

Simplifying the contribution process for both contributors & maintainers

A case study of the  integration in 



FOSDEM 2018

Testing and Automation devroom

Feb 3rd 2018, Brussels (Belgium)

Kenneth Hoste

kenneth.hoste@ugent.be

GitHub: @boegel

Twitter: @kehoste



*GitHub is a code hosting service
for git repositories*

<https://github.com>

- launched in April 2008
- currently most popular platform for collaborating on FOSS
 - over 24M users, 67M repositories
 - over 100M *merged* pull requests
- also provides issue tracker, code review facilities, wiki, ...
- used by Docker, Python, TensorFlow, (Linux kernel), ...



*Git is a free and open source (GPLv2)
distributed version control system*

<https://git-scm.com>

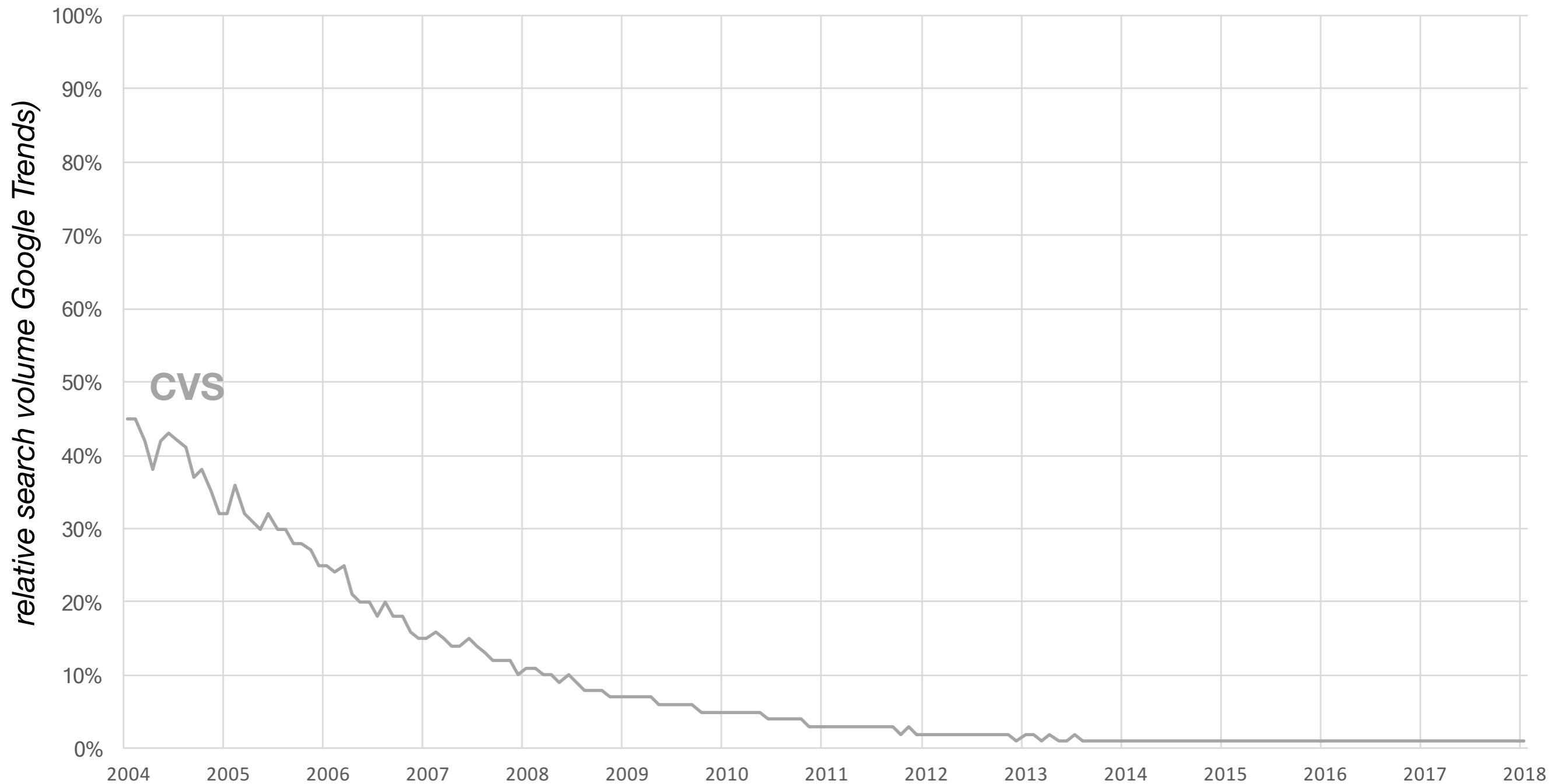
- created in 2005 by Linus Torvalds
- currently (by far) most popular version control system (VCS)
- design goals:
 - good performance, even for large projects (Linux kernel)
 - support for non-linear development & distributed workflow
 - to not be like CVS (Concurrent Versions System)
- internals similar to modern *blockchain* technologies (Merkle trees)
- used by Linux kernel, Windows, Android, Python, TensorFlow, ...

Git workflow: branch, stage, commit, push

```
# edit file to contribute  
# (you can use emacs too, if you insist)  
$ vim file  
  
# create branch, and check it out  
$ git checkout -b branch_name  
  
# stage file  
$ git add file  
  
# commit changes with sensible commit message  
$ git commit -m "this is a sensible commit message"  
  
# push branch to remote repository  
$ git push remote_repo branch_name
```

A little bit of VCS history

(inspired by: <https://blog.gitprime.com/git-didnt-beat-svn-github-did>, May 2016)



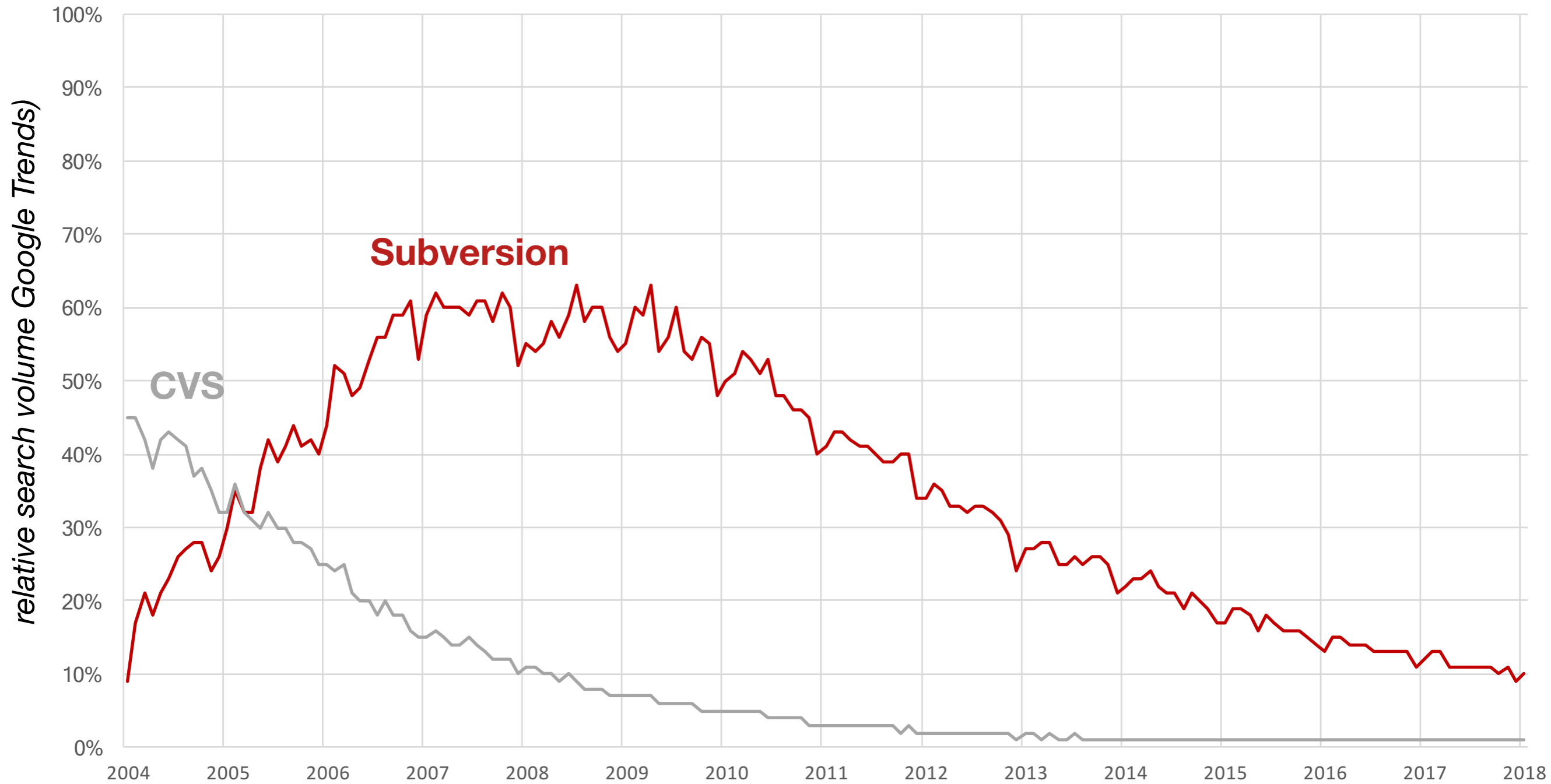
Concurrent Versions System (CVS)

<http://www.nongnu.org/cvs>

initial release: Nov 1990, last release: May 2008

A little bit of VCS history

(inspired by: <https://blog.gitprime.com/git-didnt-beat-svn-github-did>, May 2016)



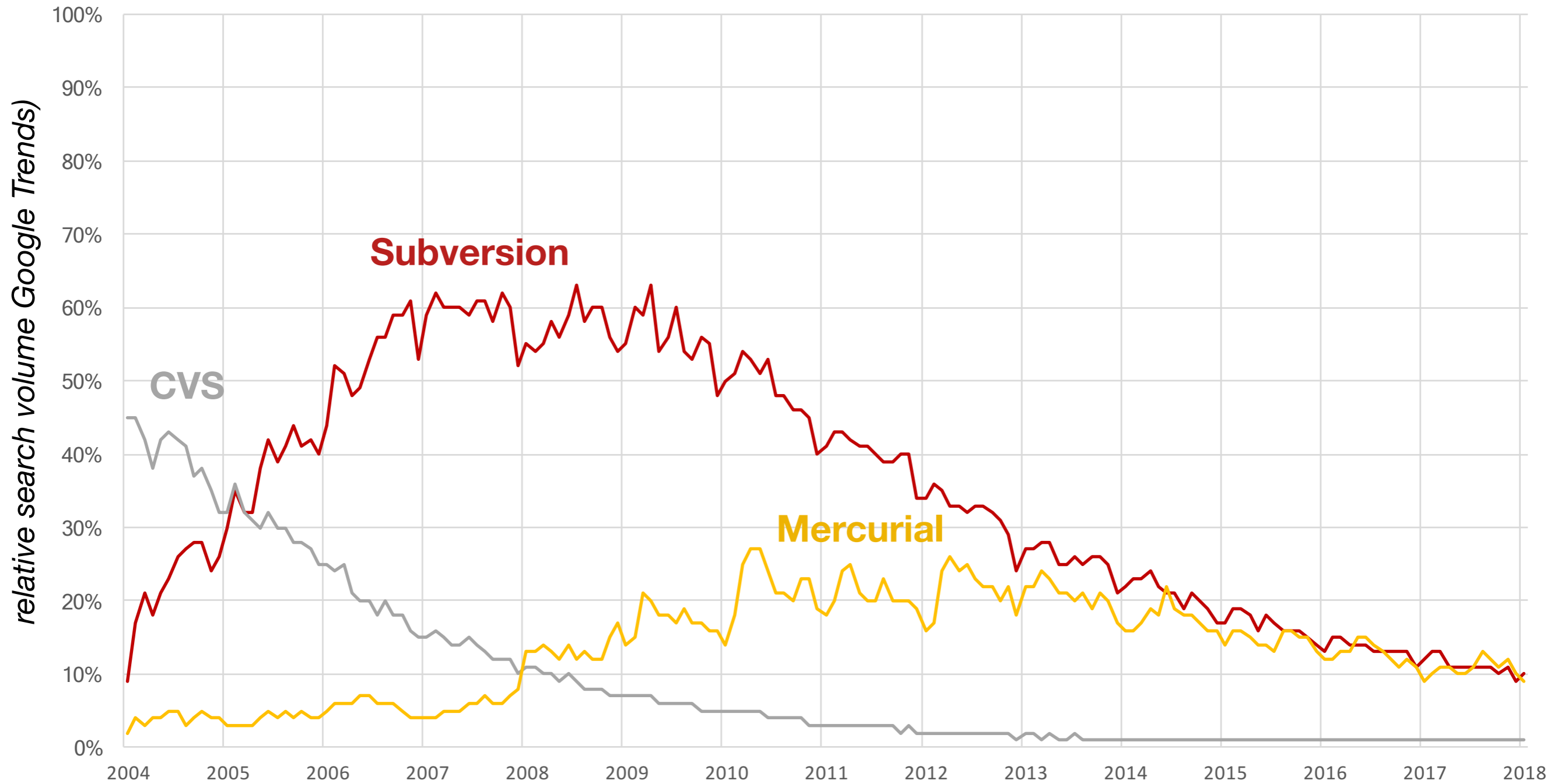
Apache Subversion (SVN)

<https://subversion.apache.org>

initial release: Oct 2000, last release: Aug 2017

A little bit of VCS history

(inspired by: <https://blog.gitprime.com/git-didnt-beat-svn-github-did>, May 2016)



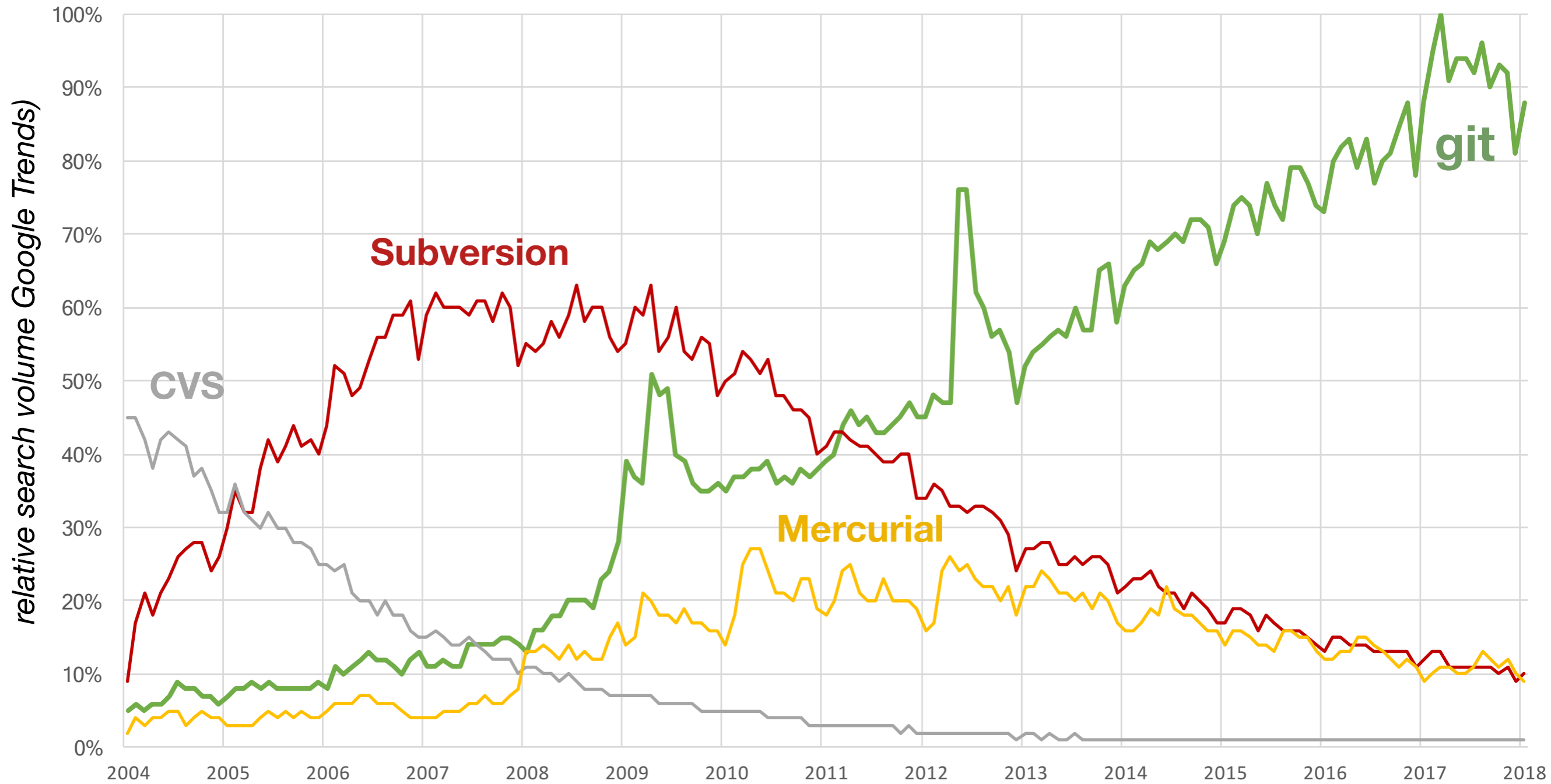
Mercurial

<https://www.mercurial-scm.org>

initial release: April 2005, last release: Dec 2017

A little bit of VCS history

(inspired by: <https://blog.gitprime.com/git-didnt-beat-svn-github-did>, May 2016)



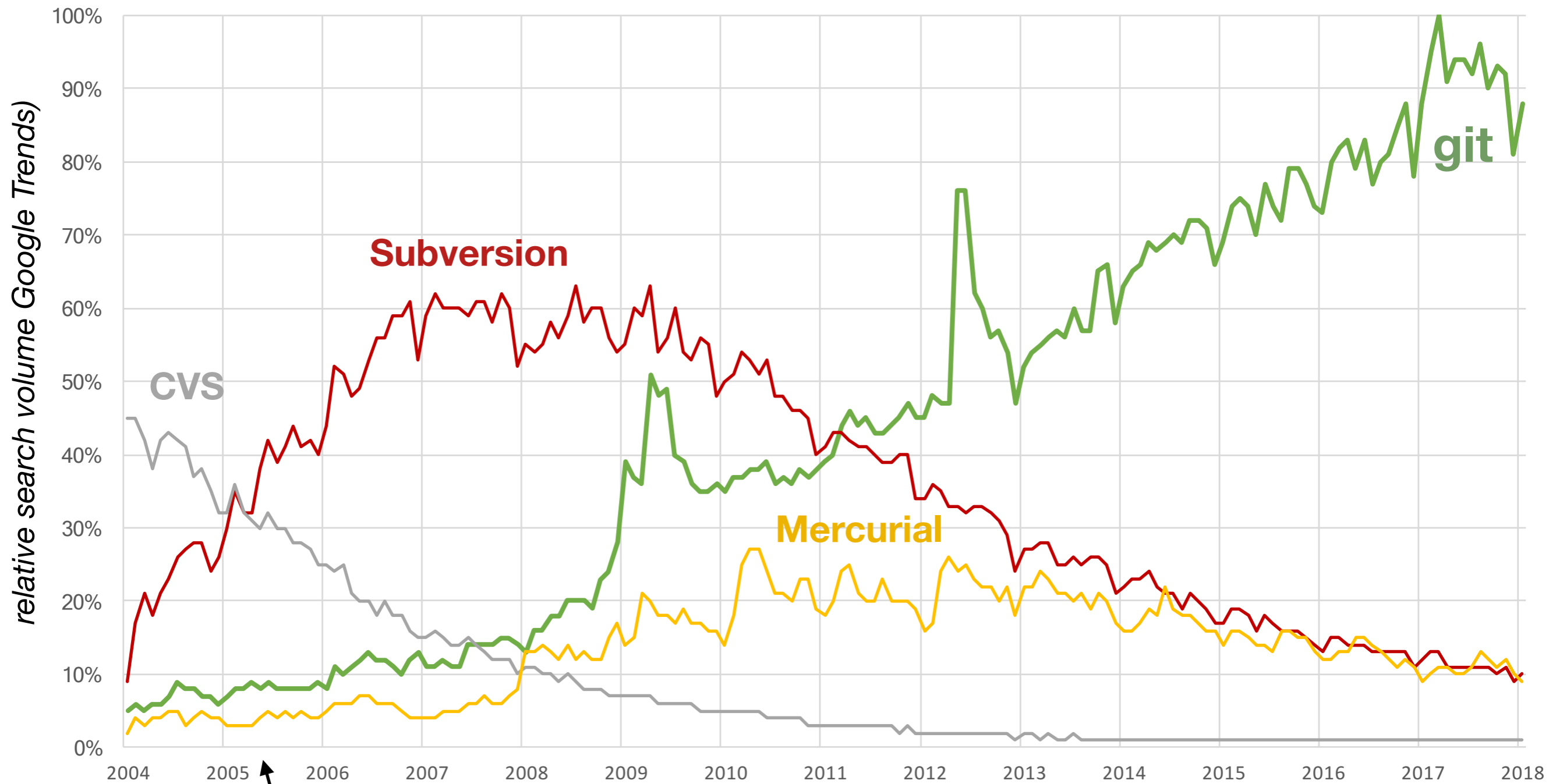
git

<https://git-scm.com>

initial release: April 2005, last release: Jan 2018

Why did git become so popular?

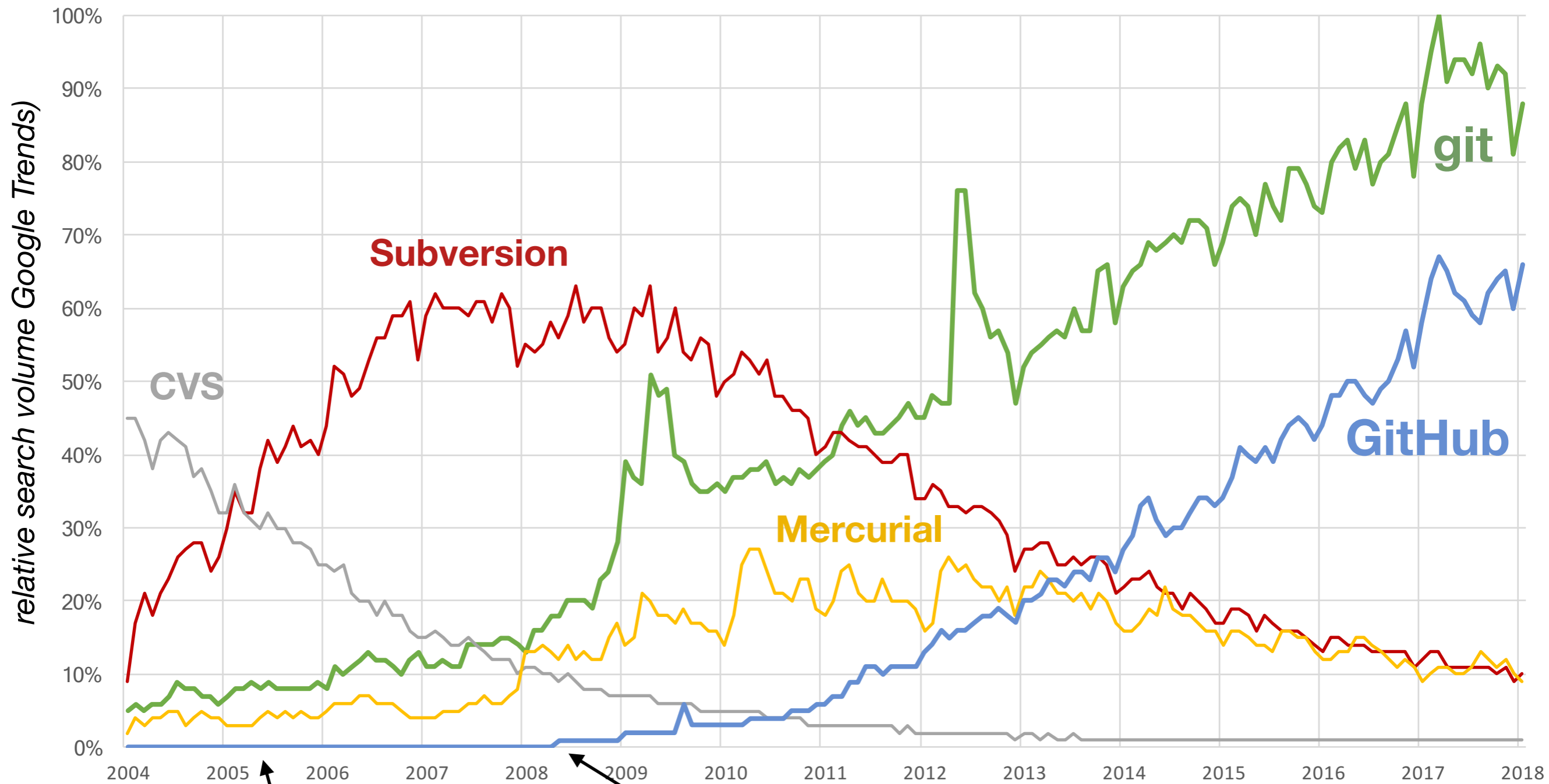
(inspired by: <https://blog.gitprime.com/git-didnt-beat-svn-github-did>, May 2016)




 git initial release
(April 2005)

Is git so popular thanks to GitHub?

(inspired by: <https://blog.gitprime.com/git-didnt-beat-svn-github-did>, May 2016)

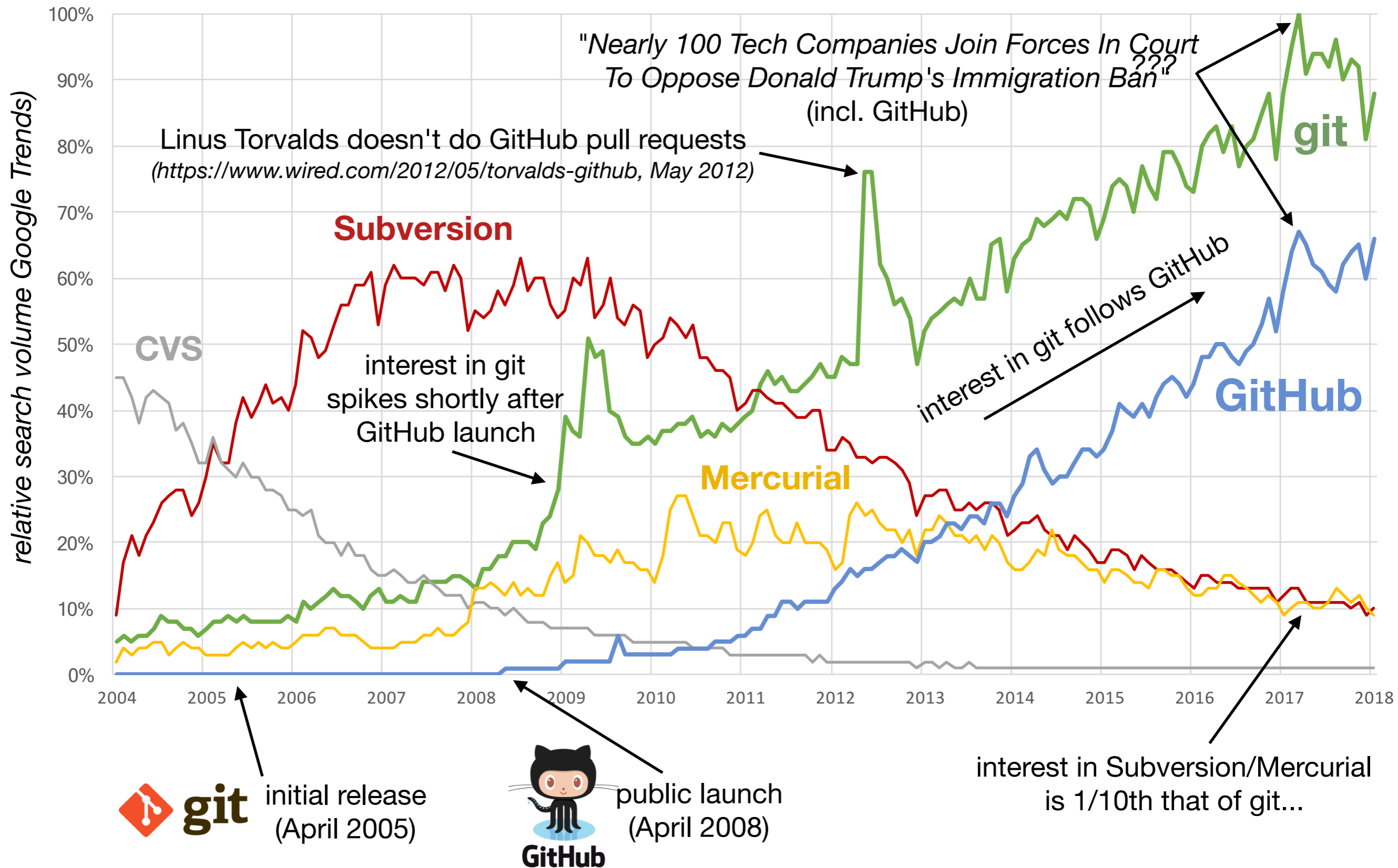


 **git** initial release
(April 2005)

 public launch
(April 2008)
GitHub




Is git so popular thanks to GitHub?

(inspired by: <https://blog.gitprime.com/git-didnt-beat-svn-github-did>, May 2016)



Is **git** so popular thanks to **GitHub**?



- Hard to be sure, but it's likely  has had a major impact.
GitHub
- Maybe there even was some sort of feedback loop...
(**GitHub** increases interest in , which in turns makes **GitHub** more popular?)
- ... or maybe it's just because **GitHub** has 'git' in its name, and I can't get Google Trends to behave...
- Whatever the reason,  **git** is the most popular (D)VCS today.
- You may need it in order to contribute to a FOSS project.
- *Which means you will need to learn how to use it...*



git has a steep learning curve

<https://stackoverflow.com/questions/tagged/git>

16362

votes

75
answers

5.9m
views

How to undo the most recent commits in Git?

I accidentally committed wrong files to Git, but I haven't pushed the commit to the server yet. How can I undo those commits from Git?

git git-commit git-reset git-revert

community wiki

61 revs, 40 users 14%

Peter Mortensen

12440

votes

40
answers

5.1m
views

How do I delete a Git branch both locally and remotely?

I want to delete a branch both locally and on my remote project fork on GitHub. Failed Attempts to Delete Remote Branch \$ git branch -d remotes/origin/bugfix error: branch 'remotes/origin/bugfix' ...

git git-branch git-remote

asked Jan 5 '10 at 1:12



Matthew Rankin

227k ● 33 ● 100 ● 139

9236

votes

44
answers

2.1m
views

What is the difference between 'git pull' and 'git fetch'?

What are the differences between git pull and git fetch?

git git-pull git-fetch

asked Nov 15 '08 at 9:51



Pablo

94.9k ● 96 ● 258 ● 387

(in comparison: the infamous "How to exit Vim" has 'only' 1.3M views...)

git has a steep learning curve

<https://twitter.com/msccccc/status/954098595623391233>



Mike Coutermarsh

@msccccc

Just googled "undo last commit but keep changes"

And I work for GitHub.

This stuff is hard!

6:10 PM - 18 Jan 2018

244 Retweets 855 Likes



 34

 244

 855



git has a steep learning curve

(from <https://www.quora.com/Why-is-Git-so-hard-to-learn>)

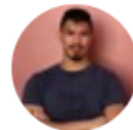


Terry Lambert, former Sr. Software Engineer at Google (2011-2012)

Answered Nov 10, 2016

The primary problem with git is that there are too many ways to do the same few, conceptually simple things.

***It's the perl of source code control systems.
This is not a compliment.***



William Estoque, Ruby, Ruby on Rails, Javascript, Engineer, Startups

Answered Nov 4, 2016

Git is hard to learn because people try to learn everything at once.



Ori Shalev, a software engineer

Answered Jan 2, 2017

Because probably like most people, you skipped the basics.



Jesse McGrew, 15+ years as a professional and open source developer

Answered Jan 8, 2017

Git is hard to learn because its developers gave up on making it easy.

git is not for everybody


- main target audience is *software developers*
- in particular *teams* of people working together on software
- a lot of people don't need it for their *daily work*
- some feel they lack the *time* to learn it
- others *try* to learn, but can't seem to get their head around it...
- not everybody knows an experienced git user to *help* them
- some people don't **want** to learn it (and that's fine)

git can be a major hurdle for contributors

- Potential contributors may not be familiar (yet) with `git`...
- Do they *have* to learn `git` first before they can contribute?



The burden for contributors

- be sufficiently familiar with the project they want to contribute to
 - programming language(s) used
 - overall structure of the project's codebase
 - project-specific policies & expectations,
incl. code style rules, legal aspects, (git) workflow, testing, ...
 - recommended approach to implement a bug fix/enhancement
- know enough about  **git** & **GitHub** to open a '*correct*' pull request
- initial effort for newcomers is often (heavily) underestimated
- documentation is very useful, but not sufficient (for everyone)

A little bit about easybuild ...

<http://easybuilders.github.io/easybuild>

- **framework for installing (scientific) software on **HPC** systems**
- uses easyconfig files as 'recipes'
 - key-value definitions for easyconfig parameters (Python syntax)
 - specify software name, version, compiler toolchain, dependencies, ...
- common contributions to EasyBuild:
 - add easyconfig file for a new software version, or for other toolchain
 - very often based on an existing easyconfig file (copy & modify)
 - statement that "this particular (set of) recipe(s) works for me"

"high-performance computing"

Would you like to know more?

Installing software for scientists on a multi-user HPC system

A comparison between conda, EasyBuild, Guix, Nix & Spack

 **CONDA**

 **easybuild**

 **Guix**

 **Nix**

 **Spack**

Sunday February 4th 2018 - 9am, room H.1302 (Depage)

FOSDEM'18 HPC, Big Data & Data Science devroom

https://fosdem.org/2018/schedule/event/installing_software_for_scientists

Example usage of EasyBuild

```
$ eb sklearn.eb
== temporary log file in case of crash /tmp/eb-Wqpzyc/easybuild-5ll606.log
== processing EasyBuild easyconfig /home/boegel/sklearn.eb
== building and installing scikit-learn/0.19.1-intel-2017b-Python-3.6.3...
== fetching files...
== creating build dir, resetting environment...
== unpacking...
== patching...
== preparing...
== configuring...
== building...
== testing...
== installing...
== taking care of extensions...
== postprocessing...
== sanity checking...
== cleaning up...
== creating module...
== permissions...
== packaging...
== COMPLETED: Installation ended successfully
== Results of the build can be found in the log file(s) /example/software/scikit-learn/0.19.1-
intel-2017b-Python-3.6.3/easybuild/easybuild-scikit-learn-0.19.1-20180125.210511.log
== Build succeeded for 1 out of 1
== Temporary log file(s) /tmp/eb-Wqpzyc/easybuild-5ll606.log* have been removed.
== Temporary directory /tmp/eb-Wqpzyc has been removed.
```

*EasyBuild consumes **easyconfig files**,
which specify what to install*

Example easyconfig file

```
easyblock = 'PythonPackage'

name = 'scikit-learn'
version = '0.19.1'
versionsuffix = '-Python-%(pyver)s'

homepage = 'http://scikit-learn.org/stable/index.html'
description = "Scikit-learn is a Python library for machine learning algorithms."

toolchain = {'name': 'intel', 'version': '2017b'}

source_urls = [PYPI_SOURCE]
sources = [SOURCE_TAR_GZ]
checksums = ['5ca0ad32ee04abe0d4ba02c8d89d501b4e5e0304bdf4d45c2e9875a735b323a0']

dependencies = [
    ('Python', '3.6.3'),
    ('matplotlib', '2.1.0', versionsuffix),
]

options = {'modulename': 'sklearn'}

sanity_check_paths = {
    'files': [],
    'dirs': ['lib/python%(pyshortver)s/site-packages/sklearn'],
}

moduleclass = 'data'
```

easy to tweak: **software version**,
compiler toolchain (version),
dependencies, configure options, ...

So you want to contribute to easybuild ...

To contribute an easyconfig file,
you first need to prepare:

I shall say do zis only once!



1) create a  account via **GitHub**

[Sign up for GitHub](#)

2)  [Fork https://github.com/easybuilders/easybuild-easyconfigs](https://github.com/easybuilders/easybuild-easyconfigs)

3) `git clone git@github.com:YOU/easybuild-easyconfigs.git`

(you *may* want to change this bit)

Branch, stage, commit, push

Once you have cloned the easyconfigs repository, you can really get to work...

```
# create and check out a new branch
$ git checkout -b example

# stage the new/changed easyconfig file(s)
$ git add easybuild/easyconfigs/s/scikit-learn/*.eb

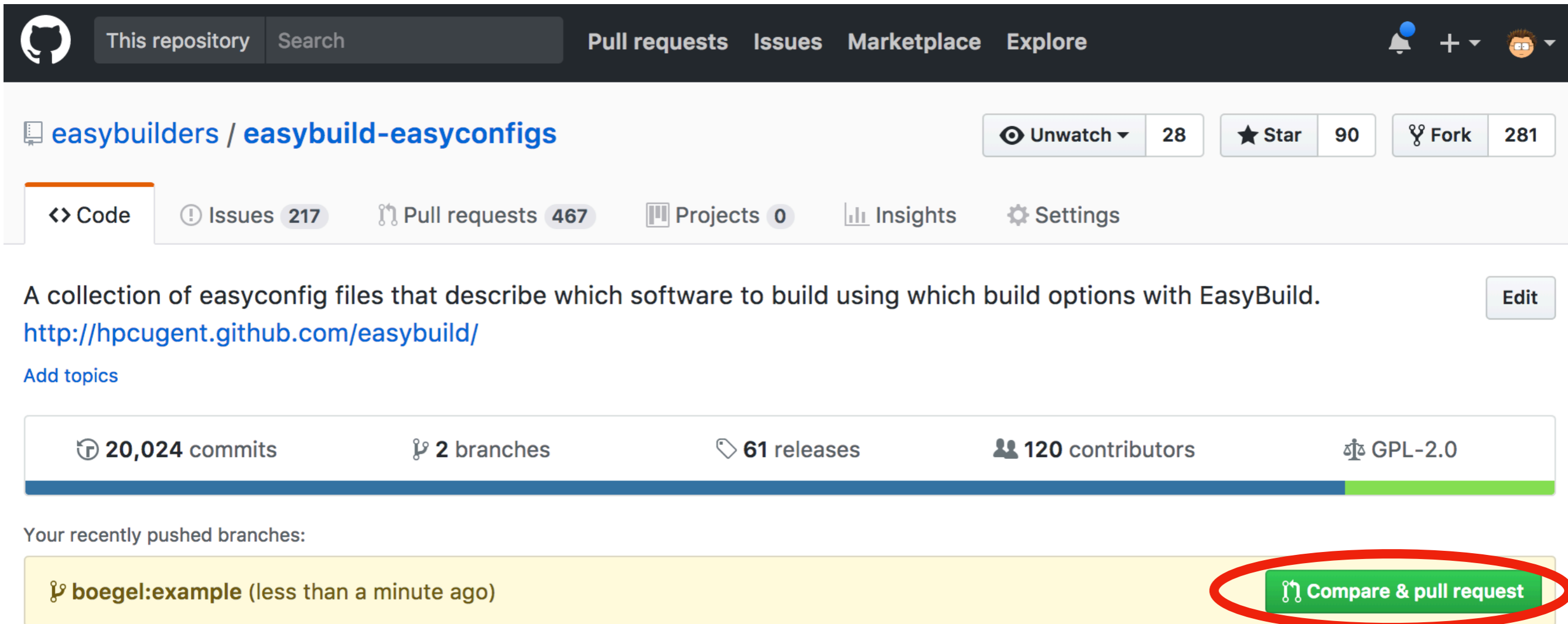
# commit the changes with a sensible commit message
$ git commit -m "update scikit-learn to 0.19.1"

# push your branch to your fork on GitHub
$ git push origin example
```

(this may be different in your setup, see `git remote -v`)

Clickety, clickety, click.

Next, log on to  and open a pull request...
GitHub



The screenshot shows the GitHub interface for the repository 'easybuilders / easybuild-easyconfigs'. At the top, there's a navigation bar with 'This repository', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below this, the repository name is displayed along with statistics: 'Unwatch' (28), 'Star' (90), and 'Fork' (281). A secondary navigation bar includes 'Code', 'Issues' (217), 'Pull requests' (467), 'Projects' (0), 'Insights', and 'Settings'. The repository description is 'A collection of easyconfig files that describe which software to build using which build options with EasyBuild.' with an 'Edit' button. Below the description, repository statistics are shown: '20,024 commits', '2 branches', '61 releases', '120 contributors', and 'GPL-2.0' license. At the bottom, a section titled 'Your recently pushed branches:' shows a branch named 'boegel:example' (less than a minute ago). A green button labeled 'Compare & pull request' is circled in red, with an arrow pointing to it from the text below.

You can use this useful button , if you're quick enough!



Almost there...

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base fork: easybuilders/easybuild-eas... **base: develop** head fork: boegel/easybuild-easyconfigs compare: example

✓ **Able to merge.** These branches can be automatically merged.

{data}[intel/2017b] scikit-learn 0.19.1 (Python 3.6.3)

Write Preview AA B i “ <> @

This is my first contribution to EasyBuild, please be gentle...

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Allow edits from maintainers. [Learn more](#)

Create pull request


Not too quick...

- target **develop**
- tweak PR title
- brief description
- double-check changed files

If you're sure, click the shiny green button...

1 commit **1 file changed** 0 commit comments 1 contributor

Are we done yet?

- once your pull request is open, your changes will be:
 - (automagically) quickly tested by  Travis CI
 - reviewed by a maintainer
 - thoroughly tested by at least one person
- additional changes may be requested...
- which means you may have to go back to `git`...
- maybe even multiple times... (spread over multiple hours/days)



Finetuning your contribution

Do you still remember the funny branch name you used for that PR?

```
# check out your branch again  
$ git checkout example  
  
# edit some files, as requested by the friendly reviewer  
$ vim ...  
  
# commit the ALL changes  
$ git commit -am "fixed suggestions by reviewer"  
  
# push your updated branch to GitHub to update the PR  
$ git push origin example
```

Too much overhead, manual steps...

- Isn't there an easier/more efficient way for doing this?
 - come up with a sensible funny branch name
 - 4 git commands to prepare the contribution
 - at least 4 well aimed mouse clicks to open the pull request
 - if requested, N x 3 more git commands for finetuning the PR
 - + a bit of cleanup afterwards
(delete branch both in local repository clone and on GitHub)
- All of this assuming you:
 - take into account the project policies
 - don't mess up with `git` at any time...

Back to easybuild

- in 2015, we saw a steady increase in incoming contributions
 - ~500 easyconfig pull requests in 2013-2014
 - almost double that in 2015...
- several people were struggling with the contribution procedure
 - mainly because they were not sufficiently familiar with `git`
 - most candidate contributors were not software developers
 - resulted in lots of frustration & wasted time
 - some people even gave up on contributing :(
- quite annoying for a project that aims to automate tedious software installation procedures...

FIXME
add plot

GitHub integration in easybuild

http://easybuild.readthedocs.io/en/latest/Integration_with_GitHub.html

- goals:
 - automate contribution workflow
 - avoid direct interaction with `git` as much as possible
- initial implementation added support for:
 - opening new PRs & updating existing PRs **eb --new-pr**
 - downloading & using easyconfigs from a PR **eb --update-pr**
 - uploading test reports to a PR **eb --from-pr**
 - uploading test reports to a PR **eb --upload-test-report**
- limited to easyconfig files, for now...
 - but implemented such that it is easy to extend to easyblocks, and even the EasyBuild framework itself...

Weapons of choice

- **GitPython** (<https://pypi.python.org/pypi/GitPython>)
 - Python library to interact with  **git**
- **keyring** Python package (<https://pypi.python.org/pypi/keyring>)
 - Python library that provides access to system keyring
 - required to securely store GitHub tokens
-  **API** (<https://developer.github.com/v3>)
 - to automate interaction with GitHub (less clickety, clickety)
- outlined approach is specific to Python, but it should be easy to implement something similar in other programming languages

Opening new pull requests

eb --new-pr

```
$ eb --new-pr sklearn.eb
```

```
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
    https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
    /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...
```

Opening pull request

```
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
```

```
"""
```

```
(created using `eb --new-pr`)
```

```
"""
```

```
* overview of changes:
```

```
.../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++
1 file changed, 31 insertions(+)
```

```
Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
```

```
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
```

```
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

one single 'eb' command
no 'git' commands, no clickety on GitHub

Opening new pull requests

eb --new-pr

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
/tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...
```

Opening pull request

```
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
```

```
"""
```

```
(created using `eb --new-pr`)
```

```
"""
```

```
* overview of changes:
```

```
.../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++
1 file changed, 31 insertions(+)
```

```
Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
```

```
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
```

```
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

repository is **cloned** to temporary location
develop branch is **checked out** as a base

Opening new pull requests

eb --new-pr

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
    https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
    /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...
```

*auto-derived
branch name*

Opening pull request

```
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs 20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
"""
```

GitHub repository to push to

```
(created using `eb --new-pr`)
```

*a new branch is created & pushed
(you can specify the branch name yourself)*

```
* overview of changes:
.../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++
1 file changed, 31 insertions(+)
```

```
Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

Opening new pull requests

eb --new-pr

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
    https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
    /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...
```

Opening pull request

```
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
```

```
""
(created using `eb --new-pr`)
```

the easyconfig file is *renamed*
and *moved to the right location*

```
""
* overview of changes:
```

```
../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++++
1 file changed, 31 insertions(+)
```

```
Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

Opening new pull requests

eb --new-pr

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
    https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
    /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...
```

Opening pull request

```
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
```

```
* description:
"""
(created using `eb --new-pr`)
"""
```

the pull request is prepared:
target develop branch, **title & description**

```
* overview of changes:
.../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++
1 file changed, 31 insertions(+)
```

```
Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

Opening new pull requests

eb --new-pr

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
    https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
    /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...
```

Opening pull request

```
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
"""
```

```
(created using `eb --new-pr`)
```

```
"""
```

```
* overview of changes:
.../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++++
1 file changed, 31 insertions(+)
```

a high-level *overview of the changes* is shown (similar to output of 'git stat')

```
Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

Opening new pull requests

eb --new-pr

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
    https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
    /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...
```

Opening pull request

```
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
```

```
"""
```

```
(created using `eb --new-pr`)
```

```
"""
```

```
* overview of changes:
```

```
.../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++
1 file changed, 31 insertions(+)
```

the *pull request is opened automatically*,
by leveraging the GitHub API

```
Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
```

```
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
```

```
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

Long story short

eb --new-pr

```
$ mv sklearn.eb scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb
$ mv scikit-*eb easybuild/easyconfigs/s/scikit-learn
$ git checkout -b scikit_learn_0191_intel_2017b
$ git add easybuild/easyconfigs/s/scikit-learn
$ git commit -m "{data}[intel/2017b] scikit-learn v0.19.1"
$ git push origin scikit_learn_0191_intel_2017b
```

... and then log into GitHub, and open the pull request (clickety, clickety...)



eb --new-pr sklearn.eb

Updating existing pull requests

```
eb --update-pr
```

- foo

Using easyconfigs from a PR

```
eb --from-pr
```

- foo

Uploading test reports to a PR

```
eb --upload-test-report
```

- foo

More GitHub integration, MORE !!!1!

GitHub integration was extended gradually over time:

- checking configuration w.r.t. GitHub integration `eb --check-github`
- verifying & installing a GitHub token `eb --install-github-token`
- running `--new-pr` and `--update` in dry run mode `eb --dry-run`
- downloading and using *partial* PRs `eb --from-pr 123 example.eb`
- (p)reviewing PRs `eb --preview-pr` `eb --review-pr`
- merging PRs, after checking they're eligible `eb --merge-pr`
- improve performance by avoiding to clone from scratch

Checking configuration for **GitHub** integration

```
eb --check-github
```

- foo

Verifying and installing a **GitHub** token

```
eb --install-github-token
```

- foo

Dry run mode

```
eb --new-pr --dry-run
```

- foo

Using *specific* easyconfigs from a PR

```
eb --from-pr 123 example.eb
```

- foo

Reviewing pull requests

```
eb --review-pr
```

- foo

Previewing pull requests

```
eb --preview-pr
```

- foo

Merging pull requests

```
eb --merge-pr
```

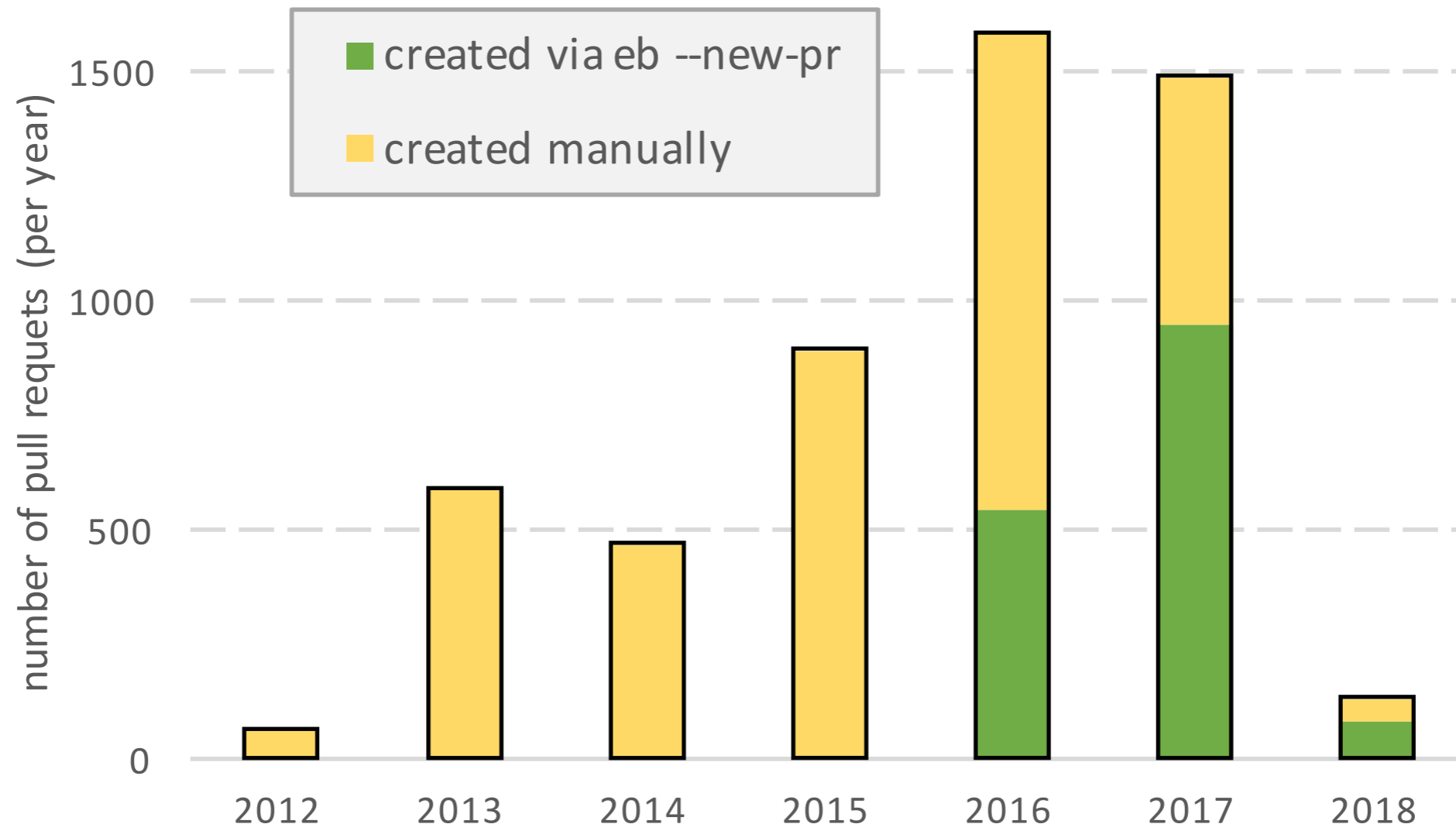
- foo

Improving the performance of `eb --new-pr`

- foo

Impact of `eb --new-pr` on contributions

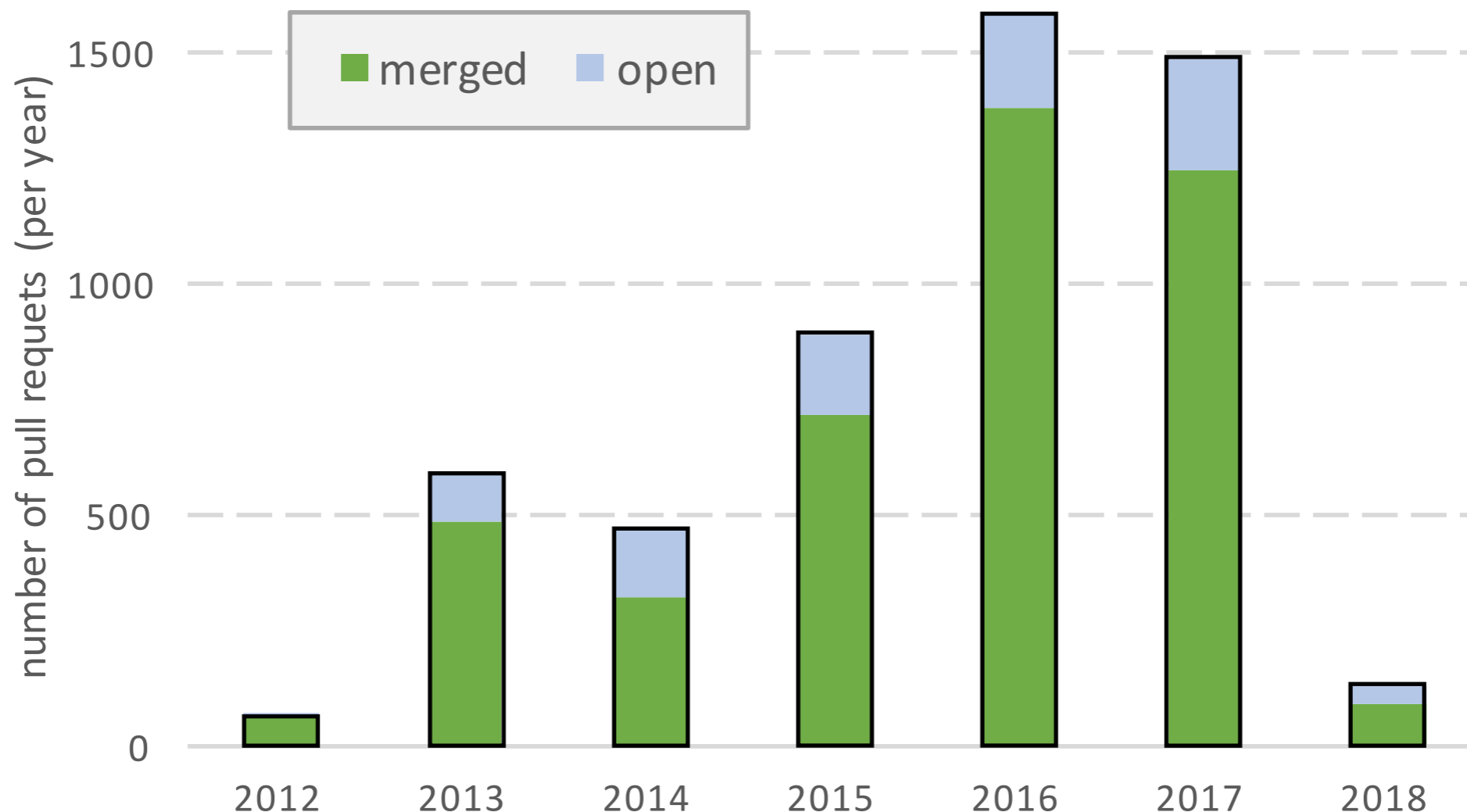
<https://github.com/easybuilders/easybuild-easyconfigs/pulls>



- > 500 PRs opened using `eb --new-pr` in 2016 (34%)
- almost 1000 PRs in 2017 (63%)
- 1.7-1.8x total amount of PRs in 2016/2017 compared to 2015


Impact of `eb --merge-pr` on contributions

<https://github.com/easybuilders/easybuild-easyconfigs/pulls>



- `eb --merge-pr` has helped to keep pace with incoming PRs
- ~85% of PRs gets merged (usually a good reason for the ones that are not)
- would have been difficult without automation due to amount of PRs

Conclusions

- for a significant number of people,  **git** is a challenge
- therefore, it can be a hurdle for potential contributors