



13. Verifica e Validazione del Software

Andrea Polini

Ingegneria del Software
Corso di Laurea in Informatica

Verifica e Validazione - generalità

Verifica: stiamo sviluppando il sistema correttamente?

Validazione: stiamo sviluppando il prototipo corretto?

Approaches to V&V

Approcci statici: ispezione del software, verifica formale, analisi statica dei sorgenti

Approcci dinamici: testing, monitoring

Debugging?

Ispezione del software

Può sembrare inefficace ma molte esperienze reali hanno dimostrato il contrario. In molti casi ispezione si è rivelata più efficace del testing

Vantaggi sul testing:

- Non permette mascheramento di errori
- possibilità di applicazione a parti del software non ancora eseguibili
- possibilità di verificare aderenza agli standard, ed a stile di codifica

Processo di ispezione

Come tutte le attività si svolge strutturando un processo e degli attori che svolgono specifiche attività

Attori:

- Autore
- Ispettore
- Lettore
- Scriba
- Moderatore
- Responsabile del processo

Il Processo di ispezione

- Pianificazione
- Overview
- Preparazione Individuale
- Riunione di ispezione
- Rielaborazione
- Follow-up

Uso di checklist

- Errori sui dati: inizializzazione delle variabili, uso di nomi per le costanti, indici di vettori, buffer overflow
- Errori nel controllo: specifica delle condizioni, terminazione dei cicli, uso delle parentesi, `case` comprende tutte le possibilità, uso dei comandi di `break`
- Errori negli input/output: uso di tutte le variabili, assegnamento di valori alle variabili di uscita, comportamento in caso di ingressi imprevisti
- Errori nelle interfacce: verifica del corretto uso dei parametri, ordine dei parametri
- Errori nella gestione della memoria: uso e modifica di una struttura con collegamenti, allocazione e deallocazione dinamica dello spazio
- Errori nella gestione delle eccezioni: possibili sorgenti di errore sono considerate

Considerazioni quantitative

- 500 linee all'ora durante overview
- 125 righe di codice all'ora durante preparazione individuale
- tra i 90 e le 125 righe durante il meeting

Verifica formale

Applicabili nel caso di uso di formalismi di specifica formali.

Correttezza può essere derivata con argomenti matematici

Comunque:

- specifica formali potrebbe non rispecchiare i requisiti reali
- le prove potrebbero essere errate
- prove potrebbero fare assunzioni non valide nell'ambiente finale d'uso

Il testing del software prevede l'esecuzione di alcuni “esperimenti” al fine di poter acquisire sufficiente fiducia sul suo funzionamento. Testing riguarda tipicamente **proprietà funzionali** ma può riguardare anche **caratteristiche extra-funzionali**.

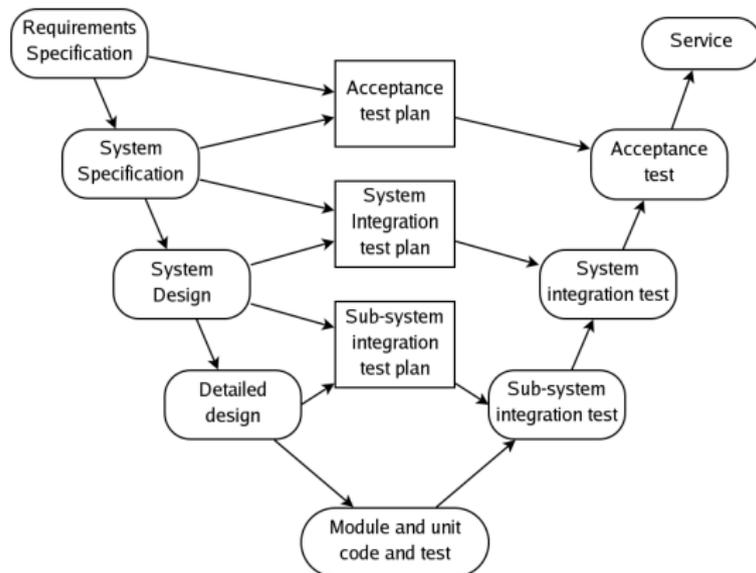
- Quali esperimenti devono essere eseguiti?
- Quando posso dire che gli esperimenti sono sufficienti?
- Come posso decidere riguardo al risultato?
- Ho bisogno di riprodurre particolari ambienti?

Un po' di tassonomia

- Test Case
- Test suite
- Driver
- Sistem Under Test (SUT)
- Stubs
- Testbed
- Controllo/Osservo
- Oracolo
- Workload
- ...

Fasi del testing e V model

- Unit testing
- Integration testing
- System Testing
- Acceptance Testing



Criteri di scelta

- Dati - Partition testing, Boundary Condition
- Controllo - Path, Branch, Statement

Diverse Tipologie

- Robustness Testing
- Regression Testing