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.

Open Source at Mozilla

Introduction

The independent Mozilla Organization was born in the late 1990s, at the apogee of the Silicon Valley dot-com boom. Even the proponents of free software who were nonetheless skeptical of the application of free-software doctrine to commercial interests saw the release of Communicator source code, the quickening of new projects under open source licenses, and the spinoff of the Mozilla Organization as allied with the intentions of a grassroots movement that began in the 1960s.

Today, the Mozilla Project is building a recursive public, a *constituency* dedicated to the improvement and distribution of Mozilla products via the Internet. Individuals volunteer their time and expertise to the Mozilla Project because the Internet is essential to their daily lives. They participate because they have identified a reason to do so, and because they can: Mozilla provides online, group-based structures for collaboration. These generalizations foresee the potential of similar models of participation to link private citizens to decision-making processes within government.

One way to begin understanding the relevance of the Mozilla Project to emerging forms of collaborative governance is to note that Mozilla's reason for existing is not solely to distribute products. In its manifesto, the Mozilla Project abstractly identifies the Internet as "an integral part of modern life." Slightly more concrete, the Internet is "a global public resource," one promoted by free and open source processes. If we combine the idea of the Internet as a public resource with another tenet in the Mozilla manifesto—that "individuals must have the ability to shape their own experiences on the Internet"—a portable picture of participation begins to emerge. We can see the end-user *advocating* the terms of her continued participation in the maintenance of the Internet as a public resource via her own input in the improvement of the products that enable her online experience.

Applied to technical and nontechnical enterprises, an open source process may influence the development, distribution, and ongoing improvement of products and services that are collaborative in nature. If for instance a potential participant in open-source software development is physically impaired *and* skilled in programming, he may by dint of his personal experiences and technical expertise self-select himself to collaborate in the creation of software that limits the number of keystrokes required for him to access resources on the Internet. We may modify this example in anticipation of our discussion on nontechnical input from participants in the Mozilla experience by imagining the same participant, physically impaired but in this case nontechnical in his avocation or vocation. If he has a vested interest in online services, he may still contribute his

input to an organization like Mozilla if a structure is in place to receive it. He may describe the limitations his disability places on his access to the Internet. In fact, some of the strides that Mozilla has made in increasing Internet accessibility for the visually- and mobility-impaired were born from the input of volunteers and organizations historically unaffiliated with Mozilla.[27]

In another example that comingles browser design and the inherent skills of browser-users, when version 3.5 of the Firefox browser was released in June of 2009, it shipped in over 70 languages; the development of these foreign-language versions of the browser was volunteer-based.[28]

One final way to introduce the parallels between open source and collaborative governance is to recall common rulemaking procedures that traditionally give private citizens a voice in government. Passed into law in 1946, the Administrative Procedure Act (APA) affords private citizens the right to comment on the specifics of new laws enacted by Congress and the president. As the constituents affected by a new law, individuals are invited to respond to the rules and regulations that interpret the language of that law. Known as Notice and Comment Rulemaking, this period of public comment is inaugurated with the publication of the new law (as a "notice of proposed rulemaking" [NPRM]) in the *Federal Register*, and generally remains open for between 30 and one 180 days.

Though the comparison between open-source software development and Notice and Comment Rulemaking is abstract at this juncture, several commonalities between these enterprises prefigure our discussion of collaborative governance:

- New laws and software modules are published and made available for public review.
- Both enterprises determine the nature of the feedback they are seeking. Both set the agenda.
- Participants offer feedback on a voluntary basis.
- Public feedback is garnered for set periods of time; in the case of open source software, code modifications are subject to deadlines, as predicated by project timelines.
- Just as self-selecting software developers are experts in their field, individuals who respond to NPRMs are commonly self-selecting because their professions or private lives will be impacted by the new law. Or they are experts summoned by governmental agencies because they can contribute scientific- or industry-specific expertise crucial to the wording of the rules and regulations under review.

Though this list is more suggestive than exhaustive, it articulates an *ideal* with regard to the potential not only for an organization to tailor products and/or policies to the needs of constituencies, but also for individuals to respond to policies of governing that most impact their daily lives. Of these policies, which would each of them be most qualified to work on, based on personal experience and enthusiasm? And if qualified to do so, how will they provide useful feedback to policy makers—their representatives—in government?

In this section our focus is on software development. Here, in anticipation of our expanding discussion of the potential influence of open source software on open government, we introduce a contemporary culture of collaboration between volunteer

software developers and the Mozilla Foundation. In describing the protocols by which the Mozilla Foundation solicits expertise from the public in the management of the Firefox browser, we present a system of governance best described as a *hierarchical meritocracy*. We begin with an explanation of the practice of distributed decision making, and its attendant system of distributed peer review.

Module Owners and Their Peers

Distributed decision making, a concept well documented by organizational and industrial psychologists, refers to a work environment in which "decision making is a continuous, interpersonal process, usually involving several 'decision makers' aiming at dynamic and cooperative control of the state of affairs at work."[29] Specific to the Mozilla Foundation and its appropriation of this concept, an introduction to aspiring hackers at mozilla.org states the following:

> The Mozilla project is far too big for any one person—or even a small set of people—to make ongoing decisions regarding code appropriateness, quality, or readiness to be checked into the CVS source repository. . . . The code is large and complex; the number of daily decisions to be made is enormous. The project would slow to a crawl if a small set of people tried to make the majority of decisions regarding particular pieces of code.[30]

This statement is the opening remark in a primer entitled "Distributed Decision-Making: Mozilla Modules and Module Ownership," which describes the role of the module owner in the production of the Firefox browser. A module owner leads the

development of a module of code. A code module is a collection of related source files. "Modules are chunks of code," explains Mitchell Baker, "and there are quite a number of them. The module owner is responsible for that area, that module; no change is made without her or his okay. You need prior review."[31]

The module owner designates peers to help him determine the utility of patches. Peers are developers with a proven track record from within the Mozilla community. Together with his peers, the module owner makes final decisions about modifications of the module he oversees.

Committers

A developer who successfully submits a patch to a code module associated with the Firefox browser is known as a committer. A committer receives permission from a module owner to modify its source code. If a potential committer is not one of the original developers of the Firefox browser, he seeks approval from established committers in the Mozilla community. "If you want to participate," explains Baker, "you can't put the code into the tree. There's another layer, called a committer. Before you're allowed to combine your work in the public asset, Mozilla needs to know you. Mozilla needs to be comfortable with your work."[32]

To become a committer, a volunteer developer begins by finding a project to work on and, after talking with established developers in Mozilla's programming community, submits a formal application to become a committer. While becoming

active in the Mozilla community by contributing to the online dialog at mozilla.org and joining mailing lists, the aspiring committer next submits a patch to a code module for review. He does so by nominating potential peers to "vouch" for his proposed patch. These individuals—awkwardly termed "vouchers"—act as mentors to the volunteer and, accordingly, assume responsibility for the newcomer's patch. If the volunteer's formal application is approved and his patch is proven effective, he becomes a committer. He has established relationships with active committers and module owners, who "vouch" for him as an expert, and he is granted "commit privileges" by the virtual management team at Mozilla. If he decides to continue with Mozilla as a committer, he may make a formal application to become a "voucher" or peer to incoming volunteers. He may also ascend to the role of module owner—an individual who manages the maintenance and development of a specific module in the Firefox code tree.

It is important to note that when occasionally Mozilla pays a developer to work on a specific module, that developer matriculates via the same process as a volunteer.[33]

In sum, individual recognition and advancement in the community-based production of software at Mozilla is a meritocracy, predicated on the utility of the developer's contributions and his resultant visibility and effectiveness within the online community. Brian Behlendorf, a founding member of the Apache Software Foundation and a board member of the Mozilla Foundation explains that the standing of a developer in the Mozilla community is formed on the basis of "the assignment of capabilities to various people, such as those with 'commit privileges.' If you've been awarded those, that can carry some

moral weight when having a conversation; [A] sense of who someone is, is based on [his] informal reputation in the community, his track record of contributions, that sort of thing."[34]

In an earlier section we noted that Mozilla leadership might actively steer programmers toward underrepresented projects. As we will see in greater detail in subsequent sections, this concept of oversight cannot be stressed enough in differentiating Mozilla's open source approach from other contemporary examples of crowdsourcing. Unlike entries published in the online encyclopedia *Wikipedia*, peer review of new code dedicated to developing Mozilla products happens before that code is implemented. Mozilla does not publish works-in-progress. In this way, Mozilla combines the knowledge base inherent in a self-selecting crowd of experts with the kind of leadership that defines a representative democracy.

Mozilla's Module Ownership System

Despite the word-of-mouth—and as such, *social*—nature of individual advancement in the Mozilla community, the Mozilla Foundation has published a specific protocol by which developers are qualified as committers.[35] Once an individual achieves committer status, she may join the virtual management team at Mozilla not only as a module owner, but also as a super-reviewer or a release driver.[36] A brief description of these roles further illustrates the application of distributed decision making to the production of software at Mozilla.

To summarize this process, we begin with the volunteer developer: she identifies a bug in a module—a problem she wants to work on. She submits her bug fix at mozilla.org with her

application to become a committer. An established committer acts as her voucher. This voucher often solicits the backing of a second voucher to determine whether or not the submitted bug fix requires super-review. Super-reviewers differ from module owners in that they scrutinize patches on the basis of the interoperability of code modules. Super-reviewers are good at integrating modules. They conduct what we may accurately term *integration review*.[37]

Whereas committers submit patches to Mozilla in response to their specific software needs as users of the Firefox browser, release drivers (another managerial role) steer developers toward bug fixes in anticipation of what Mozilla calls "milestone" software releases.[38] In contrast with module owners and super-reviewers, release drivers do not focus on the specific technical advances in source code; instead, they oversee management of the source tree in the time leading up to the release of a new version of a software application. Release drivers participate seasonally in the development of the Firefox browser. They are thought leaders and innovators from within Mozilla and also from the software industry at large—from universities and such companies as IBM and Red Hat—who periodically volunteer their time, enthusiasm, and expertise to particular projects.

The Governance Module

The module ownership system is mirrored in nontechnical projects with the creation of so-called activities modules. Each activities module has an owner, at least one peer reviewer (and often more than one), a volunteer newsgroup dedicated to the collection and dissemination of information about module

activities, and a specific list of responsibilities. Examples include the governance module, which is a module dedicated to the administration—staffing, scheduling, conflict resolution—of code modules, and Planet Mozilla, which is the module that, comparable to a virtual press office, maintains Mozilla's image in the blogosphere. (Planet Mozilla is a Web site that syndicates blogs devoted to the Mozilla Project.[39] The Planet Mozilla module owner and his peers are responsible for determining not only which blogs will be included at planet.mozilla.org, but also what content from selected blogs will be published.)

The governance module is broadly responsible for the processes by which the Mozilla Foundation distributes decision making. Though the governance module owner and her peers are not necessarily software developers, their management extends to oversight of the source code repository.

There are also submodules that manage governance functions. One fills vacancies on existing projects, staffs new modules, reviews the performance of module owners, and resolves conflicts involving module owners, peers, and contributing developers. Another governance submodule manages incubator repositories, which are temporary source code repositories for volunteer developers who are seeking commit privileges but are not yet well known in the Mozilla development community. Such repositories help educate new participants.

A Hierarchical Meritocracy

We have already suggested the power of both crowdsourcing and the oversight described by Linus's Law to source and utilize expertise. Self-selection means that everyone is invited to

contribute his time and expertise to the development of the Firefox browser. Individuals who make useful contributions to the Mozilla Project, and who demonstrate their desire to take on greater responsibilities within the virtual community by becoming increasingly involved in the online *culture* of Mozilla, gain prestige in the Mozilla development community. They may choose—and be chosen to—take on a managerial role. Despite the centrality of volunteer peer review in this process, module owners make final decisions. Most are volunteers. As such, the module ownership system is democratic, but also hierarchical and meritocratic at the same time.

Delegation of authority not only expands the knowledge base of the delegator—in this case, the module owner—but also distributes ownership of the consequences of final decisions. Delegation also increases the sense of belonging on the part of the individual who, on the basis of her abilities, has been given authority.

Despite these benefits, a manager's delegation of responsibilities to individuals in a group does not necessarily enable that group to make a collaborative decision. What makes participation in the maintenance of a code module collaborative is a developer's sense of autonomy, in combination with a shared sense of mission. She is autonomous in that she writes code in response to her personal experiences with the software. She is a collaborator because she submits her patch to a group of peers for review and possible implementation. And she is invited to choose what she wants to do.