



PROJECT MUSE®

Peer Participation and Software

Booth, David R.

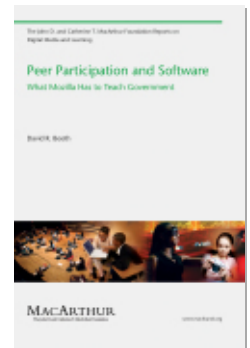
Published by The MIT Press

Booth, David R.

Peer Participation and Software: What Mozilla Has to Teach Government.

The MIT Press, 2010.

Project MUSE.muse.jhu.edu/book/19561.



➔ For additional information about this book

<https://muse.jhu.edu/book/19561>

Access provided at 1 Apr 2020 19:30 GMT with no institutional affiliation



This work is licensed under a Creative Commons Attribution 4.0 International License.

Licensing

Introduction

Essential to the success of the Mozilla Project today is Netscape's historic decision to license the browser software to the public under an open source license. Communicator source code was released in 1998 under "Project Source 331." This project marked Netscape's effort to release open source code to the public and resulted in the Netscape Public License (NPL), which became the Mozilla Public License (MPL). While GNU used the General Public License (GPL) to guard against businesses co-opting open source code for their own private benefit, the Mozilla Foundation licenses the Firefox browser source code under one of three open source licenses designed to encourage innovation while maintaining the integrity of the Mozilla brand. They are the Mozilla Public License, the GNU General Public License, and the GNU Lesser General Public License. Our focus on the MPL illustrates how licenses govern the redistribution of work by volunteers, while at the same time promoting participation. We conclude this section with the presentation

of two definitive features of open source software: forking and portability.

The Mozilla Public License

Like a constitution, a license is a set of rules that governs the rights of use, in this case with regard to the terms under which a programmer modifies code for distribution by Mozilla *and* himself, when his contributions are applied to other programs. There are many different kinds of licenses. Many organizations have developed licenses appropriate to their products and ideologies of distribution. From the point of view of the licensee, an open source license enables him to:

- Use open source software for any purpose whatsoever.
- Make copies of open source software and to distribute them without payment of royalties to a licensor.
- Create derivative works of open source software and to distribute them without payment of royalties to a licensor.
- Access and use the source code of open source software.
- Combine open source and other software.

The main question facing the licensee concerns how much he needs to contribute to the community. How much can he go off on his own? Open source is software that is available to anyone free of charge. Nevertheless, at Mozilla, if you improve software you have to make that improvement available to everyone, and have a social incentive to do so. This does not mean that a licensee necessarily has to publish at mozilla.org.

But he does have to make his modification available under the same license that granted him source code use in the first place.

The MPL creates recursion. Its reciprocity provisions create *return* and an incentive to participate as a member of the community. If a licensee modifies and distributes a file containing either the original source code or a prior modification to the original code, he must distribute his modification under the MPL. The licensee is permitted to use all prior modifications of the source code; at the same time, he is permitting future modification of his contribution.

Firefox as a Project Fork

In the late 1990s, the Mozilla Organization took over the development and management of the source code for the Netscape Communicator browser, which included the Netscape Navigator browser. The Mozilla Organization was in operation from 1998 to 2003, when it became the independent Mozilla Foundation. Today, the Foundation, which is synonymous with the Mozilla Project, owns the intellectual property (trademarks, brands, logos) and infrastructure (servers) related to Mozilla. Contributors keep copyright to their additions. This is the covenant between Mozilla and its contributors: copyright is ownership.

The creation of the Firefox browser under the management of the Mozilla Organization illustrates an essential aspect of open source coding. In 1998, one of the challenges Netscape faced was the right of an individual to apply her contribution to the Mozilla source code to the founding of a new project. The big

question: To what extent did Netscape need to guard against other businesses co-opting—or “forking”—its open source code for their own private benefit?

In software engineering, a project fork occurs when programmers base their development of a new software package on the source code of existing software. Open source software may be forked without permission.⁴⁰ Accordingly, forks can be sanctioned—“friendly forks”—or hostile. One of the essential advantages of forking is that it allows for and invites experimentation and innovation. The entire module ownership system at Mozilla is predicated on the fecundity of sanctioned forking. Sanctioned forking expands community by simultaneously increasing the number of participants and, by way of their participation, deepening the knowledge base of the community. The possibility that a programmer could appropriate Mozilla source code and then, after collaborating with the development community, abandon Mozilla necessitates a hierarchical and formal process of gaining commit privileges, as summarized earlier in this report. In short, the threat of a hostile fork requires strong leadership on the part of Mozilla and a public commitment to the Mozilla community on the part of the contributor.

The Firefox browser is itself the result of a sanctioned fork. The Mozilla Organization began development of what would become Firefox under the name Phoenix. Phoenix became the Firebird project, before the Firefox browser, a project launched as an experimental alternative to the Mozilla Suite, emerged as the main product of the newly formed Mozilla Foundation. As a free and open source Web browser, Firefox has consistently gained market share since its debut in November 2004. Each

incarnation of what became the Firefox browser was developed by a community of individual programmers extending beyond the employees of Netscape and Mozilla.

Bugzilla: An Example of Portability

A final role available to volunteer developers—one similar to the module owner—is that of the Bugzilla component owner. Bugzilla is an online, open-source bug-tracking system that merits mention because it is a profound example of *portability*, an aspect of open source that is conversely related to the practice of forking. To port software is to use it without modification, but to apply it to platforms for which it was not originally intended. Portability means that innovations can be adopted for unforeseen uses.

Licensed under the MPL, Bugzilla is like the Mozilla source code repository in that it too is a Version Control System (VCS). Designed by Netscape and launched in tandem with mozilla.org in 1998 via an anonymous VCS, Bugzilla allows registered users to report bugs encountered in their use of the Firefox browser and other software. Because the system is licensed under the MPL, it is portable: an organization other than Mozilla can adapt the system to any open source or proprietary platform free of charge, instead of creating a fork. Portability suggests the potential reach of open source software into the technological infrastructures of NGOs and governmental agencies alike, a potential supported by the fact that over eight hundred organizations are known to use Bugzilla, though the number may be much higher. These organizations include free software projects

such as Gnome, the Apache Project, and Open Office; Linux distributions such as Red Hat and Novell; and companies like Facebook, the New York Times, and NASA.⁴¹

This is the paradox of open source. The license creates the freedom to splinter off and develop new projects, while the peer production of distributed work creates the incentive to collaborate as a community. Anticipating the optimal level of forking versus coming together that will produce innovation is the key to success.

In conclusion, the Mozilla Public License is one component of the shared responsibility of transparency and collaborative governance. A viral license mandates that collaboration and transparency are repeatable and repeated. But the open source license is not sufficient. The license is the set of rules under which community norms are practiced and proliferate over time. Transparency requires vigilance on the part of the principals at Mozilla and, in the context of software, the programming community at large. As we have seen, this vigilance is made possible by the online infrastructure of the open source process.