



**HAL**  
open science

## Building Great Documentation

Jean-François Boujut, Romain Pinquié

► **To cite this version:**

Jean-François Boujut, Romain Pinquié. Building Great Documentation. Mies, Robert; Bonvoisin, Jérémy; Villum, Christian; Dekoninck, Elies; Jochem, Roland. Open-Source Hardware Development. A handbook for collaborative product creation, Berlin Universities Publishing, pp.72-95, 2024, 978-3-98781-014-5. 10.14279/depositonce-16548 . hal-04867452

**HAL Id: hal-04867452**

**<https://hal.science/hal-04867452v1>**

Submitted on 16 Jan 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

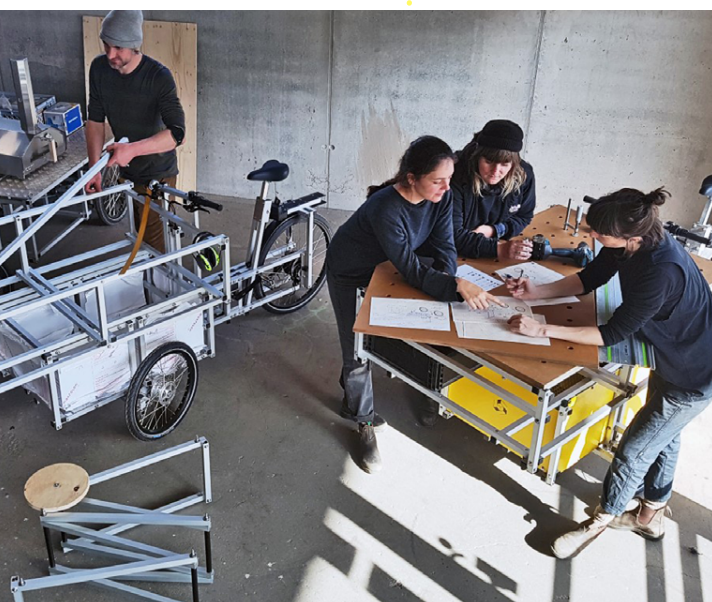


Distributed under a Creative Commons Attribution - NoDerivatives 4.0 International License

# OPEN SOURCE HARDWARE DEVELOPMENT

A handbook for  
collaborative product  
creation

*Tools and  
Guidance  
for your  
Company*



Robert Mies  
Jérémy Bonvoisin  
Christian Villum  
Elies Dekoninck  
Roland Jochem  
eds.

# CREDITS

This handbook was produced from the outset with the aim of creating a coherent and unified collective output, gathering and combining the contributions of an **interdisciplinary team of authors** from the Horizon 2020 project OPENNEXT and close external partners, all of whom deserve the highest recognition. The manuscript was produced by November 2022, the end of the project, and reflects the state of the art at the time.

The content conveyed in this handbook draws heavily on the practical knowledge and insights of the **pioneering companies** and **makerspaces** observed, many of whom have been engaged as reliable cooperation partners over an extended period of time, which together add up to the weight of this book and for which utmost gratitude is expressed.

Special thanks go to the **team of associate editors** who were all very closely involved in the elaboration of this handbook and made essentially all decisions together with the editors and without whom the realisation of this handbook would not have been possible:

- > **Rafaella Antoniou** (University of Bath)
- > **Stephan Bohn** (Alexander von Humboldt Institut für Internet und Gesellschaft – HIIG)
- > **Miriam Gradel** (independent journalist)
- > **Melle Zijlstra** (University of Bath)

All **reviewers, internal** and **external**, are thanked sincerely for their detailed comments and suggestions, which enabled the authors and editors to create a high-quality result. The review process was conducted in the spirit of open source, with full transparency and open access for all participants. Each chapter of this book was reviewed by two or three internal and external reviewers in at least two rounds of scientific quality control. In addition to the consortium members of the H2020 project OPEN\_NEXT, internationally recognised experts in the fields of design science and open source hardware have provided very insightful comments and suggestions, including particularly:

- > **Peter Troxler** (Hogeschool Rotterdam – University of Applied Sciences)
- > **Serena Cangiano** (The University of Applied Sciences and Arts of Southern Switzerland – SUPSI)
- > **Chris McMahon** (University of Bristol)

This handbook was produced in the frame of the project OPENNEXT which received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 869984.<sup>01</sup>

01 <https://cordis.europa.eu/project/id/869984>

# BUILDING GREAT DOCUMENTATION

How can an OSH project improve, scale and drive innovation if there is little or no information about it available? Documentation is essential to making an OSH project available to all; great documentation is what will bring momentum. This chapter acts as a guide for OSH practitioners looking to create and facilitate comprehensive project documentation. The methodological foundation and structure for documentation presented in this chapter enable both the recording and the retrieving of information in OSH projects and can be used to catalyse and streamline mass participation.

## JEAN-FRANÇOIS BOUJUT

*Professor at the School of Industrial Engineering, Grenoble Institute of Technology, and leader of the collaborative design group of the G-SCOP Laboratory. Boujut has extensive knowledge of OSH design and is one of the originators of the OPENNEXT project.*

## ROMAIN PINQUIÉ

*Associate Professor at the School of Industrial Engineering, Grenoble Institute of Technology, and a research fellow at the G-SCOP Laboratory. Pinquié's research focuses on design processes and methods and tools for complex engineered systems.*

# With great OSH comes great documentation

Documentation in OSH is not only tightly linked to the development process itself; it is inherently a part of it. However, documentation itself is not an easy task. Often, a project manager or a company is faced with the dilemma of wanting to document information without having the resources or know-how. In OSH, where the sharing of information is at the core of both design principles and practices, this becomes a much larger issue. More often than not, the information pertaining to an OSH project has been written with a specific aim in mind, which may not match the motivation of the reader or their purpose of use. The detail and accuracy of a product's design information and purpose vary greatly from one project to another, creating both frustration and missed opportunities.

This chapter explains the financial and strategic benefits of investing time and effort into ensuring proper documentation. Furthermore, through the establishment of a more comprehensive documentation structure for general use, readers will be provided with a set of guidelines for good documentation supported by examples of well-documented hardware projects from successful OSH initiatives. This chapter seeks to create a unanimous language for makers, entrepreneurs and OSH practitioners to use when producing and retrieving information in OSH development. One that enables anyone to easily select and sort relevant knowledge, as well as document it, thereby enabling any and all users to study, make and further develop OSH.

# Why document OSH at all?

The release of documentation is the first criterion a piece of hardware must meet in order to qualify as open source.<sup>45</sup> This includes design files that allow for modification and redistribution. When the entire world is invited to participate in a project, documentation becomes the essential language guiding both project managers and participants through the process, making endeavours publicly known and enabling a project or product to gain momentum. The better the documentation of an OSH project is, the more likely it also is to receive contributions. People within the DIY and prototyping maker movement in particular seek out projects with adequate documentation to learn from and contribute to. In other words, documentation enables a project to reach a wider set of skills through an expanded audience, and potentially enable new projects and collaborations in the long run.

Proper documentation also makes it possible for an OSH project to become immortal. With proprietary product development, where designs and innovations are kept hidden, both knowledge and progress risk being lost when a company or project collapses. For a company, opening up a product and providing proper documentation means that an idea, design or vision can live on even when the brand does not. Although the stakeholders of an OSH project may have different motivations for creating, using and maintaining good project documentation,<sup>46</sup> their activities complement each other to ensure the longevity of a project even after completion.

Practically speaking, documentation is vital for the success of an OSH project, as it enables the con-

45 <https://www.oshwa.org/definition/> (accessed 8 November 2023).

46 Makers might be interested in documenting the development process to understand how and why iterations came about, whereas a company might not find this information useful, let alone relevant, once the actual product has been commercialised.

veying of knowledge to users and other developers, allows for a design to be reused or a part or system to be replicated, and enables product use, repair and recycling. The requirements for documentation depend a lot on the aim and maturity of a project.

Learning is often the primary motivation for users seeking out documentation, and knowledge sharing has always been a key dimension and an important motivation for the open source movement.<sup>47</sup> Furthermore, reusing design information is a core feature of industrial product development, and modular approaches and product data management systems have been implemented successfully for many decades to reduce redesign efforts. In OSH, the ability to reuse an existing solution along with the possibility to improve it are key features. Therefore, an important motivation for documentation should be to allow anyone who has the required technical skills to understand what has been created and how, in order to use it, improve it and potentially contribute to increasing the quality of the overall system.

Within the maker movement, replication of existing design is a key motivator for participating in OSH projects, and being able to build a functioning version of a system based on information provided by a third party is one of the fundamental freedoms claimed by the Open Source Hardware Association (OSHW). To be able to benefit from this freedom, individuals should be able to access the minimum amount of information necessary to perform the technical tasks of replication. In other words, without the information, participants cannot benefit from the freedom.

Assembly or disassembly instructions are part of the standard information made available by open source projects today. However, using a product also requires some basic information in order to

operate it. A user manual is an example of readily available documentation that is often included with electronic devices or project kits. Projects may also provide some information about replacing parts or maintaining the product, although this is rare outside the open source community. A good practice is to document OSH with post-use included. This empowers users to recycle or reprocess the hardware rather than simply throwing it away.

Lastly, documentation fulfils the freedom to study and the freedom to make,<sup>48</sup> which are essential elements of human flourishing and enjoyment. The freedom to study implies the accessibility of technical information, such as the native source files, the requirements, the functional models, etc. The freedom to make refers to all the information related to the replication of the product, including manufacturing files, bills of materials and assembly instructions. While the freedom to make is commonly enabled by the relevant information in a project's repositories, supporting the freedom to study can be a lot harder to achieve.

## Nice to know about documentation

To fully understand and leverage the power of documentation, some software and / or coding knowledge is essential. A company or project manager looking to develop OSH should first and foremost be prepared to either acquire some basic knowledge of code or allocate staff or resources to understand the digital aspect of OSH development. Although hardware refers to physical objects, as digital connectivity and the online

47 Bessen, J., & Nuvolari, A. (2011). Knowledge sharing among inventors: Some historical perspectives. LEM Working Paper Series, No. 2011/21, Scuola Superiore Sant'Anna, Laboratory of Economics and Management (LEM), Pisa.

48 <https://www.oshwa.org/definition/> (accessed 8 November 2023).

sharing of source code are the starting points of open source, there is no way around the world of software. The documentation guidelines have been developed specifically to enable anyone to begin structuring documentation in a way that allows the information to be used for multiple purposes and according to the motivations and aims of the user retrieving it.

Currently, there are not many software options on the market whose central function is the generation of documentation. Most online users make use of GitHub and develop their documentation directly using Markdown formatted files. Markdown is a coding language similar to HTML that allows formatted text to be edited using plain text editors. This makes it possible to enable text on different platforms regardless of formatting. Other types of software, like Sphinx<sup>49</sup> or DocuBricks,<sup>50</sup> are specifically dedicated to easing document generation and management. GitHub is a highly useful platform for structuring and providing access to the source documents in native formats (e.g., .docx, .pdf, excl. and so on), as well as for managing access to the latest versions and iteration history. Sphinx, on the other hand, is a document generator that translates files from various Markdown formats into HTML or PDF layout. It is a convenient way to translate GitHub-based Markdown files into fully standalone documentation for use outside of the platform. As such, GitHub and Sphinx can be seen as complementary, as together they enable widespread documentation, and the sharing and retrieving of information. In the example below, the documents necessary for replicating hardware and the requirements for doing so are shared on a Sphinx-based website, with direct access provided through a simple GitHub link. Other alternatives, such as HedgeDoc,<sup>51</sup> a collaborative platform for editing with Markdown, are also used frequently

by open source practitioners. HedgeDoc's highly collaborative and easy-to-use interface makes it even easier for both individuals and teams to use Markdown files to create documentation. As such, there is no one unique, mandatory tool to support documentation, but rather many modes and options to mix and match. When choosing which option and platform to go for, the project team and / or company should first clarify the goal of the documentation.

## EXAMPLE

### STRØM project – translating documentation into different formats

*The documentation for the STRØM project from Lyd by Dissing has been written using the Sphinx documentation platform. Sphinx is a document generator that translates files from reStructuredText or Markdown format into HTML or PDF layout. It is a convenient way to translate GitHub-based Markdown files into fully standalone documentation. It can also be enriched with blog posts. The documentation as shown below is generated from the project repository on GitHub.*

*Find also more information here:  
<https://github.com/LydByDissing/stroem>.*



**Figure 4.1**  
Lyd by Dissing A/S, CC BY-SA 4.0, <https://creativecommons.org/licenses/by-sa/4.0/>, Strøm speaker in conifer linoleum and blue fabric on the soft top, <https://stroem.readthedocs.io/en/latest/#>

49 <https://www.sphinx-doc.org/en/master/> (accessed 16 November 2022).

50 <http://www.docubricks.com/> (accessed 16 November 2022).

51 <https://hedgedoc.org/> (accessed 16 November 2022).

## Choosing a platform: GitHub or Wikifactory?

Software and platform are two inseparable factors that enable appropriate documentation. GitHub is perhaps the most well-known and celebrated platform among the open source community and the most widely used among makers for accessing and sharing documentation relating to OSH. As a documentation management platform, GitHub allows users to implement a change management and version control system based on what is referred to as a pull-request mechanism. This function allows a contributor to fork a project (i.e., split it into two separate processes) and create a *branch* for the contribution. The iteration is then communicated to the project owner via a *pull request*. The project owner may then choose to integrate the modification or new piece of documentation into the project. On GitHub, the latest version of the project's documentation is always displayed. From the project management point of view, the issue tracking system allows for a very open distribution of tasks to be achieved, including documentation. However, there is currently no standardised format or structure for the documentation on GitHub, making it time-consuming and highly difficult for users to find their way in a profusion of files.

An alternative to GitHub is the distributed manufacturing platform Wikifactory. Wikifactory provides great visibility to open source projects and an optimised skill-matching and search engine interface (see Chapter 6 for further benefits). This enables the participation of several people at once and functions to connect designers, makers, entrepreneurs and other users who may contribute towards the project. Wikifactory is not an open source platform per se, as a user profile is required in order to make contributions to a project. It is not mandatory to make a project open source in order to have it hosted on the platform. All

of these things in combination may limit the accessibility of an OSH project for contributors. However, Wikifactory greatly compensates for this by specifically focusing its purpose and software on identifying and connecting competencies, thus enabling a company to more easily find and attract the right skills to a project. The platform also provides an online CAD and manufacturing tool, as well as support, templates and guides for best development practices, including the documentation guidelines that were developed as part of the OPENNEXT project.

A project may benefit from both platforms. Essentially, this depends on the goal of the project, and in relation, the purpose of documentation.

## DIN SPEC 3105: Requirements for technical documentation

A major breakthrough for OSH viability is the DIN SPEC 3105 standard.<sup>52</sup> DIN SPEC 3105 provides a single initial definition of documentation that is fully compliant with the practice of OSH contributors and explicitly references the OSHA definition of OSH described earlier. Furthermore, the standard defines technology-specific documentation criteria (TsDC) that make the documentation more searchable. The standard provides practitioners with the basic requirements for content and accessibility of the documentation, and also includes the aspect of licensing (see Chapter 5). The standard also adopts a product life cycle approach, acknowledging that documentation is not only about making a product, but also about the usage and subsequent processing of materials and components. As such, it is highly recommended to follow this standard through the documentation process.

.....  
<sup>52</sup> [https://wiki.opensourceecology.org/wiki/DIN\\_SPEC\\_3105](https://wiki.opensourceecology.org/wiki/DIN_SPEC_3105) (accessed 12 November 2022).





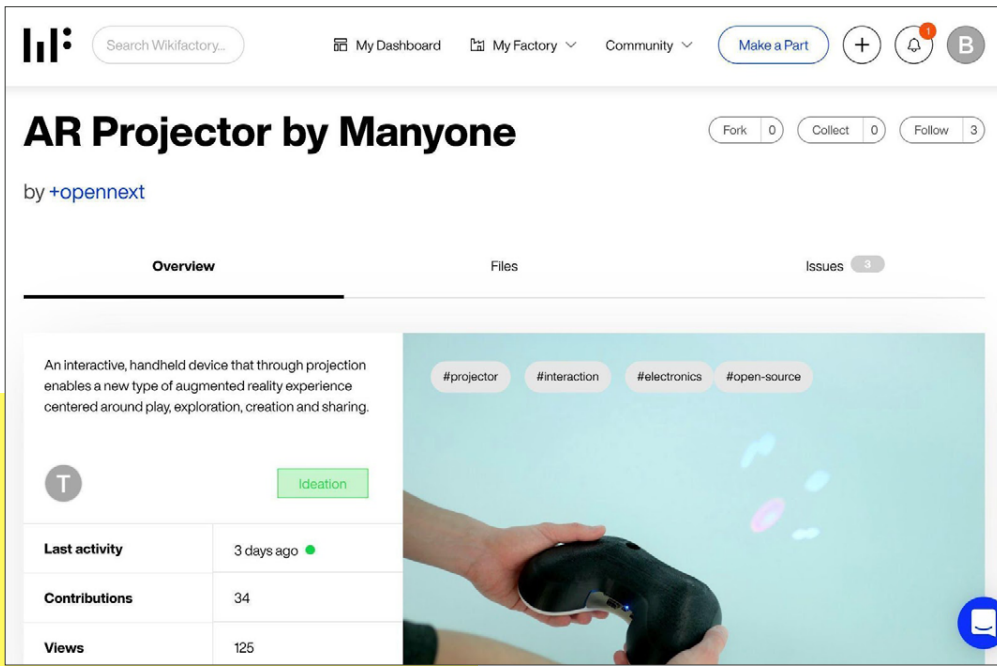
## Step 1: Project introduction

Like any good book, movie or TV series, a proper introduction is essential in order to engage, attract and manage the expectations of potential contributors. An introduction must provide an overview of the project goal and relevant context. Often an introduction may also include news relating to the project (i.e., developments, product enhancements, etc.) and a code of conduct. All of these components enable the reader to understand the motivations of the project and which stakeholders (e.g., participants and sponsors) are or have been involved in its creation. The context also provides information on the technical choices made by the project, including decisions related to compliance and desired impact. Authors may also provide here information on the type of licence(s) applied to let participants know the level of openness granted. Finally, a section on how to contribute is recommended in order to encourage contributions and to provide guidance for the process of doing so. Ideally, an introduction should be able to fully explain the project without the need for visual aids, such as images, although including them will only benefit the understanding.

### EXAMPLE

#### **Manyone – an augmented reality projector**

*The Minimax AR Projector project by Manyone is an example of a well-documented introductory section explaining the motivation and aim of the project. The authors provide a concise and clear picture of their approach, followed by sections detailing motivation, goals and the required skills to contribute.*



**Figure 4.4**  
Tillmann Toben from Manyone, CC BY-SA 4.0, <https://creativecommons.org/licenses/by-sa/4.0/>, Minimax AR – by Manyone's webpage, <https://wikifactory.com/~opennext/manyone>

## Approach

*“How can we use the power of open collaboration to design the experience-enhancing devices of tomorrow?”*

*We believe that a new type of augmented reality experience can be unlocked by combining the functionality of a projector with a portable controller. With this project, we aim to explore that opportunity space by developing an interactive handheld device. Using projection enables this new type of augmented experience centred around play, exploration, creation and sharing.*

*The resulting product will be for everyone in the consumer market aged between 8–70 years. We will realise the open source development by applying collaboration, rapid prototyping and field testing.”*

**Figure 4.5**  
Tillmann Toben from Manyone, CC BY-SA 4.0, <https://creativecommons.org/licenses/by-sa/4.0/>, Minimax AR – by Manyone's webpage, <https://wikifactory.com/~opennext/manyone>

## Step 2: Specifications

Specifications describe a product with a black box-like external view that captures the input-output interactions the product has with its environment (i.e., the users, makers, the external interfaces with other devices, etc.), and any type of constraint related to usage or operation. This is part of the elicitation of the context in a more formal way and serves to elucidate the product requirements. It is generally a tricky task to capture product specifications, and, as a result, examples of hardware documentation that provide clearly described specifications are scarce. When specifications are provided, they are most often limited to technical aspects of the product. Without functional specifications, users are unable to scratch under the surface and understand the context and rationale underlying the technical specification, which limits the potential for further development and innovation. In addition, the way specifications are formatted varies greatly depending on the type of technology embedded in the product. For example, designers who come from a systems engineering background tend to use block diagrams (as shown in the first example below), while mechanical engineers often prefer lists of specifications with performance indicators (as shown in the second example below). In order to attract contributors and foster relevant contributions, it is therefore highly recommended to give equal attention to technical specifications and functional specifications during the documentation process.

## Step 3: Design

In this section of the documentation, readers should be able to find all the relevant technical information related to the structure of the product, as well as native files and formats that allow for modification (e.g., editable CAD files). As this section focuses specifically on information relevant to the design of a product, electrical, mechanical and software are usually the three sub-repositories found here. Sometimes it is relevant to insert behavioural models that simulate the behaviour of relevant aspects of the product. If the intention is to facilitate the modifiability of the product as opposed to allowing completely free and unguided modifications, behavioural models are an important aspect to consider.



**Figure 4.6**  
Valerian Saliou, Public domain, <https://github.com/makar/blob/master/LICENSE.md>, abstract-readme-standard, <https://github.com/makar/blob/master/res/assets/abstract-readme-standard.jpg>



The examples below show a repository in which CAD files are available. OSH projects are encouraged to provide this information as it dramatically increases the openness of the project. While the vast majority of open source projects provide CAD files, very few tend to go beyond this and provide simulation models of the behaviour of the product, which is recommended for greater facilitation on part of the project lead or company.

Notably, the electronic aspects of a product's behaviour are usually more fully documented than mechanical ones. Arguably, this is due to the fact that open source development grew out of the software industry but remains less explored in the aspect of hardware manufacturing. As more hardware companies and developers adopt open source principles, it is likely we will see an increase in documentation files pertaining to the mechanics of the design.

Some projects can benefit from the use of industrial approaches such as model-based system engineering (MBSE), as in the previous MakAir example, where Capella<sup>53</sup> open source software has been used to design the system. The MBSE approach is very efficient in addressing the design of mechatronic objects where the product is made of both mechanical structures and electronics, and controlled with software.

.....  
53 <https://www.eclipse.org/capella/arcadia.html> (accessed 8 November 2023).

## EXAMPLE

### Specification list from SatNOGS Rotator

*The Libre Space Foundation's method for documenting functional and technical specifications for its SatNOGS Rotator is a simple list that eliminates the need for complex software.*

**Figure 4.8**

SatNOGS Wiki, CC BY-SA-3.0, <https://creativecommons.org/licenses/by-sa/3.0/>, SatNOGS Rotator v3, <https://wiki.satnogs.org/File:V3.jpg>



## Specifications

### SatNOGS v3 Rotator

Plastic Parts	15
Non Printed Parts	38
Cost	~ \$220
Controller Electronics	<a href="#">SatNOGS Rotator Controller</a>
Type	AZ/EL (possible X/Y)
Motors	2x NEMA 17 Stepper or 2x DC Motors
Frame Material	Aluminum T-slot 20x20
Speed (deg/sec)	7
Torque (Nm)	?, ~30
Brake Torque (Nm)	?
Dimensions (mm)	280x140x140 (AZ/EL)
Weight (kg)	~5

### Design Goals

Considering the specifications as detailed in the [Mechanical Analysis](#), we would also like for the 3rd version of SatNOGS rotator to be:

- inexpensive (less than €300, if possible)
- lightweight and portable (~6Kg, size:~300x~150x~150mm)
- rigid and durable
- easy to build and fix (try to use easily available materials)
- weatherproof
- electromagnetically shielded, so that noise in reception is reduced
- accurate (<1deg, backlash reduction and use of encoders at the axis)

**Figure 4.9**  
SatNOGS Wiki, CC BY-SA-3.0, <https://creativecommons.org/licenses/by-sa/3.0/>, SatNOGS webpage, [https://wiki.satnogs.org/SatNOGS\\_Rotator\\_v3#Specifications](https://wiki.satnogs.org/SatNOGS_Rotator_v3#Specifications)



## EXAMPLE

### JPL Open Source Rover Project

The JPL project provides a repository on GitHub that contains the mechanical, electrical and software subsystems, source CAD files for modifying the assembly, specifications and technical attributes, schematics, Arduino codes, images and so on. CAD files and other general source files allow the modifiability attribute of open source to remain active.

## Step 4: Manufacturing

Manufacturing instructions mean full descriptions and instructions concerning raw materials, operating conditions and processes to be employed for the manufacture and assembly of the product. As such, the purpose of manufacturing documentation is to provide guidance for the process of replicating the product. Here, companies or project leads may also provide the standard files necessary to operate manufacturing devices such as 3D printers and laser cutting machines.





d /

ories

ories

ories

ories

ories

ories

ories



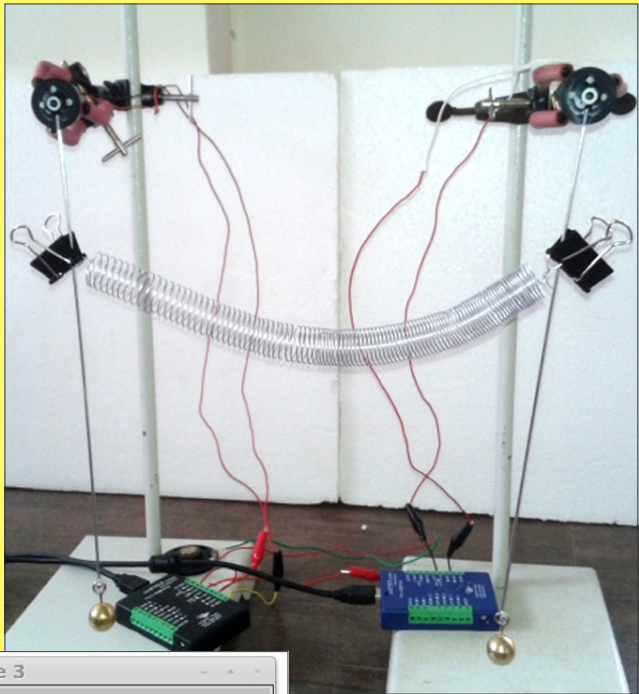
**Figures 4.10**

nasa-jpl-opensource-rover, Apache License 2.0, <http://www.apache.org/licenses/LICENSE-2.0>, <https://github.com/nasa-jpl/opensource-rover>

## EXAMPLE

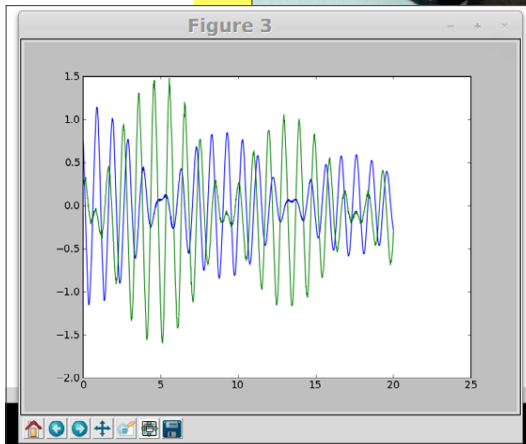
### PSLab behavioural model

Documenting a product's behaviour can also be important for allowing more users to interact with and understand the OSH project. A good example of this is the behavioural analysis of the PSLab oscillator. PSLab's OSH device can be connected to a great number of sensors and measuring devices, and the software interface allows the collection, storage and analysis of the data. The example below displays a simulation of the system's behaviour.



**Figure 4.11**

wavicles-Pocket-Science-Lab, GNU General Public License v3.0, <https://www.gnu.org/licenses/gpl-3.0.html>, The waveforms generated using DC motors after amplification ... <https://github.com/wavicles/Pocket-Science-Lab/blob/master/experiments/Coupled-Pendulum/coupledpendulum.md>



**Figure 4.12**

wavicles-Pocket-Science-Lab, GNU General Public License v3.0, <https://www.gnu.org/licenses/gpl-3.0.html>, Figure 3, <https://github.com/wavicles/Pocket-Science-Lab/blob/master/experiments/Coupled-Pendulum/coupledpendulum.md>

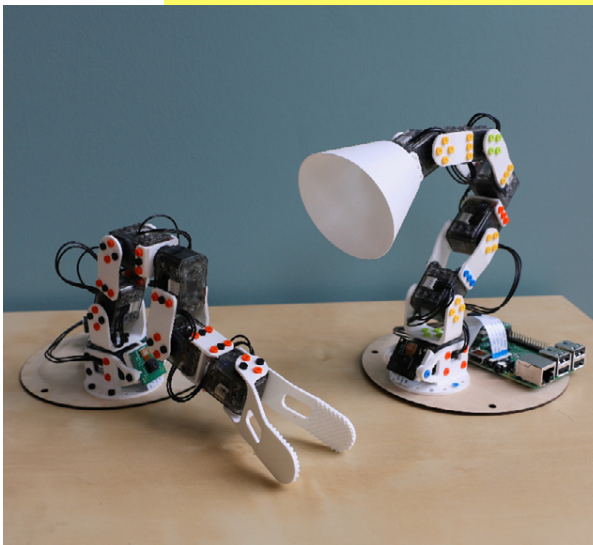
## EXAMPLE

### JPL – Skills and manufacturing roadmap

The JPL Rover repository provides both an overview of skills needed for manufacturing, necessary tools and a roadmap detailing assembly from creating the parts to putting the rover together. JPL Rover also provides CAD files, an online 3D model and a collection of other users' successful builds and modifications.

## Step 5: Assembly

Whilst manufacturing carries great importance in OSH, as most hardware products involve various materials that cannot all be 3D printed, another crucial element of the replication process is how a product is put together. Assembly instructions guide the readers in this process, whilst also giving tips on how to disassemble the product again. The steps necessary for constructing the product are most often explained pedagogically with visual materials, such as photos or videos. In addition, the necessary parts are listed via bills of materials in the



**Figure 4.13**

Poppy projects forked from Jan T. Sott, CC BY-SA 4.0, <https://creativecommons.org/licenses/by-sa/4.0/>, Ergo Jr, <https://github.com/poppy-project/poppy-docs/blob/master/en/assembly-guides/ergo-jr/img/ErgoJr.jpg>

## EXAMPLE

### The Poppy Ergo Jr robot

Information on all steps required to assemble a Poppy Ergo Jr is available on GitHub and includes:

1. Bill of Material (BOM)
2. Electronic assembly
3. Motors configuration
4. Mechanical construction

The assembly sequences are also illustrated by numerous photos and videos that make the process more friendly.

Visit the project here: <https://docs.poppy-project.org/en/assembly-guides/ergo-jr/>.

form

of lists and tables. These lists include the parts references, providers and sometimes prices.

Mount the motor on the 3D printed base.



To do so, prepare 8 small rivets. put the first part in the second part without putting them in the motor. Then, place the motor in the base, with the horn facing up and near the more open side. Use the Ollio to grab a rivet between the first and the second part, then put the rivet in one of the assembly holes. Once the rivet is in place, lock it by pushing the part 1 of the rivet in part 2.

Figure 4.14

Poppy projects forked from Jan T. Solt, CC BY-SA 4.0, <https://creativecommons.org/licenses/by-sa/4.0/>, step\_1-2, [https://github.com/poppy-project/poppy-docs/blob/master/en/assembly\\_guides/ergo-jr/img/assembly\\_steps/step\\_1-2.jpg](https://github.com/poppy-project/poppy-docs/blob/master/en/assembly_guides/ergo-jr/img/assembly_steps/step_1-2.jpg)

## Step 6: Usage

The user guide consists of information that allows end-users to operate the product properly and allows non-technical individuals to pinpoint and solve simple problems without requiring expert assistance. In most cases, these user guides can be found on the website of the projects rather than on the repository platform. Troubleshooting chapters can be very useful if a significant community starts to grow around the project.

## EXAMPLE

### Assembly instructions from SatNOGS Rotator

*Most parts for the Libre Space Foundation's SatNOGS Rotator can be fabricated by 3D printing and laser cutting, whereas the other parts are made from standard aluminium profiles. The assembly instructions are presented in the form of technical drawings with a very precise step-by-step sequence.*

Figure 4.15

SatNOGS Wiki, CC BY-SA 3.0, <https://creativecommons.org/licenses/by-sa/3.0/>, SatNOGS Rotator v3's webpage, [https://wiki.satnogs.org/SatNOGS\\_Rotator\\_v3#Build\\_Sequence](https://wiki.satnogs.org/SatNOGS_Rotator_v3#Build_Sequence)

**Assembly**

Follow the instructions for mechanical assembly and also you can watch timelapse Also, exploded views and instructions are present here.

Prior to Step 8, the rotary encoders must be ready and prior to Step 11 the motor must be mounted in A1070-1. For the rotary encoder section Rotator\_Controller.

Some notes for assembly:

- Resolve collision between of end-stop arm and C1042-1 end-stop mount, community post
- Running the worm gear and lapping the worm gear to get gear box running smooth
- Collision between bolts and motor, this issue can be created by using wrong M4 screws(H1100-5) and washers(H1110-1)

Step 1, Prepare the assembly of worm gear

Step 2, Prepare the assembly of worm gear mount, 2 variants -1 and -2 (mirror)

Step 3, Prepare the assembly of shaft collar for worm wheel

Step 4, Prepare the encoder gear

Step 5, Prepare the Motor mount

Step 6, In case of DC motor configuration

Step 7, Bearing side without encoder and end-stop mounts

Step 8, Bearing side with encoder and end-stop mounts



Pocket Science Labs Documentation

Introduction to PSLab

Introduction

Getting Started

Hardware

PSLab Android App

PSLab Desktop Application

Python Library

Science Experiment Tutorials

Oscilloscope

Multimeter

Logic Analyzer

Sensors

Wave Generator

Power Source

Lux Meter

Accelerometer

Barometer

Compass

Gyroscope

Thermometer

# Oscilloscope

## What Is An Oscilloscope

An instrument used to measure voltage changes over a period of time in real time.



## How To Use It

1. Connect SI1 and SI2 pins on PSLab board to CH1 and CH2 pins respectively for Analog mode or connect SQ1,SQ2,SQ3 pins to CH1, CH2, CH3 pins respectively for Digital mode.
2. Go to Wave Generator instrument in the PSLab Android application.

## EXAMPLE

### PSLab user documentation

*This PSLab OSH electronics kit includes all the necessary information needed to build and further develop the electronics components. Due to the diversity and size of PSLab's community, it is important to have a consistent set of user instructions so that relevant information on how to operate the product properly can easily be retrieved from the community. In the example below, the authors have provided a highly detailed and pedagogical description of the step-by-step operations needed to run the different devices of the kit.*

Visit PSLab to discover more: <https://docs.pslab.io/tutorials/oscilloscope.html>.

Figure 4.16

© Kevin Sheppard, All rights reserved, <https://github.com/bashage/sphinx-material/blob/main/LICENSE.md>, Pocket Science Labs Documentation's webpage, <https://docs.pslab.io/tutorials/oscilloscope.html>

## EXAMPLE

### Poppy Station user forum and troubleshooting

In extensive communities and projects, it can be highly valuable to split the information amongst several channels. Poppy Station has chosen to develop a website for their OSH robots where both user manuals and the history and solutions of troubleshooting are recorded. Furthermore, the website includes a forum for topical discussion, allowing for direct interactions and dynamic cross-fertilisation among the users.

For troubleshooting, appropriate structure and logical wayfinding can make the user experience highly proactive. Poppy Station has chosen to add its troubleshooting log and results as part of its FAQ section on the website, making it easier for users to discover in online searches and via the website.

Visit the forum for more details: <https://poppy.discourse.group/about>.

## Step 7: Maintenance

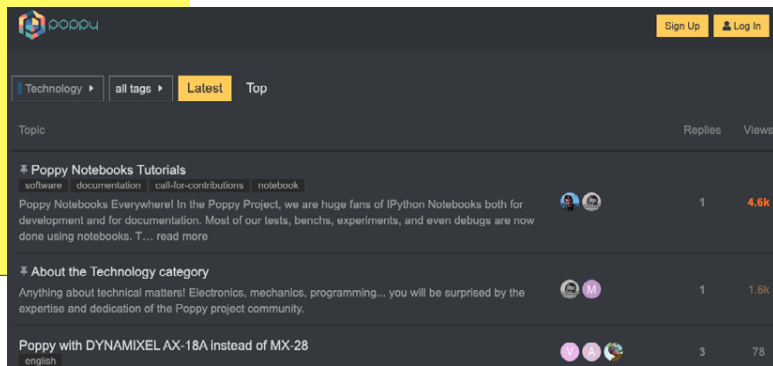
Maintenance is a key element of product performance. A company's ability to ensure the correct operation of a system can be crucial to the safety of the users and others interacting with a product. Specifically, this aspect is an important factor of success for products that have a significant mechanical dimension and that operate in demanding environments, such as exposed outdoor areas. A website forum can sometimes fulfil this need by providing troubleshooting sections for and by users. This, in turn, helps individuals to understand the process of modifying a system or component in order to either correct faults, improve performance and other attributes, or adapt to a changing environment.

## Troubleshooting

### In this section

- Problem 1: `poppy.local` does not answer.
- 🛠 Diagnostic procedure
  - I. Make a ping test
  - II.A. If the ping test succeeds
  - II.B. If the ping test fails
- Problem 2: My robot makes shaky motions
- Problem 3: I can see frequent errors about the `pos/speed/load sync loop`
- Problem 4: Motors are not detected
- Problem 5: My robot makes weird motions or auto-collides
- Problem 6: What is the default SSH password of my robot?
- Problem 7: The STL files seem corrupted

### Problem 1: `poppy.local` does not answer.



Figures 4.17

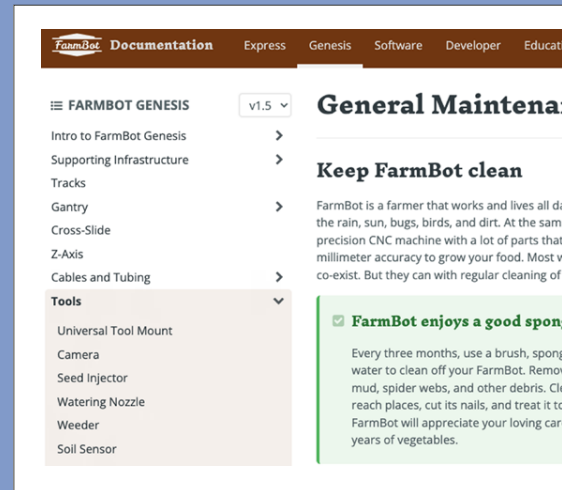
Poppy Forum, CC BY-NC-SA 3.0, <https://creativecommons.org/licenses/by-nc-sa/3.0/deed.en>, Poppy Forum's webpage, <https://poppy.discourse.group/about>

## Step 8: Disposal

With the pending climate crisis calling for more circularity and sustainability practices in hardware, the end-of-life of an open source product becomes a key consideration for companies to include in both development and documentation. This section of the documentation enables users to understand the process of discarding all or part of the product after it has reached its end-of-life or is no longer needed. This aspect is often forgotten, and, when addressed, end-of-life considerations often focus on what comes after the design process, rather than looking at the design process itself.

Although some projects are developed specifically to leverage hardware production waste, such as Precious Plastic<sup>54</sup> (which builds and enables the development of plastic recycling machines), seeking to minimise waste and create more environmentally friendly products as part of the initial design process remains a marginal practice in OSH like in other industrial contexts. As such, this final step of the documentation practice aims to provoke and influence designers into anticipating the end-of-life of their products, which may result in iterations to materials or practices from the beginning to the end of the design process. Key elements to take into consideration are whether the product is recyclable or not, under what conditions and assessment of environmental impact.

54 <https://preciousplastic.com/> (accessed 8 November 2023).



## EXAMPLE

### The FarmBot project

*FarmBot hardware has reached a maturity and level of compliance that allows it to be manufactured, shipped and sold all over the world. Besides providing source files for the product and instructions for operating it, the project also includes a maintenance guide. The FarmBot forum allows users to interact and share use cases, as well as tips and tricks for product longevity.*

*Visit the forum to find out more:  
<https://genesis.farm.bot/v1.5/Extras/maintenance>.*



ay and night outside with  
e time, FarmBot is a  
allow it to move with  
ould say these things can't  
your device.

ge bath

ge, broom, and a little  
ve built up bird poop,  
ean out the hard to  
a massage. Your  
e and pay you back with

Figure 4.18

FarmBot, CC0 1.0, <https://creativecommons.org/publicdomain/zero/1.0/>, FarmBot's web page: <https://genesis.farm.bot/v1.5/extras/maintenance/general-maintenance.html>

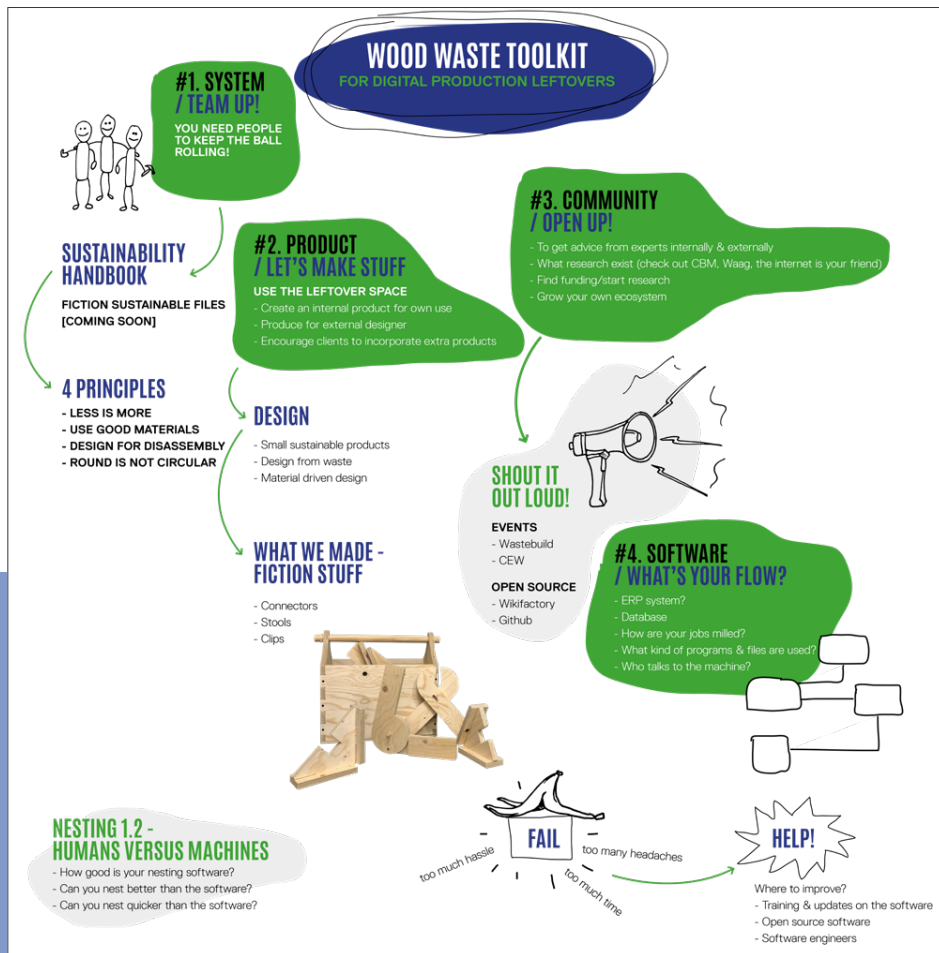


Figure 4.19

Fiction Factory, CC BY 4.0 – rearranged, Wood waste Toolkit by Fiction Factory, <https://wikifactory.com/+fictionfactory/wood-waste-toolkit/file/WoodWasteToolkit.pdf>

**EXAMPLE**

The wood waste tool kit by Fiction Factory

Fiction Factory studied their wood waste and leftovers from production projects to understand how to make production practices even more circular, more efficient and to reduce the quantities of leftover materials. This resulted in the creation of a wood waste toolkit to help designers and craftsmen operating on CNC machines to reduce the wood waste during production. The toolkit is a software that works by analysing the leftover space digitally before production of an already defined product template and using the analysis to define new and smaller parts that can be nested along with the original template. The smaller parts themselves become a new design developed for the occasion and based on the particular leftover space.

For more information: <https://www.fictionfactory.nl/>

# Documentation in practice

Documentation is an iterative process that accompanies the development process and continues during the life cycle of the product. As open source projects are often open-ended endeavours, unlike industrial projects that may have a fixed duration, it is important to recognise that the motivation for documentation is related to the motivation to participate in the project itself. Recent work has shown that participation in OSH projects is motivated by reputation and learning,<sup>55</sup> as well as the fact that most OSH participants are often users of the products. These are positive factors that advocate for the increased use of documentation. Nonetheless, the process of documentation remains painful and cumbersome in the eyes of many OSH participants, and, as a result, many are reluctant to do it.

The methodology below offers an easy and comprehensible point of entry to the process of documentation that allows both project managers and participants to facilitate contributions. This methodology in combination with the eight-step structure makes up the OSH documentation tool that this chapter explores.

## Define the aim of documentation

The documentation process should be motivated by a specific goal. This goal can evolve along the design process and may act as a compass for the authors of the documentation. For example, at the initial stage of the project, a company or a project initiator may be interested in collecting ideas and concepts from contributors by publicising the

project. The documentation should then contain the problem definition and the context, along with some technical elements in the form of a functional analysis that allows the potential contributors to generate ideas in the form of sketches, simplistic 3D models or simple text descriptions. As the project progresses, the interest may shift towards attracting expertise in a specific technical domain to strengthen the development of the product or help to solve a technical issue. Then, clear documentation explaining the problem and detailing the requirements can be good practice. In other words, the evolution of the focus for the documentation closely follows the evolution of the project.

Additionally, the openness of a project (and consequently the level and type of documentation) is directly linked to the business model. For example, if a company develops and commercialises a proprietary product design with the desire to open up certain aspects of the hardware, the documentation must take this factor into account and concentrate on the information related to the open part of the product. Therefore, it is important to have a clear vision of the aim and current stage of the project before structuring documentation, looking at openness, desired contributions and long-term development goals.

55 Hausberg, J. P., & Spaeth, S. (2020). Why makers make what they make: motivations to contribute to open source hardware development. *R&D Management*, 50(1), pp. 75–95.



## Uncover the current design step

The documentation follows the logical flow of a design process flow, and, as such, if starting a new project from scratch, it is useful to first clarify what step of the design process the project has reached. It is inconvenient to create a user manual before having designed and tested the product, for example. However, if a company is planning to wait with opening up a product until after the design and prototyping phases, it is worth asking how to document previous phases (especially the functional ones) according to the objectives of opening up. A company can also envisage a reverse documentation process by starting from the current stage of the project development and retracing previous phases.

To sum up, with a relatively clear understanding of the aims and objectives of the project, as well as the current phase of the development, it is more feasible for a company to clearly scope and plan the documentation process.

## Specify the appropriate sections to document

The choice and usability of the activities for a company or a community differ according to the aim of the project, the level of maturity (i.e., the design phase) and the business model envisaged. It is therefore unrealistic for practitioners to engage in a rigid step-by-step documentation process. Rather, there are important sets of development activities that correspond to different levels of maturity of the projects (see list below). For example, if a project is still in the ideation phase, it is more likely that the documentation will only cover the Project Introduction and Specification. In comparison, if a project is at “Step 3: Concept Development” of the model development process introduced earlier (see Figure 4.3), it is important to allow technical contributions to be documented by

including the Project Introduction, Specification, Design, Manufacturing and Assembly sections.

The standard activities listed below have been defined based on an analysis of OSH projects coupled with best-practice design models from conventional engineering. They are displayed according to the development steps proposed in Figure 4.3:

### 1. Ideation

- *Problem description:* 1. Project Introduction, 2. Specification
- *Idea contribution:* 1. Project Introduction, 2. Specification, 3. Design

### 2. Specification

- *Functional specification and requirement collection:* 1. Project Introduction, 2. Specification, 3. Design

### 3. Concept development

- *Technical contribution:* 1. Project Introduction, 2. Specification, 3. Design, 4. Manufacturing, 5. Assembly

### 4. Product development and prototyping

- *Prototyping and testing:* 1. Project Introduction, 2. Specification, 3. Design, 4. Manufacturing, 5. Assembly
- *Customisation:* 1. Project Introduction, 2. Specification, 3. Design, 4. Manufacturing, 5. Assembly

### 5. Produce and replicate

- *Replicate / Produce:* 1. Project Introduction, 4. Manufacturing, 5. Assembly
- *Using the product:* 1. Project Introduction, 6. Usage
- *Service / Maintenance:* 3. Design, 4. Manufacturing, 5. Assembly, 7. Maintenance
- *Dismantle for recycling or disposal:* 4. Manufacturing, 5. Assembly, 8. Disposal

## How to populate the documentation sections?

Once the most important and relevant documentation sections have been identified, the next step is to write and develop the documentation itself. Two main principles inhabit OSH development, namely transparency and replicability. These two overarching purposes define and categorise the documentation steps, making it clearer for OSH practitioners what steps to focus on when addressing one or the other.

### Transparency

Full transparency is necessary to ensure that newcomers and contributors can study and modify the content of the design. The modification can be either a contribution to the initial design or a reuse of the design for another purpose. The presence of the following three sections should therefore be included in a project's documentation guidelines when seeking out transparency:

6. *Introduction*
7. *Specifications*
8. *Design*

### Replicability

The freedom of replicability and the right to repair are undeniably linked to OSH. Consequently, in order to ensure replicability without necessarily allowing for modifications, it is important that the following sections are present in the documentation:

9. *Manufacturing*
10. *Assembly*
11. *Usage*
12. *Maintenance*
13. *Disposal*

# Outro: Prosperity through paper trails

As readers will find, documentation is highly relevant to the success of an OSH project, and, as such, this chapter cannot stand alone. Documentation will undeniably be connected to the motivation and aim of the project, but more importantly, it will be ruled by the business model and the openness strategy of the project. Opening the source of a project is often seen by project owners as a potential threat to their activities. However, this is necessary when looking to invite collaboration into a project, as outlined in Chapters 6 and 7. The adoption of open source should be part of a clear strategy that takes into account the value added by mass participation and open access to a product's documentation. A company or project owner should therefore design the documentation so that the amount of available information is consistent with the strategic approaches defined in the project's business model (as specified in Chapter 3). Furthermore, whilst documentation is relevant to ensuring the adaptation and innovation of a product, the question of how to licence it holds equal importance (as described in Chapter 5), and should be compliant with the strategic approaches. In other words, companies should not forget that licensing and business development make up the strategic purpose of the documentation. Documentation can therefore be seen as the consequence of the two.