

Podcast

Disciplina: Teste e inspeção de software: técnicas e automatização

Título do tema: Processos e Casos de Testes automatizados

Autoria: Stella Marys Dornelas Lamounier

Leitura crítica: Marcilyanne Moreira Gois

Olá, ouvinte! No podcast de hoje vamos falar da Técnica TDD – Técnica Orientada a testes.

Algumas empresas de desenvolvimento de software ainda possuem uma grande resistência na utilização de testes de software alegando que este processo pode atrasar a entrega do sistema e quando se trata de automação de testes relatam que ainda podem gerar um elevado custo devido a aquisição de ferramentas de testes.

Sabemos que quanto mais cedo se começar a testar os softwares, menor é o custo com sua manutenção. Existe hoje no mercado uma técnica bastante curiosa que é utilizada por empresas que adotam a Metodologia Ágil para o desenvolvimento de software. É um novo paradigma de teste, conhecido como TDD ou *Test Driven Development* / Desenvolvimento orientado a teste, em que primeiramente é escrito o teste automatizado para depois desenvolver o código fonte. Esta técnica está muito ligada no desenvolvimento de código que, utilizando *Clean Code* ou Código Limpo, tem como estilo a criação de códigos focados em boa escrita, que facilite a leitura, ou seja, de forma limpa, entendível e objetiva, melhorando assim a sua produtividade. Estes testes têm como finalidade nortear o programador a fim de facilitar o entendimento do projeto, propor melhorias, inclusão de novas funcionalidades, reduzindo erros, antecipando problemas e propondo soluções.

Sim, está é a função do TDD, escrever o teste antes do código.

De início esse novo modelo pode soar um tanto quanto estranho, mas vejam como esse técnica funciona na prática.

Podemos utilizar antes mesmo do escrita de códigos os testes de unidades, observar as falhas e posteriormente implementar as alterações necessárias, até chegar a um ponto ótimo, acarretando assim um código mais limpo e menos suscetível a erros ao longo do tempo.

A técnica pode ser executada em diferente linguagens de programação e, o que deve ser levado em consideração é basicamente o seu ciclo de criação, que é naturalmente simples, vejamos:

O primeiro passo é ler, entender e analisar recursos a serem implantados ou a indicação de erros.

Posteriormente, na etapa vermelha, o desenvolvedor escreve o código de teste para uma funcionalidade, que ainda não foi implementada. Com a utilização de

ferramentas automatizadas o teste é executado e muito provavelmente irá falhar, mas isso é o que o desenvolvedor já espera, pois a funcionalidade ainda não foi planejada.

Na segunda etapa (verde) é construído o código da funcionalidade que irá passar na etapa realizada anteriormente, isto é, no teste que foi executado na etapa vermelha, se estiver tudo certo, não há necessidade de repetir o processo.

É realizada a refatoração, onde é feita a limpeza do código, sua estruturação, diminuição da quantidade de linhas e melhora da sua performance.

Volte ao passo 1 e faça tudo novamente.

Mas qual a diferença de se aplicar o TDD e executar testes só depois da implementação pronta?

Quando os testes são executados ao final, o sistema está concluído e o desenvolver passou muito tempo codificando e não obteve nenhum retorno com relação ao código, o que pode ocasionar atrasos e aumento nos custos, caso seja necessária alguma alteração. Concordam que, quanto menos código escrito, mais fácil a realização de mudanças? Ao aplicar o TDD o desenvolvedor divide seu trabalho em pequenos pedaços, escrevendo pequenos testes, implementando módulos pequenos de funcionalidade, repetindo sempre que possível e a cada teste escrito, e mesmo que pequeno, o desenvolvedor recebe um feedback da funcionalidade testada e quanto mais precoce for recebido o feedback melhor para a equipe de desenvolvimento.

Apesar de possuir excelentes benefícios como redução nos custos, criação de códigos mais confiáveis e limpos, atendimento aos requisitos levantados, é importante que o desenvolvedor tenha bastante conhecimento nos detalhes e especificação do sistema e, principalmente compreensão das regras de negócios.

Este foi nosso podcast de hoje! Até a próxima!