Stefano Carta

Domanda 1:

Perché il numero di righe di codice non sono una buona misura della complessità del programma?

Perché i programmatori che possono avere diverse abilità e conoscenze, possono scrivere codice “diverso” ottenendo lo stesso risultato.

Uno programmatore esperto, a volte, può scrivere in poche righe quello che un programmatore alle prime armi scriverebbe di sicuro in molte più righe di codice.

Domanda 2:

Suggerire una metrica per migliorare la qualità di un SW. Quali sono i difetti della metrica offerta?

Una buona metrica da poter adottare a mio parare, una volta valutato il team, è la valutazione della complessità computazionale e quindi dei tempi di risposta di un SW.

Un problema legato a questo fatto è naturalmente il costo derivato dall’assunzione di un team di lavoro che dia determinate garanzie.

Fabio Carta

Domanda 1:

Perché un programma scritto in poche righe può avere una complessità maggiore di uno con più righe, basta considerare per esempio il caso di un programma con una serie di for annidati rispetto ad un programma con semplici printf ma più lungo. Il primo ha una complessità maggiore.

Domanda 2:

Una metrica per misurare la qualità del software può essere il tempo d’esecuzione del programma. Il difetto è che questa può dipendere da fattori esterni quale per esempio la velocità del processore del computer.

Omar Desogus

**1. Perché il numero di righe di un programma non è una buona misura della comp lessità del programma?**

­ Il numero di righe di un programma non è un preciso parametro di misurazione della complessità poiché lo stesso programma può essere scritto in diversi modi e con lunghezza differente ma avere lo stesso risultato, due programmi aventi lo stesso scopo possono dunque avere lo stesso numero di righe ma complessità differenti.

**2. Suggerite una metrica per misurare la qualità del software. Quali sono i difetti della vostra metrica?**

­ Una buona metrica potrebbe essere il numero di righe in rapporto al tempo di esecuzione. Il problema di questa metrica è il fatto che nel caso di problemi complessi il risultato desiderato può essere leggermente diverso da caso a caso ed entrano in gioco altri fattori.

Luigi Zucca

**Perché il numero di righe di un programma non è una buona misura della complessità del programma?**

Pur essendo un parametro misurabile, il numero di righe non fornisce informazioni soddisfacenti sulla complessità in quanto è possibile implementare programmi che soddisfano gli stessi requisiti

ma con lunghezze completamente diverse. Un esempio di questo si ha nella programmazione ad oggetti, nella quale è possibile realizzare metodi nelle nostre classi, oppure semplicemente utilizzare metodi ereditati dalla superclasse che compiano le stesse azioni.

**Suggerite una metrica per misurare la qualità del software. Quali sono i difetti della vostra metrica?**

Una possibile metrica è la modularità, in quanto consente di suddividere un progetto software in un insieme di moduli (divide et impera). Il deficit della modularità è dato dal fatto che a volte, visto che i diversi moduli possono essere sviluppati da programmatori diversi, è possibile che si verifichino errori dovuti alla scarsa comunicazione tra i componenti del team.

Giovanni Bertulu

**Perché il numero di righe di un programma non è una buona misura della com­plessità del programma?**

Ogni riga di un programma è composta di una serie d’istruzioni legate al linguaggio tramite il quale è scritto.

Nella programmazione, come in altri contesti, è possibile raggiungere degli obiettivi seguendo percorsi differenti.

E’ possibile svolgere operazioni, utilizzando una serie d’istruzioni elementari oppure usare un numero minore d’istruzioni leggermente più complesse e raggiungere lo stesso scopo.

Se per complessità intendiamo il grado di difficoltà a raggiungere un determinato obiettivo,

il numero di righe non è una buona misura, proprio per il motivo espresso sopra.

**Suggerite una metrica per misurare la qualità del software. Quali sono i difetti della vostra metrica?**

Una metrica potrebbe essere la velocità di completamento di un’azione.

Esempio classico degli algoritmi di ordinamento.

Il difetto consiste nell’input assegnato. Al variare dell’input, un algoritmo potrebbe avere tempi migliori o peggiori rispetto ad un altro nel completare il task.

Luca Fadda

**Perché il numero di righe di un programma non è una buona misura della com­plessità del programma?**

Non è una buona misura della complessità del programma perchè quello che si può fare in 100 righe di codice è possibile implementarlo anche in 50 righe di codice in modo semplificato, cambiando anche l'identazione.

**Suggerite una metrica per misurare la qualità del software. Quali sono i difetti della vostra metrica?**

Vuoto

Alessando Argiolas

**Perché il numero di righe di un programma non è una buona misura della complessità del programma?**

Il numero di righe di un programma potrebbe essere maggiore o minore ad un altro, prima di tutto, per questioni di indentazione, quindi può cambiare a seconda del programmatore. Inoltre, un programma con meno righe non vuole dire che occupi meno memoria di un altro, spesso usando più righe si risparmia molta memoria. Un esempio potrebbe essere l’uso di un vettore statico e l’uso di un vettore dinamico: il vettore statico viene implementato con una riga di codice, ma occupa memoria che potrebbe non essere utilizzata, mentre un vettore dinamico viene implementato con più righe di codice ma utilizza solo la memoria necessaria. Per finire, un programma con più righe di codice non vuol dire che abbia più funzioni o che sia più efficiente, spesso vi sono pezzi di codice ripetuti e inutili.

**Suggerite una metrica per misurare la qualità del software. Quali sono i difetti della vostra metrica?**

Due criteri molto importanti per misurare la qualità di un software sono la velocità di esecuzione e la memoria utilizzata. Tra due software che svolgono lo stesso compito, quello più veloce e che occupa meno memoria è sicuramente il migliore. Però questi due criteri possono essere utilizzati solo per comparare software che svolgono lo stesso tipo di funzioni, infatti un software che permette di ascoltare musica occupa sicuramente meno memoria di un software per videogiochi, ma questo non vuol dire che sia più efficiente. Inoltre un software più lento e più pesante, potrebbe essere più comprensibile di un software più veloce e più leggero, e quindi potrebbe essere preferito dall’utente.