

# Making Research Software FAIR: Background and Practical Steps

Bhavesh Patel, Ph.D.  
Research Professor



CALIFORNIA MEDICAL  
INNOVATIONS INSTITUTE

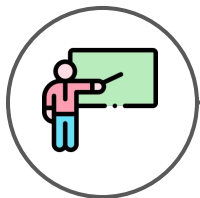


FAIR DATA  
INNOVATIONS HUB

# About This Presentation



45 min + Q&A



Learn why & how to make  
research software FAIR

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



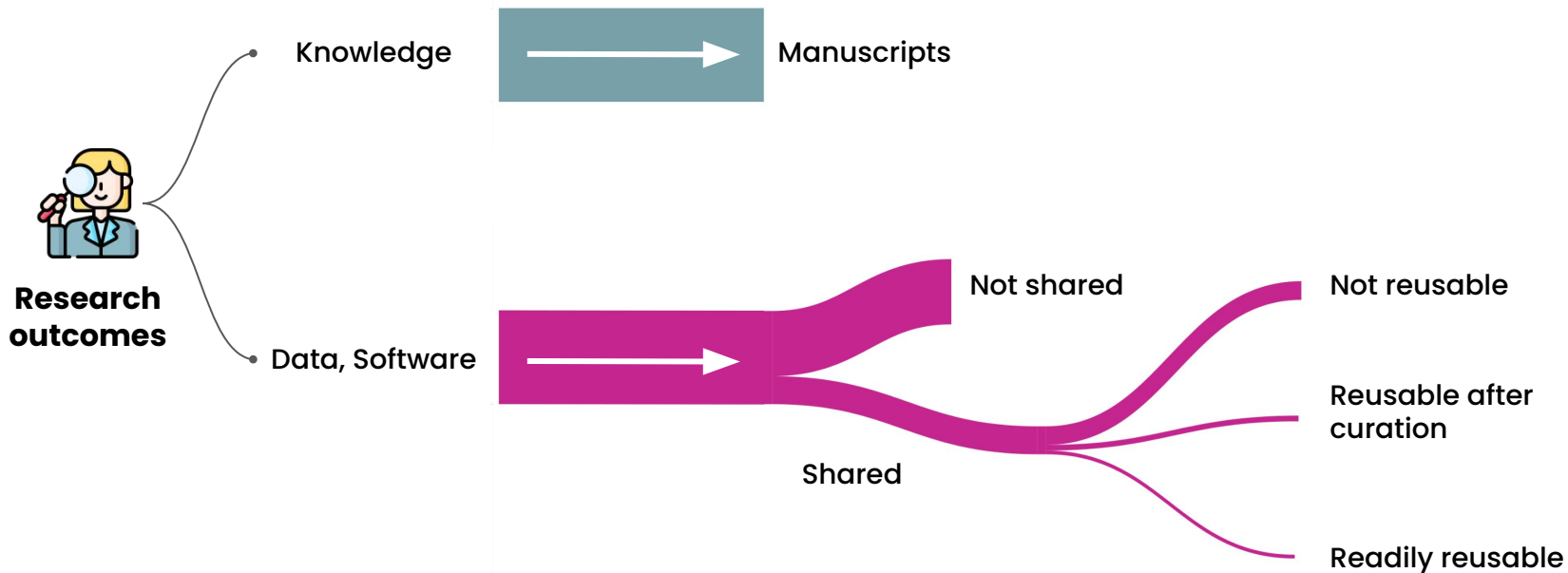
More and more research projects include development of research software



Research software is the main outcome of many research projects

# 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

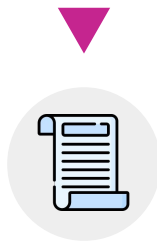


# **Making Software Reusable**

# 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)**

# 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

# Making Software Reusable

## FAIR principles – Adoption

The G20 logo consists of the letters "G20" in a bold, black, sans-serif font, centered within a light blue circle. This circle is itself centered within a larger, light gray rounded square.

**G20**

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



“Turning FAIR into Reality”  
report (2018)



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**

# 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?





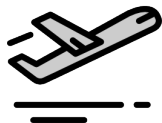
# **FAIR-BioRS Guidelines**

# 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

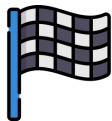


**August 2023**  
Manuscript published

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

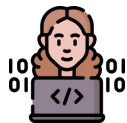
# FAIR-BioRS Guidelines

## 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

# 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



# 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](mailto:bpatel@calmi2.org)

# Actionable FAIR4RS Task Force

## Progress

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

17 Principles

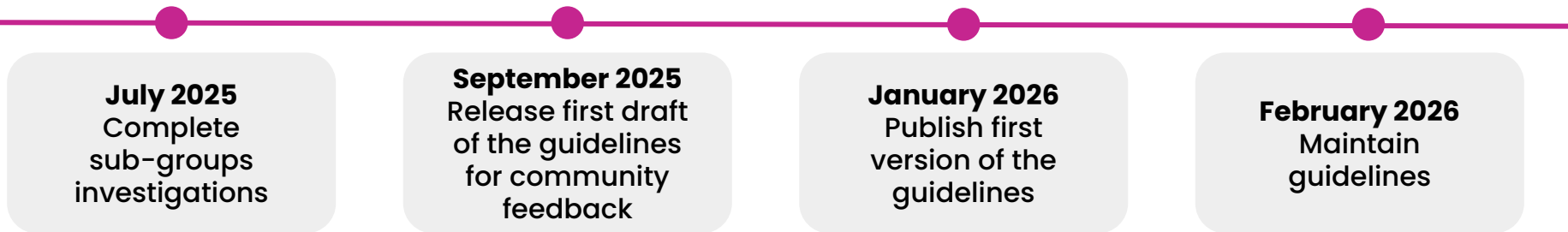
6 categories

Many questions!

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

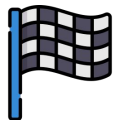


<https://www.researchsoft.org/tf-actionable-fair4rs>

# **Making Software FAIR Today**

# Making Software FAIR Today

Take a few simple steps



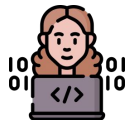
## Before Starting

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

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

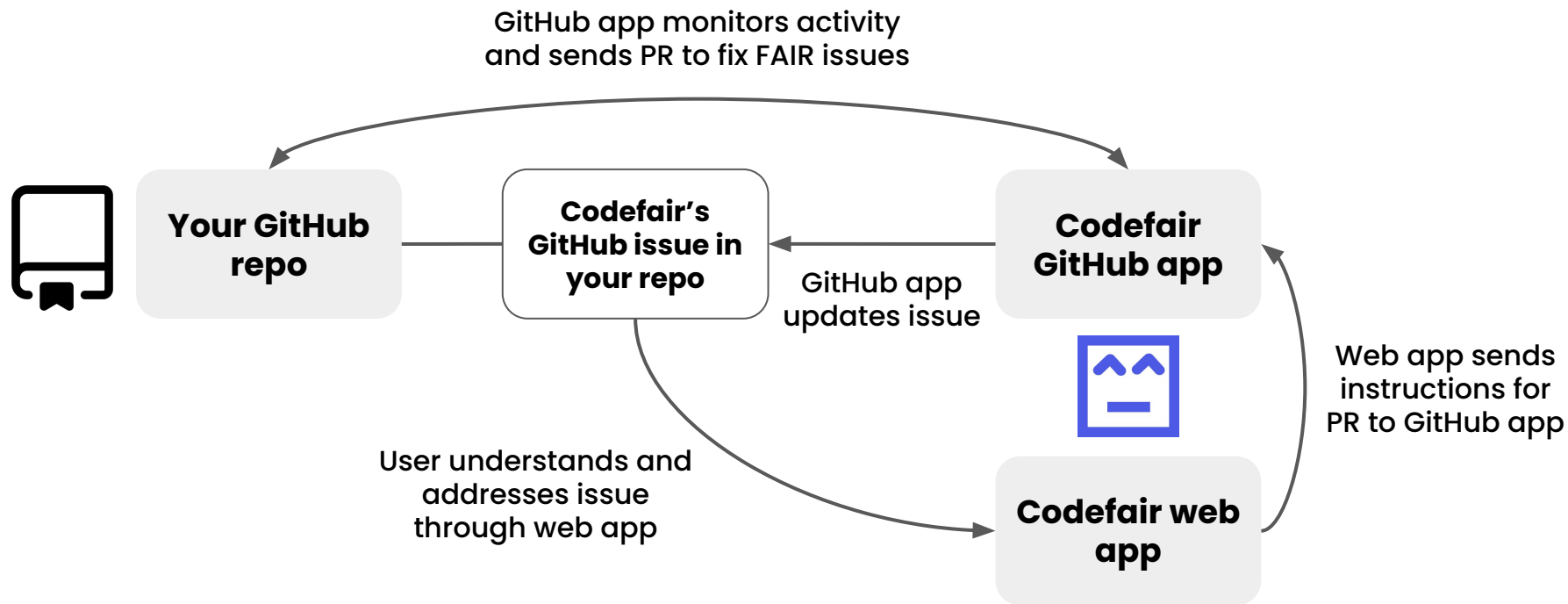
3

Address  
GitHub issues  
and pull  
requests from  
the app



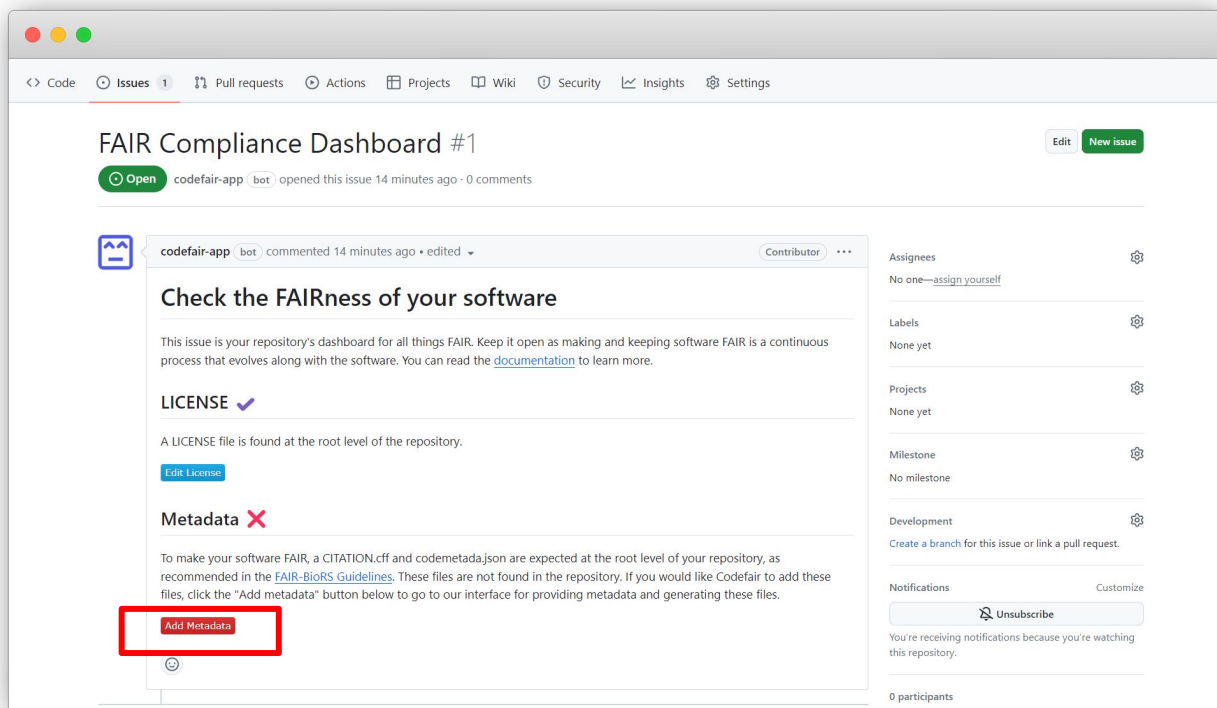
# Making Software FAIR Today

## Codefair architecture



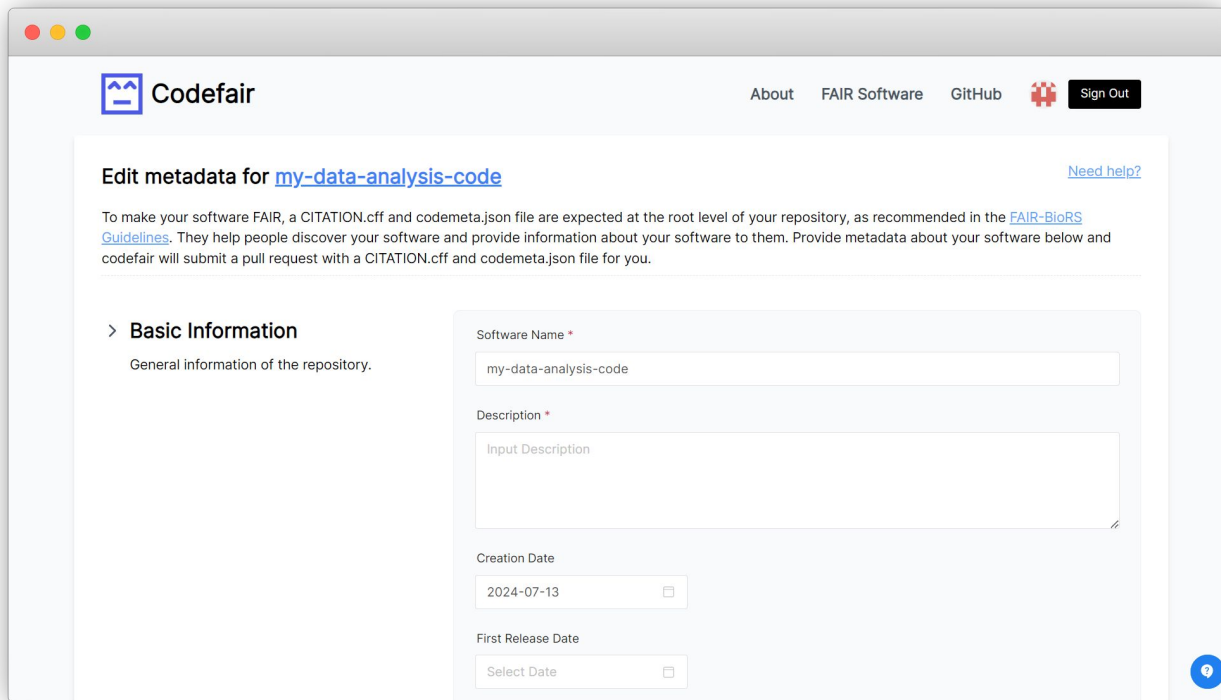
# Making Software FAIR Today

## Codefair GitHub issue dashboard



# Making Software FAIR Today

## Codefair UI



The screenshot shows a web browser window with the Codefair logo and navigation links (About, FAIR Software, GitHub, Sign Out). The main content area is titled "Edit metadata for my-data-analysis-code" with a "Need help?" link. Below this is a paragraph explaining the purpose of the metadata files. A section titled "Basic Information" is expanded, showing a form with fields for Software Name, Description, Creation Date, and First Release Date.

**Codefair** About FAIR Software GitHub Sign Out

Edit metadata for [my-data-analysis-code](#) [Need help?](#)

To make your software FAIR, a CITATION.cff and codemeta.json file are expected at the root level of your repository, as recommended in the [FAIR-BioRS Guidelines](#). They help people discover your software and provide information about your software to them. Provide metadata about your software below and codefair will submit a pull request with a CITATION.cff and codemeta.json file for you.

> **Basic Information**  
General information of the repository.

Software Name \*  
my-data-analysis-code

Description \*  
Input Description

Creation Date  
2024-07-13

First Release Date  
Select Date

# Making Software FAIR Today

## Codefair UI

The screenshot shows a web form for creating a software release. At the top, there is a text input field containing the URL `https://example.com/download/1.0.0`. Below this is a section titled "Current Version Release Notes" with a text area containing the text "Initial stable release.". A horizontal line separates this from the "Additional Information" section. On the left side of this section, there is a heading "> Additional Information" and a subheading "Additional information about the software.". On the right side, there are three form fields: "Development Status" with a dropdown menu showing "Select Category", "Is Source Code Of" with a text input containing "Bigger Application", and "Is Part Of" with a text input containing "Bigger Suite". At the bottom of the form, there are two buttons: "Save draft" on the left and "Save and push to repository" on the right. The "Save and push to repository" button is highlighted with a red rectangular border. A blue circular help icon is located in the bottom right corner of the form area.

https://example.com/download/1.0.0

Current Version Release Notes

Initial stable release.

> Additional Information

Additional information about the software.

Development Status

Select Category

Is Source Code Of

Bigger Application

Is Part Of

Bigger Suite

Save draft

Save and push to repository

# Making Software FAIR Today

## Codefair GitHub issue dashboard

The screenshot shows a GitHub issue page for 'FAIR Compliance Dashboard #1'. The issue was opened by 'codefair-app' bot 22 minutes ago and has 0 comments. The issue content includes a title 'Check the FAIRness of your software', a description of the dashboard's purpose, and two sections: 'LICENSE' and 'Metadata'. The 'Metadata' section is highlighted with a red box. The right sidebar shows various issue management options like Assignees, Labels, Projects, Milestone, Development, and Notifications.

FAIR Compliance Dashboard #1

[Open](#) codefair-app bot opened this issue 22 minutes ago · 0 comments

codefair-app bot commented 22 minutes ago · edited

### Check the FAIRness of your software

This issue is your repository's dashboard for all things FAIR. Keep it open as making and keeping software FAIR is a continuous process that evolves along with the software. You can read the [documentation](#) to learn more.

#### LICENSE ✓

A LICENSE file is found at the root level of the repository.

[Edit License](#)

#### Metadata ✓

A CITATION.cff and a codemeta.json file are found in the repository. They may need to be updated over time as new people are contributing to the software, etc.

[Edit Metadata](#)

Assignees  
No one—[assign yourself](#)

Labels  
None yet

Projects  
None yet

Milestone  
No milestone

Development  
[Create a branch](#) for this issue or link a pull request.

Notifications  
[Unsubscribe](#) Customize

You're receiving notifications because you're watching this repository.

0 participants

# Making Software FAIR Today

## Codefair – Future work



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



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



Need your support – go to [codefair.io](https://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](https://bit.ly/FAIR-HPC)