

XP - Extreme Programming

Prof: Nilson Júnior
nilson.junior@jaboatao.ifpe.edu.br

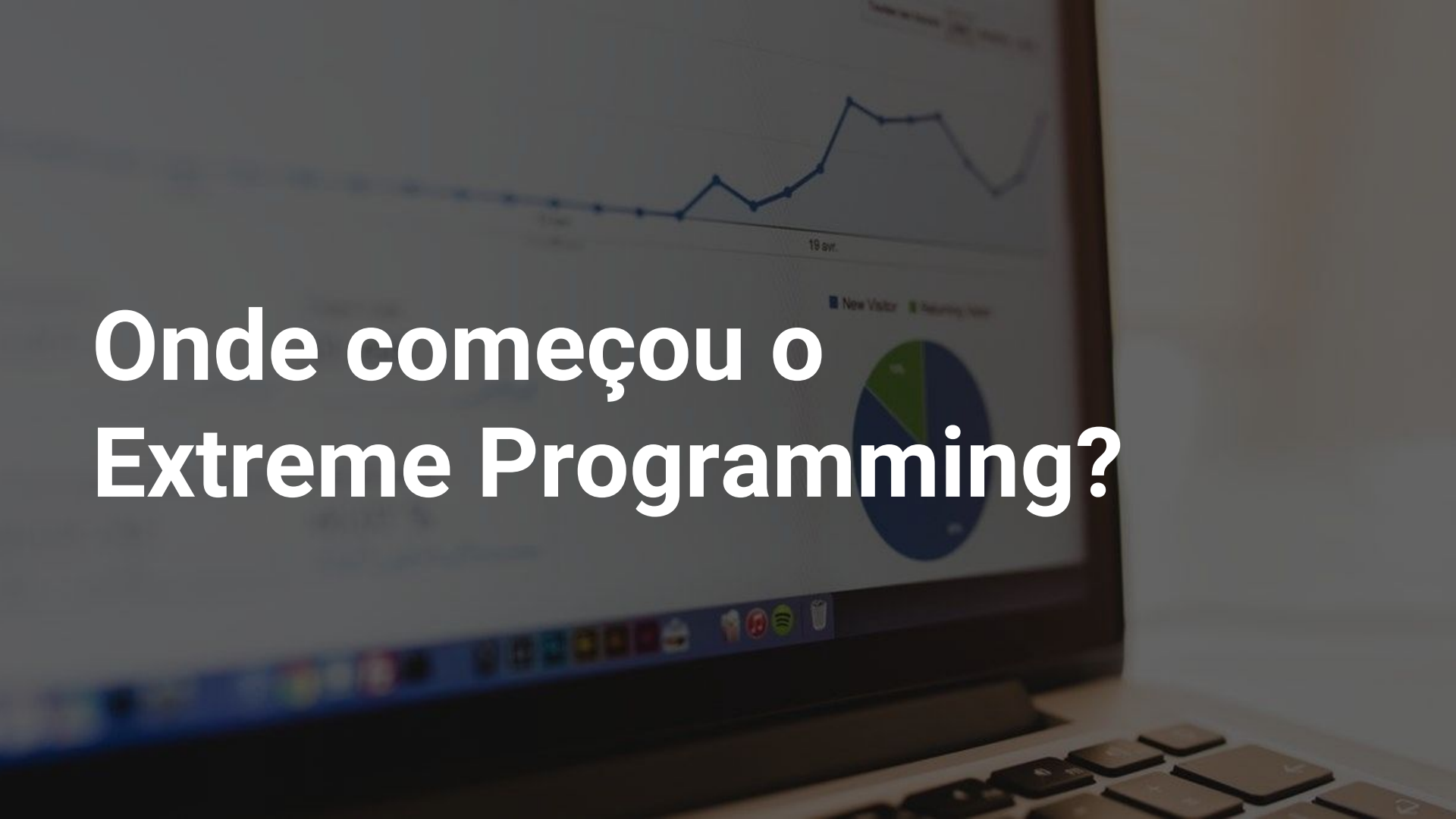
www.professornilson.wix.com/ifpe



Agenda

- História do Extreme Programming
- Princípios do Extreme Programming
- Práticas do Extreme Programming

Onde começou o Extreme Programming?

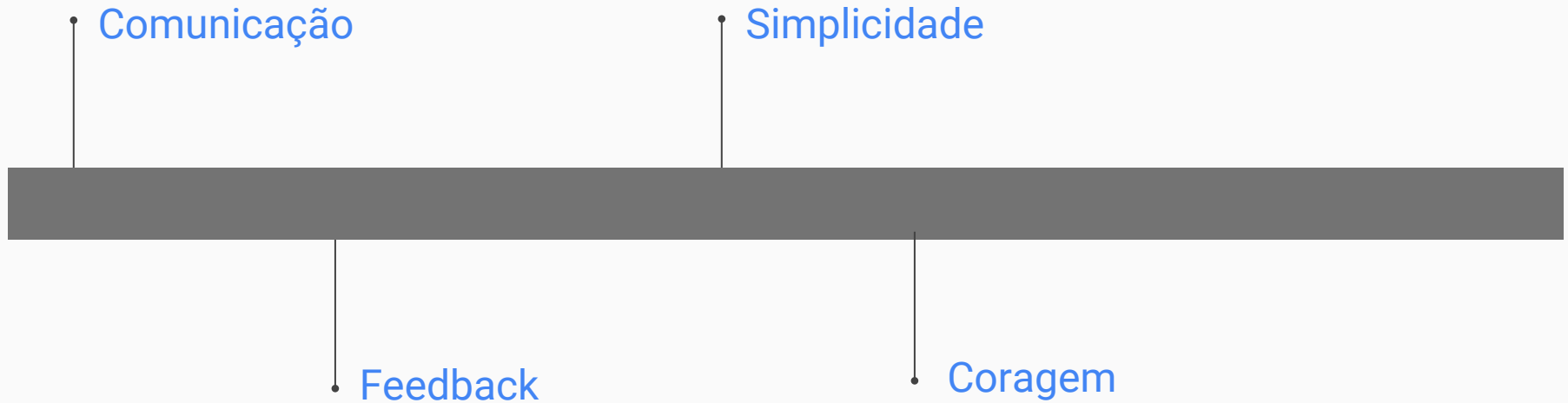


Extreme Programming - XP

O Extreme Programming é um modelo de desenvolvimento de software ágil, criado em 1996, por Kent Beck, no Departamento de Computação da montadora de carros Daimler Chrysler.

O XP possui muitas diferenças em relação a outros modelos, podendo ser aplicado a projetos de alto risco e com requisitos dinâmicos.

Dimensões do XP



Características



Características

- Equipe de 2 a 10 pessoas
- Desenvolvimento guiado por testes
- Equipe Proativa
- Cliente sempre disponível

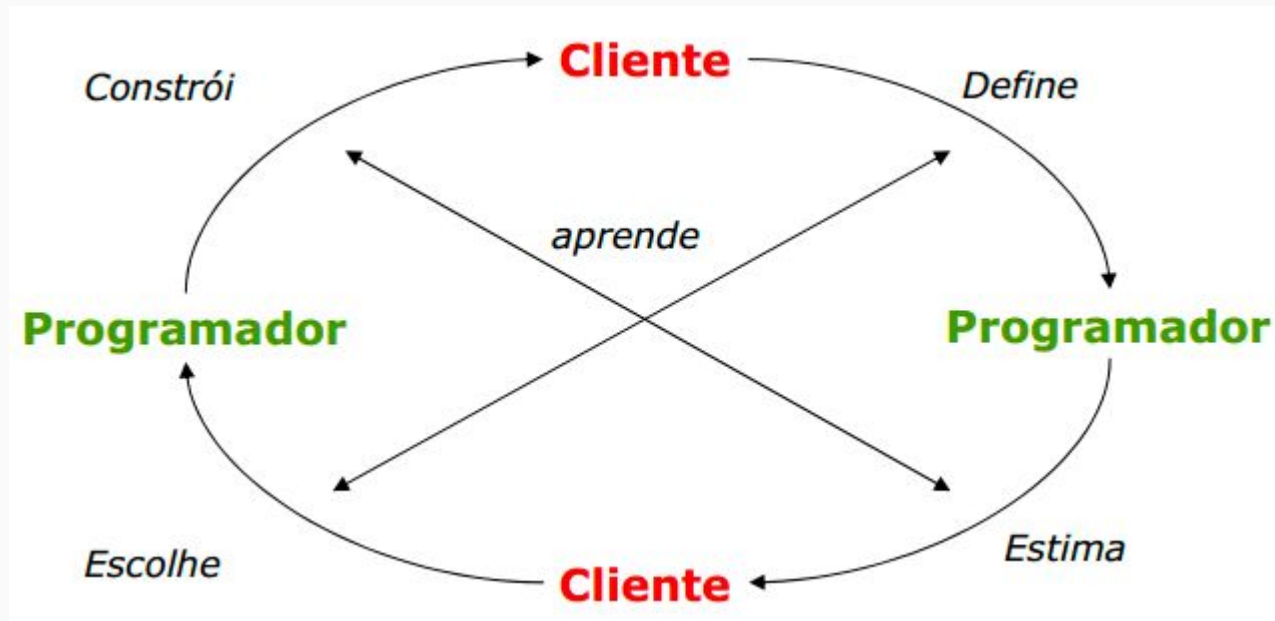
Benefícios

- Desenvolvimento de software ágil.
- Agilidade no planejamento (Planning game).
- Não foca em uma especificação completa e formal dos requisitos.
- Comunicação intensa.
- Equipe Integrada

Fases do XP

The background features a dark green gradient. In the center, there are three interlocking gears of different sizes. Two human figures, rendered in a light green, stylized, almost transparent manner, are positioned on the left and right sides, appearing to push or support the large gears. The overall aesthetic is industrial and collaborative.





Práticas do XP

The background features a dark teal gradient. On the left side, there is a faint, stylized illustration of a clipboard with a checklist. The checklist has several items, some of which are marked with checkmarks. A pen is positioned at the bottom right of the clipboard. On the right side of the image, the words "BEST PRACTICES" are written in a large, bold, sans-serif font, with "BEST" on the top line and "PRACTICES" on the bottom line. The text is a lighter shade of teal, blending into the background.

BEST
PRACTICES

Práticas do XP

- Jogo de **Planeamento** (Planning Game)
- **Pequenas** Versões (Small Releases)
- Projeto **Simple** (Simple Design)
- Time **Coeso** (Whole Team)
- Testes de **Aceitação** (Customer Tests)
- Ritmo **Sustentável** (Sustainable Pace)
- Reuniões em **pé** (Stand-up Meeting)
- Posse **Coletiva** (Collective Ownership)
- Programação em **Pares** (Pair Programming)
- **Padrões** de Codificação (Coding Standards)
- Desenvolvimento Orientado a **Testes** (Test Driven Development)
- **Refatoração** (Refactoring)
- Integração **Contínua** (Continuous Integration)

Jogo do planejamento (Planning Game)



O desenvolvimento é feito em iterações semanais. No início da semana, desenvolvedores e cliente reúnem-se para priorizar as funcionalidades. Essa reunião recebe o nome de Jogo do Planejamento. Nela, o cliente identifica prioridades e os desenvolvedores as estimam. O cliente é essencial neste processo e, assim, ele fica sabendo o que está acontecendo e o que vai acontecer no projeto.

Pequenas versões (Small releases)



A liberação de pequenas versões funcionais do projeto auxilia muito no processo de aceitação por parte do cliente, que já pode testar uma parte do sistema que está comprando. As versões chegam a ser ainda menores que as produzidas por outras metodologias incrementais, como o RUP.

Metáfora (Metaphor)



a duck out of water

Procurar facilitar a comunicação com o cliente, entendendo a realidade dele. O conceito de rápido para um cliente de um sistema jurídico é diferente para um programador experiente em controlar comunicação em sistemas de tempo real, como controle de tráfego aéreo. É preciso traduzir as palavras do cliente para o significado que ele espera dentro do projeto.

Time Coeso (Whole Team)



A equipe de desenvolvimento é formada pelo cliente e pela equipe de desenvolvimento.

Coesão está, na verdade, ligado ao princípio da responsabilidade única, que foi introduzido por Robert C. Martin no início dos anos 2000 e diz que uma classe deve ter apenas uma única responsabilidade e realizá-la de maneira satisfatória.

Testes de Aceitação (Customer Tests)

São testes construídos pelo cliente em conjunto com analistas e testadores, para aceitar um determinado requisito do sistema.



Ritmo Sustentável (Sustainable Pace)



Trabalhar com qualidade, buscando ter ritmo de trabalho saudável (40 horas/semana, 8 horas/dia), sem horas extras. Horas extras são permitidas quando trouxerem produtividade para a execução do projeto.

Reuniões em pé (Stand-up Meeting)



Reuniões em pé para não se perder o foco nos assuntos de modo a efetuar reuniões rápidas, apenas abordando tarefas realizadas e tarefas a realizar pela equipe.

Posse Coletiva (Collective Ownership)



O código fonte não tem dono e ninguém precisa ter permissão concedida para poder modificar o mesmo. O objetivo com isto é fazer a equipe conhecer todas as partes do sistema.

Programação em Pares (Pair Programming)



É a programação em par/dupla num único computador. Geralmente a dupla é criada com alguém sendo iniciado na linguagem e a outra pessoa funcionando como um instrutor. Como é apenas um computador, o novato é que fica à frente fazendo a codificação, e o instrutor acompanha ajudando a desenvolver suas habilidades. Dessa forma o programa sempre é revisto por duas pessoas, evitando e diminuindo assim a possibilidade de erros (bugs). Com isto, procura-se sempre a evolução da equipe, melhorando a qualidade do código fonte gerado.

Padrões de Codificação (Coding Standards)

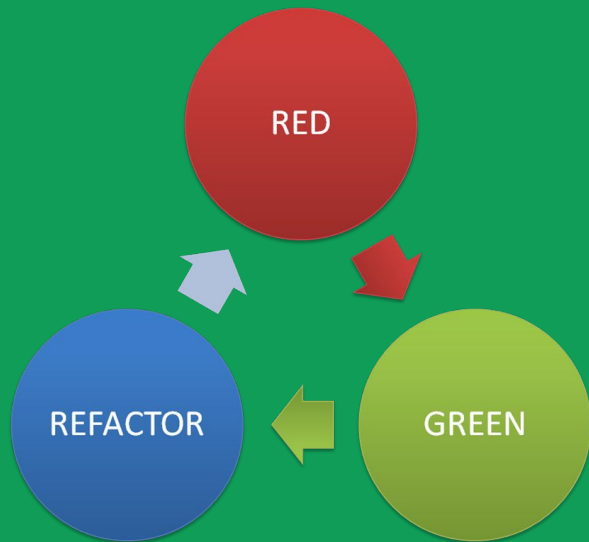
A equipe de desenvolvimento precisa estabelecer regras para programar e todos devem seguir estas regras. Dessa forma parecerá que todo o código fonte foi editado pela mesma pessoa, mesmo quando a equipe possui 10 ou 100 membros.



```
namespace SilverlightApplication
{
    public partial class MainPage : UserControl
    {
        public MainPage()
        {
            InitializeComponent();

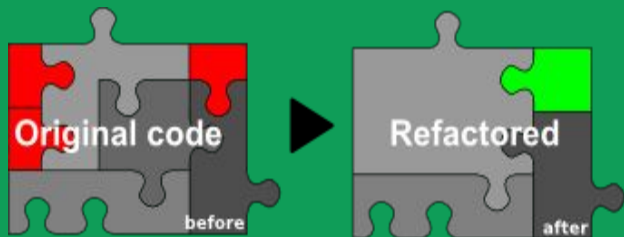
            IValidator validator = new Validator();
            IRule rule = new Rule();
            bool result = validator.Validate(rule);
        }
    }
}
```

Desenvolvimento Orientado a Testes (Test Driven Development)



Primeiro crie os testes unitários (unit tests) e depois crie o código para que os testes funcionem. Esta abordagem é complexa no início, pois vai contra o processo de desenvolvimento de muitos anos. Só que os testes unitários são essenciais para que a qualidade do projeto seja mantida.

Refatoração (Refactoring)



É um processo que permite a melhoria contínua da programação, com o mínimo de introdução de erros e mantendo a compatibilidade com o código já existente. Refatorar melhora a clareza (leitura) do código, divide em módulos mais coesos e de maior reaproveitamento, evitando a duplicação de código-fonte.

Integração Contínua (Continuous Integration)

Sempre que realizar uma nova funcionalidade, nunca esperar uma semana para integrar na versão atual do sistema. Isto só aumenta a possibilidade de conflitos e a possibilidade de erros no código fonte. Integrar de forma contínua permite saber o status real da programação.



A close-up photograph of a hand sorting through a large, thick stack of papers. The papers are slightly aged and have a textured appearance. The hand is visible in the upper right corner, with fingers gently separating the sheets. The overall lighting is soft, and the colors are muted, giving the image a vintage or archival feel.

Papéis

Coach

- Diferente de um arquiteto ou gerente
- Basicamente, um desenvolvedor experiente
- Ajuda novos desenvolvedores
- Interage com gerentes

Tracker

- Medir e comunicar o progresso da equipe
- Feito através de métricas coletadas regularmente

Métodos ágeis costumam ter uma forte ênfase no monitoramento.

- Feito através de métricas coletadas regularmente
- Coleta com overhead mínimo
- Comparação regular entre tempo estimado para uma tarefa e o tempo real necessário
- Big Visible Charts

Processo Iterativo

- Requisitos de sistema SEMPRE evoluem no curso de um projeto
- Algum retrabalho é necessário
- A abordagem iterativa pode ser aplicada a qualquer um dos modelos genéricos do processo.

Duas abordagens (relacionadas)

- Entrega incremental;
- Desenvolvimento espiral.



Dúvidas ?