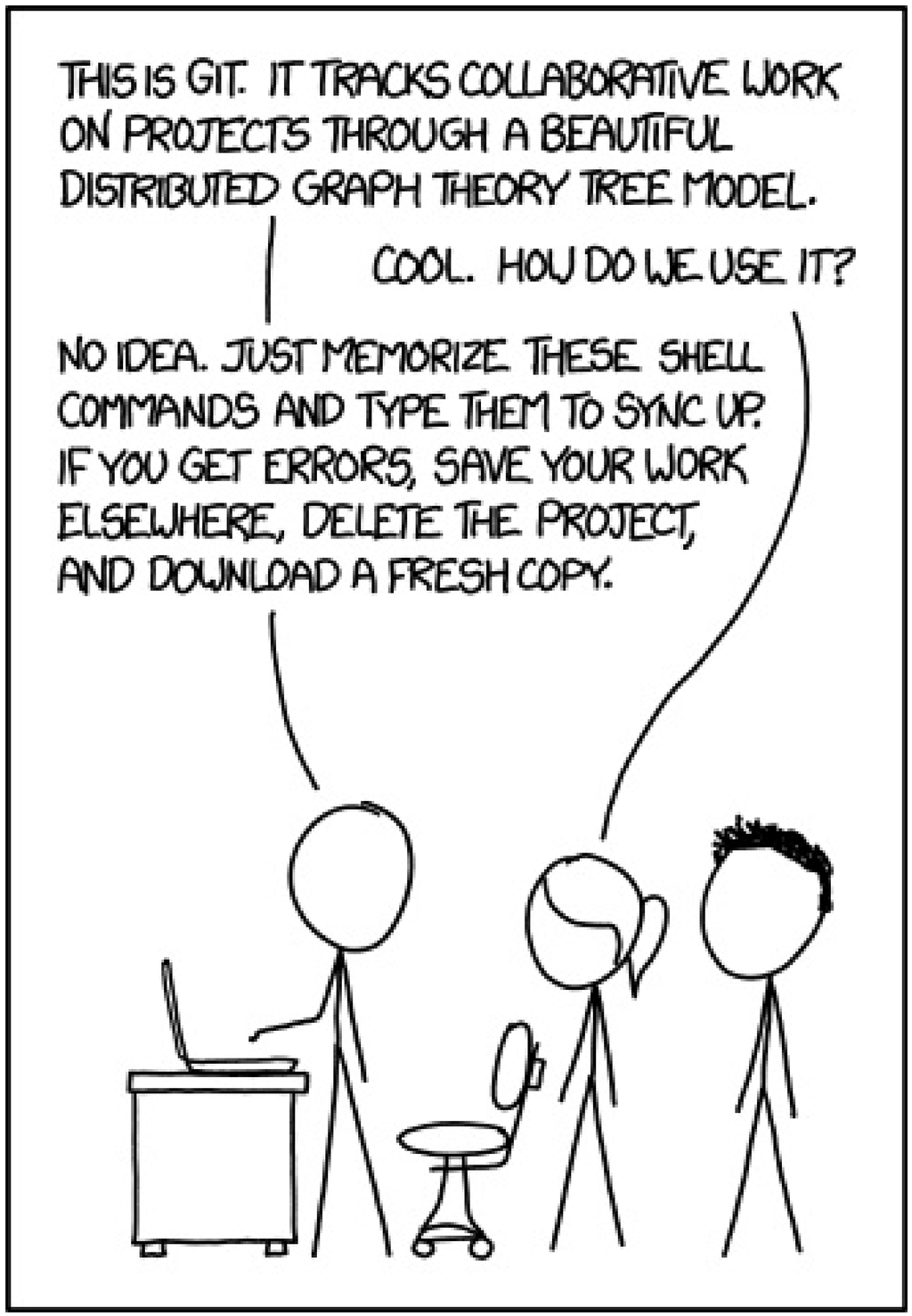
Git, The Developers Perspective



What is Git?

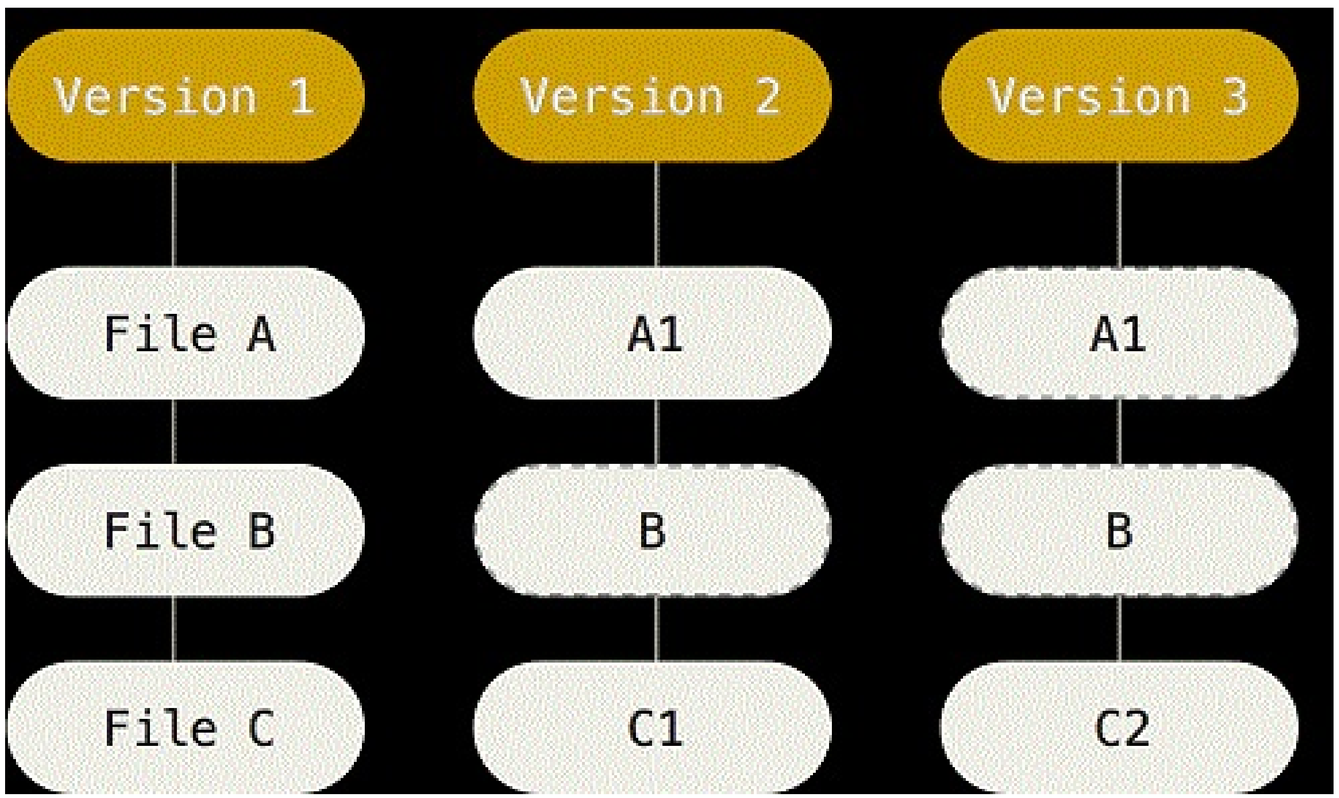
- A command line tool written by Linus Torvalds for Linux Kernel collaboration to replace a proprietary Distributed Version Control System called Bit Keeper
- Originally designed to be a fast, CLI based tool for large non-linear development
- Now has multiple GUI options available
- Fully Distributed!
- Free Book – http://git-scm.com/book

Version Control

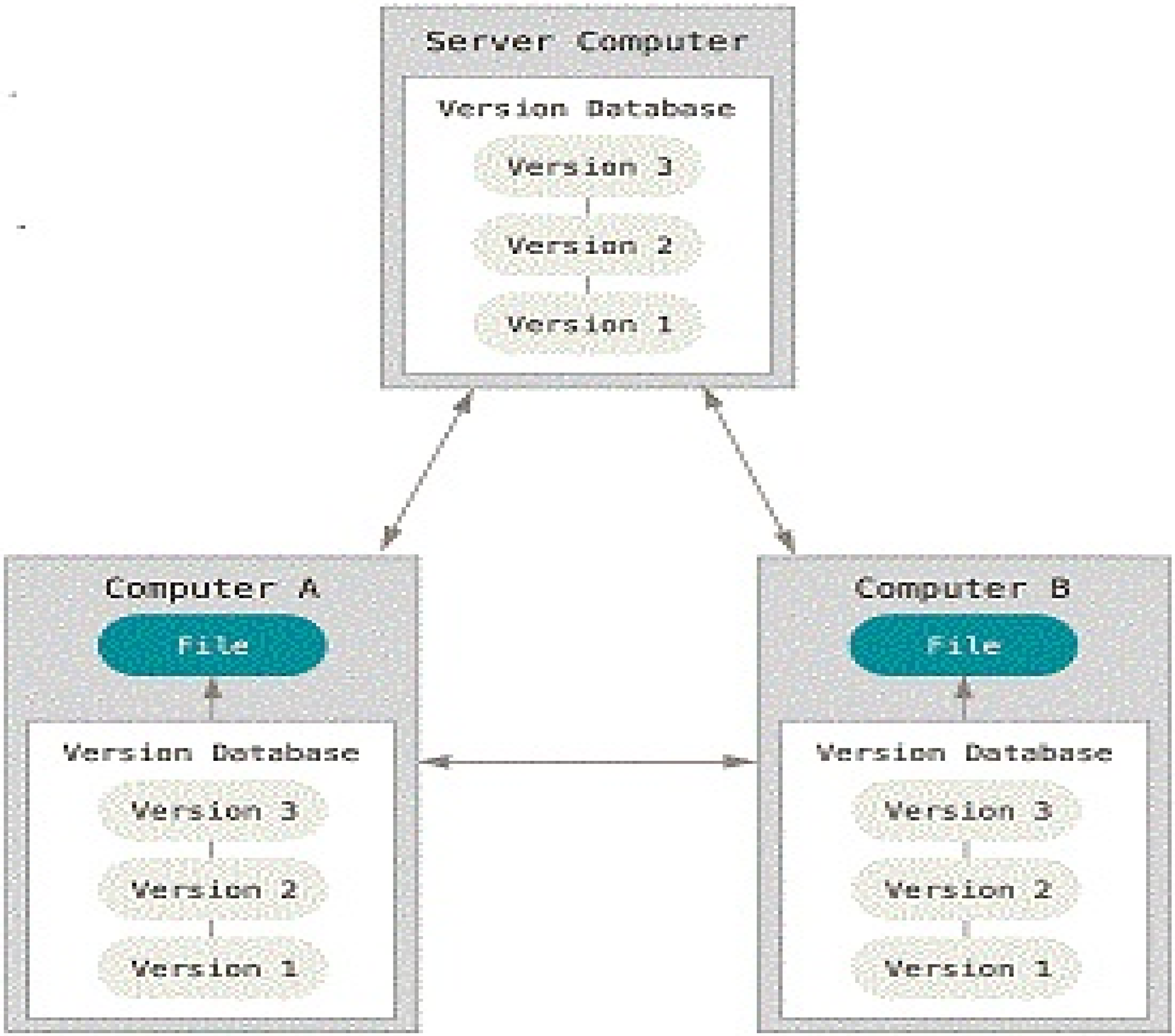
- Version control answers:
  1) WHO made a change
  2) WHAT they changed
  3) WHEN then changed it
  4) WHY it was changed
- Version Control allows:
  1) Multiple people to work on the same files and merge their changes
  2) Roll Back to previous versions

Git Basics

- Operations are local, fast, and accessible offline
- Clone/pull and pushes are remote
- Snapshots over time – Not differences – make the git repository more like a file system
- File states are modified, staged, and committed



Distributed Model



Git Command basics

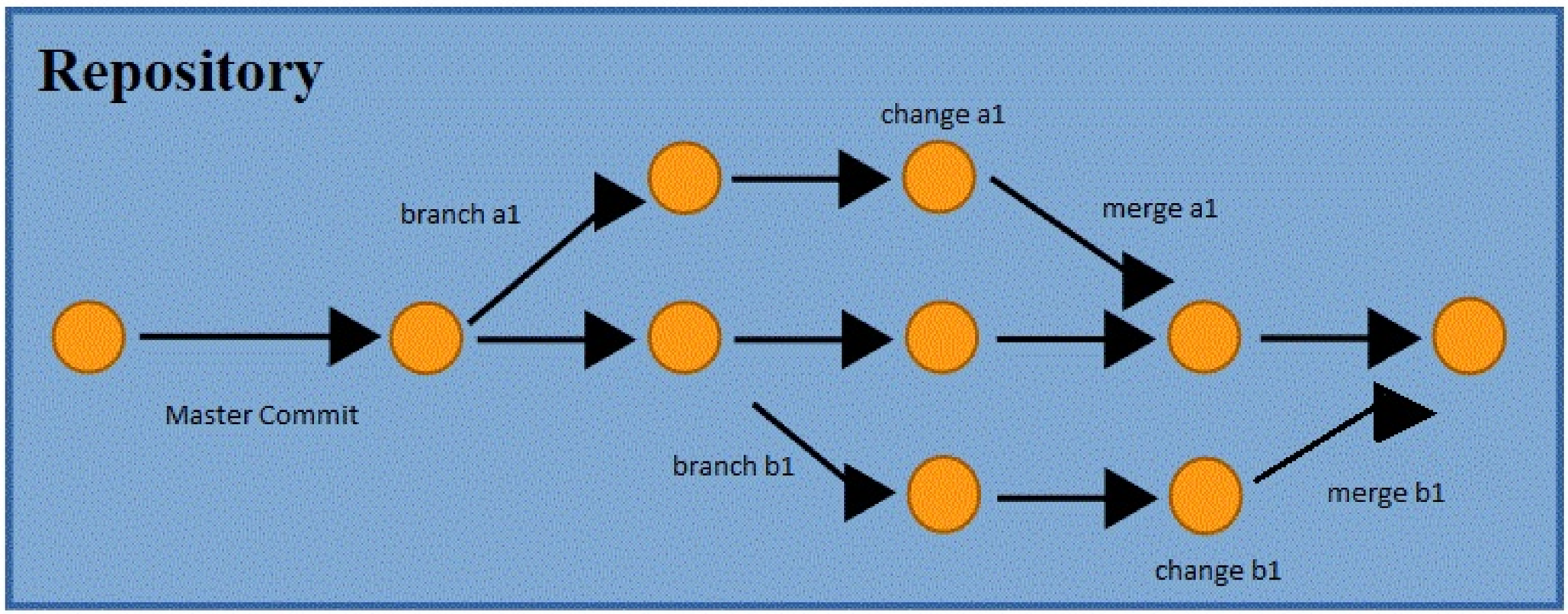
- add to add a file to be tracked or updated
- rm remove a file from being tracked
- commit create changes
- Status show the state of your files, what files are not tracked, and other useful information
- mv – move a file from one name to another
- clone to setup your local repo

Anatomy of a git Command

- Git
- …..Comand (checkout)
- ……..Flags (-b for new branch)
- ……..Arguments (branch name, branch to duplicate)
- git checkout –b sPreston_solo_1 master

Branching

- Allows you to make changes without bothering the code in the “master” line.
- When you switch branches, git resets the your working directory to look like it did when you last committed to the branch.



Basic Branch commands

- branch to create a new set of changes or list existing branches
- Git diff – show differences between a file and a target branch
- status to see your changes in your branch
- push send commited changes to remote repo
- Pull get branch updates from a remote repo

Not so basics

- Git log – show commit history to a branch. Shows who made a change, when the change was made, and what their change comment was. Also shows a commit string (ex …5e842b) that can be used to checkout the branch from that commit

- Git merge – Take the data end points and merge them together into a new commit containing all sets of data. Git automatically choses the best common ancestor to use for its merge base – saving the developer from figuring it out themselves!

- Git rebase – Instead of merging data end points, apply all changes in order to a branch and clean up branches and commits. Rebasing creates a cleaner merge history, as seen in git log. However, don’t rebase commits that exist outside your repository as it can cause headaches for people who are depending on commits that were pushed publicly.

- Merge vs Rebase is a whole discussion in its own right!

Meet our Developers, Bill and Ted

The Project, Wyld Stallyns

- Our developers are working on 3 files
- 1) The most triumphant video script (script.txt)
- 2) Eddie Van Halen Solo (solo.sh)
- 3) Music Lesson Plan
- (MusicLesson.plan)

Bill and Ted Sample

- Create new branch from master
  git checkout –b sPreston_better_solo master
- Fix bug in solo.sh
- Stage change via add
  git add solo.sh
- Check status
  git status
- Commit
  git commit
- Merge change to master
  git checkout master
  git merge tLogan_better_solo
- Delete branch
  git branch –d sPreston_better_solo
- View master log
  git checkout master
  git log

Getting into, and out of, trouble

- Merge conflicts

Auto-merging index.html CONFLICT (content): Merge conflict in index.html Automatic merge failed; fix conflicts and then commit the result.

```
<<<<<<< HEAD:index.html
<div id="footer">contact : email.support@github.com</div>
=======
<div id="footer"> please contact us at support@github.com </div>
>>>>>>> iss53:index.html
```

Getting into and out of trouble (cont)

- Accidently rm'd a file
  git reset HEAD <file> (clear transaction)
  git checkout <file>
- Nuclear option – save your files elsewhere, create a new branch (or even clone again), and copy your files back into your new branch
- Pushing to origin, rebase, push to origin
  Nuke it from orbit! AHHHHH!!!1!

Resources

- Git-scm.com/book – (source of inspiration)
- Github.io
- Git kraken
- Tortise git
- Git desktop
- Code academy git