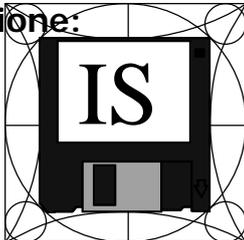


 Verifica e validazione:
introduzione

IS 2001-9
Corso di Ingegneria del Software
V. Ambriola, G.A. Cignoni,
C. Montangero, L. Semini
Con aggiornamenti di: T. Vardanega (UniPD)



Dipartimento di Informatica, Università di Pisa 1/26

 Verifica e validazione
Contenuti

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

Dipartimento di Informatica, Università di Pisa 2/26

 Verifica e validazione
Verifica e validazione

- **Verifica**
 - Accertare che l'esecuzione del processo non abbia introdotto difetti nel prodotto
 - Anche per singole attività di notevole ampiezza, complessità e criticità
- **Validazione**
 - Accertare che il prodotto realizzato sia davvero quello atteso

validazione

verifica

requisiti → prodotto intermedio → ... → prodotto intermedio → prodotto

Dipartimento di Informatica, Università di Pisa 3/26

 Verifica e validazione
Verifica e validazione

Diagramma delle attività di verifica e validazione:

```
graph TD
    RU[RU] -- "validazione (vista cliente)" --> TA[TA]
    RS[RS] -- "validazione (vista fornitore)" --> TS[TS]
    DA[DA] -- "validazione" --> TI[TI]
    DD[DD] -- "validazione" --> TU[TU]
    RS -- "verifica" --> RU
    TS -- "verifica" --> TA
    TI -- "verifica" --> DA
    TU -- "verifica" --> DD
```

Dipartimento di Informatica, Università di Pisa 4/26

 Verifica e validazione
Analisi statica e dinamica

- **Analisi statica**
 - Non comporta esecuzione del codice
 - Usata prevalentemente come forma di verifica
 - Anche per la validazione di requisiti non funzionali
 - Effettuata su componenti
 - Più raramente sul sistema intero (per motivi di complessità)
- **Analisi dinamica (prove = test)**
 - Comporta esecuzione del codice
 - Usata sia per verifica che per validazione
 - Effettuata su componenti ma anche sul sistema

Dipartimento di Informatica, Università di Pisa 5/26

 Verifica e validazione
Ambiente di prova

- **Ripetibilità della prova**
 - Ambiente definito (*hardware*, condizioni iniziali, ...)
 - 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 e analisi dei dati di prova**

Dipartimento di Informatica, Università di Pisa 6/26

Verifica e validazione

Esempio

Unità U composta dai moduli M1 - 3

Test di integrazione di U

Con driver

Con stub

Dipartimento di Informatica, Università di Pisa 7/26

Verifica e validazione

Test di unità

□ **Approcci possibili**

- **Analisi statica**
 - Dal controllo di *routine* (*desk-check*) all'ispezione
- **Analisi dinamica**
 - 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* dei moduli è responsabilità del programmatore

Dipartimento di Informatica, Università di Pisa 8/26

Verifica e validazione

Test di integrazione

□ **Costruzione e verifica del sistema**

- Componenti realizzati in parallelo
- Componenti verificati indipendentemente
- In condizioni ottimali l'integrazione è priva di problemi

□ **Problemi**

- Errori nella realizzazione dei componenti
- Modifica delle interfacce o cambiamenti nei requisiti
- Riutilizzo di componenti dal comportamento oscuro o inadatto
- Integrazione con altre applicazioni non ben conosciute

Dipartimento di Informatica, Università di Pisa 9/26

Verifica e validazione

Test di sistema e collaudo

□ **Validazione**

- Test di sistema come attività interna del fornitore
 - α alpha test (pre-collaudo)
- Collaudo come attività controllata dal committente
 - β beta test (collaudo)
 - 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

Dipartimento di Informatica, Università di Pisa 10/26

Verifica e validazione

Forme di analisi statica

□ **Effettuabili senza esecuzione del codice**

□ **Metodi di lettura [del codice] (*desk check*)**

- Impiegati frequentemente
 - Ma di scarsa efficacia per sistemi di elevata complessità
- Documentati più o meno formalmente

□ **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

Verifica e validazione

Metodi di lettura [del codice]

□ **Inspection e Walkthrough**

□ **Metodi pratici**

- Basati sulla lettura [del codice]
- 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**

Dipartimento di Informatica, Università di Pisa 12/26

 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

 Verifica e validazione

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

Dipartimento di Informatica, Università di Pisa 14/26

 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

Dipartimento di Informatica, Università di Pisa 15/26

 Verifica e validazione

Attività di *walkthrough*

- ❑ Fase 1: pianificazione
- ❑ Fase 2: lettura [del codice]
- ❑ Fase 3: discussione ←
- ❑ Fase 4: correzione dei difetti
- ❑ In ogni fase
 - Documentazione come rapporto delle attività svolte

Dipartimento di Informatica, Università di Pisa 16/26

 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

Dipartimento di Informatica, Università di Pisa 17/26

 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?
 - 4 caratteristiche nella visione utente e 2 nella visione produttore
 - La qualità in uso è valutata a posteriori

Dipartimento di Informatica, Università di Pisa 18/26

 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

Dipartimento di Informatica, Università di Pisa 19/26

 Verifica e validazione

Affidabilità

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

Dipartimento di Informatica, Università di Pisa 20/26

 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

Dipartimento di Informatica, Università di Pisa 21/26

 Verifica e validazione

Efficienza

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

Dipartimento di Informatica, Università di Pisa 22/26

 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

Dipartimento di Informatica, Università di Pisa 23/26

 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

Dipartimento di Informatica, Università di Pisa 24/26

	Verifica e validazione
	Riepilogo
<ul style="list-style-type: none">❑ Concetti e terminologia❑ Verifica, validazione, integrazione e collaudo❑ Verifica statica❑ <i>Inspection e walkthrough</i>❑ Verifica e validazione di qualità	
Dipartimento di Informatica, Università di Pisa	25/26

	Verifica e validazione
	Riferimenti
<ul style="list-style-type: none">❑ Standard for Software Component Testing, British Computer Society SIGIST, 1997❑ M.E. Fagan, Advances in Software Inspection, <i>IEEE Transaction on Software Engineering</i>, 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