Universidade de Brasília Faculdade UnB Gama

CURSO:	**Engenharia de Software**	
DISCIPLINA:	**Técnicas de Programação em Plataformas Emergentes**	**TURMA: 01**
SEMESTRE:	**2022-1**	
PROFESSOR:	**André Luiz Peron Martins Lanna**	**MATRÍCULA:**
ALUNOS:	**Antônio Aldísio de Sousa Alves Ferreira Filho**	**202028211**
	Antonio Ruan Moura Barreto	**180030272**
	Davi Antônio da Silva Santos	**150058462**
	Fernando Miranda Calil	**190106565**
	Guilherme Verissimo Cerveira Braz	**180018159**

Trabalho 3

1- Simplicidade

Descrição:

Uma das características mais importantes de um código bem escrito, a simplicidade é algo que pode ser fácil de entender. Basicamente se trata de um código simples, coerente e consistente, escrito de forma enxuta e de fácil legibilidade.

Relação da característica com mau-cheiros:

Essa característica está relacionada a muitos ao mau-cheiro de código muito longo (bloaters) definido por Fowler, que em sua teoria é um código que cresceu demais, na maioria das vezes porque suas responsabilidades não foram bem delimitadas durante o planejamento da aplicação.

Pelo menos uma operação de refatoração capaz de levar o projeto de código a ter a característica em análise :

Uma das operações que ajudam a atingir essa característica é a de Extrair Método pelo fato de simplificar a maneira que o código é escrito.

2- Elegância

Descrição:

A elegância incorpora os aspectos estéticos do design e muitas vezes anda de mãos dadas com a simplicidade. Basicamente significa que o código escrito não é confuso ou excessivamente complexo. Segundo o Goodliffe podemos identificar algumas características:

- O design não está repleto de casos especiais.
- Cada parte complementa as outras, adicionando algo valioso e distinto.
- Ele associa coisas semelhantes.
- Há uma pequena localidade de mudança: uma única e simples mudança em um lugar não leva a modificações do código em muitos outros lugares.

Relação da característica com mau-cheiros:

Essa característica se relaciona com o mau-cheiro inibidores de modificação, também conhecido com código espaguete. O código espaguete é aquele no qual, para alterar um ponto, precisamos também fazer alterações em diversos outros, o que torna a manutenção uma dor de cabeça.

Pelo menos uma operação de refatoração capaz de levar o projeto de código a ter a característica em análise :

Uma das operações que ajudam a atingir essa característica é a de Extrair Método e Parametrizar Método.

3- Modularidade (baixo acoplamento e alta coesão)

Descrição:

Um código considerado modular é o código construído de forma distribuída, utilizando de módulos, interfaces e componentes para diminuir a complexidade de cada módulo utilizado, fazendo com que cada módulo seja mais fácil de entender, testar e refatorar de forma independente das outras partes do código.

A qualidade da aplicação dessa característica está ligada a coesão e acoplamento do código. Podemos identificar erros na aplicação desta característica de código em métodos que possuem várias funcionalidades, funcionalidades não relacionadas ou a dependência de muitos outros métodos.

Relação da característica com mau-cheiros:

Essa característica se relaciona com diversos maus-cheiros definidos por Fowler, por exemplo o de código muito longo(Bloated) que em sua teoria são códigos que cresceram demais, na maioria dos casos porque suas responsabilidades não foram bem delimitadas durante o planejamento da aplicação.

Pelo menos uma operação de refatoração capaz de levar o projeto de código a ter a característica em análise :

Uma das operações que ajuda a atingir essa característica é a de *Extrair Método.*

4- Ausência de duplicidades

Descrição:

A duplicação é a inimiga do design simples e elegante, além de que redundâncias desnecessárias podem gerar um simples problema, achar um bug e corrigi-lo em um trecho do código, mas esquecer de corrigir outro, comprometendo a segurança do mesmo. Dessa forma, unificar trechos de código diferentes que fazem coisas similares, facilita na hora de corrigir problemas, além de tornar o código mais claro.

Relação da característica com mau-cheiros:

Essa característica está relacionada intrinsecamente com o mau-cheiro identificado por Fowler como código duplicado, pois esta define "Se o mesmo trecho de código aparecer em vários pontos do projeto, saiba que sua solução será melhor se você conseguir unificá-los.", além de ter ligação também com mau-cheiros como Classe grande, que também é um indício de código duplicado.

Pelo menos uma operação de refatoração capaz de levar o projeto de código a ter a característica em análise :

Uma das operações que ajuda a atingir essa característica é a de *Extrair Método e Extrair constante*

5 - Boa documentação

Descrição:

Uma boa documentação é fundamental para que um software possa evoluir rapidamente e com qualidade. Uma boa documentação consiste não somente em documentos

escritos à parte: ela exige cuidado ao se escrever o código, principalmente nos nomes das variáveis, nos comentários adequados e no fluxo de execução.

Relação da característica com mau-cheiros:

Essa característica está relacionada principalmente com os code smells de comentários excessivos e nomes misteriosos. Eles acabam ocorrendo justamente a essa falta de clareza no processo de desenvolvimento de software, sendo que geralmente são várias equipes trabalhando paralelamente no mesmo software, trazendo assim complexidade maior de lógica e padronização. Portanto é importante dispor de uma boa documentação do seu software para facilitar manutenções e evitar possíveis maus-cheiros como os citados anteriormente.

Pelo menos uma operação de refatoração capaz de levar o projeto de código a ter a característica em análise:

Umas das operações que ajudam a atingir essa característica é a de *Renomear Variável e Renomear Campo.*

Referências Bibliográficas

***Code Craft: The Practice of Writing Excellent Code*, No Starch Press, Incorporated, 2006. *ProQuest Ebook Central*.**

***Refactoring: Improving the design of Existing Code*, Pearson Education, Incorporated, 2019.**