

Universidade Federal do Ceará
Pró-Reitoria de Pesquisa e Pós-Graduação
PIBIC 2023/2024 - Edital Nº 02/2023

Investigando o impacto da refatoração dos test smells no código de testes

Resumo

Atualmente, os testes automatizados estão sendo cada vez mais utilizados pelos desenvolvedores para a realização de avaliações de qualidade em um projeto de software. Desse modo, os desenvolvedores devem se preocupar com anomalias conhecidas como test smells. Test smells representam um conjunto de testes mal elaborados, que podem prejudicar os critérios de manutenibilidade e qualidade de um código de teste. Refatoração surge como uma alternativa para a remoção dos test smells, uma vez que estudos já comprovaram a sua eficácia para remover os smells do código de teste. A maioria dos estudos relacionados aos test smells investigam os impactos dos smells na manutenibilidade e legibilidade do código de teste, além de analisarem as percepções dos desenvolvedores sobre os riscos da ocorrência dos test smells em um sistema. Embora passos fundamentais para entender os test smells tenham sido realizados, ainda há uma evidente carência de estudos que avaliem o impacto da refatoração dos test smell nos atributos internos de qualidade, como tamanho, coesão, acoplamento e complexidade, bem como as devidas estratégias de refatoração para remover tais anomalias. Portanto, este projeto tem como objetivo investigar o impacto da refatoração dos test smells nos atributos internos de qualidade e catalogar os test smells mais prejudiciais. Também iremos investigar qual a relação dos test smells com os flaky tests, que são comportamentos inesperados da execução de um teste automático. Para tal, serão analisados: (i) os test smells mais prejudiciais para os atributos internos de qualidade; (ii) as percepções dos desenvolvedores sobre os test smells mais prejudiciais; (iii) investigar as devidas estratégias de refatoração utilizadas para remover tais smells; e, (iv) investigar o impacto dos test smells na ocorrência de flaky tests. Para alcançar esses objetivos serão realizados estudos experimentais em projetos de código aberto com a participação de desenvolvedores. Serão realizadas análises quantitativas e qualitativas a partir da perspectiva dos desenvolvedores.

1. Introdução

A Qualidade de software visa determinar como um projeto de software está projetado e o quão bom é o grau da estrutura do software. Ela pode ser medida por diferentes atributos de qualidade (MALHOTRA e CHUG, 2016). Os atributos de qualidade podem ser classificados em i) Atributos internos de qualidade e ii) Atributos externos de qualidade. Atributos externos de qualidade são aqueles que indicam a qualidade do sistema baseado em fatores que não podem ser mensurados somente utilizando artefatos de software, como por exemplo a manutenibilidade, que depende de um conjunto de fatores externos para que seja avaliada, como o tempo de vida do sistema e o ambiente para o qual o sistema está sendo modificado (AL DALLAL, 2013). Atributos internos de qualidade, tais como tamanho, coesão e acoplamento podem ser medidos com artefatos de software. Quantificar atributos internos de qualidade é muito mais fácil que quantificar atributos externos, por exemplo o tamanho de uma classe pode ser medida através do métrica LOC (Linhas de código) (MORASCA, 2009).

Durante seu desenvolvimento, o software é submetido constantemente a mudanças que podem prejudicar seus atributos internos de qualidade, que são medidos pela análise do software separadamente de seu comportamento, e atributos externos de qualidade, medidos em termos de como o software se relaciona com o ambiente (KAUR; DHIMAN, 2019). Para realizar as avaliações de qualidade adequadas em um sistema, os desenvolvedores recorrem a testes de software (MYERS et al., 2011). Nesse contexto, testes automatizados, implementados via frameworks ou ferramentas de teste, tornaram-se um processo fundamental para melhorar a qualidade do software. Testes automatizados podem auxiliar no processo de identificação e remoção de erros e garantir que o código de produção seja robusto sob diversas condições de uso (CANDEA et al., 2010).

No entanto, a automação do código de teste demanda um esforço equivalente ao que é exercido durante o desenvolvimento do código de produção. Portanto, os desenvolvedores devem ter cuidado com o código de teste, pois está diretamente relacionado ao código de produção (BELLER et al., 2017). Estudos recentes mostram que os desenvolvedores entendem e dão mais importância ao código de produção do que ao código de teste, causando problemas na qualidade dos testes (PALOMBA et al., 2018). Deursen et al. (2001) indicaram que a qualidade do teste de código não era tão alta quanto a do código de produção devido ao fato de o código de teste não ser tão refatorado e estudado quanto o código de produção.

Deursen et al. (2001) introduziram o conceito de test smells, que representam um conjunto de testes mal projetados que podem ser prejudiciais à legibilidade, manutenção e qualidade do código de teste. Desde sua criação, os test smells se tornaram foco de estudos de pesquisadores da área de Engenharia de Software (KIM, 2020). Mesmo após o avanço nas pesquisas relacionadas aos test smells, estudos recentes (SOARES et al., 2022; SPADINI et al., 2020) sugerem que é importante investigar algumas questões sobre os test smells, que são: (i) as principais causas encontradas pelos desenvolvedores durante a refatoração dos test smells; (ii) o impacto da refatoração dos test smells sobre os atributos internos de qualidade; (iii) os test smells são mais prejudiciais para os atributos internos de qualidade e para os desenvolvedores de software; e, (iv) avaliar as estratégias de refatoração disponíveis para remover os test smells.

Com isso, a refatoração de software surge como uma alternativa para a remoção dos test smells. A refatoração de software é um processo que visa melhorar a estrutura interna do código através da transformação do código de modo que o seu comportamento externo permaneça inalterado (FOWLER, 1999). Os test smells podem ser removidos por meio de refatoração de software, e essa remoção pode afetar positivamente a qualidade do código de teste (SPADINI et al., 2022). No entanto, revisões de literatura indicam uma carência de estudos avaliando o impacto que a refatoração dos test smells tem nos atributos de qualidade (BAVOTA et al., 2015).

Flaky tests são casos de testes automatizados que apresentam um comportamento não-determinístico, ou seja, não é possível determinar se o caso de teste irá passar ou falhar a cada execução (ROMANO et al., 2021). Os efeitos danosos da ocorrência de flaky tests foram relatados nos últimos anos por várias empresas, como por exemplo, a relatou Google que 1,5% de todos os testes executados são instáveis e quase 16% de seus 4,2 milhões de testes individuais falham independentemente de alterações no código ou nos testes (MICCO, 2017). Da mesma forma, cerca de 4,6% dos testes de cinco projetos da Microsoft são instáveis (LAM et al., 2020a). Devido à natureza não determinística dos flaky tests, detectá-los é bastante desafiador e tem se tornado uma área de pesquisa cada vez mais ativa (LAM et al., 2020b). Também é desconhecido ainda se o fato do código de testes possuir determinados test smells pode influenciar na ocorrência dos flaky tests.

Dessa forma, este projeto tem como objetivo principal investigar o impacto da refatoração de test

smells na qualidade de software através da identificação dos tipos de test smells que são mais prejudiciais para os desenvolvedores de software e para os atributos internos de qualidade, como coesão, acoplamento, complexidade e tamanho. Os objetivos específicos que pretende-se alcançar no projeto, são: (i) catalogar os test smells mais prejudiciais para os atributos internos de qualidade; (ii) investigar as percepções dos desenvolvedores sobre os test smells mais prejudiciais; (iii) investigar as devidas estratégias de refatoração utilizadas para remover tais smells; e, (iv) investigar o impacto dos test smells na ocorrência de flaky tests. Para alcançar esses objetivos serão realizados estudos experimentais com desenvolvedores refatorando códigos de testes de projetos de código aberto. Serão realizadas análises quantitativas e qualitativas a partir da perspectiva dos desenvolvedores.

2. Perguntas de Partida

Neste cenário surgem as seguintes questões de pesquisa:

- Qual o impacto da refatoração dos test smells sobre os atributos internos de qualidade?
- Quais test smells são mais prejudiciais para os atributos internos de qualidade e para os desenvolvedores?
- Qual o impacto dos tests smells na ocorrência de flaky tests?

3. Hipóteses

Baseado nas questões de pesquisa a seguinte hipótese será verificada:

- É possível que existam test smells mais prejudiciais na qualidade do código de testes.
- O fato de existir determinados tipos de test smells no código, influencia no aparecimento de flaky tests.

4. Objetivos

Objetivo Geral:

Investigar o impacto da refatoração de test smells na qualidade de software através da identificação dos tipos de test smells que são mais prejudiciais para os desenvolvedores de software e para os atributos internos de qualidade.

Objetivos específicos:

- Identificar os test smells que mais afetam os atributos de qualidade.
- Investigar as principais dificuldades encontradas pelos desenvolvedores durante a refatoração dos test smells.
- Catalogar os test smells são mais prejudiciais no ponto de vista dos desenvolvedores.
- Investigar se os test smells influenciam na ocorrência de flaky tests.
- Catalogar os test smells que influenciam no aparecimento de determinados tipos de flaky tests.

5. Materiais e Métodos

Para atender aos objetivos do projeto foram definidas cinco etapas de trabalho: (i) selecionar projetos de código aberto; (ii) detectar os test smells; (iii) medir os atributos interno de qualidade; (iv) selecionar e treinar os desenvolvedores; (v) refatorar os test smells; (vi) identificar flaky tests; e, (vii) catalogar test smells.

A etapa (i) consiste em selecionar projetos de código aberto para identificar e refatorar os test smells. Para a seleção de tais sistemas, será levado em consideração o trabalho desenvolvido por Pecorelli et al. (2021), que identificou um conjunto de sistemas de código aberto, visando analisar a relação entre os test smells e a qualidade de software.

A etapa (ii) visa a detecção dos tests smells utilizando a ferramenta JNose Test (VIRGÍNIO et al., 2020). Os test smells serão detectados nos projetos selecionados por meio da ferramenta JNose. Um subconjunto dos test smells identificados será utilizando para serem refatorados pelos desenvolvedores.

Na etapa (iii) visa medir os atributos internos de qualidade. Após a detecção de todos os test smells, serão medidos todos os atributos internos de qualidade, utilizando a ferramenta Understand. Os atributos a serem medidos são: coesão, acoplamento, complexidade e tamanho.

Na etapa (iv), é visa selecionar e treinar os desenvolvedores para refatorar os test smells. Antes de iniciar a refatoração dos test smells, é necessário selecionar os desenvolvedores responsáveis pela refatoração. Após a seleção dos desenvolvedores, será realizada uma sessão de treinamento com todos os desenvolvedores colaboradores. Serão discutidos todos os conceitos necessários para a compreensão do estudo, como: (i) identificação de test smells, (ii) atributos de qualidade de código de teste e (iii) técnicas de refatoração de test smells. Também serão apresentados exemplos práticos que mostram os processos de refatoração que podem ser aplicados, dependendo do test smell.

A etapa (v) tem como objetivo refatorar os test smells. Esta etapa consiste na remoção dos test smells detectados nos sistemas. Os test smells serão encaminhados para os desenvolvedores para que eles apliquem a refatoração manual e removam os smells repassados. Cada desenvolvedor será responsável por refatorar exemplos de todos os tipos de test smells que serão analisados. Após a refatoração, será realizada uma validação dos commits de refatoração, afim de verificar se todos os test smells foram devidamente removidos dos sistemas. Também será aplicado um questionário com os desenvolvedores, com o intuito de identificar os test smells que são mais prejudiciais para a qualidade de um sistema, no ponto de vista dos desenvolvedores.

Na etapa (vi) executar os testes automatizados para identificar flaky tests. Identificar os tipos de flaky tests ocorrem nos projetos antes e após a refatoração dos tests smells. Dessa forma, é possível investigar se o fato de test smells estarem presentes no código de testes influencia a ocorrência de determinados tipos de test smells.

Na etapa (vii) consiste em catalogar os test smells mais prejudiciais identificados nos passos anteriores. Esse catálogo será disponibilizado de forma organizada em um site para disseminação e utilização do catálogo por outros desenvolvedores.

6. Referências Bibliográficas

AL DALLAL, J. Object-oriented class maintainability prediction using internal quality attributes. *Information and Software Technology*, 55(11), 2028-2048, 2013.

BAVOTA, G.; QUSEF, A.; OLIVETO, R.; LUCIA, A. D.; BINKLEY, D. Are test smells really harmful?

an empirical study. *Empirical Software Engineering*, Springer, v. 20, n. 4, p. 1052%u20131094, 2015.

BELLER, M.; GOUSIOS, G.; PANICHELLA, A.; PROKSCH, S.; AMANN, S.; ZAIDMAN, A. Developer testing in the ide: Patterns, beliefs, and behavior. *IEEE Transactions on Software Engineering*, IEEE, v. 45, n. 3, p. 261%u2013284, 2017.

CANDEA, G.; BUCUR, S.; ZAMFIR, C. Automated software testing as a service. In: *Proceedings of the 1st ACM symposium on Cloud computing*. [S. l.: s. n.], 2010. p. 155%u2013160.

DEURSEN, A. V.; MOONEN, L.; BERGH, A. V. D.; KOK, G. Refactoring test code. In: *CITeseer. Proceedings of the 2nd international conference on extreme programming and flexible processes in software engineering (XP2001)*. [S. l.], 2001. p. 92%u201395.

FOWLER, M. *Refactoring: improving the design of existing code*. [S. l.]: illustrated edition, 1999.

LAM, W.; MUslu, K.; SAJNANI, H.; THUMMALAPENTA, S. A study on the lifecycle of flaky tests. In: *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2020a. (ICSE %u20132020), p. 1471%u20131482. ISBN 9781450371216. Disponível em: <https://doi.org/10.1145/3377811.3381749>.

LAM, W.; WINTER, S.; WEI, A.; XIE, T.; MARINOV, D.; BELL, J. A large-scale longitudinal study of flaky tests. *Proc. ACM Program. Lang.*, Association for Computing Machinery, New York, NY, USA, v. 4, n. OOPSLA, nov 2020b. Disponível em: <https://doi.org/10.1145/3428270>.

KAUR, A.; DHIMAN, G. A review on search-based tools and techniques to identify bad code smells in object-oriented systems. In: *Harmony search and nature inspired optimization algorithms*. [S. l.]: Springer, 2019. p. 909%u2013921.

KIM, D. J. An empirical study on the evolution of test smell. In: *IEEE. 2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. [S. l.], 2020. p. 149%u2013151.

MALHOTRA, R.; CHUG, A. An empirical study to assess the effects of refactoring on software maintainability. In: *IEEE. 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. [S. l.], 2016. p. 110%u2013117.

MICCO, J. The state of continuous integration testing @Google. 2017. Disponível em: <https://research.google/pubs/pub45880/>. Acesso em: 14 abr. 2023.

MORASCA, S. A probability-based approach for measuring external attributes of software artifacts. In: *IEEE COMPUTER SOCIETY. 3rd ESEM*. [S. l.], 2009. p. 44%u201355.

MYERS, G. J.; SANDLER, C.; BADGETT, T. *The art of software testing*. [S. l.]: John Wiley & Sons, 2011.

PALOMBA, F.; ZAIDMAN, A.; LUCIA, A. D. Automatic test smell detection using information retrieval techniques. In: *IEEE. 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. [S. l.], 2018. p. 311%u2013332.

PECORELLI, F.; PALOMBA, F.; LUCIA, A. D. The relation of test-related factors to software quality: A case study on apache systems. *Empirical Software Engineering*, Springer, v. 26, n. 2, p.

1%u20132021.

ROMANO, A.; SONG, Z.; GRANDHI, S.; YANG, W.; WANG, W. An empirical analysis of ui-based flaky tests. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). [S. l.: s. n.], 2021. p. 1585%u20131597. ISSN 1558-1225.

SOARES, E.; RIBEIRO, M.; GHEYI, R.; AMARAL, G.; SANTOS, A. M. Refactoring test smells with junit 5: Why should developers keep up-to-date. IEEE Transactions on Software Engineering, IEEE, 2022.

SPADINI, D.; SCHVARCBACHER, M.; OPRESCU, A.-M.; BRUNTINK, M.; BACCHELLI, A. Investigating severity thresholds for test smells. In: Proceedings of the 17th International Conference on Mining Software Repositories. [S. l.: s. n.], 2020. p. 311%u2013321.

7. Plano de Atividades

Mês	Bolsista 1	Bolsista 2
1	Estudar referencial teórico de test smells.	Estudar referencial teórico de test smells e Flaky tests.
2	Estudar referencial teórico de test smells.	Estudar ferramentas de test smells e Flaky tests
3	Selecionar projetos de código aberto	Selecionar projetos de código aberto
4	Planejar experimento com os desenvolvedores	Executar ferramentas de test smells
5	Medir os atributo de qualidade dos projetos	Executar ferramentas de flaky tests
6	Refatorar test smells	Refatorar test smells
7	Refatorar test smells	Refatorar test smells
8	Investigar quais os test smells mais prejudiciais de acordo com as métricas e os desenvolvedores	Executar novamente as ferramentas de test smells e flaky test após os test smells refatorados
9	Investigar quais os test smells mais prejudiciais de acordo com as métricas e os desenvolvedores	Investigar se determinados tipos de test smells influenciam na ocorrência dos flaky tests
10	Elaborar um catálogo com os test smells mais prejudiciais	Investigar se determinados tipos de test smells influenciam na ocorrência dos flaky tests
11	Escrever artigo para publicação	Escrever artigo para publicação
12	Escrever relatório final de atividades, reunião de encerramento do projeto e publicação dos resultados no Encontro de Iniciação Científica da UFC	Escrever relatório final de atividades, reunião de encerramento do projeto e publicação dos resultados no Encontro de Iniciação Científica da UFC