INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

Análise e Desenvolvimento de Sistemas

Instituto Federal da Bahia

Análise e Desenvolvimento de Sistemas

INF022 – Tópicos Avançados

Gerencia de Configuração

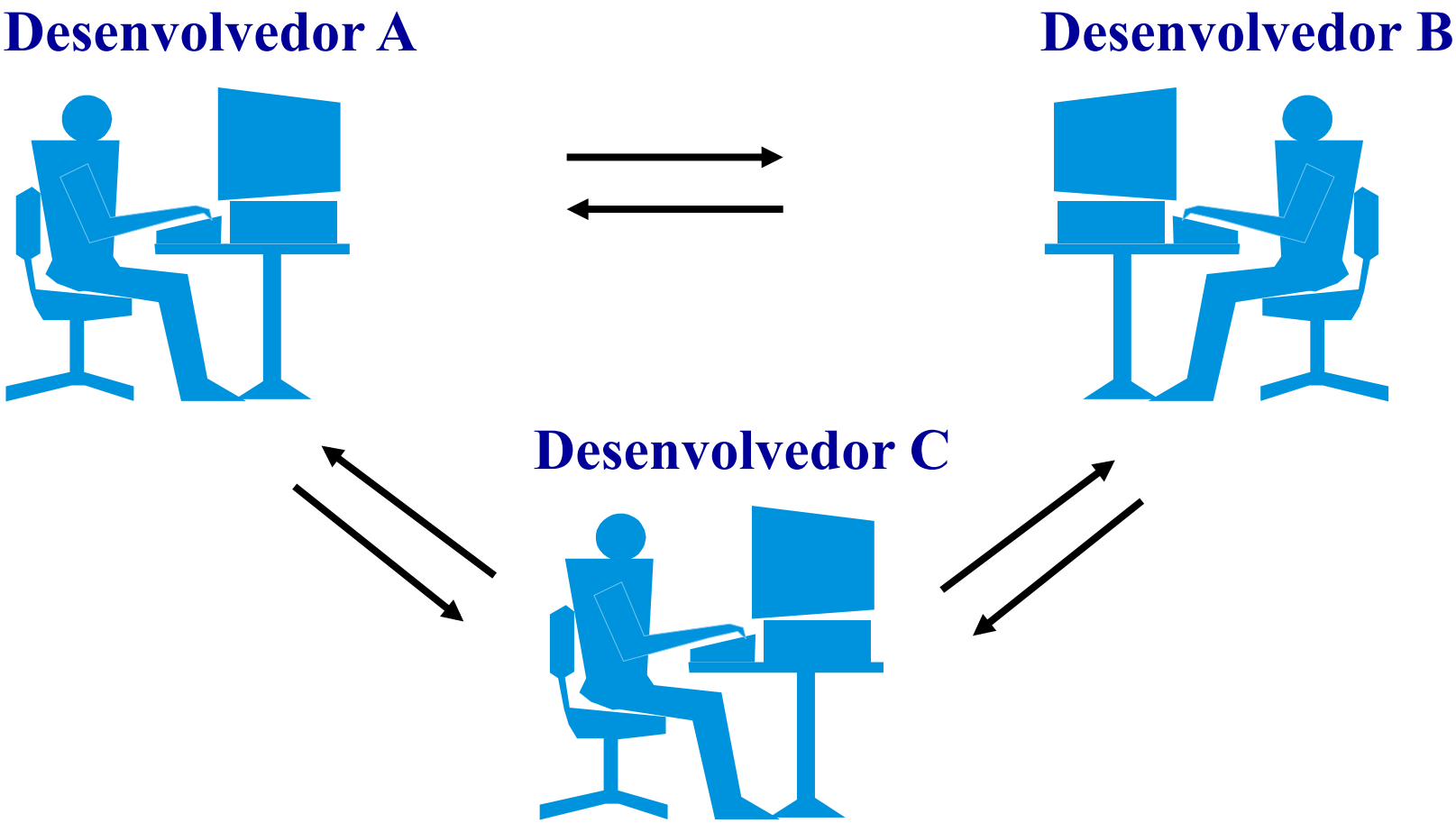
Prof. Dr. Renato L. Novais
renato@ifba.edu.br

- Objetivo
  - Compreender a importância do uso de mecanismos de gerência de configuração e de mudança, seus métodos, processos e ferramentas.
  - Fornecer os principais conceitos relacionados a GC.
  - Criar uma visão geral de como GC pode ser aplicada a um projeto de software.

Análise e Desenvolvimento de Sistemas

Contexto para Gerência de Configuração

Problema da Quebra de Comunicação



Problema da Quebra de Comunicação (continuação)

- Falhas de comunicação em equipes
- Ocorre pelas mais diversas razões:
  - Vocabulários incompatíveis
  - Culturas de desenvolvimento diferentes
  - Distância geográfica
  - Dificuldade de expressão
- Quando este problema acontece:
  - Os sistemas produzidos não atendem aos requisitos
  - Força de trabalho é desperdiçada

Problema dos Dados Compartilhados

Desenvolvedor A

Desenvolvedor B

Programa de A

A1 A2 A3

Componente Compartilhado

Programa de B

B1 B2 B3

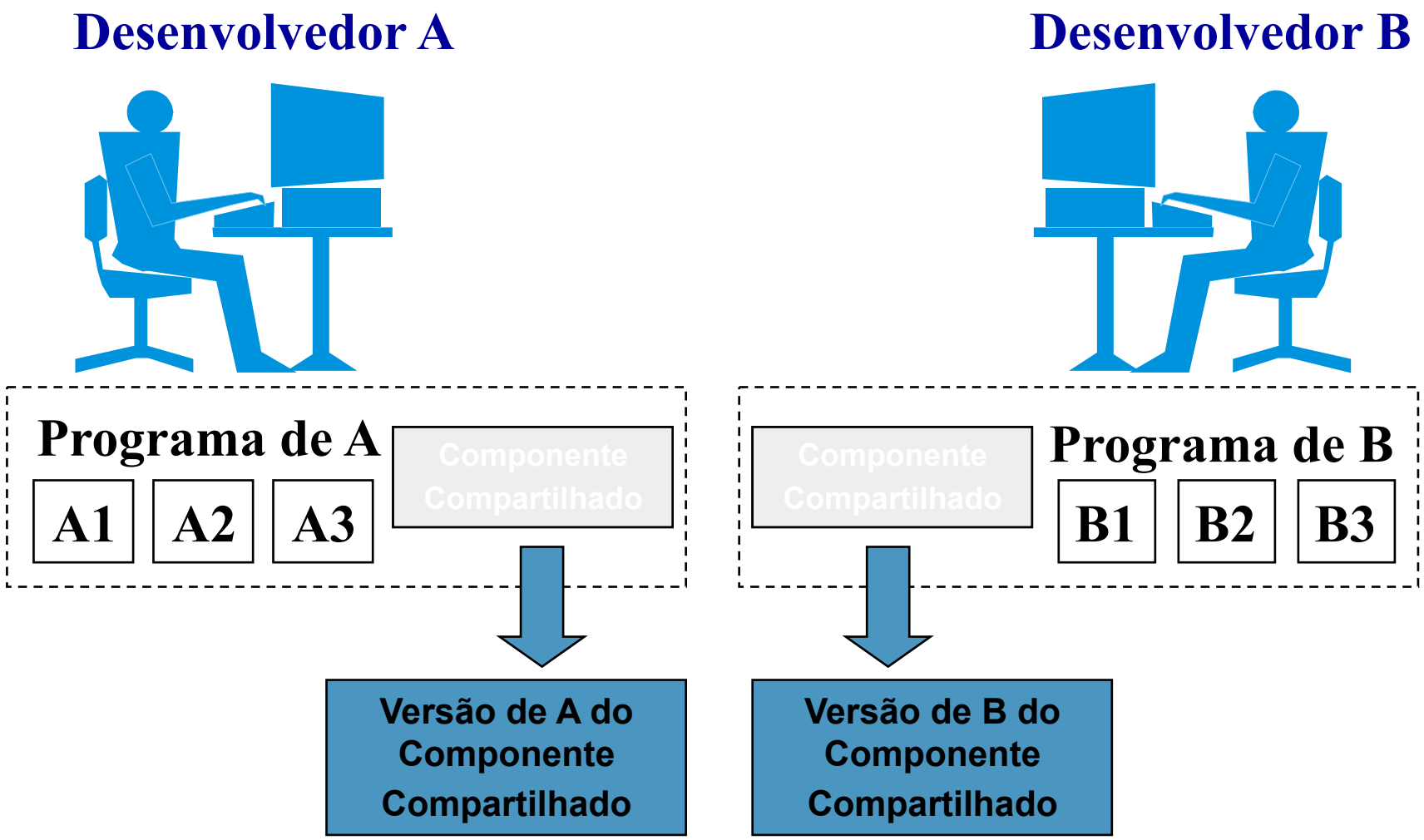
Problema dos Dados Compartilhados - Cenário

- O desenvolvedor A modifica o componente compartilhado
- Mais tarde, o desenvolvedor B realiza algumas alterações no mesmo
- Ao tentar compilar o componente, erros são apontados pelo compilador, mas nenhum deles ocorre na parte que B alterou
- O desenvolvedor B não tem a menor ideia sobre a causa do problema

Problema dos Dados Compartilhados - Solução simplista

- Solução simplista:
  - cada desenvolvedor trabalha em uma cópia "local" do componente
  - resolve o Problema dos Dados Compartilhados, mas cria um novo problema

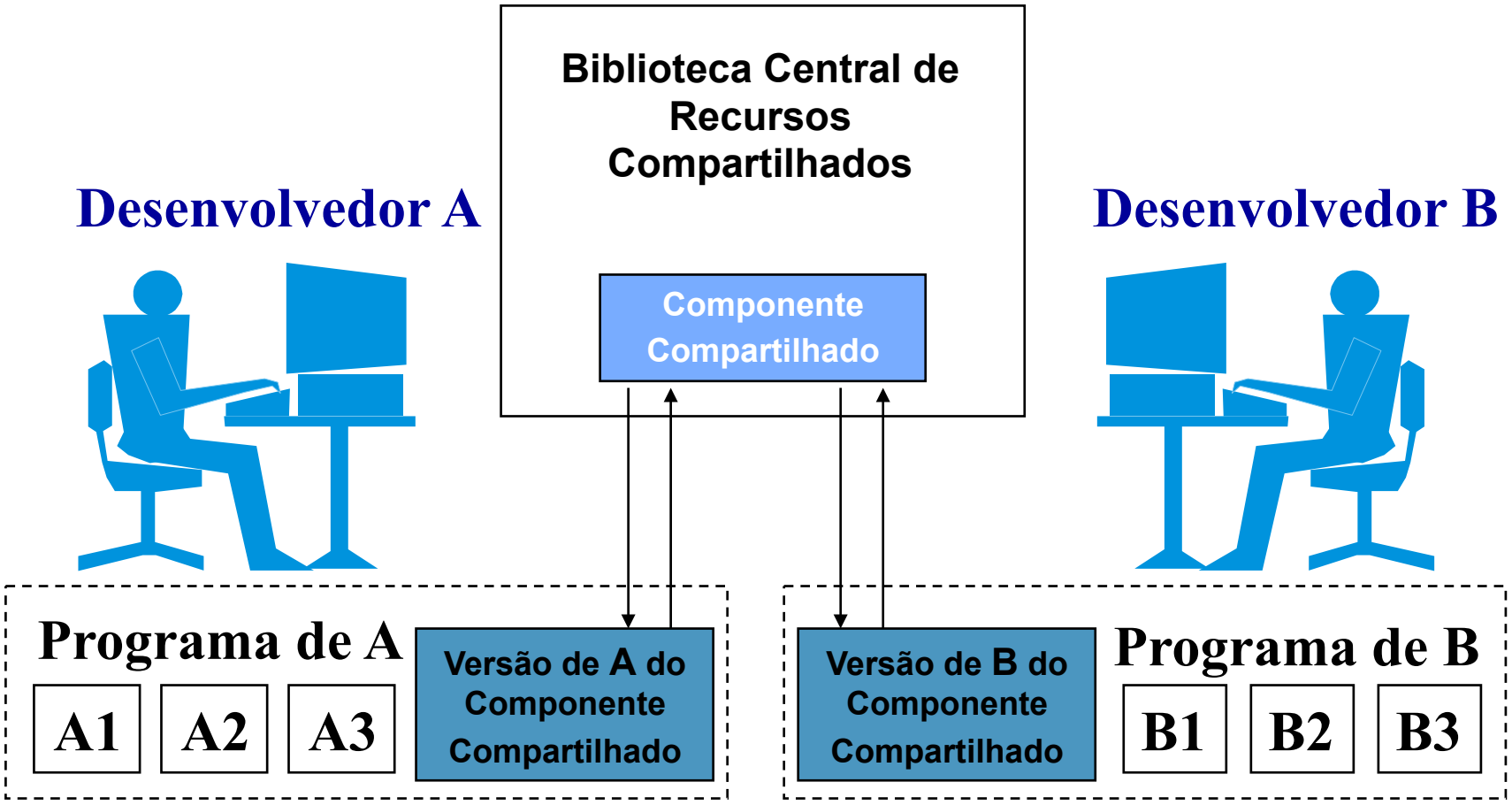
Problema da Manutenção Múltipla



Problema da Manutenção Múltipla (continuação)

- Ocorre quando cada desenvolvedor trabalha com uma cópia “local” do que seria o mesmo componente
- Dificuldade para saber:
  - Que funcionalidades foram implementadas em quais versões do componente
  - Que defeitos foram corrigidos
- Evitado através de uma biblioteca central de componentes compartilhados
  - Nesse esquema, cada componente é copiado para a biblioteca sempre que alterado
  - Resolve o Problema da Manutenção Múltipla, mas...

Problema da Atualização Simultânea



Problema da Atualização Simultânea – Cenário 1

- O desenvolvedor A encontra e corrige um defeito em sua versão do componente compartilhado
- Uma vez corrigido, o componente modificado é copiado para a biblioteca central
- O desenvolvedor B encontra e corrige o mesmo defeito em sua versão do componente por não saber que A já tinha feito isso
- O trabalho de A é desperdiçado

Problema da Atualização Simultânea – Cenário 2

- O desenvolvedor A encontra e corrige um defeito em sua versão do componente compartilhado
- Uma vez corrigido, o componente modificado é copiado para a biblioteca central
- O desenvolvedor B encontra e corrige um outro defeito em sua versão do componente, sem saber do defeito corrigido por A
- O desenvolvedor B copia sua versão do componente para a biblioteca central
- Além de o trabalho de A ser desperdiçado, a versão do componente que se encontra na biblioteca central continua apresentando um defeito
- O desenvolvedor A julga o problema como resolvido

Como Resolver?

- O problema da atualização simultânea não pode ser resolvido simplesmente copiando componentes compartilhados para uma biblioteca central
- Algum mecanismo de controle é necessário para gerenciar a entrada e saída dos componentes

O que é Gerência de Configuração?

- Gerência de configuração (GC) é o processo de identificar, organizar e controlar modificações ao software sendo construído
- A idéia é maximizar a produtividade minimizando os enganos

Objetivos de GC

- Definir o ambiente de desenvolvimento
- Definir políticas para controle de versões, garantindo a consistência dos artefatos produzidos
- Definir procedimentos para solicitações de mudanças
- Administrar o ambiente e auditar mudanças
- Facilitar a integração das partes do sistema

Benefícios

- Aumento de produtividade no desenvolvimento
- Menores Custos de Manutenção
- Redução de defeitos
- Maior rapidez na identificação e correção de problemas

Análise e Desenvolvimento de Sistemas

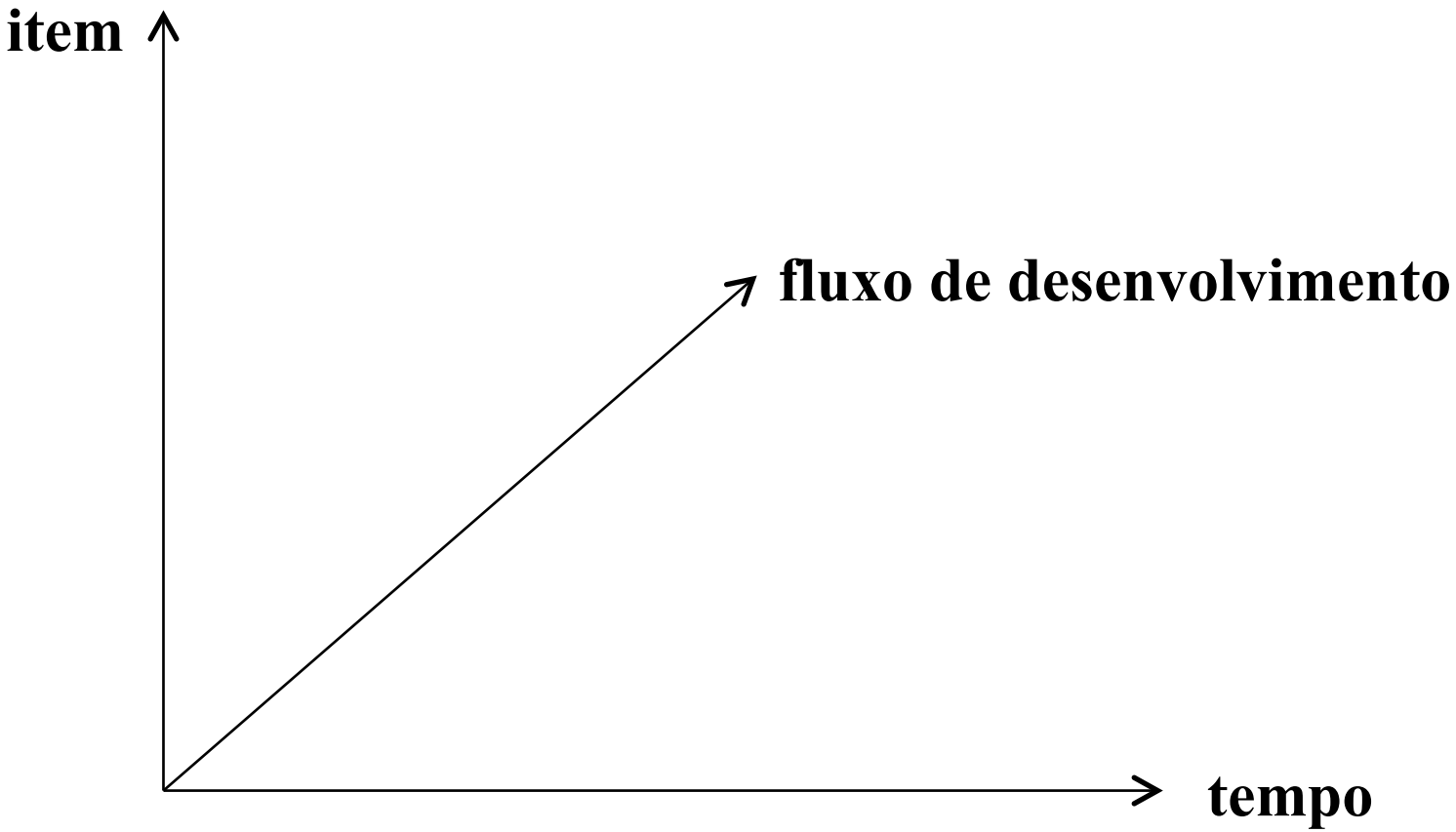
Conceitos Básicos

Configuração

- Um projeto de desenvolvimento de software produz os seguintes itens:
  - Programas (código fonte, programas executáveis, bibliotecas de componentes, etc.)
  - Documentação (manuais do usuário, documento de requisitos, modelo de análise e projeto, etc.)
  - Dados (dados de teste e do projeto)
- Esses conjuntos de itens são chamados, coletivamente, de configuração do software

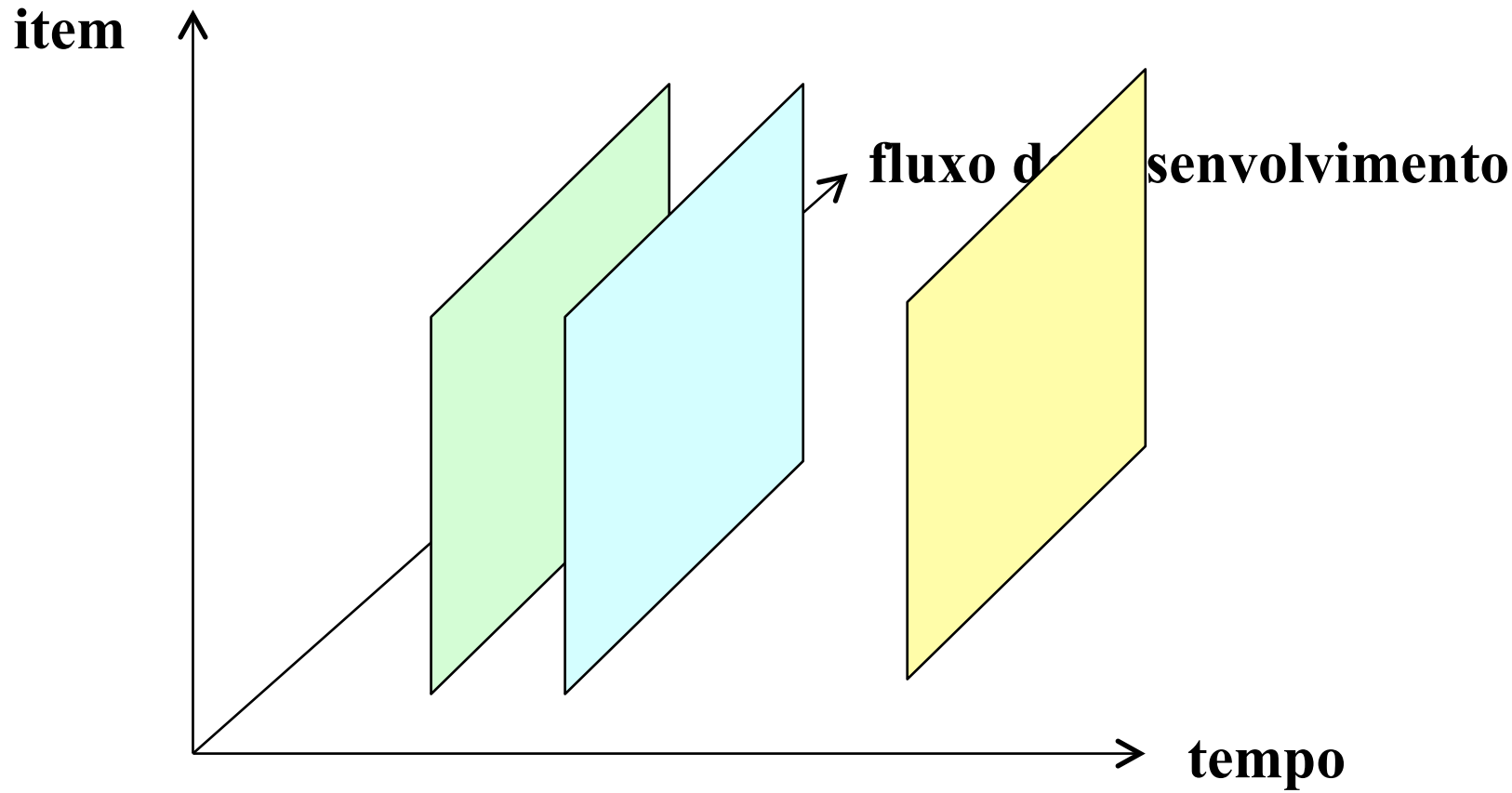
- Um conjunto de itens de hardware e/ou software são vistos como uma entidade única para fins de gerênc configuração

- Um item de configuração está sujeito a mudanças e essas devem obedecer às políticas estabelecidas
- Normalmente, um item de configuração é estabelecido para cada pedaço de software que pode ser projetado, implementado e testado de forma independente


item
fluxo de desenvolvimento
tempo

Baseline

- Uma especificação ou produto que foi formalmente revisado e aceito
  - Serve como base para os passos posteriores do desenvolvimento
- A configuração do software em um ponto discreto no tempo
- Só pode ser modificado através de procedimentos formais (solicitações de mudança)
- Um artefato ou conjunto de artefatos só se torna um item de configuração depois que um baseline é estabelecido

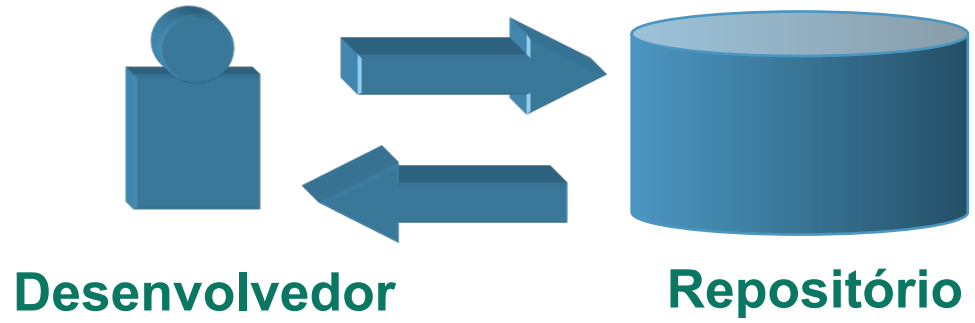

item
fluxo d senvolvimento
tempo

Baselines importantes

- Baselines são considerados marcos no processo de desenvolvimento:
  - Funcional : requisitos
  - De Produto : releases, iterações

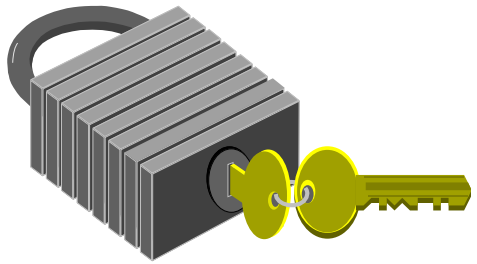
Repositório

- Local (físico e lógico) onde os itens de um sistema são guardados
- Pode conter diversas versões do sistema
- Utiliza mecanismos de controle de acesso

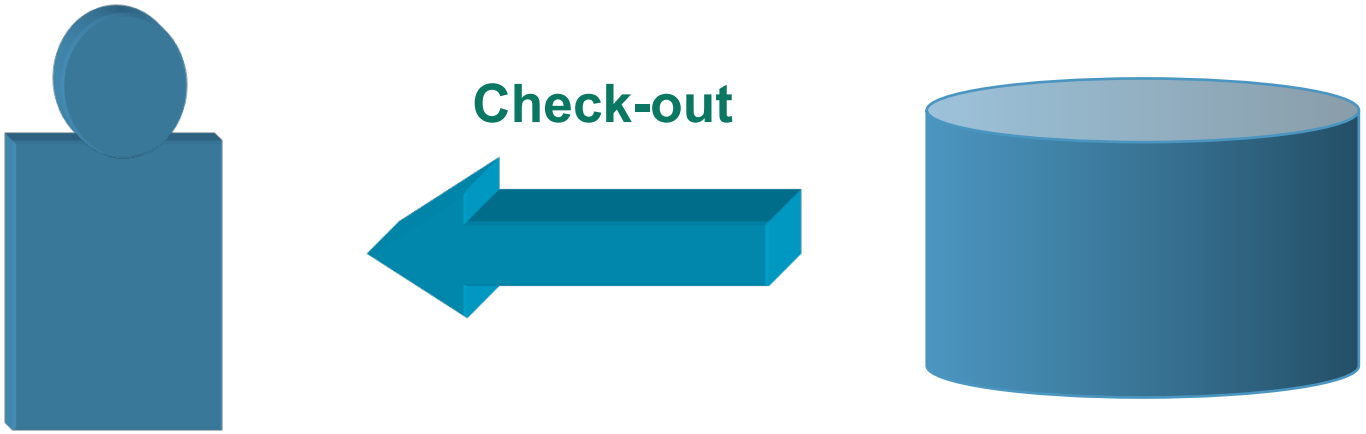


Lock

- Resolve a Atualização Simultânea
- Garante que apenas o usuário que detém o lock pode alterar o arquivo
- Problema: “serializa” o trabalho dos desenvolvedores



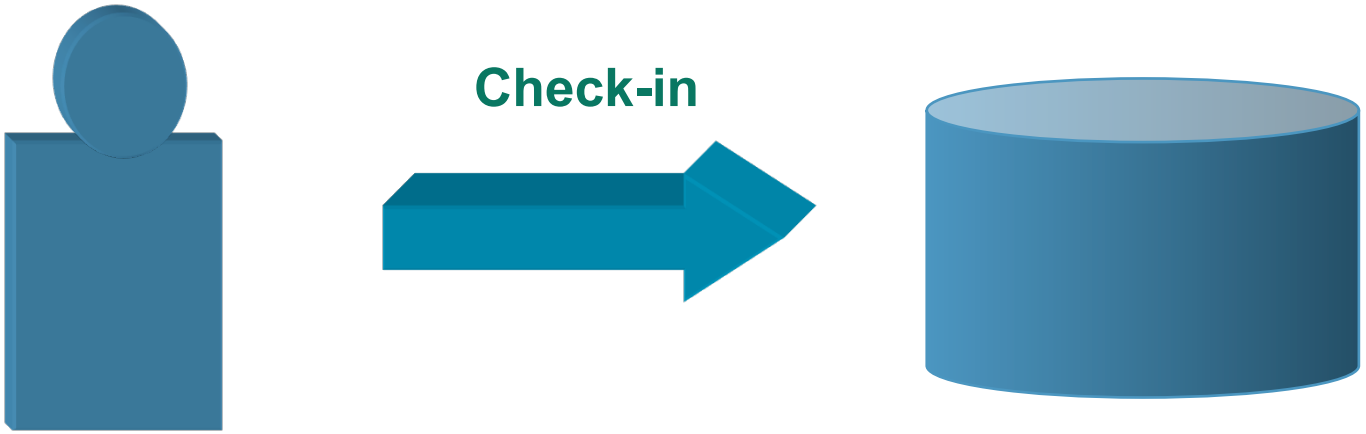
Check-Out



Check-Out (continuação)

- Recupera a (última) versão de um item de configuração guardada no repositório
  - Escrita
    - Verifica que ninguém detém o lock do item de configuração
    - Obtém o lock do item
    - Cria uma cópia, para edição, no cliente
  - Leitura
    - Verifica que alguém já detém o lock
    - Cria uma cópia, apenas para leitura, no cliente

Check-In



Check-In (continuação)

- Ação de inserir/atualizar um item de configuração no repositório
  - Verifica o lock do item de configuração, caso o mesmo já exista
  - Verifica e incrementa a versão do item
  - Registra informações das mudanças (autor, data, hora, comentários)
  - Inclui/atualiza o item

Build

- Representa uma versão ainda incompleta do sistema em desenvolvimento, mas com certa estabilidade
- Costuma apresentar limitações conhecidas
- Espaço para integração de funcionalidades
- Inclui não só código fonte, mas documentação, arquivos de configuração, base de dados, etc.
- A política de geração dos builds deve ser bem definida na estruturação do ambiente

Os Problemas na Geração de Builds

- Fazer os builds do sistema manualmente é muito demorado
- Pode ser difícil saber qual a versão "correta" de um arquivo
- Os pedaços do sistema podem estar em diversos locais diferentes
  - Alguns arquivos podem ser esquecidos

Os Problemas na Geração de Builds

- A integração das partes de um sistema em desenvolvimento normalmente é:
  - Realizada poucas vezes, apenas perto de sua implantação
  - Feita em frequência inversamente proporcional à complexidade do sistema
- Integrar as partes de um sistema é uma tarefa trabalhosa e sujeita a erros
  - Quanto maior o sistema, mais difícil

- Consequência: problemas de integração tornam-se difíceis de detectar cedo no desenvolvimento
  - Costumam ser encontrados muito depois de sua introdução
  - É muito difícil rastrear suas causas

- Geração frequente (pelo menos diária) de builds do sistema
  - As partes do sistema são integradas constantemente
  - Problemas de integração passam a ser encontrados logo que introduzidos, na maioria dos casos
- Considerada uma das “melhores práticas” no desenvolvimento de software
- A geração de builds deve ser automatizada e realizada com frequência adequada

- Identificação e empacotamento de artefatos entregues ao cliente (interno ou externo) ou ao mercado
- Um release implica no estabelecimento de um novo baseline, de produto
- Produto de software supostamente sem erros
  - Versão do sistema validada após os diversos tipos de teste
  - Garantia de que todos os itens de configuração foram devidamente testados, avalidos, aceitos e estão disponíveis no novo baseline
- Processo iterativo/incremental produz, em geral, mais de um release

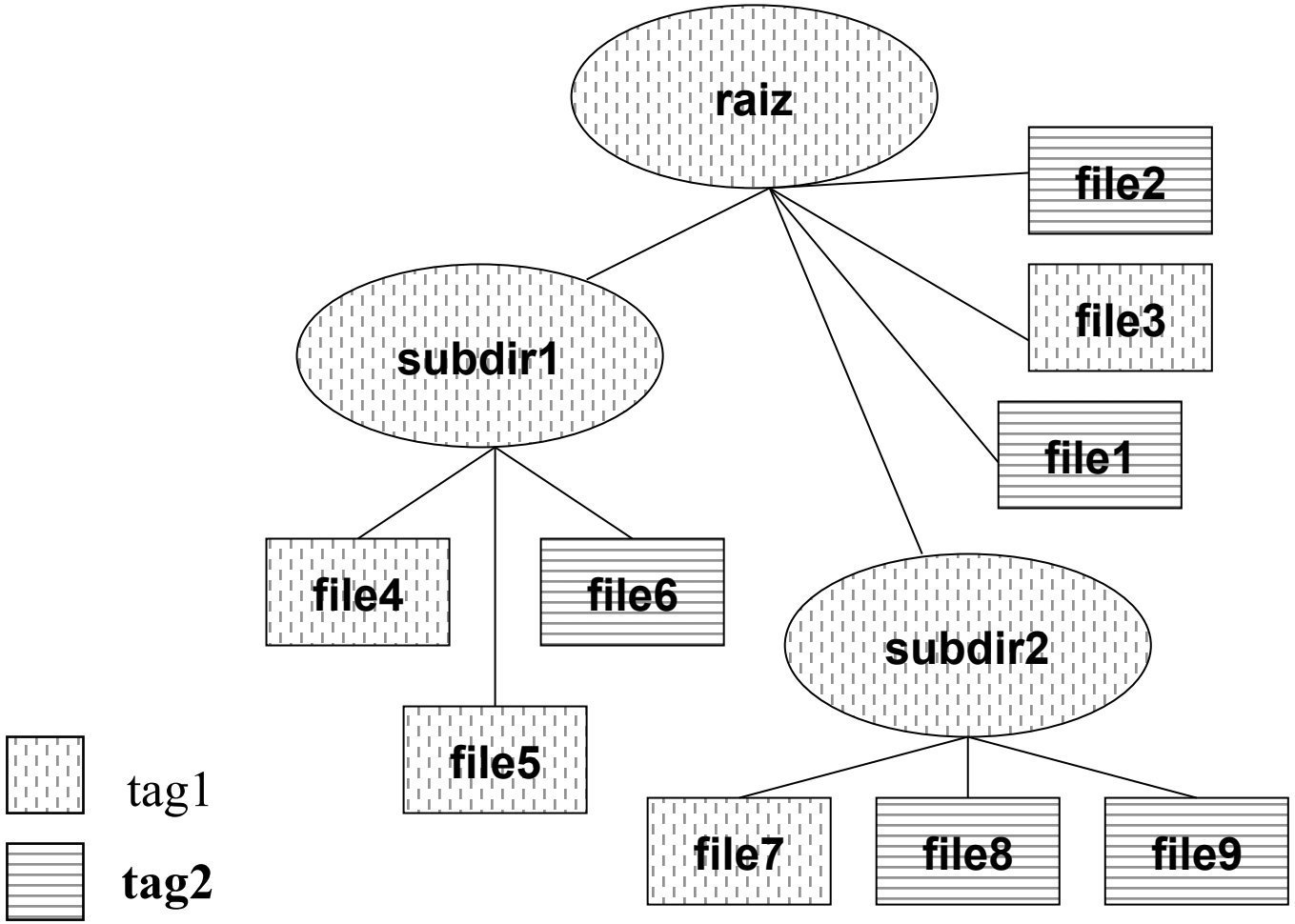
Tipos de release

- Normalmente, releases estão associados aos milestones do plano de projeto
- Internos
  - Controle de qualidade, acompanhamento de projeto, controle de riscos, aceitação, aquisição de conhecimento através da coleta de feedbacks, desenho da estratégia de implantação
- Externos
  - Implantado e utilizado pelo cliente

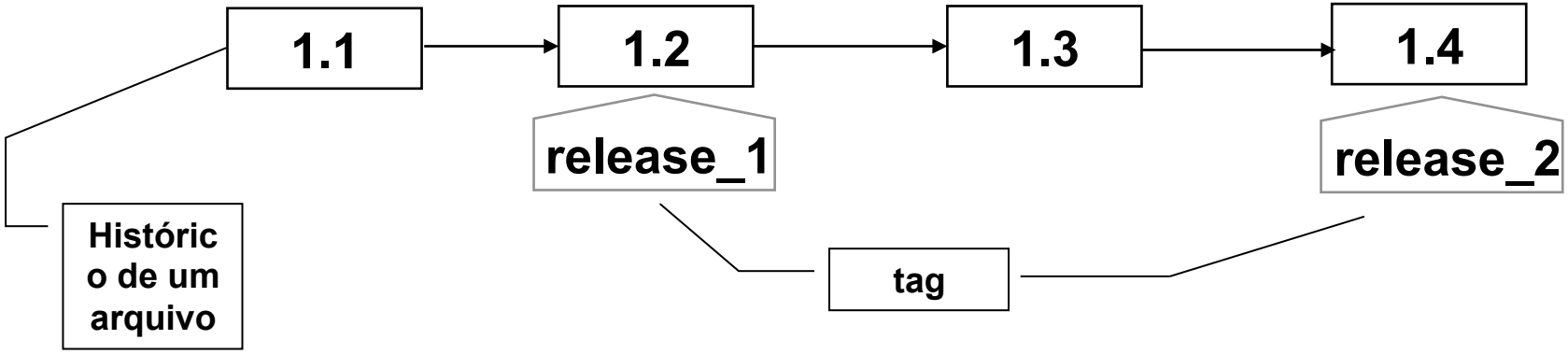
Tags

- Rótulos que são associados a conjuntos de arquivos
- Um tag referencia um ou mais arquivos em um ou mais diretórios
  - Costuma-se usar tags para:
    - Denominar projeto rotulando todos os arquivos associados ao projeto
    - Denominar uma versão do projeto (um build ou release) rotulando todos os arquivos associados ao build ou release

Tags – Cenário 1



Tags – Cenário 2



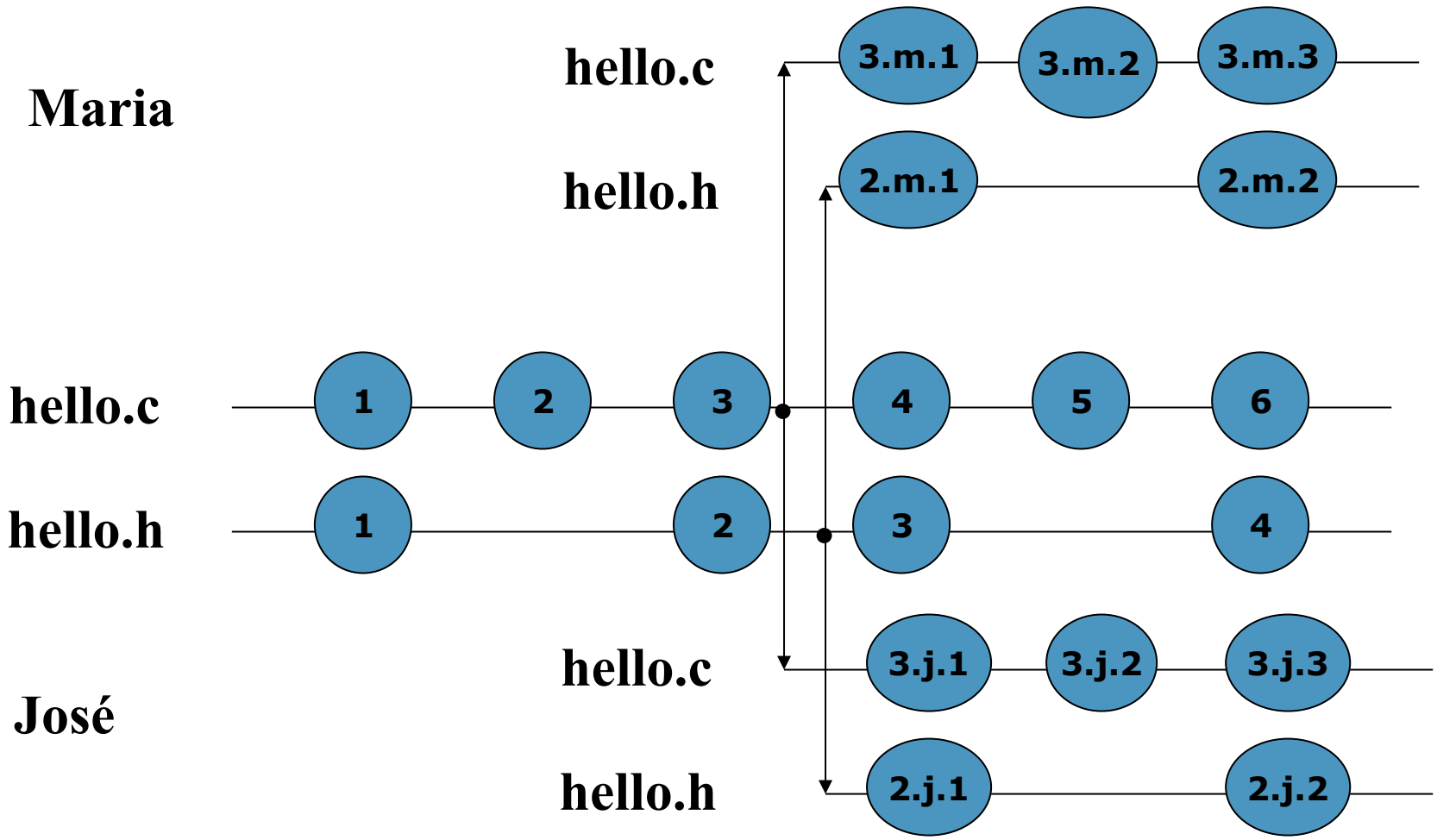
Branch

- Criação de um fluxo alternativo para atualização de versões de itens de configuração
- Recurso muito poderoso
- Devem existir regras bem definidas para criação de branches
  - Por que e quando devem ser criados?
  - Quais os passos?
  - Quando retornar ao fluxo principal?

Branch (continuação)

- Uso de lock inviabiliza a criação de branches
- Branches normalmente se originam de correções em versões anteriores

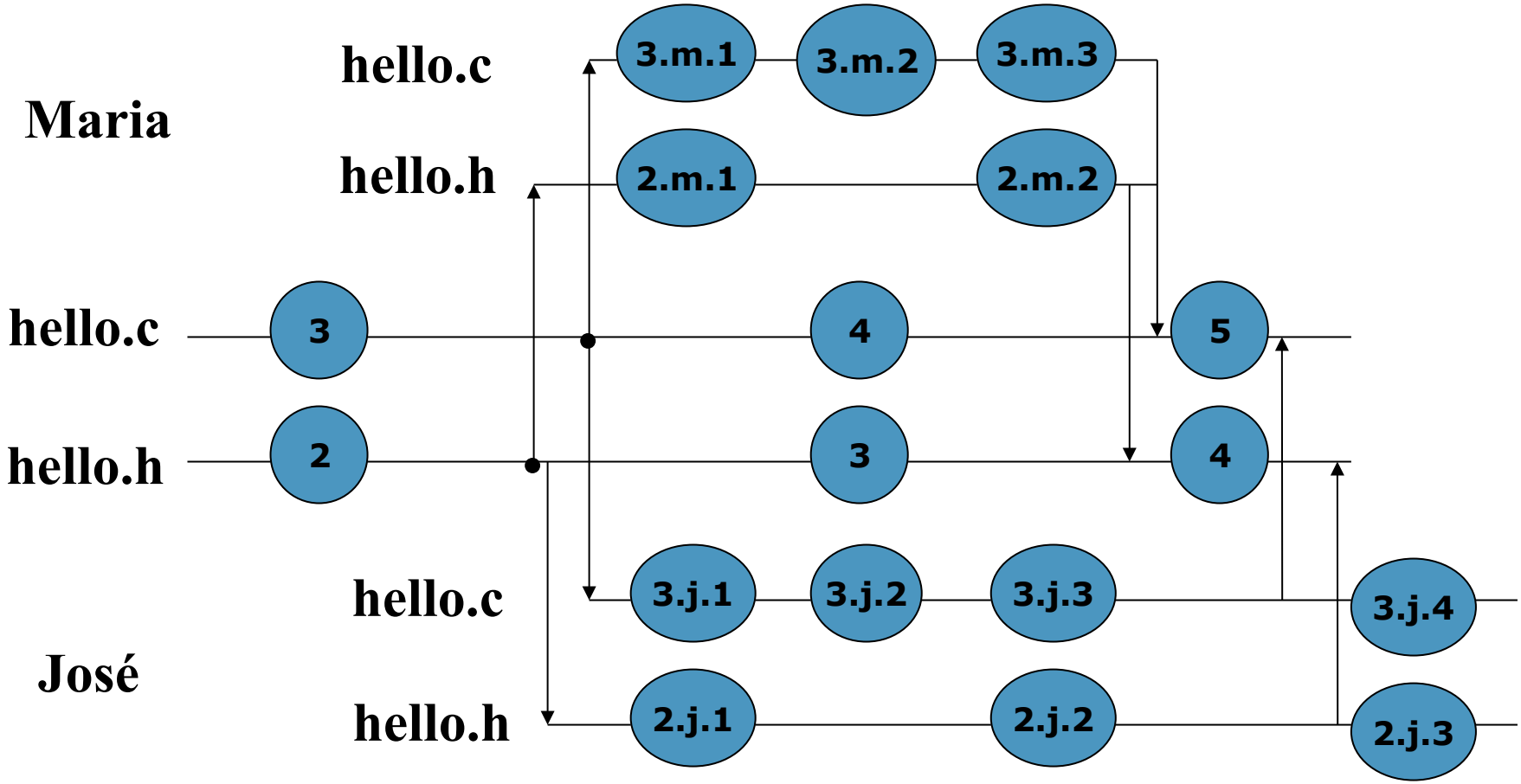
Branch (exemplo)



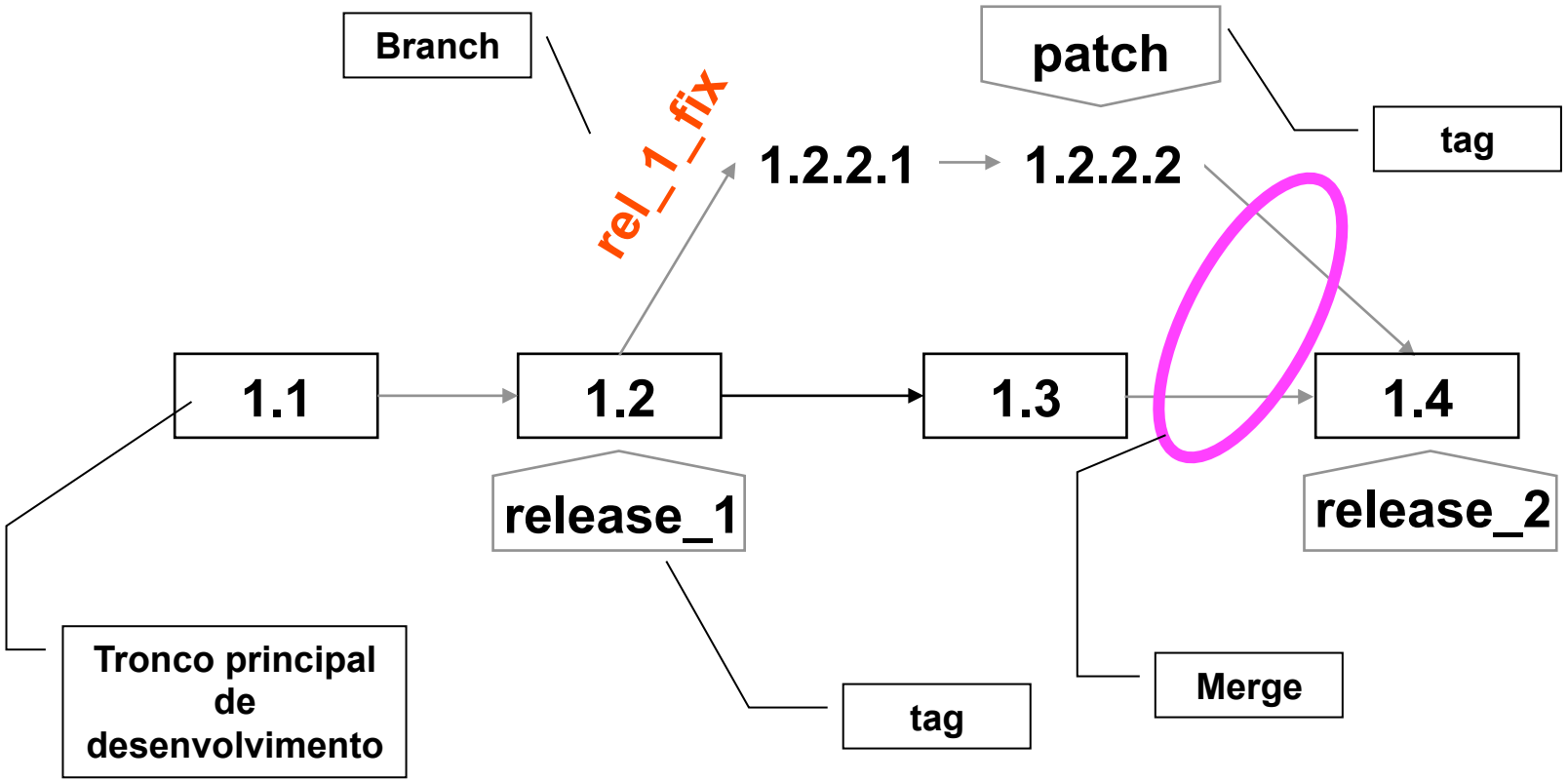
Merge

- Unificação de diferentes versões de um mesmo item de configuração
- Integração dos itens de configuração de um branch com os itens de configuração do fluxo principal
- Check-out atualizando a área local
- Algumas ferramentas fornecem um mecanismo automático para realização de merges
  - Mesmo com o uso de ferramentas, em vários casos há necessidade de intervenção humana

Merge (exemplo)



Branching e Merging: esquema gráfico



Oportunidades criadas com GC

- Reuso de itens de software
  - Artefatos
  - Componentes
- Automação de processo
  - Construção de builds
  - Geração de releases
  - Testes
  - Integração
- Aumento da produtividade das equipes
- Redução de re-trabalho
- Melhoria do acompanhamento do projeto

Ferramentas

- Determinadas CASE ou IDEs são preparadas para operar com ferramentas de gerência de configuração
  - Característica positiva!
- Algumas fornecem funções rudimentares de gerência de configuração integradas
  - Característica negativa!
- O ideal seria a utilização de uma ferramenta externa, própria para gerência de configuração
- As ferramentas de gerência de configuração atuais permitem de forma transparente desenvolvimento distribuído, paralelo e concorrente

Exemplo de ferramentas de controle de versões

- Livre
  - Aegis
  - Bazaar
  - CVS
  - Git
  - Mercurial
  - SubversioN
- Comercial
  - BitKeeper (BitMover)
  - ClearCase (IBM Rational)
  - Perforce
  - PVCS (Serena)
  - StarTeam (Borland)
  - Synergy/CM (Telelogic)
  - Visual SourceSafe (Microsoft)
  - Visual Studio Team Foundation (Microsoft)

Exemplo de ferramentas de controle de modificações

- Livre
  - Bugzilla
  - Mantis
  - Roundup
  - Scarab
  - Trac
  - Redmine
- Comercial
  - ClearQuest (IBM Rational)
  - JIRA (Atlassian)
  - StarTeam (Borland)
  - Synergy/Change (Telelogic)
  - TeamTrack(Serena)
  - Visual Studio Team Foundation (Microsoft)

Exemplo de ferramentas de controle de construção e liberação

- Livre
  - Ant
  - CruiseControl
  - Nant
  - Make
  - SCons
- Comercial
  - ClearMake (IBM Rational)
  - MSBuild (Microsoft)
  - Synergy/CM Object Make (Telelogic)

Perguntas

?

•5/28/15

Referências

- Material compilado de
  - http://subversion.assembla.com/svn/gco/livros/FluxoDeGerenciaDeConfiguracao.ppt
  - http://www.ic.uff.br/~leomurta/courses/2011.1/gc/aula6.pdf