The art of selling free software

CREATING A BUSINESS

The open source community doesn't have a fleet of attorneys and PR consultants. When it is time to make the case for free software, you might just have to be the advocate. **BY JON "MADDOG" HALL**

Last night, I was talking with a free software advocate who had started a company. I asked what his company did, and he answered "free software." Now this hit a sore spot with me, since I run into a lot of people who tell me they want to make a living "selling free software." Sometimes these people have actually borrowed some money (or used their own money), started a business, hired some programmers, created a product, published the source and object code, then found anguish because they couldn't generate enough revenue from their software to make a profit.

The best way to get ahead is not to have a business that creates free software but to use free software to create and facilitate a business. When my friend answered that his company did "free software," I asked him to define what his customers could actually purchase from him and how his customers would benefit from his software. He did not have a ready answer. In forming any company, you have to set a vision, a mission statement, and a concept of what you are going to sell the customer. Free software is not enough.

To illustrate a productive relationship between business and free software, I often use the example of the steam turbine industry. Five turbine test centers around the world help engineers evaluate their steam turbine designs. Four of these centers use proprietary software that often requires a month for a simple change. The fifth turbine test center uses free software (MySQL, Linux, GNUplot, Tcl/TK, and Python) to create much the same web-based functionality, but the free software shop can often make small changes to the software overnight. All the test centers offer testing services to engineers designing the turbines, but the one that uses free software also offers rapid software customization services. I think it is obvious the test center that provides easy software customization offers a competitive advantage.

While I am on the topic of starting a business, on the same trip I talked with some young programmers who were going to write "free software" and put it in the "public domain." I started asking them questions about exactly what they were doing, and I found out they were a little confused about what "public domain" software is and what the GPL as a license afforded them. When they said "public domain," they were really thinking about writing software under the GPL license. In fact, they were

more than a little confused about the topics of copyright, licenses, trademarks, and other legal issues surrounding the production and publishing of software.

I will not attempt to define all of these topics here, as they have been covered many times in a variety of places. What I will do is caution programmers who are moving into the free software business to investigate and really understand the issues of copyright, the various free and open source licenses, and the implications of each license. These issues can have a dramatic effect on the revenue flow of your business.

Finally, as a tip for all of you who are thinking about starting your own consulting business using free software: Instead of starting out on your own as an individual, try forming a cooperative with some of your free software friends. Although there are many ways of forming a cooperative, usually the best way is when the workers in the business are also the owners. Companies often balk at working with a single individual. They wonder what will happen if that one person loses interest in programming or moves on in their career. A cooperative offers a reassuring permanent presence. If the main programmer leaves the cooperative, the organization can recruit a replacement. Additionally, the members of the cooperative can save money by sharing costs for legal, advertising, sales, and other services. ■

Free Software