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

Entrevistado. Beleza

P. Bom, reforçando um item do termo de consentimento nossa conversa está sendo gravada. O objetivo disso é para comprar todos os detalhes, mas ao mesmo tempo ser capaz de manter uma conversa atenta com você. Garanto que todos seus comentários permanecerão confidenciais e que no relatório final conterá os comentários dos participantes sem nenhuma referência aos indivíduos, ok?

E. Beleza

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

E. Então, eu comecei por necessidade da empresa e um pouco por necessidade da minha carreira profissional mesmo, porque ao longo do meu crescimento profissional, eu vi que há uma necessidade de ter um software bem coberto por teste para manter uma qualidade de código afim de ter uma boa utilização, automatização de processos, é uma prática boa, o teste de unidade.

P. Ok. Essa necessidade da empresa que você trabalha atualmente ou de outra empresa?

E. Dessa empresa mesmo.

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

E. primeiro comecei. Por leitura de artigos na internet, principalmente empresas que tem blogs também no meio de li alguns artigos, alguns tutoriais gratuitos e cursos gratuitos na internet e depois eu através de um curso pago que a parte de escrita de testes era uma parte do curso, não era apenas de teste, ele envolvia mais processos de desenvolvimento, mas uma das principais premissas dele era o TDD, que é o processo que você escreve o teste antes de escrever o código e vai validando.

P. Em quais linguagens você cria testes?

E. Em Swift, que a linguagem que eu trabalho mesmo nativa da Apple. Para desenvolvimento de aplicativos IOS.

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

E. Então, nos projetos meus da empresa no momento é um mesclado de escrever teste após escrever uma funcionalidade, às vezes também fazer um TDD. Depende muito da tarefa que chega. Às vezes a tarefa, ela permite fazer um TDD, as vezes ela não permite fazer um TDD porque uma funcionalidade está implementada, então às vezes eu implemento o código de produção primeiro, às vezes implemento o teste primeiro. Nos meus projetos pessoais, que são mais de aprendizagem mesmo, estou tentando apenas aplicar o processo de escrita de teste antes do código de produção, que é o TDD.

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

E. Então, eu uso por hora para escrita mesmo o framework da Apple chamado XCTest que a própria Apple disponibiliza esse framework para gente. Há algumas outras práticas no auxílio só, como o processo de BDD que é você define bem os cenários, os casos de teste antes, de você escrever um teste porque você vai saber o que testa quais cenários você tem que cobrir com o teste, então também tem essa ideologia do BDD por trás, mas não é diretamente aplicada na escrita entendeu?

P. Entendi. Existe algum fluxograma template que padroniza, documente esse processo?

E. Não. Não uso nenhum fluxograma para padronizar não a parte de teste.

P. OK. Como é feito o processo de verificação da qualidade desse código de teste criado, a verificação da escrita alguém revisa esse teste depois de codificado

E. Então, a primeira premissa para garantir uma qualidade de código de cobertura de teste tudo mais é a definição mesmo de cenários, que entra nesse processo de BDD, e aí depois você escreve os testes para cobrir todos os cenários, cobrindo todos os cenários você sabe que seu código de teste está cobrindo bastante, está sendo eficaz. Segundo, também entra a parte de validação de outros desenvolvedores através de merge request via git, em que eles validam o código, eles verificam o código de produção, verificam o código de teste e tem que passar em alguns checklists de padrões de sintaxe, de qualidade mesmo, de organização, de qualidade do código e tudo mais.

P. Esse checklist é documentado?

E. A gente não tem documentado ele fica apenas como um template disponível através das documentações do projeto via git mesmo.

P. Entendi.

E. Uma pessoa quando vai fazer uma requisição de avaliação do código ele pode usar o template para ser aplicado lá, já fica exibido e ele só customiza o que ele precisar customizar.

P. Quantas pessoas fazem essa revisão?

E. No momento o time são 4 desenvolvedores IOS. Quando um faz a requisição, um ou mais outros desenvolvedores podem fazer a validação, geralmente é feita por um, mas pode ter uma segunda ou terceira opinião sim.

P. Por curiosidade, quando ocorre a necessidade de solicitar mais um desenvolvedor faça a revisão?

E. As vezes acontece quando a pessoa que vai fazer a primeira revisão não tem tanta certeza do que ela está marcando ou não, as vezes ela não está tão certa, tem alguma dúvida aí solicita alguma outra pessoa para fazer.

P. Passando para o processo de manutenção desse código de teste de unidade, como ele ocorre dentro da empresa, ou do time que você trabalha?

E. Geralmente ele ocorre junto com a manutenção dos projetos, caso for feita a manutenção em alguma parte do projeto que já está coberta por testes, são rodados os testes, é verificado se é necessário fazer algum ajuste, porque às vezes o código de produção é feito de uma forma que os testes passa a ser inválido ou passa a ter mais ou menos cenários então é necessário um ajuste.

P. Entendi, nesse caso por exemplo, se tem uma necessidade de excluir o teste, ele é excluído, ele é comentado...?

E. Então, excluir um teste por inteiro não costuma ocorrer, as vezes ele é reescrito e reajustado, não chega ocorrer a deletar um teste, mas a prática de escrita de teste, de manutenção de teste ainda é uma coisa que é nova aqui na nossa empresa, então é uma cultura que está crescendo. Então no futuro pode ser que surja essa necessidade, mas até hoje nunca foi necessário deletar um teste por completo não.

P. Entendi. O que você conhece sobre test smell?

E. Não conheço o termo.

P. Tudo bem. Eu vou te passar o conceito sobre ele, vou passar alguns exemplos são dois exemplos de Java, mas eu te explico porque ele é um smell, porque ele é muito mais baseado na estrutura do que na linguagem. E caso você tenha alguma dúvida, eu tiro contigo e depois eu faço as perguntas a seguir que estão relacionadas a esse tópico, tudo bem?

E. Beleza

P. Test smells são os 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. Entendi.

P. Vou te mandar agora o Skype dois exemplos. O primeiro que eu te mandei, ele é chamado o Condicionam Test Logic, ele é considerado um smell porque dentro da sua estrutura existem condicionais e repetição o que uma vez esse teste falha acaba dificultando a identificação do motivo real desse teste ter falhado. O outro exemplo, ele é o chamado Empty Test, ele é considerado o smell porque ele não tem nenhum script executável, então uma vez que exista testes dessa forma, é possível que a gerencia simplesmente desista de executar os testes por que teria que identificar todos para poder apagar e ela pode optar simplesmente por deixar de testar, de realizar testes e focar apenas no desenvolvimento porque não necessariamente são falhas que estão ocorrendo no software e sim testes que não deveriam existir e estão dentro da suíte.

E. Entendi.

P. Ou então por exemplo, não vai ter identificação de porque esse teste foi comentado. Ele poderia ter sido comentado em apenas um momento, por alguma circunstância, mas permaneceu dessa forma por exemplo. Então são situações onde a estrutura do código tenha a possibilidade de uma melhoria em justamente por essa questão da complexidade, de interferir na independência dos Testes e da sua repetição e estabilidade.

E. certo

P. Tranquilo? Posso passar para a próxima pergunta?

E. Pode sim

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

E. Na minha empresa?

P. Isso

E. Então, geralmente a gente não chega a ter muitos esses casos não, a gente não passa numa validação de código um teste que esteja com funções comentadas ou então um teste que tenha uma interação como essa de looping do for, mas que não tenha uma mensagem customizada em caso de falha que a gente consiga entender o motivo. Geralmente a gente consegue fazer uso também de alguns atributos que são passados no escopo da função de teste para casos de reaproveitamento de testes então através desses atributos eu consigo mandar meu compilador direcionar para linha correta do projeto em que o teste falhou, então as vezes isso evita muito esse tipo de falha de problema, de você não entender bem qual que é o contexto em que o teste falhou e tudo mais, então através desse atributo e também da mensagem bem definida de teste, definindo qual o resultado esperado qual que é o resultado que foi obtido, a gente consegue evitar esse tipo de problema.

P. Entendi. Esse cuidado é realizado também no processo de manutenção, tem algum processo especifico na manutenção para tratar os test smells?

E. Não, os processos durante manutenção e a criação são os mesmos. E tanto na criação quanto na manutenção, a gente tem essa etapa de validação do código e se for encontrado alguma irregularidade, alguma possível brecha, já é notificado e o código não é aceito. Então tem que ser restrito, então essa é a forma que a gente tem para evitar. E por enquanto tem funcionado bem.

P. Para você, como seria possível prevenir ou evitar a inserção de test smell no teste de unidade no momento em que ele é criado? Ou seja, aplicar alguma técnica ou ter algum processo na empresa que o desenvolvedor não insira um smell no momento que ele está criando o código. Porque pelo que a gente tá conversando, por exemplo, o desenvolvedor, ele cria o teste, ele tem essa autonomia para criar o teste e dentro da revisão pode ser identificado algumas oportunidades de melhorias e volta para ele para alterar, mas você identifica alguma possibilidade e se sim através de que forma para que o desenvolvedor não insira um smell ao criar?

E. Às vezes a criação de um template para facilitar a escrita do teste, não sei. Não consigo pensar no momento agora, talvez algum script para rodar e verificar possíveis brechas, não sei. Não sei ao certo dizer uma boa resposta para sua pergunta não.

P. Não existe uma boa resposta, como comentei no início. Qualquer sugestão que você pense agora para isso é bem vinda. É justamente com base nisso que a gente vai identificar alguma possibilidade para ser aplicada.

E. Eu acredito que eu já cheguei a ver em algum momento algumas bibliotecas que medem isso, que conseguem retornar para gente esse tipo de informação, esse tipo de métricas, mas eu não sou familiarizado com o uso de nenhuma delas não.

P. Tudo bem. Sem problemas. Estou chegando no final da entrevista, você tem alguma pergunta para mim, quer acrescentar alguma informação ou tem alguma sugestão para melhorar a entrevista?

E. Não, eu achei bacana, acho o tema muito pertinente da pesquisa, qualidade de código, escrita e manutenção é muito importante para escalabilidade dos produtos, confiabilidade nos produtos. Eu acho bacana o tema, eu gostei da entrevista. Muito bacana.