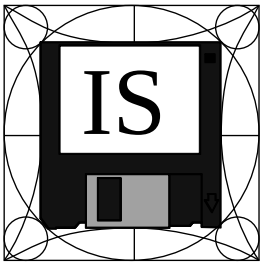
Verifica e validazione: introduzione



Corso di Ingegneria del Software

V. Ambriola, G.A. Cignoni,
C. Montangero, L. Semini

Con aggiornamenti di: T. Vardanega (UniPD)

Contenuti

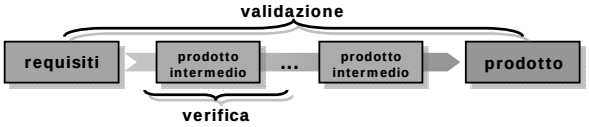
- Concetti e terminologia
- Verifica, validazione, integrazione e collaudo
- Verifica statica
- *Inspection* e *walkthrough*
- Verifica e validazione di qualità

Verifica e validazione – 1

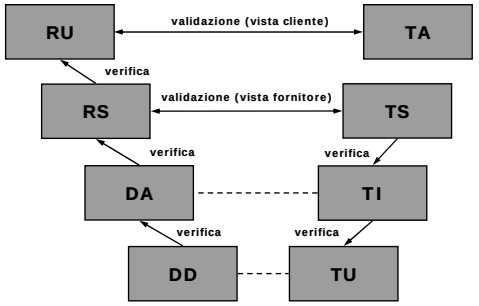
- Software verification
  - Provides objective evidence that the design outputs of a particular phase of the software development life cycle meet all of the specified requirements for that phase
  - Software verification looks for consistency, completeness, and correctness of the software and its supporting documentation, as it is being developed, and provides support for a subsequent conclusion that software is validated
- Software validation
  - Confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled

Verifica e validazione – 2

- Verifica
  - Accertare che l'esecuzione dei processi attivati non abbia introdotto difetti nel prodotto
- Validazione
  - Accertare che il prodotto realizzato sia quello atteso



Verifica e validazione – 2



Forme di analisi

- Analisi statica
  - Non comporta esecuzione
  - Usata prevalentemente come forma di verifica
    - Ma anche per la validazione di requisiti non funzionali
- Analisi dinamica
  - Comporta esecuzione
  - Usata sia per verifica che per validazione
  - Effettuata tramite prove (dove prova = *test*)

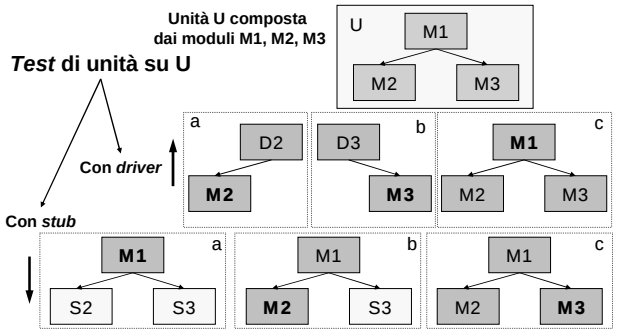
Analisi dinamica: ambiente di prova

- **Ripetibilità**
  - **Ambiente (HW, stato iniziale, …)**
  - **Specifica (ingressi richiesti, comportamenti attesi)**
  - **Procedure (esecuzione, analisi dei risultati)**
- **Strumenti**
  - ***Driver*** **componente attiva fittizia per pilotare una parte**
  - ***Stub*** **componente passiva fittizia per simulare una parte**
- **Registrazione dei dati per analisi dei risultati**

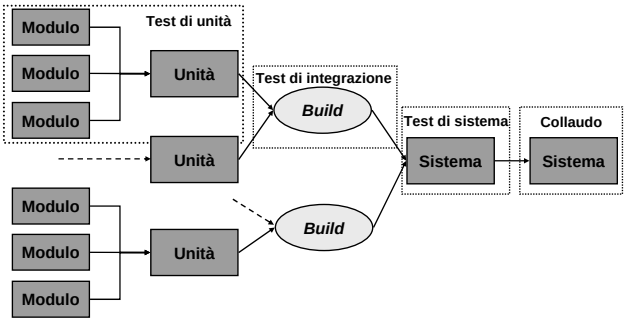
Analisi dinamica: definizioni

- **Unità**
  - **La più piccola quantità di SW che è utile verificare singolarmente**
    - La realizzazione di una singola unità è affidata a un singolo programmatore
  - **La definizione concreta dipende dal linguaggio di programmazione utilizzato ma va sempre intesa in senso architetturale**
    - Non linee di codice ma vere entità sintattiche (procedura, classe, *package*, …)
- **Modulo**
  - **Una parte dell'unità**
    - Un metodo rispetto alla classe
    - Una classe rispetto al *package* (in Java)
- **Componente o sottosistema**
  - **L'integrazione di più unità**

Stub e *driver*



Tipi di *test*



Test di unità

- **Approcci possibili**
  - **Analisi statica**
    - Dal controllo di *routine* (*desk-check)* all'ispezione in relazione alla complessità
  - **Analisi dinamica**
    - Con realizzazione di *driver* e/o *stub* in relazione all'architettura dell'unità
- **Responsabilità**
  - **Del programmatore autore dell'unità per unità semplici**
  - **Con l'intervento di una controparte (meglio se un automa) altrimenti**
- **Quali problemi rileva**
  - **Permette di verificare la correttezza del codice "*as implemented*"**
    - *White box* invece che *black box* (che serve per la verifica di soddisfazione dei requisiti)

Test di integrazione

- **Costruzione e verifica incrementale del sistema**
  - **Componenti realizzati in parallelo**
  - **Componenti verificati indipendentemente**
  - **In condizioni ottimali l'integrazione è priva di problemi**
- **Quali problemi rileva**
  - **Errori residui nella realizzazione dei componenti**
  - **Modifica delle interfacce o cambiamenti nei requisiti**
  - **Riuso di componenti dal comportamento oscuro o inadatto**
  - **Integrazione con altre applicazioni non ben conosciute**

Test di sistema e collaudo

- **Validazione**
  - **Test di sistema come attività interna del fornitore**
    - Per verifica di copertura dei requisiti SW
  - **Collaudo come attività controllata dal committente**
    - Su casi di prova definiti nel contratto
- **Valore contrattuale**
  - **Il collaudo è un'attività formale**
  - **Al collaudo segue il rilascio del sistema**
  - **Conclusione della commessa**
    - A meno di garanzia e servizi di manutenzione

Test di regressione

- **L'insieme di *test* necessari ad accertare che la modifica di una parte P di S non causi errori in P o nelle altri parti di S che dipendono da P**
  - **Consiste nella ripetizione di *test* già previsti ed effettuati per ogni parte coinvolta**
- **Modifiche effettuate per aggiunta, correzione o rimozione non devono pregiudicare le funzionalità già verificate**
  - **Il rischio aumenta all'aumentare dell'accoppiamento e al diminuire dell'incapsulazione**

Forme di analisi statica

- **Effettuabili senza esecuzione**
- **Applicano a ogni prodotto di processo e non solo al codice**
- **Metodi di lettura (*desk check*)**
  - **Impiegati solo per prodotti semplici**
- **Metodi formali**
  - **Basati sulla prova assistita di proprietà**
    - La cui dimostrazione dinamica può essere eccessivamente onerosa
  - **Verifica di equivalenza o generazione automatica**

Metodi di lettura

- ***Inspection* e *Walkthrough***
- **Metodi pratici**
  - **Basati sulla lettura della documentazione sul prodotto**
  - **Di efficacia dipendente dall'esperienza dei verificatori**
    - Nell'organizzare le attività di verifica
    - Nel documentare le attività svolte e i risultati ottenuti
- **Modalità relativamente complementari**

Inspection

- **Obiettivi**
  - **Rivelare la presenza di difetti**
  - **Eseguire una lettura mirata [del codice]**
- **Agenti**
  - **Verificatori distinti e separati dai programmatori**
- **Strategia**
  - **Focalizzare la ricerca su presupposti**
    - *Error guessing*

Attività di *inspection*

- **Fase 1: pianificazione**
- **Fase 2: definizione della lista di controllo**
- **Fase 3: lettura [del codice]**
- **Fase 4: correzione dei difetti**
- **In ogni fase**
  - **Documentazione come rapporto delle attività svolte**

Verifica e validazione

Walkthrough

- **Obiettivo**
  - **Rivelare la presenza di difetti**
  - **Eseguire una lettura critica [del codice]**
    - A largo spettro
    - Senza l'assunzione di presupposti
- **Agenti**
  - **Gruppi misti ispettori/sviluppatori ma con ruoli ben distinti**
- **Strategia (per il codice)**
  - **Percorrerlo simulandone possibili esecuzioni**

Verifica e validazione

Attività di *walkthrough*

- **Fase 1: pianificazione**
- **Fase 2: lettura**
- **Fase 3: discussione**
- **Fase 4: correzione dei difetti**
- **In ogni fase**
  - **Documentazione come rapporto delle attività svolte**

Verifica e validazione

Inspection contro *walkthrough*

- **Affinità**
  - **Controlli basati su *desk check***
  - **Programmatori e verificatori su fronti opposti**
  - **Documentazione formale**
- **Differenze**
  - ***Inspection* basato su (errori) presupposti**
  - ***Walkthrough* richiede maggiore attenzione**
  - ***Walkthrough* più collaborativo**
  - ***Inspection* più rapido**

Verifica e validazione

Verifica e validazione di qualità

- **Evidenza di qualità**
  - **A fronte di una metrica e di livelli definiti**
  - **Verificare (validare) per dare evidenza**
  - **Controllo (interno) e accertamento (esterno)**
- **ISO/IEC 9126 come riferimento**
  - **Quali strumenti per quali caratteristiche?**
  - **La qualità in uso è valutata a posteriori**

Verifica e validazione

Funzionalità

- **Dimostrabile tramite prove**
- **Analisi statica come attività preliminare**
- **Liste di controllo rispetto ai requisiti**
  - **Completezza ed economicità**
    - Tutte le funzionalità richieste per tutti e soli i componenti necessari
  - **Interoperabilità**
    - Accertata la compatibilità tra le soluzioni realizzative adottate
  - **Sicurezza del prodotto e dei suoi componenti**
  - **Adesione alle norme e alle prescrizioni**
- **Valutazione di accuratezza**

Verifica e validazione

Affidabilità

- **Dimostrabile tramite analisi statiche e dinamiche**
- **Verifica statica come attività preliminare**
- **Liste di controllo rispetto ai requisiti**
  - **Robustezza**
  - **Capacità di ripristino e recupero da errori**
  - **Adesione alle norme e alle prescrizioni**
- **Valutazione di maturità**

Verifica e validazione

Usabilità

- Le prove sono imprescindibili
- Analisi statica come attività complementare
- Liste di controllo rispetto ai manuali d'uso
  - Comprensibilità
  - Apprendibilità
  - Adesione a norme e prescrizioni
- Questionari sottomessi agli utenti
  - Facilità e piacevolezza d'uso

Verifica e validazione

Efficienza

- Le prove sono necessarie
- Liste di controllo rispetto alle specifiche norme di codifica
  - Nel tempo
  - Nello spazio
- Margini di miglioramento e confidenza
  - L'efficienza provata fornisce confidenza nel prodotto
  - L'analisi statica fornisce indicazioni specifiche sui margini di miglioramento prestazionale

Verifica e validazione

Manutenibilità

- Analisi statica come strumento ideale
- Liste di controllo rispetto alle norme di codifica
  - Analizzabilità
  - Modificabilità
- Liste di controllo rispetto alle prove
  - Ripetibilità
  - Verificabilità
- Prove di stabilità

Verifica e validazione

Portabilità

- Analisi statica come strumento ideale
- Liste di controllo rispetto alle specifiche norme di codifica
  - Adattabilità
- Prove come strumento complementare
  - Facilità d'installazione
  - Compatibilità ambientale
  - Facilità di sostituzione

Verifica e validazione

Riepilogo

- Concetti e terminologia
- Verifica, validazione, integrazione e collaudo
- Verifica statica
- *Inspection* e *walkthrough*
- Verifica e validazione di qualità

Verifica e validazione

Riferimenti

- Standard for Software Component Testing, British Computer Society SIGIST, 1997
- M.E. Fagan, Advances in Software Inspection, *IEEE Transaction on Software Engineering*, luglio 1986
- G.A. Cignoni, P. De Risi, "Il test e la qualità del software", Il Sole 24 Ore, 1998