

# EXPLICAR O PROCESSO DE DESENVOLVIMENTO ÁGIL EXTREME PROGRAMMING(XP)

## DESENVOLVIMENTO ÁGIL DE SOFTWARE

Até meados dos anos 1990, os processos de desenvolvimento de software prescritivos exigiam um detalhado planejamento de projeto, um sistema de garantia da qualidade estabelecido, a aplicação de métodos de análise e projeto, tal como a modelagem de caso de uso, apoiados por ferramentas CASE e controlados por um disciplinado e formal modelo de processo de desenvolvimento de software.



Essa visão era influenciada por equipes de engenheiros de software engajados em sistemas de software complexos que exigiam um esforço significativo de planejamento, projeto e documentação de sistema.

Mas esse esforço justifica-se em sistemas de menor grau de complexidade?

Em 2001, um grupo de 17 metodologistas e outros profissionais da indústria de software se reuniram no estado norte-americano de Utah para discutir e propor uma alternativa aos processos prescritivos dominantes à época. Ao final da reunião, esse grupo lançou as bases para um novo conceito de processo de software, as quais foram registradas em um documento que chamaram de *Manifesto para o Desenvolvimento Ágil de Software*, que inclui quatro valores:

- 1. Indivíduos e interações, mais do que processos e ferramentas.**
- 2. Software em funcionamento, mais do que documentação abrangente.**

**3. Colaboração com o cliente, mais do que negociação de contratos.**

**4. Resposta a mudanças, mais do que seguir um plano.**

A maioria dos métodos ágeis de desenvolvimento de software têm foco no próprio software, em vez de em seu projeto e documentação, assumem um fluxo de processo iterativo e incremental para entrega de software e apoiam aplicações de negócios nas quais os requisitos possuem alta volatilidade durante o desenvolvimento.

Um projeto de software ágil busca a capacidade de implantar uma nova versão a cada incremento, enfatizando o software funcional como uma medida primária de progresso.

A entrega de software rápida permite que os clientes validem de forma imediata o incremento, indicando possíveis erros ou alterações de requisitos, bem como descrevendo novos requisitos para as próximas iterações.

### **Atenção**

Métodos ágeis também enfatizam comunicações em tempo real entre equipe de desenvolvimento e clientes.

## **EXTREME PROGRAMMING - XP**



A Extreme Programming, ou simplesmente XP, é considerada um processo de desenvolvimento de software ágil, tendo sido proposta por **Kent Beck** em 1999. A XP adota valores que servem como critérios que norteiam as partes interessadas de um projeto, incluindo: comunicação, simplicidade, feedback, coragem e respeito.

Vejamos uma breve descrição de cada valor:

## Comunicação

A **comunicação** objetiva construir um entendimento dos requisitos com o mínimo de documentação formal e o máximo de interação entre equipe de desenvolvimento e clientes.

## Simplicidade

A **simplicidade** sugere que a equipe adote soluções mais simples, criando um ambiente onde o custo de mudanças no futuro seja baixo.

## Feedback

O **feedback** permite que os programadores também tenham feedback quando da execução dos testes e quando da validação funcional por parte dos clientes, permitindo a correção de erros e alterações nos requisitos.

## Coragem

A **coragem** é necessária para lidar com o risco de erro, que deve ser considerado natural e se traduz em confiança nos seus mecanismos de prevenção e proteção.

## Respeito

O **respeito** é o valor mais básico e que dá sustentação a todos os demais, e sem o qual não haverá nada que possa salvar um projeto de software do insucesso.

## Saiba mais

Assim como um segundo nível de abstração, a XP possui cinco princípios que servem para ajudar a equipe de software na escolha de alternativas de solução de problemas durante o projeto. Os princípios incluem: feedback rápido, assumir simplicidade, mudança incremental, abraçando mudanças e trabalho de qualidade.

A XP possui doze práticas que se enquadram nos valores e princípios descritos na tabela a seguir:

**TABELA 1 – Doze práticas enquadradas nos valores e princípios da XP**

Prática	Descrição
Jogo de Planejamento	Os requisitos são registrados em cartões, ou fichas, denominados “história de usuários”, sendo a descrição e a priorização realizadas pelos clientes. A equipe de desenvolvimento estima a duração de cada <i>release</i> .

**TABELA 1 – Doze práticas enquadradas nos valores e princípios da XP**

Pequenas Releases	Cada <i>release</i> deve ser a menor possível, incluindo os requisitos mais importantes para o negócio. As <i>releases</i> frequentes permitem um maior feedback para clientes e equipe de desenvolvimento, facilitando o aprendizado e a correção dos defeitos do software.
Metáfora	Facilita a comunicação com o cliente, entendendo a sua realidade, isto é, procura traduzir as palavras do usuário para o significado que ele espera dentro do projeto. Como exemplo, a metáfora “Estoque 4.0” para um subsistema de gestão virtual dos estoques no contexto de um sistema de logística.
Projeto Simples	Projeto simples inclui projetar somente as histórias de usuários, ou funcionalidades, previamente definidas, bem como se concentrar em soluções de projeto simples e bem estruturadas, tal como o menor número possível de classes e métodos.
Cliente no local de Trabalho	Incluir um cliente na equipe de desenvolvimento para responder perguntas, ou alguém da parte do cliente com conhecimento do negócio. O cliente é protagonista em um projeto XP.
Semana de 40 horas	Estimula a busca de um ritmo sustentado de trabalho, tal como 40 horas semanais. Horas extras são permitidas quando trouxerem produtividade para a execução do projeto, devendo estar restringidas a duas semanas seguidas.
Programação Pareada	Todo código deve ser gerado por dois programadores trabalhando em um único computador. O programador codifica e o parceiro revisa e sugere. Dessa forma, o programa sempre é revisto por duas pessoas, evitando e diminuindo, assim, a possibilidade de defeitos. Com isso, busca-se sempre a evolução da equipe, melhorando a qualidade do código fonte gerado.
Propriedade Coletiva	O código fonte pode ser alterado por qualquer membro da equipe de desenvolvimento, de modo que esta detém a propriedade do código. O objetivo com isso é fazer a equipe conhecer todas as partes do sistema. Os testes de unidade e integração garantem a estabilidade dessa prática.
Padronização do Código	A prática da propriedade coletiva de código exige o estabelecimento de regras para programar a serem seguidas por todos. O objetivo é facilitar o entendimento do código.
Desenvolvimento Orientado a Testes	<p>Um framework automatizado é sugerido para a escrita dos testes unitários, antes que determinada funcionalidade seja implementada, técnica denominada “<i>test first</i>”.</p> <p>Os testes de integração também são realizados sistematicamente.</p> <p>Os testes de validação são especificados pelos clientes a partir das histórias de usuários.</p>
Refatoração	A refatoração é um processo de modificar a estrutura interna do software sem alterar o seu comportamento externo, permitindo a melhoria contínua da programação com o mínimo de introdução de erros e mantendo a compatibilidade com o código já existente.
Integração Contínua	O sistema deverá ser integrado sempre que produzir uma nova funcionalidade. Essa continuidade permite a detecção imediata de erros de integração, evitando problemas de compatibilidade e interface.

## PROCESSO XP

A XP adota o paradigma orientado a objetos, aplicando seus valores, princípios e práticas em um processo com quatro atividades metodológicas:

planejamento, projeto, codificação e testes. A Figura 10 ilustra essas atividades metodológicas propostas por Pressman (2016).

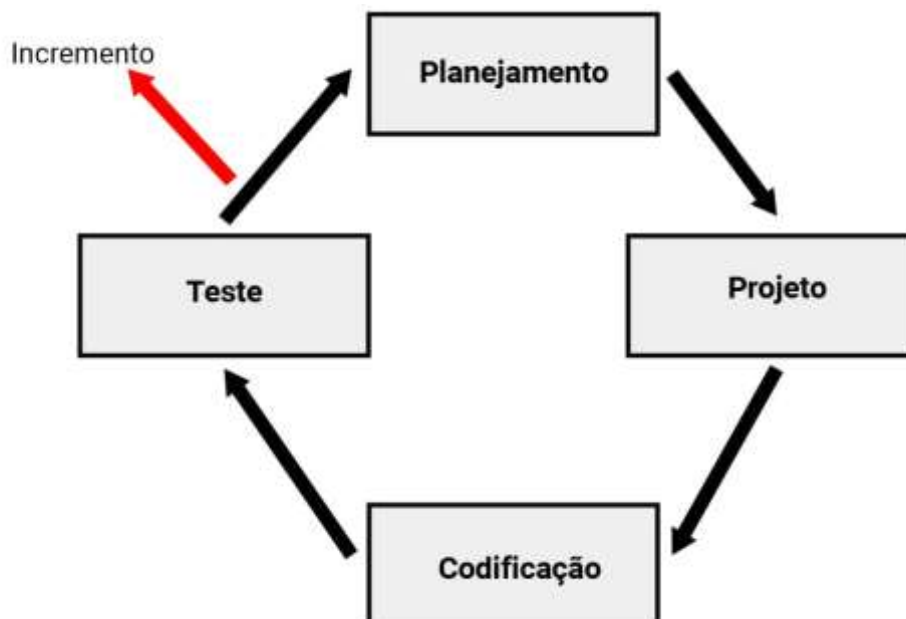


Figura 10 – O processo XP. Fonte: Autor

## O PROCESSO ÁGIL EXTREME PROGRAMMING (XP)

Assista, a seguir, o processo XP e suas etapas de planejamento, projeto, codificação e teste:

### O processo ágil eXtreme Programming (XP)

Veja mais detalhes sobre o processo XP e suas etapas:

#### PLANEJAMENTO

A primeira atividade é ouvir o cliente, realizando um levantamento de requisitos que possibilite o entendimento do escopo do problema. Essa atividade gera um conjunto de histórias de usuários, escritas pelos clientes, juntamente à equipe de desenvolvimento que, de forma simplificada, compõem as necessidades destes. Cada história é colocada em uma ficha ou cartão, sendo priorizadas pelos clientes em função da agregação de valor ao negócio.

Após a elaboração dos cartões, a equipe de desenvolvimento avalia as histórias e as divide em tarefas, estimando o esforço e os recursos necessários à implementação. Caso o esforço exija mais do que três semanas de desenvolvimento, é preciso que o cliente fatore a história analisada em histórias menores.

As histórias que farão parte de cada incremento são selecionadas de forma conjunta por equipe de desenvolvimento e clientes, considerando os seguintes critérios: as histórias de maior risco serão priorizadas, em seguida, as que agregam maior valor ao negócio.

Após a implantação da primeira versão do software, é calculada a velocidade do projeto que é o número de histórias de usuários implementadas na primeira release. Essa velocidade serve para avaliar se o compromisso firmado está compatível e auxiliar na estimativa de prazos das versões seguintes.

## **PROJETO**

Aqui temos a máxima: “Não complique!”

A XP estimula a modelagem com cartões CRC (Classe-Responsabilidade-Colaborador), que permitem identificar as classes orientadas a objetos relevantes para a iteração corrente do software. Os cartões CRC são os únicos artefatos de projeto gerados durante o processo XP.

Caso algum requisito necessite ser validado em função de sua complexidade, é recomendada a prototipação, como solução pontual, a fim de reduzir o risco dessa especificação no projeto.

A refatoração também é estimulada, tendo em vista uma melhor organização do código interno do software sem impactar a sua funcionalidade, o que minimiza as possibilidades de erros.

## **CODIFICAÇÃO**

A XP sugere que dois programadores trabalhem juntos em um mesmo computador a fim de criarem código para uma história, fornecendo um mecanismo de solução de problemas em tempo real, cabendo a máxima de que duas cabeças pensam melhor que uma, além de garantir a qualidade do código, pois a sua revisão é imediata. Ambos desenvolvedores têm papéis distintos, tal como, uma pessoa implementa o código, outro verifica os padrões de codificação.

Quando da conclusão da codificação, esse código é integrado aos demais. A estratégia de integração contínua possibilita a identificação precoce de erros.

## **TESTE**

Como afirmado anteriormente, a equipe de desenvolvimento avalia cada história e a divide em tarefas, que representa uma característica discreta do sistema. Antes da codificação, a equipe de desenvolvimento elabora os testes de unidades a serem aplicados nas tarefas. A automação desses testes é fortemente recomendada, o que estimula a realização dos **testes de regressão** toda vez que o código for alterado.

A estratégia de integração contínua está vinculada aos testes de integração, realizados de forma sistemática a cada novo código gerado a ser integrado, o que permite lançar alertas logo no início, evitando propagações indesejadas. Os testes de aceitação são elaborados pelos clientes com base nas histórias de usuários especificadas anteriormente e que serão implementadas na próxima release. O foco está nas funcionalidades externas visíveis aos clientes. Em outras palavras, os testes verificam se o sistema atende às reais necessidades dos usuários.

## RESUMINDO

Neste módulo, compreendemos as principais características das metodologias ágeis e, em particular, da Extreme Programming – XP. A partir do questionamento do excesso de formalismo dos processos prescritivos dominantes em meados da década de 1990, foi redigido o Manifesto para o Desenvolvimento Ágil de Software, com quatro valores: Indivíduos e interações, mais do que processos e ferramentas; software em funcionamento, mais do que documentação abrangente; colaboração com o cliente, mais do que negociação de contratos; e resposta a mudanças, mais do que seguir um plano.

A maioria dos métodos ágeis de desenvolvimento de software tem como foco o software, em vez de formalismos de projeto e documentação, assumem um fluxo de processo iterativo e incremental e apoiam aplicações com requisitos altamente voláteis durante o desenvolvimento.

A XP possui um conjunto de valores, princípios e práticas como base aos que assumem a metodologia XP.

O processo XP possui quatro atividades metodológicas:

O **planejamento** está fundamentado nas histórias de usuários.

O **projeto** tem um mínimo de formalismo em função de gerar um único tipo de artefato, os cartões CRC, sendo sugeridas a prototipação e refatoração nessa etapa.

A **codificação** sugere o trabalho em par de programadores alocados em um único computador, permitindo uma revisão de código em tempo real.

Os **testes** têm uma característica peculiar, são escritos antes da codificação e aplicados de forma sistemática em ambiente automatizado; na sequência, temos os testes unitários, de integração e aceitação.

