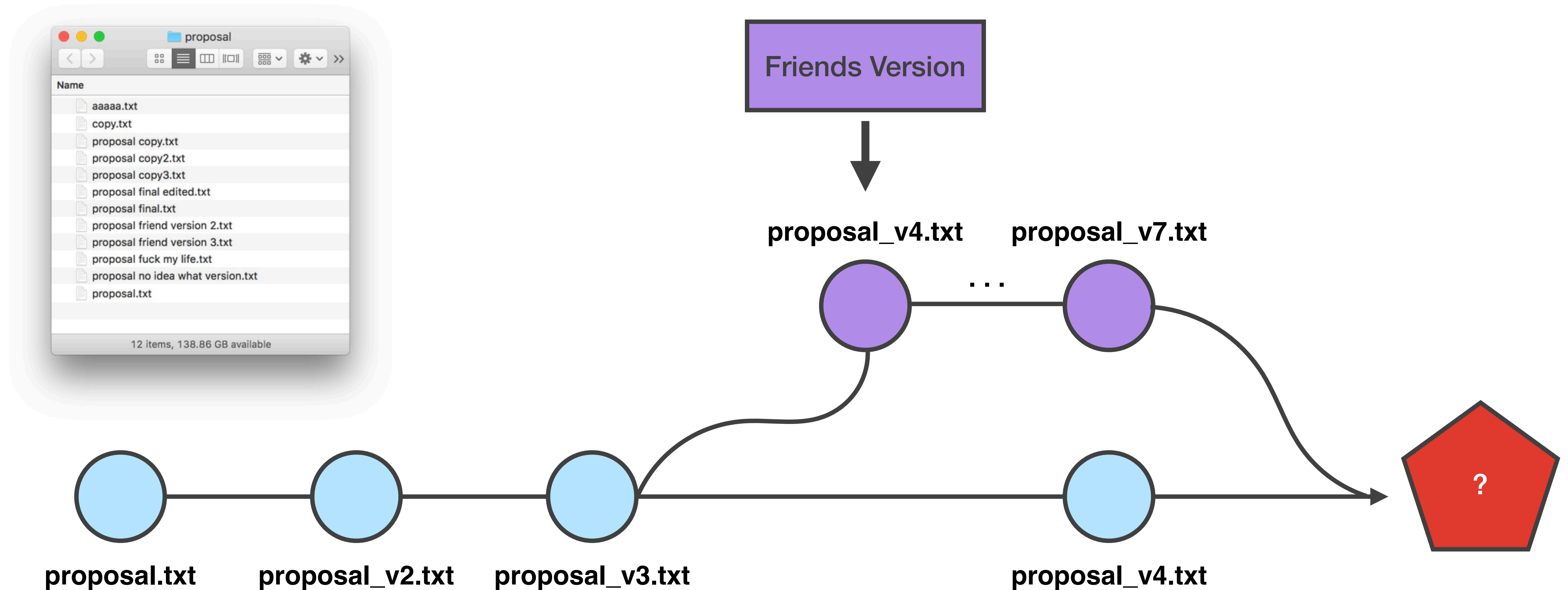


Introduction to Git & GitHub

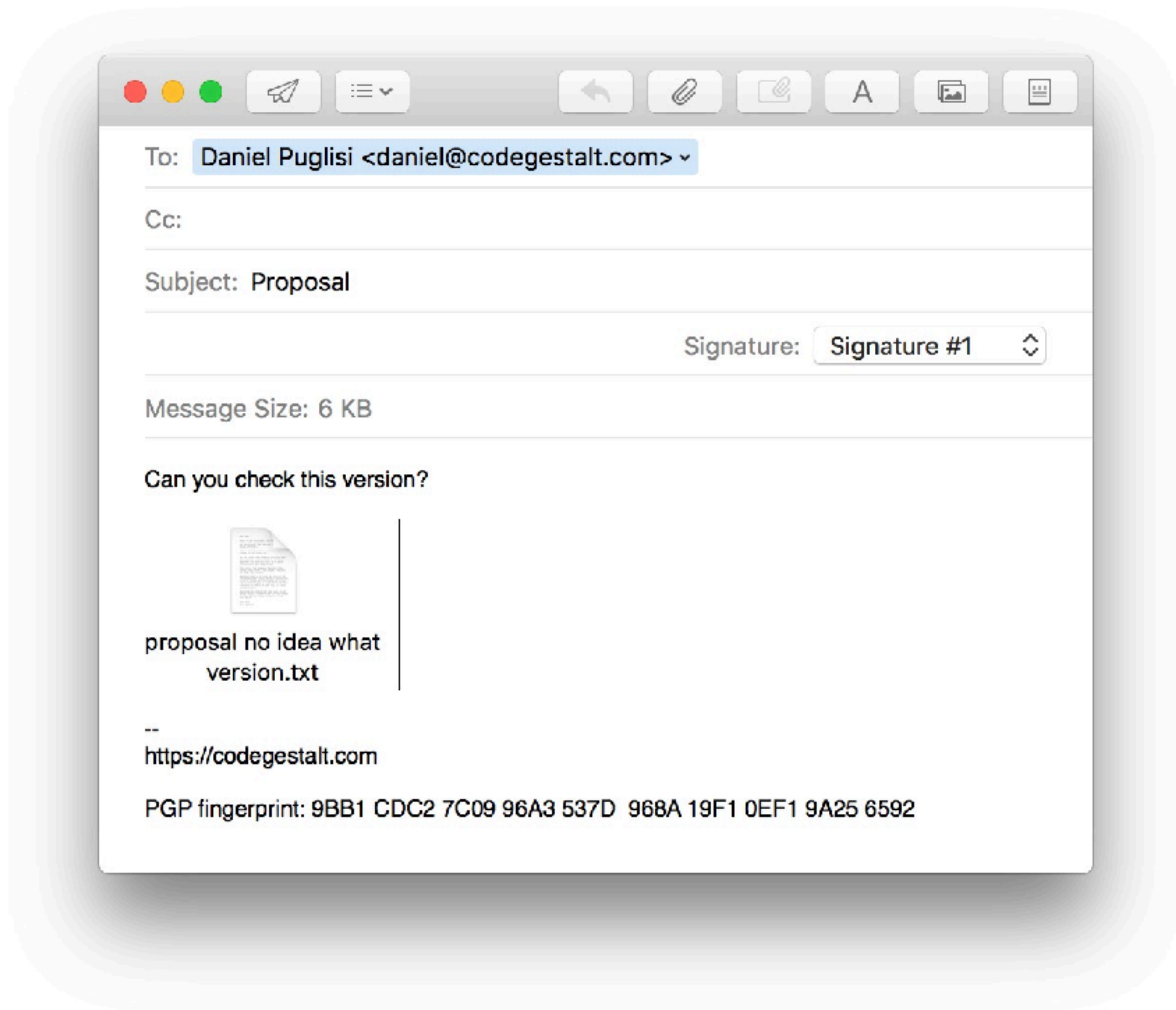
Git



The problem of classic version control.



What about collaboration?



What is git?

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.

Version control is a **system that records changes to a file or set of files over time** so that you can recall specific versions later.

If you are a graphic or web designer and want to keep every version of an image or layout, a Version Control System (VCS) is a very wise thing to use. It allows you to revert selected files back to a previous state, revert the entire project back to a previous state, compare changes over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more.

Who created git?

- Git was created by **Linus Torvalds** in 2005 to help with the development of the Linux kernel after using a proprietary system called BitKeeper.
- The goals for Git were speed, simple design, strong support for non-linear development, full distribution and the ability to handle large projects like the Linux kernel efficiently.



Git in action

Git takes snapshots of your project and lets you describe them.

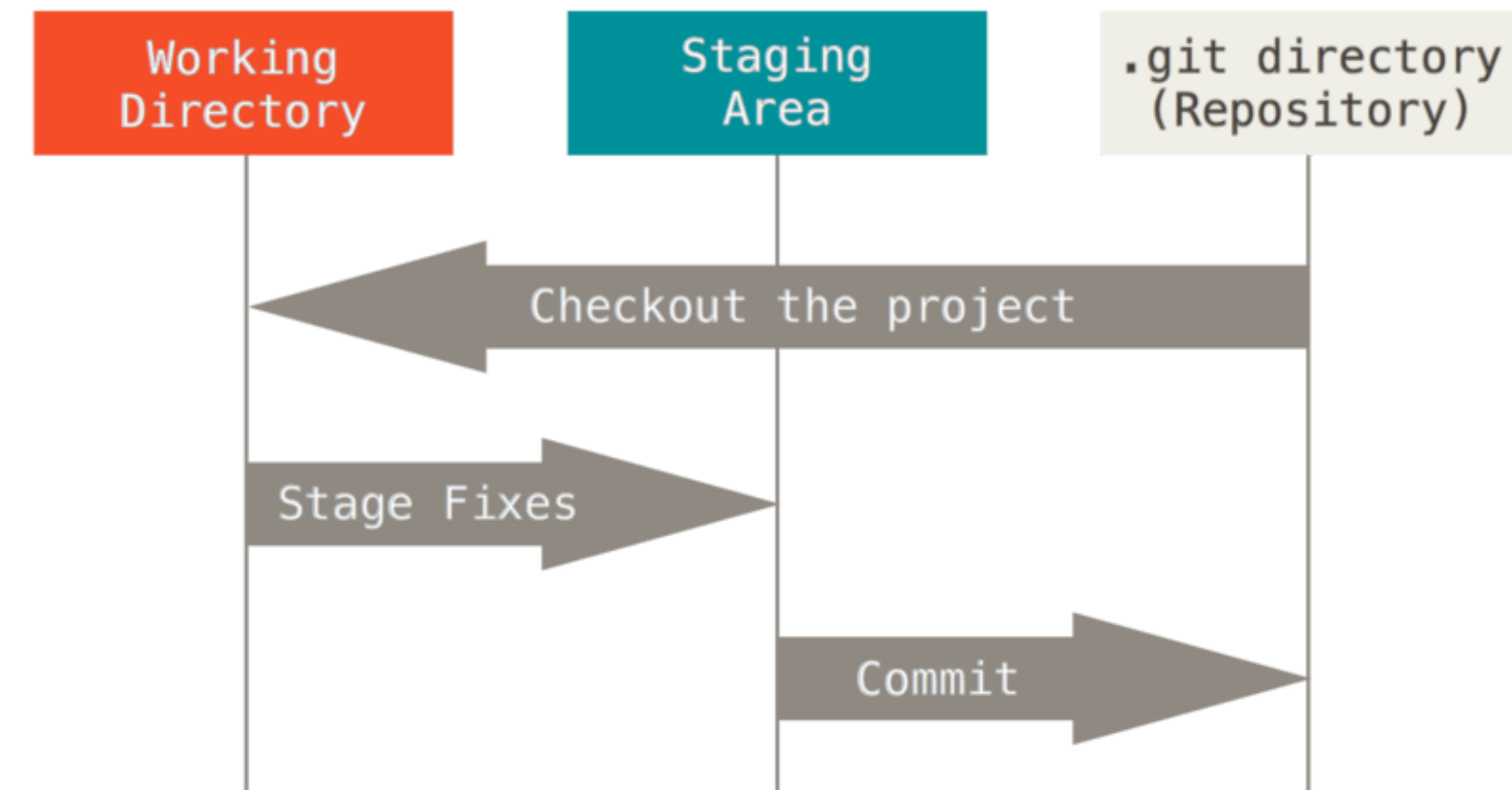
- View a history of your whole project and all the different changes
- Reverse changes as necessary
- Add meaningful descriptions to changes
- Collaborate easily, even with thousands of people.

Git in Action- Demo 🧐

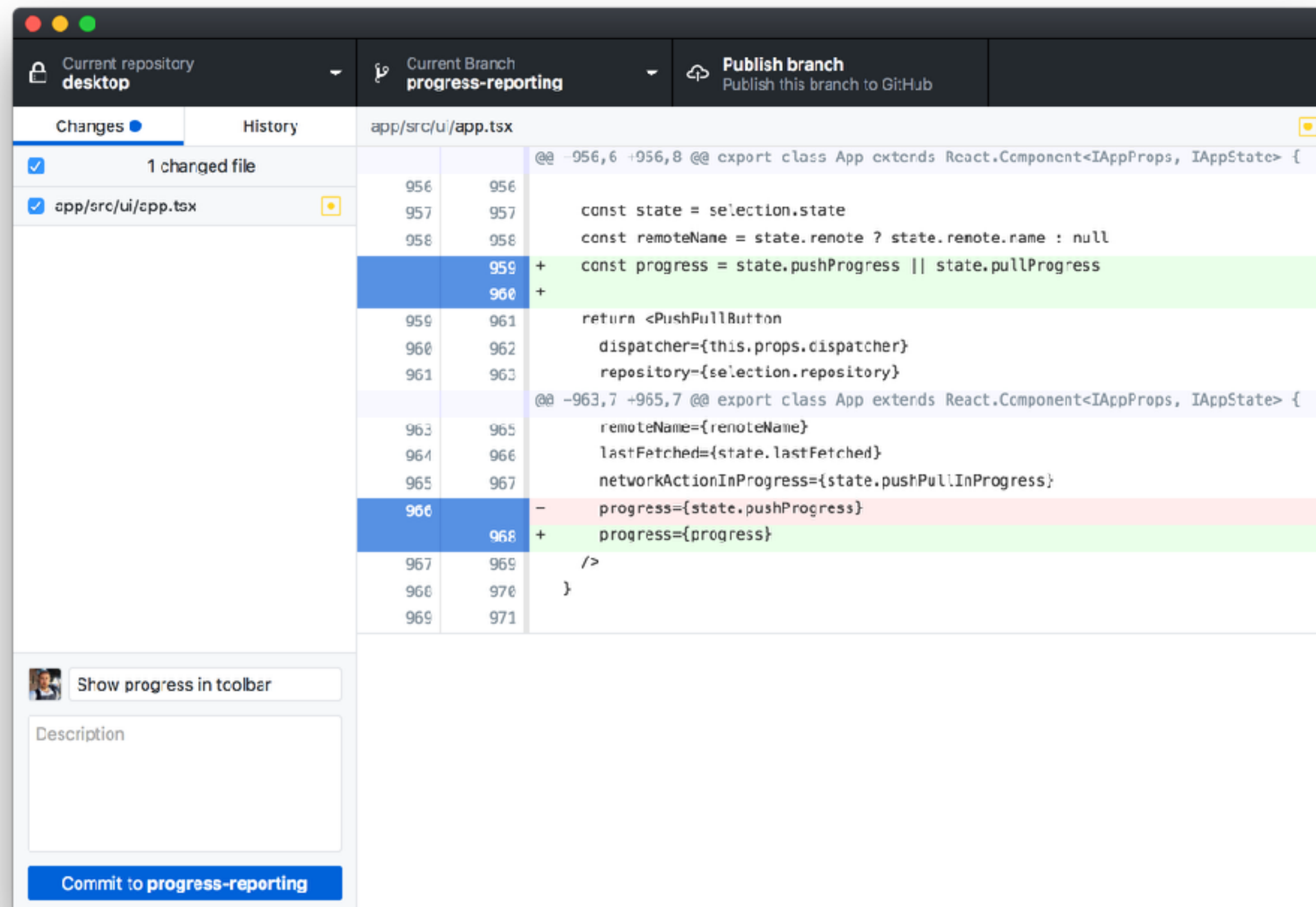


Remember the three states!

1. You modify the files in your working tree.
2. **You selectively stage just those changes you want to be part of your next commit**, which adds only those changes to the staging area
3. **You do a commit**, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.



Do I have to use the command line?



<https://desktop.github.com>

<https://www.git-tower.com/mac/>

GitHub



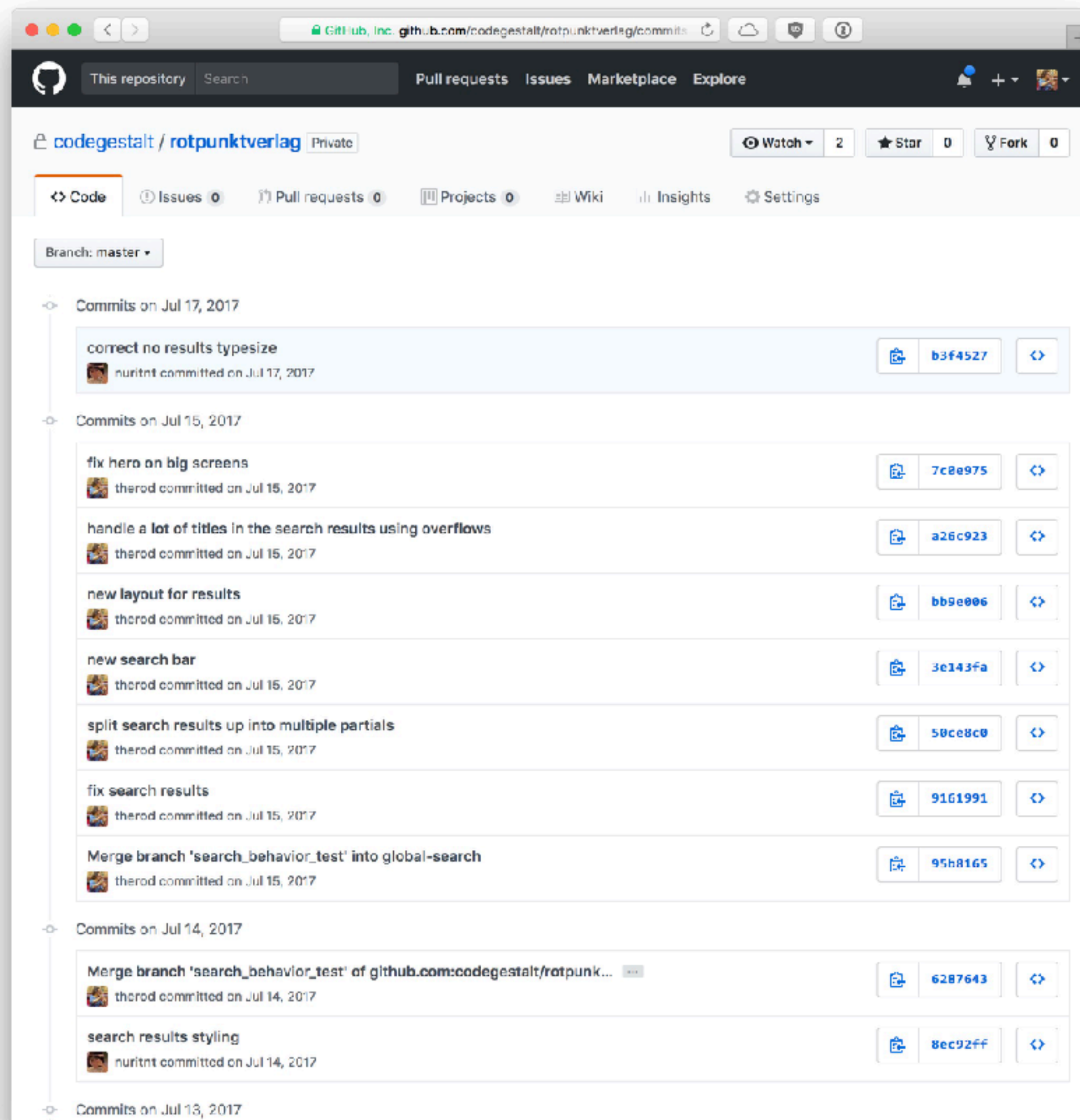
Difference between git & GitHub.

- **git** - a distributed version control system written in C. You install it on your local machine, there is no GUI by default. Everything is done via the command line and is used locally on your machine.
- **GitHub** - A web application built on Ruby on Rails that uses **git** in the background. You can have an account, create new projects and every project is a **git repository**. GitHub allows you to collaborate together on one project by offering tools like todos, issues, milestones and allows you to comment new changes in the source code.

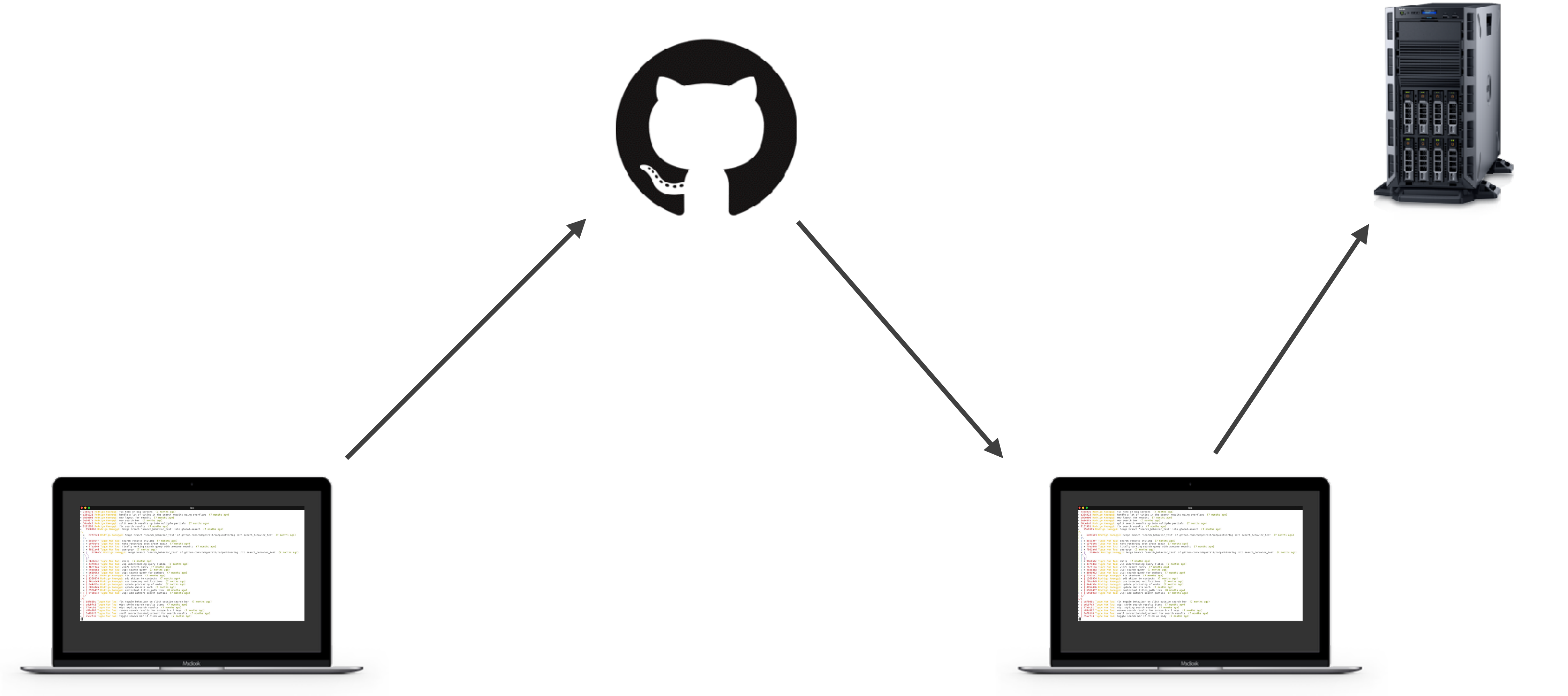
Git commit history in the Terminal

```
less
* 7c0e975 Rodrigo Haenggi: fix hero on big screens (7 months ago)
* a26c923 Rodrigo Haenggi: handle a lot of titles in the search results using overflows (7 months ago)
* bb9e006 Rodrigo Haenggi: new layout for results (7 months ago)
* 3e143fa Rodrigo Haenggi: new search bar (7 months ago)
* 50ce8c0 Rodrigo Haenggi: split search results up into multiple partials (7 months ago)
* 9161991 Rodrigo Haenggi: fix search results (7 months ago)
* 95b8165 Rodrigo Haenggi: Merge branch 'search_behavior_test' into global-search (7 months ago)
\
* 6287643 Rodrigo Haenggi: Merge branch 'search_behavior_test' of github.com:codegestalt/rotpunktverlag into search_behavior_test (7 months ago)
|
| \
| * 8ec92ff Tugce Nur Tas: search results styling (7 months ago)
| * c978efe Tugce Nur Tas: make rendering soon great again (7 months ago)
| * 7fea640 Tugce Nur Tas: finally working search query with awesome results (7 months ago)
| * f8d1a4d Tugce Nur Tas: queryyyy (7 months ago)
| * | 2f40e5c Rodrigo Haenggi: Merge branch 'search_behavior_test' of github.com:codegestalt/rotpunktverlag into search_behavior_test (7 months ago)
| \
| |
| | \
| | * 9b6b64e Tugce Nur Tas: chelp (7 months ago)
| | * 4379d3e Tugce Nur Tas: wip understanding query blabla (7 months ago)
| | * fhcf1aa Tugce Nur Tas: wip2: search query (7 months ago)
| | * 4eada5a Tugce Nur Tas: wip: search query (7 months ago)
| | * d600992 Tugce Nur Tas: wip: search query for authors (7 months ago)
| | * | 71b1cc1 Rodrigo Haenggi: fix checkout (7 months ago)
| | * | 1366974 Rodrigo Haenggi: add aktien to contacts (7 months ago)
| | * | f6bade9 Rodrigo Haenggi: use basecamp notifications (7 months ago)
| | * | 044d59e Rodrigo Haenggi: update processing of order (7 months ago)
| | * | d8544d6 Rodrigo Haenggi: update daniela koch (8 months ago)
| | * | 696b417 Rodrigo Haenggi: contextual titles_path link (8 months ago)
| | * | 5f6b91c Tugce Nur Tas: wip: add authors search partial (7 months ago)
| |
| | /
| /
* | b8780bc Tugce Nur Tas: fix toggle behaviour on click outside search bar (7 months ago)
* | adcb7c3 Tugce Nur Tas: wip: style search results items (7 months ago)
* | 77a6cb1 Tugce Nur Tas: wip: styling search results (7 months ago)
* | a84a962 Tugce Nur Tas: remove search results for escape & > 2 keys (7 months ago)
* | 3a791f6 Tugce Nur Tas: small corrections/adjustment for search results (7 months ago)
* | c552f33 Tugce Nur Tas: toggle search bar if click on body (7 months ago)
:
```


Git commit history on GitHub



Workflow



My laptop

Friends Laptop

Async collaboration



[rails / rails](#)

[Watch](#) 2,530 [Star](#) 38,480 [Fork](#) 15,588

[Code](#) [Issues](#) 350 [Pull requests](#) 709 [Projects](#) 0 [Insights](#)

`nil` test parameters should be `nil` in the controller #31845

[Open](#) tenderlove wants to merge 2 commits into `master` from `fix-nil-params-in-ap`

[Conversation](#) 7 [Commits](#) 2 [Files changed](#) 2 +12 -0

tenderlove commented 10 hours ago Owner

This commit fixes a regression where `nil` parameters in the test would get converted to empty strings in the controller. I added a test to `4-2-stable` to demonstrate the existing behavior in `3acf0de`.

This commit will fix the regression, but raises a broader issue about the behavior of `to_query`. It seems that our `to_query` implementation doesn't generate query strings that round trip via Rack's query parser:

```
irb(main):011:0> { :foo => nil }.to_query
=> "foo="
irb(main):012:0> Rack::Utils.default_query_parser.parse_nested_query({ :foo => nil }.to_query)
=> {"foo"=>""}
```

I'm not sure if we *should* fix this, but I guess it's something we need to know about.

/cc @mistydemeo

3

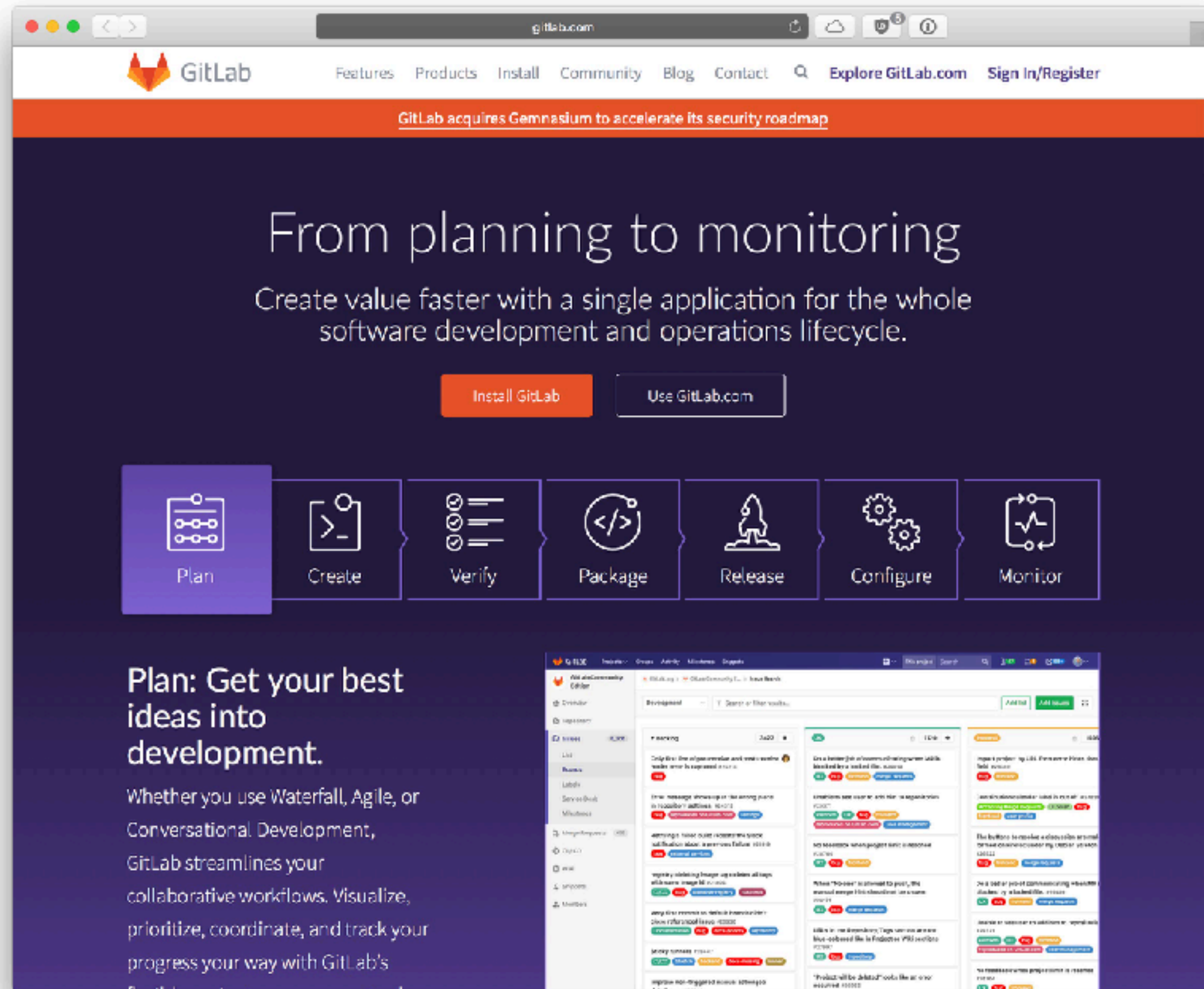
rafaelfranca commented 10 hours ago Owner

Do you think this would cause a different regression now for applications that already changed their tests to match the new behavior?

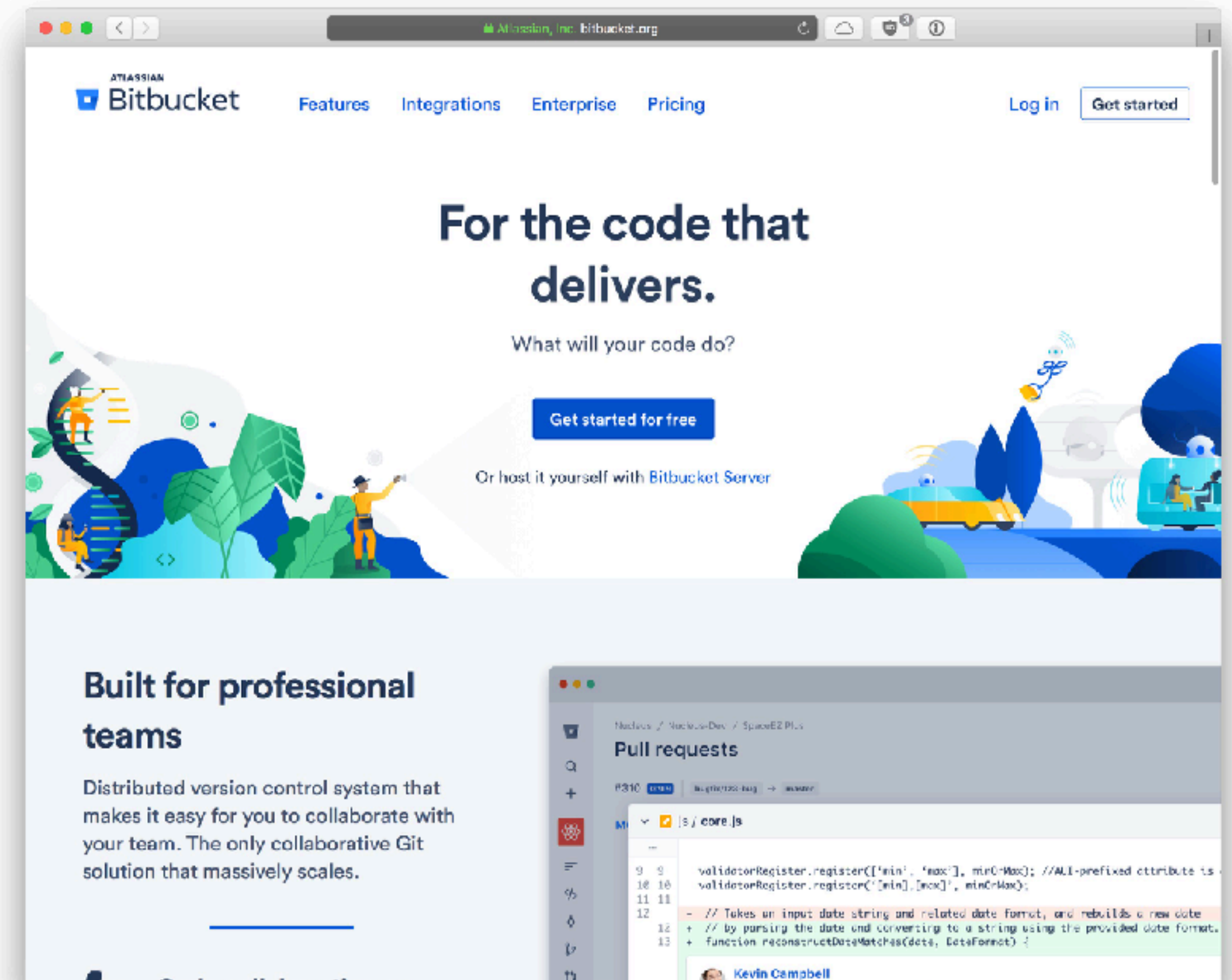
verified 1187e43

4 participants

Alternatives to GitHub



<https://gitlab.com>



<https://bitbucket.org>

GitHub in action - Demo 🧐

Git & GitHub - Your turn 🏃

Another cheatsheet



GIT CHEAT SHEET

presented by TOWER > Version control with Git - made easy

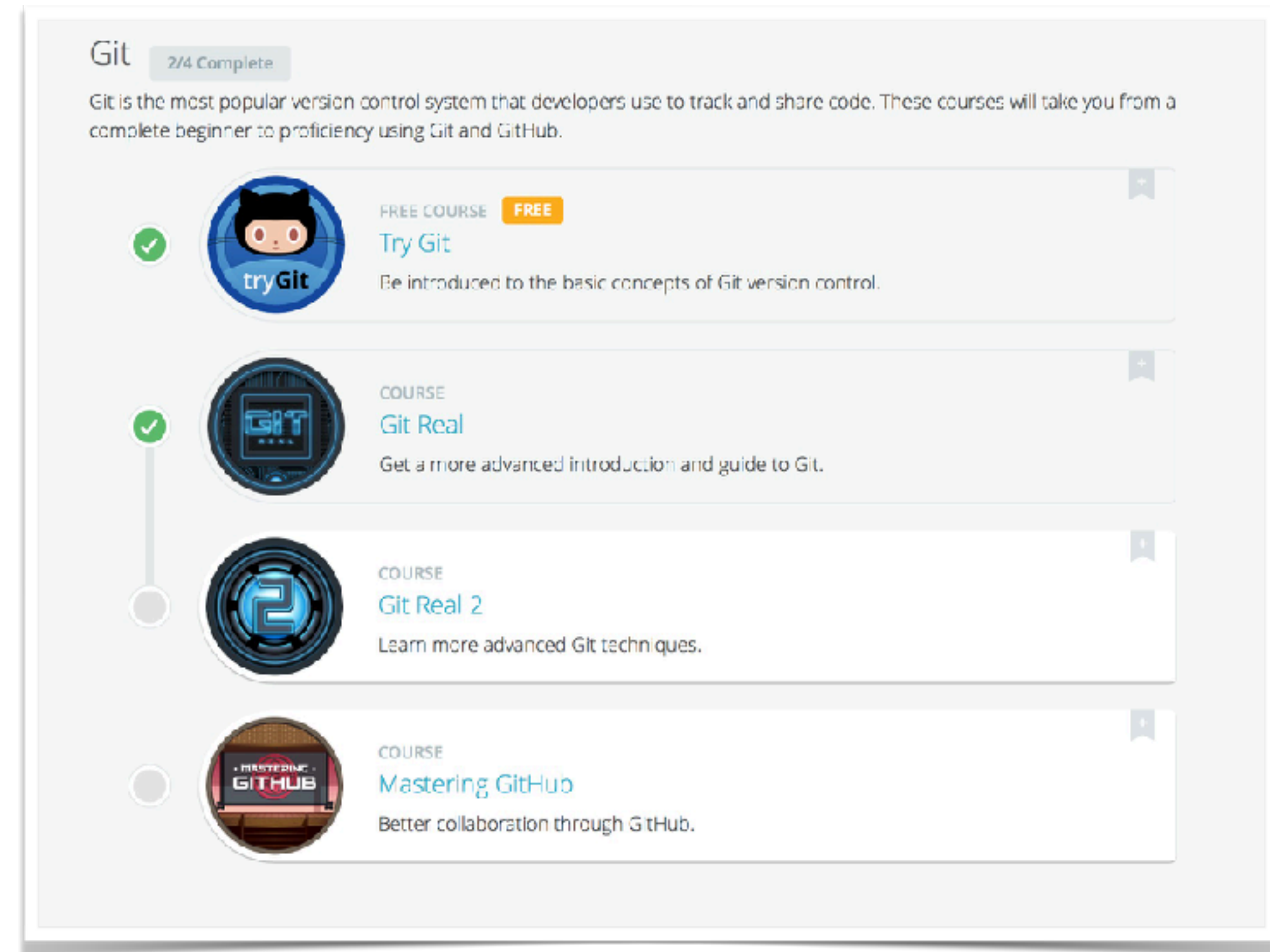
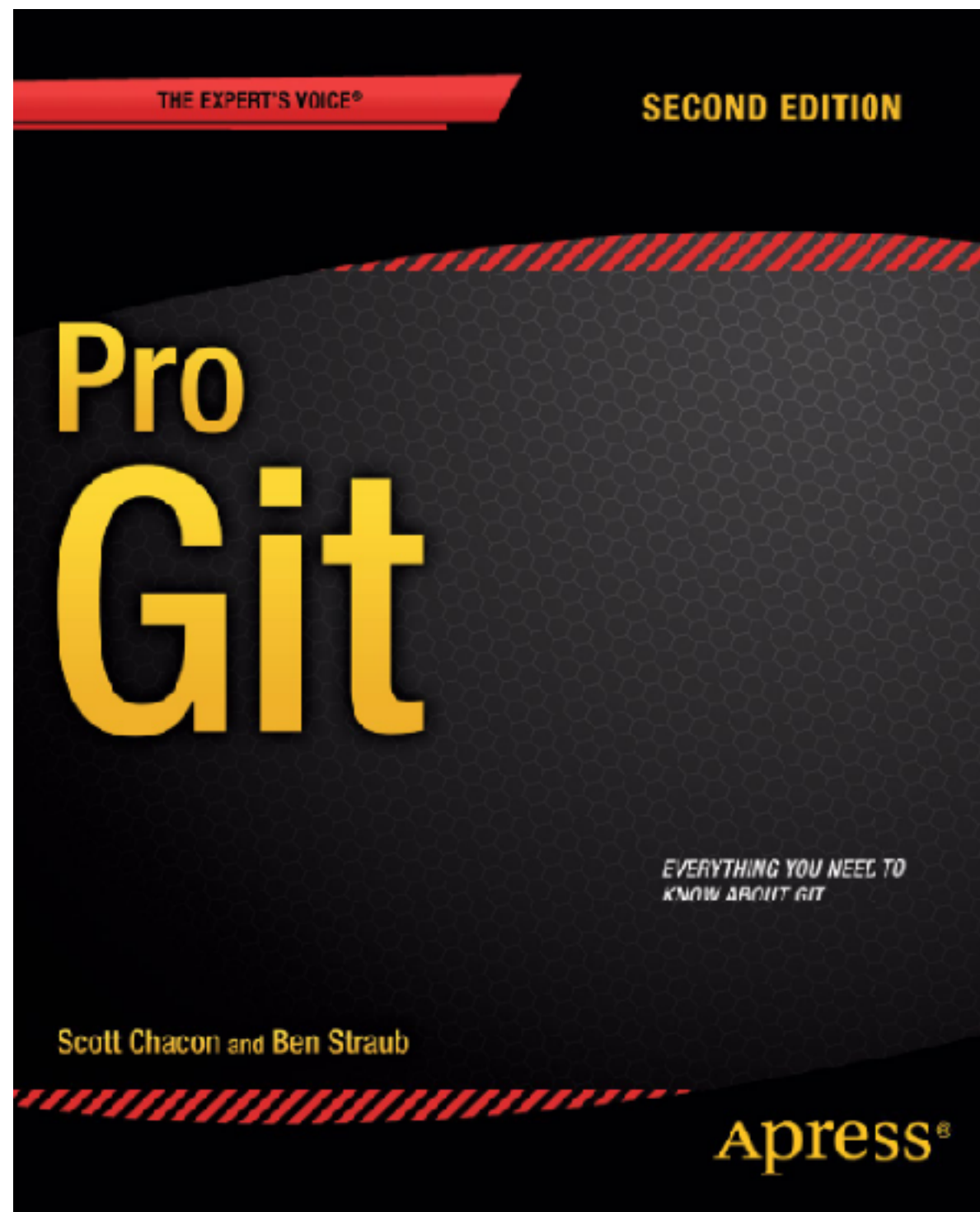


| | | |
|---|---|--|
| CREATE | BRANCHES & TAGS | MERGE & REBASE |
| Clone an existing repository <code>\$ git clone ssh://user@domain.com/repo.git</code> | List all existing branches <code>\$ git branch -av</code> | Merge <branch> into your current HEAD <code>\$ git merge <branch></code> |
| Create a new local repository <code>\$ git init</code> | Switch HEAD branch <code>\$ git checkout <branch></code> | Rebase your current HEAD onto <branch> <i>Don't rebase published commits!</i> <code>\$ git rebase <branch></code> |
| LOCAL CHANGES | Create a new branch based on your current HEAD <code>\$ git branch <new-branch></code> | Abort a rebase <code>\$ git rebase --abort</code> |
| Changed files in your working directory <code>\$ git status</code> | Create a new tracking branch based on a remote branch <code>\$ git checkout --track <remote/branch></code> | Continue a rebase after resolving conflicts <code>\$ git rebase --continue</code> |
| Changes to tracked files <code>\$ git diff</code> | Delete a local branch <code>\$ git branch -d <branch></code> | Use your configured merge tool to solve conflicts <code>\$ git mergetool</code> |
| Add all current changes to the next commit <code>\$ git add .</code> | Mark this current commit with a tag <code>\$ git tag <tag-name></code> | Use your editor to manually solve conflicts and (after resolving) mark file as resolved <code>\$ git add <resolved-file></code> <code>\$ git rm <resolved-file></code> |
| Add some changes in <file> to the next commit <code>\$ git add -p <file></code> | UPDATE & PUBLISH | UNDO |
| Commit all local changes in tracked files <code>\$ git commit -a</code> | List all currently configured remotes <code>\$ git remote -v</code> | Discard all local changes in your working directory <code>\$ git reset --hard HEAD</code> |
| Commit previously staged changes <code>\$ git commit</code> | Show information about a remote <code>\$ git remote show <remote></code> | Discard local changes in a specific file <code>\$ git checkout HEAD <file></code> |
| Change the last commit <i>Don't amend published commits!</i> <code>\$ git commit --amend</code> | Add new remote repository, named <remote> <code>\$ git remote add <shortname> <url></code> | Revert a commit (by producing a new commit with contrary changes) <code>\$ git revert <commit></code> |
| COMMIT HISTORY | Download all changes from <remote>, but don't integrate into HEAD <code>\$ git fetch <remote></code> | Reset your HEAD pointer to a previous commit and discard all changes since then <code>\$ git reset --hard <commit></code> |
| Show all commits, starting with newest <code>\$ git log</code> | Download changes and directly merge/integrate into HEAD <code>\$ git pull <remote> <branch></code> | and preserve all changes as unstaged changes <code>\$ git reset <commit></code> |
| Show changes over time for a specific file <code>\$ git log -p <file></code> | Publish local changes on a remote <code>\$ git push <remote> <branch></code> | and preserve uncommitted local changes <code>\$ git reset --keep <commit></code> |
| Who changed what and when in <file> <code>\$ git blame <file></code> | Delete a branch on the remote <code>\$ git branch -dr <remote/branch></code> | |
| | Publish your tags <code>\$ git push --tags</code> | |

40-day free trial available at
www.git-tower.com

TOWER
Version control with Git - made easy

Further reading



<https://git-scm.com/book/en/v2>

<https://www.codeschool.com/learn/git>