HELMHOLTZ
Metadata Collaboration

Best Practice Made Easy:
Deploying Tools for FAIR Research Software Development

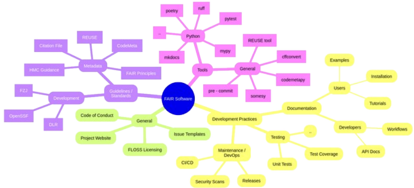
Anton Pirogov https://orcid.org/0000-0002-5077-7497
Mustafa Soylu https://orcid.org/0000-0003-2637-0432
Volker Hofmann https://orcid.org/0000-0002-5149-603X
Stefan Sandfeld https://orcid.org/0000-0001-9560-4728

HMC FAIR Data Commons / Forschungszentrum Jülich https://ror.org/02nv7yv05

07.03.2024, deRSE24

www.helmholtz-metadaten.de

Overview



Motivation: The State of RSE

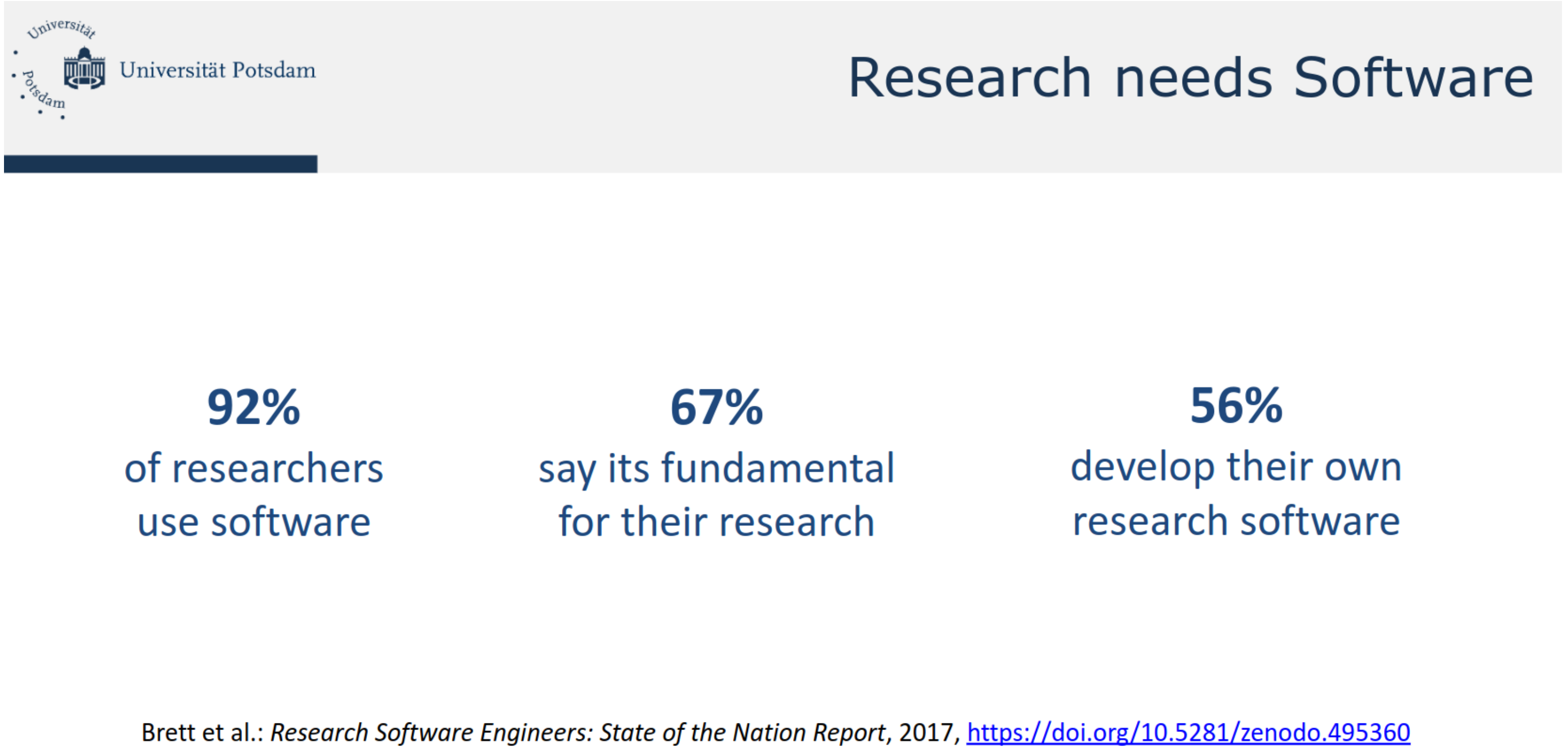
FAIR PYTHON COOKIECUTTER

SOMESY

Motivation

Idiosyncrasies of Research Software Development

Importance of Software for Modern Research



Universität Potsdam

Research needs Software

92%
of researchers
use software

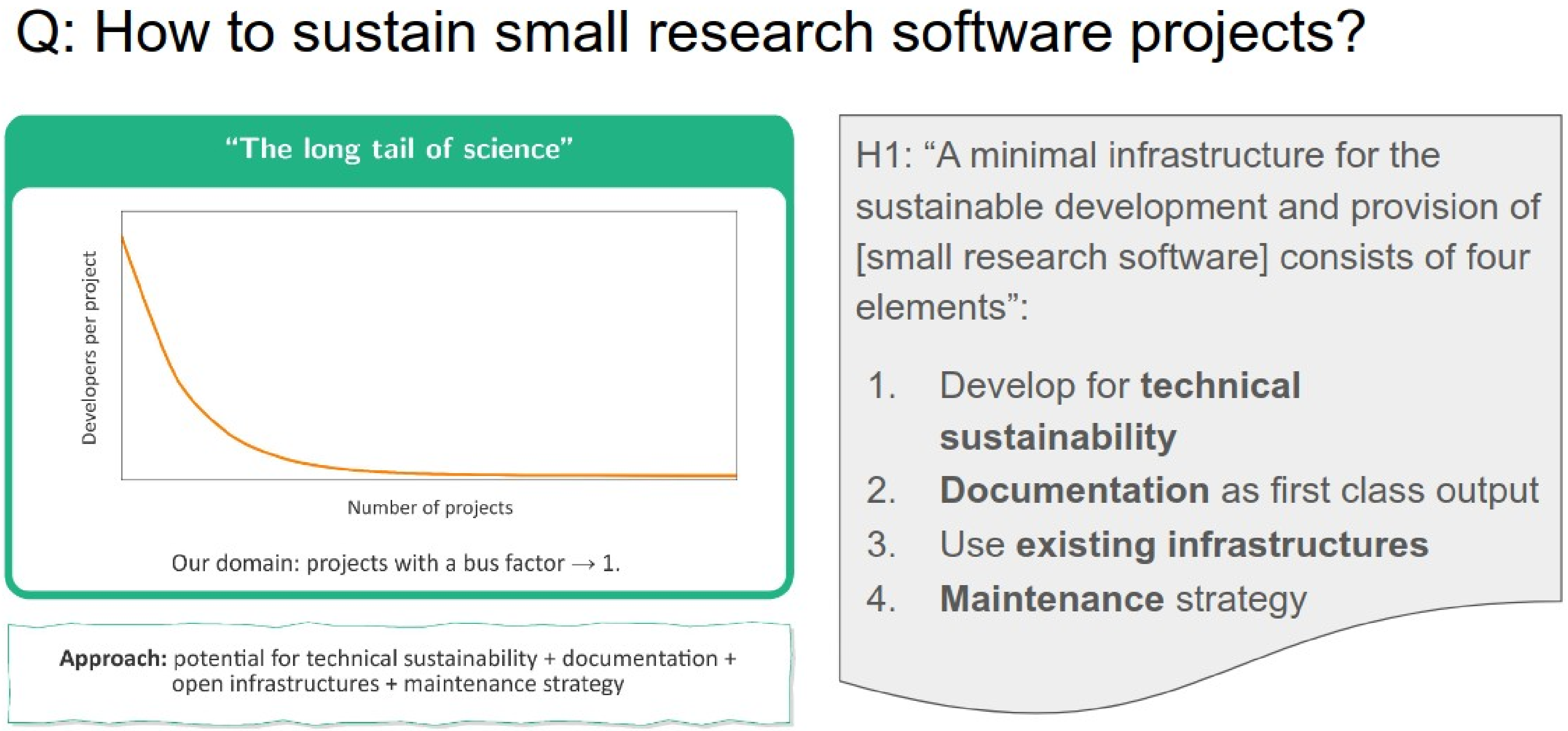
67%
say its fundamental
for their research

56%
develop their own
research software

Brett et al.: *Research Software Engineers: State of the Nation Report*, 2017, https://doi.org/10.5281/zenodo.495360

Slide from talk: “Research Software, Software Research, and more”
HiRSE Seminar 18.09.2023, By Anna-Lena Lamprecht, Uni Potsdam

The “Long Tail” of Research Software Projects



Q: How to sustain small research software projects?

“The long tail of science”

Developers per project
Number of projects

Our domain: projects with a bus factor → 1.

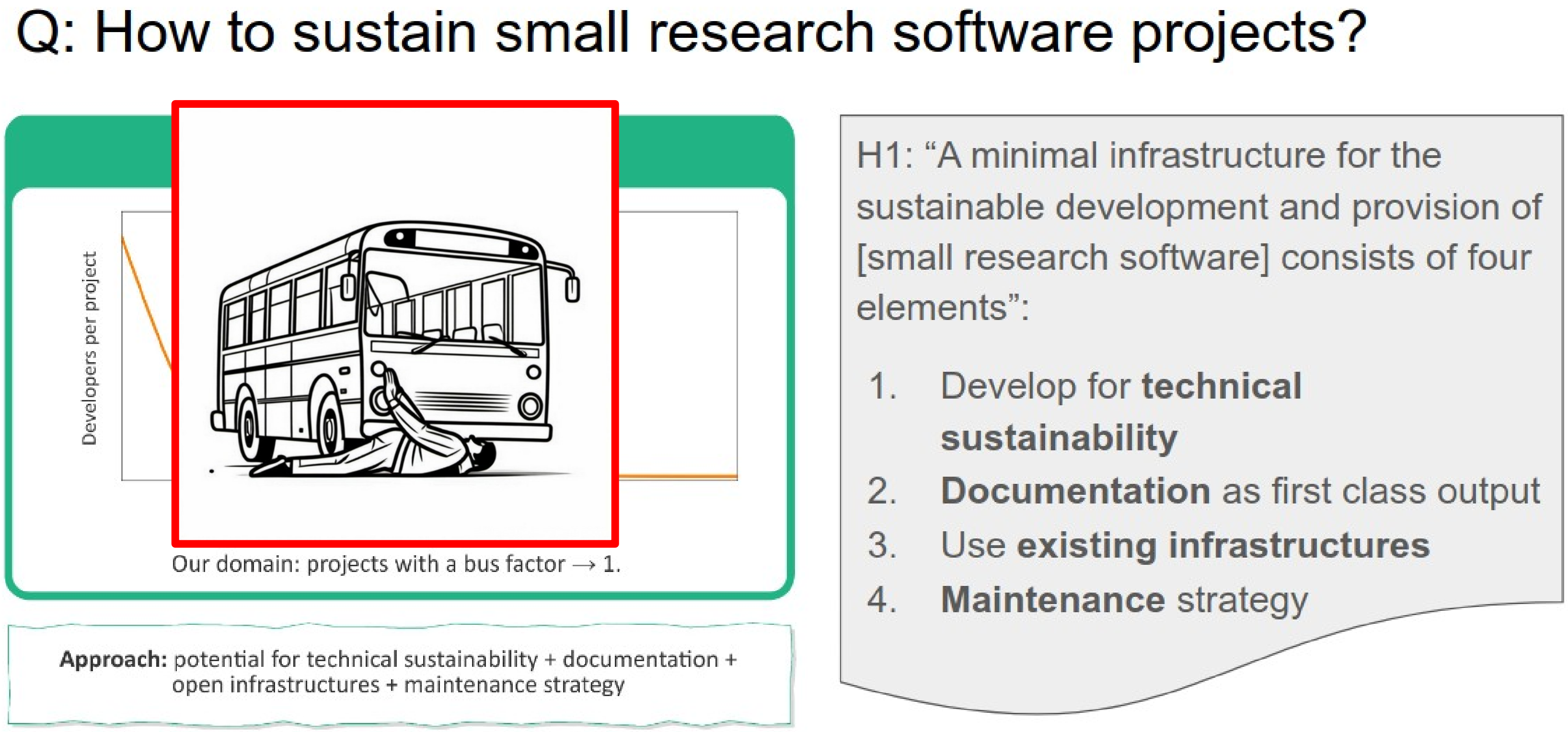
Approach: potential for technical sustainability + documentation + open infrastructures + maintenance strategy

H1: “A minimal infrastructure for the sustainable development and provision of [small research software] consists of four elements”:

1. Develop for **technical sustainability**
2. **Documentation** as first class output
3. Use **existing infrastructures**
4. **Maintenance** strategy

Slide from talk: “The four elements of achieving research software sustainability for long tail projects” deRSE 2023 Conference, Stephan Druskat & Thomas Krause

The “Long Tail” of Research Software Projects



Q: How to sustain small research software projects?

Developers per project

Our domain: projects with a bus factor → 1.

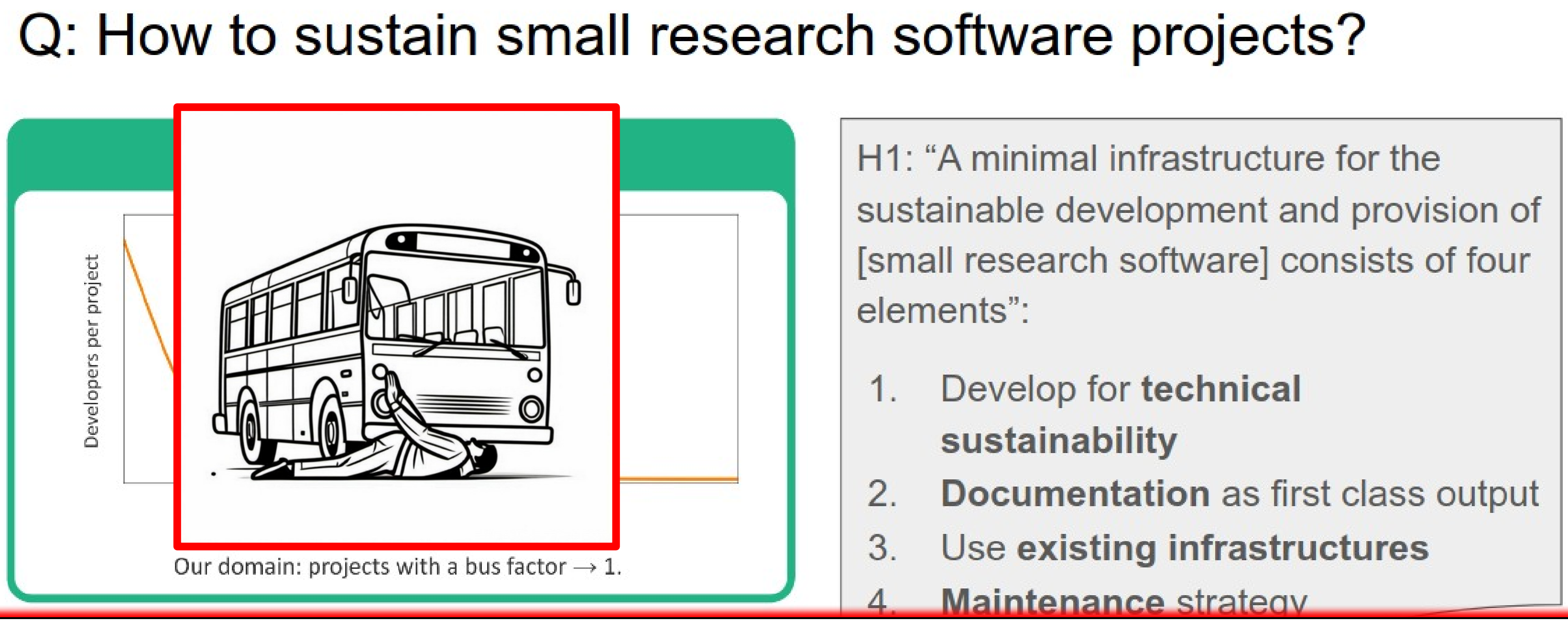
Approach: potential for technical sustainability + documentation + open infrastructures + maintenance strategy

H1: “A minimal infrastructure for the sustainable development and provision of [small research software] consists of four elements”:

1. Develop for **technical sustainability**
2. **Documentation** as first class output
3. Use **existing infrastructures**
4. **Maintenance** strategy

Slide from talk: “The four elements of achieving research software sustainability for long tail projects” deRSE 2023 Conference, Stephan Druskat & Thomas Krause

The “Long Tail” of Research Software Projects



Q: How to sustain small research software projects?

Our domain: projects with a bus factor → 1.

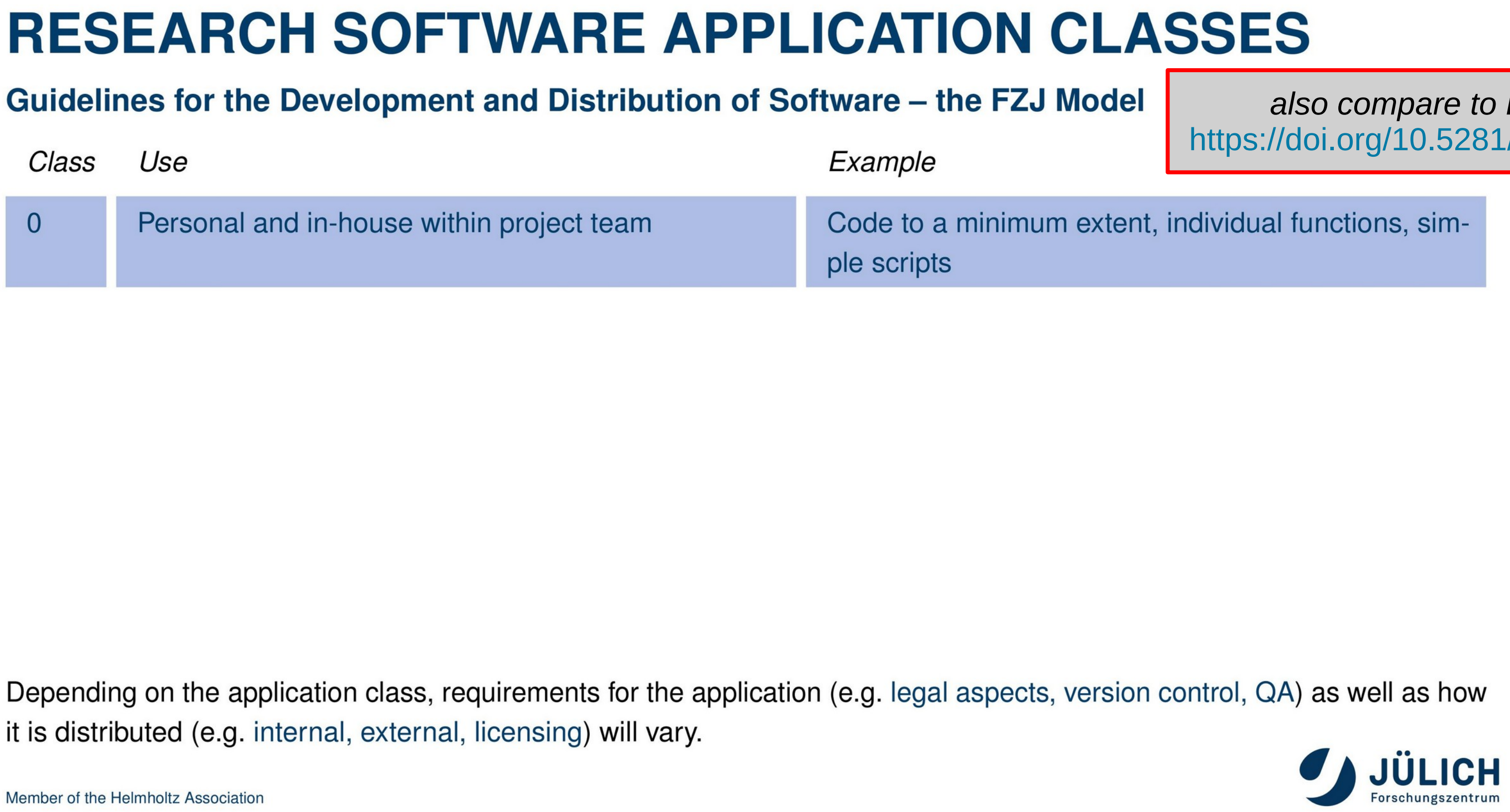
H1: “A minimal infrastructure for the sustainable development and provision of [small research software] consists of four elements”:

1. Develop for **technical sustainability**
2. **Documentation** as first class output
3. Use **existing infrastructures**
4. **Maintenance** strategy

Technical sustainability (a.k.a. follow good practice)

Slide from talk: “The four elements of achieving research software sustainability for long tail projects”
deRSE 2023 Conference, Stephan Druskat & Thomas Krause

Classifying Different Types of Research Software



RESEARCH SOFTWARE APPLICATION CLASSES

Guidelines for the Development and Distribution of Software – the FZJ Model

Class	*Use*	*Example*
0	Personal and in-house within project team	Code to a minimum extent, individual functions, simple scripts

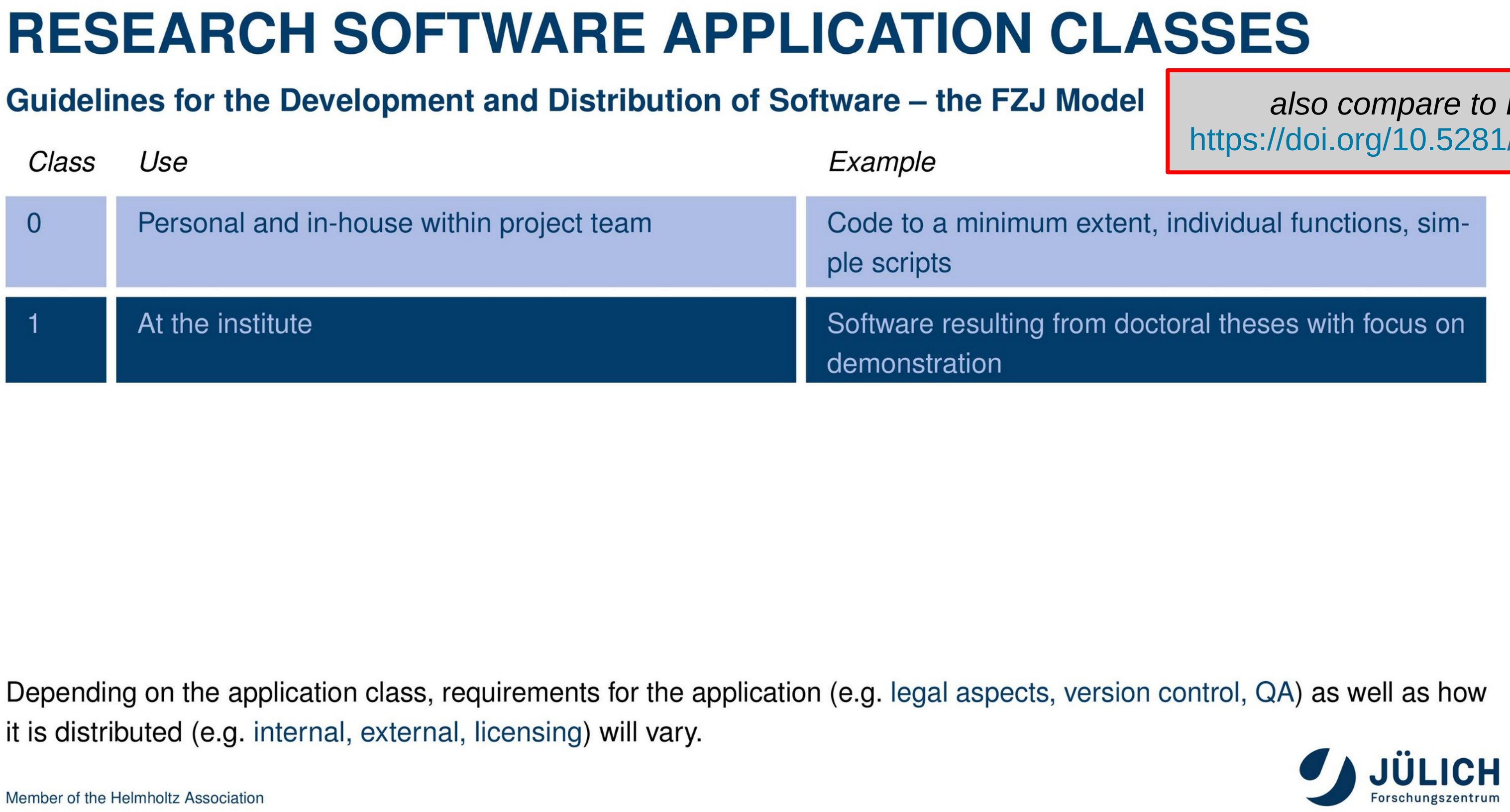
Depending on the application class, requirements for the application (e.g. legal aspects, version control, QA) as well as how it is distributed (e.g. internal, external, licensing) will vary.

Member of the Helmholtz Association

JÜLICH Forschungszentrum

also compare to DLR model:
https://doi.org/10.5281/zenodo.1344612

Classifying Different Types of Research Software



RESEARCH SOFTWARE APPLICATION CLASSES

Guidelines for the Development and Distribution of Software – the FZJ Model

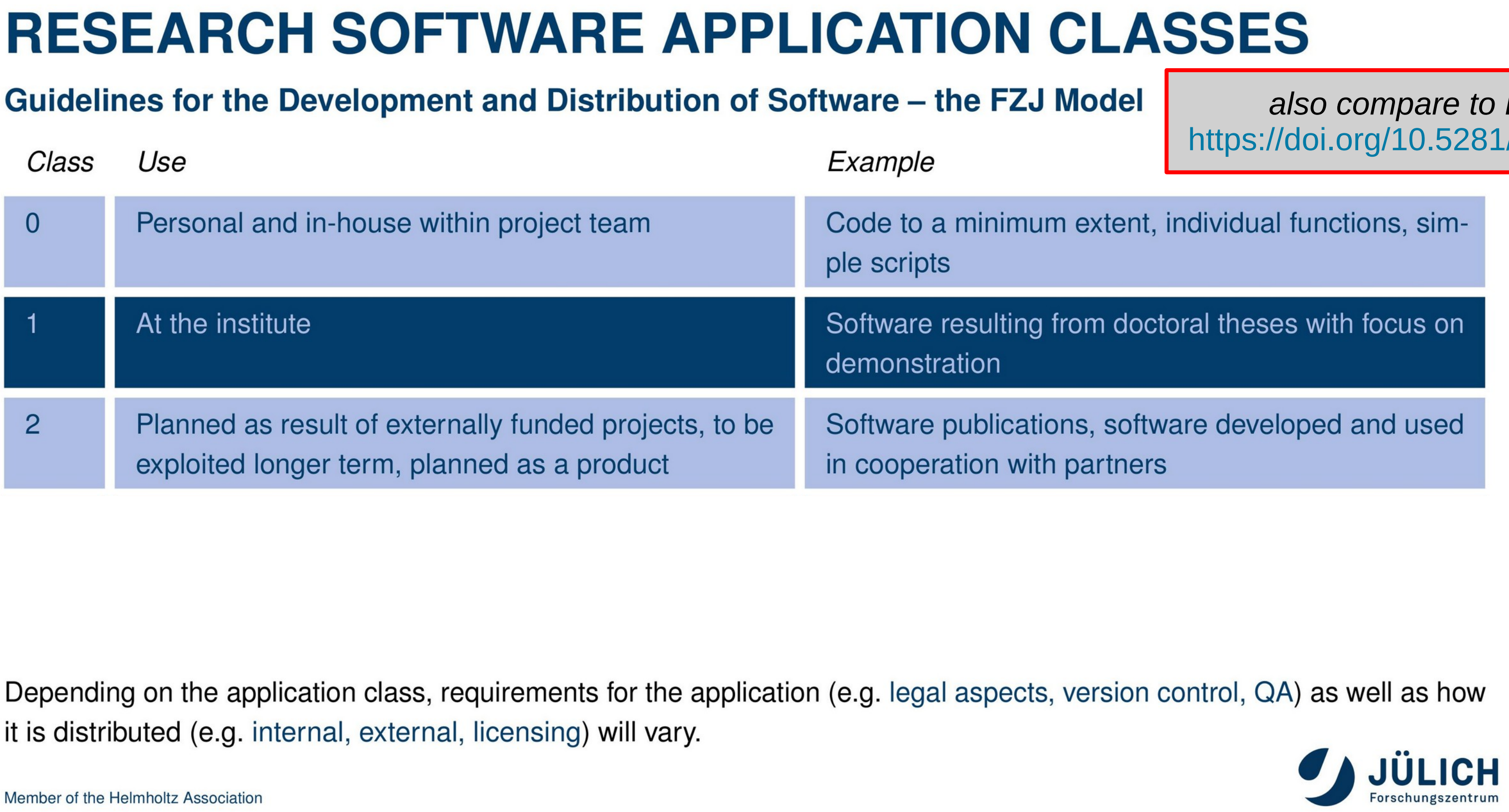
Class	*Use*	*Example*
0	Personal and in-house within project team	Code to a minimum extent, individual functions, simple scripts
1	At the institute	Software resulting from doctoral theses with focus on demonstration

Depending on the application class, requirements for the application (e.g. legal aspects, version control, QA) as well as how it is distributed (e.g. internal, external, licensing) will vary.

Member of the Helmholtz Association

also compare to DLR model:
https://doi.org/10.5281/zenodo.1344612

Classifying Different Types of Research Software



RESEARCH SOFTWARE APPLICATION CLASSES

Guidelines for the Development and Distribution of Software – the FZJ Model

Class	*Use*	*Example*
0	Personal and in-house within project team	Code to a minimum extent, individual functions, simple scripts
1	At the institute	Software resulting from doctoral theses with focus on demonstration
2	Planned as result of externally funded projects, to be exploited longer term, planned as a product	Software publications, software developed and used in cooperation with partners

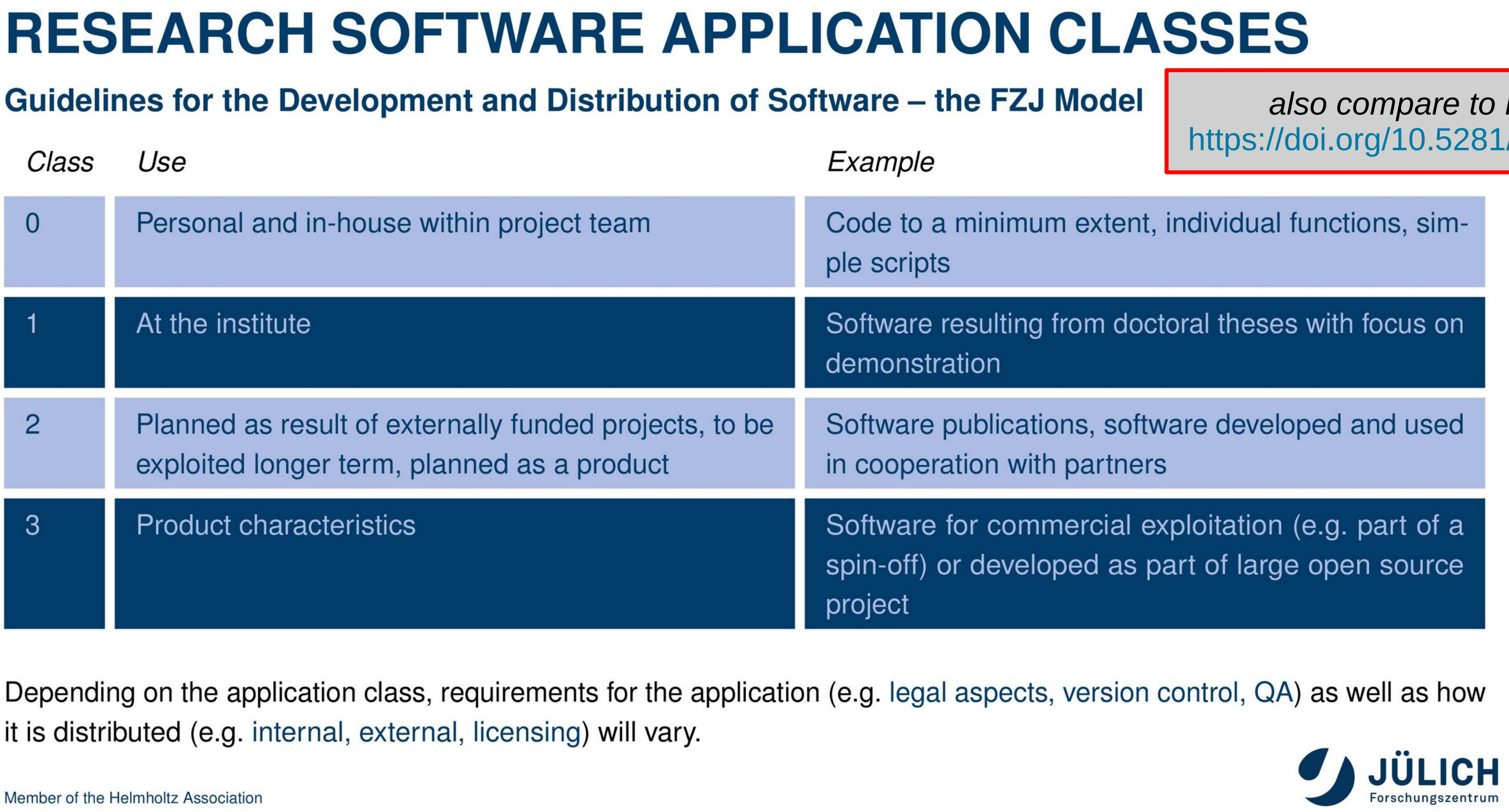
Depending on the application class, requirements for the application (e.g. legal aspects, version control, QA) as well as how it is distributed (e.g. internal, external, licensing) will vary.

Member of the Helmholtz Association

also compare to DLR model:
https://doi.org/10.5281/zenodo.1344612

Classifying Different Types of Research Software

also compare to DLR model:
https://doi.org/10.5281/zenodo.1344612



RESEARCH SOFTWARE APPLICATION CLASSES

Guidelines for the Development and Distribution of Software – the FZJ Model

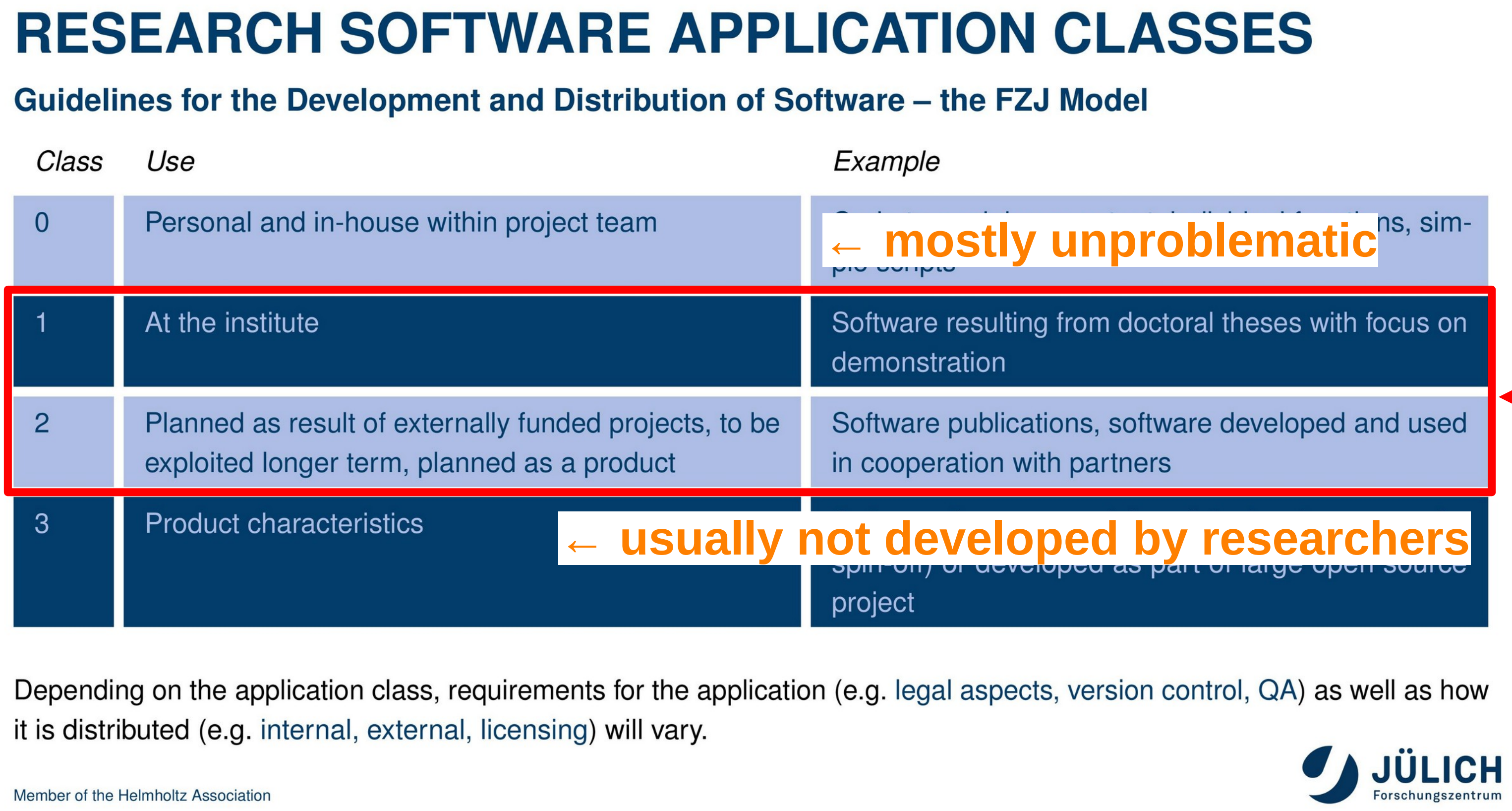
Class	*Use*	*Example*
0	Personal and in-house within project team	Code to a minimum extent, individual functions, simple scripts
1	At the institute	Software resulting from doctoral theses with focus on demonstration
2	Planned as result of externally funded projects, to be exploited longer term, planned as a product	Software publications, software developed and used in cooperation with partners
3	Product characteristics	Software for commercial exploitation (e.g. part of a spin-off) or developed as part of large open source project

Depending on the application class, requirements for the application (e.g. legal aspects, version control, QA) as well as how it is distributed (e.g. internal, external, licensing) will vary.

Member of the Helmholtz Association

JÜLICH
Forschungszentrum

Classifying Different Types of Research Software



RESEARCH SOFTWARE APPLICATION CLASSES

Guidelines for the Development and Distribution of Software – the FZJ Model

Class	*Use*	*Example*
0	Personal and in-house within project team	ns, simple scripts
1	At the institute	Software resulting from doctoral theses with focus on demonstration
2	Planned as result of externally funded projects, to be exploited longer term, planned as a product	Software publications, software developed and used in cooperation with partners
3	Product characteristics	spin-off) or developed as part of large open source project

Depending on the application class, requirements for the application (e.g. legal aspects, version control, QA) as well as how it is distributed (e.g. internal, external, licensing) will vary.

Member of the Helmholtz Association

JÜLICH Forschungszentrum

← mostly unproblematic ✓

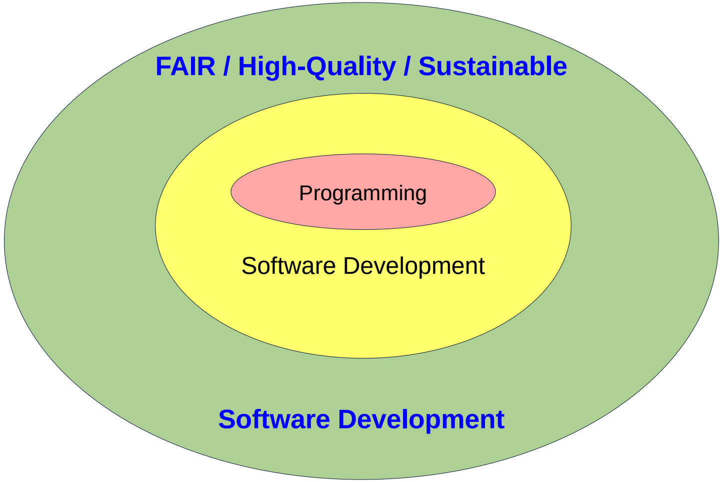
←!!?

← usually not developed by researchers ✓

Sustainable Software Development ≠ Programming

A lot of

- *knowledge and experience* is required
- *big + small decisions* must be made
- *pieces of technology* must be combined



Sustainable Software Development ≠ Programming

A lot of

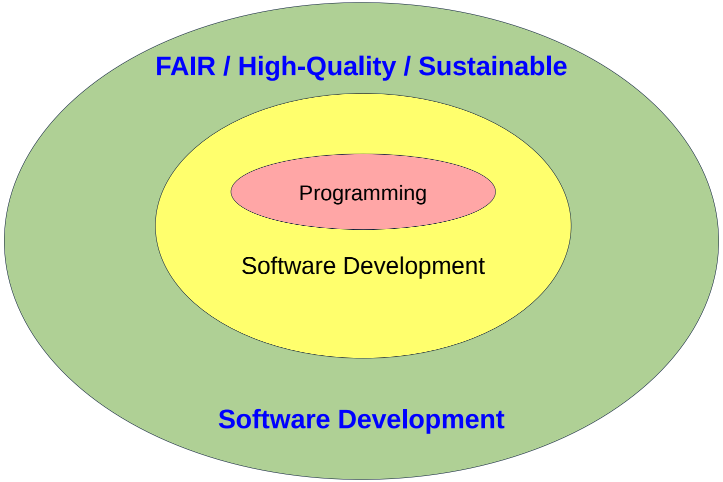
- *<u>knowledge and experience</u>* is required
- *<u>big + small decisions</u>* must be made
- *<u>pieces of technology</u>* must be combined

Surely, **it can be done, but…**

does a typical domain scientist have enough...

- ... **motivation** to research good development and metadata practices?
- ... **spare time** on the side to ponder these issues and evaluate options?
- ... **experience** to make good technical choices?

No!



Sustainable Software Development ≠ Programming

1) Most **tools** in research are created **by researchers**, for their own work
2) The job of (most) researchers is **doing research**, not writing software

Sustainable Software Development ≠ Programming

1) Most **tools** in research are created **by researchers**, for their own work
2) The job of (most) researchers is **doing research**, not writing software

Bottom line:

We cannot + should **not** expect

that researchers want (or can) turn into software developers!

Sustainable Software Development ≠ Programming

1) Most **tools** in research are created **by researchers**, for their own work

2) The job of (most) researchers is **doing research**, not writing software

Bottom line:

We cannot + should **not** expect

that researchers want (or can) turn into software developers!

So how do we get more FAIR and robust research software?

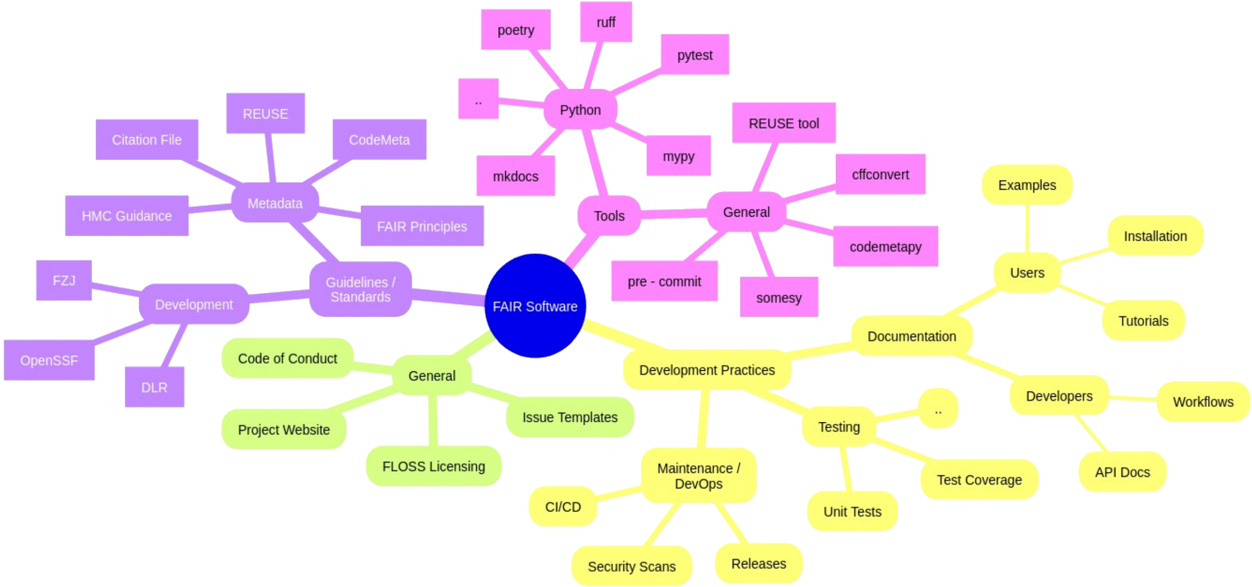
→ we should try to *make „doing it right“* **as easy as possible**!

Deploying Best Practices

FAIR Python Cookiecutter Template

Deploying Best Practices in the Form of a Template

Why not **provide a ready-to-use template** showing how it can be done?



A template that contains

- Condensed hands-on *“expert knowledge”* and guidance
- Detailed explanations that help to *“learn the skills”*

Feature Overview

- **generic**, directly **(re)usable** template repository
- aligned with **recommendations** from various sources
- **saves** a lot of **time** for researchers starting a software project

Feature Overview

- **generic**, directly **(re)usable** template repository
- aligned with **recommendations** from various sources
- **saves** a lot of **time** for researchers starting a software project

Combines many tools, covering:

- **development**, dependency management, versioning, **testing**/QA
- **continuous integration** (GitHub + GitLab), **releases**…
- user + contributor **documentation** (project website/landing page)
- maintenance of rich, consistent + up-to-date software project **metadata***

(*powered by SOMESY)

Feature Overview

- **generic**, directly **(re)usable** template repository
- aligned with **recommendations** from various sources
- **saves** a lot of **time** for researchers starting a software project

Combines many tools, covering:

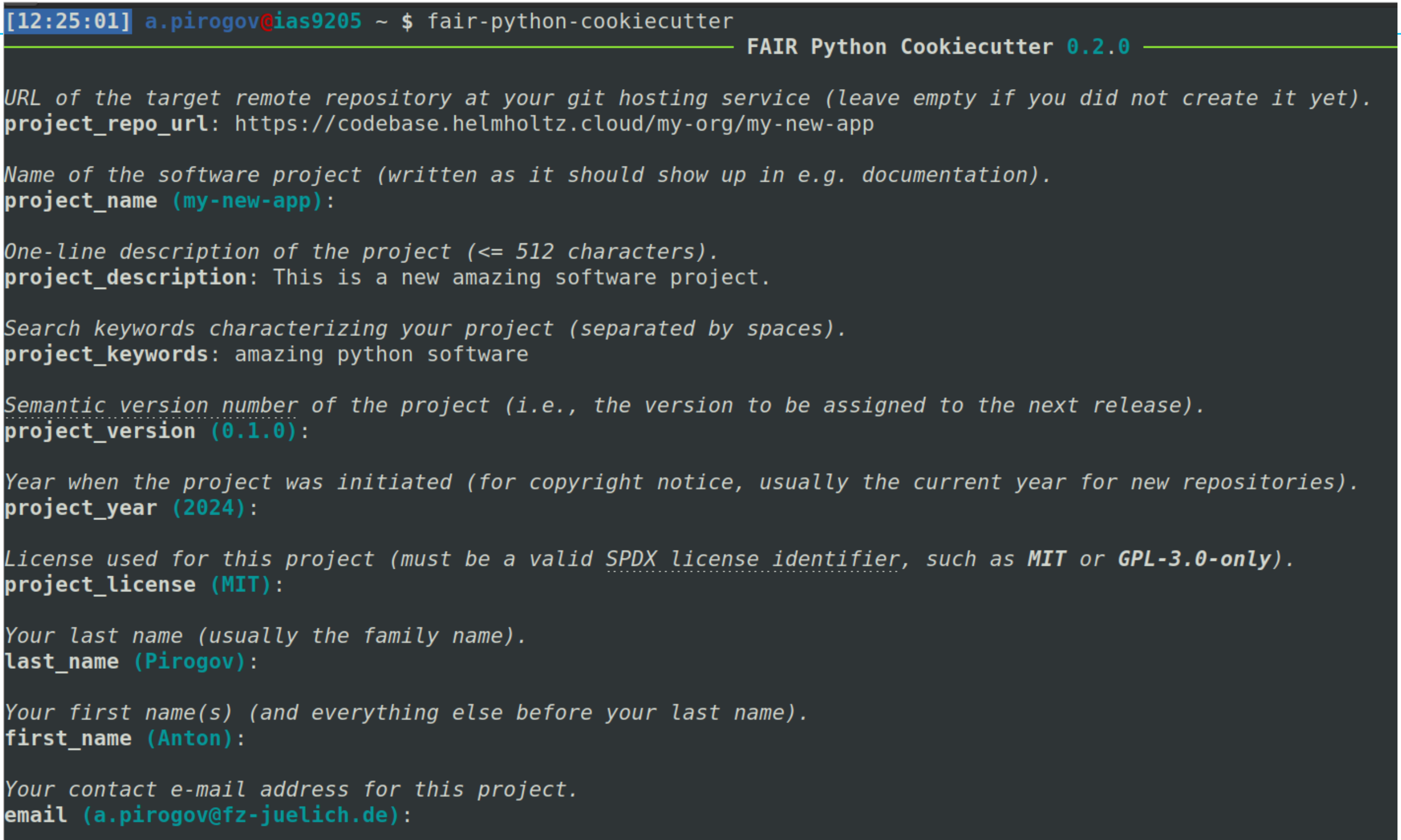
- **development**, dependency management, versioning, **testing**/QA
- **continuous integration** (GitHub + GitLab), **releases**…
- user + contributor **documentation** (project website/landing page)
- maintenance of rich, consistent + up-to-date software project **metadata***

(*powered by SOMESY)

Self-documenting:

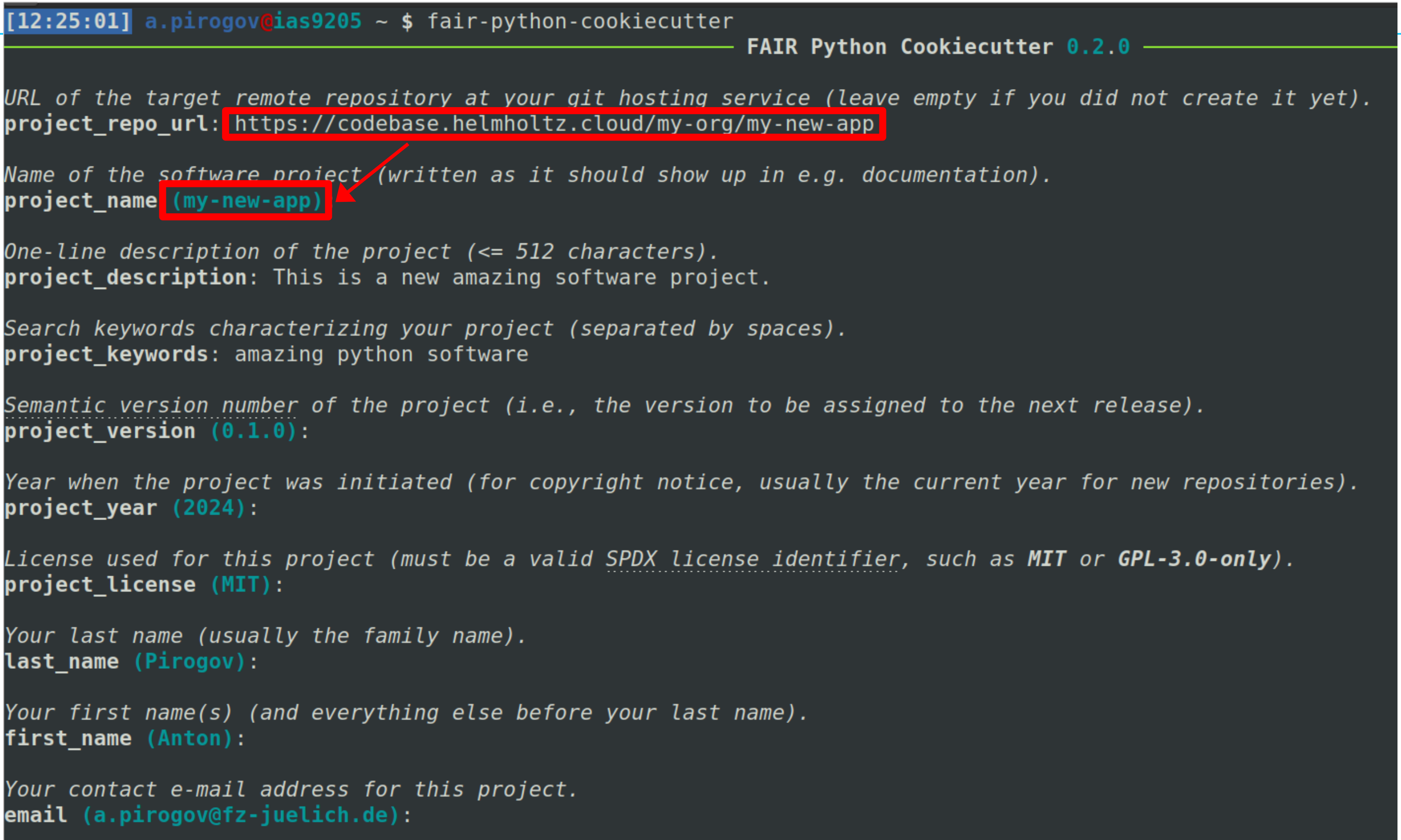
- many comments, explanations, pointers to other resources
- usable as demonstrator + educational resource

Feature Overview



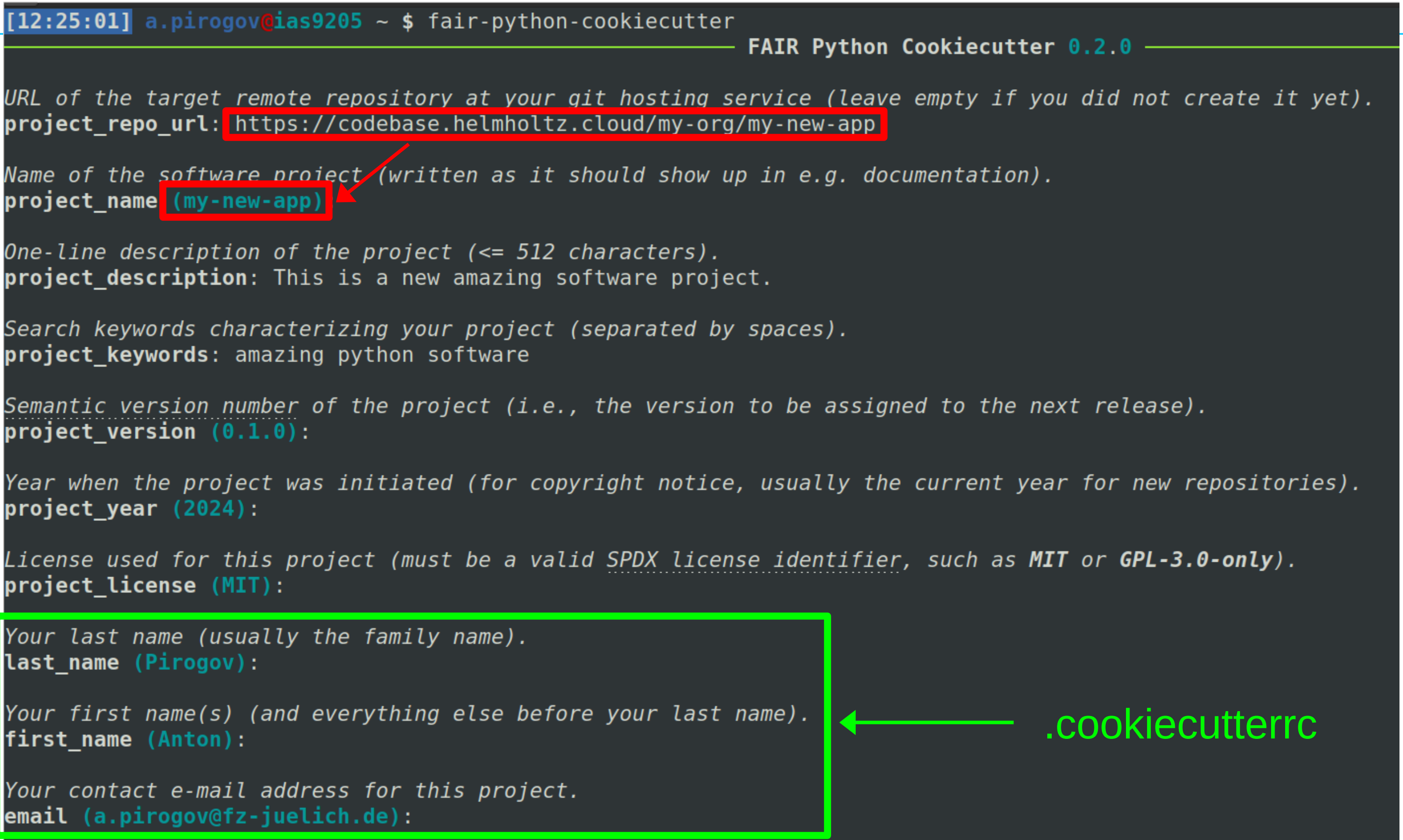
```
[12:25:01] a.pirogov@ias9205 ~ $ fair-python-cookiecutter
──────────────────────────────────────────────── FAIR Python Cookiecutter 0.2.0 ────────────────

URL of the target remote repository at your git hosting service (leave empty if you did not create it yet).
project_repo_url: https://codebase.helmholtz.cloud/my-org/my-new-app

Name of the software project (written as it should show up in e.g. documentation).
project_name (my-new-app):

One-line description of the project (<= 512 characters).
project_description: This is a new amazing software project.

Search keywords characterizing your project (separated by spaces).
project_keywords: amazing python software

Semantic version number of the project (i.e., the version to be assigned to the next release).
project_version (0.1.0):

Year when the project was initiated (for copyright notice, usually the current year for new repositories).
project_year (2024):

License used for this project (must be a valid SPDX license identifier, such as MIT or GPL-3.0-only).
project_license (MIT):

Your last name (usually the family name).
last_name (Pirogov):

Your first name(s) (and everything else before your last name).
first_name (Anton):

Your contact e-mail address for this project.
email (a.pirogov@fz-juelich.de):
```

Feature Overview



```
[12:25:01] a.pirogov@ias9205 ~ $ fair-python-cookiecutter
────────────────────────────────────────────── FAIR Python Cookiecutter 0.2.0 ──────────────────

URL of the target remote repository at your git hosting service (leave empty if you did not create it yet).
project_repo_url: https://codebase.helmholtz.cloud/my-org/my-new-app

Name of the software project (written as it should show up in e.g. documentation).
project_name (my-new-app)

One-line description of the project (<= 512 characters).
project_description: This is a new amazing software project.

Search keywords characterizing your project (separated by spaces).
project_keywords: amazing python software

Semantic version number of the project (i.e., the version to be assigned to the next release).
project_version (0.1.0):

Year when the project was initiated (for copyright notice, usually the current year for new repositories).
project_year (2024):

License used for this project (must be a valid SPDX license identifier, such as MIT or GPL-3.0-only).
project_license (MIT):

Your last name (usually the family name).
last_name (Pirogov):

Your first name(s) (and everything else before your last name).
first_name (Anton):

Your contact e-mail address for this project.
email (a.pirogov@fz-juelich.de):
```

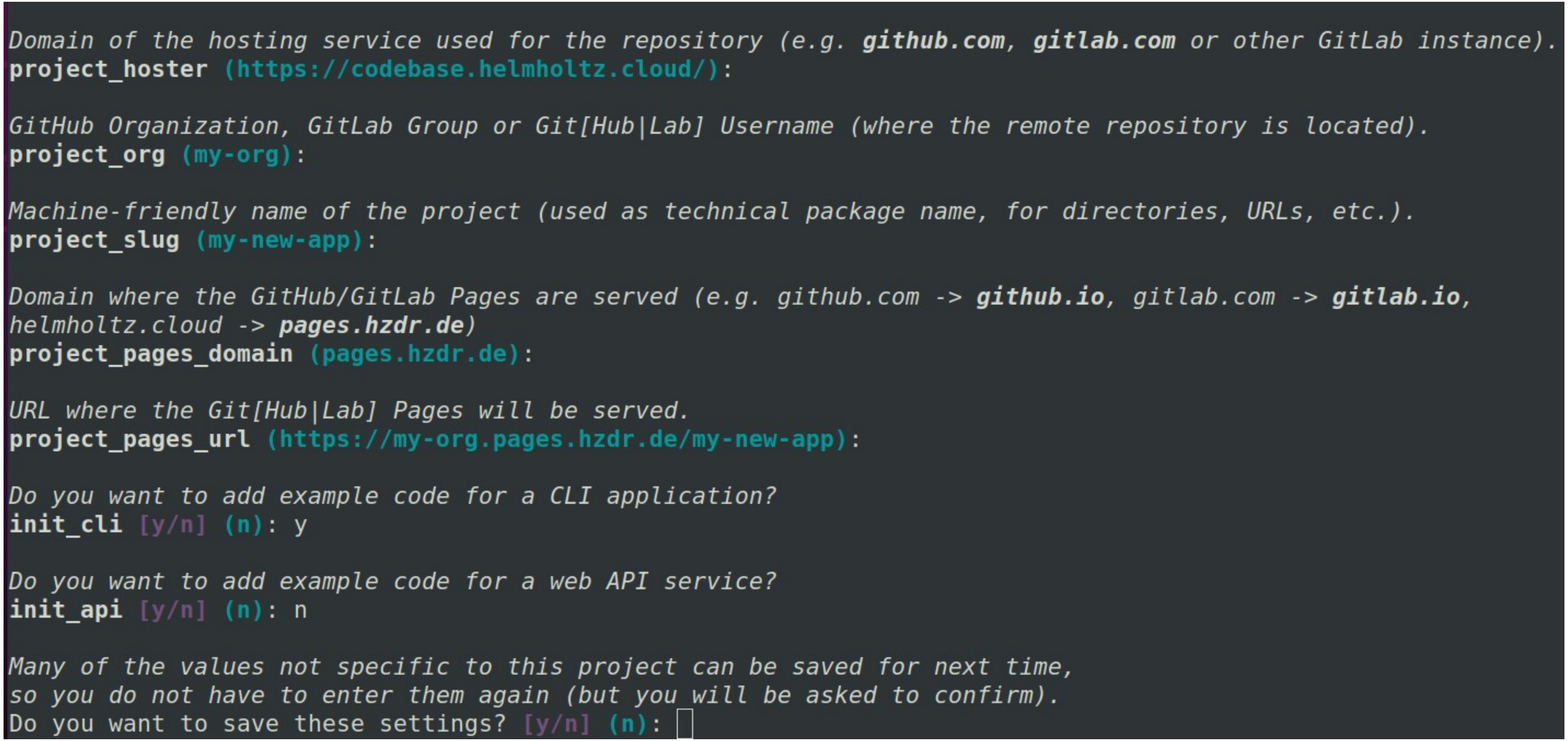
Feature Overview



```
[12:25:01] a.pirogov@ias9205 ~ $ fair-python-cookiecutter
──────────────────────────────── FAIR Python Cookiecutter 0.2.0 ──────

URL of the target remote repository at your git hosting service (leave empty if you did not create it yet).
project_repo_url: https://codebase.helmholtz.cloud/my-org/my-new-app

Name of the software project (written as it should show up in e.g. documentation).
project_name (my-new-app)

One-line description of the project (<= 512 characters).
project_description: This is a new amazing software project.

Search keywords characterizing your project (separated by spaces).
project_keywords: amazing python software

Semantic version number of the project (i.e., the version to be assigned to the next release).
project_version (0.1.0):

Year when the project was initiated (for copyright notice, usually the current year for new repositories).
project_year (2024):

License used for this project (must be a valid SPDX license identifier, such as MIT or GPL-3.0-only).
project_license (MIT):

Your last name (usually the family name).
last_name (Pirogov):

Your first name(s) (and everything else before your last name).
first_name (Anton):

Your contact e-mail address for this project.
email (a.pirogov@fz-juelich.de):
```

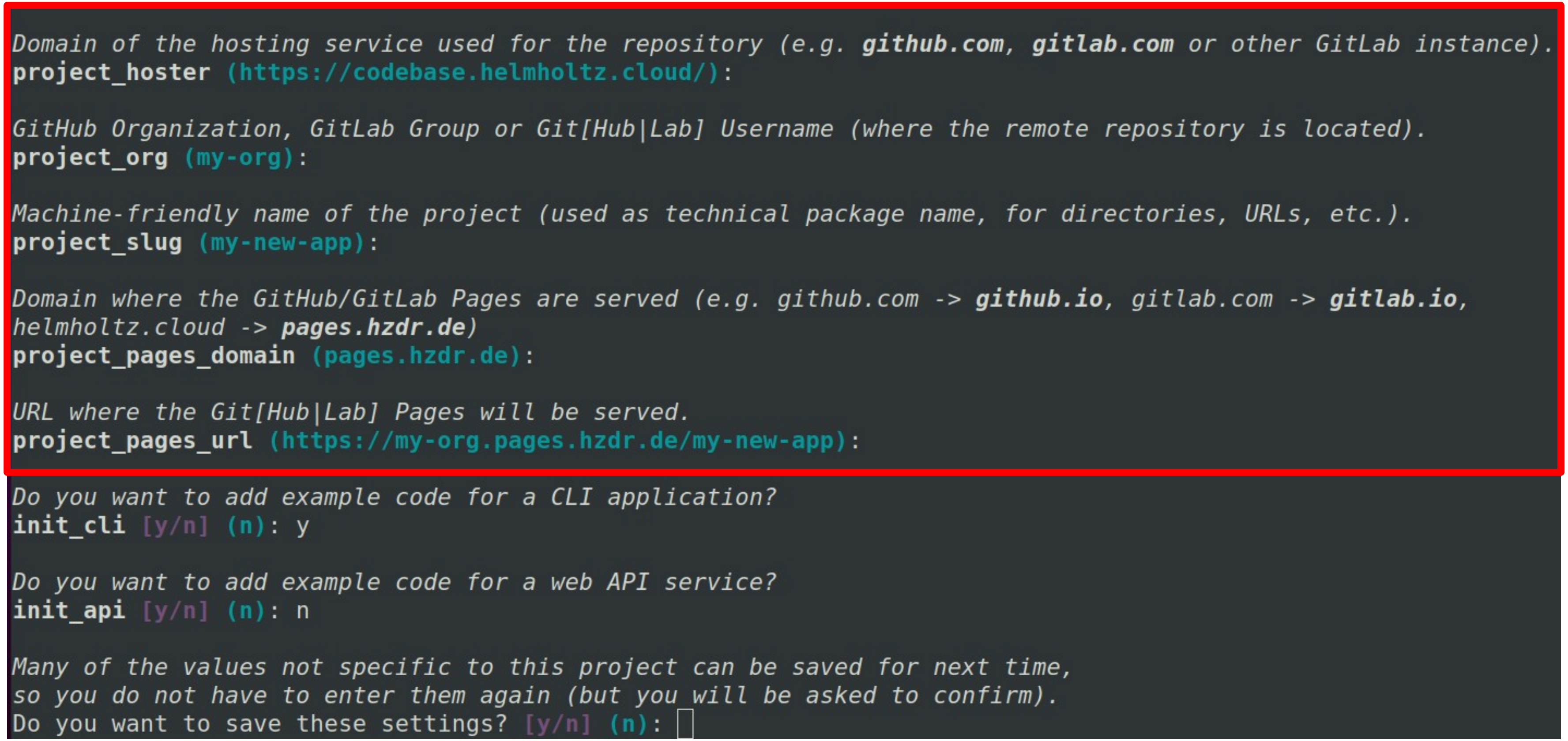
.cookiecutterrc

Feature Overview



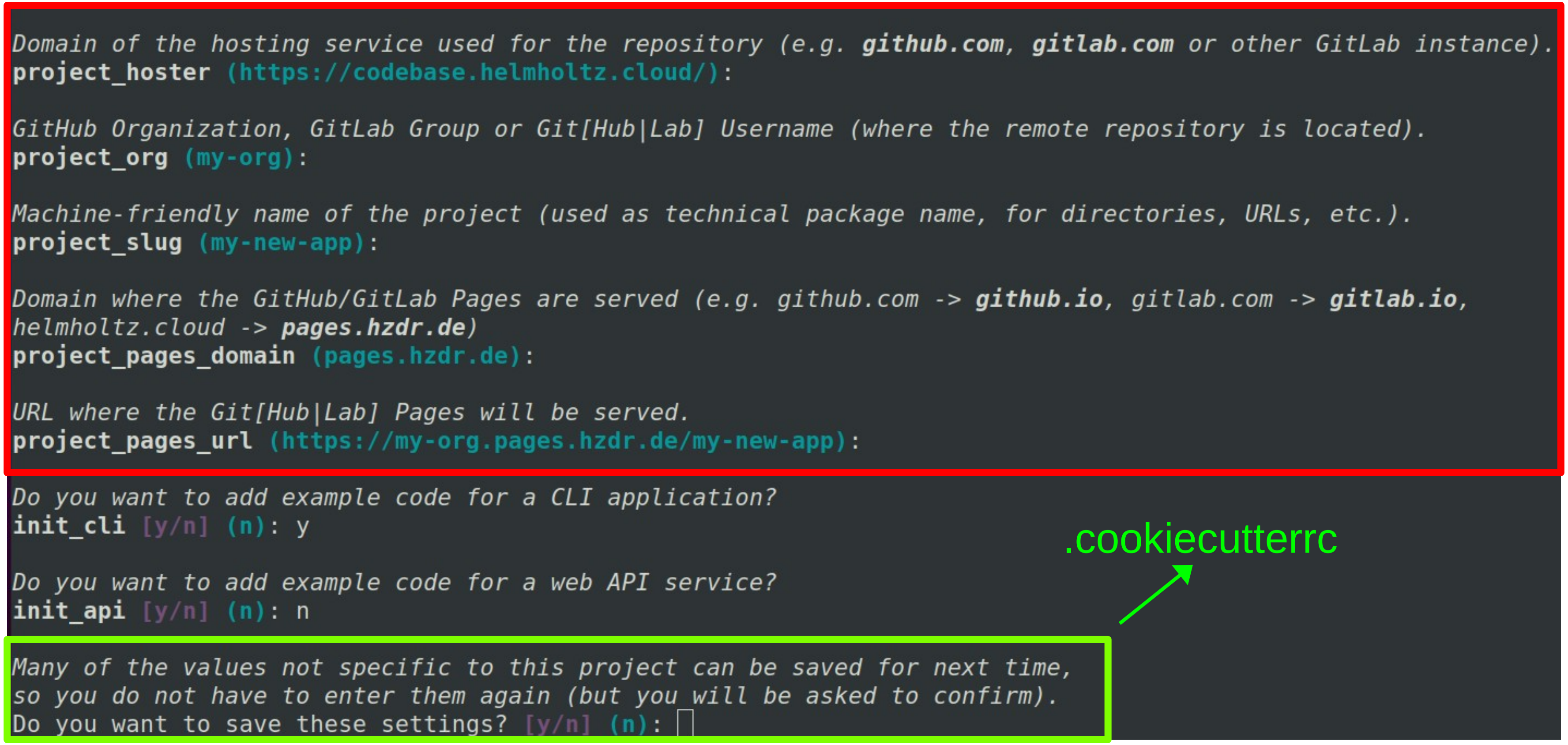
```
Domain of the hosting service used for the repository (e.g. github.com, gitlab.com or other GitLab instance).
project_hoster (https://codebase.helmholtz.cloud/):

GitHub Organization, GitLab Group or Git[Hub|Lab] Username (where the remote repository is located).
project_org (my-org):

Machine-friendly name of the project (used as technical package name, for directories, URLs, etc.).
project_slug (my-new-app):

Domain where the GitHub/GitLab Pages are served (e.g. github.com -> github.io, gitlab.com -> gitlab.io,
helmholtz.cloud -> pages.hzdr.de)
project_pages_domain (pages.hzdr.de):

URL where the Git[Hub|Lab] Pages will be served.
project_pages_url (https://my-org.pages.hzdr.de/my-new-app):

Do you want to add example code for a CLI application?
init_cli [y/n] (n): y

Do you want to add example code for a web API service?
init_api [y/n] (n): n

Many of the values not specific to this project can be saved for next time,
so you do not have to enter them again (but you will be asked to confirm).
Do you want to save these settings? [y/n] (n):
```

Feature Overview



```
Domain of the hosting service used for the repository (e.g. github.com, gitlab.com or other GitLab instance).
project_hoster (https://codebase.helmholtz.cloud/):

GitHub Organization, GitLab Group or Git[Hub|Lab] Username (where the remote repository is located).
project_org (my-org):

Machine-friendly name of the project (used as technical package name, for directories, URLs, etc.).
project_slug (my-new-app):

Domain where the GitHub/GitLab Pages are served (e.g. github.com -> github.io, gitlab.com -> gitlab.io,
helmholtz.cloud -> pages.hzdr.de)
project_pages_domain (pages.hzdr.de):

URL where the Git[Hub|Lab] Pages will be served.
project_pages_url (https://my-org.pages.hzdr.de/my-new-app):
Do you want to add example code for a CLI application?
init_cli [y/n] (n): y

Do you want to add example code for a web API service?
init_api [y/n] (n): n

Many of the values not specific to this project can be saved for next time,
so you do not have to enter them again (but you will be asked to confirm).
Do you want to save these settings? [y/n] (n):
```

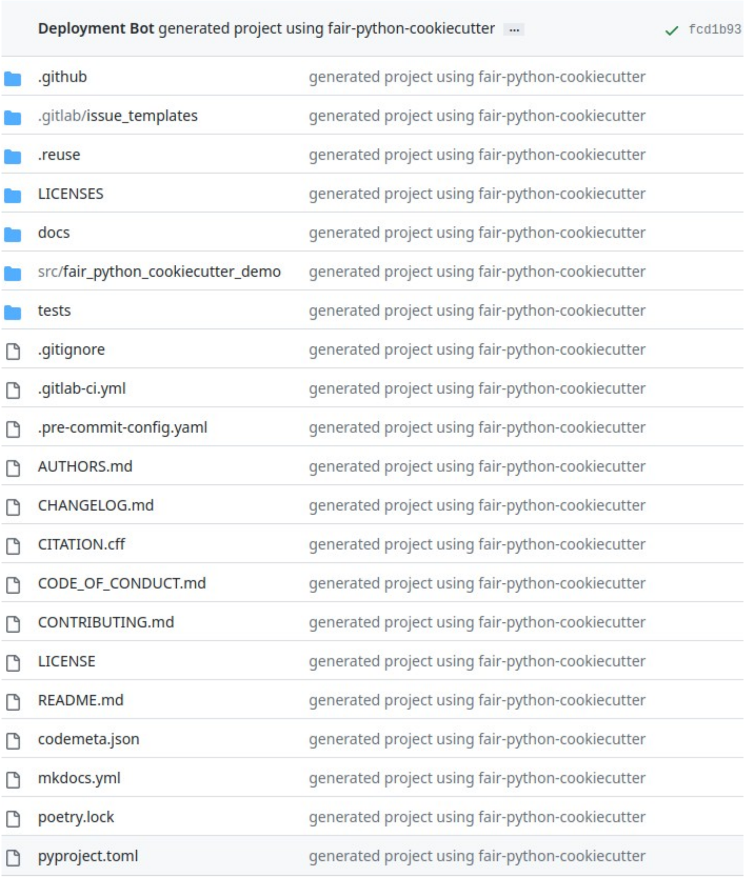
Feature Overview



```
Domain of the hosting service used for the repository (e.g. github.com, gitlab.com or other GitLab instance).
project_hoster (https://codebase.helmholtz.cloud/):

GitHub Organization, GitLab Group or Git[Hub|Lab] Username (where the remote repository is located).
project_org (my-org):

Machine-friendly name of the project (used as technical package name, for directories, URLs, etc.).
project_slug (my-new-app):

Domain where the GitHub/GitLab Pages are served (e.g. github.com -> github.io, gitlab.com -> gitlab.io,
helmholtz.cloud -> pages.hzdr.de)
project_pages_domain (pages.hzdr.de):

URL where the Git[Hub|Lab] Pages will be served.
project_pages_url (https://my-org.pages.hzdr.de/my-new-app):

Do you want to add example code for a CLI application?
init_cli [y/n] (n): y

Do you want to add example code for a web API service?
init_api [y/n] (n): n

Many of the values not specific to this project can be saved for next time,
so you do not have to enter them again (but you will be asked to confirm).
Do you want to save these settings? [y/n] (n):
```

.cookiecutterrc

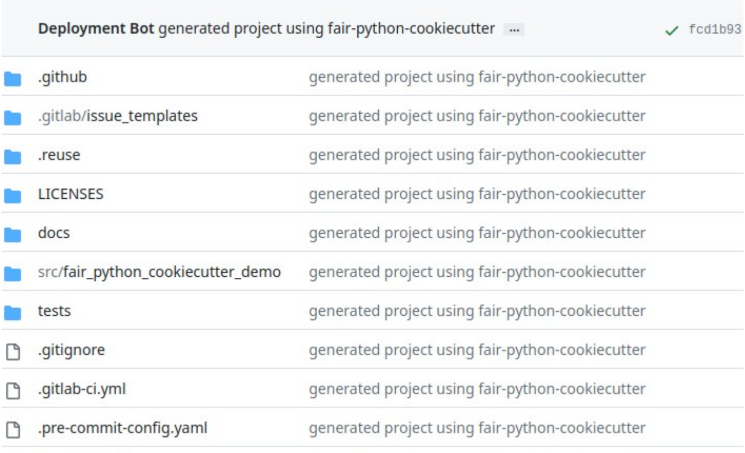
Feature Overview



Deployment Bot generated project using fair-python-cookiecutter ... ✓ fcd1b93

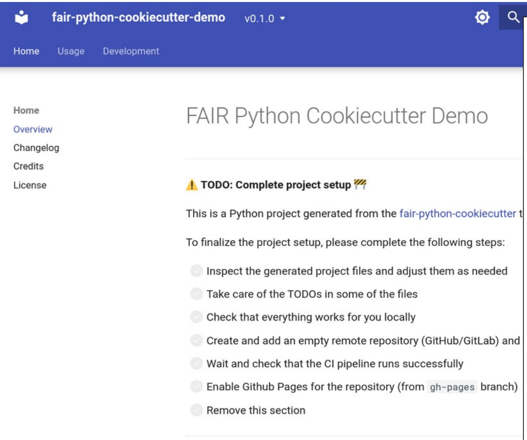
.github	generated project using fair-python-cookiecutter
.gitlab/issue_templates	generated project using fair-python-cookiecutter
.reuse	generated project using fair-python-cookiecutter
LICENSES	generated project using fair-python-cookiecutter
docs	generated project using fair-python-cookiecutter
src/fair_python_cookiecutter_demo	generated project using fair-python-cookiecutter
tests	generated project using fair-python-cookiecutter
.gitignore	generated project using fair-python-cookiecutter
.gitlab-ci.yml	generated project using fair-python-cookiecutter
.pre-commit-config.yaml	generated project using fair-python-cookiecutter
AUTHORS.md	generated project using fair-python-cookiecutter
CHANGELOG.md	generated project using fair-python-cookiecutter
CITATION.cff	generated project using fair-python-cookiecutter
CODE_OF_CONDUCT.md	generated project using fair-python-cookiecutter
CONTRIBUTING.md	generated project using fair-python-cookiecutter
LICENSE	generated project using fair-python-cookiecutter
README.md	generated project using fair-python-cookiecutter
codemeta.json	generated project using fair-python-cookiecutter
mkdocs.yml	generated project using fair-python-cookiecutter
poetry.lock	generated project using fair-python-cookiecutter
pyproject.toml	generated project using fair-python-cookiecutter

Feature Overview



Deployment Bot generated project using fair-python-cookiecutter ... ✓ fcd1b93

.github	generated project using fair-python-cookiecutter
.gitlab/issue_templates	generated project using fair-python-cookiecutter
.reuse	generated project using fair-python-cookiecutter
LICENSES	generated project using fair-python-cookiecutter
docs	generated project using fair-python-cookiecutter
src/fair_python_cookiecutter_demo	generated project using fair-python-cookiecutter
tests	generated project using fair-python-cookiecutter
.gitignore	generated project using fair-python-cookiecutter
.gitlab-ci.yml	generated project using fair-python-cookiecutter
.pre-commit-config.yaml	generated project using fair-python-cookiecutter



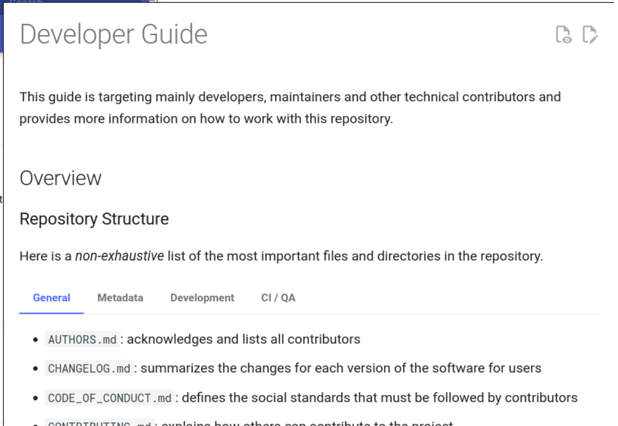
FAIR Python Cookiecutter Demo

⚠️ **TODO: Complete project setup** 🚧

This is a Python project generated from the fair-python-cookiecutter t

To finalize the project setup, please complete the following steps:

- ○ Inspect the generated project files and adjust them as needed
- ○ Take care of the TODOs in some of the files
- ○ Check that everything works for you locally
- ○ Create and add an empty remote repository (GitHub/GitLab) and
- ○ Wait and check that the CI pipeline runs successfully
- ○ Enable Github Pages for the repository (from `gh-pages` branch)
- ○ Remove this section



Developer Guide

This guide is targeting mainly developers, maintainers and other technical contributors and provides more information on how to work with this repository.

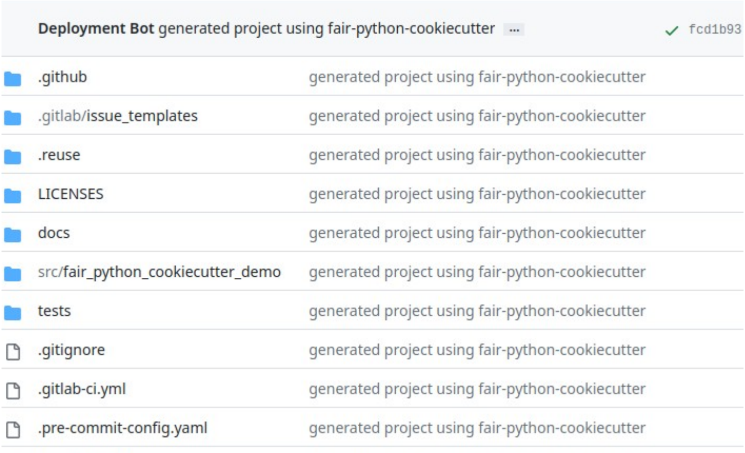
Overview

Repository Structure

Here is a *non-exhaustive* list of the most important files and directories in the repository.

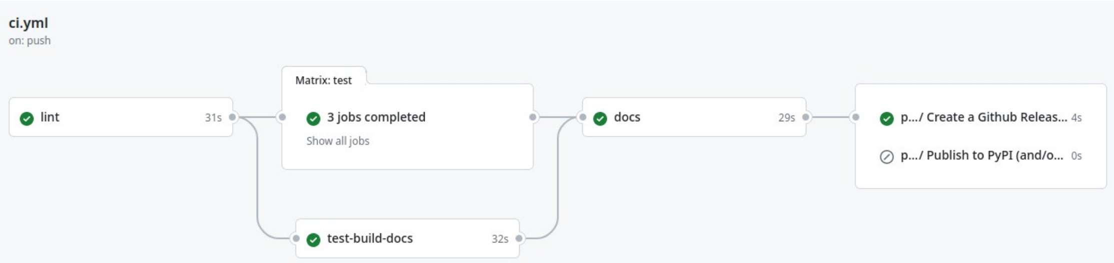
- `AUTHORS.md` : acknowledges and lists all contributors
- `CHANGELOG.md` : summarizes the changes for each version of the software for users
- `CODE_OF_CONDUCT.md` : defines the social standards that must be followed by contributors
- `CONTRIBUTING.md` : explains how others can contribute to the project

Feature Overview



Deployment Bot generated project using fair-python-cookiecutter ··· ✓ fcd1b93

.github	generated project using fair-python-cookiecutter
.gitlab/issue_templates	generated project using fair-python-cookiecutter
.reuse	generated project using fair-python-cookiecutter
LICENSES	generated project using fair-python-cookiecutter
docs	generated project using fair-python-cookiecutter
src/fair_python_cookiecutter_demo	generated project using fair-python-cookiecutter
tests	generated project using fair-python-cookiecutter
.gitignore	generated project using fair-python-cookiecutter
.gitlab-ci.yml	generated project using fair-python-cookiecutter
.pre-commit-config.yaml	generated project using fair-python-cookiecutter

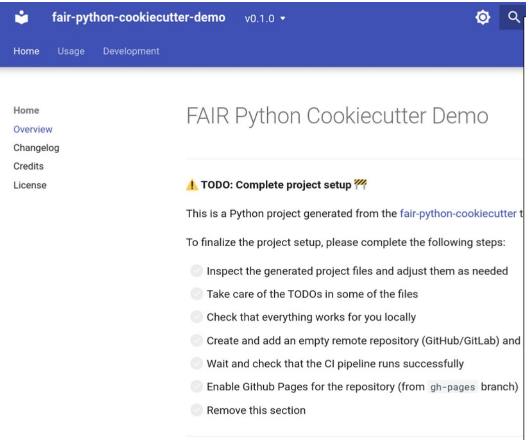


ci.yml
on: push

✓ lint 31s
Matrix: test
✓ 3 jobs completed
Show all jobs
✓ test-build-docs 32s
✓ docs 29s
✓ p.../ Create a Github Releas... 4s
⊘ p.../ Publish to PyPI (and/o... 0s

GitHub

GitLab



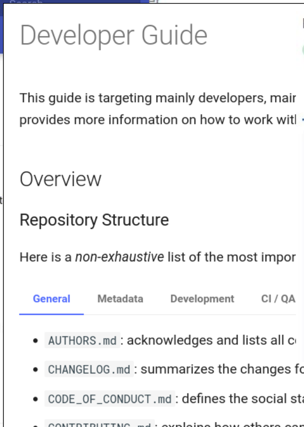
FAIR Python Cookiecutter Demo

⚠️ **TODO: Complete project setup** 🚧

This is a Python project generated from the fair-python-cookiecutter t

To finalize the project setup, please complete the following steps:

- ○ Inspect the generated project files and adjust them as needed
- ○ Take care of the TODOs in some of the files
- ○ Check that everything works for you locally
- ○ Create and add an empty remote repository (GitHub/GitLab) and
- ○ Wait and check that the CI pipeline runs successfully
- ○ Enable Github Pages for the repository (from `gh-pages` branch)
- ○ Remove this section



Developer Guide

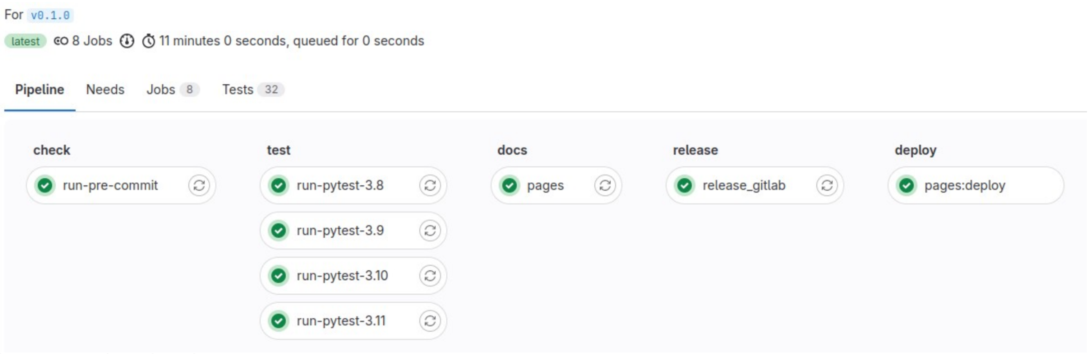
This guide is targeting mainly developers, mair provides more information on how to work witl

Overview

Repository Structure

Here is a *non-exhaustive* list of the most impor

- `AUTHORS.md` : acknowledges and lists all c
- `CHANGELOG.md` : summarizes the changes for each version of the software for users
- `CODE_OF_CONDUCT.md` : defines the social standards that must be followed by contributors
- `CONTRIBUTING.md` : explains how others can contribute to the project

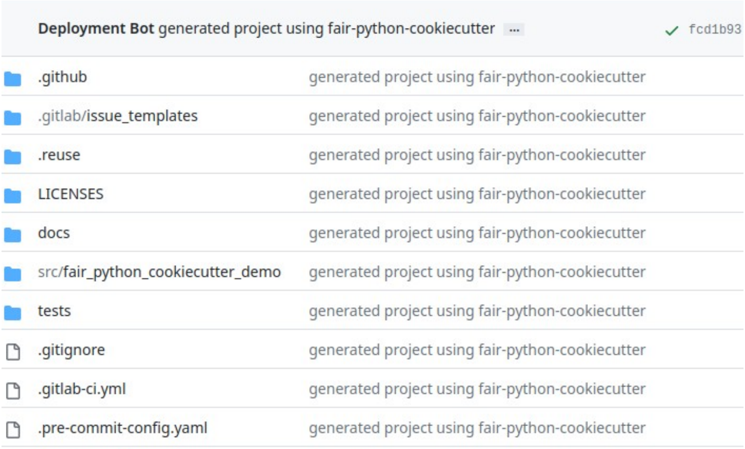


For v0.1.0

latest 8 Jobs 11 minutes 0 seconds, queued for 0 seconds

check	test	docs	release	deploy
✓ run-pre-commit	✓ run-pytest-3.8	✓ pages	✓ release_gitlab	✓ pages:deploy
	✓ run-pytest-3.9			
	✓ run-pytest-3.10			
	✓ run-pytest-3.11			

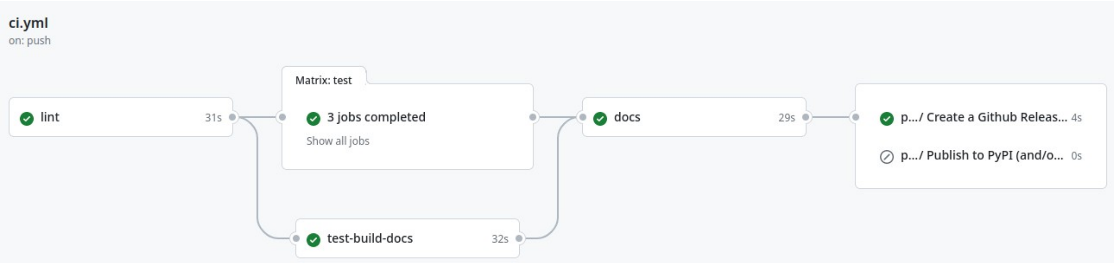
Feature Overview



Deployment Bot generated project using fair-python-cookiecutter ... ✓ fcd1b93

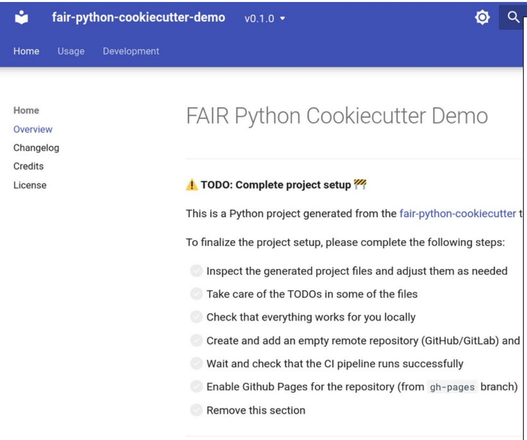
.github	generated project using fair-python-cookiecutter
.gitlab/issue_templates	generated project using fair-python-cookiecutter
.reuse	generated project using fair-python-cookiecutter
LICENSES	generated project using fair-python-cookiecutter
docs	generated project using fair-python-cookiecutter
src/fair_python_cookiecutter_demo	generated project using fair-python-cookiecutter
tests	generated project using fair-python-cookiecutter
.gitignore	generated project using fair-python-cookiecutter
.gitlab-ci.yml	generated project using fair-python-cookiecutter
.pre-commit-config.yaml	generated project using fair-python-cookiecutter

https://github.com/Materials-Data-Science-and-Informatics/fair-python-cookiecutter-demo



GitHub

GitLab



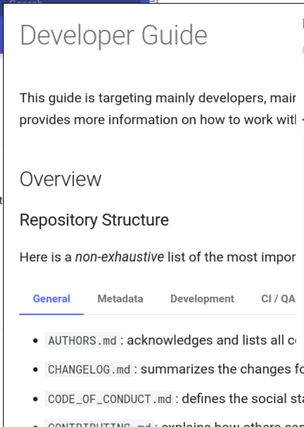
FAIR Python Cookiecutter Demo

⚠️ **TODO: Complete project setup** 🚧

This is a Python project generated from the fair-python-cookiecutter t

To finalize the project setup, please complete the following steps:

- Inspect the generated project files and adjust them as needed
- Take care of the TODOs in some of the files
- Check that everything works for you locally
- Create and add an empty remote repository (GitHub/GitLab) and
- Wait and check that the CI pipeline runs successfully
- Enable Github Pages for the repository (from `gh-pages` branch)
- Remove this section



Developer Guide

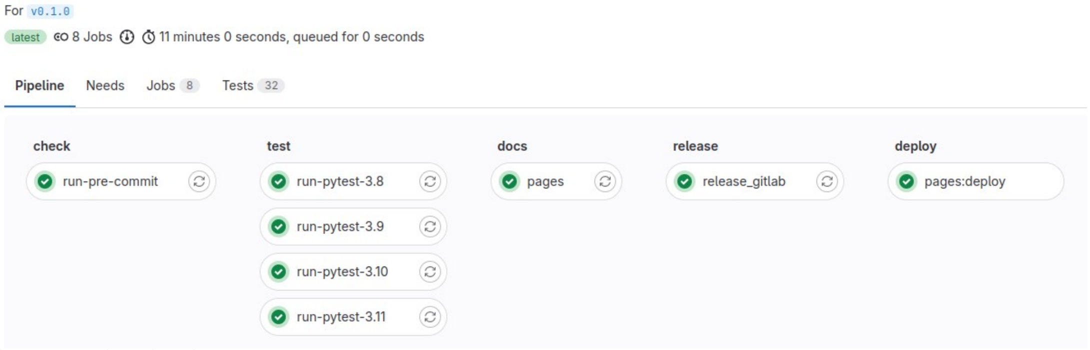
This guide is targeting mainly developers, mair provides more information on how to work witl

Overview

Repository Structure

Here is a *non-exhaustive* list of the most impor

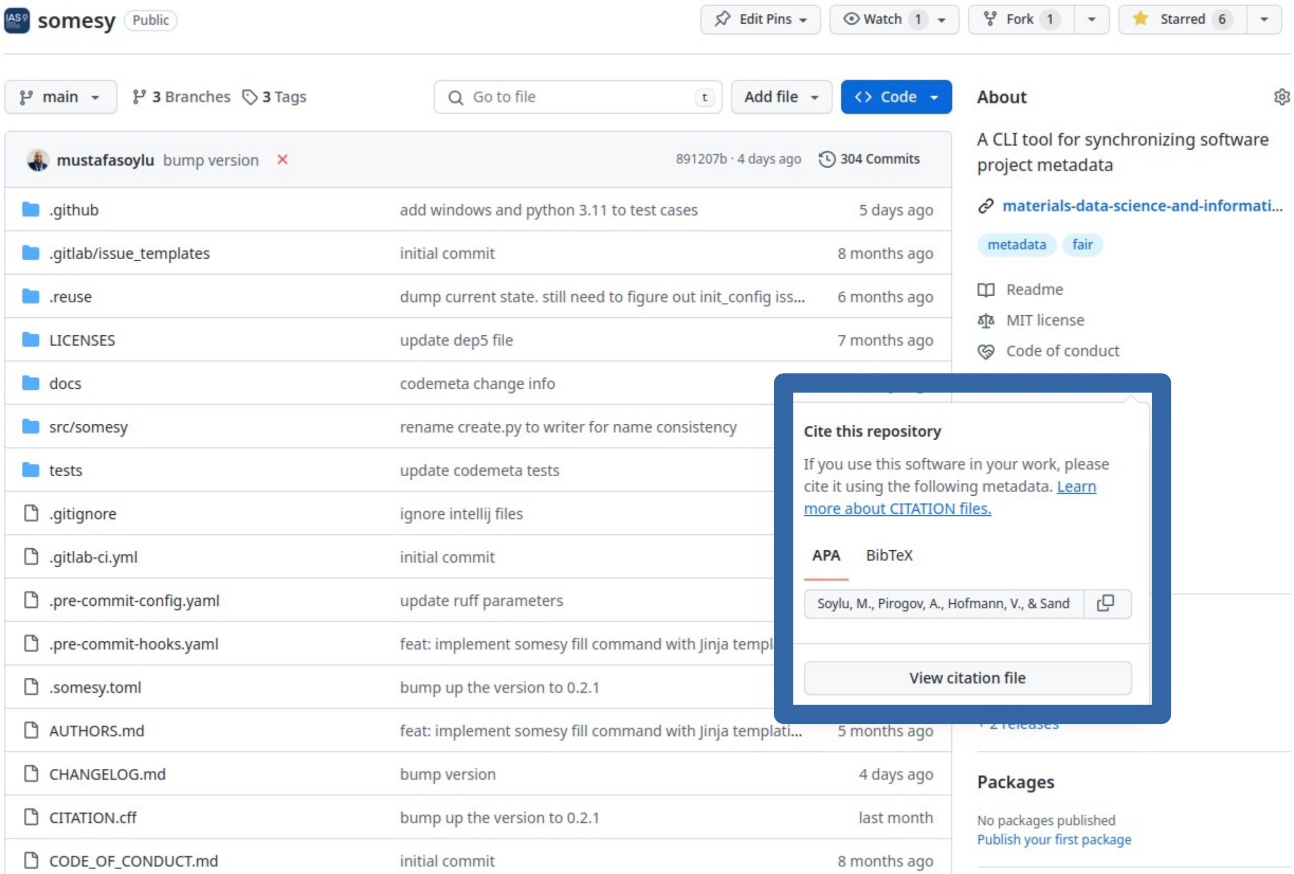
- `AUTHORS.md` : acknowledges and lists all c
- `CHANGELOG.md` : summarizes the changes for each version of the software for users
- `CODE_OF_CONDUCT.md` : defines the social standards that must be followed by contributors



Software Metadata Synchronization

somesy

Software Metadata: Did you know that you can do this?



somesy Public

Edit Pins · Watch 1 · Fork 1 · Starred 6

main · 3 Branches · 3 Tags · Go to file · Add file · Code

mustafasoylu bump version ✕ · 891207b · 4 days ago · 304 Commits

.github	add windows and python 3.11 to test cases	5 days ago
.gitlab/issue_templates	initial commit	8 months ago
.reuse	dump current state. still need to figure out init_config iss...	6 months ago
LICENSES	update dep5 file	7 months ago
docs	codemeta change info	
src/somesy	rename create.py to writer for name consistency	
tests	update codemeta tests	
.gitignore	ignore intellij files	
.gitlab-ci.yml	initial commit	
.pre-commit-config.yaml	update ruff parameters	
.pre-commit-hooks.yaml	feat: implement somesy fill command with Jinja templ	
.somesy.toml	bump up the version to 0.2.1	
AUTHORS.md	feat: implement somesy fill command with Jinja templati...	5 months ago
CHANGELOG.md	bump version	4 days ago
CITATION.cff	bump up the version to 0.2.1	last month
CODE_OF_CONDUCT.md	initial commit	8 months ago

About

A CLI tool for synchronizing software project metadata

materials-data-science-and-informati...

metadata · fair

Readme

MIT license

Code of conduct

+ 2 releases

Packages

No packages published

Publish your first package

Cite this repository

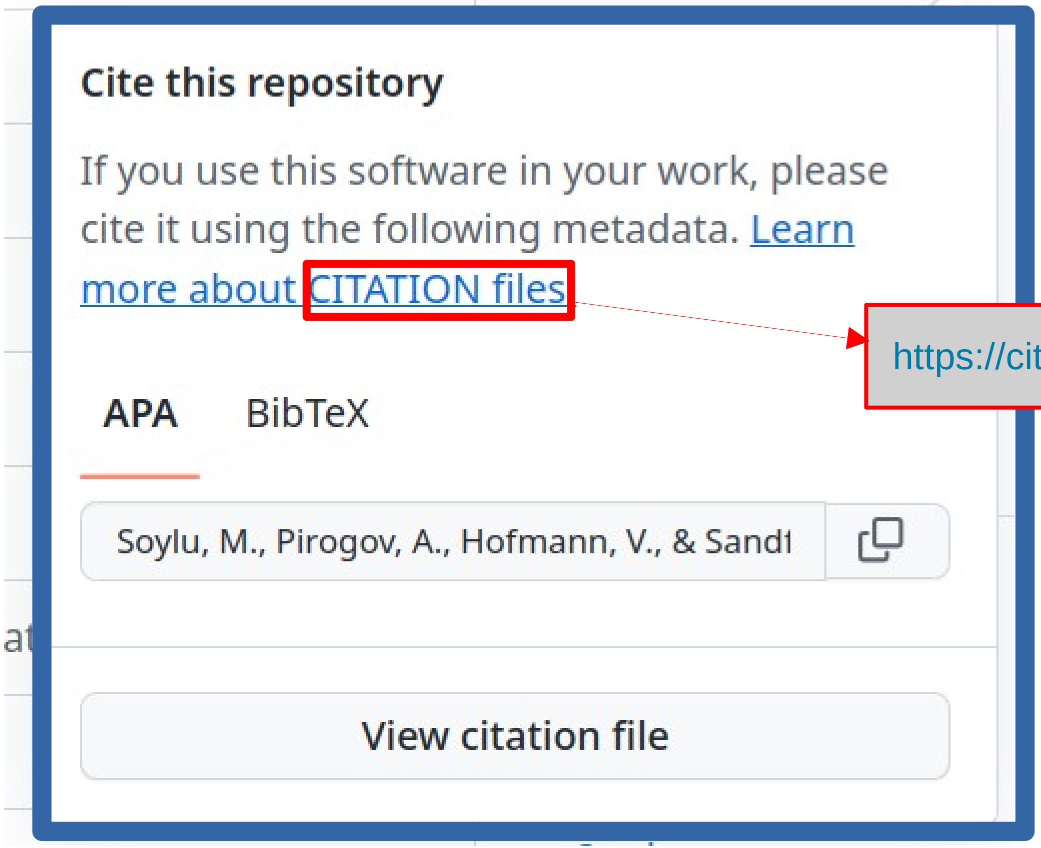
If you use this software in your work, please cite it using the following metadata. Learn more about CITATION files.

APA · BibTeX

Soylu, M., Pirogov, A., Hofmann, V., & Sand

View citation file

Software Metadata: Did you know that you can do this?



Cite this repository

If you use this software in your work, please cite it using the following metadata. Learn more about CITATION files

Soylu, M., Pirogov, A., Hofmann, V., & Sandt

https://citation-file-format.github.io/

Some FAIR-related Software Metadata Files

CITATION.cff:

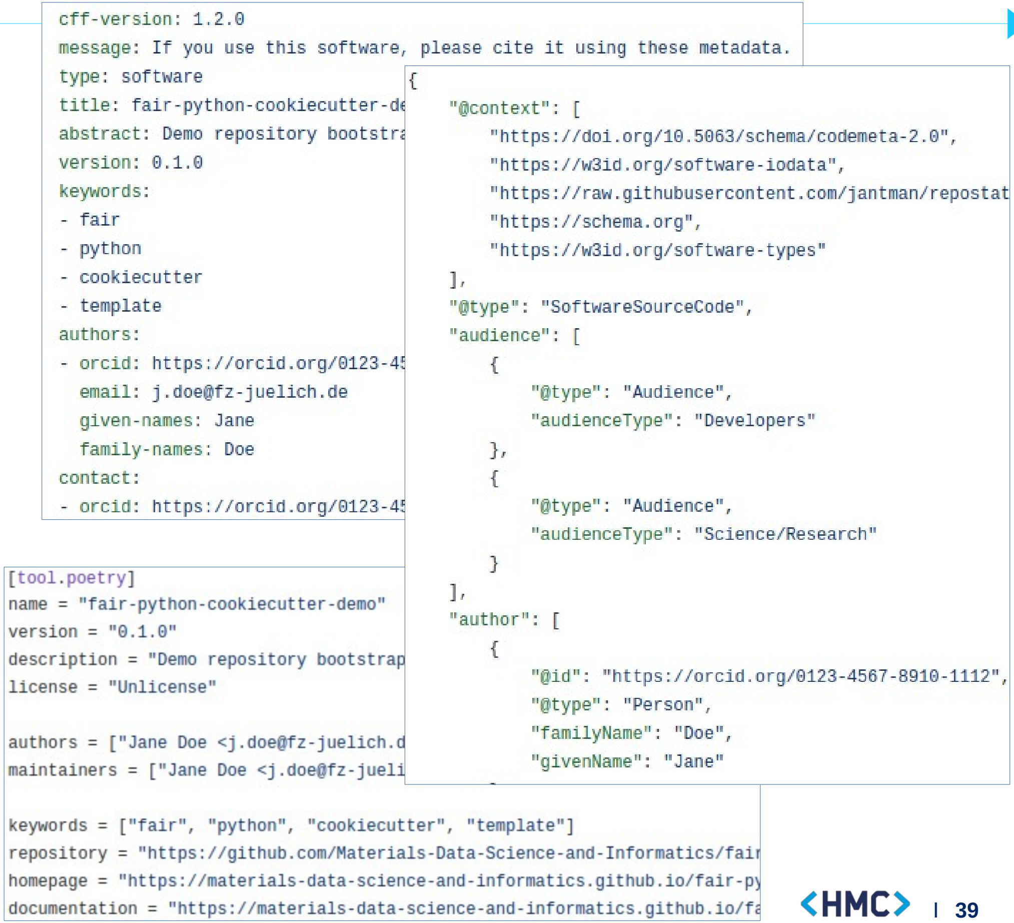
- Human+Machine readability
- GitHub/Zenodo integration
- General project/bibliographic metadata
- Usable for software and datasets

codemeta.json:

- Machine readability
- Based on linked data (schema.org)
- Rich software-related metadata
- Language-agnostic

Language-specific files:

- pyproject.toml, package.json, …
- Some project metadata
- Some software metadata
- Most detailed, but least interoperable



```
cff-version: 1.2.0
message: If you use this software, please cite it using these metadata.
type: software
title: fair-python-cookiecutter-de
abstract: Demo repository bootstra
version: 0.1.0
keywords:
- fair
- python
- cookiecutter
- template
authors:
- orcid: https://orcid.org/0123-45
  email: j.doe@fz-juelich.de
  given-names: Jane
  family-names: Doe
contact:
- orcid: https://orcid.org/0123-45
```

```
{
    "@context": [
        "https://doi.org/10.5063/schema/codemeta-2.0",
        "https://w3id.org/software-iodata",
        "https://raw.githubusercontent.com/jantman/repostat
        "https://schema.org",
        "https://w3id.org/software-types"
    ],
    "@type": "SoftwareSourceCode",
    "audience": [
        {
            "@type": "Audience",
            "audienceType": "Developers"
        },
        {
            "@type": "Audience",
            "audienceType": "Science/Research"
        }
    ],
    "author": [
        {
            "@id": "https://orcid.org/0123-4567-8910-1112",
            "@type": "Person",
            "familyName": "Doe",
            "givenName": "Jane"
```

```
[tool.poetry]
name = "fair-python-cookiecutter-demo"
version = "0.1.0"
description = "Demo repository bootstrap
license = "Unlicense"

authors = ["Jane Doe <j.doe@fz-juelich.d
maintainers = ["Jane Doe <j.doe@fz-jueli

keywords = ["fair", "python", "cookiecutter", "template"]
repository = "https://github.com/Materials-Data-Science-and-Informatics/fair
homepage = "https://materials-data-science-and-informatics.github.io/fair-py
documentation = "https://materials-data-science-and-informatics.github.io/fa
```

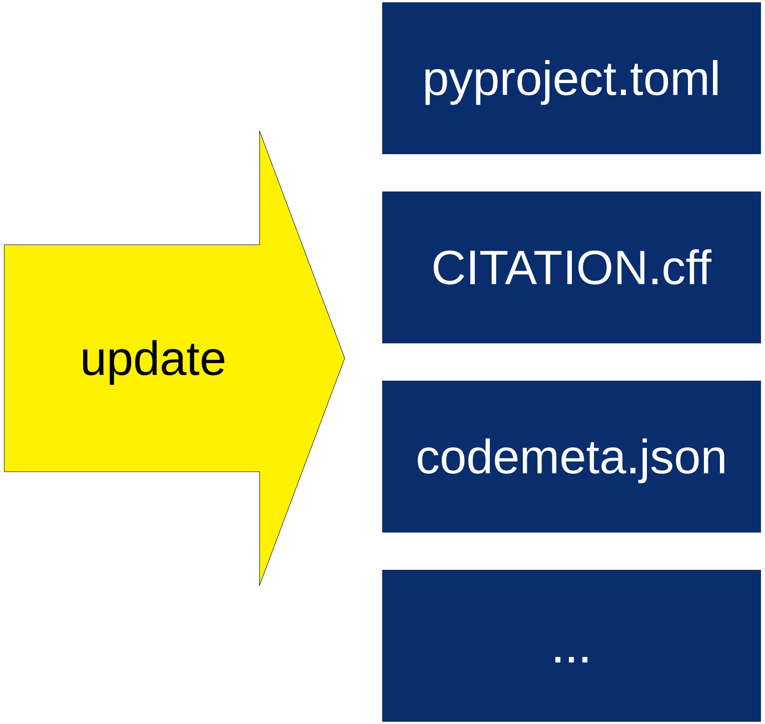
So now we have metadata...

… and the project is evolving over time:

- A new version is released

- New contributors join the project
- Old contributors leave

- The website or repository is moved
- Description and keywords are updated
- ...



update

pyproject.toml

CITATION.cff

codemeta.json

...

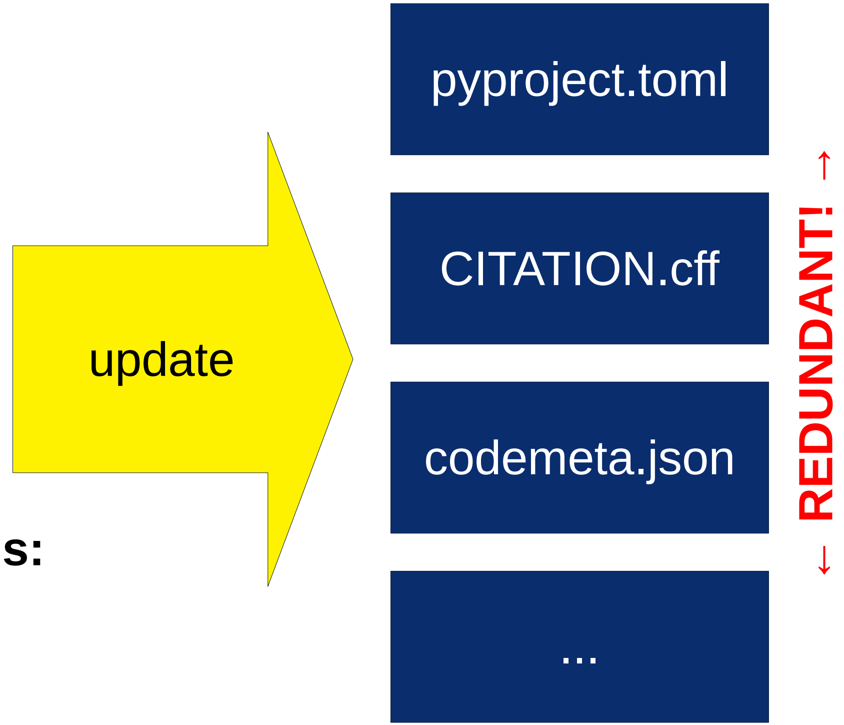
So now we have metadata...

… and the project is evolving over time:

- A new version is released

- New contributors join the project
- Old contributors leave

- The website or repository is moved
- Description and keywords are updated
- …



Manual maintenance of software metadata is:

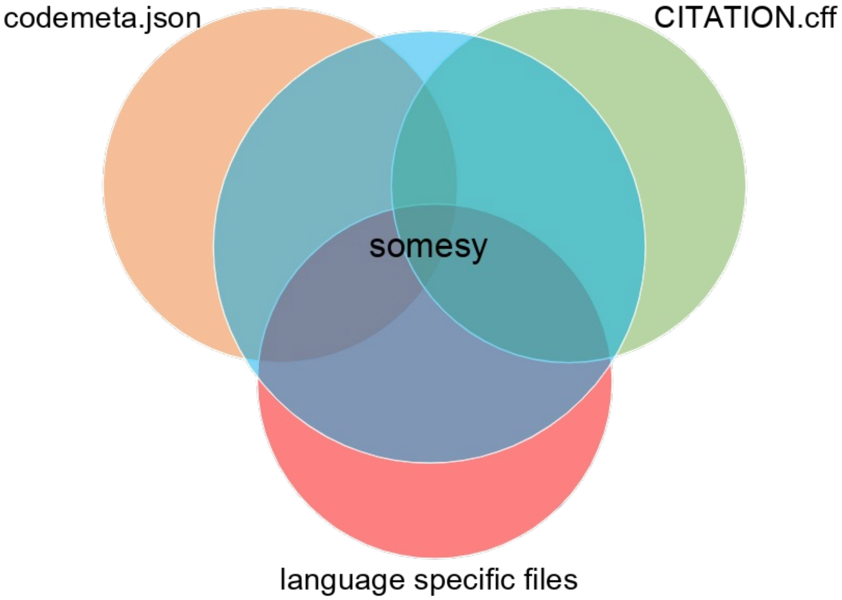
- easy to forget → quickly outdated
- error-prone → inconsistent
- time-consuming → creates overhead

SOMESY: A tool for software metadata synchronization

- **Single source of truth** for software metadata
- **Avoids duplication** with rich metadata
- **Automates** the metadata synchronization
- **Preserves** other content and comments

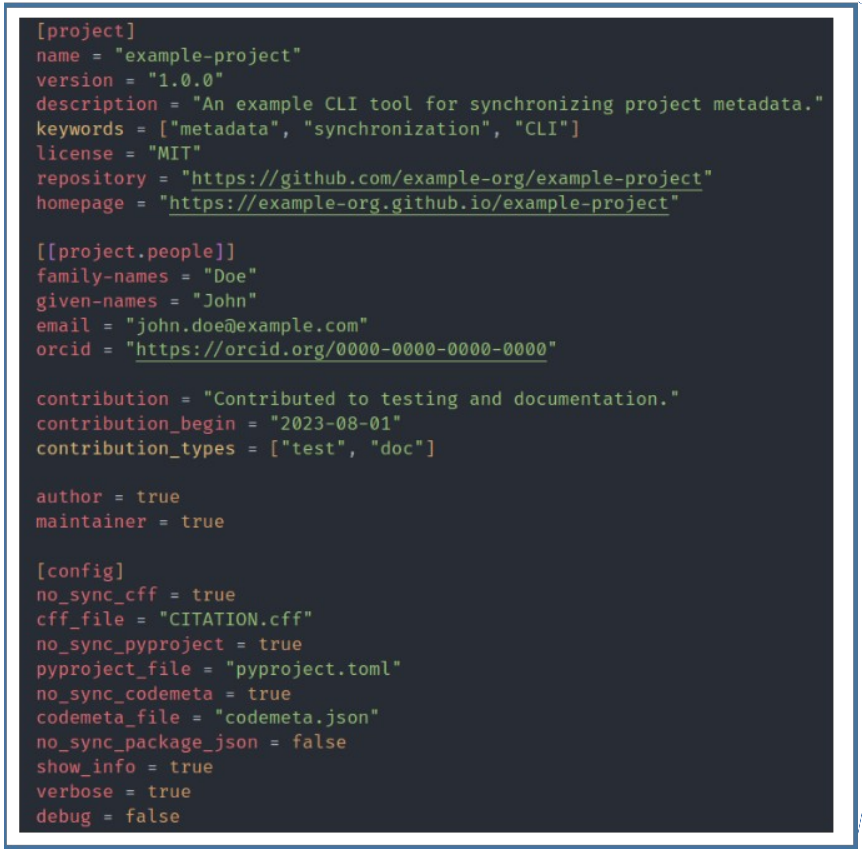
  in existing files

  - Person information is **matched, checked and merged** with the existing information in the files
- Provides a **pre-commit hook**
- Extensible and **modular**
- Works with **Windows, Ubuntu, and macOS**

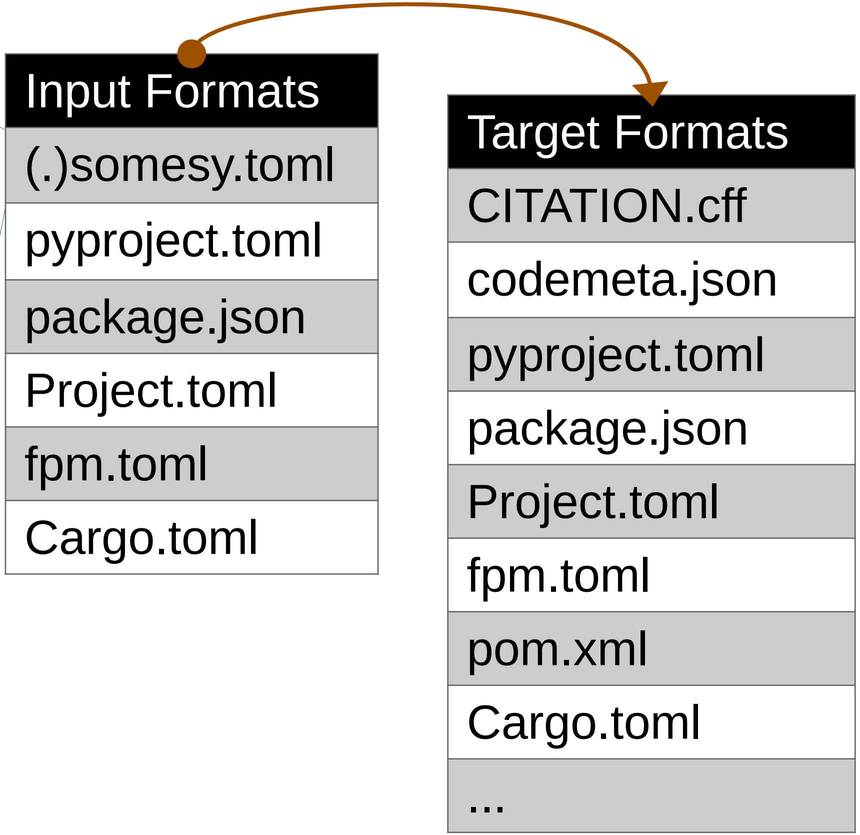


SOMESY: A tool for software metadata synchronization

somesy.toml



```toml
[project]
name = "example-project"
version = "1.0.0"
description = "An example CLI tool for synchronizing project metadata."
keywords = ["metadata", "synchronization", "CLI"]
license = "MIT"
repository = "https://github.com/example-org/example-project"
homepage = "https://example-org.github.io/example-project"

[[project.people]]
family-names = "Doe"
given-names = "John"
email = "john.doe@example.com"
orcid = "https://orcid.org/0000-0000-0000-0000"

contribution = "Contributed to testing and documentation."
contribution_begin = "2023-08-01"
contribution_types = ["test", "doc"]

author = true
maintainer = true

[config]
no_sync_cff = true
cff_file = "CITATION.cff"
no_sync_pyproject = true
pyproject_file = "pyproject.toml"
no_sync_codemeta = true
codemeta_file = "codemeta.json"
no_sync_package_json = false
show_info = true
verbose = true
debug = false
```



Input Formats
(.)somesy.toml
pyproject.toml
package.json
Project.toml
fpm.toml
Cargo.toml

Target Formats
CITATION.cff
codemeta.json
pyproject.toml
package.json
Project.toml
fpm.toml
pom.xml
Cargo.toml
...

Interested? Give it a try and get in touch!

A modern best-practice Python template for research software developers

FAIR PYTHON COOKIECUTTER

https://github.com/Materials-Data-Science-and-Informatics/fair-python-cookiecutter

A language-agnostic software project metadata synchronization tool

SOMESY

https://github.com/Materials-Data-Science-and-Informatics/somesy