

Caracterizando a Atividade de Code Review no GitHub

Hugo Poletto Alacoque Gomes¹, Matheus Nolasco¹,
Maria Aryene Costa¹, Lucas Santos Rosa¹

¹Instituto de Ciências Exatas e Informática –
Pontifícia Universidade de Minas Gerais (PUC Minas)
Belo Horizonte – MG – Brasil

Resumo. *O estudo investigou a relação entre métricas de revisão de código em repositórios do GitHub. No entanto, não foi encontrada correlação significativa entre tamanho dos Pull Requests, tempo de análise, descrição e interações nos PRs, e o feedback final das revisões, assim como o número de revisões. Isso destaca a complexidade do processo de revisão de código e a necessidade de considerar outros fatores para sua melhoria. Pesquisas adicionais são necessárias para entender melhor esses aspectos e desenvolver práticas mais eficazes de revisão de código.*

1. Introdução

O Code Review é uma prática constante não só nos repositórios do GitHub, como também em outras plataformas de repositórios de Git. Ela consiste na interação dos desenvolvedores com os revisores, visando inspecionar o código antes de integrá-lo na branch principal. No contexto de repositórios abertos do GitHub, a prática do Code Review é feita por meio de Pull Requests (PR).

Neste projeto, pretende-se analisar a atividade de code review desenvolvida em repositórios populares do GitHub, identificando variáveis que influenciam no merge de um Pull Request, sob a perspectiva de desenvolvedores que submetem código aos repositórios selecionados.

Para isso, será utilizado uma matriz de correlação destas variáveis para identificar a relação entre as métricas e, assim, responder as seguintes perguntas de pesquisa:

A. Feedback Final das Revisões (Status do PR):

1. RQ: Qual a relação entre o tamanho dos PR's e o feedback final das revisões?
2. RQ: Qual a relação entre o tempo de análise dos PR's e o feedback final das revisões?
3. RQ: Qual a relação entre a descrição dos PR's e o feedback final das revisões?
4. RQ: Qual a relação entre as interações nos PR's e o feedback final das revisões?

B. Número de Revisões:

1. RQ: Qual a relação entre o tamanho dos PR's e o número de revisões realizadas?
2. RQ: Qual a relação entre o tempo de análise dos PR's e o número de revisões realizadas?
3. RQ: Qual a relação entre a descrição dos PR's e o número de revisões realizadas?
4. RQ: Qual a relação entre as interações nos PR's e o número de revisões realizadas?

1.1. Hipóteses Iniciais

A partir de cada uma dessas perguntas, hipóteses iniciais (HI) foram formuladas. Essas “hipóteses informais” são, respectivamente:

1. HI-RQ: PR's maiores, recebem feedbacks maiores, uma vez que para haver uma observação, é necessário avaliar uma quantidade maior de mudanças.
2. HI-RQ: PR's que tem um tempo de análise menor, geralmente são aceitos, ou seja, merged ou closed de forma positiva.
3. HI-RQ: PR's com descrições mais detalhadas geralmente são aceitos, ou seja, merged ou closed de forma positiva.
4. HI-RQ: PR's com mais interações tendem a alcançar uma resolução positiva, porque recebem mais colaboração.
5. HI-RQ: O tamanho dos PR's é proporcional ao número de revisões realizadas.
6. HI-RQ: PR's que demoram mais para serem analisados geralmente têm um maior número de revisões em comparação com PR's analisados rapidamente.
7. HI-RQ: PR's que tem descrições mais claras, tem um maior número de revisões.
8. HI-RQ: PR's com um número maior de interações (comentários, discussões) geralmente têm um maior número de revisões em comparação com PR's com menos interações.

2. Metodologia

2.1. Criação do Dataset

O dataset utilizado neste laboratório será composto por PR's de 200 repositórios que tenham passado pelo processo de Code Review nos repositórios mais populares do GitHub que possuam pelo menos 100 PR's mesclados e fechados (Merged + Closed), no mínimo 1 PR com revisão nos últimos 1000 PR's e que levou pelo menos uma hora de revisão para remover PR's revisadas de forma automática por meio de bots ou ferramentas de CI/CD.

2.2. Definição de Métricas

Para cada dimensão, as correlações com as métricas serão feitas da seguinte forma:

- **Tamanho:** número de arquivos; total de linhas adicionadas e removidas.
- **Tempo de Análise:** intervalo entre a criação do PR e a última atividade (fechamento ou mesclagem).
- **Descrição:** número de caracteres do corpo de descrição do PR.
- **Interações:** número de participantes; número de comentários.

2.3. Processo de Desenvolvimento

- **Sprint 01:** Lista de repositórios selecionados + Criação do script de coleta dos PR's e das métricas definidas.
- **Sprint 02:** Dataset completo, com os valores de todas as métricas necessárias + Primeira versão do relatório final, com as hipóteses iniciais.
- **Sprint 03:** Análise e visualização de dados + Elaboração do relatório final.

3. Resultados

Para responder cada uma das questões, podemos utilizar uma matriz de correlação de Pearson na Figura 1. Este tipo de gráfico é útil para realizar a correlação entre as métricas de Pull Request, já que retorna o nível de correlação de -1 sendo uma correlação inversamente forte, 0 sendo uma correlação inexistente, e 1 sendo uma correlação forte.

Sendo assim, utilizando a matriz de correlação como base, a resposta para as perguntas de pesquisa é, respectivamente:

1. RQ: Comparando o *número médio de PR's mesclados* com o *número médio de número de arquivos, linhas adicionadas e linhas deletadas*, é possível observar que não há correlação entre elas, já que os valores são menores que 0,5 e maiores que -0,5.
2. RQ: Comparando o *número médio de PR's mesclados* com o *número médio de análise dos PR's*, é possível observar que não há correlação entre elas, já que o valor é menor que 0,5 e maior que -0,5.
3. RQ: Comparando o *número médio de PR's mesclados* com o *número médio de caracteres no corpo de descrição dos PR's*, é possível observar que não há correlação entre elas, já que o valor é menor que 0,5 e maior que -0,5.
4. RQ: Comparando o *número médio de PR's mesclados* com o *número médio de participantes e comentários*, é possível observar que não há correlação entre elas, já que os valores são menores que 0,5 e maiores que -0,5.
5. RQ: Comparando o *número médio de revisões* com o *número médio de arquivos, linhas adicionadas e linhas deletadas*, é possível observar que não há correlação entre elas, já que os valores são menores que 0,5 e maiores que -0,5.
6. RQ: Comparando o *número médio de revisões* com o *número médio de análise dos PR's*, é possível observar que não há correlação entre elas, já que o valor é menor que 0,5 e maior que -0,5.
7. RQ: Comparando o *número médio de revisões* com o *número médio de caracteres no corpo de descrição dos PR's*, é possível observar que não há correlação entre elas, já que o valor é menor que 0,5 e maior que -0,5.
8. RQ: Comparando o *número médio de revisões* com o *número médio de participantes e comentários*, é possível observar que não há correlação entre elas, já que os valores são menores que 0,5 e maiores que -0,5, apesar do índice entre revisões e comentários ser 0,38.

4. Discussão

Na seção 3 foi mostrado que não houve correlação entre as métricas analisadas, logo as hipóteses apresentadas na seção 1.1 não se tornaram verdadeiras após comparação com os resultados obtidos.

5. Conclusão

A análise realizada neste estudo buscou investigar a relação entre diversas métricas relacionadas ao processo de revisão de código em repositórios populares do GitHub. No entanto, os resultados obtidos mostraram que não houve correlação significativa entre as variáveis estudadas, indicando que as hipóteses iniciais formuladas não foram confirmadas.

Em outras palavras, não foi possível estabelecer relações diretas entre o tamanho dos Pull Requests, tempo de análise, descrição dos PRs, interações nos PRs e o feedback final das revisões, assim como o número de revisões realizadas. Isso sugere que outros fatores podem estar influenciando esses aspectos do processo de revisão de código, além dos considerados neste estudo.

Portanto, é necessário realizar pesquisas adicionais e mais detalhadas para identificar outros possíveis fatores que influenciam o processo de revisão de código e entender melhor como eles interagem entre si. Essas descobertas podem ajudar a desenvolver práticas mais eficazes e eficientes de revisão de código, contribuindo assim para a melhoria da qualidade do software desenvolvido em colaboração.

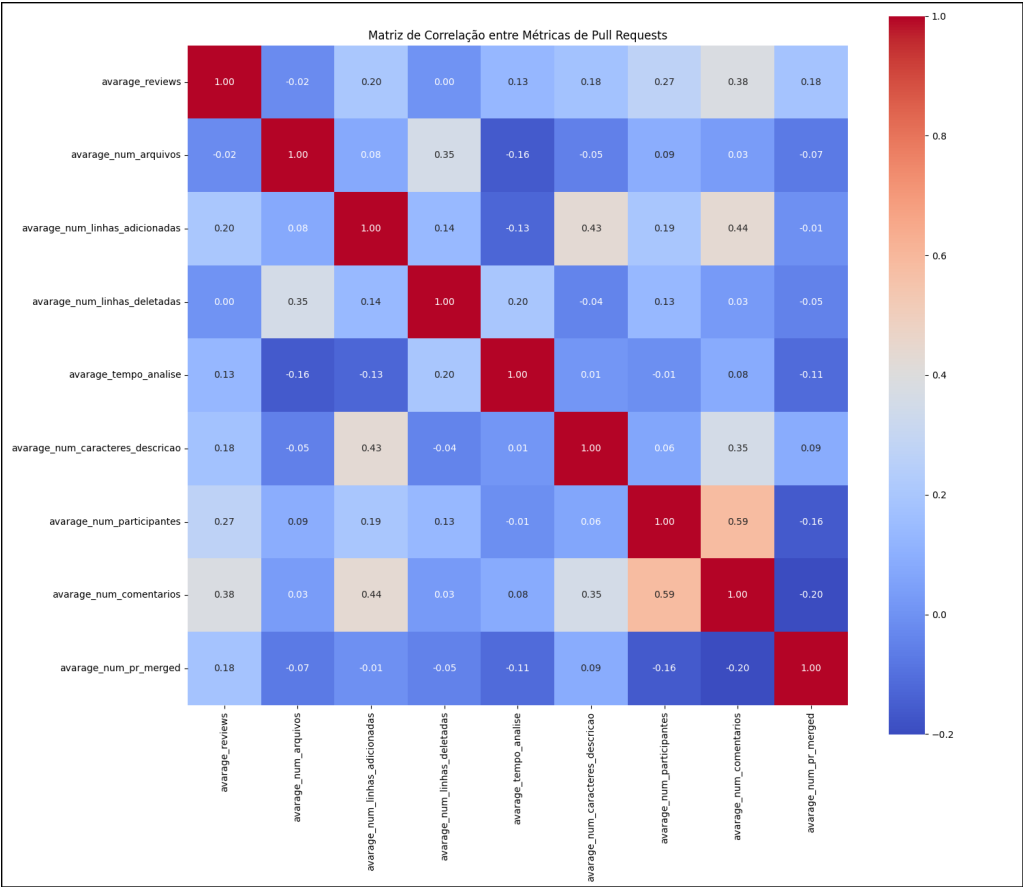


Figure 1. Esta figura é um matriz de correlação entre as métricas .

References