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 ou desejáveis ​​ou indesejáveis. Dessa forma, senta-se à vontade para dizer o que realmente pensa. Ok?

Entrevistado. Certo. Perfeito

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. Tudo bem.

E. Tudo bem, perfeito, sem problema.

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

E. Depois de algumas trocas de empresa, na minha terceira empresa que eu trabalhei, isso começou a ser relevante porque era uma empresa maior, ela se preocupava um pouco mais com isso.. e eu aprendi um pouco disso em eventos, sabia muito a teoria, comecei a praticar nessa empresa que tinha um porte um pouco maior e foi basicamente isso. Era um projeto basicamente legado que a gente começou a adicionar teste de unidade na época aos poucos em cada parte que a gente mexia no código.

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

E. Eu já li clean code, um livro bem conhecido do kent back, uncle bob na verdade. Fui em vários eventos do TDC, The Develops Conference que acontece em alguns lugares do país, já fui em Florianópolis, em porto alegre, em minas gerais. E vi alguns vídeos, sigo algumas pessoas no twitter onde eu consigo me informar, vou em alguns eventos locais onde a gente fala sobre qualidade de testes, já compartilhei conhecimento em alguns mas recentemente, e gosto bastante de ir em eventos em geral.

P. Em quais linguagens você cria testes?

E. A primeira que eu comecei a trabalhar foi em C#, depois Java, Javascript e dentro de javascript algumas variações como React, NodeJs, AngularJS, usando ferramentas diferentes para cada uma delas.

P. Qual plataforma do seu projeto atual?

E. É um front end onde a gente utiliza principalmente Typescript com React

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

E. Isso depende muito, varia muito de acordo com cada situação, cada problema. eu gosto muito de TDD, test driven development, é uma coisa que eu tenho praticado bastante, inclusive a empresa que eu trabalho hoje ela prega muito por isso, por uma cultura de TDD, de testar antes de efetivamente implementar o código, então, eu geralmente começo pensando nos testes que eu quero fazer antes de criar qualquer pedaço de código que seja, isso eu consigo escrever por inteiro eu tento deixar mais ou menos um rascunho dele e conforme eu for escrevendo os rascunhos deles e os testes em si, eu começo a rodar, começa a falhar e eu vou implementando o código até que os meus testes passem.

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

E. Eu já usei uma época um conceito de BDD que é uma coisa um pouco diferente do TDD, o teste de unidade em si, mas é basicamente para usar um pouco de contexto de onde é que a gente está, o que a gente faz e o que a gente espera, e um padrão de Given When Then, o Given é dado que eu tenha alguma situação, o when é o quando eu fizer alguma ação, fizer uma chamada de API que eu fizer algum clique de botão, alguma coisa nesse sentido, e o then é o então eu vou fazer uma asserção, um assert verificar se alguma coisa que eu esperava aconteceu dado o contexto que eu estava e a ação que eu fiz. Não é um template muito técnico, é algo mais conceitual eu diria.

P. Isso é um padrão dentro da equipe ou você tem a cultura de realizar?

E. Isso é mais meu mesmo. Em outras equipes que eu trabalhei foi um acordo de time fazer testes assim, inclusive deixar alguns comentários descrevendo algumas situações, cada cenário, o que era o give, o que é o when e o que é o then, mas não. Hoje não é uma coisa da equipe é uma coisa mais que eu trouxe de experiências anteriores.

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

E. Hoje eu trabalho mais com front end, então as ferramentas que eu utilizo são Mocha, Indesign, ReacTest library, Testcafe, Testcafe não é para teste de unidade, é para testes de jornada. Jest, o principal eu diria que é o Jest.

P. Como é o processo de verificação de qualidade do código de teste criado, por exemplo: verificação de escrita, se algum desenvolvedor, gerente revisa o teste depois de criado, verificar compatibilidade com o requisito?

E. A gente não tem o papel de QA no time, o que é comum em outras empresas, a gente não tem esse papel. Nós como time de desenvolvimento somos responsáveis pela codificação, entrega e qualidade do sistema, então todos nós somos responsáveis, o que acontece é que geralmente a gente trabalha no modo de peer programing, duas pessoas desenvolvendo o código ao mesmo tempo. Depois disso a gente tem a parte de pull requests, a parte de efetivamente outras pessoas que não participaram do desenvolvimento podem revisar o que foi escrito naquela história, naquele ticket, naquele card, dado o pull request uma pessoa pelo menos tem que aprovar, uma pessoa que não participou do desenvolvimento, feito isso a gente tem uma etapa que a gente chama de QA, que e uma validação de qualidade, mas quem faz isso é uma outra pessoa ainda, é uma pessoa que não tenha participado da codificação, do desenvolvimento, que não tenha participado do code review, da parte do pull request, ela então é a próxima pessoa que pode participar da parte de QA, onde efetivamente vai ser testado em ambiente um pouco mais isolado, que a gente chama de standing, preview ou alguma coisa nesse sentido que a gente possa testar a aplicação em si e não só olhar o código dela. O code review é bem específico para a gente olhar o código, melhores práticas, se tem teste suficiente, esse tipo de coisa, e a etapa de QA a gente vê efetivamente num site, como se fosse um pré-produção, alguma coisa assim, para efetivamente se a tarefa foi concluída, se dado os critérios de aceitação que ela tem.

P. No code review, quantas pessoas fazem a verificação e um código submetido?

E. Isso varia de card para card, mas o mínimo que a gente espera é que uma pessoa faça isso.

P. Como é o processo de manutenção do código de teste criado?

E. A gente sempre vai atualizando de uma biblioteca um pouco mais antiga para uma que a gente acha que vai nos deixar mais produtiva, no sentido de testes, então saindo mais do Indesing e indo mais para o react testlibrary, a manutenção é basicamente quando a gente for atualizar alguma coisa de código, fazer alguma alteração, a gente com certeza atualiza os testes também, se a gente for implementar uma coisa nova, a gente cria testes novos. É muito comum não, porque isso não acontece, que a gente já tem consciência disso, mas se chegar um pull request no nosso time que não tivesse um teste e tivesse novas implementações esse pull request provavelmente seria negado. A gente teria que devolver esse pull request para quem o criou para que ele fizesse a implementação de testes também.

P. O que você conhece sobre test smell?

E. É tipo um code smell de teste, testes que dão falso positivo, por exemplo? Eles estão passando, vamos dizer assim, estão verdes, mas que mesmo se tu remover alguma parte crítica da ação do teste ele continua verde quando não deveria.

P. Também. Não se limita a isso, mas tem a ver.

E. É a outra parte seria o quão bem o teste está escrito, ou se o teste de unidade está testando mais de uma coisa, enfim, um teste de unidade deve testar uma unidade de código e não testar várias dependências.

P. Como você conheceu este assunto?

E. Eu acho que veio junto com a minha prática profissional, trabalhando mesmo acho que eu ouvia bastante na teoria disso, eu ouvia bastante em eventos também, na faculdade eu não cheguei a ouvir, mas na pós graduação eu ouvi e na prática, no ia a dia de trabalho, aprendendo com colegas, acho que mais aconteceu.

P. por curiosidade, quando eu perguntei sobre test smell, você fez uma analogia com code smell, você já tinha ouvido falar de test smell ou você fez analogia por code smell?

E. Na verdade eu fiz uma analogia, eu nunca tinha ouvido o termo test smell, mas a gente usa a frase "tem um smell nesse teste" vamos dizer assim. Pode ser melhorado, pode ser melhor escrito, pode ser melhor aproveitado esse banco de dados, esse tipo de coisa. Pode colocar isso no before each ao invés de colocar isso em cada um dos testes, repetindo várias vezes.

P. Você tem alguma dúvida sobre esse tema?

E. Eu acho que não, eu acredito que consigo responder o básico se precisar, mas obviamente sempre aprendendo.

P. Eu perguntei pois as perguntas a seguir têm a ver com esse tópico. É um pouco do que a gente já conversou, mas focado agora nesse tópico, em test smell. Tudo bem?

E. Perfeito.

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

E. Acho que isso faz parte do processo de pull request que a gente tem, a gente sempre revisa tanto o código que está sendo implementado da aplicação quanto o código de teste e volta e meia a gente faz sugestões do tipo "a estrutura de teste não está legal, a gente pode resolver ela assim", a gente pode remover o códigos duplicados desse jeito nesse teste, a gente pode testar de outra forma, ao invés de testar simulando um evento de um botão, a gente pode escrever de outra forma, a gente pode ao invés de fazer asserção pelo texto, fazer pela classe, buscar um elemento ao invés de ser por um id, que é um componente específico, fazer por um data test id, enfim a gente faz esse processo no pull request, no peer programing também, eu diria.

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

E. eu acho que é um pouco do mesmo, sempre que a gente tiver uma oportunidade de encontrar um teste nosso a gente tenta fazer, mesmo que não esteja na história que a gente faz, a gente tem bastante autonomia para fazer esse tipo de coisa, vai nos deixar mais produtivo numa próxima história, por exemplo. Mas acho que é isso, quando a gente muda um teste também no processo de pull request a gente vai revisar as alterações e vai olhar se fazem sentido ou se não fazem sentido, outras pessoas e nossas opiniões. Como a gente faz a maioria das coisas pareando, duas pessoas ao mesmo tempo trabalhando juntas, já rola uma discussão ali principalmente.

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

E. Isso vai muito da experiência, do conhecimento sobre o código que a gente está testando. Por exemplo um dev experiente que entra num time novo, não tem tanto conhecimento sobre o code base aquele repositório, a pessoa possivelmente quando for criar testes para esse repositório vai ter alguma dificuldade, não vai saber quais os problemas do projeto e pode publicar coisas que não precisa pode não saber que existe algumas funcionalidades do projeto que já ajudam a testar algumas coisas, acho que esse tipo de coisa pode acontecer. Como evitar, é muito difícil. Acho que peer programing é uma forma, fazer isso pareando é uma forma de ter sempre duas opiniões no momento da criação do teste e fora isso, eu diria fazer treinamento com as pessoas do time, workshops ou em dojos para ir praticando a criação de testes das melhores maneiras possíveis. Eu diria que peer programing é uma boa forma dado que são duas pessoas já, ao invés de só uma olhando para esse código e são duas opiniões diferentes sobre o que pode ser um test smell ou não.

P. Estamos 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. Depois eu queria saber mais da história, a gente pode falar isso depois se quiser, sobre como está sendo as entrevistas, sobre como está sendo a experiência, o que te fez chegar até aqui, o que te fez querer fazer uma pesquisa sobre teste de unidade e um pouco disso, só trocar uma ideia sobre isso. Eu gostei das perguntas, achei interessantes achei o termo test smell, complicado no conceito de que isso não me parece pelo menos ser uma expressão comum de mercado, mas acho que é uma associação com más práticas de testes, acho que e relativamente possível ser feito. Eu acho que é isso.

P. Eu vou falar sobre, porque se você tiver alguma coisa a acrescentar já fica na agravação. Eu tenho 8 anos de experiência como QA, eu trabalhava numa empresa, a gente realizava teste principalmente teste manual, mas já vinha numa movimentação para teste automatizado e um pouco antes de começar essas mudanças lá eu fui desligado, mas eu já estava com o pensamento voltado para o mestrado e eu consegui ingressar no mestrado no mesmo ano. Um mês depois eu já estava dentro do mestrado. Quando eu entrei no mestrado, dentro do grupo de pesquisa do laboratório de lá da UFBA eu tive acesso a esse termo, de test smell, sendo que eu já estava atuando na área relativamente a 8 anos e nunca tinha ouvido falar. Então para mim é bacana esse termo, tem a ver com o que eu já trabalho, minha pesquisa já estava engatinhando na parte de testes, mas quando chegou esse termo me chamou atenção e eu abracei ele para trabalhar. E eu decidi voltar para a indústria, que é o contexto que eu já atuava para investigar até que ponto a indústria conhece, até que ponto esse tema já vem sendo trabalhado, independente de conhecer ou não esse assunto e ver de que forma é possível melhorar esse processo industrialmente, não ficar algo restrito dentro da academia. Minha pesquisa partiu daí. Eu já fiz um estudo, inicialmente voltado para conhecer parte do cenário industrial, essa é a segunda etapa e a ideia é concluindo essa segunda etapa a gente consiga identificar um ponto de melhoria e trabalhar nele numa terceira para concluir a dissertação. Em relação as entrevistas em si, tem a ver com o cenário que você me apresentou, não está muito diferente do que eu tenho entrevistado outras pessoas, eu não posso passar muitos detalhes, mas a conversa que a gente teve não é muito divergente do que eu tenho ouvido de relato dos outros profissionais. Lógico que tem algumas particularidades, que tem a ver com background ou com conhecimento específico daquele profissional, mas o que a gente conversou não está diferente do cenário que eu tenho ouvido das pessoas que eu tenho entrevistado.

E. Massa. O termo realmente eu nunca tinha ouvido especificamente, mas sempre a gente fala sobre más práticas e boas práticas de teste. Você costuma fazer teste em alguma linguagem específica?

P. Eu tenho estudado muito mais em Java, mas como meu estudo é voltado para test smell em geral, possa ser que em algum momento eu identifique alguma oportunidade de abordar em outra linguagem eu vou ter que ir para uma outra linguagem para atacar. Mas eu tenho estudado até o momento em Java.

E. É para te indicar ferramentas mais apropriadas. Em Java a gente tem o JUnit, que um dos frameworks mais conhecidos, de plataformas para teste e por exemplo, para mim, me parece que um code smell seria não fazer um setup próprio e fazer várias repetições que a gente tiver uma classe, por exemplo. Repetir um contexto bárias vezes sem usar propriamente um cleanup ou setup, são as principais funções que tem para rodar antes de todos os testes e depois de todos os testes. Mas é isso, eu nunca tinha ouvido falar sobre esse termo, apesar de achar que ele tem bastante referência com o que eu já considerava bons ou mas práticas na criação de teste.

P. Muito obrigado.