



**UNIVERSIDADE
FEDERAL DO CEARÁ**
CAMPUS DE QUIXADÁ

CURSO DE ENGENHARIA DE SOFTWARE

**RELATÓRIO – TRABALHO FINAL QUALIDADE DE SOFTWARE
CONTEST**

Equipe:

Ian Mateus Torres Pompeu

Professora:

Carla Ilane Moreira Bezerra

QUIXADÁ

Março, 2021

SUMÁRIO

1	DESCRIÇÃO DO PROJETO	2
2	AVALIAÇÃO DO PROJETO	2
2.1	Medição 1 – Antes de refatorar o projeto.....	2
2.2	Detecção dos Code Smells	4
2.3	Medição 2 – Após Refatorar Code Smell Feature Envy	5
2.4	Medição 3 – Após Refatorar Code Smell Shotgun Surgery	6
2.5	Medição 4 – Após Refatorar Code God Class	7
2.6	Medição 5 – Após Refatorar Code Intensive Coupling	8
2.7	Medição 6 – Após Refatorar Code Dispersed Coupling.....	10
2.8	Visão final das refatorações	11
3	COMPARAÇÃO DOS RESULTADOS	12
	REFERÊNCIAS	14
	APÊNDICE A	15

1 DESCRIÇÃO DO PROJETO

Constest, é um sistema de gerenciamento de eventos e submissões de produções acadêmicas da UFC campus Quixadá. Auxiliando os organizadores de eventos da UFC Quixadá a terem um controle de submissões, avaliações e certificações dos participantes.

Link do projeto: <https://github.com/ianmateusES/contest>

Autores: Júlio Serafim

Tabela 1 – Características do Projeto

Projeto	LOC	# de classes	# de releases
CONTEST	4.946	82	?

2 AVALIAÇÃO DO PROJETO

Para realização das avaliações do sistema, foram utilizadas as ferramentas Understand para realização das métricas de qualidade, JSpiRIT para identificação dos code smells e a IDE Eclipse para realização das refatoração.

2.1 Medição 1 – Antes de refatorar o projeto

Para realização de uma análise de melhoria do sistema, foi realizada uma medição antes das refatorações. Assim sendo possível a verificação de melhoria ou piora das métricas de qualidade do sistema. Para entendimento das métricas, pode-se analisar as descrições das métricas na tabela da Figura 1.

Figura 1 - Medição dos atributos internos de qualidade (MACCABE, 1976; CHIDAMBER; KEMERER, 1994; LORENZ; KIDD, 1994; DESTEFANIS *es al.*, 2014).

Atributos	Métricas	Descrição
Coesão	<i>Lack of Cohesion of Methods (LCOM2)</i> (CHIDAMBER; KEMERER, 1994)	Mede a coesão de uma classe. Quanto maior o valor dessa métrica, menos coesiva é a classe.
Acoplamento	<i>Coupling Between Objects (CBO)</i> (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	<i>Average Cyclomatic Complexity (ACC)</i> (MCCABE, 1976)	Média da complexidade ciclomática de todos os métodos. Quanto maior o valor dessa métrica, mais complexa são a classes e métodos.
	<i>Sum Cyclomatic Complexity (SCC)</i> (MCCABE, 1976)	Somatório da complexidade ciclomática de todos os métodos. Quanto maior o valor dessa métrica, mais complexos são as classes e métodos.
	<i>Nesting (MaxNest)</i> (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.
	<i>Essential Complexity (EVG)</i> (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	<i>Number Of Children (NOC)</i> (CHIDAMBER; KEMERER, 1994)	Número de subclasses de uma classe. Quanto maior o valor dessa métrica maior é o grau de herança de um sistema.
	<i>Depth of Inheritance Tree (DIT)</i> (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.
	<i>Bases Classes (IFANIN)</i> (DESTEFANIS et al., 2014)	Número imediato de classes base. Quanto maior o valor dessa métrica, maior o grau de herança de um sistema.
Tamanho	<i>Lines of Code (LOC)</i> (LORENZ; KIDD, 1994)	Número de linhas de código, excluindo espaços e comentários. Quanto maior o valor dessa métrica, maior é o tamanho do sistema.
	<i>Lines with Comments (CLOC)</i> (LORENZ; KIDD, 1994)	Número de linhas com comentários. Quanto maior o valor dessa métrica maior o tamanho do sistema.
	<i>Classes (CDL)</i> (LORENZ; KIDD, 1994)	Número de classes. Quanto maior o valor , maior o tamanho do sistema.
	<i>Instance Methods (NIM)</i> (LORENZ; KIDD, 1994)	Número de métodos de instância. Quanto maior o valor dessa métrica maior é o tamanho do sistema.

A Tabela 2 apresenta as medições das métricas com seus respectivos valores. Os valores das métricas antes a refatoração ficaram: Coesão, Complexidade, Herança, Acoplamento e Tamanho são no total de 3.875, 1.162, 189, 525, 6.329 respectivamente.

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

Qualidades	Métricas	Valores
COESÃO	LCOM	3.875
COMPLEXIDADE	Average Cyclomatic	87

	Sum Cyclomatic	852
	Essential	144
	Nesting	79
HERANÇA	DIT	97
	NOC	2
	Base Classes	90
ACOPLAMENTO	CBO	525
TAMANHO	LOC	4.946
	CLOC	665
	Número de Métodos	636
	Número de Classes	82

2.2 Detecção dos Code Smells

Para a detecção de code smell como já foi citado, ferramenta JSpiRIT, foram encontradas 58 instâncias de code smell de 5 tipos diferentes: Feature Envy, Shotgun Surgery, God Class, Dispersed Coupling e Intensive Coupling. A Tabela 3, demonstra a distribuição das instancias para cada code smell detectado pela ferramenta.

Tabela 3 – Code smells do projeto.

Code Smells	Quantidade
Feature Envy	36
Shotgun Surgery	8
God Class	2
Intensive Coupling	1
Dispersed Coupling	11
Total	58

Dada a identificação dos code smell a Tabela 4, apresenta como o sistema ficou depois da refatoração e quantos code smell foram removidos de cada tipo. Uma observação importante sobre o experimento é que tinha a restrição de fatorar pelo menos 40 instâncias dos code smells e que fossem de pelo menos 5 tipos diferentes. Foram refatorados 40 instâncias dos seguintes code smells: Feature Envy, Shotgun Surgery, Intensive Coupling, God Class e Dispersed Coupling.

Tabela 4 – Refatoração dos code smell.

Code Smell	Quantidade	Depois refatoração	Removidos
Feature Envy	36	7	29
Shotgun Surgery	8	4	4
God Class	2	0	2
Intensive Coupling	1	0	1
Dispersed Coupling	11	7	4
Total	58	18	40

2.3 Medição 2 – Após Refatorar Code Smell Feature Envy

Foram refatoradas 27 instâncias de 36 dos Feature Envy detectados pelo JSpiRIT. Para a refatoração foram utilizados os métodos *Extract method* e *Move field*. A Tabela 5 apresenta as medições das métricas com seus respectivos valores. Os valores das métricas após a refatoração desse code smell ficaram: Coesão, Complexidade, Herança, Acoplamento e Tamanho são no total de 4.081, 1.223, 189, 525, 6.559 respectivamente.

Tabela 5 – Medição dos atributos depois da refatorar do code smell Feature Envy.

Qualidades	Métricas	Valores
COESÃO	LCOM	4.081
COMPLEXIDADE	Average Cyclomatic	83
	Sum Cyclomatic	915
	Essential	147

	Nesting	78
HERANÇA	DIT	97
	NOC	2
	Base Classes	90
ACOPLAMENTO	CBO	525
TAMANHO	LOC	5.132
	CLOC	662
	Número de Métodos	683
	Número de Classes	82

Houve um aumento na métrica LCOM por conta do Extract method, para diminuir a quantidade de responsabilidade do método, foi delegada essa funcionalidade a um método diferente da classe que poderia ser reaproveitado por todos os métodos da classe. Com essa separação dos métodos, houve um aumento da complexidade total de 1.162 para 1.223, mas houve uma diminuição na complexidade da média ciclomatica das classes. A Herança e o acoplamento se mantiveram por não haver mudanças na delegação de responsabilidade e nem utilização das chamadas de métodos de outras classes. O Tamanho, como podemos ver houve o número de métodos aumentadas por conta do Extract method.

O nível de dificuldade para refatoração do Feature Envy é muito baixa, considerando que foi só a desestruturação do método para delegação de responsabilidades únicas.

2.4 Medição 3 – Após Refatorar Code Smell Shotgun Surgery

Foram refatoradas 4 instâncias de 8 dos Shotgun Surgery detectados pelo JSPirit. Para a refatoração foram utilizados os métodos *Move method*, *Move field*, *Extract method* e *Extract class*. A Tabela 6 apresenta as medições das métricas com seus respectivos valores. Os valores das métricas antes a refatoração ficaram: Coesão, Complexidade, Herança, Acoplamento e Tamanho são no total de 4.107, 1.191, 195, 548 e 6.593 respectivamente.

Tabela 6 – Medição dos atributos depois da refatorar do code smell Shotgun Surgery.

Qualidades	Métricas	Valores
------------	----------	---------

COESÃO	LCOM	4.107
COMPLEXIDADE	Average Cyclomatic	84
	Sum Cyclomatic	879
	Essential	148
	Nesting	80
HERANÇA	DIT	100
	NOC	2
	Base Classes	93
ACOPLAMENTO	CBO	548
TAMANHO	LOC	5.161
	CLOC	662
	Número de Métodos	685
	Número de Classes	85

Houve um aumento na métrica LCOM por conta do *Extract method*, para diminuir a quantidade de responsabilidade do método, foi delegada essa funcionalidade a um método diferente da classe que poderia ser reaproveitado por todos os métodos da classe. Além disso, a complexidade, herança, acoplamento e tamanho aumentaram por conta do *Extract class*.

O nível de dificuldade para refatoração do Shotgun Surgery é muito alta, considerando que foi preciso identificar todos os pontos de chamadas do método em todo o sistema, estruturar uma estratégia de ponto único de chamada e modificar em todos os trechos de códigos.

2.5 Medição 4 – Após Refatorar Code God Class

Foram refatoradas 2 instâncias de 2 dos God Class detectados pelo JSPirit. Além disso, foram refatoradas 2 instâncias de Feature Envy por haver interseção de code smell nas God Class, na reorganização foram refatorados por transitividade. Para a refatoração foram utilizados os métodos *Extract Class* e *Move method*. A Tabela 7 apresenta as medições das métricas com seus respectivos valores. Os valores das métricas após a refatoração ficaram: Coesão, Complexidade, Herança, Acoplamento e Tamanho são no total de 4.217, 1.181, 199, 552 e 6.592 respectivamente.

Tabela 7 – Medição dos atributos depois da refatorar do code smell God Class.

Qualidades	Métricas	Valores
COESÃO	LCOM	4.217
COMPLEXIDADE	Average Cyclomatic	88
	Sum Cyclomatic	876
	Essential	140
	Nesting	77
HERANÇA	DIT	102
	NOC	2
	Base Classes	95
ACOPLAMENTO	CBO	552
TAMANHO	LOC	5.097
	CLOC	733
	Número de Métodos	675
	Número de Classes	87

Houve um aumento no Tamanho por conta do número de classes e uma diminuição Complexidade por delegação das responsabilidades para outra classe de funções específicas. Mas houve um aumento no acoplamento por a classe com o code smell God Class precisar das funcionalidades, por exemplo, para calcular os prazos de submissão dos eventos e prazos de envios de trabalhos.

O nível de dificuldade para refatoração do God Class é muito alta, considerando que é preciso entender qual deveria ser a responsabilidade única da classe e fazer o Extract class para classes que tenha responsabilidades únicas das partes retiradas do God Class.

2.6 Medição 5 – Após Refatorar Code Intensive Coupling

Foram refatoradas 1 instâncias de 1 dos Intensive Coupling detectados pelo JSPirit. Para a refatoração foi utilizado o método *Comentado*, função obsoleta (não chamada no projeto)

no sistema e extremamente complexa sua estrutura. Mas como estratégia de amenizar Intensive Coupling no método, foi realizada uma extração de métodos. A Tabela 8 apresenta as medições das métricas com seus respectivos valores. Os valores das métricas após a refatoração ficaram: Coesão, Complexidade, Herança, Acoplamento e Tamanho são no total de 4.217, 1.177, 199, 549 e 6.597 respectivamente.

Tabela 8 – Medição dos atributos depois da refatorar do code smell Intensive Coupling.

Qualidades	Métricas	Valores
COESÃO	LCOM	4.217
COMPLEXIDADE	Average Cyclomatic	85
	Sum Cyclomatic	879
	Essential	138
	Nesting	75
HERANÇA	DIT	102
	NOC	2
	Base Classes	95
ACOPLAMENTO	CBO	549
TAMANHO	LOC	5.067
	CLOC	769
	Número de Métodos	674
	Número de Classes	87

As métricas Coesão e Herança se mantiveram, enquanto a de Complexidade e Acoplamento diminuirão pôr o método ter deixado de existir na classe. Mas o Tamanho aumentou por conta do número de linhas comentadas, pôr o método ter sido totalmente comentado.

Para avaliação do nível de dificuldade para refatoração do Intensive Coupling, é preciso considerando que nesse sistema só havia uma instância desse code smell, o método estava obsoleto no projeto e estratégia foi comentar a função. Logo, não é possível definir um nível de dificuldade para a refatoração.

2.7 Medição 6 – Após Refatorar Code Dispersed Coupling

Foram refatoradas 4 instâncias de 11 dos Dispersed Coupling detectados pelo JSPirit. Para a refatoração foram utilizados os métodos *Move method* e *Extract method*. A Tabela 8 apresenta as medições das métricas com seus respectivos valores. Os valores das métricas após a refatoração ficaram: Coesão, Complexidade, Herança, Acoplamento e Tamanho são no total de 4228, 1.181, 199, 550 e 6.638 respectivamente.

Tabela 8 – Medição dos atributos depois da refatorar do code smell Dispersed Coupling.

Qualidades	Métricas	Valores
COESÃO	LCOM	4228
COMPLEXIDADE	Average Cyclomatic	87
	Sum Cyclomatic	881
	Essential	139
	Nesting	74
HERANÇA	DIT	102
	NOC	2
	Base Classes	95
ACOPLAMENTO	CBO	550
TAMANHO	LOC	5.098
	CLOC	769
	Número de Métodos	684
	Número de Classes	87

Houve um aumento na Complexidade por conta do Extract method, os vários métodos das classes se utilizavam de vários métodos repetido de várias outras classes. Logo, foi separado em métodos diferentes e reaproveitados em toda a classe, retirando códigos repetidos e dependência de muitas chamadas em um método só, havendo um aumento de Coesão e Complexidade.

O nível de dificuldade para refatoração do Dispersed Coupling é intermediária, considerando que foi preciso identificar todos os métodos utilizados e fazer um reaproveitamento de código na classe delegando responsabilidade únicas a cada método extraído.

2.8 Visão final das refatorações

Com a finalização das medições, podemos visualizar na Tabela 9 o histórico de evolução das métricas do sistema Contest ao decorrer das refatorações e como elas se comportam a cada estratégia utilizada para solucionar os code smells.

Tabela 9 – Histórico de medição dos atributos de qualidade do sistema.

Sistema	Coesão	Complexidade				Herança			Acoplam.	Tamanho			
	LCOM2	ACC	SCC	EVG	MaxNet	DIT	NOC	IFANIN	CBO	LOC	CLOC	NIM	CDL
S1 antes da refatoração	3.875	87	852	144	79	97	2	90	525	4.946	665	636	82
S1 após refat. CS Feature Envy	4.081	83	915	147	78	97	2	90	525	5.132	662	683	82
S1 após refat. CS Shotgun Surgery	4.107	84	879	148	80	100	2	93	548	5.161	662	685	85
S1 após refat. CS God Class	4.217	88	876	140	77	102	2	95	552	5.097	733	675	87
S1 após refat. CS Intensive Coupling	4.217	85	879	138	75	102	2	95	549	5.067	769	674	87
S1 após refat. CS Dispersive Coupling	4.228	87	881	139	74	102	2	95	550	5.098	769	684	87

Agora fazendo uma avaliação de como o sistema estava antes da refatoração e de como ficou depois. A Tabela 10 apresenta o comparativo das medições das métricas com seus respectivos valores. Houve um aumento nos atributos: Coesão, Herança, Acoplamento e

Tamanho no total de 9,11%, 10,71%, 4,76% e 32,36% respectivamente. E uma diminuição de 0,06% na complexidade do sistema.

Tabela 10 – Comparativo de antes e depois da refatoração

Atributos		Antes da refatoração	Depois da refatoração	Diferença (%)
Coesão	LCOM2	3875	4228	9,11%
Complexidade	ACC	87	87	0,00%
	SCC	852	881	3,40%
	EVG	144	139	-3,47%
	MaxNet	79	74	-6,33%
Herança	DIT	97	102	5,15%
	NOC	2	2	0,00%
	IFANIN	90	95	5,56%
Acoplamento	CBO	525	550	4,76%
Tamanho	LOC	4946	5098	3,07%
	CLOC	665	769	15,64%
	NIM	636	684	7,55%
	CDL	82	87	6,10%

3 COMPARAÇÃO DOS RESULTADOS

Podemos observar que de acordo com as estratégias utilizadas para refatoração de cada tipo de code smell, tivemos melhoras, piores e não alterações nos atributos de qualidade que foram medidos pela ferramenta Understand. A Tabela 11 apresenta as porcentagens de melhorias e piores das métricas.

Ao analisar a Tabela 11, podemos observar que a única melhoria que houve nos atributos de qualidade foi na Complexidade, pela diminuição das métricas *Essential* e *Nesting*, inferindo que as classes estavam bem mais estruturadas e com menos aninhamentos entre os métodos. O atributo de Tamanho, teve aumento significativo por conta do número de classes para delegação de responsabilidades, extrações de métodos para responsabilidades específicas e o aumento de linhas de códigos para escrita das classes e dos métodos. Houve um aumento nas métricas de Coesão, Herança e Acoplamento. A Coesão aumentou por conta da extração de métodos nas classes para reaproveitamento de código e funções para solução dos Feature Envy. A Herança aumentou por conta do número de classes bases. E o Acoplamento obteve seu aumento por conta da delegação de responsabilidade para novas classes na refatoração dos code smells God Class e Shotgun Surgery.

Tabela 11 – Melhorias e piores dos atributos de qualidade

Atributos de qualidade	Valores (%)
Coesão	9,11%
Complexidade	-0,06%
Herança	10,71%
Acoplamento	4,76%
Tamanho	32,36%

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.