



Introdução à Gerência de Configuração

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br

Definição

- Jacky Estublier
 - GC é a disciplina que nos permite evoluir produtos de software de forma controlada, e, desta forma, contribui na satisfação de restrições de qualidade e de tempo
- IEEE Std 610
 - GC é uma disciplina que aplica procedimentos técnicos e administrativos para identificar e documentar as características físicas e funcionais de um item de configuração, controlar as alterações nessas características, armazenar e relatar o processamento das modificações e o estágio da implementação e verificar a compatibilidade com os requisitos especificados

Histórico

- Anos 50
 - GC para produção de aviões de guerra e naves espaciais
- Anos 60 e 70
 - Surgimento de GCS (S = Software)
 - Foco ainda em aplicações militares e aeroespaciais
- Anos 80 e 90
 - Mudança de foco (MIL → EIA, IEEE, ISO, etc.)
 - Surgimento das primeiras normas internacionais
 - Assimilação por organizações não militares



Sistema x Funções de GC

Espaço de trabalho:

Ambiente de Desenvolvimento de Software

Processos:

Identificação

Controle

Contabilização

Avaliação

Liberação

Sistemas:

Controle de Modificações

Controle de Versões

Gerenciamento de Construção



**Perspectiva de
integração**

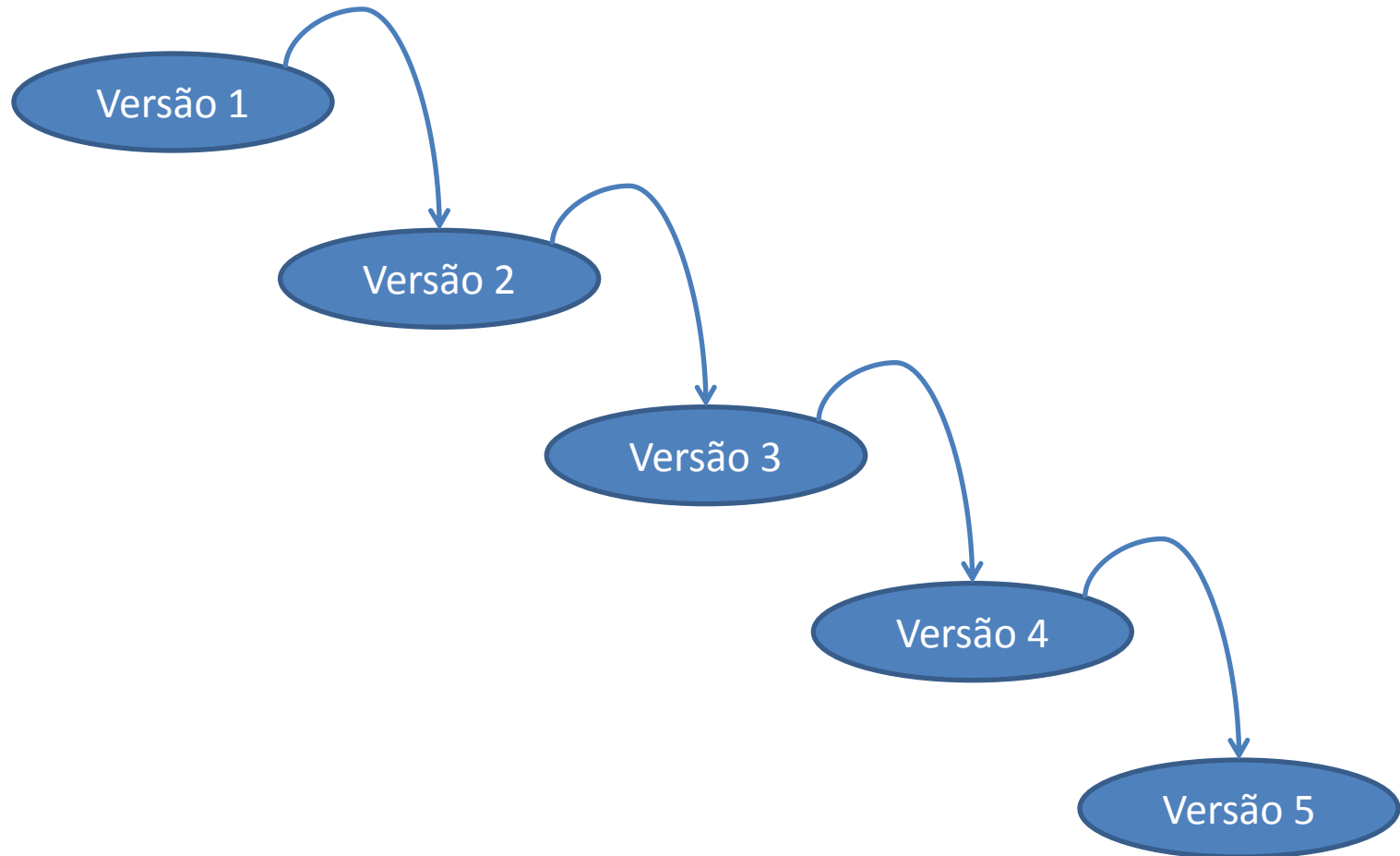


**Perspectiva
gerencial**

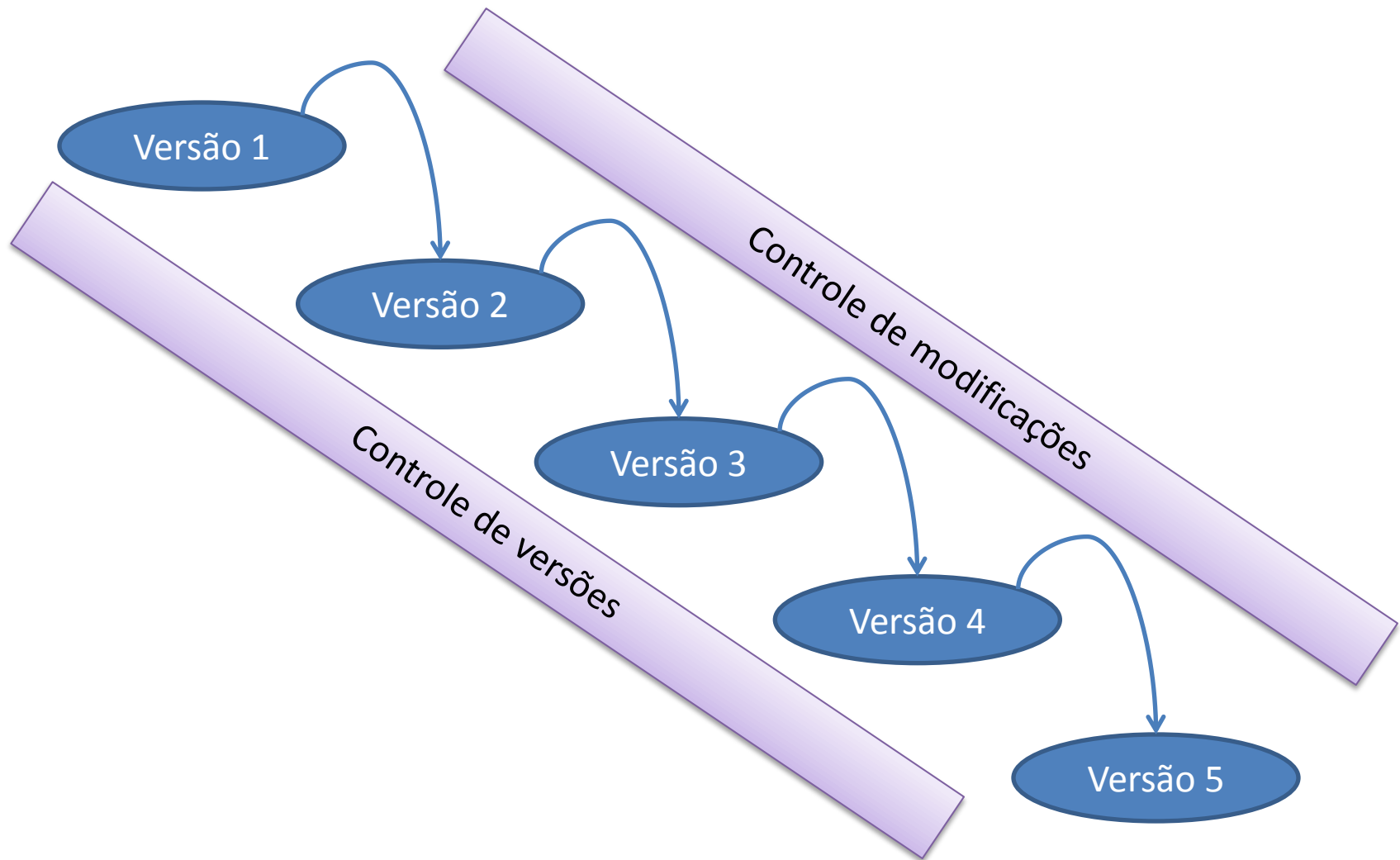


**Perspectiva de
desenvolvimento**

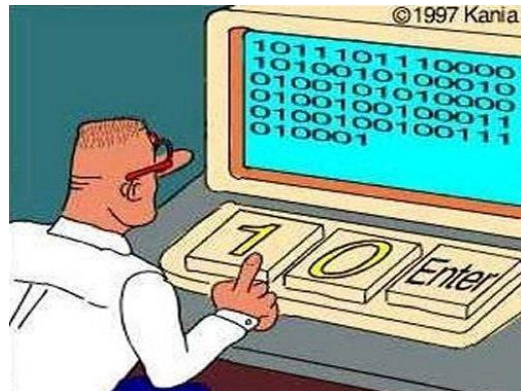
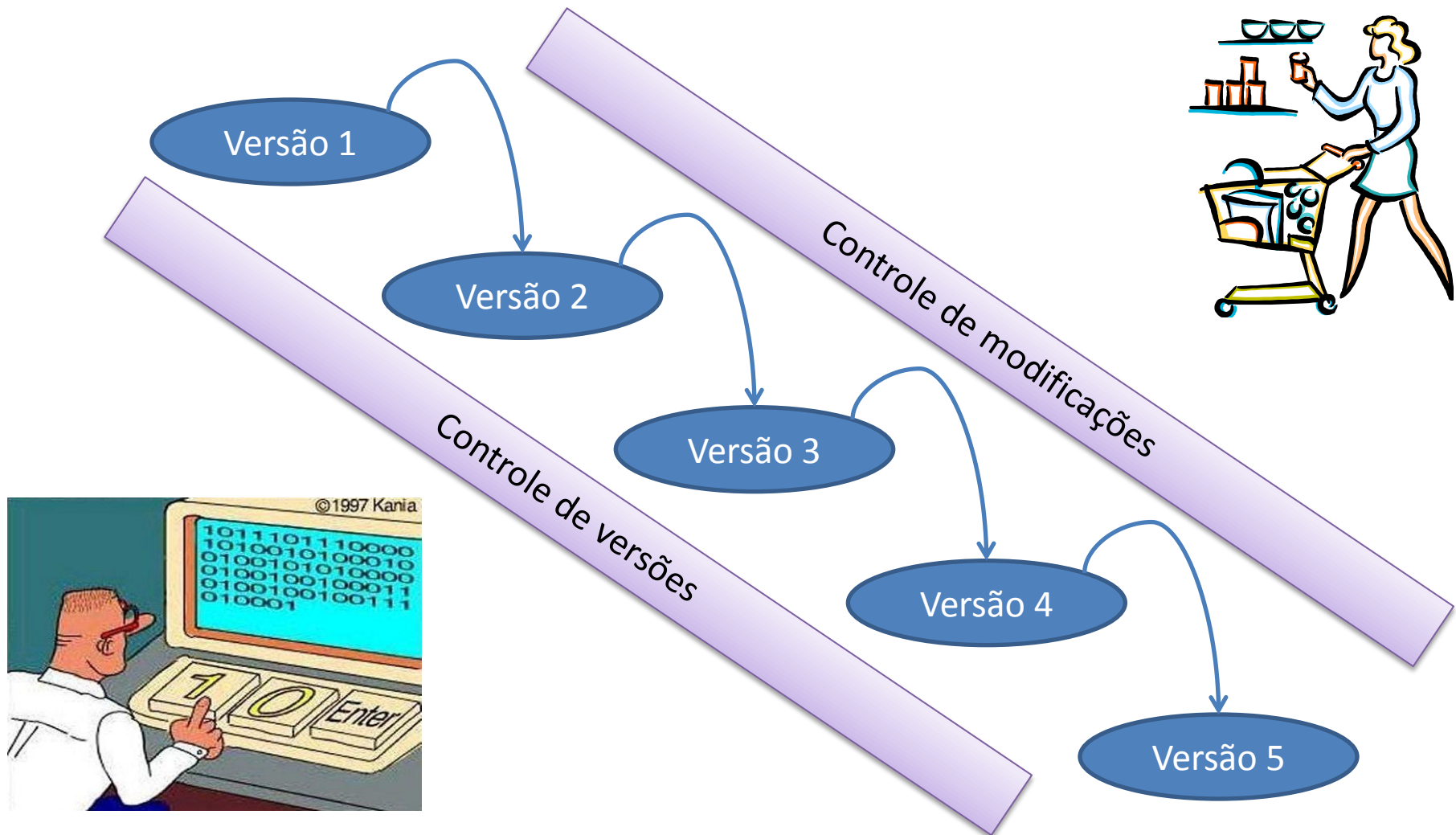
Sistema de Gerência de Configuração



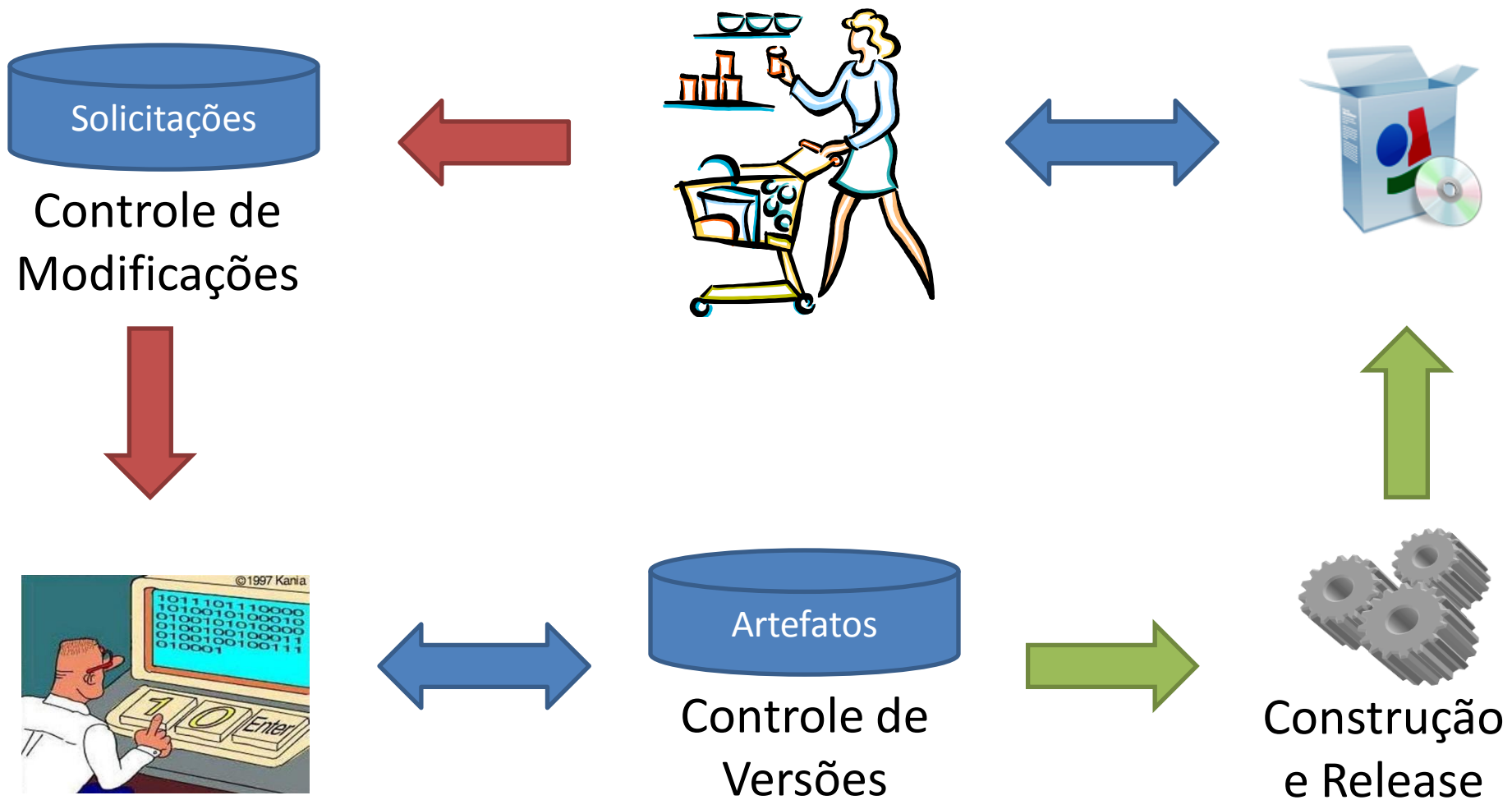
Sistema de Gerência de Configuração



Sistema de Gerência de Configuração



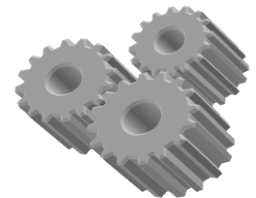
Sistema de Gerência de Configuração



Sistema de Gerência de Configuração



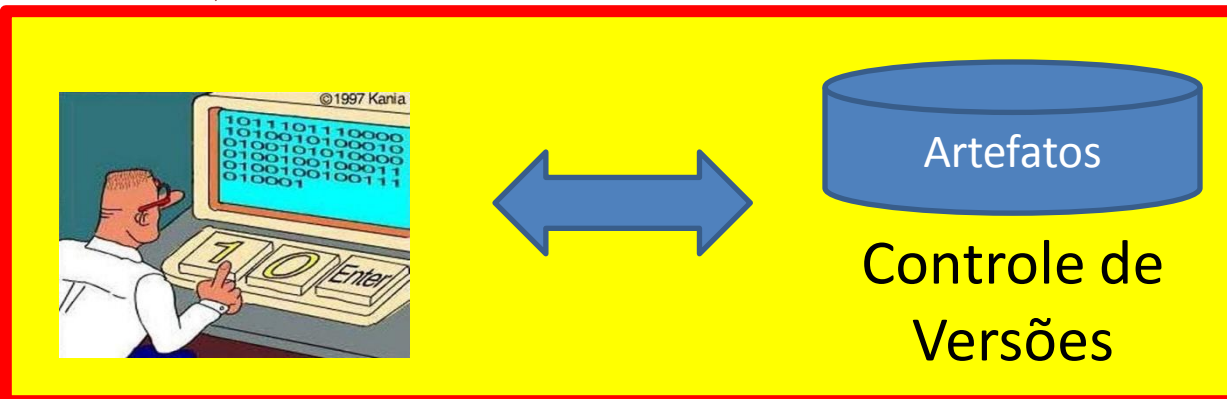
Controle de
Modificações



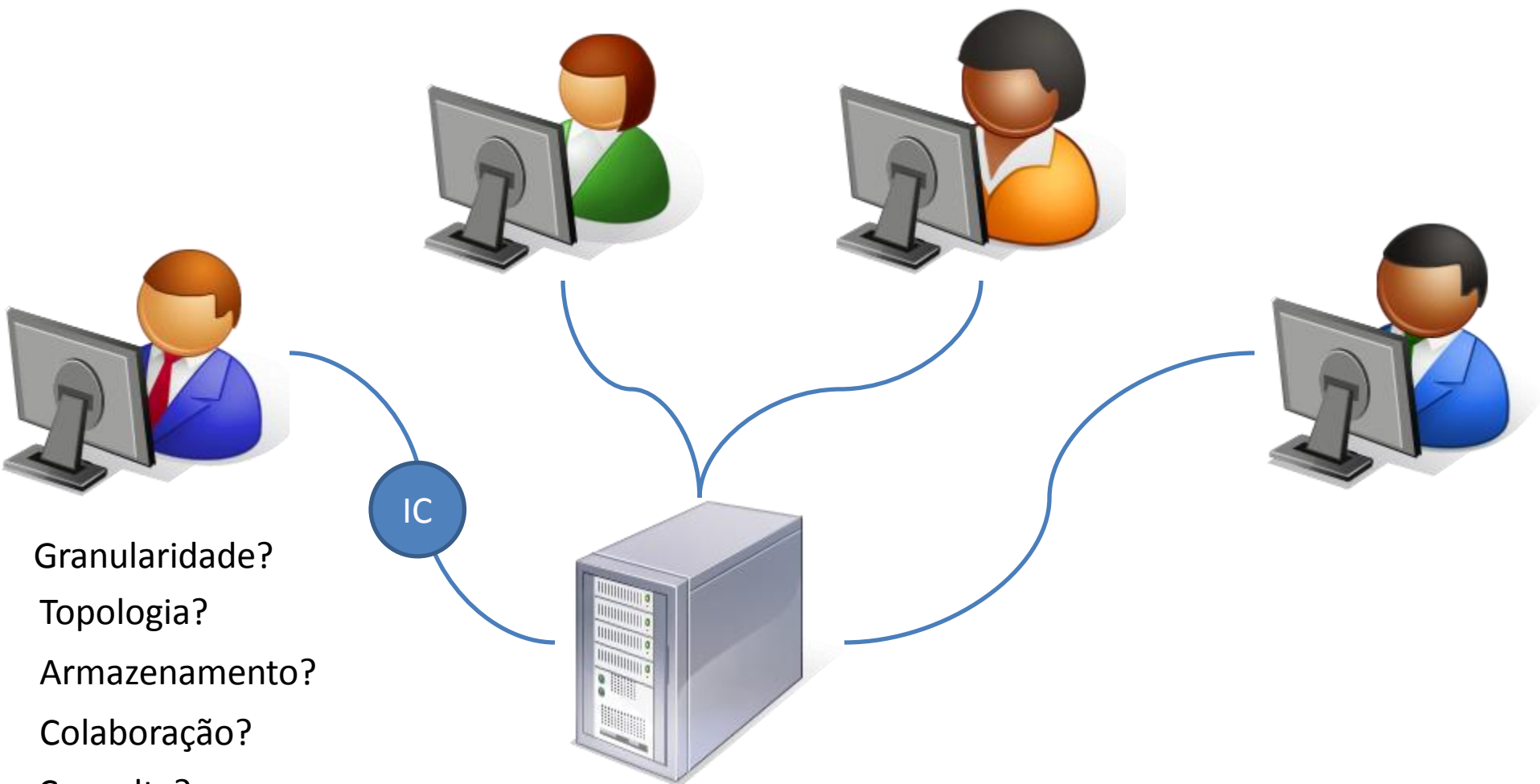
Construção
e Release



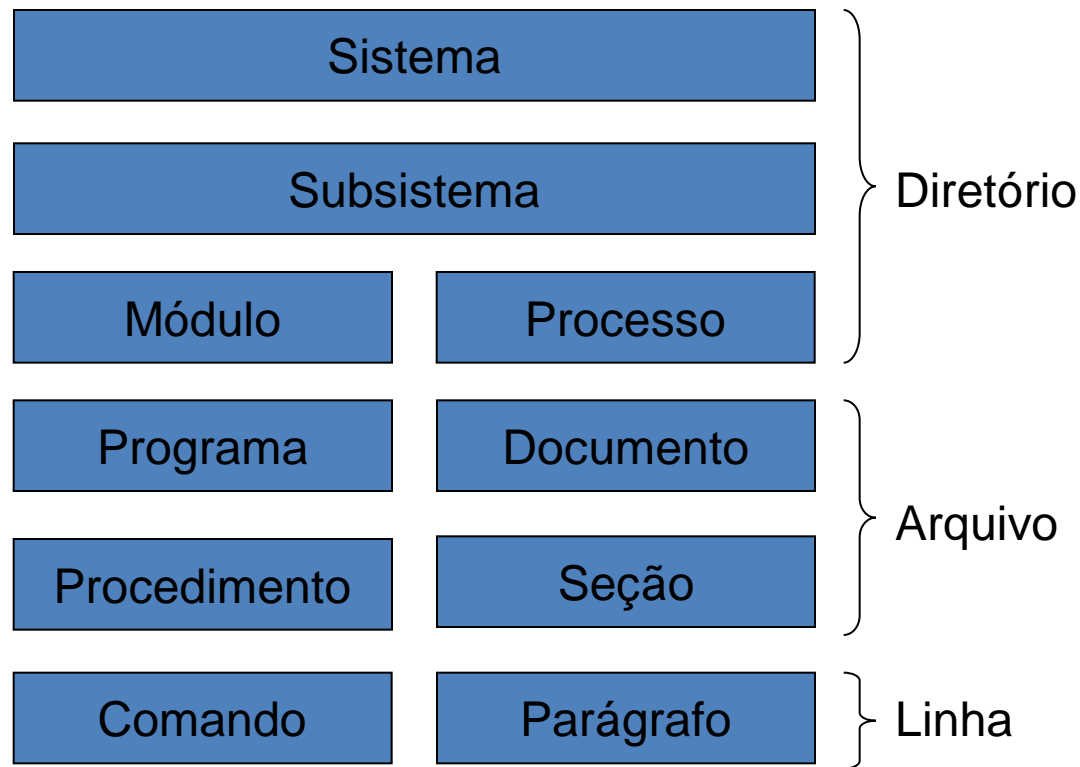
Controle de
Versões



Controle de versões

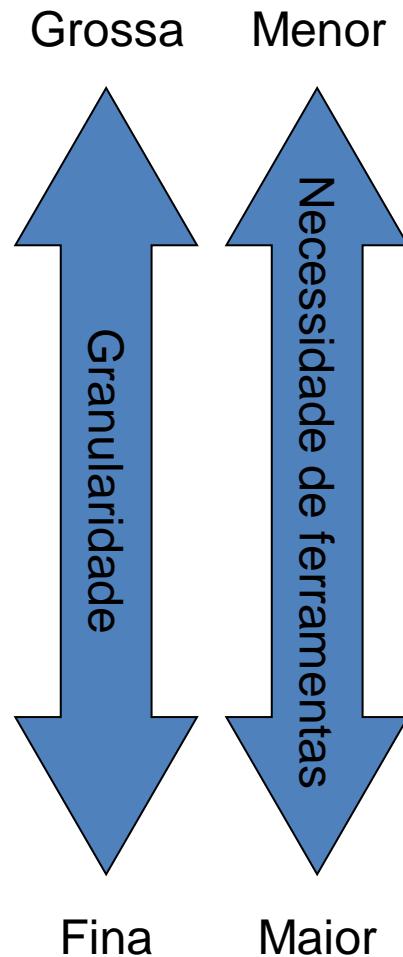


Item de configuração

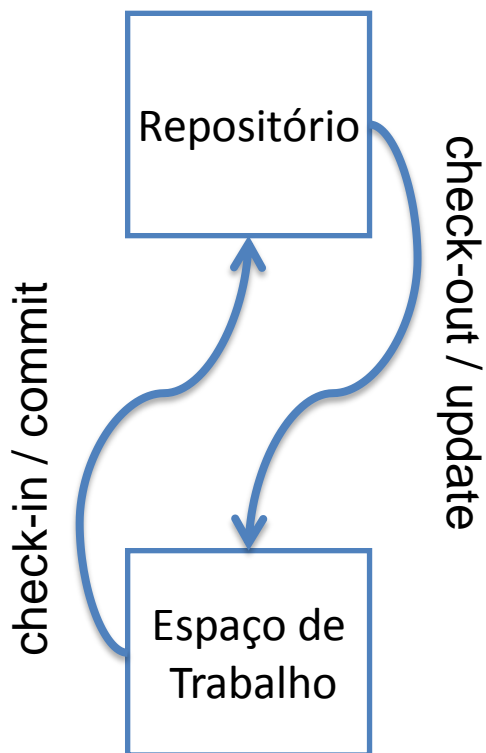


Perspectiva: Desenvolvimento

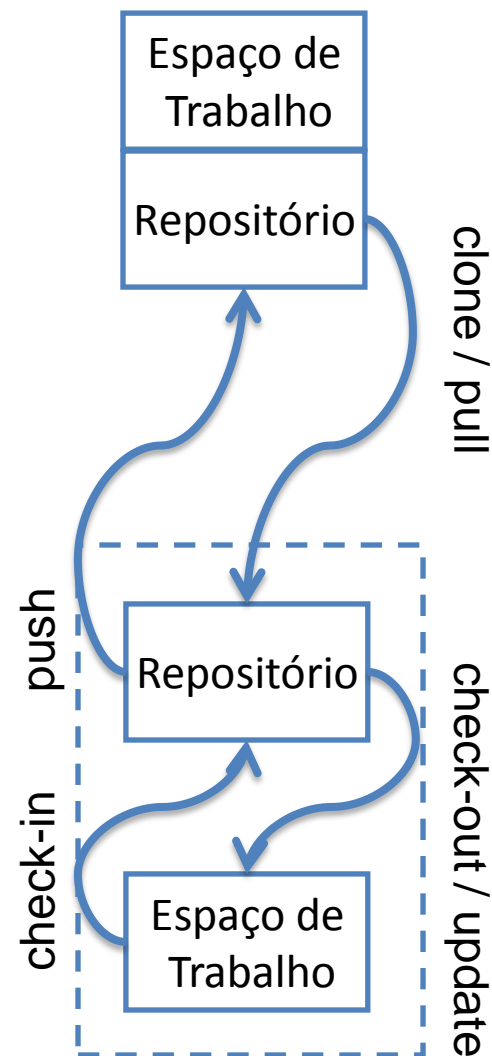
Gerencial



Topologia

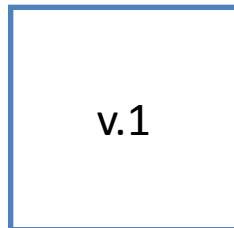
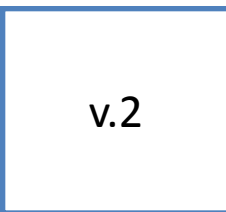
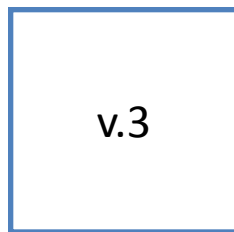


Centralizado

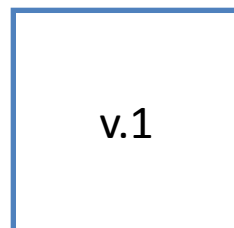


Distribuído

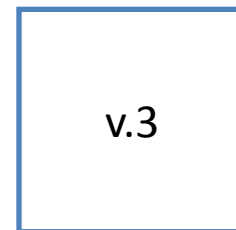
Armazenamento



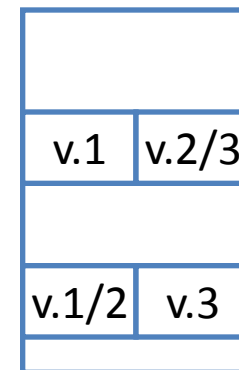
Completo



Forward

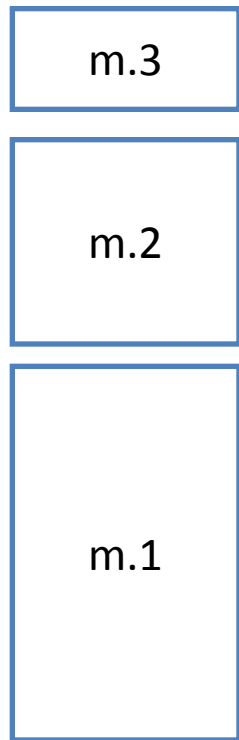


Reverse

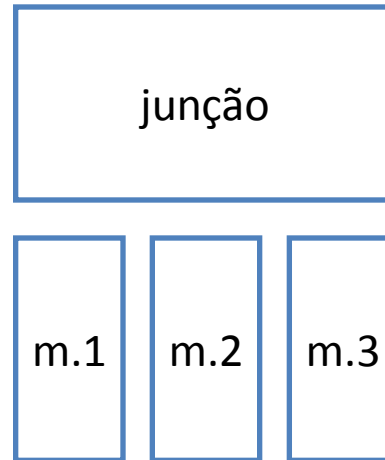


In-line

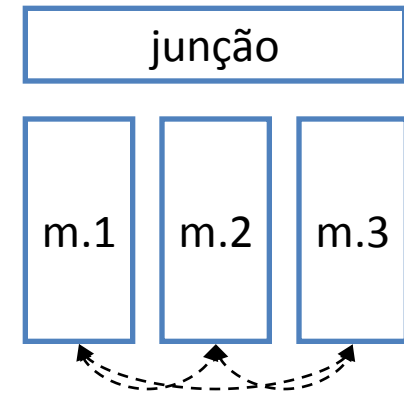
Colaboração



Pessimista

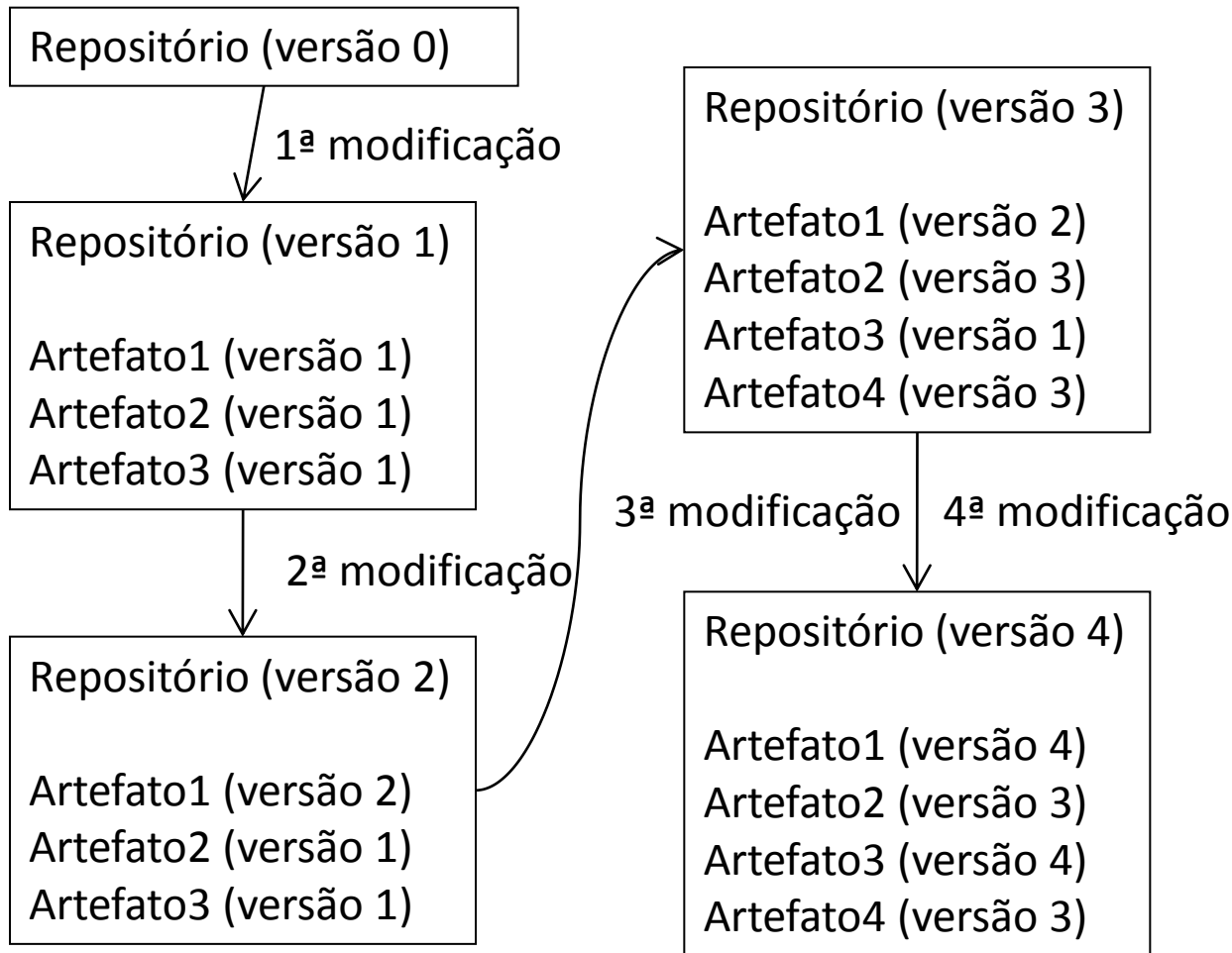


Otimista



Misto

Consulta



Consulta por artefato

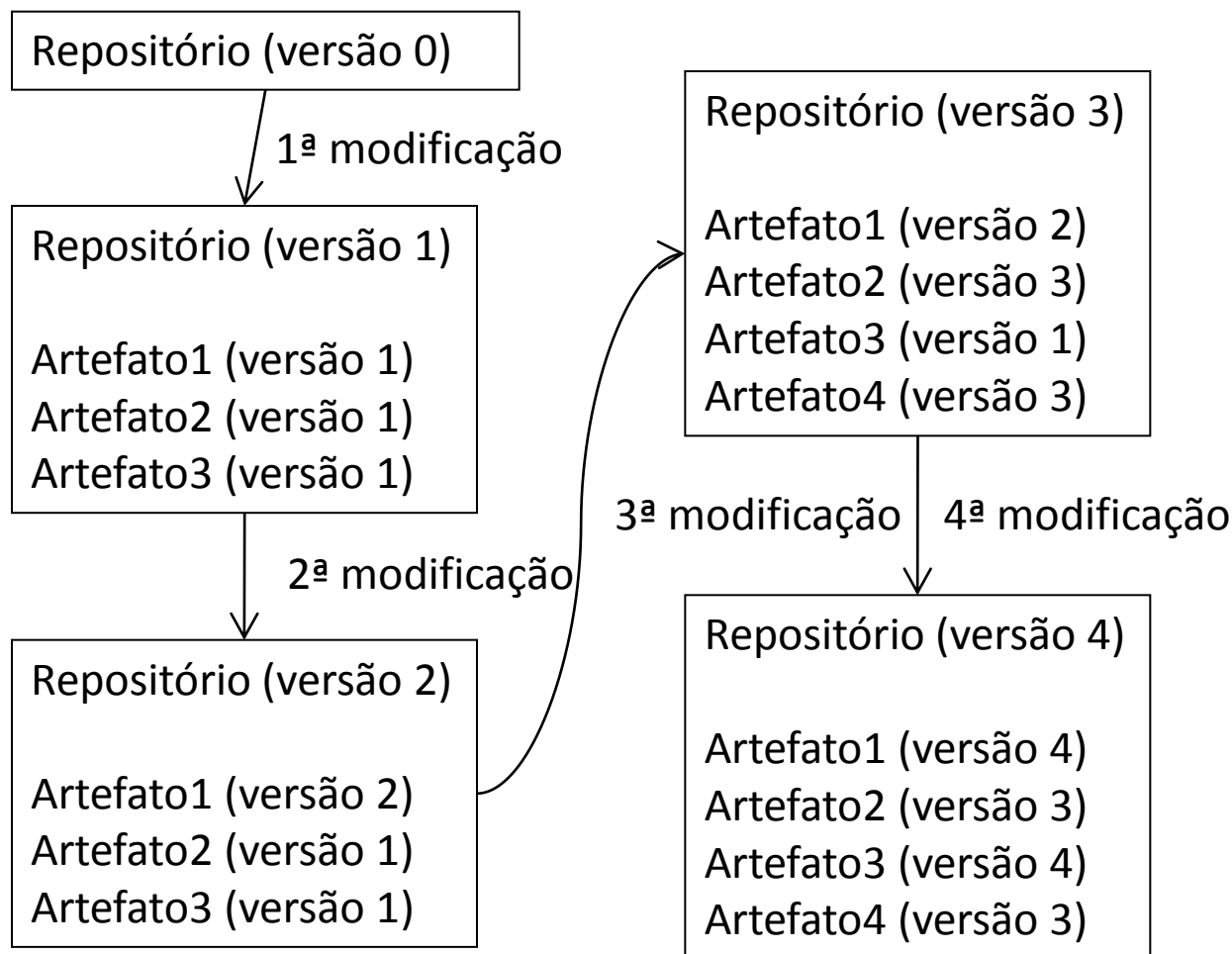
Artefato1
Versão 1
Versão 2
Versão 4

Artefato2
Versão 1
Versão 3

Artefato3
Versão 1
Versão 4

Artefato4
Versão 3

Consulta



Consulta por modificação

1ª modificação

Artefato1 adicionado
Artefato2 adicionado
Artefato3 adicionado

2ª modificação

Artefato1 modificado

3ª modificação

Artefato2 modificado
Artefato4 adicionado

4ª modificação

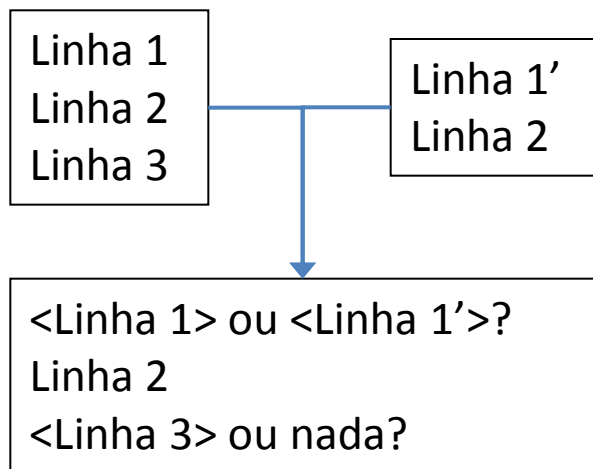
Artefato1 modificado
Artefato3 modificado

Ramos (branches)

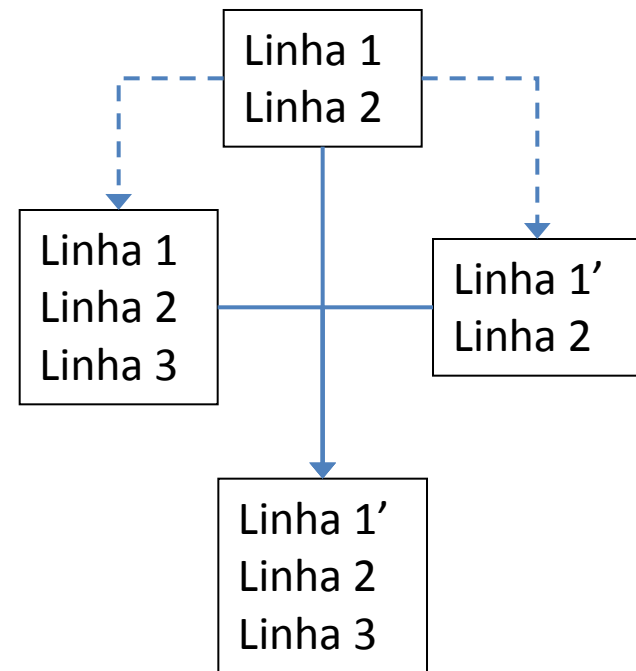
- Versões que não seguem a linha principal de desenvolvimento
- Fornecem isolamento para o processo de desenvolvimento
 - Ramos usualmente são migrados à linha principal de desenvolvimento
 - A migração pode ser complicada no caso de isolamento longo
- Características dos ramos se comparados a espaços 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

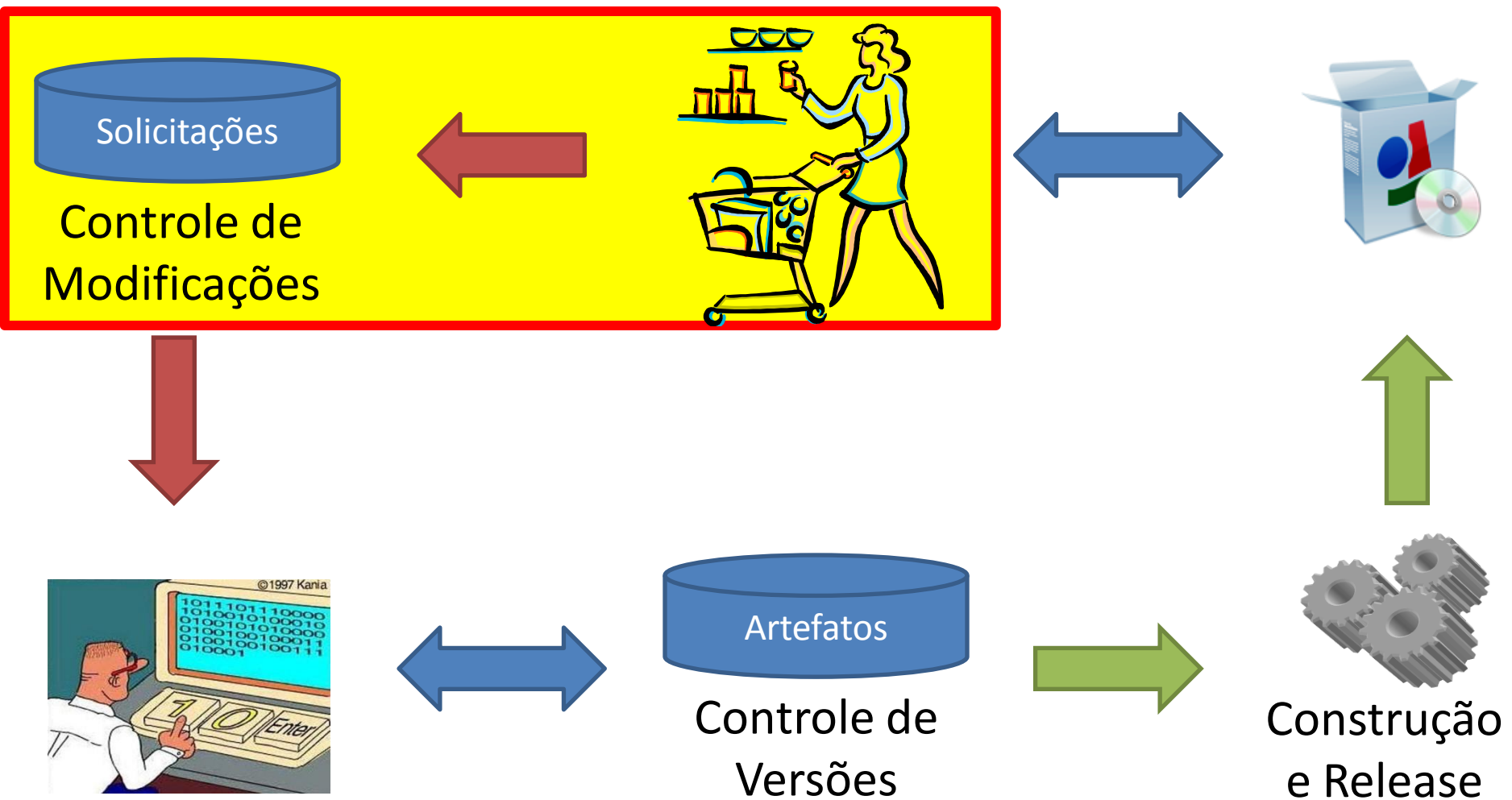


2-way merge



3-way merge

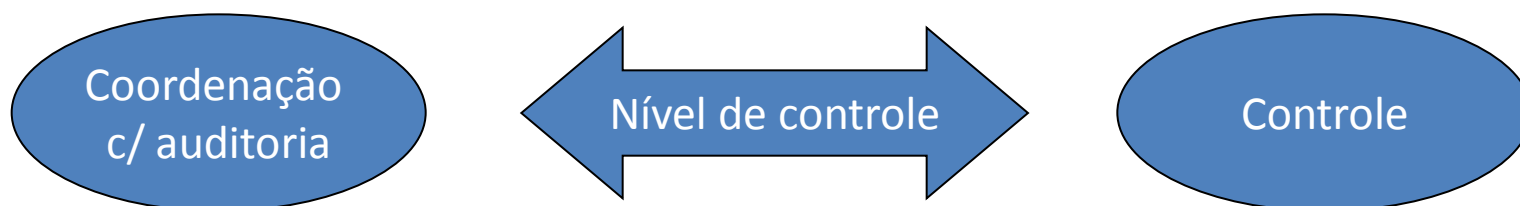
Sistema de Gerência de Configuração



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)



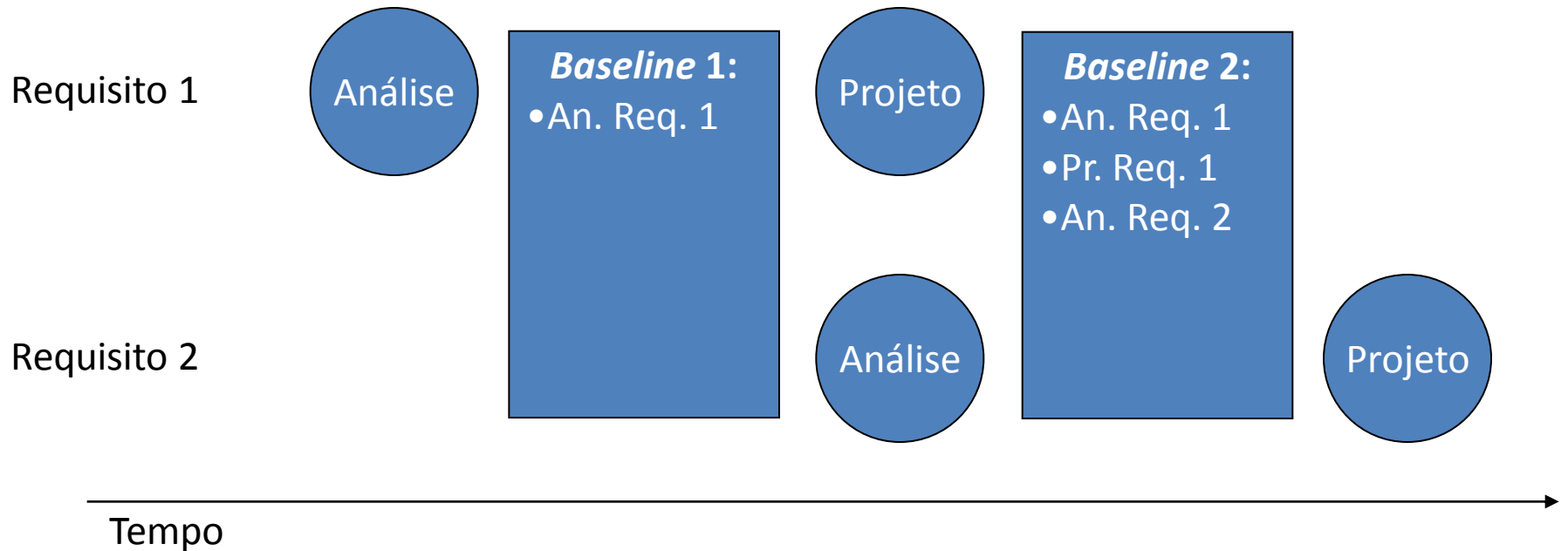
Informal:

- 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.	Análise	Projeto
1	Inform.	-
2	-	-

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:

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

[Leon, 2000] Requisição de modificação

Controle de modificações

- O critério de **classificação** da modificação deve estar explicitado no plano de GC
- A **classificação** visa priorizar modificações mais importantes (críticas, fatais, não fatais, cosméticas)
- A **análise** visa relatar os impactos em custo, cronograma, funcionalidades, etc. da implementação da modificação
- 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

Change Analysis Document

No.: _____

CR No.: _____

Date: _____

System/project: _____ Item to be analyzed: _____

Analyzed by: _____

Implementation alternatives:

Items affected

Item ID	Item description	Version no.	Nature of change

Estimated effort: _____

Impact on schedule: _____

Impact on cost: _____

Recommendation:

[Leon, 2000] Análise de modificação

Controle de modificações

- A **avaliação** utilizará a requisição de modificação e o laudo da análise para tomar a decisão
 - A requisição pode ser aceita, rejeitada ou adiada
- 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 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

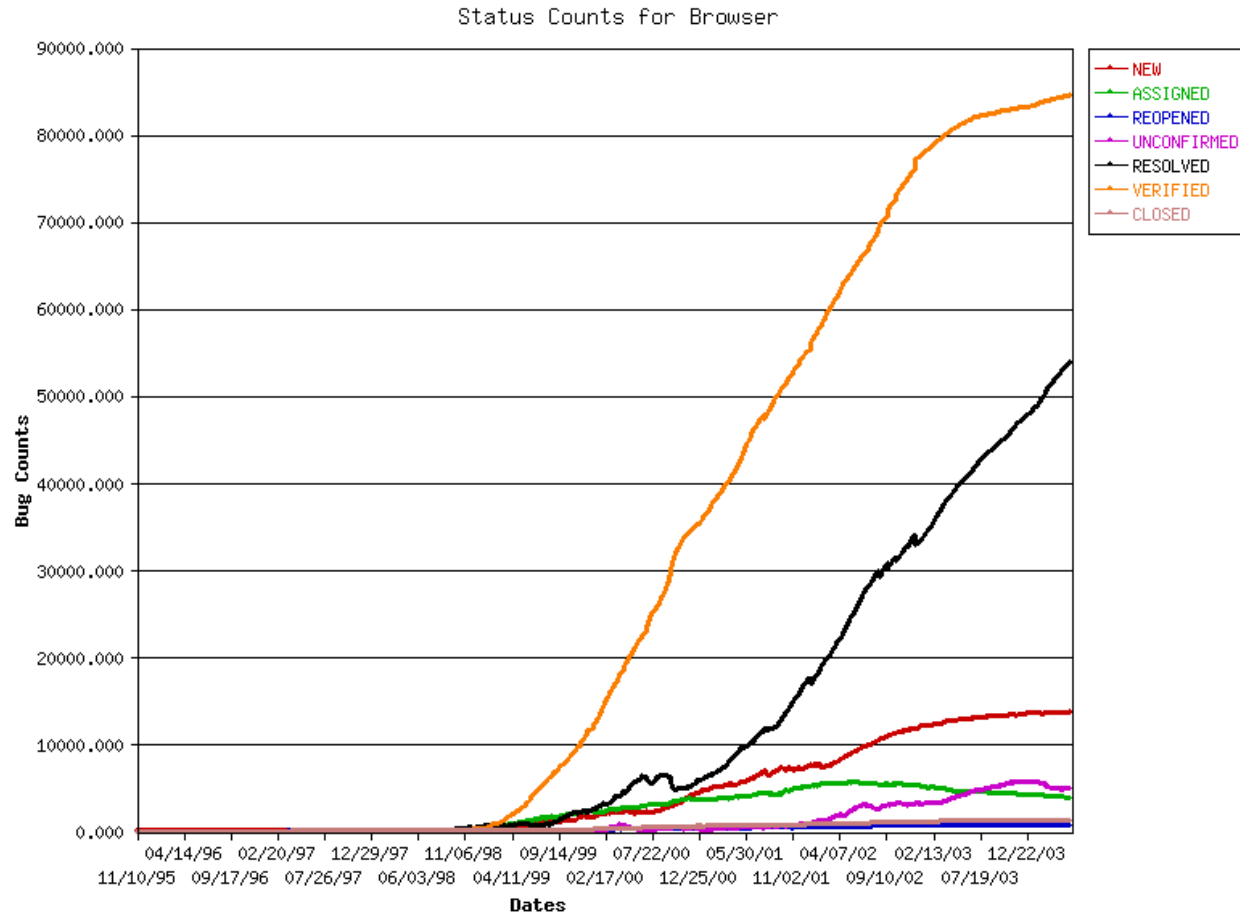
- Caso especial: Defeitos
 - Alguns sistemas tratam defeitos de forma diferente das demais requisições
 - A correção de defeitos é um tratamento sintomático
 - É importante descobrir o real motivo para o acontecimento do defeito para possibilitar a prevenção de defeitos futuros
 - A análise de causa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



Contabilização da situação

- Tarefas
 - Armazenamento das informações geradas
 - Propagação dessas informações aos interessados através de relatórios
- Metáfora de conta bancária para item de configuração
- Permite que métricas sejam utilizadas com o intuito de melhoria do processo e estimativa de custos futuros
- Fornece relatórios gerenciais *ad-hoc*

Contabilização da situação



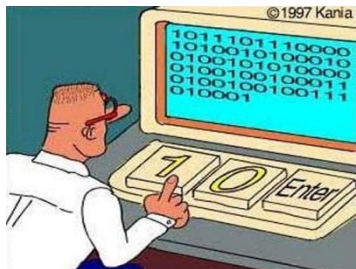
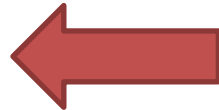
Resultado da consulta sobre séries no Bugzilla

Sistema de Gerência de Configuração



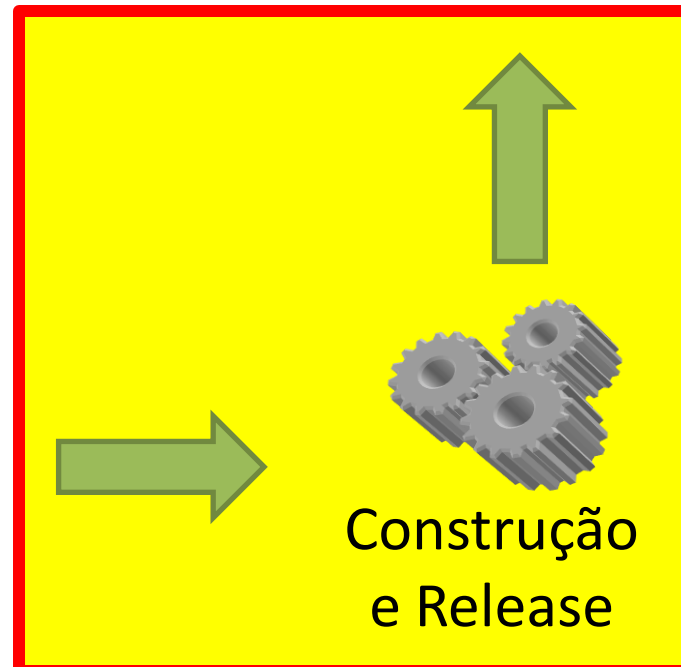
Solicitações

Controle de
Modificações



Artefatos

Controle de
Versões





Item derivado

- Item de configuração que pode ser obtido a partir de outro item de configuração (item fonte)
- Exemplo
 - Os itens de configuração que compõem o código-fonte são itens fonte para o programa executável, que é item derivado
- Estratégias
 - Versionamento do item derivado
 - Documentação do processo de derivação (roteiro, ferramentas, ambiente, etc.)

Construção (*building*)

- Processo de compilação do sistema a partir dos itens fonte para uma configuração alvo
- Utiliza arquivo de comandos que descreve como deve ocorrer a construção
- Exemplo: *makefile*, *build.xml*, *pom.xml*
- Os arquivos de comandos também devem ser considerados itens de configuração

Auditoria da configuração

- Deve ocorrer ao menos antes de uma liberação (*release*)
- Tarefas
 - Verificação funcional, assegurando que a *baseline* cumpre o que foi especificado
 - Verificação física, assegurando que a *baseline* é completa (todos os itens de configuração especificados)
- Auditorias servem para garantir que os procedimentos e padrões foram aplicados

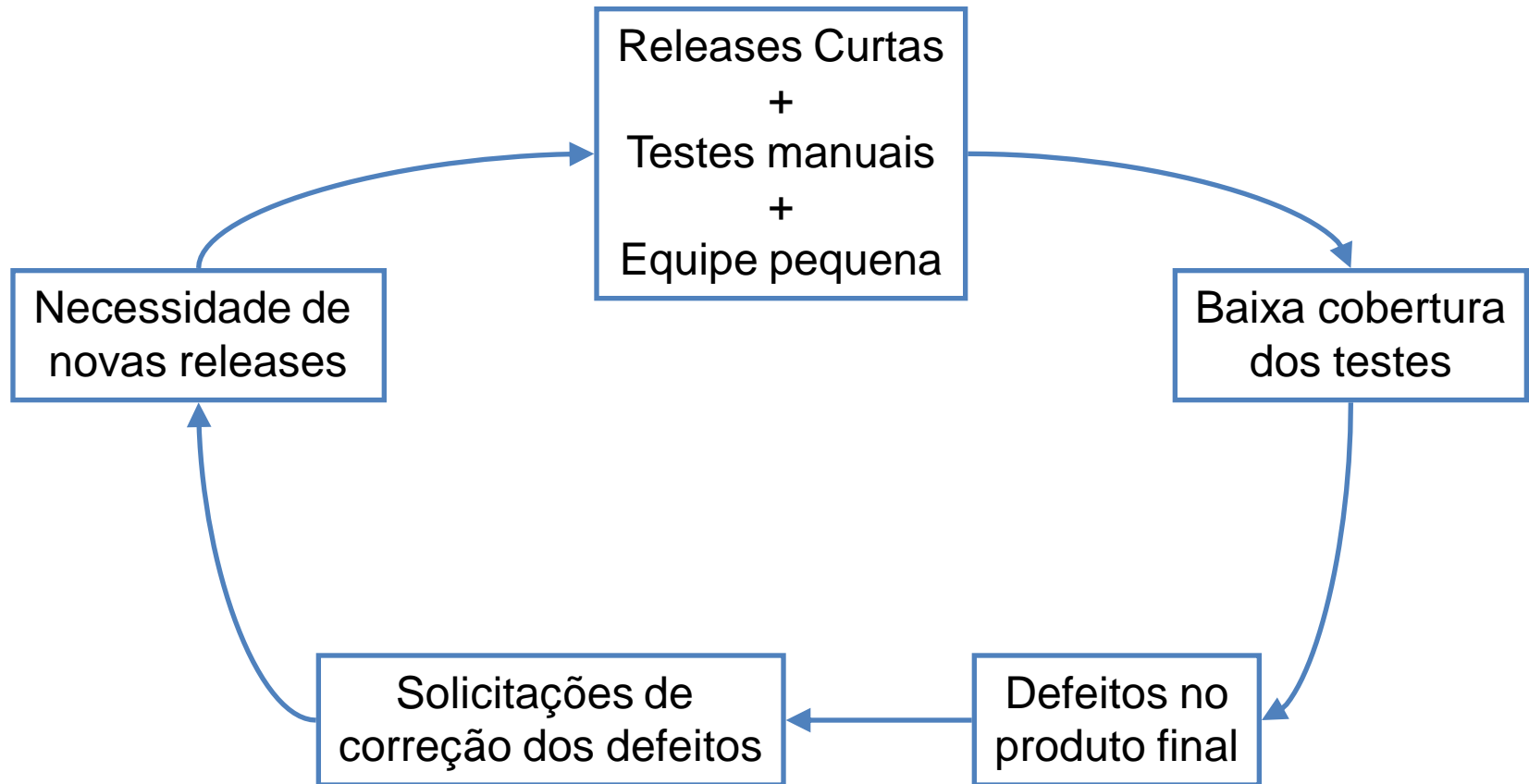
Auditoria da configuração

- A auditoria funcional ocorre através da revisão dos planos, dados, metodologia e resultado dos teste, para verificar se são satisfatórios
- A auditoria física examina a estrutura de todos os itens de configuração que compõem a *baseline*
- A auditoria física é efetuada após a auditoria funcional
- Podem ocorrer auditorias no próprio sistema de GC pelos mantenedores do plano de GC, para verificar se as políticas e procedimentos estão sendo cumpridos

Gerenciamento de *releases*

- Descrição de como construir, liberar e entregar o sistema
 - Linguagem natural (conhecimento)
 - Linguagem computacional (automação)
 - Manter os descritores e documentos sob gerência de configuração!
- Definição das situações onde o processo pode ser temporariamente desviado
- Cuidado: *Releases* muito curtas podem levar a círculo-vicioso de defeitos...

Gerenciamento de *releases*





Introdução à Gerência de Configuração

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br