UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ

CURSO DE ENGENHARIA DE SOFTWARE

RELATÓRIO – TRABALHO FINAL QUALIDADE DE SOFTWARE
State Progress Bar

Equipe:

Douglas da Silva Holanda

Weslley Itallo Vieira da Silva

Professora:

Carla Ilane Moreira Bezerra

QUIXADÁ

Junho, 2021

SUMÁRIO

1 DESCRIÇÃO DO PROJETO

StateProgressBar é uma biblioteca Android para exibir e transitar entre os vários status de uma barra de progresso. Desenvolvida por Kofi Gyan, a biblioteca oferece diversas funcionalidades para barras de progresso, podendo ser utilizada para personalizar da melhor maneira. O projeto possui 2265 linhas de código Java e 1173 linhas de código xml.
Link do projeto: https://github.com/kofigyan/StateProgressBar

Tabela 1 – Características do Projeto

Projeto	**LOC**	**# de classes**	**# de releases**
StateProgressBar	2165	57	1

2 AVALIAÇÃO DO PROJETO

2.1 Medição 1 – Antes de refatorar o projeto

Tabela 2 – Medição dos atributos antes de refatorar o projeto.

Sistema	**Coesão**	**Complexidade**	**Herança**			**Acoplamento**	**Tamanho**			
	LCOM	CC	DIT	NOC	FANIN	CBO	LOC	CLOC	NIM	CDL
Antes da refatoração	-187	401	33	0	55	199	1721	444	287	57

Tabela 3 - Métricas do Designite que não foram relacionadas com as do Understand.

Sistema	NOF	NOPF	NOPM	WMC	FANOUT	CC	PC
Antes da refatoração	157	12	175	401	55	401	274

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

Atributos	Métricas	Descrição
Coesão	*Lack of Cohesion of Methods (LCOM2) (CHIDAMBER; KEMERER, 1994)*	Mede a coesão de uma classe. Quanto maior o valor dessa métrica, menos coesiva é a classe.
Acoplamento	*Coupling Between Objects (CBO) (CHIDAMBER; KEMERER, 1994)*	Número de classes que uma classe está acoplada Quanto maior o valor dessa métrica, maior é o acoplamento de classes e métodos.
Complexidade	*Average Cyclomatic Complexity (ACC) (MCCABE, 1976)*	Média da complexidade ciclomática de todos os métodos. Quanto maior o valor dessa métrica, mais complexa são a classes e métodos.
	Sum Cyclomatic Complexity (SCC) (MCCABE, 1976)	Somatório da complexidade ciclomática de todos os métodos. 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 de classes e métodos.
	Essential Complexity (EVG) (MCCABE, 1976)	Mede o grau na qual um módulo contém construtores não estruturados. Quanto maior o valor dessa métrica mais complexas são as classes e métodos.
Herança	*Number Of Children (NOC) (CHIDAMBER; KEMERER, 1994)*	Número de subclasses de uma classe. Quanto maior o valor dessa métrica maior é o grau de herança de un sistema.
	Depth of Inheritance Tree (DIT) (CHIDAMBER; KEMERER, 1994)	O número de níveis que uma subclasse herda de métodos e atributos de uma superclasse na árvore de herança. Quanto maior o valor dessa métrica maior é o grau de herança de um sistema.
	Bases Classes (IFANIN) (DESTEFANIS et al., 2014)	Número imediato de classes base. Quanto maior o valor dessa métrica, maior o grau de herança de um sistema.
Tamanho	*Lines of Code (LOC) (LORENZ; KIDD, 1994)*	Número de linhas de código, excluindo espaços e comentários. Quanto maior o valor dessa métrica, maior é o tamanho do sistema.
	Lines with Comments (CLOC) (LORENZ; KIDD, 1994)	Número de linhas com comentários. Quanto maior o valor dessa métrica maior o tamanho do sistema.
	Classes (CDL) (LORENZ; KIDD, 1994)	Número de classes. Quanto maior o valor , maior o tamanho do sistema.
	Instance Methods (NIM) (LORENZ; KIDD, 1994)	Número de métodos de instância. Quanto maior o valor dessa métrica maior é o tamanho do sistema.

2.2 Detecção dos Code Smells

Tabela 3 – Code smells do projeto.

Nome do Code Smell	Quantidade
Long Identifier	26
Magic Number	58
Long Statement	40
Unutilized Abstraction	36
Complex Method	2

2.3 Medição 2 – Após Refatorar Magic Number

Definimos cada um dos Magic Numbers como enum, limitando os possíveis valores para essa propriedade, eliminando ainda as chances de erros de digitação. Removemos 10 ocorrências de Magic Number no sistema.

Fazendo uma análise direta, comparando as tabelas 1 e 3, notamos que as métricas de Coesão e Complexidade melhoraram. Já que a métrica de coesão aumentou em 10 e a complexidade diminuiu em 20. A métrica de herança sofreu alteração no número de classes base, indicando menor grau de herança no sistema. Não sabemos exatamente como essa métrica foi afetada, já que a técnica implicada consistia apenas na substituição de uma variável.

Com relação às outras métricas não houve diferença, apenas no tamanho onde as linhas de código contabilizadas diminuiu, porém nós acreditamos que possa ser um problema da ferramenta Designite, já que nós adicionamos algumas linhas e não removemos nenhuma.

Tabela 3 – Medição dos atributos após a primeira refatoração.

Sistema	**Coesão**	**Complexidade**	**Herança**			**Acoplamento**	**Tamanho**			
	LCOM	CC	DIT	NOC	FANIN	CBO	LOC	CLOC	NIM	CDL
1º Refatoração	-177	381	30	0	47	199	1656	444	287	57

Tabela 4 - Métricas do Designite que não foram relacionadas com as do Understand.

Sistema	NOF	NOPF	NOPM	WMC	FANOUT	CC	PC
1º Refatoração	157	12	175	381	55	401	274

2.4 Medição 3 – Após Refatorar Complex Method

Nossa primeira medição do sistema identificou 2 smells do tipo Complex Method, sendo que enquanto refatorávamos alguns smells do tipo Magic Number, que estavam no mesmo arquivo que um Complex Method, um deles acabou “sumindo” na nova medição. Pesquisando sobre, identificamos que para solucionar esse smell seria necessário, nesse caso, diminuir a quantidade de casos no switch, while e IFs, utilizando métodos pequenos e reduzindo as declarações. Porém acreditamos que como foi constatado anteriormente que a complexidade diminuiu, então os casos do switch não ficaram suficientemente complexos para ser considerado um Complex Method.

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=xcwL1BwaRFUAAAAA:wZjXB0Wx-0FiMSpZSzyi0b7iRe7ZJOr8FdwihzEkvzeQHh0Iz6mxPCF769JgRiZ69TyfI5l8BP0

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.

NÚMEROS mágicos: O que são e como corrigir. DevMedia. Disponível em: <https://www.devmedia.com.br/numeros-magicos-o-que-sao-e-como-corrigir/38711>. Acesso em: 18 de jun. de 2022.

REDUCING Cyclomatic Complexity and NPath Complexity: Steps for Refactoring. Axelerant, SHASHIKANTH REDDY, AXELERANT ALUMNI. Disponível em: <https://www.axelerant.com/blog/reducing-cyclomatic-complexity-and-npath-complexity-steps-for-refactoring>. Acesso em: 18 de jun. de 2022.

APÊNDICE A

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