Version control with Git

Why using version control?

- **Backup**
- **Easy experimentation** with parallel branches
- **Collaboration** across teams
    - e.g. All of Google's code lies in one repository
- Formal **software development process**
    - Tagging of releases
    - Continuous integration
    - Pull requests, code reviews

History of (non-proprietary) version control systems

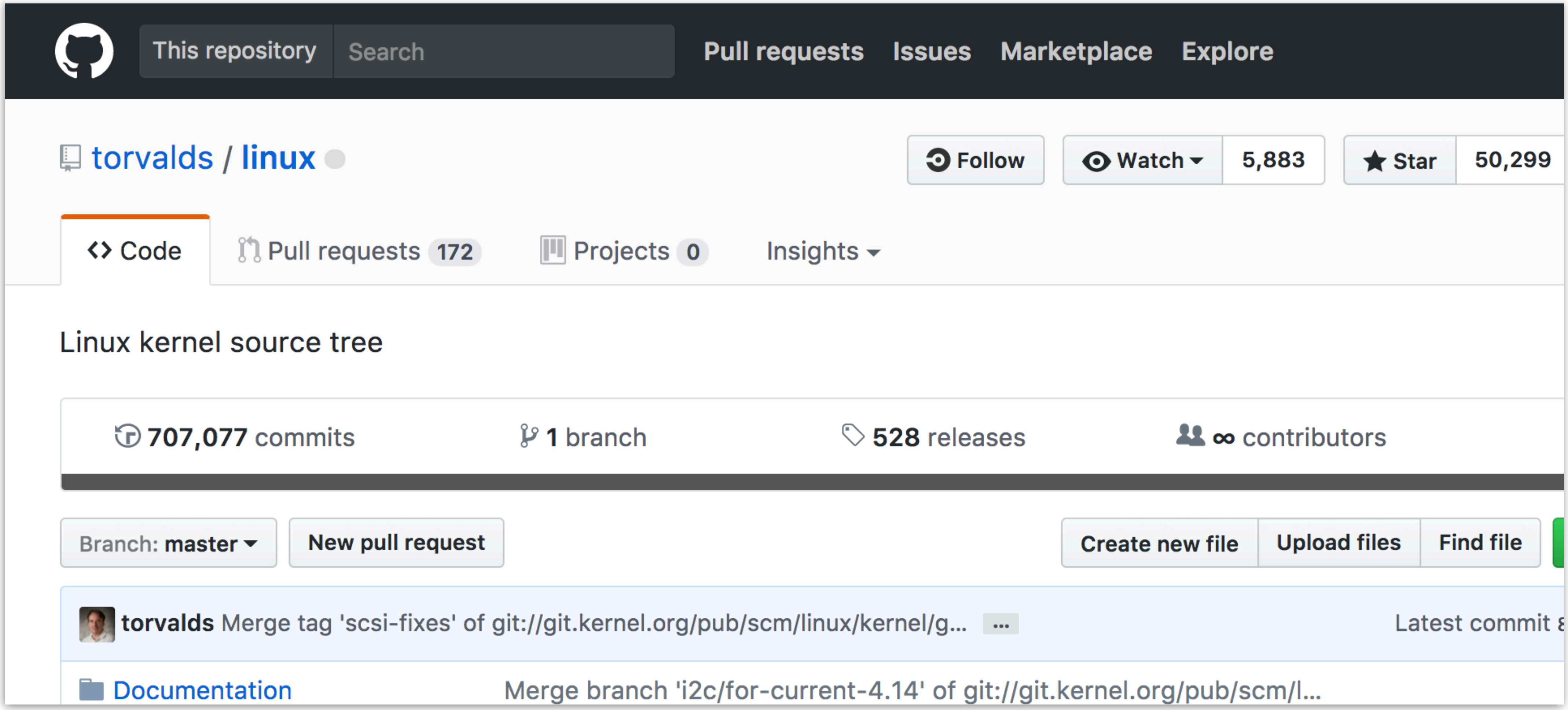
- 1970s: Print everything out.

History of (non-proprietary) version control systems

- 1990s: **CVS**
  - Independent history for each single file.
- 2000: **Subversion**
  - Client-server architecture.
  - Repository snapshots, introductions of branches.
- 2005: **Distributed Version Control (Git, Mercurial, Darcs, Bazaar, ...)**
  - Peer-to-peer network of replicated copies.
- 2010s: **Services on top (Github, Gitlab, ...)**
  - Issue tracking, pull requests, continuous integration, ...

Git

- Created by Linus Torvalds (1969-, Finnish), author of Linux
- Created for Linux kernel development



This repository Search Pull requests Issues Marketplace Explore

torvalds / linux

Follow Watch ▾ 5,883 Star 50,299

Code Pull requests 172 Projects 0 Insights ▾

Linux kernel source tree

707,077 commits 1 branch 528 releases ∞ contributors

Branch: master ▾ New pull request Create new file Upload files Find file

torvalds Merge tag 'scsi-fixes' of git://git.kernel.org/pub/scm/linux/kernel/g... ... Latest commit

Documentation Merge branch 'i2c/for-current-4.14' of git://git.kernel.org/pub/scm/l...

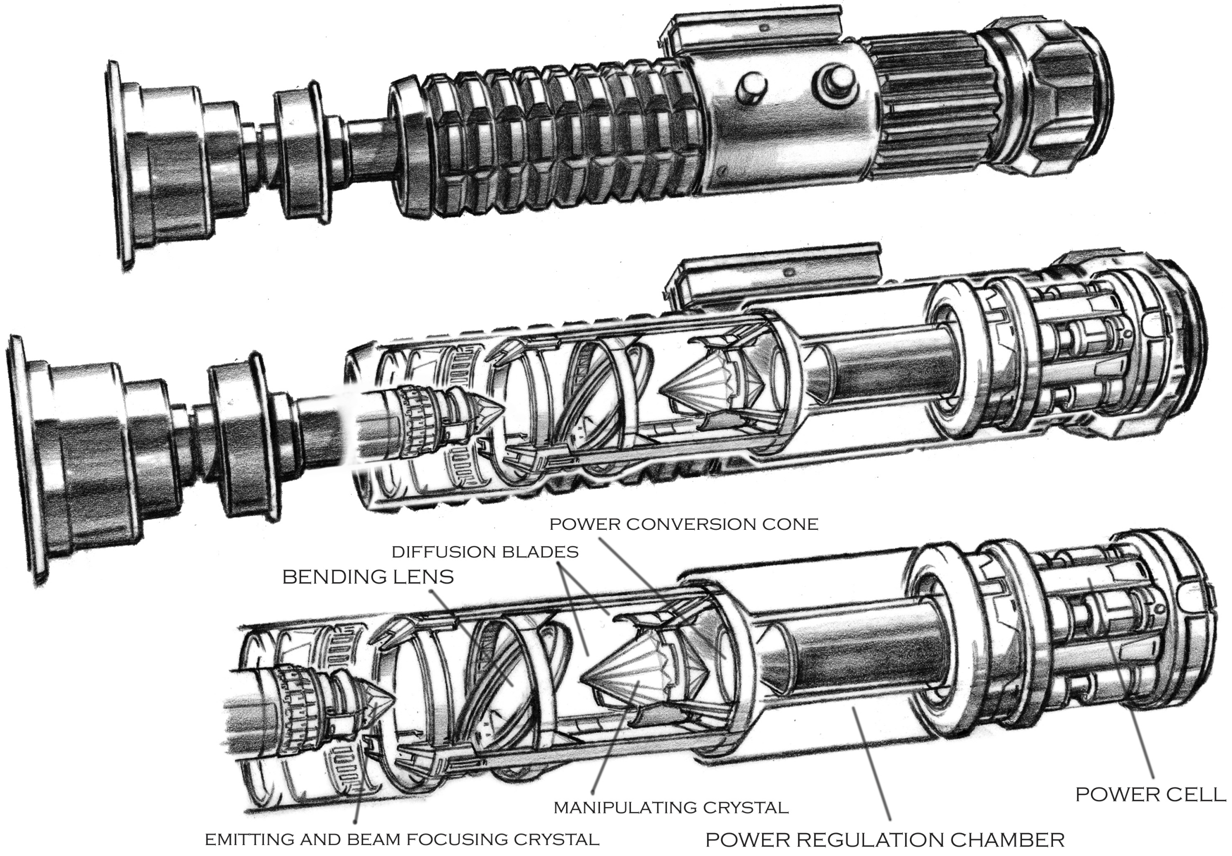
Git

- Created by Linus Torvalds (1969-, Finnish), author of Linux
- Created for Linux kernel development

When asked why [Linus Torvalds] called the new software, "git," British slang meaning "a rotten person," he said: “I'm an egotistical bastard, so I name all my projects after myself. First Linux, now git."

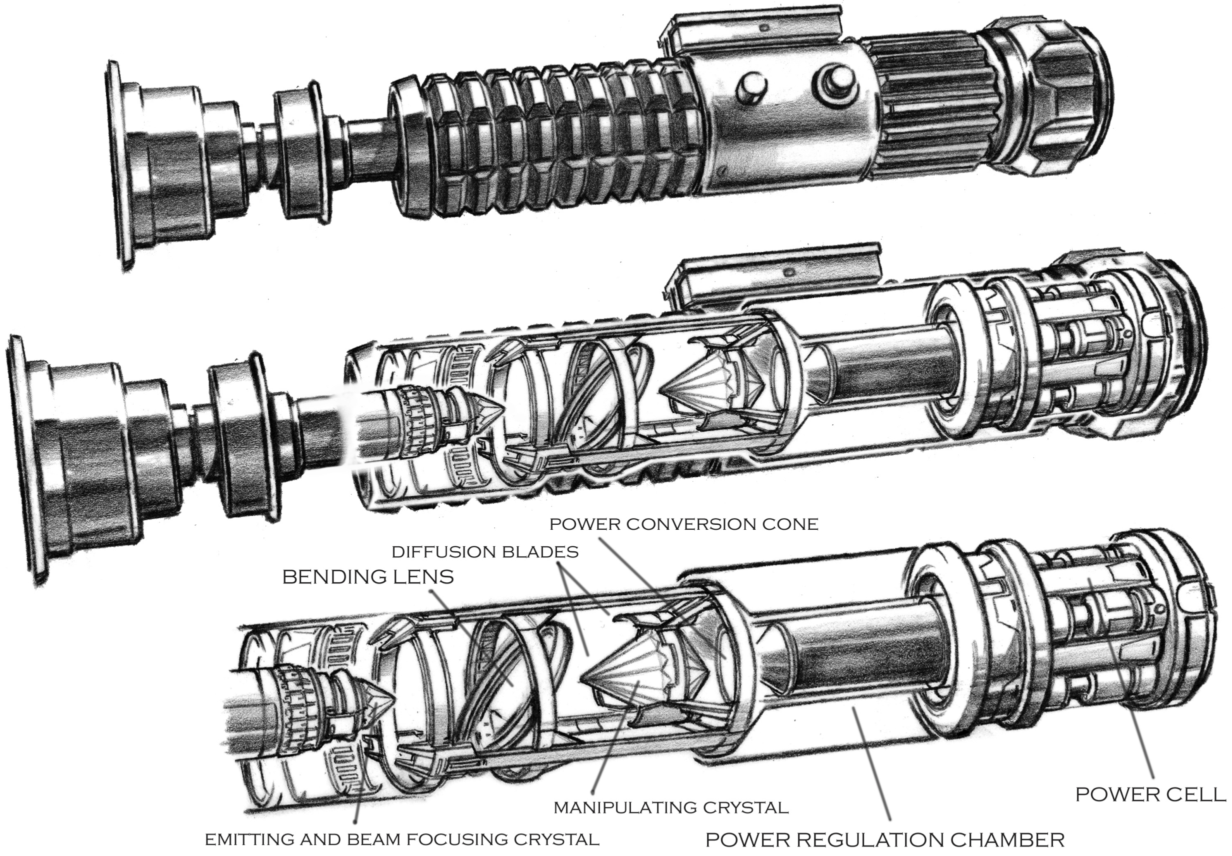
source

Jedi knights build their own lightsabers

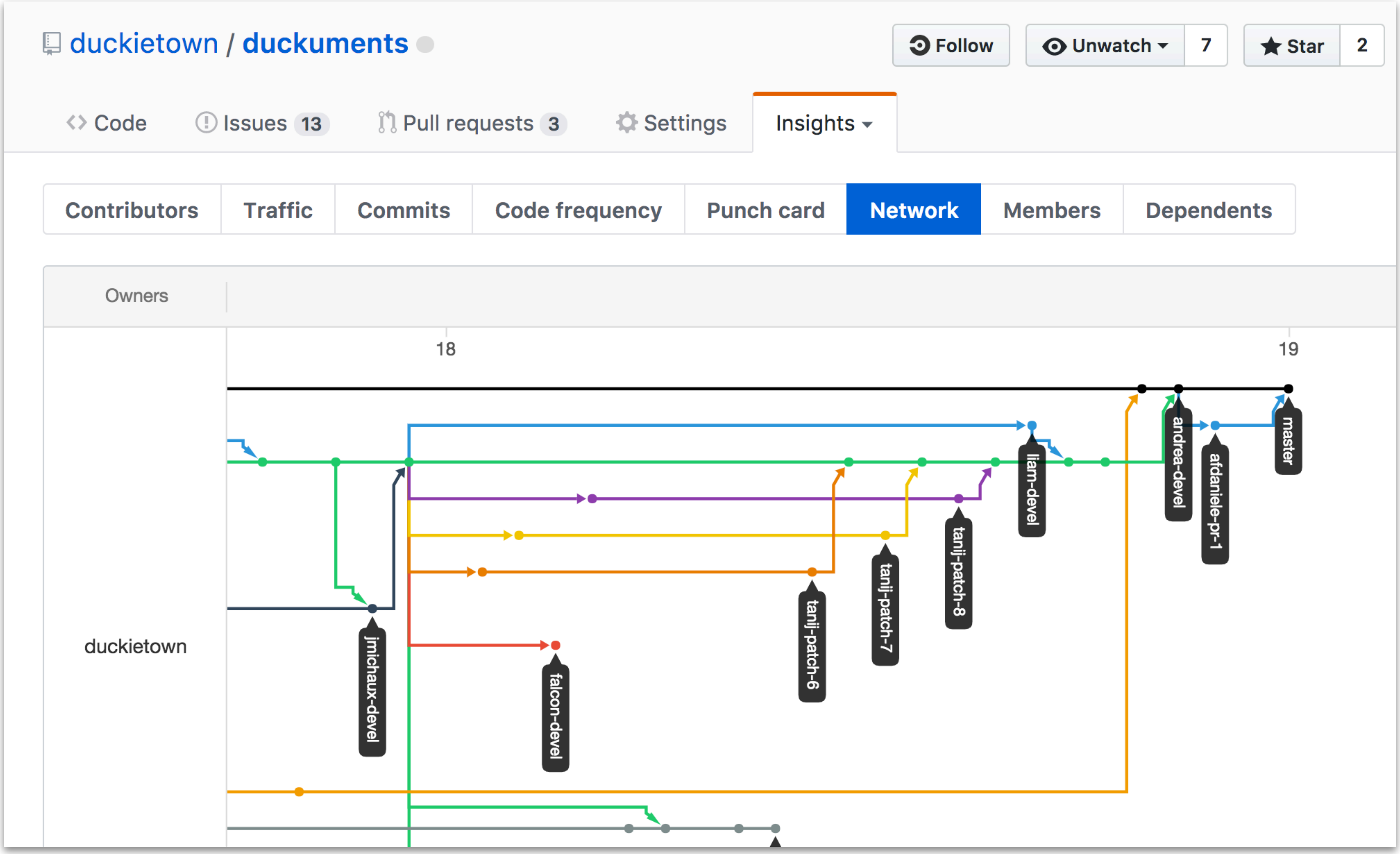


A roboticist's output is typically limited by tools

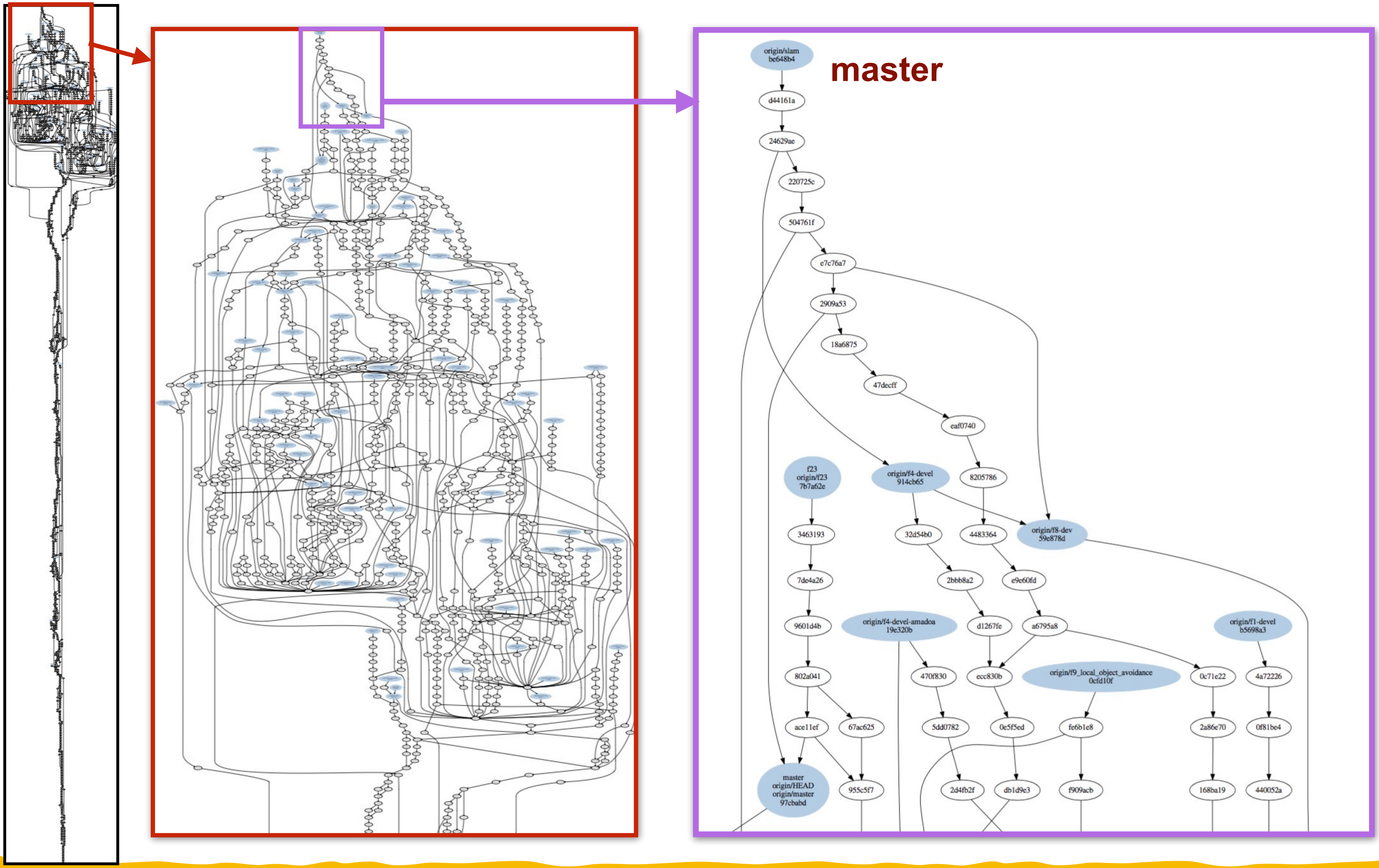
- Learn to use the tools that you have.
- Modify the tools to fit your needs.
- Create new tools.
  - Learn enough to be able to create the tools that you need.

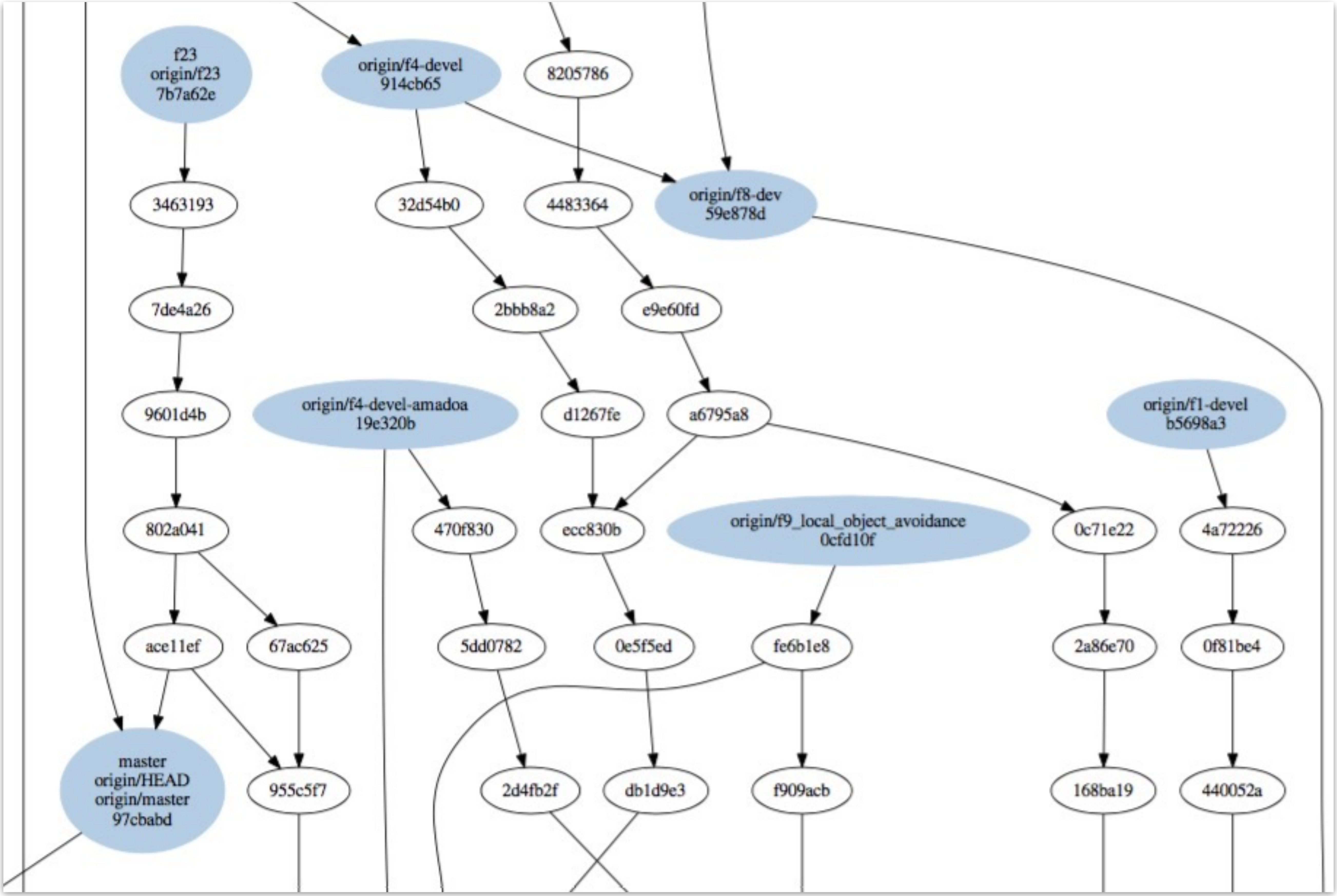


Commit graph



Commit graph




f23
origin/f23
7b7a62e
origin/f4-devel
914cb65
8205786
3463193
32d54b0
4483364
origin/f8-dev
59e878d
7de4a26
2bbb8a2
e9e60fd
9601d4b
origin/f4-devel-amadoa
19e320b
d1267fe
a6795a8
origin/f1-devel
b5698a3
802a041
470f830
ecc830b
origin/f9_local_object_avoidance
0cfd10f
0c71e22
4a72226
ace11ef
67ac625
5dd0782
0e5f5ed
fe6b1e8
2a86e70
0f81be4
master
origin/HEAD
origin/master
97cbabd
955c5f7
2d4fb2f
db1d9e3
f909acb
168ba19
440052a

Git explained with dynamical systems

Dynamical system

state	$s \in S$
command	$u \in U$
transition function	$f : S \times U \to S$

Version control / “patch theory”

$S \doteq$ repository states

$U \doteq$ editing actions

Patches

$$\mathrm{diff} : S \times S \to U$$

$$f(s_1, \mathrm{diff}(s_2, s_1)) = s_2$$

Commit graph

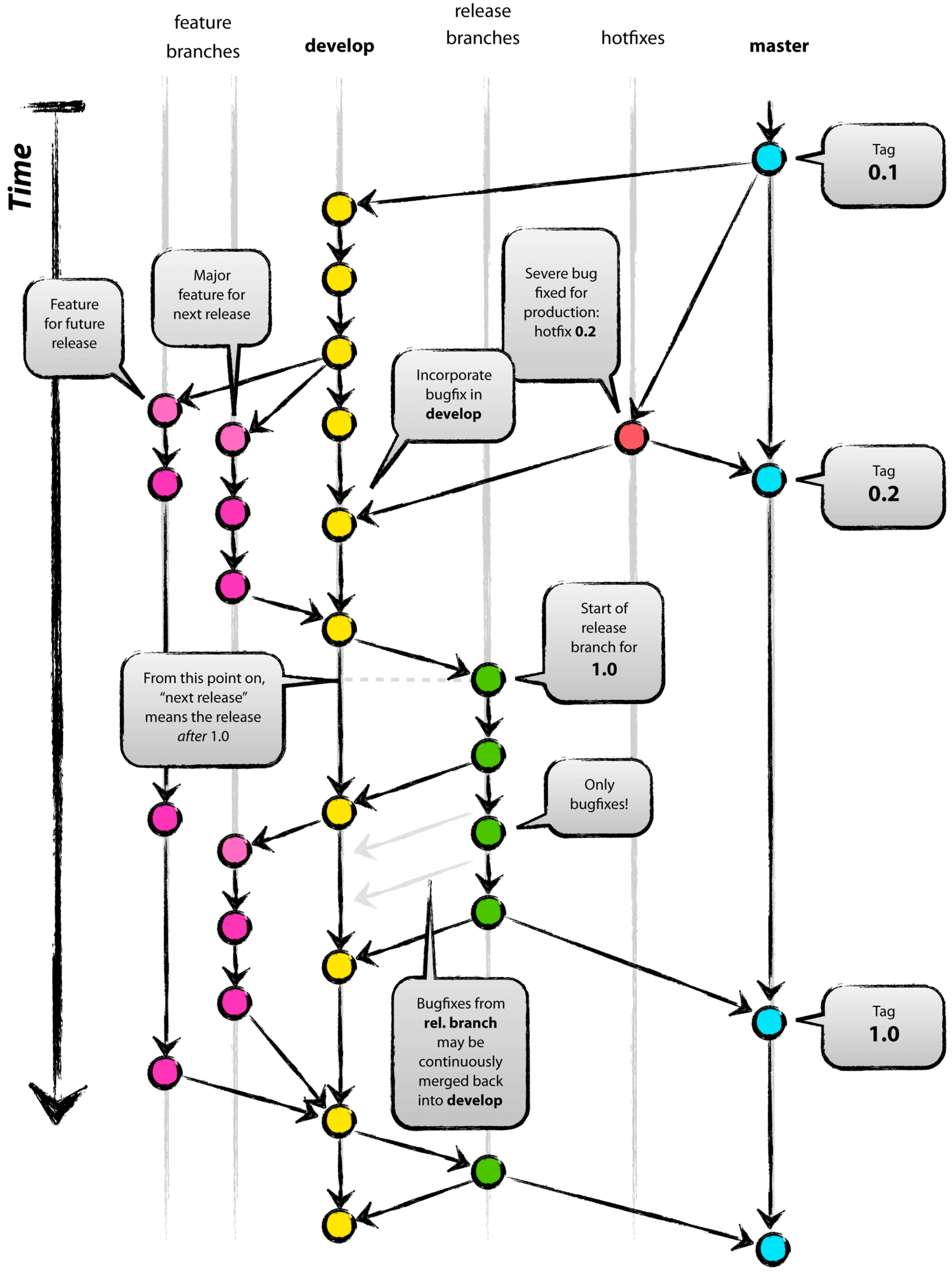
$$s_1 \xleftarrow{u} s_2 \quad \equiv \quad s_2 = f(s_1, u)$$

hash function (pretend injective)

$$\mathrm{hash} : \mathrm{char}^\star \to 2^{128}$$

"Git Flow" branching model

- Git does not impose any semantics on the branches.
- Conventions ("branching models") are useful.
- Git Flow (right) is one of the most widely used.


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

Git/Github demo checklist

- Basics
  - Clone a repository
  - Create a local branch
  - Commit on the branch
  - Push a branch remotely
  - Merge branches
  - Delete branch
- Pull requests
  - Create a pull request
  - Review / approve / merge
  - Issues

Resources for learning Git

- Github video guides: **https://www.youtube.com/githubguides**
- Git book: **https://git-scm.com/book/en/v2**
- The Github tutorial: **https://guides.github.com/activities/hello-world/**
- The Github flow guide: **https://guides.github.com/introduction/flow/**