

Capítulo 1

Extreme Programming: visão geral

Extreme Programming, ou XP, é um processo de desenvolvimento de software voltado para:

- Projetos cujos requisitos são vagos e mudam com frequência;
- Desenvolvimento de sistemas orientados a objeto;
- Equipes pequenas, preferencialmente até 12 desenvolvedores;
- Desenvolvimento incremental (ou iterativo), onde o sistema começa a ser implementado logo no início do projeto e vai ganhando novas funcionalidades ao longo do tempo.

Existe uma categoria de processos de desenvolvimento conhecida como *Processos Ágeis de Desenvolvimento*, dentro da qual o XP e outros processos se encaixam. Eles compartilham a premissa de que o cliente aprende sobre suas necessidades, na medida em que é capaz de manipular o sistema que está sendo produzido. Com base no *feedback* do sistema ele re-avalia as suas necessidades e prioridades, gerando mudanças que devem ser incorporadas ao software. O aprendizado é importante, porque permite que o cliente direcione o desenvolvimento de modo que a equipe produza sempre aquilo que tem o maior valor para o seu negócio.

O XP é um processo de desenvolvimento que busca assegurar que o cliente receba o máximo de valor de cada dia de trabalho da equipe de desenvolvimento. Ele é organizado em torno de um conjunto de valores e práticas que atuam de forma harmônica e coesa para assegurar que o cliente sempre receba um alto retorno do investimento em software.

Valores do XP

O XP se baseia em quatro valores fundamentais:

- Feedback
- Comunicação
- Simplicidade
- Coragem

Quando o cliente aprende com o sistema que utiliza e re-avalia as suas necessidades, ele gera feedback para a equipe de desenvolvimento. Isto é, ele realimenta a equipe com alterações nas necessidades que ainda serão implementadas e, eventualmente, naquelas que já fazem parte do software. O feedback é o mecanismo fundamental que permite que o cliente conduza o desenvolvimento diariamente e garanta que a equipe direcione as suas atenções para aquilo que irá gerar mais valor.

Para que o cliente possa compartilhar o seu aprendizado com a equipe, é necessário que ele utilize permanentemente o valor da comunicação. **A comunicação entre o cliente e a equipe permite que todos os detalhes do projeto sejam tratados com a atenção e a agilidade que merecem.** O XP procura assegurar que a comunicação ocorra da forma mais direta e eficaz possível. Sendo assim, ele busca aproximar todos os envolvidos do projeto de modo que todos possam se comunicar face-a-face ou da forma mais rica que for viável.

A comunicação, embora seja essencial, não é suficiente para garantir que o cliente possa aprender durante o projeto e gerar feedback rapidamente. Também é necessário que a equipe compreenda e utilize **o valor da simplicidade, que nos ensina a implementar apenas aquilo que é suficiente para atender a cada necessidade do cliente.** Ou seja, ao codificar uma funcionalidade devemos nos preocupar apenas com os problemas de hoje e deixar os problemas do futuro para o futuro. Não devemos tentar prever o futuro, pois raramente acertamos nas previsões. Ao evitar especular sobre o que acontecerá amanhã, ganhamos tempo e permitimos que o cliente tenha acesso à funcionalidade mais rapidamente. Isso permite que ele a utilize no seu negócio, gerando valor para ele e tornando viável que ele dê feedback para a equipe o quanto antes.

Eventualmente, com base no feedback, a equipe poderá fazer generalizações quando elas se fizerem necessárias. Neste caso, entretanto, elas virão na forma de uma necessidade explícita e não como a especulação de algo que poderia vir a ser necessário no futuro.

Dado que o sistema é desenvolvido de forma incremental, a equipe está continuamente fazendo a manutenção do software e criando novas funcionalidades. Em muitos casos, ela irá alterar algo que vinha funcionando corretamente, o que leva ao risco de gerar falhas no sistema. Por esta razão, **a equipe precisa ser corajosa e acreditar que, utilizando as práticas e valores do XP, será capaz de fazer o software evoluir com segurança e agilidade.**

Práticas do XP

O XP se baseia nas seguintes práticas:

- Cliente Presente
- Jogo do Planejamento
- Stand Up Meeting
- Programação em Par
- Desenvolvimento Guiado pelos Testes
- Refactoring
- Código Coletivo
- Código Padronizado
- Design Simples
- Metáfora
- Ritmo Sustentável
- Integração Contínua
- Releases Curtos

Cliente Presente

O XP trabalha com a premissa de que o cliente deve conduzir o desenvolvimento a partir do feedback que recebe do sistema. Para que a troca de feedback possa ocorrer e o cliente possa obter o máximo de valor do projeto, é essencial que ele participe ativamente do processo de desenvolvimento. Além disso, a sua presença viabiliza a simplicidade do processo em diversos aspectos, especialmente na comunicação.

Jogo do Planejamento

O XP utiliza diversas formas de planejamento para assegurar que a equipe esteja sempre trabalhando naquilo que é o mais importante para o cliente. Por esta razão, todo projeto em XP é dividido em **releases** e **iterações**, de modo que cliente e equipe tenham diversas oportunidades de se reunir para revisar o planejamento.

Releases são módulos do sistema que geram um valor bem definido para o cliente. Iterações, por sua vez, são períodos de tempo de poucas semanas (em torno de duas, em média) no qual a equipe implementa um conjunto de funcionalidades acordado com o cliente. No início de cada release e de cada iteração ocorre o **jogo do planejamento**. Trata-se de uma reunião onde o cliente avalia as funcionalidades que devem ser implementadas e prioriza aquelas que farão parte do próximo release ou da próxima iteração.

No XP, as funcionalidades são descritas em pequenos cartões e são chamadas de **estórias**. Durante o jogo do planejamento as estórias são estimadas, para que o cliente possa conhecer o custo da implementação de cada uma delas. A estimativa é feita utilizando-se uma unidade especial que recebe o nome de **ponto**.

O ponto representa uma unidade de tempo que varia ao longo do desenvolvimento de acordo com a **velocidade** da equipe, onde a velocidade indica o quanto a equipe foi capaz de implementar na iteração anterior. Todos os detalhes sobre releases, iterações, estórias, pontos e velocidade serão descritos com detalhes nos capítulos subseqüentes.

Stand Up Meeting

A equipe de desenvolvimento se reúne a cada manhã para avaliar o trabalho que foi executado no dia anterior e priorizar aquilo que será implementado

no dia que se inicia. Trata-se de uma reunião rápida que recebe o nome de *stand up meeting*, que em inglês significa reunião em pé.

Programação em Par

No XP, os desenvolvedores implementam as funcionalidades em pares, ou seja, diante de cada computador, existem sempre dois desenvolvedores que trabalham juntos para produzir o mesmo código. Esta prática, que recebe o nome de **programação em par** permite que o código seja revisado permanentemente, enquanto é construído. Também contribui para que a implementação seja mais simples e eficaz, já que os desenvolvedores se complementam e têm mais oportunidades de gerar soluções inovadoras.

Desenvolvimento Guiado pelos Testes

O XP é destinado à construção de sistemas com alta qualidade, o que leva à necessidade de diversos mecanismos de validação para assegurar que o software está correto. Um destes mecanismos é a programação em par, tal como foi citado anteriormente. Além dela, o XP também utiliza a técnica de **desenvolvimento guiado pelos testes**.

Os desenvolvedores escrevem testes para cada funcionalidade antes de codificá-las. Fazendo isso, eles aprofundam o entendimento das necessidades do cliente (o que aprimora a análise), se preocupam com as interfaces externas dos métodos e classes antes de codificá-los (o que melhora o design), sabem até onde codificar cada funcionalidade (o que ajuda a manter o sistema simples) e passam a contar com uma massa de testes que pode ser usada a qualquer momento para validar todo o sistema.

Refactoring

Para que o sistema possa evoluir de forma incremental, a equipe deve fazer com que ele expresse os seus objetivos facilmente e esteja sempre claro e fácil de compreender. Frequentemente, isso levará a equipe a modificar partes do sistema que estejam funcionando para facilitar a sua manutenção.

O **refactoring** é o ato de alterar um código sem afetar a funcionalidade que ele implementa. É utilizado para tornar o software mais simples de ser manipulado e se utiliza fortemente dos testes descritos anteriormente para garantir que as modificações não interrompam o seu funcionamento.

Código Coletivo

No XP o sistema não é segmentado em partes, de modo que cada desenvolvedor fique responsável por uma delas. Os desenvolvedores têm acesso a todas as partes do código e podem alterar aquilo que julgarem importante sem a necessidade de pedir autorização de outra pessoa, pois o **código é coletivo**.

Isso fornece maior agilidade ao processo e cria mais um mecanismo de revisão e verificação do código, já que aquilo que é escrito por um par hoje, acaba sendo manipulado por outro amanhã. Se alguma coisa estiver confusa no código, o par deverá fazer refactoring para torná-lo mais legível.

Código Padronizado

Para que todos os desenvolvedores possam manipular qualquer parte do software de forma mais rápida, a equipe estabelece padrões de codificação, que servem também para tornar o sistema mais homogêneo e permitir que qualquer manutenção futura seja efetuada mais rapidamente.

Design Simples

Para que o cliente possa obter feedback logo, a equipe precisa ser ágil no desenvolvimento, o que a leva a optar pela simplicidade do design. Ao invés de criar generalizações dentro do código, de modo a prepará-lo para possíveis necessidades futuras, a equipe deve sempre optar por um código que seja suficiente para atender às necessidades da funcionalidade que está implementando. Os desenvolvedores se baseiam na premissa de que serão capazes de incorporar qualquer necessidade futura quando e se ela surgir. Para isso, eles contam com o refactoring, os testes e as demais práticas do XP para apoiá-los.

Metáfora

Para facilitar a criação de um design simples, a equipe de desenvolvimento utiliza metáforas, já que elas têm o poder de transmitir idéias complexas de forma simples, através de uma linguagem comum que é estabelecida entre a equipe de desenvolvimento e o cliente.

Ritmo Sustentável

A qualidade do design, do código, das metáforas e do sistema é determinada diretamente pela qualidade dos desenvolvedores e a capacidade que eles têm de se manter atentos, criativos e dispostos a solucionar problemas. Para garantir que a equipe tenha sempre o máximo de rendimento e produza software com melhor qualidade possível, o XP recomenda que os desenvolvedores trabalhem apenas oito horas por dia e evitem fazer horas-extras, visto que é essencial estar descansado a cada manhã, de modo a utilizar a mente na sua plenitude ao longo do dia.

Integração Contínua

Quando uma nova funcionalidade é incorporada ao sistema, ela pode afetar outras que já estavam implementadas. Para assegurar que todo o sistema esteja sempre funcionando de forma harmoniosa, a equipe pratica a **integração contínua** que leva os pares a integrarem seus códigos com o restante do sistema diversas vezes ao dia. Cada vez que um par faz isso, ele executa todos os testes para assegurar que a integração tenha ocorrido de forma satisfatória.

Uma integração sempre pode produzir erros no sistema. Sendo assim, a equipe utiliza os testes para descobrir eventuais defeitos o mais rapidamente possível já que descobri-los logo facilita e acelera a correção e diminui a probabilidade de pequenos problemas se transformarem em grandes dores-de-cabeça no futuro.

Releases Curtos

Como explicado anteriormente, o XP tem como objetivo gerar um fluxo contínuo de valor para o cliente. Sendo assim, ele trabalha com **releases curtos**, ou seja, a equipe produz um conjunto reduzido de funcionalidades e coloca em produção rapidamente de modo que o cliente já possa utilizar o software no dia-a-dia e se beneficiar dele. Durante todo o projeto, a equipe colocará o sistema em produção diversas vezes, cada vez incorporando mais funcionalidades e gerando mais valor.

A figura 1.1 apresenta todas as práticas do XP de forma resumida:

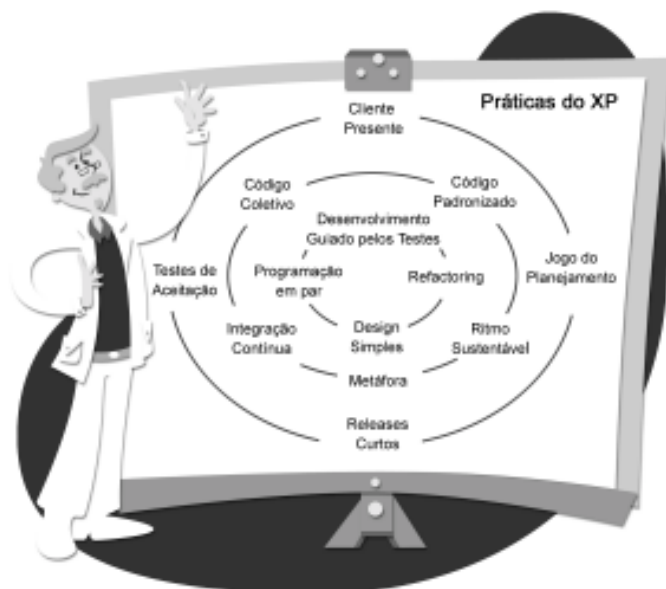


Figura 1.1 – Práticas do XP (fonte: www.xprogramming.com).

Características da equipe

Uma equipe que utilize o XP normalmente é composta por pessoas que representam os seguintes papéis:

- Gerente de Projeto
- Coach
- Analista de Teste
- Redator Técnico
- Desenvolvedor

Gerente de Projeto

O gerente de projeto é responsável pelos assuntos administrativos do projeto. Ele procura liberar a equipe de questões que não estejam diretamente ligadas à atividade diária de desenvolvimento. Além disso, administra o

relacionamento com o cliente assegurando que o mesmo participe ativamente do desenvolvimento e forneça as informações essenciais para que a equipe possa implementar o sistema desejado.

Coach

O coach é o responsável técnico do projeto. O XP recomenda que um profissional tecnicamente bem preparado seja destacado para orientar a equipe de modo que ela siga as boas práticas recomendadas pelo XP. Embora também possa atuar na implementação do sistema, sua tarefa principal é assegurar o bom funcionamento do processo e buscar formas de melhorá-lo continuamente.

Analista de Teste

O analista de teste é responsável por ajudar o cliente a escrever os testes de aceitação. Quando estes testes não são automatizados, o analista de teste deve fazer com que eles sejam executados diversas vezes ao longo das iterações do projeto. Ele procura fazer com que os eventuais defeitos do sistema sejam identificados tão logo apareçam. Desta forma, fornece feedback para a equipe rapidamente, de modo que ela possa fazer as correções com rapidez e evitar que os problemas se propaguem.

Redator Técnico

O redator técnico ajuda a equipe de desenvolvimento a documentar o sistema. A sua presença permite que os desenvolvedores se concentrem prioritariamente na implementação do software. Embora eles possam continuar fazendo algumas documentações, o redator técnico é quem faz a maior parte do trabalho de documentação.

Desenvolvedor

O desenvolvedor é a pessoa que analisa, projeta e codifica o sistema. Em suma, é a pessoa que efetivamente constrói o software. Dentro do XP, não existem divisões entre analista, projetista, programador etc. Cada desenvolvedor exerce estes diferentes papéis em diversos momentos do projeto.