

(How to) Be a Good Citizen in Open-Source Documentation

Open Source Summit Japan 2019

Robert Kratky

Principal Technical Writer



twitter.com/rkratky



rkratky@redhat.com

In this presentation

Why docs & how docs differ from code

Good practices

- Take docs seriously

- Be reasonably humble

- Make it easy for contributors

How to downstream docs efficiently

Why (upstream) docs?

(Good) docs help the project

(Downstream) product docs made simpler

Easy way to get involved in a community

For you

For new contributors

How docs differ from code?

They don't! (Or they don't have to.)

Plain-text source code ✓

Live in Git ✓

CI/CD for testing & deployment (publishing) ✓

⇒ **“Docs as code”**

Take docs seriously

Make docs a part of your development process

Build a dedicated docs team (get tech writers)

Ensure docs are up-to-date & tested

Be reasonably humble

If a community exists, respect it

If you're building a community, don't behave like you own it (even if you do)

Don't ignore community contributions

Community Docs Collaboration Guide

(example: <https://redhat-documentation.github.io/community-collaboration-guide/>)

Make it easy for contributors

Provide docs for contributors (CONTRIBUTING.md)

Define terminology & conventions + create or choose a style guide
(example: <https://redhat-documentation.github.io/asciidoc-markup-conventions/>)

Use templates
(example: <https://redhat-documentation.github.io/modular-docs/>)

Allow for local building of docs on all platforms

How to downstream efficiently

Upstream first (develop in upstream and then productize)

Use the same markup language in upstream and downstream

Don't diverge from upstream; even if you do, single-source

Automate the downstreaming process

Takeaways & questions

Don't underestimate docs

Be a good community member

Make life easy for all contributors

Develop in upstream for a healthy community & efficient productization

Thank you



twitter.com/rkratky



rkratky@redhat.com