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
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.relenq.linuxfoundation.org/en/latest/
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
- M3/M1 Status
- M4 Status
- RCx
- Sign-off

Procedure

M1:
1. Find JIRA task attached to M1 Epic for your project, if you haven’t already done so.
2. In the table below, fill in:
   - project name
   - doc title
   - doc owner (note Confluence @ mentions for name, e.g., @Jiann/Schacht"
   - effort estimate (note"
   - 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 (not a “new in Current Document State” column). If this is not a new document, just enter “Null” in the Preliminary Document Status column.
3. For new documents, at minimum create a placeholder document (e.g., document with a title, line on notecard), and commit to the repo. Add a link to the get into the Preliminary Document column in the table above.
4. Update the status of the JIRA task as Done.

Post M4
1. Find the details 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.

01. Example (release management confluence)

02. Example (release management confluence)
- Provide templates with expected level of content.
- Tracking documentation tasks in Jira.
- The documentation is stored in respective project repos to enforce responsibility.
Documentation should be a core part of your product/project

- Including documentation as a part of "definition of done".
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.
Recognizing contributions

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

New vs. Returning contributors

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
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