

# Programação Extrema na Prática

Engenharia de Software Conference - 13:40 - 15:00

maio/09

São Paulo

**AgilCoop** 

Dairton Bassi - [dbassi@gmail.com](mailto:dbassi@gmail.com)

# Assuntos de Hoje

- Métodos Ágeis
  - Valores Ágeis
- Programação Extrema
  - Princípios
  - Práticas
- Cases

# Métodos Tradicionais

1 - Levantamento de Requisitos

2 - Análise de Requisitos

3 - Desenho da Arquitetura

4 - Implementação

5 - Testes

6 - Produção / Manutenção

# Qual é o foco?

- Produção de software não é uma operação mecânica
- Desenvolver software requer análise, compreensão e criatividade

# O que é preciso?

- Flexibilidade
- Colaboração
- Comprometimento
- Disposição
- Sinergia
- Confiança
- Coragem
- Comunicação
- Respeito
- Foco nas Pessoas

# Manifesto Ágil

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

# Vários Métodos Ágeis

- eXtreme Programming (XP)
- Scrum
- Lean
- Crystal Clear
- DSDM
- FDD
- ASD
- EVO

# O que é?

*“XP é sobre mudança social.”*

Kent Beck

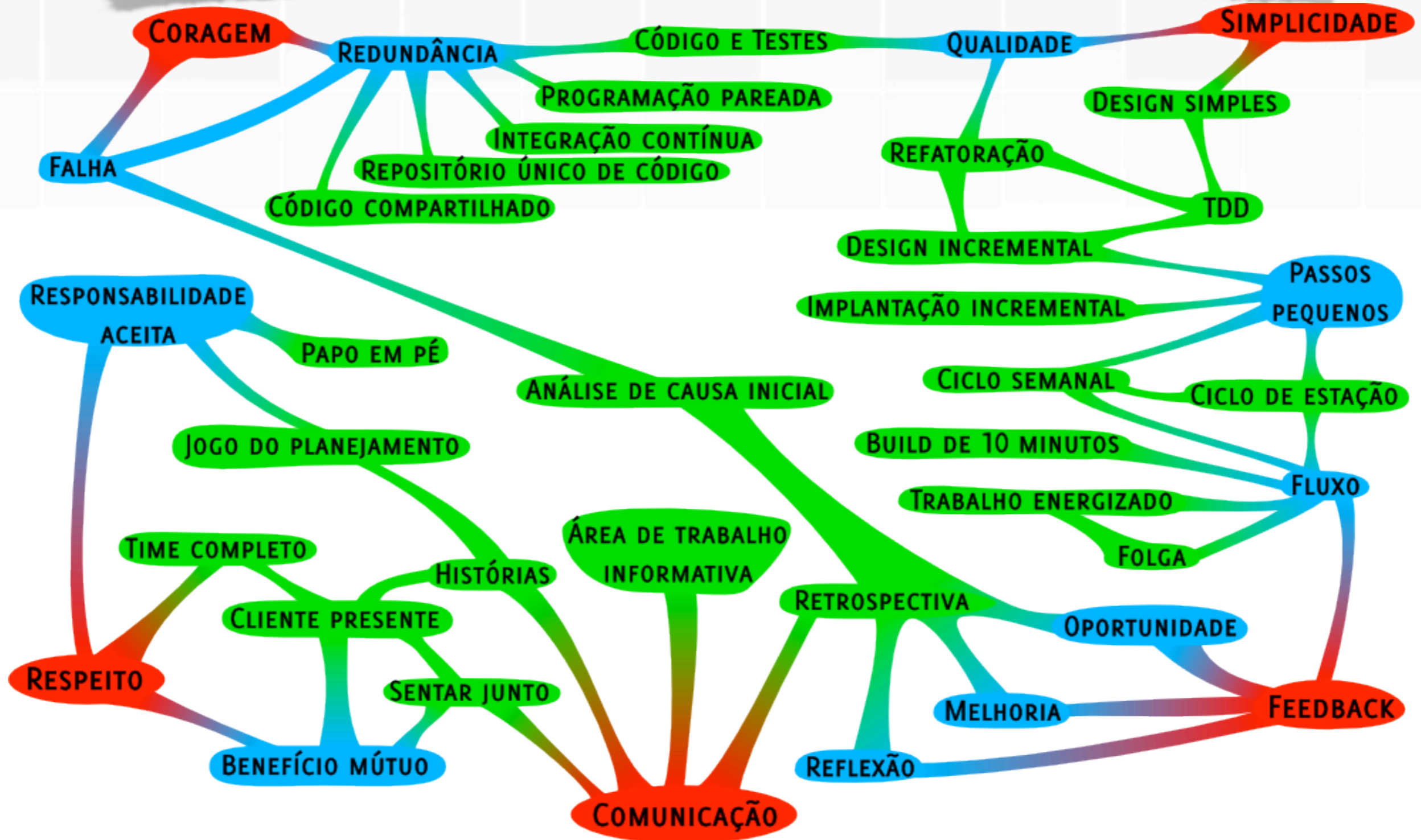
# Valores

- Tenha e forneça feedback rápido
- Simplicidade é o melhor negócio  
**Feedback - Simplicidade**
- Faça mudanças incrementais  
**Coragem - Comunicação**
- Carregue a bandeira das mudanças
- Alta qualidade do código  
**Respeito**
- Respeite as pessoas

# Princípios

- Qualidade
- Fluxo
- Redundância
- Oportunidade
- Falha
- Melhoria
- Benefício mútuo
- Reflexão
- Passos pequenos
- Responsabilidade aceita

# Valores, Princípios e Práticas



# Principais Práticas de XP

- Histórias
- Time completo
- Área de Trabalho
- TDD
- Build Ágil
- Folga
- Programação Pareada
- Integração Contínua
- Trabalho energizado
- Design Incremental
- Sentar junto
- Ciclo semanal e trimestral

# Programação em Pares

- Erros detectados imediatamente
- Diversidade de idéias, técnicas e algoritmos
- Enquanto um escreve, o outro pensa em contra-exemplos, problemas de eficiência etc.
- Vergonha de escrever código feio (gambiarras)
- Pareamento de acordo com especialidades

# O Ambiente



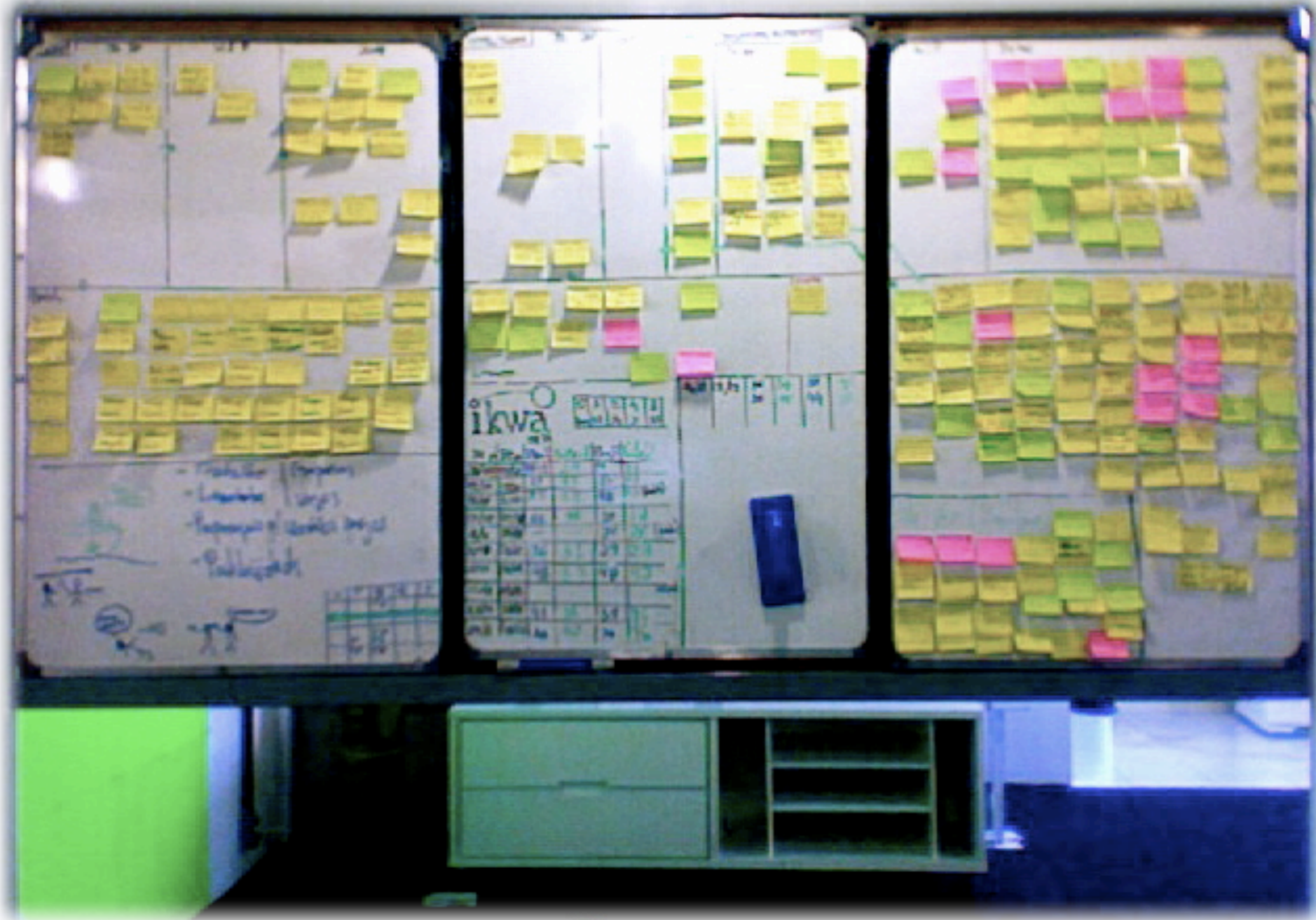
# Testes

- Fundamental para o sucesso de XP
- Dá segurança e coragem ao grupo
- Testes de unidades
  - escritos pelos programadores
- Testes de funcionalidades
  - escritos pelos clientes

# Um mural de histórias



# Um quadro mais complexo



# Cliente

- Responsável por escrever “histórias”
- Pode ser representado por um programador
- Trabalha no mesmo espaço físico do grupo
- Avalia as novas funcionalidades entregues
- Dá Feedback rapidamente
- Solicita ou aprova mudanças

# Coach

- Um desenvolvedor experiente
- Identifica as habilidades da equipe
- Lembra das regras do jogo (XP)
- Eventualmente faz programação em pares
- Não desenha arquitetura
- Seu papel diminui com o tempo

# Tracker

- Coleta estatísticas e as exibe
- Mantém histórico do progresso
- Alguns exemplos:
  - Número de histórias implementadas
  - Número de testes
  - Número de classes e linhas de código

# Refatorações

- Não altera a funcionalidade
- Melhora alguma qualidade não-funcional:
  - simplicidade
  - flexibilidade
  - clareza
  - desempenho\*

# Exemplos de Refatorações

- Nomes de variáveis
  - `valtotcont => valor_total_da_conta`
- Mudanças arquiteturais
- Encapsular código repetido
- Generalização de métodos
  - `raizQuadrada(float x) + raizCubica(float x)`  
 $\Rightarrow$  `raiz(float x, int n)`

# O Código

- Padrões de estilo adotados por todo o grupo
- O mais claro possível
- XP não é baseada em documentação
- Comentários sempre que necessários
- Comentários padronizados

# Propriedade coletiva

- Se eu perceber uma oportunidade para simplificar, consertar ou melhorar o código escrito por outra pessoa, peço autorização verbalmente ou por escrito?

# E as Reuniões?

- Reunião da versão
- Reunião da iteração
- Stand up meeting

# Um dia de um programador XP

- Escolhe uma história do cliente
- Procura um par livre
- Escolhe um computador para programar
- Discute as modificações recentes no sistema
- Discute a história com o parceiro

# Um dia de um programador XP

- Repensa a arquitetura
- Pensa em novos testes
- Faz a implementação e os testes
- Executa os testes
- Integra o novo código ao repositório

# Atividades Frequentes

- Executa testes antigos
- Busca oportunidades para simplificação
- Modifica a implementação incrementalmente
- Escreve novos testes
- Faz todos os testes passarem
- Integra novo código ao repositório

# Case 1: Grande Órgão Público

- Apoio da gerência
- Conhecimento técnico heterogêneo
- Alguns estavam animados
- Alguns estavam apreensivos
- Projeto complicado e problemático
- A diretoria queria resultados

# Abordagem

- Equipe de consultores em Métodos Ágeis
- Selecionamos a metodologia (XP), as tecnologias e as ferramentas
- Implantação gradual e em etapas
- Revesamento dos coaches
- \*Bons resultados fizeram o projeto crescer\*



# Ambiente Energizado



Prazo para a 1ª versão em produção: 6 meses  
Realizado: 5 meses

# Para fazer XP

- Supere o medo de mudanças
- Escreva código
- Confie no trabalho da sua equipe
- Escreva testes
- Negocie e busque a colaboração do cliente
- Entregue software com frequência

# Como começar?

- Repense suas práticas
- No início, se dizia: *“Ou você é 100% eXtremo ou não é eXtremo.”*
- Implemente mudanças gradualmente
- Converse com as pessoas. Se preciso, peça ajuda.



Dairton Bassi  
dbassi@gmail.com

