VIII CONGRESSO DE INICIAÇÃO CIENTÍFICA DA UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

Pró-Reitoria de Pesquisa e Extensão

CNPq

PIBIC/CNPq/UFCG-2011

ANALISANDO REFATORAMENTOS EM REPOSITÓRIOS DE SOFTWARE

Solon B. de Aguiar Neto[1], Rohit Gheyi[2]

RESUMO

Análises de refatoramentos feitas em repositórios de software hoje em dia ocorrem manualmente. Essa atividade é tendenciosa a erros, demorada e não se aplica ao estudo de toda história do software. Esse tipo de conhecimento pode ajudar gerentes de projeto não só a prever erros e mudanças, como também entender de que maneira se dão e qual a periodicidade dos refatoramentos durante o processo de desenvolvimento. Nesse trabalho apresentamos uma técnica automática de análise de frequência, granularidade e escopo de refatoramentos em repositórios de software. Essa técninca é baseada no SAFEREFACTOR, uma ferramenta que analisa transformações através da geração de testes para detecção de mudanças comportamentais – ela já encontrou um número considerável de bugs em implementações de refatoramentos de IDEs como Eclipse e Netbeans. Utilizamos a técnica para analisar cinco projetos open source Java (JHotDraw, ArgoUML, SweetHome 3D, HSQLDB e JEdit). A partir de mais de 40723 versões de software, 39 anos de desenvolvimento, 80 desenvolvedores e 1.5 TLOC descobrimos que 73% das versões não são refatoramentos. Em relação aos prováveis refatoramentos, 63.83% são Low level e 71% são locais. O tamanho médio dos refatoramentos é de 45 LOC. Nossos resultados mostram que refatoramentos são frequentes. Isso motiva desenvolvedores a melhorar a qualidade das ferramentas de refatoramentos e automatizar novos tipos destes.

Palavras-chave: Melhoria design código, repositórios de software, preservação de comportamento

ANALYZING REFACTORINGS ON SOFTWARE REPOSITORIES

ABSTRACT

Currently analysis of refactoring in software repositories is performed manually, which is time-consuming, error-prone, and lacks scalability in full historical data. Such analysis is useful to understand how and when refactorings actually occur throughout development. In this work, we propose a fully automatic technique to analyze refactoring frequency, granularity and scope in software repositories. It is based on SAFEREFACTOR, a tool that analyzes transformations by generating tests to detect behavioral changes – it has found a number of bugs in refactoring implementations within some IDEs, such as Eclipse and Netbeans. We use our technique to analyze five open source Java projects (JHotDraw, ArgoUML, SweetHome 3D, HSQLDB and JEdit). From more than 40,723 software versions, 39 years of software development, 80 developers and 1.5 TLOC, we have found that: 73% of versions are not refactorings. Regarding the likely refactorings, 63,83% of them are Low level, and 71% are Local. The average size of a refactoring is 45 LOC. Our results show refactorings are frequent, which motivates developers to improve existing refactoring tools and automate new ones.

Keywords: Refactorings, software repositories, behavioral preservation

[1] Aluno do Curso de Ciência da Computação, Centro de Engenharia Elétrica e Informática, Departamento de Sistemas e Computação, UFCG, Campina Grande, PB, E-mail: solonban@lcc.ufcg.edu.br

[2] Ciência da Computação, Professor. Doutor, Centro de Engenharia Elétrica e Informática, Departamento de Sistemas e Computação, UFCG, Campina Grande, PB, E-mail: rohit@dsc.ufcg.edu.br