

Modulo I

Método Ágil

XP – Extreme Programming

Prof. Ismael H F Santos

Bibliografia

- *Vinicius Manhaes Teles, **Extreme Programming**, Novatec Editora*
- *Agile Software Development*
- *Scrum and XP from the Trenches*
- *Martin Fowler, **Analysis Patterns - Reusable Object Models**, Addison-Wesley, 1997*
- *Martin Fowler, **Refatoração - Aperfeiçoando o projeto de código existente**, Ed Bookman*

Ementa

- Introdução
- Valores do XP
- Bugs e Tests
- Valores e Princípios do XP
- Praticas do XP

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

3

MA-XP

Introdução



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

4

A Quem se Destina XP?

- Grupos de 2 a 10 programadores
- Projetos de 1 a 36 meses (calendário)
- De 1000 a 250 000 linhas de código
- Papéis:
 - Programadores (foco central)(sem hierarquia)
 - “Treinador” ou “Técnico” (*coach*)
 - “Acompanhador” (*tracker*)
 - Cliente

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

5

E Se Eu Não Me Encaixo Nesses Casos?

- Você ainda pode aprender muito sobre desenvolvimento de software.
- Terá elementos para repensar as suas práticas.
- No início se dizia:
 - “Ou você é 100% eXtremo ou não é eXtremo. Não dá prá ser 80% XP.”
 - *XP is now like teenage sex.*
 - Hoje não é mais necessariamente assim.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

6

Programação eXtrema *XP*

- Metodologia de desenvolvimento de software aperfeiçoada nos últimos 10 anos.
 - Ganhou notoriedade a partir da OOPSLA'2000.
 - Projetos com requisitos muito voláteis
 - Equipes pequenas (até ~10 pessoas)
 - Desenvolvimento incremental
-
- Nomes: **Kent Beck**, Ward Cunningham, and Ron Jeffries em 2000

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

7

Visão Geral de XP

Valores:

- **Simplicidade**
- **Comunicação**
- **Feedback**
- **Coragem**

Princípios:

- Feedback Rápido
 - Assumir Simplicidade
 - Abraçar as Mudanças
 - Trabalho de Qualidade
-
- Ensinar Aprendendo
 - Responsabilidade Aceita
 - Medição Honesta
 - Comunicação Aberta e Honesta
 - **Adaptação Local**

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

8

Práticas de XP

- **Jogo do Planejamento**
- **Pequenas Versões**
- **Projeto Simples**
- **Refatoração**
- **Teste Primeiro**
- **Programação em Pares**
- Metáfora
- Padrões de Codificação
- Propriedade Coletiva do Código
- Integração Contínua
- Semana de 40 horas
- **Testes de Aceitação**
- **Cliente "on-site"**

- **Nível do Cliente**
- **Nível da Equipe**
- **Nível do Par**

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

9

Baseado no gráfico de Ron Jeffries

Variáveis de Projeto – Visão XP

■ **Modelo Tradicional:**

- Tempo
- Escopo
- Custo

- **Manipula-se a Qualidade**

■ **Modelo XP:**

- Tempo
- Custo
- Qualidade

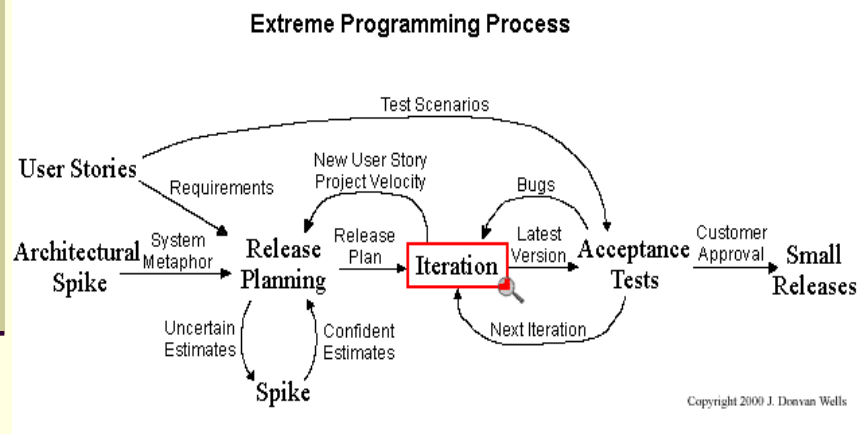
- **Manipula-se o Escopo**

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

10

O Ciclo de Vida de XP



April 05

Prof. Ismael H. Santos - ismael@tecgraf.puc-rio.br - www.extremeprogramming.org

Sobre o Extreme Programming (XP)

XP é o mais importante movimento em nosso campo hoje em dia. Eu estou prevendo que ele será tão essencial para a geração atual quanto o SEI e o Capability Maturity Model (CMM) foram para o passado.

Tom DeMarco (2000)

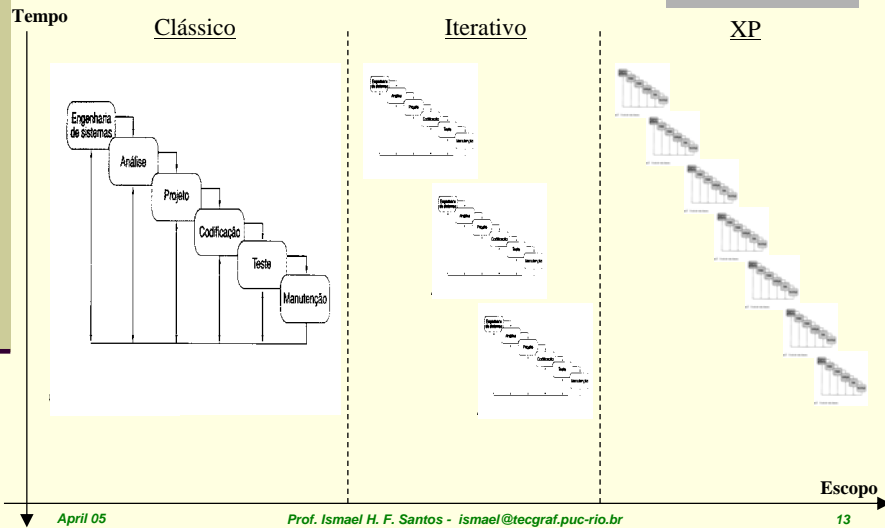


April 05

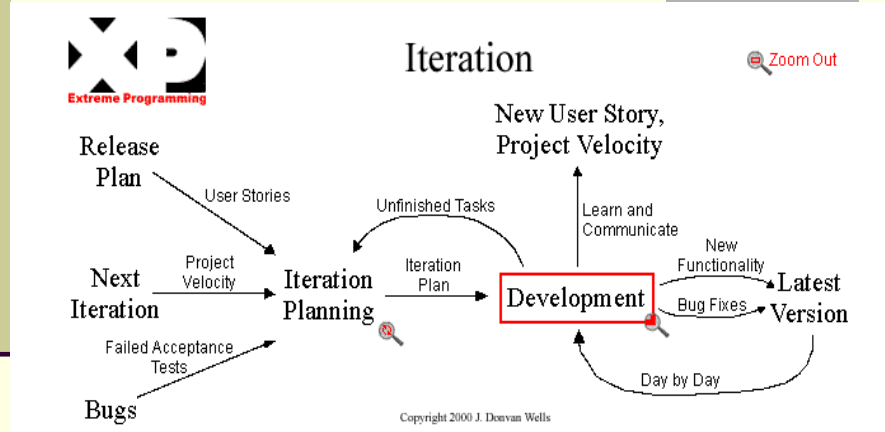
Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

12

Sobre o Extreme Programming (XP)



Sobre o Extreme Programming (XP)



Sobre o Extreme Programming (XP)

- XP não é uma idéia totalmente terminada
- Os limites de sua aplicação ainda não estão bem definidos
- As práticas do método não precisam ser adotadas todas de um vez
- O principal objetivo do XP é reduzir o ciclo de desenvolvimento de alguns anos para alguns dias ou horas...

Sobre o Extreme Programming (XP)

- As práticas do XP:
 - Jogo de planejamento
 - As decisões sobre os prazos e escopo são tomadas pelos clientes...
 - Pequenas liberações
 - Devem ser feitas liberações o mais rápido possível para o ambiente de produção...
 - Metáfora
 - É definida uma metáfora para o objetivo do sistema...
 - Projeto simplificado
 - O código deve ser sempre o mais simples possível...
 - Testes
 - Os testes unitários são escritos pelos programadores com bastante frequência. Os clientes escrevem os testes de aceitação. Os resultados dos testes são pu-blicados e ficam visíveis para todos da equipe...

Sobre o Extreme Programming (XP)

- As práticas do XP...
 - Redesenho
 - O código vai sendo melhorado aos poucos...
 - Programação em pares
 - Todo o código é escrito por um par de programadores...
 - Integração contínua
 - Novas classes e métodos são integrados imediatamente...
 - Propriedade coletiva do código
 - Qualquer programador, a qualquer momento, pode alterar qualquer porção do código fonte...
 - Cliente disponível
 - O cliente ou usuário fica integralmente disponível para a equipe...

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

17

Sobre o Extreme Programming (XP)

- As práticas do XP...
 - Semana de 40 horas
 - Se houver necessidade de trabalho extra, é sinal que há problemas...
 - Ambiente aberto
 - O time trabalha em um ambiente bastante espaçoso. O grupo de programação trabalha em estações de trabalho localizadas no centro do ambiente
 - Somente regras
 - As regras podem ser adaptadas e melhoradas, de acordo com a necessidade. Elas são importantes mas são apenas regras...

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

18

Sobre o Extreme Programming (XP)

- Sobre o processo proposto pelo XP...
 - Ele se inicia com o cliente escolhendo as funcionalidades que serão implementadas. Estas funcionalidades são chamadas de *estórias do usuário* (*user stories*). A escolha leva em conta a estimativa de custo/tempo feita pelos programadores
 - o desenvolvimento é fortemente guiado a testes (TDD: *Test-Driven Development*). Os programadores escrevem testes unitários, que são classes que automatizam sequências de testes sobre outras classes. São escritos antes do código estar completo...
 - Normalmente no par de programadores procura-se unir um com muita experiência em TDD e outro com pouca ou nenhuma ex-periência nesta técnica.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

19

Sobre o Extreme Programming (XP)

- Sobre as estórias dos usuários...
 - Uma estória do usuário é uma unidade funcional...
 - Ela deve ser entendida pelos clientes e usuários, deve ser testável, deve ter valor para o cliente e deve ser pequena o bastante para que os programadores possam construir dúzias delas em um ciclo de iteração...
 - Ela é formada por uma ou duas sentenças que descrevem algo com valor para o cliente:
 - *o sistema deve verificar se o CPF do cliente é um número válido...*
 - Os programadores deverão ser capazes de estimar o custo/tempo para implementar a estória. Caso isto não seja possível, a estória deve ser subdividida em estórias menores, para serem estimadas...
 - As estórias do usuários devem ser criadas pelos clientes e usuários. Os desenvolvedores concentram-se nas decisões técnicas...

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

20

Sobre o Extreme Programming (XP)

■ Priorizando as estórias dos usuários...

- O principal critério de ordenamento das estórias é o valor para o negócio...
- Não existe consenso nisto:
 - *“vamos confessar que nós não concordamos neste ponto (priorização das estórias com base no risco técnico). Martin é muito mais inclina-do a trazer as estórias de maior risco técnico para o início do plano do que Kent o é. Kent considera Martin um covarde por esta atitude. Martin concorda.”*

*Kent Beck e Martin Fowler, em
Planning Extreme Programming,
Addison-Wesley, 2000*

- Busca-se consenso negociado quando houver problemas de priorização. Entretanto, a última palavra é dos clientes: eles é que estão pagando o projeto...

Sobre o Extreme Programming (XP)

■ Reportando o progresso de um projeto XP

- O progresso de cada iteração é medido e reportado por uma pessoa chamada *tracker*...
- A cada programador, o *tracker* faz duas perguntas básicas:
 - *Quantos dias ideais você já trabalhou nesta tarefa?*
 - *Mais quantos dias ideais você precisa para completar a tarefa?*
- Se o *tracker* descobre que um programador não vai conseguir terminar sua tarefa, ele tenta redistribuir o encargo para outro programador que esteja com alguma folga. Caso isto não seja possível, o cliente deve ser informado...
- As iterações no XP terminam na data estimada. As estórias implementadas, são apresentadas aos clientes, que decidirá se está adequada, a qual, neste caso, será considerada completa. As estórias incompletas serão consideradas para a próxima iteração...

Premissa do desenvolvimento ágil

- APRENDIZADO
 - O cliente aprende durante o desenvolvimento
 - Descobre novas possibilidades
 - Muda as prioridades
 - Sistema deve acompanhar o aprendizado
- FEEDBACK
- DESENVOLVIMENTO ITERATIVO

Aprendizado

- O cliente aprende durante o desenvolvimento
- Descobre novas possibilidades
- Muda as prioridades
- Sistema deve acompanhar o aprendizado



MA-XP

*Valores e
Princípios XP*



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

25

Valores do XP

- Feedback
- Comunicação
- Simplicidade
- Coragem

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

26

Vencendo os Medos

- Escrever código.
- Mudar de idéia.
- Ir em frente sem saber tudo sobre o futuro.
- Confiar em outras pessoas.
- Mudar a arquitetura de um sistema em funcionamento.
- Escrever testes.

Princípios Básicos do XP

- Feedback rápido
- Simplicidade é o melhor negócio
- Mudanças incrementais
- Abraçar mudanças - Carregue a bandeira das mudanças / não valorize o medo (*Embrace change*)
- Alta qualidade.

MA-XP

Práticas
do XP



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

29

As 12 Práticas de XP (versão 2000)

1. Planejamento
2. Fases Pequenas
3. Metáfora
4. Design Simples
5. Testes
6. Refatoramento
7. Programação Pareada
8. Propriedade Coletiva
9. Integração Contínua
10. Semana de 40 horas
11. Cliente junto aos desenvolvedores
12. Padronização do código

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

30

User Stories

- Funcionalidades são informadas através de *user stories*
- Estórias devem ser simples
- Desenvolvedores estimam tempo



Cliente

- Responsável por escrever “histórias”.
- Muitas vezes é um programador ou é representado por um programador do grupo.
- Trabalha no mesmo espaço físico do grupo.
- Novas versões são enviadas para produção todo mês (ou toda semana).
- *Feedback* do cliente é essencial.
- Requisitos mudam (e isso não é mau).

Coach (treinador)

- Em geral, o mais experiente do grupo.
- Identifica quem é bom no que.
- Lembra a todos as regras do jogo (XP).
- Eventualmente faz programação pareada.
- Não desenha arquitetura, apenas chama a atenção para oportunidades de melhorias.
- Seu papel diminui à medida em que o time fica mais maduro.

April 05

Prof. Ismael H. F. Santos - ismael@tegraf.puc-rio.br

33

Tracker (Acompanhador)

- A “consciência” do time.
- Coleta estatísticas sobre o andamento do projeto.
Alguns exemplos:
 - Número de histórias definidas e implementadas.
 - Número de *unit tests*.
 - Número de testes funcionais definidos e funcionando.
 - Número de classes, métodos, linhas de código
- Mantém histórico do progresso.
- Faz estimativas para o futuro.

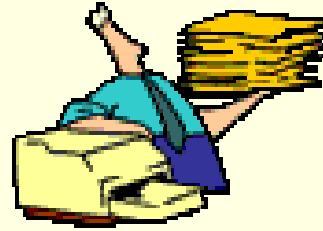
April 05

Prof. Ismael H. F. Santos - ismael@tegraf.puc-rio.br

34

Jogo do Planejamento

- De posse das estimativas, é possível priorizar
- Cliente entrega as estórias priorizadas para serem implementadas
- Desenvolvedores respeitam as prioridades dos clientes
- Projeto é dividido em partes: releases e iterações



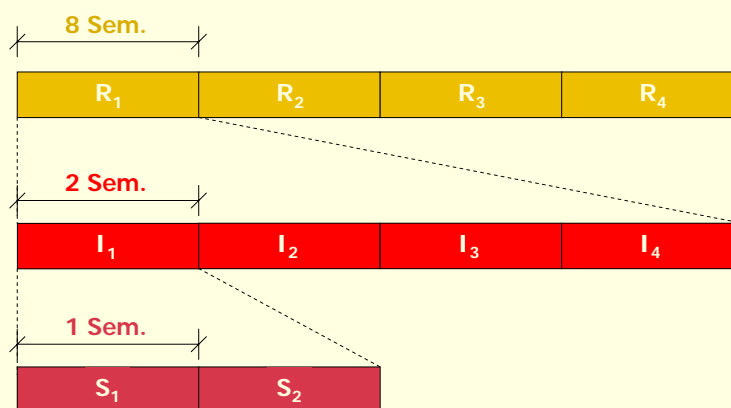
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

35

Release e Iteração

Projeto: 8 meses = 32 semanas



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

36

Release Planning

8 Sem.



- Cliente recebe um “orçamento” dos desenvolvedores
- Seleciona as histórias prioritárias que possam ser implementadas dentro do orçamento
- Pode trocar histórias durante o release

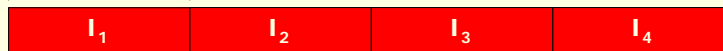
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

37

Iteration Planning

2 Sem.



- Cliente recebe um “orçamento”
- Não pode haver troca de funcionalidades durante uma iteração
- Velocidade: quantidade de histórias que a equipe consegue implementar em uma iteração

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

38

Stand-up meeting

- Reunião rápida
- Diária
- Usada para atualização da equipe



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

39

Desenvolvimento: orientação a objetos

- Sistema é composto por pequenas peças
- Peças têm objetivos específicos
- Facilita construção
- Facilita evolução



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

40

Pair Programming

- Todo código é escrito em par
- Duas pessoas trabalham em um único computador
- Uma digita, enquanto a outra revisa, corrige e sugere



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

41

Programação Pareada

- Erro de um detectado imediatamente pelo outro (grande economia de tempo).
- Maior diversidade de idéias, técnicas, algoritmos.
- Enquanto um escreve, o outro pensa em contra-exemplos, problemas de eficiência, etc.
- Vergonha de escrever código feio (*gambiarra*) na frente do seu par.
- Pareamento de acordo com especialidades.
 - Ex.: Sistema Multimídia Orientado a Objetos

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

42

Collective code ownership

- Todos são responsáveis por todas as partes do sistema
- Código tem que estar sempre “limpo”
- Todos são capazes de modificá-lo



Propriedade Coletiva do Código

- Modelo tradicional: só autor de uma função pode modificá-la.
- XP: o código pertence a todos.
- Se alguém identifica uma oportunidade para simplificar, consertar ou melhorar código escrito por outra pessoa, que o faça.
- Mas rode os testes!

O Código

- Padrões de estilo adotados pelo grupo inteiro.
- O mais claro possível.
 - XP não se baseia em documentações detalhadas e extensas (perde-se sincronia).
- Comentários sempre que necessários.
- Comentários padronizados.
- Programação Pareada ajuda muito!

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

45

Refactoring

- Melhorar o código permanentemente
- Alterações na implementação sem afetar a interface



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

46

Refatoramento (*Refactoring*)

- Uma [pequena] modificação no sistema que não altera o seu comportamento funcional
- mas que melhora alguma qualidade não-funcional:
 - simplicidade
 - flexibilidade
 - clareza
 - desempenho

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

47

Exemplos de Refatoramento

- Mudança do nome de variáveis
- Mudanças nas interfaces dos objetos
- Pequenas mudanças arquiteturais
- Encapsular código repetido em um novo método
- Generalização de métodos
 - `raizQuadrada(float x) ⇒ raiz(float x, int n)`

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

48

MA-XP

Bugs e Tests



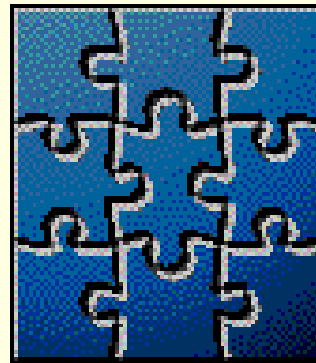
April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

49

Buscando bugs

- Sistemas podem ter defeitos
- É difícil descobrir qual é a peça defeituosa
- É mais fácil se o problema for detectado logo após a concepção da peça



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

50

Testes

- Fundamento mais importante de XP.
- É o que dá segurança e coragem ao grupo.
- Testes de unidades (*Unit tests*)
 - escritos pelos programadores para testar cada elemento do sistema (e.g., cada método em cada caso).
- Testes de funcionalidades (*Functional tests*)
 - escritos pelos clientes para testar a funcionalidade do sistema.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

51

Testes

- Testes das unidades do sistema
 - Tem que estar sempre funcionando a 100%.
 - São executados várias vezes por dia.
 - Executados à noite automaticamente.
- Testes das funcionalidades
 - Vão crescendo de 0% a 100%.
 - Quando chegam a 100% acabou o projeto.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

52

Unit Test

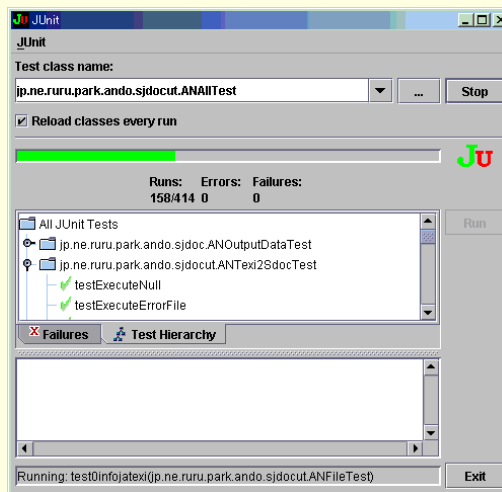
- Teste feito sobre cada classe
- Cada classe possui um unit test associado a ela
- Testes são automatizados
- Quando uma nova classe entra no sistema, todos os testes são executados



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

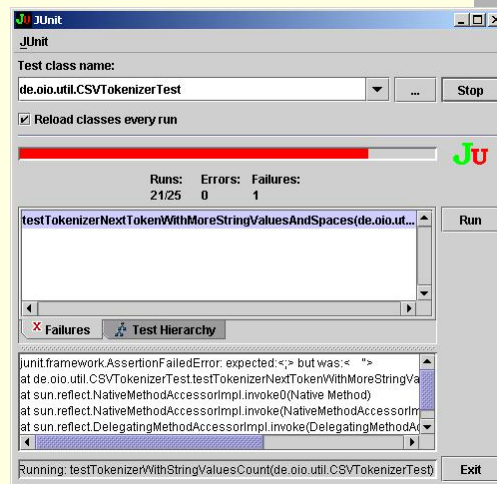
53



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

54



Um Projeto XP

- Fase de Exploração
 - duração: 2 a 6 meses.
 - termina quando a primeira versão do software é enviada ao cliente.
 - clientes escrevem “historias” (*story cards*).
 - programadores interagem com clientes e discutem tecnologias.
 - Não só discutem, **experimentam** diferentes tecnologias e arquiteturas para o sistema.
 - Planejamento: 1 a 2 dias.

Um Dia na Vida de um Programador XP

- Escolhe uma história do cliente.
- Procura um par livre.
- Escolhe um computador para programação pareada (*pair programming*).
- Seleciona uma tarefa claramente relacionada a uma característica (*feature*) desejada pelo cliente.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

57

Um Dia na Vida de um Programador XP

- Discute modificações recentes no sistema
- Discute história do cliente
- Testes
- Implementação
- Desenho
- Integração

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

58

Um Dia na Vida de um Programador XP

- Atos constantes no desenvolvimento:
 - Executa testes antigos.
 - Busca oportunidades para simplificação.
 - Modifica desenho e implementação incrementalmente baseado na funcionalidade exigida no momento.
 - Escreve novos testes.
 - Enquanto todos os testes não rodam a 100%, o trabalho não está terminado.
 - Integra novo código ao repositório.

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

59

Desenvolvimento em equipe

- Como conciliar diversos desenvolvedores?
- Estrutura de diretórios
- Atualização do código por várias pessoas diferentes

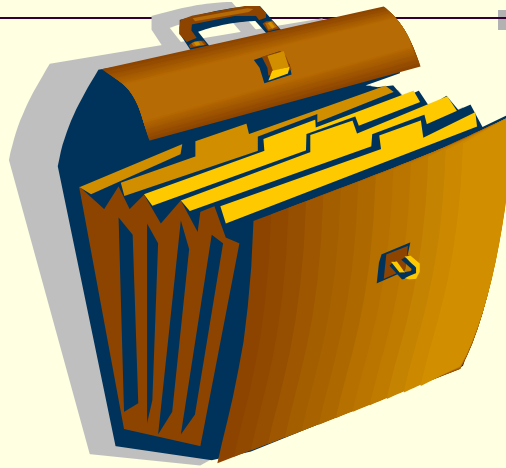


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

60

Repositório



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

61

CVS

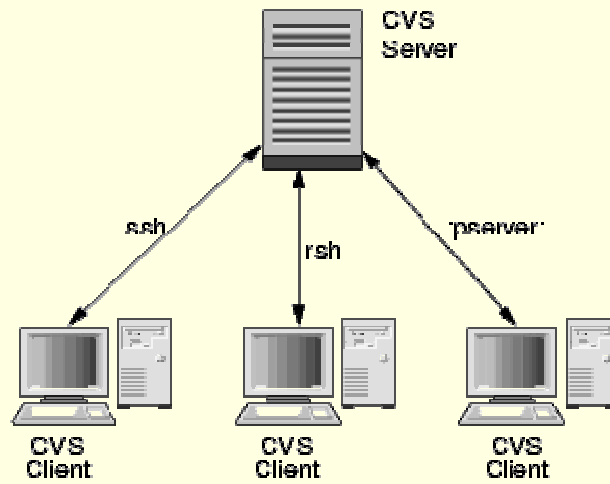


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

62

CVS

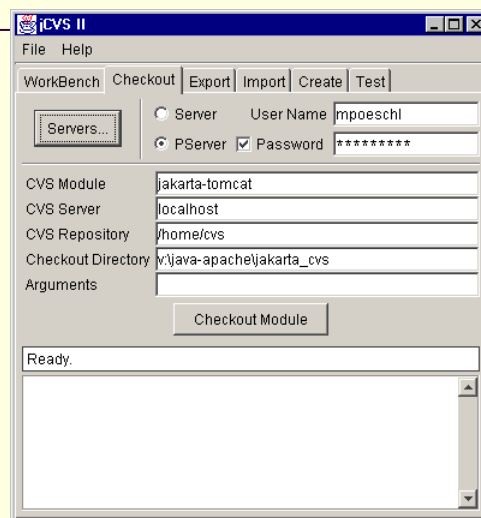


April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

63

jCVS



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

64

CVS – Principais conceitos

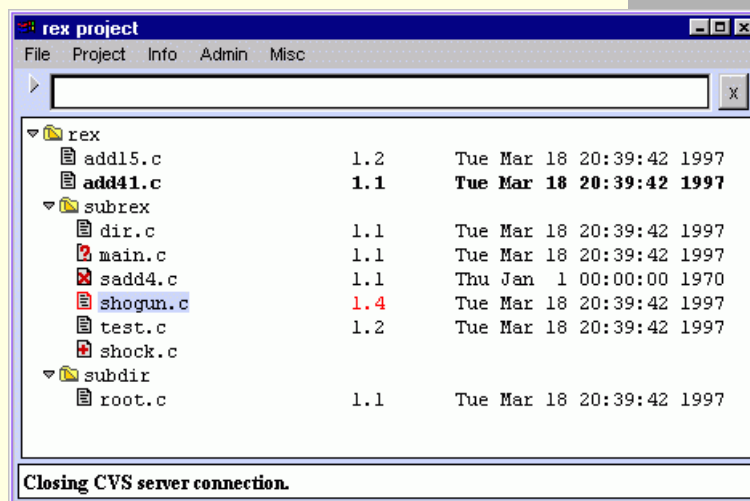
- Check out
- Add
- Commit
- Update
- Conflict
- Remove

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

65

jCVS



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

66

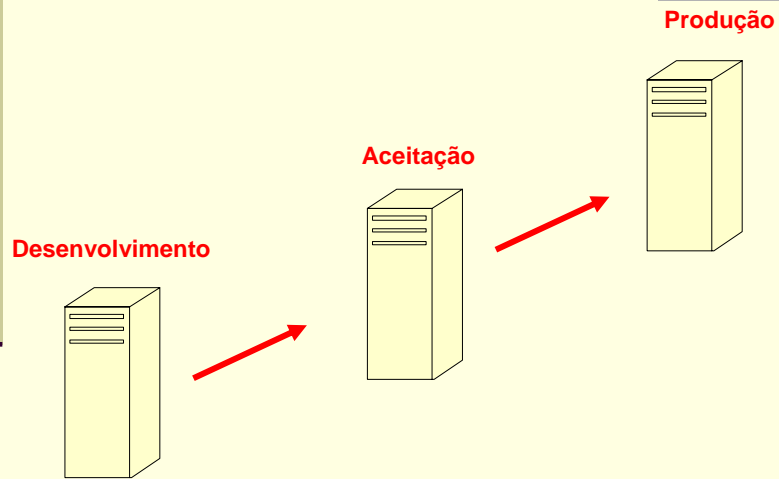
Integração contínua

- Máquina destacada para a integração
- Integração ocorre diversas vezes ao dia
- Os testes são executados em cada integração
- Correções são feitas imediatamente

Ambientes para deployment

- Como lidar com diferentes ambientes?
- Exemplo:
 - Desenvolvimento
 - Aceitação
 - Produção

Ambientes para deployment



April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

69

Problemas de ter vários ambientes

- Possíveis erros devido a tarefas manuais repetitivas
- Gasto de tempo

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

70

Automatização do deployment

- Agiliza as integrações
- Evita erros

Ant



Exemplo simples

```
<project name="projeto" default="desenvolvimento">  
  
  <target name="desenvolvimento">  
    <mkdir dir="build"/> ← tasks  
    <javac srcdir="src" destdir="build"/> ← tasks  
    <junit>  
      <test name="test.SystemTester"/> ← tasks  
    </junit>  
  </target>  
  
</project>
```

target

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

73

Ambiente de Desenvolvimento

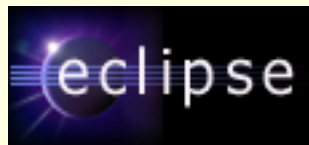
- Desejável que tenha:
 - Code completion
 - Validação on-line
 - Suporte à depuração
 - Suporte a refactoring
 - Integração com junit
 - Integração com CVS
 - Integração com Ant

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

74

Exemplos



Importância da IDE para o XP

- Ganho de tempo
- Automação
- Facilidade de uso
- Verificação permanente do código

Importância da IDE para o XP

```
1  
2  
3 import java.io.BufferedReader;  
4 import java.io.InputStreamReader;  
5 import java.io.FileInputStream;  
6 import java.io.FileNotFoundException;  
7  
8  
9 /**  
10  * Created by IntelliJ IDEA.  
11  * User: Pti  
12  * Date: Jun 25, 2003  
13  * Time: 12:30:13 PM  
14  * To change this template use Options | File Templates.  
15  */  
16 public class ReflectionSample {  
17     public static void main(String[] args) throws FileNotFoundException {  
18         BufferedReader reader = new BufferedReader(new InputStreamReader(new FileInputStream(  
19             String line = reader.readLine();  
20             name (line);  
21             " + forName.getName() + " extends " + forName.getS  
22         }  
23     }  
24 }
```

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

77

XP no Brasil

- 2000: algumas práticas começam a ser usadas em projetos no Sul
- Dez/2002: primeiro XP Brasil
- Jan/2003: fundação do XP Rio
- Fev/2003: projeto *full-XP* em uma grande empresa brasileira (em andamento)
- Set/2003: lançamento previsto do primeiro livro brasileiro sobre XP

April 05

Prof. Ismael H. F. Santos - ismael@tecgraf.puc-rio.br

78

Referências sobre XP

- Site Xispê
<http://www.xispe.com.br>
- XP Rio
<http://www.yahoogroups.com/groups/xprio>
- Lista Xpers
<http://www.yahoogroups.com/groups/xpers>

Referências sobre as ferramentas

- jUnit
<http://www.junit.org>
- CVS
<http://www.cvshome.org>
- Ant
<http://ant.apache.org>
- IntelliJ
<http://www.intellij.com>