

Análise GQM “Programadores mais ativos no repositório escrevem códigos com mais comentários?”

Douglas Willian e João Teles

FASE 1

1.1 Elaborar o GQM

No primeiro momento do desenvolvimento do trabalho elaboramos as três partes do GQM, Objetivos, Questão e Métricas. Foi avaliado pelo professor Lesandro Ponciano a qualidade da proposta, como resultado final após revisão obteve-se o seguinte GQM:

Objetivo: O objetivo deste estudo é entender se programadores com mais atividade, seguidores, escrevem código com menos comentários.

Questão: A questão principal a ser respondida: existe uma relação de qualidade entre número de comentários no código e a atividade dos programadores no repositório?

Métricas: 1) número de comentarios (modelo): número de código comentados; 2) Atividade dos programadores (modelo): número de commits feitos no repositório,

número de linhas adicionadas ao repositório, quantidade de forks, quantidade de following.

FASE 2

2.1 Coleta e Análise

Após a definição do objetivo e das métricas, fizemos uma análise de quais variáveis selecionar e como coletar. Utilizando ferramentas como Octokit, uma biblioteca que permite a coleta de informações como número de forks do usuário, número de stars no perfil do usuário, followers, repositórios que participa entre várias outras funcionalidades para coleta de dados da plataforma GitHub.

As variáveis foram escolhidas a partir das métricas, cada uma tem o peso relacionado com o objetivo do GQM.

Começando pela métrica de stars e followers, feito uma análise de referência ao artigo “DevRank: Mining Influential Developers In Github”, Percebemos que as métricas fazem sentido na medição de influência dos desenvolvedores. “DevRank, that using not only “follow” but also “commit” to measure the developers’ influence”. Coletamos as estrelas, seguidores, commits e forks utilizando a ferramenta Octokit e a própria API da plataforma GitHub.

No primeiro momento percebemos que a atividade dos programadores e a influência dos mesmos no repositório era fácil de coletar, a coleta de comentários por programador seria uma variável mais complexa de coletar os dados e associar aos programadores.

Feito a coleta dos comentários utilizando a API do GITHUB, temos os dados coletados para cada métrica, a tabela 1 representa o final da medição feita para cada uma das métricas selecionadas.

Programador	Linhas	Linhas Comentadas	star	forks	commits	%
wongsyrone	1427	254	411	166	150	18%
Noisyfox	1001	86	326	146	110	9%
GangZhuo	91	2	3351	764	98	2%
celeron533	952	55	48	7	84	6%
icylogic	1082	19	135	594	75	2%
breakwa11	308	3	2406	966	16	1%
wzxjohn	1932	37	331	129	14	2%
chenshaoju	42	0	3	0	15	0%
everyx	74	1	69	17	11	1%
rwasef1830	576	9	2	2	12	2%

Tabela 1: medição dos desenvolvedores

Inferimos a hipótese de que mais linhas de código alteradas e commits, logo tem mais atividade no repositório. Quanto mais influência (stars, forks) esperávamos desenvolvedores que deixassem menos comentários no código. Consideramos os dados parcialmente conclusivos, visto que da amostra de dados temos desenvolvedores com mais de mil linhas de código alteradas no repositório, seguidores e quantidade de commits maior que a média e ainda assim um percentual maior de código comentado.

O Código utilizado para análise pode ser encontrado como link nas referências.

FASE 3

3.1 CONCLUSÃO

Uma grande quantidade de código comentado pode mostrar que o código não está intuitivo e não segue as melhores práticas, um código bem escrito não precisa de muito comentários ou só alguns comentários pontuais para um melhor entendimento de funções, por exemplo.

Feita nossa análise de atividade por código comentado, percebemos que usuários com mais atividade e influência (medidas pelas métricas de fork e estrelas) não necessariamente fazem um código sem comentários ou com menos comentários.

Referência Bibliográfica

Liao, Zhifang & Jin, Haozhi & Li, Yifan & Zhao, Benhong & Wu, Jinsong & Liu, Shengzong. (2017). DevRank: Mining Influential Developers In Github.

Repositório do estudo

<https://github.com/douglaswcastro/PQSanaliseDocumentacao>