

Corso di Laurea in Informatica - Ingegneria del Software 2 1



10. Verifica e validazione: prove dinamiche

Docente: Tullio Vardanega
tullio.vardanega@math.unipd.it

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 2



Definizione

- ◆ Il test software consiste nella verifica dinamica del comportamento di un programma
 - ◆ Su un insieme finito di casi di test
 - ◆ Selezionati nell'ambito del dominio delle esecuzioni possibili, che è normalmente *infinito*
 - ◆ Ciascun caso di test specifica i valori di ingresso e lo stato del sistema all'inizio del test
 - ◆ Ciascun caso di test deve produrre un esito decidibile
 - ◆ *Il problema dell'oracolo*
 - ◆ Verificati rispetto ad un comportamento atteso

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 3



Caratterizzazione

- ◆ È parte irrinunciabile del processo di verifica
- ◆ Produce una misura della qualità del sistema
 - ◆ Come sottoprodotto, ne aumenta il valore di qualità identificando e rimuovendo difetti
- ◆ Il suo inizio non deve essere differito al termine della fase di codifica
- ◆ Le sue esigenze costituiscono un fattore di influenza fondamentale per la costruzione del sistema

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 4



Fattori da bilanciare

- ◆ La definizione della strategia di test comporta sempre un bilanciamento tra
 - ◆ La quantità minima di casi di test sufficienti a fornire certezza adeguate sulla qualità del prodotto
 - ◆ Fattore governato da criteri tecnici
 - ◆ Il livello massimo di sforzo, tempo e risorse utilizzabile per il completamento della verifica
 - ◆ Fattore governato da criteri gestionali

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 5



Criteri guida - 1

- ◆ Oggetto del test
 - ◆ Il sistema nel suo complesso
 - ◆ Parti di esso, in relazione funzionale, d'uso, di comportamento, di struttura
 - ◆ Unità singole
- ◆ Obiettivo del test
 - ◆ Specificato per ogni caso di test
 - ◆ In termini il più possibile precisi e quantitativi
 - ◆ Varia al variare dell'oggetto del test
 - ◆ Richiede risposte alle domande "quanti" e "quali" test

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 6



Classificazione delle problematiche

| | | | | |
|--------------------------------|-----------|--------------------------------|--------------|-----------------------|
| Test software | | | | |
| Concetti e definizioni di base | Livelli | Tecniche | Valutazione | Gestione del processo |
| Terminologia | Oggetto | Intuizione ed esperienza | Dell'oggetto | Vincoli di progetto |
| Fondamenti teorici | Obiettivo | Secondo specifica | Dei test | Attività di test |
| Relazione con altre attività | | Sulla base del codice | | |
| | | Sulla base dei difetti | | |
| | | Secondo l'uso | | |
| | | Secondo il tipo d'applicazione | | |
| | | Strutturale (white-box) | | |
| | | Funzionale (black-box) | | |
| | | Per commistione | | |

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 7

Dentro la classificazione – 1

- ◆ La classificazione appena vista sottende numerose aree di studio e di conoscenza
- ◆ Alcuni esempi
 - ◆ Terminologia
 - ◆ Difetti (*fault*), errori (*error*) e guasti (*failure*)
 - ◆ Fondamenti teorici
 - ◆ Decidibilità, testabilità, criteri di test
 - ◆ Oggetto del test
 - ◆ Unità, aggregati, sistema completo

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 8

Dentro la classificazione – 2

- ◆ Alcuni esempi (segue)
 - ◆ Obiettivo del test
 - ◆ Accettazione, qualifica, conformità, installazione, regressione, prestazione, ...
 - ◆ Vincoli di progetto
 - ◆ Definizione del processo, definizione dei prodotti, personale di test (interno o indipendente), stima e controllo dei costi, criteri di terminazione
 - ◆ Attività di test
 - ◆ Pianificazione, specifica dei casi di test, sviluppo dell'ambiente di test, esecuzione, valutazione, trattamento dei problemi (anomalie, discrepanze)

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 9

Criteri guida - 2

- ◆ Una visione riduttiva, ma espressiva, del test
 - ◆ “Il processo di eseguire un programma con l'intento di trovarvi difetti”
 - The Art of Software Testing, G.J.Myers, Wiley-Interscience, 1979
- ◆ Ben rappresenta lo spirito critico e l'atteggiamento scettico alla base di una strategia efficace di test
- ◆ La testabilità del software va assicurata durante lo sviluppo, non a valle della codifica
 - ◆ Disegno architetturale e di dettaglio vanno raffinati finché assicurino testabilità
 - ◆ La complessità è nemica della testabilità

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 10

Attività di test

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 11

Esecuzione delle attività di test

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 12

Test di unità - 1

- ◆ Una unità software è composta da uno o più moduli
 - ◆ Modulo come componente elementare di disegno di dettaglio
- ◆ Usa il disegno di dettaglio come termine di riferimento
- ◆ Completa quando le unità testate sono quelle definite a livello di disegno architetturale (DA)
- ◆ Circa 2/3 dei difetti identificati mediante verifica dinamica vengono rilevati dal test di unità
 - ◆ La meta dei quali viene identificata mediante test strutturale (*white-box*)

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 13

Test di unità - 2

- ◆ Il piano di test di unità viene stabilito al termine della fase di disegno di dettaglio
- ◆ Per ogni test si definiscono l'oggetto, la strategia, le risorse necessarie ed il calendario di esecuzione
- ◆ La quantità minima di test necessario è quella sufficiente ad eseguire almeno una volta tutte le *linee di comando* di ciascun modulo compreso nell'unità (statement coverage)
- ◆ Può essere necessario verificare *anche* ciascun *ramo della logica* del flusso di controllo (branch coverage)

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 14

Regole di integrazione di moduli

- ◆ Assembla moduli in modo incrementale
 - ◆ Ciò fa sì che difetti rilevati nel corso di un test di unità siano più probabilmente da attribuirsi al modulo integrato per ultimo
- ◆ Assembla moduli produttori prima dei moduli consumatori
 - ◆ La verifica dei prima fornisce ai secondi flusso di controllo e flusso dei dati corretti
- ◆ Assembla in modo che ogni passo di integrazione sia reversibile
 - ◆ Ciò consente di retrocedere verso uno stato noto e sicuro

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 15

Esempio - 1

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 16

Test di unità - 3

- ◆ Test strutturale (*white-box*)
 - ◆ Tende a verificare la logica interna del codice dell'unità
 - ◆ Ciascun test deve essere progettato per attivare ciascun cammino di esecuzione all'interno del modulo
 - ◆ Ciascun insieme di dati di ingresso che attivano un percorso costituiscono un caso di test
 - ◆ L'uso di debugger ne agevola l'esecuzione ma non esonera dalla progettazione dei casi di test

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 17

Test di unità - 4

- ◆ Test funzionale (*black-box*)
 - ◆ Rimedia alla carenza del test strutturale, incapace di accertare la completezza della logica interna dell'unità
 - ◆ Fa riferimento alla specifica dell'unità ed utilizza dati di ingresso capaci di provocare l'esito atteso
 - ◆ Ciascun insieme di dati di ingresso usati per ottenere un dato comportamento costituisce un caso di test
 - ◆ Riduce l'infinità di valori in ingresso mediante classi di equivalenza
 - ◆ Dati di ingresso nella medesima classe produrranno il medesimo comportamento

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 18

Classi di equivalenza

- 3 classi di equivalenza
 - Valori nominali interni al dominio
 - Valori nominali di limite
 - Valori illegali

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 19



Test di integrazione - 1

- ◆ Si applica alle componenti del disegno architetturale
 - ◆ L'integrazione di tali componenti costituisce il sistema nella sua interezza
- ◆ Segue prevalentemente una logica di integrazione funzionale
 - ◆ Seleziona le funzioni da integrare
 - ◆ Identifica le componenti che eseguono le funzioni
 - ◆ Ordina le componenti per numero di dipendenze
 - ◆ Quelle con meno dipendenze hanno precedenza
 - ◆ Rappresenta quelle successive ma necessarie con "driver di test"
 - ◆ Eseguce l'integrazione a partire da quelle con precedenza

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

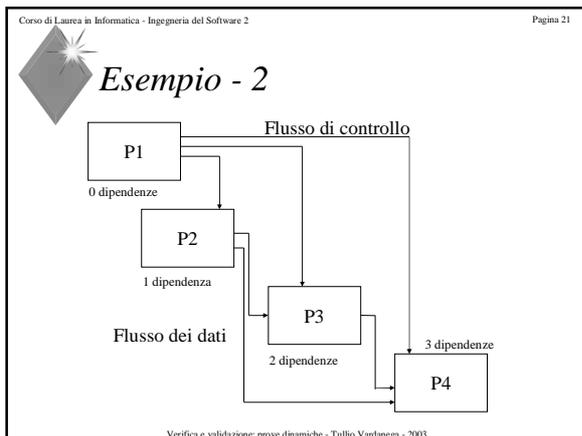
Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 20



Test di integrazione - 2

- ◆ I problemi rilevati durante i test di integrazione
 - ◆ Sono generalmente poco numerosi a valle di un buon test di unità
 - ◆ Sottendono difetti di identificazione e correzione assai laboriose
 - ◆ Difetti architetturali
- ◆ La quantità di test richiesti risponde alla esigenza di
 - ◆ Accertare che tutti i dati scambiati attraverso ciascun interfaccia aderiscano alla loro specifica
 - ◆ Accertare che tutti i flussi di controllo previsti in specifica siano stati effettivamente realizzati

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003



Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 22



Test di sistema

- ◆ Verifica il comportamento dinamico del sistema completo rispetto ai requisiti software
- ◆ Ha inizio con il completamento del test di integrazione
- ◆ È inerentemente funzionale (*black-box*)
 - ◆ Non dovrebbe richiedere conoscenza della logica interna del software

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003

Corso di Laurea in Informatica - Ingegneria del Software 2 Pagina 23



Altri tipi di test

- ◆ Test di regressione
 - ◆ Ripetizione *selettiva* di test di livello sistema o integrazione
 - ◆ Per accertare che modifiche intervenute non causino comportamenti erronei del sistema
 - ◆ È oneroso al punto da richiedere ogni sforzo possibile per evitarne la necessità
- ◆ Test di accettazione
 - ◆ Accerta il soddisfacimento dei requisiti utente

Verifica e validazione: prove dinamiche - Tullio Vardanega - 2003