

# Release 1.0

The conversation starts here

[Home](#) > [Fresh Produce](#) > Open Source Community: How to win friends and influence developers

## Fresh Produce

### Open Source Community: How to win friends and influence developers

Jul 22, 2005 By [Dave Rosenberg](#)

*Part one in a three-part series on how to cultivate a strong community around an open-source product.*

If you were to contemplate starting a software company in today's market, you might consider one of many open source models. Clearly the IT marketplace wants open-source applications. Why else would they keep moving up the stack and continue to take market share from proprietary software companies? Databases, ERP, CRM, business intelligence, mobile applications - there is no category that can't spawn an open-source counterpart. There's a low barrier to entry, adoption and business acceptance is on the rise and there is a huge developer community to help build your product. Ah yes, the revered open source community, the invisible mass that supports projects and leads to success! If only you could harness the power and interest of the community, this whole thing would be a cakewalk.

But first, think about this: Open source is a "how," not a "what." It's a development methodology and distribution model that uses the Internet as its main communication method, both for managing globally dispersed development teams and as a vehicle for marketing and customer support. For many companies that sell open-source products commercially, their communities have been key to their initial and ongoing success. Providing everything from product roadmap input, to donating code, to performing QA, the community often forms the basis for sustainability of the project, and ultimately the business model. These communities usually evolve from developers who support the individual projects, helping to build the larger open source meta-community, which ultimately helps to define the open source space in general.

When forming a company that participates in and interacts with the open source community one has to consider this larger meta-constituency of end-users, technologists, developers and pundits. Admittedly, this idea is a bit odd. But the rules in open source are different than much of the rest of the technology world. Within the open source space there is an underlying thread of reciprocity and support of the group as a whole over individual concerns. Certainly, not everything done in the open source realm is noble or even beneficial, but by and large the community does support a greater good - the furthering of open source and open standards across all technology platforms and devices.

#### Open source is the new punk rock

It's very hard to draw a non-esoteric social parallel to the open source movement. My favorite analogy is the "scene" in the world of independent music - specifically, punk rock.

Punk Rock Scene	Open Source Community
Help put on concerts	Contribute code
Buy a record or t-shirt	Buy a license (or t-shirt!)
Put up flyers	Post on message boards
Start a record label to publish albums	Start a website to publish work

Change a business model by playing songs that aren't radio-friendly 2.5 minutes long	Change a business model by offering software as a service
Etc.	Etc.

The end results are the same: the furthering of the meta-constituency via participation and collaboration by a broader group than those who started the project. The notion of reciprocity is also embedded throughout both concepts; putting up flyers for a band you like gives you direct access to the members, just like contributing code. It also makes you visible to other scene-sters who you would like to support your efforts in the future.

If we abstract the analogy a little further into present day, we can see how market demand drives the evolution and "progress" of both software and punk rock. We've begun to see open source technologies emerge as business models. Similarly we've seen bands go from underground to mass-market with an ever-shifting fan base. Take Linux as an example; OSDL CEO Stuart Cohen says that "90 percent of Linux development is now done by developers who work for large companies like Novell, Intel, IBM and HP." This has both positives (technological progress) and negatives (community resentment). The model has evolved to where the "product" is now built and distributed by huge corporations. It's different from its origin but not necessarily bad.

### Is community the basis for success in open source?

To a large extent the answer is clearly yes. Consider just a few of the immediate benefits that an engaged community delivers to a project: QA, bug fixes, load testing, knowledge base, and enormous cost-reduction across the entire development process. But the bigger questions are: How does a commercial/dual-license open source company go about getting the community involved and contributing to a new project? How does a company's relationship with the community work as a marketing tool? What are the prevalent models of community development that new companies can follow? Is community the basis for success in open source, or merely an adjunct support mechanism?

In this series we'll examine the three prevalent models of community development that have emerged in open source. This research was originally conducted to help entrepreneur Steven Smith, CEO of FiveRuns, an open-source startup in the systems management arena. Steven and his team felt confident that they had a good idea for an open source-product, but were not comfortable yet with how they could interact with the community and further their product development. The goal of my research was to figure out how his business model stacked up to others, and to ascertain how a new company could work symbiotically with the community. As it turns out, dealing with the community is a lot of work. Not only are you working with a globally disparate group of individuals, but you are asking them to give you their time, knowledge and support for free. In addition to this community-oriented research, FiveRuns did focus groups in 4 states to test its product ideas and determine target markets. As Smith will tell you, when starting a company, especially one in open source, there is no such thing as too much information.

I interviewed ten companies for FiveRuns. I will profile seven of them in this series. Each fit into one of the three models, but had developed its communities in different ways.

Companies profiled in this series:

- Greenplum - business intelligence and data warehousing
- MySQL - database and data-management tools
- SleepyCat - developer database
- Sourcefire - network intrusion detection
- Splunk - machine-data search and analysis
- SugarCRM - customer relationship management
- Zend - Web application platform

### Open Source Community Models

By way of explanation, my research was initially based on the hypothesis that every company with an open-source product went about creating community differently - that is, there was no visible model that companies could follow. This theory was easily disproved, as I was able to classify the large majority of open-source companies into three major models:

- Founding Technologist - Projects are built and launched by technologists who are visible and vocal in the developer world use their personality in addition to technical knowledge to garner attention and respect
- Built from Scratch - Community is built from scratch, based on the merits of the project
- Established Leader - Community relies on or is based on an "established leader" (non-founder) in the open source or other technology space.

The lines between the models often blur, as products mature or other similar products are offered. And of course, there are always derivatives and exceptions to the community models stated above, as well as new business models such as services companies like SourceLabs that have arisen to "manage" LAMP

(Linux, Apache, MySQL, PHP, Perl or Python) environments and companies like IT Groundwork that use open-source tools as the basis for their software products.

Marketing to a developer audience is still a bit of a black art, and in the case of open source relies heavily on the social domain of human action as described by John Seely Brown and John Hagel III in their recent book, "The Only Sustainable Edge." In terms of developer marketing, they write, the social aspects - "the complex relationships between how we define our individual identities and the forms of social participation that we pursue to shape these identities" - are even more heavily nuanced than one sees in consumer or even IT marketing. For many developers, the participatory nature of the community associated with their development platform is key to their engagement. Developers want to align with what they consider to be the best technology and the one that should win, even if it doesn't. Open source developers fall even deeper into this social morass, as open source goes far beyond being a technology or economics choice and instead is often viewed as a socio-political statement.

In the end, there are really only two ways to market to developers: spend a lot of money or build a community. Startups, particularly those that sell free software, are unlikely to have a hefty marketing budget and will rely on the community model as well as other guerilla marketing tactics such as blogs, t-shirts, newsgroups and developer events. Larger, more well-funded corporations often find it easier to buy advertisements in trade pubs, blanket train stations in ads and get their message out to as large an audience as possible. As we'll see further on, community support provides an invaluable viral marketing tool.

### **Founding Technologist Community Model**

The first community model is the founding technologist model. I'll focus on the other two models in future installations of this series.

In the past, the majority of commercial open-source products received lukewarm attention from most of the community unless there was a highly vocal and visible technologist behind the project. The obvious advantage in this model is that community members have direct interaction with the "people in charge," which is not only a big selling point in the community but also in relation to users feeling that the company hears what they say. One could say that Apache, with Brian Behlendorf building the community, and Slashdot.org, with Rob "Cmdr. Taco" Malda as a prominent voice, were the biggest impetus for this approach as the community embraced vocal advocates.

In many ways the Founding Technologist model is more of a challenge than even starting a community from scratch, if you consider the odds of finding a visionary engineer who is capable of developing a good product in addition to being a good front man, communicating with developers and media while fending off detractors. Let's also not underestimate a true believer's ability to communicate excitement about what many would consider mundane or obscure details about software code. Both MySQL and Sourcefire were founded as commercial entities after the key technologists introduced the open source versions of their products out to the community. Each has a specific technologist who is recognized and respected among open source developers in the community, who delivered support initially, and who continues to be a vocal proponent today. According to FiveRuns CEO Smith, this is probably the ideal situation for starting a company in the open source space. But, he notes, "It's a rare find."

### **MySQL: The world's most popular open-source database**

MySQL develops and distributes the MySQL database and supports the product with a variety of tools and services. As it serves as the "M" in the open source LAMP (Linux, Apache, MySQL, PHP, Perl or Python) stack, to say that MySQL is a darling of the open source space is an understatement. The company has almost single-handedly legitimized the open source business model. This is not to say that it did this alone, or doesn't have detractors, but as a business MySQL has been quite successful, with 65 percent growth in revenues expected from 2004 to 2005. The company appears to be poised to continue its success based on the increased adoption of the MySQL database in large enterprises and bulk sales contracts with large vendors such as Novell and Veritas. The critical success factor going forward will be to transition non-paying customers to paid customers and getting existing database-only users to adopt the paid support services and additional tools the company develops and offers.

MySQL was started in 1995 and began taking venture funding in 2001. To date, the company has raised roughly \$21 million from Benchmark Capital, Index Ventures, Holtron Ventures, Scope Capital and a variety of angel investors. The company currently employs about 200 people, including a semi-traditional sales staff - an important factor in that a direct sales staff runs somewhat counter to the open source distribution model. The staff is in place to make the bigger deals with large enterprises that have licensing concerns, or for specific applications that require professional services. In 2004 MySQL added telesales capability. It also hired account managers for large accounts and to resell MySQL in volume to independent software vendors. While 53 percent of the units sold in 2003 had price points below \$1,000 (yielding only 4 percent of the company's revenue), 69 percent of the company's revenue came from deals larger than \$10,000. This suggests that MySQL may have no choice but to continue to grow its sales force. Estimated revenue for 2004 was \$12 million and is expected to grow approximately 65 percent, reaching about \$20 million. Not bad for free software.

In addition to the database and data-management software products, MySQL offers packaged and

custom consulting and training services as well as membership in the MySQL Network, which provides software updates, proactive alerts, and other maintenance items. The Network is supported by core MySQL developers who are intimately familiar with the code base, which contributes to the quality of the support as well as the communication with the user community. MySQL database consulting and training is available for everything from architecture to performance to tuning. The company launched a certification program in 2003 to provide credentials for varying levels of expertise.

CEO Marten Mickos estimates the total number of downloads of MySQL at roughly 50 million, but is unsure due to its bundling with other products such as the Apache Web server. The average number of downloads per day is about 40,000, meaning roughly 1.2 million per month. The company estimates the conversion rate of free-download to paying customer (in any form) is about 1 percent. Over the last few years, with the establishment and "normalization" of the LAMP stack, MySQL, along with programming language PHP, have become easier to use and more enterprise-ready. As such, they are being adopted as legitimate tools to support enterprise application development. At this writing, word is that the upcoming PHP5 will not enable MySQL directly, which means that it will be much easier for those installing PHP to choose another database, or simply not use one at all. This example is a key strategic factor in relying on the bundling of your product with other open source components, as other projects very often have different roadmaps and goals.

One very important finance lesson from MySQL is the need to balance the costs of goods sold (COGS) against the savings benefits of the open source model. In fact, preventing sales or marketing expenses from become too high is one aspect of the open source business model: First, limiting costs by using open source distribution methods such as using the Internet as the primary medium for delivery of product sales, marketing materials and customer support; and second, converting casual product downloaders to customers with minimal human interaction. Keeping COGS down is key in any startup, but even more so when only 1 to 6 percent of the people downloading your product are paying for it.

### **The MySQL community**

As a commercial company, MySQL has been well received by the community, media and analysts. Much of the kudos go to the executive team, which is viewed as members of the community first and of a corporate entity second. You'd be hard pressed to hear anyone say a bad word about any of the key staff. The project founders, Michael "Monty" Widenius, David Axmark and Allan Larsson, are still visible in the community, speaking and writing and - possibly most important - working on the code.

MySQL enjoys fantastic brand awareness, both among customers and the tech industry in general. The fact that the LAMP acronym was developed certainly helped with brand visibility as well as adoption, as there is now a standard open source stack. Early adopters of the product appreciated its performance, reliability and ease-of-use, as well as the fact that MySQL presented an alternative to Microsoft SQL Server and Oracle database products. As mentioned above, the community relationship clearly helped in building a user base, but it has been CEO Marten Mickos that has been hugely successful as an engaging and funny media personality and brand spokesman. He has helped demystify open source with his ability to explain the business model in a straightforward, strategic context and to pretty much anyone. He is also a consummate professional, in stark contrast to the layman's perception of days past in which open-sourcers sport pony-tails and hacker t-shirts, eat pizza, and don't care about business. To some extent, Mickos legitimized the entire open source movement via his interviews with mainstream media outlets such as Forbes and Fortune, and represented the "business-ification" of the space for trade publications such as CNET Networks' News.com and InformationWeek.

Beside the media visibility, MySQL has done an excellent job of relating to its users on a one-to-one basis, with developers directly responding to e-mails and newsgroup postings. Mickos said that the company is "extremely diligent in responding to user requests and feedback." It also should be noted that the core MySQL developers (including CTO Widenius) chime in to help when needed. This helps to maintain relationships with the community and add a level of direct contact that isn't easily obtainable from proprietary software vendors. However, large vendors such as Microsoft and Sun have realized the value of this involvement and have made significant efforts to engage with their constituents (and beyond) through weblogs, developer events and other communication vehicles.

Could MySQL have been as successful without the community involvement? More to the point, would the community efforts have been as successful if MySQL didn't have the founding technologists to base its community on? It's hard to say, but I lean towards no. The product itself is well-liked, but is unlikely to have received such widespread visibility if not for its mass of non-employee supporters. Also, when you consider the vast number of other open-source products that didn't make it into the LAMP stack (including PostgreSQL, a competing database that many consider to be technically superior), you have to think that the MySQL juggernaut is not based on technology alone. If MySQL hadn't had the home team of developers who were supporting the product and building the roadmap, it's unlikely that the company would be doing nearly as well as it is. According to Mickos, "The community involvement in MySQL is and has been absolutely critical for the success of the product. When Monty and David released the first version back in 1996, they took great care in delivering true value to the community: great software with source code open; thorough documentation that was constantly improved upon; super-fast responses to e-mails from users; affordable prices for the commercial offerings. In return, the community rewarded MySQL with many invaluable things: lots of users; lots of viral marketing; lots of bug reports and bug

fixes; lots of add-on tools, utilities and APIs; lots of books and articles about MySQL; and, as a result, some early paying customers who kept the young company going."

### **Sourcefire: Network security**

Sourcefire develops Snort, open-source intrusion detection software developed in 1998 by Martin Roesch as a network-sniffing utility. The company also offers security appliances that address a variety of network defense issues.

While Snort itself has been available and in use publicly since 2001, Sourcefire, the commercial company behind it, wasn't well-known until about 12 months ago. In fact, the company is still not that well-known, an issue I discussed at length with its chief marketing officer, PR head and outside PR professionals. They all suggested that this was on purpose - they wanted to grow the brand organically. I think it can be attributed to two factors: Positive - the company is doing very well and didn't need much PR or marketing for the short term. Negative - there was a lack of a communications strategy in place. Regardless, the company is now making an effort to be seen and heard, and in fact has a good story to tell both as an open source and as a security company. The company has raised about \$34 million in funding from investors including Sierra Capital, New Enterprise Associates and Sequoia Capital.

Of all the companies profiled, Sourcefire appears to have the broadest reach in terms of potential customers: any company that connects itself to the Internet. With the increased proliferation of phishing schemes, viruses and DOS (denial-of-service) attacks, network security and management is more important than ever.

In addition to a business's own need to secure its network against data theft and spam, data-protection regulations such as Sarbanes-Oxley require businesses to be more methodical about data security and storage or face government fines, potential customer loss and stock devaluing. This has clearly been a boon for Sourcefire, which had 2004 revenues of about \$20 million and is on target for \$35 million in 2005 revenues.

### **The Sourcefire community**

Founder and CTO Roesch "lives and breathes" the products and remains an active participant in both the business and the community. Highly respected and approachable, Roesch hasn't benefited from the open source PR machine that MySQL has, but is very well-known in the security space. He continues to post to the Sourcefire newsgroups and mailing lists on a regular basis, and remains a member and often the leader of the Snort community. Sourcefire has built a very engaged user base that contributes to testing the base product releases and the weekly Snort rule updates, and continually makes suggestions on how to improve the product. Beyond the online community support, there are Snort user groups all over the globe (including the US, Brazil, Japan and Germany) that meet in person to talk about Snort, networking and security in general. The e-mail lists alone average 500 to 600 postings per month.

According to Roesch, the open source community "continues to be vital to the success of Snort. The Snort community provides three essential contributions: It constitutes the world's largest early-threat network, it provides a way for Snort to quickly get tested on many different platforms and configurations - much faster than any single company could ever dream of, and it makes ongoing contributions to the base product with bug-fixes, documentation and product suggestions."

The community support itself is not dramatically different than most other open-source products, but what's interesting is that the majority of the Snort community members are security and network administration staffers rather than developers. This is an excellent example of a situation where the benefit to the open source nature of the product is in the distribution and subsequent adoption - i.e. viral marketing - and not in the visibility of the code, which becomes a nice-to-have feature. It also reaffirms the idea that open-source products are not just for development tools, but can provide legitimate, cost-effective solutions to a host of IT problems.

*The next piece in this three-part series will focus on the "Built from Scratch" community development model and features SugarCRM, Zend, and Sleepycat, three companies that based their community development efforts on the merits of the open source products they produce.*

## **Related Produce:**

### **Opening the Door on Open Source**

Mitchell Baker, Kim Polese

originally presented at PC Forum

### **Counterpoint: Open Dialogue**

Mitch Kapor, Open Source Applications Foundation  
Tim O'Reilly, O'Reilly & Associates

(originally presented at PC Forum 2003)

---

[About CNET Networks](#) [Jobs](#) [Copyright](#) ©1995-2005 CNET Networks, Inc. All Rights Reserved. [Privacy Policy](#) [Terms of Use](#)