

Um Estudo das Características de Qualidade de Sistemas Java

Matheus Machado de Oliveira Andrade¹, Rafael Parreira Chequer², Samara Martins Ferreira³, Victor Alexandre Peters Fonseca⁴

Instituto de Ciências Exatas e Informática
Pontifícia Universidade de Minas Gerais (PUC Minas)
Belo Horizonte – MG – Brasil

{500391, 1275759, 1328767, 1314281}@sga.pucminas.br

***Resumo.** Este laboratório analisa aspectos da qualidade de repositórios desenvolvidos na linguagem Java, correlacionando-os com características do seu processo de desenvolvimento, sob a perspectiva de métricas de produto calculadas através da ferramenta CK.*

1. Introdução

No desenvolvimento de sistemas open-source, a colaboração entre diversos desenvolvedores pode impactar a qualidade interna do código. A evolução dos atributos de qualidade, como modularidade, manutenabilidade e legibilidade, pode ser comprometida em um ambiente colaborativo. Para mitigar esses riscos, diversas abordagens modernas buscam aperfeiçoar o processo de desenvolvimento, incluindo práticas como revisão de código e análise estática com ferramentas de integração contínua/entrega contínua (CI/CD).

Este relatório tem como objetivo analisar aspectos da qualidade de repositórios desenvolvidos na linguagem Java, correlacionando-os com características do processo de desenvolvimento. Para isso, serão utilizadas métricas de produto calculadas através da ferramenta CK.

1.1. Questões de Pesquisa e Hipóteses Informais

RQ 01 - Qual a relação entre a popularidade dos repositórios e as suas características de qualidade?

Hipótese: Repositórios mais populares provavelmente apresentam métricas de qualidade mais baixas, tendo em vista que a popularidade tende a refletir um reconhecimento da qualidade do sistema pela comunidade. Ou seja, os índices de CBO, DIT e LCOM devem ser baixos demonstrando maior qualidade do sistema.

RQ 02 - Qual a relação entre a maturidade dos repositórios e as suas características de qualidade ?

Hipótese: Repositórios mais maduros tendem a ter qualidade superior, pois tiveram maior período evolutivo e consequentemente o amadurecimento do código. Sendo assim, as métricas de CBO, DIT e LCOM devem ser baixos.

RQ 03 - Qual a relação entre a atividade dos repositórios e as suas características de qualidade?

Hipótese: Repositórios mais ativos, ou seja, com um maior número de *releases* recentes, podem indicar um esforço contínuo para aprimorar e manter a qualidade do código. As métricas derivadas do CK: CBO, DIT e LCOM; possivelmente são baixas.

RQ 04 - Qual a relação entre o tamanho dos repositórios e as suas características de qualidade?

Hipótese: Repositórios com maior tamanho podem apresentar uma boa qualidade, pois códigos extensos que não mantenham boa qualidade (métricas de qualidade baixas) são difíceis de se manter e compreender.

2. Metodologia

2.1. Definição do Problema

Análise de métricas dos códigos fonte dos repositórios dos 1.000 repositórios Java mais populares do GitHub para compreender os padrões de qualidade, atividade e maturidade desses projetos.

2.1. Formulação de Hipóteses

Definir hipóteses informais subjacentes às Questões de Pesquisas, pré-definindo o modelo interpretativo do levantamento de dados.

2.2. Coleta de Dados

Utilizando a API do GitHub, os dados de cada repositório que atende aos critérios (número mínimo de estrelas e arquivos Java) são coletados. Através do GraphQL são extraídas, para cada repositório, as informações: nome, proprietário, popularidade (número de estrelas), atividade (número de releases) e idade. Posteriormente os repositórios são clonados localmente para serem analisados.

2.3. Processamento e Análise de Dados

Para cada repositório, são executadas métricas do código fonte usando a ferramenta CK (Chidamber & Kemerer). As métricas incluem: Acoplamento entre objetos de classe (CBO - *Coupling between object classes*), Extensão da árvore de herança (DIT - *Depth of Inheritance Tree*), Falta de coesão em métodos (LCOM - *Lack of Cohesion in Methods*) e Linhas de código (LOC - *Lines of Code*). As métricas do conjunto de classes de cada repositório são somadas e armazenadas para posterior análise.

Os dados obtidos dos repositórios e métricas dos código fontes são submetidos a análises estatísticas gerando informações como médias, medianas e modas; além de gráficos comparativos.

2.4. Interpretação dos Resultados

As informações geradas pela Análise de Dados são interpretadas a partir das hipóteses formuladas e dos tratamentos estatísticos. A interpretação pode levar a conclusões sobre a relação entre popularidade, atividade, idade dos projetos e a qualidade de seu código-fonte.

3. Resultados

Inserir aqui os resultados

4. Discussão

Discutir os resultados

5. Conclusões

Conclusão do trabalho