

Gerência de Configuração

Conceitos

Prof. Miquéias Lopes
miqueias.lopes@gmail.com

Agenda

- Configuração de software
- Item de Configuração
- Codeline
- Baseline
- Repositório
- Tags
- Branch
- Merge
- Build
- Release



??????

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

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



Codeline

- Codeline = Linha de desenvolvimento
 - Uma codeline armazena todas as versões (mudanças) de cada artefato em uma determinada linha de tempo.

Arq1.txt

Arq2.txt



Baseline

● Uma **CONFIGURAÇÃO** do software em um dado momento que está estável

- Serve como base para os passos posteriores do desenvolvimento
- Só pode 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 uma 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 duas *baselines* ajuda na criação de documentação
- **Controle de Mudanças** – referencial para comparações, discussões e negociações

Artefatos de trabalho X baseline

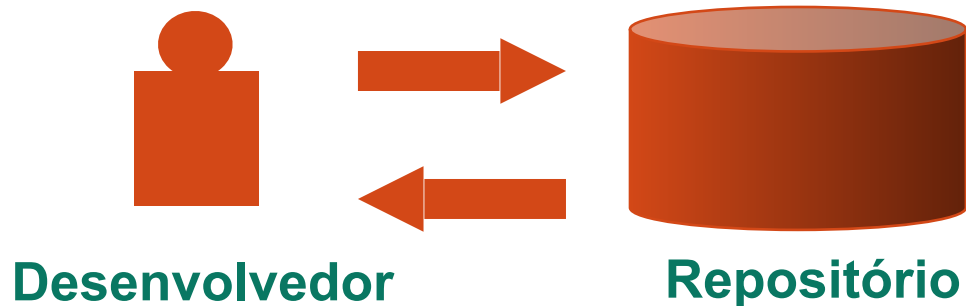
● Quais dos artefatos de trabalho deveriam fazer parte da baseline?

● Exemplos:

- Documento de Especificação de Requisitos
- Documento de Caso de Uso
- Código Fonte
- Documento de Projeto de Testes

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

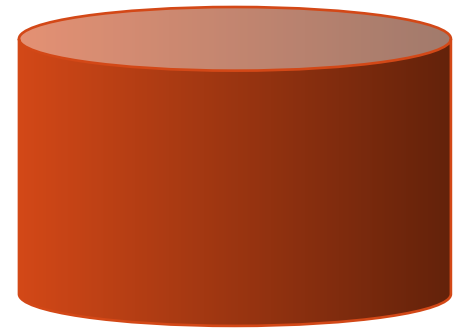


Check-out



cliente

Check-out



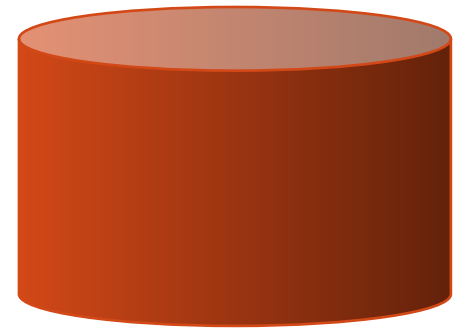
Repositório

Check-in



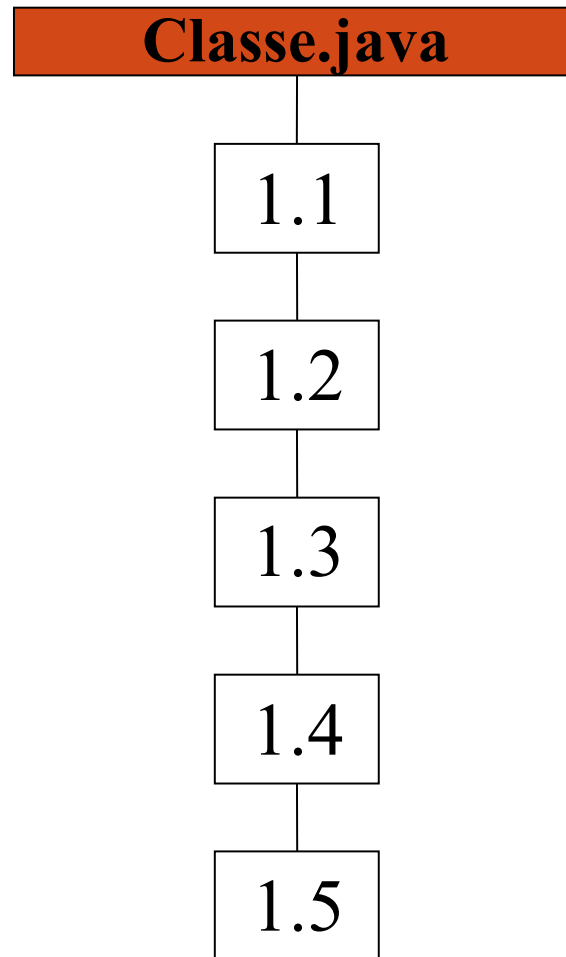
cliente

Check-in



Repositório

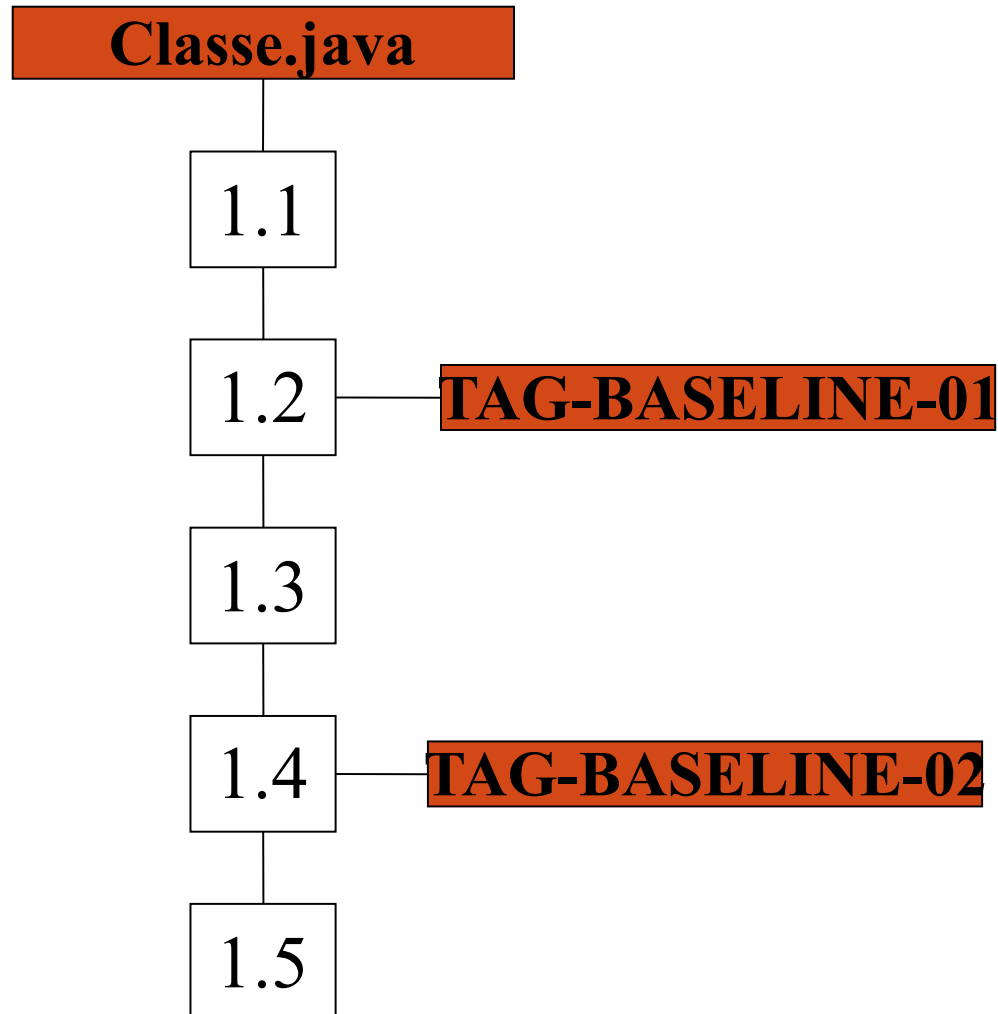
Árvore de versões



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 versão do projeto (*build* ou *baseline*) rotulando todos os arquivos associados.

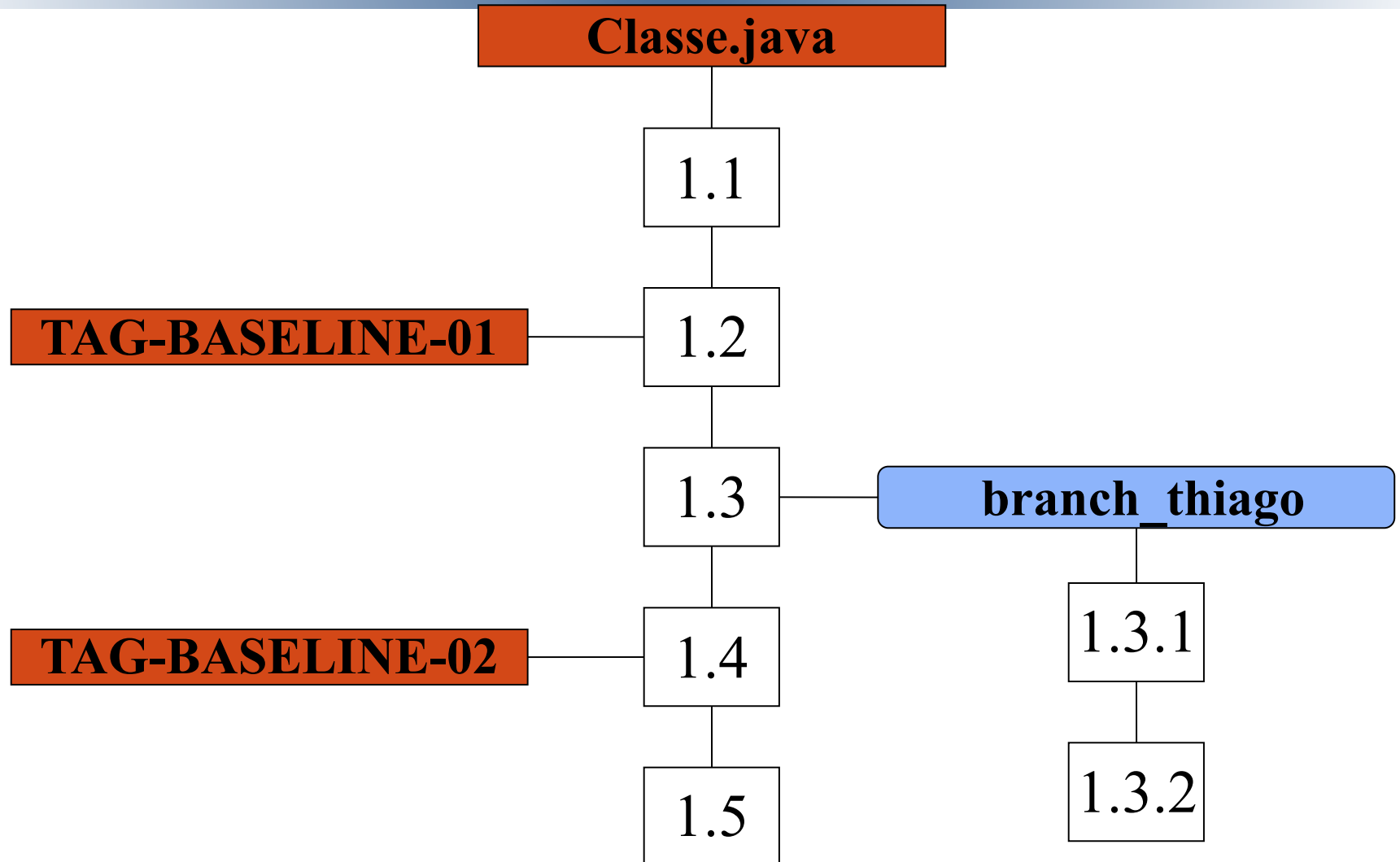
Tags



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?

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

Classe.java

1.1

TAG-BASELINE-01

1.2

1.3

branch thiago

1.3.1

1.4

1.3.2

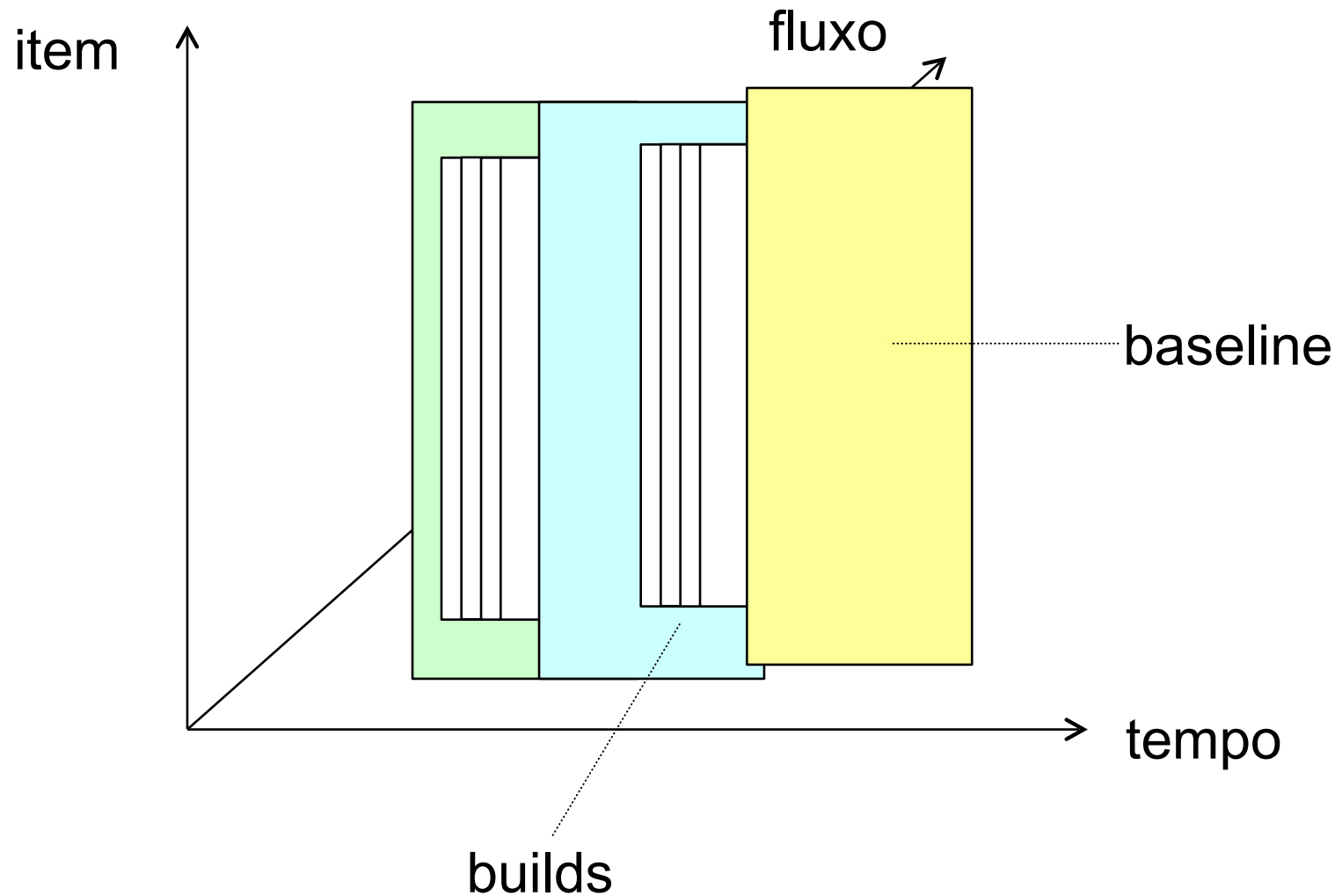
Versão resultante
do merge da versão 1.3.2
com a versão 1.4

1.5

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 marcadas por uma *tag*).

Baseline x Builds



Build

- A política de geração das *builds* deve ser bem definida no planejamento do projeto.
- A geração de *builds* deve ser automatizada e realizada com a frequência adequada.

O que acontece quando uma *build* do sistema é testada com sucesso?

Release

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

Release

- Implantada no cliente
- Deve ser devidamente mantida enquanto a *codeline* principal evolui.
 - Uso de *branches* para dar manutenção a *releases* antigas

Conclusões

- A Gerência de Configuração 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.

Perguntas

