

CURSO DE ENGENHARIA DE SOFTWARE

RELATÓRIO – TRABALHO FINAL QUALIDADE DE SOFTWARE Canarinho

Equipe:

Eduardo Henrique Brito da Silva

537777

Francisco Jerônimo da Silva Júnior

433399

Professora:

Carla Ilane Moreira Bezerra

QUIXADÁ

Junho, 2022

SUMÁRIO

1	DESCRIÇÃO DO PROJETO	2
	AVALIAÇÃO DO PROJETO	
2.1	Medição 1 – Antes de refatorar o projeto	2
2.2	Detecção dos Code Smells	3
	Medição 2 – Após Refatorar Code Smell X	
	Medição 3 – Após Refatorar Code Smell Y	
2.5	Medição Z – Após a refatoração de todos os code smells do projeto	4
3	COMPARAÇÃO DOS RESULTADOS	4
REI	FERÊNCIAS	4
ΛDÍ	ÊNDICE A	-

1 DESCRIÇÃO DO PROJETO

Nessa Seção deve ser descrito o projeto. Importante incluir uma Tabela com as seguintes informações: Número de Linhas de Código (LOC), Número de Classes e Número de Releases. Incluir o link do github do projeto. Falar sobre a natureza do projeto, se é JAVA OO, se é de código aberto ou fechado, quais tecnologias utilizadas no projeto.

Link do projeto: https://github.com/concretesolutions/canarinho.git

Esta biblioteca é um conjunto de utilitários para trabalhar com padrões brasileiros no Android, por exemplo: validação e formatação de CPF, CNPJ e outros cadastros comuns no Brasil. Inspirado em: https://github.com/caelum/caelum-stella. O foco aqui é o Android. Portanto, não é compatível com aplicações Java puras.

ProjetoLOC# de classes# de releasesCanarinho2555622.0.2

Tabela 1 – Características do Projeto

2 AVALIAÇÃO DO PROJETO

2.1 Medição 1 – Antes de refatorar o projeto

Nessa Seção deve ser incluída a Tabela com a medição das métricas de coesão, acoplamento, complexidade, herança e tamanho, antes do projeto ser refatorado. Para isso será utilizada a ferramenta Understand. A Tabela 2 apresenta a descrição das métricas, faça uma tabela similar.

Tabela 2 – I	Medição	dos a	tributos	antes	de refatorar	o projeto.

Sistema	Coesão	Complexidade			Hera	Herança Acoplamento			Tamanho				
	LCOM2	ACC	SCC	EVG	MaxNet	DIT	NOC	IFANIN	СВО	LOC	CLOC	NIM	CDL
S1 antes da	1071	412	140	1573	184	288	34	96	90	10122	2231	134	739
refatoração	10.1			20.0	10.	200				10122	2201		
S1 após													
refat. CS X													
S1 após													
refat. CS X													

Tabela 3 – Métricas dos atributos internos de qualidade (MCCABE, 1976; CHIDAMBER; KEMERE, 1994; LORENZ; KIDD, 1994; DESTEFANIS et al., 2014)

Atributos | Atributos | Conservation | Atributos | Conservation | Conserva

Coesão (CHIDAMBER, RÉMERER, 1994)

Acoplamento Complexis (CBO) (CHIDAMBER, RÉMERER, 1994)

Acoplamento Complexis (CBO) (CHIDAMBER, RÉMERER, 1994)

Complexidade Average Cyclomatic Complexity (ACC) (MCCABE, 1976)

Sum Cyclomatic Complexity (SCC) (MCCABE, 1976)

Nomero de classes que uma classe está acoplada de classes e métodos.

Sum Cyclomatic Complexity (ACC) (MCCABE, 1976)

Somatório da complexidade ciclomática de todos os métodos.

Nesting (MaxNest) (LORENZ, KIDD, 1994)

Nível máximo de aninhamento de construções de controle. Quanto maior o valor dessa métrica, mais complexos são as classes e métodos.

Nesting (MaxNest) (LORENZ, KIDD, 1994)

Nível máximo de aninhamento de construções de controle. Quanto maior o valor dessa métrica, maior é a complexidade ciclomática de todos os métodos.

Nível máximo de aninhamento de construções de controle. Quanto maior o valor dessa métrica, maior é a complexidade ciclomática de todos os métodos.

Resting (MaxNest) (LORENZ, KIDD, 1994)

Nível máximo de aninhamento de construções de controle. Quanto maior o valor dessa métrica, maior é a complexidade ciclomática de todos os métodos.

Resting (MaxNest) (LORENZ, KIDD, 1994)

Nível máximo de aninhamento de construções de controle. Quanto maior o valor dessa métrica, maior é a complexidade ciclomática de todos os métodos.

Number Of Children (NOC) (CHIDAMBER; KEMERER, 1994)

Depth of Inheritance Tree (DIT) (CHIDAMBER; KEMERER, 1994)

Bases Classes (IFANIN) (DESTEFANIS et al., 2014)

Lines with Comments (CLOC) (LORENZ; KIDD, 1994)

Lines of Code (LOC) (LORENZ; KIDD, 1994)

Classes (CDL) (LORENZ; KIDD, 1994) Instance Methods (NIM) (LORENZ; KIDD, 1994)

2.2 Detecção dos Code Smells

Tamanho

Nessa Seção deve ser indicado quais e quantos code smells foram detectados no projeto. Faça uma Tabela indicando os code smells detectados pela ferramenta JSPirit e quantos code smells para cada tipo foram detectados.

Tabela 3 – Code smells do projeto.

cuartorium. Quanto maior o valor dessa métrica mais complexas são as classes e métodos.

Quanto maior o valor dessas métrica maior é o grau de herança de un sistema.

O número de neveis que uma subclasses heréa de métodos e O número de nas supreciseos en árvore de herança.

Quanto maior o valor dessa métrica maior é o grau de herança de um sistema.

Número imediato de classes base. Quanto maior o valor dessa métrica, maior o grau de herança de um sistema. Número de linhas de código, excluindo espaços e comentátios.

rios. Quanto maior o valor dessa métrica, maior é o tamanho do

Quanto maior o valor dessa métrica maior o tamanho do sis terna.

Número de classes. Quanto maior o valor, maior o tamanho do sistema.

Número de subclasses de uma classe

Número de linhas com comentários

Nome do Code Smell	Quantidade

2.3 Medição 2 – Após Refatorar Code Smell X

Nessa Seção você deve indicar os valores de todas as métricas da Tabela 2, após refatorar um determinado code smell. Esse code smell deve ser totalmente refatorado até não ser mais detectado pela JSPirit. Você deve também incluir a técnica de refatoração utilizada para retirar o code smell. Isso deve ser feito para cada code smell detectado no projeto. Após a refatoração de cada code smell deve ser realizada uma nova medição na ferramenta Understand. Deve ser realizada também uma análise dos 5 atributos de qualidade e que métricas pioram ou melhoram de acordo com a retirada desses code smells.

2.4 Medição 3 – Após Refatorar Code Smell Y

.

2.5 Medição Z – Após a refatoração de todos os code smells do projeto

Após todos os code smells refatorados, deverá ser realizada a medição final do projeto conforme as métricas da Tabela 2. Deve também ser feita a análise final se as métricas pioraram ou melhoraram de acordo com a retirada dos code smells.

3 COMPARAÇÃO DOS RESULTADOS

Leia o artigo:

https://www.sciencedirect.com/science/article/pii/S0950584920301142?casa_token=xcwL1B waRFUAAAAA:wZjXB0Wx-

 $\underline{0FiMSpZSzyi0b7iRe7ZJOr8FdwihzEkvzeQHh0Iz6mxPCF769JgRiZ69TyfI518BP0}$

Faça uma comparação dos resultados do seu projeto de acordo com esse artigo.

REFERÊNCIAS

AZEEM, Muhammad. Machine learning techniques for code smell detection: A systematic literature review and meta-analysis. Information and Software Technology, v. 108, p. 115-138, 2019.

SABIR, Fatima. A systematic literature review on the detection of smells and their evolution in object-oriented and service-oriented systems. Software: Practice and Experience, v. 49, n. 1, p. 3-39, 2019.

APÊNDICE A

Incluir possíveis documentos que possam ser gerados no desenvolvimento do sistema.