

Ensuring that documentation is a first-class citizen in open source projects

Ray Paik (Cube Dev) & Sofia Wallin (Ericsson Software Technology)

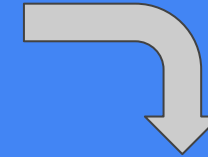
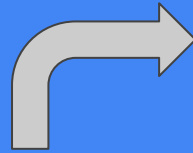
Agenda

- Why is documentation important?
- Common challenges with open source documentation
- Documentation projects in LF Networking
- Documentation at company-led open source projects
- Events to encourage docs contributions
- Summary/recommendations

Why is documentation important?

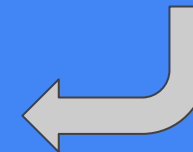
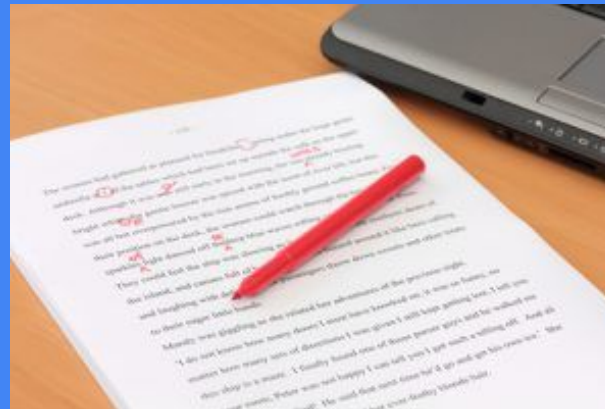
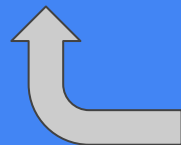


Common challenges with open source documentation



CONSIST
E
N
C
Y

Vicious cycle



LF Networking documentation

A couple of years ago, a cross community working group was established for LF Networking (LFN) with the goal of providing a common way for documentation handling.

LF Networking Documentation

Achievements:

- Consistency across LFN
- Provided clear and simple guidelines for how to work with documentation
- Treating documentation contributions the same as code contributions (e.g. for metrics as well as recognitions)
- Local project style guides and templates

Common documentation and tool guide:

<https://docs.relog.linuxfoundation.org/en/latest/>

LF Networking documentation

Example from Open Network Automation Platform (ONAP)

- Designated milestones for documentation for each release cycle.
 - Planning, repo structure with templates, reviews and sign-off.
- Don't let projects “get away” with not completing their documentation.
- Two documentation hackathons per release cycle.

This page summarizes the projects participating in the Frankfurt Release.

- Exceptions to Java 11 and Python 3 Migration at End of Frankfurt Release
- Frankfurt Code Impact View per Component
- Frankfurt Defect Status
- Frankfurt Deliverables by Milestone
- **Frankfurt Documentation**
- Frankfurt Milestone and Requirements Exception Requests
- Frankfurt Milestone Status
- Frankfurt Release Key Updates
- Frankfurt Release Platform Maturity
- Frankfurt Release Requirements
- Frankfurt Risks

- M1 Status
- M2/M3 Status
- M4 Status
- RCx
- Sign-off

01. Example (release management confluence)

Procedure

M1

1. Find JIRA task attached to M1 Epic for your project, if you haven't already done so.
2. In the table above, fill in
 - project name
 - doc title,
 - doc owner (use Confluence @ notation for name, e.g., @David McBride)
 - current doc state
 - activity planned for release
3. Update status of JIRA task as Done

M2/M3

1. Find JIRA task attached to M2 Epic for your project, if you haven't already done so.
2. Note that this task applies to new documents only (i.e., "New" in Current Document State column). If this is not a new document, just enter "N/A" in the Preliminary Documentation Completed column in the table above.
3. For new documents, at a minimum create a placeholder document (e.g., document with a title, or an outline), and commit to the repo. Add a link to the gerrit to the Preliminary Documentation column in the table above.
4. Update the status of the JIRA task as Done.

Post M4

1. Find the dates for the two documentation hackathons.
2. The document owner should plan on attending at least one of the two hackathons.
3. Enter a green checkmark in the appropriate column in the table above for the hackathon that was attended.

RC0

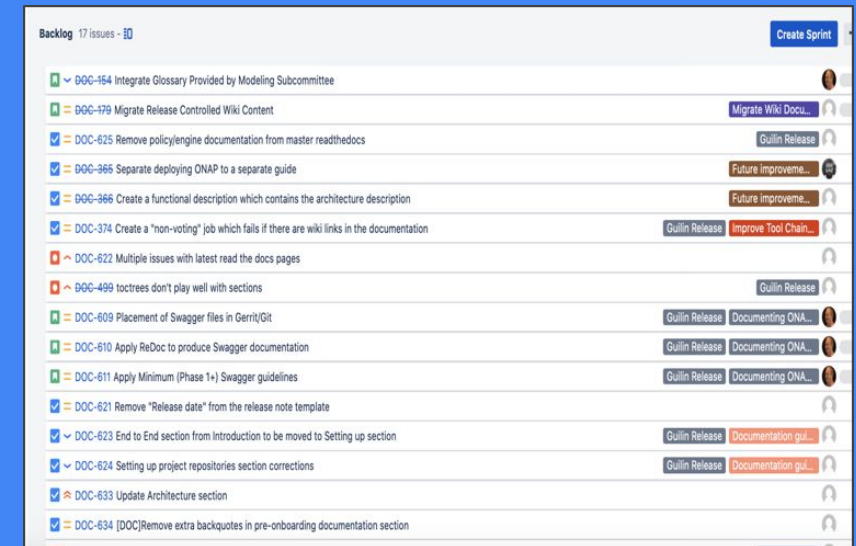
1. Find JIRA task attached to RC0 Epic for your project, if you haven't already done so.
2. Add a link to the document in ReadTheDocs in the appropriate column in the table above.
3. Update the status of the JIRA task as Done.

02. Example (release management confluence)

LF Networking documentation

Example from Open Network Automation Platform (ONAP)

- Provide templates with expected level of content.
- Tracking documentation tasks in Jira.
- The documentation is stored in respective project repos to enforce responsibility.



03. Example (Jira tracking)

Documentation should be a core part of your product/project


- Including documentation as a part of “definition of done”.


The left screenshot shows a GitLab repository interface. The file browser on the left lists various folders: app, bin, builds, cable, changelogs, config, danger, db, **doc** (highlighted with a red box), docker, ee, file_hooks, fixtures, generator_templates, haml_lint, and lib. The right screenshot shows the 'cube.js' repository on GitHub. The commit history table lists recent commits, with the 'docs' folder highlighted in a red box. The table includes columns for the commit message, the commit hash, the time since the commit, and the number of commits. The 'docs' folder is associated with the commit 'docs: connectedComponent (#1185)'.

Commit	Message	Hash	Time	Commits
Levhav docs: connectedComponent (#1185)	docs: connectedComponent (#1185)	4081e5a	10 hours ago	2,281
.circleci	Move prePublishonly to build as fails in publish put npm in undefine...		10 hours ago	
.github	Run build on CI		11 hours ago	
docs-gen	Typedoc gen signature fix (#1080)		29 days ago	
docs	docs: connectedComponent (#1185)		10 hours ago	
examples	misc(examples): Add Product Hunt badge to Slack Vibe		7 days ago	
guides	example: Material UI Dashboard (#717)		15 days ago	
packages	Move prePublishonly to build as fails in publish put npm in undefine...		10 hours ago	
.eslintrc.js	feat: date range comparison support (#979)		2 months ago	
.gitignore	docs(cubejs-client-react): react docs generation (#787)		3 months ago	
.nvmrc	Feature: Introduce .nvmrc with version similar to CI		22 days ago	
.prettierrc	fix(cubejs-client-core): table pivot (#672)		4 months ago	
CHANGELOG.md	v0.21.0		10 hours ago	
CODE_OF_CONDUCT.md	Create CODE_OF_CONDUCT.md		2 years ago	
CONTRIBUTING.md	feat: Add server-core options validate (#1089)		25 days ago	
LICENSE	Create LICENSE		10 months ago	
README.md	Update README.md		2 months ago	
SECURITY.md	Update SECURITY.md		5 months ago	
lerna-debug.log	feat: angular query builder (#1073)		8 days ago	
lerna-publish.sh	Lerna publish through version		3 months ago	
lerna.json	v0.21.0		10 hours ago	
package.json	feat: Introduce Druid driver (#1099)		15 days ago	
rollup.config.js	feat: angular query builder (#1073)		8 days ago	
yarn.lock	feat: Introduce Druid driver (#1099)		15 days ago	

Contribution process for documentation

- The process for contributions/improvements should be the same between code and documentation.
- There should also be “Edit this page” button on each page to make contributions easier.


 **Help & feedback**



Docs

[Edit this page](#) to fix an error or add an improvement in a merge request

[Create an issue](#) to suggest an improvement to this doc

 **cube.js docs**

[Home](#)

[Getting Started](#)

[Cube.js Introduction](#)

[Cube.js CLI](#)

[Cube.js Backend](#)

[Cube.js Frontend](#)

[Data Schema](#)

Cube.js Introduction

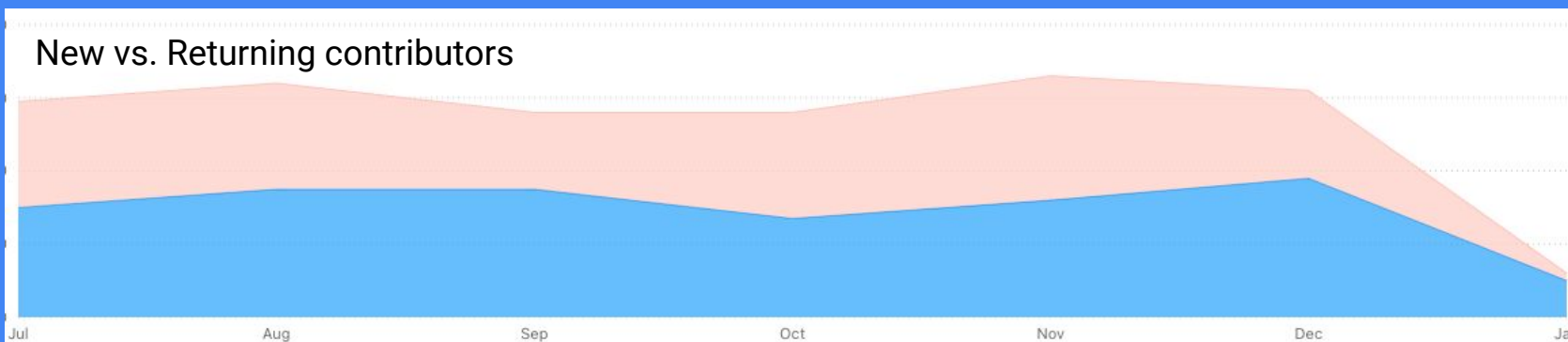
Cube.js is an open source modular framework to build analytical web applications. It is primarily used to build internal business intelligence tools or to add customer-facing analytics to an existing application.

Cube.js was designed to work with Serverless Query Engines like AWS Athena and Google BigQuery. Multi-stage querying approach makes it suitable for handling trillions of data points. Most modern RDBMS work with Cube.js as well and can be tuned for adequate performance.

[Edit this page](#)

Recognizing contributions

- For community contributions, there should be no distinction between code vs. documentation contributions.
 - For recognitions, community metrics, community leadership, etc.



I just received my [#myfirstMRmerged](#) cup from [@gitlab](#) ❤️



Ambassador

Advisory Board

Core Team

Heroes

Steering Committee

.....

Organizing events for your documentation

- Synchronous events to encourage people to contribute to your docs or even to get started in your open source project.
 - Have an event on people's calendars and provide opportunities to network (even in virtual settings).
 - For new contributors, having a list of documentation issues ([example](#)) provides an easy way to get started and experience early success.
 - Highlight areas of documentation that participants can help “debug” (e.g. “Getting started”, “Deployment guide”, “Tutorials”, etc.)
 - There are also industry events for people interested in documentation such as [Write the Docs](#)



Each year we run an event called the Writing Day during the conference. This event is modeled after the concept of [sprints](#), which are common in open-source conferences. The main premise is to get a bunch of interesting people in a room together and have them work towards shared goals.

To make the most out of the Writing Day, we recommend that you either come with a project, or be ready and willing to contribute to someone else's project.

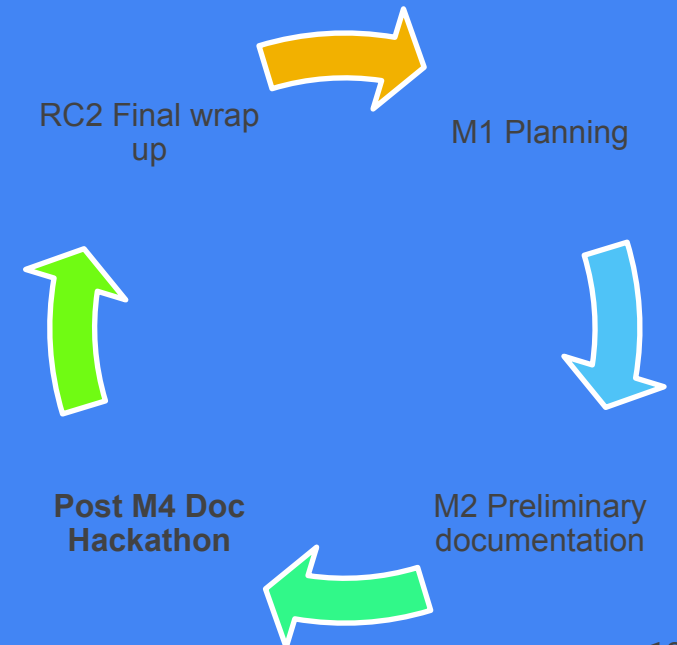
Here are some examples of projects that you might want to work on:

- Open source software documentation
- General documentation writing
- Best practices manual (For your company, or the world)
- Blog posts
- Tips and tricks
- Great works of fiction
- Love letters
- The Documentarian Manifesto

How to encourage community engagement, doc specific events (LFN)

- 2 hackathons/release cycle (virtual participation)
- A full day dedicated release related content
- We perform reviews, are available for questions and general documentation introduction
- Participation is part of the release requirements
- Dates always set with respect to code freeze

ONAP Guilin release hackathon



Summary/recommendations

- Ensure that documentation is a key part of your product/project.
- Consistent process between submitting/reviewing/merging code and documentation.
- Have a well documented & easy to follow process for contributing to documentation.
- Everyone in the community recognizes and understands the value of documentation.

Our blog post on opensource.com:

<https://opensource.com/article/20/8/documentation-open-source-projects>

