

Extremme Programming

`http://ideia.me | jonatasdp@gmail.com`

7 de maio de 2010

1 Princípios

2 Valores

3 Práticas

Princípios de uma equipe

Princípios de uma equipe

Oportunidade Como é encarada cada situação de um projeto

Princípios de uma equipe

Oportunidade Como é encarada cada situação de um projeto

Diversidade Quantos tipos de pessoas contribuem para o projeto?

Princípios de uma equipe

Oportunidade Como é encarada cada situação de um projeto

Diversidade Quantos tipos de pessoas contribuem para o projeto?

Passos de Bebê Uma coisa de cada vez, em pequenos passos

Princípios de uma equipe

Oportunidade Como é encarada cada situação de um projeto

Diversidade Quantos tipos de pessoas contribuem para o projeto?

Passos de Bebê Uma coisa de cada vez, em pequenos passos

Auto-semelhança boas práticas devem ser replicadas

Princípios de uma equipe

Oportunidade Como é encarada cada situação de um projeto

Diversidade Quantos tipos de pessoas contribuem para o projeto?

Passos de Bebê Uma coisa de cada vez, em pequenos passos

Auto-semelhança boas práticas devem ser replicadas

Benefício Mútuo Programador feliz + Cliente feliz + Gerente feliz

Princípios de uma equipe

Oportunidade Como é encarada cada situação de um projeto

Diversidade Quantos tipos de pessoas contribuem para o projeto?

Passos de Bebê Uma coisa de cada vez, em pequenos passos

Auto-semelhança boas práticas devem ser replicadas

Benefício Mútuo Programador feliz + Cliente feliz + Gerente feliz

Economia O que gera mais retorno financeiro para o cliente?

Princípios de uma equipe

Oportunidade Como é encarada cada situação de um projeto

Diversidade Quantos tipos de pessoas contribuem para o projeto?

Passos de Bebê Uma coisa de cada vez, em pequenos passos

Auto-semelhança boas práticas devem ser replicadas

Benefício Mútuo Programador feliz + Cliente feliz + Gerente feliz

Economia O que gera mais retorno financeiro para o cliente?

Falha Experimente, descubra, tente, falhe!

Princípios de uma equipe

Oportunidade Como é encarada cada situação de um projeto

Diversidade Quantos tipos de pessoas contribuem para o projeto?

Passos de Bebê Uma coisa de cada vez, em pequenos passos

Auto-semelhança boas práticas devem ser replicadas

Benefício Mútuo Programador feliz + Cliente feliz + Gerente feliz

Economia O que gera mais retorno financeiro para o cliente?

Falha Experimente, descubra, tente, falhe!

Fluidez Software não se faz em fábricas

Princípios de uma equipe

- Oportunidade** Como é encarada cada situação de um projeto
- Diversidade** Quantos tipos de pessoas contribuem para o projeto?
- Passos de Bebê** Uma coisa de cada vez, em pequenos passos
- Auto-semelhança** boas práticas devem ser replicadas
- Benefício Mútuo** Programador feliz + Cliente feliz + Gerente feliz
- Economia** O que gera mais retorno financeiro para o cliente?
 - Falha** Experimente, descubra, tente, falhe!
 - Fluidez** Software não se faz em fábricas
 - Metáfora** Você sabe o que é Lixeira e Janela no computador?

Princípios de uma equipe

- Oportunidade** Como é encarada cada situação de um projeto
- Diversidade** Quantos tipos de pessoas contribuem para o projeto?
- Passos de Bebê** Uma coisa de cada vez, em pequenos passos
- Auto-semelhança** boas práticas devem ser replicadas
- Benefício Mútuo** Programador feliz + Cliente feliz + Gerente feliz
 - Economia** O que gera mais retorno financeiro para o cliente?
 - Falha** Experimente, descubra, tente, falhe!
 - Fluidez** Software não se faz em fábricas
 - Metáfora** Você sabe o que é Lixeira e Janela no computador?
- Humanismo** Programador também têm sentimentos

Princípios de uma equipe

- Oportunidade** Como é encarada cada situação de um projeto
- Diversidade** Quantos tipos de pessoas contribuem para o projeto?
- Passos de Bebê** Uma coisa de cada vez, em pequenos passos
- Auto-semelhança** boas práticas devem ser replicadas
- Benefício Mútuo** Programador feliz + Cliente feliz + Gerente feliz
 - Economia** O que gera mais retorno financeiro para o cliente?
 - Falha** Experimente, descubra, tente, falhe!
 - Fluidez** Software não se faz em fábricas
 - Metáfora** Você sabe o que é Lixeira e Janela no computador?
- Humanismo** Programador também têm sentimentos
- Melhoria** Software estraga

Princípios de uma equipe

- Oportunidade** Como é encarada cada situação de um projeto
- Diversidade** Quantos tipos de pessoas contribuem para o projeto?
- Passos de Bebê** Uma coisa de cada vez, em pequenos passos
- Auto-semelhança** boas práticas devem ser replicadas
- Benefício Mútuo** Programador feliz + Cliente feliz + Gerente feliz
 - Economia** O que gera mais retorno financeiro para o cliente?
 - Falha** Experimente, descubra, tente, falhe!
 - Fluidez** Software não se faz em fábricas
 - Metáfora** Você sabe o que é Lixeira e Janela no computador?
- Humanismo** Programador também têm sentimentos
 - Melhoria** Software estraga
 - Qualidade** Quem não gosta?

Princípios de uma equipe

- Oportunidade** Como é encarada cada situação de um projeto
- Diversidade** Quantos tipos de pessoas contribuem para o projeto?
- Passos de Bebê** Uma coisa de cada vez, em pequenos passos
- Auto-semelhança** boas práticas devem ser replicadas
- Benefício Mútuo** Programador feliz + Cliente feliz + Gerente feliz
 - Economia** O que gera mais retorno financeiro para o cliente?
 - Falha** Experimente, descubra, tente, falhe!
 - Fluidez** Software não se faz em fábricas
 - Metáfora** Você sabe o que é Lixeira e Janela no computador?
- Humanismo** Programador também têm sentimentos
 - Melhoria** Software estraga
 - Qualidade** Quem não gosta?
 - Reflexão** Saber por que você está fazendo isso

Princípios de uma equipe

- Oportunidade** Como é encarada cada situação de um projeto
- Diversidade** Quantos tipos de pessoas contribuem para o projeto?
- Passos de Bebê** Uma coisa de cada vez, em pequenos passos
- Auto-semelhança** boas práticas devem ser replicadas
- Benefício Mútuo** Programador feliz + Cliente feliz + Gerente feliz
 - Economia** O que gera mais retorno financeiro para o cliente?
 - Falha** Experimente, descubra, tente, falhe!
 - Fluidez** Software não se faz em fábricas
 - Metáfora** Você sabe o que é Lixeira e Janela no computador?
- Humanismo** Programador também têm sentimentos
 - Melhoria** Software estraga
 - Qualidade** Quem não gosta?
 - Reflexão** Saber por que você está fazendo isso
- Responsabilidade Aceita** Tarefas devem ser aceitas ao invés de atribuídas

Valores mantidos na equipe

Valores mantidos na equipe

Comunicação Fale, escute, converse

Valores mantidos na equipe

Comunicação Fale, escute, converse

Coragem Encare a situação

Valores mantidos na equipe

Comunicação Fale, escute, converse

Coragem Encare a situação

Feedback Receba e dê o seu feedback

Valores mantidos na equipe

Comunicação Fale, escute, converse

Coragem Encare a situação

Feedback Receba e dê o seu feedback

Respeito Respeite as pessoas com quem trabalha

Valores mantidos na equipe

Comunicação Fale, escute, converse

Coragem Encare a situação

Feedback Receba e dê o seu feedback

Respeito Respeite as pessoas com quem trabalha

Simplicidade Seja simples: não tente complicar

Práticas Primárias

Práticas Primárias

Ambiente Informativo Quadro de informações do projeto

Práticas Primárias

Ambiente Informativo Quadro de informações do projeto
Build de Dez Minutos Build em no máximo 10 minutos

Práticas Primárias

- Ambiente Informativo Quadro de informações do projeto
- Build de Dez Minutos Build em no máximo 10 minutos
- Ciclo Semanal Entregue um fragmento de software em 1 semana

Práticas Primárias

- Ambiente Informativo Quadro de informações do projeto
- Build de Dez Minutos Build em no máximo 10 minutos
- Ciclo Semanal Entregue um fragmento de software em 1 semana
- Ciclo Trimestral Entregue uma versão em 3 meses

Práticas Primárias

- Ambiente Informativo Quadro de informações do projeto
- Build de Dez Minutos Build em no máximo 10 minutos
- Ciclo Semanal Entregue um fragmento de software em 1 semana
- Ciclo Trimestral Entregue uma versão em 3 meses
- Desenvolvimento Orientado a Testes Escreva testes, depois programe

Práticas Primárias

- Ambiente Informativo** Quadro de informações do projeto
- Build de Dez Minutos** Build em no máximo 10 minutos
- Ciclo Semanal** Entregue um fragmento de software em 1 semana
- Ciclo Trimestral** Entregue uma versão em 3 meses
- Desenvolvimento Orientado a Testes** Escreva testes, depois programe
- Design Incremental** Crie o código mínimo para suprir a necessidade

Práticas Primárias

Ambiente Informativo Quadro de informações do projeto

Build de Dez Minutos Build em no máximo 10 minutos

Ciclo Semanal Entregue um fragmento de software em 1 semana

Ciclo Trimestral Entregue uma versão em 3 meses

Desenvolvimento Orientado a Testes Escreva testes, depois
programe

Design Incremental Crie o código mínimo para suprir a necessidade

Equipe Integral O cliente faz parte da equipe

Práticas Primárias

Ambiente Informativo Quadro de informações do projeto

Build de Dez Minutos Build em no máximo 10 minutos

Ciclo Semanal Entregue um fragmento de software em 1 semana

Ciclo Trimestral Entregue uma versão em 3 meses

Desenvolvimento Orientado a Testes Escreva testes, depois
programe

Design Incremental Crie o código mínimo para suprir a necessidade

Equipe Integral O cliente faz parte da equipe

Folga Um dia por semana para trabalhar em tarefas técnicas

Práticas Primárias

Ambiente Informativo Quadro de informações do projeto

Build de Dez Minutos Build em no máximo 10 minutos

Ciclo Semanal Entregue um fragmento de software em 1 semana

Ciclo Trimestral Entregue uma versão em 3 meses

Desenvolvimento Orientado a Testes Escreva testes, depois
programe

Design Incremental Crie o código mínimo para suprir a necessidade

Equipe Integral O cliente faz parte da equipe

Folga Um dia por semana para trabalhar em tarefas técnicas

Estórias Cenários de software

Práticas Primárias

Ambiente Informativo Quadro de informações do projeto

Build de Dez Minutos Build em no máximo 10 minutos

Ciclo Semanal Entregue um fragmento de software em 1 semana

Ciclo Trimestral Entregue uma versão em 3 meses

Desenvolvimento Orientado a Testes Escreva testes, depois programe

Design Incremental Crie o código mínimo para suprir a necessidade

Equipe Integral O cliente faz parte da equipe

Folga Um dia por semana para trabalhar em tarefas técnicas

Estórias Cenários de software

Integração Contínua Software atualizado e compartilhado constantemente

Práticas Primárias

Ambiente Informativo Quadro de informações do projeto

Build de Dez Minutos Build em no máximo 10 minutos

Ciclo Semanal Entregue um fragmento de software em 1 semana

Ciclo Trimestral Entregue uma versão em 3 meses

Desenvolvimento Orientado a Testes Escreva testes, depois
programe

Design Incremental Crie o código mínimo para suprir a necessidade

Equipe Integral O cliente faz parte da equipe

Folga Um dia por semana para trabalhar em tarefas técnicas

Estórias Cenários de software

Integração Contínua Software atualizado e compartilhado
constantemente

Programação em Par Piloto e co-piloto

Práticas Primárias

Ambiente Informativo Quadro de informações do projeto

Build de Dez Minutos Build em no máximo 10 minutos

Ciclo Semanal Entregue um fragmento de software em 1 semana

Ciclo Trimestral Entregue uma versão em 3 meses

Desenvolvimento Orientado a Testes Escreva testes, depois
programe

Design Incremental Crie o código mínimo para suprir a necessidade

Equipe Integral O cliente faz parte da equipe

Folga Um dia por semana para trabalhar em tarefas técnicas

Estórias Cenários de software

Integração Contínua Software atualizado e compartilhado
constantemente

Programação em Par Piloto e co-piloto

Trabalho Energizado Trabalhar de forma inteligente

Práticas Corolárias

Práticas Corolárias

Análise da Raiz do Problema Detecção de problemas

Práticas Corolárias

Análise da Raiz do Problema Detecção de problemas
Base de Código Unificada Evite o desperdício de códigos fonte

Práticas Corolárias

Análise da Raiz do Problema Detecção de problemas
Base de Código Unificada Evite o desperdício de códigos fonte
Código Coletivo Todos devem conhecer todos os códigos

Práticas Corolárias

Análise da Raiz do Problema Detecção de problemas
Base de Código Unificada Evite o desperdício de códigos fonte
Código Coletivo Todos devem conhecer todos os códigos
Código e Testes Artefatos permanentes no projeto

Práticas Corolárias

- Análise da Raiz do Problema Detecção de problemas
- Base de Código Unificada Evite o desperdício de códigos fonte
- Código Coletivo Todos devem conhecer todos os códigos
- Código e Testes Artefatos permanentes no projeto
- Continuidade da Equipe Mantenha boas equipes que trabalham juntas

Práticas Corolárias

- Análise da Raiz do Problema Detecção de problemas
- Base de Código Unificada Evite o desperdício de códigos fonte
- Código Coletivo Todos devem conhecer todos os códigos
- Código e Testes Artefatos permanentes no projeto
- Continuidade da Equipe Mantenha boas equipes que trabalham juntas
- Contrato de Escopo Negociável Custo, prazo e escopo não são previsíveis

Práticas Corolárias

- Análise da Raiz do Problema Detecção de problemas
- Base de Código Unificada Evite o desperdício de códigos fonte
- Código Coletivo Todos devem conhecer todos os códigos
- Código e Testes Artefatos permanentes no projeto
- Continuidade da Equipe Mantenha boas equipes que trabalham juntas
- Contrato de Escopo Negociável Custo, prazo e escopo não são previsíveis
- Envolvimento do Cliente Real Usuários finais também dão pitacos no projeto

Práticas Corolárias

- Análise da Raiz do Problema** Detecção de problemas
- Base de Código Unificada** Evite o desperdício de códigos fonte
- Código Coletivo** Todos devem conhecer todos os códigos
- Código e Testes** Artefatos permanentes no projeto
- Continuidade da Equipe** Mantenha boas equipes que trabalham juntas
- Contrato de Escopo Negociável** Custo, prazo e escopo não são previsíveis
- Envolvimento do Cliente Real** Usuários finais também dão pitacos no projeto
- Equipes que Encolhem** A medida que a capacidade de produção aumenta...

Práticas Corolárias

- Análise da Raiz do Problema** Detecção de problemas
- Base de Código Unificada** Evite o desperdício de códigos fonte
- Código Coletivo** Todos devem conhecer todos os códigos
- Código e Testes** Artefatos permanentes no projeto
- Continuidade da Equipe** Mantenha boas equipes que trabalham juntas
- Contrato de Escopo Negociável** Custo, prazo e escopo não são previsíveis
- Envolvimento do Cliente Real** Usuários finais também dão pitacos no projeto
- Equipes que Encolhem** A medida que a capacidade de produção aumenta...
- Implantação Diária** Claro, se você tiver menos que 5 bugs por ano...

Práticas Corolárias

- Análise da Raiz do Problema** Detecção de problemas
- Base de Código Unificada** Evite o desperdício de códigos fonte
- Código Coletivo** Todos devem conhecer todos os códigos
- Código e Testes** Artefatos permanentes no projeto
- Continuidade da Equipe** Mantenha boas equipes que trabalham juntas
- Contrato de Escopo Negociável** Custo, prazo e escopo não são previsíveis
- Envolvimento do Cliente Real** Usuários finais também dão pitacos no projeto
- Equipes que Encolhem** A medida que a capacidade de produção aumenta...
- Implantação Diária** Claro, se você tiver menos que 5 bugs por ano...
- Implantação Incremental** Grande migrações não funcionam

Práticas Corolárias

- Análise da Raiz do Problema** Detecção de problemas
- Base de Código Unificada** Evite o desperdício de códigos fonte
- Código Coletivo** Todos devem conhecer todos os códigos
- Código e Testes** Artefatos permanentes no projeto
- Continuidade da Equipe** Mantenha boas equipes que trabalham juntas
- Contrato de Escopo Negociável** Custo, prazo e escopo não são previsíveis
- Envolvimento do Cliente Real** Usuários finais também dão pitacos no projeto
- Equipes que Encolhem** A medida que a capacidade de produção aumenta...
- Implantação Diária** Claro, se você tiver menos que 5 bugs por ano...
- Implantação Incremental** Grande migrações não funcionam
- Pagar Por Uso** Revolucione os objetivos do seu software

Outras Práticas

Outras Práticas

Reunião em Pé Sem embromation

Outras Práticas

Reunião em Pé Sem embromation

Refatoração Melhoramento contínuo do código

Outras Práticas

Reunião em Pé Sem embromation

Refatoração Melhoramento contínuo do código

Metáfora Aprimore o relacionamento com o cliente

Referências e outros recursos

Site da Improve It - <http://improveit.com.br/xp>

XP - <http://www.extremeprogramming.org>

Meu site: <http://ideia.me>

Apresentação: http://ideia.me/apresentacao_xp.pdf

Em Latex: http://ideia.me/apresentacao_xp.tex