White Paper

Weathering the Economic Crisis in Engineering

Top 25 Open Source Projects That Will Help Trim Development Budgets*

Theresa Bui Friday

December 2008

www.palamida.com

*This paper discusses the 25 top Open Source projects companies are using for cost control and competitiveness.

OVERVIEW

In challenging economic times, how do internal application development teams continue to deliver higher quality software and Web applications with fewer resources? Unlike in past economic downturns, development teams today have a resource they can turn to in order to lower the costs of development, maintain high-quality, and decrease cost of ownership for the long run: open source software.

The use of open source, one of the most groundbreaking trends in the software industry, is more than just for experimental or for internal-use only. With experience in auditing billions of lines of code for Fortune 100 as well as start-up companies, Palamida has seen some of the most productive and cost-saving use of open source from market leaders across all industries. This paper will list the top 25 open source projects reviewed by Palamida that have proven to be among the most reliable, innovative, and enterprise-ready open source projects available on the market.

This paper is for senior engineering and IT executives who are looking for resourceful ways to trim budgets, while ensuring that application development work continues to be a competitive advantage for their businesses. It will list some of the best open source projects Palamida has seen used inside organizations of all sizes. As well, it will provide senior managers with best practices basics in defining an effective open source usage strategy, while ensuring the integrity and security of their applications and their business model.

So now it's your move. If your team is not already using these open source projects, you should be sitting down with your lead engineers to review this list today.

25 HOT OPEN SOURCE PROJECTS ORGANIZATIONS SHOULD BE USING TODAY

Development Tools

There is no question that a good toolkit consisting of an integrated development environment, unit testing, code coverage and code quality will enhance productivity and overall application excellence.

Project Name	Description	Overview	Cost to Develop In-

			house[1]
NetBeans	IDE providing developers with all the tools they need to create professional cross-platform desktop, enterprise, web and mobile applications, with support for Java, JavaScript, C, and C++. Runs on many platforms including Windows, Linux, Solaris, and the MacOS. It is easy to install and use straight out of the box.	Established in 2000, with sponsorship from Sun Microsystems. Maintains an active developer community of over 500 contributors.	235 person years or $12,946,401
Eclipse	Complete development platform comprising extensible application frameworks, tools and a runtime library for software development and management. It is written primarily in Java to provide software developers and administrators an integrated development environment (IDE).	Established in 2001 with sponsorship from IBM and other vendors. Maintains an active developer community of over 200 contributors.	2,264 person years or $12,501,084
JUnit	Java testing framework that enables developers to cheaply and incrementally build a test suite that will help them measure their	Established in 2002 with sponsorship from Object Mentor. Maintains active developer community with 7 contributors.	4 person years or $213,508

[1] There are several online calculators that evaluate the return on investment of open source use. One such calculator, Ohloh.net, uses the Constructive Cost Model (COCOMO), an algorithmic model that uses a basic regression formula, with parameters that are derived from historical development data and current software project characteristics. The Ohloh calculator is an attempt to estimate how much it would cost to create a project from scratch. It is not capable of telling what a project is "worth," only what it would cost to produce. It assumes an average developer salary of $55,000/year, which may be less than half of the actual cost in high tech areas such as Silicon Valley, Boston, New York, etc.

	progress, spot unintended side effects, and focus their development efforts.		
httpunit	Automated web testing framework that emulates the relevant portions of browser behavior, including form submission, Javascript, basic HTTP authentication, cookies, and automatic page redirection, and allows Java test code to examine returned pages either as text, an XML DOM, or containers of forms, tables, and links. When combined with a framework such as JUnit, it is fairly easy to write tests that very quickly verify the functioning of a Web site.	One dedicated project maintainer since 2003, with over 40 contributors.	103 person years or $5,659,573
PMD	Java source code analyzer. It finds unused variables, empty catch blocks, unnecessary object creation, and more. It includes CPD, a tool to detect chunks of identical code.	Dedicated community of 16 team members, established since 2002.	66 person years or $3,642,640
Valgrind	Suite of tools for debugging and profiling Linux programs. Users can automatically detect many memory management and threading bugs,	Dedicated community of 11 team members, established since 2003.	60 person years or $3,307,271

	avoiding hours of frustrating bug-hunting, making programs more stable. Users can also perform detailed profiling, to speed up and reduce memory use inside applications.		
FindBugs	Looks for bugs in Java programs. It can detect a variety of common coding mistakes, including thread synchronization problems, misuse of API methods, etc.	Dedicated community of 9 team members, established since 2003, with sponsorship from both Google and Sun Microsystems.	45 person years or $2,457,935

Database and Mapping Tools

Database buying patterns have shifted significantly in the past few years, with a sharp focus on cost-effectiveness. Open source database solutions can now tout both speed and the ability to handle very demanding processing tasks.

Project Name	Description	Overview	Cost to Develop In-house
Hibernate	A powerful, high performance object/relational persistence and query service for Java. It lets engineers develop persistent objects following common Java idiom, including composition, association, inheritance, polymorphism, and the Java collections	Dedicated community of 16 team members, established since 2002.	233 person years or $12,813,393

	framework.		
SQLite	Small C library that implements a self-contained, embeddable, zero-configuration SQL database engine. Implements most of SQL92 and ACID (atomic, consistent, isolated, and durable) transactions; no setup or administration needed.	Dedicated community of 25 contributors, established since 2000.	29 person years or $1,606,773
MySQL	A relational database management system with more than 11 million installations.	MySQL is owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now a subsidiary of Sun Microsystems, which holds the copyright to most of the code base	unavailable
Apache Derby	A relational database implemented entirely in Java. Some key advantages include a small footprint (about 2 megabytes for the base engine and embedded JDBC driver) and being based on the Java, JDBC, and SQL standards.	Dedicated community of over 32 contributors, established since 2005.	166 person years or $9,086,983
PostgreSQL	A relational database system that has earned a strong reputation for reliability, data integrity, and correctness. It runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI	Dedicated community of over 30 contributors, established since 1996.	146 person years or $8,039,337

	IRIX, Mac OS X, Solaris, Tru64), and Windows		

Core Utility Classes

Utility classes are programming libraries designed to perform common, often-used functions. This has been one of the most popular and earliest implementations of open source software inside organizations.

Project Name	Description	Overview	Cost to Develop In-house
zlib	Designed to be a free, general-purpose, lossless data-compression library for use on virtually any computer hardware and operating system.	Dedicated community started by Mark Adler and Jean-loup Gailly in 1997. Stable version v1.2.3 since July 2006.	unavailable
libpng	The official PNG reference library. It supports almost all PNG features, is extensible, and has been extensively tested for over 12 years.	One dedicated project maintainer, with 8 contributors, since 1996.	unavailable
FFmpeg	A complete solution to record, convert and stream audio and video.	Dedicated community of almost 80 contributors, established since 2000.	81 person years or $4,439,936
Freetype	Software font engine that is designed to be small, efficient, highly customizable and portable while capable of producing high-quality output (glyph images). It can be	Dedicated community of almost 30 contributors, established since 1997.	64 person years or $3,526,986

	used in graphics libraries, display servers, font conversion tools, text image generation tools, and many other products as well.		

Reporting and Charts

Reporting and charting solutions have really caught up with the major players and offer a new level of sophistication. As they continue to implement most of the features available from the biggest commercial rivals, these open source solutions will give corporate IT managers a good reason to evaluate them.

Project Name	Description	Overview	Cost to Develop In-house
JFreeChart	Chart library for the Java platform that supports a wide range of charts including pie charts (2D and 3D), bar charts (horizontal and vertical, regular or stacked, with optional 3D-effects), line charts, XY plots, scatter plots, time series charts, high/low/open/close charts, candlestick plots, Gantt charts, Pareto charts, combination charts, and more.	Founded in 2004, with one dedicated project maintainer, with four contributors and a user base of almost 50,000 developers.	43 person years or $2,338,036
Velocity	A simple yet powerful Java-based template engine that renders data from plain Java objects to text, xml, email, SQL, Post	Dedicated community of 9 contributors established in 2002 and part of the larger Apache Software Foundation.	13 person years or $701,339

	Script, and HTML etc. The template syntax and rendering engine are both easy to understand and quick to learn and implement.		
Pentaho Reporting	A class library for generating reports. XML-based templates provide flexible reporting and printing functionality using data from multiple sources. It supports output to display devices, printers, PDF, Excel, HTML, XHTML, PlainText, XML and CSV files.	Established in 2002 with almost 20 contributors.	444 person years or $24,425,468
JasperReports	Java reporting library. XML report templates are used to generate ready to print documents using data from customizable data sources, including JDBC. The output can be delivered to the screen, printer, or stored in PDF, HTML, XLS, RTF, ODT, CSV, TXT and XML format.	Established in 2005 with 15 contributors and part of the JasperForge.	63 person years or $3,486,668

Web 2.0

It can be argued that Web 2.0 has been built on the back of open source. Some of the most popular Web 2.0 companies -- Google, Facebook, Flickr, etc. – are all built using open source technologies. So it is no surprise that some of the best open source projects on the market today support major Web 2.0 functionality.

Project Name	Description	Overview	Cost to Develop In-house
Prototype	A JavaScript framework that aims to ease development of dynamic web applications. Featuring a unique, easy-to-use toolkit for class-driven development and the nicest Ajax library around.	Established in 2005, with a dedicated community of 11 engineers. Major users include NBC, CNN, Apple and amazon.com.	2 person years or $127,497
script.aculo.us	Provides easy-to-use, cross-browser user interface JavaScript libraries to make web sites and web applications fly.	Established in 2005, with a dedicated community of 7 engineers. Major users include NASA, Apple and CNN.	3 person years or $169,895
Direct Web Remoting	A simple servlet plug-in that allows users to expose selected Java classes via JavaScript. DWR makes calling Java code directly from a web form simple. DWR can make writing interactive DHTML pages (like GMail) very much simpler.	Established in 2005, with 10 contributors.	50 person years or $2,762,729
Yahoo! User Interface	A set of utilities and controls, written in JavaScript, for building richly interactive web applications using techniques such as DOM scripting, DHTML and AJAX.	Part of the larger The Yahoo! Developer Network, which offers Web Services and APIs that make it easy for developers to build applications and mashups that integrate data sources in new ways.	unavailable
jQuery	A fast, concise, JavaScript library that simplifies how users traverse HTML	Founded in 2005, with almost 60 contributors.	74 person years or $4,071,203

	documents, handle events, perform animations, and add Ajax interactions to Web pages.		

CRITICAL ISSUES WHEN DEFINING AN EFFECTIVE OPEN SOURCE STRATEGY

Whether it is these 25 open source projects or others, most organizations are already using open source as part of their IT infrastructure or inside their shipping product and Web applications -- whether senior managers realize it or not.

For open source used in software or Web projects, Palamida has found that applications built in the last five years will typically be composed of at least 50% open source code. As much as 70% of that open source use is invisible to senior managers – which means that it has likely not been given a security review and security updates are not being patched in to protect the company and its customers. This growing void in application security leaves organizations open to the risk of introducing vulnerabilities through undocumented code.

Traditional application security solutions such as intrusion detection, ID management, and firewalls, are critical for securing traffic *to* applications. But Gartner, Inc. research shows that since 2002, 70% of successful security attacks exploit *application* vulnerabilities – issues with specification, design or implementation once the traffic has arrived. Managers need to realize that what is inside their applications can be just as harmful as what is coming to their applications. Organizations need a new layer of application security that allows them to protect themselves against vulnerabilities in application code even if they do not know what they are using.

Gartner, Inc. has identified this new layer of security as “software composition analysis” (SCA) – technologies that should be used along with static application security testing (SAST) and dynamic application security testing (DAST). SAST/DAST inspect applications internally, while SCA classifies

external components by name, version and associated vulnerabilities[2]. Considering that only 1 out of every 10 open source project has commercial support[3], organizations must realize that a management system for vulnerability review, tracking and patch/remediation of open source is now a mandatory part of an application security strategy.

Often times, the philosophy of "spend small, think small" prevails for most IT organizations. Unless an organization is adopting a large open source project such as Linux, special resources are not being allotted to the management of open source adoption. Organizations now can implement a policy of "spend small" but just as with any other type of third-party software, they cannot afford to "think small" about what they are using, meaning they cannot stop constantly monitoring for new security information about it. In the past, with commercial software packages, if developers wanted to incorporate third-party code into their applications, a joint development agreement or in-bound licensing contract would be negotiated. The process would have also included a development manager, procurement lead, and a lawyer.

In "User Survey Analysis: Open Source Software, Worldwide, 2008," Gartner, Inc. reports that almost 40% of organizations cite "lack of governance policies regarding open source adoption" as one the key organizational challenges in using open source. In today's 24/7 world of persistent network access, developers dispersed across multi-national sites can include open source, freeware, public domain, evalware (demos of commercial software), etc., into the code they are writing without triggering the usual checkpoints in the procurement process. Without these controls, the open source is unlikely to be detected, monitored, or tracked.

Best Practices Basics for Open Source Management

Organizations that employ best practices in open source software management maximize the benefits of open source and minimize any security or operational risks. A best practices management workflow does not have to be disruptive, once policy and procedures are established. Each organization's own implementation will be different, depending on size and business model, but will always contain the following stages: assessment, policy, open source repository, code audit, and ongoing management.

Assessment

A process must be in place for assessment and registration of open source introduced into the code base by individual developers. An organization should establish an Open Source Review Board,

[2] "*Gartner Hype Cycle for Data and Application Security 2008*," 30 September 200, ID Number: G00160731

[3] Based on Palamida research conducted February 29, 2008 - March 4, 2008, examining support structure for 3,168 popular open source projects

which in small companies may consist of one person, while in larger ones, it might consist of multiple members representing cross-functional roles. In smaller companies, requests or notifications of intended use may be simply captured through emails between developers and managers and updating Excel spreadsheets. In medium to large companies, a robust web-enabled request system is needed. The best system is one that enables auto-approvals based on defined criteria, and brings only the exceptional cases to the attention of authorizing personnel.

Policy

An open source policy is critical to any open source management process. It defines the criteria for allowable open source use. Policies can range from broad and simple to more granular and tightly managed; policies are typically dependent on an organization's business and software distribution model. Some organizations may employ a policy for liberal open source use, only specifying that developers use the latest version of an open source project. Other organizations may only allow those open source projects that have passed a detailed security review. Still others may be more concerned about intellectual property issues, and may employ an open source policy that is dependent on licenses and license terms. If organizations are using a web-enabled system that enables auto-approvals based on policy, they should ensure that their policy sufficiently handles at least 80% of open source software use cases, so that the Open Source Review Board reviews only exceptions.

Open Source Repository

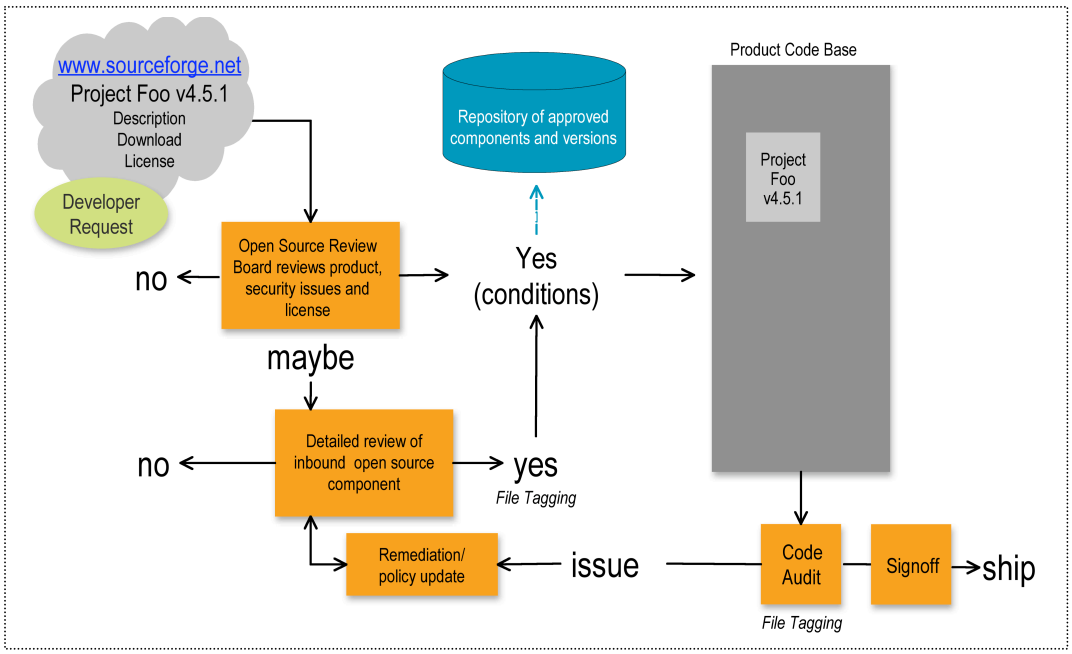
As organizations take time to review and approve open source projects and license terms and obligations, they can establish a "gold vault" of versions approved for use. The existence of an open source repository greatly reduces the risk of including outdated, vulnerable or unstable project versions. It also makes it easier for security and management teams to track and monitor patch, remediation and version updates.

Code Audit

Code audits are essential for ensuring adherence to policy. Audits can be done manually through simple string search or with automated tools triggered by build systems. Depending on the size of the organization, code audits can be conducted continually during the development and QA cycle or at a specific checkpoint stages. Organizations should ensure that their chosen audit methodology enables identification via source or binary code to ensure they do not miss any open source in use. The most important result of the code audit should be a clear, concise inventory listing all open source software in use, version, description, and location in the code base. The inventory should reveal that all open source is in compliance with corporate policy and alert users to any open source that is not.

Ongoing Management After Shipment or Deployment

Good decision-making and risk management depends on the ready availability of good information. Effective open source management requires tracking open source components, their attributes (including their versions and download origin), license terms and compliance, their owners within the organization, and where they are used. In addition, a good management process can document the initial decision approving use, any modifications to the project, and its maintenance history. Most important, however, is the ongoing ability of security professionals and managers to receive new security vulnerability alerts regarding open source projects in use. The smallest organizations may find that manually monitoring various open source community pages may be sufficient. Most organizations, however, are already using dozens to hundreds of open source projects. Since they cannot manually monitor hundreds of open source community sites, these organizations need to rely on an automated system that will push them new security information and patch and update alerts.



Best Open Source Management Workflow

Summary

Because of its self-service support and maintenance conventions and informal procurement process, open source software often requires different management techniques than commercial software. Organizations that manage their use of open source software realize substantial productivity benefits, as seen by the popularity and usefulness of the Top 25 open source projects. Those organizations that do not, however, incur operational, financial and legal risks.

Companies employing best practices in open source management track what open source they are using, where they are using it, and how they are maintaining it. They do this by employing an articulated strategy, a clear and concise usage policy, and an efficient process for ongoing management.

About Palamida, Inc.

Palamida provides the industry's first application security solution exclusively for open source software. Palamida's Enterprise Edition uses component-level analysis to quickly identify and track undocumented code and associated security vulnerabilities, as well as intellectual property and compliance issues. Using Palamida, organizations can cost-effectively manage and secure mission critical Web and software applications. Customers include Avaya, Cisco Systems, EMC, Microsoft, and Sun Microsystems, among others.

For more information visit: www.palamida.com.