

Developing a strategy for your open source project

Creating a strategy for any software project is difficult. Developing a strategy for your open source software project is no different.

“The hardest part about launching a free software project is transforming a private vision into a public one,” writes Karl Fogel in *Producing Open Source Software*. “You or your organization may know perfectly well what you want, but expressing that goal comprehensibly to the world is a fair amount of work. It is essential, however, that you take the time to do it.”

That is because successful open source software projects require more than accessible code and a willingness to collaborate. The most popular and effective projects set clear goals and thoroughly understand both their communities and their target markets. This guide offers a series of 7 questions you and your team should ask as you undertake the essential work of developing an effective strategy for your open source project.

1. What is the project?

This is the fundamental question. How does the project help its users? What problem does it solve?

Before developing anything further, you must be able to describe your project to a potential user in the most succinct way possible, without specialized language or jargon. Too often, open source projects describe themselves with terminology unfamiliar to many potential users.

They also tend to focus on what a project does, rather than on the problems a project solves. For example, consider the open source project Istio. Asked “What is Istio?” one might be tempted to respond:

“Istio is a service mesh.”

But is that the simplest, clearest, and least confusing way to describe the Istio project? Do everyday users understand what a service mesh is? Do most developers? The shorthand doesn’t help anyone who can’t understand it.

So consider the audience for your project explanation. Talking to someone with a background in networking, for example, you might describe the project as:

“Istio is a framework that allows you to manage each application’s data plane from a central control center.”

Alternatively, for a system administrator or operations team member, a description that focuses on application observability will provide a more useful frame of reference:

“Istio is a central switchboard that lets you control the observability of network traffic between microservices across your enterprise.”



facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

And speaking to someone completely unfamiliar with the concept of a service mesh, we might instead choose more metaphorical language:

“Istio is like a traffic system for your application. A proxy used by Istio sits in front of each application node and directs traffic coming into and out of the node. Traffic rules are set in a central control center for the application.”

Ask as many probing follow-up questions as you must. In the end, just be sure that you understand the project, know what it can (and can't) do, and can explain it to someone else easily and clearly.

2. Who are the project's users?

Successful open source projects require users. But many open source projects do not clearly understand who uses the software they produce, nor do they typically have a sense of what those users do with the software.

Consider developing a set of personas to characterize archetypes of key users engaging with your project. Ask yourself:

- What job must be done, and how does the software help someone perform that job?
- Who (what type of user, occupying what type of role in an organization) is most likely to need to perform that job, and to use this software to do it?

Your answer will likely be something like IT professionals, mobile application developers, or web developers. Next, refine your target user persona by asking follow-up questions designed to hone in on a number of criteria, including:

- Is the project more useful to an individual or an organization?
- Is your project particularly useful in a specific industry vertical or business domain?
- What size organization will find your project most useful? Are you targeting systems administrators in large enterprises, or small- and medium-sized businesses?
- What are the job titles of the people who will be downloading, installing, and using your project? Are they the same people, or are the users different from the project administrators?
- What is the relationship between the people who download and install open source projects and the people who evaluate and purchase commercial products?

Your answers to these questions will impact your priorities for structuring the project, promoting it, and even engineering it (as your answers will prompt you to prioritize developing certain features over others). For example, if your project runs in a datacenter or on a cluster of servers, your audience will typically be a business-focused audience, people running IT professionally (or as a volunteer in a university). For a mobile application or a web development framework, the majority of your audience will be running your project on personal computers, workstations, or mobile phones. Each of these groups is solving different problems and has distinct motivations—and those factors impact your engineering, marketing, and community-building decisions.

3. How do you engage with your user base today?

When people begin adopting your project, you will want them to not only use it but also engage with the project community in order to improve it. Engagement is therefore key to growing both your user base and your project's community. But you can engage with your users in many ways, each requiring different amounts of effort and producing different outcomes. Determining where you should focus your community-building efforts can be difficult.

Take stock of the various ways you are currently engaging with project users. If possible, categorize these as low touch, medium touch, and high touch strategies. Low-touch methods involve little interaction between potential users and your community; that is, they allow potential users to engage with your project in a self-directed way. High touch methods include one-to-one or one-to-few efforts; they require much more direct, sustained interaction between existing community members and potential new ones.

Here are some examples of activities you can categorize this way:

- **Low touch:** Project website and documentation, pre-recorded or automated online training, newsletters, podcasts, and blogs
- **Medium touch:** Project mailing list, bug tracker, community forum, webinars, user groups, and conference presentations
- **High touch:** Phone calls, personalized and guided training, conversations at conferences, and community meetings

Ideally, your project will have a healthy mix of each of these. By listing them this way, you are able to identify gaps in your end-to-end community engagement strategy (one that allows someone initially unfamiliar with your project to start using it, and then, over time, gain seniority in the project to the point of becoming a core contributor). Low-touch activities are best for raising awareness of your project and encouraging users to look at it for the first time. Medium-touch activities can produce a network effect, not only by facilitating communication with users but also by enabling those users to help one another. And high-touch activities are especially effective for building relationships with key community users, gathering community case studies, and turning particularly active users into advocates for your project.

Additionally, listing your engagement activities this way helps you effectively allocate community-building resources. Low-touch activities are excellent tools for assisting new users without intervention from senior community members but are generally impersonal, and high-touch activities can be valuable on an individual basis but do not scale well.

4. What alternatives to your project already exist?

Any open source project strategy must account for a project's competition. Next, you will need to assess the field of alternative projects in which your own project participates. Considerations here include:

- **Does your project have any direct competition?** If not, why is that? Are you breaking ground in an area of emerging technology? Or are people already using other projects to do the same kinds of work or solve the same sorts of problems, albeit in a different way?

- **How do other open source projects solve the problem differently than yours does?** If your competitive field features a strong incumbent, what can you learn by analyzing that project, its community, and its target users? How will you differentiate your work from theirs, and what will motivate people to adopt your project instead of others? If your field of competitors features no clear market leader, then is joining another project—rather than competing with it—a viable option? If not, why? Answering these questions will ultimately help you prioritize features and determine the best ways to engage potential users (for instance, perhaps you can join existing in-person or virtual gatherings related to your competitor’s technology to spread your message).

Answering these questions will also help you and your community hone your project’s messaging. For example, if your survey of the competitive field reveals that you are an upstart disruptor in a field dominated by an incumbent, then you will most likely anchor your messaging to your competition (stressing how your project is less expensive than, or an open source alternative to, or simpler and faster than another option competing for users’ attention). If you are participating in a new market and your goal for the project is to achieve market dominance, you will need to focus on spreading your message quickly—which means a higher marketing budget or more aggressive community plan, and more focus on defining the problems you solve for potential users.

5. Are you already associated with adjacent projects?

If your software is frequently used with (or is particularly useful to users of) another project, then your association with those projects presents a great opportunity for growing awareness of your project in its early stages of development. Accounting for these complementary, adjacent projects will help you define your own project’s strategy.

For example, Ceph can be used to manage storage for OpenStack® or Kubernetes. So the OpenStack and Kubernetes communities are adjacent to Ceph’s community. Catering to adjacent projects to find an audience may affect your technology roadmap, the events you target with your community outreach efforts, and your investments in specific integration features.

An adjacent project provides you with a potentially friendly audience working to solve the same problem your own project does. You may be able to engage in market research, UX testing, and community-building activities with this group in order to raise awareness of your project and meet potential new users. Adjacent communities important to your competitors will be important to you, too.

6. What are your goals for the project?

This is every open source project’s existential question—and surprisingly, many projects have difficulty answering it. Next, then, you need to consider what you want to achieve by investing in your project. Why, in other words, does your project exist?

This is especially critical for projects released by or driven by a vendor. If this is the case for your project, ask yourself why you are choosing to open source this particular software. Your answer might be something like:

- To grow a market.
- To promote a standard.
- To disrupt a competitor.
- To increase demand for another product in our portfolio.

Each of these requires a different set of investments—and a different marketing strategy.

Understanding your reasons for open sourcing the project will help you clarify the investment required to achieve your goal. In the absence of a strong common vision for the project's goals, you may find yourself underfunding the open source project, in part because it is perceived as competition to the products you build on top of it.

Asking this important question also will help you align engineering, product, and sales teams as you develop an open source product strategy (see [Creating an open source product strategy](#) for more on that).

7. Who are your key stakeholders?

In every project, only a small group of people will be deeply invested in the success of the project and can represent the diverse set of interests that are important to its development. These people are your stakeholders.

In the case of vendor-sponsored projects, this group will typically include an engineering lead, someone from product management, a member of product marketing, and a representative from the field (a field engineer, a sales lead, etc.). You may also want to ensure the group includes someone from your content services or support organizations, as well as someone from product security. You will brief this group on your initial project strategy and reconvene the group once or twice every year to report on the state of the project and ensure continued alignment between project goals and the investments necessary for achieving those goals.

Getting started

Answering these 7 questions should enable you to prepare a single-page document to guide your open source project strategy. You will understand the problems your project solves, and for whom. You will be able to explain your project in language that makes most sense to potential users. You will understand how your project fits into the market and what you want to achieve with it. And you will have identified the organizational stakeholders who should always be abreast of and aligned with your strategy.

See the attached worksheet for help getting started.

Worksheet

Define your open source project strategy by completing the items below.

➤ Explain your project

1. Describe your project for a general audience in the simplest terms possible. Focus on the problem your project solves. Avoid industry-specific jargon.

➤ Analyze your target users

2. Describe your project's target user. Explain why this user would encounter (and need to address) the problem your software solves (from Question 1).

3. List your current strategies for engaging with the target user(s) you have identified (in Question 2). Be sure to differentiate low-touch, medium-touch, and high-touch strategies.

➤ Survey the competitive landscape

4. What other projects might your target users (from Question 2) adopt to solve the problem your project addresses? How is your project different from these?

5. What projects might people often use in conjunction with yours?

➤ Set your goals

6. Why are you choosing to open source this software? What do you hope to achieve through your investment in it?

➤ Identify your stakeholders

7. Who in your organization should be aware of your open source project strategy? Who will allocate the resources necessary for this strategy to succeed?

About Red Hat



Red Hat is the world's leading provider of enterprise open source software solutions, using a community-powered approach to deliver reliable and high-performing Linux, hybrid cloud, container, and Kubernetes technologies. Red Hat helps customers integrate new and existing IT applications, develop cloud-native applications, standardize on our industry-leading operating system, and automate, secure, and manage complex environments. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500. As a strategic partner to cloud providers, system integrators, application vendors, customers, and open source communities, Red Hat can help organizations prepare for the digital future.



facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

North America
1 888 REDHAT1
www.redhat.com

**Europe, Middle East,
and Africa**
00800 7334 2835
europe@redhat.com

Asia Pacific
+65 6490 4200
apac@redhat.com

Latin America
+54 11 4329 7300
info-latam@redhat.com

redhat.com
#F23343_0420