

[Back](#) | [SoftwareCEO Home](#)

Tips & Tactics

© Copyright 2002-2005
SoftwareCEO Inc.

June 7, 2005

How to open source your software: Three key success factors you need to watch

by [Gordon Graham](#), Editor, **SoftwareCEO**

To hear some tell it, open source represents salvation for every understaffed, underpowered ISV.

You get killer code developed by a devoted global community of hotshots, you get to slash your sales and marketing budgets, and you get the warm and fuzzy feeling that comes from being one of the "good guys." Right?

Don't count your penguins before they hatch.

"Most CEOs and VCs underestimate the work involved in going to open source," says David Taber of [David Taber & Associates](#). And he should know.

Taber is a software industry veteran who worked on open source products for Sun Microsystems as well as startups in Germany and India. He was quoted in [Business Week](#) saying open source means "you have to give away your baby."

"No two commercial open source projects are the same," he says.

"They don't start under the same conditions, or have the same objectives. For example, do you have a commercial product that's already been downloaded 100,000 times? Or a million lines of code with only 100 customers?"

In other words, doing the right thing depends on where you are and where you're going.

To succeed with open source, you need to make all the right moves in three critical areas: strategy, marketing, and operations.

Here is Taber's considered advice:

Success factor #1: Plan on one key strategy.

"Most CEOs take this for granted," says Taber. "But your strategic rationale requires real thought, since it drives dozens of other decisions and tradeoffs along the way."

The goal of most open source projects is to deal a strategic blow to your competitors.

Some common goals include:

- Killing a competitor's revenue stream
- Increasing the size of your developer and partner base

- Improving the quality and feature set of a product with near-zero revenues
- Increasing goodwill with customers
- Creating a popular baseline on which you can build a value-added business

Taber advises CEOs to prioritize from this list, because by definition you can only have one prime objective; all the others must be traded off.

Beyond this, an open source project must be coherent.

Any contradictions or insincerity will be immediately detected by the community, and criticized mercilessly by the same zealous people you're trying to enlist.

"Unfortunately, many management teams have bipolar behavior around their intellectual property (IP)," says Taber.

"They want to do an open source project, but still maintain the integrity of their patents, and make sure that no competitor can use their IP against them."

That may be possible, he says, but make no mistake about it: "The reality of open source is that you're going to lose control of your IP."

Four steps to an open source strategy

To formulate your open source strategy, Taber recommends the following four steps.

Step 1: Establish your priorities, objectives, and agenda, along with metrics for measuring your success.

Step 2: Create a budget that accounts for both one-time and ongoing costs.

Step 3: Specify the "rules of engagement" for all employees.

Step 4: Publish your written plan to the team, but never let it get outside your company.

For example, when Taber worked on Sun's NetBeans open source project, the primary goal was to increase the size of the developer and partner base for Java.

"Everything else was subordinate. And we got to over 4 million downloads," he says.

"Sun achieved tremendous success because the company had dedicated teams and budgets, a two-year timeline, and metrics for the number of downloads, developer community members, partner products, and contributions.

"And we made the internal ownership very clear before we started on the parts of the project visible from outside the organization."

Success factor #2: Develop a sustained marketing

campaign.

"The whole point of open sourcing is to change market dynamics," says Taber.

But it takes probably two years and lots of ongoing marketing to get there.

"It's a common mistake to assume that your marketing is complete once you do the press release and launch the open source website. In fact, that's just the beginning."

Your open source project needs to generate ongoing awareness, enthusiasm, and adoption. But software developers are allergic to conventional marketing, so you have to be subtle.

He says the best "marketing" is often done by engineers who architect the project, and by savvy techies who act as shepherds for the community.

Here are eight key issues Taber says you must consider as you develop a marketing strategy for your open source project:

Open source marketing issue #1: Analyst strategy

Which ones, if any, do you need on your side? When do you need to brief them about your open source project? What do you need them to say as a reference for the media/industry?

Open source marketing issue #2: Press strategy and messaging

Which reporters do you want writing about your project? What is your message you want the trade journals to echo?

Open source marketing issue #3: Trademarks

Are you inventing a new mark for the project? Are you using your old product trademark in some way? What do the lawyers say about protecting your mark? And what will the open source zealots say about a project named after an existing commercial product?

Open source marketing issue #4: Licensing model

Will you use GNU? Mozilla? There are probably 30 different open source nuances, so you need to be clear on this, and work with lawyers and writers who understand open source.

Open source marketing issue #5: Intellectual property

Do you own all of the IP in the open source project? Did you license-in any technology and embed it in the code? How can you tell for sure?

Open source marketing issue #6: Website design

Can you get the right ".org" domain to match your project name? How and where will the community interact? Do you want two sites, one for users and another for contributors?

Open source marketing issue #7: Partners

What is your offer to product and service providers? What business models will you support? Who will recruit partners? What are the revenue flows, and how will they make money? Is it

OK for "competitors" to also be partners?

Open source marketing issue #8: Rules of the road

What are the ground rules, bylaws, and social contract between your company and community members? How will they contribute? Why will they stay?

Taber advises that after your launch, your marketing people must look for ways to get monthly PR exposure for the community and the project. They will also have to manage the evolution of the project and the community membership on an ongoing basis.

Surprise: This is a full-time job.

Most marketing costs will stay the same or even increase slightly with open source. The only marketing cost that will go down is lead generation, because that happens virally.

For example, Taber worked on [Niku Corporation's](#) OpenWorkbench project to open source a project management tool that competes squarely against Microsoft Project.

In the first six months, it was downloaded 30,000 times.

Taber attributes Niku's success to the fact that the company had a careful plan that adhered to all the principles outlined above.

And CEO Joshua Pickus championed the project throughout the process.

"Niku developed three part-time roles dedicated to ongoing marketing and community management," says Taber. "The bigger the community, the larger the effort required. Niku showed a serious commitment, and it is paying off."

Success factor #3: Prepare your operations carefully.

"This is the area of open source projects most often ignored and underestimated by CEOs and VCs," says Taber.

"Engineering will actually have more to do on an ongoing basis. Support will have some work to prepare for the conversion to open source. And sales will have to be re-trained on some issues.

"All this can come as a big surprise at budget time."

He says most ISVs want to build a business around the new open source base, so you'll still depend on the evolution of the code — only you've given up a significant degree of control over what was once yours.

"All this means making a significant cultural change, and learning some new behaviors."

That will take some work on the part of your engineering, QA, support, and sales teams.

Nine must-dos for operations

Taber says attending to the following operational considerations

can make or break your open source project.

1. Clean code: Make sure that the IP is clean of "contamination." Engineer modular APIs around any technology you will need to replace.

2. Logistics: Get the logistics right, particularly in how you expose the source tree to the community and which tools you use (CVS, ANT, JUNIT, and CACTUS are the de facto standards).

3. Testing: Expect to test more often and more thoroughly than in the past to ensure ongoing compatibility.

4. Architecture: Strengthen the architecture, so you can design out proprietary libraries (such as MFCs), increase portability, make the code base more modular, and establish "private interfaces" for any value-added code of your own.

5. Politics: Work the community political issues such as feature roadmaps, release planning, or dealing with partners who start behaving like control freaks.

6. Sales policies: Develop a "party line" on customer issues, such as the old favorite, "I just paid for this product, and now you're giving it away free?!"

7. Sales support: Train and compensate your sales and channel teams so they won't disparage the open source version.

Give them tools so they can upsell your revenue-bearing products without falling into pot-holes.

8. Training: Train sales and support on how to work with partners. You will get new partners in the community, and you must not treat them like competitors.

9. Support offerings: Create new support offerings for your existing customers, and price them realistically.

If you do all that, your open source project stands a chance of success.

But if all this sounds like too much work, maybe you should reconsider going open source after all.

Even in the brave new world of open source, there's no such thing as a free lunch.

Resources to learn more about open source

[Open Source Code: Managing the Opportunity](#)

This 90-minute panel discussion in December, 2004 featured a good selection of thought-leaders in open source software:

- Veteran entrepreneur Dan Bricklin of [Software Garden](#),
- IP lawyer Karen Copenhaver from [Black Duck Software](#),
- Development manager Paul Cormier from [Red Hat](#), and
- VC Ben Howe, co-founder of [America's Growth Capital](#).

[Living with Open Source: How ISVs Can Turn the Monster Under the Bed into a Profit Partner](#)

This 90-minute seminar from Software University was presented by Donald K. Rosenberg of [Stromian Technologies](#), a consulting firm specializing in software and open source licensing.

Rosenberg is also the author of "[Open Source: The Unauthorized White Papers](#)" a book that applies a business approach to open source software.

[What is Open Source Marketing?](#)

If you — or someone you report to — need a remedial lesson in why people are excited about open source in the first place, check out this intriguing manifesto from marketing guru James Cherkoff.

It's not specifically on software, and it's not how-to, but it is thought-provoking.

And it may be just the thing you need to get your colleagues or managers all stirred up about open source.

[Back](#) | [SoftwareCEO Home](#)

© Copyright 2002-2005 SoftwareCEO Inc.