Pesquisador. Muito obrigado por participar a 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. Eu trabalhava na empresa M, uma empresa do grupo 1 e nessa época eu comecei a participar de dojos e daí fiquei próximo dessa forma de fazer teste, de fazer código orientado a teste e se faz meio que um jogo, você cria um cenário você tem que fazer um teste para passar esse cenário, aí vai trocando os participantes e tal. E daí quando eu percebi que eu poderia aplicar aquilo no meu dia a dia, eu passei a ver vários pontos positivos, mas a gente não tinha um ambiente de testes, a gente não tinha um código estruturado para testes, então eu comecei a fazer as coisas meio que de forma não acadêmica, não como a gente vê testes unitários hoje em dia ou testes de integração enfim. Eu fazia os scripts, quando eu tinha que escrever o meu script lá eu sempre fazia um outro script para testar ele de alguma forma bem rudimentar comparado com o que a gente tem hoje em dia de suíte de testes e tal. Eu comecei a ver o potencial daquilo ali porque me fazia escrever código cada vez melhor e com uma garantia de sucesso.

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

E. Na época eu trabalhava com PHP então eu me baseei muito na comunidade em volta do PHP que falava sobre o PHPUnit. Naquela época ainda era um projeto que não era usado em larga escala. Tinha algumas pessoas utilizando, mas muitas coisas eu tinha que buscar em stackoverflow, nuns fóruns mais escondidos em coisas mais estranhas, fora da superfície da comunidade de desenvolvimento, mas sempre buscando online.

P. Em quais linguagens você cria testes?

E. Hoje em dia em todas as linguagens que eu programo eu crio teste. Hoje em dia eu programo pouco, mas já fiz teste em C#, Javascript, PHP, Java, Kotlin, Groove, Typescript, Phyton e eu acho que é isso.

P. Qual linguagem do seu projeto atual?

E. É mais de um projeto. Eles estão divididos em PHP, Java Phyton e Typescript.

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

E. Eu sempre pego o requisito, avalio qual o cenário que eu preciso garantir para aquele requisito ser atingido e normalmente descrevo os cenários em modelo, que nem aquele modelo ganch do BDD, que é mais próximo da linguagem natural. "Se o usuário apertar o botão tal, tem que acontecer isso aqui" "dado que eu tive essa entrada no campo, eu tenho que ter esse resultado no final" e daí depois disso eu vou transformando essas premissas, essas frases em código e as vezes isso gera 1 ou 2 cenários e os cenários vão sempre acompanhando essa forma de ler, sempre preocupando não só com o teste, mas com a forma que aquele teste vai ser lido por quem está de fora ou um novo integrante do projeto.

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

E. Eu uso um template, não oficial, não formal que é. Eu sempre tento descrever o setup do que eu preciso, que são as informações necessárias para aquele cenário existir. Depois disso eu coloco a ação que aquele cenário executa, a ação necessária para validar aquele cenário e depois as asserções necessárias para garantir que aquele cenário foi validado. Normalmente fica num modelo "dado que eu tenho tal valor, quando eu executo determinada ação, eu tenho resultado tal" normalmente essas 3 frases são repetidas diversas vezes para compor um cenário.

P. Isso é documentado, ou é algo que está dentro da cultura da equipe ou da empresa.

E. Então, na equipe que estou agora não tem uma cultura de testes tão forte. Estou lá a 4 meses, e eu não estou mais como desenvolvedor, estou como coordenador e faço um acompanhamento mais de fora, eu não executo, não crio testes embora eventualmente faça para guiar alguém, trabalhar alguma coisa com alguém, mas eu tento documentar isso e tento convencer as pessoas seguirem esse modelo de acordo com os exemplos e mostrando os benefícios disso. Em outras empresas que trabalhei, isso já tinha ficado bem naturalizado com a equipe e a galera sempre seguia esse modelo. Eu sempre falo dos benefícios, eu sempre mostro os benefícios disso e elas são convencidas com o argumento.

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

E. Eu utilizo os frameworks de teste de cada ferramenta de cada framweork que estou utilizando. Dentro do ecossistema Java eu utilizo o JUnit e as ferramentas auxiliares, mockito, spend tests e essas coisas. Dentro do ecossistema PHP eu utilizo PHPUnit, dentro do ecossistema Javascript eu utilizo o Jest ou o Mocha. Para Phyton tem o Phytest que é uma ferramenta de tests, uma suíte de teste também.

P. Como é o processo de verificação de qualidade do código de teste criado, por exemplo: verificação de escrita, existe um code review ou seja alguém revisar o teste depois de criado. Como isso ocorre?

E. Sim. Eu sempre trabalhei com code review, quando alguém faz o seu código esse código tem que passar pelo crivo de outra pessoa pelo menos para garantir que atendeu as necessidades e uma das premissas do code review, uma das regras é você validar o cenário de teste do seu colega. Então você tem que ver se o teste está testando o que ele deveria testar mesmo, se todos os cenários que foram descritos para garantir aquela funcionalidade foram implementados, se não tem falhas no teste, por exemplo, deixar um valor hard coded que não garante que o teste está sendo efetivamente comparado com o código implementado, às vezes passa alguma coisa, um valor fixo ali que na verdade seu teste passa mas não e porque o código está certo, mas porque tem um valor fixo que deixa o teste passar. Então sempre tem essa variação, normalmente eu peço para alguma pessoa fazer e depois eu faço em cima de novo para garantir que as demais pessoas vão avaliando o código dos seus colegas e aprendendo dentro do que eles estão fazendo e como fazer uma crítica construtiva e acompanhar o desenvolvimento dos outros sem precisar estar fazendo junto.

P. Quantas pessoas fazem esse code review, pelo menos?

E. Normalmente são dias. Uma pessoa que não está envolvida na tarefa e eu, eu sempre acabo participando de todos os code reviews em outras empresas a gente conseguiu colocar quanto trabalhava em outras empresas, já tinha uma cultura bem disseminada, já tinha padrão bem rígido, então bastava uma pessoa qualquer validar, eu não precisava estar sempre validando. Se uma pessoa validava já seguia a diante. Nessa empresa que estou agora que não tem uma cultura muito grande, dessa equipe não tem uma cultura muito grande de testes então eu ainda me coloco como um avaliador necessário para garantir que a gente está tendo uma qualidade ali.

P. Passando para o processo de manutenção do código de teste de unidade, como ele é realizado? Por exemplo, em que momento é realizada a edição, exclusão desses testes?

E. Sempre que a gente rem novas funcionalidades, a gente tem que garantir que os testes anteriores continuam passando. Quando a gente implementa algo novo, por exemplo, tem uma nova feature eu ignoro os testes anteriores enquanto estou desenvolvendo e vou trabalhar só na suíte de testes necessárias para garantir aquela feature e quando eu concluir aquela feature, que ela está fazendo o que deveria, os testes estão passando, os testes fazem sentido, aí eu rodo todos os testes da aplicação. Eventualmente outros testes vão quebrar por mudança na estrutura para atender aquela funcionalidade e nesse momento é que faz a manutenção desses testes antigos. Vai ter que ver se algum cenário de teste já não faz mais sentido, se a lógica mudou então aquele cenário precisa ser adaptado ou então se é necessário criar outros cenários para garantir.

P. O que você conhece sobre test smell?

E. Test smell não conheço o termo, faço uma rápida correlação com code smell, que quando a gente fala que faz a gente identificar que tem um problema sem saber direito qual que é o problema a gente sabe que determinado comportamento é um code smell porque ele sinaliza que tem um problema ali, a gente bem qual é o problema. Acredito que test smells a gente pode fazer um paralelo de determinado tipo de teste identifica que tem algum problema.

P. O test smell ele descende do code smell. Eu vou te passar o conceito. Até você chegou a comentar um pouco antes a característica de um smell, e depois disso eu posso te passar uns exemplos, são dois exemplos que eu tenho utilizado para a entrevista em Java. A gente conversa um pouco se você tiver alguma dúvida você pode me perguntar porque as questões a partir de agora vão ter a ver com esse tema. “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.” Por exemplo. Você chegou a comentar de eventualmente de um teste ter um valor específico e não exatamente do código, e ele eventualmente vai passar ao invés de estar fazendo a verificação do código em si. Isso é considerado um smell.

E. Olha só. Nem sabia.

P. Existem smells relacionados a asserts, a construtores, a formas de codificação desse código. Eu vou compartilhar minha tela com você e vou te mostrar dois exemplos. Eles estão relacionados mais a estrutura, mas o exemplo que você citou de ter um valor fixo no código de teste, é considerado um smell. É algo simples de se entender o motivo. Não é algo que vai estar diretamente relacionado ao código do software mas vai ter impacto na realização dos testes. Esse exemplo, ele é chamado Condtional test logic smell. Ele é considerado um smell porque ele foi criado com estrutura condicional e de repetição. Então uma vez que o teste falhe vai ser mais difícil identificar de o porquê esse teste falou. Além do que ao depender do profissional ele pode ter uma dificuldade para entender a lógica desse teste em si, por exemplo. O outro exemplo que eu vou te passar é o chamado Empty test smell. Ele é considerado um smell porque ele não tem nenhum script executável. Então vai ser mais difícil de identificar de porque esse teste está comentado, porque esse teste não foi excluído. Eventualmente se tiver vários testes dessa forma dentro da suíte de testes a gerencia pode identificar que vai dar muito trabalho para localizar todos os testes e fazer a exclusão e optar por não testar mais uma vez que não está sendo acusada uma falha no código de software, mas que são testes que estão comentados e que não apresenta nenhum defeito no código

E. Gera incerteza. Não sabe porque isso está comentado.

P. Exato. Então são códigos que existem dentro da estrutura do código de teste que dificulta a compreensão, impacta na sustentabilidade da atividade de teste, justamente por essas questões, porque vai ter algo que eventualmente que o teste vai passar direto por um valor, um parâmetro específico, como você exemplificou, ou um teste vai ser mais difícil do programador compreender sua escrita ou a lógica dele vai tornar ele mais complexo. Então são coisas que dificultam a manutenção da atividade de teste, a sustentabilidade da atividade de teste pela forma que ele foi codificado.

P. Alguma dúvida?

E. Não achei bem legal. Já tinha esse conceito, mas não tinha nome para ele.

P. As vezes acontece. A pessoa trabalha com isso no dia a dia, mas não conhece o termo especificamente. Eu vou finalizar o compartilhamento e a voltar para as perguntas. Ok?

E. Sem problema.

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

E. Eu sempre trabalhei com sistemas ditos legados, são sistemas que já existiam e em determinado momento caiu no colo de alguém ou de uma equipe. E as pessoas tendem a olhar esse sistema sempre de uma forma ruim, de uma forma pejorativa "ah é um sistema legado e tal" e quando a gente tem essa visão dificilmente a gente consegue convencer a galera a testar e tudo mais. Então quando não tem teste ou quando os profissionais de teste são muito confusos e tal. Quando a gente assume, quando eu assumo projetos novos, códigos novos assim, minha tendência inicial é validar o que está legal de teste e o que não está legal excluir, porque geralmente gera muita incerteza e essa incerteza gera muitas indefinições e a gente fica na dúvida de o que que faz, porque que faz, porque que existe se não existe. O que eu tento fazer é apagar esses caras. Claro, dado que eu estou dentro de um sistema de versionamento de código, que eu vou poder recuperar, olhar para traz e ver. E a partir daquele momento se faz um marco que dali em diante a gente não vai em teoria não vai ter test smell. Claro, eventualmente acontece, mas a eu sempre tento seguir essa lógica tipo, o que veio até aqui, se a gente não conseguiu entender não faz sentido a gente se debruçar e acabar perdendo muito tempo dificultando muito nosso trabalho só para entender porque o cara fez o teste dessa forma. Muito se dá por causa desses test smells.

P. Vocês têm um cuidado na criação de não codificar testes dessa forma e na manutenção pelo que você comentou eventualmente vocês nem refatoram, vocês excluem, tendo o cuidado com o versionamento, mas vocês excluem e criam novos testes para cobrir aquele cenário.

E. Isso, exatamente. Às vezes o teste por exemplo o primeiro exemplo que você mostrou é um exemplo que com certeza ele tem um proposito, está testando alguma coisa, mas tem tanta lógica dentro do teste que fica difícil de entender o que é regra de negócio e o que é extração para fazer teste funcional. Então a gente tenta nesse cenário, é o tipo de coisa que eu já vi em outros lugares e a gente, quando quebra aquilo ali acaba quebrando em 4, 5 cenários diferentes e fica mais fácil de entender depois. Eu sempre coloco uma política de você não é responsável por todo o código, porque a gente acaba pegando coisas legadas, mas onde você encostou, você é responsável. Então se você fez uma alteração que está quebrando testes que podem estar errado, não importa quem fez aquilo, não importa porque feito, você é responsável por aquilo a partir daquele momento, e ai isso gera uma vontade das pessoas de sempre querer fazer algo correto para não ter esse stress de ser responsabilizado por coisa. "ah não foi eu que fiz" beleza, não foi você quem fez mais viu que estava errado e não corrigiu então a culpa é sua também.

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, eu acabo passando um pouco por esse cenário, pela forma que eu crio, pelo framework que eu comentei de levantar o cenário e fazer esses 3 passos tipo, o setup, execução e asserção que você acaba se obrigando a se limitar a cenários pequenos em testes auto-contidos. Então a gente acaba não chegando a esse. Acaba não gerando tanto test smell. Então não sei como prevenir isso porque não é algo que eu acabe me deparando no cotidiano.

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, achei super pertinente a entrevista. Achei bem legal.. Ainda estou curioso para saber detalhes do projeto...

P. Agora que já passei pelas perguntas da entrevista eu posso comentar um pouco mais contigo. Minha pesquisa é focada em test smell e entender o ponto que a indústria está em relação a esse tópico. Por exemplo se os profissionais da indústria conhece esse tema diretamente ou indiretamente, por exemplo. Você comentou antes de eu comentar desse assunto você comentou de um smell, mas eventualmente você não conhecia esse termos especificamente. Você fez um paralelo com code smell que está relacionado ao dia a dia que você já tenha trabalhado com desenvolvimento, mas o termo específico test smell você não tinha ouvido falar. Então num primeiro momento, saber se eventualmente a indústria conhece o tema e mais para frente identificar uma oportunidade de melhoria sobre esse assunto dentro da indústria. é mapear primeiramente esse ponto e identificar uma oportunidade e tentar implementar a tempo. Mestrado a gente sempre corre com o tempo de pesquisa para poder desenvolver alguma coisa e apresentar.

E. Muito legal. Bem legal a iniciativa e o projeto em si.. Eu gosto muito dessa área de teste, teve uma época que eu me considerava um evangelista de testes porque como eu falei. Eu sempre trabalhei com sistemas legados e a galera sempre tem medo de testar coisas legadas então eu vi o benefício disso e tinha que convencer as pessoas a usar isso porque ia ser favorável para mim também como parte da equipe.

P. O engraçado é que quem já tem uma vivência de teste acaba pensando na verdade principalmente o sistema legado o melhor que se tema se fazer e se fazer testes para saber o primeiro local que você vai atacar. Principalmente em se tratando de teste de unidade, a gente vai para a equipe de desenvolvimento que não tem tanto essa cultura de teste, então tem que ser um início de cultura para daí fazer esse processo que você comentou de evangelização a respeito de testes..

E. Muito legal o seu projeto. Quero ver os frutos disso. Ler essa tese depois

P. Eu vou deixar anotado aqui. A gente sempre tem o habito de a partir de uma publicação ter uma divulgação com as pessoas que participaram da pesquisa e contribuíram para ela, então a partir do momento que a gente tiver uma publicação a respeito, não só desse estudo, mas eu defendendo a dissertação da pesquisa eu vou entrar em contato contigo e compartilhar o resultado.

E. show. Sempre que precisar, eu estou aí. É um assunto que muito me interessa. Eu não segui a vida acadêmica. nao conclui a graduação e seguir a diante, mas o que eu puder contribuir na área é super legal.

P. Eu agradeço bastante. Agora o que eu preciso são de mais pessoas para poder entrevistas, porque o quanto mais informação eu tiver, vai ser mais enriquecedor para pesquisa.

E. Com certeza eu posso indicar algumas pessoas que trabalharam comigo na outra empresa das quais eu consegui convencer a fazer teses. Eram pessoas que não faziam testes e eu consegui convencer a fazer, dos benefícios e executar isso e criar uma cultura em cima disso e hoje evangelizam junto.