



XXVIII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS
SEMANA NACIONAL DE CIÊNCIA E TECNOLOGIA - 2024

**UMA INVESTIGAÇÃO PRELIMINAR SOBRE A RELAÇÃO ENTRE
ESFORÇO E ANOMALIAS NO CÓDIGO**

Gabriel Cordeiro Moraes¹; José Amancio Macedo Santos²

1. Bolsista – Modalidade Bolsa/PVIC, Graduando em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: gcmorais@ecomp.uefs.br
2. Orientador, Departamento de Tecnologia, Universidade Estadual de Feira de Santana, e-mail: zeamancio@uefs.br

PALAVRAS-CHAVE: qualidade de software; code smell; mineração de repositórios.

INTRODUÇÃO

“Code Smell descreve padrões no código-fonte que podem sinalizar problemas de design, dificultando a manutenção e prejudicando a evolução do software” (Fowler et al., 1999). Detectar e corrigir essas anomalias com antecedência ajuda a reduzir custos, aumentar a eficiência das equipes de desenvolvimento e evita a degradação do código. Entender, detectar e corrigir code smells não é uma tarefa que se limita a manutenção do código; é uma abordagem dinâmica para a qualidade e a vida útil do código. Estudos indicam a relação entre: code smells e a propensão a bugs (Cairo, Carneiro e Monteiro, 2018), complexidade do código (M. Lehman, 1996) e maior esforço de manutenção (Nascimento e Sant’Anna, 2017). Os code smells são como avisos de que algo pode estar errado, portanto, abordar esses indicadores de problemas potenciais, podem ajudar a criar um software mais sustentável, eficiente e fácil de manter.

Inserido em um contexto onde a qualidade é de suma importância para o sucesso de projetos de TI, a identificação de code smells pode desempenhar um papel importante para garantir a qualidade do código. Assim, este estudo explora a relação entre atributos de qualidade do software e code smells. Investigamos como a intensidade do trabalho dos desenvolvedores está relacionada à ocorrência de anomalias no código. Seguindo o trabalho de Shin et al. (2011), utilizamos métricas como “número de linhas editadas” e “número de commits” para representar a intensidade das atividades dos desenvolvedores. Essas métricas foram escolhidas com base em estudos anteriores que ressaltam a importância de relacionar métricas de desenvolvimento com aspectos da qualidade do software.

A questão de pesquisa que orienta a investigação é: “Existe uma relação entre a participação dos desenvolvedores e a ocorrência de code smells?”. Tufano et al. (2015), encontrou que desenvolvedores mais experientes introduzem menos anomalias no código. Já nossos resultados indicam que desenvolvedores mais ativos tendem a inserir menos code smells por commit ou edição de linha. Esse estudo também oferece uma perspectiva adicional aos resultados de Palomba et al. (2018), que investigou a relação entre a incidência de code smells e práticas de desenvolvimento. Ou seja, nossas

descobertas fornecem insights valiosos para acadêmicos e profissionais da área de software, ajudando a aperfeiçoar práticas de programação e a melhorar a qualidade do software.

MATERIAL E MÉTODOS OU METODOLOGIA (ou equivalente)

Os dados para a pesquisa foram extraídos diretamente do repositório do GitHub da versão dois do The Technical Debt Dataset. O conjunto de dados do The Technical Debt Dataset é oriundo da mineração de repositórios de software, apoia estudos na dívida técnica e tem sido utilizado em outros estudos, ou seja, fornece dados sobre vários aspectos da qualidade do software, incluindo code smells.

Conforme apresentado na Figura 1, a estratégia de análise adotada para o estudo foi dividida em duas fases. Na primeira fase, após extraídos os dados minerados dos repositórios de software, os dados foram normalizados, em seguida foi conduzido o teste de normalidade de Mann-Whitney para avaliar a distribuição dos valores amostrais e por fim os testes de correlação foram realizados.

Na segunda fase, os desenvolvedores foram divididos em grupos “HIGH” e “LOW” que representam o impacto causado nos projetos com base no número de commits e no número de linhas editadas. Em seguida foram definidas variáveis que representam a quantidade de code smells por número de commits e por número de linhas de código editadas. Por fim, foram realizados os testes de hipótese com o objetivo de responder a questão de pesquisa citada anteriormente.

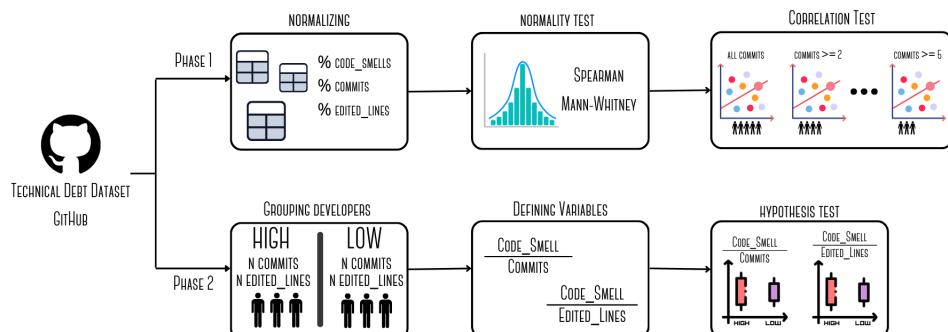


Figura 1: estratégia de análise adotada no estudo.

RESULTADOS E/OU DISCUSSÃO (ou Análise e discussão dos resultados)

Para os testes de correlação de Spearman, os resultados obtidos estão organizados na Tabela 1 e indicam uma forte relação entre code smells e o número de linhas editadas, ou seja, um número significativo de linhas editadas está associado a um aumento na incidência de anomalias nos códigos. Por sua vez, a correlação entre code smells e o número de commits é moderada, indicando que embora menos intensa que o observado em número de linhas editadas é um aspecto que deve ser observado.

Tabela 1: Resultados da correlação de Spearman para code smells.

Métricas	($ r_s $) coefficient	p-value
Code smells e linhas editadas	0,7627	$5,27 \times 10^{-46}$
Code smells e commits	0,5621	$8,26 \times 10^{-13}$

Criamos as variáveis $cspercommits = \frac{codesmells}{commits}$ e $csperlinesedited = \frac{codesmells}{linesedited}$ com o objetivo de quantificar o número de anomalias que cada desenvolvedor insere/mantém no código por commit e por linhas editadas, respectivamente. Conforme a Figura 2, é possível observar a distribuição de code smells por commits associados aos grupos de desenvolvedores com mais (HIGH) ou menos (LOW) commits. Por questões de espaço só apresentamos os resultados para número de commits, mas os resultados da distribuição de número de linhas editadas são semelhantes.

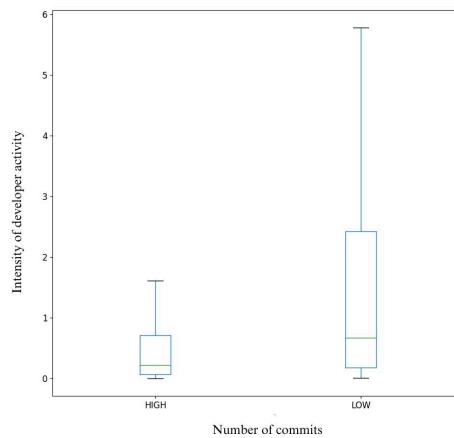


Figura 2: Distribuição de code smells por commit para grupos de desenvolvedores classificados por nível de atividade (HIGH e LOW).

Com base nos dados definimos as seguintes hipóteses:

- Hipótese nula: não há diferença na distribuição de code smells por commit entre grupos de desenvolvedores mais ativos (HIGH) e menos ativos (LOW).
- Hipótese alternativa: há uma diferença na distribuição de code smells por commit entre grupos de desenvolvedores mais ativos (HIGH) e menos ativos (LOW).

Como as amostras não apresentam distribuição normal, adotamos o teste de hipótese de Mann-Whitney e os resultados podem ser observados na Tabela 2.

Tabela 2: Resultados do teste de hipótese.

Variável	Resultado	u-test	p-value
cspercommits	Hipótese nula falsa	16567,0	$9,08e \times 10^{-7}$
csperlinesedited	Hipótese nula falsa	14652,0	0,0038

O p-value obtido foi $9,08e \times 10^{-7}$; portanto, a hipótese nula foi rejeitada, mostrando que os desenvolvedores que fazem menos commits inserem/mantém mais anomalias de código por commit.

CONSIDERAÇÕES FINAIS (ou Conclusão)

Este estudo realizou uma análise do impacto dos desenvolvedores em ações de manutenção e qualidade de software, utilizando como fonte de dados o trabalho denominado “The Technical Debt Dataset Version 2.0” (Lenarduzz; Saarimäki; Taibi, 2019). O conjunto de dados inclui 31 projetos, 749 desenvolvedores e 153.994 commits. A análise apresentada neste trabalho considerou todos os 31 projetos e 317 desenvolvedores associados a algum tipo de code smells e explorou a relação entre code smells, quantidade de commits e quantidade de linhas editadas, fornecendo informações sobre a influência dos desenvolvedores nos padrões de qualidade do software.

Os resultados indicam que code smells podem ser considerados uma métrica importante na medição de qualidade de software e que tem alguma relação com a intensidade da atividade dos desenvolvedores em termos do número de commits e linhas editadas. Foi observada uma correlação entre um maior número de linhas editadas, número de commits e o surgimento de anomalias nos códigos, especialmente entre desenvolvedores menos ativos. Este estudo corrobora com o entendimento de que a estabilidade e a continuidade dos desenvolvedores em projetos específicos contribuem para manter a qualidade do software.

REFERÊNCIAS

- CAIRO, Aloisio; CARNEIRO, Glauco; MONTEIRO, Miguel. The Impact of Code Smells on Software Bugs: A Systematic Literature Review. *Information*, v. 9, n. 11, p. 273, 6 nov. 2018. Disponível em: <https://doi.org/10.3390/info9110273>. Acesso em: 8 set. 2024.
- FOWLER, Martin. Refactoring: Improving the Design of Existing Code. [S. l.]: Addison-Wesley, 1999.
- LANZA, Michele; MARINESCU, Radu. Object-oriented metrics in practice: using software metrics to characterize, evaluate, and improve the design of object-oriented systems. [S. l.]: Springer Science & Business Media, 2007.
- LENARDUZZ, Valentina; SAARIMÄKI, Nyyti; TAIBI, Davide. The Technical Debt Dataset. 15th Conference on Predictive Models and Data Analytics in Software Engineering, 2019.
- M LEHMAN, Manny. Laws of software evolution revisited. European workshop on software process technology, p. 108-124, 1996.
- NASCIMENTO, Rogeres; SANT'ANNA, Cláudio. Investigating the relationship between bad smells and bugs in software systems. Proceedings of the 11th Brazilian Symposium on Software Components, Architectures, and Reuse, p. 1-10, 2017.
- PALOMBA, Fabio et al. On the diffuseness and the impact on maintainability of code smells: a large scale empirical investigation. Proceedings of the 40th International Conference on Software Engineering., p. 482-482, 2018.
- SHIN, Yonghee et al. Evaluating Complexity, Code Churn, and Developer Activity Metrics as Indicators of Software Vulnerabilities. *IEEE Transactions on Software Engineering*, v. 37, n. 6, p. 772-787, nov. 2011. Disponível em: <https://doi.org/10.1109/tse.2010.81>. Acesso em: 8 set. 2024.