Pesquisador. Obrigado por participar dessa entrevista. Ela tem como objetivo compreender os processos de criação e de manutenção de código de testes e a qualidade do código de teste. 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.

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

E. Eu já tinha, eu trabalho na empresa J hoje, tem 2 anos, um ano e 6 meses que eu trabalho na empresa J, já tinha ouvido falar na empresa anterior, mas eu tinha posto em prática. Então eu comecei a pôr em prática mesmo testes de integração, testes de unidade, mais ou menos um ano e 6 meses que foi na minha entrada na empresa atual e eles querem fazer entregas com qualidade e aí eles determinam para a gente fazer teste nos softwares.

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

E. Geralmente eu costumo aprender por documentação, mas sempre que eu fazia algum curso tinha um básico de testes, eu tinha uma mania de pular telas de testes, não dar muita atenção, por conta de perceber algumas fragilidades, alguns preconceitos na parte de testes mesmo, quando eu comecei não via muito sentido, aí fui ver sentido nos testes depois e com o tempo. Então depois que eu vi a importância nos testes eu comecei a aprender tanto por documentação das bibliotecas de testes como por vídeo aulas também, as vezes comentário do pessoal, e às vezes até entre os colegas de trabalho mesmo que passavam algumas dicas de testes

P. Em quais linguagens você cria testes?

E. Atualmente eu crio apenas para Javascript, no caso NodeJS. Javascript e Typescript, no caso seria a mesma coisa. Também já criei para PHP.

P. Qual plataforma do seu projeto atual?

E. Node, Typescript.

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

E. É bem diferente do recomendado. As pessoas recomendam que se deve criar os testes primeiro e depois começar o desenvolvimento. Só que em startup em geral essa realidade não funciona. A gente sempre começa primeiro a aplicação e depois que a gente começa a aplicação é que a gente começa a aplicar o teste. A gente pensa num cenário depois que a aplicação está pronta, pensa em um cenário e começa a criação do teste em si. Daquela determinada coisa que a gente quer testar. Geralmente a gente faz mais teste de integração do que teste de unidade mesmo.

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

E. Não. É apenas programador que define ali mesmo. Está dando falha em tal coisa. Quando a gente passa tal argumento dá problema. Então através desse parâmetro que a pessoa passa, a gente cria um teste. Não tem nada que desenhe, um fluxograma, para poder começar o teste em si.

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

E. Como trabalho com Javascript a gente utiliza o Jest. Não sei se já ouviu falar, tem o Jest. A gente faz integrações ou com banco de dados usando o Postgree, para fazer o teste de integração e as vezes a gente utiliza mocks mesmo, sem precisar colocar em banco de dados, só o próprio Jest. A ferramenta que a gente utiliza no backend é o Jest, mas outras pessoas utilizam para teste end to end o Sypress, mas eu já fiz só em projetos pessoais, nunca fiz profissionalmente não.

P. Como é o processo de verificação de qualidade do código de teste criado, por exemplo: verificação de escrita, existe uma pessoa que revisa o teste depois de criado?

E. Essa parte é interessante, porque na verdade quem cria o teste em si, às vezes a gente recebe a plataforma pronta e começa a desenvolver, o DevOps em si prepara as pipelines para checar a cobertura dos testes, a gente não tem exatamente quem olha os testes para dizer "precisa colocar mais, precisa melhorar a forma de escrita" não, existe a parte automatizada mesmo, para dizer se tem a cobertura suficiente. A empresa define que tem que ter pelo menos 70% de cobertura de teste no código, senão não passa o deploy. Geralmente é outra pessoa que configura essa parte. E também se caso alguém perceber alguma coisa de errado no teste, sempre tem um code review, mas não exatamente por conta do teste, mas por conta do código (do software) mesmo. A gente escreve o código, escreve o teste e se a pessoa identificar alguma coisa no code review ali é barrado e a gente corrige.

P. Quantas pessoas fazem o code review, por exemplo, você faz um código, quantas pessoas revisam esse código?

E. Existe várias estratégias, mas na empresa é aberto. A gente aceita no mínimo 2 aprovadores. Não existe uma pessoa cerrta para olhar. A gente só pergunta se alguém pode fazer esse code review. Então se as pessoas olham ali e 2 pessoas provam já é o suficiente para a gente fazer o merge do código.

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

E. A manutenção é feita geralmente quando algo quebra mesmo e que não é relacionado aquele teste em si, a essência do teste. Quando a gente está mexendo em alguma coisa nova ali e vai criar um teste novo e quebra algo que já estava sendo utilizado, a gente percebe que aquele teste precisa ter uma manutenção para poder consertar o que ele quebrou. A gente vê o motivo e quando a gente percebe qual foi o motivo que quebrou aquele teste a gente faz a manutenção adaptando ele ao novo formato da feature que a gente está criando.

P. O que você conhece sobre test smell?

E. Smell? Nunca ouvi falar. Se você quiser passar uma referência para ver se talvez eu lembre pela forma de trabalhar, mas pelo conceito assim, smell, não consigo lembrar de nada não.

P. Eu vou te passar o conceito, vou te passar também dois exemplos de test smell para que fique mais evidente do que se trata, caso você tenha alguma dúvida você me pergunta, que as perguntas seguintes vão estar relacionadas a esse tema, 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. Então é algo mais conceitual, não é algo no código mesmo?

P. É algo no código, mas não algo que vai estar diretamente relacionado a um bug por exemplo, mas vai ser algo que eventualmente dificulte a manutenção da atividade de teste.

E. Algo que dificulte a atividade de teste.

P. Eu vou te mostrar os exemplos, acho que vai ficar mais fácil de você captar conceito. Esse exemplo é o chamado Conditional Test Logic smell. Ele é considerado um smell porque ele tem estrutura condicional de repetição dentro dele, então uma vez que esse teste falhe fica mais difícil de identificar de porque esse teste falou.

E. É, eu já ouvi falar desses conceitos do tipo que teste não seria para você colocar regra de negócio no teste, o teste seria simplesmente para jogar os parâmetros e saber o que você envia e o que você recebe. Isso daí do jeito que está por exemplo, com loops e ifs eu nunca cheguei a fazer em código de teste.

P. O outro exemplo é o chamado Empty Test. Ele é um teste que não tem nenhum script executável.

E. Eu já vi esse tipo de coisa, de gente comentar teste para poder passar.

P. Existem outros tipos de smells relacionados a construtor, a valor de parâmetro, então são coisas que são feitas, práticas de programação que são feitas na estrutura do código de teste que acabam prejudicando a manutenção da atividade de teste, seja porque fica mais difícil de entender porque esse teste falhou, ou porque eventualmente atrapalha a repetição, independência ou estabilidade desses testes

E. Entendi.

P. Alguma dúvida?

E. Não. Acho que entendi bem.

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

E. Hoje em dia, depois que a gente fez um refactory no sistema que a gente está trabalhando, isso não acontece mais. Não se permite logica de negócio, como você mostrou aí, não permite comentar o teste. E se for pego no code review a pessoa pede para remover, mas antes do refactory tinha uma bagunça as vezes a gente chega até a fazer isso, fazer essas más práticas, eu considero isso como más práticas, de comentar o código ou então passava por isso mesmo. A gente priorizava a entrega que a qualidade da entrega.

P. Como esse processo, os test smells são tratados na manutenção do código de teste?

E. Atualmente na verdade não chega nem a cair sobre a manutenção já é resolvido ali mesmo. Se caso a gente identificar, vê o blame aí e diz para pessoa "isso é uma má prática e não seria bom você fazer novamente. Você faz só dessa forma aqui que é o padrão que a gente segue" então a gente evita que esse tipo de situação aconteça.

P. Por curiosidade. Como ocorreu o processo de refactory? Houve um momento só para refatorar?

E. Isso. Por conta da instabilidade do sistema, estava realmente dando alguns problemas, então a gente sentiu falta que era por conta dos testes. A culpa era justamente porque a gente não testava e pelo fato da gente não testar foi necessário criar uma solução mais robusta, e o que a gente fazia era um monolito e também precisávamos criar microserviços. E aí foi necessário fazer esse refactory, quebrar esse monolito que era antes em microservoços e cada microserviços para evitar que acontecesse os mesmos problemas que estavam acontecendo no sistema anterior obrigou todo mundo a escrever os testes. Que cada feature tivesse no mínimo 70% de cobertura.

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

E. Eu acho que mais essa política mesmo que você tem que fazer. Você tem uma equipe grande de pessoas trabalhando no sistema. Acho que a forma mais fácil de evitar que isso aconteça seria criar uma política entre o time e criar uma política na hora de code review para evitar que caso alguém for fazer alguma coisa do tipo, porque sempre as vezes sem querer acontece, então para evitar seria bom ter um code review, sempre alertar pessoas da equipe sobre os padrões que a gente deseja para aquele projeto. Então acho que seria a melhor forma de evitar que isso acontecesse.

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

E. Não. Acho que está bem. Fiquei bem satisfeito com as perguntas. Eu acho que consegui, não sei se consegui contribuir bem com alguma coisa, mas espero que tenha contribuído.

P. Contribuiu sim. Toda contribuição é bem vinda.

E. Você como mestrando e como QA, você tem sua opinião na própria pesquisa?

P. Então, eu tenho hipóteses a respeito do que pode acontecer de cenário e a minha pesquisa ela vai comprovar minha hipótese ou rejeitar a minha hipótese. Então depende muito do que eu vou aplicando e obtendo de resultado.

E. Qual seria sua hipótese? É algo que você pode falar agora ou só no final da pesquisa mesmo?

P. Agora que a gente já conversou, eu posso falar para você e o que de fato eu peço para você é que não converse com ninguém a respeito. A minha pesquisa está diretamente relacionada a test smell, então por isso que sempre que se alguma pessoa não conhece o assunto eu tenho que passar o conceito e seguir com as perguntas porque elas estão diretamente relacionadas com esse tópico, então a ideia da pesquisa é primeiro mapear como é que os test smells ocorrem dentro da indústria. Não necessariamente como eles são inseridos ou não, mas saber qual o conhecimento da indústria sobre isso, então por exemplo. Você mencionou "Conheço isso como boas práticas mas eu não conheço o termo" então isso para mim é uma informação de certa forma relevante e...

E. Eu entendi como má prática, desculpe. Eu entendi como más práticas, comentar o código como você comentou

P. Isso. Desculpe, me expressei errado. Test smell acaba sendo ima má prática dentro da codificação, mas você não conhecia pelo termo "test smell" então essa informação pra mim é relevante porque por exemplo, você tem o cuidado disso de uma certa forma no dia a dia, mas o termo em si e as derivações, os tipos de test smell é algo que teoricamente você tem como conhecimento, então isso é uma oportunidade de melhoria na indústria, por exemplo. Outro ponto é, como eu te perguntei, se existe alguma sugestão de como prevenir ou evitar a inserção de test smell, ao depender das sugestões, ao depender do cenário, uma oportunidade de intervir nesse processo. Então minha pesquisa está primeiro no mapeamento do conhecimento industrial sobre os test smells e segundo identificar e se der tempo implementar alguma coisa para melhorar esse processo dentro da indústria.

E. Entendi. Interessante, eu gostei do tema.

P. Obrigado.

E. Quando você pretende finalizar essa pesquisa e entregar?

P. Meu prazo original era agora em agosto, mas por conta da pandemia e tudo, a gente vai estar postergando esse prazo, mas a ideia é que, pelo menos a minha ideia pessoa, é que eu não encerre esse ano nem estar com a pesquisa concluída e o mestrado concluído também.

E. Bacana. Boa sorte na finalização de sua pesquisa.

P. Obrigado.