Pesquisador. Muito obrigado mais uma vez 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. Ok

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.

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

E. Eu comecei na empresa anterior que eu trabalhava e até então eu não tinha experiência com teste de software apesar de estar trabalhando a uns 3 anos. Na época a gente trabalhava com PHP e utilizava o PHP Unit como ferramenta de teste unitário. Então eu acabei aprendendo muito na prática mesmo do dia a dia e tomando alguns que já existiam como referência, e fui evoluindo a partir disso.

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

E. Eu pesquisei muito na internet. É difícil falar agora os sites específicos, mas tanto na própria documentação da ferramenta de teste quanto muitas vezes em fóruns, em sites voltados para tecnologia que trazem artigos, o próprio Medium tem muito material, hoje como desenvolvedora Java, que eu trabalho atualmente, a gente usa o JUnit e todas as ferramentas que trabalham junto o Mockito, a Remcrash, o Hash Assurant essas coisas para fazer também testes de integração, e agora está mais fácil de citar nomes porque tem um site que não sei se a pronúncia é essa, mas se chama By dung que tem muita coisa voltada pra Java e lá tem bastante conteúdo referente a muito do que se precisa na linguagem, então tem sido uma grande fonte de conhecimento para mim. Tanto no que a gente utiliza do Spring boot quanto as ferramentas de teste.

P. Em quais linguagens você cria testes?

E. PHP e Java

P. Qual plataforma do seu projeto atual?

E. Java

P. Me conte como é seu processo de criação de teste de unidade?

E. Eu já tentei aplicar o TDD mas ainda não consegui, então acaba que o teste é feito depois que a função que vai ser testada já está feita, e assim, eu busco levantar os cenários possíveis, até recentemente utilizando o JUnit 5 ele te dá mais possibilidades de ter testes parametrizados e tudo mais, ficou um pouco mais fácil de cobrir variadas possibilidades de teste, principalmente quando, no nosso caso lá, as vezes vai ter algum processamento relacionado ao valor da entrada, valor monetário mesmo, e eu procuro cobrir o método todo, pelo menos o caminho logico de tudo que o método faz. Uma coisa que acaba servindo como parâmetro é que a gente utiliza o Sonar e a verificação do quality gate, então antes não era necessário, de certa forma não era obrigatório, mas passou a ser ter o nível mínimo de cobertura de testes, então foi uma coisa que foi entrando na nossa rotina e era muito comum as vezes você faz o teste mas só está cobrindo o cenário principal, mas não cobre as variações que podem dar uma falha ou que vai entrar numa condição ou num bloco condicional ou não, então eu procuro tomar cuidado de realmente estar cobrindo todas as funcionalidades, mas como eu falei no início hoje acaba sendo um passo final. Depois que está tudo feito é que eu faço o teste.

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

E. Não.

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

E. JUnit, mas eu tenho colegas da minha equipe que usam Nock, que os testes são estruturados de uma forma diferente, dá para faze-los em Groove, se não me engano. Eu não conheço a fundo, só vi uma coisa ou outra, mas o que eu tenho costume de fazer é o JUnit 5.

P. Como é o processo de verificação de qualidade do código de teste criado, por exemplo: verificação de escrita desse código, alguém revisar o teste depois que ele foi criado, como é verificada a compatibilidade com o requisito e assim por diante?

E. A gente tem a prática de fazer code review, então vai ter uma ou outra pessoa olhando seu código, mas pelo que eu percebo, o teste ele mais olha se você fez o teste do que se ele está de acordo com o que seria necessário, entende? Eu penso que ele não tem tanta atenção quanto o código principal e muitas vezes eu falo até por experiência própria, dependendo como você estrutura os testes ou o que você estava validando nos testes, principalmente quando se está validando métodos void, que não dá simplesmente para fazer o retorno e bater com o valor, as vezes o teste passa, mas se você for fazer um debug, está tendo até uma exceção dentro do método e você não viu. Então realmente essa parte da qualidade do teste eu acho que ainda é negligenciada, as vezes, nem digo por, como diz, não é só uma questão de falta de atenção, mas as vezes a gente não tem tanto conhecimento a fundo para mitigar essas possibilidades, esse exemplo mesmo de estar havendo uma exceção dentro do método que o teste passava, foi algo recente pra mim e na minha cabeça eu tinha feito tudo certo mas tinha essa brecha aí.

P. Quando é submetido um código para code review, quantas pessoas revisam esse código?

E. geralmente uma só. as vezes mais pessoas. eu vejo na minha equipe que tem algumas pessoas que já tem aquele costume de, vamos supor, a gente utiliza o git hub, como nosso servidor de código, né? e a pessoa aprova e imediatamente já faz o merge, já toca pra frente, então as vezes quem fez o código, mandou para mais de uma pessoa, só que quando a segunda pessoa vai olhar, já subiu. Então acaba não adiantando muito, mas eu não gosto muito de fazer isso, prefiro fazer meus apontamentos, se for o caso aprovo, mas eu deixo para a pessoa que submeteu para fazer o merge na hora que deve até porque tem dessas situações também. E é interessante que seja mais de uma pessoa quando é possível até pela bagagem de cada um, as vezes um vai ver uma coisa ali que o outro não viu e no fim acaba ficando o resultado melhor.

P. como é o processo de manutenção do código de teste?

E. Eu percebo que a manutenção do código é realmente assim: "Bom, fiz uma alteração de um código principal que já existia teste para ele, mas esse teste não vai passar mais porque mudou aqui a estrutura, mudou a regra, algo desse tipo. Então ok, vou lá no teste e vou fazer as adaptações necessárias para que volte a passar, mas eu não vejo uma manutenção periódica ou coisas desse tipo no teste somente.

P. O que você conhece sobre test smell?

E. Test smell. Eu na verdade eu imagino que seja, mas eu nunca li nada sobre, então não vou dizer que conheço não. Considerando o que a gente utiliza sobre code smell, são más práticas, coisas que deixam o código não somente as vezes visualmente ruim, como também que acabam comprometendo seja o padrão que você está usando, seja orientado a objeto, seja o BDD, o que for, compromete aquele padrão ou simplesmente vai ser uma pratica ruim. Eu imagino que seja algo semelhante relacionado a teste, mas eu não tenho conhecimento, sobre.

P. Ok. Eu vou te passar o conceito, vou compartilhar com você dois exemplos de test smells, mas o princípio é exatamente isso que você falou. Na verdade, inclusive eles descendem dos code smells, mas existe uma suíte especifica de smells só para testes que difere dos de código. Então o conceito de test smell, 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.

E. Interessante.

P. Tem muito a ver do que você comentou sobre práticas de programação e que acabam tendo esse impacto na atividade. Vou compartilhar dois exemplos com você. O primeiro é o chamado Conditional Test Logic smell. Ele é considerado um smell porque na estrutura existem condicionais, repetição, que além de tornar a leitura mais complexa, no momento que esse teste falha fica mais difícil identificar de porque o teste falhou, então acaba prejudicando por exemplo a independência desses testes. O ideal seria que para cada condição fosse criado um teste separado para que eles fossem feito de forma independente.

E. Entendi.

P. Já o segundo é chamado de Empty test smell. Ele é considerado um smell porque não tem nenhum script executável. Então não vai ter informação de porque esse este teste está comentado, ao invés de estar sendo executado ou ter sido excluído. Então são coisas que acontecem na estrutura do código que podem impactar na compreensão desse teste ou porque ele está sendo executado dessa forma e também acaba impactando na independência, na repetibilidade e estabilidade dos testes.

E. Entendi. Uma coisa que eu já vi usando sonar lint por exemplo, é que quando você tem um teste lá e você não tem nenhum assertion por exemplo, não verifica nada, não faz nenhuma conferência. Ele mocka o que tem que mockar, ele chama o método principal e fim. Então eu acredito que seria um exemplo desse test smell. Essa falta de asserções e verificações.

P. Exato. Ou então dependência de testes, testes que estão relacionados a um mesmo recurso e dependendo da ordem que você executa esses testes, ele falha ou não, por exemplo, ou existem uma séries de test smells relacionadas a situações diferentes que podem existir dentro da estrutura do código.

E. entendi

P. Você tem alguma dúvida sobre isso?

E. Não

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

E. Algumas coisas eu realmente nunca soube, algumas acabam sendo code smells também, então as vezes você está com o olhar muito mais atento, mas eu acho que dizer assim que eu já sabia pensando "ok, não vou colocar a condicional e o looping porque isso é um test smell" eu confesso que não faço. Agora eu tomo o cuidado de sempre verificar alguma coisa no meu teste mesmo que seja lá algo que eu comecei a usar depois do que eu falei. Assert do not troll, por exemplo. Eu tenho que garantir que pelo menos que o meu método void não está lançando nenhuma exceção. Esse tipo de coisa eu já tomo mais cuidado durante o desenvolvimento do teste. E em relação as pessoas com quem eu trabalho, os códigos que eu já vi. Isso varia muito de pessoa para pessoa, mas de modo geral também que não levam, não tomam muito esse cuidado em relação a test smell.

P. Eventualmente esse cuidado, esse tratamento de smell ele é feito durante a manutenção do código de teste?

E. As vezes sim porque como a gente utiliza o sonar e ele mostra lá questões que precisam de melhoria e tudo mais, as vezes a pessoa que está fazendo o desenvolvimento, manutenção de alguma coisa já, pega aquela lista que está ali e vai mitigando aqueles pontos, entendeu. E por vezes tem a ver com o código, mas eu vejo que realmente o teste ele é dado bem menos atenção a ele de uma forma geral.

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

E. eu acho que o primeiro momento, é o desenvolvedor ter essa instrução. Porque às vezes a gente é passada essa necessidade de fazer o teste, a gente se preocupa em evoluir tecnicamente o conhecimento da linguagem, mas não nas ferramentas de teste e nas boas práticas de teste. Então, para esse primeiro momento, o uso de um linter ajuda. Porque ele já vai te chamar atenção de alguma coisa que você não vai ter reparado de cara e eu acho que é uma questão da prática mesmo da pessoa que está desenvolvendo de dar a devida atenção a esses pontos que se não eu acho bem difícil no momento da criação já conscientemente não ter test smells.

P. Estou chegando no final da entrevista, você tem alguma pergunta para mim, quer acrescentar alguma informação ou tem alguma sugestão para melhorar nossa entrevista?

E. Não. Sugestão de melhoria eu não tenho. Talvez a única coisa que de repente iria agilizar o processo é talvez o termo ter sido enviado antes, desde quando a gente conversou ou hoje mais cedo. Se você tiver me enviado, me perdoe, eu realmente não vi, eu me passei, porque quando a gente começou, porque assim só porque acaba sendo um pouco extenso e a gente já poderia começar com tudo alinhado. Em relação a pesquisa, eu achei ótima bem interessante, eu gostaria de saber o que motivou, pelo o que eu entendi, o objetivo é realmente ver isso no mercado de trabalho, nas empresas e tudo, né? Como que está acontecendo. Porque na teoria é muito bonito, todo mundo fala sobre teste. Tem um ponto também, na minha experiência anterior, eu fazia os testes também mas haviam os profissionais de QA e na minha experiência atual não existe, isso na equipe então nós somos responsáveis por essa qualidade e aí é um perde e ganha. Com certeza um QA vai ter mais domínio técnico dessas boas práticas e tudo mais, mas eu acho que também a gente como desenvolvedor se não aplicar a devida qualidade isso acaba voltando para a gente em forma de manutenção, várias coisas então o teste ajuda muito. Não sei se você quer falar mais um pouco sobre a motivação para a pesquisa, acho que vai ser legal saber.

P. Em relação ao formulário, na verdade eu não mandei mesmo antes, até porque já aconteceu de eu mandar o formulário e eventualmente a pessoa ou ela não responde e acaba não participando da entrevista então eu acabo tendo informação desnecessária, além de ter que procurar para vincular o formulário a entrevista, então eu tenho preferido mandar no primeiro momento logo quando a gente vai começar a entrevista mesmo. Sobre a motivação, tem um pouco a ver com o que você comentou. Eu tenho 8 anos de experiência de mercado como QA. eu trabalhei como QA durante 8 anos em uma empresa. Foi a empresa que eu tive contato com a área de teste de software, tirei certificação e tudo, e só quando eu entrei no mestrado, que eu voltei para a academia que eu tive acesso ao conceito de test smell, por exemplo. Então primeiro ponto foi justamente buscar entender se de fato existe esse "abismo" entre academia e mercado e o que está sendo discutido academicamente está chegando no mercado de trabalho, na indústria, os profissionais estão tendo esse acesso e num segundo momento identificar justamente no mercado, trazendo do mercado para a academia uma necessidade e possivelmente um ponto de melhoria a ser aplicado dentro das industrias. Então estou fazendo essas entrevistas para mapear esse cenário, identificar uma oportunidade de implementação e desenvolver se der tempo e apresentar uma melhoria para o mercado de trabalho e tentar aplicar ainda durante a pesquisa e ver se de fato é aquilo que atende a necessidade e melhorar esses processos direto na indústria e não manter esse conceito exclusivamente na academia.

E. Entendi, Legal. E as vezes a gente entre nós ali, discute algumas coisas do tipo: até que ponto e realmente benéfico ou faz sentido eu dizer "ah, minha API tem 100% de cobertura de teste" entende? Então às vezes a gente acaba se guiando muito só pelo percentual, só o número do que de fato a qualidade do que a gente está fazendo. Então isso é uma linha tênue mesmo também.

P. Exato, por que por exemplo, o test smell ele verifica a qualidade do teste então você pode ter uma cobertura alta, mas eventualmente na prática está deixando passar alguma coisa e devido a estrutura que o teste foi codificado e uma vez que você começa a trabalhar com esse conceito e melhorar as práticas de programação é uma forma de melhorar, não é a única, mas é uma forma de melhorar a qualidade do código e conseguintemente melhorar o produto que está sendo desenvolvido.

E. É verdade.

P. Muito obrigado mais uma vez por ter participado da entrevista.

E. Muito obrigada pelo convite. Gostei bastante

P. Caso você tenha algum contato que possa me passar para que eu possa realizar a entrevista eu vou agradecer bastante essa ajuda.