# GERENCIAMENTO DE CONFIGURAÇÕES DE SOFTWARE USANDO A FERRAMENTA "GFORGE"

Artur Sampaio - ICMC/USP

Agosto de 2010

## SCM - Definição

- Conjunto de práticas adotadas para administração de códigos fonte, controle de alterações e distribuição de pacotes de software.
  - Gerenciar diferentes versões dos produtos de *software*, controlando, auditando e reportando as alterações realizadas
  - Garantir a integridade, confiabilidade e reprodutibilidade do desenvolvimento de produtos de *software*, desde o projeto até a distribuição



## SCM – Visão geral

- Rastrear e controlar alterações realizadas no *software*
  - Controlar revisões (VCS)
  - Rastrear defeitos (*Issue/Ticket Tracking*)
  - Manter histórico da evolução do produto
- Estabelecer *baselines* (distribuições estáveis)
- Garantir aderência ao processo de desenvolvimento



## A edição concorrente - 1




## A edição concorrente - 2




## A edição concorrente - 3

# VCS (Version Control Systems) - Controle de revisões

- Gerenciamento de alterações em arquivos
  - Alterações recebem um identificador único, para posterior referência
  - Permite resgatar versões anteriores dos arquivos
  - Permite a criação de linhas de desenvolvimento paralelas (*branches*)
  - Auxilia o usuário no processo de *merge* (mesclagem) de diferentes versões do arquivo
  - Utilização eficiente de espaço, armazenando apenas as alterações entre versões

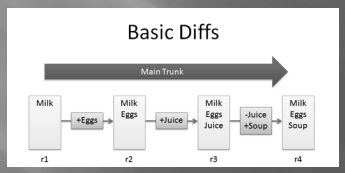


# VCS - Ferramentas

- Gratuitos / comunitários
  - CVS (Concurrent Version System)
  - SVN (Subversion)
  - Git (Linus Torvalds)
- Comerciais
  - IBM Rational ClearCase
  - Borland Starteam
  - Microsoft Visual SourceSafe

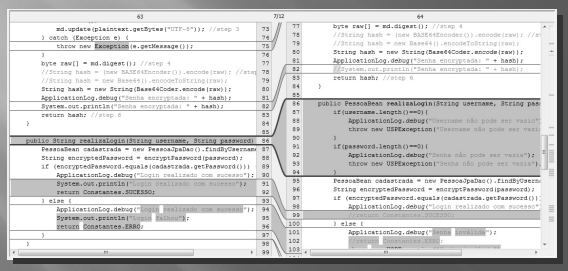


# VCS Diff



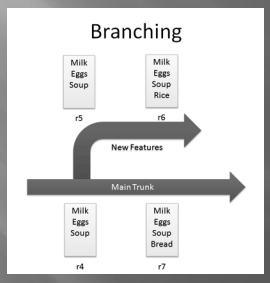



# VCS Compare



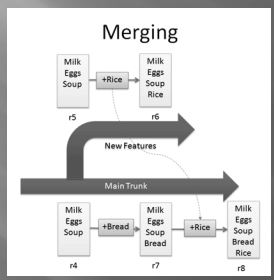


# VCS Branch

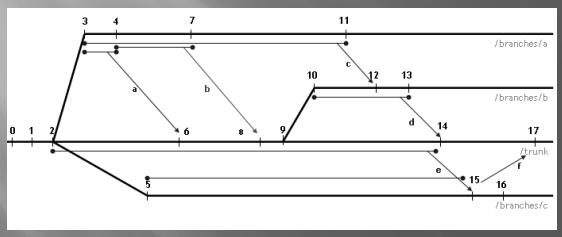





# VCS Merge



## Histórico de mudanças






## Bug Tracking

- Registro e rastreamento de requisitos (defeitos existentes ou melhorias propostas)
  - Auxilia na distribuição de atividades entre membros da equipe
  - Atribui graus de prioridade a cada requisito
  - Acompanha o ciclo de vida de cada requisito, registrando seu histórico e estado
  - Suporta o processo de desenvolvimento adotado pela equipe

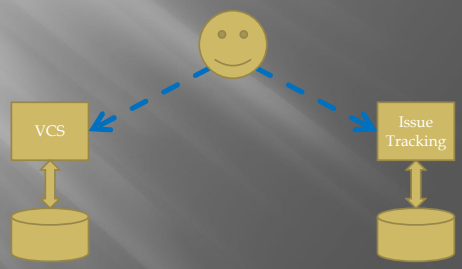


## Bug Tracking - Ferramentas

- Gratuitos / comunitários
  - Bugzilla
  - MantisBT
  - Trac
- Comerciais
  - IBM Rational Clear Quest



## Ferramentas não integradas



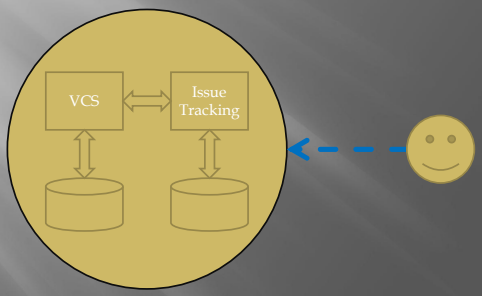



## Ferramentas não integradas

- Processo centrado no usuário
  - Usuário é responsável por manter consistência entre VCS e *tracker*
  - Não há ligação entre os artefatos *commitados* e a atividade do *tracker*
  - Boa prática: cada *commit* recebe como comentário o número do *tracker* associado, tentando garantir rastreabilidade



## Ferramentas integradas



# Ferramentas integradas

- Processo centrado em ferramentas
  - Consistência automática entre VCS e *tracker*
  - Possibilidade de só permitir *commits* com *tracker* associado
  - Criação automática vínculo entre artefatos *commitados* e atividades de *tracker*
- Sourceforge
  - Só hospeda aplicações *open source*
- JIRA
  - Produto comercial

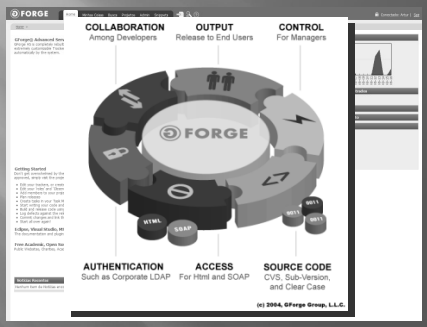


# GForge

- Ambiente colaborativo para desenvolvimento de software
  - Derivado do SourceForge; mesma equipe de desenvolvimento
  - Suporta múltiplos projetos e múltiplos usuários
  - Integra módulos de VCS, tracking, comunicação, fórum e baselines
  - Realiza levantamento de métricas de produtividade por projeto e por membro

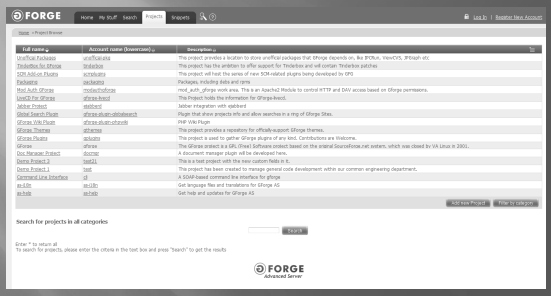


# Tela inicial do GForge



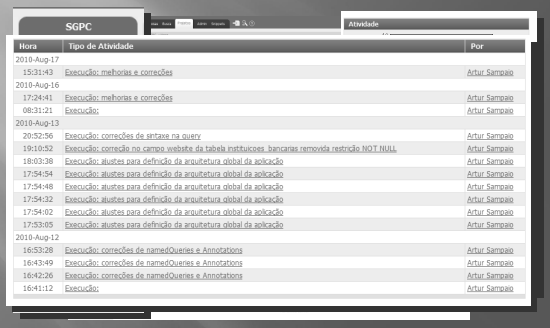



# Listagem de projetos



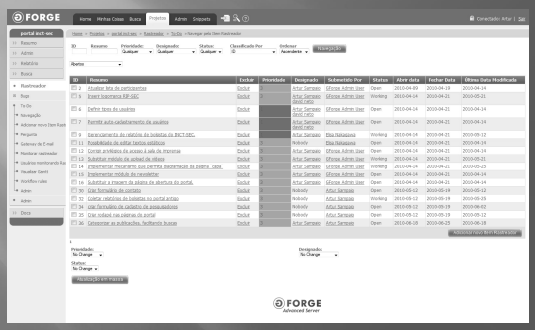


# Resumo de um projeto

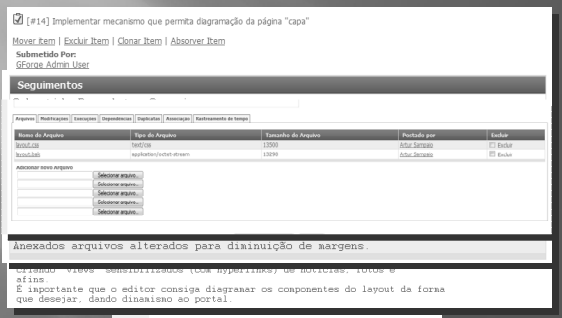





# Listagem de BUG´s



# Detalhamento de BUG

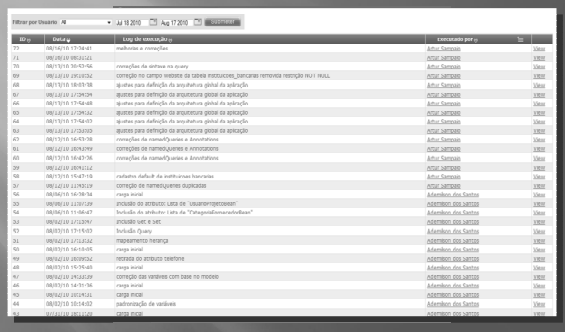



Anexados arquivos alterados para diminuição de margens.

Criando views sensibilizadas (com hyperlinks) de matrizes, totais e afins.
É importante que o editor consiga diagramar os componentes do layout da forma que desejar, dando dinamismo ao portal

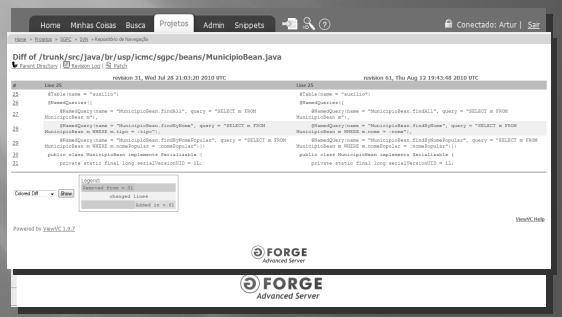


# Listagem de commits SVN



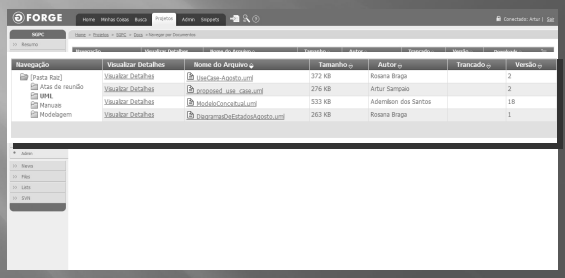


# Detalhes de commit SVN



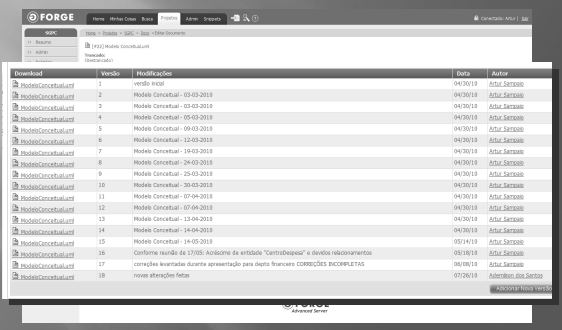



# Listagem de documentos



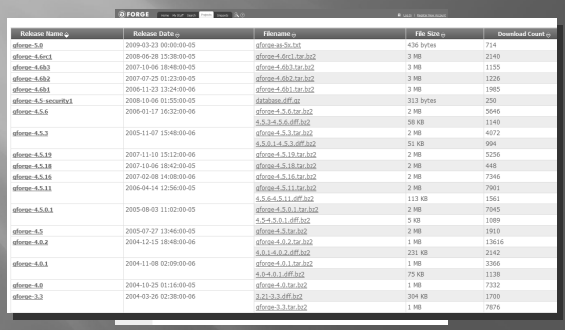



# Detalhes do documento

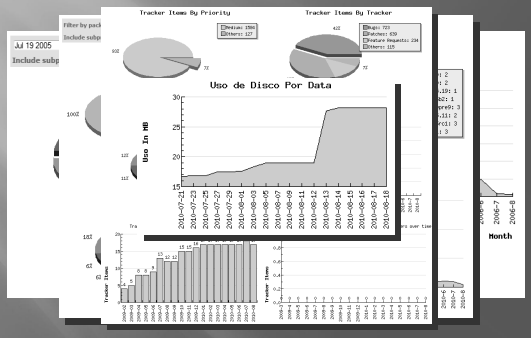





# Releases



# Relatórios






# Conclusões

- Práticas de SCM são fundamentais para a maturidade de todo e qualquer produto de software
- É necessário o uso de ferramentas adequadas
- Ferramentas isoladas dependem muito da disciplina do desenvolvedor
- Ferramentas integradas garantem consistência no processo de desenvolvimento e simplificam a gestão do projeto



# Perguntas



# Quero saber mais

- http://gforge.org/
- http://subversion.tigris.org/

- artur@icmc.usp.br

*"The only constant in software development is change."*