Pesquisador. Muito obrigado por ter aceito participar da entrevista. Ela tem como objetivo compreender os processos de criação, manutenção e verificação da qualidade do código de teste de unidade. Não há respostas certas ou erradas ou desejáveis ​​ou indesejáveis. Dessa forma, senta-se à vontade para dizer o que realmente pensa.

Entrevistado. Claro

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. Tudo bem?

E. Tudo bem. Perfeito, sem problemas

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

E. Na verdade eu acho que foi a questão de sentir a necessidade de fazer teste foi mais um trauma de aplicações que eu desenvolvi no passado e que realmente viraram um grande monolito e que a manutenção ficou totalmente inviável, você adicionava uma nova feature e quebrava duas, três outras partes da aplicação. Eu acho que foi daí que partiu a necessidade de ter testes. De todos os tipos, teste unitários de integração, testes funcionais, surgiu dessa necessidade sí.

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

E. Foram alguns livros que eu já li, a maior fonte de aprendizados são artigos da internet, porque hoje qualquer tipo de aprendizagem que você pesquisa na internet você encontra. Então sempre livros e internet.

P. Em quais linguagens você cria testes?

E. PHP, Javascript e Elixir

P. Qual plataforma do seu projeto atual?

E. Atualmente estou trabalhando com elixir

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

E. na maioria das vezes, a não ser quando um projeto é muito simples, a gente trabalha utilizando o TDD como metodologia de desenvolvimento. A gente parte desse princípio, a gente desenvolve o teste, o teste falha, você cria a funcionalidade e vê como pode melhorar aquele código deixando ele um pouco mais limpo, menos verboso, utilizando boas práticas, vendo se encaixa algum pattern dentro da aplicação por exemplo. Hoje a maioria dos projetos que a gente desenvolve a gente utiliza o TDD como metodologia de desenvolvimento

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

E. Não. Não existe não

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

E. No caso do Elixir a gente está utilizando o Phoenix que é o framework mais popular da linguagem para desenvolvimento web e ele já vem com uma biblioteca de teste integrada que chama XUnit

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 de compatibilidade com o requisito entre outros fatores?

E. Na verdade tudo isso ainda está em fase de aprimoramento. Essa parte de qualidade de código aí, no máximo que a gente faz é um code review, para realmente sentir os impactos sobre as outras features que por acaso não tenham sido testado do sistema legado ou alguma coisa em particular, mas em relação a qualidade de código, eu acho que o que mais ajuda hoje é o próprio lint. o lint da linguagem e dentro do pipeline de integração contínua você estar sempre rodando o lint para deixar sempre o código dentro da qualidade esperada, mas no momento é o que a gente usa.

P. No code review quantas pessoas fazem a revisão de um código submetido?

E. São poucas pessoas, uma pessoa só faz o review bem rápido.

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

E. Eu vou ser bem sincero que ultimamente está muito difícil de fazer revisão de código, as necessidades por novas funcionalidades estão sempre, isso é culpa da gestão da empresa que, eu sei da necessidade, da preocupação de manter o código, de estar sempre refatorando aquilo e sempre buscando formas de melhorar, a aplicação a nível de código, mas a empresa por si exige funcionalidades, então hoje acho que a gente está mais voltado para o desenvolvimento de novas funcionalidades do que manutenção de código, para ser sincero.

P. Quando ocorre, como o processo é feito?

E. eu vou ser bem sincero, eu vou falar da empresa anterior, porque nesse momento estou em transição para uma nova empresa. Eu trabalhei cerca de um ano na empresa anterior e durante esse um ano foi aplicado só a questão de desenvolver nova funcionalidade, acho que a gente não voltou para fazer nenhum tipo de manutenção.

P. O que você conhece sobre test smell?

E. Esse eu nunca ouvi falar não.

P. Eu vou te passar o conceito porque eventualmente você pode já ter trabalhado algum aspecto mas não conhece o termo especificamente, vou te mandar uns exemplos, são alguns test smells em Java, mas o importante é em relação a estrutura e não especificamente a linhagem e depois as perguntas a parti de agora terão a ver com esse tópico, está bom?

E. Ok, tudo bem.

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.”

E. OK

P. Esse exemplo que estou te mostrando ele é o chamado Conditional Test Logic smell. Ele é considerado um smell porque existe estrutura de condição e de repetição dentro desse teste. Então ao depender da experiência, o profissional pode ter dificuldade para entender, ou uma vez que esse teste falhe fica mais difícil identificar diretamente o porque esse teste falhou. O outro exemplo é o chamado Empty test smell. Ele é considerado um smell porque não tem nenhum script executável nele. Vai ser mais difícil de identificar porque esse teste está comentado e não está sendo executado ou então porque ele não foi excluído e está apenas comentado. São coisas que vão impactar na independencia na estabilidade dos testes. Tem outros tipos de test smells relacionados a dependência a valor de parâmetro, a asserts e assim por diante.

E. Ok

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

E. Na verdade para evitar esse tipo de má prática a gente tinha o costume de desenvolver em peer programing, o desenvolvimento em pares, então isso ajuda bastante a gente a evitar a tomar esse tipo de caminho tanto no desenvolvimento de teste quanto da própria funcionalidade, e acho que desenvolver em pares ajuda bastante a você não cometer esse tipo de erro.

P. A outra pergunta é sobre manutenção, mas como você comentou que basicamente você trabalhava com desenvolvimento do que manutenção, eu vou passar para a próxima pergunta, ok?

E. Ok, tranquilo

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? Você comentou de peer programing, mas se você tiver alguma outra sugestão, fique a vontade para comentar a respeito

E. Eu acho que uma das coisas importantes é o desenvolvedor saber o objetivo daquilo que ele está fazendo, porque quando você sabe o objetivo daquilo, quando você consegue interpretar o objetivo daquela funcionalidade da maneira correta, fica muito mais claro para você desenvolver também. Então uma das coisas importantes também é trabalhar com metodologias ágeis que eu acho que você vai ter um entendimento melhor da funcionalidade e vai poder aplicar isso de uma maneira mais consistente, tanto no desenvolvimento do código quanto na hora de desenvolver funcionalidade. Eu acho que juntando as duas metodologias, a metodologia ágil com a metodologia do Xtreme program eu acho que você consegue desenvolver de uma forma bem bacana e evitar bastantes erros.

P. Você comentou sobre pair programing, mas era um processo na empresa que você trabalhava, você também considera que isso pode evitar a inserção de smell?

E. Eu acredito que evita, com certeza. Um dos motivos para você fazer pair programing é justamente isso, acho que quando se tem outra pessoa, está programando com outra pessoa, um fica meio que fiscalizando, dando uma atenção ali para que erros simples não sejam cometidos

P. Estou chegando ao final da entrevista, você tem alguma pergunta para mim, quer acrescentar alguma informação ou ainda tem alguma sugestão para melhorar nossa entrevista?

E. Não, eu acho que está bem tranquilo. Está Ok. Para ser sincero eu acho que é importante, não sei qual o objetivo final da entrevista, mas por exemplo na empresas que eu estava anteriormente ela era uma fintech de credito universitário, por exemplo e não era uma prática, há um ano atrás, fazer testes dentro da empresa. Você pega todas as aplicações que tem lá, quando eu saí a gente saiu do bitbucket para o git hub, a gente migrou cerca de 120 repositórios e nenhum desses repositórios tinha teste de unidade para as aplicações, por exemplo. Então uma das coisas importantes é isso, as pessoas ainda não estão conscientizadas da necessidade de ter testes no desenvolvimento, principalmente quando se fala de TDD que a maioria das pessoas acham que o desenvolvimento orientado a teste é como se estivesse codando duas vezes, é como se estivesse trazendo mais trabalho. As pessoas não conseguem a enxergar a longo prazo ainda. Isso estou falando com a experiência da empresa que trabalhei, lógico que estou me resumindo aquele grupo. Eu acho que também a empresas por si só, a própria diretoria deveria exigir esse tipo de qualidade do desenvolvimento. Toda essa urgência, quantas vezes eu ouvia "sobe sem teste mesmo" ou o teste não está passando "para de realizar o teste e foca na funcionalidade" isso era bastante comum eu escutar lá e não me agradava muito, mas de um ano para cá mudou tudo isso e bastante aplicação está sendo testada, tem aplicação hoje que tem cobertura de 90% de teste. Tudo bem que cobertura não quer dizer muita coisa, mas eu acho que dava equipe conseguir já entender a necessidade daquilo e começar a colocar em prática, eu acho que já é muito legal. Basicamente é isso aí. Talvez eu pude acrescentar muito sobre a questão de qualidade ou de manutenção da qualidade do código é por isso, eu vinha de uma empresa onde isso não era uma prática comum, que isso estava se adequando as novas necessidades nesse momento.

P. Não tem problema, essa é uma realidade que eventualmente eu posso me deparar com outras empresas, então o fato de você não ter um relato sobre manutenção não impacta na minha pesquisa. Ela só acrescenta porque justamente fala de um cenário que é como se fosse o cenário de implantação que você acabou vivenciando. O contexto era muito mais de criação do que se manter aquele código de teste. Isso é super tranquilo e enriquece a pesquisa porque é uma outra perspectiva. Minha pesquisa é focada na qualidade de teste de unidade, especificamente sobre test smell, então ter um relato de por exemplo a gente não tem uma preocupação em relação a manter esses testes porque o meu contexto é muito mais de criação. "ok" bacana. Você não tinha o conceito do assunto, mas d alguma forma isso já era cuidado no dia a ainda do trabalho. isso também é relevante para minha pesquisa pois mostra que já existe internamente uma qualidade desse código de teste mesmo sem saber especificamente um termo que é utilizado academicamente e em alguma empresas. Então o seu relato ajuda e enriquece a pesquisa também

E. Maravilha então. Fico feliz de poder ter ajudado

P. Beleza. Muito obrigado.