Open-Source Software

Andrea Guerrieri

University of Applied Sciences Western Switzerland

30.05.2024

Field of Expertise/Background

Research and Teaching

- Embedded Systems and Reconfigurable System-on-Chip
- Electronics Design Automation (HLS)
- Security, Space, Astronomical applications

Books

- System-on-Chip Design in collaboration with Arm
- Application Enabled by FPGA-based Technology

Open-source tools for FPGAs

- MIPSFpga (https://github.com/MIPSfpga)
- Dynamatic (https://dynamatic.epfl.ch/)
- DynaRapid (not released yet)

Experience

- Serve as technical committee member and artifacts evaluator for EDA and FPGA communities such as DAC, FPGA, FCCM, ICCD
- Swissunivesities ORD program
- Senior member of IEEE and ACM

arm Education Media

Fundamentals of System-on-Chip Design on Arm Cortex-M Microcontrollers

TEXTBOOK

René Beuchat, Florian Depraz, Andrea Guerrieri, Sahand Kashani

SoC Design

electronics

Special Issue Reprint

Applications Enabled by FPGA-Based Technology

Edited by
Andres Upegui, Andrea Guerrieri and Laurent Gantel

mdpi.com/journal/electronics

MDPI

Introduction

Open-Source Software is a philosophy in software engineering.

Open-Source Software and Free Software both have common goals of collaboration and innovation but they are distinct in terms of why they are doing it and prioritize different aspects of software development and distribution.

Disclaimers

- If you are a software developer and you already published open-source code…
- Or you would like to understand how to efficiently use git from your IDE…

If you are interested in understanding how to start for sharing your project and code as open-source…

Agenda

- Introduction
- History
- Open-Source Definition
- Licenses
- Hosting Platforms
- Open-source code and Research Artifacts
- Artifact evaluation process of major IEEE/ACM conferences
- Good practices and Common errors
- Conclusions

History

The “open source” label was created during a strategy session held on February 3rd, 1998 in Palo Alto, California, after the announcement of the release of the Netscape source code.

The goal was to create an opportunity to educate and advocate for the superiority of an open development process.

They proposed a pragmatic, business-case grounds that had motivated Netscape to release their code illustrated a valuable way to engage with potential software users and developers, and convince them to create and improve source code by participating in an engaged community.

They wanted to have a single label that identified this approach and distinguished it from the philosophically- and politically-focused label “free software.”

Open Source Initiative

OSI was jointly founded by Eric Raymond and Bruce Perens in late February 1998. OSI was conceived as a general educational and advocacy organization to execute the mission agreed on at the Free Software Summit held in April 1998.

Eric Raymond

Bruce Perens

The Board accepted this general mission and decided to focus specifically on explaining and protecting the “open source” label. OSI was also supporting a petition to encourage the US government to use open source software in Jan. of 1999.

Open Source Definition (OSD)

Open source does not mean just provide access to the source code

The distribution terms of open-source software must comply with the 10 precise criteria:

1. Free Redistribution
2. Source Code
3. Derived Works
4. Integrity of The Author's Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Distribution of License
8. License Must Not Be Specific to a Product
9. License Must Not Restrict Other Software
10. License Must Be Technology-Neutral

Open Source Definition (OSD)

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources.

The license shall not require a royalty or other fee for such sale.

Open Source Definition (OSD)

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost, preferably downloading via the Internet without charge. **The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed.** Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

Open Source Definition (OSD)

4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. **The license must explicitly permit distribution of software built from modified source code**. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

Open Source Definition (OSD)

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program’s being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program’s license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

Open Source Definition (OSD)

9. License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open source software.

10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

Open Source Licenses

Navigate into the Open Source License

Open source licenses must comply with the Open Source Definition: software should be freely used, modified, and shared.

To be approved by the Open Source Initiative a license must go through the OSI's license review process.

open source initiative Approved License®

Today there are more than 200 licenses!

Here are three common open source license models.

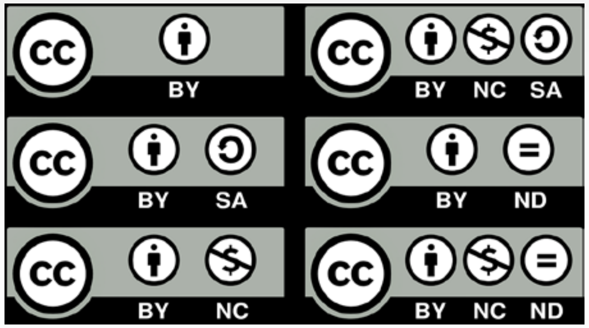
- Public domain
- Permissive
- Copyleft

Public Domain

This is the most permissive model. Anyone can modify and use the software without restrictions.

Creative Commons

Licenses give everyone from individual creators to large institutions a standardized way to grant the public permission to use their creative work under copyright law.



Be careful: but even if a component is free and comes without any legal strings attached, you should always make sure it is secure before adding it to your own codebase, especially for commercial purposes.

https://creativecommons.org/share-your-work/cclicenses/

Permissive

Contain minimal requirements about how the software can be modified or redistributed. This type of license is one of the most common and popular with free open source software.

MIT License

Created by the Massachusetts Institute of Technology, the MIT license is both one of the most popular licenses and also the most permissive. In contrast to many other open-source licenses, this one is quite short and clear. It states that the source code can be used in any way with the only requirement to preserve copyright and license notices.

BSD License

Berkeley Software Distribution (BSD) license grants developers similar rights to distribute the work without source code and under different license terms, but asks to include the copyright notice and license text in the copy of the software.

Apache License

The Apache license was released by Apache Software Foundation (ASF). As a permissive license, it grants developers the flexibility to distribute their work under any preferred license, as long as they credit the initial one and carefully document all modifications made to it.

Copyleft

Copyleft licenses, also known as reciprocal licenses or restrictive licenses, allow you to modify the code and distribute new works based on it, as long as you meet the requirements for redistribution under the same license.

Different licenses have different scopes. Some copyleft licenses allow you to release the modified code only; others require you to release the entire application under the same license.

GNU GPL (General Public License)

Any derivative work must be distributed under the same or equivalent license terms.

GNU LGPL (Lesser General Public License)

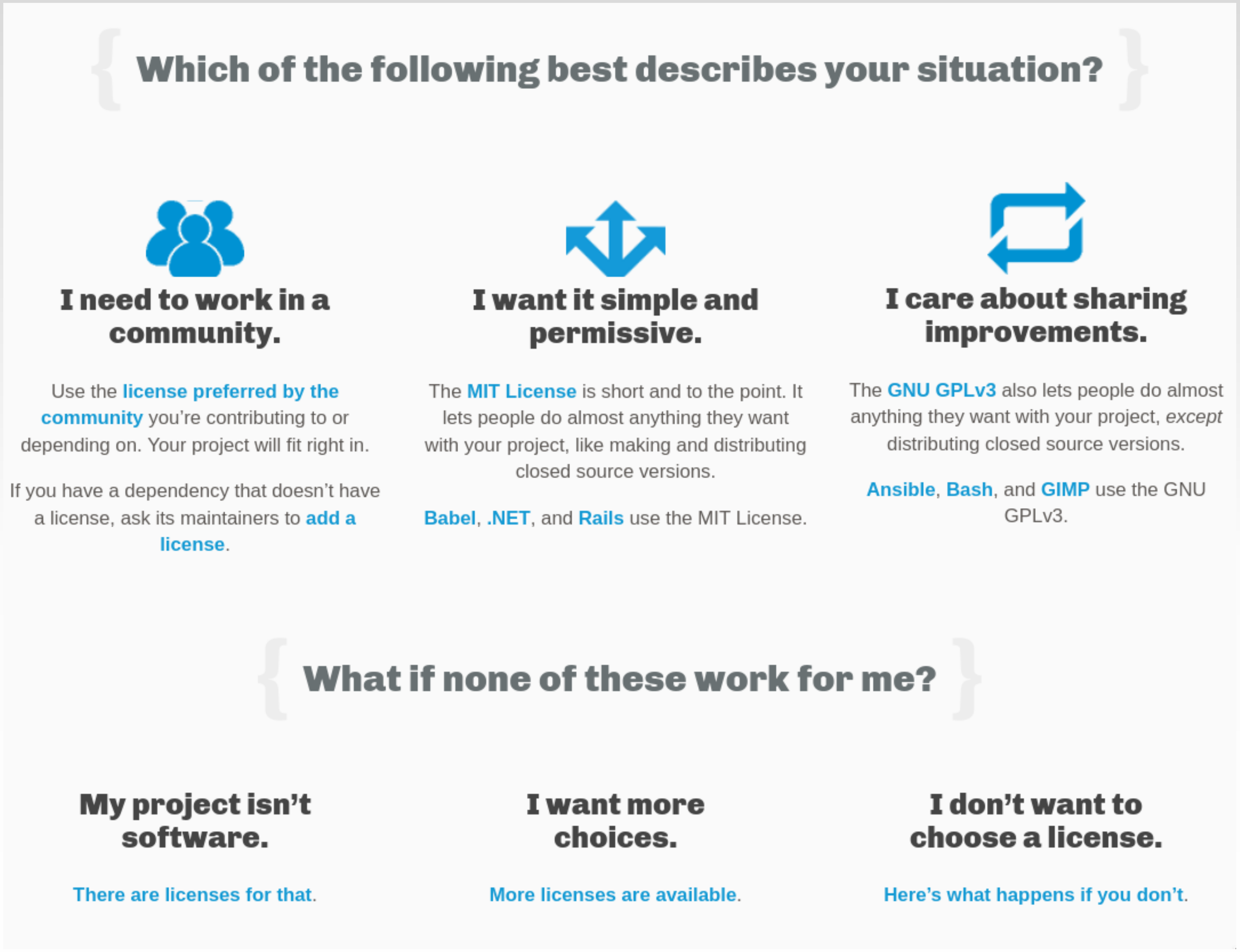
Primarily used for software libraries, although it is also used by some stand-alone applications.

GNU AGPL (Affero General Public License)

It is intended for software designed to be run over a network, adding a provision requiring that the corresponding source code of modified versions of the software be prominently offered to all users who interact with the software over a network.

https://www.gnu.org/licenses/gpl-3.0.en.html

Choose an Open Source License



{ Which of the following best describes your situation? }

I need to work in a community.

Use the license preferred by the community you're contributing to or depending on. Your project will fit right in.

If you have a dependency that doesn't have a license, ask its maintainers to add a license.

I want it simple and permissive.

The MIT License is short and to the point. It lets people do almost anything they want with your project, like making and distributing closed source versions.

Babel, .NET, and Rails use the MIT License.

I care about sharing improvements.

The GNU GPLv3 also lets people do almost anything they want with your project, *except* distributing closed source versions.

Ansible, Bash, and GIMP use the GNU GPLv3.

{ What if none of these work for me? }

My project isn't software.

There are licenses for that.

I want more choices.

More licenses are available.

I don't want to choose a license.

Here's what happens if you don't.

https://choosealicense.com/

Top 10 Open Source Licenses 2022-23*

Rank	License	OSI Approved
1	MIT License	YES
2	Apache License 2.0	YES
3	BSD 3-Clause “New”	YES
4	BSD 2-Clause "Simplified" License	YES
5	Generic Public Domain* Varies	YES
6	ISC License	NO
7	Creative Commons Zero v1.0 Universal	NO
8	GNU Lesser General Public License	YES
9	Mozilla Public License 2.0	YES
10	The Unlicense	YES

*source: Synopsys April 2024

Where to Host Open-Source Code?

Multiple Platforms

- GitHub
- GitLab
- Bitbucket
- Google Code: Home of Android, Google Web Toolkit, App Engine, and other projects
- Sourceforge: open-source repository
- Freecode: Unix software directory
- Apache Software Foundation projects; around 100 high quality and visibility projects
- Savannah: GNU software repository
- CPAN: Perl module directory
- CRAN: R Project archive
- CTAN: TeX software repository

GitHub

Key Strength

Large, active community of over 100 million developers. Used for open-source projects, remote teams, and collaborative development. GitHub offers an extensive range of features:

- **GitHub Issues:** project management tools are included alongside code repositories, including issue tracking.
- **Pull request review process:** allows product managers and leaders to effectively monitor pull requests.
- **GitHub Actions:** automate your build, test, and deployment workflows using secure CI/CD features.
- **Social features:** engage with followers, collaborate on code, and stay connected through notifications. Participate in a vibrant dev community.
- **Advanced security:** make apps and products more secure with GitHub Advanced Security, which scans your code for vulnerabilities.

https://github.com/

GitLab

Key Strength
A comprehensive suite of DevOps tools.
It excels in integrating security throughout the software development lifecycle.

- **Automation:** for developers that includes planning to governance
- **GitLab Duo:** an AI-powered code assistant that helps developers write code faster with fewer errors.
- **Custom workflows:** tailor a project management system in GitLab to fit your internal processes.
- **Scalability:** can handle enterprise-level dev cycles, DevSecOps, DevOps, security, and deployments.
- **Container registry:** manage your software containers and their code.
- **Code review:** get a comprehensive view of your code, ongoing work, and maintain a robust CI/CD pipeline.
- **Customizable issue tracking and PM tools:** manage projects directly in GitLab or integrate with existing PM and bug tracking tools.
- **Security scanning:** ensure your code is secure both pre- and post-deployment, protecting your products from vulnerabilities.

https://about.gitlab.com/

Bitbucket

Key Strenght
integrations with Jira, Confluence, Trello, and other Atlassian products.

Other features that BitBucket has that the others don’t include:

- **Integrated issue tracking:** project management solutions, natively compatible with Jira.
- **Native support for Mercurial**: while GitHub & GitLab only support Git.
- **Private code repositories:** (similar to the others, but Atlassian-focused).
- **Code insights:** provide team members with detailed visibility into code quality through performance metrics and analytics.
- **Advanced branch permissions:** for code review and greater security.

https://bitbucket.org/

Which one is the best?

GitHub is usually the first choice for many use cases.

GitLab has the widest and most robust suite of tools for DevSecOps teams.

BitBucket is useful for integration with Jira or Confluence integrations or are developing software specifically for the Atlassian ecosystem.

In academia, except for internal or confidential projects, GitHub is the most used.

Open-source Software and Research in Academia

Open source software vs Research Artifacts

The goal of this initiative is to promote reproducibility of published results by highlighting papers supported with open-source code.

These papers are identified by badges promoted by ACM and IEEE.

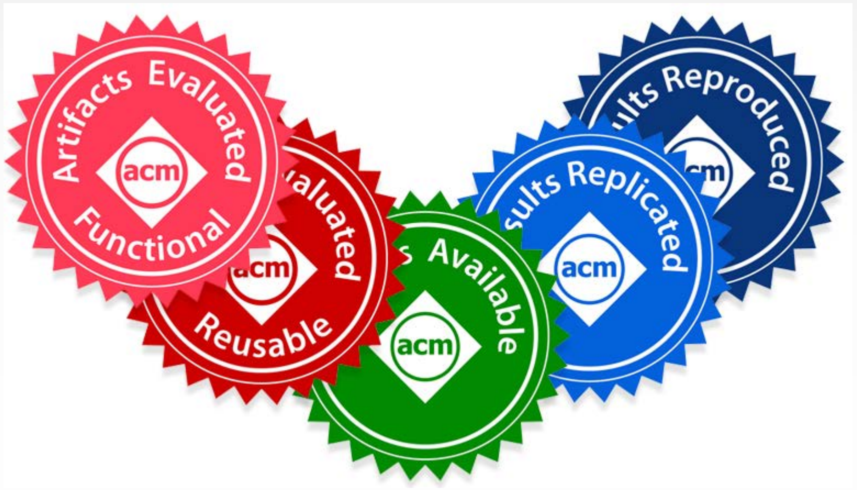
Impact in the research community

1. Reproducibility of the results (validation)
2. Reuse of the results to move forward your research
3. Make stronger you research impact

ACM Artifact Evaluation

A paper can be awarded with multiple badges from among the following choices:

- Artifacts Evaluated (Functional)
- Artifacts Evaluated (Reusable), Artifacts Available, Results Replicated
- Results Reproduced.

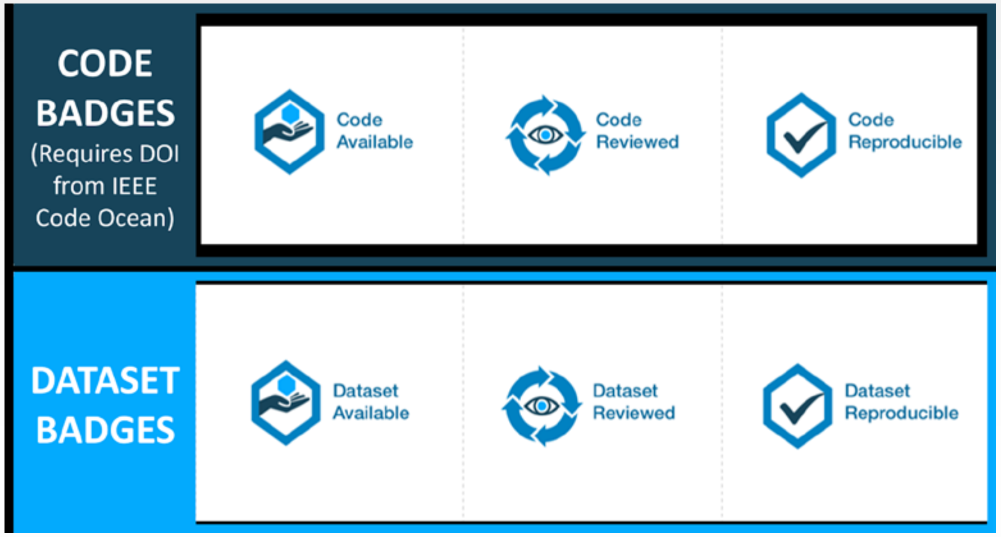


https://www.acm.org/publications/policies/artifact-review-and-badging-current

IEEE Artifact Evaluation

A paper can be awarded with multiple badges from among the following choices:

- Code and Dataset Available
- Code and Dataset Reviewed
- Code and Dataset Reproducible



https://ieeexplore.ieee.org/Xplorehelp/overview-of-ieee-xplore/about-content#reproducibility-badges

Good Practices and Commons Mistakes

Good Practice

When you submit a paper, be ready to provide research artifacts.

You will need to create a persistent link with a DOI.

Running examples and datasets must be available as well (consider putting pre-compiled binaries on your repository).

Common Mistakes

The code is open-source, but part of it uses libraries that cannot be open-sourced.

The code is open-source, but no documentation on how to compile nor run the code.

Third-parties dependencies: everything was working fine until the last version.

Conclusion

Check-list to open-source your project

1. Choose the license
2. Choose the hosting platform
3. Document your code, readme, running examples
4. Pay attention before accepting contributors

There are multiple ways for releasing open-source code

Everything depends on your specific use case and application

Thank you for your attention

Any Questions?