

# But I'm not a developer...how can I contribute to Open Source?

Dru Lavigne <[dru@theopensourceadvocate.org](mailto:dru@theopensourceadvocate.org)>

October 11, 2004

## **1 Abstract**

The BSDs and other Open Source projects have made great strides in the past decade. But it doesn't take a marketing analyst to see that we've still a long way to go. We've all seen friends and family fight their way through viruses and spyware because their operating system of choice is "easier to use". We've all had customers or bosses who chose solutions involving hefty license fees over free software because of the "support". And we've all at one point or another had to maintain a non-BSD install because there wasn't an Open Source application available that provided a required functionality.

The question is, "what can be done about this state of affairs?" As a developer, your role in Open Source is fairly straightforward: write good code, add new features, fix bugs. However, the non-developer's role is less clearly defined. This often leaves the end-user feeling isolated and intimidated; not only are they unsure how to contribute, they may not even see their contribution as worthwhile.

This talk will address the importance of non-developers to the success of Open Source software as well as what you, as a developer, can do to assist in this process. We'll do this by taking a look at both the developer's and non-developer's points of view which will bring to light various misconceptions. Once these are dealt with, concrete roles can be defined.

## **2 Introduction**

Those who stay in Open Source tend to be technically minded. For the developer, sysadmin, and power user, the ability to look under the hood and play with the guts of a system holds a powerful attraction. Yet, history has proven time and time again that the success of a project isn't based solely upon technical merit. There is a danger in only approaching things from the technical perspective.

This presentation provides an opportunity to temporarily step away from the technical aspects of Open Source in order to view the larger picture—which includes the "softer" side of computing. As we explore different perspectives, take the time to consider the questions which are raised and to examine your own attitudes and actions.

## **3 The Novice User:**

Being a new user can be overwhelming at times—and that initial learning curve often seems insurmountable! While you may be brimming with enthusiasm over

your new discoveries, you probably also feel the twang of inexperience. How can you possibly contribute when you feel surrounded by gurus?

*Start by becoming aware of the resources available to you.* Browse the available mailing lists and subscribe to those that look interesting. Find an IRC channel. Go to your project's documentation and FAQ pages; bookmark them and start reading. Pick up a cheap system to try things out on so you don't ruin your main system. Become good friends with Google and look for tutorials and how-tos. Scour your local library or friends' bookshelves for quality books. Above all, join a local user group! Even if you're a shy person, hanging out with like-minded people does wonders for softening that learning curve. If you can't find a group that concentrates on your particular operating system or project, join a similar group. Keep your eye out for installathons, conferences and local events which you can attend.

*Once you find the resources, don't be afraid to use them.* However, do yourself and others a big favour by doing your research first. For example, are you considering posting a question to a mailing list? Before shooting off that email, use Google to see if the answer to your question is already documented somewhere. If you can't find an answer, have you researched the best mailing list to post the question? And more importantly, have you read the posting rules for that list? Have you included the details of your situation? Other users may be gurus, but they're not psychics. And if the rules say "don't email the developer directly", don't email the developer directly!

*Never underestimate the value of your perspective.* Remember, you're not the only new user out there. If you receive an answer and still don't understand something, ask for clarification. You'll not only help yourself, you'll help the other users who were afraid to ask. And, while documentation is steadily improving, there are still pockets of non-userfriendly information out there that assume a lot of previous knowledge. If you have to struggle to figure something out, write a clearer explanation and post it for the benefit of others. Keep that documentation ball rolling: whenever you find a useful tutorial or article, take a minute to send an appreciative email to the author.

Before you know it, you may find that you've matured into...

## **4 The Experienced, Non-developer**

*As your technical skills improve, try not to forget what it was like to be a new user.* For now the coin has turned and newer users are looking to you for advice and encouragement. Try not to let a bad or harried day result in a rude or negative answer to a question. For example, while you may be tempted to respond with "RTFM", something as simple as a "You're in luck. There's a really good tutorial

on this very subject at URL” maintains the user’s dignity while pushing them off into their own explorations. If you don’t have the time or patience to find suitable documentation, you don’t have to answer the post!

*Remember, it’s not so much what you say as how you say it.* What you say in a public forum continues to have a ripple effect long after you’ve forgotten about it. You can probably remember at least one incident when you were made to feel stupid for asking a question. Don’t repeat that cycle for someone else.

*Attitude has impact.* Watch for negativity when answering posts and for fanaticism when promoting your project over others. Watch the tone of your posts. Do you find yourself promoting Open Source successes or negatives? Take a step back if you find yourself repeatedly saying ”that won’t work” or ”things were always that way”.

*Now’s the time to start taking a closer look at the prevalent atmosphere of your community.* Does it promote exclusivity? Are ”outsiders” welcome or does it feel more like a ”techie boy’s club”? Is there an obvious gender skew? A recent survey published in Software Development magazine (1) probed the reasons why Open Source suffers from a greater lack of female participation than the IT industry as a whole. Does your community tolerate inappropriate remarks or turn a blind eye to discriminatory behaviour? Does it encourage female users to actively participate and promote the successes of those that do? As a female, are you lurking in the woodwork or are you an active participant?

*What about your project’s goals?* Does it require users to maintain an all or nothing attitude? Are users denigrated if they continue to use an alternate operating system instead of converting over to your operating system of choice? After all, isn’t open source about choice?

*Have you found that your initial enthusiasm has devolved into a general apathy?* How many times in the past year have you:

- found a bug and neither reported it nor submitted a patch?
- scoured the Internet in order to cobble together a solution yourself without sharing it so the next person didn’t have to reinvent the wheel?

Will Open Source continue to get better if noone:

- contributes feedback,
- files bugs,
- submits feature requests,
- donates money or hardware,

- creates patches,
- writes missing documentation,
- provides examples, support, or advocacy?

NO!

Take a look at your own systems. On top of the operating system you're probably running literally hundreds of Open Source applications. When's the last time you visited the websites of those applications, or for that matter, Sourceforge or Freshmeat? You won't have to look very far to find a project in dire need of exposure, beta testers or documentation. *Pick a project and contribute!*

## 5 Using your Pet Peeves

Sometimes you see so much need you don't know where to start. Well, anything that bugs you is a good indicator of potential action on your part. For example:

- *Do grammatical errors, undocumented switches and out-dated manpages drive you batty?* Sounds like you should investigate the submission process of your community's documentation project.
- *Frustrated by the lack of documented material available in your native language?* You may be a much needed translator.
- *Bugged that your place of work or school doesn't use Open Source?* Install what you can on the systems available to you. Show others what they're missing. Start an informal group which meets at lunch or after hours. Volunteer to setup and maintain an Open Source computer lab at your child's school or at the local seniors residence.
- *Tired of reading bad press regarding Open Source?* Write opinion pieces and product reviews which showcase the positives of Open Source. This is a largely neglected area as most writing efforts concentrate on technical how-tos. The irony is that we live in a world where artists and musicians who spend obsessive amounts of time honing their craft are "creative"; likewise, athletes are "driven". Yet the technically adept are relegated to "geek" or "nerd", both of which have connotations of social ineptness and fall pretty low on the "cool" scale. Negative terms like these do damage, but often this damage is unseen as potential users are turned off and go elsewhere. Media pieces are needed showing just how cool and fun it is to be involved

in Open Source. Or perhaps you'd prefer to speak to students at your local high school or community college.

- *Don't like the look of your project's website?* Volunteer your HTML talent.

Perhaps you're more in tune with your likes than your dislikes:

- *If your dream job would involve installing and playing with software,* find a project that is looking for beta testers or install current on a spare system and join the current mailing list.
- *Are you an avid reader?* Most publishers offer free books if you'll write and post a review. When you're finished, donate the book to your local user group's lending library.
- *Love to talk and help others?* Find a forum, list, or IRC channel and look for opportunities to assist new users.
- *Were you born to be around other people?* Organize and/or volunteer at a local installfest. Even if the venue's main event isn't specifically about your project, see if you can get a booth.
- *Do you wish to see Open Source promoted in business?* Research and create a list of vendors in your area that support, use or promote open source. Find or create an association that promotes Open Source in your community.
- *Just don't have the time?* Use your coffee money to buy a CD subscription for yourself and a friend. Commit yourself to make a dent in a project's donation page. Sponsor someone's admission fee to a conference.
- *Do you want to see more articles and how-tos but feel your writing isn't up to par?* Online ezines such as Daemonnews may still be interested in your submission. Do you find that your favourite ezine isn't always published on time? It's probably because the ezine is short on proofreaders and formatters.

Finally, don't be intimidated when you're...

## **6 Dealing with Developers**

*As a developer, you may not realize just how intimidating you are to non-developers. There is a definite mystique regarding the ability to decipher what appears to be so much mumbo-jumbo (to the non-programmer) in order to solve a problem. You*

really do speak another "language" (pun intended). Developers also tend to attract the spotlight, much more so than the average Open Source user. These factors discourage many users from reporting bugs or making feature requests.

*There are several things you can do as a developer to promote participation within your project.* First, ensure there is a supportive infrastructure including mailing list(s), FAQs, to-do lists and a bug reporting system such as Bugzilla (2). Make it clear on your website what a user can do to contribute and the steps they should follow when doing so. If you really don't have the time to respond to individual requests, say so and give users an alternative. If you do respond to individual requests, please don't patronize the user or trivialize their request. What's obvious to you is probably something they've never even heard about.

When dealing with a developer, the usual rules regarding researching your request apply, if not more so. When you send an email to a mailing list it is seen by thousands of users, some of which probably have the time, inclination and knowledge to reply. When you send an email to a developer, you're subject to one personality and one person's time constraints. Make sure the developer is willing to receive personal email and that the email includes the information required to respond in a helpful manner. Finally, don't take it personally if the developer doesn't immediately respond.

## **7 Food for Thought**

I'd like to leave you with some additional points to consider. Not every user of Open Source is interested in becoming a technical guru. This is actually a good thing and such users should be encouraged to contribute to Open Source in non-technical ways. Imagine the boost to Open Source if it could benefit from the talents of those who:

- are involved in media or have promotional skills.
- work in government, understand governmental process and have contacts within government.
- are participants within the legal process.
- are educators who are familiar with the process of creating and submitting curricula.
- are successful business persons.

Remember, successful "networking" has nothing to do with cabling or switches. Our ability to make contacts and to encourage others to contribute their talents directly affects the success of Open Source.

## **8 Additional Resources:**

<http://www.theopensourceadvocate.org>

## **9 Footnotes:**

1. <http://www.sdmagazine.com>
2. <http://www.bugzilla.org>