EXHIBIT 14

NOVEMBER 8, 2023

The product approach to open source communities

A business wouldn’t take its product development for granted, so why would you neglect the OSS community that’s fundamental to the project’s very existence?

Credit: Alexandra Francis

Open-source communities, just like physical products or software applications, can be viewed and managed with a product-oriented approach. In fact, product management principles and strategies can better serve both the communities that support open-source software (OSS) projects and the businesses that depend on them.

Let’s get one thing clear before we start: OSS communities are made up of humans—generally passionate, devoted humans at that—and should never be thought of as commodities. Rather, *because they aren’t commodities,* I argue that OSS community management deserves the same intentionality your organization shows to product management. A business wouldn’t take its product development for granted, so why would you neglect the OSS community that’s fundamental to the project’s very existence?

The world runs on open source technology, but open source communities notoriously face challenges ranging from resource constraints, governance issues, and a dependence on volunteerism, which itself is dependent upon individual passions and constrained time commitments. Executing open source technology well takes strategy—and a different mindset. To be effective, we need to think of it more like a product—dare I say, business—if we hope to achieve an open-source community’s goals..

My career has spanned both product management and engineering, often in an entrepreneurial setting where everyone is building something from nothing. Throughout it all, open source has been at the center. The products that my colleagues and I created wouldn't exist without the extensive use of open source software. Now at Sauce Labs (whose founders are also the Selenium creators), open source is even more relevant to the work I do every day.

If there’s one thing I’ve learned on my journey, it’s that open-source projects and the companies that depend on them fail to grow if they lack good product management principles and don’t balance business interests with community stakeholders. We don't require additional contributors, features, or bug reports; instead, what we need is alignment in processes and procedures. This means we need strong and streamlined product management practices and the right product market fit in place in order to accommodate the duality of success in both practices.

Viewing a community as a product in the context of software development means treating the community surrounding a product, such as users, contributors, and supporters, as if it were a product itself (again, not a commodity!). This perspective focuses on intentionally developing a community so that it satisfies customer needs.. The journey starts by understanding product-market fit.

What is product market fit?

While specific definitions of ”product market fit” (PMF) vary, the concept is the degree to which a product satisfies the needs and preferences of a target market. Product managers know this well; we spend countless hours reading stories from companies that have reached this milestone and, unfortunately, even more headlines about those that do not. Let’s break this definition down:

The market

In an open-source context, the market is the community of people who use the code and care enough about your project to contribute additional code, ideas, feature requests, testing, translations, bugs, security audits, blog posts, and more. It can be quite tricky to figure out, but available market research should be used at all times.

For example, developer surveys from GitLab, Stack Overflow, and others provide great insight into the different types of developer groups and their needs. Similar to a product, you need to continually re-evaluate the market you operate in to understand if that market is changing or evolving.

The community is incredibly important because these are the people who will contribute to the project in some way (at least a small subset will). We look at community as a way to give back and give forward. What we mean by this is contributions from the user community play an indispensable role in the success of open source projects, allowing them to continuously evolve, remain relevant, and serve a wider audience. It fosters collaboration, innovation, and community-driven development. Serving the open source community and building upon it means that contributions will be user-centric, offer value, implement feedback and iteration, maximize growth strategies, and monetize on long-term sustainability. In many ways, this is a symbiotic journey; an OSS community can make a business' PMF more precise, and the business can make the OSS community stronger because of it.

Sometimes the users of the project are very different from those with the expertise or desire to contribute back, meaning the users of the software may have different backgrounds, skills, or motivations compared to those who actively participate in the development of the software. This is a sticky situation to be in because you might be forced to go down the path of finding "mercenary programmers" to keep the project in motion. This is another area to keep top of mind and re-evaluate continuously; if the user community is not naturally contributing to the project, project leaders need to regularly evaluate the options and strategies for keeping the project alive and progressing. It's important to consider whether paid developers or alternative means of support are necessary to sustain the project over time.

The product

The "product" is essentially the solution you have created to solve a problem for a group of users in your market. My favorite approach (and the approach of the best product companies) to finding the right product is to think about it based on four key risk factors:

1. **Value** - Will customers find enough value in my solution to choose it?
2. **Usable** - How easy is it to use my product?
3. **Viable** - Can I afford to build the product?
4. **Feasible** - Do I possess the skills and technology to build it?

Here's how you can implement the key concepts that successful product companies use in your open-source project; Cal.com is a great example of an open source project that builds with a PMF focused approach. Ask yourself these questions, and you'll discover the answers you need to move forward. Don't worry, you don't have to be a product manager with decades of experience to get this right:

Value

- What is your ideal customer profile and persona?
- Who is the decision maker (it might not be the developer)?
- What problem do you solve?
- Who is the competition? How do you differentiate yourself?

Usable

- How easy is it to onboard? What is the time-to-value for your project?
- What friction points exist in the workflow of your product? Do you know?
- How do you get feedback from users?

Viable

- At a product company, you are only viable if you can afford to pay people to build your product. The equivalent is building a community in OSS.
- How do you plan to build the community? Is what you are building interesting enough to build a community around? Will your personas find it interesting to contribute to?
- Are you planning on building your community from the employees of a company?

This is where a business model becomes essential, and many companies choose between models such as the Red Hat model or open source managed services.

Feasible

- Related to the community, do the skills of your target persona match those who need to contribute to your tool?
- - For example, we've observed instances of GenAI projects that don't align properly, where the tool is *created* by data scientists but *for* developers.

Open-source technology is changing rapidly due to the collaborative nature of these projects, increased financial support, market demand, and the ability to adapt quickly to emerging technologies and changing circumstances. Amid all these changes (or perhaps because of them), open-source project maintenance rates are in steep decline, and the future of open source is at a crossroads.

We need to underscore the idea that successful open-source communities require planning, organization, attention to user needs, and ongoing improvement, similar to the way a product is developed and refined. Open source plays a central role in business, but businesses are taking actions that are reshaping the dynamics of this relationship. Due to various conflicting interests and project licensing intrigue (way too much to get into here), an “Us vs. Them” mentality pervades the discourse around open source.

Successful open source projects and thriving open-core businesses are not mutually exclusive. In fact, many of the most successful OSS projects depend on corporate sponsorship. Out of context, it might sound callous to adopt a product manager's mindset to an open-source project made up of the collective efforts of passionate humans.

But it’s my belief that maintainers who align the interests of both community and business are the ones who build many of the most sustainable, enduring projects. You're not a sellout, you’re a buy-in.

AUTHORS

Mike Donovan ›
SVP of Product at Sauce Labs

open source | coding community | product management | Code for a Living | Business Hub

RECENT ARTICLES

OCTOBER 23, 2024

How to improve the developer experience in today’s ecommerce world

OCTOBER 23, 2024

Enterprise 2024.7: Empower your subject matter experts to contribute

OCTOBER 22, 2024

CEO Update: Building trust in AI is key to a thriving knowledge ecosystem

OCTOBER 17, 2024

Brain Drain: David vs Goliath

LATEST PODCAST

OCTOBER 29, 2024

How can you get your kids into coding? We asked an 8-year-old app builder.

Add to the discussion

Login with your **stackoverflow.com** account to take part in the discussion.

stack**overflow**

Light Dark Auto

Stack Overflow for Teams

Pricing
Customers
Our solution
Integrations
Features
Customer Success
Security
Return on Investment (ROI)

OverflowAI NEW
Now available on Enterprise.

Try free

Use cases

Engineers
Data Scientists
DevOps & SRE
Support
Product Managment

Resources

Productivity
AI/ML
Guides and Insights
Customer Academy

Log in

FAQ

Help center

Stack Overflow Advertising

Why Stack Overflow?

What to expect

Advertise to developers

Attract tech talent

Post a job

Powered by Indeed

Use cases

Marketing Teams

Employer Branding Teams

DevRel Teams

Talent Teams

Technology Teams

Agencies

Resources

Product guides & insights

Community insights

Advertising best practices

Talent best practices

Company

OverflowAPI NEW

Stack Overflow's subscription-based API service to train and fine-tune large language models.

About

Leadership

Social Impact

Press

Careers

Open positions

Contact us

Blog

Newsletter

Podcast

Labs

Annual Developer Survey

Terms Privacy policy Cookie policy Cookie settings Go to stackoverflow.com