

Extreme Programming

Objetivo da Aula

Conhecer o modelo de desenvolvimento ágil **Extreme Programming**, conhecido como modelo XP e saber quais são os valores e as características deste modelo.

Apresentação

Os métodos de desenvolvimento ágeis apresentam alguns importantes modelos que vêm sendo amplamente utilizados no mercado para o desenvolvimento de software.

Um dos mais conhecidos modelos é o chamado **Extreme Programming**, ou simplesmente XP. Este modelo tem como valores fundamentais a comunicação entre equipes de desenvolvimento, clientes e usuários.

Outra característica importante deste modelo é a chamada Programação em Pares e o refatoramento do código.

Nesta aula você vai conhecer um pouco das características, valores e detalhes de uso deste modelo de desenvolvimento ágil.

1. O Modelo XP

O modelo de desenvolvimento **Extreme Programming**, ou simplesmente XP, nasceu por volta de 1999 quando o desenvolvedor Kent Beck, um dos autores do **Manifesto Ágil**, batizou esse processo de trabalho que tinha a intenção de impulsionar boas práticas de desenvolvimento, lançando o livro **Extreme Programming Explained: Embrace Change**.

A ideia era utilizar o modelo iterativo em condições em que era possível criar várias versões do mesmo software desenvolvidas, integradas e testadas por programadores diferentes em um único dia.

Link

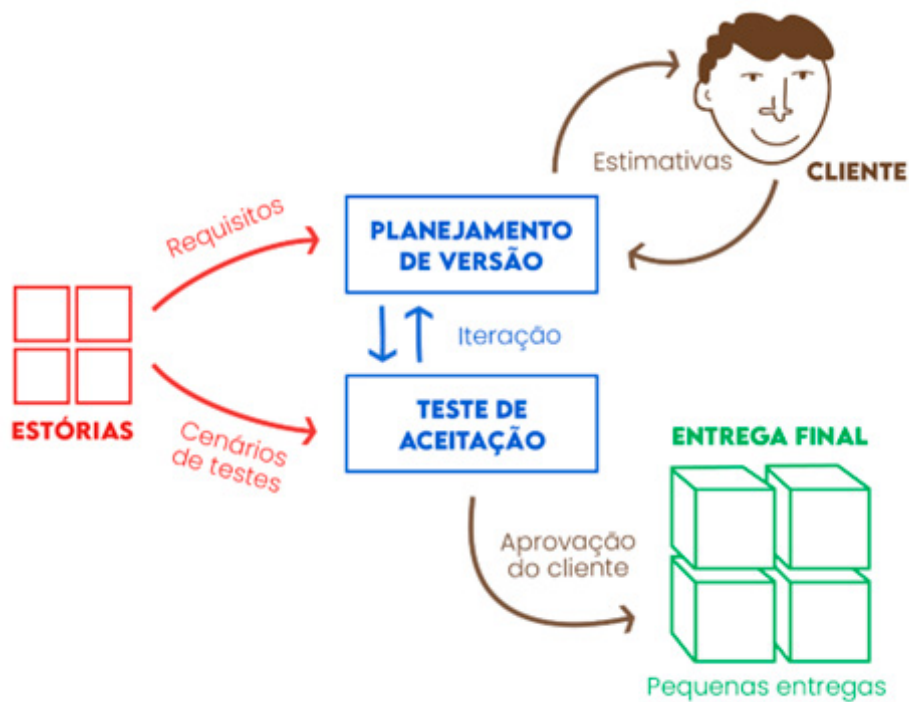
Conheça os detalhes do XP na comunidade brasileira de desenvolvimento ágil. Disponível em: www.desenvolvimentoagil.com.br/xp/. Acesso em: 15 nov. 2022.



O XP trabalha com a definição de requisitos no que chamamos de “estórias do usuário” que são implementadas a partir de uma série de tarefas. O desenvolvimento é feito a partir da programação em pares.

A estória deverá ser descrita pelo cliente, que é quem detém o conhecimento do funcionamento do negócio. Quando uma estória for muito grande, ela deve ser dividida em tarefas com duração máxima de alguns dias.

Figura 1: Ciclo de Vida do Modelo XP



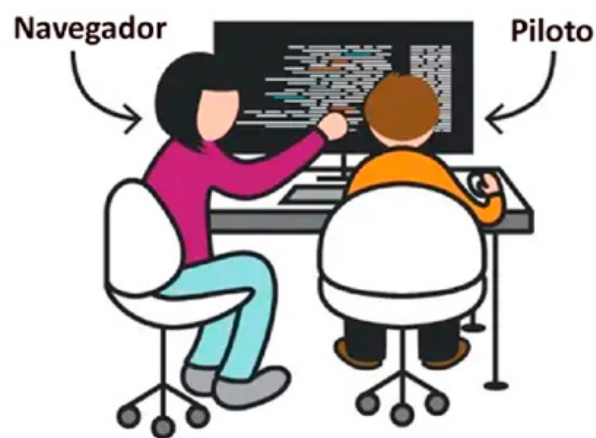
Fonte: Adaptado de www.digite.com/pt-br/agile/programacao-extrema-xp/. Acesso em: 15 nov. 2022.

Na Figura 1 vemos o ciclo de vida do XP, onde é possível ver o seu processo de funcionamento, onde os requisitos são estimados, as versões são planejadas e, a cada iteração, há um novo teste de aceitação junto ao cliente, que aprova a nova versão do sistema. As estórias do cliente se transformam em requisitos validados em cenários de testes.

2. Programação em Pares

A programação em pares (**Pair Programming**) é um conceito chave no XP. Nesta abordagem, todo e qualquer código implementado no projeto deverá ser feito a partir de uma dupla de programadores que desenvolvem testes para cada tarefa antes de elaborarem o código, ao finalizarem a tarefa, os códigos são novamente testados, mas desta vez com o incremento integrado ao sistema.

Figura 2: Programação em Pares



Fonte: Adaptado de medium.com/@pedron.ketlin/coding-dojos-e-pair-programming-d4de64da0d20 Acesso em: 15 nov. 2022.

Na Figura 2 vemos a dinâmica de trabalho da programação em pares: dois programadores, onde um é identificado como piloto e o outro como navegador.

Na técnica de desenvolvimento em pares, dois desenvolvedores trabalham no mesmo problema, ao mesmo tempo, e no mesmo computador. Um deles é responsável pela digitação, ele é chamado de piloto, e o outro acompanha o trabalho da sua dupla, ele é chamado de navegador.

Enquanto um deles codifica o problema, o outro tem a responsabilidade de revisar o código digitado, onde pequenos erros de programação que demorariam algumas horas para serem depurados são percebidos mais rapidamente pelo navegador da dupla.

Um dos grandes benefícios da programação em pares é a troca de experiências e ideias entre os desenvolvedores, permitindo que um veja um problema que o outro não conseguiu identificar. Há alguma discussão, porém, sobre a questão do custo de se manter dois profissionais envolvidos em uma mesma tarefa.

A programação em pares resulta no chamado código coletivo, onde o código fonte não tem dono, fazendo com que ninguém precise solicitar permissão para poder modificar o mesmo. O objetivo é fazer a equipe conhecer todas as partes do sistema, o que faz com que toda a equipe domine os problemas que estão sendo resolvidos com o sistema.

3. Refatoramento

Outra característica marcante no XP é o refatoramento. A filosofia por trás do XP considera que toda vez que o desenvolvedor encontrar um código duplicado, pouco legível, mal codificado, sem padronização, lento, com código legado ou uso incorreto de outras implementações, ele deverá modificar este código tornando-o mais legível e simples, porém esta alteração deve garantir que o comportamento do código em questão não mude o comportamento e o resultado final do código.

A prática do refatoramento é um processo que permite a melhoria contínua da programação, a eliminação de erros visíveis, e a compatibilidade com o código já existente. Refatorar ajuda na melhora da clareza, leitura do código e facilita a manutenção do código, garantindo uma maior longevidade ao mesmo. Além disso, o processo de refatoração garante maior coesão ao código, levando ao melhor aproveitamento e evitando duplicação no código-fonte.

4. Outras Práticas

Além das marcantes práticas de Programação em Pares e Refatoramento, o XP possui outras práticas importantes em sua gestão diária.

A participação ativa do cliente é uma importante questão na metodologia, pois ele poderá acompanhar o progresso do desenvolvimento, além de aplicar diversas validações do que vem sendo construído pela equipe de desenvolvimento.

O desenvolvimento é iterativo, com ciclos realizados em períodos de 1 ou 2 semanas, onde tarefas são agrupadas a partir de conversas e negociações com o cliente, que é quem vai determinar as suas prioridades.

As reuniões são realizadas a partir de **Stand-Up Meetings**, reuniões de curta duração feitas em pé, com o objetivo de saber o que cada um está fazendo, em que ponto está o projeto e se alguém está tendo problemas para executar suas tarefas. Mesmo que surja algum indício de problema, essa reunião não tem o objetivo de pensar em soluções naquele momento.

Sobre testes, o XP usa o conceito de Desenvolvimento Orientado a Testes (**Test Driven Development**), conhecido como TDD. Em um primeiro momento são realizados testes unitários, e depois é criado o código para que o teste funcione. Os testes unitários são essenciais para que a qualidade do projeto seja mantida durante todo o ciclo de vida do desenvolvimento.

5. Valores do XP

Assim como toda metodologia, o XP possui um conjunto de valores e princípios. Os valores do XP são definidos por comunicação, simplicidade, feedback, coragem e respeito.

- Comunicação – é o valor básico do XP, a comunicação deve ocorrer o tempo todo, entre equipes, usuários e patrocinador do projeto;
- Simplicidade – se existe um modo mais simples de se fazer algo, por que não seguir por esse caminho?
- Feedback – o **feedback** deve ser rápido e constante. Se há algo para ser corrigido, isso deve ser feito o mais breve possível;
- Coragem – é preciso coragem para falar a verdade, mesmo que essa verdade signifique trazer notícias ruins, como estimativas e custos honestos;
- Respeito – o respeito deve imperar entre todos os stakeholders internos e externos ao projeto.

Já os princípios são definidos por: **feedback** rápido, presumir simplicidade, abraçar mudanças, trabalho de alta qualidade, pequenos passos, melhoria, diversidade, reflexão.

6. Papéis no XP

Em uma equipe organizada para trabalhar como o método ágil XP, normalmente teremos a seguinte configuração:

- Cliente – principal interessado no resultado final do projeto;
- Gerente de Projeto – responsável por liderar o projeto e resolver problemas administrativos;
- **Coach** – responsável por questões técnicas do projeto;
- Analista de testes – responsável por garantir a qualidade do sistema a partir da realização de testes;
- Redator técnico – responsável por documentar o sistema;
- Desenvolvedores – responsáveis por analisar, projetar e implementar o sistema.

Kent Beck, criador da filosofia XP, afirma que uma equipe madura de XP não deve confiar em papéis rígidos, mas reconhecer que os papéis podem ser úteis para as equipes iniciantes, até que elas comecem a atrapalhar a colaboração, ou seja, a ideia é não burocratizar o processo por conta das definições de papéis fixas.

Considerações Finais

Nesta aula falamos sobre a história da metodologia ágil **Extreme Programming**, mais conhecida como metodologia XP. A partir desta aula podemos conhecer a filosofia, os valores e os princípios desta que é uma das principais metodologias ágeis existentes no mercado.

Conhecemos um pouco também sobre as suas principais características como a programação em pares e a refatoração de código, e como essas características agregam valor ao XP.

As práticas do XP são importantes e construídas sobre valores, princípios e práticas, visando como objetivo permitir que equipes pequenas e médias produzam software de alta qualidade, de maneira rápida, e se adaptem a requisitos em evolução e mudança, como o mercado requer que estejamos preparados hoje.

Materiais Complementares

**Vídeo**

Pair Programming – Programação em Pares. Disponível em: <https://www.youtube.com/watch?v=5M8yNQSFBPg>. Acesso em: 15 nov. 2022.

**Vídeo**

O que é XP? Disponível em: <https://www.youtube.com/watch?v=bWTTPLrQ-8>. Acesso em: 15 nov. 2022.

**Site**

O que é Programação Extrema (XP) e seus valores, princípios e práticas? Disponível em: <https://www.digite.com/pt-br/agile/programacao-extrema-xp/>. Acesso em: 15 nov. 2022.

Referências

PRESSMAN, R.G. **Engenharia de Software**. 9ª ed. Rio de Janeiro: McGraw-Hill, 2021.

SOMMERVILLE, I. **Engenharia de Software**. 10ª ed. São Paulo: Pearson Addison. Wesley, 2019.