Open Source Software for Government

by Matthew Burton

Project Overview

I am founding an organization that will allow independent Web developers to create free and open source software (FOSS) for governments. (FOSS is software that is freely licensed to all and whose inner workings--source code--is freely viewable by anyone.) In an essay I wrote three months ago, I proposed this idea to the Web developer community. Because of their enthusiastic response, I have decided to pursue the idea.

My organization will address three issues:

- The inability of citizens to become more than just voters and constituents, and interact on a daily basis with their government. While bureaucracies are the most common way a government affects the daily lives of its citizens, those citizens seldom affect those bureaucracies' daily operations. By allowing those citizens to write software for government offices, my organization will give people an avenue to actively participate in their own governance.

- Governments' dependence upon outdated, poorly designed, and extremely expensive software. Software manages myriad vital government services. Bad software keeps the government from providing these services effectively, and wastes billions of tax dollars in the process. The cost of software is especially prohibitive for governments of developing nations. FOSS, as the definition implies, is free: it is created by volunteers and can be used by anyone at no cost. And because those volunteers will come from the largest possible talent pool, our products will be superior to what the government is used to working with.

- Government transparency and corruption. In the last eight years, the internal workings of the United States executive have become more and more opaque. And the developing world, in addition to being unable to afford quality software, is plagued by government corruption. Open source software code is public, so anyone and everyone can find out what it does and how it works. Were the government to use such software, their internal operations would inherently become more transparent.

Background

In the U.S., almost every government employee performs their job in front of a computer. The government is aware that they are plagued by poor software. But efforts to solve this problem often lead to huge amounts of money being wasted on products that are never used. The most famous example is the Virtual Case File, the FBI's 2001 effort to update their document tracking system. After five years and $170 million, the contractor, SAIC, had completed 10% of the job. An outside assessment recommended that because the completed work was so poor, the FBI scrap the entire project. They did.

More recently, the National Counter-Terrorism Center completed Railhead, a $500 million project to rebuild their troubled terrorism watch list, which has famously kept everyone from babies to senators from boarding planes. According to a study by the House of Representatives, the new system is deeply flawed and does not even allow users to search for names.

These high-profile examples are clear cases of bad software hurting the public: software was supposed to fill gaping holes in national security, and it failed. But there are many more cases around the world that, though short on public exposure, keep government employees from performing vital public services. Millions of anonymous civil servants perform jobs that are essential to our daily lives. Their reliance upon inadequate technology makes the government just as inefficient as the faulty software

that runs it.

Meanwhile, the computer service most of us use at home--the World Wide Web--works beautifully. Web sites load quickly and are easy to navigate. Almost any question can be answered with a few minutes of Googling. Web tools are created by design-conscious developers, and if a product isn't easy to use, there is likely a better alternative. (In government, the opposite is true: once a contract is awarded, the winning bidder is free to deliver a poor product without the risk of being outperformed.)

And instead of exorbitant prices, many products created by Web developers are free--not free in the sense that users don't have to pay to use Wikipedia, but free in the sense that anyone can download the software that runs Wikipedia and start their own custom wiki in less than 20 minutes, without paying a dime. This is the essence of free and open source software.

The open source software movement is based on the idea that software should be free to use, modify, and redistribute. This movement has spawned some of the world's most important and popular software products: the Firefox Web browser, the WordPress blogging platform, Apache (which runs almost all Web servers), MediaWiki (which runs Wikipedia), and Linux are all free. The open source developer community is passionate about their principles: they believe in freedom, transparency, and accountability.

This community is also politically active. In the U.S. alone, many Web sites have been created in recent years let the public scrutinize DC politicians: what laws they are passing, who is funding their campaigns, etc. These are noble goals; no democracy is strong if its citizens are not monitoring their leaders.

But these sites only scratch the surface of our government, delving no deeper than the 536 elected officials in Washington. None of these sites can scrutinize the millions of civil servants who carry out the core functions of the U.S. government, or any other nation's government. These people work in cubicles, where the public cannot peer.

Aside from quitting their jobs and becoming full-time bureaucrats, how can these politically active citizens affect what goes on in those cubicles? I propose that they design the software that runs government. Why not take the talent, political activism and volunteer spirit of the open source developer community and channel it into improving government IT? The result would be billions of dollars saved, more efficient governance due to better designed software, a more engaged citizenry, and the inherent transparency achieved through open access to the code that runs governments around the world.

Planned Outputs

I am seeking funding to seed the initial stages of this project. After this initial stage, I anticipate the project will continue as long as it is solvent and useful.

Initial Stage Outputs

- Before my organization can create useful software for government agencies, I must research the logistics: What governments are amenable to such work? Can I start in the U.S., and if so, should I target federal, state, or local? What sort of corporate entity is best for this organization? (Such technicalities are extremely important to the success of this project: government procurement is a murky world, and small details regarding funding and corporate structure could make the difference between an attractive government partner and an illegal one.) Should I affiliate my organization with an existing one, such as a university? What is the business model? How will I sustain an organization whose products are free?

  Over the past two months, I've been talking to lawyers and former government contracting officers to find out the answers to these questions, and I'll continue to do so until they are resolved.

- I also need to find project champions in both the open source and government communities.

This will require time and travel. So far, I have been doing this at networking events in New York, and remotely, having floated a concept document with the directors of multiple university institutes.

- Finally, I must evangelize this project. Some of this will be done during the previous task in talking with government leaders. But I will also write about this project's progress on my Web site, in order to sustain the interest created by my original article; attend conferences; give presentations on the project to Web developers; and generate interest in the project through interviews with tech publications.

Long-term Outputs

This organization's natural place is as a go-between for computer programmers and government agencies. Our primary output will be software code based on specifications requested by government customers.

In a later section on achieving my goals, I describe how this process might work.

Measuring Success

I see a few tangible, objective metrics for the success of this project: services improved, tax dollars saved, products built, and citizens engaged.

- The primary goal of this project is to make governments more efficient and transparent. There is no single objective metric of efficacy, but there will be indicators as to the success of our products:
    - Noticeable improvements in the execution of public services. The Social Security Administration could report fewer payment errors after adopting our product. A nation's elections could go more smoothly upon deploying our voting software. Or perhaps we cut the FBI's FOIA response time in half. These would all be signs that our products were successful.
    - Positive feedback from government employees. Critical to a successful open source software project is the development team's ability to listen to users, so our products will encourage feedback from those users. This feedback will let us gauge user satisfaction.
- Government software contract prices can reach nine figures, even for products that will only be used by a few thousand people. My products, on the other hand, will be almost free. Upon completing a product, I can use past contracts as a guide to determine how much a government customer would have spent had they contracted a private firm.
- FOSS products can be freely redistributed, modified, and used for any purpose. I expect our products to be useful not only to the original government customer, but to outside organizations and other governments as well. We can track how often our products are downloaded and use that number as a measure of their utility.
- Finally, there is the aspect of citizen engagement. I am asking everyday people to play a new role in their governance. Each volunteer developer that I recruit is one more citizen that is playing an active role in their governance.

Outreach

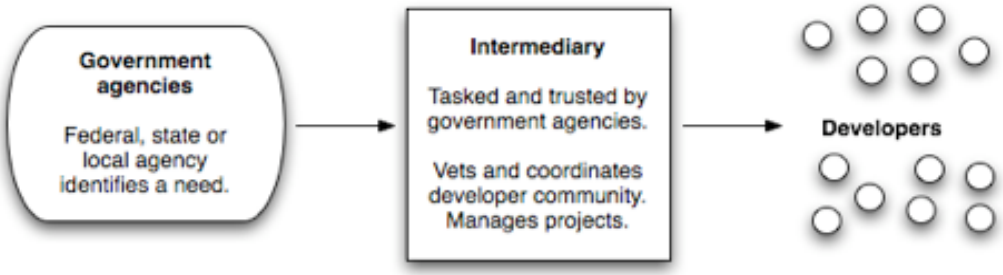
I've divided this section into two parts: one explains the software production process, and the other explains my efforts to found the organization and market the concept.

Production

Here is the lifespan of a hypothetical GovFOSS project. I'll use the example of a new election system.

The New York Secretary of State determines that its election system is outdated. It would like a new ballot infrastructure that includes an electronic ballot. The Secretary of State's office approaches my organization and outlines their needs. We create a set of specifications for the ballot system and explain to the SoS how such a system will practically work. We then notify our developer community about this new project and prioritize the various development tasks. As the volunteer developers add features to the software, we verify its usability and stability. The project, though incomplete, is also available over the Web for public testing.

Throughout this time, we keep the NY SoS abreast of our progress. Once the state is satisfied with its product and it has undergone sufficient public scrutiny and testing, we assist the state in deploying this software. The software code continues to be freely available for review, and contributions will continue to be accepted from vetted developers. Other governments may download the software and use it for their own elections.



Marketing

Since floating this idea in June, I have gotten a very enthusiastic response from Web developers and leaders of the open source movement. I first mentioned the concept of citizen-produced government software in an online essay, *Why I Help "The Man," And Why You Should Too* (attached to this application; also available here).

The attention from the essay led to an online interview in which I discussed the idea further. It is available here. As a result of the coverage, several developers have written to say that they are ready and willing to volunteer their time to this project. In order to sustain this interest, I've built a mailing list to keep interested parties informed. I'll also write about my progress on my personal Web site (I am storing all project details online here) and will market the concept at conferences; if able, I will do this through formal presentations.

Because of my experience in both government and Web development, I have contacts in both realms. I already have the support of some of the most prominent voices in the open source developer community, so I expect to be able to recruit enough volunteers.

Attracting government-side champions will be more difficult. Because of my history as both a government employee and a government transparency advocate, I have several contacts at U.S. government agencies--both at the federal and state levels--who play a role in technology procurement. I will also approach government offices friendly to experimental projects, such as DARPA, IARPA, the National Science Foundation, and the National Institutes of Health. I have few contacts outside the U.S.; the OSI network would be quite helpful on this front.

Why Me?

This project will connect two very different communities: bloated, slow-moving bureaucracies, and free-wheeling open source Web developers. I have experience in both communities. After college, I spent over two years as an intelligence analyst with the Department of Defense. While there, I gained recognition for my ideas to revamp our computer systems. Since leaving in 2005, I've created two Web sites that encourage government transparency and citizen engagement. Throughout this time, the government has retained my services as a technology consultant. I am a member of what must be a very small group: former civil servants who now have close ties to the Web developer community. This background uniquely positions me to carry out my proposal.

My recent experience has also prepared me for this role. For the past 18 months, I have worked with a government office to create a customized Web application. This entailed working with the government office to create specifications; converting those specifications into a prototype; hiring programmers to make the prototype a reality; managing the project and testing the software as it was developed; and communicating with the government buyers throughout the development process to make sure my product was aligned with their needs.

The manner in which I've created this software is very similar to the method I outlined in the hypothetical New York scenario, and it is very different from the traditional manner in which private firms develop software. While this current project is not open source, I have in many ways already done a trial run of my proposed project.

Timeline

Over the coming months, I will identify a few organizations that would make ideal long-term supporters. When I choose one is based mostly on their response, but my decision will based on their financial commitment, marketability/reputation, and their reach in both the developer and government communities.

Throughout the next year, I will be promoting this idea at technology and government conferences, and writing about it online and ideally in some print publications. This will attract volunteer developers and government leaders interested in participating.

By this time next year, I hope to have joined an organization and identified at least one pilot project: a government champion with enthusiasm for the idea, and a problem meaningful enough to inspire programmers to volunteer their time toward the cause of solving it.