



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

XXVII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS **SEMANA NACIONAL DE CIÊNCIA E TECNOLOGIA - 2023**

Análise de Padrões de Projeto e Code Smells: Uma Abordagem Quantitativa

Bianca Santana de Araújo Silva¹ e Jose Amancio Macedo Santos²

1. Bolsista FAPESB, Graduando em Engenharia de computação, Universidade Estadual de Feira de Santana, e-mail: bianca.silva2344@gmail.com
2. Orientador, DEXA, Universidade Estadual de Feira de Santana, e-mail: zeamancio@gmail.com

PALAVRAS-CHAVE: padrões de projeto, code smell, análise quantitativa .

INTRODUÇÃO

Padrões de projeto são soluções reutilizáveis para problemas comuns encontrados no desenvolvimento de software. Esses padrões ajudam a melhorar a qualidade do código, facilitam a manutenção e aumentam a escalabilidade do software (Gamma et al., 1994). No entanto, a aplicação inadequada desses padrões pode levar ao surgimento de code smells, que são características indesejáveis no código, que indicam problemas de design e podem afetar negativamente a qualidade do software. Fowler (1999) descreve code smells como "indícios de que algo pode estar errado" e enfatiza a importância de abordar esses problemas para evitar a deterioração da qualidade do código. Devido à relevância dos conceitos, há estudos buscando compreender a relação entre padrões de projeto e code smells (Yamashita e Moonen, 2013; Ederson Assunção e Rodrigo Souza; 2019).

Neste artigo, analisamos a relação entre padrões de projeto e code smells através de uma abordagem quantitativa. Buscamos entender como a aplicação correta de padrões de projeto pode ajudar a prevenir ou mitigar a ocorrência de code smells no código. Este estudo preenche uma lacuna existente na literatura, ao abordar não apenas a relação geral entre padrões de projeto e code smells, mas também ao identificar relações específicas entre diferentes tipos de padrões de projeto e code smells. Além disso, ao utilizar uma abordagem quantitativa, este artigo contribui para um melhor entendimento melhor da interação entre padrões de projeto e qualidade de software.

METODOLOGIA

Neste estudo, os sistemas analisados são baseados em um conjunto de dados originalmente disponibilizado por Walter et al[3]. No total, 92 sistemas Java são analisados. Walter et al[3] empregaram seis ferramentas distintas para a detecção de 'Code Smells', identificando um total de 14 tipos diferentes de 'smells'. Santos et al [4] realizaram um estudo utilizando um subconjunto dos sistemas disponibilizados no

dataset apresentado por Walter et al. como fonte de dado para identificação das classes com incidências de code smells. Os software, então, foram submetidos ao DPD (Design Pattern Detection Using Similarity Scoring) para detectar padrões de projeto. Esta ferramenta, aplicada aos 61 sistemas presentes no conjunto de dados de Santos et al, detecta 15 tipos de padrões de design. O DPD foi empregado positivamente em vários estudos, como por exemplo [9] e [10]. O dataset disponibilizado por Santos et al contém, então, informações sobre os code smells e padrões de projeto associados aos 61 sistemas Java. O dataset está disponível em <https://zenodo.org/record/3731749>.

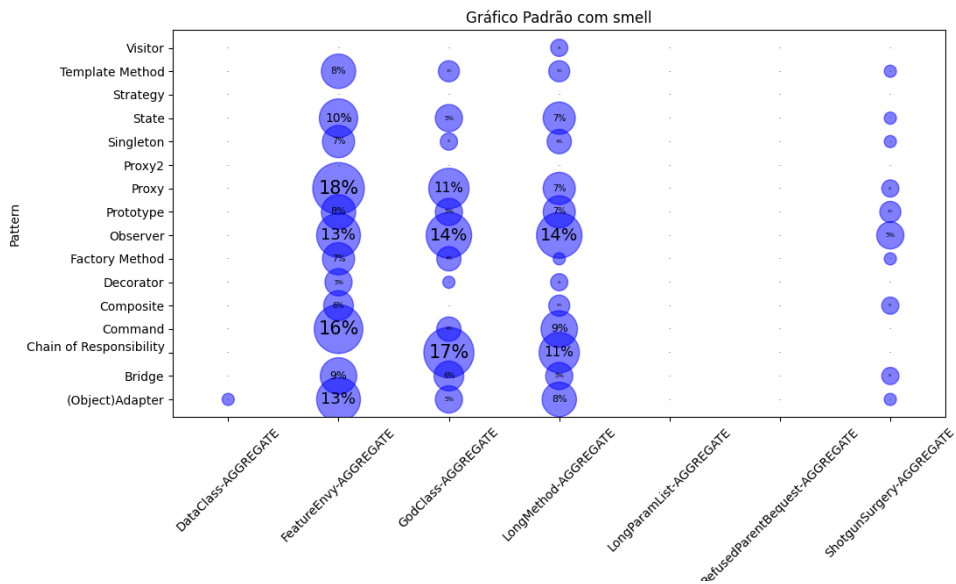
Este trabalho evolui o estudo realizado por Santos et al [4], investigando a relação entre code smells e padrões de projeto, considerando tipos específicos de padrões de projeto e code smells. A análise apresentada por Santos et al [4] desconsidera os tipos de code smells e padrões. Neste trabalho observamos a relação entre os padrões de projeto e os 'code smells' em cada sistema e classe individualmente. Esta relação foi identificada através do registro de 4.049 'smells' e 5.869 padrões, levando em consideração as diversas tipologias. Com base nestes números, foi realizada uma normalização dos dados, tomando como referência a porcentagem de 'smells' e padrões. Esta normalização por percentual foi necessária, pois permitiu uma interpretação mais justa e equilibrada dos dados. A partir disso, calculamos o percentual de classes com smells que não tem padrão e vice-versa.

RESULTADOS

A análise dos dados revelou vários insights sobre a relação entre padrões de projeto e a presença de code smells. Primeiramente, notou-se uma correlação negativa entre a aplicação adequada de certos padrões de projeto e a ocorrência de code smells. Por exemplo, sistemas que faziam uso do padrão "Singleton" mostram uma redução na presença de code smells relacionado à "God Class", comparado com os outros padrões.

O primeiro gráfico ilustra a distribuição percentual de diferentes code smells em relação aos padrões de projeto implementados. Neste gráfico, padrões como "Visitor" e "Decorator" mostraram baixas percentagens de ocorrência de smells. Isso corrobora os insights anteriores e sugere que esses padrões, quando bem aplicados, podem efetivamente reduzir a incidência de certos code smells.

Gráfico 1. Porcentagem de padrão associados a cada smell.



Por outro lado, alguns padrões, quando mal aplicados, resultam em um aumento na incidência de code smells. O padrão "Factory Method", quando usado de forma inadequada, tende a introduzir code smells como "God Class" e "Long Method". Este resultado é coerente com a literatura existente, que sugere que a aplicação inadequada de padrões de projeto pode, de fato, deteriorar a qualidade do código. Também calculamos o percentual dos padrões de projeto que não parecem afetar a presença de code smells. Percebemos que, alguns padrões como "Adapter" e "Observer", parecem não ter impacto significativo na mitigação de code smells.

O Segundo gráfico, "Smell com Padrão", foca em quais padrões de projeto são comumente associados a cada code smell. Este gráfico é útil para entender como evitar a introdução de code smells ao implementar padrões de projeto específicos. Por exemplo, é evidente que os smells "Long Method", "God Class", "Feature Envy" e "Shotgun Surgery" estão frequentemente associados aos padrões "Adapter" e "State", sugerindo cautela ao implementar esse padrão. Além disso, observamos também que os padrões "Long ParamList" e "Refused Parent Bequest" não obtiveram relações com os padrões.

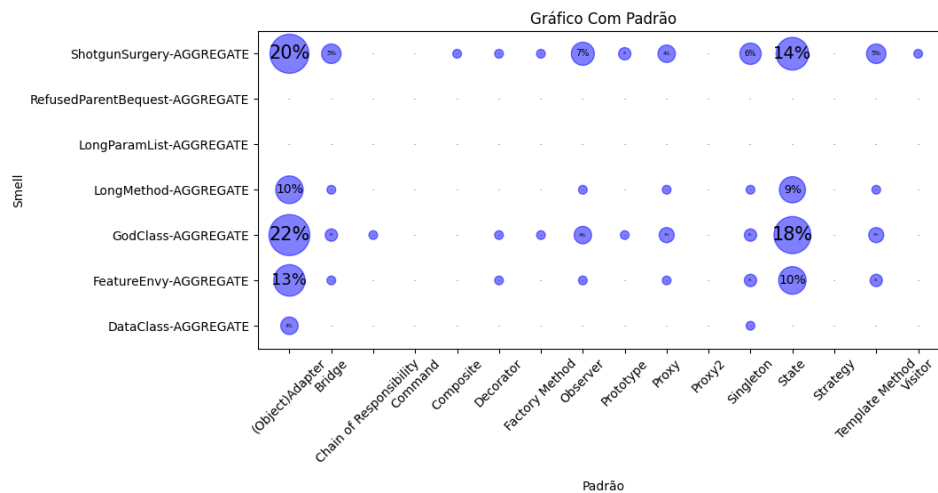
CONSIDERAÇÕES FINAIS

Este estudo proporciona uma visão sobre como padrões de projeto e code smells interagem no contexto do desenvolvimento de software. Comprova-se que a utilização de padrões de projeto pode ser uma ferramenta eficaz para prevenir ou mitigar a presença de code smells, corroborando estudos anteriores (Yamashita e Moonen, 2013).

A principal contribuição deste trabalho é a identificação de relações específicas entre diferentes tipos de padrões de projeto e code smells. Essa informação pode ser útil para desenvolvedores e engenheiros de software na hora de tomar decisões sobre quais padrões de projeto aplicar em diferentes cenários. Além disso, o estudo contribui para a literatura existente ao incorporar uma análise rigorosa baseada em dados empíricos, avançando assim o conhecimento na área de qualidade de software. Como trabalhos futuros, pretendemos expandir esta pesquisa ao investigar como diferentes métodos de

implementação para um mesmo padrão de projeto podem afetar a ocorrência de code smells. Também pretendemos explorar a eficácia de outras técnicas de refatoração para eliminar code smells em sistemas que fazem uso extensivo de padrões de projeto.

Gráfico 2: Porcentagem de smell associados a cada padrão



REFERÊNCIAS

- [1] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.
- [2] Yamashita, A. and Moonen, L. (2013). The impact of code smells on software maintainability: a case study of two open source systems. In Proceedings of the 2013 ICSE 2013, pages 792-801. IEEE Press.
- [3] Walter B, Fontana FA, Ferme V. Code smells and their collocations: A large-scale experiment on open-source systems. J Syst Softw. 2018;144:1-21.
- [4] Santos, José Amancio Macedo, & Petrolino, Gadiel Xavier Antunes. (2020). The relationship between code smells and design patterns: an external replicated experiment.
- [5] Assunção, E., & Souza, R. (2019). Incidence of code smells in the application of design patterns: a method-level analysis. SBCARS '19, Setembro 23–27, 2019, Salvador, Brasil. ACM.
- [6] Fowler M. Refactoring: Improving the Design of Existing Code: Addison-Wesley Longman Publishing Co., Inc.; 1999.
- [7] Zenodo. Disponível em: <<https://doi.org/10.5281/zenodo.3731749>>. Acessado em: 26 de setembro de 2023
- [8] Terra R, Miranda LF, Valente MT, Bigonha RS. Qualitas.class Corpus: a compiled version of the Qualitas Corpus. Softw Eng Notes. 2013;38:1-4. 20.
- [9] Cardoso B, Figueiredo E. Co-occurrence of design patterns and bad smells in software systems: an exploratory study. In: 2013 Brazilian Symposium on Information Systems. SBC; 2015:347-354.
- [10] Pettersson N, Löwe W, Nivre J. Evaluation of accuracy in design pattern occurrence detection. IEEE Trans Softw Eng. 2010;36:575-590.