Identificando Commits Pure e Floss-Refactoring e seu Impacto na Qualidade de Código

Gabriel Lacerda[1], Everton Alves [2]

RESUMO

Durante o ciclo de desenvolvimento de software, é comum a realização de edições no código para adição de novas funcionalidades ou melhorias de aspectos estruturais e não funcionais do software. Essas edições são classificadas como evolutivas ou refatoramentos, sendo este último um processo essencial para melhorar a qualidade do código e sua manutenção. O refatoramento reorganiza o código de forma a torná-lo mais legível, modular e eficiente, reduzindo a complexidade do software e facilitando sua evolução. Existem duas abordagens principais para o refatoramento: Pure Refactoring e Floss Refactoring. O Pure Refactoring visa apenas melhorar o código sem alterar sua funcionalidade, tornando-o mais fácil de entender e manter. Já o Floss Refactoring busca melhorar tanto a qualidade do código quanto a funcionalidade do software. Neste projeto desenvolvemos uma uma estratégia para analisar e classificar os commits de repositórios GIT entre as categorias "pure" e "floss-refactoring", permitindo estudos sobre o impacto dessas abordagens na qualidade do código dos projetos.

Palavras-chave: refatoramento, Pure Refactoring, Refactoring

[1]Aluno do curso Ciência da Computação, Centro de Engenharia Elétrica e Informática, UFCG, Campina Grande, PB, e-mail: gabriel.lacerda@ccc.ufcg.edu.br
[2]Doutor, Professor,UASC, UFCG, Campina Grande, PB, e-mail: everton@computacao.ufcg.edu.br

Identifying Pure Commits and Floss-Refactoring and their Impact on Code Quality.

Gabriel Lacerda[3], Everton Alves [4]

ABSTRACT

During software development, developers often make code edits to add new features or improve structural and non-functional aspects of the software. These edits are classified as evolutionary or refactoring, with the latter being an essential process to improve code quality and maintenance. Refactoring reorganizes code to make it more readable, modular, and efficient, reducing software complexity and facilitating its evolution. There are two main approaches for refactoring commits: Pure Refactoring and Refactoring Floss. Pure Refactoring aims to improve code without altering its functionality, making it easier to understand and maintain. Refactoring Floss, on the other hand, seeks to improve both code quality and software functionality. In this project, we propose a strategy to automatically analyze and classify commits into "pure" and "floss-refactoring" categories, allowing future studies on the impact of these types of refactoring commits on the projects code quality.

Keywords: refactoring, Pure Refactoring, Refactoring Floss.

[3]Aluno do curso Ciência da Computação, Departamento de <Nome do Departamento>, UFCG, Campina Grande, PB, e-mail: gabriel.lacerda@ccc.ufcg.edu.br
[4]Doutor, Professor,UASC, UFCG, Campina Grande, PB, e-mail: everton@computacao.ufcg.edu.br