

Contexto para Gerência de Configuração

Problema da Quebra de Comunicação

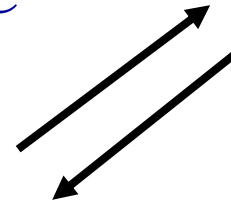
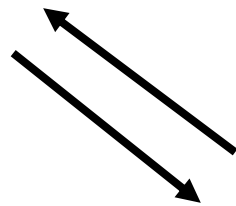
Desenvolvedor A



Desenvolvedor B



Desenvolvedor C



Problema da Quebra de Comunicação

- 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 idéia sobre a causa do problema

Problema dos Dados Compartilhados - Solução simplista

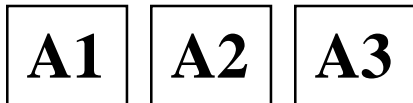
- Solução simplista:
 - cada desenvolvedor trabalha em uma cópia “local” do componente
 - resolve o Problema dos Dados Compartilhados, mas cria um novo problema

Problema da Manutenção Múltipla

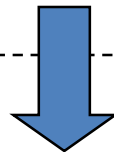
Desenvolvedor A



Programa de A



Componente
Compartilhado



Versão de A do
Componente
Compartilhado

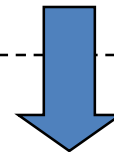
Desenvolvedor B



Programa de B



Componente
Compartilhado

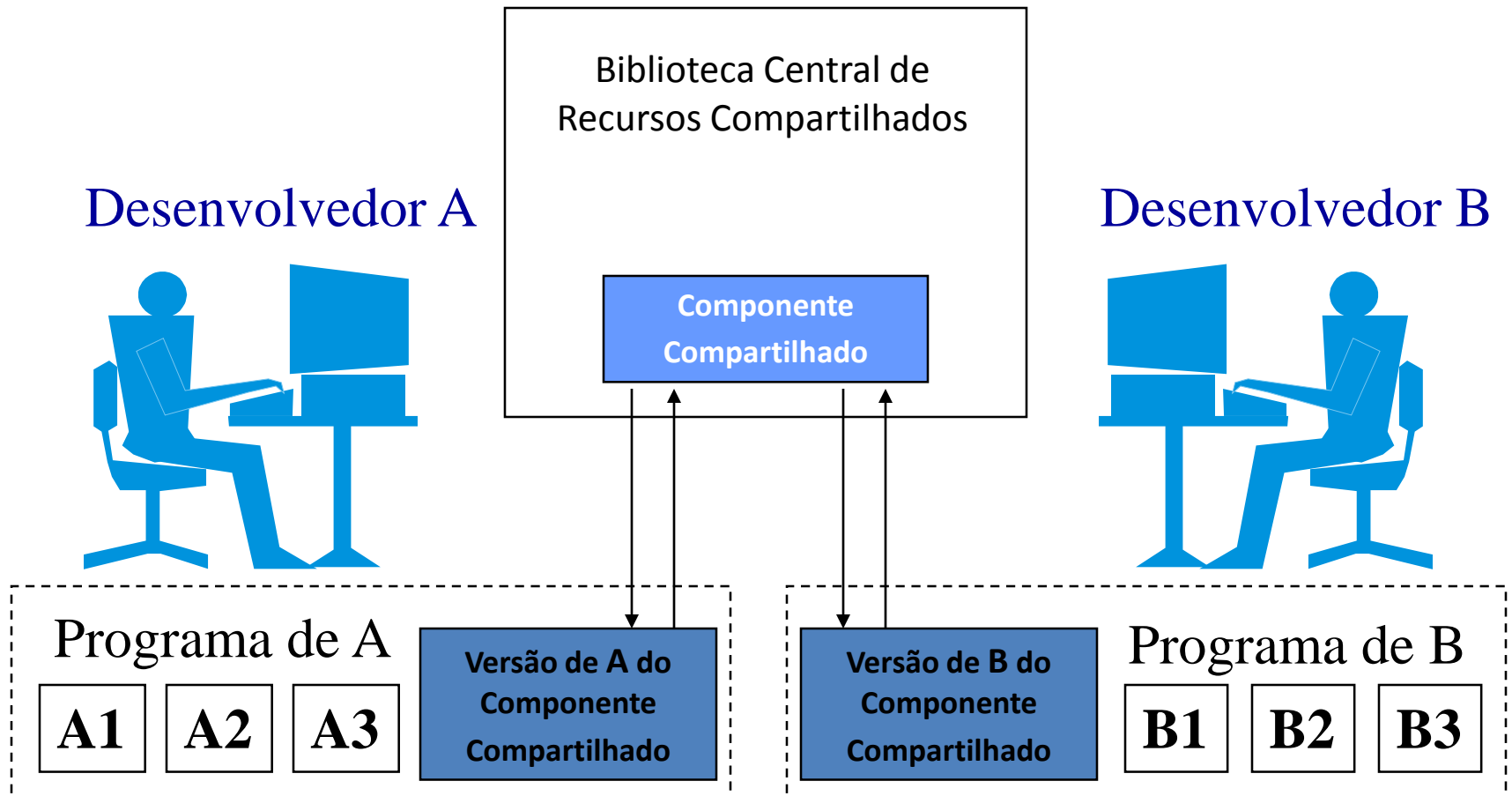


Versão de B do
Componente
Compartilhado

Problema da Manutenção Múltipla

- 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

Gerência de Configuração

Gerência de Configuração

- Desenvolvimento e o uso de padrões e procedimentos para o gerenciamento de sistemas de software em desenvolvimento
 - é o processo de identificar, organizar e controlar modificações ao software sendo construído
 - A idéia é maximizar a produtividade minimizando os enganos

Gerência de Configuração

- Fundamental para sistemas complexos
- Sem Gerência de Configuração o que pode dar errado?



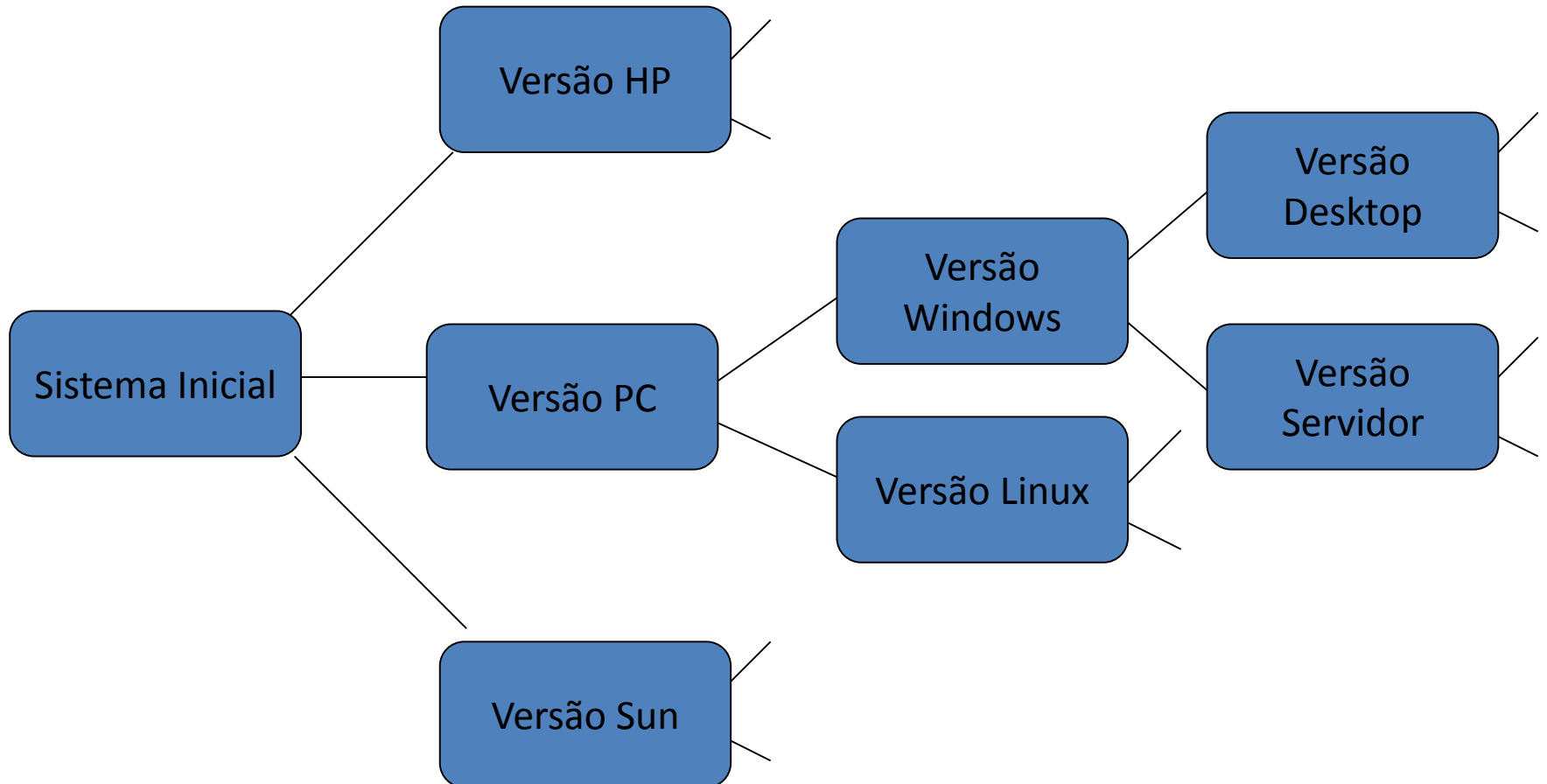
O que acontece Durante e após o Desenvolvimento de Software?

- Requisitos de sistemas sempre mudam durante o desenvolvimento e uso
 - Incorporar estes requisitos as novas versões do sistema
 - Gerenciar os sistemas em desenvolvimento
 - Fácil perder a rastreabilidade de quais mudanças foram incorporadas em qual versão
 - As versões incorporam propostas de mudanças, correções de defeitos e adaptações para diferentes hardware e sistemas operacionais para diferentes

Sem Gerência de Configuração o que pode dar errado?

- Desperdiçar esforços modificando a versão errada de um sistema
- Entregar a versão errada de um sistema para aos cliente
- Perder a rastreabilidade de onde o código-fonte do software está armazenado

Família de Sistemas



Procedimentos de Gerência de Configuração

- Definem como registrar e processar mudanças de sistema
- Como relacioná-las aos componentes de sistema e métodos usados para identificar diferentes versões dele

Ferramentas de Gerência de Configuração

- Usadas para armazenar versões de componentes do sistema, sistemas construídos com base nesses componentes
- Rastrear os releases das versões do sistema para os clientes

Conceitos Fundamentais

Configuração

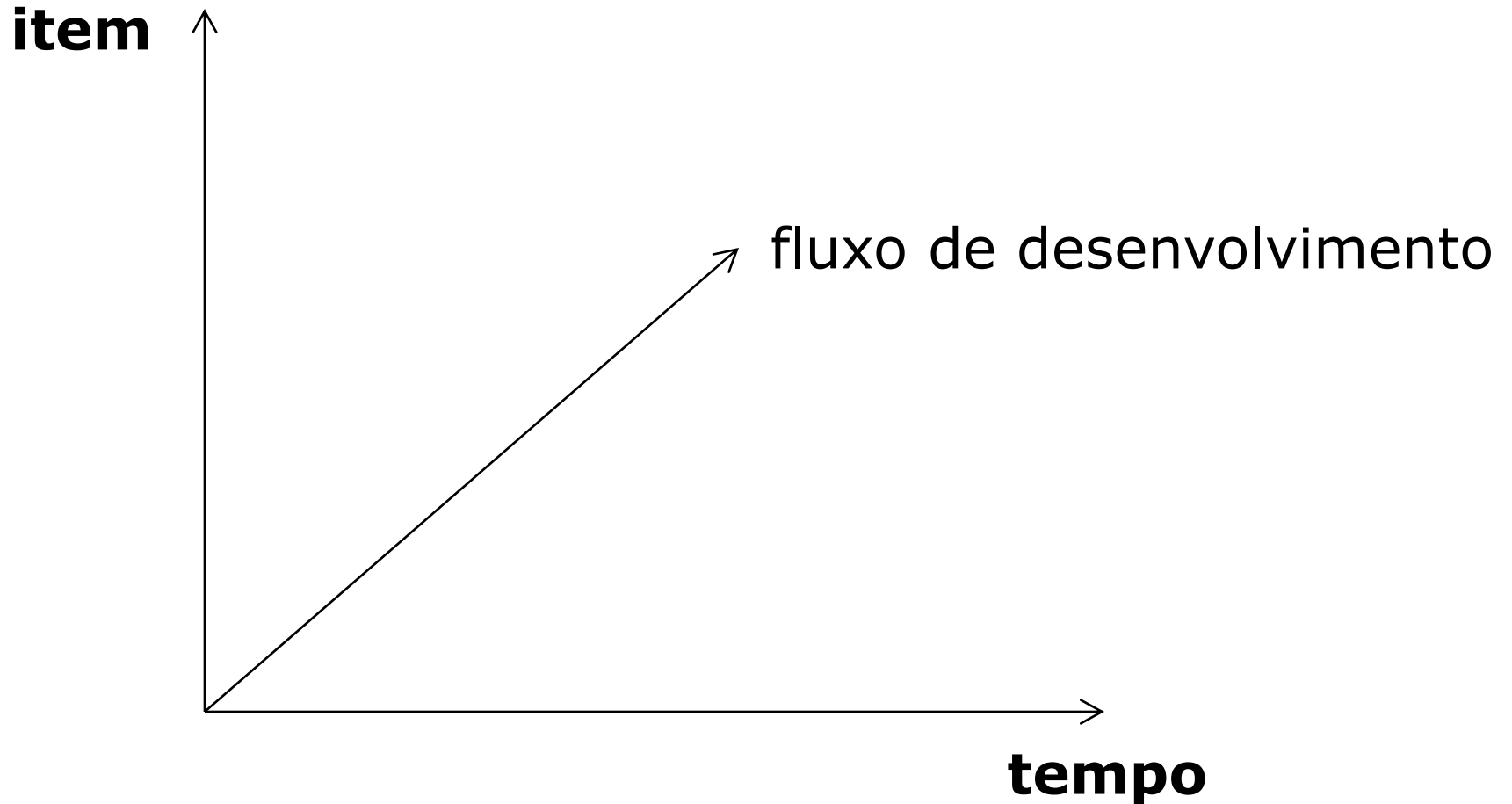
- Um projeto de desenvolvimento de software produz os seguintes itens:
 - Programas (código fonte, programas executáveis, bibliotecas de componentes, etc.)
 - Documentação (manuais do usuário, documento de requisitos, modelo de análise e projeto, etc.)
 - Dados (dados de teste e do projeto)
- Esses conjuntos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

Item de Configuração

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



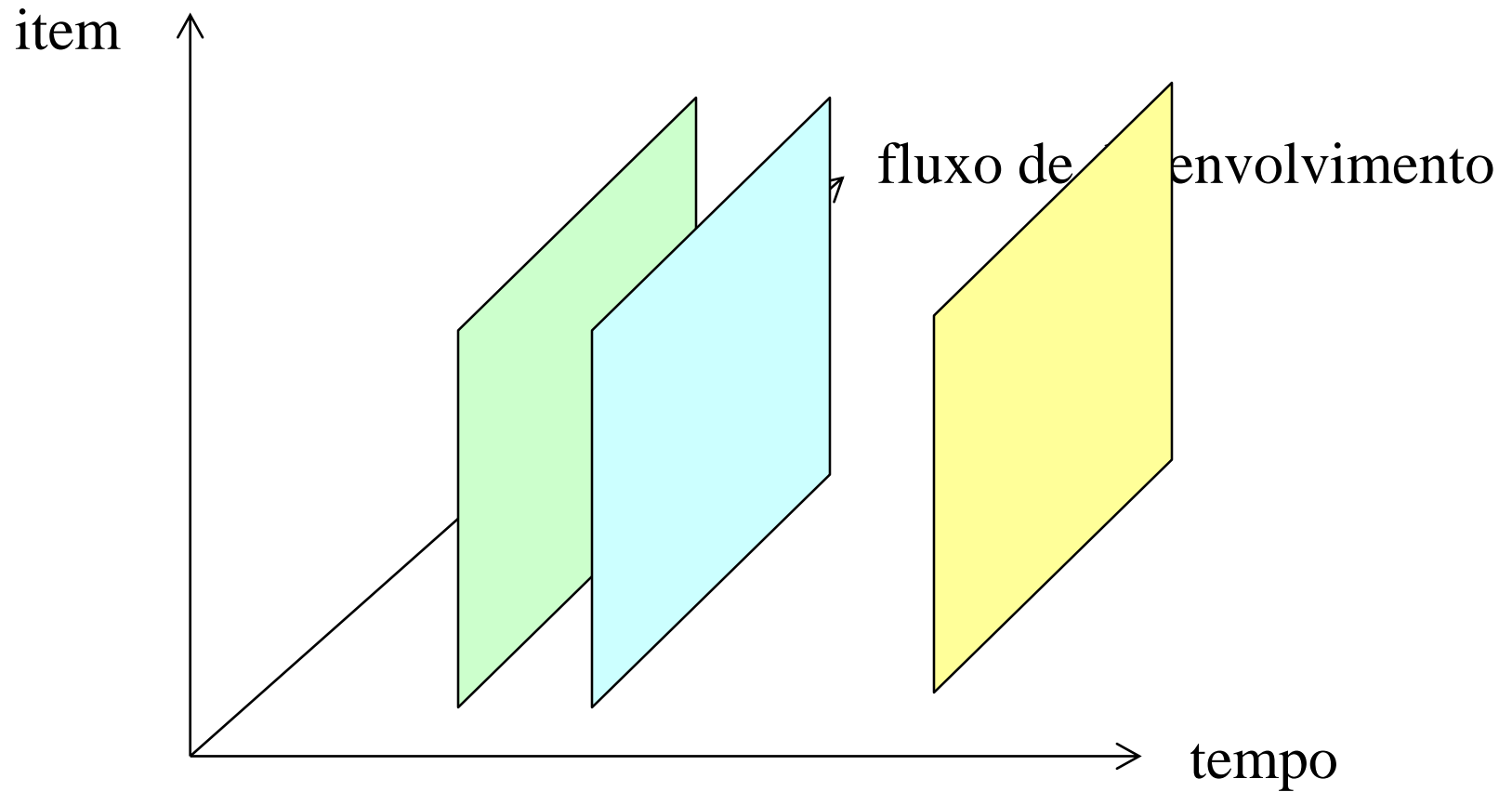
Configuração de Software



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

Baseline



Razões para Criar um Baseline

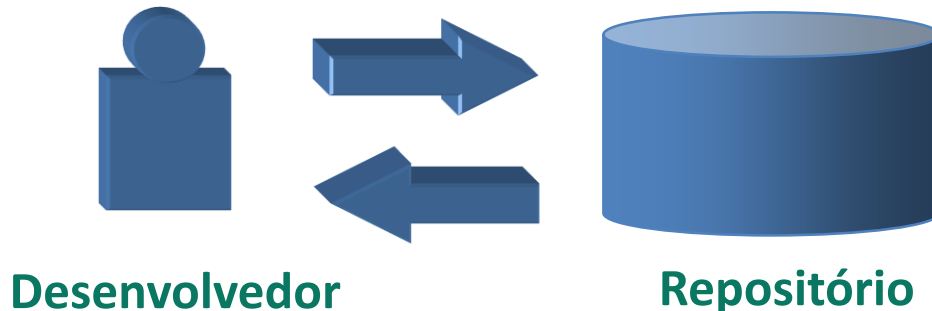
- **Reprodutibilidade** – a habilidade de reproduzir uma versão anterior do sistema
- **Rastreabilidade** – Estabelece uma relação predecessor-sucessor entre artefatos do projeto (projeto satisfaz requisitos, código implementa projeto, etc.)
- **Geração de Relatórios** – A comparação dos conteúdos de dois *baselines* ajuda na depuração e criação de documentação
- **Controle de Mudanças** – referencial para comparações, discussões e negociações

Baselines importantes

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

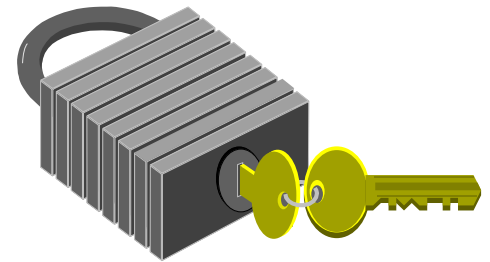
Repositório

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



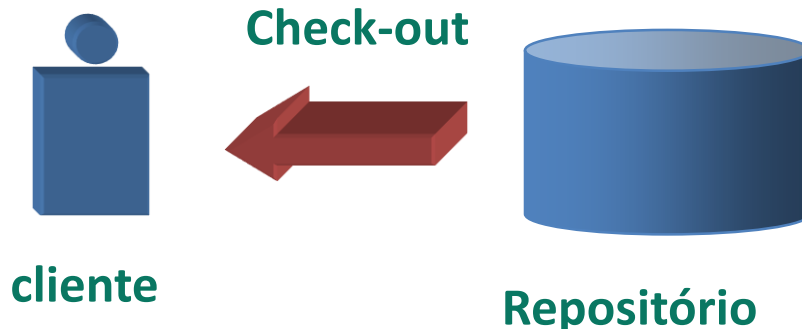
Lock

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



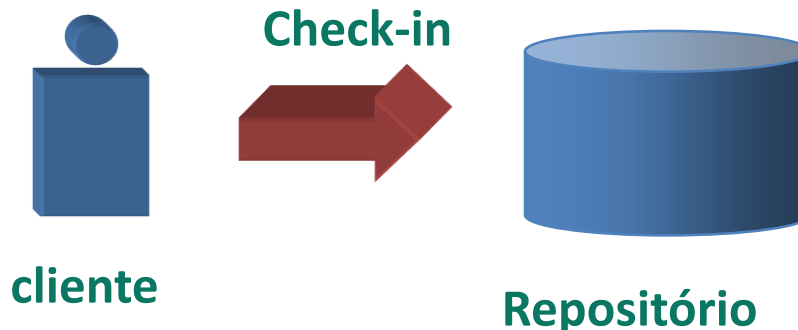
Check-Out

- 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

- 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
- Inclue 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

Os Problemas na Geração de Builds

- Conseqüência: 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 de Builds através da Integração Contínua

- Geração freqüente (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 freqüência adequada

Release

- 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, avaliados, 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

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

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

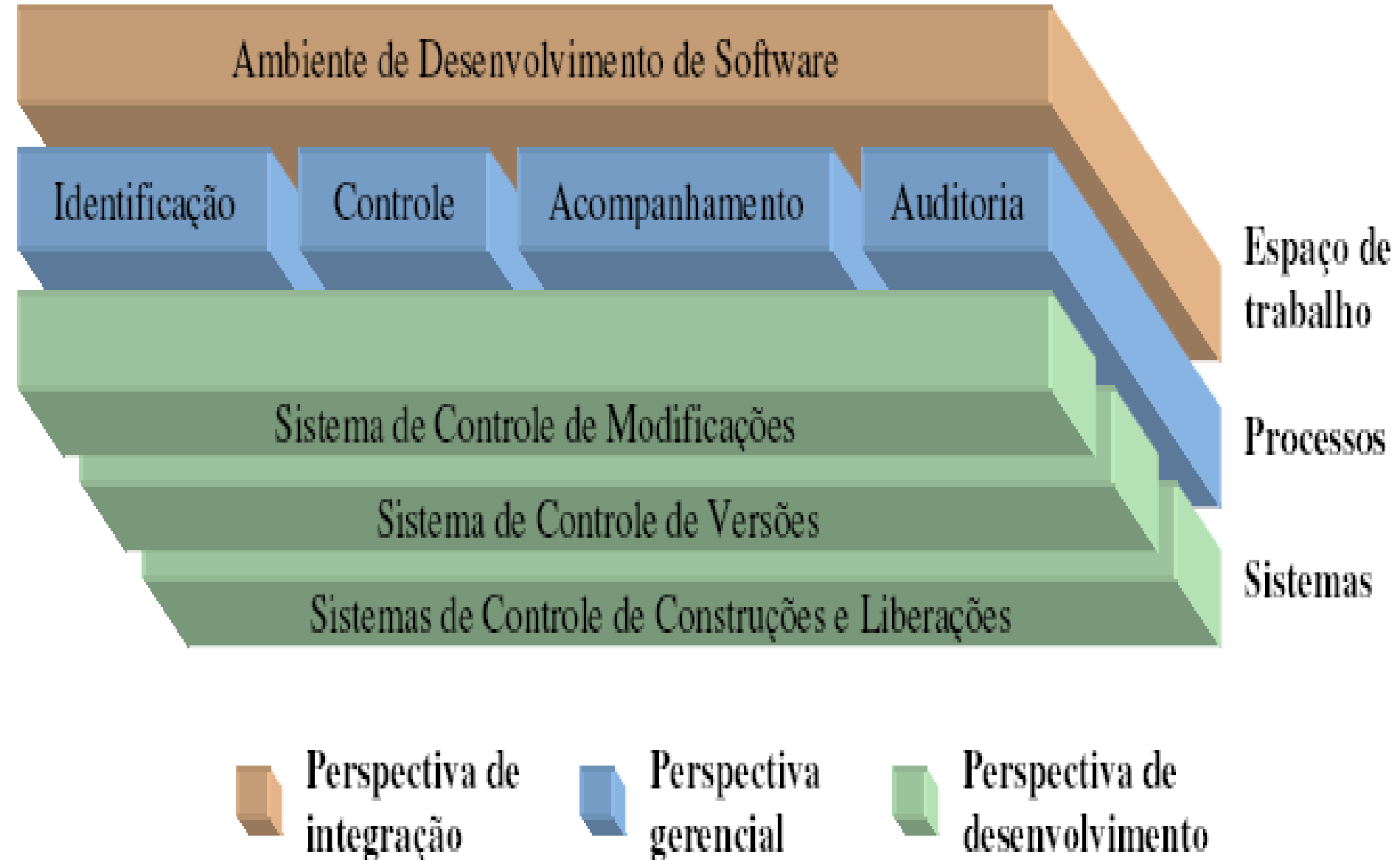
Merge

- 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

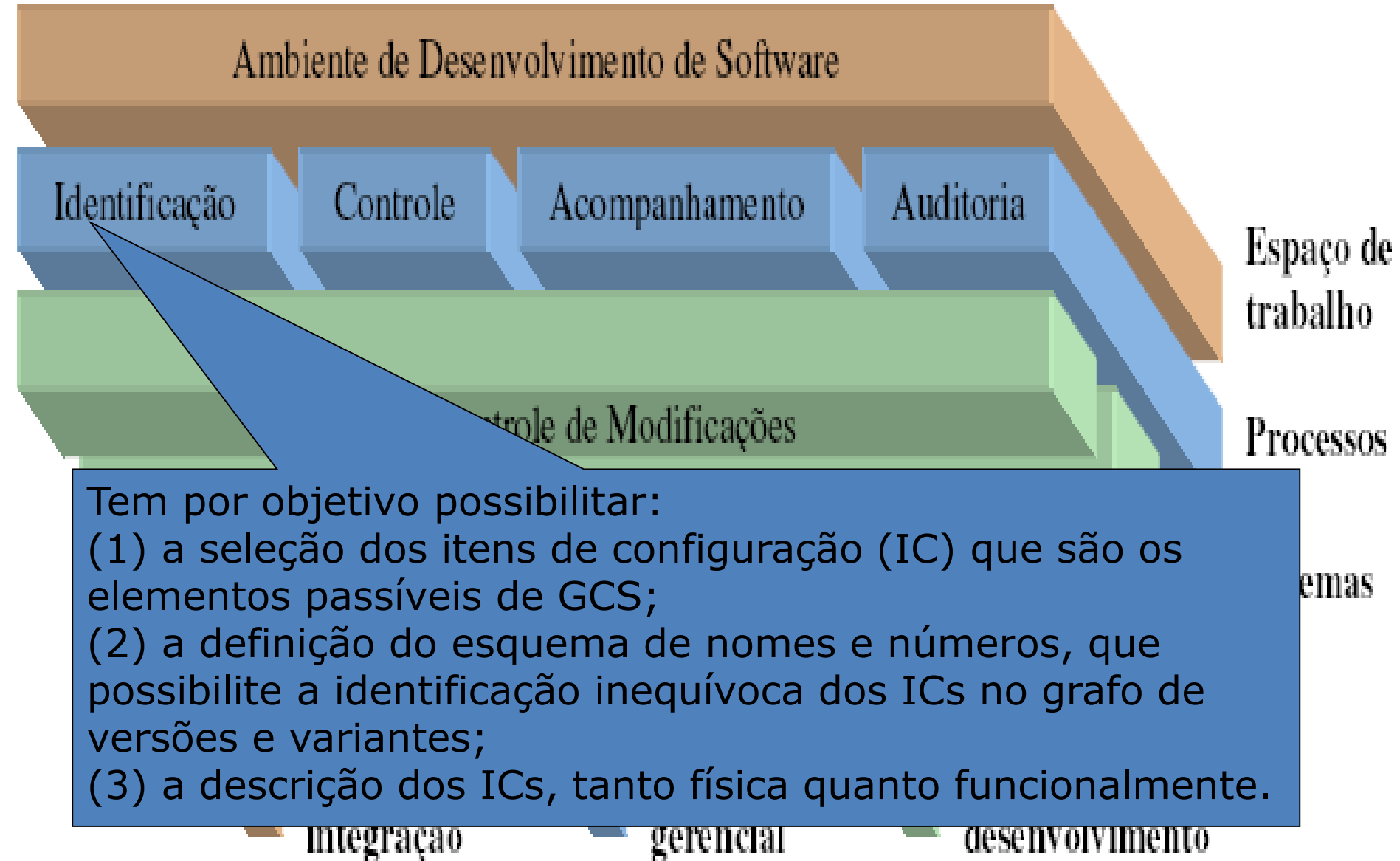
Oportunidades criadas com GC

- 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

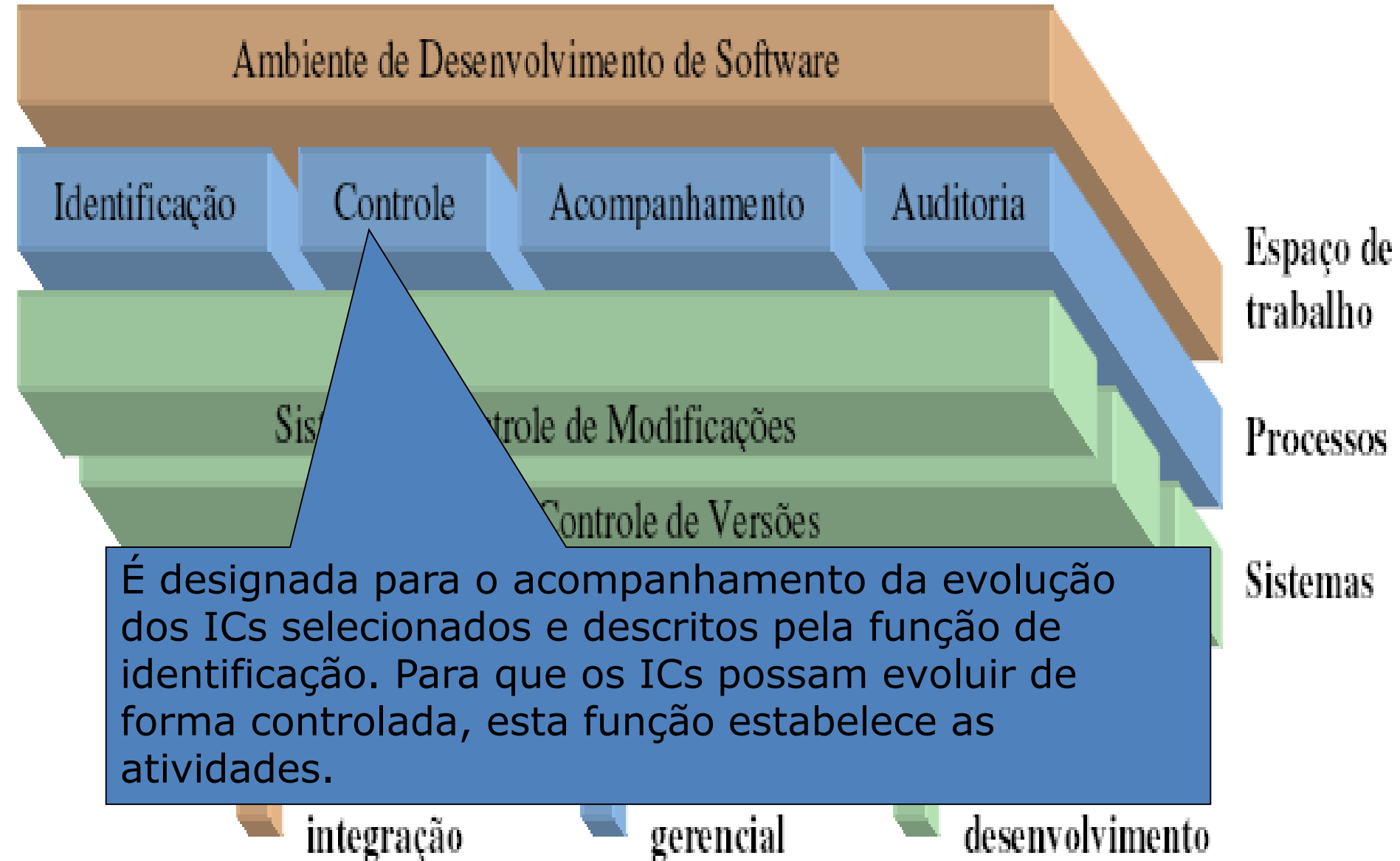
Gerência de Configuração



Gerência de Configuração



Gerência de Configuração



Controle da Configuração

1. Requisição de modificação, iniciando um ciclo da função de controle dado um pedido de manutenção, que pode ser corretiva, evolutiva, adaptativa ou preventiva
2. Classificação da modificação, que estabelece a prioridade do pedido em relação aos demais pedidos efetuados anteriormente;
3. Análise da modificação, que visa relatar os impactos em esforço, cronograma e custo e definir uma proposta de implementação da manutenção;
4. avaliação da modificação pelo Comitê de Controle da Configuração
 - Estabelece se o pedido será implementado, rejeitado ou postergado, em função do laudo fornecido pela análise da modificação;
5. Implementação da modificação, caso o pedido tenha sido aprovado pela avaliação da modificação;
6. Verificação da modificação, aplicando uma bateria de testes de sistema e validando com a proposta de implementação levantada na análise da modificação;
7. Geração de configuração de referência, que pode ou não ser liberada para o cliente em função da sua importância e questões de marketing associadas.

Gerência de Configuração

Ambiente de Desenvolvimento de Software

Identificação

Controle

Acompanhamento

Auditoria

Espaço de
trabalho

Processos

Sistemas

(1) armazenar as informações geradas pelas demais funções;
(2) permitir que essas informações possam ser acessadas em função de necessidades específicas. Essas necessidades específicas abrangem o uso de métricas para a melhoria do processo, a estimativa de custos futuros e a geração de relatórios gerenciais.

integração

gerencial

desenvolvimento

Gerência de Configuração

Ambiente de Desenvolvimento de Software

Identificação

Controle

Acompanhamento

Auditoria

Espaço de
trabalho

Processos

Sistemas

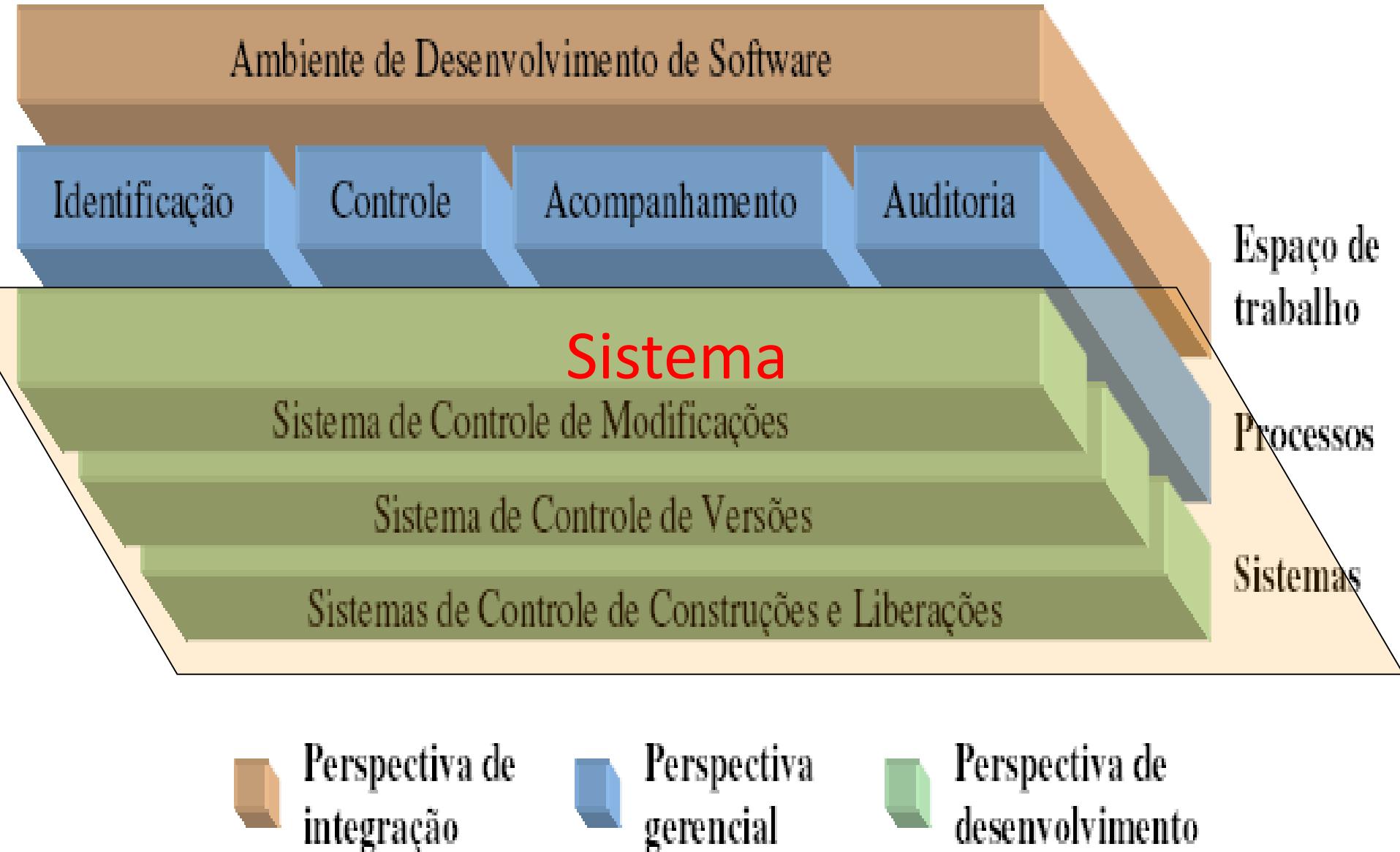
Ocorre quando a configuração de referência, gerada na função de controle da configuração, é selecionada para ser liberada para o cliente.

Suas atividades compreendem:

(1) verificação funcional da configuração de referência, através da revisão dos planos, dados, metodologia e resultados dos testes, assegurando que a mesma cumpra corretamente o que foi especificado;

(2) verificação física da configuração de referência, com o objetivo de certificar que a mesma é completa em relação ao que foi acertado em cláusulas contratuais.

Gerência de Configuração



Sistema de Controle de Modificações

- É encarregado de executar a função de controle da configuração de forma sistemática
- Como?
 - Armazenando todas as informações geradas durante o andamento das requisições de modificação e relatando essas informações aos participantes interessados e autorizados, assim como estabelecido pela função de acompanhamento da configuração.

Sistema de Controle de Construções e Liberações

- Automatiza o complexo processo de transformação dos diversos artefatos de software que compõem um projeto no sistema executável propriamente dito, de forma aderente aos processos, normas, procedimentos, políticas e padrões definidos para o projeto.
- Além disso, esse sistema estrutura as configurações de referência selecionadas para liberação, conforme necessário para a execução da função de auditoria da configuração.

Sistema de Controle de Versões

- Permite que os ICs sejam identificados, segundo estabelecido pela função de identificação da configuração, e que eles evoluam de forma distribuída e concorrente, porém disciplinada
- Essa característica é necessária para que diversas requisições de modificação efetuadas através da função de controle da configuração possam ser tratadas em paralelo, sem corromper o sistema de GCS como um todo

Desafios

- Cultura organizacional
 - Agrupamento de solicitações em releases bem definidos e estabelecidos deve ser negociado com os *stakeholders* do projeto
 - Releases internos utilizados de forma efetiva como ferramenta de gestão de projeto
- Integração entre sistemas de controle de versão e mudanças

Conceitos Fundamentais

Ferramentas

Ferramentas de Apoio à Gerência de Configuração

Ferramenta de Controle de Versões (CVS, por exemplo)

- Manter todos os arquivos em um repositório central
- Controlar o acesso a esse repositório, de modo a garantir a consistência dos artefatos

Ferramentas de Geração de Builds (Ant, por exemplo)

- Automatizar o processo de geração de *builds*

Ferramentas de Gestão de Solicitações de Mudanças (Bugzilla, por exemplo)

- Automatizar o processo de submissão e gestão de SMs