

Publishing software through Github

Ioannis Tsanaktsidis, CERN
Xiaoli Chen, CERN

Madrid, November 2016



Agenda

1. Introduction to software sharing
2. Using Github for collaborative software development
3. Archiving Github repositories with Zenodo
4. Q&A





Why share software?

- Share ideas and skills, avoid reinventing the wheel
- Improve software quality, invite community input
- Support reproducible science, bridge the gap between input and output
- Create sustainable open-source community





Why share software?

- First class citizen
 - Scientific research softwares are increasingly being recognized as legitimate research output, should be credited alongside articles and data.
- Call for good practice
 - Most of the time softwares are not openly archived, nor easily discoverable -- barriers to sharing and reuse.
- Emerging policies, recommendations and services



1. Software citation principles

The main contribution of this document are the software citation principles, written fairly concisely in this section and discussed further later in the document (§6). In addition, we also motivate the creation of these principles (§2), describe the process by which they were created (§3), summarize use cases related to software citation (§4), and review related work (§5). We also lay out the work needed to lead to these software citation principles being applied (§7).

- Importance:** Software should be considered a legitimate and citable product of research. Software citations should be accorded the same importance in the scholarly record as citations of other research products, such as publications and data; they should be included in the metadata of the citing work, for example in the reference list of a journal article, and should not be omitted or separated. Software should be cited on the same basis as any other research product such as a paper or a book, that is, authors should cite the appropriate set of software products just as they cite the appropriate set of papers.
- Credit and Attribution:** Software citations should facilitate giving scholarly credit and normative, legal attribution to all contributors to the software, recognizing that a single style or mechanism of attribution may not be applicable to all software.
- Unique Identification:** A software citation should include a method for identification that is machine actionable, globally unique, interoperable, and recognized by at least a community of the corresponding domain experts, and preferably by general public researchers.
- Persistence:** Unique identifiers and metadata describing the software and its disposition should persist – even beyond the lifespan of the software they describe.
- Accessibility:** Software citations should facilitate access to the software itself and to its associated metadata, documentation, data, and other materials necessary for both humans and machines to make informed use of the referenced software.
- Specificity:** Software citations should facilitate identification of, and access to, the specific version of software that was used. Software identification should be as specific as necessary, such as using version numbers, revision numbers, or variants such as platforms.



The Journal of Open Source Software

A **developer friendly** journal for research software packages.

Learn more »

All papers (75)

🔗 <https://sourceforge.net/p/kactus2/code/HEAD/tree/trunk/> JOSS Submitted

Kactus2 is a graphical EDA tool targeted for packing and assembling building blocks for ASICs and FPGAs, based on the IP-XACT standard. It supports Verilog, VHDL, and other formats.

DOI pending

🔗 [warrenlr / RAILS](#) JOSS Under Review

Work with Github





What is GitHub?

- GitHub is a code hosting platform for version control and collaboration.
- It lets you and others work together on projects from anywhere.
- No coding necessary.





Basic components

- Repositories
- Branches
- Commits
- Pull requests





Repositories

- Usually used to organize a single project.
- Can contain folders and files, images, videos, spreadsheets, and data sets.
- Anything that your project needs.





Creating a repository

Owner **Repository name**

PUBLIC   **hubot** / 

Great repository names are short and memorable. Need inspiration? How about **petulant-shame**.

Description (optional)

☒  **Public**
Anyone can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: **None** ▼

Add a license: **None** ▼



Create repository



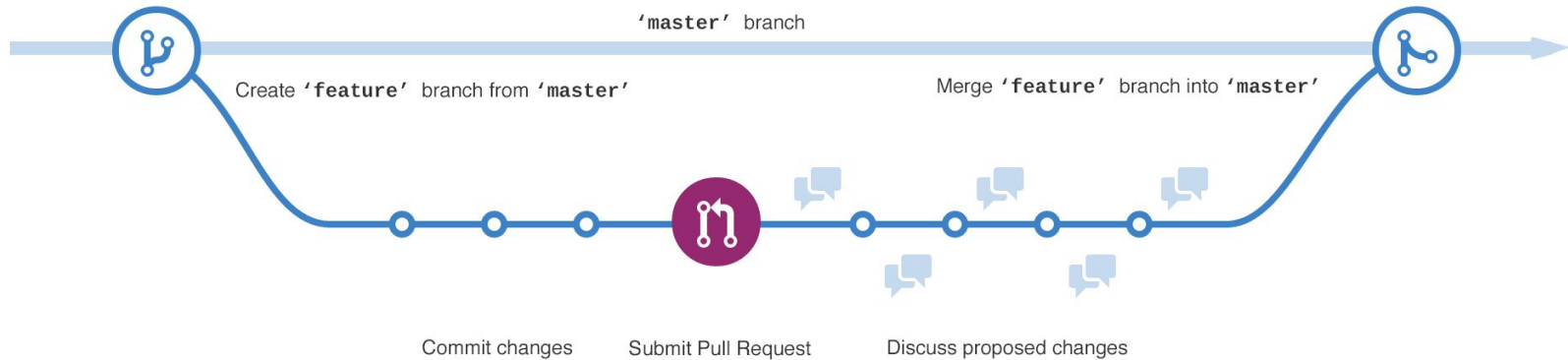


Branches

- Branching is the way to work on different versions of a repository at one time.
- By default the repository has one branch called master which is considered to be the definitive branch.
- Use of branches to experiment and make edits before committing them to master.
- By creating a branch off the master you are making a snapshot of master as it was at that point in time.



Lifetime of branch





Commits

- On GitHub, saved changes are called commits.
- Each commit has an associated commit message.
- Description explaining why a particular change was made.
- Commit messages capture the history of your changes, so other contributors can understand what you've done and why.



Making a commit



hubot / hello-world

Unwatch 1Star 0Fork 0

<> CodeIssues 0Pull requests 0WikiPulseGraphsSettings

hello-world / README.md or cancel

<> Edit filePreview changesSpaces2Soft wrap

1 # hello-world
2
3 Hi Humans!
4
5 Hubot here, I like Node.js and Coffeescript (that's what I'm made of!).
6 I've had tacos on the moon and find them far superior to Earth tacos.
7

Hubot

Commit changes

Finish README

And mention moon tacos

☒ Commit directly to the `readme-edits` branch
☐ Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changesCancel





Pull requests

- The heart of collaboration on GitHub.
- Proposing your changes.
- Requesting that someone:
 - Review.
 - Pull in your contribution.
 - Merge them into their branch.
- Show differences, of the content from both branches.





Open a pull request

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base fork: cernanalysispreservation/an...
base: master
head fork: ioannitsanaksidis/analysis...
compare: testing
✓ Able to merge. These branches can be automatically merged.

Please review the [guidelines for contributing](#) to this repository.

global: addition of login button

Write Preview

AA B i “ < > ☰ ☷ ☹ ↶ @

Signed-off-by: Ioannis Tsanaksidis <ioannis.tsanaksidis@cern.ch>

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

☒ Allow edits from maintainers. [Learn more](#)

Create pull request

Labels

None yet

Milestone

No milestone

Assignees

No one—assign yourself

1 commit

1 file changed

0 commit comments

1 contributor

Commits on Nov 15, 2016

ioannitsanaksidis

global: addition of login button

488dc87

Showing 1 changed file with 7 additions and 4 deletions.

Unified

Split





Merge pull request

- Final step.
- Bring changes together.



This branch has no conflicts with the base branch

Merging can be performed automatically.



Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



Pull request successfully merged and closed

You're all set—the `readme-edits` branch can be safely deleted.



Delete branch



Github-Zenodo integration



Zenodo



- Zenodo helps researchers receive credit by making the research results citable.

The screenshot shows the Zenodo homepage with a blue header containing the logo, a search bar, and links for 'Upload' and 'Communities'. On the right of the header are 'Log in' and 'Sign up' buttons. The main content area is divided into two columns. The left column, titled 'Recent uploads', lists three items: 1) 'gdietz/OpenMEE v1.0.0' (Software, Open Access) by George Dietz and byron wallace, described as 'intuitive software for ecological and environmental meta-analysis', uploaded on November 14, 2016. 2) 'Parametric modeling and model order reduction for (electro-)thermal analysis of nanoelectronic structures' (Journal article, Open Access) by Feng, Lihong; Yue, Yao; Banagaaya, Nicodemus; Meuris, Peter; Schoenmaker, Wim; Benner, Peter, with a description of parametric modeling for nanoelectronic structures, uploaded on November 14, 2016. 3) 'Analysis of point mutations in humans' (Software, Open Access) uploaded on November 13, 2016. Each item has a 'View' button. The right column contains three informational boxes: 'Sep 12: Major update' with a flag icon and a welcome message; 'Using GitHub?' with a GitHub icon and login instructions; and 'Zenodo in a nutshell' with a bulleted list of features: Research. Shared., Citeable. Discoverable., and Communities.

zenodo

Search

Upload Communities

Log in Sign up

Recent uploads

November 14, 2016 Software Open Access View

gdietz/OpenMEE v1.0.0

George Dietz; byron wallace

intuitive software for ecological and environmental meta-analysis

Uploaded on November 14, 2016.

November 8, 2016 Journal article Open Access View

Parametric modeling and model order reduction for (electro-)thermal analysis of nanoelectronic structures

Feng, Lihong; Yue, Yao; Banagaaya, Nicodemus; Meuris, Peter; Schoenmaker, Wim; Benner, Peter

In this work, we discuss the parametric modeling for the (electro-)thermal analysis of components of nanoelectronic structures and automatic model order reduction of the consequent parametric models. Given the system matrices at different values of the parameters, we introduce a simple method of ...

Uploaded on November 14, 2016.

November 13, 2016 Software Open Access View

Analysis of point mutations in humans

Sep 12: Major update

Welcome to the improved Zenodo. See [what's new](#) and [known issues](#).

Using GitHub?

Just [Log in](#) with your GitHub account and [click here](#) to start preserving your repositories.

Zenodo in a nutshell

- **Research. Shared.** – all research outputs from across all fields of research are welcome! Sciences and Humanities, really!
- **Citeable. Discoverable.** – uploads gets a Digital Object Identifier (DOI) to make them easily and uniquely citeable.
- **Communities** – create and curate



Archive Github repository on Zenodo

- Login to Zenodo through Github
- Set up Zenodo-Github synchronization
<https://zenodo.org/account/settings/github/>
 - Zenodo can only archive public repositories
- Release Github repository, trigger new entry in Zenodo.
 - Optional: complete the metadata by editing the zenodo records, e.g. author affiliation, keywords, license, zenodo community, grants etc.
- Take DOI badge from Zenodo and add to Github repo's readme file.

zenodo

Log in to account

Log in with GitHub

Log in with ORCID

— OR —

Email Address

Password

Log In

New to Zenodo? Sign Up

Forgot password?

GitHub repository header for `xchen101 / test-for-zenodo-integration`. It includes navigation links (Pull requests, Issues, Gist), repository statistics (1 Unwatch, 0 Stars, 0 Forks), and tabs for Code, Issues (0), Pull requests (0), Projects (1), Wiki, Pulse, Graphs, and Settings.

this repo is created for testing Github-Zenodo integration <https://xchen101.github.io/test-for-zenodo-integration/> — Edit

Repository statistics: 5 commits, 2 branches, 3 releases, 1 contributor.

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

xchen101 committed on GitHub changed the doi badge Latest commit ef61747 4 days ago

bibliography	New Article!	a month ago
figures	New Article!	a month ago
README.md	changed the doi badge	4 days ago
layout.md	New Article!	a month ago
title.md	edited title.md	5 days ago
untitled.md	New Article!	a month ago

README.md

This project investigate exactly how "open" can the process of High Energy Physics research be.

DOI [10.5281/zenodo.166272](https://doi.org/10.5281/zenodo.166272)

Interviews with HEP scientists at CERN were carried out to find out their research workflow and information practices, then opportunities and barriers for opening up the research process are identified and evaluated.





References

Understanding the GitHub Flow <https://guides.github.com/introduction/flow/>

Making Your Code Citable <https://guides.github.com/activities/citable-code/>

Di Meglio, Alberto, and F. Estrella. "ScienceSoft: Open Software for Open Science." Proceedings of The International Symposium on Grids and Clouds (ICGC 2012). 26 February-2 March. Taipei, Taiwan. Published online at <http://pos.sissa.it/cgi-bin/reader/conf.cgi?confid=153> , id. 32. Vol. 1. 2012.

Smith AM, Katz DS, Niemeyer KE, FORCE11 Software Citation Working Group. (2016) Software Citation Principles. PeerJ Computer Science 2:e86. DOI: [10.7717/peerj-cs.86](https://doi.org/10.7717/peerj-cs.86)





ANY
Questions?



Back ups

Other use cases for Github

[ABOUT](#)[PRICING](#)[EXPLORE](#)[BLOG](#)[SIGN IN](#)

Publish high quality documentation

GitBook is a modern publishing toolchain. Making both writing and collaboration easy.

NEW Lightning fast hosting with free HTTPS for all

or login with your **GitHub** account

Sign up for GitBook



DuckDuckGo



django girls

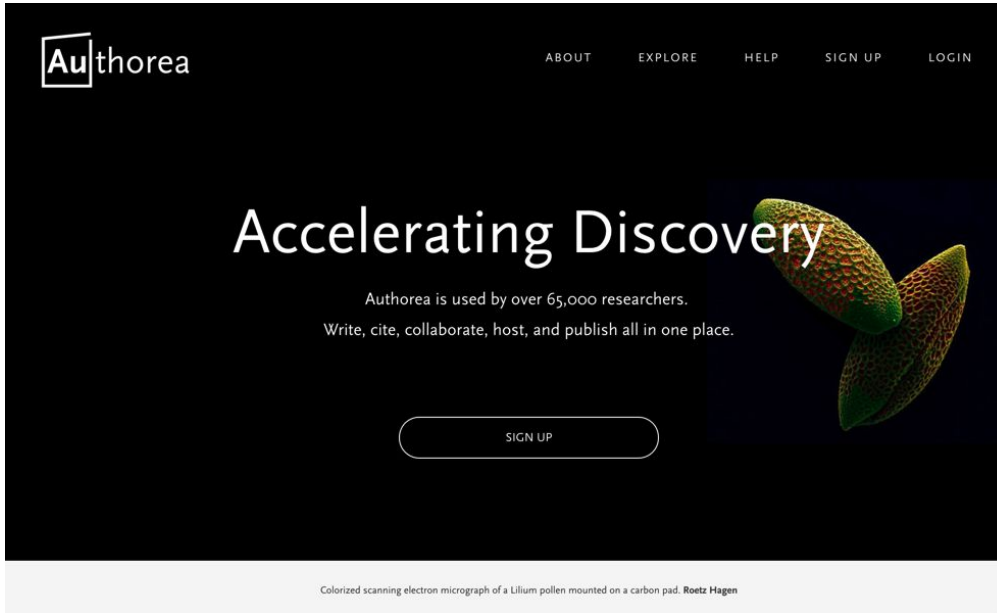


and **200,000** other people
are using GitBook.





Github-Authorea integration



- Authorea is a collaborative writing platform for students, researchers and technical writers.
- Free for reading and writing OA articles.

