



Gerência de Configuração: Ramificação e Integração

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br

Estratégias de Ramificação

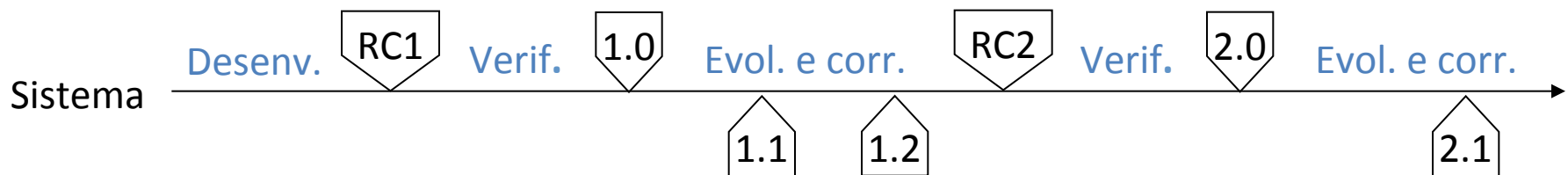
- Existem diferentes estratégias de criação de ramos, que podem ser combinadas
- A seleção correta de estratégias possibilita
 - Desenvolvimento/manutenção em paralelo de liberações
 - Isolamento entre desenvolvedores e equipes
 - Identificação das requisições de modificação que pertencem a cada *baseline*
 - Adição ou remoção de uma requisição de modificação em uma *baseline*
 - Customização do sistema para diferentes clientes ou plataformas
 - Manutenção da integridade de *baselines*

Estratégias de Ramificação

- Estratégias de manutenção
 - Caótica
 - Em série
 - Em cascata
- Estratégias de organização
 - Por desenvolvedores
 - Por subprojetos (componentes da arquitetura)
 - Por requisições
 - Por customizações
- Estratégias de verificação
 - Contínua
 - Periódica
 - Pré-liberação

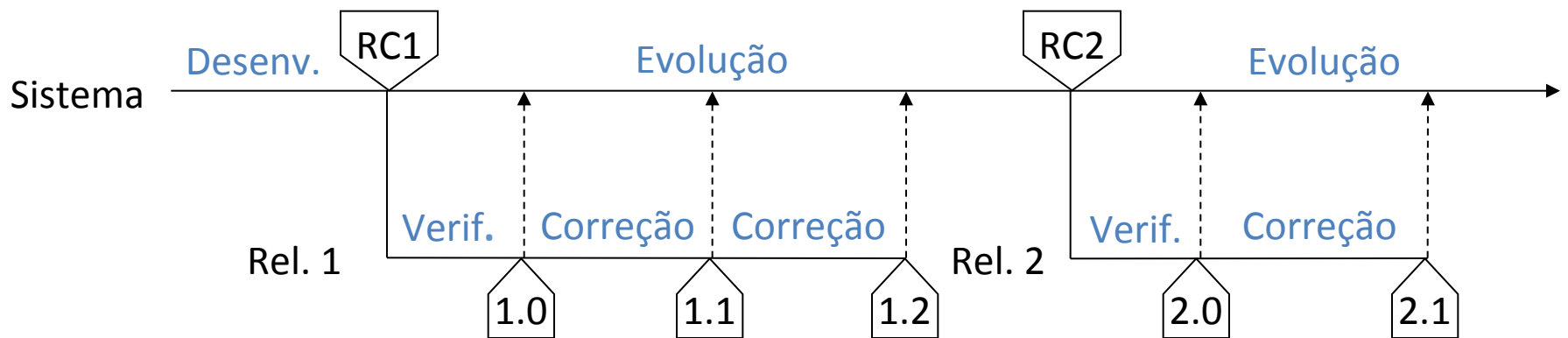
Estratégias de Ramificação

- Manutenção Caótica
 - Ramo principal: evolução e correções
- Usualmente praticado por organizações imaturas
- Impossibilidade de separar o que é manutenção corretiva de evolutiva
- Usuários de uma dada versão, ao reclamarem de um erro, recebem novas funcionalidades junto com a correção do erro



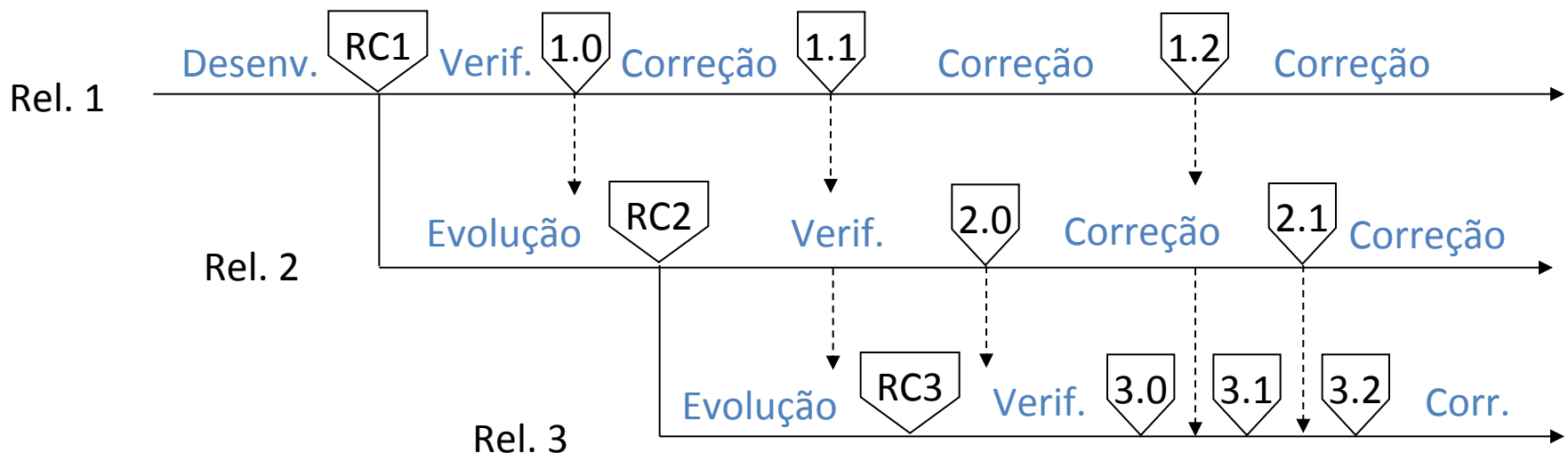
Estratégias de Ramificação

- Manutenção em série
 - Ramo principal: evolução
 - Ramos auxiliares: correções
- Foco
 - Desenvolvimento *in-house*
 - Cliente único (e.g.: aplicações *Web*)
- Dificuldade de manutenção de várias liberações em paralelo



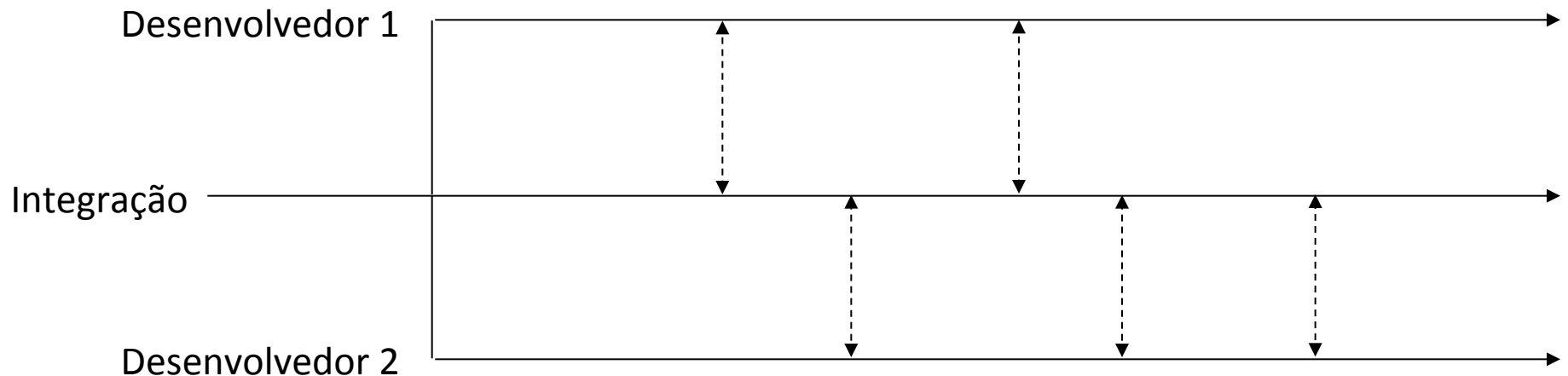
Estratégias de Ramificação

- Manutenção em cascata
 - Ramo principal: correções
 - Ramo auxiliar: evolução
- Foco
 - Manutenção em paralelo de múltiplas *releases*
- Dificuldade de coordenar check-outs evolutivos em andamento



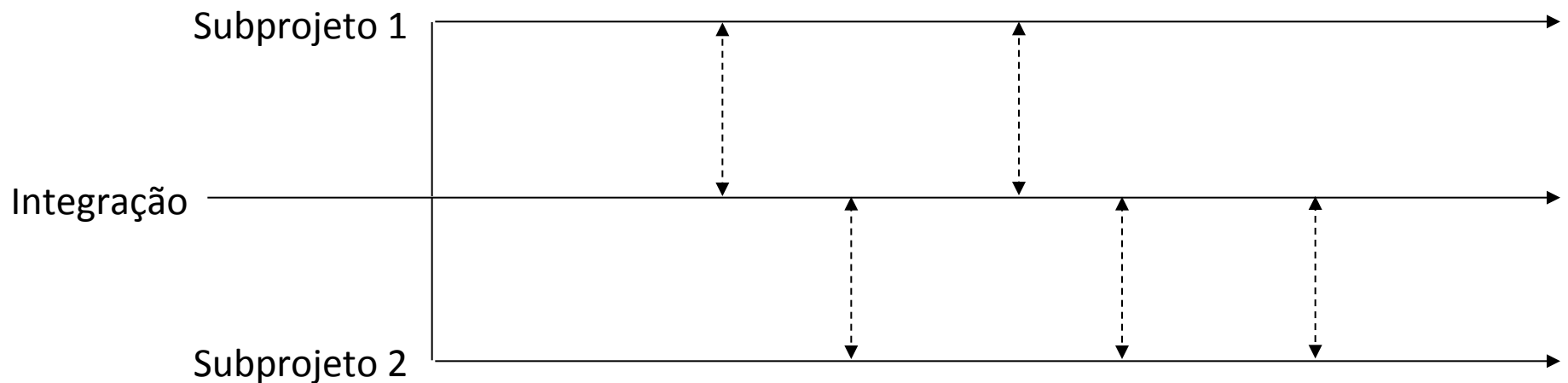
Estratégias de Ramificação

- Organização por desenvolvedores
 - Ramo principal: integração
 - Ramos auxiliares: desenvolvedores
- Permite que cada desenvolvedor possa fazer check-ins intermediários sem que os outros desenvolvedores sejam afetados



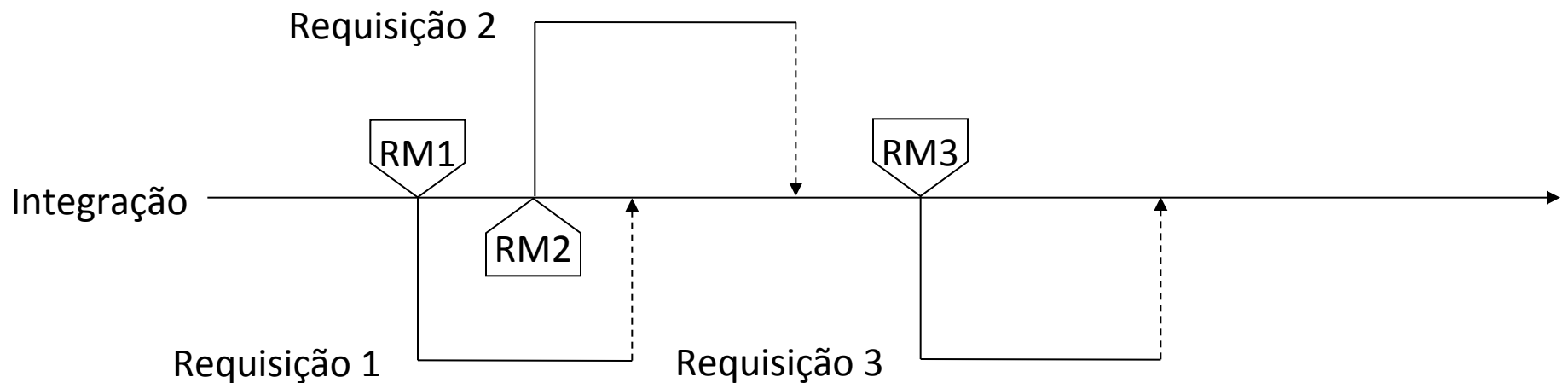
Estratégias de Ramificação

- Organização por subprojetos (componentes da arquitetura)
 - Ramo principal: integração
 - Ramos auxiliares: projetos
- Fornece baixo isolamento entre membros de um mesmo subprojeto, mas isola a equipe do subprojeto das demais



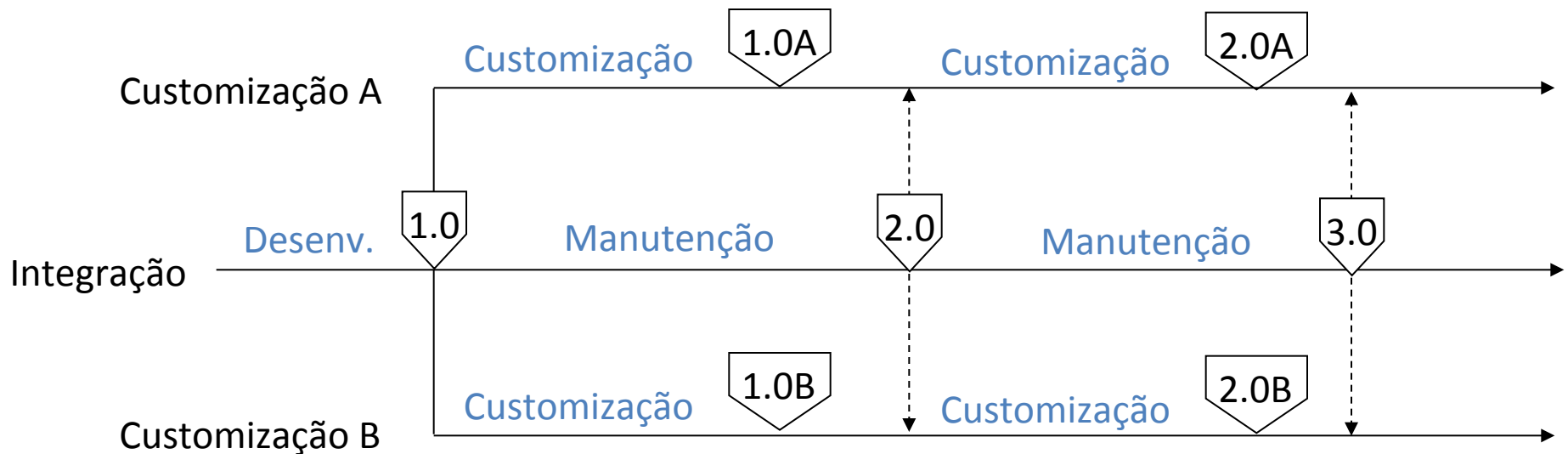
Estratégias de Ramificação

- Organização por requisições
 - Ramo principal: integração
 - Ramos auxiliares: requisições
- Permite que cada requisição seja identificada (change sets)
- Torna possível a remoção de uma requisição do produto



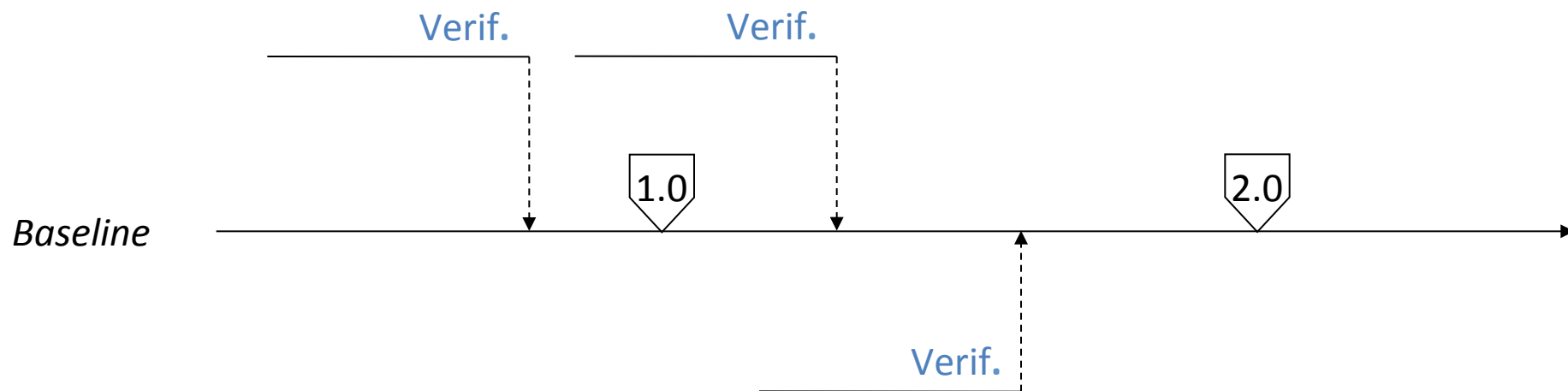
Estratégias de Ramificação

- Organização por customizações
 - Ramo principal: integração
 - Ramos auxiliares: customizações
- Seleção de variabilidade em tempo de construção usualmente é mais apropriado



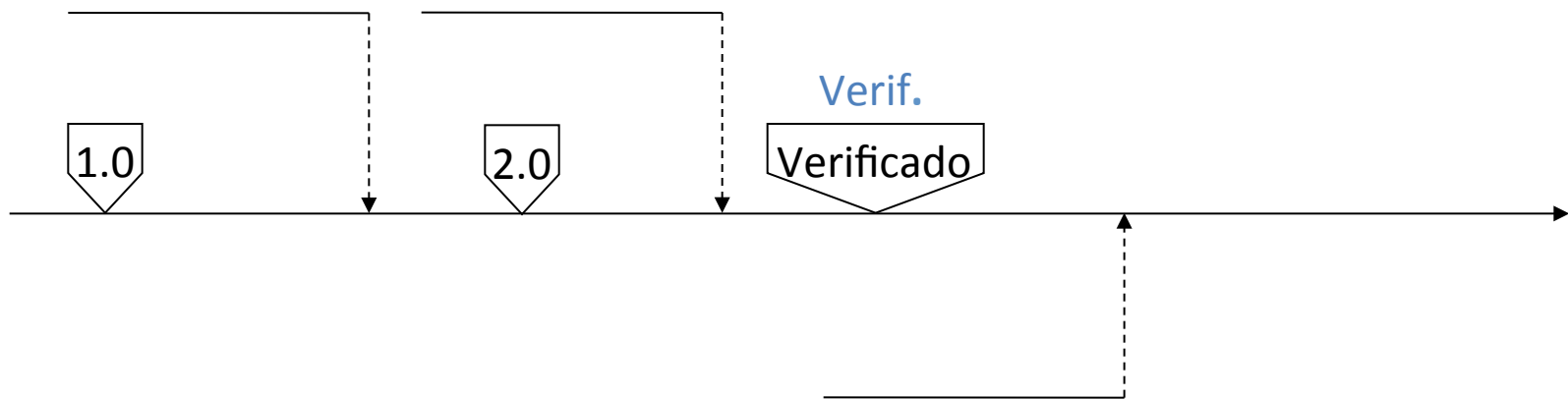
Estratégias de Ramificação

- Verificação contínua: toda e qualquer integração só é efetuada após passar pela verificação
 - Conceito de estado seguro
 - Ramo principal sempre pronto para liberação
 - Alto custo em caso de verificação não automatizada



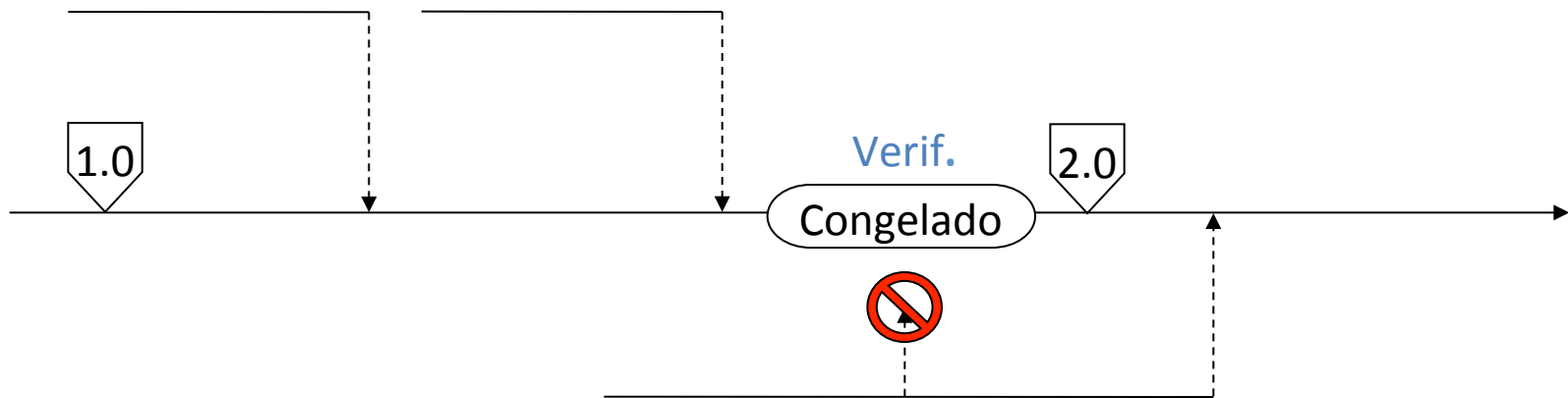
Estratégias de Ramificação

- Verificação periódica: as integrações são feitas de forma não sincronizadas com verificações, e as verificações são executadas de tempos em tempos
 - Torna independente a atividade de verificação
 - Utiliza rótulo para demarcar ponto já verificado
 - Custo médio em caso de verificação não automatizada



Estratégias de Ramificação

- Verificação pré-liberação: A verificação é executada somente antes da liberação
 - Conceito de congelamento do ramo principal
 - Só aceita correções durante congelamento
 - Custo baixo em caso de verificação não automatizada



Integração

- Processo de combinação de componentes de software e/ou hardware no sistema como um todo
- Possíveis momento da integração
 - Comando “salva” (integra um artefato)
 - Comando “*check-in*” (integra vários artefatos)
 - Junção de ramo (integra uma modificação como um todo)
- O ideal é fazer integrações de algo que faça sentido como um todo

Integração

- Estratégia *Push*
 - O desenvolvedor/testador envia para a área de integração a contribuição
- Estratégia *Pull*
 - O integrador busca nas áreas de desenvolvimento/testes a contribuição
 - O desenvolvedor/testador deve demarcar com rótulos o que já está pronto para integração

Integração

- O sistema pode ser integrado via referência ou inclusão de componentes
- Integração via referência
 - Cada componente é construído no seu próprio ambiente
 - O ambiente de integração referencia as liberações dos componentes
- Integração via inclusão
 - O ambiente de integração inclui os ambientes de cada componente (código fonte dos componentes)
 - A construção ocorre toda no ambiente de integração

Integração

- Rótulos podem ser utilizados durante o processo de integração
 - Rótulo CANDIDATA: Indica que a configuração é candidata a *baseline* ou liberação
 - Rótulo CONSTRUÍDA: Indica que a configuração não apresentou nenhum erro de compilação
 - Rótulo VERIFICADA: Indica que a configuração passou por todas as etapas de verificação
 - Rótulo REJEITADA: Indica falha durante o processo de verificação
 - Rótulo BASELINE_X: Indica uma *baseline*
 - Rótulo LIBERAÇÃO_X: Indica uma liberação

Integração

- Ramos também podem ser utilizados no processo de integração
 - Testes de unidade
 - Espaço de trabalho → ramo de desenvolvimento
 - Teste de integração
 - Ramo de desenvolvimento → ramo de integração
 - Teste de sistema
 - Ramo de integração → ramo de homologação
 - Teste de aceitação
 - Ramo de homologação → ramo de produção

Exercício

- Descreva uma estratégia de ramificação para as necessidades a seguir
 - Liste os pontos positivos dessa estratégia
 - Liste possíveis pontos negativos
 - Defina o que poderia ser feito para aprimorar a estratégia
- Necessidades a serem consideradas
 - Manutenção de versões anteriores
 - Customizações para diferentes clientes
 - Diferentes sistemas que fazem uso de bibliotecas comuns



Gerência de Configuração: Ramificação e Integração

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br