Test-Driven Development

Erick Sasse
Gerente de Tecnologia
Cadena Sistemas

Interface

- O que é TDD
- Como Funciona
- Frameworks de Apoio (xUnit)
- Benefícios
- Perguntas

cadena sistemas

O que é TDD

- Uma técnica muito utilizada em Extreme Programming (XP).
- Pode ser usado sem XP.
- Todas as classes tem testes unitários que são executados regularmente.
- Testes unitários são testes estruturados que executam os métodos das classes e checam os resultados.

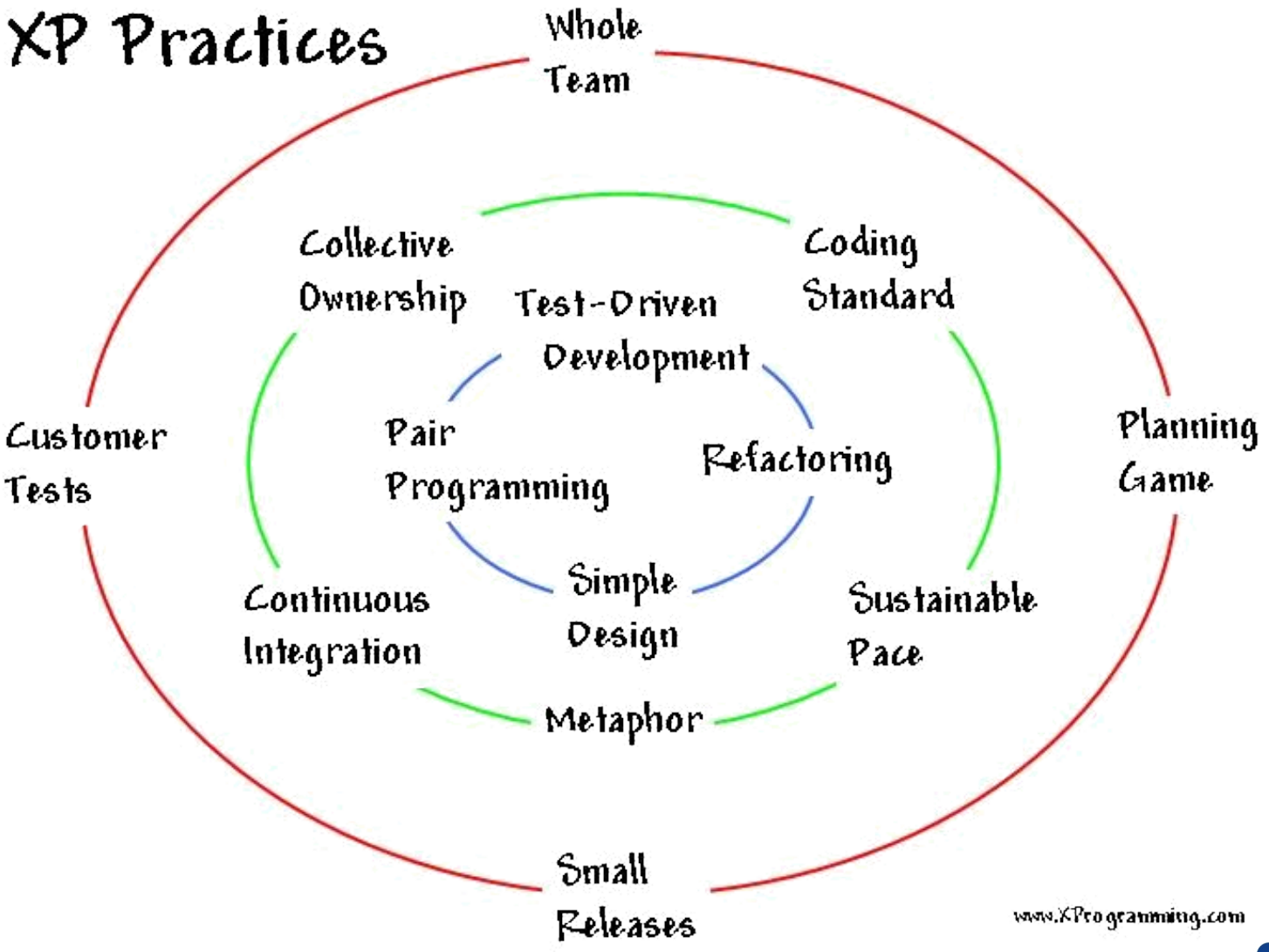
cadena sistemas

Extreme Programming

- Abordagem ágil de desenvolvimento
  - Funciona melhor para grupos pequenos e médios
- Baseada em 4 princípios:
  - Simplicidade
  - Comunicação
  - Feedback
  - Coragem

cadena sistemas

XP Practices



www.XProgramming.com

cadena sistemas

Como funciona?

- Antes de escrever código, pense no que ele fará. Escreva um teste que use os métodos que você ainda nem escreveu.
- Por que escrever teste antes? Desta forma você cobre praticamente 100% do seu código com testes, pois só escreve código que é usado pelos testes.
- Você escreve os testes, e eles são executados repetidamente durante todo o processo de desenvolvimento do aplicativo.

cadena sistemas

O Ciclo do TDD

- Comece com um teste que não compila porque o método que você está testando ainda não existe.
- Escreva o mínimo de código para que ele compile criando o método sem implementação por exemplo.
- Compile o teste. Veja ele falhar.
- Implemente o código.
- Execute o teste. Veja o teste passar.
- Melhore o código (refactoring).
- Execute o teste para certificar-se que tudo continua funcionando.

cadena sistemas

Refactoring

- Refactoring é o processo de alterar e otimizar o código de maneira que seu comportamento externo não seja alterado.
- Em código que está funcionando não se mexe? TDD permite que este medo deixe de ter fundamento, pois ele atua como uma rede de segurança, capturando qualquer bug que seja inserido durante o refactoring.

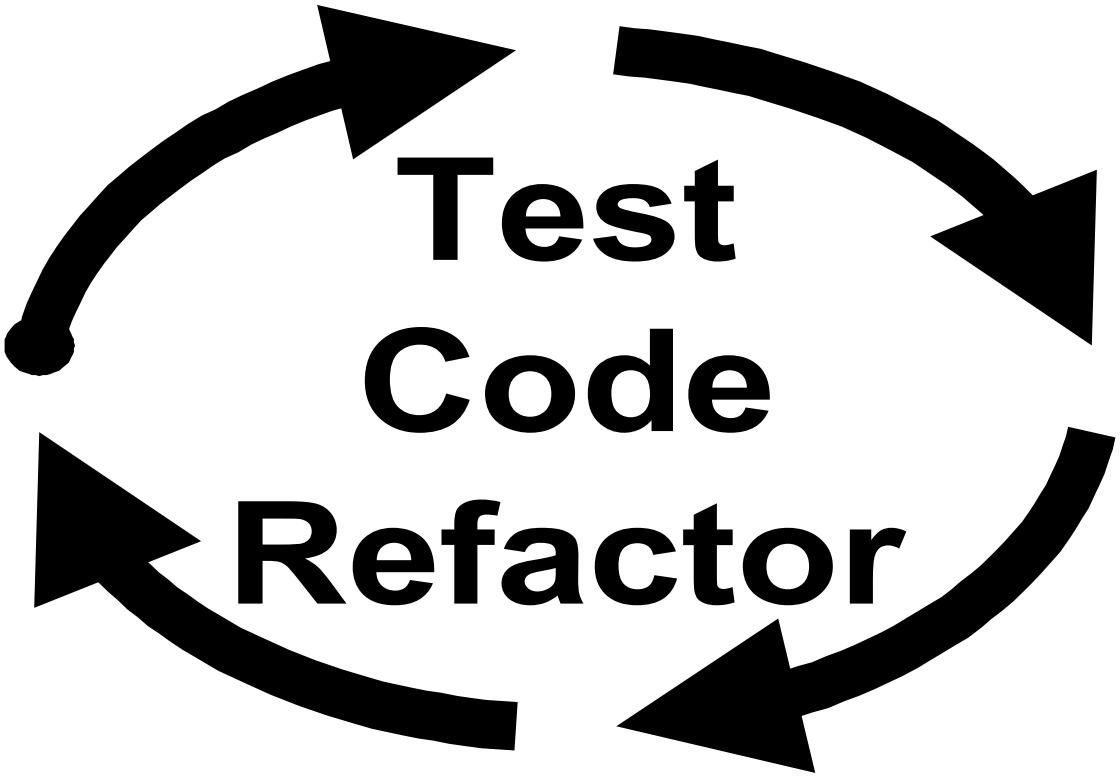
cadena sistemas

Benefícios

- Melhor código
  - Menos bugs (encontrados mais rapidamente)
  - Código mais simples
- Melhor design
  - Código evolui com o tempo ao contrário de degradar como no processo normal.
- Melhor uso de orientação a objetos
  - Você acaba desenvolvendo classes menos acopladas e mais coesas
  - Estimula a separação da lógica de negócios da camada de apresentação
- Menos stress
  - Mais confiança no código
  - Alterações mais simples
- Fundamental para prática de Refactoring

cadena sistemas

O Ciclo do TDD



cadena sistemas

Frameworks de Apoio (xUnit)

- Fornecem classes que facilitam a escrita e execução dos testes:
  - .Net Framework (C#, VB.NET, Delphi.NET)
    - NUnit (www.nunit.org)
    - MbUnit (www.mbunit.org/)
  - JUnit (Java)
  - DUnit (Delphi Win32)
  - xUnit (www.xprogramming.com/software.htm)
- Fáceis de aprender

cadena sistemas

NUnit

- Pode ser usada em qualquer linguagem .NET
- O xUnit mais popular para .Net
- Portada inicialmente a partir do JUnit
- Aproveita vantagens exclusivas da plataforma .Net (atributos, etc)
- Compatível com Mono
- Permite automatizar processo incluindo testes no build gerando logs (inclusive XML).

- www.nunit.org

cadena sistemas

Estrutura de uma classe de testes

```
[TestFixture]
public class MinhaClasseTeste()
{
   [Test]
   public void Teste1();
   { /* ... */ }

   [Test]
   public void Teste2();
   { /* ... */ }
}
```

cadena sistemas

Atributos NUnit

- Obrigatórios
  - [TestFixture]
    - Usado para indicar que uma classe contém métodos de teste
  - [Test]
    - Usado para indicar que o método dentro de um “test fixture” é um teste.

cadena sistemas

Atributos NUnit (cont.)

- Opcionais
  - [TestFixtureSetUp]
    - Método de inicialização. Executado apenas uma vez, antes dos testes começarem.
    - Inicialização de objetos, etc.
  - [TestFixtureTearDown]
    - Método executado apenas uma vez, após todos os testes tenha sido executados.
    - Melhor lugar para liberar objetos criados na [TestFixtureSetUp].
  - Vários outros atributos na documentação.

cadena sistemas

Onde Usar

- Qualquer código que inclua lógica.
- Testar limites do aplicativo (stress)
- Qualquer coisa que possa falhar!
- Muito mais difícil testar GUI, principalmente páginas web.

- Entenda a técnica, pratique um pouco e decida como você pode tirar proveito dela.

cadena sistemas

Melhores Práticas

- Crie um projeto de testes separado dentro de sua solução.
- Separe suas classes em um ou mais class libraries separadas do projeto.
- Não compile o código de teste junto com os binários do projeto projeto.

- MinhaSolucao
  - MeuProjeto
    - MinhaClasse1.cs
    - MinhaClasse2.cs
  - MeuProjetoTestes
    - MinhaClasse1Testes.cs
    - MinhaClasse2Testes.cs

cadena sistemas

Dicas

- TDD está se tornando uma habilidade cada vez mais requisitada (EUA)
- TDD parece difícil, estranho, e pouco intuitivo no começo. Aceite e não desanime.

cadena sistemas

Referência

- Livro: Kent Beck's *Test Driven Development: By Example*
- www.extremeprogramming.org
- www.yahoogroups.com/group/testdrivendevelopment
- www.yahoogroups.com/group/xpers
- www.testdriven.com
- www.xispe.com.br
- Google: TDD, Test-Drive Development, etc.

cadena sistemas

Filosofia Purista

"Any program feature
without an automated test
simply doesn't exist."

-- Kent Beck

cadena sistemas

Obrigado!

- Perguntas?

- erick.sasse@cadena.com.br
- www.cadena.com.br
- www.ericksasse.com.br

- Assert.IsTrue(Voce.VaiTentarTDD);

cadena sistemas