Pesquisador. Muito obrigado por ter aceito participar da entrevista. Como eu tinha mencionado, ela tem como objetivo compreender os processos de criação e de manutenção de código de testes e 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. 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.

E. Ok

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

E. Teste de software foi na verdade uma demanda que surgiu dentro do meu trabalho como um processo de amadurecimento do desenvolvimento. Lá a gente tem diversas áreas e na área especificamente onde eu trabalho que é na área de desenvolvimento na área de robótica, a gente nos projetos anteriores era muito heterogêneo, então atualmente, a forma com que se trabalhava, existe um pessoal de software, para desenvolvimento de aplicativos de modo geral. Eles já tinham uma forma de trabalhar mais estruturado, que usava teste unitário e tudo mais e no caso da minha área começou a ser usado recentemente como um passo no sentido de melhoria de qualidade das soluções de software aplicadas.

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

E. Basicamente internet. Conversa com alguns colegas mais experientes que já tinham mais experiência de código de teste de unidade, mas basicamente internet. Sites, artigos de site mesmo, não necessariamente artigo científico, mas basicamente internet.

P. Em quais linguagens você cria testes?

E. Eu atualmente em C++

P. Qual plataforma do seu projeto atual?

E. C++ e existe um framework que a gente utiliza lá, que é o ROS, que é o Robot Operating System, que é um framework de robótica amplamente utilizado no mundo e aí esse framework possibilita utilizar testes unitários para sua estrutura, digamos assim.

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

E. Ainda não é uma coisa estruturada. O meu particularmente é muito baseado uma vez que o código está desenvolvido, eu analiso quais são os pontos necessários para serem testados em termos de unidade e funcionalidades básicas. Então basicamente funções, testando basicamente funções públicas, uma vez que o GTest, que é o que a gente usa, ele não permite que a gente faça testes de funções privadas, então testando basicamente as funções públicas que são os pontos de acessos das classes do C++ no caso. Algumas pessoas têm processos diferentes, fala-se em TDD lá, mas nunca estudei muito sobre e nem é um processo bem definido dentro da nossa equipe por enquanto. Eu basicamente desenvolvo a biblioteca ou a classe e depois vejo quais são as características interessantes a serem testadas.

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

E. Não. Na verdade está sendo nova dentro a minha área. Como falei tem várias áreas então, por exemplo, o pessoal que desenvolve, que é da área de software, eles estão mais acostumados, mas na área de robótica tem algumas pessoas de ciência da computação, engenharia da computação, você tem engenheiros. Na minha área específica está sendo uma coisa nova, mas a gente não tem procedimento específico que define um fluxograma que o que deve ser testado, como deve ser testado, não existe nenhum procedimento organizando isso não. E meio que é por desenvolvimento e análise de pull request. Você abre o pull request, você vai ter algumas pessoas mais experientes ou que tem outras experiências e que vão fazer comentários no sentido de "faça teste nisso, ou corrija o teste" um aprendizado mais empírico, ou melhor, uma padronização mais empírica.

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

E. A gente usa para criar o teste, não sei dizer assim. A gente usa para rodar os testes unitários o Gtest, os testes unitários do Google e a criação de teste. Para criação do teste a gente usa o VS code.

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

E. Basicamente pela análise de pull request. Faço o pull request, solicito a avaliação por outras pessoas da equipe e aí pela experiência delas, e tem algumas pessoas que são mais experientes, e basicamente tem um desenvolvedor sênior lá que ele frequentemente é solicitado que avalie os pull requests e ele avalia a qualidade dos testes e outras pessoas também, mas em termos de avaliação da qualidade é basicamente isso. Pela análise de outras pessoas digamos assim.

P. Geralmente são quantas pessoas que analisam?

E. Varia mas em média 2 ou 3 pessoas.

P. Aí nesse caso é verificada a escrita do teste, se está condizente com os requisitos...?

E. Isso.

P. Como é realizada a manutenção do código de teste de unidade?

E. Não é mantido. Assim, o código de teste particularmente não é mantido. O código, todo o desenvolvimento de código, incluindo o código de teste é feito através de versionamento de github, mas se a pessoa vai desenvolver uma nova funcionalidade naquele módulo, é esperado que ela também faça o teste para aquela nova funcionalidade. A manutenção seria nesse sentido e pelo controle de versionamento.

P. Você já teve a necessidade de editar algum teste ou excluir algum teste já existente?

E. Não. Nunca teve essa condição não.

P. Já surgiu a necessidade de editar o código de teste de produção?

E. Sim, sim. Frequentemente, quando acontece um bug, que é descoberto, ou uma nova funcionalidade, é uma coisa frequente, porque como o processo de desenvolvimento é iterativo, então é desenvolvida uma funcionalidade que foi definida, mas com o processo de teste e hardware e com a integração com outros módulos, vai se percebendo algumas outras funcionalidades que são requeridas ou alguns bugs que são descobertos e o processo iterativo de aprimoramento do código da funcionalidade, e nesse aprimoramento do código principal, pode implicar em alterações de testes, do código de teste uma vez que uma funcionalidade foi alterada, o teste para aquela funcionalidade talvez tenha que ser alterado, ou incluir um novo teste de acordo com o bug que foi detectado e que o teste unitário não detectou anteriormente e deveria ter sido detectado. Então a manutenção é feita dessa forma.

P. O que você conhece sobre test smell?

E. Nada.

P. Eu vou te passar o conceito sobre o assunto, porque você pode não conhecer o termo, mas lida com ele de alguma outra forma. E aí se você tiver alguma dúvida você me pergunta. Vou te passar alguns exemplos e te dizer porque esses exemplos são considerados smells e depois a gente segue porque as perguntas a seguir têm a ver com esse tópico, tá bom?

E. Tá certo.

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 te mandar dois exemplos agora e te explico o porquê eles são considerados smells e caso você tenha alguma dúvida é só perguntar.

E. Ok.

P. A primeira imagem corresponde ao Conditional Test Logic smell. Esses dois testes são em Java, mas a ideia é que ele é considerado um smell porque ele tem estrutura de repetição e condicional dentro dele. Uma vez que esse teste falhe é difícil de identificar em que momento da condicional ou repetição esse teste falhou. Então seria interessante dividir esse teste em diferentes testes com cada condição, testes separados para que seja mais fácil compreender e caso ele falhe, identificar mais fácil o motivo. O segundo ele é o chamado Empty test, teste vazio. Então é um teste que não tem nenhum script executado, está tudo comentado, então não tem o que ser executado ali. Então é mais difícil de identificar porque esse teste está comentado, ou, porque esse teste ainda está nessa suíte, pois uma vez que ele está comentado, não tem nada a ser executado ele poderia ser excluído, por exemplo. Então são testes que podem falhar, mas que não é necessariamente um defeito de software e que a rastreabilidade dela acaba sendo mais complicada.

E. Concordo e assim, eu não tenho nenhuma formação, meu conhecimento é básico, mas particularmente não faria um teste desse vazio, pois uma vez que ele está vazio não precisa ser um teste. Não precisa estar aí se não tem nada nele. E essa questão de você colocar vários condicionais e um looping com várias coisas é uma coisa que eu acho que evitaria fazer justamente o que eu tenho feito lá é justamente se você tem várias condições você define todas as condições para justamente ter a rastreabilidade quando quebrar e saber o que você estava testando quando quebrou o teste.

P. Existem outros tipos de smells relacionados a asserts, a valor de parâmetro, a construção do teste. Então são tipos diferentes, mas que no geral são informações que acabam os testes mais difíceis de entender, modificar e que prejudica a independência, estabilidade e repetibilidade dos testes.

E. Ok

P. Alguma dúvida?

E. Não, por enquanto não.

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

E. Como a gente não tem uma estrutura e não conhecia esse conceito, eles não são tratados. Por esse senso comum, alguns casos acabam não sendo utilizados, porque pelo senso comum lá define-se que os testes têm que ser bem definidos para que tenham a rastreabilidade, mas dentro da características dos test smells serem tratados, eles não são tratados porque não é um conceito que a gente aborda lá.

P. Na mesma forma ocorre na manutenção do código de teste, nesse caso.

E. Exato. Exatamente. Seria pela questão do bom senso. Caso tenha uma situação dessa que você tem um teste e que ele fica difícil de mensurar, pelo comentário de outras pessoas ele é evitado. Um exemplo que a gente teve, é que a gente está no desenvolvimento de um software que envolvia um protocolo de comunicação e aí você manda mensagem e existia duas formas de verificar se essa mensagem enviada estava sendo correta. Uma era você, não lembro agora o tipo de condicional, mas uma você via que a mensagem tinha sido mandada errada, mas você não detectava o que estava sendo feito errado, e a outra eu acho que era por aquele assert equal, você colocava uma referência e colocava uma mensagem a ser comparada e com esse tipo de comparação ele dizia o que era para ser e como estava. Aí esse tipo de uso ele foi priorizado que, caso dê erro você sabe qual é a casa do erro de maneira mais simples, de uma maneira mais direta. Porque antes dava um código de erro, mas que não dizia muita coisa.

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

E. Uma vez que isso vai ser utilizado, que você vai ser monitorado, eu diria por procedimento. Procedimento no tipo de caracterizar os casos de test smell e que são casos que devem ser evitados, de serem feitos, como diretrizes de como devem ser feitos os testes e talvez se citar alternativas como você comentou. Ao invés de você colocar um for com diversos ifs, eu boto vários testes diferentes ou eu coloco um “for” mas que ao invés de vários condicionais com testes diferentes para esses condicionais eu penso em procedimento assim, de como desenvolver os testes.

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

E. Não. Não teria não. Como eu tinha conversado com você antes, eu não era tão experiente, mas como você disse que não tinha problema, eu resolvi me propor assim mesmo. Talvez possa ajudar de alguma forma.

P. Com certeza ajuda. Qualquer informação, qualquer experiência, considerando que seu conhecimento é básico, mas são informações que para a pesquisa é relevante. Porque como eu comentei, você pode não conhecer o termo test smell, mas de alguma forma você acaba tendo o cuidado na sua codificação de fazer um teste mais simples e mais direto. Independente da experiência ou se conhece o termo específico ou uma prática específica, o que interessa para minha pesquisa é justamente o que você faz no seu dia a dia. que é o foco do meu estudo e uma forma de melhoria no geral, identificando dentro desse processo, oportunidades de melhoria e propor uma solução dentro disso. Muito obrigado pela sua participação.