Software Carpentry:

What We've Done, What We've Learned, Where We're Going

Greg Wilson

ICERM Workshop on Reproducibility in Computational and Experimental Mathematics

December 2012

What Would Ya Say Ya Do Here?

TL;DR

- I usually describe myself as a teacher
- My real job is to sell scientists on the idea of better computing practices
- Most don't care about quality or reproducibility because there's no incentive to do so
- But they *do* care—a lot—about productivity
- So my pitch is, "The only way to work faster is to work better."
- Followed by, "And I can prove it."

- Which is mostly untrue...

History

- Did HPC in the 1980s and 1990s
- Eventually realized I was doing more harm than good
- Started teaching at LANL in 1998: "show up or shut up"
- Updated and open sourced materials 2004-06
- April 2010-April 2011: full-time teaching and creating online videos
- Jan 2012-present: full-time again for Mozilla, funded by Sloan Foundation

What We Teach

- Unix shell
- Subversion
- Python
- SQL
- xUnit

What We Teach

- Unix shell
- Subversion
- Python
- SQL
- xUnit

bait and switch

What We *Actually* Teach

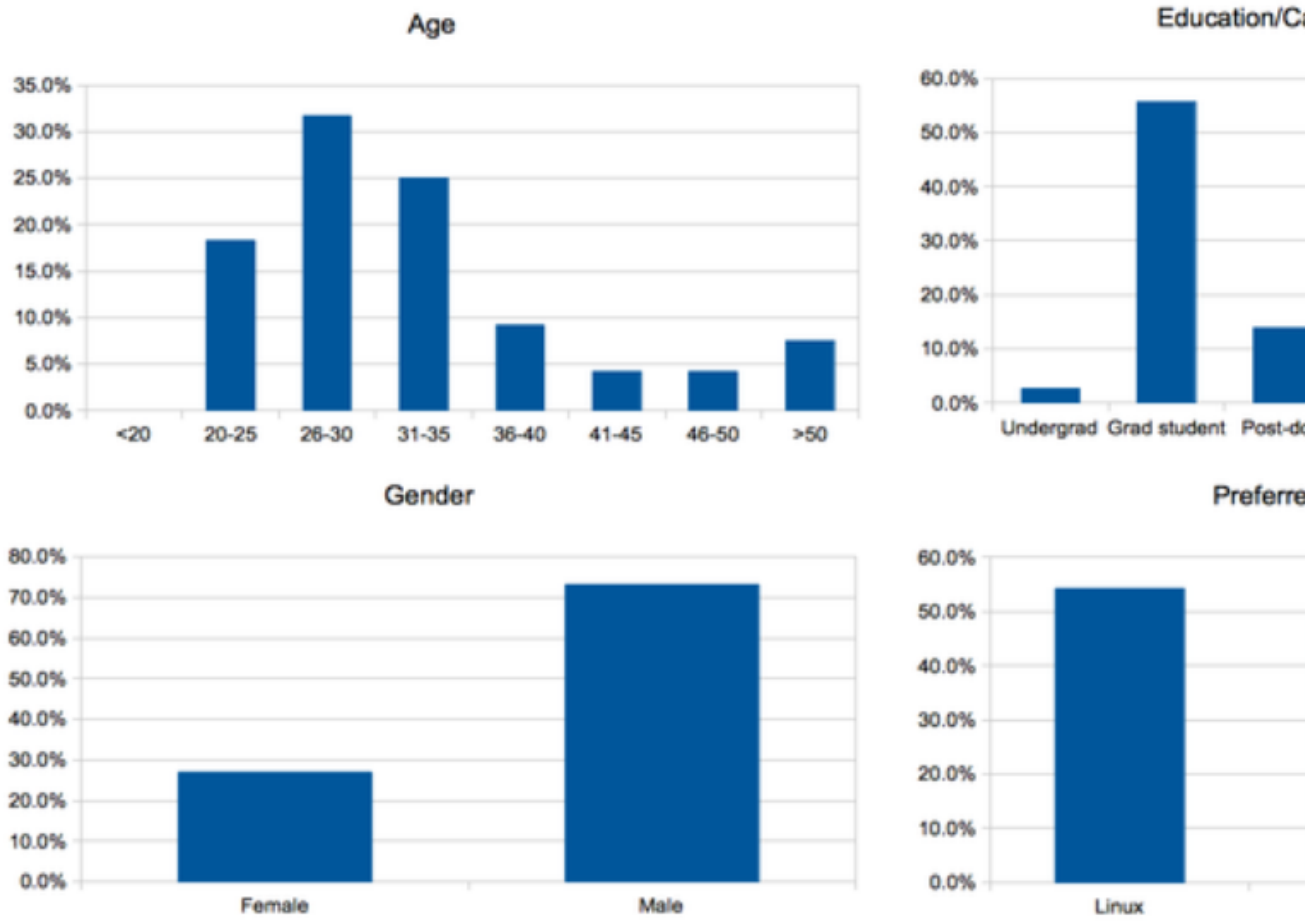
- A program is just another piece of lab equipment
- Programming is a human activity
- Little pieces loosely joined
- Let the computer repeat it
- Paranoia makes us productive
- Better algorithms beat better hardware

What We *Actually* Teach

- A program is just another piece of lab equipment
- Programming is a human activity
- Little pieces loosely joined
- Let the computer repeat it
- Paranoia makes us productive
- Better algorithms beat better hardware

how to think like a programmer

Who We Teach



Who We Teach

Who We Are






Where We've Been

Example: Version Control with Subversion

- Check out, write a short biography, add and commit:
  "Look, a better Dropbox"
- Edit someone else's:
  "Look, you can merge conflicts"
- Put `$Revision: 123$` in files and print to output:
  "Look, provenance"
- *A biscuit every 10 minutes*

Counter-Example: Version Control with Git

- Simple things are more complicated (even with `git commit -am`)
- Most tutorials emphasize branching, but that's a step too far for novices
- No simple out-of-the-box support for provenance (version "numbers" aren't)

Counter-Example: Version Control with Git

- The tool is intrinsically more complicated...
- ...because it's doing more (and more complicated) things
- So gratification is delayed and short-term reward/pain is lower
- But all the cool kids are hanging out on GitHub these days...

What We've Learned

- Most scientists regard programming as a tax

What We've Learned

- Most scientists regard programming as a tax
- Most don't care about quality or reproducibility

What We've Learned

- Most scientists regard programming as a tax
- Most don't care about quality or reproducibility
- They *do* care about productivity

What We've Learned

- Most scientists regard programming as a tax
- Most don't care about quality or reproducibility
- They *do* care about productivity
- They learn better from and among their peers

What We've Learned

- Most scientists regard programming as a tax
- Most don't care about quality or reproducibility
- They *do* care about productivity
- They learn better from and among their peers
- And they respond well to peer-reviewed research

Best Practices

Greg Wilson, D.A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H.D. Haddock, Katy Huff, Ian M. Mitchell, Mark Plumbley, Ben Waugh, Ethan P. White, and Paul Wilson: "Best Practices in Scientific Computing"
(http://arxiv.org/abs/1210.0530).

Scientists spend an increasing amount of time building and using software. However, most scientists are never taught how to do this efficiently. As a result, many are unaware of tools and practices that would allow them to write more reliable and maintainable code with less effort. We describe a set of best practices for scientific software development that have solid foundations in research and experience, and that improve scientists' productivity and the reliability of their software.

Our Goal

Number of Reviewers	Fraction of Papers
2	10%
3	40%
4	40%
5	10%

P(individual reviewer is a believer)	20%
P(at least one reviewer is a believer)	53.5%

Obstacles

- The curriculum is full

Obstacles

- The curriculum is full
- The blind leading the blind

Obstacles

- The curriculum is full
- The blind leading the blind
- The top 5% don't realize how big the gap is

Obstacles

- The curriculum is full
- The blind leading the blind
- The top 5% don't realize how big the gap is
- Installation, installation, installation...

Obstacles

- The curriculum is full
- The blind leading the blind
- The top 5% don't realize how big the gap is
- Installation, installation, installation...
- *The payoff is distant and uncertain*

So You Wanna Change the World, Eh?

Conclusion

- Our sales pitch mostly works for most scientists

Conclusion

- Our sales pitch mostly works for most scientists
- We'll reach 2000 people in 2013, and double that in 2014

Conclusion

- Our sales pitch mostly works for most scientists
- We'll reach 2000 people in 2013, and double that in 2014
- We'd like to teach reproducibility, but we don't know how

Conclusion

- Our sales pitch mostly works for most scientists
- We'll reach 2000 people in 2013, and double that in 2014
- We'd like to teach reproducibility, but we don't know how
- Would you like to attend/host/teach?

Conclusion

- Our sales pitch mostly works for most scientists
- We'll reach 2000 people in 2013, and double that in 2014
- We'd like to teach reproducibility, but we don't know how
- Would you like to attend/host/teach?
- *What can you give me to sell them?*

Conclusion

- Our sales pitch mostly works for most scientists
- We'll reach 2000 people in 2013, and double that in 2014
- We'd like to teach reproducibility, but we don't know how
- Would you like to attend/host/teach?
- *What can you give me to sell them?*
- *How the hell do we evaluate this stuff?*

Conclusion

- Our sales pitch mostly works for most scientists
- We'll reach 2000 people in 2013, and double that in 2014
- We'd like to teach reproducibility, but we don't know how
- Would you like to attend/host/teach?
- *What can you give me to sell them?*
- *How the hell do we evaluate this stuff?*

http://software-carpentry.org

info@software-carpentry.org