Good and Bad Reasons to Open Source Your Software

How do you measure up?

by Ibrahim Haddad and Frederic Benard

A company's decision to contribute to open source projects is usually business-driven. This article offers a review of the top reasons that can influence your company to contribute source code to open source or to start new open source projects.

About the Authors

Dr. Ibrahim Haddad is Director of Embedded & Open Source Technology at Motorola. Prior to Motorola, Dr. Haddad managed the Carrier Grade Linux and Mobile Linux Initiatives at the Open Source Development Lab (now the Linux Foundation). Prior to joining OSDL, Dr. Haddad was a Senior Researcher at the "Research and Innovation" department of Ericsson's Corporate Unit of Research. He received his Ph.D. in Computer Science from Concordia University in Montreal, Canada.

ibrahim.haddad@motorola.com

Dr. Frédéric Bénard is Engineering Manager at Motorola and leads the Open Source Software Center of Excellence, which is part of the Motorola "Embedded Systems, Open Source and Linux Technology Group". He holds a B.Sc. in Physics from McGill University, a M.Sc. and a Ph.D. in Physics from the University of Toronto, and an MBA from McGill University.

frederic.benard@motorola.com

1. Provide a reference implementation to a standard:

Open source is a potential venue to launch reference implementations of a specification or standard, with the added benefit that such implementations can gain faster acceptance than if they were kept proprietary. Examples of companies that launched open source projects to provide a reference implementation to a standard include:

• Motorola, which open sourced its high-availability middleware stack, called OpenSAF, to provide a reference implementation to the Service Availability Forum specifications. For more information on OpenSAF visit http://www.opensaf.org/.

• And Ericsson, which open sourced the Transparent Inter-Process Communication (TIPC) protocol to provide a reference implementation for the Linux Foundation Carrier Grade Linux specification version 2.0. Today TIPC is integrated into the Linux kernel. For more information on TIPC visit http://tipc.source-forge.net/.

2. Ensure that critical software remains viable:

If you use an open source project and need to ensure that it continues to progress, remains active, and attracts new contributors, it's important that your company step in and start contributing. If you're already a contributor, you may want to increase the level of your contributions and possibly take a leading role in the project and motivate others to contribute by making the project interesting and challenging.

### 3. Ensure that new features are implemented ("scratch your own itch"):

If you're using open source code and realize that you'd like a certain capability implemented, the best way to get it done is to do it yourself. Leveraging open source happens when others share the same "itch." However, don't expect a project community to implement features that are of interest only to you. One example of "scratching your own itch" is Motorola implementing and open sourcing Precise Process Accounting (PPA) functionalities, a Linux kernel patch that improves the accounting of CPU and scheduling activities in carrier-grade servers and results in increased performance, capacity, and reliability. For more information on PPA visit http://sourceforge.net/projects/ppacc/.

4. Take control of your own destiny:

In many cases, companies create open source projects or support existing open source projects in the hopes that these projects become successful, thrive, and eventually become viable alternatives to the propriatey solutions of commercial software vendors. This allows such companies to reduce or eliminate commercial vendor "lock-in".

5. Undercut the competition:

Companies can use open source software to reduce their development costs by sharing these costs with others. Moreover, they can collaborate with the open source community and reuse open source components to accelerate the development of their products
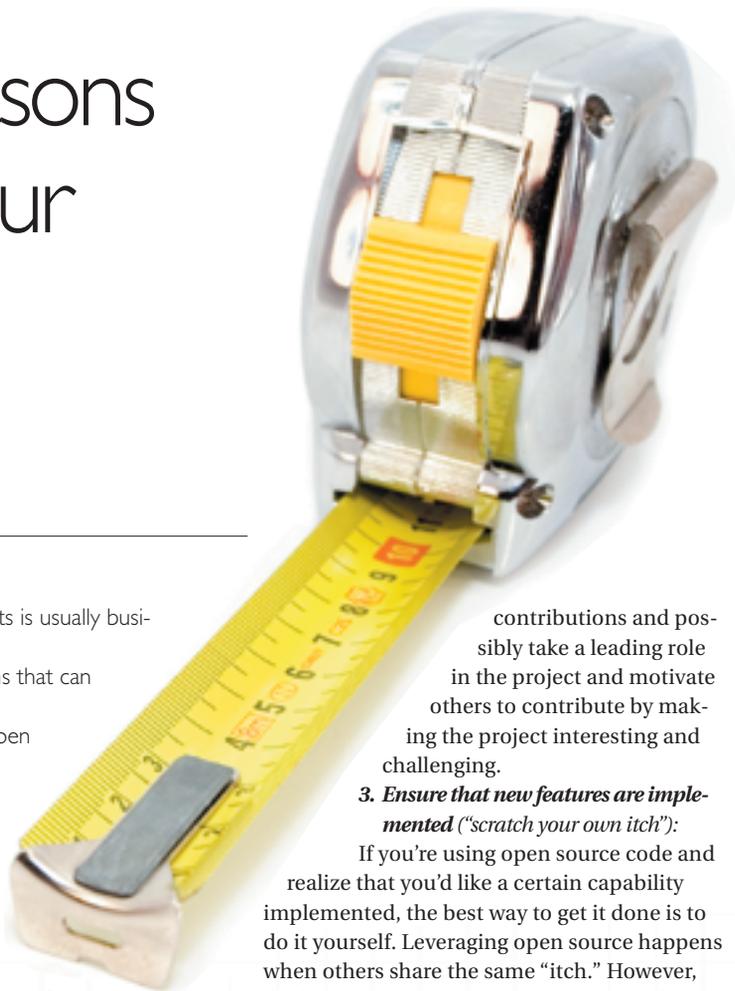
and reduce time-to-market. As a result, open source can provide cost and cycle time advantages with respect to the competition.

6. Commoditize a market:
When open source software meets the requirements of most users, the effect is to commoditize a market, reducing the pricing power of commercial vendors. For instance, free or low-cost Linux distributions have disrupted the market for Unix-based operating systems; as a result, traditional Unix vendors such as Sun and IBM have shifted focus to offer services. As a company, you may be interested in commoditizing a market to benefit yourself (via lower costs) or to put your competitor in a difficult situation.

7. Partner with others and promote goodwill for your company in the developer community:
There are many examples of companies that work with the open source community and contribute to open source projects to advance projects as well as promote themselves as good open source citizens that not only use open source but also contribute. By doing so, such companies are establishing a relationship with their software suppliers, which, in this case, are open source software developers. Examples of such companies include:
- Motorola, which developed a Web portal to foster collaboration between Motorola and the open source community http://opensource.motorola.com.
- IBM, which was one of the early adopters and supporters of open source and its significant participation and collaboration with the open source community is highly regarded and respected. To access IBM's open source zone visit http://www-128.ibm.com/developer-works/opensource.
- And Nokia, which has successfully partnered with the open source community to develop its Linux-based Internet tablets, the N770 and N800. For more information visit http://www.maemo.org/.

8. Drive market demand by building an ecosystem:
Open source software and the open source community can help companies create an ecosystem around their products and, as a result, drive market demand.
- Motorola has launched the MOTODEV initiative, which provides comprehensive resources to developers and enables developers to create applications for Motorola devices, leading to higher demand for Motorola products. See http://developer.motorola.com/ for details.
- Hardware vendors such as AMD and Intel have partnered with the open source community to create code enabling their hardware and chips to work with Linux, increasing demand for their hardware.

9. Offer your customers the ability to support themselves and add custom features for them:
This is a great approach to work with your customers, allowing them to contribute to your product development and help them support themselves.
- Motorola is partnering with the open source community to translate the interface of its A1200 cellular phone (Motorola Ming) into different languages. See http://opensource.motorola.com for details.

… And the Three Worst Reasons
Open sourcing your software isn't always the best business strategy. This section reviews some of the worst reasons you can have to open source a particular technology or create a new open source project:
1. You have obsolete software that you want to get rid of and you think that open sourcing it is a good way to get rid of it.
2. You're looking to leverage free engineering from the open source community and by open sourcing you think open source developers will lineup to contribute to your project.
3. You have software that you don't need anymore and instead of terminating the project or canceling it, you decide to open source it and then call it a win.
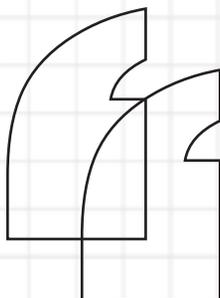
To be successful in open sourcing a project, you must have the right reasons or motivations. Furthermore, it's always recommended to do due diligence: If there's any other open source project with which your project might compete then study the opportunity to join and contribute to that project. Otherwise, you'll be creating competition and, in the spirit of open source, the strongest and best project wins mainstream.

Conclusion
There are many reasons to contribute to open source and there are various benefits to be realized from such engagements. In all cases, it's important to remember that it's a "give and take" relationship: be a good open source citizen, contribute to the community in good faith, and respect and follow community practices.

In a follow-up article, we'll discuss the process to follow after you decide to contribute such as selecting a license, doing a legal review, understanding the intellectual property implications, training employees, building a project infrastructure, announcing the project, following the open source development model, being visible, and being a good open source citizen driving the success of your project.

Stay tuned! ⊙

> To be successful in open sourcing a project, you must have the right reasons or motivations. Furthermore, it's always recommended to do due diligence: **If there's any other open source project with which your project might compete then study the opportunity to join and contribute to that project.**