



Auditoria e Qualidade de Sistemas

Prof. Edgard Davidson C. Cardoso



Qualidade de Sistemas

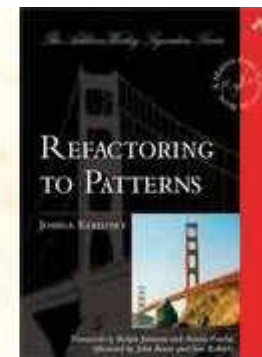
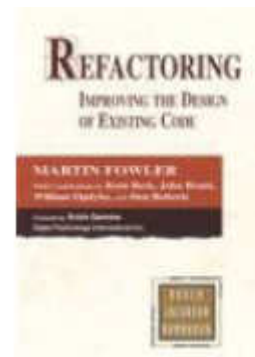
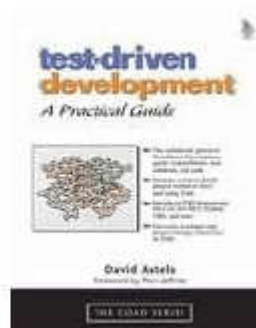
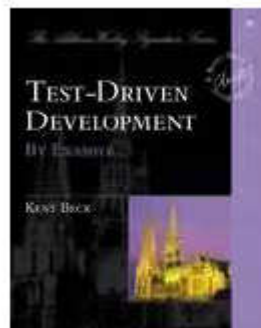
TDD – TEST DRIVEN DEVELOPMENT



Referências

- Livros:

- Kent Beck, “Test-Driven Development: By Example”, Addison-Wesley Professional, 2002
- David Astels, “Test Driven Development: A Practical Guide”, Prentice Hall PTR, 2003
- Martin Fowler et al, “Refactoring: Improving the Design of Existing Code”, Addison-Wesley Professional, 1999
- Joshua Kerievsky, “Refactoring to Patterns”, Addison- Wesley, 2004





O que é Desenvolvimento Dirigido por Testes?

“Código limpo que funciona”

-- Ron Jeffries





TDD- “*Código Limpo que Funciona*”

- É uma forma previsível de desenvolver. Você sabe quando acabou sem ter que se preocupar com uma longa trilha de erros.
- Dá a você a chance de aprender todas as lições que o código tem para ensinar
- Melhora as vidas dos usuários de seu software
- Permite que seus colegas de equipe contem com você, e você com eles.
- É bom escrevê-lo



O que é Desenvolvimento Dirigido por Testes?

“Técnica de desenvolvimento que estimula o uso constante de testes unitários e refatoração”



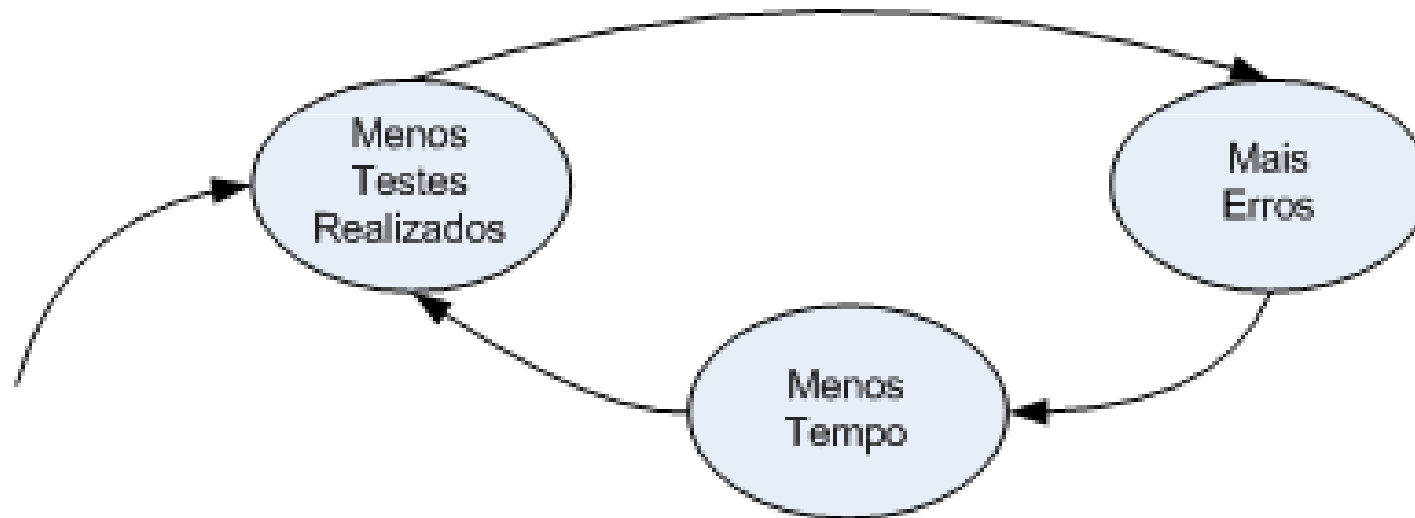
O que é Desenvolvimento Dirigido por Testes?

- **TDD** é uma técnica de desenvolvimento de software cujo processo é formado por pequenas iterações para o desenvolvimento de uma nova funcionalidade, começando pela implementação de um caso de teste, depois pelo código necessário para fazer o teste passar, e finalmente pela refatoração do código visando melhor acomodar as mudanças feitas



“Espiral da Morte” dos teste

- O ciclo mortal do “estou sem tempo para testar”:



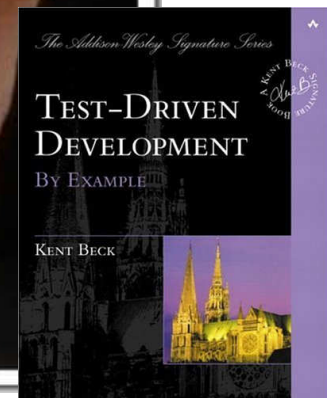
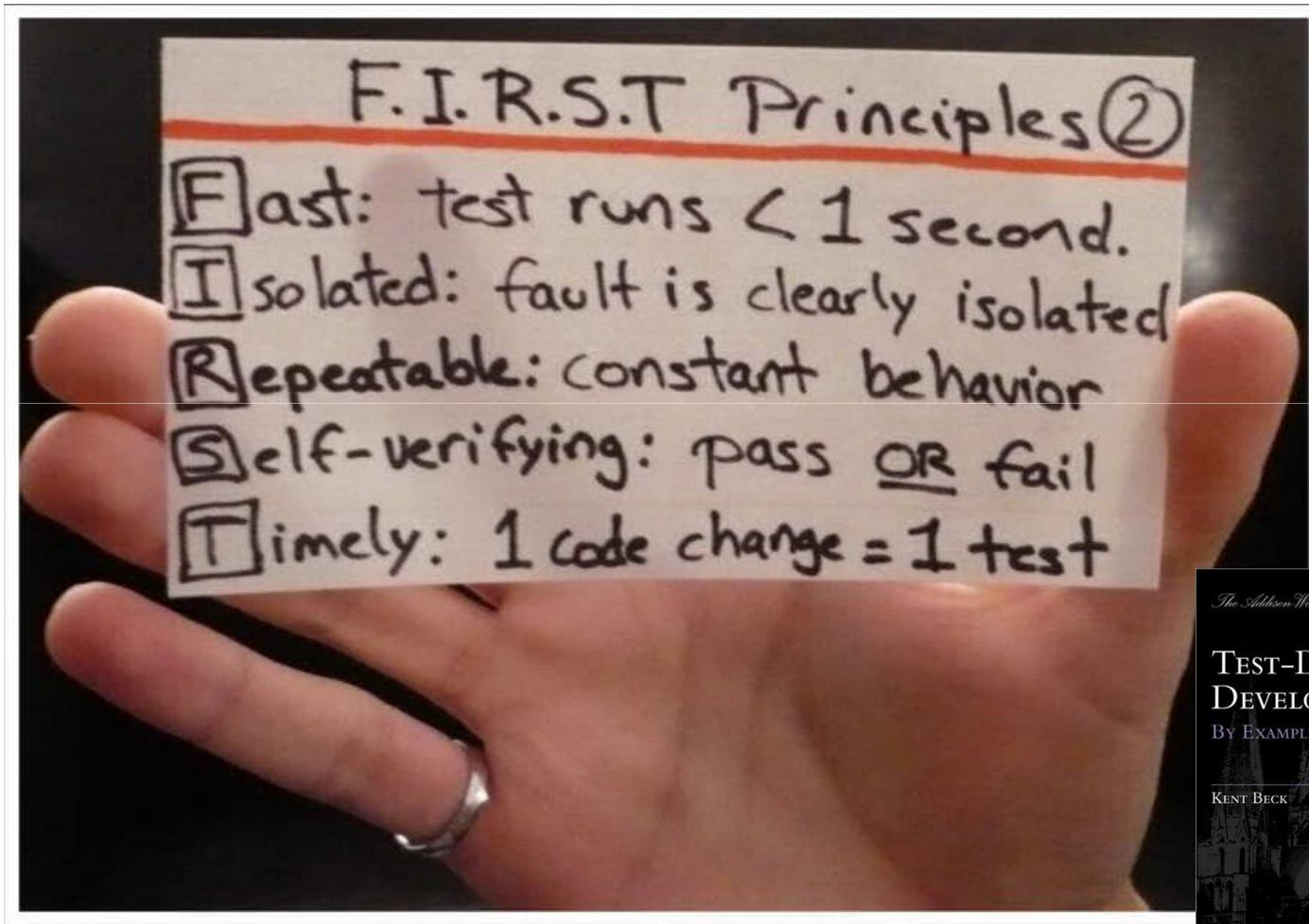


TDD - Quebra de paradigmas!

- O teste é escrito antes do código.
- Escrever mais não é desperdício de tempo, é um investimento.
- A implementação não é tão importante.
- Não é um método para testar software, mas para construir software. **Apesar de testar!**

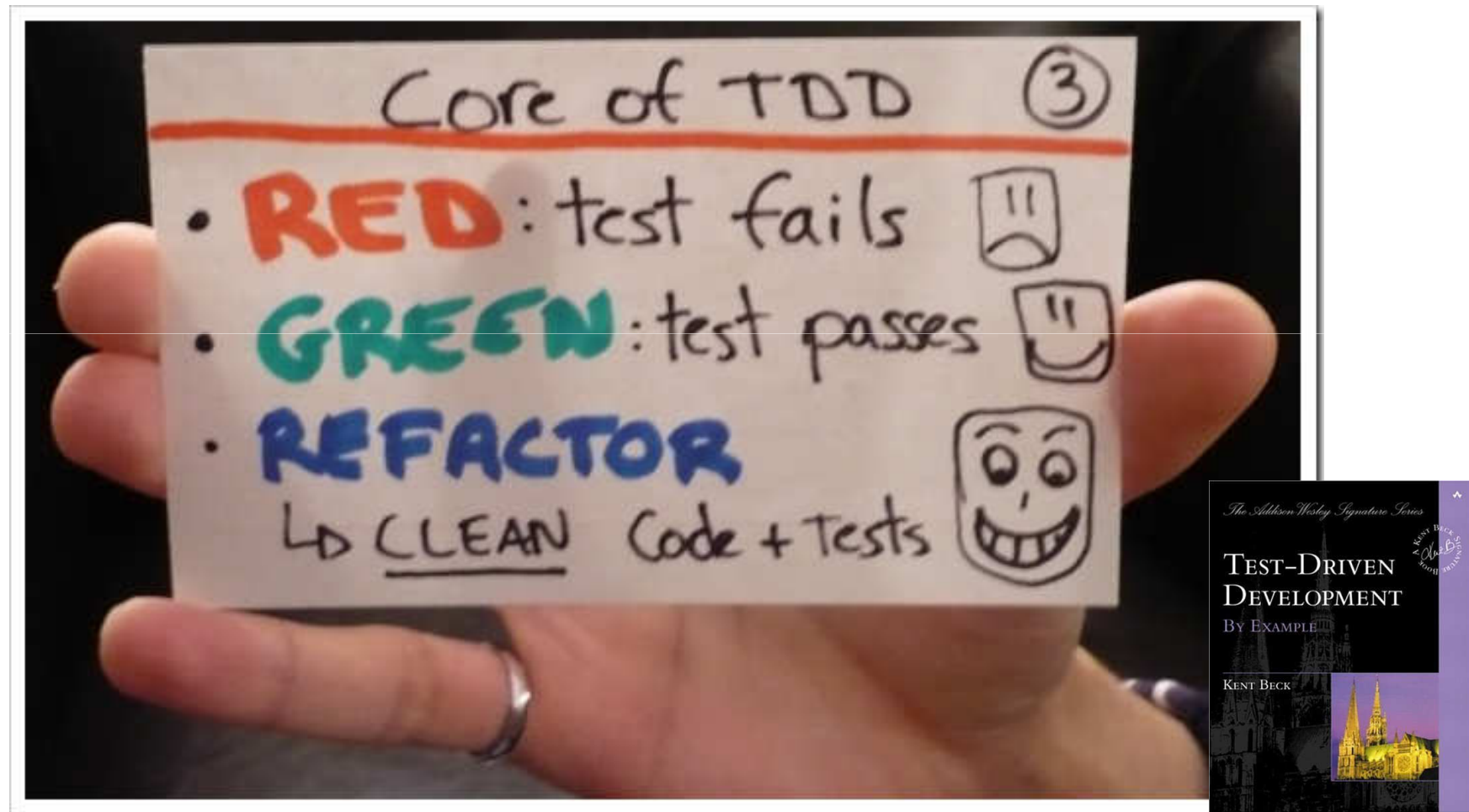


O que é Desenvolvimento Dirigido por Testes?





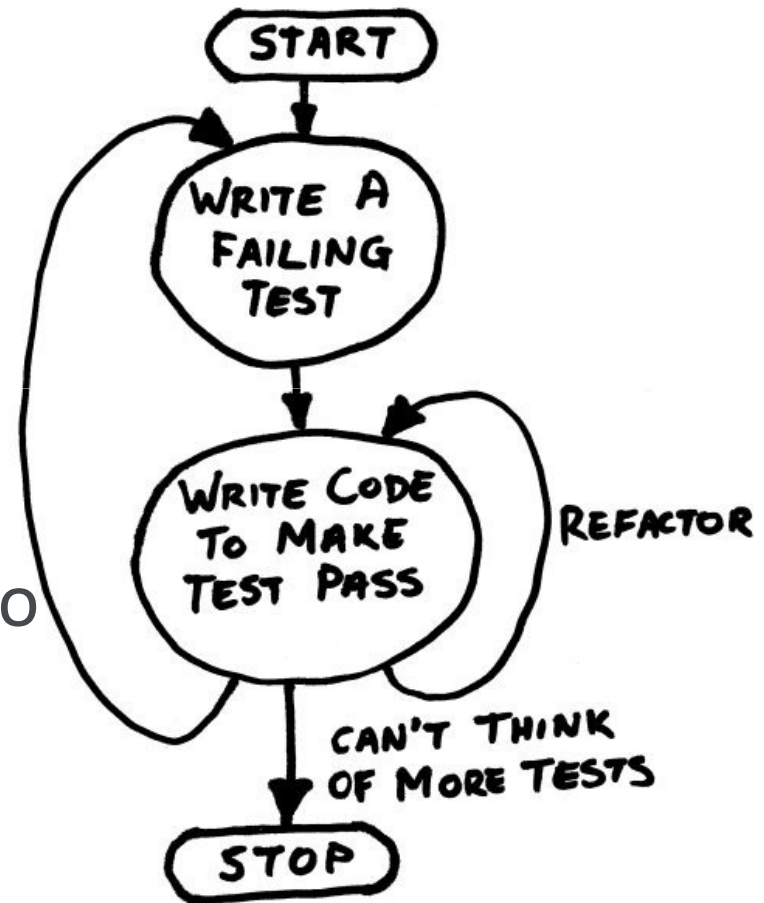
O que é Desenvolvimento Dirigido por Testes?





O que é Desenvolvimento Dirigido por Testes?

- Ciclo em passos pequenos:
 - 1. Escreva um teste que falha
 - 2. Faça o teste passar rapidamente
 - 3. Refatore
- Conseqüência: Testes não são esquecidos devido a falta de tempo, pressão ou estresse, etc





O que é Desenvolvimento Dirigido por Testes?

1. Vermelho – Escrever um pequeno teste que não funcione e que talvez nem mesmo compile inicialmente.
2. Verde – Fazer rapidamente o teste funcionar, mesmo cometendo algum pecado necessário no processo.
3. Refatorar – Eliminar todas as duplicações criadas apenas para que o teste funcione.



Ciclo do TDD





Principais Benefícios do TDD

- Garante a existência de testes unitários completos e atualizados, que:
 - Eliminam o medo de alterarmos alguma coisa que funciona (testada manualmente), e acabarmos introduzindo algum problema;
 - Nos permite utilizar refatoração (substituir uma implementação por outra equivalente) de forma muito mais agressiva devido à facilidade dos testes verificarem o resultado.
- Diminui a quantidade de erros por linha de código (código-fonte de mais qualidade)
- Testes unitários servem como especificação de como os componentes do sistema funcionam;
- Nos leva a produzir componentes de software mais desacoplados, para garantir o isolamento dos testes, o que acaba favorecendo o projeto do sistema.



Mantenha sua casa arrumada. Use TDD





TDD na sua linguagem

- Java -> junit
- .NET -> NUnit
- Python -> unittest
- Ruby -> Test::Unit
- PHP -> SimpleTest
- Flex -> FlexUnit
- Javascript -> jsUnit
 - jQuery -> Qunit
- etc



Exemplo do Taxi

- Especificações:
 - Taxa inicial = R\$3,50
 - Valor por KM = R\$1,00
- Expectativa:
 - Se o taxi rodar 5KM, valor cobrado será de R\$8,50
 - Taxa inicial (R\$3,50) + (5KM * R\$1,00) = R\$8,50



Conclusão

- *Design* evolui junto com o conhecimento
 - *Code for tomorrow, design for today (Kent. Beck)*
- Desenvolvimento com “Passos Pequenos”
- Expressa a intenção do programador com testes
- Servem como documentação



Conclusão

- O Código:
 - Nome dos testes definem o comportamento esperado
 - Fatorados (sem duplicação)
 - Alta cobertura
 - Evita código inútil
 - Alta qualidade do código
 - Refatorações são seguras com testes automatizados