GanttProject – Complexity Metrics

**O que é uma métrica de complexidade?**

Uma métrica de complexidade trata de calcular a complexidade do código a partir do número de decisões que o mesmo tem de fazer.

**Porquê calcular a métrica de complexidade?**

Um código complexo tende a ser mais incompreensível e provavelmente contém mais erros, tornando o tempo de desenvolvimento e de teste mais lento. Sendo assim, promove-se o uso de código mais simples para evitar tais situações e, para tal, usa-se esta métrica para poder estudar casos em que a complexidade tem valores alarmantes para poder reescrever o código complexo ou dividi-lo em porções mais simples.

Algo a ter em conta é a correlação que existe entre esta métrica e a métrica de linhas de código. Um programa com um maior número de linhas de código tende a ser mais complexo, porém existem casos em que isto não acontece, e, portanto, recomenda-se a conciliação destas duas métricas para um melhor estudo do código.

**Como são feitos os cálculos?**

A métrica mais utilizada para calcular a complexidade é a v(G), também nomeada de Cyclomatic Complexity (CC). A mesma é calculada da seguinte forma:

V(g) = number of decisions + 1

Ou seja, a cada decisão tomada por uma “conditional statement” (if's, while's, for's, do's, switch cases, catches, &&’s e ||’s) é adicionado o mesmo número de decisões ao valor de V(g).

O valor mínimo é 1, não existindo máximo, porém quanto maior for o valor de V(g) maior é a complexidade do método.

Outras métricas de complexidade como iv(G), ev(G), CogC variam ligeiramente de v(G), testando a complexidade do design do método, a estrutura do método e a compreensibilidade do método, respetivamente.

**Como é que esta métrica se aplica ao GanttProject?**

Aqui temos o histograma completo que demonstra a relação de V(G) por método em todo o programa:

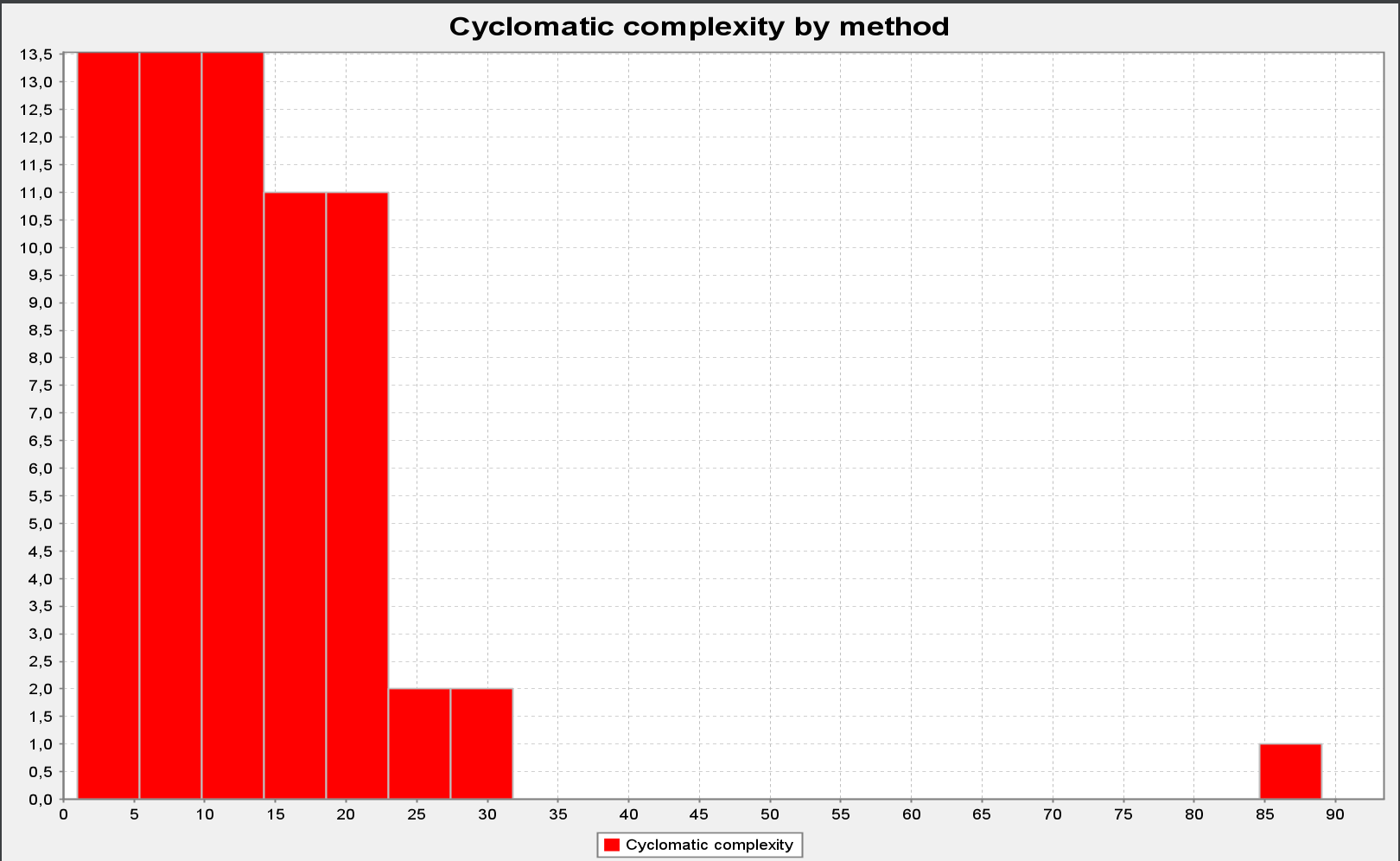
Uma imagem com mesa

Descrição gerada automaticamente

Como podemos ver a maioria dos métodos do programa apresentam valores entre 1 e 5, que são valores “normais” ou “não alarmantes”.

A maioria destes valores pertencem a 1, visto que a média total é 1,82, o que seria de esperar, porque a maioria do métodos não têm que necessariamente fazer escolhas.

Encontramos também uma quantia de métodos com valores entre 5 e 15 que também não são particularmente alarmantes, comparativamente aos valores que se podem encontrar ao aproximar da escala vertical:



Com mais detalhe, podemos encontrar valores entre 15 e 35, e até mesmo entre 85 e 90. Estes valores são muito alarmantes e provam que existem métodos com demasiada complexidade que deveriam ser simplificados ou divididos em métodos mais simples, visto que dificultam a leitura do código, assim como a testagem do programa. É provável encontrar code smells do tipo Large Method/Large Class.

**Como seria um exemplo particular de um método demasiado complexo?**

O método startElement(String, String, String, Attributes) localizado na classe GanttOptions.java é um exemplo de um método demasiado complexo. O seu valor de v(G) é 89 e isto deve-se a uma grande quantidade de if’s e else if’s encontrados no método que se encontra entre as linhas 512 e 721, provando então que se trata de um Large Method localizado numa Large Class.

O método é praticamente ilegível e impossível de se testar e, numa refatorização do código, tanto o método como a classe em que se localiza deveriam ser simplificados.

(A classe encontra-se em: ganttproject/src/net/sourceforge/ganttproject/GanttOptions.java)