DIETI UNINA UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE TECNOLOGIE DELL'INFORMAZIONE

Corso di Laurea in Ingegneria Informatica

Corso di Ingegneria del Software

Metriche del software

Sommario

- Definizioni
- Classificazione
- Metriche di prodotto
- Metriche di processo e di qualità
- Metriche object oriented

Riferimenti

C.Ghezzi, M. Jazayeri, D. Mandrioli; *Ingegneria del Software. Fondamenti e principi*, II edizione. Pearson.
Cap 6 § 6.9.3.1, 6.9.3.2

Misure nel processo di produzione del software

La gestione di un progetto software richiede di monitorarne l'avanzamento e di valutarne quantitativamente caratteristiche relative non solo al prodotto finale, ma anche ai semilavorati intermedi, anche al fine di decidere eventuali correttivi da apportare al processo di produzione.

Si ha quindi necessità di tecniche per "**misurare**" opportune grandezze al fine di rispondere a domande quali:

- Ci sono ritardi nel progetto?
- Si è consumato o si sta consumando di più (o di meno) del previsto in termini di risorse (in particolare il *tempo*)?
- Il software prodotto rispetta i requisiti di qualità?

Misure nel processo di produzione del software

- Ciò richiede:
  - **Stima a priori** di tali grandezze
  - **Misurazione** di tali grandezze, **durante il processo** o **a posteriori**,

- È un campo che ha prevalentemente basi empiriche (confronto con esperienze acquisite).

Definizioni

- Una **metrica** caratterizza in modo quantitativo e misurabile attributi semplici di una entità
  - *Es. considerando come entità il prodotto software finale in forma sorgente (codice), e come attributo la "dimensione", si possono adottare metriche quali:*
    - *il numero di linee di codice sorgente*
    - *il numero di linee di codice né vuote né bianche*
    - *il numero di caratteri*
    - *il numero di linee di commento*
    - *…*
- Una **misura** è una assegnazione oggettiva ed empirica di un valore ad un attributo.

Definizioni

- **Attributi esterni**: visibili e di interesse per l'utente (facilità d'uso, prestazioni, efficienza, …)
- **Attributi interni**: visibili e di interesse per i produttori/sviluppatori (modularità, complessità, …)

- In generale accade che
  - le metriche sono associate ad attributi interni
  - gli attributi esterni dipendono da attributi interni e sono più complessi.

Metriche del software

- Le **metriche** del software sono lo strumento attraverso il quale l'ingegnere del software predice e/o misura gli aspetti di processo, di gestione delle risorse, e di prodotto, rilevanti nell'ingegnerizzazione del software.

- Servono da guida nella ricerca di soluzioni efficienti ed efficaci a problemi complessi.

Utilità delle metriche

- Miglioramento del processo software
- Pianificazione e controllo dell'andamento di un progetto
- Valutazione della qualità del prodotto
- Stima dello sforzo richiesto per lo sviluppo, per il testing, per un intervento di manutenzione, …

Classificazione delle metriche

- Due classificazioni, non necessariamente in contrapposizione

- Metriche di prodotto
- Metriche di processo
- Metriche di qualità

- Metriche funzionali
- Metriche strutturali
  - Metriche dimensionali

Metriche di Prodotto

- Metriche dimensionali (*size metrics*):
  - Ad es., LoC, *statement* eseguibili,
- Metriche funzionali:
  - Punti Funzione
- Metriche di Complessità
  - Ad es., Complessità ciclomatica (di McCabe)
- Metriche di *Halstead* (*Software Science metrics*)
  - Lunghezza dell'implementazione, volume di programma, sforzo di programmazione
- *N.B. Alcune di esse sono fortemente correlate*

Metriche di prodotto. Metriche dimensionali

- La *dimensione del programma* è generalmente valutata in termini di

numero di linee di codice (LOC)

- È un indicatore considerato tradizionalmente importante, anche se in realtà non sempre significativo, specie se riferito a linguaggi di programmazione differenti, dove diverse sono le densità di significato per ogni linea di codice.

- Non è ancora chiaro se nel conteggio siano da considerare anche i commenti (in genere no), utili per la documentazione, e la parte dichiarativa, che contribuisce in maniera rilevante a determinare la qualità del SW.

Metriche di prodotto. Metriche dimensionali

- Altre metriche dimensionali legate alle LoC:
  - **NCNB** = no-comment no-blank, indicate anche con NLoC (non comment LoC), ELoC (Effective LoC) => calcola tutte le linee di codice eccetto commenti e linee vuote
  - **Statement Eseguibili.** Tra le metriche basate su LoC è quella meno dipendente dallo stile di programmazione
- **CLoC** = Linee di commento
- **CP** = (CLoC) / (NLoC+CLoC)
  - Utile anche come metrica di qualità => per valutare la comprensibilità, manutenibilità riusabilità
    - Secondo studi empirici il CP ideale si attesta attorno al 30%

Grandezze misurabili indirettamente dalle LoC

- **Produttività (P)** P = LOC / M
  - (M = num. mesi uomo; tempo occorso per sviluppare il programma)

- **Qualità (Q)** Q = LOC / E
  - (E = numero errori rilevati. 1/Q dà la densità di errori rilevati nel codice)

- *NOTA: è da considerare di qualità "migliore" un programma in cui non sono stati rilevati errori in fase di testing, o uno in cui sono stati rilevati (e corretti) molti errori?*

Grandezze misurabili indirettamente

- **Livello di documentazione (D)** D = PD / LOC
  - (PD = pagine di documentazione)
- **Densità dei commenti in un programma** CLOC / LOC
- **Stabilità dei requisiti**
  - numero iniziale di requisiti / numero totale di requisiti

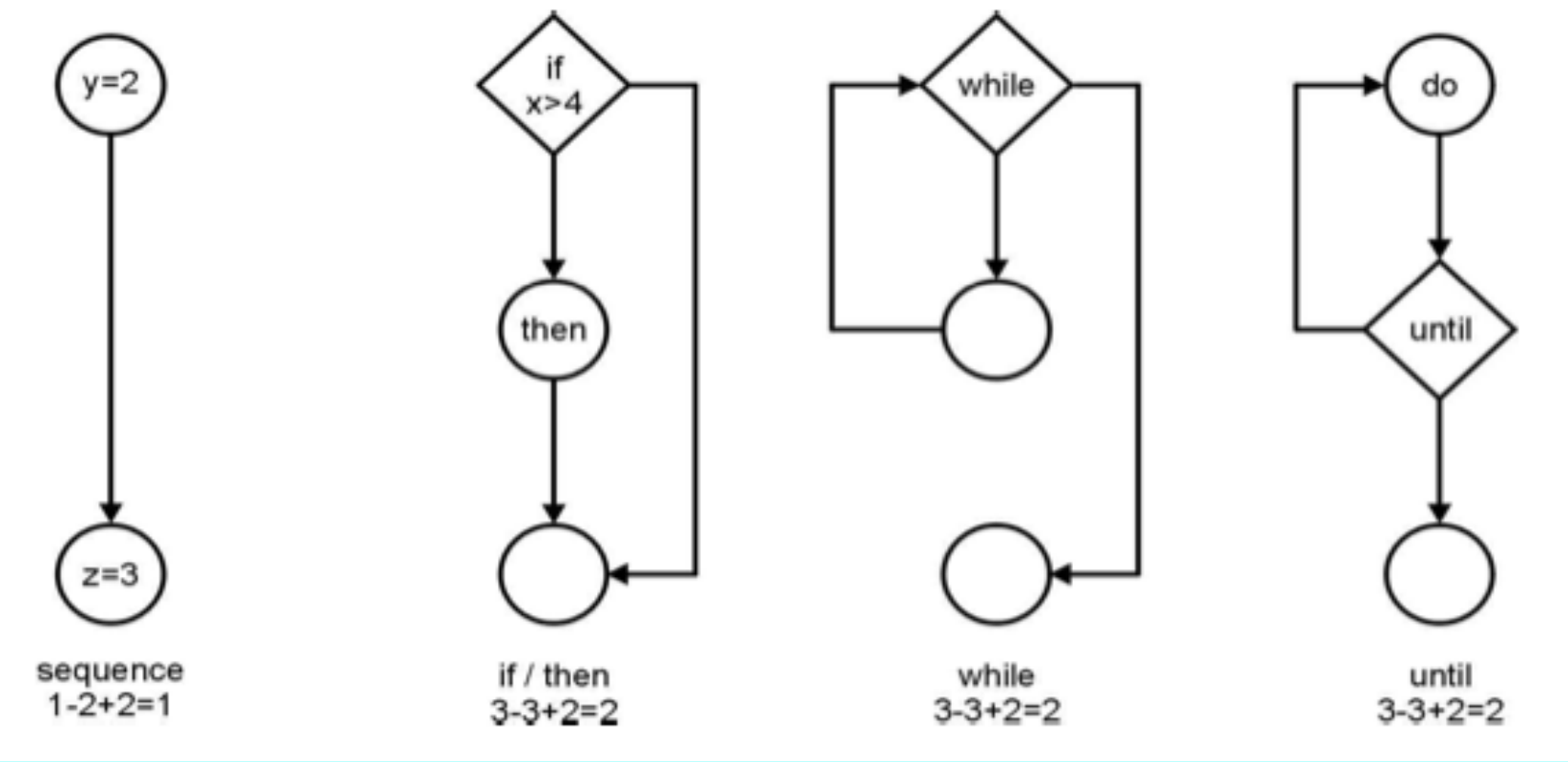
- **LOC misura la lunghezza del codice, non va considerata come un indicatore della sua complessità**
- La metrica descritta permette di *misurare* a posteriori alcune caratteristiche di un programma e del suo sviluppo.
- Al fine di verificare il corretto svolgimento del processo di sviluppo, è necessario avere metodi di *stima* a priori.

Metriche di prodotto. Metriche di complessità

- **Numero ciclomatico.**
- *Dà una misura, su base matematica, della complessità del software in funzione dei possibili cammini percorribili nel programma*.

- La struttura del programma è vista come un grafo i cui nodi sono i principali punti di scelta del programma (if, while,....) => GRAFO di FLUSSO
- È dato da una delle seguenti espressioni:
  1) V(g)= Numero di regioni del grafo
  2) V(g)= F-N+2C dove F=archi, N=nodi, C=numero dei componenti del grafo disconnessi
  3) V(g)= P+1 dove P=nodi predicati

Numero Ciclomatico: Esempi



Metriche di prodotto. Metriche di complessità

- **Flusso di Informazione**

$$C = L * (\text{Fan-in} * \text{Fan-out})^2$$

L: metrica dimensionale (LoC)

Fan-in: numero di flussi entranti nel componente

Fan-out: numero di flussi uscenti dal componente

Metriche di Processo

- Servono per la valutazione di un progetto
- Metriche di Modularità
  - Organizzative, di risorse, di personale, di gestione tecnologica
- Metriche di Gestione
  - Del progetto, di configurazione
- Metriche del ciclo di vita
  - Di definizione del problema, di analisi, di progetto, di implementazione

Metriche di Qualità

Metriche relative ai fattori di qualità; ad es.:

- Metriche di *dependability*
  - Affidabilità (probabilità di fallimento in [0,t])
  - MTTF (Mean Time To Failure), MTBF (Mean Time Between Failures)
  - Disponibilità (probabilità di fallimento in *t*)
- Metriche di manutenibilità (ad es., di correggibilità, di espandibilità)
- Metriche di usabilità, di prestazioni, ecc.
- *N.B. Per alcuni di questi fattori, è difficile trovare metriche rappresentative (ad es., come misurare l'usabilità?)*
- *N.B. Anche per metriche ben definite (come l'affidabilità) può essere complicato avere misure accurate*

Metriche *object-oriented*

- La diffusione dei linguaggi OO ha reso obsolete alcune metriche tradizionali che consideravano dati e funzioni
- Si focalizzano
  - Sulla struttura interna dell'oggetto
  - Sulla complessità esterna, che misura l'interazione tra le entità

Metriche *object-oriented*: Metriche di pianificazione

- **Application size:**
  - Danno una misura della quantità di lavoro necessaria
    - Ad es., *Number of key classes, Number of support classes, Number of subsystems*
- **Staffing size**
  - Danno una misura del personale necessario
    - Ad es., *Person-Days per Class, Classes per Developer*
- **Scheduling**
  - Forniscono uno strumento di pianificazione temporale dei processi nell'ambito della gestione dei progetti

Metriche *object-oriented*: Metriche di progetto

- Metriche di complessità di **Chidamber e Kemerer (CK)**: Sei metriche:
  - *Weighted Method per Class*: definita come numero di metodi per classe *oppure* come somma delle complessità di tutti i metodi delle classi
  - *Response for a class*: cardinalità dell'insieme di tutti i metodi che possono essere chiamati in risposta ad un messaggio ricevuto da un oggetto della classe.
  - *Coupling between object classes*: misura dell'accoppiamento
    - Numero di classi con la quale una data classe è accoppiata

Metriche *object-oriented*: Metriche di progetto

- *Lack of Cohesion*: misura della (mancanza di) coesione
  - Misura il grado di diversità fra metodi in relazione alle variabili usate
  - Si può calcolare contando gli insiemi disgiunti prodotti dall'intersezione fra insiemi di attributi usati dai metodi. I metodi sono più simili se operano sugli stessi attributi
- *Depth of inheritance tree*: rappresenta la profondità dell'albero in una gerarchia di classi
  - Una metrica di supporto è il *Number of Methods Inherited*
- *Number of children*: numero di sottoclassi immediatamente subordinate a una certa classe della gerarchia
  - Maggiore è tale numero, maggiore è il riuso
  - Tuttavia, maggiore è tale numero maggiore è la probabilità di avere una astrazione errata

Metriche *object-oriented*: Ulteriori metriche

- Altre classificazioni considerano:
  - **Metriche strutturali**
    - Ad es., *numero di oggetti, numero di messaggi, numero di variabili globali in ogni classe, rapporto metodo/classe, rapporto metodi pubblici/privati, rapporto tra classi astratte/oggetti istanziati, tra LoC/metodi*
  - **Metriche orientate al riuso**
    - Ad es., *numero oggetti riusati, percentuale di oggetti riusati che vengono modificati*