



UNIVERSIDADE DA REGIÃO DE JOINVILLE - SC

<http://www.univille.edu.br/>

Disciplina: Engenharia de Requisitos de Software

Professor: GLAUCIO SCHEIBEL ([glaucio.scheibel@univille.br](mailto:glaucio.scheibel@univille.br))

Aluno: Alessandro Dos Santos Marques ([alessandromarques@univille.br](mailto:alessandromarques@univille.br))

## Análise de código fonte

### 0.1 BraveSoftware

#### 1. Avaliando a nota SQALE do projeto BraveSoftware?

##### a) Confiança

O programa possui 1000 "codesmells" e 34 "bugs" o que denota uma possível instabilidade no programa e um possível travamento inesperado.

##### b) Segurança

O programa possui uma vulnerabilidade exposta na última revisão.

##### c) manutenibilidade

Encontrada uma deficiência relativa a complexidade cognitiva, que elicitava pontuações a cada utilização de laços de comparação ou repetição, a regra para aceitação dessa pontuação é de 15, entretanto a pontuação chegou a 17.

##### d) Cobertura

No momento da criação dessa análise foi realizado um novo estudo pelo sonarCloud que denota um acompanhamento corrente para o software 10/11/2021.

##### e) Duplicações

No código existem 5530 linhas duplicadas, que representa 2.1% do total, ocorreu um decréscimo desse percentual em 1.4% comparado com o valor da última análise.

##### f) Tamanho

Não representa alongamento extremo ou desnecessário.

##### g) Complexidade

Possui 9 avisos relativos a complexidade.

#### 2. Qual a dívida técnica do projeto?

Existe uma implementação em uma marcação evidenciada que foi mantida na última análise.

3. Existem problemas blocker (impeditivo)? Se sim, quais foram?

Existe uma variável repetida em dois escopos diferentes que podem gerar um bug por concorrência de atuação, e existe um "hotspot" que pode ser utilizado através de um "shellScript".

## 0.2 Apache Struts 2

1. Avaliando a nota SQALE do projeto Apache Struts 2?

- a) Confiança

O programa passou em testes de segurança e funcionais, mostrando uma possível confiança, entretanto existe um "bug" relativo a um método de acesso que o compilador pode entender como sobrecarga, pois possui um acesso a uma variável com nome diferente do usual para um "getters/setters".

- b) Segurança

O programa passou nas regras de segurança.

- c) manutenibilidade

O programa passou em testes de manutenibilidade

- d) Cobertura

O ultimo teste executado do programa foi no dia 02/11/2021

- e) Duplicações

No código existem 6364 linhas duplicadas, que representa 3% do total.

- f) Tamanho

O programa não transgrediu as regras de tamanho.

- g) Complexidade

Existe um erro de complexidade que pontuou com 47 de um total aceitável de 15.

2. Qual a dívida técnica do projeto?

Em classes que implementam interfaces não foi colocada as anotações pertinentes.

3. Existem problemas blocker (impeditivo)? Se sim, quais foram?

Existem 94 "blockers", um deles é uma possível má interpretação do compilador com um método "input" em uma classe que pode ser confundido com um campo com mesmo nome na interface.

## 0.3 Mediawiki-core

1. Avaliando a nota SQALE do projeto mediawiki-core?

a) Confiança

Existem 81 "bugs" relatados com um aumento de 42 comparando a ultima analise, tambem existem 736 "code smells", 1 vulnerabilidade e 31 "hotspots" com um aumento de 14 da ultima comparação. Mas o software ainda passou nas regras estipuladas.

b) Segurança

Existe um redirecionamento visível pelo usuário, que pode não ser desejável para o desenvolvedor.

c) manutenibilidade

existe um comentário "tagado" com "Todo" que refere se a uma possível implementação futura e um alerta de complexidade cognitiva de 16 pontos para uma regra de 15, entretanto as regras relativas a manutenibilidade não foram infringidas.

d) Cobertura

Ultima analise feita dia 10/11/2021.

e) Duplicações

Existem 32730 linhas duplicadas que representam 2% do total, que teve um decréscimo em 2.3%.

f) Tamanho

Não houve regra descumprida com 431790 linhas de código.

g) Complexidade

Existe um alerta de complexidade cognitiva de 16 pontos para uma regra de 15.

2. Qual a dívida técnica do projetos?

Uma implementação futura "tagada" com um comentário "Todo".

3. Existem problemas blocker (impeditivo)? Se sim, quais foram?

Existem 14 "blockers", uma delas faz referência a uma implementação em "PHP" que usa variáveis dinamicamente renomeadas, que não estão diretamente ligadas a um erro mas que não participam de boas praticas de desenvolvimento, infringindo uma regra que solicita a exclusão de tais variáveis.

## 0.4 Questões genéricas aos 3 projetos analisados

1. item O que ele quer dizer quando detecta que a complexidade está muito alta?

Significa que o código pode ser simplificado retirando laços de repetição desnecessários ou comparações no código.

2. Qual o problema mais comum encontrado em cada projeto?

Problemas relativos a complexidade e Divida técnica.