

EasyCyclomaticComplexity Tool: Uma ferramenta de apoio ao ensino da métrica de Complexidade Ciclomática de McCabe

Bruno César Silva Vital, Vitor Gomes Miranda, Marco Antônio Pereira Araújo

Faculdade Metodista Granbery
Bacharelado em Sistemas de Informação
Rua Batista de Oliveira, 1145 – Granbery, Juiz de Fora – MG – 36010-530
{bvital, vgmiranda}@si.granbery.edu.br, maraujo@granbery.edu.br

Resumo. *EasyCyclomaticComplexity (ECC) é uma ferramenta Web com o propósito de apoiar o ensino da métrica de Complexidade Ciclomática de McCabe, que calcula a quantidade de caminhos independentes a partir do código fonte. Por meio da ferramenta, algoritmos são inseridos e o resultado da Complexidade Ciclomática (CC) é calculado, bem como o grafo associado é desenhado, de forma a apoiar o ensino da técnica.*

Abstract. *EasyCyclomaticComplexity (ECC) is a web tool in order to support the teaching of the metric of McCabe Cyclomatic Complexity, which calculates the number of independent paths from source code. Through the tool, algorithms are inserted and the result of complexity Cyclomatic (CC) is calculated and the associated graph is drawn, so as to assist the teaching of the technique.*

1. Introdução

A métrica de Complexidade Ciclomática (CC), ou complexidade condicional, proposta por McCabe (1976), é uma das mais usadas entre as métricas associadas a código fonte, sendo originária do paradigma procedimental, ou estruturado, e tendo como referência a teoria dos grafos com o propósito de fornecer a medida quantitativa de complexidade lógica de um algoritmo. O valor encontrado utilizando esta métrica caracteriza o número de caminhos de execução independentes em um algoritmo e, em decorrência, pode-se utilizar esse parâmetro para definir o número de casos de testes que devem ser implementados, possibilitando que todos os caminhos possam ser executados ao menos uma vez.

Segundo Vandoren (1997), a CC pode ser aplicada em várias áreas, como: (1) análise de risco no desenvolvimento de código, onde são coletadas medidas sobre o código fonte desenvolvidas para antecipar e mitigar riscos; (2) análise de risco para manutenção, na qual se mede a complexidade antes e depois de uma mudança, para poder assim avaliar os riscos de mudança; (3) planejamento de testes, nos quais os pontos do programa com maior complexidade precisam de maior esforço durante a fase de teste.

Com base no SEI (*Software Engineering Institute*) o número da CC pode determinar os riscos de um algoritmo, sendo (i) um código fonte com CC de 1 a 10 é classificado como simples, fácil de testar e de baixo risco, (ii) com a CC entre 11 a 20 tem-se a complexidade e risco moderados, (iii) com a CC entre 21 e 50 a função é

considerada complexa e de alto risco, e (iv) algoritmos com a CC superior a 50 é classificado como muito difícil de testar e com risco muito alto.

Com a preocupação em criar algoritmos tendo a CC reduzida e com o objetivo de possibilitar um maior dinamismo na exposição do assunto em aulas de Engenharia de Software, a *ECC Tool* tem por objetivo apoiar o ensino da métrica de Complexidade Ciclométrica de McCabe. A ferramenta dispõe de uma interface Web e possibilita o aprendizado de maneira mais atrativa, sendo possível ao usuário inserir códigos e estimar valores para esta métrica. A *ECC Tool* possibilita a geração de um grafo relativo a um algoritmo previamente inserido, facilitando o cálculo da métrica, e seu entendimento.

A ferramenta de apoio *ECC Tool* tem como objetivo principal ser um recurso adicional para o ensino da métrica CC, com o propósito de estimular a curiosidade e prover uma motivação para o aprendizado, tendo em vista que o usuário terá o *feedback* de seus cálculos.

Este artigo contém 2 seções além desta introdução. Na seção 2 será apresentado o funcionamento da ferramenta, e na seção 3 as conclusões e trabalhos futuros.

2. *ECC Tool*

A Figura 1 apresenta a tela inicial da ferramenta, na qual um usuário pode inserir um algoritmo de acordo com a linguagem selecionada. Como a ferramenta se propõe a ser um instrumento de auxílio ao ensino e aprendizagem do cálculo da métrica, também é solicitado ao usuário que informe o suposto valor da complexidade ciclométrica do algoritmo em questão.

Após inseridos os dados, a *ECC Tool* irá calcular a complexidade e verificar se o valor preenchido pelo usuário está correto. O usuário também dispõe do recurso de geração do grafo do algoritmo automaticamente, com o intuito de facilitar a visualização e análise da complexidade do algoritmo. Com o grafo desenhado, o usuário poderá identificar vértices, arestas e áreas. Sendo que as arestas do grafo são direcionais, identificando os caminhos de execução do algoritmo.

Conforme Pressman (2006), a CC pode ser calculada de três formas: (1) pelo número de regiões do grafo de fluxo, (2) através da fórmula $V(G) = E - N + 2$, onde $V(G)$ é a complexidade ciclométrica, G é representa o grafo de fluxo, E é representa a quantidade de arestas no grafo e N é a representa a quantidade de ramos no grafo, e (3) através da fórmula $V(G) = P + 1$, onde P representa a quantidade de vértices predicativos, ou seja, que representam estruturas condicionais. A apresentação gráfica do grafo referente ao algoritmo possibilita ao usuário visualizar as formas como a complexidade ciclométrica de um algoritmo é calculada, conforme pode ser verificado na Figura 2, onde também os caminhos de execução do algoritmo são exibidos, facilitando o planejamento de casos de teste que percorram todos os seus caminhos (DELAMARO *et al.*, 2007). Os vértices são numerados em relação às expressões condicionais encontradas no algoritmo que, através de outra funcionalidade da ferramenta, pode ser exibido com comentários ao final de cada linha identificando o número do vértice no grafo.

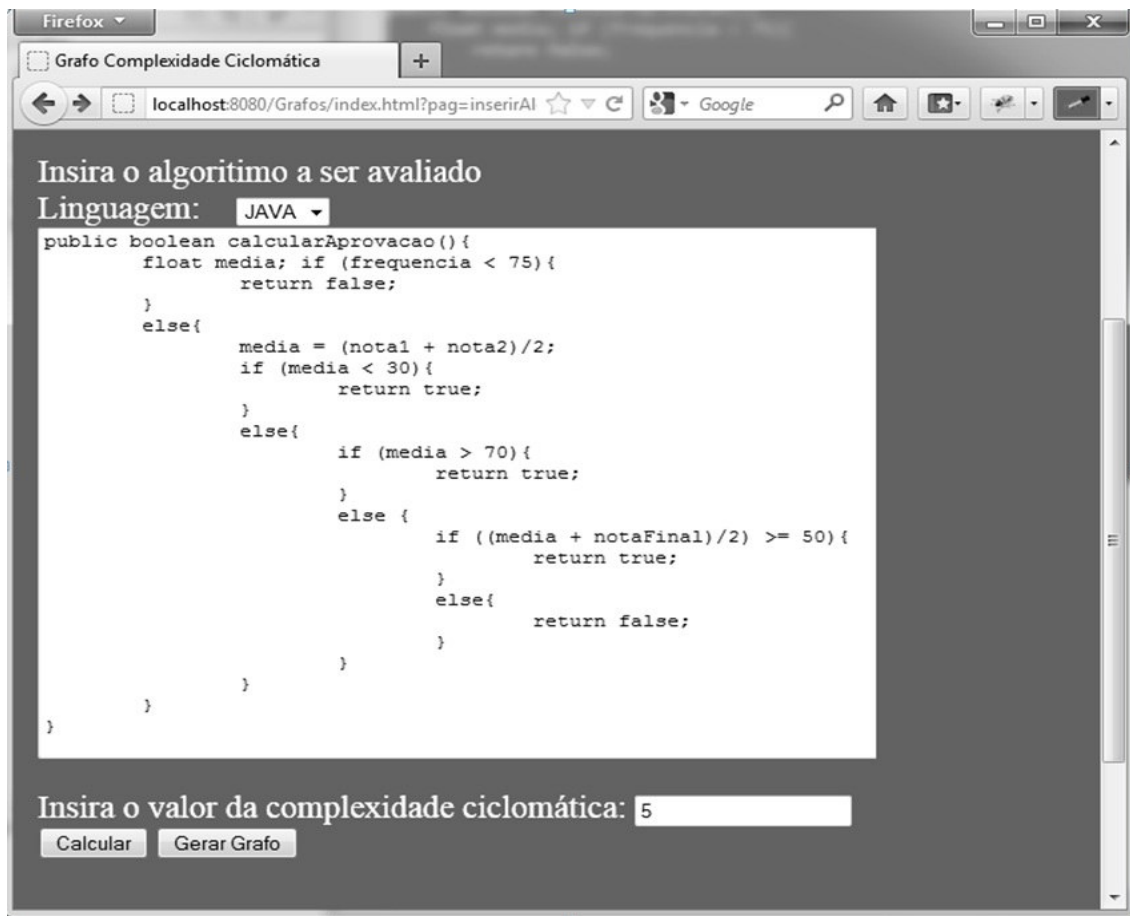


Figura 1. Inserção do algoritmo

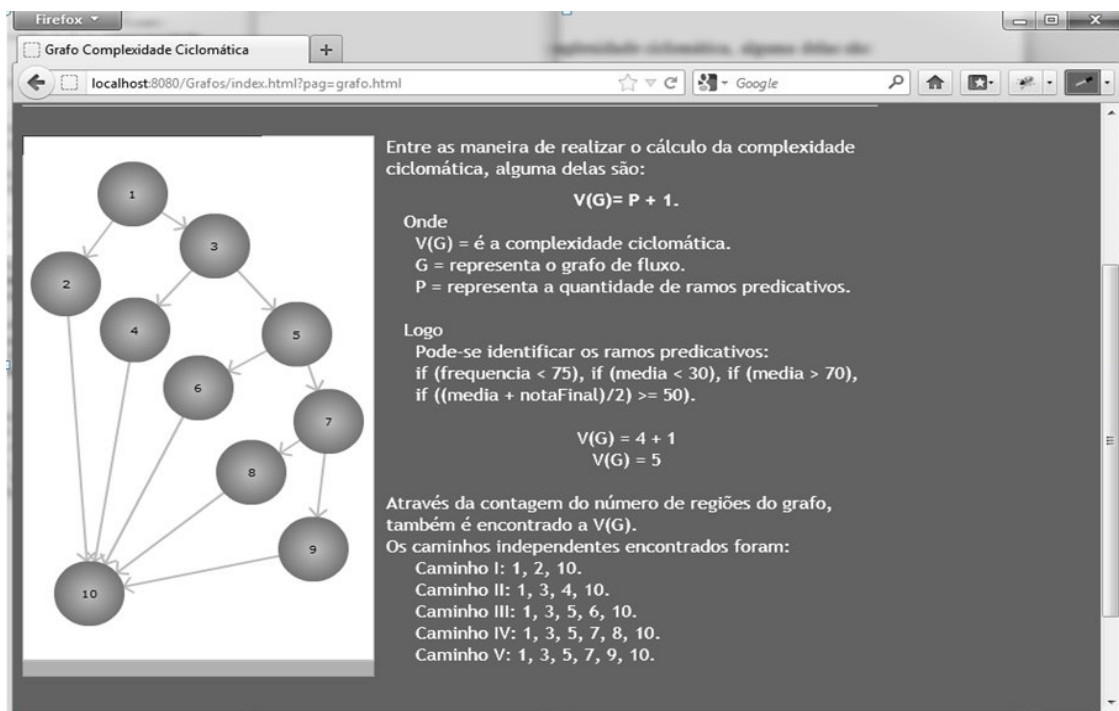


Figura 2. Geração do grafo e explicação da métrica

3. Conclusões e Trabalhos Futuros

Como próximo passo da pesquisa, a ferramenta *ECC Tool*, será utilizada em turmas de graduação do curso de Bacharelado em Sistemas de Informação da Faculdade Metodista Granbery, na disciplina de Verificação, Validação e Teste de Software, para que seja possível avaliar experimentalmente o desempenho dos alunos no aprendizado da técnica com o apoio da ferramenta. Essa disciplina é obrigatória no currículo do curso, e o ensino da métrica de Complexidade Ciclométrica é realizado com apoio restrito de ferramentas, que apenas exibem o resultado final do cálculo da métrica para um determinado algoritmo numa linguagem de programação específica. É importante salientar que não foi encontrada na literatura técnica uma ferramenta livre e que agregue simultaneamente as funcionalidades previstas na ferramenta apresentada neste trabalho.

A avaliação experimental do estudo é de grande importância, de forma a avaliar a efetividade do apoio da ferramenta proposta ao processo de ensino-aprendizagem da métrica de Complexidade Ciclométrica. O planejamento do estudo prevê a divisão aleatória de uma turma da disciplina de Verificação, Validação e Teste em dois grupos, onde o primeiro terá o conteúdo apresentado da forma atual, e o segundo com a utilização da ferramenta. Os dois grupos serão posteriormente submetidos a uma verificação de conhecimento sobre o assunto, e os resultados serão avaliados mediante análise estatística apropriada. Dessa forma, será possível avaliar a efetividade da ferramenta para aquela turma em particular. Repetições do estudo experimental, em diferentes turmas, poderão consolidar os resultados obtidos, e apoiar a evolução da ferramenta.

Como novas funcionalidades previstas para a ferramenta, será possível que o usuário selecione a opção de utilizar códigos já previamente cadastrados, podendo ser apresentados de forma aleatória ou por ordem de complexidade do algoritmo.

Além disso, a ferramenta não se limitará a uma linguagem de programação específica. Os algoritmos poderão ser escritos nas linguagens Java, PHP, Pascal, C, Ruby, entre outras, sendo que as duas primeiras já estão disponíveis.

Espera-se que este trabalho possa incentivar o desenvolvimento de ferramentas para o auxílio ao ensino de questões relacionadas à Engenharia de Software, sendo atualmente um tema pouco explorado e de vital importância para a qualidade de produtos e processos de desenvolvimento de software, bem como apoiar o ensino e aprendizagem da métrica de Complexidade Ciclométrica de McCabe.

Referências

- DELAMARO, M. *et al.*, 2007. Introdução ao Teste de Software. Ed. Campus.
- MCCABE T. J., 1976. Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric, Computer Systems Laboratory National Institute of Standards and Technology - Gaithersburg, MD.
- PRESSMAN, R., 2006. Engenharia de Software. 6. ed. Rio de Janeiro: McGraw-Hill.
- VANDOREN, E., 1997, "Cyclomatic Complexity", Web Publication. Disponível em: <http://www.sei.cmu.edu/library/abstracts/reports/97hb001.cfm>. Acessado em: 06/05/2012.