Test del Software

Definizione

- Verifica dinamica del comportamento del software rispetto a quello atteso, utilizzando un insieme finito di casi di test, appropriatamente selezionati nel dominio di tutti i casi possibili nel dominio applicativo.
  - Dinamico, perché il test prevede l'esecuzione del programma
  - Atteso, perché dalle specifiche deve essere possibile conoscere a priori il comportamento del software ai casi di test
  - Finito, perché, in pratica, è possibile sottoporre il software a pochi casi di test
  - Selezionati, perché le tecniche consentono di scegliere i casi che partecipano alla verifica

SCOPO

- Identificare la presenza di difetti di un software
- Valutare la rispondenza del software realizzato ai requisiti

LIMITI DEL TEST

- Non vi è garanzia che se alla n-esima prova una componente od un sistema abbia risposto correttamente, altrettanto possa fare alla (n+1)-esima
- Non si possono produrre tutte le possibili configurazioni di input in corrispondenza di tutti i possibili stati interni di un sistema software
- Non è detto che gli strumenti e la preparazione e la esecuzione dei test non introducano altri difetti
- Non consente di rilevare le funzioni omesse

APPROCCI AL TEST ...

⇨TEST FUNZIONALE. La componente od il sistema viene vista come BLACK BOX ; ad esso viene sottoposto un insieme di input che attivino tutte le transazioni previste dalla stessa componente, ad ognuno di questi corrisponde un output atteso, confrontando l'output ottenuto con quello atteso si valuta l'esito del test

... APPROCCI AL TEST

⇨TEST STRUTTURALE. La componente o sistema viene visto come WHITE BOX e ne vengono verificati i dettagli implementativi; in questo caso gli input a cui è sottoposto il software devono verificare tutti i percorsi all'interno del software. Anche in questo caso l'esito del test è dato dal confronto dell'output atteso con quello reale.

LIVELLI DI TEST

⇨Unit
⇨Integrazione
⇨Sistema
⇨Accettazione

Unit Testing

Unit

⇨Un insieme di uno o più componenti del software con associati dati di controllo, procedure di uso ed operative, che soddisfi le seguenti condizioni.

- Tutte le componenti appartengano allo stesso sistema
- Almeno una tra le componenti nuove o modificate non ha completato lo unit test
- L'insieme delle componenti con i dati e procedure associati costituisce l'oggetto unico del processo di test

Tecniche

⇨Può essere effettuato attraverso le seguenti tecniche:

- Partizione in classi di equivalenza
- Analisi dei valori estremi
- Strutturale

Classi di Equivalenza

Classi di Equivalenza

⇨E' una tecnica di test di tipo funzionale

⇨Ogni componente del sistema è vista come black-box

Tecnica delle Classi di equivalenza

⇨Ripartire il dominio dei dati in input in sottoinsiemi tali che per tutti i valori di ogni sottoinsieme la componente si comporti allo stesso modo.

⇨Ogni sottoinsieme così individuato è detto Classe di equivalenza

Partizioni in Classi di Equivalenza

⇨Passi da eseguire per il disegno dei casi di test

1. Identificare le classi di equivalenza
2. Progettare i casi di test a copertura delle classi

Identificazione delle Classi di Equivalenza...

⇨Le classi di equivalenza si identificano analizzando le condizioni sui dati in input.

⇨Esse possono essere:

- Valide, se comprendono valori di dati in input validi
- Non valide, se comprendono valori di dati di input non validi

... Identificazione delle Classi di Equivalenza

⇨L'identificazione viene effettuata considerando ciascuna condizione in ingresso e prevedendo per ognuna di queste:

- Una o più classi di equivalenza valide
- Una o più classi di equivalenza non valide

Criteri della Identificazione delle Classi di equivalenza

⇨I criteri per la ricerca delle classi possono essere:

- Intervalli di valori
- Insiemi di valori
- Condizioni vincolanti

Intervallo di valori

⇨Se una condizione d'ingresso specifica un intervallo di valori allora devono essere identificate:

- Una classe di equivalenza valida per i valori compresi nell'intervallo
- Due classi di equivalenza non valide di cui:
  - una per i valori inferiori all'estremo sinistro dell'intervallo
  - una per i valori superiori all'estremo destro dell'intervallo

Intervallo di valori: Esempio

⇨Condizioni d'ingresso:

- Il Codice Cliente può avere una valore compreso tra "001" e "999"

⇨Classi di equivalenza:

- Valida CE_1 : $001 \leq COD \leq 999$
- Non valide: CE_2 : $COD < 001$ e CE_3 : $COD > 999$

Insiemi di valori

⇨Se una condizione d'ingresso specifica un insieme di valori allora devono essere identificate:

- Una classe di equivalenza valida per ciascun elemento dell'insieme
- Una classe di equivalenza non valida per un elemento non appartenente all'insieme

Insiemi di Valori: Esempio...

Condizioni di ingresso:

Il Tipo_Cliente può essere : Privilegiato, Normale o Potenziale.

... Insiemi di Valori: Esempio

⇨Classi di equivalenza

- Valide:
  - CE_4: TIPO CL. = PRIVILEGIATO
  - CE_5: TIPO CL. = NORMALE
  - CE_6: TIPO CL. = POTENZIALE
- Non valida:
  - CE_7: TIPO CL. = DIFFICILE

Condizioni Vincolanti

⇨Se una condizione di ingresso specifica una situazione che deve essere, allora sono identificate:

- Una classe di equivalenza valida per gli elementi che verificano la condizione
- Una classe di equivalenza non valida per gli elementi che non verificano la condizione

Condizioni Vincolanti: Esempio

⇨Condizione di ingresso

Il Tipo_Prodotto deve essere uguale a Fabbricato

⇨Classi di equivalenza

- Valida
  - CE_8: TIPO PROD. = FABBRICATO
- Non valida
  - CE_9: TIPO PROD. = ACQUISTATO

Progettazione dei Casi di Test...

⇨Classi di equivalenza valide

- Le classi di equivalenza individuate devono essere utilizzate per identificare casi di test che:
  - Minimizzino il numero complessivo di test
  - Risultino significativi (affidabili)
- Si devono individuare tanti casi di test da coprire tutte le classi di equivalenza valide, con il vincolo che ciascun caso di test comprenda il maggior numero possibile di classi valide ancora scoperte.

... Progettazione dei Casi di Test

⇨Classi di equivalenza non valide

- Si devono individuare tanti casi di test da coprire tutte le classi di equivalenza non valide, con il vincolo che ciascun caso di test copra una ed una sola delle classi non valide

Progettazione dei Casi di Test : Esempio...

CONDIZIONI ESTERNE	CLASSI DI EQUIVALENZA			
	#CE	Valide	#CE	Non valide
Valore del Cod. cli. tra 001 e 999	CE_1	001 ≤cod. cli ≤ 999	CE_2 CE_3	Cod.cli< 001 Coc.cli> 999
Tipo di cliente: Privilegiato, Normale, Potenziale	CE_4 CE_5 CE_6	Privilegiato Normale Potenziale	CE_7	Difficile
Il tipo deve essere *Fabbricato*	CE_8	*Fabbricato*	CE_9	*Acquistato*

... Progettazione dei Casi di Test : Esempio...

I casi di test e le classi coperte sono:

Test case	TC_1	TC_2	TC_3
Cod. cli	005	005	005
Tipo cliente	Privilegiato	Normale	Potenziale
Tipo prodotto	Fabbricato	Fabbricato	Fabbricato
Classi coperte	CE_1, CE_4, CE_8	CE_1, CE_5, CE_8	CE_1, CE_6, CE_8

… Progettazione dei Casi di Test : Esempio

I casi di test e le classi coperte sono:

Test case	TC_4	TC_5	TC_6	TC_7
Cod. cli	000	1000	008	008
Tipo cliente	Normale	Normale	Difficile	Potenziale
Tipo prodotto	Fabbricato	Fabbricato	Fabbricato	Acquistato
Classi coperte	CE_2, CE_5, CE_8	CE_3, CE_5, CE_8	CE_1, CE_7, CE_8	CE_1, CE_6, CE_9

Analisi dei Valori Estremi

Concetti di base…

⇨E' un test funzionale basato sulla tecnica delle classi di equivalenza.

- Esplora le situazioni sugli estremi ed intorno agli estremi stessi delle partizioni di equivalenza
- L'esperienza mostra che i casi di test esploranti le condizioni estreme sono molto produttivi

… Concetti di base

⇨Le condizioni estreme sono quelle situazioni in cui i valori si trovano:

- direttamente sugli estremi delle classi di equivalenza
- appena al di sopra
- appena al di sotto

⇨agli estremi di

- classi di equivalenza d'ingresso
- classi di equivalenza di uscita

Intervallo dei valori: Esempio...

⇨Per ogni condizione d'ingresso individuare le classi di equivalenza.

- Valide per i valori sugli estremi dell'intervallo
- Non valide per i valori immediatamente sotto il minimo e sopra il massimo

... Intervallo dei valori: Esempio

⇨Esempio

Il valore dell'IVA è compreso tra 1 e 999.999

⇨Classi di equivalenza:

- Valida

CE_1: IVA = 0

CE_2: IVA = 999.999

- Non valida

CE_2 : IVA = -1

CE3 : IVA = 1.000.000

Insieme ordinato: Esempio...

⇨Se in ingresso o in uscita al programma c'è un insieme ordinato di elementi, le classi di equivalenza definibili sono:

- Valide

una per il primo elemento dell'insieme

una per l'ultimo elemento dell'insieme

- Non valide

una per l'elemento minore dell'estremo sinistro

una per l'elemento maggiore dell'estremo destro

... Intervallo dei valori: Esempio

⇨Esempio

I possibili valori di CODICE sono {A, B, C, ..., H}

- Valide

CE_1: CODICE = A

CE_2: CODICE = H

- Non valide

CE_3: CODICE = I

Casi di Test

⇨Si segue la stessa tecnica esposta per le Classi di Equivalenza.

Test Strutturale

Test Strutturale

⇨E' un test basato sui possibili percorsi effettuati dai dati all'interno della struttura della procedura (WHITE-BOX-TESTING)

Rappresentazione di una Procedura

Il grafo sarà costruito secondo la seguente notazione:

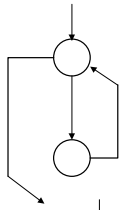
NODO  **rappresenta un'istruzione, sarà identificato da una lettera minuscola**

ARCO **rappresenta il passaggio del flusso di controllo, sarà identificato da un numero**

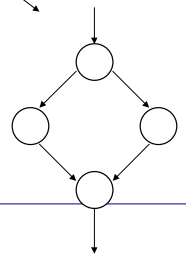
Costruzione di un Grafo...

Esempi di traduzione in grafo dei più comuni costrutti di decisione di salto

- **do while**

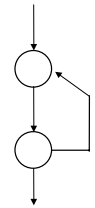


- **if then else**

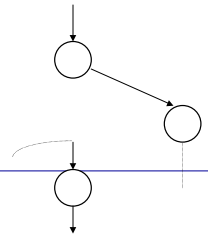


... Costruzione di un Grafo

- **do until**



- **go to**

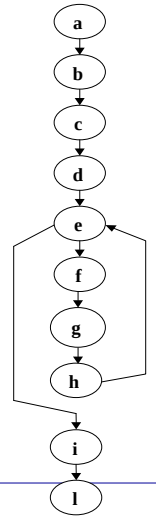


Costruzione di un Grafo : Esempio...

```
    begin
a       risult:=0
b       read dividendo
c       read divisore
d       resto:=dividendo
e       while (dividendo minore o uguale divisore)
        do
f               risultato:=risultato+1
g               dividendo:=dividendo-divisore
h               resto:=dividendo
        enddo
i       write risultato
l       write resto
    end
```

CHE DIVENTA

... Costruzione di un Grafo : Esempio

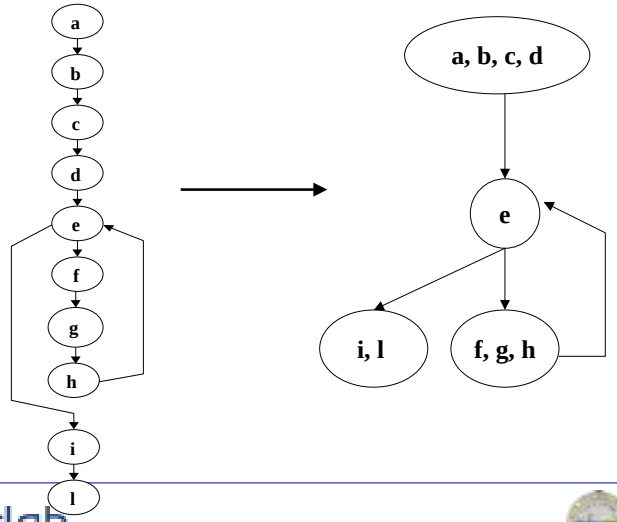


Semplificazione di un Grafo

⇨Un nodo è semplificabile se esso stesso, ed il suo successore, hanno un solo punto d'ingresso ed un solo punto di uscita

⇨I due nodi si riducono ad uno nella rappresentazione del grafo

Semplificazione di un grafo: Esempio

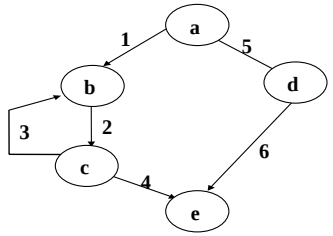


Determinazione dei percorsi

⇨E' possibile effettuare attraverso i seguenti metodi:

- Metodo Euristico con copertura delle istruzioni e delle decisioni
- Ricerca dei Percorsi Indipendenti con copertura minimale delle istruzioni e delle decisioni

Espressione di un percorso: Esempio



Arco	1	2	3	4	5	6	
Percorso							
a b c e	1	1	0	1	0	0	P_1
a b c b c e	1	2	1	1	0	0	P_2
a d e	0	0	0	0	1	1	P_3

P_{ij} = N.ro di attraversamenti dell'arco j-mo nel percorso P_i

Combinazione Lineare dei percorsi

Dicesi combinazione lineare dei percorsi $\{P_i\}$ la combinazione lineare dei vettori associati ai percorsi:

$a_1 P_1 + a_2 P_2 + ... + a_n P_n$

dove

- $P_1, ..., P_n$ sono i vettori associati ai percorsi 1, ..., n

e

- $a_1, ..., a_n$ sono coefficienti numerici della combinazione lineare

Combinazione Lineare dei percorsi: Esempio

Arco	1	2	3	4	5	6	
Percorso							
a b c e	1	1	0	1	0	0	P_1
a b c b c e	1	2	1	1	0	0	P_2
a d e	0	0	0	0	1	1	P_3
a b c b c b c e	1	3	2	1	0	0	P_4

$P_4 = 2 * P_2 + (-1) * P_1$

infatti

Coefficienti

$2 * P_2 + (-1) * P_1 =$
$=2(abcbce) - abce$ $= 2\,(1,2,1,1,0,0)\ -1\,(1,1,0,1,0,0) =$

$= (2,4,2,2,0,0) - (1,1,0,1,0,0) =$

$= \boxed{1,3,2,1,0,0} = P_4$

Percorsi Indipendenti

⇨Dato un grafo, è possibile individuare l'insieme dei percorsi che:

- sono linearmente indipendenti tra loro;
- consentono di generare tutti gli altri percorsi del grafo attraverso la loro combinazione lineare

Numero di percorsi indipendenti...

⇨Il numero dei percorsi indipendenti presenti in un grafo è calcolato usando una delle seguenti formule:

1) $C = A - N + 2$

dove

A = numero di archi

N = numero di nodi

2) $C = N_d + 1$

dove

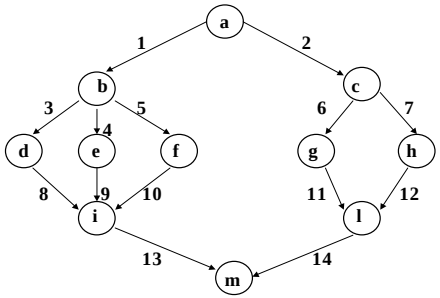
N_d = numero di nodi a due vie

... Numero di percorsi indipendenti

⇨Per nodi a n vie, ogni nodo vale come n-1 nodi a due vie.

⇨ Tale numero è detto numero ciclomatico.

Numero di percorsi indipendenti : Esempio



1) C = A - N + 2 = 14 - 11 + 2 = 5

2) C = N_d + 1 = 4 + 1 = 5

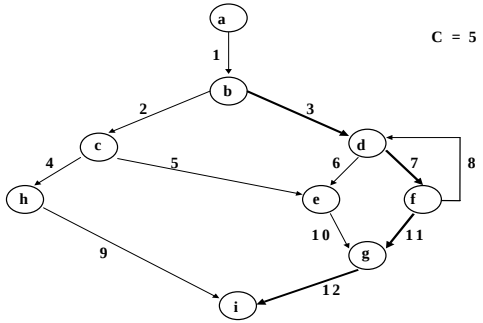
Ricerca dei Percorsi Indipendenti: Processo ...

⇨Il numero ciclomatico determina il numero di percorsi da provare per il test.

⇨La ricerca di tali percorsi è effettuata attraverso la tecnica delle baseline successive.

... Ricerca dei Percorsi Indipendenti: Processo...

Passo 1

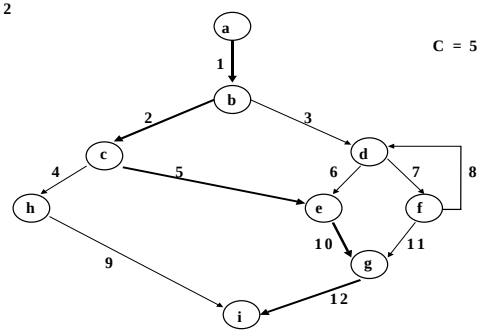


Individuare il percorso che contiene il maggior numero di punti di decisione.

Tale percorso è detto: BASELINE

… Ricerca dei Percorsi Indipendenti: Processo…

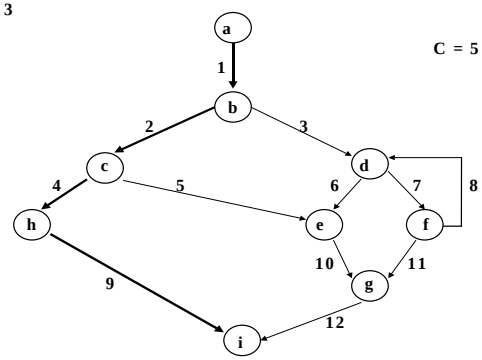
Passo 2



Il secondo percorso è uno di quelli alternativi al primo, secondo il primo punto di decisione della baseline, scegliendo sempre quello con il maggior numero di punti di decisione.

… Ricerca dei Percorsi Indipendenti: Processo…

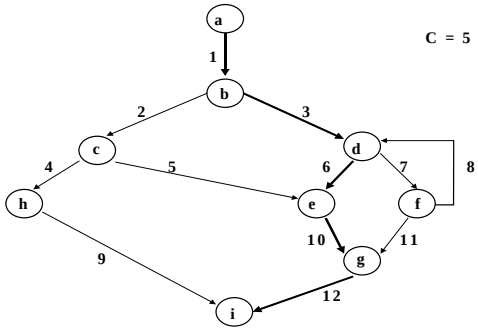
Passo 3



Individuare altri percorsi alternativi, cambiando man mano i punti di decisione appartenenti all'ultimo percorso ricavato.

… Ricerca dei Percorsi Indipendenti: Processo…

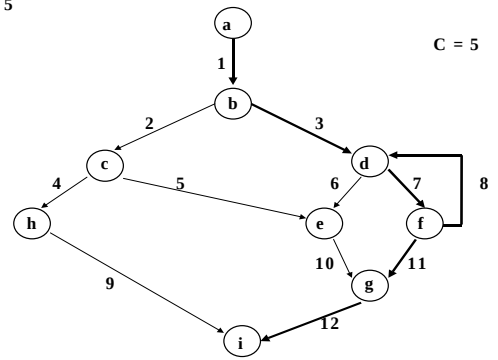
Passo 4



Quando sono terminati i percorsi alternativi per esaurimento dei punti di decisione, si ritorna alla BASELINE, si cambia il percorso nel successivo punto di decisione, mantenendo il maggior numero di punti di decisione.

… Ricerca dei Percorsi Indipendenti: Processo

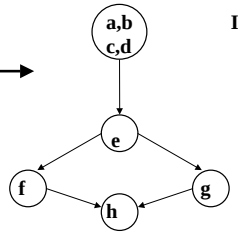
Passo 5



Iterare il passo 3 e il passo 4 finché tutti i punti di decisione non sono stati esaminati in ogni via o si è raggiunti il numero di percorsi indipendenti calcolato

Ricerca dei Percorsi Indipendenti: Processo : Esempio...

```
begin
a       READ B,D,F
b       A = B * D
c       C = D + 3
d       E = C + 2 * A
e       IF E - F < 10 THEN
f               PROCEDURA 1
    ELSE
g       PROCEDURA 2
h       END
```



I = (B,D,F)

Progettazione dei Casi di Test

⇨Si danno alle variabili di input i valori tali che siano attivati tutti i percorsi indipendenti.

... Progettazione dei Casi di Test : Esempio

⇨Percorsi: P_1: {a, b, c, d, e, f, h} e P_2: {a, b, c, d, e, g, h}

⇨La distinzione tra P_1 e P_2 è creata nel punto di decisione “Dec”. La funzione di decisione corrispondente è:

- Dec(D,F):E - F < 10 ≡ C + 2 * A - F < 10 ≡ D + 3 + 2 * A - F < 10 ≡ D + 3 + 2 * B * D - F < 10
- Dec = Vero ≡ D + 2 * B * D - F < 7: V_{11} = (3,2,8)
- Dec = Falso ≡ D + 2 * B * D - F ≥ 7: V_{21} = (3,2,7)
  - V_{11} è il caso di test che attiva P_1
  - V_{21}è il caso di test che attiva P_2

Test di Integrazione

Obiettivi

⇨Verificare che le parti del sistema siano state collegate correttamente.

Test Strutturale di Integrazione

⇨La esecuzione del est di integrazione si basa sull'architettura del sistema

⇨Si devono progettare i casi di test necessari per provare il comportamento di tutte le interfacce tra componenti previste dal sistema

⇨Per ogni interfaccia si definiscono i casi di tests con le tecniche black box viste prima, basandosi sulle specifiche delle stesse interfacce

Test di Sistema

Obiettivo

Verificare il comportamento del sistema software nella sua interezza rispetto ai vincoli ed ai requisiti espressi dal cliente.

Tipi di Tests

Può essere suddiviso nelle seguenti attività di test:

1. Funzionale [normalmente attraverso le tecniche Black Box]
2. Prestazione [ad hoc]
3. Configurazione [ad hoc]
4. Recuperabilità [ad hoc]
5. Sicurezza [ad hoc]
6. Stress [ad hoc]

Test di Accettazione

Generalità

- Ha le stesse caratteristiche del test di sistema si differenzia da questo per due caratteristiche
  - E' eseguito nelle condizioni e negli ambienti di esercizio
  - E' eseguito, in genere in contradditorio, con l'utilizzatore od un suo rappresentante
- Qualche volta è chiamato collaudo

Test di Regressione

Generalità

- Si usa durante l'esercizio di un'applicazione, dopo ogni cambiamento effettuato per manutenzione
- E' una ripetizione di casi di test, selezionati tra quelli già eseguiti, per verificare che le modifiche non hanno causato effetti collaterali non desiderati sul software già funzionante.