Pesquisador. Mais uma vez muito obrigado por participar dessa entrevista. Ela tem como objetivo compreender os processos de criação e manutenção de código de testes e compreender como é realizada a verificação da qualidade desse código. Não há respostas certas ou erradas ou desejáveis ​​ou indesejáveis. Dessa forma, senta-se à vontade para dizer o que realmente pensa.

Entrevistado. Uhum.

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. O relatório final conterá os comentários dos participantes sem nenhuma referência aos indivíduos. Ok?

E. Uhum, ok.

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

E. Com teste mesmo, faz uns 10 anos que a gente começou a aplicar tecnologias ágeis. Nesse caso, com tecnologias ágeis os desenvolvedores tem um protagonismo na parte de teste também. Era trabalhando como software engineer só que como a gente utilizava TDD, nossa mentalidade de teste era bem mais forte então se acabava tendo a visão de testes grande, mas como tester mesmo foi a 5 anos na empresa I, mas no caso o produto era uma biblioteca então o teste era codificado também, então era tester com perfil desenvolvedor.

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

E. No começo foi documentação JUnit, mas principalmente procurando no Google tutoriais, mas o que realmente teste unitário isso. Test Driven foi uma palestra que eu assisti que tinha uma demonstração vendo como é que era realmente na prática o Test driven.

P. Em quais linguagens você cria testes?

E. Java, Javascript, Python, Rubi e C#

P. Qual plataforma do seu projeto atual?

E. Phyton e Javascript porque a gente testa o frontend também.

P. Conte como é seu processo de criação de código de teste?

E. Como faz TDD o código de teste, a primeira coisa que eu testo é se existe aquela classe. Então a partir disso eu vou iniciando a criação daquela classe, eu vejo se ela está instanciada e depois eu crio o teste para ver se ela está com os atributos corretos, testar se o construtor dela está com os atributos corretos, no caso se aplicado, teria que criar um factory method para criação do método, instanciar, passar os parâmetros e o próprio código, então vai desde criação do objeto.

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

E. Não. É a cargo do desenvolvedor, mas o teste dele vai passar por uma revisão. Então se não tiver teste unitário não vai entrar no branch master do git.

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

E. Eu utilizo para o backend o PhyTest em Phyton e para o frontend eu uso o Jest Javascript para React junto com Enzyme.

P. Como é feita a verificação de qualidade do código de teste?

E. A gente segue o fluxo normal do git. A gente trabalha on branch, faz o push, esse push vai passar pelo build contínuo que é no circle CI que é uma ferramenta de build na nuvem e aí vai passar por todos os checks, desde os linkers que são para verificar o estilo do código se está identado se está padronizado, tamanho de linha e tudo mais, até mesmo rodando todos os testes unitários daquele código, então se falhar, já lá no push request do git hub já avisa que falhou. Então quem está revisando nem vai se preocupar com isso, vai se preocupar se teve teste, se ele está escrito. É um revisor e o código só vai para master se ele aprovar. Também existe a cobertura de teste que é verificar se o teste novo está sendo coberto ou não. Então ele vai dizer qual o percentual de código novo que está sendo testado.

P. Passando para o processo de manutenção do código de teste, como ele é realizado?

E. Geralmente você está trabalhando na feature, você vai rodar na sua máquina localmente os testes unitários, então você vai saber se teve algum teste que você alterou vai quebrar e aí você faz a alteração.

P. O que você conhece sobre test smell?

E. Test smell? De cheiros? Muito pouco.

P. Eu vou te passar o conceito, posso te passar uns exemplos porque as perguntas a seguir tem a ver com esse tema, pode ser?

E. Beleza.

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.” Eu vou compartilhar com você dois exemplos de test smells, te explico o porque é considerado smell e caso você tenha alguma dúvida é só me perguntar.

E. Beleza

P. Esse smell ele é considerado um smell porque possui estruturas de condição e de repetição dentro dele. Então uma vez que esse teste falhe é mais difícil de identificar em qual momento dessa estrutura esse teste falou. Então seria interessante dividir esse teste. Cada uma das condicionais num teste diferente. O outro exemplo é o chamado empty test. Ele é chamado dessa forma porque não tem nenhum script executável dentro da rotina. São estruturas dentro do código que podem comprometer a manutenção da atividade de teste. Pode ter a ver com construtor, com assert e com a própria estrutura do teste mesmo. Alguma dúvida?

E. Não.

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

E. A gente faz isso naturalmente na revisão do código. Quem está revisando bate o olho e verifica essas más práticas.

P. Nesse caso essa verificação na revisão é feita tanto na criação quanto na manutenção?

E. Sim. Na manutenção querendo ou não quando você vai manter você vai ter que fazer um push request e só vai ser aprovado se alguém ver. A revisão vai existir sempre.

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

E. Existe algumas ferramentas, a gente não usa, mas eu já usei umas, que elas verificam se você está fazendo teste errado, então se você está testando algo que não está relacionado ao código. Na criação, eu acho que talvez tenha alguma ferramenta para ver isso, porém não dá para detectar todas as más práticas. Vai precisar do fator humano.

P. Estou chegando ao final da entrevista, você tem alguma pergunta para mim, quer acrescentar alguma informação ou tem alguma sugestão para melhorar nossa entrevista?

E. Não, eu acho legal, eu acho que você pegou as ferramentas, o fluxo de desenvolvimento, eu acho que é isso. As perguntas foram boas.

P. Muito obrigado por ter participado.