Pesquisador. Muito 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 verificar como é feita a verificação da qualidade desse 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. Está certo.

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. Beleza

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

E. Foi em 2011 quando eu comecei a escrever teste. Foi bem antes da empresa que eu trabalho atualmente, estava na região de Aquavida em SP, em um projeto. Eu estava lendo sobre testes, já tinha ouvido algumas palestras, até não havia praticado e decidi começar a escrever algumas coisas.

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

E. Cursos pela internet. Medium e artigos.

P. Em quais linguagens você cria testes?

E. Em C# e em typescript.

P. O que vai ser relatado vai ser sobre sua experiência anterior. Nesse caso, qual era a plataforma do projeto?

E. Era em .NET

P. Como era o processo de criação de teste de unidade?

E. Eu já tinha a maior parte dos componentes prontos, então não teria como praticar o TDD, então nós rodávamos um script que fazia uma análise do código para ver o quanto desse código estava coberto por teste. Tinha que cobrir ao menos 80% dele. Tinha o teste unitário, o teste integrado que checava se havia comunicação de entre diferentes hosts.

P. Quais ferramentas são utilizadas para criar testes de unidade?

E. Visual Studio

P. Existe algum fluxograma, template... que padronize esse processo de criação de código de teste de unidade?

E. Não existia. Já existia alguns prontos eu copiava o que estava pronto e escrevia mais ou menos em cima daquilo.

P. Como era feita a verificação de qualidade do código de teste, por exemplo: verificação de escrita, havia algum code review. Você tinha comentado que existia a verificação de cobertura, mas em termos da escrita do teste, tinha alguma verificação de qualidade?

E. Do teste não. Não ocorria. Porém o que eu usava era o "Arrange, Act, Assert" (AAA). No Arrange a gente prepara o nosso teste, Act é propriamente as ações que a gente executa e Assert é que a gente vai chegar se o teste está retornando aquilo que a gente espera.

P. Como era realizada a manutenção do teste de unidade?

E. Em si, só tinha mudança no código unitário quando tinha mudança no código (do software), quando havia novos requisitos, então a gente adicionava mais testes unitários, mais casos de teste tinham que ser adicionado. Manutenção do que já existia, só se houvesse mudança de regra de negócio mesmo.

P. O que você conhece sobre test smell?

E. Não cheguei a conhecer ainda. Eu mais praticava teste unitário para poder cobrir aquilo que eu estou entregando.

P. Eu vou te passar o conceito sobre test smell, vou e mandar uns exemplos sobre, porque as perguntas a partir de agora vão ter a ver com esse tópico. “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.” Existem vários tipos de smells relacionados a asserts, redundância, mas eu vou te mandar dois exemplos para que a gente possa conversar a respeito e caso você tenha alguma dúvida depois disso é só perguntar e a gente vai conversando. Esse primeiro que eu te mandei é o chamado Conditional Test Logic. Ele é considerado um smell porque tem estruturas de repetição, condicional dentro dessa classe. Então uma vez que esse teste falhe, é mais difícil de identificar em que momento, baseado nessa estrutura, o teste falhou. Então seria interessante por exemplo refatorar esse teste e dividir ele de acordo com as condicionais e repetições. O outro exemplo é o chamado Empty test. Ele é considerado um smell porque não tem nenhum script executável. Torna-se mais difícil de identificar porque que o tyeste está comentado ao invés de ser retirado, por exemplo. São atividades que podem colocar em risco a atividade de teste pela complexidade, pela dificuldade, de ter uma dependência e assim por diante.

E. Acho que já vi uma coisa assim acontecer.

P. Vou passar para a próxima pergunta, ok?

E. Ok.

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

E. Nos projetos que eu passei, alguns quando era concessão da equipe, eles diziam que eram para ter o AAA. Não poderia estar vazio, na verdade, a aplicação ela tinha um processo de Continuos Integration então quando o código falhava o desenvolvedor recebia uma informação daquela falha que estava ocorrendo. Então a gente tinha que checar porque que falhou. Agora esse tipo de erro aqui, o primeiro fica complicado mesmo de entender em si. Colocar o AAA ajuda muito.

P. Esses cuidados também ocorriam também na manutenção do teste ou existia algum processo diferente para tratamento de smells?

E. Não, alguns projetos que eu participei que havia concessão da equipe toda então, sempre havia colega que tinha que fazer revisão do código antes de eu subir com ele, antes de ser aceito, fazia o pull request então algumas pessoas dava uma olhada nele antes. Tinha 4, 5 pessoas na equipe e geralmente a gente tinha que indicar duas.

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

E. Educação básica sobre como o teste deve ser escrito. A forma mais eficaz.

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

E. Não eu acho que está ocorrendo bem. Não conseguiria dar nenhuma sugestão agora.