

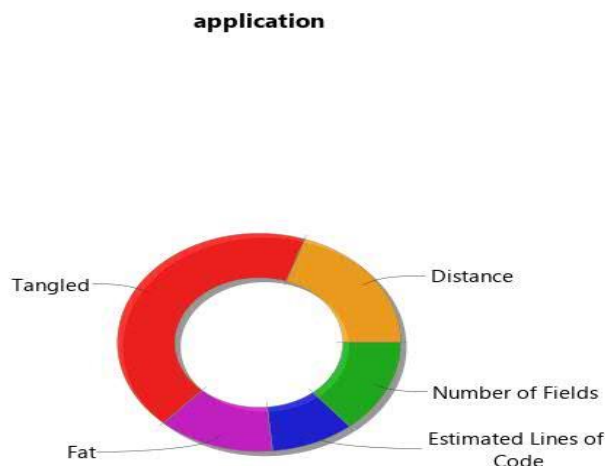
ANALISI STAN4J

Lavoro svolto da:

-Magno Alessandro : 4478234

Analisi svolte

A livello di applicazione il tool segnala una violazione della metrica tangle, la quale rientra nell'insieme di metriche riguardanti la complessità. Un tangle è una porzione di un grafico delle dipendenze in cui gli elementi coinvolti sono interdipendenti, presentano cioè una dipendenza ciclica. La presenza dei tangles comporta un aumento della complessità del progetto. Stan4j calcola i tangles in percentuale e in questo caso il valore violato, associato alla cartella 'application', è di '10.95%'. Essendo un valore abbastanza elevato, bisognerebbe ridurlo trasformando il grafo in esame in un grafo aciclico. Ovviamente il numero di archi è legato al numero di linee di codice dipendenti, ed è molto più facile rimuovere un arco riferito ad un legame leggero, piuttosto che ad uno pesante. Per eliminare le dipendenze cicliche bisogna ridurre il più possibile il numero di archi, oppure, se questo è già ridotto al minimo, eliminare gli archi che puntano nella direzione sbagliata. Dal Pollution Chart, di seguito, il quale indica il grado di 'inquinamento' del codice e la percentuale in base a questo grado, di tutte le metriche violate, si può vedere come la metrica 'Tangled' rappresenti la maggior parte (43%) del valore totale della pollution (1.62).



Pollution: 1.62

A livello di package vengono segnalate 4 violazioni della metrica distance (D) di Robert C.Martin, la quale indica la distanza che il progetto, o i suoi elementi, posseggono dalla retta ideale Instabilità-Astrazione. Minore è la distanza del software dalla linea, maggiore è la sua qualità. La distance è data dalla formula $D = A + I - 1$, ed è sempre compresa tra -1 e +1: i valori desiderati sono quelli intorno allo zero (main sequence). I pacchetti che violano la metrica sono 'utils', 'rifornimenti', 'utenti', 'vendite', 'model'. Prendendo in esame il package 'utenti', dista dalla retta '-0.71' e coordinate di [0.09,0.2] (le quali rappresentano [instabilità, astrazione]), essendo entrambi i valori molto vicini allo zero, la violazione intende esprimere l'elevata concretezza e rigidità dell'elemento in questione. Per ridurre la distanza di questo pacchetto dalla main sequence, si dovrebbe aumentare l'astrazione del progetto, sintetizzando il più possibile i modelli concettuali ed

escludendo i dettagli inutili. Per gli altri pacchetti il valore della coordinata riguardante l'astrazione è più piccolo di quello del pacchetto precedente, questo indica un aumento della concretezza e dell'instabilità, che avvicina il comportamento dei pacchetti al valore ideale.

A livello di classe, vengono segnalate 3 violazioni, nella classe Controller, riguardanti le seguenti metriche: fat, fields, eloc. La metrica fat è calcolata come il numero di archi presenti nel grafo delle dipendenze tra tutti i metodi della classe. Il suo valore è 120, il quale risulta essere critico e quindi potrebbe comportare dei problemi alla funzionalità del software. Questo lo si può vedere dal Traffic Light Rating, un grafico che suddivide l'intervallo di valori metrici accettabili e inaccettabili in subrange verdi (30 è perfetto, 60 va bene), gialli (61 fino a 120 critico) e rosse (sopra 120 è davvero pessimo e 240 è il peggiore). L'obiettivo di questa metrica è quello di limitare il più possibile la dimensione del codice riducendo il numero dei metodi presenti. Non esiste un valore standard per questa metrica, ma tutti dev'essere commensurato all'entità del progetto. La metrica fields, corrisponde al numero di attributi di una classe e il suo valore 44 risulta essere pessimo, essendo nella porzione rossa (compresa tra 41 e 80) del grafico Traffic Light Rating. La metrica eloc, estimated lines of code, si riferisce al numero di codice sorgente (senza importazioni, commenti o righe vuote) che compongono un artefatto, l'approssimazione assume alcune regole di stile di codifica ed è più significativa di un valore esatto, poiché quest'ultima dipende dagli stili di codifica dei singoli sviluppatori, il che rende i valori meno confrontabili. Il suo valore 344 è critico perché fa parte della zona del grafico gialla (in questo caso compresa tra 301 a 400). A mio parere, queste metriche sono state violate perché la classe Controller oltre a possedere la gui, esegue molte operazioni. Un modo per rifattorizzare il codice, per risolvere le violazioni, è dividere la classe Controller in più classi, in questo modo si ridurrebbero il numero di attributi e la diminuzione del codice porterebbe alla riduzione delle dipendenze tra tutti i metodi.

DISALLINEAMENTI CODICE-DESIGN

Nel class diagram viene descritta la classe Gui, collegata alla classe Controller, che però nel codice non compare. Inoltre la classe Controller presenta nel codice, delle dipendenze alla classe Util non presente nel design. Quest'ultima classe, Util, presenta nel codice dipendenze alla classe Magazzino, alla classe Vendita, alla classe PagamentoDebito, alla classe Persona; dipendenze non presenti nel class diagram.