Pesquisador. Obrigado por participar dessa entrevista. Ela tem como objetivo compreender os processos de criação e de manutenção de código de testes de unidade como também a verificação da qualidade desse código de teste criado. 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 entrevista 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. Ok.

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

E. Eu comecei há aproximadamente uns 10 anos, criando teste unitários através de TDD. A gente fazia um design dos cenários primeiro, implementava os testes e após isso, implementava as funcionalidades propriamente dita. Bem num fluxo de TDD mesmo. Basicamente comecei minha carreira com essas práticas bem enraizadas no processo.

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

E. São através de posts, hoje muito através do medium, discussões que costumam ter dentro da empresa, que temos um canal muito aberto para levantamento dessas práticas e quais os tipos, quais as formas e pontos que a gente precisa testar, mas de aprendizado são os dois principais que eu diria.

P. Em quais linguagens você cria testes?

E. Hoje eu majoritariamente eu crio em Java, já criei testes para React, que é Javascript no final e Node JS também Javascript.

P. Qual plataforma do seu projeto atual?

E. Hoje é Java.

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

E. Hoje a gente não segue um TDD, a gente segue através do desenvolvimento da feature primeiro e depois a gente pensa e cria todos os cenários através dos critérios de aceitação que as features possuem, então o nosso pessoal responsável por escrever os critérios de aceitação, ele coloca vários cenários que a aquela feature precisa atender e esses critérios de aceitação são traduzidos em testes unitários prevendo esses cenários que a gente precisa cobrir

P. Quais ferramentas são utilizadas para criação desses testes?

E. A gente usa o JUnit, no Java que é um dos maiores, o Mockito, a gente já usou um pouco do PowerMockito, que é derivado do Mockito e um pouco do Assert J, para validações.

P. Existe na empresa ou no time algum fluxograma, template... que padronize esse processo de criação?

E. Dos testes? Tirando o critério de aceitação que é um padrão, não.

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, verificar compatibilidade com o requisito, cobertura de código... como é feita essa verificação da qualidade desse código de teste?

E. A gente tem o code review. Todos os nossos PRs (product release) precisam passar por aprovações, que a gente analisa tanto a qualidade do código quanto os testes que foram criados para saber se aquele teste está cumprindo o propósito mesmo, ou simplesmente está cobrindo código, a gente utiliza o SonarQube para análise estática, então coleta tanto o coverage da nossa aplicação quanto a complexidade e boas práticas que a gente está usando. E a gente garante a qualidade do software através de um CI que execute esses testes na nossa pipeline. Então todo pull request precisa passar pela pipeline, então todos os testes precisam passar nesse controle de qualidade.

P. Quantas pessoas fazem o code review no pull request?

E. Isso depende. A gente coloca uma quantidade mínima de 2 pessoas desenvolvedoras para analisar, mas hoje nosso time tem 3, 4 desenvolvedores, então pode aumentar. A gente tem um número mínimo, mas pode ser mais de 2.

P. Como é o processo de manutenção desse código de teste?

E. Basicamente duas formas. Alguma evolução de alguma feature, os nossos testes consequentemente vão quebrar e a gente faz os ajustes de acordo com essas alterações para cobrir os cenários que foram propostos, e a evolução das nossas features, criando novos testes. Se alguma regra de negócio é alterada, o nosso teste é alterado para contemplar essa alteração na regra, agora se uma nova regra de negócio foi adicionada, a gente cria um novo teste cobrindo esse novo cenário.

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

E. Não me é completamente desconhecido, mas eu não estou associando agora.

P. Como você conheceu este assunto?

E. Acredito que tenha sido de alguma conversa de projeto.

P. Eu vou te passar o conceito, vou te passar uns exemplos, a gente conversa um pouco sobre, tiro suas dúvidas, se tiver a respeito, pois as perguntas a seguir estarão relacionadas a esse tópico, ok?

E. Perfeito.

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.”. Então por exemplo, ele pode estar relacionado a própria estrutura, ou então algum parâmetro, construção desse teste. Eu vou te mandar dois exemplos, um vai estar relacionado ao código e o outro a característica que foi criado. O primeiro que eu te mandei, que deve ter chegado para você primeiro, ele é chamado de "Empty test", ele é considerado um test smell, porque ele não tem nenhum script executável. Vai ser difícil identificar porque o teste está comentado ao invés de ser executado ou excluído. O outro é o chamado Conditional Test Logic e ele é um smell porque ele tem estrutura condicional, de repetição dentro dele e que, uma vez um teste com essa estrutura falhe, fica difícil de identificar em que circunstâncias de fato esse teste falhou. O ideal é que ele fosse separado em 2, 3 testes de acordo com as condicionais ou repetições fossem necessárias para isso. Os smells acabam prejudicando, ele pode facilitar por um lado, a questão da cobertura e de cenário, então não necessariamente um smell vai estar relacionado a bug, mas vai acabar dificultando mais para frente a manutenção desse código de teste e que ao depender da empresa pode suspender a atividade de teste, focando só no desenvolvimento.

E. Eu não conhecia com o nome smell, mas pelo que eu pude entender, eu considero que sejam muito algumas boas práticas que eu fui adquirindo com o decorrer da experiência. Tipos de testes que fazem sentido outros que você não está validando, você está criando uma complexidade desnecessária ali.

P. Você comentou que apesar de não conhecer o termo, eram boas práticas que você passou a desenvolver com o passar do tempo. Como test smells são tratados no processo de criação do código de teste.

E. Eu acredito que muito por conhecimento e experiência do desenvolvedor, hoje a gente não tem uma regra, cada um tem a sua própria regra interna, sabe o que é um teste utilizando uma má prática ou indevido, e fica muito subjetivo esse ponto. Não tem uma fórmula ali.

P. Por exemplo, você tem esse cuidado enquanto está criando um teste, e eventualmente enquanto está revisando, mas não seria uma prática disseminada na equipe.

E. Não, não temos um momento para discutir esses pontos ou levantar isso.

P. Isso ocorre também na manutenção ou é exclusivo na criação?

E. Eu acredito que seja nos dois, a gente não faz esse levantamento.

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

E. Eu acredito que difundindo o conhecimento na equipe, seria uma das formas. A gente definir um padrão de boas práticas que os nossos testes precisam ter, alguns parâmetros que dizem que esse é um teste ruim, esse é um teste que a gente considera um test smell, que a gente não deveria estar fazendo, e o code review seria o momento mais adequado para aplicar esse conhecimento. Todos do time validarem e levantarem essa preocupação.

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

E. Acho que não. Acho que está indo de acordo.