



(XP) e**X**treme_**P**rogramming

- Mauricio de lima
- Murillo de Sousa Sales
- Olair Soares de Almeida
- Pablo Henrique Caldas Silva
- Pedro Henrique Juliano Cardoso

O que é Extreme Programming ?



Extreme Programming (XP) é uma metodologia de desenvolvimento de software, nascida nos Estados Unidos ao final da década de 90. Vem fazendo sucesso em diversos países, por ajudar a criar sistemas de melhor qualidade, que são produzidos em menos tempo e de forma mais econômica que o habitual.

São métodos e formas de priorizar quais as partes mais importantes do Projeto, e quem diz quais as partes importantes são os próprios clientes.

De forma informal, podemos compará-lo como uma dança de salão:

- Desenvolvedor é a dama;
- Cliente é o cavalheiro;
- Devem estar sempre juntos;
- Cliente conduz o desenvolvimento.



Características Extreme Programming

O XP possui algumas características marcantes que são:

- Feedback constante.
- Abordagem incremental.
- Encoraja a comunicação entre as pessoas envolvidas.



Características Extreme Programming

- Simplicidade
- Comunicação
- Coragem
- Respeito
- Feedback



Vantagens

- Um dos benefícios é tornar o processo ágil e mais flexível.
- Fornecem mais valor do que cada uma poderia fornecer individualmente.
- Análise prévia dos acontecimentos dentro do projeto.
- O XP é ideal para projetos em que o cliente não sabe exatamente o que quer.
- Entregas constantes de partes funcionais do software.
- O cliente está sempre por perto.
- Programação em dupla.
- Erros são encontrados em um estágio inicial.



Funcionamento

Análise

Design

Implementação

Teste

Implantação

Manutenção



Exemplos Práticos

Jogo de Planejamento (*Planning Game*): O desenvolvimento é feito em interações semanais.

Fases pequenas (*Small Releases*): A liberação de pequenas versões funcionais do projeto.

Metáfora (*Metaphor*): Procura facilitar a comunicação com o cliente.

Testes de Aceitação (*Customer Tests*): São testes construídos pelo cliente e conjunto de analistas e testadores

Propriedade Coletiva (*Collective Ownership*): O código fonte não tem dono e ninguém precisa solicitar permissão para poder modificar o mesmo.

Programação Pareada (*Pair Programming*): é a programação em par/dupla num único computador.

Padronização do Código (*Coding Standards*): A equipe de desenvolvimento precisa estabelecer regras para programar e todos devem seguir estas regras.

Desenvolvimento Orientado a Testes (*Test Driven Development*): Primeiro crie os testes unitários (*unit tests*) e depois crie o código para que os testes funcionem.

Integração Contínua (*Continuous Integration*): Sempre que produzir uma nova funcionalidade, nunca esperar uma semana para integrar à versão atual do sistema.



Obrigado a todos.