How to attribute a single commit to multiple developers?

Asked 8 years, 2 months ago   Active 6 months ago   Viewed 23k times

▲

92

▼

★

17

The way all version control systems I'm familiar with work is that each commit is attributed to a single developer. The rise of Agile Engineering, and specifically pair programming, has lead to a situation where two developers have made a significant contribution to the same task, a bug fix for example.

The issue of attribution won't be too much of a big deal in a work environment since the project manager will be aware of the work the pairs are doing, but what about if two open source contributors decide to pair up and push out some code to a particular project that has no idea they're working together. Is there any way for a version control system like Git to attribute a particular patch to multiple developers?

git   svn   version-control   bazaar

edited Feb 19 '18 at 14:57   asked Sep 16 '11 at 8:38

    Ciro Santilli 新疆改造中   user360907
    心法轮功六四事件

180k   39   681   552

10   This should be split for each version control system. – Ciro Santilli 新疆改造中心法轮功六四事件 Sep 12 '14 at 14:58

8 Answers

▲

44

▼

```
Commit title

Commit body

Co-Authored-By: name <additional-dev-1@example.com>
```

- Supported by Github: https://github.blog/2018-01-29-commit-together-with-co-authors/
- Still waiting on Gitlab: https://gitlab.com/gitlab-org/gitlab-ce/issues/31640
- Used by others: https://git.wiki.kernel.org/index.php/CommitMessageConventions

One problem with this approach is that you can't create a signed key for this group of devs, so you could essentially add anybody to this list even if they didn't work on a feature and github would treat it as if they did. However this shouldn't be an issue in most cases.

e.g. `Co-Authored-By: Linus Torvalds <torvalds@linux-foundation.org>`

With normal authors or signing groups (the old method) you would see it's not signed and know that you can't trust the commit. However there is no signing process on co-authors.

Mostly outdated answer:

One solution would be to set a name for the pair:

```
 git config user.name "Chris Wilson and John Smith"
```

Here is a related bug report with other temporary solutions:

Bug git-core: Git should support multiple authors for a commit

edited Apr 27 at 3:49   answered Sep 16 '11 at 8:51

                                             Gerry
                                             8,382   3   32   45

What about if you don't know the person's Email address? Any way to link GitHub username etc? – Aaron Franke May 6 at 8:14

If they made a previous commit you can just check that: `git show <COMMIT_ID> --format=email` – Gerry May 8 at 4:32

1   This seems to already be working in GitLab (see gitlab.com/gitlab-org/gitlab-foss/merge_requests/17919) – Meiogordo Sep 17 at 16:05

A git convention is to use *Co-Authored-By* at the end of the commit message (git kernel: Commit Message Conventions, referring to Openstack Commit Messages). This is also one of the solutions on the git-core bug linked in Gerry's answer.

In that comment on May 5, 2010, Josh Triplett also suggests implementing corresponding support in git.

As Llopis pointed out in a comment, GitHub announced support for this on their blog on Jan 29, 2018: Commit together with co-authors (details).

edited Feb 1 '18 at 1:10

answered Jan 25 '17 at 9:01

K   Kariem
         2,898   2   31   52

---

7   This is now supported by GitHub. – llrs Jan 31 '18 at 8:06

     @Llopis Just saw the blog post and came here to update my answer. Good to see someone was faster :) – Kariem Jan 31 '18 at 20:00

     I wrote a simple git plugin to make it easier to manage these Co-authored-by trailers automatically: github.com/cac04/git-pair – c-- Oct 27 '18 at 13:48

     This approach looks like a kludge (cf. the Bazaar way). And there's a principal author this way, to which the co-authors are just an addition, if I understand it correctly. – Ruslan Oct 22 at 9:42 ✎

For Bazaar:

26
```
bzr commit --author Joe --author Alice --author Bob
```

Those names will be shown in the log separately from committer name.

edited Sep 19 '11 at 9:04

answered Sep 16 '11 at 9:42

         bialix
         12.7k   8   40   60

---

git-pair

19
https://github.com/pivotal/git_scripts#git-pair

This simple script from Pivotal to automate Git pair programming attribution

```
# .pairs - configuration for 'git pair'
pairs:
  # <initials>: <Firstname> <Lastname>[; <email-id>]
  eh: Edward Hieatt
  js: Josh Susser; jsusser
  sf: Serguei Filimonov; serguei
email:
  prefix: pair
  domain: pivotallabs.com
  # no_solo_prefix: true
#global: true
```

and then:

```
git pair sp js
```

sets:

```
user.name=Josh Susser & Sam Pierson
user.email=pair+jsusser+sam@pivotallabs.com
```

for you.

edited Feb 19 '18 at 14:57   answered Sep 12 '14 at 15:02

Ciro Santilli 新疆改造中
心法轮功六四事件
180k   39   681   552

▲

13

▼

git distinguishes between a commit's `author` and `committer` [1]. You could use it as a work-around, e.g. sign yourself as the `committer` and your co-author as the `author` :

```
GIT_COMMITTER_NAME='a' GIT_COMMITTER_EMAIL='a@a' git commit --author 'b <b@b>'
```

This way, both you and your co-author will be recorded in the git history. Running `git log --format=fuller` , will give you something like:

```
AuthorDate: Tue Apr 12 06:53:41 2016 +0100
Commit:     a <a@a>
CommitDate: Tue Apr 12 09:18:53 2016 +0000

    Test commit.
```

[1] Difference between author and committer in Git?

edited Mar 26 at 7:16

answered May 11 '16 at 12:11

Jakub Kukul

3,839   1   26   29

We add our names to each commit message at the end as a convention eg :   `Implemented cool feature <Aneesh | Hiren>`

2

edited Sep 19 '11 at 9:44

answered Sep 16 '11 at 8:56

tripleee

107k   16   156   208

Aneesh

808   9   17

---

4   This is similar to the git convention `Co-Authored-By` I mentioned in a separate answer – Kariem Jan 25 '17 at 9:23

Try git-mob, we built it for attributing co-authors on commits.

2

E.g.

```
git mob <initials of co-authors>
git commit
git solo
```

answered Apr 8 '18 at 22:54

Dennis

38.4k   19   115   113

2

to set an alias in order to create co-autored commits as follows:

```
$ git co-commit -m "Commit message" --co "co-author <co-author-email>"
```

Using this approach, you are able to create co-authored commits without a graphical interface.

edited Jun 10 '18 at 23:44   answered Jun 10 '18 at 23:17

Thiago Nobayashi
101   1   2

---

4   From what I can see, this is a git alias that adds "Co-authored-by: " plus the "co-author <co-author-email>" to the end of the commit message. – Kariem
Jul 10 '18 at 21:33