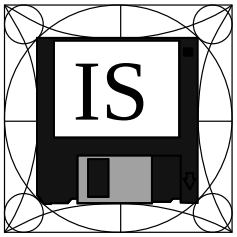
Verifica e validazione



IS 2001-5
Corso di Ingegneria del Software

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

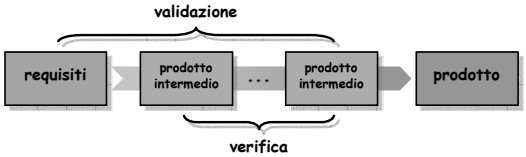
Con aggiornamenti di: T. Vardanega (UniPD)

Contenuti

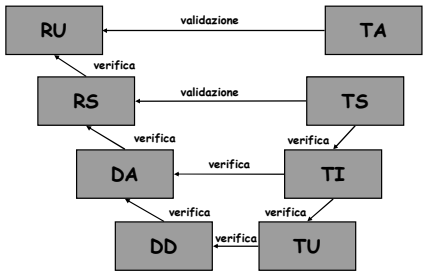
- Concetti e terminologia
- Verifica, validazione, integrazione e collaudo
- Verifica statica
- *Inspection* e *walkthrough*
- Verifica e validazione di qualità

Verifica e validazione

- Attività necessarie
  - Accertare che il processo non abbia introdotto difetti nel prodotto
  - Accertare che il prodotto realizzato sia quello atteso



Verifica e validazione

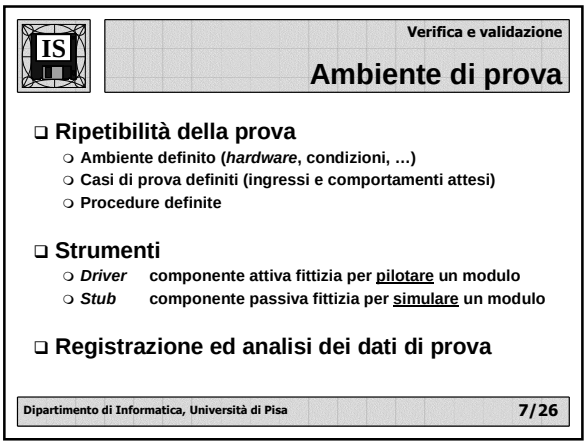


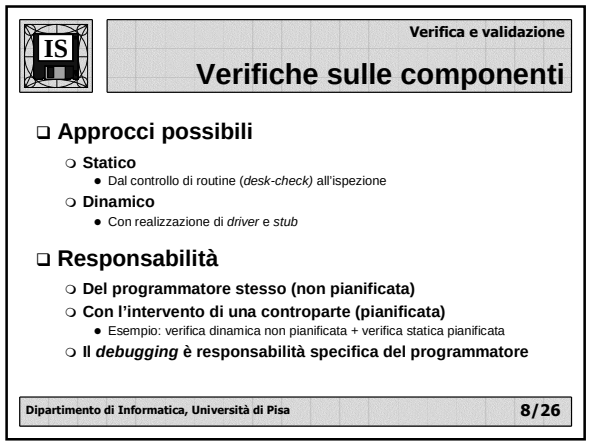
Guasti, difetti, malfunzionamenti

- Guasto → *Fault*
  - Atto od omissione, possibile causa (anche umana) di comportamento fuori norma — programmatore distratto
  - Può non avere effetto
- Errore o Difetto → *Error*
  - Stato erroneo di sistema introdotto da un guasto — dato erroneo in una procedura
  - Può restare latente, va eliminato
- Malfunzionamento → *Failure*
  - Effetto di un errore, risulta in deviazione da specifica — emissione di un valore sbagliato
  - Può causare danni rilevanti

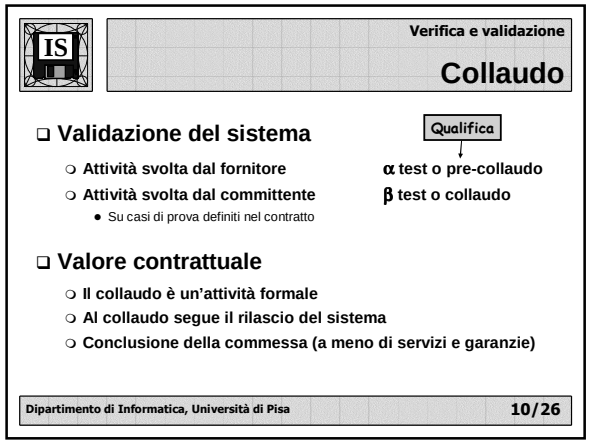
Verifiche statiche e dinamiche

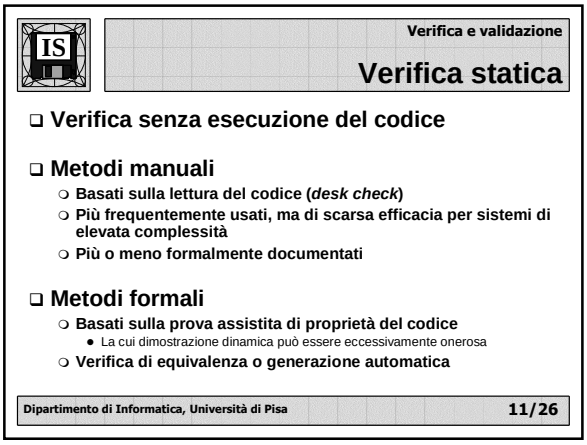
- Verifiche statiche
  - Non comportano esecuzione del codice
  - Usate prevalentemente per attività di verifica
  - Effettuate sulle componenti (non sul sistema)
- Verifiche dinamiche (prove ↔ *test*)
  - Comportano esecuzione del codice
  - Usate per attività di verifica e/o di validazione
  - Effettuate sulle componenti e/o sul sistema


IS
Verifica e validazione
Ambiente di prova
Ripetibilità della prova
Ambiente definito (*hardware*, condizioni, ...)
Casi di prova definiti (ingressi e comportamenti attesi)
Procedure definite
Strumenti
Driver componente attiva fittizia per pilotare un modulo
Stub componente passiva fittizia per simulare un modulo
Registrazione ed analisi dei dati di prova
Dipartimento di Informatica, Università di Pisa
7/26

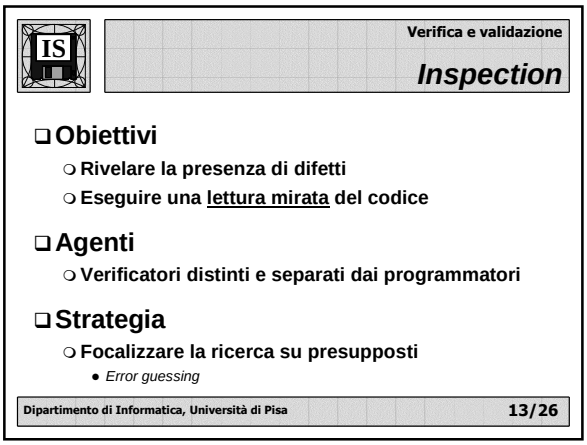

IS
Verifica e validazione
Verifiche sulle componenti
Approcci possibili
Statico
Dal controllo di routine (*desk-check)* all'ispezione
Dinamico
Con realizzazione di *driver* e *stub*
Responsabilità
Del programmatore stesso (non pianificata)
Con l'intervento di una controparte (pianificata)
Esempio: verifica dinamica non pianificata + verifica statica pianificata
Il *debugging* è responsabilità specifica del programmatore
Dipartimento di Informatica, Università di Pisa
8/26


IS
Verifica e validazione
Integrazione
Costruzione e verifica del sistema
Componenti realizzate in parallelo
Componenti realizzate e verificate indipendentemente
In condizioni ottime, l'integrazione è priva di problemi
Problemi
Errori nella realizzazione delle componenti
Modifica delle interfacce o cambiamenti nei requisiti
Riuso di componenti dal comportamento oscuro o inadatto
Integrazione con altre applicazioni non ben conosciute
Dipartimento di Informatica, Università di Pisa
9/26


IS
Verifica e validazione
Collaudo
Validazione del sistema
Attività svolta dal fornitore
Attività svolta dal committente
Su casi di prova definiti nel contratto
Qualifica
α test o pre-collaudo
β test o collaudo
Valore contrattuale
Il collaudo è un'attività formale
Al collaudo segue il rilascio del sistema
Conclusione della commessa (a meno di servizi e garanzie)
Dipartimento di Informatica, Università di Pisa
10/26


IS
Verifica e validazione
Verifica statica
Verifica senza esecuzione del codice
Metodi manuali
Basati sulla lettura del codice (*desk check*)
Più frequentemente usati, ma di scarsa efficacia per sistemi di elevata complessità
Più o meno formalmente documentati
Metodi formali
Basati sulla prova assistita di proprietà del codice
La cui dimostrazione dinamica può essere eccessivamente onerosa
Verifica di equivalenza o generazione automatica
Dipartimento di Informatica, Università di Pisa
11/26

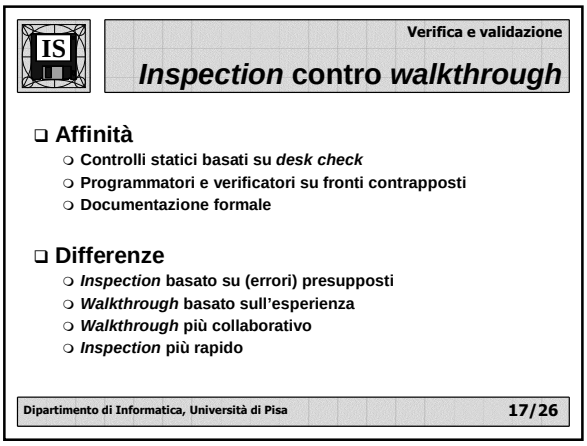

IS
Verifica e validazione
Metodi di lettura del codice
Inspection e *Walkthrough*
Metodi pratici
Basati sulla lettura del codice
Efficacia dipendente dall'esperienza dei verificatori
Per organizzare le attività di verifica
Per documentare l'attività ed i suoi risultati
Tra loro complementari
Dipartimento di Informatica, Università di Pisa
12/26

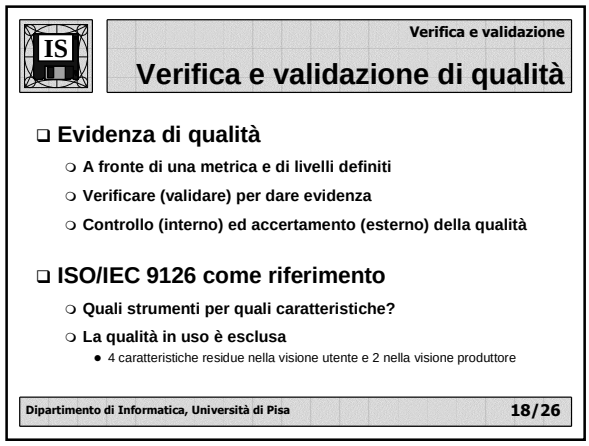

IS
Verifica e validazione
Inspection
Obiettivi
Rivelare la presenza di difetti
Eseguire una lettura mirata del codice
Agenti
Verificatori distinti e separati dai programmatori
Strategia
Focalizzare la ricerca su presupposti
Error guessing
Dipartimento di Informatica, Università di Pisa
13/26


IS
Verifica e validazione
Attività di ispezione
Fase 1: pianificazione
Fase 2: definizione della lista di controllo
Fase 3: lettura del codice
Fase 4: correzione dei difetti
Documentazione (rapporto delle attività)
Dipartimento di Informatica, Università di Pisa
14/26


IS
Verifica e validazione
Walkthrough
Obiettivo
Rivelare la presenza di difetti
Eseguire una lettura critica del codice
A largo spettro
Agenti
Gruppi misti ispettori/sviluppatori, ma con ruoli ben distinti
Strategia
Percorrere il codice simulandone possibili esecuzioni
Dipartimento di Informatica, Università di Pisa
15/26


IS
Verifica e validazione
Attività di walkthrough
Fase 1: pianificazione
Fase 2: lettura del codice
Fase 3: discussione
Fase 4: correzione dei difetti
Documentazione (rapporto delle attività)
Dipartimento di Informatica, Università di Pisa
16/26


IS
Verifica e validazione
Inspection contro walkthrough
Affinità
Controlli statici basati su desk check
Programmatori e verificatori su fronti contrapposti
Documentazione formale
Differenze
Inspection basato su (errori) presupposti
Walkthrough basato sull'esperienza
Walkthrough più collaborativo
Inspection più rapido
Dipartimento di Informatica, Università di Pisa
17/26


IS
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) ed accertamento (esterno) della qualità
ISO/IEC 9126 come riferimento
Quali strumenti per quali caratteristiche?
La qualità in uso è esclusa
4 caratteristiche residue nella visione utente e 2 nella visione produttore
Dipartimento di Informatica, Università di Pisa
18/26

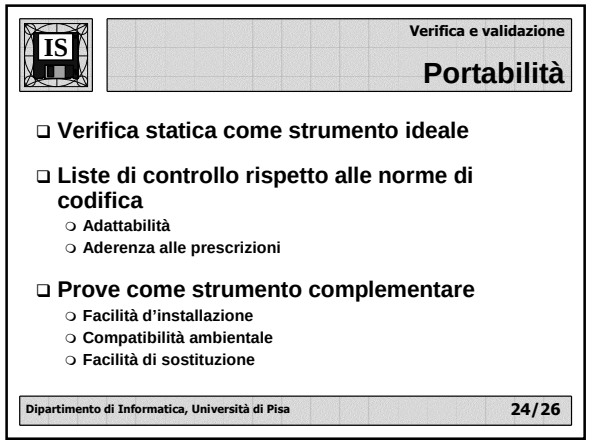

IS
Verifica e validazione
Funzionalità
Dimostrabile tramite prove
Verifica statica come attività preliminare
Liste di controllo rispetto ai requisiti
Completezza ed economicità
Tutte e sole le funzionalità richieste
Interoperabilità
Compatibilità tra le soluzioni adottate
Sicurezza (delle soluzioni adottate)
Adesione alle norme ed alle prescrizioni
Prove per accuratezza
Dipartimento di Informatica, Università di Pisa
19/26

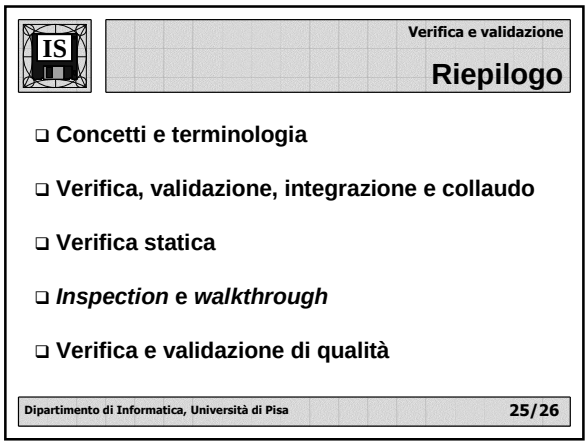

IS
Verifica e validazione
Affidabilità
Dimostrabile tramite analisi e prove
Verifica statica come attività preliminare
Liste di controllo rispetto ai requisiti
Robustezza
Tolleranza ai guasti (fault tolerance)
Capacità di ripristino e recupero da errori
Aderenza alle prescrizioni
Prove per maturità
Dipartimento di Informatica, Università di Pisa
20/26

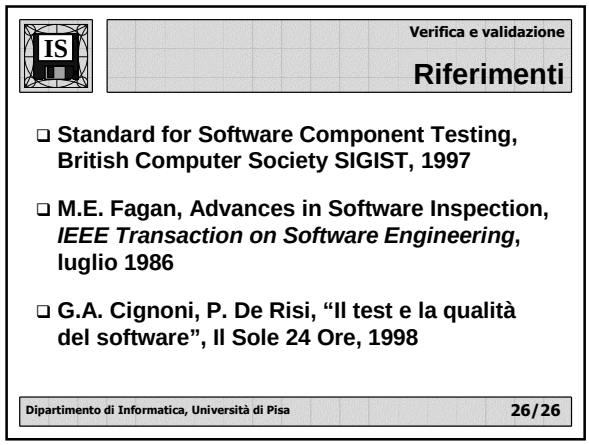

IS
Verifica e validazione
Usabilità
Le prove sono imprescindibili
Verifica statica come attività complementare
Liste di controllo rispetto ai manuali d'uso
Comprensibilità
Apprendibilità
Aderenza alle prescrizioni
Questionari all'utenza (a seguito di prove)
Facilità d'uso
Piacevolezza d'uso
Dipartimento di Informatica, Università di Pisa
21/26


IS
Verifica e validazione
Efficienza
Prove necessarie con la tecnologia attuale
Verifica statica come attività preliminare
Liste di controllo rispetto ai criteri realizzativi richiesti
Efficienza temporale
algoritmica
Efficienza spaziale
Uso delle risorse
Miglioramento e confidenza
L'efficienza provata fornisce confidenza
La verifica statica non da confidenza (attualmente), ma indicazioni importanti per migliorare il codice
Dipartimento di Informatica, Università di Pisa
22/26


IS
Verifica e validazione
Manutenibilità
Verifica statica come strumento ideale
Liste di controllo rispetto alle norme di codifica
Analizzabilità
Modificabilità
Aderenza alle prescrizioni
Liste di controllo rispetto all'insieme delle prove
Ripetibilità
Verificabilità
Prove per la stabilità
Dipartimento di Informatica, Università di Pisa
23/26


IS
Verifica e validazione
Portabilità
Verifica statica come strumento ideale
Liste di controllo rispetto alle norme di codifica
Adattabilità
Aderenza alle prescrizioni
Prove come strumento complementare
Facilità d'installazione
Compatibilità ambientale
Facilità di sostituzione
Dipartimento di Informatica, Università di Pisa
24/26


IS
Verifica e validazione
Riepilogo
Concetti e terminologia
Verifica, validazione, integrazione e collaudo
Verifica statica
Inspection e *walkthrough*
Verifica e validazione di qualità
Dipartimento di Informatica, Università di Pisa
25/26


IS
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
Dipartimento di Informatica, Università di Pisa
26/26