



UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

Métodos Formais em Engenharia de Software

Análise e Teste de Software

Métricas e Smells para C++ e MSP
(1ª Fase)

Carlos Sá - A59905
Filipe Oliveira - A57816
Sérgio Caldas - A57779

Resumo

O presente documento constitui o relatório da primeira fase do projeto desenvolvido no âmbito da UCE de Análise e Teste de Software.

Este projeto consiste no desenvolvimento de um catálogo de métricas para a linguagem C- e MSP. Assim, pretende-se definir um catálogo de métricas que permita avaliar código fonte escrito na linguagem C- e MSP. Tal será feito com o recurso ao **Tom** - uma linguagem para manipulação de árvores e **Gom** que permite a representação da árvore numa especificação algébrica. Através destas métricas poderemos extrair indicadores que permitam localizar pontos de *correção*.

Entre as várias métricas temos, por exemplo, as que vão contabilizar o número de linhas de código, o número de funções, ciclos, chamadas a funções recursivas entre outras.

Com o presente catálogo pretende-se localizar *bad smells* em C- em função de valores dessas métricas. Baseando-nos num conjunto de bons programas em C- definem-se o valor de referência para algumas dessas métricas. Caso seja analisado um programa que não siga esse valor de referência então um "bad smell" é localizado (por exemplo, uma função com muitas linhas). Ao longo deste documento explicaremos como foram criadas essas métricas e como foram utilizadas para um conjunto de programas. O projeto pode ser acedido através do seu respectivo repositório **GitHub** aqui.

1 Métricas de Software

Abaixo segue um conjunto de métricas/contadores implementadas pelo grupo até ao final da primeira fase do projeto:

```
Integer numberFunctions;
HashMap <String,Integer> functionsNumberParameters;
```

1
2

- Contar o número total de funções do código;
- Contar o número de parâmetros das funções;

2 Trabalho Futuro

Como trabalho futuro na próxima fase, o grupo tem como objectivo implementar os seguintes métodos/contadores:

```
HashMap <String,Integer> cyclomaticComplexityMap;
HashMap <String,Integer> LongMap;
Integer totalLinesOfCode;
```

1
2
3

- Calcular a complexidade ciclomática para as diferentes funções;
- Calcular o comprimento das funções (**LongMap**);
- Contabilizar o número total de linhas de código;

Denote que, caso consigamos implementar os métodos/métricas anteriormente definidas, pretendemos calcular mais métricas que permitam avaliar padrões de código de maior complexidade;

3 Dificuldades e limitações

Apesar desta fase ser ainda uma fase muito preliminar do projecto, para obter os resultado incluídos neste relatório o enfrentou um conjunto de várias dificuldades que limitou a progressão do trabalho.

A primeira dificuldade está relacionada com o código fornecido pela equipa docente. Precisamos de despende de muito tempo para conseguir perceber o programa e perceber como é definiríamos as métricas utilizando o respectivo código. O facto de nunca termos utilizado Tom e Gom antes desta unidade curricular poderá estar na origem dessas dificuldades.