

June 28, 2006

# The Forrester Wave™: Open Source Projects, Q2 2006

by Michael Goulde

TECH CHOICES



June 28, 2006

## The Forrester Wave™: Open Source Projects, Q2 2006

Eclipse IDE And Apache HTTP Server Lead In Our Project Evaluation

by **Michael Gould**

with Randy Heffner and Megan Daniels

### EXECUTIVE SUMMARY

Forrester evaluated 13 leading open source software projects across approximately 40 criteria and found that six of the projects stand out as examples of excellence. The strongest Leaders are MySQL, Eclipse IDE, Apache HTTP Server, Apache Tomcat, JBoss Application Server, and PHP — thanks to their strong governance, well-defined road maps and release plans, and large and active communities. These six projects are followed closely in the Leaders category by Hibernate and Apache Velocity, the former because of its wide usage and strong support, and the latter because of its maturity and stability. Strong Performers are PostgreSQL and Spring Framework, both of which have capable leadership, active communities, and good presence in the market but don't have some of the benefits that a strong commercial supporter can provide. Although they trail PostgreSQL and Spring Framework, Apache Struts, Apache Cocoon, and Apache Geronimo projects are also Strong Performers. Nevertheless, Apache Struts is being eclipsed by newer technologies, such as Spring Framework and JavaServer Faces; Apache Cocoon appeals to a limited audience; and Apache Geronimo is a relative newcomer that only released its first version at the beginning of 2006.

### TABLE OF CONTENTS

- 2 The Role Of Open Source Software In Application Development Is Growing**
  - Open Source Software Producers
- 3 Open Source Projects Evaluation Overview**
  - Companies Want Active, Stable Projects
  - Notable Open Source Projects
- 6 Many Leading Open Source Projects Exist**
- 8 Project Profiles**
  - Leaders
  - Strong Performers

#### RECOMMENDATIONS

- 12 Open Source Software Is Ready For Enterprise Use**
- 12 Use This Wave As An Evaluation Model**
- 14 Supplemental Material**

### NOTES & RESOURCES

Forrester conducted evaluations in October 2005 and followed up in March 2006. We interviewed leaders or committers from the 13 projects. Developers from Barclays Global Investors, Boston University, McKesson, OpenLogic, Optaros, National Leisure Group, PriceGrabber.com, SourceLabs, SpikeSource, and Yellow Online were interviewed.

#### Related Research Documents

["Open Source Usage Is Up, But Concerns Linger"](#)  
June 23, 2005, Trends

["Evaluating The Health Of Open Source Software"](#)  
April 23, 2004, Tech Choices

## TARGET AUDIENCE

Application development executive, enterprise architecture executive, technology product management executive

## THE ROLE OF OPEN SOURCE SOFTWARE IN APPLICATION DEVELOPMENT IS GROWING

The existence, quality, and functionality of the software being developed by open source communities have spurred a revolution not only in application development but also in the software industry as a whole. The explosive growth of the World Wide Web in the 1990s was partly enabled by the availability of open source software, especially the Linux operating system, Apache HTTP Server, and MySQL database, as well as the PHP, Perl, and Python scripting languages. That success has led to a continuing expansion in the number of open source projects that are producing software to meet a wide variety of application development needs. There are more than 100,000 projects listed at SourceForge.net and thousands of projects underway elsewhere.

Open source middleware and infrastructure software — including application servers, Web servers, database servers, and frameworks — are playing a growing role in enterprise application development. In 2005, among the 56 percent of North American companies using open source software, more than half were using projects like Apache HTTP Server, Apache Tomcat, and MySQL.<sup>1</sup> Open source tool usage also continues to grow. This trend is increasing in parallel with the widespread adoption of Linux, an open source operating system, but it is not directly tied to or limited by the growth of Linux.

### Open Source Software Producers

Many open source communities are staffed by passionate volunteer programmers working on a piece of code they care about deeply or for which they have a strong need. However, open source software comes from a variety of sources other than those prototypical communities of volunteers. Open source software is also being developed by open source software companies (e.g. JBoss and MySQL), a consortia of open source and traditional software companies (e.g., The Eclipse Foundation), and consulting firms that both develop and support an open source project (e.g., Interface21/Spring). The challenge for companies is to be able to sort through all the various kinds of projects and find those that are strong and stable enough to provide a durable product they can incorporate into their technology architecture. This task is complicated for several reasons. Decision-makers have to:

- **Decide to consider an open source alternative.** There are few, if any, major open source projects that don't compete with a commercial product. The first step in engaging in open source software evaluation is to decide to accept potential tradeoffs in functionality and support, and consider open source alternatives to commercial products.

- **Choose from multiple options.** There are often many different open source projects in the same functional category. Since many open source projects are focused on implementing standards, multiple implementations are to be expected. Wisely choosing open source software requires careful analysis of the health of the open source project that produces it. This report focuses primarily on the nonfunctional criteria necessary to select, from among different open source projects, the one(s) that compared with commercial alternatives.
- **Comparing open source and commercial.** Commercial alternatives may provide more or less — and better or worse — functionality than open source alternatives. A step-by-step approach to choosing between competing commercial and open source products is beyond the scope of this report; however, there are some areas where common criteria can be applied to both open source and commercial software — such as functionality and standards support — and other areas where the criteria for one makes no sense for the other. For example, revenue and number of employees are not reasonable criteria for evaluating an open source project.
- **Find a way to assess project maturity.** Companies are used to evaluating the viability of a traditional vendor before making a product choice. Those evaluations are often comparatively simple because information on the company, customers, and revenue is often readily available. But for open source projects, the same kind of evaluation is much more difficult, because there is often no formal company behind a project.

Open source projects use several models for obtaining the resources necessary to develop software. Some rely on volunteers; others rely on contributed resources; and others are supported by a paid development staff. There are tradeoffs to each model, and evaluating the various models is beyond the scope of this report. When you are selecting from among competing open source projects, you need to be aware that you are comparing software originating from different models.

Apart from Linux, the bulk of activity in open source today is in the middleware and tools categories. Developers have a continually increasing choice of open source projects from which to choose when building applications. The decision of which projects to use is generally guided by word of mouth, as developers make use of email lists and discussion groups to share their experiences with various projects. As open source continues to move into the mainstream, IT managers are looking for additional criteria to use to evaluate and choose projects to include in mission-critical development efforts.

## OPEN SOURCE PROJECTS EVALUATION OVERVIEW

To assess the state of leading open source projects and see how selected projects compare, Forrester evaluated the strengths and weaknesses of some of the top open source projects for application development and deployment. Of the more than 100,000 open source projects, Forrester chose 13 projects that are well known, widely used, offer critical functionality to application developers, and

that we believe can stand as proxies for other open source projects that have similar approaches to governance, development, community, and business.<sup>2</sup> Our evaluation of these 13 projects provides an evaluation framework that companies can use to perform similar evaluations on the thousands of less-widely used open source projects they may consider adopting.

Our in-depth evaluation of 13 leading projects reveals that the open source development model is working. These projects were led by dedicated people and all had good governance models and development processes in place. All are achieving their primary goal of developing software that is useful to developers and available without commercial constraints. However, there are implementation differences among the projects we looked at that reveal that there is no single best approach for an open source project but rather that projects have unique characteristics based at least in part on the individuals who comprise the community.

### Companies Want Active, Stable Projects

After examining past research, user need assessments, and vendor and expert interviews, we developed a comprehensive set of evaluation criteria (see Figure 1). We evaluated vendors against approximately 40 criteria, which we grouped into three high-level categories:

- **Current offering.** Standard Forrester Wave methodology, focusing on head-to-head product selection, evaluates current offerings in terms of the functionality required within the product category being assessed. The purpose of this Wave is, by contrast, to evaluate the viability of the projects behind the products. As a result, this Wave is made up of products in different categories and cannot evaluate product functionality head-to-head across all the products. To assess each project's strength, this Wave's current offering section evaluates each project against five groups of criteria: history, activity, training and publications, release history, and perceived health. In a sense, these reflect the functionality of the community rather than the technology itself.
- **Strategy.** We considered each project's current strategy, as well as its vision for the future. Current strategy was evaluated based on its ability to facilitate market adoption, including such criteria as the project's approach to licensing; its strategy for platform support; and its response to security issues, governance model, and competition. Future strategy looks at each project's ability to articulate a vision for the next several releases.
- **Market presence.** We combined information about each project's user and vendor adoption along with other commercial support available to gauge the extent of presence in the market for each project.

**Figure 1** Evaluation Criteria

CURRENT OFFERING	
History	What is the history of the project?
Current Activity	What is the current level of activity around the project?
Training and publications	What support/documentation is available for the project?
Releases	How many production releases of the project have there been in its history?
Perceived health	What is the perceived health of the project?
STRATEGY	
License	What is the licensing strategy? What open source license is used?
Platforms supported	What platforms are supported by the current release?
Security	Is there a formal process for handling security issues? What is the process?
Governance	How is the project governed?
Vision	How clear is the project's vision?
Competition	How many open source projects compete in this category?
MARKET PRESENCE	
User adoption	How much has the project been adopted?
Vendor adoption	What is the vendor support for the project?
Commercial support	How many companies offer end user support for the product? Is there one company primarily identified with the project that provides end customer support?

Source: Forrester Research, Inc.

### Notable Open Source Projects

Forrester included 13 open source projects in the assessment: Apache Cocoon, Apache Geronimo, Apache HTTP Server, Apache Struts, Apache Tomcat, Apache Velocity, Eclipse IDE, Hibernate, JBoss Application Server, MySQL, PHP, PostgreSQL, and Spring Framework. Each of these projects has:

- **Utility in application development and deployment.** Compared with previous Forrester evaluations of open source software, this evaluation was more focused on application development. Open source projects used by application developers to create business applications and platforms on which to deploy them were included. Also included were development tools, middleware, frameworks, and Web and application servers. Open source

projects like operating systems (e.g., Linux), utilities (Samba), and applications (OpenOffice.org) that had previously been included in Forrester's Wave evaluation of open source software were not included in this evaluation.

- **A significant Web presence.** Since the software is available from many different locations that don't measure activity, it is hard to measure actual adoption of open source projects. The number of Web page hits from popular search engines was used as a proxy for the amount of interest and adoption of an open source project. These results typically included discussions, news articles, conference presentations, blogs, reports, documentation, and other artifacts that reflected the overall amount of material available on the Web that related to the project.
- **Industry buzz.** Developers, architects, and managers talk about their experiences with open source software in many venues, including blogs, meetings, phone calls, conferences, and the press. These exchanges leave impressions or perceptions about which projects are hot and which are not. Perception was also used in the selection of projects for this evaluation.

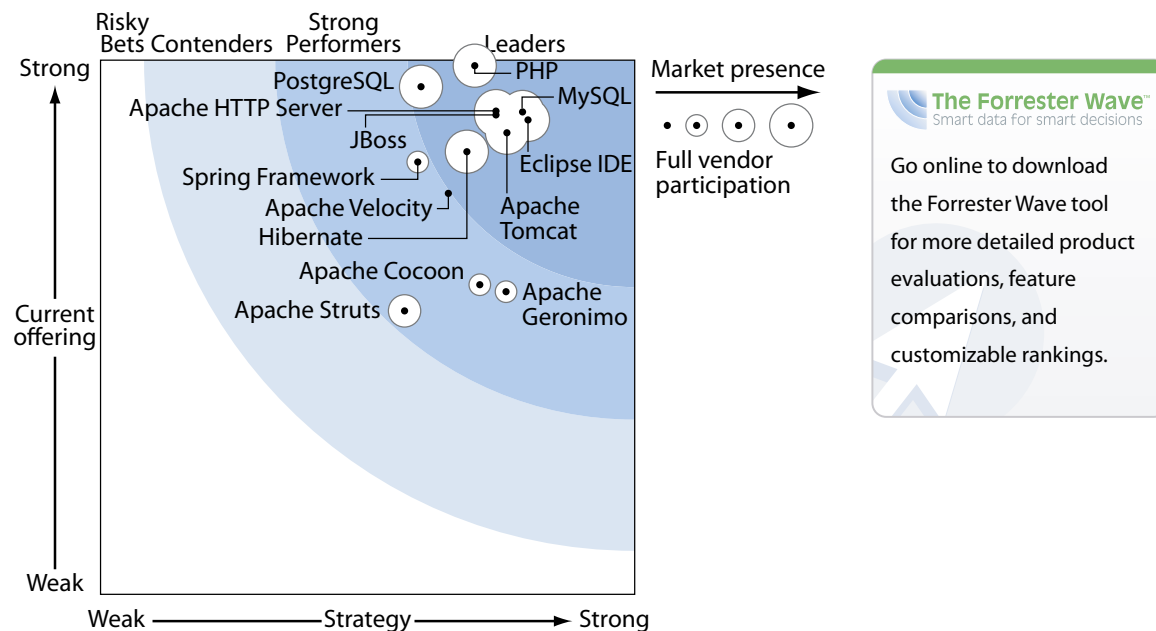
### MANY LEADING OPEN SOURCE PROJECTS EXIST

Our evaluation landed six of the 13 projects in the Leaders category. Three of these six — Apache HTTP Server, MySQL, and PHP — are components of the LAMP stack and have a long history of extensive use by developers. To some extent, these three projects pioneered the way for other open source projects to gain acceptance by the industry at large. Eclipse made it to the front of the pack based on its strong professional governance and widespread support among vendors. JBoss' Professional Open Source model has become a model for many open source companies. The positioning of the 13 projects reveals that (see Figure 2):

- **MySQL, Eclipse IDE, Apache Tomcat, JBoss Application Server, Apache HTTP Server, and PHP are exemplary.** These projects not only have the widest adoption but also their communities are vibrant and active, producing a regular stream of updates, training materials, and attractive service suppliers and complementary products. It is not a coincidence that two of these projects have commercial backers, MySQL and JBoss, giving them extra resources and motivation. Eclipse benefits from strong professional leadership, and PHP has a highly visible commercial supporter. All of these factors suggest keys to success for an open source project.
- **Hibernate and Apache Velocity trail in the Leaders category.** Hibernate is a project that demonstrates one of the foundational principles of open source: focus on one thing and do it well. It meets developers' need for a tool to use in database programming, and it meets that need well. As a result, it is widely used and supported. Velocity is an example of a project that achieved its goals and continues to be used, even though it isn't being actively updated. Its history and use has given it high visibility, and the project has many strengths, but it lags the Leaders and can be expected to decline in popularity as newer technologies emerge on the scene.

- PostgreSQL and Spring Framework are Strong Performers.** PostgreSQL and Spring are both strong projects, but for several reasons, they fall just outside the Leaders category. PostgreSQL is competing head-to-head with a commercially sponsored open source product but does not itself have a visible commercial entity behind it. While this doesn't affect its value as an open source project, it does put it at a disadvantage vis-à-vis MySQL. Spring is a relatively new project whose popularity and usage is on the rise. It is likely that after a few years of continued adoption and support, it will be among the Leaders in the industry.
- Apache Struts, Cocoon, and Geronimo lag in the Strong Performers category.** Of the 13 projects evaluated, Struts, Cocoon, and Geronimo clearly lagged, although for different reasons. Struts' technology has been co-opted by new Java standards that are forcing a re-architecting of the Struts project. This has led to a decline in the use of the current Struts and decreased interest in the current project. Cocoon is a project whose governance has not been able to drive widespread adoption or partner or vendor interest. Cocoon has been in use for many years and, as a result, is well known; but it does not appear to be on a strong growth path. Geronimo, on the other hand, is a relatively new project that generated a lot of attention while it was in development. Even though the project only achieved production status in January of this year, much was known and written about it. This is largely because of its status as an Apache top-level project. IBM's adoption of Geronimo within its product line has also helped ensure the Geronimo project's health.

**Figure 2** Forrester Wave™: Open Source Projects, Q2 '06



Source: Forrester Research, Inc.

**Figure 2** Forrester Wave™: Open Source Projects, Q2 '06 (Cont.)

	Forrester's Weighting	Apache Cocoon	Apache Geronimo	Apache HTTP Server	Apache Struts	Apache Tomcat	Apache Velocity	Eclipse IDE	Hibernate	JBoss App Server	MySQL	PHP	PostgreSQL	Spring Framework
<b>CURRENT OFFERING</b>	60%	2.84	2.78	4.49	2.65	4.30	3.75	4.42	4.14	4.46	4.48	4.91	4.73	4.02
History	10%	4.00	0.50	5.00	3.00	3.50	3.00	2.50	3.00	3.00	5.00	5.00	5.00	1.50
Current activity	30%	4.55	3.75	3.60	2.80	4.30	2.55	4.60	3.90	4.85	3.95	4.70	4.40	3.35
Training and publications	15%	3.80	1.80	4.40	3.00	4.10	2.90	5.00	4.10	4.40	5.00	5.00	4.40	4.10
Releases	10%	5.00	0.00	5.00	5.00	5.00	5.00	5.00	3.00	5.00	5.00	5.00	5.00	5.00
Perceived health	35%	0.00	3.80	5.00	1.60	4.40	5.00	4.40	5.00	4.40	4.40	5.00	5.00	5.00
<b>STRATEGY</b>	40%	3.55	3.80	3.70	2.90	3.80	3.50	4.00	3.40	3.70	3.95	3.50	3.00	3.00
License	20%	4.00	4.00	4.00	4.00	4.00	4.00	4.00	3.00	3.00	5.00	4.00	4.00	4.00
Platforms supported	20%	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
Security	10%	3.00	3.00	3.00	0.00	3.00	3.00	1.00	1.00	5.00	5.00	5.00	3.00	3.00
Governance	20%	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	3.00	3.00	1.00
Vision	20%	3.75	5.00	4.50	2.00	4.00	3.50	5.00	5.00	4.50	3.75	4.50	3.00	5.00
Competition	10%	1.00	1.00	1.00	1.00	3.00	1.00	5.00	1.00	1.00	1.00	1.00	1.00	1.00
<b>MARKET PRESENCE</b>	0%	2.30	2.40	5.00	3.20	4.70	0.15	5.00	4.70	4.70	5.00	4.70	4.55	2.25
User adoption	30%	0.50	0.50	5.00	4.00	5.00	0.00	5.00	5.00	5.00	5.00	4.00	4.50	3.50
Vendor adoption	30%	0.50	3.50	5.00	0.00	4.00	0.50	5.00	4.00	4.00	5.00	5.00	4.00	0.00
Commercial support	40%	5.00	3.00	5.00	5.00	5.00	0.00	5.00	5.00	5.00	5.00	5.00	5.00	3.00

All scores are based on a scale of 0 (weak) to 5 (strong).

Source: Forrester Research, Inc.

This evaluation is intended to be a starting point only. Readers are encouraged to view detailed product evaluations and adapt the criteria weightings to fit their individual needs through the Forrester Wave Excel-based vendor comparison tool.

## PROJECT PROFILES

### Leaders

There was a tightly grouped collection of projects that were clearly Leaders in the Wave assessment, with two projects, Hibernate and Apache Velocity, only slightly trailing the other six. Therefore, the eight Leaders are listed below alphabetically.

- **Apache HTTP Server.** The Apache HTTP Server project launched the Apache Software Foundation (ASF). The advantages of the foundation have led to the creation of new projects to the point where, today, ASF is a leading provider of open source software. The Apache HTTP Server continues to hold a dominant position on the Web, with estimates as high as 70 percent share, and its extensible architecture, using Apache Modules — or mods — has attracted broad support.<sup>3</sup>
- **Apache Tomcat.** Apache Tomcat is the reference implementation of the Java Servlet and JavaServer Pages specifications, and it has achieved widespread use among those who choose not to use the full J2EE functionality, particularly for Web applications. Tomcat has broad support in the industry, with many options for service, support, and training. Tomcat's position as a de facto standard platform for Java Servlet applications and its strong ecosystem make it a favorite target for Java developers.<sup>4</sup>
- **Eclipse IDE.** The Eclipse Foundation was founded after IBM donated code that launched the Eclipse project. The foundation assumed responsibility for evolving both the Eclipse IDE as well as complementary technologies. Since then, it has steadily increased the scope of its charter and is able to do so because of its full-time professional governance. The many Eclipse projects, including the Eclipse IDE, are staffed by volunteers, many of whom hold full-time positions at major vendors like IBM, Novell, and Hewlett-Packard but are assigned to work on Eclipse projects. Nevertheless, Eclipse IDE is distributed under an open source license and has a large and active community supporting it. Many commercial products take advantage of the liberal Eclipse Public License (EPL) and are based on Eclipse technology.<sup>5</sup>
- **Hibernate.** Hibernate is an object/relational mapping component that is in wide use among Java programmers today. It began as an independent project but was brought into JBoss by its founder and is now managed within the JBoss organization. It continues as an open source project and has an active community, which provides bug reports and feature requests as well as training and support.<sup>6</sup>
- **JBoss Application Server.** The JBoss Application Server is a J2EE-certified application server that comprises one part of the JBoss Enterprise Middleware Suite. The JBoss Application Server is distributed under the Lesser General Public License (LGPL) only. While this is more commerce-friendly than the General Public License (GPL), it is still more restrictive than the more permissive BSD license, Apache 2.0 license, or Eclipse Public License used by other projects. However, its development and evolution is under control of JBoss, which was recently acquired by Red Hat. The JBoss community does participate to a limited degree in JBoss Application Server development, but most of it is done by JBoss employees. The greater community will submit bug reports, fixes, and feature requests. JBoss Application Server has achieved wide usage, initially as an inexpensive, standards-based development platform but more recently as a full-fledged J2EE deployment platform.<sup>7</sup>

- **MySQL.** MySQL is an open source relational database that is one component of the popular LAMP stack. It became a standard part of many Web sites as dynamic content began to replace static HTML pages. The product evolved and has begun to approach parity with commercial database products through the efforts of its parent company, MySQL AB. MySQL's high score is largely attributable to the stability and resources available because of the backing of the MySQL company. Unlike many open source projects, the code for MySQL is developed almost exclusively by employees of MySQL AB. However, it is distributed under both an open source license and commercial licenses, making it a genuine open source product.<sup>8</sup> MySQL AB engages its community of users, receiving bug reports, feature requests, and sometimes fixes, but doesn't rely on its community for contributions of code to the project.<sup>9</sup>
- **PHP.** PHP is an open source scripted programming language that is ubiquitous on the Web, but it's often used for many other types of database applications as well. An active community has continuously evolved PHP — even object-oriented features to make PHP more competitive with other programming languages — and there is broad industry support for it. PHP is distributed along with most Linux distributions, and it works with virtually all open source and commercial databases. There are many training resources for people learning PHP development, as well as a large number of PHP-scripted components developers can use in their applications.<sup>10</sup>
- **Apache Velocity.** Apache Velocity is a Java-based template engine that makes it easy to create dynamic Web content. The goal of the project is to make it easy for developers to render data from Java objects in a variety of formats, such as XML, HTML, or even SQL and PostScript. The Apache Velocity project is mature and stable, so much so that it has been several years since the project has released a new version. The existing version continues to be used by developers because of its simplicity and straightforward approach to creating templated Web content. Velocity development is often used as an alternative to using a scripting language like PHP.<sup>11</sup>

### Strong Performers

- **PostgreSQL.** The PostgreSQL project has a very long history, extending back to pioneering academic work in object/relational database technology. The PostgreSQL open source community emerged in 1995, adopting Postgres95 — the first version to include SQL support — as its foundation. The community is strong and stable but has only recently begun to establish close ties with major vendors like Sun Microsystems — which can help expand adoption by companies. The project competes head-to-head with MySQL for many database applications.<sup>12</sup>
- **Spring Framework.** Working with Spring, developers create business logic as POJOs that have no dependencies on the Spring container and little code other than pure business logic. Using an approach called Inversion of Control (IoC), Spring manages transactions, security, and other housekeeping tasks, calling the business logic at the correct point within the flow of control.

Spring's community is managed by a single company but is a very open, active, and stable community. The Spring project has been available for just a few years and will likely become stronger once it has been around for a longer period.<sup>13</sup>

- **Apache Geronimo.** Apache Geronimo is a J2EE 1.4 standard-compliant application server that uses a microkernel architecture with various services implemented by other projects in what is called the Geronimo Mosaic. The project just released version 1.0 in January 2006, so, although it received a lot of publicity because of its status as the Apache Software Foundation's Java application server, its actual market penetration is small at present. It has a strong and active community, aided by the acquisition by IBM of Gluecode, a company comprised of several key Geronimo committers. IBM's incorporation of Geronimo in its WebSphere application server product line also lends a lot of credibility to the Apache Geronimo project.<sup>14</sup>
- **Apache Cocoon.** The Apache Cocoon project is an XML Web framework designed to assist developers creating complex Web applications. Designed around the notion of pipelines, it is very useful for transforming data from one format into another. Cocoon has a strong and active community; however, it is relatively small, and the overall adoption of Cocoon has not been widespread. Also, there is no commercial entity behind Apache Cocoon to increase its market penetration. And it has substantial competition from other open source and commercial products, further limiting its adoption.<sup>15</sup>
- **Apache Struts.** Apache Struts was one of the earlier efforts to simplify the development of Web applications in Java. It pioneered many concepts and had been widely adopted. The project's community had been large and active, but both its user base and community are in decline. Struts' capabilities are being supplanted by JavaServer Faces, recently incorporated into the Java standards. This has forced the existing Struts community to rethink the Struts architecture and to begin to look at developing a next-generation Struts.<sup>16</sup>

## RECOMMENDATIONS

### OPEN SOURCE SOFTWARE IS READY FOR ENTERPRISE USE

In the past, decision-makers involved in making tactical and strategic choices for application infrastructure and tools faced two questions when open source software came up. The first was, "Is open source software ready for prime time?" and the second, "Is <fill in the name of an open source project being considered> ready for prime time?" It is time to put aside the initial screening question. These 13 projects clearly demonstrate that open source projects can produce software that is completely acceptable in enterprise environments. The approach to take when evaluating open source projects today is:

- **Use these evaluations in context.** All of the 13 projects evaluated in this Wave are suitable for their intended purpose, and companies should be able to successfully employ them in development. However, as is the nature of open source projects, they are generally focused on performing a smaller set of tasks well, as opposed to providing broadly based functionality, as is often the case with commercial products. Make sure you choose the projects that best fit your needs rather than fitting your needs to the capabilities of the project.
- **Make decisions on a project-by-project basis.** The suitability of an open source project depends on its ability to meet the specific criteria your company establishes for selecting a project. It is not based on a conceptual-level evaluation of the maturity of open source software.
- **Follow established evaluation methodologies.** With so many projects to choose from, it would be extraordinarily wasteful if every company started at square one assembling its own assessment methodology. While Forrester will continue to add open source projects to its Wave evaluations, we will never be able to evaluate the thousands that are worthy of consideration. Your company can adapt the Forrester Wave methodology and use it as a starting point for doing your own evaluations.

### USE THIS WAVE AS AN EVALUATION MODEL

As you evaluate whether to use other open source software projects beyond those covered here, use this Forrester Wave as a model and evaluate other projects against our criteria. To help you assess the results, Forrester has identified three major types of open source projects as well as benchmark projects for each:

- **Foundation- and consortium-sponsored projects.** The evaluated Eclipse Foundation and Apache Software Foundation projects represent a much larger group of open source projects that come under the governance of these foundations. There are several other

foundation consortia that play a similar role. The strong governance model and professional management lend a great deal of credibility to these projects in addition to freedom from excess vendor influence.

- **Vendor-sponsored projects.** JBoss Application Server, Hibernate, MySQL, and, to some extent, Spring Framework represent a growing number of open source projects that are sponsored by a vendor that often licenses this software under a dual license. These vendors can lend resources to the project in the form of developers, marketing, and professional management, but the tradeoff is limited community participation.
- **Community-based projects.** PostgreSQL and PHP represent the large number of projects that are pure community-based. This is the traditional open source model that is based on volunteers and no outside influence. These projects can be every bit as effective as projects in the other category, but the odds tend to work against them because the volunteer nature of participation is hard to predict.

You can accelerate your evaluation process by understanding where a project fits in this taxonomy and focusing your attention on where a project's strengths and weaknesses may lie in comparison to one of the benchmark projects listed above.

## SUPPLEMENTAL MATERIAL

### Online Resource

The online version of Figure 2 is an Excel-based vendor comparison tool that provides detailed product evaluations and customizable rankings.

### Data Sources Used In This Forrester Wave

Forrester used a combination of three data sources to assess the strengths and weaknesses of each solution:

- **Project surveys.** Forrester surveyed projects on their capabilities as they relate to the evaluation criteria. Once we analyzed the completed project surveys, we conducted calls with representatives of the projects where necessary to gather details of vendor qualifications. Respondents for most projects were volunteers who were regular contributors. In the case of JBoss, Hibernate, and MySQL, the person responding to the survey was an employee of the company.
- **Developer reference calls.** To validate project qualifications, Forrester also conducted reference calls with 10 users of open source software. Most projects did not have a means for supplying references, so we used other means to identify users of open source, including referrals from other projects and people known to Forrester as users of open source. In all cases, those called used multiple open source projects in the course of their development work.
- **Review of online resources.** One characteristic of open source projects is their transparency — little information is held privately. We were able to review developer and user email list archives, bug tracker databases, and project road maps, as well as information that had been published to the Web, including in blogs and discussions. In the case of information obtained from the Web, any information had to be corroborated by at least one independent source before it was used.

### The Forrester Wave Methodology

Evaluating the health of open source projects required that we modify the standard Forrester Wave methodology for several reasons:

1. In most — but not all — cases, there is no single vendor promoting an open source project.
2. Project leaders are often, but not always, volunteers.
3. Customer references are often not available, because projects often do not have a direct method for determining who is using their technology.

In addition, this Wave compared different technologies, including tools, databases, application servers, and frameworks; therefore, a comparison of features was not relevant. This Wave focused on the health, maturity, and viability of the open source projects reviewed.

We conducted primary and secondary research to develop a list of open source projects that had high visibility and were relevant to application developers. We chose those projects based on:

1. Search engine visibility
2. Direct role in application development
3. Degree to which the project was representative of open source projects in general

After examining past research, user need assessments, and vendor and expert interviews, we developed the initial evaluation criteria. To evaluate the projects against our set of criteria, we gathered details of product qualifications through a combination of questionnaires, interviews, secondary research, and/or discussions with client references. We sent evaluations to the vendors for their review, and we adjusted the evaluations to provide the most accurate view of vendor offerings and strategies.

We set default weightings to reflect our analysis of the needs of large user companies — and/or other scenarios as outlined in this document — and then scored the projects based on a clearly defined scale. These default weightings are intended only as a starting point, and readers are encouraged to adapt the weightings through the Excel-based tool to fit their individual needs. The final scores generated the graphical depiction of the market based on current offering, strategy, and market presence. Forrester intends to update project evaluations regularly as product capabilities and vendor strategies evolve.

## ENDNOTES

- <sup>1</sup> See the June 23, 2005, Trends “[Open Source Usage Is Up, But Concerns Linger](#).”
- <sup>2</sup> The method used to narrow down the field included polling Forrester analysts familiar with open source projects that address application development; informal discussions with developers; assessments of the amount of Web content about projects; and a brief review of the currency of content, recent release activity, and amount of training material available. Thirteen projects emerged as the Leaders as a result of this process.
- <sup>3</sup> View the vendor summary for more detailed analysis on how Apache HTTP Server fared in this evaluation. See the June 28, 2006, Tech Choices “[Apache HTTP Server Is A Strong Leader In Open Source Projects](#).”
- <sup>4</sup> View the vendor summary for more detailed analysis on how Apache Tomcat fared in this evaluation. See the June 28, 2006, Tech Choices “[Apache Tomcat Is A Strong Leader In Open Source Projects](#).”

- <sup>5</sup> View the vendor summary for more detailed analysis on how Eclipse IDE fared in this evaluation. See the June 28, 2006, Tech Choices “[Eclipse IDE Is A Strong Leader In Open Source Projects.](#)”
- <sup>6</sup> View the vendor summary for more detailed analysis on how Hibernate fared in this evaluation. See the June 28, 2006, Tech Choices “[Hibernate Is A Leader In Open Source Projects.](#)”
- <sup>7</sup> View the vendor summary for more detailed analysis on how JBoss Application Server fared in this evaluation. See the June 28, 2006, Tech Choices “[JBoss Application Server Is A Strong Leader In Open Source Projects.](#)”
- <sup>8</sup> For current purposes, an open source license is defined as a license that has been reviewed by the Open Source Initiative (OSI) and is listed on its Web site. This should not be taken to mean that OSI listing is necessary for a license to be considered by Forrester Research to be an open source license, but we consider it to be sufficient. OSI-approved licenses can be found at <http://www.opensource.org>.
- <sup>9</sup> View the vendor summary for more detailed analysis on how MySQL fared in this evaluation. See the June 28, 2006, Tech Choices “[MySQL Is A Strong Leader In Open Source Projects.](#)”
- <sup>10</sup> View the vendor summary for more detailed analysis on how PHP fared in this evaluation. See the June 28, 2006, Tech Choices “[PHP Is A Strong Leader In Open Source Projects.](#)”
- <sup>11</sup> View the vendor summary for more detailed analysis on how Apache Velocity fared in this evaluation. See the June 28, 2006, Tech Choices “[Apache Velocity Is A Leader In Open Source Projects.](#)”
- <sup>12</sup> View the vendor summary for more detailed analysis on how PostgreSQL fared in this evaluation. See the June 28, 2006, Tech Choices “[PostgreSQL Is A Strong Performer In Open Source Projects.](#)”
- <sup>13</sup> View the vendor summary for more detailed analysis on how Spring fared in this evaluation. See the June 28, 2006, Tech Choices “[Spring Framework Is A Strong Performer In Open Source Projects.](#)”
- <sup>14</sup> View the vendor summary for more detailed analysis on how Apache Geronimo fared in this evaluation. See the June 28, 2006, Tech Choices “[Apache Geronimo Is A Strong Performer In Open Source Projects.](#)”
- <sup>15</sup> View the vendor summary for more detailed analysis on how Apache Cocoon fared in this evaluation. See the June 28, 2006, Tech Choices “[Apache Cocoon Is A Strong Performer In Open Source Projects.](#)”
- <sup>16</sup> View the vendor summary for more detailed analysis on how Apache Struts fared in this evaluation. See the June 28, 2006, Tech Choices “[Apache Struts Is A Strong Performer In Open Source Projects.](#)”

# FORRESTER®

Helping Business Thrive On Technology Change

## Headquarters

Forrester Research, Inc.  
400 Technology Square  
Cambridge, MA 02139 USA  
Tel: +1 617/613-6000  
Fax: +1 617/613-5000  
Email: [forrester@forrester.com](mailto:forrester@forrester.com)  
Nasdaq symbol: FORR  
[www.forrester.com](http://www.forrester.com)

## Research and Sales Offices

Australia	Israel
Brazil	Japan
Canada	Korea
Denmark	The Netherlands
France	Switzerland
Germany	United Kingdom
Hong Kong	United States
India	

*For a complete list of worldwide locations,  
visit [www.forrester.com/about](http://www.forrester.com/about).*

For information on hard-copy or electronic reprints, please contact the Client Resource Center at +1 866/367-7378, +1 617/617-5730, or [resourcecenter@forrester.com](mailto:resourcecenter@forrester.com). We offer quantity discounts and special pricing for academic and nonprofit institutions.

Forrester Research (Nasdaq: FORR) is an independent technology and market research company that provides pragmatic and forward-thinking advice about technology's impact on business and consumers. For 22 years, Forrester has been a thought leader and trusted advisor, helping global clients lead in their markets through its research, consulting, events, and peer-to-peer executive programs. For more information, visit [www.forrester.com](http://www.forrester.com).