



TDC2012
the developer's conference

Não existe Agile sem Design Ágil

Trilha Agile - TDC GYN 2012

Eder Ignatowicz...



@ederign

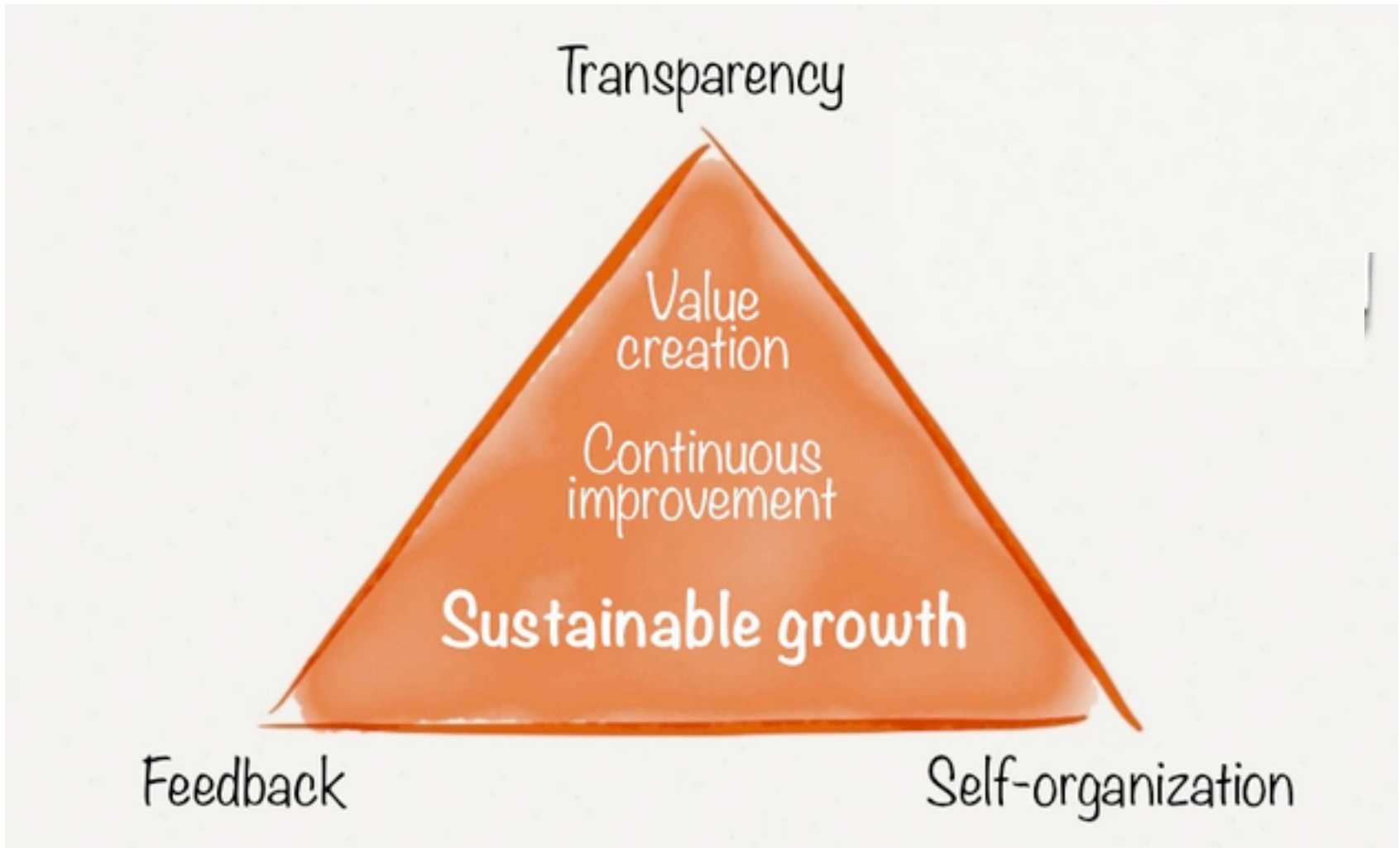
Software Craftsman @ Dextra
(Arquitetura, NoSQL, Devops, QA)

Doutorando na Unicamp
(Polyglot Persistence em Cidades Digitais)

Professor na Faccamp e Unisal

Editor líder no InfoQ Brasil

The Agile Triangle



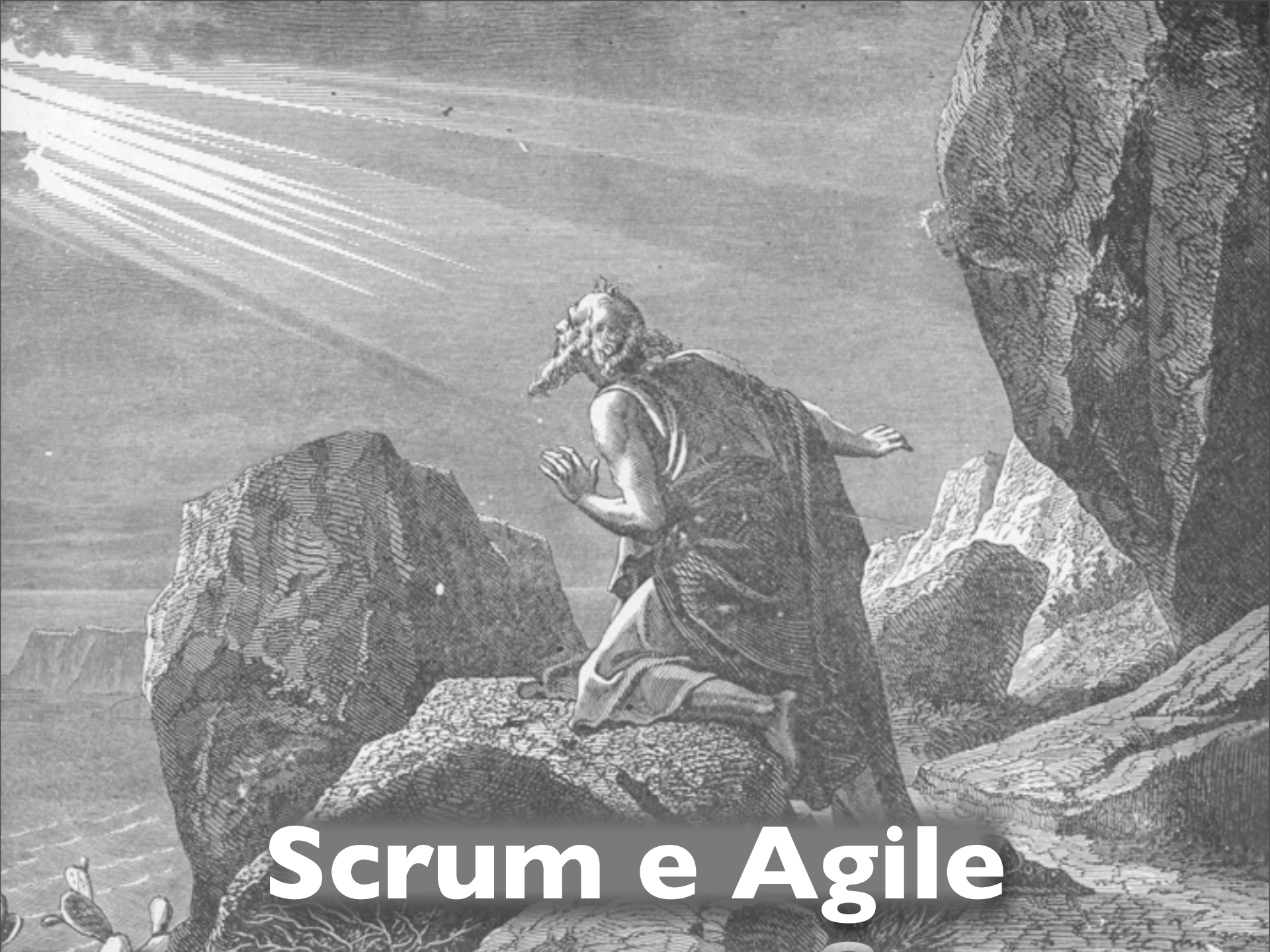
200x - Unicamp/Stefanini



quinta-feira, 18 de outubro de 2012







Scrum e Agile

quinta-feira, 18 de outubro de 2012



Mainstream

Mas eu sou ágil...

***Agile** significa responder **rapidamente** a qualquer mudança...*

*Responder **rapidamente** a oportunidades de negócio*

*Responder **rapidamente** a mudanças de mercado e desafios*

*Agile significa **entrega sustentável***





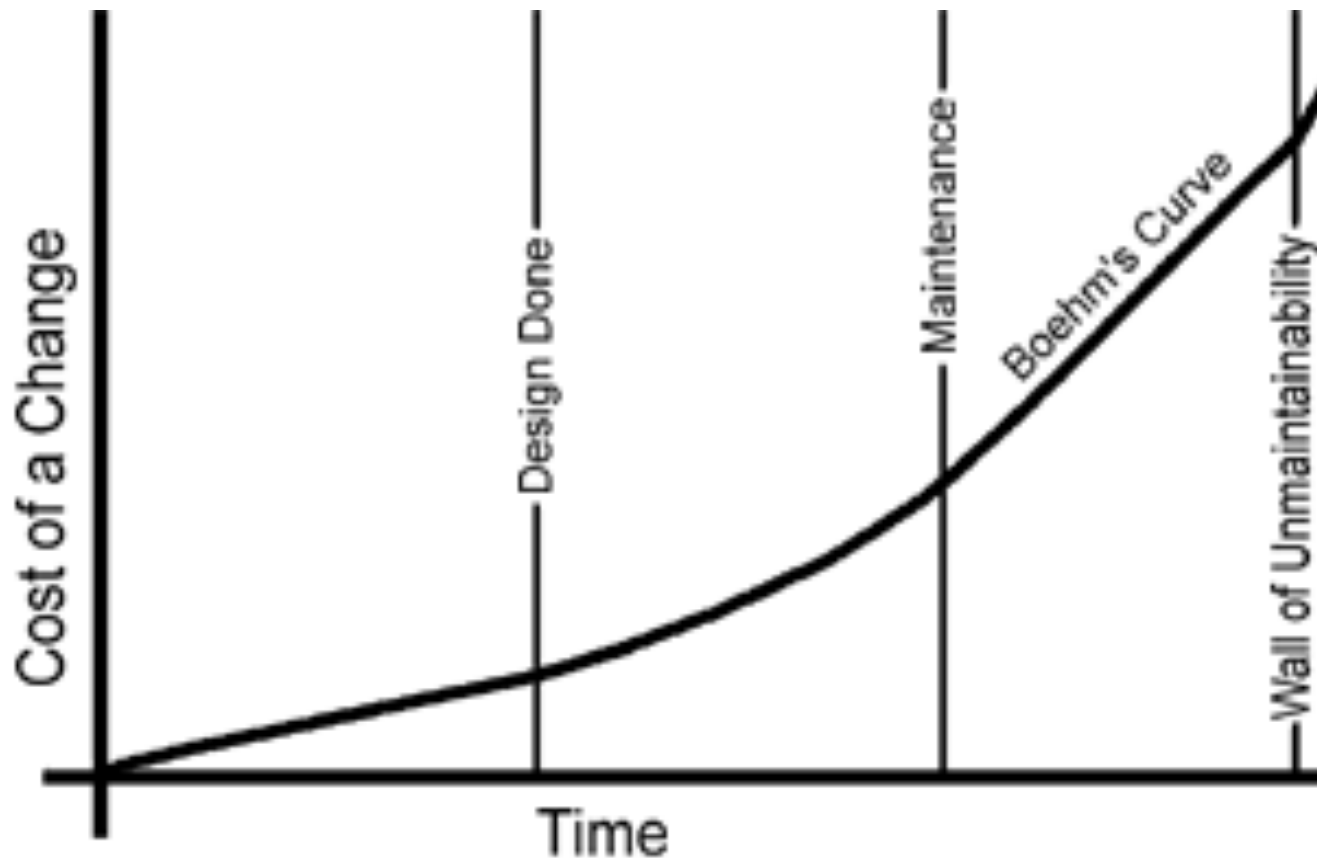
Green Field



Brown Field

quinta-feira, 18 de outubro de 2012

Software apodrece



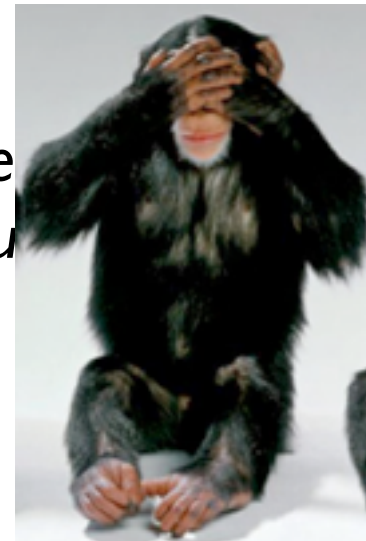
Scrum é o novo Waterfall

Ken Schwaber está correto?

*“Eu estimo que 75% das organizações **utilizando Scrum** não encontraram os benefícios que esperam da metodologia.”*

E Kent Beck?

*“No mundo corporativo, a palavra **Ágil** tornou-se sinônimo de Scrum, que significa, ironicamente, que o processo prevaleceu sobre as pessoas”*



Scrum é o novo Waterfall

As empresas adotando Agile realmente entendem o que é ser ágil?

*Adoção em massa está tornando um agile um **PROCESSO** consistente*

Scrum Masters e Coaches são os novos fiscais do processo

Modelo de negócio construído baseado no Agile

*Process and tools **OVER** individuals*



Onde erramos?

Mitos, Falácias...

(outras palestras desta trilha)...

Falta de entendimento do que realmente é
desenvolvimento e design
de software



Como criar sistemas resistentes a mudanças?

Como criar sistemas capazes de acomodar constantes e contínuas mudanças?

Como prover crescimento sustentável?

Design Ágil

aka Design Sustentável



Software Design

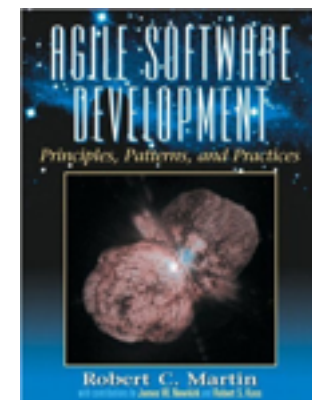
O código fonte é o design

Código abandonado apodrece e fede (smells)

Rigidity - Fragility - Immobility

Viscosity - Opacity

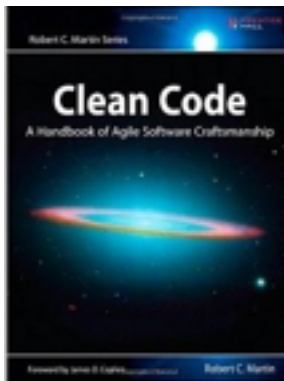
Needless Complexity and Repetition



Software Design

SOLID

Single Responsibility Principle
Open Closed Principle
Liskov Substitution Principle
Interface Segregation Principle
Dependency Inversion Principle



Refactoring

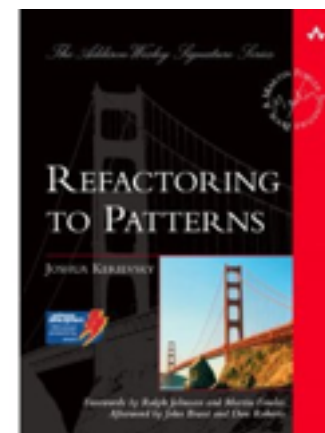
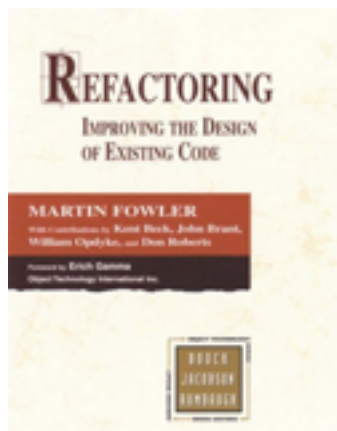
A qualidade do design é em grande parte a soma de pequenas decisões diárias

Lave a louça após cozinhar

Combata a entropia

Testing/Feedback (CI)

Refactoring / Dívida Técnica



E principalmente...



Aumente o sarrafo



**Esqueça o barato, esqueça os
“recursos”**



Contrate os melhores



i ♥ programming

*Contrate os apaixonados, os
craftsman.
Pois só eles cuidarão do seu design*

**SERÁ QUE GRANDE PARTE DO
SUCESSO DO AGILE**



**FOI POR TER ATRAÍDO NO INÍCIO OS MELHORES
DESENVOLVEDORES?**

memegenerator.net



TDC2012

the developer's conference



@ederign