Storytelling with Git

by Avi Flax
electronic mail: avi@aviflax.com
slides: `https://git.io/Je2Ew`

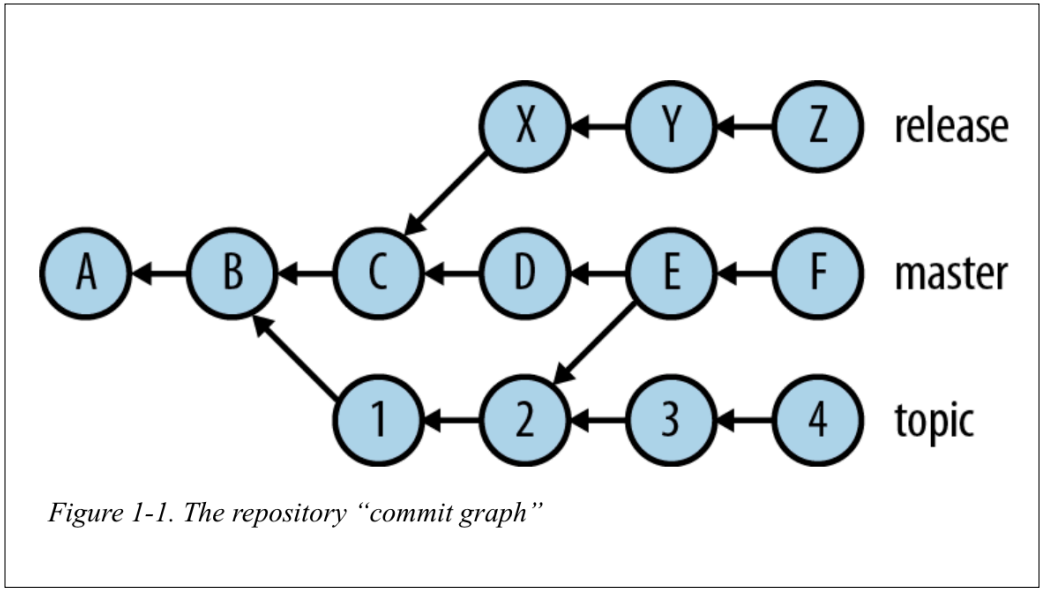
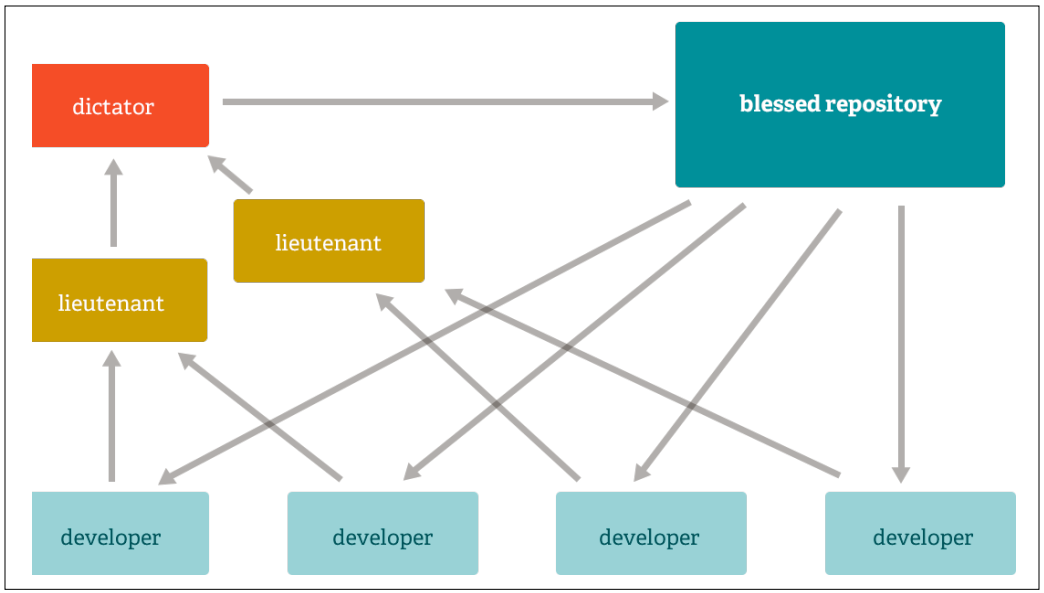


Figure 1-1. The repository "commit graph"

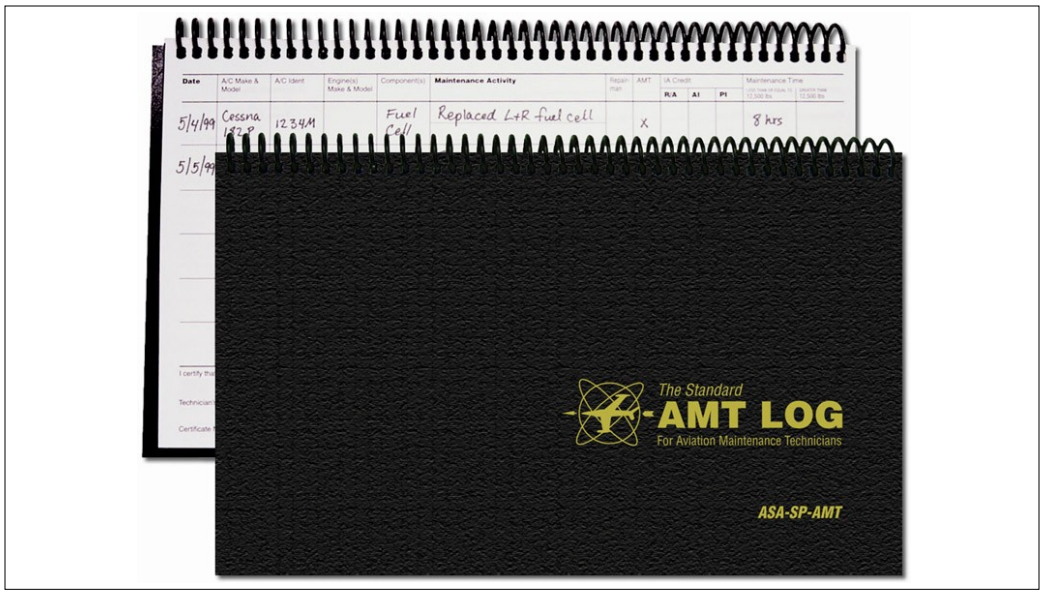
To lots of people, this is the essence of Git: a graph of changes.

Source: https://www.oreilly.com/library/view/git-pocket-guide/9781449327507/ch01.html



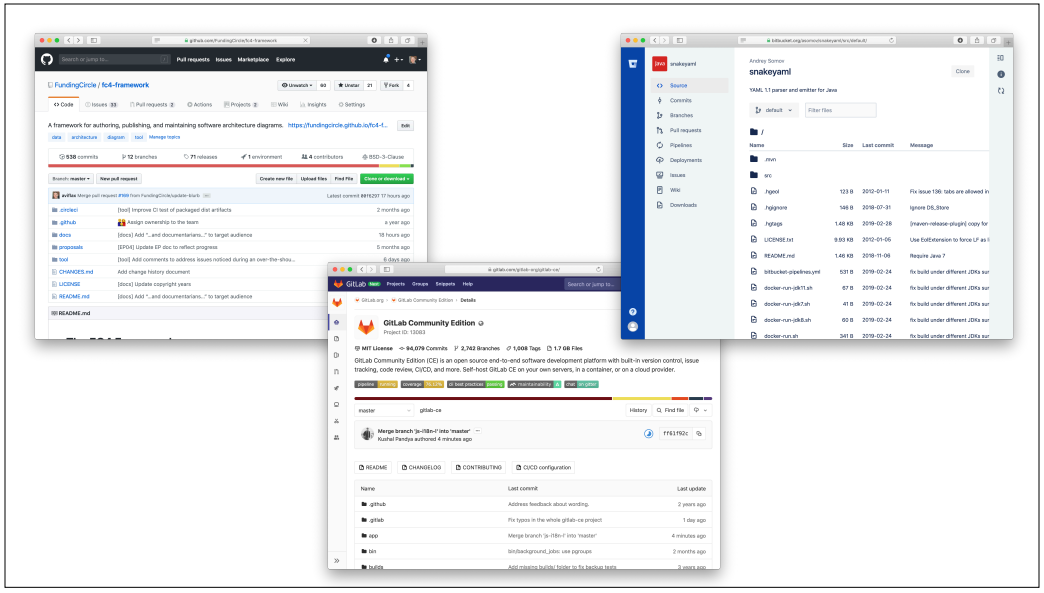
Some focus on the *distributed* aspect of git.

Source: https://git-scm.com/about/distributed

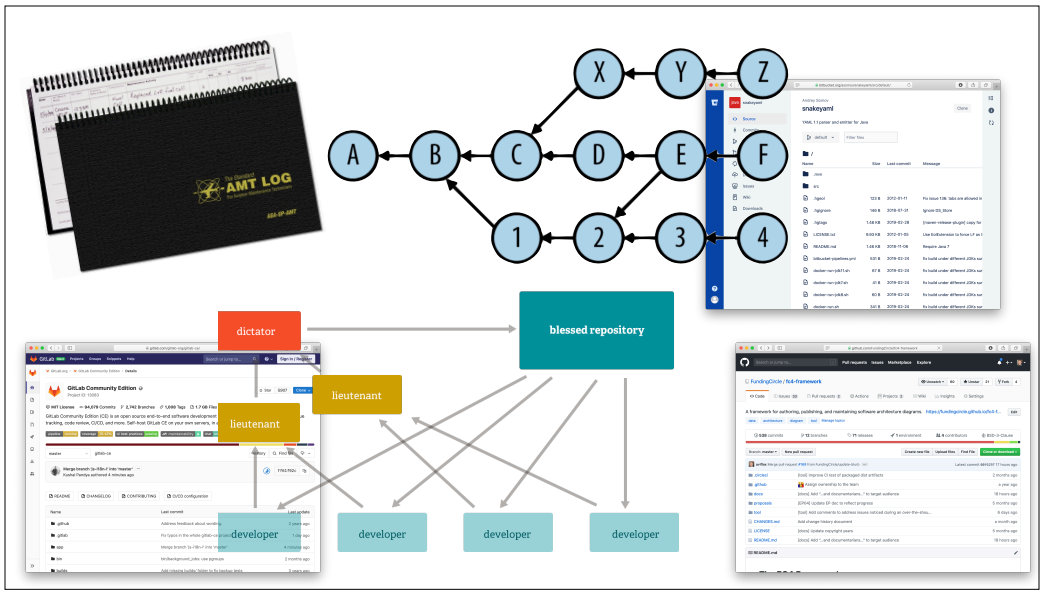


Many view it as a matter-of-fact log of work performed, like this aircraft maintenance log.

Source: http://www.chiefaircraft.com/asa-sp-amt.html



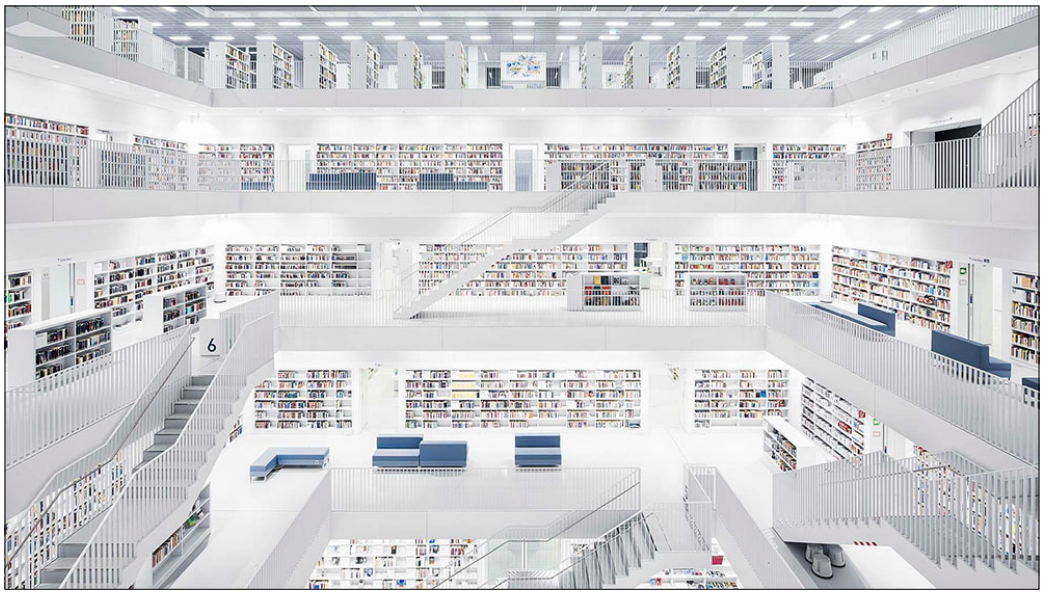
Git is also used as the foundation of some fantastic Web-based collaboration platforms.



Those are all valid and important aspects of git.

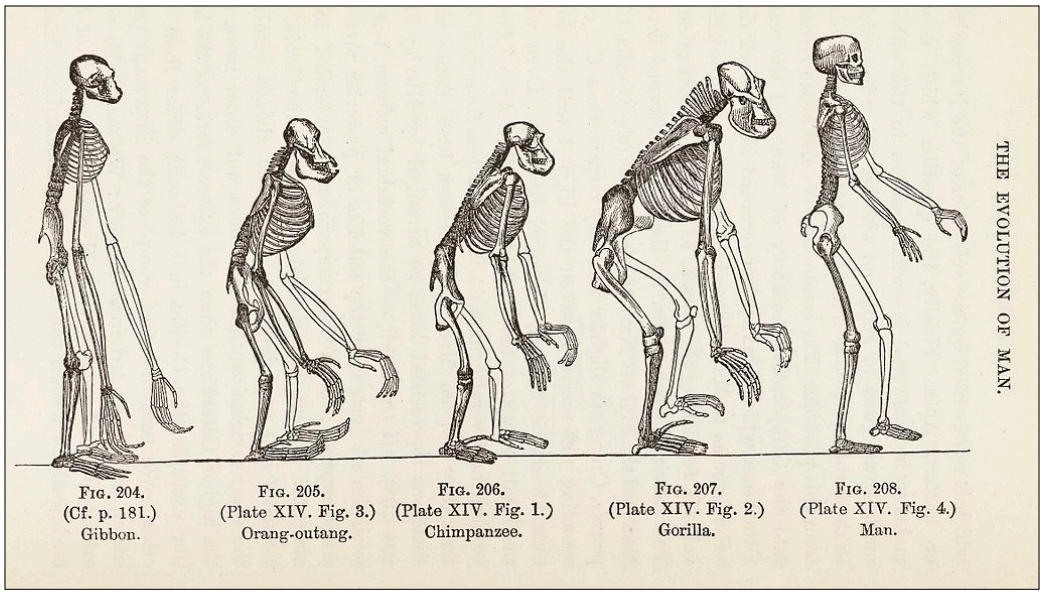
Fundamental, really.

But there's another



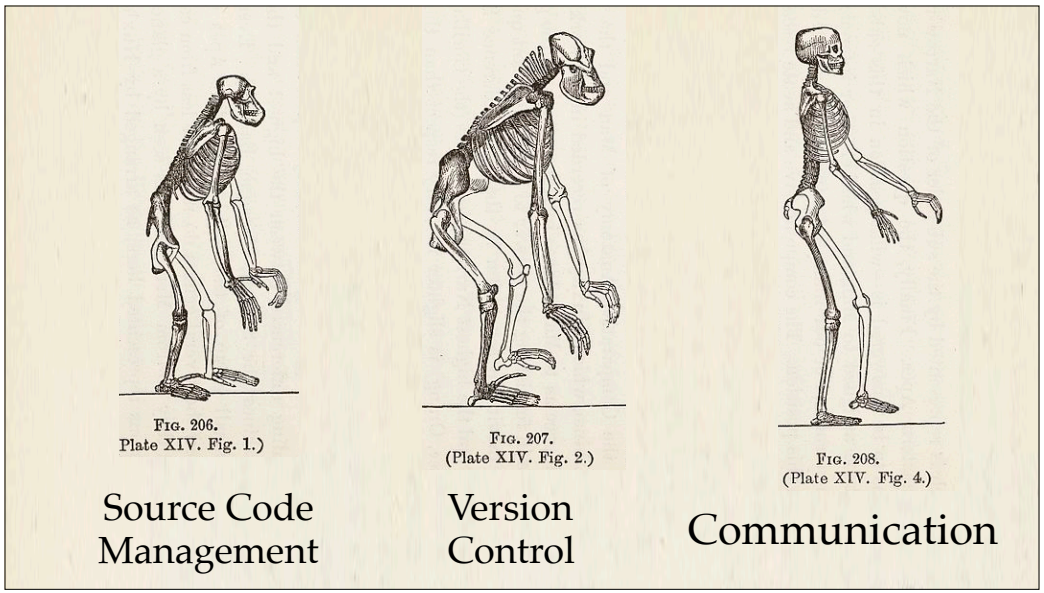
- When used well, a git repository can become a rich trove of history and knowledge about the system, platform, corpus of documentation, or even organization that it hosts.
- Git can be to your organization as a library is to a civilization:
- A collection of stories — thousands of them,
- which we go to when we want to dig deep into the history of our world
- to understand
- not only *how* our particular context came about
- but *why.*
- But in order to do so… →

Photo: Stadtbibliothek, Stuttgart, 2011, by Thibaud Poirier, via Colossal: https://www.thisiscolossal.com/2017/06/enchanting-libraries-by-photographer-thibaud-poirier/



We need to evolve our view of what git **is**.

Source: https://commons.wikimedia.org/wiki/File:Illustration;_%27The_Evolution_of_Man%27_Wellcome_L0063036.jpg



- From *SCM* tool to *Communication* tool.
- Git can be a powerful communication tool.
- We need only to stand up straight and look again, see it that way.
- Probably the most effective form of communication is…

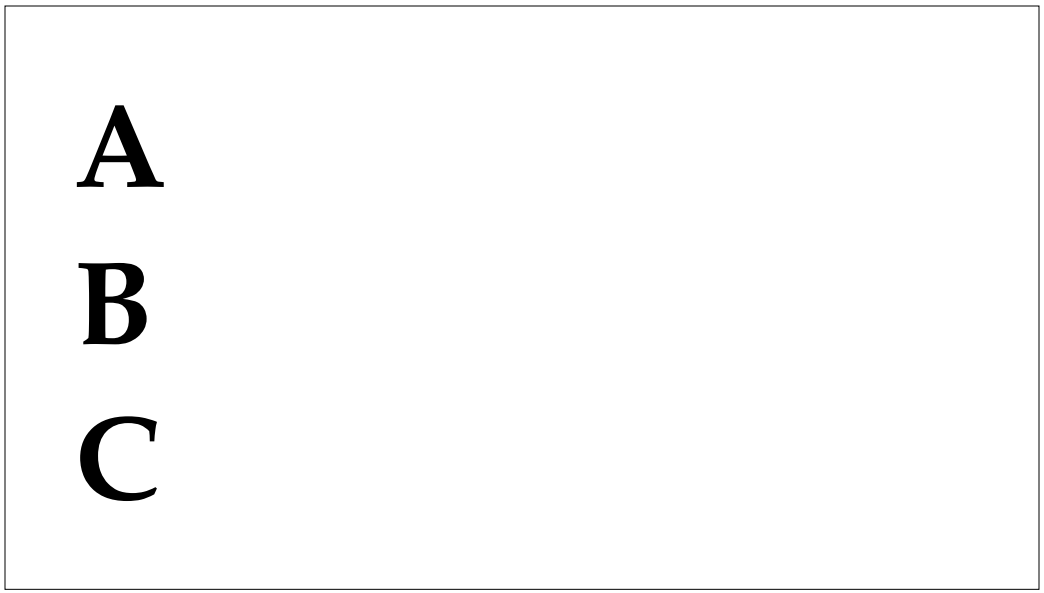
Source: https://commons.wikimedia.org/wiki/File:Illustration;_%27The_Evolution_of_Man%27_Wellcome_L0063036.jpg

- **…storytelling.**
- We can and should tell stories with git
- May sound bizarre — let me show you how →

Source: https://www.tate.org.uk/art/artworks/millais-the-boyhood-of-raleigh-n01691 (CC-BY-NC-ND (3.0 Unported))

1. Log each step
2. Craft your commits
3. Share the story

Log Each Step


A
B
C

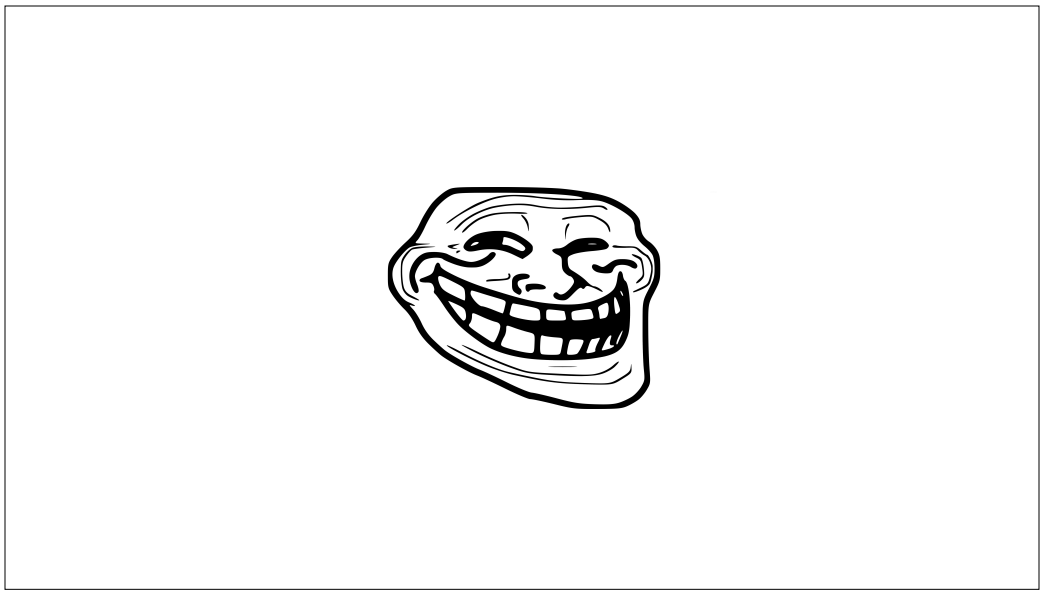
Always

B

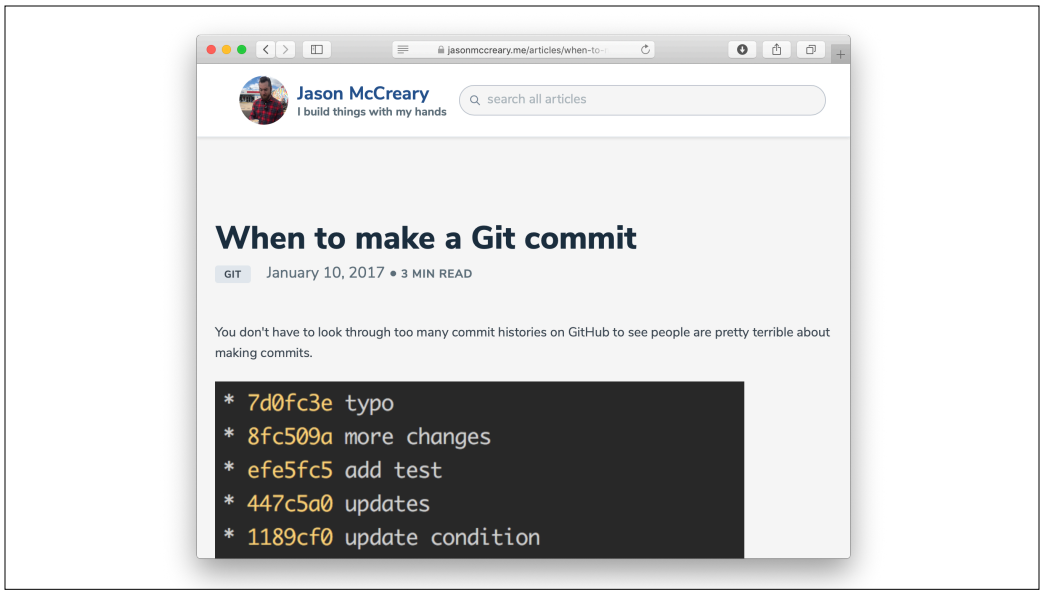
C

Always
Be
C

Always
Be
Committing



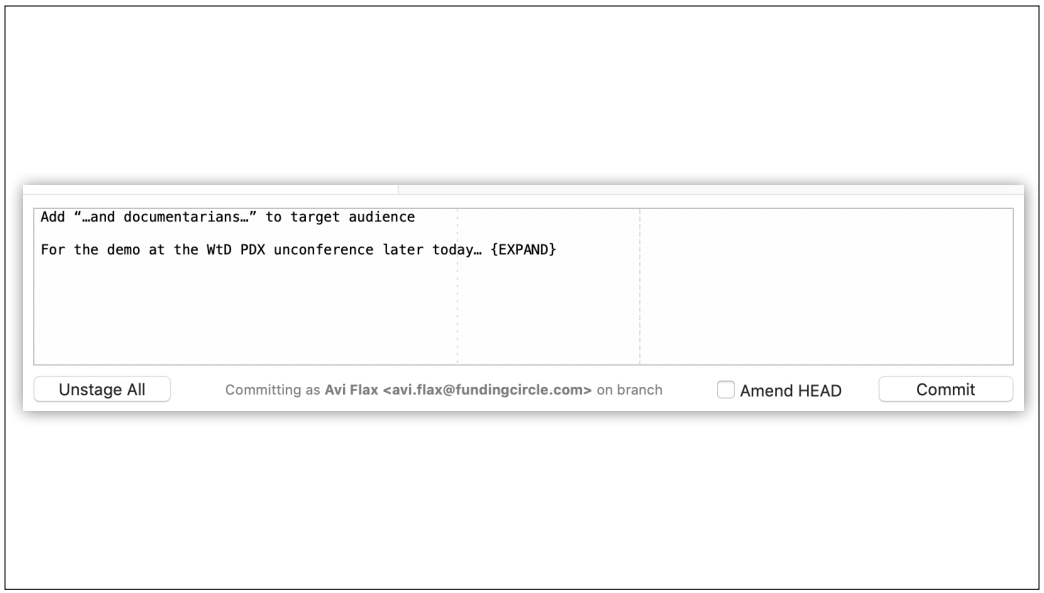
- Except not really.
- You can’t literally always be committing,
- so while you’re working,
- follow the advice of Jason McCreary →



- Who says to make a commit when you:
  - Complete a unit of work
  - Are making changes you may want to undo
- https://jasonmccreary.me/articles/when-to-make-git-commit//

- The point is: you start in logging mode,
- not storytelling mode.
- Write what you're doing and why, but concisely —
    - these are notes just for your own eyes at this stage.
- Or maybe even better, think of this as your rough draft —
    - you want to keep it *truly* rough at this stage.
- However,
    - Even at this stage,
    - make sure to use the standard structure for a git commit message

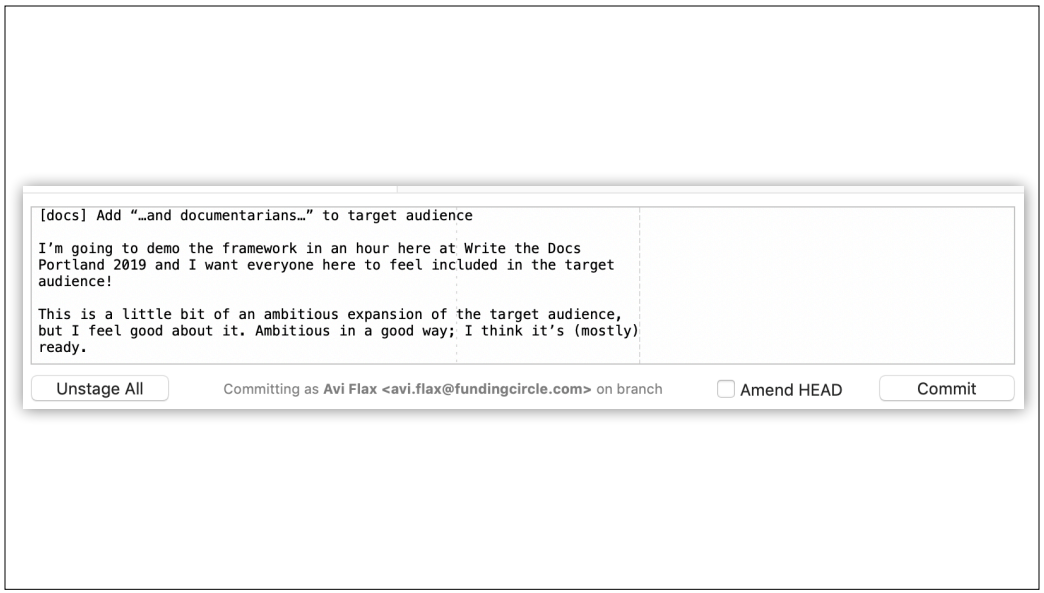
Source: https://commons.wikimedia.org/wiki/File:Logs.jpg



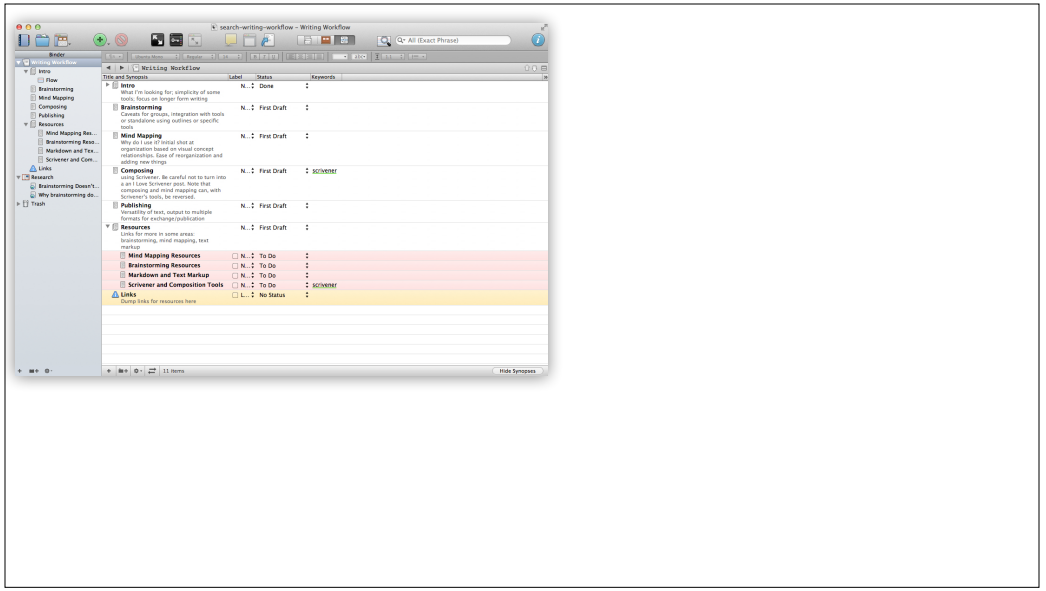
- This is a commit that I made while in this mode —
    - you can see that I have a note to myself in there to expand on the back story, but I planned to do it later.
- First line: **subject** or **title** or **summary**
- Blank line
- The rest: body
    - Go nuts
- I didn’t make this up!
    - It’s in the the Git manual.
- It’s really powerful and many or most git tools take advantage of this structure
    - So use it!

Craft Your Commits

- Next: craft your commits.
- This is when you take your rough draft and fix it up into something coherent and informative.

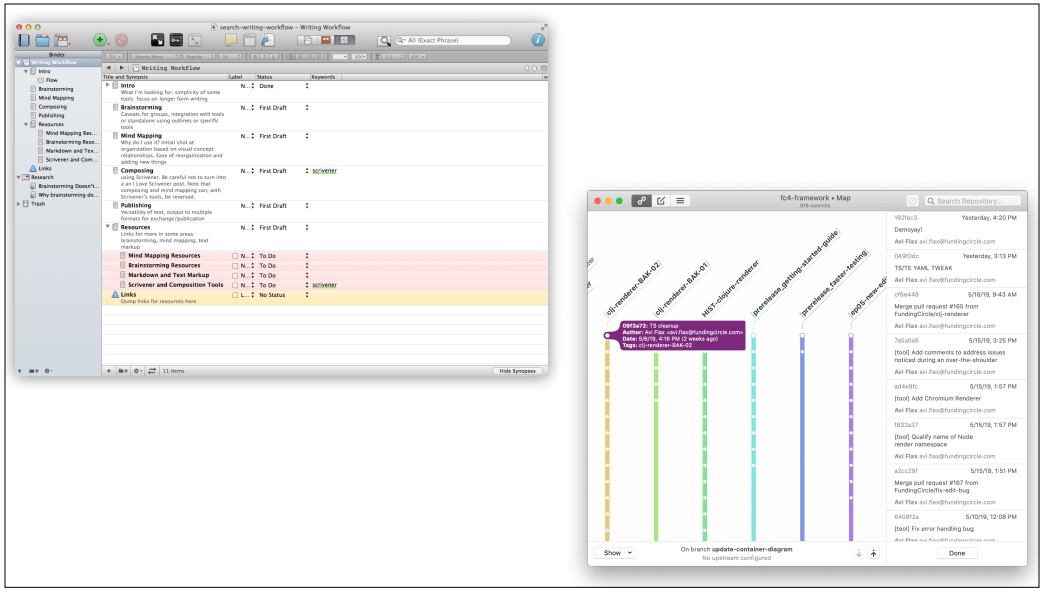


- The most fundamental kind of change
    - you’d make during this stage:
    - to revise your commit messages
- This is the same commit as before, now containing the full backstory.
- Remember: the subject/title/summary plus the diff captures the *what* —
    - the body of your message is where you capture the *why*

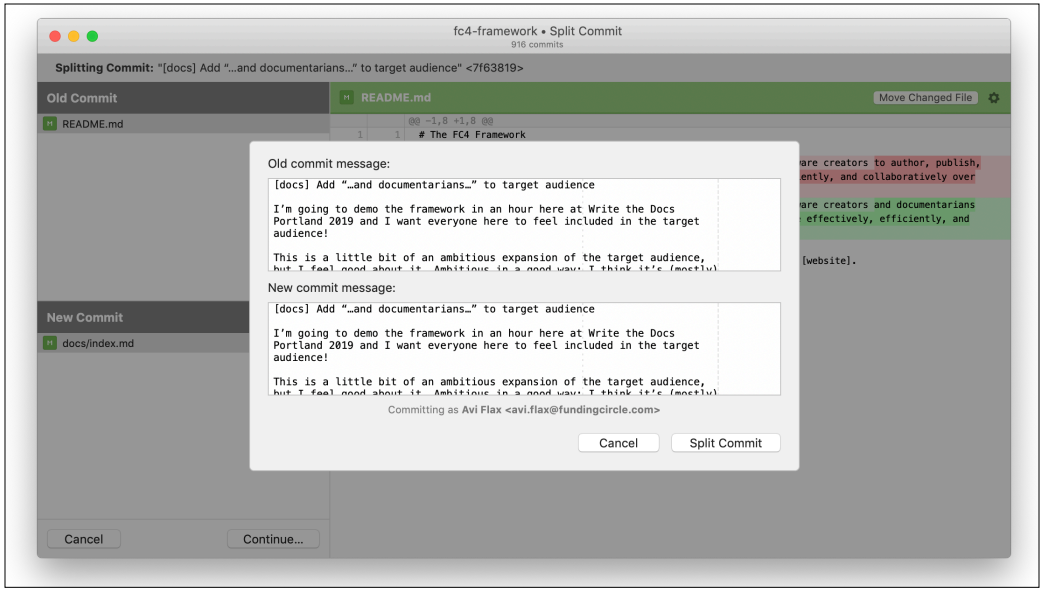


- This is a screenshot of a writing tool called Scrivener
- I don’t use it, but it seemed like a good example of using a writing tool to craft a story — to rearrange sentences and paragraphs, merge them together, split them apart, etc.

Source: https://www.flickr.com/photos/fncll/8576415696



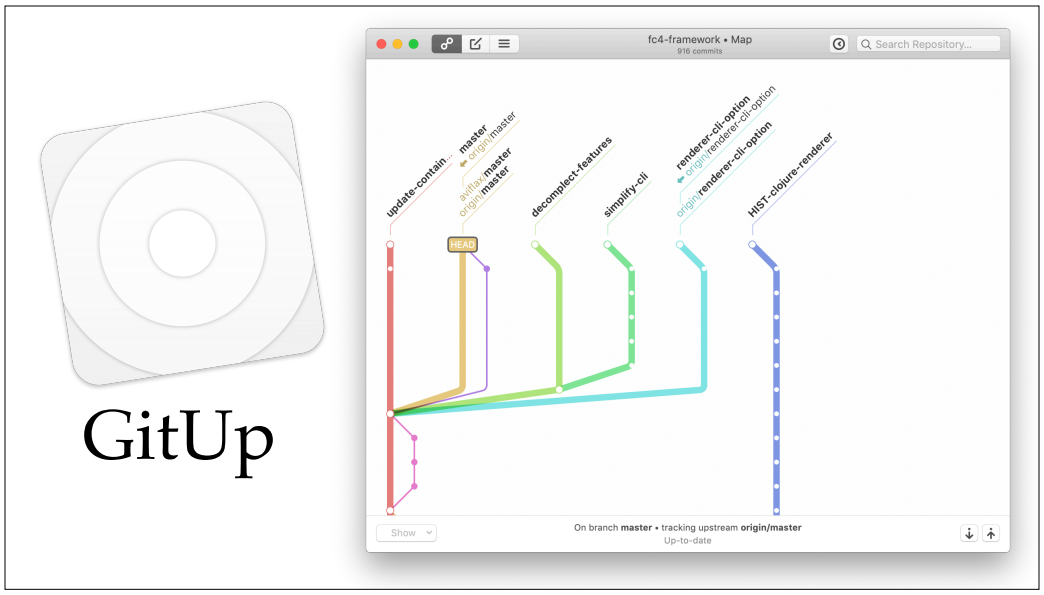
- This is a screenshot of a Git interface called GitUp
- I use it to do the same thing, but with Git commits.
- When I started using GitUp a few years ago, it was like gaining a superpower —
    - It just made it super easy for me to edit my commits — to merge them together, split them apart, reorder them, etc



For example: {describe screenshot}

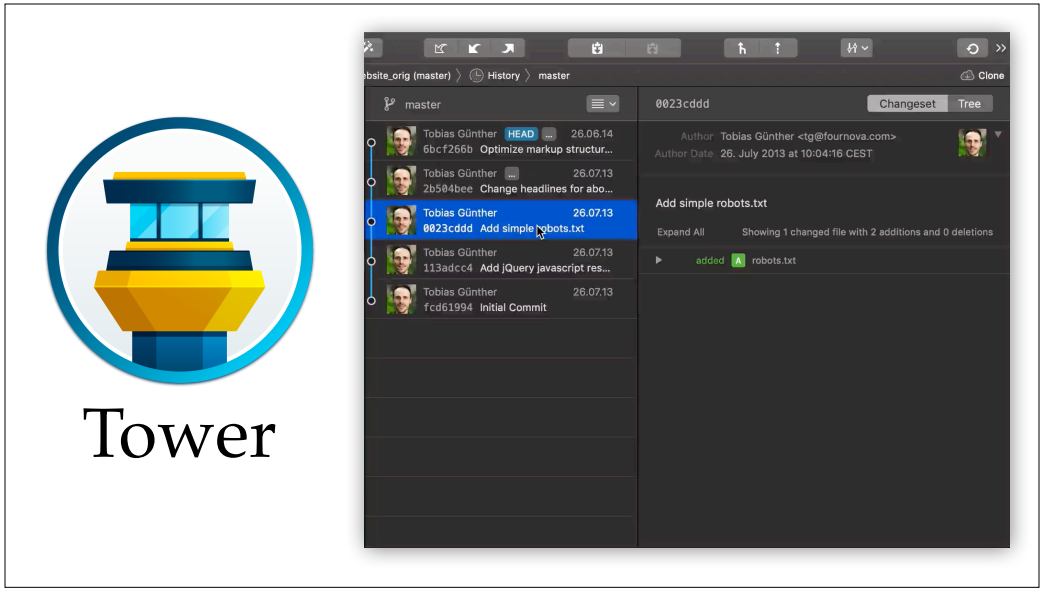
Tools Matter

- As I mentioned above, my *Commit Crafting* skills made a huge leap when I started using a new Git GUI. You (and your peers) might have a similar experience.
- I'll suggest two tools that I have personal experience with, but this is by no means exhaustive — if these suggestions aren't a good fit for you, I'm sure you can find other solid options quickly and easily.



GitUp is Mac-only, but if you (or your peers) *are* using Macs, I recommend it wholeheartedly.

https://gitup.co/



Tower is available for Mac and Windows. I've used it sporadically over the years and always been impressed by it.

https://www.git-tower.com/

Share the Story

- Next: share the story

~~Push~~

Publish

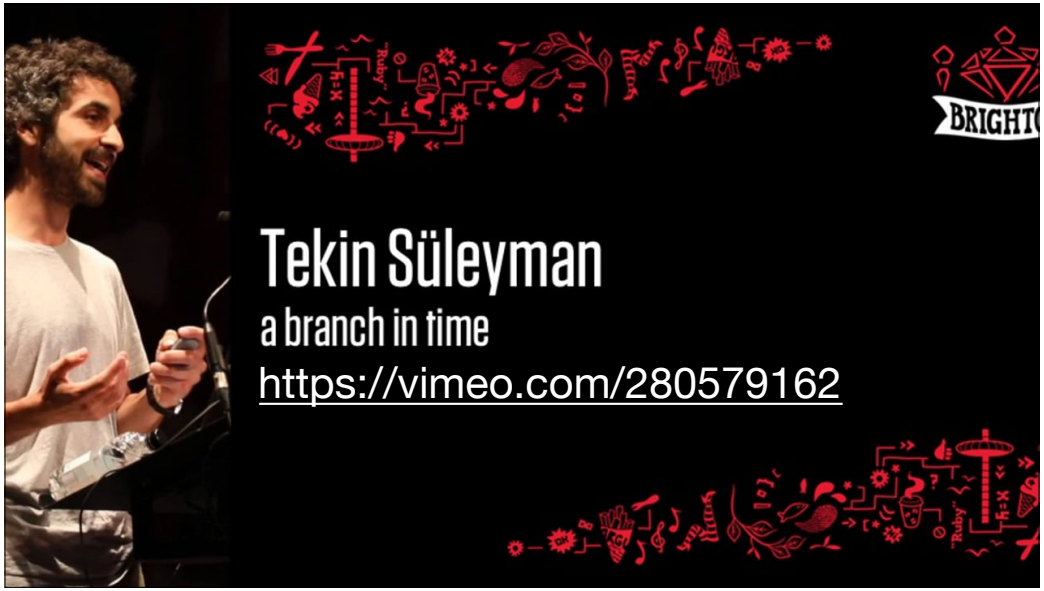
- Yes, technically, when you share your commits with your colleagues, you're using git's **push** feature to do so
- But don't just think of it as push,
    - think of it as **publishing.**
- As you publish your work,
    - you're also publishing the story of your work —
    - the **backstory**.

1. Log each step
2. Craft your commits
3. Share the story

- So remember:
    - {read slide}
- I hope you'll give this a try whenever you're working in a git repo,
    - and I hope you'll encourage your fellow software creators and documentarians to try this as well.
- From personal experience I can tell you:
    - it pays off, many times over.

Dive Deeper

Here are some pointers to learn more →



- *A Branch in Time* by Tekin Süleyman
- A wonderful (and short) talk: https://vimeo.com/280579162

https://chris.beams.io/posts/git-commit/

- *How to Write a Git Commit Message* by Chris Beams
- Excellent writeup on how and why to write a good commit message
- https://chris.beams.io/posts/git-commit/


🙏
Feedback, discussion, link to slides:
WtD Slack:
#docs-as-code
Electronic mail:
avi@aviflax.com
WWW:
aviflax.com

Thank you!