“Princípios e práticas de eXtremme Programming”

Tiago Eugenio de Melo
tiago@comunidadesol.org

Sumário

- Introdução
- Princípios
- Práticas
- Quando não usar
- Conclusões
- Referências

eXtreme Programming

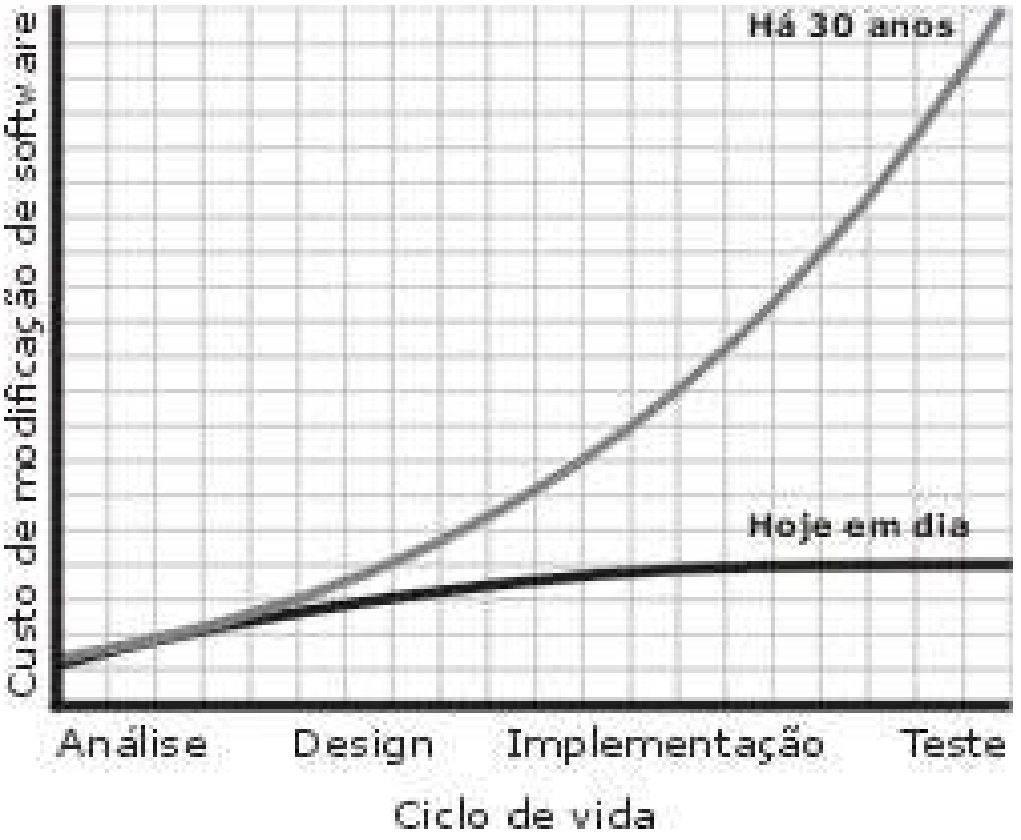
- É uma metodologia ágil para equipes pequenas e médias e que irão desenvolver software com requisitos vagos e em constante mudança.
- A codificação é a atividade principal.
- Ênfase
  - Menor em processos formais de desenvolvimento.
  - Maior na disciplina rigorosa.

eXtreme Programming

- Idéia
  - Baseia-se na revisão permanente do código, testes freqüentes, participação do usuário final, refatoramento contínuo, integração contínua, planejamento, design e redesign a qualquer hora.
- Usuários
  - Ford.
  - Borland.
  - BMW.
  - Symantec.

eXtreme Programming

- Ciclo de vida



eXtreme Programming

- Princípios
  - Comunicação
  - Simplicidade
  - Feedback
  - Coragem

eXtreme Programming

- Comunicação
  - Várias práticas do XP promovem uma maior comunicação entre os membros da equipe.
  - A comunicação não é limitada por procedimentos formais. Usa-se o melhor meio possível, por exemplo:
    - Uma conversa ou reunião informal.
    - E-mail, bate-papo ou telefonema.
  - Preferência à comunicação mais ágil
    - Telefonema é melhor do que e-mail.
    - Presença física é melhor do que remota.
  - O XP utiliza a comunicação com as respostas mais rápidas possíveis.

eXtreme Programming

- Simplicidade
  - O XP incentiva as práticas que reduzem a complexidade do sistema.
  - A solução adotada deve ser sempre a mais simples para se alcançar os objetivos esperados:
    - Usar as tecnologias, design, algoritmos e técnicas mais simples que permitirão atender aos requisitos do usuário final.
    - Design, processo e código podem ser simplificados a qualquer momento.
    - Qualquer design, processo ou código criado pensando nas iterações futuras deve ser descartado.

eXtreme Programming

- Feedback
  - Várias práticas do XP garantem um rápido feedback sobre as várias etapas/fases do processo.
    - Feedback sobre qualidade do código (testes de unidade, programação em pares, posse coletiva).
    - Feedback sobre estado do desenvolvimento (estórias do usuário final, integração contínua, jogo do planejamento).
  - Permite maior agilidade
    - Erros detectados e corrigidos imediatamente.
    - Requisitos e prazos reavaliados mais cedo.
    - Facilita a tomada de decisão.
    - Permite estimativas mais precisas.
    - Maior segurança e menos riscos para investidores.

eXtreme Programming

- Coragem
  - Testes, integração contínua, programação em pares e outras práticas do XP aumentam a confiança do programador e ajudam-no a ter coragem para:
    - Melhorar o design de código que está funcionando para torná-lo mais simples.
    - Jogar fora o código desnecessário.
    - Investir tempo no desenvolvimento de testes.
    - Mexer no design em estágio avançado do projeto.
    - Pedir ajudar aos que sabem mais.
    - Dizer ao cliente que um requisito não vai ser implementado no prazo prometido.
    - Abandonar processos formais e fazer design e documentação em forma de código.

Práticas de XP

- A equipe
  - Todos em um projeto XP são parte de uma equipe.
  - Esta equipe deve incluir um representante do cliente, que:
    - estabelece os requisitos do projeto.
    - define as prioridades.
    - controla o rumo do projeto.
  - O representante é o usuário final e que conhece o domínio do problema e das suas necessidades.

Práticas de XP

- Jogo de planejamento (planning game)
  - Prática XP na qual se define
    - estimativas de prazo para cada tarefa.
    - as prioridades: quais as tarefas mais importantes.
  - Dois passos chave:
    - Planejamento de um release
      - Cliente propõe funcionalidade desejadas (estórias).
      - Programadores avaliam a dificuldade de implementá-las.
    - Planejamento de uma iteração (até 3 semanas)
      - Cliente define as funcionalidades prioritárias para a iteração.
      - Programadores as quebram em tarefas e avaliam o seu custo.
  - Ótimo feedback para que o cliente possa dirigir o projeto
    - É possível ter uma idéia clara do avanço do projeto.
    - Clareza reduz riscos, aumenta a chance de sucesso.

Práticas de XP

- Testes de aceitação
  - No jogo de planejamento, o usuário-cliente elabora estórias que descrevem cada funcionalidade desejada. Programador as implementa.
    - Cada estória deve ser entendida suficientemente bem para que programadores possam estimar sua dificuldade.
    - Cada estória deve ser testável.
  - Testes de aceitação são elaborados pelo cliente
    - São testes automáticos.
    - Quando rodarem com sucesso,a funcionalidade foi implementada.
    - Devem ser rodados novamente em cada iteração futura.
    - Oferecem feedback: pode-se saber, a qualquer momento, quantos % do sistema já foi implementado e quanto falta.

Práticas de XP

- Pequenos lançamentos (small releases)
  - Disponibiliza, a cada iteração, software 100% funcional.
    - Software disponível imediatamente.
    - Menor risco (se o projeto não terminar, parte existe e funciona).
    - Cliente pode medir com precisão o quanto já foi feito.
    - Feedback do cliente permitirá que problemas sejam detectados cedo e facilitará a comunicação entre o cliente e o desenvolvedor.
  - Cada lançamento possui funcionalidades prioritárias.
    - Valores de negócio implementados foram escolhidos pelo cliente.
  - Lançamento pode ser destinado a:
    - usuário-cliente (que pode testá-lo, avaliá-lo e oferecer feedback).
    - usuário-final (sempre que possível).
  - Design simples e integração contínua são práticas essenciais para viabilizar pequenos e freqüêntes lançamentos.

Práticas de XP

- Design simples
  - O design está presente em todas as etapas do XP
    - O projeto começa simples e se mantém simples através de testes e refinamento do design (refatoramento).
  - Todos buscamos design simples e claro. Em XP, levamos isto a níveis extremos.
    - Não permitimos que se implemente nenhuma função adicional que não será usada na atual iteração.
  - Implementação ideal é aquela que:
    - Roda todos os testes.
    - Expressa todas as idéias que você deseja expressar.
    - Não contém código duplicado.
    - Tem o mínimo de classes e métodos.
  - O que não é necessário AGORA não deve ser implementado.

Práticas de XP

- Programação em duplas
  - Todo o desenvolvimento em XP é feito em pares
    - Um computador, um teclado e dois programadores.
    - Um piloto, um co-piloto.
    - Papéis são alternados freqüentemente.
    - Pares são trocados periodicamente.
  - Benefícios
    - Melhor qualidade do design, código e testes.
    - Revisão constante do código.
    - Nivelamento da equipe.
    - Maior comunicação.

Práticas de XP

- Testes
  - O seu desenvolvimento é guiado por testes
  - Test first, then code
    - Os testes puxam o desenvolvimento.
    - Os programadores XP escrevem testes primeiro, escrevem código e rodam testes para validar o código escrito.
    - Cada unidade de código só temvalor se seu teste funcionar 100%.
    - Todos os testes são executados automaticamente, o tempo todo.
    - Testes são a documentação executável do sistema.
  - Testes dão maior segurança: coragem para mudar
    - Que adianta a OO isolar a interface da implementação se o programador tem medo de mudar a implementação?
    - Código testado é mais confiável.
    - Código testado pode ser alterado sem medo.

Práticas de XP

- Refinamento do design (refatoramento)
  - Não existe uma etapa isolada de design em XP
    - O código é o design.
  - O design é melhorado continuamente através de refatoramento.
  - O refatoramento é um processo formal realizado através de etapas reversíveis
    - Passos de refatoramento melhoram, incrementalmente, a estrutura do código, sem alterar sua função.
    - Existência prévia de testes é essencial (elimina o medo de que o sistema irá deixar de funcionar por causa da mudança).

Práticas de XP

- Integração contínua
  - Projetos XP mantêm o sistema integrado o tempo todo
    - Integração de todo o sistema pode ocorrer várias vezes ao dia.
    - Todos os testes (unidade e integração) devem ser executados.
  - Integração contínua reduz o tempo passado no inferno da integração.
  - Benefícios
    - Expõe o estado atual do desenvolvimento (viabiliza lançamentos pequenos e freqüentes).
    - Estimula design simples, tarefas curtas, agilidade.
    - Oferece feedback sobre todo o sistema.
    - Permite encontrar problemas de design rapidamente.

Práticas de XP

- Posse coletiva
  - Em um projeto XP, qualquer dupla de programadores pode melhorar o sistema a qualquer momento.
  - Todo o código em XP pertence a um único dono: a equipe
    - Todo o código recebe a atenção de todos os participantes resultando em maior comunicação.
    - Maior qualidade (menos duplicação, maior coesão).
    - Menos riscos e menos dependência de indivíduos.
  - Todos compartilham a responsabilidade pelas alterações.
  - Testes e integração contínua são essenciais e dão segurança aos desenvolvedores.
  - Programação em pares reduz o risco de danos.

Práticas de XP

- Padrões de codificação
  - O código escrito em projetos XP segue um padrão de codificação, definido pela equipe
    - Padrão para nomes de métodos, classes, variáveis.
    - Organização do código.
  - Todo o código parece que foi escrito por um único indivíduo, competente e organizado.
  - Código com estrutura familiar facilita e estimula
    - Posse coletiva
    - Comunicação mais eficiente
    - Simplicidade
    - Programação em pares
    - Refinamento do design

Práticas de XP

- Metáfora
  - Equipes XP mantêm uma visão compartilhada do funcionamento do sistema.
    - Pode ser uma analogia com algum outro sistema que facilite a comunicação entre os membros da equipe e cliente.
  - Facilita a escolha dos nomes de métodos, classes, campos de dados etc
    - Serve de base para estabelecimento de padrões de codificação.

Práticas de XP

- Ritmo saudável
  - Projetos XP estão na arena para ganhar
    - Entregar software da melhor qualidade.
    - Obter a maior produtividade dos programadores.
    - Obter a satisfação do cliente.
  - Projetos com cronogramas apertados que sugam todas as energias dos programadores não são projetos XP
    - Semanas de 80 horas levam à baixa produtividade.
    - Produtividade baixa leva a código ruim, relaxamento da disciplina, dificulta comunicação, aumenta a irritação e o stress da equipe.
    - Tempo ganho será perdido depois.
  - Projeto deve ter ritmo sustentável por prazos longos.
    - Eventuais horas extras são aceitáveis quando produtividade é maximizada no longo prazo.

Práticas de XP

- Vencer barreiras culturais
  - Deixar alguém mexer no seu código.
  - Trabalhar em pares
    - Ter coragem de admitir que não sabe.
    - Pedir ajuda.
  - Vencer hábitos antigos
    - Manter as coisas simples.
    - Jogar fora código desnecessário.
    - Escrever testes antes de codificar.
    - Refatorar com freqüência (vencer o medo).

Quando não usar XP

- Equipes grandes e espalhadas geograficamente
  - Comunicação é um valor fundamental do XP.
  - Não é fácil garantir o nível de comunicação requerido em projetos XP em grandes equipes.
- Situações onde não se tem controle sobre o código
  - Código legado que não pode ser modificado.
- Situações onde o feedback é demorado
  - compile-link-build-test que leva 24 horas.
  - Testes são muito difíceis, arriscados e que levam muito tempo.
  - Programadores espalhados em ambientes físicos distantes e sem comunicação eficiente.

Conclusões de XP

- eXtreme Programming (XP) é uma metodologia de desenvolvimento de software baseada nos valores simplicidade, comunicação, feedback e coragem.
- Para implementar XP não é preciso usar diagramas ou processos formais. É preciso fazer uma equipe se unir em torno de algumas práticas simples, obter feedback suficiente e ajustar as práticas para a sua situação particular.
- XP pode ser implementada aos poucos, porém a maior parte das práticas são essenciais.
- Nem todos os projetos são bons candidatos a usar uma metodologia ágil como XP.
- XP é mais adequado a equipes pequenas e médias.

Referências

- Internacionais:
  - http://www.extremeprogramming.org/
  - http://www.xprogramming.com/
  - http://www.pairprogramming.com/
  - http://www.xp123.com/
- Nacionais:
  - http://www.xispe.com.br/
- Download da Palestra:
  - http://www.tiagodemelo.info