



Repositórios de Componentes nas perspectivas de Gerência de Configuração de Software e Reutilização de Software

Leonardo Gresta Paulino Murta

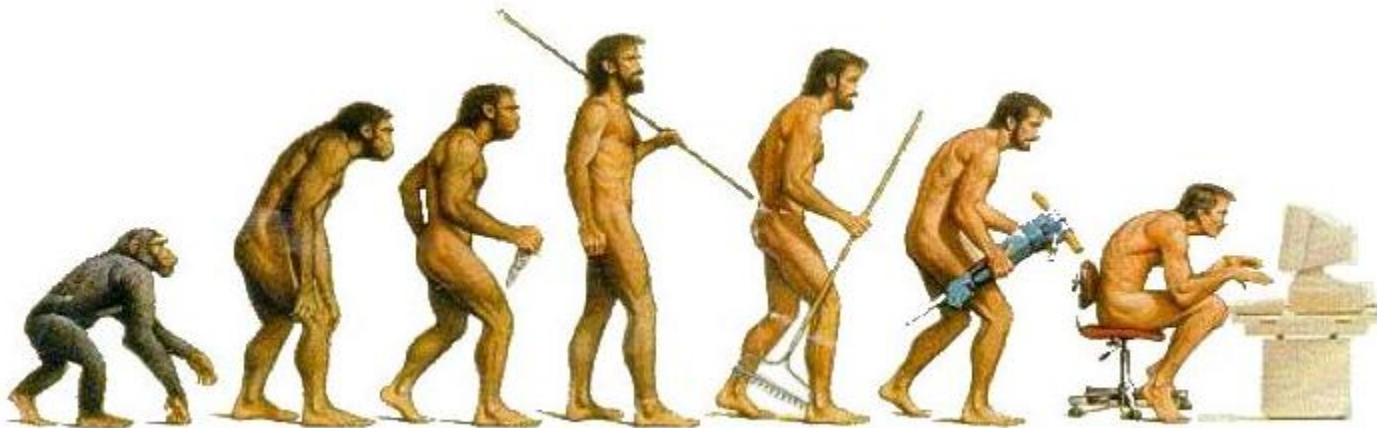
leomurta@ic.uff.br

Agenda

- **Introdução**
- Repositório na perspectiva de Gerência de Configuração
- Repositório na perspectiva de Reutilização
- Integração dos repositórios
- Conclusão

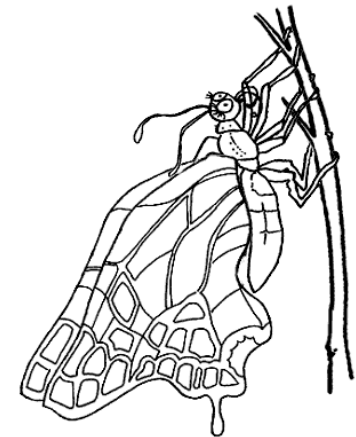
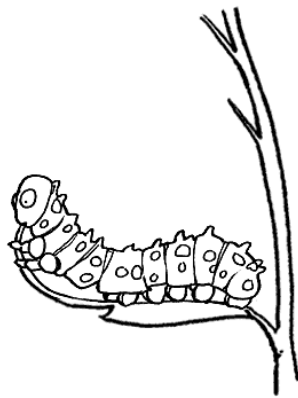
Introdução

- A Engenharia de Software...
 - Abordagem disciplinada para o desenvolvimento de software
 - Grande diversidade de metodologias



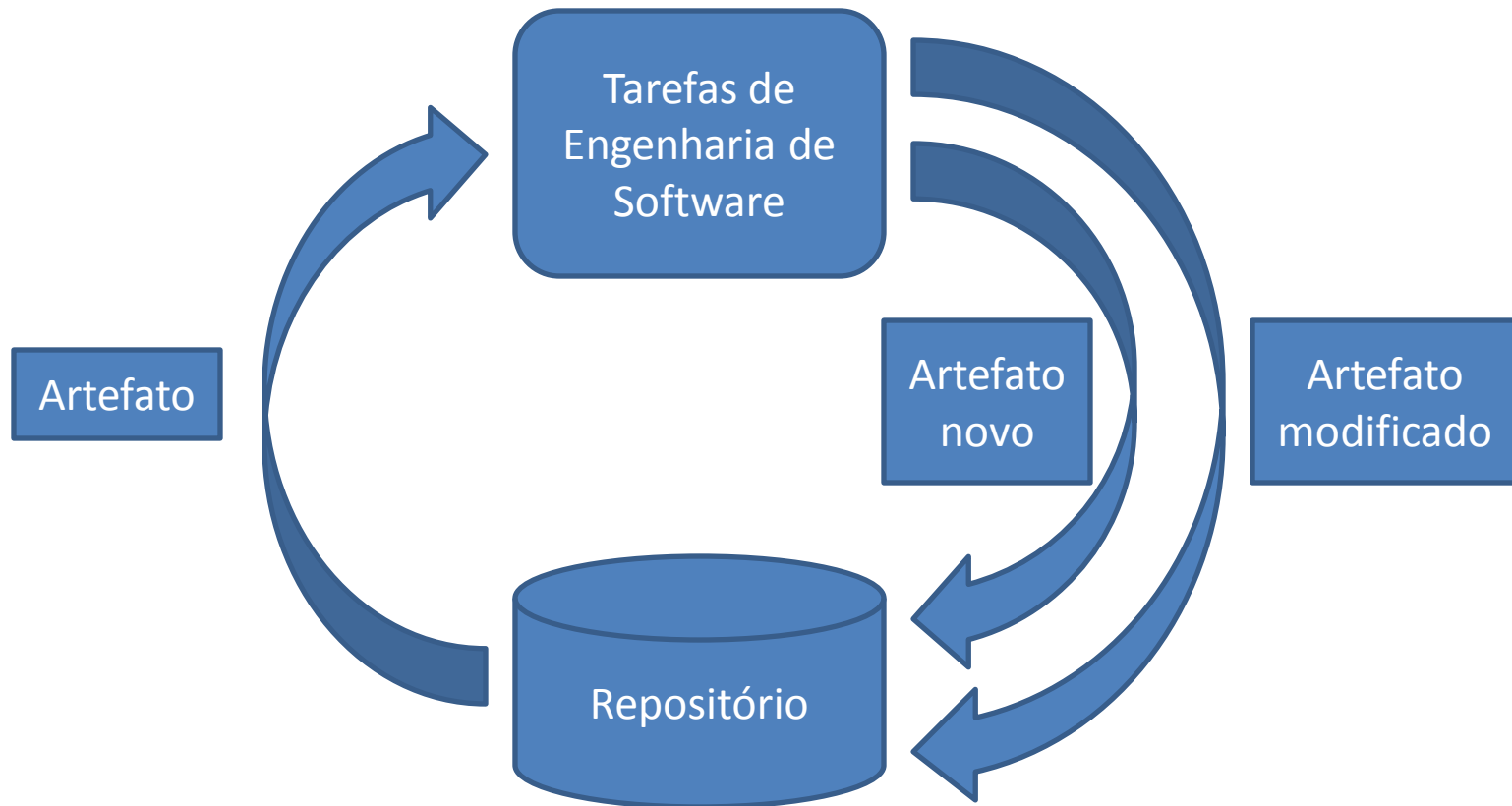
Introdução

- Ponto em comum nas metodologias:
 - refinamentos sucessivos de artefatos



<http://www.colegiosaofrancisco.com.br>

Mas aonde ficam esses artefatos?



O que são repositórios?

- Repositórios
 - Lugar seguro onde artefatos são depositados
 - Permitem armazenamento, busca, e recuperação de artefatos
 - Servem como um ponto de referência
 - Apóiam no aumento da memória organizacional

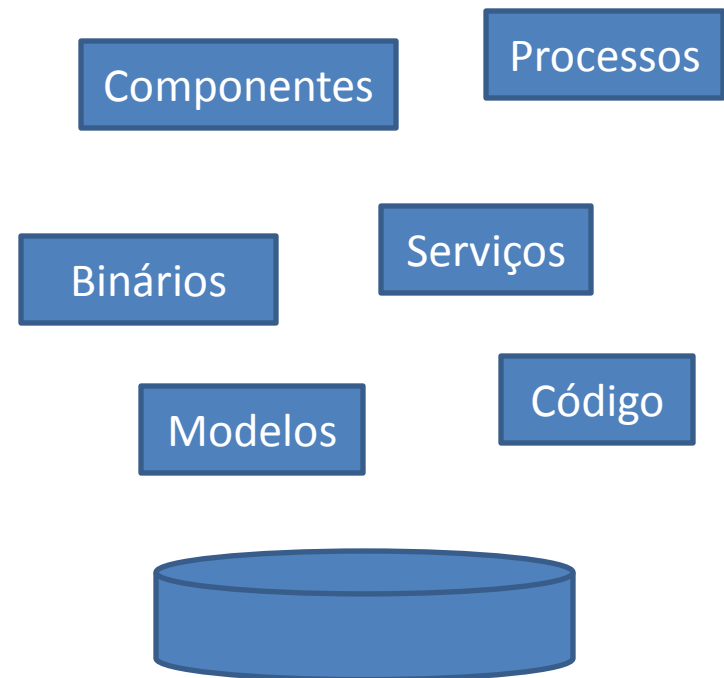


Características de repositórios

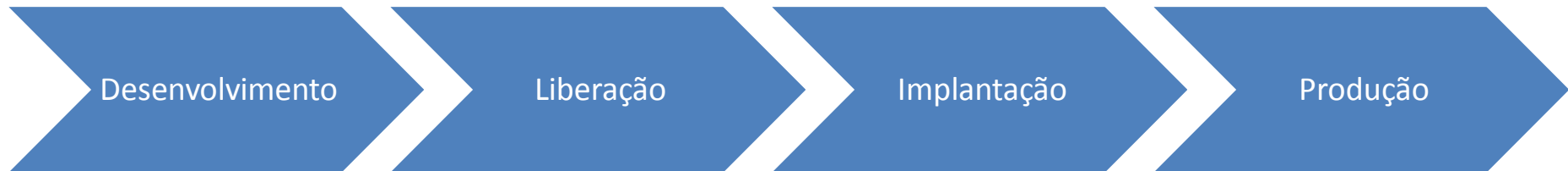
- Controle de acesso
- Auditoria
- Colaboração
- Catalogação
- Controle de solicitações
- Versionamento
- Busca
- Recuperação
- Controle de dependências
- Tarifação
- Medição
- Monitoramento
- Integridade
- Escalabilidade
- Interoperabilidade

Exemplos

- Component Source (<http://www.componentsource.com>)
- AquaLogic (<http://www.bea.com>)
- WebSphere (<http://www.ibm.com>)
- Rational Asset Manager (<http://www.ibm.com>)
- Repository Manager (<http://www.soa.com>)
- DA Manager (<http://www.digitalassets.com.br>)
- Brechó (<http://reuse.cos.ufrj.br>)
- CORE (<http://www.rise.com.br>)
- X-CORE (<http://www.compose.ufpb.br>)
- Debian (<http://packages.debian.org>)
- Gentoo (<http://packages.gentoo.org>)
- Ubuntu (<http://packages.ubuntu.com>)
- Code Beach (<http://www.codebeach.com>)
- Savannah (<http://savannah.gnu.org>)
- SourceForge (<http://sourceforge.net>)
- CVS (<http://www.cvshome.org>)
- Subversion (<http://subversion.tigris.org>)



Tipos de repositórios



Gerência de Configuração

Reutilização

Sob o ponto de vista do MPS.BR...

Repositório de Gerência de Configuração

- Um **Sistema de Gerência de Configuração** é estabelecido e mantido
- A **situação** dos **itens de configuração** e das *baselines* é **registrada** ao longo do tempo e **disponibilizada**
- Modificações em **itens de configuração** são **controladas e disponibilizadas**
- O armazenamento, o manuseio e a liberação de **itens de configuração** e *baselines* são **controlados**.

Repositório de Reutilização

- Um **mecanismo de armazenamento e recuperação** de **ativos reutilizáveis** é implantado
- Os **dados de utilização** dos **ativos reutilizáveis** são **registrados**
- Os usuários de **ativos reutilizáveis** são **notificados** sobre problemas detectados, modificações realizadas, novas versões disponibilizadas e descontinuidade de ativos.

Potenciais semelhanças

- Controle de acesso
- Auditoria
- Integridade
- Interoperabilidade
- Escalabilidade
- Versionamento



Potenciais diferenças

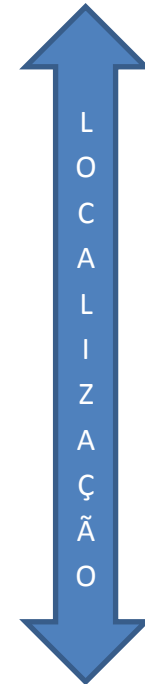
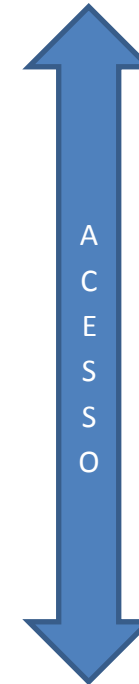
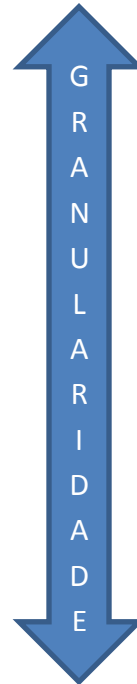
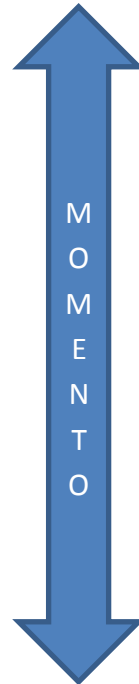


Desenvolvimento

Fina

Edição

Contexto



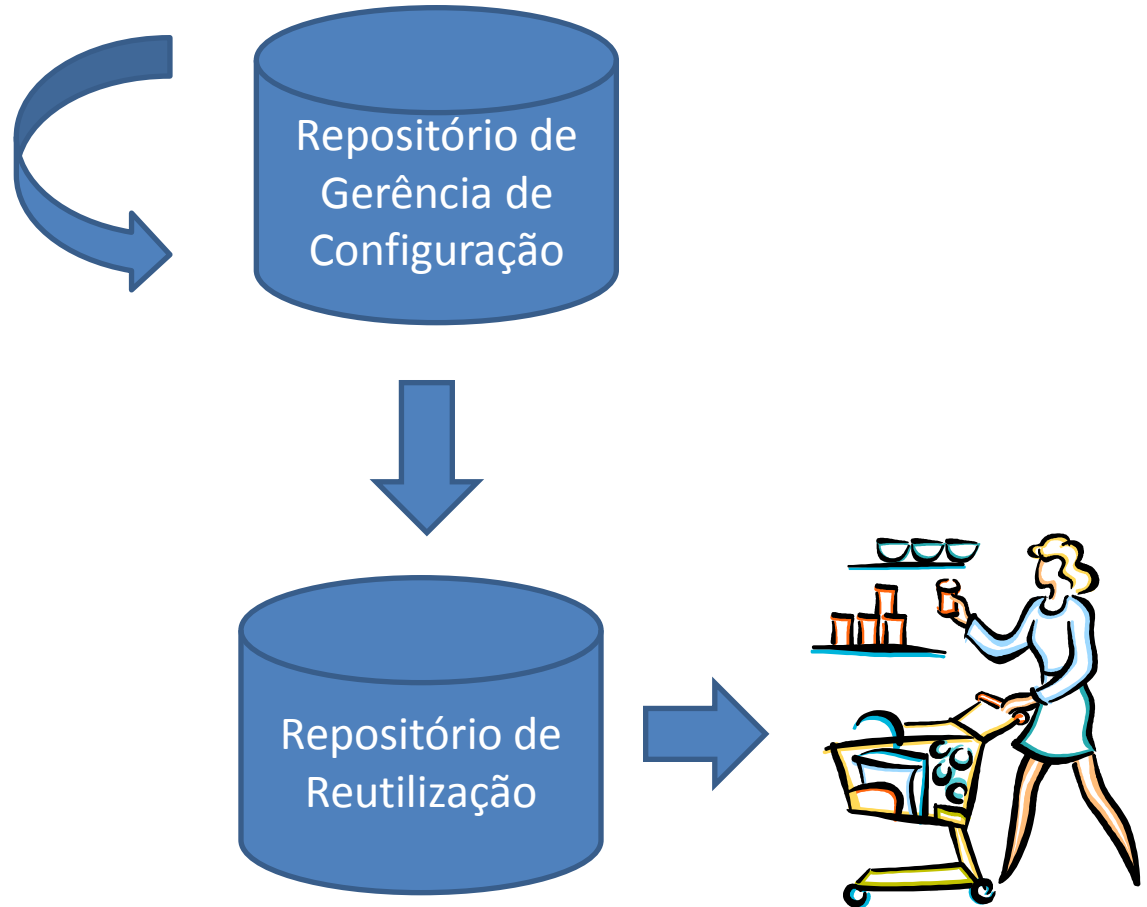
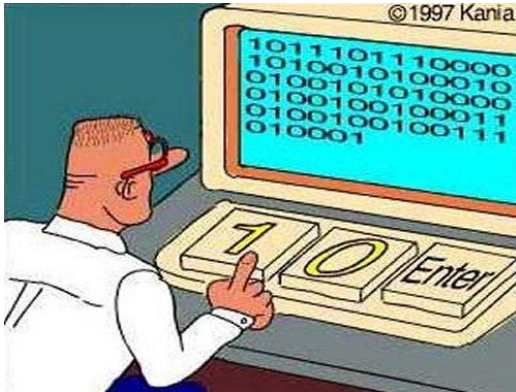
Produção

Grossa

Leitura

Busca

Cenário de utilização



Agenda

- Introdução
- **Repositório na perspectiva de Gerência de Configuração**
- Repositório na perspectiva de Reutilização
- Integração dos repositórios
- Conclusão

Exemplos utilizados



Gerência de Configuração

Gerência de configuração de software é uma **disciplina** para o **controle da evolução de sistemas de software**

Susan Dart

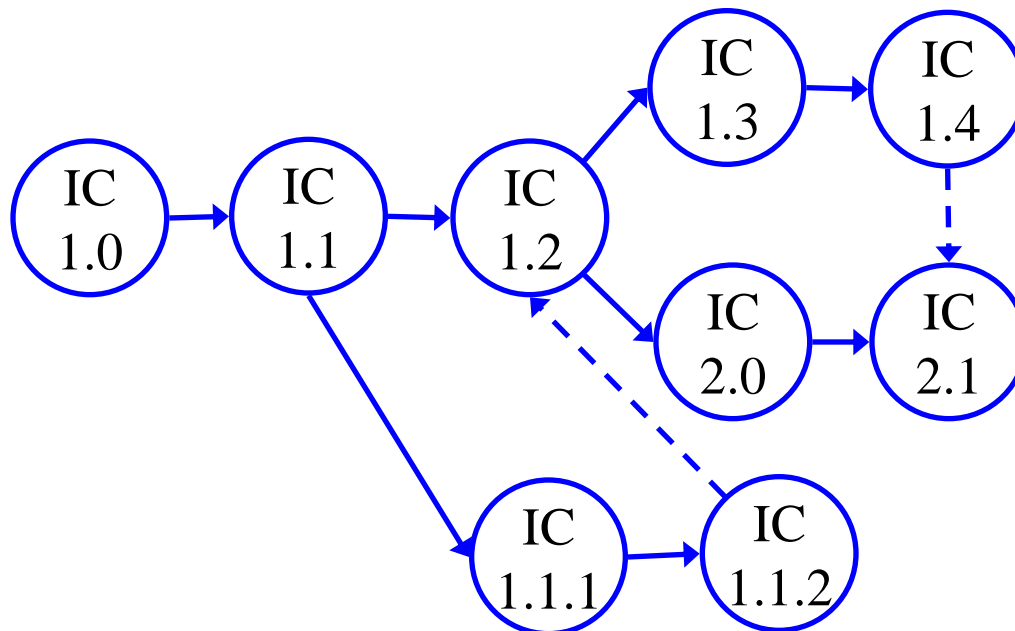
- Principais sistemas
 - **Controle de versões**
 - **Controle de modificações**
 - Gerenciamento de construção e release

Item de configuração

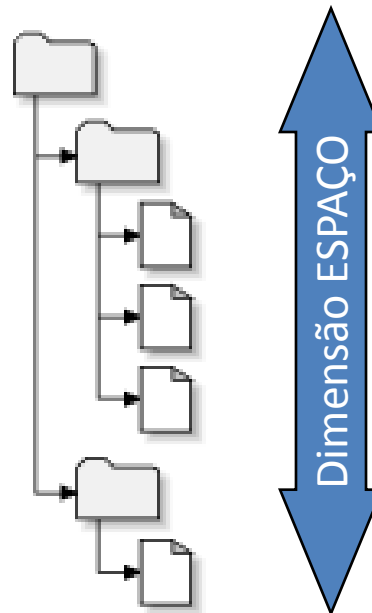
- Agregação de hardware e/ou software que será passível de gerência de configuração e tratado como um elemento único
- Tipos de ICs
 - Produtos de trabalho do projeto
 - Produtos de trabalho de processos
- Exemplos: plano de GC, requisitos, modelos, código-fonte, etc.

Versão

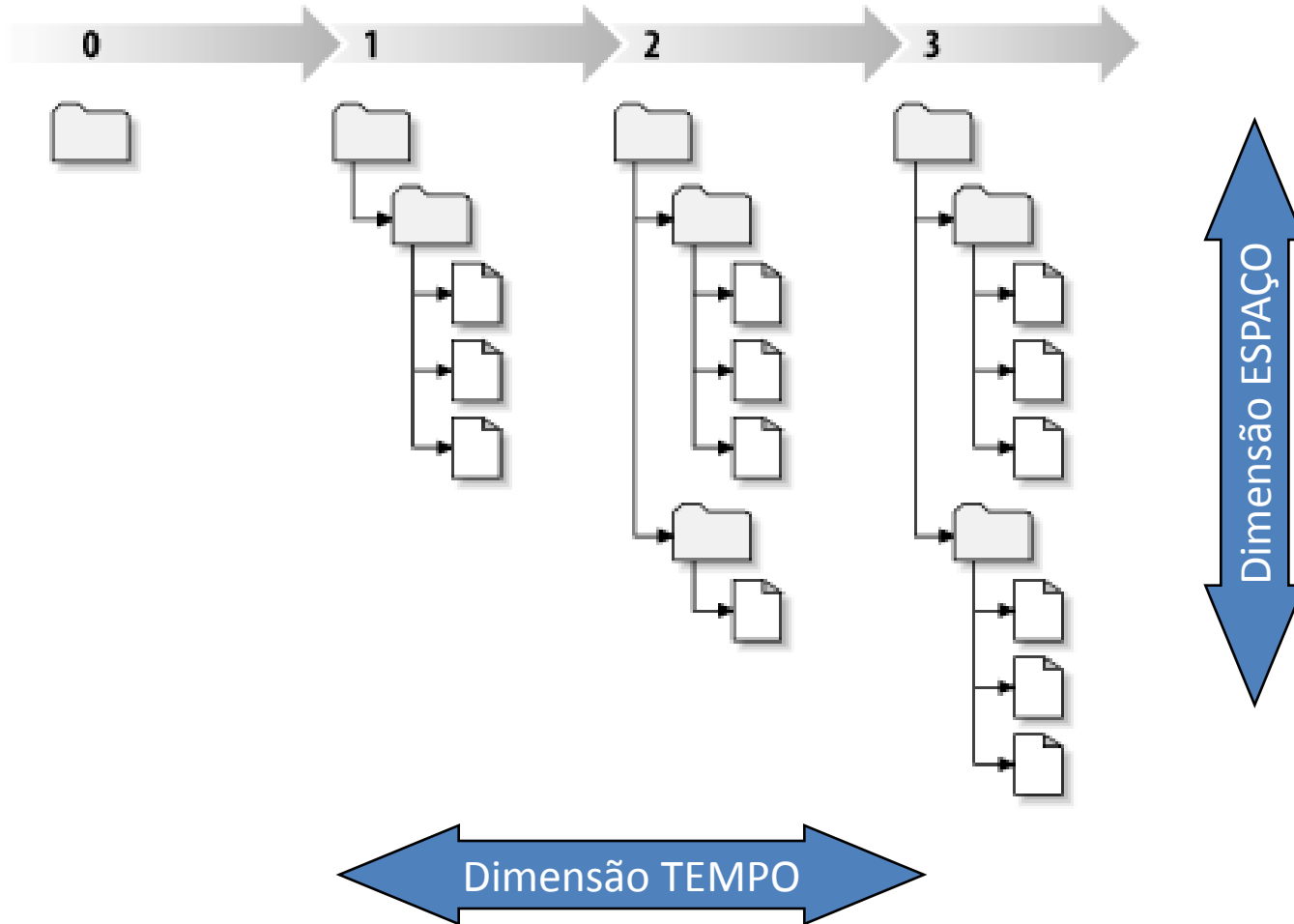
- Instâncias de um mesmo item de configuração que diferem entre si em algo (sinônimo: revisões)



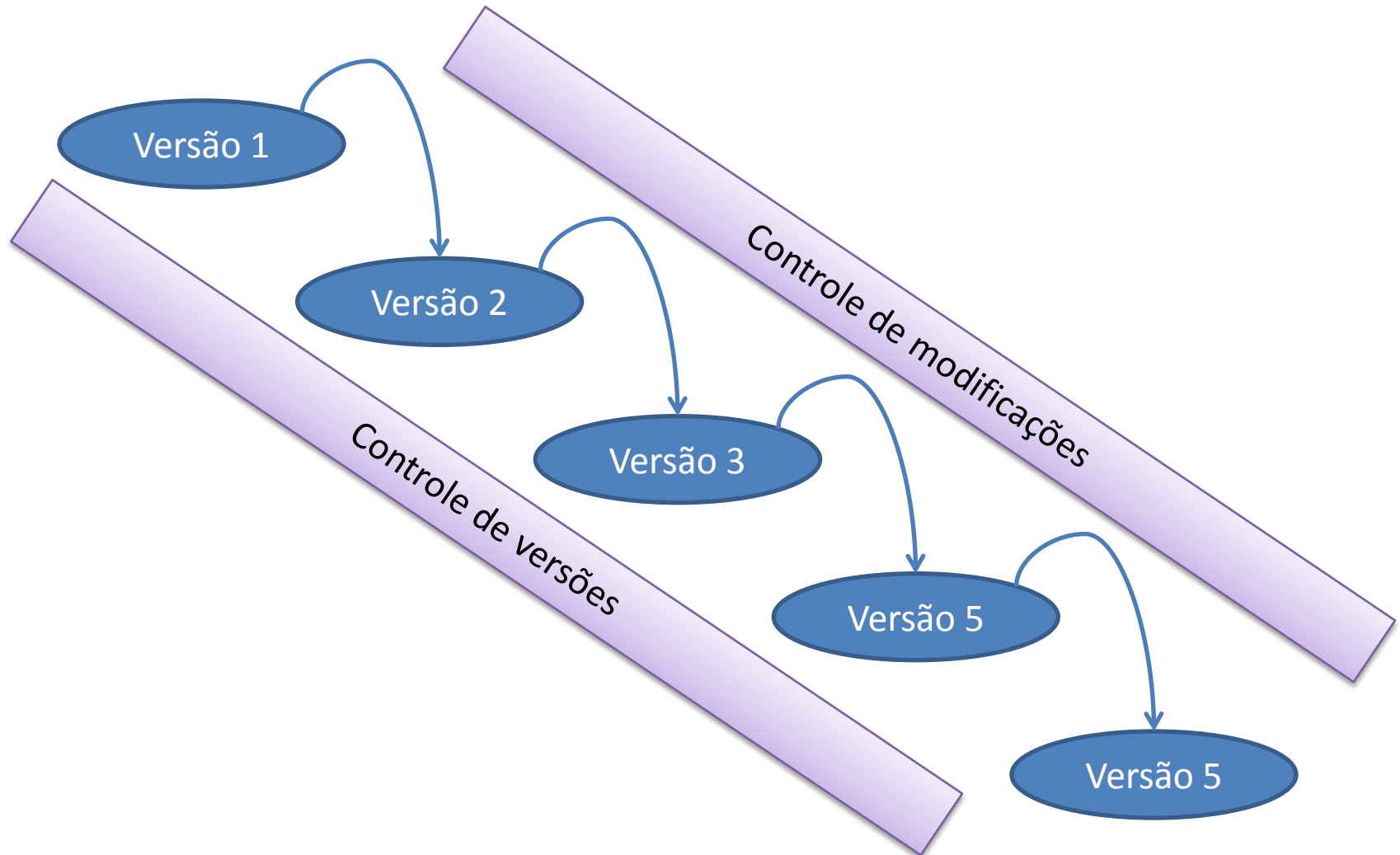
Sistema de arquivo tradicional



Sistema de arquivo versionado



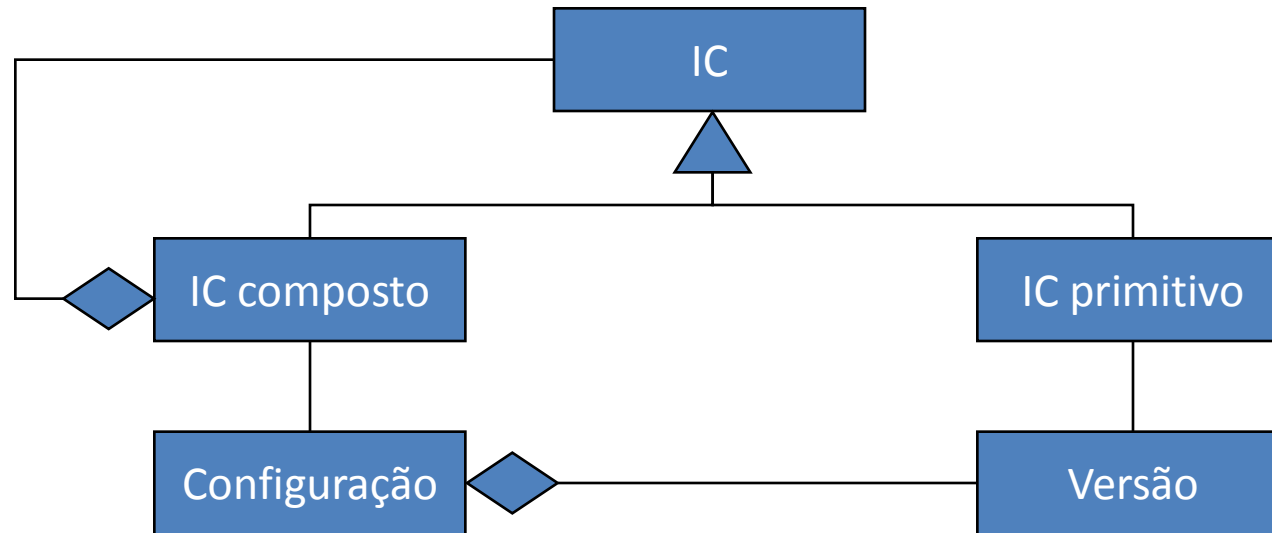
Sistema de Gerência de Configuração



Configuração

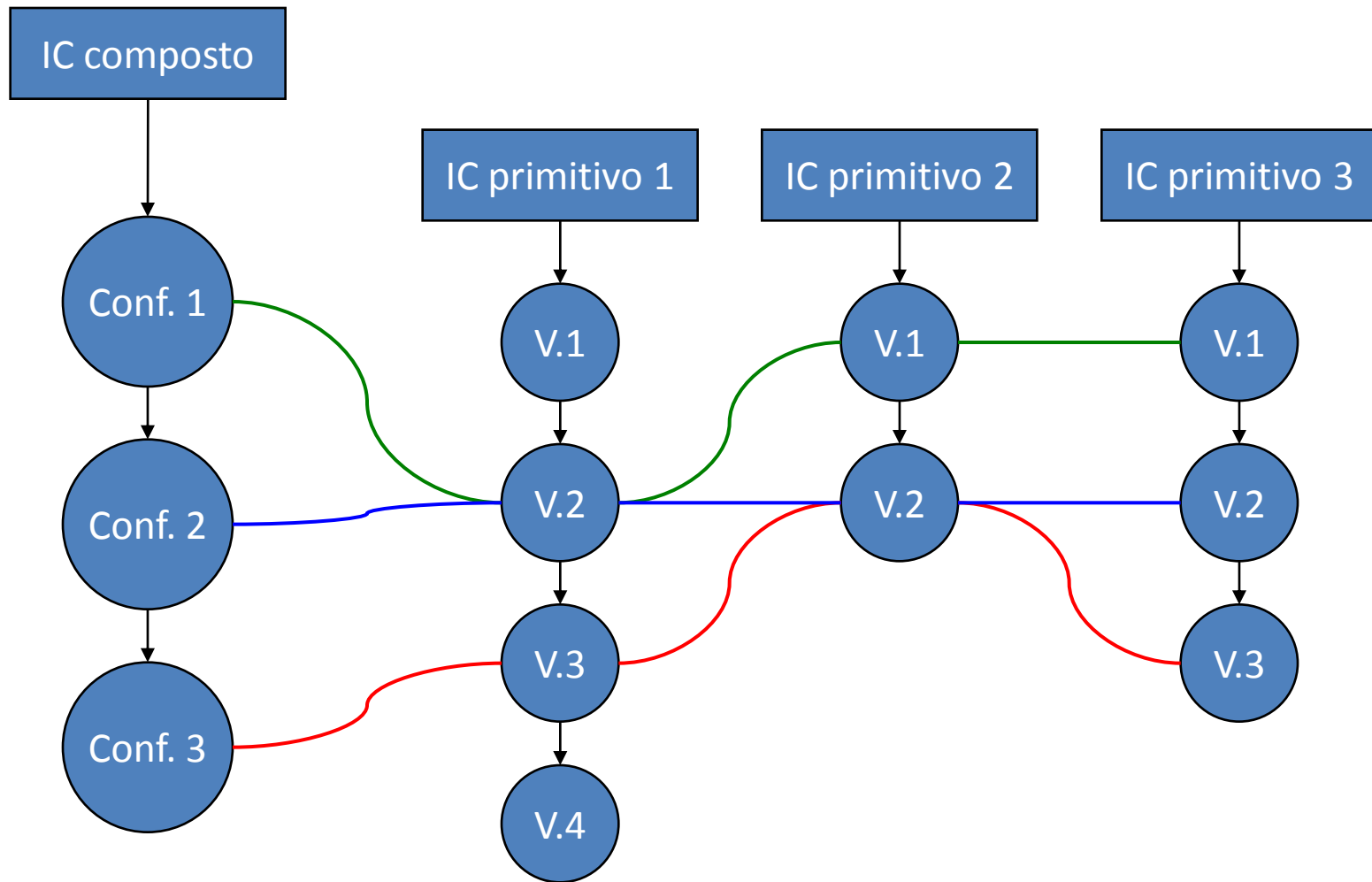
- Um conjunto de versões de Itens de Configuração (IC), onde existe somente uma versão selecionada para cada IC do conjunto
- Uma configuração pode ser vista como um IC composto de outros ICs
- Exemplos
 - Configuração do sistema
 - Configuração do processo
 - Configuração do módulo X
 - Configuração dos requisitos do sistema
 - Configuração do código fonte

Configuração x versão

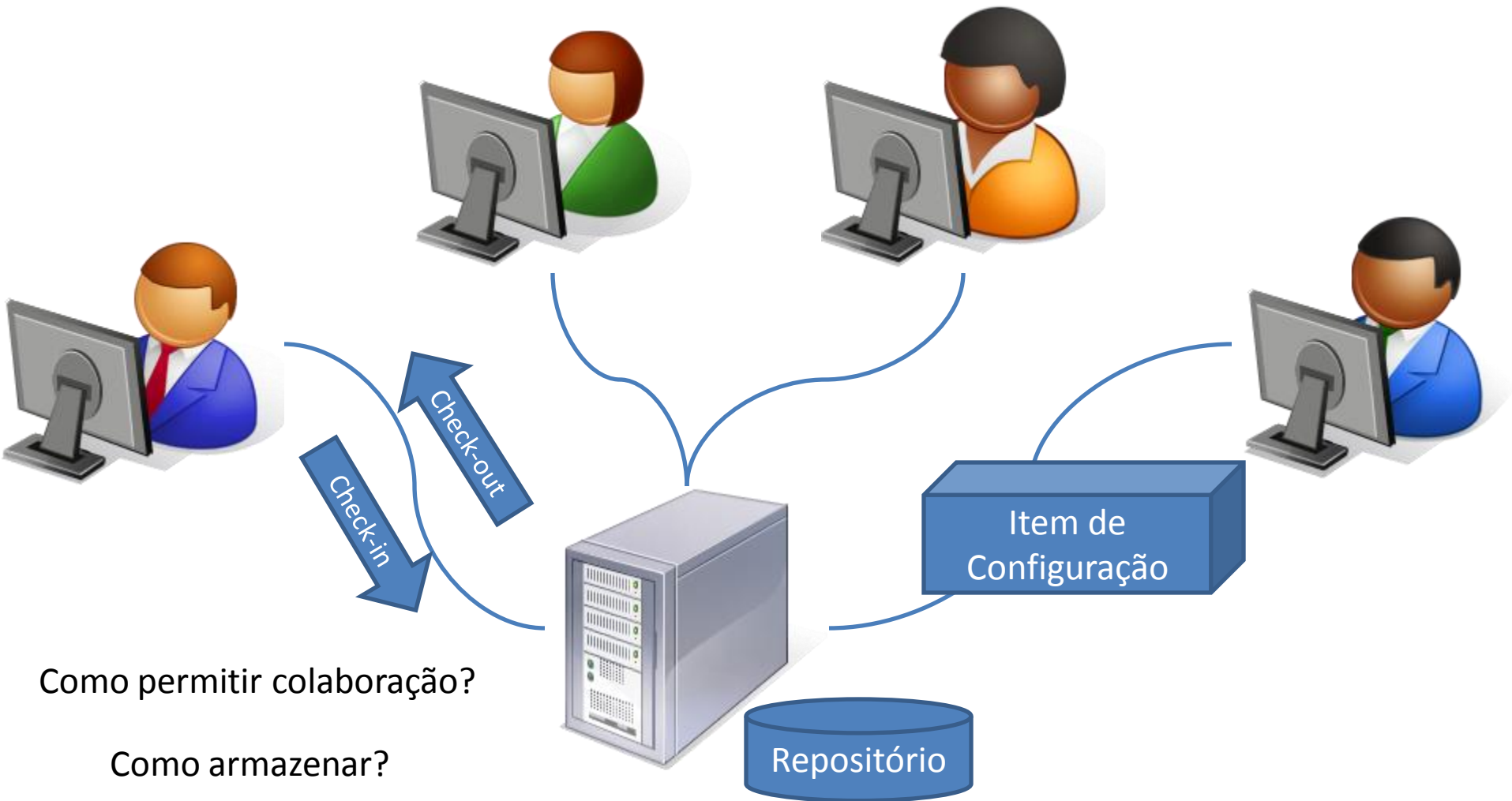


- Genericamente
 - O sistema S é composto pelos arquivos X, Y e Z
- Concretamente
 - A configuração 5 do sistema S é composta pela versão 2 do arquivo X, versão 4 do arquivo Y e versão 6 do arquivo Z

Configuração x versão



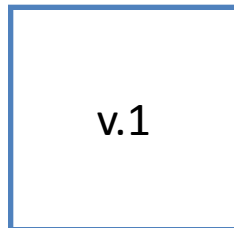
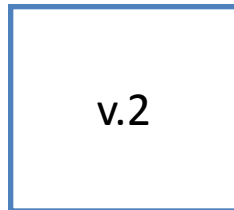
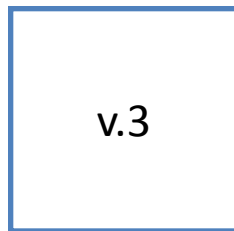
Controle de versões



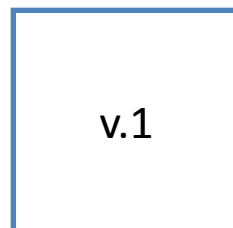
O Repositório

- Local onde os ICs são armazenados
 - Armazena o histórico do projeto
 - Controle na entrada e saída de ICs
 - Poucos por projeto (normalmente, somente um)
- Utiliza diferentes mecanismos de armazenamento
 - Versionamento completo
 - Versionamento de diferenças (delta)
- Utiliza diferentes mecanismos de controle de concorrência
 - Pessimista
 - Otimista
 - Misto
- Permite a geração de diferentes relatórios
 - Por arquivo
 - Por check-in

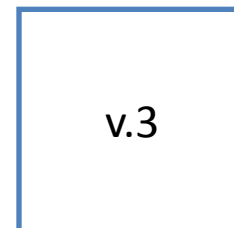
Mecanismos de armazenamento



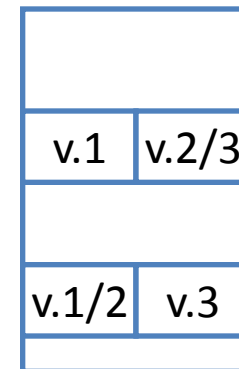
Completo



Forward

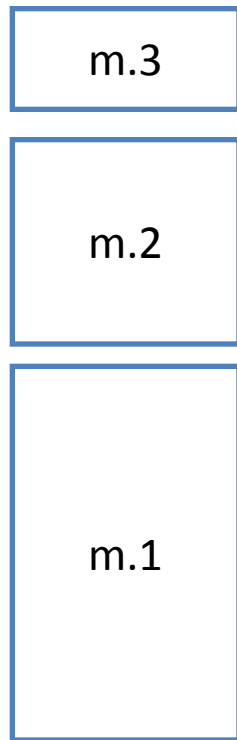


Reverse

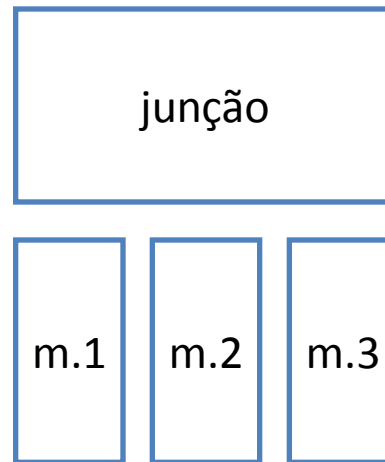


In-line

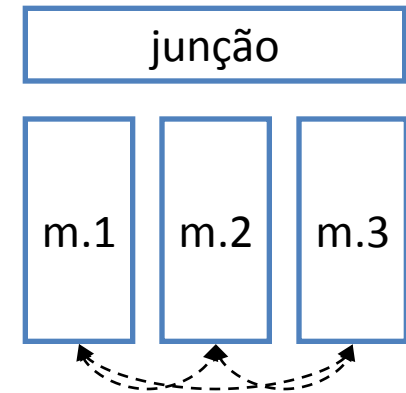
Controle de concorrência



Pessimista



Otimista

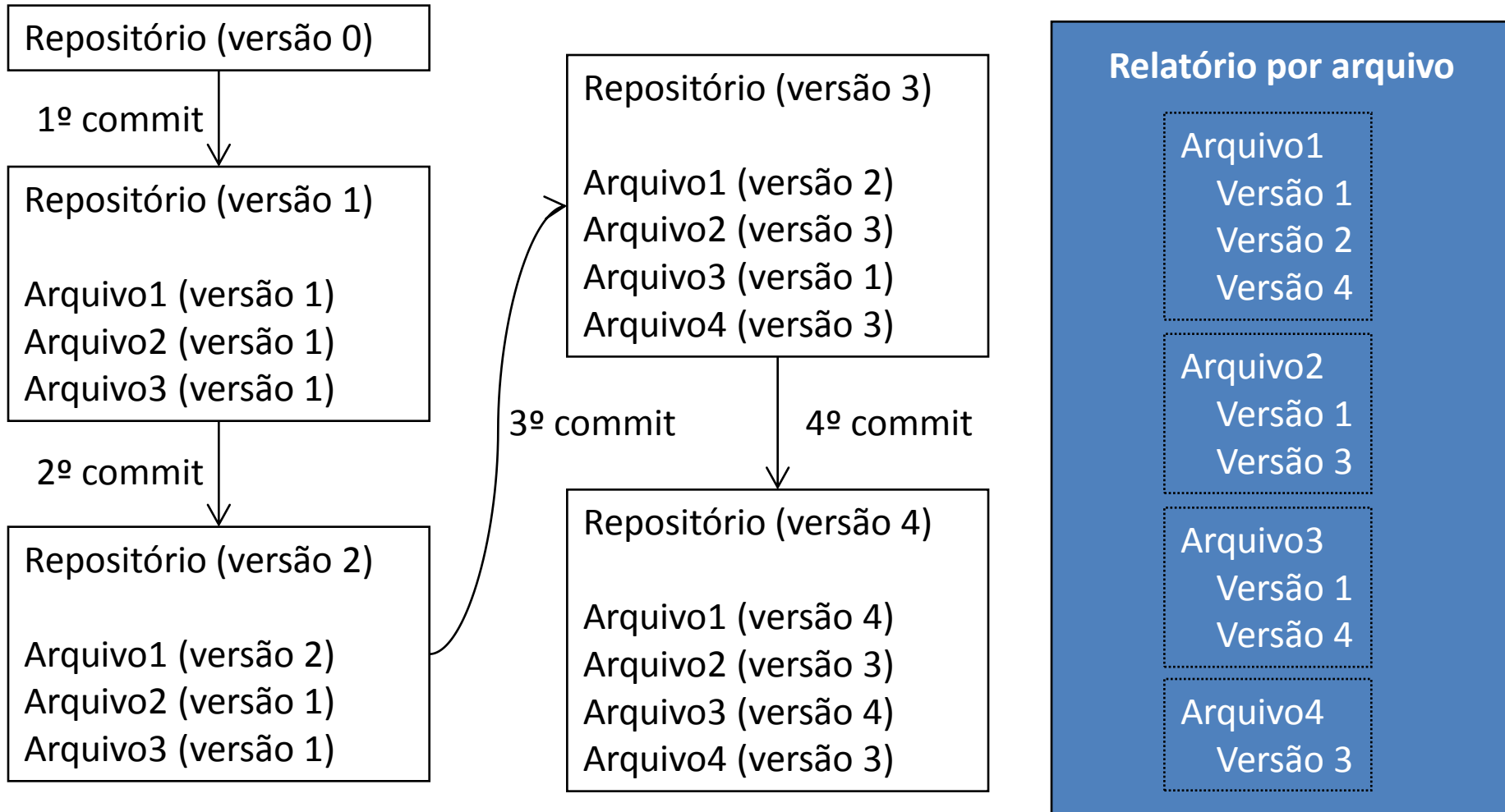


Misto

Relatórios

- É possível fornecer relatórios sob diferentes dimensões
 - Por arquivo
 - Por check-in
- Informações a serem relatadas
 - Identificação (número da modificação)
 - Quem (autor)
 - Quando (data)
 - Onde (caminhos)
 - Como (ação nos caminhos)
 - O que (mensagem)
 - Por que (número da solicitação de modificação)

Exemplo (relatórios no Subversion)



Exemplo

(relatórios no Subversion)

Sintaxe

```
svn log [-q] [-v] [-r VERSÃO] [URL]
```

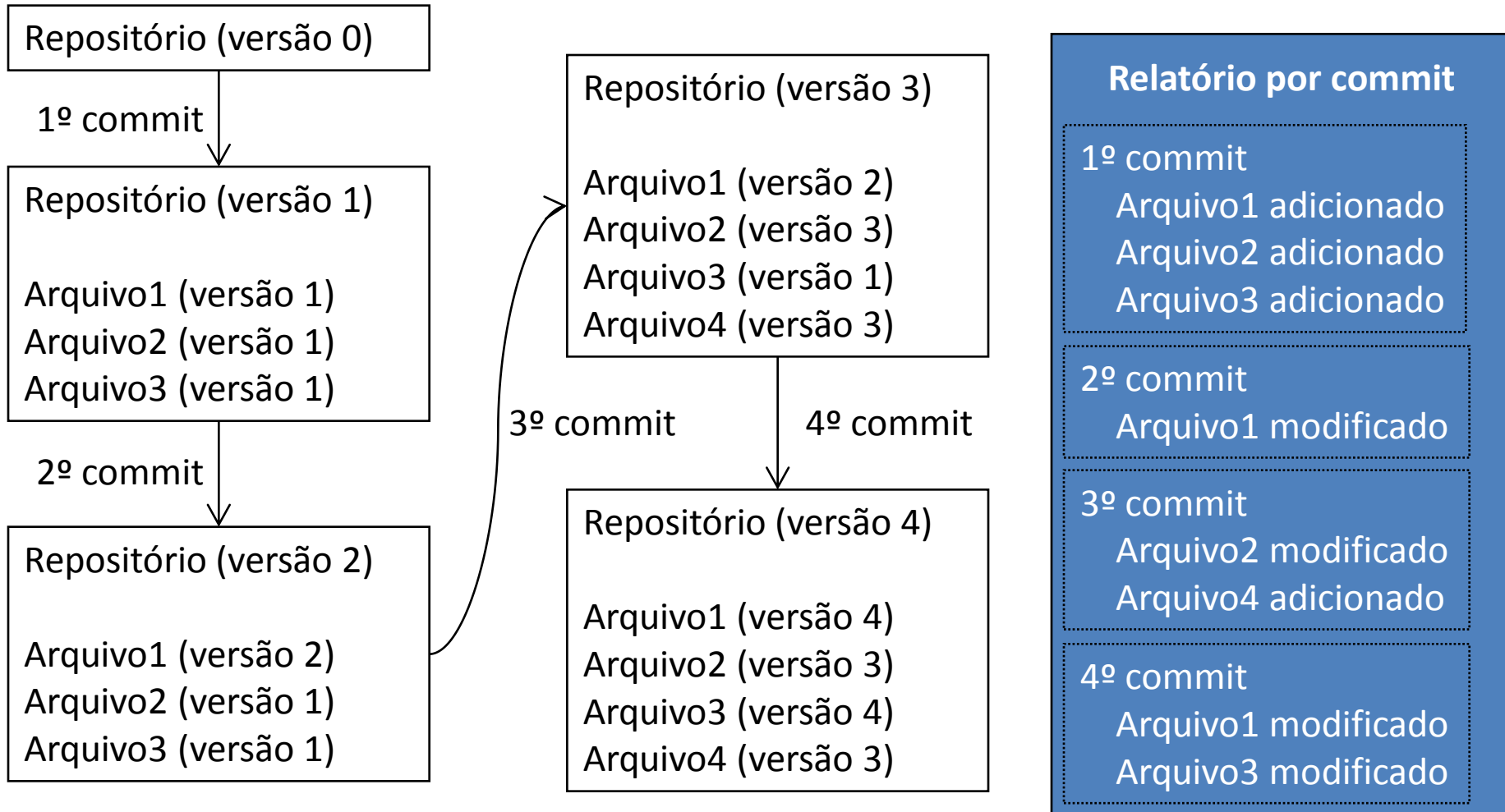
Exemplo

```
svn log -q https://reuse.cos.ufrj.br/svn/brecho/trunk/build.xml
```

```
r92 | ronaldo | 2007-04-01 17:28:55 -0300 (dom, 01 abr 2007)
r91 | paulacibele | 2007-03-19 12:53:47 -0300 (seg, 19 mar 2007)
r90 | paulacibele | 2007-03-19 12:44:20 -0300 (seg, 19 mar 2007)
r51 | marinho | 2006-01-18 19:03:39 -0200 (qua, 18 jan 2006)
r47 | alexrd | 2006-01-07 10:44:46 -0200 (sáb, 07 jan 2006)
r37 | mlopes | 2005-09-27 00:46:04 -0300 (ter, 27 set 2005)
r31 | alexrd | 2005-09-12 11:15:33 -0300 (seg, 12 set 2005)
```

...

Exemplo (relatórios no Subversion)



Exemplo

(relatórios no Subversion)

Sintaxe

```
svn log [-q] [-v] [-r VERSÃO] [URL]
```

Exemplo

```
svn log -v -r 92 https://reuse.cos.ufrj.br/svn/brecho
```

```
r92 | ronaldo | 2007-04-01 17:28:55 -0300 (dom, 01 abr 2007) | 1 line
```

Caminhos mudados:

```
M /trunk/build.xml
```

```
M /trunk/src/br/ufrj/cos/reuse/biblioteca/category/CategoryUtil.java
```

```
A /trunk/src/br/ufrj/cos/reuse/biblioteca/category/Suggestions.java
```

```
M /trunk/src/br/ufrj/cos/reuse/biblioteca/category/dao/HibernateCategoryDAO.java
```

Issue #234: Troca do algoritmo de sugestão de categorias

Espaço de trabalho

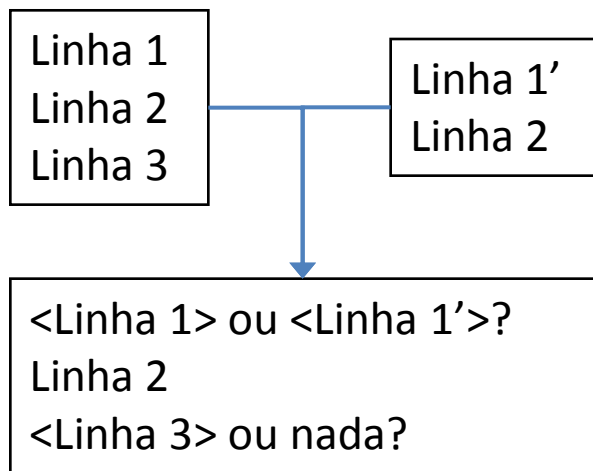
- Local onde o usuário pode fazer seu trabalho de forma isolada
 - Armazena um momento específico do projeto
 - Controle sobre quando sincronizar com o repositório
 - Muitos por projeto (normalmente, um ou mais por desenvolvedor)
 - Sinônimo: caixa de areia (sandbox)

Ramos (branches)

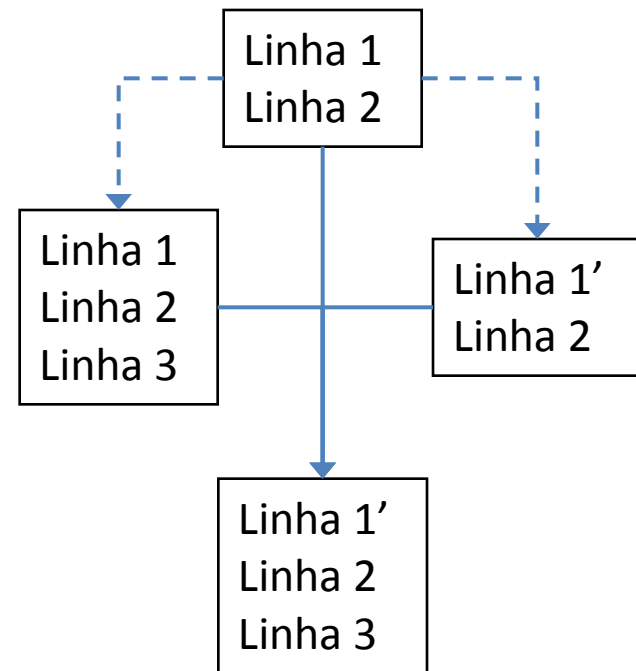
- Versões que não seguem a linha principal de desenvolvimento
- Fornecem isolamento para o processo de desenvolvimento
 - Ramos usualmente são migrados à linha principal de desenvolvimento
 - A migração pode ser complicada no caso de isolamento longo
- Características dos ramos se comparados a espaços de trabalho
 - compartilhados por outras pessoas (espaços de trabalho são isolados)
 - residem no servidor (espaços de trabalho residem no cliente)
 - históricos (espaços de trabalho são momentâneos)
 - permanentes (espaços de trabalho temporários)

Junção

- Processo de migração de
 - Espaços de trabalho
 - Ramos



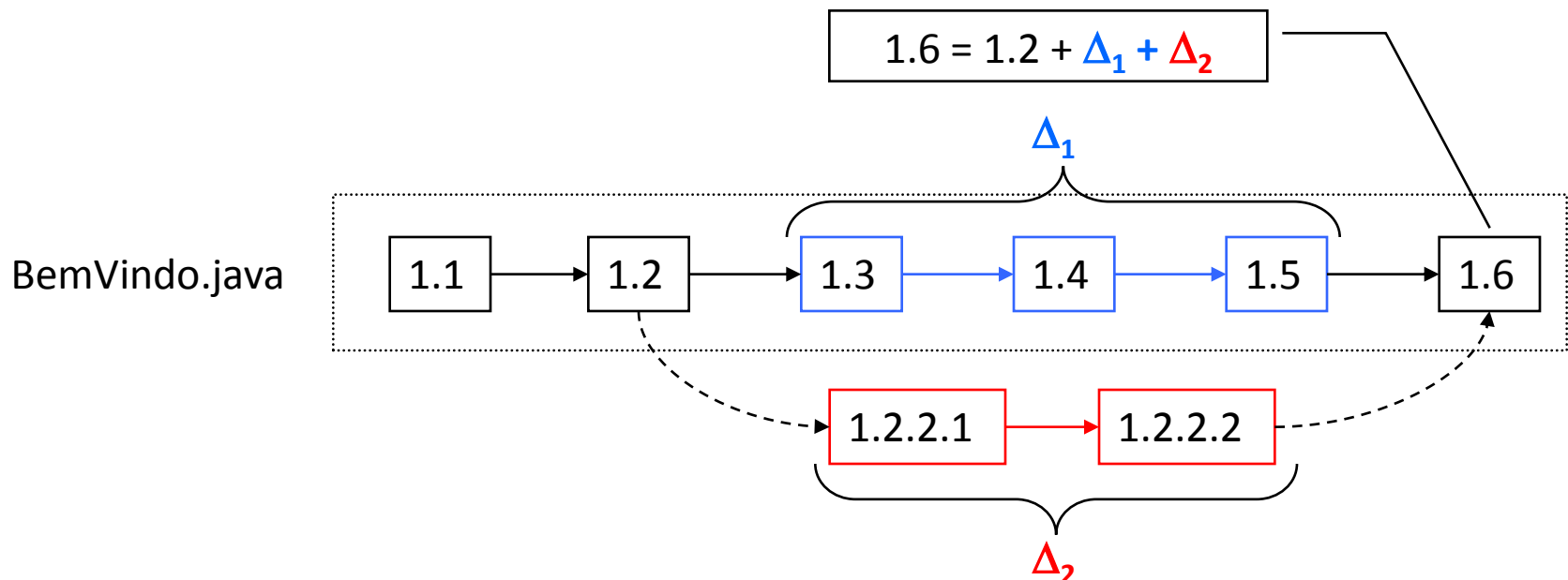
2-way merge



3-way merge

Exemplo de junção

- A junção é efetuada para cada artefato do ramo
- São levadas em consideração todas as modificações desde o ancestral em comum



Exemplo (junção no Eclipse)

Java - AloMundo.java - Eclipse Platform

File Edit Source Refactor Navigate Search Project Run Window Help

Synchronize - Incoming Mode

Structure Compare

- alomundo [sety.cos.ufrj.br]
 - src
 - AloMundo.java (ASCII -kkv)

Java Structure Compare

- Compilation Unit
 - AloMundo
 - main(String[])

Java Source Compare

Common file: AloMundo.java 1.1.1.1

```

public static void main(String[] args
    System.out.println("Oi, Alo Munde
}

```

Repository file: AloMundo.java 1.2 [joao]

```

public static void main(String[]
    // Escreve uma mensagem em 2
    System.out.println("Oi, ");
    System.out.println("Alo Mundo
}

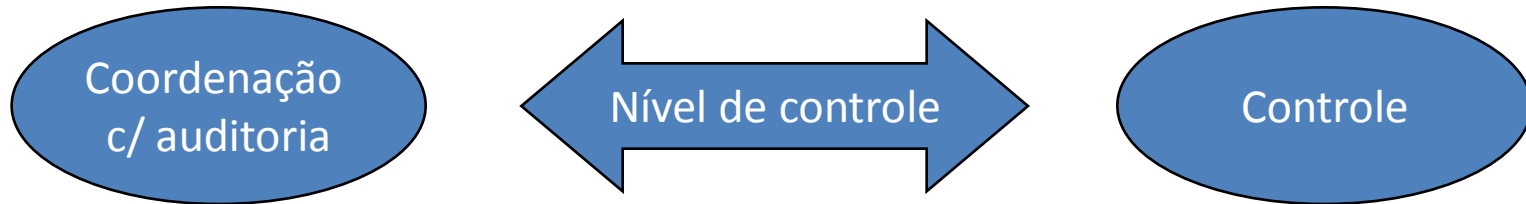
```

1 conflicts, no incoming changes, no outgoing changes, no new resources.

Baseline

- Configuração revisada e aprovada que serve como base para uma próxima etapa de desenvolvimento e que somente pode ser modificada via processo formal de GCS
- São estabelecidas ao final de cada fase de desenvolvimento
 - Análise (functional)
 - Projeto (allocated)
 - Implementação (product)
- Momento de criar: balanceamento entre controle e burocracia

Baseline (níveis de controle)



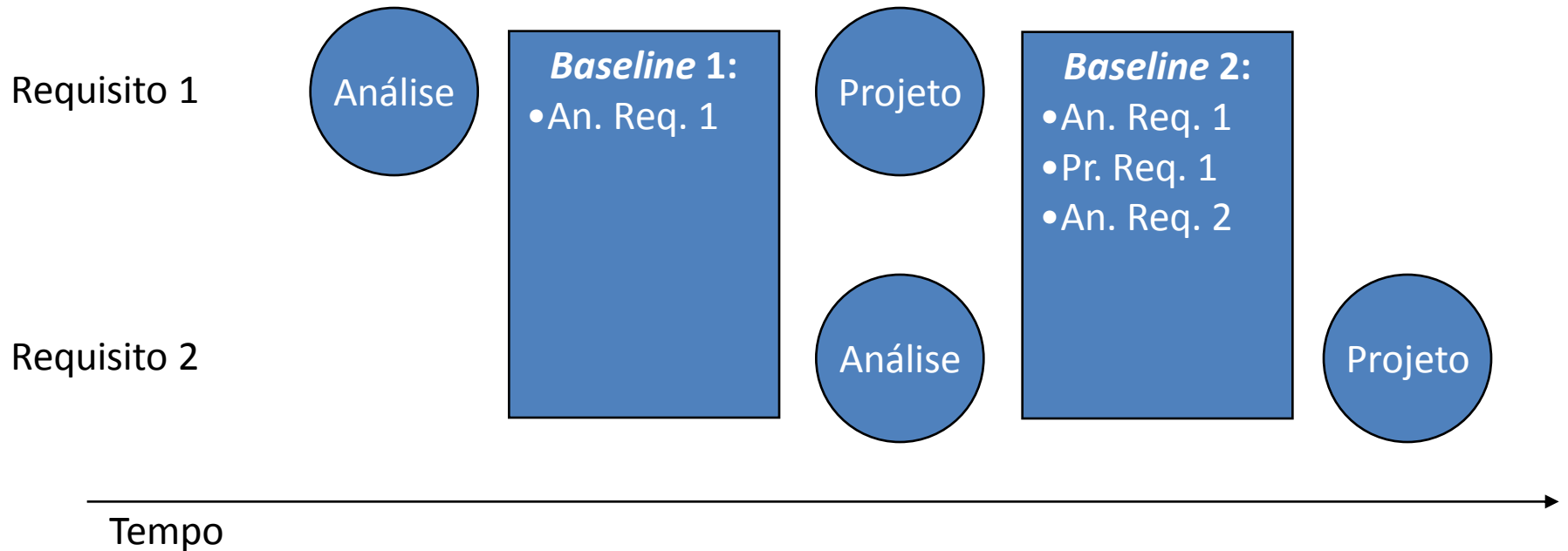
Informal:

- Pré *baseline*
- Sem requisição
- Sem aprovação
- Sem verificação
- Ágil
- Ad-hoc

Formal:

- Pós *baseline*
- Com requisição
- Com aprovação
- Com verificação
- Burocrático
- Planejado

Baseline (níveis de controle)



Req.
1
2

Análise	Projeto
Inform.	-
-	-

Análise	Projeto
Formal	Inform.
Inform.	-

Análise	Projeto
Formal	Formal
Formal	Inform.

Controle de modificações

- Tarefas
 - Solicitação de modificação
 - Classificação da modificação
 - Análise da modificação
 - Avaliação da modificação
 - Implementação da modificação
 - Verificação da modificação
 - Geração de baseline

Controle de modificações

CHANGE REQUEST

CR No.: _____

Analysis Document No.: _____

System/project: _____ Item to be changed: _____

Classification: Enhancement / Bug fixing / Other: _____

Priority: Immediate / Urgent / As soon as possible / Desirable

Change Description:

[Leon, 2000] Solicitação de modificação

Status	Date	By	Remarks
Initiated			
Received			
Analyzed			
Action (A / R / D)*			
Assigned			
Check-out			
Modified and tested			
Reviewed			
Approved			
Check-in			
Baselined			

Controle de modificações

- O critério de classificação da modificação deve ser explicitado no planejamento
- A classificação visa priorizar modificações mais importantes
 - Críticas
 - Fatais
 - Não fatais
 - Cosméticas

Controle de modificações

- A análise visa relatar os impactos da implementação da modificação
 - Custo
 - Cronograma
 - Funcionalidades
- Caso a análise conclua que não existe chance de aprovar a modificação (casos extremos), pode ocorrer rejeição antes da avaliação para poupar custos no processo

Controle de modificações

- A avaliação utilizará a solicitação de modificação e o laudo da análise para tomar a decisão
 - Aceitar
 - Rejeitar
 - Adiar
- Essa decisão é tomada pelo Comitê de Controle da Configuração (CCC)
 - Líder do projeto e de gerência de configuração
 - Representantes da garantia de qualidade, marketing e cliente

Controle de modificações

- A implementação deve ser seguida por testes de unidade
- Durante a verificação, devem ser aplicados testes de sistema
- Após a geração da nova baseline, deve ser decidido pelo CCC se ela será considerada uma nova liberação

Controle de modificações

- Caso especial: Correções emergenciais
 - No caso de correções emergenciais, podem ser criados ramos sem a necessidade do processo formal
 - Em algum momento esses ramos deverão sofrer junção para a linha principal de desenvolvimento
 - Esse procedimento deve estar explicitado no processo!

Controle de modificações

- Caso especial: Defeitos
 - Alguns sistemas tratam defeitos de forma diferente das demais requisições
 - A correção de defeitos é um tratamento sintomático
 - É importante descobrir o real motivo para o acontecimento do defeito para possibilitar a prevenção de defeitos futuros
 - A análise de causa é útil para descobrir falhas no processo de desenvolvimento (e.g. falta de treinamento, padrões inadequados, ferramentas inadequadas)

Agenda

- Introdução
- Repositório na perspectiva de Gerência de Configuração
- **Repositório na perspectiva de Reutilização**
- Integração dos repositórios
- Conclusão

Exemplos utilizados



Biblioteca Brechó

Reutilização de Software

Reutilização de Software é uma **disciplina** responsável pela **criação de sistemas de software a partir de software preexistente**

Krueger

- Principais processos (Jacobson, 1997):
 - Gerenciamento (programa de reutilização)
 - Criação (engenharia de domínio)
 - **Apoio (repositório)**
 - Reutilização (engenharia de aplicação)

Ativo reutilizável

- Qualquer artefato relacionado a software que esteja preparado, isto é, empacotado de maneira própria a ser reutilizado pelos processos da organização
- Tipos de ativos
 - Horizontais (apoio)
 - Verticais (domínios específicos)
- Exemplos: código, binários, modelos, processos, etc.

Tipos de repositório

- Repositório local
 - Guarda componentes de uso genérico
- Repositórios de domínio
 - Guarda componentes específicos de um determinado domínio
- Repositórios de referência
 - Funciona como um catálogo
 - Guarda as características dos componentes
 - Referencia repositórios externos

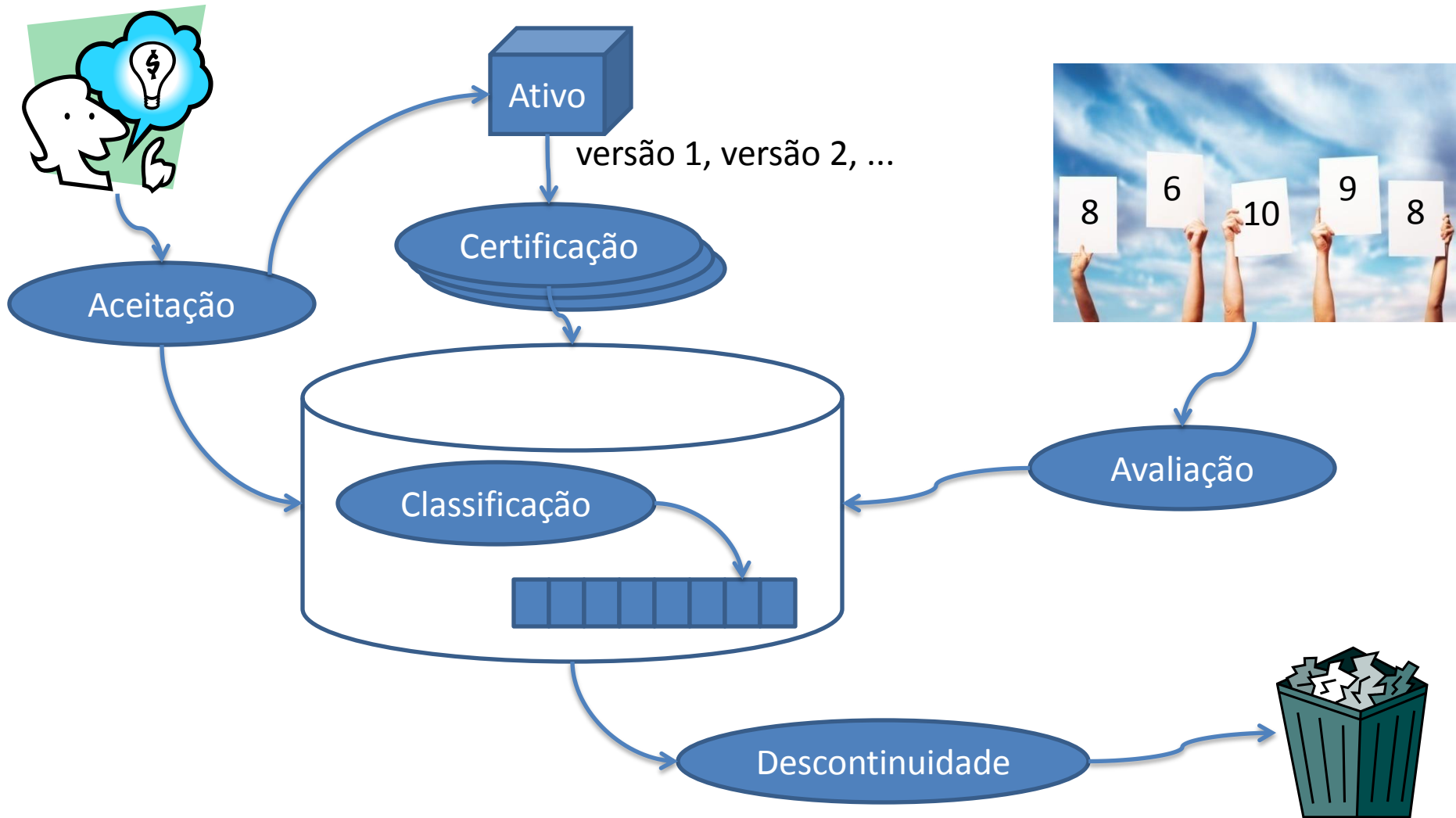
Papéis envolvidos

- Produtor
 - Desenvolve artefatos para serem disponibilizados no repositório
- Consumidor
 - Busca por artefatos no repositório
 - Fornece *feedback* sobre os artefatos
- Gerente
 - Cataloga os artefatos do repositório
 - Aplica os critérios definidos sobre os artefatos do repositório

Atividades em um repositório

- Gerenciamento
 - Como determinar o que deve ser aceito no repositório?
- Catalogação
 - Como organizar os ativos dentro do repositório?
- Busca
 - Como descobrir os ativos disponíveis?
- Recuperação
 - Como obter um empacotamento de determinados ativos?
- Análise
 - Como entender a dinâmica de utilização do repositório?

Gerenciamento de ativos



Critérios de Aceitação

- Estabelecem os atributos que credenciam a existência de um ativo no repositório
- Atua no nível conceitual do ativo
 - Propósito do ativo
- Ex.: Fazer parte de um domínio de interesse da organização

Critérios de Certificação

- Estabelecem os atributos que credenciam a inclusão de uma versão concreta do ativo no repositório
- Atua no nível físico do ativo
 - Artefatos que compõem uma versão específica do ativo
 - Deve ser repetido para cada nova versão do ativo
- Ex.: Aprovação nos testes antes de inclusão no repositório

Critérios de Classificação

- Estabelecem a organização dos ativos no repositório
- Fundamental para posterior busca do ativo
- Ex.: Classificação quanto ao tipo (código, modelo, processo, etc.) ou quanto ao domínio (agropecuária, telecomunicações, etc.) do ativo

Critérios de Descontinuidade

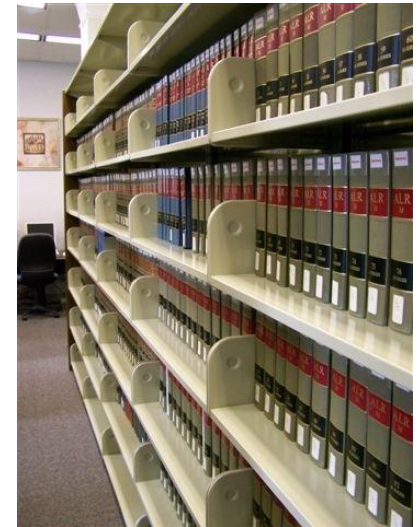
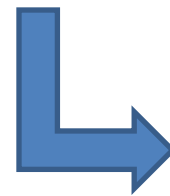
- Estabelecem quando os ativos devem ser removidos do repositório
- Ativos inúteis no repositório podem ser custosos
 - Recursos consumidos pelo repositório
 - Custos de manutenção dos ativos
 - Diminuição da precisão das buscas
- Ex.: Ausência de ocorrências de uso por um determinado período

Critérios de Avaliação

- Estabelecem os atributos que definem o grau de qualidade de um ativo
- Podem ser utilizados
 - Na certificação, pelo gerente do repositório
 - Após a utilização, pelos consumidores
- Ex.: Grau de satisfação do uso de uma determinada versão de um ativo

Mecanismos de Catalogação

- Lista
- Palavras-chave
- Vocabulário controlado
 - Estruturas enumeradas
 - Facetas
- Conhecimento
 - Redes semânticas
 - Regras de produção
 - Frames
- Hipertextos



Lista

- Não faz uso de nenhum mecanismo de catalogação
- Facilita a incorporação de ativos na repositório
- Dificulta a busca posterior por ativos em repositórios grandes
 - Problema pode ser minimizado com o uso de mecanismos de busca aprimorados



Palavras-chave

- Classifica o ativo em termos de palavras-chave informadas
- Não restringe o vocabulário a ser utilizado
- Tem baixa precisão semântica

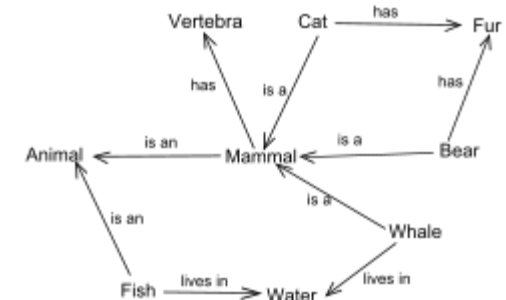


Vocabulário controlado

- Estruturas enumeradas
 - Classificação do ativo em categorias exclusivas sob uma única dimensão
 - Opções de classificação predefinidas
 - Necessita alto conhecimento sobre o domínio
- Facetas
 - Classificação do ativo sob múltiplas dimensões
 - Dimensões possíveis são predefinidas

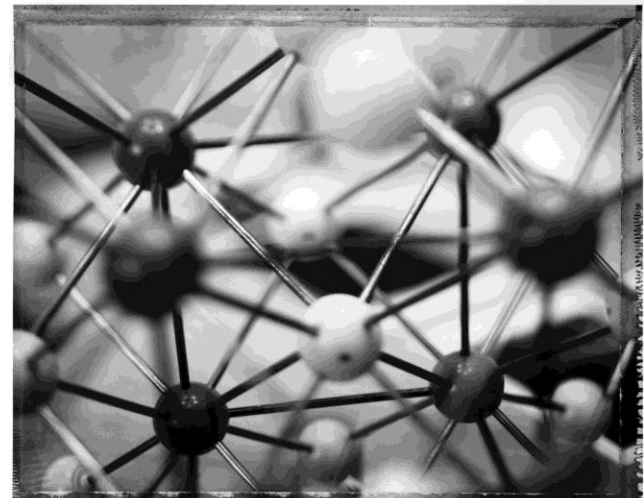
Conhecimento

- Redes semânticas
 - Expressam relações entre os ativos
 - Baixo suporte a heurísticas
- Regras de produção
 - Amplamente utilizado em sistemas especialistas
 - Permite estabelecer regras para inferência sobre ativos
- Frames
 - Ativos representados com relações nome:valor
 - Uso de herança como mecanismo de inferência



Hipertexto

- Ativos são relacionados sob diferentes dimensões
 - Ativos similares
 - Ativos complementares
 - Dependências entre ativos
 - Exclusões mutuas entre ativos
 - Relações todo-parte entre ativos
 - Etc.
- Normalmente complementa algum outro mecanismo de catalogação



Exemplo (cadastro de formulários)

Biblioteca de Componentes
Grupo de Reutilização de Software

Olá, supervisor
Sair

Detalhes do formulário

abc

label for text (*)

label for textarea

label for file Procurar...

label for radio radio value 1 radio value 2

label for checkbox checkbox value 1 checkbox value 2 checkbox value 3
 checkbox value 4 checkbox value 5

label for select select value 1
 select value 2

solitaire checkbox value

(*) Campos obrigatórios

Início
Listar formulários
Instalar formulário

Listar categorias
Nova categoria

Listar usuários
Novo usuário
Editar meu perfil

Concluído Intranet local

Exemplo (cadastro de categorias)

Biblioteca de Componentes - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço <http://localhost/bib/category/showCategoryAction.do?method=edit> Ir

Links [Google](#) [O Globo On Line](#) [Yahoo! Groups](#) [Yahoo! Mail](#) [Gmail](#) [Banco Real](#) [Bco do Brasil](#) [Dicionário Português](#) [Babel Fish](#) [Merriam-Webster](#)

Biblioteca de Componentes
Grupo de Reutilização de Software

Olá, supervisor [Sair](#)

Cadastro de nova categoria


Nome(*)

Descrição

Formulários associados abc

(*)Campos obrigatórios

Início
[Listar formulários](#)
[Instalar formulário](#)
[Listar categorias](#)
[Nova categoria](#)
[Listar usuários](#)
[Novo usuário](#)
[Editar meu perfil](#)



Intranet local

Exemplo

(catalogação do componente)

Biblioteca de Componentes - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://localhost:8080/brecho/component/editComponentAction.do?method=edit&id=1

Biblioteca de Componentes
Grupo de Reutilização de Software

Olá, supervisor
Sair

Edição de dados de componentes

Excluir este componente

Nome(*)

Descrição(*)

Release

Executable

URL Implantada

Arquivo(*) Procurar...

Categorias associadas
 J2EE J2ME

Licenças associadas
 Free GPL Shareware

(*)Campos obrigatórios

Listar formulários
Instalar formulário

Listar categorias
Nova categoria

Listar licenças
Nova licença

Meus componentes
Novo componente

Listar usuários
Editar meu perfil

UFPE
COPPE

Intranet local

Mecanismos de Busca

- Navegação
- Recuperação da informação
- Palavras-chave
- Estrutural
- Semântica
 - Dados
 - Funcionalidades
- Baseado em conhecimento



Navegação

- Não faz uso de consultas
- Permite que se recupere ativos relacionados a partir de outros ativos
 - Uso da organização usada na catalogação (e.g., hipertexto, estruturas enumeradas, facetas, etc.)
 - Uso de navegação seqüencial (força bruta)
- Usualmente é um mecanismo de busca auxiliar

Recuperação da informação

- Usualmente se baseia em pré-análise dos ativos
 - Construção de base de conhecimento
- Consultas são tratadas e comparadas com a base de conhecimento
 - Uso de linguagem natural e operadores lógicos
- Resultados são ordenados em função da proximidade com a consulta
 - Precisão: Fração dos ativos recuperados que são relevantes
 - Revocação: Fração dos ativos relevantes que foram recuperados

Palavras-chave

- Ativos foram previamente catalogados em termos de palavras-chave
- Consultas são feitas em termos de palavras-chave
- Ativos e consulta são relacionados em função do número de palavras chave em comum
- Necessita de mecanismos extras para unificar a terminologia usada na catalogação e consulta
 - Thesaurus
 - Ontologias
- Amplamente utilizado devido a simplicidade

Estrutural

- Leva em consideração a estrutura interna do ativo
 - Crença: estruturas parecidas levam a funcionalidades parecidas
 - Demanda acesso à estrutura interna dos ativos
- Consulta consistem em uma breve especificação estrutural
- Resultado consiste nos ativos que necessitam de menos modificações estruturais para atenderem à consulta

Semântica (dados)

- Leva em consideração o aspecto executável do ativo
 - Analisa os dados requeridos e providos
- Consultas consistem em exemplos de uso dos ativos desejados
- Resultados consistem em ativos que melhor atendem aos exemplos

Semântica (funcionalidades)

- Leva em consideração os serviços providos pelo ativo
 - Assinaturas de interfaces providas
 - Especificação formal do comportamento dessas interfaces
- Consultas consistem em assinaturas desejadas ou especificações algébricas do comportamento desejado
- Resultados consistem em ativos que melhor atendem aos critérios de consulta
 - Casamento de interfaces
 - Prova de teoremas

$$(\text{pre}_{\text{consulta}} \Rightarrow \text{pre}_{\text{resultado}}) \wedge (\text{pre}_{\text{consulta}} \wedge \text{pos}_{\text{resultado}} \Rightarrow \text{pos}_{\text{consulta}})$$

Baseado em conhecimento

- O sistema interage com o usuário fazendo perguntas
- Ao final, são sugeridos ativos ou informado que não existem ativos com aquele perfil
- Caso não existam ativos disponíveis, o gerente do repositório é notificado

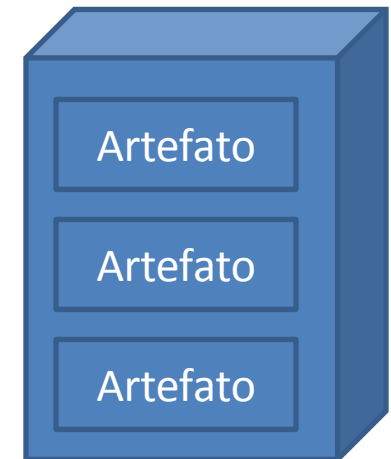
Recuperação

- A partir dos resultados da busca, o ativo deve ser disponibilizado para o Cliente
 - Via referência
 - Via empacotamento
 - Via serviço
- Tarifação se torna mecanismo chave para recuperação de ativos em repositórios comerciais



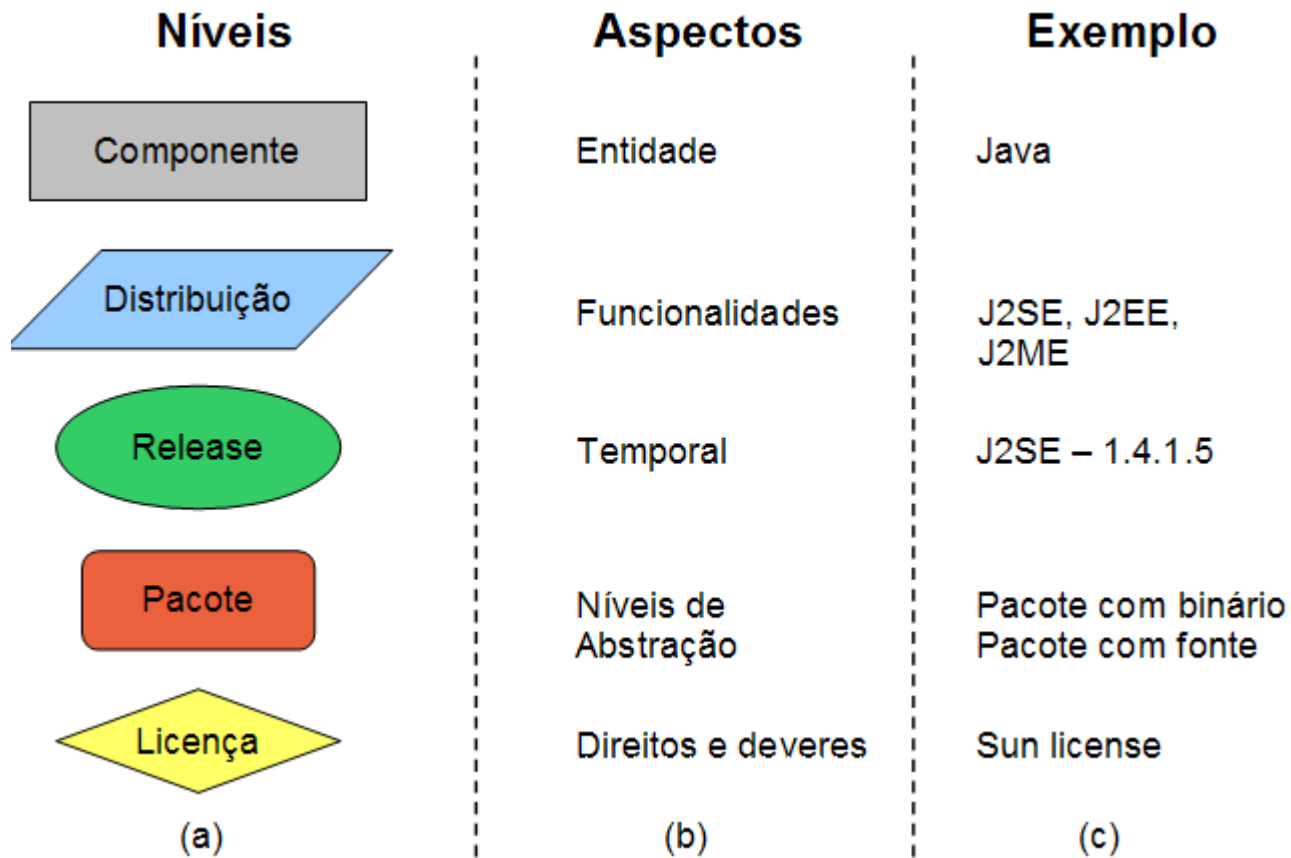
Empacotamento

- Composição do ativo em função das demandas do cliente
- Empacotamento consiste na composição de ativos a partir de
 - Artefatos
 - Licenças
 - Descrição do empacotamento
- Mecanismo útil para equacionar relação custo/benefício



Ativo


Exemplo (níveis de organização)



Exemplo (cadastro de release)

Brechó - Biblioteca de Componentes
Grupo de Reutilização de Software

Olá, Anderson
Sair

Meus Créditos  Meu Carrinho

Cadastro de nova release

Nome(*)

Descrição

Artefato Src

Arquivo Preço

Artefato Bin

Arquivo Preço

(*)Campos obrigatórios

Listar categorias

Listar licenças

Meus componentes
Novo componente



Minhas avaliações
Editar meu perfil



Exemplo (cadastro de pacote)

Brechó - Biblioteca de Componentes
Grupo de Reutilização de Software

Olá, Anderson
Sair

 Meus Créditos  Meu Carrinho

Edição de dados de pacotes

Nome(*)

Descrição

Preço(*)

Artefatos associados

bin src

Licença default

Licenças associadas

suporte telefônico por 1 ano manutenção por 1 ano

GPL Open Source

Listar categorias

Listar licenças

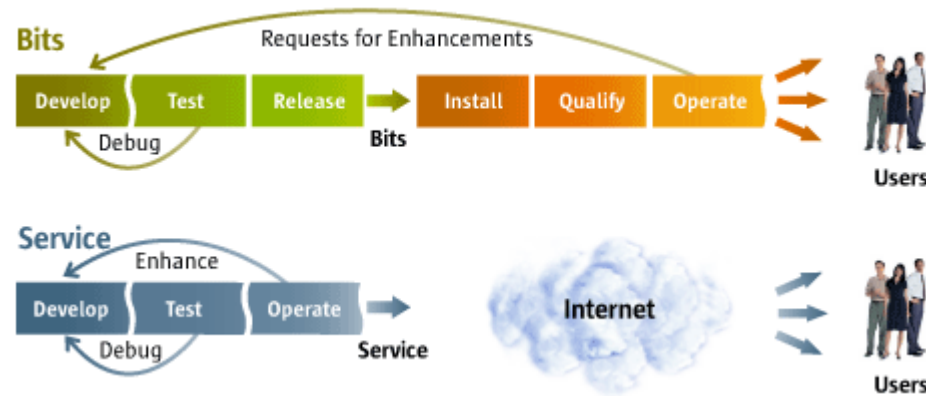
Meus componentes
Novo componente

Minhas avaliações
Editar meu perfil



Serviços

- Passo no sentido de SaaS (*Software as a Service*)
 - Serviços externos podem ser catalogados no repositório
 - Ativos existentes no repositório podem ser transformados em serviços





[Sun Microsystems]

Exemplo (geração de serviços)

Brechó - Biblioteca de Componentes
Grupo de Reutilização de Software

Olá, Anderson
Sair

 Meus Créditos  Meu Carrinho

Cadastro de novo serviço



Selecione as operações:

OperacoesGeometricas

- tan(double)
- tanh(double)
- seno(double)
- senoh(double)
- cosseno(double)
- cossenoh(double)

OperacoesGerais

- log(double)
- min(int, int)
- max(int, int)
- raizQuadrada(double)
- fatorial(int)

 Voltar  Início

Listar categorias

Listar licenças

Meus componentes
Novo componente

Minhas avaliações
Editar meu perfil



Tarifação



- Diferentes possíveis modelos para tarifação de ativos
- Pré-pago
 - paga e depois adquire os ativos
- Pós-pago
 - adquire os ativos e depois paga
- Bancário
 - Uso de créditos, limite e saldo
 - Permite emular modelos pré-pago e pós-pago

Exemplo

(tarifação no modelo bancário)

Brechó - Biblioteca de Componentes
Grupo de Reutilização de Software





Olá, Anderson
Sair



 Meus Créditos  Meu Carrinho

Meus Créditos

Créditos: 100.0
Limite: 200.0
Saldo: 300.0

Últimas Transações:

Tipo	Componente	Valor	Desde	Detalhes
<i>Serviço</i>	Operações Matemáticas	4.0	Jan 10, 2008	
<i>Download</i>	Operações Matemáticas	30.0	Jan 15, 2008	
<i>Download</i>	Odyssey	90.0	Fev 12, 2008	
<i>Download</i>	Odyssey SCM	35.0	Fev 12, 2008	

 Voltar  Início

Listar categorias

Listar licenças

Meus componentes
Novo componente

Minhas avaliações
Editar meu perfil



Análise

- O uso de um repositório gera grande quantidade de informações
 - Registros de utilização
 - Feedbacks dos usuários
 - Coleta de métricas em geral
- Essas informações são importantes para tomadas de decisões
 - Descontinuidade de ativos
 - Notificação de problemas com ativos
 - Notificação de novas versões de ativos

Exemplo

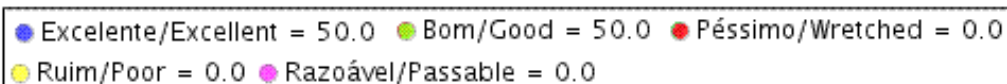
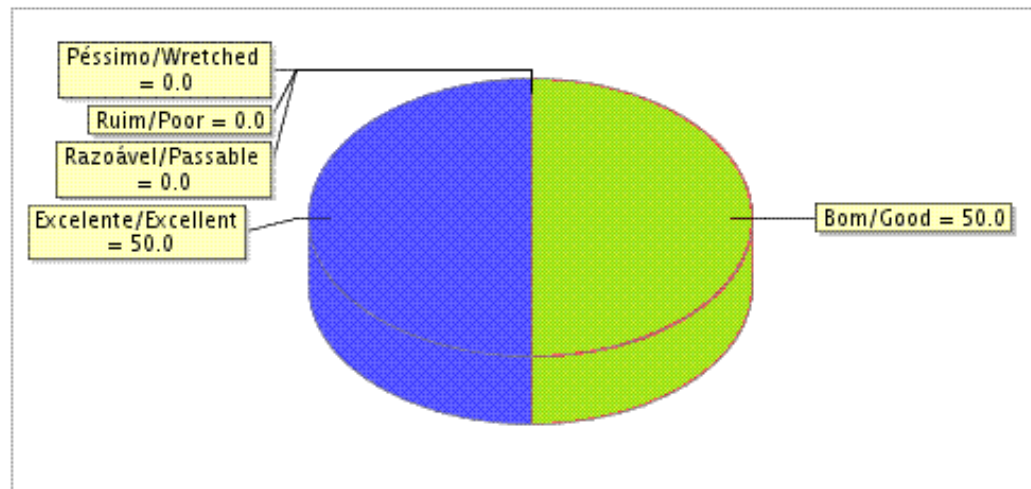
(mapa de reutilização)

Nome	E-mail	Distribuição	Release	Pacote	Licença	Desde
Ana Paula Terra Bacelo	ana.bacelo@pucri.br	Default	1.6.0	Default	Default	Out 10, 2007
Administrador	paulacibele@cos.ufrj.br	Default	1.6.0	Default	Default	Jul 09, 2007
Paula Cibele	paulacibele@cos.ufrj.br	Default	1.6.0	Default	Default	Jul 06, 2007
Lincoln Rocha	lincoln@great.ufc.br	Default	1.6.0	Default	Default	Set 11, 2007
Rafael Krindges	rkrindges@yahoo.com.br	Default	1.6.0	Default	Default	Out 10, 2007
Usuário Odyssey	paulacibele@cos.ufrj.br	Default	1.6.0	Default	Default	Ago 23, 2007
Ronaldo Raposo	ronaldo.raposo@gmail.com	Default	1.6.0	Default	Default	Ago 23, 2007

Exemplo (feedback dos usuários)

Listing evaluations

User	Evaluation	Comment	Since
Usuário Odyssey	Excellent	Gostei tanto do Odyssey que estou participando do desenvolvimento e dos testes para as próximas versões. Simples e muito fácil de usar!	Dez 09, 2007
Ronaldo Raposo	Good	Senti um pouco de dificuldade no início para entender como funciona o Odyssey. Agora, já é bem mais fácil.	Dez 09, 2007



Agenda

- Introdução
- Repositório na perspectiva de Gerência de Configuração
- Repositório na perspectiva de Reutilização
- **Integração dos repositórios**
- Conclusão

Exemplos utilizados



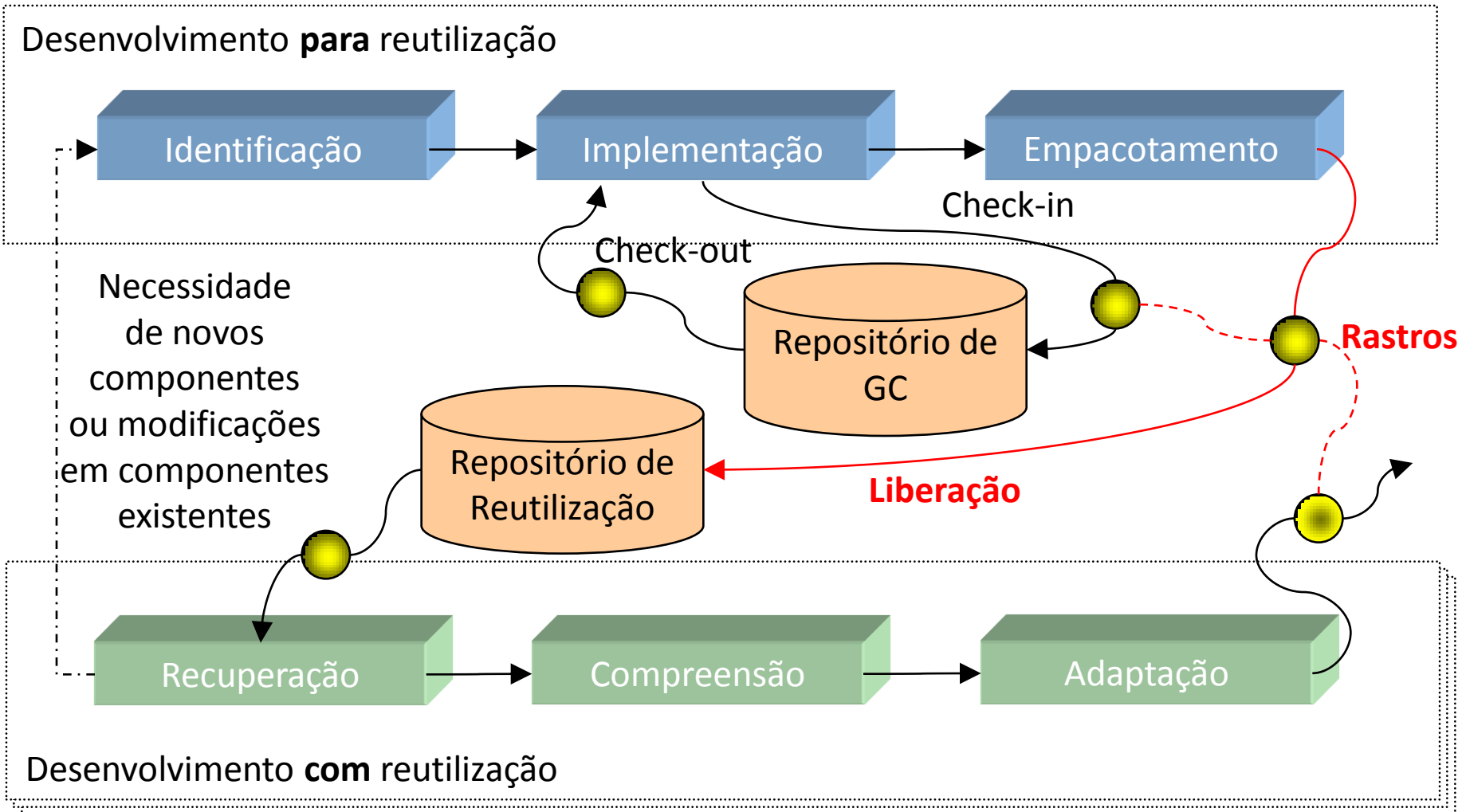
Biblioteca Brechó



Integração dos repositórios

- Apesar das diferenças, os repositórios devem trabalhar de forma integrada
 - Lado a lado
 - Um sobre o outro
- Elemento chave para integração
 - Rastreabilidade
- Funções principais da integração
 - Publicação de releases: Desenvolvimento → Produção
 - Notificação de defeitos: Produção → Desenvolvimento

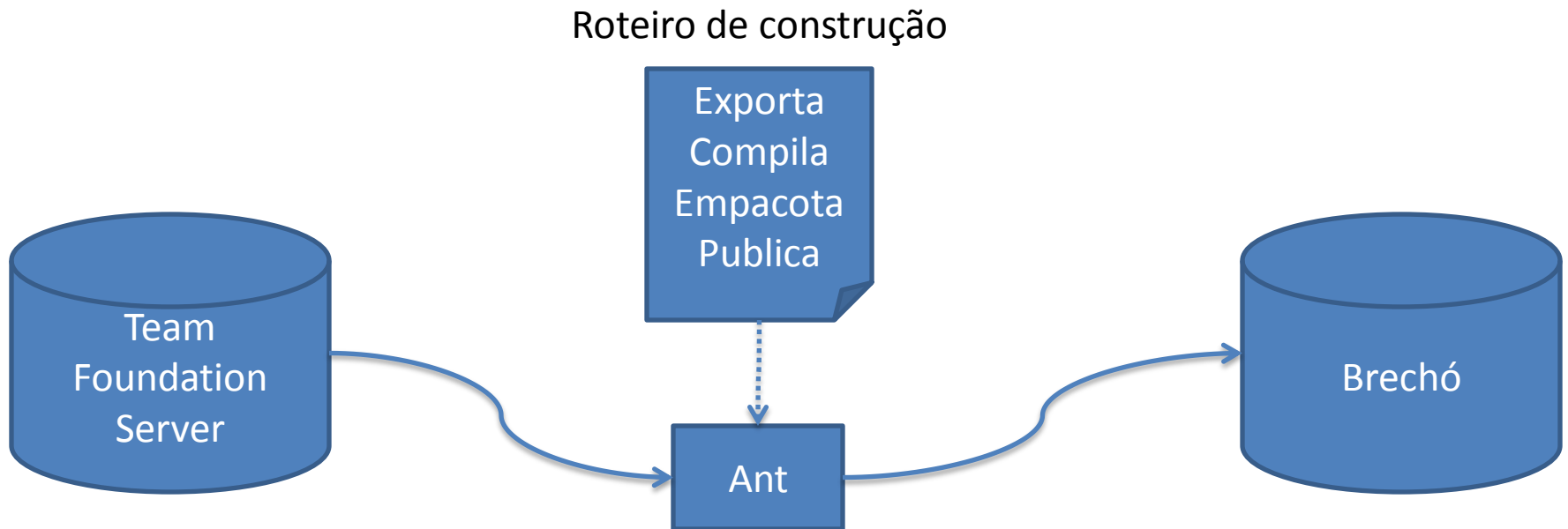
Integração dos repositórios



Exemplo

(Publicação via tarefas Ant)

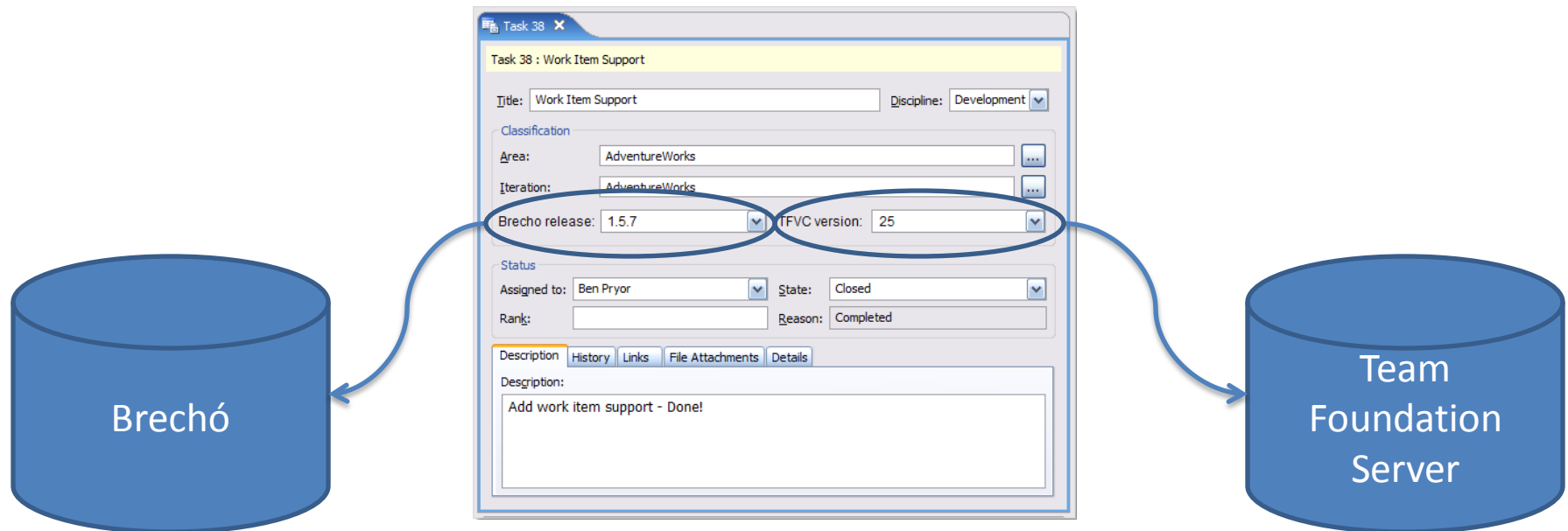
- Projeto Brechó-Foundation
 - Publicação no repositório Brechó releases feitas no Team Foundation Server



Exemplo

(Requisição de modificações)

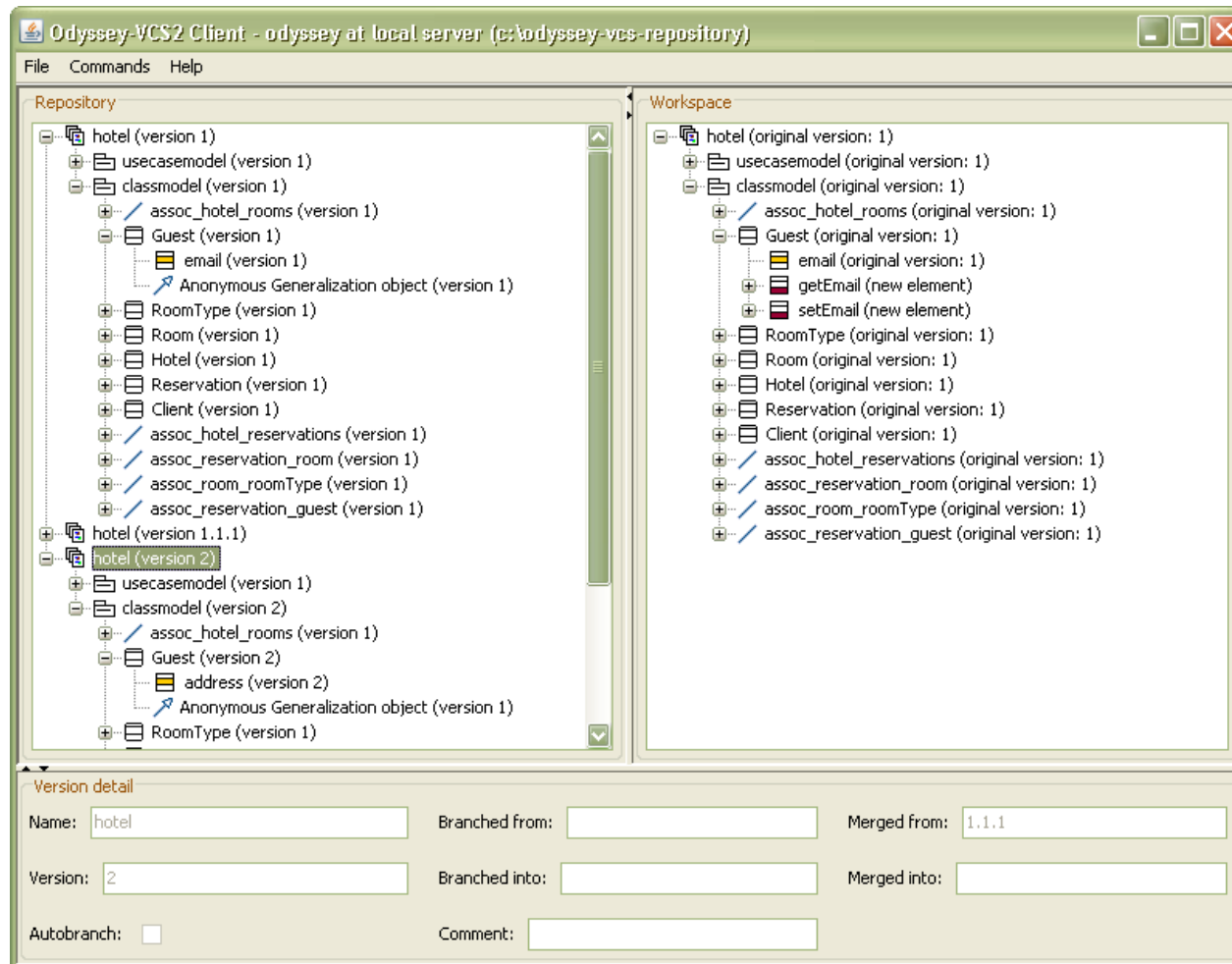
- Projeto Brechó-Foundation
 - Vinculação entre versões de produção (Brechó) e desenvolvimento (Team Foundation Server)



Agenda

- Introdução
- Repositório na perspectiva de Gerência de Configuração
- Repositório na perspectiva de Reutilização
- Integração dos repositórios
- **Conclusão**

Algumas pesquisas em andamento... (controle de versões de modelos)



Algumas pesquisas em andamento... (visualização do histórico)

The screenshot displays the Eclipse IDE interface for the 'EvolTrack' project. The main editor shows a UML Class Diagram for 'evoltrack.umlclass_diagram'. The diagram includes a Package 'm3' containing a 'pkg' folder, a Class 'Classe1' with attributes 'numero', 'operations', and 'classes', and other elements. The Project List on the left shows a tree structure under 'Example' with elements m1 through m6. The Control Panel at the bottom includes a Timeline and an Info section with configuration details.

Project List:

- Example
 - m1
 - m2
 - m3
 - m4
 - m5
 - m6

UML Class Diagram:

- Package: m3
 - pkg
 - packages
 - classifiers
 - other
 - Classe1
 - attributes: numero
 - operations
 - classes

Control Panel:

- Timeline: << > O >>
- Info:
 - Configuration's Author: Rafael Cepeda
 - Date: 2008-05-09 03:13:13
 - Version: 2

Algumas pesquisas em andamento... (visualização do histórico)

The screenshot shows the Eclipse IDE with the EvoTrack plugin. The main editor displays a UML class diagram for 'evoltrack.umlclass_diagram'. The diagram includes a package 'm4' containing a 'pkg' folder and two classes, 'Classe1' and 'Classe2'. The 'Classe1' class has attributes 'numero', operations 'operations', and 'classes'. The 'Classe2' class has attributes 'attributes', operations 'operations', and 'classes'. The left sidebar shows a project list with an 'Example' folder containing models 'm1' through 'm6'. The bottom control panel includes a timeline and an info section with configuration details.

Control Panel

Timeline

Info

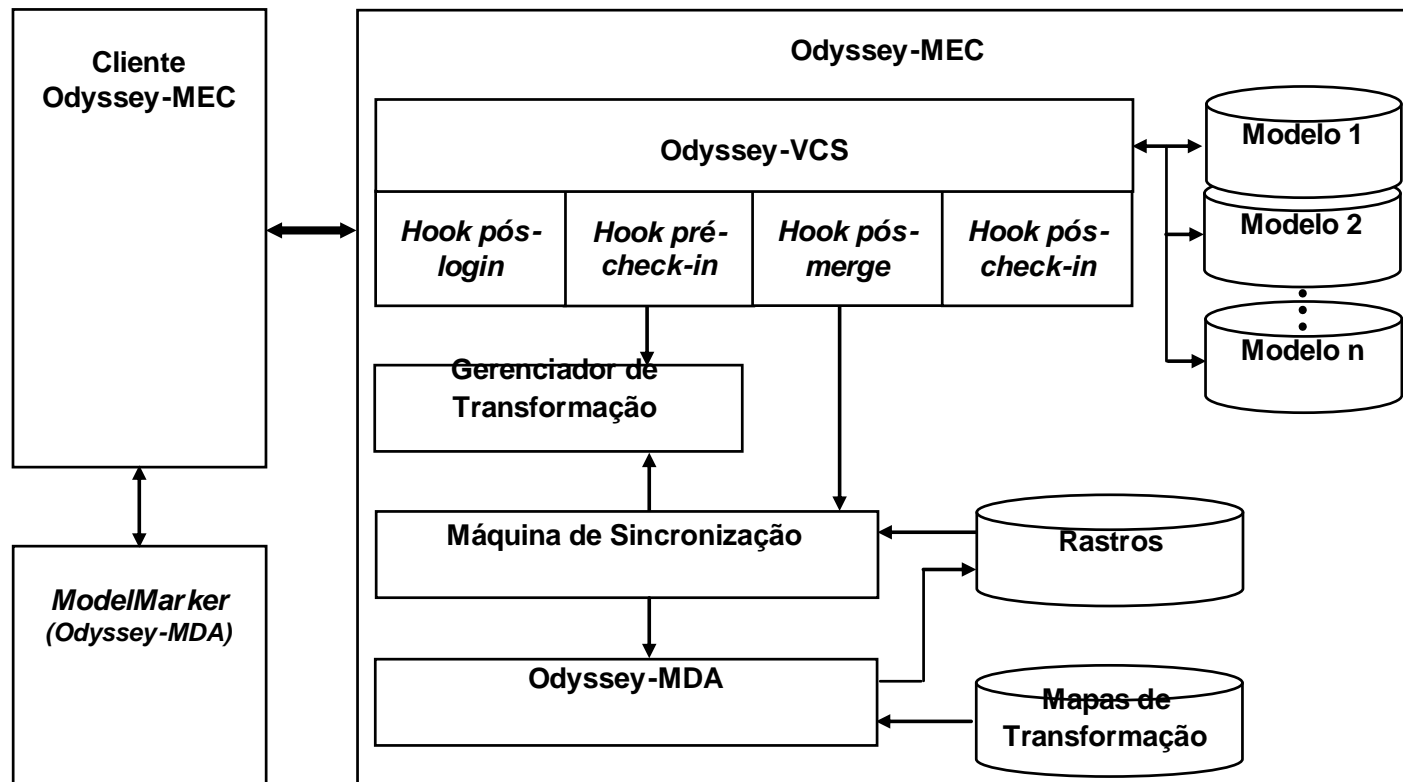
Configuration's Author: Rafael Cepeda
Date: 2008-05-09 03:13:23
Version: 3

45M of 120M

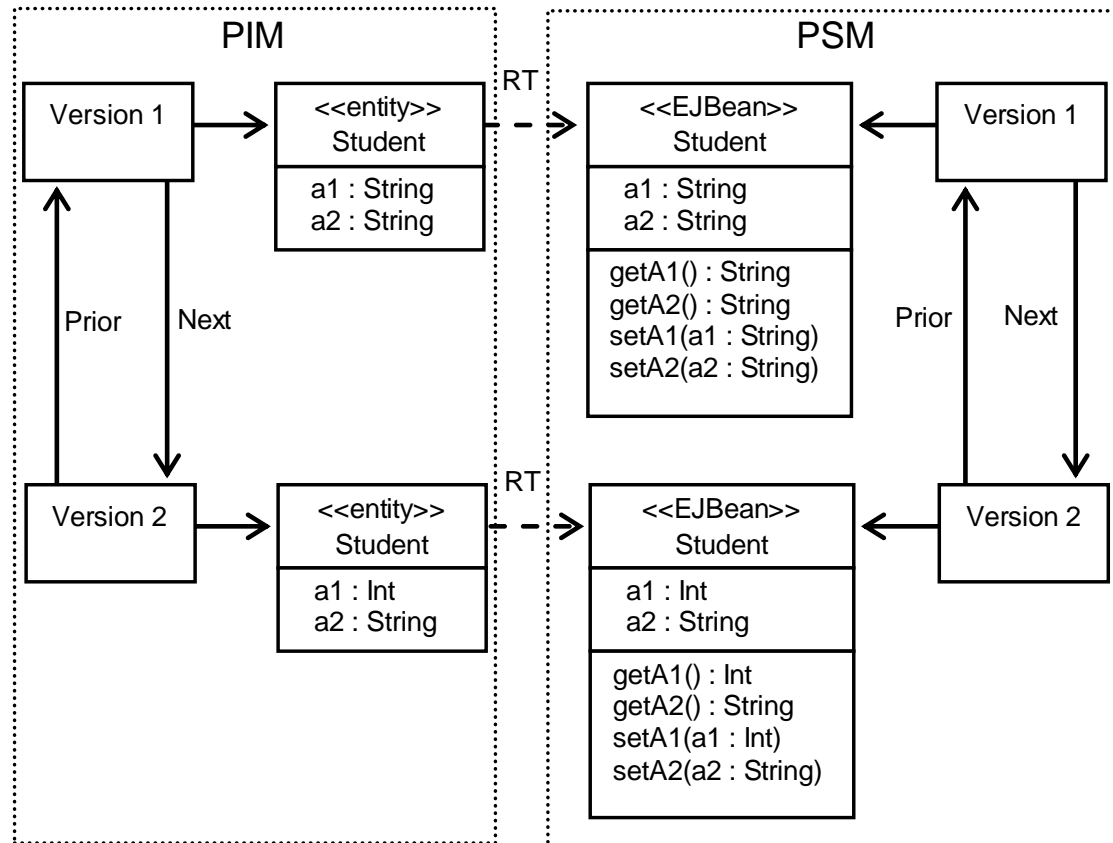
Algumas pesquisas em andamento... (visualização do histórico)

The screenshot displays the Eclipse IDE interface for editing a UML Class Diagram. The main workspace shows a package named 'm6' containing a sub-package 'pkg' and two classes, 'Classe1' and 'Classe2'. 'Classe1' is associated with 'Classe2' via a vertical line. The diagram elements are organized into compartments: 'Classe1' has 'attributes' (containing 'numero'), 'operations', and 'classes'. 'Classe2' has 'attributes' (containing 'attr'), 'operations', and 'classes'. The interface includes a Project List on the left, a Palette on the right, and a Control Panel at the bottom with a timeline and configuration info.

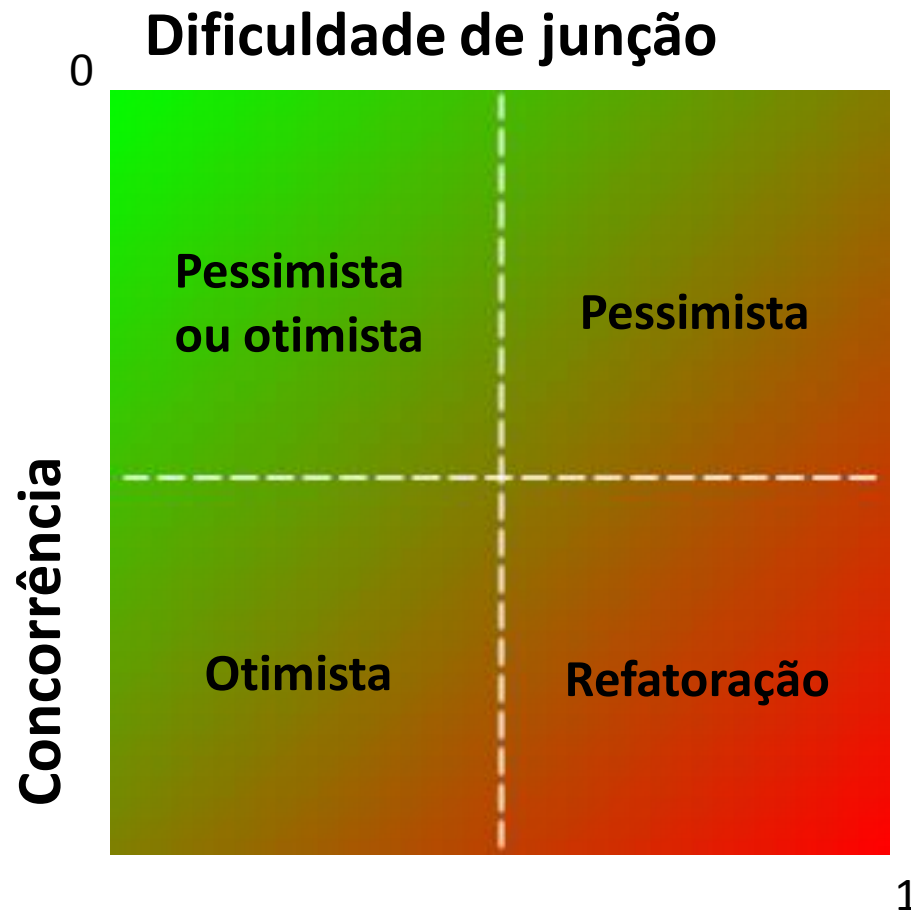
Algumas pesquisas em andamento... (evolução de modelos no MDD)



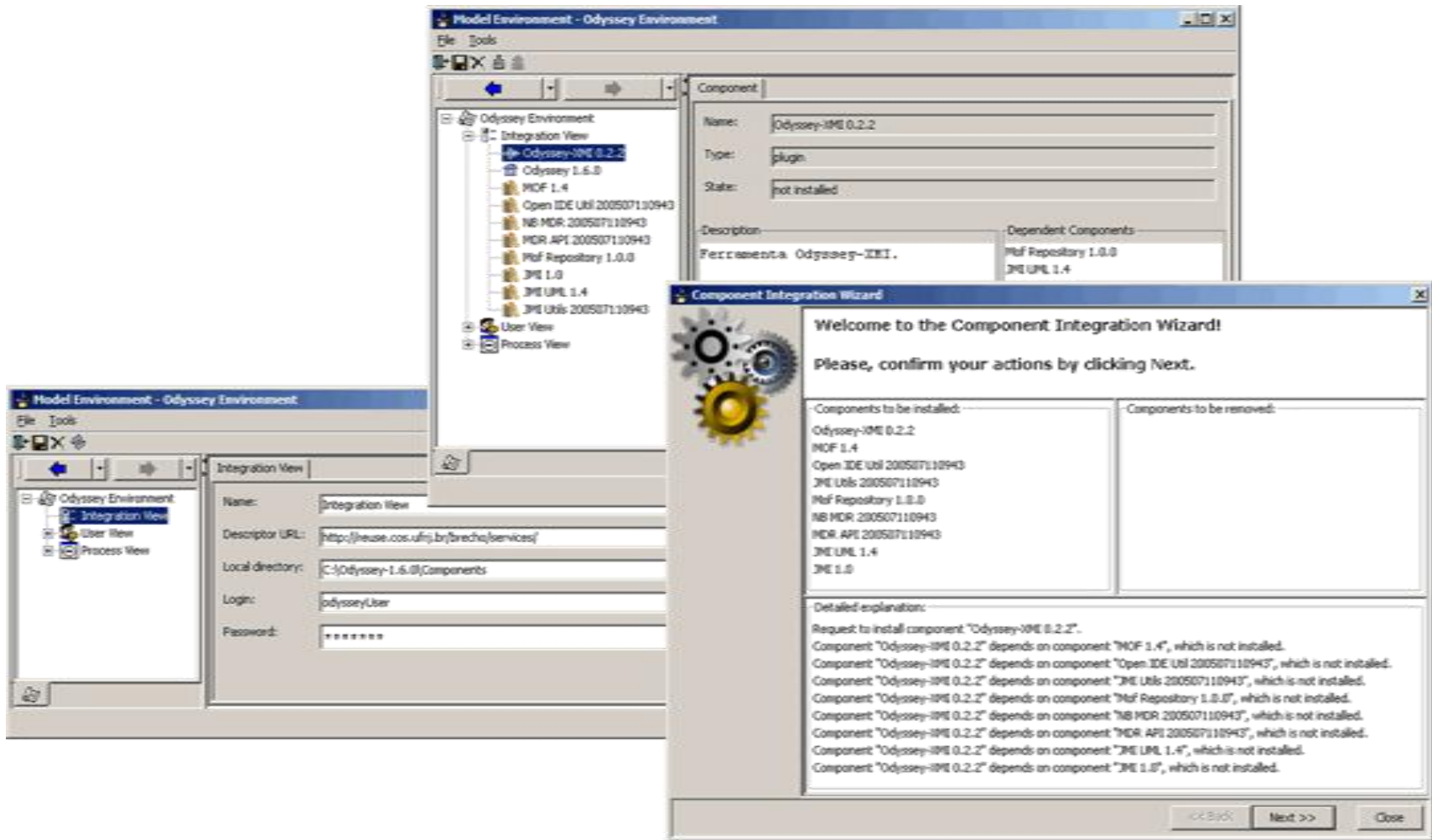
Algumas pesquisas em andamento... (evolução de modelos no MDD)



Algumas pesquisas em andamento... (sugestão de políticas de concorrência)



Algumas pesquisas em andamento... (reuso em tempo de execução)



Conclusão

- Repositórios de Gerência de Configuração e Reutilização são bastante diferentes
 - Propósitos diferentes
 - Funcionalidades diferentes
 - Participantes diferentes
- Contudo, a combinação de ambos em um ambiente integrado caracteriza um cenário ideal para controle dos artefatos!

Principais Referências Bibliográficas

- Alexis Leon, “A Guide to Software Configuration Management”, Artech House Publishers, 2000.
- Anne Hass, “Configuration Management Principles and Practices”, Boston, MA, Pearson Education, Inc.
- Sametinger, J., 1997, “Software Engineering with Reusable Components”, Springer-Verlag New York, Inc.
- Krueger, C.W., 1992, "Software Reuse", ACM Computing Surveys, v. 24, n. 2 (June), pp. 131-183.

Principais Referências Bibliográficas

- Mili, A., R. Mili, R. T. Mittermeir. 1998. “A survey of software reuse libraries”. *Ann. Softw. Eng.* 5: 349-414.
- Frakes, W. B., P. B. Gandel. 1989. “Representation methods for software reuse”. *Proceedings of the conference on Tri-Ada '89: Ada technology in context: application, development, and deployment*, 302-314. Pittsburgh, Pennsylvania, United States.
- MPS.BR – Melhoria de Processo do Software Brasileiro, Guia Geral (versão 1.2) e Guia de Implementação (versão 1.1)
- Pressman, R. S. (1997). “Software Engineering: A Practitioner's Approach”, McGraw-Hill.
- Biblioteca Brechó: <http://reuse.cos.ufrj.br/brecho>



Repositórios de Componentes nas perspectivas de Gerência de Configuração de Software e Reutilização de Software

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br