Pesquisador. Obrigado por participar dessa entrevista. Ela tem como objetivo compreender os processos de criação e de manutenção de código de teste de unidade, como também verificar como ocorre a verificação da 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.

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

E. Ok

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

E. Eu entrei na empresa G ano passado para trabalhar como estagiário. Fui submetido a um curso de desenvolvimento de Java, juntamente com teste unitário utilizando JUnit, e conforme o desempenho do curso eu fui promovido para trabalhar dentro de um projeto da empresa e foi assim que eu comecei a trabalhar, a ter contato com desenvolvimento de software e teste unitário.

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

E. Muita coisa eu aprendi sozinho, em slackoverflow, o site bem utilizado por programadores, e pesquisa no Google mesmo, que a gente pesquisa por teste unitário, como desenvolver, às vezes surge uma dúvida, as vezes dá um erro e a gente já e vai aprendendo, adquirindo conhecimento. Mas o conhecimento de verdade só vem de acordo com a prática.

P. Em quais linguagens você cria testes?

E. Apenas em Java.

P. Qual plataforma do seu projeto atual?

E. Java.

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

E. Primeiro a gente faz o desenvolvimento do sistema de acordo com o requisito dos clientes, depois a gente vai para a parte de teste unitário que é testes unitariamente o código, testar cada classe, cada função do código a gente tem que testar. A gente testa as possibilidades, tem que bater 100% de cobertura, o código, por exemplo, as vezes tem um “if” com duas condições, a gente tem que passar pelas duas condições, porque senão não dá 100% de cobertura. A gente desenvolve o sistema e depois vai testando unitariamente, a gente bola os testes na cabeça todas as possibilidades que podem acontecer com aquele código, inclusive erros a gente testa.

P. Quais ferramentas são utilizadas para criar teste?

E. A gente trabalha apenas com JUnit para poder fazer o teste. Ele é bem completo, já mostra a cobertura também para gente. Utilizamos o Jacoco também, mais para cobertura de testes.

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

E. Existe, eles mandam uma planilha com os casos de teste, nós chamamos de cenários de teste. Vamos testar com essa variável. Se a gente mandar uma variável desse tipo, o retorno tem que ser esse. Então a gente pega no teste, a gente monta o teste, manda a variável daquele tipo e espera o retorno para ver se vai bater com o que foi descrito na planilha.

P. Dentro da empresa ou da equipe, como é o processo de verificação de qualidade do código de teste criado, por exemplo: verificação de escrita, existe de alguém revisar o teste depois de escrito...?

E. Sim. A gente desenvolve o teste utilizando esse padrão necessários para uma boa qualidade de código. Depois que a gente desenvolve esses testes tem alguém superior, de conhecimento superior, que faz o code review do nosso código. Ele analisa, vê se está batendo a nomenclatura dos métodos, está tudo certinho e ele vê se pode subir para produção ou não.

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

E. Quando acontece a questão de ambiguidade. Às vezes você está testando a mesma coisa, aí se a gente vê questão de ambiguidade, testes que não são válidos, que não agregam nada, então a gente faz a remoção deles. Pode acontecer bastante, você testar duas vezes a mesma coisa. A gente analisa essas ambiguidades quando a gente submete esse código para code review, a gente está tão viciado no código que não encontra essas falhas, aí alguém que tem mais experiência acaba batendo olho e identifica, e solicita a remoção. A gente recebe a planilha, faz o teste de acordo com a planilha. Dificilmente ocorre da gente editar um teste porque durante o desenvolvimento do teste já está escrito o que deve fazer, então dificilmente a gente vai editar.

P. O que você conhece sobre test smell?

E. Nunca ouvi falar.

P. Eu vou te passar o conceito de test smell, te passar dois exemplos, falar um pouco sobre eles e depois as perguntas seguintes tem a ver com esse tema, então caso você tenha alguma dúvida é só me perguntar, a gente alinha e eu sigo a diante, ok?

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.” O primeiro exemplo é chamado de Conditional Test Logic, ele é considerado um smell porque ele tem estrutura de condição e de repetição dentro dele. Já o segundo exemplo ele é chamado de Empty test, porque ele não tem nenhum script executável, todas as linhas desse teste estão comentadas. Existem outros tipos de smells, relacionados a construtor, parâmetros, asserts, mas todos deles estarão prejudicando a independência, estabilidade e repetibilidade desses testes, ok?

E. ok

P. Pensando nesse conceito, como eles são tratados no processo de criação do código de teste?

E. Se aparecesse esses testes com comentário, sem validade nenhuma, ele nem passaria pelo code review dos nossos superiores. Isso ocorreria no code review se o desenvolvedor não tiver conceituado de que não pode mandar um teste totalmente comentado, com falta de legibilidade, depende do programador. Eu por exemplo não faria um teste desse, mas pode ser que algum novato que não tenha experiência ele pode fazer um teste desse e chegando no code review, é identificado que não é válido e que não vai passar.

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

E. É o mesmo processo da criação. A gente pode ter edições o durante a criação do código. Às vezes você está desenvolvendo o código, identifica uma falha e corrige ou apaga, ou na parte do code review. Pode acontecer nas duas etapas.

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

E. Eu acredito que conhecimento. Conhecer bastante, você consegue já ter uma boa noção para não cometer esse tipo de gafe. Fazer uma pausa durante o processo também pode ajudar a identificar erros ou uma forma de fazer melhor

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

E. Não tenho não. Foi bem completa a entrevista. Não tenho nada para complementar.