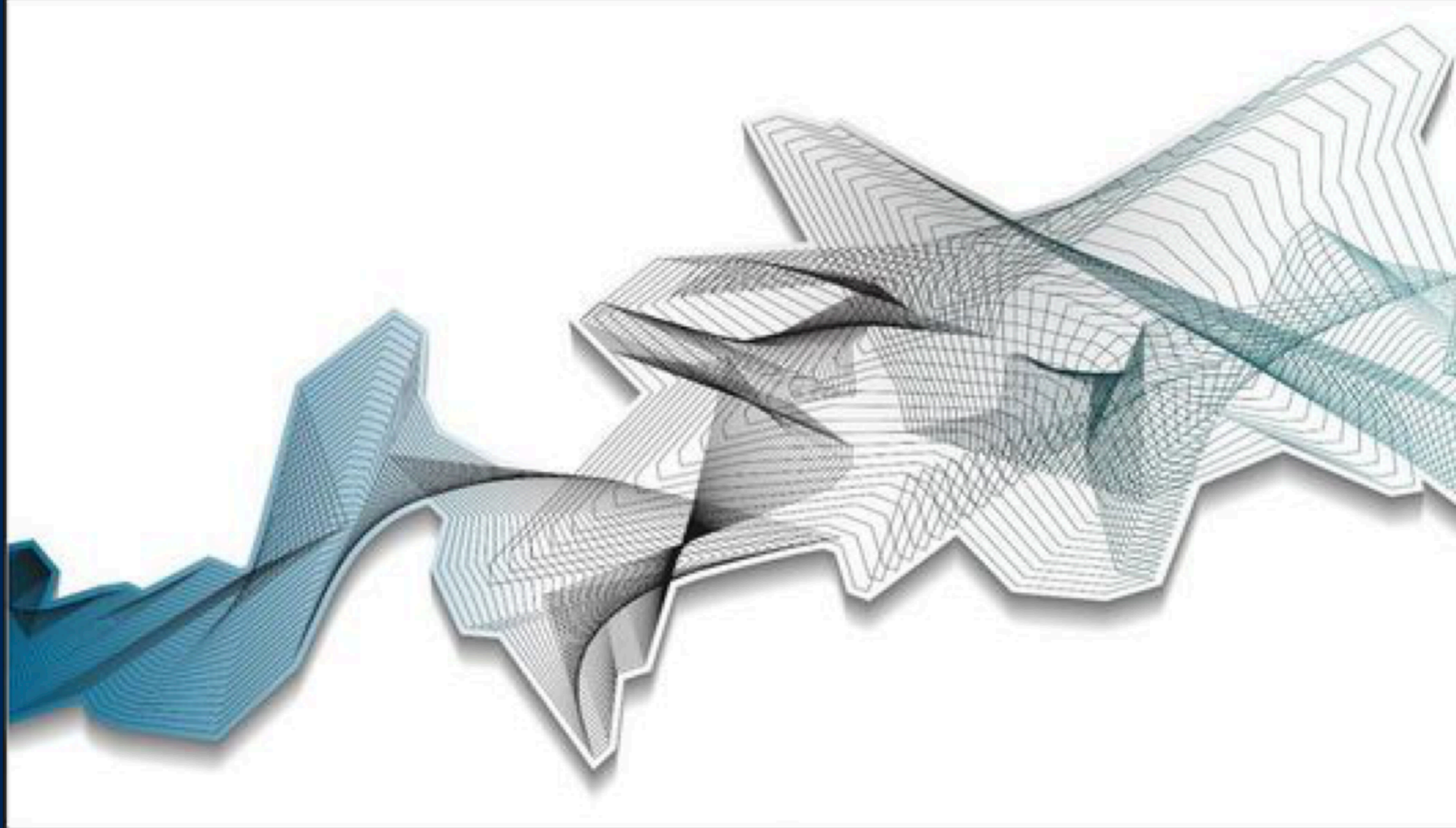


git at scale: 2019

brettkoonce.com/talks

february 24th, 2019



- **git gets easier once you get the basic idea that branches are homomorphic endofunctors mapping submanifolds of a Hilbert space.**

overview

- **who uses homebrew, git, open source?**
- **how to make a good pull request**
- **what a maintainer does, rebasing**
- **project/release management**
- **purpose: convince you to make a contribution to an open source project**

version control

- **why use it? what is the purpose?**
- **project history, save progress**
- **concentrate on a subset of codebase**
- **reason about changes more simply**
- **allows us to collaborate with others, future self**

st. linus quote

- what he thinks a good pr looks like
- <https://github.com/torvalds/subsurface-for-dirk/blob/a48494d2fbed58c751e9b7e8fbff88582f9b2d02/README#L88>

Also, please write good git commit messages. A good commit message looks like this:

Header line: explain the commit in one line (use the imperative)

Body of commit message is a few lines of text, explaining things in more detail, possibly giving some background about the issue being fixed, etc etc.

The body of the commit message can be several paragraphs, and please do proper word-wrap and keep columns shorter than about 74 characters or so. That way "git log" will show things nicely even when it's indented.

Make sure you explain your solution and why you're doing what you're doing, as opposed to describing what you're doing. Reviewers and your future self can read the patch, but might not understand why a particular solution was implemented.

Reported-by: whoever-reported-it

Signed-off-by: Your Name <youremail@yourhost.com>

where that header line really should be meaningful, and really should be just one line. That header line is what is shown by tools like gitk and shortlog, and should summarize the change in one readable line of text, independently of the longer explanation. Please use verbs in the imperative in the commit message, as in "Fix bug that...", "Add file/feature ...", or "Make Subsurface..."

what i like

- **single tree with release tags**
- **clean descriptions, common keywords**
- **lots of little commits and branches**
- **rapid merges and releases**
- **my config: git + textmate + gitx + github**

how to make a pr

- **demo of pull request with github + homebrew + package**
- **list of packages to be upgraded**
- **free commits for whomever wants them**
- **if you know git already, help somebody else**

**WRITES ALL THE DNA CODE
FOR ALL SPECIES**

**DOESN'T WRITE A SINGLE
COMMENT**

quickmeme.com

maintenance

you're gonna forget

- **why was this bug a big deal?**
- **what hardware did you test it on?**
- **which other tools did we use along the way?**
- **what other stuff was going on at the time?**
- **coding a solution is half the solution, make it future proof**

maintenance

- **code cleanup and style**
- **document history of project**
- **gatekeeper for features, project releases**
- **grow project —> need more contributors**
- **don't scare the newbies!**

git rebase

- **moving commits around**
- **rewording descriptions**
- **fixing spelling mistakes, whitespace issues**
- **rebasing onto master**
- **merging/splitting a branch**

release workflow



release workflow

- **release current version, prep next version**
- **bump api/version number, tag commit**
- **rebase release branch**
- **hand upload first build, continuous integration**
- **daily release cycle**

readme.md

- **add a license to a new repo**
- **markdown all the things**
- **document your build process**
- **give to somebody else, have them make build**
- **checkin keys, try to keep docs up to date**



What my friends think I do



What my mom thinks I do



What society thinks I do



What my boss thinks I do



What I think I do



What I actually do

goto end

why do i code?

minor spelling fix, trys->tries #9173

Closed asparagui wants to merge 1 commit into `Homebrew:master` from `asparagui:patch-1`

Conversation 1 Commits 1 Checks 0 Files changed 1

Changes from all commits File filter... Jump to... +1 -1

Diff settings

Review changes

2 Library/Formula/luarocks.rb

Copy path

View file



```
@@ -9,7 +9,7 @@ class Luarocks < Formula
  9      9
 10     10      depends_on use_luajit? ? 'luajit' : 'lua'
 11     11
 12     - fails_with_llvm "Lua itself compiles with llvm, but may fail when other software trys to link."
 12     + fails_with_llvm "Lua itself compiles with llvm, but may fail when other software tries to link."
 13     13
 14     14      def patches
 15     15          DATA if HOMEBREW_PREFIX.to_s == '/usr/local'
```

- code for self, code for \$\$, code for others

your homework

- **make a commit to something**
- **find a project you like/use and help**
- **doesn't have to be code**
- **docs, wiki, faq, documentation (or \$\$)**

golden rule of git

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.



thanks for coming!