why sustain scientific software?
Two Caveats
What is this?

#1
What is this?

#1
Straw person

#1
Original research not included.
Where are we?
WSSSSPE1.1

(SciPy 14)
WSSSPE2

(SC14)
WSSSPE2.1

(SciPy 2015)
WSSSPE3

(here[^1])

[^1]: I am extremely happy we are not at Supercomputing this year.
“The International Conference for High-Performance Computing, Networking, Storage and Analysis”

“Scientific Computing with Python”
Software is getting harder.  Software is getting better.
(pause for self-reflection)
why sustain scientific software?
sustain?
scientific software?
why?
sustain?
(scientific software)
(scientific software)
prizes?

scientific software
Sustainable?

Keeps up with bug reports.
Sustainable?

Adds new features.
Sustainable?

Works on new hardware architectures.
Sustainable?

People can learn how to use it.
Sustainable?

Keeps getting funded.
Sustainable?

Grows new features.
Sustainable?

Produces the same results.
Sustainable?

Transitions between people.
Models

Funded

Productized

Volunteer
Is “productized” synonymous with self-sufficient?
Top-down  Bottom-up
Large-scale?  

Small-scale?
Large-scale?  \[\quad\]  Small-scale?

well, no
scientific software?
What are we sustaining?
Scientific software, obviously.
Scientific software, obviously.

Scientific inquiry.
Scientific software, obviously.

Scientific inquiry.

Conducting inquiry via software.
Scientific software, obviously.

Scientific inquiry.

Conducting inquiry via software.

Specific software?
Scientific software, obviously.

Scientific inquiry.

Conducting inquiry via software.

Specific software?

People? Career? Future?
Top-down  Bottom-up
why?
Motivation: a series of ansatz[^3]

"The Noble Scientist"
Fun?
Useful?
Craft?
Identity?
One person[^4] asked not, "How can we sustain?" but instead asked, "When do we end?"

[^4]: That person was Greg Wilson.
Without answering that, how do we know how to proceed?
Wishlist
Wishlist

How do we “ship” a “product” “on time” when dealing with a mix of funding models and motivations?
Wishlist

How can we maintain a “process” that doesn’t turn people away?
Wishlist

How do we know when it’s time to move on?
Wishlist

How can “productized” software balance its future versus its past?
Wishlist

How can we help avoid burnout?
Wishlist

Can we reduce systemic bias?
thank you