Version Control and Git

Philipp Roebrock

Institute for Photonics and ICP (IPI)

September 17, 2019

Fachhochschule Graubünden
University of Applied Sciences

Contents I

Version Control Systems

Version Control Systems:

Basics

Exercise

What is a *version control system*?

Have you used one before?

What is a version control system?

A version control system provides:

- Logging: Who changed what? Why? When?
- Who signed off?
- Archiving, Integrity
- Collaboration
- Branching = Parallel work on multiple interdependent versions (stable, develop, feature, ...)

What is a version control system?

Search

Diff | Old version | New version | Lines of context: 3 | Ignore space change | Line diff

```
Author: Philipp Roebrock <philipp.roebrock@htwchur.ch>  2019-07-08 16:33:28
Committer: Philipp Roebrock <philipp.roebrock@htwchur.ch>  2019-07-08 16:33:28
Parent: ea142162df7145591db1ee05e18f1a6662c7b48a (Adds Tremaux class)
Child:  b8e8505f7146b6a4e0b5ca2377106275861575ae (Fixes Tremaux algorithm)
Branches: master, remotes/origin/master
Follows:
Precedes:

    Adds debugging capabilities

---------------------------- examples/maze/agent.cpp ----------------------------
index 8973b6d..c3f0afd 100644
@@ -50,6 +50,14 @@ Agent::~Agent()



+void Agent::show() const
+{
+        full_maze.show(real_pose);
+        std::cout << "Real pose " << real_pose << std::endl;
+}
+
+
+
 void Agent::show(Screen &screen) const
 {
         full_maze.show(screen);
```

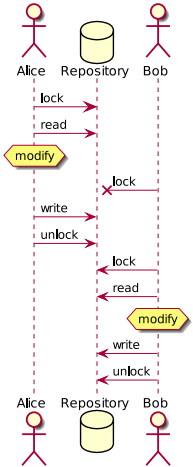
Patch | Tree

Comments
examples/maze/agent.cpp
examples/maze/agent.h
examples/maze/data/overlook.maze
examples/maze/field_type.h
examples/maze/main.cpp
examples/maze/mapping_agent.cpp
examples/maze/mapping_agent.h
examples/maze/maze.cpp
examples/maze/maze.h
examples/maze/tremaux.cpp
examples/maze/tremaux.h

Version Control Systems:

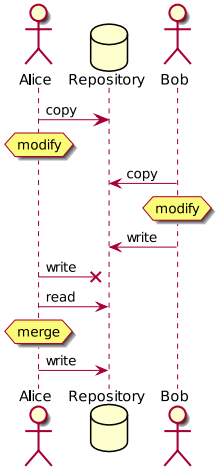
Concepts

Concept: Lock-Modify-Unlock



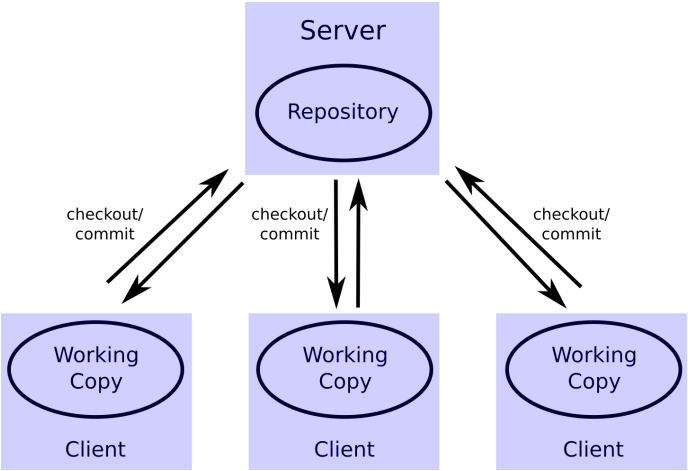
- Used by very early version control systems
- Forces serial work: Very inefficient for collaboration
- Makes still some sense for centralized version control systems and binary files

Concept: Copy-Modify-Merge

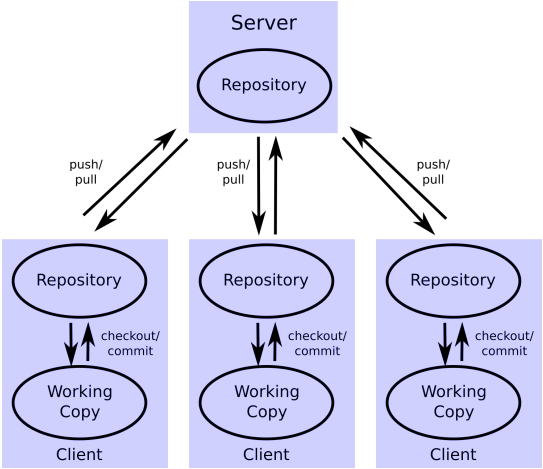


- Usually used today
- Very efficient parallel work, good collaboration
- A lot of the merges are done by software automatically
- If not, user has to manually fix these *conflicts*

Central version control systems



Distributed version control systems



See other distributed workflows.

Examples

Central:

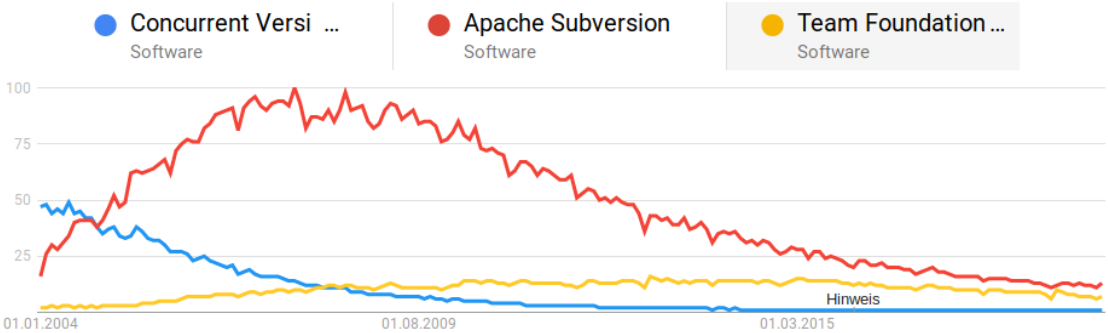
- SCCS[1], RCS[1], CVS[1] (historical reference, all outdated!)
- Subversion (SVN)[1]
- Perforce[2], ClearCase[2], Team Foundation Server (TFS)[2]

Distributed:

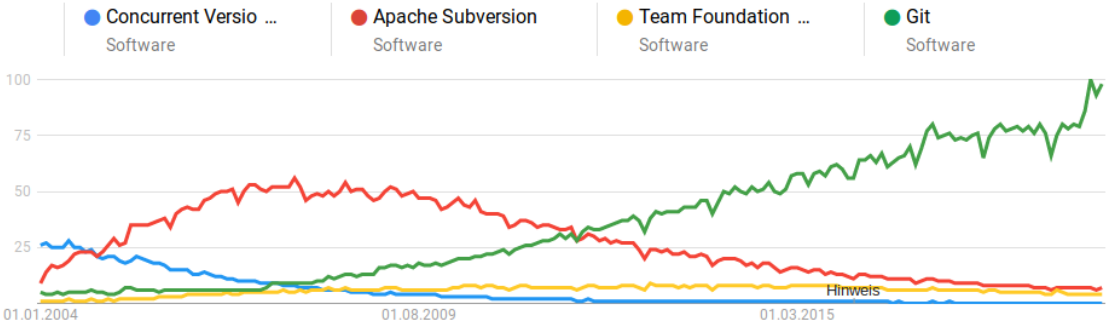
- Bitkeeper[1], Mercurial[1], Git[1]

[1] open source
[2] proprietary

Which to choose: Google trends



Which to choose: Google trends



Today, *git* clearly is the leading version control solution available!

Git

Git:

Software

Git basics

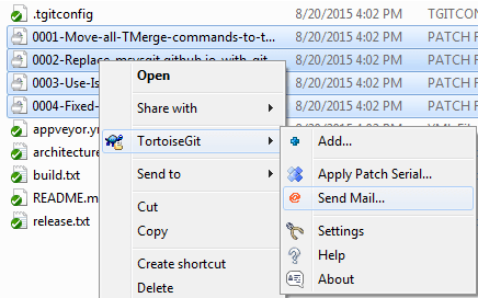
- Created 2005 by Linus Torvalds
- Website: https://git-scm.com/
- Open Source Software

Git console

```
phil@nithril2:~/Projects/fhgr/computer_science_3_lecture$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   chapters/git/git.tex

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        chapters/git/images/git_logo.png

no changes added to commit (use "git add" and/or "git commit -a")
phil@nithril2:~/Projects/fhgr/computer_science_3_lecture$ git diff
diff --git a/chapters/git/git.tex b/chapters/git/git.tex
index 0ffaf8c..453e4fb 100644
--- a/chapters/git/git.tex
+++ b/chapters/git/git.tex
@@ -14,7 +14,7 @@ A version control system provides:
 \begin{itemize}
 \item Logging: Who changed what? Why? When?
 \item Who signed off?
-\item Archiving
+\item Archiving, Integrity
 \item Collaboration
```

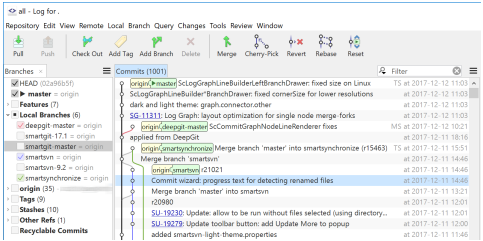
- All time classic
- For Linux and Windows
- Full control
- GUI viewer `gitk`
- https://git-scm.com/downloads

TortoiseGit



- Windows Explorer integration
- https://tortoisegit.org/

SmartGit



- Advanced Git GUI
- Commercial, free for open source development, students
- https://www.syntevo.com/smartgit/

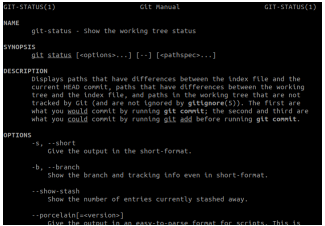
Git:

Documentation

Documentation

- Build-in help:

  `git status --help`



```
GIT-STATUS(1)                    Git Manual                    GIT-STATUS(1)

NAME
       git-status - Show the working tree status

SYNOPSIS
       git status [<options>...] [--] [<pathspec>...]

DESCRIPTION
       Displays paths that have differences between the index file and the
       current HEAD commit, paths that have differences between the working
       tree and the index file, and paths in the working tree that are not
       tracked by Git (and are not ignored by gitignore(5)). The first are
       what you would commit by running git commit; the second and third are
       what you could commit by running git add before running git commit.

OPTIONS
       -s, --short
           Give the output in the short-format.

       -b, --branch
           Show the branch and tracking info even in short-format.

       --show-stash
           Show the number of entries currently stashed away.

       --porcelain[=<version>]
           Give the output in an easy-to-parse format for scripts. This is
```

- Book from Scott Chacon and Ben Straub: Pro Git
- Book from Ryan Hodson: Ry's Git Tutorial
- Search engine keywords: git cheat sheet

Git:

Hosting

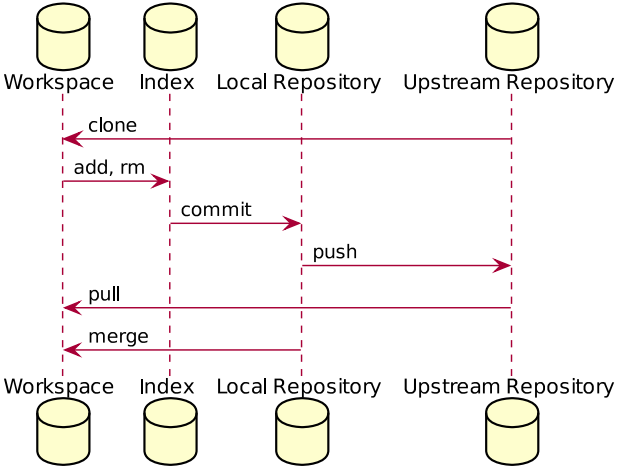
GitHub

- Most famous Git hoster
- Bought by Microsoft in 2018
- Some restrictions for non-commercial use
- Register

GitLab

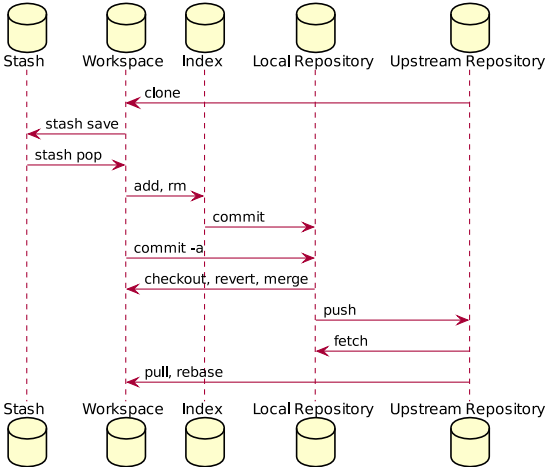
- Extensive system: Issue tracking, CI/CD, Wiki, ...
- Some restrictions for non-commercial use
- Register

Workflow (simple)



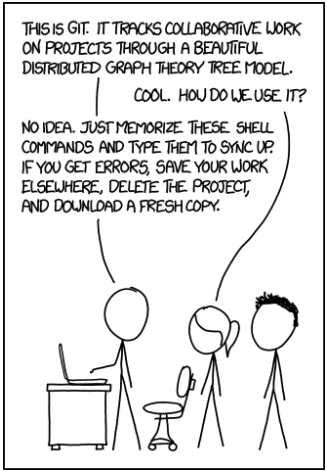
See the Git Cheatsheet!

Workflow (full)



See the Git Cheatsheet.

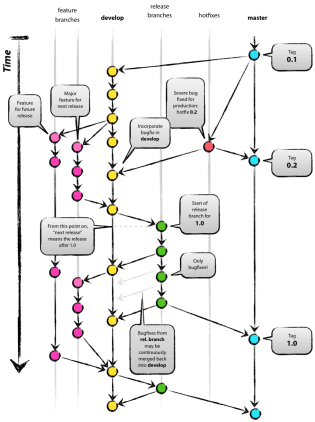
For the desperate...



Git:

Branching Model

Git Flow



- TODO