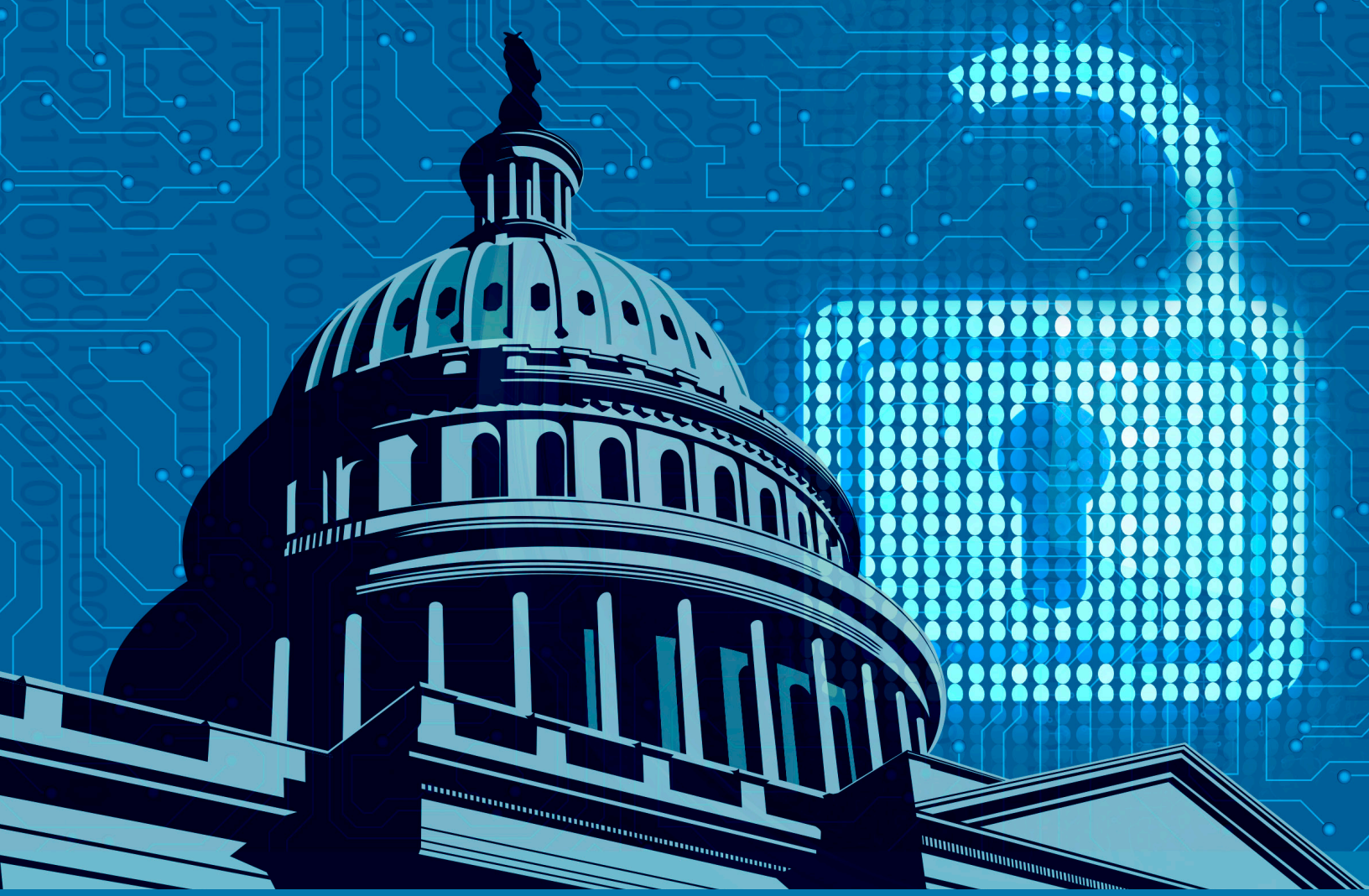


Open Source Software in Government

Challenges and Opportunities

August 2013

Homeland Security

Science and Technology

Open Source Software in Government: Challenges and Opportunities

Dr. David A. Wheeler, Institute for Defense Analyses (IDA)
Tom Dunn, Georgia Tech Research Institute (GTRI)

August 29, 2013

This document identifies key challenges and opportunities in the government application of Open Source Software (OSS), as reported in interviews of experts, suppliers, and potential users. There are many challenges to the collaborative development and use of such software in the government. To maximize the use of limited resources, the U.S. government must address these challenges, which can be grouped into categories such as: inertia, fears about low quality and malware, concerns about commercial support, procurement issues, and certification and accreditation (C&A) issues. Interviewees reported a critical need for OSS guidance and education. Specific interviewee recommendations included requiring that software and C&A materials developed with government funding be developed collaboratively and widely shared, that the government receive full data rights for such material, and that the government release such software as OSS by default.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Open Source Software (OSS) can be defined as "software for which the human-readable source code is available for use, study, reuse, modification, enhancement, and redistribution by the users of that software."[1] This "lessons learned" document identifies key challenges and opportunities in the government application of OSS, so that inappropriate roadblocks can be countered or mitigated. These challenges and opportunities were identified in interviews of OSS experts, suppliers, and potential users; users included both government contractors and government employees. These interviews were conducted and summarized as part of the Department of Homeland Security (DHS) Science and Technology (S&T) Directorate's Homeland Open Security Technology (HOST) project.

The interviewee comments highlighted the following themes and recommendations for the government (no priority is implied):

1. *Current OSS use and development in government*. Create and distribute success stories as "case studies," so that others can build on those experiences (e.g., as examples to replicate).

2. *Inertia*. Counter fear of change by publishing case studies and diminish fear of transition costs by increasing emphasis on modularity and standards.

3. *Fears about low quality or malware*. Ensure that the government and its contractors understand that OSS can assert higher quality in a more transparent way and that there are ways to evaluate OSS.

1 "Clarifying Guidance Regarding Open Source Software (OSS)," Department of Defense (DoD), Chief Information Officer (CIO), 2009. http://dodcio.defense.gov/Portals/0/Documents/FOSS/2009OSS.pdf




4. *Concerns about commercial support and warranties.* Ensure that more government employees and contractors are aware that there are many options for support and warranty of OSS, including self-support.

5. *Procurement.* Incentivize government program offices and contractors to build collaborative communities and to share code. Request for proposal developers should not presume that respondents have a particular business model and should not impose unnecessary paperwork burdens. The government should require sharing software and release software as OSS by default if it was developed with public funds; this may require changes to contracting strategies.

6. *Certification and Accreditation (C&A).* Refocus C&A efforts on risk management instead of implementing inflexible processes. Share C&A data and authorization to operate data where possible (e.g., by hosting a "summer of C&A" event). Ensure that all relevant parties, including OSS suppliers, are involved when the government is developing related specifications (e.g., Common Criteria Protection Profiles). The government should invest in security evaluations of key OSS.

7. *Standards/interoperability.* Switch from proprietary formats and protocols to modular systems and open standards, as this enables the transition to alternatives, including OSS. The government should take a more active role in developing these open standards and developing OSS implementations of them. One specific area noted was the Security Content Automation Protocol (SCAP) and Open Vulnerability and Assessment Language (OVAL) specifications, where there is a need for OSS tools and open supporting OVAL data.

8. *Challenges to the release of code from government.* Simplify processes to release software (including modifications) developed using government funding. Clarify that release does not obligate the government to support it or use it, and that identifying authors is acceptable. Speed the government's internal review processes, particularly for export control, and discourage unnecessarily creating new competing projects (aka "forks"). When releasing new OSS projects to the public, the government should use existing public-access commercial collaborative software development sites (aka "forges") whenever possible. The government needs to support broad participation, discovery, and active development.

9. *Need for guidance.* Create guidance on evaluating OSS. This should include guidance on the impact of OSS licenses (such as the General Public License (GPL), the most widely-used OSS license), for contributing back to the OSS community, and for releasing new government-funded projects as OSS.

10. *Need for education.* Provide education on OSS in general; on intellectual rights and OSS licenses (for both government employees and contractors); on government procurement (for potential suppliers); and on C&A.

We expected many challenges to relate to security or perceptions of security. We did find these, but many of the challenges were in other areas. Nevertheless, these "non-security" challenges can impede security, because they can impede the appropriate use or development of OSS programs that implement security features or are themselves more secure.

To maximize the use of limited resources, the U.S. government must address these challenges to reduce the unnecessary barriers to the use and development of OSS. Many of these challenges can be addressed by promulgating education and guidance on OSS for different roles. The U.S. government should also transition to increased transparency and openness. In addition, many interviewees stressed that contracts should require that software and C&A materials developed with government funding be developed collaboratively and widely shared, provide full data rights to the government (unless it can be justified that fewer rights benefit the government as a whole), and be released as OSS by default.




INTRODUCTION

Open Source Software (OSS) is "software for which the human-readable source code is available for use, study, reuse, modification, enhancement, and redistribution by the users of that software."[2] OSS can provide advantages in developing and updating software-based capabilities, anticipating new threats, responding to continuously changing requirements, and supporting software reliability and security efforts. OSS continues to become more commonplace in the commercial marketplace. Yet government's use of OSS, and commercial products developed using OSS, are hindered by existing government policies and practices.

This document identifies key challenges and opportunities in the government application of OSS so that inappropriate roadblocks can be countered or mitigated. These challenges and opportunities were identified in interviews with experts, suppliers, and potential users, where users include both government contractors and government employees. These interviews were conducted in 2011 as part of the Department of Homeland Security (DHS) Science and Technology Directorate's (S&T) Homeland Open Security Technology (HOST) project <http://www.dhs.gov/csd-host>.

Issue and Approach

While there are reports of difficulties applying OSS in government[3],[4], S&T had questions about what the real problems actually were. Thus, the HOST project performed interviews to help determine what the real challenges were, as well as to identify approaches for resolving them.

As a result, Dr. David A. Wheeler (Institute for Defense Analyses) and Tom Dunn (Georgia Tech Research Institute) interviewed 31 people who were (1) OSS experts, (2) suppliers (especially non-government OSS suppliers), or (3) potential users (aka the demand side). The potential users included both government contractors and government employees (military and non-military, federal and non-federal). Many people could be placed in more than one category or had significant experience in more than one category, but for the purposes of Table 1 each person is assigned exactly one category. This paper focuses on the federal government (rather than state, local, or tribal governments), though there were some interviews to identify state and local government issues. Table 1 shows the breakdown of interviewees by category.

2 "Clarifying Guidance Regarding Open Source Software (OSS)," DoD CIO, 2009. http://dodcio.defense.gov/Portals/0/Documents/FOSS/2009OSS.pdf

3 "Clarifying Guidance Regarding Open Source Software (OSS)," DoD CIO, 2009, says that "there have been misconceptions and misinterpretations … that have hampered effective DoD use and development of OSS."

4 "Open Source Software (OSS) in U.S. Government Acquisitions," David A. Wheeler, *Journal of Software Technology*, June 2007, https://www.thecsiac.com/journal/open-source says that "Applying OSS can sometimes be a challenge in U.S. government acquisitions."

Table 1. Interviewees

CATEGORY	NUMBER
OSS experts	7
OSS suppliers	7
Contractors/integrators	5
Government employees	12
TOTAL	**31**

These interviews were conducted so as to avoid prejudging what the issues were. The interviewers worked to find out from the interviewees what the real problems were, for example, by asking only a few broad questions instead of asking many detailed questions. When interviewees answered the broad questions, the interviews became a free-form conversation in which the interviewers probed the interviewees to gain a greater understanding. Nearly all interviews took more than hour. Determining what the real problems were was felt to be more important than attempting to create a random sample (with fixed narrow questions) and conducting quantitative surveys.

The interviews were not for attribution (unless the interviewee specifically agreed otherwise), so that interviewees could speak their minds without fear of retribution. Many interviewees, particularly potential users, agreed to be interviewed only because they knew that these interviews would be published without attribution.

Document Organization

The interviews were analyzed to identify major themes, and those themes were used to organize this report. The themes are: current OSS use and development in government; inertia; fears about low quality or malware in OSS; concerns about commercial support and warranties; procurement; certification and accreditation (C&A); standards/interoperability; challenges to release of code from government; need for guidance; and need for education. These themes are subdivided further. Headings beginning with the word "Solution" discuss solutions proposed by interviewees. This paper ends with conclusions.

This document summarizes and presents the opinions of various interviewees. Views and opinions expressed do not necessarily reflect those of DHS. Also, it is unlikely that all interviewees would agree on every point; what is shown here is the merging of different interviewees' comments. Quotations are included throughout to illustrate some of the issues in the interviewees' own words. Non-heading text is italicized to emphasize where we believe an especially interesting point is made.

This document presents only a small subset of material that (in the authors' opinion) is especially important, extracted from the large amount of data from the interviews.

CURRENT OSS USE AND DEVELOPMENT IN GOVERNMENT

OSS is being used in government, as well as being released by the government (as both minor improvements and whole new projects), and the government is receiving benefits from doing so. However, many in government are unaware of this. An OSS supplier said, "Government is using a lot of OSS, but may not be completely aware of OSS that's out there …" while a contractor said, "Much of [one agency] follows a 'don't ask, don't tell' policy for open source software."

Many interviewees expected that OSS use will increase as budgets shrink. One government employee said, "OSS will succeed due to cost; the government is bankrupt and will have to cut [the] budget … we keep re-inventing software every time we renew a contract … money is the driving issue." An OSS expert said, "The downturn in the economy is going to require the government save … OSS can be part of that process, though it is not a silver bullet."

Several interviewees suggested that success stories should be distributed as "case studies," so that others could build on their experiences (e.g., as examples to replicate). One interviewee said, "The more OSS is used, and the more success stories are out there, the more people will use it and truly save time and money." Another said, "[Get] people using OSS to share their experiences with their peers, that's more trustworthy than if it comes from a vendor. Formalize it with case studies. I'm a big fan of case studies …" The HOST program is developing case studies so that others can build on previous experience.

INERTIA

Although OSS is used and developed in the government, there are many problems in doing so. One of the key challenges widely noted was inertia—that is, resistance to change.

Fear of change

Many interviewees felt that changes are perceived as risky, whether they are or not, and many are afraid of changes. A government employee said that the government's "rejection of OSS is a combination of fear and inertia. [They] don't like to move outside of their comfort zone, and [there's] the fear of the unknown." An OSS expert stated, "For OSS in the government, the biggest impediment is habit; they're used to buying what they've bought before." Another government employee stated, "There's a lot of risk aversion." A different OSS expert noted that "[OSS is] generally mature, but in the last 10 years it has been viewed more as cutting edge. It represents a change in the business model, and government entities are adverse to risk. Any change is viewed as a risk." *Publishing case studies and making it easier for government employees to contact other government employees who have done something similar may help reduce these fears.*

High transition costs inhibit switching to anything else, including OSS

Interviewees felt that transitioning from existing systems to any other system (OSS or otherwise) is often very costly. One OSS supplier stated, "*It rarely makes sense to convert from one tool to another* … [It's difficult to change] and hard to weigh the costs of the change." Thus, the supplier focused on selling to organizations starting a new project "rather than to grow [via] an existing project." It can be impractical to try to convert government users from one tool to another, even if it would save money in the long run. An OSS expert said, "*You have the 'movement problem.'* You need to be opportunistic to move to OSS or closed source. Moving people from one platform to another is hard."

Incumbent products are rarely disputed

Interviewees felt that once the government begins using some commercial off-the-shelf (COTS) software, additional copies and maintenance are often purchased using sole source requests that are rarely disputed—even if the potential transition costs are low or the transition benefit might be high. This makes it difficult for any product to compete, OSS or not, regardless of how effective it is.

One interviewee noted, "Most companies [and government] over-buy and get large enterprise license agreements with proprietary [software] stacks. [They] end up with access to a whole stack with lots of stuff. [Once they have these large, expensive stacks, their] mindset is that if they already own it, there's no cost." Enterprise licenses are intended to save money, but if people think of enterprise licenses as "free" they may fail to consider lower-cost alternatives, resulting in higher overall costs to the government.

Lack of government software expertise

Several interviewees felt that the government lacked software expertise, resulting in many agency organizations being effectively run by contractors. As a military officer stated, "Most [government] organizations don't have software experts, they rely on a vendor … to supply expertise … Real understanding is all on the vendor side. Vendors want to continue that relationship since that's how they get their money." A government employee explained, "The government got rid of [computer programmers] … [We] purged our technical knowledge expecting the private sector to pick up the slack, which they did, [but their interests are not always aligned with the government's]. That puts the government at a severe disadvantage. Now we have senior managers who don't understand software technology. The managers default to what they know, hiring more contractors and layering on bureaucracy." Another interviewee said, "Government employees are surrounded

by these contractors. [Vivek] Kundra called these the 'IT Cartel.'[5],[6] That's your biggest challenge; it's not that technology doesn't work or that it can't be contractually procured. [The problem is that] you have a web [of interdependency] that's grown up over time …" This lack of government software expertise makes it more difficult to adopt newer approaches like OSS.

People don't know or ignore current OSS policies

The government does have some OSS policies. The Office of Management and Budget (OMB) has released a federal government-wide memo that acquisition rules apply to "all software, whether it is proprietary or Open Source Software,"[7] placing OSS on an equal footing. The Department of Defense (DoD) has a more detailed OSS policy[8] that makes it clear that OSS is acceptable and must be considered, as well as supporting frequently asked questions (FAQs)[9] and best practices[10] documents. Such policies and supporting material can help, but they are sometimes not enough. Many contractors and government employees did not understand laws and policies regarding OSS. For example, a government employee stated "Federal departments are not in touch with the power already in their hands with existing policy. They are waiting for some legislation or executive order, and are not willing to stick their neck out." Another government employee concurred, noting "There is no barrier to the use and development of OSS or public domain software." Also, the interviewers encountered pervasive use of the term "commercial software" as an antonym of OSS. Yet U.S. law defines commercial software in a way that includes most OSS.[11],[12]

Reports of government employees resisting existing policies are not uncommon. As an example, one OSS supplier reported, "You have a well-established policy that says don't be stupid with OSS, but it is still common for individual people to ignore that written policy … The fundamental problem is open antagonism … The policies are already crystal clear."

5 "Tight Budget? Look to the 'Cloud'," Vivek Kundra, *The New York Times*, 30 August 2011, <http://www.nytimes.com/2011/08/31/opinion/tight-budget-look-to-the-cloud.html?_r=1>.

6 "Outgoing federal CIO warns of 'an IT cartel'," Patrick Thibodeau, *ComputerWorld*, 18 July 2011, <http://www.computerworld.com/s/article/9218466/Outgoing_federal_CIO_warns_of_an_IT_cartel>.

7 "Software Acquisition," OMB, 1 July 2004, M-04-16, <http://www.whitehouse.gov/omb/memoranda_fy04_m04-16>.

8 "Clarifying Guidance Regarding Open Source Software (OSS)," DoD CIO, 2009, <http://dodcio.defense.gov/Portals/0/Documents/FOSS/2009OSS.pdf>.

9 "DoD Open Source Software Frequently Asked Questions (FAQ)," DoD CIO, <http://dodcio.defense.gov/OpenSourceSoftwareFAQ.aspx>.

10 "Open Technology Development (OTD): Lessons Learned & Best Practices for Military Software," DoD CIO and Acquisition, Technology, and Logistics, <http://dodcio.defense.gov/Portals/0/Documents/FOSS/OTD-lessons-learned-military-signed.pdf>.

11 See section 403 of title 41 of the U.S. code. By definition, anything (other than real estate) is commercial if it is (1) customarily used for non-governmental purposes, and (2) has been sold, leased, or licensed to the general public.

12 "Open Source Software Is Commercial," *Journal of Software Technology*, David A. Wheeler, February 2011, Vol. 14, Number 1, <https://www.thecsiac.com/journal/dod-and-open-source-software>.

The same OSS supplier provided an example of the problem: "[One] person just didn't like OSS for ideological reasons and was openly hostile. Policies are used as weapons in office politics to compete with each other … you can't change attitudes, but you can take away their weapons … The actual written policy had no effect. It took … active intervention [by] a living, breathing, talking person [who] spoke from a position of authority … *Maybe [we need] an ombudsman who could intervene* when [people are] misinterpreting policy … It's very frustrating to see clear written policy and run into folks who ignore it."

FEARS ABOUT LOW QUALITY OR MALWARE IN OSS

Interviewees agreed that there was not an increased risk of low quality and malware in OSS compared to proprietary software.[13] However, interviewees reported encountering the belief that it is easier to insert low quality or malicious code into OSS. One government employee said, "[Many] worry that since so many people have access to code (and anyone can contribute), there must be a lack of control process over who can put stuff in the code. [This is] usually people not familiar with how it's developed …" Particular programs need to be evaluated on their own merits.

Software can have low or high quality, regardless of whether it is OSS or not. As one contractor put it, "*There can be good and bad OSS, but there are metrics to figure out which is which.*" Another contractor said "*[OSS] can assert higher quality in a more transparent way*; they can show their source code is quite solid through the use of software quality assurance tools … You can do an automated verification that you conform to government-style policies, let alone other policies, so that source code is less likely to have memory issues, or whatever the case may be. Being able to assert those quality metrics would have a lot of value … *[With OSS you're] able to assert in a transparent way that you're producing quality code*, especially when securing people's data."[14]

Several interviewees praised OSS's supply chain transparency as it enables countering risks. One contractor stated, "You can get more trust in OSS origins than proprietary software."

13 *Coverity Scan 2011 Open Source Integrity Report*, Coverity, <http://softwareintegrity.coverity.com/coverity-scan-2011-open-source-integrity-report-registration.html>, reported that "Coverity Scan [is] the largest public-private sector research project in the world focused on open source integrity, originally initiated in 2006 with the U.S. Department of Homeland Security … Based upon the sample of active projects in Coverity Scan, we found the quality of open source software is above average … The average defect density, or the number of defects per thousand lines of code, across the top 45 active open source projects in Scan is .45. Coverity's experience working with commercial software development projects has shown that a good benchmark for high quality software is an average defect density of equal to or less than 1.0."

14 *Software Security Assurance: A State-of-the-Art Report (SOAR)*, Goertzel et al., 31 July 2007, <http://iac.dtic.mil/iatac/download/security.pdf>, describes the "state of the art" in software security assurance as of the time it was written.

A government employee stated, "There is a concern over the ease of getting malware into OSS. Actually, it's pretty easy to get malware into proprietary software too. [OSS is unique in that it gives complete visibility into the supply chain.]" Another government employee said, "*Just because you cannot [review] the source [of proprietary software] does not mean the software is safe … I would rather know where it came from so I know what to target in my evaluation.*" Yet another government employee stated, "OSS typically has better configuration control, so the argument [that OSS has no configuration control] makes no sense … it's just not true."

In short, interviewees stated that the *perception* that OSS always has an increased risk of low quality or malware has inhibited the use of OSS.

CONCERNS ABOUT COMMERCIAL SUPPORT AND WARRANTIES

There were many comments about commercial support and warranties, though the specific issues differed. *Some organizations wanted commercial support and incorrectly believed that this is never available for OSS.* One interviewee explained that there is a "perception that [OSS] will not have any support or anyone to call." Other organizations wanted commercial support but found it more difficult to determine who could provide commercial support for OSS programs or determined that there was no commercial support for a particular OSS program. Another interviewee said it is often "not hard to find someone to support OSS, but it is not as easy as with commercial [proprietary] software that comes with support built in. Having people understand the business model is the problem."

A fundamental challenge is that with OSS, acquiring the software is typically separate from acquiring support. Many businesses give away OSS and make money by selling services (e.g., support and improvements); *this provides benefits to customers, such as the ability to evaluate the software before buying services and the flexibility to change support providers.* This separation is different from many proprietary programs, where acquiring the software and support is a single bundled purchase.

Yet other organizations had a local policy of *always* requiring commercial support, even when the organization could self-support. This local policy:

- Prevented appropriate use of OSS if no commercial support could be found. One interviewee said that one military service "policy implies there needs to be contractual support from the vendor, [and there are] similar requirements from other … commands. [But the] vendor may be a loose knit federation of developers, so contractual support may be impossible."

- Wasted money on commercial OSS support when it was available but known to be unnecessary. One interviewee stated, "In one case we were required to purchase a support contract for $30K [by management, even though there was no expectation that it was needed]."

Some government organizations also impose additional requirements on commercial support (OSS or not) that can greatly increase support costs. One government employee revealed that, by policy, anyone in their organization "who has IT knowledge of your system [configuration or bugs] has to be a U.S. citizen on U.S. soil." Yet many companies outsource IT support, so as a result of this policy, in at least one case, support costs increased by 3 to 8 times.

Several commented that the "warranties" provided by many proprietary software developers were essentially worthless. For example, one interviewee said, "Just look at the … end user license agreement [of a widely-used proprietary software product]. It doesn't have liability on fitness of use, etc. … [it's] sold without liability or recourse. And that's the model for most software." Yet the presence of these warranties caused some to believe that proprietary software is better than OSS.

PROCUREMENT

Wrong incentives within government and contractors

Increased reuse and collaborative development of software is in the interest of the government as a whole, as it reduces costs (by eliminating duplicative efforts) and can increase quality (through increased review).

Yet government program offices are disincentivized to reuse and collaboratively develop software, including OSS, and are instead incentivized to create local "fiefdoms" that control potentially sharable components. One government employee said, "We keep re-inventing software every time we renew a contract. Each program manager has his own kingdom to invent his own software and doesn't have to reuse stuff from other kingdoms." Another interviewee said that organizations "are quite threatened by the potential of open source anything because it will infringe on headcount in the program management organization. This threatens people's status, headcount, [and] rank." Another said, "It's all these layers of middle management … their success is measured by the size of their budget, number of people, etc.—how much they control. [They have] no incentive to reduce costs." Yet another interviewee noted, "Even government-created code that is distributed within the government is often not distributed with the source code. This is intentional. It gives the sponsor control over it, so they maintain funding and control, [but users can't depend on it because they may stop development/support]. In short, OSS can threaten the status and power of managers, as some see their own status measured primarily through budget

sizes and manpower counts. *We need clear interagency guidance to [require government organizations to] distribute software source code [to other government agencies].*"

Contractors are also disincentivized to reuse and collaboratively develop software. Contractors are incentivized by profit to gain follow-on business. A contractor-exclusive solution, even where it is not required, increases the likelihood of follow-on business—and thus they are incentivized to propose them. As one government employee said, "Contractors do not want to share with each other … they see that as a detriment; it affects follow-on contract likelihood." In addition, contractors are disincentivized from using COTS components, including OSS, and from co-developing OSS. One contractor noted that "for every COTS product [a contractor buys or uses, the contractor only gets] to add anywhere from four to seven percent," but when they develop a new component, a contractor gets to "charge a whole lot more … what's the incentive to buy or reuse stuff?" The Federal Acquisition Regulations (FAR) is intended to "promote competition in the acquisition process,"[15] but when the government permits exclusive components in systems where other options are available, competition is impeded. Contractors are incentivized to insert these exclusive components.

One government employee pointed out that the "industrial base has the same concerns as the government acquisition community. If I [as a contractor] use open source, then I can't charge as much, more people go on overhead, so it threatens my rate structure. This is an influence on power, real or perceived self importance, and functions that you do, so [managers] will be against it."

A government official observed, "It's an interesting problem—how do we change contracts to incentivize sharing? *We need to change incentives to foster building a collaborative community and share code.* I'm not sure we know how to do that. We could explicitly require past performance on how forthcoming they are on data rights and sharing code."

Difficult to sustain investment in infrastructure or OSS

Government procurement is experienced in procuring (including the development of) specific systems that they need. Yet some software can be easily repurposed, allowing many agencies to pool their resources. Examples of such software include development tools, community management tools, security tools, and security libraries. Unfortunately, it is often more difficult to sustain continued investment in software that is used by many users that are outside of a particular funding agency.

One OSS expert said, "Usually we fund things we use ourselves, rather than funding things that will be used by lots of people. *Things like*

15 FAR 1.102-2(a)(5).

OpenSSL[16] should be funded by everyone … [We] need to figure out the funding model to be used by everyone in government. Who has the mission to make tools better for the entire nation? Who has the budget?"

Acquisition process doesn't match typical OSS business model

Acquisition is defined as "acquiring by contract with appropriated funds of supplies or services (including construction) … through purchase or lease …"[17] The acquisition process often includes the release of a request for proposal (RFP), in which suppliers are invited (often by bidding) to submit a proposal for a specific commodity or service.

There were many complaints of the difficulty of even responding to an RFP; RFPs are often designed presuming a particular business model, and thus make it difficult to respond when there is a different one. For example, commercial OSS suppliers often sell subscriptions to services as their business model, but this can be a poor match with government acquisition processes. One OSS supplier noted that their OSS "software itself is free; you pay for support, consulting, [and] helping to implement [it]. A lot of OSS businesses are set up this way. [Yet the government acquisition process] is set up for proprietary licenses. It would take more time, material, and resources to figure out how to respond to the RFPs." Another COTS OSS supplier noted that "many times the procurement is broken up into time and materials or is a fixed-price task order for 30 or 60 days. You can put [support costs] in as 'other direct cost,' but often that isn't one of the options on the RFP. [The RFP] doesn't break out into time and materials." One interviewee said, "OSS may be a phenomenal fit [for a particular government need], but the RFP template is set up for a proprietary product."

The government often puts out bids that require support staff to be on site, even when there are other options, with the result that potential solutions are not considered. One OSS supplier lamented, "We don't do staff augmentation. Often the government wants people to come in and the government gives them a desk. We have our people here that they can call 24x7, but because there's such a limited amount of expertise in the field, our guys are the best. We can't have those people be on site and be a staff augmentation; there [is] much more value serving a lot of customers."

Procurement paperwork impedes small businesses

Many OSS suppliers are small businesses. It's very difficult for a small business to sell to the government due to the massive paperwork burdens that the government imposes. This has the unintentional effect of shutting out many OSS suppliers:

- One OSS supplier said, "The government artificially inflates the cost of software, with *unnecessary … hoops* to jump through and creating the need for middlemen where you don't need any." He clarified this by saying, "We have a number of

16 OpenSSL is an OSS cryptographic toolkit widely used in industry and government. For more information, see <http://www.openssl.org/>.

17 FAR 2.101.

users of OSS software in the government, but few customers. When we helped a government agency who wanted the support, they put out a lengthy competitive bid RFI [request for information]/RFP that was incredibly laborious to comply with for a small business, especially for software they were already using (the OSS version). Open competition for an OSS product they were already using resulted in costs they didn't need to spend. When we get government requests that include a lengthy paperwork drill, we send it to [an] integrator who adds their markup."

- A different supplier said, "[In some cases] you get the full force and fury of the contracting process. *The [government] paperwork burden is [huge]* … A small company can't really sell directly to the government; you have to go through a prime contractor with all the disadvantages that implies … Many contracts have more to it than the formal process … The barrier to entry is so huge for a small company … "
- A third supplier said, "GSA [General Services Administration], in terms of gating, the complexity of the ecosystem, and maintaining/understanding the distributor/retailer channel, is an obstacle. For a two- or three-person shop, it is impossible. You can't do that and still run a business. This is a small business issue. All problems other than [OSS-specific] FUD [fear, uncertainty, and doubt] are common among any small proprietor."

OSS does not cost enough

While OSS is certainly not "free" in the monetary sense, it can often be a bargain. OSS development methods perform cost sharing between developers and typically cost little or nothing to download. There are certainly other costs, such as those for installation, training, and support, but those can often be competed. Even if an organization needs to make a change, it typically only needs to pay for the changes that it needs. *If the change is then accepted back into the main OSS project, future maintenance of the software itself is mostly or completely paid for by others.*

OSS sometimes costs dramatically less than alternatives, causing acquirers to sometimes refuse to investigate it. A government employee stated, "In contracting, we tend to throw out [the] most expensive and least expensive and only deal with folks in the middle," and thus would ignore a significantly lower-priced approach (including OSS). Others believe that it is simply not possible for a product to meet their needs at lower prices than they are used to, or that such products will inherently have lower quality. A different government employee explained that OSS rejection may "be a cultural thing of 'you get what you pay for.' If you aren't spending millions of dollars, [others believe] you aren't being serious about the problem … We've heard that [OSS doesn't cost enough]."

Concerns about the GNU General Public License

The GNU General Public License (GPL),[18] the most common OSS license, requires that under certain conditions source code must be released to the recipients of executable code. Many organizations were very concerned about the potential impact of the GPL. One interviewee said, "[The] government is concerned with being forced to release code under GPL … If you're developing firewall code, you wouldn't want to release it to let everyone figure out how to penetrate it." Other interviewees said some organizations inhibit source code release to maintain their exclusive funding and control, even if such withholding is detrimental to the government as a whole (see the section "Wrong incentives within government and contractors").

It was clear that some people do not understand the GPL and thus reject the use of any GPL-licensed software, even when it would be perfectly fine for their purposes. For example, it is widely believed that the GPL requires that all changes be released to the public, yet this is simply not true. In reality, the GPL only requires that if an executable is distributed[19] to someone outside their organization, that organization must offer the recipient the source code as well.

Nevertheless, many government organizations use GPL'ed software. One interviewee stated, "Lots of DoD agencies use [GPL'ed] tools to develop code." In addition, the government has already released software under the GPL. One government employee pointed out the high-profile example where the White House uses Drupal and has released the TechStat toolkit and IT Dashboard software, all of which are under the GPL.

Requirements inflexibility

A general problem with government procurement is its inflexibility on requirements. The government, particularly its contracting officers, often expect to have their requirements specification (that they wrote or paid to develop) met exactly, no matter the cost. If they can't find a commercial item that is an exact fit, the government often ends up building a brand new system.

Yet, if the government were willing to accept an "80 percent solution"—that is, an application that met most (not all) of the requirements"—the government would often be able to use COTS solutions (including OSS) at a tiny fraction of the price and be able to use these solutions immediately. One contractor explained that historically, "[The agency we support] has been able to write 'we need X to do Y' and someone will build to it. Instead of using COTS for the 80 percent solution, then

18 GNU GPL, Free Software Foundation. At the time of writing this document, the current version is GPL version 3, 29 June 2007, <http://www.gnu.org/licenses/gpl.html>. GPL version 2, June 1991, is also widely used and is available at <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>. GNU stands for "GNU's not Unix."

19 GPL version 2 uses the term "distribute" whereas version 3 uses the term "convey."

realizing what you really need after using it for a year, [the agency will] build a brand new system."

The same interviewee stated that similarly, one agency's "contracting officers haven't evolved to how they support" maintaining or modifying OSS "to meet the 100 percent … [even though] it's a lot easier to find an 80 percent solution [and improve it] than to build a 100 percent solution from scratch (eight to ten years to get something to the users, but by the time they get it, it's dated)." This is unfortunate because, by definition, OSS provides the rights to modify and redistribute the software.

A related challenge is that the government sometimes has difficulty procuring modular systems. Again, government procurement typically focuses on a list of requirements, and procurers expect a simple yes-or-no answer to "Does your product do X?" Procurers should instead ask, "What is required for your product to do X?" One interviewee explained that OSS is "modular like Lego blocks. You can add on additional pieces for free, but some assembly [may be] required. It's hard when people ask, 'Can it do this?' [We can then answer,] 'Well, yes if you install the relevant module.' Then they think you need customization [even when you don't]."

Section 508 accessibility

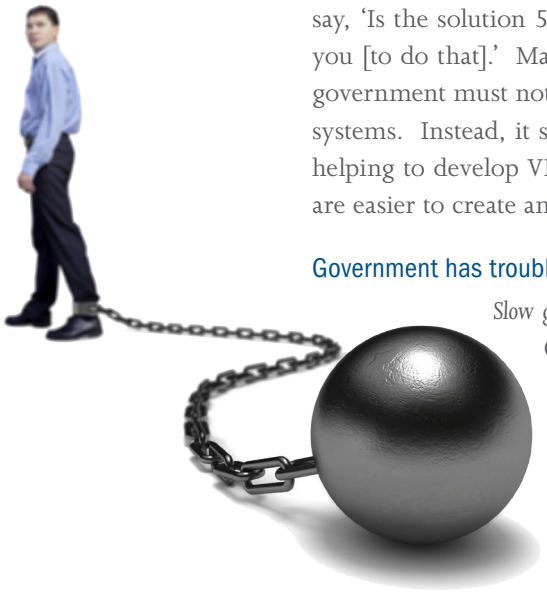
Section 508 (29 U.S.C. 794d) requires that federal executive agencies give disabled employees and members of the public access to information that is comparable to the access available to others. Agencies must do this when developing, procuring, maintaining, or using electronic and information technology (EIT). It is the government (not suppliers) who must comply with section 508; however, a supplier who wishes to sell EIT products or services to the government must design and manufacture them so that they meet the applicable section 508 provisions. There are some exceptions (e.g., national security systems and when it would impose an undue burden).[20]

Several interviewees noted that section 508 accessibility requirements (for individuals with disabilities), including paperwork, were often challenges to the use of OSS. One government web application developer stated, "There's not a good answer for the disability assistance within web browsers. Mostly, it's a function of the operating system, but many have not done the paperwork you have to do … If

20 Additional information on section 508 is available at <http://www.section508.gov>; relevant standards are available at <http://www.access-board.gov/guidelines-and-standards/communications-and-it>.

the VPAT [Voluntary Product Accessibility Template®][21] was put in for that, and it was integrated with the web browsers, it would go a long way."

A supplier noted that section 508 is not something "achieved out-of-the-box … it's not a destination, it's more of a … process that you adhere to as you go … [yet] many RFPs say, 'Is the solution 508 compliant?' [The answer would be,] 'Yes it can [be,] but it's on you [to do that].' Many times they don't understand what their requirements are." The government must not expect that some piece of software automatically creates accessible systems. Instead, it should invest in clearly documenting accessibility information (e.g., helping to develop VPAT information for some OSS programs) so that accessible systems are easier to create and justify.



Government has trouble keeping up with COTS/OSS development speed

Slow government review and deployment processes can inhibit the use of COTS, including OSS. As an OSS expert observed, "The OSS development model is based on a very fast evolutionary cycle that relies on the developer and code being very agile and nimble. In commercial and non-governmental [acquisitions], that is viewed as an asset or benefit. In government settings there are additional requirements (in security, IA [information assurance], [and] acquisition) that require an extended time period/process of review and consideration. That [faster development speed of OSS] really does not necessarily endear [OSS to those who must ensure these additional requirements are met] … OSS still moves at its own brisk pace, and that's always going to be a challenge … for government adoption. The government is not set up to evolve as fast as the OSS model."

Government users must often spend significant time filling out paperwork before they are allowed to use each new version release, as well as re-performing tasks such as code scanning and remediation. *This paperwork process can be especially difficult in major version upgrades or when a program's name changes, and name changes are not uncommon in OSS.* The result is that the government often does not upgrade its software; one interviewee said, "Government users are left in the dust and cannot commit to a major version upgrade." One interviewee said that the difficulty in upgrading "[makes sense for proprietary software because] when you pay for a version upgrade it is important, but with OSS it … [typically] costs you nothing. [Just] do the upgrade, [and] if it breaks, roll it back. *[We] should not need to resubmit a full paperwork package just because the version number changed.*" A military officer said, "We can only

21 The government may request accessibility information from suppliers, and it may request that this information be put in a format such as the VPAT. The VPAT format is defined in the *Voluntary Product Accessibility Template (VPAT)®*, Information Technology Industry Council, <http://www.itic.org/public-policy/accessibility>. The VPAT states that it is used to "document a product's conformance with the accessibility standards under Section 508 of the Rehabilitation Act" and that its purpose is to "assist Federal contracting officials and other buyers in making preliminary assessments regarding the availability of commercial 'Electronic and Information Technology' products and services with features that support accessibility."

use [some systems with built-in software] unofficially. To make it a program of record, we need to add a year and at least a million dollars. To change it requires another year and one million dollars."

In one case, an interviewee noted a program manager (PM) who was angry when an upgraded OSS library provided additional capabilities, at no charge, because the PM had not authorized those changes. Many PMs are familiar with managing development of nearly all software in-house, and thus expect to authorize every change to any software. This mindset does not work well with COTS software, including OSS, because changes to COTS components are funded and made outside the government organization.

One OSS supplier noted that web browsers in the government were a "very interesting case" be-cause the government has "everything from [the] latest to ancient technology that nobody supports anymore. Not so much the web browser brand, but also release levels … [The government doesn't] keep them current. It makes it harder to update our product."

SOLUTION: Require in contracts that contractors share and provide full rights in software they develop

The government often does not receive unlimited data rights in software it pays to develop. This can prevent meaningful competition of any contract that involves that software, as well as inhibiting the release of the software (or portions of it) as OSS. One solution would be to ensure that contracts *require* that the government receives unlimited rights for all software it pays to develop, and that the government receives the source code. As one government employee said, "If you require it, the vendors will do it. There is a requirement to share all data in the DoD today. Source code is data."

Another government employee said, "We [the government] could definitely do a better job on contract data rights. Standard [DoD contracting clauses give] the government pretty good rights, but contractors [include] IRAD [internal research and development], etc., to give government purpose rights, instead of unlimited rights. We need to understand what data rights we have, so that we can contribute back to the OSS community."

The default contract clauses in the FAR and DoD FAR Supplement provide the government with many rights. However, these can be negotiated away, enabling the government to lose rights to software that the public paid to develop. Note that "unlimited rights" gives the government all the copyright-related rights it needs to release that software as OSS.

SOLUTION: Release government-funded software as OSS by default

We repeatedly heard from interviewees that software developed using public (government) funds should be released as OSS by default, unless there was an approved justification to do otherwise. Indeed, we were surprised how often we heard this. For example, one government employee said that "standard guidance [should state an] intent to release things developed with public funds under an OSS license, except when we have a specific reason why we don't" and called this "*default to open*." Another government employee said that the government should have to "justify *not* releasing the code [developed with public funds] instead of the reverse."[22] Either the government or the contractor could accomplish releasing software as OSS.

One interviewee stated this would *require the government to stop thinking about software as a product or service, and instead view software as a resource around which individuals (and companies) work.* "You can [then] hire a maintenance contractor to maintain it, another contractor to add a feature or two … by making software a resource to all, that agency can use improvements by all."

The government's general procurement policy is to enable competition; releasing government-funded software as OSS would make it much easier to have different potential developers compete to add additional capabilities. This would also make it easier for other government agencies to find existing government-funded software (because there are commercial services for finding OSS).

CERTIFICATION AND ACCREDITATION

Certification may be defined as the "comprehensive evaluation of the technical and non-technical security safeguards of an information system to support the accreditation process that establishes the extent to which a particular design and implementation meets a set of specified security requirements."[23] Similarly, accreditation may be defined as the "formal declaration by a Designated Accrediting Authority (DAA) or Principal Accrediting Authority (PAA) that an information system is approved to operate at an acceptable level of risk, based on the implementation of an approved set of technical, managerial, and procedural safeguards."[24] Interviewees discussed many OSS issues related to the government's C&A processes.

---

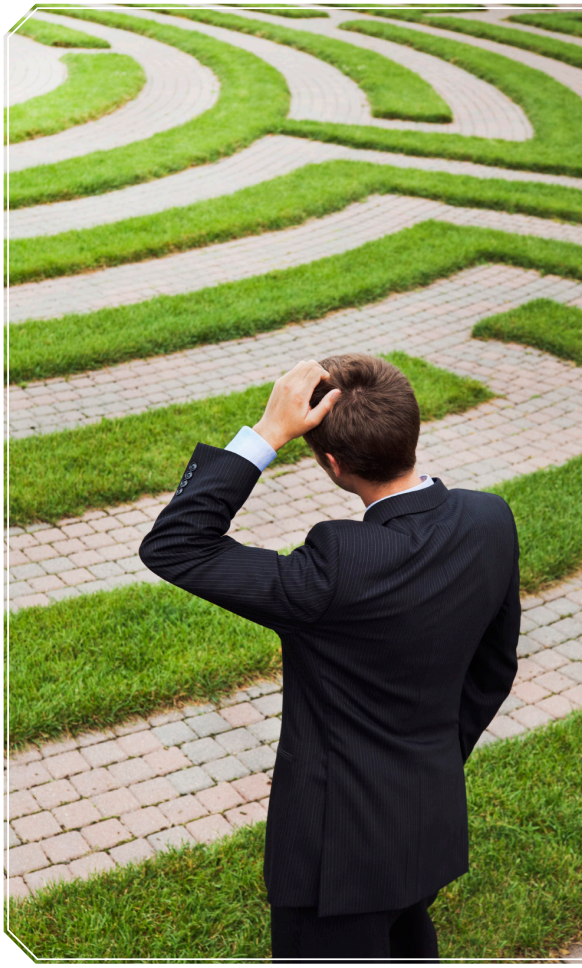
22 The U.S. Federal Government Consumer Financial Protection Bureau, as stated in its 2012 "Source Code Policy" <http://www.consumerfinance.gov/developers/sourcecodepolicy/>, already requires that software source code written by its staff or its contractors is to be released by default as OSS.

23 *National Information Assurance (IA) Glossary*, Committee on National Security Systems (CNSS), CNSS Instruction No. 4009, 26 April 2010, <http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf>

24 Ibid. Note that the same document also defines DAA as "Designated Approval Authority (DAA)."

Some like the clear, specific requirements of government security requirement specifications

There were several positive comments about government C&A processes, in particular, several appreciated the clear and specific requirements in some government security requirements documents. One commercial supplier said, "The government does a lot of things that are really good … Government security standards like [the] DISA [Defense Information Systems Agency] STIG[25] [Security Technical Implementation Guide] and Common Criteria[26] are very clear [about] what the security requirements are … [I] really like the clarity and ability to create test cases." Another commercial supplier said that "NIST [National Institute of Standards and Technology] SP 800-53[27] has become very, very important. It is [an] unambiguous set of controls we can refer [to]. It is an objective way of documenting and gives an objective vocabulary, and for the same reasons the SCAP [Security Content Automation Protocol][28] work is enormously important."



Government security specifications are inflexible

The clarity and specificity of government C&A processes such as the Federal Information Security Management Act of 2002 (FISMA)[29] were perceived by some to have a downside: They tended to be inflexible (as applied in practice). *Even small "policy blemishes" would prevent superior products' use or impose unnecessary and costly changes.* One supplier said, "Even one tiny, little thing can block [a program's] adoption … [Widely-used and commercially accepted software] may lack something required by government policy, such as DoD CAC [Common Access Card] support, X.509[30] support, or FIPS [Federal Information Processing Standard]

25 The STIG is a DISA standardized guide for installation and maintenance of computer software and hardware.

26 The Common Criteria for Information Technology Security Evaluation, abbreviated as "Common Criteria," is a set of specifications to permit comparability between the results of independent security evaluations. Products are evaluated against a "Protection Profile," which addresses a real-world security need.

27 *Recommended Security Controls for Federal Information Systems and Organizations*, NIST Special Publication 800-53 <http://csrc.nist.gov/publications/nistpubs/800-53-Rev3/sp800-53-rev3-final_updated-errata_05-01-2010.pdf>, provides guidelines for selecting and specifying security control for U.S. government information systems.

28 SCAP is a standard developed by NIST. SCAP defines automated vulnerability management, measurement, and policy compliance evaluation. See <http://scap.nist.gov/> for more information.

29 FISMA is a U.S. law that requires each federal agency to develop, document, and implement an agency-wide program to provide information security for the information and information systems that support the operations and assets of the agency, including those provided or managed by another agency, contractor, or other source. See the NIST Computer Security Resource Center <http://csrc.nist.gov/groups/SMA/fisma/overview.html> for more information.

30 X.509 is a standard for a public key infrastructure (PKI) that was developed by the International Telecommunications Union – Telecommunication Standardization Sector (ITU-T).

140-2[31] validation … You may need to purchase a commercial clone just to comply with policy," even if the alternative doesn't add any value in its situation.

This inflexibility of C&A processes was particularly a problem for those who want to "try out" systems or do small-scale development before making large commitments to a proposed system. Some potential users may even have difficulty getting permission to install a development environment. These inflexible processes make it difficult to "try before you buy" in real or realistic environments. As a result, it is difficult for organizations to experiment with new systems (including OSS systems) while using real or representative data. A government employee said, "If you have to spend $50,000 to try it, only to find it doesn't do what you want, you've just wasted $50,000. This can eliminate the 'try before you buy' benefits of OSS."

Accrediting authorities should manage risk, not delegate to processes

An accrediting authority[32] is the official with the authority to formally assume responsibility for operating a system at an acceptable level of risk.[33],[34],[35] As one OSS expert said, "[There are] two kinds of [accrediting authorities]: [the good ones] that see their jobs to mitigate risks [and the rest who] have delegated decision-making authority to [processes defined in existing] C&A schemes." The result of the latter may be inappropriate or inflexible requirements; for example, an accrediting authority may require a Common Criteria evaluation even if the product to be evaluated is not an information assurance system.

Need to share/co-develop C&A and authority-to-operate information

Government security evaluation efforts often duplicate each other. There is often little collaboration between different parts of government, even though there are opportunities for sharing and co-developing information. Often, organizations find it useful to know if software they're considering has been approved for use elsewhere.

One OSS supplier said, "Any unique deployment has to have its own ATO [authority to operate]… Just because [you] have an ATO in one set of circumstances … you don't have it in all environments … Some of it is duplication; *it seems like a more modular approach would be appropriate*. If [a product] is certified it should be reusable across agencies … within the same parameters … *There is incredible duplication of effort … It would be nice if [C&A and ATO] standards existed in the government across agencies/organizations* … We have [an agency] as the customer and [its] branches as customers. No overlap, no communications between them … for security/certification *it would be nice if it could reach a certain level and be done*."

---

31 *Security Requirements for Cryptographic Modules*, FIPS 140-2 <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf> is a U.S. government standard used to accredit cryptographic modules.

32 An accrediting authority may be referred to as a PAA, DAA, or various other terms.

33 Security Guide for Interconnecting Information Technology Systems, NIST Special Publication 800-47, section 3.3, August 2002, <http://csrc.nist.gov/publications/nistpubs/800-47/sp800-47.pdf>.

34 Information Assurance (IA), DoD Directive 8500.01E, 24 October 2002, E2.1.13, <http://www.dtic.mil/whs/directives/corres/pdf/850001p.pdf>.

35 National Information Assurance (IA) Glossary, CNSS, CNSS Instruction No. 4009, 26 April 2010, <http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf>.

A different OSS supplier said, "I'm hopeful for things like SCAP. SCAP is shared, machine readable, and disentangled from [the] assumption of risk … People are sensitive [about sharing C&A information]."

A contractor noted, "Building the documents for C&A is the easy part, but then there are test documents and the walk [through at] the customer site. One problem is fragmentation in C&A; different organizations require different things … yet they're basically the same. There is a big need to collaborate … It is critically important that C&A resources be shared [and they currently aren't]. *Develop C&A materials using the same OSS development methods as for developing software*, figure out your weaknesses and mitigate them, and create common reports … the government should sponsor a 'summer of C&A' like Google's 'summer of code.' Get fifty guys trained in C&A, break them up into teams, have them generate documents/artifacts, [and have them] review each other's [work]. Generate documents in one day, critique and fix them the next. Then everyone can see the documents … "

The same contractor believed that "organizations are typically willing to accept an ATO from elsewhere" (at least when the contexts are similar). In particular, the contractor believed it was important to know when someone had approved the use of some software because "no one wants to be first."

C&A cost barriers to entry

Some federal evaluation systems essentially presume that a vendor will pay for an evaluation to sell to the government. Such systems include the Common Criteria and FIPS 140-2 evaluations. These systems create, per one government employee, "a significant barrier to entry that most non-profits and OSS can't tackle … Common Criteria [validation could] cost $250,000 [at least]. How is anyone in the government going to get the software if we can't get it validated for use?"

Another interviewee said, "Can [many OSS suppliers] justify the cost to get [Common Criteria] certified? … Someone needs to sponsor an incubator to get OSS through Common Criteria. We'll share the expense because the government sees a benefit. Commercial OSS companies aren't going to pay for that, [they] need to see benefit this year … [We] need to help companies get through EAL[36] [Evaluation Assurance Level] certification … [We could] advocate/incubate getting stacks through DIACAP [DoD Information Assurance Certification and Accreditation Process] certification for commercial OSS companies … [It] needs to be done [in a] cooperative, CRADA [Cooperative Research and Development Agreement]-type[37] fashion …"

36 The EAL is a numeric value assigned to a product upon completion of a common criteria validation. Increasing numbers indicate validation was conducted against more assurance requirements; these numbers do not measure the security of the system itself.

37 A CRADA is an agreement between the U.S. government and a private organization to collaboratively work on a research and development project.

A different government employee noted that funding certification regimes have been addressed "in some isolated cases like OpenSSL and … mod_nss[38]" but that this is still a problem in general.

An OSS supplier suggested, "*Part of what's missing is [a government-validated distribution with a government issued 'seal of approval']*. The proprietary solutions have those check boxes, they received 'approval' from somewhere and once they're in, the government can buy, buy, buy. They are 'FISMA-certified.' Any sort of validation helps."

Include OSS projects when creating specifications

A serious problem is that when the government creates commercial product specifications (such as Common Criteria Protection Profiles) it sometimes *only* discusses drafts with proprietary software suppliers. This risks creating government specifications that favor proprietary products and exclude OSS programs. One OSS supplier explained as an example, "About a year ago the NIAP [National Information Assurance Partnership] / CCEVS [Common Criteria Evaluation and Validation Scheme] sunsetted all operating system protection profiles [PPs] … [and] when we got to see [its replacement]" the OSS supplier found that the new profile[39] had the following issues:

- It required "ECC [Elliptical Curve Cryptography], which has 25 patents on it [at the time]"—these patents appeared to prevent OSS implementation.
- "Cryptography has to have auditable events, but Linux needs POSIX [Portable Operating System Interface] compliance. The result … required **`setuid`** root for any end-user application that used cryptography [on POSIX systems]." Yet requiring end-user applications to be **`setuid`** root is a terrible security result; it can turn minor security vulnerabilities into major ones.

The same OSS supplier continued, "There was no way for us to interact with NIAP to preview [the] protection profile to give feedback." OSS suppliers simply want the same opportunities that proprietary software developers receive—the opportunity to learn of draft requirements and then have a chance to review and comment on them. *The government should proactively identify and invite all relevant projects (proprietary and OSS) by publicly announcing its intents in many forums (not just the federal register) and seeking out participation of all likely participants.*

SOLUTION: Make software assurance tools freely available to OSS projects

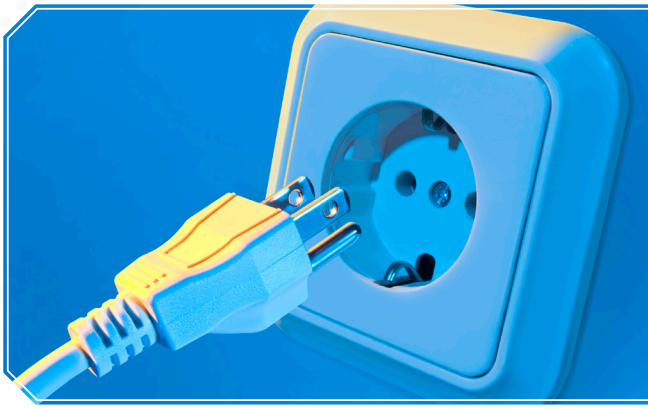
If the government wishes to improve software assurance, some interviewees suggested that there should be free-to-use tools for examining OSS before it is released, both for static and dynamic analysis. One supplier said, "If you want to improve software assurance,

38 mod_nss is a module for the Apache web server that provides strong cryptography via the Secure Sockets Layer and Transport Layer Security standards using the Network Security Services library.

39 "U.S. Government Protection Profile for General-Purpose Operating Systems in a Networked Environment, Version 1.0," which itself was sunsetted on 28 January 2012. This PP is different from previously-used PPs for operating systems, such as the Controlled Access PP (CAPP).

there needs to be tools that are free to the lone wolf developer who can test his code before he releases it … Compare that to [a proprietary static source code analyzer], which may cost $100,000 in commercial settings and you may not be able to share the results. For a lone wolf programmer, what tools do you have for testing beyond your user base?"

STANDARDS/INTEROPERABILITY



Several comments involved standards and interoperability. Interviewees tended to emphasize the use of "open standards" as standards that anyone can read and implement (OSS or otherwise), that do not lock customers into a particular product, that do not favor a particular supplier, and that enable anyone (including OSS projects) to implement the standard (e.g., that do not have license/royalty fees).

There are a variety of reasons to use open standards, because they have the potential to:

- *Improve competition.* A contractor said, "Many problems are not technical, but more policy … The de facto standard becomes one particular vendor. [I recommend that government, both federal and not,] adopt as many standards as possible [and] become vendor agnostic. Then OSS can conform to the standard, and it puts them in the game." An OSS supplier said, regarding the SCAP and OVAL [Open Vulnerability and Assessment Language][40] specifications, "We've asked several times for Windows OVAL probes… We want an open, community standard that's free to use … [and] open [OSS] tools to both author and run scans … We don't want to depend on a proprietary vendor stack … " The contractor recommended that the government emphasize releasing supporting data in OVAL format.

- *Ease integration.* An OSS supplier stated, "Anything standards-compliant easily federates; [a product that] uses all open standards is easier to integrate." This is important for OSS, as it allows the use of different OSS components.

- *Ease access to archival information.* A government interviewee said, "Try to open Word Perfect or WordStar files today [—you often can't]. We've all had files that you can no longer open. Everyone on the planet who uses these tools knows you're going to age out of these formats … *A standardized format that everyone must export to, and then you can extract data from it later, would be really valuable* … Everyone doesn't have to use one format, but at least export to it for government purposes … The ODF [Open Document Format[41]] stuff comes to mind …"

40 OVAL is a standard, funded by DHS's United States Computer Emergency Readiness Team (US-CERT), to promote the open sharing and transfer of publicly available security content. OVAL is used by SCAP.

41 ODF is an International Organization for Standardization (ISO) standard for storing electronic office documents, such as word processing documents, spreadsheets, and presentations. It is a free and open standard that anyone is allowed to use, and it is supported by several office suites.

The government can and should help set standards, especially for security. One OSS supplier said that a "great role the government can play is [in] setting standards. They can hire people who know security very well and run a committee for a long time to create a good standard."

CHALLENGES TO RELEASE OF CODE FROM GOVERNMENT

Interviewees identified many other challenges to releasing software developed using government funding, including releasing modifications of existing software. These challenges impeded releasing software to the public as OSS or even releasing software to another government organization.

Fear that a release obligates the government to support it or use its derivatives

One concern was the perception that releasing software to the public or another government organization would obligate the releasing organization to provide support. One state government employee said, "Agencies [were] uncertain how to answer requests [for software; they were concerned about a perceived] requirement to support questions about code they released … " Agencies believed they were expected to provide support, but they did not have the staff to support it. The same interviewee noted that there "may only be one guy who knows how it works, and it wasn't documented well … [It's] still a problem today, not a legal obligation, but a practical concern. If you share the code you just know you're going to get a phone call. [You] don't want to look bad, but you're not funded to be an IT consultant either. *[It] creates a dilemma the agency would prefer to avoid.* Now, agencies are starting to see the value of building a community [since such communities provide an alternative approach for answering questions and finding support]."

A related concern was that if the government released software, some believed they were obligated to use its derivatives. *As one interviewee put it, some people "honestly believe that if they let the code out, someone will wreck it."* Some people do not realize that if the government releases source code to the public, the government is not obligated to use its derivatives.

Attribution of government employees sometimes considered unacceptable

Some government organizations have modified OSS code, but did not want to publicly release the modifications in a way that would identify (1) the author, (2) the government agency that funded it, and/or (3) the government agency that uses it. Reasons included:

- *Fear of self-aggrandizement.* Some government employees noted, "Really, it's the public's code, they paid your salary, [but] what's the appropriate attribution [not just the individual, but should there be a marking if the government paid for it]? Individuals are all over the source code for contact purposes, so how does the government view that? Are you self-aggrandizing at the government expense? In many cases that's not the intent, but there is the perception that this might be using one's public office to further one's position or further personal gain … it seems like self-aggrandizement … "

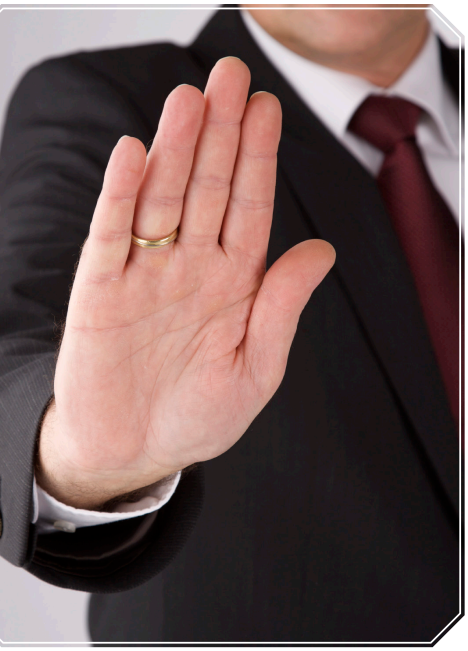
The desire to omit author names does not work well with typical existing OSS projects. While OSS projects typically do not care if the government is identified as a funder or user, OSS projects typically *do* require that the actual author name be included in their submission. This desire for personal identification conflicts with the desire of some organizations to have their software developers remain anonymous.

- *Restrictions on public interaction.* The same employees noted, "[Our agency] likes to stay mute in the press, [our staff is] asked to turn down press interviews. Our agency doesn't want to be in the press in any way other than a very narrow public-facing office. [Other agencies] would probably have those constraints … too." Note that *government organizations that fund development, but want to remain anonymous, could use a generic label stating that the government funded its development (but not exactly which agency).*

- *Concerns about operational security (OPSEC).*[42]. A different government employee stated, "We've found a few cases where we think we have something others may want, but don't want to ask for permission to release to the community … [Our] community is already doing cyber security and most have clearances. Most have an OPSEC poster that [says their agency] will come and get you if you wear your badge too far from the building. You don't want to advertise your [software] stack too far from the organization unless you're at a meeting with like-minded professionals." The employee noted that some organizations are secretive about what software they run, adding that "those in the intelligence community, law enforcement community, and DoD have a fear of [being accused of violating] OPSEC." This concern is not always valid, because "what we call the attack surface is already completely and totally obvious anyway. There's a double standard here. Everyone knows the government uses [common products from large suppliers] … you can go to GSA and look at what contracts were awarded and to which vendors.[43] Maybe either educate everyone that it is not policy, or make policies more clear regarding OPSEC. To not participate in the community is more costly than someone reading the mailing list and figuring out the government is using OSS."

42 OPSEC is "a systematic and proved process by which the U.S. government and its supporting contractors can deny to potential adversaries information about capabilities and intentions by identifying, controlling, and protecting generally unclassified evidence of the planning and execution of sensitive government activities." "National Operations Security Program," National Security Decision Directive 298, 28 July 1992, <https://www.iad.gov/ioss/media/pdf/nsdd298.pdf>.

43 Some contracts, e.g., some classified and lower-tier subcontractors' contracts, are not available to the public. Nevertheless, publicly available information makes it clear that the government uses many products that are also widely used in industry.



Export control and other policies make contributing to the public too slow

Government policies, particularly export control laws and regulations, can create laborious paperwork and review processes before anything is released to the public. These can be so slow that collaboration becomes impractical.

A contractor stated their "big problem" was releasing government or contractor work to the OSS community. Different companies would modify software developed for the government, but it "took a long time to get everyone's attention that these changes need to be fed outside … Most changes are small (fortunately), but it drives everyone crazy. One of the biggest threats to successful OSS adoption is [requiring every single change to go] through the [entire formal] review process … [The] mechanics of moving information … [inside to outside the government] is the problem … [we] need operators or people who understand how the OSS model works."

One government employee said that "the real obstacle [caused by the International Traffic in Arms Regulation (ITAR)] comes with how you collaborate and start a project on github[44] with zero software and develop with external partners. Every commit [to the] OSS world is a release. Unless you have an agreement with external developers to keep it a private project until it's reviewed, you suffer … How do you start a project with external partners? How do you do ITAR reviews of software developed outside? We'll pay more to develop software since we cannot do that one simple thing [develop OSS in public collaboration with those outside the government], especially as OSS gets more popular … We're still not able to collaborate outside of [my government organization] because every change has to be reviewed, e.g., for fear that it could grow into something that's ITAR restricted … Currently every commit has to go through a commit review process. *Every time you delay the feedback loop, you slow the project down to the point of killing it. Delay means death in OSS.*"

Often permission is simply not granted. As another government employee (who is an expert on export controls) said, "Export control is something that could be a roadblock since so much of OSS is dual use by its nature … Both [the Departments of] State and Commerce[45] don't have a lot of depth on the right criteria/parameters for how to look at OSS (or any software) to know if it's really militarily critical. They don't have the staff to do that analysis; by default they may say no, even though they don't know if they should [or shouldn't]. *It's just easier to say no.*"

---

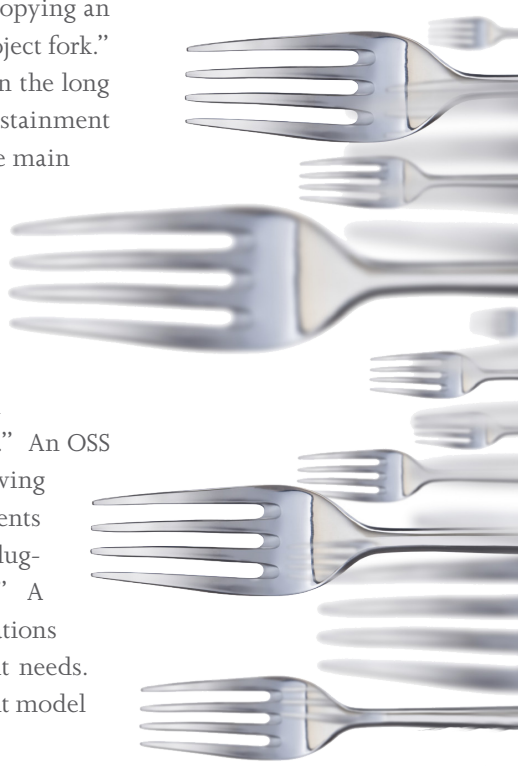
44 GitHub <https://github.com> is a popular network-accessible site that supports cross-organizational collaborative development and sharing of software source code and related data.

45 The U.S. Department of State's Directorate of Defense Trade Controls is "in accordance with 22 U.S.C. 2778-2780 of the Arms Export Control Act and the International Traffic in Arms Regulations (ITAR) (22 CFR Parts 120-130), is charged with controlling the export and temporary import of defense articles and defense services covered by the United States Munitions List (USML)" (see <http://pmddtc.state.gov/index.html>). The U.S. Department of Commerce's Bureau of Industry and Security is "charged with the development, implementation and interpretation of U.S. export control policy for dual-use commodities, software, and technology" (see <http://www.bis.doc.gov/index.php/policy-guidance>).

The same government employee said, "[It would be helpful] if the community itself could put out some authoritative guidance 'if the software had this capability, to this degree, then we don't want it out there, in other cases who cares.' The DoD … probably needs to get more involved with the export control lists, so [the government doesn't] penalize our innovators adversely."

Government creates too many project forks

For the purposes of this paper, creating an independently governed project by copying an existing OSS project is called "forking," and the resulting project is called a "project fork." A project fork is typically far more expensive for the government to maintain in the long term because the government must pay for every change (instead of sharing sustainment costs with others), and the fork is also cut off from the future innovations in the main OSS project. *OSS literature strongly recommends avoiding creating a project fork wherever possible. Yet the government and its contractors often unnecessarily encourage creating project forks, instead of discouraging it.*



One OSS supplier stated, "If an upstream project doesn't meet government requirements, [the government or its contractors will] fork it and create their own [and maintain it] instead of contacting the upstream project to add features … The government does its own work instead of working with others." An OSS expert stated, "We [recommend] not to do another fork … Technology is a living evolution, *the worst thing you can do is take a snapshot and fork it* … contribute improvements back to the core project, keep it unclassified [where you can, and] support plug-in architectures for unique/classified components that can be segmented out." A government employee noted that "OSS [has] particularly good value in situations where you have active development communities building to the government needs. The government… has done a poor job nurturing the collaborative development model … [It needs] to foster building a collaborative community and share code."

Since project forks dramatically reduce the value proposition of OSS, the government should strongly discourage creating project forks. Part of the problem is perverse incentives (see the section on "Wrong incentives within government and contractors" on page 10).

Difficult to release government code even within government

One government employee explained that "within agencies and between, sharing code is ridiculously hard." Some organizational policies require reviews that are quite similar to public releases, even when no public release will occur. The employee also believed that "we need an OSS-like 'license' for use inside the government for those things that truly can't be released—an internal OSS-like license to allow us to share it. [For example, it] would have [text] that says 'this contains ITAR data' that the [government agencies such as the] Army, Navy, [and] NASA have to abide by [where appropriate]. Currently this kind of sharing within the government doesn't happen much." Interviewees also pointed out that the DoD's Forge.mil site was difficult to access outside the DoD. *It might be useful to*

release guidance to encourage the intra-governmental release of source code that would encourage intra-governmental software releases and collaborative development, as well as recommendations for marking source code.

Need a default open government forge, not just a depository

A "forge"[46] is a network-accessible site that supports cross-organizational collaborative development and sharing of software source code and related data. Such sites typically include functions for version control, tracking bug reports, tracking enhancement requests, discussions (e.g., a mailing list), and so on. Forges are intended to enable collaborative development, so a forge should focus on making it easy to maximize the number of participants (including federal, state, and local employees and their contractors); find projects; learn the basics about projects; create new projects; and coordinate software development (including the use of distributed version control systems).

Forges may support public access, restricted access, or both. A forge supports public access if a project can be set up so software (code, documentation, and related material) can be downloaded by the public without restriction. A forge supports restricted access if a project can limit download access. Forges can be created by commercial organizations or by the government. A "government forge" is a forge created and run at the behest of the government. As with other important services, the government should ensure that it is not locked into a single contractor/supplier for maintaining government forge(s). One way to foster collaborative development is to require contractors to develop on a collaborative forge outside their organization from the beginning of development, including posting discussions about changes.

As one interviewee put it, "if you can do it in public, you should." If it's public, the government should use an existing public-access forge. This includes not just the final software, but also anything related to it (where possible). Even discussions between people who work in the same organization should be made public where possible, as this enables others to participate, provides a sense that the project is open to external contributions, and documents the rationale for later participants. Many commercial organizations provide a public-access forge for OSS projects at little or no cost; existing commercial forges should be preferred where appropriate.[47]

However, there are cases where a public-access forge cannot be used, such as for classified or export-controlled software. There are existing government forges that can be used

46 This naming convention stems from the name of SourceForge, an early and still popular OSS forge.

47 The HowTo.gov "Negotiated Terms of Service Agreements" lists tools that have federal-compatible Terms of Service agreements. As of July 11, 2013, it includes GitHub and SourceForge <http://www.howto.gov/social-media/terms-of-service-agreements/negotiated-terms-of-service-agreements>.

in such cases, such as Forge.mil <http://forge.mil>, but many interviewees found them severely wanting. Interviewees stated that:

- *Government forges must allow broad participation.* These sites typically did not easily allow broad participation, including federal, state, and local governments (both employees and their contractors). One government employee said, "Forge.mil … is a great idea, but it doesn't work [for collaborating with those outside of the military]. To get there, [you] need a DoD CAC card or an ECA [External Certificate Authority] certificate … it won't accept the [other federal agency badges]… It needs to be completely open ([so] anyone can get involved) to government. The scope may need to include state and local government too." A contractor said, "There's a business model for an internal OSS community within the walls [of government. However,] there is an artificial and unnecessary boundary between inside and outside the walls."

- *Government forges must provide an easy way for most people to quickly search and browse the projects that are available to determine if they are of interest.* As one government employee said, "[In Forge.mil] every page has a separate permission with no link [to whom you get permission from. There is] no abstract to find out if you want it or not; [you have to go through the drill of getting permissions only to find out it's not what you need]. *If access is not 'default open,' people won't use it.*"

- *Government forges must provide all of the services wanted by developers when doing collaborative development as compared to typical commercial forges.* Government forges often merely act as repositories of "finished" works, instead of operating as collaborative development environments. An OSS expert said, "Gocc.gov [is] now a parking lot. Failed. It was interesting to let people talk to each other and share code, but [it was] not a development environment." A site that is only a repository of abandoned code is sometimes called "the island of lost toys." For example, many OSS developers want to use distributed version control systems (such as **`git`**[48] **and `mercurial`**[49]), but at the time of our interviews, Forge.mil did not directly support one.

In short, it needs to be easier for the government to create and run projects collaboratively across organizational boundaries.

48 Git is a widely used distributed version control system released under an OSS license.
49 Mercurial is a widely used distributed version control system released under an OSS license.

NEED FOR GUIDANCE

There is a need for guidance on evaluating and selecting OSS, for contributing back to the OSS community, and for releasing government-funded software as OSS.

Guidance on evaluating and selecting OSS

One government employee said, "We have no sense or guidance, in terms of NIST-type guidance, for OSS. There's no government document recommending the best way to approach adoption; for example, how do you evaluate the stability of a project vs. the risk of taking it on? [A] lot of developers are comfortable with a particular technology, [but] there's no standardized framework for evaluating OSS projects … [There] doesn't seem to be a decent way to evaluate risk … *[The government] should come out with a government-wide framework for evaluating the risk of an OSS project.* [It] shouldn't decide if an OSS project is suitable—don't decide for the users—but [it] should provide an assessment framework that allows government projects to measure risk … The lowest hanging fruit in terms of government is probably a risk adoption model … Individual agencies, depending on what they're trying to accomplish, may take on more risk." This risk evaluation framework might cover COTS software in general, since OSS is (in nearly all cases) COTS software. Doing so avoids creating OSS-specific requirements.

Another interviewee said, "[The government should explain] 'how to facilitate the adoption of OSS,' where the scenarios would bring benefits to end users [in cybersecurity] … Help them understand the OSS community and how they can leverage it to their advantage."

Another government employee stated, "The government would be assisted with more [development] standards … There is no guidance today on user interface, language, construction kit, [or databases]. I can choose one of five options (Java, C#, Wx widgets, etc.). Some coordination group that issues up-to-date recommendations would lead to the same set of skills. This is not just within the government; industry has this problem too … There's a tendency to use many tools—we should maintain consistency. Guidance can go too far, but some [software development] guidance would be helpful and make a huge difference. We need [development] standards, recommendations, and guidance … [and] someone inside the government should do it." As an example, the employee suggested the need for a C# validation suite and guidance to "not use this list of [C#] APIs [application programming interfaces] because they are not supported on the Mono[50] [C# implementation]."

50 Mono is a cross-platform development framework for the .NET environment released under an OSS license.

Guidance for contributing back to the OSS community

One of the major features of OSS is contribution by the community; those who modify or enhance OSS projects can release their enhancements back to the community as a whole, so that their enhancements can be included in future versions. One government employee stated, "We need guidance and a roadmap for releasing modifications to existing OSS (that the government brought in) back to the upstream project." Another government employee illustrated this need by asking, "Under what circumstances can the government contribute back to the community? Is there a level of appropriateness? … If I modify the code to make it work, am I forced to give it back? Can I? Should I? Even to the point of attribution?"[51]

Another interviewee stated "The best way to get stuff into OSS projects is to implement it as a nice clean patch that follows [that OSS project's] guidelines and is platform portable. The chances of getting it adopted are quite high [in that case]." This means that the government and its contractors need to know (1) what the project guidelines are, and (2) that they need to follow them. Thus, *there is a need for easy-to-follow guidance for releasing the government's modifications to existing OSS back to the upstream projects.*

Guidance about releasing government-funded OSS

A government employee noted, "[we] need a roadmap on how to release government-funded software as OSS under mainstream licenses. Here I'm only talking about software that was developed in-house by government funding, not about modifying an OSS program. *[We] need a federal OMB-level policy [on releasing government-funded software on OSS].*" For example, "How do you do that, and under what licenses? … *We should have a [government-approved] white paper on which licenses are appropriate and the implications of each license in several common use cases.* [Common OSS licenses can only be used] if there's a legal risk analysis; currently it has to be done every time. There needs to be a blanket risk analysis … for mainstream [OSS] licenses … Such a blanket risk analysis would help other lawyers be comfortable releasing under OSS. If the guidance that says that the BSD [Berkeley Software Distribution] license[52] is OK, they … may be more comfortable releasing OSS. …"A different interviewee thought that one way to make it easier for the government to release software as OSS is "for them to *spend a little time [to describe] a series of scenarios that would be accepted by the government for ways to release code as OSS.* We're at a point in the adoption cycle where any time someone comes to a problem, they view it as unique, but it is not unique. [Sharing] where things have worked and where they haven't [helps others when] the scenario is consistent with their needs. [It]

51 Some federal organizations do have policies, though they only cover their organization. Examples include the DoD's 2009 OSS policy and the Consumer Financial Protection Bureau's "Source Code Policy."

52 The BSD licenses are a widely used family of permissive, free OSS licenses that originated with BSD UNIX.

must define when it is appropriate, since OSS is not one-size-fits-all and the answer to all problems." High-level guidance could help people determine if it is appropriate to release a given piece of software as OSS, and how to select an OSS license.[53] Several interviewees pointed out the value of official guidance, but most did not specify at what level such policy, guidance, and release decisions should be made (e.g., at the agency level, higher, lower).

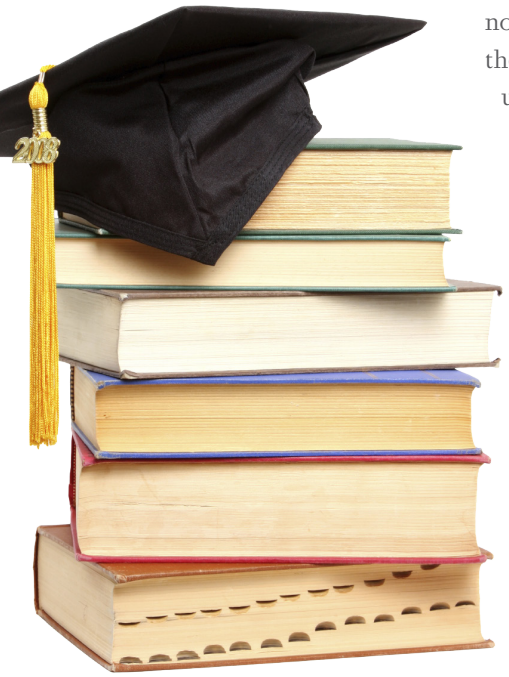
Note that this need for guidance applies whether or not government-funded software is released as OSS by default (see page 16).

NEED FOR EDUCATION

There is a pervasive need for education related to OSS at all levels of government. This lack of education inhibits the effective use and development of OSS in the government. Because many government organizations rely on contractors to supply software expertise, any education initiative must include both the government staff and their supporting contractors.

General OSS education

There is a need for general education about OSS, including the meaning of OSS and how it is developed, among both government employees and contractors. As one OSS expert noted, "A lot of [the problems stem from] the government not understanding how the OSS model works." Another government employee said, "In terms of [OSS] use, the barriers are most typically education. People have a lack of information, [there's] still some FUD at the management level, [and] they think that OSS may be insecure (that's still out there)." Another OSS supplier said, "Another reason (to oppose OSS) is ignorance."



Several emphasized that there is a pressing need to clearly and strictly define OSS and then educate people about what OSS means. A contractor said, "[The government] needs a strict and clear definition of OSS. [Years ago, the term] MOSA [Modular Open Systems Approach] … was a generic term that anyone could define [and this made the term meaningless]. OSS can devolve [the same way] … [We] need a standard definition of the term 'OSS.' Europe gets [OSS] and we don't." This was not hypothetical; another interviewee stated, "Lots of collaboration source software has an engine or some portion that's still proprietary, yet they claim it's OSS … I worked on a project … [where] the analytical piece was proprietary software code. The system was advertised as OSS even through there was a critical path component that was

53 "Publicly Releasing Open Source Software Developed for the U.S. Government," *Journal of Software Technology*, David A. Wheeler, February 2011, Vol. 14, Number 1, <https://www.thecsiac.com/journal_article/publicly-releasing-open-source-software-developed-us-government>, discusses when the government and contractors have the necessary rights to do such releases.

proprietary." A government employee noted that some companies "aren't helping since they [supply] OSS, but they want to drive you to buy additions [which are not OSS]." As a result, the government sometimes does not understand what it will be receiving, potentially leading to poor decisions.

The authors believe that there is a lack of awareness or understanding of the federal government's definition of OSS and not really a lack of policy. After all, there is already a definition of OSS for the federal government,[54] and the DoD has a very clear definition of OSS.[55] It appears that many government employees and contractors are unaware of these definitions and may become confused by suppliers. A formal definition of OSS for the entire federal government (e.g., from NIST) that is clearer than the current definition might help.[56] However, education (not the lack of a policy definition) seems to be the main problem.

Education on intellectual rights and OSS licenses

Unfortunately, many lack an understanding of basic intellectual rights[57] laws, such as copyright. As one government employee explained, "There is an utter lack of knowledge on copyright. For example, NIST requires their name on public domain code with a copyright-based license saying, 'You may not remove our name.' That's illegal[58]. I have seen [code developed by a government employee] with a GPL copyright statement. That's illegal."

There is also a need for education and guidance on the implications of OSS licenses specifically. There are some articles, papers, and organization-specific guidance, but these are not the same. One government employee said, "[OSS] licensing is an issue because people don't understand it … but that's training. It … involves legal people and purchasing people, not just engineers." Another government employee explained, "At the high level, the question is *how do we take advantage of OSS, bring it into [an] agency under a license and know the implications of those licenses*. For example, with the GPL, you modify it and [the] Army wants it.

54 "Software Acquisition," OMB, July 1, 2004, M-04-16, states that "Open Source Software's source code is widely available so it may be used, copied, modified, and redistributed." http://www.whitehouse.gov/omb/memoranda_fy04_m04-16>

55 "Clarifying Guidance Regarding Open Source Software (OSS)," DoD CIO, 2009, states that "Open Source Software is software for which the human-readable source code is available for use, study, reuse, modification, enhancement, and redistribution by the users of that software."

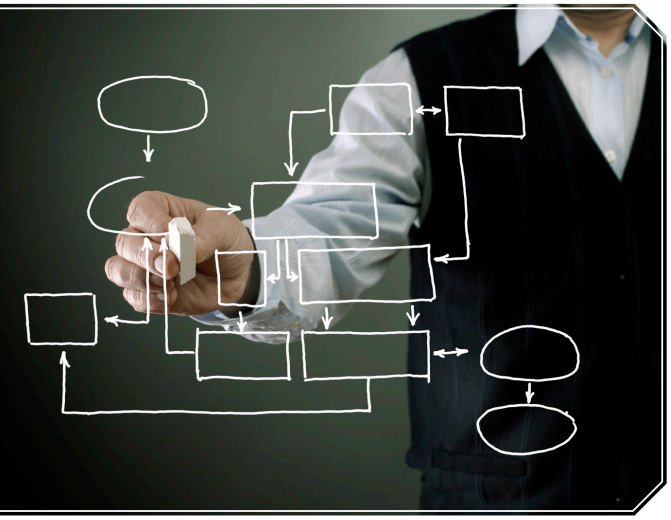
56 For example, "The NIST Definition of Cloud Computing," NIST Special Publication 800-145, provides a concise definition of cloud computing. This NIST publication cuts through confusion about cloud computing by providing standardized definitions.

57 "Intellectual rights" are also called "data rights" and "intellectual property rights." We avoid using the term "property" to avoid confusion. Unlike physical property, intellectual works can be copied without loss to its previous holder. Also, when determining what can be done with government-funded intellectual works, often what matters are the rights that different people have, and not who holds the copyright.

58 Actually, there are some very narrow exceptions that allow certain works of NIST to have U.S. copyright, as explained in CENDI's "Frequently Asked Questions About Copyright Issues Affecting the U.S. Government," CENDI/2008-1, 8 October 2008. We do not know if this code met the criteria for that exception, but the speaker's point is valid, regardless.

If you give them the binaries and they want the source code, you have to give it to them [the Army], but not the public. GPL isn't much of a 'threat' unless you release binaries to the public … Make it clear what you can and cannot do. *Basically a guide for how to use OSS*, [discussing licensing] issues such as the implications for bringing in GPL, modifying it, and sharing binaries (both within and without the government). If you have binaries with ITAR, can you give it [to] someone outside? … Maybe create a checklist… The biggest problem [in using existing OSS] is nobody knows what the current rules are; they are driven by bias and misinformation more than what the licenses actually are."

Another government employee stated, "There are lots of OSS licenses … *It would be nice to standardize on a small set of OSS licenses* … [OSS licenses] can be grouped and probably there are resources to tell you the grouping and what you can and cannot do. E.g., if it's [the Apache license][59] you can and cannot do the following … if it's GPL, under what circumstances do you have to push it back to the community, and so on. Basically, *explain the differences between the groups*. It'd be nice to *have that [OSS licensing information] available in one spot*, with an explanation of the various licenses, and other fundamentals on the licensing issues."[60]



Procurement education

There is a pressing need for education on OSS in procurement. Suppliers today must sometimes become procurement process experts and educate government employees and contractors. One OSS supplier said, "I was forced to become an expert in … procurement, security, what-have-you, because people hear 'OSS' and feel they have to reexamine all assumptions … There is [already] the regular friction in the process [that knocks out the small players]. Adding OSS makes it even more complicated because you have to address all the FUD that goes with it … Procurement hurdles are just as bad for OSS as they are for anyone else. Similar to C&A, but instead of educating the DAA, you [the supplier] are educating the Contracting Office or Contracting Officer's technical representative."[61]

59 The Apache License is a popular OSS license released by the Apache Software Foundation; it is available at <http://www.apache.org/licenses/LICENSE-2.0.html>.

60 CENDI's "Frequently Asked Questions about Copyright and Computer Software" briefly discusses licenses and implications; the DoD has an OSS FAQ that recommends certain licenses. However, there appears to be a need for educational material and guidance that applies across the government and addresses these issues.

61 An example of the need for those involved in OSS to become procurement experts occurred from 2007 to 2009. At that time, acquisition officials did not understand that nearly all OSS met the FAR definition for a "commercial item." It took an OSS expert's presentation (David A. Wheeler's March 2007 presentation at the "OSS in DoD" conference) to explain that OSS did meet the FAR definition for a "commercial item." By June 2007 the Navy had released a policy memo stating this, and in 2009 DoD released its OSS policy memo that broadened this clarification to be DoD-wide.

A government employee suggested that the federal government should "push it to the acquisition community. If you can, *get DAU [Defense Acquisition University (DAU)] to recognize OSS and include it in their teaching materials.* Anyone who does acquisition [in the DoD] must be certified by DAU. They would be the best attack point [in the DoD]." For other federal organizations, ensure that their training programs for information technology acquisition include information on OSS.

Certification and Accreditation education

Few people understand the government's C&A process. A government employee said, "Nobody understands the C&A process."

An OSS supplier shared, "*FISMA is all about teaching the customer.* [For example, FISMA is] not a certification, it's an accreditation. Customers ask for FISMA documentation [from us]. We can't give them [our corporate] chapter of FISMA documentation; it's woven into the documentation … Having an objective vocabulary for [the] C&A process is crucial. Having customers understand the difference between certification and accreditation is crucial."

CONCLUSIONS

Collaborative software development—including the use and release of OSS—can reduce costs, reduce development time, and improve overall quality, including security (through increased transparency and mass peer review). However, there are many challenges in the government to the collaborative development of software (including OSS), as well as to the use of such software. Many of these challenges occur across multiple agencies. These interviews of OSS experts, suppliers, and potential users identified a number of these challenges. These challenges (and some suggestions for addressing them) can be grouped into the following categories:

1. *Current OSS use and development in government.* OSS is used and developed in government; more is expected as budgets shrink. However, there is a lack of documented examples to emulate. We recommend the government create and distribute OSS success stories as "case studies" so that others can build on those experiences and replicate these examples.
2. *Inertia.* One of the key challenges widely noted was inertia—that is, resistance to change. We recommend the government publish case studies (as described above), as this may partly counter fear of change, and increase emphasis on modularity and standards to partly counter transition costs.
3. *Fears about low quality or malware.* Interviewees agreed that there was not an increased risk of low quality and malware in OSS, but that the perception that it does has inhibited its use. We recommend the government educate its employees

and contractors that the "continuous and broad peer review enabled by publicly available source code supports software reliability and security efforts,"[62] and that there are ways to evaluate OSS.

4. *Concerns about commercial support and warranties.* There were many comments about commercial support and warranties. We recommend the government educate their employees and contractors about the many options for support and warranty of OSS; often there are commercial support vendors for OSS, and self-support is a valid support approach.
5. *Procurement.* Government program offices and contractors should be incentivized to build collaborative communities and to share code. We recommend the government:
    a. Avoid presuming, when developing RFPs, that respondents will have a particular business model.
    b. Avoid imposing unnecessary paperwork burdens.
    c. Consider contributing Section 508 material (such as VPATs) for major OSS projects, to ensure that accessibility capabilities are documented.
    d. Emphasize the value of accepting "80 percent solutions" and tailoring as necessary, instead of "100 percent solutions," which have much larger costs and delays.
    e. Update processes to become faster and more nimble.
    f. Include requirements in RFPs and contracts that the government must receive source code and unlimited rights if the development is government-funded, unless special waivers are granted.
    g. Clearly state that source code must be shared within the government, as appropriate, if its development is government-funded.
    h. Consider switching to releasing unclassified software as OSS by default if its development is government-funded.
6. *Certification & Accreditation.* There are numerous problems with the current C&A processes. We recommend the government:
    a. Refocus C&A efforts on risk management, instead of implementing inflexible processes.
    b. Share C&A data and ATO data where possible (e.g., by hosting a "summer of C&A").
    c. Ensure that all relevant parties, including OSS suppliers, are involved when the government is developing specifications (such as Common Criteria Protection Profiles), instead of only inviting suppliers of proprietary products.
    d. Develop approaches for funding security evaluations (e.g., through Common Criteria and FIPS 140-2) in cases where evaluation is important but it will not be funded otherwise. The cost barrier is a difficult but

62 "Clarifying Guidance Regarding Open Source Software (OSS)," DoD CIO, 2009.

important problem to tackle.

7. *Standards/interoperability.* Often government uses proprietary formats and protocols instead of open standards. We recommend the government switch to modular systems and open standards, enabling the government to more easily transition to alternatives, including OSS. We also recommend the government take a more active role in developing these open standards and in developing OSS implementations of them.
8. *Challenges to the release of code from government.* Current policies make it unnecessarily difficult to release software developed using government funding; this impedes the release of changes to existing OSS, as well as the release of entire programs as OSS. We recommend the government:
    a. Clarify, in policy, that government release of software does not necessarily obligate the government to support it or use it.
    b. Clarify, in policy, that it is acceptable for people to put their names on code releases, even if the government funds it, and that it is acceptable to identify their sponsor as the "U.S. government" if that particular department or agency does not wish to be identified.
    c. Speed the government's internal review processes, particularly for export control, as these can greatly slow the release of software changes. As one interviewee put it, "Delay means death in OSS."
    d. Discourage government-unique project forks (changes should normally be fed back to the "main" project).
    e. Where possible, develop software in public and use an existing public-access commercial forge.
    f. Support broad participation, discovery, and active development of software.
9. *Need for guidance.* We recommend the government develop and release guidance on evaluating OSS (including the impact of OSS licenses such as the GPL), on contributing to OSS communities, and on releasing government-funded projects as OSS.
10. *Need for education.* We recommend the government ensure that both employees and contractors are educated on OSS, as appropriate. This includes education on OSS in general, on intellectual rights and OSS licenses, on the impact of OSS on government procurement (for potential suppliers), and on C&A. For example, DAU should consider adding an OSS courseware module to reach acquisition professionals.

We had expected many challenges to relate to security or perceptions of security. We did find these, but many of the challenges were in other areas. Nevertheless, these "non-security" challenges can impede security because they can prevent the appropriate use or development of OSS programs that implement security features or are themselves more secure.

To maximally use its limited resources, the U.S. government must address these challenges and reduce the unnecessary barriers to the use and development of OSS. Many of these challenges can be addressed by promulgating education and guidance on OSS for different roles. The U.S. government should also transition to increased transparency and openness. In addition, many interviewees stressed that contracts should require that software and C&A materials developed with government funding be maximally shared, developed collaboratively, and provide full data rights to the government (unless it can be justified that fewer rights benefit the government as a whole); interviewees also emphasized that the government should release such software as OSS by default. Indeed, sharing software within (or outside of) an OSS context ensures that the largest government population is aware of that software and able to apply it to further their organization's mission. The advantages of such collaboration often far outweigh their potential negatives. By reducing barriers, the government can create new public/private and public/public partnerships to help meet its missions in the years ahead.

ACRONYMS

ATO	Authority to Operate
BSD	Berkeley Software Distribution
C&A	Certification and Accreditation
CAC	Common Access Card
CC	Common Criteria (short name for Common Criteria for Information Technology Security Evaluation)
CCEVS	Common Criteria Evaluation and Validation Scheme
CNSS	Committee on National Security Systems
COTS	Commercial Off-The-Shelf
CRADA	Cooperative Research and Development Agreement
DAA	Designated Accrediting Authority or Designated Approval Authority
DAU	Defense Acquisition University
DHS	Department of Homeland Security
DIACAP	DoD Information Assurance Certification and Accreditation Process
DISA	Defense Information Systems Agency
DoD	Department of Defense
EAL	Evaluation Assurance Level
ECA	External Certificate Authority
ECC	Elliptical Curve Cryptography
EIT	Electronic and Information Technology
FAQ	Frequently Asked Question
FIPS	Federal Information Processing Standard
FISMA	Federal Information Security Management Act
FAR	Federal Acquisition Regulations
FUD	Fear, Uncertainty, and Doubt
GTRI	Georgia Tech Research Institute
GNU	GNU's Not Unix
GPL	GNU General Public License
GSA	General Services Administration
HOST	Homeland Open Security Technology
IDA	Institute for Defense Analyses
IA	Information Assurance
IRAD	Internal Research And Development
ISO	International Organization for Standardization
IT	Information Technology

ITAR	International Traffic in Arms Regulation
MOSA	Major Open Systems Approach
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
ODF	Open Document Format
OPSEC	OPerational SECurity
OSS	Open Source Software
OVAL	Open Vulnerability and Assessment Language
PAA	Principal Accrediting Authority
PKI	Public Key Infrastructure
POSIX	Portable Operating System Interface
PP	Protection Profile
RFP	Request For Proposal
S&T	Science and Technology
SCAP	Security Content Automation Protocol
STIG	Security Technical Implementation Guide]
VPAT	Voluntary Product Accessibility Template®

DAN MASSEY

HOST Program Manager
Cyber Security Division
Science & Technology Directorate
Department of Homeland Security
Host@hq.dhs.gov
http://www.dhs.gov/csd-host