



UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Autorizada pelo Decreto Federal nº 77.496 de 27/04/76
Recredenciamento pelo Decreto nº 17.228 de 25/11/2016



PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
COORDENAÇÃO DE INICIAÇÃO CIENTÍFICA

XXVI SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS **SEMANA NACIONAL DE CIÊNCIA E TECNOLOGIA - 2022**

INVESTIGANDO A RELAÇÃO ENTRE CARACTERÍSTICAS DE DESENVOLVEDORES E ANOMALIAS DE CÓDIGO

Daniel Lucas Alves Ferreira de Jesus¹; José Amancio Macedo Santos²

1. Bolsista PROBIC, Graduando em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: daniel.lucasafj@gmail.com
2. Orientador, Departamento de Tecnologia, Universidade Estadual de Feira de Santana, e-mail: zeamancio@uefs.br

PALAVRAS-CHAVE: code smell; software; desenvolvedor.

INTRODUÇÃO

A variedade de solicitações de manutenção sobre diferentes elementos de código-fonte frequentemente desafia os desenvolvedores de software. Esse desafio normalmente resulta da complexidade estrutural do código-fonte exigindo um esforço considerável de leitura e compreensão para realizar até mesmo tarefas simples de desenvolvimento, Bennett & Rajlich (2000). Para mitigar esses esforços, uma prática-chave aborda continuamente a identificação e o combate à incidência de code smells, Palomba et al. (2018). Code smells são conhecidos como indicadores de problemas mais profundos dentro de um código-fonte, comumente introduzido devido à negligência das boas práticas de programação. O objetivo principal dessa pesquisa é compreender quais características de desenvolvedores estão relacionados com a implementação de code smells, ou anomalias de código, nos código-fonte do projeto.

MATERIAL E MÉTODOS

Para esse estudo foi utilizado o Dataset disponibilizado por Valentina et. al. (2019), que foi criado com o objetivo de permitir que os pesquisadores trabalhem em um conjunto de dados e comparar seus resultados. Os dados dessa base de dados foram obtidos através de repositórios clonados do GitHub, ela dispõe de 33 projetos Java da Apache Software Foundation, com 78 mil commits e foram submetidos às ferramentas de análise SonarQube e Ptidej, e que atenderam os seguintes critérios: ser desenvolvido em Java, ter mais de três anos de mercado, ter mais de 500 commits, mais de 100 classes e usar o sistema de rastreamento de problemas do Jira com mais de 100 problemas.

Com o dataset escolhido o próximo passo foi verificar a consistência dos dados, para ter certeza de um trabalho de análise consistente, para isso foi utilizado do método de mineração de dados, onde foram feitas várias análises para a validação dos dados. Em todo o estudo foi utilizado scripts sql para a obtenção dos dados da base de dados e a linguagem R para a manipulação desses dados. Depois dos dados validados, criamos uma base de dados com somente os dados que seriam necessário para nossa pesquisa.

RESULTADOS E DISCUSSÃO

Nosso objetivo é compreender como os desenvolvedores e suas características afetam a qualidade de um software, para isso foram selecionados os seguintes atributos de desenvolvedores: nome do desenvolvedor, número de linhas adicionadas e removidas, quantidade de dias trabalhados no projeto, quantidade de commit realizado. Nossa análise foi realizada em cima dos fatores de participação e experiência, para obter os resultados foi feito o levantamento de algumas métricas baseadas em trabalhos já realizados. Consideramos neste trabalho como métrica code churn, quantidade de dias do desenvolvedor (xp_days) e commit.

Para evitar dados inconsistentes foi trabalhado somente commits que causaram alguma alteração no código, chamados commits de mudança. Para isso foram selecionados os autores e seus respectivos commits e adicionados na tabela COMMIT_MUDANÇAS_2 (CM2) figura 1, contendo os campos nome do projeto, nome do autor, commitHash de mudança e o commitHash original. As informações de anomalias de código no dataset são oriundas das análises das ferramentas SonarQube e Ptidej, para este trabalho foram considerados apenas as anomalias de código, os code smells, detectados apenas pela ferramenta Ptidej.

Como resultado da análise conseguimos dados de 2436 desenvolvedores, 345 code smells detectados pelo Ptidej com 12 tipos de code smells diferentes, são eles: anti singleton, base class abstract, complex class, long method, lazy class, speculative generality, refused parent bequest, long parameter list, class data private, spaghetti code, large class, many field attributes not complex.

Descrição do Dataset criado

Para facilitar as análises o dataset criado contém 15 tabelas, sendo 12 tabelas figura 2 referente aos tipos de code smells detectados pelo Ptidej, nessas tabelas contém os campos nome do projeto, nome do autor e a quantidade de smell implementado pelo desenvolvedor. A tabela QTD_COMMITS figura 3, referente a quantidade de commit de cada desenvolvedor, contendo os campos author (nome do autor), projectID (nome do projeto) e n_commits. A tabela EXPERIÊNCIA_DEV, referente a experiência do desenvolvedor, contém informações sobre a quantidade de dias e semanas que o desenvolvedor trabalhou no projeto, esta tabela contém os campos nome do autor, xp_in_days (dias), xp_in_weeks (semana). Por fim a tabela BASE_ULTIMATE, que contém a relação dos desenvolvedores com todos os tipos de code smells por eles implementado, nela contém os campos referente ao nome do projeto, nome do desenvolvedor, quantidade de linhas editadas (code churn), quantidade total de code smell implementado e a quantidade de cada smell específico implementado.

projectID	author	commit_hash	commit_hash_changes
Filtro	Filtro	Filtro	Filtro
accumulo	Keith Turner	e0880e263...	e0880e263e4bf8662ba...
accumulo	Billie Rinaldi	e8774c5ec3...	e8774c5ec3a35e042f3...
accumulo	Keith Turner	2032ebbd0...	2032ebbd0ed90734da...
accumulo	Billie Rinaldi	de297d493...	de297d4932e08625a5...
accumulo	Billie Rinaldi	34efaae876...	34efaae87639a83b60f...
accumulo	Keith Turner	7e9bba037...	7e9bba037c4fbd70c7e...

Figura 1: Recorte da Tabela COMMIT_MUDANCAS_2

▶	Baseclass_abstract
▶	Class_data_private
▶	Complex_class
▶	Large_class
▶	Lazy_class
▶	Long_method
▶	Long_parameter_list
▶	Many field attributes not com...

Figura 2: Recorte das 12 tabelas referente aos tipos de smells detectados pelo Ptidej

	n_commits	author	projectID
	Filtro	Filtro	Filtro
1	56	Adam Fuchs	accumulo
2	13	Adam J. ...	accumulo
3	3	Adam Lerman	accumulo
4	2	Al Krinker	accumulo
5	6	Alex ...	accumulo
6	2	Andrew ...	accumulo
7	1	Andrew ...	accumulo

Figura 3: Recorte da tabela QTD_COMMITS

Para estabelecer a correlação entre as características dos desenvolvedores e code smells, utilizamos o teste de correlação de Spearman. Neste estudo foi possível somente realizar o teste de correlação com a variável code churn, foram realizado três testes, o primeiro teste tabela 1 estamos levando em consideração todos os 2436 desenvolvedores, a tabela 2 estamos levando em consideração apenas os desenvolvedores que foram detectado code smell, no total de 345 desenvolvedores.

Tabela 1: Correlação entre Code Churn e code smells, considerando os 2436 desenvolvedores

Variável	Correlação (rho)	Significância (p-value)
Code churn	0.4373372	2.334284e-114

Tabela 2: Correlação entre Code Churn e code smells, considerando os 345 desenvolvedores com smells detectados pelo Ptidej

Variável	Correlação (rho)	Significância (p-value)
Code churn	0.6825855	1.209077e-48

CONSIDERAÇÕES FINAIS

Através dos dados das tabelas 1 e 2, podemos perceber que existe correlação positiva entre o code churn e os code smells. Neste resumo não foi possível acrescentar a terceira tabela que faz mostra a correlação entre code churn e cada tipo de smell específico, mas a partir desta terceira tabela podemos observar que 4 tipos de code smells apresentaram um maior número de correlação, três desses tipos de smells, complex class, long method, long parameter list, são os que têm maior número de desenvolvedores, totalizando um total de 796 profissionais. Podemos concluir que conforme um desenvolvedor edita mais linhas de códigos no projeto, maiores são as chances de ser implementado code smells, com maiores chances de implementar um dos tipos de smells listados acima. Recomendamos para pesquisas futuras dar continuidade na análise do dataset escolhido, implementar as correlações das métricas não utilizadas e fazer o estudo das anomalias detectadas pela ferramenta SonarQube.

REFERÊNCIAS

LENARDUZZI, Valentina, SAARIMÄKI, Nyyti, and TAIBI, Davide. 2019. The Technical Debt Dataset. In The Fifteenth International Conference on Predictive Models and Data Analytics in Software Engineering (PROMISE'19), September 18, 2019, Recife, Brazil. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3345629.3345630>

BENNETT ,Keith andRAJLICH,Vaclav. 2000. Software Maintenance and Evolution: A Roadmap. (2000).

PALOMBA, Fabio, BAVOTA, Gabriele, DI PENTA, Massimiliano, FASANO Fausto, OLIVETO Rocco, and DE LUCIA, Andrea. 2018. On the diffuseness and the impact on maintainability of code smells: a large scale empirical investigation. Empirical Software Engineering 23, 3 (2018), 1188–1221.