Pontifícia Universidade Católica de Minas Gerais

Disciplina: Processos e Qualidade de Software

Alunos: Douglas Willian / João Pedro

Tema: Documentação de código e Programadores

Link repositório:
https://github.com/douglaswcastro/PQSAnaliseDocumentacao

1 PRIMEIRA PARTE

1.1 DIMENSÕES DE ANÁLISE

1.1.1 - 10 commits de cada programador.

Pegamos 10 commits para cada programador do repositório, a escolha dos 10 commits foi para estabelecer um padrão para a verificação, com isso coletamos as linhas adicionadas nos commits contando o total de linhas adicionadas e as linhas comentadas.

1.1.2 – Repositório

O repositório escolhido foi: https://github.com/shadowsocks/shadowsocks-windows, a escolha se baseou no link que estava no documento de requisito do trabalho, selecionamos por mês, e escolhemos a linguagem C#, e o selecionamos

1.1.3 - Documentação.

Um código bem comentado, ajuda a outros desenvolvedores a entender melhor aquele trecho de código ou alguma função especifica.

1.1.4 – Escolha da Linguagem.

Escolhemos a linguagem C# pela facilidade que seria para a verificação, pois é uma linguagem que dominamos, e para ter um conhecimento melhor sobre documentação

1.1.5 – Quantidade de linhas commits.

No repositório escolhido pegamos os maiores contribuidores dele e pegamos 10 commits dele fazendo requisições via api do github, para ter os commit de cada programador, depois de pegar os commits, fizemos uma verificação manualmente para cada commit com as linhas de códigos adicionadas e as linhas comentadas, por esse motivo de ter pego só os commit de determinado programador algumas commits podem ter baixa quantidade de alterações com isso não estabelemos uma métrica de quantidade de linhas por programador, mas sim a métrica de commit dele.

1.1.6 - Quantidade de comentários.

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

Mas também um código bem escrito pode não ter comentários nenhum, já que ele possui nome de métodos e variáveis bem intuitivos, com isso quase que não é preciso de comentário ou não precisa de nenhum.

1.1.7 - Complexidade

Quando se tem uma maior complexidade nos métodos e aconselhável que tenha pelo menos algum comentário para explicar o que um método ou função faz, para que outra pessoa que pegue o código não fique perdida tentando entender

1.2 MÉTODOS APLICADOS NA PESQUISA

Fizemos a verificação usando api do github para verificar os commits dos programadores selecionados que foram os 10 que mais tiveram atividade no projeto, depois dessa verificação fizemos um levantamento com os commits de

cada programador que pegamos usando a api, para tornar mais fácil e ágil a nossa analise fizemos a contagem de linhas de alterações e linhas de comentários manualmente.

1.3 REPOSITÓRIO ESTUDADO

O repositório estudado foi o https://github.com/shadowsocks/shadowsocks-windows, que foi selecionado com base trending do mês, ele foi o primeiro da lista, escolhemos o top 1 do mês na linguagem C#, que foi a escolhida por nós. Como mostra na imagem abaixo como selecionamos o repositório.