PUTTING OPEN SOURCE TO WORK IN THE ENTERPRISE: A GUIDE TO RISKS AND OPPORTUNITIES

CONTENTS

Open source software has never been more ready for use in the enterprise. With Linux leading the way, a diverse parade of software has marched from the open source community Web sites directly into the largest data centers in the world. Today companies like Novell, IBM, Hewlett-Packard (HP), SAP, and others support Linux and most common open source software as an ordinary course of business. Developers and end users alike have many attractive open source programs to meet their needs. Now the key question on the mind of most IT professionals is this: What am I getting myself into when I use open source software?

Unfortunately, there is no simple answer to this question. The right answer for your company depends on choices you must make about the level of skill that you want to maintain and the models for support that you are comfortable with. When open source first became credible for enterprise use, many companies were uncomfortable about the unique features of open source licenses. These fears, promoted in some cases by opponents of open source, have been largely overcome.

In most industries, waiting is no longer an option. Ignoring the opportunity to incorporate open source into your environment to enhance commercial offerings, to reduce costs, and to gain access to valuable new tools and applications means leaving money on the table. In this booklet, we describe practical ways you can put open source to work. However, rushing ahead and installing any sort of open source software without a coherent strategy and plan to manage the risks and responsibilities is not a viable option either. Use of open source software is not an all-or-nothing proposition. There is a continuum of adoption from using as much as possible to using only the most mature forms of open source to using none at all. For most companies, success in using open source means combining elements of commercial enterprise software and the most mature open source software to create an environment that offers the best of both worlds with lower costs than any other path. Of course, chances are that your organization is already relying on open source software for parts of your IT infrastructure or applications.

From a strategic perspective, utilizing open source provides an opportunity to free up money for innovation and differentiation. Bruce Perens, a pioneer of open source who wrote one of the early definitions of the concept, put it this way: "… to make your business more desirable to customers, you should spend more on differentiating software that makes your business more desirable. … Open source is the key to spending less on nondifferentiators."[1]

---

1. Bruce Perens, "The Emerging Economic Paradigm of Open Source" (Cyber Security Policy Research Institute, George Washington University, 2005), http://perens.com/Articles/Economic.html.

To successfully use open source, you must become skilled at understanding the productization gap for the software you intend to use and the way the open source functionality will fit into your existing environment. The size of the gap differs from project to project. It is small in the most mature projects like Linux for which there are lots of companies selling consulting and support services to close the gap. Linux is productized, for instance, by Novell and Red Hat with enterprise-level support, documentation, and services, just as customers would expect from non–open source software companies. Open source software makes economic sense because the cost of closing the gap is lower than the costs of commercial alternatives. For most open source software, however, you must take responsibility for closing that gap either by developing and maintaining skills internally or by purchasing services from a third-party provider. This booklet is a guide to understanding and closing the productization gap so that you can successfully put open source to work.

The key to making open source work is understanding that in most cases open source software has powerful functionality that is only partially productized, meaning that all of the mechanisms provided by enterprise software vendors such as documentation, installation scripts, configuration mechanisms, and support services may or may not exist. We call this difference the productization gap.[2]

---

2. The idea of the productization gap was introduced in *Open Source for the Enterprise* by Dan Woods and Gautam Guliani (O'Reilly Media, 2005).

QUESTIONS ANSWERED IN THIS BOOKLET

How can I put open source to work? What strategies will work best?
This booklet aims to prepare you to recognize the different types of open source projects in the marketplace so that you can evaluate whether they are ready for enterprise use at your company. To do this, we cover all the questions below and then detail the specific ways of using open source that are likely to be most beneficial.

See "Putting Open Source to Work," a section that identifies specific opportunities.

What is open source and where did it come from?
We take a condensed tour of the history of open source, from its roots in the free software movement to the first large-scale successful projects like Apache and Linux to commercial conversions like Eclipse and youthful enterprise applications and service offerings like SugarCRM and SpikeSource.

See "What Is Open Source?"

How is open source different?
We analyze the unique character of open source by looking at its progression from the first generation, which was created by communities of developers for their own use, to each successive generation of open source, which has improved the productization. Despite this progress, open source remains fundamentally different from commercial enterprise software.

See "How Is Open Source Different?"

How is open source being productized?
We examine the many forms of open source productization. Fundamental infrastructure has been productized through distribution services such as those available for Linux. Some commercial software that had been fully productized has been converted to open source. Some open source projects have been converted into fully realized products. Services have been created to supply productized support for integrated stacks. Understanding the level of productization of an open source project is the key to rapid assessment of its potential value to you.

See "The Productization Gap in Open Source Software."

How can open source be applied?
The amount of fully productized open source available is greater than ever before. We make specific recommendations for applying open source in the following three categories:
- Infrastructure and operations
- Software development
- Applications

We review the most common use cases, the level of productization of the software, and the business value.

See "Three Categories of Opportunities."

THE DIMENSIONS OF AN OPEN SOURCE ADOPTION STRATEGY

As you read through the rest of this booklet, keep in mind the dimensions shown in Figure 1, which are expressed as slide controls from one extreme to the other. The goal is to help you answer the questions related to each dimension. We will return to them later when we discuss various strategies for open source adoption.

Maturity of an Open Source Project:
How well does the open source software fit your problem?
Is the software stable?

Youthful Functionality ⟞——————[?]——————⟝ Mature Functionality

Productization of an Open Source Project:
How completely productized is the open source software?
Are productization services available to close the gap?

Raw Functionality ⟞——————[?]——————⟝ Productized Functionality

Internal Skills Versus Support Services:
What is your ability and desire to maintain skills
to overcome the productization gap?

Maintain High Level of Skills ⟞——————[?]——————⟝ Use Productization Services
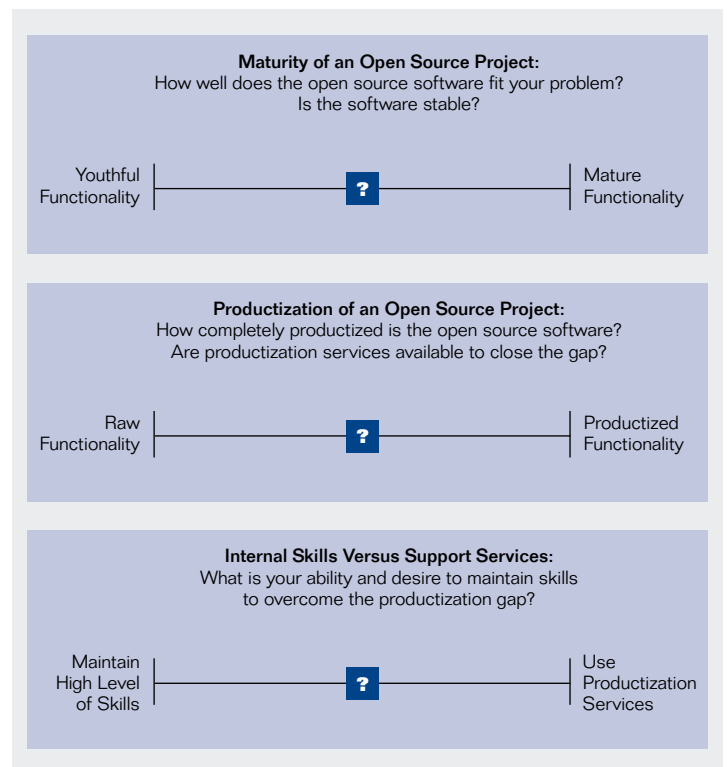
Figure 1: The Dimensions of an Open Source Adoption Strategy

WHAT IS OPEN SOURCE?

The world of open source has grown so rapidly and undergone so many evolutions that the phrase itself has almost lost its meaning. Is open source the work of the Apache Foundation, which created the most popular HTTP server in the world based on the original National Center for Supercomputing Applications (NCSA) server? Is open source the Firefox browser, which was created by Mozilla.org, which was spun out of Netscape? Is open source SUSE Linux Enterprise, the Linux distribution from Novell, or the MaxDB database converted to open source by SAP, or the development tools from Eclipse.org that originated with IBM? Is open source the enterprise software created by SugarCRM and Alfresco? Is open source the certified stacks of many open source components offered by SpikeSource and SourceLabs?

The reason that the world of open source is so confusing is that all the projects just mentioned claim to be open source, but each one uses the fundamental structure of open source in a different way.

Oddly enough, open source was not called open source when it was first invented. It was called free software, and others point to even earlier efforts at sharing software and ideas as the roots of open source. Richard Stallman, founder of the GNU Project and the Free Software Foundation, got the ball rolling by creating a set of development tools that are still the foundation of a huge amount of open source. Perhaps more importantly, the Free Software Foundation also created the GNU General Public License (GNU GPL), also known as copyleft, a software license that specifies that the source code for the software used under it must always be available if the software is distributed to others. The GNU GPL, along with many other similar licenses for open source from the Apache Foundation, Mozilla.org, universities, and other organizations, produced a huge explosion of activity, creating the world of open source as we know it today.

The term open source came into use because some people felt that Stallman's views and his leadership style were driving people away from using free software. In 1998, a group of people, including Eric Raymond, Tim O'Reilly, and several others came together and started using the term open source instead of free software. This took hold, although many people use the terms separately or use the combined term free and open source software, abbreviated as FOSS. As a practical matter, the meaning of open source really depends on who you are and why you are using it.

PERSPECTIVES ON OPEN SOURCE

Open source has become such a large world with so many projects and players that the most useful definition may be one that applies on the basis of the way that open source is being used. Your organization may have some or all of the following types of users for which open source would have different meanings:

Users of software
For users of software who have the skills to download and install software, open source means choice and freedom. The choice comes from the huge array of options like Linux, Firefox, or OpenOffice.org that are available. The freedom comes from the fact that you can choose how you want to use the software. You can compile it yourself, which few do, use bare-bones distributions, or use fully productized suites of products based on open source.

Developers and engineers
For developers and engineers, open source has many additional meanings. To the founders of a successful project, open source can mean fame, recognition, and sometimes even money from consulting or other sources. Other developers see open source as a masterful software development methodology founded on the virtues of collaboration, incremental evolution, and working code. For most developers, open source is both a source of tools to help solve problems and a constant source of exciting new things to learn.

Enterprises and IT professionals
For enterprises and IT professionals, open source provides all the benefits it offers to developers as well as a multitude of business advantages. By using open source, IT departments can save money or support their businesses better through access to huge amounts of powerful software. Open source helps businesses cost-effectively support infrastructure and operations, develop software, and deploy enterprise applications. Large companies with the most demanding needs for high performance make extensive use of open source software.

Commercial software vendors
Commercial software vendors use open source in many different ways. Open source projects are used as components in commercial products. Open source is used for integration and for development tools. Open source development methods are imitated in the creation of communities and the release of source code in various ways. Commercial vendors sometimes donate proprietary software into the public domain as open source to promote the sale of services, create a friendly environment for their products, or challenge a competitor by creating a free alternative.

Systems integrators and services vendors
Systems integrators and other companies selling services use open source as a way to generate demand for services. Some companies like Optaros specialize in building applications using open source. Others, such as Novell and Red Hat, sell open source packaging and distribution subscriptions. Other companies sell packages of open source software that have been preintegrated and tested to work together.

WHERE DOES OPEN SOURCE COME FROM?

When evaluating whether an open source project is right for you, it is crucial to understand the origin of the project. Projects that were formed from open source communities (community-created) may have thriving groups of developers, but the projects may lack documentation, installation scripts, and other features that early adopters typically could get along without. Projects that were open sourced from commercial roots (commercial conversions) may have smaller, less viable communities, but those projects are more likely to come with additional features, such as documentation and installation scripts. Projects that use open source as a marketing vehicle (commercial open source) may only offer open source versions with limited features or may require fees for items, such as documentation or full-feature versions. The manner in which open source is created determines the strengths and weaknesses as shown in the following table.

Origin	Examples	Strengths	Weaknesses
Community-created open source projects	Apache, Linux, GNU Project, Perl, Python, Drupal, TWiki	They frequently have strong communities of developers. Requirements are proven through years of use and experimentation. An eco-system has grown organically to meet needs. Many organizations may invest in continued development and support.	They may lack productization, although this is not typically important to early adopters.
Commercial open source projects	JBoss, MySQL, SugarCRM, Alfresco	Commercial software vendors create software using a hybrid of enterprise software design based on formal requirements gathered from users combined with community involvement. Community-supported open source versions are used for marketing and testing.	Productization and a support ecosystem are available for a fee, but the level of maturity of productization varies considerably. The developer community may be smaller and less vibrant compared with a community-created open source project.
Commercial conversions to open source projects	Eclipse, NetBeans, XFS, Ingres	These projects have a head start on producti-zation, breadth of features, and configurability. Integration into common enterprise infrastructure may be stronger than for community-created projects.	A true community may never form to support the converted open source software. Commer-cial conversions can end up as communities, as commercial open source, or as both.

HOW IS OPEN SOURCE DIFFERENT?

The world of open source is a large umbrella that includes a wide range of software created in many different ways by different sorts of people for different reasons. Because of diverse origins, significant and persistent differences exist between open source software and commercial enterprise software. You must take into account the following information about open source when designing an optimized infrastructure that combines both types of software:

- **The source code is available.**

  While commercial software vendors and their customers engage in elaborate contract negotiations about how to maintain escrows of source code to protect the customer if the vendor goes out of business, in the open source world a complete source code version of the software is available from day one. While this provides a great sense of comfort and a massive opportunity for innovation, the reality is that in order to take advantage of the availability of source code, you must have and maintain the necessary development skills. This level of skill is one reason that raw open source tends to be used most by highly skilled organizations. For most companies, the availability of source code is great insurance against the worst-case scenario.

- **It is free to download, but support must be paid for.**

  Perhaps the most commonly understood aspect of open source that distinguishes it from enterprise software is that open source is free to download and install. The benefits of this are quite profound. You can install open source on as many machines as you want without any extra license fees. If you want to try an open source program, you can do so without having to sign a trial agreement. Of course, using open source requires that you either have the skills to support it or can acquire support from some other source – usually for a fee.

- **It is made by developers and for developers.**

  For most of the community-created open source, the developers are the users of the software, which is usually created to serve developers' needs (Linux, Apache, MySQL, Perl, and so on). The requirements process consists of developers thinking about what they want the software to do, discussing it with the community, and then writing the code. The advantage of this process is that requirements emerge in an evolutionary fashion, based on the need and the intensity of interest. For most enterprise software, the developers are not the users, and they must gather and evaluate requirements in a more formal manner. Although this process sometimes misses the mark, it facilitates the creation of software for users who are not developers.

- **The productization gap varies.**

  The most important aspect of open source as it relates to enterprise use is that, except for the most mature and thriving open source projects, open source software is rarely fully productized. The evolutionary, developer-driven requirements process leaves gaps in areas that are not interesting to developers, such as installation scripts, documentation, integration adapters, and other features that fully productized software has. For the most popular open source projects, the community provides all of this, but for most projects, productization or feature gaps are common. It is no accident that the heaviest users of open source are the most highly skilled engineering organizations, the early adopters and innovators, to use Geoffrey Moore's classification from *Crossing the Chasm* (Harper Collins Publishers, 1999). For them, the skill barrier does not exist, and open source is easy to use.

> While the freedom of open source is a great comfort and provides significant benefits, as a practical matter, to use open source in an enterprise context, you must either develop the skills to be able to solve problems yourself or identify a means to overcome the gaps in productization by getting help from someone else. If this were easy, there would not be a business in productizing open source. The license and support fees in the enterprise model are mirrored by substantially lower levels of service fees for productization and support when using open source.

THE PRODUCTIZATION GAP IN OPEN SOURCE SOFTWARE

The open source opportunity is managed differently by the various segments of technology adopters on the basis of their skill levels.

| Innovators | Early Adopters | Early Majority | Late Majority | Pragmatists |

While innovators and early adopters develop skills to use raw and youthful open source projects when the payoff is large, they use productized offerings to reduce costs.

The early and late majority can use raw and youthful open source projects with help from service providers; however, most of their opportunity is found in productized offerings.

Pragmatists use open source only when it is fully productized, usually when it is embedded in other offerings.

Figure 2: Open Source Adoption by *Crossing the Chasm* Categories

Figure 2 helps explain the current state of open source adoption. At first, open source was created by innovators and early adopters to meet their needs. All the early open source projects focused on infrastructure and developers' tools of interest. As time went on, various companies formed to provide business models and services to bridge the productization gap for community-created open source. All commercial open source ventures are aimed in one way or another at bridging the productization gap so that the early and late majority can safely use open source.

The Productization of Open Source

Open source has been productized in several waves; each one of the following waves has taken a different approach on the basis of the problem that needed to be solved at the time:

- **Packaged distributions**

    The first wave of productization involved making the core community-created infrastructures like Linux and Apache easier to use in an IT environment. Companies like Novell and Red Hat packaged Linux into distributions that provided the missing productization, such as installation scripts, easier configuration, documentation, and online support. Companies could purchase subscriptions to distributions so that updates and fixes for various problems would arrive in a reliable stream.

- **Open source–based productized software**

    The second wave of productization took the form of products that were created as open source but were intended to be commercial enterprises from the beginning. Products such as Stronghold were secure versions of the Apache server that helped support its usage in e-commerce and other environments where security was important. JBoss created a suite of open source Java infrastructures that was available as open source but had elements that completed the productization, such as documentation and support that could be purchased. SugarCRM created a customer relationship management (CRM) system based on PHP that had an open source version available free of charge and a premium version with advanced features available for a fee. MySQL is available as an open source download or as a fully supported commercial product. Although vibrant communities formed around many of these products, most of the development was performed by a core team of employees who worked at the founding company. People claim that Linux has become this sort of project, since most of the people who work on the code are full-time employees of companies involved in commercializing the software.

- **Open source support and services companies**

    Another wave of youthful service companies like SpikeSource and SourceLabs sell support, consulting services, and subscriptions to certified stacks in various combinations, along with documentation. These companies are attempting to productize the significant effort required to make Linux, Apache, MySQL, and many other programs work together reliably and scale under load. For example, Linux comes in many flavors, each with multiple versions, and keeping track and supporting all of them is not a trivial task. Other companies offer support for specific open source projects. Companies like Optaros offer systems integration services based on using open source as a foundation for development, as does major solutions vendor HP, which offers a multiplatform, heterogeneous approach that incorporates validated open source and commercial software, depending on customer need.

Many other forms of open source productization and commercialization are also underway. Publishers offer books that provide missing documentation. Large companies like Yahoo!, Google, and Amazon have created online services on the basis of open source that are forms of open source productization.

Forms of Open Source Productization

Open source productization also takes several forms. The offerings and the cost vary, depending on the type of form, as you see in the following table.

Productization Offer	What You Pay For	Example Offerings
Subscriptions to packaged distributions	Installation scripts, documentation, support in various forms, updates, and patches	SUSE Linux Enterprise, Red Hat, Red Flag Linux, Ubuntu, Debian
Enterprise applications	Productized enterprise applications with all traditional forms of support	SugarCRM, JBoss, Alfresco, WebSphere Application Server, Community Edition (from IBM)
Appliances	Hardware that was created for a special purpose based on an open source foundation – frequently a form of Linux embedded in a device	TiVo, the SAP NetWeaver® Business Intelligence Accelerator software
Hosting packages	Preconfigured open source environments with a specific application or collection of applications designed to solve a specific problem	Web site–hosting companies, content management hosting, Wiki and blog hosting
Certified stacks	Distribution of combinations of open source that have been tested and optimized to work together; documentation, support, and consulting services	SpikeSource, SourceLabs, HP
Consulting services	Systems integration services based on using open source as a foundation	Optaros, many other large and small integrators
Online services	Products offered over the Internet that make heavy use of open source in their infrastructure	Yahoo!, Google, Amazon, eBay

PUTTING OPEN SOURCE TO WORK

What factors should determine your open source strategy? Let's say your company is populated with the best engineers in the world, and you have a profitable line of business to support massive R & D spending. Using the three dimensions mentioned earlier, a reasonable strategy might appear this way:

Perhaps your company is a start-up company that has highly skilled engineers, and you want to use open source for corporate infrastructure and operations to keep operating costs as low as possible. In this situation, you might want to use mature open source but support it with an internal team. Then your strategy might look like this:
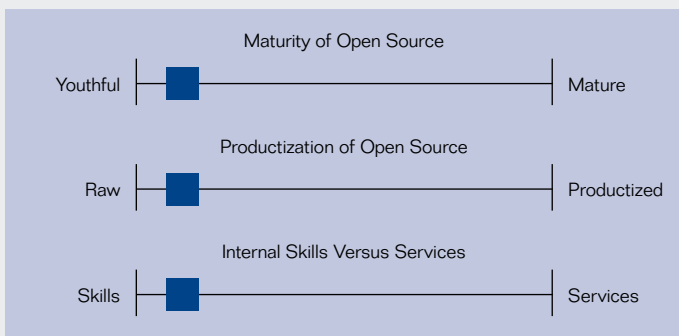
This highly skilled company uses all open source software available with a preference toward raw functionality that can be optimized over productized software, and the company prefers to support the use of open source by maintaining a high level of skills.
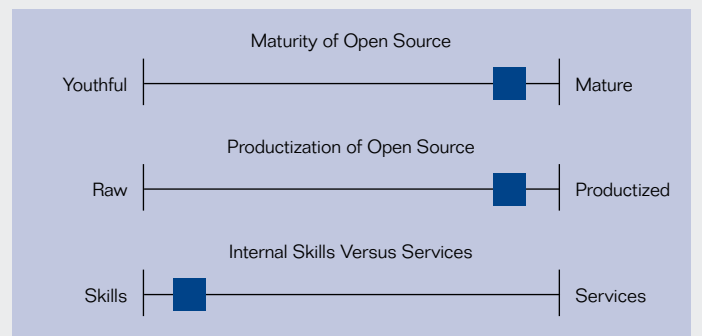
This highly skilled company uses mature open source supported by internal skills to support corporate infrastructure and applications, relying on internal support to reduce spending.

Perhaps your successful firm is a competitive low-margin business that operates at a massive scale. Technology supports the context of your business, but it is not strategic unless you can create and protect a cost advantage. Then your strategy might look like this:

This low-margin company uses mature, highly productized open source that is supported by services in order to reduce costs to the lowest possible levels without having to invest in a highly skilled IT staff.
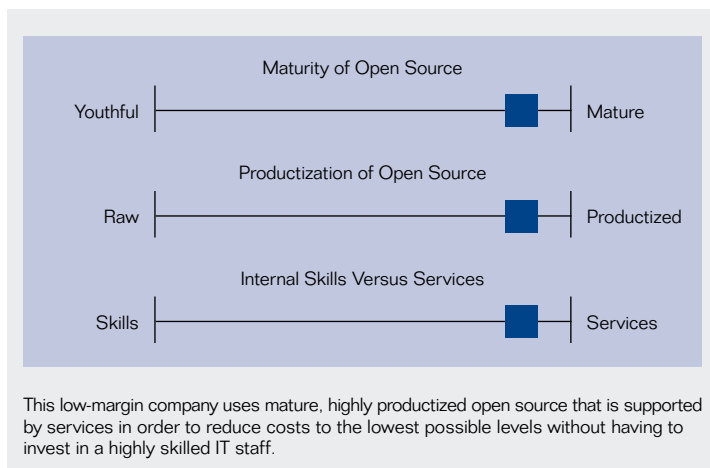
Many other strategies could be defined using these dimensions. It is possible to have different strategies in different areas. An online retailer, for example, might choose a high-skill strategy for its Web site but a low-margin strategy for supporting enterprise resource planning and other back-office operations.

Defining a strategy in the terms just set forth can help you quickly and clearly communicate the general approach you are taking to open source. An easy-to-understand strategy aligns efforts in your company and helps determine what sort of open source is appropriate. However, the business value comes from applying these strategies and putting open source to work.

THREE CATEGORIES OF OPPORTUNITIES

Although thousands of open source projects in the world are dedicated to almost every possible human endeavor, a relatively small number of projects – perhaps a few hundred – are relevant to the business enterprise. Of this small number, many projects have been productized in ways we have already described, and they represent an opportunity for any company regardless of skill level. Other open source software offers robust, raw functionality that can be applied only if the users have the right skills or can develop them. The opportunities we will analyze fall into three broad categories:

- **Infrastructure and operations**
  This type of software helps run servers and data centers and provides the foundation to make other software and technology work properly. Linux is perhaps the best example of this category.

- **Software development**
  This type of software has frameworks, platforms, and tools, such as programming languages and development environments, that help create new applications. The Ruby programming language, the Apache Struts framework, and the Eclipse IDE are prominent examples.

- **Applications**
  This type of software for end users does useful, functional tasks or helps automate a business process. OpenOffice.org, TWiki.org, the Firefox Web browser, SugarCRM, and Compiere are all entries in this area, which is also the most sparsely populated of the three categories.

Infrastructure and Operations

The largest return on the use of open source is found in the data center. Linux, of course, is the marquee offering in this category, because it provides so many ways to reduce costs and increase functionality and flexibility. What drives adoption most consistently is the fact that Linux-based infrastructures provide the stability, availability, reliability, scalability, and security that IT professionals expect from a mature operating system – all at a very compelling price-to-performance ratio.

The fact that Linux has the design elegance of UNIX, has the flexibility offered by open source, and is as mature, secure, and reliable as any operating system on the planet leads to many different types of opportunities, almost all of them fully productized by leading vendors.

For example, the UNIX characteristics of Linux make it a perfect platform to support **virtualization** of a computing environment. Offerings like Xen-powered virtualization technology integrated into SUSE Linux Enterprise Server from Novell or the Adaptive Computing Controller tool from SAP allow the definition and deployment of many different configurations of a computing environment on a single hardware infrastructure. A collection of servers can appear as one large computer, many small computers, or just about any combination in between.

Because Linux is supported on almost every major hardware platform, it is the perfect choice for **consolidation** and **standardization**. A heterogeneous collection of hardware can be made to look like one platform by installing Linux. Deploying Linux can cut hardware costs as well by enabling you to replace more expensive servers with less expensive ones.

Linux is also **profoundly secure.** It shares a common design philosophy with the UNIX operating system, and, unlike some other operating systems where security was added later, security has been an integral part of Linux from the start.

All of these features have made Linux the focus of many efforts toward **enhanced performance and scalability.** For example, Linux is supported on all major 64-bit computing platforms. Special-purpose software like the SAP NetWeaver® Business Intelligence Accelerator (SAP NetWeaver BI Accelerator) software runs on SUSE Linux Enterprise Server on HP or IBM hardware, using a special chip optimized for the application by Intel. SAP NetWeaver BI Accelerator is just one example of the way companies are collaborating by using Linux as a platform to create compelling new products.

Open source has had a significant impact on infrastructure and operations, because many offerings are mature and provide low-cost, robust, high-performance solutions.

The following examples describe how some companies are putting open source to work:
- **Scalability**
  Linux runs on inexpensive hardware that makes a scaling strategy of using multiple small machines very attractive. Hardware and software vendors have taken advantage of this trend by offering blade servers based on Linux that can be plugged in and used in a cluster. SAP NetWeaver BI Accelerator uses such an approach. The software is based on a Linux server optimized for use as a blade. Intel has provided an optimized chip for this purpose, and HP and IBM both offer a blade based on the chip that powers the system.

- **Virtualization**
  Another way that the power of many small Linux servers is being used is to provide a virtual computing environment. Companies are replacing computers distributed across the globe with clusters of Linux machines that deliver enormous computing power and can be reconfigured as quickly as needed.

  The following are specific examples of virtualization:
  - SUSE Linux Enterprise Server with integrated Xen-based technology creates a configurable virtualized computing environment that reduces the cost of providing computing resources.
  - HP offers products for virtualization of Linux computing resources based on VMware for HP ProLiant servers and HP Integrity Virtual Machines for HP Integrity servers. Both the ProLiant and Integrity product lines come with a complete suite of management tools.

- **Standardization and consolidation**
  Because Linux is supported by so many hardware and software vendors, it can be a natural choice when you are standardizing a heterogeneous environment. Most existing hardware does not need to be retired but can run Linux and play a role in a new standardized architecture. Because of the nature of Linux, it is also ideal for virtualization, which enables you to consolidate many smaller machines into fewer large ones. The Linux Standard Base has created a standard for how Linux distributions should work to make software more interoperable across versions.

- **Databases**
  All major database platforms, including Oracle, are fully supported on Linux. MySQL, MaxDB, and Postgres are just a few of the numerous open source databases that can be applied in various situations.

Servers, integration, and deployment
Integrating Windows and Linux is a key concern for most enterprises. Samba, a long-standing open source project, integrates Windows and Linux file systems, print sharing, and domain services. Samba allows Linux hosts to act as part of a Windows server domain or as a primary domain controller. Linux hosts can also participate in Active Directory domains. Many open source servers are available, from Web servers to application servers to mail servers, including the Axis open source SOAP server, the Apache Web server, the Geronimo and JBoss application servers, and a wide variety of mail servers, not to mention special-purpose tools, such as spam filters and mail-archiving programs.

Data center operations
A vast array of open source tools and special-purpose utilities exist for many different aspects of data center operations, including operational monitoring, backups, and automation of system administration tasks through scripting. Programs like Nagios for monitoring servers, Amanda for backups, and the Postfix e-mail mail server are all in wide use. The Zenoss enterprise monitoring system, written in Python, unifies monitoring, alerting, and trending in an extensible, scalable system. The Advanced Packaging Tool (APT), apt-get and its variants, makes it easy to deploy and update open source servers and software across the enterprise. The Asterisk open source PBX is widely used and saves the cost of purchasing a private branch exchange server.

End-user infrastructure
The Firefox browser with Greasemonkey scripting extensions, open source antivirus tools, and many different e-mail clients are just a small sample of open source infrastructure for end-user computing. Jabber, an open source instant messaging server, allows corporate use of instant messaging.

Software Development

The roots of open source can be traced to programmers who wanted to create better tools with which to do their jobs. The result is a rich collection of open source projects for supporting software development, as follows:

Programming and scripting languages
Languages like Perl, Python, PHP, and Ruby were all developed by communities seeking a better way to create software. There are also open source implementations of existing programming languages like the GNU C/C++ compiler, and various compilers and virtual machines for Java.

The following are examples of programming and scripting languages:
- The use of Ruby for business intelligence and PHP for user interfaces by Dan McWeeney of Colgate-Palmolive resulted in increased user friendliness and access to new communities of developers.
- PHP-based content management systems are used to create user interfaces to the SAP® ERP application.

Software development tools
The Eclipse integrated development environment with its massive developer community and myriad plug-ins is one of the best examples of the open source movement in action. The Ant build tool for Java and the JUnit testing framework are just two of hundreds of tools and utilities available to enhance software development. Hibernate is used to manage persistence, Tomcat is a popular servlet container, and the JBoss open source application server is widely used. More structured application server environments are also available. Sponsored by IBM, WebSphere Application Server, Community Edition, is a layered application server platform built on the Apache Geronimo application server. WebSphere Application Server, Community Edition, incorporates security and management features.

- **Content management, portals, and Web publishing**
Open source communities have created hundreds of different systems for managing content and Web site authoring. Joomla!, Drupal, OpenCms, TWiki.org, Plone, and Ruby on Rails are among the most widely used. Alfresco is an enterprise content management system built on JavaServer Faces. The Plone content management system runs on the Zope open source application server. The Openads Web ad server is widely used, is better than many commercial ad servers, and scales very well.
- **Searching**
Apache Solr provides fully faceted indexing and searches for Web sites and other enterprise servers in a world where users expect a Google-level quality of search results. It interfaces with the Lucene search library.
- **Project management, bug tracking, and version control**
Trac is a lightweight, Web-based project management and documentation tool used by many open source development teams. It combines a Wiki, a Web interface to Subversion version tracking, and defect tracking functionalities. RT — which stands for Request Tracker — is an open source ticketing system that enables trouble-ticket tracking for a variety of applications, from in-house users to customer support.
- **Development communities**
Open source software proved that a community of developers working together over the Internet could create great software. This lesson has been learned by commercial companies that have imitated some of the practices pioneered by open source communities to improve their internal development and communication processes. SAP, HP, Novell, and IBM now have thriving developer communities with large populations of users who share information and knowledge about using open source in the enterprise.

Applications

Applications based on open source technologies developed later than their counterparts in the infrastructure and development tools area. This is due, at least in part, to the fact that these two areas had to be in place first in order for applications to be built. Furthermore, because commercial applications are created by developers, but not necessarily for developers, you have to gather and understand user requirements even more carefully, which takes additional time. Types of applications that are based on open source include:

- **End-user application suites**
OpenOffice.org is a fully functional open source application suite. Firefox and Thunderbird provide key end-user productivity tools.
- **Document standards**
Standards such as the OpenDocument Format provide organizations with a way to store information in an open source format.
- **Collaboration**
TWiki, MediaWiki, MoinMoin, and DokuWiki are just a few of the open source Wiki environments that can be used as collaboration tools.
- **Statistical packages**
Statistical analysis is increasingly important in the business world, but licenses for SPSS and SAS data are prohibitively expensive. The R statistical language, based on S from Bell Labs, is free and comes with plenty of sample code and add-ons to provide statisticians and applied demographers with the tools they need to do market analytics and more.

Armed with information about the different types of open source projects that are available, you can evaluate to what degree your company's IT environment should be a combination of open source and commercial software. Any evaluation your company makes must take into account the history of open source, how it is different, and how it is being productized. The end result should be a combination that delivers value for your company.

PARTNER PAGES

HEWLETT-PACKARD

Hewlett-Packard (HP) has been at the forefront of technical and customer commitment since certifying Linux on its servers more than a decade ago. Since then, HP has delivered innovative, forward-thinking offerings, such as the first integrated, fully supported multivendor open source and commercial software and hardware stack (HP Open Source Middleware Stacks, or HP OSMS), a Linux indemnity program, Debian Linux platform enablement and Care Packs, and open source license management and governance services.

HP has made a significant investment to deliver open source and Linux "primed for business advantage," as highlighted by the following:
- 2,500+ developers focused on Linux and open source projects
- 6,500+ service professionals trained to support customers in 160 countries
- 80+ customer education centers worldwide
- 1,200+ open source printer drivers provided to communities
- 200+ open source tools, utilities, libraries, and packages ported
- 200+ products shipped with embedded open source software
- 100+ open source software projects initiated
- Widespread deployment of open source technology across the HP IT infrastructure

HP has led the market in terms of both Linux server shipments and revenues for more than nine years. The latest count has HP selling more than 2 million Linux servers and generating almost US$9 billion in cumulative Linux server revenues worldwide.

Infrastructure and Operations

HP's open source and Linux offerings are encapsulated in the HP Open Source Integrated Portfolio (OSIP), a collection of software, middleware, services, operating systems, and platforms designed to tailor customers' IT infrastructure according to their business needs. HP OSIP features a variety of offerings, including:
- Tier-one support of Linux across market-leading, standards-based platforms, including the high-value 32- and 64-bit HP ProLiant server family, the integrated, consolidated infrastructure of HP BladeSystem, and the highly scalable 64-bit HP Integrity servers based on the 64-bit Intel Itanium 2 processor
- Market-leading Linux distributions via worldwide business and technical alliances with Red Hat and Novell, tested hardware configurations with the new Oracle Enterprise Linux, country-level relationships with Mandriva and participatory members of Asianux, and platform enablement and services for community-led Debian GNU/Linux
- Common Criteria Security for HP ProLiant and HP Integrity servers supporting Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server from Novell, recognized by more than 20 governments worldwide to determine the security and assurance of various technology products
- Open source governance services to help customers institute a comprehensive program to govern the acquisition, licensing, code use and reuse, and distribution of open source technology
- Total solution accountability with an array of support offerings worldwide covering HP's multivendor portfolio

In fact, HP Services' records show that of the 83,636 open source and Linux support customer calls received in 2005 and 2006, 99.7% were handled independently and successfully by either level 1, HP Global Solution support, or level 2, HP Engineering. Less than half a percent of these calls needed independent software vendor (ISV) escalation, which was also handled by HP serving as the primary interface with the customer.

Software Development

HP Open Source Middleware Stacks (OSMS) offer customers three ways to develop open source technology on HP platforms:

- **Building blocks** of best-of-breed supported software components, including Symas, MySQL, JBoss, BEA, Oracle, Apache, Tomcat, PHP, Jabber, Serviceguard, and HP SIM
- Do-it-yourself **blueprints** of integrated and supported middleware stacks, including Web-application, Java Enterprise Edition, directory, and database servers
- **Consulting services** for a services-led engagement to generate fully customizable middleware stacks from open source and commercial software

HP OSMS can be complemented by traditional HP value-added offerings, including:

- High availability via award-winning HP Serviceguard for Linux, which automatically transfers control of a fail node's package to another node within the cluster to maintain service availability with minimal interruption
- HP management tools ranging from rapid deployment and workload management to remote management and high availability, with products like HP Systems Insight Manager (SIM), the all-in-one, Linux-centric management and deployment package called HP Insight Control Linux Edition, and service-level management from HP OpenView software
- HP products for virtualization, such as HP SIM, HP Virtual Machine Manager, VMware for ProLiant servers, HP NetTop, SystemImager, and HP Integrity Virtual Machines for HP Integrity servers
- HP products for HPC, including XC System Software, an HP-supported variation of Red Hat Enterprise Linux enhanced with clustering technology from the open source community and integrated by HP engineering, with cluster tools and utilities developed by HP and its partners

Applications

In addition to the market-leading open source and Linux partner products and the traditional HP technology offerings represented above, HP and its ISV business partners in the Developer and Solution Partner Program (DSPP) offer a wide spectrum of applications that can help clients achieve their desired business outcomes. HP is dedicated to its current and prospective open source and Linux partners, providing them with product and technology information as well as key equipment and development resources, including the test-drive program for 24x7 access to HP Linux servers. HP and its channel partners have embraced Linux for years and provide unmatched enterprise-grade support. HP Open Source and Linux Partner Program resellers benefit from HP confidential information, including project and marketing support to expand their open source and Linux business. HP's combined commitment to open source and open standards like Linux have made the partner program top in the industry.

Contact

For additional information, contact HP at the following Web sites:

- www.hp.com/go/osip
- www.hp.com/go/linux
- opensource.hp.com

IBM

Open source is an integral part of IBM's strategy for meeting the challenges of enterprise computing. Novell and Red Hat Linux have been deployed in more than 15,000 IBM customer engagements. Since 2001, IBM has invested more than US$1 billion to create a line of hardware, software, and services to allow customers to integrate Linux into their enterprise. Linux is the only operating system supported on **every** IBM server brand. While IBM's x86/x64 processor-based System x servers account for the majority of Linux volume, IBM has seen significant growth on higher-end platforms. The Linux on Power operating system can take advantage of the high-performance architecture of the advanced POWER5+, POWER6, and PowerPC 64-bit chips. Furthermore, approximately 25% of System z mainframe capacity currently being shipped to customers is designated for Linux workloads.

IBM software supports Linux and open source with more than 500 IBM middleware products available for Linux across the IBM software product lines. In addition, open source components (including Apache and Eclipse projects) are integrated into many of IBM's commercial software products.

IBM Global Services offers consulting, integration, support, and hosting to help customers with Linux integrations; one unit is devoted fully to SAP.

IBM is actively engaged in development of more than 150 open source projects. IBM's investment in Linux ranges from product support on the Linux operating system to more than 600 IBM developers contributing to, leading, and collaborating on the development of the Linux kernel and supporting tools and applications. IBM was an early supporter and contributor to Apache. The Eclipse software development environment was spun out as an open source project from IBM and has become a foundation for Lotus and Rational technologies. From hardware to databases to developer tools to support for applications, IBM can deliver value on the basis of open source software and community innovation to customers through its own offerings, through solutions created by its business partners, or through services. IBM is a one-stop shop for deploying productized, enterprise-ready open source software solutions in the enterprise.

Infrastructure and Operations

IBM's solutions for managing computing infrastructure and operations allow businesses to start small and grow elegantly. As a business faces pressures to provide expanded IT services at a lower cost, IBM's Linux-based offerings can help provide the ability to consolidate servers, improve interoperability, and manage multiplatform environments, all on an open standards–based Linux platform. IBM servers, ranging from x86/x64 to scale-up POWER and System z mainframes, running Linux provide a cost-effective path to scalability of the computing infrastructure. Targeted service and support offerings from IBM help to ensure a smooth delivery with reduced risk and enterprise-grade support for SAP solutions. The following examples list how IBM is putting open source to work:

- **Scalability**

  IBM's blade servers help allow for scalability through cluster configurations that you can quickly expand or repurpose to meet demand. IBM System p scales with the industry-leading POWER5+ and POWER6 processors for maximum performance, RAS, and lower environmental costs. IBM System z takes Linux to a new level, supporting 1,000 Linux instances running on a single system. IBM BlueGene/P and BlueGene/L systems running Linux top the list of "Top 500" supercomputers worldwide.

- **Virtualization**

  Invented by IBM more than 40 years ago, virtualization continues to be led by IBM's award-winning Advanced Power Virtualization engine and IBM System z/VM mainframe virtualization, which helps provide virtual failover, eliminate single points of failure, and regulate and control servers. IBM has also been an active contributor to open source virtualization technologies, such as the Xen hypervisor. Microsoft, Red Hat, VMware, and Novell offerings are also supported on IBM Systems.

- **Information management**

  IBM's DB2 and Informix databases are fully supported on Linux. IBM Cloudscape, an embeddable open source database that IBM contributed to the open source community as Apache Derby, is also supported on Linux. In addition, the Apache Lucene search technology is a core component of IBM OmniFind Yahoo! Edition, a free, robust enterprise search offering.

- **Collaboration offerings**

  Lotus Notes, Domino, Sametime, Quickr, and Connections all support Linux and leverage several open source components, including the Eclipse Rich Client Platform and Ajax technologies.

- **Data center operations**

  IBM's Tivoli data center management can run on Linux and help manage a heterogeneous environment, including Linux, Windows, Mainframe, and UNIX environments.

- **Service offerings**

  IBM's service offerings support the use of open source throughout the life cycle, from installation and configuration of Linux through deployment of open source tools and applications. IBM will support customers using Linux as a Tier 1 operating system similar to IBM's own AIX, i5/OS, and z/OS operating systems.

Software Development

IBM is a leader in the following open source development tools:

- **Software development tools**

  Eclipse was donated to the open source community by IBM and is supported and productized in many of IBM's offerings. Rational Software Delivery Platform is built on top of the Eclipse open source integrated development environment (IDE) framework.

- **Programming and scripting languages**

  IBM supports the development of open source languages, such as PHP, Python, Ruby, and Java.

- **Content management, portals, and Web publishing**

  IBM offers a rich set of software components for enterprise content management, social networking, search, and portals, all of which are supported on Linux.

Applications

IBM ISV business partners provide many of the applications that clients need. IBM provides its business partners with a comprehensive computing and development environment to build and test applications. IBM and its business partners have embraced Linux for years and can provide the enterprise-grade support you need.

Contact

For additional information, contact IBM at the following Web sites:

- www.ibm.com/linux
- www.ibm.com/opensource

NOVELL

SUSE Linux Enterprise 10 from Novell is a pioneering example of open source productization and leads Novell's numerous open source offerings. SUSE Linux Enterprise is the best-engineered, most interoperable platform for mission-critical computing, including running SAP software on Linux. SUSE Linux Enterprise is fully supported by Novell, a vendor who understands your business needs.

From the desktop to the data center, SUSE Linux Enterprise is a platform that is rock solid and enterprise grade; one that easily plugs into existing infrastructure; and one that's secure, simple to manage, and easy to use. With the Xen virtualization hypervisor built into the operating system, SUSE Linux Enterprise is also the leading open source virtualization platform.

SUSE Linux Enterprise is also supported by a broad range of hardware and software partners, including HP, IBM, Microsoft, and SAP. In fact, SAP recommends SUSE Linux Enterprise Server as a preferred platform for companies who want to run SAP software on Linux. Because of the partnership agreement between Novell and Microsoft, Microsoft recommends SUSE Linux Enterprise for customers who want to add Linux to their Microsoft environment. Only SUSE Linux Enterprise delivers complete Windows interoperability in areas like virtualization, directory and identity interoperability, and standards-based systems management. As a mixed source software vendor, Novell addresses every need that businesses have through a combination of open source and proprietary software. This combination enables Novell to work well with leading application vendors like SAP. In addition, Novell offers 24x7x365 Linux support worldwide through enterprise-level service agreements. Novell has certified the SAP Business Suite family of business applications under its Novell YES Certified Program; SUSE Linux Enterprise Server was also the first Linux platform to have certified integration with the SAP NetWeaver technology platform.

Finally, in the areas of intellectual property coverage, Novell continues to raise the bar. SUSE Linux Enterprise customers are covered through the Novell Indemnification Program (a covenant not to sue from Microsoft) and Novell's recent victory in court that established Novell's ownership of the UNIX copyrights. From the desktop to the data center, only SUSE Linux Enterprise offers a complete open source platform for the mission-critical applications that drive your business. For performance, reliability, security, and world-class support, choose SUSE Linux Enterprise from Novell.

Infrastructure and Operations

The following examples list how Novell is putting open source to work:

- **Scalability and performance**
  SUSE Linux Enterprise Server runs on the full range of hardware platforms from Intel-based servers to IBM zSeries mainframes. Performance gains can be dramatic; Siemens IT Solutions and Services found that it could use SUSE Linux Enterprise Server as a substitute for UNIX in SAP application servers at 40% of the cost of a UNIX-only configuration and get dramatically better performance. [3]

- **Virtualization**
  Adaptive computing from SAP and Novell provides a platform for virtualization, allowing computing resources to be dynamically allocated. SUSE Linux Enterprise Server also incorporates Xen for virtualization. In addition, as part of Novell's partnership with Microsoft, customers can run Microsoft Windows Server 2008 as a paravirtualized guest on SUSE Linux Enterprise Server.

- **Stability and security**
  Customers such as MTU Aero Engines GmbH give SUSE Linux Enterprise Server exceptionally high marks for stability and have experienced 100% reliability during several years of use.[4] Catholic Healthcare West has been able to reduce server administration time by 25% through its use of SUSE Linux Enterprise Server.[5]

---

3. See www.novell.com/success/siemens.html.
4. See www.novell.com/success/mtu.html.
5. See www.novell.com/success/catholic_healthcare_west.html.

▪ **Data center operations**
Management costs are key to long-term TCO, and companies that deploy Novell's ZENworks management solutions report an average cost savings of more than US$350,000 per 100 users.

Software Development

Novell believes that open source and open standards are the value-creation engines for the next generation of software. Novell is one of the most active contributors to the open source community.

This contribution starts with openSUSE.org, the open source project that Novell sponsors that becomes the foundation for SUSE Linux Enterprise distribution. openSUSE.org is a vibrant community with more than 40,000 active community members and more than 714,000 confirmed installations. The openSUSE project is a worldwide community program sponsored by Novell that provides a central hub for a community of users and developers who all have the same goal in mind: to create and distribute the world's most usable Linux.

Novell also contributes to the following projects:
▪ The Bandit project, which was launched in early 2006, drives cooperative work around identity management.
▪ The Higgins project, launched in early 2006 by IBM, Novell, and Parity Communications, focuses on software for user-centric identity management so users actively manage and control online information.
▪ Mono makes applications built on the Microsoft .NET framework run on Linux and other platforms. Many open source projects today are based on Mono, including Helix, Banshee, F-Spot, and Beagle.
▪ Novell is a key contributor to Openwsman, a project with a goal of exposing system management information.
▪ Open Management with CIM (OMC) is an open source umbrella project sponsored by Novell to promote standardization of data center management processes and integrated systems management.

▪ AppArmor project, lead by Novell, is designed to provide a simplified security framework for Linux.
▪ The Linux-HA project provides a high-availability (clustering) solution for Linux, which promotes reliability, availability, and serviceability. The most well-known component is heartbeat.
▪ The Aperi storage management project is a recently launched project that delivers a standards-based storage management framework.

Applications

Novell plays a key role in supporting and expanding the scope of open source applications for enterprise use. Novell is the number two contributor to OpenOffice.org and has done important work around VBA macros, including:
▪ As a sponsor of the GNOME project and a patron of KDE, Novell continues to contribute to desktop and document-format projects. Novell is a founding member of the ODF Alliance, a group designed to promote the adoption of OpenDocument Format (ODF) by governments worldwide.
▪ Novell Evolution e-mail and calendaring client offers built-in support for Microsoft Exchange.

Contact

For additional information, contact Novell at the following Web site:
www.novell.com/linux

www.sap.com/contactsap

50 086 426 (07/09)