Pesquisador. Muito obrigado mais uma vez por ter aceito participar da pesquisa. Ela tem como objetivo compreender os processos de criação e manutenção e verificação da qualidade de 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.

Entrevistado. Perfeito

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.

E. Tudo bem

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

E. Só uma coisa que estou me lembrando, eu coloquei que tinha dois anos de experiência, mas na verdade são 3. A empresa onde eu trabalhava tinha alguns testes, poucos ainda, estavam incluindo testes num sistema legado e aí era uma das competências que eu precisei desenvolver para atender as necessidades da empresa.

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

E. Livros e vídeos, principalmente do Robert Martin, vídeos de clean code. Ele aborda também parte de testes ele faz palestras também e alguns livros.

P. Em quais linguagens você cria testes?

E. C#, Javascript e agora estou com Java.

P. Qual plataforma do seu projeto atual?

E. Java e Javascript. É que tem testes que são feitos em Javascript e outra em Java.

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

E. A partir da identificação de um fluxo de negócios de uma determinada regra de negócios que eu preciso testar no sistema, que passa por um fluxo de condições do sistema eu testo uma das funcionalidades do sistema tenho a preocupação de que um teste não pode interferir no resultado do outro, os testes devem ser rápidos, deve agregar valor ao negócio, devem ter o código muito limpo porque quando alguém ler ter facilidade para ler e entender. Trabalho algumas vezes com TDD dependendo do cenário que estou testando, ou seja, faço o teste em alguns casos antes do código. É mais ou menos isso.

P. Existe algum fluxograma, template na empresa ou na equipe que padronize esse processo?

E. Não. Está tudo na minha cabeça. Ao estudar o assunto eu acabei me interessando bastante, entendendo o quanto pode ajudar na qualidade do sistema entregue e eu acabei me aprofundando. Mas as empresas que trabalhei com teste nunca tiveram procedimento normal de captura desse conhecimento. Não tinha definição de como fazer.

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

E. Eu trabalhava com Visual Studio. Trabalhava com NUnit e agora com Javascript trabalho com o Jasmine.

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

E. Tem empresas que o teste é desenvolvido pelo desenvolvedor junto com a implementação, então ele faz uma implementação e também cria o teste que vai demonstrar que aquela implementação está correta, está atendendo os requisitos de qualidade. Então quando ele submete o código para revisão vai com o teste junto que é o processo de revisão do código e outro desenvolvedor valida esse código, revisa esse código.

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

E. Ele é feito junto com o código fonte. Ao manter o código fonte, aí depende como o desenvolvedor trabalha e da implementação que ele vai fazer, mas ele pode trabalhar com o TDD, desenvolve o código de teste antes, cria o teste que vai quebrar porque não tem o código no sistema para atender aquele requisito, cria o código, aí ele quebra, ele faz a implementação no sistema para aquele teste funcionar, para implementação responder aquela funcionalidade de negócio ou ele cria o código, faz a implementação e depois cria o teste depois.

P. O que você conhece sobre test smell?

E. Conheço algumas práticas, assim, não formalizado, eu estudei algumas coisas, bom se um teste está lento demais, é um problema, tem que ser verificado, se o código é muito confuso, é um smell. Se dentro de um conjunto de testes a execução de um teste está afetando o outro teste, ou seja, se você roda cada um dos testes de forma independente, funciona e quando você roda todos juntos, um após o outo na sequência, dá um problema, isso é um test smell, um teste que você roda uma vez e roda várias vezes e tem resultados diferentes e cada vez que você roda, é um test smell. Bom, os principais que eu consigo lembrar são esses. Ah outra coisa também. Você tem um código que testa uma determinada funcionalidade, aí você vai na funcionalidade altera o código para forçar que o teu código dê erro. Que é uma das coisas que tem que fazer você depois que desenvolveu o teste você quebra o seu teste, ou seja aquela função que geralmente retorna número, você pede para retornar texto, retornar null para ver se o teste vai quebrar, então se o seu teste não quebra, mesmo você quebrando a funcionalidade é um test smell também. Esses são os principais que eu consigo me lembrar.

P. Como você conheceu este assunto?

E. Eu li mais de um livro, mas o melhor libro de todos em termos conceituais inclusive em termos práticos também é o XUnit test pattern, é uma bíblia para mim, apesar de não ser um livro novo ele tem muitos conceitos que podem ser utilizados hoje e tem uma parte do livro dedicada a test smell.

P. Você possui alguma dúvida sobre esse tema?

E. Não, é um assunto que eu explorei até um certo ponto que eu poderia explorar mais porque eu vou ter a necessidade de repassar mas nada que eu me lembre agora. Tem uma fonte bibliográfica que eu gosto bastante, mas eu ainda não me aprofundei tanto em test smell, eu sei alguns pela prática, e porque ter começado a ler, mas é uma assunto que eu quero me aprofundar, mas eu não tenho nenhuma dúvida.

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

E. Não existe um processo formal. Acaba sendo um pouco que eu estudei sobre o assunto e a minha experiência e como estou acostumado a fazer código de teste unitário eu vejo um código de outra pessoa e acaba me chamando atenção algumas características e eu explico. Mas não rem um processo formal.

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

E. A manutenção do código acaba sendo motivador de alteração do sistema, você tem que alterar algum teste. Lembrei de outro test smell, a gente tem que ter a preocupação do teste estar o mais acoplado possível do sistema, senão você pode ter problema de cada alteração que você tiver, cada alteração do sistema obrigar a mexer em muitos testes, isso é um test smell ou você criar testes que acessam diretamente classes internas do sistema, você expor detalhes da sua implementação para os teus testes. Os seus testes não devem conhecer isso. Voltando, na verdade os testes acabam sendo mantidos quando tem uma nova implementação ou quando tem uma necessidade de refatoração do código de teste. Às vezes você vai desenvolvendo e por questões de prazo você sabe que algumas coisas podem evoluir que não foram feitas da melhor forma possível, foram feitas de uma forma suficientemente boas mas podem ser melhoradas e aí você sinaliza isso em algum momento que você tem um tempinho a mais você refatora esses testes.

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

E. As pessoas precisam estar devidamente treinadas, principalmente consciente dos benefícios para elas mesmas, dos desenvolves e da empresa mesmo. A empresa no sentido dar condições em termos de tempo porque no começo é uma atividade nova e gera muitas dificuldades mas as pessoas precisam ter consciência dos benefícios que aquilo pode dar. Conhecendo de uma forma, tanto os benefícios como as técnicas e os test smells para poder diminuir isto e entender que o assunto não é um assunto tão básico, tão simples que qualquer um faz. Muitos desenvolvedores têm essa impressão e é um assunto que tem muitas técnicas que podem ser utilizadas. É um assunto mais amplo que a maioria dos desenvolvedores acreditam parecer ter. Basicamente isso, treinamento e conscientização do que o teste pode ajudar.

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

E. Não. Acho que não. Deu para você capturar... não consigo pensar em outras informações que conseguiria te ajudar no momento em termos de teste unitário.

P. Muito obrigado pelas informações, pelas suas contribuições.

E. Lembrei de uma coisa: É muito mais fácil o processo de inclusão do teste numa empresa quando vem de cima para baixo do que ir do meio para cima. É uma coisa que eu percebo. Eu trabalhei empresas onde a área de informática definiu, mas os usuários, os diretores, alta gerencia não comprava a ideia e não se conseguia vender e então era difícil de se conseguir tempo versus uma empresa que a diretoria é consciente dos benefícios e não abre mão e impõe para que existam essas práticas, é muito mais difícil quando a prática vem do meio, versus quando a prática vem de cima para ser implementada.