

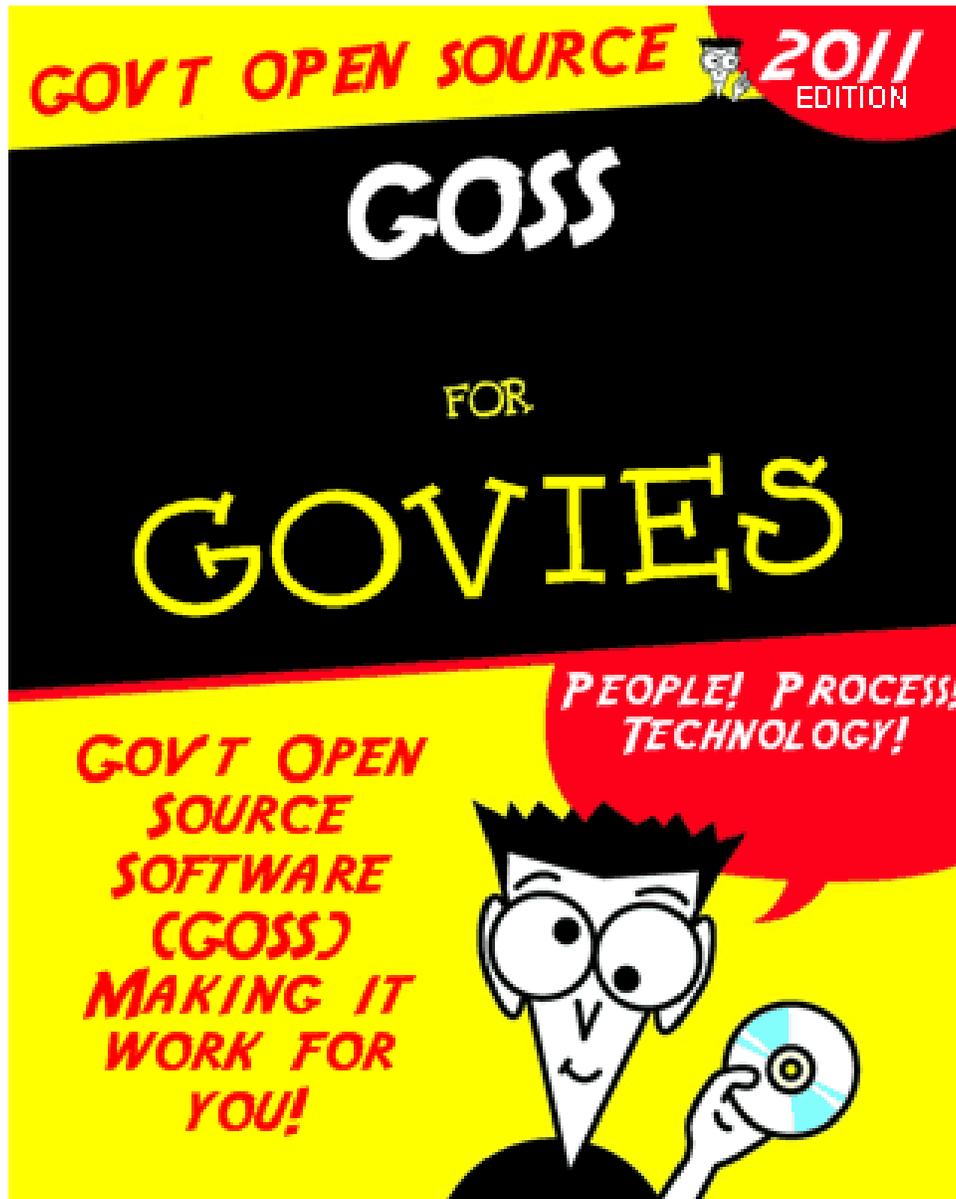
Office of the Director of National Intelligence



**Government Open-Source Software (GOSS)
“Handbook for Govies”**

Version 1.7

29 June 2011



Prepared by:

IC Enterprise Engineering
NBP 304, Annapolis Junction, MD

INTRODUCTION

“Build it, and they will come” - *Field of Dreams*

“Good programmers know what to write. Great ones know what to rewrite (and reuse).”

Eric S. Raymond, Author, *The Cathedral and the Bazaar* (Why open-source works so well)

Software development efforts in the government and Intelligence Community (IC) have been going on for years, costing millions and millions of dollars, and for years, various agencies have identified duplicative endeavors between separate, but similar, software development efforts. One such agency, the National Security Agency (NSA), determined that an initiative of sharing in development is the ideal way to leverage a talented pool of people, using “best practice” processes and ever evolving modern software technology. The NSA drafted their Ozone Widget Framework (OWF) Government Open Source Software (GOSS) initiative, an approach based on sharing and collaboration that began a new era in software development for the IC and that eventually will be shared across the government.

The purpose of this document is to ***share the GOSS experience*** with you and your agency, organization, or partners. This document outlines an “in-use” framework (OWF GOSS) on which you can model a GOSS development environment for your applications, project, or program.

The IC Enterprise Engineering (ICEE) Team gratefully acknowledges the OWF GOSS Advisory Board (GAB) members and the many GOSS advocates who lent their time, knowledge, and technical expertise to the writing of this handbook. Some of this content was adapted or derived directly from the GOSS Intellipedia web site, numerous GOSS presentations, and GAB meeting notes.

Throughout the extended government enterprise, GOSS will be developed with close and continuing coordination and cooperation with our partners in the Intelligence, Defense, Homeland Security, Law Enforcement, and Foreign Affairs communities. As our targeted service partners and stakeholders, they will be referred to throughout this document as our constituency.

The current OWF GAB is comprised of representatives from NSA (Project Sponsor/Lead), Department of Defense (DoD) Chief Information Officer (CIO), IC CIO, Central Intelligence Agency (CIA), and Defense Information Systems Agency (DISA) and meets monthly or as needed. It is necessary to have a guiding body for your GOSS efforts. The GOSS model has allowed the OWF constituency to use a repeatable process for OWF software development and refinement under a single framework. On 10 February 2010, the Architecture, Standards, and Engineering Committee (ASEC) endorsed GOSS as the preferred model for future IC enterprise solutions.

How is this handbook organized?

Chapter One – What is GOTS? What is GOSS? A Brief History

Chapter Two – Why GOSS? What’s in it for me?

Chapter Three – People, Processes, and Technology

Chapter Four – Ozone Widget Framework (OWF), GOSS in Action

Chapter Five – GOSS Frequently Asked Questions (FAQ)

Appendix A – Open-Source Acronym List

“When it comes to software, I much prefer free software, because I have very seldom seen a program that has worked well enough for my needs, and having sources available can be a life-saver.”

Linus Torvalds, Free Open Source Software (FOSS) Advocate,
Linux Founder

CHAPTER ONE – What is GOTS? What is GOSS?

What is GOTS? GOTS is the acronym for "Government Off-The-Shelf." It refers to software produced by a government entity, rather than by a commercial source. A single agency will fund and prioritize the development of the software, but after completion, other agencies who have a need for the software can share it freely.

GOTS Challenges: The challenge of GOTS is that despite the fact that the product can be used by many agencies, often, individual agencies require additional features that are unique and/or mission specific. In some cases, GOTS software has technical aspects that enable those agencies to make changes locally. Unfortunately, GOTS generally lacks the people, processes, and technologies to receive those local changes for integration back into the software baseline. As a result, many different versions of the same software exist throughout the different network fabrics.

GOTS Drawbacks: OWF users encountered the following problems while using the GOTS model, which eventually, compromised product interoperability:

- Because most GOTS products are delivered as a bundled product, they are not extendable.
- Because OWF has aspects which allow it to be locally manipulated, agencies extended the product with local patches, which, after time caused source code baseline forks.
- There was extensive duplication of development efforts, increased Level of Effort (LOE), delays in time to deploy to the user market, and duplication of lifecycle costs.

...“the key to GOSS is to leverage People, Process, and Technology”...

Mr. Alex Voultepsis, IC CIO

What is GOSS? GOSS is the acronym for Government Open-Source Software. GOSS is computer software for which the source code and certain other rights normally are reserved for copyright holders in the private sector that is available, in source code form, to approved government users/developers. GOSS users are provided the software code that permits the developers to review, change, improve, and at times, to distribute the software. GOSS is made available not only to government users, but also to the private sector, for use on government projects.



GOSS software should be developed in a completely *open and collaborative* manner. GOSS products' open-source development can be compared to user-generated content or open-source products like Apache HTTP Server; the wiki platform, MediaWiki, used for Wikipedia; and the internet browser, Mozilla Firefox. One of the most successful open-source products is the GNU/Linux operating system, an open-source, Unix-like operating system. NSA extended the Linux Operating System to build SELinux, which it released back into the Linux mainstream. See <http://www.nsa.gov/research/selinux/> for more information.



GOSS project's outreach staff should declare their mechanisms for receiving code from the GOSS contributor community. Open-source software projects use a variety of models, many of which should be usable on a GOSS project. OWF GOSS development employs a core developer/patch contributor model. In this model, interested parties may submit patches for inclusion in the baseline. Patches may be bug fixes or new features. Patch submission follows patterns utilized in the open-source community. The OWF program chose this model for the following reasons, which may or may not hold true for other GOSS projects:

- Use of this model ensures that the core development team inspects all submitted code for quality concerns.
- This model for OWF ensures that the OWF cadre is ultimately responsible for establishing and maintaining the code baseline.

All development of the core capabilities for OWF ***GOSS is currently done at the unclassified level.*** However, GOSS development should be done at the appropriate domain classification level, based on the classification of the software being developed. Minimally, read-only access to the source code repository and a common issue, (e.g., features and bugs, tracking system) should be available and open to interested parties who have a desire to track the baseline and optimally, contribute new features or bug fixes back as patches. For the OWF GOSS program, these systems are currently hosted to provide maximum availability of the infrastructure to the development and consumer constituency.



“Eliminate the FUD right up front!”
FUD = Fear, Uncertainty and Doubt

GOSS DoDIIS Briefing, April 2011

A Brief History

The Birth of GOSS: Why open-source is the preferred choice? The OWF program was transformed from a GOTS solution to a GOSS solution in the summer of 2010. A single agency developed OWF in its GOTS form and then, circulated it to multiple agencies as a downloadable application or via software distribution. Prior to the transition to GOSS, agencies utilizing the software had no viable option to implement their requests and priorities within the software baseline. The GAB guides and drives OWF's (in its GOSS form) development priorities, and any agency or program who wishes to obtain the source code of the software may request it for its own use to extend, enhance, or tailor to meet their specialized needs. Agencies are encouraged to contribute any modifications or feature enhancements back to the baseline, for their own benefit and for the good of the overall constituency.

Why OWF? The OWF is a common thin-client environment used for dynamic workflows, leveraging and employing lightweight web-applications (widgets) distributed across an enterprise. Based upon the use of the OWF product by a broad array of agencies, OWF was chosen for a GOSS initiative. It was not based upon a particular agency's process or technique, but a product that was popular and being used broadly throughout the national security enterprise. Care should be given when choosing an application to become GOSS. If the software product is of a general nature with general applicability beyond the government's national security enterprise, transitioning from GOSS to Free and Open-Source Software (FOSS) may be the ultimate goal.

“... considering factors such as performance, cost, security, interoperability, ability to share or re-use, and availability of quality support.”

Office of the President of the United States, OMB guidance memorandum, 7 Jan 2011

CHAPTER TWO – WHY GOSS? WHAT’S IN IT FOR ME?

The choice to use and develop GOSS is determined by the product you are developing and the product uptake external to your organization. GOSS may not always be the best answer for your product or organization; however, the primary advantage of GOSS is that a better product is made available in a more timely and efficient fashion when compared to the GOTS approach to development.

What most notably distinguishes GOSS from other open-source software is that GOSS leverages the synergy of the constituency’s people, processes, and technology to support the government, and the product’s customers, stakeholders, and contributors are limited to the government and its partner support organizations.

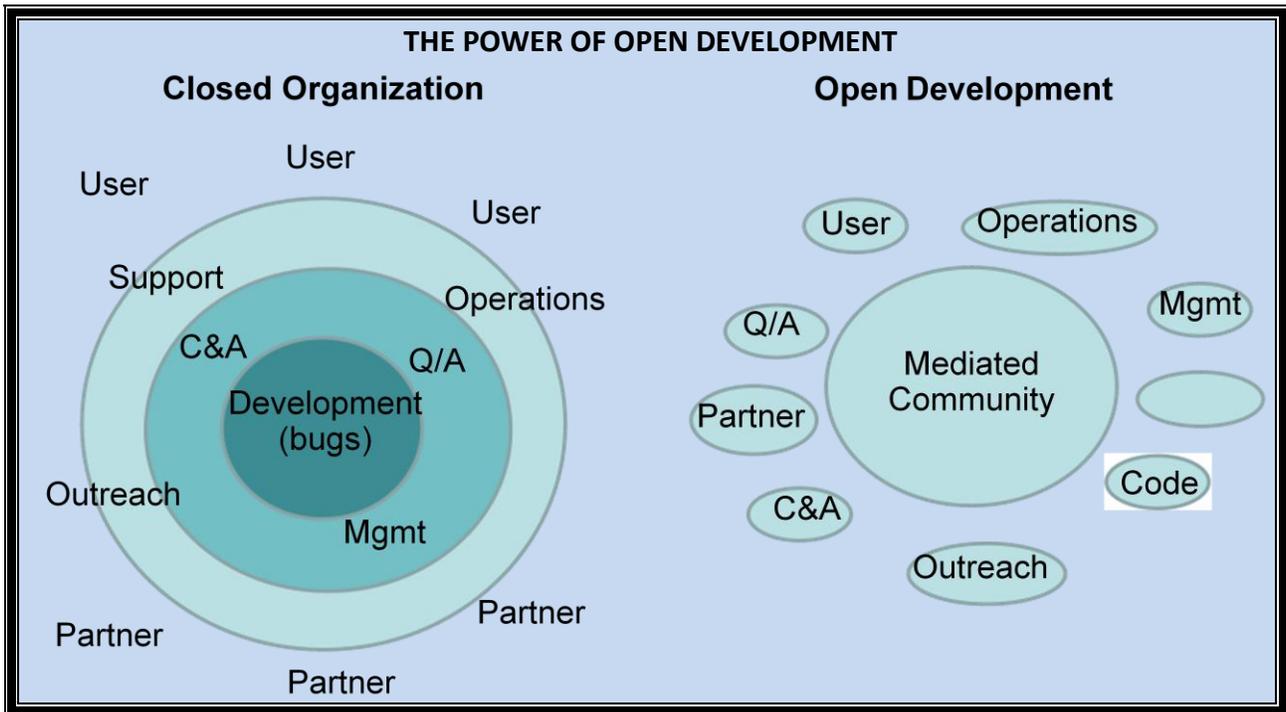


Figure 1 – Power of Open Development

The Power of Open Development: An essential element of GOSS development is that the source code is available and legally modifiable, allowing for the continued refinement and improvement of your chosen software application. GOSS is computer software that is available, in source code form, for which the source code and certain other rights normally reserved for the development agency are provided broadly for development, enhancement, and testing. GOSS follows the FOSS community development model, which keeps the software open to all developers so that they can continually test, manipulate, and change the software in an open, collaborative forum. The current GOSS model for OWF, however, does not use volunteer developers, but government paid contractors or government developers.

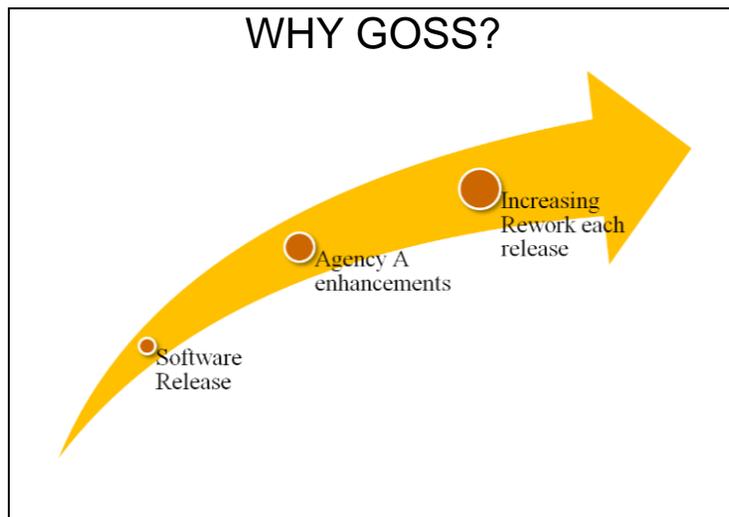


Figure 2 – Why GOSS?

“...agencies should analyze alternatives that include proprietary, open-source, and mixed source technologies...”

Office of the President of the United States, OMB guidance memorandum, 7 Jan 2011

GOSS Benefits

GOSS will provide significant advantages to the entire community that uses the GOSS sponsored applications. Users, mission, and leadership drive the future of GOSS. One of the tenets of IC Enterprise Engineering is STARS, which represents that the services will be **Simple, Timely, Agile, Robust, and Self Service**. GOSS software deployment embodies the cornerstone of this ideal. Some other benefits are:

- GOSS permits you and your agency to study, change, and **improve** the software in a cohesive and synchronized fashion.
- It is an **evolutionary way** of developing, distributing, licensing, and consuming software that supports your mission and enables entire constituency involvement by giving members and partners ownership and responsibility.
- It gives you and your agency the ability to submit software patches, features, and enhancements for inclusion in the baseline.
- **Costs are reduced** when patches are accepted and released. When agencies accept added enhancements to the software baseline, they **enhance** application capabilities.
- The common software baseline increases **interoperability** across the government, agencies, and networks.
- GOSS gives you **influence** to request features.
- **Risk is easily managed** as a result of full visibility during ongoing, continual development.
- GOSS works to **leverage proven models** seen in the open-source software model, with as few alterations as necessary to benefit the “closed-garden” of the government space.
- GOSS is about **working together** to determine priorities for your GOSS program; through collaboration, groups discover shared needs and opportunities.

- Being a member of a GAB gives you **voting power** to set priorities for features and recommend patches and bug fixes in upcoming releases and enables you to represent your user community, mission, and leadership focus.
- GOSS gives you **influence** to request features.
- Agencies can **leverage the talent and technology** throughout the constituency in a cohesive and productive fashion.

CHAPTER THREE – PEOPLE, PROCESSES, AND TECHNOLOGY

“...minimize bureaucratic costs (time, money, and opportunity cost) through use of a light-weight governance model.”

OWF GOSS Mission and Charter, June 2010

GOSS Process: The GOSS Process has been adopted from Open Technology Development (OTD) industry best practices and the software development lifecycle model. First, you need to identify a GOSS application for sharing. Once a GOSS software proposal has been accepted, a GOSS software development program can be implemented, using the following process steps:



Process Steps:

Step 1: Determine if an existing product or a product idea may have value to a group beyond its original set of developers or users. A program with no interested audience is likely best left as an internal project.

Step 2: Identify the projects that can be adopted into a GOSS framework. Each project must have source code that can be used and improved and that will allow for collaboration by software developers and users. **Note:** A project could be initiated fully under GOSS and therefore, not require pre-existing code.

Step 3: Establish governance. Any open-source program (FOSS or GOSS) needs to determine its appropriate governance structure (i.e., who should determine the set of features for inclusion in a release and who should decide whether a patch may or may not join the baseline). Government leadership deemed a more formal governance structure was necessary for OWF because GOSS software is often built not via a true volunteer effort but rather, through agencies directing their development teams (contractor and otherwise) to build out capabilities. Other projects may consider proven models such as the benevolent dictator model or consensus-based democracy. Whatever the model chosen, it should be clearly described for the benefit of any groups seeking to utilize or contribute to the baseline. The OWF GAB has a mission statement and charter located at: <http://go.usa.gov/TvX> that could serve as an example on how to build a GOSS charter for your agency.

Step 4: Establish collaboration spaces. This step requires an agency sponsor and host for the open source software and development spaces. Forge.mil is an existing government collaboration area, though it is restricted to DoD credentialed users. One may consider using areas such as SourceForge.net or GitHub for programs that do not require protection within the closed walls of government spaces.

Step 5: Establish technical priorities. The OWF GAB determines the direction for its project development and the priorities of the development. Similarly, your program should have a mechanism to determine its technical priorities to better shepherd the collective efforts of the contributing community.

Step 6: Select the appropriate license for the project, as well as any copyright restrictions for submitters.

Step 7: Establish communications gateways with communities of interest (COI). Be aware of any communications concerns, such as ITAR (International Traffic in Arms Regulations) restrictions. ITAR is a set of United States government regulations that control the export and import of defense related

services on the United States Munitions List (USML). These regulations implement the provisions of the Arms Export Control Act (AECA). Along with actual weapons, amongst the restricted arms is "strong" encryption application software. Software that implements SSL in the US cannot be exported because of these rules. For specific information regarding ITAR, visit the State Department website: http://www.pmdtc.state.gov/regulations_laws/itar_official.html.

Step 8: Establish a patch submission process.

Step 9: Continuously review steps 1-8. GOSS development is a continuous process, where projects should constantly be cycling through software patches, releases, and updates. Over time, the constituency will naturally grow, bringing in new people, improvement of processes, and new technology.

The OWF GOSS website, leveraging the technology for collaborative software development can be found here: <http://www.owfgoss.org>.

All GOSS community members can request access to see the working state of the project. There are dashboards, per product area, to show the current state of the release. Individual issues link to related code in the repository.

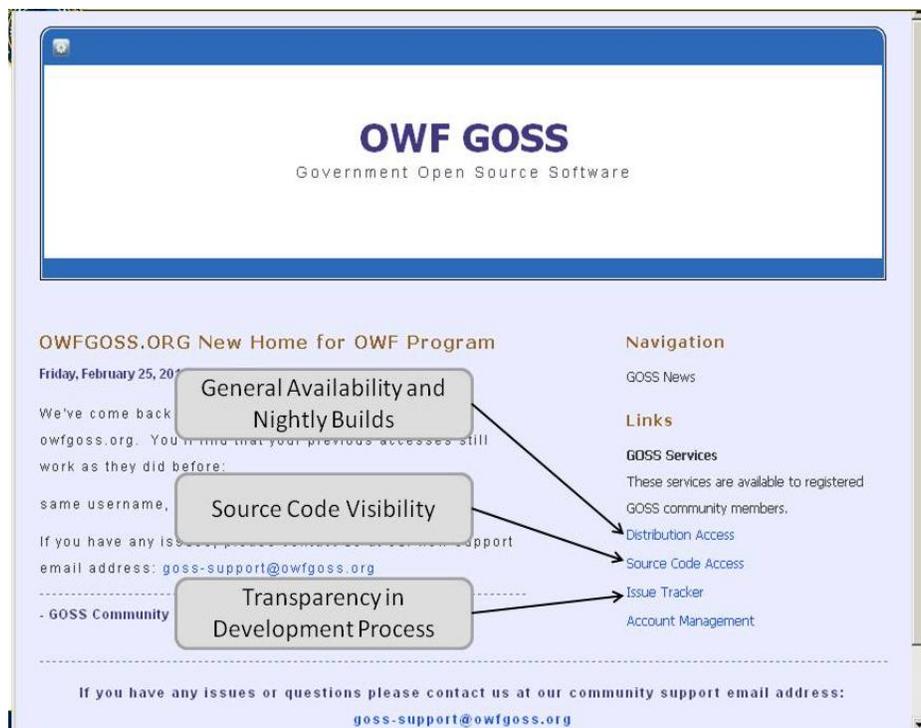


Figure 3 – The OWF GOSS Website Screen Shot <http://www.owfgoss.org>

Note: OWF does not have nightly builds as yet, though they have recently begun making available snapshot/alpha builds.

CHAPTER FOUR – OWF GOSS IN ACTION

“Mission Statement: To enable the community to make best use of the Ozone Widget Framework and its corollary products.”

OWF GOSS Mission and Charter, June 2010

OWF goes to GOSS: OWF is an NSA originated and sponsored program. OWF was developed, as part of the Ozone Program, to help solve a set of problems encountered when NSA analysts were trying to bring together information from disparate sources. OWF is the presentation tier integration solution that came out of those GOTS development efforts. Early prototypes of the capabilities had shown great practical use and promise to the analysts testing the product, but it was not until 2009 that the project began to mature.

NSA works with other agencies in shared initiatives, and in those initiatives, agencies bring to bear tools that they think may help solve issues. The Infrastructure, Innovations, Prototyping (I2P) program began to use OWF to allow analysts at different agencies to share information within a common end user application. This demonstrated the usefulness of OWF to those agencies using the product, which, in turn, exposed it to their internal programs. **OWF began to spread like wildfire.** NSA made OWF available, as a GOTS application, to all those agencies and software development groups that wanted to use it. OWF is now distributed on the government’s TS, Secret, and Unclassified fabrics, as well as through the owfgoss.org site.

OWF is a web application that makes heavy use of JavaScript; its client-side code is exposed. As the use of OWF flourished beyond a few elements within NSA, software development teams began to extend client-side capabilities by updating their local JavaScript files. When a new version of OWF was released, the development teams, at a minimum, would be forced reapply their software changes, and often, they would have to rewrite them, as OWF capabilities evolved for the platform. There was neither transparency nor visibility of changes made by various OWF developers. OWF developers were not aware of other agency software changes, and they were not even aware of NSA’s OWF changes until the software was distributed. This fragmentation and code forking frequently was a source of software rework and frustration. The DoD CIO was making use of OWF for the DoD Storefront project in 2010, and that team was extending OWF’s capabilities. The DoD CIO voiced concern with the GOTS OWF approach and resulting challenges and formally requested that NSA release the full code. This request led to the transition from GOTS to GOSS.

Why was OWF transitioned to GOSS? OWF is a wildly popular, successful, and extensively used software application. Today, it is used widely across the DoD and the IC and its partners. Therefore, it made sense that OWF be moved toward a mechanism that allowed the software’s capabilities to be extended by multiple groups of developers and interested users. NSA has fully engaged with other agencies and software development groups to extend the functionality of OWF and to expand its use across multiple agencies and analyst disciplines. OWF’s success has been built upon a passionate group of developers and users.

GOSS Goals: The OWF GAB established a baseline of goals for its GOSS program. Some of these goals already have been met, and some are actively being worked as of this publishing. If you are to establish a GOSS program for your agency, it is important that you establish goals for the program and meet them on time. The following are examples of how the OWF GOSS program went forward:

The Short-Term goals for GOSS were:

- Draft a CIO-level agreement – Completed.
- Form an Initial set of Advisory Board members – Completed.
- Draft a Roadmap driven by the Advisory Board – Completed.
- Adapt Core-Contributor Model, with patch submissions – Completed.

The Long-Term goals for GOSS were:

- Transition to government-hosted infrastructure.
- Add other agency stakeholders to GAB - On-going.
- Extend Core Contributor Set - On-going.
- Promote widget sharing, in addition to framework development - On-going.
- True FOSS - Proposed Goal.

Key facets of GOSS development are speed and agility. If you understand software development within the government and even within the private sectors, there are some lessons that you can learn from the most recently published OWF GOSS Timeline:

- 30 Jun 2010: OWF GOSS Mission and Charter review, initial GAB meeting. The OWF GOSS Charter is located here: <http://go.usa.gov/TvX>.
- 11 Aug 2010: First Roadmap review, the GAB revised priorities, need for functionality extension identified, GOSS member volunteers to submit patch.
- Aug 2010: Roadmap was made available to the OWF community.
- 01 Sep 2010: Stand-up of GOSS environment site at <http://www.owfgoss.org>.
- 13 Dec 2010: GAB meeting, prioritized release features, established membership criteria; DISA representative was included on the Board.
- Feb 2011: GOSS approach endorsed by the IC's Joint Systems Engineering Committee (JSEC) and ASEC.
- 08 Mar 2011: GAB meeting; GAB meetings are now held quarterly to discuss progress on the previous release, proposed priorities for the upcoming release, and any other Board business. Recently, the OWF program has worked to increase its communications with the Board to allow it to better prioritize the needs of the constituency by implementing a mid release review (on-line session), as well as beginning a constituency conference/Technical Exchange Meetings (TEM), at which the overall constituency can express their desires and priorities.

OMB Guidance on Technology Neutrality: On 7 January 2011, The Office of Management and Budget (OMB) issued executive guidance to all government agencies that any technology acquisitions should be based on performance and value, not on preferences for how technology is licensed or distributed.

The memo from the Federal CIO, Vivek Kundra; Administrator for Federal Procurement Policy, Daniel Gordon; and Federal Intellectual Property Enforcement Coordinator, Victoria Espinel, advised agencies of the policy that helps to ensure "federal investments in IT are merit-based, improve the performance of our government, and create value for the American people." This memorandum encouraged agencies to consider alternatives to brand names, such as proprietary, open-source, and mixed source technologies. "This means selecting suitable IT on a case-by-case basis to meet the particular operational needs of the agency by considering factors such as performance, cost, security, interoperability, ability to share or reuse, and availability of quality support."

OMB Guidance Memorandum Excerpt



EXECUTIVE OFFICE OF THE PRESIDENT
OFFICE OF MANAGEMENT AND BUDGET
WASHINGTON, D.C. 20503

January 7, 2011

MEMORANDUM FOR CHIEF INFORMATION OFFICERS AND
SENIOR PROCUREMENT EXECUTIVES

FROM: Vivek Kundra *Vivek Kundra*
U.S. Chief Information Officer
Daniel I. Gordon *D. I. Gordon*
Administrator for Federal Procurement Policy
Victoria A. Espinel *V. A. Espinel*
U.S. Intellectual Property Enforcement Coordinator

SUBJECT: Technology Neutrality

The purpose of this memorandum is to remind agencies of the Federal Government's policy of selecting and acquiring information technology that best fits the needs of the Federal Government, including of being technology and vendor neutral in acquisitions for information technology (IT). This long-standing policy helps ensure that federal investments in IT are merit-based, improve the performance of our government and create value for the American people.

The Federal Acquisition Regulation and Office of Management and Budget Circulars A-11 and A-130 guide agency information technology IT decisions, among other things. The policies in these documents are built around the use of merit-based requirements development and evaluation processes that promote procurement choices based on performance and value, and free of preconceived preferences based on how the technology is developed, licensed or distributed. In the context of developing requirements and planning acquisitions for software, for example, this means, as a general matter, that agencies should analyze alternatives that include proprietary, open source, and mixed source technologies. This allows the Government to pursue the best strategy to meet its particular needs.

Accordingly, as program, IT, acquisition, and other officials work together to develop requirements and plan acquisitions, they should follow technology neutral principles and practices. This means selecting suitable IT on a case-by-case basis to meet the particular operational needs of the agency by considering factors such as performance, cost, security, interoperability, ability to share or re-use, and availability of quality support.

All technology used must be properly licensed in accordance with applicable law.

If you have any questions regarding this memorandum, please contact Thomas L. Stoll, Office of Management and Budget, at (202) 395-1810 or Thomas_L._Stoll@omb.eop.gov.

Figure 4 – OMB Memorandum

CHAPTER FIVE - GOSS FREQUENTLY ASKED QUESTIONS



GOSS frequently asked questions:

How do I become a GAB member? You should establish a formal process for adding new members to your GAB. The OWF GAB indicated at the December 2010 meeting that, in the interest of allowing all interested and committed parties to participate in the GAB, OWF GOSS uses the following approach to determine whether an agency or program should be directly represented on the GAB:

- The existing voting membership of the GAB votes in new members.
- Prospective new members must attend one GAB meeting, as non-voting members, prior to submitting a request to join the Board.
- Prospective new members are expected to be positive ambassadors for the OWF community, familiar with the charter of the GAB, and interested in supporting not only the needs of their own agency or program, but the needs of the community, overall.
- Prospective new members should submit a brief to the GAB describing what their agency is doing or has done with OWF in order to demonstrate OWF's impact on its mission, the impact that the requestor has had or is having in the growth and use of OWF, and/or any contributions the requestor has already made to the OWF community. The GAB will make these briefs available to the OWF community to help foster a collective understanding of what capabilities are being built or reused across agencies and programs.
- Members must be government personnel.
- The Board recognizes that there may be instances in which an agency or program may put forth more than one representative, representing differing interests within that agency or program. The Board will work with the appropriate agency to determine if a single representative can be proffered, and will otherwise handle on a case-by-case basis.

How do I become a GOSS team member? What are the OWF GOSS eligibility criteria?

When you establish your GOSS product, you also need to set eligibility criteria. For OWF GOSS, those criteria are the levels of need-to-know and need-to-use. For example, for an applicant to the Ozone Developers Google Group (closed group to which membership must be granted), the OWF GOSS team required the following information: name, email address, U.S. citizenship declaration, company, agency, and the project on which OWF is to be used. The same set of rules applies for use of the OWF web site to download distribution; the additional request is that the email address for access to the OWF web site must be a corporate address. To access the source code and issues repository (GOSS contributor level account), the designated approving authority further requires identification/verification of a government sponsor, and signature of a usage agreement that indicates what that user can expect of the infrastructure.

What is the process to share your contribution to the GOSS applications? To contribute, aka share back a patch or feature, the individual must be a GOSS contributor. The development group will navigate through a patch process to determine whether to apply the patch and then, integrate it into the work performed by the core development team to apply the patch, as well as make any other necessary changes to the OWF software codebase. Submitted patches are visible to all GOSS contributors, in the issues repository on the OWF GOSS Development Website. They are managed through PATCHES projects (OWF-PATCHES and MP-PATCHES). Groups can download patches and apply them to their local instances, if they desire. The advantage of the core/patch model is that the patch is integrated into the full baseline, including testing the capability and updating documentation, as necessary.

Can I attend an OWF GAB meeting? Interested government and contractor personnel may attend OWF GAB meetings as non-voting (observing) members, providing them the valuable opportunity to observe and understand how a GOSS governance body conducts business.

What Software Development Model (SDM) is used for GOSS products? The SDM used by OWF GOSS is an Agile process, based on Scrum. Scrum is an SDM methodology for incremental project management, often seen in agile software development, a type of software engineering. Although the Scrum approach was originally suggested for managing product development projects, its current use focuses on management of software development projects, and it can also be used to run software maintenance teams or as a general project management approach. Quarterly releases are built out according to pre-determined intervals known as sprints; sprints can be daily, weekly, or monthly. With each sprint, a set of capabilities is built out, tested, and documented. At the end of the release, the team will go through a final code clean-up session. All of this is visible to GOSS contributors; they can see the trouble tickets that have been submitted and track their progress. You should adapt a SDM process that is specific to your agency's needs, mission, personnel, and product.

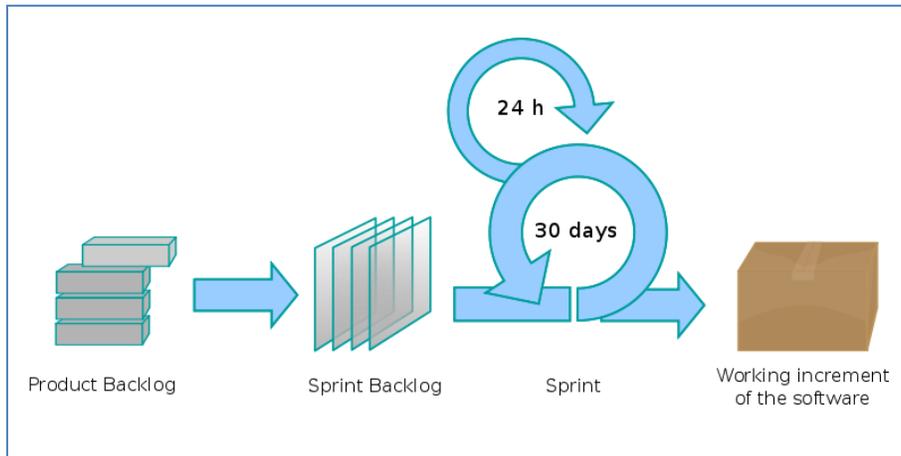


Figure 5 - SDM Scrum Methodology

Where can I find more information about GOSS? The following websites contain additional information about GOSS:

- <http://www.owfgoss.org/>
- http://www.intelink.gov/wiki/OWF_GOSS
- <http://www.mil-oss.org>

APPENDIX A: RELEVANT OPEN SOURCE ACRONYM LIST:

ABC	Adopt before Buy, Buy before Create
AECA	Arms Export Control Act
ARPANET	Advanced Research Projects Agency Network
ASCII	American Standard Code for Information Interchange
ASEC	Architecture, Standards and Engineering Committee
CIO	Chief Information Officer
COI	Community of Interest
COTS	Common (Commercial) Off The Shelf
DNS	Domain Name Service
DOD	Department of Defense
EEE	Embrace, Extend and Extinguish
FOSS	Free/Open Source Software
FSF	Free Software Foundation
FSG	Free Standards Group
FUD	Fear, Uncertainty, Doubt
GAB	GOSS Advisory Board
G2B	Government to Business
GOSS	Government Open Source Software
GOTS	Government Off The Shelf Software
GSM	Global System for Mobile Communications
GUI	Graphics User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IC	Intelligence Community
ICEE	Intelligence Community Enterprise Engineering
IA	Information Assurance
IANA	Internet Assigned Numbers Authority
ICT	Information and Communications Technology
IEEE	Institute of Electrical and Electronics Engineers

IMAP Internet Message Access Protocol

IOSN International Open Source Network

IP Internet Protocol

IPR Intellectual Property Rights

IS Information System

ISO International Organization for Standardization

ISV Independent Software Vendor

IT Information Technology

ITAR International Traffic in Arms Regulations

ITU International Telecommunication Union

ITU-D International Telecommunication Union - Development Sector

ITU-R International Telecommunication Union - Radio Communication Sector

ITU-T International Telecommunication Union - Telecommunication Standardization Sector

JSEC Joint Systems Engineering Committee

JTC Joint Technical Committee

LAN Local Area Network

LDAP Lightweight Directory Access Protocol

LSB Linux Standard Base

NCSA National Center for Supercomputing Applications

NFS Network File System

OASIS Organization for the Advancement of Structured Information Standards

OMB Office of Management and Budget

ORG Organization

OS Operating System

OSDL Open Source Development Labs

OSI Open Source Initiative

OSS Open Source Software

OTD Open Technology Development

OWF Ozone Widget Framework

PC Personal Computer

PDA Personal Digital Assistant
PDF Portable Document Format
RFC Request for Change
RTF Rich Text Format
SDM Software Development Model
SMTP Simple Mail Transfer Protocol
SOAP Simple Object Access Protocol
SQL Structured Query Language
SSL Secure Sockets Layer
SSO Standard-Setting Organization
STD Standard
SVG Scalable Vector Graphics
SVID System V Interface Definition
STARS Simple, Timely, Agile, Robust Self-Service
TCP Transmission Control Protocol
TEM Technical Exchange Meeting
USML United States Munitions List
VHS Video Home System
Wi-Fi Wireless Fidelity
WWW World Wide Web
XML Extensible Markup Language

