


MAC0475

Alfredo Goldman

IME - USP

06 de Maio de 2021

Métodos Ágeis de desenvolvimento de software: uma abordagem humana



Alfredo Goldman
Professor do IME – USP

Pequeno histórico

- Docente do IME - USP desde 1993
- Começamos uma disciplina:
 - Laboratório de Programação Extrema em 2001
 - Um dos pioneiros no mundo
- De uma forma natural começamos a ensinar no que acreditávamos ser bom

Programação hoje

- Não é mais uma atividade solitária
- Mais de uma pessoa para criar software
- Pessoas vão manter o software
 - Geralmente outras
- Atividade cooperativa
 - Melhor analogia: Alpinismo
- Fator complicador
 - Pessoas não são estáticas

Solução: virou um problema

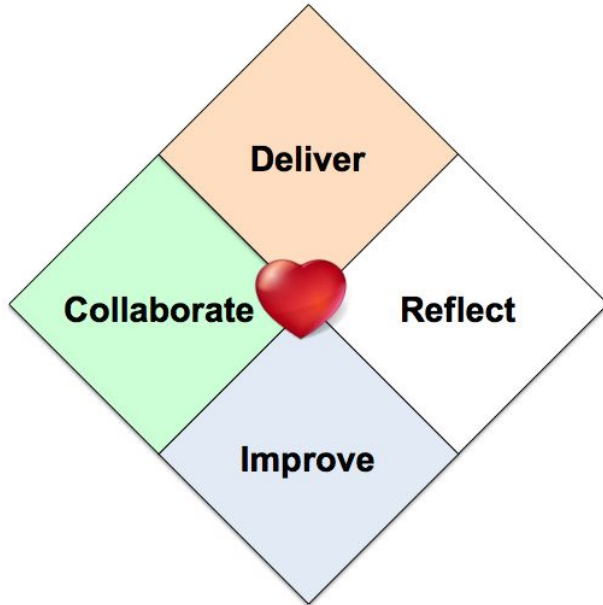
- Métodos ágeis fizeram sucesso
- *Agile virou buzzword*
- Muita gente se diz ágil hoje
 - Mas, infelizmente poucos são ágeis de verdade
 - Hoje, ser ágil é moda

Métodos ágeis NAO são processos ou práticas


- Artigo do Highsmith
 - “Stop doing agile start being agile”
- Não adianta tirar um certificado CSM e achar que basta
- Agilidade é um estado de espírito

O Coração do Ágil

- Um diagrama muito significativo
 - Do Alistair Cockburn



Agora sim: linhas gerais da apresentação

- Motivação para o surgimento de métodos ágeis
 - Isto é: problemas 
- Princípios comuns a métodos ágeis
 - Manifesto ágil
- Programação extrema
 - Mas, existem outros: Crystal, FDD, Lean, Scrum, etc.

Um caso de outra área

Apolo 13

- Acidente em módulo de serviço
- 1970
- *Houston, we have a problem*
- Problema de refrigeração
- Equipe
- Entrosamento
- Improvisação

Columbia (depois da Challenger)

- Tecnologia mais avançada
- 2003
- Ônibus espacial
- 28 missão
- Pequeno dano na asa, na decolagem
- Três pedidos de revisão
- Negados pelos gerentes

Modelo Cascata

Requisitos

Análise

Arquitetura e
Design

Implementação

Testes

Produção

Ainda melhor que o Go Horse!

Wiston Royce, 1970

Analogia incorreta 2

Engenharia de Software \neq Engenharia Civil



Programar é fazer Design!

O que é desenvolvimento de software (A. Cockburn)?

What is / isn't software development?

Model Building	(Jacobson)
Engineering	(Meyer)
Discipline	(Humphreys)
Poetry	(Cockburn)
Math	(Hoare)
Craft	(Knuth)
Art	(Gabriel)

**If you know what it is,
you can apply known solutions.**

manifesto

- Manifesto ágil:
 - **Indivíduos e interações** são mais importantes que processos e ferramentas
 - **Software funcionando** é mais importante que documentação completa e detalhada
 - **Colaboração com o cliente** é mais importante que negociação de contratos
 - **Adaptação a mudanças** é mais importante que seguir um plano

Manifesto Ágil

- Leitura obrigatória
- Não dá para transformar tudo em processo
 - Há um trabalho intelectual/criativo
- Hoje o valor ágil principal para mim é
 - **Melhoria contínua**

Algo mais concreto:

Programação eXtrema

- Metodologia de desenvolvimento de software aperfeiçoada desde 1999
- Ganhou notoriedade a partir da OOPSLA'2000
- Nome principal: Kent Beck

O que é programação extrema?

- Conjunto de práticas a serem adotadas no cotidiano da equipe

As práticas:

- são adaptáveis a diferentes contextos
- se suportam e complementam
- permitem adoção em pequenos passos
- apoiam os valores essenciais por trás do método

Princípios básicos de XP

- *Feedback* rápido
- Simplicidade é o melhor negócio
- Mudança incrementais
- Carregue a bandeira das mudanças / não valorize o medo (*Embrace chance*)
- Alta qualidade do código

Os valores de XP

- Comunicação
- *Feedback*
- Coragem
- Simplicidade
- Respeito

As 12 práticas de XP (versão 2000)

Planejamento

Fases Pequenas

Metáfora

Design Simples

Testes

Refatoração

Programação Pareada

Propriedade Coletiva

Integração Contínua

Semana de 40 horas

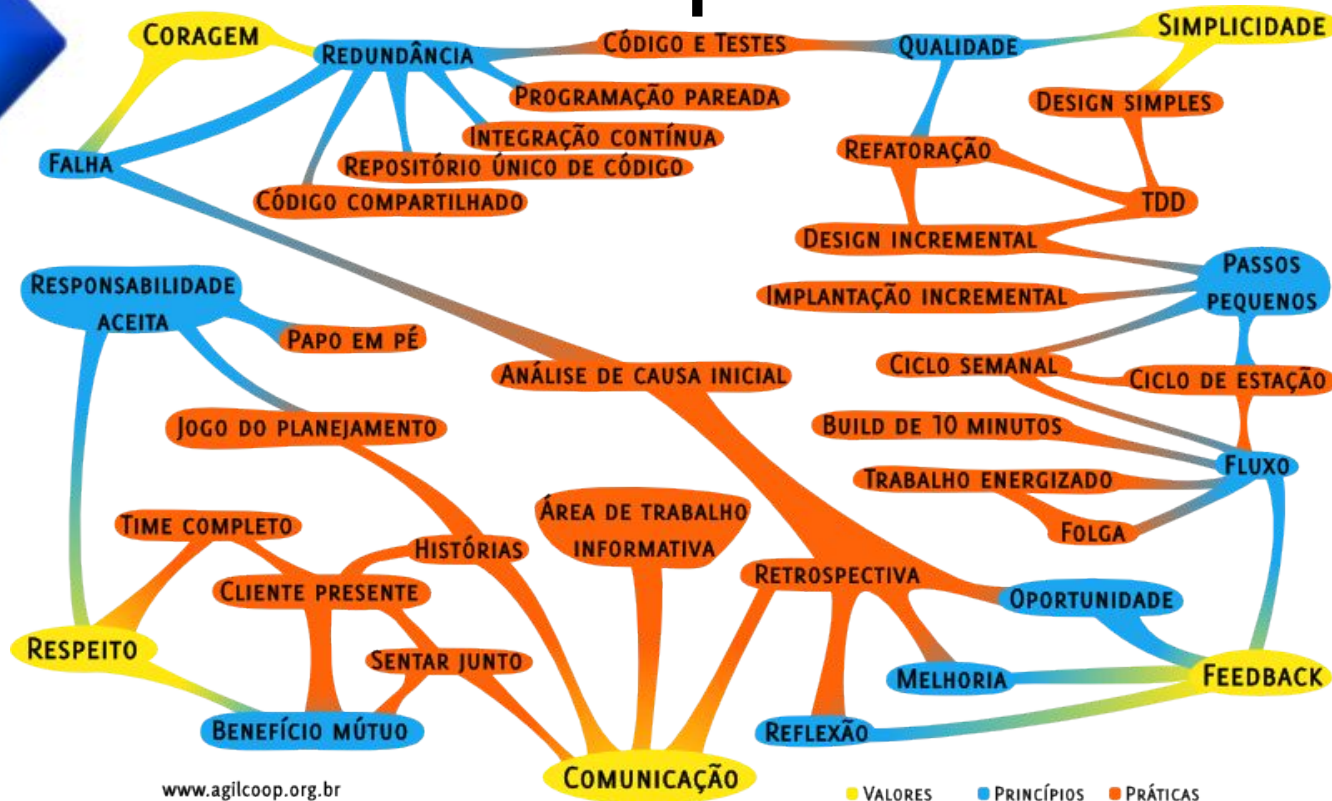
Cliente junto aos
desenvolvedores

Padronização do código

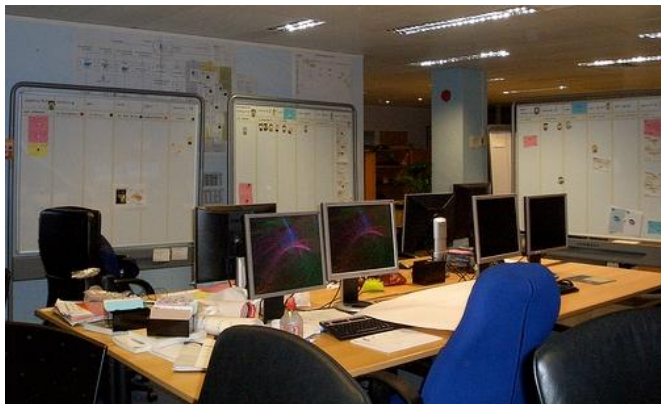
XP também se adaptou

- Novas práticas
 - Principais
 - trazem benefícios imediatos
 - Corolárias
- Adapta-se a diferentes ambientes
- Visão muito mais humana

Resumo: valores, princípios e práticas



Área de trabalho informativa

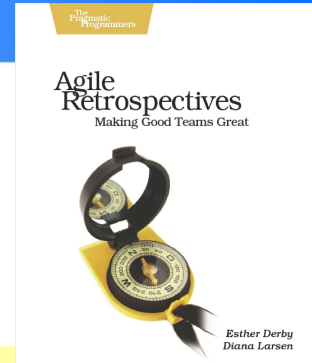
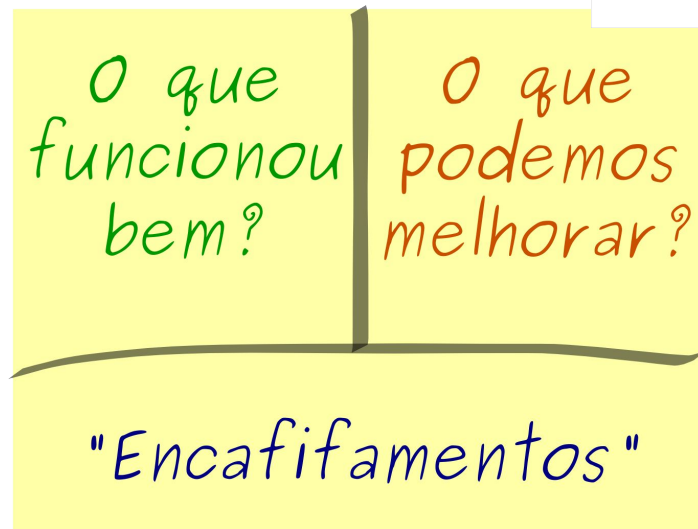


Sem exagerar

- Trabalho energizado:
Balancear a intensidade do trabalho para não desgastar a equipe
- Folga:
Não planeje até o último minuto. Deixe uma folga para imprevistos porque eles sempre surgem

Melhorando sempre

- Análise de causa inicial
- Retrospectiva:
Nada é perfeito.
Tudo sempre pode
melhorar e, para
isso, precisa
entender o que
deu certo e errado



Mais informação

- AgileBrazil
- AgileTrends
 - Participe, discuta e aprenda com os outros
- Agilcoop (deprecated)
- Página da ThoughtWorks
- AgileAlliance

Leia mais

- Livros
 - XP, edição 1 e 2. Kent Beck
 - Refactoring, Martin Fowler
 - TDD, Kent Beck
 - Livros da casa do código
- Livros não técnicos
 - Think Fast Think Slow, Daniel Kahneman
 - Drive, Daniel Pink (tem TED)
 - Management 3.0, Jurgen Appelo



Dúvidas?

Obrigado !

?

www.agilcoop.org.br
www.ime.usp.br/~gold

Agilcoop

- [Agilcast](#)
- [Aprenda mais](#)
- ▶ [Artigos](#)
- [Contato](#)
- [Cursos e Palestras](#)
- [Dissertações](#)
- [Eventos](#)
- [Membros](#)
- [Projetos de Software](#)

área de membros

Usuário *

Senha *

- [Recuperar senha](#)

Entrar

Compartilhamento de Conhecimento entre Equipes Ágeis: Fatores Influenciadores

Enviado por viviane.almeida em ter, 14/08/2012 - 08:34

Após pesquisas preliminares realizadas pelo nosso grupo em empresas ágeis no período de 2010 a 2011 (Disponível em: <http://bit.ly/technical-report-2012>), identificamos que empresas ágeis reconhecem a necessidade de compartilhar conhecimento entre equipes, portanto aplicam diversas práticas com propósito, de baixo custo e focadas em gerar interação entre seus profissionais. Além disto, identificamos que fatores organizacionais influenciam na efetividade deste processo.

Os grupos de pesquisa em métodos ágeis da Universidade de São Paulo e da Universidade Estadual do Ceará convidaram empresas ágeis do ranking das 95 melhores empresas de TI para trabalhar de 2011 da revista ComputerWorld (disponível em: http://computerworld.uol.com.br/gptw/2011/ranking_geral) para responder um questionário sobre a influência das condições organizacionais no processo de compartilhamento de conhecimento entre equipes ágeis.

[Leia mais](#)

Slides de "Uma introdução ao Desenvolvimento de Software Lean"