

Configuration Management: Branching

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br

Branching Strategies

- Branching strategies allow:
 - Segregating corrective from perfective maintenance
 - Isolating newcomers
 - Dealing with massive development teams
 - Isolating commits that belong to a specific feature
 - Promoting quality levels (unstable, testing, stable)
 - Customizing the system to different clients
 - Etc.
- Branching strategies are usually blended to provide specific results

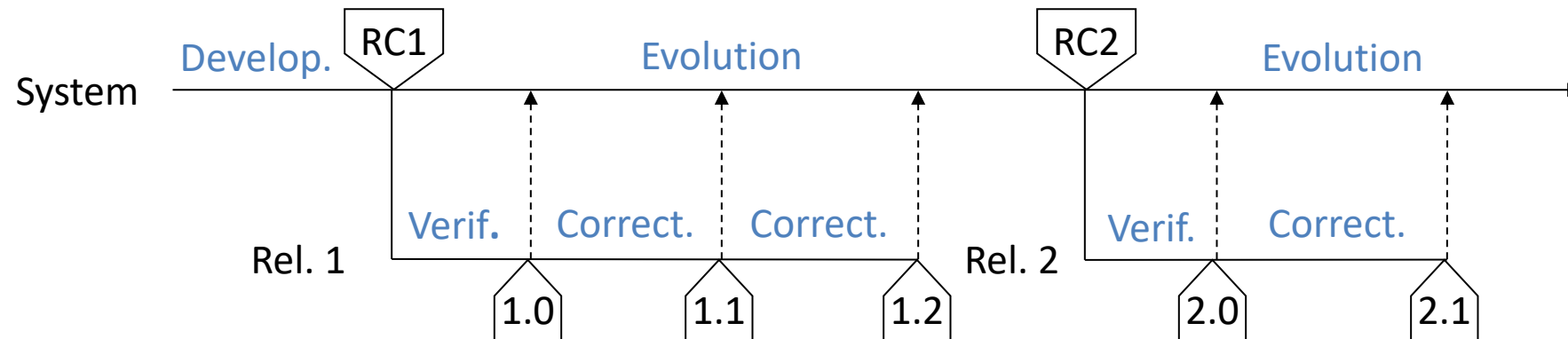
Branching Strategies

- **Maintenance Strategies**
 - Chaotic
 - Sequential
 - Waterfall
- Organization Strategies
 - Developer
 - Subsystem
 - Issue
 - Customization
- Verification Strategies
 - Continuous
 - Periodic
 - Pre-release

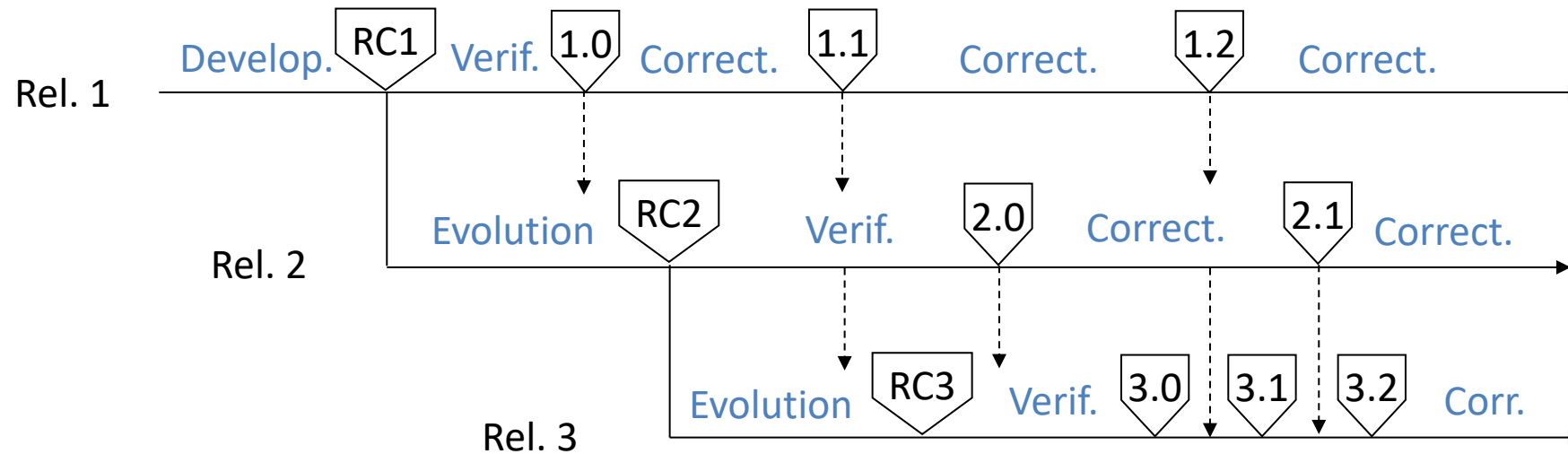
Branching Strategies: Chaotic



Branching Strategies: Sequential



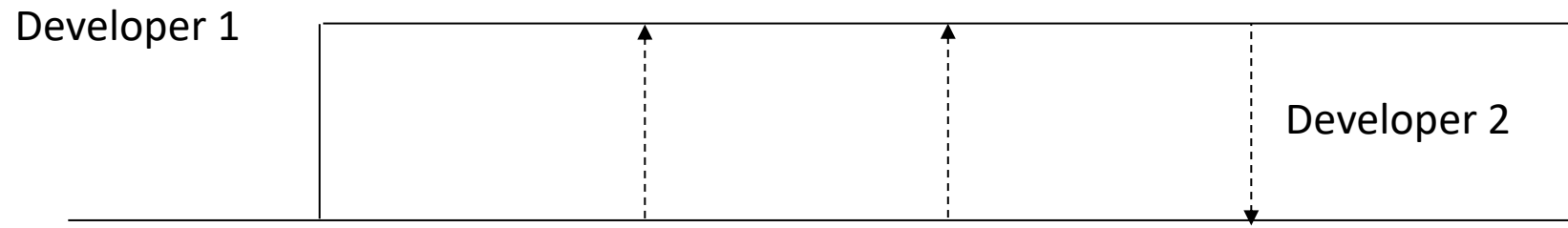
Branching Strategies: Waterfall



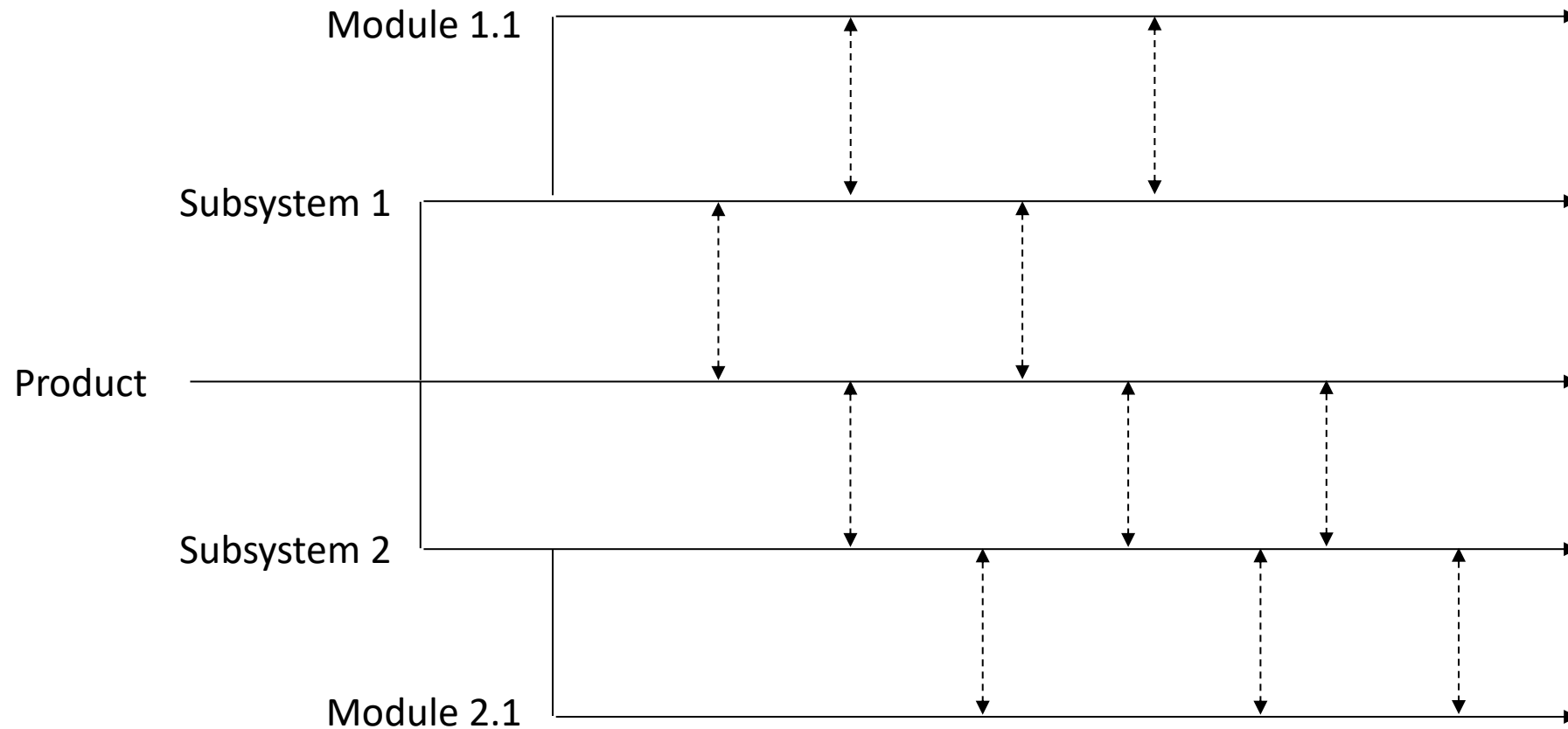
Branching Strategies

- Maintenance Strategies
 - Chaotic
 - Sequential
 - Waterfall
- **Organization Strategies**
 - **Developer**
 - **Subsystem**
 - **Issue**
 - **Customization**
- Verification Strategies
 - Continuous
 - Periodic
 - Pre-release

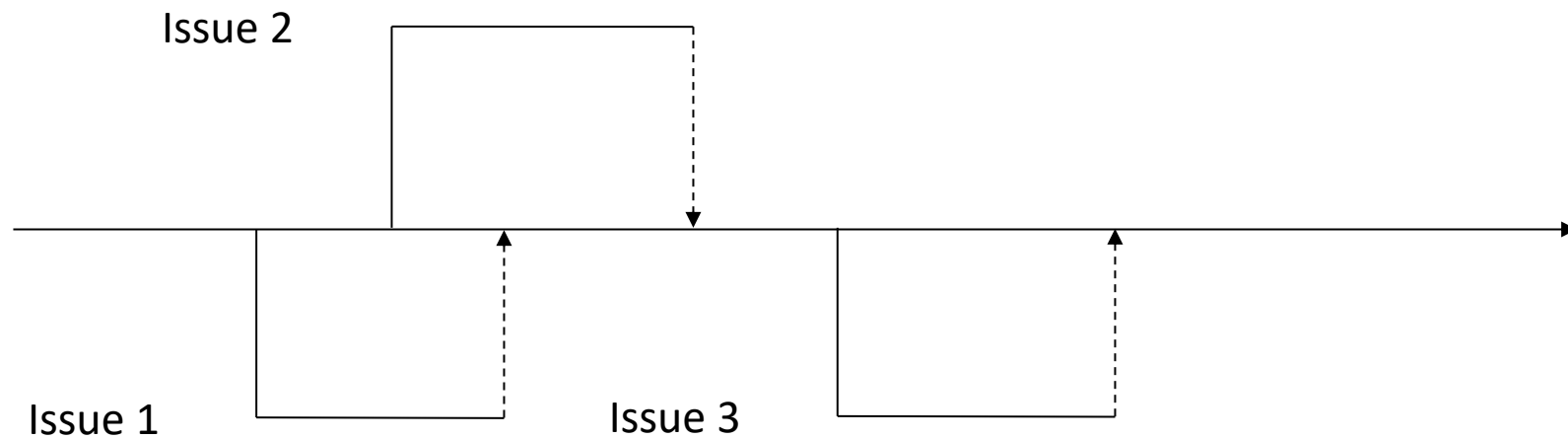
Branching Strategies: Developer



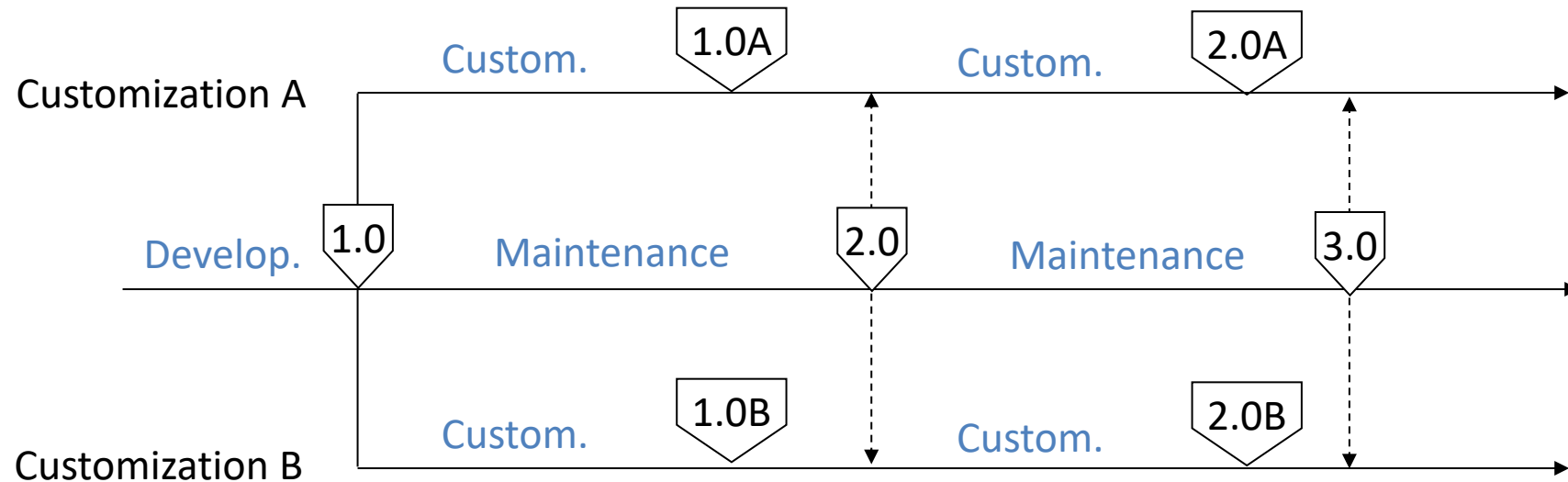
Branching Strategies: Subsystem



Branching Strategies: Issue



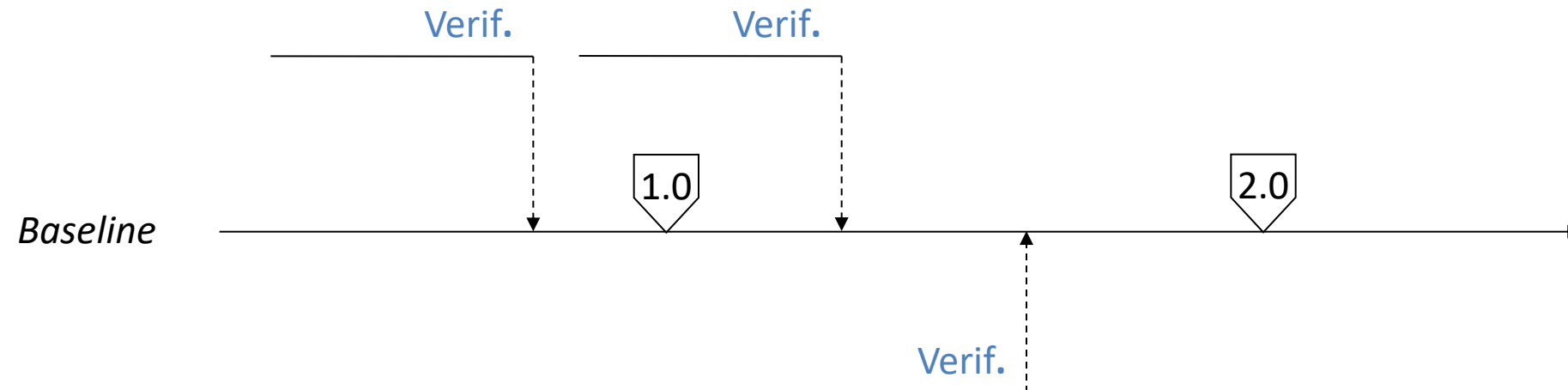
Branching Strategies: Customization



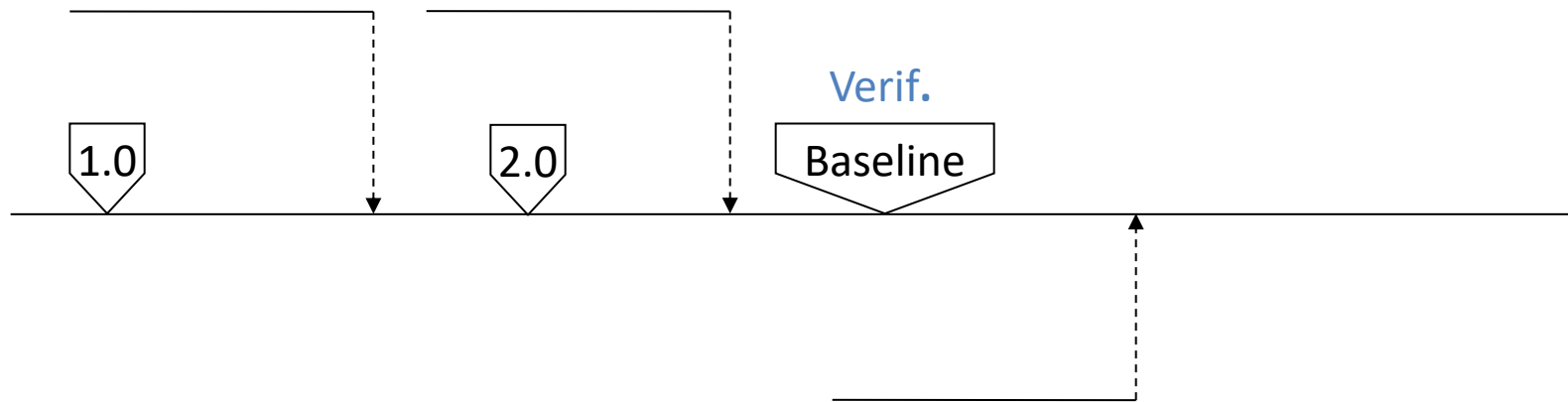
Branching Strategies

- Maintenance Strategies
 - Chaotic
 - Sequential
 - Waterfall
- Organization Strategies
 - Developer
 - Subsystem
 - Issue
 - Customization
- **Verification Strategies**
 - **Continuous**
 - **Periodic**
 - **Pre-release**

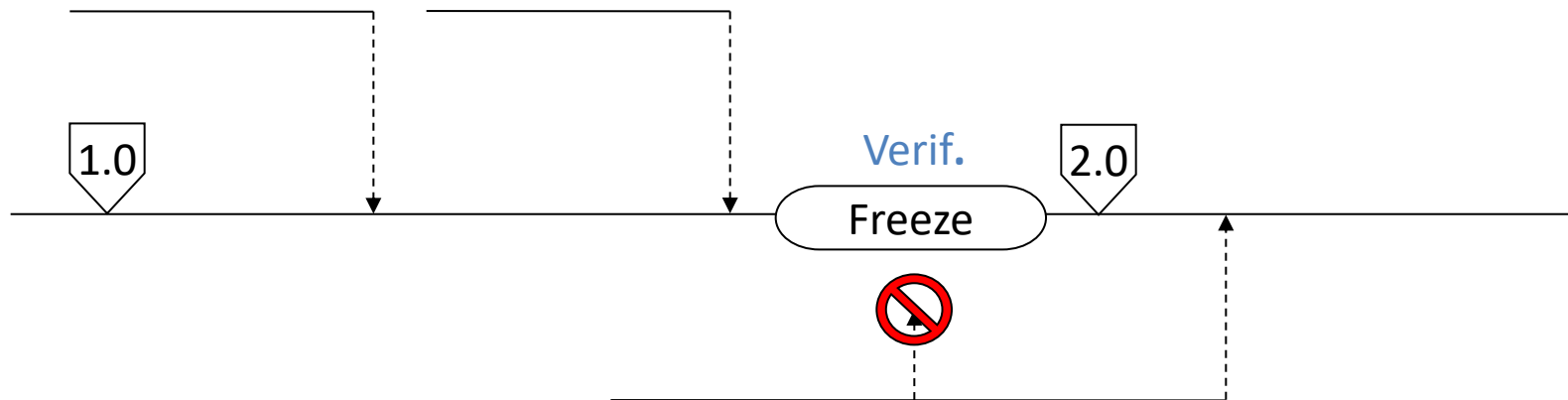
Branching Strategies: Continuous



Branching Strategies: Periodic



Branching Strategies: Pre-release



Integration

- Process that combines components into the system as a whole
- Moments
 - Commit
 - Branch merge
- Integration criteria
 - Only integrate commits/branches with changes that make sense to the other developers

Integration

- Push strategy
 - Developers send contributions to an integration branch
- Pull strategy
 - Developers request the integration of specific branch/tag
 - Integrators retrieve contributions from the specified branch/tag and merge them to the integration branch

Integration

- Integration by reference
 - Each component is build independently using their own build scripts
 - The integration environment references to the derived CI version of each component
- Integration by inclusion
 - The integration environment includes the source CI version of each component
 - All components are build together using only one build script

Integration

- Possible criteria for promoting code using branches
 - Workspace → Feature branch (unit tests ok)
 - Feature branch → Integration branch (integration tests ok)
 - Integration branch → Staging branch (system tests ok)
 - Staging branch → Production branch (acceptance tests ok)

Exercise

- Describe a branching strategy for the following requirements:
 - Maintenance of old versions
 - Customization for different clients
 - Home-made components that are reused in multiple systems
- For your branching strategy, explain:
 - Its positive aspects
 - Its negative aspects
 - What can be done to suppress its negative aspects

Configuration Management: Branching

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br