Leveraging Open Source Processes and Techniques in the Enterprise

A VA Software White Paper
November 2004

www.vasoftware.com

Table of Contents

Executive Summary

The success of Open Source innovations has exceeded the wildest expectations of many observers. In just a few short years, Open Source technologies like Apache, Linux, JBoss and MySQL have not only become viable alternatives, but market leading and preferred solutions for a growing number of critical enterprise IT systems.

While Open Source tools and technologies are achieving mainstream acceptance, mounting competitive pressures have forced enterprise IT and R&D managers to come up with better ways to deliver high quality software in less time—and to do so with greater manageability and consistency. Given the rapid pace of development and innovation witnessed in the Open Source community, enterprise development organizations are taking a closer look at the Open Source development model itself. After all, how is it that widely distributed, informal groups can deliver such high quality results faster than many well-organized commercial endeavors?

As the creator and operator of SourceForge.net, VA Software is in a unique position to help. SourceForge.net is the world's largest Open Source software development web site, hosting more than 90,000 Open Source projects and over 950,000 registered users with a centralized resource for managing projects, issues, communications and code. Based on experiences gained working with the Open Source community, VA Software developed SourceForge Enterprise Edition, an application which retains the spirit and effectiveness of the SourceForge.net web site while providing enterprise-grade security, performance, extensibility, capabilities and ease-of-use.

Since the debut of SourceForge Enterprise Edition in 2001, VA Software has helped numerous Fortune 1000 firms and government agencies improve development efficiency and manageability. VA Software has also helped a number of these organizations to adapt and apply key Open Source best practices inside their enterprise. Based on our combined experience working with both the Open Source community and commercial development organizations, we have identified four key opportunity areas for leveraging Open Source processes and techniques inside the enterprise:

- **Skills and organization:** Visibility, openness and flexibility enhances the organization. To leverage the benefits of Open Source development methods, enterprises must lay the groundwork for, and encourage, a higher level of transparency, collaboration, cooperation and resource/project flexibility.
- **Processes:** Organizations must focus on three key Open Source process areas: user/developer collaboration; short, rapid, iterative development cycles; and "must have" features. These processes minimize waste and maximize efficiency by keeping development efforts focused on building the right software in a more agile manner.
- **Reuse:** Effective reuse means leveraging existing corporate assets by searching, consuming, commenting and contributing, and by making evolutionary improvements to those assets over time. Reusable assets should be construed broadly to include all assets captured digitally, not just code.
- **Tools:** Many Open Source practices rely on a centralized system and web-based tools to improve access, visibility and collaboration throughout the software development lifecycle. To aid adoption and use in the enterprise, a centralized system must be highly-flexible in order to work with existing development tools and processes.

Applying Open Source methods to enterprise development challenges yields results that vary from firm to firm. Commonly reported benefits include:

- Increased development efficiency and agility
- Higher/Improved software quality
- More efficient use of available resources and intellectual property (IP) assets
- Improved trust and morale
- Better predictability

Introduction

Open Source tools and technologies have rapidly achieved mainstream acceptance in the enterprise. According to a recent poll by InformationWeek, two-thirds of companies use Open Source products and an additional 16% plan to use Open Source in 2005.[1] At the same time, much of the Internet's software backbone is powered by Open Source applications such as DNS, Sendmail, Apache and Linux, as well as by Open Source programming languages like PHP, Python and Perl. Today, Apache is the number one web server application with 69% market share of active web servers.[2] Linux now commands greater than 50% market share for infrastructure applications like file and print servers, cache and firewall.[3] Linux is also quickly gaining ground in traditional UNIX server markets. Nearly everywhere you look in business, government, academia, and even at the consumer level, the use of Open Source tools, technologies and techniques is on the rise.

The reasons behind the growing interest, acceptance and even preference for Open Source products are many. The most commonly cited reasons include low-cost, high value, quality and reliability, security, increased freedom and flexibility (both hardware and software), and adherence to open standards. In addition, a supportive online community is increasingly being augmented by professional support from a growing number of vendors, including some of the biggest names in IT such as IBM, Oracle, Hewlett-Packard and Sun. These forces and interests have coalesced, resulting in increased adoption of—and confidence—in Open Source technologies.

While Open Source technologies are gaining momentum in the enterprise, the rapid pace of development and innovation in the Open Source community continues to intrigue many industry observers. How can far flung, informal groups of developers establish projects, execute them, and in the process deliver more applications more quickly, and often with better quality, than many well-organized commercial endeavors?

From an operational standpoint, the secret lies largely in the Open Source development model and in how developers leverage the Internet and centralized resources and tools to get things done. Fortunately, many of these same processes, techniques and key enabling technologies can be applied to today's enterprise development challenges.

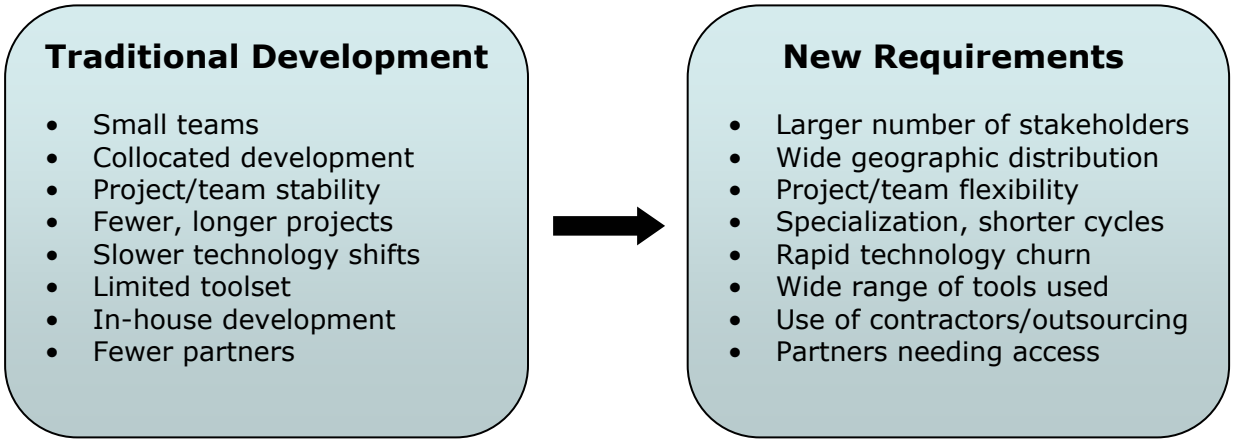
> Many of the practices and staffing models used by Open Source projects are relevant to corporate IT; managers and developers should study and adopt these.
>
> Liz Barnett, Vice President, Research Leader
> Forrester Research

Enterprise Development Challenges Have Changed

The environment for software development is not like it used to be. Rapidly changing market conditions, competitive pressures, fixed resource constraints and stringent new regulatory requirements have put intense demands on today's IT and R&D organizations. Never before have the strains been so great—nor the costs of mistakes so high.

- Development teams are under increased scrutiny and pressure as both applications and application development become increasingly mission-critical to the business
- Teams are often tasked to deliver more, better, faster—and often with less
- Businesses need their software development organization to be increasingly agile so they can respond swiftly to competitive threats and new opportunities

Developing, delivering and maintaining high quality software in these conditions is challenging. To make matters worse, application size and complexity are rising as the traditional software development environment has given way to a new, more fast-paced environment with a whole new set of challenges and requirements.



New requirements pose significant new challenges on today's development organizations.

In today's more demanding development and business climate, the importance of software development efficiency, manageability and agility have become more critical than ever. From a management perspective, leaders must find better ways to enable their organization to:

- Respond more quickly and effectively to changing requirements and goals without adding significantly to their headcount or cost base
- Make more efficient use of available resources and assets
- Improve management oversight and governance

Experienced managers know that today's challenges cannot be adequately addressed by just "trying harder". The status quo will no longer do. For many, this means optimizing distributed development by organizing and tightening up processes and ensuring that the tools and systems are in place so individuals and teams can be as efficient and productive as possible. Unfortunately, few top-down process improvement initiatives have won the hearts and minds of developers. Thus, interest in leveraging best practices from the Open Source community – methods that have been honed by developers and proven to scale – has grown dramatically. The convergence of Open Source and enterprise software development methodologies is inevitable and underway.

> To reduce the risks and hidden costs of distributed development, companies should consider solutions that give them effective visibility and control of distributed engineering activity while improving communication, collaboration and reuse to drive greater development efficiency.
>
> Joseph Feiman, VP and research director
> Gartner, Inc.

Open Source Development Model

The essence of the Open Source development model is the rapid creation of solutions within an open, collaborative environment. Despite the many challenges associated with geographically-distributed development, the Open Source development model thrives on collaboration and agile development practices. Key characteristics and practices of the Open Source development model include:

- **Projects started instantly**: Open Source projects are established and organized swiftly to capitalize on new and emerging ideas and technology.

- **Requirements defined quickly**: Requirements are defined quickly and collaboratively with input from both developers and users.

- **Distributed development**: Open Source projects are worked on by virtual teams of globally-distributed developers.

- **Collaborative development**: Development and QA is done collaboratively, with peer reviews, and typically accomplished with small, agile teams as the primary contributors (though many others have visibility into the process and can contribute as needed).

- **Short, fast, iterative development cycles**: Open Source development mirrors Agile development techniques like eXtreme Programming (XP), emphasizing short, fast, iterative development and release cycles. There are frequent, automated builds, continuous integration and test, and a regular hierarchy of releases such as "nightly", "development", "stable," etc.

- **Reuse:** The Open Source community is highly pragmatic and will readily reuse existing code and other artifacts to accelerate development. Many times, reuse and refinement begins with source code rather than compiled binaries or executables.

- **Multi-project flexibility**: Developers are typically involved with multiple projects concurrently, though at varying levels of involvement depending on their skills, expertise and interests.

- **Asynchronous communication**: Project members rely heavily on network-enabled, asynchronous communication methods like email and discussion forums for both ad hoc and structured communication. The centralized storage of these communications significantly aids new project members in understanding the history of the project so they can contribute more swiftly.

- **Leverage centralized system**: Open Source developers leverage a centralized, web-accessible resource (ala SourceForge.net) for the management of projects, communications, code, issues/defects, documentation and knowledge, as well as for the recruitment of experts.

- **Openness and trust**: Openness (transparency) of development builds trust and confidence. Consumers of the results of an Open Source project can often see the complete project history, including where things came from, who has been contributing, popularity/activity, plus feedback from the community.

Overall, the Open Source development model is a very fluid, collaborative and agile process. It works and works well despite the challenges posed by distributed development. By using collaborative development processes and a centralized system and tools for managing projects issues, communication and code, the Open Source development model allows virtual teams to produce high quality results with surprising efficiency.

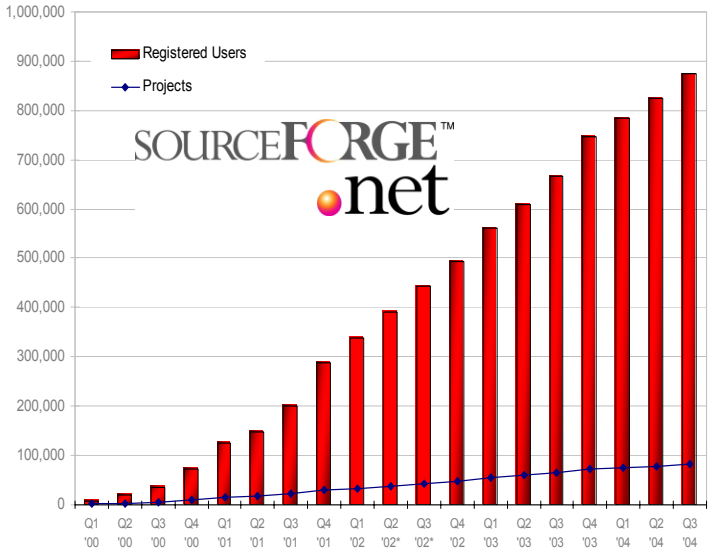
Role of SourceForge.net in the Growth of Open Source

A key enabling technology behind the success of the Open Source development model is that virtual teams leverage the Internet and a centralized, web-accessible resource like SourceForge.net. VA Software founded SourceForge.net in 1999 after studying how the Open Source development worked. One of our early discoveries was that there was no central place to find and coordinate work on Open Source projects. This lack of visibility and access created numerous inefficiencies, redundancies and missed opportunities in the Open Source development process.

To overcome these issues and accelerate Open Source projects, VA Software launched SourceForge.net. It was the first web site to provide a single, centralized location with integrated tools to support a broad range of Open Source projects and initiatives. Our early goals for the site were to:

- Make it easy to establish projects and recruit experts
- Minimize administrative work
- Maximize throughput of communications, collaboration and visibility
- Let developers continue working with their favorite tools
- Preserve project knowledge
- Find and leverage existing code
- Do all of this on a global scale
- At no cost to users

SourceForge.net galvanized the Open Source community and has enjoyed remarkable success. Since 1999 it has steadily grown to host more than 90,000 Open Source projects and support over 950,000 registered users. As the world's largest Open Source software development web site, SourceForge.net has become the global nexus for the Open Source community. It has also put VA Software and our subsidiary, OSTG, in a central and vital role in the Open Source movement.



Strong growth on SourceForge.net testifies to the rapidly expanding interest in Open Source. (Source: VA Software)

Separately, in response to requests from several Fortune 500 firms who were impressed with what was happening on SourceForge.net and wanted something similar behind their firewall, VA Software introduced SourceForge Enterprise Edition. Based on experiences gained working with the Open Source community, SourceForge Enterprise Edition retains the spirit and effectiveness of the SourceForge.net web site while providing enterprise-grade security, performance, extensibility, capabilities and ease-of-use.

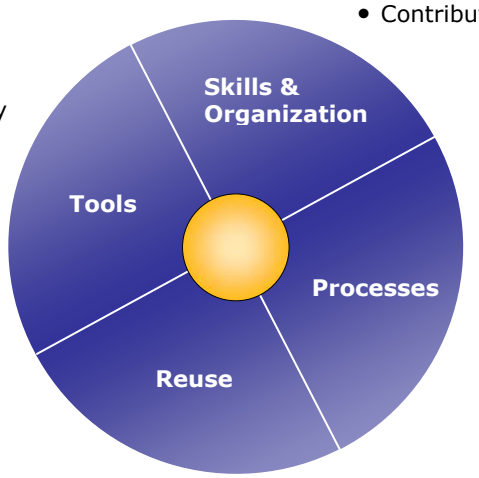
Since the debut of SourceForge Enterprise Edition in 2001, VA Software has helped numerous Fortune 1000 firms and government agencies improve the efficiency and manageability of their commercial and scientific software development efforts. We have also helped a number of these organizations leverage Open Source development methods inside their enterprise. The following section describes the four key opportunity areas for leverage Open Source processes and techniques for internal software development.

Four Key Opportunity Areas for Leveraging Open Source Development Processes and Techniques

VA Software's combined experience working with both the Open Source community as well as large commercial and governmental development organizations has enabled us to identify four key opportunity areas and best practices for leveraging Open Source methods in the corporate development organizations.

- Work visible & shared with others
- Greater resource/project flexibility
- Contributions based on merit

- Central repository & visibility
- Networked communication
- Flexible development environment



- User / Developer collaboration
- Short, fast, iterative, development cycles
- Focus on "need to have"

- Leverage existing assets
- Search, consume, comment, contribute
- Evolutionary improvements

1) Skills & Organization

→ **Key Takeaway**: Visibility, openness and flexibility enhances the entire organization. To leverage the benefits of Open Source development methods, enterprises must lay the groundwork for, and encourage, a higher level of transparency, collaboration, cooperation and resource/project flexibility.

Make Work Visible and Shared with Others

Visibility and transparency enhances exposure and trust. When people know their work will be visible to others, it encourages best efforts and better behavior. Rather than hiding in one's cubicle and attempting individual heroics, visibility and openness encourages sharing of problems with others and asking for help earlier. This helps solve issues before they become problems that may impact other staff and project timescales. Visibility and openness also facilitates a collaborative peer community and reduces the need for managers to "enforce" processes. In many cases, deviations from the standard will be immediately seen and rejected by the peer group.

> Firms should increase transparency – the visibility of the project information to people who aren't team members – to improve coordination and help increase productivity. Increased transparency can also improve the accountability and the performance of individual employees by providing a documented workflow and a basis for feedback.
>
> - Liz Barnett, Vice President, Research Leader
> Forrester Research

Promote Greater Resource / Project Flexibility
Open Source developers work on several projects at once, but are more involved in some projects than others. Surveys have shown that core project contributors tend to be contributors on two to 10 other projects as well.[4] This flexibility speeds development as experts can lend skills and expertise to more quickly solve the problem(s) at hand.

In a corporate setting, applying the right resources to the right problems at the right time is critical for making the most of available resources. But it is a significant challenge for most organizations. In many cases:

- Both the problem owner and potential problem solver(s) fail to connect in a timely manner—if at all—due to lack of awareness of the issue and each other
- The "right" resources are "unavailable" as they are assigned to another project

Within the enterprise, behavioral, organizational, geographic and technical barriers can impede issue visibility and resolution. Companies can take steps to overcome these challenges by employing a centralized and highly-accessible system to improve visibility and connect people with issues—regardless of location.

Solving the second problem requires more direct management support. Time- and resource-constrained managers may resist having key contributors "wander" to other projects, even if only temporarily. Yet a higher degree of project/resource flexibility is required for the organizational give and take necessary to make optimum use of available resources. While it is unrealistic to allow complete freedom in a corporate setting, an increasing number of organizations have realized that a moderate amount of self-determination boosts efficiency and morale. Today, Google requires that staff spend 20% of their time on projects of their own choosing. 3M has been noted for encouraging its engineers to spend 10% of their time on self-directed projects and activities. And cccording to managers at Hewlett-Packard, developers benefit from the broader exposure that results from their participation in multiple projects, and are generally more engaged and more satisfied than their peers[5].

Accept Contributions Based on Merit Rather than Origin
In the Open Source development model, contributions are based largely on their own merit rather than their origin. This egalitarian practice encourages participation, collaboration, critical examination and innovation by a wider range of contributors. To work in a corporate setting, there needs to be greater openness to evaluating potential contributions based on their own merit, regardless of the title of the person making the contribution. Healthy debate should not only be tolerated but encouraged. This approach helps companies improve quality while harnessing better ideas and innovations from across their extended enterprise.

2) Processes

→ **Key Takeaway**: While there are many Open Source development processes, enterprise development organizations usually gain the most by focusing on user/developer collaboration; short, rapid iterative development cycles; and "essential" features. These processes keep development efforts focused on building the right software in an agile manner and lead to reduced effort and rework, improved quality, and shorter time-to-market.

Encourage Frequent User / Developer Collaboration
Collaboration between users and developers is not unique to the Open Source development model. After all, the importance of understanding and capturing user

requirements at the start of development is well-known. What is special about user/developer collaboration in the Open Source model is the level of this collaboration and how it extends throughout the software development lifecycle. Enterprises should do the same. To the extent possible, requirements, designs, prototype testing, development and testing should be done with frequent user involvement rather than just at the beginning and the end. Frequent collaboration between users and developers speeds trade-off decisions and is a key tenet of several agile development methodologies. Advantages include:

- Faster development of more accurate, clear and realistic requirements
- Reduced waste and rework throughout the development lifecycle as users provide feedback on prototypes and other early releases
- Greater mutual ownership and satisfaction with results through more equal involvement of both users (consumers) and developers (producers)

Like many distributed organizations, Open Source collaboration is often conducted electronically via mailing lists, discussion forums, etc. Besides their suitability for globally-distributed teams, electronic communications that are organized and stored within a centralized system have the added benefit of providing a documented project history. This allows new team members to come up to speed quickly while providing a valuable information asset that can be tapped for future development efforts.

> "...only 54% of the originally defined features are delivered, and even more troubling is the realization that, of those features that are delivered, a full 45% are never used. The requirements that are developed and approved (and supported by large paper documents) often do not represent what the customer wants or needs - but this isn't apparent until significant cost has been devoted to development and testing based on these flawed requirements. The rework required and the trade-offs expected from the customer are simply mind boggling!"
>
> Standish Group
> 2003 Chaos Chronicles Report

Employ Short, Fast, Iterative Development Cycles

The Open Source development model largely eschews traditional waterfall development approaches in favor of short development cycles and frequent releases that make incremental progress towards the goal. Enterprises can benefit from this same approach. They should also invest in setting up environments that automate the build, integration, test and release processes. Automating these processes reduces errors while aiding the "release early, release often" approach. Practicing shorter, incremental development cycles enables organizations to be more responsive and accommodating to the inevitable changes that occur during the development lifecycle. It also reduces the risk of longer, more costly efforts that miss the mark.

Shorter development cycles also breaks the strong sense of ownership and defensiveness some software developers may feel for the code they have produced. Such strong feelings may make it difficult for developers to accept criticism about their work and/or agree to changes. If the focus is on short development cycles within a collaborative environment, there is greater mutual "ownership" of the code. Changes based on merit are more readily accepted since emotional resistance is lowered.

Focus on "Need to Have" Features

Open Source projects typically focus on essential, "must have" features. Focusing on "need to have" versus "nice to have" features speeds development by getting essential functionality into the application sooner, validating and stabilizing it, and then building upon it incrementally. By focusing on "must have" functionality with

fast, iterative development and release cycles, enterprise development teams can speed on-going trade-off decisions based on genuine need. This not only reduces time-to-market, it can reduce development costs, rework and wasted effort while improving overall quality.

3) Reuse

→ **Key Takeaway**: Effective reuse in the enterprise means leveraging existing corporate assets by searching, consuming, commenting and contributing, and by making evolutionary improvements to those assets over time. Reusable assets should be construed broadly to include all assets captured digitally, not just code.

Capitalize on Existing Corporate Assets
A common practice of Open Source developers is to build on (i.e. reuse) the work of others. In a typical enterprise, there are also numerous opportunities for reuse based on materials created throughout the organization. One of the challenges with reuse is that in many cases the left hand does not know what the right hand is doing (or even that there is a right hand!). Teams may therefore duplicate effort across a wide range or work: research, design, architecture, processes, source code, test cases, etc.

A key part of the solution is putting a system in place that allows staff to more easily store, find and comment on digital assets. If staff have an easy way of seeing what other people have done—or are doing—they can determine if there is a body of work or expertise they can tap into to accelerate their efforts. Plus, as noted by Larry Wall, father of Perl, developers are lazy. They would rather build on the efforts of others than re-invent the wheel themselves. A centralized system helps "lazy" developers reuse code by making it easier to store, search and find it for reuse purposes.

Corporate reuse initiatives have tended to focus on code. But software is only one type of reusable asset. Other high value assets with reuse potential include: documents, designs, architectures, models, templates, plans and processes. And while assets that are specifically designed for reuse typically have greater reuse feasibility, it is not a pre-requisite. Sometimes "close enough" is "good enough" to build on.

Another source of reusable assets is the Open Source community itself. On Open Source web sites like SourceForge.net, there is a rich and rapidly expanding base of Open Source assets to draw from. However, commercial organizations should have processes in place for determining the acceptability of Open Source licensing restrictions and obligations, plus means of ensuring compliance.

Search, Consume, Comment, Contribute to Corporate Assets
Reuse is difficult if assets cannot easily be stored, found, understood and trusted. For Open Source development, asset discovery is aided by a centralized and searchable Open Source repository like SourceForge.net, or through Internet search engines in general. Understanding, trust and consumption are aided by the transparency surrounding that asset: its source, its development context, plus its relative popularity and quality based on community feedback. Organizations seeking to increase asset reuse initiatives should:

- Put systems and processes in place to improve asset accessibility, searchability and transparency
- Encourage/reward team members to reuse assets, comment on those assets and contribute them back to the organization

Make Evolutionary Improvements over Time

In the Open Source community (and in many commercial organizations), applications, approaches and ideas are subject to the principle of natural selection. Better ideas and technologies gain prominence and carry on while less useful, less capable and/or lower quality alternatives fade away. Each reuse (or rejection) provides opportunity to incrementally improve assets. Within an enterprise, the same principles apply. By reusing existing assets and making evolutionary improvements to those assets improve over time, these assets form the basis of an intellectual property "war chest" that can give companies a distinct competitive advantage.

4) Tools

→ **Key Takeaway**: Many Open Source practices rely on a centralized system and web-based tools to improve access, visibility and collaboration throughout the software development lifecycle. To aid adoption and use in the enterprise, a centralized system must be highly-flexible in order to work with existing development tools and processes.

Most enterprises have no shortage of software development and project management tools. Quite often these tools—especially in larger, distributed organizations—are heterogeneous (i.e. come from a variety of sources) and are not well-connected with each other, nor with any centralized system. Without a common "backbone", tools and process gaps are plentiful and visibility into development activity and assets is limited. Typically these gaps are filled with inconsistent manual processes, meetings, "heroic" efforts, and numerous other inefficiencies that slow development and contribute to higher direct and indirect costs.

By contrast, Open Source developers typically leverage a centralized, web-based collaborative environment like SourceForge.net. Enterprises seeking to leverage more Open Source methods should focus on providing teams with a collaborative development environment of their own: a secure, web-based system that connects existing tools and processes with a centralized repository and tools. From a tools perspective, managers should focus on establishing a system with the following key characteristics:

- Centralized repository and visibility
- Network-enabled collaboration
- Flexibility to work with existing development tools and processes

Centralized Repository and Visibility
Leveraging a centralized, highly-accessible system like SourceForge is a key enabling technology for many Open Source processes and techniques. A centralized repository enables development assets, activities and communications to become centrally located and managed. This improves asset accessibility for reuse and collaboration purposes while improving visibility and control throughout the software development lifecycle. It allows distributed development activity to go from opaque to clear as developers and other stakeholders can simply open their web browsers and see what is going on—regardless of location. To access this kind of capability within complex enterprise environments, organizations should seek solutions like SourceForge Enterprise Edition that provide secure, behind-the-firewall operation and enterprise-grade tools that integrate with and augment existing environments.

Network-Enabled Collaboration
Open Source developers thrive on tight communication and collaboration. To accomplish this despite wide geographical and time-zone differences, they rely heavily on the Internet and network-enabled collaboration tools like mailing lists, online discussion forums, document repositories and wikis. These online

collaboration tools have proven to work on a global scale and most enterprises use at least some of them today. However, for much of the Open Source community, the effectiveness of network-enabled collaboration is enhanced by a centralized system like SourceForge.net, which automatically captures, archives and connects communications with a centralized repository. This allows a tighter coupling of collaboration with development activity and code and provides a richer collaborative development environment for improving development efficiency.

Provide a Flexible Development Environment to Aid Adoption and Use
One thing virtually all developers have in common is that they like to use their favorite tools. Open Source developers tend to favor proven Open Source tools like the GNU toolset and CVS (Concurrent Versions System). Many also use Open Source automated build tools such as Ant, CruiseControl and Tinderbox. Regardless of the tools used on any given project, the key enabling technology (or backbone) for over 90,000 Open Source projects is the SourceForge.net web site. SourceForge.net provides web-based tools and resources that allow individuals and teams to manage projects, defects and issues, communications and code in a centralized, collaborative development environment. And it is available to everyone 24x7x365.

In enterprise settings, providing a flexible development environment is essential for realizing the benefits of Open Source processes. Developers prefer to continue using many of their favorite (and often disparate) tools, even on a single project. And enterprises may need to protect significant investments in current tools and training. Having a flexible, integration-ready system to fill tools gaps and connect processes is far more feasible and effective than trying to force multiple groups into using a single process dictated by a "heavyweight" toolset. Organizations should therefore focus on tools and a tools infrastructure that emphasizes flexibility to ease adoption and use.

Conclusion

The rapid pace of development, and the success, of numerous Open Source projects has proven that the Open Source development model can overcome distributed development barriers and produce high-quality results with remarkable efficiency. Given the mounting competitive pressures and challenges facing today's enterprise IT and R&D organizations, leaders should study and apply proven Open Source processes and techniques in order to optimize their development operations. By focusing on the four key opportunity areas that have proven most effective for leveraging Open Source development methods in the enteprise (skills and organization, processes, reuse, and tools), corporate development organizations can improve development efficiency, agility and manageability to better meet their business goals.

Footnotes

1. "Open Source Software Use Joins the Mix", D'Antoni, Helen, *InformationWeek*, November 1, 2004.
2. "Market Share for Active Web Servers, June 2000 – June 2004," Netcraft, June 2004.
3. "Open Source Software Hype Cycle 2004," Gartner note #G00120900, June 2004.
4. Walt Scacchi, "Open Source Under the Microscope," January 5, 2004, CNET News.com.
5. Based on telephone conversation with Hewlett-Packard managers, cited in, "Applying Open Source Processes in Corporate Development Organizations", Barnett, Liz, Forrester Research, Inc., May 20, 2004.

About VA Software

VA Software Corporation (NASDAQ: LNUX) is a leading provider of software, information and community support to IT managers and development professionals. VA Software is at the center of today's Open Source technology revolution, inciting innovation across the Open Source and enterprise development communities by helping them share, evolve and realize ideas more efficiently.

VA Software is the creator of SourceForge and the parent company of OSTG (the Open Source Technology Group), the world's leading community-driven media network. SourceForge.net is the world's largest Open Source software development web site and the global nexus for the Open Source community. SourceForge Enterprise Edition is a redesigned, enterprise-grade version of SourceForge for optimizing and managing distributed development across the enterprise. Today, SourceForge Enterprise Edition is being used by numerous Fortune 1000 companies and government agencies to improve development efficiency and manageability.

For more information on VA Software or SourceForge Enterprise Edition, visit www.vasoftware.com.

VA Software Corporation
46939 Bayside Parkway
Fremont, CA 94538 USA
1.510.687.7000 tel
1.877.825.4689 toll-free

www.vasoftware.com
info@vasoftware.com