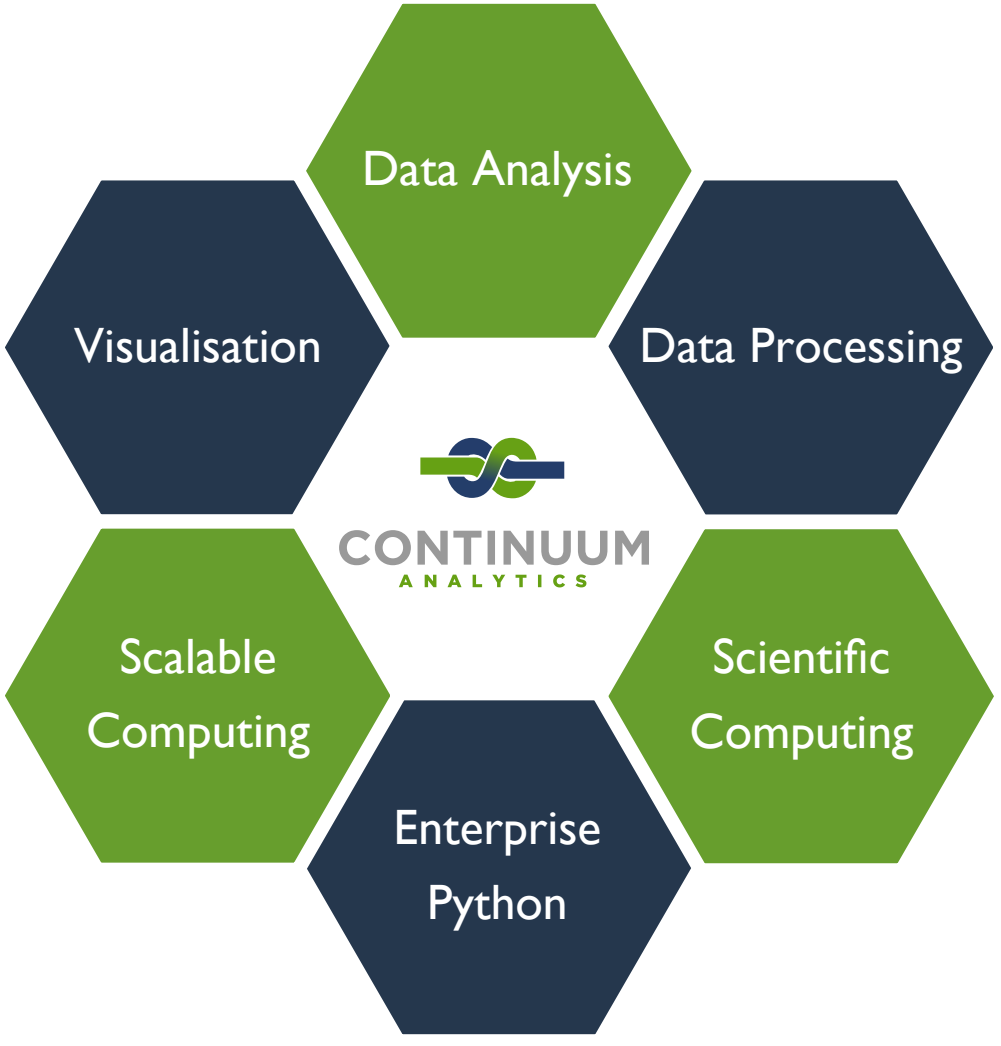
Scientific Computing while Supercomputing

Aron Ahmadia

software carpentry

Continuum Analytics Spring 2013 Sabbatical

- Products
- Training
- Support
- Consulting



CONTINUUM ANALYTICS

This Talk is a Fork

Dark Matter, Public Health, and Scientific Computing

Greg Wilson

software carpentry

“Dark Matter Developers”

Scott Hanselman (March 2012)

[We] hypothesize that there is another kind of developer than the ones we meet all the time. We call them Dark Matter Developers. They don't read a lot of blogs, they never write blogs, they don't go to user groups, they don't tweet or facebook, and you don't often see them at large conferences... [They] aren't chasing the latest beta or pushing...limits, they are just producing.

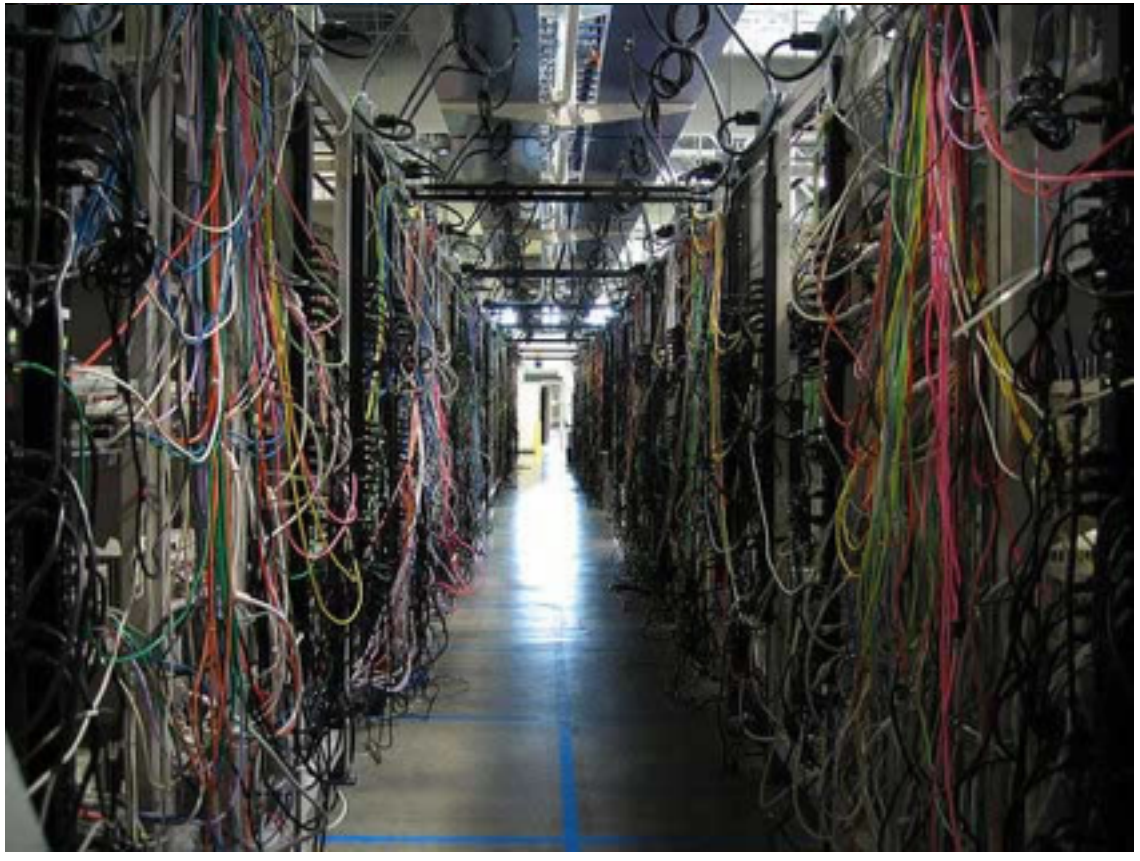
http://www.hanselman.com/blog/DarkMatterDevelopersTheUnseen99.aspx

We're Not That Optimistic

- 90% of scientists have their heads down
  - Doing science instead of talking about using Charm++ to asynchronously distribute their CUDA-accelerated heterogeneous N-body Universe simulations
- Not because they don't *want* to do all that Star Trek stuff
- But because exascale is out of reach

Shiny Toys

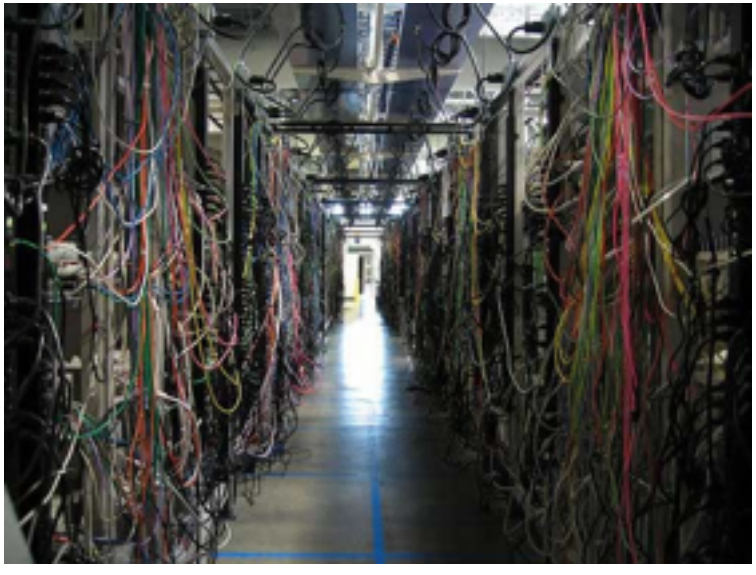
Grimy Reality



So Here We Are

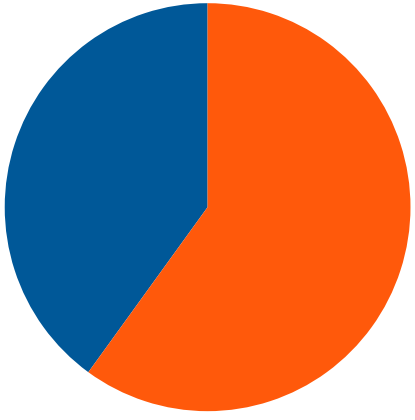
Us

Them



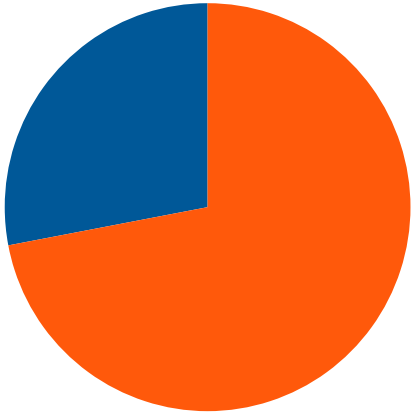
Surely You're Exaggerating

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



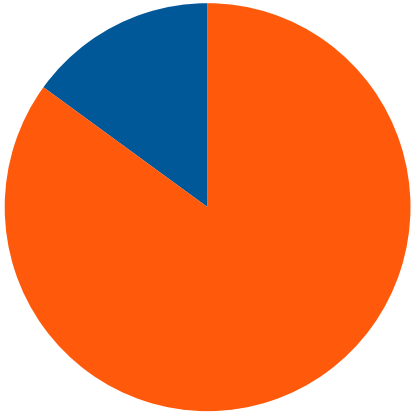
Surely You're Exaggerating

2. How many of them use version control to keep track of their work and collaborate with colleagues?



Surely You're Exaggerating

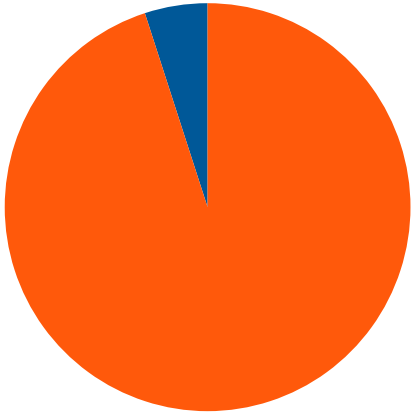
3. How many routinely break large problems into pieces small enough to be
   - comprehensible,
   - testable, and
   - reusable?



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?



We've Left the Majority Behind

We've Left the Majority Behind

Other than
Googling for things,

the majority of scientists

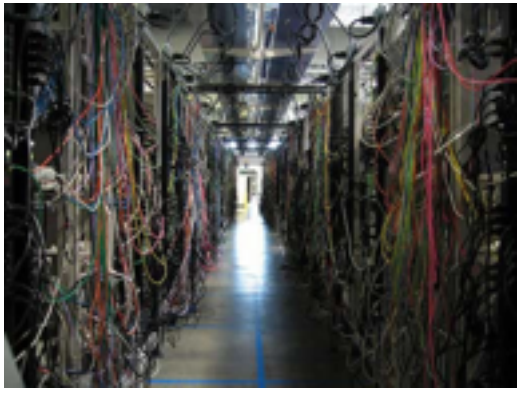
do not use computers

more effectively today

than they did 28 years ago.

“Not My Problem”

Actually your *biggest* problem



If your colleagues aren’t using your computers, you don’t benefit from their contributions.

And Science Loses

“Not My Problem”

Actually your *biggest* problem

If your colleagues **ARE** using your computers: they are:

- filling your front-end with useless jobs
- and filling all of your queues
- and filling all of your disk

And Science Loses

Some Quotes from ATPESC 2013

Your code will outlive the machine. Most successful machine lasts 5 years. The most successful codes may reach 5 decades. There are currently 40 year-old quantum chemistry codes still being used. **- Jeff Hammond**

Design for the future. Software lifecycles should be long, but often are not **- Salman Habib**

I recommend you invest in... nightly test and build, configuration, embedded versioning and metadata **- Pete Beckman**

Where Are Your Goalposts?

A scientist is *computationally competent* if she can build, use, validate, and share software to:

- Manage software and data
- Tell if it's working correctly
- Find and fix problems when it hasn't been
- Keep track of what they've done
- Share work with others
- Do all of these things efficiently

A Driver's License Exam

- Developed with the Software Sustainability Institute for the DiRAC Consortium
- Formative assessment
  - Do you know what you need to know in order to get the most out of this gear?
- Pencils ready?

Note: actual exam allows for several different programming languages, version control systems, etc.

A Driver's License Exam

1. Check out a working copy of the exam materials from a git repository.
2.
3.
4.
5.
6.
7.

Could do it easily	1.0
Could struggle through	0.5
Wouldn’t know where to start	0
Don’t understand the question	-1

A Driver's License Exam

1. Use find and grep in a pipe to create a
2. list of all .dat files in the working copy,
3. redirect the output to a file, and add that
4. file to the repository.
5.
6.
7.

Could do it easily	1.0
Could struggle through	0.5
Wouldn’t know where to start	0
Don’t understand the question	-1

A Driver's License Exam

1. Write a shell script that runs a legacy
2. program for each parameter in a set.
3.
4.
5.
6.
7.

Could do it easily	1.0
Could struggle through	0.5
Wouldn’t know where to start	0
Don’t understand the question	-1

A Driver's License Exam

1. Edit the Makefile provided so that if any
2. .dat file changes, analyze.py is re-run to
3. create the corresponding .out file.
4.
5.
6.
7.

Could do it easily	1.0
Could struggle through	0.5
Wouldn’t know where to start	0
Don’t understand the question	-1

A Driver's License Exam

1. Write four unit tests to exercise a function
2. that calculates running sums. Explain why
3. your four tests are most likely to uncover
4. bugs in the function.
5.
6.
7.

Could do it easily	1.0
Could struggle through	0.5
Wouldn't know where to start	0
Don't understand the question	-1

A Driver's License Exam

1. Explain when and how a function could
2. pass your tests, but still fail on real data.
3.
4.
5.
6.
7.

Could do it easily	1.0
Could struggle through	0.5
Wouldn't know where to start	0
Don't understand the question	-1

A Driver's License Exam

Do a code review of the legacy program from Q3 (about 50 lines long) and list the four most important improvements you would make.

Could do it easily	1.0
Could struggle through	0.5
Wouldn't know where to start	0
Don't understand the question	-1

How Are We Doing?

a) How well did you do?

How Are We Doing?

a) How well did you do?

b) How well do you think most computational scientists would do?

How Are We Doing?

a) How well did you do?

b) How well do you think most computational scientists would do?

c) Do you think a computational scientist could use your petascale supercomputer without having these skills?

How Are We Doing?

a) How well did you do?

b) How well do you think most computational scientists would do?

c) Do you think a computational scientist could use your petascale supercomputer without having these skills?

d) Or a grasp of the **principles** behind them?

So Why Is This *Your* Problem?

If you're only helping the (small) minority lucky enough to have acquired the base skills that use of your supercomputer depends on...

then your potential user base is many times smaller than it could be.

It Is Therefore Obvious That...

- Put more computing courses in the curriculum!
  - Except the curriculum is already full

It Is Therefore Obvious That...

- Put a little computing in every course!
  - Still adds up: 5 minutes/lecture = 4 courses/degree
  - First thing cut when running late
  - The blind leading the blind

What Has Worked

- Target graduate students
- Intensive short courses (2 days to 2 weeks)
- Focus on practical skills

What Has Worked

- Target graduate students
- Intensive short courses (2 days to 2 weeks)
- Focus on practical skills

http://software-carpentry.org


ALFRED P. SLOAN FOUNDATION
1934


mozilla
FOUNDATION


INDIANA UNIVERSITY


MICHIGAN STATE
UNIVERSITY


Microsoft


MITACS


Queen Mary
University of London


SCIMATIC SOFTWARE


SciNet

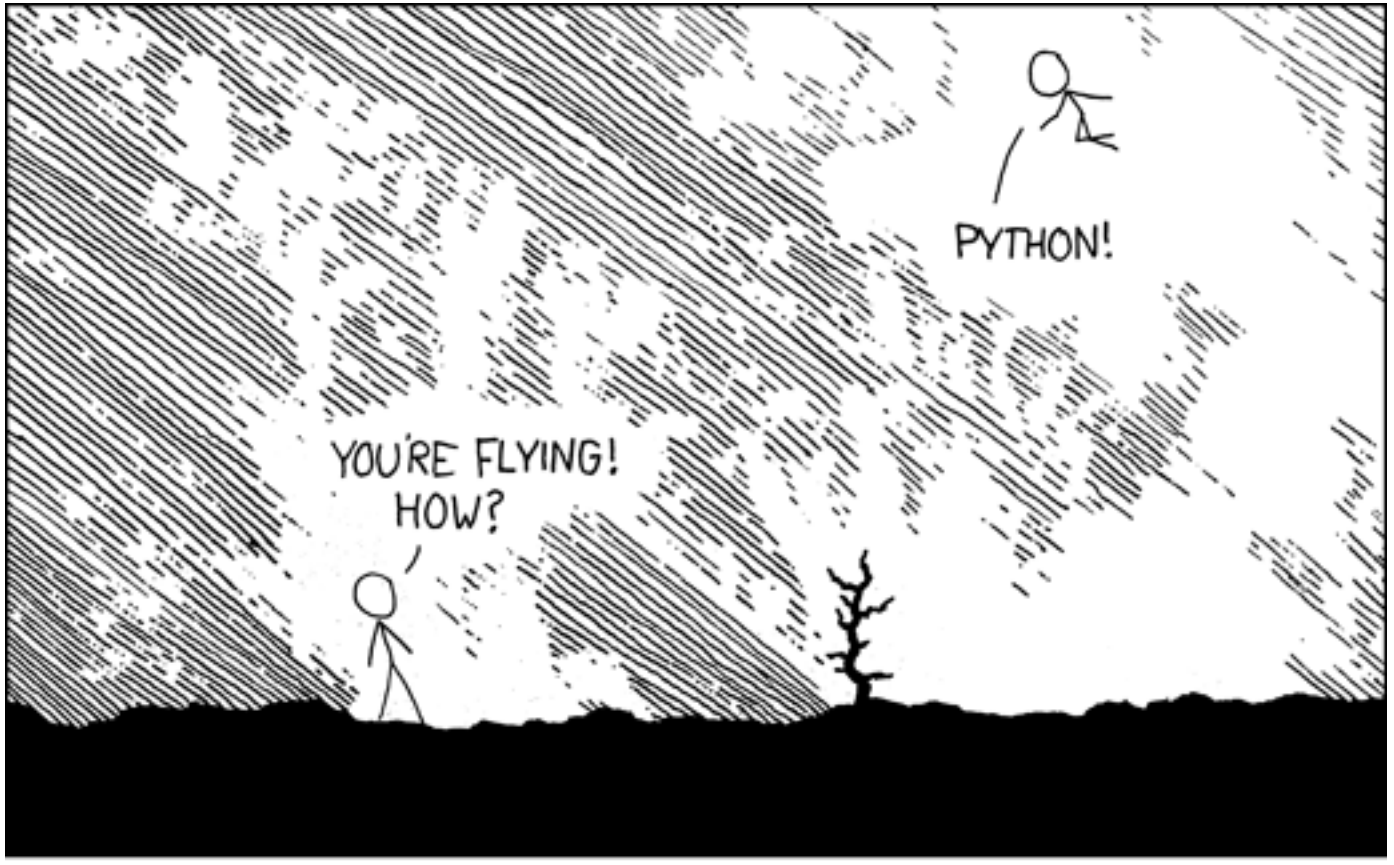

www.software.ac.uk


Met Office

What We Teach

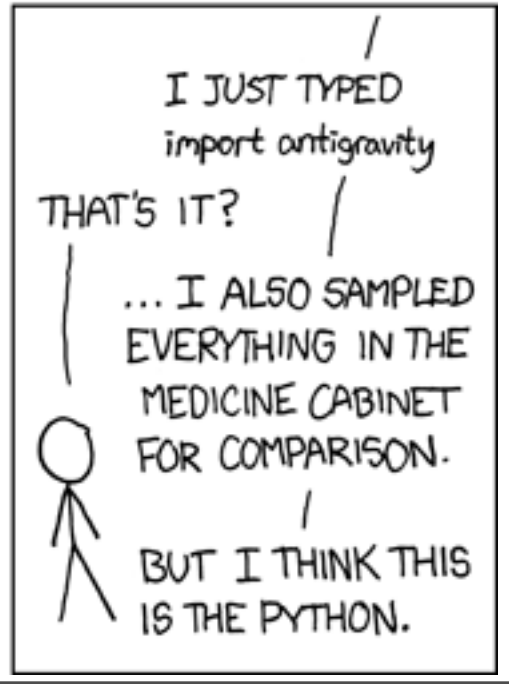
- Unix shell
- Python
- Version control
- Testing
- Array computing,
  image processing,
  SQL, ...

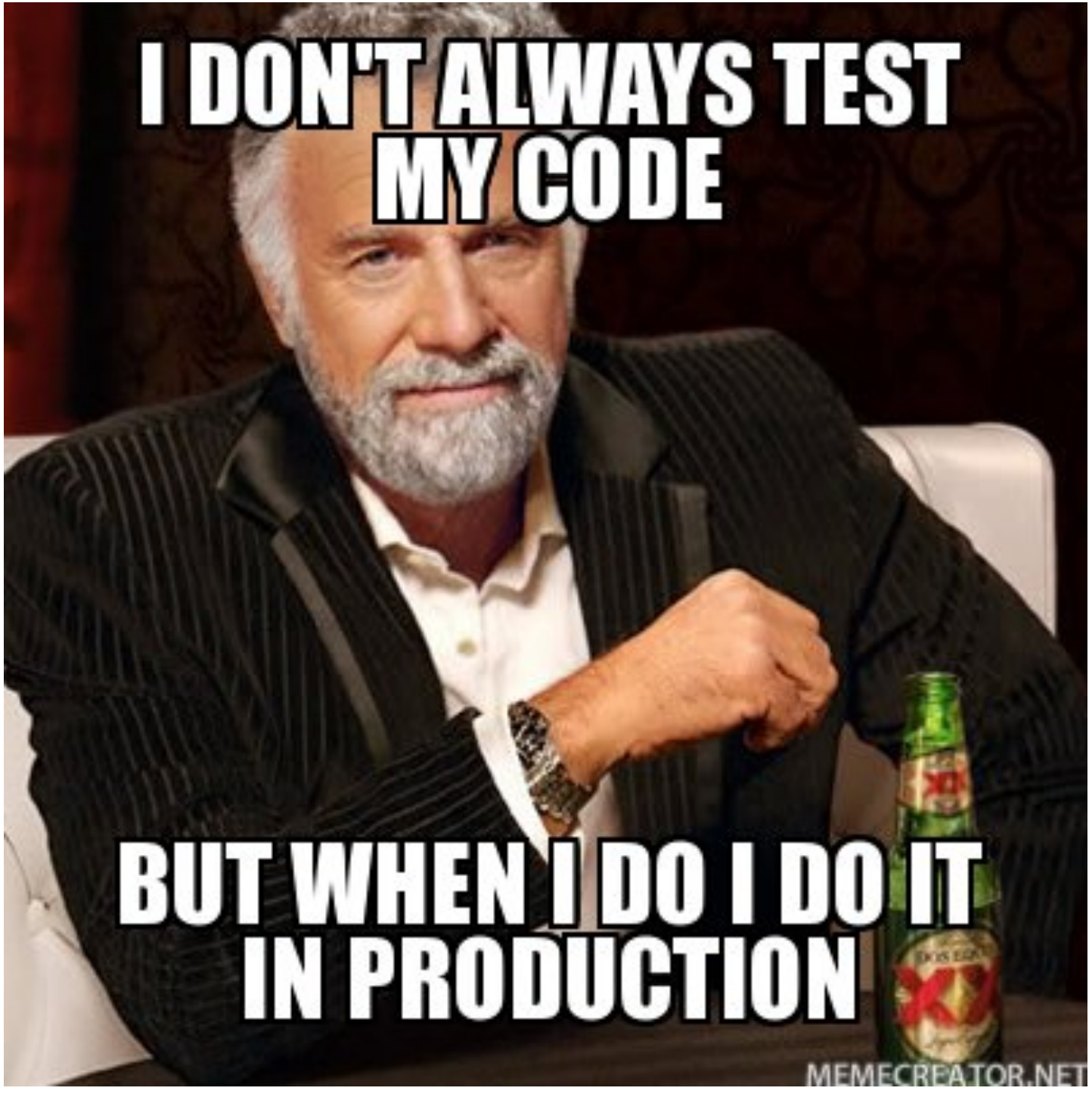

GO AWAY
OR I WILL REPLACE YOU WITH A VERY SMALL SHELL SCRIPT
quickmeme.com

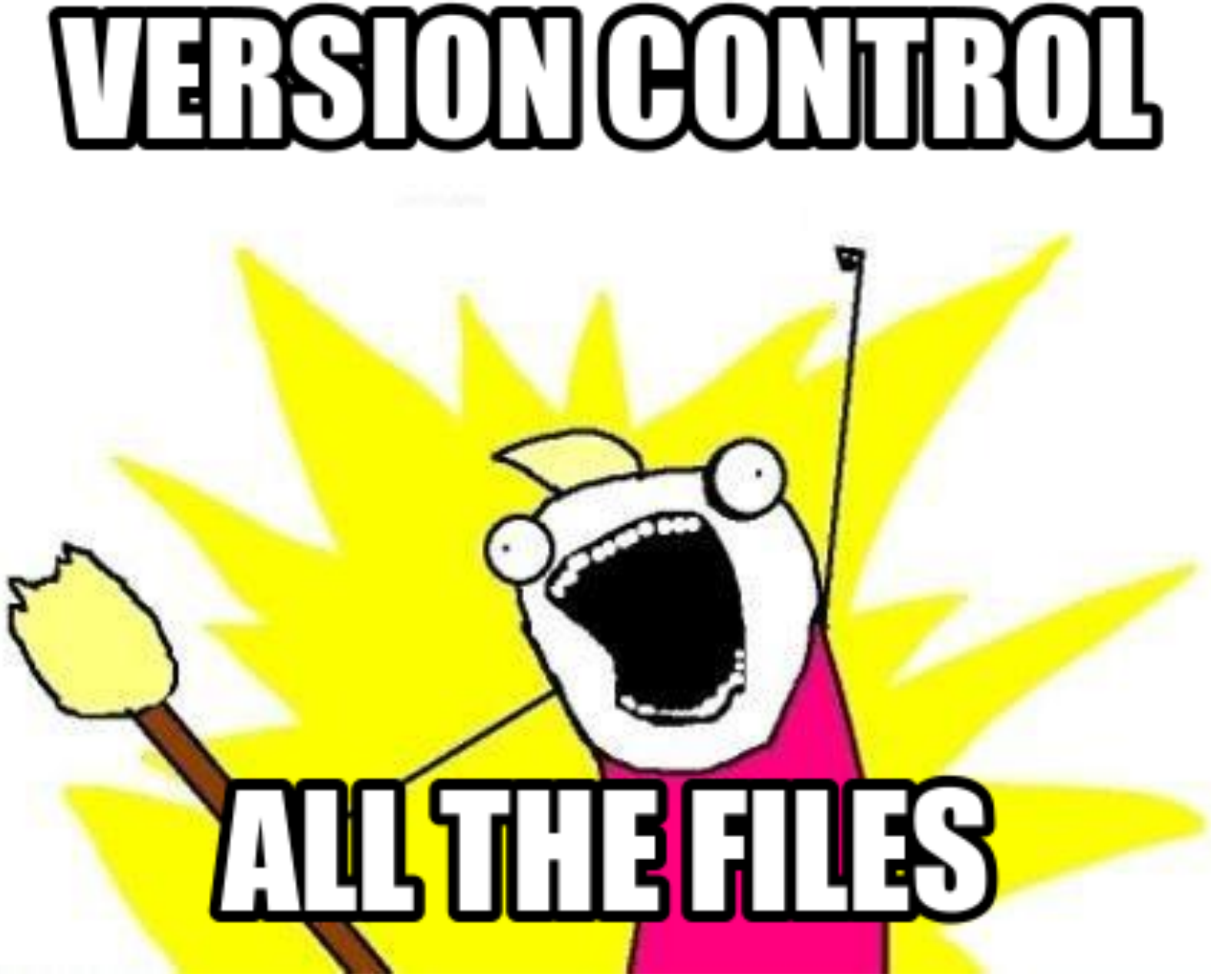

PYTHON!
YOU'RE FLYING! HOW?


I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"


I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?


I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.


I DON'T ALWAYS TEST MY CODE
BUT WHEN I DO I DO IT IN PRODUCTION
MEMECREATOR.NET


VERSION CONTROL
ALL THE FILES

What We Teach

- Unix shell
- Python
- Version control
- Testing
- Array computing, image processing, SQL, ...

“That's Merely Useful”

- None of this is publishable any longer
  - Which means it's ineffective career-wise for academic computer scientists

“That's Merely Useful”

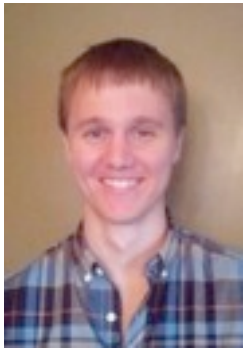
- None of this is publishable any longer
  - Which means it's ineffective career-wise for academic computer scientists
- But *it works*
  - Two independent assessments have validated the impact of what we do

Host a Workshop

Teach a Workshop

- All our materials are CC-BY licensed
- We will help train you





Shine Some Light

- Include a few lines about your software stack and working practices in your scientific papers
- Ask about them when reviewing
  - Where's the repository containing the code?
  - What's the coverage of your unit tests?
  - How did you track provenance?

Links!

ATPESC2013 on Facebook:
http://bit.ly/atpesc2013

Me:
http://aron.ahmadia.net

Continuum:
http://continuum.io

Software Carpentry:
http://software-carpentry.org

Performance Challenge:
https://github.com/ahmadia/atpesc-2013