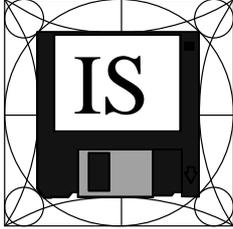


Verifica e validazione

Verifica e validazione



IS 2001-4
Corso di Ingegneria del Software

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

Con aggiornamenti di: T. Vardanega

Dipartimento di Informatica, Università di Pisa1/25

Verifica e validazione

Contenuti

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

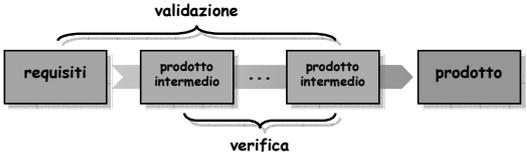
Dipartimento di Informatica, Università di Pisa2/25

Verifica e validazione

Verifica e validazione

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

validation



verification

Dipartimento di Informatica, Università di Pisa3/25

Verifica e validazione

Guasti, difetti, malfunzionamenti

- Guasto**
 - Atto od omissione, possibile causa (anche umana) di comportamento fuori norma
 - Può non avere effetto
- Errore o Difetto**
 - Stato erroneo di sistema introdotto da un guasto
 - Può restare latente, va eliminato
- Malfunzionamento**
 - Effetto di un errore, risulta in deviazione da specifica
 - Può causare danni rilevanti

Fault
un impiegato distratto

Error
un dato sbagliato in una pratica

Failure
emissione di una multa ingiusta

Dipartimento di Informatica, Università di Pisa4/25

Verifica e validazione

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

Dipartimento di Informatica, Università di Pisa5/25

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 Pisa6/25

 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 7/25

 Verifica e validazione

Integrazione

- Costruzione e verifica del sistema**
 - Componenti realizzate in parallelo
 - Componenti realizzate e verificate indipendentemente
 - In condizioni ottime, l'integrazione è banale
- Problemi**
 - Errori nella realizzazione delle componenti
 - Modifica delle interfacce o cambiamenti nei requisiti
 - Riutilizzo di componenti
 - Integrazione con altre applicazioni

Dipartimento di Informatica, Università di Pisa 8/25

 Verifica e validazione

Collaudo

- Validazione del sistema**
 - Attività svolta dal fornitore
 - Attività svolta dal committente
 - Su casi di prova definiti nel contratto
- Valore contrattuale**
 - Il collaudo è un'attività ufficiale
 - Al collaudo segue il rilascio del sistema
 - Conclusione della commessa (a meno di servizi e garanzie)

Qualifica
↓
α test o pre-collaudo
β test o collaudo

Dipartimento di Informatica, Università di Pisa 9/25

 Verifica e validazione

Verifica statica

- Verifica senza esecuzione del codice (analisi)**
- 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 10/25

 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
- Metodi tra loro complementari**

Dipartimento di Informatica, Università di Pisa 11/25

 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 12/25

 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 13/25

 Verifica e validazione

Walkthrough

- Obiettivo
 - Rivelare la presenza di difetti
 - Eseguire una lettura critica (a largo spettro) del codice
- Agenti
 - Gruppi misti ispettori/sviluppatori, ma con ruoli ben distinti
- Strategia
 - Percorrere il codice simulandone possibili esecuzioni

Dipartimento di Informatica, Università di Pisa 14/25

 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 15/25

 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 16/25

 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 17/25

 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)
 - Aderenza alle norme ed alle prescrizioni
- Prove per accuratezza

Dipartimento di Informatica, Università di Pisa 18/25

Verifica e validazione

	Verifica e validazione
	Affidabilità
<ul style="list-style-type: none"><input type="checkbox"/> Dimostrabile tramite analisi e prove<input type="checkbox"/> Verifica statica come attività preliminare<input type="checkbox"/> Liste di controllo (rispetto ai requisiti)<ul style="list-style-type: none"><input type="checkbox"/> Robustezza (tolleranza ai guasti)<input type="checkbox"/> Capacità di ripristino<input type="checkbox"/> Aderenza alle prescrizioni<input type="checkbox"/> Prove per maturità	
Dipartimento di Informatica, Università di Pisa	19/25

	Verifica e validazione
	Usabilità
<ul style="list-style-type: none"><input type="checkbox"/> Le prove sono imprescindibili<input type="checkbox"/> Verifica statica come attività complementare<input type="checkbox"/> Liste di controllo (rispetto ai manuali d'uso)<ul style="list-style-type: none"><input type="checkbox"/> Comprensibilità<input type="checkbox"/> Apprendibilità<input type="checkbox"/> Aderenza alle prescrizioni<input type="checkbox"/> Questionari all'utenza (a seguito di prove)<ul style="list-style-type: none"><input type="checkbox"/> Operabilità<input type="checkbox"/> Attraenza	
Dipartimento di Informatica, Università di Pisa	20/25

	Verifica e validazione
	Efficienza
<ul style="list-style-type: none"><input type="checkbox"/> Prove necessarie con la tecnologia attuale<input type="checkbox"/> Verifica statica come attività preliminare<input type="checkbox"/> Liste di controllo (rispetto a criteri realizzativi)<ul style="list-style-type: none"><input type="checkbox"/> Efficienza temporale (algoritmica)<input type="checkbox"/> Efficienza spaziale (uso delle risorse)<input type="checkbox"/> Miglioramento e confidenza<ul style="list-style-type: none"><input type="checkbox"/> L'efficienza provata fornisce confidenza<input type="checkbox"/> La verifica statica non da confidenza (attualmente), ma indicazioni importanti per migliorare il codice	
Dipartimento di Informatica, Università di Pisa	21/25

	Verifica e validazione
	Manutenibilità
<ul style="list-style-type: none"><input type="checkbox"/> Verifica statica come strumento ideale<input type="checkbox"/> Liste di controllo (norme di codifica)<ul style="list-style-type: none"><input type="checkbox"/> Analizzabilità<input type="checkbox"/> Modificabilità<input type="checkbox"/> Aderenza alle prescrizioni<input type="checkbox"/> Liste di controllo (batterie di prove)<ul style="list-style-type: none"><input type="checkbox"/> Verificabilità<input type="checkbox"/> Prove per la stabilità	
Dipartimento di Informatica, Università di Pisa	22/25

	Verifica e validazione
	Portabilità
<ul style="list-style-type: none"><input type="checkbox"/> Verifica statica come strumento ideale<input type="checkbox"/> Liste di controllo (norme di codifica)<ul style="list-style-type: none"><input type="checkbox"/> Adattabilità<input type="checkbox"/> Aderenza alle prescrizioni<input type="checkbox"/> Prove come strumento complementare<ul style="list-style-type: none"><input type="checkbox"/> Facilità d'installazione<input type="checkbox"/> Compatibilità ambientale<input type="checkbox"/> Facilità di sostituzione	
Dipartimento di Informatica, Università di Pisa	23/25

	Verifica e validazione
	Riepilogo
<ul style="list-style-type: none"><input type="checkbox"/> Concetti e terminologia<input type="checkbox"/> Verifica, validazione, integrazione e collaudo<input type="checkbox"/> Verifica statica<input type="checkbox"/> <i>Inspection e walkthrough</i><input type="checkbox"/> Verifica e validazione di qualità	
Dipartimento di Informatica, Università di Pisa	24/25

	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	25/25