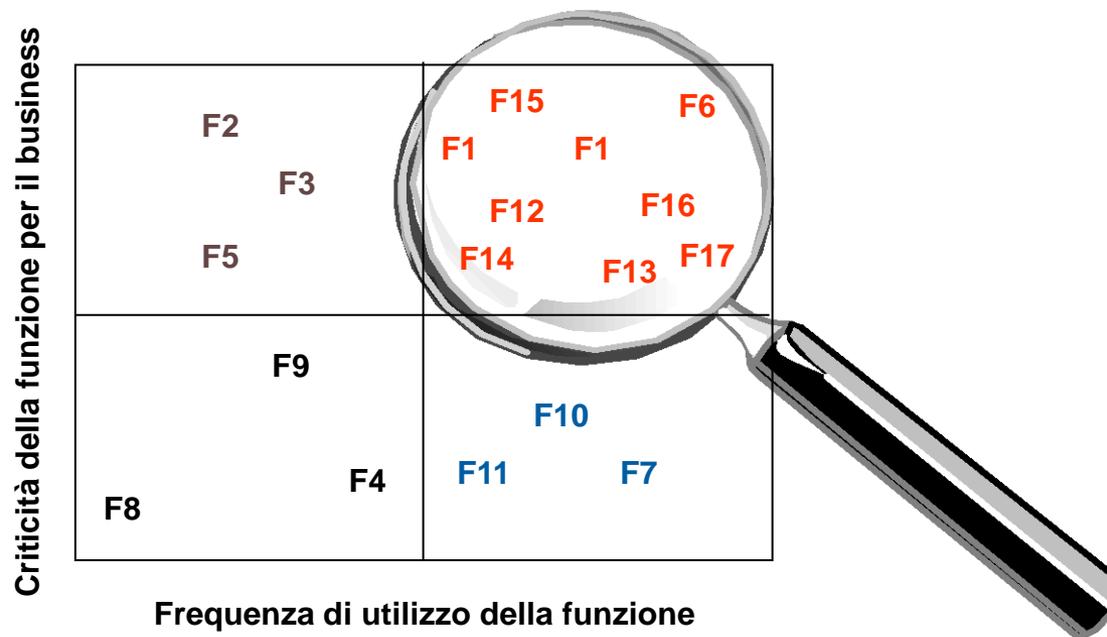
- Immissione e rimozione degli errori nel software tramite ispezioni e test
 - **Curva di rimozione degli errori durante il ciclo di vita**



Tasso di rimozione	Requisiti	Specifiche tecniche	Disegno	Codifica e Test unitario	Test d'integrazione	Test di sistema
Piano	3,9	9,2	13,8	21,4	12,0	0,12
Risultati	1,2	3,7	11,2	27,4	13,4	5,2

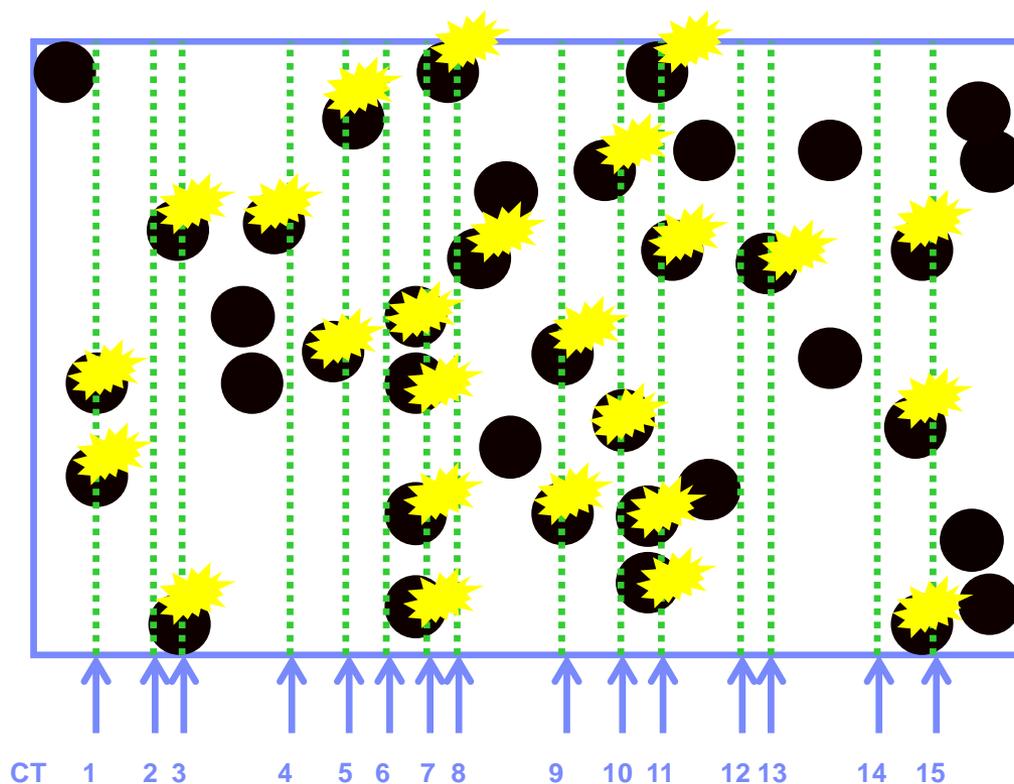
Controllo e monitoraggio del test

- ❑ Ottimizzazione dei casi di prova
- ❑ Progettazione dei casi di prova (flussi principali, flussi secondari, condizioni di errore, condizioni limite, ecc.)



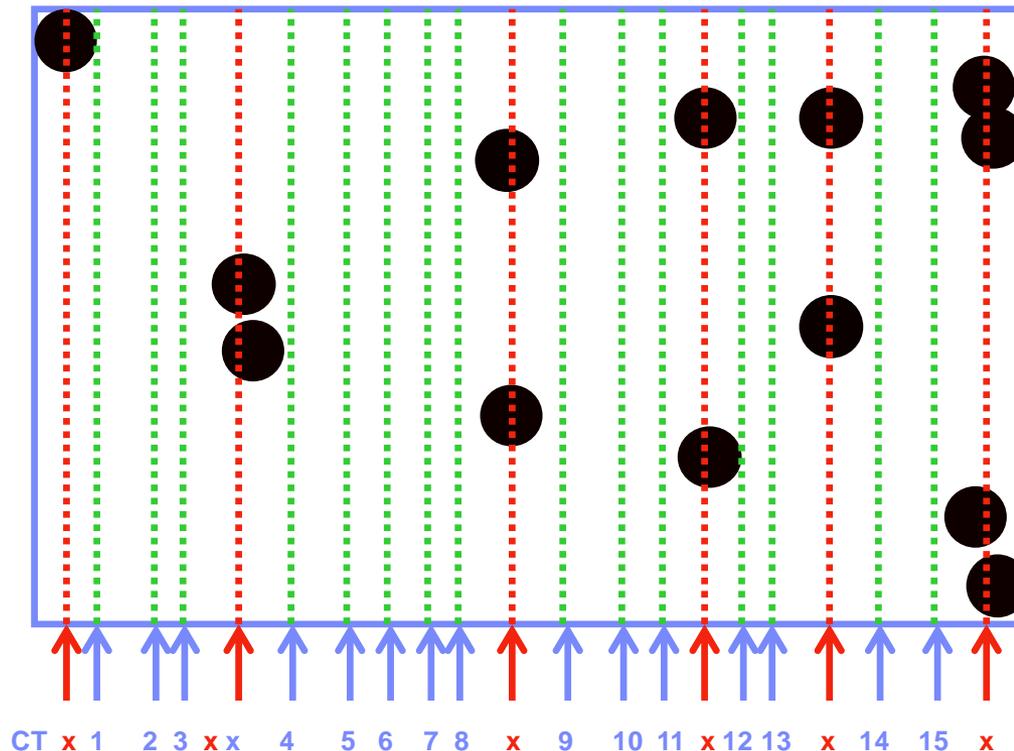
Controllo e monitoraggio del test

- ❑ Rimozione degli errori durante i test con i casi di prova progettati



Controllo e monitoraggio del test

- ❑ Errori residui nel software testato
- ❑ Occorrono ulteriori casi di prova (particolari)



Controllo e monitoraggio del test

❑ Registrazione degli errori rimossi durante i test (in forma cumulativa)

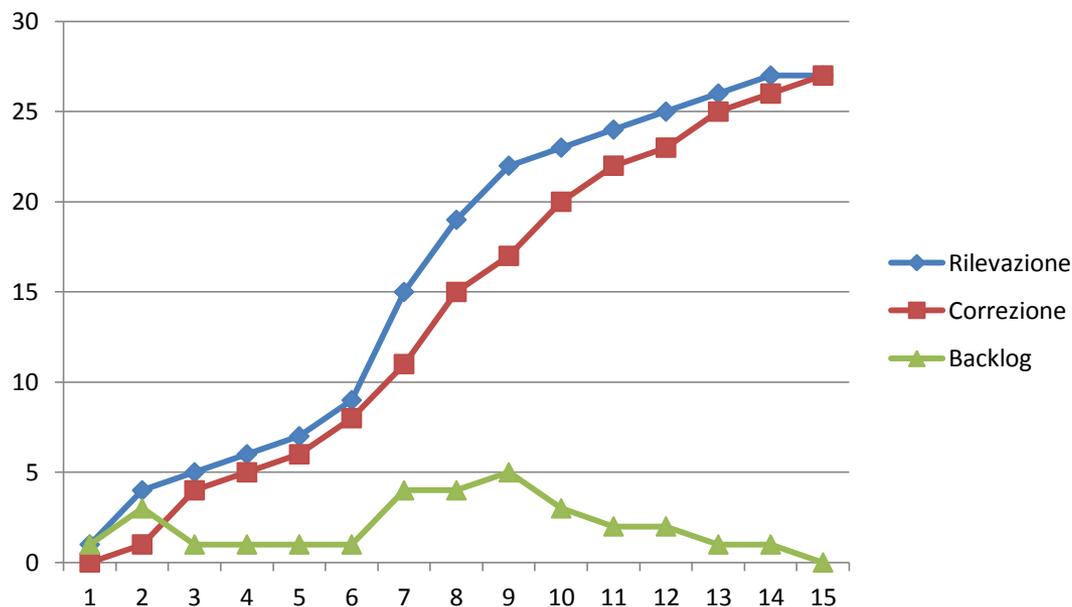


Figura 22. Andamento cumulativo delle curve degli errori durante il test

Controllo e monitoraggio del test

Curva di saturazione degli errori rilevati

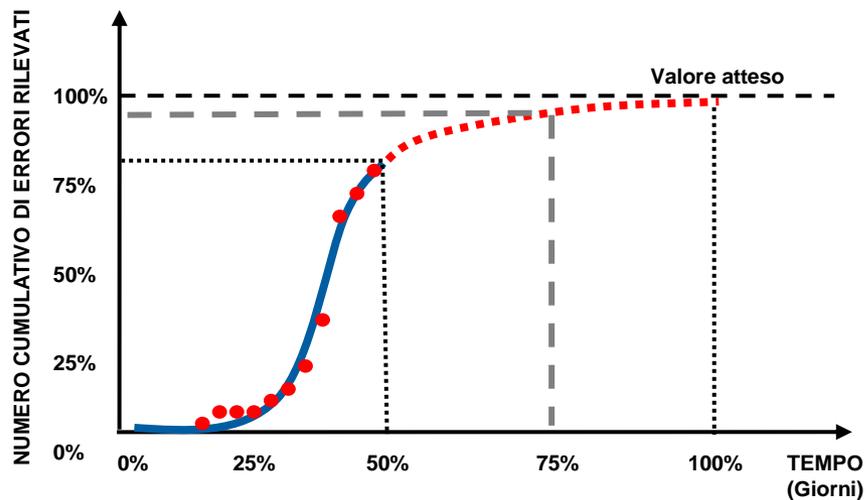
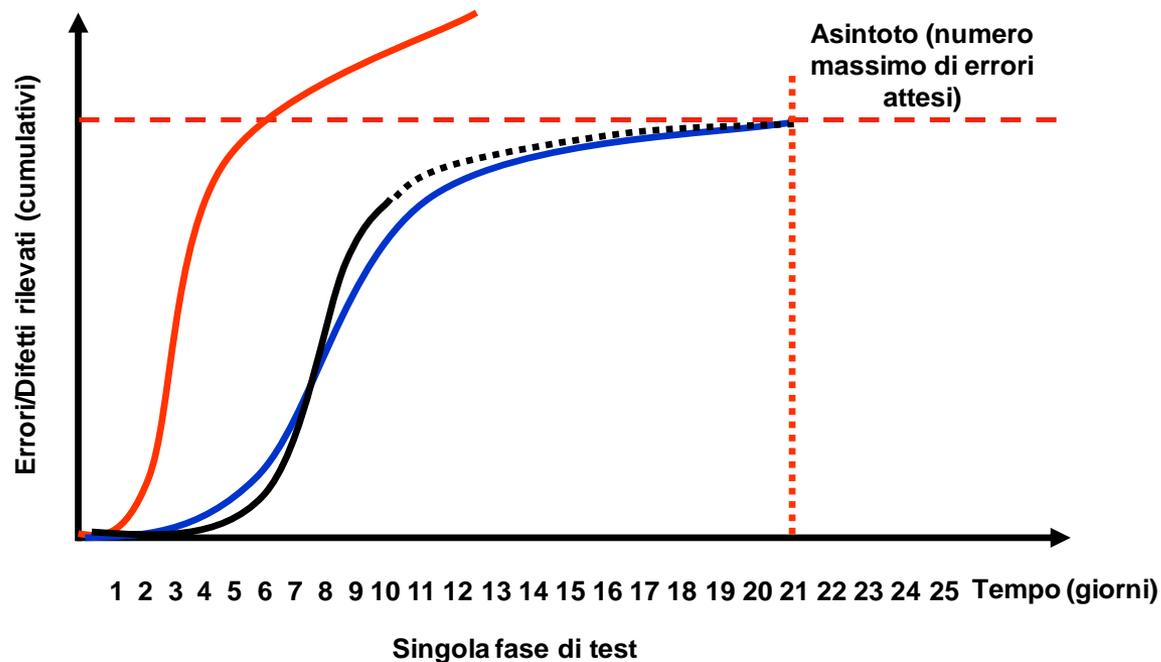


Figura 38. Curva "reale" di rimozione degli errori durante il test del software

Controllo e monitoraggio del test

- ❑ **Diverse tendenze della curva di saturazione degli errori rilevati rispetto all'asintoto (errori attesi)**



Controllo e monitoraggio del test

- ❑ **Quando fermare i test?**
- ❑ **Quale sarà il valore degli errori residui?**
- ❑ **Quale sarà l'impatto sugli utenti finale in esercizio?**
- ❑ **E' un problema di Risk Management!**

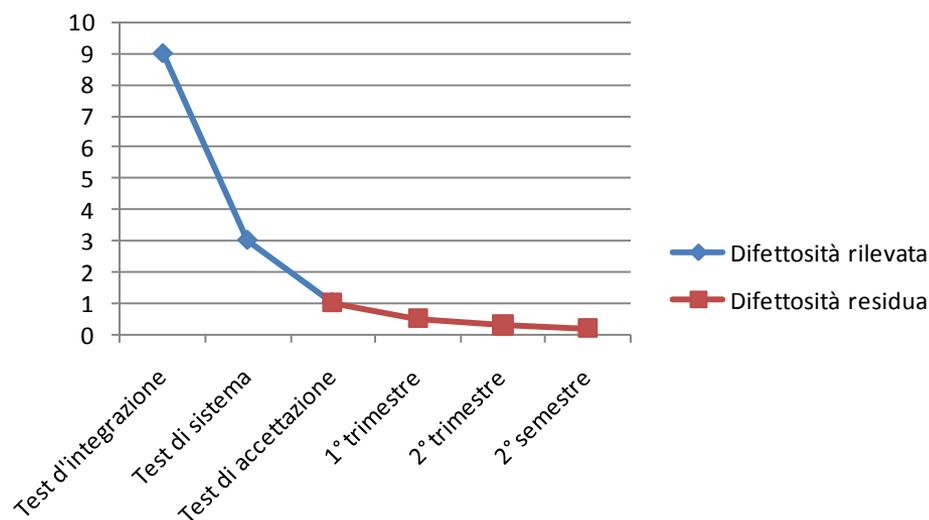


Figura 40. Curva di estrapolazione degli errori residui nel software

Grazie per l'attenzione

Relatore

Ercole Colonese, Consulente di Direzione, Organizzazione e IT

ercole@colones.it

www.colonese.it