O'REILLY®

Compliments of indeed

Investing in Open Source: The FOSS Contributor Fund

A Blueprint for Funding Open Source Critical to Your Business

Duane O'Brien & Mandy Grover

REPORT

indeed

Open source helps people get jobs

Open source is at the core of everything we do here at Indeed. Our collaboration with the open source community enables us to help people get jobs—and better jobs lead to better lives.

Have a passion for open source and making the world of work a better place? Join our team! We have fully remote roles open to engineers and developers anywhere in the US.

Check out our open roles at **indeed.jobs!**

Investing in Open Source: The FOSS Contributor Fund

A Blueprint for Funding Open Source Critical to Your Business

Duane O'Brien and Mandy Grover

Beijing · Boston · Farnham · Sebastopol · Tokyo

O'REILLY®

Investing in Open Source: The FOSS Contributor Fund
by Duane O'Brien and Mandy Grover

Copyright © 2022 O'Reilly Media. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: Rachel Roumeliotis
Development Editor: Sarah Grey
Production Editor: Christopher Faucher
Copyeditor: nSight, Inc.
Proofreader: nSight, Inc.
Interior Designer: David Futato
Cover Designer: Randy Comer

October 2021: First Edition

Revision History for the First Edition
2021-10-22: First Release

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Investing in Open Source: The FOSS Contributor Fund*, the cover image, and related trade dress are trademarks of O'Reilly Media, Inc.

The views expressed in this work are those of the authors and do not represent the publisher's views. While the publisher and the authors have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the authors disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

This work is part of a collaboration between O'Reilly and Indeed. See our statement of editorial independence.

978-1-098-11188-5

[LSI]

Table of Contents

Introduction

Open source is many things. It's the definition of a license. It's a way of solving shared problems. It's a community. It's a project. It's an experiment. It's an ecosystem. But there is one thing that sits at the heart of open source:

People.

People write and interpret licenses, solve shared problems, run the experiments, and power the ecosystem. Without people, open source doesn't work. Open source is intentionally, and necessarily, participatory.

One way to empower participation is through funding. Developer-to-developer and user-to-creator funding has experienced wonderful growth over the last 10 years, and some open source creators derive significant income through this kind of funding.

However, organization-to-project and company-to-creator funding is complex and needs goals and structure to be successful. We're excited to share how Indeed has created a program to fund projects and developers, and we want to inspire you to develop a Free and Open Source Software (FOSS) Contributor Fund at your own organization.

Getting Involved in Funding Open Source

We started an open source program at Indeed because we wanted to do our part to help open source thrive, which means we want as many people participating as we can get—mindfully, intentionally, and authentically. We see the evidence of this type of participation everywhere in the open source community.

Project maintainers invest significant time and energy in making it easy to participate. They write detailed onboarding guides to give new contributors clear guidance on how to get started. They host hackathons, meetups, and community events to encourage project participation. They give talks, speak on podcasts, run livestream coding events, write blog posts, and so much more.

The community has changed the way it recognizes open source participation, looking beyond code contributors to honor organizers, designers, testers, translators, project managers, triagers, and many others. So it seems only natural that we also want more people to participate in deciding how best to fund open source projects.

Yet very few people within organizations make large funding decisions.[1] At most companies, major budgeting decisions are made well above the level of the typical end user of open source software. A senior leader in an organization decides where to allocate budget, with the best of intentions, based on their experiences and the experiences of their advisors. But it can be difficult for an individual developer or end user to advocate for the needs of a project that might be critical to them. We created the FOSS Contributor Fund to provide a way for individual developers and end users to advocate for the open source projects they use and want to support.

The FOSS Contributor Fund

In 2019, Indeed launched a new program to grow participation in funding decisions as much as we could possibly get. The FOSS Contributor Fund enables Indeed employees to help us decide which open source projects we support monetarily. Every month, anyone who participated in an open source project is eligible to cast a vote for one of the open source dependencies Indeed relies on. The winning project receives a $10,000 no-strings-attached donation to use as they know best.

We've learned a lot from running our FOSS Fund. We gained a deeper understanding about the open source that we use, how

[1] We're notably excluding dev-to-dev and user-to-creator funding from this conversation. Some open source creators derive significant income through this kind of funding, but that is outside the scope of this report. Our goal is to widen participation in organization-to-project and organization-to-developer funding decisions.

developers feel about open source, and how to increase the number of participants. We've found an effective way to distribute the decision-making process for funding open source projects—one that connects organizational funding decisions with end users and developers. We've built a funding process that's transparent and invites participation. We've unlocked a new way to connect with open source advocates and identify software that is vital to the business but might otherwise go unnoticed by senior leaders.

Again, we want to inspire you to start your own FOSS Fund. This report will equip you with everything you need to propose, build, and sustain your own FOSS Fund. We will tell the story of Indeed's three-year endeavor to build and implement a funding program that supports the open source community. And we will provide a detailed implementation framework for you to run your FOSS Fund effectively. We hope this work helps you invite open source participation at all levels of your own organization.

The FOSS Fund Blueprint

The FOSS Contributor Fund at Indeed is a living blueprint designed to evolve with iteration. We created this program to increase meaningful participation in open source and fund the open source community in sustainable, democratized ways.

We'll introduce you to the components of the blueprint: *project eligibility*, *voter eligibility*, *funding amounts*, *voting framework*, and *iteration*. We'll also provide insights from our own experiences to demonstrate how our goals guide our implementation of the FOSS Fund and why we believe such a dynamic approach is necessary. As you read about each of the blueprint components, we encourage you to consider how your own organizational goals would modify or work within this model.

Open source is people. Now, let's look at how we can involve more of them in making funding decisions within your organization.

CHAPTER 1

Project Eligibility

The first component is *project eligibility*, that is, defining which projects are eligible for your FOSS Fund program. Your organization most likely relies on hundreds or even thousands of pieces of free and open source software. As much as we would like to define all projects as eligible, we have found pitfalls with that idea. The following criteria, considerations, and lessons learned will help you reduce confusion about which projects are eligible for funding.

Project Usage

The broadest possible criterion is "any project the organization uses." If you needed tighter constraints, you could limit eligibility to projects that you use in a business unit or in a product.

Project License List

Which list of free and open source licenses are you working from? Rather than make your own list of licenses, there are several existing lists that are actively curated and well maintained. Two examples are the OSI Approved License List and the curated list of licenses, which follow the Debian Free Software Guidelines. Don't write one yourself, though! There are hundreds of open source and free software licenses, each with its own nuances. Curating a list yourself is no small task, and well-maintained lists are already available.

Lesson Learned: Don’t Keep a Rigid Project List

Your list of eligible projects you use to determine nominations and eventual funding should align with your goals. If this list isn’t flexible, you can easily become beholden to process over principle.

For example, someone might nominate a project that fits into the goals of your FOSS Fund but doesn’t appear on your canonical list. This could include an important open data project, a project focused on content that is licensed Creative Commons, or a project with an unusual license. If so, find a way to say yes. You may need to create a transparent process for how you’ll handle these types of nominations. Remember that you still have a voting process to go through, which will help you make good decisions.

Project Funding

At a minimum, require that projects have a publicly posted method of receiving financial support. You may decide to be constrained to projects run by nonprofit foundations. Any significant procurement restrictions will necessitate tighter controls.

Lesson Learned: Don’t Fund Projects That Haven’t Requested Funding

Not all projects want to raise or accept money. The maintainers may be well funded already or decide not to accept funds on principle. The project might not be large enough for fundraising or lack a governance structure mature enough to handle the funding responsibly. Whatever the reason, if the project doesn’t want to receive funds, you need to filter it out before the vote.

You may be tempted to reach out and work with the project to set up a way to receive funds. This approach can be problematic. If you reach out to the project maintainers before the vote, you may be asking them for work or attention that doesn’t yield any positive result for them. Further, you can have an unexpected effect if they haven’t yet worked through the logistics of how to receive and distribute funds: adding money to an open source project will change things, and those changes should be intentional. It’s not always easy to decide how to spend funds or which contributors should be paid, and you can unintentionally lead projects into muddy waters by showing up with the best of intentions and offering money.

Lesson Learned: Don’t Fund Projects That Have Strong Corporate Backing

If a project receives a substantial amount of funding from corporations, you may want to remove it from consideration. Some large open source projects are funded by groups of companies or trade associations; others might be wholly owned by one company. These projects have strong name recognition, multimillion-dollar budgets, and significant support.

While there isn’t anything wrong with directing donations to these projects, sometimes the only way to give such projects money is to pay for consulting or services. Doing so may be a good idea, but it will require more time, effort, and paperwork—and some companies don’t allow it at all. You can make a more significant impact with your one-time donation by finding projects that are underfunded or lack strong corporate backers.

Project Ownership

Employee-owned projects are good candidates for exclusion. You may also want to set some constraints around single-vendor open source projects. You should think about how you feel about projects owned by large trade organizations.

Lesson Learned: Avoid Employee-Owned Projects

Consider legal constraints at the company level to fund projects owned by an employee. These constraints are inherently difficult to navigate and add complexity to the FOSS Fund. There can be bias in voting with an employee-owned project, because the voters may know or recognize the employee. You will also want to consider the optics: if you’re running a fund designed to financially contribute to open source, and it appears to be returning that money to the company by funding an employee, how will that look? For these reasons, we encourage you to remove employee-owned projects from eligibility.

Recent Awards

Projects with strong name recognition are more likely to receive more votes, so it's a good idea to limit the number of times a given project can receive an award or how often a project is eligible. You may want to exclude any project that receives funding from elsewhere in the organization or apply a "cool-down" period before the project can receive a subsequent award.

Project Reputation

If you work in an organization that is strongly driven by values, you could find yourself in an awkward position if a project is nominated that is inconsistent with the values of your organization. This could show up in the project's culture ("historically toxic"), leadership ("racist maintainer"), or reputation ("popular with trolls"). You can head off this situation by making it clear that projects must be consistent with your organization's values to be eligible. If you take this approach, you may need to give examples of projects that would not be eligible or actions that would disqualify a project.

Indeed: Finding Flexibility in Project Criteria

Tight eligibility constraints do come at a cost. The more a participant has to read and understand before they can nominate a project, the less likely they are to participate. If they participate without understanding eligibility, they might ignore the constraints and nominate any project, which will complicate the processes in running your FOSS Fund. Keeping your eligibility criteria short will make the process easier for participants and easier to run.

To that end, we use project eligibility criteria that align with our funding principles and allow for flexibility. Initially, we started by accepting any project in use at Indeed—regardless of the context—as long as it was relevant to work. Then, we used the OSI-Approved License List as our canonical license list. We required that projects have some existing way to receive funds, such as an Open Collective or GitHub Sponsors or a PayPal link. Finally, we excluded all employee-owned projects as well as any project that had received a FOSS Fund award in the last 12 months.

Discovering Projects at Indeed

The primary mechanism for discovering eligible projects is the nomination process. For example, curl is one of the most widely used pieces of free and open source software in the world. The project was nominated for a FOSS Fund. As we looked at the nomination and checked the eligibility criteria, we were surprised to see that curl's license was unique—almost but not precisely an MIT license. Because the license was not quite MIT, the license was not technically on the OSI-Approved License List. Though this should have meant that curl was disqualified, we determined that curl was eligible because the broad consensus viewed curl as free and open source software.

We also noticed that many organizations rely on OpenStreetMap for location-based services. It is licensed under an open data license, which would have made it technically ineligible for our FOSS Fund: the OSI-Approved License List only deals with software licenses, not data licenses. However, it was demonstrably in the spirit of the FOSS Fund, and the OpenStreetMap Foundation is an affiliate of the Open Source Initiative. We determined the project was eligible.

Aside from nominations, we also found other mechanisms to explore eligible projects.

Compliance tools

Most organizations have a tool in place for ensuring software security and license compliance. These tools typically generate inventory reports or a Software Bill of Materials that contain links back to the source for every free and open source library that you use. We looked at these reports as a starting place.

Dependency analysis tools

We also used online tools to take a project manifest (such as a *package.lock-json* or *pom.xml*) and analyze its dependencies, which gave us projects to add to our project funding list. Tools like Back Your Stack will show you which projects have a way to receive funds so you can ensure you're not funding a project that doesn't want funding.

GitHub sponsors

Like many organizations, Indeed has repositories on GitHub, which means the GitHub Sponsors program will automatically scan dependencies and display which projects are part of the GitHub Sponsors program. This does not give you a complete view, especially if your organization uses a different source control system internally, but GitHub-sponsored projects can easily go on your project list if they meet your criteria.

As you discover projects and assess them against your established criteria, allow for flexibility and growth. No one list will perfectly capture your funding goals, but staying close to those goals means your choices will be more meaningful, and that encourages more meaningful participation.

CHAPTER 2

Voter Eligibility

The second blueprint component is *voter eligibility*. At this point, you have some nominations for projects. Now, you need to decide who is eligible to vote on the nominations. The pool of voters that you identify will have a direct impact on the goals of your FOSS Fund. The ability to cast a vote will be an incentive for people. This incentive will encourage behaviors that have effects beyond your immediate organization. Since you are encouraging these behaviors, it's important that you are clear about who is in your pool and why you selected them.

Voter Eligibility Goals and Behaviors

Your FOSS Fund can do more than help you identify supportable projects and open source enthusiasts. You can use the voter eligibility criteria as a way to encourage open source participation and an open source mindset within your organization.

If your organization is small, or if you are trying to encourage broad participation and discussion, you could decide to allow anyone in your organization to vote. In a larger organization, you might decide to limit voter eligibility to only those people who have made contributions to open source communities or to specific projects. If you are trying to build engagement between your office and another business unit, you might limit voter eligibility to specific people or business units.

Regardless of your larger goal, you need to consider two important outcomes when setting voter eligibility criteria: the effects *within* your organization and the effects *beyond* your organization.

Your Criteria Will Have Effects Within Your Organization

If someone can become eligible to vote by performing some action, then you are encouraging that action. Be clear about what you are encouraging and do so intentionally.

If your eligibility criteria unintentionally encourage low-quality contributions to open source projects, you're validating this behavior from your company within the community. You will set the tone for your own contributors that low-quality contributions are to be encouraged, when you likely meant to generate more impactful contributions. In the end, the receiving projects will likely get very little for their time, and you're working counter to any goals you may have about meaningful participation.

You can help avoid this bad outcome by proactively communicating the importance of making more valuable contributions. For a contribution to be valuable to an open source project, it must ultimately add more to the project than the attention it draws from the maintainers.

Hacktoberfest Example

For Hacktoberfest 2019, the open source community saw a large influx of readability edits to README files that consisted of removing a single space or correcting "Github" to "GitHub." These edits did not meet the criteria we set for meaningful participation in open source. Those types of contributions might be technically valid, but they don't actually help the project. Instead, they create more burden on the maintainer than the value of the pull request.

The Behaviors You Encourage Will Have Effects Beyond Your Organization

This is especially true if you are encouraging people to participate in open source communities. Consider the effects that will ripple beyond your organization. Look for opportunities to communicate the intent of your program externally. Take advantage of the unexpected ways people will participate and the unexpected outcomes of your criteria.

Sentry Example

Sentry was an open source project wholly owned by a company of the same name, which provides value-added services to the project. While Sentry was technically eligible and had an OSI-Approved license when it was nominated (it later changed to a non-open source license), the project wasn't in the spirit of the FOSS Fund we were running. It was a well-funded startup. We decided to let the voters determine the outcome, and they voted for Sentry. While this wasn't the outcome those of us running the FOSS Fund at Indeed imagined, we reached out to Sentry to ask how they wanted to proceed. They matched our donation and passed all the funds down to free and open source projects that *they* depended on. We later adjusted our nomination curation process.

Kubernetes Example

Kubernetes is a huge project in the corporate open source space. Because it's already so well funded, we didn't expect the voters to select it for an award. One month, however, they did. Some voters argued that Kubernetes wasn't in the spirit of the FOSS Fund and that we should be funding smaller, more independent projects rather than projects that were already well funded. This sparked a great conversation. In the end, the Cloud Native Computing Foundation (the fiscal host of Kubernetes) received our donation and applied it to their scholarship fund.

If you know the goals and behaviors you want to achieve, you can then establish more effective voter eligibility criteria. Consider the following goals and behaviors you may want to encourage.

Encouraging Open Source Participation

If you want to encourage regular open source participation, you can decide that anyone who participates gets a vote. You can make this as broad or as narrow as you like: any kind of participation in any project, or only specific contributions to specific projects.

Lessons Learned: Don’t Limit Who Can Nominate

In addition to voter eligibility criteria, for our first two FOSS Fund rounds, we tried using different eligibility criteria for *nominators* of a project. Specifically, to nominate a project, participants had to have made an open source contribution at least once. By contrast, to vote, they had to have contributed in the preceding month.

In practice, we found this to be a needless complication, and it didn’t actually encourage open source participation. People who were not eligible to vote were eligible to nominate, which was confusing. And we had to audit nominations more deeply, which took additional time. Specifically, we had to make sure that the person who nominated the project had also made some contribution previously. It was a manual verification step that ultimately didn’t add anything but toil. We simplified it by saying, “anyone can nominate.”

Building Organizational Expertise

If you want your organization to get better at using or understanding a specific technology, you can decide that anyone who uses or contributes to the technology gets a vote. This could take many forms, such as developing proof-of-concept applications, driving adoption, evangelism, or direct contribution. You want to ensure that the project’s maintainers are prepared for a fresh influx of activity and that they understand and agree with your motivations.

Building Program Engagement

If you are attempting to build or initiate relationships across program boundaries, you could limit participation to specific business units or subgroups. This approach would need to be paired with some outreach, especially when trying to form new relationships.

Showing Up Authentically

Occasionally, you will be faced with situations where something “feels off”: perhaps someone nominates a project that’s technically eligible but isn’t what you had in mind, or a participant tries to affect the outcome more than you expected.

When you are faced with these situations, here are some guiding principles that may help.

Respect the Project

Each project has a unique set of needs that should be at the center of any decision that you make. Respect the maintainers' wishes if they say "no thank you" to the donation, or if they tell you that the participation you are encouraging is having negative side effects, that the procurement process is a headache, or that they need a different form of support.

Trust the Voters

When in doubt, trust the voters. Not sure if a project should be eligible? Document your concerns and put it to a vote. Worried that a project might be too small or that its usage is too limited? Ask the voters what they think. One of the intentions behind running a FOSS Fund is to involve more people in the process. Embrace this and trust the voters.

Do Better Next Time

If the outcome of a vote is less than ideal, you get another chance to do things better in the next voting cycle. The most likely "bad" outcome is that a free or open source project gets some support—just maybe not one you'd choose again. Commit to improving for the next voting cycle and put your energy into refining your next iteration.

Indeed: A Broad View of Voter Eligibility

The voter eligibility criteria we established at Indeed is "anyone who made any kind of open source contribution in the previous month." While this sounds simple in principle, in practice it's more complicated. We had to decide what "counts" as contribution and how to identify eligible voters.

For our purposes, we took a broad view of "open source contribution." We included any kind of contribution of code or documentation, as well as opening or triaging issues, performing code reviews, organizing on behalf of the project, testing, and more. Essentially, if someone said, "I made a contribution," we took them at their word and counted the contribution. For contributions that were difficult to identify using any automated process, we provided a self-identification form that anyone could fill out.

To help identify eligible voters, we wrote an internal tool called Goby that takes in a list of employee-provided GitHub IDs and asks GitHub if there was any activity in the previous month that indicated some kind of open source contribution—pull requests submitted; issues opened; and comments on issues, pull requests, and code reviews. In general, we didn't try to limit the number of voters, and we erred on the side of inclusion. We trusted the voting process to handle a few extra voters.

Remember to follow reproducible processes to reduce errors in implementation, such as forgetting to verify some nominations or not verifying voter eligibility consistently.

CHAPTER 3

Funding Amounts

The third blueprint component is *funding amounts*. Your total budget is, naturally, one of the most significant constraints. You will have a fixed amount of money that you can spend. How you divide this money will determine how frequently you hold voting cycles and how many awards you issue. Let's take a closer look at how this works.

How Much Do You Have?

When we first launched our FOSS Fund, we were fortunate to get strong executive buy-in on awarding $10,000 every month for a year. We hope you have the same support when you launch your fund, but you may find yourself working with a smaller budget in the beginning. Here are some ways to work with budgets of varying sizes.

$100,000 or More

With a budget this size, you can issue substantial awards monthly or quarterly. You might even issue multiple awards every voting cycle.

$20,000–$100,000

At this level, you can issue substantial awards a few times a year. Some FOSS Fund adopters start by issuing a single award on a quarterly basis.

$10,000 or Less

You may only have enough funding for a single substantial award. Don't let that stop you—any amount is great! You can still use this process to engage a broad group of people in deciding which project you want to support. Alternatively, you could start to shift your focus to funding individual open source contributors and maintainers, so that smaller funding amounts can have a more direct impact.

How Often Do You Vote?

Regardless of your budget, you can choose the cadence of your voting cycles. Each option has pros and cons. It's important to choose a cadence that fits your organization's culture and constraints, because you'll need to stick with it for a while: changing the frequency of voting cycles will be confusing for your voters.

Monthly Voting

A good thing about holding votes every month is that you get a lot of opportunities to iterate on your process. You also get constant input, which helps you identify new projects to explore.

A concern with monthly voting is voter fatigue. You may get strong engagement in the process in the beginning, because the FOSS Fund is new and growing in your organization. But over time, this energy can wane; if so, you may want to shift to less frequent voting. Monthly voting can also come with a lot of operational overhead, so the more you can automate this process, the easier it will be to run your fund effectively.

Quarterly Voting

Holding votes once a quarter can head off potential voter fatigue, but it means you have just 4 iterations a year rather than 12: not a lot of opportunities to learn and adjust. This voting cadence can work well if you've been running your FOSS Fund for a while or if you're simply working with a smaller budget. Quarterly voting also requires less operational overhead, which may mean that you decide to automate less of the process.

One-Off Voting

If you're using this process to make a single award, voter fatigue and reproducibility won't be concerns. However, you'll want to limit how much time you spend designing constraints and automating processes. You can get a lot done with a couple of spreadsheets and an email or a simple show of hands. Put your energy into building excitement about the award and engaging with voters. A good outcome for everyone who participates could help you secure funding for future awards.

Indeed: Experimenting with Budget and Voting Cadence

In 2019, we launched the FOSS Fund with a budget of $120,000, which we distributed using a monthly voting cadence. For 2020, we established a FOSS Fund budget of $160,000, which would allow us to continue our monthly awards while enabling quarterly experiments with other open source funders. We used these additional quarterly experiments to participate in initiatives like the 2020 MOSS Speed Dating event and FOSS Responders. For our 2021 FOSS Fund, we kept the FOSS Fund budget at $160,000 and shifted to a quarterly voting cycle, selecting four projects every cycle. We changed our voting cadence to reduce both voter fatigue and operational overhead. We had some bumps when we did this, as we were still determining eligibility based on monthly open source participation, so we had to clarify our voting process after the first round using this process.

Why did we decide on $10,000 per award? First, we wanted the award to be big enough to have a significant impact. It's difficult for a project to do much with a couple hundred dollars: maybe reward a contributor or two, pay a hosting bill, or order some stickers. We wanted the award to be big enough to cover significant expenses or bring the developer team together for a sprint.

Second, we believed that one large award was preferable to multiple smaller awards. If we had broken the FOSS Fund award into smaller amounts, we could have supported a broader range of projects, but each donation would achieve less than a single large donation.

These two design constraints contributed to our decision to keep our awards at $10,000.

Other Ways to Fund

Indeed isn't the only company to run a FOSS Contributor Fund. The following companies have taken a different approach to funding.

Salesforce

Salesforce was the first company besides Indeed to launch a FOSS Fund. It launched in early 2020 with a budget of $40,000, distributed using a quarterly voting cadence.

Microsoft

Microsoft launched its FOSS Fund in mid-2020 to support open source projects it uses and, more recently, those investing in diversity, equity, and inclusion. Initially issued once per quarter, it has since shifted to monthly awards of $10,000, sometimes issuing multiple awards in a single month through GitHub Sponsors.

CHAPTER 4

Voting Framework

The fourth blueprint component is the *voting framework*. Now that you have some eligible projects, some voters, and some funding, you need a framework for the voting process. You need to prepare to make several key decisions that will inevitably arise.

How Do I Further Curate Eligible Nominations?

At a minimum, you need to ensure that nominated projects meet your eligibility criteria, but you will likely need to curate for other reasons as well.

Curating for Volume

If you have a large organization or a large number of nominations, you will likely need to put a subset of the nominations in front of the voters to make the election manageable. Your voters will find it challenging to select from a dozen or more projects—simply because it's difficult to know enough about every choice to make an informed decision. We strongly encourage you to form a curation committee to help you narrow things down. Pull in representatives from a variety of disciplines, backgrounds, and interests in the program.

If you have a small organization or a small number of nominations, you will need to put in some extra effort to curate nominations. We've used a number of techniques to identify eligible projects, such as revisiting previous nominations or runners-up. We've also investigated our own dependencies to see which have an identified funding source.

Curating for Culture

You might receive nominations for projects that meet the eligibility criteria but have a problematic culture, leadership, or reputation. To mitigate problematic nominations, stay close to your organizational values and the spirit of the program you're creating. A curation committee can also help to ensure that alignment guides your choices.

Curating for Legal Considerations

There might also be business reasons for your company to avoid associating itself with a specific project. From restrictions around countries your company can or cannot interact with to restrictions around competitors, work with leadership and your legal department to make sure that you have clear guidance on the highest risk projects.

Regardless of the reason for curating nominations, look for an opportunity to involve others in this process. You can do this formally, through a nomination committee, or informally, holding public discussions on email lists or in chat channels—or both. You can involve all of your voters in this process if you provide information that can help the voters curate the nominations for themselves.

How Do I Get the Word Out About Participating in the FOSS Fund?

If you want to empower your voters to make the best possible decisions, you need to provide information about the nominated projects. The single best source of information is the person who made the nomination. They likely know how it's used, where it's used, and how important it is to their work.

After that, you need to collect information yourself. The program administrators should either be open source subject-matter experts themselves or at least familiar enough with the ecosystem that they can do a little research and Googling to compile this information.

Many voters won't know about individual projects or understand their importance. You can document this in the voting form or in a blog post or by creating quick presentations about each of the projects. If you can provide project information in the context of the vote itself, you ensure that your voters have the information they need.

Some of the information you'll want to provide includes:

How your company uses the project
: At a minimum, provide context about where and how your organization uses the project.

Project funding requests
: If the project has made a recent call for help or support or a specific request for funding, call out this information for your voters. Requests such as these can have a significant impact on the voting process because of the perceived urgency and articulated need from the project itself.

Current project support
: Your voters may want to know how well your eligible projects are currently supported and which organizations support them. Not all projects publish their budgets, so this information can be tricky to gather. For projects that don't have public budgets, look for another way to indicate the level of support for the project, such as how many maintainers or contributors it has.

What Tools Can I Use to Make the Entire Process More Efficient?

If you have more than a handful of voters, you'll benefit from a voting tool. Let's look at some of the options available.

Hosted Polling System

There are a number of hosted services, free and paid, that run polls and elections using a number of different methodologies.

Chat-Hosted Polls

If you use an internal chat platform—like Slack, for example—in your organization, it may have polling functionality built into the platform or available as a plug-in. Because polls hosted in chats tend to be used for low-stakes decisions, they usually offer fewer options than hosted polling systems. However, there are advantages to hosting the poll in a medium where you can hold active discussions.

Hosted Survey Forms

There are many free and paid survey providers available, and your organization may already have a paid account with one of them. You may have to implement your own polling methodology if the platform doesn't provide one. However, if you're building the survey yourself, you can exercise more control over how it contextualizes project information.

In-Person Voting

When safe and possible, in-person voting can be extremely engaging. At a conference or in-person summit, you can set up a booth, explain your FOSS nominations, and allow others to vote. This type of engagement builds external interest in open source funding and allows you to demonstrate your process.

Indeed: Finding a Framework That Works for Us

In the first year of our FOSS Fund, we used a hosted version of the Condorcet Internet Voting System (CIVS). This is a great system for ensuring fair elections that easily addresses most ties, but the hosted version we chose wasn't a good fit for our process. CIVS sent impersonal emails that were sometimes filtered into spam folders. It was also difficult to provide contextual information about the projects.

Starting in our second year, we switched to using private Slack channels with a polling plug-in. We didn't love this, because we generally favor a wholly transparent voting process. But it was difficult to enforce voter eligibility requirements in public channels. We were also using a more rudimentary polling methodology, and one of our early results was an unexpected tie that we had to resolve ourselves.

After using Slack for a few voting cycles, we realized that we were constrained by the available options in the polling plug-ins that were supported by IT. We could only create a limited number of polls per month, and if we ran the vote at the end of the month, polling was no longer available because we had already met the limit. We've since switched over to OpaVote and are experimenting with that platform.

Other Ways to Vote

Even with a wide range of tools already available, you may want to write your own tool that meets your specific needs.

Johns Hopkins University

When Johns Hopkins University launched its FOSS Fund at the beginning of 2021, the organizers decided to build their own polling application. This gave them full control over the look and feel of the polls and allowed them to have common URLs for each vote. Most importantly, it guaranteed that the university had full control over the data that was collected and how it was used.

CHAPTER 5

Iteration

The fifth blueprint component is *iteration*. Every time you hold a FOSS Fund vote, you have a chance to iterate on your process and improve your outcomes. If you're fortunate enough to be able to iterate frequently, this means you get a lot of chances to improve. If you are holding fewer votes, then it's important to make sure that you're learning as much as possible from each iteration. Here are some best practices we recommend.

Frequent Iterations

If you're iterating frequently, you get a lot of chances to make changes. This can be helpful if you see something wrong and want to make a quick change. It can also cause friction in the process by introducing too much churn.

Don't Change Big Things Frequently

If you change big things about the process every voting cycle, like your polling mechanism or your voter eligibility criteria, you'll generate a lot of confusion. Big things should change less frequently than smaller things, like the emails you send about voting or the way you contextualize content about the projects.

Don't Worry About Small Errors

Process errors matter less when you are iterating frequently. If you somehow miss a project that should have been included in the vote or fail to identify some eligible voters, don't worry. Add the project or the voters to your next voting cycle and look forward to the next vote. Frequent iteration allows more room to relax and look forward.

Infrequent Iterations

If you're running infrequent iterations, you have less opportunity to make changes. This can be helpful in stabilizing the process or cause friction because you have fewer chances to fix a problem.

Don't Make Changes Without Clear Intention

With infrequent iterations, the changes you make need to be more intentional. It might be months before you get to try again. However, while it's important to consider your changes and your process more carefully, don't focus *too* much on this, or you may hesitate to experiment at all!

Don't Read Too Much into the Data

Because you have less data coming in about your experiments, your results will be more susceptible to unrelated effects. For example, a large hackathon could significantly change the number of eligible voters between quarters. New engineering efforts might raise the visibility of new dependencies. Make sure to examine your results closely and try to identify the outcomes caused by your experimentation.

As you run your FOSS Fund over time, you're going to see some outcomes that you don't agree with or that you think are "wrong." Embracing and accepting this now will save you some worry later. Remember that the most likely "bad" outcome is that a free or open source project gets some support—just maybe not the one your organization would choose again. Spend your energy learning from the iteration and focus on getting things "more right" next time. Part of showing up authentically (our recommendation in Chapter 2, "Voter Eligibility") is reflected here when we encourage you to listen to the voters, trust the project, and do better next time. Iteration isn't just about the logistics of the program; it's also about the philosophy behind a dynamic blueprint for a FOSS program.

Indeed: Deciding When to Make Changes

There are specific aspects of our FOSS Fund that we decided we could change every iteration, and some aspects we only change once a year.

We changed the polling mechanism (CIVS or Slack polls) once a year. We also keep our voting cadence as consistent as possible. For the first two years, we ran monthly polls that started and ended around the same time every month. In our third year, we switched to a quarterly voting cadence and committed to that cadence for the full year.

However, we are constantly making changes to how we structure discussions or contextual project content. We try to improve this every voting cycle.

If we need to change a requirement because we find a loophole, we try to make these changes as soon as necessary. For example, as discussed earlier, we dropped the idea of "nomination eligibility" in only our second monthly vote. We did this because it was confusing the voters and complicating our implementation.

You now have all five components of the blueprint—*project eligibility*, *voter eligibility*, *funding amounts*, *voting framework*, and *iteration*. We use this blueprint to run the Indeed FOSS Contributor Fund, and we believe you can run your own fund with this blueprint as well.

However, we're not quite done yet. Now you have to implement the blueprint in your own organization. While we've provided some lessons and examples so far, we want to go deeper. In the next chapter, we will provide some specific steps that will enable you to implement your version of this blueprint effectively.

CHAPTER 6
Blueprint Implementation

We've taken you through the blueprint for running a FOSS Contributor Fund. We looked closely at each component of the FOSS Fund and the purpose each one serves. In this chapter, we will focus entirely on how to implement your FOSS Fund. We'll start with the most important implementation step: *securing funding*. Then we'll cover everything that comes next: *communication*, *voter identification*, *voting*, *payment*, and (yet again) *iteration*. We'll provide more information about how we implemented our FOSS Fund at Indeed and dive deeper into some specific places where your implementation framework may vary.

Securing Funding

As the Head of Open Source at Indeed, I answer a lot of questions about our blueprint and our processes, but the question I'm asked most frequently is "Duane, how did you get funding for this?" Whether I'm presenting about the FOSS Fund at a conference or chatting casually with an open source peer from another company, this is *the* question. With so many competing internal priorities, it isn't always easy to secure financing for a project like the FOSS Fund. While this might seem like the most important question to answer, I want to challenge you to answer a different question first: why are you committed to running a FOSS Fund?

Without a commitment to everything that makes the FOSS Fund a program—the blueprint, unexpected challenges, iteration, growth, iteration, and more growth—your FOSS Fund will not succeed. So, again: why are you committed to running a FOSS Fund?

If you don't know how to answer this question, you're not ready to ask for funding. Spend some time building an understanding of what you're trying to accomplish with your FOSS Fund and why it is important to you. "I think it's a great idea" isn't usually enough for executive sponsors. Be clear with yourself on why you want to run a FOSS Fund, and you'll be clearer with your sponsors.

Now back to the central question: "How did you get funding for this?" My answer is always the same: "I was fortunate to be working with executive sponsors who understood the value of giving back to the open source community. In fact, I sought out this job because the role was built by someone who wanted a program for giving back to the open source community. And because I work for someone who gets it, they were willing and ready to work with me toward this shared goal."

On the surface, that isn't a very helpful answer, especially if you feel like you work for someone who will be hard to convince about funding. But there are two sides to this answer. If you don't work for someone who gets it—who doesn't immediately understand the value of giving back to the open source community—then your best option may be to identify a problem that your sponsor cares about more deeply. If you can help them solve a problem they already have, you'll convert them from a skeptic to an ally.

Some questions that may help you:

- How can we help ensure that our dependencies are more secure?
- How can we build connections with the open source community and/or tech community?
- How can we make the best use of unspent surplus funds?
- How can we learn more about what our developers love?

If you already work for someone who understands the value of giving back to the open source community, then you already have a powerful ally in securing funding. You can work together with them to craft a compelling story and sell the idea upward.

Your story should cover at least the following proposal requirements:

- The problem you're solving for your organization
- How the FOSS Fund will solve this problem
- Your proposed budget and time frame
- How you're going to measure the outcome
- Who will own the implementation plan

When you ask for funding, you need to say two things quickly, right up front: how much you're asking for, and when you'd like to have it. These two pieces of information will enable your sponsor to balance your request against competing considerations and requests. If the amount is too large or the time frame is too short, they can say no immediately and spare you both time and effort. If the amount is within budget and fits within a good time frame, they can ask you to explain the rest of the initiative.

Understanding how your larger organization handles budgeting is critical. You will need to do some legwork to find out when budgets are submitted and approved, so you can time your request appropriately. If you submit a request the day before budgets are due or two weeks after they've been submitted, you're unlikely to get funding.

When I submitted my proposal for what would become our FOSS Fund, I created a one-page write-up that led with this paragraph:

> **Subject:** $120,000 for a Sustainability Fund in 2019
>
> **From:** Duane O'Brien
>
> I am asking for a budget of $120,000 allocated in 2019 for distribution to open source projects that Indeed consumes. Each month, Indeed would distribute $10,000 to a project voted on by Indeedians who made an open source contribution that month. This would democratize where some of Indeed's open source dollars go while giving open source participants motivation to self-identify and engage with the program.

I opened with how much ($120,000) and when (next fiscal year). I went back and forth on this number a lot as I tried to decide if I should ask for less ($60,000? $40,000?). In the end, I asked for the full amount that I could imagine for the first year. I was delighted when we received full funding for the program, and I learned a valuable lesson: *ask*. Give your sponsor the opportunity to say *yes*. I sent my sponsor the proposal more than six months before the 2019 budget would be set so we would have ample time to discuss and plan. The rest of the document went into detail about a rough implementation plan, but this structure gave my sponsor the ability to quickly assess the program's feasibility before reading the whole proposal.

Communication

If you live in the United States, your mailbox is probably inundated every election cycle with mailers, postcards, door knockers, phone calls, and letters from political candidates. Voter outreach is an effective way to share crucial information.

While you don't have to go to these kinds of extremes for your FOSS Fund, you should put together a communications strategy to connect with eligible voters. One thing is for sure: if no one knows about your FOSS Fund, you won't get any voters.

As you put together your communications strategy, you will need to make sure you cover some specific communication scenarios.

Project nominations and voter eligibility

You will need to periodically post reminders about project nomination and voter eligibility. At the very least, you should send these reminders a couple days before starting a vote and a couple days after a vote has concluded, and the audience should be as broad as possible. *We have a new FOSS Fund round starting on the first of the month! Make sure you've nominated any projects you'd like to see in the next vote, and get your contributions in so you're eligible to vote!* Make sure you include all important information, such as relevant dates and links. These messages are a good fit for synchronous chat discussions.

Voting results

You should post the results of each voting round in a persistent location—a web page, a blog, a maintained wiki, a Git repo, etc. Having a public historical record of the voting outcomes will help everyone understand the process and their ability to impact the outcome of the votes. When possible, including a statement from the winning project will help your voters connect more closely with the outcome.

Maintaining a landing page for pertinent information
: You need a place to send people who have questions about how it all works. Keeping this landing page up to date with changes is necessary, but you should also send an email that outlines important program changes when they occur.

Iterations and changes
: You will occasionally want to change how you run your FOSS Fund. When you do, be clear about what you are changing and why you are changing it. Add this information to the landing page.

If you are communicating or executing on an erratic schedule, it is harder for your voters to stay in the loop. You will get more consistent participation if you are in a consistent cadence with your voting and communications. Be proactive about keeping people informed. You may need to experiment with message frequency and medium to find a balance that works with your internal culture. Don't overlook opportunities to place messages about the FOSS Fund in existing newsletters or in organizational meetings. This is an effective way to expand your reach.

Voter Identification

We wanted our process for identifying eligible voters to be well defined, transparent, and reproducible. To ensure this was the case, we created and released an open source tool called Starfish to help anyone who wanted to run a FOSS Fund identify voters, using eligibility criteria that were similar to ours.

To use Starfish, you will need three things:

- A list of GitHub IDs that you want to check for eligibility
- A list of GitHub Events that define voter eligibility
- A little experience running utilities on the command line

For input, Starfish expects a file containing a list of GitHub IDs as a list of comma-separated values, such as you might export from a spreadsheet. For each of these GitHub IDs, Starfish asks the GitHub API for a list of events generated by that user and checks each event against a user-configurable list of events that indicate some eligible activity. When an event is found, Starfish prints out the ID and moves on to the next ID in the list. When Starfish has completed running, it will have printed a list of every GitHub ID it thinks is eligible to vote.

This methodology is not perfect. At the time of writing, Starfish does not check to see if the repository corresponding to the event is licensed with an OSI-approved license. This means that some people will be identified as eligible voters when they did not make an open source contribution. Our approach has been to err on the side of counting eligible voters, and trust in the voting process to handle any imbalance.

You can find detailed and up-to-date documentation on running Starfish in the Starfish repository. A complete and current list of the events that GitHub publishes resides in its API Documentation.

However, not all open source lives on GitHub, and not all forms of open source contribution show up as GitHub Events.

How Can I Identify Voters Who Aren't on GitHub?

It is important to have a way to identify people who may be eligible to vote but who might be missed by the Starfish tool.

The following methods will allow you to identify more voters:

Offer a means of self-identification
: If your voters are engaged and know they should be eligible, give them an easy means of self-identifying their contributions. If you find yourself overwhelmed with self-identified contributions, you may want to ensure that your criteria are well set and well understood.

Employ metrics tools with a broader reach
: Your organization may be deeply involved with a large number of projects that don't use GitHub. These projects likely have their own metrics solutions or dashboards, or there may be tools owned by the Community Health Analytics Open Source Software (CHAOSS) project that can help you broaden your reach.

Allow noncontributors to vote
: You can set your eligibility criteria in a way that doesn't require tooling at all. The simplest form of this would be giving everyone in your organization a vote. You could also limit voter

eligibility based on internal organization structures, such as everyone in a specific business unit. For this to be effective, you will need to ensure that you have a well-defined set of boundaries that describe your eligible voters.

When in doubt, err on the side of giving someone a vote and trust the voting process. If they are interested in participating in your FOSS Fund, get them involved!

Voting

When it comes to holding the vote, you don't need to write your own voting software. There are many options already available, many of which are open source or free. We will look at two approaches we tried and highlight some potential problem areas.

Condorcet Internet Voting Service

In the first year of our FOSS Fund, we used a hosted version of the Condorcet Internet Voting System (CIVS). This is a free voting service hosted by Cornell University. It has been a popular choice in the past for open source projects and foundations. CIVS implements a well-proven voting methodology that all but eliminates the potential for a tied outcome.

The methodology behind CIVS is sound, but the hosted version we chose wasn't a good fit for our process. It was difficult to provide voters with contextual information about the nominated projects. Voters received an email about how to vote that did not contain much information about the vote itself. While we always felt confident about the outcomes generated by CIVS, setting up and running the vote was a very manual process. It turned out to be a lot of overhead for us, and we felt the friction was inhibiting overall voter participation.

Slack Polls

In the second year of our FOSS Fund, we used Slack and a simple third-party polling app to run our votes. Every month, we invited all eligible voters to a private Slack channel where we hosted the poll. This allowed us to discuss and to provide better context for the nominated projects prior to voting.

While this helped us reduce friction in the voting process and improved overall voter participation, the third-party polling app we used did not have a robust set of options for running the elections. This resulted in several tie outcomes that we had to resolve. We also didn't have the level of control we wanted over voter participation—using private channels meant giving up some transparency, but using public channels invalidated our voter eligibility process.

Regardless of how you run your votes, you need to take the following considerations into account:

- Can you demonstrate that the vote was run fairly?
- Will voters have the information they need to make an informed decision?
- Do you have a plan for ties?

If you do end up with a tie, don't split the funds. You've made a public commitment to fund projects at a stated level, and splitting the funds will undermine this commitment. Instead, look for other solutions. Can you find more funds and support both projects? If not, you can hold a run-off vote or ask your sponsor to cast a deciding vote.

Payment

Before you hold your first vote, you need to make sure that you understand how payment will be issued in your organization. If you have a procurement department, set up a meeting and make sure you understand your organization's constraints around issuing payments. If the recipient has to navigate a complex procurement process, you may be doing more harm than good.

Here are some questions that will likely arise:

Single-maintainer project, no fiscal host
: Do you know how to pay someone who wants to be paid via PayPal? Check? International wire transfer?

Vendor-owned project
: How do you engage with a vendor-owned project that may involve a legal agreement?

Charitable contributions versus project sponsorship
: Do you know the differences at your company between donating to a charity and funding a project? Procurement departments are likely to treat a donation to a 501c3 in the name of a project very differently than a sponsorship that comes with perks and may have things like logo placement or a contract.

Regulated payments
: Is the payment subject to constraints from regulation? Is your organization prohibited from issuing payments to certain kinds of organizations?

If your procurement policy makes it impossible to issue payments to certain kinds of projects, it is best to know that when the project is nominated rather than after the vote has concluded.

Iteration

Every round of voting for your FOSS Fund is a chance to iterate and improve. This process can lead you to a strong implementation model that works with your organization but only if you are intentional about the way you iterate.

Remember the iteration guidance we gave in Chapter 5, "Iteration": don't change too much at once, communicate your changes, and don't make big changes too often.

You may find it useful to maintain a log of changes to the program in your public documentation. Infrequent participants might miss big changes, and a log will help them to understand why the FOSS Fund might look different now than it did a year ago.

Consistency is crucial to running an effective FOSS Fund. You want your voters to be well informed and engaged, so that together you can make the best possible decisions about how to invest in open source. Do your best to identify any potential pitfalls in your voter eligibility, election, and procurement processes. Then embrace the process of iteration and improving your FOSS Fund. Do this well, and you will learn a lot about your organization and your open source infrastructure.

CHAPTER 7
Beyond the FOSS Fund

The FOSS Contributor Fund is a useful framework for you to invest in the open source software your organization uses every day. Running a FOSS Fund will deepen your connections to projects you depend on, give you insights into the open source culture at your organization, and—most importantly—bring new perspectives to bear on your open source funding decisions. By democratizing this process, you're also bringing people within your company together to make important decisions.

In the three years we've been running our FOSS Fund at Indeed, we've found ourselves asking new questions that have opened up new avenues of exploration and fresh investigations. We plan to cover some of them in a subsequent Investing in Open Source report, but for the moment, here are a few of the questions we're thinking about.

Should We Focus on Funding Projects or Funding People?

A single open source maintainer might do incredibly valuable work—supporting dozens, hundreds, or even thousands of small open source projects. Those projects might not be large or important enough to win a FOSS Fund vote by themselves, but they are valuable, and we think the work of those maintainers deserves recognition and support. That's why, at the beginning of 2021, we launched a new program to identify and fund specific individuals who have an outsized impact on the open source we use every day.

How Can a FOSS Fund Help You Invest in Your Open Source at Scale?

Your organization depends on tens of thousands of individual free and open source projects. You can supplement your learnings from your FOSS Fund by doing a deeper analysis of these projects, either by using existing tools or by creating new ones. We will discuss how we approached this analysis for ourselves using the output of source code analysis tools, new utilities, and some good old-fashioned elbow grease.

How Can Internship and Fellowship Programs Bring New Maintainers into Projects?

Through programs like Outreachy, you can fund interns who will make direct contributions to important open source projects. We will look back over our approach to funding interns through programs like these and share observations based on our participation in these programs.

More Questions

Of course, there are plenty more questions—the dynamics of investing in open source are complex.

To name just a few:

- How do you account for geography or local economy when deciding to support an individual maintainer?
- What effect does money have on project governance?
- How do you know when a project isn't a good candidate for funding?
- How can you understand and improve the long-term health of the projects you are supporting?
- How do you shift from writing checks to writing code?

People like you sit at the heart of open source. Whether you're investing time or money, a great deal or a small amount, you can use the FOSS Fund blueprint to involve more of your colleagues in thoughtful, authentic decisions with tangible, substantive benefits for the whole open source community.

We started our FOSS Fund and wrote about it because we want open source to thrive. We hope you will join us in approaching open source investment mindfully, intentionally, and authentically. We hope our blueprint does enough of the work that you can envision what a FOSS Fund looks like at your organization each step of the way.

About the Authors

Duane O'Brien leads the vision for open source at Indeed. He manages the people, policies, and ideas to grow open source participation within the company. He loves telling the story of open source through collaboration and conversation. Duane is a force of chaotic good using his high stats in intelligence and charisma to advocate for the open source community. If you encounter him in forested areas, he will share his fire, drink, and philosophy.

Mandy Grover helps technical leaders at Indeed communicate effectively. She is a Technical Content Architect and manages a team within the Technical Content organization. She loves bringing orderly writing out of the idea cloud and developing frameworks for producing and consuming content at scale. In her free time, she collects vintage clown art and works as a consulting detective.