Pesquisador. Muito obrigado por participar da entrevista. Ela tem como objetivo compreender os processos de criação e de manutenção de código de testes e verificação da qualidade do código de teste de unidade. Não há respostas certas ou erradas ou desejáveis ​​ou indesejáveis. Dessa forma, senta-se à vontade para dizer o que realmente pensa. 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. O 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. A minha primeira experiência foi no local que estou trabalhando atualmente, porque a cultura da empresa está voltando muito agora para implementação de testes unitários, todas as aplicações novas que estão sendo desenvolvidas tem a meta de cobertura total. E desde que entrei lá todas as minhas contribuições foram testadas unitariamente por mim mesmo

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

E. foi compartilhamento de conhecimento na empresa, porque esse cargo de estar testando e implementando boas métricas de qualidade é o nosso tech leader, gestor do time estabeleceu como meta e guiava a gente e muito em comunidade, porque antes dessa situação que a gente está, eu participava de comunidade de desenvolvedor aqui no estado e trocava ideia de qual eram as melhores ferramentas, melhores metodologias e como aplicar isso da melhor forma.

P. Em quais linguagens você cria testes?

E. Javascript e typescript.

P. Qual são as linguagens do seu projeto atual?

E. Sim. Em todos os projetos a gente usa Javascript, mas eu já fiz alguns ajustes na linguagem Go.

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

E. Isso é uma coisa que está até mudando agora, porque quando eu entrei lá, eu tinha sempre o hábito de fazer o código pensando no comportamento desejado e depois criava o teste para ela, mas recentemente eu tenho adotado a metodologia inversa, que é o TDD, que primeiro eu escrevo o teste para ver ele falhando e eu vou escrevendo o código para atender o teste e as outras especificações. E eu já estou nisso desde o início desse ano e tem funcionado muito bem.

P. Existe algum fluxograma, template... que padronize esse processo de criação de teste?

E. Não, só tem as ferramentas que são estabelecidas para isso. É a única medida que a gente tem. E define como a gente escreve teste. Inclusive a gente trocou de ferramenta em função disso porque ela influenciava o desenvolvimento depois da aplicação pronta, todos os detalhes da implementação foi definido, mas também mudamos para uma outra ferramenta, chamada test libraries, que permite a gente testar os detalhes de requisitos, de usabilidade e não de implementação. Esse é o único padronização que nós temos em teste não tem fluxograma, não existe um processo bem definido disso.

P. Quais ferramentas são utilizadas para criar/ executar teste?

E. respondido anteriormente

P. Como é o processo de verificação de qualidade do código de teste criado, por exemplo: verificação da escrita, da estrutura desse código, alguém faz a revisão desse código escrito por um programador? Como é feita essa verificação da qualidade?

E. É feita por meus colegas de trabalho, a gente trabalha com o code review, que tanto o código que a gente escreve quanto o teste que a gente escreve e revisado por programadores para eles entenderem e também opinarem questões de melhoria e possíveis casos que a cobertura de testes não está adequada ainda.

P. Quantos desenvolvedores fazem esse code review? Você submete o código, essa revisão é feita por quantos programadores?

E. o mínimo para a gente subir, dando um merge no projeto são 2, mas geralmente são 3 ou 4 que fazem esse review.

P. Para aumentar esse número de revisores, existe uma pré-condição? Como isso ocorre?

E. Ocorre dependendo da complexidade, ou então da prioridade.

P. Como ocorre o processo de manutenção do código de testes?

E. Não tem bem um processo. A manutenção de testes, ele evolui juntamente com o código fonte que ele testa. A gente trabalha com círculo de CI, sempre que a gente faz algum... ele sempre roda os testes que foram escritos. E se alguma coisa falha ele não sobre tem que corrigir o teste, aquele comportamento que quebrou.

P. O que você conhece sobre test smell?

E. Não conheço sobre test smell.

P. Eu vou te passar o conceito vou te mandar dois exemplos, os exemplos que eu tenho são em Java, mas qualquer dúvida que você tiver, você pode me perguntar, eu passo para você, mas o foco em relação a test smell é a estrutura do teste e não a linguagem. O conceito de test smells, eles 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.” O primeiro é o chamado Conditional Test Logic smell. Ele é considerado um smell porque ele tem uma estrutura de condição, de repetição, o que uma vez que esse teste falhando dificulta a definição de em que momento das condicionais ou de repetição que esse teste falhou. O outo exemplo é chamado Empty test smell. Ele é considerado um smell porque não tem nenhum script executável então não tem informação de porque esse teste está comentado, não está em execução ou não foi excluído, então era algo que nao deveria existir e está impactando na execução dos testes. Existem outros tipos de smells relacionados a parâmetros, a assertes, a dependência entre testes e todas elas estão relacionadas justamente ao impacto na repetibilidade, independência e estabilidade dos testes. OK

E. Ok, compreendido

P. Alguma dúvida?

E. Não só não conhecia o termo específico, mas esse sentimento esse problema com o teste de fato ocorre e eu entendo.

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

E. Eles são geralmente filtrados no processo de code review porque como eu falei tanto o código quanto os testes são lidos e compreendidos por outros programadores, tanto para facilitar a manutenção quanto para certificar que aquilo está realmente testando o que deveria. E também como eu mencionei anteriormente, a troca de ferramenta é justamente por isso, porque antes a gente usava uma ferramenta que acabava gerando muito test smells porque ela manipulava demais as propriedades que eram passadas e era uma coisa muito sintética e apesar de nós termos uma cobertura imensa não refletia em um bem aplicado e sempre aparecia diversos bugs, seguindo em frente a gente busca olhar mais como o item estão interagindo de fato e deixar o teste bem específico.

P. Esse processo que você falou de code review para tratamento de test smells ele ocorre também para manutenção de testes?

E. Sim porque tudo que a gente faz a gente faz rodando em integração contínua. Se você adiciona uma nova feature, você tem que corrigir algum teste e também com essa questão de testes, quando um bug é encontrado. Primeiro a gente escreve um teste para corrigir esse bug e depois a gente escreve o teste

P. Para você, como seria possível prevenir/evitar a inserção de test smell no código de teste no momento da criação desse código?

E. A maior parte é um review detalhado dos testes, porque quando acontece de vir algum teste muitas vezes foi o review que não foi muito atencioso e é uma coisa que acaba sendo paga mais para frente como débito técnico e também a escolha de ferramentas é fundamental, porque como eu mencionei, algumas ferramentas parecem que elas propiciam muito a implementação de test smells.

P. Você tem alguma pergunta para mim, quer acrescentar alguma informação ou tem alguma sugestão para melhorar nossa entrevista?

E. eu queria um pouco, se puder me falar de sua pesquisa.

P. Agora que estamos no final posso falar um pouco mais. Minha pesquisa é focada em test smell. Então a ideia é conhecer esses processos de criação e manutenção de um código de teste de unidade dentro da indústria e identificar se em algum momento esses test semlls são tratados nessa etapa e identificar um ponto de melhoria para focar. Nessa etapa estou mapeando esses processos na indústria, sabe de conde conhecem esse assunto, especificamente pelo termo ou caso não conheça pelo termo trata de alguma forma como uso de boas práticas para a redução de inserção de smells e mais para frente a gente vai analisar esses dados e identificar alguma oportunidade de implementação para melhorar esses processos dentro da indústria.

E. É bem interessante. Eu digo do ponto de vista do profissional que trabalha com o front end, ia verificar que essa questão de testes é bem recente. Que o pessoal de Java e o pessoal de backend, da empresa onde trabalho, tem uma experiência maior, mas o front end começou uma cultura, começou a ser implementada recentemente.

P. muito obrigado pelas suas informações elas vão me ajudar bastante na pesquisa.