

## CONCEITOS BÁSICOS DE GERÊNCIA DE VERSÃO


Gibeon Aquino  
gibeon@dimap.ufrn.br

### GERÊNCIA DE CONFIGURAÇÃO


“é o conjunto de atividades projetadas para **controlar** as mudanças através da identificação dos produtos do trabalho que serão alterados, estabelecendo um relacionamento entre eles, definindo o mecanismo para o gerenciamento de diferentes versões destes produtos, controlando as mudanças impostas, e **auditando** e **relatando** as mudanças realizadas.”

*Roger Pressman,  
Software Engineering: A Practitioner's Approach*

## OBJETIVOS DA GERÊNCIA DE CONFIGURAÇÃO

- Definir políticas para controle de versões garantindo a consistência dos artefatos produzidos
  - Definir procedimentos para solicitações de mudanças
  - Administrar e auditar o ambiente de GC
  - Facilitar e automatizar a geração de build do sistema
  - Facilitar a geração de Release
- 

## BENEFÍCIOS

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

SIM, MAS....

# O que é essa Configuração que estamos tentando Gerenciar ?

## CONFIGURAÇÃO DE SOFTWARE

- 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)
- O conjunto destes itens é chamado de **configuração do software**

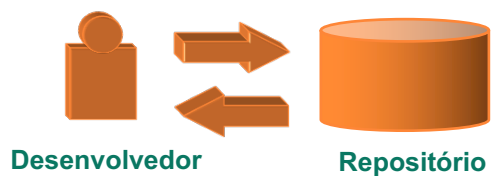
## ITEM DE CONFIGURAÇÃO

- Um conjunto de artefatos de software, que representa uma entidade única para gerência de configuração;
- Um item de configuração **está sujeito a mudanças** e essas devem obedecer às políticas estabelecidas.

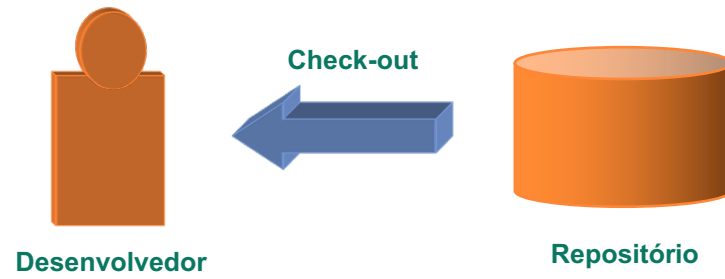


## REPOSITÓRIO

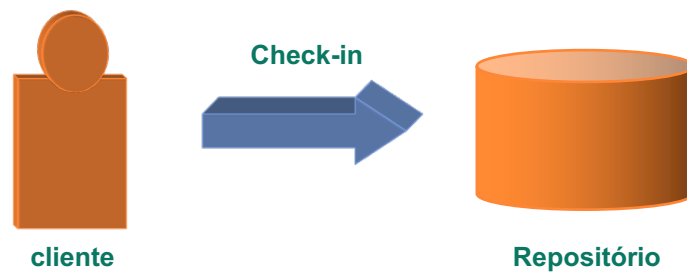
- Local (físico e lógico) onde os itens de um sistema são guardados
- Pode conter diversas versões do sistema
- Utiliza mecanismos de controle de acesso



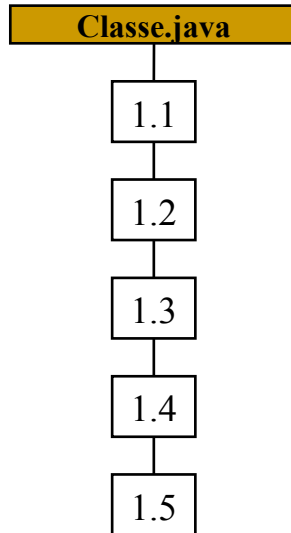
## CHECKOUT



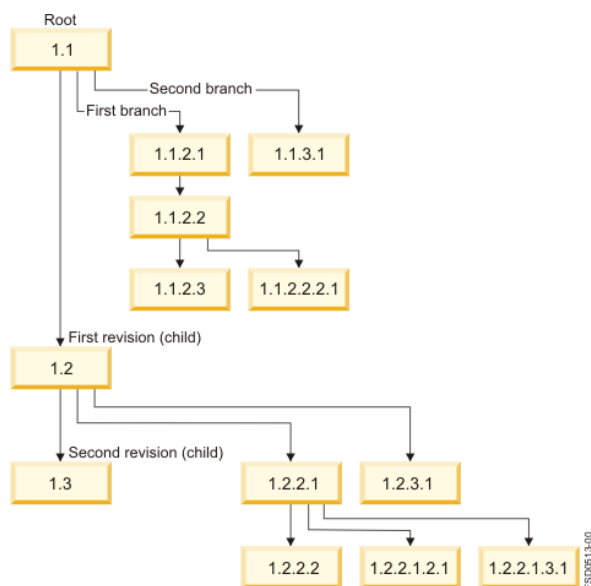
## CHECKIN



## ÁRVORE DE VERSÕES



## ÁRVORE DE VERSÕES

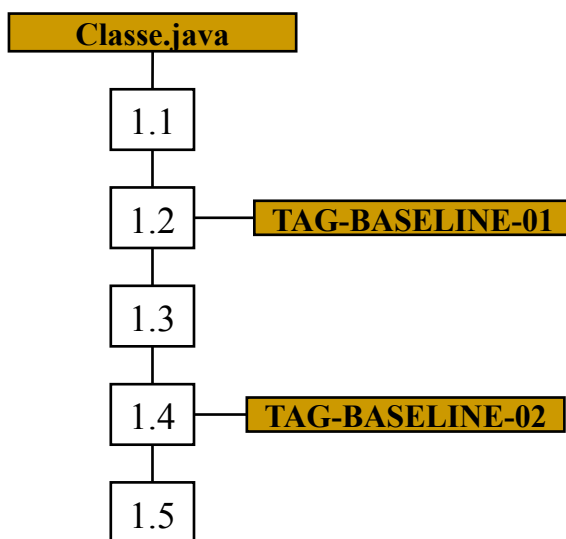


## TAGS

- Rótulos que são associados a conjuntos de arquivos
- Uma tag referencia um ou mais arquivos em um ou mais diretórios, **mas em somente UMA versão de cada arquivo.**
  - Costuma-se usar tags para:
    - Denominar uma da versão do projeto (um build ou baseline) rotulando todos os arquivos associados ao build ou release

| 13

## TAGS



| 14

## BRANCH

- Criação de uma codeline alternativa para trabalho paralelo em um mesmo item de configuração.
- Regras bem definidas para criação de branches
  - Por que e quando devem ser criados?
  - Quais os passos?
  - Quando retornar ao fluxo principal?

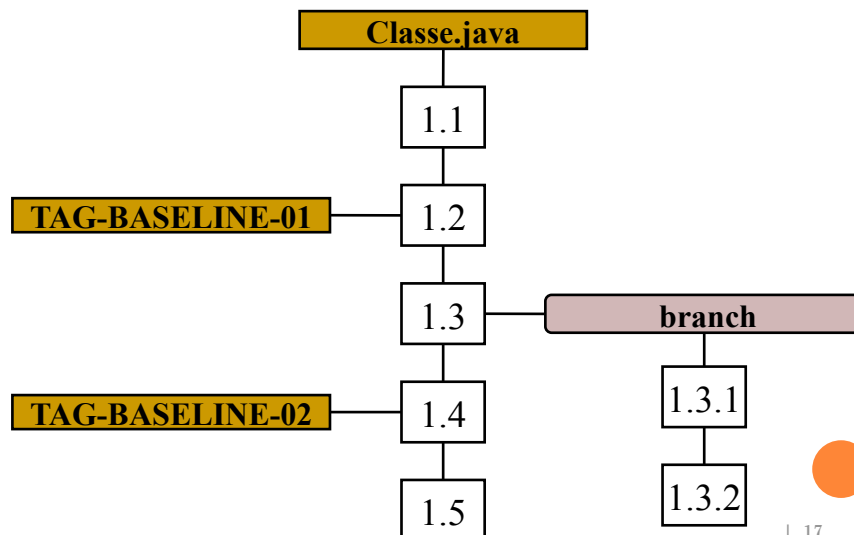
| 15

## BRANCH





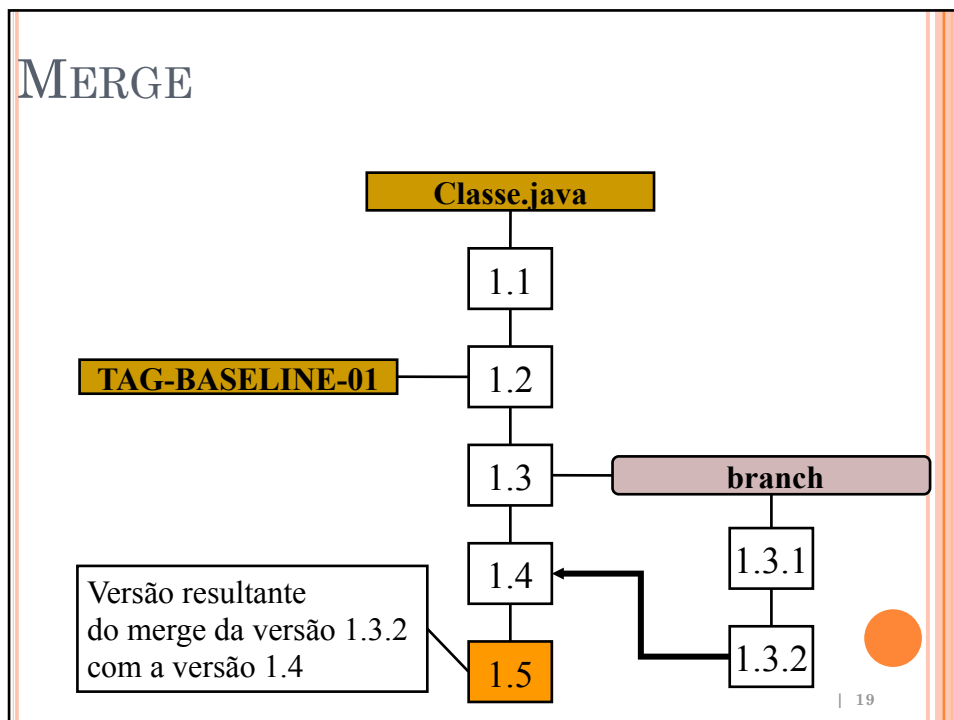
## BRANCH



## MERGE

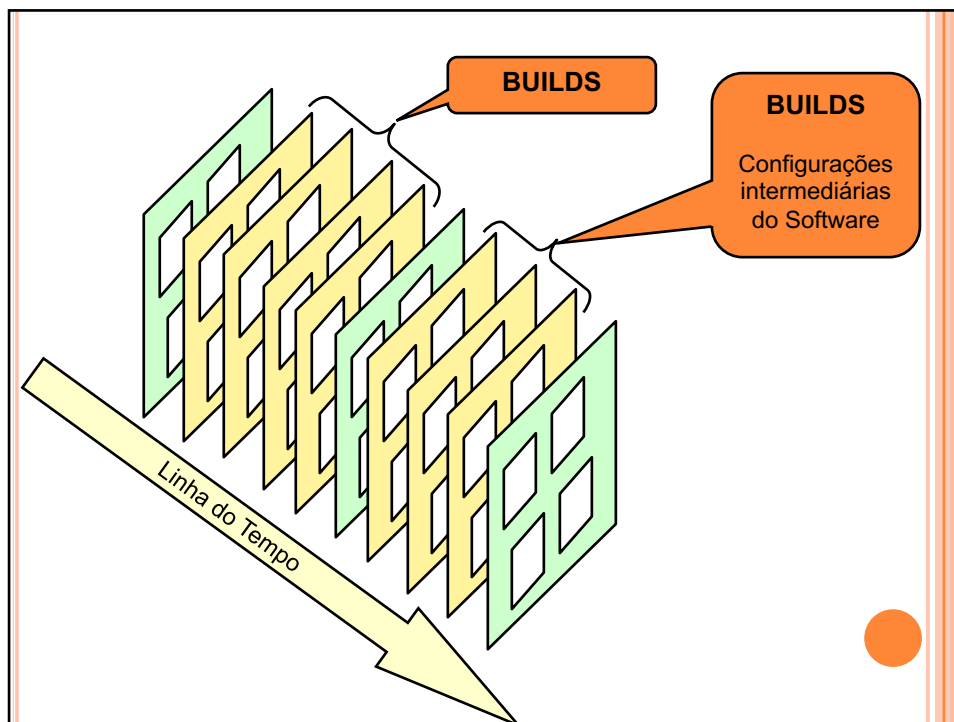
- Unificação de diferentes versões de um mesmo arquivo.
- Geralmente o merge ocorre de um branch para a codeline principal.
- Algumas ferramentas fornecem um mecanismo automático para realização de merges
  - Mesmo com o uso de ferramentas, em vários casos há necessidade de intervenção manual

## MERGE



## BUILD

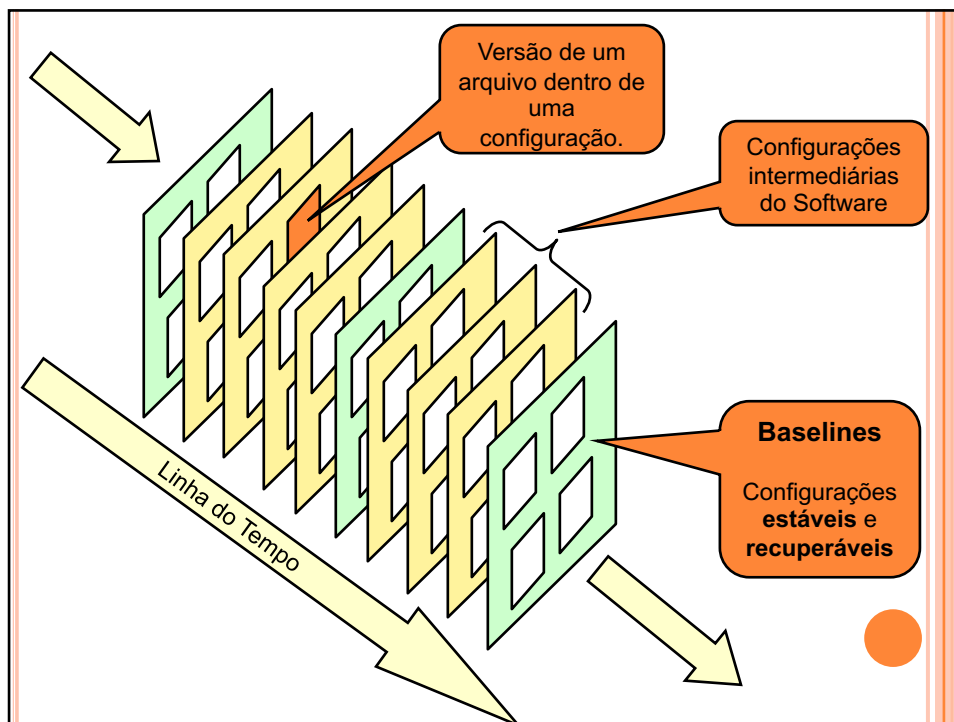
- Representa uma versão ainda incompleta do sistema em desenvolvimento, mas com certa estabilidade
- Costumam apresentar limitações conhecidas
- Geralmente os testes são realizados em builds (que são marcados por uma tag).



## BASELINE

- Uma **CONFIGURAÇÃO** do software em um dado momento que está estável
  - Serve como base para os passos posteriores do desenvolvimento
  - Só deve ser modificada através de **procedimentos formais** (solicitações de mudança)

**Um artefato só se torna um item de configuração depois que ele faz parte da baseline.**



## RAZÕES PARA CRIAR BASELINES

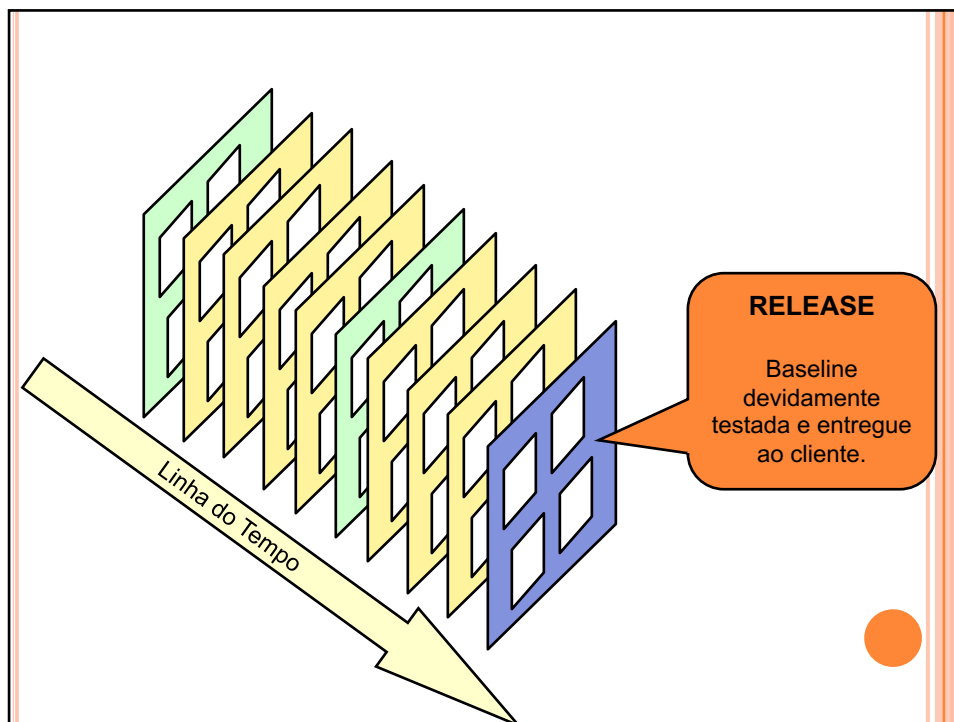
- **Reproducibilidade** – a habilidade de reproduzir uma versão anterior do sistema
- **Rastreabilidade** – Estabelece uma relação predecessor-sucessor entre artefatos do projeto (projeto satisfaz requisitos, código implementa projeto, etc.)
- **Controle de Mudanças** – referencial para comparações, discussões e negociações

## ARTEFATOS DE TRABALHO X BASELINES

- Quais dos artefatos de trabalho deveriam fazer parte da baseline?
  - Documento de Especificação de Requisitos
  - Documento de Caso de Uso
  - Código Fonte
  - Documento de Projeto de Testes
  - ...
- Quando um artefato de trabalho entra na baseline ele vira o que?
  - Item de configuração

## RELEASE

- **Release** = Entrega, liberação.
- Versão do sistema validada após os diversos tipos de teste.
- Supostamente sem erros :P
- Entregue ao cliente ou ao mercado
- Processo **iterativo/incremental** produz, em geral, mais de um release



## CONCLUSÕES

- GC é um fluxo de apoio ao projeto como um todo
- Passos iniciais para a adoção de um processo de software
- Requer uma certa disciplina na manipulação de itens de configuração
- Apoio de ferramentas sempre que possível

