Software Carpentry: Lessons Learned

Greg Wilson
Anatole France (1844-1924)

“The law, in its majestic equality, forbids the rich and poor alike to sleep under bridges, to beg in the streets, and to steal bread.”
Now

Anatole France (1844-1924)

“The law, in its majestic equality, forbids the rich and poor alike to sleep under bridges, to beg in the streets, and to steal bread.”

Today, thanks to computers, every scientist can devote her working life to getting software installed.
5-15% GPU clusters to analyze petabytes in the cloud

85-95% Sending each other spreadsheets by email
Surely You're Exaggerating

1. How many graduate students write shell scripts to analyze each new data set instead of running those analyses by hand?
Surely You're Exaggerating

2. How many of them use version control to keep track of their work and collaborate with colleagues?
Surely You're Exaggerating

3. How many routinely break large problems into pieces small enough to be
   - comprehensible,
   - testable, and
   - reusable?
Surely You're Exaggerating

3. How many routinely break large problems into pieces small enough to be
   - comprehensible,
   - testable, and
   - reusable?
   And how many know those are the same things?
Goalposts

A *computationally competent* scientist can:

- Manage and process data
- Tell if it's been processed correctly
- Find and fix problems when it hasn't been
- Keep track of what she has done
- Share her work with others

*Efficiently*
It Is Therefore Obvious That...

- Put more computing courses in the curriculum!

- But it's already full
It Is Therefore Obvious That...

- Put a little computing in every course!
  - 5 minutes/lecture = 4 courses/degree
  - First thing cut when running late
It Is Therefore Obvious That...

- And no matter what we do...

- The blind leading the blind
If you build a man a fire,
you'll keep him warm for a night.
If you set a man on fire,
you'll keep him warm for the rest of his life.

— Terry Pratchett
What We Teach

Unix shell
Version control
Python/R/MATLAB
SQL
Make
What We Actually Teach

Unix shell => Task automation
Version control => Track and share work
Python/R/MATLAB => Modular programming
SQL => Data management
Make => Reproducibility
How It's Going

Workshops
How It's Going

Workshops
How It's Going

Learners
How It's Going

Instructors
How It's Going

Instructors
What We Learned (Version 1)

- Software engineering isn't appropriate for most scientists
- Week-long workshops are easy to schedule, but bad for learning
What We Learned (Versions 2-3)

- Hard to fit this into existing curricula
- Hard to convince Computer Science departments to care
What We Learned (Version 4)

- Videos aren't cost-effective
- “It's the maintenance, stupid”
And Now... Version 5

- Instructor training creates community
- Collaborative lesson development
- Early joiners are atypical
And Now... Version 5

- Every partner has different needs
- People would rather argue about technology than pedagogy
And Now... Version 5

- Never teach alone
- Learners use their own machines
- Live coding
And Now... Version 5

- Sticky notes
- Collaborative note taking
- Debriefing
And Now... Version 5

- Iterate
- Iterate
- Iterate
Why People Volunteer

- Make the world a better place
- Self-defense
- Learn this stuff themselves
- Make new friends
- Boost their careers
Learning About Learning
A Puzzle

- Thousands contribute patches to open source software projects
- Millions have edited Wikipedia
- Why don't people build lessons this way?
All Together Now

- 187 contributors to our lessons in the run-up to publication
- A *culture of contribution*
The Model Transfers

- Domain-specific lessons
- Shared instructor pool
- Next: librarians, humanities, ...
Why You Should Care

• Make scientists more productive
• A new model for curriculum development
• Give everyone a say in shaping 21st Century science
How You Can Help

- Come learn
- Host a workshop
- Become an instructor
- Contribute to our lessons
- Build tools
Thank You

software carpentry

http://software-carpentry.org