Next Generation Researcher Skills

Software and Data Carpentry

Feb 11, 2016

Jonah Duckles, Executive Director of The Software Carpentry Foundation

jduckles@software-carpentry.org - @jduckles

History

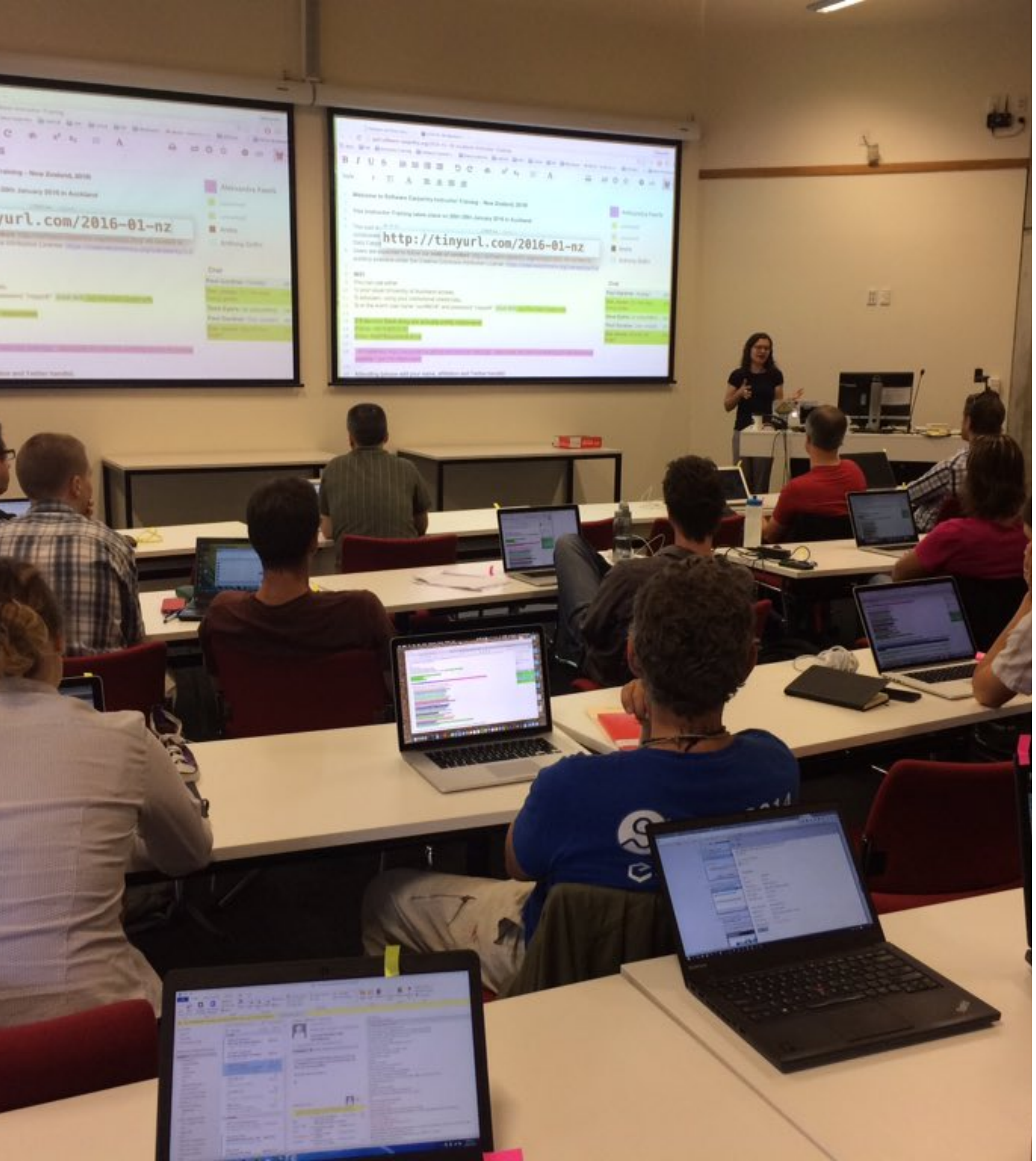
- Greg Wilson and Brent Gorda at Los Alamos National Laboratory
- Taught basic software engineering skills to scientists
- Five full days
- Online lessons
- In 2011 support from Moore and Sloan foundations
- 2-day workshop

Reboot What we learned

- peer instruction!
- helpers + type-along pedagogy
- instructional material as collaborative object
- fun and dynamic community of instructors

We need training!

- We often think of "training" as gear that once we have it, our problems will be solved.
- Technical experts often don't think about the beginner's mind and how to target a training to the needs of their audience.
- To be useful training has to be impactful



Instructor training

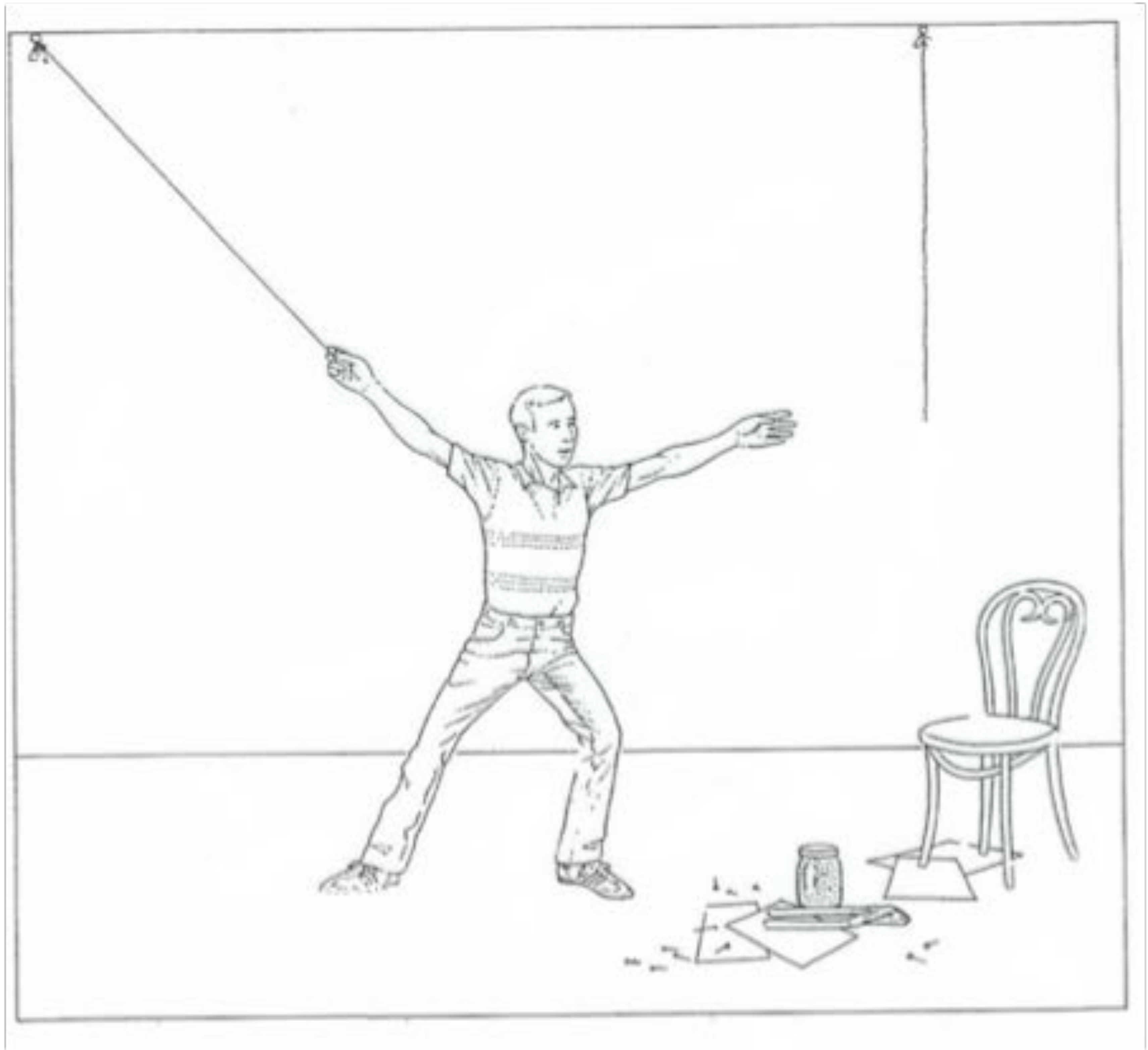
- We train scientists how to teach technology skills with more impact:
  - various pedagogical approaches
  - take into account learner's background
  - reduce cognitive load
  - give and receive constructive feedback

Jugyokenkyu - lesson study

- Coordinated collaboration, testing and continuous improvement of lessons.
- Collaboration on lessons, conversation about teaching of lessons
- Instructor community that discusses the ongoing improvement of a lessons
- Kaizen of teaching

Preparing the unconscious mind

- Problem solving skills can be inaccessible to the conscious mind
- How can we hack our unconscious to work on problems when we're not thinking about them?
- Self directed play is a way to seed the unconscious mind
- Training is more than just the things you teach



The path from novice to expert

- prepare the unconscious mind to solve problems
- mentorship
- play, tinkering
- community support

Teaching as performance art

- Excitement
- Engagement
- Passion for the topic
- Improv
- Lessons are a loosely sketched script

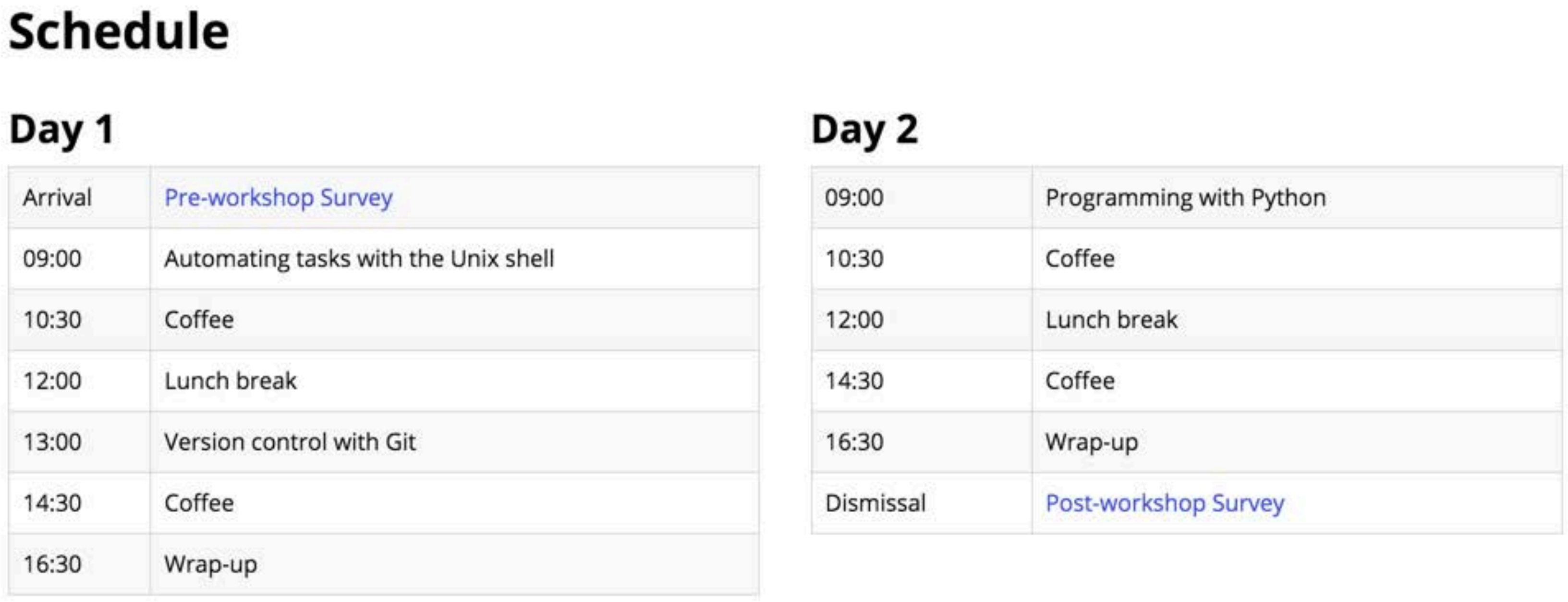
software carpentry

Our Workshops

- 2-days, max
- team taught (at least two instructors)
- sticky notes
- challenges
- helpers
- feedback

About Software Carpentry

- Unix Shell
- A programming language, Python, R, or Matlab
- Version control (Git, Svn or Mercurial)



Schedule

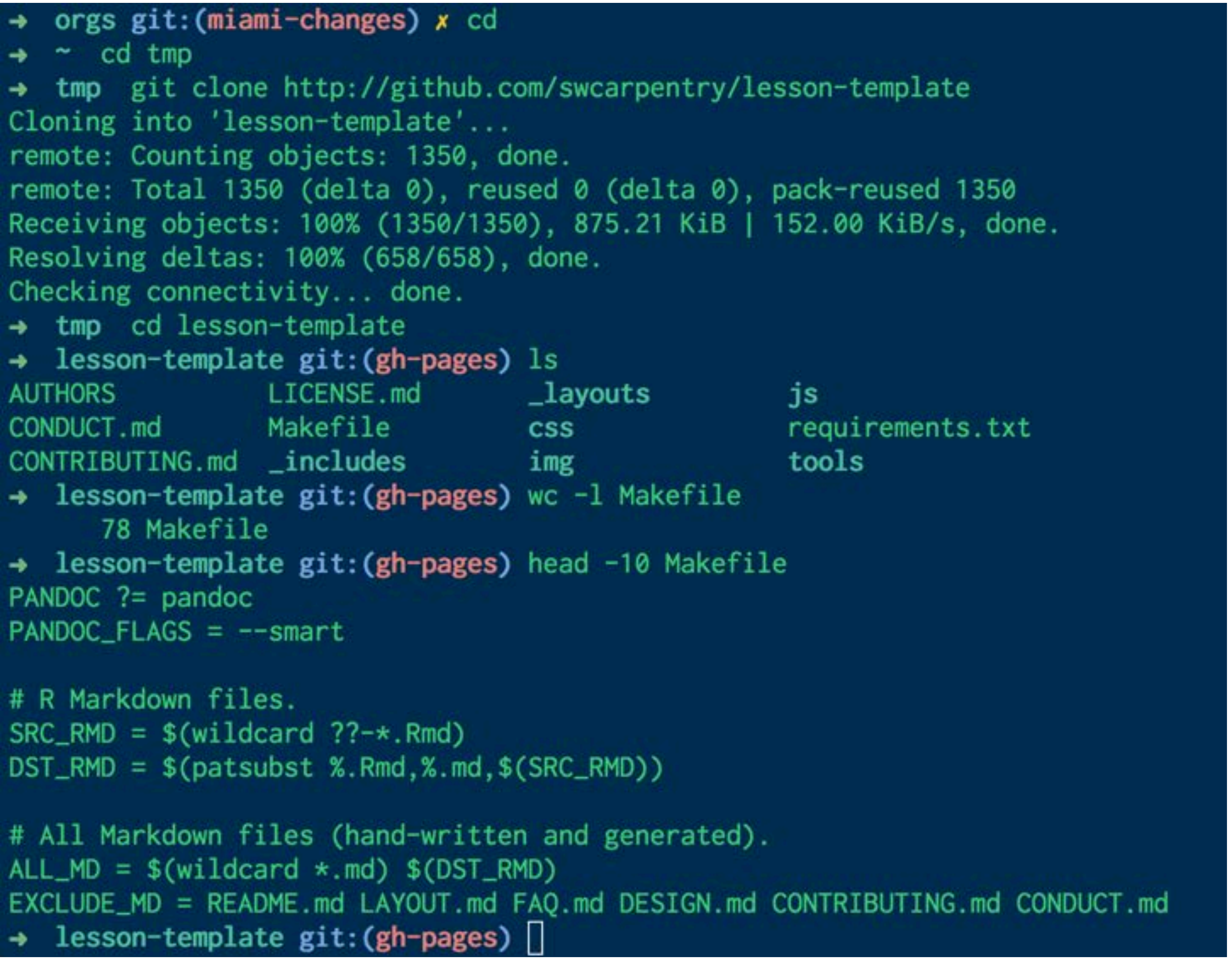
Day 1

Arrival	Pre-workshop Survey
09:00	Automating tasks with the Unix shell
10:30	Coffee
12:00	Lunch break
13:00	Version control with Git
14:30	Coffee
16:30	Wrap-up

Day 2

09:00	Programming with Python
10:30	Coffee
12:00	Lunch break
14:30	Coffee
16:30	Wrap-up
Dismissal	Post-workshop Survey

Unix Shell



```
➜ orgs git:(miami-changes) ✗ cd
➜ ~  cd tmp
➜ tmp  git clone http://github.com/swcarpentry/lesson-template
Cloning into 'lesson-template'...
remote: Counting objects: 1350, done.
remote: Total 1350 (delta 0), reused 0 (delta 0), pack-reused 1350
Receiving objects: 100% (1350/1350), 875.21 KiB | 152.00 KiB/s, done.
Resolving deltas: 100% (658/658), done.
Checking connectivity... done.
➜ tmp  cd lesson-template
➜ lesson-template git:(gh-pages) ls
AUTHORS          LICENSE.md       _layouts         js
CONDUCT.md       Makefile         css              requirements.txt
CONTRIBUTING.md  _includes        img              tools
➜ lesson-template git:(gh-pages) wc -l Makefile
      78 Makefile
➜ lesson-template git:(gh-pages) head -10 Makefile
PANDOC ?= pandoc
PANDOC_FLAGS = --smart

# R Markdown files.
SRC_RMD = $(wildcard ??-*.Rmd)
DST_RMD = $(patsubst %.Rmd,%.md,$(SRC_RMD))

# All Markdown files (hand-written and generated).
ALL_MD = $(wildcard *.md) $(DST_RMD)
EXCLUDE_MD = README.md LAYOUT.md FAQ.md DESIGN.md CONTRIBUTING.md CONDUCT.md
➜ lesson-template git:(gh-pages) 
```

- Demystification of command line incantations
- Read → Evaluate → Print loop
- Automating tasks based on file names
- A simple interface to all "programs"

Programming language

Version Control

DATA CARPENTRY

MAKING DATA SCIENCE MORE EFFICIENT

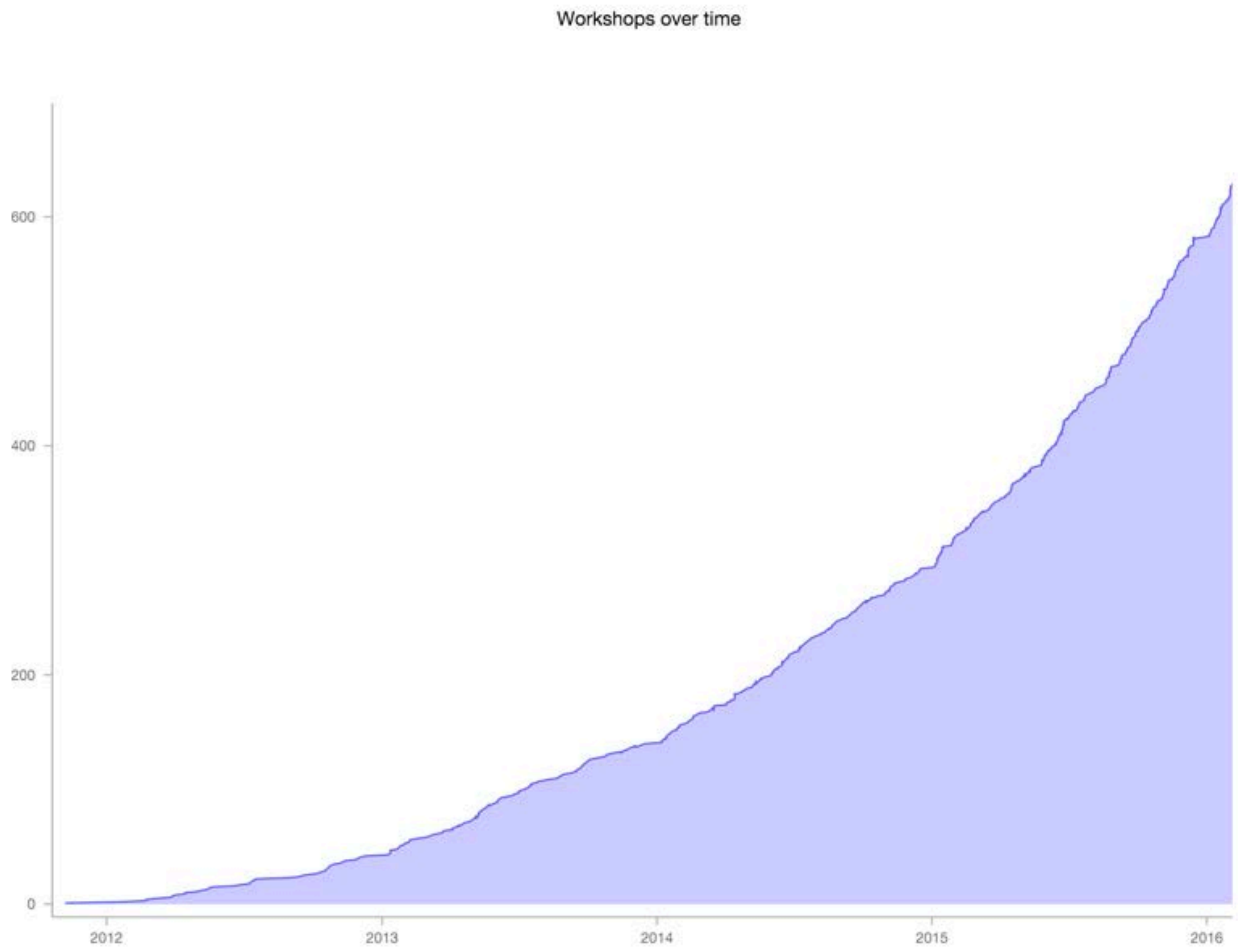
Data Carpentry

- Focused on data scientist skills necessary for research
- Lessons for discipline-specific needs
- Not as much about creating code:
    - scripting existing code
    - using tools that leave reproducibility artifacts
- Best practices in data management

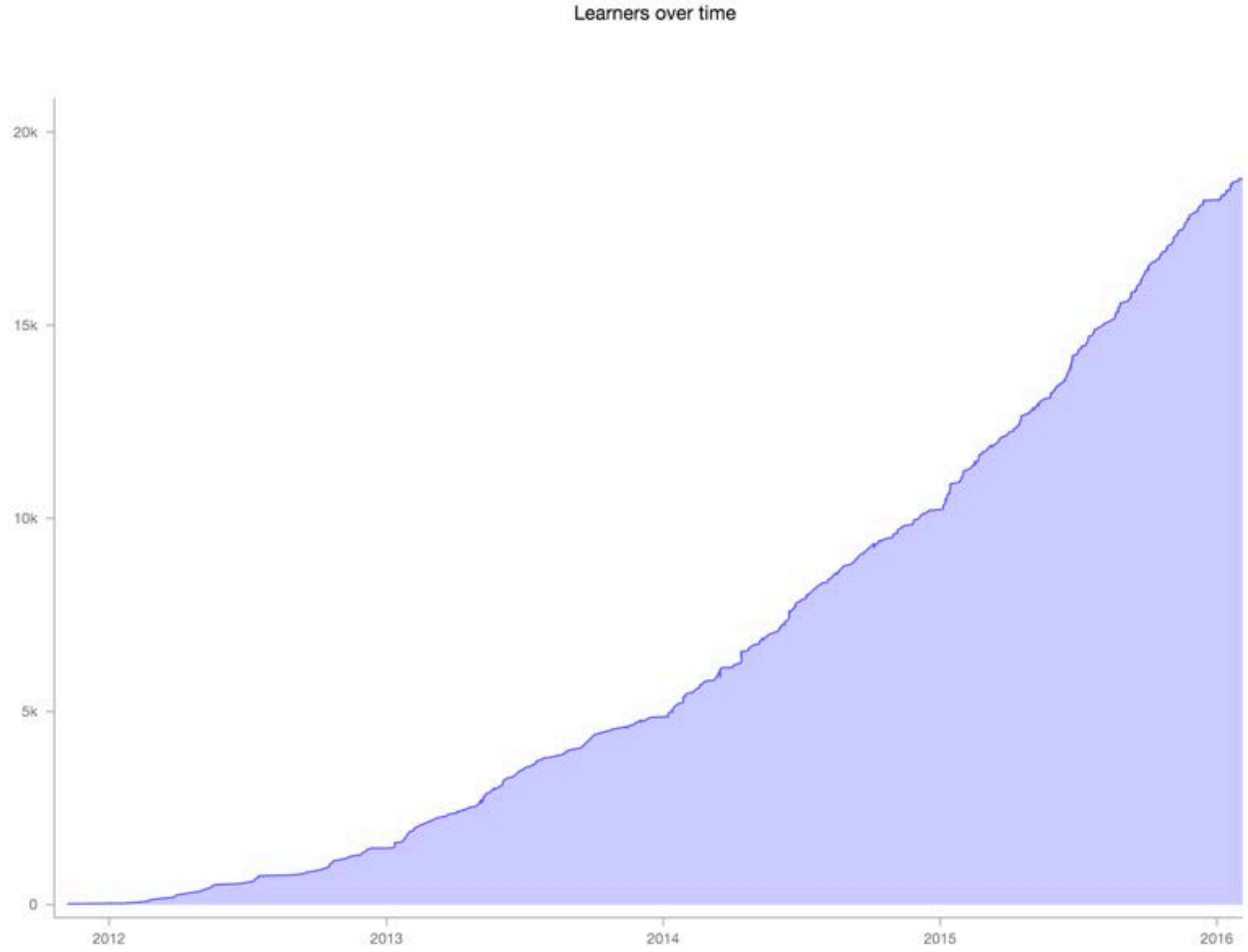
What does Software Carpentry the organization do?

- Coordinate instructors
- Run workshops
- Mentorship communities for instructors
- Catalyze capacity building at member orgs
- Curate lessons
- Support and expand on the culture of collaborative instruction

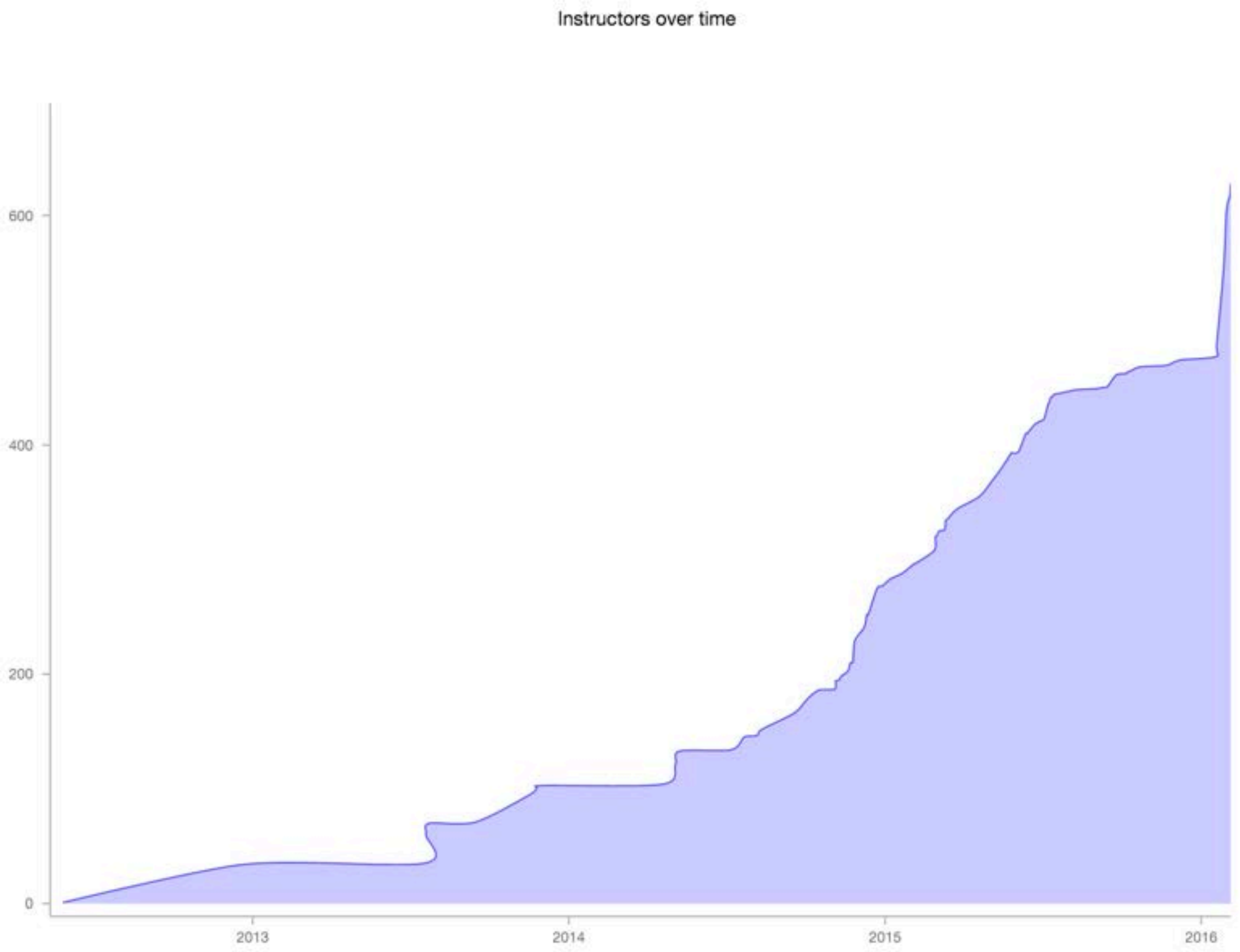
Workshops



Learners



Instructors



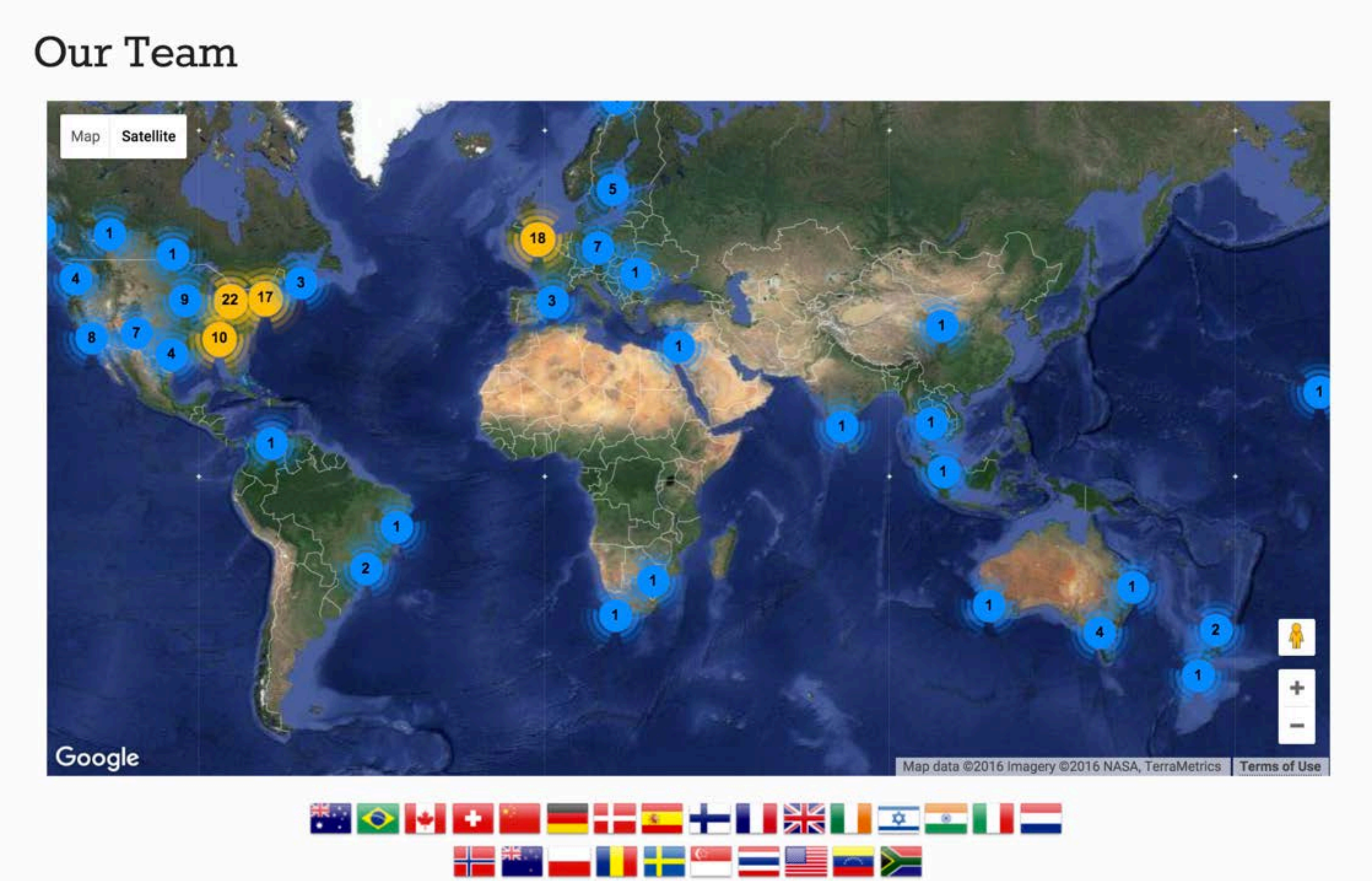
WE WANT YOU

Organizational Memberships

- We train instructors in your community
- They join a global mentorship community
- Two tiers, train either ~6 instructors per year or ~15 per year

Our Team

500 instructors



Our Team

Thank You!

Jonah Duckles, Executive Director

jduckles@software-carpentry.org

Twitter: @swcarpentry @jduckles

Run a workshop - admin@software-carpentry.org