



A Brief History of Free, Open Source Software and Its Communities

Jesus M. Gonzalez-Barahona, Universidad Rey Juan Carlos

Free, open source software (FOSS) has a long history, beginning with the origins of software itself, when the terms free software and open source software were not yet defined. Learning about the milestones of this history may help to understand FOSS today.

The concept of “free software” (with *free* as in *freedom*) dates from the early 1980s. The term *open source* is much younger, from the late 1990s.

But before free and open source software (FOSS) existed as such, some programs were paving the way. In fact, until the late 1960s, most software worked as FOSS: it was shared with relative ease between people who took care of computers. Only a few companies manufactured

computers, with IBM being, by a large margin, the market leader. For all of them, software was just a companion to hardware: as long as you paid for maintenance, you had access to the software catalog of the manufacturer. User groups, such as SHARE (IBM) and the DECUS [Digital Equipment Corp. (DEC)] favored software sharing. To some extent, prior to 1970, software was just an add-on to hardware, not something considered valuable in itself.

The situation changed in 1969, when IBM announced the unbundling of software: part of its catalog was to be sold separately. From that moment on, users had to purchase some of the software they needed. Various companies began to flourish with a business model based on producing software to be run on hardware sold by others. This kicked off the software market and, with it, the change of software’s status. Vendors implemented technical and legal means to limit sharing, modifying, and even studying programs. During the mid-1970s, proprietary (non-FOSS) software was already the norm. However, by the early 1980s, some programs were distributed in ways similar to what we now consider FOSS, among them, SPICE

FROM THE EDITOR

Welcome back, and welcome to a new thematic arc in our “Open Source Expanded” column! Until now, we have looked only at using open source software, mostly from a company perspective. Now, we will examine how community open source projects work, collaborating across volunteers and companies. I’m very happy to have convinced Jesus M. Gonzalez-Barahona, a long-time open source researcher and enthusiast, to write this opening article about the history of open source and its communities. He will take us through what are, by now, several decades of open source history. Enjoy! And, as always, happy open hacking, everyone, and be safe! — *Dirk Riehle*

(Simulation Program with Integrated Circuit Emphasis), TeX, and Unix.

In 1973, SPICE and its source code were placed in the public domain by their author, Donald O. Pederson. The program was a tool for learning integrated circuit (IC) design, and it was quickly adopted by several universities. With time, SPICE and its derivatives evolved into the industry’s preferred tools to design ICs, becoming

those parties formed a community composed mainly of academic institutions and research centers that worked in a similar way to later FOSS groups. Its members shared and improved the code, and the Computer Systems Research Group (CSRG) at the University of California, Berkeley, began producing its own Unix distributions. This was a keystone of the emergence of FOSS during the late 1980s. Earlier in

Until the late 1960s, most software worked as FOSS: it was shared with relative ease between people who took care of computers.

the de facto standard. It was the first example of how a FOSS-based strategy could lead to market dominance.

TeX was developed by Donald Knuth in 1978, during a sabbatical, as a typesetting system to produce quality output. Knuth intended to use it for typesetting his own books but distributed it as source code as well, through an authorization that today would be considered quite similar to a FOSS license. Since then, TeX has become the standard in scientific typesetting, and it is still popular.

Unix was created by Thompson, Ritchie, and others at AT&T Bell Labs, starting in 1972. Since 1973, Unix has been distributed to many universities, with a license permitting academic use. The software could not be disseminated beyond the signatories of the license, but during the late 1970s,

that decade, these cases provided some experience with how basic FOSS-enabled mechanisms worked.

THE 1980S: GNU, BERKELEY SOFTWARE DISTRIBUTION, AND THE INTERNET

In 1983, Richard Stallman announced the GNU Project, with the aim of producing a Unix-like system composed only of free software. Stallman, who was by then a programmer at the Massachusetts Institute of Technology (MIT) Artificial Intelligence Lab, quit his job to ensure that he had full ownership of the software he wrote. The project began with an editor (Emacs) and some other tools and quickly produced various key components. By 1987, it delivered a compiler (GNU Compiler Collection), a debugger (GNU

Debugger), and several utilities. In 1985, Stallman founded the Free Software Foundation to support and foster the GNU Project and free software in general.

He also established the philosophical principles of free software, including the definition of the concept. This characterization was based on “four freedoms” for any user of a free software program: use, study and modify, redistribute copies, and distribute modifications. The GNU Project produced licenses for the software it was releasing. Those licenses were the legal projection of the four freedoms. In 1989, they were unified in the GNU General Public License (GPL), the first of one of the most successful families of FOSS licenses. The GPL was a clever hack: it protected software users’ freedoms by using copyright law.

The GNU Project’s work was structured in small teams of volunteers who produced different pieces of software, according to a carefully designed plan. Around 1990, the project had almost completed an operating system. However, its tools were always running on top of proprietary or non-free kernels because it still lacked their own versions.

Meanwhile, during the 1980s, the CSRG lead a large community that was busy working on improving Unix, producing Berkeley Software Distribution (BSD) Unix. The community included people from the University of California at Berkeley and the University of California at Los Angeles to MIT; Stanford; Carnegie Mellon; and others. There were industry members, too, notably AT&T and Bolt, Beranek, and Newman, the company producing the first implementations of Internet protocols. The efforts were funded mainly by R&D grants from the U.S. government via DARPA.

With time, BSD Unix had less and less code from the original AT&T Unix and more and more code produced by its contributors. The original AT&T code was covered by the Unix license, but not all the new code was. In 1989, the code not covered by the Unix license was offered as Networking Release 1

(Net/1) under the BSD license, which was free. Net/1 still lacked some modules to be a complete, working operating system. At that time, several companies were using BSD Unix (including Unix licensed code) as the basis of their operating systems, and some of them were contributing to BSD with ports to specific hardware, new applications, and bug fixes. The effort of incorporating all this into the BSD code base was coordinated by the CSRG.

Another remarkable project of the late 1980s was the X Window System, which produced a platform-independent graphics system incorporating a protocol that enabled applications to use a graphics terminal, even remotely. X Window was released in 1986 under the MIT license, which was also free and, in many aspects, similar to the BSD license.

During the 1970s and early 1980s, another development community was creating software under similar models: the Internet (at first, the Arpanet) community. Since the early 1970s, it had collaboratively been producing requests for comments (RFCs) (specifications of standards) as open documents that were accessible to anyone. The protocols were complemented by reference applications, which were designed to be easily portable to manufacturers' systems. During the 1980s, the community developing Internet Protocols and applications was closely related to the Unix BSD group since BSD Unix was the usual target for developments. Later, as the Internet became popular at universities, its tools and protocols became fundamental for the development of communities supporting free software projects. In an epoch when remote coordination was still usually done via phone and postal mail, free software communities were already communicating via email lists and sharing software electronically via FTP or its poor-man version, the Unix-to-Unix copy network (UUCPnet).

Those were also years of testing sustainability models for FOSS. Projects

quickly became a mixture of people working on their own time as volunteers collaborating with people hired to assist them. In the beginning, hired developers mainly worked at universities, such as the teams at the CSRG and other BSD Unix contributors. In many cases, their funding came from R&D institutions, especially DARPA. But companies were involved in two major ways: by directly funding FOSS projects and by making their employees work on FOSS projects.

The most prominent case of a project funded by companies was X Window, developed at MIT, which jointly funded the work with DEC and IBM. This was one of the first projects to evolve from proprietary software (several licenses of X Window were sold) to being later released as FOSS in 1986. The project was so successful that several companies used it as the basis of their GUIs, at

led the development of some of those tools, such as the GNU debugger, assembler, and linker, all of which were fundamental to the GNU Project and the FOSS community at large. This not only helped to bring stability and resources to the project but it showed how companies could profit and grow from maintaining and building FOSS components by becoming a focal point of expertise. The fact that Cygnus was directly involved in the production and maintenance of FOSS projects signals how interesting such efforts were to many companies. Those organizations were using GNU tools in production environments, and they were ready to pay Cygnus for support and new functionality.

Another remarkable case of a company with a business model centered on FOSS was Aladdin Software, although for a different reason. Since 1986,

As the Internet became popular at universities, its tools and protocols became fundamental for the development of communities supporting free software.

a moment when GUIs were a key characteristic of workstations. Several of these companies assigned large teams to port X Windows to their systems and to build new applications for it. Some of the resulting software was contributed back to the FOSS project, showing the benefits of sharing upstream. In 1988, X Window was so important to numerous vendors that they decided they needed to formalize a neutral point to drive its evolution, forming the MIT X Consortium. This was the first case of competing companies establishing a nonprofit to provide stewardship of a FOSS project.

The best-known case of a company assigning employees to work on FOSS projects at that time was Cygnus Support, which was founded in 1989 to commercially sustain some of the GNU tools. Its employees had

Aladdin had developed Ghostscript, a PostScript interpreter, and released it under the GNU Project as GNU Ghostscript. But the company used a dual licensing model, maintaining its own version, Aladdin Ghostscript, under a non-FOSS license. With this model, it was exploring how dual licensing could prevent its competition from using the latest features in the software while maintaining a FOSS version that sustained the popularity of the program.

At the end of the 1980s, FOSS communities were complex in many ways, with people and companies collaborating by sharing software and, indirectly or directly, resources. They were exploring several sustainability models: public funding (via R&D grants), donations collected via nonprofits, direct funding from companies, the direct involvement of

companies via neutral consortiums, pure volunteer work, and combinations thereof. They set up a legal infrastructure centered on the two families of FOSS licenses that are still in use today: those based on the principles of the GPL and those established on the principles of the BSD and MIT licenses. They had a solid philosophical basis, formalized in several documents that were widely known in their communities. And they were

from the beginning as such, BSD had steering committees, X Window organized the MIT X Consortium, and so on.

THE EARLY 1990S: LINUX, *BSD, AND COMPANY

During the early 1990s, developments started during the previous decade converged in the first complete systems composed only of FOSS components: *BSD and Linux. In the BSD Unix camp, the CSRG had reimplemented

people involved in FOSS development and maintenance kept growing.

It was during the mid-1990s that the Internet evolved from an academic curiosity to a mass market service, with the web becoming the primary mechanism for accessing information and, later, digital services. The importance of FOSS components for Internet infrastructure was evident, being one of the enablers of the expansion of this technology. Most of the implementations of Internet Protocols were either FOSS or derived from FOSS projects. Many of the most popular services were implemented as FOSS, such as Sendmail and NCSA HTTPd (and later Apache), which were dominant among email and HTTP servers, respectively.

A new kind of FOSS-related company appeared that was linked to Linux-based distributions. In fact, many of the major Linux-based distributions were promoted by companies: Red Hat, SuSe, Mandrake, and others. They all began by marketing a Linux-based distribution and expanded to offer a mixture of services, from training to support, that were, in general, loosely based on their distribution and, to some extent, their brand. Other companies, such as VA Linux, joined this growing market of FOSS-based solutions.

FOSS emerged as a strategic tool that could be harnessed to build neutral consortiums, where competitors could collaborate.

producing software of interest to individuals and companies, both for ethical and practical reasons.

Companies were also learning how to benefit from FOSS development. Some small companies were trying pure FOSS business models. Others used FOSS as a viable model. And FOSS emerged as a strategic tool that could be harnessed to build neutral consortiums, where competitors could collaborate to produce software that they all found interesting. Some companies noticed how existing FOSS components could be employed to build large parts of complex systems, enabling them to leverage their own developments at a fraction of the cost of creating the modules themselves.

By this time, FOSS development communities used digital means for communication (mailing lists, Usenet groups, FTP servers, and UUCPnet), enabling them to work in large, geographically distributed networks. They explored organizational mechanisms that included, in some cases, appointed figures (such as GNU “maintainers,” who acted as leaders of their development communities) and de facto coordinators (as CSRG personnel were for BSD to some extent). Formal organizations for stewardship projects were already common: GNU was conceived

most of the missing components to produce a complete Unix-like system under the BSD license. This was distributed as Net/2. In 1992, 386BSD was released with an implementation of the small pieces that Net/2 needed, thus resulting in the first full FOSS system. NetBSD, FreeBSD, and OpenBSD were later evolutions.

Meanwhile, in 1991, Linus Torvalds announced his project for writing an operating system kernel, which would soon be named Linux. It quickly gained traction and contributions from other developers. In 1994, Torvalds released Linux 1.0, the first “stable” version, although the software was already usable in 1993, and, in some respects, in 1992. Many tools, including from GNU and BSD, were ported to it, and different groups started to produce Linux-based distributions (such as Slackware, Debian, or Red Hat).

Around 1993, *BSD and Linux-based distributions were perfectly usable, complete operating systems that could be installed on PCs. With time, Linux became the most popular, and during most of the 1990s, many cohorts of young developers, including students at numerous universities, were exposed to it. New FOSS projects, small and large, launched in many places and domains, and the number of

THE LATE 1990S AND THE 2000S: THE AGE OF FOUNDATIONS AND CORPORATIONS

In 1998, Netscape announced that its flagship application, Netscape Communicator, was to be released as FOSS. Netscape Communicator was one of the two web browsers that dominated the market (the other was Microsoft Internet Explorer), and Netscape was one of the most prominent companies of the new Internet era. Because of this, the announcement received plenty of attention from the media. To some extent, this event signaled that FOSS was becoming something real for companies, something that they could use as a part of their strategy. In preparation for the announcement, the term

open source software was coined as an alternative for *free software*, and the Open Source Initiative was formed.

At about the same time, large FOSS communities emerged. The GNU Project included a growing number of tools and members. People were also joining the Free Software Foundation. New projects were bootstrapped. Debian was one of them. In 1993, it was established to maintain the Debian Linux-based distribution, and it was soon joined by tens, and eventually hundreds, of developers. Debian was a community of individuals, where companies didn't have a role. In this respect, it followed the GNU tradition, although from the beginning, its governing rules, which were explicit, led to a much more horizontal organization.

Another community of developers was Apache, first built around the Apache HTTP server and then expanded with other FOSS components. In 1993, the project was born as the Apache Group, which expanded and formed the Apache Software Foundation in 1999. This, too, was a community of individuals, although many of its members were hired by companies. However, Apache tried to remain neutral with respect to companies, following a spirit that resembled Debian's.

The group of developers producing the Linux kernel was one of the major software development communities formed during the 1990s. From its beginning, the project was very clearly directed by Torvalds, with only a few formal governing rules. Although companies had no direct role, they hired many Linux developers, who often had clear interests in the system's development. In 2000, the Linux Foundation was formed to organize contributions from these companies and support the project although, in general, technical decisions remained relatively separate. Later, the Linux Foundation extended the model to many other projects that came under its umbrella.

In 1996, the Kool Desktop Environment (KDE) was born to develop a

FOSS desktop application. Partly as a reaction to KDE using some non-FOSS components, the GNU Network Object Model Environment (GNOME) was announced in 1997, with similar objectives. Soon, hundreds of developers joined both of them. Various companies began hiring developers to work on the projects because they wanted to drive the evolution of certain applications. This was the case, for example, with SuSe and Red Hat: the desktop environment of their Linux-based distributions was to be improved. Some others, such as Helix Code and Eazel, were small start-ups funded to develop specific applications.

GNOME and KDE established nonprofits to support the projects, and both found ways to let companies participate directly. KDE's nonprofit was formed in 1997, and the GNOME Foundation was incorporated in 2000. Using different mechanisms, companies that contributed significant resources participated in the projects' decision making, and combined with the influence they obtained by hiring developers, they had a real impact on the initiatives. GNOME and KDE were the most prominent organizations exploring the path toward communities of companies, which had begun with the MIT X Consortium.

Netscape launched Mozilla to produce the FOSS version of Netscape Communicator. But Netscape's new owner, AOL Time Warner, lost interest in the initiative. In 2003, the Mozilla Foundation was formed to legally steward the project, independently from AOL. From then on, the Mozilla Foundation searched for lines of revenue, which it found in agreements with companies, notably Google, that were interested in its flagship program, Firefox. Thanks to this revenue, Mozilla hired a large team of developers, and it also built a large community of volunteer supporters.

In 2001, the Eclipse project was created by IBM and supported by a group of software companies to produce a FOSS integrated development environment

and related tools, which then extended to many other domains. In 2004, the Eclipse Foundation was established as a neutral nonprofit to steward the project. It was formed by companies providing financial resources and by Eclipse developers. They all participated in strategic decisions.

These software development communities and their corresponding nonprofits have explored different relationship models between developers and the companies with interests in their projects. From the very developer-centric Debian and Apache to those with significant direct company participation (KDE, GNOME, and Eclipse), from those originated by companies (Mozilla and Eclipse) to those with origins in individual developers (almost all the others mentioned previously), from those with clear and detailed governance and participation rules (such as Apache, Debian, KDE, GNOME, and Eclipse) to those based more on practices and the personal charisma of some individuals (GNU and Linux), they all have produced FOSS components of interest. They have proved to be sustainable, remained attractive to developers (either hired or volunteer), and devised their own approaches to structuring productive FOSS communities. The current landscape of FOSS development is the result of this history. There has been progress, and there have been contradictions. There has been collaboration but also fierce competition between models, aims, and mechanisms. Today's FOSS is the product of it all. **■**

JESUS M. GONZALEZ-BARAHONA

is with the Universidad Rey Juan Carlos, Fuenlabrada, 28943, Spain. Contact him at jesus.gonzalez.barahona@urjc.es.