Pesquisador. Muito obrigado por ter aceito participar da entrevista. Ela tem como objetivo compreender os processos de criação, manutenção e verificação da qualidade do código de teste de unidade. Não há respostas certas ou erradas, desejáveis ​​ou indesejáveis. Dessa forma, senta-se à vontade para dizer o que realmente pensa. Tudo bem?

Entrevistado. Tudo certo. Tudo bem.

P. Reforçando um item do termo de consentimento, nossa conversa está sendo gravada. O objetivo disso é para que eu possa obter todos os detalhes, mas ao mesmo tempo ser capaz de manter uma conversa atenta com você. Garanto que todos os seus comentários permanecerão confidenciais e que no relatório final conterá os comentários dos participantes sem nenhuma referência aos indivíduos. Ok?

E. Ok

P. Como você começou a trabalhar com teste de software?

E. Eu comecei durante o trabalho depois da faculdade e eu comecei no projeto que estou trabalhando agora. Uma das diretrizes da empresa desse ano era justamente aumentar a qualidade de software, então uma das ações foi produzir testes unitários para os componentes que a gente desenvolve.

P. Quais foram suas fontes de aprendizado sobre código de teste?

E. Eu estou utilizando a internet mesmo, alguns tutoriais, alguns sites como o Medium por exemplo que tem alguns fóruns, trazem artigos sobre isso e a documentação. Tenho lido a documentação do Jest e é um framework para desenvolver testes e as documentações que a gente encontra como as documentações das tecnologias que a gente usa na empresa como BuildJS, coisas assim.

P. Em quais linguagens você cria testes?

E. No projeto que eu estava trabalhando a gente estava testando componentes no frontend então estava trabalhando com Javascript e para o Java no backend.

P. São as duas linguagens que você trabalha atualmente?

E. Isso.

P. Como é seu processo de criação de teste de unidade?

E. Primeiro eu tento decidir um escopo para o qual eu vou utilizar o teste, então tento levantar quais os casos positivos e negativos daquele componente que estou testando. Feito isso, depois de decidir quais são os tipos de assertos eu começo a implementar os testes para verificar esses escopos. Tento trabalhar nessa linha.

P. Existe algum fluxograma, template documentado que padronize esse processo?

E. Não temos. Pelo menos aqui na empresa não temos isso.

P. Quais ferramentas são utilizadas para criar/ executar teste? Você comentou do Jenkins, mas tem alguma outra ou algumas outras?

E. A gente usa o Jest, o framework para executar tudo. Temos também o Sonar, a gente está implementando o Sonar para fazer a leitura desses... gerar os relatórios de testes e tal.

P. Como é o processo de verificação de qualidade do código de teste criado, por exemplo: verificação de escrita, alguém revisa o teste depois de codificado, compatibilidade com o requisito, assim por diante?

E. Ele passa por um processo de code review assim como as demais funcionalidades dos programas. Essa parte ainda está em desenvolvimento aqui na empresa então é uma coisa bem nova. No mais ele está passando mesmo pelo processo de code review padrão.

P. Você pode falar um pouco mais sobre esse processo de code review padrão?

E. Quando a gente faz uma funcionalidade nova ou desenvolve alguma coisa, ele sobe o código para o repositório do git e assim que a gente sobe para o repositório ele abre um merge request dessa funcionalidade e algum desenvolvedor do time ele precisa aprovar aquilo, Então outro desenvolvedor que não teve contato com a atividade, ele vai fazer o code review, ele vai olhar o código ver se o código segue os padrões de escrita que a gente usa aqui, os padrões de Javascript ou de java, se a lógica está coesa, se não tem nenhum erro ou se todo o escopo foi abordado. No caso de teste ele verifica se seguiu a sintaxe correta, se os casos de teste estão abrangendo a maioria das possibilidades, esse tipo de análise.

P. Quantos desenvolvedores fazem essa revisão de um código submetido?

E. Depende do time. Varia muito de time para time aqui, mas em geral é necessário apenas um único desenvolvedor. Por exemplo se eu subir uma atividade, outro desenvolvedor precisa olhar ela.

P. Passando para o processo de manutenção de código de teste, como esse processo é realizado?

E. Depende muito de time para time aqui na empresa. No time que eu estava trabalhando, se mudasse algum componente do código, por exemplo e ele tivesse impacto nos testes aí sim, a gente mudaria os testes. Rodaria os testes e veria o que está quebrado devido aquela modificação e assim faria uma manutenção disso, mas fazer uma manutenção contínua ainda não é previsto aqui. Justamente por estar no início o projeto.

P. O que você conhece sobre test smell?

E. Eu conheço sobre code smell, não sei se seria o mesmo conceito aplicado para testes, mas são levantamentos pontuais "ah, você pode melhorar essa parte do código", "aqui a sintaxe pode ser melhorada", eu já vi isso no Sonar. O Sonar implementa algumas métricas para code smell. Não sei se seria isso aplicado para testes.

P. Test smell ele descende de code smell, então a princípio seria o mesmo conceito. Eu vou ler o conceito para você só para a gente alinhar a informação e caso você tenha alguma dúvida, você pode me fazer e a gente alinha também o conhecimento, porque as perguntas a partir de agora vão estar relacionadas a esse tópico, ok?

E. Ok

P. “Test smells são pontos fracos na estrutura do código de teste, tendo um impacto negativo na sustentabilidade da atividade de teste, pois os tornam complexos, difíceis de entender e modificar, prejudicando a repetibilidade, independência e estabilidade dos testes.”

E. Ok.

P. Então tem a ver com o que você comentou, tem a ver com code smell, mas tem tipos de test smells específicos que diferem de code smells.

E. Essa parte eu não cheguei a trabalhar.

P. Pensando nessa questão conceitual, que é o que é preciso para as próximas perguntas, você tem alguma dúvida, quer perguntar alguma coisa?

E. Não, eu acho que consegui entender pelo que você me explicou.

P. Vou passar para a próxima pergunta. Caso você tenha qualquer dúvida, qualquer coisa você me pergunta e a gente alinha. Está bom?

E. Está bem.

P. Como test smells são tratados no processo de criação do código de teste?

E. Aqui a gente ainda não tem esse nível de maturidade nos testes então fica a cargo do desenvolvedor analisar o código e também no code review, a pessoa que for fazer o code review daquele código de teste ela verificar isso. Até a possibilidade das ferramentas como o Sonar realizarem isso nos códigos de teste.

P. Esse tipo de tratamento, apesar de ter comentado que não tem tanta necessidade de manutenção no momento, mas em um determinado momento, vocês precisem editar algum código de teste e se deparem com algum tipo de smell, é feita alguma alteração nesse momento ou o foco no momento da edição do teste acaba sendo apenas a correção para atender requisito, cobertura?

E. Se você se deparar com uma situação dessa, você pode ser corrigida sim. Mesmo que o foco que você está mexendo naquele código é para atender alguma funcionalidade, ou mudar o escopo do teste alguma coisa assim, a qualidade também é vista como algo importante então você tem liberdade para fazer essa melhoria.

P. Para você, como seria possível prevenir ou evitar a inserção de test smell no código de teste no momento que esse teste é criado?

E. Eu acho que um dos principais padrões seria treinamento. Treinar as pessoas para ter um olhar crítico para esse sentido para que na hora que ela fosse desenvolver ela tomasse cuidado, cuidados necessários para evitar o máximo possível esse tipo de inserção no código.

P. Estou chegando ao final da entrevista. Você tem alguma pergunta para mim, quer acrescentar alguma informação ou ainda tem alguma sugestão para melhorar a entrevista?

E. É muito bacana a pesquisa, espero que dê tudo certo. Nessa área de teste eu acho que não tinha visto mesmo nenhuma pesquisa sobre isso. Umas coisas que eu tive dificuldade de aprender talvez, não sei se vai ser necessário para sua pesquisa foi pesquisas em front end para definir o escopo disso. É ainda bem escuro, não sei. Não vi muita coisa a respeito disso para definir os escopos de teste de componente unitários. É algo que eu tive dificuldade de aprender, ainda tenho. Saber no backend é mais fácil. Vai voltar 200, ou vai voltar 400 se eu fizer tal operação, agora no front end é uma coisa mais ampla e fica mais difícil de você concentrar o escopo do teste para que você possa definir isso.

P. Entendi. É um ponto interessante. Minha pesquisa ela não tem esse foco, o foco dela é voltado na qualidade do teste unitário, especificamente em se tratando de test smell, porque você tem alguns pontos que a acabam interferindo na qualidade do teste unitário mas no meu caso o foco é test smell. Então não entra tanto nessas questões de dificuldade para front end ou back end, mas pensando nesse seu relato talvez de acordo com os relatos dos entrevistados, a gente ainda vai fazer essa análise, possa ser que a gente encontre algum ponto por exemplo de oportunidade de melhoria.

E. Até pelo fato da qualidade como você comentou. Eu não sei a fundo essa teoria assim, mas talvez por você elicitar um bom escopo, pode aumentar o nível da qualidade daquele código.

P. Com certeza. Muito obrigado pelas informações que você me passou. Com certeza elas vão ajudar bastante. A gente está no final dessa fase de levantamento de dados, então a gente vai passar para análise, para consolidação desses dados e com certeza essas informações que você me passou vão ajudar bastante na entrevista.

E. Que bom! Boa sorte com a sua pesquisa.

P. Ok, eu vou parar a gravação. Ela fica disponível tanto para mim quanto para você no skype, mas eu preciso salvar porque eu preciso fazer a transcrição, tirar qualquer referência pessoal que a gente tenha acabado comentando durante a entrevista.