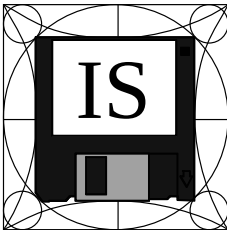




Misurazione del *software*



IS 2001-4
Corso di Ingegneria del Software
V. Ambriola, G.A. Cignoni,
C. Montangero, L. Semini
Con aggiornamenti di: T. Vardanega

Dipartimento di Informatica, Università di Pisa1/29



Misurazione del *software*

Contenuti

- ❑ La misura come strumento scientifico
- ❑ Misure, metriche e indicatori
- ❑ Metriche del *software*
- ❑ Metriche per *software* orientato agli oggetti
- ❑ Approfondimenti: accuratezza delle metriche

Dipartimento di Informatica, Università di Pisa2/29



Misurazione del *software*

Misurare

- ❑ Un'attività quotidiana
 - Grandezze fisiche: massa, lunghezza, velocità ...
 - Grandezze economiche: prezzi, inflazione, PIL, Mib30, ...
 - Di tutto di più: gradimento, reddito, caratteristiche fisiche, ...
- ❑ Misurare per conoscere
- ❑ Misurare per giudicare
- ❑ L'effetto di semplificazione ed imprecisione

Dipartimento di Informatica, Università di Pisa3/29



Misurazione del *software*

Scienza e misura

- ❑ Finalità della misura
 - Rendere oggettivi i risultati degli esperimenti
 - Oggettività implica ripetibilità, confronto, confidenza
 - Avere dati da cui ricavare modelli matematici
- ❑ Limiti intrinseci della misura
 - È una approssimazione ← misure di realtà fisiche
 - È una astrazione ← misure per valutare o stimare

Dipartimento di Informatica, Università di Pisa4/29



Misurazione del *software*

Misurazione

- ❑ Misurazione
 - Processo che assegna numeri o simboli ad attributi di entità del mondo reale, per descriverle secondo regole definite
- ❑ Concetti rilevanti
 - Entità persona | semaforo | programma
 - Attributo età | stato | dimensione
 - Valore 18 anni | rosso | 35 linee
(numerico, simbolico)

Dipartimento di Informatica, Università di Pisa5/29



Misurazione del *software*

Misura e metrica

- ❑ Misura
 - Risultato della misurazione
 - Assegnazione empirica ed oggettiva di un valore (numerico o simbolo) ad un'entità, per caratterizzarne un attributo specifico
- ❑ Metrica, un insieme di regole
 - Per stabilire le entità da misurare
 - Per definire gli attributi rilevanti
 - Per definire l'unità di misura
 - Per definire una procedura per assegnare numeri e simboli

Dipartimento di Informatica, Università di Pisa6/29



Misurazione del software

Indicatori

- ❑ Caratteristiche difficilmente misurabili
- ❑ Metriche usate per stimare caratteristiche
 - Identificare attributi misurabili
 - Per stimare (con incertezza) caratteristiche non misurabili
- ❑ Esempio
 - La dimensione del software è un attributo misurabile
 - La manutenibilità è stimata in base alla dimensione (ma non solo!)

Dipartimento di Informatica, Università di Pisa7/29



Misurazione del software

Le metriche nella produzione

- ❑ Strumenti di valutazione e controllo
- ❑ Entità da misurare
 - *Processi* insiemi correlati di procedure astratte
 - *Progetti* attività concrete legate a tempo e risorse
 - *Prodotti* beni e servizi in uscita dai progetti
 - *Risorse* elementi impiegati (e consumati) dal progetto per produrre prodotti
- ❑ Categorie di attributi
 - *Attributi interni* → misurabili rispetto alle entità
 - *Attributi esterni* → misurabili rispetto all'ambiente

Dipartimento di Informatica, Università di Pisa8/29



Misurazione del software

Le metriche del software

- ❑ Un problema aperto
 - Il software è difficile da misurare
 - Il software presenta aspetti diversi
 - Le tecnologie cambiano in fretta
 - L'ambiente ha una grande influenza
- ❑ Tipi di metriche
 - Del prodotto in sé → quanto è grande
 - Delle sue funzionalità → cosa deve fare
 - Del suo comportamento → cosa e quando succede

Dipartimento di Informatica, Università di Pisa9/29



Misurazione del software

Linee di codice sorgente (SLOC)

- ❑ La metrica più intuitiva e più usata
- ❑ Conteggio dei costrutti
 - Semplificato ed adattato alle funzionalità degli editor
 - Usata per derivare informazioni di costo e produttività
- ❑ Limiti
 - Dipendente dal linguaggio
 - Dipendente dallo stile di codifica

Dipartimento di Informatica, Università di Pisa10/29



Misurazione del software

Software Science - 1

- ❑ Maurice H. Halstead. *Elements of Software Science*, Elsevier, Amsterdam, 1977
- ❑ Complessità del codice
 - Misura a posteriori o stima durante il progetto
 - Controversa
 - Basata sul numero e sul tipo dei costrutti
 - Basata su elementi di psicologia cognitiva e statistica
- ❑ Limiti
 - Adatta solo ai primi linguaggi di programmazione
 - Oggi è statisticamente inefficace

Dipartimento di Informatica, Università di Pisa11/29



Misurazione del software

Software Science - 2

Unità di misura
n1: numero di operatori distinti usati dal programma
n2: numero di operandi distinti usati dal programma
N1: numero di occorrenze degli operatori
N2: numero di occorrenze degli operandi

Usate per determinare (tra l'altro)
* La dimensione (lunghezza) N di un programma
$$N = n1 \log n1 + n2 \log n2$$
dove log è in base 2
* Il volume V di un programma
$$V = N \log (n1 + n2)$$
per cui V varia con il variare del linguaggio di programmazione

Halstead assegna un valore (livello) I come costante di linguaggio, ma la successiva ricerca ha mostrato che essa dipende anche dal programmatore

Dipartimento di Informatica, Università di Pisa12/29



Misurazione del software

Complessità ciclomatica - 1

- **Proposta da Thomas McCabe nel 1976**
 - Indipendente dal linguaggio di programmazione
- **Complessità del flusso di controllo**
 - Funzione dei possibili cammini indipendenti sul grafo di flusso
 - Rappresentazione astratta del codice
- **Limiti**
 - Fallibilità dimostrata (ciononostante usata)
 - Costosa da applicare prima di scrivere il codice

Dipartimento di Informatica, Università di Pisa

13/29



Misurazione del software

Complessità ciclomatica - 2

- **Il valore determinato viene confrontato con soglie prefissate**
 - P.es.: 1-10 → complessità bassa
 - 21-50 → complessità elevata
 - >50 → complessità inaccettabile
- **Corrisponde all'esatto numero di casi di prova necessari per verificare ogni possibile esito di ogni decisione dell'unità**
- **È preferibile calcolarla con l'ausilio di strumenti automatici**

Dipartimento di Informatica, Università di Pisa

14/29



Misurazione del software

Complessità ciclomatica - 3

- **Definizione algebrica**

$v(G) = e - n + p$	numero di percorsi lineari in G
e	numero degli archi (flusso)
n	numero dei nodi (espressioni o comandi)
p	numero delle componenti connesse da ogni arco (=2)
- **Esempio, una sequenza, $v(G) = 3 - 4 + 2 = 1$**



Dipartimento di Informatica, Università di Pisa

15/29

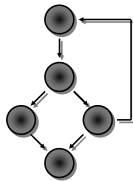


Misurazione del software

Complessità ciclomatica - 4

- **Errore in difetto**
 - La complessità reale è maggiore di quella misurata
- **Esempio, $v(G) = 6 - 5 + 2 = 3$**
 - Ma codice sorgente offuscato!

```
int _INT = 1 ; int _INT_ =
0; int Int_ ( int _int )
{ if ( _int == INT_ ) return
_INT ; else return _int
* Int_ ( _int - _INT ) ; }
```



Dipartimento di Informatica, Università di Pisa

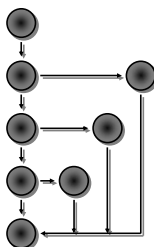
16/29



Misurazione del software

Complessità ciclomatica - 5

- **Errore in eccesso**
 - La complessità reale è inferiore a quella misurata
- **Esempio, $v(G) = 10 - 2 + 2 = 4$**
 - Switch visto come una sequenza di if annidati
 - $v(G) = C + 1$
 - C numero dei casi (= 3)



Dipartimento di Informatica, Università di Pisa

17/29



Misurazione del software

Function Point - 1

- **Una proposta di A. J. Albrecht, 1977**
 - Non misura il software, ma le sue entità logico-funzionali
 - Misura di dimensione del progetto e di produttività
 - Indipendente dal linguaggio di programmazione
 - International Function Point User Group (IFPUG)
- **Conteggio dei punti funzione**
 - Funzionalità differenziate per categorie
 - Funzionalità pesate in base agli attributi del prodotto
- **Limiti**
 - Misurazioni diverse possono dare risultati leggermente diversi
 - Risultati ottimali per sistemi gestionali

Dipartimento di Informatica, Università di Pisa

18/29



Misurazione del software

Function Point – 2

Componenti misurate	Definizione
External Input (EI)	Processo elementare a seguito del quale dati entrano nella componente
External Output (EO)	Processo elementare a seguito del quale dati escono dalla componente
External Inquiry (EQ)	Processo elementare per il quale specifici dati interni sono richiesti alla componente e da essa emessi
Internal Logical File (ILF)	Gruppo di dati correlati, interni alla componente, alimentati tramite EI
External Interface File (EIF)	Gruppo di dati correlati, esterni alla componente, usati come riferimento

Dipartimento di Informatica, Università di Pisa

19/29



Misurazione del software

Function Point – 3

- **Unadjusted FP (valore grezzo)**
 - Ciascuna componente viene associata ad uno specifico grado di complessità
 - Ogni componente pesa quanto il suo grado di complessità
- **Adjusted FP (valore raffinato)**
 - 14 fattori d'influenza (IF), ciascuno con peso da 0.00 a 0.05
 - $AFP = UFP \times (0.65 + \sum IF)$

Componente	Complessità			Totale
	Bassa	Media	Alta	
EI	x 3	x 4	x 6	
EO	x 4	x 5	x 7	
EQ	x 3	x 4	x 6	
ILF	x 7	x 10	x 15	
EIF	x 5	x 7	x 10	

Dipartimento di Informatica, Università di Pisa

20/29



Misurazione del software

Punti oggetto

- Boehm *et al.*, 1995, insieme a COCOMO 2.0
- **Conteggio delle componenti**

Tipo	semplice	medio	difficile
Screen	1	2	3
Report	2	5	8
3GL Component	-	-	10
- **Fattore di aggiustamento rispetto alla percentuale di riuso**

Dipartimento di Informatica, Università di Pisa

21/29



Misurazione del software

Standard per misure funzionali

- **ISO/IEC 14143 – Functional size measurement**
 - Definire la terminologia del settore
 - Definire i criteri per valutare le metriche funzionali
 - Definire i criteri per accreditare i professionisti che le usano
- **Concetti definiti**
 - Accuratezza di una misura funzionale
 - Accuratezza di una metrica funzionale
 - Ripetibilità e riproducibilità di una metrica funzionale
 - Soglia di sensibilità di una metrica funzionale
 - Applicabilità ad un dominio funzionale

Dipartimento di Informatica, Università di Pisa

22/29



Misurazione del software

Metriche per software OO

- **Occorrono metriche dedicate**
 - Le metriche tradizionali non sono accurate
 - Complessità funzionale e complessità strutturale
 - Non linearità del codice (la misura SLOC non funziona!)
 - Uso di strutture e funzioni complesse (McCabe non funziona come misura assoluta!)
- **Metriche per i metodi**
- **Metriche per le classi**
- **Metriche per i sistemi**

Dipartimento di Informatica, Università di Pisa

23/29



Misurazione del software

Metrica per i metodi

- **Un metodo è sostanzialmente procedurale**
 - Si può usare McCabe come metrica di base
 - Corretta per tenere conto di interfacce e variabili locali
- **Complessità di un metodo**

$$MC = w_1 MIC + w_2 MLVC + w_3 MCoC$$

MIC complessità dell'interfaccia

MLVC complessità delle variabili locali

MCoC complessità ciclomatica del codice

w_i pesi determinati statisticamente

Dipartimento di Informatica, Università di Pisa

24/29



Misurazione del *software*

Metrica per le classi

□ Complessità di una classe

$CC = w_3 CCL + w_4 CCI$ CCI $CCL = w_5 ECCL + w_6 ICCL$ $ECCL = \sum_i MIC_i$ $ICCL = w_7 CACL + w_8 CMCL$ $CACL = \sum_n CA_n$ $CMCL = \sum_k MC_k$ CA w_i	complessità dell'IF della classe complessità locale complessità locale esterna complessità locale interna complessità degli attributi locali complessità dei metodi locali costante o CC di classe attributo pesi determinati statisticamente
--	--

Dipartimento di Informatica, Università di Pisa

25 / 29



Misurazione del *software*

Metrica per i sistemi

□ Proprietà della metrica per le classi

- Considera solo la parte locale della classe
- Considera la complessità dell'uso delle classi negli attributi

□ Complessità del sistema

$$SC = \sum_n CC_n$$

□ Metriche utilizzabili come fattori d'influenza

NC	Numero totale di classi
NRC	Numero di classi radice
NLC	Numero di classi foglia

Dipartimento di Informatica, Università di Pisa

26 / 29



Misurazione del *software*

Accuratezza delle metriche

□ Precisione

- In termini di esattezza e ripetibilità
- SLOC e McCabe sì, FP no

□ Aderenza alla realtà

- La misura corrisponde all'evidenza sperimentale
- Quando le metriche sono usate come indicatori

□ Esempi su McCabe, misura di complessità

Dipartimento di Informatica, Università di Pisa

27 / 29



Misurazione del *software*

Riepilogo

□ La misura come strumento scientifico

□ Misure, metriche e indicatori

□ Metriche del *software*

□ Metriche per *software* orientato agli oggetti

□ Approfondimenti: accuratezza delle metriche

Dipartimento di Informatica, Università di Pisa

28 / 29



Misurazione del *software*

Riferimenti

- N.E. Fenton, S.L. Pfleger, "Software Metrics: A Rigorous & Practical Approach, II Ed., Int. Thompson Computer Press, 1997
- A.J. Albrecht, J.E. Gaffney, "Software Function, [...] A Software Science Validation", IEEE Trans. on Software Engineering, 1983
- A. Alessandroni, "La stima dei costi dei sistemi informativi automatizzati", AIPA, <http://www.aipa.it>
- B. Boehm e altri, "Cost Models for Future Software Life Cycle Processes: CoCoMo II", Centre for Software Engineering, <http://sunset.usc.edu/>
- P. Nesi, T. Querci, "Stima della complessità e analisi di sistemi object oriented", De qualitate, 1996

Dipartimento di Informatica, Università di Pisa

29 / 29