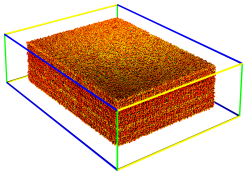
From Science to Open Source and Back Again

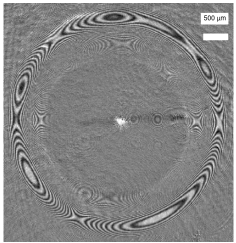
Thomas A Caswell

2022-11

Who is this guy talking to you?

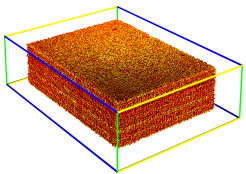
- Trained as a physicist
  - studied jammed amorphous systems + dynamics of Leidenfrost drops

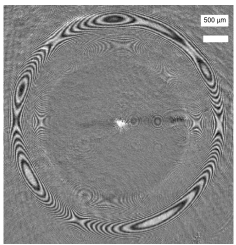




Who is this guy talking to you?

- Trained as a physicist
  - studied jammed amorphous systems + dynamics of Leidenfrost drops
- Currently developing data acquisition at NSLS-II
- Current Project Lead of Matplotlib
- PSF Fellow (Q2 2022)





matplotlib

bluesky

Acknowledgments

- Michael Droettboom, Eric Firing, and the Matplotlib team

Acknowledgments

- Michael Droettboom, Eric Firing, and the Matplotlib team
- Dan Allan, Ken Lauer

Acknowledgments

- Michael Droettboom, Eric Firing, and the Matplotlib team
- Dan Allan, Ken Lauer
- Dora Caswell

Software is core to the future of science

- Software is critical to (almost) all modern experimental science [1].

[1] https://doi.org/10.5281/zenodo.7295423

Software is core to the future of science

- Software is critical to (almost) all modern experimental science [1].
- ... lets you ask different questions.

[1] https://doi.org/10.5281/zenodo.7295423

Software is core to the future of science

- Software is critical to (almost) all modern experimental science [1].
- ... lets you ask different questions.
- ... encapsulates techniques in a re-usable and transferable way.

[1] https://doi.org/10.5281/zenodo.7295423

What is "Big Data" anyway?

- "Bigger than you are used to"

What is "Big Data" anyway?

- "Bigger than you are used to"
- Jean Baptiste Perrin did initial work on Brownian motion of particles in 1908
  - Traced colloidal motion by hand using *camera lucida*.
  - Measured order thousands of displacement vectors.

What is "Big Data" anyway?

- "Bigger than you are used to"
- Jean Baptiste Perrin did initial work on Brownian motion of particles in 1908
  - Traced colloidal motion by hand using *camera lucida*.
  - Measured order thousands of displacement vectors.
  - Physics Nobel Prize in 1926 (proved molecules exist).

What is "Big Data" anyway?

- "Bigger than you are used to"
- Jean Baptiste Perrin did initial work on Brownian motion of particles in 1908
  - Traced colloidal motion by hand using *camera lucida*.
  - Measured order thousands of displacement vectors.
  - Physics Nobel Prize in 1926 (proved molecules exist).
- I also tracked particles undergoing Brownian motion

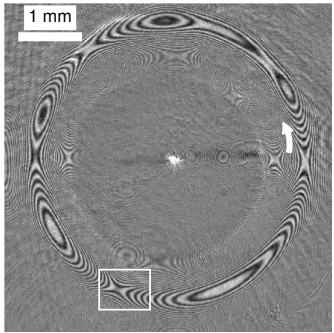
What is "Big Data" anyway?

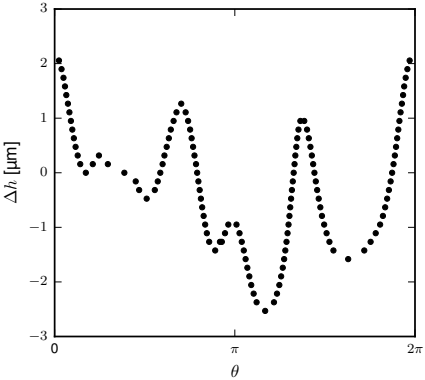
- "Bigger than you are used to"
- Jean Baptiste Perrin did initial work on Brownian motion of particles in 1908
  - Traced colloidal motion by hand using *camera lucida*.
  - Measured order thousands of displacement vectors.
  - Physics Nobel Prize in 1926 (proved molecules exist).
- I also tracked particles undergoing Brownian motion
  - ... did not get the Nobel prize for my work

What is "Big Data" anyway?

- "Bigger than you are used to"
- Jean Baptiste Perrin did initial work on Brownian motion of particles in 1908
  - Traced colloidal motion by hand using *camera lucida*.
  - Measured order thousands of displacement vectors.
  - Physics Nobel Prize in 1926 (proved molecules exist).
- I also tracked particles undergoing Brownian motion
  - ... did not get the Nobel prize for my work
  - 20,000 particles per frame over thousands of frames
  - could only (practically) be done with software

tracking fringes

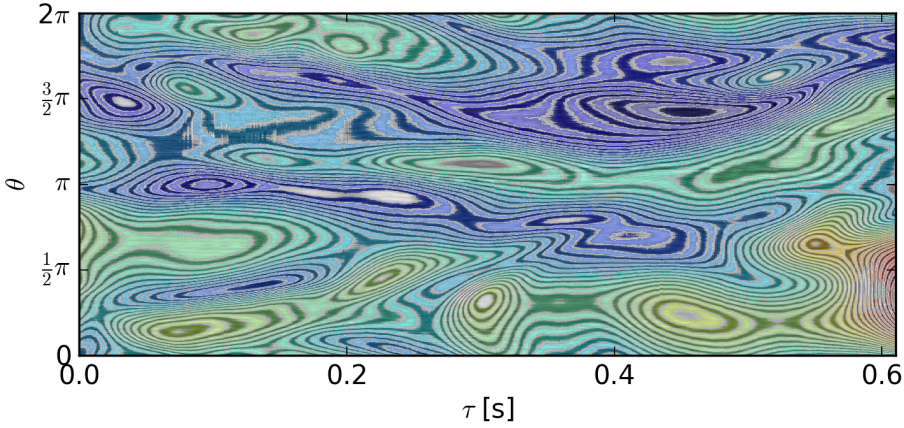




- Justin Burton manually counted fringes, got science out!
- "I bet you can automate that and get a time series of the profile" - Me

- 2011-11 got prototype working
- 2012-06 pulled `trackpy` out
- 2013-12 defended PhD & paper

tracking fringes



- 1TB of high-speed video to extract dynamics of the rim
- Could not have done this "by hand"

Software is a team sport

Oct 11, 2012, 7:55 PM
Subject: tracking.py

Hi Thomas,

I'm a physics grad at JHU working for Bob Leheny, who is himself a former student of Sidney Nagel. I came across your website a few years ago by way of Eric Weeks' page. Today I checked in on your site and found your pure Python implementation of track.pro.

Over the last few months I implemented the Crocker/Weeks particle location algorithm in pure Python (with a little C). I know you have done the same in C++. Currently, I'm still calling IDL to do the trajectory-linking step, but I am about to try switching to your tracking.py. I'm letting you know that I'm using it, as you request on your web page. Also, maybe we should consider contributing to each other's repositories.

Here's mine: https://github.com/danielballan/mr

Cheers, Dan

Software is a team sport

- This led to merging our code bases
  - About 30 people have now contributed to `trackpy`

Software is a team sport

- This led to merging our code bases
  - About 30 people have now contributed to `trackpy`
- Dan joined me at BNL and is my closest collaborator

Science and software are team activities

- Myth of "lone genius" is harmful in both science and software

Science and software are team activities

- Myth of "lone genius" is harmful in both science and software
- Science (as a whole) is a collaborative effort to understand how the world works.
  - the nominal reason we publish is to share knowledge

Science and software are team activities

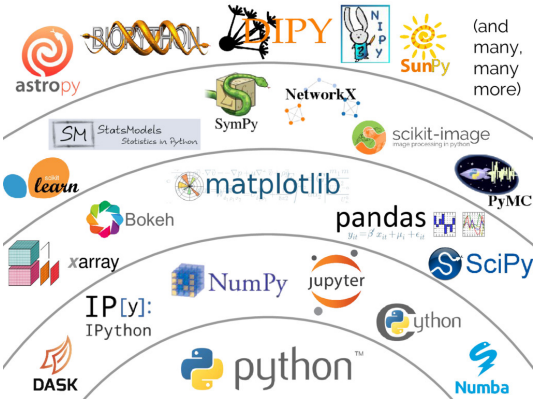
- Myth of "lone genius" is harmful in both science and software
- Science (as a whole) is a collaborative effort to understand how the world works.
  - the nominal reason we publish is to share knowledge
- Useful software is bigger than one person can physically write
  - particularly if you include the whole stack!

Science and software are team activities

- Myth of "lone genius" is harmful in both science and software
- Science (as a whole) is a collaborative effort to understand how the world works.
  - the nominal reason we publish is to share knowledge
- Useful software is bigger than one person can physically write
  - particularly if you include the whole stack!
- Every piece of software I've written together is better than any I wrote alone

"If I have seen further it is by standing on the shoulders of Giants" -Newton

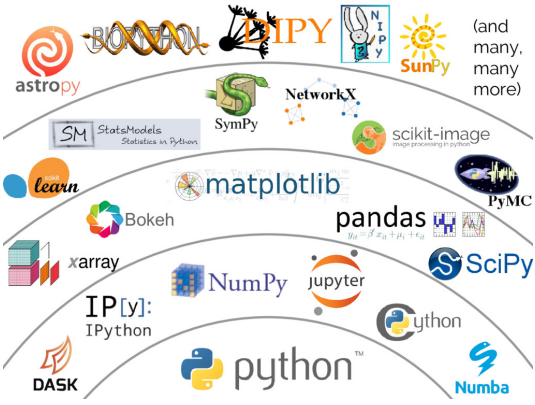
All of this was made possible by the Scientific Python ecosystem



Credit: Jake VanderPlas's SciPy 2015 Keynote

"If I have seen further it is by standing on the shoulders of Giants" -Newton

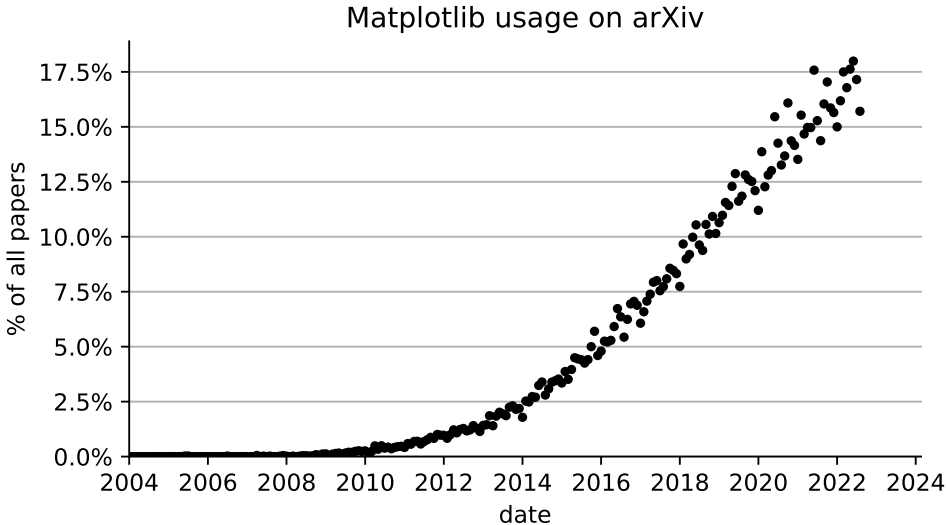
All of this was made possible by the Scientific Python ecosystem



Credit: Jake VanderPlas's SciPy 2015 Keynote

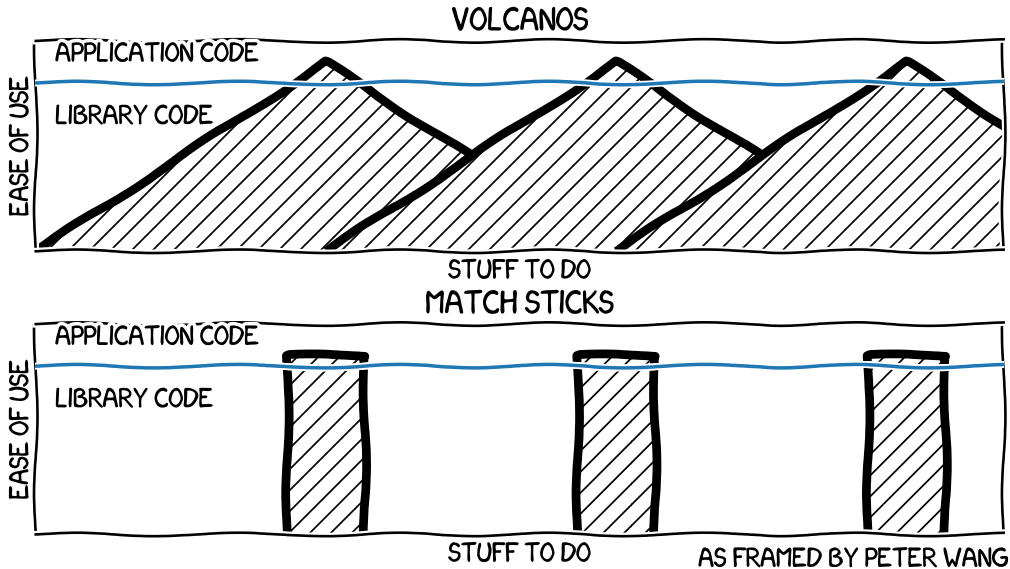
- trackpy has been cited 296 times
- Leidenfrost paper cited 40 times (Physical Review E 90 (1), 013014)

Matplotlib's impact

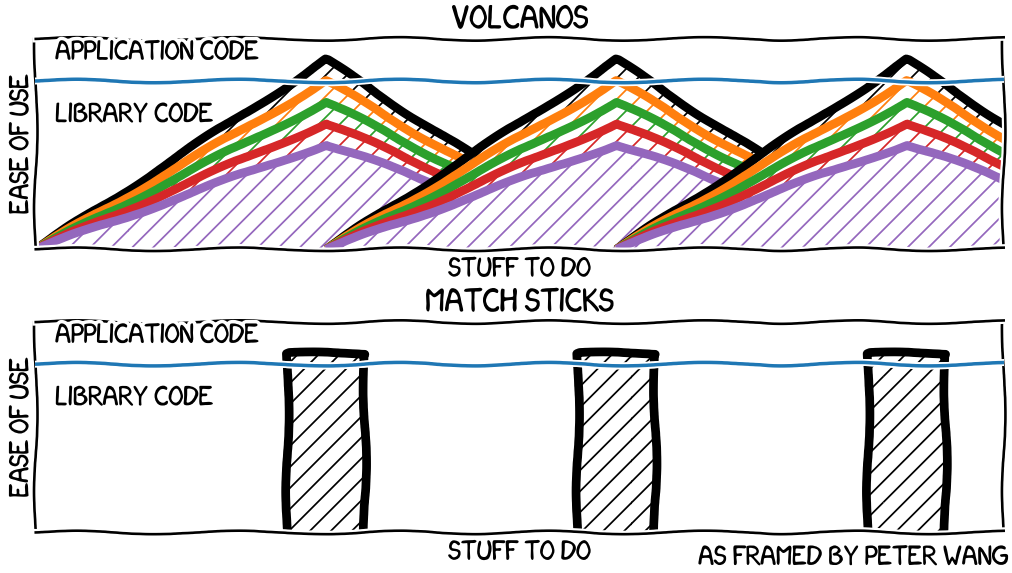


https://www.coiled.io/blog/how-popular-is-matplotlib

Build volcanos not match sticks



Build volcanos not match sticks



Application vs Library

Application-like

Library-like

Application vs Library

Application-like

- solves a particular problem for me *right now*

Library-like

- solves a class of problems for someone else at some other time

Application vs Library

Application-like

- solves a particular problem for me *right now*
- used by humans

Library-like

- solves a class of problems for someone else at some other time
- used by other code

Application vs Library

Application-like

- solves a particular problem for me *right now*
- used by humans
- many assumptions

Library-like

- solves a class of problems for someone else at some other time
- used by other code
- few assumptions

Application vs Library

Application-like

- solves a particular problem for me *right now*
- used by humans
- many assumptions
- opinionated about I/O

Library-like

- solves a class of problems for someone else at some other time
- used by other code
- few assumptions
- agnostic (as possible) to I/O

Application vs Library

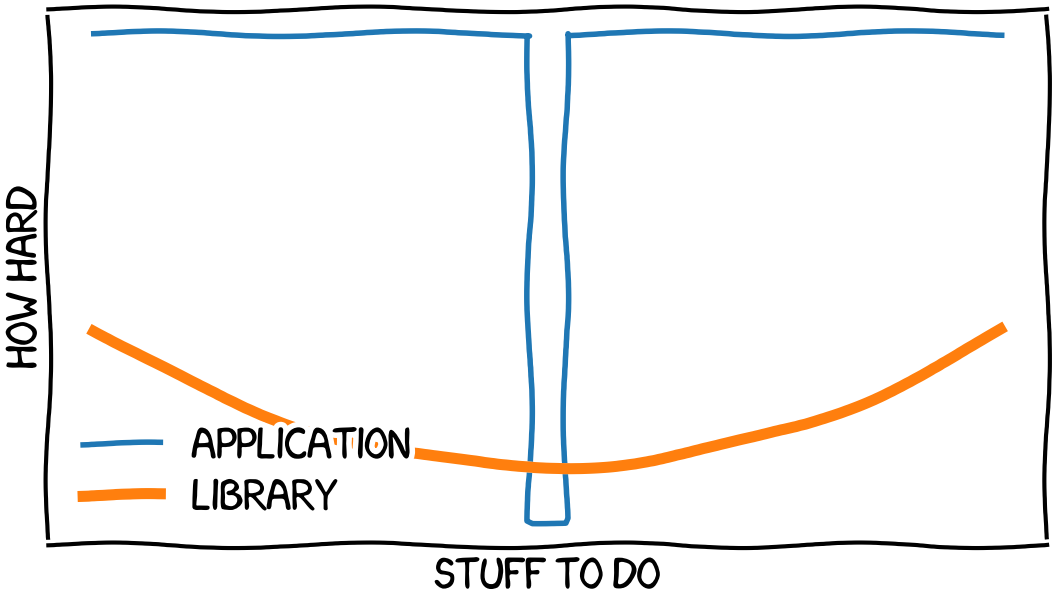
Application-like

- solves a particular problem for me *right now*
- used by humans
- many assumptions
- opinionated about I/O
- never fail

Library-like

- solves a class of problems for someone else at some other time
- used by other code
- few assumptions
- agnostic (as possible) to I/O
- fail hard and fast

Application vs Library



Case Study

Directory of images named like `./{sample_name}/x_y_{T}C_z.tiff`

Case Study

Directory of images named like `./{sample_name}/x_y_{T}C_z.tiff`

Application-like

```
g = Path(".").glob("**/*.tiff")
out = {}
for f in g:
    data = tifffile.imread(f)
    sample = f.parents.parts[-1]
    fn = f.name
    T = int(fn.split("_")[2][:-1])
    out[sample] = data.sum() / T
```

Case Study

Directory of images named like `./{sample_name}/x_y_{T}C_z.tiff`

Library-like

```
def load_data(fn):
    data = tifffile.imread(fn)
    sample = f.parents.parts[-1]
    T = f.name.split("_")[2][:-1]
    return data, sample, int(T)

g = Path(".").glob("**/*.tiff")
gen = map(load_data, g)
out = {
    sample: data.sum() / T
    for data, sample, T in gen
}
```

Case Study

Directory of images named like `./{sample_name}/x_y_{T}C_z.tiff`

Application-like

```
g = Path(".").glob("**/*.tiff")
out = {}
for f in g:
    data = tifffile.imread(f)
    sample = f.parents.parts[-1]
    fn = f.name
    T = int(fn.split("_")[2][:-1])
    out[sample] = data.sum() / T
```

Library-like

```
def load_data(fn):
    data = tifffile.imread(fn)
    sample = f.parents.parts[-1]
    T = f.name.split("_")[2][:-1]
    return data, sample, int(T)

g = Path(".").glob("**/*.tiff")
gen = map(load_data, g)
out = {
    sample: data.sum() / T
    for data, sample, T in gen
}
```

Library mindset

- What assumptions am I making?

Library mindset

- What assumptions am I making?
- If I make this choice, what won't I be able to do in the future?

Library mindset

- What assumptions am I making?
- If I make this choice, what won't I be able to do in the future?
- What is the next (next) thing I'll want to do? Does this have a path to that?

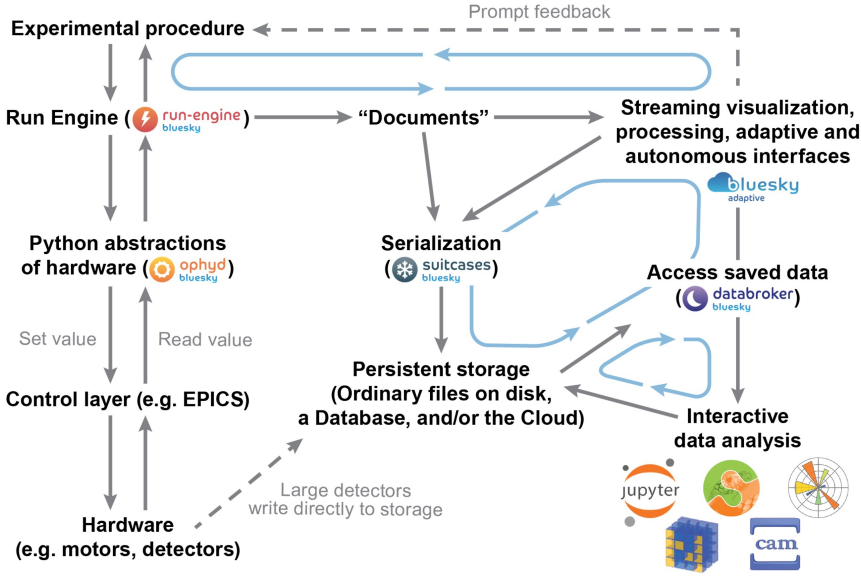
Library mindset

- What assumptions am I making?
- If I make this choice, what won't I be able to do in the future?
- What is the next (next) thing I'll want to do? Does this have a path to that?
- What information does each part of the program need to know?

Library mindset

- What assumptions am I making?
- If I make this choice, what won't I be able to do in the future?
- What is the next (next) thing I'll want to do? Does this have a path to that?
- What information does each part of the program need to know?
- If I make this change who do I break?

Dawn of Bluesky



Dawn of Bluesky

Impact and adoption

- Running NSLS-II user operations
  - (1.3k+ researchers|550+ papers)/yr

If you have experimental apparatus you want to automate talk to me!

tcaswell@bnl.gov

Dawn of Bluesky

Impact and adoption

- Running NSLS-II user operations
  - (1.3k+ researchers|550+ papers)/yr
- also used by...
  - other DOE light sources
  - international synchrotrons
  - university labs
  - individual researchers

If you have experimental apparatus you want to automate talk to me!

tcaswell@bnl.gov

Dawn of Bluesky

Impact and adoption

- Running NSLS-II user operations
  - (1.3k+ researchers|550+ papers)/yr
- also used by...
  - other DOE light sources
  - international synchrotrons
  - university labs
  - individual researchers

Why did this work?

- co-designed but independent libraries
- Developed in the open from the start
  - anyone can see the code
  - anyone can see and join discussions
  - anyone can influence the code
- invite users to become collaborators

If you have experimental apparatus you want to automate talk to me!

tcaswell@bnl.gov

What do you need to "maintain" an open source project?

You are a maintainer if you feel responsible for the project.

What do you need to "maintain" an open source project?

You are a maintainer if you feel responsible for the project.

- A stewardship (not ownership) mindset
  - the project has value because it solves (other) people's problems
  - project may have existed before you and you want it to exist after you

What do you need to "maintain" an open source project?

You are a maintainer if you feel responsible for the project.

- A stewardship (not ownership) mindset
  - the project has value because it solves (other) people's problems
  - project may have existed before you and you want it to exist after you
- Empathy
  - everyone is coming from a different place
  - frequently getting users who are having a (very) bad day

What do you need to "maintain" an open source project?

You are a maintainer if you feel responsible for the project.

- A stewardship (not ownership) mindset
  - the project has value because it solves (other) people's problems
  - project may have existed before you and you want it to exist after you
- Empathy
  - everyone is coming from a different place
  - frequently getting users who are having a (very) bad day
- Community Management and people skills
  - Software is a team sport
  - See Melissa Weber Mendonça's keynote tomorrow!

What do you need to "maintain" an open source project?

You are a maintainer if you feel responsible for the project.

- A stewardship (not ownership) mindset
  - the project has value because it solves (other) people's problems
  - project may have existed before you and you want it to exist after you
- Empathy
  - everyone is coming from a different place
  - frequently getting users who are having a (very) bad day
- Community Management and people skills
  - Software is a team sport
  - See Melissa Weber Mendonça's keynote tomorrow!
- Technical skills
  - "the easy part"

..and you still do science?!

..and you still do science?!

- no

..and you still do science?!

- no
  - . . . but have enabled a lot of science and helped a lot of scientists
- Scientists do not have time to be domain experts AND professional developers

..and you still do science?!

- no
  - . . . but have enabled a lot of science and helped a lot of scientists
- Scientists do not have time to be domain experts AND professional developers
- However basic programming is a required professional skill
  - like math, paper writing, and grant writing

..and you still do science?!

- no
  - . . . but have enabled a lot of science and helped a lot of scientists
- Scientists do not have time to be domain experts AND professional developers
- However basic programming is a required professional skill
  - like math, paper writing, and grant writing

Enter the Research Software Engineer (RSE)

"We like an inclusive definition of Research Software Engineers to encompass those who regularly use expertise in programming to advance research. This includes researchers who spend a significant amount of time programming, full-time software engineers writing code to solve research problems, and those somewhere in-between"

https://us-rse.org/about/what-is-an-rse/

Research Software Engineer (RSE)

What a RSE role is not?

- Computer Science research

https://us-rse.org | https://us-rse.org/jobs/

Research Software Engineer (RSE)

What a RSE role is not?

- Computer Science research
- "1-800-fix-mycode"

https://us-rse.org | https://us-rse.org/jobs/

Research Software Engineer (RSE)

What a RSE role is not?

- Computer Science research
- "1-800-fix-mycode"

What is a good RSE role?

- Communication is key
  - what assumptions are safe to make?
  - where are the likely extension points?
  - What are the right trade offs?

https://us-rse.org | https://us-rse.org/jobs/

Research Software Engineer (RSE)

What a RSE role is not?

- Computer Science research
- "1-800-fix-mycode"

What is a good RSE role?

- Communication is key
  - what assumptions are safe to make?
  - where are the likely extension points?
  - What are the right trade offs?
- Partnership and Collaboration
  - RSE conversant in domain
  - Domain scientist conversant in SW

https://us-rse.org | https://us-rse.org/jobs/

Advice and Conclusion

- Use version control

Advice and Conclusion

- Use version control
- No really, use version control

Advice and Conclusion

- Use version control
- No really, use version control
- Software is essential to doing science

Advice and Conclusion

- Use version control
- No really, use version control
- Software is essential to doing science
- Programming is a required professional skill for working scientists

Advice and Conclusion

- Use version control
- No really, use version control
- Software is essential to doing science
- Programming is a required professional skill for working scientists
- Have a library mentality from the start

Advice and Conclusion

- Use version control
- No really, use version control
- Software is essential to doing science
- Programming is a required professional skill for working scientists
- Have a library mentality from the start
- Work in the open from the start

Advice and Conclusion

- Use version control
- No really, use version control
- Software is essential to doing science
- Programming is a required professional skill for working scientists
- Have a library mentality from the start
- Work in the open from the start
- Make friends! Build collaborations! Have fun!

Links

- US-RSE home page: https://us-rse.org
- CSCCE home page: https://www.cscce.org
- Bluesky project homepage: https://blueskyproject.io
- My blog: https://tacaswell.github.io
- Use VC: https://tacaswell.github.io/how-i-learned-to-love-vcs.html
- mpl on arXiv source: https://github.com/mrocklin/arxiv-matplotlib
- Daniel Katz talk: https://doi.org/10.5281/zenodo.7295423
- These slides: https://github.com/tacaswell/talk_2022-11_pydataNYC

tcaswell@gmail.com | tcaswell@bnl.gov | @tacaswell@fosstodon.org

backup buffer

How did I start to work on Matplotlib?

- opened a PR to solve a problem I had (10yrs ago in August)
  - https://github.com/matplotlib/matplotlib/pull/1062
- started answering questions on StackOverflow about Matplotlib while procrastinating my research (sorry Sid)
- eventually started reporting & fixing bugs upstream
- was offered commit rights and asked to start reviewing other PRs
- was offered co-lead in 2014
- was offered lead in 2016