OPEN SOURCE DOCUMENTATION GUIDE

#ASKnet

LEGEND

Steps you absolutely have to take!

Optional steps to make it extra special

OPTIONAL: *Provide a development branch for people trying new features.*

Each Release provides better reference and enables people to fork (multiple sub streams and variations of the project) and improve your outcome, spread the word and gain new resources and users.

5 Release

Release the project in early packages. Give it a version number and provide the best access you can. Version control (eg. Git) based platforms can help with this.

types of release:
unstable release (development),
beta version (pre-release),
stable release (ready package),
new upgrades/features etc.

1 Openness

In order to make sure your project has open source freedom for others to use and adapt, you need to give it an Open Source License. Choose from one of these Open Source Licenses:

Copyleft Licenses	Permissive Licenses
CC-BY-SA 4.0	Apache 2.0
CERN OHL	CC-BY 4.0
GPLv3	MIT License

Anything that restricts the use and remix of the project (like NC/ND) or has no license, is NOT Open Source.

4 Guidance

Collaboration is key in Open Source projects and needs maintenance, moderation and constant activity.

Create guidance around your project as visually as you can. Facilitate and encourage collaboration with instructions to various target groups.

For example:

contribution guide: how can people contribute

how to guide: how to build something

user tutorials: how to use the product

developer guide: technical steps for developers

3.1. MAINTENANCE

Someone is taking care of follow-up: responding to suggestions, making changes, fixing bugs. Even if you are the product owner, take care of documentation when you make improvements.

WHAT IS THE CONCEPT?

Describe & outline your project idea. You want to share your project with the world? Give it a name, give it a reason to be & give it a go!

1.1. README.MD

Create a description file and put the license text inside your repository. Do this before you publish anything about your project!

3 Source

Share your source files as soon as possible! Original content files, text, sources are essential from the beginning to help understand what you're doing and with the process of documentation itself.

Anything READ ONLY is NOT a source file! Don't make people reverse engineer your work to create or adapt something new.

Keep in mind that Git based systems don't work well with big sized binary files (videos, large photos or PDFs, etc.), you might upload them somewhere else and share the download link.

2 Platform

Choose a platform that fits your needs. Where will you publish or share content? eg: github, gitlab, nextcloud, website/wiki, wikifab, wikifactory

Things to keep in mind:

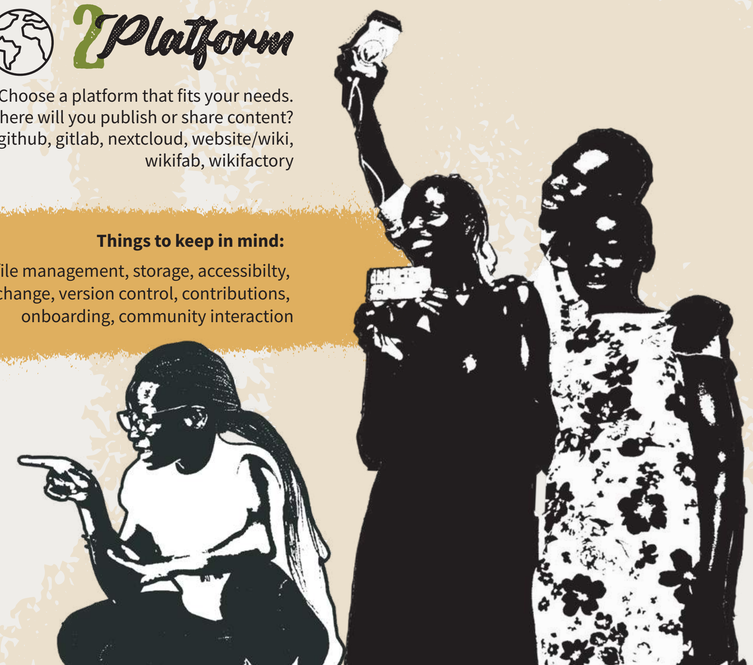
file management, storage, accessibilty, exchange, version control, contributions, onboarding, community interaction

2.1. FORUM

Issue management
A place where people can discuss questions and issues with the project. Onboarding / forum connected to platform choice eg: github has issues management and tracking included. If people have trouble making the thing, how to suggest improvements, track changes, bugs.

2.2. READ ONLY

Generate READ ONLY output files on each platform to visualise the project. Give people access to VIEW and use the project without making modifications or changes. Ability to READ across platforms to view, without having to use proprietary software.



#ASKnet

Access to Skills and Knowledge Network

OPEN SOURCE DOCUMENTATION GUIDE

WHY OPEN SOURCE DOCUMENTATION?

Share your knowledge with the world!

To build up free knowledge for everyone.

Increase the momentum of innovation with collaborative development & gain access to resources, tools and knowledge by sharing your work.

Find communities that are already using these tools and platforms

Open Source helps with access to information in fragile or post-conflict scenarios & areas where you need to build up infrastructure fast and increase capacity. Strengthen your local community and join forces with an international open source alliance!

Accessibility - you will need to provide a clear structure and give details on how to contribute to the project.

Be clear on WHY do you do it and able to explain that.

Connect to the community that is going to use and help develop it.

Talk to people, share examples and knowledge.

By collaborating publically everyone contributes their own resources to the project. Especially in cases where you don't have all the knowledge or skills to finish a project, it's the best way to get it done. If you want to save resources by not revinenting the wheel, the best way is to join forces and collaborate. Avoid complicated contracts with an open source license, and share your work with everybody who wants to participate.

DOCUMENTATION SCENARIOS & COMMUNITY

What to document so everyone is on board?

Understand the structure of information and processes to explain what you have made. Understand how open source documentation works so you know what to keep in mind: eg. file types, templates, formats, video/photo text/diagrams, step-by-step instructions.

- processes, tools, readme files
- how to contribute to this project
- platform choice to collaborate
- back-up structure to work offline
- team of people to do documentation

Know what to keep in mind:
eg. file types, templates,
formats, video/photo text/diagrams,
step-by-step instructions.

Open Source Guide to Building Community

opensource.guide/building-community/

»Good documentation only becomes more important as your community grows. Casual contributors, who may not otherwise be familiar with your project, read your documentation to quickly get the context they need.«

»Your README is more than just a set of instructions. It's also a place to talk about your goals, product vision, and roadmap. If people are overly focused on debating the merit of a particular feature, it may help to revisit your README and talk about the higher vision of your project. Focusing on your README also depersonalizes the conversation, so you can have a constructive discussion.«
Opensource.guide

Any question that comes up in the process might be relevant for collaborators so include answers to these in your documentation!

HOW TO MAKE IT OPEN SOURCE?

Use an open source license!

There are different types of Open Source Licenses to choose from.

COPYLEFT LICENSES

»Copyleft or Viral licenses allow anyone to use, explore, distribute or modify the projects, but you must publicly contribute and commit all the modifications that you have made to the original project. The original project is kept updated, and it has evolved. A person or organization using or depending upon it is legally bound to share their own modification, help maintain the project, and contribute to the updates.«

CC-BY-SA 4.0	**creativecommons.org/licenses/by-sa/4.0/**
CERN Open Hardware Lic.	**www.ohwr.org/project/cernohl/wikis/home**
GPLv3	**tldrlegal.com/license/gnu-general-public-license-v3-(gpl-3)**
Mozilla Public License 2.0	**tldrlegal.com/license/mozilla-public-license-2.0-(mpl-2)**

PERMISSIVE LICENSES

»Permissive licenses allow creators to release a project as open source so anyone can explore, use, modify and distribute it in any way. Also known as "copyright licenses".«

Apache 2.0	**www.apache.org/licenses/LICENSE-2.0**
CC-BY 4.0	**creativecommons.org/licenses/by/4.0/**
MIT License	**en.wikipedia.org/wiki/MIT_License**

If you choose CC-NC Non-Commercial or ND - no derivatives - these restrict use and remix of the project. Therefore it is NOT Open Source. Also, if you don't choose any license then you have not given permission for anyone else to adapt or use your work.

OPEN SOURCE HARDWARE & THE LICENSES

- How do you open your work to collaboration?
- Does it make sense to use a git, cloud server or wiki?
- Where do you want to "host" your community?
- What are the steps to make contributions open?

What are you sharing?
Is it documentation of a product, technical process, hardware, software (sourcecode), concept/idea, book, training module? Understand the framework you need based on the open tools available.

The main thing is you always share the "source code" that means all your working files that you use to create this output: that means technical drawings & CAD models, text and svg files for design layout.

What type of content are you documenting?
This will define what tools, platforms or processes to use!

Examples for platforms according to use-case
- Open hardware & software 'source code' repository for files (original, versionised, compare): Github/Gitlab
- Step-by-Step Instructions: wikifab/wikihow
- Text based collaboration: etherpad, codimd

Create a visual process:
show technical diagrams, illustrations, roadmaps

»An open source project with no license attached — no matter how remarkable it is — is avoided for use by everyone. For others to use, distribute, and build upon projects the creator must have first given their express permission outlining use of their designs, constituting it as open source. No license, in practice, means you are abandoning your hard work in the wild instead of owning it and sharing with others.« *Usama Abid on Medium*

Open by license:
medium.com/inventhub/open-source-hardware-the-licenses-a244733e6cb7

#ASKNET PROGRAM

ACCESS TO SKILLS AND KNOWLEDGE NETWORK

#ASKnet provides access to skills and knowledge to empower youth, address community challenges and transform cultural patterns.

#ASKnet (Access to Skills and Knowledge Network) is a capacity building and hub development program linking six community based youth-led innovation hubs in South Sudan, Uganda and Kenya. The #ASKnet community aims to build a sustainable network of trainers and empowered individuals, to address challenges their communities are facing and transform cultural patterns that fuel conflict and inequality. Providing youth access to skills and knowledge, Training of Trainers (ToT) workshops in open source hardware and software using the #ASKotec, entrepreneurship, media production, gender equality awareness, trauma healing and financial literacy. #ASKnet innovation training is an initiative by r0g_agency for open culture and critical transformation (Berlin), funded by the 'Access to Information and Supporting Freedom of Expression' program of the German Federal Ministry of Economic Cooperation and Development (BMZ).

This Open Source Documentation Guide aims to help you understand what you can do to prepare real open source documentation, how you come up with open source solutions, finalised products and overcome challenges in process and methods.

There are many different reasons to document your work, process, product or invention: for experts, learners, social media team, documentation team, PR, funding and to invite people to work with you on developing the project, hardware or software further.

GLOSSARY OF TERMS

CodiMD: Open source platform to write & collaborate on markdown text to host on your own server and that you can use for free. *github.com/codimd/server, demo.codimd.org/*

HackMD: Closed source markdown platform similar to CodiMD.

Markdown: Lightweight markup language with plain-text-formatting syntax. Easily written and displayed by lots of platforms. Because of its simplicity it's useful for documenting your user guides, meeting minutes and readme files. Not meant for layout / design!

Issue report: When something isn't working, alert the developer to fix it! When you find something that can be improved, suggestions to make or general technical issue.

Feature request: Special type of issue report that shares ideas to improve the project.

Fork: When you copy all the files of a project into a new repository, name it and create your own new offshoot - make the changes yourself.

Merge/pull request: Where you merge your new work back into the existing repository, with the additional features or work you have done.

Issue tracker: Management area for your issues.

Telegram: It's a text based messenger that you can use to share files, management teams, have news channels, to help organise your project.

GitHub: It's a git based collaboration platform, that allows you to share the source (code/text/files) of your work and invite other people to adapt, develop, work on it and contribute to new versions!

»(/gɪt/): is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files.« */Wikipedia*

GitLab: An open source example of git based development platform - mainly software, can be used for hardware and open source projects. You can work on decentralised, versionalised project on these platforms. More tools you can use for free.

The magic about git is you can work on your computer locally, and push it later to the repository. The platform is the project & file management front end, where you track issues, and feature requests, manage your community and collaborate with the team.

Repository: A place where your files sit / are stored / but also shared… Public Open Visible. Keep a local copy of your files on your computer, especially when using any online / cloud platform or demo version.

Wiki: A website or database developed collaboratively by a community of users, allowing any user to add and edit content. Wiki is also a Hawaiian word which means "fast or quick".

Creative Commons Licenses A set of rules you can choose from to release your creative work. Some of the CC options are open source licenses, like CC-BY or CC-BY-SA where BY = attribution, and SA = share alike. *creativecommons.org*

"That's what the magic is about open source! You can take multiple ideas, realised - although your idea takes a new path from the original. Whenever you fork something and you IMPROVE it by adding features or new design. Always link back to the original design. In case you fork something, make it known. Look at the license that it has and follow the terms and conditions of the license before you start!" Timm Wille

CREDITS

#ASKnet Open Source Documentation Guide

GitHub Repository to add info and contribute:

github.com/opencultureagency/Open-Documentation-Guide

Project by: In partnership with: With financial support from the:

#ASKnet Hubs:

Published by r0g_agency for open culture and critical transformation gGmbH under a creative commons Attribution-ShareAlike 4.0 International license. Version 1. 2020

FILE FORMATS & TIPS

Share your source files: everything you need to collaborate without reconstructing the whole thing.

Understand what platform to use (versioning*): e.g. GitLab, Wikifab, Wikifactory. Version control allows you to keep track of each new contribution and releases! *Each change is embedded in history of repository.

DESIGN

SVG - open illustration format

JPG & PNG (these are OUTPUT formats) – understand the best practices to save, name and share your actual source files!

WORDS

Text files -
odt / docx / txt / pad / Readme

TECHNICAL FILES

CAD Files - technical drawings

STEP/STP & SVG - interoperabale formats in case your working file is not open software based STL.

PNG & PDF - provide platform independent viewable formats to enable more people to read, understand and share the project/concept. (For each release!)

BINARY FILES

Videos, photos and other big sized files (e.g. non-text-based, packaged) should either be outsourced to an upload platform (Cloud, Video platform etc.) or kept to a minimum.

TO DO'S AND TO DONT'S

gdoc
to gdoc or not to gdoc? Why use/don't use? (issues for privacy, google using your data as product, force you to sign up with email address to participate)

Privacy
data issues - open source doesn't mean you share everything, need to protect the privacy of other people

Contribution
list to see who put what effort into x project - git based documentation let's you see who did what in terms of project - if you make a product or service for instance.

The "Recipe" for an Open Documentation

EXAMPLE: REPAIR CAFE

1. Select your project to document: in this case, electronics or hardware repairs.
2. Create a guide, simple how-to repair radio, mobile phone or how to use a soldering iron etc.
3. Technical examples, diagrams, outline steps and give overview for the basic process.

This event involves hands on use of tools to repair broken appliances. The #ASKotec materials will come in handy, as people who have their appliances or electronics broken see the repair process and learn the skills to repair their gadgets next time. Document the event on Wikifab.

WHAT IS A REPAIR CAFE?

Repair Cafe is a space to repair broken electronic appliances of the public. Ranging from phones, radios, television and solar panels and more. The Repair Cafe as an open source documentation scenario combines training, workshop and event aspects. Trainers help community members evolve within their repair sessions.

TOOLS

- #ASKotec kit
- soldering iron
- hands
- camera
- computer/laptop

MATERIALS

- soldering wire
- wires
- soletape
- masking tape
- internet
- pens and
- notebooks

(Documentation)

SERVING SUGGESTION

Audience:
Build your community, people with repair needs, keen to learn or create spaces for community interaction, socially & ecologically minded!

Resources:
repaircafe.org

Get more insights via
wikifab.org/wiki/Repair_Cafe'

METHOD

Instructions to make it tasty!

1. **Team Setup**
Provide Blueprint of space setup, briefing teams

2. **Space Setup (repair/maker space)**
Reception team near the door / entry Expert tables (U/C Shape)
Waiting Space / Community Management

3. **Registration Process**
Register devices under the owners name
Tag devices with sticky notes and numbers
Sort devices that can/can't be repaired

4. **Repair Process**
Tool Keeping & maintenance

5. **Education**
Share the skills and steps required to do this at home!

6. **Documentation**
Documentation & list of all repairs to track success
Statistics – repairs completed, diagrams, pictures.

7. **Social Media**
Share Repair Cafe success stories on your social media channels. Use hashtags #ASKnet #RepairCafe.

#ASKnet Open Training on GitHub

Examples of other training content:

OPEN TRAINING GUIDE

How to set up your own training formats

github.com/opencultureagency/Open-Training-Guide