

Federal University of Acre  
Computer Science Postgraduate Program

## Distributed Software Development



# Software Configuration Management (SCM)

---

Prof. Dr. Daricélio Soares

# Agenda

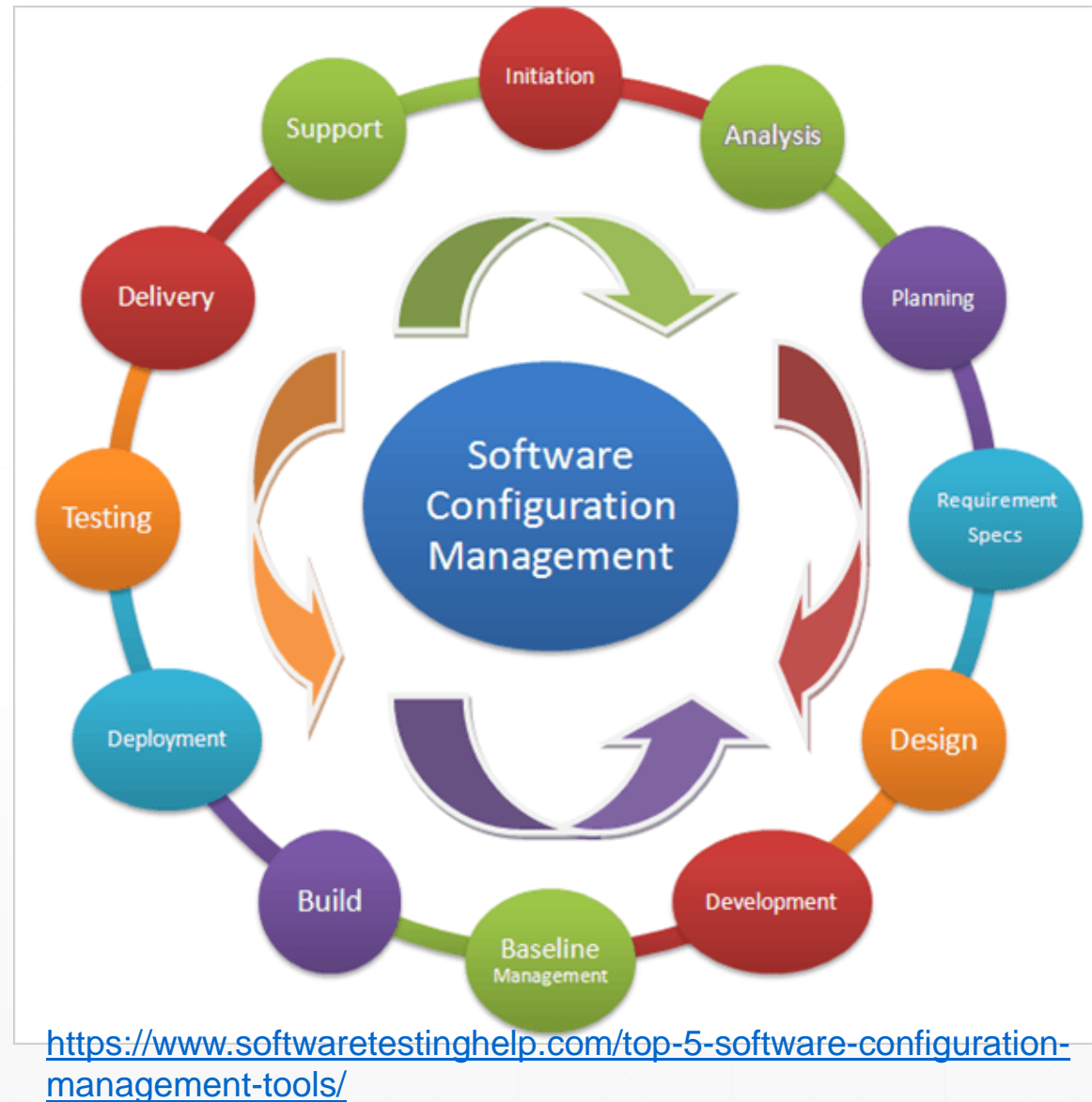
- Introduction
- Basic concepts
- Version Control Systems (VCS)

# Introduction

---

# Context

- How to control infinity?



# Context

- Software requirements

- How many?
- How many iterations?
- Do they evolve? When?

- Code

- How many versions will be created?
- Functional dependency?

- Core Team Size

- Evolution

- Bug fixes
- Refactoring
- Improvement
- New versions

# General Concept

- Software configuration management (SCM) is the discipline of controlling the evolution of complex software systems

(Walter Tichy)

- The **art** to identifying, organizing and controlling modifications in the developing software, maximize productivity by minimizing mistakes

(Wayne Babich)

# Basic concepts

---

# Software Configuration

- A software development project produces the following items
  - Programs (source code, executable programs, component libraries)
  - Documentation (user manuals, requirements document, analysis and design model)
  - Data (test and project data)
- These sets of items are collectively called **software configuration**



# Configuration items

- Typically, a configuration item is established for each software artifact that can be designed, deployed, and independently tested
- A set of hardware and/or software items viewed as a single entity for configuration management purposes
- A configuration item is subject to change and must conform to established policies

# Baselines

- A specification or product that has been formally revised and accepted
  - It serves as the basis for the next steps
  - The software configuration in a discrete point in time
  - It can only be modified through formal procedures (i.e. change requests)
  - A set of artifacts only becomes a **configuration item** after a baseline is established

# Version

- Different instances of the same CI
- Versions types
  - Revision
    - created to replace previous version following a timeline
    - response to correction and/or evolution
  - Variant
    - coexistent versions designed for different purposes
    - different architectures and/or platforms
  - Cooperation
    - Draft versions

# Revision



<https://www.tecmundo.com.br/windows-10/64136-windows-1-windows-10-29-anos-evolucao-do-so-microsoft.htm>

# Variant





# Cooperation



Base Version



Joe's WS



Zeus's WS

# Merged (Draft)



REVISIONS

# Diff

- Remove
  - Microfone
  - Camisa



- Add
  - Barba
  - Blusa



# Configuration Management System

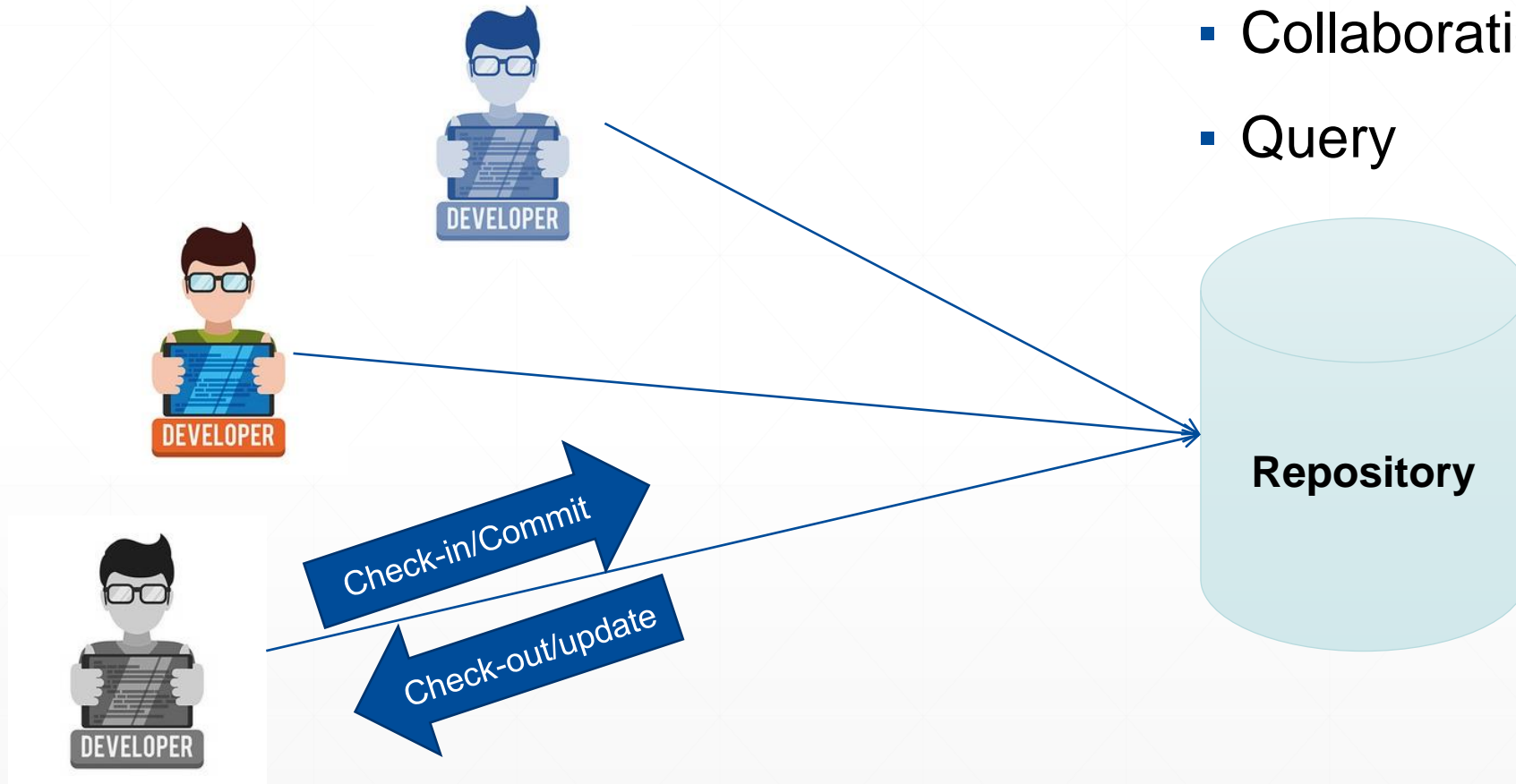


# Version Control Systems (VCS)

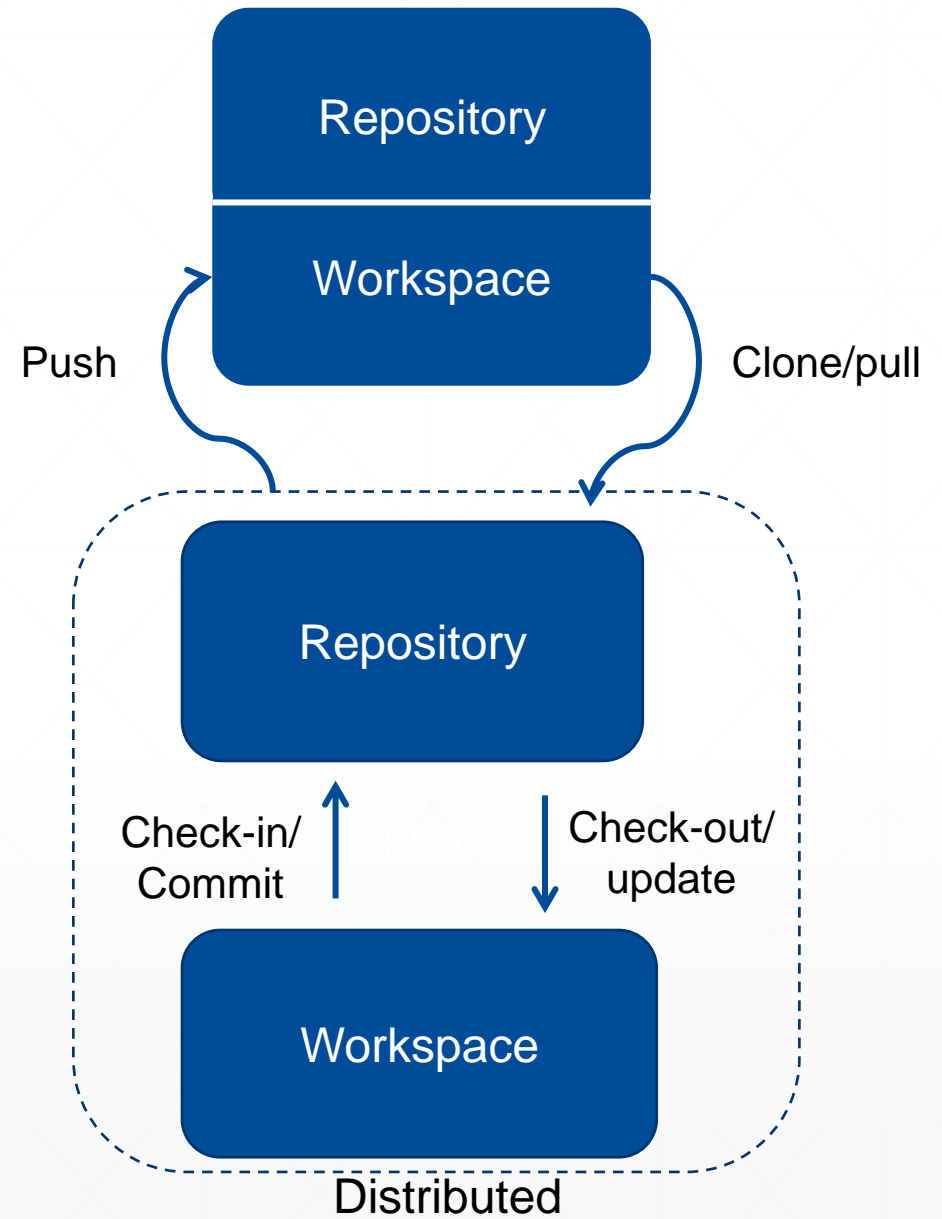
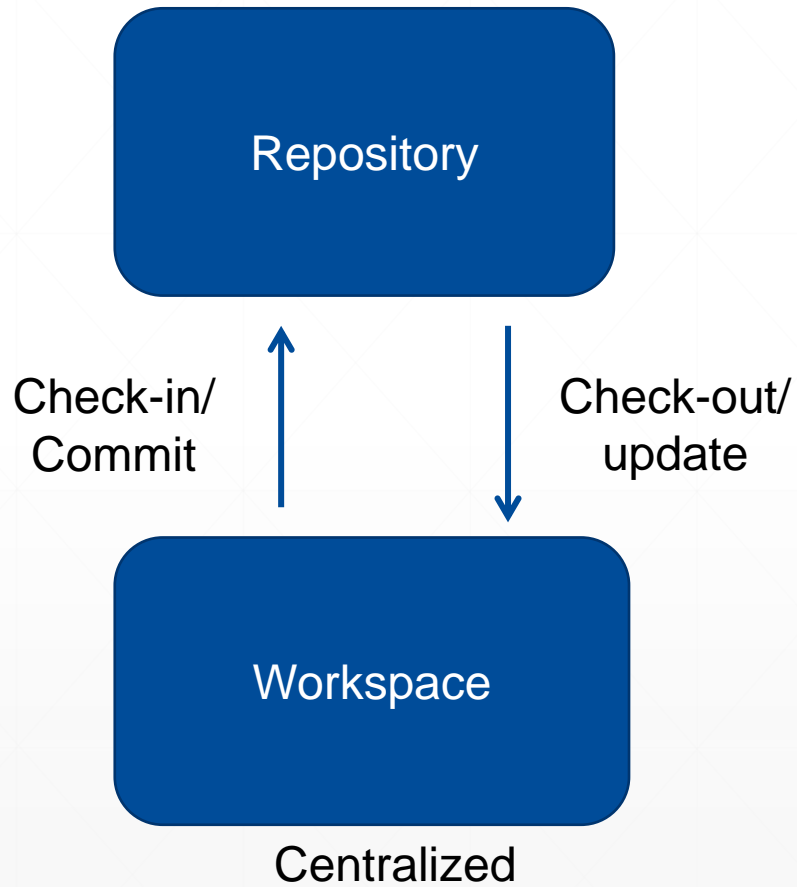
---

# Version Control System

- Topology
- Storage
- Collaboration
- Query



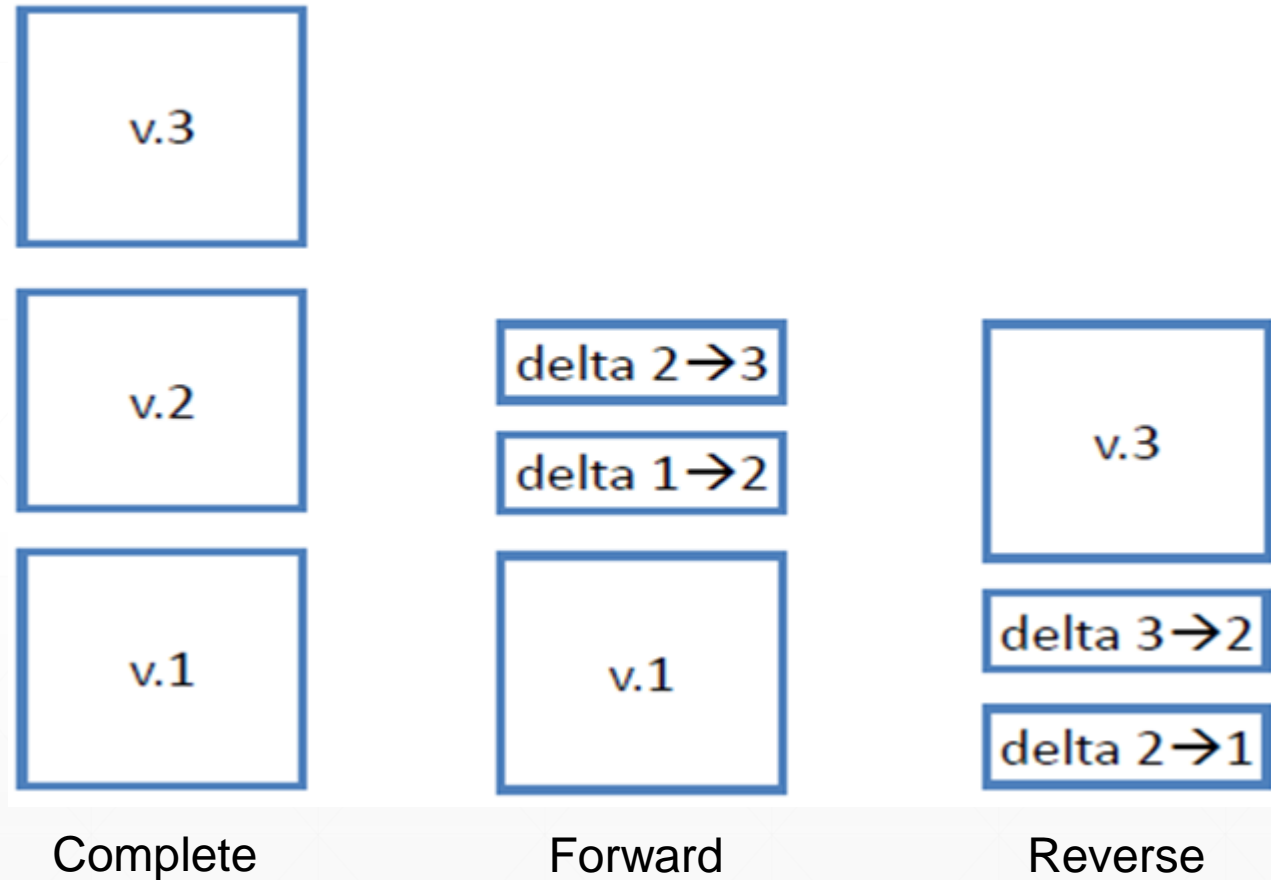
# Topology



# Storage

- Complete
  - Large disk space
  - Fast recovery
- Differences
  - Minor disk space
  - Versions derived from deltas
  - High processing cost
  - Types:
    - Forward / Reverse

# Storage

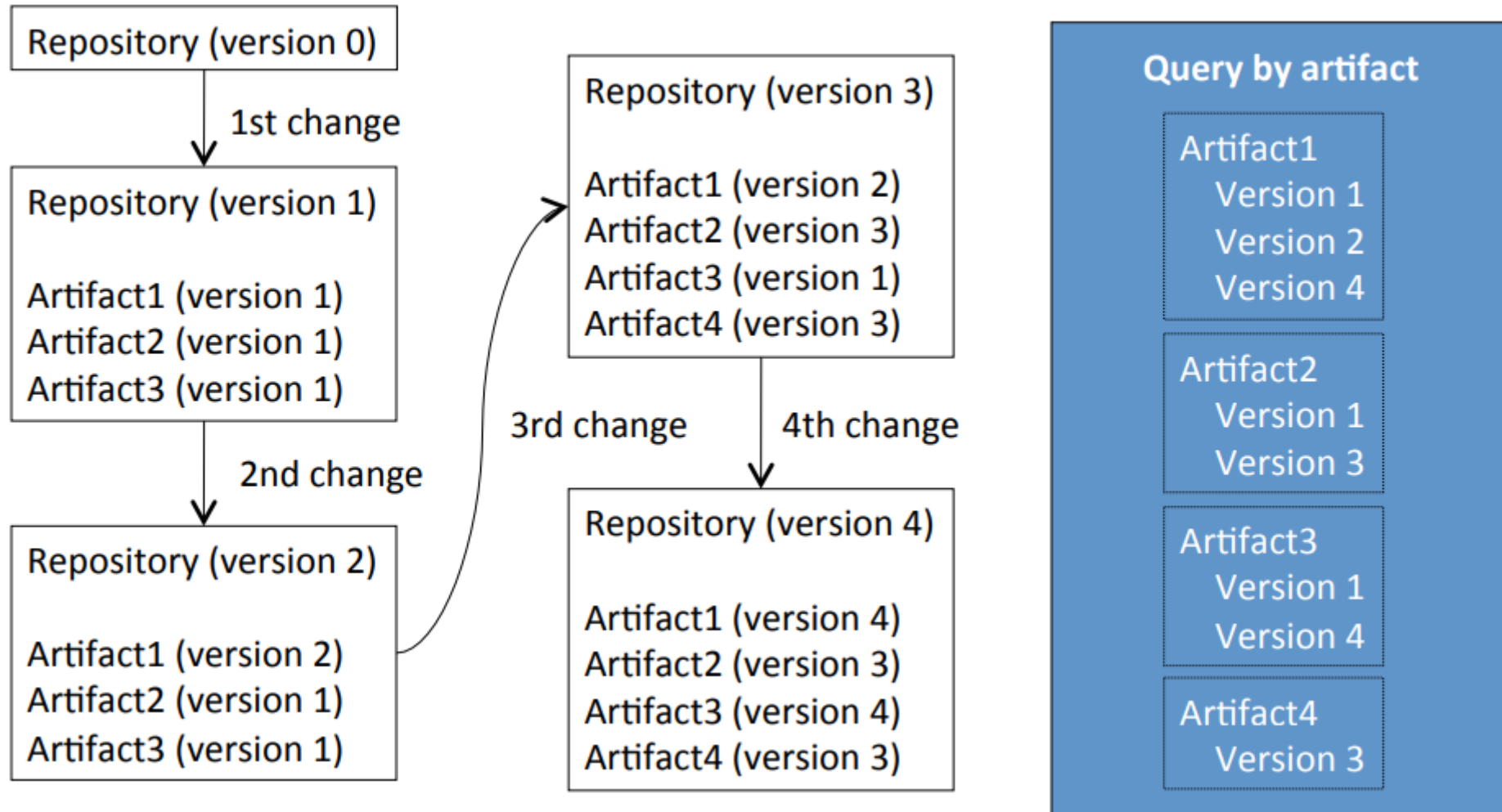


<http://www2.ic.uff.br/~leomurta>

# Collaboration

- Pessimist
  - Only one developer modifies an CI at any given time
  - No merge cost
  - Does not allow parallel work
- Optimist
  - Multiple developers can modify an CI at the same time
  - High merge cost
  - Allows parallelism
- Optimist with notification
  - Good cost-benefit
  - Allows any developer to know who is modifying an CI

# Query

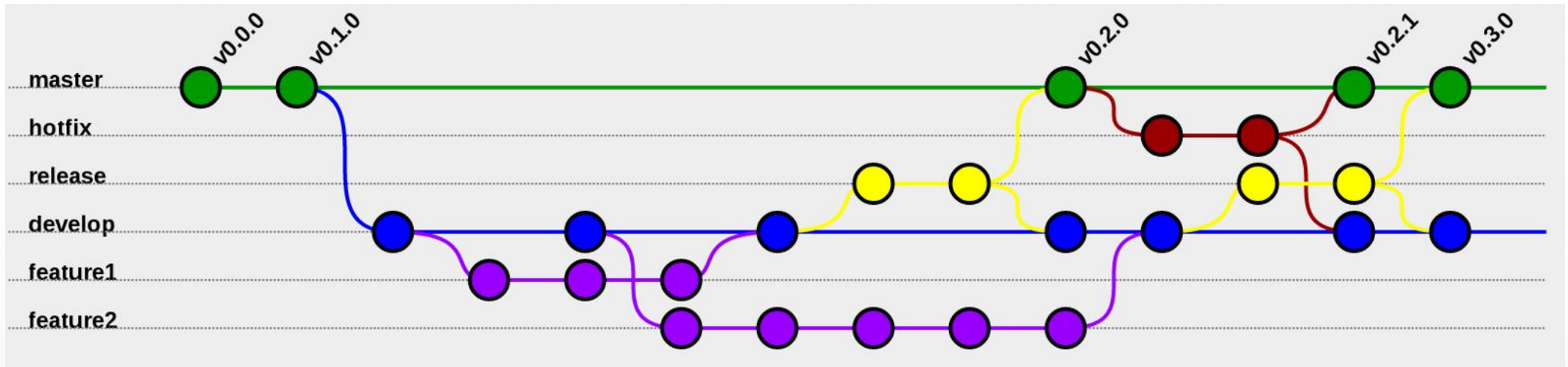




# Branches

- Versions that do not follow the main line of development
- Provide insulation for the development process
  - Branches are usually migrated to the main line of development
  - Migration can be complicated in case of long insulation
- A developer's **workspace** can be viewed as a branch
  - Extremely isolated (branches are shared by other people)
  - Resides on the client (branches reside on the server)
  - Temporary (branches are permanent)

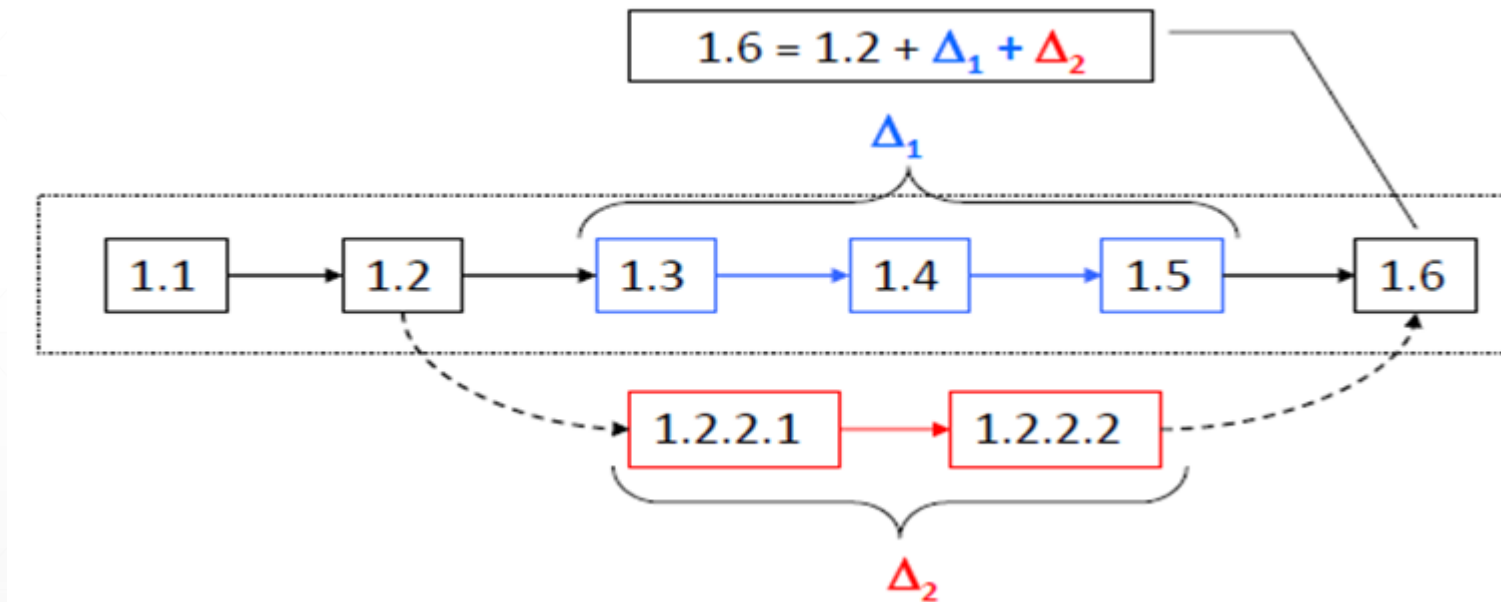
# Branches



<https://medium.com/@stansarr/git-workflow-branches-strategy-4d29f9b2a417>

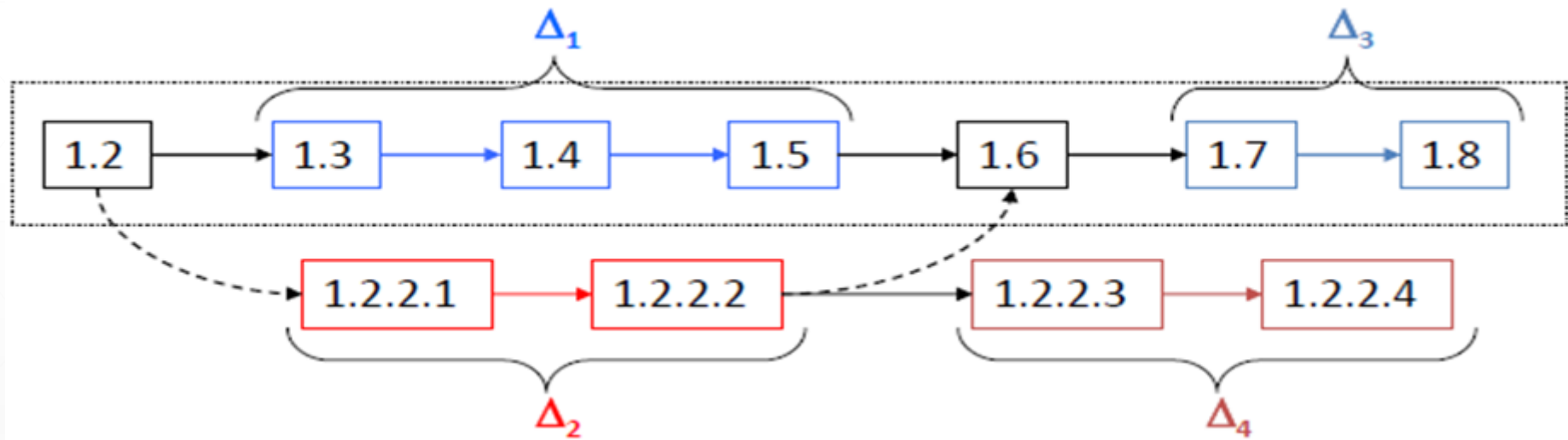
# Merge

- HelloWorld.java



# Merge

- HelloWorld.java



# Questions?

