



LEIDEN UNIVERSITY MEDICAL CENTER

# Introduction to Version Control

**Jeroen F. J. Laros**

**Leiden Genome Technology Center**

**Department of Human Genetics**

**Center for Human and Clinical Genetics**



## *Version control*

*The management of changes to documents, computer programs, large web sites, and other collections of information.*  
— Wikipedia.

## *Version control*

*The management of changes to documents, computer programs, large web sites, and other collections of information.*  
— Wikipedia.

General features:

- Keeping track of your files in an orderly manner.
  - Hiding old versions.
  - Recording who made changes and when.
- Enables collaboration.

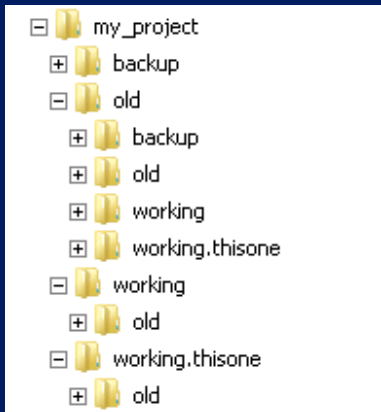
*Version control*

Figure 1: “I have my own system.”

## *Why should I use it?*

For a single user:

- Revert files to a previous state.
- Revert the entire project back to a previous state.
- Review changes made over time.
- Backup.

## *Why should I use it?*

For a single user:

- Revert files to a previous state.
- Revert the entire project back to a previous state.
- Review changes made over time.
- Backup.

For multiple users:

- A reliable way to share files between people/computers.
- Allow multiple people working on the same project at the same time.
- Conflict resolution.
- See who made which changes at which time.

## *Why should I not use it?*

A list of common excuses:

- It is too much work.
- I have my own system.
- I am the only one working on this project.
- This code will not be used by anyone else.
- The bugs can be tracked forever.
- ...

## *Why should I not use it?*

A list of common excuses:

- It is too much work.
- I have my own system.
- I am the only one working on this project.
- This code will not be used by anyone else.
- The bugs can be tracked forever.
- ...

Eventually leading to:

- I'm too busy rewriting the code I accidentally deleted.



## *Local repository*

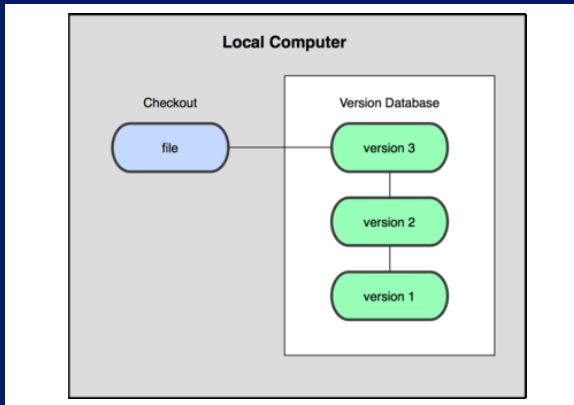


Figure 2: Local version control diagram.

## *Central repository*

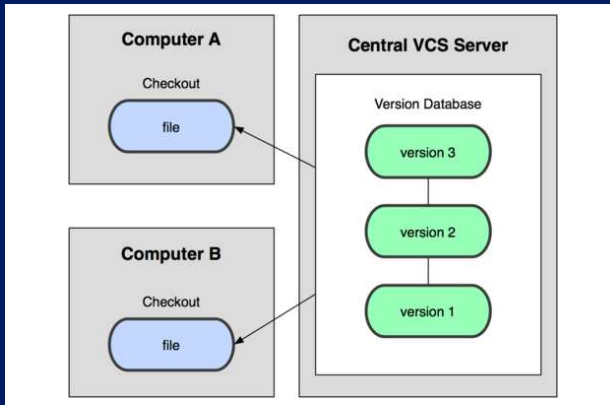


Figure 3: Centralised version control diagram.

## *Distributed repositories*

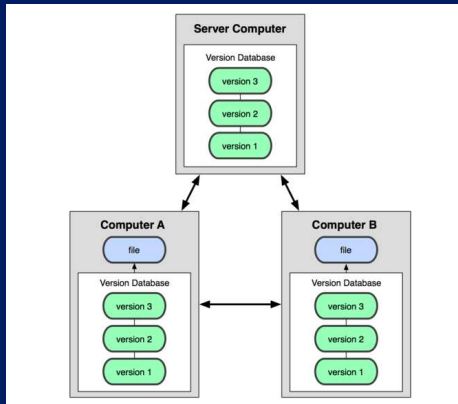


Figure 4: Distributed version control diagram.

## *The name*

*Git (n): A person who is deemed to be despicable or contemptible. — WordNet.*

## *The name*

*Git (n): A person who is deemed to be despicable or contemptible. — WordNet.*

*I'm an egotistical bastard, and I name all my projects after myself. First "Linux", now "git". — Linus Torvalds.*

## *History*

Designed to replace the commercial package *BitKeeper*.

- Speed.
- Simple design.
- Strong support for non-linear development (thousands of parallel branches).
- Fully distributed.
- Able to handle large projects like the Linux kernel efficiently (speed and data size).

## *A lot of choices*

GitHub.

- Only free for open source projects.

SourceForge, BitBucket, Gitorious, ...

GitLab.

- Issue tracking.
- Wiki.
- “Project wall”.
- Snippet.
- User profiles.

<https://github.com/>





## *Outline*

We are going to:

- Create and configure a user account on the GitLab server.
- Upload an **ssh-key** to work with Git.

<https://git.lumc.nl/>



## Acknowledgements:

Martijn Vermaat  
Wibowo Arindrarto  
Szymon Kielbasa  
Zuotian Tatum

<https://git.lumc.nl/humgen/gitcourse>