### Testes unitários. Junit.

#### Um pouco de XP

XP é um apelido carinhoso de uma metodologia ágil de desenvolvimento designada Extreme Programming, com foco em agilidade de equipes e qualidade de projetos, apoiada em valores como simplicidade, comunicação, feedback e coragem.

XP é uma metodologia muito comportamental, onde prima mudanças de atitudes e práticas. Sua principal mudança está na máxima integração entre pessoas e, principalmente, estimulando uma participação maior do cliente. Portando, literalmente, temos que “FAZER O PROGRAMA COM O CLIENTE”.

**O que são Testes Unitários?**

Imagine por exemplo, se um avião só fosse testado após a conclusão de sua construção, com certeza isso seria um verdadeiro desastre, é nesse ponto que a engenharia aeronáutica é uma boa referência em processos de construções de projetos de software, principalmente em sistemas de missão crítica, pois durante a construção e montagem de um avião todos os seus componentes são testados isoladamente até a exaustão, e depois cada etapa de integração também é devidamente testada e homologada.

O teste unitário, de certa forma se baseia nessa ideia, pois é uma modalidade de testes que se concentra na verificação da menor unidade do projeto de software. É realizado o teste de uma unidade lógica, com uso de dados suficientes para se testar apenas à lógica da unidade em questão.

Em sistemas construídos com uso de linguagens orientadas a objetos, essa unidade pode ser identificada como um método, uma classe ou mesmo um objeto.

**Porque usar Testes Unitários?**

Abaixo temos alguns dos principais fatores que motivam o uso sistemático da prática de testes unitários:

* Previne contra o aparecimento de “BUG’S” oriundos de códigos mal escritos;
* Código testado é mais confiável;
* Permite alterações sem medo (coragem);
* Testa situações de sucesso e de falha;
* Resulta em outras práticas XP como: Código coletivo, refatoração, integração contínua;
* Serve como métrica do projeto (teste ==requisitos);
* Gera e preserva um “conhecimento” sobre as regras de negócios do projeto.

#### Quando fazer Teste Unitário?

No início: Primeiro projetar e escrever as classes de testes, depois as classes com regra de negócios.

Diariamente: É SUGERIDO que seja rodado os testes várias vezes ao dia (é fácil corrigir pequenos problemas do que corrigir um “problemão” somente no final do projeto).

**O que é JUnit?**

Agora sim, depois dessa pequena conceituação acima sobre testes unitários, você verá a partir desse tópico que o **JUnit é um framework que facilita o desenvolvimento e execução de testes unitários em código Java**.

Ele fornece uma completa API (conjunto de classes) para construir os testes e Aplicações gráficas e em modo console para executar os testes criados.

**Por que usar JUnit?**

Os principais motivos que favorecem o uso desse framework são:

* JUnit pode verificar se cada unidade de código funciona da forma esperada;
* Facilita a criação, execução automática de testes e a apresentação dos resultados;
* É Orientado a Objeto;
* É free e pode ser baixado em: www.junit.org

**Como instalar o JUnit?**

Para **usar o JUnit em sua máquina** basta ter em mente essas duas idéias:

* Caso você **não tenha o JUnit instalado**, faça o download do arquivo junit.jar em www.junit.org, após inclua-o no classpath para compilar e rodar os programas de teste.
* Porém o JUnit já vem configurado nas versões recentes de IDE’s como Eclipse, NetBeans, JBuilder, BlueJ e outros.

## **Conclusão**

Testes unitários são tão complicados quanto as unidades a serem testadas, mas acrescentam uma camada de segurança importante à aplicação. Qualquer alteração que gere impacto no funcionamento previsto será percebida rapidamente, dando possibilidade ao desenvolvedor de tomar as ações necessárias: seja corrigir a implementação ou atualizar os testes.