Práticas ágeis, Anti Patterns e Shift Left

Pirâmide de teste

A piramide de teste da uma ideia de quantos testes devemos ter em cada um dos grupos ou camadas, unidade, integração, UI.

O modelo cupcake deve ser evitado, é o modelo onde se tem muitos testes automatizados em todas camadas.

Outro modelo a ser evitado é o Ice Cream, onde tem poucos testes de unidade, poucos em serviços e muitos testes em UI, exploratorios e manuais.

Isso pode acontecer quando o time tem pouca conversa para planejamento e modelagem de uma estrategia de teste.

BDD

BDD significa Behaviour Driven Development ou desenvolvimento orientado a comportamento.

É uma abordagem que se apoia em testes e comportamentos do usuario ou sistema.

Outro é uma evolução feita a partir da implantação das técnicas TDD (test driven development) Desenvolvimento orientado a teste.

**Dado** um condição

**Quando** eu executo uma ação

**Então** o resultado é esse

Gherkin

O vocabulario do BDD é o Gherkin, que é um linguagem que foi criada para descrições de comportamento, ela remove detalhes de logica de programação e foca no coportamento que a funcionalidade deve ter.

Para eliminar linhas e condições repetitivas podemos usar o **Contexto** com o **Dado.**

Cucumber

// Utilizar “#language: pt” para definir idioma em potugues.

//Contexto para diminuir linha iguais, exemplo:

“Contexto:

Dado”

//Estrutura de cenários, exemplo:  
“Cenário: (texto)

**Dado** um condição

**Quando** eu executo uma ação

**Então** o resultado é esse “

//Para usar tabela de exemplos deve usar esquema de cenário, exemplo:

“Esquema do cenário: Soma de 2 números

**Dado** que eu acesse a calculadora

**Quando** eu somar <numero 1> + numero 2>

**Então** o resultado deve ser <resultado>

Criar tabela,

“Exemplos:

| numero 1 | numero 2 | resultado |

| 1 | 1 | 2 |

| 1 | 2 | 3 |