COMMUNITIES DRIVE OPEN SOURCE SOFTWARE

GUNNAR HELLEKSON AND GREG PRYZBY

US Government agencies are drowning in software. Much of it is old, or redundant, or unresponsive to its users' needs. An agency will procure a particular information system only to discover that another agency has already built the same system. Worse, the program is so large, and the contracting process so long, that the private sector has long since innovated away from the original requirements, saddling the agency's program with either antiquated software or an endlessly amended project plan. Agencies cannot buy their way out of this problem.

Even if a given program has the foresight to partner with another program to pool resources, deep and meaningful cooperation is limited by "color of money" issues, agency politics, and a security regime specifically designed to prevent cooperation. These limitations incent government software projects to rebuild the same capabilities many times over, at a very dear price, with no means of ensuring that one program's success or failure informs the other.

Despite these hurdles, a number of initiatives to encourage inter- and intra-agency collaboration have succeeded. Some of these successes have been inspired by the open source software (OSS) community, which collectively represents one of the most scalable and vigorous collaborative projects on earth. By successfully adopting open source development principles, these projects are now more viable, more responsive, and work more productively with their partners, both inside and outside agency walls. The open source development process presents an intriguing opportunity for these afflicted agencies, as well as entrepreneurs in the private sector who can support and facilitate this approach.

The tedium of open source licensing on government programs and the fitness of open source software for US Government agencies are well-trod topics, and will not be covered here. Suffice to say that OSS use is pervasive in civilian, DOD, and intelligence communities. As agencies and services have grown comfortable consuming OSS, and grow more sophisticated in their understanding of the open source model, they are better-positioned to fully exploit it.

Rather than treating open source software as only a low-cost alternative, these sophisticated users take advantage of the development model, and use it to solve much larger problems. The National Security Agency, for example, took its SELinux project to the open source community not only because it relieved the NSA of its support burden, but because the development model allowed good ideas from a much larger community of users to improve the software[i]. Once the software was open-sourced, that also permitted the code to be integrated into other mainstream open source projects, in this case the Linux® operating system, making SELinux ubiquitous.

A research project on operating system security has now become a feature in one of the most widely used operating systems in the world, with an absolute minimum of investment on the part of the NSA. Once the software became ubiquitous, vendors like Red Hat and Tresys began to support the project and the software, offering products, services, training, and support to end-users. By releasing SELinux to the world, the NSA was able to lower its internal support costs, drive widespread adoption of the technology, and ultimately make the SELinux project more valuable by creating a community around its use.

Creating a community around software is, in fact, what makes open source software work so well. Simply applying an open source software license to a piece of code will not make it more useful, more responsive, or more secure. An open source software license will, however, allow a community of developers to collect around a shared set of problems, and work together on the solution. In contrast to notional or symbolic efforts to improve collaboration in a particular community, open source is effective because it is organized around useful work.

There are many models for engaging a community, but there is broad consensus that a meritocracy is the most effective. The project leader or leaders exercise control over what changes are permitted, what capabilities will be provided, and the project's priorities. At the same time, the leadership is obligated to lower the barrier-to-entry for new contributors to the project, lest good ideas are missed. Decisions are made transparently, tools are well-understood, and in every other way the project is conducted such that an interested party can get involved with a minimum of effort. A project leadership who encourages contributions, welcomes new participants, and stays responsive to the needs of the community will ensure the ongoing viability of the project.

A broad community of users and developers collaborating together reduces the resource burden on each contributor, but also encourages pleasant surprises. The US Navy, for example, had a need for a deterministic computing platform for its shipboard command-and-control systems. Unlike the general-purpose commercial computing platforms already in place on the ship, these systems needed to operate within specific latency requirements: if a sailor pulled a trigger, he couldn't afford to wait while the computer performed another task before the weapon was fired. Weary of paying for expensive "real-time" operating systems, the Navy and a coalition of vendors worked to develop patches to the open source Linux operating system[ii]. For the Navy, these improvements meant that the hardware, operations, and maintenance would be identical for a real-time workload and a regular workload. It also meant that they now shared the ongoing maintenance burden with the broader Linux community.

Some in that broader community are companies that trade securities and commodities. They have a similar requirement for real-time systems. When someone places an order, they must execute the order in a predictable amount of time, not unlike the Navy's weapons systems. They shared the Navy's problem: reliance on expensive, proprietary software that locked them to a single vendor and required new training for their

i Frank Mayer, Karl MacMillan, and David Caplan, SELinux by Example: Using Security Enhanced Linux, 1st ed. (Prentice Hall, 2006), 12.

ii "Linux ready for real-time computing in financial services," http://searchenterpriselinux.techtarget.com/news/article/0,289142,sid39_gci1309650,00.html.

systems administrators. Fortunately, they were able to take the same software the Navy developed and apply it to their trading systems. Now they spend less on hardware and training, and have a broad portfolio of hardware and software they can use[iii]. Developers on Wall Street make improvements to the real-time software that the Navy can then use. Unwittingly, the Navy has drastically improved the operation of the financial markets, and the financial markets are improving the operation of the Navy's ships. This kind of unwitting collaboration would be untenable outside of the open source development process.

It is worth noting that this remarkable collaboration occurred without making drastic changes to policy or the Defense Acquisition Regulations. Nor did it require a Cooperative Research and Development Agreement. The Navy awarded task orders to companies, and those companies worked with the open source software community. Entrepreneurs will recognize the opportunity presented by leveraging the open source community to provide capabilities faster, and with a more viable long-term maintenance strategy than a vendor of proprietary software.

Collaboration and consensus are encouraged by the open source process, which is why open source software is an excellent vehicle for the development and adoption of standards. The world of government software, of course, is very much a world of standards. These standards facilitate interoperability not just between information systems, but between the organizations that use them.

The Hypertext Transfer Protocol (HTTP), for example, is the standard that defines the connective tissue between users of the world wide web. It is a technical standard, and also facilitates the interaction of billions of Internet users.

HTTP was developed and refined largely through open source software. Early tools were created based on early drafts of the standard[iv] and, through their use, the standard was improved to the robust HTTP 1.1 that has been in place since June of 1999[v]. Without open source implementations of the early drafts of this standard, it would be far more difficult to encourage widespread adoption. This broad adoption, in turn, improves the amount of testing and validation brought to bear on the standard. This is in stark contrast to closed or proprietary standards that do not as easily benefit from real-world use before they are adopted, and consequently find it difficult to ensure that the standard will be adopted when it is publicly released.

Government agencies interested in developing standards for interacting with other agencies would do well to insist on open source reference implementations of these standards. This would ensure the sound development of the standard, but would also provide easily consumable starting points for vendors who wish to adopt the standard. Rather than relying on a human interpretation of a highly technical standard, prospective vendors can start with software they can be sure will work properly. Also, the open and collaborative nature of open source reference implementations discourages the development of government-specific standards, which can be unnecessary or redundant to standards in the private sector.

The NHIN CONNECT project illustrates this approach. Healthcare Information Exchanges (HIEs) are being asked to comply with a bewildering number of standards for electronic healthcare records. They are also asked to work with the Department of Health and Human Services-funded National Health Information Network (NHIN), a kind of interstate highway for healthcare records. Given the number of use cases, the number of operative standards, and the nascent infrastructure for this task, HHS elected to conduct the CONNECT project as an open source software project[vi]. Software vendors, HIEs, and the government now

iii "Wall Street becoming Linux stronghold," http://www.networkworld.com/news/2008/061208-linux-wall-street.html.

iv "W3C Open Source Software," http://www.w3.org/Status.html.

v "Hypertext Transfer Protocol—HTTP/1.1," http://www.w3.org/Protocols/rfc2616/rfc2616.html.

vi "What is CONNECT? | CONNECT Community Portal - NHIN CONNECT Gateway," http://www.connectopensource.org/about/what-is-CONNECT.

collaborate on a reference implementation of this software. In so doing, HHS ensures that it has not inadvertently created a monopoly vendor of this critical infrastructure. HHS has also lowered the barrier-to-entry for new participants in the exchanges, as well as the barrier for new software vendors who wish to participate in the market. Finally, HHS has made it simple for third-party vendors to ensure compatibility with the NHIN, as they have a freely available reference implementation to test against.

So the open source software development process is much more than a means of lowering acquisition costs. It is also the means by which agencies can cooperate with each other and the public in real and meaningful ways, demonstrated by the ad hoc collaboration between the Navy and Wall Street on the real-time computing problem.

A larger community of developers and users creates more value for agencies, and more value for the private sector, shown by the NSA's SELinux project. SELinux could very well be withering in an academic journal, but is instead actively protecting millions of computers around the world from attack, and forms the basis for several commercial ventures that would not exist otherwise.

The open source development process also encourages standardization and consensus, as it has done in the NHIN CONNECT project. With an open, accepted, manifestly workable standard, it is now much easier for new companies to participate in the NHIN. In the past, this kind of cooperation was only possible by fiat, which would have been expensive, error-prone, and inefficient. Using the open source approach, the standards process itself becomes more viable and more valuable to the NHIN community.

The current Administration has put an emphasis on transparency, collaboration, and participation. Open source software is a physical manifestation of all these principles. It offers unique opportunities to both the public and the private sector—most especially when they work together.

RED HAT SALES AND INQUIRIES

RED HAT SALES AND INQUIRIES NORTH AMERICA
1-888-REDHAT1
www.redhat.com
www.redhat.com/government

GUNNAR HELLEKSON
Chief Technology Strategist
Red Hat US Public Sector
gunnar.hellekson@redhat.com

www.redhat.com
#2973527_0610