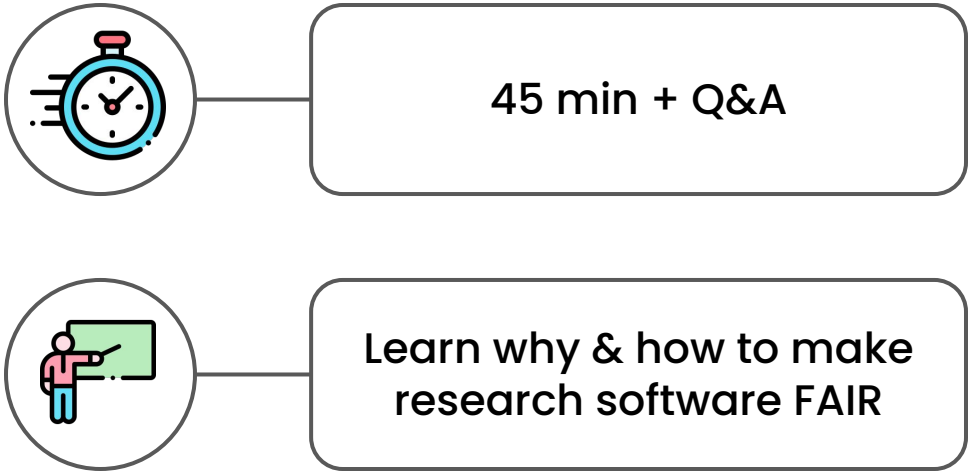
Making Research Software FAIR: Background and Practical Steps

Bhavesh Patel, Ph.D.
Research Professor

About This Presentation



FAIR = Findable, Accessible, Interoperable, and Reusable

Background

We started thinking about it around December 2021

We develop software to make data FAIR

We should make our software FAIR!

How to make research software FAIR?

About Research Software

About Research Software

Definition

Any software created during the research process or for a research purpose

Source: Gruenpeter, M. et al. Defining Research Software: a controversial discussion. Zenodo https://doi.org/10.5281/zenodo.5504016 (2021).

Excel used to analyze and visualize data

Python script developed to analyze and visualize data

About Research Software

There are many different types

Various formats

Python script

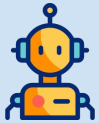
C/C++ code

Fortran code

Various applications

Data analysis

Computational model



AI/ML model

About Research Software

It is an essential element of research

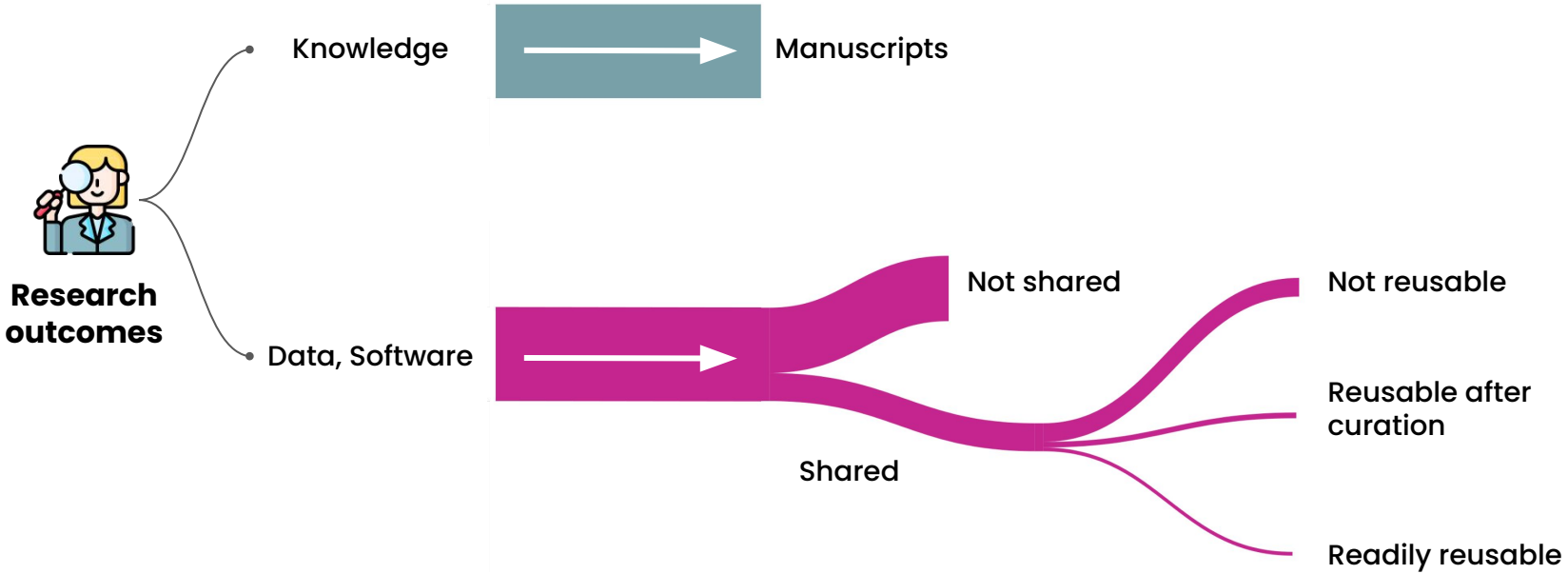
More and more research projects include development of research software

Research software is the main outcome of many research projects

https://www.researchsoft.org/resa-resources

About Research Software

Yet it is not made reusable



About Research Software

Sharing and making it reusable is critical

Enable reproducible, transparent research

Prevent duplicate effort

Get recognition for development effort

Making Software Reusable

Making Software Reusable

FAIR principles - Origin

How to make all research outcomes, including software, optimally reusable by humans and machines?

Findable, Accessible, Interoperable, and Reusable (FAIR) Principles (2016)

https://doi.org/10.1038/sdata.2016.18

Making Software Reusable

FAIR principles - 15 principles

To be Findable:
F1. (meta)data are assigned a globally unique and persistent identifier
F2. data are described with rich metadata (defined by R1 below)
F3. metadata clearly and explicitly include the identifier of the data it describes
F4. (meta)data are registered or indexed in a searchable resource

To be Accessible:
A1. (meta)data are retrievable by their identifier using a standardized communications protocol
A1.1 the protocol is open, free, and universally implementable
A1.2 the protocol allows for an authentication and authorization procedure, where necessary
A2. metadata are accessible, even when the data are no longer available

To be Interoperable:
I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
I2. (meta)data use vocabularies that follow FAIR principles
I3. (meta)data include qualified references to other (meta)data

To be Reusable:
R1. meta(data) are richly described with a plurality of accurate and relevant attributes
R1.1. (meta)data are released with a clear and accessible data usage license
R1.2. (meta)data are associated with detailed provenance
R1.3. (meta)data meet domain-relevant community standards

https://doi.org/10.1038/sdata.2016.18

Making Software Reusable

FAIR principles - Adoption

G20

Leaders at the 2016 G20 meeting released a join press release expressing their intention to support implementation of FAIR principles in publicly funded research

"Turning FAIR into Reality" report (2018)

NIH

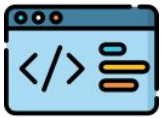
National Institutes of Health

New data sharing policy (January 2023) requires all proposal to include a Data Management and Sharing Plan (DMSP) that describes how data will be made FAIR

Making Software Reusable

FAIR principles - Problem for software

Many in the research software community found that the
FAIR Principles were not suitable for software



- ➔ Granularity
- ➔ Dependencies
- ➔ Multiple versions

Making Software Reusable

FAIR4RS Principles - Background

FAIR for Research Software (FAIR4RS) Working Group

200+ stakeholders involved

2020 - 2022

FAIR Principles for Research Software or FAIR4RS Principles

https://doi.org/10.1038/s41597-022-01710-x

Making Software Reusable

FAIR4RS Principles - 17 principles

Findable	F1. Software is assigned a globally unique and persistent identifier.
	F1.1. Components of the software representing levels of granularity are assigned distinct identifiers.
	F1.2. Different versions of the software are assigned distinct identifiers.
	F2. Software is described with rich metadata.
	F3. Metadata clearly and explicitly include the identifier of the software they describe.
	F4. Metadata are FAIR, searchable and indexable.
Accessible	A1. Software is retrievable by its identifier using a standardised communications protocol.
	A1.1. The protocol is open, free, and universally implementable.
	A1.2. The protocol allows for an authentication and authorization procedure, where necessary.
	A2. Metadata are accessible, even when the software is no longer available.
Interoperable	I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards.
	I2. Software includes qualified references to other objects.
Reusable	R1. Software is described with a plurality of accurate and relevant attributes.
	R1.1. Software is given a clear and accessible license.
	R1.2. Software is associated with detailed provenance.
	R2. Software includes qualified references to other software.
	R3. Software meets domain-relevant community standards.

Making Software Reusable

FAIR4RS Principles - Interpretation

Findable → **Software is easy to find**
Software has a unique identifier (e.g. a DOI)
Metadata are provided and indexed in search engines

Accessible → **Software access process is clear**
Software is accessible using a standard protocol (e.g., HTTP)
Access requirements are clearly stated

Interoperable → **Software interoperate with other software**
Software reads, writes and exchanges data following applicable standards
Connections to related resources are documented

Reusable → **Software is easily usable**
Software has a clear usage license
Software is well documented

Very similar to what we do when sharing manuscripts!

Making Software Reusable

FAIR4RS Principles - Problem

The FAIR4RS Principles, by design,
do not provide actionable instructions

How do I assign a unique identifier?

How do I provide rich metadata?

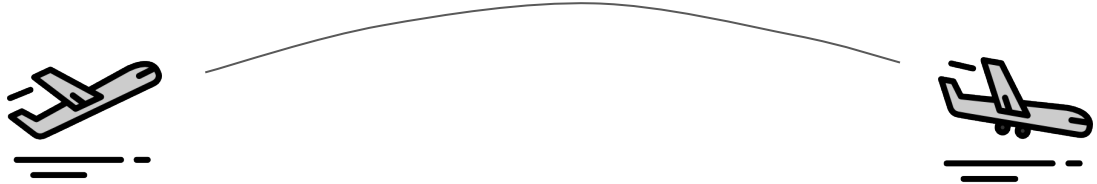
FAIR-BioRS Guidelines

FAIR-BioRS Guidelines

About

FAIR Biomedical Research Software (FAIR-BioRS) Guidelines

Minimal, actionable, step-by-step guidelines for complying with each of the FAIR4RS principles



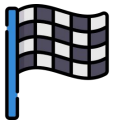
December 2021
Beginning of this effort

August 2023
Manuscript published

https://doi.org/10.1038/s41597-023-02463-x

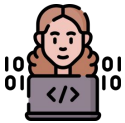
FAIR-BioRS Guidelines

Snapshot



Before starting

- Work from a version controlled system platform (e.g. GitHub)
- Select a license and include a LICENSE file



While developing the software

- Record dependencies
- Maintain a README

When releasing a new version

- Include metadata in codemeta.json, CITATION.cff, CHANGELOG
- Archive source code on a DOI-issuing repository like Zenodo

Full guidelines: https://doi.org/10.1038/s41597-023-02463-x

FAIR-BioRS Guidelines

Limitations

The FAIR-BioRS guidelines were developed only through inputs from the biomedical research community

Actionable FAIR4RS Task Force

Actionable FAIR4RS Task Force

Background

The Actionable Guidelines for FAIR Research Software Task Force
started in December 2024 under the Research Software Alliance (ReSA)

Establish actionable guidelines to make any research software FAIR in line with the FAIR4RS Principles

https://www.researchsoft.org/taskforces

Actionable FAIR4RS Task Force

New challenges compared to the FAIR-BioRS effort

How to develop domain-agnostic guidelines?

How to address open vs closed source software?

How to keep up with evolving standards?

Actionable FAIR4RS Task Force

Members

~12 active members

Various geographical locations (USA, Canada, Germany, Spain, UK, Netherland, etc.)

Various research domains (Biomedical, Data Science, Knowledge Representation, etc.)

Email me to join! bpatel@calmi2.org

Actionable FAIR4RS Task Force

Progress

Task 1: Interpretation of the FAIR4RS Principles (Dec 2024 - March 2024)

1. Identifier
2. Metadata for software publication & discovery
3. Standards inputs/outputs
4. Qualified references
5. Metadata for software reuse
6. License

Actionable FAIR4RS Task Force

Timeline

July 2025
Complete sub-groups investigations

September 2025
Release first draft of the guidelines for community feedback

January 2026
Publish first version of the guidelines

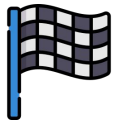
February 2026
Maintain guidelines

https://www.researchsoft.org/tf-actionable-fair4rs

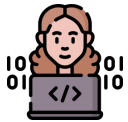
Making Software FAIR Today

Making Software FAIR Today

Take a few simple steps



Before Starting	
Work from a version controlled system platform (e.g. GitHub)	*Findable, Accessible, Reusable*
Select a license and include a LICENSE file	*Reusable*



While developing the software	
Record dependencies	*Interoperable, Reusable*
Maintain a README	*Reusable*

When releasing a new version of the software	
Include/update metadata in codemeta.json, CITATION.cff, and CHANGELOG	*Findable, Reusable*
Archive source code on a DOI-issuing repository like Zenodo	*Findable, Accessible*

Making Software FAIR Today

Even these simple steps could be time consuming

For instance, for each software release, you have to update release date, version number, and authors at multiple places including:

- codemeta.json
- CITATION.cff
- CHANGELOG
- Archival repository metadata

Making Software FAIR Today

Use Codefair

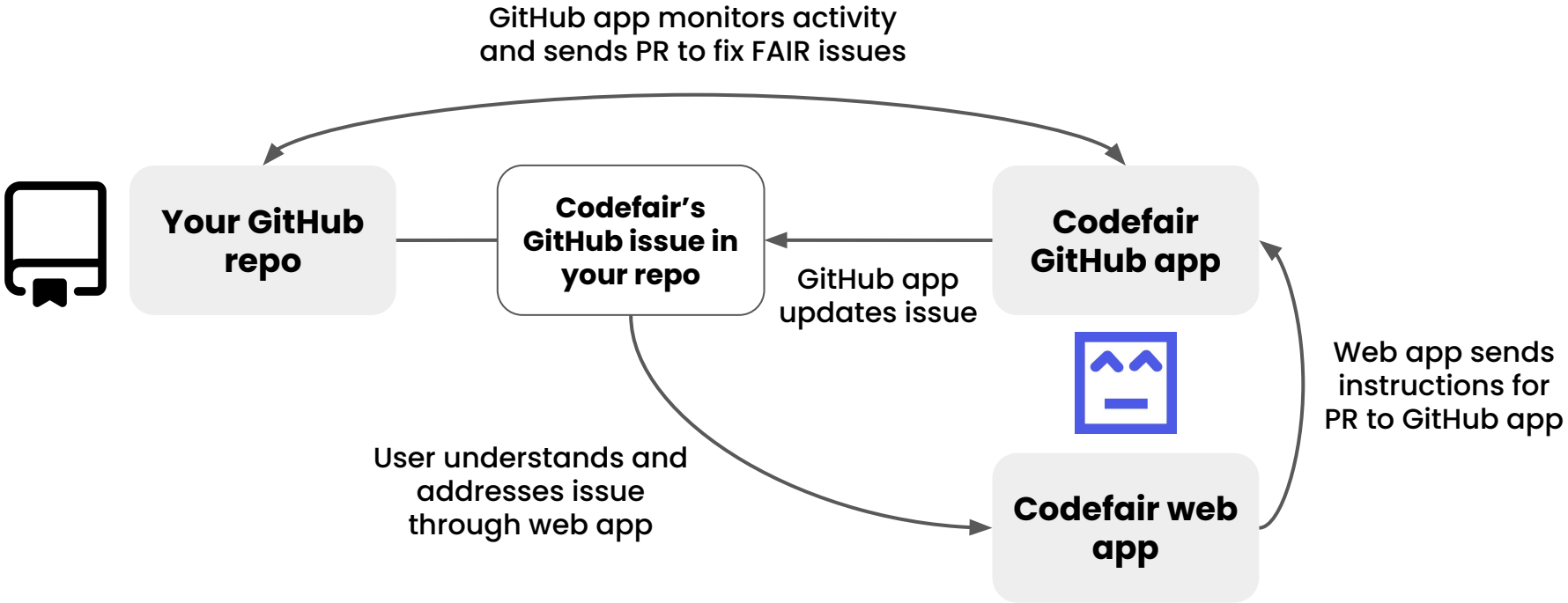
Open source and free GitHub app that acts as your personal assistant for making software FAIR

1. Install the app from GitHub Marketplace
2. Code and develop software as usual
3. Address GitHub issues and pull requests from the app

codefair.io

Making Software FAIR Today

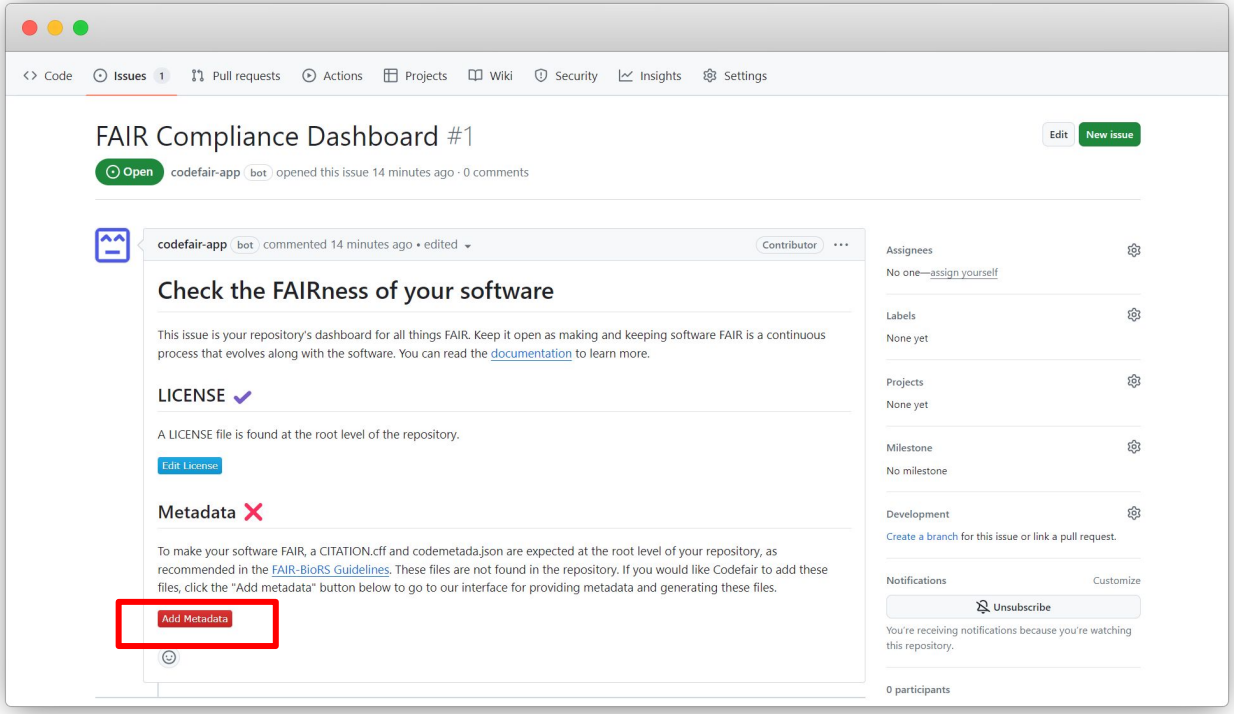
Codefair architecture



codefair.io

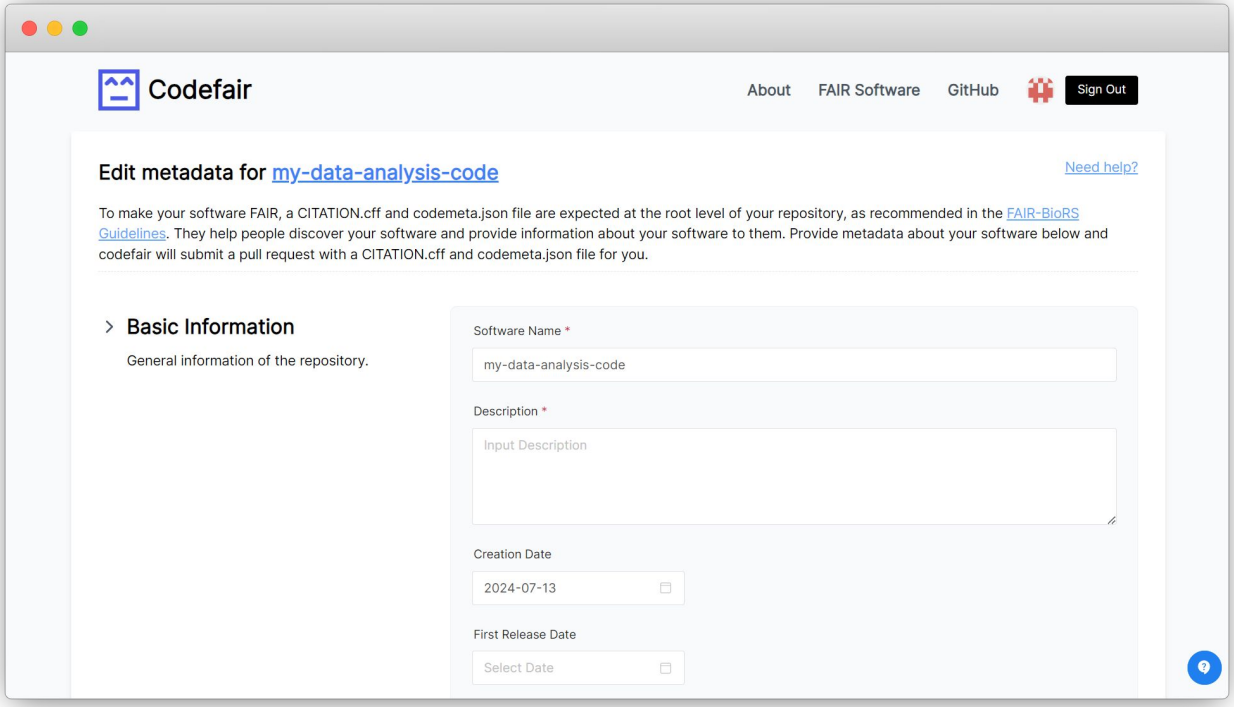
Making Software FAIR Today

Codefair GitHub issue dashboard



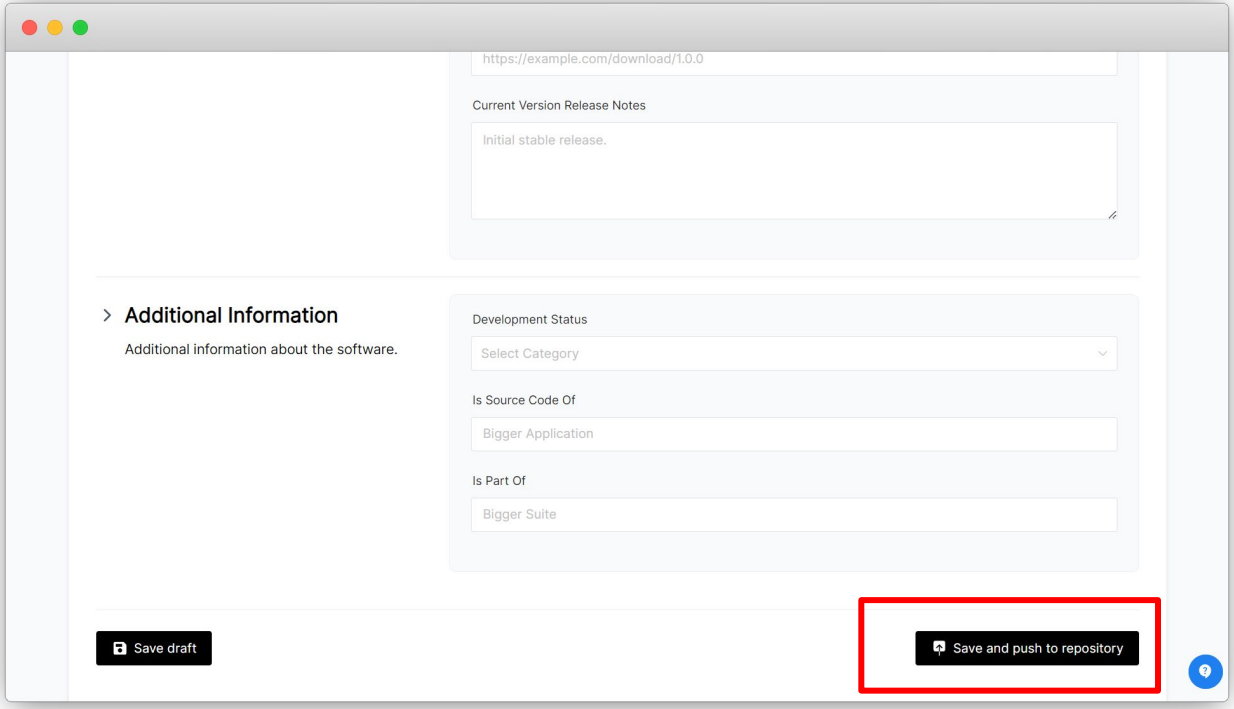
Making Software FAIR Today

Codefair UI



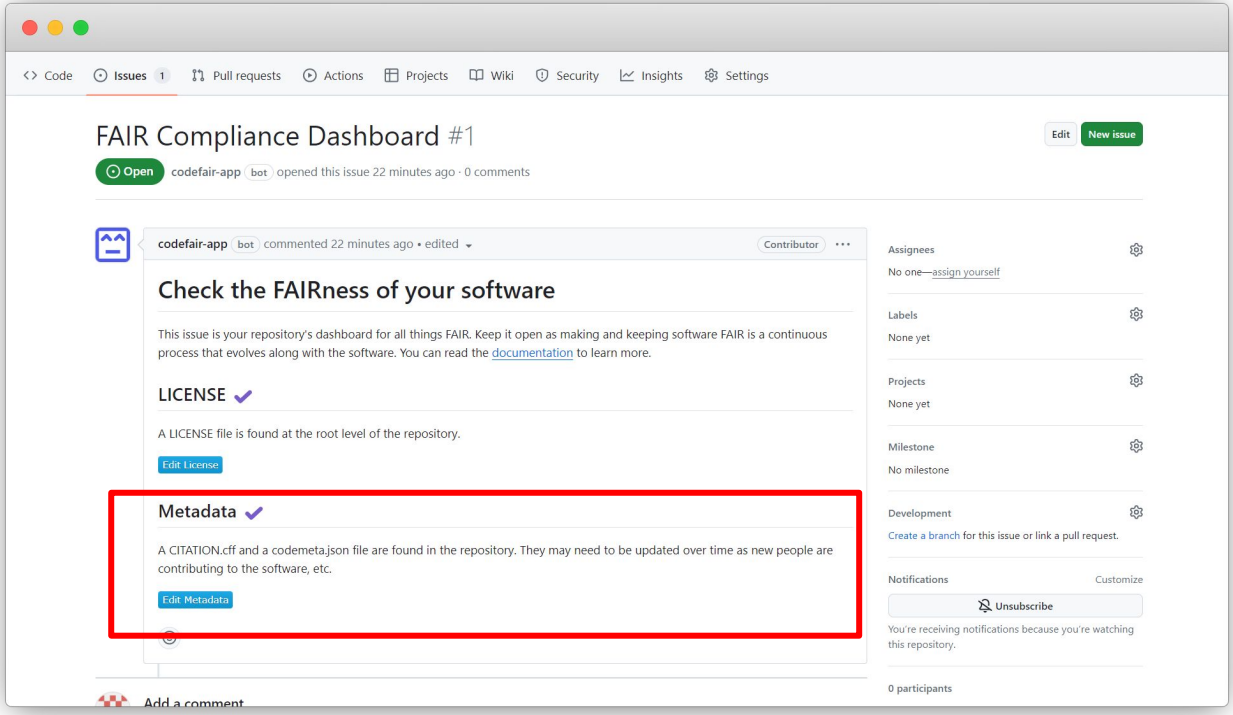
Making Software FAIR Today

Codefair UI



Making Software FAIR Today

Codefair GitHub issue dashboard



Making Software FAIR Today

Codefair - Future work

More features to make software FAIR (code formatting, bio.tools registration, etc.)

Features to support researchers beyond just FAIR (software quality, security, etc.)

Need your support - go to codefair.io and try it out!

Closing Comments

Closing Comments

Summary

Background	Research software is an essential element of research and making it FAIR is critical
Problem	The FAIR4RS principles only provide high-level instructions for making software FAIR
Solution	The Actionable FAIR4RS Task Force is developing minimal and actionable guidelines to make software FAIR
Support us!	Join the Task Force and use Codefair

Thank You!

bpatel@calmi2.org

fairdataihub.org

Find these slides and all resources here

bit.ly/FAIR-HPC