

Adobe's Open Source Tightrope Walk

April 19, 2014

Adobe views its open source initiatives as part of a strategy to assist and grow the developer community in line with the company's products and markets. Adobe constantly evaluates new hardware and software capabilities. Over the past six years, it has developed and refined a process that strives to align with the company's interests while meeting the needs of its internal developer communities.

Open source software continues to gain momentum -- but what is not growing is an open desire among individual software developers to port their commercial Windows wares to Linux.

Open source support is not a mere in or out decision. Some software makers shape their business models to take advantage of open source support for some products but not others. For other software developers, making a commitment to support FOSS (free and open source software) often goes well beyond the notion of putting one's money where one's mouth is.

For software developers that are not dedicated open source firms, lending support to open source communities is one thing. Providing Linux users with subscription versions of their commercial products is yet another.

[Adobe](#) is a perfect example. The company has long supported open source. It has applied an open source strategy that aligns with its business interests first and operating system end users a distant second. Still, Adobe has ventured into a cross-platform world -- a place other software developers fear to go.

The shades of purity in open source technology are graying. Like many software developers that use open source code, Adobe's products are a mix of open code and proprietary additions. This can make it more difficult to port the product to Linux or open-source it without a cumbersome change in the company's business model or product license.

"Adobe has been strategically pursuing the concepts of open source and open development. Like most companies today, Adobe is a mix of open source and closed source, a hybrid model as it is best known," Dave McAllister, director of open source and accessibility for Adobe, told Linux Insider.

Case Study

When computer users -- and businesses -- migrate to the Linux OS, they do so knowing that favored Windows programs generally do not run on the free open source platform. In most cases, adequate Linux alternatives make the transition easy. Also, workarounds sometimes exist for running a licensed Windows program, either in a virtual machine within Linux or by using a Windows environment emulator such as Wine or CrossOver.

A few preferred Windows programs are available for purchase to users who migrate in the other direction to the Mac platform. That is not the case for Linux users.

The idea to create a groundswell of support for Adobe to offer paid access to Linux versions of some of its Windows workhorse programs began several months ago as a [petition](#) on Change.org. A European Linux server administrator posted it, asking Adobe to make available its commercial products like Photoshop as a licensed (paid) product for Linux.

Progress has been slow. So far, nearly 500 people have digitally signed the petition. However, Adobe has not directly responded to it or indicated if it would consider the request.

However, McAllister discussed with LinuxInsider some of the issues involved with Adobe's open source strategy.

Not an Open-and-Shut Case

Adobe views its open source initiatives as part of a strategy to assist and grow the developer community in line with the company's products and markets. Adobe constantly evaluates new hardware and software capabilities, and it builds products and features to take advantage of the latest technology, McAllister explained.

Over the past six years, Adobe has developed and refined a process that strives to align with the company's interests while meeting the needs of its internal developer communities. For example, Adobe is a member of the Linux Foundation.

"Obviously, such a process walks quite a tightrope between the business of generating profits and the benefits of open source involvement," McAllister noted.

Strategizing Open Source

Adobe has expanded its presence in open source technology releases, a commitment that's evident on [github](#) and [Sourceforge](#).

This support is in recognition of the power and capability open development provides to Adobe's customers and partners. Given the company's inbound and outbound activities, the decisions

made in order to release technology under an open source license and to an open source community are intriguing, said McAllister.

Adobe participates in numerous open source projects. Some of them involve contributing significant programming assistance, such as Cordova, which can be found at the [Apache Foundation](#).

Adobe directly maintains some projects, including Brackets, Topcoat and Snap.SVG.

It also contributes technologies from its research arena, including projects such as Thesis and Fondue.

Linux Support Coming?

Adobe support for Linux is unlikely, however. The company is not inclined favorably toward proposals to port its products to Linux through paid or subscription options or to develop specific Linux initiatives. McAllister pretty much ruled out those possibilities, saying, "At this time, Adobe has no plans to support Linux via a Creative Cloud subscription."

Evaluation of new hardware and software capabilities is an ongoing process as the company plans new releases. There's increasing interest among its customers in developing, delivering and using mobile technologies such as smartphones and tablets.

"Adobe is focused on those platforms, including developing and delivering technology on top of the Android platform, itself an open source project with roots in Linux," said McAllister.

What's the Hitch?

Part of the problem with porting commercial software to Linux may be licensing restrictions for open source software. Allowing competitors to look at a program's code and allow its free reuse involves proprietary concerns over intellectual property rights. Another factor for established companies like Adobe could be the cost versus potential revenue gains.

It is much easier for small organizations to go with open source than it is for big companies. Usually the words "open source" are associated with something of lesser quality that is not as reliable.

While that may have been true 10 to 15 years ago, open source has changed dramatically since then, suggested Igor Uzhviev, project manager and technical lead at [DataArt](#).

"The trend now is that more and more companies are going to open source with a pay-for-support model. A striking recent example is Microsoft, which put C# compiler with many other tools to open source and opened the [.NETfoundation](#) website," Uzhviev told LinuxInsider.

Process vs. Cost

The complexity of the application is a factor in determining whether to bring a Windows program to Linux.

"It often is a question of the cost of porting versus the net new revenue that cost will deliver. Certain languages port well into Linux. Languages like C++ and php and Java applications will recompile and work reasonably well," Mark Baker, Ubuntu server and cloud product manager at [Canonical](#), told LinuxInsider.

The vast number of Linux distros also can be problematic. Bringing an existing Windows program to Linux means doing the porting many times.

"A lot of software developers see too much work involved in bringing their existing Windows programs over to the Linux platform," Baker pointed out. "There still are a lot of independent software providers out there. When they look at the number of Linux combinations and have to decide about supporting one or more of them, they see too much confusion and back away."

Technical challenges are less likely to govern the decision than commercial considerations, though, he acknowledged. Ultimately, the bottom line is likely to drive any software company's decision.

From Wish List to Shopping Cart

So, is there a workable option for Windows software vendors to port their code to Linux? That depends on how the Windows software was implemented, according to DataArt's Uzhviev.

"Unfortunately, in general if software was not intended to be ported on Linux, like platforms from the very beginning, it is an extremely hard and resource-consuming process to do that. There is no silver bullet how to port the applications, and it is always a very specific case," he explained.

For example, if the software is written on C++, usually you need to re-implement the code (probably partly) to utilize only cross-platform libraries (e.g., Qt, Boost, STL, etc.) and cross-platform coding approaches. Only after that can you try cross-platform work. However, for huge enterprise systems, that is usually almost impossible. The amount of code is gigantic, and the systems are very complex.

If the software is based on .NET, as another example, then you can take a look at [Mono](#) project .NET implementation for Linux. It is limited, but there is still a chance you can run your application on Linux.

"If you write the application from scratch, the recommendation is obvious. Build your code only on cross-platform libraries, technologies and coding approaches. Then later you will have more flexibility to run it everywhere," noted Uzhviev.

Pay to Play

A paid subscription model could bring popular Windows programs to a particular Linux distro.

"That is a workable possibility," said Ubuntu's Baker. "If you look at Ubuntu, we already have a delivery system for hundreds of open source products. We also have paid products. This gives us a definite revenue stream. It also provides revenue for the ISP who takes a share. A good example of this subscription delivery is Steam on the gaming side," explained Baker.

Steam worked that out with Ubuntu. Steam as a company saw that there was a viable market -- its games run very well on Ubuntu, Baker added.

"It was that experience that pushed Steam developers to continue to develop their products for the Linux platform. That led to their development of the Steam Machine. That, of course, is Linux under the hood," Baker added.

Open Modeling

The notion of paying for support was always a key part of the business model associated with open source software -- but there are variations on the open source model.

Similar to having a variety of Linux distros, software developers have a variety of business models to follow in distributing their programs to Linux users. Perhaps more of those choices being available to end-users on the Linux desktop will make a smoother transition from Windows to the Linux desktop.

"There are many business models for software development with respect to open source and open development -- and with all of them it is possible to find successful companies," observed Adobe's McAllister.

"The model of open source distribution coupled with pay-for-support is certainly a viable model for companies whose choices align with that strategy," he said. "However, again, it is not the only business model, even in a pure open source environment."