

# 8.3 – Introduzione al controllo di versione

## Git



# Agenda

- Concetto di controllo di versione (versioning)
  - Motivazioni
  - Vantaggi
- Schemi di controllo di versione
- Git



# Evoluzione del software

- Attività (sviluppatore singolo):
  - Nuove funzionalità (aggiungere software)
  - Modifica delle funzionalità esistenti (modificare software)
  - Eliminazione delle funzionalità (cancellare software)



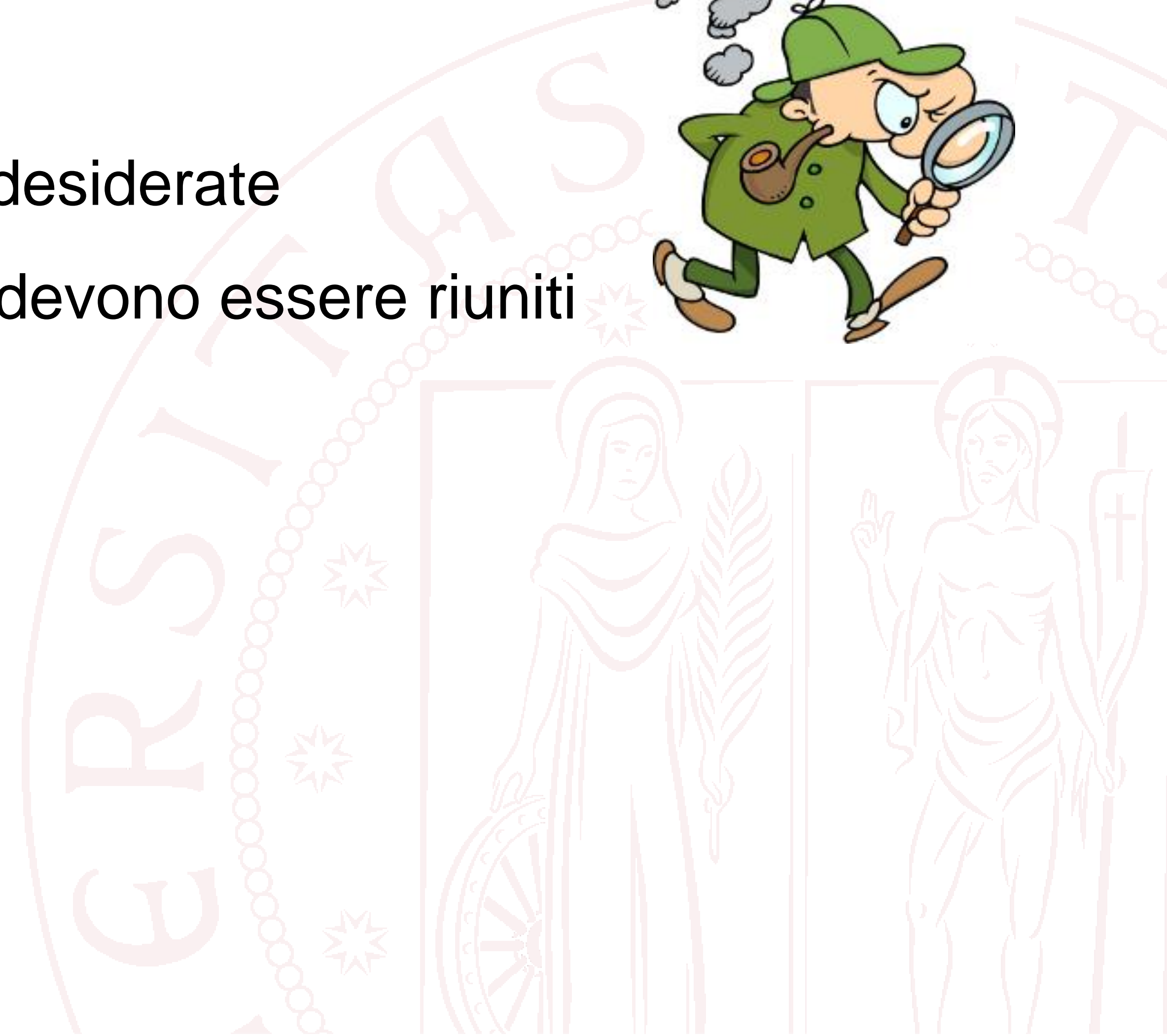
# Evoluzione del software

- Attività (sviluppatori multipli):
  - Creazioni di diversi moduli in parallelo
  - Modifiche nello stesso modulo in punti diversi
  - Modifiche nello stesso modulo nello stesso punto (conflitto)
  - Merge delle modifiche
  - Utilizzo di software scritto da altri sviluppatori



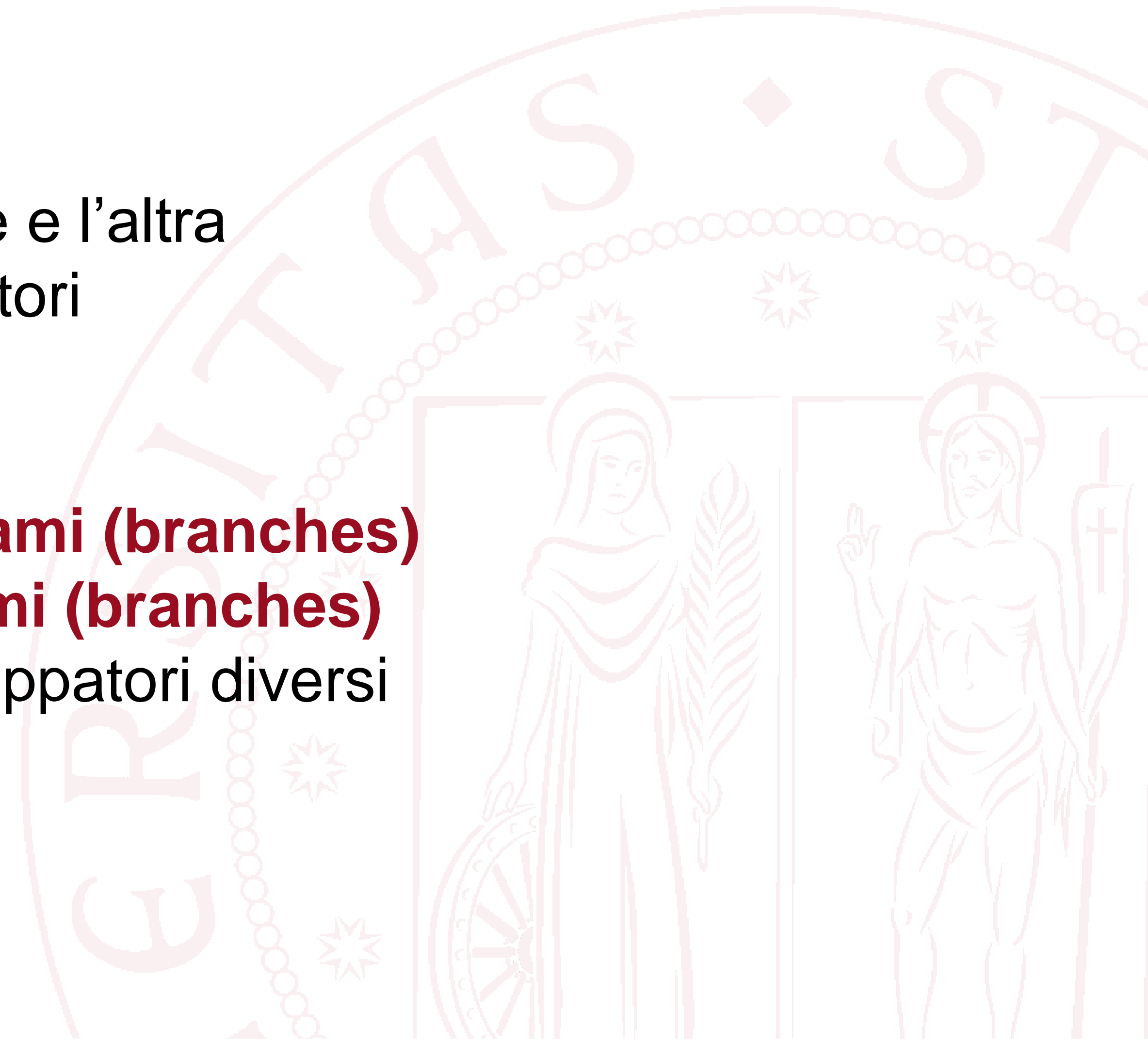
# Tracciamento del software

- Ha senso tener traccia delle modifiche?
  - Salvare vari "punti di ripristino" (cit.) stabili
  - Rimozione di modifiche stabili ma non più desiderate
  - Evoluzione non sempre lineare (rami) che devono essere riuniti
  - Condivisione del codice e delle modifiche
  - ...



# Tracciamento del software

- Che attività richiede?
- **Base:**
  - Salvare le varie versioni
  - Evidenziare le modifiche tra una versione e l'altra
  - Distribuire le modifiche a tutti gli sviluppatori
- **Avanzate:**
  - Suddivisione dello sviluppo software in **rami (branches)**
  - Condivisione delle modifiche tra i vari **rami (branches)**
  - Merge intelligente delle modifiche di sviluppatori diversi
  - Tool di lettura della storia passata



# Tracciamento del software

- **Soluzione 1:** salvataggio di varie copie
  - Ogni salvataggio contiene una copia completa
  - Calcolo delle modifiche con diff/meld
  - Applicazione manuale delle modifiche di uno sviluppatore sul codice degli altri sviluppatori
  - Facile tornare indietro, non andare avanti ☺
- Poco versatile, molto overhead, non fornisce gli strumenti avanzati



# Tracciamento del software

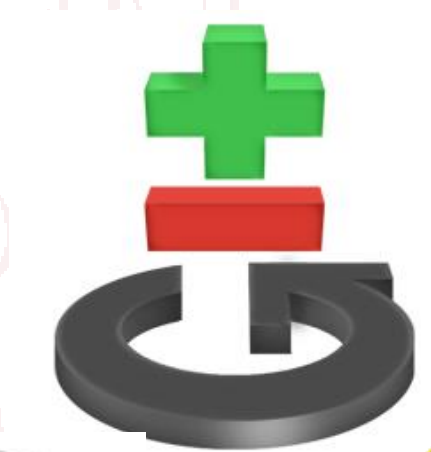
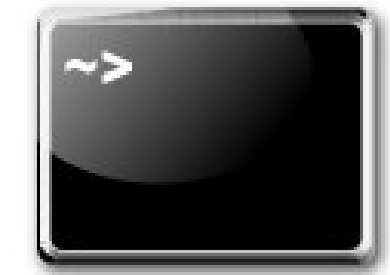
- **Soluzione 2:** software di controllo di versione
  - Implementa in maniera automatica gli strumenti che abbiamo visto
- SCM: software configuration management
  - "Is the task of controlling changes in the software. Configuration management practices include revision control" (Wikipedia)
- Revision control – version control – source control



# Controllo di versione

- Diverse soluzioni:

- Diff/meld (manuale)
- CVS/SVN (controllo di versione centralizzato)
- GIT/Mercurial/Bazaar (controllo di versione distribuito)



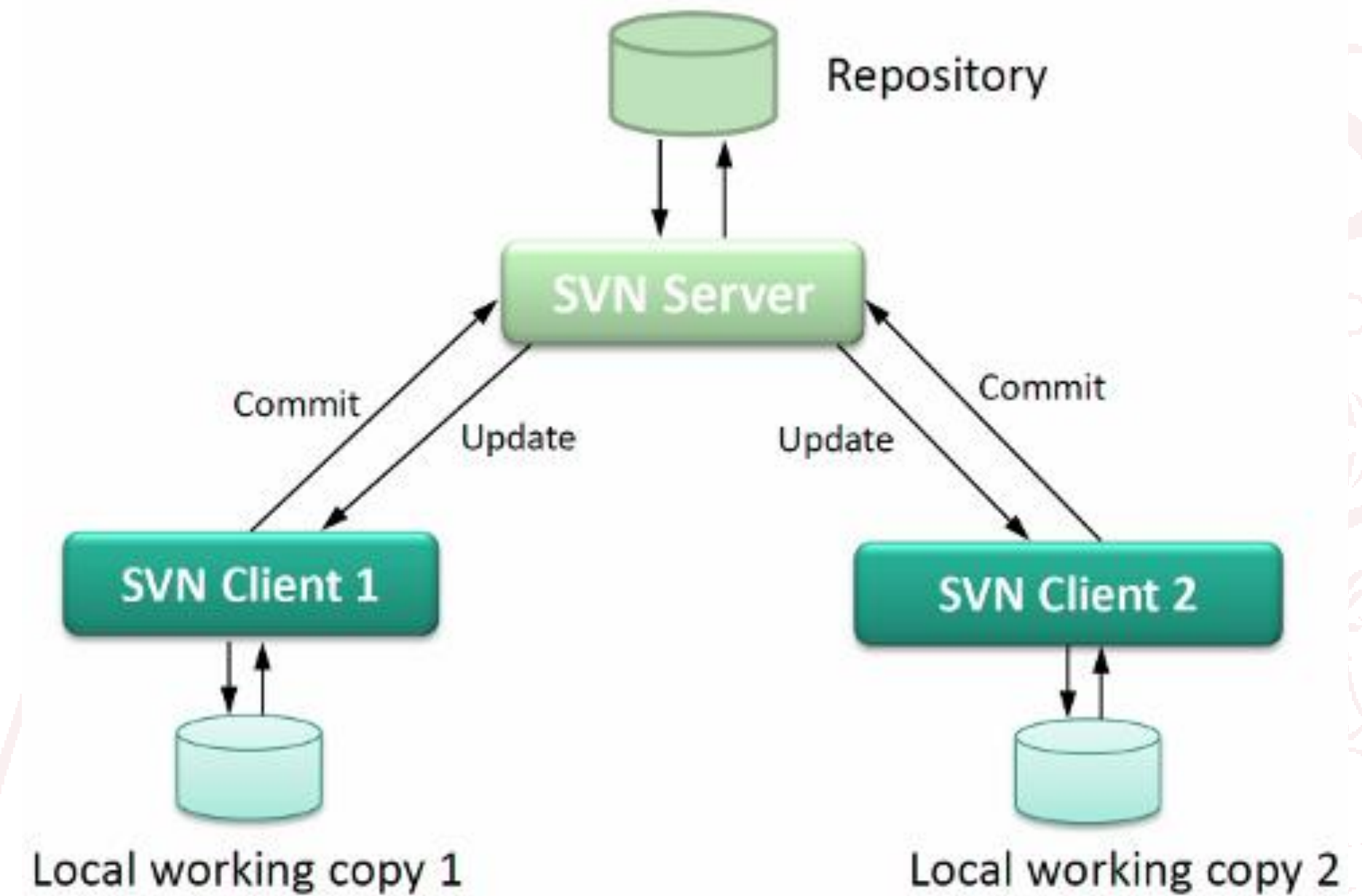
mercurial



Bazaar

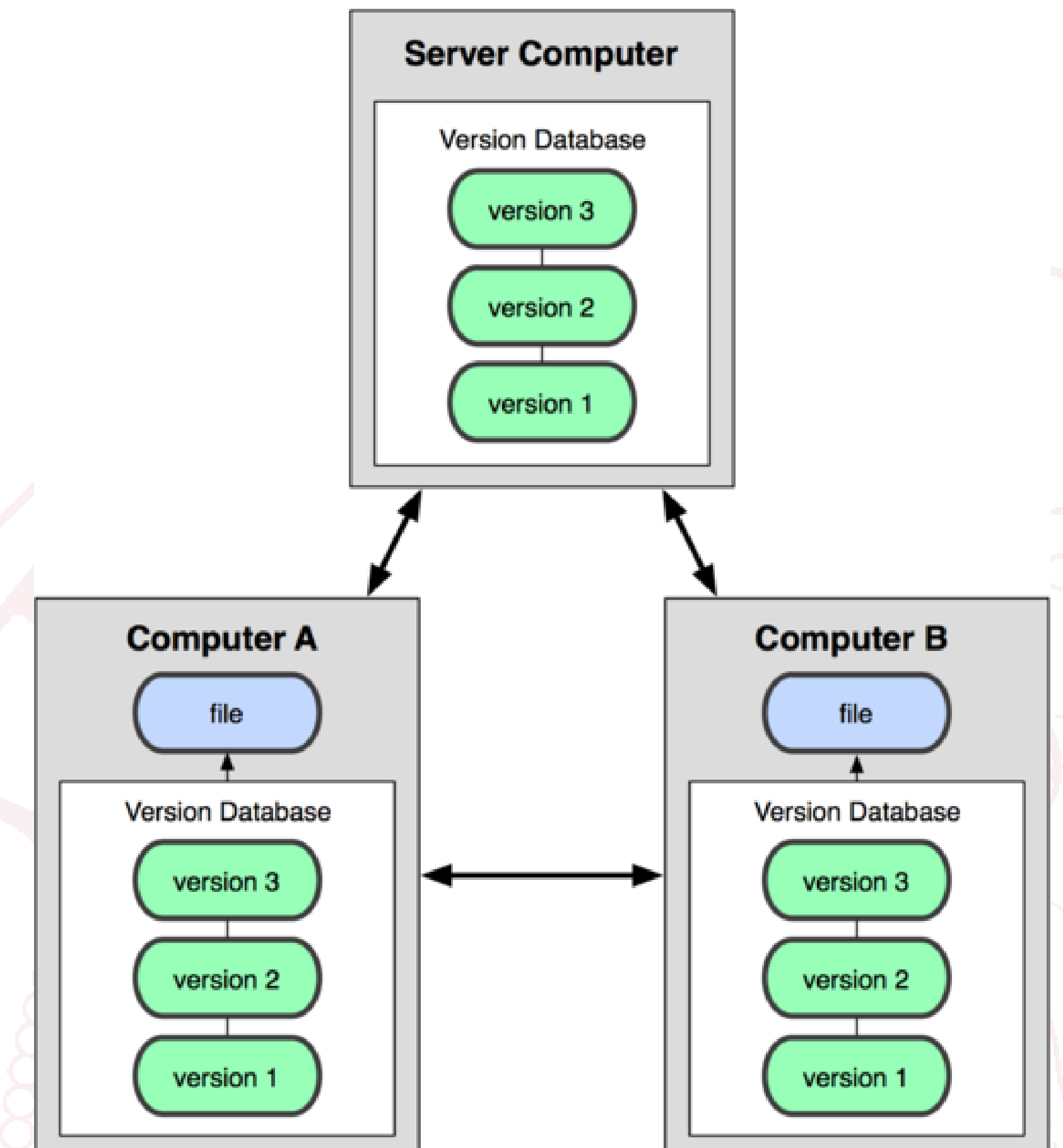
# Modello centralizzato (e.g., SVN)

- Server (repository)
  - Contiene tutta la storia e i rami
  - Riceve le modifiche dai client
- Client
  - Il client scarica una versione
  - Modifiche in locale
  - Upload delle modifiche
- La stratificazione del codice avviene mediante l'upload sul server di una nuova versione



# Modello distribuito (e.g., Git)

- Scompare la distinzione tra client e server
- Esistono solo alberi che sono pari (repository)
- Tutta la storia passata è presente in locale
- Stratificazione locale
- Sincronizzazione tra gli alberi





- Controllo di versione ideato da Linus Torvalds
- Usato per il kernel Linux, git (!), eclipse, gnome, kde, Qt, android, debian, X.org, ...
- Per linux, windows e mac!
- <http://git-scm.com/>
- Prevalentemente a riga di comando
  - Ma esistono tool di ispezione



Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is [easy to learn](#) and has a [tiny footprint with lightning fast performance](#). It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like [cheap local branching](#), convenient [staging areas](#), and [multiple workflows](#).



### About

The advantages of Git compared to other source control systems.



### Documentation

Command reference pages, Pro Git book content, videos and other material.



### Downloads

GUI clients and binary releases for all major platforms.



### Community

Get involved! Bug reporting, mailing list, chat, development and more.



**Pro Git** by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).



[Windows GUIs](#)



[Tarballs](#)



[Mac Build](#)



[Source Code](#)

### Companies & Projects Using Git

Google

FACEBOOK

Microsoft



LinkedIn

NETFLIX



PostgreSQL



About this site

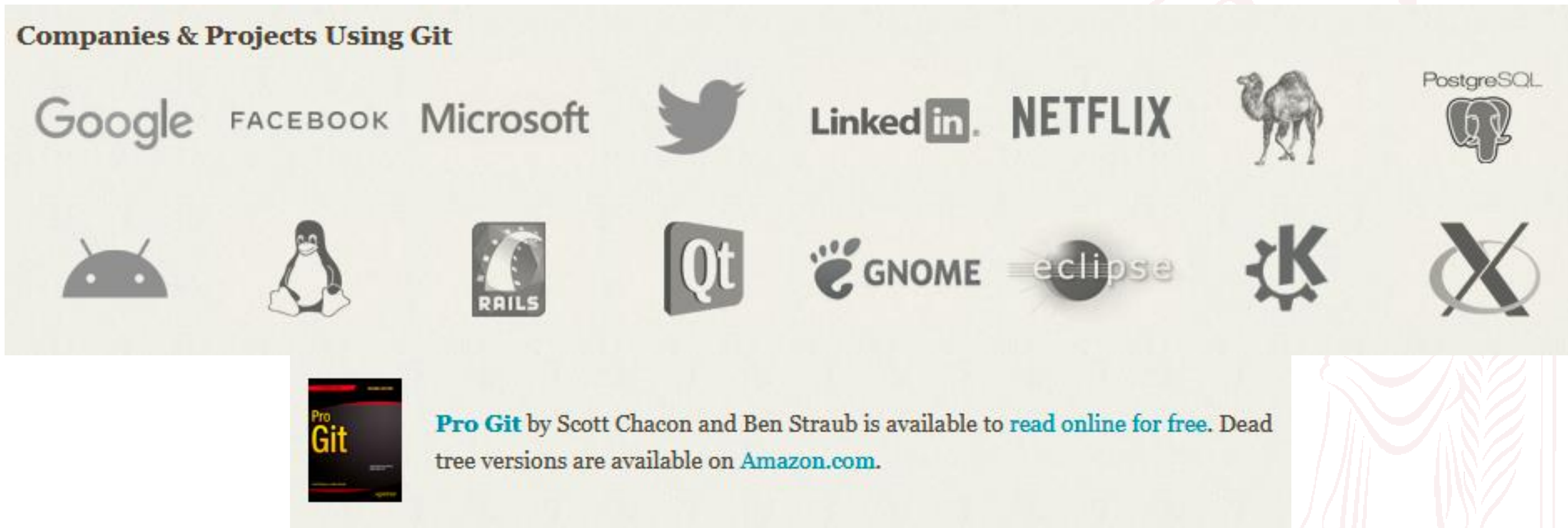
Patches, suggestions, and comments are welcome.

Git is a member of [Software Freedom Conservancy](#)



# Git

- Focus



# Interfaccia

- Linux:
  - shell (bash)
  - Tool per ispezione
- Windows:
  - Git for Windows
  - git bash (command line)
  - Git GUI
  - Shell integration
  - IDE integration



The screenshot shows the 'git for windows' website. At the top, there's a navigation bar with links for 'FAQ', 'REPOSITORY', and 'MAILING LIST'. Below the navigation bar, the version 'VERSION 2.29.2.2' is displayed. The main header features the Git logo (a red diamond with a white branching diagram) and the text 'We bring the awesome Git SCM to Windows'. Below this, there are two buttons: 'Download' and 'Contribute'. A blue banner below the header reads 'Tools & Features'. The main content area has a blue background and contains three sections: 'Git for Windows focuses on offering a lightweight, native set of tools that bring the full feature set of the Git SCM to Windows while providing appropriate user interfaces for experienced Git users and novices alike.', 'Git BASH' (describing the BASH emulation), and 'Git GUI' (describing the graphical user interface). To the right of the text, there are two screenshots: the top one shows a terminal window with Git commands and output, and the bottom one shows the Git GUI application interface.

git for windows  
VERSION 2.29.2.2

We bring the awesome  
**Git** SCM to Windows

Download Contribute

## Tools & Features

Git for Windows focuses on offering a lightweight, native set of tools that bring the full feature set of the [Git SCM](#) to Windows while providing appropriate user interfaces for experienced Git users and novices alike.

### Git BASH

Git for Windows provides a BASH emulation used to run Git from the command line. \*NIX users should feel right at home, as the BASH emulation behaves just like the "git" command in LINUX and UNIX environments.

### Git GUI

As Windows users commonly expect graphical user interfaces, Git for Windows also provides the Git GUI, a powerful alternative to Git BASH, offering a graphical version of just about every Git command line function, as well as comprehensive visual diff tools.

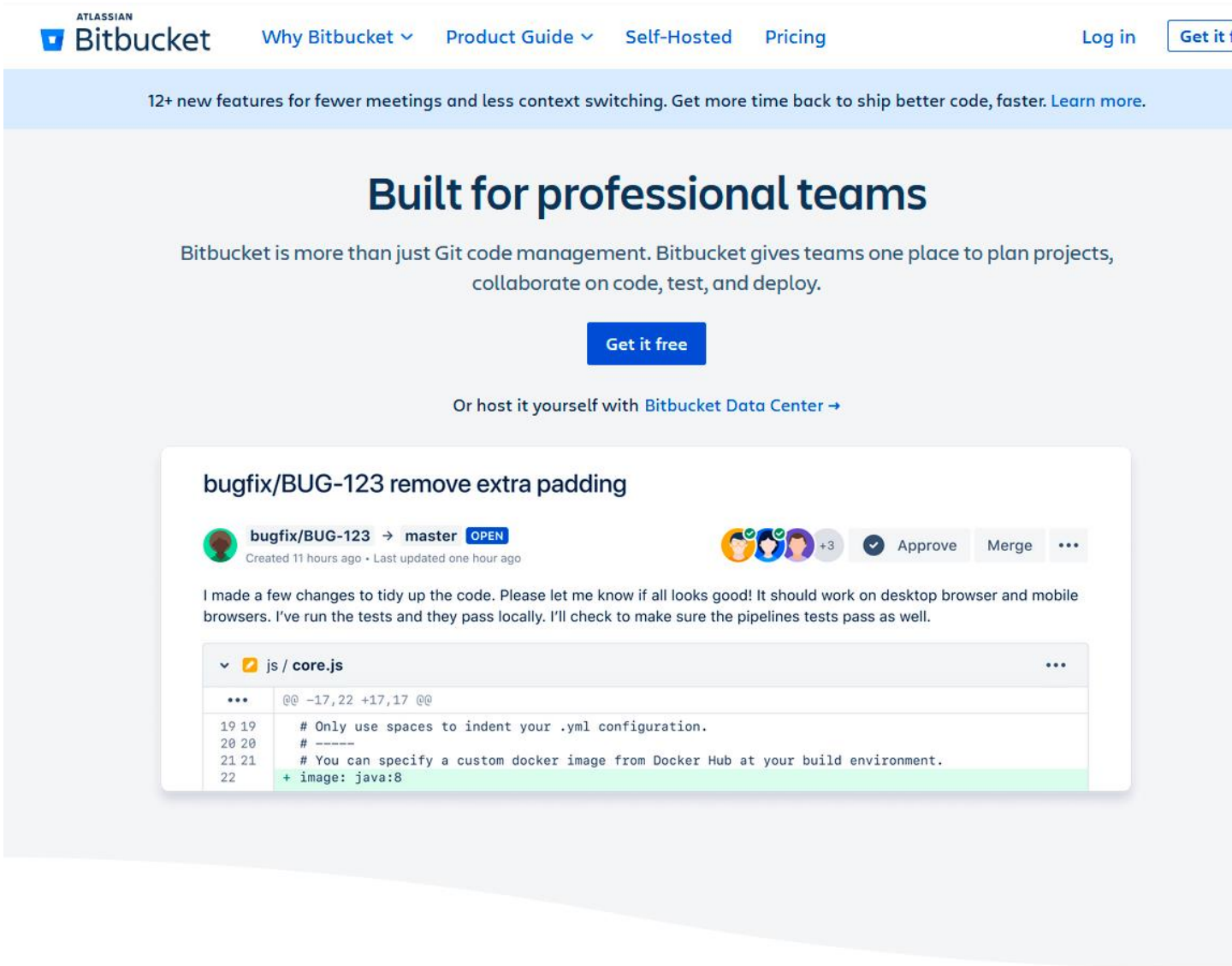
### Shell Integration

Simply right-click on a folder in Windows Explorer to access the BASH or GUI.



# Interfaccia web

- Interfacce web spesso usate per:
  - Gestire i repository
  - Visualizzare i repository



#### Free unlimited private repositories

Free for small teams under 5 and priced to scale with Standard (\$3/user/mo) or Premium (\$6/user/mo) plans.



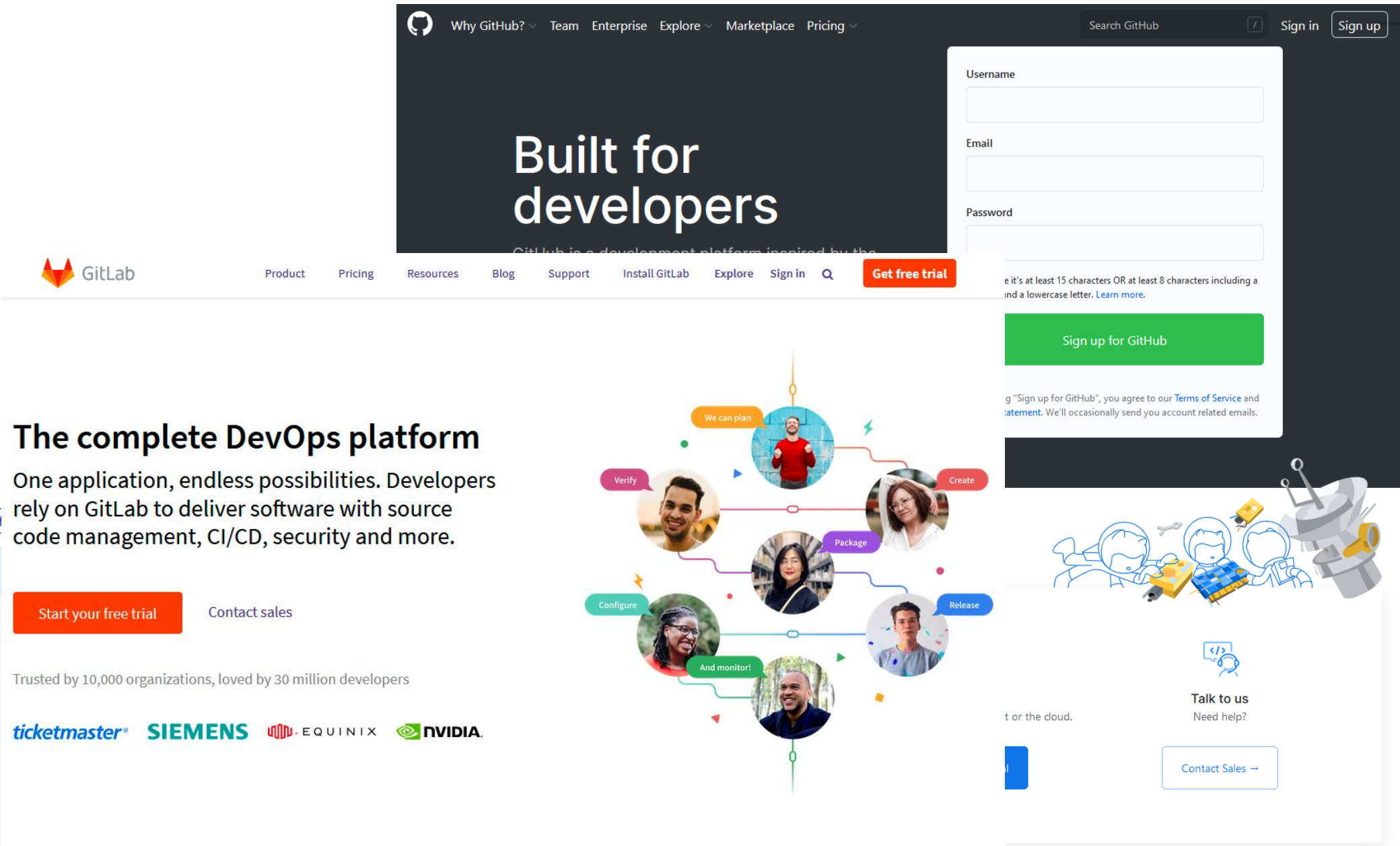
#### Best-in-class Jira & Trello integration

Keep your projects organized by creating Bitbucket branches right from Jira issues or Trello cards.



#### Built-in Continuous Delivery

Build, test and deploy with integrated CI/CD. Benefit from configuration as code and fast feedback loops.



Better collaboration, more automation, less complexity.



#### Single source of truth

Manage projects, not tools. With GitLab, you get a complete DevOps Platform delivered as a single application—one interface, one conversation thread, one data store, zero headaches.



#### Continuous everything

Bridge Dev and Ops once and for all. Our industry-leading CI/CD empowers all teams to work together efficiently. Powerful, scalable, end-to-end automation is possible with GitLab.



#### Real-time security

See everything that matters. Built-in everything brings automated security, code quality, and vulnerability management. With GitLab, tight governance and control never slow down DevOps speed.

# Recap

- Tracciare lo sviluppo del software
- Sistemi di controllo di versione
  - Centralizzati
  - Distribuiti
- Git

