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

E. Certo

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

E. Eu estava cursando gestão em TI, que é um curso tecnológico do Paraná e não estava satisfeito com o curso e mudei para Sistemas de Informação. E nesse período tinha cursado um pouco de gestão em TI, tinha tido um pequeno contato com desenvolvimento, estrutura de dados e aí eu me inscrevi numa vaga numa empresa aqui na minha cidade que é uma empresa de ferrovia, para trabalhar com manutenção em informática, suporte, help desk. Que era o que eu já tinha experiência antes. Aí o entrevistador, que era o gerente da engenharia de software da empresa, tinha me entrevistado, mas aí a vaga não deu mais certo. Três meses depois ele me ligou e falou assim "Eu tenho uma vaga que precisa trabalhar com teste de software". Eu não sabia o que era, não vou mentir. E aí que eu aceitei o desafio sem saber o que era. Foi aí que que eu comecei a trabalhar com teste de software.

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

E. Cursos on-line, stack overflow e etc.

P. Em quais linguagens você cria testes?

E. Atualmente eu trabalho com Robot Framework em Phyton, mas já trabalhei com C# e Java também.

P. Qual plataforma do seu projeto atual?

E. Nós utilizamos microsserviços utilizando Javascript com Typescript, tudo na nuvem.

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

E. Meu time atualmente é o time ágil, então nós temos o trabalho com as histórias, então quando chegam as histórias, eu trabalho num time ágil de 5 desenvolvedores e eu sou o QA, então como base nas narrativas das histórias e também com perguntas mais específicas ao Product Owner e eu vou elaborando as minhas histórias com o robot, eu tambem Behavior Driven Development (BDD) para poder trabalhar os cenários das histórias junto com os cenários que eu estou criando no código em si.

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

E. Não. Atualmente nas coisas que eu trabalho não.

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

E. Visual Studio Code, e Mock and Chai para JavaScript.

P. Dentro da empresa, como é o processo de verificação de qualidade do código de teste criado, por exemplo: verificação de escrita, alguém revisar o teste depois de feito, verificar compatibilidade com o requisito ou classe de produção...?

E. Inicialmente não existia. Nós temos basicamente uma validação estática, pois como ele roda num pipeline na nuvem ele sempre tem que estar funcionando, senão o pipeline vai começar a quebrar. Mas essa verificação começou quando mais pessoas além de mim, eu trabalhei praticamente um ano sozinho e aí começaram a trabalhar mais QAs aqui comigo, aí eu estava fazendo essa questão... Então eu vi que tinha nosso repositório, aí eles trabalhavam em branch separado e eu acompanhava esses merges, ajudava eles e fazia esse code review, mas até então não tinha. Ninguém revisa os códigos que eu escrevo.

P. Passando para o processo de manutenção do código de teste, como é este processo na empresa?

E. Como comentei, dentro dessas histórias, e como os nossos serviços são baseados em micro serviços, essa manutenção ajuda, porque nós sempre vamos ter a ideia de quais serviços serão alterados e aí a gente sempre se depara com os testes. Eles vão começar a falhar com os nossos monitoramentos que a gente fica executando funcionalmente os testes. Então ele sempre vai trazer basicamente o fluxo do TDD, os testes vão começar a falhar e aí conforme eles vão corrigindo, e o código corrigido eles vão começar a passar novamente.

P. Me conte o que você conhece sobre test smell.

E. Nada.

P. A partir de agora a gente vai começar a falar sobre processos de criação e manutenção envolvendo test smell, então vou te passar o conceito, se você quiser eu vou te passar alguns exemplos e qualquer dívida, a gente continua a conversa e a gente segue a entrevista depois daí, ok?

E. Tem algo relacionado com o termo de desenvolvimento code smells?

P. “Test smells são pontos fracos na estrutura do código de teste, tendo um impacto negativo na sustentabilidade da atividade de teste, pois tornam os testes mais complexos, difíceis de entender e modificar, prejudicando a repetibilidade, independência e estabilidade dos testes.”. Você comentou que tem um certo conhecimento sobre code smells, não é isso?

E. Sim. Dentro de fatores de desenvolvimento. Patterns.

P. Os test smells eles descendem dos code smells, então o mesmo princípio, mas voltado para a área de teste de unidade especificamente. Eu vou te mandar um exemplo de test smell e te explico porque ele é um test smell, se você tiver alguma dúvida pode perguntar. Eu te mandei dois exemplos: O primeiro ele é do Conditional Test Logic. Ele é considerado um smell porque os testes eles têm que ter um objetivo específico. Então a partir do momento que você inclui uma estrutura condicional ou de repetição, uma vez que esse teste falha, dificulta o processo de identificar por qual motivo especificamente esse teste falhou. O segundo, ele é um Empty Test, então ele é considerado um smell porque ele não tem nenhum script executável dentro dessa classe. Os dois acabam prejudicando o processo de manutenção da atividade de teste porque, uma vez que você tem muitos testes sem um script executável, você vai ter que remover esse teste, ou identificar depois porque não tem nenhum script executável e isso prejudica a atividade, e no Conditional Test Logic, é justamente compreender, conseguir no momento da execução, no momento que ele falha, identificar o momento da falha, em que momento da repetição ou em qual estrutura de decisão ele falhou.

E. Certo.

P. Como existem nos code smells, há uma infinidade de test smells também, mas como você já conhece e principalmente em termos conceituais, ou seja, informações da estrutura do código de teste que acabam prejudicando a manutenção, a repetição e estabilidade deles e independência, OK?

E. Sim.

P. Você possui alguma dúvida sobre esse tema?

E. Não

P. Me conte como test smells são tratados no processo de criação do código de teste.

E. De criação... não tem um processo em si. No meu caso, quando eu crio, eu realmente tento lembrar, principalmente desses conceitos de codificação, sempre tento trazer isso, que é mais ou menos isso em relação aos test smells, principalmente essa questão que você falou, da complexidade, da repetição, da interdependência, então eu não tenho um processo, o que acontece mais quando tem a revisão, não de manutenção, mas quando eu vou revisar eu sempre tento trazer isso, a gente faz os comentários na parte do code review para que sejam satisfeitas as condições.

P. Era justamente a pergunta seguinte: "Me conte como test smells são tratados na manutenção do código de teste". Então é basicamente nesse processo de code review, é analisada a questão de estrutura e se há alguma necessidade de revisão.

E. Uhum.

P. Para você, como seria possível prevenir/evitar a inserção de test smells no código de teste no momento da criação do código?

E. Eu creio que tendo alguma ferramenta como, charper ou TS Linch, que você consegue programar regras customizáveis, eu creio que pode ajudar nessa análise prévia do código.

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

E. Eu tenho uma pergunta. Quando você fala teste de unidade, estamos falando não tanto de testes funcionais, mas de unit tests em si, certo.

P. Isso, exato.

E. É porque tem muitas ferramentas que são para testes funcionais, mas eu já trabalhei com teste unitários com C++, tratando mais sobre teste unitário.

P. Mais algum comentário, alguma pergunta?

E. Não.