

# **Extreme Programming (XP): Pontos Fortes e Fracos da Metodologia Ágil**

Bruna Mendonça, Bruno Gazineu, Diego Gazineu, Elisa Antoniazzi, Luan Nordio, Thacielle Santos

## **Resumo:**

A metodologia Extreme Programming (XP) é uma abordagem ágil que visa aumentar a qualidade do software e a capacidade de resposta da equipe às mudanças nos requisitos dos clientes. Implementada oficialmente na década de 1990, XP se distingue por suas práticas rigorosas como Programação em Par, Desenvolvimento Orientado a Testes (TDD) e Integração Contínua. Seus pontos fortes incluem a alta qualidade do código, a satisfação do cliente e a redução de riscos, enquanto os pontos fracos abrangem a dificuldade de implementação inicial, a necessidade de disciplina rigorosa e o investimento em capacitação da equipe. Embora desafiante, a XP oferece uma estrutura valiosa para equipes que buscam um desenvolvimento de software ágil e eficiente.

## **Introdução**

Os métodos ágeis são considerados uma forma de gestão de projetos, que segundo Pressman (2011), surgiram a partir da necessidade de solucionar alguns problemas dos métodos de desenvolvimento tradicionais. O mundo tem apresentado mudanças consideráveis em pequenos períodos, e esse movimento impacta as necessidades dos usuários. Acompanhando essa movimentação, estão as empresas com o desenvolvimento dos seus projetos lidando com grandes dificuldades devido aos longos escopos, e então torna-se necessária a agilidade para se adaptar.

Nesse cenário cada vez mais repleto de constantes mudanças, as metodologias ágeis ganham espaço em comparação às metodologias tradicionais. A flexibilidade e rápida adaptação são características importantes que contribuíram para essa evolução (PASCUTTI et al., 2019). A implementação de metodologias com tais características tornaram-se indispensáveis, e hoje são requisitos básicos para quem busca sucesso quando falamos em produtividade. Atualmente, entre as mais conhecidas do mercado estão a Extreme Programming (XP) e Scrum, ambas baseadas em princípios ágeis.

A metodologia Scrum busca através das suas práticas, uma gestão otimizada através de curtos ciclos de trabalho iterativos, conhecidos como "sprints". Destaca a

auto-organização das equipes permitindo autonomia para resolução de problemas. Por outro lado, a metodologia Extreme Programming (XP) tem o foco em aumentar a qualidade do software e a capacidade da equipe em ter respostas rápidas às mudanças, resultados que são obtidos através de um rigoroso conjunto de práticas de desenvolvimento. Outro ponto em destaque é a forte comunicação com o cliente (SBROCCO & MACEDO, 2012).

### **O que é a Metodologia XP?**

Considerada uma das metodologias de desenvolvimento de software mais conhecidas do mercado, entre as que baseiam-se nos princípios ágeis, a Extreme Programming (XP) foi implementada oficialmente no final da década de 1990 por Kent Beck. Este marco, que ocorreu especificamente no ano de 1996, enquanto Kent Beck fazia a análise do desempenho de um projeto chamado C3 da empresa Chrysler, foi resultado de uma década de pesquisas ao lado de Ward Cunningham enquanto trabalhavam na Tektronix, Inc. Esta metodologia foi a solução para resolver o grande desafio encontrado na empresa Chrysler, onde seria necessário trabalhar na unificação de quatro sistemas legado sobre um controle de folha de pagamentos de aproximadamente 86 mil funcionários. (CASTRO, 2007)

A XP tem como pontos centrais o aumento da qualidade do software e uma equipe capaz de responder rapidamente às mudanças de requisitos dos clientes, com algumas características específicas como discutiremos a seguir. Para alcançar esses objetivos, esta metodologia trabalha um conjunto rigoroso de práticas responsáveis por provocar uma comunicação constante, entregas frequentes devido aos curtos períodos de desenvolvimento e feedback rápido para caso necessário, fazer adaptações em relação às modificações dos requisitos por parte do cliente (ASTELS, 2002).

A trajetória de Kent Beck na Tektronix e durante a análise do C3 na Chrysler apresentavam exatamente o contexto de pequenos prazos de entrega, modificações frequentes nos requisitos que já haviam sido definidos, além da exigência e necessidade sobre a alta qualidade do software (BECK, 2000).

A XP é composta por alguns princípios considerados fundamentais para a metodologia que segue a linha ágil, entre eles encontramos o respeito, coragem, simplicidade, feedback e a comunicação. Neste último princípio, a comunicação, podemos considerar como um dos mais importantes, devido a necessidade de existir um alinhamento entre os

desenvolvedores, gerentes e clientes. Qualquer modificação deve ser comunicada o mais breve possível para que não impacte no andamento do projeto e para isso é comum que este princípio ganhe destaque. O feedback deve ser constante para garantir o desenvolvimento de um software que esteja alinhado às expectativas dos clientes, que são traduzidas em requisitos. A simplicidade destaca a necessidade de um caminho otimizado, com a menor quantidade de empecilhos que possam prejudicar o sucesso das sprints, mas ao mesmo tempo que atendam aos requisitos do cliente. Respeito, um dos princípios fundamentais para uma equipe equilibrada e saudável, onde para ter o desenvolvimento de um software de alta qualidade, é necessário também ter o respeito e reconhecimento do trabalho de todos os membros que trabalham nesta construção. A coragem traz em destaque a importância do preparo de uma equipe para passar por mudanças necessárias que contribuam para a melhorar constantemente o código (BECK & ANDRES, 2004).

A XP também é conhecida por suas práticas diferenciadas para o desenvolvimento de software, entre elas a Programação em Par, Desenvolvimento Orientado a Testes (TDD), Integração Contínua, Refatoração Contínua, Pequenas Entregas, Propriedade Coletiva do Código, Metáforas, e a Semana de 40 Horas. Na programação em par temos dois programadores que realizam a construção do código juntos utilizando apenas um computador. Enquanto um deles escreve o código, o outro programador pode fazer a revisão em tempo real para possíveis correções, além de sugestões. Esse método proporciona uma troca de conhecimento contínua, além de garantir a qualidade do código que está sendo escrito e uma menor taxa de erro.

O Desenvolvimento Orientado a Testes (TDD, do inglês Test-Driven Development), é um método onde os testes são construídos/escritos antes do início da programação do código. O início é marcado através da criação de um teste automatizado, responsável por definir uma nova funcionalidade. A primeira vez que o teste rodar, vai falhar devido a ausência da melhoria, após, o código é construído de forma que passe naquele teste previamente montado antes de ir para uma refatoração. Desta forma existe a garantia de que cada funcionalidade será testada. Em um cenário de desenvolvimento de software, é comum que o mesmo seja construído por diferentes programadores, por isso a necessidade da Integração Contínua (CI, do inglês Continuous Integration), para que seja testado frequentemente. Desta forma permite a rápida detecção de erros e conflitos, o que leva ao aumento da velocidade sobre as correções. Outra prática destacada na XP é a Refatoração Contínua, sendo a prática de

melhoria constante da estrutura interna do código, mas de modo sem alterar seu comportamento em relação às funcionalidades estabelecidas. As Pequenas Entregas referem-se a pequenos períodos de desenvolvimento para as novas melhorias, que são ajustadas para compor as sprints. A Propriedade Coletiva do Código torna possível que qualquer desenvolvedor que faça parte do time, possa modificar qualquer parte. As Metáforas têm como função guiar o desenvolvimento com uma linguagem comum e mais compreensível. A Semana de 40 horas representa o equilíbrio saudável entre a vida pessoal e profissional de um membro da equipe que trabalha com XP (SBROCCO, 2012).

### **Pontos Fortes da Metodologia XP**

Para alcançar o sucesso dos projetos de desenvolvimento de software, encontramos na metodologia XP alguns pontos em destaque, que serão apresentados na sequência, responsáveis por este resultado. Para os pontos fortes da XP iniciamos com a alta qualidade do código que é obtida através de suas práticas. O Desenvolvimento Orientado a Testes (TDD) em conjunto com a Refatoração Contínua, garantem um código constantemente testado e melhorado. Através deste método é possível garantir que cada funcionalidade estabelecida nos requisitos através das reuniões com o cliente, seja entregue conforme a expectativa do mesmo, além de ser considerado um código confiável (MARTIN, 2002).

Outro ponto forte de grande relevância nesta metodologia é a melhoria em relação a satisfação do cliente. São definidos prazos curtos de desenvolvimento para garantir entregas frequentes, onde os clientes passam a visualizar de uma forma mais efetiva o andamento do projeto, as etapas concluídas e o que ainda está em aberto. Desta forma é possível manter um feedback constante para rápidos ajustes nos requisitos caso seja necessário. Esse processo contribui para a construção de um produto que atenda às reais necessidades do cliente (FOWLER, 2001).

Comunicação efetiva e limpa, colaboração entre a equipe são resultados positivos encontrados através das práticas da XP. Propriedade coletiva do código e programação em par contribuem para estes aspectos como a troca de conhecimento contínua (WILLIAMS & KESSLER, 2003).

A redução de riscos é consequência positiva de outra prática encontrada na XP, a Integração Contínua. Em uma equipe de desenvolvimento composta por diferentes programadores é fundamental esta prática para garantir um código com alta frequência de testes, permitindo a identificação e correção precoce de erros antes que se tornem graves e

prejudiquem o cliente. Essa prática em conjunto com as pequenas entregas reduzem consideravelmente os riscos, o que torna o código e processo de desenvolvimento mais confiantes na visão do cliente (DUVALL, MATYAS & GLOVER, 2007).

O aumento da produtividade das equipes de desenvolvimento é evidente quando implementada a metodologia XP. Com a Semana de 40 Horas é possível obter para o membro da equipe, uma vida equilibrada entre vida pessoal e profissional, mantendo-o produtivo.

O Planejamento do Jogo é a prática responsável por promover a colaboração entre desenvolvedores e clientes para definir os requisitos do projeto e definir as prioridades (BECK, 2000).

Todas as práticas mencionadas levam a equipe de desenvolvimento para um caminho mais ágil e que se adapta quando necessário seguindo as necessidades do cliente, que são destacadas através da comunicação, que é muito trabalhada na XP. (BECK & ANDRES, 2004).

### **Pontos Fracos da Metodologia XP**

Tudo o que é bom também pode ter seu lado negativo, e com a metodologia XP não seria diferente. Apesar de muitas vantagens, tem seus pontos fracos. Uma das grandes dificuldades encontradas é a implementação inicial. Empresas acostumadas com metodologias tradicionais podem sentir dificuldades na adaptação da equipe. A rotina baseada nas novas práticas necessita do ajuste individual de cada membro da equipe nos processos do trabalho. Uma das práticas que apresenta grande resistência é a Programação em Par e TDD, que podem ser desconfortáveis (BECK, 2000).

Para que os benefícios sejam alcançados corretamente, é necessária a disciplina e compromisso de forma rigorosa da equipe, sendo algo que nas metodologias tradicionais não é cobrado de forma tão intensa no dia a dia. A dificuldade está em conseguir manter a equipe alinhada e destacando a importância desses requisitos para o sucesso do desenvolvimento (FOWLER, 2001).

Implementar a metodologia XP pode envolver um grande investimento inicial para capacitar a equipe a executar corretamente suas práticas. A Programação em Par, TDD e Integração contínua, podem exigir um reforço técnico através da capacitação dos membros da

equipe. Um investimento que a longo prazo pode trazer retornos positivos (DUVALL, MATYAS & GLOVER, 2007).

Uma das práticas peculiares da XP é a Programação em Par. Além de ser desconfortável para alguns desenvolvedores e apresentar resistência sobre sua aderência, a dificuldade pode ir além. Empresas buscam otimizar seus recursos para o desenvolvimento dos produtos, e no caso desta prática, o meio corporativo pode ver como uma redundância e gasto desnecessário, logo que duas pessoas estariam trabalhando na mesma tarefa (WILLIAMS & KESSLER, 2003).

Um dos grandes desafios a ser destacado na metodologia baseada nos princípios ágeis é o equilíbrio entre a qualidade do software e a velocidade exigida. As entregas em curto prazo, integração contínua e a constante comunicação com o cliente podem facilmente gerar uma pressão na equipe, onde caso não seja bem gerenciada pode comprometer os processos e práticas, e consequentemente a qualidade do produto e até das relações na equipe (MARTIN, 2002).

## **Conclusão**

A Extreme Programming (XP) é uma das metodologias baseadas nos princípios ágeis considerada eficaz no processo de desenvolvimento de software. Considerada um dos destaques devido suas práticas específicas que contribuem para alcançar um software de alta qualidade em um processo apto a receber adaptações quando necessário. A redução dos riscos no processo de desenvolvimento, ênfase na satisfação do cliente, alta qualidade do código e forte comunicação, são indicadores do potencial dessa metodologia na construção de um produto que realmente atenda as necessidades do cliente de forma rápida, objetiva e com qualidade.

Mesmo sendo considerada uma opção atrativa para empresas que buscam resultados em curto espaço de tempo, a implementação da XP contém alguns desafios relacionados à sua implementação inicial. O custo inicial, grande comprometimento da equipe e adaptação relacionada às novas práticas podem dificultar o processo. E para finalizar, o grande desafio sobre manter o equilíbrio entre qualidade e velocidade exigida.

Apesar das limitações da XP, é uma metodologia com uma estrutura que pode agregar muito em equipes comprometidas a melhorar seus processos. Com a devida disciplina,

através dela pode ser realizada uma transformação nos processos de desenvolvimento de software.

## **Referências**

**PRESSMAN**, Roger S. Engenharia de Software. 6. ed. São Paulo: McGraw-Hill, 2006.

**PASCUTTI**, Márcia Cristina Dadalto et al. Engenharia de Software. Maringá-Pr.: Unicesumar, 2019.

**SBROCCO**, José Henrique Teixeira de Carvalho; **MACEDO**, Paulo Cesar. Metodologias Ágeis: Engenharia de Software sob medida. São Paulo: Erica, 2012.

**CASTRO**, Vinicius A. Desenvolvimento Ágil com Programação Extrema. Monografia – Curso de Ciência da Computação, Universidade Federal de Sergipe, São Cristóvão, Sergipe, 2007.

**ASTELS**, David; **MILLER**, Granville; **NOVAK**, Miroslav. Extreme Programming – Guia prático. Rio de Janeiro, Ed. Campos, 2002.

**BECK**, K. (2000). Extreme Programming Explained: Embrace Change. Addison-Wesley.

**BECK**, K., & **ANDRES**, C. (2004). Extreme Programming Explained: Embrace Change (2nd ed.). Addison-Wesley.

**MARTIN**, R. C. (2002). Agile Software Development, Principles, Patterns, and Practices. Prentice Hall.

**WILLIAMS**, L., & **KESSLER**, R. (2003). Pair Programming Illuminated. Addison-Wesley.

**DUVALL**, P. M., **MATYAS**, S., & **GLOVER**, A. (2007). Continuous Integration: Improving Software Quality and Reducing Risk. Addison-Wesley.

**FOWLER**, M. (2001). Refactoring: Improving the Design of Existing Code. Addison-Wesley.

**WILLIAMS**, L., & **KESSLER**, R. (2003). Pair Programming Illuminated. Addison-Wesley.