

TYPES OF CONTRIBUTIONS

In this chapter, we will discuss the types of contribution that open source projects involved.

Majorly, there are two types of contributions. Which are;

1. Non-coding type of contribution
2. Coding type of contribution

A. Non-Coding Type Of Contributions

Non-coding contributions are valuable contributions made to open source projects that do not involve writing or modifying code. These contributions are essential for the overall health and success of open source communities and projects. They help improve documentation, user experience, project management, and community engagement.

Here are some common types of non-coding contributions

1. Documentation
2. Graphic design
3. Blog writing

1. DOCUMENTATION

Documentation is a crucial type of open source contribution that involves creating, improving, or maintaining the written resources that accompany open source software projects. Good documentation can make a project more accessible and user-friendly, and it can help attract new contributors.

Contributing to documentation in open source projects is a valuable way to make a meaningful impact, even if you're not a developer. Clear and well-organized documentation enhances the overall quality of the project and fosters a more welcoming and inclusive open source community.

Importance of good documentation

Ease of Use: Well-documented open source projects are more accessible to users and developers. Clear and comprehensive documentation helps users understand how to install, configure, and use the software effectively.

Onboarding: Documentation is essential for onboarding new contributors to a project. It provides them with the necessary information to get started, including project goals, coding standards, and development workflows.

Troubleshooting: When users or developers encounter issues or bugs, documentation often contains troubleshooting guides and FAQs that can help them resolve problems without needing to contact the project maintainers.

Knowledge Sharing: Documentation is a means of sharing knowledge and best practices within the open source community. It allows experienced developers to pass on their expertise to others and ensures that knowledge is not lost as contributors come and go.

Community Growth: High-quality documentation can attract more users and contributors to an open source project. It builds trust and confidence in the project, which can lead to a larger and more active community.

Maintainability: Documentation can also encompass guides on how to contribute to the project, including coding guidelines and codebase architecture. This helps maintain consistency and makes it easier for developers to contribute effectively.

If you're interested in contributing to documentation, here are a few tips:

- Start by reading the project's documentation and getting familiar with the project's code base. This will help you understand what the project needs and how you can contribute.
- Find a task that's a good fit for your skills and interests. If you're not sure where to start, ask the project's maintainers for suggestions.
- Be patient and persistent. It takes time to write good documentation. Don't be afraid to ask for help if you get stuck.

Here are some specific examples of documentation contributions:

- Writing a README file that explains how to install and use the project

Example: Here is an example of such.

Create a Bootstrap Example

<https://github.com/Mirrorful/mirrorful/pull/321>

- Creating a tutorial that walks users through the project's features

Example: Here is an example of such

adding chart

<https://github.com/rupali-codes/LinksHub/pull/1632#event-10288094144>

- Updating the project's documentation to reflect new changes

Example: Here is an example of documentation that indicate that.

Add Dev Cards to Getting Started

<https://github.com/open-sauced/docs/issues/156>

- Translating the project's documentation into other languages

Example: Here is an example of such.

Translate the course to other language :

<https://github.com/open-sauced/intro/issues/16>

- Fixing typos and grammatical errors in the documentation

Example:

- Improving the formatting of the documentation

Example: Here is an example of such.

Incorrect Link Destination in Documentation

<https://github.com/open-sauced/docs/pull/144>

- Adding missing information to the documentation

Example: Here is an example of such

Add More Information to Use This Template in the README

<https://github.com/open-sauced/100-days-of-oss-template/pull/8>

Contributing to documentation is a great way to help open source projects and make them more accessible to everyone. If you have the skills and the interest, I encourage you to get involved!

2. GRAPHIC DESIGN

Graphic design is a type of open source contribution that involves creating visual assets, such as logos, icons, illustrations, website designs, and other graphical elements, for open source projects. These visual assets play a significant role in enhancing the overall user experience and branding of the project. Graphic design contributions are important because they can make open source software more appealing, user-friendly, and recognizable.

Here are some aspects of graphic design as a type of open source contribution:

Logo and Branding: Designing a unique and memorable logo for an open source project can help establish its identity and brand. A well-designed logo can make the project instantly recognizable and give it a professional image.

Example: Insights social card generation

<https://github.com/open-sauced/opengraph/pull/51>

Icon Design: Icons are essential for user interfaces. Graphic designers can create custom icons for software applications or websites, improving the visual appeal and usability of the project.

Example: Redesign footer

<https://github.com/open-sauced/opengraph/pull/51>

Website Design: Open source projects often have websites to provide information, documentation, and download links. Graphic designers can contribute by designing website

layouts, graphics, banners, and other visual elements to make the site more attractive and user-friendly.

Example: Add highlights to user profile page

<https://github.com/open-sauced/app/pull/859>

User Interface (UI) Design: For software projects, UI design is crucial. Designers can create mockups, wireframes, and high-fidelity designs for the user interface, making it more intuitive and visually appealing.

Example: Form page UI implemented

<https://github.com/zillBoy/pokemon-app/pull/6>

User Experience (UX) Design: UX designers focus on improving the overall user experience by considering how users interact with the software. This includes designing user flows, navigation, and usability testing.

Example: Improved the user experience of the highlights input form

<https://github.com/open-sauced/app/pull/1362>

Accessibility Design: Ensuring that the project's graphical elements are accessible to all users, including those with disabilities, is an important aspect of graphic design. Designers can create accessible color schemes and ensure that visual elements are perceivable and operable.

Example: Redesign footer

<https://github.com/cesko-digital/pomahejukrajine-web/pull/222>

Responsive Design: Designing visuals that work well on various devices and screen sizes is crucial for modern web applications. Responsive design ensures that the project looks good and functions properly on both desktop and mobile devices.

Example: Enhance feed layout responsiveness

<https://github.com/open-sauced/app/pull/1553>

3. BLOG WRITING

Blog writing as a type of open source contribution involves creating and publishing blog posts, articles, or tutorials related to open source software, projects, technologies, or best practices. These written contributions serve various purposes within the open source community and can be valuable in several ways.

Blog writing can be a great way to contribute to open source projects, even if you don't have any coding skills. You can write about your experiences using the project, or about the latest developments in the project. You can also write about the project's history, its community, or its impact on the world.

Contributing blog posts to open source projects or community websites is a valuable way to share knowledge, support the community, and raise awareness about important topics. It can also help open source projects reach a wider audience and attract new contributors. Many open source communities welcome blog contributions and have established platforms or publications where contributors can share their insights and expertise.

Here are some specific examples of how blog writing can be used to contribute to open source projects:

- A blog post could be written about the latest features of an open source project, such as a new version of the software or a new plugin.

Example: Announcing the 0.2.0 release of Analog

<https://dev.to/analogjs/announcing-the-020-release-of-analog-aa1>

- A blog post could be written about how to use an open source project, such as a tutorial on how to install and configure the software.

Example: How To Make a Delicious Contributing Guide

<https://dev.to/opensauced/how-to-make-a-delicious-contributing-guide-4bp3>

Example: Open Source 101: A Beginner's Guide to Getting Started

<https://opensauced.pizza/blog/open-source-101-a-beginner's-guide-to-getting-started>

- A blog post could be written about the impact of an open source project, such as how it has been used to make a difference in the world.

Example: Retaining the best engineers

<https://opensauced.pizza/blog/retaining-the-best-engineers>

B. Coding Type Of Contributions

Coding contributions in the context of open source software development involve writing, modifying, or improving the actual source code of an open source project. These contributions are at the core of open source development and are crucial for building and maintaining software

Coding contributions are the heart of open source development, as they directly impact the functionality and quality of the software. Contributors collaborate with project maintainers and the community to ensure that their code contributions align with project goals and coding standards. Open source projects typically have guidelines and processes for accepting and integrating coding contributions, such as code reviews and continuous integration.

Here are some specific examples of how coding can be used to contribute to open source projects:

Writing New Code: This is the most common type of coding contribution. Developers write new code to add features, fix bugs, or enhance the functionality of an open source project. These contributions are typically submitted in the form of pull requests (PRs) or patches.

Example: show reactors who reacted to highlights

<https://github.com/open-sauced/app/pull/1591>

Bug Fixes: Identifying and fixing bugs is a critical coding contribution. Developers locate issues within the codebase, diagnose the problem, and then write code to correct it. Bug fixes improve the stability and reliability of the software.

Example: Lint check happening two times on commit

<https://github.com/open-sauced/app/pull/1635>

Code Refactoring: Refactoring involves restructuring and improving existing code without changing its external behavior. This helps enhance code readability, maintainability, and scalability, which are vital for the long-term health of the project.

Example: CalcDistanceFromToday function

<https://github.com/open-sauced/app/pull/1633>

API Design and Maintenance: For projects with public APIs, developers can contribute to designing, maintaining, and improving the API to ensure it remains stable and developer-friendly.

Example: Refresh token

<https://github.com/IridiumIdentity/iridium/pull/134>

Feature Development: Developers contribute by implementing new features or functionalities in response to user needs or project goals. Feature development often involves planning, designing, and coding new components or capabilities.

Example: Added the contact page

<https://github.com/TechIsHiring/techishiring-website/pull/53>

Performance Optimization: Coders may work on optimizing the software's performance by refactoring code, improving algorithms, or optimizing resource usage. This type of contribution aims to make the software run more efficiently.

Example: Overflows in insights page on md screen size

<https://github.com/open-sauced/app/pull/1590>

Testing and Quality Assurance: Writing automated tests, including unit tests, integration tests, and end-to-end tests, is a coding contribution that ensures software reliability. Testing helps catch and prevent regressions and issues.

Example: Add force login component to design system

<https://github.com/open-sauced/app/pull/1330>