



The Four Opens: Open Source Beyond the Code

Ildikó Vancsa, Open Infrastructure Foundation

This article describes a set of guiding principles that open infrastructure communities, such as OpenStack, use to create and maintain balanced ecosystems around projects and navigate the challenges and intricacies of open collaboration.

The 20th century was a time of breakthroughs in technology through which humanity began to take advantage of tools and automation to improve the quality of life. It was also the era when digital technologies began to enable the rise of computers, information technology, and open source software. As we got past the millennium, technology became more than just nice to have: it became necessary for the further evolution of humanity.

In practice, this translates to depending on hardware and software infrastructure through daily use and needing to support and influence their evolution. Access to infrastructure and its evolution path is becoming part of

basic human rights. A trivial way to provide it is through open source, which means people have access to infrastructure, including the ability to get involved at any stage of the lifecycle of its building blocks. This is what we call *open infrastructure*.

The Open Infrastructure Foundation (formerly known as the OpenStack Foundation)¹ has a mission to build communities that write code that runs in production all around the globe to power critical infrastructure as well as services in high demand. This article introduces the guiding principles—the Four Opens²—that the Open Infrastructure Foundation and its communities follow to navigate the (sometimes stormy) waters of open source software development, open collaboration, and more.

OPEN SOURCE: BEYOND CODE

The term *open source* does not require explanation anymore. However, it is important to note that we have entered an era when, in certain areas, such as infrastructure, it is becoming the mainstream way to develop software

FROM THE EDITOR

Community open source software is the backbone of the Internet and other infrastructure. But what exactly does community mean? There are many definitions beyond that of the Open Source Initiative. On the basis of experience with OpenStack and related projects, in this article, Ildikó Vancsa, of the Open Infrastructure Foundation, codifies community as the Four Opens, a riff on Stallman's four freedoms. These insights follow on the heels of a previous article in this column that discussed the many challenges of the OpenStack project. It is good to see how such issues can be resolved in a constructive way. Again, progress! Happy hacking, be open, and be safe!—Dirk Riehle

and design hardware,⁴ as opposed to mostly hobbyist projects. At the same time, there is a growing number of single-vendor projects⁵ that are rather tightly controlled by one organization that employs the majority of the contributors and has the most influence over the road map, design and development processes, and so on. This is not a new phenomenon, nor is it always intentional. We all need to recognize that building a diverse community around a project has to be a goal set early on. From there, it remains a continuous effort to make it a reality.

Recognizing this challenge, the OpenStack community decided to prioritize organizational and other types of diversity by codifying a set of principles for all contributors to follow—the Four Opens. Once communities began to grow, balancing these environments, where contributors came from different cultures, spoke different languages, and often worked for competing companies, grew more difficult. Striking that balance is even tougher in newly forming communities, where the majority of the contributors may be new to the concept and practices of open source software development.

To overcome these challenges, the Four Opens provide guidance to create and maintain an environment where contributors feel safe, empowered, and included to collaborate and work toward a shared mission and set of goals. These principles became fundamental to the processes that the Open Infrastructure Foundation and all its communities

follow. So, what are the Four Opens, and what do they mean in practice?

THE FOUR OPENS

The Four Opens are open community, open design, open development, and open source. Each represents a crucial component of navigating the

immediately expect it to be accessible one way or another. It is easy to forget the people who work on the design, implementation, and maintenance of software, even though software would not exist without them. The dynamics of people working together are always complicated, even in a closed corporate environment. When you transform that environment into an open and thus public space, you remove boundaries, such as safety and common understanding of processes. Public environments can feel intimidating and unsafe, even for experienced contributors.

The open community is shedding light on the difficulty of creating and maintaining a space that is accessible and inclusive to anyone and everyone. Beyond being a safe place, a community has to be a level playing field for

The Four Opens provide guidance to create and maintain an environment where contributors feel safe, empowered, and included.

dynamically changing environment of an open source community. You might think all this is self-explanatory and would happen organically. What do we gain from these guiding principles?

While the accessibility of open source code got established a while back, forming and organizing a group of people in a public environment is a delicate process in which you have to maintain the accessibility of not only the software but the community, as well. Established contributors and newcomers need to be able to access and understand processes and how they fit in. The Four Opens were defined to set the foundational aspects to enable individuals from different organizations to work together in a sustainable and efficient manner. Let's examine each one means and how it helps.

OPEN COMMUNITY

When we talk about open source, most people associate it with code and

everybody regardless of gender, race, company affiliation, and many other factors. This is fundamental to ensure that a group of individuals who are interested in enhancing and maintaining a piece of software can exist as a balanced, sustainable, and productive ecosystem. As part of this, those who have ambition need to be able to rise, take responsibilities, and assume leadership positions. There is no room for reserved seats in governance bodies and for favoring one person's contributions over those of others due to money provided by a contributing organization. Contribution is the only currency. A common way to choose leaders within a project is to follow a meritocratic model and have principals elected from the community by the community. If contributors can grow and rise to take on more responsibilities, they will treat projects as their own. At the same time, leaders have to regard their position as

a set of responsibilities and serve the community, as opposed to maintaining power and control.

What makes a great community is that contributors believe in its mission and share the same vision and goals. The only way to achieve this is to ensure that a community's processes, tools, and artifacts are open and available to everyone to read, understand, shape, and maintain. An open community also creates opportunities for users to find entry points and become part of a feedback loop. While this might not seem like a mandatory step, the longevity and sustainability of an open source project highly depend on close relationships between those who build and maintain the software and those who use it.

There are multiple factors that motivate contributors to join and keep participating in a community. Some

as you might think. As one of the latest trends in this space, companies often begin design and development work in-house and eventually decide to release a project under an open source license. The motivation for this can be different, but the challenges of building a community and ecosystem around those projects are the same.

These challenges also exist when an open source project has some level of organizational diversity at the time of its launch. You need to achieve a high level of transparency and shared responsibility within your community to ensure that it is a balanced and sustainable ecosystem. Especially for projects that begin as single-vendor communities, it is challenging to remove the perception of control that the initiating organization has at the beginning. It becomes crucial early on to provide visibility into the entry points to the

As I mentioned, having users involved in a community is crucial for its long-term success. In this example, the predictability of the steps throughout a release cycle provides an opportunity for users to provide feedback and influence a project's road map. Once newcomers are familiar with the processes that a community uses and begin to feel more comfortable taking the next steps, they need to have a chance to take on not just tasks but also responsibilities. Having access to the entirety of the design process paves the way for contributors in a community to be heard and valued.

OPEN DEVELOPMENT

We arrive at the next step, the development process itself. You might think that by this stage, everything is straightforward and that there isn't much more to talk about, but you might be wrong. Transparency is a critical factor during the development phase, including the clear and visible documentation of processes, practices, and expectations as well as open meetings and other communication channels. Writing code can be considered an art, and writing documentation is like writing prose. In other words, creating high-quality software is a subjective process, and with that, it can be a challenging experience in an open and public environment.

Tools and automation ensure the validation and verification of software that is created by a diverse group of people. Once it is proved by automated tests that a change to code works and delivers the expected functionality, there is less room to argue about the nuances of coding styles and formatting. It is a balancing act for contributors to find the right approach to apply coding techniques, educate newcomers, and accept new concepts and preferences for how code should be structured and formatted and what the most efficient methods are for a particular language in a particular environment. While coding styles are always up for debate and can become a religious fight,

It is easy to forget the people who work on the design, implementation, and maintenance of software, even though software would not exist without them.

people are driven by their employers to contribute on behalf of a company, and some have more personal reasons. Successful communities always create an environment where individuals feel welcome and valued and where they believe they can make an impact. The only way to achieve this is to ensure that design and development processes are open and transparent, where no one is in control and everyone is in control at the same time. Now, let's dive into open design.

OPEN DESIGN

Open design is essential to achieve transparency and, with that, ensure that no one person, organization, or group of people maintains control over a project. While it sounds self-explanatory, this step is not as straightforward

community. How can someone report a bug? How can someone contribute code or fix something in the documentation?

When we focus on design, the question to ask is, How can a contributor suggest new ideas and influence the software's road map? As an example, the OpenStack community uses a time-based release cycle with a clear and visible timeline for contributors to propose features and enhancements. In most cases, feature proposals need to be in the form of specifications; the community calls them *specs*. The information to include in a spec is guided by a template, which helps new and established contributors describe their ideas in a standard way. With the process and timeline in place, it becomes easy for anyone to approach the community with an idea and begin to participate.

automated testing can provide much-desired objectivity.

Once you have the tooling in place, you have two more steps to take. First, you need to ensure that the tools the community uses are accessible to the contributors so that they can understand how the automation works and participate in maintaining the tool-chain. For example, the OpenStack community set high standards and decided to build a framework to continuously test the code base. This starts right at the step when someone proposes a patch. The project, called Zuul, is now a stand-alone framework used by numerous communities and organizations. It also acts as a gating entity. While humans review code and documentation and have the right to approve them, changes get merged only if all the tests are successfully executed.

Reviewing code and documentation changes is a very important responsibility that you need to encourage the entire community to take on. With an open development process, you get experts from all around the globe who have access to proposed code changes and can share their knowledge and expertise through their review and other contributions. As people get a better understanding of how automation works, they also need to utilize it to the highest degree. Adding tests to a code change has to be mandatory for everyone as a first step toward quality assurance. The community can also choose to adapt certain standards that every change needs to follow. It must make sure the standards are well documented and that the guidelines are accessible to new as well as established contributors.

OPEN SOURCE


Finally, we examine the most fundamental principle, which is also the most well-known one: open source. In the context of the Four Opens, it means that software has to be released under an Open Source Initiative-approved

license. This ensures that anyone is able to access, use, modify, and redis-

tribute software, either an original or a modified version. To expand on this, the Four Opens also disable the “open core” model, where a controlling organization withholds and prevents certain features from being added to the open version of a project so that it can offer an “enterprise” version for money. Unlike the previous items, this one focuses on code and other artifacts as well as their availability and accessibility and ensures that licenses and legal implications don’t limit collaborative efforts to enhance and maintain them.

While the Four Opens don’t guarantee the success and longevity of a community and the artifacts it is working on, it provides a strong foundation to build a sustainable and balanced ecosystem around open source projects. We often put technology in the spotlight and forget to calculate the human factor and importance of creating environments where individuals from any background and all around the world can participate. We all have daily lives, challenges, cultures to embrace and sometimes to overcome, battles with a language we do or do not speak, career aspirations, and politics we want to leave behind but still must navigate, and this makes it tricky to handle the dynamics of the open and therefore very public environments that open source communities are. The Four Opens were defined to recognize and address these challenges by providing

There is no room for reserved seats in governance bodies and for favoring one person’s contributions over those of others.

contributing to a cause they believe in and are excited about. 

REFERENCES

1. Open Infrastructure Foundation, 2020. [Online]. Available: <https://openinfra.dev>
2. Four Opens Book, 2021. [Online]. Available: <https://opendev.org/openinfra/four-opens>
3. Y. Zhang, H. He, and M. Zhou, “Commercial participation in OpenStack: Two sides of a coin,” *Computer*, vol. 55, no. 2, pp. 78–84, 2022. [Online]. Available: <https://dirkriehle.com/2022/02/21/commercial-participation-in-openstack-zhang-et-al-ieee-computer-column/>, doi: 10.1109/MC.2021.3133052.
4. F. Hannig and J. Teich, “Open source hardware,” *Computer*, vol. 54, no. 10, pp. 111–115, 2021. [Online]. Available: <https://dirkriehle.com/2021/10/04/open-source-hardware-hannig-teich-ieee-computer-column/>, doi: 10.1109/MC.2021.3099046.
5. D. Riehle, “Single-vendor open source firms,” *Computer*, vol. 53, no. 4, pp. 68–72, 2020. [Online]. Available: <https://dirkriehle.com/2020/04/13/single-vendor-open-source-firms-dirk-riehle-ieee-computer-column/>, doi: 10.1109/MC.2020.2969672.

ILDIKÓ VANCSA is a senior manager of communities and ecosystems at the Open Infrastructure Foundation, Seattle, 98121, USA. Contact her at ildiko@openinfra.dev.