Refatoração de *Code Smells* e seu Impacto na Qualidade do Software

Carla Bezerra
carlailane@ufc.br

Elixir em Foco
Setembro - 2023

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Sobre Mim

- Formação:
  - Graduação de Ciências da Computação UECE (2006)
  - Mestrado em Informática Aplicada UNIFOR (2009)
  - Doutorado em Ciência da Computação UFC (2016)

- Experiência profissional:
  - Analista de Testes – SERPRO – 2005 a 2006
  - Analista da Qualidade Atlântico – 2006 a 2010
  - Consultora e Implementadora de Processos – 2007 a 2010
  - Professora Adjunta UFC Campus Quixadá – 2010 até o momento

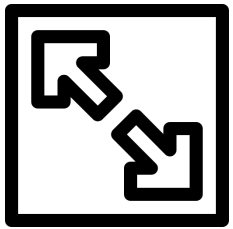
Áreas de Interesse

- Qualidade de Software
- Medidas de Manutenibilidade de Código
- *Code Smells / Test Smells* Refatoração de Software
- Integração Contínua
- Revisão de Código
- Linhas de Produto de Software
- Sistemas Autoadaptativos
- Internet das Coisas
- Educação em Engenharia de Software

Sobre o Campus UFC Quixadá

- Foi fundado em 2008
- Possui 6 cursos de graduação em TI:
  - Sistemas de Informação
  - Engenharia de Software
  - Redes de Computadores
  - Ciência da Computação
  - Design Digital
  - Engenharia de Computação
- Possui Mestrado em Ciência da Computação (PCOMP)
- Link: https://www.quixada.ufc.br/

Motivação



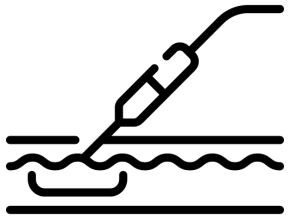
Ao longo da sua evolução, o software sistematicamente sofre alterações que podem levar à **deterioração** de sua estrutura de qualidade

(LANZA e MARINESCU, 2007)

Nesse contexto, surge o conceito de ***code smells***, que são estruturas de código anômalas que representam sintomas que afetam a manutenibilidade de sistemas em diferentes níveis, como classes e métodos

(FOWLER, 2018)

Motivação



Refatoração é o processo de melhorar a qualidade de software através de transformações no código fonte

(FOWLER, 2018)

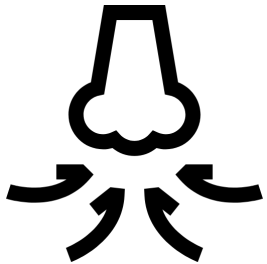
Refatoração mantém você pronto para mudanças te deixando confortável com a mudança do seu próprio código

(Ken Auer and Roy Miller, Extreme Programming Applied, page 189)

Code Smells

- *Code smells* são anomalias que podem indicar problemas relacionados à aspectos da qualidade do código

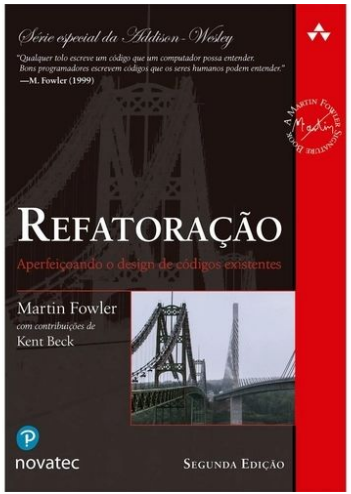
- Podem trazer prejuízos para:
  - *Design* do software
  - Atributos de qualidade (manutenibilidade, legibilidade)
  - Arquitetura de Software



Code Smells

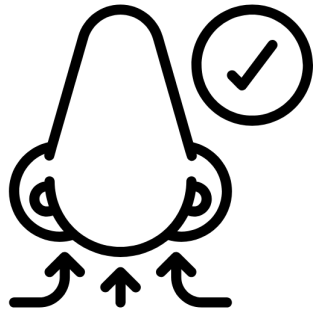
- (FOWLER, 2018) define um catálogo de 22 tipos de *code smells* que afetam diferentes níveis do código fonte e que podem trazer problemas para a manutenibilidade do software

- Outros autores definem outros tipos de *code smells* (LANZA e MARINESCU, 2007)

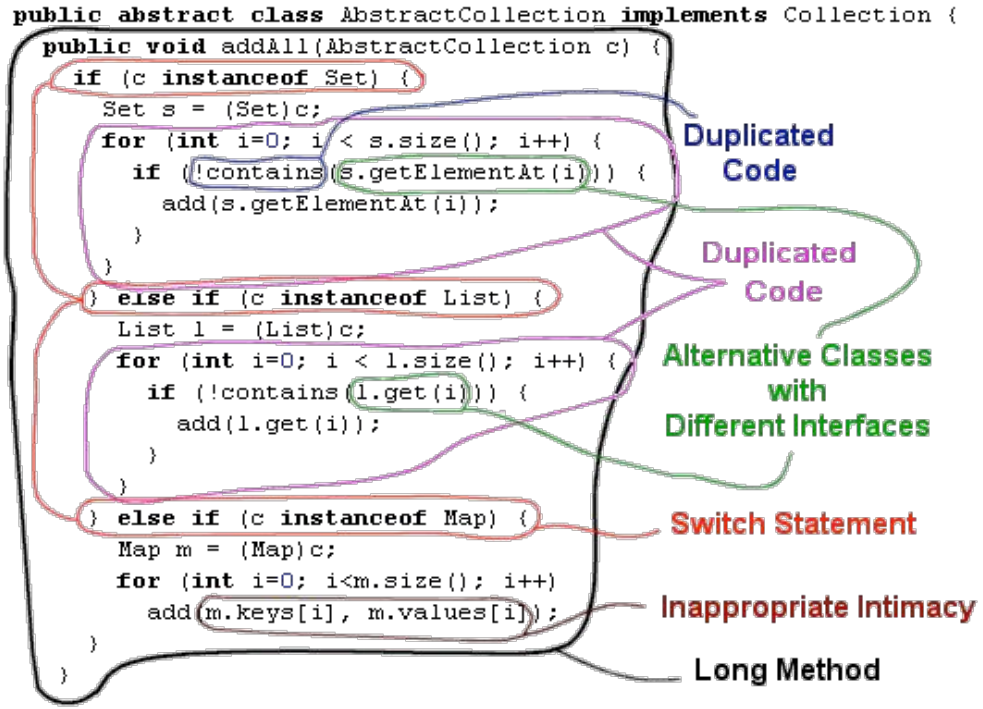


Exemplos de *Code Smells*

- *Data Class*
- *God Class*
- *Long Method*
- *Brain Method*
- *Brain Class*
- *Duplicated Code*
- *Dispersed Coupling*
- *Intensive Coupling*
- *Refused Parent Bequest*
- *Shotgun Surgery*
- *Tradition Breaker*
- *Type Checking*



Code Smells


public abstract class AbstractCollection implements Collection {
public void addAll(AbstractCollection c) {
if (c instanceof Set) {
Set s = (Set)c;
for (int i=0; i < s.size(); i++) {
if (!contains(s.getElementAt(i))) {
add(s.getElementAt(i));
}
}
} else if (c instanceof List) {
List l = (List)c;
for (int i=0; i < l.size(); i++) {
if (!contains(l.get(i))) {
add(l.get(i));
}
}
} else if (c instanceof Map) {
Map m = (Map)c;
for (int i=0; i<m.size(); i++)
add(m.keys[i], m.values[i]);
}
}
Duplicated Code
Duplicated Code
Alternative Classes with Different Interfaces
Switch Statement
Inappropriate Intimacy
Long Method

Code Smells

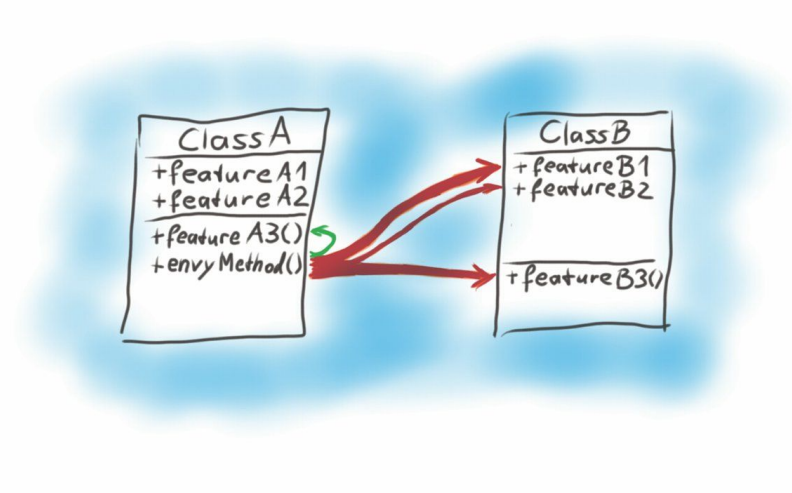
Long Method

- Métodos com muitas condicionais, variáveis e loops
- Métodos que fazem muita coisa
- Alguns autores dizem que um método não pode ter mais que 10 linhas de código

Code Smells

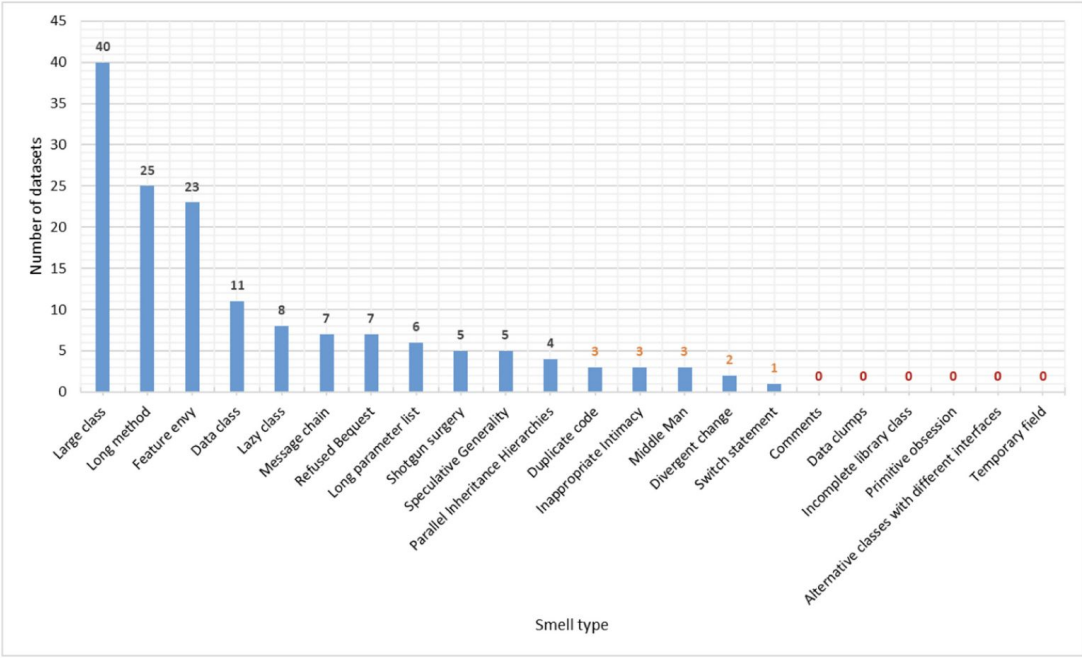
Feature Envy

- Método de uma classe que usa mais coisas de outra classe do que a sua própria



Code Smells

Zakeri-Nasrabadi, M., Parsa, S., Esmaili, E., & Palomba, F. (2023). A systematic literature review on the code smells datasets and validation mechanisms. ACM Computing Surveys..



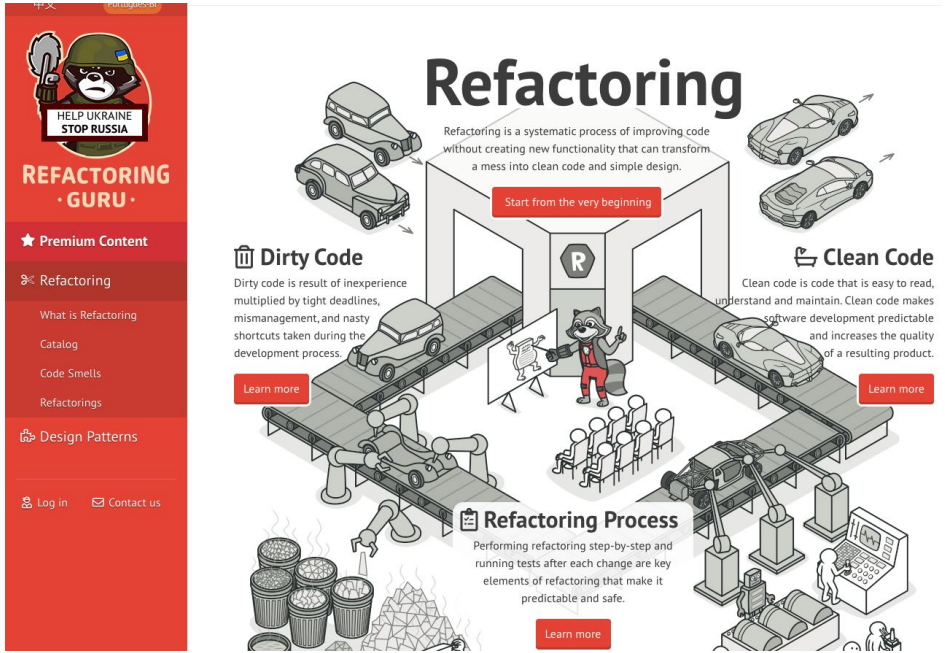
Code Smells em Elixir

Francisco da Matta Vegi, Lucas, and Marco Tulio Valente. "Understanding code smells in Elixir functional language." Empirical Software Engineering 28.4 (2023): 102.

Traditional smell	Sources	#Sources
COMMENTS	G1, G10, G12, G14	4
LONG PARAMETER LIST	G1, G16, E2, M3	4
LONG FUNCTION	G1, E2, E7	3
PRIMITIVE OBSESSION	G3, M4, M13	3
SHOTGUN SURGERY	G1, G17, M5	3
DUPLICATED CODE	G1, M26	2
FEATURE ENVY	G1, G6	2
DIVERGENT CHANGE	G1	1
INAPPROPRIATE INTIMACY	G1	1
LARGE CLASS	G1	1
SPECULATIVE GENERALITY	G1	1
SWITCH STATEMENTS	M10	1

Code Smells

- Refactoring Guru: https://refactoring.guru/refactoring



Ferramentas para Detecção de *Code Smells*

- Existem várias ferramentas para fazer a detecção de *code smells*

Ferramentas para Refatoração de *Code Smells*

- Existem poucas ferramentas para fazer a refatoração de *code smells*

Coocorrências de *Code Smells*

- Coocorrências de *code smells* ocorrem quando existe um relacionamento e dependência entre dois ou mais *code smells*

- Antes de detectar coocorrências de *code smells*, deve-se primeiro realizar a detecção das ocorrências individuais de *smells*

- Após essa detecção é possível então verificar as coocorrências de *code smells* a nível de **método** e a nível de **classe**

Coocorrências de *Code Smells*

Pacote	**Classe**	**Método**	**LM**	**FE**	**GC**
pacote1	Classe1	método1			X
pacote1	Classe2	método2	X		

Coocorrências de *Code Smells*

- Coocorrência a **nível de classe** e a **nível de método**

Pacote	**Classe**	**Método**	**LM**	**FE**	**GC**
pacote1	Classe1	método1	X		X
pacote1	Classe2	método2	X	X	

Atributos Internos de Qualidade

- Atributos internos de qualidade são aqueles que podem ser medidos utilizando apenas artefatos de software

- Atributos internos de qualidade:
  - Coesão
  - Acoplamento
  - Complexidade
  - Herança
  - Tamanho

Esses atributos serão medidos através de **métricas**

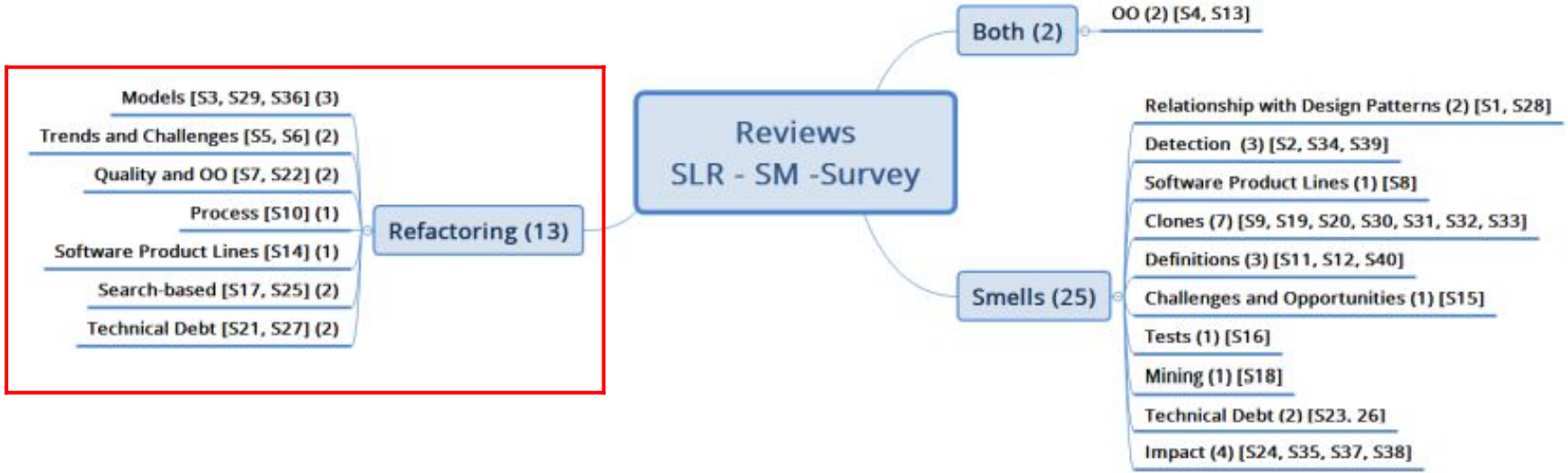
Refatoração e *code smells*

- (FOWLER, 2018) define um conjunto de métodos de refatoração para cada um dos 22 *code smells*:
  - *Extract Class*
  - *Extract Subclass*
  - *Extract Method*
  - *Move Method*
  - *Move Field*
  - *Replace Inheritance with Delegation*
  - *Replace Temp with Query*

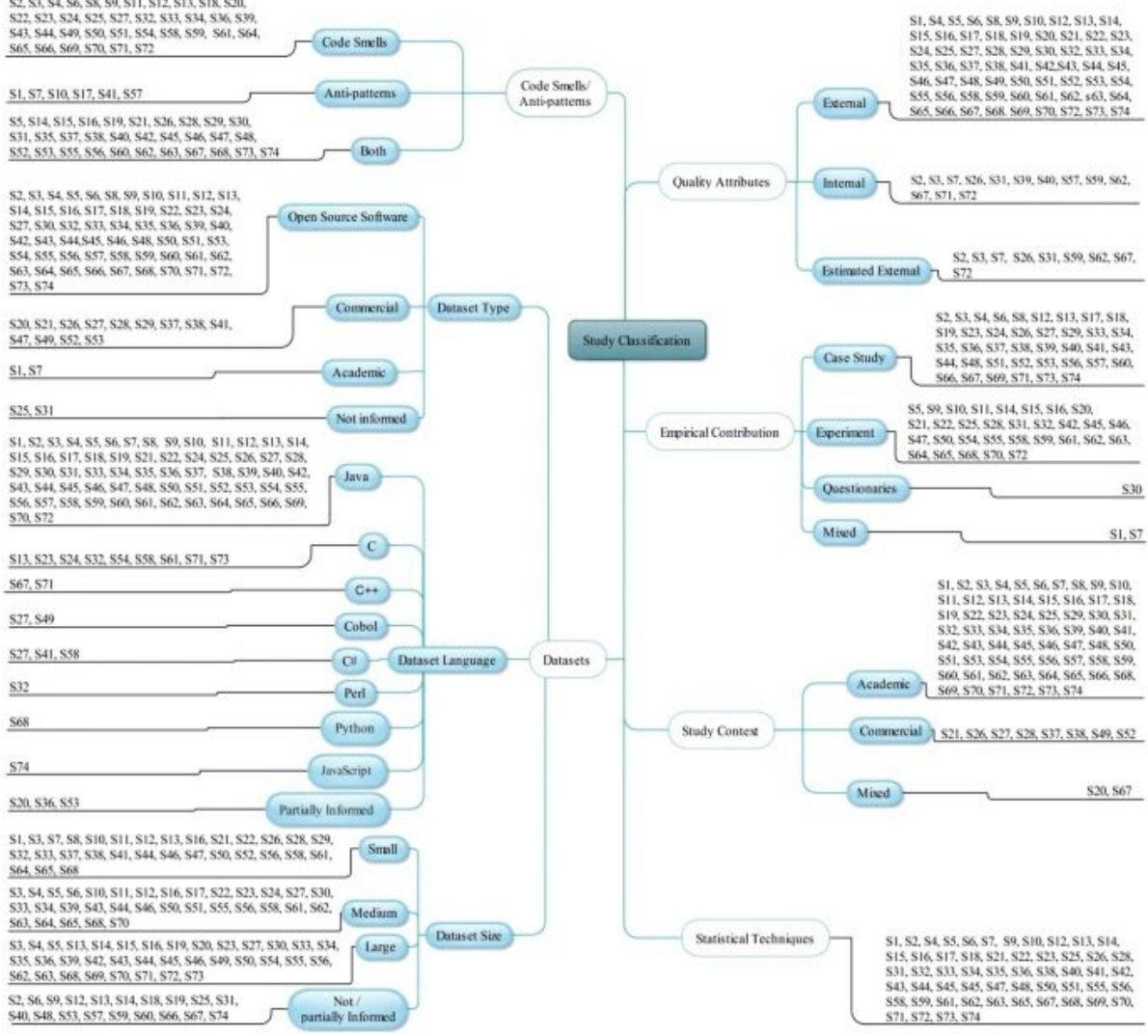
Refatoração de *code smells* e impacto na qualidade

- Na literatura existem poucos estudos que investigam o impacto da remoção dos *code smells* na qualidade do software
- A maior parte dos estudos investigam refatorações de código em aspectos de qualidade:
  - atributos externos de qualidade
  - atributos internos de qualidade
  - consumo de energia (relacionado à aplicações android)

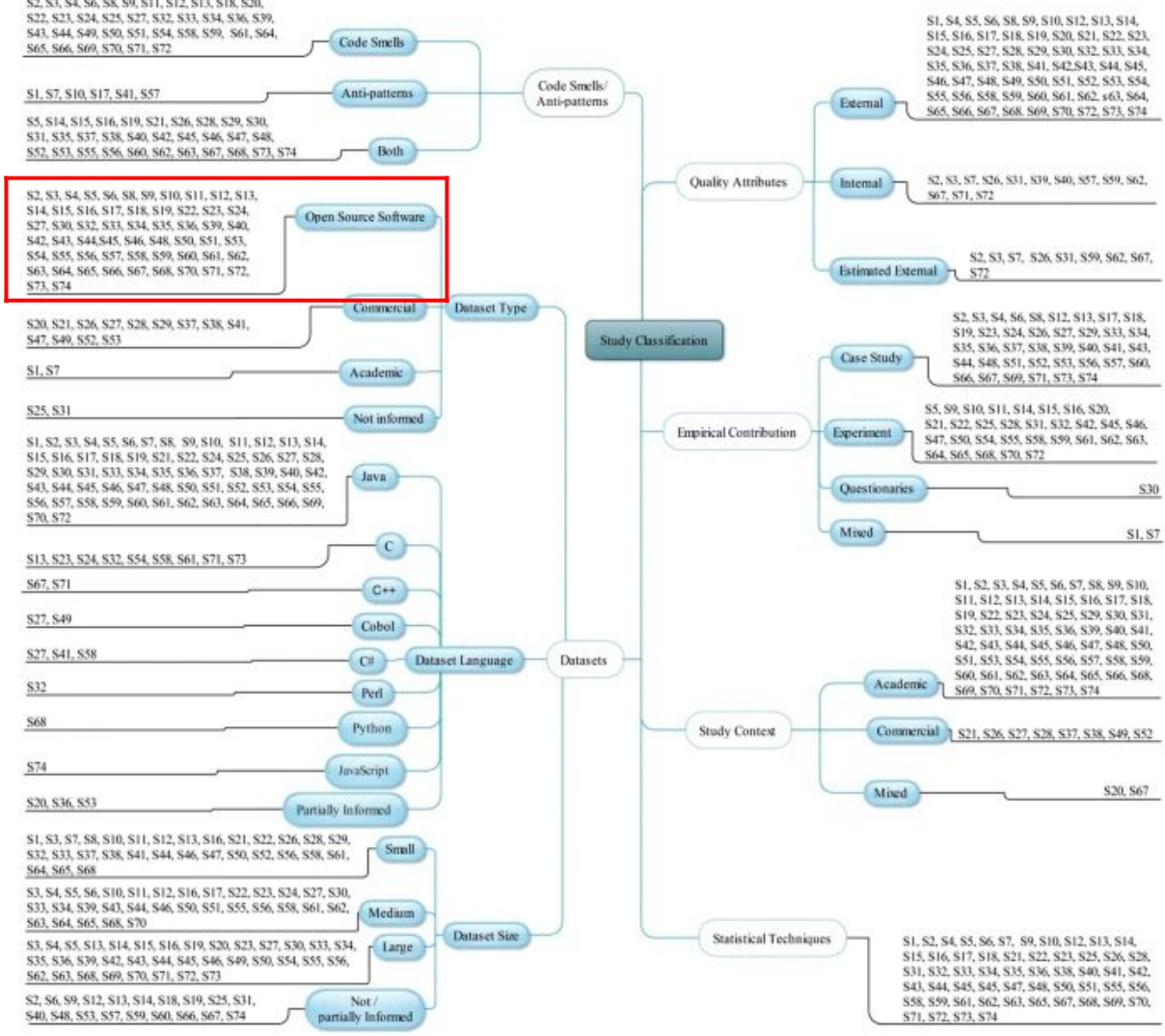
Code smells and refactoring: A tertiary systematic review of challenges and observations (Lacerda et al., JSS 2020)



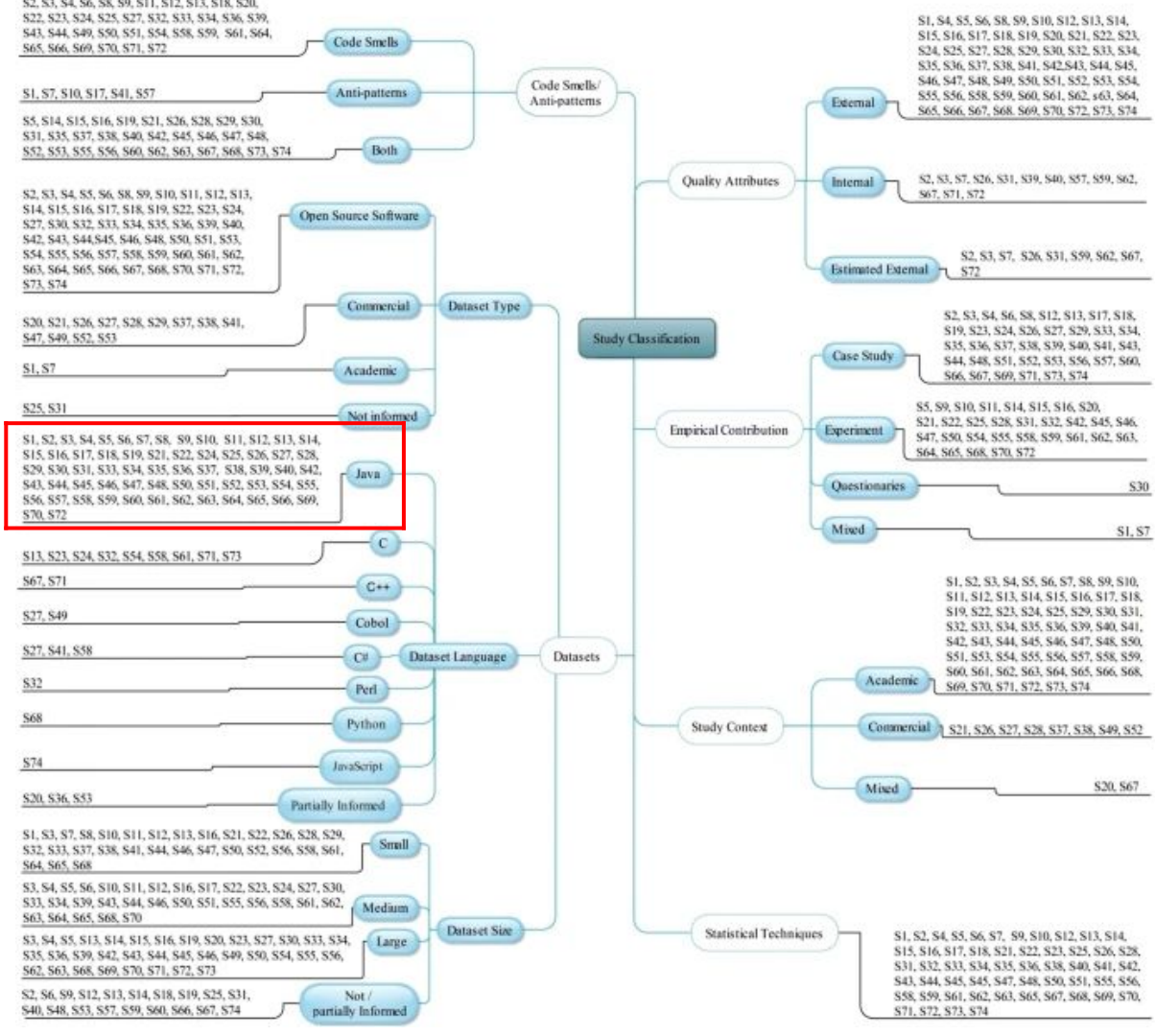
A Systematic Literature Review on Empirical Analysis of the Relationship Between Code Smells and Software Quality Attributes (Amandeep Kaur, ACME 2020)



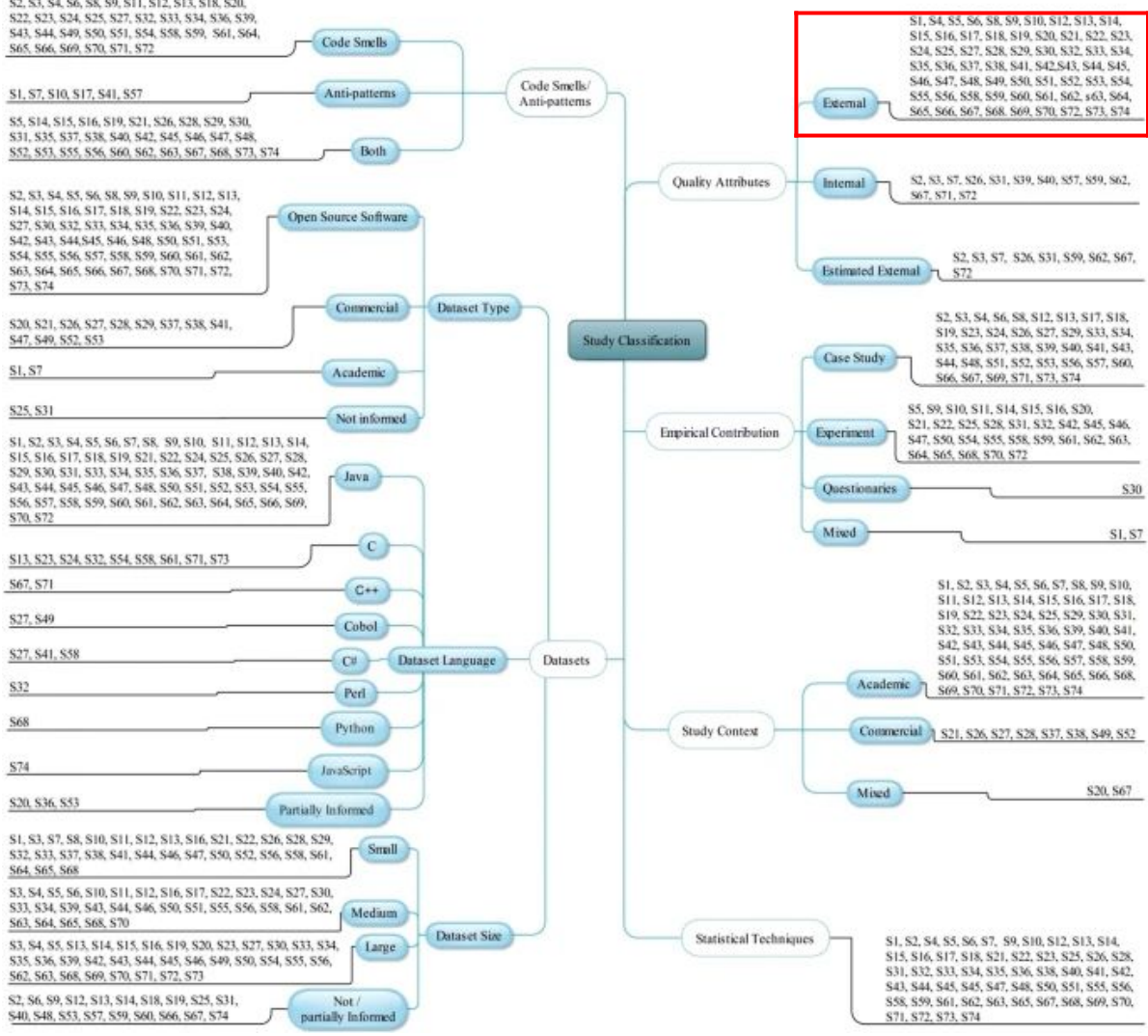
A Systematic Literature Review on Empirical Analysis of the Relationship Between Code Smells and Software Quality Attributes (Amandeep Kaur, ACME 2020)



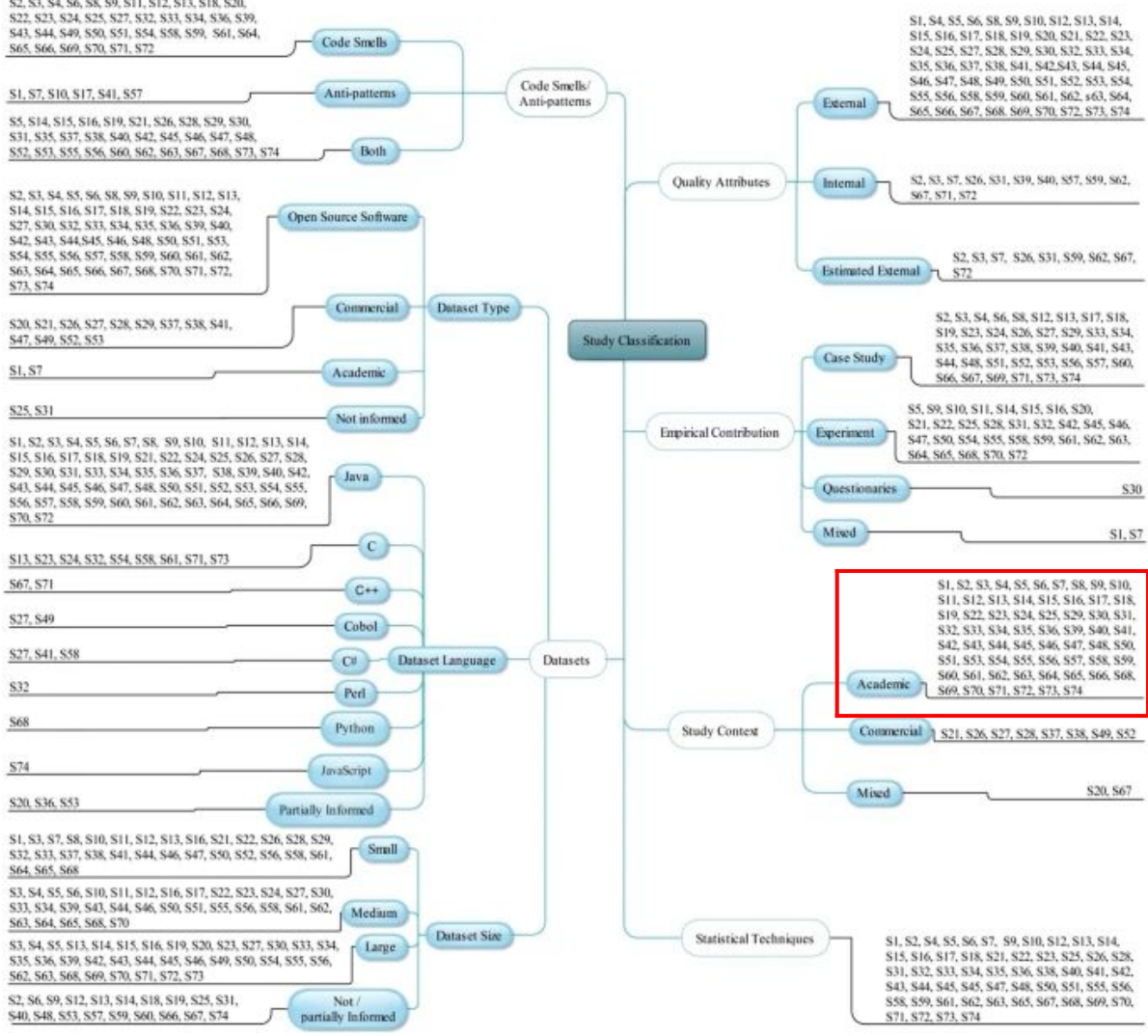
A Systematic Literature Review on Empirical Analysis of the Relationship Between Code Smells and Software Quality Attributes (Amandeep Kaur, ACME 2020)



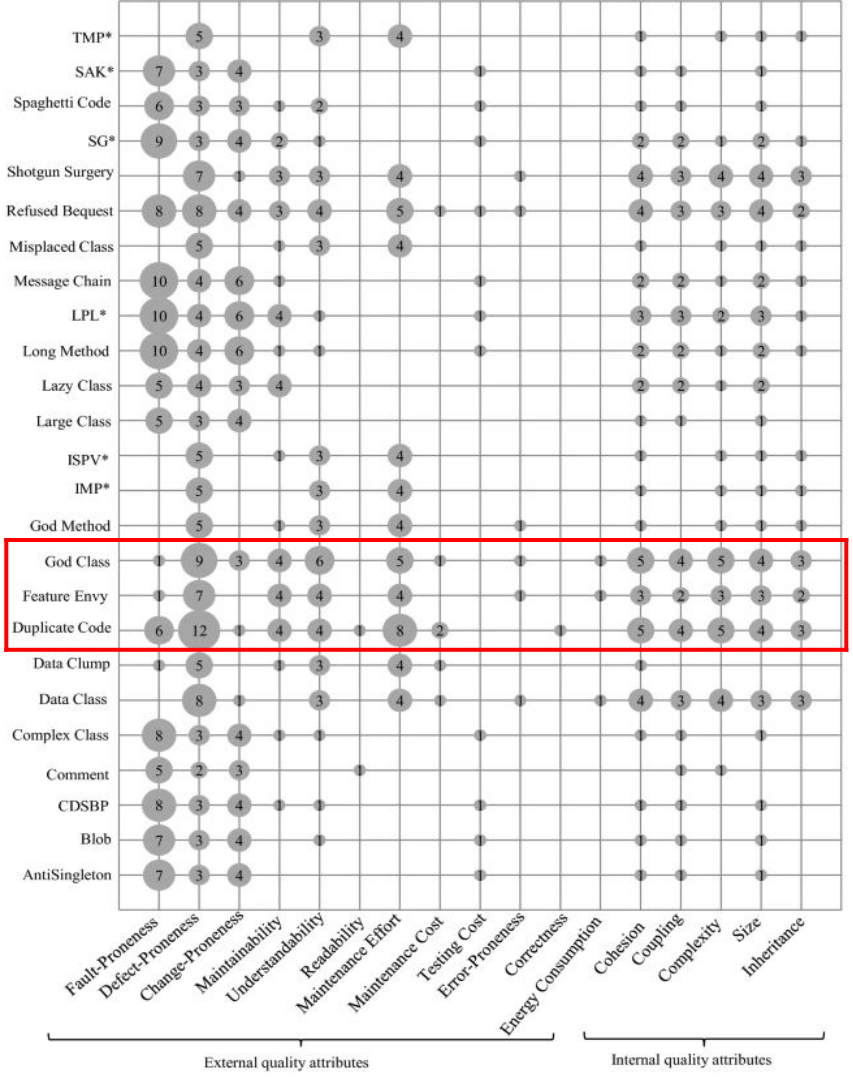
A Systematic Literature Review on Empirical Analysis of the Relationship Between Code Smells and Software Quality Attributes (Amandeep Kaur, ACME 2020)



A Systematic Literature Review on Empirical Analysis of the Relationship Between Code Smells and Software Quality Attributes (Amandeep Kaur, ACME 2020)



A Systematic Literature Review on Empirical Analysis of the Relationship Between Code Smells and Software Quality Attributes (Amandeep Kaur, ACME 2020)



An experimental investigation on the innate relationship between quality and refactoring (Bavota et al., IST 2015)

- Investigar se as atividades de refatoração ocorrem em componentes de código para os quais determinados indicadores sugerem que pode haver necessidade de operações de refatoração
  - **Qtd de sistemas:** 3
  - **code smells:** *Class data should be private, Complex class, Feature envy, Blob class, Lazy class, Long method, Long parameter list, Message chain, Refused bequest, Spaghetti code, Speculative generality*
  - **Métricas utilizadas:** LOC, WMC, DIT, NOC, RFC, CBO, LCOM, NOA, NOO, CCBC, C3

Impacto na qualidade:

- Na maioria das vezes, as métricas de qualidade não mostram uma relação clara com a refatoração

How Does Incomplete Composite Refactoring Affect Internal Quality Attributes? (Bibiano et. al., ICPC 2020)

- Identificar as formas mais comuns de composições incompletas e seus efeitos nos atributos de qualidade
  - **Qtd de sistemas:** 5
  - **code smells:** *Feature Envy e God Class*
  - **Métricas utilizadas:** LCOM2, CBO, MAxNEST, CC, WMC, LOC, CLOC, STMTC, NIV, NIM

Impacto na qualidade:

- Refatorações compostas incompletas tendem a não alterar os atributos de qualidade internos em classes que possuem *code smells*
- No entanto, apesar das composições incompletas não removerem totalmente os code smells, eles mantêm a qualidade estrutural interna das classes afetadas.

Artigos do LEAN

Investigando o Impacto das Coocorrências de *Code Smells* nos Atributos Internos de Qualidade

Júlio Serafim Martins[1], **Carla I. M. Bezerra**[1]

[1]Programa de Pós-Graduação em Computação (PCOMP)
Universidade Federal do Ceará (UFC) – Quixadá – CE – Brasil
Ingresso: 02/2019 - Defesa: 09/2021

Are Code Smell Co-occurrences Harmful to Internal Quality Attributes?: A Mixed-Method Study

Authors: Júlio Martins, Carla Bezerra, Anderson Uchôa, Alessandro Garcia Authors Info & Claims

SBES '20: Proceedings of the 34th Brazilian Symposium on Software Engineering • October 2020 • Pages 52–61 • https://doi.org/10.1145/3422392.3422419

Code Smell Co-occurrences: A Systematic Mapping

Antonio Neto
Federal University of Ceara
Quixadá, CE, Brazil
antonioxavierdesousaneto@gmail.com

Carla Bezerra
Federal University of Ceara
Quixadá, CE, Brazil
carlailane@ufc.br

Júlio Martins
Federal University of Ceara
Quixadá, CE, Brazil
juliomserafim@gmail.com

How do Code Smell Co-occurrences Removal Impact Internal Quality Attributes? A Developers' Perspective

Authors: Júlio Martins, Carla Bezerra, Anderson Uchôa, Alessandro Garcia Authors Info & Claims

SBES '21: Brazilian Symposium on Software Engineering • September 2021 • Pages 54–63 • https://doi.org/10.1145/3474624.3474642

Analyzing the Impact of Inter-smell Relations on Software Maintainability: An Empirical Study with Software Product Lines

Authors: Júlio Martins, Carla Ilane Moreira Bezerra, Anderson Uchôa Authors Info & Claims

SBSI'19: Proceedings of the XV Brazilian Symposium on Information Systems • May 2019 • Article No.: 44 • Pages 1–8 • https://doi.org/10.1145/3330204.3330254

Are Code Smell Co-occurrences Harmful to Internal Quality Attributes?: A Mixed-Method Study (Martins et. al., SBES 2020)

- Investigar as coocorrências de code smells mais prejudiciais nos atributos internos de qualidade: coesão, acoplamento, complexidade, herança e tamanho
  - **Qtd de sistemas:** 3 *closed-source*
  - **Code smells:** 7 (JSPirit e JDeodorant)
  - **Métricas utilizadas:** LCOM2, CBO, ACC, SCC, MAxNEST, EVG, NOC, DIT, IFANIN, LOC, CLOC, CDL, NIM

Impacto na qualidade:

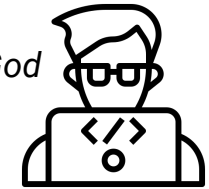
- *Coocorrências que mais ocorreram: God Class–Long Method, Disperse Coupling–Long Method, Feature Envy–Long Method, Disperse Coupling–Feature Envy, Feature Envy–God Class*
- *Coocorrências mais difíceis de refatorar*: *God Class–Long Method e Disperse Coupling–Long Method*
- A remoção das cooocorrências teve impacto positivo de ~20% na complexidade e impacto negativo na coesão e acoplamento

How do Code Smell Co-occurrences Removal Impact Internal Quality Attributes? A Developers' Perspective (Martins et. al., SBES 2021)

- Investigar o impacto da remoção das coocorrências de code smells sob a perspectiva dos desenvolvedores e dos atributos internos de qualidade: coesão, acoplamento, complexidade e herança
  - **Qtd de sistemas:** 5 *closed-source*
  - **Code smells:** 6 (JSPirit e JDeodorant)
  - **Métricas utilizadas:** LCOM2, LCOM3, CBO, FANIN, FANOUT, WMC, SCC, MAxNEST, EVG, NOC, IFANIN, NOC, DIT

Impacto na qualidade:

- A remoção das coocorrências *Disperse Coupling-God Class e God Class-Long Method* melhoraram todos os atributos de qualidade
- As coocorrências mais prejudiciais de acordo com os desenvolvedores são: *Dispersed Coupling–God Class, Feature Envy–God Class e God Class–Long Method*
- A remoção de *Feature Envy-Long Method* sugere um efeito negativo na coesão, acomplamento e complexidade



Code Smell Co-occurrences: A Systematic Mapping (NETO et. al., SBES IIER 2022)

- Mapeamento sistemático sobre o estado da arte de coocorrências de *code smells*
- Principais achados:
  - 13 estudos foram analisados
  - *Code smells* que mais ocorrem: *Feature Envy, Long Method, Long Parameter List, God Class, Data Class Should Be Private (DCSBP) e Shotgun Surgery*
  - As coocorrências mais comuns são DCSBP + FE e FE + LM
  - Em geral, os estudos mostram que as coocorrências prejudicam os atributos de qualidade
  - A maioria dos estudos detecta coocorrências de forma manual
  - Nenhum estudo apresentou métodos de refatoração mais adequados para refatoração de coocorrências
  - A remoção dessas coocorrências pode afetar positivamente a qualidade do código

Oportunidades de Pesquisa em Refatoração de *Code Smells*

- Avaliar o impacto da remoção de *code smells* em outras linguagens e também em projetos industriais
- Desenvolver uma ferramenta para detectar e visualizar coocorrências de *code smells* automaticamente
- Usar técnicas de aprendizagem de máquina para investigar como *code smells* foram introduzidos
- Investigar o impacto da remoção de coocorrências de *code smells* para manutenção, arquitetura de software, modularidade e compreensão
- Identificar critérios eficientes para classificar coocorrências de *code smells* para priorizar problemas críticos de *design*

Oportunidades de Pesquisa em Refatoração de *Code Smells*

- Investigar como coocorrências de *code smells* se relacionam entre si ao longo da evolução do software
- Investigar as principais estratégias de refatoração usadas para remover coocorrências de *code smells*

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Obrigada!

carlailane@ufc.br

Lean: https://leanresearchlab.github.io/quixada/

Referências

LANZA, M.; MARINESCU, R. 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.

FOWLER, M. Refactoring: improving the design of existing code. [S. l.]: Addison-Wesley Professional, 2018.

YAMASHITA, A.; MOONEN, L. Exploring the impact of inter-smell relations on software maintainability: An empirical study. In: IEEE. 35th ICSE. [S. l.], 2013. p. 682–691.

YAMASHITA, A.; ZANONI, M.; FONTANA, F. A.; WALTER, B. Inter-smell relations in industrial and open source systems: A replication and comparative analysis. In: IEEE. 31st ICSME. [S. l.], 2015. p. 121–130.

PIETRZAK, B.; WALTER, B. Leveraging code smell detection with inter-smell relations. Extreme Programming and Agile Processes in Software Engineering, Springer, p. 75–84, 2006.

SOBRINHO, E. V. de P.; LUCIA, A. D.; MAIA, M. de A. A systematic literature review on bad smells—5 w's: which, when, what, who, where. IEEE Trans. Softw. Eng., IEEE, 2018.

TARWANI, S.; CHUG, A. Sequencing of refactoring techniques by greedy algorithm for maximizing maintainability. In: IEEE. Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI). [S. l.], 2016. p. 1397–1403.