Version Control System &

Collaboration Tools

Quan Vuong
Trainer

Agenda

I. Version Control concepts
II. Version Control tools
III. GIT basics
IV. Branches on GIT
V. Tools in GIT
VI. SVN basics and tools
VII. Games

Course Audience and Prerequisite

- The course is for freshers and newcomers
- The following are prerequisites to Version control and Collaboration tools:
  - Windows
  - Computer skills
  - File management
  - Teamwork

Course Objectives

- At the end of the course, you will have acquired sufficient knowledge to:
  - Understand the Version control concepts, tools and workflows
  - Manage the version of source code or files on GIT and SVN repositories
  - Practice the basic operations on Version control system
  - Do the branching and merging on Version control repositories

Version Control Concepts

Scenario


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.
CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.
corrections9.MORE.30.doc
FINAL_rev.22.comments49.
corrections.10.#@$%WHYDID
ICOMETOGRADSCHOOL????.doc

What is Version Control?

- Method to centrally store files
- Record a copy per change
- Log who, when, where, what
- Recover files if something wrong
- Also called as revision control / source control

Why Version Control?

- Ability to back-up
- Integrate sources / sub systems
- Collaboration with other people
- Troubleshooting
- Productivity

How it helps you …

- Change code in small steps
- Log changes in versions
- Feel safe when changing code
- Easy to try out things

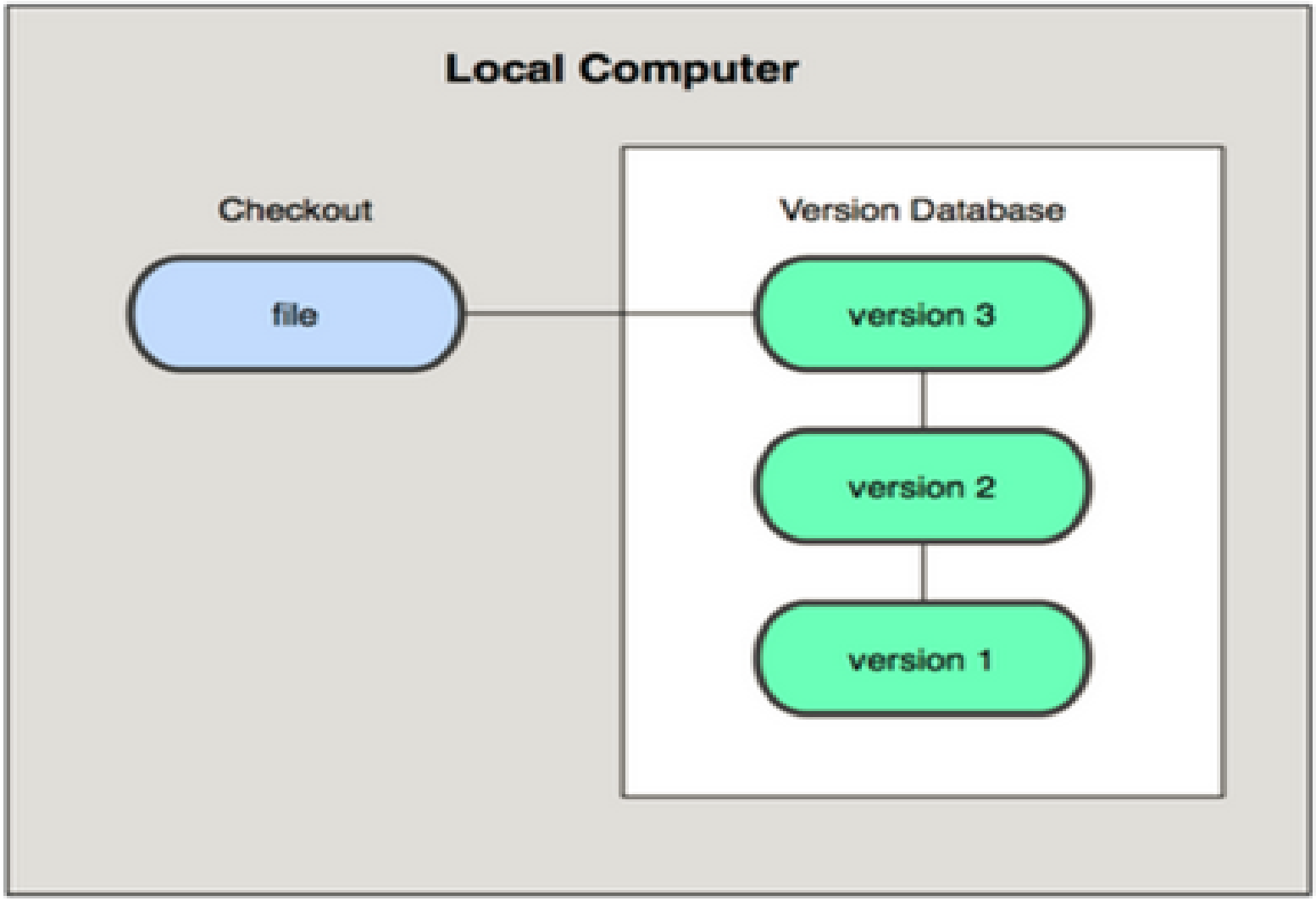
How it helps team...

- Allow team to work on same source base
- Handle collision by merging function
- Answer who did what
- Team work productivity

Version Control System

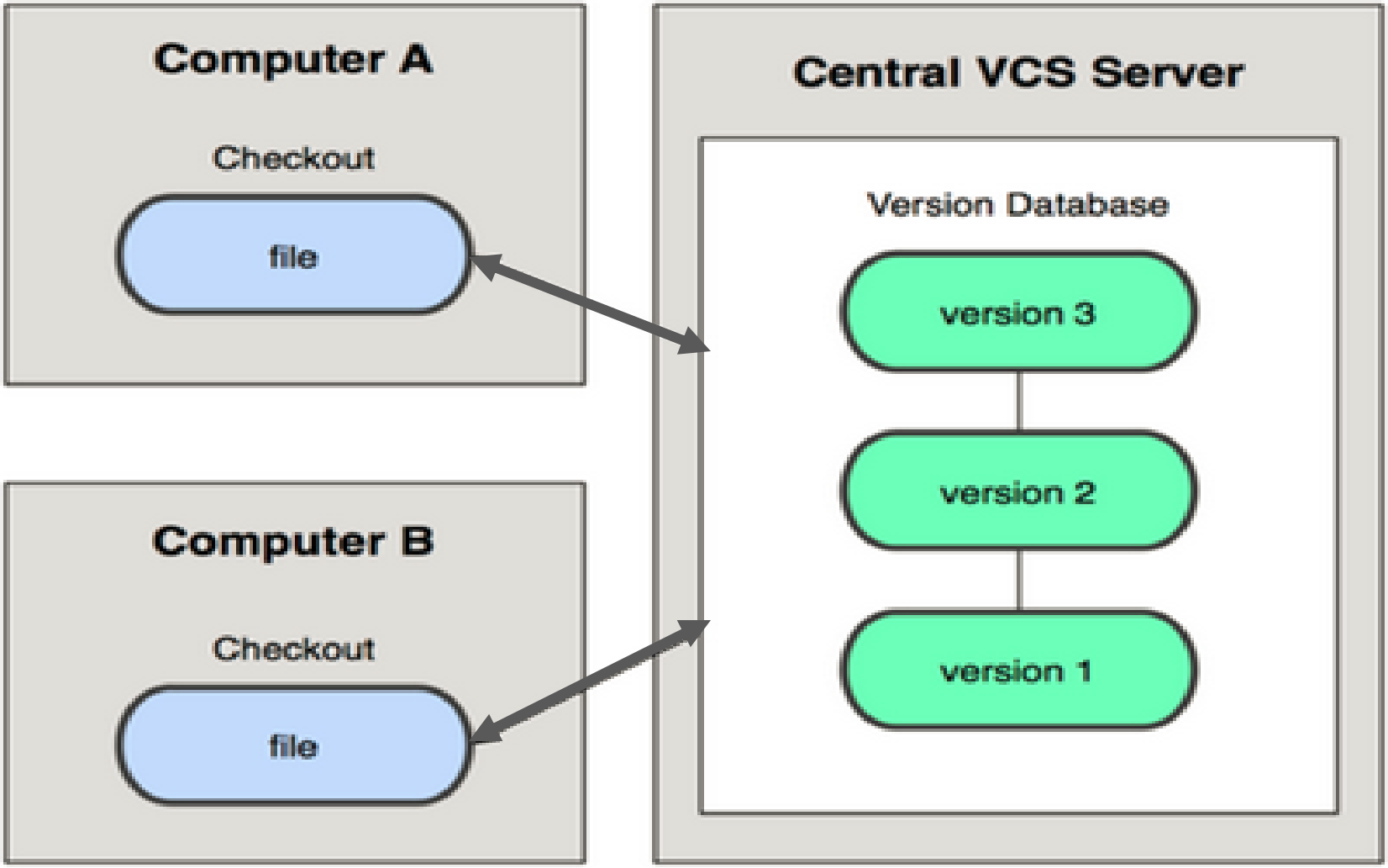
- Tool to manage file changes
- Three types
  - Localized
  - Centralized
  - Distributed
- Also called
  - Revision control software
  - Version management software
  - Source control software
  - Configuration management software

Localized Version Control System



Check out files on Local computer

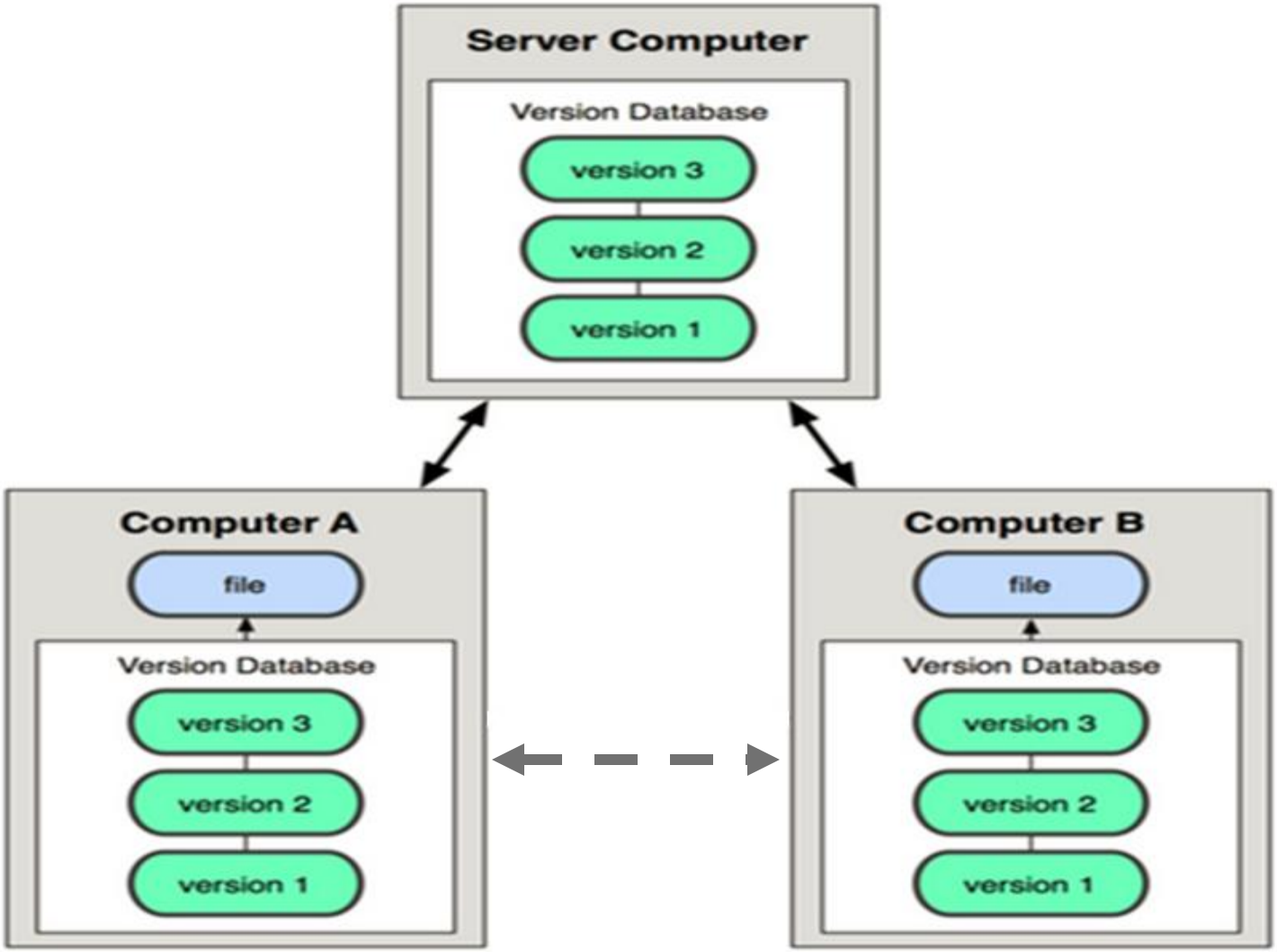
Centralized Version Control System



Check out files from Remote server

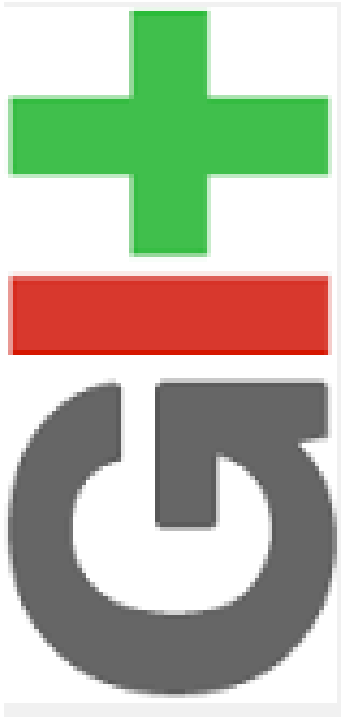
Centralized Tools

Distributed Version Control System

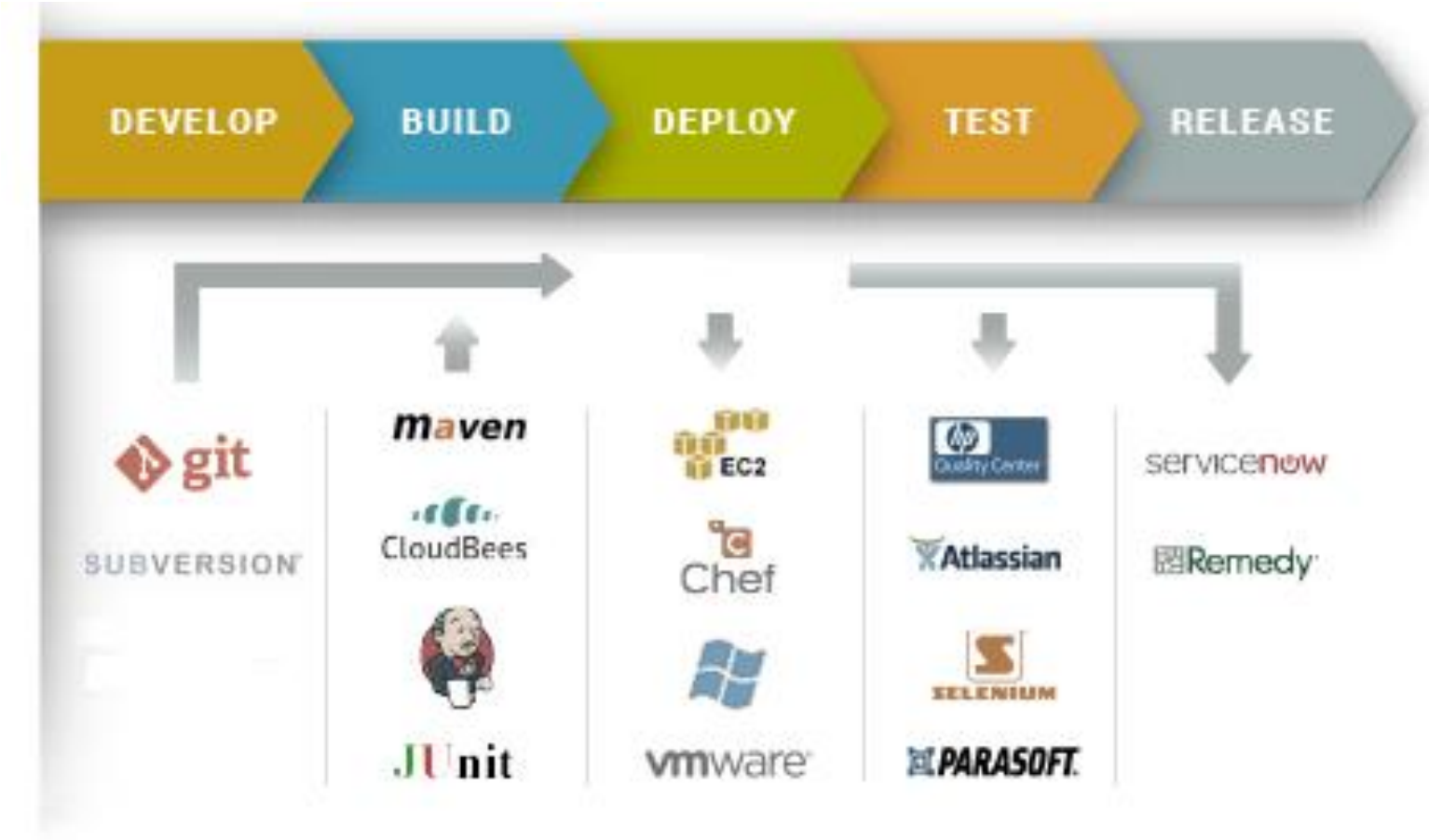


Check out files from Server or from peer

Distributed Tools



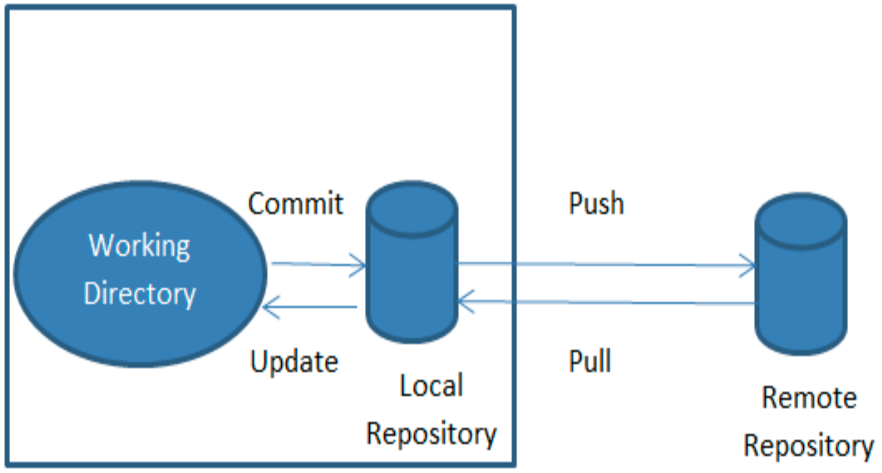
Version Control in Real World



Continuous Integration (CI) process

Version Control Vocabulary

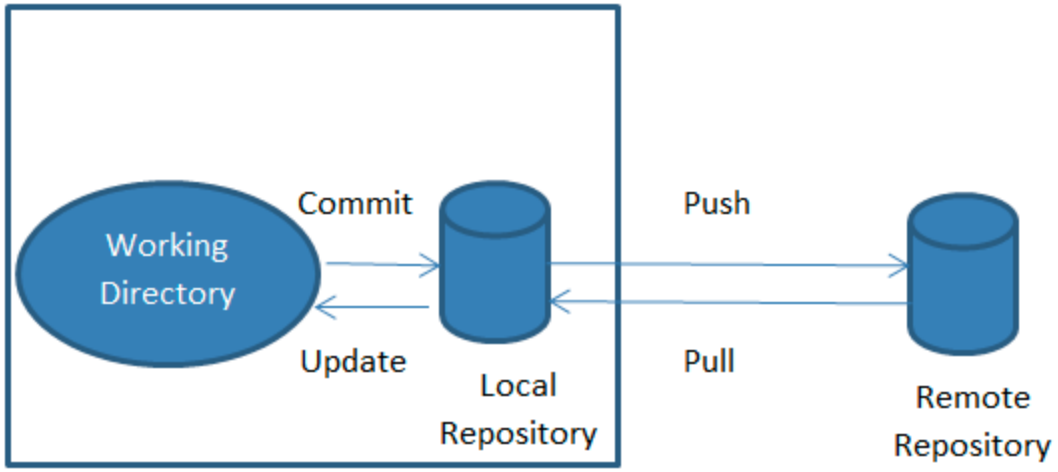
- Repository
- Server
- Client
- Working Copy



Version control basic operation

Basic Operations

- Add
- Change
- Commit
- Revert
- Update
- Tags
- …



Version control basic operation

What is branch?

- Copy of current tree / workspace
- Support parallel development

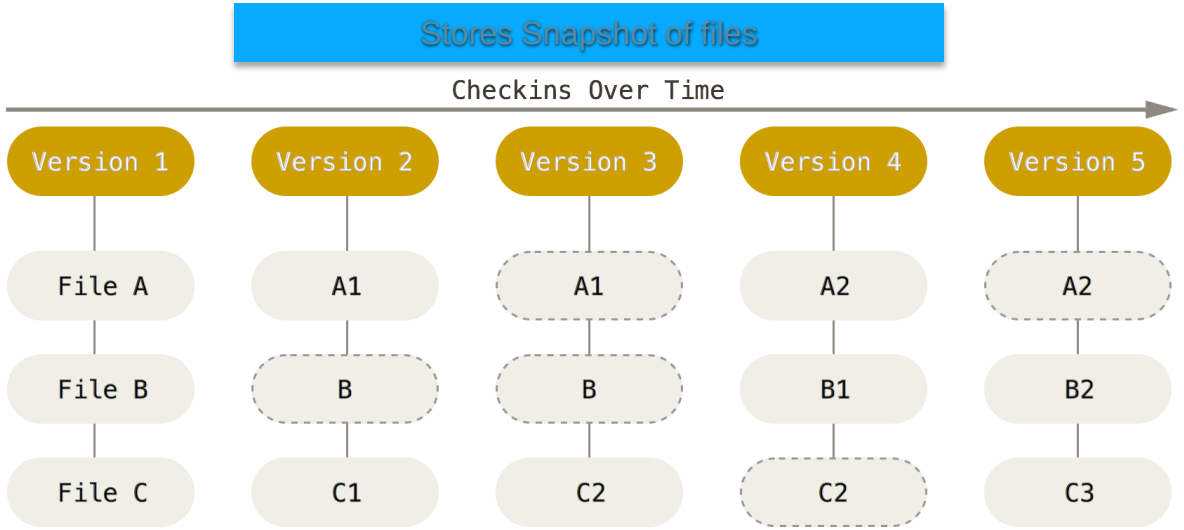
Version Control Tool

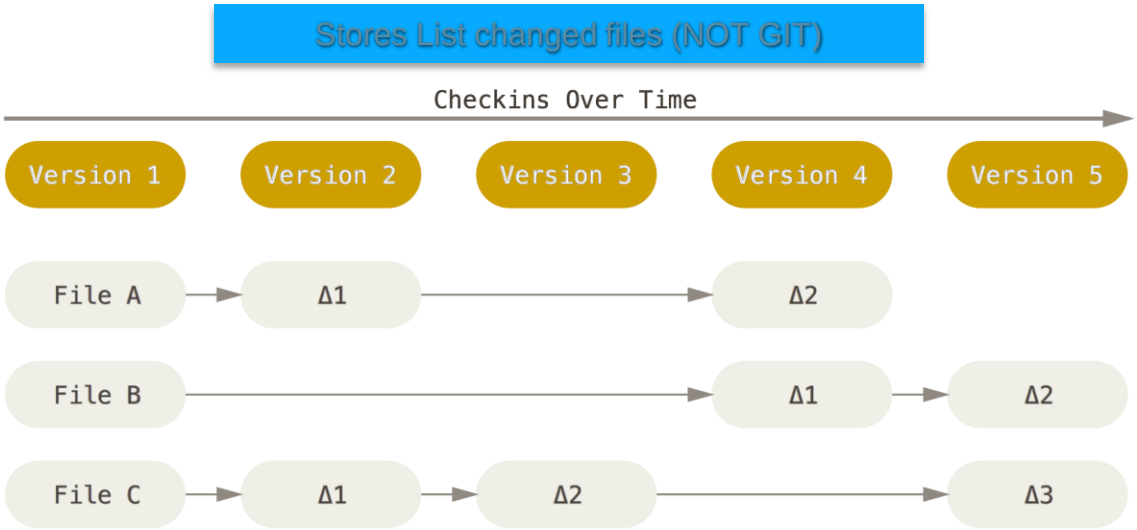
GIT

GIT

- Distributed Version Control System
- Multiple redundant repository
- Each dev have a copy of repository & history
- Community support and use widespread currently
- Reference at https://git-scm.com/downloads

GIT & Not GIT behavior





GIT check in (top) and not GIT check in (bottom)

Why GIT?

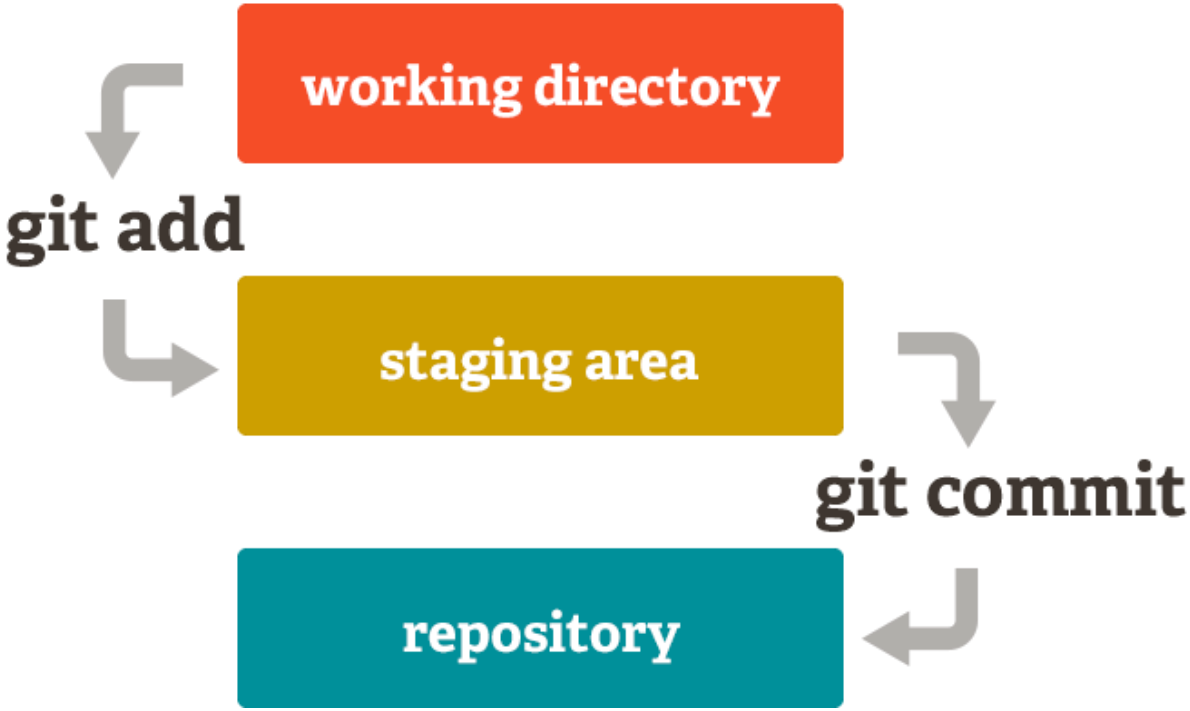
- Best branching & merging mechanism
- Work offline
- Easy switch, no checkouts
- Data Assurance
- Undo mistake
- Free Opensource

Git Basics

The three stages of GIT

The basic Git workflow:

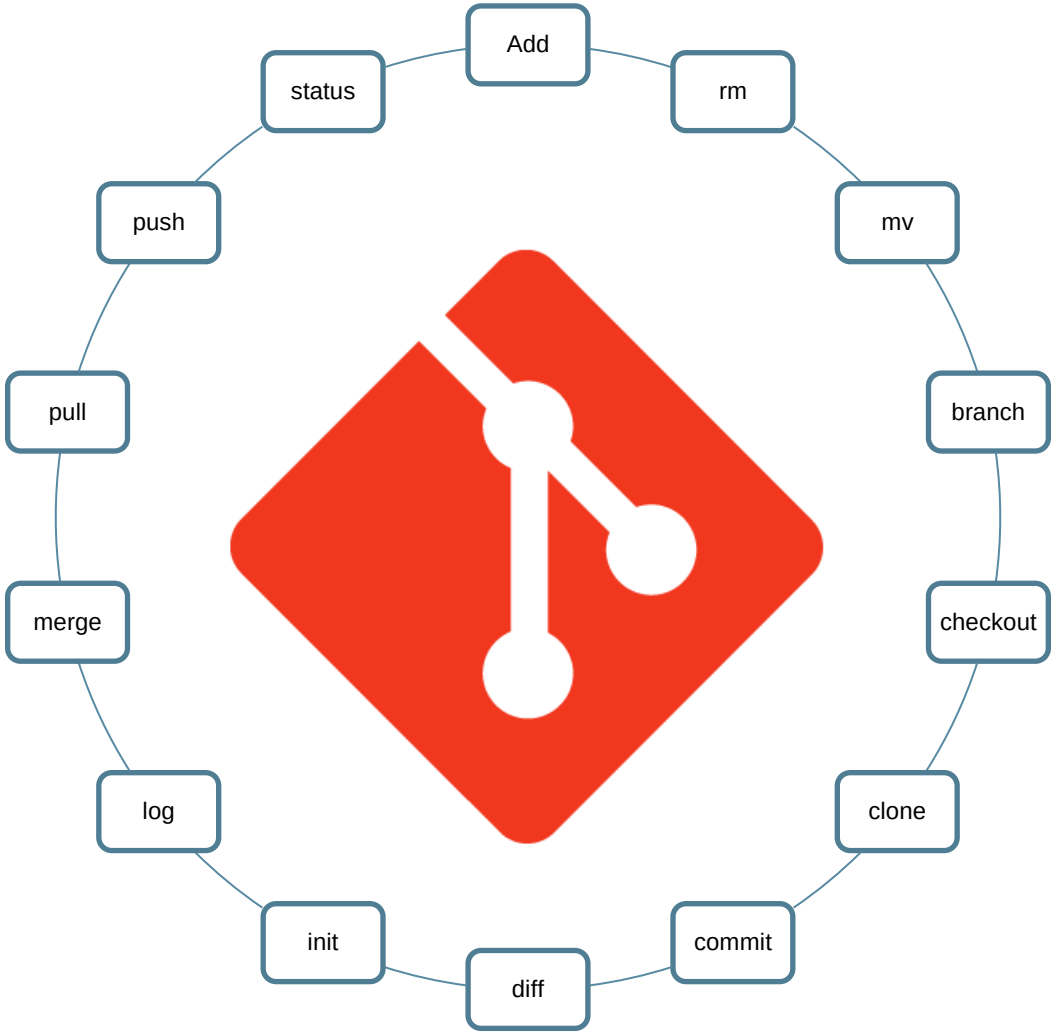
- Create: You get files in your working directory.
- Stage: adding snapshots of them to your staging area.
- Commit: takes the files in the staging area and stores that snapshot permanently to your Git directory.



GIT commit workflow

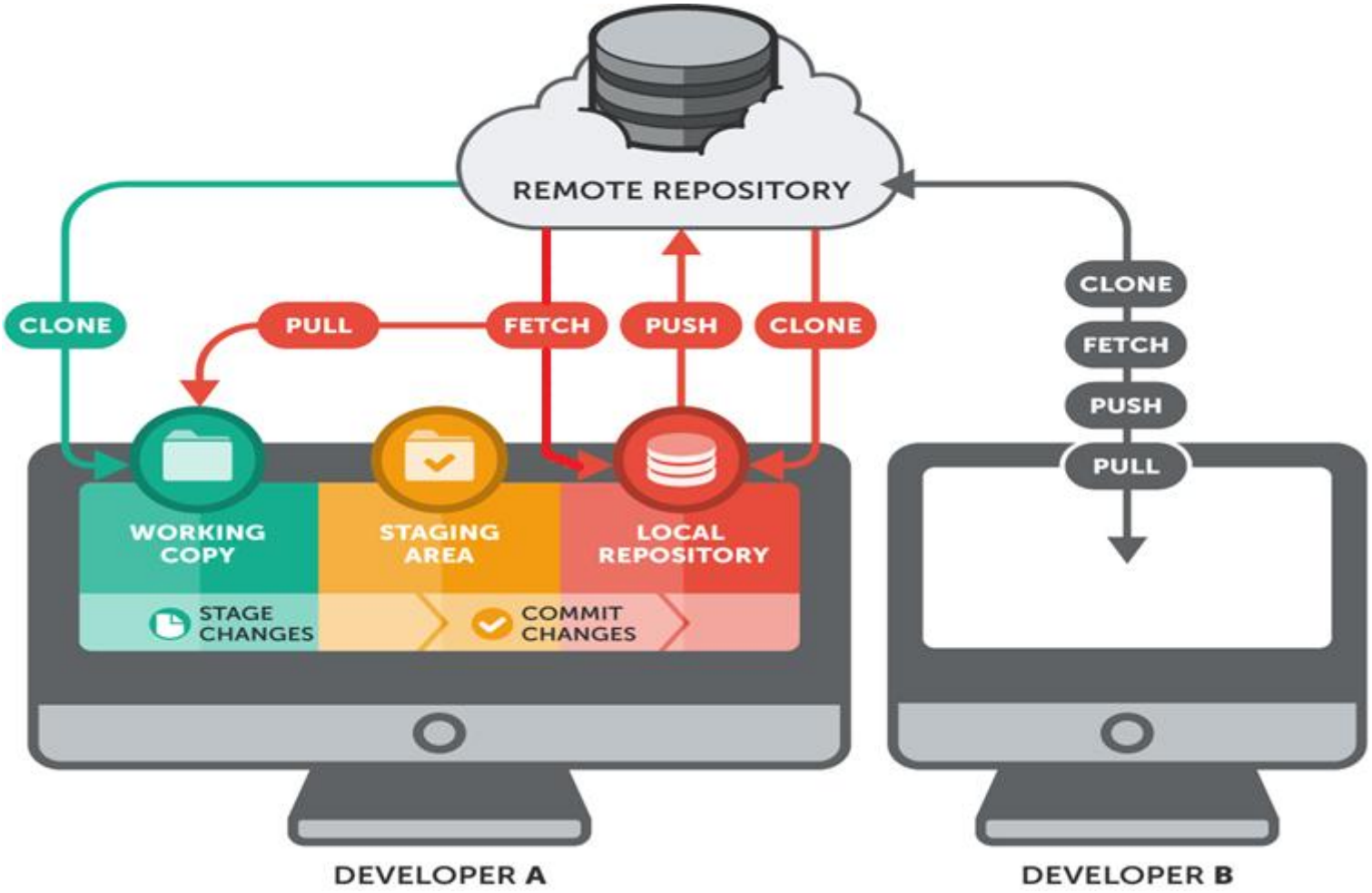
GIT Common Basic Commands

- Add file a:
  - git add a
- Commit file a:
  - git commit –m ‘add a’
- Push file a to server:
  - git push origin master



GIT basic commands

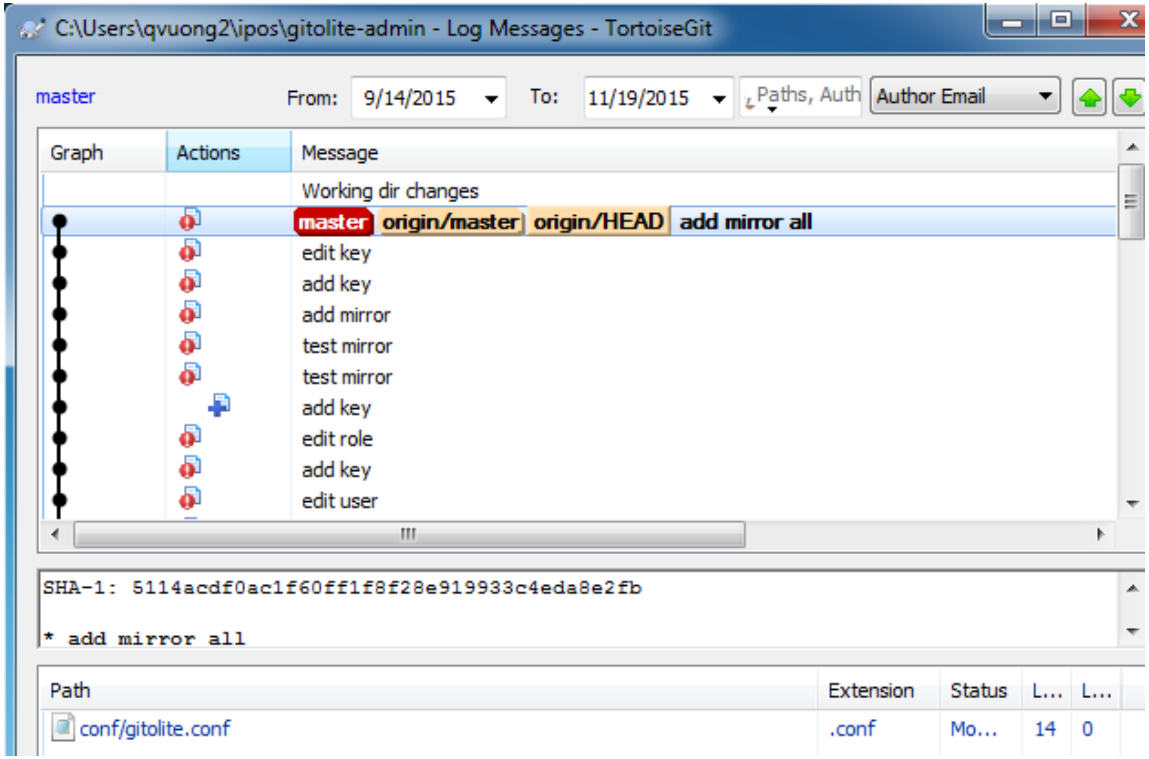
GIT workflow



GIT clone, fetch, push and pull from server

Git Basics - Viewing the Commit History

- How to view history
- Limit log output
- Use GUI to visualize history



Git Basics

- **Summary**
  - Know how to create repository
  - How to make, stage and commit changes
  - Ways to view the history of changes

Branches on Git

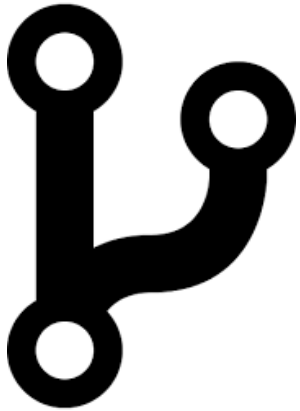
Branches on Git

- **What is branch?**
  - A branch is a lightweight movable pointer to the commits
  - The default branch name is master
- **Create a branch**
  - Create and check out a new branch: *git checkout –b test*
- **Switch to an existing branch**
  - Check out ‘master’ branch: *git checkout master*

Branches on Git – Merging

- **Merging**
  - Combine two branches,i.e: master and test
  - Command on master branch: *git merge test*
- **Resolving merge conflicts**
  - Control the different between two branches.
- **Managing branches**
  - Show branches with command: *git branch*

Git Branches



Default Branch

- master

Create new branch

- git branch develop-v1

Create and switch to new branch

- git checkout –b develop-v3

Switch to branch and merge existing branch as well

- git merge develop-v3 master

Branches on Git

- **Summary**
  - Covered basic branching and merging in Git.
  - Create, switch between branches and merge / rebase local branches
  - Share branches by pushing them to a shared server

Tools in Git

Git Installation

- Download TortoiseGit from URL:https://tortoisegit.org/download/
- Choose your version 32bit or 64bit OS.
- Double click TortoiseGit-1.8.15.0-64bit.msi to install it.

Git Demo

- Create and manage a file version on Git repository

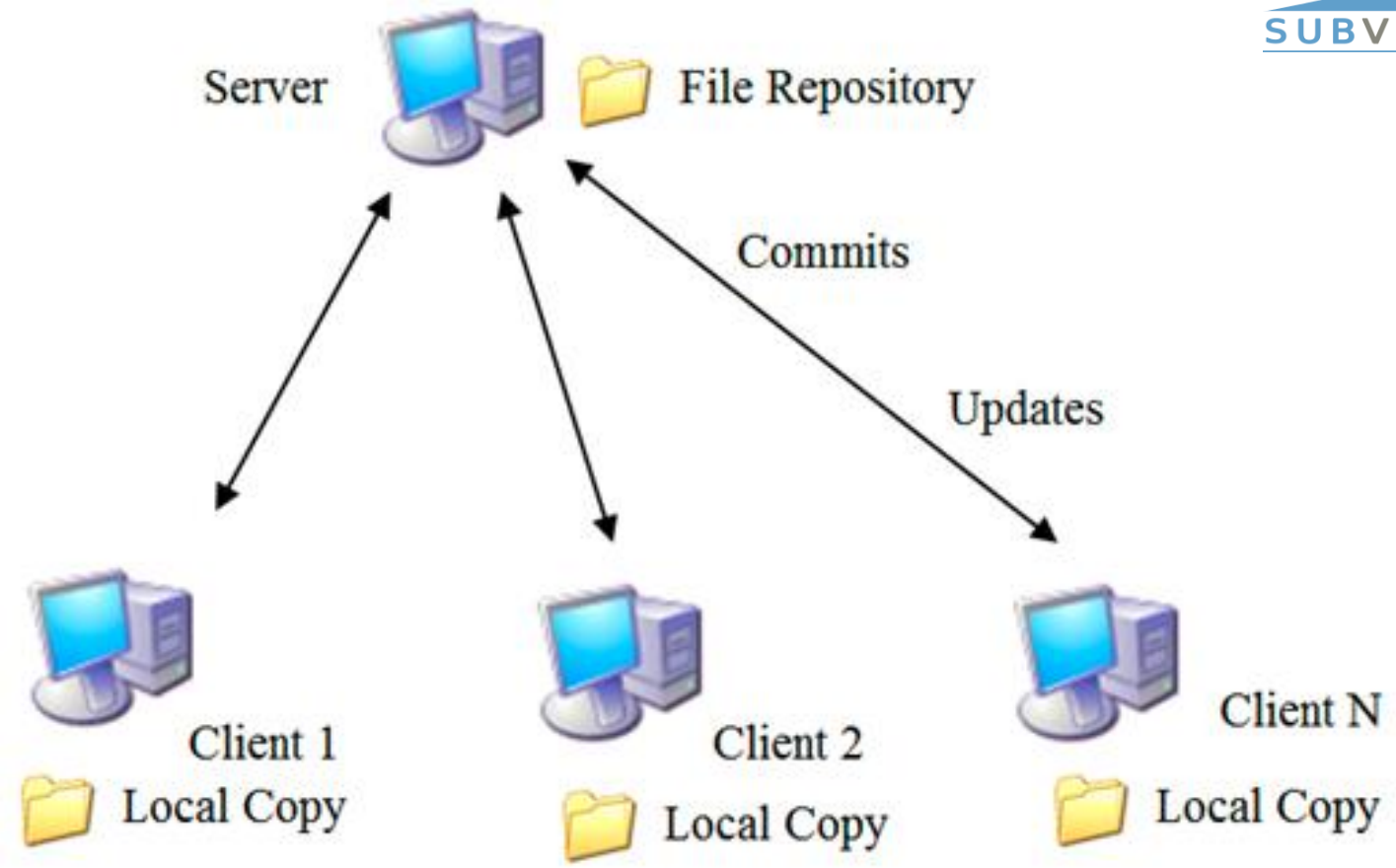
Version Control Tool

Subversion

Subversion - SVN

- A centralized system with core repository
- Designed as compelling replacement for localized
- Command includes *checkout*, *commit*, *update*…
- Several features:
  - Not file repository but database
  - Version not file-based
  - Directory based tags and branches
- Download at http://tortoisesvn.tigris.org/

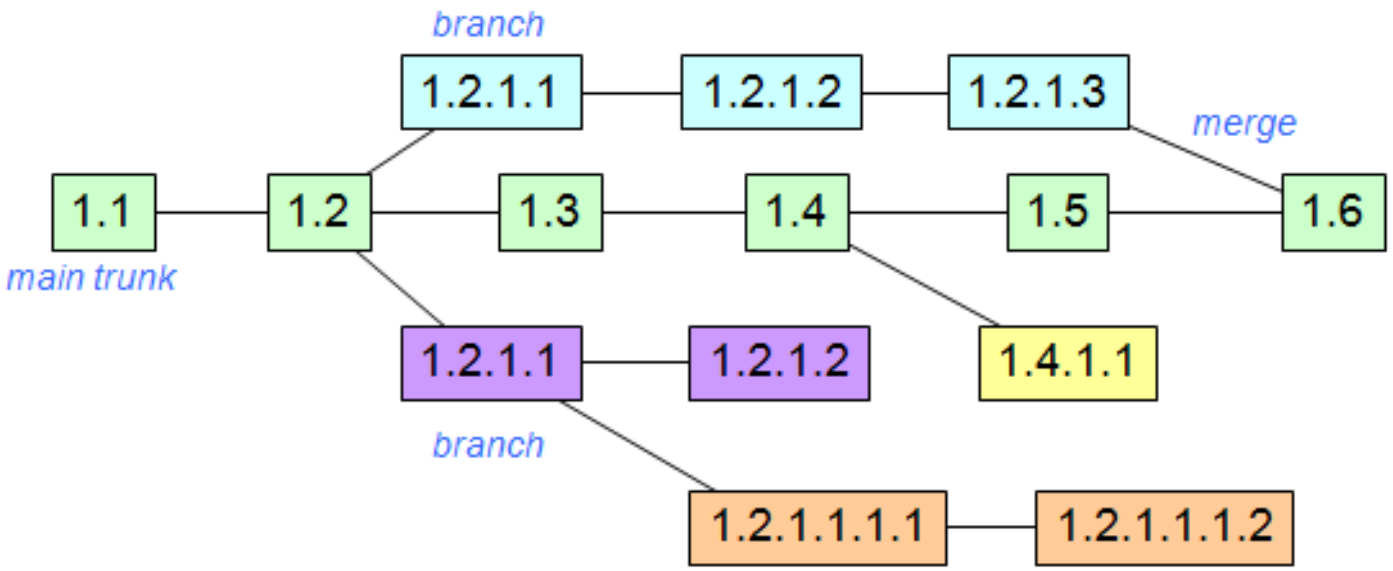
Hierarchy



Client server model

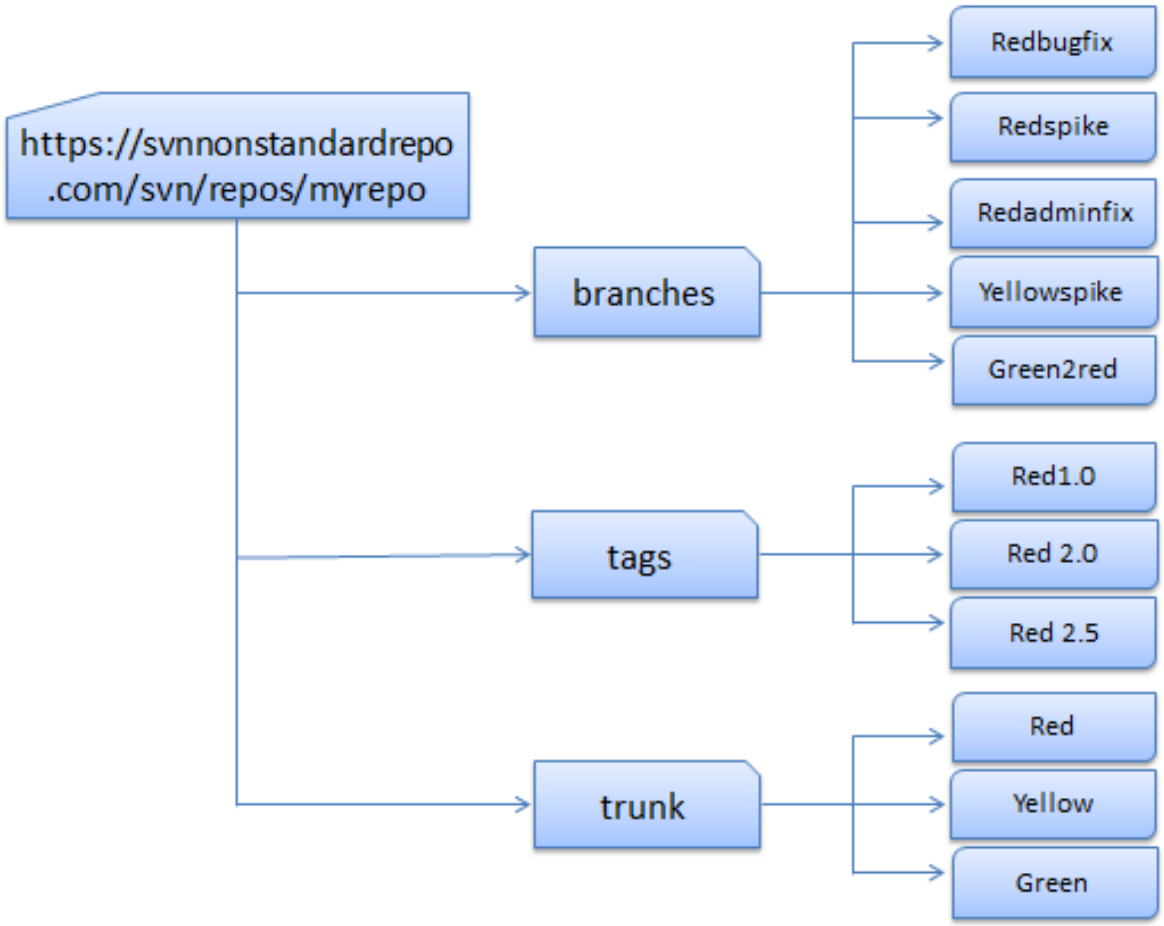
SVN - Steps

- File history tracked by revision numbers
- Each revision has a log entry



SVN branch and trunk

Branches and Tags



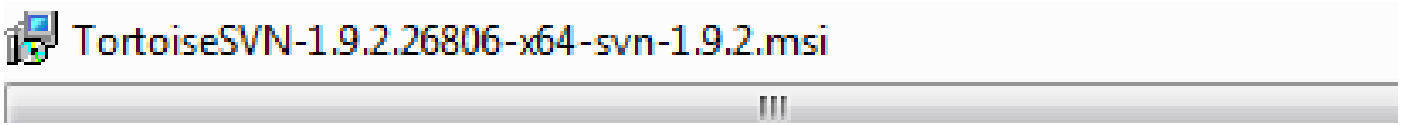
SVN repo with branches, trunk, tags

Sample Commands

- $ svn checkout [URL]
- $ svn add [file/directory]
- $ svn delete
- $ svn commit
- $ svn update
- $ svn mv(move)
- $ svn help
- $ svn diff
- $ svn revert
- $ svn list

SVN Installation

- Download TortoiseSVN from http://tortoisesvn.net/downloads.html
- Choose your version 32bit or 64bit OS.
- Double click TortoiseSVN-1.9.2.26806-x64-svn-1.9.2.msi to install it.

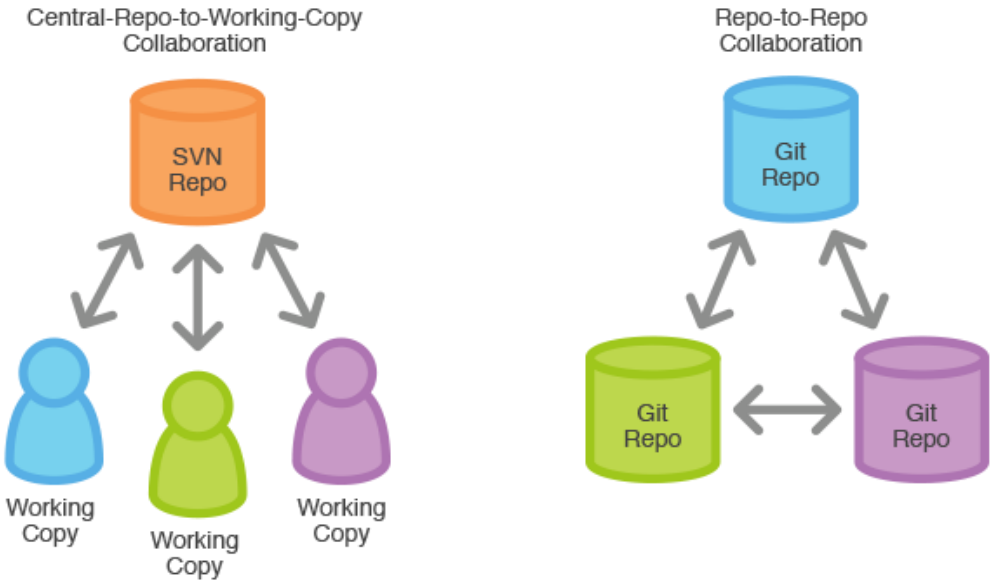


SVN Demo

- Create and manage a file version on SVN repository

Git and SVN comparison

- ✓ Git operates more faster
- ✓ Storage at database
- ✓ Git more advantages than SVN
- ✓ Distributed approach



Centralized vs Distributed

Summary

- Version Control
  - Method to centrally store files
  - Log who, when, where, what
  - Also called as revision control / source control
- Three types of VCS:
  - Localized
  - Centralized (SVN)
  - Distributed (GIT)
- GIT vs SVN
  - Git operations more faster
  - Storage at database
  - Distributed approach

References

- Version Control
  - https://en.wikipedia.org/wiki/Revision_control
- SVN
  - http://tortoisesvn.net
- GIT
  - https://git-scm.com/downloads

Thank you!

Q & A

Games

Games

- Clone ‘test’ repo from git@20.203.6.65:test to your local machine.
- Create a new file in the cloned repo.
- Add and commit to remote repo.
- Push file to Git server.

Revision History

Date	Version	Description	Updated by	Reviewed and Approved By
Aug 10, 2015	0.5	Ready for review	Thien Tran	Quang Tran
Nov 26, 2015	1.0	Ready for review	Quan Vuong	Quang Tran


CSC
BUSINESS SOLUTIONS
TECHNOLOGY
OUTSOURCING