

Universidade Federal do Ceará
Pró-Reitoria de Pesquisa e Pós-Graduação
PIBITI 2023/2024 (Desenvolvimento Tecnológico e Inovação) - Edital
Nº 03/2023

**Desenvolvimento de uma ferramenta para o ensino e prática de
refatoração de code smells e test smells**

Resumo

Code smells são estruturas que podem trazer prejuízos para a qualidade e manutenibilidade do código. Test smells representam um conjunto de testes mal elaborados, que podem prejudicar os critérios de manutenibilidade e qualidade de um código de teste. Refatoração é uma prática comum usada por desenvolvedores para eliminar test smells, code smells e suas coocorrências. A refatoração auxilia o código ser mais fácil de ser entendido e modificado através da eliminação de possíveis problemas e melhorando os atributos de qualidade. Uma grande problemática é que as atividades de refatoração são realizadas de geralmente de forma manual e indisciplinada, o que pode causar degradação do código. Apesar dos conceitos de code smells e test smells serem importantes para melhorar a manutenibilidade e qualidade do código, a indústria e a academia não tem ensinado esses conceitos para os desenvolvedores. Profissionais da indústria de software desconhecem os conceitos de code smells e test smells, e os professores não atualizam seus alunos sobre conceitos mais atuais de práticas de análise do código. Além disso, existem poucas ferramentas propostas para o ensino de refatoração de code smells e test smells, e a maior parte delas se concentram em refatoração. Dessa forma, o principal objetivo do projeto é propor uma ferramenta de ensino para refatoração de code smells e test smells. Os objetivos específicos que pretende-se alcançar no projeto, são: (i) desenvolver material sobre conceitos essenciais sobre code smells e test smells e refatoração dessas anomalias; (ii) consolidar ferramentas automatizadas para detecção de code smells e test smells; (iii) construir uma metodologia no ambiente de ensino para guiar docentes e alunos na prática de refatoração de code smells e test smells; e, (iv) investigar se a ferramenta construída foi efetiva para o ensino e prática de refatoração de code smells e test smells. Para alcançar esses objetivos serão realizados estudos experimentais e a construção de uma infraestrutura automatizada para o ensino e prática da refatoração de code smells e test smells.

1. Introdução

Ao longo da sua evolução, o software sistematicamente sofre alterações que podem levar à deterioração de sua estrutura. Nesse contexto, code smells podem indicar problemas relacionados com aspectos da qualidade do código, tal como legibilidade e a modificabilidade (Fowler, 2018).

Diferente de um bug ou defeito no código fonte de um sistema, a presença de code smells não significa a presença de defeitos no software. No entanto, essas anomalias podem trazer outras consequências negativas, como impacto de forma negativa na manutenção e evolução de um determinado sistema (Lacerda et al., 2020). Trabalhos anteriores (De Mello et al., 2019; Uchoa et al., 2020) mostraram evidências de que code smells são fortes indicadores de partes do código afetadas por pobre decomposição de features em sistemas de software.

Fowler et al., 2018, definem um catálogo de 22 tipos de code smells que afetam diferentes níveis do código fonte e que podem trazer problemas para a manutenibilidade do software. Outros autores definem outros tipos de code smells (Lanza e Marinescu, 2007). Exemplos de code smells mais comuns são: God Class - uma classe muito grande ou muito complexa; Long Method - método longo, complexo, com muitas variáveis e que centraliza a inteligência de uma classe; Duplicated Code - código duplicado em mais de um lugar no código fonte; Disperse Coupling - método que chama um ou mais métodos de várias classes, entre outros. Esses code smells são detectados individualmente a nível de classe ou a nível de método, e existem várias ferramentas que apoiam essa detecção. No entanto, essas ferramentas detectam apenas um pequeno número de smells (Lacerda et al., 2020).

Van Deursen et al. (2001) introduziram o conceito de test smells, que representam um teste mal projetado que pode prejudicar a legibilidade, manutenção e qualidade do código de teste. Desde sua conceituação, os test smells tornaram-se foco de diversos estudos na área de teste de software devido a sua importância para a qualidade do código de teste (Bavota et al. 2015, Kim 2020). Alguns estudos indicam que os test smells afetam negativamente a manutenção e a compreensão do código de teste (Bavota et al. 2015, Spadini et al. 2018). Portanto, é necessário entender seus prováveis efeitos e propor mecanismos, estratégias e ferramentas para mitigá-los. A refatoração de software consiste em pequenas transformações de código para melhorar a qualidade do código-fonte sem comprometer sua funcionalidade geral e comportamento observável (Paixão et al. 2020). Os test smells podem ser removidos por meio de refatoração, o que pode impactar positivamente na qualidade do código de teste (Spadini et al. 2018, Campos et al. 2021).

Refatoração pode remover code smells e test smells, podendo ter um impacto direto na qualidade do código. As pesquisas realizadas em refatoração de software já tem mais de 30 anos (Abid et al., 2020). A prática de refatoração de código se popularizou com o lançamento do livro "Improving the Design of Existing Code" em 1999 escrito pelo Martin Fowler e atualizado pelo autor em 2018 (Fowler, 2018). O autor define refatoração como sendo um conjunto de pequenas mudanças que são realizadas na estrutura interna do código sem alterar o comportamento externo (Lacerda et al., 2020).

A refatoração pode ser feita de forma manual ou pode contar com o apoio de ferramentas (Agnihotri e Chug, 2020). No entanto, o refatoramento é geralmente realizado de forma manual ou indisciplinada (Alizadeh et al., 2020). Existem evidências na literatura que sugerem efeitos indesejados de refatorações equivocadas, como degradação do código, principalmente para refatorações mais complexas (Abid et al., 2020).

Apesar da importância da detecção, refatoração e remoção dos e code smells e test smells para qualidade do código e código de testes, pouco se tem dado atenção a esses conceitos no ensino e prática desses conceitos na academia e na indústria (Bai et al. 2021). Poucos trabalhos da literatura tem apresentado conteúdos e práticas para o ensino de refatoração de code smells e test smells (Aniche et al. 2019, Bai et al. 2021, Aljedaani et al. 2023). Isso pode ser percebido também nos conceitos apresentados no ensino de testes de software no currículo de cursos de computação (Garousi et al. 2020).

Dessa forma, o principal objetivo do projeto é propor uma ferramenta de ensino para refatoração de code smells e test smells. Os objetivos específicos que pretende-se alcançar no projeto, são: (i) desenvolver material sobre conceitos essenciais sobre code smells e test smells e refatoração dessas anomalias; (ii) consolidar ferramentas automatizadas para detecção de code smells e test smells; (iii) construir uma metodologia no ambiente de ensino para guiar docentes e alunos na prática de refatoração de code smells e test smells; e, (iv) investigar se a ferramenta construída foi

efetiva para o ensino e prática de refatoração de code smells e test smells. Para alcançar esses objetivos serão realizados estudos experimentais e a construção de uma infraestrutura automatizada para o ensino e prática da refatoração de code smells e test smells.

2. Perguntas de Partida

Neste cenário surgem as seguintes questões de pesquisa:

- Como melhorar o ensino e práticas de refatoração de code smells e test smells em um ambiente automatizado?
- O ambiente automatizado para o ensino de refatoração de code smells e test smells tem um impacto positivo na perspectiva dos alunos?

3. Hipóteses

Baseado nas questões de pesquisa as seguintes hipóteses serão verificadas:

- É possível automatizar atividade de refatoração de code smells e test smells em uma ferramenta para o ensino desses conceitos.
- Existem efeitos positivos nos alunos a partir do uso de uma ferramenta para o ensino de refatoração de code smells e test smells.

4. Objetivos

Objetivo Geral:

Desenvolver uma ferramenta para o ensino e prática de refatoração de code smells e test smells.

Objetivos específicos:

- desenvolver material sobre conceitos essenciais sobre code smells e test smells e refatoração dessas anomalias;
- consolidar ferramentas automatizadas para detecção de code smells e test smells;
- construir uma metodologia no ambiente de ensino para guiar docentes e alunos na prática de refatoração de code smells e test smells; e,
- investigar se a ferramenta construída foi efetiva para o ensino e prática de refatoração de code smells e test smells.

5. Materiais e Métodos

Para atender aos objetivos do projeto foram definidas quatro etapas de trabalho: (i) selecionar e desenvolver material base para o ensino de refatoração de code smells e test smells; (ii) identificar e selecionar ferramentas para detecção de code smells e test smells; (iii) construir uma metodologia que possa ser incorporada a ferramenta para o ensino e prática de refatoração de code smells e test smells; (iv) implementar ferramenta para o ensino e prática de refatoração de code smells e test smells; e (v) validar se a ferramenta foi efetiva no ensino e prática desses conceitos.

A etapa (i) consiste em selecionar e desenvolver material base para o ensino de refatoração de code smells e test smells. Existem diversos materiais espalhados na literatura sobre code smells e test smells. Também existem dois catálogos que podem nos guiar no desenvolvimento de material para esses conceitos: <https://testsmells.org/> (test smells) e <https://refactoring.guru/refactoring/smells> (code smells). Sobre code smells e refatoração existem também algumas revisões sistemáticas da

literatura (Larcerda et al., 2020, Abid et al. 2020, Agnihotri et al. 2020), e uma revisão sistemática sobre ferramentas de test smells (Aljedaani et al., 2021).

A etapa (ii) consiste em identificar e selecionar ferramentas para detecção de code smells e test smells. Existem diversas ferramentas para detecção de code smells e test smells. Essas ferramentas geralmente são utilizadas apenas para detecção e poucas fazem refatoração automática dessas anomalias. Para compor a ferramenta de ensino é importante selecionar as ferramentas mais utilizadas e mais precisas.

A etapa (iii) visa construir uma metodologia que possa ser incorporada a ferramenta para o ensino e prática de refatoração de code smells e test smells. A metodologia a ser construída vai guiar o desenvolvimento da ferramenta para o ensino desses conceitos e irá se basear nas etapas (i) e (ii).

A etapa (iv) visa implementar ferramenta para o ensino e prática de refatoração de code smells e test smells. A construção da ferramenta irá ter como base os resultados das etapas (i), (ii) e (iii). Também será baseada em outras ferramentas existentes no ensino de refatoração. A ferramenta irá permitir tanto docentes quanto profissionais da indústria a ensinarem e praticarem conceitos de refatoração de code smells e test smells.

Por fim, a etapa (v) tem como objetivo validar a ferramenta implementada em turmas de qualidade de software, manutenção de software ou engenharia de software, de forma a verificar se há um impacto positivo no aprendizado dos conceitos de refatoração de code smells e test smells. Pretende-se aplicar a ferramenta em várias turmas e em projetos de código aberto.

6. Referências Bibliográficas

Abid, C., Alizadeh, V., Kessentini, M., Ferreira, T. D. N., & Dig, D. (2020). 30 Years of Software Refactoring Research: A Systematic Literature Review. arXiv preprint arXiv:2007.02194.

Agnihotri, M., & Chug, A. (2020). A systematic literature survey of software metrics, code smells and refactoring techniques. *Journal of Information Processing Systems*, 16(4), 915-934.

Aljedaani, W., Peruma, A., Aljohani, A., Alotaibi, M., Mkaouer, M. W., Ouni, A., & Ludi, S. (2021). Test smell detection tools: A systematic mapping study. *Evaluation and Assessment in Software Engineering*, 170-180.

Alizadeh, V., Kessentini, M., Mkaouer, M. W., O Cinn eide, M., Ouni, A., and Cai, Y. (2020). An interactive and dynamic search-based approach to software refactoring recommendations. *IEEE Transactions on Software Engineering*, 46(9):932-936.

Aljedaani, W., Mkaouer, M. W., Peruma, A., and Ludi, S. (2023). Do the test smells assertion roulette and eager test impact students' troubleshooting and debugging capabilities? arXiv preprint arXiv:2303.04234.

Aniche, M., Hermans, F., and van Deursen, A. (2019). Pragmatic software testing education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education, SIGCSE 2019*, page 414-420, New York, NY, USA. Association for Computing Machinery.

Bai, G. R., Smith, J., and Stolee, K. T. (2021). How students unit test: Perceptions, practices, and pitfalls. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, pages 248-254.

- Bavota, G.; Qusef, A.; Oliveto, R.; Lucia, A. D.; Binkley, D. (2015) Are test smells really harmful? an empirical study. *Empirical Software Engineering*, Springer, v. 20, n. 4, p. 1052-11094.
- Campos, D., Rocha, L., and Machado, I. (2021). Developers perception on the severity of test smells: an empirical study. *XXIV Ibero-American Conference on Software Engineering*.
- de Mello, R., Uchôa, A., Oliveira, R., Oizumi, W., Souza, J., Mendes, K., Garcia, A. (2019, September). Do research and practice of code smell identification walk together? a social representations analysis. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (pp. 1-6). IEEE.
- Deursen, A. V.; Moonen, L.; Bergh, A. V. D.; Kok, G. (2001) Refactoring test code. In: *CITeseer. Proceedings of the 2nd international conference on extreme programming and flexible processes in software engineering (XP2001)*. [S. l.]. p. 92-1395.
- Fowler, M. (2018). *Refactoring: improving the design of existing code*. Addison-Wesley Professional.
- Garousi, V., Rainer, A., Lauv Aas Jr, P., and Arcuri, A. (2020). Software-testing education: A systematic literature mapping. *Journal of Systems and Software*, 165:110570.
- Lacerda, G., Petrillo, F., Pimenta, M., & Guéhéneuc, Y. G. (2020). Code smells and refactoring: A tertiary systematic review of challenges and observations. *Journal of Systems and Software*, 167, 110610.
- Lanza, M., & Marinescu, R. (2007). *Object-oriented metrics in practice: using software metrics to characterize, evaluate, and improve the design of object-oriented systems*. Springer Science & Business Media.
- Kim, D. J. An empirical study on the evolution of test smell. (2020) In: *IEEE. 2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. [S. l.]. p. 149-151.
- Paixão, M., Uchoa, A., Bibiano, A. C., Oliveira, D., Garcia, A., Krinke, J., and Arvonio, E. (2020). Behind the Intents: An In-Depth Empirical Study on Software Refactoring in Modern Code Review, page 125-136. *Association for Computing Machinery, New York, NY, USA*.
- Spadini, D., Palomba, F., Zaidman, A., Bruntink, M., and Bacchelli, A. (2018). On the relation of test smells to software code quality. In *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 1-12.
- Uchôa, A., Barbosa, C., Oizumi, W., Blenilio, P., Lima, R., Garcia, A., & Bezerra, C. (2020). How Does Modern Code Review Impact Software Design Degradation? An In-depth Empirical Study. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 511-522). IEEE.

7. Plano de Atividades

Mês	Bolsista 1	Bolsista 2
1	Estudar referencial teórico de code smells, test smells e técnicas de refatoração para remoção dessas anomalias	Estudar referencial teórico de code smells, test smells e técnicas de refatoração para remoção dessas anomalias

2	Identificar e selecionar material sobre conceitos de refatoração code smells e test smells	Identificar e selecionar ferramentas para detecção de code smells e test smells
3	Desenvolver material para o ensino e prática de refatoração de code smells e test smells	Instalar e analisar ferramentas para detecção de code smells e test smells
4	Desenvolver material para o ensino e prática de refatoração de code smells e test smells	Instalar e analisar ferramentas para detecção de code smells e test smells
5	Desenvolver metodologia para o ensino e prática de refatoração de code smells e test smells	Implementar ferramenta para o ensino e prática de refatoração de code smells e test smells
6	Desenvolver metodologia para o ensino e prática de refatoração de code smells e test smells	Implementar ferramenta para o ensino e prática de refatoração de code smells e test smells
7	Implementar ferramenta para o ensino e prática de refatoração de code smells e test smells	Implementar ferramenta para o ensino e prática de refatoração de code smells e test smells
8	Implementar ferramenta para o ensino e prática de refatoração de code smells e test smells	Implementar ferramenta para o ensino e prática de refatoração de code smells e test smells
9	Implementar ferramenta para o ensino e prática de refatoração de code smells e test smells	Realizar testes na ferramenta
10	Aplicar e Validar a ferramenta em turmas de qualidade de software, manutenção de software e engenharia de software	Aplicar e Validar a ferramenta em turmas de qualidade de software, manutenção de software e engenharia de software
11	Escrever artigo para publicação	Escrever artigo para publicação
12	Escrever relatório final de atividades, reunião de encerramento do projeto e publicação dos resultados nos Encontros Universitários	Escrever relatório final de atividades, reunião de encerramento do projeto e publicação dos resultados nos Encontros Universitários