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. Está ok?

Entrevistado. Está 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. ok

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

E. Comecei quando eu ainda programava em PHP, e na época eu trabalhava com o framework chamado kohana, e no kohana tinha o PHPUnit. Ele tinha esses testes de unidade e eu participava da comunidade de código fonte aberto deles lá e eu comecei a me interessar. Eu só não lembro a versão que era, mas eu comecei com o PHP.

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

E. Ultimamente eu tenho usado bastante o Linkedin, mas eu tenho vários mailings relacionados, porque atualmente eu tenho usado muito Typescript e NodeJS e não tenho utilizado outras plataformas de teste. Apesar de eu ter feito um desenvolvimento em Rust a um tempo, inclusive é uma linguagem que eu tenho interesse, pois o teste de unidade deles é dentro do código, geralmente na parte de baixo dos arquivos, e no caso do Kotlin eu aprendi no desenvolvimento com o Android Studio, aprendi o end to end deles lá, mas não me aprofundei. É mais ou menos padrão. Tem os spectations lá, os mocks, aquelas coisas que tem normalmente em todos os frameworks de teste.

P. Em quais linguagens você cria testes?

E. Rust, NodeJS, Kotlin e PHP, mas eu não tenho mexido com PHPUnit faz muito tempo

P. Qual linguagem do seu projeto atual?

E. NodeJS

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

E. Eu até tentei fazer aquele BDD, mas não me dei muito bem com ele, eu normalmente sigo o TDD, mas eu não faço os testes antes do código, eu normalmente crio o código para pensar nas interfaces, na API, mas eu já crio o código pensando no teste. Eu deixo vários membros da classe que for, deixo eles expostos para que se eu precisar fazer um mock, precisar fazer uma injeção de dependência, alguma coisa assim, para eu conseguir fazer os testes da forma como eu deveria. Pela experiência mesmo, eu já tenho uma ideia de como fazer isso, então eu acabo não criando os testes antes, pois tem gente que cria os testes antes de ter a API finalizada para depois criar o código mas não é o meu caso.

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

E. Não. Só na experiência mesmo.

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

E. No meu caso eu uso muito o Jest atualmente, que é uma ferramenta de teste unitário, mas que também serve para end to end. A ferramenta dentro do Jest é o Jasmine, que é usado para fazer os spects e de vez em quando eu uso o Chai ou Sinon para fazer os mocks e o Chai para deixar os spectations mais flexíveis, mas eu também uso outros frameworks como o AVA, Mocha, mas depende se o projeto já está em andamento eu tenho que escrever de acordo com o que já está pronto, mas normalmente se eu fosse começar algo do zero, eu normalmente usaria o Jest mesmo.

P. Como é o processo de verificação de qualidade do código de teste criado, por exemplo: verificação de escrita, existe algum processo de code review, compatibilidade com o requisito e assim por diante?

E. Tem. Tem tanto o code coverage quanto o test coverage quando o code review, então no caso passa por esses 3 e claro o teste de integração e o teste de funcionalidade no stading.

P. Dentro do code review, como funciona o processo, quantas pessoas revisam um código submetido?

E. Normalmente são umas 4 pessoas, mas a gente tem vários repositórios públicos, então acabam que várias pessoas acabam revisando ao mesmo tempo e também depende da disponibilidade. A gente não tem uma equipe específica, o que é o Quality Assurance. Por ser uma startup é uma coisa menos engessada digamos assim.

P. Como ocorre o processo de manutenção do código de teste de unidade?

E. A medida que o código vai sendo atualizado. Por si só ele não é atualizado a não ser que o código novo quebre os testes, aí ele necessariamente tem que ser atualizado.

P. O que você conhece sobre test smell?

E. Eu já ouvi falar, mas acredito que seja parecido com code smell, mas voltado para testes?

P. Você lembra onde você ouviu falar sobre o assunto?

E. Foi em um dos mailings que eu recebo.

P. Tem a ver sim com code smell. Você tem alguma dúvida sobre esse assunto?

E. Não, normalmente a gente usa essas ferramentas de análise de complexidade e também os linters ajudam bastante nesse sentido. A gente usa em todos processos, normalmente acaba deixando pelo menos o código antes de ser testado um pouco mais, uma qualidade um pouco melhor.

P. Essas ferramentas de complexidade estão dentro do Jenkins ou é uma ferramenta a parte.

E. É a parte. Code climent. Acho que é isso.

P. Você comentou que vocês usam uma ferramenta de complexidade para fazer essa verificação, mas além disso os tests smells são tratados de outras formas dentro do processo de criação do código de teste? E se sim, de que forma?

E. Normalmente, uma visão muito otimista do código, tipo, esperar que uma API vai sempre funcionar daquela forma, para a gente é um test smell. Você criar um mock com uma resposta sempre certinha, não testar outros cenários dentro do mesmo código, com códigos errados por exemplo, não esperar nenhuma exceção do código, isso é um test smell que eu imagino que seja muito grande. Pensar sempre de forma otimista.

P. Dentro do processo de manutenção do código de teste, você comentou que que os testes são mantidos a partir do momento que tem a necessidade de alguma alteração no código de software. Nesse momento, os test smells são tradados, caso existam dentro desse código de teste?

E. De vez em quando sim. Nem sempre. Para os testes, na verdade é mais aquele se está funcionando não é para mexer.

P. Então ele só vai ser alterado, refatorado se ele estiver alguma necessidade de alteração, fora isso ele vai ser mantido na forma que foi escrito originalmente.

E. Isso.

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

E. Aí é complexo. Sinceramente eu acho que vai mais da experiência, porque eu acho que você pegar um programador junior por exemplo, é muito difícil você evitar que ele faça algo assim. Eu acho que é meio complicado sem experiência você saber o que é um mau ou um bom teste. Então por exemplo, eu acredito que é aquele negócio, não criar statement de console dentro do teste, por exemplo, não tentar usar um martelo para fixar um parafuso, por exemplo. Tem gente que faz muito isso para tentar fazer com que o teste passe, mas não é o certo. Passar o teste ele só deveria passar o teste se tiver testando a funcionalidade e não passar por passar.

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

E. Se você estiver fazendo essa pesquisa pensando em ferramentas, em dar poder para quem cria ferramentas de qualidade, seria muito legal se fosse lançada alguma coisa usando esse estudo que eu fosse avisado.

P. Bacana. Minha pesquisa voltada para test smell, então entender como funciona esse processo dentro da indústria. Se de alguma forma os profissionais conhecem, se está sendo tratado de alguma forma no contexto industrial e se de alguma forma posso estar melhorando esse processo, e como eu vim da indústria e tenho esse pensamento mais voltado para o mercado, a ideia é que de alguma forma esse conhecimento de volta a indústria que é o meu foco de pesquisa. De qualquer forma eu sempre tenho o cuidado de, uma vez que tem publicação voltada para a pesquisa, compartilhar com as pessoas que fizeram parte desse processo. Então tenha certeza que assim que a gente tenha algum artigo, alguma publicação ou até mesmo se eventualmente minha dissertação for aprovada antes de qualquer artigo, com certeza vai ter algum compartilhamento com vocês.

E. Muito bom, excelente.

P. Muito obrigado por participar da entrevista. Caso você tenha algum contato que possa me indicar para que eu possa realizar a entrevista também, eu vou agradecer bastante.

E. Eu vou te passar um sim. Ele trabalha justamente com o que eu não trabalho que é o PHPUnit. É só falar que eu indiquei.

P. Ok, perfeito. Muito obrigado, boa tarde.

E. Boa tarde. Tchau.