

# Community-Building in Open Source Scientific Software

Nathan Goldbaum  
@njgoldbaum

ATPESC  
August 9, 2016

# A little about myself

- PhD in Astronomy & Astrophysics from UCSC, 2015, mostly on simulations of idealized isolated Milky Way analogues
- Currently postdoc at NCSA in the Data Exploration Lab ([dxl.ncsa.illinois.edu](http://dxl.ncsa.illinois.edu))
- Core developer of yt ([yt-project.org](http://yt-project.org)) - an analysis and visualization framework for simulation data
- Contributor to Enzo ([enzo-project.org](http://enzo-project.org)) - an open source AMR cosmological hydrodynamics code
- Small contributions to many other projects (sphinx, matplotlib, IPython, mercurial, homebrew)

# Outline

- Choosing a license
- Hosting and releasing code
- Making code friendly to newcomers
- Reaching out: building a community of practice

# Why do you need to license your code?

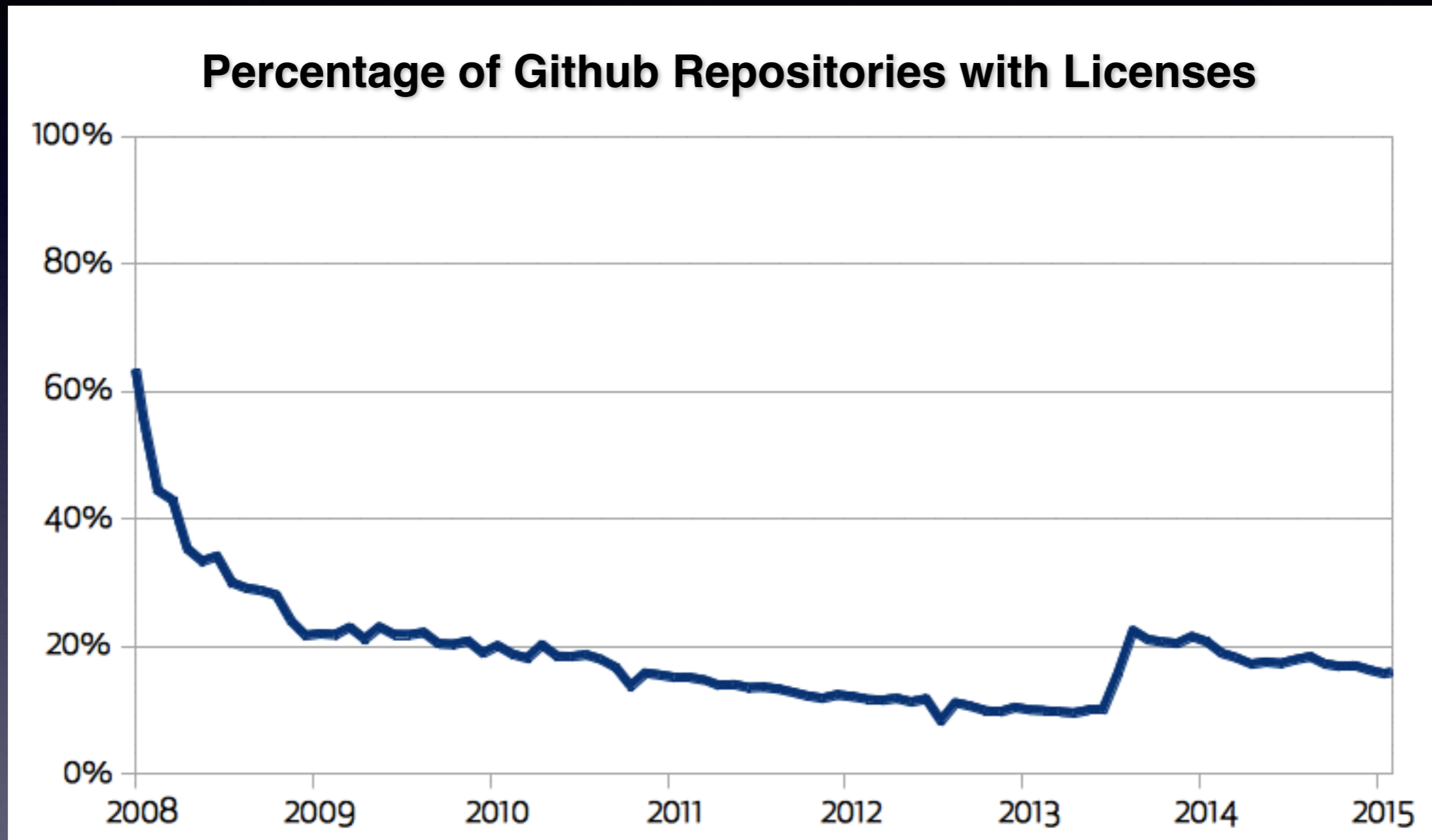
“Generally speaking, the absence of a license means that default copyright laws apply.”

“This means that you retain all rights to your source code and that nobody else may reproduce, distribute, or create derivative works from your work.”



The license is the social contract between a code's developers and users

# Sadly, most people don't choose a license



<https://github.com/blog/1964-open-source-license-usage-on-github-com>

# How to license software?

- Easy! Create a LICENSE or COPYING file and drop it into the root of the source distribution
  - <https://github.com/BoxLib-Codes/Castro>
  - <https://bitbucket.org/enzo/enzo-dev/src>
  - <https://github.com/pencil-code/pencil-code>

# What not to do?

- Make up your own license
- Use a license that has not been certified as a free software license
- Make up terms to add to an existing license

**Downloading** [REDACTED]

## **Licence**

You are allowed to use [REDACTED] and the other software provided on this page free of charge on condition that:

- You acknowledge use of the software in publications
- You consider the proposed references in the 'README' file for citation
- If you identify any bugs, please report them to [REDACTED]

**If you agree to the licence please fill in the form below and press "Submit".**



# Free software != Open Source



Free as in beer



Free as in freedom

# Free software != Open Source

- The freedom to run the program as you wish, for any purpose.
- The freedom to study how the program works, and change it so it does your computing as you wish.
- The freedom to redistribute copies so you can help your neighbor.
- The freedom to distribute copies of your modified versions to others. By doing this you can give the whole community a chance to benefit from your changes.



# Two types of free software licenses

Permissive



Copyleft



# Two types of free software licenses

## Permissive

Add reference to original license for reused code

No further restrictions on reuse

## Copyleft

If you release something that uses copyleft code, your code must be publicly available and must be licensed under a copyleft license

GPL: If your code links against a GPL library, it must be released under the GPL



# The holy war



**Jake VanderPlas**  
@jakevdp



Following

I was going to do some real work this morning, but there was a GPL vs. BSD email flame-war to attend to...

RETWEETS

3

LIKES

11



10:50 AM - 8 Sep 2014



# Licenses used by open source projects

Permissive

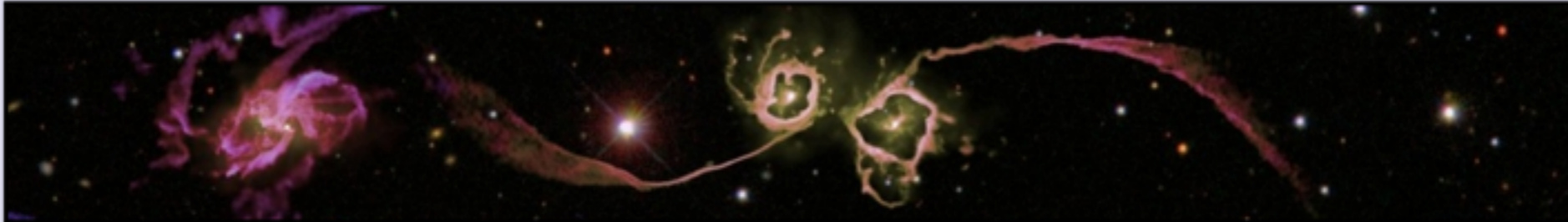
Copyleft



Building a community is  
more than choosing a  
license



# A Cautionary Tale



## GADGET - 2

A code for cosmological simulations of structure formation

<b>General</b> <ul style="list-style-type: none"><li><a href="#">Description</a></li><li><a href="#">Features</a></li><li><a href="#">Authors and History</a></li><li><a href="#">Acknowledgments</a></li><li><a href="#">News</a></li></ul>	<b>Description</b> <p><b>GADGET</b> is a freely available code for cosmological N-body/SPH simulations on massively parallel computers with distributed memory. <b>GADGET</b> uses an explicit communication model that is implemented with the standardized MPI communication interface. The code can be run on essentially all supercomputer systems presently in use, including clusters of workstations or individual PCs.</p> <p><b>GADGET</b> computes gravitational forces with a hierarchical tree algorithm (optionally in combination with a particle-mesh scheme for long-range gravitational forces) and represents fluids by means of smoothed particle hydrodynamics (SPH). The code can be used for studies of isolated systems, or for simulations that include the cosmological expansion of space, both with or without periodic boundary conditions. In all these types of simulations, <b>GADGET</b> follows the evolution of a self-gravitating collisionless N-body system, and allows gas dynamics to be optionally included. Both the force computation and the time stepping of <b>GADGET</b> are fully adaptive, with a dynamic range which is, in principle, unlimited.</p> <p><b>GADGET</b> can therefore be used to address a wide array of astrophysically interesting problems, ranging from colliding and merging galaxies, to the formation of large-scale structure in the Universe. With the inclusion of additional physical processes such as radiative cooling and heating, <b>GADGET</b> can also be used to study the dynamics of the gaseous intergalactic medium, or to address star formation and its regulation by feedback processes.</p>
<b>Software</b> <ul style="list-style-type: none"><li><a href="#">Download GADGET</a></li><li><a href="#">Download N-GenIC</a></li><li><a href="#">Requirements</a></li><li><a href="#">License</a></li><li><a href="#">Mailing List</a></li><li><a href="#">Change-Log</a></li><li><a href="#">Examples</a></li></ul>	<b>Features</b> <ul style="list-style-type: none"><li>▶ Hierarchical multipole expansion (based on a geometrical oct-tree) for gravitational forces.</li><li>▶ Optional TreePM method, where the tree is used for short-range gravitational forces only while long-range forces are computed with a FFT-based particle-mesh (PM) scheme. A second PM layer can be placed on a high-resolution region in 'zoom'-simulations.</li></ul>
<b>Documentation</b> <ul style="list-style-type: none"><li><a href="#">Code Paper</a></li><li><a href="#">Users Guide</a></li><li><a href="#">Code Reference</a></li></ul>	
<b>Publications</b> <ul style="list-style-type: none"><li><a href="#">Scientific Papers</a></li><li><a href="#">Pictures</a></li><li><a href="#">Movies</a></li><li><a href="#">Links</a></li><li><a href="#">Contact Address</a></li></ul>	



# A Cautionary Tale

## Download GADGET

You may download the **GADGET-2** code as a compressed tar-file:

[gadget-2.0.7.tar.gz](#) (~21.5 MB)

Please use the commands ``gunzip gadget-2.0.7.tar.gz`` and ``tar -xvf gadget-2.0.7.tar`` to unpack the files. You will obtain a directory ``Gadget-2.0.7/``, and various subdirectories containing the actual source code, the code documentation, as well as a number of simulation examples and very basic analysis scripts. Please refer to the `README` file, and **GADGET**'s User's Guide for further directions about installation and usage. A brief guide to parameters in the Makefile and the parameterfile, as well as a cross-referenced source code documentation is accessible with a web-browser in the ``html/``-subdirectory (open the ``index.html`` file). Note that the large size of the download is caused by the included example initial conditions.

“The second public version (GADGET-2, released in May 2005), contains most of these improvements, **except the numerous physics modules developed for the code that go beyond gravity and ordinary gas-dynamics.**”

# Where Gadget Went Wrong

- Extremely Popular (2800 citations!)
- Many variants, some public, most private
- Overworked maintainer / single point of failure
- Further improvements (Gadget3, Arepo) do not get released for general use
- Private physics routines preclude reproducibility despite the bulk of the code being public
- Little interoperability between versions used by different research groups, each developed separately

# Best Practices for Hosting and Releasing Code



# Use version control

- Use distributed version control in a **public** repository.
- git or hg, \*not\* centralized version control (SVN, CVS) or a folder somewhere ('my-code.v2.v3.08-2016/')
- Distributed version control makes it possible for a newcomer to perform full development workflow without your explicit permission (or knowledge!)
- Having history available makes it much easier to track down and bisect bugs, see how code evolved



































# Social Coding

The Enzo Project / Untitled project / enzo-dev

## Pull requests

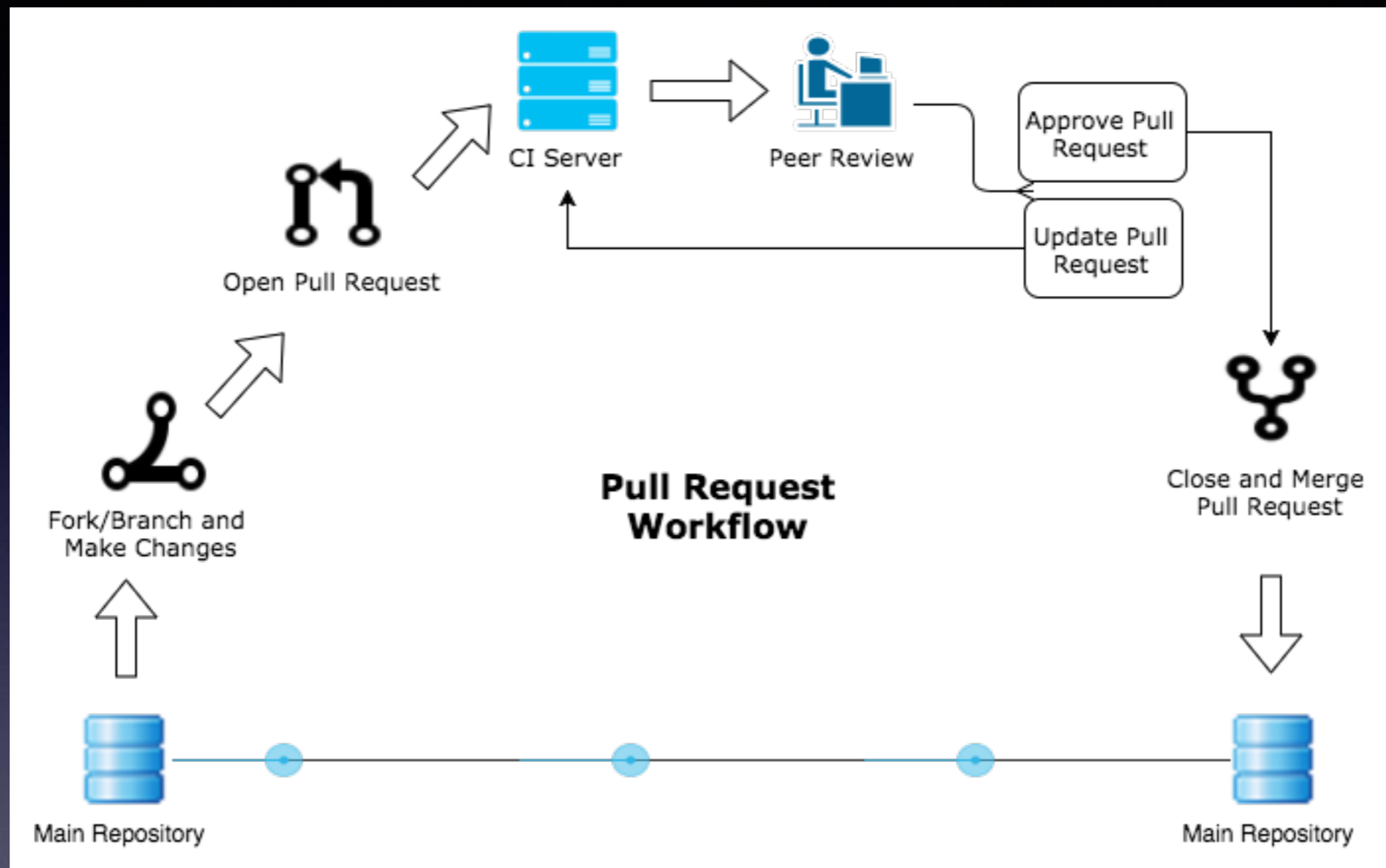
Create pull request

FILTER BY: Merged Author Target branch I'm reviewing

Summary	Reviewers	Builds
 <b>Removing unused parameters.</b> → week-of-code Britton Smith - #342, last updated on 22 Jul 2016		
 <b>Fixed conduction timestep calculation.</b> → week-of-code Duncan Christie - #341, last updated on 22 Jul 2016	 	
 <b>Updating the test suite to work with yt-3</b> → week-of-code Nathan Goldbaum - #304, last updated on 08 Jul 2016	   	
 <b>Make all test problems use a power-of-two refinement factor</b> → week-of-code Nathan Goldbaum - #340, last updated on 30 Jun 2016	 	
 <b>Make verbose test output print details about which tests are b...</b> → week-of-code Nathan Goldbaum - #339, last updated on 28 Jun 2016		
 <b>GalaxySimulation bug fix when using DiskGravity</b> → week-of-code Andrew Emerick - #337, last updated on 21 Jun 2016	  	
 <b>Particle Support for DiskGravity</b> → week-of-code Andrew Emerick - #336, last updated on 21 Jun 2016	  	
 <b>Bug fix in rarefaction fan in flux_twoshock.F</b> → week-of-code Greg Bryan - #338, last updated on 16 Jun 2016	  	
 <b>Redshift for Grackle UVB in non-cosmological simulations</b> → week-of-code Andrew Emerick - #334, last updated on 15 Jun 2016	   	

<https://bitbucket.org/enzo/enzo-dev/pull-requests>

# Social Coding



Fork the repository, write code, commit it, run tests  
contribute back to original repository, repeat

No special permissions! Anyone can follow this workflow!

# Code Review

- One or more people should look over each contribution in detail and signal approval or offer comments for further revision. This is an iterative process.
- Encourage smaller, more easily reviewable contributions. Major development efforts should happen over many pull requests.
- Advanced: for major development efforts, consider rewriting history to ease code review. Each commit should change only one thing. This is \*much\* easier to review.
- CONTRIBUTING file spells out style, code review expectations.



# Continuous Integration

- Tests are only useful if they're being run
- Every pull request should pass all the tests
- Contributions and bugfixes should add new tests to prevent regressions



Travis CI



AppVeyor



circleci



**Jenkins**

# Do regular releases

- Make regular stable releases including both sources and binaries (if applicable)
  - Compiling code can be a big barrier to entry
- Stable releases are a quantum of accomplishment you can point to
- Accounce via e-mail, facebook, twitter, whichever other communication mechanism you prefer.
  - Getting contributors to announce releases is a good way to build community, recognize contributions





# CONDA- FORGE

A community led collection of recipes, build infrastructure and distributions for the conda package manager.

[conda-forge.github.io](https://conda-forge.github.io)



# Make your code friendly for newcomers

- Documentation.
  - API Docs (automatically generated from docstrings)
  - Narrative docs
  - Worked examples
  - Cookbook / Gallery

# Building and hosting your documentation

- Build examples in docs as part of your test suite
- Store docs in the same repository as the code, so docs changes come in as part of the same pull request as code changes



**Read the Docs**

Create, host, and browse documentation.

# Traditional Software Communities



Developers

Users



# Communities of Practice



The code should not be a black box!

“Scaling a Code in the Human Direction”, Turk (2013)

<http://arxiv.org/abs/1301.7064>

# Code of Conduct

- Community agrees to participate under binding code of conduct.
- Treat each other with respect, professional demeanor, and give others the benefit of the doubt.
- Abuse and violations may lead to time-outs, community bans, or other sanctions.

<http://contributor-covenant.org/>

<https://www.python.org/psf/codeofconduct/>


# Recognize, Praise, and Promote your Community

## Gallery of Examples

Here are some examples of yt in the real world.

We welcome you to submit your own images made with yt from publications, talks, webpages, etc. Just fork our [repository](#) and issue a pull request with your image at the top of the page. Images should be about 400 pixels wide, and please include a link to any published work.

Volumetric rendering of human CT Scan via yt



### NeuroDome

yt has been used to create volumetric renderings of human CT scans in support of the [NeuroDome](#) project. For more information, check out the NeuroDome website.

<http://yt-project.org/gallery.html>

<http://yt-project.org/community.html>

<http://yt-project.org/members.html>