Free Software Needs Profit

John Ousterhout

Sooner or later, every successful open-source software package will have one or more profitable businesses associated with it. For example, the Linux operating system is supported by several companies including Red Hat and Caldera; the GNU tools are supported by Cygnus; the Sendmail email delivery system is supported by Sendmail, Inc.; and the Tcl scripting language is supported by my company, Scriptics. This association between open-source software and business is not just a historical artifact, but a necessity. On its own, open-source software lacks essential ingredients for mainstream adoption. Commercial businesses tied to open-source packages provide these ingredients and help drive open-source packages into the mainstream. At the same time, they provide additional resources for developing the free components of the software.

Most open-source projects start out with a single programmer solving a personal problem and making the solution available to others. The early adopters of an open-source package tend to be like the creator: sophisticated programmers with a particular problem that the open-source package solves. These people can tolerate the risk of using a package without much documentation or support, and they are happy to help improve the package so that it better meets their needs.

However, the most successful open-source packages fill a much broader need that goes beyond the core community of sophisticated programmers. As the user community for an open-source package grows, it eventually becomes dominated by mainstream corporate users who have less risk tolerance and are often not sophisticated programmers. These users have different needs than the core developer community, and open-source packages don’t usually meet all their needs. For example, open-source packages typically have weak documentation and often lack introductory or tutorial material. They are often packaged inconveniently, requiring compilation or configuration to install. Development tools are often missing. Informal support may be available through a mailing list or network news group, and the quality is often excellent, but there is no guarantee that any particular problem will be resolved quickly.

As an open-source package becomes more and more popular, a widening gap develops between the needs of the user community and the features provided by the core developers. Eventually, commercial ventures will form to fill this gap. The user community will demand the additional services and offer to pay for them, so the business opportunities eventually become irresistable.

The commercial activities around an

open-source package tend to evolve in a predictable fashion. The first products for a package are typically books and/or training to help people learn the package. For example, the publishing company O'Reilly & Associates has worked closely with the developers of packages such as Unix, the X Window System, Perl, and Tcl, producing a variety of books targeted at users of those packages.

The next products to appear after learning materials are typically packaged versions of the software and support. These products don't change the functionality of the software but make the software easier to install and guarantee that problems will be resolved quickly. One of the oldest open-source companies is Cygnus Solutions, which sells support and consulting for open-source tools from the GNU project, such as the gcc C compiler. Several of the Linux companies, such as Red Hat and Suse, offer packaging and support as their primary products.

Another popular area for commercial products is development tools. For example, Scriptics provides tools for the Tcl scripting language and Cygnus has recently begun selling additional development tools for use in conjunction with gcc. These tools are usually proprietary, even though they are used in conjunction with open-source packages.

Finally, as mainstream use of an open-source package grows, proprietary extended versions appear. The extended versions add features that are needed by corporate users but aren't as important to the open-source community. For example, Sendmail, Inc., will sell an extended version of the Sendmail package that includes better management tools, and Scriptics will sell Tcl extensions that are needed in specific commercial applications such as automated testing and finance. These companies often use hybrid business models—they continue to develop the open-source core and release it freely while also creating proprietary extensions that are sold.

My experience with the Tcl scripting language follows roughly the path I've described here. As the Tcl user community grew in the early 1990s I noticed that the lack of a good introductory book was a major hurdle to Tcl adoption, so I used a sabbatical leave to write one. The availability of my book (and several others that followed) helped fuel major increases in Tcl usage, but it became clear to me that additional commercial activities would be needed to sustain the language over the long run. In 1994 I left my faculty position at U.C. Berkeley to start a Tcl project at Sun Microsystems. The additional resources provided by Sun allowed us to port Tcl to the PC and Macintosh platforms; this required 5–10 engineer years of work, which would have been difficult to amass through the open-source community alone. In 1998 I left Sun to form Scriptics, which sells Tcl development tools, services, and platform extensions while continuing to develop the open-source Tcl core.

My experience at Sun and Scriptics is that open source and commercial developments are symbiotic. For example, Scriptics develops new open-source versions of Tcl and gives them away for free. This helps the open-source community, but also helps to grow the Tcl developer community, which increases the market for Scriptics products. Scriptics' position as the leader in Tcl development gives us a competitive advantage over any future competitors in the Tcl marketplace. As the company's revenues grow, Scriptics will be able to invest additional resources in open-source development, which feeds a positive spiral. Scriptics also maintains a large Web-based resource center offering easy access to all Tcl-related information on the Internet. This provides many benefits to the Tcl community, but also serves as an easy way for Scriptics to reach potential customers.

Some free-software purists object to the commercialization of open-source software. They believe that the most important aspect of open-source software is that it is free; by this standard, any association with proprietary products tarnishes the software. However, I believe that the most important property of open-source software is that it fosters the development of new technologies. With open-source software, a single smart programmer with a good idea can reach thousands of other people. Adding commercial products to an open-source package allows the package to reach beyond its early adopter community to a much larger mainstream audience. By this standard, commercial activities amplify the benefits of open-source software and help to ensure that the best ideas survive in today's competitive marketplaces. C

John Ousterhout (john.ousterhout@scriptics.com) is CEO of Scriptics Corporation; www.scriptics.com