

# Engenharia de Software – Metodologias Ágeis(XP)

Prof. Washington Almeida, MSC, ISF 27002

## XP

- **EX**treme **P**rogramming (XP) é talvez o mais conhecido e mais utilizado dos métodos ágeis.
- O nome foi cunhado por Beck (2000), pois a abordagem foi desenvolvida para impulsionar práticas reconhecidamente boas, como o desenvolvimento iterativo, a níveis 'extremos'.
- Por exemplo, em XP, várias novas versões de um sistema podem ser desenvolvidas, integradas e testadas em um único dia por programadores diferentes.

## XP

- Os requisitos são expressos como cenários (chamados de **estórias do usuário**), que são implementados diretamente como uma série de tarefas.
- Os programadores trabalham em pares (**programação em pares**) e desenvolvem testes para cada tarefa antes de escreverem o código (**TDD**).
- Quando o novo código é integrado ao sistema (**integração contínua**), todos os testes devem ser executados com sucesso.
- Há um curto intervalo entre os releases do sistema.

## Práticas

1. O desenvolvimento incremental é sustentado por meio de pequenos e frequentes releases do sistema. Os requisitos são baseados em cenários ou em simples histórias de usuários, usadas como base para decidir a funcionalidade que deve ser incluída em um incremento do sistema.
2. O envolvimento do cliente é sustentado por meio do engajamento contínuo com a equipe de desenvolvimento. O representante do cliente participa do desenvolvimento sendo responsável por definir os testes de aceitação para o sistema (**ATDD**).
3. Pessoas — não processos — são sustentadas por meio de programação em pares, propriedade coletiva do código do sistema e um processo de desenvolvimento sustentável que não envolve horas de trabalho excessivamente longas.

## Práticas

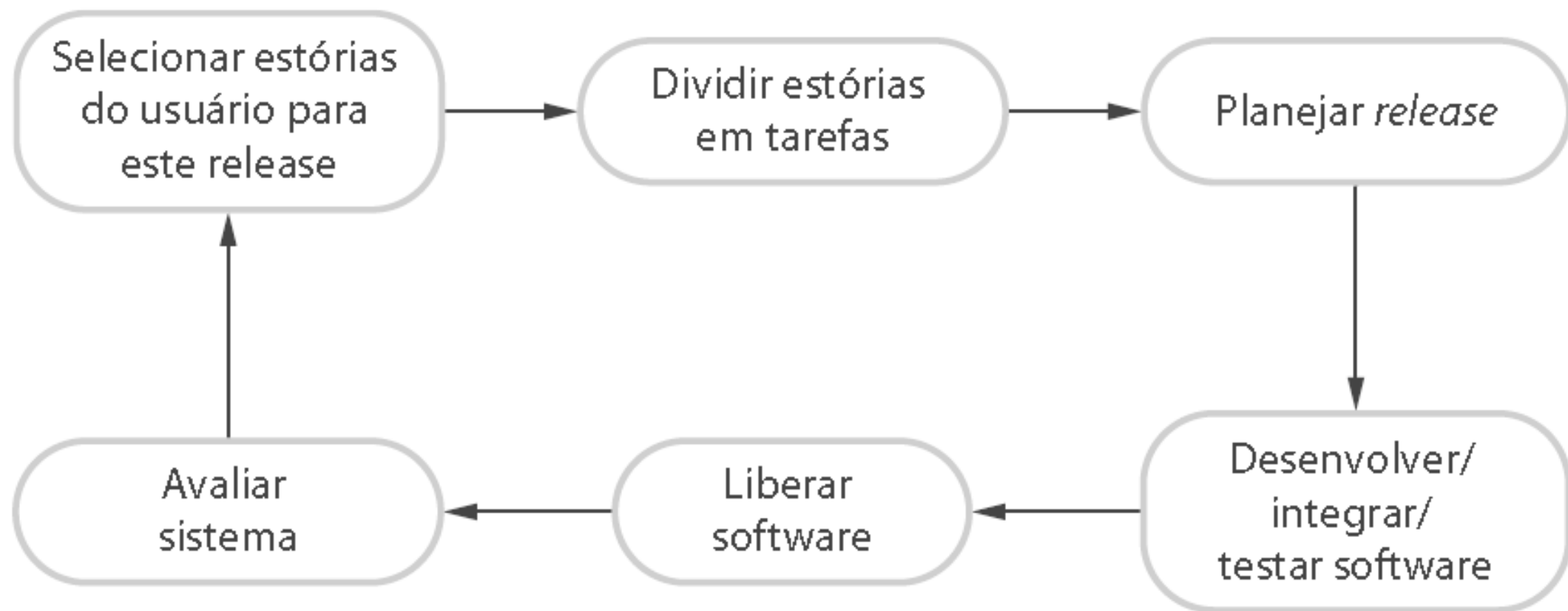
4. As mudanças são aceitas por meio de releases contínuos para os clientes, do desenvolvimento test-first, da refatoração para evitar a degeneração do código e integração contínua de nova funcionalidade.
5. A manutenção da simplicidade é feita por meio da refatoração constante, bem como por meio de projetos simples que não antecipam desnecessariamente futuras mudanças no sistema.



## Processo XP

- Em um processo XP, os clientes estão intimamente envolvidos na especificação e priorização dos requisitos do sistema.
- Os requisitos não estão especificados como uma lista de funções requeridas do sistema.
- Pelo contrário, o cliente do sistema é parte da equipe de desenvolvimento e discute cenários com outros membros da equipe.
- Juntos, eles desenvolvem um 'cartão de estória' - CRC, englobando as necessidades do cliente.

## O ciclo de um *release* em Extreme Programming



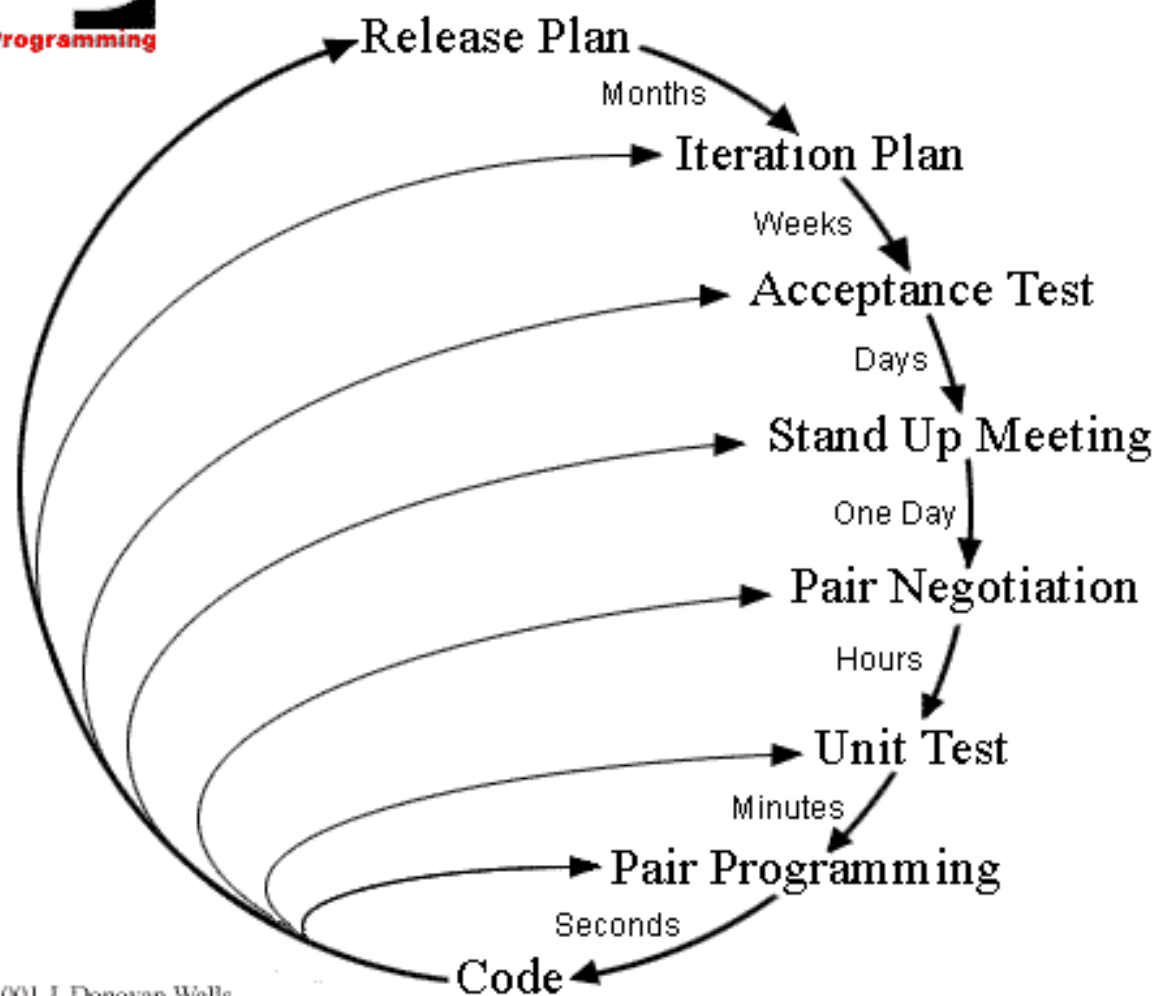
# Práticas

Princípio ou prática	Descrição
Planejamento incremental	Os requisitos são gravados em cartões de estória e as estórias que serão incluídas em um release são determinadas pelo tempo disponível e sua relativa prioridade. Os desenvolvedores dividem essas estórias em 'Tarefas'. Veja os quadros 3.1 e 3.2.
Pequenos <i>releases</i>	Em primeiro lugar, desenvolve-se um conjunto mínimo de funcionalidades útil, que fornece o valor do negócio. <i>Releases</i> do sistema são frequentes e gradualmente adicionam funcionalidade ao primeiro <i>release</i> .
Projeto simples	Cada projeto é realizado para atender às necessidades atuais, e nada mais.
Desenvolvimento <i>test-first</i>	Um <i>framework</i> de testes iniciais automatizados é usado para escrever os testes para uma nova funcionalidade antes que a funcionalidade em si seja implementada.
Refatoração	Todos os desenvolvedores devem refatorar o código continuamente assim que encontrarem melhorias de código. Isso mantém o código simples e manutenível.
Programação em pares	Os desenvolvedores trabalham em pares, verificando o trabalho dos outros e prestando apoio para um bom trabalho sempre.
Propriedade coletiva	Os pares de desenvolvedores trabalham em todas as áreas do sistema, de modo que não se desenvolvam ilhas de <i>expertise</i> . Todos os conhecimentos e todos os desenvolvedores assumem responsabilidade por todo o código. Qualquer um pode mudar qualquer coisa.
Integração contínua	Assim que o trabalho em uma tarefa é concluído, ele é integrado ao sistema como um todo. Após essa integração, todos os testes de unidade do sistema devem passar.
Ritmo sustentável	Grandes quantidades de horas-extra não são consideradas aceitáveis, pois o resultado final, muitas vezes, é a redução da qualidade do código e da produtividade a médio prazo.
Cliente no local	Um representante do usuário final do sistema (o cliente) deve estar disponível todo o tempo à equipe de XP. Em um processo de Extreme Programming, o cliente é um membro da equipe de desenvolvimento e é responsável por levar a ela os requisitos de sistema para implementação.





## Planning/Feedback Loops Zoom Out



Copyright 2001 J. Donovan Wells.

# Questão 1

Ano: 2016 Banca: [FCC](#) Órgão: [CREMESP](#) Prova: [FCC - 2016 - CREMESP - Analista de Tecnologia da Informação - Análise de Sistemas](#)

Considere que nos projetos do CREMESP baseados em XP pratica-se a propriedade coletiva de código, de forma que todos os desenvolvedores podem fazer alterações e refatoração de qualquer parte do código a qualquer momento. Para isso, é necessário que também haja:

- a) padrões de codificação.
- b) *time-box* de 40 horas.
- c) testes apenas depois da codificação.
- d) *releases* grandes.
- e) integração das funcionalidades, mesmo com erros.

**LETRA A**

**Justificativa:** Padrões de Codificação - Todos mexem em todos os códigos, todos refatoram e todos trabalham em pares. Assim é interessante mantermos um padrão para termos algo solidificado. Por isso a melhor forma é a equipe definir um padrão de codificação sempre no início dos projetos.

## Questão 2

Ano: 2018 Banca: [COMPERVE](#) Órgão: [UFRN](#) Prova: [COMPERVE - 2018 - UFRN - Analista de Tecnologia da Informação - 103](#)

Programação Extrema (XP - *Extreme Programming*) é uma das principais metodologias ágeis já propostas. Considere as seguintes afirmativas a respeito de suas práticas:

- I. A técnica de refatoração promove mudanças no código que visam à adição de novas funcionalidades.
- II. XP determina a produção de um executável do sistema desenvolvido a cada iteração.
- III. XP motiva a criação de projetos simples onde requisitos futuros não são inicialmente contemplados.
- IV. Integração contínua consiste na geração de *builds* diários do sistema.

Estão corretas as afirmativas

- a) II e IV.
- b) I e IV.
- c) I e III.
- d) II e III.

**LETRA D**

### Justificativa:

**Refatoração** - Focaliza a lapidação do projeto do software e está presente em todas as etapas do desenvolvimento. A refatoração deve ser feita sempre que possível, buscando principalmente simplificar o código atual sem perder nenhuma funcionalidade.

**Projeto simples** - O software desenvolvido de acordo com a metodologia XP deve ser o mais simples possível e satisfazer os requisitos atuais, sem a preocupação de requisitos futuros.

**Integração contínua** - É a prática de interagir e construir o sistema de software várias vezes por dia, mantendo os programadores em sintonia, além de possibilitar processos rápidos. Integrar apenas um conjunto de modificações de cada vez é uma prática que funciona bem porque fica óbvio quem deve fazer as correções quando os testes falham.

# Gabarito

Questão	Resposta
1	LETRA A
2	LETRA D

## Continua...

- XP
- TDD
- ATDD
- BDD
- Kanban
- Outros Tópicos Relevantes





# Referências

- PRESSMAN, Roger S. ; Bruce R. Maxim. Engenharia de Software, Uma Abordagem Profissional, 8° ed. Porto Alegre: AMGH, 2016. ISBN 978-85-8055- 533-2.
- SOMMERVILLE, Ian. Engenharia de Software, 9. ed. São Paulo: Pearson Prentice Hall, 2011. ISBN 978-85-7936-108-1.