

DCC / ICEx / UFMG

## Programação Extrema (XP)

Eduardo Figueiredo  
<http://www.dcc.ufmg.br/~figueiredo>

## Programação Extrema (XP)

- Proposta a partir de boas práticas de desenvolvimento incremental
- Propõe o envolvimento do cliente ao extremo
  - O cliente (ou seu representante) deve estar disponível durante todo o desenvolvimento
- Programadores trabalham em pares

## Dos Requisitos aos Testes

- Os requisitos são escritos como cenários (estórias do usuário)
  - Estas estórias são implementadas diretamente por um conjunto de tarefas
- Para cada tarefa, é desenvolvido um conjunto de testes
  - Testes são feitos antes da implementação

## Modelo de Processo XP

- O espaço de tempo entre releases é curto

```

graph TD
    A[Selecionar histórias do usuário] --> B[Dividir histórias em tarefas]
    B --> C[Planejar a versão e os testes]
    C --> D[Desenvolver, integrar e testar]
    D --> E[Liberar software]
    E --> F[Avaliar o sistema]
    F --> A
  
```

## Práticas de XP

- Planejamento incremental
- Pequenas releases
- Projeto simples
- Desenvolvimento dirigido por testes
- Refatoração
  - Reestruturação constante de código

## Práticas de XP

- Programação por pares
- Propriedade coletiva do código
  - Qualquer um pode modificar qualquer coisa (não há ilha de conhecimento)
- Integração contínua
- Ritmo sustentável
- Cliente sempre disponível

## Pequenas Releases

- Novas versões do sistema podem ser compiladas várias vezes por dia
  - Testes unitários automatizados devem ser executados após cada compilação
- Incrementos são entregues ao cliente a cada duas semanas

## Refatorações

- XP prega que modelar o sistema para mudanças futuras é um esforço inútil
- Refatorações são constantemente aplicadas para permitir adaptações

## Testes em XP

- XP enfatiza mais as atividades de teste que outros métodos ágeis
- Práticas
  - Desenvolver primeiro os testes
  - Codificação incremental a partir dos testes
  - Envolvimento do usuário na escrita e validação dos testes
  - Uso de ferramentas para testes automatizados
- Testes de aceitação também são incrementais

## Desvantagens de Testar Primeiro

- Programadores preferem programar do que testar
  - Testes podem ser mal feitos ou incompletos
- Alguns testes são difíceis de escrever
  - Pode ser tão difícil quando implementar
- É difícil avaliar a abrangência dos testes

## Programação por Pares (PP)

- Dois programadores sentam juntos na frente de um mesmo computador
- Os pares nem sempre são os mesmos
  - A alocação dinâmica dos pares é sugerido, pois favorece a propriedade coletiva do código

## Vantagens de PP

- Responsabilidade comum
  - As acertos e falhas são de responsabilidade de toda a equipe
- Processo informal de revisão
  - Enquanto um programa, o parceiro revisa informalmente o código
- Favorece a melhoria da qualidade
  - Os parceiros discutem oportunidades para refatorações

### [ Bibliografia Principal ]

- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
  - Cap. 3 (Seção 3.3)

### [ Bibliografia Adicional (XP) ]

- Kent Beck and Cynthia Andres. **Extreme Programming Explained: Embrace Change**, 2nd Edition. Addison-Wesley Professional, 2004.