



WRITE HISTORY, KEEP A DIARY

DEVOPS PRO EUROPE 2019 21/3/2019

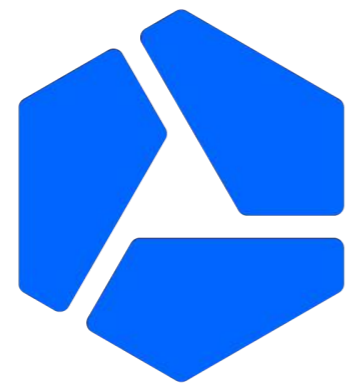
Jachim Coudenys

jachim.coudenys@combell.group

[@coudenysj](#)



Jachim Coudenys



combell

A close-up photograph of a hand holding a silver pen, writing on a spiral-bound notebook. The notebook is open, and the pages are filled with cursive handwriting in blue ink. The text is mostly illegible due to being out of focus. The spiral binding is visible on the right side. The overall scene suggests a person writing a diary or journal.

WRITE HISTORY
KEEP A DIARY

A diary is a record (originally in handwritten format) with discrete entries arranged by date reporting on what has happened over the course of a day or other period.

- <https://en.wikipedia.org/wiki/Diary>

Diaries undertaken for institutional purposes play a role in many aspects of human civilization, including government records, business ledgers and military records.

- <https://en.wikipedia.org/wiki/Diary>

slido

Active poll

HALL 5

Display next survey question

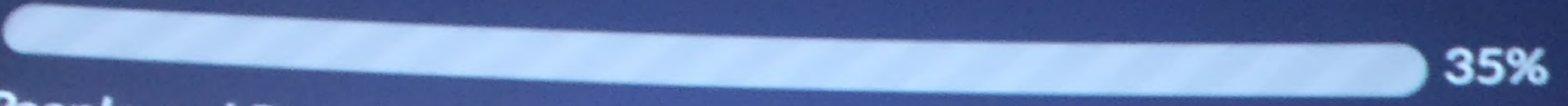
What is your primary occupation field:

138

Development



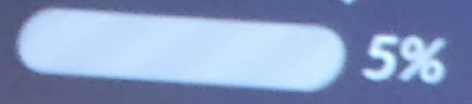
IT Operations



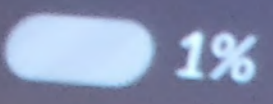
People and Process Management



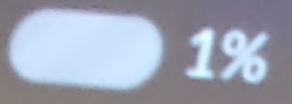
Consultancy



QA



Business



Join at
slido.com
#devops2019

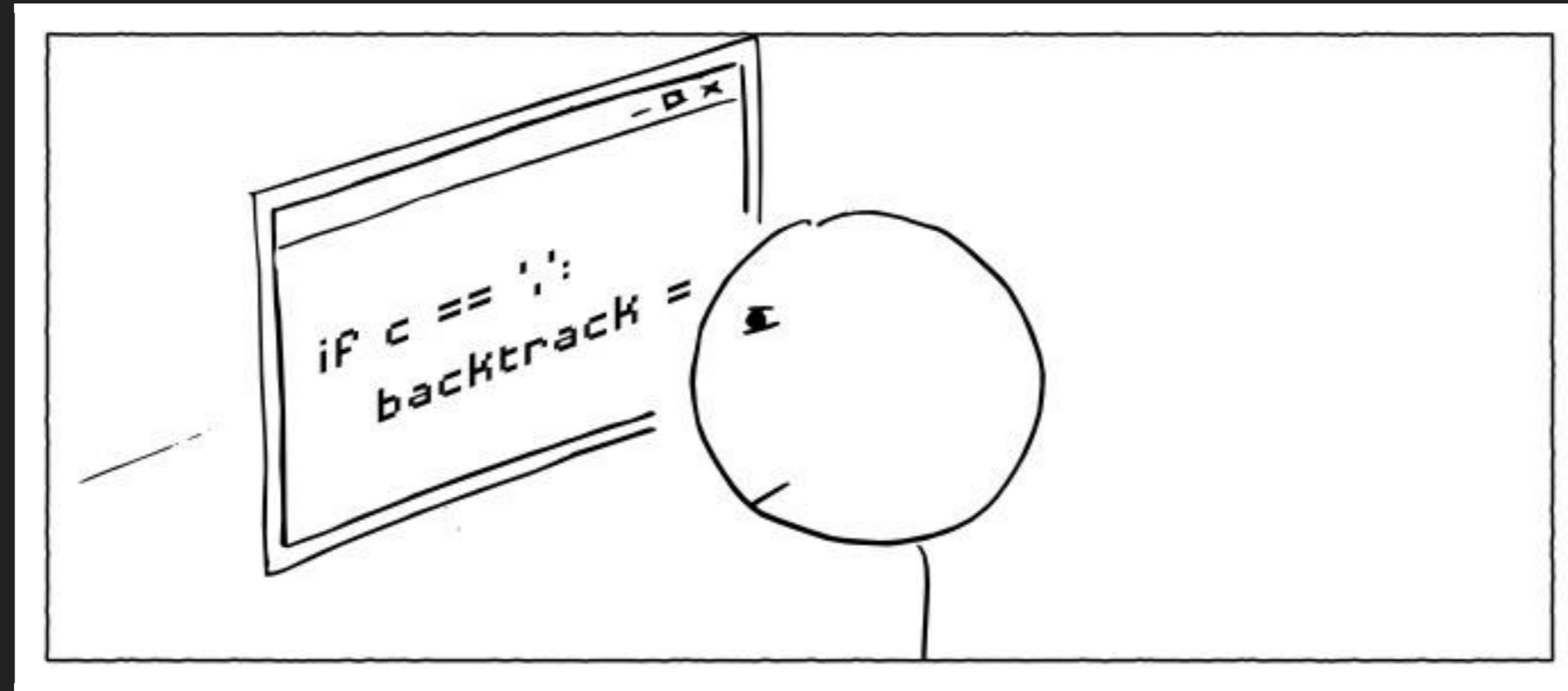


CODE

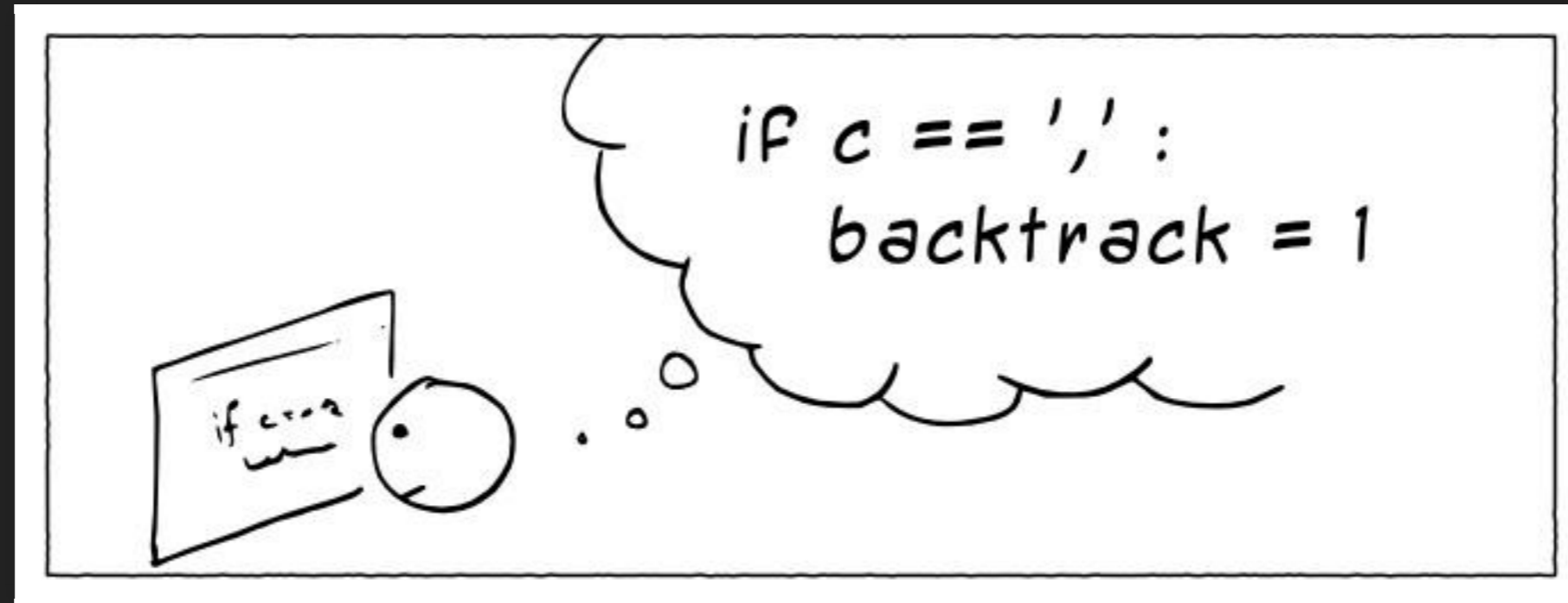
*The primary goal of a software developer
should be to communicate their intent to
future developers*

— Louise Crow / @crowbot

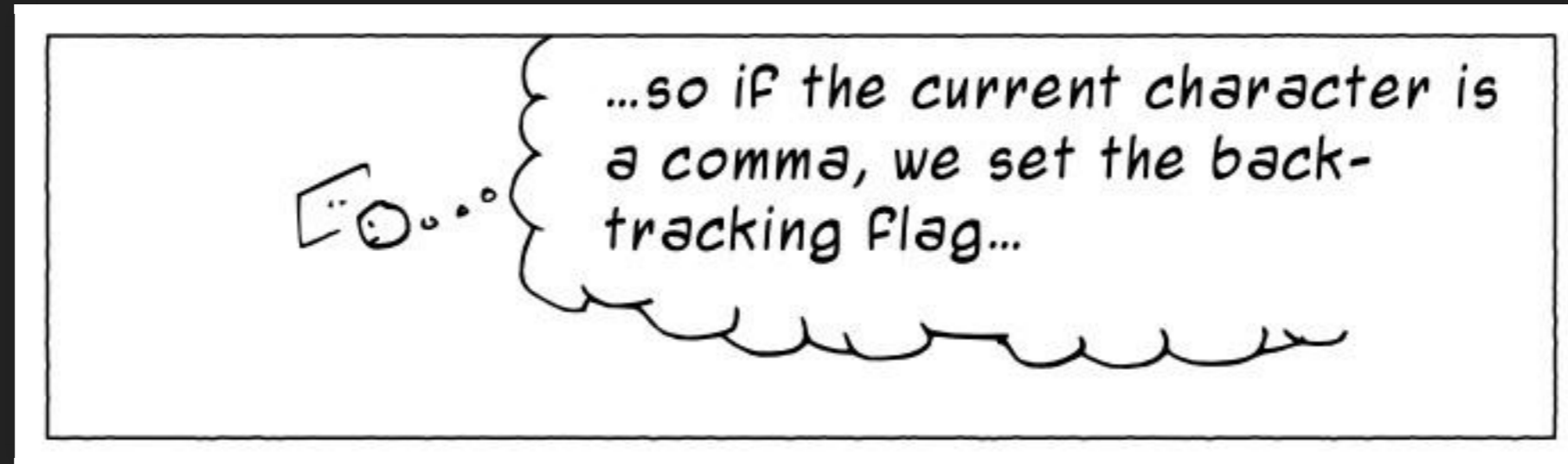
CONTEXT



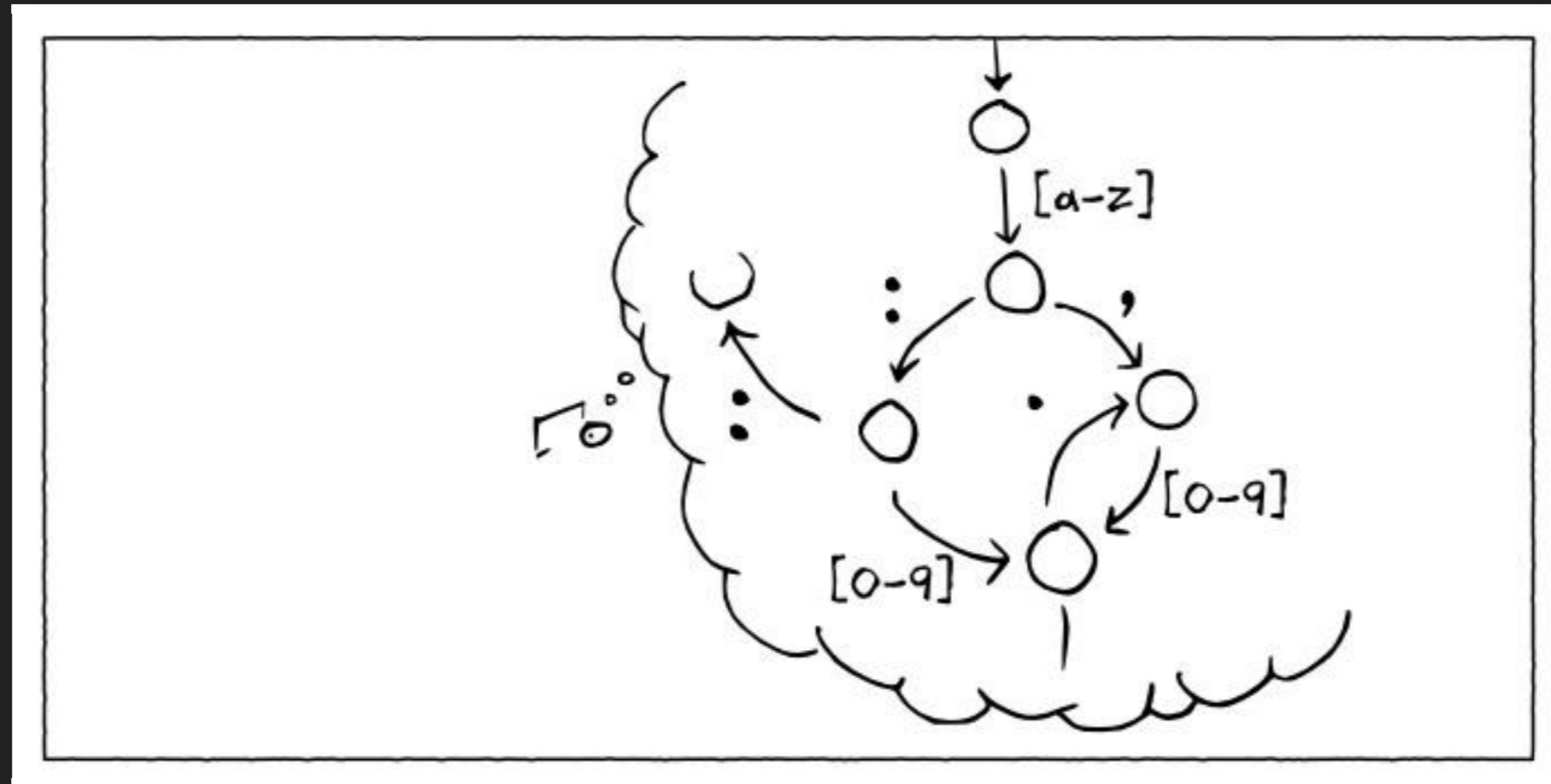
<http://heeris.id.au/2013/this-is-why-you-shouldnt-interrupt-a-programmer/>



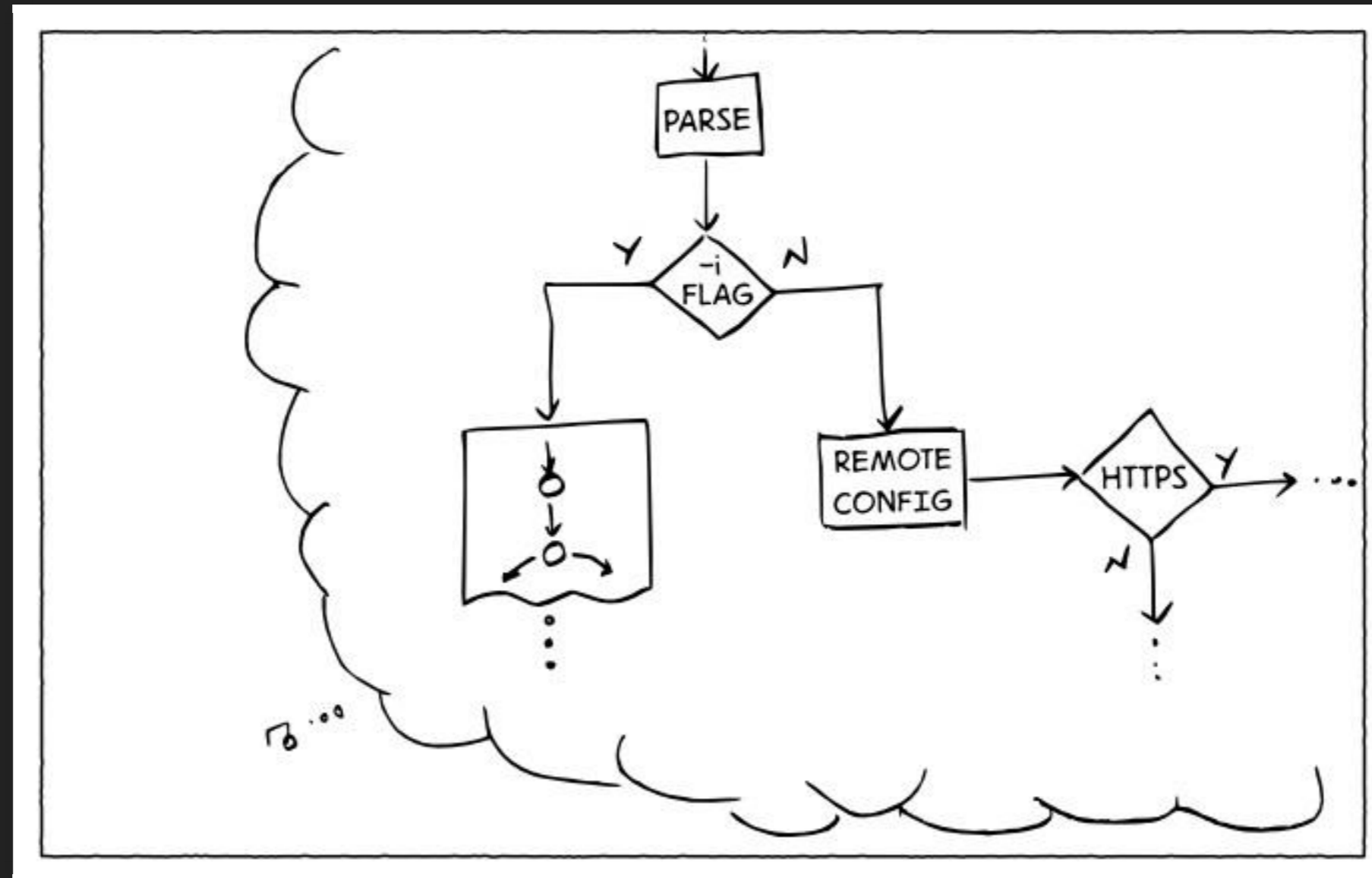
<http://heeris.id.au/2013/this-is-why-you-shouldnt-interrupt-a-programmer/>



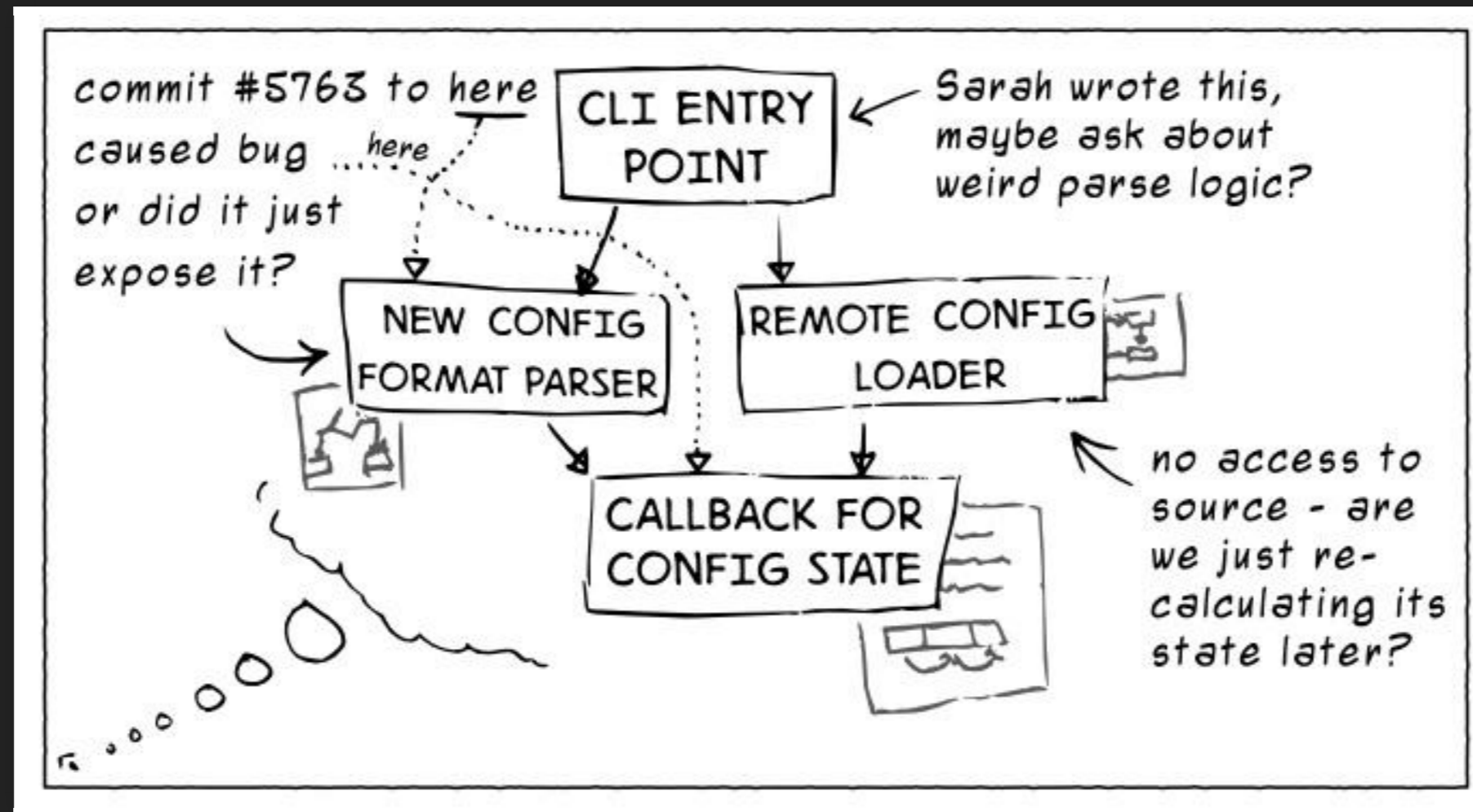
<http://heeris.id.au/2013/this-is-why-you-shouldnt-interrupt-a-programmer/>

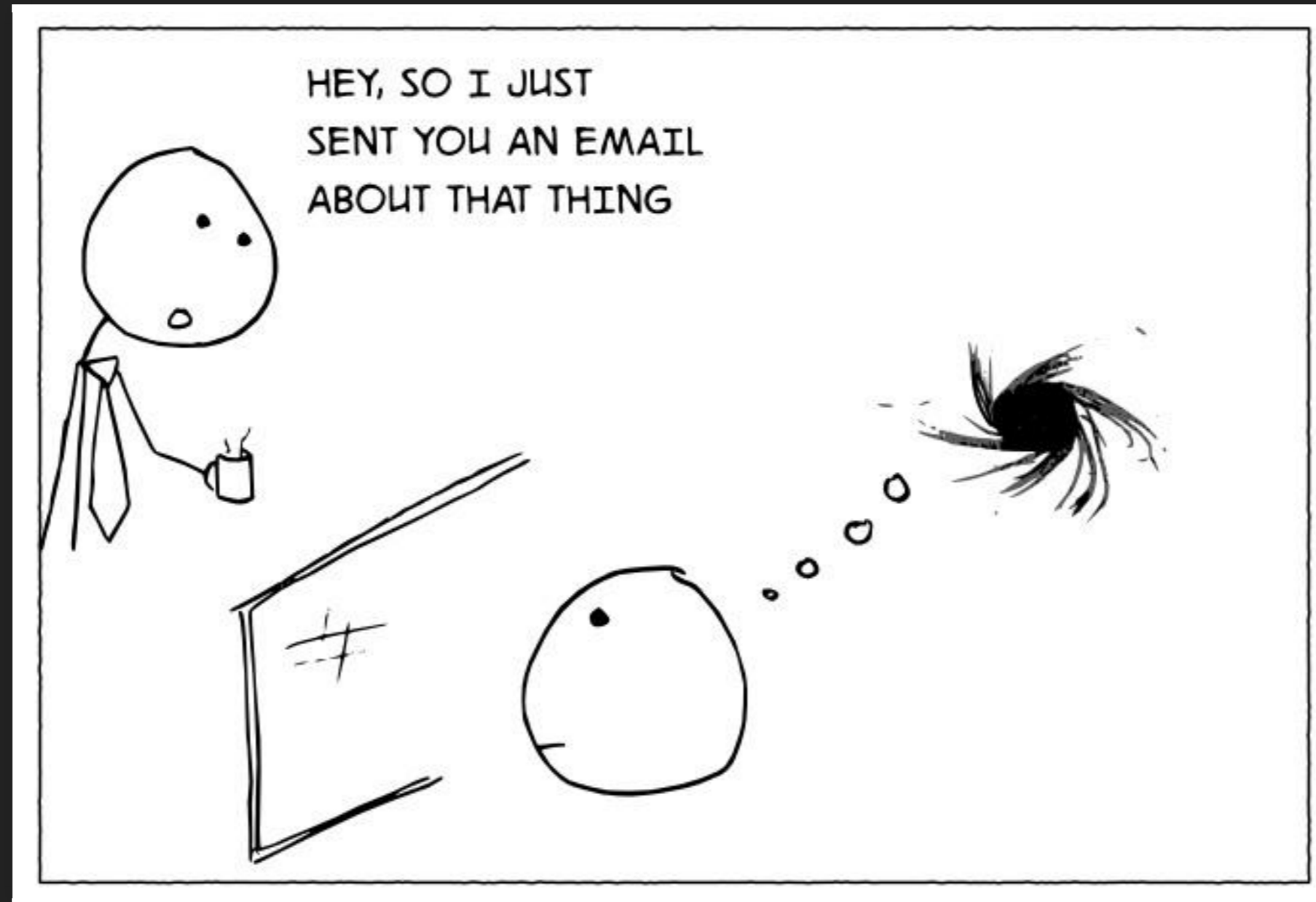


<http://heeris.id.au/2013/this-is-why-you-shouldnt-interrupt-a-programmer/>

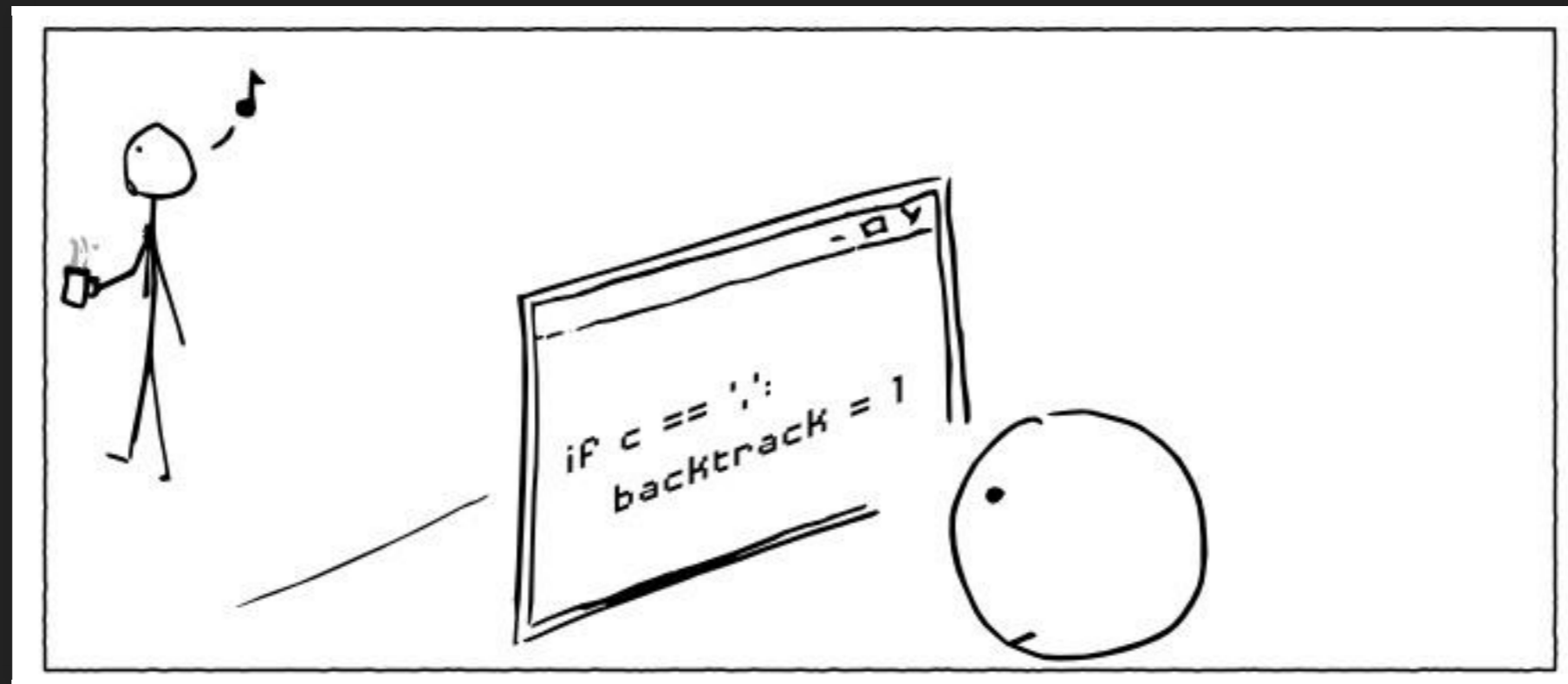


<http://heeris.id.au/2013/this-is-why-you-shouldnt-interrupt-a-programmer/>





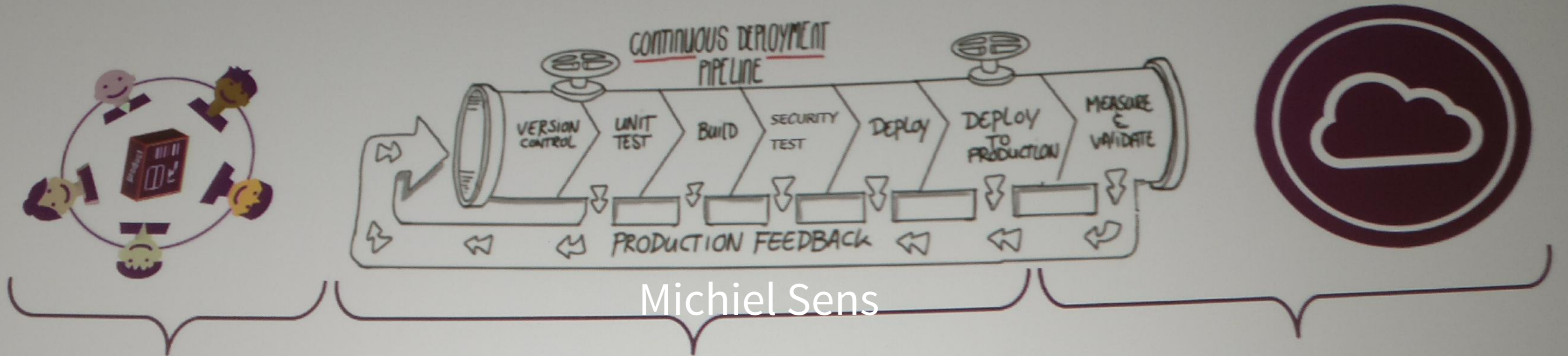
<http://heeris.id.au/2013/this-is-why-you-shouldnt-interrupt-a-programmer/>



<http://heeris.id.au/2013/this-is-why-you-shouldnt-interrupt-a-programmer/>

VERSION CONTROL SYSTEM

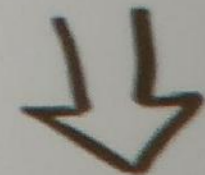
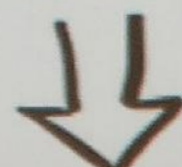
Everything **measured** for effectiveness



- Employee Satisfaction
- DevOps Maturity

- Deploy frequency
- Lead time for changes
- Change Failure Rates
- Mean Time to Recover

- Actual Value (\$\$\$)
- Customer satisfaction (NPS, ...)



Continuous Improvements



Strategic Direction

Re-establishing the context of a piece of code is wasteful. We can't avoid it completely, so our efforts should go to reducing it [as much] as possible. Commit messages can do exactly that and as a result, a commit message shows whether a developer is a good collaborator.

— <http://who-t.blogspot.be/2009/12/on-commit-messages.html>

RICHARDSON MATURITY MODEL

Glory of REST



Level 3: Hypermedia Controls




Level 2: HTTP Verbs

Level 1: Resources

Level 0: The Swamp of POX



JACHIM'S COMMIT MATURITY MODEL

	Commits	Messages	CLI & Tools
Level 0		git basics	
Level 1			
Level 2			
Level 3			

LEVEL 0



GIT BASICS

Local Computer

Checkout

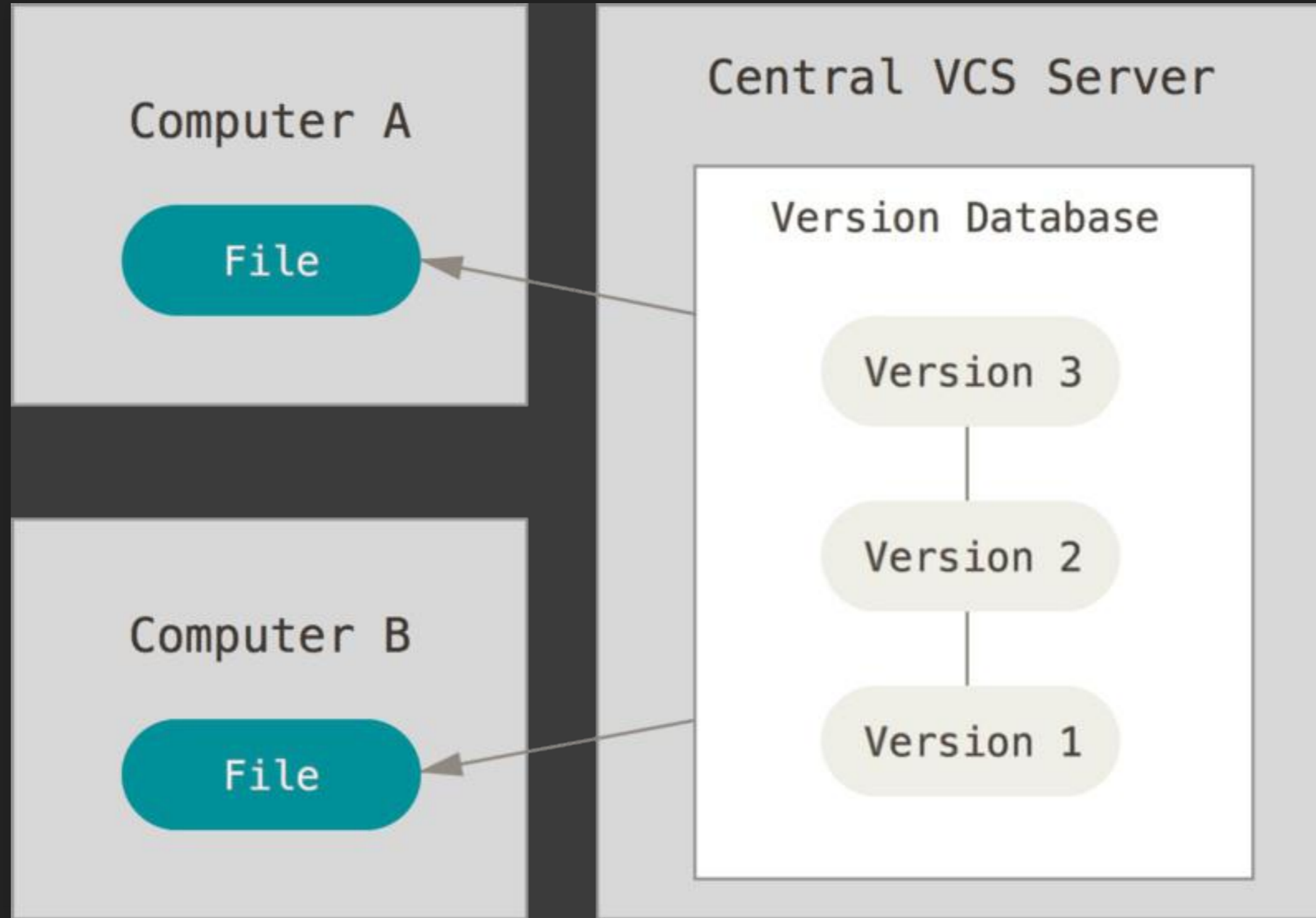
File

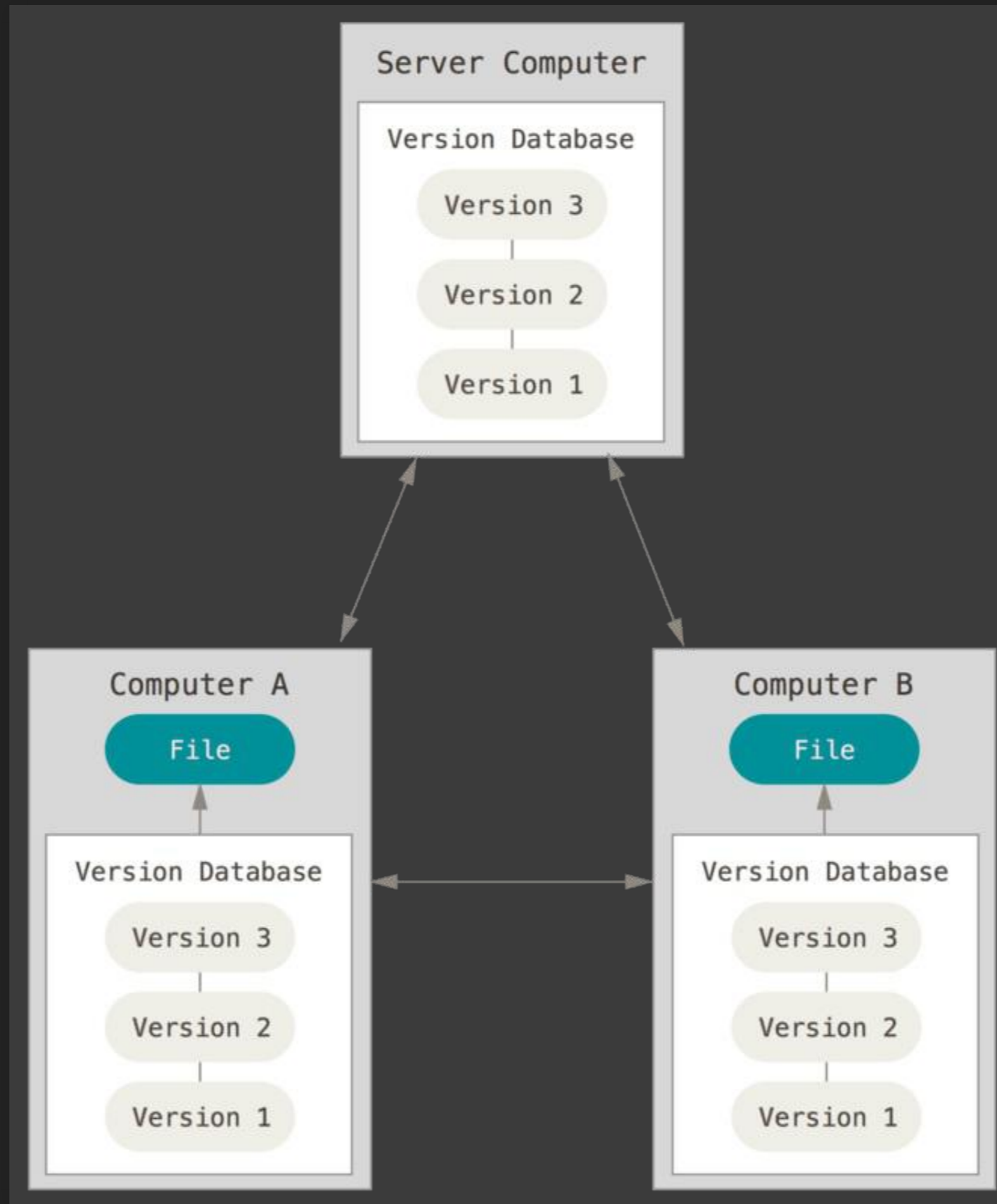
Version Database

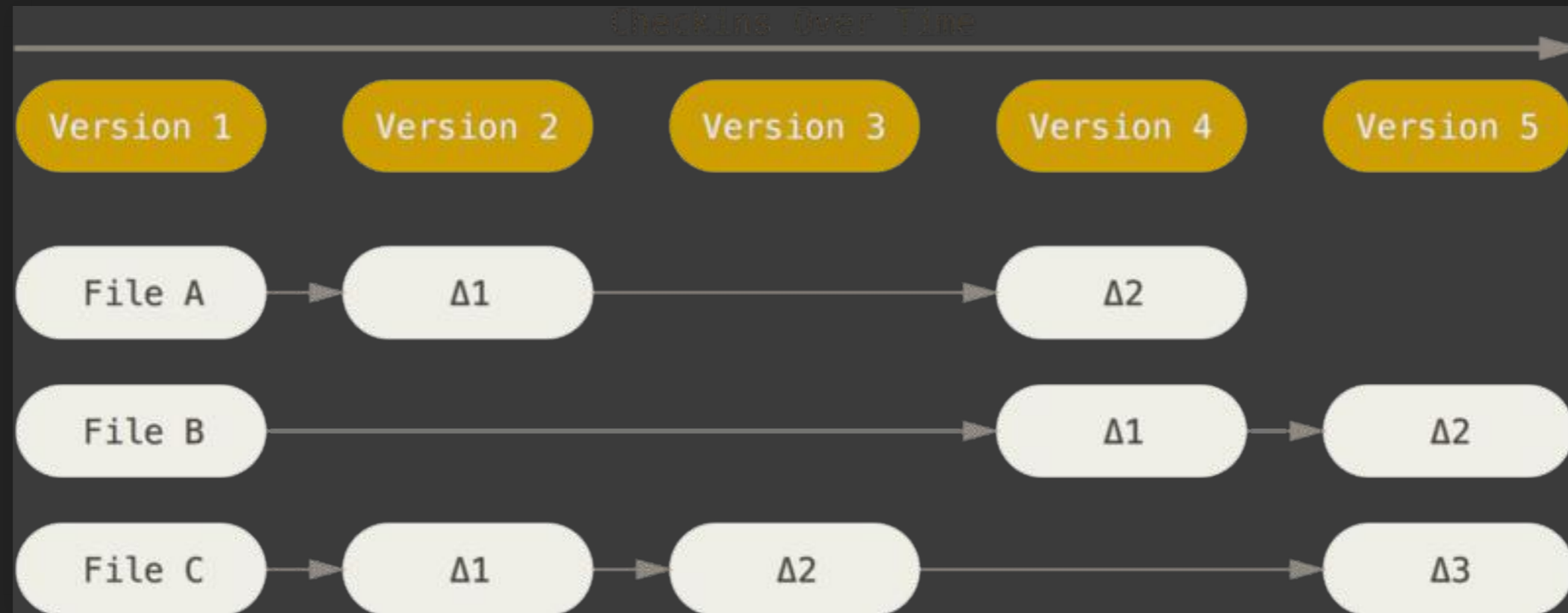
Version 3

Version 2

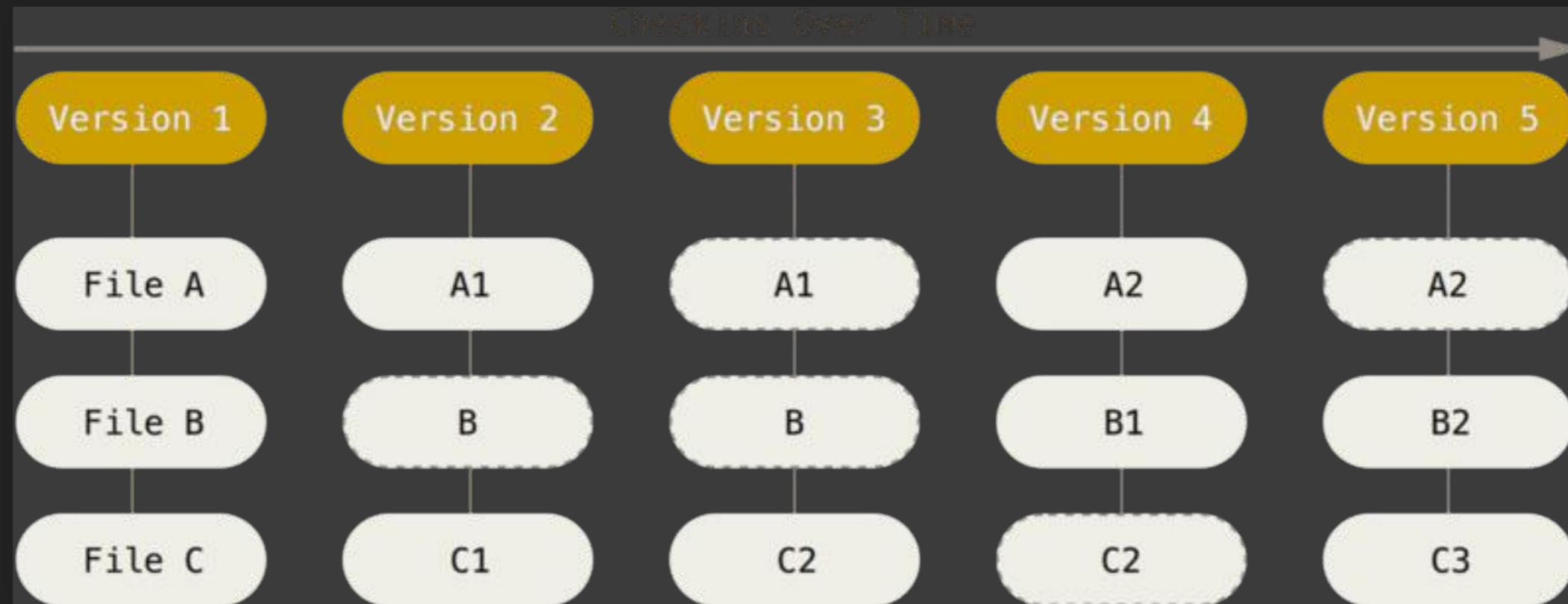
Version 1



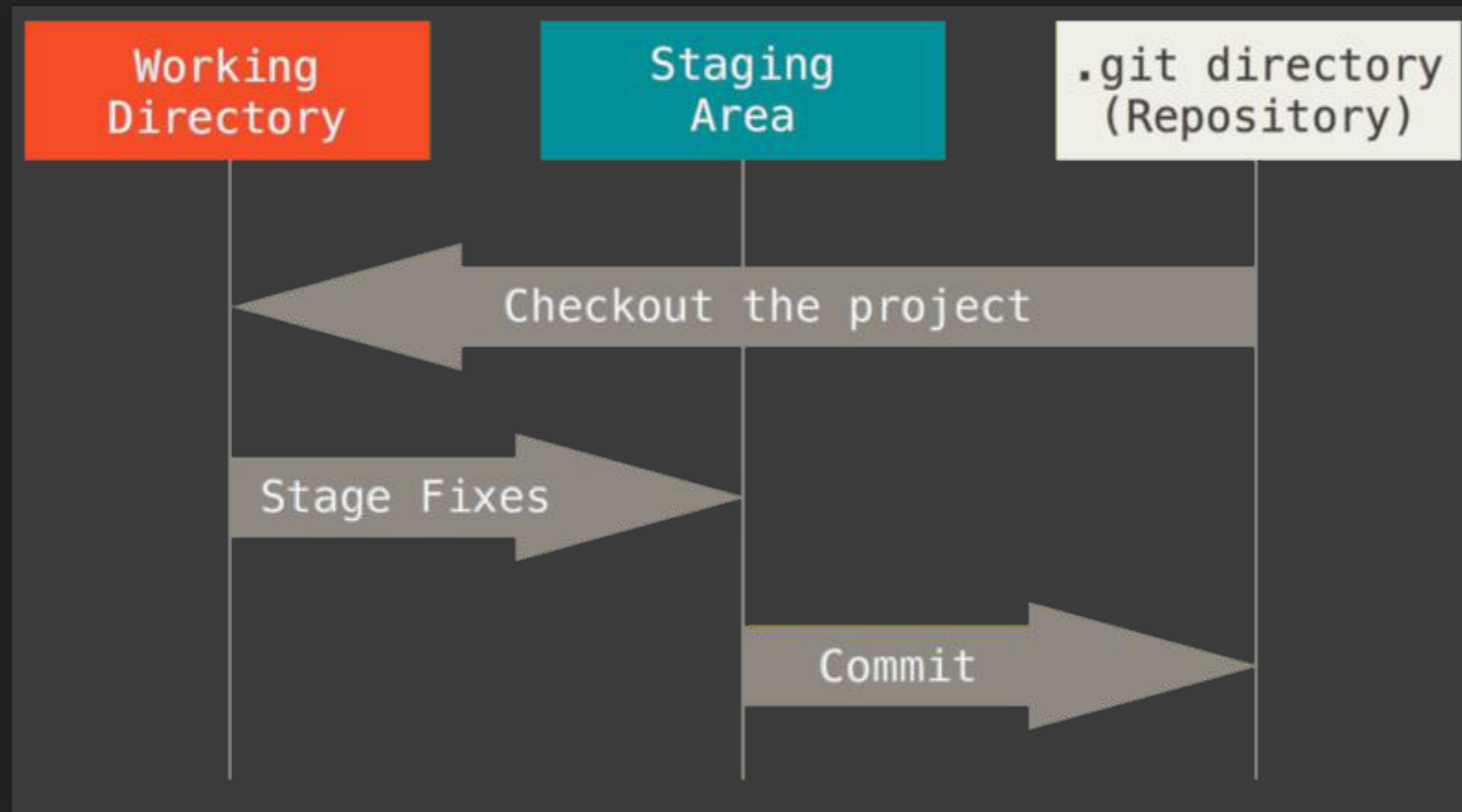


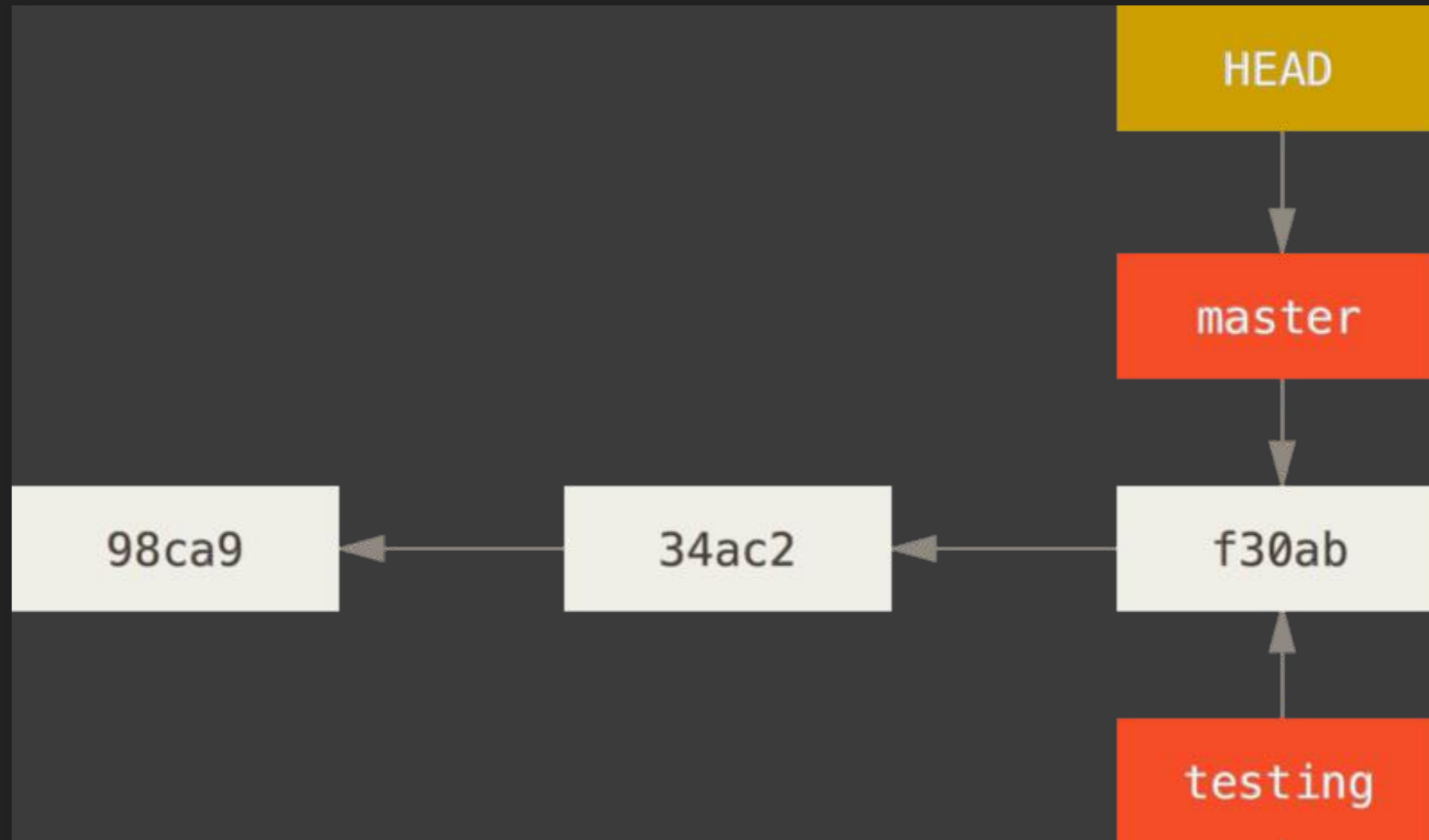


<https://git-scm.com/book/en/v2/>

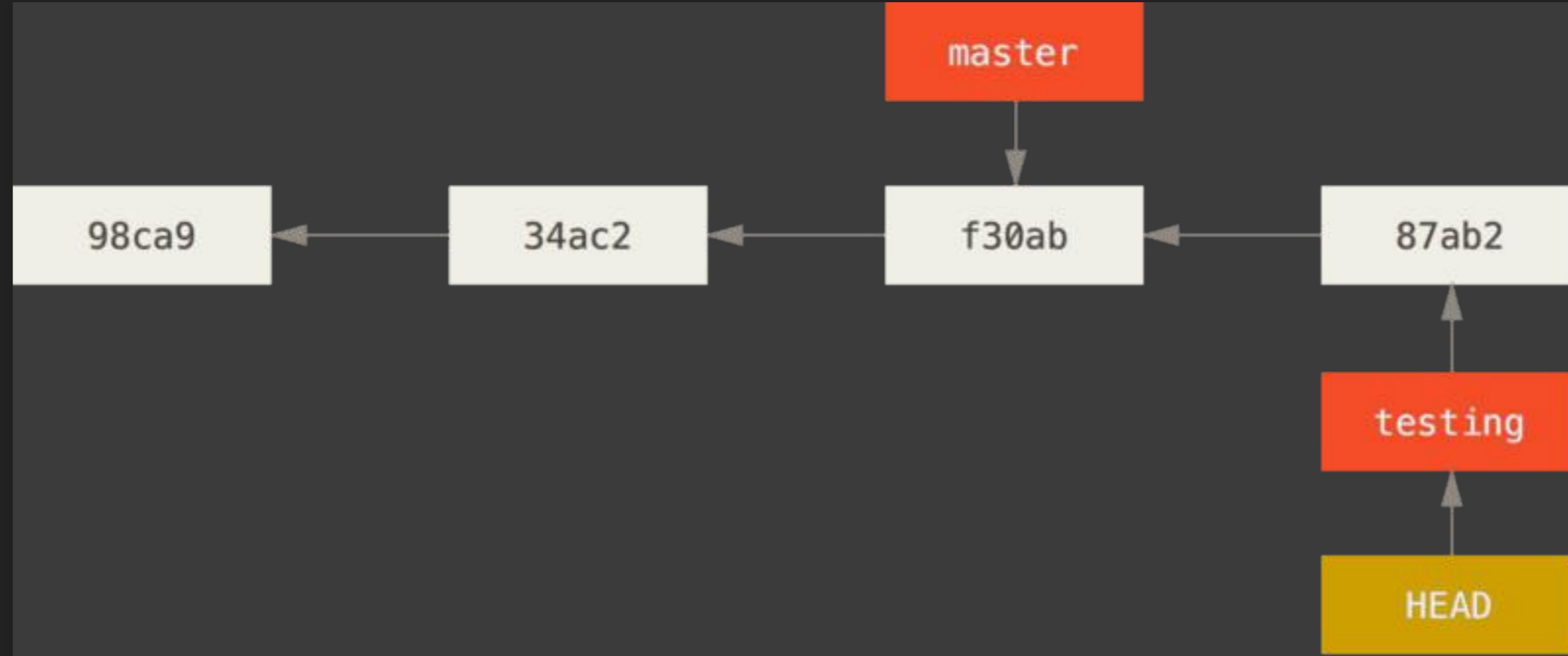


<https://git-scm.com/book/en/v2/>

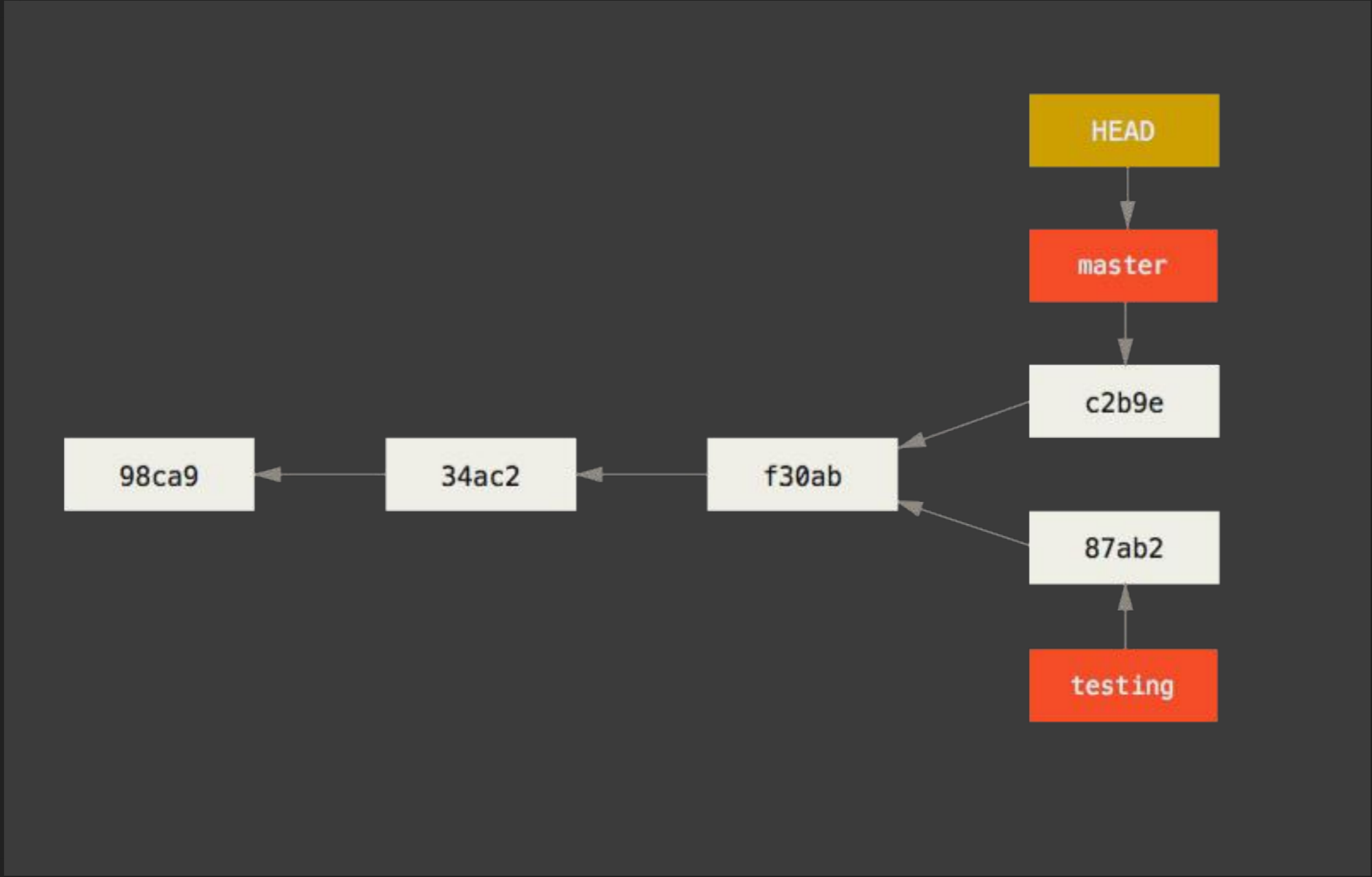




<https://git-scm.com/book/en/v2/>



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



SWAMP OF PORC

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

<https://xkcd.com/1296/>

LEVEL 1



ATOMIC COMMITS

- Small
- One problem / case at a time

COMMIT MESSAGES

- What and why instead of how
- Use subject and body
(if required, small changes don't need a body)

**200,000,000 Pull Requests
From 31 Million Developers**

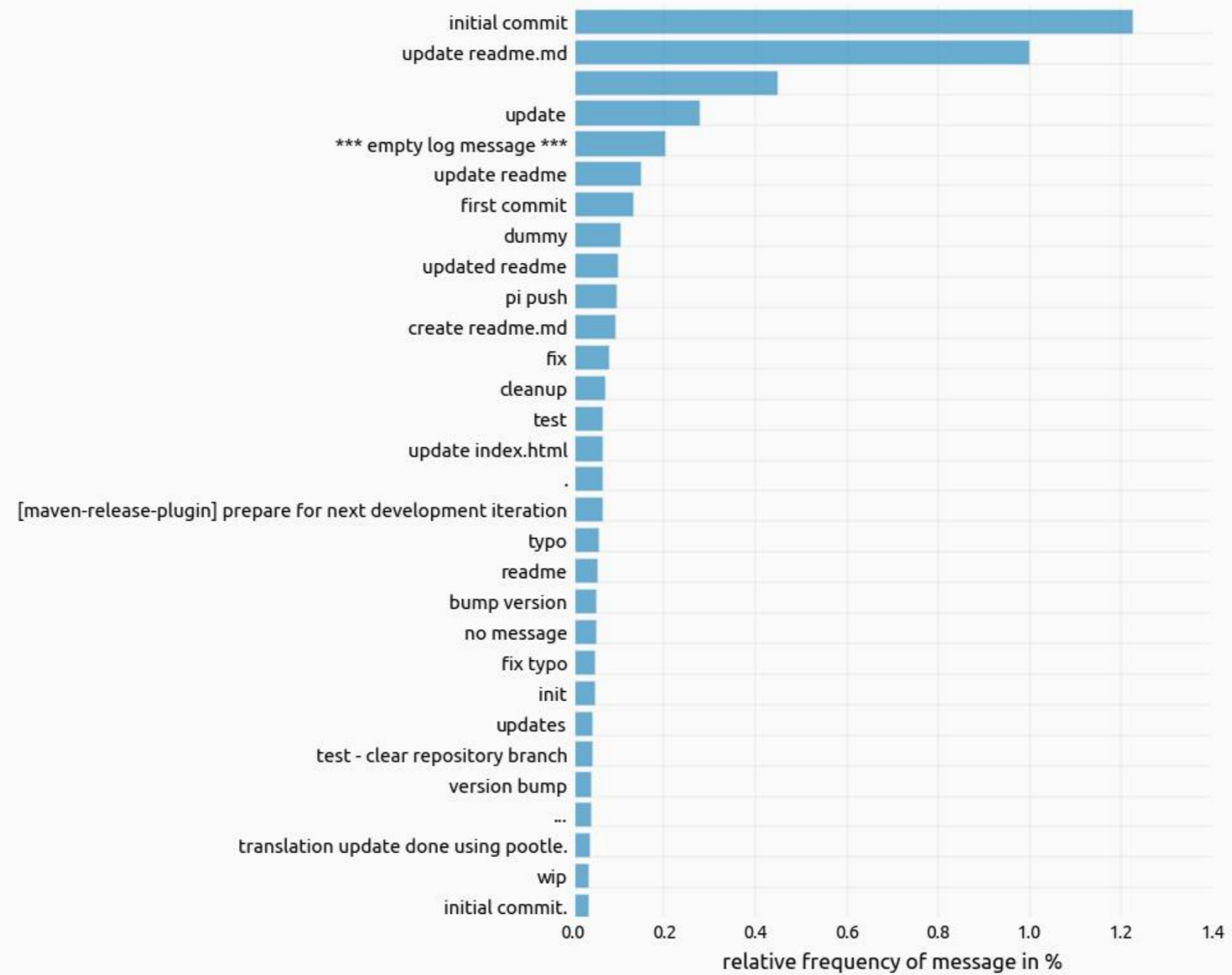
Johannes Nicolai



Join a



The 30 most frequent commit messages on GitHub till June 2016



Based on 151,573,032 commit messages, that were lowercased and trimmed of whitespace before aggregation. Messages starting with 'merge' were ignored.
Author: Ramiro Gómez - ramiro.org • Data: GitHub on BigQuery - bigquery-public-data:github_repos.commits

git log

commit [601b665ba356a1b2626b62b63aa23660a507afbd](#)

Author: Jorge González <scrub.mx@gmail.com>

Date: Thu Aug 17 23:47:12 2017 -0500

Fix in comment in config/app.php

I know it is a small thing but it's driving me crazy!

...

git log --grep "queue"

commit d1fd41882696b6220cca7abfce57a1bb45f5656c

Author: Drew Budwin <drew@budw.in>

Date: Fri Jan 20 19:51:53 2017 -0500

Fixed a typo in a block comment queue.php

Normally, words that start with vowels (like "unified") are preceded by "an" instead of "a." However,

Source: https://www.a-or-an.com/a_an/unified

...

git log -S bcrypt

commit 792dcd48c8ad9887f08db6ad9efd0890603ddd53

Author: Adam Wathan <adam.wathan@gmail.com>

Date: Sun Aug 21 14:23:50 2016 +0200

Compute hash only once and store in static variable

diff --git a/database/factories/ModelFactory.php b/database/factories/ModelFactory.php

...

'email' => \$faker->safeEmail,

- // Use a precomputed hash of the word "secret" instead of using bcrypt directly.

- // Since bcrypt is intentionally slow, it can really slow down test suites in

- // large applications that use factories to generate models in many tests.

- 'password' => '\$2y\$10\$PCCpaPQ69KQ1fdrAIL0eptYCCG/s/NmQZizJfVdB.QOXUn5mGE6',

+ 'password' => \$password ?: \$password = bcrypt('secret'),

LEVEL 2



PURE ATOMIC COMMITS

- Code formatting ALWAYS in separate commit!
- Create different commits from a “dirty” workspace
git add -p (--patch)
git add -i (--interactive)

REWRITE HISTORY IF NECESSARY

- Git is
 - distributed
 - private
 - flexible
- `git commit --amend`
- `git rebase -i`

GOOD COMMIT MESSAGES

- Length
 - Do you like reading rambling paragraphs on one long line? No? T
 - 50 characters
 - 72 characters wrap
- Stop using `git commit -m`
- Reference to tickets, pattern pages, manual pages, etc....
with full urls

COMMAND LINE: LENGTH

- git log --pretty=oneline
- git rebase --interactive
- merge.summary
- git shortlog
- etc...



Commit changes

 **ProTip!** Great commit summaries contain fewer than 50 characters. Place extra information in the extended description.

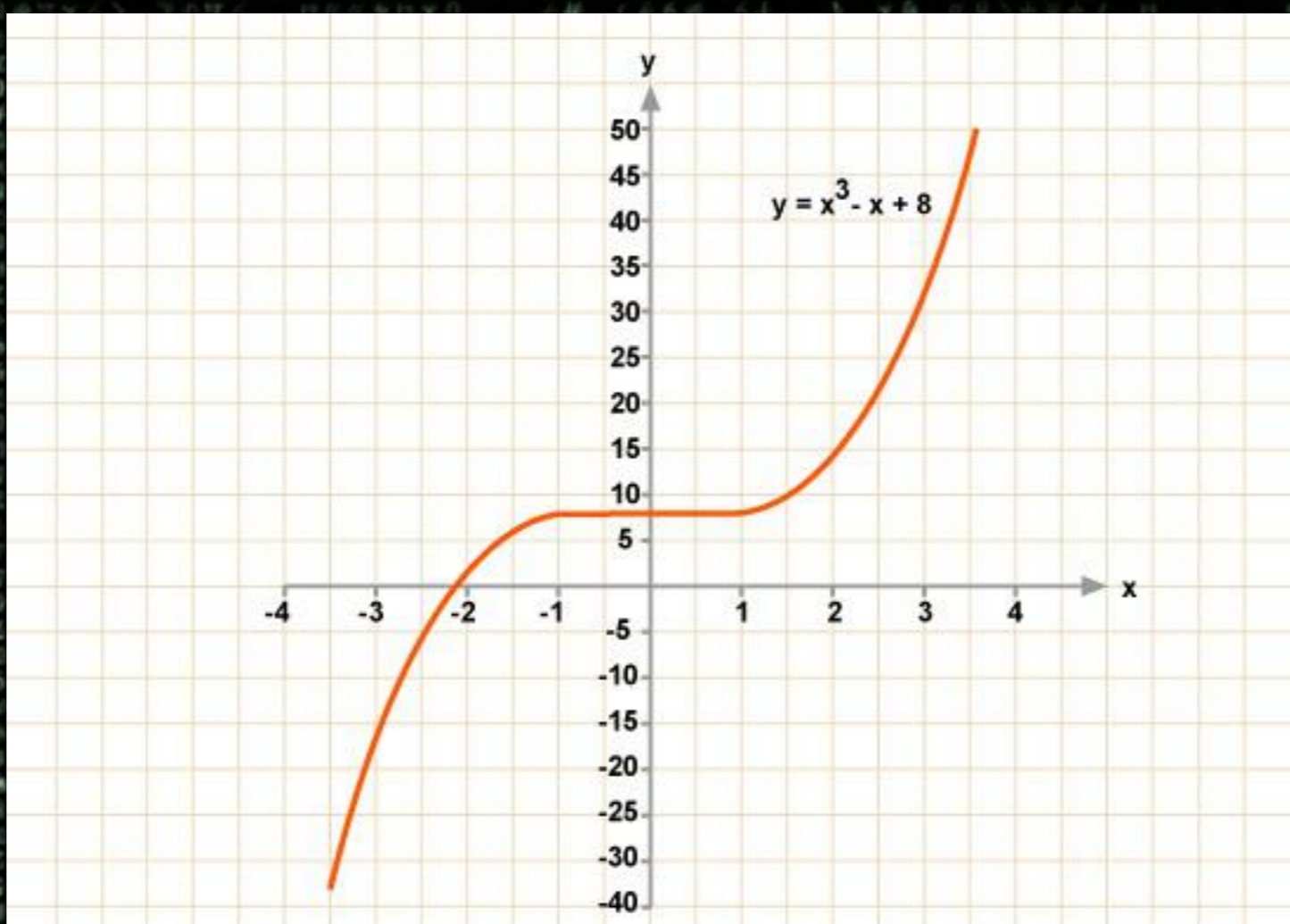
This is a commit message that is way to long and does not

Add an optional extended description...

- Commit directly to the `gh-pages` branch.
- Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

Cancel



COMMAND LINE: GRAPHS

Commit structure

```
$ git log --graph --oneline
* 76fcd9d50 Merge #9309: [qa] Wallet needs to stay unlocked for whole test
|
| * 9359f8ad3 Wallet needs to stay unlocked for whole test
| * | a1dcf2e10 Merge #9240: Remove txConflicted
| \
| * | a874ab5cc remove internal tracking of mempool conflicts for reporting to wallet
| * | bf663f8e9 remove external usage of mempool conflict tracking
| * | | d38b0d7a6 Merge #9307: Remove undefined FetchCoins method declaration
| \ \
| * | | fe41f5836 Remove undefined FetchCoins method declaration
| * | | | 815640ec6 Merge #9295: [Wallet] Bugfix: Fundrawtransaction: don't terminate when keypool is
```

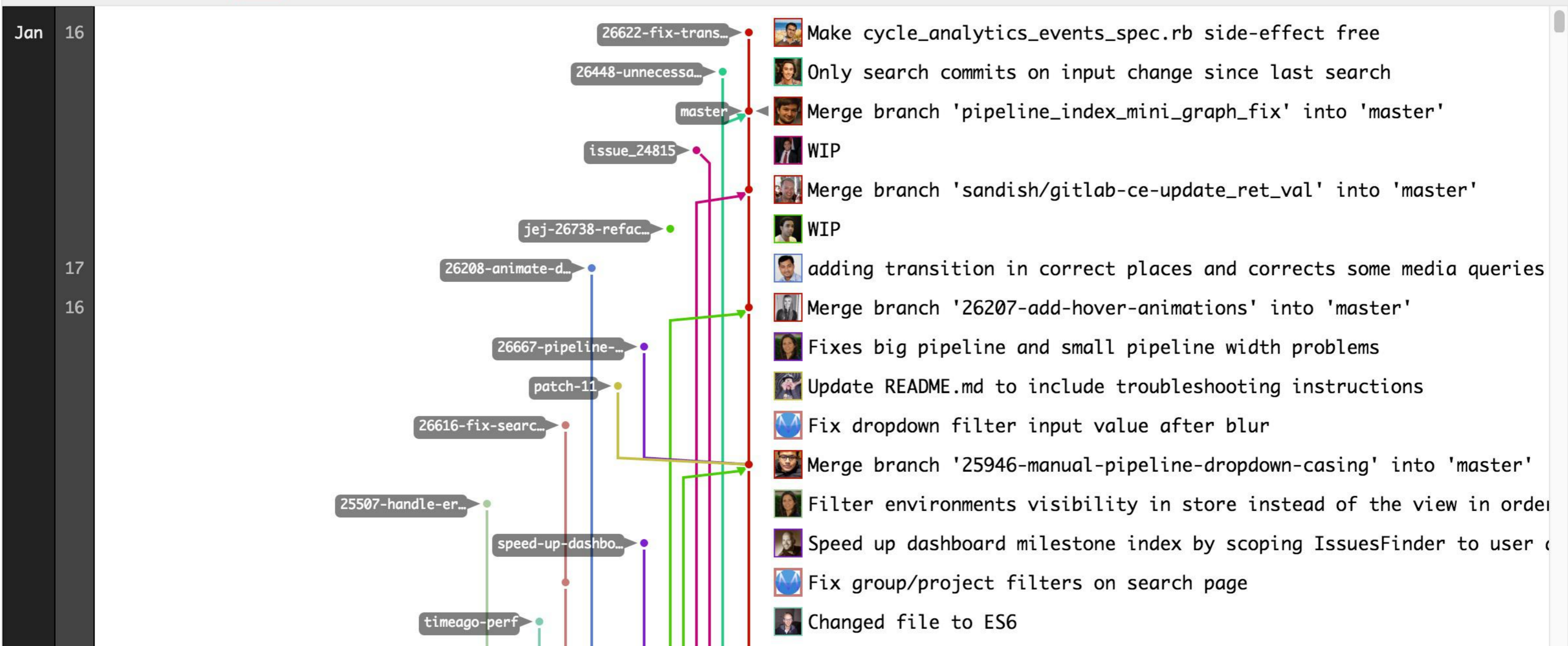
master

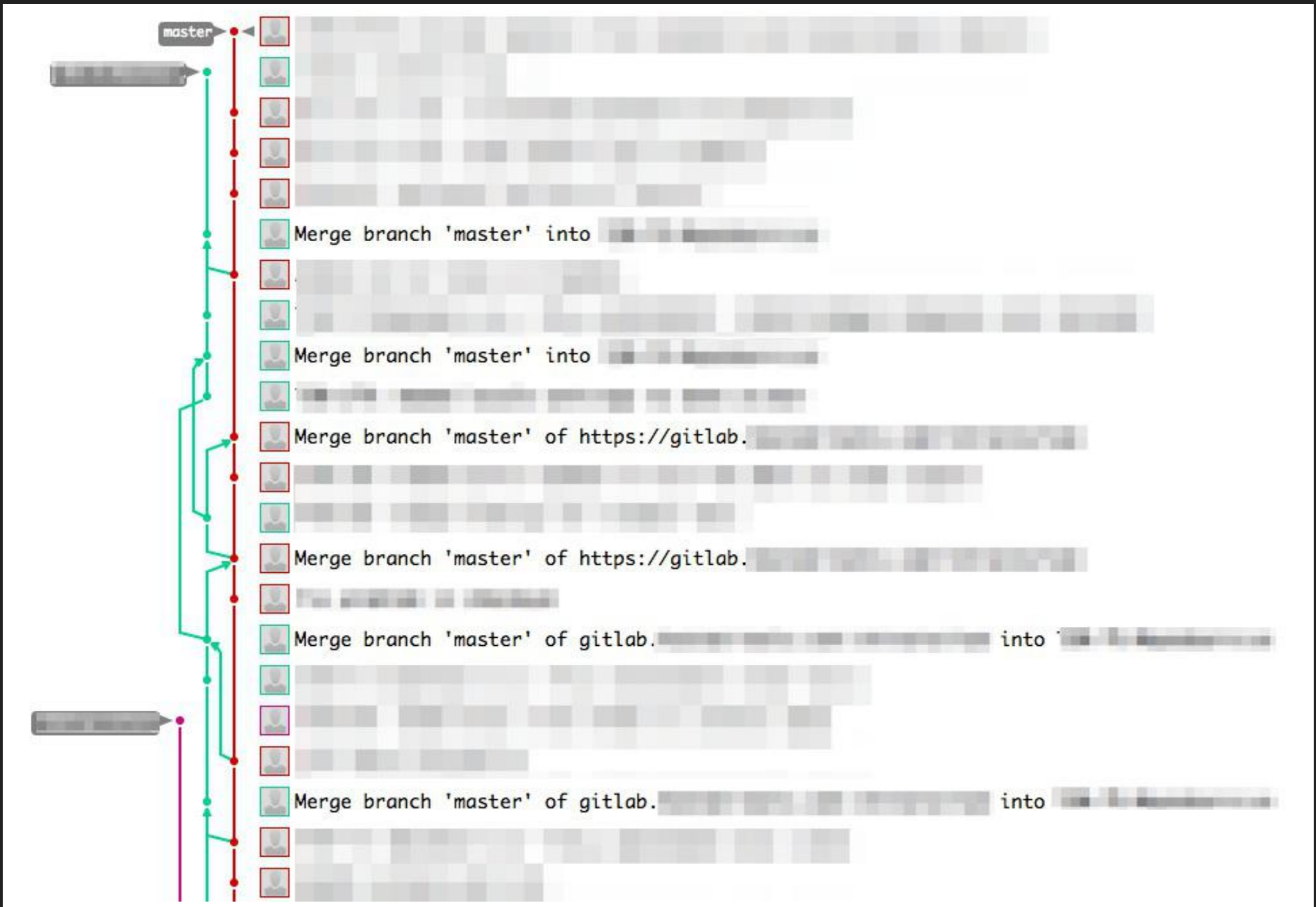
You can move around the graph by using the arrow keys.

Git revision



Begin with the selected commit





COMMAND LINE: MERGE BUBBLES

- *Merge branch 'master' of github.com*

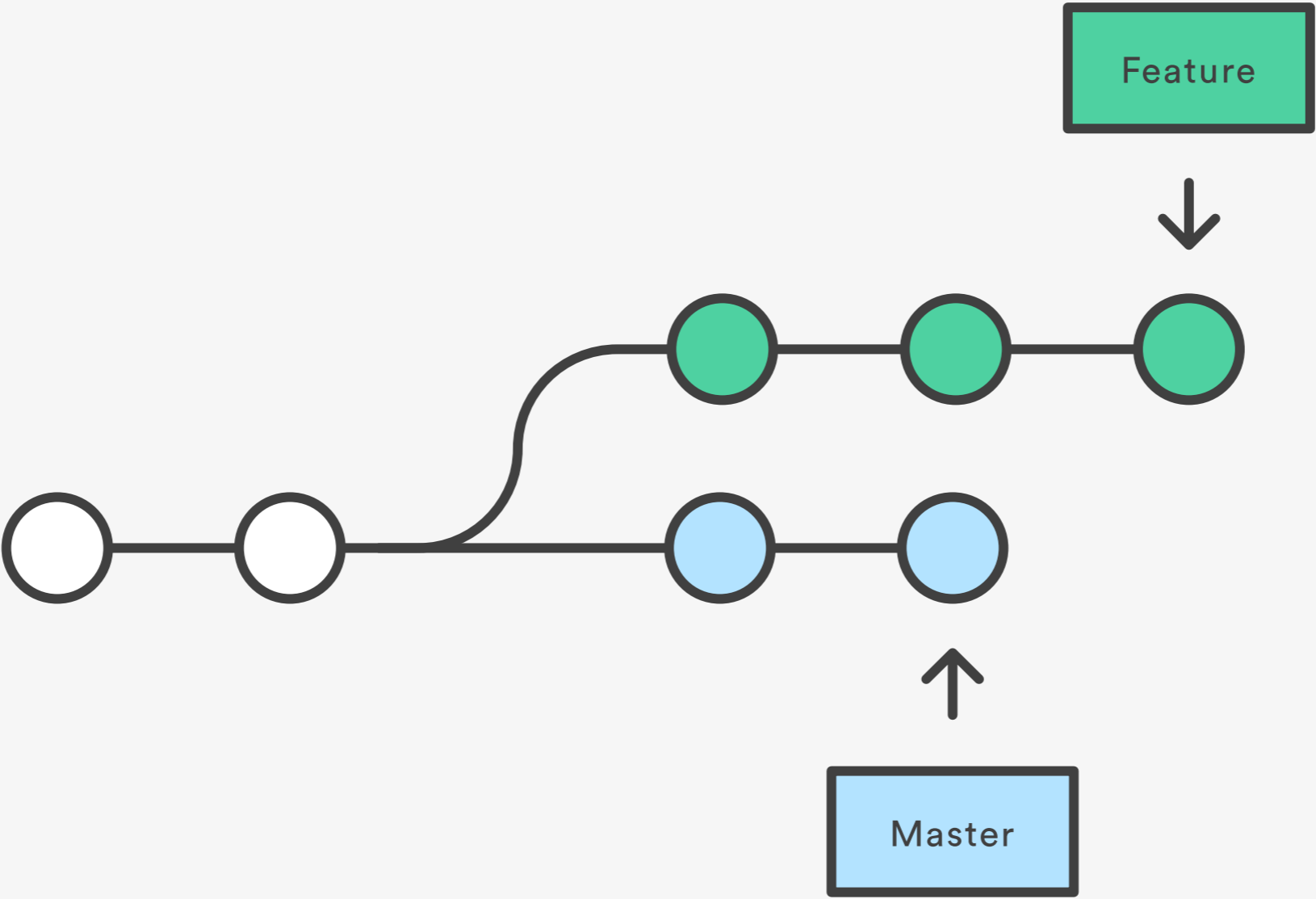
 Repositories	3
 Code	1,009,854
 Commits	5,625,124
 Issues	149,375

COMMAND LINE: MERGE BUBBLES

```
* 5c6f7fd + tests indicated by /schemas
* cc725db Merge branch 'master' of github.com
\
| * ff1787c + desc of some regexp coverage
| * | 652883f + validation tests for 1bnodeRefORRefMinLength
|/
* 5e5efaf ~ s/Some/One/ in test and file names
```

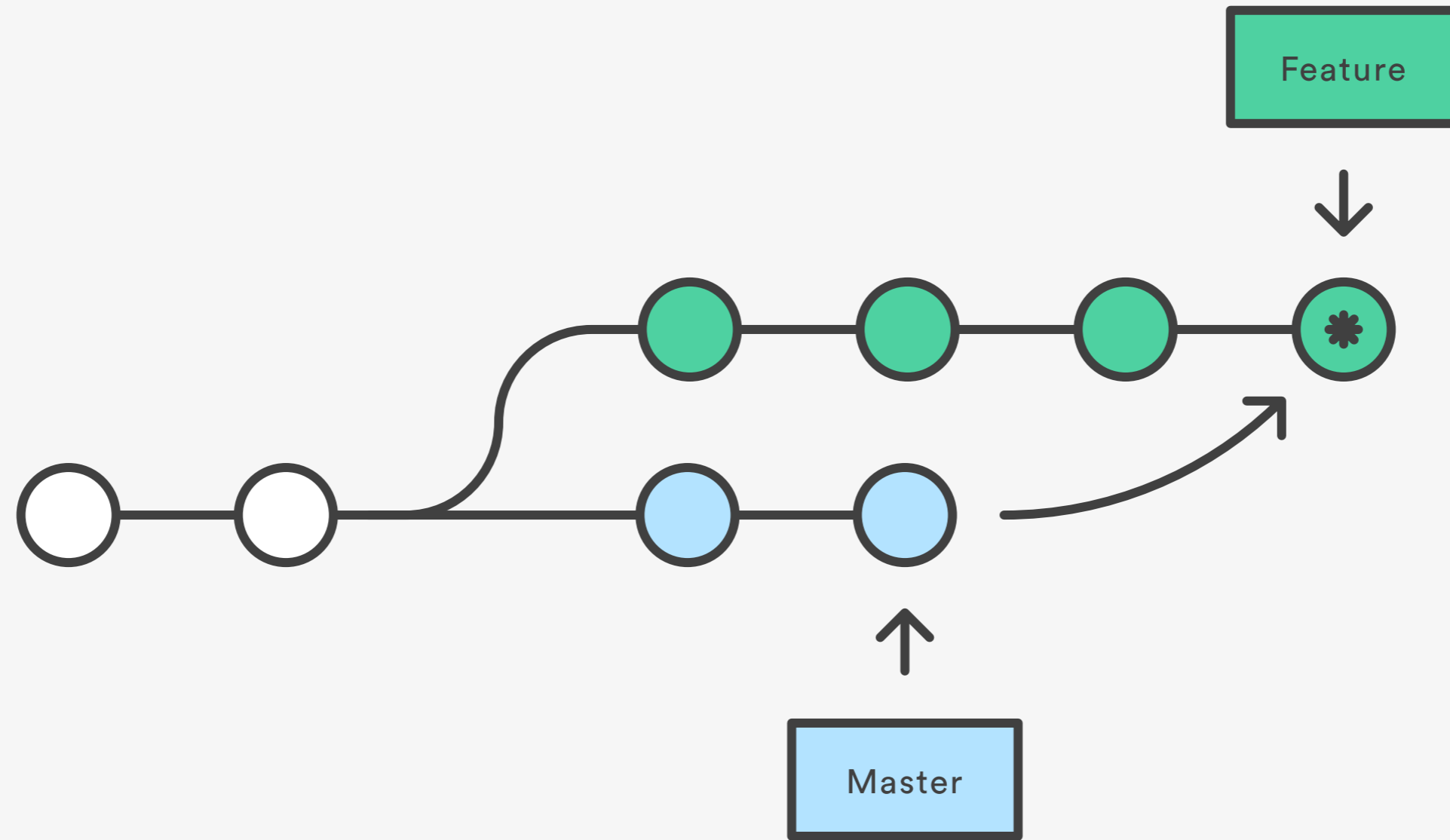
- Use rebase when pulling

A forked commit history



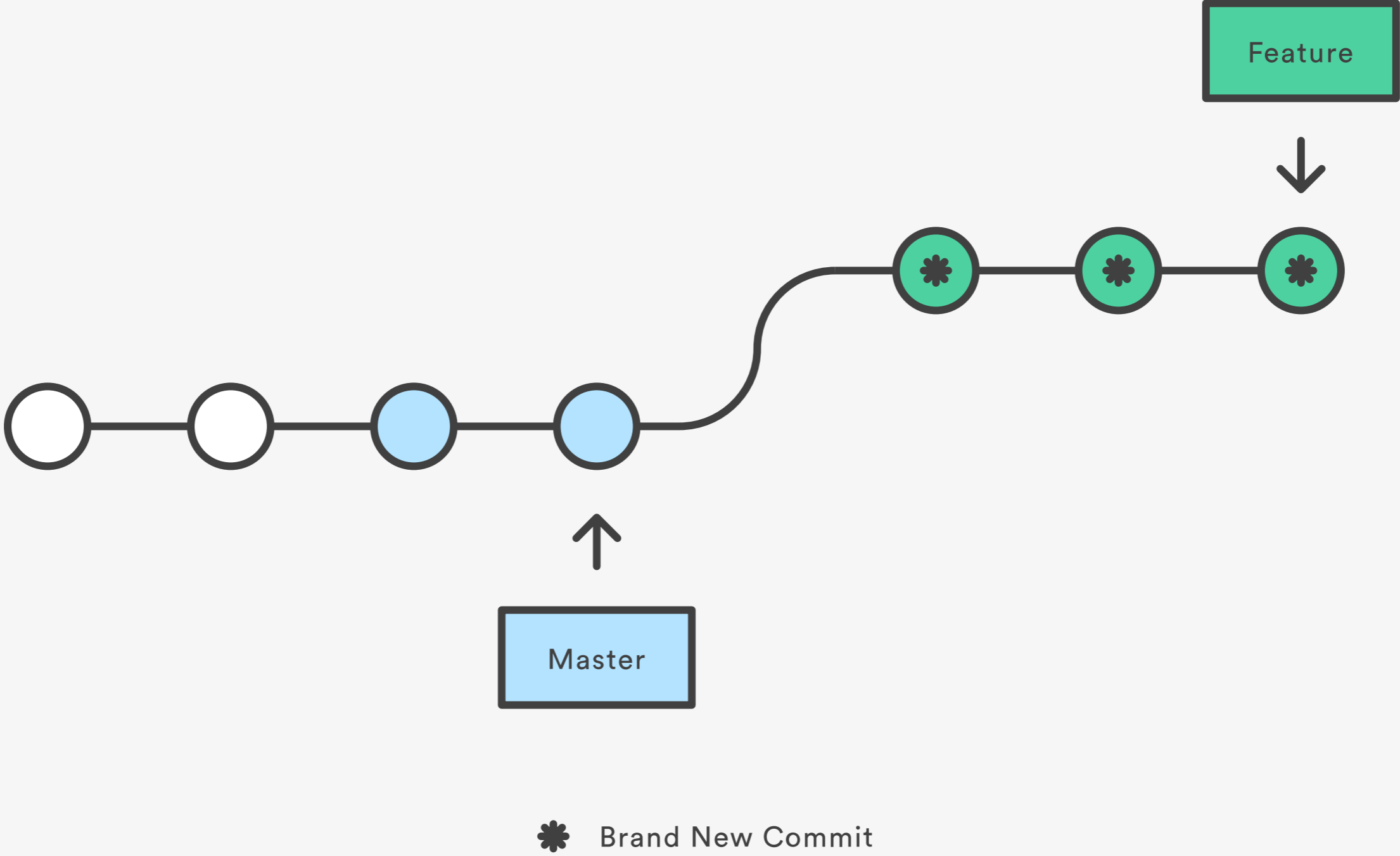
<https://wac-cdn.atlassian.com/dam/jcr:01b0b04e-64f3-4659-af21-c4d86bc7cb0b/01.svg?cdnVersion=lh>

Merging master into the feature branch



* Merge Commit

Rebasing the feature branch onto master



LEVEL 3



SUPERB COMMIT MESSAGES

- Answer some basic questions
 - Why is this change necessary?
 - How does it address the issue?
 - What side effects does this change have?

Add some information about properly formatted commit messages

[Browse files](#)

It does seem like a lot of github users are not used to good commit message rules, and may never have used git for a project that actually cares about good logs and nice summary lines.

Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>

 master  v4.5.96  v1.0



torvalds committed on 6 Sep 2011

1 parent [454a456](#)

commit [b6590150d68df528efd40c889ba6eea476b39873](#)

 Showing **1 changed file** with **21 additions** and **0 deletions**.

Unified

Split

21  README

View



@@ -49,3 +49,24 @@ the right to pass it on as an open source patch.

49 49

50 50

See: <http://gerrit.googlecode.com/svn/documentation/2.0/user-signedoffby.html>

51 51

```
52 +Also, please write good git commit messages. A good commit message
53 +looks like this:
54 +
55 +     header line: explaining the commit in one line
56 +
57 +     Body of commit message is a few lines of text, explaining things
58 +     in more detail, possibly giving some background about the issue
59 +     being fixed, etc etc.
60 +
61 +     The body of the commit message can be several paragraphs, and
62 +     please do proper word-wrap and keep columns shorter than about
63 +     74 characters or so. That way "git log" will show things
64 +     nicely even when it's indented.
65 +
66 +     Reported-by: whoever-reported-it
67 +     Signed-off-by: Your Name <youremail@yourhost.com>
68 +
69 +where that header line really should be meaningful, and really should be
70 +just one line. That header line is what is shown by tools like gitk and
71 +shortlog, and should summarize the change in one readable line of text,
72 +independently of the longer explanation.
```


SUPERB COMMIT MESSAGES

commit 0fdf810d267928b4c26a2e2cbb8c02b79e7b0ccb
Author: Wladimir J. van der Laan <laanwj@gmail.com>
Date: Fri Oct 28 14:15:46 2016 +0200

Change default confirm target from 2 to 6

Recent discussion (in IRC meetings, and e.g. #8989) has shown a preference for the default confirm target for smartfees to be 6 instead of 2, to avoid overpaying fees for questionable gain.

6 is also a compromise between the GUI's pre-#8989 value of 25 and the bitcoind `-txconfirmtarget`` default of 2. These were unified in #8989, but this has made the (overly expensive) default of 2 as GUI default.

SUPERB COMMIT MESSAGES

- Imperative
 - "Fix bug" vs "~~Fixed bug~~" or "~~Fixes bug~~"
 - like git merge and git revert
 - If applied, this commit will your subject line here
- Link to previous commits (like "git revert" does)

POWER COMMITS: HOOKS

- Automate stuff
- Closing issues via commit messages
- Prepend ticket number from branch to message
- Validate commit message format

ISSUE NUMBERS

Every line of code is always documented

— <http://mislav.net/2014/02/hidden-documentation/>

COMMAND LINE: WHO'S TO BLAME?

```
$ git blame README.md
202f612a41 (Pavel Janík 2015-10-05 21:20:43 1) Bitcoin Core integration/staging tree
c8af33aa75 (Wladimir J. 2013-12-17 11:50:26 2) =====
20d5945505 (Colin Dean 2012-12-12 21:27:58 3)
5a0823a09b (randy-waterh 2014-08-26 21:10:52 4) [![Build Status](https://travis-ci.org/bitcoin/bitcoin.svg
5a0823a09b (randy-waterh 2014-08-26 21:10:52 5)
b07b103e8a (BtcDrak 2016-01-15 07:45:39 6) https://bitcoincore.org
20d5945505 (Colin Dean 2012-12-12 21:27:58 7)
20d5945505 (Colin Dean 2012-12-12 21:27:58 8) What is Bitcoin?
20d5945505 (Colin Dean 2012-12-12 21:27:58 9) -----
...

```

git config alias.praise blame

SUMMARY

- Use a project/company/etc... wide commit format and document it!
 - Style Content Metadata
- Ditch Git GUIs and start using the command line
- Read your code (see next)
- Split up commits (atomic commits)
- Don't do "end-of-day" or "code drop" commits
- Check your impact on the commit graph
 - commit merge rebase squash
- Rewrite history if necessary

THANK YOU

slido.com #devops2019

jachim.coudenys@combellgroup.com
@coudenysj

