

Manutenção de Software e Gerência de Configuração

Leonardo Gresta Paulino Murta
murta@cos.ufrj.br



Cenário atual

90% do custo total [Erlikh 2000]

60% do esforço total [Pressman 1997]

70 bilhões de dólares anuais
nos EUA [Edelstein 1993]

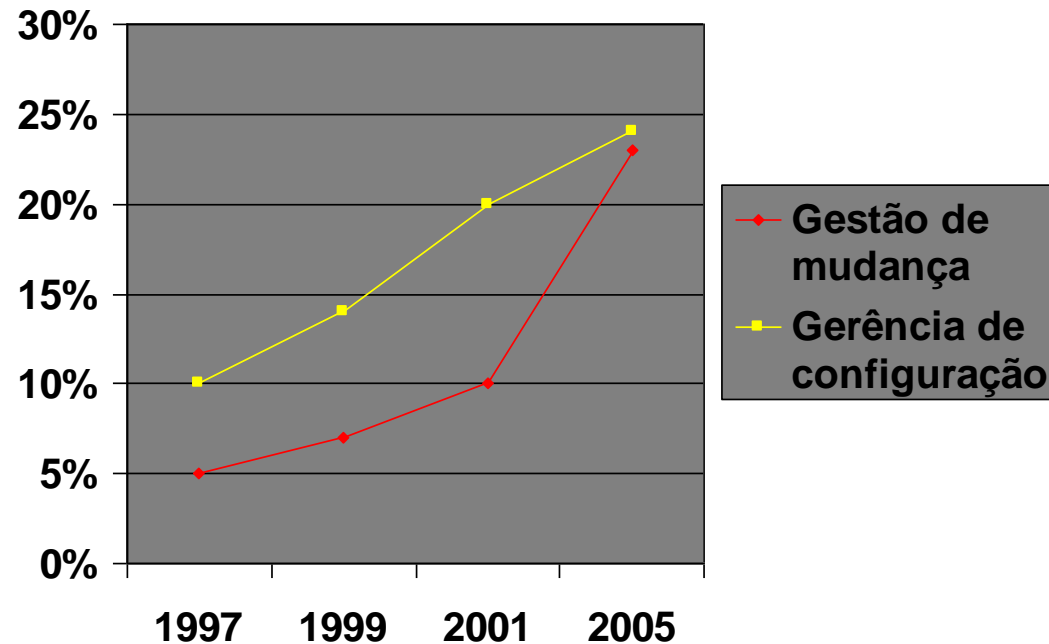
250 bilhões de linhas de código em
manutenção [Sommerville 2000]

Cenário atual

- Somente 21% do esforço de manutenção é despendido em correções de erros (Bennett and Rajlich 2000)
- Manutenção é necessária mesmo em sistema com qualidade
 - Adaptação para novas regulamentações legais
 - Novos requisitos
 - Torná-lo mais simples de compreender e manter

Cenário atual

- Manutenção no Brasil [MCT 2006]
 - 2500 empresas entrevistadas
 - 95% de confiabilidade
 - Erro máximo de 5,5%



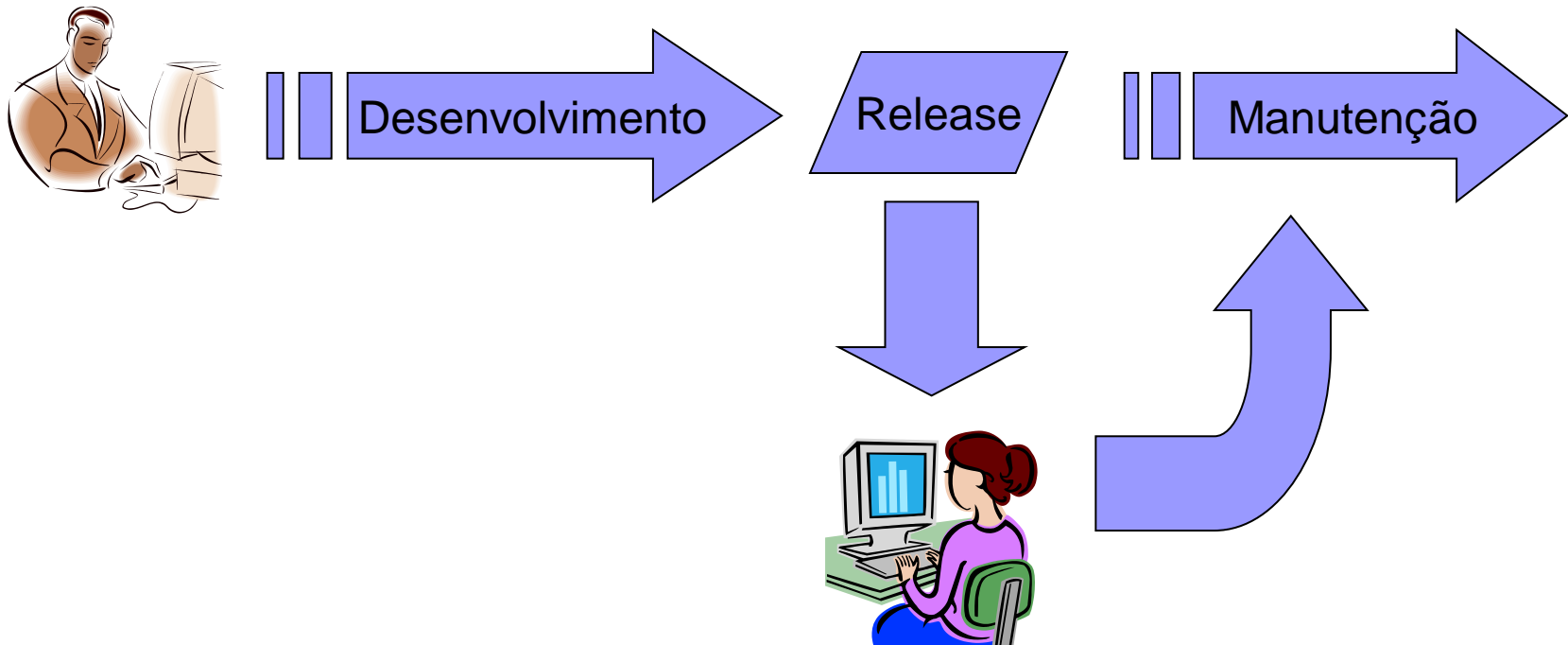


O que é a manutenção?

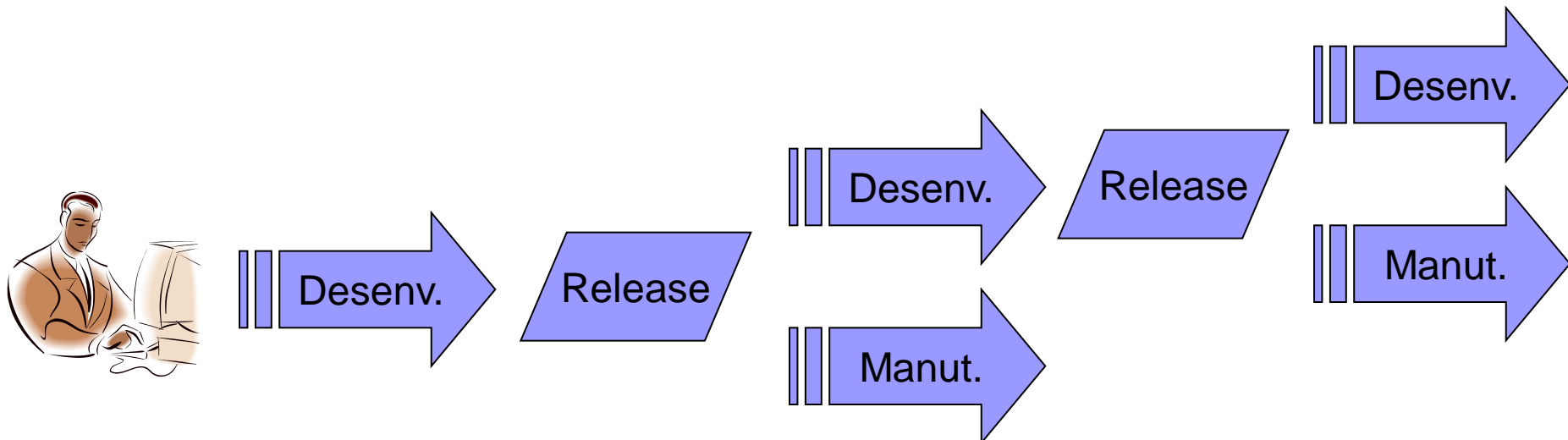
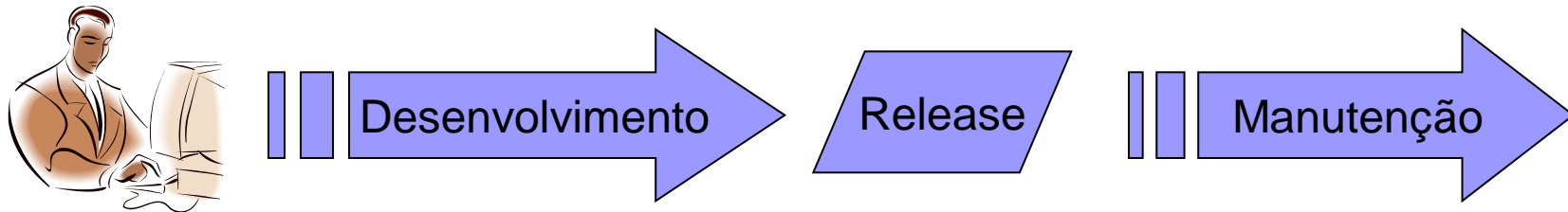
O processo de **modificar um sistema de software** ou componente, **depois da entrega**, para **corrigir falhas**, **melhorar desempenho ou outros atributos**, ou **adaptar a mudanças no ambiente**.

IEEE Std 620.12 1990

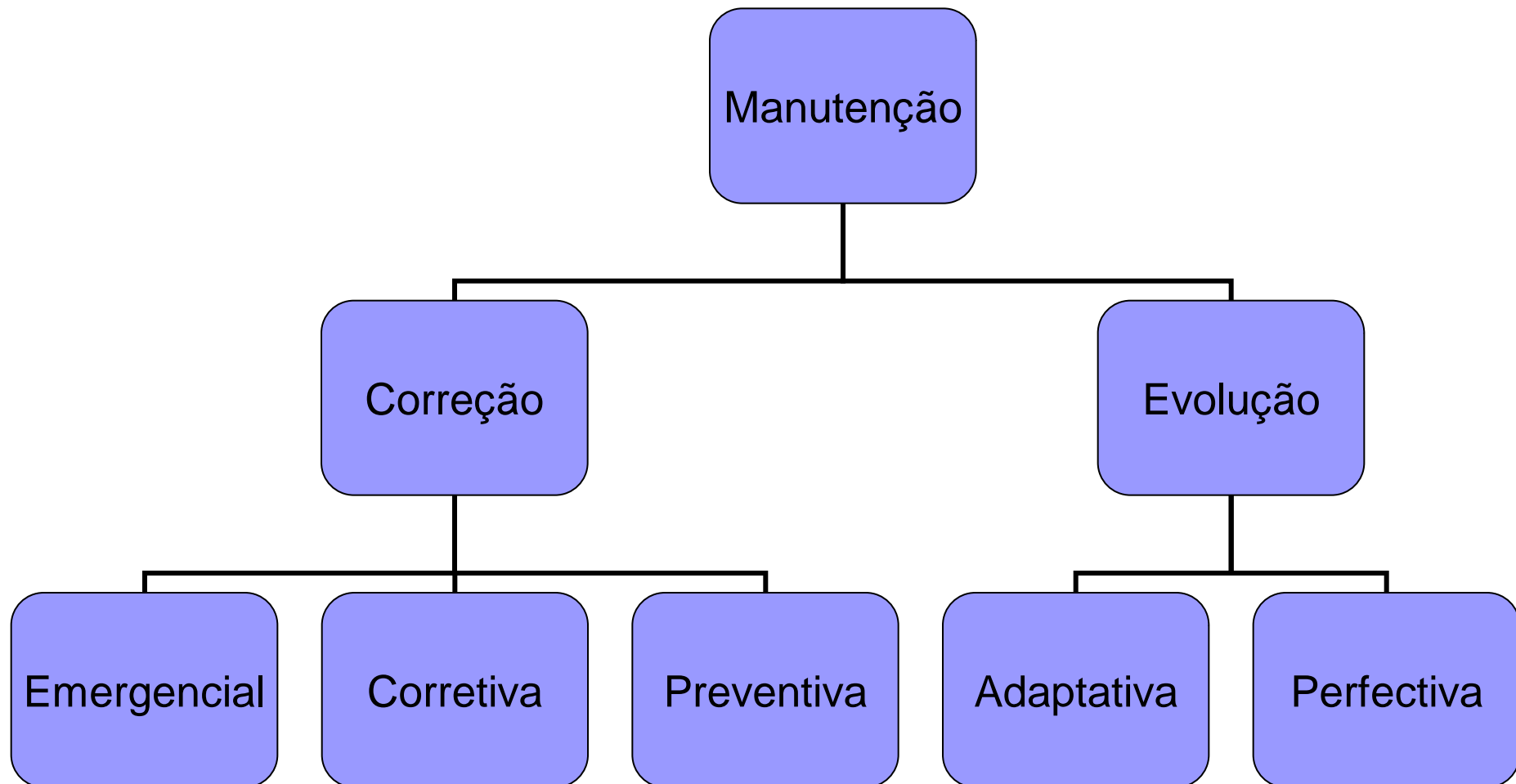
Quando inicia a manutenção?



Quando inicia a manutenção?



Quais são os tipos de manutenção?





Quais são os tipos de manutenção?

■ Manutenção corretiva

- ☐ Reativa
- ☐ Corrige problemas reportados
- ☐ Faz o software voltar a atender aos requisitos

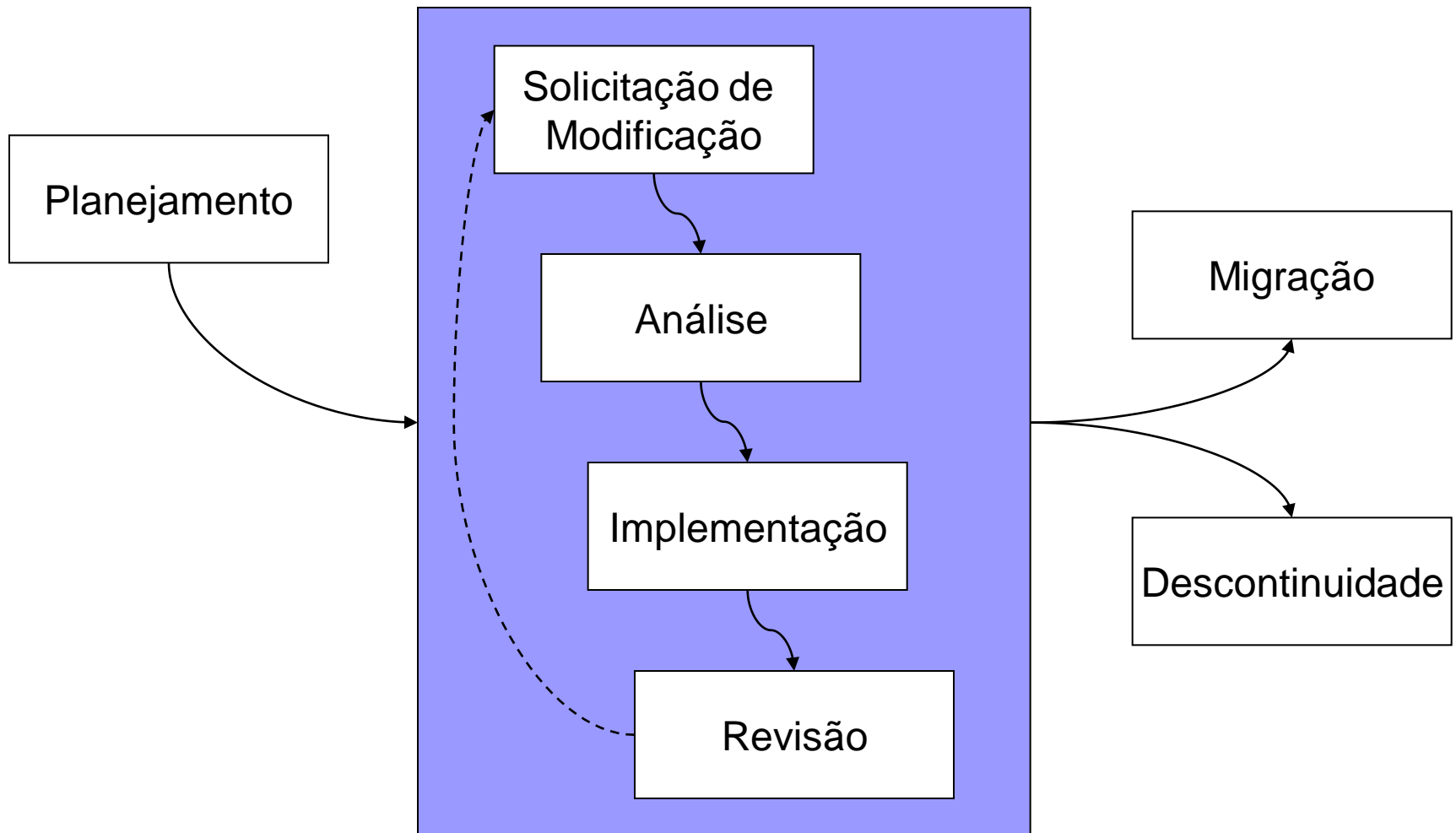
■ Manutenção emergencial

- ☐ Não programada
- ☐ Mantém temporariamente o sistema funcionando
- ☐ Necessita uma manutenção corretiva posterior

Quais são os tipos de manutenção?

- Manutenção preventiva
 - ☐ Pró-ativa
 - ☐ Corrige problemas latentes
- Manutenção adaptativa
 - ☐ Mantém o software usável após mudanças no ambiente
- Manutenção perfectiva
 - ☐ Provê melhorias para o usuário
 - ☐ Melhora atributos de qualidade do software

Processo de manutenção



Contratos de manutenção

■ Tipo 1

- ☐ Um único contrato para desenvolvimento e manutenção
- ☐ Valor fixo, disponível para todos os tipos de manutenção

■ Tipo 2

- ☐ Contrato separado para manutenção
- ☐ Período de manutenções corretivas predefinido
- ☐ Cada manutenção preventiva, adaptativa ou perfectiva contratada separadamente

Apoio à manutenção

- Boas práticas
- Processos de apoio
 - ☐ Documentação
 - ☐ **Gerência de configuração**
 - ☐ Garantia da qualidade
 - ☐ Verificação
 - ☐ Validação



Boas práticas para manutenção (de código)

- Legibilidade
- Estruturação
- Redução da complexidade
- Comentários precisos
- Indentação e espaçamento
- Evitar o uso de armadilhas clássicas das linguagens
 - Ex.: Goto, herança múltipla, etc.
- Usar técnicas que ajudam a rastrear erros
 - Ex.: Controle de exceções

Boas práticas para manutenção (de código)

■ Rastreabilidade

- ☐ Código para requisitos, análise e projeto
- ☐ Código para solicitações de modificação

■ Padronização

■ Inspeções

■ Testes

■ Atualização da documentação

- ☐ Ex.: Modelos

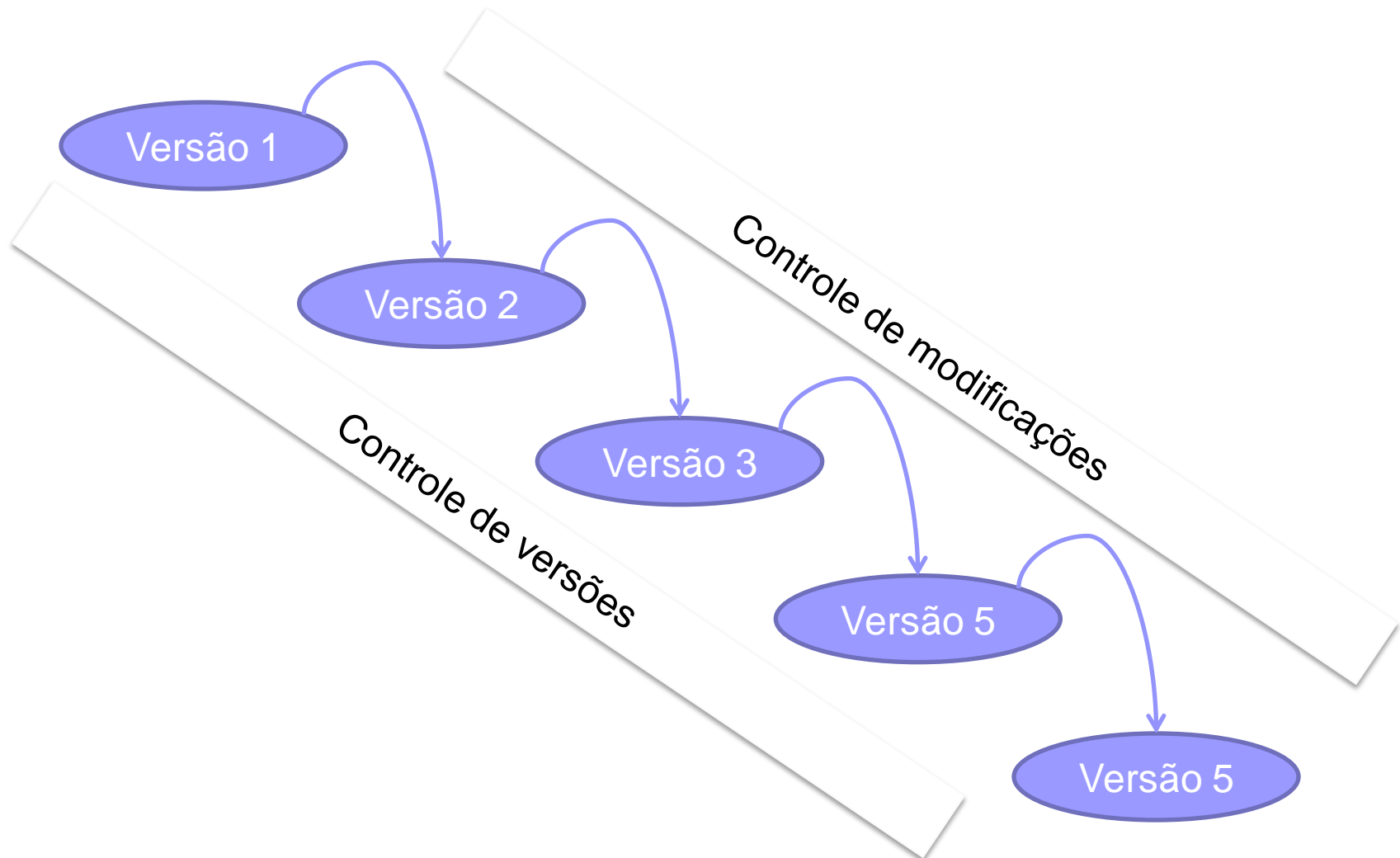
Gerência de configuração

Gerência de configuração de software é uma **disciplina** para o **controle da evolução** de sistemas de software

Susan Dart

- Sistemas de gerência de configuração
 - ☐ Controle de versões
 - ☐ Controle de modificações
 - ☐ Gerenciamento de construção e release

Gerência de configuração

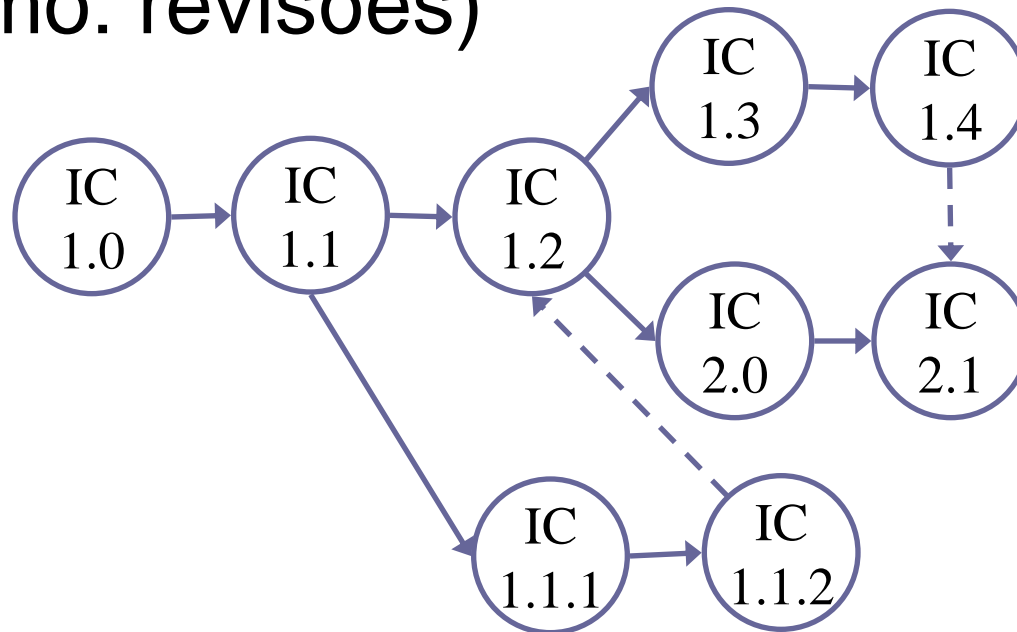


Item de configuração

- Agregação de hardware e/ou software que será passível de gerência de configuração e tratado como um elemento único
- Tipos de ICs
 - ☐ Produtos de trabalho do projeto
 - ☐ Produtos de trabalho de processos
- Exemplos: plano de GC, requisitos, modelos, código-fonte, etc.

Versão

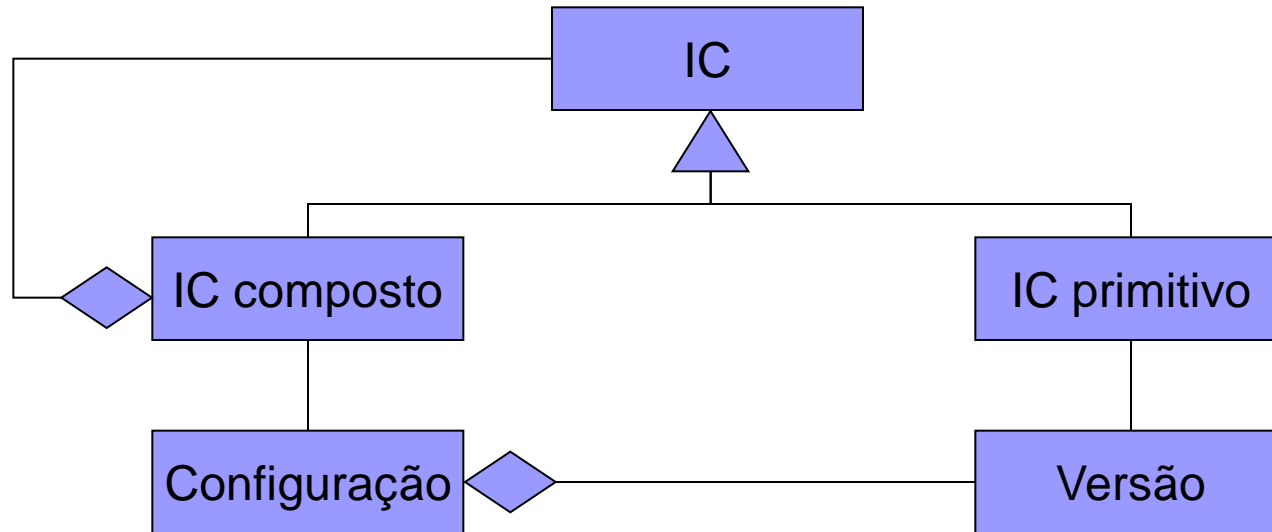
- Instâncias de um mesmo item de configuração que diferem entre si em algo (sinônimo: revisões)



Configuração

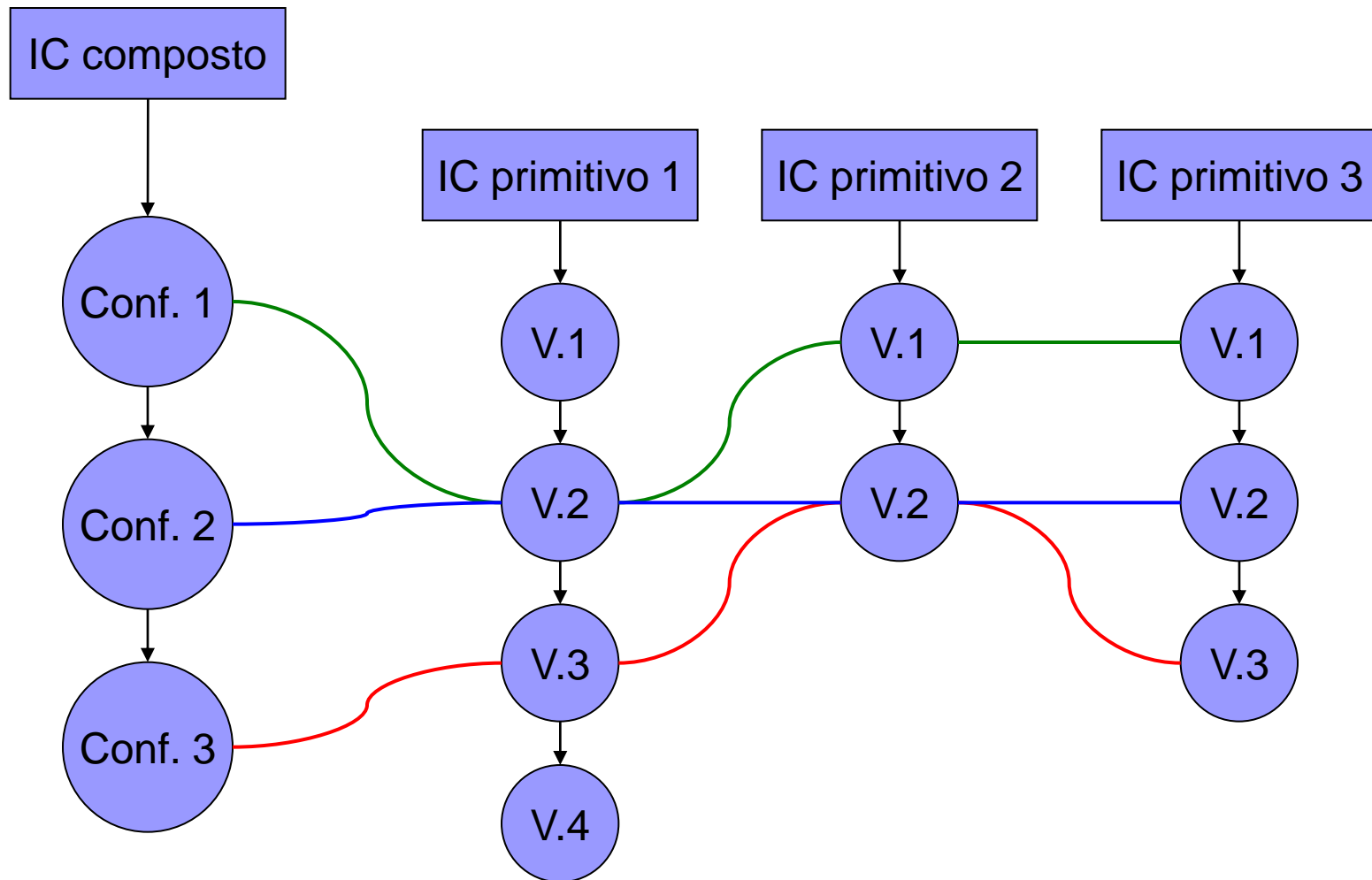
- Um conjunto de versões de Itens de Configuração (IC), onde existe somente uma versão selecionada para cada IC do conjunto
- Uma configuração pode ser vista como um IC composto de outros ICs
- Exemplos
 - ☐ Configuração do sistema
 - ☐ Configuração do processo
 - ☐ Configuração do módulo X
 - ☐ Configuração dos requisitos do sistema
 - ☐ Configuração do código fonte

Configuração x versão

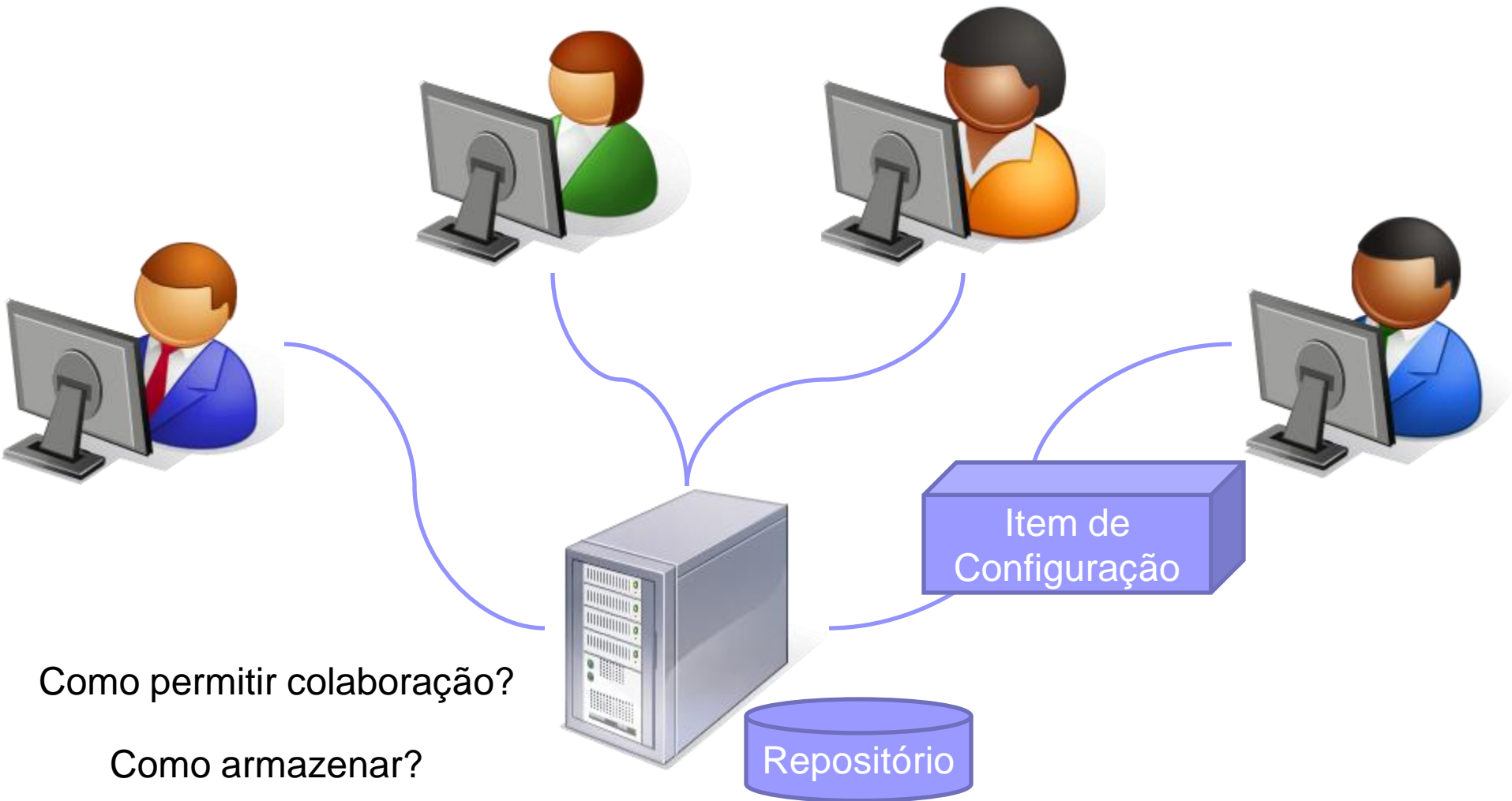


- Genericamente
 - O sistema S é composto pelos arquivos X, Y e Z
- Concretamente
 - A configuração 5 do sistema S é composta pela versão 2 do arquivo X, versão 4 do arquivo Y e versão 6 do arquivo Z

Configuração x versão



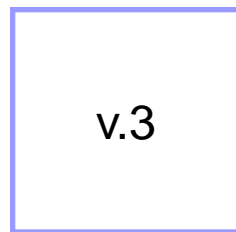
Controle de versões



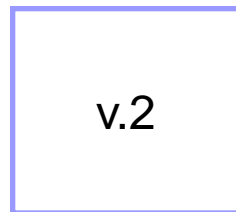
Repositórios

- Local onde os ICs são armazenados
 - ☐ Armazena o histórico do projeto
 - ☐ Controle na entrada e saída de ICs
 - ☐ Poucos por projeto (normalmente, somente um)
- Utiliza diferentes mecanismos de armazenamento
 - ☐ Versionamento completo
 - ☐ Versionamento de diferenças (delta)
- Utiliza diferentes mecanismos de controle de concorrência
 - ☐ Pessimista
 - ☐ Otimista
 - ☐ Misto

Mecanismos de armazenamento

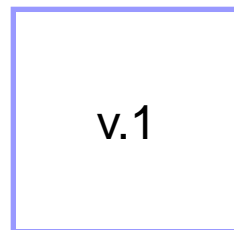


Completo

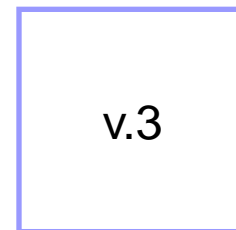


delta 2→3

delta 1→2



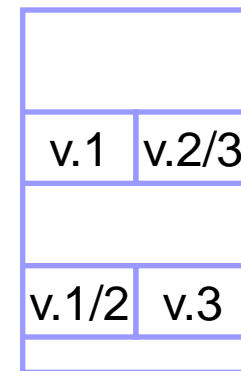
Forward



delta 3→2

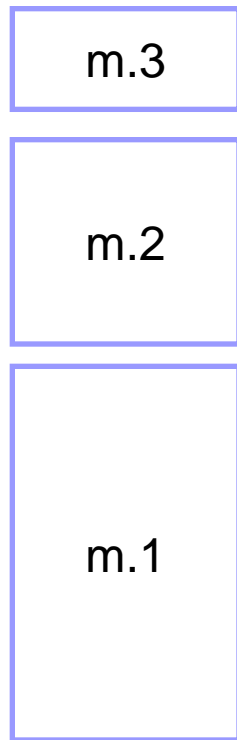
delta 2→1

Reverse

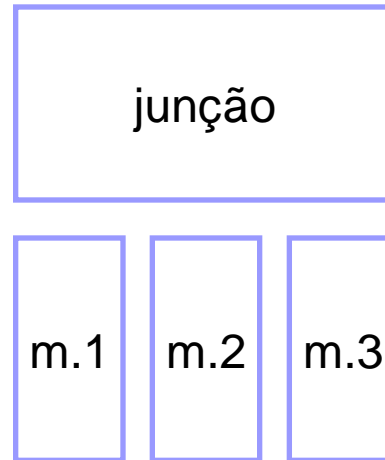


In-line

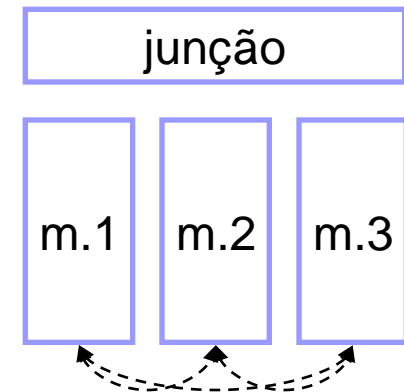
Controle de concorrência



Pessimista



Otimista



Misto

Espaço de trabalho

- Local onde o usuário pode fazer seu trabalho de forma isolada
 - Armazena um momento específico do projeto
 - Controle sobre quando sincronizar com o repositório
 - Muitos por projeto (normalmente, um ou mais por desenvolvedor)
 - Sinônimo: caixa de areia (*sandbox*)

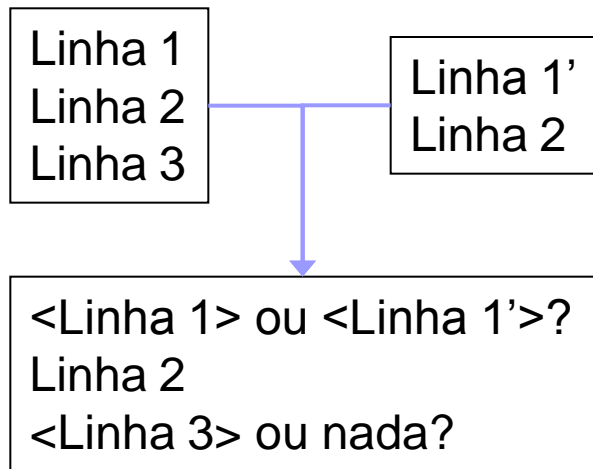
Ramos (*branches*)

- 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)

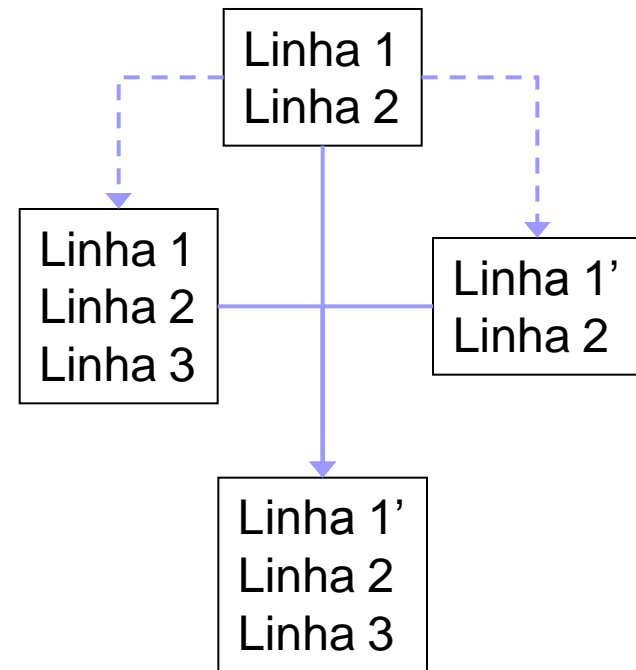
Junção

■ Processo de migração de

- Espaços de trabalho
- Ramos



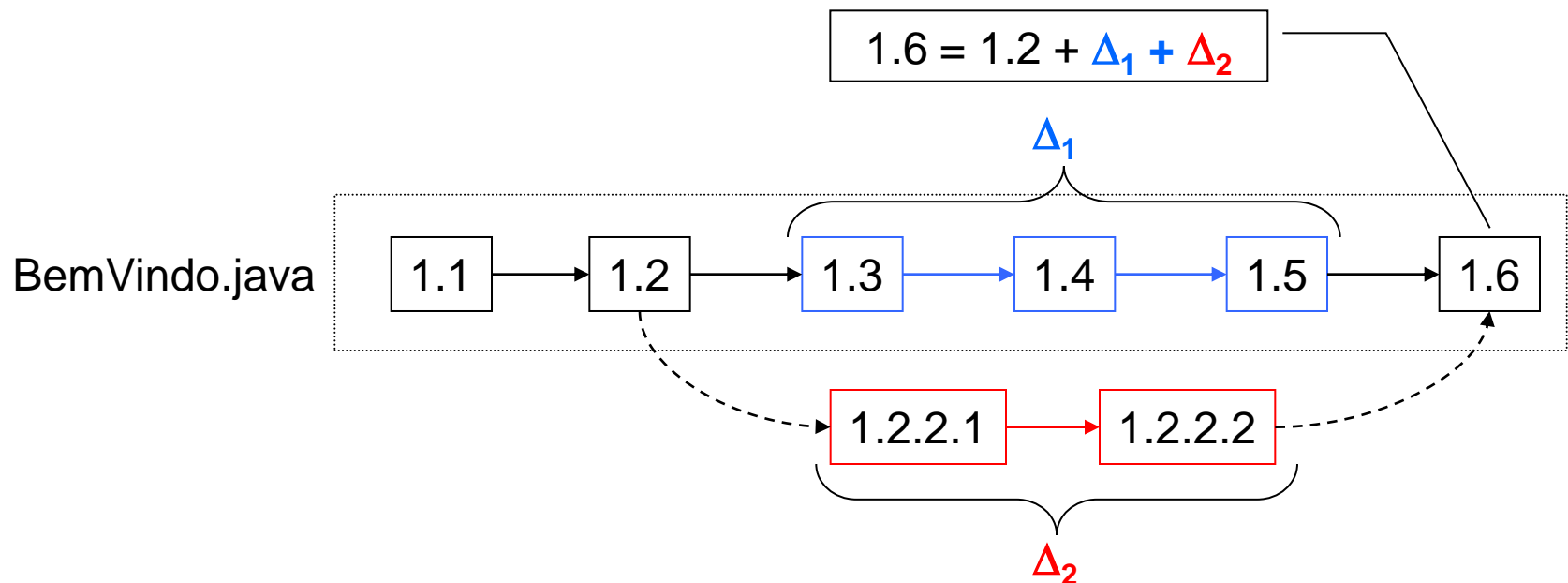
2-way merge



3-way merge

Exemplo de junção

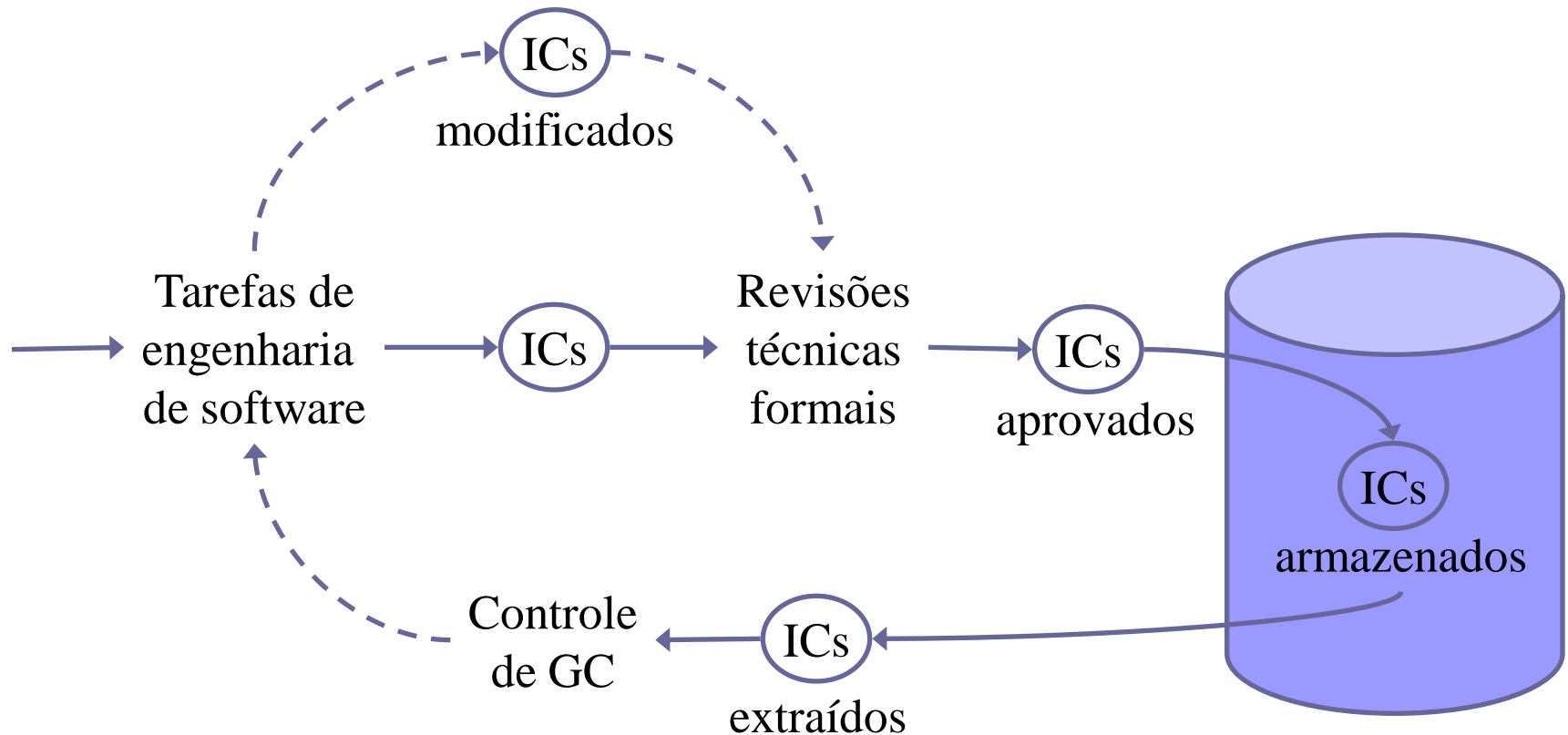
- A junção é efetuada para cada artefato do ramo
- São levadas em consideração todas as modificações desde o ancestral em comum



Baseline

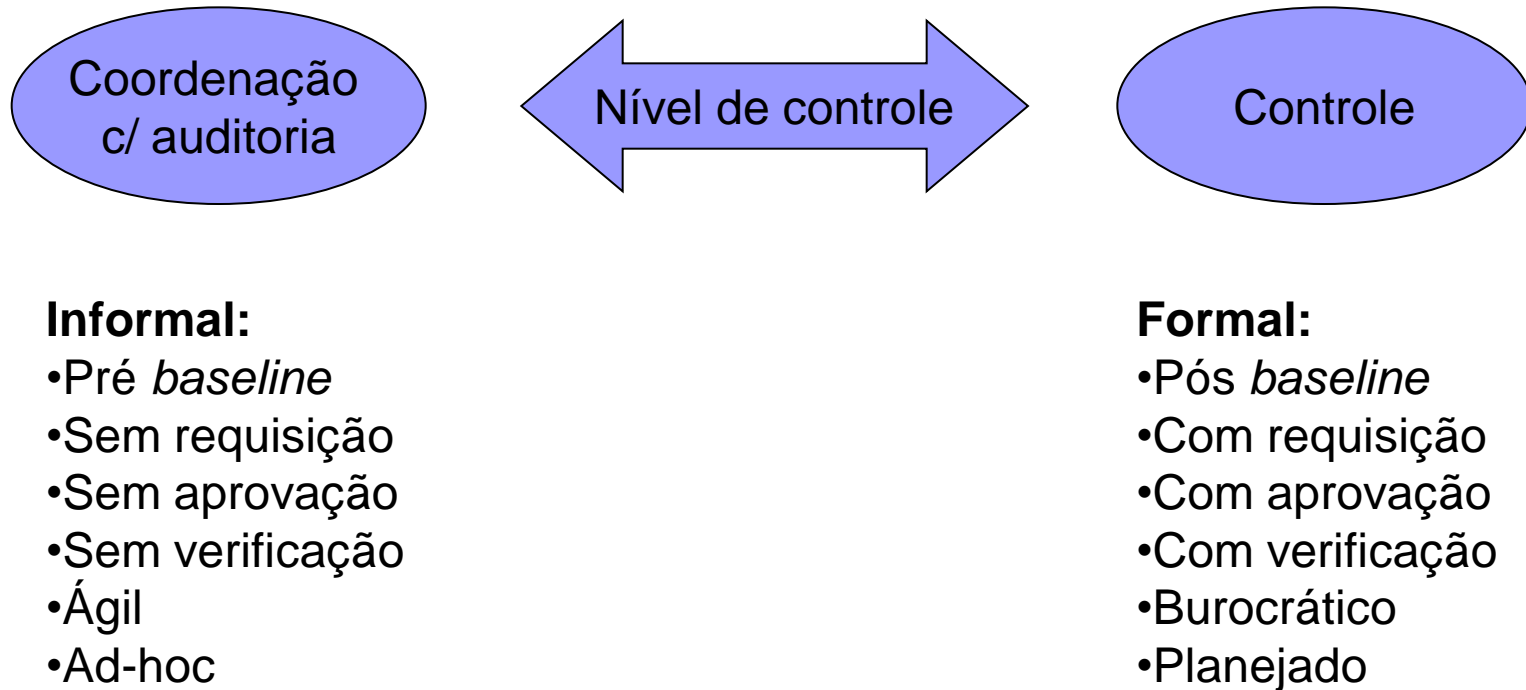
- Configuração revisada e aprovada que serve como base para uma próxima etapa de desenvolvimento e que somente pode ser modificada via processo formal de GCS
- São estabelecidas ao final de cada fase de desenvolvimento
 - ☐ Análise (*functional*)
 - ☐ Projeto (*allocated*)
 - ☐ Implementação (*product*)
- Momento de criar: balanceamento entre controle e burocracia

Baseline

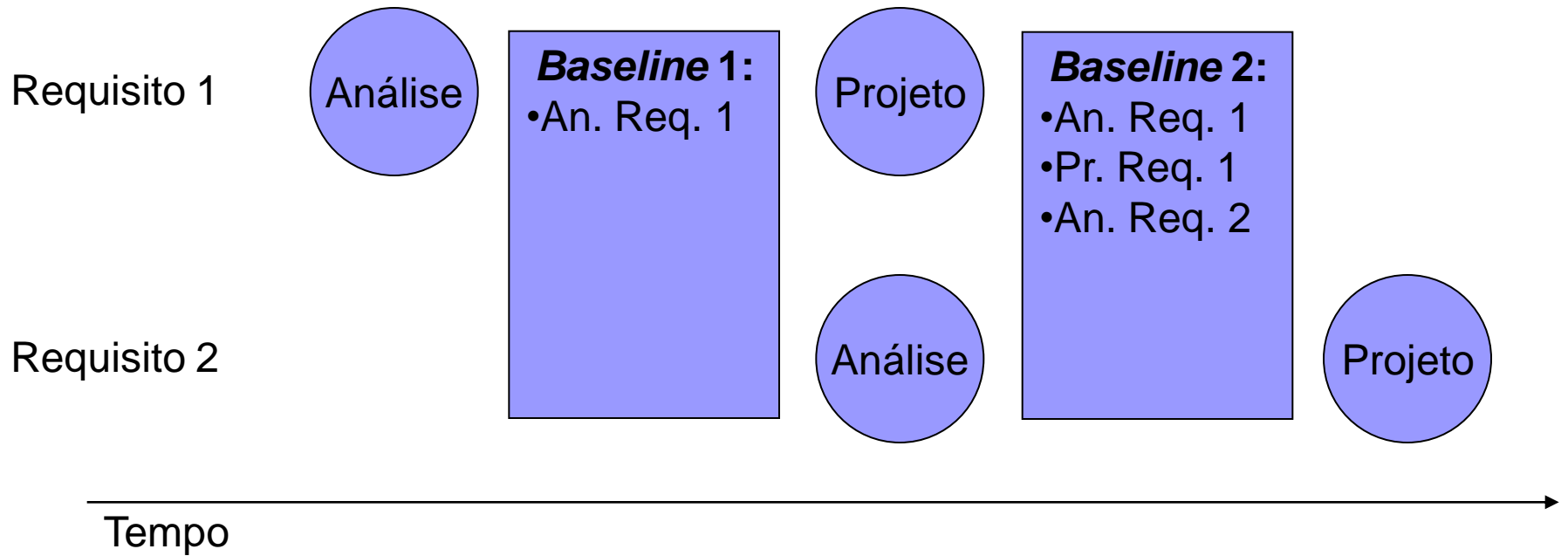


[Pressman, 1997] Processo de atualização de *baseline*

Baseline (níveis de controle)



Baseline (níveis de controle)



Req.
1
2

Análise	Projeto
Inform.	-
-	-

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:

[Leon, 2000] Solicitação de modificação

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

Controle de modificações

- O critério de **classificação** da modificação deve ser explicitado no planejamento
- A **classificação** visa priorizar modificações mais importantes
 - ☐ Críticas
 - ☐ Fatais
 - ☐ Não fatais
 - ☐ Cosméticas

Controle de modificações

- A **análise** visa relatar os impactos da implementação da modificação
 - ☐ Custo
 - ☐ Cronograma
 - ☐ Funcionalidades
- 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 impacto

Controle de modificações

- A **avaliação** utilizará a solicitação de modificação e o laudo da análise para tomar a decisão
 - ☐ Aceitar
 - ☐ Rejeitar
 - ☐ Adiar
- Essa decisão é tomada pelo Comitê de Controle da Configuração (CCC)
 - ☐ Líder do projeto e de gerência de configuração
 - ☐ Representantes da garantia de qualidade, marketing e cliente

Controle de modificações

- 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 pelo CCC 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)



Ferramentas de controle de versões

■ Livre

- ☐ Aegis
- ☐ CVS
- ☐ GNU Arch
- ☐ Subversion

■ Comercial

- ☐ BitKeeper (BitMover)
- ☐ ClearCase (IBM Rational)
- ☐ Perforce
- ☐ PVCS (Serena)
- ☐ StarTeam (Borland)
- ☐ Synergy/CM (Telelogic)
- ☐ Visual Studio Team Foundation (Microsoft)



Ferramentas de controle de modificações

■ Livre

- ☐ Bugzilla
- ☐ Mantis
- ☐ Trac

■ Comercial

- ☐ ClearQuest (IBM Rational)
- ☐ JIRA
- ☐ StarTeam (Borland)
- ☐ Synergy/Change (Telelogic)
- ☐ TeamTrack (Serena)
- ☐ Visual Studio Team Foundation (Microsoft)

Quer saber mais?

■ Fóruns

- ☐ Journal of Software Maintenance and Evolution: Research and Practice (JSME)
- ☐ International Conference on Software Maintenance (ICSM)
- ☐ International Workshop on Principles of Software Evolution (IWPSE)
- ☐ Workshop de Manutenção de Software Moderna (WMSWM)

■ WMSWM 2008

- ☐ 3 de junho de 2008 – Florianópolis/SC



Principais Referências Bibliográficas

- Alexis Leon, “A Guide to Software Configuration Management”, Artech House Publishers, 2000.
- Anne Hass, “Configuration Management Principles and Practices”, Boston, MA, Pearson Education, Inc.
- Bennett, K. H. and V. T. Rajlich (2000). Software maintenance and evolution: a roadmap. International Conference on Software Engineering, The Future of Software Engineering, Limerick, Ireland, ACM Press.
- Brian A. White, “Software Configuration Management Strategies and Rational ClearCase – A Practical Introduction”, Addison-Wesley, 2000.

Principais Referências Bibliográficas

- Edelstein, D. (1993). "Report on the IEEE 1219-1993- Standard for Software Maintenance." ACM SIGSOFT Software Engineering Notes 18(4): 94-95.
- Erlikh, L. (2000). "Leveraging Legacy System Dollars for E-Business." ITProfessional 2(3): 17-23.
- IEEE Std 14764-2006, "Software Engineering – Software Life Cycle Processes – Maintenance".
- MCT (2006). Qualidade e Produtividade no Setor de Software Brasileiro. Brasília, DF, Ministério de Ciência e Tecnologia, Secretaria de Política de Informática.
- Pressman, R. S. (1997). Software Engineering: A Practitioner's Approach, McGraw-Hill.
- Sommerville, I. (2000). Software Engineering, Addison-Wesley.

Manutenção de Software e Gerência de Configuração

Leonardo Gresta Paulino Murta
murta@cos.ufrj.br