

Open Source Software No Longer Optional

Open development and sharing of software gained widespread acceptance 15 years ago, and the practice is accelerating.

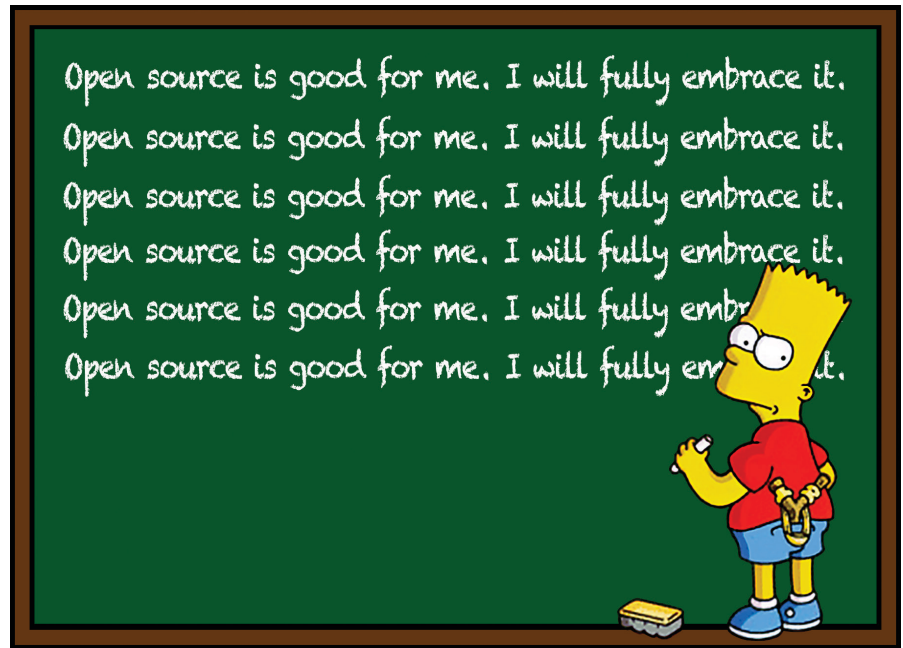
IN THE SPRING of 1991, a 21-year-old Finnish student named Linus Torvalds sat down to write code that would ultimately revolutionize the world of software development. In a Usenet newsgroup post late that summer, he told the world about his work: “I’m doing a (free) operating system (just a hobby, won’t be big and professional like gnu) for 386(486) AT clones,” he wrote. “This has been brewing since April and is starting to get ready. I’d like to know what features most people would want. Any suggestions are welcome, but I won’t promise I’ll implement them :-).”

Indeed, users of his Linux operating system have wanted a lot of features over the past quarter-century, and Torvalds has not had to add them himself. Linux today has more than 18 million lines of source code and some 12,000 participating developers. There are tens of millions of Linux users worldwide, from owners of Android smartphones to corporate data center managers to scientists at supercomputer centers.

It is a remarkable success story by almost any measure, and much of that success owes to a model of software development called “open source.” There are slightly different definitions of the idea, but essentially open source refers to software that is publicly available as source code and may be freely used, modified, and redistributed, without charge for the license (but sometimes with a charge for the service of distributing the software).

History

Software offered to the public in open source format is not a new idea. Beginning in the 1950s, a user group called SHARE, working with IBM, published applications and utilities as source code for use on IBM mainframes. In



the 1970s, AT&T Bell Labs licensed Unix source code to government and academic researchers pretty much without restriction.

The modern open source software (OSS) movement got its start with Richard Stallman, a staunch “free software” advocate, who wrote an open source text editor called Emacs in 1976. In 1983, he launched the GNU Project to develop a free Unix-like operating system and related utilities. At the same time he also founded the non-profit Free Software Foundation to promote wide collaboration in the development, distribution, and modification of free software, including GNU Project software such as GNU Emacs and the GCC C compiler.

The creation and use of OSS grew steadily after 1983, but users and developers, especially large corporations, often have looked at it askance. “Who would trust a mission-critical application to software written by a bunch of long-haired anarchists?” us-

ers asked. “Why would you give away software created at great expense to competitors?” developers wanted to know. William Scherlis, a computer science professor at Carnegie Mellon University and director of the university’s Institute for Software Research, says these objections are no longer valid, if they ever were.

Scherlis says the view of the open source movement as a combination of anarchy and demagoguery is a myth. “It looks like hundreds or thousands of people contributing from around the world, but generally it’s a very small core group that has intellectual ownership of the code base,” he says. “There’s usually a hierarchical structure so that control is maintained, and the major successful projects, like Apache and Eclipse, have elaborate ownership and governance structures: the Apache Foundation, the Eclipse Foundation, and so on.”

Allison Randal, president of the Open Source Initiative, which advo-

cates for OSS and maintains a list of industry-standard OSS licenses, says the open source community essentially declared victory in 2010, by which time she says the tide of opinion had flowed overwhelmingly from proprietary software to OSS. She cites a recent survey of 1,300 IT professionals by Black Duck Software that showed the percentage of companies running part or all of their operations on OSS had almost doubled between 2010 and 2015, from 42% to 78%. The number reporting they contribute to open source projects rose from 50% in 2014 to 64% last year, she adds.

Why Do It?

"It comes down to economic necessity," says Randal, who is also a development manager at Hewlett Packard Enterprise. "If nobody else was using open source, you could ignore it, but if others use it, they are getting something free and you are not, and they have an advantage. You can't be a start-up in Silicon Valley and not use it."

There are also reasons why large companies like HP increasingly use OSS, Randal says. "They get bogged down by their massive patent portfolios. OSS is a good way for them to innovate because they can pool their patents." For example, she says, the 500 companies that participate in the OpenStack project for cloud computing, including AT&T, IBM, and Intel, agree to license their patents to the neutral, non-profit OpenStack Foundation and thereby to all OpenStack users and contributors. "The companies have agreed not to attack each other with patent disputes around their collaborative work," Randal says. "It's a safe space for all of them to work in."

Last year, Google surprised some observers by releasing for general use the source code for its TensorFlow software, a set of tools for developing deep learning applications, including neural networks. It is the AI engine behind various Google apps—such as Google Photos, which can identify objects in pictures it has never seen before—but the code previously was off limits to external parties wanting to develop such apps.

Work on a predecessor system for deep learning, called DistBelief, be-

gan four years ago and resulted in a development and production tool for Google, but it was not flexible enough for others to adapt to their purposes, says Google senior fellow Jeffrey Dean. "If you wanted to do a more exotic neural network, some of those were hard," he says. TensorFlow was developed from the start to be open sourced and was written to minimize its dependencies on other internal Google tools and libraries.

Dean says Google traditionally has published the ideas behind its technologies in journals. By open sourcing TensorFlow, Google has gone further by making it easier for others to try Google's ideas and code in their own software. That will enable users to try different machine-learning techniques, spawning advances that may help Google in return. "We hope that a whole community will spring up around this, and we will get a wide variety of contributors, from students and hobbyists to large companies," Dean says.

For years, Microsoft lagged behind many other developers in its embrace of OSS. In a speech in 2001, Microsoft senior vice president Craig Mundie said, "The OSS development model leads to a strong possibility of unhealthy 'forking' of a code base, resulting in the development of multiple incompatible versions of programs, weakened interoperability, product instability, and hindering businesses ability to strategically plan for the future ... It has inherent security risks and can force intellectual property into

the public domain ... [OSS] isn't successful in building a mass market and making powerful, easy-to-use software broadly accessible to consumers."

Yet in 2004, Microsoft dipped its giant toe into the OSS waters with the release of the open source Windows Installer XML Toolset (WiX). In 2005, Microsoft open sourced the F# programming language and, soon after, a number of other things. Last year, it released the open source development framework and runtime system .NET Core, a free implementation of its .NET Framework, for Windows, Linux, and Mac OS X.

Microsoft now participates in more than 2,000 open source projects, says Anders Hejlsberg, a technical fellow and a lead developer of the open source tools C# and TypeScript. "New projects today are open source by default, unless there are good reasons why they shouldn't be," he says. "That's a complete switch from the proprietary mind-set of earlier days."

Microsoft is collaborating with Google on the development of Google's next version of Angular, the popular Web user-interface framework. The open source project will combine features of Angular with features from Microsoft's TypeScript, a superset of JavaScript. Says Hejlsberg, "Previously it was, 'Those are our competitors; we can't work with them.' But now it's, 'Oh, gosh, they are trying to solve problems we've already solved, and vice versa. We should work together for the benefit of both companies and the community at large.'"

While altruism toward the external development community sometimes plays a part, the open sourcing of .NET Core was mostly a financial decision, says Randal of the Open Source Initiative. ".NET is pretty old, and they hit a point where they realized they'd get more value out of releasing it as open source, getting a lot of eyes on the code, and getting contributions back."

Carnegie Mellon's Scherlis says recent open source projects have shown an increased focus on software assurance. "With OSS we have a chance to do better at providing users with not just code, but evidence of quality." That might take the form of test cases, performance evaluations, code analyses, or inspection reports, he notes.

Allison Randal, president of the Open Source Initiative, says the open source community essentially declared victory in 2010.

Downsides

Scherlis cautions not to get swept away with open source euphoria; major efforts like Apache may have tight governance, but some projects do not. He points to the devastating Heartbleed security bug discovered in the OpenSSL library in 2014 as an example; the bug left an estimated 500,000 trusted computers vulnerable to breaches of cryptographic security. “OpenSSL wasn’t a well-funded consortium, it was just a small group doing it,” Scherlis says. “But it was so good and so essential, everybody used it.”

Scherlis says users are dreaming when they think, “Many eyes have cast their gaze upon this code, and so it is good.” He explains, “It’s possibly true for shallow bugs, but not so much for the deep bugs that vex all projects—the global quality properties of the system, architectural flaws, concurrency problems, deep security problems, timing and performance problems, and so on.”

Finally, Scherlis warns OSS is not really “free.” In reality, most users will pay someone to adapt the software to work in their data centers and will incur internal support and maintenance costs for it. If the software is mission critical, the company will want to devote staff to the external open source project to ensure its needs are met over time.

OSS has become big business since Stallman started the Free Software Foundation. The company GitHub has become the go-to place for developers and users of open software, from large companies like Apple, Google, and Microsoft to thousands of startups. According to GitHub’s Brandon Keepers, the company hosts 31 million open source projects used by 12 million developers.

Keepers, GitHub’s head of open source software, commends Apple for the way it released in 2014 its open source programming language Swift. “The way they did their launch was one of the most impressive we’ve seen,” Keepers says. “They invited the community into the process.”

That is the wave of the future as developers take open source more and more seriously, Keepers predicts. “We are seeing companies treating open source launches like product launches. They want to make a big splash, but

“New projects today are open source by default, unless there are good reasons why they shouldn’t be. That’s a complete switch from the proprietary mindset of earlier days.”

they want to make sure there is support for the project after the launch. So you have not just coders, you have community managers, marketing teams, and product managers looking at what is the experience of users coming to the project.” **C**

Further Reading

Charny, B.
Microsoft Raps Open-Source Approach,
CNET News, May 3, 2001
<http://www.cnet.com/news/microsoft-raps-open-source-approach/>

Google
Interviews with Google’s Angular team about their use of Microsoft’s open source TypeScript
<https://www.youtube.com/watch?v=hvYnjJc88OI>

Kon, F. and Souza, B.
The Open-Source Ecosystem, *Open Source Initiative*, 2012
<http://flosscc.org/spread-the-word> [video]

Meeker, H.
Open (Source) for Business: A Practical Guide to Open-Source Software Licensing, *CreateSpace Independent Publishing Platform*, April 6, 2015
<http://www.amazon.com/exec/obidos/ASIN/1511617772/fatwave-20>

Stallman, R.
Free Software, Free Society: Selected Essays, 3rd ed., *Free Software Foundation*, Oct. 2015
<http://shop.fsf.org/product/free-software-free-society-3-paperback/>

Gary Anthes is a technology writer and editor based in Arlington, VA.

© 2016 ACM 0001-0782/16/08 \$15.00

ACM Member News

MAKING SOFTWARE SAFER, RATHER THAN FASTER



Michael Franz, a professor in the computer science department at the University of California,

Irvine, is co-inventor of the trace compilation technology that eventually became the JavaScript engine in Mozilla’s Firefox browser. He has spent much of his career making software go faster.

Franz says his “original background” is in programming languages and compilers.

He recalls reading a book on data structures and algorithms by Niklaus Wirth, the designer of Pascal and recipient of the 1984 ACM A.M. Turing Award, then writing him a letter saying, “this would be interesting” and asking “what can I do to study under you?” He received a response from the university registrar’s office, Franz recalls, “saying they had taken the liberty to enroll me in their series of entrance examinations, and asked could I be there in two weeks. I passed, and that is how I got the opportunity to study computer science at ETH Zurich, the Swiss Federal Institute of Technology.”

Today, Franz sees computer security as a more immediate problem. As a result, he has been helping to pioneer artificial diversity. “The idea behind artificial diversity is to generate multiple versions of a program, so an attacker doesn’t know which version is running, making it far more difficult to exploit.”

During the last few years, Franz says he has been focused on making things safer, rather than faster, although not at the expense of performance. “We invent a security feature, which creates a drag, and then we put on our other hat and optimize the drag away again. So in the end, we have a new security feature which is almost performance-neutral.”

—John Delaney

Copyright of Communications of the ACM is the property of Association for Computing Machinery and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.