

CIRM Transparent Development Requirements

To support transparency, broad usability, and long-term impact of software developed under the Data Science and Software Engineering Awards (INFR9), all projects must adhere to best practices and the open source software development requirements outlined below. These requirements apply throughout the award period.

Requirements

Code is publicly accessible

The full codebase of the project must be hosted in a publicly accessible repository that allows users and contributors to review and download the code. GitHub is CIRM's preferred hosting platform. Applicants proposing to use a different hosting service must obtain CIRM approval.

A new user accessing the code for the first time should be able to download and install the code on their own and should find clear instructions on how to use the software and contribute to the project if desired. In order to achieve this, the repository must include the following elements in the root directory:

- A README file that includes:
 - A brief description of the project
 - Instructions for installing, running, and testing the code
 - Author and contributor information, with information on how to contact the development team
- Contributing guidelines
 - Detailed instructions on how to access and contribute to the project, such as how to file a bug report or suggest a new feature
 - A description of the issues tracking process
- A code of conduct
 - This can be based on an open standard such as [Contributor Covenant](#)
- The license file (see below)

Code and software are shared under an open source license

The software and all source code must be made available under a recognized open source license that allows for free and unrestricted reuse, modification, and redistribution. There are many examples of open source licenses; the [MIT License](#) is a popular choice, with several derivatives available for specific use cases. The Open Source Initiative provides a [definition](#) of open source licensing and a [database](#) of approved licenses.

Clear documentation is provided

Documentation is an integral part of the project that is developed and maintained alongside the code at all points in the project's timeline. Documentation should provide context and explanations to support the source code in the form of developer documentation as well as user documentation.

- Developer documentation should include:
 - A development roadmap
 - Human-readable code comments that explain what sections of code are doing and provide relevant context for why the code was written in a certain way
 - Information about benchmarking and testing the software
- User documentation should include:
 - Instructions for installing and running the software
 - A description of settings and parameters
 - How-to guides and tutorials

Documentation will evolve with the project; at the beginning documentation will likely only consist of the readme and code comments. As the project grows you should expand the documentation to fit the needs of contributors and users. Incorporating documentation from the start of the project will ensure that documentation does not become an afterthought and will make it easier for contributors to maintain and update the documentation as the project grows.

- Documentation resources
 - [How to document your research software](#)
 - [The Documentation System](#)
 - [Diátaxis documentation framework](#)
 - [Read the Docs](#)

Repository includes openly-available sample data

The repository containing the source code must also contain or link to at least one sample dataset that complies with CIRM's applicable data sharing and management requirements, including applicable FAIR and CARE principles. If the dataset is in a controlled access repository or requires IRB approval to access (e.g., patient data or EHR), this must be in place by the time of application. The available data should be sufficient for benchmarking, validation, and demonstration of the tool.

A deployment, adoption, and sustainability plan

The above requirements work together to support a software development process that is open, accessible, and inclusive, leading to an extensible and sustainable software product. However, making a tool openly available alone is not sufficient to ensure that the software is discovered and deployed by outside users. Feedback and contributions from the user community are essential for developing the most appropriate tool and ensuring that it will be used and maintained by other researchers. CIRM requires community engagement activities to solicit feedback on the software tool and inform its development to ensure relevance of the software tool to the field, improve its usability and accessibility, accelerate its development, align with emerging standards, and empower users to become contributors. These activities should be designed to position the software project for community maintenance by promoting awareness of the proposed tool and ensuring there is adequate documentation to enable community maintainers. The deployment, adoption, and sustainability plan should be a living document that evolves to reflect the project's status and needs. Elements of this plan should include:

- Community activities
 - Community engagement activities to promote the tool, attract users and contributors, and gather feedback; e.g., hackathons, virtual demo sessions, and community calls
 - A communication channel where contributors and users can interact with each other to discuss priorities, ideas, and use cases; where users can submit feedback; and where developers can announce and publicize updates to the software
 - Training and support offerings
 - A governance plan that accounts for expected future development and maintenance needs
- Technical activities
 - Plans for continued testing and quality assurance
 - Deployment and release processes
 - Issue tracking system

Ensuring sustainability requires a continuous effort to engage with and cultivate a user community that will eventually support a tool that is owned and maintained by a broad user base beyond the original development team. These activities can be kick-started and supported by engaging with existing communities focused on software development for biology research or regenerative medicine. Leveraging existing communities drives engagement and provides technical benefits by exposing developers to existing methods, tools, and previous learnings from teams exploring similar avenues. Adopting existing, well-documented standards can speed up development, save time, and facilitate extensibility of the tool.

Appendix

Suggested resources to include in the repository

- A [changelog](#)
- A [citation file](#) to enable others to easily and correctly cite your work
- An interactive demo in an executable environment such as [Binder](#)
- Use the [All Contributors bot](#) to acknowledge contributors

General open source resources

Apart from CIRM's requirements and resources, these are externally developed general resources that may aid in adhering to open source software development requirements set forth herein.

- [Open Source Guides](#)
- [The Turing Way](#)
- [The Open Source Initiative](#)
- [The Software Sustainability Institute](#)