




Artigo

Conceitos Básicos sobre Metodologias Ágeis

Este artigo tem como objetivo fazer uma apresentação conceitual sobre uma das mais conhecidas Metodologias Ágeis para Desenvolvimento de Software, a Extreme Programming, apresentando vantagens e desvantagens do uso em relação à Metodologias Clássicas

Marcar como concluído Anotar 

Artigos



Canal Mais



Conceitos Básicos sobre Metodologias Ágeis

Metodologias Clássicas (Tradicionais)

Também conhecidas como Metodologias orientadas a planejamento, as Metodologias Clássicas dominaram a forma de desenvolvimento de softwares até o início da década de 90, Entretanto, estas metodologias devem ser aplicadas apenas em situações em que os requisitos do sistema são estáveis e os requisitos



recursos para processos pesados de produção de software. Por esta razão, as organizações pequenas acabam por não usar nenhum processo. Isto pode trazer



dificultar a entrega do software nos prazos, custos e funcionalidades previamente definidas.

[Saiba mais](#)[Série eXtreme Programming na prática](#)

Metodologias Ágeis e o Manifesto Ágil

A expressão “Metodologias Ágeis” tornou-se conhecida em 2001, quando especialistas em processos de desenvolvimento de software representando entre outros, os métodos Scrum e Extreme Programming (XP), foram estabelecidos princípios e características comuns destes métodos. Assim foi criada a “Aliança Ágil” e efetuou-se o estabelecimento do “Manifesto Ágil”.

Principais conceitos do Manifesto Ágil

- Pessoas e interações, ao contrário de processos e ferramentas.
- Software executável, ao contrário de documentação extensa e confusa.
- Colaboração do cliente, ao contrário de constantes negociações de contratos.
- Respostas rápidas para as mudanças, ao contrário de seguir planos previamente definidos.



modificam rapidamente. Entre as principais diferenças da XP em relação às Metodologias Clássicas estão o feedback constante, a abordagem incremental e o



A maioria das regras da XP causa surpresa no primeiro contato e muitas não fazem sentido se aplicadas isoladamente. É a força de seu conjunto que sustenta o sucesso da XP, trazendo uma verdadeira revolução no desenvolvimento de software.

O principal objetivo da XP é dar agilidade ao desenvolvimento do projeto e busca garantir a satisfação do cliente. As práticas, regras, e os valores da XP garantem um agradável ambiente de desenvolvimento de software para os seus seguidores, que são conduzidos por quatro princípios básicos:

- **Princípio da Comunicação** - Busca manter o melhor relacionamento possível entre clientes e desenvolvedores, preferindo conversas pessoais a outros meios de comunicação.
- **Princípio da Simplicidade** - Entende-se como simplicidade, a busca do objetivo de implementar o software com o menor número possível de classes e métodos. Outra ideia importante deste princípio é procurar implementar apenas requisitos atuais, evitando assim adicionar funcionalidades que podem ser importantes apenas no futuro. A aposta da XP é que é melhor fazer algo simples hoje do que implementar algo complicado hoje que talvez não venha a ser usado.
- **Princípio do Feedback** - A prática do feedback constante significa que o desenvolvedor terá informações constantes do código e do cliente. A informação do código é dada pelos testes constantes, que indicam os erros tanto individuais quanto do software integrado.



princípio também dá suporte à simplicidade, pois assim que a oportunidade de simplificar o software é percebida, a equipe pode experimentar e buscar



constantemente um feedback do cliente.

Principais práticas da Extreme Programming (XP):

- Planejamento - Define o que é ou não necessário ser feito no projeto. A XP baseia-se em requisitos atuais para desenvolvimento de software, não em requisitos futuros.
- Entregas Frequentes - Baseiam-se no desenvolvimento de um software simples, e conforme os requisitos aparecem, há a atualização da versão do software. Cada versão entregue deve ter o menor tamanho possível, contendo os requisitos de maior valor para o negócio. É recomendado que as versões devem ser entregues a cada mês, ou no máximo a cada dois meses, aumentando a possibilidade de feedback rápido do cliente.
- Metáfora - São as descrições de um software sem a utilização de termos técnicos com o objetivo de guiar o desenvolvimento do software com a maior transparência possível para o cliente.
- 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. Eventuais requisitos futuros devem ser adicionados assim que eles realmente existirem.
- Testes - A Extreme Programming (XP) prioriza a validação do projeto durante todo o processo de desenvolvimento. Os desenvolvedores implementam o software criando primeiramente os testes.
- Programação em pares - A implementação do código é feita em dupla, ou



e devem ser alterados sempre que possível.

- Refatoração - Focaliza a lapidação do projeto do software e está presente em



possível, buscando principalmente simplificar o código atual sem perder nenhuma funcionalidade.

- Propriedade coletiva - O código do projeto pertence a todos os membros da equipe. Isto significa que qualquer pessoa que percebe que pode adicionar valor a um código, mesmo que ele próprio não o tenha desenvolvido, pode fazê-lo, desde que faça os testes necessários e não prejudique as funcionalidades atuais. Isto é possível porque na XP todos são responsáveis pelo software. Uma grande vantagem desta prática é que, caso um membro da equipe deixe o projeto antes do fim, a equipe consegue continuar o projeto sem grandes dificuldades, pois todos conhecem todas as partes do software, mesmo que não seja de forma detalhada.
- 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. Esta prática é facilitada com o uso de apenas uma máquina de integração, que deve ter livre acesso a todos os membros da equipe.
- 40 horas de trabalho semanal - A XP assume que não se deve fazer horas extras constantemente. Caso seja necessário trabalhar mais de 40 horas pela segunda semana consecutiva, existe um problema sério no projeto que deve ser resolvido não com aumento de horas trabalhadas, mas com melhor planejamento.





Por Daniel

Em 2008



Suporte ao aluno - Deixe a sua dúvida.

[Tecnologias](#)[Exercícios](#)[Cursos](#)[Artigos](#)[Revistas](#)[Fale conosco](#)[Trabalhe conosco](#)[Assinatura para empresas](#)



sanar todas as dúvidas de requisitos, evitando atrasos e até mesmo construções erradas. Uma ideia interessante é manter o cliente como parte



- Código padrão - Baseia-se na padronização da arquitetura do código, para que este possa ser compartilhado entre todos os programadores e até mesmo entre outros softwares.

Extreme Programming (XP): Uma metodologia qu...



Saiba mais: [Curso de eXtreme Programming](#)

