



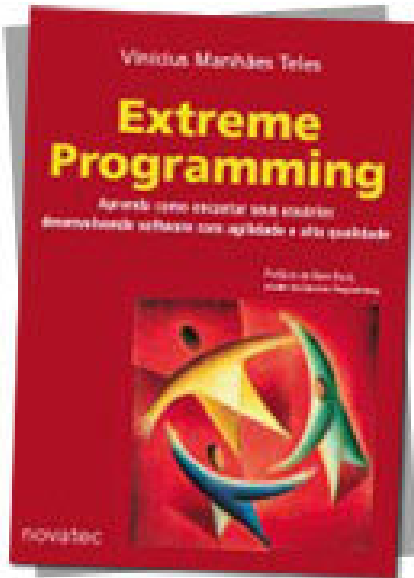
Fundamentos de Engenharia de Software

Prof. Anderson Belgamo

XP

(*Extreme Programming*)

Slides baseados em...



Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade

Autor: **Vinícius Manhães Teles**

Editora: **Novatec Editora**

Páginas: 320;

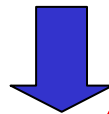
Ano: 2004;

ISBN: 85-7522-047-0;



Origem

- Década de 1990:
 - Novos processos de desenvolvimento que fossem mais adequados para lidar com aspectos humanos dos projetos de software.
- Fevereiro de 2001:
 - Propostas alternativas para evitar os processos de desenvolvimento excessivamente baseados em documentações e formalismos.



Desenvolvimento Ágil de Software



Manifesto Ágil de Software

- Valores

- **Indivíduos e interações** ao invés de processos e ferramentas;
- **Software funcionando** ao invés de documentação abrangente;
- **Colaboração com o cliente** ao invés de negociação de contratos e
- **Responder a mudanças** ao invés de seguir um plano.

<http://www.agilemanifesto.org>.



Manifesto Ágil de Software

- **Indivíduos e interações** ao invés de processos e ferramentas;
- **Software funcionando** ao invés de documentação abrangente;
- **Colaboração com o cliente** ao invés de negociação de contratos e
- **Responder a mudanças** ao invés de seguir um plano.

A proposta é que, embora exista valor nos itens à direita (sem negrito), os processos ágeis valorizam mais os itens que estão à esquerda (em negrito).



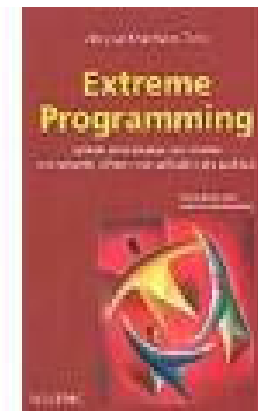
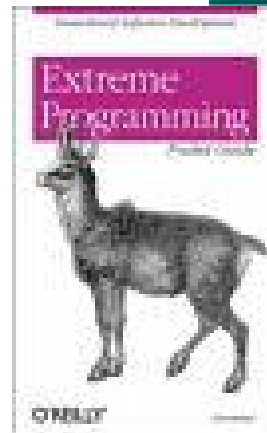
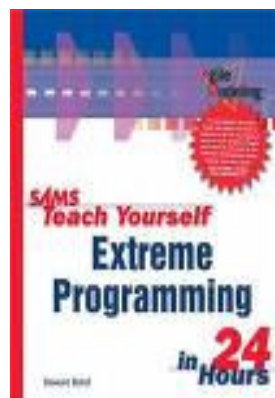
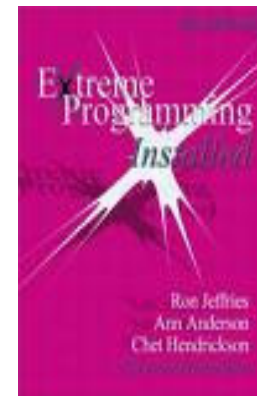
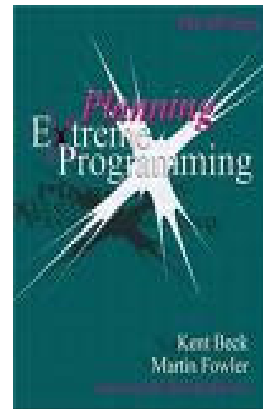
Manifesto Ágil de Software

- Principal diferença: conceito chamado de *barely sufficient*, ou seja, **mínimo necessário**.
 - processos ágeis sugerem o uso de um conjunto bastante reduzido de práticas.
- Processos ágeis existentes:

Scrum, Dynamic Systems Development Method (DSDM), Crystal Methods, Feature-Driven Development (FDD), Lean Development (LD), Extreme Programming e Adaptative Software Development

Extreme Programming (XP)

- Precursores: Kent Beck e Ward Cunningham





Valores do XP

- Feedback
- Comunicação
- Simplicidade
- Coragem



Valores do XP

- Feedback

- A compreensão das necessidades dos usuários é uma das atividades mais difíceis e importantes de serem realizadas pelos desenvolvedores de um software, pois ela direciona todos os demais esforços.
- Assim, um dos é obter feedback, interpretá-lo, e colocar o que foi aprendido de volta dentro do sistema o mais rapidamente possível.



Valores do XP

- Feedback

- Assim, o XP é organizado em ciclos curtos de feedback, o que procura assegurar:
 - Que pouco trabalho seja efetuado e concluído de cada vez.
 - A equipe segue adiante apenas se o resultado estiver correto.
 - Caso surjam falhas, as mesmas são corrigidas logo, antes de iniciar o desenvolvimento de outras funcionalidades, reduzindo as chances de eventuais falhas.



Valores do XP

- Comunicação
 - No desenvolvimento de software existe, no mínimo, um usuário e um desenvolvedor.
 - Com frequência, equívocos no processo de comunicação, causam desentendimentos ou compreensão incorreta de algum aspecto do projeto.
 - XP prega o uso de mecanismos de comunicação mais ricos (face-a-face)



Valores do XP

- Comunicação
 - Projetos XP envolvem ativamente seus usuários fazendo com que se tornem parte integrante da equipe de desenvolvimento.
 - Na prática, significa que o usuário está presente no local de desenvolvimento.



Valores do XP

- Simplicidade
 - 45% das funcionalidades jamais são usadas
 - 19% raramente são utilizadas
 - Isso equivale a 64% que poderiam nem sequer ser implementadas (estudos do Standish Group).



Valores do XP

- Simplicidade

- Em diversos projetos observa-se a ocorrência de três fenômenos que ajudam a esclarecer as razões dos números apresentados:
 - O escopo é fixado no início e alterações no mesmo são evitadas;
 - Os desenvolvedores criam soluções genéricas para “facilitar” possíveis alterações que possam ocorrer no escopo e
 - Os desenvolvedores produzem funcionalidades adicionais na tentativa de antecipar o que o usuário “certamente” irá solicitar no futuro.



Valores do XP

- Simplicidade

- O XP prega que as iterações sejam curtas, com um pequeno escopo de funcionalidades no início de cada iteração e possibilidade de implementá-las completamente dentro de um curto prazo de tempo.
 - Como consequência cria-se a necessidade de concentrar esforços apenas no essencial para implementar as funcionalidades da iteração, evitando generalizações que ainda não se mostrem necessárias e a criação de funcionalidades que ainda não tiverem sido solicitadas pelos usuários



Valores do XP

■ Coragem

■ Medos que exercem influência significativa nos processos de desenvolvimento:

■ Clientes temem:

- Não obter o que pediram;
- Pedir a coisa errada;
- Pagar demais por muito pouco;
- Jamais ver um plano relevante;
- Não saber o que está acontecendo e
- Fixarem-se em suas primeiras decisões e não serem capazes de reagir a mudanças nos negócios.



Valores do XP

■ Coragem

- Ter coragem em XP significa ter **confiança** nos mecanismos de segurança utilizados para proteger o projeto.
 - Ao invés de acreditar que os problemas não ocorrerão e fazer com que a coragem se fundamente nesta crença, projetos XP partem do princípio de que problemas irão ocorrer, inclusive aqueles mais temidos.
 - Entretanto, a equipe utiliza redes de proteção que possam ajudar a reduzir ou eliminar as conseqüências destes problemas.



Valores do XP

■ Coragem

- Medos que exercem influência significativa nos processos de desenvolvimento:
 - Desenvolvedores, por sua vez, temem:
 - Ser solicitados a fazer mais do que sabem fazer;
 - Ser ordenados a fazer coisas que não façam sentido;
 - Ficar defasados tecnicamente;
 - Receber responsabilidades, sem autoridade;
 - Não receber definições claras sobre o que precisa ser feito;
 - Sacrificar a qualidade em função de prazo;
 - Ter que resolver problemas complicados sem ajuda e
 - Não ter tempo suficiente para fazer um bom trabalho.



Práticas do XP

- | | |
|---------------------------------------|-------------------------|
| 1) Cliente Presente | 9) Refatoração |
| 2) Jogo do Planejamento | 10) Integração Contínua |
| 3) Stand up meeting | 11) Releases Curtos |
| 4) Programação em Par | 12) Metáfora |
| 5) Código Coletivo | 13) Ritmo Sustentável |
| 6) Código Padronizado | |
| 7) Design Simples | |
| 8) Desenvolvimento Orientado a Testes | |



Práticas do XP

- Cliente Presente

- A presença do cliente ao longo do desenvolvimento viabiliza o ciclo contínuo de feedback entre ele e os desenvolvedores.
- Este ciclo permite que pequenas mudanças sejam feitas ao longo do desenvolvimento, de forma rápida.



Práticas do XP

- Jogo do Planejamento
 - O XP considera o planejamento uma atividade contínua a ser desempenhada ao longo de todo o projeto.
 - O maior valor não está nos planos gerados, mas sim no exercício de criá-los.
 - Projetos XP procuram dividir o tempo disponível para o projeto utilizando dois conceitos: **releases** e **iterações**.



Práticas do XP

- Jogo do Planejamento

- O XP tem releases que tomam alguns poucos meses, que se dividem em iterações de duas semanas, que se dividem em tarefas que tomam alguns dias.

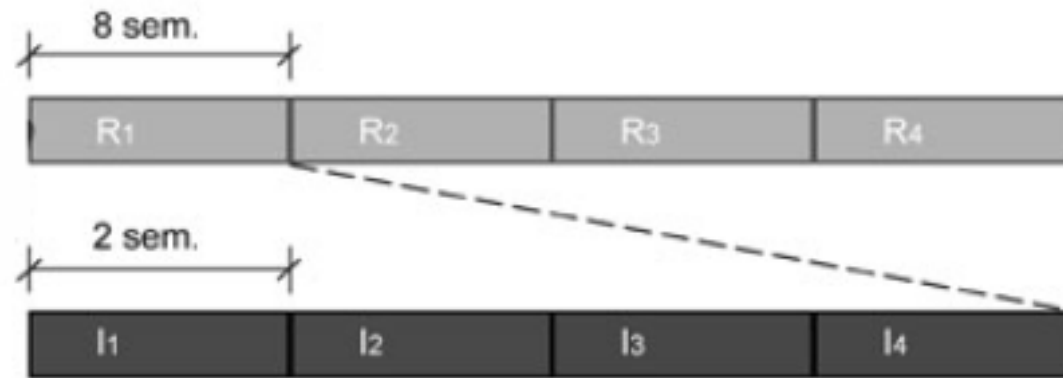


- O planejamento em XP aloca histórias (fragmentos de funcionalidades) em releases e iterações. Elas são registradas em pequenos cartões, fáceis de serem manipulados pelo cliente e pelos desenvolvedores.

Práticas do XP

- Jogo do Planejamento

releases



iterações



Práticas do XP

- Stand up meeting

- O *stand up meeting* é uma reunião que força uma aproximação dos desenvolvedores de forma diária e contínua.
 - Ela procura alinhar os membros da equipe informando os resultados obtidos no dia anterior e permitindo que os participantes priorizem as atividades do dia que se inicia.
- Projetos XP procuram assegurar que a equipe trabalhe sempre naquilo que é mais prioritário primeiro.



Práticas do XP

- Programação em Par

- Um estilo de programação no qual *dois* programadores trabalham lado a lado em um computador, continuamente colaborando no mesmo design, algoritmo, código e teste.
 - Um efeito conhecido como **pressão do par** costuma estar presente durante a programação em par e exerce influência sobre a produtividade dos desenvolvedores.



Práticas do XP

- Programação em Par

- Embora a programação em par ofereça oportunidades de melhoria nos projetos, a preocupação é se haverá um aumento de 100% do programador-hora para fazer o trabalho que um pode fazer.
 - No entanto, pesquisas demonstram que a programação em par eleva o número de programador-hora em apenas 15%.



Práticas do XP

- Código Coletivo

- Todas as classes e métodos pertencem à equipe e qualquer membro da equipe pode melhorar o que for necessário.
 - A equipe avança mais rapidamente porque ninguém precisa esperar até que outra pessoa apareça para consertar alguma coisa.
 - Todo trabalho fica menos suscetível de ser afetado se alguém ficar doente (...) ou faltar (...)



Práticas do XP

- Código Padronizado

- Em projetos XP, os programadores codificam seguindo um padrão de código acordado.
- A adoção de um padrão ajuda a simplificar a comunicação, a programar em par e a tornar o código coletivo.



Práticas do XP

- Design Simples

- Existem pelo menos quatro coisas que os usuários de um software esperam dele:
 - Que faça a coisa certa;
 - Que funcione;
 - Que seja fácil de utilizar e
 - Que possa evoluir com o tempo.



Práticas do XP

- Design Simples

- Equipes XP procuram assegurar que o design seja mantido simples e eventuais flexibilidades sejam adiadas até que se tornem efetivamente necessárias em função das histórias solicitadas pelos usuários.



Práticas do XP

- Desenvolvimento orientado a testes
 - Consiste em escrever um mecanismo de teste automatizado antes de codificar cada história e cada método do sistema
 - Os testes automatizados procuram comprovar que as solicitações dos usuários estão sendo atendidas de forma correta.



Práticas do XP

- Desenvolvimento orientado a testes
 - O XP se concentra sobretudo em dois tipos de testes: o **teste de unidade** e o **teste de aceitação**.
 - O primeiro tenta assegurar que cada componente do sistema funcione corretamente.
 - O segundo procura testar a interação entre os componentes, as quais dão origem às funcionalidades.



Práticas do XP

- Refatoração

- Projetos XP investem diariamente no aprimoramento do design e na identificação de pontos da arquitetura que estejam se degradando.

- A **refatoração** é o processo de fazer mudanças em um código existente e funcional sem alterar seu comportamento externo. Em outras palavras:

- Alterar *como* ele faz, mas não *o que* ele faz



Práticas do XP

- Integração Contínua

- Equipes XP normalmente são compostas por diversos programadores, trabalhando em pares de acordo com a prática de código coletivo. Isso cria dois problemas práticos:
 - 1) sempre que diversos indivíduos estão trabalhando na mesma coisa, ocorre uma necessidade de sincronização
 - 2) os pares precisam ser capazes de evoluir rapidamente sem interferir no trabalho uns dos outros.



Práticas do XP

- Integração Contínua

- A solução é a integração contínua:

- Os pares trabalham de forma isolada, porém integram o que produzem com a versão mais recente do código de produção, diversas vezes ao dia.
 - Isto é, os pares se sincronizam com frequência à medida que terminam pequenas atividades de codificação.
 - FERRAMENTAS: CVS, Subversion, SourceSafe



Práticas do XP

- Releases curtos

- Consiste em colocar o sistema em produção com frequência, em prazos curtos, normalmente de dois ou três meses.
 - Trabalhando com releases curtos e priorização permanente, a equipe procura assegurar que os primeiros releases gerem a maior parte do valor do sistema (por conterem, em princípio, as funcionalidades com maior valor para o negócio).
 - Releases curtos e freqüentes provêm benefício cedo para o cliente enquanto fornecem feedback rápido para os programadores.



Práticas do XP

■ Metáfora

- Para que exista um mecanismo capaz de alinhar o pensamento dos diversos projetistas assegurando que todos compartilhem uma visão única de como adicionar e manter as funcionalidades do sistema, é utilizado o conceito de **metáfora**.
 - Um exemplo recorrente de uso eficaz de uma metáfora são as interfaces gráficas baseadas em janelas



Práticas do XP

- Ritmo sustentável
 - Em síntese, essa prática recomenda que os membros da equipe de desenvolvimento trabalhem apenas durante o tempo regulamentar, ou seja, oito horas por dia, e evitem horas-extras tanto quanto possível.
 - Além disso, sugere-se que os prazos sejam mantidos fixos e as cargas de trabalho sejam ajustadas (através de priorização) para se adequar aos prazos.